CENTRO FEDERAL DE EDUCAÇÃO TECNOLÓGICA
CELSO SUCKOW DA FONSECA

Uma avaliação experimental de métodos de pré-processamento para identificação de atrasos aéreos

Christofer Marinho Raquel Dantas
Leonardo Castelo Branco Oliveira

Prof. Orientador:
Eduardo Soares Ogasawara, D.Sc.

Rio de Janeiro,
Setembro de 2017

CENTRO FEDERAL DE EDUCAÇÃO TECNOLÓGICA
CELSO SUCKOW DA FONSECA

Uma avaliação experimental de métodos de pré-processamento para identificação de atrasos aéreos

Christofer Marinho Raquel Dantas
Leonardo Castelo Branco Oliveira

Projeto final apresentado em cumprimento às normas do Departamento de Educação Superior do Centro Federal de Educação Tecnológica Celso Suckow da Fonseca, CEFET/RJ, como parte dos requisitos para obtenção do título de Bacharel em Ciência da Computação.

Prof. Orientador:
Eduardo Soares Ogasawara, D.Sc.

Rio de Janeiro,
Setembro de 2017

Ficha catalográfica elaborada pela Biblioteca Central do CEFET/RJ

D192 Dantas, Christofer Marinho Raquel
Uma avaliação experimental de métodos de pré-processamento para identificação de atrasos aéreos/ Christofer Marinho Raquel Dantas, Leonardo Castelo Branco Oliveira.—2017.
xii, 44f. : il. (algumas color.) , grafs. , tabs. ; enc.

Projeto Final (Graduação) Centro Federal de Educação Tecnológica Celso Suckow da Fonseca , 2017.
Bibliografia : f. 41-44
Orientador : Eduardo Soares Ogasawara

1. Ciência da computação. 2. Aprendizado de máquina. 3. Mineração de dados. I. Oliveira, Leonardo Castelo Branco. II. Ogasawara, Eduardo Soares (Orient.). III. Título.

CDD 004

DEDICATÓRIA

A Deus e à minha estimada família que me ajudaram, guiaram e deram suporte ao longo de toda a minha trajetória.
Christofer Marinho Raquel Dantas

A todos os meus professores e familiares que me ensinaram, guiaram e deram suporte ao longo de toda a minha vida.
Leonardo Castelo Branco Oliveira

AGRADECIMENTOS

Agradece-se as contribuições de Alice Sternberg, que realizou a pesquisa de Análise de Padrões Frequentes de Atrasos em Voos Nacionais.

Agradece-se também as contribuições de Breno Trotta, que iniciou e disponibilizou a pesquisa sobre o tema abordado.

Por fim, agradece-se as contribuições de Jônatas Coelho, que realizou a integração dos dados meteorológicos juntamente aos dados aéreos.

RESUMO

Atrasos aéreos causam diversos transtornos para as companhias aéreas, aeroportos e passageiros. Segundo os dados disponibilizados pela Agência Nacional de Aviação Civil (ANAC), entre 2009 e 2015, cerca de 22% dos voos domésticos, realizados no Brasil, apresentaram atraso acima de 15 minutos. A previsão desses atrasos é fundamental para mitigar sua ocorrência e otimizar o processo de tomada de decisão de um sistema de transporte aéreo. No entanto, a falta de modelos precisos de previsão faz com que muitas decisões tomadas por companhias aéreas, aeroportos e investidores não levem em consideração todos os fatores associados aos atrasos. Neste contexto, este trabalho realiza uma avaliação experimental de diversos métodos de pré-processamento para geração de modelos de previsão baseados em aprendizado de máquina, aplicados sobre um *data warehouse* integrado, contendo informações de operações de voo e condições meteorológicas. Uma avaliação da literatura relacionada ao processo de mineração de dados e criação de modelos de previsão foi realizada, resultando num *workflow* que define a metodologia aplicada neste trabalho, as etapas de pré-processamento e o desenvolvimento dos modelos de previsão de atrasos aéreos.

Palavras-chave: Atrasos aéreos; Pré-processamento; Aprendizado de máquina; Modelos de previsão

ABSTRACT

Flight delays causes various inconveniences to the airlines, airports and passengers. According to the data provided by National Civil Aviation Agency (ANAC), between 2009 and 2015, about 22% of the domestic flights, performed in Brazil, were delayed more than 15 minutes. The forecast of these delays is fundamental to mitigate its occurrence and optimize the decision-making process of an air transportation system. However, the lack of accurate forecasting models forces the airlines, airports and investors to not consider all factors associated with delays. In this context, this research performs an experimental evaluation of several pre-processing methods in order to generate forecasting models based on machine learning, applied over an integrated data warehouse, that contains flight operations information and meteorological conditions. An evaluation of the literature related to the process of data mining and creation of forecast models was performed, resulting in a workflow that defines the methodology applied in this work, the pre-processing steps and the development of flight delay forecasting models.
Keywords: Flight delays; Pre-processing; Machine learning; Forecasting Models

SUMÁRIO

Lista de Figuras

LISTA DE TABELAS

LISTA DE ABREVIAÇÕES

ANAC	Agência Nacional De Aviação Civil
CFS	Seleção De Atributos Baseada Em Correlação
ETL	Extração, Transformação E Carga
GSPS-CI	Resolução De Problemas De Sistemas Gerais Integrando Técnicas De Inteligência Computacional
INFOGAIN	Ganho De Informação
KNN	K-Vizinhos Mais Próximos
LASSO	Encolhimento Mínimo Absoluto E Operador De Seleção
MMH	Margem Máxima Do Hiperplano
NB	Classificadores Bayesianos Ingênuos
PCA	Análise De Componentes Principais
RF	Florestas Aleatórias
RN	Redes Neurais
RU	Subamostragem Aleatória
SMOTE	Técnica De Sobreamostragem Minoritária Sintética
SVM	Máquina De Vetores De Suporte
VRA	Voo Regular Ativo
WU	Weather Underground

Capítulo 1

Introdução

O atraso é um dos principais indicadores de desempenho de qualquer sistema de transporte. Contudo, no cenário da aviação comercial, eles têm um alto impacto financeiro para as companhias aéreas, como multas, custos de operação adicionais e ainda a queda da fidelidade dos clientes. Além disso, dada a incerteza da sua ocorrência, muitos passageiros são forçados a replanejar suas viagens a fim de chegar no destino a tempo, o que muitas vezes leva ao aumento do custo da viagem [Britto et al., 2012]. Sendo assim, métodos de previsão de atrasos aéreos são fundamentais para mitigar sua ocorrência, e como consequência reduzir os custos gerados.

Um atraso deve ser representado pela diferença entre o tempo programado e o tempo real de partida ou chegada de um voo [Wieland, 1997]. No contexto da aviação comercial esses atrasos podem ocorrer por diversos motivos, dentre eles, falhas no processo de voo, condições meteorológicas, problemas mecânicos, atrasos no solo, controle de tráfego aéreo e restrições de capacidade. Portanto, métodos de previsão são necessários, dada a complexidade dos motivos e condições que geram atrasos.

Um grande volume de dados tem sido coletado em bancos de dados de instituições públicas e privadas com o objetivo de estudar e compreender as operações do sistema de transporte aéreo. A análise dessa grande quantidade de dados, como um problema de *big data*, permite-nos obter o conhecimento necessário para detectar e prever os atrasos. Nesse contexto, existem diversas análises, que envolvem o entendimento do domínio, da relação entre os dados, além da aplicação de modelos para resolver o problema [Sternberg et al., 2016; Dhar, 2013; Jagadish et al., 2014; Matsudaira, 2015].

Embora existam bases de dados públicas e regulamentações quanto divulgação de casos de atrasos e cancelamentos, são poucos os trabalhos que visam análise das condições que geram atrasos e cancelamentos no que tange a ciência de dados. A falta de trabalhos na área faz com que muitas companhias aéreas, aeroportos e investidores tomem decisões que podem não considerar todos os fatores associados aos atrasos e cancelamentos [Sternberg et al., 2016]. Neste cenário, este trabalho tem como objetivo realizar uma avaliação experimental de métodos de pré-processamento de dados com o objetivo de otimizar a precisão dos modelos de previsão

de atrasos, considerando todos os fatores envolvidos e coletados pelas bases de dados.

Este trabalho usa como base um *data warehouse* desenvolvido em [Sternberg et al., 2016] e construído em PostgreSQL [PostgreSQL Global Development Group, 2015], que integra uma base de dados contendo informações das operações de voo [ANAC, 2015], e uma segunda base possuindo informações sobre as condições meteorológicas [The Weather Company, 2016]. Com base nesse *data warehouse*, serão realizadas as etapas de pré-processamento, além da construção de modelos de previsão baseados em aprendizado de máquina.

O pré-processamento é fundamental para refinar os dados e tornar os modelos de previsão mais eficientes. Embora existam estudos e modelos de previsão de atrasos aéreos como os desenvolvidos em Lu Zonglei [2008], Chen et al. [2008] e Rebollo and Balakrishnan [2014], são poucos os que levam em consideração dados meteorológicos e utilizam o pré-processamento de dados antes da aplicação do modelo de aprendizado de máquina, como em Belcastro et al. [2016]. Sendo assim, este trabalho contribui explorando diversos métodos de pré-processamento a fim de produzir modelos de previsão mais precisos. Após a exploração dos métodos de pré-processamento, este trabalho ainda realiza a avaliação da precisão dos diferentes modelos de aprendizado de máquina supervisionados aplicados sobre o *dataset* refinado.

Além dessa introdução, este trabalho está estruturado da seguinte forma. Os Capítulos 2 e 3 são referentes a fundamentação teórica. O Capítulo 2 introduz o conceito de pré-processamento de dados e apresenta os métodos utilizados no trabalho para refinar e preparar o *dataset*, assim como uma análise qualitativa de cada método. No Capítulo 3, são apresentados os principais conceitos de aprendizado de máquina, assim como os modelos supervisionados utilizados neste trabalho. São eles: Redes Neurais (RN), Máquina de Vetores de Suporte (SVM), k-Vizinhos mais próximos (kNN), Classificador Bayesiano Ingênuo (NB) e Florestas Aleatórias (RF). O Capítulo 4 apresenta os trabalhos relacionados que utilizam modelos de aprendizado de máquina para prever e classificar os atrasos aéreos. O Capítulo 5 discute a metodologia utilizada na realização do trabalho, apresentando cada etapa de pré-processamento aplicada ao *dataset* e o *workflow* realizado para a criação, treinamento, teste e avaliação dos modelos de aprendizado de máquina. O Capítulo 6 analisa os resultados encontrados a partir da execução do *workflow*. Finalmente, o Capítulo 7 encerra a dissertação desse trabalho.

Capítulo 2

Pré-processamento de Dados

A fase de pré-processamento é fundamental para tratar fatores acerca da qualidade dos dados. Além disso, essa qualidade é totalmente relativa ao propósito para o qual os dados serão usados. Portanto, alguns dos fatores de qualidade como precisão, completude, consistência e credibilidade são cruciais para o desenvolvimento de um modelo de previsão eficiente. O pré-processamento de dados pode ser dividido em cinco etapas principais, sendo elas, Integração (Seção 2.1), Limpeza (Seção 2.2), Redução (Seção 2.3), Transformação (Seção 2.4) e Balanceamento (Seção 2.5) [Han et al., 2011]. A Figura 1 ilustra as principais etapas do pré-processamento de dados.

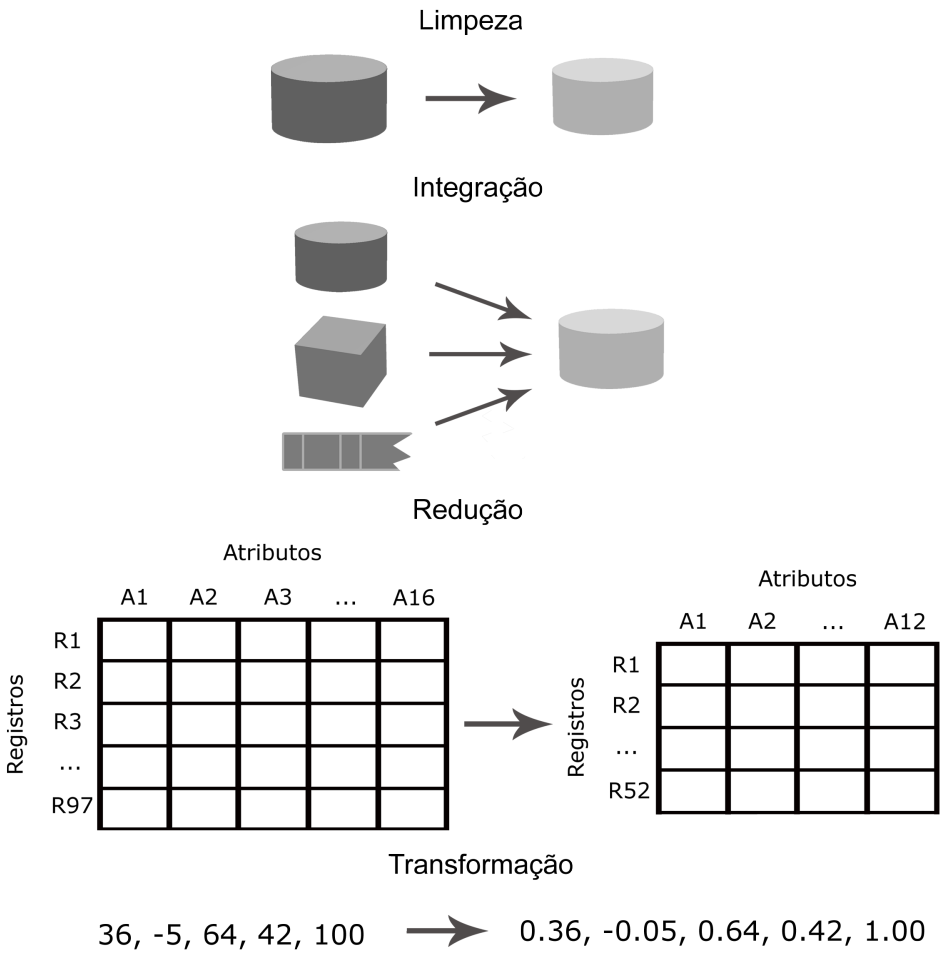

Figura 1: Principais etapas do pré-processamento, adaptado de Han et al. [2011]

A partir do uso de algumas técnicas de pré-processamento presentes nessas etapas é possível melhorar a qualidade dos dados e prepará-los para utilizar e otimizar a aplicação de métodos de aprendizado de máquina.

2.1 Integração de Dados

A etapa de Integração permite a integração de diferentes bases de dados a fim de criar uma base de dados unificada para uma análise completa de todos os dados envolvidos [Han et al., 2011]. A base de dados utilizada neste trabalho é um *data warehouse* desenvolvido em Sternberg et al. [2016], construído em PostgreSQL [PostgreSQL Global Development Group, 2015], a partir da integração de uma base de dados com informações de operações de voos [ANAC, 2015], e uma segunda base de dados pública de condições meteorológicas, chamada *Weather Underground (WU)* [The Weather Company, 2016].

O *data warehouse* utilizado neste trabalho foi construído em Sternberg et al. [2016] para a análise de padrões fequentes, utilizando um processo chamado de Extração, Transformação e Carga (ETL). Este processo é muito utilizado quando há a necessidade de realizar a integração de múltiplas fontes de dados [Han et al., 2011]. O processo começa com a extração de dados de diferentes fontes identificadas para o problema. Em seguida, os dados extraídos são tratados para evitar redundâncias e inconsistências, e depois carregados para formar o *data warehouse*.

2.2 Limpeza de Dados

Os dados coletados no dia a dia tendem a ser incompletos, possuir ruídos e inconsistências. No entanto, quando o objetivo é utilizar os dados para gerar modelos de classificação, é fundamental que os dados estejam completos, corretos e compatíveis com a realidade, para evitar que o desempenho do classificador seja afetado negativamente. Assim, a etapa de Limpeza permite que se faça um tratamento nos dados, seja identificando e removendo outliers, suavizando dados ruidosos ou preenchendo valores perdidos [Han et al., 2011].

2.3 Redução de Dados

A etapa de Redução é capaz de criar uma representação reduzida do conjunto de dados e ainda assim produzir o mesmo resultado analítico. Um conjunto de dados a ser analisado pode

conter diversos atributos. Contudo, uma parte deles podem ser irrelevantes durante o processo de mineração, ou até redundantes [Han et al., 2011]. Por exemplo, se queremos classificar as empresas de linhas aéreas baseado nos atrasos aéreos, atributos como o número de voo tendem a ser irrelevantes, diferentemente do tempo de atraso de partida ou o tempo de partida real, que são atributos que podem agregar valor à análise.

A seleção de atributos visa reduzir o conjunto de dados por meio da remoção destes atributos irrelevantes ou redundantes, com o objetivo de encontrar um conjunto mínimo de atributos que possua a distribuição de probabilidade que melhor represente a distribuição original. Com isso, o número de atributos é reduzido, melhorando a performance do algoritmo de mineração aplicado[Han et al., 2011]. As estratégias de redução de dados utilizadas neste estudo são Encolhimento Mínimo Absoluto e Operador de Seleção (LASSO) (Seção 2.3.1), Ganho de Informação (Seção 2.3.2), Seleção de Atributos baseada em Correlação (CFS) (Seção 2.3.3), Análise de Componentes Principais (PCA) (Seção 2.3.4) e Amostragem (Seção 2.3.5).

2.3.1 Encolhimento Mínimo Absoluto e Operador de Seleção

O Encolhimento Mínimo Absoluto e Operador de Seleção (LASSO) é um método de regressão que envolve penalizar o tamanho absoluto dos coeficientes de regressão. A penalização faz com que algumas estimativas dos coeficientes sejam exatamente zero. Quanto maior a penalidade aplicada, maior será a quantidade de coeficientes zerados [Hastie et al., 2009].

Ao identificar um pequeno número de indicadores em que um modelo de confiança pode ser construído, o LASSO aborda dois pontos importantes que a regressão linear sofre de acordo com o aumento do número de preditores: o *overfitting*, e a interpretação do modelo ajustado.

2.3.2 Ganho de Informação

O Ganho de Informação (INFOGAIN) é um método que avalia, individualmente, o ganho de informação de cada atributo. Ele é definido como a diferença da entropia antes e após a distribuição dos dados, a partir de um determinado atributo. Sendo assim, ele nos informa o quanto é ganho ao se obter o valor de um determinado atributo.

O uso deste método possibilita a obtenção de atributos que minimizam a quantidade de informação necessária para a classificação dos dados. Ele é usado para selecionar os atributos mais influentes, i.e, aqueles que possuem menor entropia. Além disso, ele nos permite tratar

valores faltantes separadamente, ou distribuir as contagens entre si, proporcionalmente à sua frequência [Witten et al., 2011].

2.3.3 Seleção de Atributos baseada em Correlação

A Seleção de Atributos baseada em Correlação (CFS) é um algoritmo simples de filtro que classifica subconjuntos de atributos de acordo com uma função de avaliação heurística baseada em correlação. O *bias* dessa função é para subconjuntos que contêm atributos que são altamente correlacionados com a classe e não correlacionadas entre si [Hall, 1998].

Os atributos irrelevantes devem ser ignorados, porque possuirão baixa correlação com a classe. Os atributos redundantes devem ser prevenidos, já que eles serão altamente correlacionados com um ou mais atributos restantes. A aceitação de um atributo dependerá da extensão em que ele prevê as classes em áreas do espaço não já previstos por outros atributos [Hall, 1998].

2.3.4 Análise de Componentes Principais

A PCA é um algoritmo matemático que reduz a dimensionalidade dos dados, preservando a maior parte da variação no *dataset* [Jolliffe, 2002]. O PCA combina a essência dos atributos por meio da criação de um conjunto menor de variáveis [Han et al., 2011]. E, a partir de pontos no espaço n-dimensional, ele apresenta padrões de similaridade entre as observações e as variáveis.

Seu objetivo é extrair as informações importantes dos dados e expressá-las como um conjunto de novas variáveis ortonormais, chamadas de componentes principais. Para isso, é realizado o seguinte processo: (i) a normalização dos dados de entrada; (ii) o cálculo dos componentes principais; (iii) a ordenação dos componentes principais por ordem decrescente de significância ou força; (iv) a redução do tamanho dos dados a partir da eliminação dos componentes mais fracos, i. e., aqueles com menor variância [Han et al., 2011].

Por conveniência, consideremos um conjunto de dados em duas dimensões. Como é apresentado na Figura 2, este conjunto pode ser representado como pontos num plano. O PCA calcula os componentes principais, num novo sistema de coordenadas, disponibilizando assim informações sobre a variância. Com isso, encontramos os dois primeiros componentes principais, *y1* e *y2*, que auxiliam na identificação de grupos ou padrões contidos nos dados.

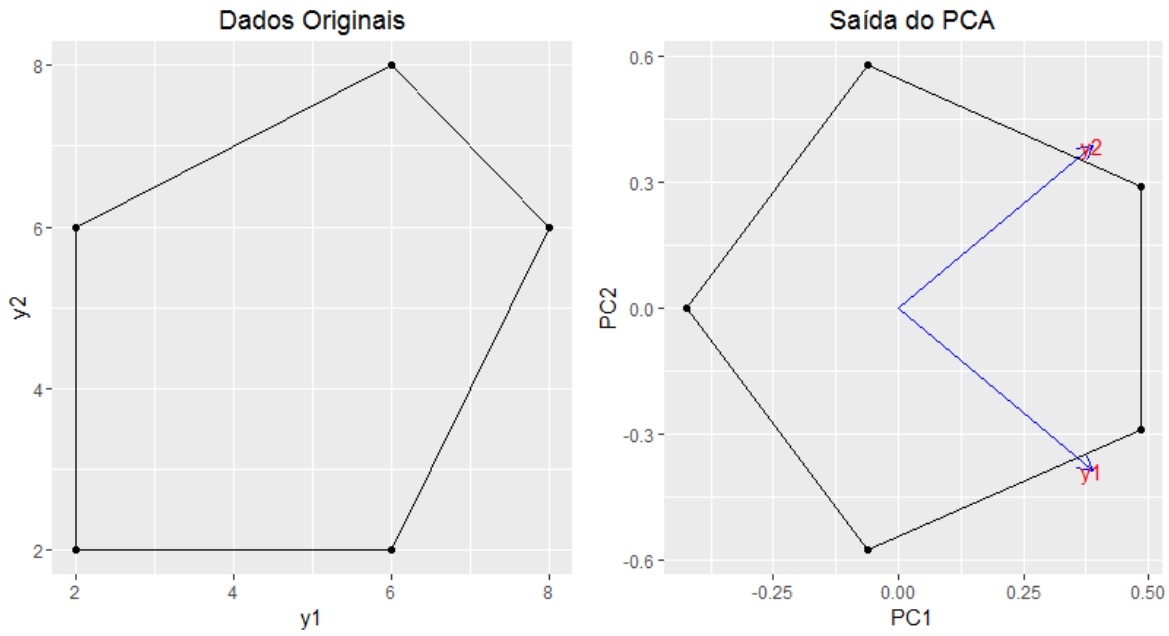

Figura 2: Exemplo de utilização do algoritmo PCA

Os resultados do PCA dependem fortemente do pré-processamento dos dados e da seleção das variáveis. Sendo assim, a análise dos gráficos do PCA pode fornecer conhecimentos sobre diferentes opções de pré-processamento e seleção de variáveis, além de poder servir como um passo importante antes da clusterização ou classificação de amostras [Ringnér, 2008].

2.3.5 Amostragem

A Amostragem pode ser usada como uma técnica de redução de dados porque nos permite representar um grande conjunto de dados por meio de uma amostra de dados aleatórios, ou subconjuntos, muito menores [Han et al., 2011]. O modo como a amostra é realizada depende do tipo de abordagem adotada. Os tipos de amostragem utilizados neste estudo são Amostragem Aleatória e Amostragem Estratificada.

A Amostragem Aleatória consiste em criar um subconjunto onde cada tupla pertencente a um conjunto de dados possui a mesma probabilidade de ser selecionada para compô-lo. A Amostragem Estratificada consiste em separar o conjunto de dados em partes mutuamente disjuntas, denominadas estratos, para então extrair uma amostra de cada estrato gerado [Han et al., 2011]. Sendo assim, a Amostragem Estratificada cria um conjunto reduzido dos dados que tenta manter a mesma proporção entre as classes existentes no conjunto de dados original.

2.4 Transformação de Dados

A etapa de Transformação é responsável por transformar e consolidar os dados em um formato apropriado para tornar o processo de mineração de dados mais eficiente e facilitar o entendimento de padrões nos dados [Han et al., 2011]. As estratégias de transformação de dados utilizadas neste estudo são Normalização Min-Max (Seção 2.4.1), Hierarquia Conceitual (Seção 2.4.2) e Alisamento (Seção 2.4.3).

2.4.1 Normalização Min-Max

A normalização transforma a escala dos valores de um atributo para que se enquadrem em um novo intervalo. Por exemplo, como a unidade de medida utilizada pode afetar a análise dos dados, alterar a unidade de medida de quilómetros para milhas pode levar a diferentes resultados. Portanto, para evitar utilizar unidade de medidas, os dados devem ser normalizados [Han et al., 2011].

Essa normalização é realizada transformando os dados para que ocupem um intervalo menor, geralmente, como [0.0, 1.0] ou $[-1.0, 1.0]$. A normalização dos dados é muito importante para algoritmos de classificação como redes neurais ou classificação por k-vizinhos mais próximos, já que aumenta a velocidade da fase de aprendizado e previne que atributos com valores iniciais muito grandes, como renda, sobreponham atributos com valores iniciais pequenos, como atributos binários [Han et al., 2011].

A normalização Min-Max é um dos métodos de normalização que aplica uma transformação linear nos dados originais, onde o valor mínimo, min_A, e o valor máximo, max_A, são utilizados para transformar cada valor v_i de um atributo A para um valor v'_i, no novo intervalo $[novoMin_A, novoMax_A]$, como mostra a Equação 2.1 a seguir [Han et al., 2011].

$$v'_i = \frac{v_i - Min_A}{Max_A - Min_A} \cdot (NovoMax_A - NovoMin_A) + NovoMin_A \tag{2.1}$$

2.4.2 Hierarquia Conceitual

A Hierarquia Conceitual é uma técnica de pré-processamento da etapa de Transformação. Entretanto, como em diferentes técnicas de pré-processamento, essa não se enquadra em apenas uma etapa, podendo ser também uma técnica da etapa de Redução, já que os dados originais são

substituídos por um menor número de intervalos e conceitos que os representam. Essa técnica, assim como outras técnicas da etapa de Transformação, simplifica os dados originais e tornam o processo de mineração de dados mais eficiente [Han et al., 2011].

A técnica de Hierarquia Conceitual consiste em realizar uma discretização nos dados substituindo dados detalhados por um conceito superior. Por exemplo, a partir de um atributo como rua é possível obter informações, como a cidade e o país correspondentes. Portanto, é possível representar valores de diferentes ruas apenas com sua cidade ou o país.

2.4.3 Alisamento

A técnica de Alisamento é utilizada para corrigir ruídos nos dados, que podem ser gerados por algum erro aleatório ou uma variação incomum obtida na medição de uma variável. Os métodos de alisamento suavizam os ruídos de uma amostra de dados consultando os valores mais próximos e os distribuindo em um determinado número de "baldes"ou caixas. Como os métodos de Alisamento consultam os valores vizinhos dos valores ruidosos, eles fazem uma suavização local nos dados.

Os dados são separados, e divididos em baldes de mesmo tamanho. O Alisamento pode ser realizado de diversas formas, pela média, mediana ou limites do balde. No alisamento por média, cada valor do balde é substituído pela média dos valores do balde, sendo assim, um balde com os valores 3, 5, 13, tem média 7. Em seguida, cada valor original do balde é substituído pelo valor da média obtida.

Analogamente, no alisamento por mediana, cada valor é substituído pelo mediana do balde. O alisamento pelos limites do balde é realizada utilizando seus valores mínimo e máximo. Os valores do balde então são substituídos pelo valor do limite mais próximo [Han et al., 2011]. A Figura 3 exemplica as diferentes formas de alisamento.

Amostra de dados: 3, 5, 13, 15, 16, 20, 21, 24, 30
Amostra dividida em baldes de mesmo tamanho:
Balde 1: 3, 5, 13
Balde 2: 15, 16, 20
Balde 3: 21, 24, 30

Alisamento por média:
Balde 1: 7, 7, 7
Balde 2: 17, 17, 17
Balde 3: 25, 25, 25

Alisamento por mediana:
Balde 1: 5, 5, 5
Balde 2: 16, 16, 16
Balde 3: 24, 24, 24

Alisamento por limites:
Balde 1: 3, 3, 13
Balde 2: 15, 15, 20
Balde 3: 21, 21, 30

Figura 3: Exemplo de utilização da técnica de Alisamento, adaptado de Han et al. [2011]

2.5 Balanceamento de Dados

Um problema muito frequente na mineração de dados é a distribuição das classes no *dataset*, já que uma má distribuição pode induzir o resultado dos classificadores. Em diversas aplicações, o número de registros de uma determinada classe é muito maior do que o número de registros pertencentes a outra [Prati et al., 2009]. Alguns exemplos são, a detecção de fraudes em cartão de crédito, em que a quantidade de operações fraudulentas é muito menor que a quantidade de operações legais, e os atrasos aéreos, em que apenas cerca de 25% dos voos apresentam atraso maior que 15 minutos.

A amostragem é uma abordagem direta para o problema de balanceamento de classes em um *dataset*. A partir da utilização de métodos de balanceamento é possível alterar a distribuição das classes, com o objetivo de obter uma distribuição mais balanceada dos dados e aprimorar o desempenho dos modelos de classificação de dados [Prati et al., 2009]. As estratégias de balanceamento de dados utilizadas neste estudo são Subamostragem Aleatória (Seção 2.5.1) e a Técnica de Sobreamostragem Minoritária Sintética (SMOTE) (Seção 2.5.2).

2.5.1 Subamostragem Aleatória

A RU é um método não-heurístico que tem como objetivo balancear a distribuição de classes nos dados a partir de uma eliminação aleatória das tuplas da classe majoritária, ou seja, a classe com maior frequência no conjunto de dados original [Prati et al., 2009]. Essa eliminação aleatória pode gerar uma perda de informação sobre as classes majoritárias. No entanto, em casos onde cada tupla da classe majoritária está próximo às outras tuplas de mesma classe, a perda de informação é reduzida [More, 2016]. A Figura 4 apresenta um exemplo de subamostragem aleatória.

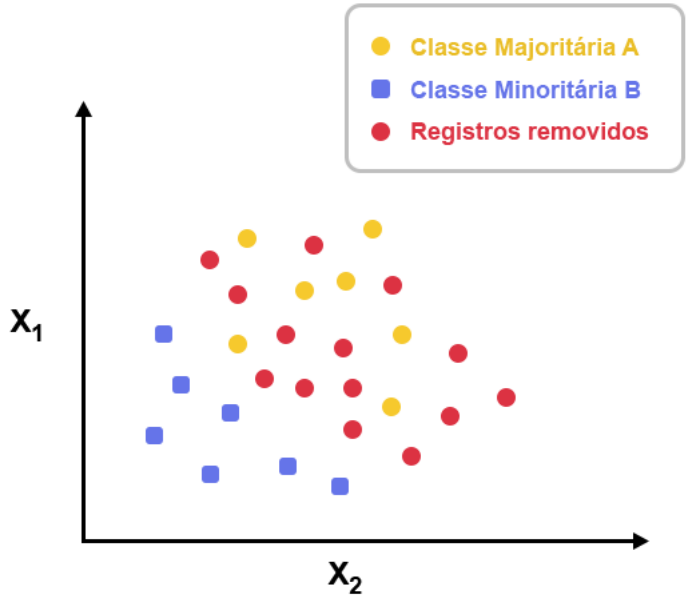

Figura 4: Exemplificação da RU

2.5.2 Técnica de Sobreamostragem Minoritária Sintética

A SMOTE é um método de balanceamento de dados que visa gerar tuplas sintéticas da classe minoritária no conjunto de dados. A sobreamostragem das tuplas da classe minoritária é realizada pela introdução de tuplas sintéticas a partir de uma tupla da classe minoritária e suas k-tuplas vizinhas mais próximas. As tuplas sintéticas são geradas pela diferença entre os atributos da tupla da classe minoritária escolhida e os atributos das suas vizinhas. Essa diferença é então multiplicada por um número aleatório entre 0 e 1, e adicionada à tupla de classe minoritária escolhida. Dependendo da quantidade de tuplas sintéticas necessárias, os k

vizinhos mais próximos são escolhidos aleatoriamente [Chawla et al., 2002]. A Figura 5 ilustra do processo de criação de tuplas sintéticas.

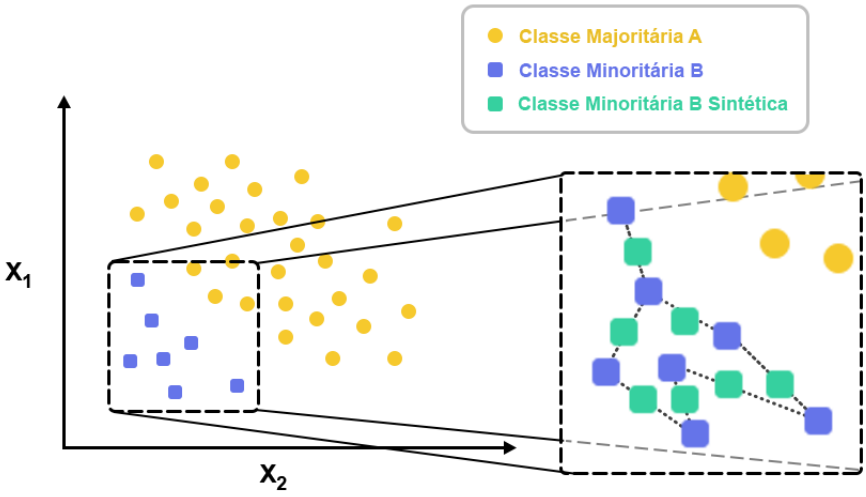

Figura 5: Exemplificação do SMOTE, adaptado de Borovicka et al. [2012]

2.6 Considerações Finais

Após a apresentação dos métodos de pré-processamento, podemos realizar a análise das principais qualidades e limitações de cada método. Assim, tendo como ponto de partida os métodos de redução de dados, podemos fazer as seguintes considerações:

(i) LASSO: O método LASSO possui vantagens estatísticas e computacionais, porque incentiva e/ou reforça a dispersão e a simplicidade na solução. Contudo, como todo algoritmo baseado na redução de dimensão, ele pode perder alguma variável independente relevante. Ou seja, o método irá selecionar apenas um atributo dentre um grupo de atributos correlacionados.

(ii) INFOGAIN: O método INFOGAIN minimiza a quantidade de informação necessária para a classificação dos dados, fazendo com que o número esperado de testes para classificar um dado seja menor [Han et al., 2011]. Em relação aos outros métodos, possui um baixo custo computacional. Contudo, por avaliar individualmente cada atributo, não leva em consideração as suas possíveis combinações, que podem ser bastante informativas.

(iii) CFS: Segundo Hall [1998], o método CFS identifica de forma rápida atributos irrelevantes, redundantes ou ruidosos, e também identifica atributos relevantes, desde que sua

relevância não dependa fortemente de outros atributos. Ele atua no espaço original dos atributos, o que faz com que qualquer conhecimento induzido por um algoritmo de aprendizado possa ser interpretado baseando-se nos atributos originais, e não nos atributos do espaço transformado. Além disso, o método é um filtro e, como tal, não é sujeito ao alto custo computacional associado às repetidas invocações do algoritmo de aprendizagem.

(iv) PCA: O algoritmo PCA pode ser utilizado em atributos ordenados ou não ordenados, e é capaz de lidar com dados esparsos ou distorcidos. Além disso, ele pode manipular dados que possuam dimensão maior que dois, por meio da redução da dimensão do problema, o que auxilia na visualização dos dados [Han et al., 2011]. Contudo, esta redução torna a interpretação dos dados mais difícil, dado que não estamos mais lidando com os atributos originais, e que os componentes principais são afetados pela escalabilidade dos atributos.

(v) Amostragem: A técnica de Amostragem exerce um papel importante na divisão do conjunto de dados em subconjuntos menores, principalmente quando tratamos da divisão dos dados de treinamento e de teste que serão utilizados pelos métodos de classificação, onde é importante assegurar que as diferentes classes sejam representadas. Além disso, ela possui um baixo custo computacional, e aumenta a velocidade da fase de aprendizado dos métodos de classificação [Han et al., 2011].

A seguir, tendo como base os métodos de transformação de dados apresentados, podemos fazer as seguintes considerações:

(i) Normalização Min-Max: A transformação realizada pela normalização Min-Max é necessária para garantir que todos os atributos tenham o mesmo peso quando realizada a execução de algoritmos de classificação, como redes neurais. Além disso, preserva a relação entre os valores transformados e os valores originais. No entanto, a normalização pode encontrar um erro se algum dos dados futuros se encontrar fora dos limites mínimo e máximo dos dados originais [Han et al., 2011].

(ii) Hierarquia Conceitual: A transformação por Hierarquia Conceitual facilita a compreensão dos dados substituindo-os por um menor número de intervalos ou conceitos. Assim os dados originais além de simplificados, são reduzidos, tornando os modelos de aprendizado de máquina mais eficientes. Além disso, a maioria das hierarquias para atributos nominais são implícitas ao esquema da base de dados e podem ser automaticamente definidas [Han et al., 2011].

(iii) Alisamento: A transformação realizada pelo Alisamento suaviza os dados, reduzindo o impacto de possíveis dados ruidosos. O método consulta os valores vizinhos de um possível dado ruidoso, aplicando assim uma suavização local. A suavização resultante é dependente do critério de suavização utilizado, como a média, mediana ou os limites do balde. Assim, quanto maior a largura dos baldes, maior o efeito da suavização sobre os dados. Além disso, os baldes devem possuir a mesma largura, onde os intervalos de valores em cada balde é constante [Han et al., 2011].

Capítulo 3

Aprendizado de Máquina

O aprendizado de máquina é o campo de estudo interessado no desenvolvimento de algoritmos computacionais com o intuito de auxiliar o ser humano a compreender grandes volumes de dados e extrair informações desejadas. Os algoritmos de aprendizado de máquina podem ser divididos em duas categorias, supervisionados e não supervisionados. Os algoritmos supervisionados possuem instruções claras dos dados que precisam ser aprendidos, como os modelos de previsão e classificação de dados, já os não supervisionados não possuem instruções dos dados que precisam ser aprendidos, como os modelos de descoberta de padrões frequentes. Independente do objetivo proposto, qualquer algoritmo de aprendizado de máquina pode ser desenvolvido a partir de cinco etapas: Coleta de dados; Pré-processamento; Treinamento; Validação e Aprimoramento [Lantz, 2013].

As etapas de coleta e pré-processamento de dados se resumem em integrar os dados em um mesmo formato, como um arquivo de texto, planilha ou base de dados e em seguida reduzir os ruídos da informação a partir dos métodos de pré-processamento para se obter uma maior qualidade na análise dos dados. Em seguida, parte dos dados é separada para a etapa de treinamento, em que o algoritmo de aprendizado de máquina escolhido analisa os dados e tenta criar um modelo que melhor os represente. Após o treinamento, a parte restante dos dados é utilizada para testar se o modelo gerado é capaz de generalizar e avaliar os dados corretamente. A etapa de aprimoramento consiste em aumentar o desempenho testando diferentes tipos de modelos, coletando mais dados, melhorando a etapa de pré-processamento ou evitando o *overfitting*, em que o modelo acaba decorando os dados de treinamento e é incapaz de generalizar e classificar os dados de teste [Lantz, 2013].

Neste trabalho serão utilizados os seguintes modelos supervisionados para previsão: Redes Neurais (Seção 3.1), k-Vizinhos mais próximos (kNN) (Seção 3.2), Máquina de Vetores de Suporte (SVM) (Seção 3.3), Classificador Bayesiano Ingênuo (Seção 3.4) e Florestas Aleatórias (Seção 3.5).

3.1 Redes Neurais

Uma abordagem para o aprendizado é tentar simular o funcionamento do cérebro humano no computador. Este foi um dos fatores motivadores para a criação das redes neurais [Burch, 2001].

As Redes Neurais (RN) compõem uma abordagem computacional que realiza o processamento da informação a partir de unidades básicas, denominadas neurônios. Elas são formadas por meio da comunicação entre os neurônios, que se dá a partir de sinapses [Tatibana and Kaetsu, 2002]. As sinapses possuem pesos associados a elas, que constituem a relevância daquela conexão. Elas são atualizadas durante a fase de treinamento, permitindo assim que a rede neural seja capaz de reconhecer e classificar os padrões [Cortiglioni et al., 2001].

A maioria das redes neurais possuem um modelo de treinamento, para que os pesos das conexões sejam atualizados conforme os padrões apresentados. A partir deste modelo, e das informações processadas que são propagadas para e entre os neurônios, a rede neural extrai as regras básicas da informação fornecida, adquirindo assim a sistemática dos padrões desejados [Tatibana and Kaetsu, 2002]. A Figura 6 ilustra, de forma resumida, a estrutura de neurônios e camadas de uma rede neural.

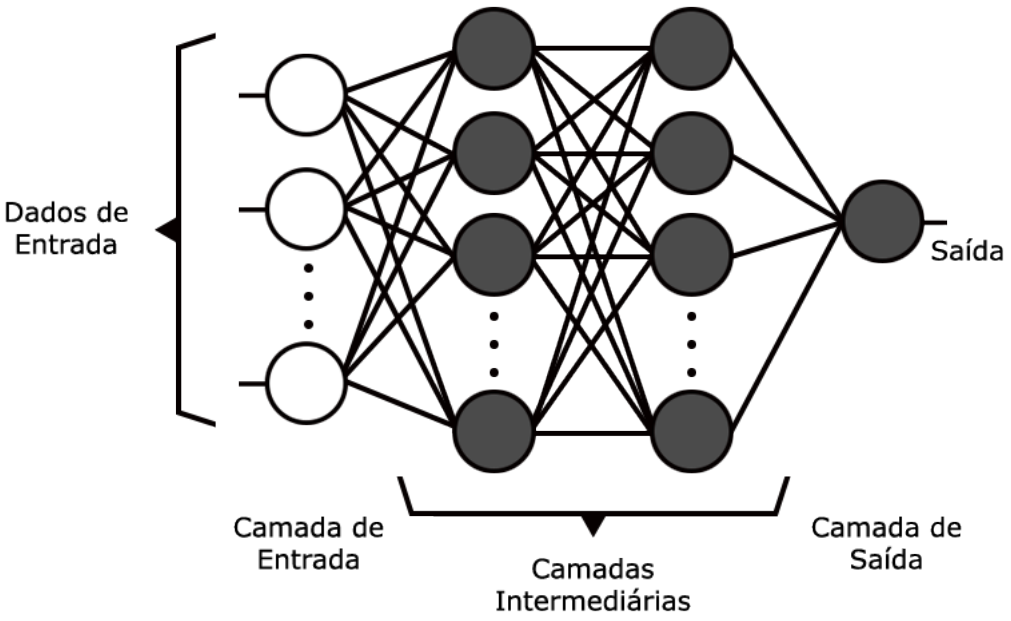

Figura 6: Ilustração da estrutura de uma rede neural, adaptado de Tatibana and Kaetsu [2002]

Sendo assim, ao final da fase de treinamento, temos como resultado um modelo capaz de reconhecer e classificar os padrões. Os parâmetros a serem considerados nesse estudo são:

RN_1 = nº de camadas intermediárias; RN_2 = nº de iterações; RN_3 = decaimento (parâmetro responsável por evitar o *overfitting* do modelo).

3.2 k-Vizinhos mais próximos

Os classificadores de k-Vizinhos mais próximos (kNN) se baseiam no aprendizado por analogia, isto é, comparando tuplas de uma base de teste com tuplas similares de uma base de treinamento. As tuplas da base de treinamento são descritas por n atributos, em que cada tupla representa um ponto em um espaço n-dimensional, fazendo com que todas as tuplas de treinamento sejam armazenadas em um espaço de padrões n-dimensional. Ao apresentar uma tupla desconhecida, o classificador de *k*-vizinhos mais próximos procura no espaço de padrão por *k* tuplas de treinamento mais próximas a tupla desconhecida. Essas *k* tuplas de treinamento então tornam-se os vizinhos mais próximos da tupla desconhecida [Han et al., 2011].

Para o classificador kNN, a tupla desconhecida é atribuída a classe mais comum dentre os *k*-vizinhos mais próximos. Assim, quando $k = 1$, a tupla desconhecida é atribuída a classe da tupla de treinamento mais próxima no espaço de padrões. O classificador também pode ser utilizado para predição numérica para retornar a previsão de um valor real, dado uma tupla desconhecida. Nesse caso, o classificador retorna a média dos valores associados aos *k*-vizinhos mais próximos da tupla desconhecida [Han et al., 2011]. A Figura 7 ilustra a classificação por k-vizinhos mais próximos.

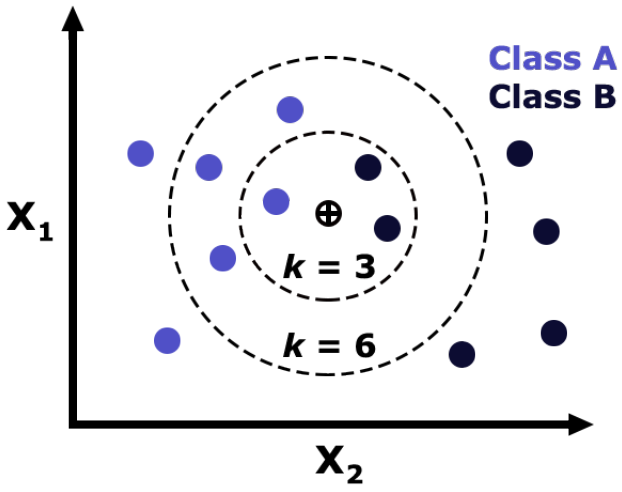

Figura 7: Ilustração da classificação por k-vizinhos mais próximos

A proximidade entre as tuplas é definida como uma distância métrica, como a distância euclidiana. Em outras palavras, é realizado o somatório do quadrado da diferença entre os

valores de um atributo de uma tupla X_1 e uma tupla X_2, para cada atributo numérico das tuplas. O classificador utiliza comparações que atribuem o mesmo peso a cada atributo, podendo assim apresentar uma baixa precisão quando trabalhado com atributos irrelevantes ou ruidosos, o que faz da fase de pré-processamento de dados fundamental para o sucesso do classificador [Han et al., 2011].

3.3 Máquina de Vetores de Suporte

A Máquina de Vetores de Suporte (SVM) é um método de classificação que pode ser usado tanto para dados lineares quanto não-lineares. Seu objetivo é criar um hiperplano, de forma que divida o espaço dimensional em partições homogêneas, servindo como uma fronteira que separa os dados mapeados no espaço [Lantz, 2013] [Han et al., 2011].

O SVM se utiliza de um mapeamento não-linear para aumentar a dimensão do espaço dos dados originais de treinamento. Nesta dimensão resultante, o método procura pelo hiperplano ideal, que melhor separe as tuplas das diferentes classes. Para encontrar este hiperplano, o SVM faz uso de vetores de suporte, utilizados para definir as margens do hiperplano. [Han et al., 2011].

Por conveniência, consideremos um caso simples em duas dimensionais, onde o SVM precisa identificar uma linha que separa duas classes. Na Figura 8, são exemplificados algumas possibilidades para a separação destas classes.

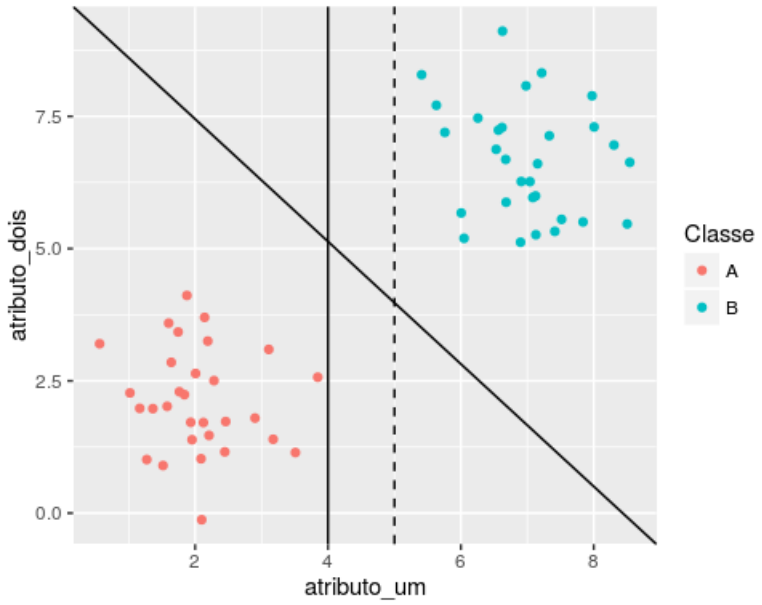

Figura 8: Exemplificação de separação de classes, adaptado de Lantz [2013]

Dentre as infinitas possibilidades existentes, é necessário que o SVM busque pela margem máxima do hiperplano (MMH), que cria a maior separação entre as classes. A margem máxima tende a melhorar a generalização dos dados futuros, e a diminuir a chance dos dados ruidosos impactarem na separação das classes [Lantz, 2013].

Os vetores de suporte são os pontos de cada classe que estão mais próximos da MMH. Cada classe precisa ter pelo menos um vetor de suporte. Eles fornecem uma maneira compacta de armazenar um modelo de classificação, independente da quantidade de atributos [Lantz, 2013].

3.4 Classificador Bayesiano Ingênuo

Os classificadores Bayesianos são classificadores estatísticos que podem ser utilizados para prever a probabilidade de uma tupla pertencer a uma classe em particular. Os Classificadores Bayesianos Ingênuos (NB) assumem que o efeito de um valor de atributo em uma determinada classe é independente dos valores dos outros atributos, criando uma independência condicional. Essa independência condicional proposta entre os atributos simplifica a computação envolvida e por isso o classificador é considerado "ingênuo"ou "simples"[Han et al., 2011].

Os NB funcionam a partir de um grupo de treinamento T e seus respectivos rótulos de classe. Normalmente, cada tupla do grupo de treinamento é representada por um vetor de atributos n-dimensional, $X = (x_1, x_2, ..., x_n)$. Supondo que existam m classes, $C_1, C_2, ..., C_m$, dada uma tupla X, o classficador irá prever que X pertence a classe que possui a maior probabilidade posterior. Sendo assim, o NB prevê que a tupla X pertence a class C_i segundo a Equação 3.1 a seguir.

$$P(X|C_i)P(C_i) > P(X|C_j)P(C_j), 1 \leq j \leq m, j \neq i. \tag{3.1}$$

Em outras palavras, a classe prevista para a tupla X é a classe C_i para qual $P(X|C_i)P(C_i)$ tem o valor máximo [Han et al., 2011].

3.5 Florestas Aleatórias

Segundo Breiman [2001], Florestas Aleatórias (RF) são a combinação de árvores de decisão, de modo que cada árvore depende dos valores de um vetor aleatório amostrado de forma independente, e com a mesma distribuição para todas as árvores na floresta. Ou seja, cada árvore de decisão é gerada por meio de uma seleção aleatória de atributos, que é feita em cada nó, a fim de determinar a divisão [Han et al., 2011].

Após a formação da floresta, o modelo utiliza o voto para combinar as predições de cada árvore. A classe mais votada é retornada como o resultado da previsão [Lantz, 2013]. Sua precisão depende da força de cada árvore, e da dependência entre elas. O ideal é preservar a força de cada árvore, sem aumentar as suas correlações [Han et al., 2011]. A Figura 9 ilustra o processo de classificação por florestas aleatórias.

O erro de generalização para uma floresta converge contanto que o número de árvores na floresta seja grande, o que faz com que o *overfitting* não seja um problema [Han et al., 2011]. Além disso, ela consegue lidar com um conjunto de dados extremamente grande, já que o conjunto se utiliza de apenas uma pequena parte aleatória do conjunto de dados original [Lantz, 2013].

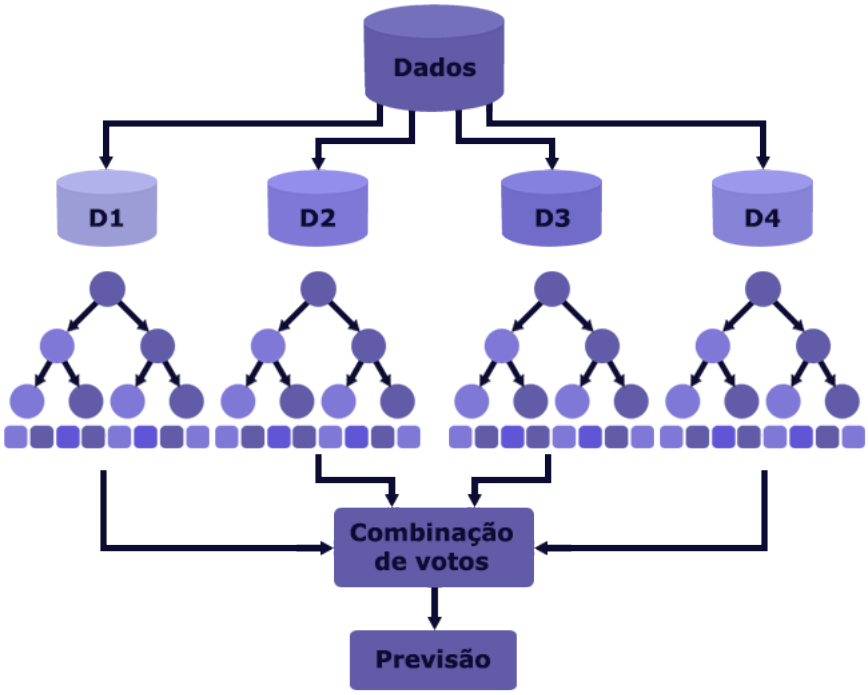

Figura 9: Ilustração da Classificação por Florestas Aleatórias

3.6 Considerações Finais

Após a apresentação dos modelos de aprendizado de máquina, podemos realizar a análise das principais qualidades e limitações de cada modelo. Sendo assim, podemos fazer as seguintes considerações:

(i) RN: O modelo de RN possui uma alta tolerância a dados ruidosos, e a habilidade de classificar padrões não treinados [Han et al., 2011]. A partir das suas camadas escondidas,

ele pode detectar todas as possíveis interações entre os atributos de entrada. Além disso, o modelo pode ser criado utilizando diferentes algoritmos de treinamento [Tu, 1996].

As desvantagens que podem ser destacadas são a inclinação ao *overfitting*, o alto custo computacional, e que sua criação se dá por meio de um processo empírico de análise dos parâmetros de treinamento [Tu, 1996].

(ii) kNN: O método kNN é simples e eficiente no campo de reconhecimento de padrões. A simplicidade é uma de suas principais vantagens, assim como a rapidez na fase de treinamento, a tolerância a dados de treinamento com ruídos e a eficiência mesmo com uma grande quantidade de dados de treinamento. No entanto, as suas desvantagens não podem ser ignoradas, já que a técnica é dependente do valor escolhido para k, possui uma alta complexidade computacional, limitações de memória, além de ser influenciado por atributos irrelevantes e ser uma técnica que utiliza um algoritmo preguiçoso de aprendizado supervisionado, isto é, demora para realizar o reconhecimento [Bhatia and Vandana, 2010].

(iii) SVM: O método SVM tende a ser menos propenso ao *overfitting* do que outros métodos de aprendizado de máquina, já que a complexidade do classificador é caracterizada pelo número de vetores de suporte, e não pela dimensionalidade dos dados [Han et al., 2011]. Dependendo da quantidade de atributos ou amostras, a fase de treinamento pode ser lenta. Além disso, o processo de encontrar o melhor modelo requer o teste de diferentes combinações de *kernels* e parâmetros [Lantz, 2013]. Contudo, devido à sua capacidade de modelar margens não-lineares complexas, o método é altamente preciso [Han et al., 2011].

(iv) NB: A técnica NB geralmente possui a menor taxa de erro em comparação a outros classificadores, além de ser tolerante a dados ruidosos. A sua suposição de independência condicional reduz o alto custo computacional para se calcular $P(X|C_i)$, dado um *dataset* com muitos atributos. No entanto, além de ser sensível a dados irrelevantes, sua taxa de erro pode aumentar, devido a imprecisões na suposição de independência condicional, ou seja, caso haja dependências entre atributos [Han et al., 2011].

(v) RF: O método de RF é robusto a erros e *outliers*. O erro de generalização para uma floresta converge contanto que o número de árvores na floresta seja grande. Assim não é

preciso se preocupar com o problema de *overfitting*. O método não é sensível ao número de atributos selecionados para cada divisão, em que geralmente são escolhidos $log_2 d + 1$ atributos. No entanto, observações empíricas apontam que utilizando apenas um atributo de entrada aleatório, o modelo resultante já possui uma boa precisão, geralmente maior do que quando utilizados diversos atributos. Além disso, considerando poucos atributos para cada divisão, o método se torna mais eficiente em bases de dados muito grandes [Han et al., 2011].

Capítulo 4

Trabalhos Relacionados

A previsão de atrasos aéreos é um tema relevante no cenário da aviação comercial, devido aos transtornos causados às companhias aéreas, aeroportos e passageiros. A Tabela 1 apresenta uma linha do tempo com as publicações sobre previsão de atrasos aéreos que utilizam aprendizado de máquina, e que nortearão a discussão sobre os trabalhos realizados sobre o tema.

Tabela 1: Publicações sobre previsão de atrasos aéreos com aprendizado de máquina.

Ano	Publicação
2008	Chen et al. Lu et al. Balakrishna et al.
2010	Balakrishna et al.
2014	Khanmohammadi et al. Rebollo and Balakrishnan et al.
2016	Belcastro et al.

Chen et al. [2008] construíram um modelo de aviso prévio de atrasos de voo baseado em uma máquina de vetor de suporte *fuzzy* com margem ponderada. Analisando as amostras de seus dados, os autores propuseram cinco graus de atraso de voo. Em seu trabalho concluíram que o desempenho da máquina de vetor de suporte *fuzzy* com margem ponderada foi 4 a 8% mais precisa do que um SVM autônomo, dependendo do grau de atraso.

Lu Zonglei [2008] utilizam os dados de voo coletados de um aeroporto da China, a fim de criar um novo método para alarmar atrasos aéreos de grande escala. Este método possui duas etapas, onde a primeira se baseia na aplicação da técnica de aprendizado de máquina não-supervisionado *k-Means* para se obter as classes de atraso. Após isso, são aplicados os métodos de classificação Bayesiano, árvore de decisão e rede neural *backpropagation*. Em seu trabalho concluíram que a árvore de decisão obteve os melhores resultados, apresentando uma confiança de previsão de no mínimo 80%.

Balakrishna et al. [2008, 2010] utilizaram um algoritmo de aprendizado por reforço para prever atrasos aéreos antes da decolagem (*taxi-out delays*). Este problema foi modelado por meio do processo de decisão de Markov, e resolvido utilizando um algoritmo de aprendizado

por reforço. O modelo foi executado 15 minutos antes da hora programa da partida, para os aeroportos JFK International Airport em Nova Iorque e International Airport de Tampa Bay. Em média, com um erro padrão de um minuto e meio, o modelo obteve a precisão de 93,7% ao prever atrasos em um determinado trimestre. Em relação aos voos individuais, com um erro padrão de 2 minutos, o modelo obteve a precisão de 81%.

Khanmohammadi et al. [2014] implementaram um *framework* de resolução de problemas de sistemas gerais integrando técnicas de inteligência computacional (GSPS-CI). O *framework* GSPS-CI possui duas funções, onde a primeira é prever atrasos aéreos por meio de uma rede adaptativa baseada em um sistema de inferência *fuzzy*, e a segunda é utilizar as predições como dados de entrada para o método de decisão *fuzzy*, com o objetivo de programar aterragens de aeronaves. Ele foi testado no aeroporto JFK International Airport em Nova Iorque.

Rebollo and Balakrishnan [2014] modelaram a rede de aeroportos dos Estados Unidos da América, a fim de prever atrasos aéreos. O objetivo de seu trabalho era prever o atraso de partida em um determinado link ou aeroporto, em algum tempo no futuro. Eles compararam os modelos de classificação e regressão para a previsão dos atrasos, e devido aos resultados, escolheram a técnica de florestas aleatórias como classificador. O modelo de previsão de Rebollo e Balakrishnan não fez uso de dados meteorológicos para atingir seu objetivo, diferente do nosso, que usará os dados coletados no banco de dados público chamado *Weather Underground* (WU) [The Weather Company, 2016].

Belcastro et al. [2016] tinham como objetivo implementar um preditor de atrasos aéreos de voos programados devido às condições climáticas. O preditor leva em consideração as informações de voo, como aeroporto de origem e destino, e as condições climáticas destes aeroportos. Tanto o conjunto de dados meteorológicos quanto os de voo foram analisados e minados usando algoritmos paralelos implementados como programas *MapReduce*, executados numa plataforma *Cloud*. Estes algoritmos são uma versão paralela do método de RF. Os resultados apresentados mostraram uma alta precisão na previsão dos atrasos acima de um determinado limiar. Em seu trabalho, verificou-se que ao desconsiderar as condições climáticas, seu modelo perdia cerca de 17% da precisão.

A partir destas publicações, a Tabela 2 apresenta uma comparação dos trabalhos que utilizam as técnicas de aprendizado de máquina aqui discutidas. Na tabela, as técnicas de pré-processamento estão agrupadas conforme suas etapas. Por meio dela, pode-se notar que são poucos os trabalhos que se utilizam de todas as etapas de pré-processamento. Além disso,

embora haja trabalhos que se utilizam de três das quatro etapas, estes não utilizam as diversas técnicas que foram aqui apresentadas. O mesmo vale para as técnicas de aprendizado de máquina. Com isso, pode-se notar uma lacuna para a avaliação da precisão dos diferentes modelos de aprendizado de máquina supervisionados, combinados com os diversos métodos de pré-processamento aqui discutidos, de modo a produzir modelos de previsão mais precisos.

Tabela 2: Comparativo das técnicas utilizadas nos trabalhos relacionados

Pub.	Pré-processamento				Aprendizado de Máquina				
	Integração	Limpeza	Redução	Transformação	RN	KNN	SVM	NB	RF
[1]			X				X		
[2]					X			X	X
[3]			X	X	X				X
[4]	X	X		X			X	X	X

[1] = Chen et al. [2008]
[2] = Lu Zonglei [2008]
[3] = Rebollo and Balakrishnan [2014]
[4] = Belcastro et al. [2016]

Capítulo 5

Metodologia

A metodologia utilizada para analisar os atrasos aéreos é composta por sete etapas, como apresentado no *workflow* da Figura 10. A etapa 1 (Seção 5.1) consiste na integração das bases de dados. A etapa 2 (Seção 5.2) é responsável por realizar a limpeza e remoção dos *outliers*. A etapa 3 (Seção 5.3) consiste na transformação e consolidação dos dados. A etapa 4 (Seção 5.4) é responsável por realizar a amostragem para treino e teste, e as diferentes seleções de atributos nos dados. A etapa 5 (Seção 5.5) consiste na aplicação dos métodos de balanceamento nos dados de treino. A etapa 6 (Seção 5.6) consiste na aplicação dos métodos de aprendizado de máquina, na realização da validação cruzada e otimização dos hiperparâmetros. Finalmente, a etapa 7 (Seção 5.7) realiza a avaliação de cada modelo gerado.

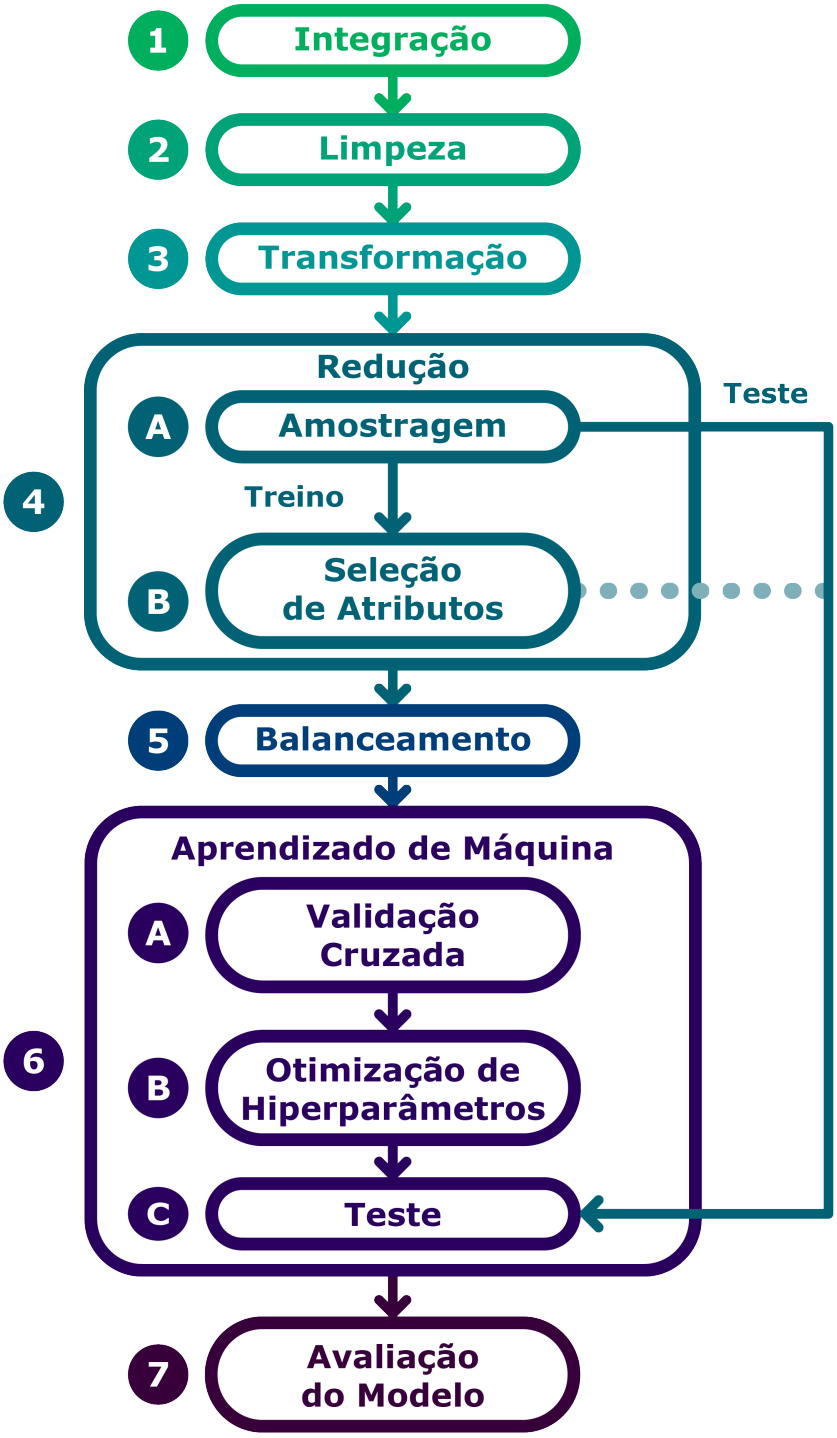

Figura 10: Processo de Mineração de dados de Atrasos Aéreos

5.1 Integração

A ANAC é responsável por regular e supervisionar as atividades da aviação civil no Brasil. Todos os voos comerciais programados para partir e chegar nos aeroportos brasileiros são controlados e tem as suas informações armazenadas nos bancos de dados da ANAC. Os dados

armazenados contam com diversas informações sobre as operações de voo, como os aeroportos, o horário programado para a partida e chegada, assim como os horários reais de partida e chegada e são disponibilizados mensalmente pela ANAC em um banco de dados público chamado Voo Regular Ativo (VRA) [Sternberg et al., 2016].

A base de dados VRA não possui dados meteorológicos. Assim, para criar um *dataset* de voos mais completo, os dados do provedor de serviços meteorológicos *WU* foram coletados. Esses dados contém diversas informações sobre temperatura, pressão e umidade para cada cidade. Assim, os dados das duas bases de dados foram integrados por Sternberg et al. [2016] para formar um *dataset* mais completo, considerando a cidade de cada aeroporto e os horários mais próximos dos horários programados de partida e chegada dos voos.

5.2 Limpeza

Após a formação do *dataset*, algumas discrepâncias na definição da ANAC para os atrasos foram identificadas. Segundo as leis de regulamentação brasileiras, um voo com atraso maior que 24 horas é considerado cancelado. Sendo assim, os voos realizados que não respeitavam esta restrição foram considerados discrepâncias e foram removidos por Sternberg et al. [2016]. As discrepâncias removidas representavam 0.02% dos registros.

Em seguida, Sternberg et al. [2016] aplicou o método *box-plot* para a remoção de *outliers* nas durações dos voos para cada par de aeroportos de partida e chegada. O mesmo método de remoção de *outliers* foi aplicado para as diferenças entre os horários reais e os horários programados de partida e chegada. Assim, os *outliers* removidos totalizaram 1.97% dos registros.

A seguir, foram removidos os voos que não foram realizados, cerca de 5.51% dos registros do *dataset*. Após realizar a limpeza dos dados de voo, foram removidos os valores faltantes dos dados meteorológicos, cerca de 5.04% dos registros. E por último, foram removidos os *outliers* dos dados meteorológicos, cerca de 3.41% dos registros.

5.3 Transformação

Após a limpeza dos dados, a etapa de transformação será realizada para discretizar atributos contínuos e consolidar os dados em um formato apropriado para a aplicação dos métodos de aprendizado de máquina. Este trabalho utiliza as técnicas de transformação de dados apresentados na Seção 2.4.

A técnica de hierarquia conceitual foi aplicada nos atributos direção do vento, data de partida e chegada prevista, para obter informações como ano, mês, dia da semana, hora, dia útil e as diferentes direções do vento. A técnica de Alisamento foi utilizada para discretizar os atributos meteorológicos contínuos de temperatura, ponto de orvalho, pressão, umidade e velocidade do vento, tanto para partida quanto para chegada. Após a aplicação do alisamento, esses atributos foram normalizados utilizando a técnica de normalização min-max para um intervalo de 0 a 1. Em seguida, os atributos que foram tratados com a técnica de hierarquia conceitual e os atributos de sigla da empresa, aeroporto de partida e chegada, e condições de partida e chegada foram representados de forma binária a partir de *n* atributos, um para cada possível valor dos atributos. A Tabela 3 apresenta os atributos transformados e as principais técnicas utilizadas para a realização das transformações.

5.4 Redução

Em Sternberg et al. [2016], uma redução de dados foi utilizada através da seleção de atributos. Essa seleção foi realizada em duas etapas: (i) seleção de aeroportos e (ii) seleção de linhas aéreas, devido a importância e aos diferentes tamanhos dos aeroportos e linhas aéreas brasileiras. Assim como em Sternberg et al. [2016], foram selecionados 17 aeroportos que correspondem a 80% das partidas programadas entre Janeiro de 2009 e Fevereiro de 2015. As principais e maiores linhas aéreas do Brasil também foram selecionadas: Tam, Gol, Avianca, e Azul, responsáveis por 93% dos voos realizados no período. Em seguida, este trabalho realizou as técnicas de redução de dados apresentadas na Seção 2.3 sobre o *dataset*, seguindo os seguintes passos:

(i) Amostragem: Após a etapa de transformação, dado o grande volume do *dataset* resultante, foi realizada uma amostragem estratificada de 10% da quantidade total dos dados, obtendo um novo *dataset* com 295.313 registros. A seguir, o novo *dataset* foi dividido em duas partições utilizando uma amostragem aleatória, uma com 80% dos dados para o treinamento dos classificadores, e outra com 20% dos dados para o teste. Ambas as partições mantiveram as mesmas proporções entre as classes de atraso do *dataset* original.

(ii) Seleção de atributos: A partir da partição de treinamento foram aplicados os métodos de seleção de atributos apresentados na Seção 2.3. Assim, foi gerado um novo conjunto de treinamento para cada método de seleção de atributos utilizado. Em seguida, os atributos

Tabela 3: Transformações

Dimensão	Atributo Original	Intervalo Original	Atributo Transformado	Valores Transformados	Técnica
Temporal	Chegada/ Partida Prevista	de 01/01/2009 0:00 até 28/02/2015 23:59	Ano	2009 até 2015	Hierarquia Conceitual
			Mês	1 à 12	Hierarquia Conceitual
			Dia da Semana	Domingo à Sábado	Hierarquia Conceitual
			Dia Útil	Verdadeiro e Falso	Hierarquia Conceitual
			Período do dia	Early morning: 5:00 até 8:59 Mid morning: 9:00 até 10:59 Late morning: 11:00 até 12:59 Afternoon: 13:00 até 16:59 Early evening: 17:00 até 19:59 Late evening: 20:00 até 22:59 Night: 23:00 até 4:49	Hierarquia Conceitual
Meteorológica	Temperatura	-3 até 41	Temperatura	Baixo: -3 até 21 Médio: 21.1 até 26 Alto: 26.1 até 41	Alisamento
Meteorológica	Ponto de Orvalho	-24 até 32	Ponto de Orvalho	Baixo: -24 até 15 Médio: 15.1 até 19 Alto: 19.1 até 32	Alisamento
Meteorológica	Pressão	996 até 1036	Pressão	Baixo: 996 até 1014 Médio: 1015 até 1018 Alto: 1019 até 1036	Alisamento
Meteorológica	Umidade(%)	5 até 100	Umidade	Baixo: 5 até 65 Médio: 66 até 83 Alto: 84 até 100	Alisamento
Meteorológica	Velocidade do Vento (km/h)	0 até 51.9	Velocidade do Vento	Baixo: 0.0 até 7.4 Médio: 7.5 até 14.8 Alto: 14.9 até 51.9	Alisamento
Meteorológica	Direção do Vento (graus)	0 à 360	Direção do Vento	N: 349 a 11 NNE: 11 a 33 NE: 34 a 56 ENE: 57 a 78 E: 79 a 101 ESE: 102 a 123 SE: 124 a 146 SSE: 147 a 168 S: 169 a 191 SSO: 192 a 213 SO: 214 a 236 OSO: 237 a 258 O: 259 a 281 ONO: 282 a 303 NO: 304 a 326 NNO: 327 a 348	Hierarquia Conceitual

selecionados foram refletidos para a partição de teste original, para criar os respectivos conjuntos de teste para cada conjunto de treinamento.

5.5 Balanceamento de Dados

O *dataset* é composto por aproximadamente 78% dos registros de voo sem atraso, e portanto, apenas 22% com atraso. Essa desproporção torna o *dataset* desbalanceado, já que as classes de atraso não são igualmente representadas, o que faz com que os classificadores tendam a fornecer um alto grau desbalanceado de acurácia, com a classe majoritária tendo perto de 100%, e a classe minoritária tendo entre 0-10% [He and Garcia, 2009].

Portanto, para obter uma distribuição mais balanceada entre as classes do *dataset* os métodos de balanceamento (Seção 2.5) foram aplicados no conjunto de treinamento. A análise dos resultados obtidos para os diferentes métodos é discutida no Capítulo 6.

5.6 Aprendizado de Máquina

Após a aplicação das técnicas de pré-processamento, o *dataset* resultante foi dividido em conjuntos de treinamento e teste. Em seguida, o conjunto de treinamento foi utilizado para a execução dos seguintes passos:

(i) Validação Cruzada: A validação cruzada foi realizada por meio do método *k-fold*. O valor adotado para k foi 10, já que a evidência empírica sugere que há pouco benefício adicional ao usar um valor maior [Lantz, 2013]. Portanto, o conjunto de treinamento foi divido em dez partições estratificadas. Essa divisão resultou em dez combinações diferentes que foram utilizadas para a modelagem dos classificadores com validação cruzada, contendo cada uma delas 90% dos dados para treinamento, e 10% para validação. Sendo assim, foi possível explorar diferentes parâmetros para cada modelo de aprendizado de máquina.

(ii) Otimização de Hiperparâmetros: A etapa de Otimização de Hiperparâmetros tem como objetivo identificar os melhores parâmetros explorados para cada modelo de aprendizado de máquina durante o treinamento com validação cruzada [Bergstra et al., 2011]. Os modelos de aprendizado de máquina foram avaliados dez vezes, conforme as combinações anteriormente geradas.

A escolha dos melhores parâmetros foi realizada por meio da análise de Média-Variância

[Wang, 2009]. Assim, foram considerados como melhores parâmetros aqueles que tiveram a melhor proporção entre seu desempenho médio e sua variância.

(iii) Teste: Por fim, os modelos de aprendizado de máquina anteriormente gerados foram aplicados sobre o conjunto de teste. Os modelos de previsão foram avaliados conforme a métrica apresentada na Seção 5.7, e os resultados desta etapa são discutidos no Capítulo 6.

5.7 Métrica de avaliação para os classificadores

Os modelos de aprendizado de máquina foram avaliados e comparados segundo sua precisão ao estimar as classes do conjunto de teste, definidos na Seção 5.6. A Tabela 4 apresenta a matriz de confusão, definida em Han et al. [2011], utilizada para avaliar os modelos de previsão gerados pelos métodos de aprendizado de máquina apresentados no Capítulo 3.

Tabela 4: Matriz de confusão, adaptado de Han et al. [2011]

	Classe Prevista		
Atraso	Sim	Não	Total
Sim	VP	FN	P
Não	FP	VN	N
Total	P'	N'	$P+N$

Dadas duas classes, como tuplas positivas em que atraso = sim e tuplas negativas em que atraso = não, P é o número de tuplas positivas e N é o número de tuplas negativas. A classe das tuplas do conjunto de teste são comparadas com a classe prevista pelo modelo gerado. Assim, podemos descrever os termos:

- Verdadeiras Positivas (VP): Número de tuplas positivas classificadas corretamente pelo modelo.
- Verdadeiros Negativos (VN): Número de tuplas negativas classificadas corretamente pelo modelo.
- Falsos Positivos (FP): Número de tuplas negativas classificadas incorretamente como tuplas positivas (isto é, tuplas da classe atraso = não para as quais o modelo classificou como atraso = sim).

- Falsos Negativos (FN): Número de tuplas positivas classificadas incorretamente como tuplas negativas (isto é, tuplas da classe atraso = sim para as quais o modelo classificou como atraso = não).

Esses termos compõem a matriz de confusão da Tabela 4. A matriz de confusão é uma ferramenta utilizada para a análise da precisão de modelos de previsão, onde P' é o número de tuplas classificadas pelo modelo como positivas ($VP + FP$) e N' é o número de tuplas classificadas como negativas ($VN + FN$). Por meio dela, é possível calcular as métricas de acurácia e sensibilidade, que serão utilizadas neste estudo, representadas, respectivamente, pelas Equações 5.1 e 5.2.

$$Acurácia = \frac{VP + VN}{P + N} \tag{5.1}$$

$$Sensibilidade = \frac{VP}{P} \tag{5.2}$$

A acurácia equivale à porcentagem de tuplas que são corretamente identificadas pelo classificador, enquanto a sensibilidade representa a proporção das tuplas positivas (de interesse) que são corretamente identificadas [Han et al., 2011].

Capítulo 6

Avaliação Experimental

Os experimentos foram conduzidos em um computador com processador Intel i5 com 16GB de memória RAM e usando sistema operacional de 64 bits Windows 7 *Professional*. A linguagem R [R Core Team, 2013] foi utilizada para a realização das técnicas de pré-processamento e aprendizado de máquina. Todas elas estão disponíveis como pacotes R (*nnet*, *kernlab*, *class*, *randomForest*, *e1071*, *FSelector*, *glmnet*, *caret*, *unbalanced*). A base de dados e os resultados experimentais alcançados encontram-se disponíveis no repositório GitHub (`https://github.com/eogasawara/flight-delay-prediction`).

A avaliação experimental foi dividida em duas partes, sendo elas: a Análise preliminar dos métodos de aprendizado de máquina (Seção 6.1) e a Análise dos métodos de pré-processamento (Seção 6.2).

6.1 Análise preliminar dos métodos de aprendizado de máquina

As combinações das técnicas de seleção de atributos e balanceamento de dados resultam num total de 15 *datasets* de treinamento. Para cada um deles, é necessário a realização da validação cruzada 10-*fold*, com o objetivo de definir os melhores parâmetros para cada modelo de aprendizado de máquina. A grande quantidade de combinações entre os *datasets* criados, as validações cruzadas a serem realizadas e o tempo necessário para a realização de todos os testes tornariam o custo computacional muito grande. Por isso, se fez necessário escolher um método de aprendizado de máquina que tornasse viável a realização de todas as combinações de treinamento e testes necessários.

O método selecionado foi escolhido a partir de uma análise preliminar dos métodos de aprendizado de máquina estudados. Utilizamos como *dataset* a partição de treinamento em que foi aplicada a técnica de seleção de atributos LASSO. Os métodos de aprendizado de máquina foram avaliados com a suas configurações de parâmetros que resultassem no menor tempo de execução, tendo como base a exploração de parâmetros em Machado et al. [2016]. A Tabela 5 apresenta o tempo aproximado de execução, a acurácia e o número aproximado de combinações

de parâmetros a serem explorados para cada método.

Tabela 5: Análise preliminar dos métodos de Aprendizado de Máquina

Método	Tempo Aprox. de Execução (horas)	Acurácia (%)	Nº Aprox. de Combinações de Parâmetros
RN	00:02	78.02	20
kNN	00:23	67.80	27
NB	00:03	74.81	-
SVM_{rbf}	05:01	77.99	9
SVM_{tanh}	03:09	77.99	9
RF	00:01	77.94	27

Portanto, com base nas análises qualitativas realizadas na Seção 3.6, na acurácia semelhante aos outros classificadores, no tempo de execução e no número de combinações de parâmetros a serem explorados, foi escolhido como classificador o método de aprendizado de máquina de redes neurais *backpropagation*.

6.2 Análise dos métodos de pré-processamento

Após a realização das três primeiras etapas, apresentadas no *workflow* da Figura 10, foram obtidos 2.953.139 registros de voo, e 187 atributos, apresentando 22% dos voos com atraso identificados com 1, e os demais 78% sem atraso, identificados com 0. Dado o grande volume do *dataset* resultante, foi realizada uma amostragem estratificada de 10%, obtendo um novo *dataset* com 295.313 registros.

Na quarta etapa, este *dataset* foi dividido em uma partição de treinamento com 236.250 registros (80%) e uma partição de teste com 59.063 registros (20%). Tanto a amostra estratificada quanto as partições mantiveram as mesmas proporções de atraso do *dataset* original. Em seguida, foram aplicadas as técnicas de seleção de atributos na partição de dados de treinamento, criando um novo *dataset* de treinamento para cada técnica utilizada. A Tabela 6 apresenta a quantidade de atributos selecionados por cada técnica utilizada. A elevada quantidade de atributos para o método de seleção PCA se deu pelo fato do *dataset* possuir variáveis categóricas, fazendo com que a variância adquirida por cada componente fosse baixa.

Tabela 6: Tabela de Seleção de Atributos

Método de Seleção	Quantidade de Atributos
Nenhum	187
LASSO	21
INFOGAIN	55
CFS	24
PCA	115

Na quinta etapa foram realizados os métodos de balanceamento nos *datasets* de treinamento para obter uma melhor distribuição entre as classes de atraso. Os métodos de balanceamento utilizados alteram a quantidade de registros dos *datasets* para realizar o balanceamento. Na aplicação do método RU, foi realizado a subamostragem da classe majoritária, de forma que ela ficasse com a mesma quantidade de registros que a classe minoritária. Na aplicação do método SMOTE, foi realizado a sobreamostragem da classe minoritária, dobrando a sua quantidade, e a subamostragem da classe majoritária, de forma que a sua quantidade de registros ficasse igual a da classe minoritária. A Tabela 7 apresenta o resultado final do balanceamento.

Tabela 7: Tabela de Balanceamento

Método de Balanceamento	Quantidade de Registros		
	Com Atraso	Sem Atraso	Total
Nenhum	184.094	52.156	236.250
RU	52.156	52.156	104.312
SMOTE	104.312	104.312	208.624

Na sexta etapa, foi realizado o treinamento e a validação dos modelos de aprendizado de máquina. Dada as combinações de técnicas de seleção de atributos e balanceamento de dados, foram criados um total de 15 *datasets* de treinamento. Para cada um dos *datasets* foi realizada a validação cruzada 10-*fold*, com o objetivo de definir os melhores parâmetros para cada modelo de aprendizado de máquina. Em seguida, a partir da validação cruzada, foram explorados os melhores parâmetros para os modelos de rede neural, com a quantidade de neurônios variando entre 3 e 12, e o decaimento de 0.01 e 0.001. Para todos os *datasets*, o melhor parâmetro para o decaimento foi 0.01. A Tabela 8 apresenta a melhor quantidade de neurônios para cada *dataset*.

Tabela 8: Tabela de Melhor Parâmetro de Neurônios

Método de Seleção	Método de Balanceamento		
	Nenhum	RU	SMOTE
Nenhum	8	5	10
LASSO	9	8	9
CFS	10	8	10
INFOGAIN	9	7	10
PCA	8	5	10

Considerando a distribuição das classes do *dataset*, um método de classificação simples, que classifique todos os voos como a classe majoritária sem atraso, apresentaria uma acurácia de 78%. Portanto, um bom modelo de classificação deve superar a acurácia do classificador mais simples. As Tabelas 9 e 10 apresentam os resultados de acurácia e sensibilidade, respectivamente, obtidos pelos modelos de rede neural gerados pelas combinações de seleção de atributos e métodos de balanceamento de dados.

Tabela 9: Tabela de Acurácia para as Redes Neurais, segundo o método de seleção de atributos e o método de balanceamento (em %)

Método de Seleção	Método de Balanceamento		
	Nenhum	RU	SMOTE
Nenhum	**78.18**	61.44	73.81
LASSO	78.06	59.14	59.04
CFS	78.10	60.20	60.32
INFOGAIN	78.12	60.23	64.65
PCA	78.00	60.52	67.24

Tabela 10: Tabela de Sensibilidade para as Redes Neurais, segundo o método de seleção de atributos e o método de balanceamento (em %)

Método de Seleção	Método de Balanceamento		
	Nenhum	RU	SMOTE
Nenhum	5.93	58.41	26.03
LASSO	1.81	58.75	58.89
CFS	1.88	56.46	56.08
INFOGAIN	3.57	58.47	49.10
PCA	5.17	**60.13**	43.47

A sensibilidade se refere à capacidade do classificador de prever corretamente a classe de interesse, no caso, a classe onde há atraso, definida pela porcentagem de registros classificados corretamente. Analisando os resultados obtidos, utilizando os métodos de seleção de atributos

sem realizar o balanceamento, apresentados na primeira coluna das Tabelas 9 e 10, foi possível observar que a acurácia e a baixa sensibilidade dos classificadores foram afetadas pelo desbalanceamento do *dataset*, aproximando-se da acurácia do classificador mais simples, 78%.

A fim de contornar este problema, foram aplicados os métodos de balanceamento de dados RU e SMOTE, que obtiveram como melhor percentual de acurácia 73.81%, com a combinação do modelo sem métodos de seleção de atributos e o balanceamento SMOTE. O melhor percentual de sensibilidade foi de 60.13%, obtido com a combinação do modelo com o método de seleção de atributos PCA e o balanceamento RU. Portanto, ainda que tenham sido aplicados os métodos de balanceamento na tentativa de melhorar a precisão dos resultados dos classificadores, estes permaneceram inferiores ao classificador mais simples. No entanto, o aumento considerável da sensibilidade com o balanceamento dos dados revela a maior precisão dos modelos em classificar os registros com atraso, quando comparados aos modelos sem balanceamento.

Capítulo 7

Conclusão

Neste trabalho foi realizada uma avaliação experimental de diversos modelos de previsão baseados em aprendizado de máquina. O trabalho se iniciou a partir de uma extensa pesquisa exploratória sobre as diferentes técnicas de pré-processamento de dados, utilizando como base registros de voos comerciais nacionais brasileiros. Uma análise exploratória dos diversos métodos de aprendizado de máquina e uma avaliação experimental também foram desenvolvidos, com o objetivo de gerar um modelo de classificação capaz de prever atrasos aéreos.

A pesquisa se concentrou em realizar uma avaliação da literatura relacionada ao processo de mineração de dados e criação de modelos de classificação. Após o estudo realizado, foi desenvolvido um *workflow* para guiar todo o processo de pré-processamento e desenvolvimento dos modelos de classificação de atrasos, seguindo as métricas e padrões estabelecidos pela literatura.

A partir do *workflow* desenvolvido e do *dataset* construído, integrando as informações de voos, relação de feriados e dados meteorológicos, foi realizado o processo de limpeza, analisando os dados, para realizar a remoção de *outliers*. Em seguida, foram analisadas e testadas diversas transformações nos atributos para se adequarem e otimizarem a aplicação dos métodos de aprendizado de máquina. Após a definição das transformações a serem realizadas, foram aplicadas as técnicas de redução de dados para a criação dos *datasets* de treinamento e teste dos modelos de classificação.

Após a aplicação das técnicas de redução de dados, foram realizados os testes para gerar os modelos de classificação. No entanto, a acurácia similar à distribuição das classes e a baixa sensibilidade, ambas obtidas por meio da matriz de confusão dos modelos, indicaram a necessidade de um balanceamento nos dados. Portanto, foram analisadas formas de realizar o balanceamento dos dados e tornar a distribuição igualitária entre as classes do *dataset*, evitando que a precisão dos modelos de previsão fosse influenciada pela distribuição desbalanceada entre as classes.

Em seguida, dada a complexidade e o alto custo computacional de realizar todas as combinações de testes necessárias para cada método de aprendizado de máquina, foi escolhido o

método de redes neurais para a avaliação experimental do *workflow* desenvolvido. Os testes foram realizados utilizando os melhores parâmetros obtidos por meio da validação cruzada, seguida da análise de média-variância, para cada um dos *datasets* de treinamento.

Os resultados obtidos não foram superiores aos de um classificador simples, que classificaria todos os registros como sem atraso, obtendo uma porcentagem de acerto de 78%. No entanto, os modelos que receberam as técnicas de balanceamento tiveram um desempenho muito superior na previsão dos registros com atraso, obtendo cerca de 60% de acerto. Esses resultados apontam a importância da aplicação das técnicas de balanceamento de dados para aprimorar a capacidade do modelo gerado de prever os registros da classe de interesse, dado os resultados de sensibilidade obtidos. Também podemos constatar o impacto que um *dataset* desbalanceado pode gerar na acurácia dos modelos gerados, em que apenas cerca 5% dos registros da classe de interesse foram classificados corretamente.

O complexo contexto da aviação comercial, onde um atraso pode ser causado por diversos motivos relacionados as falhas no processo de voo, torna difícil a criação de bons modelos de previsão. Alguns fatores como a propagação de atrasos, problemas mecânicos, problemas na decolagem ou aterrissagem, estão fortemente ligados a ocorrência de atrasos, porém são difíceis de prever utilizando os dados disponíveis. Um estudo aprofundado sobre as particularidades da aviação comercial brasileira e como transformar essas particularidades em atributos que auxiliem a predição dos atrasos é necessário para o aprimoramento dos resultados, além de uma análise mais profunda no pré-processamento dos dados já coletados. Além disso, outras soluções para lidar com o desbalanceamento do *dataset* podem ser abordadas, como algoritmos que aumentam o custo dos erros de classificação da classe minoritária. Os trabalhos futuros se concentram em uma análise exploratória mais profunda dos dados, uma análise de agrupamento após o pré-processamento para analisar a efetividade do processo e a análise dos erros, com o objetivo de aprimorar o resultado obtido pelo classificador.

Referências Bibliográficas

ANAC (2015). Agência Nacional de Aviação Civil. Technical report, http://www.anac.gov.br/.

Balakrishna, P., Ganesan, R., and Sherry, L. (2010). Accuracy of reinforcement learning algorithms for predicting aircraft taxi-out times: A case-study of tampa bay departures. *Transportation Research Part C: Emerging Technologies*, 18(6):950 – 962.

Balakrishna, P., Ganesan, R., Sherry, L., and Levy, B. S. (2008). Estimating taxi-out times with a reinforcement learning algorithm. In *2008 IEEE/AIAA 27th Digital Avionics Systems Conference*, pages 3.D.3–1–3.D.3–12.

Belcastro, L., Marozzo, F., Talia, D., and Trunfio, P. (2016). Using scalable data mining for predicting flight delays. *ACM Trans. Intell. Syst. Technol.*, 8(1):5:1–5:20.

Bergstra, J., Bardenet, R., Bengio, Y., and Kégl, B. (2011). Algorithms for Hyper-Parameter Optimization. In J. Shawe-Taylor, R.S. Zemel, P. B. F. P. K. W., editor, *25th Annual Conference on Neural Information Processing Systems (NIPS 2011)*, volume 24 of *Advances in Neural Information Processing Systems*, Granada, Spain. Neural Information Processing Systems Foundation.

Bhatia, N. and Vandana (2010). Survey of nearest neighbor techniques. *CoRR*, abs/1007.0085.

Borovicka, T., Jr., M. J., Kordik, P., and Jirina, M. (2012). Selecting representative data sets. In Karahoca, A., editor, *Advances in Data Mining Knowledge Discovery and Applications*, chapter 02. InTech, Rijeka.

Breiman, L. (2001). Random forests. *Machine Learning*, 45(1):5–32.

Britto, R., Dresner, M., and Voltes, A. (2012). The impact of flight delays on passenger demand and societal welfare. *Transportation Research Part E: Logistics and Transportation Review*, 48(2):460 – 469.

Burch, C. (2001). A survey of machine learning. Technical report, Tech. report, Pennsylvania Governor's School for the Sciences, 2001. 4.

Chawla, N. V., Bowyer, K. W., Hall, L. O., and Kegelmeyer, W. P. (2002). SMOTE: synthetic minority over-sampling technique. *Journal of artificial intelligence research*, pages 321–357.

Chen, H., Wang, J., and Yan, X. (2008). A Fuzzy Support Vector Machine with Weighted Margin for Flight Delay Early Warning. In *Fuzzy Systems and Knowledge Discovery, 2008. FSKD '08. Fifth International Conference on*, volume 3, pages 331–335.

Cortiglioni, F., Mähönen, P., Hakala, P., and Frantti, T. (2001). Automated star-galaxy discrimination for large surveys. *The Astrophysical Journal*, 556(2):937.

Dhar, V. (2013). Data science and prediction. *Communications of the ACM*, 56(12):64–73.

Hall, M. A. (1998). Correlation-based feature selection for machine learning. Technical report.

Han, J., Kamber, M., and Pei, J. (2011). *Data Mining: Concepts and Techniques*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 3rd edition.

Hastie, T., Tibshirani, R., and Friedman, J. (2009). *The elements of statistical learning: data mining, inference and prediction*. Springer, 2 edition.

He, H. and Garcia, E. A. (2009). Learning from imbalanced data. *IEEE Transactions on Knowledge and Data Engineering*, 21(9):1263–1284.

Jagadish, H., Gehrke, J., Labrinidis, A., Papakonstantinou, Y., Patel, J. M., Ramakrishnan, R., and Shahabi, C. (2014). Big data and its technical challenges. *Communications of the ACM*, 57(7):86–94.

Jolliffe, I. (2002). *Principal component analysis*. Springer Verlag, New York.

Khanmohammadi, S., Chou, C. A., Lewis, H. W., and Elias, D. (2014). A systems approach for scheduling aircraft landings in jfk airport. In *2014 IEEE International Conference on Fuzzy Systems (FUZZ-IEEE)*, pages 1578–1585.

Lantz, B. (2013). *Machine Learning with R*. Packt Publishing.

Lu Zonglei, W. Jiandong, Z. G. (2008). A new method to alarm large scale of flights delay based on machine learning. In *Knowledge Acquisition and Modeling, 2008. KAM '08. International Symposium on*, pages 589–592.

Machado, E., Serqueira, M., Ogasawara, E., Ogando, R., Maia, M. A. G., da Costa, L. N., Campisano, R., Guedes, G. P., and Bezerra, E. (2016). Exploring machine learning methods for the star/galaxy separation problem. In *2016 International Joint Conference on Neural Networks (IJCNN)*, pages 123–130.

Matsudaira, K. (2015). The science of managing data science. *Communications of the ACM*, 58(6):44–47.

More, A. (2016). Survey of resampling techniques for improving classification performance in unbalanced datasets. *arXiv preprint arXiv:1608.06048*.

PostgreSQL Global Development Group (2015). PostgreSQL. Technical report, http://www.postgresql.org/.

Prati, R. C., Batista, G. E., and Monard, M. C. (2009). Data mining with imbalanced class distributions: concepts and methods. In *Indian International Conference Artificial Intelligence*, pages 359–376.

R Core Team (2013). *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria.

Rebollo, J. J. and Balakrishnan, H. (2014). Characterization and prediction of air traffic delays. *Transportation Research Part C: Emerging Technologies*, 44:231 – 241.

Ringnér, M. (2008). What is principal component analysis? *Nature biotechnology*, 26(3):303–304.

Sternberg, A., Carvalho, D., Murta, L., Soares, J., and Ogasawara, E. (2016). An analysis of brazilian flight delays based on frequent patterns. *Transportation Research Part E: Logistics and Transportation Review*, 95:282 – 298.

Tatibana, C. Y. and Kaetsu, D. Y. (2002). Uma introdução às redes neurais artificiais. *Disponível por WWW em http://www. din. uem. br/ia/neurais (dez. 1999).[TAT 99]*.

The Weather Company, 2016. The Weather Company, 2016. Weather Underground. Technical Report. https://www.wunderground.com/history/.

Tu, J. V. (1996). Advantages and disadvantages of using artificial neural networks versus logistic regression for predicting medical outcomes. *Journal of clinical epidemiology*, 49(11):1225–1231.

Wang, J. (2009). Mean-variance analysis: A new document ranking theory in information retrieval. In *Proceedings of the 31th European Conference on IR Research on Advances in Information Retrieval*, ECIR '09, pages 4–16, Berlin, Heidelberg. Springer-Verlag.

Wieland, F. (1997). Limits to growth: results from the detailed policy assessment tool [air traffic congestion]. In *Digital Avionics Systems Conference, 1997. 16th DASC., AIAA/IEEE*, volume 2, pages 9.2–1–9.2–8 vol.2.

Witten, I. H., Frank, E., and Hall, M. A. (2011). *Data Mining: Practical Machine Learning Tools and Techniques*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 3rd edition.