UNIVERSIDADE DO VALE DO TAQUARI
CURSO DE SISTEMAS DE INFORMAÇÃO

APLICAÇÃO DE TÉCNICAS DE MINERAÇÃO DE DADOS PARA PREVISÃO DE JOGOS DE BASQUETE

João Pedro Bogoni

Lajeado, junho de 2019

João Pedro Bogoni

APLICAÇÃO DE TÉCNICAS DE MINERAÇÃO DE DADOS PARA PREVISÃO DE JOGOS DE BASQUETE

Monografia apresentada ao Centro de Ciências Exatas e Tecnológicas da Universidade do Vale do Taquari – UNIVATES, como parte da exigência para a obtenção do título de bacharel em Sistemas de Informação.

Orientador: Prof. Ms. Alexandre Stürmer Wolf

Lajeado, junho de 2019

AGRADECIMENTOS

Agradeço aos meus pais, Jairo e Joana, por estarem ao meu lado nesta caminhada e por sua dedicação ao longo dos anos, proporcionando a oportunidade de crescer a cada dia. A minha namorada, Natália, pelo companheirismo e compreensão.

Também agradeço os colegas de trabalho da OM pelo apoio e suporte durante esta trajetória.

Também sou grato aos professores que contribuíram para o meu crescimento, principalmente ao Sr. Alexandre Stürmer Wolf, pelo privilégio de sua orientação.

RESUMO

A Mineração de dados vem atraindo um grande interesse na descoberta de informações em abrangentes áreas de atuação. Na área esportiva um dos grandes interesses é a capacidade de prever resultados de jogos. O basquete em especial, oferece um conjunto de atributos estatísticos a cada jogo, que podem ser explorados para descobrir tendências de performance. Este trabalho concentra-se na aplicação de técnicas de mineração de dados e aprendizado de máquina para prever o resultado de jogos da National Basketball Association (NBA). Para isso, são utilizados dados estatísticos de cinco temporadas de jogos da NBA. Os resultados obtidos por diferentes técnicas de aprendizado de máquina, são comparados para encontrar a forma mais eficiente para prever os resultados dos jogos. Após o treinamento dos modelos e aplicação dos mesmos sobre o conjunto de teste, foi constatado que os algoritmos Multi-Layer Perceptron (MLP) e Logistic Regression obtiveram a melhor acuracidade, atingindo um percentual de 68.04% e 67.94% respectivamente. Os resultados também são comparados com aqueles obtidos de outros trabalhos do mesmo campo de pesquisa, verificando assim que o desempenho dos modelos de previsão foi muito próximo.

Palavras-chave: Mineração de Dados. Aprendizado de Máquina. Basquete. Previsão.

ABSTRACT

Data Mining has been attracting a great deal of interest in finding information in broad areas of expertise. In the sports field one of the big interests is the ability to predict game results. Basketball in particular offers a set of statistical attributes to each game, which can be exploited to uncover performance trends. This work focuses on the application of data mining and machine learning techniques to predict the outcome of National Basketball Association (NBA) games. For that, statistical data from five seasons of NBA games are used. The results obtained by different machine learning techniques are compared to find the most efficient way to predict the results of the games. After the training of the models and their application on the test set, it was verified that the Multi-Layer Perceptron (MLP) and Logistic Regression algorithms obtained the best accuracy, reaching a percentage of 68.04% and 67.94%, respectively. The results are also compared with those obtained from other works of the same research field, thus verifying that the performance of the forecast models was very close.

Keywords: Data Mining. Machine Learning. Basketball. Prediction.

LISTA DE FIGURAS

LISTA DE GRÁFICOS

LISTA DE QUADROS

LISTA DE TABELAS

LISTA DE ABREVIATURAS E SIGLAS

API	Application Programming Interface – Interface de Programação de Aplicações
CSV	Comma Separated Values – Valores Separados por Vírgula
kNN	k Nearest Neighbors – k Vizinhos Próximos
KDD	Knowledge Discovery in Databases – Descoberta de conhecimentos
MLB	Major League Baseball
MLP	Multi-Layer Perceptron
NBA	National Basketball Association
RNA	Rede Neural Artificial
SGBD	Sistema de Gerenciamento de Banco de Dados
SQL	Structured Query Language – Linguagem de Consulta Estruturada
SVM	Support Vector Machine - Máquinas de Vetores de Suporte
WEKA	Waikato Environment for Knowledge Analysis – Ambiente para Análise de Conhecimento Waikato
XLSX	Extensible Markup Language for Excel – Marcação Extensível de Linguagem para Excel

SUMÁRIO

1 INTRODUÇÃO

Com o avanço das tecnologias nas últimas décadas o volume de dados armazenados por diversos setores socioeconômicos vem crescendo de forma significativa, estima-se que até 2020 a quantidade de informação armazenada cresça 50 vezes. Neste quadro surgiu a Mineração de Dados, com o objetivo de preparar e extrair conhecimento de grandes bases de dados (CASTRO; FERRARI, 2016).

Um dos campos de pesquisa e aplicações na área de mineração de dados é o de aprendizado de máquina, tendo como objetivo criar sistemas de computadores capazes de adquirir conhecimento de forma automática, sendo capazes de reconhecer complexos padrões de dados e tomar decisões baseadas nessas informações (HAN; KAMBER; PEI, 2012).

Uma área em que a aplicação de mineração de dados vem sendo inserida é o setor esportivo. Um dos casos mais famosos desta prática é a história representada no filme *Moneyball,* no qual o diretor esportivo Billy Bean, junto ao seu assistente Peter Brand, formaram uma equipe de beisebol muito competitiva e barata. Avaliando os jogadores através do percentual de vezes que os mesmos chegavam em base, eles foram capazes de formar times competitivos e baratos. A partir disso os demais times da *Major League Baseball* (MLB), adotaram os métodos de Billy Bean e Peter Brand para avaliar jogadores e montarem seus respectivos elencos (DATAGEEK, 2018).

A *National Basketball Association* (NBA), liga nacional de basquete dos Estados Unidos, é composta por 30 equipes divididas em duas conferências, leste e oeste. Cada equipe joga no mínimo 82 jogos por temporada, uma equipe para tornar-se campeã necessita jogar no mínimo 98 jogos contado os jogos de pós-temporada (SPORTS REFERENCE LLC, 2018).

A NBA registra um enorme número de dados estatísticos após cada partida, estes dados contém um alto nível de detalhamento, contendo informações estatísticas em diversos aspectos do jogo, como, por exemplo, eficiência ofensiva e defensiva, rebotes, assistências, dias de descanso que a equipe chega para a partida e performance em diferentes tipos de arremesso. No passado as equipes dependiam de conhecimento humano de técnicos e gestores da equipe para transformar em conhecimento, na atualidade a utilização de técnicas de mineração de dados e aprendizado de máquina possibilita que reconhecimento de padrões de desempenho de equipes e atletas individualmente seja possível de forma automatizada, tornando-se uma vantagem de competitividade para os times nos fins estratégicos de jogo e montagem de elencos e de conhecimento para apostadores.

Sendo assim, este trabalho propõe a comparação de diversos algoritmos de aprendizado de máquina, avaliando o desempenho dos mesmos na previsão dos jogos da NBA, com base em dados estatísticos das temporadas de 2013 em diante. Também será proposto a criação de um modelo de previsão para jogos futuros com o algoritmo que obter o melhor desempenho.

1.1 Objetivos

Nesta seção são apresentados o objetivo geral, objetivos específicos propostos neste trabalho.

1.1.1 Objetivo Geral

O principal objetivo deste trabalho é avaliar o desempenho de diferentes algoritmos de aprendizado de máquina ao classificar o resultado de jogos da NBA, aplicando técnicas de aprendizado de máquina sobre dados estatísticos de jogos coletados nas últimas cinco temporadas.

1.1.2 Objetivos específicos

Nesta seção estão presentes os principais objetivos específicos que o trabalho busca alcançar:

- Realizar um estudo identificando quais as ferramentas e recursos usados para a aplicação de *machine learning*;
- Identificar os atributos da base de dados que possuem maior influência no resultado final da partida;
- Criar um modelo de previsão capaz de prever jogos futuros com base no algoritmo que apresentar melhor desempenho;
- Analisar e comparar resultados com outras pesquisas que utilizaram a mesma metodologia aplicada.

1.2 Justificativa

A mineração de dados é um campo de estudo que está em amplo crescimento, é cada vez mais valioso poder extrair conhecimento de forma automatizada e eficaz,

principalmente quando este conhecimento não é possível adquiri-lo nos meios tradicionais de análise de dados.

Estabelecer quais os algoritmos são mais eficientes para a resolução do problema é uma tarefa importante, na qual é requerido que os dados passem por um processo de preparação para resolver problemas existentes como valores faltantes ou valores inconsistentes.

Construir um modelo de previsão que seja capaz de atingir um alto nível de acuracidade, para o mesmo ser utilizada para fins de apostas esportivas. Sendo este estudo capaz de reconhecer padrões de performance dos times, qual o comportamento destes times frente a determinados adversários e como apostadores devem considerar isso para suas apostas. Além dos fatos listados acimas a NBA em 2018 tornou-se a primeira liga de esporte profissional dos Estados Unidos a firmar uma parceria com duas companhias de apostas esportivas, ambas tornaram-se distribuidoras oficiais dos dados fornecidos pela NBA para este fim (NBA, 2019a).

1.3 Estrutura do trabalho

O presente trabalho está divido em capítulos. Após o capítulo de introdução situa-se o capítulo que detalha a sustentação teórica aplicada para alcançar os objetivos propostos neste trabalho, com princípios relacionados a mineração de dados e aprendizado de máquina, também estão presentes as ferramentas e bibliotecas que são utilizadas e os trabalhos relacionados.

O capítulo três relaciona as metodologias utilizadas para o desenvolvimento deste trabalho, já o capítulo quatro apresenta o desenvolvimento do projeto, no qual os resultados obtidos são apresentados. Por fim, no quinto capítulo são expostas as considerações finais do projeto.

2 REFERENCIAL TEÓRICO

Este capítulo apresenta os conceitos teóricos fundamentais para o entendimento do trabalho. São apontados os conceitos gerais de mineração de dados, técnicas de aprendizado de máquina, algoritmos aplicados para a criação de modelo de previsão e ferramentas e bibliotecas de desenvolvimento. Também são listadas as estatísticas presentes nos conjuntos de dados e os trabalhos relacionados.

2.1 Mineração de Dados

O progressivo avanço tecnológico da sociedade, vem resultando na geração de um grande volume de dados, podendo considerar o atual momento da sociedade como a idade dos dados. Em um primeiro momento surgiu a necessidade de armazenar, organizar e controlar esse grande volume de dados, a evolução tecnológica natural desse processo e à necessidade de técnicas e ferramentas capazes de extrair conhecimento com relação a estes dados, a partir desta necessidade deu-se a origem da área de mineração de dados (HAN; KAMBER; PEI, 2012).

Segundo Castro e Ferrari (2016), a mineração de dados faz parte da composição de um processo mais vasto, denominado como *knowledge discovery in databases* (KDD). O KDD pode ser dividido em quatro etapas (CASTRO; FERRARI, 2016):

- Base de dados: conjunto de dados, o nível mais básico dos quais informações e conhecimentos podem ser extraídos;

- Preparação dos dados: etapa anterior a mineração dos dados, está etapa engloba a limpeza, integração, seleção e transformação dos dados com o objetivo de proporcionar uma análise eficiente e eficaz dos dados;

- Mineração dos dados: etapa do processo que diz respeito a aplicação de algoritmos capazes de extrair conhecimentos dos dados pré-processados;

- Validação: etapa de avaliação dos resultados da mineração, procurando identificar se o conhecimento extraído é útil.

A área de mineração de dados tem como característica ser multidisciplinar, envolvendo conhecimento em diversos campos entre eles inteligência artificial e aprendizado de máquina. A Figura 1 apresenta alguns destes campos (CASTRO; FERRARI, 2016).

Figura 1 - Multidisciplinaridade da mineração de dados

Fonte: Castro e Ferrari (2016, pág. 7).

Tan, Steinbach e Kumar (2009), afirmam que as tarefas da mineração de dados em geral são separadas em duas categorias:

- Tarefas de Previsão: tem como objetivo prever o valor de um atributo baseado nos demais atributos, o atributo a ser previsto é conhecido como o atributo alvo, já os demais atributos são conhecidos como as variáveis explicativas;

- Tarefas Descritivas: tem como objetivo provir padrões de correlações, agrupamentos e tendências, as tarefas descritivas são frequentemente utilizadas de forma exploratória, necessitando técnicas de pós-processamento para a validação dos dados.

2.2 Dados

Conjunto de dados divergem em um diferente número de maneiras, sendo os atributos presentes de diferentes tipos e alguns conjuntos de dados possuem relações diretas entre eles. Os tipos de dados, é determinante para definir quais técnicas e ferramentas devem ser utilizadas para a exploração e análise dos dados (TAN; STEINBACH; KUMAR, 2009).

Tan, Steinbach e Kumar (2009), afirmam que os dados podem ser definidos em duas categorias, quantitativos ou qualitativos, o Quadro 1 apresenta os tipos de dados e as operações que podem ser realizadas.

Quadro 1 - Tipos de Dados

Tipo de Atributo		Descrição	Exemplos	Operações
Qualitativos	Nominal	Os valores de um atributo nominal são apenas nomes diferentes; i.e., valores nominais fornecem apenas informação suficiente para distinguir um objeto do outro.	Códigos postais, números de ID de funcionário, cor dos olhos, sexo.	Modo, entropia, correlação de contingência, teste x^2.
	Ordinal	Os valores de um atributo ordinal fornecem informação suficiente para ordenar objetos.	Dureza de minerais, notas, números de rua.	Medianas, porcentagens, testes de execução, testes de assinatura.
Quantitativo	Intervalar	Para atributos intervalares, as diferenças entre os valores são significativas, i.e., existe uma unidade de medida.	Datas de calendário, temperatura em Celsius ou Fahrenheit.	Média, desvio padrão, correlação de Pearson, testes T e F.
	Proporcional	Para variáveis proporcionais, tanto as diferenças quanto as proporções são significativas.	Temperatura em Kelvin, quantidades monetárias, contadores, idades, corrente elétrica.	Média geométrica, média harmônica, variação porcentual.

Fonte: do autor, adaptado de Tan, Steinbach e Kumar (2009, p.31)

Han, Kamber e Pei (2012), salientam a importância de ter conhecimento sobre seus dados, conhecer os tipos de atributos presentes e quais os valores existentes são algumas das perguntas que o cientista de dados deve responder. A partir disso conhecimentos estatísticos como mediana, média e moda são capazes de auxiliar em problemas de inconsistência, valores faltando e ruídos.

Dados coletados de uma base de dados podem conter problemas em consequência de erros humanos ou defeitos nos sistemas responsáveis pelas coletas dos mesmos. A mineração de dados salienta a necessidade de constatar e corrigir esses problemas, a seguir são relacionados os principais problemas que podem ser encontrados em um conjunto de dados (TAN; STEINBACH; KUMAR, 2009):

- Ruídos: envolve a distorção de um valor ou a adição de um objeto ilegítimo, é considerado um componente aleatório de um erro de medição e frequentemente está associado a dados com elementos temporal ou espacial, tendo como resultado a inconsistência dos dados;

- Valores faltando: representam a falta de determinado atributo, por falta de coleta, por não ser obrigatório ou não ser aplicável a determinado objeto. Como estratégia para esse problema o objeto que contém um atributo faltando pode ser eliminado ou ignorado;

- Valores inconsistentes: um valor é considerado inconsistente quando um atributo é incompatível com o restante do objeto ou existir um desvio de padrão. Casos como informação de endereço onde a cidade informada difere do código postal é um exemplo de inconsistência, a correção do problema normalmente necessita de informações adicionais ou redundantes.

2.3 Pré-processamento dos dados

Castro e Ferrari (2016), afirmam que a etapa de pré-processamento é responsável em identificar problemas e prepará-los para que os resultados da etapa de aplicação de técnicas de mineração de dados não fiquem comprometidos devido problemas nos dados, nessa essência pode ser considerado o princípio de GIGO[1]. Deste modo o objetivo da etapa é preparar os dados, sendo capaz de identificar os tipos de atributos presentes na base, a existência de dados ausentes, inconsistentes ou ruidosos e também a existência de atributos irrelevantes.

A etapa de pré-processamento dos dados é essencial para a obtenção de melhores resultados e tempos de processamento na mineração dos dados (HAN; KAMBER; PEI, 2012). O pré-processamento de dados é um extenso campo que consiste de diversas

1 GIGO é a abreviação para a expressão *garbage in, garbage out* que significa, lixo entra, lixo sai. Segue o princípio de que a qualidade de saída de um sistema depende de sua qualidade de entrada.

técnicas e estratégias, nas seções seguintes estão relacionadas as principais delas (TAN; STEINBACH; KUMAR, 2009).

2.3.1 Limpeza

O processo de limpeza consiste em solucionar problemas que podem vir a ser encontrados nos dados, conforme citado na Seção 2.3 deste capítulo. Segundo Han, Kamber e Pei (2012), a etapa de limpeza pode intervir para resolver os seguintes problemas:

- Valores ausentes: nesses casos podem ser aplicadas algumas técnicas para o preenchimento do valor ausente, isso incluí a inserção de um valor constante para todos os atributos faltantes, técnicas estatísticas também podem ser utilizadas como média e moda de acordo com tipo de valor, outra opção é a dedução do valor através da aplicação de técnicas de aprendizado de máquina como árvores de decisão ou modelos de regressão;

- Dados ruidosos: um problema de difícil identificação pois pode representar um erro aleatório ou um erro de variação em uma variável de medição, como solução podem ser utilizadas técnicas como a de aproximação nos quais funções de aproximação vão substituir os valores reais, aplicação de média ou moda como valor para substituição, algoritmos de agrupamento e regressão também são opções para a substituição dos valores.

2.3.2 Integração

Conforme Castro e Ferrari (2016), o processo de integração de dados consiste em conciliar dados de fontes diferentes em uma única base. Nessa etapa podem ocorrer três problemas que devem ser destacados:

- Redundância: em mineração de dados redundância significa que um atributo ou objeto é capaz de ser alcançado de um ou mais atributos ou objetos, como exemplo podemos citar um atributo de idade e outro de data de nascimento, nesses casos a base de dados pode não necessitar de ambos atributos. Para a descoberta destas situações é necessário realizar uma análise de correlação;

- Duplicidade: situação nas quais atributos ou objetos estão repetidos na base de dados, isso é capaz de ocasionar anomalias e distorções nos dados. A prevenção pode ser realizada através da normalização da base de dados;

- Conflitos: cenário onde a mesma entidade está diferente nas diferentes fontes de dados, o problema é comum em atributos que possuem unidade de medida, sendo assim as diferentes bases apresentam diferentes unidades de medida e por consequência valores.

2.3.3 Redução

O processo de redução tem o objetivo de reduzir o tamanho da base de dados em volume e manter a integridade original dos dados, com isso o processo de mineração torna-se mais eficiente e vai produzir um resultado igual ou muito semelhante caso a mineração fosse aplicada na base de dados original (HAN; KAMBER; PEI, 2012).

Segundo Castro e Ferrari (2016), a redução da base deve ser realizada através da aplicação de técnicas e conceitos, com destaque para a compressão de dados que consiste em diminuir a dimensionalidade dos dados, com isso deve ser capaz de atingir-se os dados originais com ou sem perdas de informações. A aplicação do método de análise de componentes principais visa a compressão da base, é um procedimento nos quais os atributos correlacionados são convertidos em um objeto de atributos linearmente descorrelacionados, através de uma projeção dos dados em um espaço de menor dimensão assim maximizando as variações dos dados, onde o número de atributos é

menor ou igual ao original sendo que o primeiro apresenta a maior variabilidade e assim continuadamente.

A seleção de subconjunto de característica também é uma forma de reduzir a dimensionalidade dos dados, descartar atributos irrelevantes ou redundantes muitas vezes necessita um tratamento sistemático, isso pode ser alcançado através de abordagens internas onde o algoritmo de mineração de dados vai naturalmente escolher quais atributos utilizar e ignorar, outro método que pode ser utilizada é a de pesagem de características nas quais as características mais importantes recebem um peso maior, isso pode ser feito com base no conhecimento de domínio relativo as características ou de forma automática, através da aplicação de modelos de classificação como máquinas de vetor de suporte (TAN; STEINBACH; KUMAR, 2009).

2.3.4 Transformação

A etapa de transformação tem o objetivo de estabelecer ou alterar os dados de forma adequada para o processo de mineração, esse processo abrange a padronização dos dados nos quais são resolvidos problemas de formatos, conversão de unidades, remoção de caracteres especiais e capitalização (CASTRO; FERRARI, 2016).

Segundo Han, Kamber e Pei (2012), a normalização dos dados, tem como objetivo dispor os dados dentro de intervalos menores ou comuns, como exemplo escolher unidades de medidas nas quais os intervalos entre o maior e menor valor seja a menor possível, assim atribuindo um peso semelhante a todos os atributos, essa técnica é especialmente útil para algoritmos de classificação e agrupamento que utilizam redes neurais artificiais. Algumas técnicas de normalização de destaque são *Max-min*, escore-z e escalonamento decimal (HAN; KAMBER; PEI, 2012).

2.3.5 Discretização

Conforme Castro e Ferrari (2016), alguns algoritmos são incapazes de trabalharem com atributos numéricos, nestas situações é necessário a discretização. A discretização

pode ser realizada através de métodos de encaixotamento, análise de histograma ou distribuindo os valores em intervalos sendo que o valor de cada intervalo representa a mediana ou média dos valores. Para Han, Kamber e Pei (2012), um dos métodos de discretização mais eficiente é o de agrupamento, resultando em uma hierarquia de conceito em formas de nós.

2.4 Aprendizado de máquina

Segundo Mitchell (1997), o campo de pesquisa do aprendizado de máquina tem interesse em desenvolver programas de computadores capazes de aperfeiçoarem-se em determinadas funções através da experiência.

Com sua capacidade de tirar informações e conhecimento, técnicas de aprendizado de máquina estão sendo fortemente utilizadas no processo de mineração de dados, procurando extrair informações de forma automatizada baseado em um banco de dados (ARTERO, 2008).

Segundo Han, Kamber e Pei (2012), aprendizado de máquina trata como computadores podem adquirir conhecimento através de dados, isso incluí o reconhecimento de padrões e tomadas de decisões com base nos dados apresentados a máquina, sendo o aprendizado de máquina uma área de grande crescimento no cenário tecnológico atual. Na Seção 2.4.1 são apresentados os diferentes métodos de aprendizado de máquina.

2.4.1 Classificação do Aprendizado de Máquina

Segundo Artero (2008), o aprendizado de máquina pode ser classificado de diversas maneiras, entretanto o mais comum é agrupar o mesmo em duas categorias, supervisionado e não supervisionado.

2.4.1.1 Aprendizado supervisionado

Algoritmos de aprendizado supervisionado aprendem ao serem introduzidas aos dados pré-classificados, sendo capazes de alterarem pesos de acordo com os dados de saída e entrada (COPPIN, 2013).

Han, Kamber e Pei (2012), afirmam que o aprendizado supervisionado é um sinônimo para classificação, sendo a supervisão do aprendizado tendo como origem os exemplos presentes nos dados explorados, estes exemplos são os responsáveis pelo aprendizado.

2.4.1.2 Aprendizado não supervisionado

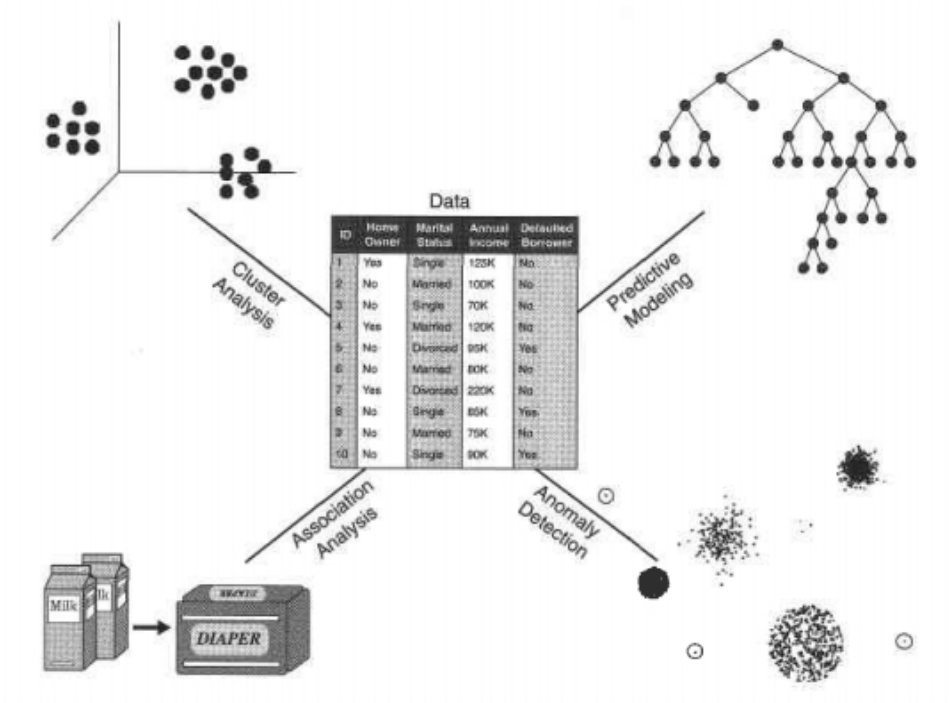
Coppin (2013), afirma que métodos de aprendizado não supervisionados, são capazes de aprender sem qualquer auxílio humano. Esses métodos são de grande valor para o agrupamento e classificação de dados previamente não conhecidos.

O aprendizado não supervisionado pode ser considerado uma técnica de agrupamento. Os dados apresentados não estão classificados e é de responsabilidade da máquina identificar padrões e classes de dados, porém como os dados não são classificados o modelo de aprendizado gerado não é capaz de indicar o significado de seus resultados (HAN; KAMBER; PEI, 2012).

2.5 Técnicas de mineração de dados

Segundo Tan, Steinbach e Kumar (2009), as técnicas de mineração de dados são relacionadas de acordo com o objetivo e tipo de informação a qual deseja-se extrair dos dados explorados. Estas tarefas são divididas em dois grupos: descritivas e preditivas. As tarefas descritivas têm como objetivo correlacionar valores e agrupar dados, já as tarefas preditivas possuem o objetivo de prever um valor de determinado atributo da base de dados. A Figura 2 apresenta tarefas da mineração de dados, que também podem ser utilizadas para associação e detecção de anomalias.

Figura 2 - As principais tarefas da mineração de dados



Fonte: Tans, Steinbach e Kumar (2009, p. 9).

Na Seção 2.5.1, são apresentadas técnicas que constituem tarefas de mineração de dados.

2.5.1 Técnicas de Classificação

Segundo Castro e Ferrari (2016), a classificação pode ser definida como um exercício de predicação baseado em registro que possuem um valor de saída, através da análise histórica destes dados é possível a criação de modelos capazes de prever o valor de saída de determinado atributo.

Sendo assim, diversos métodos de classificação foram propostos por pesquisadores nas áreas de aprendizado de máquina, estatística e reconhecimento de padrões. Ultimamente o desenvolvimento na área de mineração de dados, fez com que seja possível a criação de modelos de classificação em cima de grandes conjuntos de dados. O processo de classificação é divido em duas etapas, primeiramente a etapa de aprendizado ou treinamento, no qual é realizada a construção do modelo, e posteriormente a etapa de classificação ou teste no qual é realizada a avaliação do modelo criado (HAN; KAMBER e PEI, 2012).

2.5.1.1 Árvore de decisão

Han, Kamber e Pei (2012), afirmam que árvores de decisão funcionam como um fluxograma, sendo composto por nós internos, arestas e nós folhas. No momento em que uma classe necessita ser classificada, o caminho da árvore é percorrido. Árvore de decisão é um popular algoritmo de classificação devido sua simplicidade de utilização, não dependendo de parâmetros adicionais ou de conhecimento sobre os dados explorados, porém a sua eficiência depende dos dados utilizados na exploração.

O processo de classificação em uma árvore de decisão, acontece de maneira recursiva, conforme Figura 3, de modo que o nó inicial representa o conjunta de dados, em seguida deve ser avaliado se os objetos são da mesma classe, sendo esse o caso o nó é considerado um nó folha, caso contrário um atributo precisa ser usado para dividir os dados. Este processo deve ser executado recursivamente, ele pode ser descontinuado caso faltarem atributos para realizar testes de divisão ou caso todos os registros forem da mesma classe (CASTRO; FERRARI, 2016).

Figura 3 - Modelo baseado em árvores

Fonte: Castro e Ferrari (2016, p. 166).

2.5.1.2 *Random forests*

Random forests ou florestas aleatórias são um grupo de árvores de decisões, nos quais juntos formam uma floresta. Estas árvores são geradas com base em um atributo aleatório que é o responsável pela divisão em cada nó da árvore. A precisão de uma floresta aleatória é determinada de acordo com a força de cada classificador da árvore, e também o nível de dependência entre eles, o melhor modo de atingir essa precisão é mantendo a força dos classificadores e não aumentar a correlação entre eles (HAN; KAMBER e PEI, 2012).

2.5.1.3 Classificadores bayesianos

Classificadores bayesianos têm como função classificar, se um determinado registro faz parte de uma determinada classe. Esta tarefa é realizada através da aplicação do teorema de Bayes, um princípio estatístico que vai usar conhecimentos prévios das classes em conjuntos com novos dados (TAN; STEINBACH; KUMAR, 2009)

Segundo Castro e Ferrari (2016), os classificadores possuem alta taxa de acurácia e desempenho de processamento quando aplicados a grandes bases de dados. *Naive Bayes*, é um exemplo de algoritmo de classificação bayesiano, o mesmo assume que o valor de um atributo em determinada classe tem efeito independente em relação aos

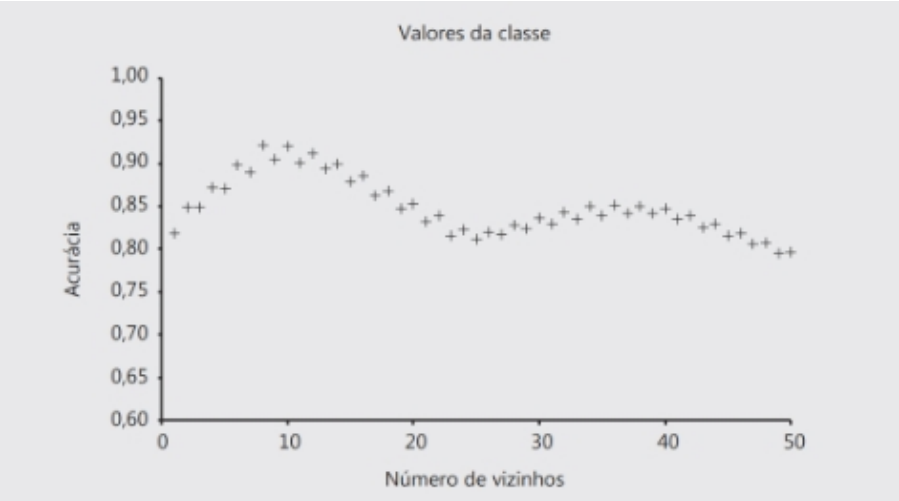
valores dos demais atributos, esse antecedente é conhecido como independência condicional da classe que tem o propósito de simplificar cálculos.

2.5.1.4 Classificadores de k vizinhos mais próximos

Segundo Han, Kamber e Pei (2012), o método de classificação k vizinhos mais próximos, tem como modo de aprendizagem a analogia, realizando a comparação de um objeto de teste com objetos semelhantes. Cada objeto é um ponto em um espaço de *n* dimensões no qual todos os objetos de treinamento são inseridos. No momento em que um objeto desconhecido é inserido no espaço, o classificador procura no espaço de padrões pelos *k* (grupos), objetos de treinamento nos quais encontram-se mais próximos do desconhecido de acordo com sua proximidade.

O método de classificação k vizinhos mais deve ser considerado como um método no qual baseia-se por instâncias, isto é, ele vai determinar a classe de um objeto desconhecido através da classe de outras instâncias. Conforme a Figura 4, um número maior de vizinhos reduz os ruídos na classificação, porém tornam as fronteiras entre as classes maiores (CASTRO; FERRARI, 2016).

Figura 4 - Acurácia do classificador de k vizinho mais próximo

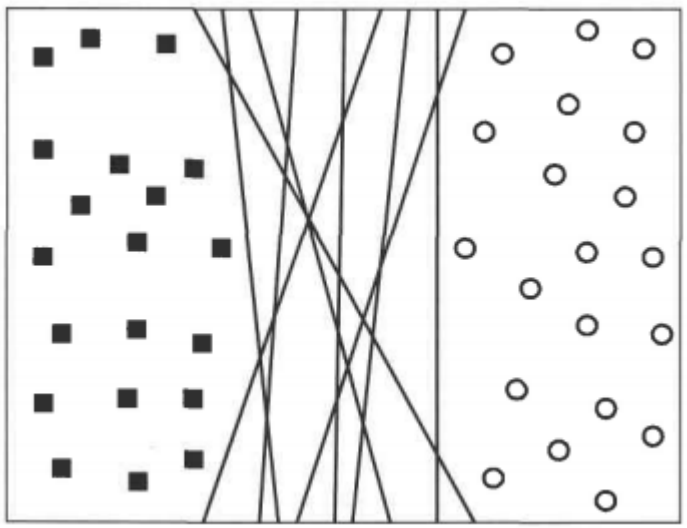


Fonte: Castro e Ferrari (2016, p. 169).

2.5.1.5 Máquinas de Vetores de Suporte

A técnica de Máquinas de Vetores de Suporte, ou *Support Vector Machine* (SVM), têm como fundamento o aprendizado em cima da estatística, o algoritmo apresenta ótima performance na utilização de dados de alta dimensionalidade. O mesmo funciona através de um conceito de hiperplano, sendo definido um limite linear neste plano para realizar a classificação, conforme a Figura 5, o algoritmo possuí a função de detectar o hiperplano de margem máxima, aquele com a maior margem separação entre as classes, com o objetivo de apresentar menos erros de generalização em relação a margens menores (TAN; STEINBACH; KUMAR, 2009).

Figura 5 - Classes separadas de forma linear



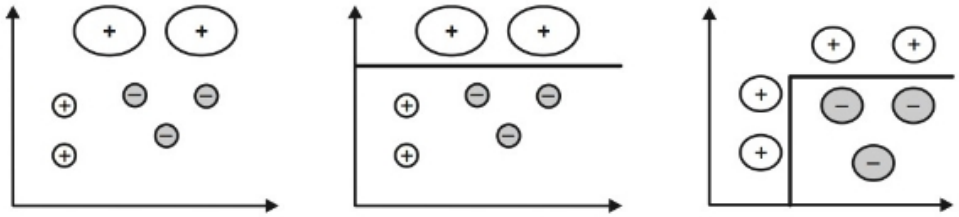
Fonte: Tan, Steinbach e Kumar (2009, p. 257).

Segundo Han, Kamber e Pei (2012), o algoritmo SVM pode ser utilizado para a previsão de dados numéricos e classificação, apresentando um alto índice de acurácia, porém a etapa de treinamento é considerada lenta. O SVM é aplicado em diversas áreas, com destaque para o reconhecimento de voz e de objetos.

2.5.1.6 Boosting

Boosting são algoritmos iterativos, que tem como objetivo central associar pesos para cada um dos atributos do conjunto de dados, focando assim em diferentes exemplos nos dados e resultando em diferentes classificadores. A cada iteração um novo classificador é gerado, sendo ele treinado com a distribuição dos pesos associados até a atual iteração, conforme Figura 6. O classificador final incorpora os classificadores assimilados de cada iteração, sendo o peso de cada classificador avaliado pela função de sua precisão. Algoritmos *boosting* possuem bom desempenho em problemas considerados do mundo real, com diferentes níveis de dificuldade de classificação (CARVALHO et. al, 2011).

Figura 6 - Iterações de classificação



Fonte: Carvalho et. al. (2011, p. 148).

2.5.1.7 Avaliação do desempenho dos classificadores

Segundo Han, Kamber e Pei (2012), um modo de avaliar o desempenho de um algoritmo de classificação, é avaliar sua capacidade preditiva. Isso deve ser realizado através da exposição do modelo de classificação a dados não vistos durante o seu treinamento, caso contrário o mesmo é incapaz de identificar ruídos ou encontra dificuldades para a generalização dos dados.

Conforme Tan, Steinbach e Kumar (2009), para a classificação de problemas binários, predicação entre duas classes, é utilizada a técnica de matriz de confusão,

indicando os dados com colunas representando as classes de previsão e linhas as classes atuais dos dados, conforme ilustrado na Tabela 1.

Tabela 1 - Matriz de confusão

		Classe Prevista	
		0	**1**
Classe Atual	**0**	Verdadeiro Negativo	Falso Positivo
	1	Falso Negativo	Verdadeiro Positivo

Fonte: do autor, adaptado de Tan, Steinbach e Kumar (2009, p.351).

Conforme os autores os termos utilizadas na composição de uma matriz de confusão são subsequentes:

- Verdadeiro Positivo (TP): número de exemplos positivos classificados corretamente;
- Falso Negativo (FN): número de exemplos negativos classificados incorretamente;
- Falso Positivo (FP): número de exemplos positivos classificados incorretamente;
- Verdadeiro Negativo (TN): número de exemplos negativos classificados corretamente.

Existem outras métricas que podem ser utilizadas para a avaliação do desempenho de classificadores conforme listado a seguir (HAN; KAMBER; PEI, 2012):

- Acuracidade: taxa dos objetos de testes classificados corretamente pelo classificador;

- Taxa de erro: percentual de objetos classificados incorretamente pelo classificador;

- Revocação: taxa de objetos positivos verdadeiros classificados de forma correta;

- Especifidade: taxa de objetos falsos verdadeiros classificados de forma correta;

- Precisão: taxa de objetos positivos classificados corretamente;

- Medida F: média harmônica entre as medidas de precisão e revocação.

Na seção seguinte são apresentadas técnicas e algorítimos de estimação, assim como métodos de avaliação de desempenho.

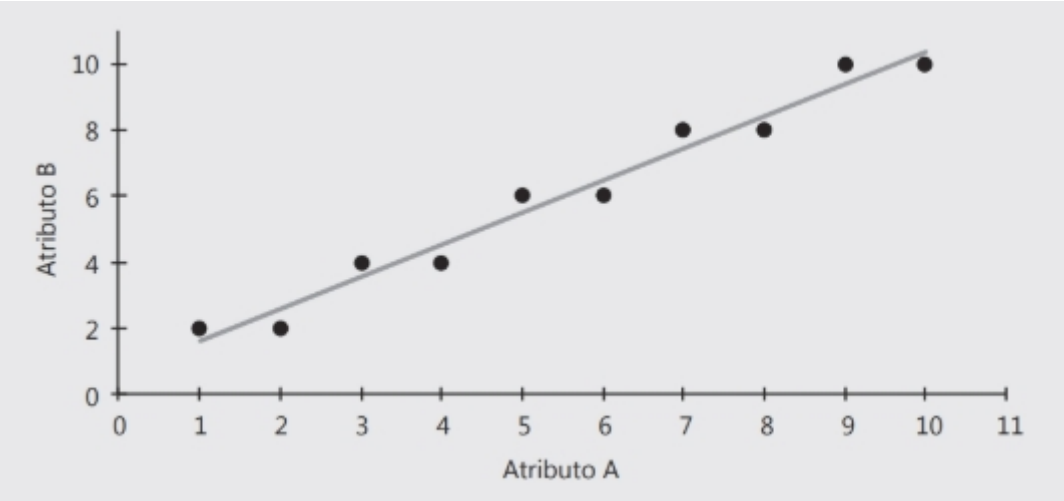
2.5.2 Técnicas de Estimação

Segundo Castro e Ferrari (2016), técnicas de estimação têm como objetivo prever um valor contínuo de uma variável alvo. As técnicas de estimação possuem muito em comum com as técnicas de classificação listadas na Seção 2.5.1, de modo que aproximadamente todos os algoritmos de estimação podem ser utilizados como classificadores, porém o contrário não pode ser dito, ainda assim algumas técnicas como árvores de decisão e classificadores bayesianos podem ser adaptados para atenderem a necessidade de estimação. A principal diferença entre as técnicas de classificação e estimação encontra-se no método de avaliação dos algoritmos.

2.5.2.1 Regressão linear

A regressão linear é responsável por modelar a associação entre uma ou mais variáveis de saída e entrada. O processo de regressão pode ser dividido em duas categorias, as paramétricas, no qual o relacionamento entre as variáveis é conhecido, e não paramétricas onde não existe conhecimento preexistente entre as variáveis. As técnicas de regressão linear procuram a relação entre duas variáveis por meio de uma equação de uma linha reta, onde é melhor representado a relação entre as variáveis conforme está ilustrado na Figura 7 (CASTRO; FERRARI, 2016).

Figura 7 - Regressão linear entre um conjunto de dados bidimensional



Fonte: Castro e Ferrari (2016, p.207).

Segundo Witten e Frank (2005), no momento no qual deseja-se estimar o valor de uma variável numérica e os atributos do conjunto de dados também são numéricos, a escolha pela técnica de regressão linear é natural, a mesma vem sendo utilizada por décadas na aplicação de problemas estatísticos, de modo que mesmo quando o conjunto de dados não apresenta uma dependência linear a aplicação do algoritmo serve como um ponto de partida para a utilização de outros algoritmos mais complexos.

Conforme Castro e Ferrari (2016), a regressão polinomial pode ser considerada uma técnica de regressão linear, a diferença está na relação entre a variável dependente

e as variáveis independentes, de modo que sua relação acaba sendo não linear e sim um polinômio de grau *n.*

2.5.2.2 Regressão logística

A regressão logística é uma técnica utilizada para a estimação de uma variável de natureza binária, estimando o valor em 0 ou 1, sendo que as variáveis independentes podem ser de natureza categórica ou não. Igualmente como na regressão linear é necessário aplicar pesos onde ajustam-se aos dados de treinamento do algoritmo, porém a regressão logística não procura a melhor reta que se ajuste aos dados, mas sim a melhor curva. A regressão logística calcula uma razão de probabilidade da variável alvo, que posteriormente é convertida em uma variável de base logarítmica, permitindo assim a classificação com base na aproximação de um dos valores (WITTEN; FRANK, 2005).

2.5.2.3 Rede neural do tipo Multi-Layer Perceptron (MLP)

Multi-Layer Perceptron (MLP) é uma rede neural artificial (RNA) de múltiplas camadas que tem o objetivo de ser capaz de solucionar problemas linearmente separáveis. Possuem como características pelo menos uma camada intermediária e altos graus de conectividade. O treinamento de uma rede MLP geralmente é realizado através de um algoritmo de retro propagação, propagação no sentido oposto. Este algoritmo funciona em duas etapas, no primeiro momento é realizada a propagação do sinal funcional com os pesos fixados, depois executa-se a retro propagação do erro na qual os pesos são ajustados de acordo com o erro (CASTRO; FERRARI, 2016).

2.5.2.4 Avaliação de desempenho da estimação

O resultado de um estimador é um valor numérico que deve ser aproximado do valor alvo desejado, a diferença entre o valor alvo e o estimado proporciona uma medida de erro de estimação, existem diversas medidas que permitem estimar o tamanho deste erro, com destaque para: Soma dos erros quadráticos; Erro quadrático médio; Erro

absoluto médio; Erro quadrático relativo; Erro absoluto relativo e Coeficiente de correlação. Qual das medidas utilizar para a avaliação do modelo de previsão, é de acordo com os dados aplicados (CASTRO; FERRARI, 2016).

2.6 Tecnologias e Ferramentas

Nesta seção são apresentadas as tecnologias e ferramentas utilizadas no desenvolvimento do trabalho, isto é, para a aplicação de algoritmos de aprendizado de máquina sobre os dados e suas respectivas análises.

2.6.1 Python

Nos últimos anos o Python tornou-se uma das mais importantes linguagens de programação para a área de *machine learning*, tanto nos ambientes acadêmicos como industriais. Uma das razões para esse crescimento é devido a melhorias que bibliotecas como Scikit-Learn e Pandas receberem, tornando a utilização da linguagem muito popular para atividades ligadas a análise de dados e ciência de dados (MCKINNEY, 2012).

2.6.2 Scikit-Learn

McKinney (2012), afirma que desde 2010 quando a biblioteca foi introduzida, tornando-se a mais popular ferramenta para a aplicação de *machine learning*, tendo a mesma disponíveis modelos de:

- Classificação: SVM, regressão logística, *random forest, boosting*;
- Regressão: Lasso, regressão de *ridge*;
- Clusterização: k vizinhos mais próximos, agrupamento;

- Pré-processamento: Normalização, seleção de atributos;

- Validação: *Cross validation, classification report*.

2.6.3 Jupyter Notebook

Jupyter Notebook é uma ferramenta que tem o objetivo de melhorar a produtividade de desenvolvimento e facilitar a interação para um interpretador da linguagem Python. Propondo um fluxo de trabalho, mas exploratório, diferente do tradicional edita, compila e executa. Desataca-se o ambiente interativo e sua facilidade para a integração com bibliotecas como Matplotlib e Pandas (MCKINNEY, 2012).

2.6.4 Pandas

A biblioteca Pandas facilita o trabalho e análise de estruturas de dados, proporcionando técnicas para agrupamento de dados, transformação e limpeza. Pandas também possibilita a manipulação de dados em forma de planilhas ou banco de dados relacionais (MCKINNEY, 2012).

As estruturas de dados na biblioteca Pandas podem ser divididas em dois grupos, *series* e *dataframes.* As *series* se assemelham com vetores, possuindo apenas uma dimensão. Os *dataframes* é uma estrutura que iguala uma planilha eletrônica, contendo uma coleção de colunas, possuindo índice e podendo receber diferentes tipos de dados. Arquivos de extensão CSV e XLSX podem ser lidos e suas estruturas são replicadas identicamente para um *dataframe* (MCKINNEY, 2012).

2.6.5 NumPy

NumPy é a biblioteca base para computação científica em Python. Proporcionando funções para realizar operações matemáticas entre vetores, álgebra linear, números

aleatórios e oferecendo possibilidade de integração nativa com outras linguagens de programação como C e C++ (MCKINNEY, 2012).

2.6.6 Matplotlib

Segundo McKinney (2012), a biblioteca Matplotlib é a mais utilizada para a visualização de dados, como, por exemplo, gráficos de duas dimensões. A biblioteca também destaca-se pela fácil integração com o restante do ambiente Python, sendo que diversas funções de outras bibliotecas como Pandas utilizam a biblioteca Matplotlib para a visualização de dados.

2.6.7 PostgreSQL

O PostgreSQL é um sistema de gerenciamento de banco de dados (SGBD) relacional, sendo uma ferramenta de código aberto permitindo que a mesma seja alterada e distribuída para fins privados, comerciais ou acadêmicos, o PostgreSQL teve como base de origem o POSTGRES 4.2 que foi desenvolvido no departamento de Ciência da Computação da Universidade da Califórnia. O PostgreSQL oferece as funções e operações bases de todos os SGBD (POSTGRESQL, 2019).

2.6.8 WEKA

O WEKA (Waikato Environment for Knowledge Analysis), é um software com o propósito de oferecer, um variado conjunto de opções para aplicações de técnicas de aprendizado de máquina e mineração de dados. Disponibilizando recursos para a realização de etapas do pré-processamento de dados assim como a aplicação de algoritmos de classificação, agrupamento e regressão. O WEKA pode ser utilizado através de interface gráfica ou interface via linha de comandos, também é disponibilizada uma API para integração com desenvolvimento Java (WITTEN; FRANK, 2005).

2.7 Estatísticas de jogos de basquete

Nesta seção são descritas as estatísticas registradas em partidas da NBA, as quais estão presentes nos dados coletados e são utilizadas para a aplicação dos algoritmos. Estão detalhadas suas respectivas descrições e equações para as que necessitam.

As informações gerais das partidas representam dados que não envolvem estatísticas do jogo em questão, e sim dados informativos. A Tabela 2 lista os dados e suas respectivas descrições.

Tabela 2 - Informações gerais

Dado	Descrição
Equipe	Equipe mandante da partida
Oponente	Equipe visitante da partida
Temporada	Temporada da partida
Dias de descanso time	Nº de dias sem jogos
% de Vitórias	Percentual de jogos vencidos na temporada
spread	Total de pontos favorito ou azarão

Fonte: do autor, adaptado de Sports Interaction (2019).

A possibilidade de apostar em jogos esportivos, de modo diferente de simplesmente escolher o seu vencedor, foi um dos motivos para que Charles K. McNeil criasse o sistema de pontos ou *spread*. O *spread* apresenta como base uma análise de ambas equipes da partida, determinando assim quantos pontos uma equipe é favorita ou azarão. Sendo assim, para o apostador vencer, a equipe no qual ele apostou, precisa ultrapassar a margem de pontos caso ele seja o favorito, ou ficar abaixo da margem de pontos no caso do azarão. O *spread* motivou apostadores a apostarem em ambas as equipes, diferente do sistema de apenas escolher o vencedor da partida, equilibrando assim as apostas nas diversas bolsas de apostas (KELLY, 2013).

A cada partida da NBA são registradas diversas estatísticas, a Tabela 3, apresenta as principais delas.

Tabela 3 - Estatísticas registradas por partida

Estatística	Descrição
PTS	Pontos marcados
FGA	Tentativa de arremesso
FGM	Arremesso convertido
2PA	Tentativa de arremesso de dentro do garrafão com valor de 2 pts
2PM	Arremesso convertido de dentro do garrafão com valor de 2 pts
3PA	Tentativa de arremesso de fora do garrafão com valor de 3 pts
3PM	Arremesso convertido de fora do garrafão com valor de 3 pts
FTA	Tentativa de arremesso de fora do garrafão com valor de 3 pts
FTM	Arremesso convertido de fora do garrafão com valor de 3 pts
AST	Passe que resulta na conversão de um arremesso do jogador recebedor
TOV	Perda da posse de bola
REB	Recuperar a bola após um arremesso não convertido
OREB	Recuperar a bola após um arremesso não convertido do próprio time
DREB	Recuperar a bola após um arremesso não convertido do time adversário
BLK	Bloquear o arremesso do adversário de chegar a cesta
STL	Roubar a bola do adversário

Fonte: do autor, adaptado de NBA (2019b).

Ao longo dos anos a análise quantitativa das partidas cresceu, sendo realizados estudos de origem acadêmica e também não tradicionais. Neste contexto foram criadas estatísticas que são conhecidas como estatísticas avançadas, baseadas em conceitos de posse de bola, conforme equação (1), onde 0.44 é o coeficiente de lances livres que resultam no fim de uma posse de bola (KUBATKO; OLIVER; PELTON; ROSENBAUM, 2007).

$$POSS = FGA + 0.44 \times FTA - OREB + TOV \quad (1)$$

Ao definir o que é uma posse de bola, ela pode ser utilizada para avaliar a eficiência das equipes por posse de bola, isolando assim a qualidade de comparação das equipes. A estatística mais comum para essa avaliação é *rating*, nas quais os pontos marcados e sofridos são avaliados a cada 100 posses de bola, conforme equações (2) e (3), onde PTS representa os pontos marcados e Oppt PTS os pontos sofridos pela equipe. A estatística de avaliação geral de eficiência de uma equipe é o *NET rating*, conforme equação (4) (KUBATKO; OLIVER; PELTON; ROSENBAUM, 2007).

$$ORTG = PTS \div POSS \times 100 \quad (2)$$

$$DRTG = Oppt\ PTS \div POSS \times 100 \quad (3)$$

$$NET\ RTG = ORTG - DRTG \quad (4)$$

Para avaliar a eficiência na tentativa de pontuar, foi criado o *True Shooting Percentage* (TS%), conforme a equação (5), estatística na qual os lances livres e arremessos de três pontos são considerados, não apenas o número total de arremessos convertidos divididos pelo número total de arremessos tentados (KUBATKO; OLIVER; PELTON; ROSENBAUM, 2007).

$$TS\% = (PTS \div 2) \div (FGA + 0.44 \times FTA) \quad (5)$$

Oliver (2004), identificou quatro fatores principais em uma partida de basquete, que com desempenho aumenta as chances de a equipe vencer a partida. Um dos fatores é o *Effective Field Goal Percentage* (EFG%), estatística de eficiência de tentativas de marcar pontos, considerando o número de arremessos de três pontos convertidos, conforme equação (6).

$$EFG\% = (FGM + 0.5 \times 3\,PM) \div FGA \quad (6)$$

Turnover Percentage (TOV%) é o segundo fator, o mesmo considera o número de perdas de bola do ataque pelo número total de posses de bola, conforme equação (7).

$$TOV\% = (TOV \div POSS) \times 100 \quad (7)$$

O terceiro fator é *Offensive Rebound Percentage* (ORB%), no qual é avaliado o número de rebotes disponíveis para o ataque que o mesmo conseguiu pegar, aumentando assim o número de tentativas de marcar pontos, conforme equação (8).

$$ORB\% = ORB \div (ORB + Oppt\ DREB) \quad (8)$$

O último fator, *Free Throw Rate* (FTR), avalia a capacidade de um time bater lances livres, em relação ao total de arremessos tentados, conforme equação (9).

$$FTR = FTA \div FGA \quad (9)$$

2.8 Trabalhos relacionados

Nesta seção são descritos dois trabalhos relacionados com a proposta de desenvolvimento deste trabalho, nos quais são analisados os resultados obtidos e técnicas de mineração de dados e aprendizado de máquina, aplicados pelos autores com o objetivo de compará-los com o que foi desenvolvido neste trabalho.

Os trabalhos apresentados nesta seção têm como principal característica o objetivo de classificar o vencedor de jogos da NBA com base em dados estatísticos históricos, nas seções seguintes são apresentados os trabalhos com suas respectivas propostas e um comparativo de ambos.

2.8.1 Descrição dos trabalhos

Cao (2012), tem como proposta em sua dissertação criar um modelo de previsão de jogos da NBA, utilizando técnicas de *machine learning.* O autor coletou dados de cinco temporadas da NBA para realizar o treinamento dos algoritmos e uma temporada para a validação do modelo. Foram utilizados algoritmos de regressão linear, redes neurais artificiais, SVM e *Naive Bayes.*

Torres (2013), propõem em seu artigo a criação de um modelo capaz de prever o vencedor de jogos da NBA e ter uma porcentagem de acertos superior do que

simplesmente prever como o vencedor o time com menos derrotas na temporada, foram coletados dados de sete temporadas, entre os anos de 2006 e 2013. Foram aplicados algoritmos de MLP, regressão linear e um classificador bayesianos.

2.8.2 Resultados obtidos

Nesta seção está presente uma comparação entre os resultados obtidos pelos trabalhos de Cao (2012) e Torres (2013), a Tabela 4, apresenta os algoritmos aplicados e os respectivos percentuais de acuracidade de cada um.

Tabela 4 - Acuracidade das técnicas aplicadas

Técnica aplicada	Cao (2012)	Torres(2013)
SVM	67.70%	Não aplicado
RNA	68.01%	68.44%
Classificadores bayesianos	66.25%	66.81%
Regressão logística	69.67%	Não aplicado
Regressão linear	Não aplicado	67.89%

Fonte: do autor.

Os resultados obtidos por estes trabalhos relacionados, serão usados para avaliar o desempenho do modelo criado neste trabalho. No capítulo 3 é apresentada a metodologia do trabalho.

3 METODOLOGIA

Este trabalho busca comparar técnicas de mineração de dados e aprendizado de máquina e criar um modelo de previsão para jogos de basquete da NBA baseado em estatísticas de jogos passados. Deste modo o trabalho caracteriza-se como um estudo experimental e exploratório. Pesquisas exploratórias, tem como objetivo permitir uma perspectiva geral acerca do assunto, em muitas sendo uma etapa inicial de um estudo mais amplo (GIL, 2008).

3.1 Tipo de pesquisa

Pesquisas exploratórios necessitam menos severidade no planejamento de sua realização, frequentemente exigindo estudo bibliográfico e a realização de entrevistas que não são padronizadas. O final deste processo resulta em um problema mais esclarecido, sujeito a averiguações por meio de processos mais sistemáticos (GIL, 2008).

Conforme Gil (2008), estudos experimentais são os que melhor representam pesquisas científicas. Este processo incluí definir um tema de estudo e a seleção de variáveis capazes de induzir o objeto de estudo, sendo preciso definir formas de observação e controle sobre o impacto das variáveis sobre o tema. Um dos métodos de um estudo experimental é através de anular a influência de fatores, sendo assim observar quais dos fatores é necessário para a produção de determinado fenômeno. Pesquisas de

características experimentais normalmente ficam limitas caso o objeto de estudo seja social, devido a questões éticas e humanas.

Segundo Wainer (2007), pesquisas experimentais são atividades caracterizadas pela manipulação de variáveis e observação de outras. Já pesquisas exploratórias devem descrever um fenômeno e realizar propostas para novas teorias, métricas para novas medições ou observações acerca do fenômeno.

Neste trabalho é realizada uma pesquisa sobre o domínio de conhecimento da mineração de dados e aprendizado de máquina, sendo que isso compõe a fundamentação teórica do trabalho. As próximas seções detalham os processos que envolvem a aplicação destes conceitos e os dados explorados.

3.2 Coleta de dados

O conjunto de dados utilizadas nesta monografia são dados estatísticos e informações de jogos da NBA de 2013 em diante, sendo coletadas cinco temporadas completas, totalizando mais de seis mil partidas no período. Todas as partidas são referentes a temporada regular da NBA, não foram coletadas partidas da pós-temporada pelo alto grau de variedade, pois a cada temporada, os participantes são diferentes e normalmente os times realizam menos de vinte e cinco partidas em cada edição da pós-temporada.

Entre os dados estatísticos e informações coletadas, estão presentes informações de cada equipe nas partidas realizadas, conforme a Seção 2.7.

Os dados foram extraídos de duas fontes em formatos CSV e XLSX, ambos conjuntos foram inseridos em um banco de dados relacional para posteriormente serem aplicadas as ações de pré-processamento dos dados. Segundo Castro e Ferrari (2016), um conjunto de dados organizados permite uma eficiente recuperação e pré-processamento dos atributos para posteriormente no processo ser realizada a extração de conhecimento.

3.3 Pré-processamento

O pré-processamento é a etapa que são realizadas ações visando preparar os dados para o processo de aprendizado, isso inclui a limpeza, integração, transformação e seleção dos atributos mais relevantes (CASTRO; FERRARI, 2016).

Neste processo a primeira ação a ser realizada é a integração dos dois conjuntos de dados, ambos os conjuntos foram inseridos em tabelas diferentes de um banco de dados PostgreSQL, com isso foi criada uma nova tabela para esta integração com todos os atributos de cada conjunto. Neste processo também foi necessária a transformação de determinados atributos enquanto, outros tornaram-se redundantes, já que estavam presentes em ambos os conjuntos.

O próximo passo do pré-processamento é a transformação das estatísticas dos jogos, essa etapa consiste em acumular os dados anteriores de cada equipe em relação a cada partida. Como o objetivo é classificar o vencedor baseado no histórico dos times, não pode ser utilizado os dados da partida em questão para a classificação. Para isso foram criadas funções no banco de dados nas quais as estatísticas são acumuladas e calculadas para cada equipe e partida.

Para a aplicação dos algoritmos de classificação ainda foi necessária a conversão de atributos textuais para numéricos, como o nome de cada equipe e temporada da partida em questão, para isso foi utilizada a biblioteca Scikit-Learn e seu pacote *preprocessing*.

3.4 Aprendizado

A etapa de aprendizado representa a exploração dos dados pré-processados, buscando extrair tendências e influências, em relação as estatísticas e informações do conjunto de dados. Para a seleção dos atributos foram utilizados algoritmos de seleção específicos para isso, posteriormente os atributos selecionados foram aplicados nos algoritmos citados nas seções 2.5.1 e 2.5.2.

Para a aplicação dos algoritmos é utilizada a biblioteca Scikit-Learn em um ambiente de desenvolvimento Python, a mesma possui todos os algoritmos necessários de classificação, neste mesmo ambiente também é utilizada a biblioteca Pandas para a manipulação dos dados. Os dados foram divididos em conjuntos de treinamento e teste, sendo que o conjunto de treinamento representa dois terços do total de dados.

3.5 Avaliação

O processo de avaliação consiste em, após o processo de treinamento, tendo como dados de entradas os atributos selecionados de cada equipe anterior a partida a ser disputada, comparar qual dos algoritmos apresenta o maior índice de acuracidade na classificação do vencedor de cada partida. Os algoritmos também são avaliados em métricas como precisão, revocação e Medida F.

A partir disso é possível concluir qual dos algoritmos apresenta o melhor desempenho para a classificação e utilizá-lo para o modelo de previsão para futuras partidas. Também são comparados os resultados obtidos com os resultados citados na Seção 2.8.2 referentes a trabalhos relacionados.

No próximo capítulo deste trabalho está presente o desenvolvimento do trabalho, detalhando as tarefas realizadas e resultados obtidos.

4 DESENVOLVIMENTO

Neste capítulo são apresentados os processos realizados durante o desenvolvimento do trabalho, visando alcançar os objetivos propostos. Nas próximas seções são apontados os dados de entrada, detalhando sua origem, atributos e os processos de pré-processamento de dados. No capítulo também são apresentados os resultados obtidos no processo de mineração de dados, com a comparação de desempenhos entre os algoritmos aplicados.

4.1 Dados de Entrada

Os dados utilizados neste projeto foram coletados de dois sites de estatísticas de jogos, bigdataball.com e kaggle.com, estes dados foram exportados em arquivos de formato XLSX e CSV respectivamente. Os dados contêm estatísticas de cada time em cada partida realizada. Foram coletados dados de cinco temporadas completas de jogos, no período entre 2013 e 2018, dentro de cada temporada cada equipe realiza no mínimo um total de 82 jogos e ao total foram coletados dados estatísticos de 7380 partidas.

A Figura 8 apresenta um exemplo dos dados coletados do site bigdataball.com, cada linha representa as estatísticas do time em questão na partida que enfrentou o time da linha abaixo. Entre as estatísticas presentes estão o total de pontos de cada time, arremessos convertidos, arremessos tentados e lance livres convertidos. Em comparação,

os dados coletados no site kaggle.com listam as estatísticas da partida de ambos os times envolvidos em uma única linha.

Figura 8 - Exemplo do conjunto de dados

1	DATASET	DATE	TEAMS	F	MIN	FG	FGA	3P	3PA	FT	FTA
2	2013-2014 Regular Season	10/29/2013	Orlando	87	240	36	93	9	19	6	10
3	2013-2014 Regular Season	10/29/2013	Indiana	97	240	34	71	7	17	22	32
4	2013-2014 Regular Season	10/29/2013	Chicago	95	240	35	83	7	26	18	23
5	2013-2014 Regular Season	10/29/2013	Miami	107	240	37	72	11	20	22	29
6	2013-2014 Regular Season	10/29/2013	LA Clippers	103	240	41	83	8	21	13	23
7	2013-2014 Regular Season	10/29/2013	LA Lakers	116	240	42	93	14	29	18	28

Fonte: do autor.

Na seção seguinte é abordado a etapa de pré-processamento dos dados, onde é apresentado como os dados foram preparados para a aplicação das técnicas de mineração de dados.

4.2 Pré-processamento dos dados

Neste processo, a primeira ação realizada foi a integração dos dois conjuntos de dados, ambos os conjuntos foram inseridos em tabelas diferentes de um banco de dados PostgreSQL, com isso foi criada uma nova tabela chamada partidas para esta integração com todos os atributos de cada conjunto. Neste processo foi necessária a transformação de determinados atributos, conforme Tabela 5, enquanto outros tornaram-se redundantes já que estavam presentes em ambos os conjuntos, sendo assim mantidos apenas a versão de um dos conjuntos.

Tabela 5 - Diferença de atributos entre os conjuntos de dados

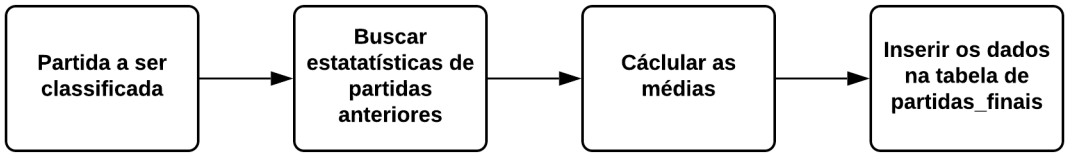
Origem dos dados	Formato de Data	Identificação das equipes
Bigdataball.com	Mês/Dia/Ano	Cidade da equipe
	10/30/2013	San Antonio
	11/02/2013	Chicago
Kaggle.com	Ano-Mês-Dia	Nome completo abreviado
	2013-10-30	SAS
	2013-11-02	CHI

Fonte: do autor.

As conversões realizadas foram padronizar os campos de data para o formato de Ano-Mês-Dia e a identificação das equipes para o nome das respectivas cidades. Após o processo de conversão dos dados já foi possível realizar a integração dos dois conjuntos.

O próximo passo do pré-processamento foi a transformação das estatísticas dos jogos, essa etapa consiste em acumular os dados anteriores a partida em questão de cada equipe, conforme Figura 9. Sendo assim a classificação dos jogos é realizada de acordo com o histórico da equipe até a data da partida.

Figura 9 - Fluxograma do processo de acumular dados



Fonte: do autor.

Para isso foram criadas funções no banco de dados nas quais as estatísticas são acumuladas e calculadas para cada equipe e partida. A Figura 10 exibe a função criada para calcular o percentual de vitórias na temporada, essa informação não estava presente nos dados anteriormente.

Figura 10 - Função SQL para calcular o percentual de vitórias

```
CREATE OR REPLACE FUNCTION fRetornaWinPercentage (team text, date_game text, season text) RETURNS TEXT AS $$
DECLARE
 retorno text;
BEGIN
 SELECT
 CASE WHEN sub3.jogos_total=0 THEN '-1'
 ELSE CAST(CAST((sub1.vitorias_casa+sub2.vitorias_fora) as decimal)/cast(sub3.jogos_total as decimal) as text)  END
 into retorno
 FROM (select COUNT(*) as vitorias_casa from partidas
 where temporada=season
 and teamabbr=team
 and gmdate<date_game
 and teamrslt='1') sub1,(select COUNT(*) as vitorias_fora from partidas
 where temporada=season
 and opptabbr=team
 and gmdate<date_game
 and opptrslt='1') sub2,(select COUNT(*) as jogos_total from partidas
 where temporada=season
 and (teamabbr=team or opptabbr=team)
 and gmdate<date_game
 ) as sub3;
 RETURN retorno;
END
$$ LANGUAGE plpgsql;
```

Fonte: do autor.

As estatísticas referentes aos fatores de Oliver, Seção 2.7, precisaram ser calculadas já que não constavam nos dados, para estes cálculos foi criada uma rotina SQL. A função recebe como parâmetros as duas equipes envolvidas na partida, data e temporada, a partir disso acumula os dados anteriores e calcula cada uma das estatísticas. Todos os dados acumulados e calculados foram inseridos na tabela partidas_finais.

A etapa final do pré-processamento foi a discretização dos dados, feito utilizando o software WEKA. Todos os atributos do conjunto de dados com exceção dos atributos de data e hora do jogo, equipes e temporada foram discretizados, estes mesmos atributos não são utilizados na aplicação dos algoritmos e por isso não passaram por esse processo. A função de discretização que o WEKA disponibiliza permite parametrizar o número de classes que devem ser criadas, com isso foram criadas 7 classes para cada atributo.

Para a utilização desses dados na biblioteca Scikit-Learn, foi necessário atribuir um número discreto para cada uma das classes criadas. Isso foi realizado utilizando o pacote preprocessing da biblioteca Scikit-Learn, conforme Figura 11.

Figura 11 - Atribuição de números para as classes de cada atributo

```
#atribuir numero para cada classe

#inicia o pacote de pré-processamento
le = preprocessing.LabelEncoder()

#percorrer todos atributos
for column in df:
    #atribuir número discreto e substitui no dataframe
    le.fit(df[column])
    df[column]=le.transform(df[column])
```

Fonte: do autor.

Depois dos procedimentos realizados nesta seção, os dados estão prontos para a aplicação dos algoritmos de aprendizado de máquina. Os resultados obtidos após a aplicação são discutidos na Seção 4.5.

4.3 Exploração dos dados

Nesta seção são explorados as estatísticas e informações presentes no conjunto de dados depois do processo de integração, para identificar tendências e quais os atributos podem ser mais influentes no resultado das partidas.

Um dos fatores mais importantes em disputas esportivas é o mando de quadra, na NBA isso não é diferente, equipes que atuam diante de seu torcedor apresentam um aproveitamento próximo a 60% em todas as temporadas analisada. A Tabela 6 apresenta estes dados para cada temporada, sendo este um fator que sofre de pouca variação entre as temporadas, isso indica que a maioria das partidas presentes no conjunto de dados vai ter como classificação correta a equipe mandante.

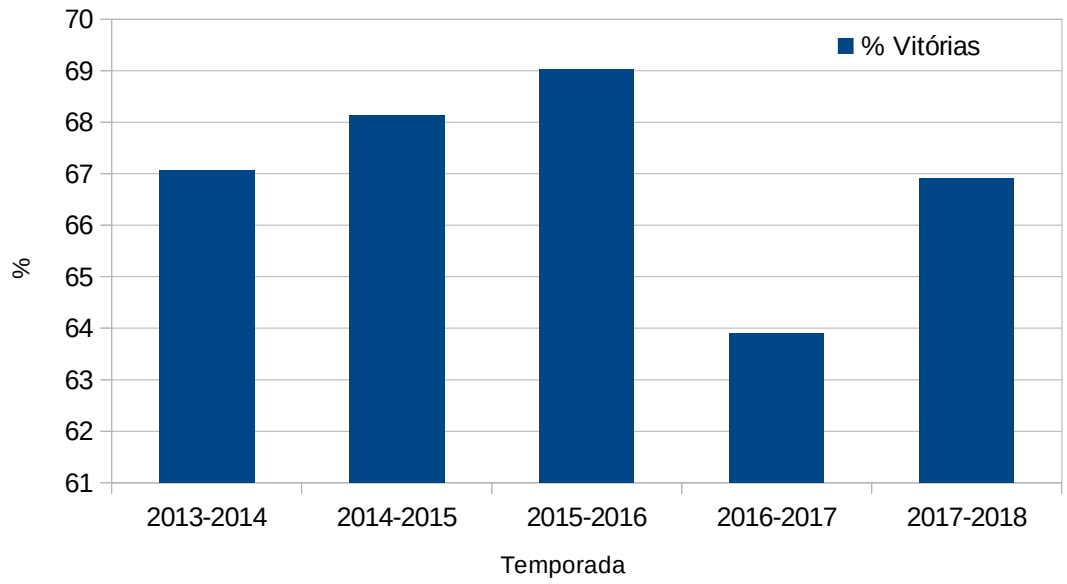
Tabela 6 - Vitórias de mandantes por temporada

Temporada	% Vitória dos mandantes
2013-2014	58.0488
2014-2015	57.4797
2015-2016	58.8618
2016-2017	58.3740
2017-2018	57.8862

Fonte: do autor.

Outro fator importante dos dados é a equipe considerada a favorita antes do confronto ser realizado. Este favoritismo é representado pelo sistema de pontos *spread*, conforme citado na Seção 2.7. Avaliando os dados, o favorito possuí uma vantagem considerável de aproveitamento, sendo superior ao aproveitamento das equipes mandantes, porém entre as temporadas avaliadas existe uma maior variação desse fator de uma temporada para outra conforme o Gráfico 1 exibe.

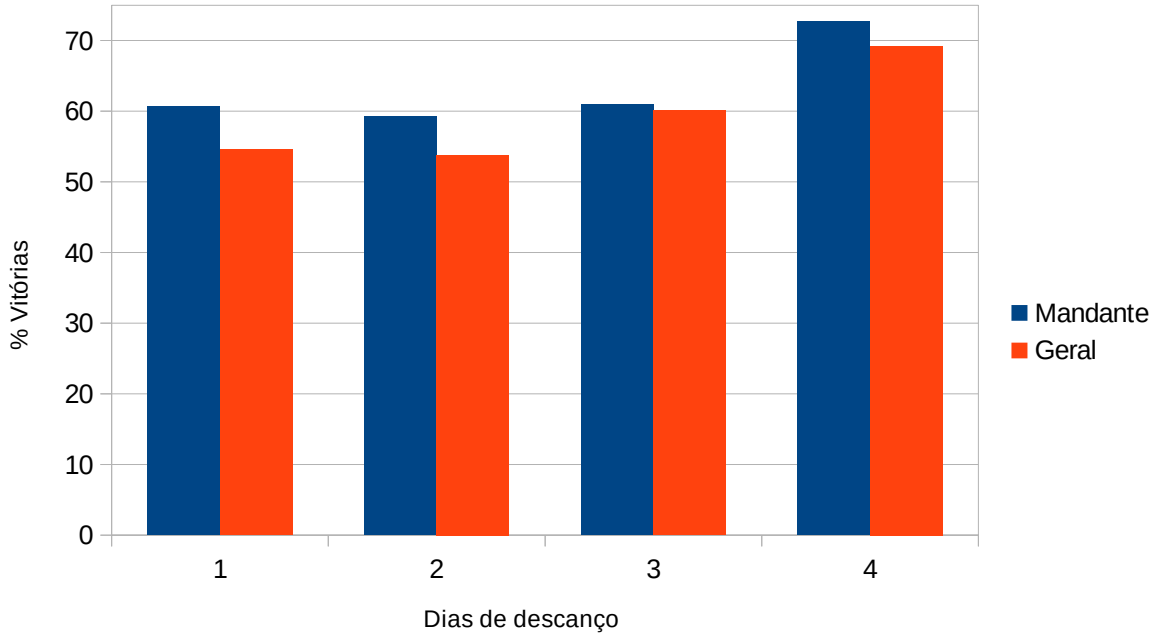
Gráfico 1 - Relação de vitória dos favoritos



Fonte: do autor.

Durante a temporada regular da NBA cada equipe em média joga quatro vezes por semana, com isso as partidas ocorrem frequentemente com uma das equipes tendo vantagem de descanso em relação a outra. Avaliando os jogos com essa situação de vantagem conforme o Gráfico 2, existe um aproveitamento superior de times com descanso de dois a quatro dias, especialmente se o mesmo é o mandante da partida. Existem situações nas quais as equipes se enfrentam com mais de quatro dias de descanso, porém estes casos são raros e de pequena amostragem, por isso não foram considerados no gráfico.

Gráfico 2 - Aproveitamento de equipes por dia de vantagem de descanso



Fonte: do autor.

O percentual de vitórias da equipe na temporada se mostrou um importante indicador de qual equipe vence a partida, a Tabela 7 apresenta essas informações por temporada e indica que um time com percentual de vitória superior jogando como mandante da partida tem aproveitamento ainda mais alto, tendo em médio 10% a mais de aproveitamento do que a avaliação geral.

Tabela 7 - Percentual de vitórias do time com melhor campanha

Temporada	% Vitórias	% Vitórias como mandante
2013-2014	64.5134	74.0283
2014-2015	66.2742	75.9717
2015-2016	64.9493	75.9931
2016-2017	59.9832	70.3041
2017-2018	62.4789	72.0486

Fonte: do autor.

4.4 Seleção dos atributos

A etapa de seleção dos atributos para a aplicação dos algoritmos, foi feita utilizando diferentes técnicas, com o objetivo de selecionar apenas os atributos de maior influência. Nesta seção são apresentados os procedimentos realizados, os algoritmos aplicados e quais os atributos selecionados.

Para a avaliação de quais atributos são mais influentes foram aplicados três algoritmos. Selecionando os dez atributos mais influentes baseados em uma média entre os resultados de cada algoritmo, os resultados destes algoritmos, avaliando o peso de influência de cada atributo, foram armazenados em uma matriz para comparação posterior e cálculo de média de peso para cada um dos atributos. Antes de armazenar os resultados foi preciso aplicar um algoritmo, MinMaxScaler, com o propósito escalonar os números dos resultados já que os algoritmos aplicados atribuem pesos em diferentes faixas de valores, a Figura 12 apresenta o fluxograma dessa etapa.

Figura 12 - Fluxograma para armazenamento dos resultados

Fonte: do autor.

A definição dos atributos utilizados nesta seção do trabalho, está presente na Tabela 8. Estes atributos foram os que apresentaram o maior peso de influência pelos algoritmos aplicados.

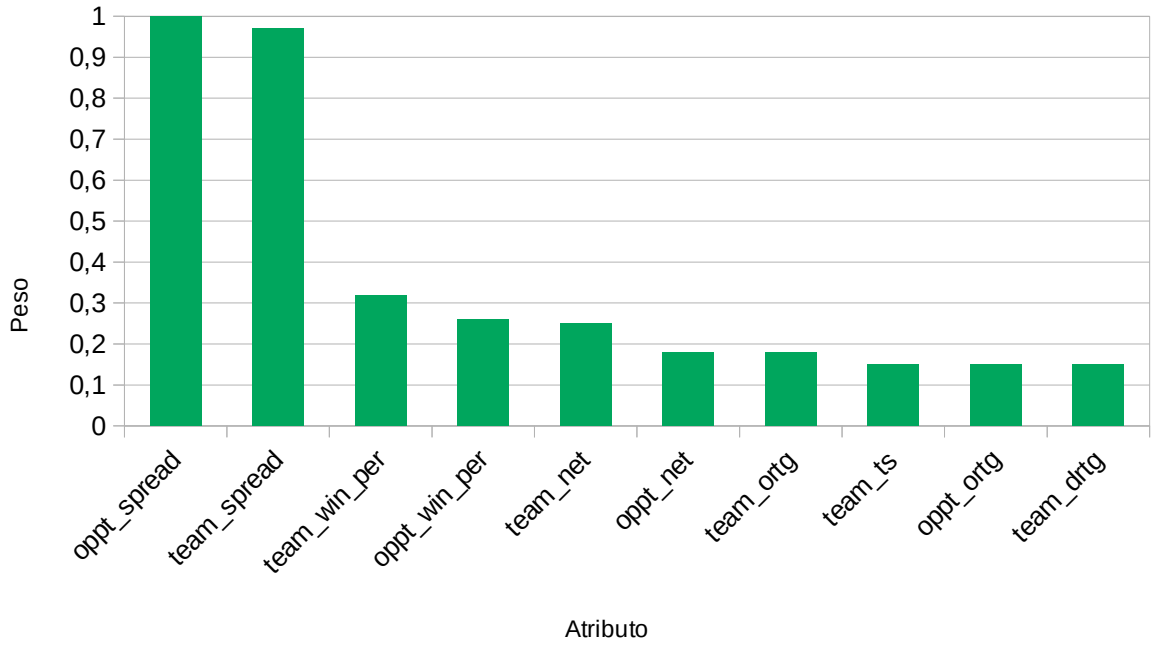
Tabela 8 - Definição dos atributos

Atributo	Estatística/Informação	Atributo	Estatística/Informação
team_spread	*Spread* de pts. do mandante	team_ass	Assistências do mandante
oppt_spread	*Spread* de pts. do visitante	oppt_ass	Assistências do visitante
team_win_per	% Vit. do mandante	oppt_ortg	*Rating* ofensivo do visitante
oppt_win_per	% Vit. do visitante	oppt_efg	EFG% do visitante
team_net	*Net rating* do mandante	team_3pm	3PM do mandante
oppt_net	*Net rating* do visitante	oppt_3pm	3PM do visitante
team_ts	TS% do mandante	team_drb	Rebotes defensivos do mandante
team_ortg	*Rating* ofensivo do mandante	oppt_drb	Rebotes defensivos do visitante
team_drtg	*Rating* defensivo do mandante	oppt_orb	Rebotes ofensivos do visitante
oppt_orb	Rebotes ofensivos do visitante	team_orb	Rebotes ofensivos do mandante
oppt_reb	Rebotes do visitante	team_reb	Rebotes do mandante

Fonte: do autor.

O primeiro algoritmo aplicado foi o ExtraTreesClassifier, sendo parametrizado 250 árvores na floresta do algoritmo. Conforme apresentado no Gráfico 3, os atributos mais influentes foram a *spread* de pontos e o percentual de vitória. Sendo que a *spread* de pontos apresenta uma influência muito grande em comparação aos demais atributos do conjunto.

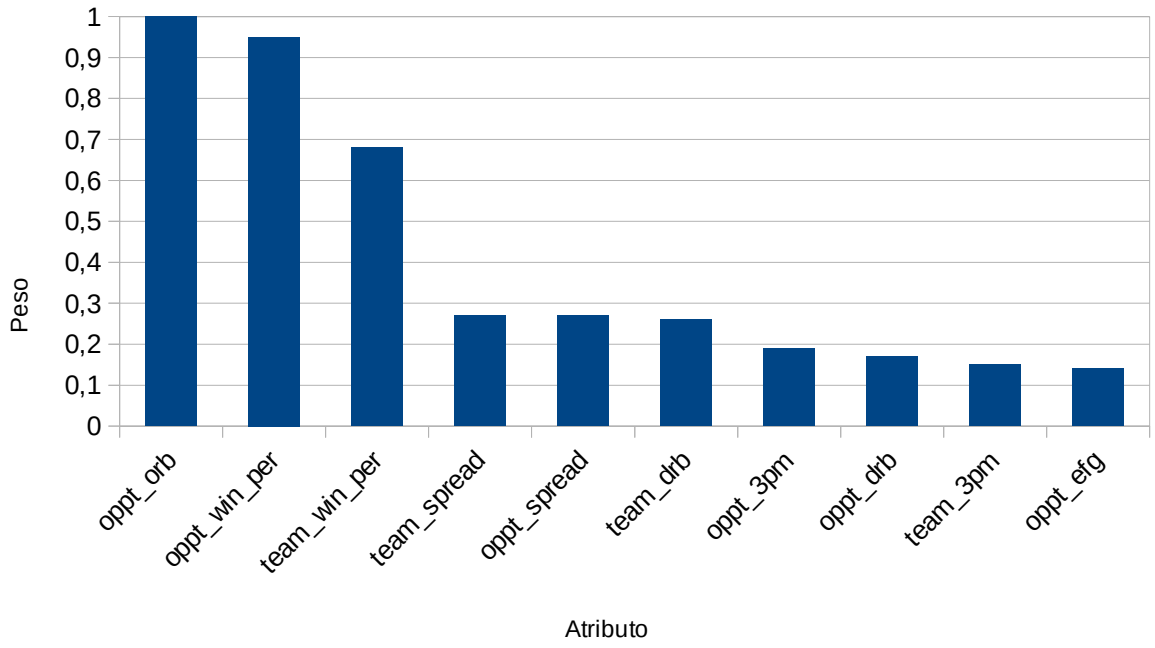
Gráfico 3 - Peso dos atributos com ExtraTreesClassifier



Fonte: do autor.

O segundo algoritmo aplicado foi o Ridge, conforme o Gráfico 4 exibe, os resultados foram diferentes do algoritmo ExtraTreesClassifier, atribuindo um peso maior para os rebotes ofensivos, o atributo de percentual de vitórias também foi atribuído como um dos mais influentes, porém sendo maior que a *spread* de pontos.

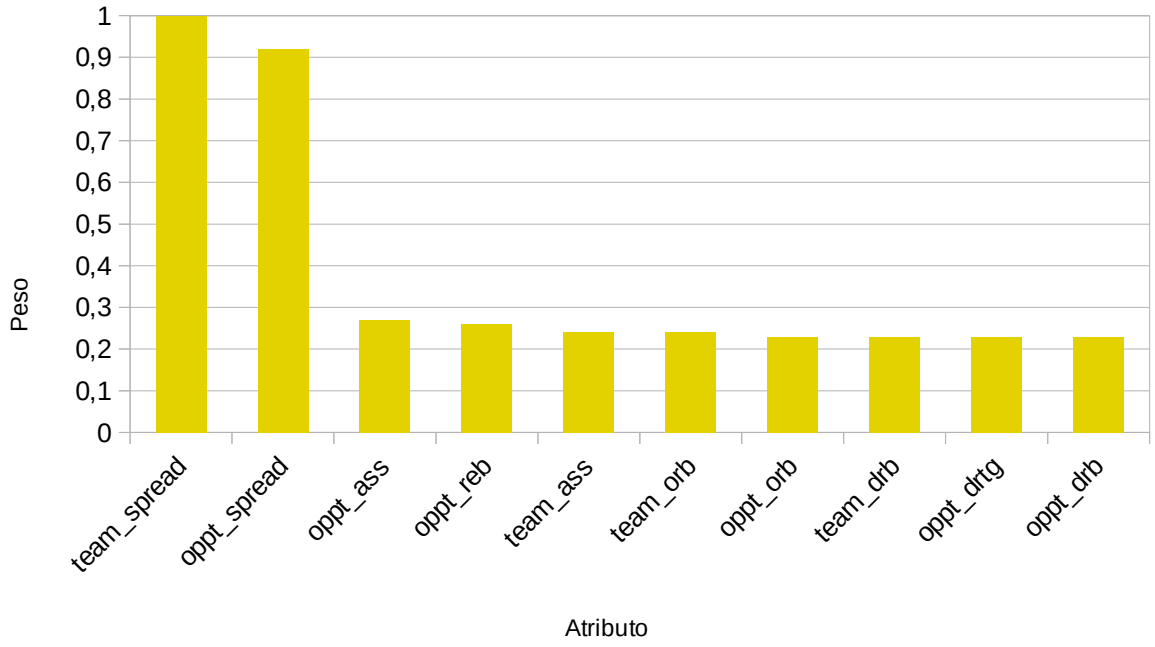
Gráfico 4 - Peso dos atributos com Ridge



Fonte: do autor.

O último algoritmo aplicado foi o RandomForestRegressor, assim como o algoritmo ExtraTreesClassifier foram parametrizadas 250 árvores para a floresta do algoritmo. Os resultados exibidos no Gráfico 5, mostram semelhança com o algoritmo ExtraTreesClassifier, com grande influência para o *spread* de pontos, porém o percentual de vitórias não aparece entre os dez atributos de maior peso. O algoritmo também não atribuiu entre os mais influentes nenhuma estatística referente a arremessos ou eficiência de arremessos.

Gráfico 5 - Peso dos atributos com RandomForestRegressor



Fonte: do autor.

Com os pesos dos atributos armazenados em uma matriz, foi calculada a média de cada um deles. A Tabela 9 relaciona os dez atributos com melhor média e seus respectivos pesos em cada um dos algoritmos aplicados. Esses dez atributos foram selecionados para a aplicação dos algoritmos de classificação dos resultados.

Tabela 9 - Comparação dos pesos dos atributos

Atributo	ExtraTreesClassifier	Ridge	RandomForestRegressor	Média
team_spread	0.97	0.27	1.00	0.75
oppt_spread	1.00	0.27	0.92	0.73
oppt_win_per	0.26	0.95	0.19	0.46
oppt_orb	0.10	1.00	0.23	0.44
team_win_per	0.32	0.68	0.17	0.40
team_drb	0.11	0.26	0.23	0.20
oppt_drb	0.10	0.17	0.23	0.17
oppt_ass	0.14	0.07	0.27	0.16
team_ass	0.14	0.06	0.24	0.15
oppt_efg	0.15	0.14	0.17	0.15

Fonte: do autor.

Depois de os atributos serem selecionados nesta seção, os algoritmos de classificação foram executados utilizando a biblioteca Scikit-Learn, os resultados obtidos e comparações são apresentados na próxima seção.

4.5 Resultados

Os resultados obtidos são resultados da aplicação de algoritmos pertencentes à biblioteca Scikit-Learn. Foi possível realizar diversos testes com a ferramenta, explorando e validando os dados, nesta essência a biblioteca foi utilizada como uma ferramenta exploratória, na qual os atributos selecionados na Seção 4.4 foram utilizados.

O conjunto de dados de entrada foram divididos em dois, conjunto de treinamento e conjunto de testes, esse processo foi realizado utilizando a função train_test_split da biblioteca Scikit-Learn. Assim os algoritmos foram submetidos ao conjunto de treinamento, gerando um modelo preditivo para posteriormente aplicar o conjunto de testes. A aplicação dos algoritmos foi realizada utilizando uma abordagem de aprendizagem supervisionada, com os classificadores precisando indicar se o vencedor da partida era o time mandante ou não. Os algoritmos foram parametrizados com seus valores padrões da biblioteca Scikit-Learn para fins de comparação.

Sendo assim o problema de classificação é considerado binário, classificando como 0 a derrota do time mandante, caso negativo, e 1 a vitória do mandante, caso positivo. Para a avaliação de desempenho dos algoritmos foram utilizadas as métricas de acuracidade, precisão, revocação e Medida F. Comparando os diferentes modelos nestas métricas para concluir, aquele de melhor desempenho para a proposta do trabalho. A Tabela 10, apresenta a acuracidade dos modelos.

Tabela 10 - Acuracidade dos modelos

Algoritmo	kNN	SVM	Random Forest	Gradient Tree Boosting	Logistic Regression	Naive Bayes	MLP
Acuracidade	61.35%	66.35%	64.05%	66.89%	67.94%	66.50%	68.04%

Fonte: do autor.

Pode ser afirmado que os algoritmos MLP, Logistic Regression, Gradient Tree Boosting e Naive Bayes obtiveram os melhores resultados, apresentando uma acuracidade próxima a 68%. Os algoritmos kNN e Random Forrest tiveram um desempenho inferior, chegando a ser 7% mais baixos que os algoritmos de melhor desempenho. O algoritmo kNN inclusive não atingiu um nível de acuracidade superior a simplesmente escolher o vencedor baseado na equipe com melhor campanha.

Para detalhar os desempenhos dos algoritmos de melhor acuracidade, foram utilizadas matrizes de confusão, conforme as tabelas 11, 12, 13, 14, nas quais podemos ver o número de instâncias classificadas corretamente por classe.

Tabela 11 - Matriz de confusão para algoritmo MLP

	Classe predita	
Classe atual	**0**	**1**
0	428	413
1	227	935

Fonte: do autor.

Tabela 12 - Matriz de confusão para algoritmo Logistic regression

	Classe predita	
Classe atual	**0**	**1**
0	436	405
1	237	925

Fonte: do autor.

Tabela 13 - Matriz de confusão para algoritmo Gradient tree boosting

	Classe predita	
Classe atual	**0**	**1**
0	457	384
1	279	883

Fonte: do autor.

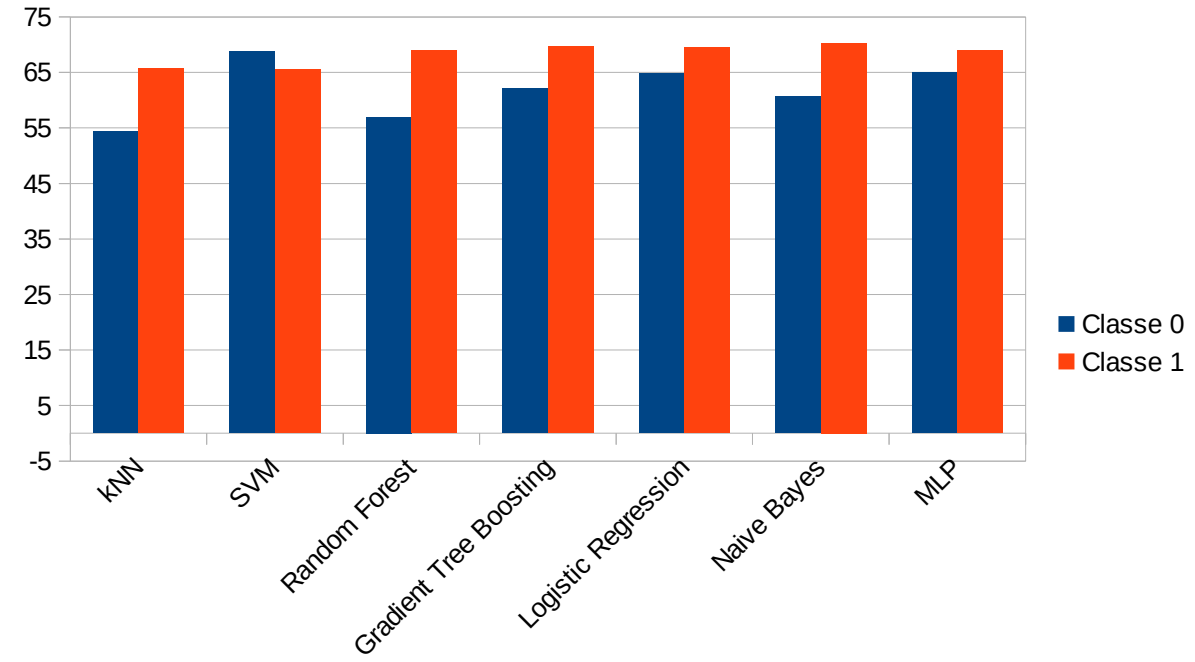
Tabela 14 - Matriz de confusão para algoritmo Naive Bayes

	Classe predita	
Classe atual	**0**	**1**
0	482	359
1	312	850

Fonte: do autor.

Conforme o Gráfico 6 exibe, é possível afirmar que os modelos possuem precisão superior para a classe 1, ou seja, exemplos de vitória da equipe mandante. Já a classe 0, qual representa a derrota da equipe mandante, os modelos apresentam acuracidade inferior, esses números refletem o que está presente no conjunto de dados, conforme a Tabela 6. O único algoritmo que obteve precisão superior na classificação da classe 0 foi o SVM.

Gráfico 6 - Precisão dos modelos por classe



Fonte: do autor.

Os algoritmos que obtiveram o melhor índice de precisão foram MLP e Logistic Regression, conforme a Tabela 15. Estes mesmos algoritmos apresentaram o melhor desempenho de acuracidade.

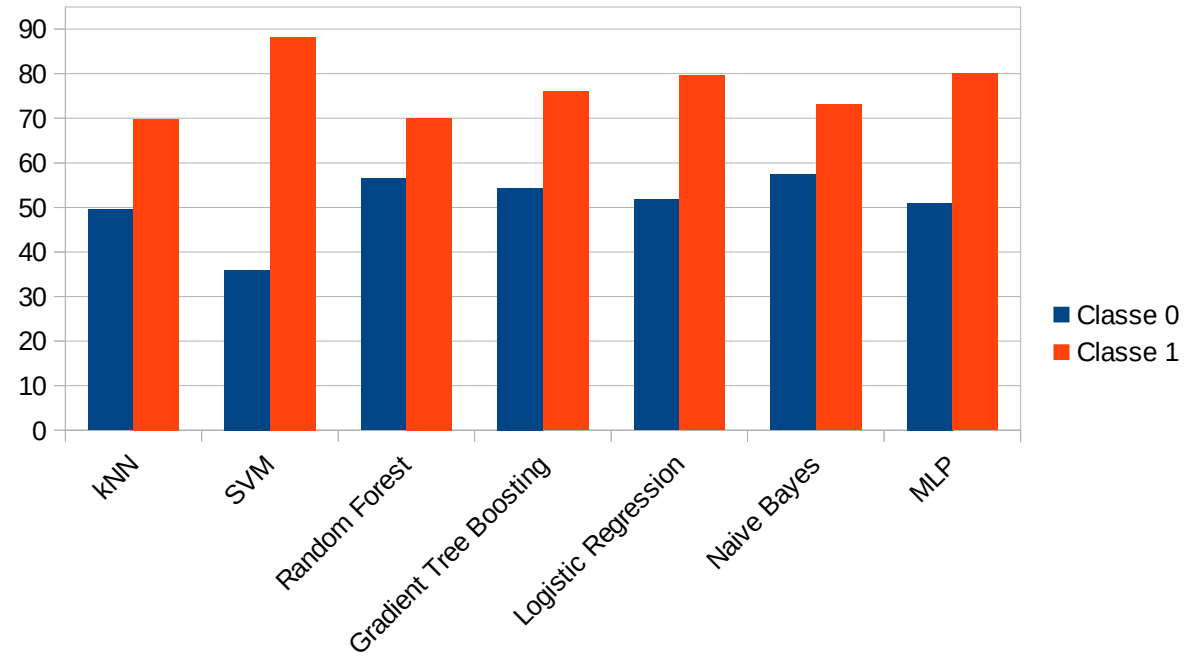
Tabela 15 - Precisão dos modelos

Algoritmo	kNN	SVM	Random Forest	Gradient Tree Boosting	Logistic Regression	Naive Bayes	MLP
Precisão	60.94%	66.90%	63.98%	66.50%	67.55%	66.27%	67.67%

Fonte: do autor.

A revocação dos modelos acompanha a mesma tendência da precisão, conforme Gráfico 7, tendo a classe 1, com resultados superiores em relação a classe 0. Porém a diferença destes resultados foi maior, todos os algoritmos apresentaram dificuldades em classificar a classe 0, inclusive os algoritmos que tiveram acuracidade alta como MLP e Logistic Regression. Ficou evidenciado que o algoritmo SVM teve uma alta revocação para a classe 1, porém teve um desempenho inversamente proporcional para a classe 0, apresentando o pior índice para esta classe.

Gráfico 7 - Revocação dos modelos



Fonte: do autor.

No índice de revocação geral os resultados seguem a tendência das avaliações da acuracidade e precisão, nas quais os algoritmos de MLP e Logistic Regression mostraram melhor desempenho, conforme a Tabela 16 apresenta.

Tabela 16 - Revocação dos modelos

Algoritmo	kNN	SVM	Random Forest	Gradient Tree Boosting	Logistic Regression	Naive Bayes	MLP
Revocação	61.36%	66.25%	64.05%	66.90%	67.95%	66.50%	68.05%

Fonte: do autor.

A avaliação da Medida F, conforme Tabela 17, apresenta os algoritmos MLP e Logistic Regression se sobressaindo sobre os demais, ambos atingindo um valor próximo a 0,7 na Medida F, considerando um bom desempenho, já que o máximo que pode ser atingido é 1,0.

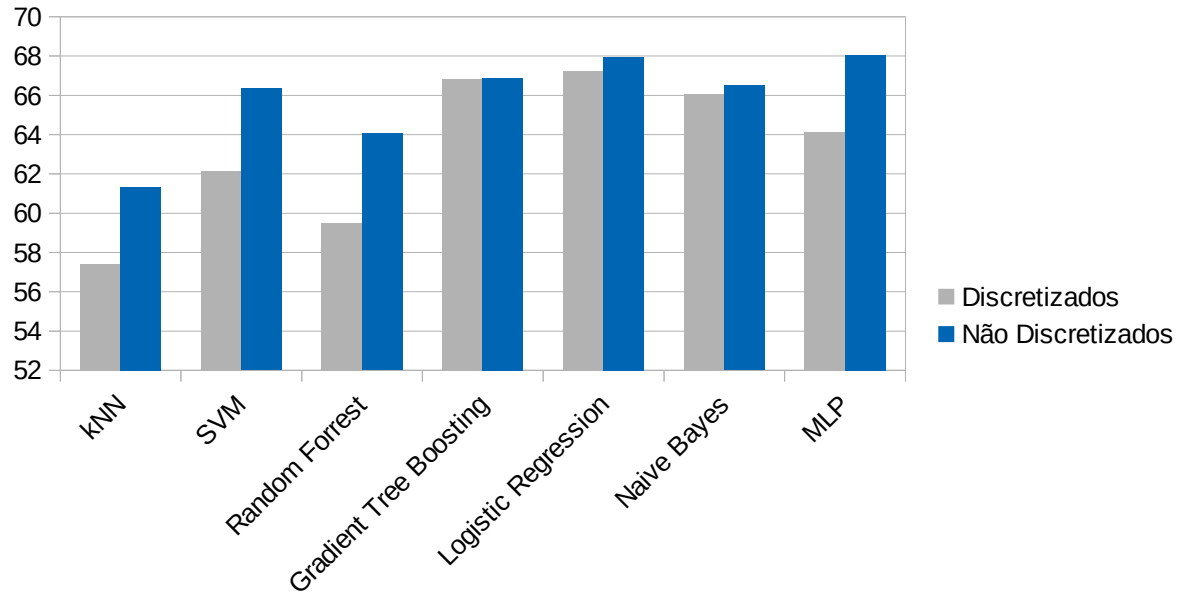
Tabela 17 - Medida F dos modelos

Algoritmo	kNN	SVM	Random Forest	Gradient Tree Boosting	Logistic Regression	Naive Bayes	MLP
Medida F	0.6107	0.6344	0.6402	0.6651	0.6725	0.6635	0.6725

Fonte: do autor.

Também foram realizados testes com o conjunto de dados discretizados, o processo de discretização foi feito utilizando a função Discretize do WEKA, sendo os dados divididos em 7 classes, estas classes foram categorizadas numericamente utilizando a função LabelEnconder da biblioteca Scikit-Learn. Conforme o Gráfico 8, os resultados de classificação com os dados discretizados não foram superiores a classificação com o conjunto de dados normais.

Gráfico 8 - Acuracidade dos modelos por dados discretizados e não discretizados



Fonte: do autor.

Em comparação com os trabalhos relacionados, os resultados obtidos neste trabalho foram muito semelhantes, em nenhum dos algoritmos comparadas a diferença ultrapassou dois pontos percentuais, conforme Tabela 18 apresenta. As principais diferenças em relação aos trabalhos, foram os atributos utilizados para a classificação e as diferentes temporadas utilizadas para treinamento e teste. Torres utilizou as temporadas de 2006 a 2012, enquanto Cao utilizou as temporadas entre 2005 a 2011.

Tabela 18 - Comparação de acuracidade entre trabalhos

Algoritmo	Cao (2012)	Torres(2013)	Este trabalho
SVM	67.70%	Não aplicado	66.35
MLP	68.01%	68.44%	68.04
Classificadores bayesianos	66.25%	66.81%	66.50
Regressão logística	69.67%	Não aplicado	67.94
kNN	Não aplicado	Não aplicado	61.35
Random Forest	Não aplicado	Não aplicado	64.05
Gradient Tree Boosting	Não aplicado	Não aplicado	66.89

Fonte: do autor.

No próximo capítulo estão presentes as considerações finais após o desenvolvimento do trabalho.

5 CONCLUSÕES

Com o estudo realizado e resultados obtidos, ficou comprovado que a aplicação de técnicas de aprendizado de máquina, podem ser utilizadas para a classificação de partidas da NBA. O uso de técnicas de classificação pode ser utilizado em benefício de apostadores e previsão de desempenho para equipes, dando a eles uma boa base de qual será o resultado da partida a ser disputada.

Foram aplicados diferentes algoritmos de classificação, em todos os casos foram aplicados o mesmo conjunto de dados e atributos. Os algoritmos que obtiveram os melhores resultados foram Logistic Regression e MLP, com acuracidade de 67.94% e 68.04% respectivamente. Os algoritmos mostraram-se mais eficientes em classificar os casos no qual o vencedor é o mandante da partida, isso pode ser pelo fato de no conjunto de dados a proporção de vencedores mandantes ser superior.

Os atributos mais influentes para a classificação do vencedor foram, o sistema de pontos de favoritismo e a campanha da equipe naquele momento da temporada. É constatado que os algoritmos não consideram atributos de estatísticas avançadas entre os mais influentes, conforme a Seção 2.7 mostra, diferente do que os autores acreditam, estas estatísticas não foram consideradas como os melhores avaliadores de bom desempenho das equipes.

No que diz respeito aos trabalhos relacionados que foram pesquisados, servindo como apoio para o desenvolvimento deste trabalho. Ficou constatado que mesmo

aplicando diferentes atributos para o treinamento dos modelos, os resultados finais com o objetivo de classificar o vencedor das partidas da NBA, pode ser alcançado com desempenho muito semelhante.

Em relação as ferramentas utilizadas para a aplicação das técnicas de mineração de dados, aprendizado de máquina e processos envolvidos, a biblioteca Scikit-Learn, mostrou-se apropriada e eficiente, a mesma possuí diversos recursos para os diferentes processos da mineração de dados, desde o pré-processamento de dados até a aplicação dos algoritmos de classificação e avaliação de seus respectivos resultados. A biblioteca é de fácil integração com outras bibliotecas de manipulação de dados, como Pandas e NumPy, e também para a exibição de gráficos e visualização de dados com a biblioteca matplotlib.

Com o objetivo de melhorar a performance dos modelos de previsão, são propostas algumas sugestões, com o objetivo de melhorar o desempenho nos casos de falsos verdadeiros.

5.1 Trabalhos futuros

Para obter melhores resultados na classificação de partidas da NBA, são propostas algumas sugestões como melhorias e projetos futuros:

- Integrar junto ao conjunto de dados das equipes as estatísticas de cada jogador da equipe, podendo assim avaliar o impacto de cada jogador, preparando assim os modelos para casos de lesões e suspensões;

- Treinar os modelos com um maior número de temporadas/jogos da NBA para aumentar o número de exemplos para o treinamento dos modelos;

- Explorar os demais atributos do conjunto de dados com auxílio de algum especialista da área do basquete.

REFERÊNCIAS

ARTERO, Almir Olivette. **Inteligência Artificial Teórica e Prática**. 1º Edição. São Paulo. Editoria Livraria da Física, 2008.

CAO, Chenjie. **Sports Data Mining Technology Used in Basketball Outcome Prediction**. Dublin Institute of Technology. 2012.

CARVALHO, André Carlos Ponce de Leon de; FACELI, Katti; LORENA, Ana Carolina; GAMA, João. **Inteligência Artificial: uma abordagem de aprendizado de máquina**. 1a Edição. Rio de Janeiro: Editora LTC, 2011.

CASTRO de, Leandro Nunes; FERRARI, Daniel Gomes. **Introdução à Mineração de Dados: conceitos básicos, algoritmos e aplicações**. 1a Edição. São Paulo. Saraiva, 2016.

COPPIN, Ben. **Inteligência Artificial**. 1º Edição. Rio de Janeiro. Livros Técnicos e Científicos Editora Ltda, 2013.

DATAGEEKS. **Moneyball e a Ciência de Dados**. Disponível em: <https://www.datageeks.com.br/moneyball-e-a-ciencia-de-dados/>. Acesso em 21 de abril de 2019.

GIL, Antonio Carlos. **Métodos e Técnicas de Pesquisa Social**. 6ª Edição. São Paulo. Editora Atlas S.A., 2008.

HAN, Jiawei; KAMBER, Micheline; PEI, Jian. **Data Mining Concepts and Techniques**. 3º Edition. Elsevier, 2012.

KELLY, Jason. Point Man. **Uchicago Magazine,** Chicago, set/out. 2013. Disponível em: <https://mag.uchicago.edu/economics-business/point-man>. Acesso em: 20 de abril de 2019.

KUBATKO, Justin; OLIVER, Dean; PELTON, Kevin; ROSENBAUM, Dan T. **A Starting Point for Analyzing Basketball Statistics**. Journal of Quantitative Analysis in Sports: Vol. 3, 2007.

MCKINNEY, Wes. **Python for Data Analysis**. 1ª edição. Sebastopol: O'Reilly Media, 2012.

MITCHELL, Tom M. **Machine Learning**. McGraw-Hill. Nova Iorque, 1997.

NBA. **NBA announce first betting-data parternship in U.S. with Sportradar, Genius Sports.** Disponível em: <https://www.nba.com/article/2018/11/28/nba-sportradar-genius-sports-partnership-official-release> Acesso em: 16 de abril de 2019.

NBA. **Stat Glossary**. Disponível em: <https://stats.nba.com/help/glossary/> Acesso em: 17 de abril de 2019.

OLIVER, Dean. **Basketball on Paper: Rules and Tools for Performance Analysis**. Potomac Books, 2004.

POSTGRESQL. **PostgreSQL 9.3.25 Documentation**. Disponível em: <https://www.postgresql.org/files/documentation/pdf/9.3/postgresql-9.3-US.pdf>. Acesso em 21 de março de 2019.

SPORTS INTERACTION. **NBA Player Stats and Leaders**. Disponível em: <https://news.sportsinteraction.com/nba/stats> Acesso em 20 de abril de 2019.

SPORTS REFERENCE LLC. **2017-2018 NBA Schedule and Results**. Disponível em: <https://www.basketball-reference.com/leagues/NBA_2018_games.html> Acesso em 16 de abril de 2019.

TAN, Pang – Ning; STEINBACH, Michael; KUMAR, Vipin. **Introdução ao Datamining: mineração de dados**. 1 a Edição. Rio de Janeiro: Editora Ciência Moderna, 2009.

TORRES, Renato Amorim. **Prediction of NBA games based on Machine Learning Methods**. University of Winsconsin Madison. 2013. Disponível em: <https://homepages.cae.wisc.edu/~ece539/fall13/project/AmorimTorres_rpt.pdf>. Aceso em 27 de Outubro de 2018.

WAINER, Jacques. **Métodos de pesquisas quantitativa e qualitativa para a Ciência da Computação**. 2007. Rio de Janeiro: Ed. PUC-Rio. IEEE, v17, nº 4.7.

WITTEN, Ian H.; FRANK, Eibe. **Data Mining Pratical Machine Learning Tools and Techniques**. 2ª Edition. Elsevier 2005.