



Universidade de Brasília - UnB
Instituto de Ciências Exatas - IE
Departamento de Estatística - EST

Modelos de Previsão para os Resultados da Temporada Regular de 2018/19 da NBA

Gustavo Pompeu da Silva

Orientador: Eduardo Monteiro de Castro Gomes

Brasília

2019

Gustavo Pompeu da Silva

Modelos de Previsão para os Resultados da Temporada Regular de 2018/19 da NBA

Relatório apresentado à disciplina de Trabalho de Conclusão de Curso II de graduação em Estatística, Departamento de Estatística, Instituto de Exatas, Universidade de Brasília, como parte dos requisitos necessários para obtenção do grau de Bacharel em Estatística.

Orientador: Eduardo Monteiro de Castro Gomes

Brasília

2019

Gustavo Pompeu da Silva

Modelos de Previsão para os Resultados da Temporada Regular de 2018/19 da NBA/ Gustavo Pompeu da Silva. – Brasília, 2019-
62 p. : il. (algumas color.) ; 30 cm.

Orientador: Eduardo Monteiro de Castro Gomes

Relatório Final – Universidade de Brasília

Instituto de Ciências Exatas

Departamento de Estatística

Trabalho de Conclusão de Curso de Graduação, 2019.

1. Modelagem. 2. Previsões. 3. NBA. 4. R. 5. Regressão. 6. Estatística.

Gustavo Pompeu da Silva

Modelos de Previsão para os Resultados da Temporada Regular de 2018/19 da NBA

Relatório apresentado à disciplina de Trabalho de Conclusão de Curso II de graduação em Estatística, Departamento de Estatística, Instituto de Exatas, Universidade de Brasília, como parte dos requisitos necessários para obtenção do grau de Bacharel em Estatística.

Eduardo Monteiro de Castro Gomes
Orientador

Leandro Tavares Correia
Membro da Banca

Donald Matthew Pianto
Membro da Banca

Brasília
2019

Resumo

Técnicas estatísticas são amplamente utilizadas para a previsão de eventos futuros nas mais diversas áreas. Neste trabalho, bancos de dados com estatísticas dos times da NBA foram criados para fazer modelagens com a aplicação de técnicas como Regressão Linear, Regressão Logística, Análise de Discriminante Linear, Máquina de Vetores de Suporte, entre outras, com o objetivo de obter previsões dos resultados dos jogos da temporada regular de 2018/19, e compará-los com os resultados reais e com as casas de aposta. Toda a implementação computacional foi feita em linguagem R.

Palavras-chave: Modelagem, Previsões, NBA, R, Regressão, Estatística.

Abstract

Statistical techniques are widely used to predict future events in several areas. In this paper, databases with stats from NBA teams were created to do modeling with the application of techniques such as Linear Regression, Logistic Regression, Support Vector Machines, among others, with the goal to get predictions for the match results of the 2018/19 regular season, and compare them to the real results and betting lines. The entire computational implementation was done in R language.

Keywords: Modeling, Predictions, NBA, R, Regression, Statistics.

Lista de tabelas

Tabela 1 – Funções e Pacotes utilizados no R para cada método	23
Tabela 2 – Exemplo de padrão de NA's	30
Tabela 3 – Médias de pontos marcados e sofridos por equipe	34
Tabela 4 – Porcentagem de Acerto das previsões dos jogos da temporada 2018/19 para cada método utilizando dados de 2000/01 a 2017/18 na modelagem	35
Tabela 5 – Porcentagem de acerto das previsões dos jogos da temporada 2018/19 para cada método utilizando temporadas diferentes na modelagem . . .	36
Tabela 6 – Porcentagem de Acerto das previsões dos jogos da temporada 2018/19 para cada método utilizando dados de 2006/07 a 2017/18 na modelagem	36
Tabela 7 – Tempo de execução do código computacional para cada método	38
Tabela 8 – Comparação das vitórias reais com as vitórias previstas - Conferência Leste	39
Tabela 9 – Comparação das vitórias reais com as vitórias previstas - Conferência Oeste	39
Tabela 10 – Acertos das previsões por time por local e resultado do jogo - Parte 1 .	40
Tabela 11 – Acertos das previsões por time por local e resultado do jogo - Parte 2 .	41
Tabela 12 – Variáveis mais significativas no modelo	43
Tabela 13 – Resumo das diferenças absolutas das Previsões da Regressão Linear vs. linhas de aposta vs. resultados reais	46

Sumário

1	INTRODUÇÃO	13
2	REVISÃO DE LITERATURA	15
2.1	Regressão Linear	15
2.2	Regressão Logística	16
2.3	Regressão de Probit	17
2.4	Seleção de Variáveis	17
2.5	Máquina de Vetores de Suporte (SVM)	18
2.6	Análise de Discriminante Linear	19
2.7	Árvores de Regressão e Classificação	20
2.8	<i>Random Forest</i>	21
3	MATERIAL E MÉTODOS	23
3.1	Implementação Computacional dos Métodos	23
3.2	NBA	23
3.3	Criação das Bases de Dados	24
3.3.1	Lidando com valores faltantes	29
3.4	Casas de Aposta	30
4	RESULTADOS	33
4.1	Resultados Reais da Temporada 2018/19	33
4.2	Previsões	34
4.2.1	“Previsões” das casas de aposta	37
4.3	Tempo de execução computacional de cada método	37
4.4	Modelo “campeão”	38
4.4.1	Comparação dos resultados reais com as previsões	38
4.4.1.1	Acertos por Equipe	40
4.4.2	Variáveis mais significativas	42
4.4.3	Comparação do modelo “campeão” com as casas de aposta	43
4.4.4	Adicionando jogos de 2018/19 na modelagem	45
4.5	Comparação do modelo de regressão linear com as “linhas” de aposta e com os resultados reais	45
5	CONCLUSÃO	47
	REFERÊNCIAS	49

APÊNDICE A – CÓDIGOS EM R	53
-------------------------------------	----

1 Introdução

Mineração de dados em esportes é um tópico que tem crescido rapidamente nos últimos anos. Jogadores de ligas de *fantasy*, apostadores e entusiastas de esportes possuem grande interesse em procurar vantagens nas apostas e previsões através de dados e números. Ferramentas e técnicas começaram a ser desenvolvidas para medir o desempenho tanto de times quanto de atletas, e esses métodos vem chamando a atenção cada vez mais.

Existe uma imensa quantidade de dados disponíveis sobre qualquer esporte. Esses dados podem ser de desempenho individual de jogadores ou da equipe, decisões da comissão técnica, eventos que acontecem nos jogos, entre outros. É preciso saber não só como coletar esses dados, mas também quais podem ser úteis e como fazer o melhor uso possível deles. Achando os meios para transformar esses dados em conhecimento, organizações esportivas tem o potencial de obter uma vantagem competitiva sobre seus oponentes. Não se deve analisar performance no sentido de ganhar ou perder, ou de marcar mais gols ou pontos do que o oponente, pois esse é o objetivo geral de qualquer esporte. O que mais interessa é encontrar padrões em outras estatísticas que mostram tendências justamente para que as vitórias sejam obtidas.

Data Mining envolve procedimentos para descobrir padrões escondidos e descobrir novas informações a partir de fontes de dados. A fundação científica de data mining pode ser dividida em três disciplinas: estatísticas, inteligência artificial e *machine learning*. *Data mining* então pode ser definido como a busca de conhecimento dentro dos dados. (SCHUMAKER; SOLIEMAN; CHEN, 2010)

O objetivo geral desse trabalho é ajustar modelos utilizando diversas técnicas estatísticas para obter previsões para os resultados dos jogos da temporada regular de 2018/19 da NBA e compará-las a fim de chegar em uma conclusão sobre qual técnica funcionou melhor, julgando principalmente pela acurácia das previsões.

Este trabalho está organizado de forma que no Capítulo 2 será abordada a revisão de literatura com resumos teóricos dos métodos estatísticos que serão aplicados, no Capítulo 3 a metodologia e a parte computacional, no Capítulo 4 os resultados obtidos, e por fim a conclusão.

2 Revisão de Literatura

Os métodos a serem descritos nesse capítulo serão utilizados para as modelagens com o objetivo de fazer previsões para os jogos da temporada 2018/19 da NBA.

Serão consideradas duas abordagens para a variável dependente: uma quantitativa, que é o saldo de pontos entre os times (subtrair a pontuação de um time pela do outro), de onde se infere qual time vence o jogo pelo sinal do saldo, e a outra qualitativa, que é uma variável dicotômica indicando somente se o time venceu ou perdeu o jogo.

Considerando essas duas abordagens, diferentes métodos serão utilizados, para que os dois tipos de variáveis possam ser aplicadas em vários modelos.

Alguns dos modelos descritos possuem suposições para serem aplicados, no entanto, como o objetivo específico é obter previsões para os resultados dos jogos, não é estritamente necessário verificar as suposições em questão.

As técnicas estatísticas utilizadas para a obtenção das previsões dos jogos são:

- Regressão Linear
- Regressão Logística
- Regressão de Probit
- Máquina de Vetores de Suporte (SVM)
- Análise de Discriminante Linear
- Árvores de Regressão
- Árvores de Classificação
- *Random Forest*

2.1 Regressão Linear

A Regressão linear é um método estatístico que compõe uma equação para se descrever o valor esperado de uma variável Y (resposta), dado os valores de outras variáveis X (explicativas). É linear pois considera que a relação da variável resposta com as variáveis explicativas é uma função linear dependente de alguns parâmetros. A equação que determina a relação entre as variáveis é:

$$Y = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \dots + \beta_p X_p + \epsilon$$

Em que Y é a variável resposta (dependente) e p é o número de variáveis explicativas. As constantes β_j são denominados coeficientes de regressão, X_j são as variáveis explicativas (independentes), com $j = 0, 1, \dots, p$ e ϵ representa o erro experimental. O parâmetro β_0 corresponde ao intercepto, e fornece a resposta média de Y quando $X_1 = X_2 = \dots = X_p = 0$. Para $j \geq 1$, os parâmetros β_j indicam uma mudança na resposta média de Y a cada unidade de mudança na variável X_j , enquanto as demais variáveis são mantidas fixas.

As suposições clássicas necessárias para o Modelo de Regressão Linear Múltipla são (YAN; SU, 2009):

- Os erros não devem ser correlacionados, devem seguir distribuição normal e ter média zero e variância σ^2 , desconhecida. Ou seja, $\epsilon_i \stackrel{iid}{\sim} N(0, \sigma^2)$;
- Deve existir uma relação linear entre a variável dependente e as variáveis independentes;

Um outro fator importante a se observar é que não deve existir multicolinearidade entre as variáveis independentes, ou seja, elas não devem ter correlação alta entre si.

Nesse trabalho, as suposições da regressão linear não serão verificadas, especialmente porque as observações não são independentes umas das outras, pois os jogos são uma sequência histórica no tempo.

2.2 Regressão Logística

A regressão logística se difere da linear essencialmente pelo fato da variável resposta ser binária, ou seja, Y tem distribuição Bernoulli $(1, \pi)$, com probabilidade de sucesso $P(Y_i = 1) = \pi_i$ e de fracasso $P(Y_i = 0) = 1 - \pi_i$.

Matematicamente, a regressão logística estima uma função de regressão linear múltipla definida por:

$$\text{logit}(\pi) = \log\left(\frac{\pi}{1 - \pi}\right) = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \dots + \beta_p X_p \quad (2.1)$$

Onde $\pi = P(Y = 1)$. Baseado em 2.1, é obtida a equação:

$$\pi = \frac{\exp(\beta_0 + \beta_1 X_1 + \beta_2 X_2 + \dots + \beta_p X_p)}{1 + \exp(\beta_0 + \beta_1 X_1 + \beta_2 X_2 + \dots + \beta_p X_p)} = \frac{e^{\beta_0 + \beta_1 X_1 + \beta_2 X_2 + \dots + \beta_p X_p}}{1 + e^{\beta_0 + \beta_1 X_1 + \beta_2 X_2 + \dots + \beta_p X_p}}$$

As suposições necessárias para o Modelo de Regressão Logística são (KASSAMBARA, 2018):

- A variável dependente é binária (dicotômica);
- Não existir multicolinearidade entre as variáveis independentes, ou seja, elas não podem ter correlação alta entre si;
- Existir linearidade entre as variáveis independentes e a função *logit*;
- Não ocorrer valores extremos nas variáveis independentes contínuas.

Novamente, não haverá uma preocupação em verificar as suposições para o modelo.

2.3 Regressão de Probit

A Análise de Probit ou Regressão de Probit (CARVALHO et al., 2017) é outro tipo de regressão binária, parecida com a regressão logística, a diferença é a função de ligação utilizada. O *link* probit é dado por:

$$\text{probit}(\pi) = \Phi^{-1}(\pi) = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \dots + \beta_p X_p \quad (2.2)$$

Onde $\pi = P(Y = 1)$. Baseado em 2.2, é obtida a equação:

$$\pi = \Phi(\beta_0 + \beta_1 X_1 + \beta_2 X_2 + \dots + \beta_p X_p)$$

Em que Φ é a Função de Distribuição Acumulada (f.d.a.) da distribuição Normal Padrão.

As suposições necessárias são as mesmas da Regressão Logística, pois são modelos lineares generalizados da mesma família (binomial). Nesse trabalho, as suposições não serão verificadas.

2.4 Seleção de Variáveis

Para os métodos de regressão citados acima, é possível aplicar um método de seleção de variáveis (SUÁREZ et al., 2017), em que modelos são analisados com base em alguma medida, e variáveis são retiradas/adicionadas para que seja obtido o modelo com o melhor valor da medida analisada. A função *step*, que faz parte do R, utiliza a medida AIC para escolher as variáveis.

Existem três formas de aplicar a função *step*:

- *forward*, em que o modelo começa sem nenhuma variável explicativa, e a cada passo adiciona a variável que mais reduziria o valor do AIC do modelo, até que o modelo com o menor AIC seja o atual, sem adicionar mais nenhuma variável.

- *backward*, em que o modelo inicial é composto por todas as variáveis explicativas, e a cada passo retira a variável que resultaria em um modelo com o menor valor de AIC, até que o modelo atual tenha o menor AIC.
- *stepwise*, é uma mistura dos dois métodos acima, em cada passo é aplicada uma iteração do *forward* e uma do *backward*. Por exemplo, em um passo, adiciona-se a variável que diminuiria mais o valor do AIC, e logo após, verifica se retirar alguma variável que já estava no modelo não diminuiria o AIC. O processo termina apenas quando nem adicionar, nem retirar nenhuma variável diminuiria o valor do AIC.

Nesse trabalho será aplicado o método *forward*, pois esse método é o mais rápido de ser aplicado computacionalmente, além disso, os métodos geralmente obtêm resultados similares.

2.5 Máquina de Vetores de Suporte (SVM)

As Máquinas de Vetores de Suporte (*Support Vector Machines* - SVMs) constituem uma técnica embasada na Teoria de Aprendizado Estatístico (VAPNIK, 1995), em que seu objetivo é reconhecer padrões nos dados.

A teoria da SVM é complexa, mas sua abordagem pode ser esboçada da seguinte forma:

- **Separação das Classes:** Para classificar dados em duas classes diferentes, tenta-se obter um plano que separe as classes no espaço p -dimensional. Esse plano é chamado de hiperplano. O objetivo é determinar o hiperplano ótimo, e isso é feito através da maximização das “margens” entre os pontos mais próximos das classes (ver Figura 1), os pontos em cima das fronteiras são chamados de vetores de suporte, e o plano no meio das margens é o hiperplano ótimo de separação;
- **Classes sobrepostas:** Observações do lado “errado” da margem discriminante são ponderadas para reduzir suas influências;
- **Não-linearidade:** Quando um separador linear não é encontrado, as observações são projetadas em um espaço de maior dimensão, onde elas se tornam efetivamente linearmente separáveis (essa projeção é feita via técnicas Kernel);
- **Solução de Problemas:** A tarefa toda pode ser formulada como um problema de otimização quadrática que pode ser resolvida por técnicas conhecidas.

Um programa capaz de realizar todas essas tarefas é chamado de uma Máquina de Vetores de Suporte. (MEYER, 2019)

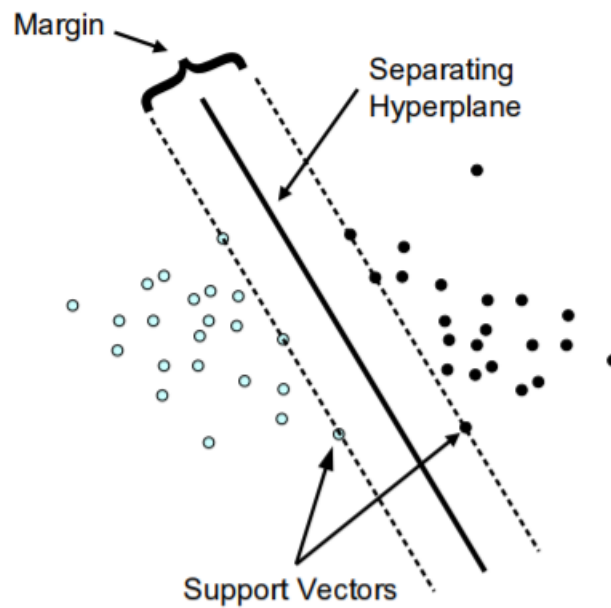


Figura 1 – Classificação (caso de separação linear)

Para a implementação computacional, será utilizada a função *svm* do pacote *e1071* da linguagem R (MEYER et al., 2018), que possui 4 opções de função Kernel: linear, base radial (gaussiana), polinomial e sigmoidal.

A função Kernel a ser utilizada nesse trabalho será a base radial, pois foi a que obteve os melhores resultados para as previsões em testes preliminares. Ela contém um parâmetro específico γ que tem como valor padrão $\frac{1}{\text{dim}}$, onde *dim* representa a dimensão dos dados.

Outro parâmetro a ser utilizado será o custo, que é um parâmetro geral de penalização para esse tipo de classificação. Seu valor padrão é 1.

Serão feitas modelagens com os valores padrão desses dois parâmetros, mas também serão encontrados os melhores valores para eles para os dados utilizados, através da função *tune*, do mesmo pacote da função *svm*.

2.6 Análise de Discriminante Linear

A Análise de Discriminante é uma técnica multivariada que tem como finalidade separar observações em grupos e alocar novas observações em algum dos grupos pré-definidos. A Análise de Discriminante é exploratória por natureza. O objetivo dessa técnica é descrever algebricamente as características diferenciais das observações. São encontrados “discriminantes” cujos valores numéricos são tais que as populações são separadas o melhor possível. (JOHNSON; WICHERN, 2007)

A Análise de Discriminante Linear é uma generalização da Discriminante Linear de Fisher. Para duas classes, a alocação de novas observações funciona de uma maneira muito simples. Primeiro, é feita uma matriz de variância-covariância estimada para os dados (\mathbf{S}_p^2):

$$\mathbf{S}_p^2 = \frac{\sum_{j=1}^{n_1} (\mathbf{x}_{1j} - \bar{\mathbf{x}}_1)^2 + \sum_{j=1}^{n_2} (\mathbf{x}_{2j} - \bar{\mathbf{x}}_2)^2}{n_1 + n_2 - 2}$$

Onde n_1 e n_2 correspondem ao número de observações da população 1 (π_1) e ao número de observações da população 2 (π_2), respectivamente. $\bar{\mathbf{x}}_1$ e $\bar{\mathbf{x}}_2$ correspondem às médias das variáveis independentes para cada população e \mathbf{x}_{1j} e \mathbf{x}_{2j} são referentes à cada observação j de cada população.

Então, para uma nova observação \mathbf{x}_0 , temos: $\hat{y}_0 = (\bar{\mathbf{x}}_1 - \bar{\mathbf{x}}_2)' \mathbf{S}_p^{-1} \mathbf{x}_0$ e $\hat{m} = \frac{1}{2}(\bar{\mathbf{x}}_1 - \bar{\mathbf{x}}_2)' \mathbf{S}_p^{-1} (\bar{\mathbf{x}}_1 + \bar{\mathbf{x}}_2)$, e a regra de alocação será que x_0 pertencerá à população π_1 se $\hat{y}_0 - \hat{m} \geq 0$, e x_0 pertencerá à população π_2 caso contrário.

As suposições necessárias para a Análise de Discriminante Linear são:

- Normalidade multivariada dos dados;
- Matriz de variância-covariância das populações devem ser iguais.

2.7 Árvores de Regressão e Classificação

As árvores de regressão e classificação são métodos estatísticos não-paramétricos utilizados baseado na teoria de árvores de decisão (ver Figura 2). Os nós terminais da árvore são resultados numéricos (caso a variável resposta seja quantitativa), ou classes (caso a variável resposta seja qualitativa). No primeiro caso, chamamos o método de Árvore de Regressão, e no segundo de Árvore de Classificação.

As árvores são construídas a partir de um particionamento recursivo binário usando a variável resposta e escolhendo divisões das variáveis explicativas. (RIPLEY, 1996)

O processo é denominado recursivo pois cada subpopulação criada pode ser dividida por um número indefinido de vezes até que o processo de divisão termine após um determinado critério de parada ser atingido.

Algoritmos para construção dessas árvores geralmente trabalham de cima para baixo, escolhendo em cada etapa uma variável que melhor divide o conjunto de observações. Algoritmos diferentes usam métricas diferentes para definir essa “melhor” divisão. Geralmente, é medida a homogeneidade da variável alvo dentro dos subconjuntos.

São alternativas não-paramétricas à regressão linear e à regressão logística, e não necessitam de pressupostos para serem aplicadas.

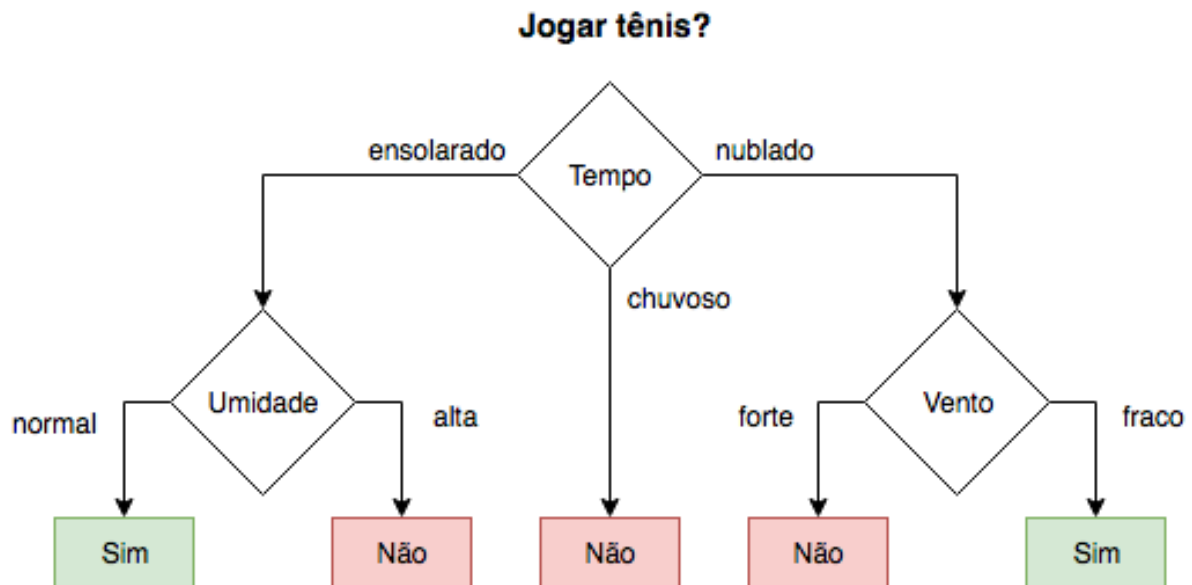


Figura 2 – Exemplo de Árvore de Decisão para jogar tênis ou não

2.8 Random Forest

Random forests são uma combinação de preditores de árvores (vistos na seção 2.7) tal que cada árvore depende dos valores de um vetor aleatório amostrado de forma independente e com a mesma distribuição para todas as árvores na “floresta”. O erro de generalização de uma floresta de árvores de classificação depende da força das árvores individuais na floresta e da correlação entre elas e converge quase certamente para um limite conforme o número de árvores na floresta se torna grande. Uma seleção aleatória de observações e de variáveis é utilizada para a criação de cada árvore de classificação. Estimativas internas monitoram o erro, a força e a correlação e são usadas para mostrar a resposta ao aumento do número de variáveis usados na separação. Estimativas internas também são usadas para medir a importância das variáveis. Todas essas ideias também são aplicáveis para regressão. (BREIMAN, 2001)

Uma vantagem desse método é a prevenção de *overfitting*, que é quando um modelo se ajusta muito bem ao conjunto de dados anteriormente observado, mas se mostra ineficaz para prever novos resultados. Em compensação é um método muito mais lento de ser computado do que uma árvore de classificação ou regressão, pois são construídas muitas árvores em vez de uma só.

3 Material e Métodos

A linguagem R (R Core Team, 2018) será a única utilizada nesse trabalho.

3.1 Implementação Computacional dos Métodos

Cada método estatístico mencionado no Capítulo 2 tem uma função em um pacote da linguagem R para sua implementação computacional, as utilizadas nesse trabalho estão representadas na Tabela 1.

Tabela 1 – Funções e Pacotes utilizados no R para cada método

Método	Função	Pacote
Regressão Linear	<i>lm</i>	<i>stats</i> (R Core Team, 2018)
Regressão Logística	<i>glm</i>	<i>stats</i>
Regressão de Probit	<i>glm</i>	<i>stats</i>
Seleção de Variáveis para as Regressões	<i>step</i>	<i>stats</i>
SVM	<i>svm</i>	<i>e1071</i> (MEYER et al., 2018)
Análise de Discriminante Linear	<i>lda</i>	<i>MASS</i> (VENABLES; RIPLEY, 2002)
Árvores de Regressão e Classificação	<i>tree</i>	<i>tree</i> (RIPLEY, 2019)
<i>Random Forest</i>	<i>randomForest</i>	<i>randomForest</i> (LIAW; WIENER, 2002)

3.2 NBA

A NBA (*National Basketball Association*) é a principal liga de basquete profissional do mundo. Atualmente, é composta por 30 times baseados em cidades da América do Norte (29 nos Estados Unidos e 1 no Canadá), divididos em 2 conferências: Leste e Oeste. É a liga onde jogam os melhores atletas de basquete do mundo, e com os maiores salários do esporte. Uma das vantagens de trabalhar com o basquete e a NBA especificamente é a grande quantidade de observações, pois, atualmente, em uma temporada regular, cada time joga 82 vezes, ou seja, são 1230 jogos por temporada. Os 8 times mais bem classificados de cada conferência se classificam para os *playoffs* para disputar o título do campeonato. Nesse trabalho o foco será apenas na temporada regular da NBA, pois os *playoffs* possuem uma dinâmica completamente diferente.



Figura 3 – Logos dos times atuais da NBA
Fonte: (SPORTSLOGOS, 2019)

3.3 Criação das Bases de Dados

Para a obtenção dos dados necessários, será utilizada uma técnica de *web scraping*, na qual informações são extraídas de alguma página da internet. Com o auxílio do pacote *rvest* (WICKHAM, 2016), serão extraídas informações do site Basketball Reference (BASKETBALL-REFERENCE, 2019), um dos maiores sites com dados numéricos sobre a NBA e basquete em geral.

Toda página da internet possui um código HTML por trás, e existe uma extensão do *Google Chrome*, chamada *SelectorGadget* (SELECTORGADGET, 2019), que permite ao usuário clicar nas áreas do site para selecionar as partes que se deseja extrair a partir de seu código HTML, mesmo que o usuário não tenha conhecimento de programação em HTML, e combinando isso com funções do pacote *rvest*, pode-se transformar esse código

em texto no R.

Foi escolhido fazer a extração dos dados de resultados de jogos a partir da temporada de 2000/01 até a de 2017/18. Com isso, são 18 temporadas para realizar as modelagens e serem feitas as previsões dos jogos da temporada 2018/19.

O número de jogos por temporada regular varia por alguns motivos. Nas temporadas de 2000/01 a 2004/05, a NBA era composta por apenas 29 times, com cada time jogando 82 jogos, o que resultava em 1189 jogos por temporada regular. Nas demais temporadas utilizadas, 30 times faziam parte da liga, resultando em 1230 jogos por temporada. A única exceção foi a temporada de 2011/12, quando aconteceu um *lockout*, os donos das equipes se recusaram a deixar os jogos acontecerem, pois o contrato da NBA com os times acabou antes do início da temporada e a NBA demorou para chegar em um acordo com os donos dos times para assinarem um novo contrato. Um novo acordo foi estabelecido depois de vários meses de negociação e a temporada começou em 25 de dezembro de 2011, com quase 2 meses de atraso. Isso diminuiu o número de jogos realizados por cada equipe de 82 para 66 jogos, que resultou em um total de apenas 990 jogos na temporada regular.

As informações obtidas de cada um dos jogos realizados das temporadas citadas são: data do jogo, nome do time visitante, pontos marcados pelo time visitante, nome do time mandante, pontos marcados pelo time mandante, se houve prorrogação no jogo, e o público presente no ginásio.

A partir do que foi obtido, podemos criar uma base de dados com muitas variáveis derivadas dessas informações e criar modelos para realizar as previsões utilizando as diversas técnicas estatísticas citadas anteriormente.

É importante ressaltar que a temporada regular de 2018/19 estava em andamento durante a realização desse trabalho, com duração de Outubro de 2018 até Abril de 2019. As informações relacionadas aos jogos dessa temporada foram extraídas gradualmente conforme os jogos aconteciam ao longo do tempo. Até o final do trabalho, a temporada regular já havia sido concluída.

A base de dados inicial, criada a partir das informações extraídas da internet, contém 2 linhas para cada jogo realizado. Cada linha contém informações referentes a um dos times envolvidos na partida, essas informações são:

Variáveis de identificação das informações do jogo:

- *Team*: Nome do time
- *Opp*: Nome do time adversário
- *Pts_S*: Pontos marcados pelo time nesse jogo.
- *Pts_A*: Pontos marcados pelo time adversário nesse jogo.

- *Home*: Se o time jogou em casa ou não.
- *Attend*: Público presente no ginásio nesse jogo. (Tem a informação apenas se o time jogou em casa)
- *OT*: Indica se ocorreu prorrogação no jogo.

Essas variáveis acima não serão utilizadas nas modelagens, elas são apenas as variáveis extraídas da internet, e terão suas informações repassadas para outras variáveis que serão citadas a seguir.

Variáveis indicadoras do resultado do jogo:

- *Win*: Se o time venceu esse jogo ou não (qualitativa, dicotômica).
- *result*: Saldo de pontos, ou seja, os pontos marcados pelo time menos os pontos marcados pelo seu adversário (quantitativa).

Variáveis de informação sobre o jogo que podem ser identificadas antes da realização da partida:

- *weekday*: Dia da semana em que o jogo foi/será realizado. Essa variável pode ser utilizada nas modelagens, pois sabemos o dia da semana que o jogo ocorrerá mesmo antes do jogo acontecer.
- *Travel*: Variável que indica se o time teve/terá que viajar da partida anterior para essa ou não. Por exemplo, se o jogo anterior foi fora de casa, o time sempre tem que viajar, ou pra voltar pra casa, ou pra ir para outra cidade fora de casa. O time só não viaja quando joga 2 jogos seguidos em casa.

Variáveis referentes à toda informação do time desde o início da temporada até antes do jogo:

- *Games_T*: Total de jogos do time até agora na temporada.
- *Games_H*: Jogos em casa do time até agora na temporada.
- *Games_A*: Jogos fora de casa do time até agora na temporada.
- *Wins_T*: Total de vitórias do time até agora na temporada.
- *Wins_H*: Vitórias em casa do time até agora na temporada.
- *Wins_A*: Vitórias fora de casa do time até agora na temporada.
- *Loss_T*: Total de derrotas do time até agora na temporada.

- *Loss_H*: Derrotas em casa do time até agora na temporada.
- *Loss_A*: Derrotas fora de casa do time até agora na temporada.
- *Streak_T*: Número indicando a sequência de vitórias (positivo) ou derrotas (negativo) do time, considerando jogos em casa e fora de casa. Exemplos: os 5 últimos jogos do time foram vitórias, e o antes desses 5 foi derrota, logo a variável vale +5. O último jogo do time foi derrota, e o penúltimo vitória, então a variável vale -1.
- *Streak_H*: Número indicando a sequência de vitórias (positivo) ou derrotas (negativo) do time, considerando apenas jogos em casa.
- *Streak_A*: Número indicando a sequência de vitórias (positivo) ou derrotas (negativo) do time, considerando apenas jogos fora de casa.
- *Mean_Pts_S_H*, *Max_Pts_S_H*, *Min_Pts_S_H*: Média, máximo e mínimo de pontos marcados do time em jogos em casa até o momento na temporada.
- *Mean_Pts_S_A*, *Max_Pts_S_A*, *Min_Pts_S_A*: Média, máximo e mínimo de pontos marcados do time em jogos fora de casa até o momento na temporada.
- *Mean_Pts_S_T*: Média de pontos marcados do time em todos os jogos até o momento na temporada.
- *Mean_Pts_A_H*, *Max_Pts_A_H*, *Min_Pts_A_H*: Média, máximo e mínimo de pontos sofridos do time em jogos em casa até o momento na temporada.
- *Mean_Pts_A_A*, *Max_Pts_A_A*, *Min_Pts_A_A*: Média, máximo e mínimo de pontos sofridos do time em jogos fora de casa até o momento na temporada.
- *Mean_Pts_A_T*: Média de pontos sofridos do time em todos os jogos até o momento na temporada.
- *Str_Sch*: A “força de calendário” do time até o momento na temporada, ou seja, a proporção de vitórias dos adversários que o time enfrentou até o momento na temporada. Divide-se o total de vitórias de todos os adversários do time pelo total de jogos de todos os adversários do time.
- *mean_attend*: Média de público do time nos jogos em casa, até o momento na temporada.

Variáveis referentes aos últimos 3, 5, 7 ou 10 jogos do time na temporada:

- *Mean_Last_X_A*, *Max_Last_X_A*, *Min_Last_X_A*: Média, máximo e mínimo de pontos marcados do time nos últimos X jogos fora de casa, onde $X = 3, 5, 7, 10$.

- *Mean_Last_X_H*, *Max_Last_X_H*, *Min_Last_X_H*: Média, máximo e mínimo de pontos marcados do time nos últimos X jogos em casa, onde $X = 3, 5, 7, 10$.
- *Mean_Last_X_T*, *Max_Last_X_T*, *Min_Last_X_T*: Média, máximo e mínimo de pontos marcados do time nos últimos X jogos, onde $X = 3, 5, 7, 10$.
- *Mean_Last_X_A_Opp*, *Max_Last_X_A_Opp*, *Min_Last_X_A_Opp*: Média, máximo e mínimo de pontos sofridos do time nos últimos X jogos fora de casa, onde $X = 3, 5, 7, 10$.
- *Mean_Last_X_H_Opp*, *Max_Last_X_H_Opp*, *Min_Last_X_H_Opp*: Média, máximo e mínimo de pontos sofridos do time nos últimos X jogos em casa, onde $X = 3, 5, 7, 10$.
- *Mean_Last_X_T_Opp*, *Max_Last_X_T_Opp*, *Min_Last_X_T_Opp*: Média, máximo e mínimo de pontos sofridos do time nos últimos X jogos, onde $X = 3, 5, 7, 10$.
- *Win_Last_X_A*, *Win_Last_X_H*, *Win_Last_X_T*: Número de vitórias do time nos últimos X jogos fora de casa, em casa, e total, respectivamente, onde $X = 3, 5, 7, 10$.

Variáveis referentes apenas ao último jogo realizado pelo time:

- *OT_Last*: Indica se houve prorrogação no jogo anterior do time.
- *Days_LG*: Quantos dias atrás foi o último jogo do time.

O resultado desse conjunto de informações extraídas foi um banco d

Isso resulta em um banco de dados referente a cada temporada, com 2 linhas para cada jogo realizado, e 125 variáveis.

A variável dependente nas modelagens será *Win* (dicotômica, qualitativa) ou *result* (quantitativa), dependendo da técnica utilizada.

Como o objetivo é realizar uma previsão para cada jogo, as duas linhas de cada jogo serão combinadas em uma só, ou seja, uma primeira parte do banco de dados final terá apenas variáveis referentes ao time visitante de cada jogo, e a segunda parte apenas variáveis referentes ao time mandante. Como existe essa separação clara entre as variáveis, é fácil remover as duplicatas (como o dia da semana do jogo e as variáveis dependentes) e as que não teriam propósito (como a média de público do time visitante, que não é aplicável, e a variável que indica se o time viajou do último jogo para o atual para o time visitante, pois ela sempre será *TRUE*). Além disso, também serão retiradas as variáveis referentes a jogos fora de casa para os times mandantes e as referentes a jogos em casa para os times visitantes.

Após a remoção dessas variáveis e das variáveis de identificação, o resultado é uma base de dados final por temporada, com uma linha por jogo e 151 variáveis, duas delas sendo as variáveis dependentes. As variáveis dependentes mantidas foram as referentes ao time visitante, ou seja, a variável *Win* se tornou *Win_Vis* e tem valor *TRUE* quando o time visitante vence, e *FALSE* caso contrário. A variável *result* se tornou *result_Vis*, ou seja, como é o saldo de pontos, a variável é positiva quando o time visitante vence e negativa caso contrário. De forma similar, para todas as variáveis restantes na base de dados, foi adicionada a extensão *_Vis* no nome das que são referentes ao time visitante e a extensão *_Home* no nome das que são referentes ao time mandante. A única variável que não é referente a nenhum dos dois é a do dia da semana do jogo (*weekday*) e o nome dela foi mantido.

3.3.1 Lidando com valores faltantes

Como existem muitas variáveis que dependem de resultados anteriores, existirão vários valores faltantes no começo de cada temporada, que nesse trabalho serão referenciados como NA (não aplicável). Por exemplo, se o time só realizou 6 jogos na temporada, não é possível obter um valor para a média de pontos marcados nos últimos 7 jogos, ou se é o primeiro jogo do time na temporada, não há como obter nenhuma informação além do dia da semana do jogo.

Na base de dados da temporada 2018/19, que é a que será feita as previsões, das 1230 observações da temporada regular, apenas 875 possuem informação completa, ou seja, nenhum NA em nenhuma variável. Nas outras 355 observações, existe pelo menos um NA em alguma variável.

No R, quando é feita uma modelagem, geralmente, as observações que possuem algum valor NA são ignoradas, e as previsões de observações com algum valor NA são retornadas como NA também. Para tornar possível fazer as previsões de todos os jogos da temporada 2018/19 e não apenas daqueles que tem informação completa, é possível identificar os padrões diferentes de NA's nas linhas para a base dessa temporada transformando a base de dados toda em uma matriz de 0's e 1's, sendo 0 quando a observação tem informação, e 1 quando ela é NA. Assim, cada linha da base se torna um vetor de 0's e 1's, e o padrão de cada linha pode ser identificado concatenando esses 0's e 1's (ver tabela 2). Feito isso, foram identificados 61 padrões diferentes para a base da temporada 2018/19. Então, para cada padrão, são identificadas as linhas que possuem aquele padrão, e as colunas das variáveis que são NA nesse padrão são retiradas. Por fim, em uma base que contém as observações das temporadas anteriores, retiramos essas mesmas colunas, e depois são retiradas as linhas em que ainda existe algum NA nas colunas que sobraram. Com isso, é possível fazer modelos com essa base das temporadas anteriores, e utilizá-los para fazer previsões para todos os 1230 jogos da temporada regular de 2018/19.

Tabela 2 – Exemplo de padrão de NA's

Variável	Valor	Vetor de 0's e 1's
<i>Wins_T_Vis</i>	1	0
<i>Loss_T_Vis</i>	1	0
<i>Mean_Last3_total_Vis</i>	NA	1
<i>Wins_T_Home</i>	2	0
<i>Loss_T_Home</i>	2	0
<i>Win_Last5_total_Home</i>	NA	1

Um exemplo fictício de um padrão de NA é visto na tabela 2. Nesse exemplo, o time visitante havia jogado 2 jogos (1 vitória e 1 derrota), ou seja, não há um valor de média de pontos nos últimos 3 jogos. O time mandante havia jogado 4 jogos (2 vitórias e 2 derrotas) e portanto não há valor do número de vitórias nos últimos 5 jogos.

3.4 Casas de Aposta

Em muitos lugares no mundo, existem plataformas onde é possível apostar dinheiro em acontecimentos futuros, e a categoria mais popular é a de esportes. Esses lugares são chamados de casas de aposta, e possuem equipes profissionais que determinam métricas para definir números para as apostas.

A “linha” é um desses artifícios que as casas de aposta dos Estados Unidos usam para equilibrar a aposta em um time em cada partida. Se a casa considera o time A favorito pra ganhar por X pontos, então uma linha é determinada com base nesse número, e para alguém apostar no time A e vencer a aposta, o time A precisa vencer o jogo por uma diferença de pelo menos X pontos. Um exemplo: uma casa de aposta determina que o Golden State Warriors é o favorito em um jogo contra o Portland Trail Blazers por 6.5 pontos, logo, a “linha” é -6.5 para os Warriors e +6.5 para o Trail Blazers, ou seja, quem apostar nos Warriors precisa que o time vença o jogo por uma diferença de 7 pontos ou mais para ganhar a aposta, e quem apostar no Trail Blazers precisa que o time perca por no máximo 6 pontos de diferença, ou vença o jogo, para ganhar a aposta. É comum o uso de meio décimo na linha para evitar empates, mas não é obrigatório, caso uma “linha” seja -7 e o time vença por exatos 7 pontos de diferença, normalmente a aposta é ressarcida ao apostador e nem ele, nem a casa ganha a aposta.

Como é feita a determinação dessas “linhas” é segredo de cada casa de aposta, mas há muitos recursos e pessoal disponíveis, além da possibilidade de levar em conta informações como a situação dos jogadores de cada time, por exemplo, se o jogador principal de um time não vai jogar em um jogo específico, isso é definitivamente levado em conta na determinação da “linha” do jogo.

Utilizando-se da mesma técnica de *web scraping* citada na seção 3.3, foi possível

extrair do site da ESPN Americana (ESPN, 2019) a “linha” de aposta da maioria dos jogos da temporada. O site da ESPN faz uma média da “linha” de várias casas de aposta diferentes logo antes do início de cada partida, e deixa na página de cada jogo.

4 Resultados

4.1 Resultados Reais da Temporada 2018/19

Nessa seção serão listados alguns resultados e estatísticas das equipes ao final da temporada regular.































Team	W	L	Team	W	L
1  Bucks	60	22	1  Warriors	57	25
2  Raptors	58	24	2  Nuggets	54	28
3  76ers	51	31	3  Trail Blazers	53	29
4  Celtics	49	33	4  Rockets	53	29
5  Pacers	48	34	5  Jazz	50	32
6  Nets	42	40	6  Thunder	49	33
7  Magic	42	40	7  Spurs	48	34
8  Pistons	41	41	8  Clippers	48	34
9  Hornets	39	43	9  Kings	39	43
10  Heat	39	43	10  Lakers	37	45
11  Wizards	32	50	11  Timberwolves	36	46
12  Hawks	29	53	12  Grizzlies	33	49
13  Bulls	22	60	13  Pelicans	33	49
14  Cavaliers	19	63	14  Mavericks	33	49
15  Knicks	17	65	15  Suns	19	63

Figura 4 – Classificação Final da NBA separado por conferência Leste (esquerda) e Oeste (direita)

Na figura 4, é observado o número de vitórias (W) e derrotas (L) de cada time na temporada 2018/19. Os 8 primeiros times de cada conferência se classificaram para os *playoffs*.

Tabela 3 – Médias de pontos marcados e sofridos por equipe
Fonte: (NBA, 2019)

Time	Média de Pontos Sofridos	Média de Pontos Marcados
Atlanta Hawks	119.4	113.3
Boston Celtics	108	112.4
Brooklyn Nets	112.3	112.2
Charlotte Hornets	111.8	110.7
Chicago Bulls	113.4	104.9
Cleveland Cavaliers	114.1	104.5
Dallas Mavericks	110.1	108.9
Denver Nuggets	106.7	110.7
Detroit Pistons	107.3	107
Golden State Warriors	111.2	117.7
Houston Rockets	109.1	113.9
Indiana Pacers	104.7	108
Los Angeles Clippers	114.3	115.1
Los Angeles Lakers	113.5	111.8
Memphis Grizzlies	106.1	103.5
Miami Heat	105.9	105.7
Milwaukee Bucks	109.3	118.1
Minnesota Timberwolves	114	112.5
New Orleans Pelicans	116.8	115.4
New York Knicks	113.8	104.6
Oklahoma City Thunder	111.1	114.5
Orlando Magic	106.6	107.3
Philadelphia 76ers	112.5	115.2
Phoenix Suns	116.8	107.5
Portland Trail Blazers	110.5	114.7
Sacramento Kings	115.3	114.2
San Antonio Spurs	110	111.7
Toronto Raptors	108.4	114.4
Utah Jazz	106.5	111.7
Washington Wizards	116.9	114

Na tabela 3, é possível ver que o Indiana Pacers teve a melhor defesa da temporada no quesito pontos, e o Atlanta Hawks a pior. Considerando pontos marcados, o Milwaukee Bucks foi o time de melhor ataque na temporada, e o Memphis Grizzlies o pior.

4.2 Previsões

Utilizando as bases de dados definidas na Metodologia, será possível aplicar os métodos estatísticos propostos no Capítulo 2 para fazer modelos e obter previsões para os 1230 jogos da temporada regular de 2018/19.

De início, foram utilizadas todas as bases disponíveis (da temporada 2000/01 até 2017/18) para o ajuste dos modelos. Diferentemente do usual na estatística, a base não foi

dividida em treinamento e validação, pois os jogos são uma sequência histórica no tempo, e não faria sentido utilizar jogos do “futuro” para prever jogos que aconteceram antes.

Tabela 4 – Porcentagem de Acerto das previsões dos jogos da temporada 2018/19 para cada método utilizando dados de 2000/01 a 2017/18 na modelagem

Método	Porcentagem de Acerto
Regressão de Probit	0.6723577
Regressão Logística	0.6707317
Análise de Discriminante Linear	0.6682927
Regressão de Probit c/ Forward	0.6682927
Regressão Logística c/ Forward	0.6674797
SVM com $cost = 8$, $gamma = 10^{-4}$	0.6666667
Regressão Linear c/ Forward	0.6658537
Regressão Linear	0.6634146
SVM padrão	0.6577236
Random Forest	0.6373984
Regressão em Árvore	0.6373984
Classificação em Árvore	0.6089431

Na tabela 4, temos a porcentagem de acerto das previsões para cada método utilizado. O melhor resultado foi obtido com a Regressão de Probit, com 0.6723577 de acurácia. Para o método SVM, como mencionado na seção 2.5, foram testados outros valores dos parâmetros $cost$ e $gamma$ com a utilização da função *tune*, e o melhor resultado foi obtido com os valores 8 e 10^{-4} , respectivamente.

Alguns métodos são mais eficientes computacionalmente, em termos de tempo decorrido para a execução das modelagens, e para esses métodos, é possível realizar as previsões modificando as temporadas a serem utilizadas na modelagem, para ser observada a evolução dos resultados. Esses métodos são: regressão linear, regressão logística, regressão de probit, análise de discriminante linear, regressão em árvore e classificação em árvore.

Nas tabela 5, é observada a porcentagem de acerto das previsões dos jogos da temporada 2018/19 para os métodos citados acima, utilizando diferentes temporadas para as modelagens. A Análise de Discriminante Linear foi abreviada para LDA. A primeira coluna indica que a modelagem foi feita usando os jogos da temporada indicada até a temporada 2017/18. Na primeira linha, que se diz que a temporada de início é 2000/2001, as modelagens foram iguais as usadas na Tabela 4. A última linha indica uma previsão utilizando apenas a temporada 2017/18 na modelagem. O melhor resultado geral foi encontrado utilizando a Regressão Logística quando foram utilizadas as temporadas 2006/2007 a 2017/18 para a modelagem. Em média, os melhores resultados são obtidos quando utilizadas essas temporadas na modelagem.

Tabela 5 – Porcentagem de acerto das previsões dos jogos da temporada 2018/19 para cada método utilizando temporadas diferentes na modelagem

Temporada de Início	Regressão Linear	Regressão Logística	Regressão de Probit	LDA	Regressão em Árvore	Classificação em Árvore
2000/2001	0.663	0.671	0.672	0.668	0.637	0.609
2001/2002	0.664	0.664	0.667	0.665	0.640	0.611
2002/2003	0.669	0.672	0.673	0.672	0.640	0.636
2003/2004	0.667	0.673	0.675	0.674	0.637	0.634
2004/2005	0.667	0.672	0.672	0.672	0.642	0.642
2005/2006	0.668	0.672	0.674	0.676	0.628	0.642
2006/2007	0.675	0.681	0.677	0.679	0.624	0.645
2007/2008	0.670	0.677	0.680	0.679	0.624	0.648
2008/2009	0.670	0.675	0.673	0.675	0.641	0.646
2009/2010	0.662	0.678	0.678	0.675	0.638	0.612
2010/2011	0.669	0.667	0.667	0.664	0.640	0.615
2011/2012	0.667	0.665	0.665	0.665	0.641	0.640
2012/2013	0.661	0.662	0.663	0.662	0.639	0.613
2013/2014	0.661	0.665	0.661	0.662	0.629	0.613
2014/2015	0.668	0.672	0.672	0.671	0.608	0.607
2015/2016	0.654	0.654	0.648	0.646	0.628	0.591
2016/2017	0.649	0.645	0.641	0.640	0.618	0.590
2017/2018	0.606	0.594	0.595	0.599	0.589	0.573

Dado esses resultados, novamente serão calculadas as porcentagens de acertos das previsões para todos os métodos, mas dessa vez utilizando apenas as temporadas de 2006/2007 a 2017/2018 na modelagem.

Tabela 6 – Porcentagem de Acerto das previsões dos jogos da temporada 2018/19 para cada método utilizando dados de 2006/07 a 2017/18 na modelagem

Método	Porcentagem de Acerto
Regressão Logística	0.6813008
Análise de Discriminante Linear	0.6788618
Regressão de Probit	0.6772358
Regressão Linear	0.6747967
SVM com $cost = 8$, $gamma = 10^{-4}$	0.6731707
Regressão Linear c/ Forward	0.6707317
Regressão Logística c/ Forward	0.6682927
Regressão de Probit c/ Forward	0.6642276
SVM padrão	0.6569106
Classificação em Árvore	0.6447154
Random Forest	0.6373984
Regressão em Árvore	0.6243902

Comparando a Tabela 4 com a Tabela 6, é observado que na Tabela 6 a maioria dos métodos obtém melhor porcentagem de acerto das previsões, e o melhor resultado

geral é o da Regressão Logística quando utilizadas as temporadas de 2006/07 a 2017/2018 na modelagem.

4.2.1 “Previsões” das casas de aposta

Considerando o que foi descrito na seção 3.4, a “linha” de aposta de cada jogo diz qual time era considerado o favorito para vencer o jogo pelas casas de aposta, e podemos obter a porcentagem de acerto média das casas de aposta para os jogos da temporada.

Na página da *web* de 4 dos 1230 jogos da temporada, o valor da “linha” não estava disponível, e não há como recuperar essa informação. Em outros 16 jogos, a “linha” era *even*, ou seja, esses jogos foram julgados tão equilibrados, que em média não foram apontados times favoritos, ou seja, se não foi determinado um time favorito, não seria justo contar esses jogos para as “previsões” das casas de aposta, pois não há possibilidade de acerto.

Excluindo esses 20 jogos citados acima, a porcentagem de acerto calculada para as casas de aposta foi de 0.6727273, em 1210 jogos. Considerando o melhor modelo obtido nesse trabalho, a taxa de acerto das previsões foi melhor do que as das casas de aposta.

4.3 Tempo de execução computacional de cada método

Para efeito de comparação, foi medido o tempo necessário para a execução computacional dos códigos de modelagem e previsão para cada método. Foi feita a modelagem das temporadas 2006/07 a 2017/18 para todos os métodos.

É lembrado que para cada método é feita não só uma, mas sim 61 modelagens, como visto na subseção 3.3.1, por conta dos valores faltantes, para ser possível ter a previsão de todos os jogos da temporada 2018/19.

Os resultados estão apresentados na Tabela 7. Além de apresentarem os melhores resultados na tabela 6, os métodos de Regressão Logística, Regressão Linear, Regressão de Probit e Análise de Discriminante Linear são dos mais rápidos de executar computacionalmente, indicando que são os melhores métodos para esse trabalho.

Tabela 7 – Tempo de execução do código computacional para cada método

Método	Tempo (em segundos)
Regressão Linear	9.733
Classificação em Árvore	19.692
Regressão em Árvore	21.069
Regressão Logística	30.542
Regressão de Probit	33.780
Análise de Discriminante Linear	35.057
Regressão Linear c/ Forward	985.550
Regressão de Probit c/ Forward	5359.306
Regressão Logística c/ Forward	6095.090
SVM	9420.622
Random Forest	31367.020

4.4 Modelo “campeão”

Como visto na seção 4.2, o modelo com o melhor resultado geral das previsões foi o de Regressão Logística quando utilizadas as temporadas de 2006/07 a 2017/2018 na modelagem. Além disso, é um dos métodos mais rápidos computacionalmente, demorando em torno de 30 segundos de execução para serem obtidos esses resultados.

Nessa seção serão analisados alguns aspectos das previsões obtidas por esse modelo.

4.4.1 Comparação dos resultados reais com as previsões

A tabela de classificação final da temporada regular foi apresentada na figura 4, agora, ela será comparada com a previsão do número de vitórias de cada equipe segundo o modelo.

É possível perceber pelas tabelas 8 e 9, que geralmente o modelo prevê mais vitórias do que o verdadeiro para os times da parte de cima da tabela e mais derrotas do que o real para os times da parte de baixo da tabela. Isso é um padrão esperado, pois não é fácil prever “zebras”, isto é, quando um time com números piores acaba vencendo um time com números melhores, o que na prática acontece de vez em quando.

Por exemplo, é muito difícil um time visitante vencer o time mandante quando o time mandante é melhor, e é ainda mais difícil prever a vitória do visitante. Para contexto, do jogo de número 300 pra frente nessa temporada, em que a maioria dos times já jogou pelo menos 20 jogos, ocorreram 434 jogos em que o time mandante tinha mais vitórias no campeonato do que o time visitante. Desses 434 jogos, em 107 o time visitante conseguiu a vitória (24.65%). Dessas 107 vitórias, o modelo campeão conseguiu prever essa “zebra” em apenas 14 jogos (13.08%). Com esse exemplo é fácil ver a dificuldade do acerto da previsão em jogos que acontecem resultados improváveis.

Tabela 8 – Comparação das vitórias reais com as vitórias previstas - Conferência Leste

Time	Vitórias Reais	Vitórias Previstas
Milwaukee Bucks	60	74
Toronto Raptors	58	69
Philadelphia 76ers	51	59
Boston Celtics	49	55
Indiana Pacers	48	54
Brooklyn Nets	42	45
Orlando Magic	42	39
Detroit Pistons	41	39
Charlotte Hornets	39	36
Miami Heat	39	32
Washington Wizards	32	28
Atlanta Hawks	29	14
Chicago Bulls	22	10
Cleveland Cavaliers	19	9
New York Knicks	17	5

Tabela 9 – Comparação das vitórias reais com as vitórias previstas - Conferência Oeste

Time	Vitórias Reais	Vitórias Previstas
Golden State Warriors	57	65
Denver Nuggets	54	65
Portland Trail Blazers	53	63
Houston Rockets	53	60
Utah Jazz	50	59
Oklahoma City Thunder	49	54
Los Angeles Clippers	48	51
San Antonio Spurs	48	49
Sacramento Kings	39	37
Los Angeles Lakers	37	32
Minnesota Timberwolves	36	32
Dallas Mavericks	33	30
Memphis Grizzlies	33	29
New Orleans Pelicans	33	29
Phoenix Suns	19	7

Por outro lado, percebe-se que a classificação de ambas as conferências terminaria na mesma ordem se consideradas as vitórias previstas pelo modelo.

4.4.1.1 Acertos por Equipe

Para uma análise mais profunda das previsões, serão verificados os acertos das mesmas separados por cada equipe.

Tabela 10 – Acertos das previsões por time por local e resultado do jogo - Parte 1

Time	Total de Acertos das Previsões (%)	Acertos nos jogos em casa (%)	Acertos nos jogos fora (%)	Acertos nas vitórias (%)	Acertos nas derrotas (%)
CLE	64 (.780)	29 (.707)	35 (.854)	4 (.211)	60 (.952)
PHX	62 (.756)	29 (.707)	33 (.805)	2 (.105)	60 (.952)
DEN	61 (.744)	35 (.854)	26 (.634)	48 (.889)	13 (.464)
DAL	61 (.744)	28 (.683)	33 (.805)	24 (.727)	37 (.755)
IND	61 (.744)	29 (.707)	32 (.780)	46 (.958)	15 (.441)
TOR	61 (.744)	31 (.756)	30 (.732)	53 (.914)	8 (.333)
POR	61 (.744)	32 (.780)	29 (.707)	43 (.811)	18 (.621)
NYK	59 (.720)	27 (.659)	32 (.780)	4 (.235)	55 (.846)
DET	59 (.720)	27 (.659)	32 (.780)	25 (.610)	34 (.829)
MIN	58 (.707)	27 (.659)	31 (.756)	20 (.556)	38 (.826)
SAC	58 (.707)	27 (.659)	31 (.756)	22 (.564)	36 (.837)
LAC	58 (.707)	26 (.634)	32 (.780)	39 (.812)	19 (.559)
ATL	57 (.695)	26 (.634)	31 (.756)	7 (.241)	50 (.943)
CHI	57 (.695)	28 (.683)	29 (.707)	3 (.136)	54 (.900)
MIL	56 (.683)	33 (.805)	23 (.561)	54 (.900)	2 (.091)
GSW	56 (.683)	30 (.732)	26 (.634)	48 (.842)	8 (.320)
UTA	55 (.671)	29 (.707)	26 (.634)	41 (.820)	14 (.438)
BOS	55 (.671)	27 (.659)	28 (.683)	41 (.837)	14 (.424)
SAS	55 (.671)	28 (.683)	27 (.659)	35 (.729)	20 (.588)
BKN	54 (.659)	27 (.659)	27 (.659)	23 (.548)	31 (.775)
CHA	54 (.659)	26 (.634)	28 (.683)	28 (.718)	26 (.605)
ORL	54 (.659)	29 (.707)	25 (.610)	23 (.548)	31 (.775)
MEM	53 (.646)	23 (.561)	30 (.732)	17 (.515)	36 (.735)
WAS	53 (.646)	23 (.561)	30 (.732)	16 (.500)	37 (.740)
PHI	52 (.634)	32 (.780)	20 (.488)	38 (.745)	14 (.452)
NOP	50 (.610)	27 (.659)	23 (.561)	20 (.606)	30 (.612)
HOU	50 (.610)	31 (.756)	19 (.463)	36 (.679)	14 (.483)
OKC	48 (.585)	27 (.659)	21 (.512)	37 (.755)	11 (.333)
MIA	48 (.585)	21 (.512)	27 (.659)	21 (.538)	27 (.628)
LAL	46 (.561)	24 (.585)	22 (.537)	20 (.541)	26 (.578)

Tabela 11 – Acertos das previsões por time por local e resultado do jogo - Parte 2

Time	Acertos nas vitórias fora (%)	Acertos nas derrotas fora (%)	Acertos nas vitórias em casa (%)	Acertos nas derrotas em casa (%)
CLE	0 (.000) (6 vit. fora)	35 (1.000)	4 (.308)	25 (.893)
PHX	0 (.000) (7 vit. fora)	33 (.971)	2 (.167)	27 (.931)
DEN	14 (.700)	12 (.571)	34 (1.000)	1 (.143)
DAL	4 (.444)	29 (.906)	20 (.833)	8 (.471)
IND	17 (.895)	15 (.682)	29 (1.000)	0 (.000) (12 der. casa)
TOR	22 (.846)	8 (.533)	31 (.969)	0 (.000) (9 der. casa)
POR	15 (.714)	14 (.700)	28 (.875)	4 (.444)
NYK	1 (.125)	31 (.939)	3 (.333)	24 (.750)
DET	8 (.533)	24 (.923)	17 (.654)	10 (.667)
MIN	4 (.364)	27 (.900)	16 (.640)	11 (.688)
SAC	7 (.467)	24 (.923)	15 (.625)	12 (.706)
LAC	17 (.773)	15 (.789)	22 (.846)	4 (.267)
ATL	2 (.167)	29 (1.000)	5 (.294)	21 (.875)
CHI	1 (.077)	28 (1.000)	2 (.222)	26 (.813)
MIL	21 (.778)	2 (.143)	33 (1.000)	0 (.000) (8 der. casa)
GSW	18 (.667)	8 (.571)	30 (1.000)	0 (.000) (11 der. casa)
UTA	14 (.667)	12 (.600)	27 (.931)	2 (.167)
BOS	14 (.667)	14 (.700)	27 (.964)	0 (.000) (13 der. casa)
SAS	10 (.625)	17 (.680)	25 (.781)	3 (.333)
BKN	6 (.316)	21 (.955)	17 (.739)	10 (.556)
CHA	8 (.571)	20 (.741)	20 (.800)	6 (.375)
ORL	5 (.294)	20 (.833)	18 (.720)	11 (.688)
MEM	5 (.417)	25 (.862)	12 (.571)	11 (.550)
WAS	3 (.300)	27 (.871)	13 (.591)	10 (.526)
PHI	9 (.450)	11 (.524)	29 (.935)	3 (.300)
NOP	5 (.357)	18 (.667)	15 (.789)	12 (.545)
HOU	10 (.455)	9 (.474)	26 (.839)	5 (.500)
OKC	12 (.545)	9 (.474)	25 (.926)	2 (.143)
MIA	9 (.450)	18 (.857)	12 (.632)	9 (.409)
LAL	5 (.333)	17 (.654)	15 (.682)	9 (.474)

Nas tabelas 10 e 11, são apresentados os totais e porcentagens de acertos das previsões por time, separados por algumas categorias: apenas nos jogos em casa, nos jogos fora de casa, nas vitórias, nas derrotas, e nas vitórias e derrotas separadas por fora de casa ou em casa. Os nomes dos times foram abreviados para as siglas oficiais de 3 letras de cada equipe. Cada time jogou 82 jogos no total, sendo 41 em casa e 41 fora de casa. Para as categorias em que não houve nenhum acerto nas previsões, foi colocado o total de ocorrências daquela categoria para se ter essa referência.

Por essas tabelas, é possível perceber que em geral, há mais acertos nas derrotas para os times que foram mal no campeonato, e mais acertos nas vitórias para os times que foram bem no campeonato. Isso fica ainda mais claro na tabela 11, onde é possível ver que não foi acertada nenhuma vitória fora de casa de times que foram muito mal (Cleveland e Phoenix), enquanto as derrotas fora de casa foram quase todas acertos. No outro extremo, não foi acertada nenhuma derrota em casa de times estatisticamente dominantes (Indiana, Toronto, Milwaukee, Golden State e Boston), por outro lado acertou-se praticamente todas as vitórias em casa desses times. Novamente, isso pode ser atribuído à dificuldade da previsão de “zebras”.

Outro aspecto que se pode inferir da tabela 11 é que a taxa de acerto de vitórias em casa é maior do que de vitórias fora de casa, e de derrotas fora é maior do que de derrotas em casa, para quase todos os times. A única exceção foi o Houston Rockets, em que acertou-se (em porcentagem) mais derrotas em casa do que derrotas fora. Isso acontece devido ao maior número de vitórias dos times mandantes no campeonato, por exemplo, tendo em vista os resultados reais, nessa temporada, em 59.27% dos jogos o time mandante venceu, e o modelo de previsões ainda superestima esse número, prevendo que o time mandante venceria em 65.61% dos jogos.

4.4.2 Variáveis mais significativas

Para o modelo campeão, foram usadas todas as variáveis do banco de dados na modelagem, e usando a função *standardize* do pacote de mesmo nome (EAGER, 2017), podemos padronizar os dados para mais fácil interpretação dos valores dos parâmetros das variáveis.

Como são feitos 61 modelos para a obtenção da previsão de todos os jogos, e seria inviável listar as variáveis mais significativas de cada um, será colocado aqui apenas para o modelo completo, com todas as variáveis, sem nenhum NA.

Com isso, na tabela 12 seguem as variáveis com p-valor de significância menor que 0.1 para esse modelo.

Isso foi feito para o modelo campeão, que é uma regressão logística, e a variável resposta usada foi a *Win_Vis* (a explicação do que é cada variável se encontra na seção 3.3),

Tabela 12 – Variáveis mais significativas no modelo

Variável	Estimativa do Parâmetro β	Erro Padrão	Z (Estatística de Teste)	p-valor
Mean_Pts_A_T_Vis	-0.34173	0.120916	-2.826	0.00471
Min_Last5home_Home	-0.18985	0.07072	-2.685	0.00726
Loss_T_Vis	-0.59691	0.227052	-2.629	0.00856
Days_LG_Vis	0.062418	0.024913	2.505	0.01223
Mean_Last3_home_opp_Home	-0.34262	0.142105	-2.411	0.01591
Mean_Pts_S_T_Vis	0.295485	0.123971	2.383	0.01715
Min_Last3home_opp_Home	0.176757	0.075284	2.348	0.01888
Mean_Pts_A_T_Home	0.312476	0.133186	2.346	0.01897
Max_Last3_home_opp_Home	0.163982	0.077887	2.105	0.03526
OT_last_HomeTRUE	0.096106	0.045954	2.091	0.0365
Min_Last5total_opp_Home	0.145477	0.069784	2.085	0.0371
Win_Last3_total_Home	0.142964	0.068954	2.073	0.03814
Str_Sch_Vis	0.052763	0.025495	2.069	0.0385
Max_Pts_S_H_Home	-0.08071	0.041444	-1.947	0.05147
Loss_A_Vis	0.231707	0.122455	1.892	0.05847
Wins_H_Home	0.23373	0.126269	1.851	0.06416
Max_Last10_total_Vis	-0.10887	0.059288	-1.836	0.06631
Max_Last5_away_Vis	0.124937	0.072045	1.734	0.08289
Max_Pts_A_H_Home	0.069059	0.040352	1.711	0.087
Days_LG_Home	-0.04212	0.024785	-1.699	0.08924
Min_Last10total_Vis	-0.09652	0.05732	-1.684	0.09221

isso quer dizer que parâmetros positivos indicam que a variável contribui para o aumento da probabilidade de vitória do time visitante, e intuitivamente, parâmetros negativos indicam que a variável contribui para o aumento da probabilidade de vitória do time mandante.

A variável mais significativa foi *Mean_Pts_A_T_Vis*. A estimativa do parâmetro dessa variável é negativa, o que significa que quando maior a média de pontos sofridos do time visitante, maior a probabilidade de vitória do time mandante, o que faz todo sentido.

Apenas por curiosidade, se forem utilizados os dados padronizados da função *standardize* para fazer as previsões dos jogos que tem informação em todas as variáveis (875 jogos), é obtida uma taxa de acerto de 0.6868571. Para os dados normais sem padronização, nesses mesmos jogos, a taxa de acerto obtida anteriormente foi de 0.6891429.

4.4.3 Comparação do modelo “campeão” com as casas de aposta

Na figura 5, está representada a porcentagem de acerto das previsões durante a temporada, do modelo campeão e das “previsões” das casas de aposta, para efeitos de comparação entre os dois.

No começo da temporada, a taxa é bem errática, pois é uma amostra pequena. Percebe-se que para as casas de aposta, a taxa de acerto se estabiliza um pouco mais rápido, o que é um indicativo de que para eles a taxa de acerto é mais constante.

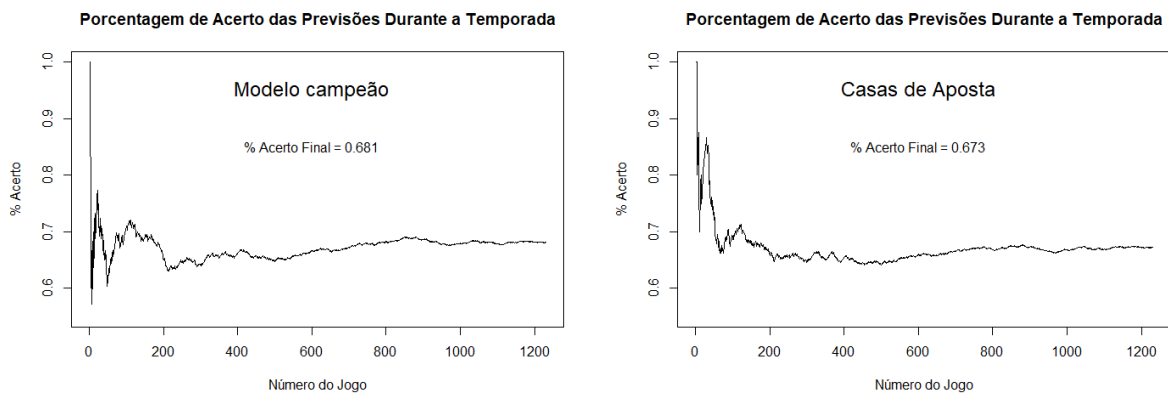


Figura 5 – Evolução da percentagem de acerto das previsões ao longo da temporada

Na figura 6, é colocada a percentagem de acerto das previsões nos últimos 61 jogos (aproximadamente 5% de 1230), para cada jogo a partir do 61º. Por exemplo, no ponto 61, está representada a percentagem de acerto das previsões dos jogos 1 ao 61, no ponto 62 está representada a percentagem de acerto das previsões dos jogos 2 ao 62, e assim por diante.

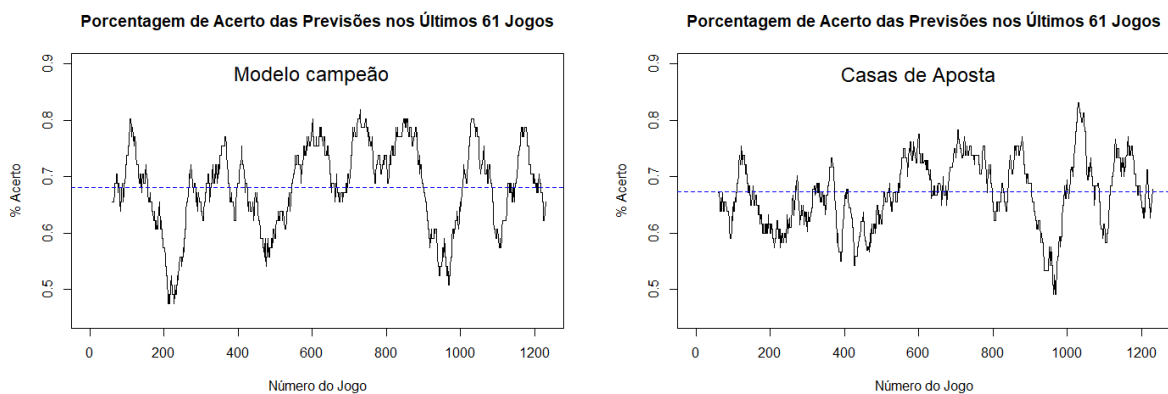


Figura 6 – Porcentagem de acerto das previsões dos últimos 61 jogos ao longo da temporada

A linha azul nesses gráficos representa a percentagem de acerto das previsões ao final da temporada, os resultados vistos anteriormente no trabalho.

Nesses gráficos alguns padrões são repetidos tanto para o modelo campeão quanto para as previsões das casas de aposta, mas a percentagem das casas de aposta é mais constante, com menos valores extremos no gráfico.

A queda brusca no gráfico do modelo campeão, em torno do jogo 200, pode ser explicada pela falta de informações no começo da temporada. Como foi visto anteriormente, há muitos valores NA no banco de dados no começo da temporada.

A queda percebida nos dois gráficos em torno do jogo 900 pode ter relação com o

fim do período de trocas de jogadores entre os times da NBA, pois muitos times trocam jogadores perto do fim desse período, que nessa temporada foi após o jogo de número 816. Nos jogos subsequentes a esse, os times estão se adaptando às trocas de jogadores e, conseqüentemente, alguns times melhoram e outros pioram, dificultando o acerto das previsões para os modelos.

4.4.4 Adicionando jogos de 2018/19 na modelagem

As previsões até aqui foram feitas sem colocar nenhum jogo da temporada 2018/19 na modelagem, pois para fazer as previsões inserindo os jogos da temporada na modelagem conforme os jogos vão acontecendo aumentaria muito o número de modelos necessários para realizar as previsões, e devido ao tempo de execução de alguns métodos seria impossível implementar dessa maneira para todos eles.

Portanto, foi decidido realizar as previsões dessa maneira apenas para o modelo campeão. Como acontecem vários jogos no mesmo dia, e muitas vezes no mesmo horário, as previsões foram sendo feitas para os jogos de cada dia em que houve partidas, e as partidas já realizadas até o dia anterior foram sendo adicionadas na modelagem.

Isso foi feito de duas maneiras: a primeira foi deixando todos os jogos de 2006/07 até 2017/18 e apenas adicionando os jogos de 2018/19, e a outra maneira foi retirando o mesmo número de jogos que foram sendo adicionados, ou seja, conforme foram entrando as partidas de 2018/19, as partidas mais antigas de 2006/07 foram saindo, desse jeito, o números de jogos para a modelagem se manteve constante.

Do primeiro jeito, a porcentagem de acerto das previsões foi 0.6756098, e do segundo foi 0.6731707. Surpreendentemente, ambos os resultados foram piores do que o obtido sem inserir nenhum jogo da temporada 2018/19 na modelagem.

4.5 Comparação do modelo de regressão linear com as “linhas” de aposta e com os resultados reais

Foi visto na tabela 6 que dos modelos que usam a variável quantitativa de saldo de pontos como variável dependente, o modelo de Regressão Linear foi o que obteve a melhor porcentagem de acerto das previsões. E como citado na seção 3.4, as casas de aposta providenciam a “linha” de aposta de cada jogo, que significa por quantos pontos os times são considerados favoritos.

Com as previsões da Regressão Linear, obtemos o número esperado do saldo de pontos para cada jogo, e podemos comparar com os resultados reais, e com as “linhas” de aposta.

Tabela 13 – Resumo das diferenças absolutas das Previsões da Regressão Linear vs. linhas de aposta vs. resultados reais

Comparação	Mín.	1º Quartil	Mediana	Média	3º Quartil	Máx.	NA's
Regressão Linear vs. Resultados Reais	0.006	3.844	8.123	10.276	14.602	50.532	-
Linhas de aposta vs. Resultados Reais	0.000	4.000	8.000	9.927	14.000	55.000	4
Regressão Linear vs. Linhas de aposta	0.001	1.055	2.359	2.905	4.114	16.991	4

Os NA's presentes na tabela 13 existem porque, como explicado na subseção 4.2.1, em 4 jogos não foi possível obter as linhas de aposta, e nos 16 em que era *even*, a linha foi considerada como sendo 0. Pela tabela, é possível ver uma grande similaridade entre as previsões da regressão linear e as linhas de aposta, onde é indicado que a média das diferenças absolutas entre elas é menos de 3, ou seja, o saldo de pontos previsto pela regressão linear está, em média, a apenas 3 pontos de diferença da linha de aposta da partida.

Comparando com os resultados reais, tanto as previsões quanto as linhas de aposta costumam ser mais diferentes do resultado real, tendo em torno de 10 pontos de diferença absoluta média. A comparação de ambas com os resultados reais são bem parecidas.

5 Conclusão

O objetivo desse trabalho foi obter previsões para os jogos da temporada regular da NBA de 2018/19, e os melhores resultados obtidos foram superiores aos inferidos de casas de aposta dos Estados Unidos, o que indica que foram bons resultados.

Os métodos de Regressão Linear, Regressão Logística, Regressão de Probit, e a Análise de Discriminante Linear se mostraram os melhores tanto em relação ao acerto das previsões, quanto em relação ao tempo necessário para a execução dos códigos computacionais.

A grande desvantagem da base de dados aplicada nas modelagens realizadas nesse trabalho é a falta de informação sobre os jogadores, pois sem isso não é possível analisar a influência de situações que poderiam ser relevantes como: jogadores poupados, lesões, retorno de lesionados, trocas durante a temporada, etc. Todas essas informações levam tempo para a base de dados se “atualizar” sozinha, pois jogadores importantes impactam bastante os números dos times, por isso as variáveis que consideram as estatísticas dos últimos X jogos são muito importantes.

Para trabalhos futuros, é possível paralelizar computacionalmente a implementação dos métodos mais demorados, como o SVM e o *random forest*, para assim possibilitar até mesmo uma busca de parâmetros melhores nesses métodos. A implementação pode ser replicada para realizar previsões de jogos de temporadas futuras.

Referências

- AGRESTI, A. *An Introduction to Categorical Data Analysis*. Wiley, 2007. (Wiley Series in Probability and Statistics). ISBN 9780470114742. Disponível em: <<https://books.google.com.br/books?id=OG9Eqwd0Fh4C>>. Nenhuma citação no texto.
- BASKETBALL-REFERENCE. 2019. <<https://www.basketball-reference.com/>>. Acessado em: 11/06/2019. Citado na página 24.
- BREIMAN, L. Random forests. 2001. Acessado em: 01/06/2019. Disponível em: <<https://www.stat.berkeley.edu/~breiman/randomforest2001.pdf>>. Citado na página 21.
- CARVALHO, J. et al. *ANÁLISE DE PROBIT APLICADA A BIOENSAIOS COM INSETOS*. [S.l.: s.n.], 2017. ISBN 978-85-64937-08-6. Citado na página 17.
- EAGER, C. D. *standardize: Tools for Standardizing Variables for Regression in R*. [S.l.], 2017. R package version 0.2.1. Disponível em: <<https://CRAN.R-project.org/package=standardize>>. Citado na página 42.
- ESPN. 2019. <<http://www.espn.com/nba/scoreboard>>. Acessado em: 16/05/2019. Citado na página 31.
- GROTHENDIECK, G. *sqlf: Manipulate R Data Frames Using SQL*. [S.l.], 2017. R package version 0.4-11. Disponível em: <<https://CRAN.R-project.org/package=sqlf>>. Nenhuma citação no texto.
- JOHNSON, R. A.; WICHERN, D. W. *Applied Multivariate Statistical Analysis*. Pearson Prentice Hall, 2007. (Applied Multivariate Statistical Analysis). ISBN 9780131877153. Disponível em: <<https://books.google.com.br/books?id=gFWcQgAACAAJ>>. Citado na página 19.
- KASSAMBARA, A. *Machine Learning Essentials: Practical Guide in R*. CreateSpace Independent Publishing Platform, 2018. ISBN 9781986406857. Disponível em: <<https://books.google.com.br/books?id=745QDwAAQBAJ>>. Citado na página 16.
- KUTNER, M.; NACHTSHEIM, C.; NETER, J. *Applied Linear Regression Models*. McGraw-Hill Higher Education, 2003. (The McGraw-Hill/Irwin Series Operations and Decision Sciences). ISBN 9780072955675. Disponível em: <<https://books.google.com.br/books?id=0nAMAAAACAAJ>>. Nenhuma citação no texto.
- LIAW, A.; WIENER, M. Classification and regression by randomforest. *R News*, v. 2, n. 3, p. 18–22, 2002. Disponível em: <<https://CRAN.R-project.org/doc/Rnews/>>. Citado na página 23.
- MEYER, D. Support vector machines, the interface to libsvm in package e1071. 2019. Acessado em: 31/05/2019. Disponível em: <<https://cran.r-project.org/web/packages/e1071/vignettes/svmdoc.pdf>>. Citado na página 18.
- MEYER, D. et al. *e1071: Misc Functions of the Department of Statistics, Probability Theory Group (Formerly: E1071), TU Wien*. [S.l.], 2018. R package version 1.7-0.

Disponível em: <<https://CRAN.R-project.org/package=e1071>>. Citado 2 vezes nas páginas 19 e 23.

NBA. 2019. <<https://stats.nba.com/>>. Acessado em: 11/06/2019. Citado na página 34.

R Core Team. *R: A Language and Environment for Statistical Computing*. Vienna, Austria, 2018. Disponível em: <<https://www.R-project.org/>>. Citado na página 23.

RIPLEY, B. *tree: Classification and Regression Trees*. [S.l.], 2019. R package version 1.0-40. Disponível em: <<https://CRAN.R-project.org/package=tree>>. Citado na página 23.

RIPLEY, B. D. *Pattern Recognition and Neural Networks*. [S.l.]: Cambridge University Press, 1996. Citado na página 20.

SCHUMAKER, R. P.; SOLIEMAN, O. K.; CHEN, H. *Sports Data Mining*. 1st. ed. [S.l.]: Springer Publishing Company, Incorporated, 2010. ISBN 144196729X, 9781441967299. Citado na página 13.

SELECTORGADGET. 2019. <<https://selectorgadget.com/>>. Acessado em: 25/05/2019. Citado na página 24.

SPORTSLOGOS. 2019. <http://www.sportslogos.net/teams/list_by_league/6/National_Basketball_Association/NBA/logos/>. Acessado em: 13/10/2018. Citado na página 24.

SUÁREZ, E. et al. Selection of variables in a multiple linear regression model. In: _____. *Applications of Regression Models in Epidemiology*. John Wiley & Sons, Ltd, 2017. cap. 5, p. 77–86. ISBN 9781119212515. Disponível em: <<https://onlinelibrary.wiley.com/doi/abs/10.1002/9781119212515.ch5>>. Citado na página 17.

UUDMAE, J. Predicting nba game outcomes. Acessado em: 28/09/2018. Disponível em: <<http://cs229.stanford.edu/proj2017/final-reports/5231214.pdf>>. Nenhuma citação no texto.

VAPNIK, V. N. *The Nature of Statistical Learning Theory*. Berlin, Heidelberg: Springer-Verlag, 1995. ISBN 0-387-94559-8. Citado na página 18.

VENABLES, W. N.; RIPLEY, B. D. *Modern Applied Statistics with S*. Fourth. New York: Springer, 2002. ISBN 0-387-95457-0. Disponível em: <<http://www.stats.ox.ac.uk/pub/MASS4>>. Citado na página 23.

WICKHAM, H. *rvest: Easily Harvest (Scrape) Web Pages*. [S.l.], 2016. R package version 0.3.2. Disponível em: <<https://CRAN.R-project.org/package=rvest>>. Citado na página 24.

WICKHAM, H. *stringr: Simple, Consistent Wrappers for Common String Operations*. [S.l.], 2019. R package version 1.4.0. Disponível em: <<https://CRAN.R-project.org/package=stringr>>. Nenhuma citação no texto.

WICKHAM, H. et al. *dplyr: A Grammar of Data Manipulation*. [S.l.], 2018. R package version 0.7.8. Disponível em: <<https://CRAN.R-project.org/package=dplyr>>. Nenhuma citação no texto.

YAN, X.; SU, X. G. *Linear Regression Analysis: Theory and Computing*. River Edge, NJ, USA: World Scientific Publishing Co., Inc., 2009. ISBN 9789812834102, 9812834109. Citado na página 16.

APÊNDICE A – Códigos em R

```
#funções para web scraping dos dados da internet

ler_jogos <- function(url){
  library(rvest)
  html <- read_html(url)
  #nodes obtidos com o SelectorGadget, extensao do google chrome
  games <- html_nodes(html, ".center+ .center ,
                        .left:nth-child(5), td+ .right ,
                        .left:nth-child(3) , .poptip.right ,
                        .left:nth-child(1)")

  texto <- html_text(games)
  tabela <- data.frame()
  j <- 1
  for(i in 1:(length(texto)/7)){
    tabela[i,1] <- texto[j]
    tabela[i,2] <- texto[j+1]
    tabela[i,3] <- texto[j+2]
    tabela[i,4] <- texto[j+3]
    tabela[i,5] <- texto[j+4]
    tabela[i,6] <- texto[j+5]
    tabela[i,7] <- texto[j+6]
    j <- j+7
  }
  names(tabela) <- tabela[1,]
  tabela <- tabela[2:length(tabela[,1]),]
  return(tabela)
}

tabela_temporada <- function(year){
  if(year == 2019){
    months <- c("october","november","december","january",
                "february","march","april","may","june")
  }else if(year == 2012){
    months <- c("december","january","february","march",
                "april","may","june")
  }else if(year == 2006 | year == 2005 | year == 2000){
    months <- c("november","december","january","february",
                "march","april","may","june")
  }
}
```

```

}else{
  months <- c("october","november","december","january",
             "february","march","april","may","june")
}
cola1 <-
  paste("https://www.basketball-reference.com/leagues/NBA_",
        year,sep="")
cola2 <- paste(cola1,"_games-",months,sep="")
url <- paste(cola2,".html",sep="")
teste <- data.frame()
for(i in 1:length(url)){
  b <- ler_jogos(url[i])
  teste <- rbind(teste,b)
}
names(teste) <- c("Date", "Visitor", "PTS_Visitor", "Home",
                 "PTS_Home", "OT", "Attend")
teste$PTS_Visitor <- as.numeric(teste$PTS_Visitor)
teste$PTS_Home <- as.numeric(teste$PTS_Home)
teste$Attend <- as.numeric(gsub(","," ", teste$Attend))
Sys.setlocale("LC_ALL","English")
teste$Date <- as.Date(teste$Date, format="%a, %b %d, %Y")
teste$OT[teste$OT == ""] <- NA
return(teste)
}

#exemplo para obter os dados dos jogos da temporada 2000/01
jogos_2001 <- tabela_temporada(2001)

#####

#carrega as bases finais de todas as temporadas
load(file="final.rda")

#criando a base grande com as temporadas para modelagem
#só com os jogos de temporada regular
final <- data.frame()
for(i in 2007:2018){
  assign("base",get(paste("final",i,sep="")))
  if(i == 2001 | i == 2002 | i == 2003 | i == 2004){
    final <- rbind(final,base[1:1189,])
  }else if(i == 2012){
    final <- rbind(final,base[1:990,])
  }
}

```

```

    }else{
      final <- rbind(final,base[1:1230,])
    }
  }

#variáveis dependentes
#coluna 1 é Win_Vis
#coluna 2 result_Vis

#encontrando os padrões de NA
#cortando a base de 18/19 só com os jogos de temporada regular
teste19 <- final2019[1:1230,]
padrao19 <- is.na(teste19[,-c(1,2)])

for(i in 1:length(teste19$Win_Vis)){
  for(j in 1:ncol(padrao19)){
    padrao19[i,j] <- as.numeric(padrao19[i,j])
  }
}

oi19 <- apply(padrao19,1,paste,collapse="")
nomes <- names(summary(as.factor(oi19)))

##### fazendo as previsões de todos os jogos para cada método
#Reg Linear
vet <- c()
start_time <- Sys.time()
for(i in 1:length(nomes)){
  full19 <- oi19==nomes[i]
  td19 <- teste19[full19,]
  colun <- !is.na(td19[1,])
  td19 <- td19[,colun]
  td <- final[,colun]
  td <- td[rowSums(is.na(td)) == 0,]
  mod <- lm(result_Vis~., data=td[, -1])
  win <- (predict(mod, newdata=td19[, -1]) > 0) ==
    (td19$result_Vis > 0)
  vet <- c(vet,win)
}
end_time <- Sys.time()
tempo <- end_time - start_time
tempo

```

```

as.numeric(tempo)
mean(vet)

#Reg Logística
vet1 <- c()
start_time <- Sys.time()
library(dplyr)
for(i in 1:length(nomes)){
  full19 <- oi19==nomes[i]
  td19 <- teste19[full19,]
  coluna <- !is.na(td19[1,])
  td19 <- td19[,coluna]
  td <- final[,coluna]
  td <- td[rowSums(is.na(td)) == 0,]
  a <- glm(Win_Vis~., data = td[, -2],
           family=binomial(link = "logit"))
  probabilities <- a %>% predict(td19[, -2], type = "response")
  predicted.classes <- ifelse(probabilities > 0.5,
                              "TRUE", "FALSE")

  win <- predicted.classes == td19$Win_Vis
  vet1 <- c(vet1, win)
}
end_time <- Sys.time()
tempo <- end_time - start_time
tempo
as.numeric(tempo)
mean(vet1)

#Reg Probit
vet2 <- c()
start_time <- Sys.time()
library(dplyr)
for(i in 1:length(nomes)){
  full19 <- oi19==nomes[i]
  td19 <- teste19[full19,]
  coluna <- !is.na(td19[1,])
  td19 <- td19[,coluna]
  td <- final[,coluna]
  td <- td[rowSums(is.na(td)) == 0,]
  a <- glm(Win_Vis~., data = td[, -2],
           family=binomial(link = "probit"))
  probabilities <- a %>% predict(td19[, -2], type = "response")

```

```

predicted.classes <- ifelse(probabilities > 0.5,
                             "TRUE", "FALSE")
win <- predicted.classes == td19$Win_Vis
vet2 <- c(vet2,win)
}
end_time <- Sys.time()
tempo <- end_time - start_time
tempo
as.numeric(tempo)
mean(vet2)

#SVM parametros mudados
vet3 <- c()
library(e1071)
for(i in 1:length(nomes)){
  if(i==1){
    full19 <- oi19==nomes[i]
    td19 <- teste19[full19,]
    colun <- !is.na(td19[1,])
    td19 <- td19[,colun]
    td <- final[,colun]
    td <- td[rowSums(is.na(td)) == 0,]
    mod <- svm(result_Vis~., data=td[, -1], cost=3, gamma=10^-4)
    win <- (predict(mod, newdata=td19[, -1]) > 0) ==
      (td19$result_Vis > 0)
    vet3 <- c(vet3,win)
  }
}
mean(vet3)

#SVM padrao
vet3.1 <- c()
start_time <- Sys.time()
library(e1071)
for(i in 1:length(nomes)){
  full19 <- oi19==nomes[i]
  td19 <- teste19[full19,]
  colun <- !is.na(td19[1,])
  td19 <- td19[,colun]
  td <- final[,colun]
  td <- td[rowSums(is.na(td)) == 0,]
  mod <- svm(result_Vis~., data=td[, -1])

```

```

win <- (predict(mod, newdata=td19[, -1]) > 0) ==
  (td19$result_Vis > 0)
vet3.1 <- c(vet3.1, win)
}
end_time <- Sys.time()
tempo <- end_time - start_time
tempo
as.numeric(tempo)*60*60

mean(vet3.1)

#LDA
vet4 <- c()
start_time <- Sys.time()
library(MASS)
for(i in 1:length(nomes)){
  full19 <- oi19==nomes[i]
  td19 <- teste19[full19,]
  colun <- !is.na(td19[1,])
  td19 <- td19[, colun]
  td <- final[, colun]
  td <- td[rowSums(is.na(td)) == 0,]
  mod <- lda(Win_Vis~., data=td[, -2])
  prd <- predict(mod, newdata=td19[, -2])
  win <- as.logical(prd$class) == td19$Win_Vis
  vet4 <- c(vet4, win)
}
end_time <- Sys.time()
tempo <- end_time - start_time
tempo
as.numeric(tempo)
mean(vet4)

#random forest
vet5 <- c()
start_time <- Sys.time()
library(randomForest)
for(i in 1:length(nomes)){
  full19 <- oi19==nomes[i]
  td19 <- teste19[full19,]
  colun <- !is.na(td19[1,])
  td19 <- td19[, colun]

```

```

    td <- final[,colun]
    td <- td[rowSums(is.na(td)) == 0,]
    mod <- randomForest(result_Vis~., data=td[, -1])
    win <- (predict(mod, newdata=td19[, -1]) > 0) ==
      (td19$result_Vis > 0)
    vet5 <- c(vet5, win)
  }
end_time <- Sys.time()
tempo <- end_time - start_time
tempo
as.numeric(tempo)*60*60
mean(vet5)

#reg arvore
vet6 <- c()
start_time <- Sys.time()
library(tree)
for(i in 1:length(nomes)){
  full19 <- oi19==nomes[i]
  td19 <- teste19[full19,]
  colun <- !is.na(td19[1,])
  td19 <- td19[, colun]
  td <- final[, colun]
  td <- td[rowSums(is.na(td)) == 0,]
  mod <- tree(result_Vis~., data=td[, -1])
  win <- (predict(mod, newdata=td19[, -1]) > 0) ==
    (td19$result_Vis > 0)
  vet6 <- c(vet6, win)
}
end_time <- Sys.time()
tempo <- end_time - start_time
tempo
as.numeric(tempo)
mean(vet6)

#class em arvore
vet7 <- c()
start_time <- Sys.time()
library(tree)
for(i in 1:length(nomes)){
  full19 <- oi19==nomes[i]
  td19 <- teste19[full19,]

```

```

colun <- !is.na(td19[1,])
td19 <- td19[,colun]
td <- final[,colun]
td <- td[rowSums(is.na(td)) == 0,]
mod <- tree(Win_Vis~., data=td[, -2])
probabilities <- predict(mod, newdata=td19[, -2])
predicted.classes <- ifelse(probabilities > 0.5,
                             "TRUE", "FALSE")
win <- predicted.classes == td19$Win_Vis
vet7 <- c(vet7, win)
}
end_time <- Sys.time()
tempo <- end_time - start_time
tempo
as.numeric(tempo)
mean(vet7)

#Reg Linear c/ forward
vet8 <- c()
start_time <- Sys.time()
for(i in 1:length(nomes)){
  full19 <- oi19==nomes[i]
  td19 <- teste19[full19,]
  colun <- !is.na(td19[1,])
  td19 <- td19[,colun]
  td <- final[,colun]
  td <- td[rowSums(is.na(td)) == 0,]
  modmin <- lm(result_Vis ~ 1, data=td[, -1])
  mod <- step(modmin, direction = "forward",
              scope=(as.formula(td[, -1])), trace=0)
  win <- (predict(mod, newdata=td19[, -1]) > 0) ==
        (td19$result_Vis > 0)
  vet8 <- c(vet8, win)
}
end_time <- Sys.time()
tempo <- end_time - start_time
tempo
as.numeric(tempo)*60
mean(vet8)

#Reg Logistica c/ forward
vet9 <- c()

```

```

start_time <- Sys.time()
library(dplyr)
for(i in 1:length(nomes)){
  full19 <- oi19==nomes[i]
  td19 <- teste19[full19,]
  colun <- !is.na(td19[1,])
  td19 <- td19[,colun]
  td <- final[,colun]
  td <- td[rowSums(is.na(td)) == 0,]
  modmin <- glm(Win_Vis~1, data = td[, -2],
                family=binomial(link = "logit"))
  a <- step(modmin, direction = "forward",
            scope=(as.formula(td[, -2])), trace=0)
  probabilities <- a %>% predict(td19[, -2], type = "response")
  predicted.classes <- ifelse(probabilities > 0.5,
                              "TRUE", "FALSE")
  win <- predicted.classes == td19$Win_Vis
  vet9 <- c(vet9, win)
}
end_time <- Sys.time()
tempo <- end_time - start_time
tempo
as.numeric(tempo)*60*60
mean(vet9)

#Reg Probit c/ forward
vet10 <- c()
start_time <- Sys.time()
library(dplyr)
for(i in 1:length(nomes)){
  full19 <- oi19==nomes[i]
  td19 <- teste19[full19,]
  colun <- !is.na(td19[1,])
  td19 <- td19[,colun]
  td <- final[,colun]
  td <- td[rowSums(is.na(td)) == 0,]
  modmin <- glm(Win_Vis~1, data = td[, -2],
                family=binomial(link = "probit"))
  a <- step(modmin, direction = "forward",
            scope=(as.formula(td[, -2])), trace=0)
  probabilities <- a %>% predict(td19[, -2], type = "response")
  predicted.classes <- ifelse(probabilities > 0.5,

```

```
                                "TRUE", "FALSE")
  win <- predicted.classes == td19$Win_Vis
  vet10 <- c(vet10,win)
}
end_time <- Sys.time()
tempo <- end_time - start_time
tempo
as.numeric(tempo)*60*60
mean(vet10)
```