SIstema de Classificação Automática de DEP Bovina (SICADEB)

Andrea Gondo

Mestrado Profissional em Computação Aplicada (FACOM/UFMS)

Data de Depósito: ___/___/_____

Assinatura: _____

SIstema de Classificação Automática de DEP Bovina (SICADEB)

Andrea Gondo

Orientador: *Prof. Dr. Edson Takashi Matsubara*
Coorientador: *Dr. Luiz Otavio Campos da Silva*

Dissertação apresentada ao Programa de Mestrado Profissional em Computação Aplicada da Faculdade de Computação (FACOM/UFMS) como parte dos requisitos para obtenção do título de Mestre em Computação Aplicada.
Área de Concentração: Tecnologias Computacionais para Agricultura e Pecuária
Linha de Pesquisa: Sistemas Embarcados e Robótica

Campo Grande - MS
28 de março de 2016

Aos meus pais, Mituo e Anice, pelo que se dedicam a mim

Ao grande amigo Dr. Luiz Otavio, por acreditar

Agradecimentos

Obrigada Deus pela minha vida, por tudo que sou...

Obrigada a meu pai e minha mãe, são tudo para mim, são meu porto seguro, minha certeza de amor incondicional. Seus esforços zelando pelo bem da nossa família me fazem a pessoa mais afortunada do mundo, só tenho a agradecer...

Obrigada minha irmã querida, professora doutora, sabe bem dos altos e baixos do percurso até aqui. Sei que estará sempre ao meu lado para o que der e vier; obrigada a seu marido, pelo carinho e torcida; obrigada pela alegria da minha vida, meu afilhadinho Gabriel...

Obrigada Professor Edson Takashi Matsubara por me aceitar como sua orientada. Uma honra e satisfação muito grande. Obrigada por nunca desistir de mim, por sua dedicação e disposição. Sempre otimista, em muitos emails você encerrava com "ganbatte!", algo como, "vai dar certo". Em um dado momento, foi uma palavra de incentivo que realmente fez a diferença. Espero um dia poder retribuir tamanha consideração...

Obrigada Dr. Luiz Otavio, por ser meu meu coorientador neste trabalho e meu orientador de 20 anos de trabalho lado a lado. Sua amizade e aconselhamentos são valiosos para mim. Obrigada por estar em meu caminho, me ajudando a ser melhor, me motivando com seu modo de viver, trabalhar e persistir. Obrigada por acreditar em mim mais do que eu mesma. Obrigada à Valéria, sua esposa, e toda a sua grande família pelo carinho de sempre...

Obrigada Dr. Paulo Nobre, pesquisador do Geneplus, a quem recorri tantas vezes para solicitar informações sobre as avaliações genéticas do Programa Geneplus. Obrigada por toda ajuda, amizade e ensinamentos...

Obrigada ao Programa Geneplus, nas pessoas do Dr. Luiz Otavio e Dr. Paulo Nobre, pela cessão dos dados para a pesquisa...

Obrigada aos professores da FACOM pela minha formação. Em especial aos professores Eraldo Luís Rezende Fernandes e Bruno Magalhães Nogueira que participaram da banca, obrigada pelas dicas e apoio. Obrigada aos professores Gedson Faria, Edson Cáceres, Luciano Gonda, Nalvo, Nahri, Irineu, Kátia,

Marcelo Henriques, e a todos os outros. Obrigada Marcos Iseki, Rosiane e Paulo. Obrigada a todos os técnicos...

Obrigada Dr. Roberto Torres Jr, pesquisador da Embrapa, pela paciência em explicar as minhas dúvidas...

Obrigada à Embrapa Gado de Corte por ser minha segunda casa. Ao Dr. Cleber Soares, chefe geral da Embrapa Gado de Corte, pelo incentivo e apoio de sempre...

Obrigada aos amigos do Programa Geneplus, Dr. Leonardo Nieto, Raul, Raphael, Renato, Danila, Lucas, David e todos os técnicos; obrigada aos amigos da Embrapa, Dr. Antonio Rosa, Fatinha, Gilberto, Camilo, Carlo, Josenei e, em especial, a Nilza da biblioteca que tanto me ajudou com as referências bibliográficas...

Obrigada à minha grande amiga Claudia Fujie, pelo apoio e palavras de incentivo; à amiga Diná Tamasiro porque me apresentou a Embrapa em 1995, por estar sempre na torcida; à amiga Christiane Nishibe pela disposição em ajudar...

Obrigada Izaura por cuidar de minha casa e da gente que nela mora...

Obrigada às minhas tias queridas que lá em Bauru se preocupam tanto comigo aqui... Hoje sou tia, sei como é... Obrigada à minha avó Gondo pelo exemplo de força e vivacidade. Obrigada a todos os meus parentes Gondo, Hara, Maeda e Hira, que fazem parte de mim.

Obrigada ao compositor Yiruma porque era com suas músicas que me concentrava nos estudos (*River Flows in You*)...

Obrigada meu Deus por todas as pessoas que cercam...

"Com meus amigos, sou rico"
William Shakespeare

Resumo

A pecuária de corte, uma das principais atividades econômicas do Brasil, é responsável pela posição de destaque do país entre os grandes produtores mundiais de carne bovina. Considerando a magnitude de sua participação no mercado é crucial que tecnologias que auxiliem o rebanho brasileiro estejam cada vez mais ao alcance do criador, fornecendo diagnósticos antecipados da qualidade do produto que será ofertado, levando em consideração fatores como competitividade, demanda do consumidor, e a relação com o meio ambiente. Há muitas décadas, o melhoramento genético animal vem contribuindo com tecnologias, como Diferença Esperada na Progênie (DEP), para o incremento da produção de carne, assegurando uma seleção de animais alicerçada em critérios. Este trabalho pretende unir resultados de avaliações genéticas entregues a criadores do Programa `GENEPLUS` e técnicas de aprendizado de máquina que resultem em melhor classificação de animais para seleção. Tal classificação trata-se de categorizar a DEP de uma característica em `elite`, `superior`, `regular` e `inferior`. Estas classes, nesta sequência, revelam a ordem dos melhores animais para os piores. Como as avaliações genéticas ocorrem semestralmente, o objetivo deste trabalho é tentar aliviar o tempo de espera deste produtor pelos novos resultados, utilizando técnicas de aprendizado de máquina. Mais especificamente será realizada a indução por árvores de decisão em um banco de dados de programa de melhoramento genético animal para gerar predições confiáveis de classes para a DEP do peso à desmama. Vários experimentos foram feitos, alternando-se entre o pacote de software WEKA (*Waikato Environment for Knowledge Analysis*) e implementações da biblioteca `SKLEARN`. Os resultados dos experimentos mostraram que o uso do classificador traz vantagem para informar as quatro classes conseguindo boas métricas. Baseado nisto, o SIstema de Classificação Automática de DEP Bovina (SICADEB) foi proposto para aplicação do classificador no *software* de retorno de resultados de avaliação genética, os chamados sumários.

Palavras-chave: árvore de decisão, melhoramento animal, escore

Abstract

Livestock is one of the most important economic activity in Brazil, it is responsible for the highlighting position of the country among the world biggest bovine meat producers. Considering the significance of Brazil's market participation it is crucial to have technologies increasingly present in everyday activities of Brazilian cattle, facilitating the farmer's decision taking provided with predictions of the product quality that will be offered for trading, taking in account factors like competitiveness, end user demand, and environment relationship. For decades, animal breeding has been contributing with technologies, such as Expected Progeny Difference (EPD), for the meat production growth, assuring an animal selection based on criteria. This work intends to join genetic evaluation results, that are already delivered to farmers by `GENEPLUS` Program, with learning machine techniques, in order that improvement succeed on animals' classification performance. This classification is based on categorizing the EPD in `elite` class, `superior` class, `regular` class and `inferior` class. Following this sequence, these classes mean ordering for the best to the worst animals. As genetic evaluations take place one by semester, the objective of this work is to try to soften the waiting time of the farmer for the next new results, by applying machine learning techniques, specifically decision trees induction, in an animal breeding program data base, so as to generate reliable predictions for EPD classes for weaning weight trait. Several experiments were executed, interchanging WEKA (*Waikato Environment for Knowledge Analysis*) and `SKLEARN` library. The results proved that using the classifier is useful to inform the four classes and that the metrics are good also. Based on this fact, a system was proposed, the Automatic Classification System of Bovine EPD (SICADEB) and is to apply the classifier as a new module in the software that the farmers consult the genetic evaluation results, the summaries.

Sumário

Lista de Figuras

Lista de Tabelas

CAPÍTULO

1 Introdução

A pecuária de corte, uma das principais atividades econômicas do Brasil, é responsável pela posição de destaque do país entre os grandes produtores mundiais de carne bovina. Há muitas décadas, o melhoramento genético animal vem contribuindo com tecnologias, como *Diferença Esperada na Progênie* (DEP), para o incremento da produção de carne, assegurando ao criador uma seleção de animais alicerçada em critérios.

Para que as Diferenças Esperadas nas Progênies possam ser geradas, são necessários bancos de dados nos quais são depositadas as informações sobre a árvore genealógica da população em avaliação e sobre as características que serão acompanhadas (pesos, medidas, idades, escores, etc), seja de um rebanho ou de vários. Participantes de programas de melhoramento abastecem estes bancos de dados e esperam ganhar subsídios para aplicar a seleção em seu plantel.

O banco de dados do *Programa Embrapa de Melhoramento de Gado de Corte* - GENEPLUS é riquíssimo em dados históricos e desde 1996, ano de seu lançamento ao mercado, vem sendo alimentado por criadores associados. Contando com cerca de 1.900.000 registros de animais, passa por duas avaliações genéticas ao ano. Os processamentos envolvem técnicas de limpeza e padronização dos dados, além de metodologias estatísticas complexas, mas não houve ainda um exercício de aprendizado de máquina nesta base.

Este trabalho pretende unir o que já está em uso, em se tratanto de resultados de avaliações genéticas entregues a criadores participantes do Programa GENEPLUS, e técnicas de aprendizado de máquina que melhorem o desempenho da classificação de animais.

Tal classificação trata-se de categorizar a DEP de uma característica, tal como *Peso à Desmama*, nas classes `elite`, `superior`, `regular` e `inferior`, que nesta sequência, revelam a ordem dos melhores animais para os piores.

1.1 Contextualização

Segundo dados do Ministério da Agricultura, Pecuária e Abastecimento[1], o Brasil aparece como segundo maior rebanho de bovinos do mundo, criador de cerca de 200 milhões de cabeças de gado, perdendo apenas para a Índia em quantidade de carne exportada por ano.

Considerando o tamanho da população bovina do país e a magnitude de sua participação no mercado mundial, é crucial que tecnologias aplicadas à pecuária de corte estejam cada dia mais presentes no rebanho brasileiro.

O melhoramento genético animal é uma das áreas de pesquisa que muito tem contribuído com tecnologias para o incremento da produção de carne bovina. Propõe-se a assegurar que a aparência não mascare a produtividade (Daly, 1977), ou seja, que a seleção dos melhores animais seja feita não pela expressão visível do animal, mas pela sua capacidade em transmitir qualidade genética a seus descendentes.

É um investimento de longo prazo. Desde o acasalamento, passando pelo nascimento, fases de pesagens, início da vida reprodutiva, até chegar na produção dos descendentes com observações válidas para uma avaliação genética mais completa, é um processo demorado. Por isso, quanto mais ferramentas auxiliares que acelerem a extração de informação, melhor para o criador, pois tem a chance de conhecer a real situação de seu rebanho e servir-se disso para buscar caminhos para o progresso genético desejado.

O banco de dados do Programa `GENEPLUS` é riquíssimo em dados históricos e desde 1996, ano de seu lançamento ao mercado, vem sendo alimentado por criadores associados que esperam justamente por essa compensação em informações que agilizem o ciclo de produtividade dos animais.

Com a oportunidade de poder utilizar técnicas de aprendizado de máquina nesse banco de dados do Programa `GENEPLUS`, muitas possibilidades surgiram, e a mais apelativa foi uma relacionada ao retorno da informação para o usuário. Para agilizar o processo de escolha dos animais, ainda no curral, seria possível treinar um classificador que pudesse distinguir animais promissores, baseando-se no peso e nos dados de seus pais?

[1] http://www.agricultura.gov.br/animal/especies/bovinos-e-bubalinos

1.2 Motivação e Justificativa

As avaliações genéticas são realizadas uma a cada semestre, em prazos que contemplam épocas de grandes coletas de dados pela maioria dos produtores. Então, chegada a época dos envios dos dados, cada produtor envia o seu *backup* para o Programa `GENEPLUS` que processará as avaliacões de todas as fazendas juntas.

Se um produtor não enviar seus dados, ou se algum lote de animais ficar sem registro de pesagem, ele somente poderá atualizar o Programa `GENEPLUS` com seus dados na próxima avaliação, não há como fazer avaliações genéticas sempre que um novo *backup* chega. Ainda não se pode fazer isso.

Sendo assim, na tentativa de aliviar o tempo de espera deste produtor que necessita saber a avaliação genética de seus produtos sem que tenha que esperar pela próxima edição dos resultados, é que se pensou em utilizar alguma técnica da área de Aprendizado de Máquina que trouxesse mais informações ou informações mais acertadas baseadas no conhecimento existente no banco de dados do Programa `GENEPLUS`.

O mercado, a competitividade, a demanda do consumidor, a relação com o meio ambiente são alguns exemplos de fatores que influenciam no incremento de produtividade. Por isso, havendo técnicas que colaborem para a qualidade final do produto, estas devem ser testadas e apresentadas aos usuários que necessitam de subsídios para tomada de decisão.

1.3 Objetivo

O objetivo deste trabalho é a indução de classificadores automáticos que possam distinguir DEPs de bovinos nas seguintes categorias: `elite`, `superior`, `regular` e `inferior`. O classificador deve ser de fácil interpretação para que se possa justificar a classificação fornecida.

1.4 Organização do Texto

Este trabalho possui cinco capítulos na seguinte forma: o Capítulo 1 trata da introdução e apresentação do contexto em linhas gerais, abordando motivação, justificativa e objetivo; o Capítulo 2 - Materiais e Métodos, aborda conceitos gerais sobre as duas grandes áreas em trabalho conjunto neste estudo, aprendizado de máquina e melhoramento genético animal; no Capítulo 3

- Avaliação Experimental, estão descritos todos os experimentos que levaram ao classificador final e ao formato de arquivo final, com eliminação e acréscimo de atributos; o Capítulo 4 - SICADEB, apresenta o protótipo do sistema que disponibiliza o classificador para o uso dos criadores; e finalmente, o Capítulo 5 conclui este trabalho.

CAPÍTULO

2 Materiais e Métodos

Neste capítulo são apresentados conceitos gerais das duas áreas em estudo, aprendizado de máquina e melhoramento genético animal. Na Seção 2.1, a indução por árvore de decisão, métodos de escolha de atributos e métricas para avaliar classificadores são enfatizados pois foram utilizados nos experimentos e nas análises dos resultados. Para melhor compreender o lado do pesquisador melhorista e do produtor de carne, um resumo básico sobre conceitos de melhoramento animal é apresentado na Seção 2.2. Na Seção 2.3 o funcionamento do Programa `GENEPLUS` é descrito, como é a troca de dados entre produtor e o Programa; na Seção 2.4 os atributos do banco de dados do Programa `GENEPLUS` são descritos e deste banco serão extraídas as bases utilizadas nos experimentos, detalhadas na Seção 2.5.

2.1 Conceitos de Aprendizado de Máquina

Dentro da *Inteligência Artificial* (IA), a sub-área de Aprendizado de Máquina (AM) abrange a pesquisa sobre métodos computacionais para organização e produção de conhecimento a partir de experimentações e observações do mundo real. Como em todo aprendizado, a quantidade de dados deve ser em volume suficiente para gerar informação de qualidade confiável e preparar para produzir respostas inteligentes para novas situações de adaptação ao ambiente e a imprevistos.

A aprendizagem está relacionada a métodos de inferência lógica em que a correta manipulação de conhecimento existente pode levar à descoberta de

novos conhecimentos (Prati, 2006). Os métodos de inferência lógica podem ser agrupados em três classes: dedução, abdução e indução.

Na dedução o conhecimento aprendido é fruto de transformações sobre o conhecimento existente, conservando a veracidade, extraindo informações contidas implicitamente nas premissas. Na abdução, produz-se um conhecimento complementar a um conhecimento particular onde a veracidade das hipóteses está condicionada à veracidade das extensões abdutivas, ou seja, na abdução não há garantias de hipóteses verdadeiras mas sim a probabilidade das mesmas (Batista, 2003).

Na indução, um conceito específico é generalizado partindo de exemplos particulares para conclusões gerais, não há garantias da preservação da veracidade dos resultados pois a inferência gerará hipóteses passíveis de avaliação pelos respectivos graus de confiança baseados na quantidade e na qualidade das premissas. Ao contrário do que sucede com um argumento dedutivo e válido, um argumento indutivo e correto pode admitir uma conclusão falsa, ainda que suas premissas sejam verdadeiras, por isso há que ser usado com cautela pois se o número de observações for insuficiente ou se os dados relevantes forem mal escolhidos, as hipóteses induzidas podem ser de pouco ou nenhum valor (Batista, 2003).

Para melhor ilustrar estes três conceitos, segue um exemplo segundo (dos Santos, 2010):

Exemplo 1

- Todos os ovos da cesta “A” são de galinha. Este ovo é da cesta “A”. Logo, é um ovo de galinha $\rightarrow$ Dedução
- Estes ovos são da cesta “A”. Estes ovos são de galinha. Logo, todos os ovos da cesta “A” são de galinha $\rightarrow$ Indução
- Todos os ovos da cesta “A” são de galinha. Este ovo é de galinha. Logo, este ovo é da cesta “A” $\rightarrow$ Abdução

A partir destes conceitos e dentro da inferência indutiva, um sistema de aprendizado de máquina vem a ser um programa ou um algoritmo ou um indutor que toma decisões analisando exemplos históricos em busca de solução de problemas anteriores (Prati, 2006). A indução é uma das principais vias de descoberta de conhecimento e previsão de eventos futuros, e por isso, uma das formas de inferência mais utilizadas em AM, mesmo sendo as hipóteses geradas passíveis de informações questionáveis.

Dado um conjunto de exemplos a ser submetido em um indutor, este deverá ter passado por um pré-processamento, pois precisa estar em um formato adequado para a análise. São necessários tratamentos para limpeza, transformação dos dados, integração e padronização (caso os dados sejam provenientes de várias fontes), redução do volume dos dados, entre outros, até que cada exemplo do produto final seja uma linha no formato atributo-valor. O resultado final desta transformação é mostrado na Tabela 2.1.

	Atributo 1	Atributo 2	...	Atributo k	Classe
X_1	X_{11}	X_{12}	...	X_{1k}	Y_1
X_2	X_{21}	X_{22}	...	X_{2k}	Y_2
...	...	...	...	...	...
X_n	X_{n1}	X_{n2}	...	X_{nk}	Y_n

Tabela 2.1: Formato da tabela atributo-valor

Na Tabela 2.1, as linhas representam os exemplos e as colunas são os atributos descritores de cada exemplo. O conjunto de linhas é constituído por vetores X_i, com $i = 1, ..., n$, onde n é o número de exemplos de treinamento. Cada vetor possui k atributos com valores contínuos, discretos ou booleanos, relevantes o suficiente para "ensinar" o indutor. O atributo Classe = Y_1, Y_2, ...,Y_n poderá estar presente ou não nesta tabela, dependendo do modo de aprendizado, e caracterizará a amostra como conjunto rotulado ou não rotulado. A Classe é atribuída pela observação do histórico ou é fornecida por um especialista de domínio (Matsubara, 2008) e é a meta do aprendizado.

Dependendo da quantidade de exemplos rotulados com a Classe, ou atributo meta, os modos de aprendizado indutivo podem ser divididos em supervisionado, semi-supervisionado e não-supervisionado. No supervisionado, o algoritmo de aprendizado tem como entrada um conjunto significativo de dados de treinamento já rotulados com a Classe, enquanto que no não-supervisionado não há exemplos rotulados, cabendo ao indutor determinar agrupamentos para exemplos similares. No modo semi-supervisionado há dois conjuntos de dados, um com observações rotuladas e outro sem o atributo Classe, e podem ser utilizados tanto em tarefas de classificação quanto de agrupamento (Sanches, 2003), sendo os exemplos rotulados responsáveis por auxiliar no treinamento do algoritmo.

Na Figura 2.9 a seguir é apresentada a hierarquia do AM, com a subdivisão pelo modo de aprendizado (supervisionado ou não) e os tipos de tarefas realizadas (classificação, regressão, agrupamento, regras de associação, entre outras). Para o caso do aprendizado supervisionado, um problema é dito ser

de classificação quando o algoritmo de indução deve determinar corretamente Classes nominais para novos exemplos ainda não rotulados. Se as Classes a serem determinadas forem de valores contínuos, o problema é então de regressão.

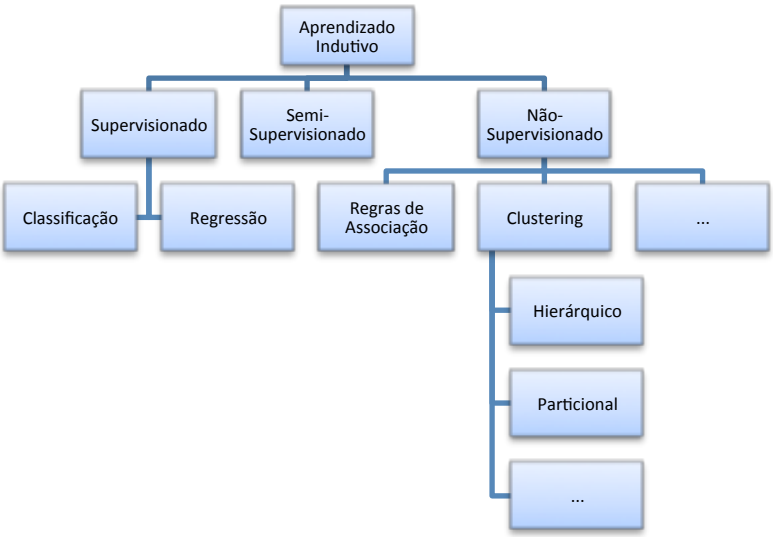


Figura 2.1: Hierarquia do aprendizado indutivo

Além da hierarquia, existem os paradigmas de aprendizado de máquina que agrupam os algoritmos de acordo com o tipo de aprendizado:

1. Simbólico: o aprendizado ocorre via representações simbólicas de um conceito através da análise de exemplos e contra-exemplos deste conceito. Estão na forma de expressão lógica, árvore de decisão, regras ou rede semântica.

2. Estatístico: o aprendizado ocorre utilizando modelos estatísticos para encontrar boa aproximação do conceito induzido. Estão na forma de combinações lineares de atributos, modelos probabilísticos como o Bayesiano.

3. Baseado em exemplos ou *lazy*: tipo de sistema de aprendizado que necessita manter os exemplos na memória para classificar novos exemplos. É necessário saber quais são os exemplos mais representativos para a memorização. São exemplos: K-Vizinhos mais próximos e Raciocínio baseado em casos.

4. Conexionista: o aprendizado envolve conexões numerosas entre unidades, como fossem os neurônios do sistema nervoso. Redes neurais são exemplos de algoritmos de aprendizado neste paradigma.

5. Genético ou evolutivo: o aprendizado ocorre com a evolução dos melhores, ou seja, há uma competição entre os elementos que fazem a predição e somente os melhores permanecem.

Para este trabalho, o foco estará voltado para problema de classificação no aprendizado supervisionado. Abaixo são listados alguns conceitos importantes, segundo Monard e Baranauskas (Monard and Baranauskas, 2003):

Indutor: programa de aprendizado ou algoritmo de indução com o objetivo de gerar um classificador capaz de rotular exemplos novos, ainda não rotulados, o mais corretamente possível. O classificador é, portanto, a saída do indutor;

Exemplo: tupla de valores de atributos que descrevem o objeto de interesse. Por exemplo, o boi é um objeto de interesse, e seus pesos são os valores de atributos e comporão um exemplo de animal;

Atributo : uma característica do exemplo. Pode ser nominal (por exemplo, a categoria do animal: touro, vaca ou produto) ou contínuo (por exemplo, o peso do animal);

Classe ou rótulo : aquilo que se deseja aprender e fazer previsões para novos exemplos. Todo exemplo possui um rótulo e para problemas de classificação, este rótulo será nominal, caso seja um valor contínuo, o problema passa a ser de regressão;

Conjunto de exemplos de treinamento: conjunto de exemplos usado para o aprendizado;

Conjunto de exemplos de teste: conjunto de exemplos usado para testar o classificador e medir a validade do que foi aprendido. Normalmente é disjunto do conjunto de treinamento para que uma avaliação final independente seja feita a partir de dados completamente diferente daqueles usados para o treinamento;

Modo de aprendizado não-incremental: ou modo *batch*, é o modo em que o aprendizado somente se dá com todo o conjunto de treinamento;

Modo de aprendizado incremental: é o modo em que o aprendizado é atualizado cada vez que se submete um novo conjunto de exemplos;

Prevalência de classe: quando o conjunto de exemplos é muito desbalanceado quanto ao número de exemplos por Classe, ocorre que deve haver um cuidado maior na interpretação das métricas obtidas, para que a Classe majoritária não ofusque a importância das Classes minoritárias;

***Overfitting*:** quando o classificador ajusta-se excessivamente ao conjunto de treinamento. O desempenho do classificador é bom para o conjunto de treino e ruim para o conjunto de teste;

***Underfitting*:** quando o classificador ajusta-se muito pouco ao conjunto de treinamento. Pode acontecer quando a amostra é pouco representativa ou quando a árvore de decisão é muito podada, por exemplo, e o classificador fica muito restrito.

Poda : técnica para gerar uma árvore de decisão ou regras de classificação de modo mais genérico a partir do conjunto de treinamento. Na pré-poda alguns exemplos são ignorados durante o aprendizado de forma que o classificador não se ajuste tanto aos exemplos. Na pós-poda, o classificador inicialmente é gerado sem restrições e somente depois os ramos de menor abrangência são cortados para a generalização do classificador.

2.1.1 Árvore de Decisão

Árvore de decisão (AD) é uma estrutura de dados definida recursivamente como um nó folha que corresponde a uma Classe ou um nó de decisão que contém um teste sobre algum atributo. Para cada resultado do teste existe uma aresta para uma subárvore. Cada subárvore tem a mesma estrutura que a árvore (Monard and Baranauskas, 2003). Geometricamente, é como se o espaço dos atributos fosse dividido em regiões que não se interpõem, cada uma delas rotuladas com determinada Classe.

Pela forma de representação do conhecimento em árvore, faz parte da família TDIDT - *Top-Down Induction of Decision Trees* - de sistemas de aprendizado, onde a AD é desenvolvida de cima para baixo, guiada pela informação da frequência nos exemplos e não pela ordem em que eles são dados, dentro de um conjunto de casos relevantes para a tarefa de classificação (Quinlan, 1986).

O objetivo de um algoritmo de AD é encontrar algum relacionamento entre os atributos e a Classe associada, de modo que o processo de classificação

ou regressão possa usar esse relacionamento para predizer a Classe de um exemplo novo e desconhecido (Rezende et al., 2003).

Dada esta propriedade de aprender com exemplos rotulados, pode-se dizer então que, na hierarquia do aprendizado de máquina, o modo de aprendizado de uma AD é o supervisionado, segundo o paradigma simbólico. A forma de aprendizado é o não-incremental, utilizando todo o conjunto de exemplos para o aprendizado.

O conjunto de exemplos pode ser dividido em três subconjuntos: um de treinamento, um de teste e outro de validação. O primeiro é utilizado para o aprendizado, para a geração das hipóteses, e deve ser amostrado de forma que represente satisfatoriamente a distribuição dos dados da população.

O segundo será utilizado para avaliar a eficiência do classificador gerado, devendo ser disjunto das amostras de treinamento. O terceiro pode ser necessário caso o modelo tenha que passar por ajustes, e é constituído de exemplos que não participaram da construção do indutor nem aparecem no conjunto de testes.

Estes conjuntos de testes e de validação são meios práticos de se evitar casos de *overfitting* ou *underfitting*, que criam árvores muito especializadas ou muito superficiais. Se o objetivo é AD eficiente, então é indispensável o uso desse tipo de validação do algoritmo, avançando para amostragens que permitam avaliar o classificador perante amostras aleatórias de exemplos, e o uso de técnicas de poda da AD.

Com a poda da AD indroduz-se uma capacidade maior de generalização do indutor, havendo dois métodos, a pré e a pós-poda. A pré-poda é executada enquanto a AD é induzida, onde um nó de decisão corrente é transformado em nó folha. Economiza-se tempo pois não é necessário construir toda a árvore, e o limiar para o particionamento deve ser cuidadosamente definido para que a ramificação ocorra somente quando houver benefício de ganho de informação.

Alguns sistemas podam a AD depois de induzi-la, processo chamado de pós-poda, reduzindo o número de nós internos e, consequentemente, a complexidade da árvore, com possível melhora no desempenho da árvore original (Prati, 2006). Pode ser um processo mais custoso porém tem a vantagem de testar todas as possíveis ramificações para depois podar as menos vantajosas.

No caso de existirem problemas nos dados, os chamados ruídos, as árvores geradas são mais profundas pois se ajustam a estes ruídos. Ocorre então o *overfitting* que pode ser resolvido com uma das técnicas de poda. O algoritmo da AD deve ser capaz de lidar com os ruídos nos atributos e nas Classes pois

podem levar a hipóteses erradas. Dados imperfeitos podem ter origem na geração dos dados de entrada, na coleta dos dados, ou até mesmo na errada atribuição de rótulos.

O algoritmo deve decidir que testar mais atributos não vai melhorar a precisão da predição e deve interromper o aumento da complexidade da árvore causado pela tentativa de acomodar o caso de ruído (Quinlan, 1986).

Árvores muito complexas diminuem a vantagem das ADs sobre outras técnicas de AM de produzirem modelos facilmente interpretados por humanos. É uma característica importante, visto que especialistas humanos podem analisar um conjunto de regras aprendidas por uma AD e determinar se o modelo aprendido é plausível, dadas as restrições do mundo real (de Souto et al., 2003). Além disso, são rapidamente construídas e incorporam naturalmente misturas de variáveis contínuas, nominais e valores ausentes.

Essa mistura de variáveis quantitativas, binárias, categóricas, com geralmente muitos valores perdidos e mal coletados, faz da AD, dentre os métodos de aprendizado mais conhecidos, a que mais se aproxima dos requisitos para servir como um procedimento *off-the-shelf* para a mineração de dados (Hastie et al., 2009). Um procedimento *off-the-shelf* é aquele que pode ser diretamente aplicado aos dados sem necessitar de grande esforço com pré-processamento ou calibração.

Independente de ser liberal quanto ao tipo de variáveis, todo o sucesso de uma AD depende da escolha acertada dos atributos que melhor representam o conhecimento e que constituirão as tuplas da lista de exemplos de treinamento.

2.1.2 O Algoritmo para AD

Segundo Monard and Baranauskas (2003), os passos para a construção de uma AD são simples. Seja X um conjunto de treinamento com n linhas e Y as Classes $\{Y_1, Y_2, ..., Y_n\}$:

1. X contém 1 ou mais exemplos, todos pertencentes à mesma Classe Y_j. A árvore para X é um nó folha identificando a Classe Y_j;

2. X não contém exemplos. A árvore é uma folha com a Classe determinada, por exemplo, pela Classe mais frequente do nó pai deste nó;

3. X contém exemplos que pertencem a várias Classes. Refinar X em subconjuntos de exemplos pertencentes a uma única Classe.

4. Aplicar os passos 1, 2 e 3 recursivamente para cada subconjunto de exemplos de treinamento de forma que, em cada nó, as arestas levem para as subárvores construídas a partir do subconjunto de exemplos X_i;

5. Pós-poda para melhorar a capacidade de generalização da AD

Dentre os algoritmos de árvore de decisão disponíveis tanto para uso comercial quanto para pesquisa, pode-se citar:

- ID3 (Quinlan, 1986): pioneiro, não aceita dados contínuos, sendo necessário esforço para tratamento dos dados com a discretização dos valores e soluções para valores ausentes;
- C4.5 (Quinlan, 2014): evolução do ID3, com método de pós-poda, aceita dados contínuos e ausentes, um dos mais utilizados por apresentar ótimos resultados em problemas de classificação;
- CART - *Classification and Regression Trees* (Breiman, 1984): tem o diferencial de permitir a utilização de combinação linear entre atributos, possui pós-poda eficiente e produz árvores binárias mais simples com boa capacidade de generalização;
- NBTree - *Naive Bayes/Decision Tree Hybrid* (Kohavi, 1996): modelo híbrido que combina AD com Naive-Bayes;
- ADTree - *Alternative Decision Tree* (Freund and Mason, 1999): AD mais *boosting*;
- LMT - *Logistic Model Tree* (Landwehr et al., 2003): combina AD com regressão logística; e
- BFTree - *Best First Tree* (Shi, 2007)): AD binárias com heurísticas *best-first* para pré e pós-poda.

2.1.3 Seleção de Atributos Usando Ganho de Informação

O ponto de partida para se construir uma árvore de decisão é justamente qual atributo escolher para ser a raiz.

Um dos critérios mais utilizados para se determinar qual atributo contribui mais é o chamado *Ganho de Informação*. Para que se possa entender como ele funciona, antes é preciso entender outro conceito, o da *Entropia*.

A Entropia

Entropia é a medida de incerteza de uma variável aleatória. Expressa o número esperado de bits necessário para codificar a Classe positiva ou negativa de um exemplo que fora retirado da amostra ao acaso.

Na Equação 2.1 é mostrado o cálculo da entropia para um caso de c Classes e para p_i sendo a proporção da amostra S que pertence à Classe i.

$$Entropia(S) = \sum_{i=1}^{c} -p_i.log_2 p_i \tag{2.1}$$

A entropia para este caso varia entre 0 e 1. Se todos os membros da amostra forem iguais, pertencentes à mesma Classe, a entropia é 0. Se o número de membros classificados como positivos e negativos for igual, a entropia vale 1. Por estar na base log_2, sua unidade está em bits.

Para um problema de duas Classes, positiva e negativa, a Equação 2.1 é simplificada para a Equação 2.2.

$$Entropia(S) = -p_{\oplus}.log_2 p_{\oplus} - p_{\ominus}.log_2 p_{\ominus} \tag{2.2}$$

Exemplo 2 Como exemplo, a entropia de se lançar uma moeda honesta pode ser calculada da seguinte maneira:

$$\begin{aligned} Entropia(moeda) &= -p_{CARA}.log_2 p_{CARA} - p_{COROA}.log_2 p_{COROA} \\ Entropia(moeda) &= -0,5.log_2 0,5 - 0,5.log_2 0,5 \\ Entropia(moeda) &= 1 \end{aligned}$$

As Equações 2.1 e 2.2 calculam a entropia da amostra, para um problema multiclasses ou para um problema de duas Classes.

O cálculo da entropia de cada atributo, que pode ter vários valores, cada um com possibilidade de particionar a árvore, é feito segundo a Equação 2.3. Então seja S uma amostra, cujo atributo A possui $\{ A_1, A_2, ..., A_v \}$ como conjunto de seus v valores. A entropia do atributo A será calculada para cada um de seus valores A_i e se dará pela média ponderada da proporção de exemplos que existem para o valor A_i perante a amostra S, multiplicando-se pela entropia de se particionar pelo atributo A_i.

$$Entropia\,(A) = \sum_{i=1}^{v} \frac{A_i}{S}.Entropia(A_i) \tag{2.3}$$

O Ganho de Informação

O ganho de informação é a estimativa de redução de entropia devido ao particionamento dos exemplos por determinado atributo (Mitchell et al., 1997). É dado pela Equação 2.4 , onde S é uma amostra e A é um atributo.

$$Ganho(S, A) \equiv Entropia(S) - Entropia(A) \tag{2.4}$$

O primeiro termo da Equação 2.4 refere-se à entropia da amostra, o segundo termo refere-se à entropia depois que a amostra é particionada pelo atributo A, ou seja, é a soma das entropias de cada subconjunto S_v, dado pelo valor v do atributo A, ponderada pela fração dos exemplos de valor v que pertencem à amostra.

Com o ganho de informação calculado para cada atributo é possível escolher os melhores atributos para construção da árvore.

O algoritmo ID3 utiliza o ganho de informação para escolher o atributo para a raiz da árvore. Calcula o ganho para todos os atributos e então seleciona o que der maior ganho. A partir daí, o processo é repetido para cada novo nó descendente da raiz, cada um com sua parte da amostra (Mitchell et al., 1997). Continua até que todos os atributos já estejam contemplados na árvore ou os exemplos alocados nas folhas tenham a mesma Classe (entropia zero).

Razão de Ganho

Um problema pode ocorrer com o uso desta técnica quando existem atributos com muitos valores, tantos que quase cada exemplo possui um valor diferente para um mesmo atributo (como o CPF, por exemplo). A escolha destes atributos pode gerar muitas folhas e, consequentemente, uma árvore de profundidade mínima, ótima para classificar a amostra de treino somente, pois para outras amostras de exemplos ela não terá bom resultado.

A razão de ganho penaliza atributos com este perfil, calculando o valor do atributo pela proporção de informação gerada pela partição promovida por ele, que seja útil para a classificação. A Equação 2.5 expressa o cálculo do valor de um atributo A, com v valores. A Equação 2.6 descreve o cálculo da razão de ganho para o atributo A, pertencente à amostra S.

$$Valor\ da\ Informacão = (A) = \sum_{i=1}^{v} \frac{p_i + n_i}{p + n} . log_2 \frac{p_i + n_i}{p + n} \tag{2.5}$$

$$Razão\_de\_Ganho = \frac{Ganho\,(S, A)}{Valor\_da\_Informacão\,(A)} \tag{2.6}$$

Exemplo 3 Exemplo prático de indução de árvore de decisão: a seguir, um exemplo de construção de árvore de decisão para o conjunto de exemplos de treinamento apresentado na Tabela 2.2, baseado em (Quinlan, 1986). A estratégia do aprendizado é não-incremental, onde todos os exemplos do treinamento são submetidos ao aprendizado. A árvore de decisão será construída da raiz até suas folhas e os atributos devem ser examinados e reexaminados em muitos estágios do aprendizado, em busca de padrões nos exemplos.

Dia	Clima	Temperatura	Umidade	Vento	Jogar Tênis?
D01	Sol	Quente	Alta	Fraco	Não
D02	Sol	Quente	Alta	Forte	Não
D03	Nublado	Quente	Alta	Fraco	Sim
D04	Chuva	Ameno	Alta	Fraco	Sim
D05	Chuva	Frio	Normal	Fraco	Sim
D06	Chuva	Frio	Normal	Forte	Não
D07	Nublado	Frio	Normal	Forte	Sim
D08	Sol	Ameno	Alta	Fraco	Não
D09	Sol	Frio	Normal	Fraco	Sim
D10	Chuva	Ameno	Normal	Fraco	Sim
D11	Sol	Ameno	Normal	Forte	Sim
D12	Nublado	Ameno	Alta	Forte	Sim
D13	Nublado	Quente	Normal	Fraco	Sim
D14	Chuva	Ameno	Alta	Forte	Não

Tabela 2.2: Conjunto de treinamento para o exemplo de indução por árvore de decisão

Para este conjunto de exemplos, os atributos e seus valores possíveis são:

- Clima: Sol, Nublado, Chuva
- Temperatura: Frio, Ameno, Quente
- Umidade: Alta, Normal
- Vento: Fraco, Forte

Cada exemplo possui um rótulo, indicando se aquele perfil de dia permite que se jogue tênis (Sim ou Não).

Uma vez construída, todas as folhas da árvore de decisão contêm uma das Classes (Sim ou Não para responder à pergunta ”Jogar

tênis?"). Os demais nós intermediários representam os atributos e as ramificações correspondem aos valores possíveis destes atributos.

Para classificar um exemplo novo, o ponto de partida é a raiz da árvore, os atributos são avaliados um a um e com isso caminha-se pelos níveis da árvore até chegar em alguma folha, onde constará finalmente a Classe à qual aquele exemplo deve pertencer. Note-se que não necessariamente todos os atributos de um determinado exemplo precisam ser analisados, pode ocorrer que apenas uma parte seja suficiente para a classificação.

Seguindo o algoritmo ID3, calcula-se para cada atributo o ganho de informação. Primeiramente, a entropia de toda a amostra S usando a Equação 2.2. Há 9 exemplos positivos e 5 negativos:

$$\begin{aligned} Entropia(9+,5-) &= -(9/14).log_2(9/14) - (5/14).log_2(5/14) \\ Entropia(S) &= 0,940285959\ bits \end{aligned}$$

Agora a entropia para o atributo Clima, que tem 3 valores (Sol, Nublado e Chuva). Para o valor Sol, há 2 exemplos positivos e 3 negativos:

$$\begin{aligned} Entropia(2+,3-) &= -(2/5).log_2(2/5) - (3/5).log_2(3/5) \\ Entropia(Clima_{Sol}) &= 0,970950594\ bits \end{aligned}$$

Para o valor Nublado, há 4 exemplos positivos, a entropia é zero:

$$Entropia(Clima_{Nublado}) = 0,0\ bits$$

Para o valor Chuva, há 3 exemplos positivos e 2 negativos:

$$\begin{aligned} Entropia(3+,2-) &= -(3/5).log_2(3/5) - (2/5).log_2(2/5) \\ Entropia(Clima_{Chuva}) &= 0,970950594\ bits \end{aligned}$$

Para calcular a entropia final para o atributo Clima, Equação 2.3, basta calcular o somatório:

$$\begin{aligned} Entropia(Clima) &= (5/14).Entropia(Clima_{Sol}) + \\ &\quad (4/14).Entropia(Clima_{Nublado}) + \\ &\quad (5/14).Entropia(Clima_{Chuva}) \\ Entropia(Clima) &= (5/14)\ .\ 0,970950594 + (4/14).0,0 + (5/14)\ .\ 0,970950594 \\ Entropia(Clima) &= 0,693536139\ bits \end{aligned}$$

Entropia para o atributo Temperatura, que também tem 3 valores (Quente, Ameno e Frio). Para o valor Quente, há 2 exemplos positivos e 2 negativos:

$$\begin{aligned} Entropia(2+,2-) &= -(2/4).log_2(2/4) - (2/4).log_2(2/4) \\ Entropia(Temperatura_{Quente}) &= 1,0\ bits \end{aligned}$$

Para o valor Ameno, há 4 exemplos positivos e 2 negativos:

$$\begin{aligned} Entropia(4+,2-) &= -(4/6).log_2(4/6) - (2/6).log_2(2/6) \\ Entropia(Temperatura_{Ameno}) &= 0,918295834\ bits \end{aligned}$$

Para o valor Frio, há 3 exemplos positivos e 1 negativo:

$$\begin{aligned} Entropia(3+,1-) &= -(3/4).log_2(3/4) - (1/4).log_2(1/4) \\ Entropia(Temperatura_{Frio}) &= 0,811278124\ bits \end{aligned}$$

Entropia final para o atributo Temperatura:

$$\begin{aligned} Entropia(Temperatura) &= (4/14).Entropia(Temperatura_{Quente}) + \\ &\quad (6/14).Entropia(Temperatura_{Ameno}) + \\ &\quad (4/14).Entropia(Temperatura_{Frio}) \\ Entropia(Temperatura) &= (4/14).1,0 + (6/14).0,918295834 + (4/14).0,811278124 \\ Entropia(Temperatura) &= 0,911063393\ bits \end{aligned}$$

Entropia para o atributo Umidade, com 2 valores (Alta e Normal). Para o valor Alta, há 3 exemplos positivos e 4 negativos:

$$\begin{aligned} Entropia(3+,4-) &= -(3/7).log_2(3/7) - (4/7).log_2(4/7) \\ Entropia(Umidade_{Alta}) &= 0,985228136\ bits \end{aligned}$$

Para o valor Normal, há 6 exemplos positivos e 1 negativos:

$$\begin{aligned} Entropia(6+,1-) &= -(6/7).log_2(6/7) - (1/7).log_2(1/7) \\ Entropia(Umidade_{Normal}) &= 0,591672779\ bits \end{aligned}$$

Entropia final para o atributo Umidade:

$$\begin{aligned} Entropia(Umidade) &= (7/14).Entropia(Umidade_{Alta}) + \\ &\quad (7/14).Entropia(Umidade_{Normal}) \\ Entropia(Umidade) &= (7/14).0,985228136 + (7/14).0,591672779 \\ Entropia(Umidade) &= 0,788450457\ bits \end{aligned}$$

Entropia para o atributo Vento, com 2 valores (Fraco e Forte). Para o valor Fraco, há 6 exemplos positivos e 2 negativos:

$$\begin{aligned} Entropia(6+,2-) &= -(6/8).log_2(6/8) - (2/8).log_2(2/8) \\ Entropia(Vento_{Fraco}) &= 0,811278124\ bits \end{aligned}$$

Para o valor Forte, há 3 exemplos positivos e 3 negativos:

$$Entropia(Vento_{Forte}) = 1,0\ bits$$

Entropia final para o atributo Vento:

$$\begin{aligned} Entropia(Vento) &= (8/14).Entropia(Vento_{Fraco}) + \\ &\quad (6/14).Entropia(Vento_{Forte}) \\ Entropia(Vento) &= (8/14).0,811278124 + (6/14).1,0 \\ Entropia(Vento) &= 0,892158928\ bits \end{aligned}$$

Com as entropias calculadas, é possível calcular o ganho de informação segundo a Equação 2.4:

$$\begin{aligned} Ganho(S,Clima) &= Entropia(S) - Entropia(Clima) \\ Ganho(S,Clima) &= 0,940285959 - 0,693536139 = 0,24674982\ bits \\ Ganho(S,Temperatura) &= Entropia(S) - Entropia(Temperatura) \\ Ganho(S,Temperatura) &= 0,940285959 - 0,911063393 = 0,029222566\ bits \\ Ganho(S,Umidade) &= Entropia(S) - Entropia(Umidade) \\ Ganho(S,Umidade) &= 0,940285959 - 0,788450457 = 0,151835502\ bits \\ Ganho(S,Vento) &= Entropia(S) - Entropia(Vento) \\ Ganho(S,Vento) &= 0,940285959 - 0,892158928 = 0,048127031\ bits \end{aligned}$$

De todos os atributos, o Clima é o que proporciona o maior ganho e será a raiz da árvore. Particionado, o novo conjunto de exemplos para cada ramificação é conforme a Figura 2.2.

Como para o valor NUBLADO a entropia é zero, ele já termina em uma folha com o resultado de classificação igual a SIM. Para os valores SOL e CHUVA, ainda é necessário prosseguir para ver qual atributo vai particionar o subconjunto da amostra S que restou para cada ramo.

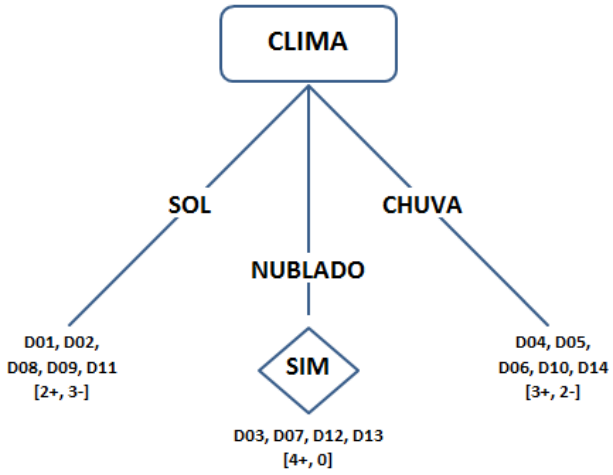


Figura 2.2: Árvore iniciada com o atributo de maior ganho de informação

Assim, para o ramo do valor SOL, temos 5 exemplos, 2 positivos e 3 negativos.

$$\begin{aligned} Entropia(2+,3-) &= -(2/5).log_2(2/5) - (3/5).log_2(3/5) \\ Entropia(S_{SOL}) &= 0,970950594\ bits \end{aligned}$$

Entropia para Temperatura. Para o valor Quente, há 2 exemplos negativos:

$$Entropia(S_{SOL}, Temperatura_{Quente}) = 0,0\ bits$$

Para o valor Ameno, há 1 positivo e 1 negativo:

$$Entropia(S_{SOL}, Temperatura_{Ameno}) = 1,0\ bits$$

Para o valor Frio, há 1 exemplo positivo:

$$Entropia(S_{SOL}, Temperatura_{Frio}) = 0,0\ bits$$

Entropia final para o atributo Temperatura:

$$Entropia(S_{SOL}, Temperatura) = 0,0 + (2/5).1,0 + 0,0 = 0,4\ bits$$

Entropia para Umidade.Para o valor Alta, há 3 exemplos negativos:

$$Entropia(S_{SOL}, Umidade_{Alta}) = 0,0\ bits$$

Para o valor Normal, há 2 exemplos positivos:

$$Entropia(S_{SOL}, Umidade_{Normal}) = 0,0\ bits$$

Entropia final para o atributo Umidade:

$$Entropia(S_{SOL}, Umidade_{Alta}) = 0,0\ bits$$

Entropia para Vento. Para o valor Fraco, há 2 exemplos positivos e 1 negativo:

$$\begin{aligned} Entropia(S_{SOL}, Vento_{Fraco}) &= -(2/3).log_2(2/3) - (1/3).log_2(1/3) \\ Entropia(S_{SOL}, Vento_{Fraco}) &= 0,918295834\ bits \end{aligned}$$

Para o valor Forte, há 1 exemplo positivo e 1 negativo:

$$Entropia(S_{SOL}, Vento_{Forte}) = 1,0\ bits$$

Entropia final para o atributo Vento:

$$\begin{aligned} Entropia(S_{SOL}, Entropia(Vento)) &= (3/5).0,918295834 + (2/5).1,0 \\ Entropia(S_{SOL}, Entropia(Vento)) &= 0,9509775\ bits \end{aligned}$$

Calculando o ganho de informação:

$$\begin{aligned} Ganho(S_{SOL}, Temperatura) &= Entropia(S_{SOL}) - Entropia(S_{SOL}, Temperatura) \\ Ganho(S_{SOL}, Temperatura) &= 0,970950594 - 0,4 = 0,570950594\ bits \\ Entropia(S_{SOL}, Umidade) &= Entropia(S_{SOL}) - Entropia(S_{SOL}, Umidade) \\ Ganho(S_{SOL}, Temperatura) &= 0,970950594 - 0,0 = 0,970950594\ bits \\ Entropia(S_{SOL}, Vento) &= Entropia(S_{SOL}) - Entropia(S_{SOL}, Vento) \\ Ganho(S_{SOL}, Vento) &= 0,970950594 - 0,9509775 = 0,019973094\ bits \end{aligned}$$

O próximo atributo a ser escolhido será a Umidade. Prosseguindo desta forma, até que se passe por todos os atributos na árvore ou até que a entropia seja zero, a árvore final é como mostra a Figura 2.3.

2.1.4 Métricas de Avaliação de Classificadores

Quando um classificador é gerado a partir de um conjunto de treinamento, a validação do aprendizado supervisionado se dá comparando os novos rótulos (preditos) do conjunto de teste com os rótulos reais. As métricas envolvendo a quantidade de acertos e de erros, dão conhecimento sobre a eficiência do classificador.

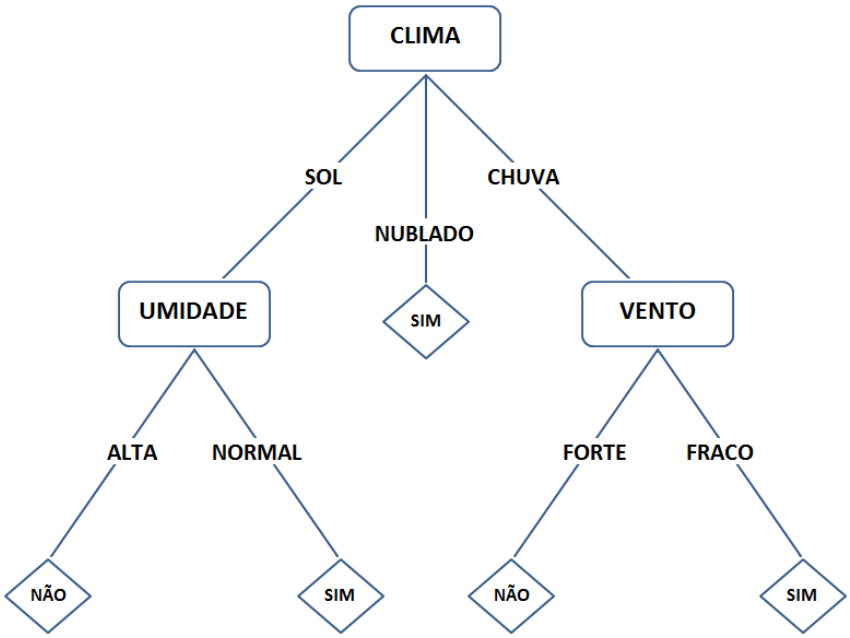


Figura 2.3: Árvore de decisão construída segundo o ganho de informação

A *Matriz de Confusão* é uma ferramenta capaz de organizar essas quantidades de erros e acertos, e uma série de equações podem ser derivadas utilizando como variáveis as células desta matriz.

Na Tabela 2.3 a Matriz de Confusão é apresentada. Na sua diagonal principal estão os exemplos classificados corretamente, como verdadeiros positivos (TP - *True Positive*) e verdadeiros negativos (TN - *True Negative*). Na diagonal secundária, os elementos classificados incorretamente, os falso negativos (FN - *False Negative*) e os falso positivos (FP - *False Positive*).

Os elementos Pos e Neg contêm a quantidade de exemplos positivos e negativos; $PPos$ e $PNeg$ contêm a quantidade de exemplos preditos como positivos e negativos; e $Total$ contém o total de exemplos analisados.

	Exemplos Positivo	Exemplos Negativo	
Preditos Positivo	TP	FN	$PPos$
Preditos Negativo	FP	TN	$PNeg$
	Pos	Neg	$Total$

Tabela 2.3: Matriz de Confusão para classificadores

As métricas que podem ser geradas para estas variáveis são:

$$\texttt{accuracy}\,(ACC) = \frac{TP + TN}{Total} \tag{2.7}$$

$$\texttt{recall}\ ou\ \textit{Sensitivity}\,(TPR) = \frac{TP}{Pos} \tag{2.8}$$

$$\textit{Miss Rate}\,(TFN) = \frac{FN}{Pos} \tag{2.9}$$

$$\textit{Fall Out}\,(FPR) = \frac{FP}{Neg} \tag{2.10}$$

$$\textit{Specificity}\,(TNR\ ou\ SPC) = \frac{TN}{Neg} \tag{2.11}$$

$$\texttt{precision}\,(PPV) = \frac{TP}{PPos} \tag{2.12}$$

$$\textit{False Discovery Rate}\,(FDR) = \frac{FP}{PPos} \tag{2.13}$$

$$\textit{False Omission Rate}\,(FOR) = \frac{FN}{PNeg} \tag{2.14}$$

$$\textit{Negative Predictive Value}\,(NPV) = \frac{TN}{PNeg} \tag{2.15}$$

$$\textit{Prevalence} = \frac{Pos}{Total} \tag{2.16}$$

$$\textit{Positive Likelihood Ratio}\,(LR+) = \frac{\texttt{recall}}{\textit{Fall-out}} \tag{2.17}$$

$$\textit{Negative Likelihood Ratio}\,(LR-) = \frac{\textit{Miss Rate}}{\textit{Specificity}} \tag{2.18}$$

$$\textit{Diagnostic Odds Ratio}\,(DOR) = \frac{LR+}{LR-} \tag{2.19}$$

$$\texttt{f1-measure} = \frac{2.TP}{2.TP + FP + FN} \tag{2.20}$$

Exemplo 4 Suponha o seguinte caso: um banco de dados de 100 pacientes submetido a um algoritmo de classificação que gera a matriz de confusão descrita na Tabela 2.4. A função é classificar os pacientes em doentes (positivo ou negativo).

Calculando as métricas para este caso, tem-se o seguinte:

	Exemplos Positivo	Exemplos Negativo	
Preditos Positivo	40	30	70
Preditos Negativo	20	10	30
	60	40	100

Tabela 2.4: Exemplo de Matriz de Confusão

- TP = "Acertou!" = 40, TN = "Acertou que não é!" = 10, FP = "Alarme falso!" = 30, FN = "Falhou!" = 20
- A proporção dos casos que foram classificados corretamente, tanto como positivos quanto negativos, sobre o total da amostra, é dada pela Taxa de Acerto ou `accuracy` = 0,50. Não é uma boa medida de eficiência do classificador pois pode apresentar resultados equivocados, principalmente se as Classes forem desbalanceadas;
- O `recall` ou *Sensitivity* ou *True Positive Rate* é a capacidade do classificador encontrar todos os casos positivos. Dentro dos 60 doentes, quantos o classificador conseguiu acertar como doente? Para o exemplo, `recall` = 0,67;
- O *Miss Rate* ou *False Negative Rate* é como fazer $1 - \texttt{recall}$. Para o exemplo, *Miss Rate* = 0,33. É diferente da Taxa de Erro do classificador ($1 - \texttt{accuracy}$) que vale 0,50 para o exemplo;
- O *Fall-out* ou *False Positive Rate* ou Taxa de Alarmes Falsos, vale 0,75 para o exemplo. Significa a porcentagem de alarmes falsos do classificador em comparação com a Classe negativa. Dos realmente negativos, qual a porcentagem de alarmes falsos;
- A *Specificity* ou *True Negative Rate* é a proporção de negativos corretamente classificados como negativos.Para o exemplo, é a probabilidade de prever que um paciente diagnosticado como não doente (sadio) é realmente sadio. *Specificity* = 0,25;
- A `precision` ou *Positive Predictive Value* é a proporção de acertos positivos por preditos positivos. Dentro do predito positivo, quantos são realmente positivos? Para o exemplo, a probabilidade de ser doente dado que o paciente foi classificado como doente. `precision` = 0,57;

- O *False Omission Rate* para o exemplo vale 0,67 e deve ser interpretado como sendo a porcentagem de falha do classificador, onde falha seja considerada como sendo prever que um paciente não é doente e na verdade ele é doente. É pior do que um alarme falso? Depende do contexto, pois ambos são falhas. Para este caso, esta medida deve informar: dentro dos preditos negativos, quantos são falhas?
- O *False Discovery Value* para o exemplo vale 0,43. Dentro do que foi predito como positivo, quantos são alarmes falsos?
- O *Negative Predictive Value* para o exemplo vale 0,33. Significa a proporção de testes preditos negativos que realmente são negativos, ou seja, dentro dos preditos negativos, quantos são realmente negativos. É uma taxa de acerto para o inverso do valor da Classe;
- Para o exemplo, $LR+$ = 0,89, $LR-$ = 1,33 e DOR = 0,67. O DOR é a razão de duas probabilidades, uma de ser positivo se o sujeito tem a doença, e outra de ser positivo se o sujeito não tem a doença. Quanto maior o seu valor, melhor o desempenho do classificador;
- A *Prevalence*, para o exemplo, vale 0,60 e indica a predominância da Classe positiva dentro da população total; e
- O `f1-measure` é a média ponderada do `precision` e do `recall` para verificar a correção do classificador. Para o exemplo vale 0,62.

Fora estas métricas para o classificador binário, neste trabalho foi utilizada mais uma, a *Mean Absolute Error - MAE* ou Erro Médio Absoluto. É uma métrica mais relacionada com problemas de regressão porém serviu para indicar uma porcentagem de erro relacionada com a diferença entre o previsto e o real.

Segundo a Equação 2.21, o MAE é calculado segundo a média da soma de todas as diferenças do valor predito, f_i, com o valor real, y_i.

$$MAE = \frac{1}{n}.\sum_{i=1}^{n}|f_i - y_i| \tag{2.21}$$

É uma medida comum em estatística e quanto maior seu valor, mais distante está o predito do real, ou seja, pior é o método de predição.

2.1.5 Análise ROC

Calculando-se uma Matriz de Confusão para um classificador, TPR e FPR passam a ser vistos como um par ordenado num espaço bidimensional, chamado espaço ROC - *Receiver Operating Characteristic* onde são plotados os pontos (FPR, TPR) para comparação dos classificadores.

O gráfico ROC é uma técnica para visualizar, organizar e selecionar classificadores baseados na sua performance (Fawcett, 2006). Nele é possível analisar o comportamento dos classificadores, quais privilegiam algumas das Classes, quais se mostraram potencialmente ótimos e quais podem ser descartados da análise (Matsubara, 2008).

É indicado para casos em que há desbalanceamento de Classes e custo de classificação diferente para uma das Classes. Na Figura 2.4 é mostrado o formato de um gráfico ROC.

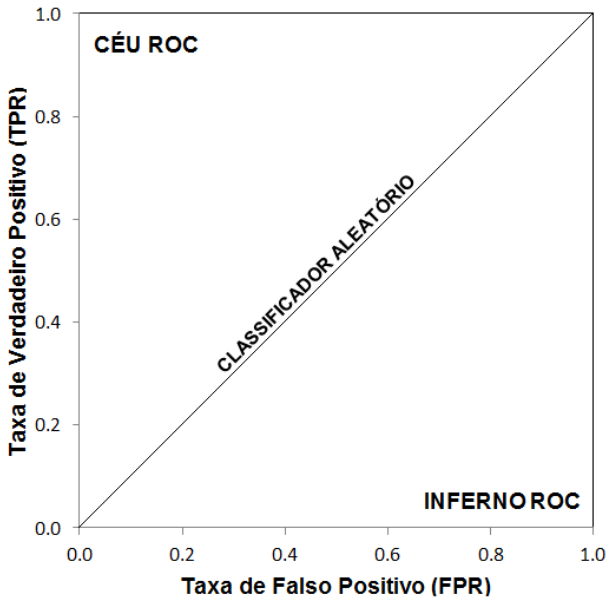


Figura 2.4: Modelo de um Gráfico ROC

Observando a Figura 2.4 é possível fazer algumas considerações:

- No ponto (0,0) o classificador não erra mas também não classifica nada como positivo. Já o ponto (1,1) simplesmente classifica os exemplos como positivos, acerta muito mas erra igualmente;
- O ponto (0,1) significa a classificação perfeita e os pontos desta região, chamada Céu ROC, correspondem a classificadores com bom desempenho, onde os acertos tendem ao máximo e os alarmes falsos aproximam-se de zero;

- No ponto (1,0), região chamada Inferno ROC, estão os classificadores com desempenho ruim, onde tendem a não acertar nada e os alarmes falsos estão perto do máximo possível. Ao inverter os rótulos, porém, obtém-se um classificador no Céu ROC;
- Nos pontos da diagonal do gráfico estão os chamados classificadores aleatórios, que, por não possuírem informação sobre a Classe, classificam aleatoriamente, deslocando-se na diagonal, dependendo da frequência com que "chutam" a Classe positiva;
- Ao comparar os pontos no espaço ROC, os que estão mais ao noroeste do gráfico são os melhores;
- Quanto mais próximo do eixo X, onde há poucos erros de alarmes falsos, o classificador é chamado conservador. Quanto mais acima e à direita no gráfico, o classificador é chamado liberal, acerta muito mas têm muitos alarmes falsos. O primeiro precisa de mais indícios do que o segundo para realizar a classificação correta.

O melhor classificador ao analisar um gráfico ROC depende do domínio de aplicação (Matsubara, 2008). Um gráfico ROC retrata as compensações entre os benefícios (TPR) e os custos (FPR) (Fawcett, 2006). É preciso analisar então custo x benefício de cada caso. O erro ao classificar um paciente sadio como positivo para cirurgia é grave? O erro ao classificar um paciente doente como negativo para doente é grave? Dependerá do contexto do problema.

Os classificadores podem além da Classe, fornecer um valor de confiança da classificação, denominado de *escore de classificação*. Em árvores de decisão o escore pode ser obtido pela proporção de exemplos positivos no nó folha utilizado para classificar o exemplo. Por exemplo, um nó folha com 4 exemplos positivos e 6 exemplos negativos, a classificação de um exemplo por este nó folha será negativa.

O critério normalmente adotado é da classificação da Classe de maior quantidade de exemplos. Ao utilizar a proporção de exemplos positivos, no exemplo é de 40%, o critério é equivalente a utilizar a regra: se a proporção de exemplos positivos for maior que 50% a Classe é positiva caso contrário é negativa.

Em análise ROC a proporção de exemplos positivos é conhecida como *escore de classificação*. O valor que divide o *escore de classificação* é denominado de limiar. Ao utilizar estes termos a classificação binária pode ser dada pela fórmula:

$$t(escore, limiar) = \begin{cases} pos & Se\ escore > limiar \\ neg & c.c. \end{cases}$$

Cada possível limiar produz uma matriz de confusão. Cada matriz produz TVP e TFP diferentes. Ao testar todos os possíveis limiares, obtém-se diferentes pontos no gráfico ROC. Ao unir todos os pontos produzidos por diferentes limiares produz-se a curva ROC.

Exemplo 5 Na Tabela 2.5 um exemplo, baseado em (Matsubara, 2008), demonstrando como calcular os pontos (FPR, TPR) para a Curva ROC de classificador, para 10 exemplos, cada um com seu `escore` e Classe. Para cada limiar um par ordenado foi calculado. Com isso, basta plotar a curva.

Exemplo	A	B	V	C	D	W	X	Y	E	Z		
Classe	+	+	-	+	+	-	-	-	+	-		
Escore	0,9	0,8	0,4	0,39	0,38	0,37	0,36	0,2	0,2	0,05	**TPR**	**FPR**
limiar = 0,9	+	-	-	-	-	-	-	-	-	-	0,2	0,0
limiar = 0,8	+	+	-	-	-	-	-	-	-	-	0,4	0,0
limiar = 0,5	+	+	-	-	-	-	-	-	-	-	0,4	0,0
limiar = 0,4	+	+	+	-	-	-	-	-	-	-	0,4	0,2
limiar = 0,39	+	+	+	+	-	-	-	-	-	-	0,6	0,2
limiar = 0,38	+	+	+	+	+	-	-	-	-	-	0,8	0,2
limiar = 0,37	+	+	+	+	+	+	-	-	-	-	0,8	0,4
limiar = 0,36	+	+	+	+	+	+	+	-	-	-	0,8	0,6
limiar = 0,2	+	+	+	+	+	+	+	+	+	-	1,0	0,8
limiar = 0,05	+	+	+	+	+	+	+	+	+	+	1,0	1,0

Tabela 2.5: Exemplo de classificação por limiar e dos pontos (FPR, TPR) para a Curva ROC

Uma maneira eficiente de se "desenhar" a curva ROC é considerar a monotonicidade das classificações por limiar (Fawcett, 2006), isto é, uma vez que o exemplo é classificado como positivo para um dado limiar, ele continuará positivo para os limiares inferiores também.

Assim, ao ordenar decrescentemente os exemplos pelo `escore`, percorre-se a lista de de exemplos, um a um, e partindo do ponto (0,0), verifica-se:

- Se a Classe for positiva, subir $\frac{1}{Pos}$, para Pos = Total de exemplos positivos;
- Se a Classe for negativa, seguir à direita $\frac{1}{Neg}$, para Neg = total de exemplos negativos;

- Se houver empate no `escore`, é preciso fazer uma média dos pontos que seriam plotados, de acordo com a quantidade de positivos e negativos. Então supondo que haja k empates, $npos$ = número de positivos dentro de k, e $nneg$ = número de negativos dentro de k. Uma diagonal será desenhada entre o ponto atual $(x,\ y)$ e o ponto $(x + \frac{nneg}{Neg}, y + \frac{npos}{Pos})$.

Seguindo este método, a curva ROC correspondente à Tabela 2.5 é conforme a Figura 2.5, cujos pontos no gráfico condizem com os pares (FPR, TPR) da tabela.

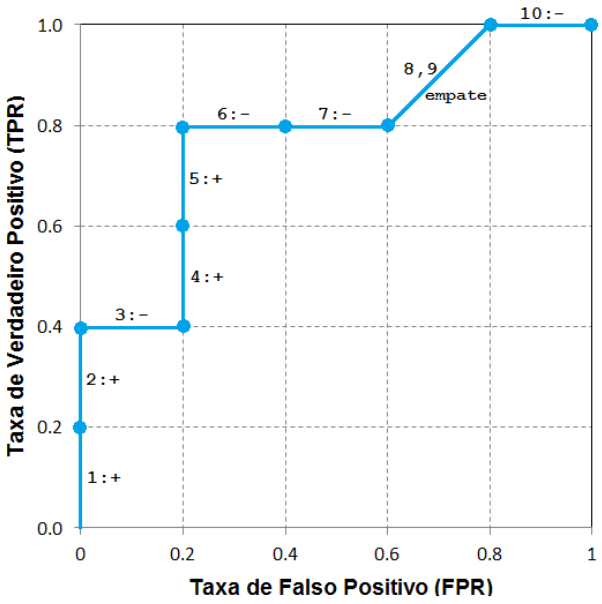


Figura 2.5: Curva ROC para o exemplo

Neste trabalho, a comparação entre os classificadores será feita via cálculo da Área Abaixo da Curva ou *Area Under ROC Curve - AUC*.

A AUC converte a curva ROC em um valor numérico calculando-se a área abaixo da curva ROC. Imaginando que o espaço ROC tem uma área total igual a 1, quanto mais próximo do ponto ótimo a curva ROC está, mais próxima de 1 estará a AUC. Da mesma forma, uma curva ROC de um classificador aleatório terá AUC = 0,5, logo, nenhum classificador razoável deverá ter AUC menor do que 0,5.

Na Figura 2.6, em cinza, a AUC calculada resulta em 0,78. Isto equivale a dizer que, ao escolher aleatoriamente dois exemplos, um positivo e outro negativo, da Tabela 2.5, há 78% de probabilidade do exemplo positivo aparecer antes do negativo.

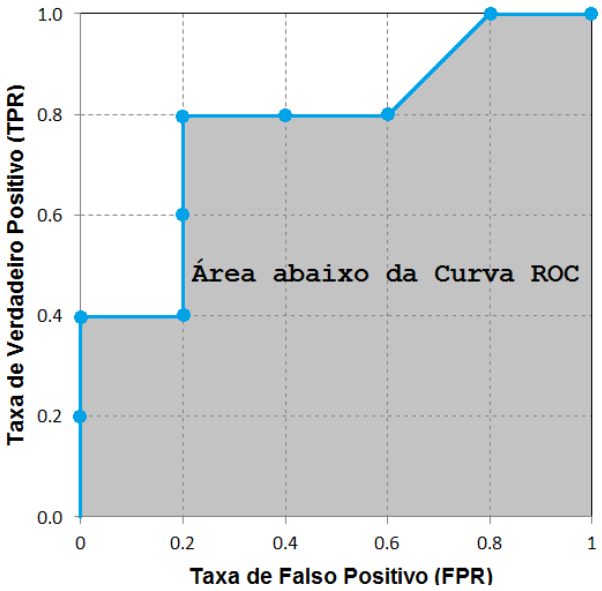


Figura 2.6: Área abaixo da curva ROC - AUC para o exemplo

2.2 Conceitos de Melhoramento Genético Animal

O melhoramento genético animal é uma das áreas de pesquisa que muito tem contribuído com tecnologias para o incremento da produção de carne bovina. É uma atividade permanente que envolve os processos de criação (práticas de alimentação, manejo e sanidade), seleção e planos de acasalamento, cujo objetivo básico é alterar continuamente as características dos animais produzidos nas gerações seguintes, em sintonia com o ambiente e o mercado (Martín Nieto and Rosa, 2013).

Propõe-se a assegurar que a aparência não mascare a produtividade (Daly, 1977), ou seja, que a seleção dos melhores animais seja feita não pela expressão visível do animal, mas pela sua capacidade em transmitir qualidade genética a seus descendentes.

Se somente a aparência for levada em consideração, um animal bem tratado pode parecer até melhor que um animal geneticamente superior. Mas como diferenciá-los? É nessa dimensão que as avaliações genéticas vêm contribuir.

Tamanha importância do melhoramento genético de bovinos de corte no Brasil pode ser constatada pela quantidade de programas de avaliação genética existentes, e a popularização do uso das estimativas de diferenças esperadas na progênie (DEPs) pela comunidade da cadeia da carne. O melhoramento genético por sua vez, não constitui apenas uma tecnologia destinada a produzir DEPs, mas de fornecer subsídos para que os produtores possam aplicar

esse conjunto de informações genéticas (Abreu et al., 2013).

A primeira decisão a ser tomada pelo criador antes de iniciar um programa de melhoramento é definir quais serão seus objetivos de seleção a médio e longo prazo. Esta definição depende do seu sistema de produção (análise da situação atual do rebanho, ambiente, nutrição, reprodução, infraestrutura da fazenda) e do mercado (identificação de clientes) (Martín Nieto and Rosa, 2013).

Definidos os objetivos de seleção, o passo seguinte é eleger os critérios de seleção que são as características que serão acompanhadas durante a vida do animal. Pode ser composto por uma única característica ou por uma combinação ponderada delas. Tais ponderações devem ser estabelecidas com base nos valores econômicos das características, de modo a representar a contribuição de cada uma delas para o retorno econômico da seleção (Martín Nieto and Rosa, 2013).

Para que se possa compreender a diferença entre objetivos de seleção, que envolvem aspectos econômicos, e critérios de seleção, que são de natureza biológica, alguns exemplos para cada um dos dois itens, segundo (Martín Nieto and Rosa, 2013):

Exemplo 6 Objetivos de seleção:

- Produzir bezerros para sistemas extensivos de produção em ambiente tropical;
- Produzir bezerros para terminação em confinamento;
- Aumentar precocidade sexual e eficiência reprodutiva;
- Melhorar a maciez da carne.

Exemplo 7 Critérios de seleção:

- Reprodução: idade ao primeiro parto, dias para parir, período de gestação, idade à puberdade, perímetro escrotal;
- Produção: pesos, ganhos de peso, altura, eficiência alimentar;
- Qualidade do produto: conformação frigorífica, área de olho de lombo, espessura de gorudra subcutânea, maciez da carne;
- Biótipo: aprumos, comprimento de pelos, cor da pelagem.

Um item fundamental para o melhoramento genético é o banco de dados com as informações coletadas dos animais. As propriedades que desejam

participar de um programa de melhoramento devem armazenar os dados de seus rebanhos a fim de poder obter um *feedback* de como caminha econômica, genética, e sanitariamente sua propriedade.

Deste modo, os produtores alimentam bancos de dados particulares, de associações de criadores, de programas de melhoramento genético, entre outros e cedem esses bancos aos programas de melhoramento dos quais participam, para que formem um banco de dados geral, com várias fazendas, que servirá de entrada para os *softwares* de avaliação genética.

As associações de criadores possuem uma riqueza de dados históricos, coletados há muitos anos, como a Associação Brasileira dos Criadores de Zebu - ABCZ [1], que registra em todo o Brasil mais de 600 mil zebuínos por ano e detém o maior banco de dados do mundo sobre o zebu, com mais de 12 milhões de animais cadastrados. Assim como criadores, algumas associações possuem seus respectivos programas de melhoramento, para quem mandam integralmente os dados de seus associados para processamento das avaliações.

Na Figura 2.7, alguns exemplos de raças de importância econômica para o Brasil, segundo (Rosa et al., 2013).

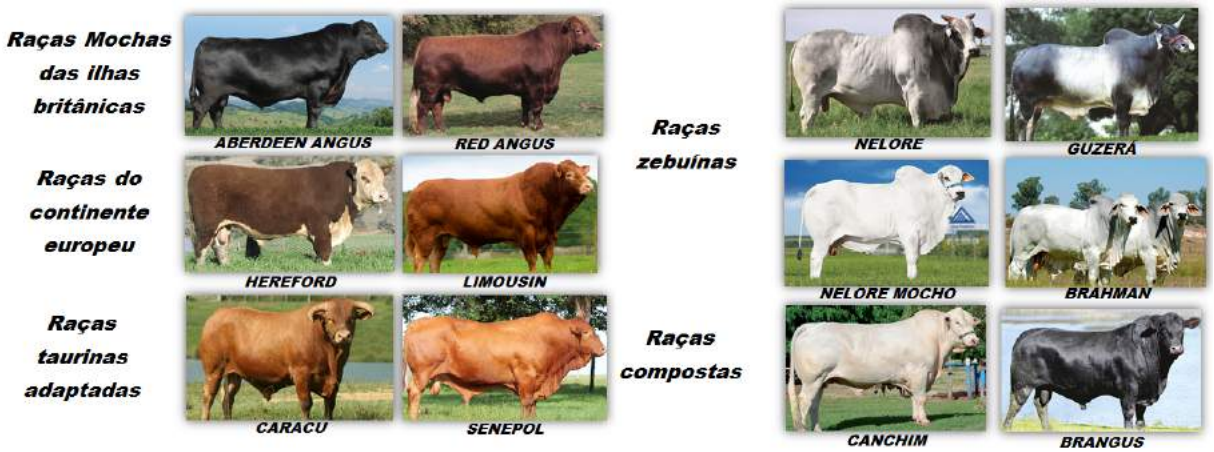


Figura 2.7: Raças de importância econômica para o Brasil

2.2.1 Um Pouco de História

Segundo (Pereira, 2012), o melhoramento animal teve origem nos trabalhos de um fazendeiro inglês, Robert Bakewell (1725-1795), responsável pela formação e evolução de raças dentro das espécies bovina, ovina, e equina. Seus trabalhos desencadearam a formação das sociedades de raças e a criação

[1] www.abcz.org.br

dos registros genealógicos. Bakewell era grande obsevador, pois ainda não existiam conhecimentos acerca da herança dos animais.

Como ciência, pode-se dizer que o melhoramento animal surgiu com a descoberta das leis da herança pelo monge austríaco Gregor Mendel. Seus resultados permenceram na obscuridade até 1900, quando os pesquisadors De Vries, Correns e Von Tschermak, independentemente, redescobriam-nas (Pereira, 2012).

A estatística ligada à herança deveu-se a o francês Galton (1822-1911) considerado o pai da biometria. Sir Ronald Fisher e Sewall Wright foram responsáveis pela moderna genética de populações. Henderson, pesquisador americano da área de genética animal, deu relevantes contribuições no desenvolvimento de metodologias analíticas como a dos modelos mistos, hoje amplamente utilizada nas avaliações genéticas (Pereira, 2012).

2.2.2 Noções Elementares Importantes

O desenvolvimento de um animal é influenciado pelo ambiente e pelo conjunto de informações genéticas (genoma) que ele possui (Daly, 1977). Mesmo que aos animais sejam oferecidos as mesmas condições ambientais, haverá diferenças entre eles devido a fontes de variação associadas ao fenótipo e ao genótipo de cada um.

Fenótipo: o fenótipo é a aparência, é o que é mensurável nas características constituintes de um animal.

Genótipo: o genótipo é a soma total dos genes do animal, seu potencial genético, cuja amostra será aleatoriamente transferida aos seus descendentes. A produção de cada indivíduo é resultado da ação de seus genes e das forças que agem sobre ele, ou seja: Fenótipo = genótipo + ambiente. É importante pois, determinar a fração do fenótipo que é devida aos efeitos dos genes e a fração que é devida aos efeitos de ambiente, pois apenas os efeitos dos genes são transmitidos à próxima geração (Eler, 2014).

Efeito aditivo: ao se considerar um gene, denomina-se efeito aditivo o efeito deste gene que resulta em uma mudança fenotípica definida, ou seja, é o que é passado para os filhos no melhoramento.

Valor genético: a soma dos efeitos aditivos de todo o genoma constitui o valor genético do indivíduo (Rosa et al., 2013). O valor genético mede a

qualidade genética de um animal quanto à produção de seus filhos (Torres Junior et al., 2013). É a realização dos seus atributos.

***Pedigree*:** ou árvore genealógica, descreve toda a genealogia de um animal, ou seja, todos os seus ancestrais.

Herdabilidade (h^2): é a herança na determinação das diferenças entre animais em uma característica determinada (Daly, 1977). É uma medida do quanto as diferenças individuais de produção, para dada característica, refletirão, em média, na superioridade da progênie (Torres Junior et al., 2013).

> **Exemplo 8** Por exemplo, suponha um lote de animais com média de peso ao sobreano de 300kg. Deste lote é selecionado um pequeno grupo de touros e novilhas com uma média de peso ao sobreano de 350kg. A amplitude da seleção para peso ao sobreano foi de 50kg (350-300). Suponha que h^2 = 0,5. O ganho genético que será obtido dessa seleção será de 25kg (0,5 * 50kg), significando que as progênies dos animais selecionados serão 25kg mais pesadas ao sobreano do que se nenhuma seleção tivesse sido feita (Daly, 1977).

A partir de modelos genético-estatísticos que combinam dados fenotípicos e de *pedigree* são feitas as avaliações genéticas e gerados os valores genéticos que amparam um programa de melhoramento na seleção dos animais superiores para as características de importância econômica.

Quanto maior o volume dos dados da população, em novos animais acrescentados ou em número de atributos preenchidos, mais precisos são os resultados de uma avaliação genética. O *pedigree*, a performance individual, a performance de irmãos e parentes e a performance da progênie contribuem para compor o *input* de um programa de avaliação genética.

A quantidade de atributos exprime a complexidade do trabalho em se tentar encontrar a melhor combinação das características que vão auxiliar o criador a atingir seus objetivos de seleção. No passado, a coleta era basicamente de pesagens e perímetros escrotais. Os escores visuais vieram depois como complemento na busca por animais morfologicamente eficientes e equilibrados com o sistema de produção. Mais recentemente, dados de ultrassom passaram a fazer parte do banco de dados e das avaliações genéticas, mesmo sendo ainda pequeno em volume com relação ao número de observações válidas para peso.

2.2.3 Grupo Contemporâneo

Cada banco de dados das fazendas e associações fará parte de uma base geral criada para conter todos os bancos juntos, formando a população que será analisada pelos *softwares* de avaliação genética.

Para que a comparação entre os animais possa ser feita justamente, é fundamental que para cada característica coletada sejam informadas corretamente as circunstâncias em que isso ocorreu. Um animal que teve pasto à vontade será mais pesado que um animal que passou por uma seca. Um animal nascido em São Paulo não pode ser comparado com outro nascido no Pará.

Para cada característica coletada é preciso informar em qual fazenda nasceu, qual o ano de nascimento do animal, a época (trimestre) de nascimento do animal, o sexo, a data da coleta da característica e o regime alimentar na coleta (pasto, semi-confinamento, suplementação). A este conjunto de dados que informa as condições ambientais em que a característica foi coletada dá-se o nome de Grupo Contemporâneo.

Segundo (Cobuci et al., 2006), os grupos contemporâneos devem permitir um número razoável de animais em um mesmo grupo, que sejam conectados geneticamente entre si e que, ao mesmo tempo, permitam o real agrupamento de animais que tiveram seus desempenhos influenciados pelas mesmas condições ambientais (clima, alimentação, sanidade, manejo, etc.). A maneira ideal para comparar animais, levando em conta tanto os fatores de ambiente quanto as interações entre eles, é a formação de grupos contemporâneos.

2.2.4 A Avaliação Genética

Segundo (Pereira, 2012), as etapas para a realização de uma avaliação genética compreendem:

1. Geração das informações nas fazendas;
2. Dados são enviados a um centro de avaliação;
3. Nas análises dos dados, formam-se os grupos contemporâneos;
4. Formam-se os arquivos de dados para análises;
5. Calcula-se o grau de parentesco genético entre todos os indivíduos;
6. Formam-se as equações para solução simultânea. Os modelos mais comuns são:

a Modelo Touro - cada touro tem uma equação;

b Modelo Animal - cada indivíduo da população tem sua própria equação;

c Modelo Animal Reduzido - existe uma equação para cada animal que tem progênie.

7. Os resultados da avaliação genética são produzidos.

A Tabela 2.6 mostra um exemplo de arquivo no formato pronto para avaliação genética. A parte de *pedigree* toda codificada (animal, pai, mãe, sexo), os grupos contemporâneos aos 120 dias e à desmama (GCM e GCD), as idades nas pesagens aos 120 dias e à desmama (IDM e IDD), os pesos ajustados para as idades-padrão de 120 dias e desmama (PM e PD), e os anos de nascimento do animal, pai e mãe (Ano, AnoP e AnoM).

Animal	Pai	Mãe	GCM	GCD	IDM	IDD	PM	PD	Ano	Sx	AnoP	AnoM
6082115	5905776	5959819	2964	2959	131	216	132.59	220.17	93	1	88	83
6082119	5904175	5960077	2964	2959	139	224	143.53	230	93	1	84	82
6082175	5909183	5961442	2964	2959	160	245	119.06	186.65	93	1	88	87
6082190	5909183	5961890	2964	2959	161	246	124.14	194.03	93	1	88	88
6082197	5962025	5961902	2964	2959	161	246	130.86	215.04	93	1	90	85
6082205	5904175	5961427	2964	2959	161	246	110.23	177.82	93	1	84	89
6082230	5904175	5961416	0	2959	888	259	8888	202.81	93	1	84	89
6082262	5909183	5961867	0	2959	888	226	8888	227.16	93	1	88	86
6082237	5904615	5961145	0	0	888	888	8888	8888	94	1	90	86
6082246	5962025	5960901	0	0	888	888	8888	8888	93	1	90	78
6082248	5905196	5965066	0	7098	888	267	8888	229.44	89	1	82	83

Tabela 2.6: Formato do arquivo para avaliação genética

Os *softwares* para as análises genéticas, utilizam a *Metodologia de Modelos Mistos* (Henderson, 1953), sendo adotado o Modelo Animal, considerando características múltiplas.

As estimativas dos componentes de variância utilizados nas análises, foram calculadas pela combinação dos resultados da amostra da população analisada e informações constantes na literatura. O *software* que possibilitou a obtenção das predições das DEPs foi gentilmente desenvolvido e disponibilizado pelo professor *Dr. Lawrence R. Schaeffer*, da Universidade de *Guelph*-Canadá (Nobre, 2014).

De modo simplificado, as características observadas servem de entrada para um modelo estatístico como o da Equação 2.22 abaixo:

$$y_{ij} = f_i + g_{ij} + E_{ij} \tag{2.22}$$

Um determinado fenótipo "y" do animal "j" no ambiente "i" depende: do conjunto de efeitos identificáveis "f" do ambiente "i", como por exemplo, o

grupo contemporâneo; do conjunto de efeitos genéticos "g" do animal "j" criado no ambiente "i"; e do conjunto de efeitos do ambiente não identificáveis "E" do animal "j" criado no ambiente "i" (Martins, 2013).

Exemplo 9 Pequeno exemplo dos passos para a geração da DEP

Supondo a característica que mede o ganho de peso da desmama ao sobreano, um modelo ilustrativo poderia ser segundo a Equação 2.23, baseado em (Martins, 2013).

$$GPD_{ij} = gc_i + g_{ij} + \epsilon_{ij} \tag{2.23}$$

A expressão mensurável, o fenótipo (GPD), é a soma dos efeitos de ambiente no grupo contemporâneo (gc); do efeito genético (g); e do resíduo (ϵ).

A Tabela 2.7 mostra um exemplo de dados coletados.

animal	sx	ano	rebanho	data da pesagem	regime	ganho
17	1	84	2	25/02/1986	1	72,40
21	1	84	2	25/02/1986	1	95,65
25	1	84	2	25/02/1986	1	79,90
26	1	84	3	03/03/1986	1	101,65
22	1	84	3	03/03/1986	1	90,40
19	1	84	3	03/03/1986	1	74,65
27	1	84	3	03/03/1986	1	99,40

Tabela 2.7: Exemplo de dados coletados

O grupo contemporâneo (gc) é formado pelas informações de sexo (sx), ano, rebanho, data da pesagem ao sobreano (data da pesagem) e regime, conforme mostra a Tabela 2.8.

animal	sx	ano	rebanho	data da pesagem	regime	gc	ganho
17	1	84	2	25/02/1986	1	**1**	72,40
21	1	84	2	25/02/1986	1	**1**	95,65
25	1	84	2	25/02/1986	1	**1**	79,90
26	1	84	3	03/03/1986	1	**2**	101,65
22	1	84	3	03/03/1986	1	**2**	90,40
19	1	84	3	03/03/1986	1	**2**	74,65
27	1	84	3	03/03/1986	1	**2**	99.4

Tabela 2.8: Criando grupos contemporâneos

O processo de estimação dos efeitos de ambiente nos grupos contemporâneos dá-se por meio do Método de Quadrados Mínimos. Para o exemplo, o efeito de ambiente do grupo contemporâneo será a média do ganho, conforme a Tabela 2.9.

gc	Efeito (média)
1	82.65
2	91.525

Tabela 2.9: Efeito do grupo contemporâneo

Para eliminar o efeito do grupo contemporâneo, ditos ser identificáveis do ambiente, faz-se um ajustamento no ganho, subtraindo-se $ganho - Efeito$, como mostra a Tabela 2.10.

animal	gc	ganho	ajustamento	Ganho Ajustado
17	1	72,40	72,40 - 82,65	-10,25
21	1	95,65	95,65 - 82,65	13,00
25	1	79,90	79,90 - 82,65	-2,75
26	2	101,65	101,65 - 91,52	10,13
22	2	90,40	90,40 - 91,52	-1,12
19	2	74,65	74,65 - 91,52	-16,87
27	2	99,40	99,40 - 91.52	7.88

Tabela 2.10: Ajustamento do GPD

Na Tabela 2.11 o resultado final da obtenção das DEPs. Os valores genéticos (VG) são obtidos pela multiplicação da herdabilidade pelo ganho ajustado. Supondo uma herdabilidade $h^2 = 0,23$, então $VG = 0,23 * Ganho\ Ajustado$. A DEP equivale à metade do VG visto que o animal recebe a metade dos genes de seus pais, portanto, $DEP = (VG/2)$.

animal	gc	ganho	Ganho Ajustado	VG	**DEP**
17	1	72,40	-10,25	-2,36	**-1,18**
21	1	95,65	13,00	2,99	**1,50**
25	1	79,90	-2,75	-0,63	**-0,32**
26	2	101,65	10,13	2,33	**1,16**
22	2	90,40	-1,13	-0,26	**-0,13**
19	2	74,65	-16,88	-3,88	**-1,94**
27	2	99,40	7,88	1,81	**0,91**

Tabela 2.11: DEPs obtidas

Este exemplo é bastante simples, somente para ilustrar como as DEPs são geradas. Tão simples que faltou considerar os parentescos no modelo. As relações de parentesco são importantes pois as observações dos parentes também serão utilizadas para predição do valor genético. O modelo é bem mais complexo.

O valor do coeficiente de herdabilidade, $h^2 = 0,23$, é um valor ilustrativo.

A avaliação genética é publicada aos criadores na forma de DEP, um valor numérico, uma estimativa que indica a diferença esperada na média das performances das progênies futuras de determinado touro em relação à média das diferenças esperadas das progênies futuras de todos os touros que participaram da mesma avaliação (para o caso de base genética móvel), quando acasalados com conjunto de vacas que tenham, entre si, o mesmo potencial genético (Nobre et al., 2013a).

> **Exemplo 10** Como exemplo, considere-se dois animais, A e B, com DEPs iguais a +15kg e -10kg, respectivamente para peso à desmama. Nesse caso, a diferença entre as DEPs desses dois animais é 25. Isso significa que considerando um grande número de filhos oriundos do acasalamento de cada um desses indivíduos com fêmeas de méritos genéticos comparáveis, a média daqueles provenientes de A deve superar em 25 unidades aquela resultante da progênie de B (Euclides Filho, 2000a) .

A DEP de um animal não é um valor estático, pode mudar em função da variação do número de informações tomadas em qualquer de seus parentes. Por isso, cada vez que uma nova avaliação genética é realizada, ela substitui a anterior. Da mesma forma, avaliações antigas ou realizadas por diferentes programas de melhoramento não podem ser comparadas entre si, visto que houve diferenças na população analisada envolvendo genealogia, metodologia de avaliação, formação dos grupos contemporâneos, características de interesse, índices, entre outros fatores.

A DEP sempre vem acompanhada da Acurácia, um valor entre 1% e 100% que indica o grau de confiabilidade da DEP. Uma acurácia acima de 70% indica baixo risco de variação da DEP em caso de alteração no número de informações relativas a determinado animal. Na prática, a DEP é o elemento de decisão de se usar ou não determinado touro, sendo a acurácia o elemento de definição da intensidade do seu uso (Nobre, 2014).

A DEP efeito materno é a diferença esperada da média das performances das progênies futuras das filhas de determinado touro, em relação à média das performances das progênies futuras das filhas de todos os outros touros que participaram da avaliação. As diferenças genéticas que existem entre as fêmeas, quanto a proporcionarem melhor ou pior meio para o desenvolvimento de suas crias, são que constituem o efeito materno (Nobre, 2014).

O Total Maternal, resultado da soma (DEP/2) + DEP efeito materno, mede a contribuição do touro para a produção de seus netos por intermédio de suas filhas (Torres Junior et al., 2013). Quando o efeito direto (capacidade genética de desenvolvimento) transmitido do pai aos netos (via filha) compensar o efeito materno negativo, significa que o touro consegue produzir boas mães.

2.2.5 Os Sumários

Já é tradição entre os produtores chamar catálogos de animais com suas avaliações genéticas de sumários. Sumários impressos ou digitais, todos têm o objetivo de informar o usuário sobre resultados da avaliação genética (DEPs e Acurácias), *rankings*, tendências genéticas, índices, médias, enfim, é uma ferramenta que auxilia o produtor na tomada de decisão.

Os sumários se apresentam como uma ferramenta que dá suporte ao criador nos procedimentos de seleção, descarte e acasalamentos necessários para que se possa alcançar o progresso genético pretendido (Silva et al., 2013).

A seleção implica na escolha dos pais da geração seguinte e na determinação da intensidade de uso dos mesnmos na reprodução. O objetivo é o aumento da frequência de alelos favoráveis à eficiência econômica do sistema de produção ou, a mudança da constituição genética da população (Martín Nieto and Rosa, 2013).

Planos de acasalamento podem ser feitos via sumários de fazendas, onde matrizes e touros são escolhidos de acordo com o critério do produtor ou técnico do programa, analisando DEPs e índices. Com isto é construído um relatório de campo que já segue com as opções de touros e de matrizes considerando tanto a qualidade da futura progênie quanto o grau de consanguinidade. O grau de consanguinidade é um coeficiente que mede o parentesco entre os pais do animal.

Sendo assim, escolher bons touros e matrizes está entre uma das práticas importantes para a propriedade que participa de um programa de melhoramento genético, mais especialmente os touros. Comparativamente, um touro pode deixar muito mais filhos do que uma vaca, quer pela monta natural ou

por inseminação artificial, então a escolha de bons pais deve seguir os requisitos da propriedade para que possa refletir em filhos com a qualidade esperada.

A Figura 2.8 apresenta uma ficha com os resultados da avaliação genética de um touro, retirada do Sumário de Touros da Raça Nelore - Programa Geneplus.

Figura 2.8: Ficha com resultados de avaliação genética de um animal

Como pode ser observado, nesta ficha estão todas as informações a respeito do animal e sua avaliação genética. No topo há uma série de dados cadastrais do animal, como nome, identificador e seus pais.

Logo abaixo, já segue um quadro com as DEPs para cada uma das características avaliadas, suas respectivas acurácias, percentis e classes, bem como a chamada DEP gráfica correspondente.

A DEP é o resultado da avaliação genética e geralmente vem acompanhada da acurácia, um indicador da sua confiabilidade.

Para algumas DEPs não há informação de acurácia pois são resultados de operações matemáticas envolvendo outras DEPs. Por exemplo, a característica Total Maternal aos 120 dias - TM120 (kg), é resultado da operação descrita na Equação 2.24.

$$TotalMaternal = \frac{DEPdireta}{2} + DEPmaterna \tag{2.24}$$

Para todas as DEPs há um valor indicando a posição do animal em relação aos demais avaliados. Este valor, chamado percentil, varia de 0,1 a 99%, e foi categorizado em quatro classes: `elite` (0,1% a 16%), `superior` (17% a 50%), `regular` (51% a 84%) e `inferior` (85% a 99%).

Um índice, chamado Índice de Qualificação Genética Geneplus (IQG/GP), reúne DEPs de diferentes características, ponderadas quanto à sua importância, em um único valor numérico que também possuirá um percentil e uma classe associada.

A chamada DEP gráfica é um artifício criado para facilitar a análise de todas as DEPs visualmente. Baseando-se no percentil de cada DEP, barras vermelhas e azuis são geradas para indicar o quanto elas são negativas e positivas, respectivamente. Pela Figura 2.8, uma DEP com percentil (AT%) de 98%, é muito negativo, por isso a barra vermelha é grande. Já para um percentil (AT%) de 63%, o animal é negativo, mas nem tanto, por isso a barra vermelha é menor. No mesmo racioncínio, um percentil (AT%) de 44% é positivo, mas nem tanto. Um percentil (AT%) de 0,1% é totalmente positivo, com dois asteriscos, para dar destaque.

No caso da raça Nelore, o sumário apresenta dois tipos de percentis: O ativo (AT%) e o da população (POP%). A diferença entre os dois é que o percentil na população classifica os animais de acordo com a população toda e o percentil nos ativos, classifica os animais de acordo com uma pequena parte da população, a dos chamados ativos, que são os animais nascidos nos últimos 5 anos ou que produziram filhos nos últimos 5 anos.

O percentil nos ativos é um indicador mais rigoroso, pois os animais são classificados com base em um grupo de animais geneticamente superiores, com médias de valores genéticos superiores àquelas da população (Nobre et al., 2013a).

Para complementar, se o animal for matriz ou touro, há ainda informações a respeito do número de filhos e em quantos rebanhos eles aparecem. Há contadores para filhos que somente nasceram e podem não ter contribuído com nenhuma informação de peso, e há contadores para somente os filhos que contribuíram com algum peso na avaliação.

Para os touros que possuem sêmen disponível em centrais de inseminação, também pode-se consultar em qual central e o telefone de contato.

Todas estas informações apresentadas no sumário serão úteis ao produtor se for de seu conhecimento como ele foi elaborado, para qual população, período, etc. O sumário varia pois a população avaliada varia. De um programa de melhoramento para outro, a população e os métodos de avaliação podem ser diferentes, e dentro de um mesmo programa de melhoramento, há mudanças frequentes pois a população está sempre crescendo, e as coletas de características acontecem em várias fases da vida do animal.

Portanto, para que o uso de um sumário seja otimizado, o criador deve ter claro o objetivo de seleção de seu rebanho e as características a serem consideradas para que o objetivo seja alcançado (Silva et al., 2013).

2.3 O Programa GENEPLUS

O Programa Embrapa de Melhoramento de Gado de Corte - GENEPLUS foi disponibilizado aos criadores em 1996, pela Embrapa Gado de Corte em parceria com a Geneplus Consultoria Agropecuária Ltda. Trata-se de um programa de melhoramento genético que presta serviço de assessoria personalizada aos criadores de gado de corte na utilização dos recursos genéticos de seu rebanho, considerando as características do seu sistema de produção e seus objetivos de seleção (Nobre et al., 2013a).

Para melhor entender, sistema de produção de gado de corte refere-se ao conjunto de tecnologias e práticas de manejo, bem como o tipo de animal, o propósito da criação, a raça ou grupamento genético e a ecorregião onde a atividade é desenvolvida (Euclides Filho, 2000b).

Devido à dimensão continental, à variedade de ecossistemas e à diversidade socioeconômica do Brasil, os sistemas de produção podem ser divididos em três tipos de acordo com o regime alimentar: sistema extensivo - regime exclusivo de pastagem; sistema semi-intensivo - pastagem mais suplementação em pasto; e sistema intensivo - pastagem mais suplementação e confinamento (Cezar et al., 2005). Informações do sistema de produção são, portanto, importantes fatores para possibilitar uma justa comparação entre os animais de diferentes fazendas e as características observadas.

Os objetivos de seleção são imprescindíveis para o norteamento de um programa de melhoramento genético e podem ser definidos como sendo uma combinação de características economicamente importantes para o rebanho, que levem a estratégias custo-efetivas para se obter animais com maiores eficiências produtiva e reprodutiva, mantendo um balanço adequado entre

características de produção, fertilidade e adaptabilidade (Júnior et al., 2007).

Definidos os conceitos de sistemas de produção e objetivos de seleção, fica claro que para a implantação de um programa de melhoramento, como o Programa GENEPLUS, é necessário definir os objetivos e critérios de seleção, conhecer a infraestrutura disponível para a condução do projeto e para a coleta dos dados, e utilizar as informações geradas no processo de seleção e/ou no estabelecimento dos planos de acasalamentos (Nobre et al., 2013b). Assim, o primeiro passo é construir um plano de trabalho que, posto tudo isto, delineie os caminhos a seguir.

No plano de trabalho, delineado pela equipe técnica do Programa GENEPLUS em conjunto com a equipe técnica da fazenda, constam quais características serão monitoradas, qual a participação de cada uma delas no processo final de decisão e em qual momento deverão ser mensuradas ao longo do processo de criação dos animais (Nobre et al., 2013b).

Ao produtor fica a responsabilidade de coletar os dados dos animais em sua fazenda, segundo as idades e a metodologia descrita no plano. Nos prazos estipulados, os dados são remetidos ao Programa GENEPLUS e passam por análises de consistência para atualização do banco de dados geral da raça. Após a completa atualização da base, uma base de dados totalmente numérica é gerada e liberada para os pesquisadores responsáveis pela execução da avaliação genética. Os resultados virão na forma de DEPs para cada característica de cada produto. As DEPs são devolvidas aos criadores em *softwares* de consulta aos resultados.

A Figura 2.9 apresenta a dinâmica do Programa GENEPLUS. Para a raça Nelore, em junho e em novembro de cada ano são executadas as avaliações genéticas. Então, em março e agosto os produtores têm que enviar os dados de suas fazendas para o Programa GENEPLUS. São variados tipos de *backups* que precisam ser padronizados para que depois possam alimentar o banco de dados geral.

Assim, à medida em que vão chegando, os dados são preparados isoladamente, passando por análises de consistência, de modo que ao final cada conjunto de dados contenha a genealogia e as características de interesse para a raça, já filtradas para os intervalos de tempo definidos no plano de trabalho.

Cerca de 173 *backups* vão alimentar o banco de dados geral. Análises de consistência são realizadas em busca de duplicidades, valores extremos, valores perdidos, entre outras incoerências. Com o banco atualizado e limpo, a avaliação genética é executada sobre toda a população e os resultados são

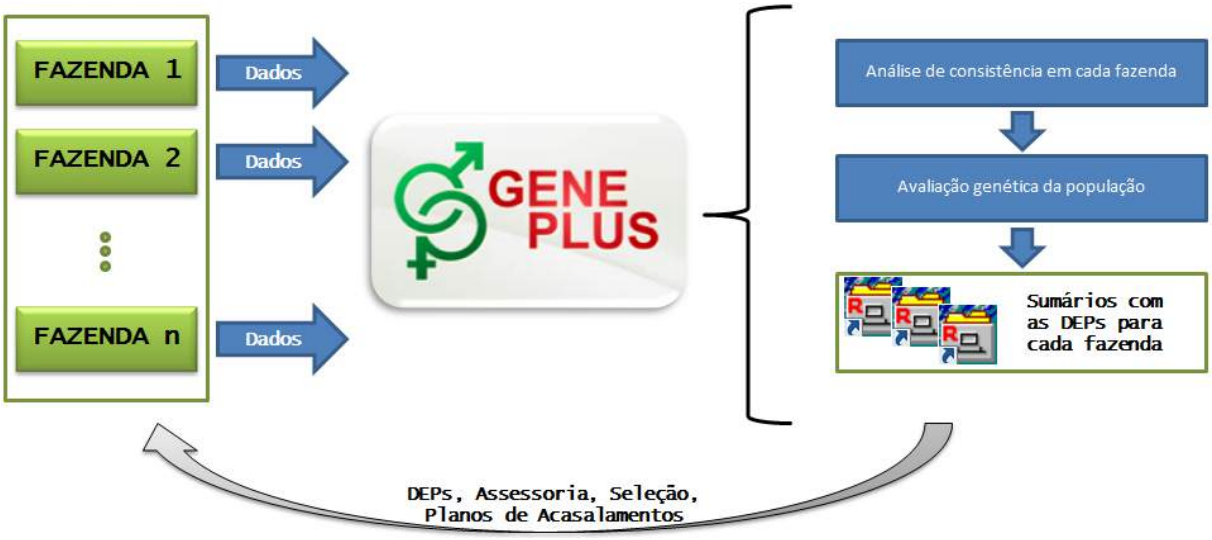


Figura 2.9: Dinâmica do Programa GENEPLUS

enviados aos usuários em forma de Sumários de Touros, Matrizes e Produtos, conforme Figura 2.10.

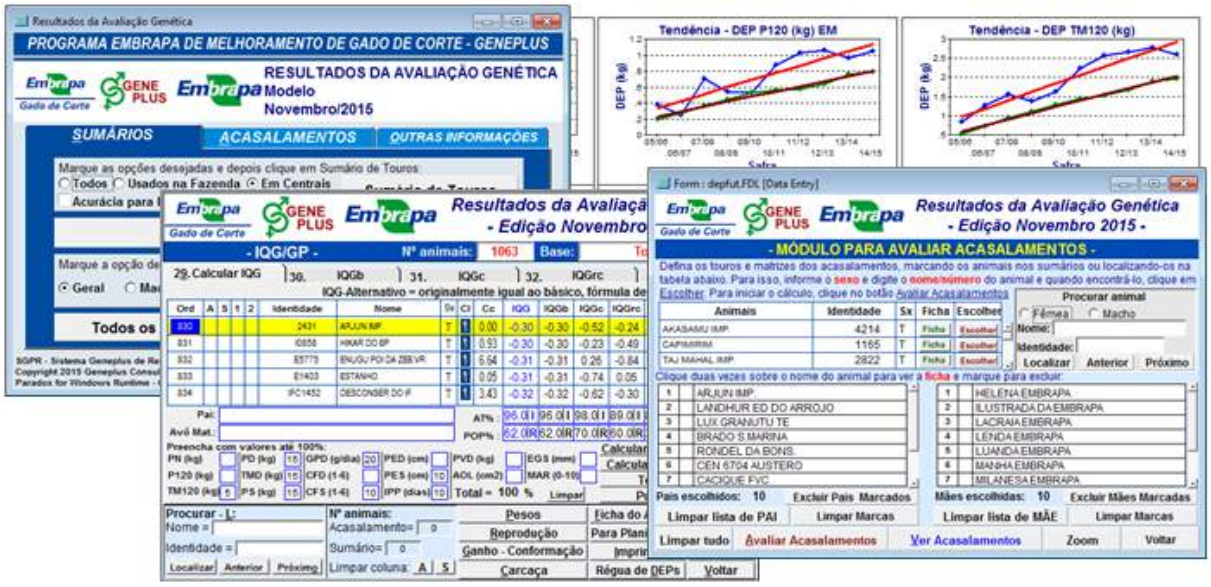


Figura 2.10: SGPR - Sistema GENEPLUS de Resultados

Estes sumários conterão as DEPs, respectivas acurácias (precisão das DEPs) e percentis (*ranking* na população), e informações gerais sobre o desempenho do rebanho de cada produtor participante do Programa GENEPLUS, como tendências genéticas e médias comparativas. São chamados SGPR - Sistema GENEPLUS de Resultados, e constituem ferramentas para tomada de decisão. Pode-se consultar o *ranking* dos animais, excluir, calcular índices, calcular acasalamentos futuros, emitir fichas com as avaliações genéticas, além de re-

latórios múltiplos.

Analisando o rebanho no SGPR, o proprietário consegue comparar sua fazenda com a média geral da população, através de tendências genéticas e médias. Consegue investigar quais touros melhoraram seu rebanho, e quantos foram utilizados no decorrer dos anos.

Com os sumários, os técnicos do Programa GENEPLUS prestam assessoria aos produtores na interpretação dos resultados, na utilização da ferramenta para calcular os acasalamentos, realizam visitas periódicas para coleta de pesagens e avaliações de escores visuais, e fazem toda a intermediação entre produtor e o Programa como um todo.

A visão do Programa GENEPLUS é que a avaliação genética é uma ferramenta de trabalho na busca de maior produção de quilogramas de carne por hectare, em determinado tempo e a menores custos (Nobre et al., 2013b). A meta a ser atingida é a perfeita correlação entre os objetivos definidos e os resultados obtidos.

O Programa GENEPLUS atende a outras raças, a saber: Canchim, Caracu, Senepol, Brangus, Braford-Hereford, Guzerá, Brahman e Santa Gertrudis. Possui técnicos de campo credenciados para atender em diversas regiões do Brasil, e conta com uma equipe de pesquisadores e analistas da Embrapa, de universidades e outras instituições. É, portanto, uma tecnologia de melhoramento animal disponibilizada à comunidade produtora de carne.

2.4 Descrição dos Atributos da Base de Dados do Programa Geneplus

As amostras utilizadas nos experimentos foram extraídas do banco de dados do Programa GENEPLUS, raça Nelore. A Figura 2.11 abaixo é um exemplo dos principais atributos do banco de dados do Programa GENEPLUS, os quais serão descritos em seguida.

O banco de dados do Programa GENEPLUS contém além dos atributos de identificação e genealogia, atributos para pesos (ao nascimento, aos 4 meses, à desmama e ao sobreano), medidas (perímetro escrotal à desmama e ao sobreano, conformação frigorífica à desmama e ao sobreano) e ultrassom (área de olho de lombo, espessura de gordura subcutânea e marmoreio), além de idade ao primeiro parto e peso da mãe à desmama de sua cria. Todos os animais, tendo coletado ou não estes atributos, terão uma avaliação genética correspondente para cada uma das características monitoradas.

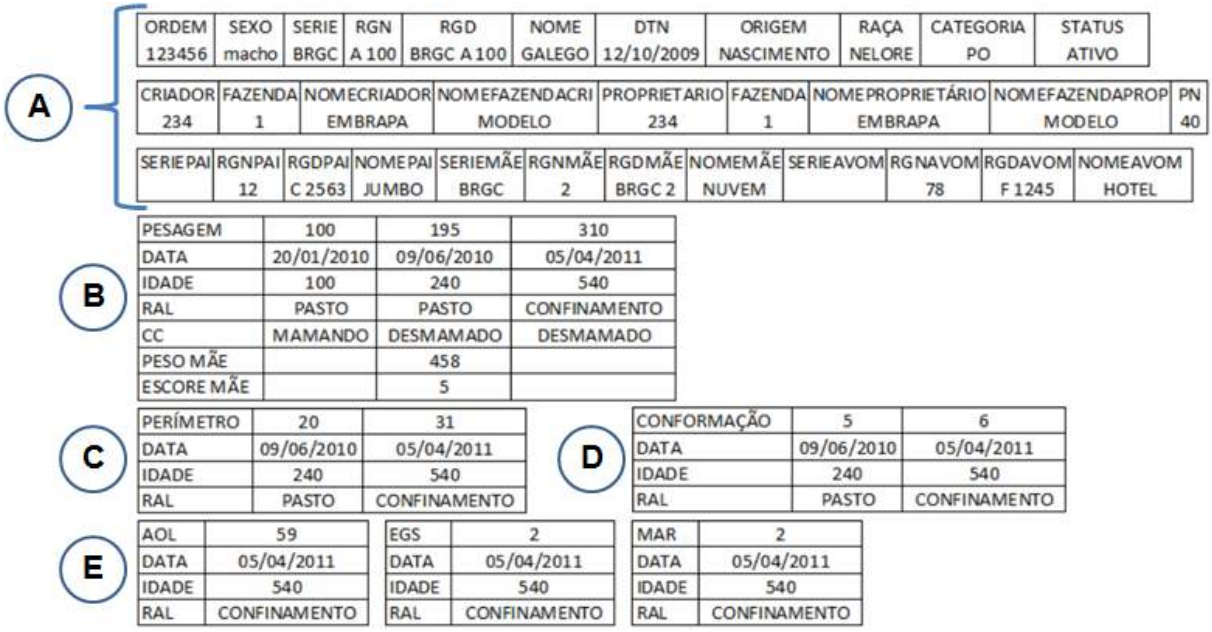


Figura 2.11: Principais atributos da base de dados do Programa GENEPLUS, raça Nelore

Atributos de identificação e *Pedigree*

Na Figura 2.11-A, estão os atributos de identificação e *pedigree*. Cada animal obrigatoriamente deve possuir o Registro Genealógico ao Nascimento (RGN) ou o Definitivo (RGD) que são os identificadores dos animais. Vários animais podem possuir o mesmo RGN mas somente um animal possui o RGD. Ambos aceitam letras no início e números no final. Na raça Nelore e nas zebuínas em geral, o número de dígitos aceito é de no máximo cinco, o número de letras para RGN é de zero a duas, e o número de letras para RGD é de até quatro.

A identificação é um problema à parte pois são atributos passíveis de erros de digitação. O correto é ter um RGD único, mas pode ocorrer de o animal não receber o RGD por ser desclassificado pela instituição que certifica a raça. Então, este animal pode permanecer no rebanho somente com o RGN, que pode ser repetido ano a ano.

Até 1997 o RGD era um registro fornecido pela Associação Brasileira dos Criadores de Zebu - ABCZ aos animais controlados, e não tinha nada em comum com o RGN. A partir de 1997 a ABCZ passou a utilizar o Sistema Único de Identificação - SUI, onde o RGD do animal passava a ser formado por uma série de letras exclusiva da fazenda mais o RGN. Para o exemplo da figura, a série do criador/fazenda igual a 234/1 é “BRGC”, e o RGD será a

junção da série com o RGN, ficando igual a “BRGC A 100”. O pai deste animal foi identificado no sistema antigo, observando-se que o RGD igual a “C 2563” é bem diferente do seu RGN igual a “12”.

Quando um animal é vendido ou comprado, ou seja, sua origem não é nascimento na fazenda onde está, seu código de criador será diferente do código de proprietário. Criador refere-se ao nascimento do animal e proprietário refere-se ao atual local onde o animal se encontra. São códigos numéricos e cada um pode ter uma ou mais fazendas, ou seja, um criador pode ter uma fazenda em Campo Grande/MS, outra em Três Lagoas/MS e outra em Marabá/PA.

Os códigos de criador, proprietário e respectivas fazendas são importantes para a formação de grupos contemporâneos, indispensáveis para que as comparações de desempenho dos animais aconteçam.

Grupos contemporâneos são formados por animais nascidos num mesmo ano e época, manejados de forma similar dentro de um determinado rebanho (Cobuci et al., 2006). Para que as comparações sejam válidas e efetuadas de forma eficiente, os grupos contemporâneos precisam ter um número razoável de indivíduos, que sejam conectados geneticamente e que agrupem animais que tiveram seus desempenhos influenciados pelas mesmas condições ambientais (clima, alimentação, sanidade, manejo, etc.). Grupos de único animal devem ser desconsiderados. O tamanho do grupo é decidido no momento da avaliação genética.

Com as informações de pai e mãe do animal fica completa a parte de identificação e de pedigree. O cadastro dos avós, bisavós, tataravós, e dos ancestrais possíveis é importante para que os laços de parentesco sejam considerados no momento da avaliação genética. Isso agrega mais precisão às estimativas de valor genético (Rosa et al., 2013).

Atributos de Pesagens

Na Figura 2.11-B, estão os atributos de Pesagens. Dentro do plano de trabalho de melhoramento genético da fazenda estão descritas quais as características a serem monitoradas, a rotina de coleta de dados envolvendo as fases de acasalamento, nascimento, cria e recria dos animais e as estratégias de melhoramento genético a serem aplicadas. Para o caso do Programa GENEPLUS, as características de desempenho interessantes para a raça Nelore são coletadas em três idades: 120 dias, desmama (240 dias) e sobreano (550 dias).

As fazendas podem fazer várias pesagens no decorrer da vida do animal, e essas pesagens são feitas em lotes ou grupos de manejo, de modo que as

idades (120, 240 e 550 dias) servem como parâmetros pois muito raramente os animais de um mesmo lote são pesados exatamente com 240 dias, por exemplo. É dado um intervalo de 45 dias para mais ou para menos para cada idade padrão, havendo duas pesagens que se encaixam para o intervalo do sobreano, vale a mais próxima dos 550 dias. As pesagens que não servem para nenhum intervalo, são descartadas.

Cada pesagem basicamente deve vir acompanhada da data e do regime alimentar (pasto, semi-confinamento, confinamento). Algumas pesagens apresentam condição de criação indicando se o animal é saudável, se está mamando ou se já desmamou. Quanto mais informação a respeito da pesagem, melhor para a formação do grupo contemporâneo.

O melhoramento genético é um investimento de longo prazo. Todo o trabalho do criador em coletar as pesagens desde o nascer e em cada fase de acompanhamento da vida do animal tem um propósito bem definido para auxiliar no progresso genético do seu rebanho. Ao se avaliar o peso ao nascimento, por exemplo, busca-se informações que auxiliem a reduzir ou eliminar problemas de dificuldade de parto. Medir o peso aos 120 dias e à desmama visa avaliar a capacidade de crescimento do animal e avaliar a habilidade da mãe em proporcionar produção de leite e cuidados dispensados à cria. Ao sobreano, a pesagem deve refletir a capacidade do animal em ganhar peso no período pós-desmama e a partir dos ganhos de peso pode-se escolher animais mais precoces (Martín Nieto and Rosa, 2013).

Atributos de Reprodução

Na Figura 2.11-C, estão os atributos de reprodução. Elevadas taxas de fertilidade e filhos saudáveis são condições essenciais para que o produtor disponha de um maior número de animais tanto para o mercado quanto para a seleção, favorecendo o retorno dos investimentos em genética superior (Martín Nieto and Rosa, 2013). Para tanto, podem ser coletadas características reprodutivas como idade ao primeiro parto, dias para criar, período de gestação, idade à puberdade, perímetro escrotal, taxa de sobrevivência.

As características reprodutivas acompanhadas pelo Programa GENEPLUS são idade ao primeiro parto e perímetro escrotal à desmama e ao sobreano. A idade ao primeiro parto, medida em dias, marca o início do processo reprodutivo das fêmeas. A redução antecipa a idade produtiva, proporciona recuperação mais rápida do investimento, aumenta a vida útil, possibilita maior intensidade de seleção nas fêmeas e reduz o intervalo entre gerações

(Martín Nieto and Rosa, 2013).

O perímetro escrotal é uma medida tomada em centímetros, e está associada ao desempenho reprodutivo dos machos, indicadora da quantidade de sêmen que pode ser produzida pelo touro. O objetivo é obter animais mais precoces sexualmente (Martín Nieto and Rosa, 2013). O perímetro escrotal geralmente é coletado na mesma data da pesagem à desmama e/ou ao sobreano, por isso, a data e o regime alimentar já estão implícitas, caso contrário, precisam de, pelo menos, a data da medida.

Atributos de Escores Visuais

Na Figura 2.11-D, estão os atributos de escores visuais. Dentre as muitas características coletadas para gado de corte, existem características ditas subjetivas, que dependem da coerência dos especialistas ao darem notas de avaliação. São diferentes de características como peso e perímetro escrotal que podem ser realmente medidas em balanças e fitas métricas. Para estas características subjetivas, somente técnicos são credenciados a avaliar os animais porque é necessário conhecimento e experiência em observar a estrutura morfológica, musculatura, harmonia, caracterização racial, caracterização sexual, entre outras.

A conformação frigorífica é uma característica subjetiva coletada juntamente com a pesagem à desmama e/ou ao sobreano. Trata-se de um indicador numérico para avaliar aspectos morfológicos dos animais, com notas variando de 1 a 6, atribuídas por avaliadores. Como não é uma medida e sim uma avaliação visual ela é dita ser subjetiva.

Quando os avaliadores dão a nota, são observadas três categorias do aspecto do animal: estrutura, musculosidade e precocidade de acabamento. A estrutura envolve a harmonia do comprimento do animal, profundidade e arqueamento de costelas. A musculosidade envolve a quantidade e a forma da massa muscular que envolve a estrutura, sendo analisados os pontos onde, abaixo do couro, predomina o tecido muscular, como por exemplo, o braço e a coxa. A precocidade de acabamento examina a deposição de gordura na carcaça, nos pontos onde, abaixo do couro, observa-se somente esqueleto e gordura. A inserção da cauda quando apresenta dobras de gordura, por exemplo, é um indicador (Nobre, 1996).

Atributos de Ultrassom

Na Figura 2.11-E, estão os atributos de ultrassom. Frente às exigências por qualidade e às novas expectativas de mercado, as fazendas têm buscado elevar seus índices produtivos com recursos da área de ultrassonografia, acompanhando características relacionadas ao produto final, como melhor rendimento de cortes cárneos, maior qualidade em termos de suculência e sabor, e maior precocidade de acabamento (Suguisawa et al., 2013).

Os animais são avaliados preferencialmente ao sobreano, idade na qual ocorre o máximo da expressão do crescimento e desenvolvimento corporal, quando o animal está mais próximo do peso de abate (Suguisawa et al., 2013). Os dados coletados são referentes a área de olho de lombo, espessura de gordura subcutânea e marmoreio.

A área de olho de lombo caracteriza-se como a área do músculo *Longissimus dorsi*, medida em cm^2, entre a 12^a e a 13^a costela e está relacionada com a porção comestível da carcaça. Do ponto de vista genético, a área de olho de lombo refere-se ao potencial do animal para musculosidade, crescimento, ganho de peso e relação músculo/osso nos cortes de maior interesse econômico da carcaça (Suguisawa et al., 2013).

A espessura de gordura subcutânea, em mm, indica o grau de acabamento da carcaça, expresso pela deposição de gordura. Sua importância está na proteção da carcaça contra a queda de temperatura nas câmaras frias, que pode provocar a perda de maciez e o escurecimento da carne nas carcaças pobres em acabamento. É indicativo de precocidade sexual e de terminação, ou seja, animais que iniciam a deposição de gordura mais cedo tendem a ser mais precoces sexualmente e tendem a apresentar carcaças prontas para o abate em menores idades (Suguisawa et al., 2013).

O marmoreio, avaliado por notas de 1 a 10, indica a quantidade de gordura intramuscular, associada à suculência e ao sabor da carne. A maior ou menor deposição deste tipo de gordura entremeada está ligada principalmente ao fator genético (Suguisawa et al., 2013).

2.5 Conjuntos de dados para os experimentos

Em janeiro de 2014 foi executada uma avaliação genética que gerou DEPs para 1.986.915 animais do banco de dados geral da raça Nelore, Programa GENEPLUS. Foi esta a base de dados utilizada para a montagem dos conjuntos de dados de treinamento e teste, e que passou por diversos filtros até que

chegasse em uma Base Melhorada final, como pode ser vista na Figura 2.12 abaixo.

Figura 2.12: Etapas de preparação da base de dados para os experimentos

Na Etapa 1 então, começa-se com a base original e seus 1.986.915 animais. Depois, na Etapa 2, o primeiro filtro: o conjunto de dados representa um conjunto de animais criados a pasto, todos com peso válido à desmama (com idade entre 166 e 285 dias) e todos com pai e mãe conhecidos, totalizando um conjunto com 200.878 animais. A amostra foi então dividida entre machos e fêmeas, 96.911 e 103.967 animais respectivamente.

Estes dados já estavam preparados de maneira que os animais já estavam alocados em grupos contemporâneos para a pesagem à desmama e estavam consistentes quanto à unicidade dos animais e intervalo válido dos valores dos pesos. Os grupos contemporâneos foram formados agrupando-se animais que foram pesados na mesma propriedade, data de pesagem, sexo, regime alimentar à fase materna e regime alimentar à desmama.

O número mínimo de 10 animais por grupo contemporâneo foi definido, buscando-se assim, trabalhar com grupos maiores e representativos da população. Segundo (Cobuci et al., 2006), o efeito do tamanho dos grupos contemporâneos está relacionado com a acurácia das predições das DEPs, ou seja, quanto maior o grupo, maior será a acurácia. Para o experimento, esta restrição entra para que o treino se dê com animais que tiveram mais chance de colaborar com os resultados das DEPs. Na Etapa 3 então, os conjuntos de

dados de machos e de fêmeas foram reduzidos para 81.940 e 85.567 respectivamente por causa da limpeza por grupos contemporâneos.

Na Etapa 4, uma nova limpeza nos dados foi executada, excluindo-se os animais que não possuíam avô materno conhecido. Como houve exclusão destes animais, alguns grupos contemporâneos ficaram com poucos animais. Por isso, todos os grupos envolvidos nesta limpeza também foram retirados. Restaram 49.169 dados de animais machos e 51.341 dados de fêmeas, os quais daqui para frente serão referenciados como Base Selecionada.

Os experimentos foram realizados na Base Selecionada dos machos, inicialmente, e os atributos que compõem esta base são descritos resumidamente na Tabela 2.12, a fim de facilitar a visualização. A tabela expandida está no Apêndice C, e é o mesmo formato para a amostra das fêmeas.

Campo	**Descrição**
Número_pai	Número do pai do animal
Número_mãe	Número da mãe do animal
PD	Peso à desmama
GCD	Grupo contemporâneo
PDaju	Peso à desmama ajustado
Peso ao nascer	DEP, Percentil e Classe para: - Animal, - Pai, - Mãe e - Avô materno
Peso aos 120 dias	
Peso à desmama	
Peso ao sobreano	
Ganho pós-desmama	
Perímetro escrotal à desmama	
Perímetro escrotal ao sobreano	
Conformação frigorífica à desmama	
Conformação frigorífica ao sobreano	
Número_Avô_Materno	Número do avô materno

Tabela 2.12: Lista resumida de atributos da Base Selecionada

Os números do pai, da mãe e do avô materno são os mesmos utilizados na avaliação genética e foram deixados na amostra para testar se há influência deles no processo de classificação dos animais, seja isoladamente ou em uma combinação dos mesmos. É muito comum entre os produtores e melhoristas se referir a um animal pelo seu pai e pelo seu avô materno. Algo como: "Este touro é filho de Graveto em vaca Dirigível", ou seja, o animal é filho do touro Graveto e neto do touro Dirigível. Por isso o interesse em manter estes atributos.

O PDaju (peso ajustado) retrata a distância do PD (peso à desmama) relativa à média do PD por grupo contemporâneo, como na Equação 2.25, onde PD é

o peso à desmama, $\bar{g}$ é a média do PD dentro do grupo contemporâneo g e h^2 é a herdabilidade.

$$PDaju = (PD - \bar{g}) .h^2 \tag{2.25}$$

> **Exemplo 11** Por exemplo, suponha um grupo com 10 animais, e que a média do PD deste grupo seja 120kg. Cada um dos 10 animais deste grupo terá PDaju = (PD – 120kg) . 0,21. O valor 0,21 trata-se do coeficiente de herdabilidade para a característica PD. É uma constante, obtida da avaliação genética da raça e neste contexto servirá para indicar a porcentagem de peso herdada geneticamente.

Os campos referentes às DEPs apresentam-nas em seu estado original, com o mesmo conteúdo apresentado nos sumários. Cada DEP possui um percentil e uma Classe correspondente. O percentil indica a posição do animal relativa ao total de animais avaliados e é dado pela Fórmula 2.26, conforme já visto no capítulo anterior.

$$Percentil_{CALC} = \frac{DEP_{Animal} - \text{Média DEP}}{\sqrt{\text{variância}_{PD}}} \tag{2.26}$$

O $Percentil_{\text{CALC}}$ servirá como índice a ser localizado na tabela apresentada no Apêndice F e desta tabela, finalmente o Percentil da DEP será extraído para indicar em qual faixa dos melhores avaliados o animal se encontra, variando de 1 a 99% em valores inteiros, e, para dar um destaque aos primeiros colocados, há a faixa dos 0,1% e dos 0,5%.

Como já informado, este pecentil servirá para agrupar os animais em quatro Classes: `elite`, com animais até 16%; `superior`, com animais de 17% a 50%; `regular`, com animais de 51% a 84%; e `inferior` com animais acima de 84%.

Para melhor ilustrar estes três campos, abaixo um exemplo:

> **Exemplo 12** Seja a DEP para peso à desmama igual a 2,44 kg. A média de todas as DEPs para peso à desmama é igual a -0,1921789 kg e a variância do peso à desmama é 13,4091064. Para o nosso exemplo, o percentil correspondente será 0,718812275, que pela tabela no Apêndice F, corresponderá a 24%, ou seja, classe `superior`. A dúvida reside em qual atributo poderá ser mais vantajoso para um classificador eficiente: a DEP (2,44) ou o percentil (24%) ou a Classe (`superior`).

A distribuição das Classes nos experimentos é conforme a Tabela 2.13 e, como pode ser visto, as amostras são proporcionais em números de animais por Classe. Os números não condizem com uma distribuição normal pois os conjuntos de dados são amostras da população.

	Distribuição das Classes					
	Machos			**Fêmeas**		
	Etapa 3	**Etapa 4**	**Etapa 5**	**Etapa 3**	**Etapa 4**	**Etapa 5**
Elite	26.206	18.977	4.395	27.971	20.399	5.431
	32%	39%	38%	33%	40%	42%
Superior	32.826	20.222	4.648	34.401	20.913	4.962
	40%	41%	41%	40%	41%	39%
Regular	17.035	7.881	1.808	17.162	7.775	1.774
	21%	16%	16%	20%	15%	14%
Inferior	5.873	2.089	608	6.033	2.254	621
	7%	4%	5%	7%	4%	5%

Tabela 2.13: Distribuição das classes nos conjuntos de dados

Os atributos descritos na Tabela 2.12 compuseram a amostra de machos da Etapa 4, utilizada nos primeiros experimentos. Posteriormente, muitos deles foram excluídos pois não estavam contribuindo para o desempenho do classificador.

Da mesma forma que excluíram-se alguns, outros foram incluídos para mais testes. Dentre eles, testamos uma série de atributos envolvendo a DEP futura do animal para cada uma das características, e que vem a ser a DEP somente calculada pela média dos seus pais.

Para um animal que possua pai e mãe conhecidos, mesmo que ele não tenha o peso à desmama, ele terá uma DEP equivalente a esta característica, baseada nas DEPs de seus pais. Então, se um animal na época da avaliação genética, não tinha idade suficiente para ter o peso à desmama, por exemplo, a sua DEP será calculada como sendo a média das DEPs de seus pais, conforme a Fórmula 2.27. Para esta $DEP_{\text{PaiMãe}}$ haverá um percentil e uma Classe correspondente.

$$DEP_{\text{PaiMãe}} = \frac{DEP_{Pai} + DEP_{\text{Mãe}}}{2} \tag{2.27}$$

Sendo assim, testes foram executados também com a inclusão da $DEP_{\text{PaiMãe}}$, percentil e Classe para cada uma das características.

Ao final dos experimentos, a Base Melhorada (Etapa 5) foi o refinamento da Base Selecionada, onde os animais que permaneceram somente possuíam o peso à desmama, não possuindo o peso ao sobreano, e os seus atributos

resumiram-se ao descrito na Tabela 2.14. Os novos atributos serão descritos nas seções futuras, à medida em que forem sendo utilizados.

Campo	Descrição
~~Número pai~~	~~Número do pai do animal~~
~~Número mãe~~	~~Número da mãe do animal~~
~~PD~~	~~Peso à desmama~~
~~GCD~~	~~Grupo contemporâneo~~
PDaju	Peso à desmama ajustado
Peso ao nascer	DEP, ~~Percentil e Classe~~ para: - ~~Animal~~, - Pai, - Mãe e - ~~Avô materno~~
Peso aos 120 dias	
Peso à desmama	
Peso ao sobreano	
Ganho pós-desmama	
Perímetro escrotal à desmama	
Perímetro escrotal ao sobreano	
Conformação frigorífica à desmama	
Conformação frigorífica ao sobreano	
~~Número Avô Materno~~	~~Número do avô materno~~
Escore E	**Escore para a Classe `elite`**
Escore S	**Escore para a Classe `superior`**
Escore R	**Escore para a Classe `regular`**
Escore I	**Escore para a Classe `inferior`**
Classe DEP Peso Desmama	Classe do animal

Tabela 2.14: Lista resumida de atributos da Base Melhorada. Os atributos em negrito foram adicionados e os riscados foram excluídos

CAPÍTULO

3 Avaliação Experimental

Dentro do objetivo de se encontrar um classificador que melhore a previsão da Classe para a DEP de Peso à Desmama para um conjunto de animais cujo peso entrará como atributo válido, vários experimentos foram executados, alternando-se entre o pacote de software WEKA (*Waikato Environment for Knowledge Analysis*)[1], e implementações da biblioteca `SKLEARN`[2]. Com a Base Selecionada montada, pôde-se dar início aos experimentos que desencadearam na seleção dos atributos com maiores ganhos de informação e do algoritmo classificador mais eficiente para atender o problema. Nas sessões que se seguem, todos os passos e questionamentos que levaram à Base Melhorada e ao algoritmo de Árvore de Decisão.

3.1 Considerando as informações de curral, quais atributos são relevantes para a predição?

Para cada animal existem duas situações:

(1) O animal já foi pesado

(2) O animal será pesado

Se um animal já foi pesado, o resultado de sua avaliação genética decorrerá da influência do *pedigree* mais a contribuição deste peso. Se ele ainda não foi

[1] `http://www.cs.waikato.ac.nz/~ml/weka/`
[2] `http://scikit-learn.org/`

pesado, sua avaliação genética é calculada somente pela média das DEPs de seus pais, segundo a Fórmula 2.27.

Para saber quanto é o erro de se classificar um animal pela situação 1 e depois pela situação 2, foi feito um estudo utilizando o conjunto de dados de 49.169 machos da Etapa 4. O resultado deste estudo comporá o *baseline*, usado para comparações com os demais classificadores.

Sendo assim, tratar $Classe_{\text{PaiMãe}}$ como a Classe da $DEP_{\text{PaiMãe}}$ do peso à desmama (situação 2), e $Classe_{\text{ALVO}}$ como a Classe da DEP_{Real} (situação 1).

A Tabela 3.1 apresenta a comparação entre prever a $Classe_{\text{PaiMãe}}$ e a verdadeira $Classe_{\text{ALVO}}$. Descreve os erros médios absolutos (MAE) calculados para cada Classe, bem como as métricas: `accuracy`, `f1-measure`, `precision`, `recall` e as taxas de Verdadeiro Positivo (TP), Verdadeiro Negativo (TN), Falso Positivo (FP) e Falso Negativo (FN).

Desempenho da previsão das Classes				
Total de animais:	**elite**	**superior**	**regular**	**inferior**
49.169 Machos	**18,977**	**20,222**	**7,881**	**2,089**
Número de Erros	3.200	3.788	2.642	709
MAE	0,169	0,187	0,337	0,345
accuracy	0,881	0,808	0,908	0,981
f1-measure	0.844	0,777	0,699	0,751
precision	0,857	0,745	0,737	0,870
recall	0,831	0,812	0,665	0,661
TP	15.777	16.434	5.239	1.380
TN	27.563	23.315	39.416	46.874
FP	2.629	5.632	1.872	206
FN	3.200	3.788	2.642	709
Pontos no Gráfico ROC	C0	C1	C2	C3

Tabela 3.1: Desempenho $Classe_{\text{PaiMãe}}$ versus $Classe_{\text{ALVO}}$

Analisando a $Classe_{\text{ALVO}} = \texttt{elite}$, com 18.977 animais: comparando-a com a $Classe_{\text{PaiMãe}}$, há 3.200 erros, onde animais deveriam ser `elite` porém foram classificados diferentemente. Destes 3.200 erros, 3.198 foram erros simples, onde $Classe_{\text{ALVO}} = 0(\texttt{elite})$ e $Classe_{\text{PaiMãe}} = 1(\texttt{superior})$; e 2 dos erros, a $Classe_{\text{ALVO}} = 0(\texttt{elite})$ e $Classe_{\text{PaiMãe}} = 2(\texttt{regular})$. Pela fórmula do MAE, o somatório dos erros foi de 3.202 entre 18.977 observações, resultando em uma porcentagem de erro de 16,87%.

`accuracy`, `f1-measure`, `precision` e `recall` foram calculados com base em taxas de verdadeiros e falsos, utilizando a estratégia um contra todos, binarizando para duas Classes (positivo e negativo) cada uma das quatro Clas-

ses do experimento. Para a $Classe_{\text{ALVO}} = \texttt{superior}$, por exemplo, 16.434 tinham $Classe_{\text{PaiMãe}} = Classe_{\text{ALVO}}$ (TP); 23.315 que também tinham $Classe_{\text{PaiMãe}} = Classe_{\text{ALVO}}$ porém não eram `superior` (TN); 5.632 eram alarmes-falsos (FP); e 3.788 eram da $Classe_{\text{ALVO}} = \texttt{superior}$ mas foram classificados diferentemente (FN).

A Figura 3.1 mostra o gráfico ROC para estas classificações discretas. Os quatro pontos estão mais para conservadores, com poucos erros, do que para liberais.

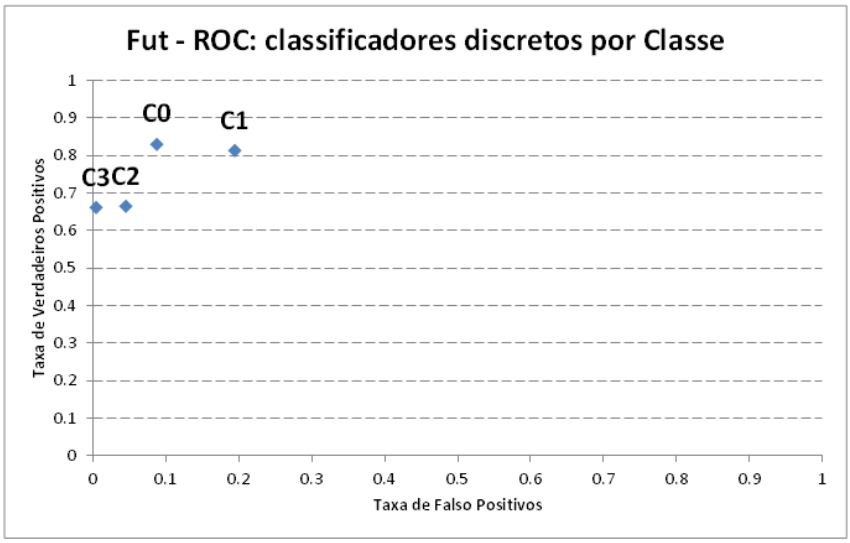


Figura 3.1: Gráfico ROC para o desempenho $\text{Classe}_{\text{PaiMãe}}$

Tudo isto foi para balizar a situação atual para que os classificadores vindouros possam ter parâmetros de comparação e assim serem escolhidos para melhorar o que já existe.

A Figura 3.2 retrata esta situação. De um lado, tem-se em mãos a DEP_{Real}, obtida via avaliação genética, que considerou os pesos e medidas coletadas do animal. Para este lado,o desempenho r_2 é de 100% de acerto já que é a $Classe_{\text{ALVO}}$.

Figura 3.2: Ilustrando a meta dos experimentos

Do outro lado, ao utilizar a Fórmula 2.27 para calcular a $DEP_{\text{PaiMãe}}$, a classificação alcança um desempenho r_1 com uma determinada taxa de acerto

menor do que r_2.

Ao induzir um classificador "c", sua taxa de acerto fica mais perto de r_1 ou r_2? Existirá vantagem se o desempenho r_x de "c" se aproximar de r_2.

Imaginando a situação de curral, em que o produtor está coletando as informações para a avaliação genética de um determinado lote de animais, o que se soma ao conjunto de dados para submissão a um classificador é basicamente o peso coletado. Neste ponto, só é possível calcular o r_1 da Figura 3.2. Quando "c" estiver definido, será possível prever a $Classe_{\text{ALVO}}$ com r_x garantindo que está mais próximo da correção e a expectativa é que o peso que acaba de ser coletado seja um atributo que faça a diferença.

Em busca do classificador que atendesse a esta meta, vários experimentos foram executados. Em um dos primeiros testes, submetendo a amostra de 49.169 machos da Etapa 4 ao WEKA, foi executado o módulo que avalia o ganho de informação de cada atributo. Percebeu-se que aqueles envolvendo as Classes contribuem um pouco menos do que as DEPs e percentis, como mostra o Apêndice B.

Ainda não havia nada conclusivo quanto à eliminação de atributos pois dentre os primeiros colocados aparecem os três tipos (DEPs, percentis e classes) misturados, envolvendo o peso à desmama, o que já era de se esperar.

Para tentar resolver esta questão inicial de discretizar ou não as DEPs, os conjuntos de dados da Etapa 4 foram desmembrados em três, conforme mostra a Figura 3.3.

O percentil e a Classe podem ser considerados dois modos de discretizar as DEPs, pois o primeiro agrupa as DEPs em categorias que vão de 0,1%, 0,5% e 1% até 99% (em números inteiros); e o segundo agrupa em quatro categorias: `elite`, `superior`, `regular` e `inferior`. Sendo assim, o Arquivo 1 contém PDaju e as DEPs, o Arquivo 2 contém PDaju e os percentis e o Arquivo 3 contém PDaju e as Classes. Os três têm como $Classe_{\text{ALVO}}$, ou rótulo, a Classe da DEP para o peso à desmama.

Os três conjuntos de dados foram submetidos ao classificador por algoritmo de Árvore de Decisão, via código em python (Apêndice D) que utiliza estratégia de classificação um versus todos, ajuste de parâmetros com busca aleatória (*RandomizedSearchCV*), validação cruzada (*StratifiedKFold*), e calibração com regressão isotônica. Na Tabela 3.2 constam os resultados das três rodadas em comparação com o `baseline`. E na Figura 3.4 os gráficos resultantes.

Como pode ser observado, as métricas para a rodada com o Arquivo 3, que categoriza as DEPs pelas quatro classes, foram as piores, ganhando somente

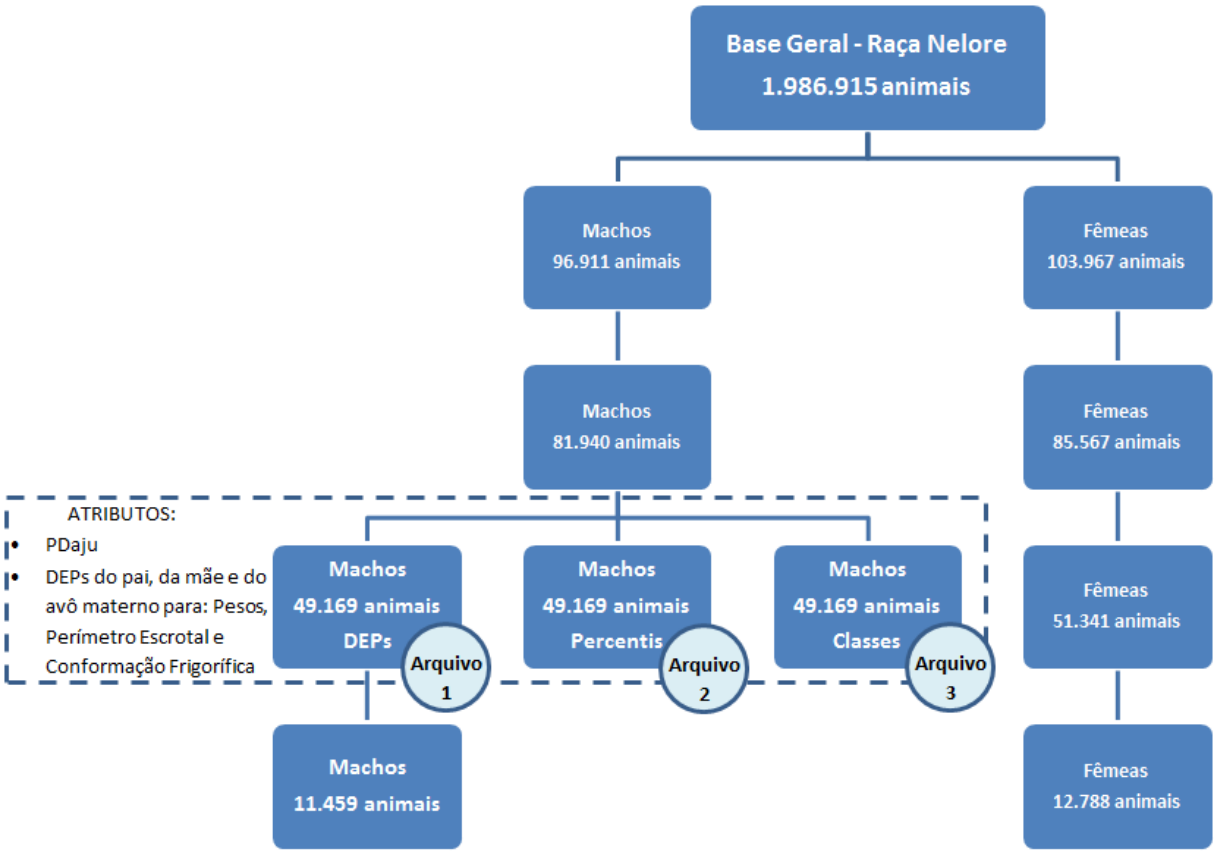


Figura 3.3: Desmembramento do conjunto de dados para experimentos

Rodada 1: Árvore de Decisão								
Classes	Métricas	Baseline	Arquivo 1		Arquivo 2		Arquivo 3	
			Média	DP	Média	DP	Média	DP
ELITE	F1	0,844	**0,919**	0,009	0,917	0,01	0,832	0,016
	Recall	0,831	0,917	0,019	**0,918**	0,02	0,828	0,032
	AUC	-	**0,969**	0,006	0,968	0,005	0,92	0,009
	Precision	0,857	**0,921**	0,011	0,916	0,011	0,838	0,014
	Accuracy	0,881	**0,938**	0,006	0,936	0,007	0,871	0,01
SUPERIOR	F1	0,777	**0,873**	0,009	0,87	0,01	0,754	0,019
	Recall	0,813	**0,878**	0,018	0,876	0,012	0,759	0,024
	AUC	-	**0,939**	0,005	0,937	0,004	0,853	0,017
	Precision	0,745	**0,869**	0,016	0,865	0,02	0,75	0,03
	Accuracy	0,808	**0,895**	0,008	0,892	0,009	0,796	0,018
REGULAR	F1	0,699	0,814	0,014	**0,816**	0,014	0,518	0,04
	Recall	0,665	0,809	0,021	**0,814**	0,021	0,385	0,041
	AUC	-	0,948	0,005	**0,949**	0,006	0,899	0,007
	Precision	0,737	**0,819**	0,018	**0,819**	0,015	0,797	0,023
	Accuracy	0,908	**0,941**	0,004	**0,941**	0,004	0,886	0,006
INFERIOR	F1	0,751	**0,82**	0,024	0,779	0,05	0,62	0,049
	Recall	0,661	**0,80**	0,041	0,717	0,082	0,481	0,059
	AUC	-	**0,951**	0,015	**0,951**	0,011	0,918	0,019
	Precision	0,870	0,842	0,031	0,86	0,031	**0,885**	0,045
	Accuracy	0,981	**0,985**	0,002	0,983	0,003	0,975	0,002

Tabela 3.2: Primeiros resultados para Árvore de Decisão para Arquivo 1 (DEPs), Arquivo 2 (Percentis) e Arquivo 3 (Classes)

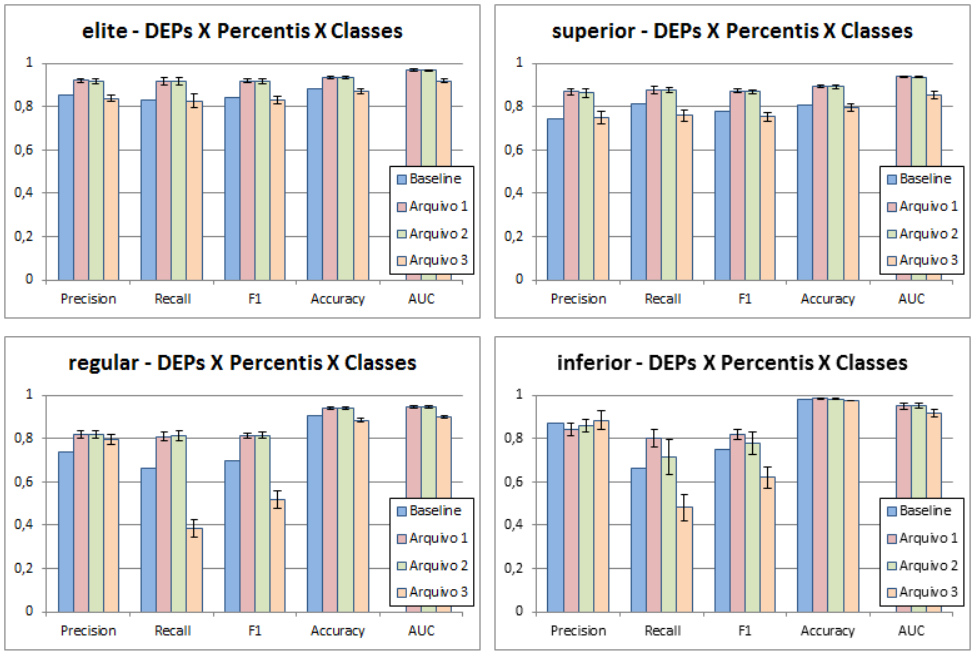


Figura 3.4: Gráficos comparativos para a Rodada 1. No topo das barras estão representados o desvio-padrão (*Standard Deviation*)

no critério `precision` para a Classe `inferior`. Disto pode-se interpretar que o classificador gerado com base no Arquivo 3 consegue rotular corretamente as instâncias sem gerar tantos alarmes-falsos, porém o bom desempenho dele resume-se à Classe `inferior`.

Os desempenhos dos classificadores gerados com base no Arquivo 1 e Arquivo 2 são praticamente iguais, havendo diferença novamente na Classe `inferior`. A taxa de alarmes-falsos é menor para o Arquivo 2 enquanto que o `recall` é melhor no Arquivo 1, conseguindo alcançar 80% de acertos dentro do esperado como positivo para a Classe `inferior`.

Desta primeira rodada, conclui-se que categorizar a DEP pela Classe não foi vantajoso pois os resultados obtidos foram os piores dentre os três conjuntos de estudo e também em comparação com o *`baseline`*.

Em testes no WEKA, constatou-se que os atributos referentes ao avô materno não contribuem para o desempenho dos algoritmos de aprendizado de máquina testados. A presença destes atributos foi uma tentativa de verificar se a relação pai X avô materno seria mais relevante que pai X mãe, já que é prática usual para os produtores analisar um animal inicialmente pelo pai X avô materno. No Apêndice A estão os valores do ganho de informação referente aos atributos do conjunto de dados de machos (weka.attributeSelection. InfoGainAttributeEval).

Na Figura 3.5, novo refinamento nos três conjuntos de dados. Agora retirando os atributos que envolvem DEPs, percentis e Classes do avô materno.

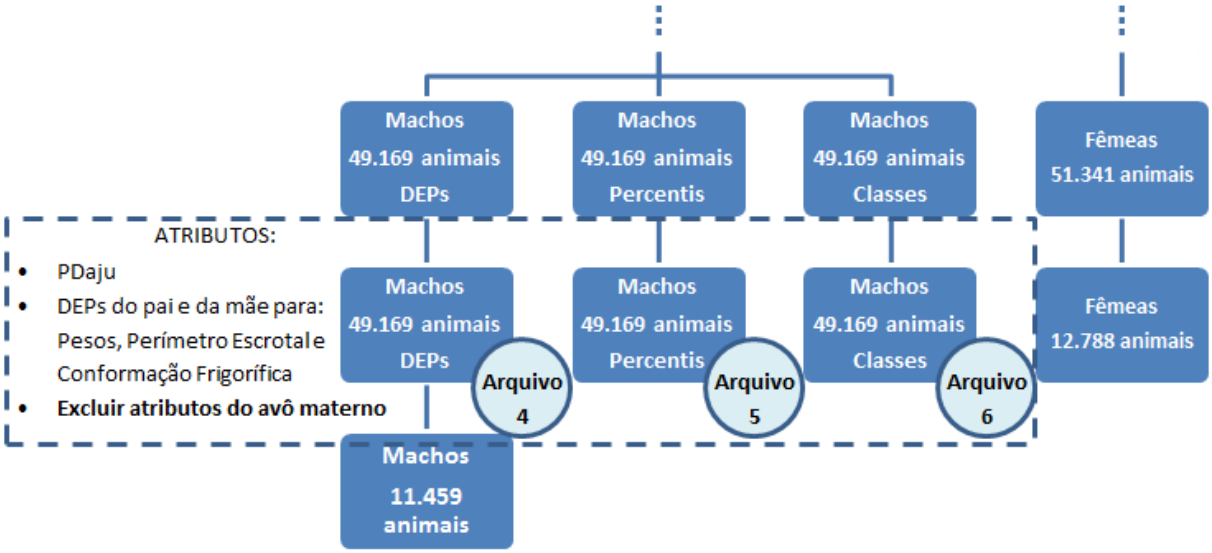


Figura 3.5: Refinamento dos atributos: Eliminação dos referentes ao avô materno

Novas rodadas foram executadas com as amostras 4, 5 e 6 excluindo-se os atributos referentes ao Avô Materno. Basicamente o formato de cada conjunto de dados ficou da seguinte forma: Peso à desmama ajustado, DEPs ou percentis ou classes do pai e da mãe do animal. Os resultados não sofreram alterações significativas pois os atributos excluídos realmente não estavam contribuindo para a construção dos classificadores.

Para verificar se acrescentando atributos de $DEP_{\text{PaiMãe}}$ para as características do animal poderiam melhorar os classificadores, outros conjuntos foram gerados, como mostra a Figura 3.6,

Os conjuntos de dados 7, 8 e 9 serviram de entrada para o algoritmo de árvore de decisão, e os resultados obtidos foram melhores que os atuais, ou seja, informar as possíveis DEPs do animal contribui para um classificador mais eficiente.

O classificador que usa o Arquivo 7, que informa as DEPs originais, foi o melhor entre os três. A diferença de desempenho entre os classificadores que utilizam as amostras 7 e 8 é praticamente nula, havendo leve vantagem para o primeiro.

Na Tabela 3.3, os melhores resultados até então. De todos estes experimentos, conclui-se que o melhor conjunto de dados para qualquer classificador deverá conter as DEPs originais, sem categorização. Se necessário for, é

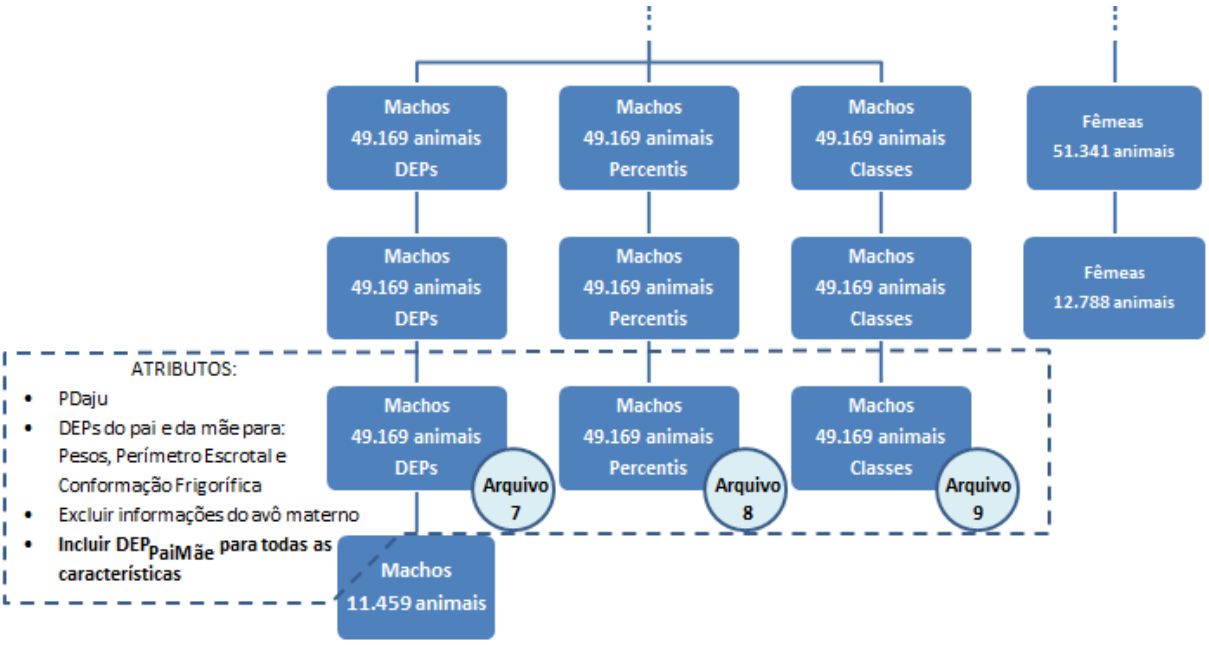


Figura 3.6: Acréscimo dos atributos de $DEP_{PaiMãe}$ para as características do animal

possível aplicar a categorização via percentil, porém sob o risco de sofrer pequena desvantagem. Portanto, o conjunto de dados definido para os próximos experimentos deverá ser o Arquivo 7.

Rodada 4: Árvore de Decisão								
Classes	Métricas	Baseline	Arquivo 7		Arquivo 8		Arquivo 9	
			Média	DP	Média	DP	Média	DP
ELITE	F1	0,844	0,926	0,008	**0,927**	0,007	0,876	0,009
	Recall	0,831	0,925	0,014	**0,927**	0,014	0,875	0,02
	AUC	-	**0,972**	0,005	**0,972**	0,005	0,945	0,005
	Precision	0,857	**0,928**	0,007	0,926	0,007	0,877	0,009
	Accuracy	0,881	**0,943**	0,006	**0,943**	0,005	0,904	0,006
SUPERIOR	F1	0,777	**0,887**	0,007	0,885	0,005	0,814	0,008
	Recall	0,813	**0,89**	0,009	0,886	0,008	0,818	0,012
	AUC	-	**0,947**	0,005	**0,947**	0,003	0,898	0,006
	Precision	0,745	**0,885**	0,012	**0,885**	0,011	0,809	0,015
	Accuracy	0,808	**0,907**	0,006	0,906	0,004	0,846	0,007
REGULAR	F1	0,699	0,84	0,012	**0,841**	0,012	0,66	0,033
	Recall	0,665	0,839	0,018	**0,842**	0,019	0,551	0,049
	AUC	-	0,955	0,006	**0,957**	0,005	0,923	0,004
	Precision	0,737	**0,842**	0,014	0,84	0,013	0,826	0,027
	Accuracy	0,908	**0,949**	0,004	**0,949**	0,004	0,909	0,006
INFERIOR	F1	0,751	**0,849**	0,018	0,842	0,03	0,724	0,031
	Recall	0,661	0,826	0,037	**0,832**	0,038	0,612	0,042
	AUC	-	0,955	0,013	**0,957**	0,012	0,937	0,011
	Precision	0,870	0,875	0,02	0,854	0,035	**0,892**	0,045
	Accuracy	0,981	**0,988**	0,001	0,987	0,002	0,98	0,002

Tabela 3.3: Resultados para Árvore de Decisão para Arquivo 7 (DEPs), Arquivo 8 (Percentis) e Arquivo 9 (Classes)

Na Figura 3.7, os gráficos para comparar os desempenhos entre os três classificadores gerados.

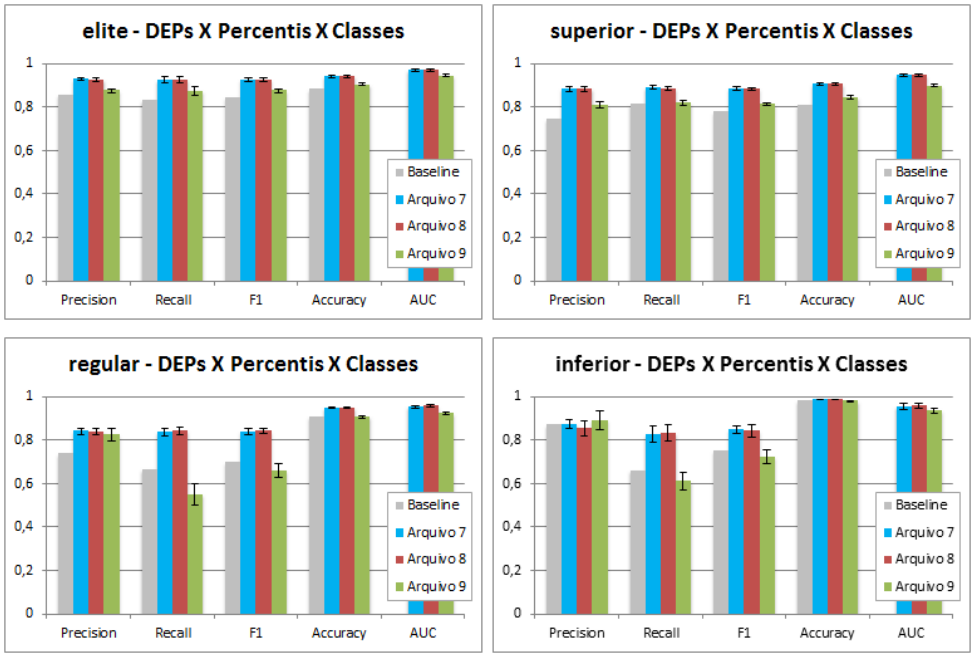


Figura 3.7: Gráficos comparativos para a Rodada 4. No topo das barras estão representados o desvio-padrão (*Standard Deviation*)

Nas Figuras 3.8, 3.9 e 3.10, os gráficos para comparar os desempenhos entre as rodadas 1 e 4, para cada um dos três tipos de conjuntos de dados analisados, DEPs, Percentis e Classes.

Encerrando esta primeira fase, conclui-se que as informações de curral são basicamente: animal, pai, mãe e peso coletado. Destes dados de entrada, serão recuperadas as DEPs, percentis e Classes correspondentes aos pais do animal e o peso coletado será ajustado para o grupo contemporâneo do animal. Os dados do avô materno não contribuem para o ganho de informação e portanto, foram excluídos da Base Selecionada.

Acrescentar $DEP_{\text{PaiMãe}}$ do animal contribuiu para melhores resultados, e são atributos que podem ser facilmente calculados se houver DEPs dos pais. Todos os atributos referentes a percentil e Classe também foram excluídos pois os resultados com estes atributos não foram tão satisfatórios quanto os dos conjuntos com somente DEPs.

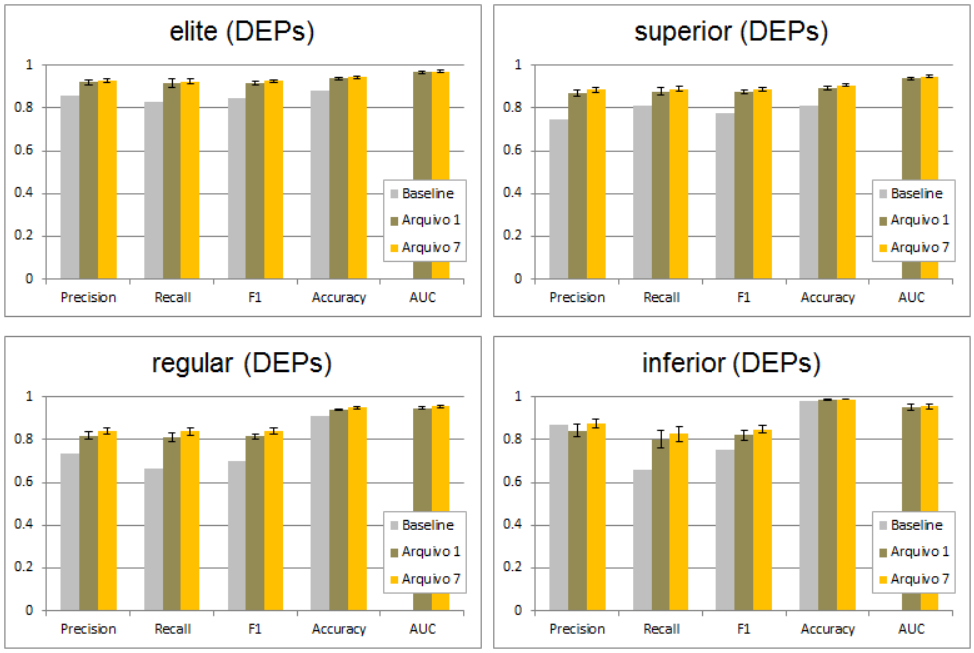


Figura 3.8: Gráficos comparativos para os conjuntos de dados de DEPs da Rodada 1 (Arquivo 1) e da Rodada 4 (Arquivo 7). No topo das barras estão representados o desvio-padrão (*Standard Deviation*)

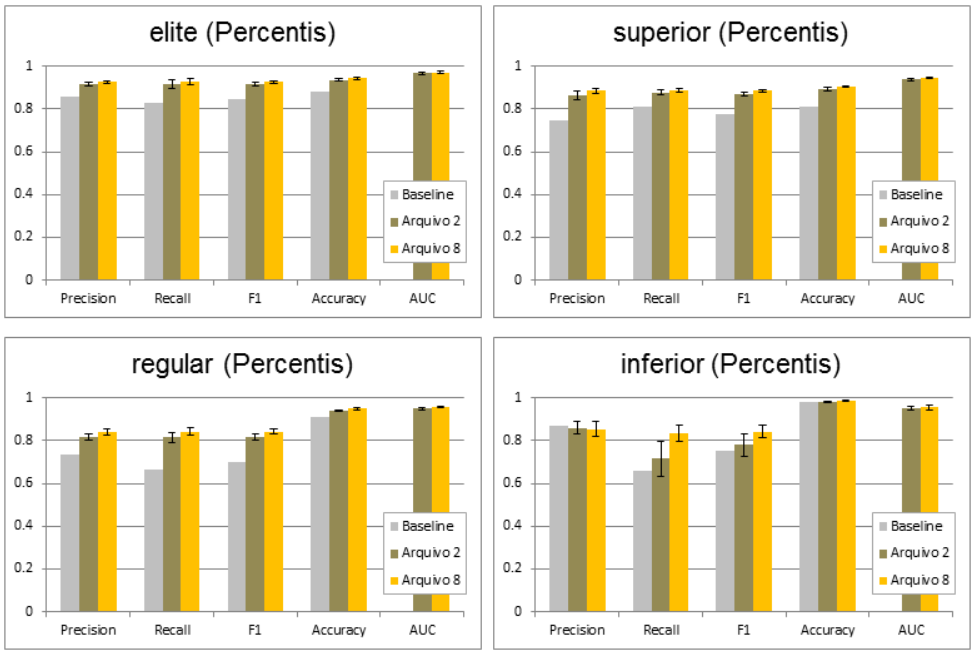


Figura 3.9: Gráficos comparativos para os conjuntos de dados de Percentis da Rodada 1 (Arquivo 2) e da Rodada 4 (Arquivo 8). No topo das barras estão representados o desvio-padrão (*Standard Deviation*)

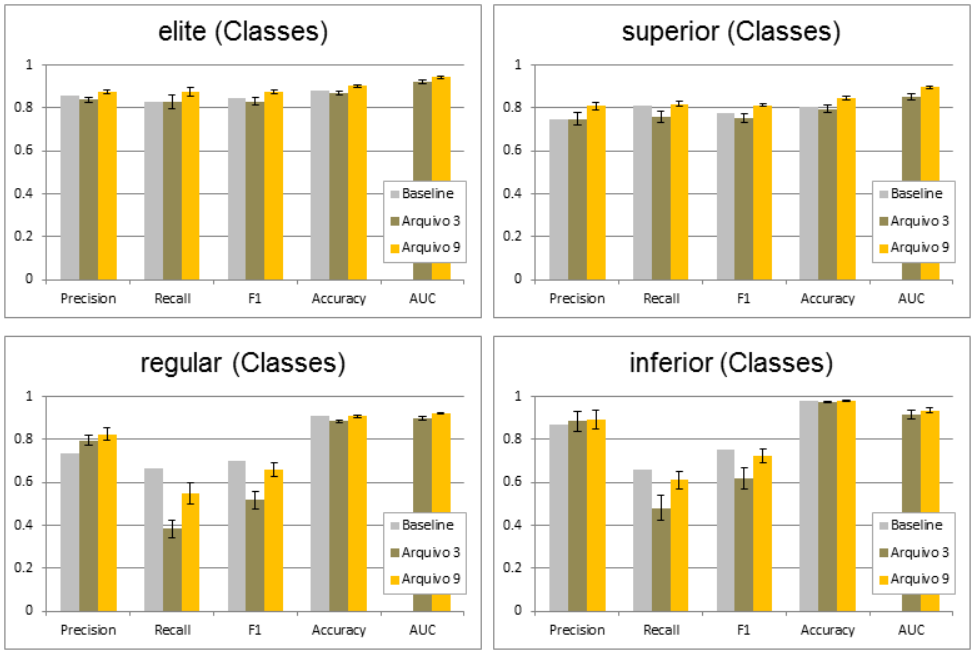


Figura 3.10: Gráficos comparativos para os conjuntos de dados de Classes da Rodada 1 (Arquivo 3) e da Rodada 4 (Arquivo 9). No topo das barras estão representados o desvio-padrão (*Standard Deviation*)

3.2 Entre os algoritmos de Árvore de Decisão, K-Vizinhos mais Próximos e *Naive Bayes*, qual o indutor mais adequado ao problema?

O presente trabalho tem foco em Árvores de Decisão (*Decision Trees* - AD). Entretanto, para uma análise comparativa com outros algoritmos clássicos de Aprendizado de Máquina, nesta seção o algoritmo de AD é comparado com K-Vizinhos mais Próximos (KNN) e *Naive Bayes*(NB). Os algoritmos testados nesta seção são implementações da biblioteca SKLEARN[3].

O parâmetro k do algoritmo KNN foi ajustado utilizando *Random Sampling* (Bergstra and Bengio, 2012) com 30 iterações e com o seguinte intervalo de valores 1-50. O Naive Bayes foi utilizado com os parâmetros padrão.

Como um dos enfoques neste trabalho é poder interpretar os resultados utilizando a perpectiva de análise ROC, o conjunto de dados que possui quatro classes (elite,superior,regular,inferior) foi convertido em quatro problemas de duas classes utilizando a abordagem um contra todos.

Nas Figuras 3.11, 3.12, 3.13 e 3.14 são apresentadas as métricas de avalição precision, recall, f1-measure, accuracy e AUC. O baseline representa a

[3]http://scikit-learn.org/

abordagem utilizada pelos usuários do GENEPLUS que consiste em aplicar limiares na $DEP_{PaiMãe}$ para a construção das classes. Note que o baseline não tem escore por Classe e por isso, neste momento, não foi possível obter o AUC para baseline.

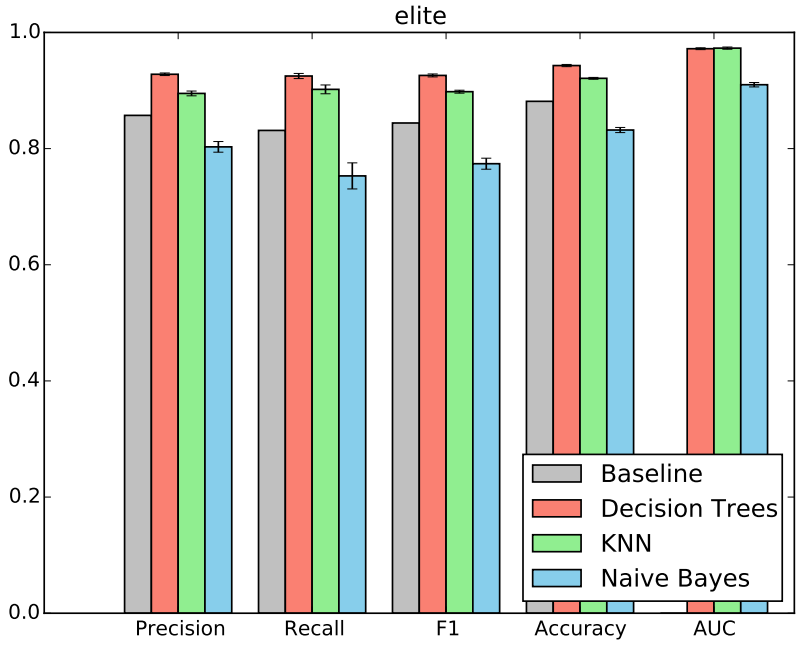


Figura 3.11: Comparativo de indutores para Classe elite. No topo das barras estão representados o erro padrão (*Standard Error*)

Na análise comparativa dos métodos, pode-se notar que o algoritmo de *Decision Trees* apresenta os melhores resultados. Entretanto, cabe ressaltar que o algoritmo de AD, por definição já retorna os valores calibrados em termos de probabilidade. Observe que como AUC não necessita de ajuste de limiar de classificação, AD e KNN, possui resultados muito próximos para as elite e regular.

Ter o escore similar a uma probabilidade faz com que AD tenha alguma vantagem perante aos outros algoritmos quando avaliados em termos de recall e precision (consequentemente em f1-measure). Entrentando para remover esta provável vantagem, os resultados de KNN e NB foram calibrados utilizando regressão isotônica.

Deste modo, ao comparar AD,KNN e NB, o algoritmo que mostrou desempenho mais favorável foi AD.

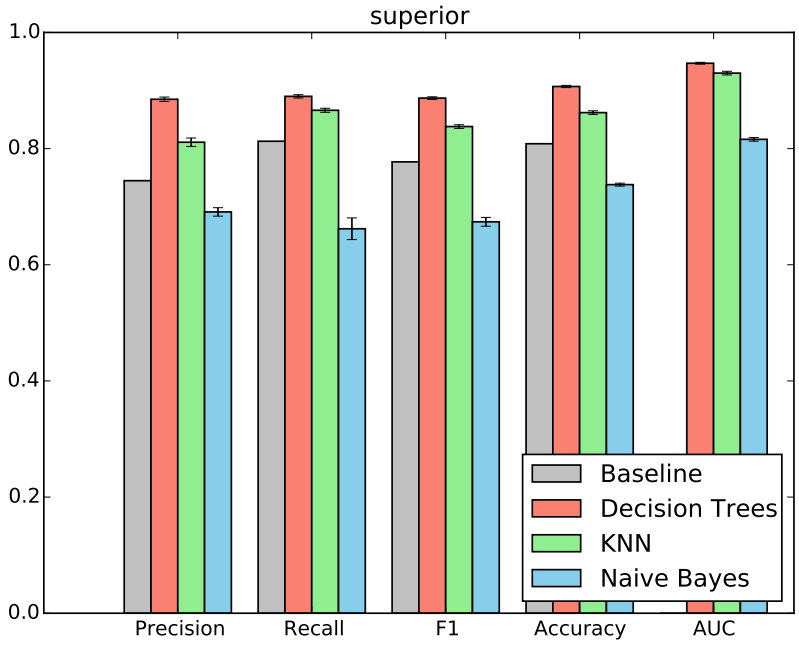


Figura 3.12: Comparativo de indutores para classe `superior`. No topo das barras estão representados o erro padrão (*Standard Error*)

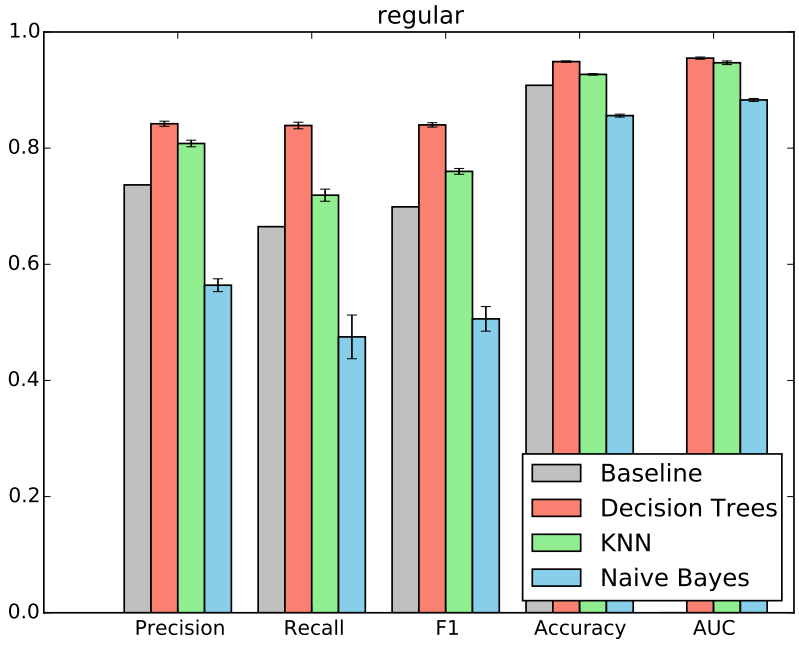


Figura 3.13: Comparativo de indutores para classe `regular`. No topo das barras estão representados o erro padrão (*Standard Error*)

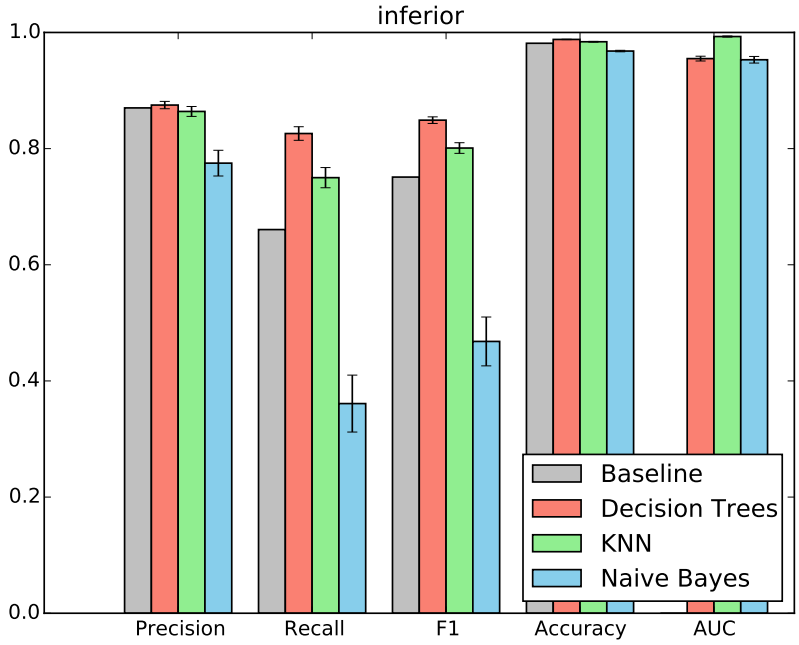


Figura 3.14: Comparativo de indutores para classe `inferior`. No topo das barras estão representados o erro padrão (*Standard Error*)

3.3 É possível melhorar os resultados construindo novos atributos?

Observando que nos procedimentos atuais de obtenção da Classe das DEPs dos animais não há valores para *Area Under ROC* - `AUC`, resolveu-se testar métodos de se obter `escores` pelos quais curvas ROC pudessem ser construídas.

Uma tentativa foi revisar os conceitos utilizados na geração dos percentis, já que informam justamente o *ranking* das DEPs. Na su

3.3.1 Percentil, `escore` e Classes

A Distribuição Normal

A distribuição normal, também conhecida como distribuição Gaussiana, é de grande importância para estatística, sendo muito utilizada para descrever fenômenos naturais e sociais que são representados por variáveis aleatórias contínuas.

Por esta distribuição, probabilidades são associadas a intervalos de números

reais (Morettin, 1981), ou seja, suponha x uma variável aleatória. A probabilidade de x estar em um intervalo $[a, b]$, é calculada como sendo a área entre a e b, sob uma curva, chamada função de densidade de x. A distribuição normal é caracterizada por esta função densidade (Morettin, 1981), dada pela Equação 3.1.

$$f(x \mid \mu, \sigma^2) = \frac{1}{\sigma\sqrt{2\pi}} exp\left(-\frac{(x-\mu)^2}{2\sigma^2}\right) \tag{3.1}$$

Onde μ é a média ($-\infty < \mu < +\infty$) e σ é o desvio padrão ($\sigma > 0$).

O teorema do limite central estabelece que se a variância populacional é finita, a distribuição da média amostral é aproximadamente normal se o tamanho amostral é grande. Isto pode ser observado na Figura 3.15, onde é apresentado um histograma da DEP do peso à desmama de toda a população da qual foi extraído o conjunto de treinamentos deste estudo. O histograma tem grande similaridade com uma distribuição Normal.

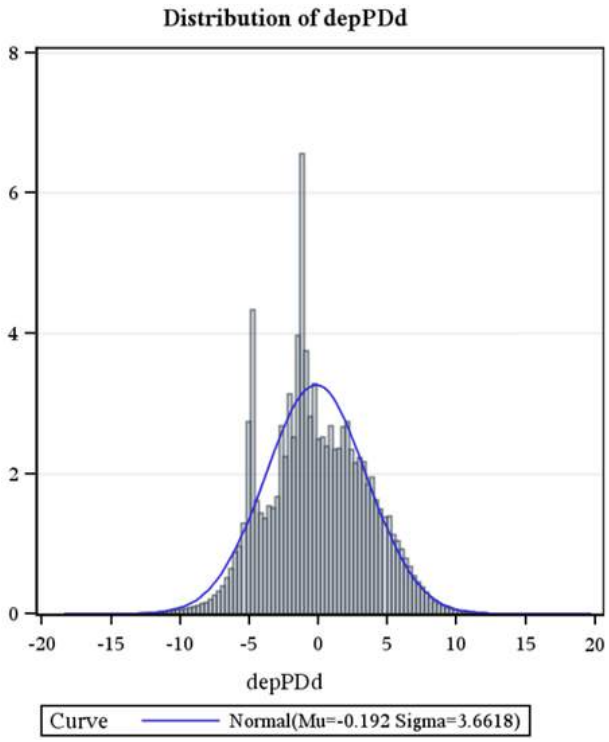


Figura 3.15: Histograma e curva normal da DEP para peso à desmama

Observações sobre as curvas normais, segundo (Morettin, 1981):

- Para um mesmo μ, quanto maior o σ, mais achatada é a curva,
- As curvas são simétricas em relação ao ponto μ, conforme Figura 3.16,

- Praticamente toda a área está concentrada entre os pontos $\mu \pm 3.\sigma$.

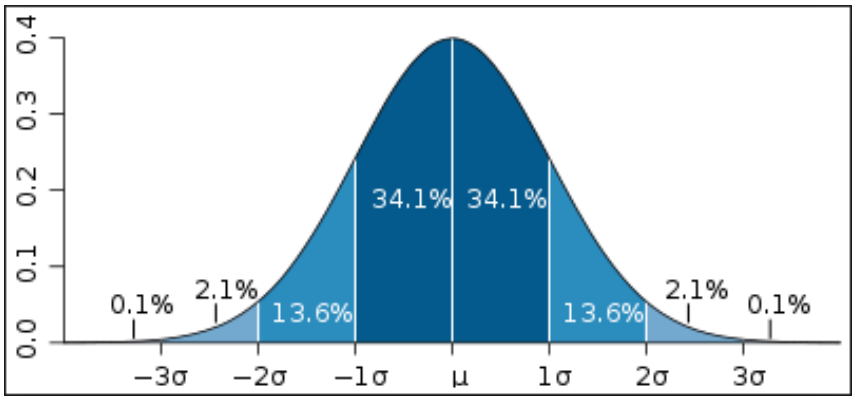


Figura 3.16: Curva normal e desvio-padrão
Fonte: Wikipedia[4]

Quando $\mu = 0$ e $\sigma = 1$, a distribuição Normal é chamada *Normal Reduzida* ou *Normal Padrão* (Morettin, 1981) e possui a propriedade de que a área total sob a curva é igual a um. Esta propriedade é interessante pois as probalidades para cada intervalo $[a, b]$ já estão calculadas em uma tabela fixa, chamada *Tabela da Distribuição Acumulada da Normal Padrão*, que pode ser vista no Apêndice F.

Toda distribuição Normal pode ser transformada em uma Normal Reduzida. Cada DEP da curva mostrada na Figura 3.15 pode ser transformado em um ponto na curva Normal Reduzida. Este mapeamento é dado aplicando-se a Equação 3.2.

$$escore\ z = \frac{x - \mu}{\sigma} \tag{3.2}$$

Onde x é o valor da DEP, μ é a média da DEP de toda a população, σ é o desvio-padrão correspondente, e o `escore z` será a região aproximada em que x estará na Normal Reduzida.

Os valores do `escore z` representam o número de desvios-padrão (σ) que separam uma variável aleatória x da média(μ).

O Percentil

A partir deste `escore z` gerado, o mapeamento da DEP na curva de distribuição acumulada para uma região na curva Normal Reduzida será feito. Cada `escore z` terá uma porcentagem de área sob a curva Normal Reduzida encontrada na Tabela da Distribuição Acumulada da Normal Padrão - Apêndice F. Por esta porcentagem de área é que calcula-se o percentil para cada DEP.

O percentil, dentro da concepção de indicador de uma DEP, foi criado para evidenciar posicionamento desta DEP dentro de uma determinada população utilizada para gerar a DEP. Simplesmente, um percentil de 10% significa que a DEP está entre 10% melhores da população.

Exemplo 13 Por exemplo, suponha que um animal alcançou DEP para o peso à desmama de 1,472 kg. Para qualquer característica examinada em uma população, dentro da amplitude possível, alguns animais serão muito bons, outros muito ruins mas a maioria estará próxima da média (Daly, 1977).

Na Figura 3.17, à esquerda está o histograma para a DEP do peso à desmama de toda a população avaliada, 1.986.915 animais. Então a questão é saber em que ponto da curva Normal Reduzida estaria posicionado o animal cuja DEP vale 1,472.

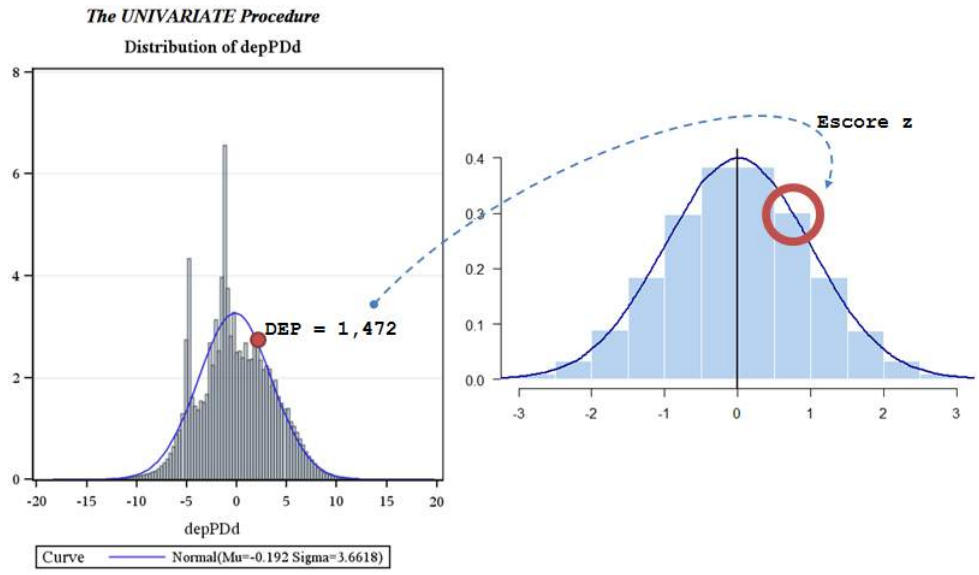
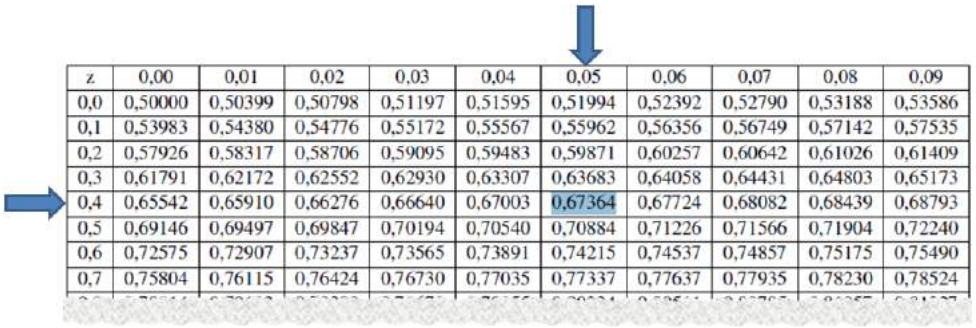


Figura 3.17: Mapeamento do percentil

Para este caso, $\mu = -0,1921789$ e $\sigma = 3,66184467$. Então o valor do `escore z` correspondente a esta DEP será:

$$escore\ z = \frac{DEP_{PD} - \mu_{PD}}{\sigma_{PD}} = \frac{1,4720 - -0.1922}{3.6618} = 0.4545$$

Então para a DEP = 1,472 e `escore z` = 0,454, a área acumulada é encontrada consultando-se a Tabela 3.18 da Distribuição Acumulada da Normal Padrão. Procura-se as décimas de `escore z` na coluna da esquerda (0,4) e as centésimas na linha de cima da tabela (0,05):



z	0,00	0,01	0,02	0,03	0,04	0,05	0,06	0,07	0,08	0,09
0,0	0,50000	0,50399	0,50798	0,51197	0,51595	0,51994	0,52392	0,52790	0,53188	0,53586
0,1	0,53983	0,54380	0,54776	0,55172	0,55567	0,55962	0,56356	0,56749	0,57142	0,57535
0,2	0,57926	0,58317	0,58706	0,59095	0,59483	0,59871	0,60257	0,60642	0,61026	0,61409
0,3	0,61791	0,62172	0,62552	0,62930	0,63307	0,63683	0,64058	0,64431	0,64803	0,65173
0,4	0,65542	0,65910	0,66276	0,66640	0,67003	0,67364	0,67724	0,68082	0,68439	0,68793
0,5	0,69146	0,69497	0,69847	0,70194	0,70540	0,70884	0,71226	0,71566	0,71904	0,72240
0,6	0,72575	0,72907	0,73237	0,73565	0,73891	0,74215	0,74537	0,74857	0,75175	0,75490
0,7	0,75804	0,76115	0,76424	0,76730	0,77035	0,77337	0,77637	0,77935	0,78230	0,78524

Figura 3.18: Tabela da distribuição acumulada da normal padrão, exemplo

Para o valor 0,67364 encontrado, significa que a área acumulada ou a probabilidade de $(\texttt{escore z} < 0,45) = 0,67364$ e que o animal está entre os $1 - 0,67364 = 33\%$ melhores da população.

Observando a Figura 3.19, lembrar que:

1. Para $\texttt{escore z} \to -\infty$ a área acumulada está próxima de 0,00;
2. Se $\texttt{escore z}$ cresce, a área acumulada também cresce;
3. Se $\texttt{escore z} = 0$, a área acumulada vale 0,50;
4. para $\texttt{escore z} \to +\infty$ a área acumulada está próxima de 1,00.

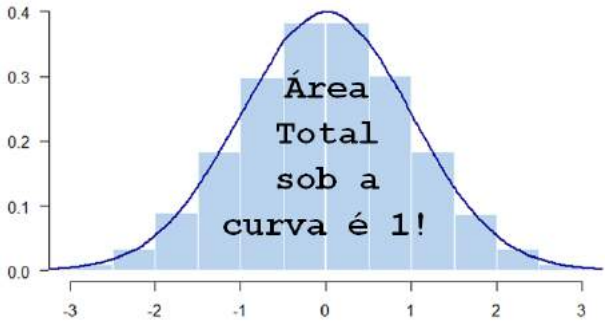


Figura 3.19: Propriedades da Normal Padrão

Por causa deste mapeamento, qualquer população avaliada pelo Programa Geneplus pode ter percentis para cada DEP. Os percentis indicarão aproximadamente onde, na curva Normal Reduzida, o animal está. O percentil portanto, não é dado pelo número de observações, nem é dado por alguma ordenação ou contagem, mas sim pela probabilidade de estar em determinada região da Normal Reduzida.

Classes no Sumário

A Figura 3.20 mostra uma curva normal subdividida em 8 regiões. Cada região tem a sua porcentagem de área por desvio-padrão. Se considerá-la como uma curva Normal Reduzida, as porcentagens das áreas correspondem à área propriamente dita e a área acumulada é o somatório das áreas à esquerda da curva até chegar no `escore z` desejado. Se `escore z` $= 3$, o somatório será aproximadamente 1.

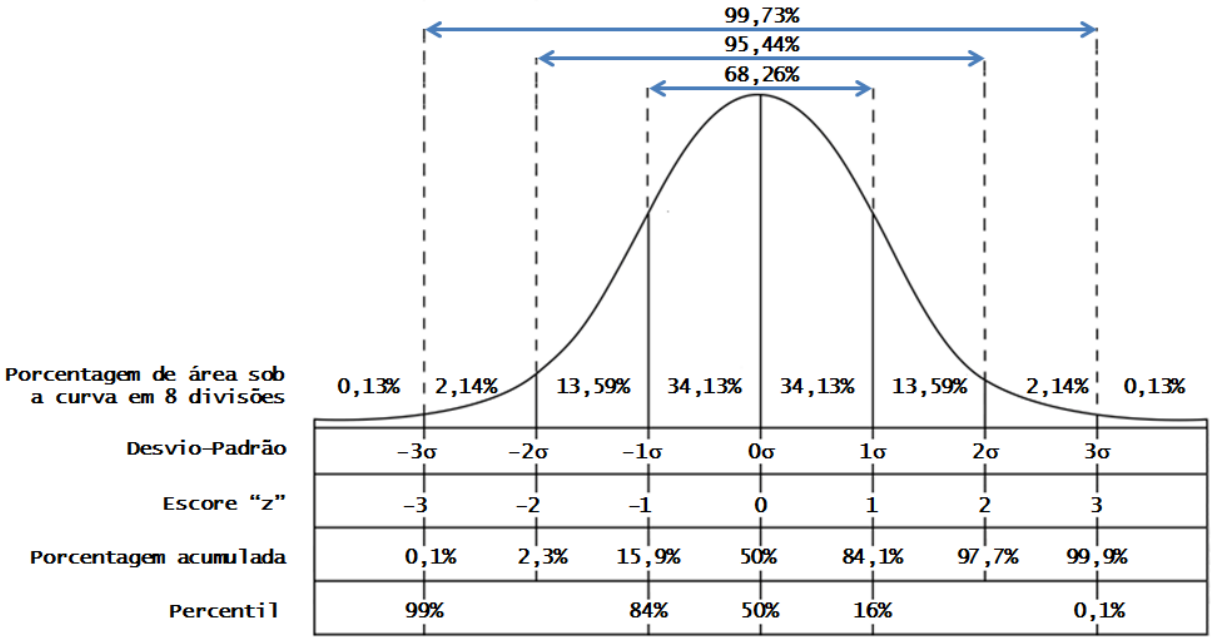


Figura 3.20: Curva normal e intervalos
Fontes: Quantile, Portal Action[5]

A porcentagem acumulada é o inverso do percentil utilizado como *ranking* para as DEPs. Para dar a idéia de que os melhores animais estão à direita da curva, o percentil é invertido, e quanto menor, melhor. Por exemplo, um animal com percentil 1% é melhor que um animal de percentil 67%.

Para facilitar a percepção de quão bom é um animal, ao invés de se utilizar percentis, adotou-se também um outro conceito, chamado "Classe". A classe define a classificação do animal em função do percentil (Nobre et al., 2013b), ou seja, ela categoriza as DEPs pelos seus percentis. Na Figura 3.21 a disposição das classes conforme o intervalo de percentis.

A Figura 3.22 mostra os limiares para os percentis e os intervalos destes que determinam a classe do animal. Pode-se dizer então, que um animal acima de 1 desvio-padrão da média, é um animal Elite.

Exemplo 14 Um animal com `escore z` até 3,0902 corresponde a

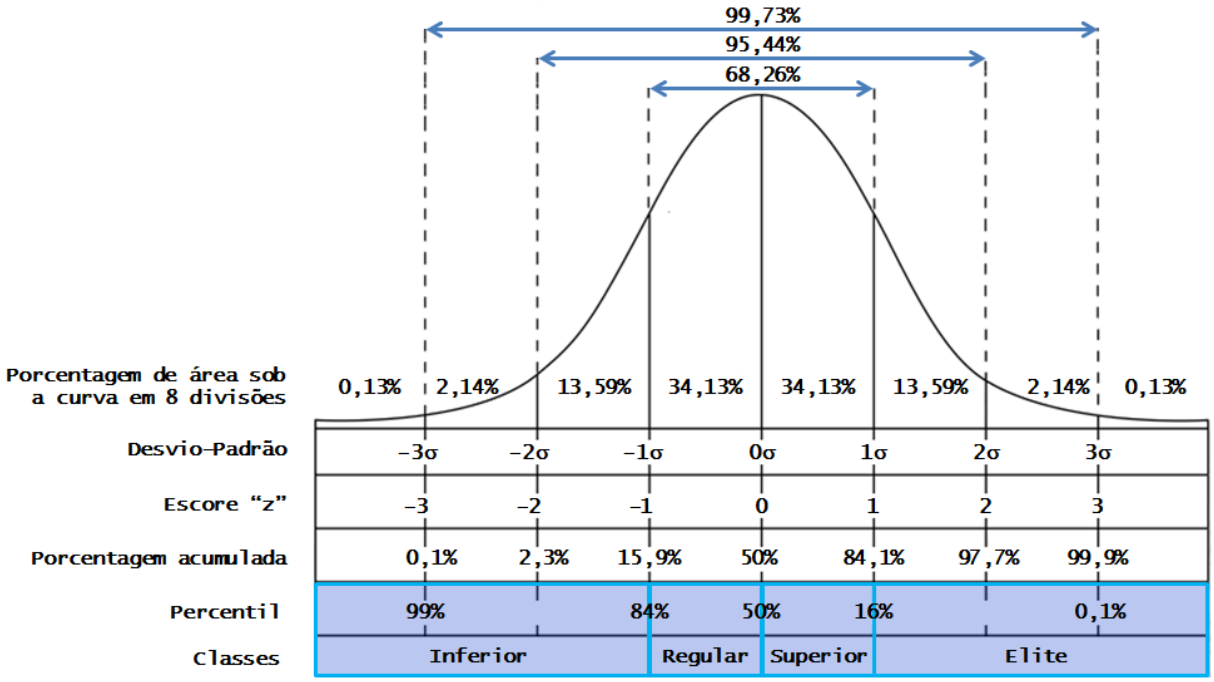


Figura 3.21: Curva normal e classes

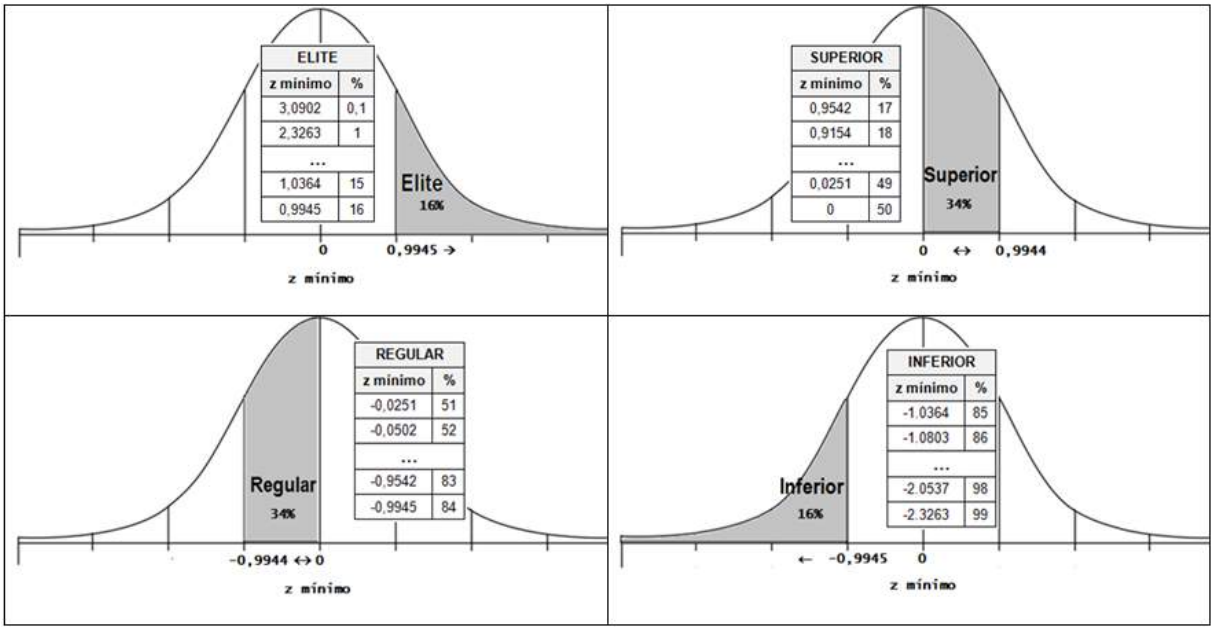


Figura 3.22: Limiares para as classes

um percentil de 0,1% e pertence à Classe `elite`. Um animal com `escore z` entre 3,0901 e 2,3263 corresponde a um percentil de 1% e também pertence à Classe `elite`. Um animal com `escore z` entre 0,0250 e 0 tem percentil de 50% e Classe `superior`.

Sendo assim, pelos valores mínimos do `escore z` temos a função $Classe$:

$escore\ z \rightarrow \{elite,\ superior,\ regular,\ inferior\}$, onde:

$$Classe(\texttt{escore z}) = \begin{cases} elite, & se\ \texttt{escore z} \geq 0,9945 \\ superior, & se\ 0,0 \leq \texttt{escore z} < 0,9945 \\ regular, & se\ -0,9945 \leq \texttt{escore z} < 0,0 \\ inferior, & se\ \texttt{escore z} < -0,9945 \end{cases}$$

3.3.2 O Conversor

O conversor consiste em utilizar o `escore z` calculado com a DEP como variável aleatória para derivar `escores` para cada uma das classes `elite`, `superior`, `regular` e `inferior`, de modo que os valores numéricos obtidos representem que quanto mais alto, maior a chance do animal pertencer a cada uma das Classes.

Para os extremos `elite` e `inferior` a resposta é trivial pois basta utilizarmos o valor numérico do próprio `escore z` e do `escore z` invertido, respectivamente. Entretanto, para obtermos `escores` para as Classes intermediárias, `superior` e `regular`, os valores de `escore z` precisam ser transformados.

A maneira mais simples e que mostrou eficácia foi usar constantes para colocar o intervalo de `escore z` que indicam a Classe `superior` à frente das demais, e do mesmo modo para a Classe `inferior`.

Na Figura 3.23, verifica-se que o `escore z` original pode ser usado como `escore` para a Classe `elite`, uma vez que ao ordenar pelo `escore z` decrescentemente, os maiores valores serão os mais próximos do positivo (ser da Classe `elite`).

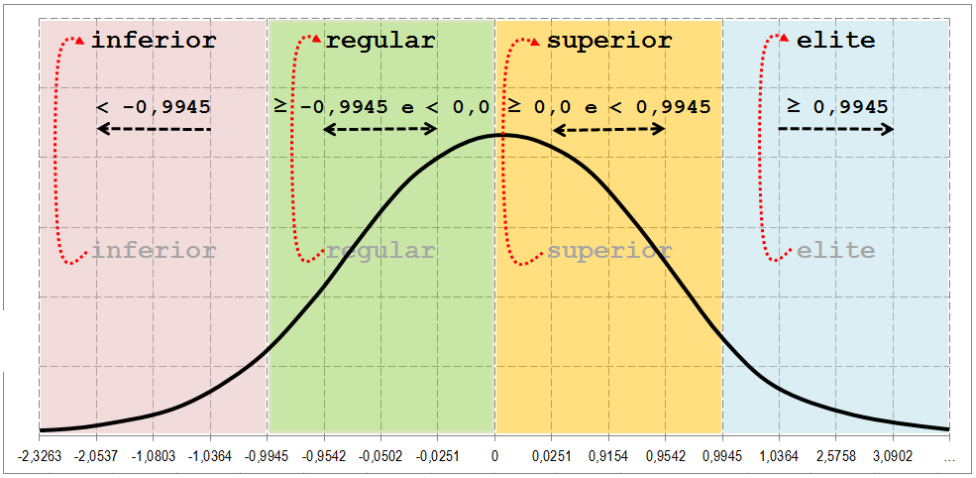


Figura 3.23: Conversor para EscoreE

Na Figura 3.24, pode-se observar que o `escore z` invertido também permite ser usado como `escore` para a Classe `inferior`, pois multiplicando-se por (-1) o `escore z`, os maiores valores serão os mais próximos do positivo (ser da Classe `inferior`).

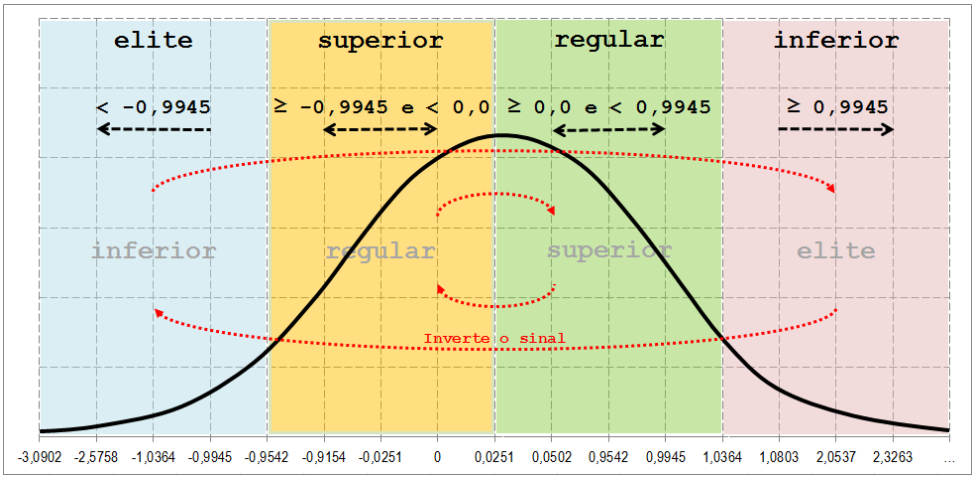


Figura 3.24: Conversor para EscoreI

A Figura 3.25 mostra uma simples maneira de colocar a região da curva referente à Classe `superior` na frente das demais Classes, sem que as demais percam sua ordem. O `escore z` da faixa entre $0,0 \leqslant$ `escore z` $< 0,9945$ é somado a uma constante (10) para que "salte" à frente da Classe `elite`. O intervalo de `escore z` referente à Classe `elite`, se mantém como o original. O intervalo de `escore z` referente às Classes `regular` e `inferior`, são colocadas mais para trás. A constante somada ao `escore z` foi tomada ao acaso, no valor 10 para que o intervalo da área referente à Classe `superior` avançasse à frente das demais. O mesmo vale para a subtração da constante 10 das Classes `regular` e `inferior`, a constante foi tomada ao acaso para afastá-la da área pertencente à Classe `elite`.

E enfim, a Figura 3.26 retrata como a região da curva referente à Classe `regular` é colocada na frente das demais Classes, sem que percam sua ordem. O `escore z` da faixa entre $-0,9945 \leqslant$ `escore z` $< 0,0$ é somado a uma constante (20, também tomada ao acaso) para que "salte" à frente da Classe `elite`. Os demais valores de `escore z` se mantêm como o original.

Com estes quatro conversores simples, foi possível aplicar um método de discretização supervisionada sobre o `escore z`, procurando por limiares ótimos, que possam melhorar as métricas de avaliação de classificação.

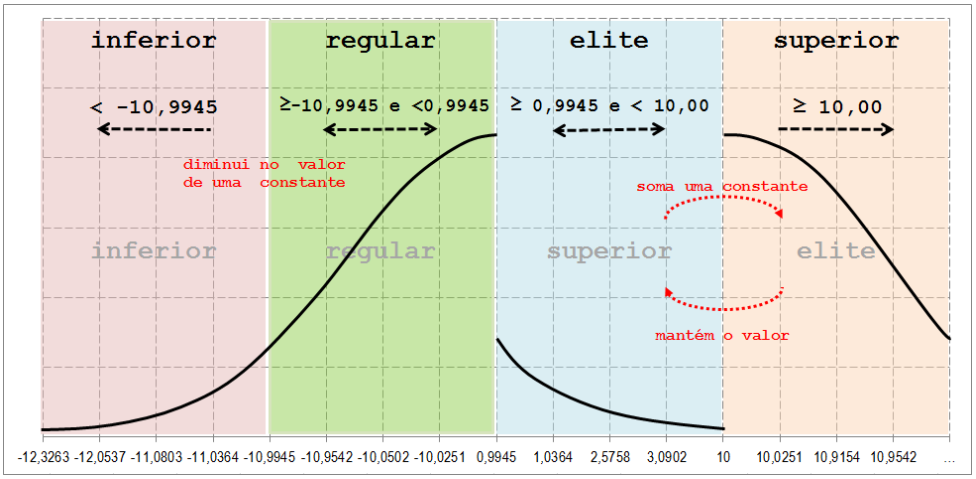


Figura 3.25: Conversor para EscoreS

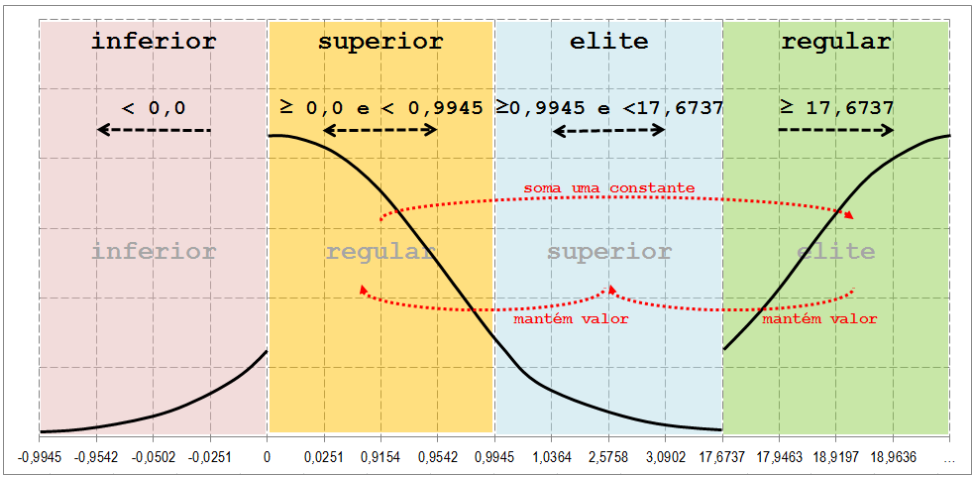


Figura 3.26: Conversor para EscoreR

3.3.3 Curvas ROC para o `baseline`

Utilizando a $\text{DEP}_{\text{PaiMãe}}$ para o Peso à Desmama, um novo `escore z` foi criado com a Fórmula 2.26 do $Percentil_{\text{CALC}}$. Então quatro `escores` foram criados para que, calibrados, pudessem gerar os gráficos ROC apresentados na Figura 3.27.

Esta estratégia possibilitou construir o gráfico ROC para cada uma das Classes, e trouxe mais quatro atributos, os `escores`, para o conjunto de treinamento.

3.3.4 Os Testes com os Novos Atributos

Um primeiro teste utilizou como `escore` a $DEP_{\text{PaiMãe}}$ do animal, e com calibração isotônica. Não atendeu à necessidade pois somente a Classe `elite`,

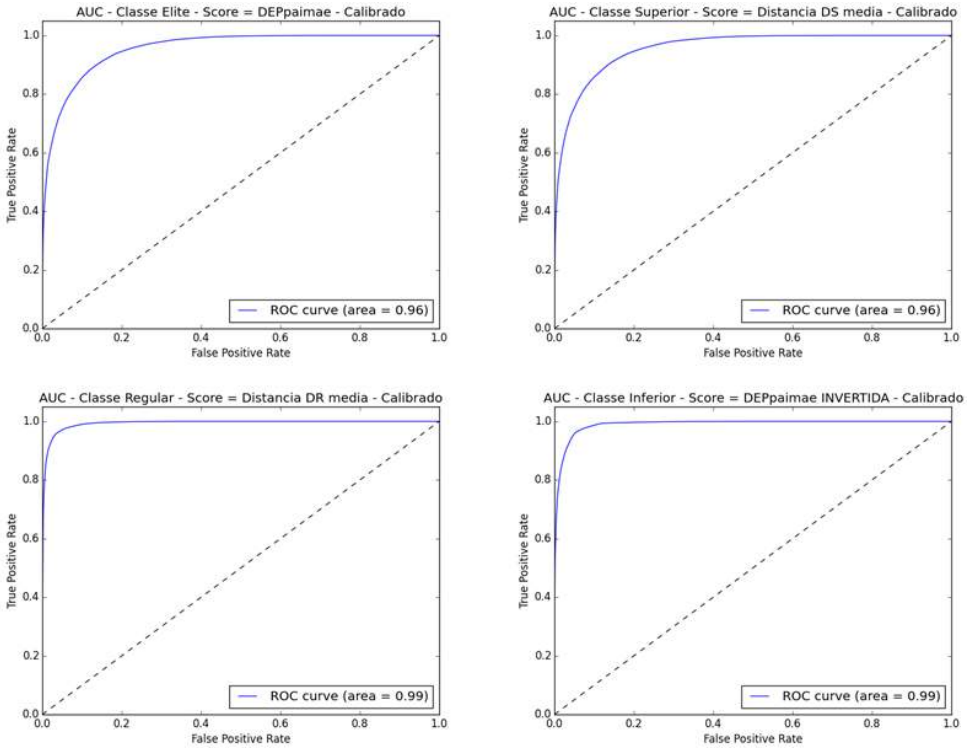


Figura 3.27: Gráficos ROC para as classes `elite`, `superior`, `regular` e `inferior` utilizando `escores`

por ser a que possui as maiores DEPs e, consequentemente estar com o `escore` mais perto do positivo para a sua rodada, apresentou uma `AUC` de 0,96. As demais, por estarem mais distantes do positivo para cada classe, foram demonstrando `AUC` cada vez menores.

Com o acréscimo de mais quatro atributos, os `escores` utilizados para calcular a área abaixo da curva ROC para o `baseline`, um classificador mais eficiente do que o definido na Rodada 4 foi encontrado.

Assim, um novo conjunto de treinamento foi gerado, a Base Melhorada, sem necessidade da $DEP_{PaiMãe}$, e acrescentando-se mais estes quatro atributos de `escore`. Somente os animais que tinham peso à desmama e não tinham o peso ao sobreano permaneceram, como pode ser visualizado na Figura 3.28.

Ao submeter a Base Melhorada ao classificador por árvore de decisão, os resultados obtidos foram expressivos conforme pode ser observado na Tabela 3.4.

Na Figura 3.29, um comparativo entre os dois melhores classificadores até então, o obtido na Rodada 4 com o Arquivo 7, e o obtido agora. Como se pode ver, em comparação com o `baseline`, todas as Classes tiveram métricas melhores. Para a Classe `elite`, os resultados comparativos com a Rodada 4

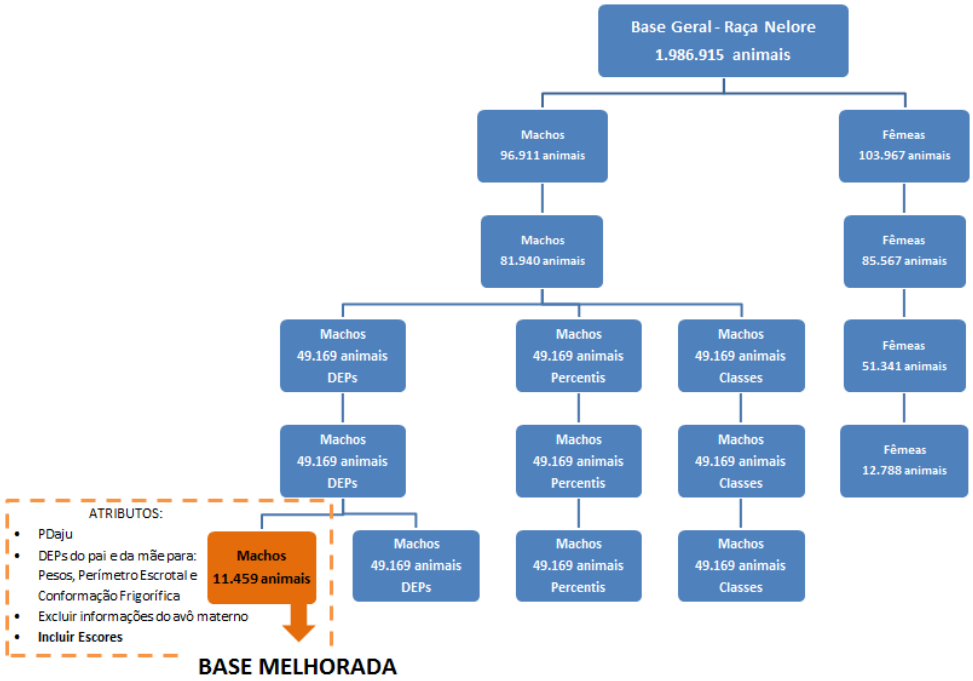


Figura 3.28: Base Melhorada e seus atributos

Rodada 7: Árvore de Decisão + Escores												
Métricas	ELITE			SUPERIOR			REGULAR			INFERIOR		
	Baseline	Média	DP	Baseline	Média	DP	Baseline	Média	DP	Baseline	Média	DP
F1	0,844	0,866	0,303	0,777	1,000	0,000	0,699	1,000	0,000	0,751	0,902	0,290
Recall	0,831	0,901	0,298	0,813	1,000	0,000	0,665	1,000	0,000	0,661	0,902	0,295
AUC	-	0,941	0,121	-	1,000	0,000	-	1,000	0,000	-	0,951	0,148
Precision	0,857	0,948	0,155	0,745	1,000	0,000	0,737	1,000	0,000	0,870	0,997	0,010
Accuracy	0,881	0,921	0,159	0,808	1,000	0,000	0,908	1,000	0,000	0,981	0,995	0,016

Tabela 3.4: Resultados para Árvore de Decisão com a Base Melhorada

mostraram que houve leve piora. Porém, deve-se levar em consideração a melhora das demais Classes utilizando este classificador com a Base Melhorada.

Os resultados foram muito bons para todas as classes, sendo este o classificador definitivo para atender a proposta deste trabalho.

3.4 É possível reaproveitar o classificador em bases de dados de outras avaliações genéticas?

Através do `PICKLE`, um módulo de persistência de dados para armazenar/recuperar os classificadores gerados, foi possível testar outros conjuntos com este mesmo classificador sem necessidade de treinar novamente; os resultados permaneceram bons. O código está no Apêndice E. Para o Arquivo 3,

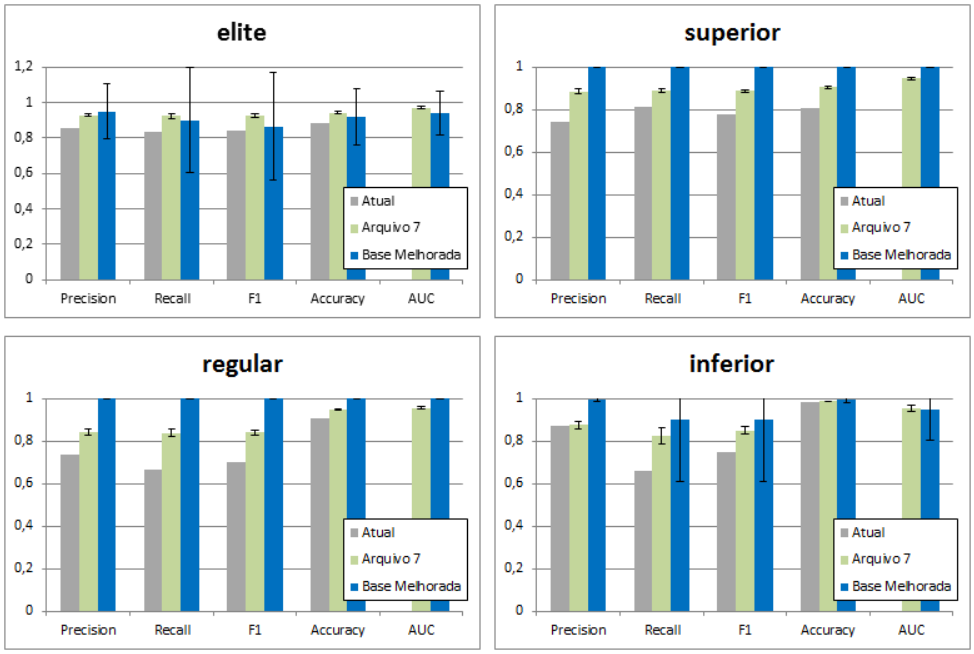


Figura 3.29: Comparativo de indutores para classe `elite`. No topo das barras estão representados o desvio-padrão (*Standard Deviation*)

amostra dos 49.169 machos que antes vinha sendo utilizado para as rodadas de treinamento, os resultados são mostrados na Tabela 3.5

Rodada 8: Classificador no Arquivo 4								
Métricas	ELITE		SUPERIOR		REGULAR		INFERIOR	
	Baseline	Classificador	Baseline	Classificador	Baseline	Classificador	Baseline	Classificador
F1	0,844	0,950	0,777	1,000	0,699	1,000	0,751	0,993
Recall	0,831	0,999	0,813	1,000	0,665	1,000	0,661	0,991
AUC	-	0,968	-	1,000	-	1,000	-	0,996
Precision	0,857	0,905	0,745	1,000	0,737	1,000	0,870	0,995
Accuracy	0,881	0,959	0,808	1,000	0,908	1,000	0,981	0,999

Tabela 3.5: Reutilização do Classificador com o Arquivo 4

Para o conjunto de dados de 85.567 fêmeas, os resultados estão na Tabela 3.6.

Rodada 9: Classificador no conjunto de dados de Fêmeas								
Métricas	ELITE		SUPERIOR		REGULAR		INFERIOR	
	Baseline	Classificador	Baseline	Classificador	Baseline	Classificador	Baseline	Classificador
F1	0,844	0,953	0,777	1,000	0,699	1,000	0,751	0,997
Recall	0,831	0,998	0,813	1,000	0,665	1,000	0,661	0,996
AUC	-	0,977	-	1,000	-	1,000	-	0,998
Precision	0,857	0,912	0,745	1,000	0,737	1,000	0,870	0,998
Accuracy	0,881	0,968	0,808	1,000	0,908	1,000	0,981	1,000

Tabela 3.6: Reutilização do Classificador com o conjunto de dados de 85.567 fêmeas

Para a amostra com 12.788 fêmeas que contribuíram somente com o Peso à Desmama, o resultado está na Tabela 3.7.

Rodada 10: Classificador no conjunto de dados de fêmeas menor								
Métricas	ELITE		SUPERIOR		REGULAR		INFERIOR	
	Baseline	Classificador	Baseline	Classificador	Baseline	Classificador	Baseline	Classificador
F1	0,844	0,955	0,777	1,000	0,699	1,000	0,751	0,995
Recall	0,831	0,999	0,813	1,000	0,665	1,000	0,661	0,995
AUC	-	0,969	-	1,000	-	1,000	-	0,998
Precision	0,857	0,916	0,745	1,000	0,737	1,000	0,870	0,995
Accuracy	0,881	0,960	0,808	1,000	0,908	1,000	0,981	1,000

Tabela 3.7: Reutilização do Classificador com o conjunto de dados de 12.788 fêmeas

Os três testes acima demonstram que o classificador gerado realmente melhora o processo atual de classificação da DEP para Peso à Desmama. E funcionou para o grupo de fêmeas que em nenhum momento foi utilizado como treinamento. Estes testes foram realizados sobre conjuntos de dados diferentes de uma mesma edição de avaliação genética, a mesma utilizada para gerar o classificador.

Outros testes foram executados com amostras de avaliações genéticas posteriores, uma em Novembro de 2014 e outra em Novembro de 2015.

Uma amostra com dados de 1.216 animais avaliados em Novembro de 2014, dez meses depois da avaliação genética cujas DEPs foram utilizadas para gerar o conjunto de treinamento, foi submetida ao classificador. Animais das Classes `elite` e `inferior`, que ficam nas extremidades da população, conseguem ser rotulados corretamente com ótima qualidade de área sob a curva ROC, de 97% e 98% respectivamente. O `recall` chega a 100% para as duas porém a `accuracy` é de 64% para a Classe `elite`, o que pode significar que tem animal que não é `elite` sendo classificado como tal. Já para a Classe `inferior`, a `accuracy` é de 92%.

As classes `superior` e `regular` não obtiveram métricas satisfatórias. De fato, o classificador não consegue classificar nenhuma instância corretamente. Conforme pode ser observado na Tabela 3.8.

Rodada 13: Classificador e dados de Novembro 2014				
Métricas	Elite	Superior	Regular	Inferior
F1	0,656	0,000	0,000	0,642
Recall	1,000	0,000	0,000	1,000
AUC	0,975	0,500	0,350	0,990
Precision	0,488	0,000	0,000	0,473
Accuracy	0,637	0,654	0,656	0,921

Tabela 3.8: Reutilização do Classificador com o conjunto de dados de Novembro 2014

Outra amostra com dados de 1.283 animais avaliados em Novembro de 2015, 22 meses depois da avaliação genética, também foi submetida ao classificador. O desempenho foi bastante semelhante à Rodada 13, com bons resultados para as Classes das extremidades e péssimos resultados para as Classes do meio, conforme pode ser observado na Tabela 3.9.

Rodada 14: Classificador e dados de Novembro 2015				
Métricas	Elite	Superior	Regular	Inferior
F1	0,725	0,000	0,000	0,634
Recall	1,000	0,000	0,000	1,000
AUC	0,970	0,500	0,364	0,991
Precision	0,569	0,000	0,000	0,465
Accuracy	0,672	0,684	0,677	0,935

Tabela 3.9: Reutilização do Classificador com o conjunto de dados de Novembro 2015

Na Figura 3.30, uma comparação dos desempenhos do classificador para as diferentes edições de avaliação genética.

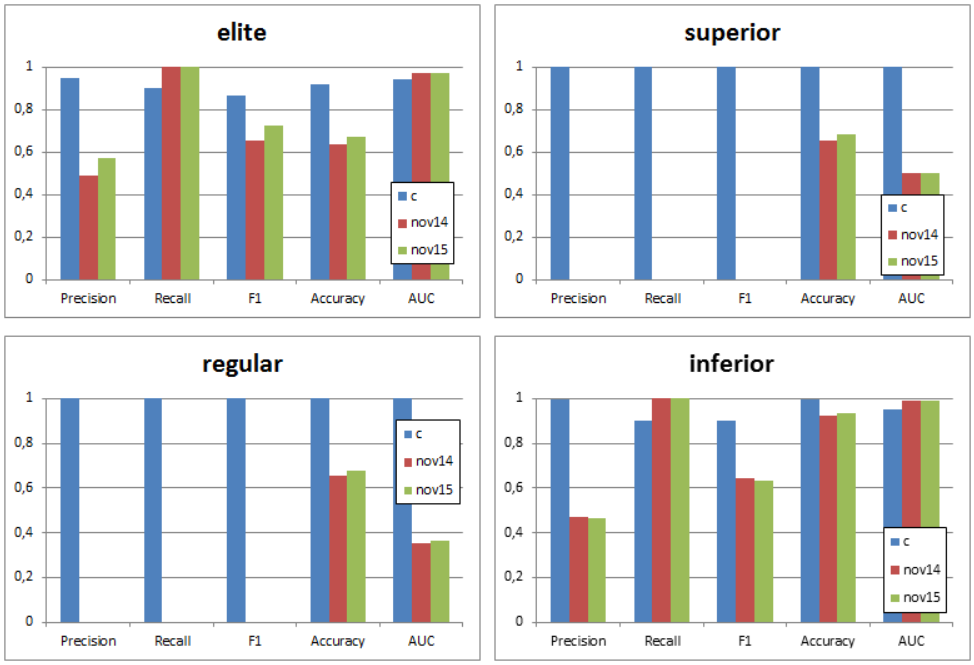


Figura 3.30: Comparativo do reaproveitamento do classificador nas avaliações de Novembro 2014 e Novembro 2015

O que se pode concluir disso é que utilizar um classificador antigo em bases de novas avaliações genéticas é possível para separar animais das Classes extremas, `elite` e `inferior`, com ressalvas ao fato de que haverá mais erros do que se criasse um novo classificador para cada edição.

3.5 Então, qual a importância de se utilizar o Peso Ajustado como atributo? Vale a pena?

Se, na Base Melhorada, não tivesse o Peso Ajustado como atributo, não mudaria quase nada, conforme pode ser visto na Tabela 3.10. Isto porque os escores são mais informativos do que ele. O classificador continuaria bom, mas praticamente o mesmo resultado da Tabela 3.4.

Rodada 11: Árvore de Decisão + Escores - PDaju												
Métricas	ELITE			SUPERIOR			REGULAR			INFERIOR		
	Baseline	Média	DP	Baseline	Média	DP	Baseline	Média	DP	Baseline	Média	DP
F1	0,844	0,867	0,302	0,777	1,000	0,000	0,699	1,000	0,000	0,751	0,902	0,290
Recall	0,831	0,901	0,298	0,813	1,000	0,000	0,665	1,000	0,000	0,661	0,902	0,295
AUC	-	0,941	0,121	-	1,000	0,000	-	1,000	0,000	-	0,951	0,148
Precision	0,857	0,948	0,154	0,745	1,000	0,000	0,737	1,000	0,000	0,870	0,998	0,005
Accuracy	0,881	0,921	0,157	0,808	1,000	0,000	0,908	1,000	0,000	0,981	0,995	0,016

Tabela 3.10: Resultados para o experimento com a Base Melhorada sem o Peso Ajustado

E se não tivesse nem o Peso Ajustado e nem os escores? Se o classificador dependesse somente das DEPs do pai e das DEPs da mãe como atributos, o resultado seria pior do que atualmente é feito, como pode ser visto na Tabela 3.11.

Rodada 12: Árvore de Decisão - Escores – PDaju												
Métricas	ELITE			SUPERIOR			REGULAR			INFERIOR		
	Baseline	Média	DP	Baseline	Média	DP	Baseline	Média	DP	Baseline	Média	DP
F1	0,844	0,537	0,363	0,777	0,551	0,241	0,699	0,357	0,155	0,751	0,575	0,338
Recall	0,831	0,575	0,425	0,813	0,501	0,269	0,665	0,295	0,203	0,661	0,557	0,365
AUC	-	0,931	0,112	-	0,777	0,168	-	0,87	0,089	-	0,932	0,115
Precision	0,857	0,769	0,314	0,745	0,708	0,185	0,737	0,797	0,234	0,870	0,812	0,343
Accuracy	0,881	0,753	0,128	0,808	0,709	0,133	0,908	0,851	0,058	0,981	0,963	0,036

Tabela 3.11: Resultados para o experimento com a Base Melhorada sem o Peso Ajustado e sem os escores

Se não houvesse escores, somente o Peso Ajustado e as DEPs do pai e as DEPs da mãe, o formato da Base Melhorada retornaria ao mesmo formato do Arquivo 4. Os resultados para o classificador utilizando a Base Melhorada sem os escores não foram tão bons quanto o classificador utilizando o Arquivo 4, que tem mais animais no treinamento, como pode ser observado nas Tabelas 3.12 e 3.13, respectivamente.

Na Figura 3.31, os gráficos comparando os seguintes itens:

- `baseline` = resultados baseados na predição da Classe da $DEP_{\text{PaiMãe}}$;
- Rodada 4 = resultados antes de incluir os `escores`, Tabela 3.3;

Rodada 15: Árvore de Decisão - Escores + PDaju												
Métricas	ELITE			SUPERIOR			REGULAR			INFERIOR		
	Baseline	Média	DP	Baseline	Média	DP	Baseline	Média	DP	Baseline	Média	DP
F1	0,844	0,807	0,216	0,777	0,59	0,308	0,699	0,504	0,192	0,751	0,627	0,282
Recall	0,831	0,83	0,257	0,813	0,546	0,316	0,665	0,427	0,234	0,661	0,595	0,312
AUC	-	0,938	0,1	-	0,859	0,149	-	0,922	0,07	-	0,93	0,131
Precision	0,857	0,883	0,145	0,745	0,744	0,252	0,737	0,875	0,178	0,870	0,883	0,209
Accuracy	0,881	0,868	0,111	0,808	0,754	0,155	0,908	0,884	0,037	0,981	0,965	0,034

Tabela 3.12: Resultados para o experimento com a Base Melhorada com o Peso Ajustado e sem os escores

Rodada 4: Árvore de Decisão												
Métricas	ELITE			SUPERIOR			REGULAR			INFERIOR		
	Baseline	Média	DP	Baseline	Média	DP	Baseline	Média	DP	Baseline	Média	DP
F1	0,844	0,926	0,008	0,777	0,887	0,007	0,699	0,84	0,012	0,751	0,849	0,018
Recall	0,831	0,925	0,014	0,813	0,89	0,009	0,665	0,839	0,018	0,661	0,826	0,037
AUC	-	0,972	0,005	-	0,947	0,005	-	0,955	0,006	-	0,955	0,013
Precision	0,857	0,928	0,007	0,745	0,885	0,012	0,737	0,842	0,014	0,870	0,875	0,02
Accuracy	0,881	0,943	0,006	0,808	0,907	0,006	0,908	0,949	0,004	0,981	0,988	0,001

Tabela 3.13: Resultados para o experimento com o Arquivo 4 com o Peso Ajustado e sem os escores

- Rodada 7 = resultados após incluir os `escores`, Tabela 3.4;
- Rodada 11 = resultados com `escores` e sem PDaju, Tabela 3.10;
- Rodada 12 = resultados sem `escores` e sem PDaju, Tabela 3.11; e
- Rodada 15 = resultados sem `escores` e com PDaju, Tabela 3.13

O que se pode concluir disso é que o Peso Ajustado tem sua importância na geração do classificador minimizada quando há `escores`, porém, na ausência destes, são informativos quanto maior for o conjunto de dados de treinamento.

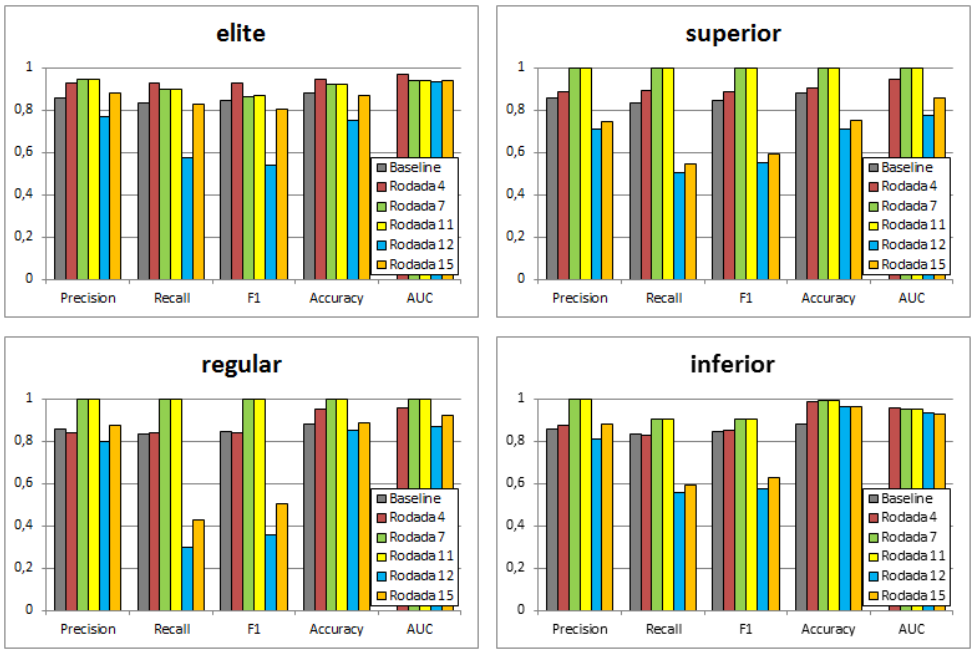


Figura 3.31: Resultados para o uso do PDaju como atributo: `baseline` = resultados baseados na predição da Classe da $DEP_{\text{PaiMãe}}$; Rodada 4 = resultados antes de incluir os `escores`; Rodada 7 = resultados após incluir os `escores`; Rodada 11 = resultados com `escores` e sem PDaju; Rodada 12 = resultados sem `escores` e sem PDaju; e Rodada 15 = resultados sem `escores` e com PDaju

CAPÍTULO

4 SIstema de Classificação Automática de DEP Bovina (SICADEB)

Neste capítulo é apresentado o protótipo do sistema desenvolvido para classificar um bovino em `elite`, `superior`, `regular` e `inferior` a partir das informações disponíveis à campo, da sua avaliação genética já existente e do classificador induzido por árvore de decisão.

4.1 Sistema

O Sistema de Classificação Automática de DEP Bovina (SICADEB) propõe-se a fazer uso de um classificador induzido por árvore de decisão, que consiga responder à pergunta: Dado um animal com seu peso e seus pais, qual a classificação (`elite`, `superior`, `regular` ou `inferior`) para a DEP do peso à desmama ?

Pretende-se que o criador use esta ferramenta para que, em tempo real, após coletar as pesagens à desmama de um lote de animais, o SICADEB consiga classificar eficientemente esse lote de animais para a DEP do peso à desmama. Com isto, o criador poderá descobrir animais promissores dentro do seu rebanho e animais candidatos a descarte. Também poderá avaliar o desempenho das vacas, mães dos bezerros, quanto à habilidade materna.

Atualmente, os sumários disponibilizados pelo Programa `GENEPLUS` já respondem a essa pergunta porém o método não leva em consideração o peso coletado.

O esquema de funcionamento do SICADEB é mostrado na Figura 4.1. O criador está com um lote de animais e faz a pesagem à desmama. Ou ele

digita o serviço ou extrai a saída de balanças eletrônicas e prepara uma lista de animais e respectivos pesos para submeter ao sumário da fazenda.

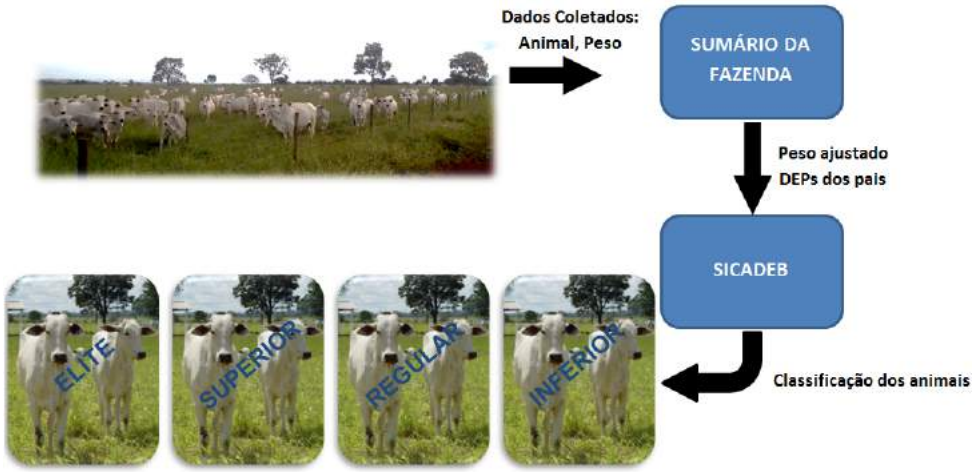


Figura 4.1: Proposta do sistema

No sumário da fazenda haverá um módulo específico para receber esta lista de animais e buscá-los na base de dados, para descobrir quem são seus pais e construir o arquivo de entrada para o SICADEB. O *input* do sistema é portanto o arquivo que tem os atributos descritos na Tabela 4.1. peso ajustado, DEPs

Atributos
Peso ajustado
DEPs do pai
DEPs da mãe
EscoreE
EscoreS
EscoreR
EscoreI
Classe para a DEP do peso à desmama

Tabela 4.1: Atributos de entrada para o SICADEB

do pai, DEPs da mãe, EscoreE, EscoreS, EscoreR, EscoreI e a classe para a DEP do peso à desmama do animal.

Esta classe que rotula os animais da lista do criador é criada com base na média da DEP do peso à desmama dos pais do animal: calcula-se a DEP média, calcula-se o `escore z`, descobre-se o percentil e atribui-se a Classe referente ao percentil.

Com o arquivo de atributos pronto, o SICADEB executa a classificação e define quem são os animais de cada Classe, exibindo também as métricas com a qualidade do resultado obtido.

Um protótipo do sistema é mostrado na Figura 4.2 . Hoje o sumário do Programa Geneplus é implementado utilizando o *software PARADOX for Windows (Corel Corporation, 1999)*, um sistema gerenciador de banco de dados antigo, com algumas incompatibilidades com sistemas operacionais mais novos. Já está em andamento um novo *software*, em linguagem JAVA, que substituirá o que está em uso. A execução do SICADEB como um módulo do sumário, que seja transparente ao usuário, só será possível em JAVA.

Figura 4.2: Tela Inicial - SICADEB no menu de entrada

No menu principal, o Módulo SICADEB seria incluído. Na Figura 4.3 o conteúdo da aba. Primeiramente é necessário preparar o arquivo que servirá de entrada para o classificador. Ao usuário será possível duas maneiras de entrada de dados: uma é escolhendo os animais via filtros e manualmente preencher os seus pesos coletados, e a outra é carregar um arquivo texto com os animais e pesagens já formatados, provenientes de saídas de outros bancos de dados ou mesmo da memória de balanças eletrônicas.

A entrada para o SICADEB é basicamente o animal e seu peso. Através do registro do animal, pode-se buscar no banco de dados do sumário da fazenda, as DEPs dos pais deste animal. A data da pesagem também é importante para que se possa fazer o ajuste do peso.

Com a lista dos animais exposta, estando todos corretamente apresentados, o usuário clica no botão “Classificar para DEP PD”, e a saída deste processamento é mostrada na Figura 4.4.

Ao clicar no botão para classificar os animais, um programa externo deve

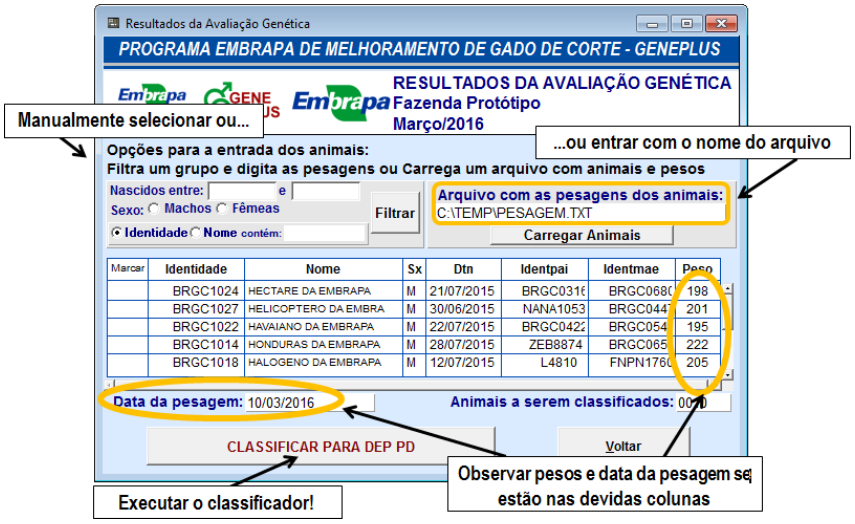


Figura 4.3: *Input* para o SICADEB

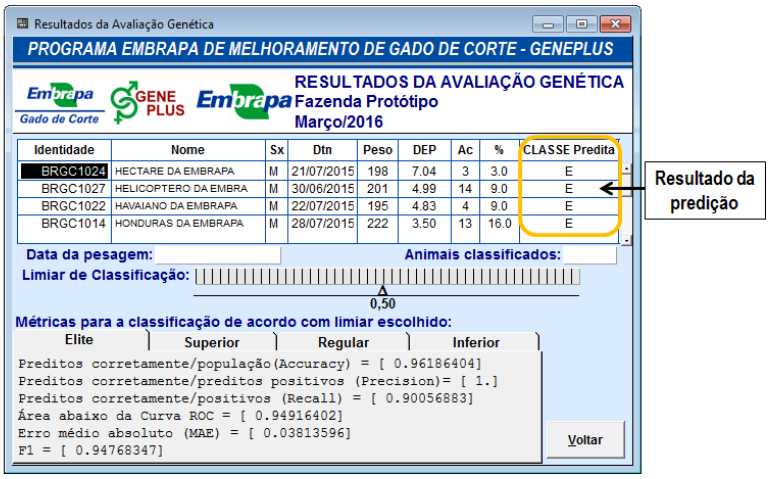


Figura 4.4: Tela Final - Resultados da classificação via SICADEB

executar o classificador salvo em arquivos de extensão "*.p", que são os classificadores salvos via módulo PICKLE. Após a execução externa, um arquivo texto é gerado com as saídas do classificador. Estas saídas serão as classes preditas.

A lista dos animais é apresentada novamente, com colunas para a DEP, acurácia, percentil (%) e a classe predita. O limiar padrão de classificação é de 50%, mas o usuário pode modificá-lo para verificar as métricas que melhor se aplicam ao seu caso. Tais métricas para cada classe são apresentadas logo abaixo do descritor de limiar.

Com este módulo SICADEB, espera-se que o usuário consiga aplicar de antemão os efeitos dos pesos de lotes de animais jovens, que na época da avaliação genética não tinham idade suficiente para coletar as medidas. As-

sim, terá uma previsão mais próxima da realidade independente de ter que esperar a próxima avaliação genética.

A princípio o estudo foi feito somente para o peso à desmama mas deverá ser ampliado para contemplar todas as outras características.

CAPÍTULO

5 Conclusões

Este trabalho teve como objetivo utilizar técnicas de aprendizado de máquina no banco de dados do Programa GENEPLUS, buscando melhorar o desempenho das predições da Classe para a Diferença Esperada na Progênie do peso à desmama.

Para se chegar no formato do arquivo de atributos final e no melhor algoritmo de classificação, vários experimentos foram feitos, alternando-se entre o pacote de software WEKA (*Waikato Environment for Knowledge Analysis*)[1], e implementações da biblioteca SKLEARN[2].

Os melhores resultados foram com Árvores de Decisão (*Decision Trees* - AD). Comparativamente com os algoritmos K-Vizinhos mais Próximos (KNN) e *Naive Bayes*(NB) demonstrou ser o mais eficiente.

Destes experimentos, algumas observações podem ser listadas:

- A Classe da DEP do peso à desmama de um animal sem pesagem à desmama, é dada pela média das DEPs de seus pais. Comparando a Classe real com a Classe predita pela média das DEPs dos pais, na base dos 49.169 machos, houve uma taxa de erro médio absoluto de 16,87% para a Classe `elite`; 18,73% para a Classe `superior`; 33,70% para a Classe `regular`; e 34,51% para a Classe `inferior`, demonstrando que a predição atual pode ser melhorada;

- Categorizar a DEP pela Classe não foi vantajoso como atributos pois os resultados obtidos foram os piores em comparação com o que já vem

[1] http://www.cs.waikato.ac.nz/~ml/weka/
[2] http://scikit-learn.org/

sendo utilizado atualmente. O melhor arquivo para qualquer classificador é o que contém as DEPs originais, sem categorização. Se necessário for, é possível aplicar a categorização por percentil, porém sob o risco de sofrer pequena desvantagem;

- Os atributos referentes ao avô materno não contribuem para melhorar o desempenho dos algoritmos. A presença destes atributos foi uma tentativa de verificar se a relação pai X avô materno seria mais relevante que pai X mãe, já que é prática usual para os produtores analisar um animal inicialmente pelo pai X avô materno;

- O Peso Ajustado (PDaju) que inicialmente esperava-se ser uma das principais informações para a mudança da $Classe_{ALVO}$, tem sua importância menor do que pressuposto. Contribui pouco porque os `escores` são mais informativos do que ele;

- Se o classificador dependesse somente das DEPs do pai e das DEPs da mãe como atributos, o resultado seria pior do que atualmente é feito. Logo, o Peso Ajustado também tem sua importância na geração do classificador porque, na ausência de `escores`, são informativos quanto maior for o arquivo de treinamento;

- Ao utilizar os escores, os erros de predição caem para 4% para a Classe `elite` e zero para as demais Classes para o classificador mais eficiente.

Antes de se utilizar os `escores` como atributos do arquivo, os melhores resultados tinham sido obtidos com o acréscimo das DEPs do animal calculadas com base na média das DEPs de seus pais.

A entrada dos `escores` como atributos contribuiu para os bons resultados. Como são calculados com base na média da DEP do Peso à Desmama da população, são campos facilmente incorporados ao arquivo de testes. Após aplicar a estratégia de `escore` para cada Classe, o classificador obteve o melhor custo-benefício.

Testes foram feitos para verificar se o reaproveitamento do classificador era possível para futuras avaliações genéticas. Como a base de dados utilizada para treinamento foi de janeiro de 2014, e estando agora em 2016, havia mais quatro avaliações posteriores para se testar a validade do classificador (junho e novembro de 2014, e junho e novembro de 2015).

Uma pequena amostra foi retirada da avaliação de novembro/2014 e outra de novembro/2015 e os resultados foram muito semelhantes. Excelente

para detectar as Classes das extremidades (`elite` e `inferior`) e péssimo para diferenciar as Classes intermediárias.

Portanto, reutilizar um classificador antigo em bases de novas avaliações genéticas é possível para separar animais das Classes extremas, com ressalvas ao fato de que haverá mais erros do que se criasse um novo classificador para cada edição. Não é possível determinar animais de Classes intermediárias.

Foi proposto um sistema, SIstema de Classificação Automática de DEP Bovina (SICADEB), para aplicação do classificador no *software* de retorno de resultados de avaliação genética, os sumários. Estes sumários já são amplamente utilizados pelos criadores associados ao Geneplus e o SICADEB seria um módulo a mais para que pudessem testar seus animais, submetendo pesagens para verificar qual a classificação consequente.

O SICADEB quando implementado, não visará substituir a avaliação genética tradicional mas dar mais subsídios para selecionar os animais com maior probabilidade de serem melhores dentre seus contemporâneos.

Para o futuro, pode-se estender este procedimento às demais DEPs possibilitando inclusive o cálculo do Índice de Qualificação Genética – IQG.

Referências Bibliográficas

Abreu, U. G. P. d., Sonohata, M. M., and Lopes, P. s. (2013). Definição de pesos econômicos e de índices de seleção para sistemas de produção. In Rosa, A. d. N., Martins, E. N., Menezes, G. R. d. O., and Silva, L. O. C. d., editors, Melhoramento genético aplicado em gado de corte: Programa Geneplus-Embrapa, chapter 11, pages 123–128. Embrapa Gado de Corte, Campo Grande, MS, 1 edition. Citado na página 31.

Batista, G. E. (2003). Pré-processamento de Dados em Aprendizado de Máquina Supervisionado. phdtese de doutorado, ICMC-USP, São Carlos - SP. Citado na página 6.

Bergstra, J. and Bengio, Y. (2012). Random search for hyper-parameter optimization. The Journal of Machine Learning Research, 13(1):281–305. Citado na página 67.

Breiman, L. (1984). Classification and regression trees. Chapman & Hall/CRC. Citado na página 13.

Cezar, I. M., Queiroz, H. P., Thiago, L., Cassales, F. L. G., and Costa, F. P. (2005). Sistemas de produção de gado de corte no Brasil: uma descrição com ênfase no regime alimentar e no abate. Embrapa Gado de Corte. Citado na página 43.

Cobuci, J., de ABREU, U., and Torres, R. (2006). Formação de grupos contemporâneos em bovinos de corte. Embrapa Pantanal. Documentos. Citado nas páginas 35, 48, e 52.

Daly, J. J. (1977). Melhoramento genético para produção de carne bovina. Queensland Department of Primary Industries. Citado nas páginas 2, 30, 33, 34, e 73.

de Souto, M., Lorena, A., Delbem, A., and de Carvalho, A. (2003). Técnicas de aprendizado de máquina para problemas de biologia molecular. Sociedade Brasileira de Computaçao. Citado na página 12.

dos Santos, C. J. G. (2010). Construção e identificação de hipóteses. [accessado em 29-Fevereiro-2016]. Citado na página 6.

Eler, J. P. (2014). Teorias e métodos em melhoramento genético animal: I bases do melhoramento genético animal. Universidade de S ao Paulo, Pirassununga, SP. Citado na página 33.

Euclides Filho, K. (2000a). Melhoramento genético animal no brasil: fundamentos, história e importância. Citado na página 39.

Euclides Filho, K. (2000b). Produção de bovinos de corte eo trinômio genótipo-ambiente-mercado. Citado na página 43.

Fawcett, T. (2006). An introduction to roc analysis. Pattern Recogn. Lett., 27(8):861–874. Citado nas páginas 26, 27, e 28.

Freund, Y. and Mason, L. (1999). The alternating decision tree learning algorithm. In icml, volume 99, pages 124–133. Citado na página 13.

Hastie, T., Tibshirani, R., Friedman, J., Hastie, T., Friedman, J., and Tibshirani, R. (2009). The elements of statistical learning, volume 2. Springer. Citado na página 12.

Henderson, C. R. (1953). Estimation of variance and covariance components. Biometrics, 9(2):226–252. Citado na página 36.

Júnior, J., Cardoso, V. L., Albuquerque, L. G. d., et al. (2007). Objetivos de seleção e valores econômicos em sistemas de produção de gado de corte no brasil. Revista Brasileira de Zootecnia, pages 1549–1558. Citado na página 44.

Kohavi, R. (1996). Scaling up the accuracy of naive-bayes classifiers: A decision-tree hybrid. In KDD, volume 96, pages 202–207. Citeseer. Citado na página 13.

Landwehr, N., Hall, M., and Frank, E. (2003). Logistic model trees. In Machine Learning: ECML 2003, pages 241–252. Springer. Citado na página 13.

Martín Nieto, L. and Rosa, A. d. N. (2013). Critérios de seleção. In Rosa, A. d. N., Martins, E. N., Menezes, G. R. d. O., and Silva, L. O. C. d., editors, Melhoramento genético aplicado em gado de corte: Programa Geneplus-Embrapa, chapter 10, pages 109–122. Embrapa Gado de Corte, Campo Grande, MS, 1 edition. Citado nas páginas 30, 31, 40, 49, e 50.

Martins, E. N. (2013). Avaliação genética: dos dados às deps. In Rosa, A. d. N., Martins, E. N., Menezes, G. R. d. O., and Silva, L. O. C. d., editors, Melhoramento genético aplicado em gado de corte: Programa

Geneplus-Embrapa, chapter 12, pages 129–148. Embrapa Gado de Corte, Campo Grande, MS, 1 edition. Citado na página 37.

Matsubara, E. T. (2008). Relações entre Ranking, Análise ROC e Calibração em Aprendizado de Máquina. phdtese de doutorado, ICMC-USP, São Carlos - SP. Citado nas páginas 7, 26, 27, e 28.

Mitchell, T. M. et al. (1997). Machine learning. wcb. Citado na página 15.

Monard, M. C. and Baranauskas, J. A. (2003). Conceitos sobre Aprendizado de Máquina, chapter 4, pages 89–114. Manole. Citado nas páginas 9, 10, e 12.

Morettin, P. A. (1981). Introdução à estatística para ciências exatas. Atual Editora Ltda. Citado nas páginas 71 e 72.

Nobre, P. R. C. (1996). Manual Técnico do Geneplus. Geneplus, Campo Grande, MS. Citado na página 50.

Nobre, P. R. C. (2014). Sumário de touros canchim, ma e charolês. Citado nas páginas 36, 39, e 40.

Nobre, P. R. C., Silva, L. O. C. d., Rosa, A. d. N., and Menezes, G. R. d. O. (2013a). Programa embrapa de melhoramento de gado de corte - geneplus. In Rosa, A. d. N., Martins, E. N., Menezes, G. R. d. O., and Silva, L. O. C. d., editors, Melhoramento genético aplicado em gado de corte: Programa Geneplus-Embrapa, chapter 19, pages 235–241. Embrapa Gado de Corte, Campo Grande, MS, 1 edition. Citado nas páginas 39, 42, e 43.

Nobre, P. R. C., Silva, L. O. C. d., Rosa, A. d. N., and Menezes, G. R. d. O. (2013b). Programa embrapa de melhoramento de gado de corte - geneplus. In Rosa, A. d. N., Martins, E. N., Menezes, G. R. d. O., and Silva, L. O. C. d., editors, Melhoramento genético aplicado em gado de corte: Programa Geneplus-Embrapa, chapter 19, pages 235–241. Embrapa Gado de Corte, Campo Grande, MS, 1 edition. Citado nas páginas 44, 46, e 75.

Pereira, J. C. C. (2012). Melhoramento genético aplicado à produção animal. FEPMVZ Editora, Belo Horizonte, MG. Citado nas páginas 32, 33, e 35.

Prati, R. C. (2006). Novas Abordagens em Aprendizado de Máquina para a Geração de Regras, Classes Desbalanceadas e Ordenação de Casos. phdtese de doutorado, ICMC-USP, São Carlos - SP. Citado nas páginas 6 e 11.

Quinlan, J. R. (1986). Induction of decision trees. Machine learning, 1(1):81–106. Citado nas páginas 10, 12, 13, e 16.

Quinlan, J. R. (2014). C4. 5: programs for machine learning. Elsevier. Citado na página 13.

Rezende, S. O., Pugliesi, J., Melanda, E., and Paula, M. d. (2003). Mineração de dados. Sistemas inteligentes: fundamentos e aplicações, 1:307–335. Citado na página 11.

Rosa, A. d. N., Menezes, G. R. d. O., and Egito, A. A. d. (2013). Recursos genéticos e estratégias de melhoramento. In Rosa, A. d. N., Martins, E. N., Menezes, G. R. d. O., and Silva, L. O. C. d., editors, Melhoramento genético aplicado em gado de corte: Programa Geneplus-Embrapa, chapter 2, pages 11–26. Embrapa Gado de Corte, Campo Grande, MS, 1 edition. Citado nas páginas 32, 33, e 48.

Sanches, M. K. (2003). Aprendizado de máquina semi-supervisionado: Proposta de um algoritmo para rotular exemplos a partir de poucos exemplos rotulados. mastersdissertação de mestrado, São Carlos - SP. Citado na página 7.

Shi, H. (2007). Best-first decision tree learning. PhD thesis, Citeseer. Citado na página 13.

Silva, L. O. C. d., Nobre, P. R. C., Torres Junior, R. A. d. A., Gondo, A., and Menezes, G. R. d. O. (2013). Uso dos sumários de avaliação genética nos processos de seleção e acasalamento. In Rosa, A. d. N., Martins, E. N., Menezes, G. R. d. O., and Silva, L. O. C. d., editors, Melhoramento genético aplicado em gado de corte: Programa Geneplus-Embrapa, chapter 14, pages 167–177. Embrapa Gado de Corte, Campo Grande, MS, 1 edition. Citado nas páginas 40 e 43.

Suguisawa, L., Matos, B. d. C. d., and Suguisawa, J. M. (2013). Uso da ultrassonografia na avaliação de características de carcaça e de qualidade da carne. In Rosa, A. d. N., Martins, E. N., Menezes, G. R. d. O., and Silva, L. O. C. d., editors, Melhoramento genético aplicado em gado de corte: Programa Geneplus-Embrapa, chapter 9, pages 97–107. Embrapa Gado de Corte, Campo Grande, MS, 1 edition. Citado na página 51.

Torres Junior, R. A. d. A., Silva, L. O. C. d., Menezes, G. R. d. O., and Nobre, P. R. C. (2013). Melhoramento animal na era das deps. In Rosa, A. d. N., Martins, E. N., Menezes, G. R. d. O., and Silva, L. O. C. d., editors, Melhoramento genético aplicado em gado de corte: Programa Geneplus-Embrapa, chapter 13, pages 149–166. Embrapa Gado de Corte, Campo Grande, MS, 1 edition. Citado nas páginas 34 e 40.

APÊNDICE

A Ganho de informação - Atributos do avô materno

```
INFO GAIN = ARQUIVO 2009-02-AnaliseAtributos.txt
=== Run information ===

Evaluator:    weka.attributeSelection.InfoGainAttributeEval
Search:       weka.attributeSelection.Ranker -T -1.7976931348623157E308
Relation:     SoPD-08.txt-weka.filters.unsupervised.attribute.Remove-R1
Instances:    49169
Attributes:   85
              des_weight_age
              des_weight_group_age
              PDaju
              DEP_PD_CL
              P_DEP_PN
              P_DEP_PM
              P_DEP_PD
              P_DEP_PS
              P_DEP_GPD
              P_DEP_PED
              P_DEP_PES
              P_DEP_CFD
              P_DEP_CFS
              P_DEP_PN_Pt
              P_DEP_PM_Pt
              P_DEP_PD_Pt
              P_DEP_PS_Pt
              P_DEP_GPD_Pt
              P_DEP_PED_Pt
              P_DEP_PES_Pt
              P_DEP_CFD_Pt
              P_DEP_CFS_Pt
```

P_DEP_PN_CL
P_DEP_PM_CL
P_DEP_PD_CL
P_DEP_PS_CL
P_DEP_GPD_CL
P_DEP_PED_CL
P_DEP_PES_CL
P_DEP_CFD_CL
P_DEP_CFS_CL
M_DEP_PN
M_DEP_PM
M_DEP_PD
M_DEP_PS
M_DEP_GPD
M_DEP_PED
M_DEP_PES
M_DEP_CFD
M_DEP_CFS
M_DEP_PN_Pt
M_DEP_PM_Pt
M_DEP_PD_Pt
M_DEP_PS_Pt
M_DEP_GPD_Pt
M_DEP_PED_Pt
M_DEP_PES_Pt
M_DEP_CFD_Pt
M_DEP_CFS_Pt
M_DEP_PN_CL
M_DEP_PM_CL
M_DEP_PD_CL
M_DEP_PS_CL
M_DEP_GPD_CL
M_DEP_PED_CL
M_DEP_PES_CL
M_DEP_CFD_CL
M_DEP_CFS_CL
AvoM_DEP_PN
AvoM_DEP_PM
AvoM_DEP_PD
AvoM_DEP_PS
AvoM_DEP_GPD
AvoM_DEP_PED
AvoM_DEP_PES
AvoM_DEP_CFD
AvoM_DEP_CFS
AvoM_DEP_PN_Pt

```
                AvoM_DEP_PM_Pt
                AvoM_DEP_PD_Pt
                AvoM_DEP_PS_Pt
                AvoM_DEP_GPD_Pt
                AvoM_DEP_PED_Pt
                AvoM_DEP_PES_Pt
                AvoM_DEP_CFD_Pt
                AvoM_DEP_CFS_Pt
                AvoM_DEP_PN_CL
                AvoM_DEP_PM_CL
                AvoM_DEP_PD_CL
                AvoM_DEP_PS_CL
                AvoM_DEP_GPD_CL
                AvoM_DEP_PED_CL
                AvoM_DEP_PES_CL
                AvoM_DEP_CFD_CL
                AvoM_DEP_CFS_CL
Evaluation mode:    10-fold cross-validation

=== Attribute selection 10 fold cross-validation (stratified), seed: 1 =

average merit      average rank  attribute
 0.49  +- 0.001        1   +- 0          7 P_DEP_PD
 0.487 +- 0.001        2   +- 0         16 P_DEP_PD_Pt
 0.377 +- 0.001        3.5 +- 0.5       43 M_DEP_PD_Pt
 0.377 +- 0.001        3.5 +- 0.5       34 M_DEP_PD
 0.369 +- 0.001        5   +- 0         25 P_DEP_PD_CL
 0.336 +- 0.013        6.6 +- 0.8        9 P_DEP_GPD
 0.334 +- 0.007        6.7 +- 0.64       8 P_DEP_PS
 0.323 +- 0.001        7.7 +- 0.46      52 M_DEP_PD_CL
 0.231 +- 0.002        9.5 +- 0.67      39 M_DEP_CFD
 0.231 +- 0.002       10   +- 0.63      48 M_DEP_CFD_Pt
 0.228 +- 0.003       10.9 +- 1.14       6 P_DEP_PM
 0.225 +- 0.002       11.6 +- 0.49      17 P_DEP_PS_Pt
 0.213 +- 0.002       13.2 +- 0.4       40 M_DEP_CFS
 0.213 +- 0.001       13.8 +- 0.4       49 M_DEP_CFS_Pt
 0.209 +- 0.002       15.2 +- 0.4       35 M_DEP_PS
 0.208 +- 0.001       16.4 +- 0.66      44 M_DEP_PS_Pt
 0.208 +- 0.002       16.7 +- 0.64      57 M_DEP_CFD_CL
 0.206 +- 0.001       17.7 +- 0.9       21 P_DEP_CFD_Pt
 0.194 +- 0.001       19   +- 0         12 P_DEP_CFD
 0.19  +- 0.001       20.1 +- 0.3       58 M_DEP_CFS_CL
 0.187 +- 0.001       21.6 +- 0.66      53 M_DEP_PS_CL
 0.186 +- 0.002       21.9 +- 1.22      22 P_DEP_CFS_Pt
 0.186 +- 0.001       22.5 +- 0.5        3 PDaju
 0.181 +- 0.001       24.4 +- 0.66      46 M_DEP_PED_Pt
```

```
0.181 +- 0.001      24.5 +- 0.5      37 M_DEP_PED
0.164 +- 0.001      26.5 +- 0.5      55 M_DEP_PED_CL
0.164 +- 0.002      26.5 +- 0.5      13 P_DEP_CFS
0.149 +- 0.001      28   +- 0        26 P_DEP_PS_CL
0.144 +- 0.001      29.3 +- 0.46     36 M_DEP_GPD
0.143 +- 0.001      29.9 +- 0.54     45 M_DEP_GPD_Pt
0.14  +- 0.002      31.4 +- 0.92     19 P_DEP_PED_Pt
0.138 +- 0.001      32.6 +- 0.66     33 M_DEP_PM
0.137 +- 0.001      33.1 +- 0.7      30 P_DEP_CFD_CL
0.136 +- 0.004      34   +- 2.86     11 P_DEP_PES
0.135 +- 0.001      34.7 +- 0.78     42 M_DEP_PM_Pt
0.133 +- 0.002      35.9 +- 0.83     10 P_DEP_PED
0.133 +- 0.002      36.4 +- 0.92      5 P_DEP_PN
0.13  +- 0.001      37.7 +- 0.64     54 M_DEP_GPD_CL
0.118 +- 0.001      39   +- 0        51 M_DEP_PM_CL
0.114 +- 0.001      40.3 +- 0.46     14 P_DEP_PN_Pt
0.113 +- 0.002      41.2 +- 0.98     18 P_DEP_GPD_Pt
0.111 +- 0.001      42.1 +- 0.7      31 P_DEP_CFS_CL
0.11  +- 0.001      42.4 +- 0.8      20 P_DEP_PES_Pt
0.107 +- 0.001      44.4 +- 0.49     38 M_DEP_PES
0.107 +- 0.001      44.6 +- 0.49     47 M_DEP_PES_Pt
0.096 +- 0.001      46   +- 0        56 M_DEP_PES_CL
0.083 +- 0.001      47   +- 0        15 P_DEP_PM_Pt
0.078 +- 0.001      48.2 +- 0.6      23 P_DEP_PN_CL
0.076 +- 0.001      49.4 +- 0.66     32 M_DEP_PN
0.076 +- 0          49.4 +- 0.49     41 M_DEP_PN_Pt
0.074 +- 0.001      51   +- 0        28 P_DEP_PED_CL
0.065 +- 0.001      52   +- 0        27 P_DEP_GPD_CL
0.061 +- 0          53   +- 0        50 M_DEP_PN_CL
0.031 +- 0          54   +- 0        29 P_DEP_PES_CL
0.024 +- 0.001      55.5 +- 0.5      63 AvoM_DEP_GPD
0.024 +- 0.001      56   +- 1.1      65 AvoM_DEP_PES
0.023 +- 0.001      57.6 +- 0.92     61 AvoM_DEP_PD
0.023 +- 0          58.8 +- 0.87     72 AvoM_DEP_GPD_Pt
0.023 +- 0.001      59.5 +- 1.8      74 AvoM_DEP_PES_Pt
0.023 +- 0          60.8 +- 0.6      68 AvoM_DEP_PN_Pt
0.023 +- 0.001      61   +- 4.67     60 AvoM_DEP_PM
0.023 +- 0          61.4 +- 1.2      62 AvoM_DEP_PS
0.023 +- 0          62   +- 0.45     59 AvoM_DEP_PN
0.022 +- 0.001      63.7 +- 1.35     66 AvoM_DEP_CFD
0.022 +- 0          64.2 +- 0.75     70 AvoM_DEP_PD_Pt
0.021 +- 0.001      66.4 +- 0.66     69 AvoM_DEP_PM_Pt
0.019 +- 0.001      67.6 +- 1.36     75 AvoM_DEP_CFD_Pt
0.019 +- 0.001      67.7 +- 1.68     76 AvoM_DEP_CFS_Pt
0.019 +- 0.001      68.5 +- 1.69     67 AvoM_DEP_CFS
0.018 +- 0          70   +- 1.1      24 P_DEP_PM_CL
```

```
0.018 +- 0          70.3 +- 0.64     73 AvoM_DEP_PED_Pt
0.017 +- 0          72   +- 0        64 AvoM_DEP_PED
0.016 +- 0          73   +- 0        71 AvoM_DEP_PS_Pt
0.014 +- 0          74   +- 0        81 AvoM_DEP_GPD_CL
0.008 +- 0          75   +- 0        80 AvoM_DEP_PS_CL
0.006 +- 0          76   +- 0        82 AvoM_DEP_PED_CL
0.006 +- 0          77   +- 0        79 AvoM_DEP_PD_CL
0.005 +- 0          78.1 +- 0.3      77 AvoM_DEP_PN_CL
0.005 +- 0          78.9 +- 0.3      78 AvoM_DEP_PM_CL
0.004 +- 0          80   +- 0        85 AvoM_DEP_CFS_CL
0.003 +- 0          81   +- 0        84 AvoM_DEP_CFD_CL
0.001 +- 0          82.1 +- 0.3      83 AvoM_DEP_PES_CL
0.001 +- 0          82.9 +- 0.3       1 des_weight_age
0     +- 0          84   +- 0         2 des_weight_group_age
```

APÊNDICE

B Ganho de informação - Atributos relacionados com DEP, Percentil e Classe

```
Evaluator:     weka.attributeSelection.InfoGainAttributeEval
Search:        weka.attributeSelection.Ranker -T -1.7976931348623157E308
Relation:      SoPD-08.txt-weka.filters.unsupervised.attribute.Remove-R1
Instances:     49169
Attributes:    56
               PDaju
               DEP_PD_CL
               P_DEP_PN
               P_DEP_PM
               P_DEP_PD
               P_DEP_PS
               P_DEP_GPD
               P_DEP_PED
               P_DEP_PES
               P_DEP_CFD
               P_DEP_CFS
               P_DEP_PN_Pt
               P_DEP_PM_Pt
               P_DEP_PD_Pt
               P_DEP_PS_Pt
               P_DEP_GPD_Pt
               P_DEP_PED_Pt
               P_DEP_PES_Pt
               P_DEP_CFD_Pt
               P_DEP_CFS_Pt
               P_DEP_PN_CL
               P_DEP_PM_CL
               P_DEP_PD_CL
               P_DEP_PS_CL
               P_DEP_GPD_CL
```

```
                P_DEP_PED_CL
                P_DEP_PES_CL
                P_DEP_CFD_CL
                P_DEP_CFS_CL
                M_DEP_PN
                M_DEP_PM
                M_DEP_PD
                M_DEP_PS
                M_DEP_GPD
                M_DEP_PED
                M_DEP_PES
                M_DEP_CFD
                M_DEP_CFS
                M_DEP_PN_Pt
                M_DEP_PM_Pt
                M_DEP_PD_Pt
                M_DEP_PS_Pt
                M_DEP_GPD_Pt
                M_DEP_PED_Pt
                M_DEP_PES_Pt
                M_DEP_CFD_Pt
                M_DEP_CFS_Pt
                M_DEP_PN_CL
                M_DEP_PM_CL
                M_DEP_PD_CL
                M_DEP_PS_CL
                M_DEP_GPD_CL
                M_DEP_PED_CL
                M_DEP_PES_CL
                M_DEP_CFD_CL
                M_DEP_CFS_CL
Evaluation mode:    10-fold cross-validation

=== Attribute selection 10 fold cross-validation (stratified), seed: 1 ===

average merit      average rank  attribute
 0.49  +- 0.001        1   +- 0          5 P_DEP_PD
 0.487 +- 0.001        2   +- 0         14 P_DEP_PD_Pt
 0.377 +- 0.001        3.5 +- 0.5       41 M_DEP_PD_Pt
 0.377 +- 0.001        3.5 +- 0.5       32 M_DEP_PD
 0.369 +- 0.001        5   +- 0         23 P_DEP_PD_CL
 0.336 +- 0.013        6.6 +- 0.8        7 P_DEP_GPD
 0.334 +- 0.007        6.7 +- 0.64       6 P_DEP_PS
 0.323 +- 0.001        7.7 +- 0.46      50 M_DEP_PD_CL
 0.231 +- 0.002        9.5 +- 0.67      37 M_DEP_CFD
 0.231 +- 0.002       10   +- 0.63      46 M_DEP_CFD_Pt
```

0.228 +- 0.003	10.9 +- 1.14	4 P_DEP_PM
0.225 +- 0.002	11.6 +- 0.49	15 P_DEP_PS_Pt
0.213 +- 0.002	13.2 +- 0.4	38 M_DEP_CFS
0.213 +- 0.001	13.8 +- 0.4	47 M_DEP_CFS_Pt
0.209 +- 0.002	15.2 +- 0.4	33 M_DEP_PS
0.208 +- 0.001	16.4 +- 0.66	42 M_DEP_PS_Pt
0.208 +- 0.002	16.7 +- 0.64	55 M_DEP_CFD_CL
0.206 +- 0.001	17.7 +- 0.9	19 P_DEP_CFD_Pt
0.194 +- 0.001	19 +- 0	10 P_DEP_CFD
0.19 +- 0.001	20.1 +- 0.3	56 M_DEP_CFS_CL
0.187 +- 0.001	21.6 +- 0.66	51 M_DEP_PS_CL
0.186 +- 0.002	21.9 +- 1.22	20 P_DEP_CFS_Pt
0.186 +- 0.001	22.5 +- 0.5	1 PDaju
0.181 +- 0.001	24.4 +- 0.66	44 M_DEP_PED_Pt
0.181 +- 0.001	24.5 +- 0.5	35 M_DEP_PED
0.164 +- 0.001	26.5 +- 0.5	53 M_DEP_PED_CL
0.164 +- 0.002	26.5 +- 0.5	11 P_DEP_CFS
0.149 +- 0.001	28 +- 0	24 P_DEP_PS_CL
0.144 +- 0.001	29.3 +- 0.46	34 M_DEP_GPD
0.143 +- 0.001	29.9 +- 0.54	43 M_DEP_GPD_Pt
0.14 +- 0.002	31.4 +- 0.92	17 P_DEP_PED_Pt
0.138 +- 0.001	32.6 +- 0.66	31 M_DEP_PM
0.137 +- 0.001	33.1 +- 0.7	28 P_DEP_CFD_CL
0.136 +- 0.004	34 +- 2.86	9 P_DEP_PES
0.135 +- 0.001	34.7 +- 0.78	40 M_DEP_PM_Pt
0.133 +- 0.002	35.9 +- 0.83	8 P_DEP_PED
0.133 +- 0.002	36.4 +- 0.92	3 P_DEP_PN
0.13 +- 0.001	37.7 +- 0.64	52 M_DEP_GPD_CL
0.118 +- 0.001	39 +- 0	49 M_DEP_PM_CL
0.114 +- 0.001	40.3 +- 0.46	12 P_DEP_PN_Pt
0.113 +- 0.002	41.2 +- 0.98	16 P_DEP_GPD_Pt
0.111 +- 0.001	42.1 +- 0.7	29 P_DEP_CFS_CL
0.11 +- 0.001	42.4 +- 0.8	18 P_DEP_PES_Pt
0.107 +- 0.001	44.4 +- 0.49	36 M_DEP_PES
0.107 +- 0.001	44.6 +- 0.49	45 M_DEP_PES_Pt
0.096 +- 0.001	46 +- 0	54 M_DEP_PES_CL
0.083 +- 0.001	47 +- 0	13 P_DEP_PM_Pt
0.078 +- 0.001	48.2 +- 0.6	21 P_DEP_PN_CL
0.076 +- 0	49.4 +- 0.49	39 M_DEP_PN_Pt
0.076 +- 0.001	49.4 +- 0.66	30 M_DEP_PN
0.074 +- 0.001	51 +- 0	26 P_DEP_PED_CL
0.065 +- 0.001	52 +- 0	25 P_DEP_GPD_CL
0.061 +- 0	53 +- 0	48 M_DEP_PN_CL
0.031 +- 0	54 +- 0	27 P_DEP_PES_CL
0.018 +- 0	55 +- 0	22 P_DEP_PM_CL

APÊNDICE

C Atributos do arquivo de treinamento

Campo	Exemplo	Descrição	
Número_animal	2074550	Número do animal	
Número_pai	2003690	Número do pai do animal	
Número_mãe	2071642	Número da mãe do animal	
PD	216.49	Peso à desmama	
GCD	967544	Grupo contemporâneo	
PDaju	3.48	Peso à desmama ajustado	
DEP_PN	-0.04	DEP do animal	Peso ao nascer
DEP_PM	-1.26		Peso aos 120 dias
DEP_PD	-1.53		Peso à desmama
DEP_PS	-3.35		Peso ao sobreano
DEP_GPD	-5.2		Ganho pós-desmama
DEP_PED	-0.1		Perímetro escrotal à desmama
DEP_PES	-0.14		Perímetro escrotal ao sobreano
DEP_CFD	-0.12		Conformação frigorífica à desmama
DEP_CFS	-0.11		Conformação frigorífica ao sobreano
DEP_PN_Pt	48	Percentil da DEP do animal	Peso ao nascer
DEP_PM_Pt	85		Peso aos 120 dias
DEP_PD_Pt	65		Peso à desmama
DEP_PS_Pt	77		Peso ao sobreano
DEP_GPD_Pt	77		Ganho pós-desmama
DEP_PED_Pt	84		Perímetro escrotal à desmama
DEP_PES_Pt	69		Perímetro escrotal ao sobreano
DEP_CFD_Pt	90		Conformação frigorífica à desmama
DEP_CFS_Pt	89		Conformação frigorífica ao sobreano

DEP_PN_CL	1	Classe da DEP do animal	Peso ao nascer
DEP_PM_CL	3		Peso aos 120 dias
DEP_PD_CL	2		Peso à desmama
DEP_PS_CL	2		Peso ao sobreano
DEP_GPD_CL	2		Ganho pós-desmama
DEP_PED_CL	2		Perímetro escrotal à desmama
DEP_PES_CL	2		Perímetro escrotal ao sobreano
DEP_CFD_CL	3		Conformação frigorífica à desmama
DEP_CFS_CL	3		Conformação frigorífica ao sobreano
P_DEP_PN	0.07	DEP do pai	Peso ao nascer
P_DEP_PM	-2.23		Peso aos 120 dias
P_DEP_PD	-0.21		Peso à desmama
P_DEP_PS	-4.33		Peso ao sobreano
P_DEP_GPD	-5.44		Ganho pós-desmama
P_DEP_PED	-0.12		Perímetro escrotal à desmama
P_DEP_PES	-0.18		Perímetro escrotal ao sobreano
P_DEP_CFD	-0.15		Conformação frigorífica à desmama
P_DEP_CFS	-0.15		Conformação frigorífica ao sobreano
P_DEP_PN_Pt	62	Percentil da DEP do pai	Peso ao nascer
P_DEP_PM_Pt	97		Peso aos 120 dias
P_DEP_PD_Pt	51		Peso à desmama
P_DEP_PS_Pt	84		Peso ao sobreano
P_DEP_GPD_Pt	78		Ganho pós-desmama
P_DEP_PED_Pt	88		Perímetro escrotal à desmama
P_DEP_PES_Pt	74		Perímetro escrotal ao sobreano
P_DEP_CFD_Pt	95		Conformação frigorífica à desmama
P_DEP_CFS_Pt	95		Conformação frigorífica ao sobreano
P_DEP_PN_CL	2	Classe da DEP do pai	Peso ao nascer
P_DEP_PM_CL	3		Peso aos 120 dias
P_DEP_PD_CL	2		Peso à desmama
P_DEP_PS_CL	2		Peso ao sobreano
P_DEP_GPD_CL	2		Ganho pós-desmama
P_DEP_PED_CL	3		Perímetro escrotal à desmama
P_DEP_PES_CL	2		Perímetro escrotal ao sobreano
P_DEP_CFD_CL	3		Conformação frigorífica à desmama
P_DEP_CFS_CL	3		Conformação frigorífica ao sobreano
M_DEP_PN	-0.23	DEP da mãe	Peso ao nascer
M_DEP_PM	-0.71		Peso aos 120 dias
M_DEP_PD	-6.45		Peso à desmama
M_DEP_PS	-5.66		Peso ao sobreano
M_DEP_GPD	-8.37		Ganho pós-desmama
M_DEP_PED	-0.14		Perímetro escrotal à desmama
M_DEP_PES	-0.32		Perímetro escrotal ao sobreano
M_DEP_CFD	-0.15		Conformação frigorífica à desmama
M_DEP_CFS	-0.14		Conformação frigorífica ao sobreano

M_DEP_PN_Pt	24	Percentil da DEP da mãe	Peso ao nascer
M_DEP_PM_Pt	72		Peso aos 120 dias
M_DEP_PD_Pt	96		Peso à desmama
M_DEP_PS_Pt	90		Peso ao sobreano
M_DEP_GPD_Pt	88		Ganho pós-desmama
M_DEP_PED_Pt	92		Perímetro escrotal à desmama
M_DEP_PES_Pt	87		Perímetro escrotal ao sobreano
M_DEP_CFD_Pt	95		Conformação frigorífica à desmama
M_DEP_CFS_Pt	93		Conformação frigorífica ao sobreano
M_DEP_PN_CL	1	Classe da DEP da mãe	Peso ao nascer
M_DEP_PM_CL	2		Peso aos 120 dias
M_DEP_PD_CL	3		Peso à desmama
M_DEP_PS_CL	3		Peso ao sobreano
M_DEP_GPD_CL	3		Ganho pós-desmama
M_DEP_PED_CL	3		Perímetro escrotal à desmama
M_DEP_PES_CL	3		Perímetro escrotal ao sobreano
M_DEP_CFD_CL	3		Conformação frigorífica à desmama
M_DEP_CFS_CL	3		Conformação frigorífica ao sobreano
Número_avô_materno	2000000	Número do avô materno	
AvoM_DEP_PN	-0.09	DEP do avô materno	Peso ao nascer
AvoM_DEP_PM	-0.09		Peso aos 120 dias
AvoM_DEP_PD	-6.2		Peso à desmama
AvoM_DEP_PS	-7.45		Peso ao sobreano
AvoM_DEP_GPD	-7.88		Ganho pós-desmama
AvoM_DEP_PED	-0.17		Perímetro escrotal à desmama
AvoM_DEP_PES	-0.16		Perímetro escrotal ao sobreano
AvoM_DEP_CFD	-0.14		Conformação frigorífica à desmama
AvoM_DEP_CFS	-0.1		Conformação frigorífica ao sobreano
AvoM_DEP_PN_Pt	42	Percentil da DEP do avô materno	Peso ao nascer
AvoM_DEP_PM_Pt	53		Peso aos 120 dias
AvoM_DEP_PD_Pt	95		Peso à desmama
AvoM_DEP_PS_Pt	96		Peso ao sobreano
AvoM_DEP_GPD_Pt	87		Ganho pós-desmama
AvoM_DEP_PED_Pt	96		Perímetro escrotal à desmama
AvoM_DEP_PES_Pt	72		Perímetro escrotal ao sobreano
AvoM_DEP_CFD_Pt	94		Conformação frigorífica à desmama
AvoM_DEP_CFS_Pt	87		Conformação frigorífica ao sobreano
AvoM_DEP_PN_CL	1	Classe da DEP do avô materno	Peso ao nascer
AvoM_DEP_PM_CL	2		Peso aos 120 dias
AvoM_DEP_PD_CL	3		Peso à desmama
AvoM_DEP_PS_CL	3		Peso ao sobreano
AvoM_DEP_GPD_CL	3		Ganho pós-desmama
AvoM_DEP_PED_CL	3		Perímetro escrotal à desmama
AvoM_DEP_PES_CL	2		Perímetro escrotal ao sobreano
AvoM_DEP_CFD_CL	3		Conformação frigorífica à desmama
AvoM_DEP_CFS_CL	3		Conformação frigorífica ao sobreano

APÊNDICE

D Código python + SKLEARN para classificar os arquivos

```
#!/usr/bin/python

import scipy
import sys
import os
import numpy as np
import pickle

from sklearn import tree
from sklearn import svm
from sklearn import neighbors
from sklearn import naive_bayes
from sklearn import linear_model
from sklearn.metrics import roc_auc_score
from sklearn.metrics import precision_score
from sklearn.metrics import recall_score
from sklearn.metrics import f1_score
from sklearn.metrics import accuracy_score
from sklearn.metrics import mean_absolute_error
from sklearn.preprocessing import LabelBinarizer
from sklearn.cross_validation import StratifiedKFold
from sklearn.grid_search import RandomizedSearchCV
from sklearn.calibration import CalibratedClassifierCV

class Dataset():
    def __init__(self):
        self.data = []
        self.target = []
        self.header = None
        self.class_att = None
```

```
        self.name = None

    def set_class_attribute(self,att = -1):
        if self.header == None:
            raise NameError('please load dataset first')
        attpos = att
        if type(att) == str:
            if att in self.header:
                attpos = self.header[att]
            else:
                raise NameError('%s not found in the loaded dataset'%(att))
        data = []
        head = []
        tail = []
        m = np.matrix(self.data)
        self.target = np.array([int(v) for v in m[:,attpos].transpose().tol
        self.data = np.delete(m,np.s_[attpos],axis=1)


    def load(self,datasetname=''):
        self.name = datasetname
        header = None
        f = open(datasetname)
        data = []
        for line in f:
            v = line.rstrip('\n').split(',')
            #print v
            if header == None:
                header = dict()
                for i in range(len(v)):
                    header[v[i]] = i
                self.header = header
            else:

                data.append([float(x) for x in v])
        self.data = np.matrix(data)

class LClassifier(object): # Abstract class for classifier learning
    def __init__(self):
        self.lm       = []
        self.lb       = LabelBinarizer()
        self.clearner = []
        self.cbest_score = []
        self.cbest_params = []
        self.cpredict = []
        self.n_iter = 10
```

```
        self.predicted_proba_ = None
        self.predicted_ = None
        self.scoring = 'precision'

    def fit(self,x,y):

        self.lb.fit(y)
        self.lm = self.lb.transform(y)
        self.clearner = []
        if len(self.lb.classes_) == 2:
            if self.param_dist == None:
                self.learner.fit(x,y)
                clf = self.learner
            else:
                # Find best classifier using a random sampling
                random_search = RandomizedSearchCV(self.learner,param_d

                random_search.fit(x,y)
                print random_search.best_score_
                print random_search.best_params_
                clf = random_search.best_estimator_
                self.cbest_score.append(random_search.best_score_)
                self.cbest_params.append(random_search.best_params_)


            # calibrate classifier

            isotonic = CalibratedClassifierCV(clf, method='isotonic')
            isotonic.fit(x,y)
            self.clearner.append(isotonic)

        else:
            for iclass in range(len(self.lb.classes_)):
                print "Running class = ", iclass
                ty = self.lm[:,iclass]

                if self.param_dist == None:
                    self.learner.fit(x,ty)
                    clf = self.learner
                    #pickle.dump(clf, open( "classificador_02"+str(icla
                else:
                    # Find best classifier using a random sampling
                    random_search = RandomizedSearchCV(self.learner,par

                    random_search.fit(x,ty)
                    print random_search.best_score_
```

```
                print random_search.best_params_
                clf = random_search.best_estimator_.fit(x,ty)
                self.cbest_score.append(random_search.best_score_)
                self.cbest_params.append(random_search.best_params_)


            # calibrate classifier

            isotonic = CalibratedClassifierCV(clf, method='isotonic')
            isotonic.fit(x,ty)
            self.clearner.append(isotonic)
            pickle.dump(isotonic, open( "classificadorISO_02"+str(iclas

    def predict_proba(self,X):
        if len(self.lb.classes_) == 2:
            clf = self.clearner[0]
            self.predicted_proba_ = clf.predict_proba(X)
            self.predicted_ = clf.predict(X)
            return self.predicted_proba_
        else:
            cpredict_proba = []
            cpredict = []
            for clf in self.clearner:
                cpredict_proba.append(clf.predict_proba(X)[:,1])
                cpredict.append(clf.predict(X))
            predicted_proba = []
            predicted = []
            for i in range(len(X)):
                prob = []
                pred = []
                for j in range(len(self.clearner)):
                    prob.append(cpredict_proba[j][i])
                    pred.append(cpredict[j][i])
                predicted_proba.append(np.array(prob))
                predicted.append(np.array(pred))
            self.predicted_proba_ = predicted_proba
            self.predicted_ = predicted
            return predicted_proba

    def compute_measures(self,x,y):

        self.predict_proba(x)
        predicted_proba = np.array(self.predicted_proba_)
        predicted       = np.array(self.predicted_)
        lm        = self.lb.transform(y)
```

```
print "predicted_proba:"
print predicted_proba
print "predicted:"
print predicted
Sai_Proba = open('Roda001-Predict-DT-DEP-M-Full.out','a')
Sai_Proba.write('------------'+'\n')
np.savetxt(Sai_Proba, predicted_proba, delimiter=',', fmt='%2.6
Sai_Proba.write('------------'+'\n')
np.savetxt(Sai_Proba, predicted, delimiter=',', fmt='%i')
Sai_Proba.write('x-x-x-x-x-x-x-x-x'+'\n')
np.savetxt(Sai_Proba, y, delimiter=',', fmt='%i')
Sai_Proba.write('fim--------------------'+'\n')
Sai_Proba.close()

auc       = []
precision = []
recall    = []
f1        = []
accuracy  = []
mae       = []

r = dict()
if len(self.lb.classes_) == 2:
    y_true = np.array(y)
    y_scores = predicted_proba[:,1]
    y_pred = predicted
    auc.append(roc_auc_score(y_true, y_scores))
    precision.append(precision_score(y_true,y_pred))
    recall.append(recall_score(y_true,y_pred))
    f1.append(f1_score(y_true,y_pred))
    accuracy.append(accuracy_score(y_true,y_pred))
else:
    for iclass in range(len(self.lb.classes_)):
        y_true   = lm[:,iclass]
        y_scores = predicted_proba[:,iclass]
        y_pred   = predicted[:,iclass]
        auc.append(roc_auc_score(y_true, y_scores))
        precision.append(precision_score(y_true,y_pred))
        recall.append(recall_score(y_true,y_pred))
        f1.append(f1_score(y_true,y_pred))
        accuracy.append(accuracy_score(y_true,y_pred))
        mae.append(mean_absolute_error(y_true, y_pred))
r['auc']       = auc
r['precision'] = precision
r['recall']    = recall
r['f1']        = f1
```

```
        r['accuracy']  = accuracy
        r['mae']       = mae
        return r

class LGNB(LClassifier):
    def __init__(self):
        super(LGNB,self).__init__()
        self.name = 'gnb'
        self.learner = naive_bayes.GaussianNB()
        self.param_dist = None

class LKNN(LClassifier):
    def __init__(self):
        super(LKNN,self).__init__()
        self.name = 'knn'
        self.learner = neighbors.KNeighborsClassifier()
        self.param_dist = {'n_neighbors': range(1,50)}
        self.n_iter = 30

class LDT(LClassifier):
    def __init__(self):
        super(LDT,self).__init__()
        self.name = 'dt'
        self.learner = tree.DecisionTreeClassifier()
        self.param_dist = None

class Experiment():
    def __init__(self,datasets,algorithms):
        self.datasets = datasets
        self.algorithms = algorithms
        self.results = None
        self.num_folds = 10
        self.results = []

    def save_results(self):
        f = open('Roda001-Results-DT-DEP-M-Full.csv','w')
        res = []
        for line in self.result:
            res.append(','.join(line))
        f.write('\n'.join(res))
        f.close()
        print 'results are saved on results.csv'

    def run(self):
        self.result = [['dataset','algorithm','measure','class','mean','std
        for data in self.datasets:
```

```
            skf = StratifiedKFold(data.target,self.num_folds)
            fr  = dict()
            for clf in self.algorithms:
                for train,test in skf:
                    clf.fit(data.data[train], data.target[train])
                    r = clf.compute_measures(data.data[test],data.targe
                    print 'retornou com r'

                    for measure in r:
                        if measure not in fr:
                            fr[measure] = []
                        fr[measure].append(np.array(r[measure]))
                for measure in fr:
                    m = np.array(fr[measure])

                    header = [data.name,clf.name,measure]
                    print  data.name,clf.name,measure
                    for iclass in range(m.shape[1]): # get number of cl
                        v = m[:,iclass]
                        print 'vvvvvv '
                        mean= np.mean(v)
                        std = np.std(v)
                        self.result.append(header+['%d'%(iclass)]+['%4.
                        print iclass,mean,std
                self.save_results()

if __name__ == '__main__':
    ok = False
    usage_str ="usage:%s <dataset in csv format|directory with csv file

    datasets = []

    if len(sys.argv) == 2:
        if os.path.isfile(sys.argv[1]):
            data = Dataset()
            data.load(sys.argv[1])
            data.set_class_attribute()
            datasets.append(data)
        elif os.path.isdir(sys.argv[1]):
            maindir = sys.argv[1]
            for filename in os.listdir(maindir):
                if (filename.find('csv') > 0):
                    data = Dataset()
                    data.load(maindir+os.sep+filename)
                    data.set_class_attribute()
                    datasets.append(data)
```

```
if len(datasets) == 0:
    print  usage_str
    sys.exit(0)
exp = Experiment(datasets,[LDT()])
exp.run()
```

APÊNDICE

E

Código python + SKLEARN para recuperar o classificador gravado

```
#!/usr/bin/python

import scipy
import sys
import os
import numpy as np
import pickle


from sklearn import tree
from sklearn import svm
from sklearn import neighbors
from sklearn import naive_bayes
from sklearn import linear_model
from sklearn.metrics import roc_auc_score
from sklearn.metrics import precision_score
from sklearn.metrics import recall_score
from sklearn.metrics import f1_score
from sklearn.metrics import accuracy_score
from sklearn.metrics import mean_absolute_error
from sklearn.preprocessing import LabelBinarizer
from sklearn.cross_validation import StratifiedKFold
from sklearn.grid_search import RandomizedSearchCV
from sklearn.calibration import CalibratedClassifierCV

class Dataset():
    def __init__(self):
        self.data = []
        self.target = []
        self.header = None
```

```
        self.class_att = None
        self.name = None

    def set_class_attribute(self,att = -1):
        if self.header == None:
            raise NameError('please load dataset first')
        attpos = att
        if type(att) == str:
            if att in self.header:
                attpos = self.header[att]
            else:
                raise NameError('%s not found in the loaded dataset'%(att))
        data = []
        head = []
        tail = []
        m = np.matrix(self.data)
        self.target = np.array([int(v) for v in m[:,attpos].transpose().tol
        self.data = np.delete(m,np.s_[attpos],axis=1)


    def load(self,datasetname=''):
        self.name = datasetname
        header = None
        f = open(datasetname)
        data = []
        for line in f:
            v = line.rstrip('\n').split(',')
            #print v
            if header == None:
                header = dict()
                for i in range(len(v)):
                    header[v[i]] = i
                self.header = header
            else:

                data.append([float(x) for x in v])
        self.data = np.matrix(data)



if __name__ == '__main__':
    ok = False
    usage_str ="usage:%s <dataset in csv format|directory with csv files>\n

    datasets = []
```

```
if len(sys.argv) == 2:
    print 'argv2'
    if os.path.isfile(sys.argv[1]):
        data = Dataset()
        data.load(sys.argv[1])
        data.set_class_attribute()
        datasets.append(data)
    elif os.path.isdir(sys.argv[1]):
        maindir = sys.argv[1]
        for filename in os.listdir(maindir):
            if (filename.find('csv') > 0):
                data = Dataset()
                data.load(maindir+os.sep+filename)
                data.set_class_attribute()
                datasets.append(data)
if len(datasets) == 0:
    print  usage_str
    sys.exit(0)

y = data.target
print y
X = data.data
print X

lb = LabelBinarizer()
lb.fit(y)
lm = lb.transform(y)
print lb.classes_
print lm

for iclasse in range(len(lb.classes_)):
    print 'vai classe ', iclasse
    ty = lm[:, iclasse]
    print ty
    clf = pickle.load(open( "classificadorISO_02"+str(iclasse)+".p"
    cpredict_proba = []
    cpredict = []

    cpredict_proba.append(clf.predict_proba(X)[:,1])
    cpredict.append(clf.predict(X))
    print cpredict_proba
    print cpredict

    predicted_proba = []
    predicted = []
    print "len(X) = ", len(X)
```

```
for i in range(len(X)):
    prob = []
    pred = []

    prob.append(cpredict_proba[0][i])
    pred.append(cpredict[0][i])

    predicted_proba.append(np.array(prob))
    predicted.append(np.array(pred))

predicted_proba_ = predicted_proba
predicted_ = predicted

predicted_proba_AG = np.array(predicted_proba_)
predicted_AG       = np.array(predicted_)
print "predicted_proba:"
print predicted_proba_AG
print "predicted:"
print predicted_AG

Sai_Proba = open('Roda001-PICKLE-DT-DEP-M-Full.out','a')
Sai_Proba.write('------------'+'\n')
np.savetxt(Sai_Proba, predicted_proba_AG, delimiter=',', fmt='%2.6f
Sai_Proba.write('------------'+'\n')
np.savetxt(Sai_Proba, predicted_AG, delimiter=',', fmt='%i')
Sai_Proba.write('x-x-x-x-x-x-x-x-x'+'\n')
np.savetxt(Sai_Proba, y, delimiter=',', fmt='%i')
Sai_Proba.write('fim--------------------'+'\n')
Sai_Proba.close()

auc       = []
precision = []
recall    = []
f1        = []
accuracy  = []
mae       = []
r = dict()

y_true   = lm[:,iclasse]
y_scores = predicted_proba_AG[:,0]
y_pred   = predicted_AG[:,0]
auc.append(roc_auc_score(y_true, y_scores))
precision.append(precision_score(y_true,y_pred))
recall.append(recall_score(y_true,y_pred))
f1.append(f1_score(y_true,y_pred))
accuracy.append(accuracy_score(y_true,y_pred))
```

```
mae.append(mean_absolute_error(y_true, y_pred))
r['auc']       = auc
r['precision'] = precision
r['recall']    = recall
r['f1']        = f1
r['accuracy']  = accuracy
r['mae']       = mae

print r
fr = dict()
for measure in r:
    if measure not in fr:
        fr[measure] = []
        fr[measure].append(np.array(r[measure]))
print "fr: ", fr

f = open('Roda001-RPICKLE-DT-DEP-M-Full.csv','a')
for measure in fr:
    m = np.array(fr[measure])
    header = measure
    f.write('Classe '+ str(iclasse) + ' ====> '+ header+ ' = '
f.close()
print 'results are saved on results.csv'
```

APÊNDICE

F

Tabela da Distribuição Acumulada da Normal Padrão

z	0,00	0,01	0,02	0,03	0,04	0,05	0,06	0,07	0,08	0,09
0,0	0,50000	0,50399	0,50798	0,51197	0,51595	0,51994	0,52392	0,52790	0,53188	0,53586
0,1	0,53983	0,54380	0,54776	0,55172	0,55567	0,55962	0,56356	0,56749	0,57142	0,57535
0,2	0,57926	0,58317	0,58706	0,59095	0,59483	0,59871	0,60257	0,60642	0,61026	0,61409
0,3	0,61791	0,62172	0,62552	0,62930	0,63307	0,63683	0,64058	0,64431	0,64803	0,65173
0,4	0,65542	0,65910	0,66276	0,66640	0,67003	0,67364	0,67724	0,68082	0,68439	0,68793
0,5	0,69146	0,69497	0,69847	0,70194	0,70540	0,70884	0,71226	0,71566	0,71904	0,72240
0,6	0,72575	0,72907	0,73237	0,73565	0,73891	0,74215	0,74537	0,74857	0,75175	0,75490
0,7	0,75804	0,76115	0,76424	0,76730	0,77035	0,77337	0,77637	0,77935	0,78230	0,78524
0,8	0,78814	0,79103	0,79389	0,79673	0,79955	0,80234	0,80511	0,80785	0,81057	0,81327
0,9	0,81594	0,81859	0,82121	0,82381	0,82639	0,82894	0,83147	0,83398	0,83646	0,83891
1,0	0,84134	0,84375	0,84614	0,84850	0,85083	0,85314	0,85543	0,85769	0,85993	0,86214
1,1	0,86433	0,86650	0,86864	0,87076	0,87286	0,87493	0,87698	0,87900	0,88100	0,88298
1,2	0,88493	0,88686	0,88877	0,89065	0,89251	0,89435	0,89617	0,89796	0,89973	0,90147
1,3	0,90320	0,90490	0,90658	0,90824	0,90988	0,91149	0,91309	0,91466	0,91621	0,91774
1,4	0,91924	0,92073	0,92220	0,92364	0,92507	0,92647	0,92786	0,92922	0,93056	0,93189
1,5	0,93319	0,93448	0,93574	0,93699	0,93822	0,93943	0,94062	0,94179	0,94295	0,94408
1,6	0,94520	0,94630	0,94738	0,94845	0,94950	0,95053	0,95154	0,95254	0,95352	0,95449
1,7	0,95543	0,95637	0,95728	0,95818	0,95907	0,95994	0,96080	0,96164	0,96246	0,96327
1,8	0,96407	0,96485	0,96562	0,96638	0,96712	0,96784	0,96856	0,96926	0,96995	0,97062
1,9	0,97128	0,97193	0,97257	0,97320	0,97381	0,97441	0,97500	0,97558	0,97615	0,97670
2,0	0,97725	0,97778	0,97831	0,97882	0,97932	0,97982	0,98030	0,98077	0,98124	0,98169
2,1	0,98214	0,98257	0,98300	0,98341	0,98382	0,98422	0,98461	0,98500	0,98537	0,98574
2,2	0,98610	0,98645	0,98679	0,98713	0,98745	0,98778	0,98809	0,98840	0,98870	0,98899
2,3	0,98928	0,98956	0,98983	0,99010	0,99036	0,99061	0,99086	0,99111	0,99134	0,99158
2,4	0,99180	0,99202	0,99224	0,99245	0,99266	0,99286	0,99305	0,99324	0,99343	0,99361
2,5	0,99379	0,99396	0,99413	0,99430	0,99446	0,99461	0,99477	0,99492	0,99506	0,99520
2,6	0,99534	0,99547	0,99560	0,99573	0,99585	0,99598	0,99609	0,99621	0,99632	0,99643
2,7	0,99653	0,99664	0,99674	0,99683	0,99693	0,99702	0,99711	0,99720	0,99728	0,99736
2,8	0,99744	0,99752	0,99760	0,99767	0,99774	0,99781	0,99788	0,99795	0,99801	0,99807
2,9	0,99813	0,99819	0,99825	0,99831	0,99836	0,99841	0,99846	0,99851	0,99856	0,99861
3,0	0,99865	0,99869	0,99874	0,99878	0,99882	0,99886	0,99889	0,99893	0,99897	0,99900
3,1	0,99903	0,99906	0,99910	0,99913	0,99916	0,99918	0,99921	0,99924	0,99926	0,99929
3,2	0,99931	0,99934	0,99936	0,99938	0,99940	0,99942	0,99944	0,99946	0,99948	0,99950
3,3	0,99952	0,99953	0,99955	0,99957	0,99958	0,99960	0,99961	0,99962	0,99964	0,99965
3,4	0,99966	0,99968	0,99969	0,99970	0,99971	0,99972	0,99973	0,99974	0,99975	0,99976
3,5	0,99977	0,99978	0,99978	0,99979	0,99980	0,99981	0,99981	0,99982	0,99983	0,99983
3,6	0,99984	0,99985	0,99985	0,99986	0,99986	0,99987	0,99987	0,99988	0,99988	0,99989
3,7	0,99989	0,99990	0,99990	0,99990	0,99991	0,99991	0,99992	0,99992	0,99992	0,99992
3,8	0,99993	0,99993	0,99993	0,99994	0,99994	0,99994	0,99994	0,99995	0,99995	0,99995
3,9	0,99995	0,99995	0,99996	0,99996	0,99996	0,99996	0,99996	0,99996	0,99997	0,99997
4,0	0,99997	0,99997	0,99997	0,99997	0,99997	0,99997	0,99998	0,99998	0,99998	0,99998