# Análise preditiva de dados em uma instituição de abrigo de animais abandonados.

Caroline Lourenço Alves, Rafael Joseph Pagliuca dos Santos, Thiago Henrique Martinelli, and Thomio Watanabe

Instituto de Ciências Matemáticas e de Computação, São Carlos, Brasil

**Resumo** Este trabalho segue as etapas do método KDD para solucionar o problema proposto pelo Austin Animal Center, cujo objetivo é predizer o destino de um animal abrigado pela entidade, a partir de algumas informações básicas sobre ele. Diversos métodos de mineração de dados foram utilizados, destacando-se a utilização do método de classificação **Random Forest** que apresentou melhores resultados. O modelo apresentado foi analisado pelo website Kaggle sendo classificado na posição **875**, grupo Mogi-Bertioga.

**Keywords:** KDD, Random Forests, SVM, MLP, Naive Bayes

## 1 Introdução

Este trabalho apresenta uma metodologia utilizada para resolver o problema de classificação do Austin Animal Center apresentado no website Kaggle[1]. O problema consiste na predição do destino dos animais abrigados pela entidade a partir de algumas informações básicas de cada animal, entre elas nome, raça, cor e idade. Baseando-se nesses dados fornecidos pela entidade, as soluções propostas devem ser capazes de estimar o destino do animal: adoção, morte, eutanásia, retorno ao dono e transferência. Para cada exemplo de teste, o modelo deve prever as probabilidades dentre as cinco possíveis opções de destino.

O Austin Animal Center, um abrigo de animais localizado no estado do Texas, EUA, tornou disponíves diversos dados sobre os animais tratados pela institução. Os animais recebem uma ID única e podem ser classificados em 5 grupos, de acordo com o destino do animal: adoção, morte, eutanásia, retorno ao dono e transferência. Espera-se que seja desenvolvido um método para prever em qual desses grupos cada animal deverá ser classificado, ou seja, qual será o destino dado a ele. Dessa forma será possível para o centro coordenar suas atividades de forma mais eficiente.

O problema proposto foi solucionado seguindo as etapas do fluxo de processos conhecido como Knowledge Discovery in Databases ou KDD, [1]. Os principais resultados estão relacionados às etapas de pré-processamento, transformação e mineração de dados. Quatro classificadores foram utilizados e avaliados. Os resultados gerados pelo melhor classificador foi submetido ao Kaggle obtendo uma classificação de 875, em 24/06/2016.

[1] URL: https://www.kaggle.com/c/shelter-animal-outcomes.

# 2 Metodologia

## 2.1 KDD

O KDD é um método de extração de conhecimento frequentemente confundido com a mineração de dados. Ele permite o desenvolvimento de soluções de forma organizada onde a mineração de dados é uma das etapas do processo, e consiste nas etapas apresentadas na Figura 1.

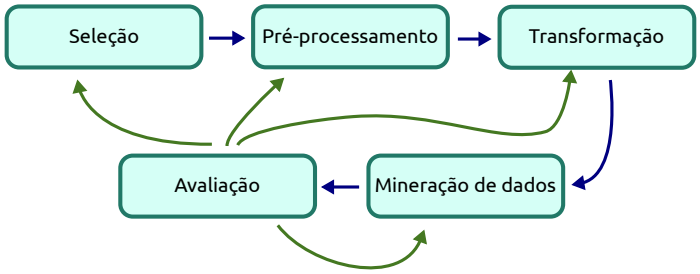


**Figura 1.** Etapas da metodologia KDD. Observa-se a natureza iterativa do processo: ao concluir-se a etapa de Avaliação, é possível reiniciar o processo a partir de qualquer uma das outras etapas.

Os métodos e procedimentos realizados serão explicados nas seções seguintes, organizadas da seguinte maneira:

1. **Análise de dados:** Será descrito como os dados foram previamente analisados de forma a se ter uma visão geral inicial problema.
2. **Preparação de dados:** Nessa seção serão descritas as etapas de pré-processamento e transformação de dados.
3. **Mineração de Dados:** Será apresentada a metodologia utilizada para mineração de dados e também é explicado como foi feita a avaliação dos modelos gerados.

# 3 Análise dos Dados

Previamente à etapa de pré-processamento de dados, foi feita uma análise do banco de dados fornecido pela Austin Animal Center. As informações auferidas são descritas abaixo, organizadas na tabela 1 e ilustradas no gráfico presente na figura 2.

1. Foram disponibilizados dois bancos de dados, um para treinamento e um para testes.
2. A tabela de treinamento possui 26.729 observações e 10 atributos.
3. A tabela de teste possui 11.456 observações e 8 atributos.

4. Devido ao pequeno número de atributos há um grande risco de perda de informação ao se descartar um dos atributos.
5. A maioria dos atributos é categórico.
6. Apenas dois tipos de animais são tratados: cachorros e gatos.
7. Há uma grande quantidade de raças e cores o que pode ser um indício que esses atributos não foram preenchidos de forma padronizada.
8. Os atributos Name e OutcomeSubtype possuem diversos campos não preenchidos.
9. A Tabela 1 apresenta as principais informações obtidas das tabelas.
10. A Figura 2 apresenta a distribuição das classes de saída. Pode-se observar que os dados estão desbalanceados, havendo uma maior quantidade de adoções e transferências.

.

**Tabela 1.** Análise das tabelas de treinamento e teste

| Atributo | Tabela de treinamento | Tabela de teste | Categoria do atributo | Níveis |
|---|---|---|---|---|
| ID | Ausente | Presente | Numérico | - |
| AnimalID | Presente | Ausente | Numérico | - |
| Name | Presente | Presente | Categórico | 6375 |
| DateTime | Presente | Presente | Numérico | - |
| OutcomeType | Presente | Ausente | Categórico | 5 |
| OutcomeSubtype | Presente | Ausente | Categórico | 17 |
| AnimalType | Presente | Presente | Categórico | 2 |
| SexuponOutcome | Presente | Presente | Categórico | 6 |
| AgeuponOutcome | Presente | Presente | Numérico | - |
| Breed | Presente | Presente | Categórico | 1380 |
| Color | Presente | Presente | Categórico | 366 |

**Tabela 2.** Classes do atributo alvo e número de instâncias em cada uma delas

| Classes | Número de instâncias |
|---|---|
| Adoção | 10769 |
| Morte | 197 |
| Eutanásia | 1555 |
| Retorno ao dono | 4786 |
| Transferência | 9422 |

Da análise do atributo alvo, Tabela 2 e Figura 2, observa-se que os dados estão desbalanceados. Há duas grandes classes: Adoption e Transfer e duas clásses minoritárias: Died e Euthanasia.

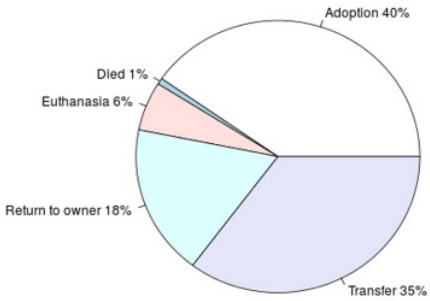


**Figura 2.** Proporção dos grupos de classificação

## 4 Preparação de Dados

Uma das etapas mais importantes do KDD é o pre-processamento dos dados. Esta etapa visa melhorar a qualidade dos dados que serão utilizados pelo classificador.

Como a quantidade de amostras e atributos existentes na tabela foi muito pequeno, evitou-se descartar tanto exemplos quanto atributos. Apenas o atributo **OutcomeSubtype** foi descartado por não estar presente na tabela de teste.

Inicialmente o atributo **Name** não seria considerado, contudo, elaborou-se a hipótese que o nome do animal muitas vezes representa características do mesmo como porte físico e comportamento e que essas características podem definir seu destino.

### 4.1 Atributos Categóricos

Como a maioria dos atributos são categóricos sem haver relação de ordem entre os valores, a codificação *one-hot* foi aplicada para cada um dos atributos categóricos, incluindo o atributo alvo. Essa codificação gera pseudo atributos para cada um dos possíveis valores do atributo original o que, em certos atributos resultou em uma alta dimensionalidade, diminuindo a eficiência dos classificadores, Tabela 3.

**Tabela 3.** Pseudo atributos gerados por cada atributo categórico

| Atributo | Número de pseudo atributos |
|---|---|
| OutcomeType | 5 |
| Name | 6375 |
| AnimalType | 2 |
| SexuponOutcome | 5 |
| Breed | 1381 |
| Color | 367 |

Cada um dos pseudo atributos apresenta zeros e uns (0,1), indicando que a instância pertence ao atributo ou não, Figura 3.

| | A | B | C | D | E |
|---|---|---|---|---|---|
| 1 | Adoption | Died | Euthanasia | Return_to_owner | Transfer |
| 2 | 0.0 | 0.0 | 0.0 | 1.0 | 0.0 |
| 3 | 0.0 | 0.0 | 1.0 | 0.0 | 0.0 |
| 4 | 1.0 | 0.0 | 0.0 | 0.0 | 0.0 |
| 5 | 0.0 | 0.0 | 0.0 | 0.0 | 1.0 |
| 6 | 0.0 | 0.0 | 0.0 | 0.0 | 1.0 |
| 7 | 0.0 | 0.0 | 0.0 | 0.0 | 1.0 |
| 8 | 0.0 | 0.0 | 0.0 | 0.0 | 1.0 |
| 9 | 0.0 | 0.0 | 0.0 | 0.0 | 1.0 |
| 10 | 1.0 | 0.0 | 0.0 | 0.0 | 0.0 |

**Figura 3.** Pseudo atributos gerados a partir do atributo alvo.

A geração de pseudo atributos resulta em um total de 8130 atributos, excluindo os atributos alvo e numéricos. Se todos esses atributos fossem utilizados os dados ficariam muito esparsos, tabela com muitos zeros, e o mal da dimensionalidade afetaria os resultados dos classificadores. Para se evitar esse problema foram selecionados os atributos mais frequêntes entre os pseudo atributos gerados pelos atributos **Name**, **Breed** e **Color**. Em outras palavras, foram selecionados os nomes, raças e cores mais frequentes nos animais. Foram selecionados os 35 nomes mais comuns, as 85 raças mais comuns e as 79 cores mais comuns, totalizando 213 atributos de entrada e removendo 7924 atributos.

A remoção desses pseudo atributos implica em perda de informação. Para se evitar isso, sugere-se como trabalho futuros a utilização de métodos de agrupamento nas raças e cores. Há indícios de que isso poderia melhorar substancialmente os resultados, visto que há pequenas variações e erros de digitação nos campos de raça e cor.

### 4.2 Atributos Numéricos

Apenas os atributos de entrada **DateTime** e **AgeuponOutcome** são numéricos e ambos foram tratados de forma diferente.

Há divergências no significado do atributo **DateTime**. A hipótese mais aceita é que esse campo representa a data de saída do animal do abrigo. Essa informação vem recebendo críticas dos participantes da competição porque ela é altamente correlacionada com o tipo de saída do animal, ou seja, com o atributo alvo. Além disso ela não estaria disponível em uma situação real na qual seria necessário prever o destino do animal. O atributo **DateTime** foi separado em dois, um pseudo atributo com as data e outro com a hora e cada um desses campos foi padronizado.

O atributo **AgeuponOutcome** representa a idade de cada animal quando o mesmo deixa o abrigo. Essa informação também não estaria disponível em

uma situação real pois não se sabe previamente quando o animal vai deixar o abrigo. Esse campo foi preenchido de forma categórica, com números e períodos de tempo como semanas, meses e anos. Ao todo esse atributo foi preenchido com 44 períodos de tempo, onde os mais frequentes são ilustrados na Tabela 4.

**Tabela 4.** Idade dos animais (top 5)

| Idade | Número de animais |
|---|---|
| 1 year | 3969 |
| 2 years | 3742 |
| 2 months | 3397 |
| 3 years | 1823 |
| 1 month | 1281 |

Os 44 períodos de tempo do atributo **AgeuponOutcome** foram transformados em dias e a informação foi ilustrada na Figura 4. Dessas informações pode-se concluir que a maioria dos animais deixa o abrigo com idade entre 1 mês e 3 anos. A informação da idade foi agrupada em 8 intervalos de tempos.

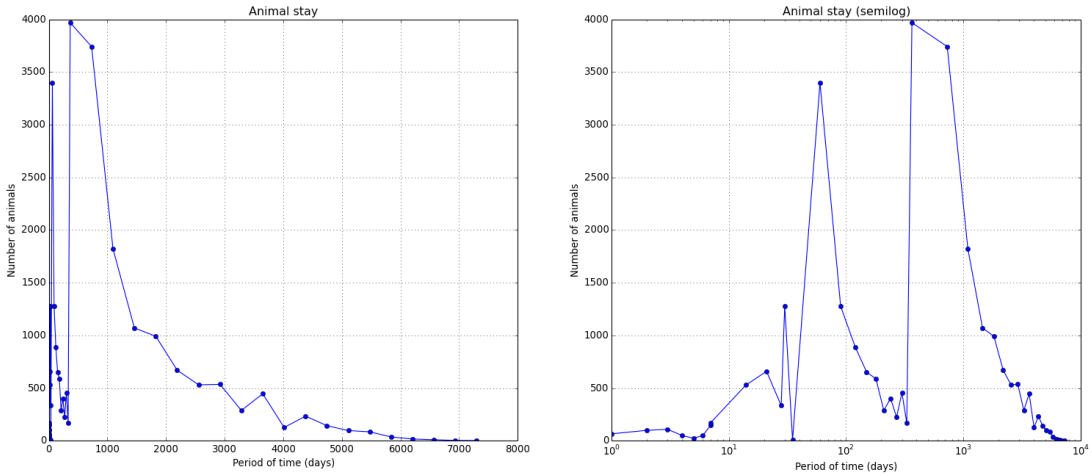


**Figura 4.** Idade dos animais do abrigo.

## 5 Mineração de Dados

Devido ao fato do atributo alvo ser categórico e de haver um grande desbalanceamento dos dados, decidiu-se gerar um modelo para cada uma das classes, totalizando 5 modelos.

Também decidiu-se avaliar o desempenho de 4 classificadores diferentes para saber qual se ajusta melhor ao problema. Os classificadores selecionados foram o Random Forest, o SVM, as Redes Neurais (multi layer perceptron) e o Naive Bayes.

### 5.1 Classificadores

O Random Forest é um método de combinação entre classificadores, no caso, árvores de decisão [2], [3], [4]. Árvores de decisão particiona um conjunto de dados heterogêneo (raiz) em classes homogêneas (folhas), gerando regras de classificação com base em atributos (nós). O critério para a partição dos dados é baseado no ganho de informação que, para classificação, é proveniente da diminuição entropia do conjunto de dados quando submetido à divisão de acordo com um atributo.

As máquinas de vetores de suporte (Support Vector Machines – SVM), são a aplicação dos resultados obtidos com a Teoria da Aprendizagem Estatística. Elas podem ser utilizadas para solucionar problemas de classificação linearmente separáveis, ou seja, que podem ser resolvidos por meio da utilização de um hiperplano marginal que divide os dados em suas respectivas classes.

Uma rede neural artificial, ANN ou MLP, é composta por várias unidades de processamento, cujo funcionamento é bastante simples. Essas unidades, geralmente são conectadas por canais de comunicação que estão associados a determinado peso. As unidades fazem operações apenas sobre seus dados locais, que são entradas recebidas pelas suas conexões. O comportamento inteligente de uma Rede Neural Artificial vem das interações entre as unidades de processamento da rede.

O Naive Bayes é um classificador probabilístico simples baseado na aplicação do Teorema de Bayes (baseado na estatística de Bayes), com fortes suposições de independência.

Para aumentar a confiabilidade do modelo utilizou-se a validação cruzada que será descrito na secção seguinte.

### 5.2 Validação Cruzada

Na Validação Cruzada o conjunto de exemplos é divididos em “r” subconjuntos de tamanho aproximadamente iguais, [5]. Objetos de “r-1” partições são usados para o conjunto treinamento de um preditor o qual é testado na partição restante. O processo é repetido “r” vezes , utilizando em cada ciclo uma partição diferente para testes. O desempenho final é dado pela média dos desempenhos observados para cada conjunto teste.

Uma variação deste modelo é p r-fold cross validation estratificado , que mantém em cada partição a proporção de exemplos de cada classe semelhante a proporção contida no conjunto de dados totais.

No caso se “r= n” em que n representa o número de casos possíveis tem-se o método leave one out (caso particular da validação cruzada) . Este modelo

estima de forma mais fiel o desempenho preditivo do modelo , contudo computacionalmente ele é mais caro.

A principal crítica a Validação Cruzada é que parte dos dados é compartilhada entre os subconjuntos de treinamento.

### 5.3 Avaliação

A qualidade do modelo gerado foi avaliada pela acurácia e pela perda logarítmica.

A perda logarítmica multi-classe foi utilizado pelo Kaggle para classificar os resultados. No presente trabalho a perda foi avaliada para cada uma das classes o que justifica a diferença entre os resultados do trabalho e do Kaggle, aproximadamente 1 ponto.

O log loss é utilizado para avaliar predições onde um resultado pode ser verdadeiro ou falso. A perda logarítmica é definida pela equação 1.

$$logloss = -\frac{1}{N}\sum_{i=1}^{N}(y_i log(p_i) + (1 - y_i) log(1 - p_i)) \quad (1)$$

### 5.4 Treinamento e Teste

A entidade Austin Animal Center forneceu duas tabelas de dados, sendo uma tabela para treinamento: **train.csv**, e uma tabela para teste: **test.csv**. Em ambas, os dados estão organizados no formato *comma-separated values* que separa as linhas por um *carriage return* e os valores das linhas por vírgula. A tabela **train.csv** foi utilizada para treinamento e teste. O melhor classificador gerado foi aplicado na tabela de **test.csv** e o resultado submetido ao Kaggle.

Após a etapa de preparação dos dados, os dados foram utilizados para treinar os 4 classificadores escolhidos. O treinamento utilizou o método de validação cruzada com 10 subconjuntos, 10-fold cross validation. E foram gerados um modelo para cada uma das classes: **Adoption**, **Died**, **Euthanasia**, **Return to owner** e **Transfer**.

Devido ao forte desbalanceamento das classes **Died** e **Euthanasia**, essas duas classes passaram por um processo de sub-amostragem para evitar o superajuste. Em ambas, foi selecionado o subconjunto com as instâncias verdadeiras e foram adicionadas a mesma quantidade de instâncias falsas, totalizando aproximadamente 400 instâncias para **Died** e 2000 instâncias para **Euthanasia**. As outras classes foram testadas com todo o conjunto de amostras.

A tabela 5 ilustra as acurácias obtidas para cada classificador e classe, já a tabela 6 ilustra as perdas. As acurácias foram obtidas da média de cada conjunto de amostras da validação cruzada.

Da análise das tabelas 5 e 6 observa-se que o melhor resultado foi apresentado pelo método Random Forest, tanto em acurácia quanto em perda logarítmica. Os métodos SVM e MLP apresentaram resultados próximos ao Random Forest.

O pior resultado foi apresentado pelo classificador Naive Bayes. Supõe-se que isso se deve a presença de atributos de entrada altamente correlacionados entre

**Tabela 5.** Acurácia para cada classe

| Classificador | Random Forest | SVM | MLP | Naive Bayes |
|---|---|---|---|---|
| Adoption | 0.754 | 0.743 | 0.741 | 0.452 |
| Died | 0.678 | 0.698 | 0.676 | 0.622 |
| Euthanasia | 0.749 | 0.721 | 0.693 | 0.431 |
| Return to owner | 0.812 | 0.809 | 0.763 | 0.290 |
| Transfer | 0.789 | 0.780 | 0.762 | 0.403 |

**Tabela 6.** Perda logarítmica para cada classe

| Classificador | Random Forest | SVM | MLP | Naive Bayes |
|---|---|---|---|---|
| Adoption | 1.078 | 7.996 | 7.928 | 11.019 |
| Died | 0.935 | 5.844 | 3.507 | 16.481 |
| Euthanasia | 0.829 | 7.848 | 7.056 | 20.605 |
| Return to owner | 0.815 | 5.909 | 5.918 | 14.695 |
| Transfer | 0.973 | 7.098 | 6.926 | 16.786 |

si, principalmente devido à codificação *one-hot*. O que vai de contra um dos fundamentos do método que é a independência entre os atributos. Além disso, há fortes indícios que os atributos DateTime e SexuponOutcome são altamente correlacionados com os atributos alvo.

## 6 Conclusão

O trabalho apresentado seguiu as etapas do KDD e foi capaz de gerar os resultados esperados na proposta do problema.

Os resultados obtidos pelo melhor classificador, o Random Forest foram submetidos ao website do Kaggle e foram classificados na posição 875 (2.14512 log loss), dia 24/06/2016, grupo Mogi-Bertioga.

A classificação obtida pelo resultado evidencia que os métodos escolhidos foram aplicados de forma correta. Espera-se que melhores resultados possam ser obtidos com as melhorias propostas na seção de trabalho futuros.

Para haver reprodutividade dos resultados descritos nesse trabalho os códigos desenvolvidos foram disponibilizados no github[2].

### 6.1 Trabalhos Futuros

A principal melhoria que pode ser feita neste trabalho é a utilização de um método de agrupamento nas classes *Breed* e *Color*. Como descrito anteriormente na secção de preparação de dados, apenas os principais valores foram utilizados para o treinamento e generalização, e os valores menos frequêntes foram descartados, acarretando em perda de informação.

[2] https://github.com/Mogi-Bertioga/shelter-animal-outcomes.

Uma outra melhoria pode ser feita no classificador Naive Bayes. Como esse classificador apresenta deficiências frente a atributos altamente correlacionados pode-se utilizar estratégias de combinação de atributos para sua melhoraria [6].

## Referências


1. Fayyad, U., Piatetsky-Shapiro, G., Smyth, P.: From data mining to knowledge discovery in databases. AI magazine **17**(3) (1996) 37
2. Ho, T.K.: Random decision forests. In: Document Analysis and Recognition, 1995., Proceedings of the Third International Conference on. Volume 1. (Aug 1995) 278–282 vol.1
3. Ho, T.K.: The random subspace method for constructing decision forests. IEEE Transactions on Pattern Analysis and Machine Intelligence **20**(8) (Aug 1998) 832–844
4. Breiman, L.: Random forests. Machine Learning **45**(1) (Oct. 2001) 5–32
5. Faceli, K., Lorena, A.C., Gama, J., Carvalho, A.C.: Inteligência Artificial - Uma abordagem de aprendizado de máquina. gen LTC (2011)
6. de Pina, A.C., Zaverucha, G.: Combining attributes to improve the performance of naive bayes for regression. In: 2008 IEEE International Joint Conference on Neural Networks (IEEE World Congress on Computational Intelligence). (June 2008) 3210–3215