



iscte

INSTITUTO
UNIVERSITÁRIO
DE LISBOA

Projeto Aplicado a Ciência de Dados
Ano Letivo 2023/2024 | 2º semestre
Projeto I

Docentes: Sérgio Moro, Diana Aldea Mendes

Grupo 7:

Diogo Aqueu – 110705

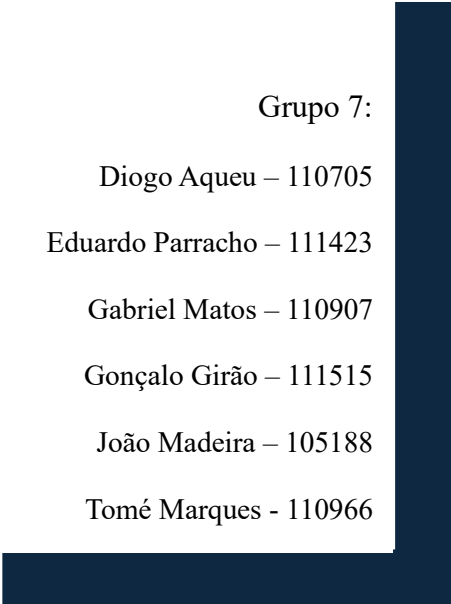
Eduardo Parracho – 111423

Gabriel Matos – 110907

Gonçalo Girão – 111515

João Madeira – 105188

Tomé Marques - 110966



INDICE

INTRODUÇÃO	2
METODOLOGIA CRISP-DM	2
BUSINESS UNDERSTANDING	3
DATA UNDERSTANDING	5
DATA PREPARATION.....	7
SQL	7
JUPYTER.....	11
GRÁFICOS E VISUALIZAÇÃO DOS DADOS.....	11
VISUALIZAÇÃO DAS VARIÁVEIS EM RELAÇÃO AOS NÚMEROS DE SETS.....	17
MODELLING	21
ALGORITMOS	21
MODIFICAÇÕES PRÉ-MODELAÇÃO	21
CORRELAÇÃO DE VARIÁVEIS E MULTICOLINEARIDADE	22
MODELOS	24
<i>Modelo1</i>	24
<i>Modelo2</i>	25
EVALUATION	27
DEPLOYMENT.....	29
CONCLUSÃO.....	31

Introdução

Este relatório tem como objetivo criar um modelo de previsão para determinar se um jogo de ténis profissional nos Países Baixos será concluído em 2 ou 3 sets. A previsão do número de sets é uma ferramenta importante para diversas partes interessadas no mundo do ténis, incluindo organizadores de torneios, patrocinadores, analistas, treinadores e atletas.

Para os organizadores de torneios, a capacidade de prever a duração dos jogos em termos de sets pode otimizar a gestão e rotação dos campos, assegurando um uso mais eficiente das instalações. Isto é particularmente útil em eventos com múltiplas partidas programadas ao longo do dia, permitindo uma melhor organização do calendário e minimizando atrasos. Os patrocinadores podem utilizar essas previsões para selecionar quais jogos quererão patrocinar, maximizando a exposição e o retorno sobre o investimento.

Analistas, treinadores e atletas podem beneficiar do modelo para estudar jogos passados e melhorar as suas estratégias de preparação. A análise dos padrões e fatores que influenciam a duração dos jogos pode fornecer insights valiosos, auxiliando na preparação tática e física dos jogadores. Além disso, um modelo preditivo robusto pode ser utilizado para ajustar estratégias em tempo real durante um torneio. A metodologia utilizada para a criação deste modelo de previsão segue um modelo CRISP-DM.

Metodologia Crisp-DM

O CRISP-DM (*Cross Industry Standard Process for Data Mining*) é um modelo amplamente usado na análise de dados em projetos de Ciência de Dados. A sua estrutura é simples de aplicar e extremamente valiosa, pois é guia de forma sequencial por todas as etapas essenciais, garantindo uma análise de projeto abrangente e detalhada. O modelo consiste em seis fases distintas que ajudam a garantir uma análise de projeto robusta e completa: *business understanding*, *data understanding*, *data preparation*, *modelling*, *evaluation* e por fim *deployment*, que neste projeto não será um dos objetivos prioritários.

Business Understanding

Para a realização deste trabalho é necessário perceber como funcionam os jogos de ténis e os torneios.

No ténis, os jogos são divididos em sets. Um jogador precisa de ganhar 6 jogos primeiro que o adversário para vencer um set, desde que tenha uma diferença de pelo menos 2 jogos em relação ao seu oponente. Se os jogadores estiverem empatados ao fim de 10 jogos, isto é um resultado de 5-5, eles continuam a jogar até que o resultado fique 7-5 ou 5-7, e consequentemente vença o set, caso o resultado permaneça empatado após o décimo segundo jogo, isto é um resultado de 6-6, o jogo vai a *tiebreak*, onde o primeiro jogador a alcançar 7 ou mais pontos e com mais dois pontos que o oponente no resultado do *tiebreak* ganha o set.

A maioria dos torneios de ténis disputam-se à menor de três sets, no entanto existem 5 torneios, os *Grand Slams*, onde os jogadores competem numa série à menor de cinco sets. Conforme avançam nas fases eliminatórias do torneio, os jogadores vão vencendo jogos até chegarem à final, onde o vencedor é coroado campeão do torneio.

O objetivo deste trabalho é prever o número de sets necessários para finalizar um jogo de ténis profissional nos Países Baixos, em concreto se um jogo será concluído com 2 ou 3 sets, pois é um país onde os torneios são todos disputados a menor de três sets, não existindo portanto Grand Slams, como será analisado no decorrer do projeto. Para isso, ir-se-ão estudar os padrões históricos dos jogos realizados nesse país, considerando diversos fatores que podem influenciar o desempenho dos jogadores.

Os fatores a serem analisados incluem o país de origem dos jogadores em relação ao clima local dos Países Baixos.

Outro aspeto importante é o tipo de superfície dos “courts”, que pode ter um impacto significativo nos resultados dos jogos. Além disso, examinar-se-á detalhadamente o perfil dos jogadores, incluindo os seus rankings, experiências anteriores e comparações com os seus adversários, para determinar como esses fatores podem influenciar o resultado final da partida, e por sua vez o número de sets.

A análise abrangente desses fatores fornecerá “insights” valiosos que podem ajudar diversos interessados, como treinadores e jogadores a adaptarem as suas estratégias de treinamento e táticas de jogo ao enfrentarem determinados oponentes.

Data Understanding

O dataset contido no ficheiro “atpplayers.json” apresenta dados recolhidos do site oficial ATP, totalizando um total de 10361 jogadores (masculinos) com jogos entre os anos de 1973 e 2022. Sendo que o ficheiro contém todos esses dados e o objetivo do trabalho ser relativo apenas aos Países Baixos, filtraram-se os dados para conter apenas os jogos realizados nos Países Baixos.

No decorrer do estudo foi também importado o csv “world.cities.csv” para uma maior compreensão dos possíveis torneios pertencentes ao país em questão.

Nesta etapa são estudadas as variáveis e a estrutura da base de dados. A base de dados tem 1308835 registos e 16 variáveis.

Atributos	Conteúdo
_id	identificador de linha
PlayerName	Nome do jogador
Born	Local de nascimento do jogador
Height	Altura do jogador
Hand	Mão dominante do jogador
LinkPlayer	Link identificador de cada jogador
Tournament	Nome do torneio
Location	Localização do torneio
Date	Data do torneio
Ground	Tipo de campo
Prize	Prémio do torneio
GameRound	Etapa do torneio

GameRank	Ranking do jogo
Oponent	Nome do oponente
WL	Jogo ganho (W) ou jogo perdido (L)
Score	Resultado do jogo

Table 1

Sendo que o objetivo é criar um modelo de previsão do número de sets necessários para a conclusão de um jogo de ténis profissional nos Países Baixos, a variável alvo é “Score”. Esta variável interpreta-se da seguinte forma:

- 46 63 – [46] é 1 set com $(4 + 6) = 10$ jogos, em que o oponente ganhou o set por uma margem de 2 jogos ao jogador principal.
- Neste caso o primeiro número dos sets contém sempre o número de jogos ganhos pelo jogador principal.

Data Preparation

Com o objetivo de prever o número de sets necessários para a conclusão de um jogo de ténis profissional nos Países Baixos, iniciou-se pela limpeza dos dados. Em seguida, criaram-se novas variáveis e substituíram-se os valores omissos por valores reais. A visualização dos dados foi uma etapa crucial para identificar possíveis outliers que poderiam comprometer a precisão do modelo, diminuindo a sua eficiência.

O dataset inicial, que abrangia todos os países, era composto por 1.308.835 linhas e 16 colunas. As variáveis incluídas no dataset inicial eram “_id”, “PlayerName”, “Born”, “Height”, “Hand”, “LinkPlayer”, “Tournament”, “Location”, “Date”, “Ground”, “Prize”, “GameRound”, “GameRank”, “Opponent”, “WL” e “Score”, todas do tipo objeto.

A primeira fase de Data Preparation é realizada em MongoDB. Recorre-se ao MongoDB para a criação das coleções: “PlayerInfoFixed”, “TournamentsFixed” e “GamesFixed” para um melhor tratamento dos dados e eventual criação do modelo relacional.

SQL

Após a criação destas tabelas passa-se então à exportação destas para SQL. Usando o phpmyAdmin como ferramenta para o uso de SQL, criam-se as tabelas “Player”, “Tournaments” e “Games”. Posto isto, importaram-se os valores das tabelas criadas em MongoDB para as respetivas tabelas em SQL, assim como o ficheiro csv relativo às cidades.

Tabela Cidades/Filtragem de país

Foi criada uma tabela “CountryCity” para onde foi importada a base de dados “world-cities.csv”. Nesta tabela foram removidos todos os registos que não sejam no país Netherlands. Posto isto é criada na tabela “Tournaments” uma coluna “focusLocation”. Os dados na coluna “focusLocation” da tabela “Tournaments” foram, por fim, atualizados com base na correspondência entre o campo “Location” da tabela “Tournaments” e os dados da tabela “CountryCity”.

Criação de NSETS-Variável Alvo

Foi adicionada uma nova coluna chamada “nSETS”, à tabela “Games” para armazenar o número de sets de cada jogo. O número de sets de cada jogo foi calculado da seguinte maneira: Primeiro foi realizada uma remoção de caracteres não numéricos da pontuação (Score) e contou-se o número de dígitos resultantes da remoção dos caracteres não numéricos. Posto isto, dividiu-se este valor em dois, uma vez que cada Set consiste em duas pontuações (uma para cada jogador).

Correção da variável “Date” nas tabelas (criação das variáveis “date_inicial” e “date_final”)

Uma vez que a data de início e de fim do torneio se encontram na mesma variável, optou-se por criar duas novas variáveis a partir desta “date_inicial” e “date_final”, e optou-se por remover a variável “Date”. As datas são divididas pelo delimitador “-” e ambas as colunas são do tipo DATE, que é usado para armazenar datas no formato YYYY-MM-DD. Este processo é aplicado na tabela “Games” e na tabela “Tournaments”.

Correção da variável “Hand” nas tabelas

A variável “Hand” encontrava-se numa situação semelhante da variável “Date”, por isso sofreu alterações parecidas. Criaram-se duas variáveis: “MainHand” e “BackHand”, estas variáveis são divididas pelo delimitador “,”, onde a parte antes da virgula é adicionada à coluna “MainHand” e a parte depois da virgula é adicionada à coluna “BackHand”.

Criação das variáveis “RankDiff” e “Tiebreak”

Foi adicionada a coluna “RankDiff” à tabela “Games”, que é obtida através da diferença entre “HomeRank” e “GameRank”. Foi também criada a coluna “Tiebreak” que é formada com valores binários (1 caso haja Tiebreak e 0 caso não haja).

Correção da variável “Prize”

Nesta variável, foram substituídos os símbolos que não eram reconhecidos por “\$” e foram especificados os jogos/torneios prémios que não tinham prémio. O valor de prémio não é relativo à ronda específica, mas sim ao prémio final do torneio e se encontra em Dólares.

Criação da variável “RoundNumber”

Nesta variável, de forma a facilitar a compreensão, cada ronda foi identificada por um número, de 0 a 9, sendo 9 a última ronda (Final) possível num torneio e 0 correspondente à Round Robin num torneio. A Round Robin é algo menos conhecido dentro dos torneios de ténis, esta situação consiste num formato que é utilizado em torneios não só de ténis, como também doutros tipos de desportos, este formato visa garantir que cada participante joga contra todos os outros pelo menos uma vez.

A identificação foi então a seguinte:

Final	9
Semi-final	8
Quartos de final	7
Ronda de 16	6
Ronda de 32	5
Ronda de 64	4
3ª ronda de qualificação	3
2ª ronda de qualificação	2
1ª ronda de qualificação	1
Round Robin	0

Table 2 – Codificação das Rondas

Criação das variáveis Vencedor Casa e Derrotado Casa

De forma a facilitar futuras análises, foi criada a variável VencedorCasa, onde se verifica quais os jogadores holandeses que vencerem os seus jogos nos Países Baixos e a variável “DerrotadoCasa”, onde se verifica os jogadores holandeses que perderam os seus jogos em casa.

Criação da Variável “Ground”

Nesta variável, de forma a facilitar a compreensão, a cada um dos quatro tipos de chão foi atribuído um número de 1 a 4.

Criação da variável “Media_Sets”

Com o objetivo de prever o número de sets necessários para a conclusão de um jogo em vista, foi calculada a média de sets necessários para a conclusão dos jogos anteriores, jogados nos

Países Baixos e entre os mesmos dois jogadores, o que nos pode dar uma melhor expectativa do número de sets do próximo encontro entre ambos.

Criação da variável RankCode

Foi criada a coluna RankCode de forma a facilitar a interpretação do RankDif. Se Rankdif for menor ou igual a 10, RANK_CODE é 'close'; se Rankdif estiver entre 11 e 50, RANK_CODE é 'medium far'; se Rankdif for maior que 50, RANK_CODE é 'far'. Esta variável pode nos oferecer uma expectativa mais realista para a competição em relação ao resultado, uma vez que um jogo com, por exemplo, RANK_CODE 'far' pode ser um bom indicativo para o encontro ser menos equilibrado, e por sua vez ser disputado em apenas dois sets vencendo o jogador com melhor rank(ou seja posicionado num rank inferior).

Correção da Variável Height

Primeiramente, foram identificados registos na tabela 'player' onde a altura dos jogadores estava registada como 'NA' ou zero. Estes valores são inválidos porque 'NA' não representa uma altura real e zero não é possível como valor de altura. Os registos com 'NA' na coluna height foram atualizados para NULL, indicando que a altura é desconhecida ou não fornecida. Da mesma forma, os registos com zero na coluna height também foram atualizados para NULL. Foi verificado que alguns jogadores tinham alturas incorretas registadas como 15 cm ou 71 cm, que são valores extremamente baixos e claramente errados, para esses jogadores específicos, a altura foi corrigida manualmente. Grant Stafford teve a sua altura corrigida para 188 cm. James Trotter, teve a sua altura corrigida para 183 cm. Johannes Ingildsen teve a sua altura corrigida para 193 cm. Nathan Healey teve a sua altura corrigida para 180 cm e Sebastian Prechtel, que tinha 71 cm registado como altura, teve a sua altura corrigida para 185 cm.

Criação da Variável HeightDif

Após a correção da variável Height foi criada a variável HeightDif, que representa a diferença de altura entre os dois jogadores a competir.

Criação da Variável YEAR

Foi criada a variável YEAR de forma a conseguirmos obter apenas o ano em que foi realizado cada torneio.

Remover duplicados e erros

Após a filtragem por país, foi realizada uma transformação nos dados, eliminando os jogos em espelho e corrigindo erros, removendo-os ou substituindo-os pelos valores corretos.

Jupyter

Na passagem de SQL para Jupyter importamos a tabela ‘unique_games’ em formato csv e assim trabalhamos apenas com essa tabela por se considerar que a tabela tinha a informação necessária para a construção dos modelos.

Posteriormente foram consideradas apenas as variáveis numéricas e categóricas numéricas, de forma a ser possível construirmos os modelos de previsão.

Gráficos e Visualização dos Dados

Rank_Code

Como se pode observar, é mais frequente jogadores com uma grande diferença de ranks se enfrentarem. Isto pode se dever ao facto de que nas primeiras rondas dos torneios, que são também as com mais jogos a decorrer e consequentemente mais jogadores ser á base de um sorteio, o que leva a jogadores com uma grande diferença de qualidade a defrontarem-se. Os jogadores com ranks semelhantes encontram-se menos, já que é provável que apenas o façam nas fases mais avançadas do torneio, onde há menos jogos e jogadores, daí as chances serem menores.

	Distânciamento de Ranks	Frequência absoluta
0	Close	462
1	Far	6578
2	Medium far	1694

Tabela 1 – Tabela de frequência para distânciamento de Ranks entre os jogadores

RoundNumber

Como se pode observar, a ronda mais comum é a ronda 5, que corresponde aos 32 avos de final, o que é de esperar, uma vez que a maioria dos torneios começa nesta ronda e esta mesma tem um maior número de jogos. À medida que os o número das rondas vai aumentando e os

torneios vão progredindo, há cada vez menos jogos, já que os torneios são, na sua grande maioria a eliminar.

Ronda (Código de Rondas)	Frequência absoluta
Round Robin (0)	91
1ª ronda de qualificação (1)	478
2ª ronda de qualificação (2)	316
3ª ronda de qualificação (3)	110
Ronda de 64 (4)	21
Ronda de 32 (5)	3963
Ronda de 16 (6)	1995
Quartos de final (7)	1009
Semi-final (8)	511
Final (9)	240

Tabela 2 – Tabela de frequência para as várias rondas

NSETS-Alvo

De acordo com a tabela, existe uma maior frequência de jogos que terminam ao fim de dois sets. Isto acontece porque o número mínimo de sets necessários para um jogo terminar é 2 e só em caso de empate, de um set para cada jogador, é que é jogado um terceiro set, havendo assim uma condição específica para a existência de um terceiro set. Entende-se então a partir deste momento que os dados relativos à variável alvo não estão balanceados.

	Nº de sets	Frequência absoluta
0	2	5892
1	3	2842

Tabela 3 – Tabela de frequências sobre o número de sets dos nossos registros

Ground_Code

Os diversos tipos de superfícies dos campos de tênis têm um papel crucial no desempenho dos jogadores durante os jogos. Com base nessa premissa, foi elaborado um gráfico de barras para ilustrar a quantidade de jogos realizados em cada tipo de superfície. A análise dos resultados permite tirar algumas conclusões sobre a preferência e a disponibilidade dessas superfícies nos jogos estudados.

Observa-se que a superfície "Clay" é notavelmente a mais popular, apresentando um número significativamente maior de jogos em comparação com os outros tipos de terreno. Isto sugere que a terra batida é a superfície predominante nos torneios de tênis realizados nos Países Baixos. Já no que diz respeito as superfícies "Hard" e "Grass", registam um número consideravelmente menor de jogos. Este dado indica que estas superfícies são provavelmente apenas utilizadas por poucos torneios. No caso do "Carpet", foi o tipo de chão menos utilizado, o que permite concluir que há uma quantidade limitada de torneios com este tipo de superfície.

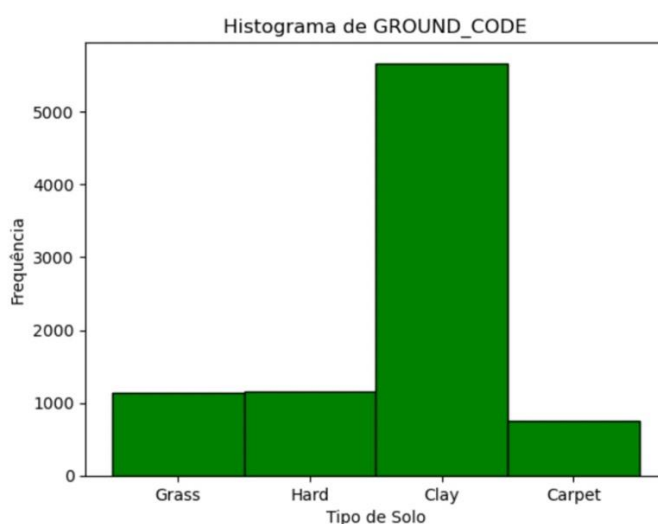


Figura 1 – Histograma da variável Ground_Code

PRIZE

O boxplot que se pode observar de seguida, demonstra-nos que a média é relativamente baixa o que indica algum equilíbrio nesta variável, por outro lado encontramos alguns outliers, superiores ao limite superior interquartis. Foi então decidido que, devido à existência de inúmeros Outliers, as variáveis dedicadas aos modelos passariam a ser as versões logaritmizadas das mesmas, de modo a não enviesar os resultados obtidos.

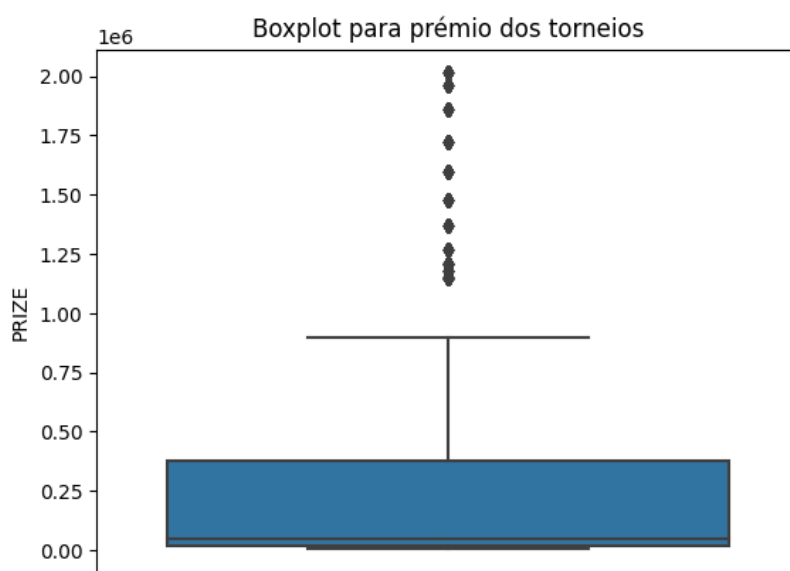


Figura 2 – Boxplot para a variável PRIZE

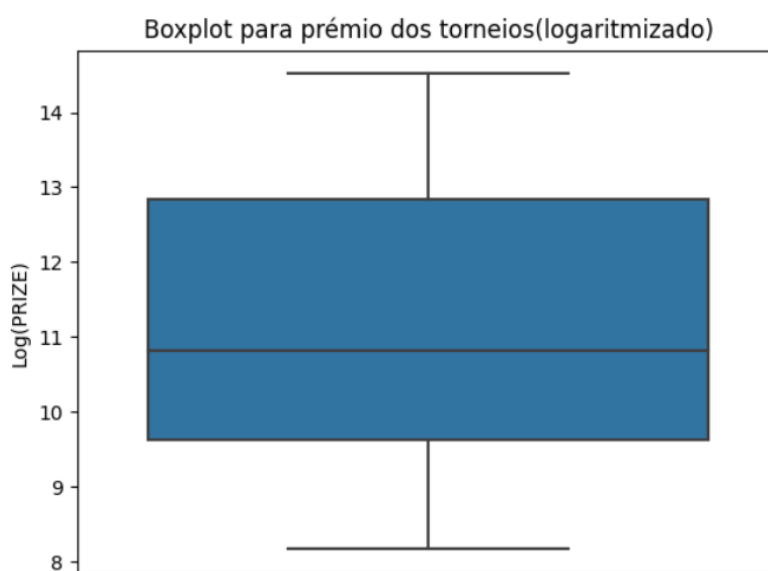


Figura 3 – Boxplot para a variável PRIZE logaritmizada

Foi então decidido que, devido à existência de inúmeros Outliers, a variável dedicada aos modelos passaria a ser a versão logaritmizada da mesma, de modo a não enviesar os resultados obtidos.

Como se pode observar no gráfico, os torneios nos Países Baixos têm prémios com valores muito baixos, excluindo ligeiras exceções. Isto pode acontecer devido ao facto de neste país não se disputarem Grand Slams ou outros torneios internacionalmente prestigiosos, o que leva a um menor financiamento e consequentemente prémios menores.

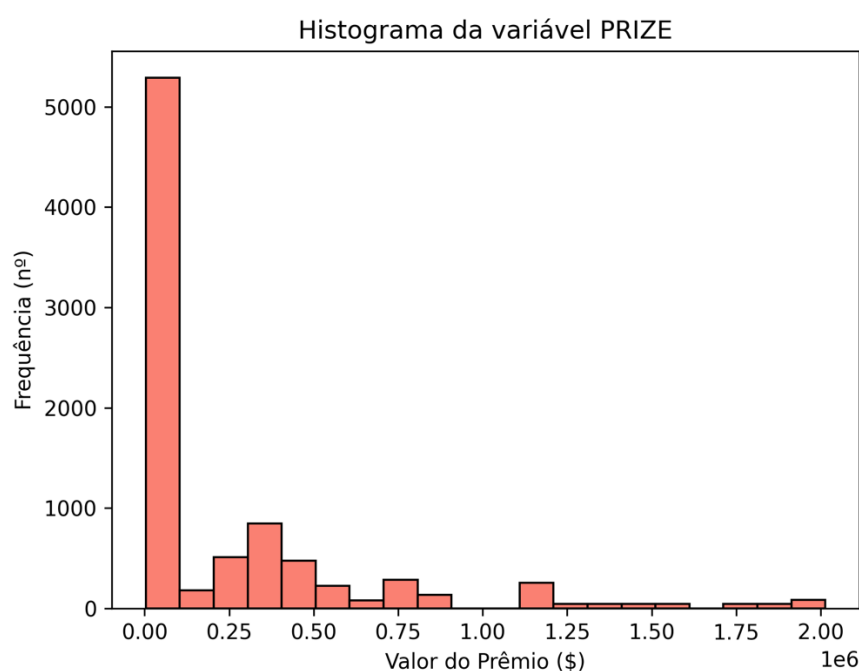


Figura 4 – Histograma da variável Prize

Torneios por ano

De acordo com o gráfico, só começaram a ser realizados torneios com elevada frequência a partir do ano de 2000, sendo realizados muitos mais torneios de 2000 a 2022 do que de 1968 a 1999. Isto pode ser produto do aumento da popularidade do desporto no século XXI como também do desenvolvimento e investimento no desporto por parte dos Países Baixos. Podemos verificar ainda uma redução drástica no número de torneios em 2020 e apanhando os anos até 2022 devido ao covid-19.

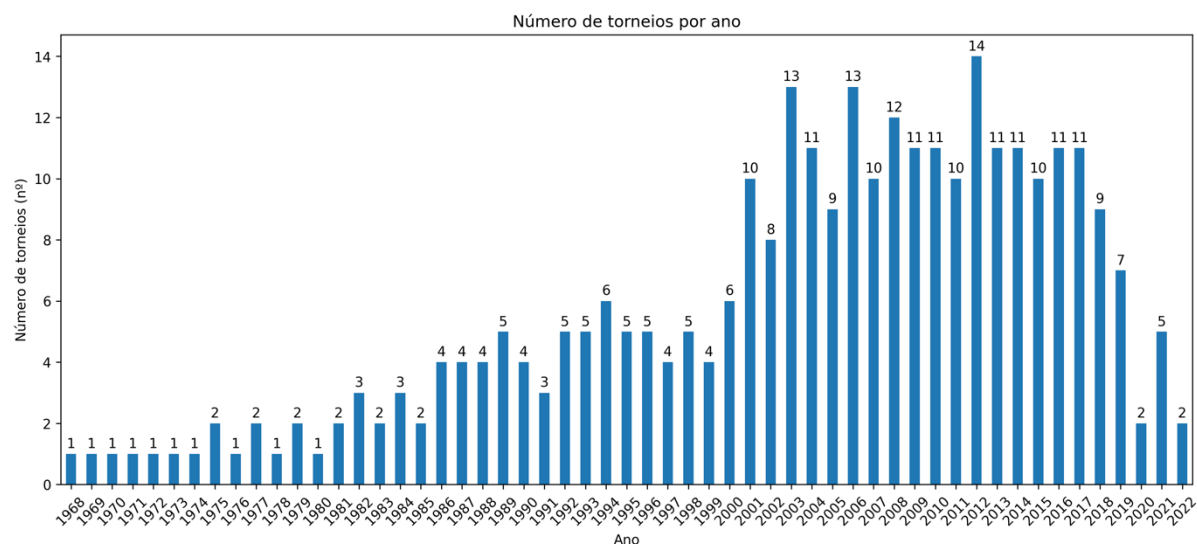


Figura 5 – Número de torneios realizados por ano

HeightDif

Num desporto como o ténis a altura de um jogador faz toda a diferença, pois quanto mais alto for um jogador em regra tem um melhor serviço, o que dificulta ao adversário ganhar jogos no seu serviço. O boxplot que se pode observar de seguida, demonstra-nos que a média é relativamente baixa o que indica algum equilíbrio nesta variável, por outro lado encontramos alguns outliers, superiores ao limite superior interquartis, isto é, superiores a 20cm o que já se pode tornar desvantajoso para o jogador com estatura mais baixa, podendo-se até prever desequilíbrio no jogo e numa última instância a ocorrência de apenas dois sets no encontro, ao invés de três.

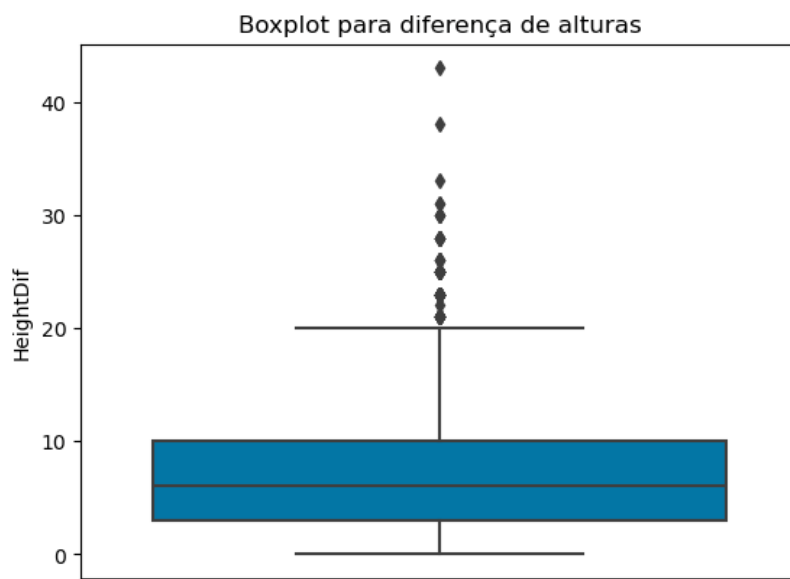


Figura 6 – Boxplot para a variável HeightDif (diferença de alturas entre os jogadores)

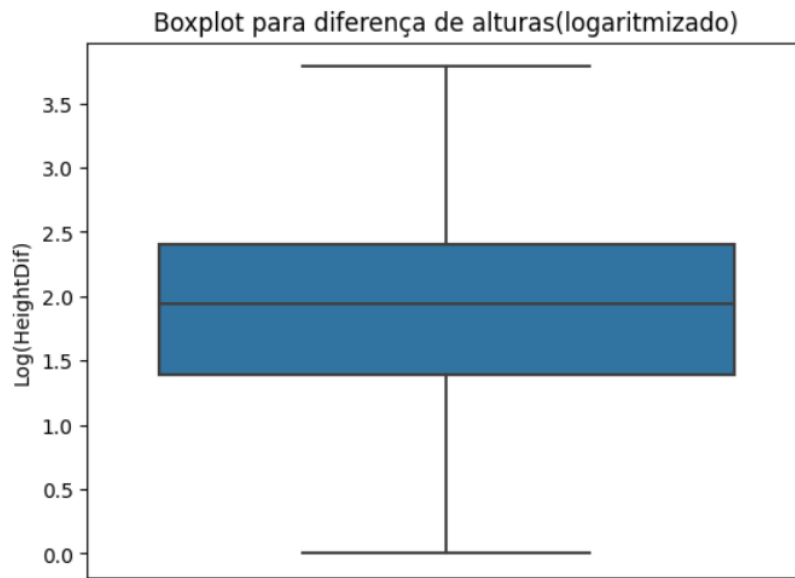


Figura 7 – Boxplot para a variável HeightDif logaritmizada

Foi então decidido que, tal como já foi feito noutra variável, devido à existência de inúmeros Outliers, a variável dedicada aos modelos passaria a ser a versão logaritmizada da mesma, de modo a não enviesar os resultados obtidos.

Visualização das variáveis em relação aos números de sets

Número de Sets por Rank_Code

Como se pode observar no gráfico, quanto maior é a diferença de rank, menor é a frequência de jogos com 3 sets. O contrário acontece quando a diferença de rank é menor. Isto deve-se ao facto de que quanto menor a diferença de rank, mais renhido é o jogo, aumentando assim a probabilidade de um jogo ter 3 sets.

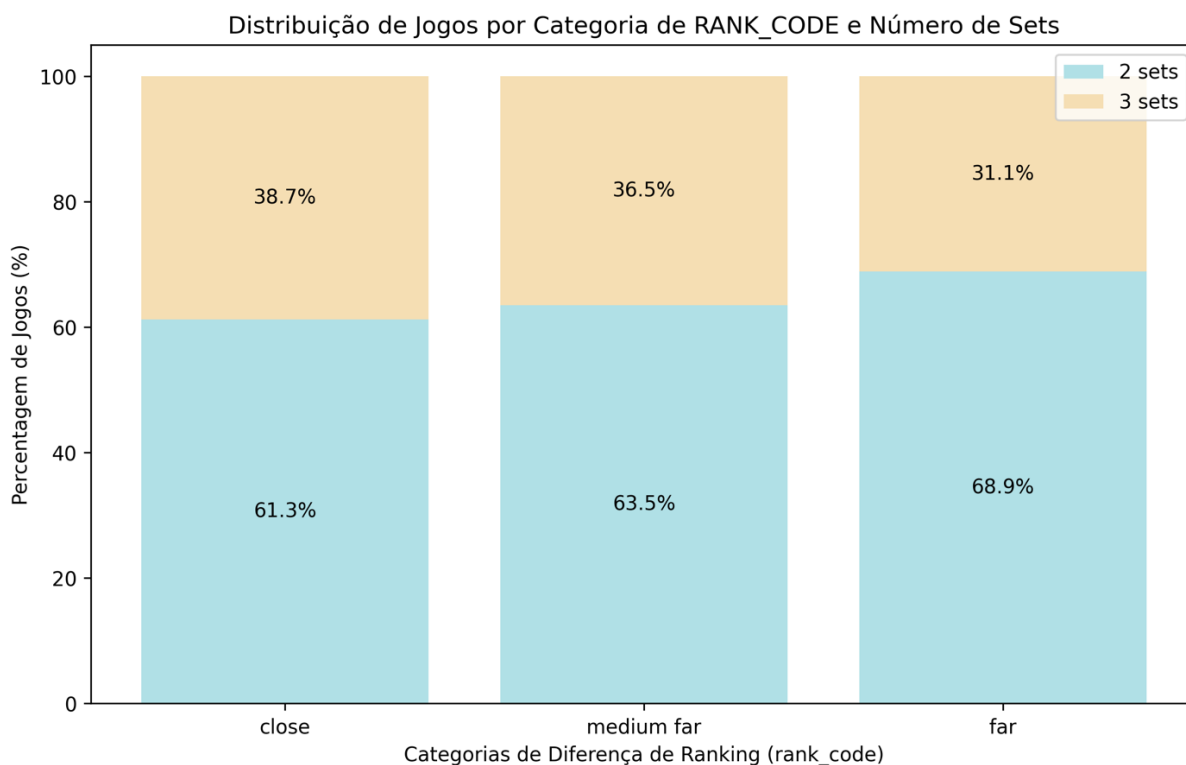


Figura 8 – Distribuição dos jogos por categoria de diferença de rank, dividido pelo número de sets

Número de Sets por Ronda

No seguinte gráfico pode-se analisar a relação entre as rondas de um torneio e o número de sets.

Nos primeiros jogos, a diversidade de habilidades entre os jogadores é maior, resultando em muitos jogos decididos em 2 sets. No entanto, há um número significativo de jogos em 3 sets, refletindo partidas competitivas onde jogadores de nível mais próximo se enfrentam.

Nas rodadas intermédias, a distribuição flutua, mas há uma tendência para um aumento gradual de jogos decididos em 2 sets. Isto pode ser devido a jogadores mais fortes se destacarem e vencerem com mais facilidade até se encontrarem com adversários de nível similar nas rondas finais.

Nas finais, a alta percentagem de jogos decididos em 3 sets (42.9%) indica que os jogadores finalistas são geralmente equilibrados em termos de habilidades, resultando em partidas mais longas e disputadas.

A relação entre as rondas e o número de sets mostra uma tendência de maior competitividade nas rondas iniciais e finais, com uma ligeira diminuição nas rondas intermédias.

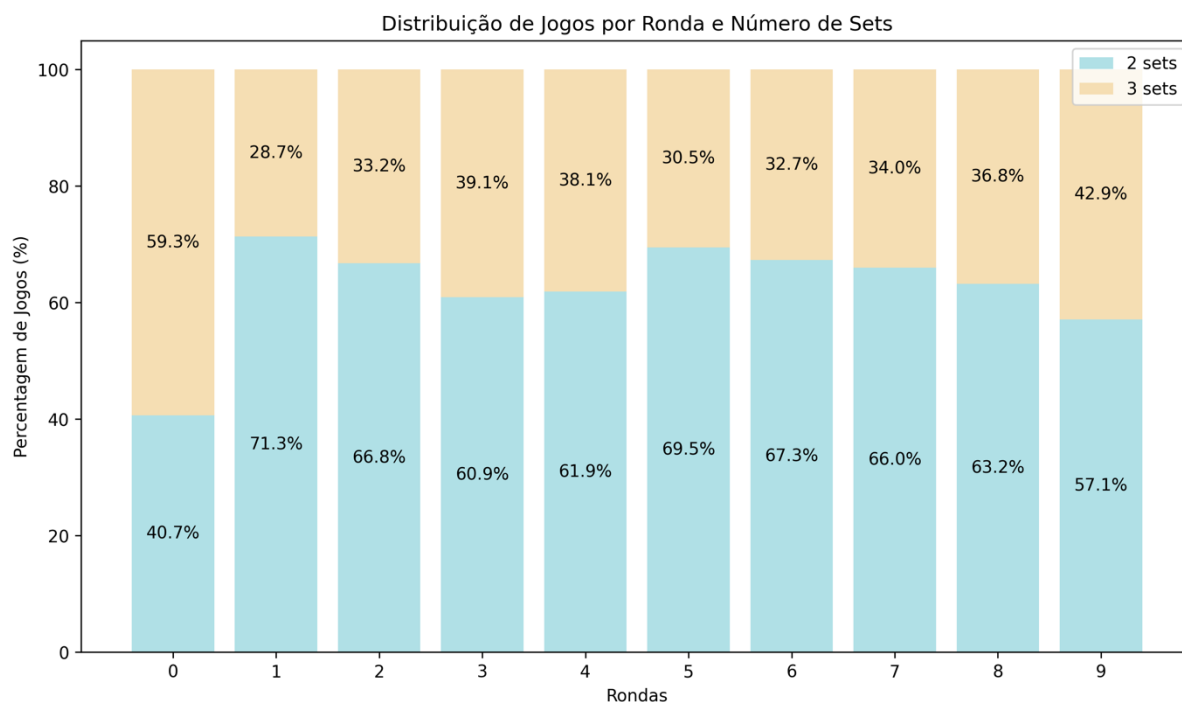


Figura 9 – Distribuição dos jogos por rondas, divididos por número de sets

Número de Sets por Ano

O gráfico que se segue, mostra-nos as percentagens de jogos finalizados em 2 e 3 sets para cada ano, as percentagens do número de sets vão oscilando de 1968 até 1984, que pode ser explicado devido ao reduzido número de torneios realizados nesses anos e posteriormente um menor número de jogos daí observar-se maior variabilidade, mas de 1984 até 2022 as percentagens mantem-se praticamente constantes.

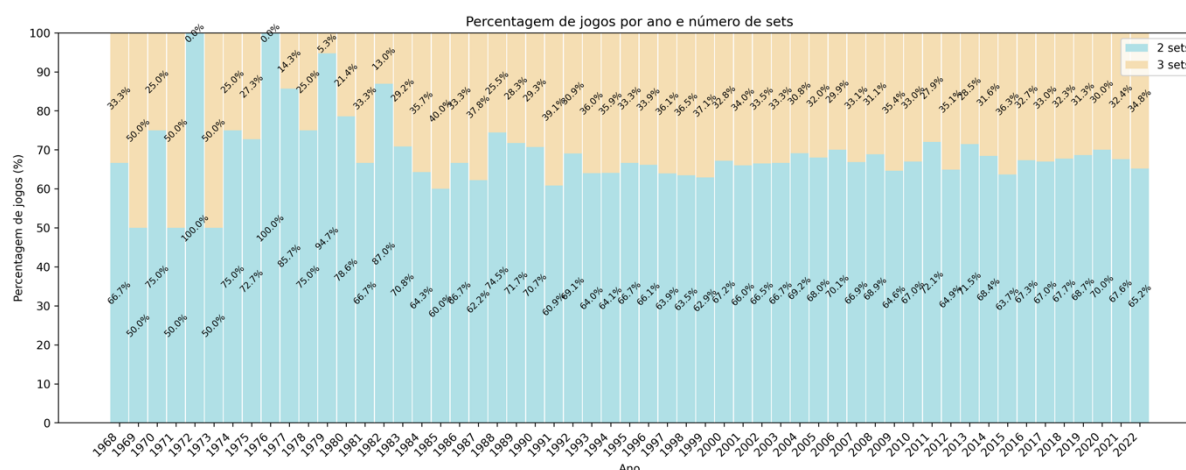


Figura 10 – Distribuição de jogos por ano, divididos por número de sets

Número de Sets por Ground

A relação entre o tipo de chão e o número de sets mostra como diferentes superfícies afetam a duração das partidas no tênis. Superfícies rápidas como grass e carpet tendem a ter uma maior percentagem de jogos decididos em 2 sets, devido à velocidade do jogo e a pontos mais curtos. Superfícies mais lentas como a clay e hard levam a jogos mais longos, embora ainda apresentem uma maioria de jogos decididos em 2 sets.

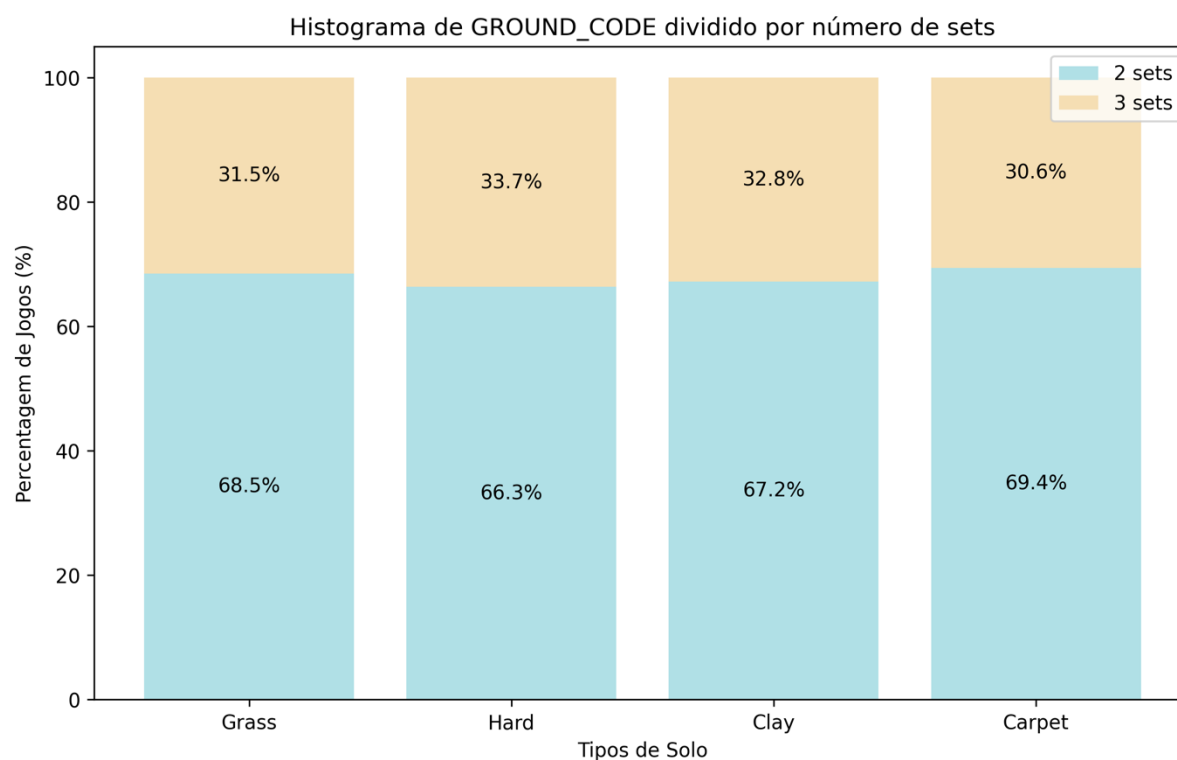


Figura 11 – Distribuição do número de jogos por tipo de solo, divididos por número de sets

Modelling

Foram nesta fase, criados vários modelos de classificação, utilizando as variáveis já existentes ou criadas/modificadas, incluindo tanto variáveis de tipo numérico como categórico.

Algoritmos

Previamente à utilização dos modelos, abordaremos todos os algoritmos utilizados de modo a serem entendidos e utilizados com vista a obter os melhores resultados, não só em relação a este dataset como a possíveis outras utilizações que poderia ter o modelo.

- *Random Forest*: É um algoritmo que combina as *Decision Tree's* e *Bagging* de forma a obter uma maior robustez nos modelos. Consiste na seleção aleatória e com reposição de amostras a partir do conjunto de treino. O modelo é treinado para cada uma dessas amostras, para o efeito, foram utilizados 10 *folds* para treinar o modelo.
- *Decision Tree*: Construídas a partir do conjunto de treino funciona dividindo iterativamente os dados em subconjuntos com base numa variável que oferece a maior distinção entre as classes.
- *KNN*: Classifica uma nova amostra com base na classe de K pontos mais próximos no conjunto de treino segundo uma medida de distância euclidiana.
- *Logistic Regression*: Algoritmo mais simples que os anteriores, calcula a probabilidade de uma ocorrência pertencer a uma determinada categoria através da função logística.
- *Naive Bayes*: Estima a probabilidade de pertencer a uma classe com base nas probabilidades condicionadas das características utilizando o produto das probabilidades condicionadas para calcular a probabilidade conjunta.
- *Gradient Boosting*: Vai iterativamente melhorando os erros anteriores, usando o gradiente descendente para minimizar a função de perda no conjunto de treino. Procura ajustar a direção que vai reduzindo os erros residuais.

Modificações Pré-Modelação

Depois de todas as alterações anteriormente verificadas foram excluídos os jogos onde os jogadores se tinham retirado, assim como substituídos os valores omissos que poderiam impedir os modelos de realizar boas classificações, foram também utilizados os valores logaritmizados das variáveis “HeightDif” e “Prize” devido à existência de *outliers*, e, posteriormente realizada a divisão do dataset em conjuntos de treino e teste com a ponderação 70/30, de forma estratificada, com o objetivo de manter a proporção da variável alvo tanto no

conjunto de treino como no conjunto de teste, sabendo de antemão que este conjunto de dados era desbalanceado.

No entanto, devido ao desequilíbrio entre as classes de variáveis, onde uma das classes é significativamente menos representada do que outra (valor 3 em relação ao valor 2 na variável alvo) e visto que este desequilíbrio pode prejudicar o desempenho do modelo, pois pode-se tornar tendencioso em favor da classe majoritária, utilizámos o *SMOTE*. O *SMOTE* é uma técnica que gera novas amostras sintéticas da classe em minoria, criando pontos de dados que são combinações interpoladas das amostras existentes da classe minoritária. Causando o aumento do número de observações da classe minoritária, tornando o conjunto de dados mais balanceado.

Dessa forma, perante um conjunto de treino inicialmente de 6113 observações com apenas 1989 observações com o valor 3 na variável alvo, foi atualizado para um conjunto de treino de 8248 observações após o uso de *SMOTE*, totalizando 4124 observações no valor 3 na variável alvo, tal como o número de observações do valor 2.

Correlação de variáveis e multicolinearidade

O passo seguinte e anterior a verificações relativas a correlações entre variáveis e possível multicolinearidade foi a identificação/divisão de tipos de variáveis. Nesse sentido, foram identificadas como numéricas as variáveis 'RANKDIF', 'PRIZE_log', 'HeightDif_log', 'YEAR' e 'nSETS', 'VENCEDOR_CASA', 'DERROTADO_CASA', 'roundnumber', 'tiebreak', 'GROUND_CODE', 'media_sets' como variáveis categóricas.

Assim, foram criados dois DataSets, um para guardar as variáveis numéricas e outro que guardasse as categóricas, sendo eles 'data_numerico' e 'data_categorico', respetivamente.

De seguida foram verificadas as correlações entre variáveis numéricas (correlação de Pearson) e retiradas algumas conclusões:

- Entre as variáveis 'PRIZE_log' e 'HeightDif_log' ocorre a maior correlação (em valor absoluto), no entanto insuficiente para se tornarem incompatíveis.
- Entre as variáveis 'PRIZE_log' e 'RANKDIF' também existe um valor de correlação elevado, mas de novo não demonstra incompatibilidade na utilização de ambos no modelo.

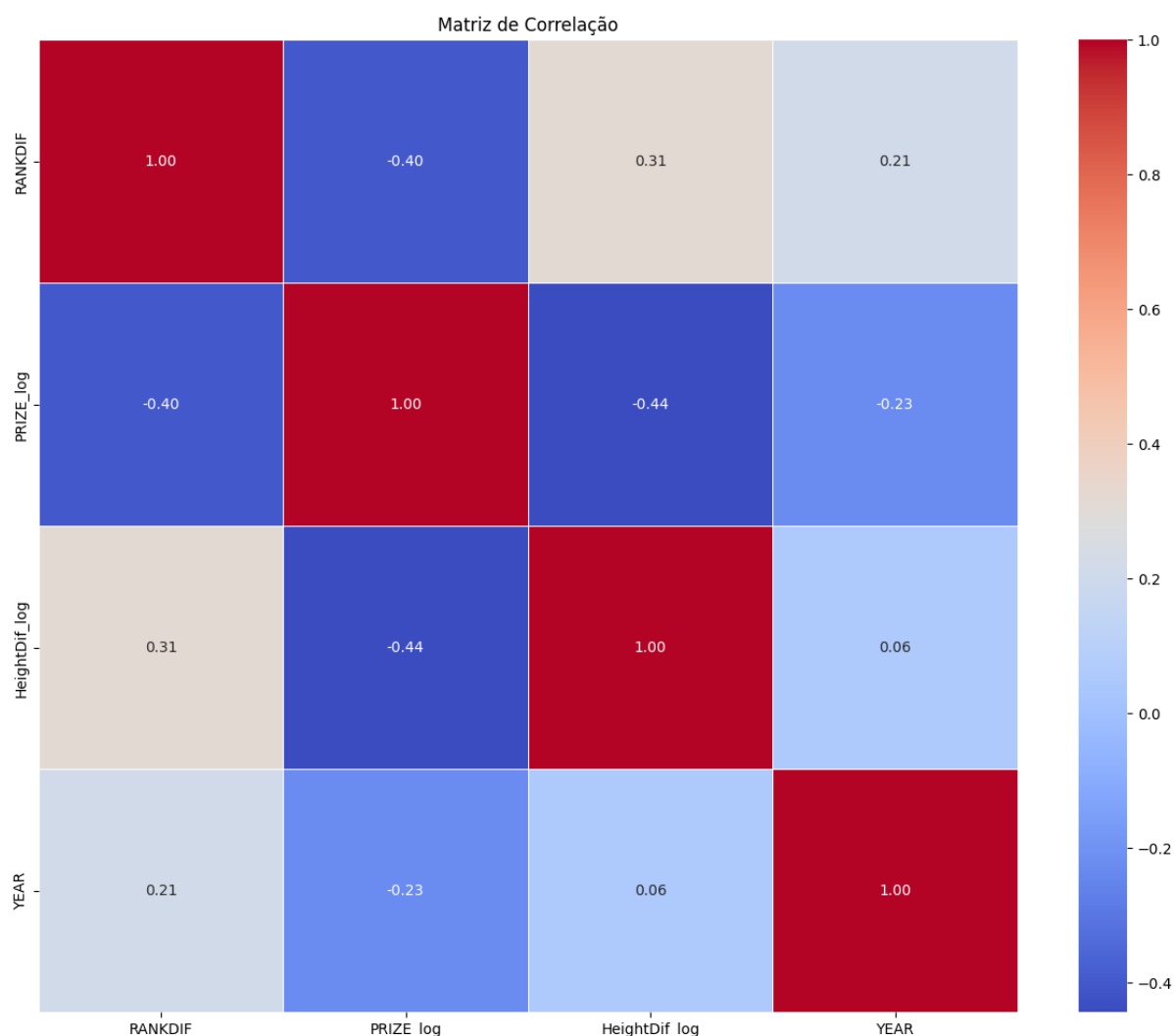


Figura 12 – Matriz de correlação para as variáveis numéricas

Estes valores de correlação mais elevados que os restantes, poderão ser explicados pela existência de variáveis logaritmizadas, pois estas, na sua criação linearizam relações não lineares, assim como reduzem a variância existente, tal como a influência de outliers.

Apesar disso, a segunda correlação negativa apresentada poderá ter alguma explicação, sendo que valores de prémio mais elevados corresponderão a diferenças de rank entre jogadores mais reduzidas, ou seja eventualmente, jogadores com um melhor Ranking, que pressupõem supostos melhores jogadores. Consequentemente, prémios de menor valor, corresponderão a diferenças de Ranking mais elevadas que poderá pressupor jogadores de pior nível.

	nSETS	VENCEDOR_CASA	DERROTADO_CASA	roundnumber	\
nSETS	1.000000	0.067390	0.061016	0.072619	
VENCEDOR_CASA	0.067390	1.000000	0.072162	0.104795	
DERROTADO_CASA	0.061016	0.072162	1.000000	0.084545	
roundnumber	0.072619	0.104795	0.084545	1.000000	
tiebreak	0.062780	0.005440	0.038827	0.066476	
GROUND_CODE	0.315155	0.185772	0.109812	0.173144	
media_sets	0.170325	0.192051	0.123589	0.088500	
	tiebreak	GROUND_CODE	media_sets		
nSETS	0.062780	0.315155	0.170325		
VENCEDOR_CASA	0.005440	0.185772	0.192051		
DERROTADO_CASA	0.038827	0.109812	0.123589		
roundnumber	0.066476	0.173144	0.088500		
tiebreak	1.000000	0.238085	0.144663		
GROUND_CODE	0.238085	1.000000	0.530809		
media_sets	0.144663	0.530809	1.000000		

Figura 13 – Matriz de correlação para as variáveis categóricas (V de Cramer)

Perante as correlações relativas às variáveis categóricas, os valores são reduzidos e também não expliquem qualquer incompatibilidade, finalizando este processo sem a exclusão de qualquer variável.

Com vista a entender se existe alguma incompatibilidade relativa à multicolinearidade entre variáveis, foi realizado o critério VIF, que indica multicolinearidade a partir do valor 5. Perante esse valor, entende-se que também não existe multicolinearidade no conjunto de dados existentes.

Modelos

Previamente, foram realizados outros modelos com um balanceamento distinto no conjunto de treino e com outros valores de K divisões diferentes de 10, no entanto os resultados incluídos eram pobres e o grupo decidiu excluir por completo os modelos.

Modelo 1

No primeiro modelo foi decidido incluir todas as variáveis, sendo que a variável de maior importância foi ‘RANKDIF’ com 33%. Foram, todos os algoritmos realizados com 10 *folds* e criadas curvas ROC para todos os algoritmos em questão, obtendo-se os resultados seguintes:

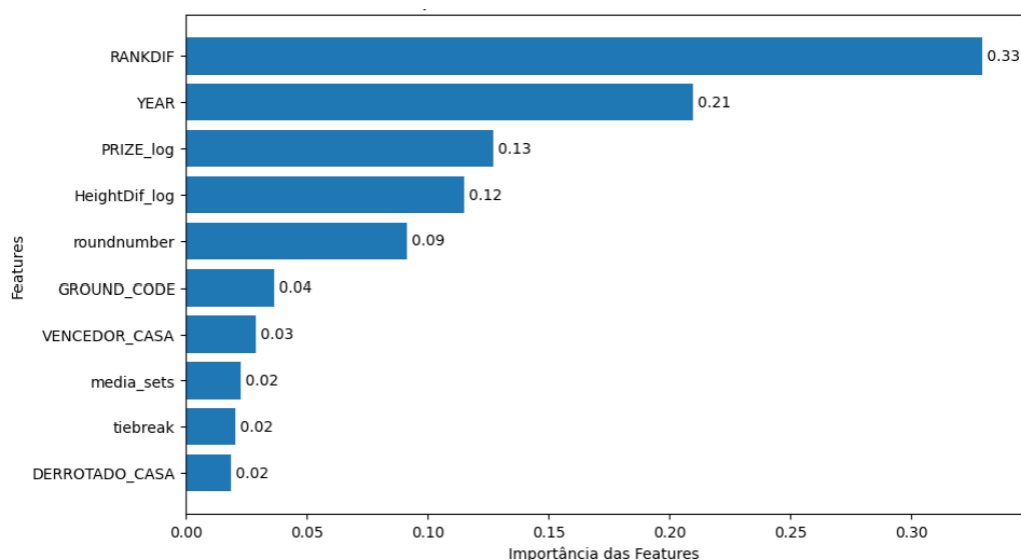


Figura 14 – Importância das features para o modelo 1

Modelo	AUC	Accuracy	Precision	Recall	F1-Score
DecisionTree	0.50	0.57	0.57	0.57	0.57
RandomForest	0.52	0.63	0.57	0.63	0.58
K-Nearest Neighbors	0.51	0.61	0.57	0.61	0.58
Logistic Regression					
Naive Bayes	0.54	0.66	0.57	0.66	0.57
GradientBoosting	0.54	0.68	0.63	0.68	0.56

Tabela 4 - Resultados MODELO1

Modelo2

Imediatamente após as visualizações acima, foi entendido que existem variáveis com muito menor importância que outras. Dessa forma, foram excluídas algumas delas para a construção do modelo2, utilizando assim apenas as 5 variáveis de maior importância. Para um algoritmo como o KNN, não será à partida benéfica a escolha de 5 variáveis porque todas teriam a mesma importância no modelo anterior, no entanto para ficar em concordância com outros modelos foi também atualizada a lista de variáveis. Como seriam de esperar, as variáveis com maior importância no modelo anterior foram escolhidas como as 5 a incluir no modelo atual.

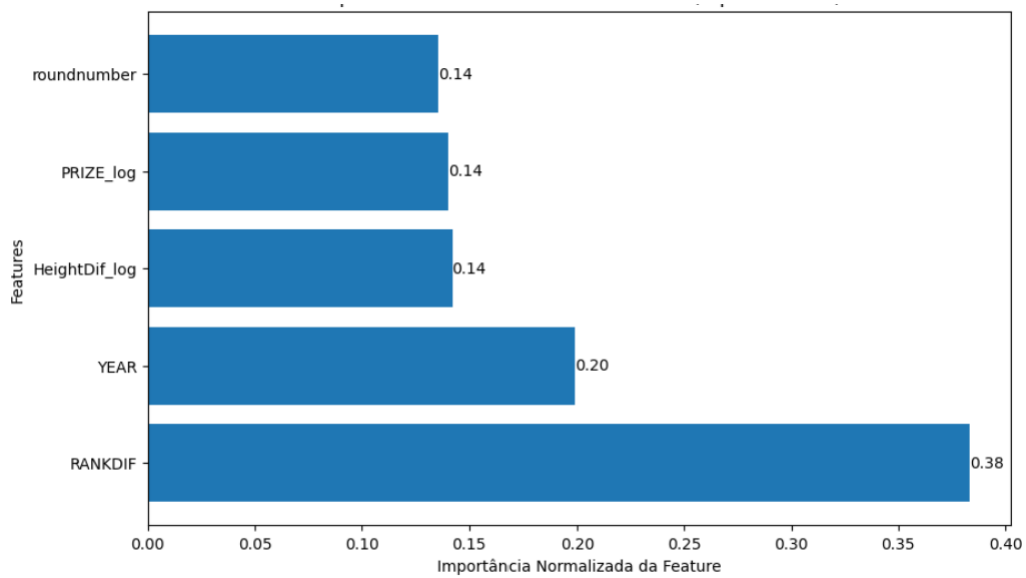


Figura 15 – Importância das features para o modelo 2

Modelo	AUC	Accuracy	Precision	Recall	F1-Score
DecisionTree	0.50	0.58	0.58	0.58	0.58
RandomForest	0.52	0.63	0.58	0.63	0.59
K-Nearest Neighbors	0.51	0.61	0.57	0.61	0.58
Logistic Regression					
Naive Bayes	0.54	0.67	0.59	0.67	0.55
GradientBoosting	0.54	0.67	0.61	0.67	0.56

Tabela 5 - Resultados MODELO2

Evaluation

Relativamente aos resultados existentes nos modelos podemos retirar conclusões para cada um dos algoritmos:

- *Logistic Regression*: Decidiu-se omitir das tabelas anteriores os parâmetros do modelo pois este classifica todas as observações do conjunto de teste numa classe, que é efetivamente um sinal de que o algoritmo não está a funcionar corretamente.
- *Decision Tree*: O valor de AUC é o mais baixo em qualquer um dos modelos, que explica o algoritmo com maior aleatoriedade na classificação da variável alvo. É também o algoritmo com valores de *accuracy* e *recall* mais reduzidos.
- *K-Nearest Neighbors*: Valores intermédios em todas as medidas de precisão, modelo mantém as suas medidas em ambos os modelos.
- *Random Forest*: Medidas de precisão melhores que o algoritmo imediatamente anterior, no entanto não são superlativas em relação a outros modelos, à exceção do F1-Score.
- *Naive Bayes*: Algoritmo que contém uma menor aleatoriedade na classificação de *target*.
- *Gradient Boosting*: Algoritmo que partilha os melhores indicativos de AUC e contém também os melhores valores para os critérios *Accuracy*, *Precision* e *Recall*.

Perante a análise obtida, e em relação às medidas de precisão, considera-se que o algoritmo *Gradient Boosting* é o mais completo, por ter os melhores indicativos em 4 das 5 medidas utilizadas, sendo apenas superado em *F1-Score* apesar de conter os melhores resultados para *precision* e *recall*, que integram a fórmula de *F1-Score*.

Fazendo a comparação inter-modelos e com o algoritmo *Gradient Boosting* como referência, entende-se que o valor de aleatoriedade (explicado por AUC) se mantém em ambos os modelos tal como o valor de *F1-Score* e que os valores de *Accuracy*, *Precision* e *Recall* baixa de Modelo1 para Modelo2. Apresentando o algoritmo *Gradient Boosting* no modelo1 a seguinte curva ROC:

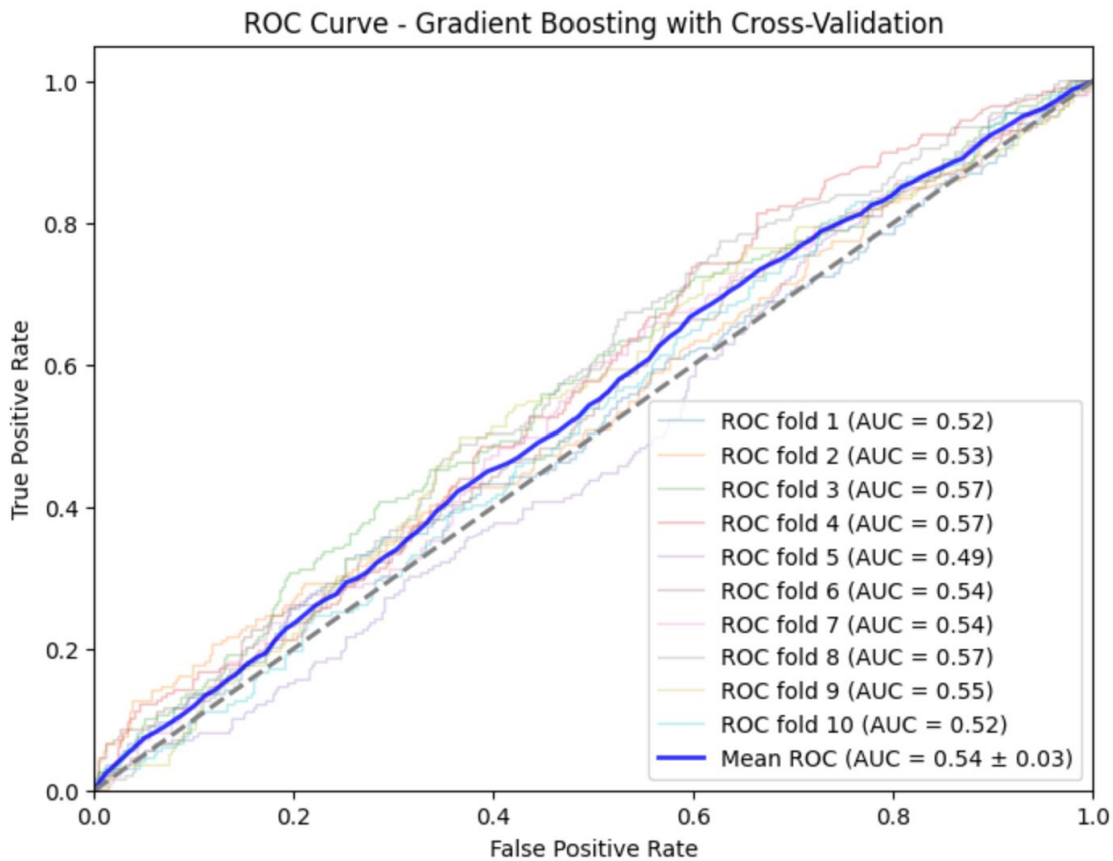


Figura 16-Curva ROC- Gradient Boosting (Modelo1)

A figura acima e através do valor médio de AUC, demonstra que o modelo está próximo de uma classificação aleatória, pois o valor está próximo de 0.5. Através das diferentes curvas, entende-se a variabilidade no desempenho de cada *fold*, variando esses valores de 0.49 a 0.57, demonstrando também alguma inconsistência entre *fold's*.

Na observação mais particularizada de cada um deles, verifica-se também um valor de AUC abaixo de 0.5 no *fold* 5 que se interpreta como um desempenho pior que aleatoriedade total.

Deployment

Os modelos desenvolvidos ao longo deste projeto têm o potencial de serem integrados em aplicações práticas com valor significativo para diversas partes interessadas. A seguir, são apresentadas duas visões originais de possíveis aproveitamentos empresariais que podem ser implementados com ajustes e acompanhamento adequados.

Uma das aplicações possíveis dos modelos é o desenvolvimento de uma aplicação voltada para empresas e investidores interessados em patrocinar jogadores de tênis ou eventos relacionados. A ferramenta utiliza modelos e análise de dados para fornecer insights detalhados, ajudando os patrocinadores a tomar decisões mais informadas e estratégicas. Os patrocinadores teriam acesso a estatísticas detalhadas sobre o desempenho passado dos jogadores, incluindo vitórias, derrotas, performance em diferentes tipos de solo e em torneios específicos, destacando jogadores que estão em ascensão ou declínio.

Será explorado como esses fatores podem moldar as estratégias de patrocínio:

- A análise do desempenho passado dos jogadores permite avaliar a consistência de um atleta ao longo do tempo;
- A performance em diferentes tipos de solo ajuda a identificar onde os jogadores são mais fracos e mais fortes;
- Destacar jogadores em ascensão ou declínio orienta os patrocinadores sobre onde investir com mais segurança;
- Os valores dos prêmios monetários conquistados são um indicador do sucesso e da capacidade do jogador em competir em torneios de alto nível;
- A discrepância de classificação entre adversários consoante as diferentes fases do torneio revela a competitividade dos jogos em que os jogadores estão envolvidos, proporcionando insights sobre a capacidade de competir contra adversários mais bem classificados.

Estes fatores irão contribuir para uma análise mais abrangente, facilitando a identificação de oportunidades de patrocínio mais vantajosas.

Outra aplicação seria a criação de uma plataforma interativa projetada para aumentar o conhecimento dos fãs de tênis. A aplicação ofereceria uma variedade de recursos educativos e

interativos que ajudam os utilizadores a entender melhor o desporto, as táticas usadas pelos jogadores e a história das partidas mais emblemáticas. Seria possível uma análise detalhada de partidas históricas e atuais, destacando os momentos decisivos e as estratégias utilizadas pelos jogadores ou como diferentes fatores influenciaram o resultado da partida.

Conclusão

Tendo em conta o que foi analisado e estudado neste relatório e com o objetivo do trabalho em mente, ou seja, a criação de modelos de classificação que tivessem utilidade na previsão do número de sets necessários (2 ou 3) para a conclusão de um jogo nos Países Baixos, foram tiradas as conclusões seguintes.

Após a análise detalhada dos dados e a implementação dos modelos, conclui-se que o objetivo de criar um modelo para prever o número de sets necessários para a conclusão de um jogo de ténis profissional nos Países Baixos foi atingido com sucesso.

Durante esta análise, foi interessante perceber como variáveis aparentemente insignificantes podem impactar o desempenho dos jogadores, como o tipo de superfície. Portanto, podemos concluir que os resultados numa partida de ténis são influenciados tanto por características internas quanto externas ao próprio jogador. Noutras palavras, dependem de atributos físicos, táticas, experiência, adaptação ao próprio país e ao tipo de competição.

A capacidade de prever a duração dos jogos, em termos de sets, oferece benefícios significativos para diversas partes interessadas no mundo do ténis. Organizadores de torneios podem otimizar a gestão dos campos e melhorar a programação dos jogos, enquanto patrocinadores podem maximizar a exposição e retorno sobre investimento. Treinadores, analistas e atletas também se beneficiam ao ajustar estratégias baseadas em padrões históricos e fatores influentes identificados.

Em suma, o modelo desenvolvido não só alcança o objetivo proposto, mas também fornece uma ferramenta valiosa para melhorar a organização e performance no contexto dos torneios de ténis nos Países Baixos.