UNIVERSIDADE TECNOLÓGICA FEDERAL DO PARANÁ

DIEGO JOSE WOIDA

CLASSIFICAÇÃO DOS ATLETAS DA NCAA PARA O *DRAFT* DA NBA POR MEIO DE TÉCNICAS DE MINERAÇÃO DE DADOS

MEDIANEIRA

2021

DIEGO JOSE WOIDA

CLASSIFICAÇÃO DOS ATLETAS DA NCAA PARA O DRAFT DA NBA POR MEIO DE TÉCNICAS DE MINERAÇÃO DE DADOS

Classification of NCAA athletes for the NBA draft through Data Mining Techniques

Trabalho de conclusão de curso de graduação apresentada como requisito para obtenção do título de Bacharel em Ciências da Computação da Universidade Tecnológica Federal do Paraná (UTFPR).

Orientador: Prof. Dr. Paulo Lopes de Menezes.

Coorientador: Prof. Dr. Arnaldo Candido Junior.

MEDIANEIRA

2021

DIEGO JOSE WOIDA

PREDIÇÃO DAS ESCOLHAS DO *DRAFT* DA NBA POR MEIO DE TÉCNICAS DE MINERAÇÃO DE DADOS

Trabalho de conclusão de curso de graduação apresentada como requisito para obtenção do título de Bacharel em Ciências da Computação da Universidade Tecnológica Federal do Paraná (UTFPR).

Data de aprovação: 03/dezembro/2021

_____

Arnaldo Candido Junior
Doutor
Universidade Tecnológica Federal do Paraná

_____

Pedro Luiz de Paula Filho
Doutor
Universidade Tecnológica Federal do Paraná

_____

Fernando Schütz
Doutor
Universidade Tecnológica Federal do Paraná

MEDIANEIRA

2021

RESUMO

Uma das maneiras de entrada de novos atletas na National Basketball Association (NBA) é feita por meio do *draft* de ordem reversa. O *draft* foi proposto como uma forma de balancear as equipes da NBA, de modo que os times com pior desempenho na temporada atual têm acesso as primeiras escolhas na temporada seguinte. Devido ao grande volume de dados produzido nas partidas foi proposto a utilização de técnicas de mineração de dados para encontrar quais são os dados mais relevantes para escolha dos atletas e classificar quais os atletas serão selecionados no *draft* da NBA.

Palavras-chave: mineração de dados; classificação; basquetebol.

ABSTRACT

One of the ways new athletes can ingress in the National Basketball Association (NBA) is through reverse order draft. The draft was proposed as a way to balance NBA teams, so the worst-performing teams in the current season have access to the first choices the following season. Due to the large volume of data produced in the parties, it was proposed to use data mining techniques to find out which data are most relevant to the choice of the athletes and classify which athletes will be selected in the NBA draft.

Keywords: data mining; classification; basketball.

LISTA DE FIGURAS

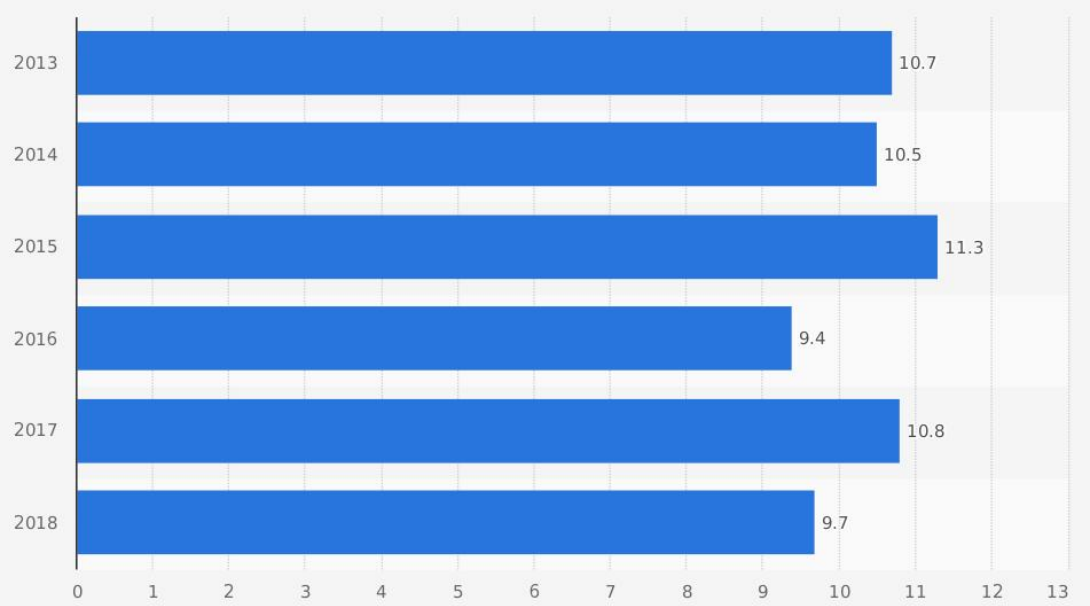
LISTA DE TABELAS

SUMÁRIO

1 INTRODUÇÃO

Criado em dezembro de 1891 por Naismith (1996), o basquete se tornou um dos esportes mais populares do mundo. A liga *National Collegiate Athletic Association* (NCAA) é a maior e mais importante liga universitária dos Estados Unidos da América (EUA), tendo os jogos decisivos do campeonato de basquete masculino ultrapassando 10 milhões de telespectadores nos últimos anos segundo dados da Statista[1] (Figura 1).

Figura 1 – Telespectadores em milhões



Fonte: STATISTA, 2019

Todos os anos diversos atletas da NCAA tentam assinar um contrato com alguma equipe da NBA, mas apenas os atletas de maior destaque conseguem ser selecionados. No *draft* da NBA, as equipes se revezam na seleção de um grupo de jogadores elegíveis. Quando uma equipe seleciona um jogador, essa equipe recebe direitos exclusivos para assinar esse contrato e nenhuma outra equipe na liga pode assinar com esse jogador.

O *draft* de ordem reversa foi proposto em 1935 por Bert Bell para a liga *National Football League* (NFL) e adotada no ano seguinte. No ano de 1947 a NBA finalmente adotou o

[1] https://www.statista.com/statistics/251560/ncaa-basketball-march-madness-average-tv-viewership-per-game/

draft de ordem reversa (FORT, 2011).

O objetivo deste modelo de escolha é promover um balanceamento entre as equipes, uma vez que as equipes que tiverem os piores resultados em uma temporada terão prioridade de escolha na temporada seguinte.

A mineração de dados possibilita a análise de grandes volumes de dados gerando resultados que não poderiam ser percebidos a olho nu. Os algoritmos computacionais utilizados na mineração de dados são formulados utilizando técnicas de Inteligência Artificial (IA). Essas técnicas focam a automação de atividades associadas com o pensamento humano, atividades como a tomada de decisão, resolução de problemas, aprendizado (CARVALHO, 2001).

1.1 **Objetivos geral e específicos**

Esse trabalho tem como objetivo classificar por meio de técnicas de mineração de dados as informações de jogadores e partidas com o objetivo de predizer quais serão as escolhas dos *drafts* futuros. Este objetivo principal pode ser dividido nos seguintes objetivos específicos:

- analisar os dados de partidas e jogadores da NCAA;
- definir os melhores dados para classificação dos atletas entre classes positivas (escolhidos) e negativas (não escolhidos no *draft*);
- aplicar os algoritmos de classificação: KNN, árvore de decisão j48, Naive Bayes, SVM, Random Forest e MLP;
- comparar os resultados dos algoritmos.

1.2 **Justificativa**

Todos os anos diversos atletas da NCAA são escolhidos no *draft* para atuar na NBA, que é uma das quatro maiores ligas de esporte profissional dos Estados Unidos e Canadá, onde se reúnem atletas de todas nacionalidades. Fazendo uma analise rápida da base de dados da NCAA observou-se um total de 21767 registros de atletas diferentes entre os anos de 2014-2017. Devido a este grande volume de dados produzidos, fica inviável uma análise

manual destes dados, portanto faz-se necessário uma solução por meio da computação. A Descoberta de Conhecimento em Bases de Dados (KDD) (do inglês, *Knowledge Discovery in Databases*) é uma solução para a analise de grandes volumes de dados, com ela é possível extrair novos conhecimentos de uma base de dados que irá ajudar em futuras escolhas. Com o desenvolvimento de um modelo de predição é possível auxiliar as equipes a realizarem melhores escolhas de atletas no *draft*.

1.3 **Organização do documento**

Este documento será organizado da seguinte forma: o Capítulo 2 apresentará uma contextualização teórica sobre os conceitos e características das técnicas de mineração de dados. Em seguida, são apresentados os trabalhos correlatos recentes, com o intuito de situar este trabalho no estágio atual do conhecimento. Os materiais e métodos utilizados se encontram no Capítulo 3, nele são descritas todas as etapas para o desenvolvimento do projeto.

No Capítulo 4 são descritos os experimentos preliminares, experimentos finais e os resultados dos mesmos. O Capítulo 5 detalha a conclusão deste projeto bem como algumas propostas para futuros trabalhos.

2 LEVANTAMENTO BIBLIOGRÁFICO

Nesta seção será descrito o estado da arte do tema escolhido. Primeiramente será dada uma breve introdução ao basquete e a escolha do *draft*, depois uma contextualização teórica sobre os conceitos e características das técnicas de mineração de dados. E, em seguida, serão apresentados os trabalhos correlatos recentes.

2.1 Basquete

Criado para atender a uma necessidade de praticar esporte durante os rigorosos invernos de Massachusetts, muitas das regras inicialmente proposta já sofreram alterações. Contudo, o objetivo principal permanece o mesmo, fazer com que a bola atravesse a cesta para marcar pontos[1].

Com o passar dos anos e a profissionalização do basquete foram sendo gerados dados estatísticos quanto ao desempenho das equipes e individualmente dos atletas, podendo assim analisar estes dados para detectar deficiências nas habilidades tanto individualmente quanto coletivamente.

Durante a análise individual de um atleta é observado a posição em que este atleta joga, informações como médias de arremessos convertidos, médias de lances livres, médias de arremessos de três pontos, roubos de bola, rebotes, assistências, bloqueios, faltas, tempo de jogo. Todas essas informações são importantes na hora de avaliar um atleta individualmente.

Além das informações de desempenho, outros dados como altura e idade também são extremamente importantes para escolha de um atleta por uma equipe da NBA durante o *draft*. Por exemplo, para um atleta ser considerado elegível para o *draft* é necessário que ele tenha idade igual ou superior a 19 anos. Atletas com uma altura muito elevada são muitas vezes escolhas preferíveis, já que é possível melhorar a habilidade de um atleta alto, porém não é

[1] http://www.cbb.com.br/a-cbb/o-basquete/as-primeiras-regras

possível aumentar a altura de um atleta habilidoso.

Todos estes pontos são analisados pelas equipes para a escolha de novos atletas durante *draft*.

2.2 **Mineração de Dados**

A mineração de dados consiste em analisar uma base de dados para descobrir padrões de comportamento e usar esta descoberta para exploração comercial. Normalmente, as vendas que foram feitas são o foco da mineração. Porém, a área comercial não é a única que pode aproveitar-se das técnicas de mineração de dados (INMON, 1997)

Segundo Issenberg (2013), em uma reportagem do MIT Technology Review, foram utilizadas técnicas de mineração de dados durante a campanha de reeleição de Barack Obama para a presidência dos EUA em 2012. Um dos modelos utilizados previa vitória com 56,4% dos votos no Condado de Hamilton, Ohio, e o resultado real da votação foi de 56,6%, tendo uma margem de erro de apenas 0,02% em relação à previsão. Isso permitiu que recursos da campanha fossem alocados de forma mais eficiente.

A mineração de dados é, na realidade, uma das etapas do KDD. O KDD é caracterizado como um processo composto pelas seguintes etapas: pré-processamento, a mineração de dados e pós-processamento (GOLDSCHMIDT; PASSOS, 2005).

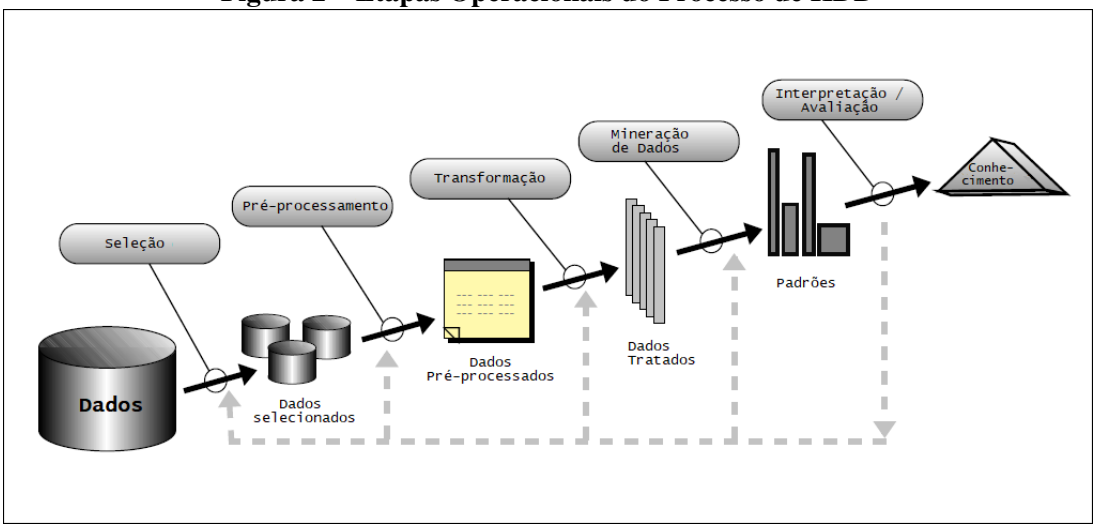
Para Fayyad (1996) a etapa de pré-processamento tem como objetivo a preparação dos dados para os algoritmos da etapa seguinte, a Mineração de Dados. Durante esta etapa, é realizada a busca efetiva por conhecimentos úteis no contexto da aplicação de KDD. A etapa de pós-processamento abrange o tratamento do conhecimento obtido durante esse processo. Tal tratamento, nem sempre necessário, tem como objetivo viabilizar a avaliação da utilidade do conhecimento descoberto.

Cada método de Mineração de Dados requer diferentes técnicas de pré-processamento. A escolha das técnicas de pré-processamento podem influenciar na qualidade do resultado do processo de KDD. Também é possível combinar diferentes técnicas para obter melhores resultados (GOLDSCHMIDT; PASSOS, 2005).

2.2.1 Descoberta de Conhecimento em Bases de Dados

O KDD ocorre de maneira interativa, ou seja, por meio da atuação do homem utilizando recursos computacionais em função da análise e interpretação dos dados. E também iterativa, podendo ter repetições integrais ou parciais do processo de KDD afim de buscar melhores resultados (FAYYAD *et al.*, 1996; HAN *et al.*, 2011).

Figura 2 – Etapas Operacionais do Processo de KDD



Fonte: Adaptado de Fayyad *et al.* (1996)

Segundo Brachman e Anand (1996), Han *et al.* (2011) o processo de descoberta de conhecimento em bases de dados representado na Figura 2 é explicado na sequência de iteração dos seguintes passos:

1. **Dados:** desenvolver um entendimento do conjunto de dados e identificar o objetivo do processo do KDD;
2. **Seleção:** selecionar um conjunto de dados ou concentrando-se em um subconjunto de variáveis ou amostras de dados, no qual a descoberta deve ser realizada;
3. **Pré-processamento:** é feita a limpeza dos dados para remoção de ruídos, tratamento de dados ausentes, limitação de valores possíveis para determinados campos;
4. **Transformação:** última fase da preparação dos dados antes da mineração. Pode ser aplicado redução de dimensionalidade ou mudança dos dados de Numérica - Categórica, de valores reais em categorias ou intervalos; ou Categórica - Numérica, representa numericamente valores de atributos categóricos;
5. A quinta etapa será a escolha do método de mineração que será utilizado posteriormente.

Por exemplo, associação, classificação, regressão, clusterização, sumarização, detecção de desvios, descoberta de sequências;

6. **Mineração de Dados:** é a análise exploratória, seleção de modelos e hipóteses. Na qual deverá ser feita uma busca efetiva por padrões nos dados, incluindo regras de classificação ou árvores, regressão e agrupamentos. A execução correta das etapas anteriores irá ajudar significativamente no resultado do método de mineração de dados;
7. **Interpretação/Avaliação:** a oitava etapa é a interpretação dos padrões minerados, possivelmente retornando a qualquer uma das etapas para novas iterações. Essa etapa também pode envolver a visualização dos padrões e modelos extraídos;
8. **Conhecimento:** usar o conhecimento obtido, seja incorporando o conhecimento em outro sistema para ação futura, ou simplesmente documentando-o e relatando-o às partes interessadas.

Para Goldschmidt e Passos (2005) o processo de KDD deve ser classificado quanto à orientação das ações a serem realizadas e o macroobjetivo pretendido, na qual a orientação das ações pode ser para validação de hipóteses postuladas ou a descoberta de conhecimento, e o macroobjetivo em predição ou descrição.

- **Validação de hipóteses postuladas:** Será apresentada uma hipótese que será contestada, chegando a uma conclusão mediante a análise dos dados;
- **Descoberta de conhecimento:** Procura de conhecimentos úteis a partir da abstração dos dados existentes;
- **Predição:** prever os valores de determinados atributos em novas situações, com base em um histórico de casos prévios;
- **Descrição:** busca-se por um modelo que represente, de forma compreensível pelo homem, o conhecimento existente em um conjunto de dados.

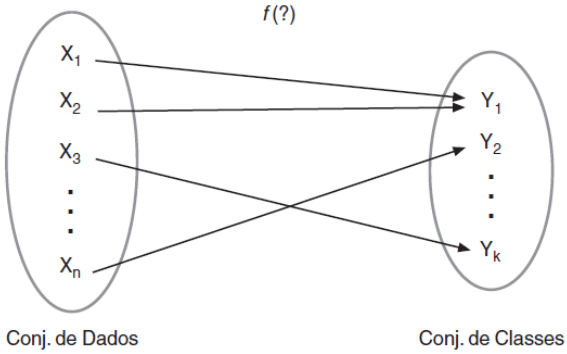
2.2.2 Métodos de Mineração de Dados

Os dois principais objetivos durante a mineração de dados tendem a ser predição e descrição. Na qual a predição (Classificação, Regressão) tenta prever valores desconhecidos ou valores futuros de determinadas variáveis, usando variáveis ou campos do banco de dados. Enquanto a descrição (Associação, Clusterização e Sumarização) se concentra em encontrar padrões que sejam facilmente interpretados por humanos que descrevam os dados (FAYYAD *et*

al., 1996; HAN *et al.*, 2011). Os principais métodos de mineração são:

- Classificação: É o método mais popular e importante, a classificação busca uma função que mapeia (classifica) um item de dados em uma das várias classes predefinidas, conforme mostra a Figura 3. Após encontrar tal função é possível aplicá-la em novos registros com o objetivo de prever a classe a qual pertence (FAYYAD *et al.*, 1996; HAN *et al.*, 2011). Considerando um par ordenado na forma *(x, f(x))*, no qual *x* é um vetor de entradas de *n*-dimensões e *f(x)* é a saída da função *f*, aplicada a *x*. Tem-se neste vetor *n*-valores que serão os dados analisados para determinar tal função *f*. A acurácia retrata a qualidade ou a precisão da função encontrada em mapear corretamente cada vetor de entradas *x* em *f(x)*. O conjunto de pares *(x, f(x))* utilizados na identificação da função é denominado conjunto de treinamento (GOLDSCHMIDT; PASSOS, 2005; HAN *et al.*, 2011).

Figura 3 – Associações entre registros de dados e classes



Fonte: Goldschmidt e Passos (2005)

- Regressão: a tarefa de regressão busca ajustar uma função, linear ou não, que mapeie os registros de um banco de dados em valores reais. Essa tarefa é similar à tarefa de classificação, sendo restrita apenas a atributos numéricos (GOLDSCHMIDT; PASSOS, 2005; HAN *et al.*, 2011). As aplicações de regressão são muitas, por exemplo: predição da soma da biomassa presente em uma floresta; prevendo a demanda do consumidor por um novo produto; predição do risco de determinados investimentos, definição do limite do cartão de crédito para cada cliente em um banco; dentre outros (FAYYAD *et al.*, 1996; GOLDSCHMIDT; PASSOS, 2005).
- Regras de Associação: nas últimas décadas, diversos pesquisadores propuseram algoritmos de mineração de regras de associação, os quais tem um grande domínio de aplicações reais, dentre negócios, finanças, economia, biologia e medicina (AGGARWAL, 2014). O método de associação, ou, análise de afinidade, procura descobrir associações entre atributos, isto é, procura descobrir regras para quantificar

o relacionamento entre dois ou mais atributos. As regras de associação tomam a forma "SE X, ENTÃO Y", juntamente com uma medida do suporte e da confiança associados à regra (GOLDSCHMIDT; PASSOS, 2005; LAROSE; LAROSE, 2014). Por exemplo, de 1.000 clientes que compraram em um determinado dia, 200 compraram fraldas e desses 200, 50 compraram cervejas. Sendo assim a regra de associação seria "SE compra fraldas, ENTÃO compra cerveja", com um suporte de $50/1000 = 5\%$ e uma confiança de $50/200 = 25\%$ (LAROSE; LAROSE, 2014).

- Clusterização: também chamada de agrupamento, não tenta classificar, estimar ou prever o valor. Os algoritmos de *clustering* procuram particionar os registros de uma base de dados em subconjuntos ou *clusters*, de tal forma que elementos em um *cluster* sejam semelhantes uns aos outros, mas diferentes dos elementos de outros *clusters* (LAROSE; LAROSE, 2014; HAN *et al.*, 2011). Em geral, o processo de agrupamento requer que o usuário especifique um número x de grupos a ser considerado. Com base nisso, os elementos da base de dados serão separados de forma que elementos parecidos fiquem nos mesmos grupos e elementos diferentes em outros grupos. Após a execução do algoritmo, é possível fazer a análise dos elementos de cada grupo, identificando os atributos comuns aos seus elementos e, assim, sendo possível criar um rótulo que identifica cada grupo (GOLDSCHMIDT; PASSOS, 2005).
- Sumarização: também denominada descrição de conceitos, envolve métodos para encontrar uma descrição compacta e compreensível, ou seja, as principais características dos dados contidos em um subconjunto de dados (FAYYAD *et al.*, 1996). Goldschmidt e Passos (2005) citam dois exemplos de aplicações envolvendo a sumarização. O primeiro exemplo consiste em classificar os assinantes de uma revista que residem em uma região específica do Brasil: "são em grande maioria, assinantes com faixa salarial de X reais, nível superior completo e que possuem residência própria". O segundo exemplo é traçar o perfil dos meninos de rua da cidade do Rio de Janeiro: "são meninos que se encontram predominantemente na faixa etária X, cujos pais utilizam drogas e possuem na faixa de Y irmãos".

2.2.3 KNN

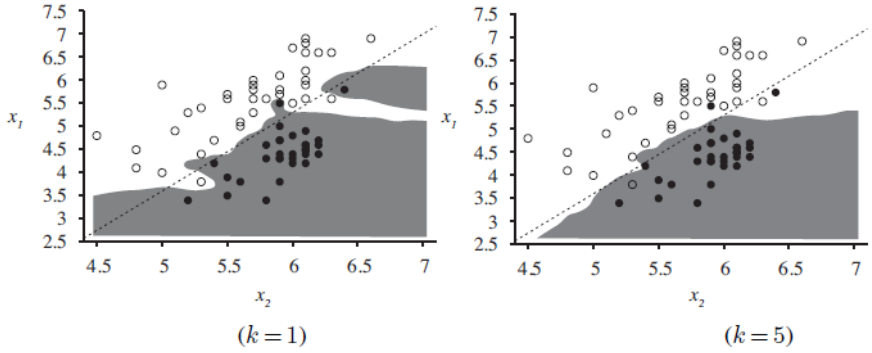
O algoritmo *k*-vizinhos mais proximos (do inglês, k-Nearest Neighbor (KNN)) é um

método de inferência altamente eficaz para muitos problemas da vida real. Ele pode ser encontrado realizando uma consulta *xq*, onde será encontrado os *k* exemplos mais próximos de *xq* (MITCHELL, 1997; RUSSELL; NORVIG, 2010).

Para problemas de classificação, inicialmente é o identificado o conjunto de k-vizinhos mais próximos, denominado $NN(k,xq)$, em seguida é realizada a votação de pluralidade dos vizinho (onde o conjunto com maior número de votos é escolhido). Sempre deverá ser escolhido um valor ímpar para *k*, assim evitando empates durante a votação (RUSSELL; NORVIG, 2010).

Na Imagem 4 tem-se um exemplo do algoritmo KNN para classificação com $k = 1$ e $k = 5$ de um *dataset* em duas classes.

Figura 4 – Exemplo de Classificação com KNN



Fonte: (RUSSELL; NORVIG, 2010)

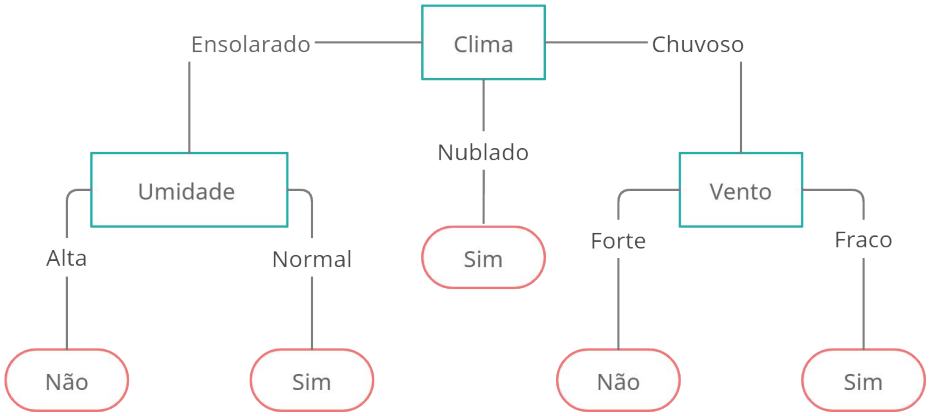
2.2.4 C4.5

Os algoritmos de árvores de decisão são uma das formais mais simples de aprendizado de máquina. A árvore de decisão classifica as instâncias da raiz da árvore até algum nó folha que deverá retorna a classificação da instância de entrada. Cada nó da árvore faz um teste de algum dos atributos do vetor de entrada. A cada entrada o vetor percorrerá a árvore iniciando pela raiz testando o atributo especifico desse nó e movendo-se para baixo na árvore correspondente ao atributo testado. O algoritmo C4.5 é atualmente o mais usado, foi proposto por Quinlan em 1993 (GOODFELLOW *et al.*, 2016; RUSSELL; NORVIG, 2010).

Para a descoberta de conhecimento são propostas hipóteses, as quais são testadas e selecionadas as melhores. Essa busca pelas hipóteses é feita de forma recursiva utilizando a

técnica de dividir para conquistar. Na Figura 5 tem-se a representação de uma árvore de decisão referente a abstração JogarTenis (WITTEN *et al.*, 2011).

Figura 5 – Árvore de decisão para a abstração JogarTênis



Fonte: adaptado de Mitchell (1997)

2.2.5 Random Forrest

O *Random Forest* é um algoritmo que combina o popular algoritmo *Random Tree*, o qual testa um determinado número de atributos aleatórios em cada nó sem realizar nenhuma poda. O *Random Forest* portanto cria uma floresta aleatória com conjuntos de árvores criados com o *Random Tree* (WITTEN *et al.*, 2011).

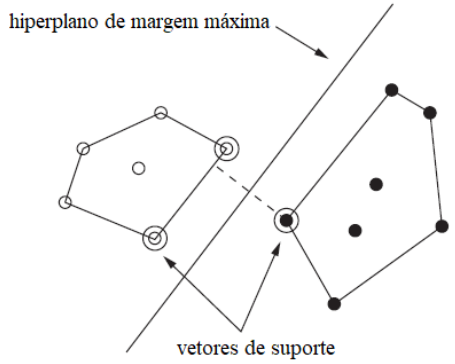
2.2.6 Support Vector Machine

Support Vector Machine (SVM) é atualmente a método mais popular para o aprendizado supervisionado, ele utiliza de modelos lineares para implementar limites de classe não lineares. São baseados em um algoritmo que resolve um tipo particular de modelo linear: o hiperplano de margem máxima. O hiperplano de margem máxima é aquele que dá uma ótima segmentação entre as categorias, não aproxima-se mais perto do que deveria de qualquer uma

das duas categorias (RUSSELL; NORVIG, 2010; WITTEN *et al.*, 2011).

Os pontos mais relevante dos SVMs são que alguns exemplos são mais importantes do que outros e, adaptar-se a eles pode obter uma melhor abstração do modelo. Outro ponto importante dos SVMs é a utilização dos *kernels*, frequentemente os dados não são linearmente separáveis no espaço de entrada original, porém são facilmente separáveis em um espaço de maior dimensão, essa técnica de aumentar a dimensão é feita utilizando *kernels* (RUSSELL; NORVIG, 2010; WITTEN *et al.*, 2011).

Figura 6 – Hiperplano de margem máxima



Fonte: adaptado de Witten *et al.* (2011)

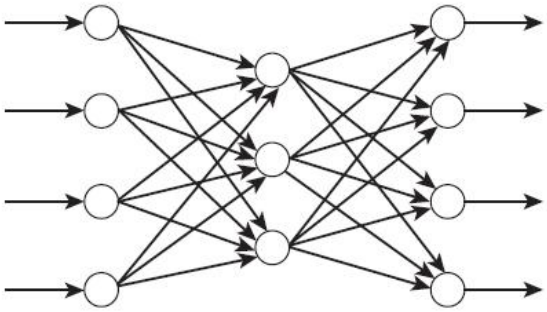
2.2.7 MultiLayer Perceptron

O *perceptron* construído em torno de um único neurônio é limitado a classificação em apenas duas categorias. Ele pode ter qualquer quantidade de entradas, às quais normalmente são estruturadas em grade. Esta grade pode ser usada para representar um campo de visão, portanto, os *perceptrons* podem ser usados para realizar tarefas simples de classificação (COPPIN, 2004; HAYKIN, 2009).

Entretanto, a maioria dos problemas do mundo real não são linearmente separáveis. Embora os perceptrons sejam um modelo relevante para estudar a maneira como os neurônios artificiais podem funcionar, algo mais eficiente é necessário. Um único *perceptron* pode ser considerado um *perceptron* de camada única, para modelar funções mais complexas é necessário o uso de *Perceptrons* multicamadas (em inglês, MultiLayer Perceptron), na Imagem 7 é apresentada uma a rede neural do tipo *feed-foward* de 3 camadas. O principal método de

treinamento usado no MPL é o algoritmo *back-propagation* (COPPIN, 2004; HAYKIN, 2009).

Figura 7 – MultiLayer Perceptron de 3 camadas



Fonte: (COPPIN, 2004)

2.2.8 Algoritmo Bayesiano

O algoritmo bayesiano tem origem no teorema de Thomas Bayes, esse teorema fornece uma maneira de calcular a probabilidade de uma hipótese com base em sua probabilidade anterior, as probabilidades de considerar vários dados a hipótese e os próprios dados considerados. A notação $P(B|A)$ pode ser lida como "a probabilidade de B, dado A", e pode ser definida como (COPPIN, 2004; GOODFELLOW *et al.*, 2016):

$$\text{Problema original: } P(B|A) = \frac{P(B \wedge A)}{P(A)}$$

O teorema de Bayes nos permite calcular o único termo $P(B|A)$ em termos de três termos: $P(A|B)$, $P(B)$ e $P(A)$. A regra de Bayes é útil na prática porque há muitos casos em que temos as estimativas de probabilidade para esses três números e precisamos calcular o quarto. Nesse caso, a regra de Bayes torna-se (GOODFELLOW *et al.*, 2016; RUSSELL; NORVIG, 2010):

$$\text{Teorema de Bayes: } P(B|A) = \frac{P(A|B) * P(B)}{P(A)}$$

O modelo Bayes ingênuo (do inglês, Naïve Bayes), "ingênuo" porque é frequentemente usado (como uma suposição simplificadora) em casos em que as variáveis de "efeito" não são realmente condicionalmente independentes dada a variável de causa. O

classificador ingênuo Bayes é um sistema de aprendizagem simples, mas eficaz. Cada dado a ser classificado consiste em um conjunto de atributos, cada um dos quais pode assumir uma série de valores possíveis. Os dados são então classificados em uma única classificação. O Naïve Bayes pode ser definido como (MITCHELL, 1997; RUSSELL; NORVIG, 2010):

$$\text{Naïve Bayes: } P(A_1|B)P(A_2|B)...P(A_n|B)P(B)$$

2.3 **Pré-processamento**

Esta etapa é de extrema importância no processo de descoberta de conhecimento, pois o pré-processamento abrange as funções relacionadas à captação, à organização, ao tratamento e à preparação dos dados para a etapa da Mineração de Dados (GOLDSCHMIDT; PASSOS, 2005). Muitos dos dados brutos contidos nos bancos de dados não são padronizados, incompletos ou ruídos. Por exemplo, os bancos de dados podem conter (LAROSE; LAROSE, 2014):

- Campos que são obsoletos ou redundantes;
- Campos vazios;
- *Outliers*;
- Dados em um formato não adequado para os modelos de mineração de dados.

Conforme o conjunto de dados, a etapa de pré-processamento sozinha pode corresponder de 10 a 60% de todo o tempo e esforço durante o processo de mineração de dados (LAROSE; LAROSE, 2014).

2.3.1 Dados

O resultado da mineração de dados e KDD depende muito da qualidade e da quantidade de dados disponíveis. Os dados são informações armazenadas por recursos de Tecnologia da Informação (TI). Eles podem ter diversos formatos e serem armazenados usando diversos modos de armazenamento diferentes. Por exemplo, planilhas, banco de dados, *data warehouse*, textos (GOLDSCHMIDT; PASSOS, 2005; HAN *et al.*, 2011; HAND, 2007).

Para facilitar a compreensão, usando um conjunto de dados fictícios referente aos

clientes de uma financeira. A Tabela 1 apresenta a estrutura deste conjunto de dados. Cada linha corresponde às instâncias e as colunas correspondem aos atributos (GOLDSCHMIDT; PASSOS, 2005; HAN *et al.*, 2011).

Tabela 1 – Estrutura do Conjunto de Dados de Clientes

Atributo	Tipo de Dado	Descrição
CPF	Char(11)	Número do CPF do Cliente. Formato: XXX.XXX.XXX-XX
Nome	VarChar(50)	Nome completo do Cliente
Sexo	Char(1)	M – Masculino F – Feminino
Data_Nasc	Date	Data de nascimento do Cliente. Formato: DD/MM/AAAA
Est_Civil	Char(1)	C – Casado S – Solteiro V – Viúvo D – Divorciado O – Outro
Num_Dep	Integer	Quantidade de pessoas que dependem financeiramente do Cliente.
Renda	Real	Total dos rendimentos mensais do Cliente.
Despesa	Real	Total das despesas mensais do Cliente.

Fonte: Adaptado de Goldschmidt e Passos (2005)

Segundo Han *et al.* (2011), Goldschmidt e Passos (2005) pode-se classificar classificar os atributos em alguns tipos, entre eles nominais (ou categóricos), discretos, contínuos e binários.

- **Nominais (ou Categóricos):** fazem referência a nomes ou rótulos de objetos. Cada valor representa algum tipo de categoria, código ou estado, por isso também são chamados de categóricos. Por exemplo, cor (preto, marrom, vermelho, cinza e branco), profissão (professor, estudante, programador, motorista). Os atributos *CPF*, *Nome*, *Sexo* e *Est_Civil* da Tabela 1 são outros exemplos de atributos nominais.
- **Discretos:** os atributos discretos são parecidos com os nominais, porém, eles possuem um ordenamento. Por exemplo, os dias da semana (domingo, segunda, ..., sábado), temperatura (quente, agradável, frio e muito frio), tamanho (pequeno, grande, médio). Na Tabela 1 tem-se *Data_Nasc* e *Num_Dep*: são exemplos de atributos discretos.
- **Binários:** novamente tem-se um tipo de atributo parecido com os nominais, porém, possuí apenas duas categorias (0 ou 1), no qual 1 quer dizer que o atributo está presente e 0 não. Também podem ser retratados como *booleanos* (verdadeiro ou falso). Por exemplo, no conjunto de dados fictícios, em vez de armazenar a quantidade de dependentes fosse armazenado apenas se possuí ou não dependentes.
- **Contínuos:** são atributos quantitativos, ou seja, é uma quantidade calculável. Eles

representam valores inteiros ou reais, por exemplo temperatura em graus Celsius, datas do calendário. No conjunto de dados fictício tem-se os atributos contínuos *Renda* e *Despesas*.

Entretanto, os dados em uma base de dados comum sofrem alterações contantes, o que atrapalhará no processo de KDD. É recomendando portanto que esses dados sejam extraídos e organizados em um única tabela bidimensional para aplicação dos métodos de KDD (GOLDSCHMIDT; PASSOS, 2005; HAND, 2007).

2.3.2 Limpeza

Em aplicações de base de dados reais, é normal que contenham dados incompletos, ruidosos ou inconsistentes sendo assim a realização da limpeza desses dados. Goldschmidt e Passos (2005) consideram que os dados são incompletos se há campos vazios para determinados atributos ou ainda se há dados pouco detalhados. Ruídos são dados errados ou que contenham valores considerados dispares, denominados *outliers*, do padrão esperado. Dados inconsistentes são aqueles que contêm algum tipo de divergência semântica entre si.

Goldschmidt e Passos (2005) e Larose e Larose (2014) consideram que a qualidade dos dados é de extrema importância, pois possuí grande influência no resultado dos modelos de conhecimento gerados a partir destes dados. Quanto mais ruídos, *outliers* ou campos redundantes possuir entre os dados informados ao processo de KDD, pior será a qualidade dos modelos de conhecimento gerados (GIGO – *Garbage in*, *Garbage out*).

2.3.2.1 Limpeza de Informações Ausentes

Mesmo os métodos de análise mais sofisticados continuam tendo problemas ao encontrar valores ausentes nos campos, especialmente em bancos de dados com uma grande quantidade de campos. E essa ausência de dados raramente é benéfica (LAROSE; LAROSE, 2014).

Entra as abordagens que podem ser feitas Larose e Larose (2014), Han *et al.* (2011), Goldschmidt e Passos (2005) citam três técnicas em comum: Preenchimento com Valores

Globais Constantes, Preenchimento com Medidas Estatísticas e Preenchimento com Métodos de Mineração de Dados. Já o Preenchimento Manual de Valores é citado por Goldschmidt e Passos (2005), Han *et al.* (2011).

- **Exclusão de Casos:** um dos métodos citados apenas por Goldschmidt e Passos (2005) consiste em excluir do conjunto de dados as tuplas que possuam pelo menos um atributo não preenchido;
- **Preenchimento Manual de Valores:** inviável na prática, visto que para ser feito este processo é necessário realizar pesquisa junto às fontes originais e feita a digitação manual de todos os dados;
- **Preenchimento com Valores Globais Constantes:** substitui todos os valores de atributos ausentes pela mesma constante, como um rótulo como "desconhecido" ou "*null*". Esse valor pode e deve ser especificado pelo especialista no domínio da aplicação;
- **Preenchimento com Medidas Estatísticas:** como alternativa à utilização de contantes globais, podem-se usar medidas estatísticas para preencher os valores ausentes com a média (atributos numéricos) ou moda (atributos categóricos);
- **Preenchimento com Métodos de Mineração de Dados:** mesmo durante a etapa de pré-processamento, algoritmos de Mineração de Dados podem ser utilizados para preenchimento de valores ausentes. Regressão, Árvore de Decisão, Redes Neurais, Modelos Bayesianos são alguns exemplos que podem ser utilizados;
- **Preenchimento com Valores Gerados 'Aleatoriamente':** Larose e Larose (2014) coloca que pode-se substituir os valores ausentes por um valor gerado aleatoriamente a partir da distribuição observada da variável. Porém, não há garantia de que os registros resultantes façam sentido.

2.3.2.2 Limpeza de Inconsistências

Uma inconsistência pode envolver uma única tupla, ou um conjunto de tuplas. A inconsistência em uma única tupla ocorre quando houver discrepância entre os valores desta tupla (GOLDSCHMIDT; PASSOS, 2005).

Por exemplo, em uma pequena base de dados de distribuição de frequência da variável "marca" foi mostrada na Tabela 2. Pode-se analisar as cinco classes, USA, França, US, Europa e Japão. Porém, duas das classes, USA e França, contam apenas com um automóvel cada. O que

aconteceu aqui, foi claramente um erro na classificação das classes USA e França (LAROSE; LAROSE, 2014).

Tabela 2 – Frequência de Marcas

Marca	Frequência
USA	1
França	1
US	156
Europa	46
Japão	51

Fonte: Adaptado de Larose e Larose (2014)

Goldschmidt e Passos (2005) propõem duas abordagem, a Exclusão de Casos, similar a Limpeza de Valores Ausentes descrito anteriormente, consiste em excluir do conjunto de dados original, as tuplas que possuam pelo menos uma discrepância. A outra abordagem seria a Correção de Erros também é citada por Larose e Larose (2014), consiste em substituir valores errôneos ou discrepâncias encontradas no conjunto de dados. No exemplo adaptado de Larose e Larose (2014) basta corrigir USA para US e França para Europa.

2.3.2.3 Limpeza de Valores não pertencentes ao Domínio

Está função pode ser considerada um caso particular da operação Limpeza de Inconsistências. Por exemplo, um determinado conjunto de dados com o atributo valor do produto como negativo (LAROSE; LAROSE, 2014). Novamente, Goldschmidt e Passos (2005) propõem as duas abordagem, Exclusão de Casos e Correção.

2.3.3 Transformação (ou Codificação)

Geralmente os dados não são homogêneos, ou seja, eles vão ter vários tipos e é crucial para o processo de Mineração de Dados que eles sejam transformados de acordo com os métodos que serão aplicados (AGGARWAL, 2015; GOLDSCHMIDT; PASSOS, 2005).

Os valores das variáveis podem variar muito um do outro. Por exemplo, no caso da

NCAA Basketball, as médias de arremessos de três pontos convertidos variam de 0 a 0.42[2], enquanto o número de pontos pode ultrapassar 80[3]. Essas diferenças nos intervalos dos valores levarão a uma tendência de a variável "pontos" ter uma influência nos resultados obtidos (LAROSE; LAROSE, 2014).

É de responsabilidade do profissional a normalização dos dados, existem algumas técnicas para isto, as principais transformações são: *Numérica – Categórica*, *Categórica – Numérica* e *Normalização dos Dados* (GOLDSCHMIDT; PASSOS, 2005).

2.3.3.1 Transformação Numérica – Categórica

A transformação de dados numéricos para categóricos também é chamada de discretização e é uma das técnicas mais comuns. Ela consiste em dividir os valores em grupos e associar estes grupos a um determinado valor simbólico. Por exemplo para um conjunto de dados referente a renda é possível dividir em faixas de valores [0.00, 2000.00], [2000.01, 5000.00], [5000.01, ...] e associar os dados dentro dessas faixas a um valor simbólico, a Tabela 3 mostra um exemplo. Entre as opções da definição dos intervalos temos: definido pelo usuário, intervalos iguais, por meio de clusterização (AGGARWAL, 2015).

Tabela 3 – Transformação Numérica – Categórica

Intervalo	Valor simbólico
[0,00, 2000,00]	Renda baixa
[2000,01, 5000,00]	Renda média
[5000,01, ...]	Renda alta

Fonte: autoria própria (2021)

2.3.3.2 Transformação Categórica – Numérica

Para aplicação de alguns métodos de mineração de dados é necessário a transformação de dados categóricos em numéricos, isso é feito com a binarização, ou seja, representar as

[2]https://www.ncaa.com/stats/basketball-men/d1/current/team/152
[3]https://www.ncaa.com/stats/basketball-men/d1/current/team/145

categorias por valores binários. Por exemplo para o atributo “Est_Civil” da Tabela 1 tem-se a seguinte transformação (GOLDSCHMIDT; PASSOS, 2005; AGGARWAL, 2015).

Tabela 4 – Transformação Categórica – Numérica

Valores Originais	Representação Binária
Casado	(0, 0, 1)
Solteiro	(0, 1, 0)
Viúvo	(1, 0, 0)
Divorciado	(0, 1, 1)
Outro	(1, 1, 0)

Fonte: (GOLDSCHMIDT; PASSOS, 2005)

2.3.3.3 Normalização de Dados

Em alguns conjuntos de dados é necessário fazer a normalização para evitar que certos atributos com uma escala de valores maior interfiram no resultado da mineração de dados, então os dados são normalizados em intervalos pequenos, como -1 a 1 ou 0 a 1. As técnicas de normalização utilizadas segundo Goldschmidt e Passos (2005), Larose e Larose (2014) são:

- **Normalização Linear:** Também chamada de normalização Min-Max, ela consiste em verificar o valor máximo e mínimo de um atributo e realizar o seguinte cálculo:

$$X' = \frac{X - \min(X)}{\max(X) - \min(X)} \tag{1}$$

  No qual:
  X′ = Valor normalizado;
  X = Valor do atributo a ser normalizado;
  min(X) = Valor mínimo do atributo a ser normalizado;
  max(X) = Valor máximo do atributo a ser normalizado;

- **Normalização por Desvio Padrão:** Muito utilizado em análise estatística, funciona calculando a diferença entre o valor do campo e o valor médio do campo, e dividindo pelo desvio padrão. É muito útil quando os valores de mínimo e máximo do atributo são desconhecidos.

$$X' = \frac{X - \text{media}(X)}{\text{DP}(X)} \tag{2}$$

No qual:

X′ = Valor normalizado;

X = Valor do atributo a ser normalizado;

media(X) = Valor médio do atributo a ser normalizado;

DP(X) = Desvio padrão do atributo a ser normalizado;

- **Normalização pela Soma dos Elementos:** É calculado o somatório dos valores do atributo, depois cada valor é dividido por este somatório.

$$X' = \frac{X}{\text{soma}(X)} \tag{3}$$

No qual:

X′ = Valor normalizado;

X = Valor do atributo a ser normalizado;

soma(X) = Somatório dos valores do atributo a ser normalizado;

- **Normalização pelo Valor Máximo dos Elementos:** Cada valor do atributo é dividido pelo valor máximo do atributo.

$$X' = \frac{X}{\max(X)} \tag{4}$$

No qual:

X′ = Valor normalizado;

X = Valor do atributo a ser normalizado;

max(X) = Valor máximo do atributo a ser normalizado;

- **Normalização por Escala Decimal:** É garantido que todo valor normalizado esteja entre -1 e 1.

$$X' = \frac{X}{10^i} \tag{5}$$

No qual:

X′ = Valor normalizado;

X = Valor do atributo a ser normalizado;

i = Número de dígitos no valor de dados com o maior valor absoluto.

Por exemplo, maior valor absoluto é $|95786| = 95786$, que possuí 5 dígitos. A escala decimal para o valor mínimo de 1245 e máximo de 95786.

$$\text{Min: } X' = \frac{1245}{10^5} = 0.01245$$
$$\text{Max: } X' = \frac{95768}{10^5} = 0.95768$$

2.3.4 Seleção de Atributos

Um volume grande de dados pode-se tornar a mineração e análise muito demorada e custosa, tornando-a assim inviável. Existem algumas técnicas de redução de dados, no qual é possível atingir uma representação sucinta do conjunto de dados, mas mantendo a representatividade dos dados originais (HAN *et al.*, 2011).

Os conjuntos de dados podem ser reduzidos de forma Horizontal e Vertical, na qual Redução de Dados Horizontal é caracterizada pela escolha de casos, por exemplo, amostragem aleatória, eliminação direta de casos, segmentação do banco de dados e agregação de informações. E, a Redução de Dados Vertical, consiste em reduzir o número de dimensionalidades, ou seja, é o processo para reduzir o número de variáveis aleatórias ou atributos em consideração (GOLDSCHMIDT; PASSOS, 2005; HAN *et al.*, 2011).

Existem uma grande variedade técnicas de limpeza, transformação e normalização dos dados, cada uma é mais recomendada para certos métodos de mineração de dados, cabe ao especialista analisar e identificar quais são mais adequadas a cada problema.

2.4 **Pós-processamento**

Na fase de pós-processamento, o especialista em KDD vai fazer a interpretação dos padrões descobertos, podendo retornar a qualquer uma das fases anteriores, também são criadas representações destes padrões de forma que fiquem compreensíveis aos usuários. Algumas abordagens que são feitas nesta etapa: Avaliação dos Padrões, Simplificação do Modelo de Conhecimento, Transformação do Modelo de Conhecimento, Organização e Apresentação dos

Resultados (FAYYAD *et al.*, 1996; LAROSE; LAROSE, 2014).

2.4.1 Avaliação dos Modelos de Conhecimento

Nem todo o conhecimento descoberto é realmente útil, durante o processo de KDD pode-se encontrar muitos conhecimentos que já são de senso comum, portanto é importante que a avaliação dos padrões encontrados quanto à qualidade e eficácia. Os métodos de Mineração de Dados possuem técnicas de avaliação específicas, que podem ser divididas em grupos com base na natureza do método, algumas delas são (LAROSE; LAROSE, 2014; HAND, 2007):

- **Estimativa ou previsão:** erro quadrático médio (*Mean Squared Error* - MSE);
- **Classificação e Clusterização:** taxa de erros, falsos positivos, falsos negativos, ajuste de custo de erro;
- **Regras de Associação:** suporte e confiança.

2.4.2 Simplificação do Modelo de Conhecimento

No KDD, é comum que os conhecimentos descobertos sejam exibidos por meio de regras. Porém, conjuntos de dados grandes tendem a uma quantidade de regras elevadas, tornando a interpretação uma tarefa árdua. Essas regras são representadas na forma "SE X, ENTÃO Y", no qual X e Y são condições que podem se tornar verdadeiras ou falsas em função de cada registro. Existem alguns métodos voltados a simplificação de regras, as principais são a Precisão da Regra e a Abrangência da Regra (GOLDSCHMIDT; PASSOS, 2005).

- **Precisão:** é calculada por meio da divisão da quantidade de vezes em que as regras X e Y aparecerem pela quantidade de vezes que aparece X.

$$\text{Precisão} = \frac{|X \wedge Y|}{|X|}. \tag{6}$$

- **Abrangência:** é calculada por meio da divisão da quantidade de vezes em que as regras X e Y aparecerem pela quantidade de vezes que aparece Y.

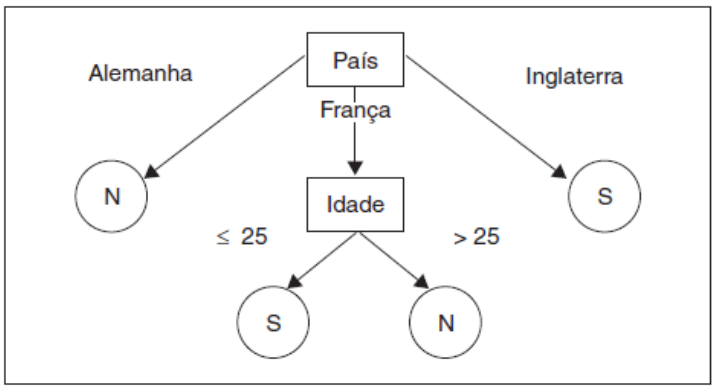
$$\text{Abrangência} = \frac{|X \wedge Y|}{|Y|}. \quad (7)$$

Para exclusão das regras é estabelecido um valor mínimo para a Precisão e Abrangência das regras e as que não satisfaçam este critério serão removidas do Modelo de Conhecimento.

2.4.3 Transformação, Organização e Apresentação dos Resultados

Durante a etapa de pós-processamento podem ser utilizados métodos de transformação dos modelos de conhecimento, por exemplo transformando um modelo de regras de associação para árvore de decisão. A Figura 8 apresenta um exemplo de modelo de conhecimento em árvore de decisão e o mesmo modelo em regras. Outros métodos de transformação envolvem por exemplo a criação de cubos para representação dos dados (GOLDSCHMIDT; PASSOS, 2005; HAN *et al.*, 2011).

Figura 8 – Árvore de Decisão e Regras



Se País=Alemanha Então Compra=Não
Se País=Inglaterra Então Compra=Sim
Se País=França e Idade ≤ 25 Então Compra=Sim
Se País=França e Idade > 25 Então Compra=Não

Fonte: (GOLDSCHMIDT; PASSOS, 2005)

2.5 **Trabalhos correlatos**

Segundo Berri *et al.* (2011), alguns elementos tornaram-se importantes na escolha de atletas da NCAA precedentes no *draft*, entre eles pode-se citar os jogadores que marcaram mais pontos durante as partidas, número total de rebotes e bloqueios. Além disso, outros dados foram implementados com suma importância para a equação analisada do *draft* como:

- se o atleta jogou as finais da NCAA;
- se ele foi campeão;
- qual a conferência em que ele atuou na NCAA.

Concluiu-se assim, que os atletas que obtiveram um número superior de pontos nas partidas são os primeiros escolhidos no *draft*, sendo os resultados das finais da NCAA ignorados. Outro detalhe observado foi que jogadores mais defensivos normalmente são as últimas escolhas da metodologia utilizada e muitas vezes o desempenho deles na NBA não atende ao esperado das equipes.

O trabalho de Haghighat *et al.* (2013) tinha como objetivo analisar diversos *datasets*, entre eles dois *datasets* da NBA (anos 2009 e 2010), para a previsão dos resultados. Durante essa pesquisa foram aplicados os algoritmos redes neurais artificiais, SVM, método Bayesiano, árvore de decisão e regressão logística. Porém a acurácia de até $69,97\%$ nos resultados obtidos para a previsão foi considerada baixa segundo os autores.

O trabalho de Zdravevski e Kulakov (2010) teve como meta utilizar técnicas de mineração de dados na previsão do vencedor em jogos esportivos. Os dados foram coletados por meio do *website* oficial da NBA. Foi executado o pré-processamento dos dados e utilizando a ferramenta Weka (do inglês, Waikato Environment for Knowledge Analysis) foram aplicados os seguintes algoritmos de classificação: ávore de decisão, método bayesiano, SVM, Random Forest. Os resultados encontrados indicam que a previsão foi possível porém a acurácia não ultrapassou de $72,8\%$.

3 MATERIAIS E MÉTODOS

Neste capítulo são descritos os materiais e o método utilizado no desenvolvimento deste projeto. São descritas as ferramentas utilizadas e as principais etapas do projeto.

3.1 Base de Dados

A base de dados utilizada neste projeto é NCAA Basketball[1]. Ela é disponibilizada publicamente pela Kaggle[2], que é uma organização de propriedade da Google LLC. Este conjunto de dados possuí informações dos jogos, das equipes e jogadores da Divisão Masculina de Basquetebol desde 2009.

A base de dados possuí 9 tabelas que armazenam diferentes informações, a seguir é apresentada uma descrição do conteúdo das principais:

- **mbb_games_sr:** Conjunto de dados com a pontuação das equipes em todos os jogos da temporada 2013-14 até 2017-18;
- **mbb_pbp_sr:** Conjunto de dados com informações de jogada-a-jogada de todos os jogos desde 2013-14;
- **mbb_players_games_sr:** Conjunto de dados com a pontuação individual de cada atleta em todos os jogos da temporada 2013-14 até 2017-18;
- **mbb_teams:** Conjunto de dados com informações gerais sobre as equipes;
- **mbb_teams_games_sr:** Conjunto de dados com a pontuação das equipes em todos os jogos da temporada 2013-14 até 2017-18. Esses dados são idênticos aos da tabela mbb_games_sr, mas estão organizados de forma diferente para facilitar o cálculo das estatísticas de uma única equipe.

[1] https://www.kaggle.com/ncaa/ncaa-basketball
[2] https://www.kaggle.com

O foco deste projeto é a tabela mbb_players_games_sr, a qual possuí os dados de cada atleta individualmente, neste conjunto de dados tem-se campos tais como: identificação do jogo, nome do atleta, posição, número de arremessos feitos, arremessos convertidos, arremessos de três pontos, faltas cometidas, etc. A Figura 9 é uma amostra deste conjunto de dados.

Figura 9 – Amostra de dados do conjunto mbb_players_games_sr

mbb_players_games_sr

full_name [Player info] Player full name	position [Player stats] Position	field_goals_made [Player stats] Field goals made	field_goals_att [Player stats] Field goals attempted	field_goals_pct [Player stats] Field goal percentage
Jimmy Wohrer	G	6	12	50
Isiah Saenz	G	7	15	46.7
Ty'rese Searles	G	10	23	43.5
Chase Coomer	G	8	14	57.1

Fonte: autoria própria (2021)

3.2 Software

Os softwares utilizados no desenvolvimento deste projetos são todos gratuitos. Nesta sessão será descrito os softwares utilizados.

- **BigQuery API[3]:** o BigQuery é um sistema de armazenamento de dados na nuvem, sem servidor, altamente escalonável e econômico com BI Engine na memória e aprendizado de máquina integrado. Faz parte do conjunto de aplicações do pacote do Google Cloud para Análise de Big Data. O BigQuery permite a análise interativa de conjuntos de dados massivamente grandes que funcionam em conjunto com o Google Cloud. A base de dados utilizada neste projeto está disponível por meio da BigQuery API;
- **Python 3.7.3[4]:** Python é uma linguagem de programação de alto nível que permite trabalhar rapidamente e integrar sistemas de maneira mais eficaz. Foi escolhida essa linguagem por ser compatível com o BigQuery API[5];
- **Weka 3.8.5[6]:** Weka é ferramenta com uma coleção de algoritmos de aprendizado de

[3]https://cloud.google.com/bigquery/
[4]https://www.python.org/downloads/release/python-373/
[5]https://googleapis.github.io/google-cloud-python/latest/bigquery/index.html
[6]https://www.cs.waikato.ac.nz/ ml/weka/

máquina utilizada para solução de problemas de mineração de dados do mundo Real. Foi desenvolvido em Java e pode ser utilizada em diversas plataformas. Os algoritmos podem ser aplicados diretamente em um conjunto de dados;

- **PostgreSQL 13.2[7]:** PostgreSQL é um robusto banco de dados relacional, possuí código aberto em desenvolvimento a mais de 30 anos;
- **HeidiSQL 11.3[8]:** HeidiSQL é software livre que permite ver e editar dados e estruturas de banco de dados MariaDB, MySQL, Microsoft SQL, PostgreSQL e SQLite.

3.3 Método

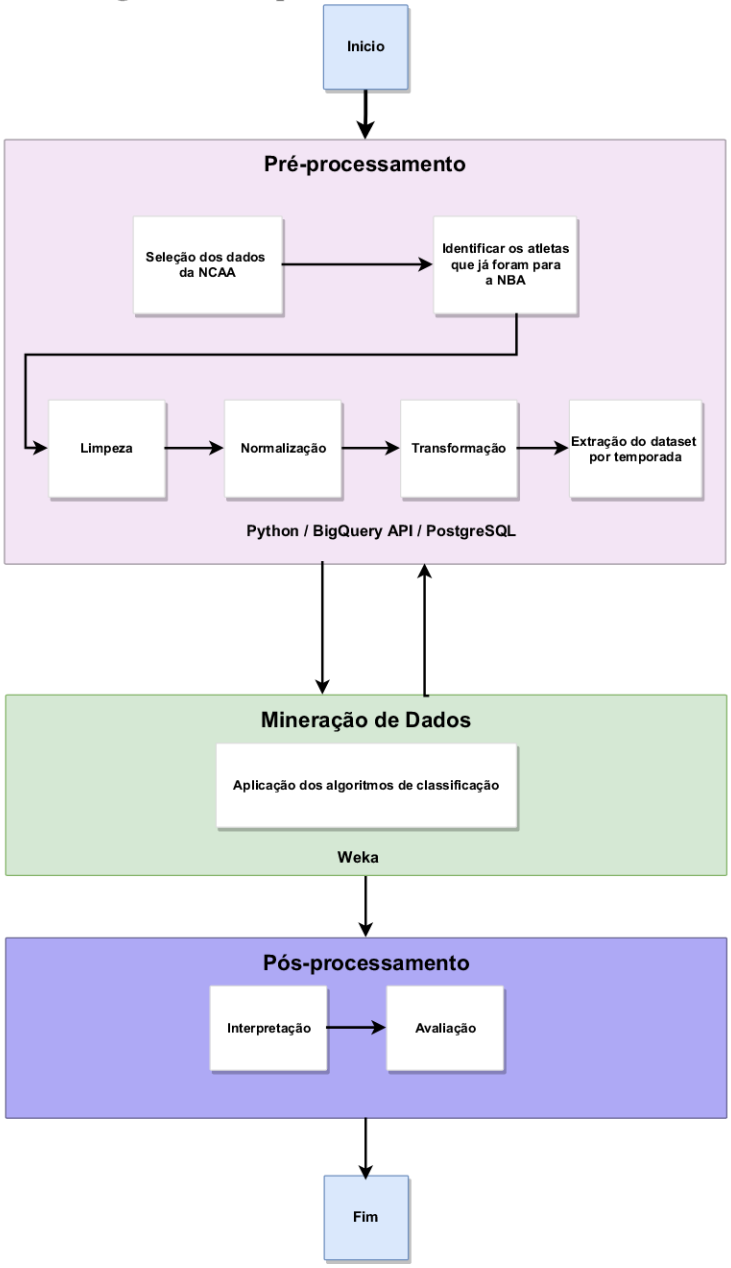
O fluxograma da Figura 10 ilustra o método utilizado neste projeto que foi dividido nas etapas:

- **Pré-processamento:** Nesta etapa foi obtido o conjunto de dados da NCAA e a relação de atletas selecionados no *draft* das temporadas 2013-14, 2014-15, 2015-16 e 2016-17[9] . Em posse dessas informações foi inserido um campo novo informando o ano do atleta selecionado no *draft*. Para tal operação foi feita uma extração dos dados do Kaggle e importado em um banco de dados PostgreSQL armazenado localmente e inserido via SQL. Após a inserção dos atletas selecionados, foi realizado uma limpeza dos dados nulos. Concluída essa etapa, foram extraídos os dados do banco de dados para um CSV, formato de arquivo compatível com Weka. Durante a extração dos dados foram separados por temporadas e removidos algumas colunas com identificação única, restando apenas as seguintes informações: *points*, *flagrant_fouls*, *tech_fouls*, *personal_fouls*, *assists_turnover_ratio*, *blocks*, *steals*, *turnovers*, *assists*, *rebounds*, *defensive_rebounds*, *offensive_rebounds*, *free_throws_pct*, *free_throws_att*, *free_throws_made*, *blocked_att*, *two_points_pct*, *two_points_att*, *two_points_made*, *three_points_pct*, *three_points_att*, *three_points_made*, *field_goals_pct*, *field_goals_att*, *field_goals_made*, *minutes*, *overall_pick*.
- **Mineração dos Dados:** Nesta etapa foram aplicados os algoritmos KNN, j48, Naive Bayes, SVM, Random Forest e MLP com o objetivo de classificar os atletas escolhidos no *draft* da NBA;

[7] https://www.postgresql.org/
[8] https://www.heidisql.com/
[9] https://www.nba.com/history/draft

- **Pós-processamento:** Após a mineração dos dados foi feita a interpretação dos padrões encontrados e avaliação da qualidade da classificação. Por fim, foi feita uma comparação entre o resultado dos algoritmos.

Figura 10 – Sequência de Métodos Adotados



Fonte: autoria própria (2021)

4 RESULTADOS E DISCUSSÃO

Neste capítulo serão apresentados os experimentos executados durante o desenvolvimento desse projeto, bem como seus resultados.

4.1 Experimentos Preliminares

Para os experimentos preliminares foi selecionado o *dataset* de 2013 contra si mesmo com e sem balanceamento. O balanceamento foi feito utilizando o filtro *resample* do Weka usando como parâmetros bias_to_uniforme_class = 1.0 e sample_size_percent = *x*, no qual:

$$total1 = \text{instâncias positivas } + \text{ instâncias negativas} \tag{8}$$

$$total2 = \text{instâncias balanceadas (positivas + negativas)} \tag{9}$$

$$x = 100 * \frac{total2}{total1} \tag{10}$$

Como o *dataset* de 2013 possuí 45 instâncias positivas e 5507 negativas, obtém-se os valores: total1 $= 45 + 5507 = 5552$, total2 $= 45 + 45 = 90$, logo x $= 1,621037463976945$.

Os classificadores selecionados para os experimentos foram: KNN, j48, *Naive Bayes*, SVM, *Random Forest* e *Multilayer Perceptron*. Obtendo os seguintes resultados:

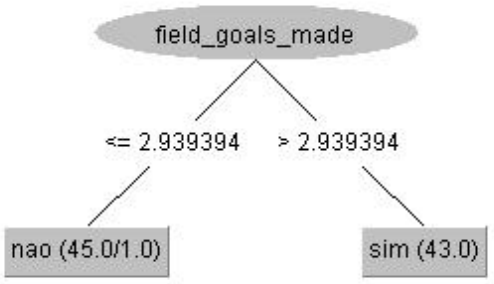
Ao observar-se os resultados da Tabela 5 é possível identificar que o balanceamento melhorou o resultado para todos os algoritmos testados. A árvore de decisão, como pode ser observado pela Figura 11, ficou bem simplificada, caso o *field_goals_made* $> 2,939394$ o jogador será da classe positiva. Como o ganhou com o balanceamento dos dados foi bastante expressivo, para os demais experimentos será executado sempre com balanceamento.

Tabela 5 – *Dataset* de 2013 dividido entre treino e teste

Classificador	Medida F pré-balanceamento	Medida F pós balanceamento
KNN	32,1 %	**97,7 %**
j48	45,2 %	**97,7 %**
Naive Bayes	14,1 %	**94,5 %**
SVM	0,0 %	**96,6 %**
Random Forest	26,4 %	**96,6 %**
MLP	33,3 %	**95,2 %**

Fonte: autoria própria (2021)

Figura 11 – Dataset 2013: Árvore de Decisão - Balanceada



Fonte: autoria própria (2021)

4.2 Experimento com Seleção de Atributos

Para seleção dos atributos foi executado o algoritmo *Classifier Attribute Eval* e selecionando os 5 primeiros atributos: *minutes*, *steals*, *assists*, *rebounds*, *defensive_rebounds*. Com a seleção de atributos os resultados foram levemente inferiores para quase todos os algoritmos, sendo o único a ter um ganho foi o *Multilayer Perceptron*, como pode ser observado na Tabela 6. Inclusive a árvore gerada acabou sendo maior como pode ser observado na Figura 12.

Apesar do resultado ser um pouco inferior com a seleção de atributos a redução no tempo de execução dos algoritmos foi bastante considerável. No caso do MLP, o algoritmo que levava $16,59$ segundos para criar o modelo com o *dataset* pré-balanceamento reduziu para $0,26$ segundos pós balanceamento e passou a ser executado em apenas $0,1$ segundos pós balanceamento e com a seleção de atributos.

Tabela 6 – *Dataset* de 2013 dividido entre treino e teste com seleção de atributos.

Classificador	Seleção de Atributos?	Medida F	Acurácia
KNN	**não**	**97,7 %**	**97,72 %**
j48	**não**	**97,7 %**	**97,72 %**
Naive Bayes	**não**	**94,5 %**	**94,31 %**
SVM	**não**	**96,6 %**	**96,59 %**
Random Forest	**não**	**96,6 %**	**96,59 %**
MLP	não	95,2 %	95,45 %
KNN	sim	97,7 %	96,59 %
j48	sim	93,2 %	93,18 %
Naive Bayes	sim	94,5 %	94,31 %
SVM	sim	95,6 %	95,45 %
Random Forest	sim	95,5 %	95,45 %
MLP	**sim**	**97,7 %**	**97,72 %**

Fonte: autoria própria (2021)

4.3 **Experimentos Finais**

Após a obtenção dos resultados dos experimentos preliminares foram executados os mesmos experimentos para os seguintes *dataset's* de 2013 contra 2014, 2014 contra 2015 e assim sucessivamente. Os resultados são apresentados na Tabela 7.

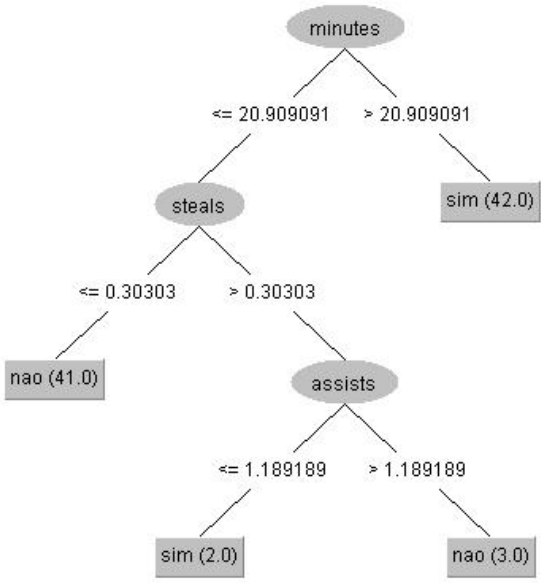
Analisando os resultados das Tabelas 6 e 7 é observável que as medidas F e a acurácia dos classificadores foi pior para as temporadas 2014, 2015, 2016 e 2017 com relação ao *dataset* 2013 (de treino), mas ainda sim tendo uma acurácia miníma superior a $71,42\%$ e medida F $72,3\%$. O algoritmo que obteve o melhor resultado no geral foi *Random Forest*, aparecendo entre os três melhores para todas as temporadas com a acurácia entre $86,25\%$ e $90,69\%$.

Tabela 7 – *Dataset* com os experimentos finais para as demais temporadas

Classificador	Seleção de Atributos?	Temporada	Medida F	Acurácia
KNN	não	2014	84,4 %	84,78 %
j48	não	2014	87,2 %	86,95 %
Naive Bayes	não	2014	**87,4 %**	**85,86 %**
SVM	não	2014	**96,6 %**	**88,04 %**
Random Forest	não	2014	**89,6 %**	**89,13 %**
MLP	sim	2014	84,8 %	83,69 %
KNN	não	2015	72,5 %	76,25 %
j48	não	2015	**83,8 %**	**85,00 %**
Naive Bayes	não	2015	75,7 %	77,50 %
SVM	não	2015	77,1 %	80,00 %
Random Forest	não	2015	**85,3 %**	**86,25 %**
MLP	sim	2015	**84,2 %**	**85,00 %**
KNN	não	2016	72,3 %	73,46 %
j48	não	2016	77,8 %	71,42 %
Naive Bayes	não	2016	**88,1 %**	**86,73 %**
SVM	não	2016	79,0 %	73,46 %
Random Forest	não	2016	**88,1 %**	**86,73 %**
MLP	sim	2016	**83,6 %**	**81,63 %**
KNN	não	2017	83,9 %	82,55 %
j48	não	2017	**88,2 %**	**87,20 %**
Naive Bayes	não	2017	86,0 %	84,88 %
SVM	não	2017	**88,9 %**	**88,37 %**
Random Forest	não	2017	**90,9 %**	**90,69 %**
MLP	sim	2017	78,6 %	79,06 %

Fonte: autoria própria (2021)

Figura 12 – Dataset 2013: Árvore de Decisão - Balanceada com Seleção de Atributos



Fonte: autoria própria (2021)

5 CONCLUSÕES

Neste trabalho foi mostrada a viabilidade do uso de técnicas de mineração de dados para a classificação dos atletas que podem ser escolhidos no *draft*. Como todos os anos diversos atletas novos entram na liga universitária o número de classes negativas será muito maior que o número de classes positivas, por isso é necessário o balanceamento do *dataset*, na qual na temporada de 2013 a qual foi dividida entre treino e teste a medida F pré-balanceamento obteve uma melhora de 0% para $96,6\%$ no caso do algoritmo SVM e para o *Random Forest* de $26,4\%$ para $96,6\%$. Entretanto, durante a seleção dos atributos não houve uma melhora no desempenho dos algoritmos.

Os objetivos do trabalho foram todos alcançados, desde o pré-processamento do *dataset* até a seleção dos melhores atributos para a utilização dos algorítimos de classificação se mostrou viável para a predição dos atletas selecionados no *draft* da NBA. Os resultados obtidos foram satisfatórios, ficando com acurácias entre $71,42\%$ no pior caso e $90,69\%$ no melhor caso.

Não foram encontrados trabalhos pra a predição da escolha do *draft* usando técnicas de mineração de dados para uma comparação mais detalhada. Com relação aos resultados obtidos por Haghighat *et al.* (2013) e Zdravevski e Kulakov (2010) na predição do resultado dos jogos da NBA tiveram um acurácia inferior a encontrada neste trabalho. Isso pode ser pelo fato das tarefas serem diferentes ou ainda por causa da qualidade dos dados do *dataset*.

5.1 Trabalhos Futuros

A mineração de dados é uma área em constante evolução, este trabalho foi apenas um pequeno passo de diversos outros experimentos que podem ser executados, a utilização de outras técnicas de mineração de dados, tais como regras de associação, ou mesmo a escolha de outros algoritmos para a seleção de atributos. Outra possível abordagem seria a divisão do *dataset* de treino em mais temporadas, aumentando assim o número de classes positivas afim

de gerar um resultado melhor.

REFERÊNCIAS

AGGARWAL, C. C. **Data Classification: Algorithms and Applications**. 1st. ed. [S.l.]: Chapman amp; Hall/CRC, 2014. ISBN 1466586745.

AGGARWAL, C. C. **Data Mining: The Textbook**. [S.l.]: Springer Publishing Company, Incorporated, 2015. ISBN 3319141414.

BERRI, D. J.; BROOK, S. L.; FENN, A. J. From college to the pros: Predicting the NBA amateur player draft. **Journal of Productivity Analysis**, v. 35, n. 1, p. 25–35, 2011. ISSN 0895562X.

BRACHMAN, R. J.; ANAND, T. The process of knowledge discovery in databases. American Association for Artificial Intelligence, USA, p. 37–57, 1996.

CARVALHO, L. de. **Datamining: a mineração de dados no marketing, medicina, economia, engenharia e administração**. [S.l.]: Érica, 2001. ISBN 9788571947665.

COPPIN, B. **Artificial Intelligence Illuminated**. USA: Jones and Bartlett Publishers, Inc., 2004. ISBN 0763732303.

FAYYAD, U. The KDD Process for Extracting Useful Knowledge from Volumes of Data. **Communications of the ACM**, v. 39, n. 11, p. 27–34, 1996.

FAYYAD, U.; PIATETSKY-SHAPIRO, G.; SMYTH, P. **From Data Mining to Knowledge Discovery in Databases) (© AAAI)**. [S.l.], 1996. v. 17, n. 3, 37–54 p.

FORT, R. **Sports Economics**. [S.l.]: Prentice Hall, 2011. ISBN 9780136066026.

GOLDSCHMIDT, R.; PASSOS, E. **Data mining: um guia Prático**. [S.l.: s.n.], 2005. ISBN 9788535218770.

GOODFELLOW, I.; BENGIO, Y.; COURVILLE, A. **Deep Learning**. [S.l.]: MIT Press, 2016. (Adaptive computation and machine learning). ISBN 9780262035613.

HAGHIGHAT, M.; RASTEGARI, H.; NOURAFZA, N. A Review of Data Mining Techniques for Result Prediction in Sports. **Advances in Computer Science: an International Journal**, v. 2, n. 6, p. 7–12, 2013. ISSN 2322-5157.

HAN, J.; KAMBER, M.; PEI, J. **Data Mining: Concepts and Techniques**. 3. ed. [S.l.]: Morgan Kaufmann, 2011. (The Morgan Kaufmann Series in Data Management Systems). ISBN 9789380931913.

HAND, D. J. Principles of data mining. **Drug Safety**, v. 30, n. 7, p. 621–622, 2007. ISSN 01145916.

HAYKIN, S. S. **Neural networks and learning machines**. Third. Upper Saddle River, NJ: Pearson Education, 2009.

INMON, W. H. Data Mining : an Architecture Part 1. **Tech Topic**, Pine Cone System Inc, v. 5, p. 1–30, 1997.

ISSENBERG, S. **A More Perfect Union**. [S.l.]: MIT Tecnology Review, 2013.

LAROSE, D. T.; LAROSE, C. D. **Discovering knowledge in data: an introduction to data mining**. 2. ed. [S.l.]: John Wiley & Sons, 2014. ISBN 9780470908747.

MITCHELL, T. M. **Machine Learning**. New York: McGraw-Hill, 1997. ISBN 978-0-07-042807-2.

NAISMITH, J. **Basketball: Its Origin and Development**. [S.l.]: Bison Books, 1996.

RUSSELL, S.; NORVIG, P. **Artificial Intelligence: A Modern Approach**. 3. ed. [S.l.]: Prentice Hall, 2010.

WITTEN, I. H.; FRANK, E.; HALL, M. A. **Data Mining: Practical Machine Learning Tools and Techniques**. 3. ed. Amsterdam: Morgan Kaufmann, 2011. (Morgan Kaufmann Series in Data Management Systems). ISBN 978-0-12-374856-0.

ZDRAVEVSKI, E.; KULAKOV, A. System for prediction of the winner in a sports game. p. 55–63, 01 2010.

ANEXO A – LISTAGEM DE CAMPOS DO *DATASET* MBB_PLAYERS_GAMES_SR.

Tabela 8 – Estrutura do Conjunto de Dados mbb_players_games_sr

Attribute	Description
game_id	[Game data] Unique identifier for the game
season	[Game data] Season the game was played in
neutral_site	[Game data] Indicator of whether the game was played on a neutral court
scheduled	[Game data] Date the game was played
gametime	[Game data] Date and time the game was played
tournament	[Game data] Whether the game was played in a post-season tournament
tournament_type	[Game data] Type of post-season tournament a game was in played
tournament_round	[Game data] Tournament round
tournament_game_no	[Game data] Tournament game number
player_id	[Player info] Player Sportradar player ID
last_name	[Player info] Player last name
first_name	[Player info] Player first name
full_name	[Player info] Player full name
abbr_name	[Player info] Player abbreviated name (”F.Last”)
status	[Player info] Player status as of 2017-18 season
jersey_number	[Player info] Player jersey number
height	[Player info] Player height
weight	[Player info] Player weight
birth_place	[Player info] Player birth place or home
birthplace_city	[Player info] Player’s home city
birthplace_state	[Player info] Player’s home state
birthplace_country	[Player info] Player’s home country
class	[Player info] Player’s class at game time
team_name	[Team info] Team name
team_market	[Team info] Team school name (using Sportradar names)
team_id	[Team info] Sportradar team ID
team_alias	[Team info] Team alias
conf_name	[Team info] Team current conference name
conf_alias	[Team info] Team alias
team_alias	[Team info] Team alias
division_name	[Team info] Team current division name
division_alias	[Team info] Team current division alias

Fonte: Adaptado da NCAA Basketball

Attribute	Description
league_name	[Team info] Team current league name
home_team	[Team info] Indicator of whether the team was the home team
active	[Player stats] Indicator of whether the player was active for the game
played	[Player stats] Indicator of whether the player played in the game
starter	[Player stats] Indicator of whether the player started the game
minutes	[Player stats] Minutes played
minutes_int64	[Player stats] Minutes played (as integer)
position	[Player stats] Position
primary_position	[Player stats] Primary position
field_goals_made	[Player stats] Field goals made
field_goals_att	[Player stats] Field goals attempted
field_goals_pct	[Player stats] Field goal percentage
three_points_made	[Player stats] Three-pointers made
three_points_att	[Player stats] Three-pointers attempted
three_points_pct	[Player stats] Three-point shot percentage
two_points_made	[Player stats] Two-pointers made
two_points_att	[Player stats] Two-pointers attempted
two_points_pct	[Player stats] Two-point shot percentage
blocked_att	[Player stats] Number of shots blocked by the other team
free_throws_made	[Player stats] Free throws made
free_throws_att	[Player stats] Free throws attempted
free_throws_pct	[Player stats] Free throw percentage
offensive_rebounds	[Player stats] Offensive rebounds
defensive_rebounds	[Player stats] Defensive rebounds
rebounds	[Player stats] Total rebounds
assists	[Player stats] Assists
turnovers	[Player stats] Turnovers
steals	[Player stats] Steals
blocks	[Player stats] Blocks
assists_turnover_ratio	[Player stats] Assist-to-turnover ratio
personal_fouls	[Player stats] Personal fouls committed
tech_fouls	[Player stats] Technical fouls committed
flagrant_fouls	[Player stats] Flagrant fouls committed
points	[Player stats] Points scored
sp_created	[Table data] Box score data entry time

Fonte: Adaptado da NCAA Basketball