Licenciatura em Ciência de Dados - 2º ano

Relatório Final

Previsão do número de sets dos torneios dos EUA

Projeto Aplicado em Ciência de Dados I

1 de junho de 2024

Discentes: João Dias nº 110305 / Samuel Ricardo nº 110884 /
Rafael Cerqueira nº 110860 / Felipe Pereira nº 110861 / David Franco nº 110733

2023/2024

iscte INSTITUTO UNIVERSITÁRIO DE LISBOA

ÍNDICE

INTRODUÇÃO **2**
BUSINESS UNDERSTANDING **3**
i. Ténis Profissional & ATP Tour 4
ii. Objetivo de Negócio 5
iii. Trabalhos Semelhantes 5
DATA UNDERSTANDING **6**
i. Initial Data 7
ii. Data Quality 8
iii. Data Exploration 9
DATA PREPARATION **14**
MONGODB 15
MYSQL 16
PYTHON 19
Importação e pré-processamento inicial dos datasets 19
Data enrichment 19
Junção dos datasets 20
Feature engineering 21
i. Todos os torneios 21
ii. Apenas torneios dos Estados Unidos da América 21
Outliers 23
Valores omissos 28
Data selection 29
Análise de correlações 30
MODELING **34**
Modelos desenvolvidos 36
i. Regressão Logística (Minimum Viable Model) 36
ii. XGBoost 37
iii. MLP Classifier 38
iv. Random Forest 40
EVALUATION **41**
i. Treino e test split (c/ oversampling) 41
ii. Validação cruzada (cross-validation) 42
iii. Performance do Minimum Viable Model 44
iv. Performance do melhor modelo selecionado 45
DEPLOYMENT **47**
CONCLUSÃO **48**
BIBLIOGRAFIA **49**

INTRODUÇÃO

Este trabalho, desenvolvido no âmbito da unidade curricular de Projeto Aplicado em Ciência de Dados I, tem como objetivo aplicar técnicas de análise de dados e modelação preditiva para prever o número de sets em partidas de ténis nos Estados Unidos da América. A previsão de resultados desportivos é um desafio na ciência de dados devido à complexidade e variabilidade inerentes aos desportos competitivos.

No contexto deste projeto, utilizámos um conjunto de dados detalhado sobre partidas de ténis, obtido de fontes como a ATP ('Association of Tennis Professionals'). O nosso objetivo foi desenvolver e avaliar vários modelos preditivos, analisando o desempenho de cada um em termos de accuracy, precision, recall, f1 score e ROC-AUC. Através desta análise, procuramos identificar as melhores abordagens para prever com maior precisão os número de sets das partidas.

Este projeto não só visa aprofundar o conhecimento técnico relacionado com a ciência de dados, mas também aplicar metodologias práticas, como o CRISP-DM (Cross Industry Standard Process for Data Mining), para estruturar e conduzir processos de análises.

Os desafios encontrados, as limitações dos modelos desenvolvidos e as sugestões para melhorias futuras são detalhadamente discutidos, proporcionando uma visão abrangente das dificuldades e aprendizagens resultantes deste projeto. Assim, este trabalho reflete uma aplicação prática dos conhecimentos adquiridos ao longo do curso, demonstrando a capacidade de enfrentar problemas complexos através da ciência de dados.

FRAMEWORK METODOLÓGICO

No presente relatório, adotaremos o framework CRISP-DM (Cross-Industry Standard Process for Data Mining) como metodologia principal para prever o número de sets de ténis em torneios dos Estados Unidos da América, já que é uma das principais metodologias em Ciência de Dados, que oferece uma abordagem sistemática e detalhada na execução do projeto. O CRISP-DM é composto por seis fases, conforme pode ser visto na Fig. 1: business understanding, data understanding, data preparation, modeling, evaluation e deployment. Dada a natureza académica do relatório, importa reforçar que a fase de deployment não contará com o mesmo destaque que as restantes, já que os modelos desenvolvidos não serão colocados em ambiente de produção.

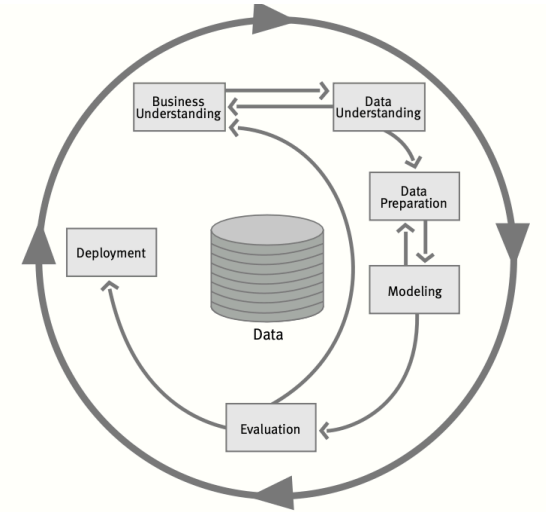


Figura 1 - Framework geral do CRISP-DM

BUSINESS UNDERSTANDING

A fase de Business Understanding envolve várias etapas fundamentais, tal como se pode ver na Fig. 2, ao nível da definição dos objetivos de negócio, onde vamos procurar compreender bem os conceitos associados ao ATP e à modalidade de ténis, a definição mais técnica em termos de objetivos e de data mining (por outras palavras, é dizer como se vai materializar a implementação deste projeto), a análise de projetos similares, bem como a elaboração e planificação das restantes fases.

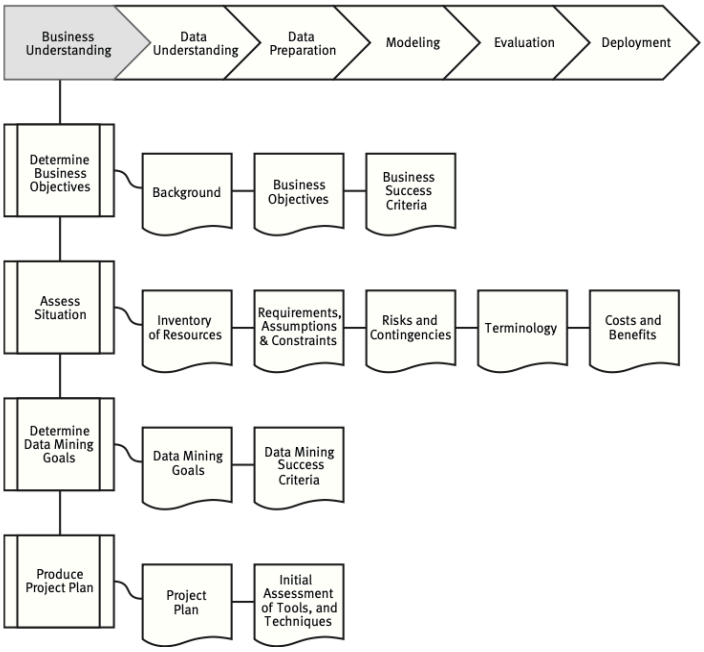


Figura 2 - Descrição das tarefas associadas à fase de Business Understanding

i. Ténis Profissional & ATP Tour

Ténis é sem dúvida um dos desportos mais populares de todos os tempos. Este desporto é jogado por milhões de pessoas em diversas regiões do nosso Planeta e os torneios profissionais atraem e revelam uma grande atenção e prestígio internacional. (De Seranno, 2020)

Ténis é um desporto jogado com raquetes onde dois adversários defrontam-se entre si ou duas equipas compostas por duplas. Durante o jogo, um dos jogadores é o 'server' enquanto o outro desempenha a função de 'receiver'. Os jogadores posicionam-se em lados opostos no campo de ténis tendo uma rede estendida a meio do campo que separa o lado de cada jogador. Os campos podem ser de pisos diferentes, 'grass', 'clay', 'hard' ou 'carpet'. Cada jogador tem duas tentativas de serviço, e após um serviço bem sucedido, os tenistas vão trocando a bola até algum deles ganhar o 'rally', obtendo desta forma um ponto. (Sipko, 2015)

Um 'game' consiste numa sequência de pontos durante a vez de serviço de um dos jogadores em que o primeiro a ganhar pelo menos quatro pontos e com uma diferença de dois do adversário, ganha o 'game'. Os pontos são contados numa sequência de 0, 15, 30, 40. Se o resultado chegar a 40-40, o jogador que ganhar o seguinte ponto encontra-se em vantagem e está apenas a um ponto de ganhar o 'game'; a isto chama-se 'deuce'. A seguir a cada 'game', os jogadores alternam quem vai servir. O primeiro jogador a ganhar pelo menos seis 'games' com uma diferença de pelo menos dois do oponente ganha o 'set'. No entanto, se o resultado do 'set' chegar a 6-6, na maioria dos torneios, um desempate é jogado, um 'game' extra em que o primeiro jogador a atingir sete pontos com uma diferença de dois pontos do adversário, ganha o 'set'. A partida acaba quando um dos jogadores atinge um determinado número de 'sets'. (ITF, 2024)

O 'tour' profissional de ténis masculino é organizado pela 'Association of Tennis Professionals' (ATP). Existem quatro eventos com diferentes níveis de competitividade: 'ATP 250 series', 'ATP 500 series', 'ATP World Tour Masters 1000 series' e 'ATP World Tour Finals'. Também fazem parte do calendário da ATP torneios de 'Grand Slam', onde consta o 'US Open'. Os torneios são jogados à melhor de três sets exceto os grand slams que têm um formato à melhor de cinco sets. Cada um destes eventos está associado a prémios monetários bastante distintos entre eles; podem variar aproximadamente entre $500,000 (para o 'ATP 250 series') até $18,000,000 (para um torneio de 'Grand Slam'). A qualificação para os torneios ATP é com base no ranking mundial, portanto jogadores com melhor rank qualificam-se para os torneios mais competitivos, mais prestigiados e com maior prémio. O rank de um jogador é calculado com base nos pontos de ranking mundial ATP obtidos num determinado espaço de tempo consoante a estrutura da organização. Por exemplo, chegar às semifinais num evento 'ATP 250 series' equivale a 90 pontos, enquanto que no 'ATP Masters 1000 series' vale 360 pontos e num evento 'Grand Slam' 720 pontos. Esta mecânica de pontuação foi desenhada de forma a estabelecer um sistema justo e transparente, permitindo que os melhores jogadores participem nos eventos mais prestigiosos e lucrativos. (Mathers, 2016)

ii. Objetivo de Negócio

O objetivo deste projeto é desenvolver um modelo preditivo para prever o número de sets em partidas de torneios realizados nos Estados Unidos. A intenção é criar uma aplicação que ajude sobretudo os treinadores, mas também organizadores dos torneios e das transmissões televisivas a planear da melhor forma possível as suas atividades.

Ter a possibilidade de prever o número de sets de uma partida abre portas para um melhor planeamento e organização do pessoal necessário, otimização do tempo na programação dos jogos para não haver atrasos e sobreposição dos mesmos por causa dos campos disponíveis, entre outros fatores que podem ser importantes para o bom funcionamento dos torneios. Em relação às transmissões televisivas e à publicidade, a previsão da duração dos jogos com base no número de sets permite uma maior precisão na agenda das suas transmissões e uma melhor gestão da exibição dos anúncios publicitários.

No que diz respeito à análise do desempenho desportivo, a aplicação pode ser uma mais valia ao dar informações sobre o desempenho dos atletas, identificando padrões que influenciam o número de sets jogados numa partida. Desta forma, os treinadores podem ajustar as suas táticas e estratégias de treino e de jogo, mas também ter uma comparação detalhada do seu jogador com o respetivo adversário, através de variáveis como a diferença de ranking, altura, número de jogos disputados no ano, histórico de confrontos diretos, entre outras.

A aplicação desenvolvida terá funcionalidades como a previsão do número de sets e detalhes sobre o desempenho histórico dos desportistas, facilitando o planeamento de treinos e estratégias de jogo. A implementação desta ferramenta trará benefícios significativos para todo o pessoal envolvido nos torneios ATP dos Estados Unidos, melhorando a gestão, o planeamento e a análise do desporto.

iii. Trabalhos Semelhantes

No âmbito da previsão desportiva, apenas um número limitado de estudos têm sido realizados, utilizando uma variedade de modelos e conjuntos de dados para prever os resultados, desempenho e duração das partidas de ténis.

De Seranno (2020) propôs uma abordagem de machine learning para prever o vencedor das partidas de ténis do circuito ATP. Ao utilizar um conjunto de dados open-source, o modelo de regressão logística destacou-se significativamente em relação a uma referência baseada no ranking oficial da ATP. Isso indica que, ao utilizar o modelo de regressão logística, foi possível fazer previsões mais precisas sobre os resultados das partidas de ténis do circuito ATP do que simplesmente seguir o ranking oficial da ATP como indicador.

Mathers (2016) descreveu um estudo de caso sobre a importância da psicologia no ténis profissional, enfatizando a necessidade de um programa com vista à melhoria da força

mental dos atletas em conjunto com outras habilidades físicas e técnicas para o sucesso no circuito ATP.

Sipko (2015) utilizou a regressão logística e uma rede neural para prever partidas da ATP de 2013-2014 com um período de treino de nove anos, gerando um retorno sobre o investimento de 4% no mercado de apostas desportivas.

Estes resultados sugerem que, apesar dos avanços na área do machine learning e dos dados disponíveis, ainda é desafiador alcançar previsões altamente precisas e consistentes em partidas de ténis, com base em dados reais.

DATA UNDERSTANDING

A fase de Data Understanding, tal como se pode ver na Fig. 3, passa por várias fases. No contexto deste projeto dividimos esta secção em três, nomeadamente ao nível da análise inicial e descrição dos dados presentes no dataset utilizado; da verificação da qualidade destes dados e da análise exploratória (note-se que para este último ponto foi feito algum pré-processamento ao nível do tratamento e criação de novas variáveis para uma melhor compreensão dos dados).

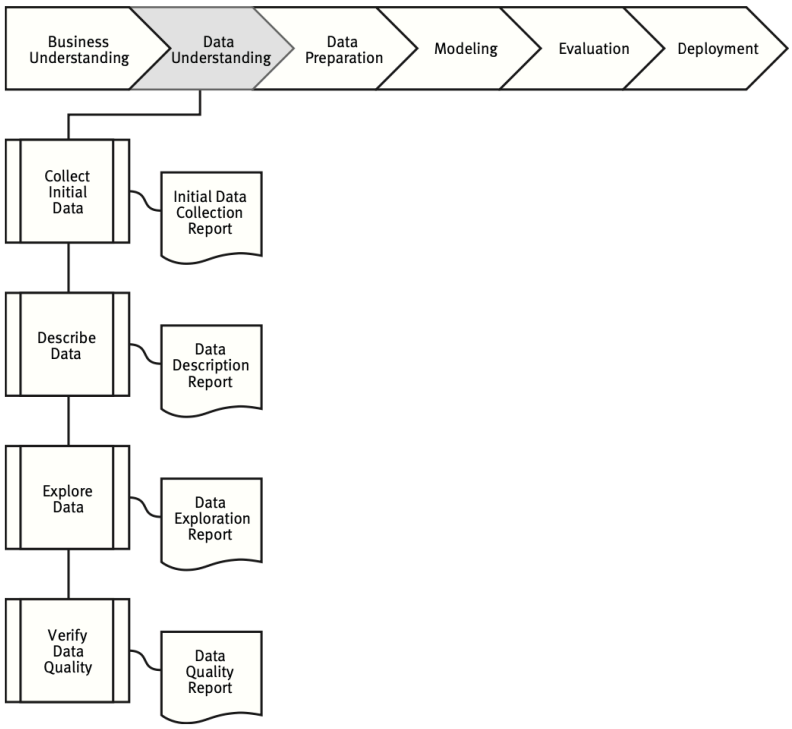


Figura 3 - Descrição das tarefas associadas à fase de Data Understanding

i. Initial Data

A primeira etapa passou por adquirir o conjunto de dados designado como "atpplayers" no formato JSON, obtido através do site da ATP. As bases de dados deste website fornecem uma gama abrangente de informações sobre partidas de ténis profissional masculino, torneios e detalhes dos jogadores, incluindo resultados de partidas, estatísticas individuais e histórico de torneios.

O dataset utilizado compreende um total de 1308835 observações distribuídas em 16 variáveis. Cada observação representa uma partida de ténis realizada, envolvendo os 500 melhores jogadores do ranking ATP.

Aqui está uma descrição geral das 16 variáveis iniciais:

- id: Um identificador único atribuído a cada entrada de dados no conjunto de dados;
- PlayerName: O nome de um dos jogadores de ténis;
- Born: O local de nascimento do jogador de ténis, podendo conter informações sobre a cidade e o país de nascimento;
- Height: A altura do jogador de ténis em centímetros;
- Hand: A preferência de mão do jogador principal (PlayerName), identificando se ele é destro (Right-Handed), canhoto (Left-Handed) ou ambidestro (Ambidextrous), ou seja, utiliza ambas as mãos com habilidade semelhante. Além disso, ela indica como são efetuados os batimentos de “esquerda”: se é a uma mão (One-Handed Backhand), a duas mãos (Two-Handed Backhand) ou é desconhecido (Unknown Backhand);
- LinkPlayer: Um link para o perfil do jogador principal no site oficial do ATP Tour;
- Tournament: O nome do torneio de ténis em que a partida ocorreu.
- Location: A localização geográfica do torneio, indicando o país e/ou cidade em que ocorreu;
- Date: Data de início e fim do torneio;
- Ground: Tipo de superfície do campo onde o torneio ocorreu. Pode ser Hard, que consiste em materiais uniformes e endurecidos, geralmente com uma camada superficial de acrílico; Grass, que é relva natural; Clay, que é em terra batida e é composto por pequenos pedaços de pedra pouco consolidada ou que se parte facilmente, podendo ter coloração vermelha ou verde; ou Carpet, que é composto por materiais sintéticos, como relva artificial com preenchimento de areia;

- Prize: O valor do prémio em dinheiro associado ao jogo (em diferentes unidades monetárias);
- GameRound: A fase específica do torneio em que a partida ocorreu (por exemplo, Round Robin, Quartos de Final, Semifinais);
- GameRank: A classificação do jogador oponente no momento da partida, refletindo a sua posição no ranking mundial da ATP;
- Oponent: O nome do adversário enfrentado pelo jogador na partida.
- WL: O resultado da partida, indicando se o jogador principal venceu (W) ou perdeu (L) o confronto;
- Score: Representa os resultados do jogo por sets. Cada par de números corresponde a um set e aos respectivos parciais.

Embora este seja o conjunto de dados inicialmente atribuído, ao longo do trabalho iremos utilizar outras bases de dados, que serão discutidas nas respetivas ocasiões.

ii. Data Quality

Embora o dataset utilizado venha do repositório oficial do ATP, existem inúmeros problemas, como a presença de jogadores diferentes com o mesmo nome (Alberto Gonzalez, Alexey Nesterov, Andreas Weber, Martin Damn, Enrique Pena, Mark Kovacs e Robert Phillips) e de torneios repetidos, nomeadamente o torneio “Valencia” que ocorreu em 2006 e em 2007.

No que diz respeito a valores omissos, estes estão presentes, sobretudo nas variáveis “Born”, “Height”, “Hand” e “Prize” com 6276, 6317, 3987 e 2639 valores omissos, respetivamente. Embora estes valores sejam pequenos comparados com a dimensão total do dataset (1308835 observações), devem ser considerados.

Verificam-se também problemas específicos na variável “Height”, particularmente em relação a valores estranhos, como jogadores com 0 cm, 3 cm, 15 cm, 71 cm ou 510 cm de altura, e na variável “Date”, onde foi possível identificar 665 torneios sem a data de término.

Outro problema identificado foi na variável relativa ao resultado dos jogos “WL”, onde notámos a presença de strings relativas a resultados "bye". Esta situação, em que um jogador não precisa de competir numa ronda inicial de um torneio, faz com que ele avance automaticamente para a próxima ronda sem jogar (Crim, n.d.), sendo erroneamente contabilizada como vitória, mesmo sem a realização do jogo propriamente dito.

Verificámos também que existiam alguns torneios, especificamente seis, em que não havia dados relativos ao piso (“Ground”) dos jogos.

iii. Data Exploration

Para melhorar a análise exploratória inicial dos dados e prever o número de sets de um jogo de ténis, começámos por examinar os atributos comparativos entre os jogadores. Entre esses atributos, consideramos a diferença no número de jogos disputados no ano anterior e a diferença nos confrontos diretos, variáveis criadas para uma melhor compreensão do problema. Explorámos a frequência de confrontos diretos entre jogadores, uma métrica essencial para avaliar a competitividade e as rivalidades no desporto (Henry, n.d.). A Figura 4 apresenta a relação entre a diferença no número de jogos disputados anualmente e o registo de confrontos diretos entre os jogadores (considerando todos os torneios em ambos os casos).

Na Figura 4 observa-se uma alta concentração de confrontos diretos quando a diferença no número de jogos é pequena (entre 0 e 50). Isso indica que jogadores com um número similar de jogos no ano tendem a enfrentar-se com mais frequência. À medida que a diferença no número de jogos aumenta, a distribuição de confrontos diretos torna-se mais heterogénea e menos densa. É menos comum encontrar jogadores com grandes diferenças no número de jogos a enfrentar-se frequentemente. Existem alguns pontos, especialmente com uma diferença de jogos superior a 100, que ainda registam um número significativo de confrontos diretos. Esses data points podem indicar situações especiais, como jogadores de alto ranking enfrentando adversários de menor ranking em fases iniciais de torneios (nomeadamente após os qualifiers). A tendência geral sugere que jogadores com um número similar de jogos tendem a competir mais entre si, possivelmente devido à maior chance de se encontrarem em torneios com estrutura similar de competição.

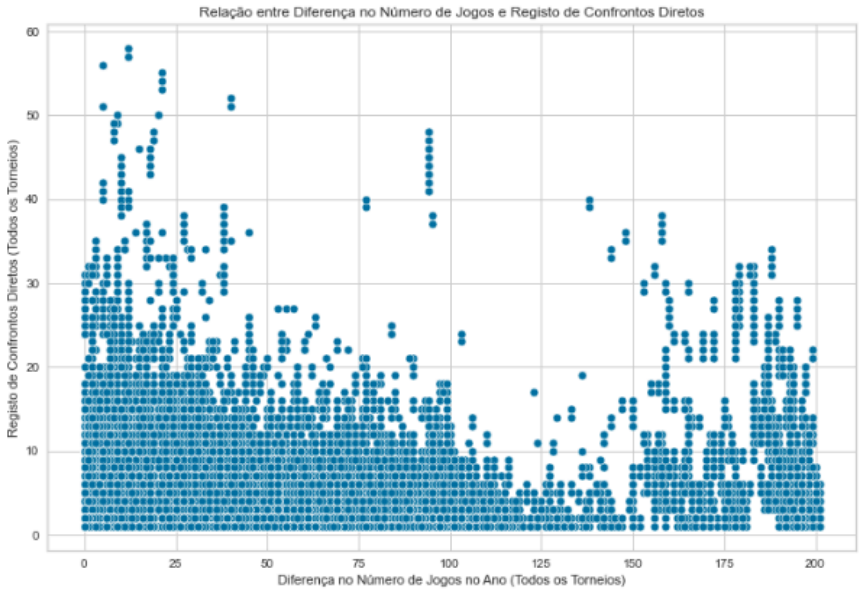


Figura 4 - Relação entre a Diferença do Número de Jogos no Ano Anterior e os Confrontos Diretos

Também examinámos a distribuição das diferenças de altura (Figura 5) e de idade (Figura 6) entre jogadores que se enfrentaram, pois estas variáveis podem influenciar

significativamente a dinâmica dos jogos (Barbosa, 2016). Observa-se uma alta frequência de confrontos diretos entre jogadores com pequenas diferenças de altura, especialmente na faixa de 0 a 5 centímetros, indicando que jogadores de estaturas similares tendem a enfrentar-se mais. À medida que a diferença de altura aumenta, a frequência desses confrontos diminui gradualmente, sendo raros os casos com diferenças superiores a 25 centímetros. Semelhante à distribuição de altura, a maioria dos confrontos diretos ocorre entre jogadores com pequenas diferenças de idade, especialmente na faixa de 0 a 2 anos. À medida que a diferença de idade aumenta, a frequência de confrontos diminui, sendo menos comuns diferenças superiores a 6 anos. Curiosamente, há um aumento na frequência para diferenças de idade em torno de 10 anos, refletindo confrontos entre jogadores de gerações diferentes. Deste modo, jogadores com características físicas e etárias semelhantes tendem a enfrentar-se mais frequentemente.

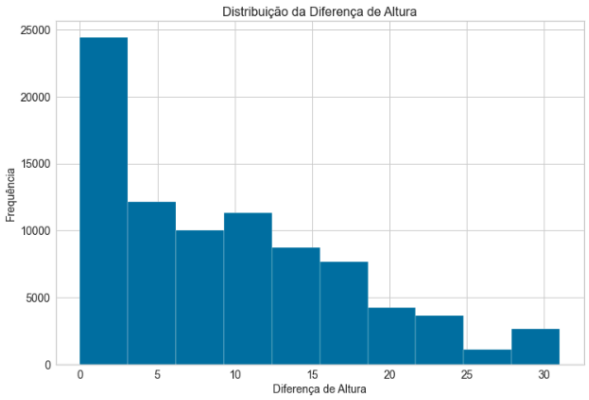


Figura 5 - Distribuição de Diferença de Altura

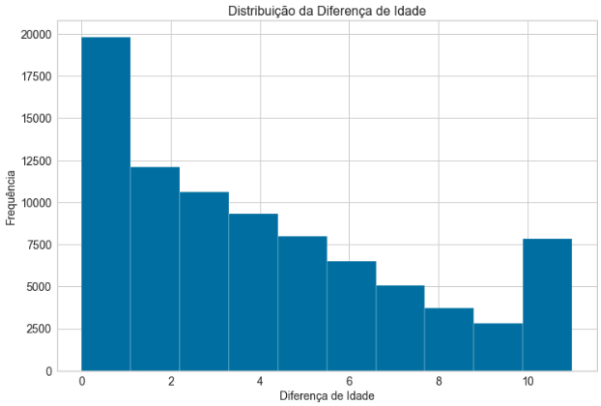


Figura 6 - Distribuição de Diferença de Idade

Para aprofundar a nossa análise exploratória dos dados de ténis, é fundamental examinar não apenas as características físicas dos jogadores, mas também a compatibilidade dos seus estilos de jogo. Especificamente, a compatibilidade de forehand e backhand entre os jogadores pode desempenhar um papel significativo na dinâmica dos confrontos diretos (Genevois, Reid, Rogowski, & Crespo, 2015). O forehand e o backhand são golpes cruciais no ténis, e as suas compatibilidades podem influenciar diretamente a estratégia, o ritmo e o resultado de uma partida. Jogadores com estilos de forehand ou backhand compatíveis podem ter confrontos mais equilibrados, enquanto incompatibilidades nesses golpes podem criar vantagens ou desvantagens significativas. Por exemplo, um jogador com um forehand muito forte pode beneficiar-se se o adversário tiver um forehand mais fraco ou menos compatível. Os gráficos a seguir (Figuras 7 e 8) apresentam a compatibilidade de forehand e backhand entre jogadores e oponentes. Por compatibilidade entende-se utilizar a mesma mão.

Os gráficos mostram que existe uma alta frequência de compatibilidade tanto de forehand quanto de backhand entre os jogadores. Observa-se que a maioria dos confrontos diretos ocorre entre jogadores cujos estilos de forehand e backhand são compatíveis. Esses resultados sugerem que jogadores com estilos de jogo compatíveis tendem a encontrar-se

com mais frequência em torneios. A alta compatibilidade nos estilos de jogo pode influenciar a dinâmica dos confrontos e as estratégias utilizadas pelos jogadores.

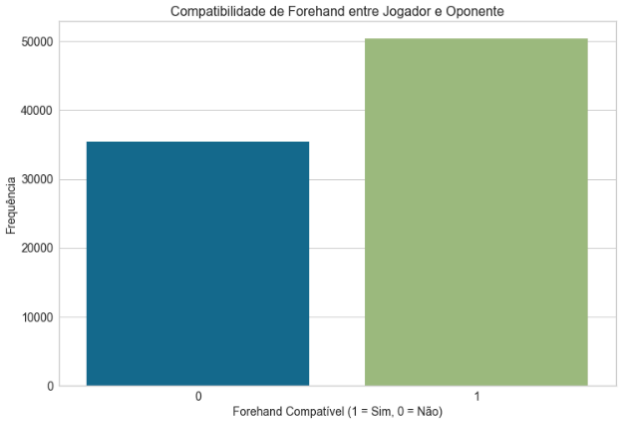


Figura 7 - Compatibilidade de Forehand

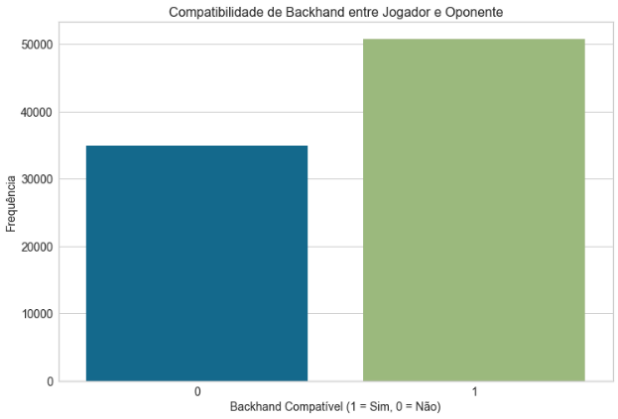


Figura 8 - Compatibilidade de Backhand

Um fator importante a ser considerado na realização de um torneio de ténis é a vantagem de jogar em "casa". No caso deste trabalho, o facto do jogador ser natural dos EUA. Jogar em casa significa que o jogador está familiarizado com as condições do campo, a altitude, o clima e outros fatores específicos do local. Além disso, ter o apoio do público local é uma vantagem significativa (Koning, 2011). Analisámos a existência de jogos em que existam nenhum, um ou dois jogadores do país anfitrião do torneio através do gráfico da Figura 9. O gráfico demonstra que menos da metade dos jogos do dataset não possuem pelo menos um jogador da "casa", cerca de pouco mais de 35.000 jogos, embora demonstre que os torneios analisados atraem competição internacional. Também é importante notar que as situações em que ambos os jogadores são da nação anfitriã são menos frequentes.

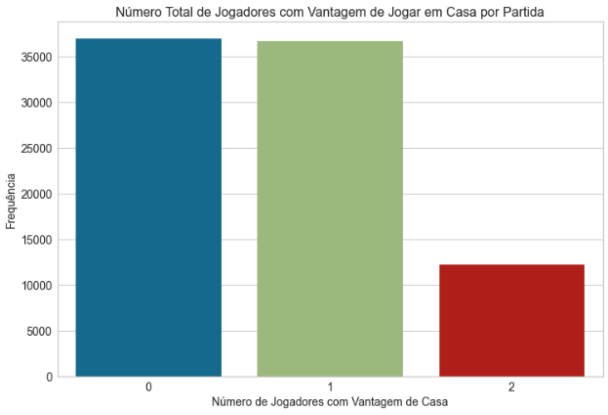


Figura 9 - Distribuição de Jogadores com Vantagem de Jogar em Casa por Partida

Outro fator importante a considerar na previsão do número de sets de uma partida é a fase do torneio em que o jogo se realiza, pois jogos que ocorrem em fases finais tendem a ser mais competitivos devido à eliminação, que já decorreu, de jogadores mais fracos nas fases iniciais (Gu & Saaty, 2019). No caso do dataset utilizado, possuímos mais jogos em rondas iniciais do que em fases mais avançadas, principalmente em jogos que disputam o pódio da competição (Figura 10), o que se considera fazer sentido já que existe um maior número de jogos em rondas iniciais, do que em fases mais avançadas dos torneios.

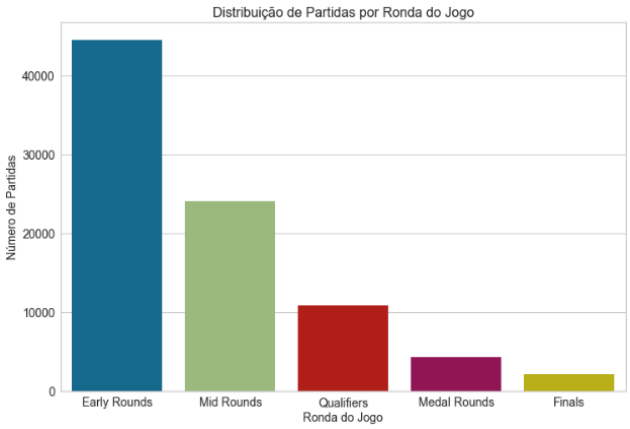


Figura 10 - Distribuição de Partidas por Rondas dos Torneios

Quando pensamos em partidas renhidas de ténis, normalmente associamos a confrontos de grandes jogadores que figuram no topo dos rankings mundiais. O mesmo vale para jogadores que possuem níveis parecidos, ou no contexto do desporto, ranks próximos. Ao analisar esta hipótese no dataset utilizado (Figura 11), verificamos que são mais frequentes jogos em que a diferença de rank entre os jogadores é mais baixa, o que pode sugerir uma presença significativa de jogos mais renhidos (Reid, McMurtrie, & Crespo, 2010).

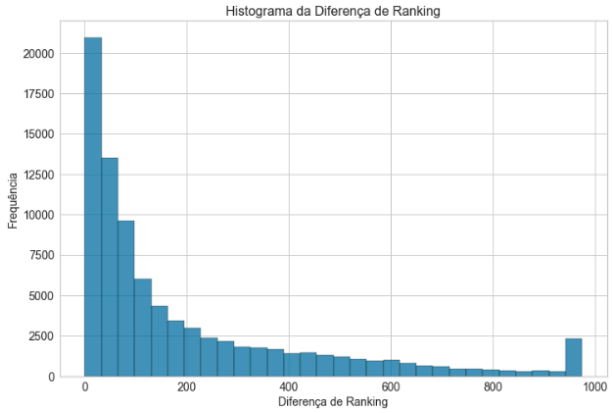


Figura 11 - Histograma de Diferença de Ranking

Similar ao analisado no início, mas agora focando-nos na componente univariada, fizemos a análise da diferença entre os jogos disputados durante o ano anterior dos dois jogadores de uma partida (Figura 12), dado que jogadores que disputam mais partidas tendem a ficar mais preparados para os jogos seguintes e com maior ritmo de competição, ao mesmo tempo que o cansaço também se pode acabar por refletir (Reid, Crespo, Lay, & Berry, 2007). Verificamos que a frequência de jogos diminui à medida que a diferença entre o número de jogos disputados no ano anterior dos dois jogadores aumenta, embora haja uma elevada frequência de registos em que essa diferença é por volta de 62 jogos. Isso sugere que jogadores com históricos de jogos mais semelhantes têm maior probabilidade de se enfrentarem regularmente em torneios, enquanto aqueles com diferenças significativas no número de jogos disputados podem ter menos confrontos diretos.

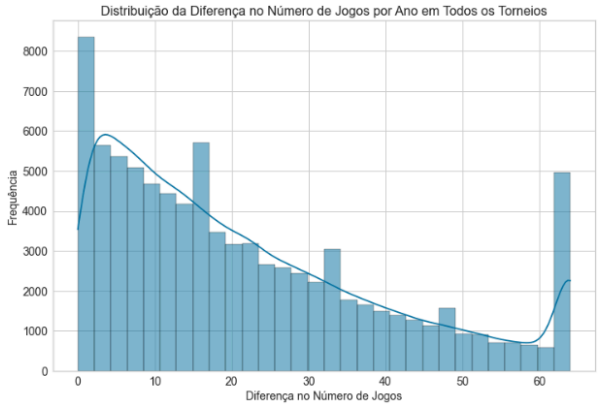


Figura 12 - Distribuição da Diferença de nº de Jogos Realizados no Ano Anterior por Jogador

Um dos aspetos mais importantes de uma partida de ténis é o tipo de piso em que ela é praticada, pois a diferença entres os pisos podem alterar a dinâmica e a velocidade do jogo, como abordado em fases anteriores deste relatório. No contexto da presente base de dados, o piso “Hard” é dominante, estando presente em mais de 60000 jogos, seguido pelo piso “Clay” com pouco mais de 15000 jogos disputados, enquanto os pisos “Grass” e “Carpet” possuem menos de 5000 jogos registados (Figura 13).

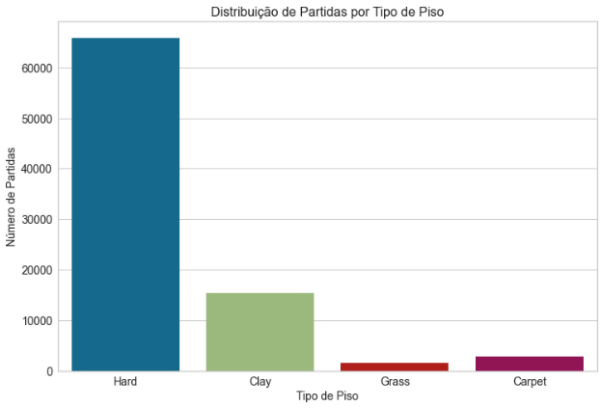


Figura 13 - Distribuição de Partidas por Tipo de Piso

DATA PREPARATION

A fase de Data Preparation foi feita em três grandes etapas, para tirar o máximo proveito de cada plataforma, nomeadamente:

1. **MongoDB**: para armazenamento e gestão dos grandes volumes de dados associados à base de dados em questão, que apresentava 1308835 linhas e 16 colunas, pela sua maior flexibilidade nas etapas de pré-processamento iniciais.
2. **MySQL**: para a organização e manipulação dos dados após o pré-processamento inicial feito em MongoDB, nomeadamente ao nível das variáveis que diziam respeito a países, de forma a garantir a integridade e a consistência destes dados.
3. **Python**: para a limpeza, transformação e análise mais completa dos dados, com recurso a bibliotecas como pandas e numpy, bem como para a aplicação de algoritmos de machine learning na fase de Modeling.

Este processo mostrou ser essencial para assegurar que os dados do ATP estivessem num formato adequado para treinar e validar os modelos desenvolvidos no capítulo seguinte. O esquema seguinte (Figura 14) ilustra algumas das principais tarefas associadas à etapa de Data Preparation do CRISP-DM, que foram sendo guidelines para a execução desta fase.

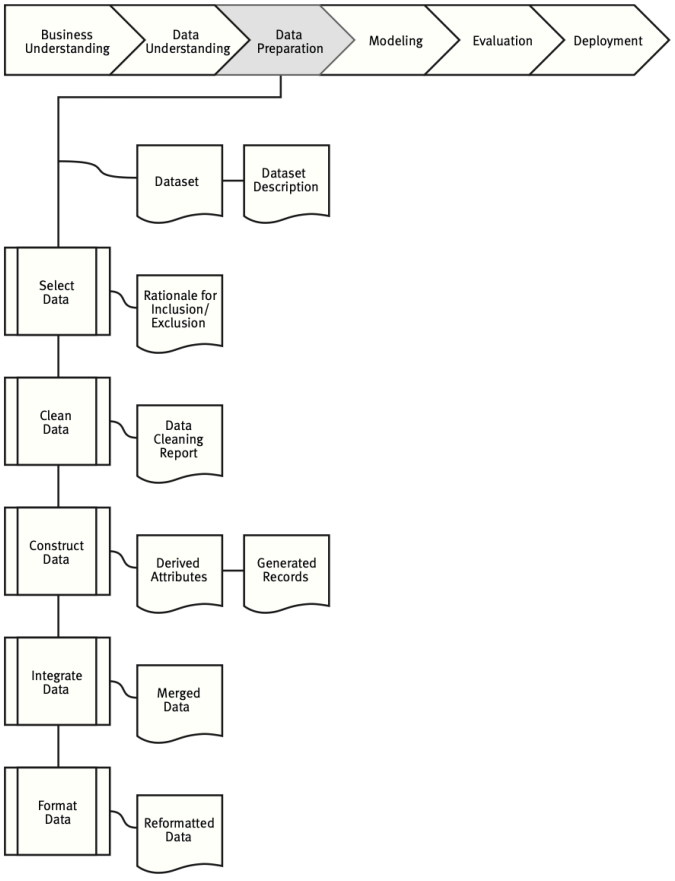


Figura 14 - Descrição das tarefas associadas à fase de Data Preparation

MONGODB

A primeira fase da limpeza de dados foi feita com recurso ao MongoDB. Primeiramente foi efetuada a importação da base de dados para o programa, de seguida foram criadas três coleções para facilitar a limpeza dos dados, ou seja, foram criadas as coleções “players”, “tournaments” e “games”.

Desta forma, iremos limpar os dados relativos à coleção “players”, que continham as variáveis relativas ao nome, à cidade/país onde nasceu, a altura, a mão com que joga e o link para a página ATP do jogador, assim, foram importados 9960 jogadores distintos. O próximo atributo tratado foi a variável da mão com que o jogador joga, este atributo vem com a *forehand* e *backhand* separadas por uma vírgula, por isso criámos dois campos a partir deste com a respetiva informação já corrigida. Existia, ainda, jogadores com valores omissos neste atributo ,razão pela qual, iremos associar estes valores omissos a “Unkown Hand/Backhand”, já que ambas as opções refletem o facto de se tratar de um “Hand/Backhand” desconhecido. Outro problema desta coleção é relativa ao atributo relacionado com a altura dos jogadores, para além de existirem valores omissos existe, também, alturas que correspondem a valores como 0, 3, 15, 71, 510, valores obviamente incorretos, e que assim aparecem nas respectivas páginas de jogador do ATP como as do Abdulrahman Alawadhi (ref.), Jorge Panta (ref.) ou Carlos Di Laura (ref.). Existem mais situações de jogadores com erros de inserção nas alturas, no entanto, estes são apenas alguns exemplos. Assim, o tratamento relativo a esta variável será feito numa fase posterior em Python.

De seguida foi limpa a coleção “tournaments” em que os atributos presentes na mesma são “Tournament”, “Location”, “Date” e “Prize”, na criação da coleção foram eliminados torneios duplicados, ou seja, torneios com o mesmo nome em datas e localizações iguais. Depois, foi verificado se existia com nomes iguais nas mesmas datas, o que se verificou, num caso em que existia dois torneios, em 2006 e 2007, com as mesmas datas mas com diferentes localizações, neste caso aparecia Valência tanto como um bairro em Santa Clarita na Califórnia, como também como a cidade espanhola. Com alguma pesquisa verificou-se que os torneios em ambos os anos decorreram na cidade espanhola, desta forma, os torneios dos EUA foram removidos, o que acaba por ser relevante já que o objetivo do presente trabalho será fazer a previsão do número de sets jogados nos Estados Unidos. O próximo atributo a tratar foi a variável “Prize” para que esta não contivesse a moeda e pudesse ser de *tipo* numérico. Removemos diretamente o símbolo e uniformizamos o campo sem considerar as diferentes moedas já que o prize money nos Estados Unidos é atribuído em dólares e só iremos utilizar as observações relativas a este país. Caso se considere fazer a previsão em países adicionais, será necessário ajustar esta transformação para ter em conta as diferentes moedas. Por fim, o atributo “Date” aparece como uma string com o intervalo de datas em que decorreu o torneio, desta forma, para uma melhor compreensão e utilização deste atributo em fases posteriores, optou-se por separá-lo em “StartDate” e “EndDate”.

Por fim, a coleção “games” foi limpa e onde foram escolhidos os seguintes atributos "PlayerName", "LinkPlayer", "Tournament", "Location", "Date", "Ground", "GameRound", "GameRank", "Oponent", "WL" e "Score". Um dos problemas corrigidos desta coleção foi a limpeza do resultado “bye”, uma situação em que o jogador não necessita de competir numa

ronda inicial de um torneio passando, assim, para a próxima ronda sem jogar qualquer jogo. Deste modo, foi decidido que esta situação não devia contar como vitória e procedeu-se à eliminação deste resultado, já que o número de sets (igual a 0) não iria contribuir positivamente para o desenvolvimento/treino dos modelos, e não se encontrar ajustado aos objetivos do presente trabalho. De seguida, foram limpos todos os jogos duplicados, já que existiam jogos do ponto de vista do vencedor e do ponto de vista do perdedor que no fundo eram o mesmo jogo. Iniciámos o processo com a criação de duas coleções distintas para organizar os dados de vitórias e derrotas dos jogadores. Para garantir a integridade da informação, eliminámos os jogos duplicados das duas coleções, removendo da coleção de derrotas qualquer jogo cujo "PlayerName" e "Oponent" fossem iguais aos da coleção de vitórias, e vice-versa. Após esta filtragem, os jogos que restaram na coleção de derrotas foram adicionados à coleção de vitórias, que renomeámos para "uniquegames". Continuando a limpeza, abordámos o atributo "Ground", desta forma, foram identificados os torneios que careciam desta informação, de seguida, e recorrendo ao site da ITF foi consultado o tipo de piso onde a maioria destes torneios foi disputada. Com esta pesquisa foi possível preencher os dados omissos com a informação correta, completando o tipo de piso em que variados torneios foram disputados. Por fim, tal como foi feito para a coleção “tournaments”, o atributo “Date” da coleção “games” foi tratado, onde se aplicou a mesma lógica anteriormente utilizada.

Por último, de forma a conseguir tratar e limpar outros atributos, todas as coleções criadas foram exportadas para o MySQL. Vale a pena salientar que todos os missings que se verificam em atributos como o “Born” e “Location” levantam problemas, contudo, todo o tratamento de atributos relacionados com localizações, cidades e países será feito na fase da limpeza de dados em MySQL.

MYSQL

Primeiro foram criadas as tabelas “Players”, “Tournaments”, “Games”, “Países/Código” e “Cidades/Países” para colocar os dados exportados de MongoDB e alguns datasets auxiliares para o pré-processamento em falta. Grande parte dos dados foi trabalhada no MongoDB, no entanto, faltavam tratar dois campos importantes: “Born” da tabela “Players” e “Location” da tabela “Tournaments” (este campo assume especial importância já que será necessário filtrar os torneios dos EUA para concretizar o objetivo deste trabalho). Para tal, usaram-se as tabelas “Países/Código”, onde estavam armazenados os nomes dos países e os respectivos códigos ISO, importado do Data Hub (ref.) e “Cidades/Países”, que continha as maiores cidades do mundo, também importado do Data Hub (ref.).

Na coluna “Born”, os valores aparecem dispostos da seguinte forma: “cidade” + “,” + “país”. No entanto, um dos maiores problemas deste campo é quando aparecia apenas o país ou apenas a cidade. Para resolver este problema, começou-se por saber a dimensão total e o número de lugares distintos onde nasceram os jogadores contidos neste dataset obtendo os resultados de 2221 lugares diferentes para os 9960 jogadores existentes. Relembrar que no MongoDB já se tinha descoberto a existência de 6276 valores em falta para este campo. De

seguida, procurou-se saber quantos jogadores têm associado apenas o país ou apenas a cidade, obtendo um valor de 541.

Posteriormente, foi criada a tabela "born_clean" para verificar se todos os países estavam escritos da mesma forma, armazenando na primeira coluna os valores de "Born" da tabela "Players" sem duplicados e na segunda os países limpos. Inicialmente foram colocados os valores que apresentavam cidade + país na primeira coluna com o país correspondente na outra coluna. Com este processo finalizado, foram introduzidos os valores que não continham vírgula (ou só países ou só cidades) como NULL value para serem posteriormente preenchidos.

No resultado obtido, foram corrigidos todos os países que apareciam escritos de formas diferentes (por exemplo "USA" e "U.S.A").

Para tratar dos nulos que surgem apenas quando aparece o nome da cidade, fez-se um comando para fazer a comparação com a tabela cities adicionada ao MySQL, para fazer logo o pré-preenchimento dos valores do país, quando a cidade da 1ª coluna de born_clean fosse igual à cidade da tabela cities.

De seguida, aplicou-se a mesma lógica, mas de forma a procurar na 1ª coluna de born_clean pelo país. Caso existisse correspondência, fazia-se também o pré-preenchimento.

Faltando apenas uniformizar os nomes que aparecem de diferentes formas, recorreu-se a vários comandos update que iam sendo atualizados manualmente e finalmente acrescentou-se um campo na tabela de players relativa ao país ("country"). Os valores para esta nova coluna de países foram atualizados com os valores de born_country da tabela auxiliar que tinha sido criada para fazer a limpeza do campo born. Desta forma obteve-se a tabela final corrigida.

Quanto à variável "Location" da tabela "Tournaments" a abordagem tomada foi bastante semelhante à da coluna "Born", pois a estrutura da tabela é a mesma ("cidade" + "," + "país") com o mesmo problema de apresentar valores só com o país ou só com a cidade.

Começou-se por verificar o número de localizações distintas (2485) e o facto de não existirem valores omissos ao contrário da coluna "Born". Para corrigir os erros desta tabela, criou-se uma tabela auxiliar "location_clean" onde a primeira coluna (location) era igual à location da tabela tournaments, sem duplicados, e a segunda coluna (location_country) corresponde aos países limpos.

Primeiramente foram inseridos os valores com país + cidade com o país correspondente na coluna da frente, seguido dos restantes valores (só países ou só cidades) como NULL value para serem definidos posteriormente.

No resultado obtido, foram corrigidos todos os países que apareciam escritos de formas diferentes (por exemplo "Australia" e " Australia"). Para tratar dos nulos que surgem apenas quando aparece o nome da cidade, fez-se um comando que compara com a tabela cities adicionada ao MySQL, para executar logo o pré-preenchimento dos valores do país, quando a cidade da 1ª coluna de location_clean for igual à cidade da tabela cities. Para os

resultados onde apenas aparecia o país, procurou-se na 1ª coluna de location_clean, onde caso existisse correspondência, fazia-se também o pré-preenchimento.

No que toca a uniformizar os nomes que aparecem de diferentes formas, recorreu-se a vários comandos update que iam sendo atualizados manualmente. Depois, criou-se a coluna de países (Country) na tabela tournaments, que contém os países das localizações dos torneios e inseriram-se os dados limpos da tabela auxiliar location_clean na nova coluna criada (Country) em tournaments.

Importa reforçar que as end_date dos torneios que estão com dummy dates ('0000-00-00') são relativas aos torneios que não têm data final. Da mesma forma, os Prizes que estão a 0 correspondem também a prémios que não estavam anunciados (não necessariamente ao facto de se tratar de um torneio sem prémio associado).

Os países que já não existem como Yugoslavia, Czechoslovakia e Soviet Union, por exemplo, apareciam em poucos torneios (apenas 45). Como tal, quando existia pouca informação sobre o torneio, para tratar estes valores e não deixá-los como nulos, decidiu-se fazer a atribuição de cada um deles ao país atual que estava no seu território com maior área (ex.: Soviet Union passou a representar-se por Rússia) ou consultar diretamente os torneios que aconteceram nesse território para fazer a atribuição do país atual que melhor representava os ex-países.

Com as tabelas prontas, foi criado o modelo relacional para garantir a integridade referencial, nomeadamente ao nível das variáveis "Born" e "Location", já que um dos pontos importantes deste trabalho passa por filtrar os torneios dos EUA .

O modelo criado pode ser observado na Figura 15. O único problema que enfrentamos na criação do modelo foi referente à criação da chave primária composta pela junção das variáveis "Tournament" e "start_date" da tabela "Tournaments", já que surgiu um erro causado pela existência de um torneio duplicado em 2009 sem "end_date". Para resolver, decidimos removê-lo. Após isso, foi efetuado o "enforcement" de keys sem dificuldades.

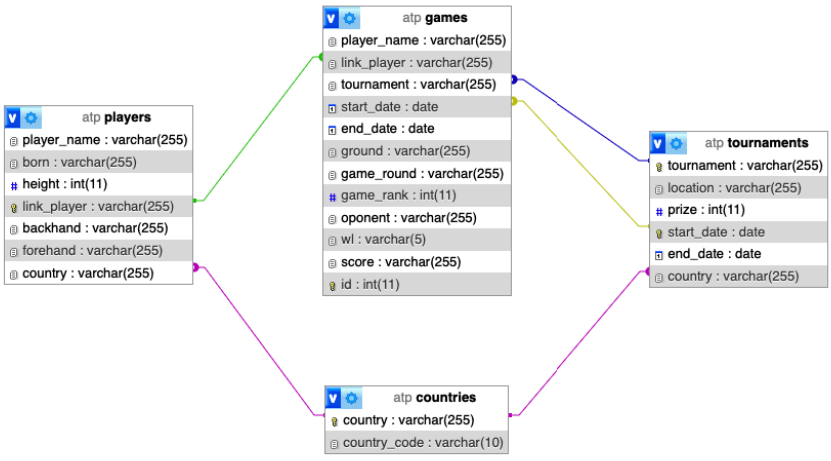


Figura 15 - Modelo relacional em MySQL

PYTHON

Importação e pré-processamento inicial dos datasets

Após o pré-processamento feito nas duas etapas iniciais deste capítulo, primeiro em MongoDB e depois em MySQL, procedeu-se à importação dos datasets já com algum pré-processamento efetuado para Python para terminar a preparação de dados. Assim, procedeu-se à importação do dataset de jogadores (players), com 9960 observações e as colunas referentes às características dos jogadores como o seu país de nascimento, a sua altura, o link para a página de ATP do jogador principal, a respetiva mão com que habitualmente joga e a forma preferencial de fazer a sua esquerda. Importou-se o dataset relativo aos torneios (tournaments), com 22232 registos e com os atributos relativos à localização, prémio, data inicial, data final e país do torneio. E, por fim, procedeu-se também à importação do dataset relativo aos jogos (games) com 683980 registos e a informação do jogador principal, o seu link na página de ATP, o torneio e as respectivas datas de início e fim, o tipo de piso, a ronda, o oponente e o seu rank no momento em que foi disputado o jogo, se foi ganho pelo jogador principal ou não, o respetivo score e uma variável de id.

Com isto em mente, e já com alguns problemas identificados nas fases anteriores de pré-processamento de dados, procedeu-se inicialmente pelo tratamento dos jogadores que tinham exatamente os mesmos nomes - Alberto Gonzalez; Alexey Nesterov; Andreas Weber; Enrique Pena; Mark Kovacs; Martin Damm; Robert Phillips -, onde aqui o que se fez foi ajustar as strings dos nomes nos dataframes relativos aos jogadores e jogos, para que pudessem ter nomes diferentes e para que se pudesse utilizar os nomes como identificadores únicos dos jogadores. É importante frisar que um destes nomes teve particular importância para o presente trabalho - Martin Damm -, já que corresponde ao nome de um atleta da Rep. Checa e a outro atleta dos Estados Unidos, com bastante presença em torneios que se realizam em solo americano, o que reflete a importância desta transformação que foi efetuada.

Data enrichment

Os datasets iniciais utilizados neste estudo apresentavam muitos valores omissos, especialmente nas variáveis relacionadas aos países de nascimento dos jogadores, além de não apresentarem a informação relativa às datas de nascimento dos jogadores. Para mitigar este problema, foi implementado um processo de data enrichment, ou enriquecimento de dados. Data enrichment é o processo de aprimorar a qualidade e a utilidade de um dataset através da adição de informações adicionais ou do preenchimento de valores omissos. Este processo envolve a integração de dados de fontes confiáveis, a validação das informações existentes e, quando necessário, a aplicação de técnicas para estimar ou inferir dados em falta. No contexto deste projeto, utilizámos um repositório do GitHub mantido por Jeff Sackmann, que é amplamente utilizado em múltiplos trabalhos de investigação e projetos open source, como o Tennis Abstract. A confiabilidade e a precisão dos dados deste repositório são amplamente reconhecidas, o que garantiu uma base sólida para o enriquecimento dos nossos dados. (ref.)(ref.)

O dataset adicional utilizado contém 64919 observações relacionadas com jogadores de ténis (não só o top 500 de atp singles) e é constituído por variáveis como o primeiro nome, último nome, mão preferencial, data de nascimento, país (no formato IOC - *International Olympic Committee*) e altura. Nesse sentido, para uniformizar os dados e permitir incorporar a informação deste dataset adicional ao dataset de jogadores, procedeu-se à junção do conteúdo das variáveis do primeiro e último nomes para ficar no mesmo formato, bem como se procedeu à transformação do formato dos países de IOC para ISO, com recurso também às tabelas utilizadas na fase de MySQL.

Junção dos datasets

A junção dos datasets é essencial, já que o objetivo é obter uma visão mais abrangente e completa das informações disponíveis e permitir trazer a informação de todos os datasets para um só que possa ser utilizado para treinar e validar modelos de classificação para prever o número de sets a ser disputado.

Primeiro, começou por se juntar o dataset de torneios com o dataset de jogos com base nas características comuns aos dois, nomeadamente no que diz respeito ao torneio, data de início e data de fim. De seguida, neste novo dataframe (df_games_tourns), verificou-se a existência de valores omissos ao nível do oponente (7 observações) e ao nível do score (77 observações) que foram removidos por não fazerem sentido e pela sua baixa representatividade tendo em conta o volume de dados no dataframe de jogos com 683980 observações. Neste dataframe é importante ainda notar que existiam torneios com datas finais no formato dummy ('0000-00-00') que foram automaticamente aplicadas em MySQL quando foi feita a importação, por terem valores omissos. Após uma análise mais aprofundada, percebeu-se que se tratavam de torneios que não terminaram ou foram interrompidos. Esses casos ocorreram com maior frequência em torneios como o de Yugoslavia F8 de 1998, onde os jogos a partir dos quartos-de-final não chegaram a ser disputados, e o torneio do Cairo de 1986, que também não foi concluído. As tensões e a instabilidade política nas regiões onde esses torneios tomaram parte explicam em grande parte o motivo pelo qual não foram finalizados. Uma vez que esses jogos nunca chegaram a acontecer, decidimos proceder à eliminação dessas observações. Ainda assim, para que não tenham um impacto negativo no treino/teste dos modelos, optou-se por imputar a data final, calculando o valor médio de dias de duração de um torneio (7 dias), que foram depois acrescentados à data inicial destes torneios, de forma a que pudesse ser obtida a data final de cada um.

A junção do dataframe de torneios + jogos (df_games_tourns) com o dataframe relativo aos jogadores é só feita depois de se filtrar os jogos dos Estados Unidos da América, com base no nome dos jogadores (note-se que foram corrigidos os nomes iguais inicialmente, precisamente para que este passo pudesse ser executado e os nomes pudessem ser como keys/identificadores únicos de cada jogador). Após esta junção, ficámos com o df_filtrado_j, com 85823 observações e que conta com a informação dos jogos e dos torneios já descrita anteriormente, bem como a informação relativa aos jogadores, que tanto se aplica ao jogador principal e ao oponente, nomeadamente no que diz respeito às variáveis de nome, link, altura, backhand, forehand, país de nascimento e data de nascimento.

Feature engineering

A fase de feature engineering é fundamental para enriquecer o dataset com informações adicionais e relevantes. Esta fase foi realizada em duas partes: primeiro, foram extraídas novas features de todos os jogos e, de seguida, apenas dos jogos realizados nos Estados Unidos.

i. Todos os torneios

Na primeira etapa de feature engineering, foi calculado o número de jogos disputados entre cada par de jogadores (head-to-head) até à data do torneio. Para isso, concatenaram-se os nomes dos jogadores em cada jogo, ordenando-os de forma consistente independentemente de aparecerem como jogador principal ou como oponente, e o dataframe foi ordenado pela data de início do jogo. Depois, utilizou-se um dicionário para armazenar e contar o número de jogos head-to-head para cada par de jogadores até à data do torneio. Este histórico de confrontos diretos revela-se bastante importante já que pode garantir informação estratégica, onde jogadores com um histórico significativo de jogos entre si podem ter um melhor entendimento das táticas do oponente, influenciando assim o resultado do jogo, e pode permitir obter algumas previsibilidade dos resultados, já que um jogador que consistentemente vence um determinado oponente pode ter uma vantagem psicológica ou técnica que se reflete nas suas performances e consequentemente no número de sets jogados.

Adicionalmente, foi calculado o número de jogos praticados por cada jogador na temporada anterior. Extraindo o ano da data de início de cada jogo, inicializou-se um dicionário para armazenar o número total de jogos por ano para cada jogador. Para cada linha do DataFrame, incrementou-se o número total de jogos tanto para o jogador principal quanto para o oponente. Além disso, foi calculada a diferença absoluta no número de jogos praticados na temporada anterior entre o jogador principal e o oponente, adicionando essa diferença como outra nova feature, que mostra uma relação bastante importante, que pode traduzir o desgaste, onde o número de jogos disputados na temporada anterior se constitui como um indicador do desgaste físico e mental de um jogador. Jogadores que competem em muitos jogos podem estar mais suscetíveis a lesões ou fadiga, afetando o seu desempenho, e consequentemente o número de sets disputados. Por outro lado, jogadores que competem frequentemente podem estar em melhor forma/mais habituados ao ritmo de competição, o que pode ser uma vantagem em torneios subsequentes. A diferença, módulo entre o número de jogos disputados pelo jogador principal e pelo oponente, na temporada anterior pode destacar discrepâncias nas condições físicas e na preparação de ambos. Um jogador menos desgastado pode ter uma vantagem em termos de resistência e recuperação durante o jogo, necessitando por isso de menos sets para vencer.

ii. Apenas torneios dos Estados Unidos da América

Esta segunda etapa teve como objetivo enriquecer o dataset com a criação de novas features com os dados dos torneios dos Estados Unidos. Cada variável criada procura oferecer insights sobre diferentes aspectos do jogo e dos jogadores. Abaixo procuramos explicar o

racional por trás de cada variável criada, bem como a forma como foi implementada (de forma muito breve).

O número de sets jogados em cada jogo (number_of_sets) é o nosso target, que vamos tentar prever na secção seguinte e dá-nos uma medida da intensidade e da duração do jogo, que se reflete diretamente na dinâmica da partida e na performance dos jogadores. Uma partida com mais sets pode indicar um jogo mais equilibrado e disputado. Para calcular o número de sets jogados, foi utilizada uma função lambda que conta o número de elementos na string que representa o resultado de cada jogo.

A duração do torneio em dias (tournament_duration) é importante para entender a extensão e a exigência do evento. Torneios mais longos podem implicar em uma carga física e mental maior para os jogadores, que se pode refletir no seu desempenho ao longo do torneio e consequentemente no número de sets de cada jogo. Foi calculada subtraindo a data de início da data de término do torneio, resultando em um intervalo de tempo em dias.

A idade dos jogadores (player_name_age e oponent_age) considera-se um fator crucial no desempenho físico dos atletas já que jogadores mais jovens tendem a ter mais energia e resistência física, enquanto jogadores mais velhos podem ter mais experiência e habilidades táticas desenvolvidas ao longo dos anos. A diferença de idade entre os jogadores (age_difference) em cada encontro pode refletir diferentes dinâmicas de jogo. A idade dos jogadores foi calculada subtraindo a data de nascimento dos jogadores da data de início de cada torneio, e o resultado foi convertido em anos e a diferença entre as idades foi calculada em módulo.

A diferença de altura entre os jogadores (height_difference) é um fator de grande importância no ténis profissional. Esta diferença pode ter um impacto significativo no decorrer de uma partida, afetando diretamente as estratégias adotadas pelos jogadores. Por exemplo, um jogador mais alto pode ter vantagem em termos de alcance e potência nos golpes, enquanto um jogador mais baixo pode compensar essa diferença com agilidade e velocidade de movimento pela quadra. Esta variável foi calculada como a diferença em módulo das alturas do jogador principal e do oponente.

A vantagem de jogar em casa (player_name_home_advantage e oponent_home_advantage) é um fenómeno muito conhecido e reconhecido em vários desportos. Nos torneios realizados nos Estados Unidos, esta vantagem pode manifestar-se de várias formas, desde o apoio da bancada até à familiaridade com as condições locais, passando pela ausência de viagens extenuantes. Estes fatores podem contribuir significativamente para o desempenho dos jogadores, refletindo-se assim no número de sets jogados.

A comparação entre as diferentes formas de fazer o forehand e backhand entre jogador principal e oponente (forehand_comp e backhand_comp) é fundamental já que as diferenças ou semelhanças entre estes golpes podem revelar muito sobre o estilo de jogo de cada jogador. Por exemplo, um jogador destro pode tentar direcionar o jogo para esse lado do campo, enquanto um jogador com um backhand mais sólido pode procurar explorar essa vantagem em momentos-chave do jogo (considere-se a esquerda a uma mão do Wawrinka,

por exemplo, que ilustra bastante bem este ponto). Portanto, ao considerar estas variáveis, podemos identificar padrões e relações na forma como cada jogador bate uma bola e como tal se reflete num maior ou menor número de sets.

A ronda em que se disputa o jogo (game_round) é uma variável multicategórica com muitas categorias. Nesse sentido, ao agrupar os jogos com base em características semelhantes, podemos simplificar a interpretação e categorização dos dados para se identificar tendências relevantes. Por exemplo, ao comparar o desempenho dos jogadores em diferentes fases do torneio, podemos observar se há uma correlação entre o avanço na competição e o número de sets jogados, já que os à medida que se avança num torneio os jogos tendem a ser cada vez mais competitivos (mais sets jogados), porque o nível competitivo tende a aumentar.

A diferença de rank (rank_difference) é um parâmetro importante para entender o 'gap' de nível entre os dois jogadores. O nível de habilidade de cada jogador está fortemente associado à sua posição no ranking. Por isso, calcular a diferença de rank entre dois adversários pode ser uma mais valia em obter conclusões importantes acerca da competitividade do confronto e também pode ajudar a prever com maior precisão os resultados das partidas. Inicialmente, tínhamos apenas a variável game_rank, que estava associada ao oponente. Por isso, renomeamos esta variável para rank_oponent e introduzimos uma nova variável rank_player, com os valores correspondentes para representar o ranking do outro jogador. A partir dessas duas variáveis, criamos o rank_difference para armazenar a diferença de ranking entre os dois jogadores, em módulo.

Outliers

O processo de tratamento de outliers foi feito em duas fases, nas variáveis baseline e naquelas geradas pelo feature engineering. Ao analisar a descrição do DataFrame filtrado com as variáveis baseline, é possível detetar a presença de outliers nas variáveis "player_name_height", "game_rank", "oponente_height" e "prize". Foi usado o método da distância interquartis, onde um outlier corresponde a um valor que se encontra acima do terceiro quartil com uma diferença superior à multiplicação de 1.5 pelo intervalo interquartil ou abaixo do primeiro quartil pela mesma diferença.

Em baixo encontra-se a tabela 1 com os valores observados destas quatro variáveis.

	mínimo	1º quartil	3º quartil	máximo	observação
prize	0	25000	575000	28619350	Outliers acima
game_rank	0	55	548	2226	Outliers acima
player_name_height	0	178	188	510	Outliers acima e abaixo

oponent_height	0	175	188	510	Outliers acima e abaixo

Tabela 1 - verificação de outliers nas variáveis baseline

Para ter a certeza que estas variáveis apresentam outliers, foram desenvolvidos boxplots (figura 16) referentes aos valores de cada uma.

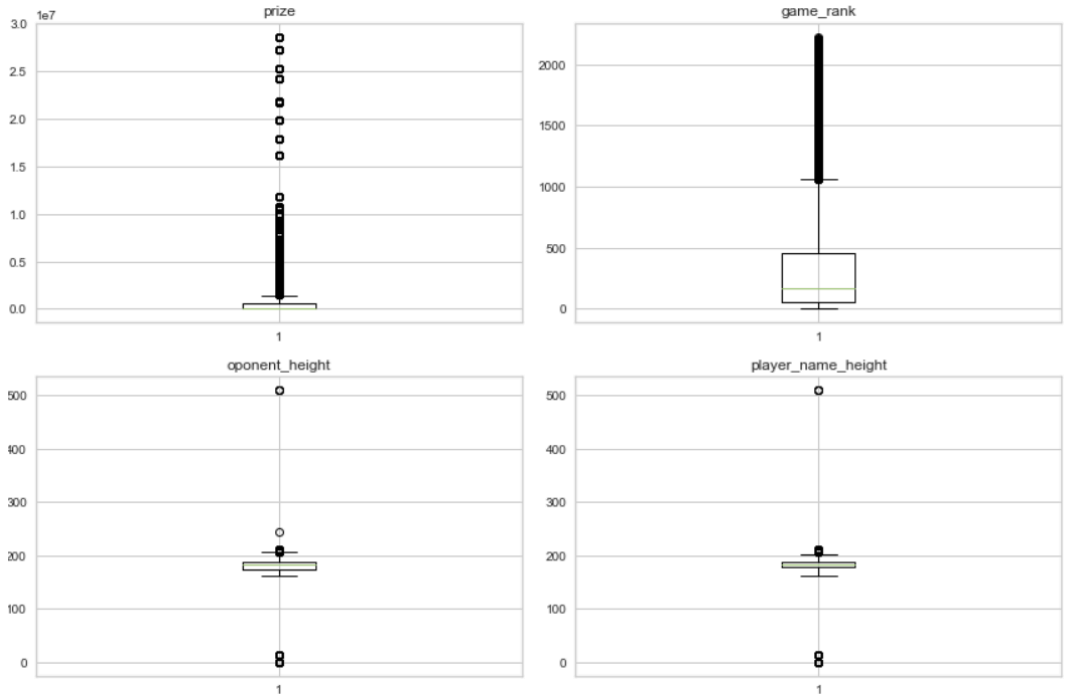


Figura 16 - Boxplots com os outliers das variáveis baseline

Ao observar as imagens, confirma-se a existência de outliers nas quatro variáveis, representados como círculos pretos por cima e por baixo dos limites mínimos e máximos dos boxplots.

Na variável “prize”, estes valores surgem tanto por erros de inserção como pela grande diferença dos prémios do torneio ‘US Open’ em comparação com os restantes torneios. Já nas variáveis “oponent_hight” e “player_name_higth” surgem apenas erros de inserção das alturas. Para a variável “game_rank”, apesar de apenas estarem presentes os jogos dos jogadores do top 500, acabam por aparecer alguns com ranks superiores a esse valor. Isso deve-se ao facto de um jogador dentro desse top poder jogar contra um jogador que não se encontra dentro do top 500, aparecendo assim na base de dados. Uma vez que indivíduos pior classificados não aparecem com tanta frequência no dataset comparado com os melhor classificados, acabam por ser considerados outliers.

Para tratar destes valores, recorreu-se à winsorização. Neste processo, é definido um novo máximo menor que o atual, caso existam outliers por cima dos boxplots, ou um novo mínimo maior que o presente, caso existam outliers abaixo dos boxplots. De seguida, em vez de remover os valores que já não se encontram dentro novo intervalo definido, foram atribuídos novos valores de forma a não impactar de forma negativa o modelo. Se os valores estiverem abaixo do intervalo, passam a ser iguais ao novo mínimo definido, caso estejam acima, ficam iguais ao novo máximo. Relembrar que pode ser necessário tratar outliers tanto acima como abaixo em simultâneo. Quando nas variáveis existe uma grande quantidade de outliers, a sua remoção pode ter um impacto negativo na criação do modelo devido à falta de dados. Como as variáveis deste DataFrame apresentam de facto uma grande quantidade de outliers, a winsorização pareceu-nos o método mais adequado, pois não remove estes valores, mas substitui por outros mais adequados para a fase de modeling. Neste método, os novos limites foram definidos com recurso a alguma experimentação para cada variável, de forma obter boxplots sem outliers.

Pode-se observar na figura 17 o resultado depois da winsorização dos outliers.

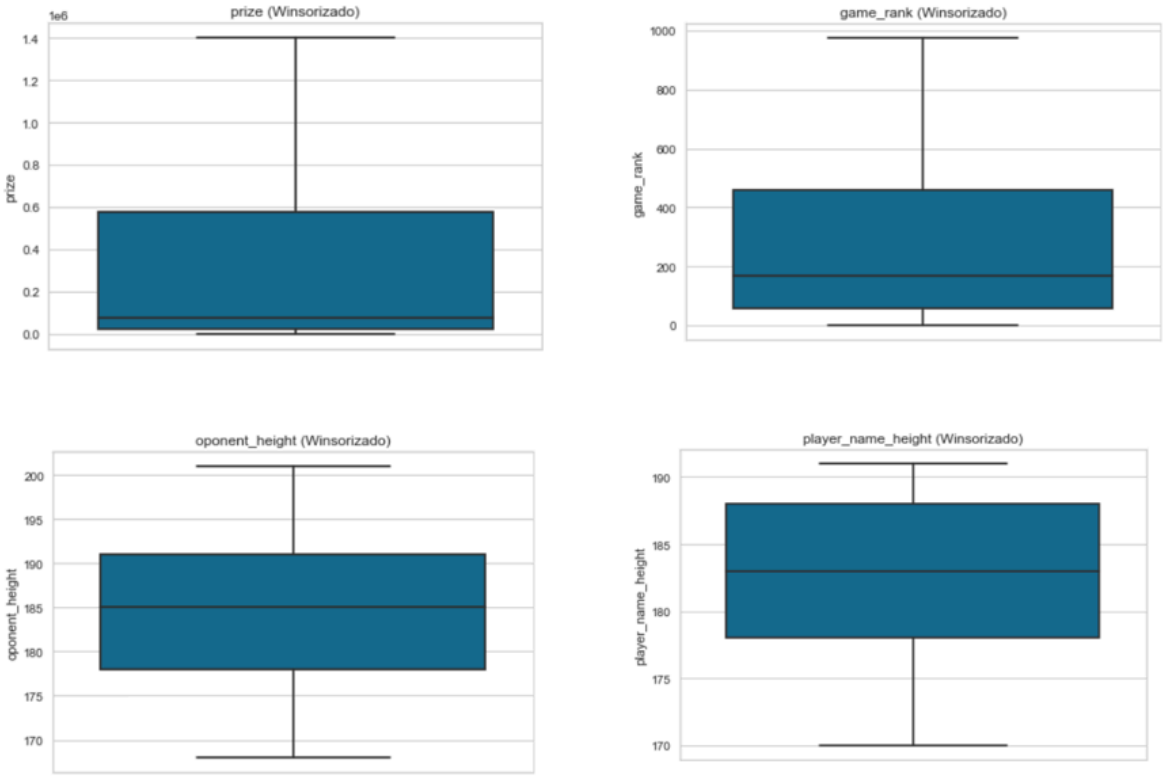


Figura 17 - Boxplot após winsorização de outliers

Após o tratamento dos outliers das variáveis baseline e ser feito o feature engineering, foi necessário tratar dos outliers das variáveis formadas neste processo.

Através da análise do DataFrame com as novas variáveis e usando o método da distância interquartis, foi possível detetar outliers nas variáveis “tournament_duration”

(duração, em dias, de um torneio), “diff_num_games_year_all_tournaments” (diferença do número de jogos realizados num ano entre os jogadores), “age_difference” (diferença de idades entre os jogadores) e “rank_diferece” (diferença de rank entre os jogadores).

Em baixo encontra-se a tabela 2 com os valores observados destas quatro variáveis.

	Mínimo	1º quartil	3º quartil	Máximo	observações
tournament_duration	-1	6	6	13	Outliers acima e abaixo
age_difference	0	2	6	59	Outliers acima
diff_num_games_year_all_tournaments	0	7	33	201	Outliers acima
rank_diference	0	35	298	975	Outliers acima

Tabela 2 - verificação de outliers nas variáveis geradas pelo feature engineering

Para confirmar, foram feitos os boxplot presentes na figura 18

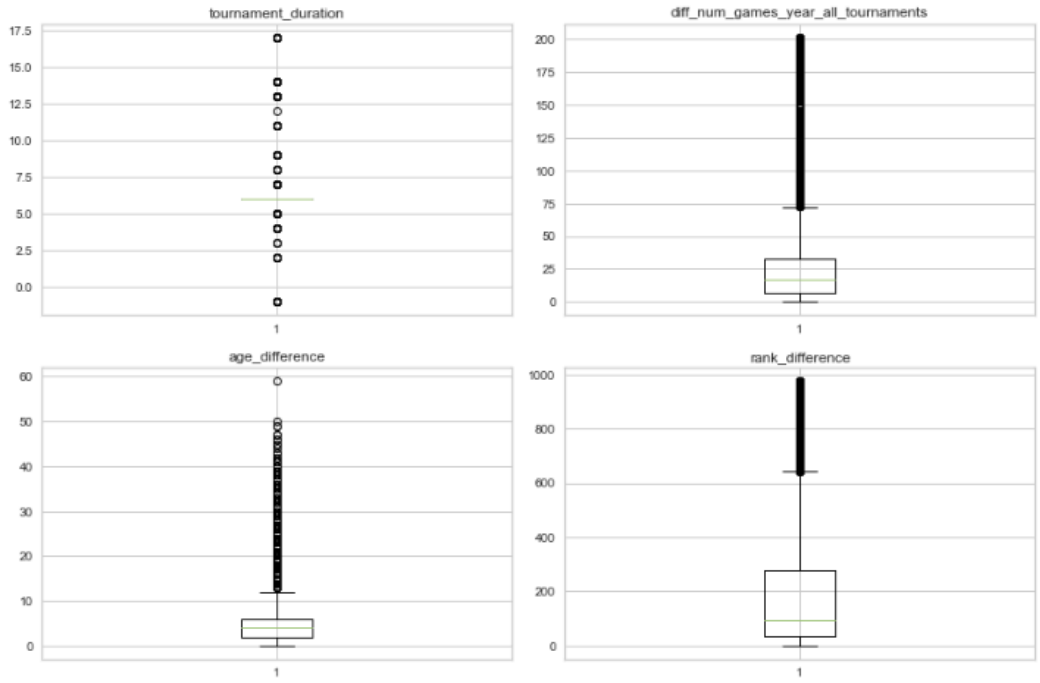


Figura 18 - Boxplot com os outliers das variáveis geradas pelo feature engineering

Analisando os gráficos, é possível observar os outliers, tal como previsto, para as variáveis “rank_difference”, ”age_difference” e “diff_num_games_year_all_tournaments”. Estes surgem devido à existência de alguns jogos em que a diferença de rank, idade ou número de jogos é muito superior ao habitual nesta base de dados (existem muitos mais jogos entre jogadores de ranks semelhantes do que ranks muito diferentes, por exemplo).

Já a variável “tournament_duration” foi revista com recurso a um barplot (figura 19) com as várias durações dos torneios e a conclusão que se tirou foi que a maioria dos torneios apresentava a mesma duração (o histograma serviu para confirmar esta ideia, pois na análise do dataframe o primeiro quartil era igual ao terceiro o que indicava que grande parte dos dados era igual). Desta forma, decidimos não tratar dos outliers desta variável nem usá-la para o treino dos modelos.

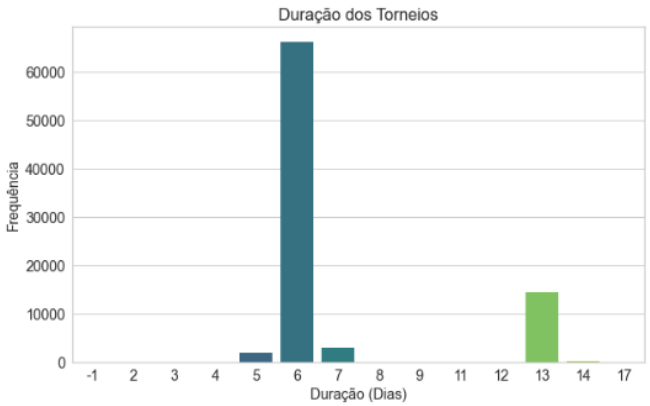
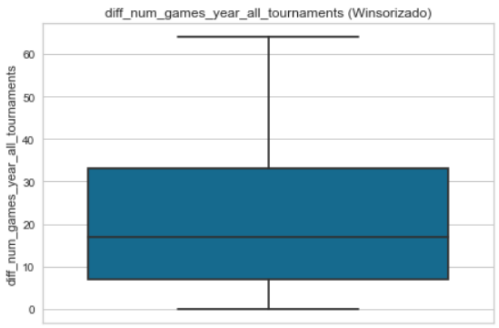
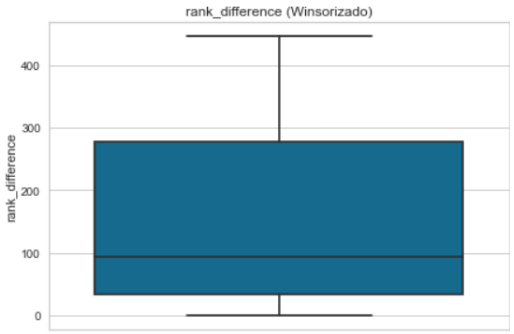


Figura 19 - Histograma da variável tournament_duration

Assim, tal como para as variáveis baseline, recorreu-se à winsorização. Os boxplots que resultaram deste processo encontram-se presentes na figura 20.





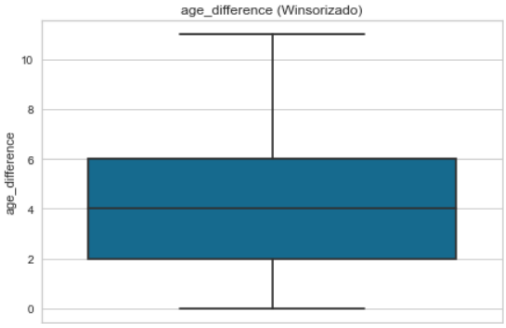


Figura 20 - Boxplot após winsorização de outliers

Valores omissos

O tratamento de valores omissos também foi feito em duas etapas. Primeiro, procurou-se utilizar a base de dados adicional, descrita no capítulo de Data Enrichment, para preencher o máximo de dados possíveis que estivessem omissos. A Tabela 3 sumariza a % de valores omissos das variáveis categóricas e mostra o antes e o após a imputação de valores com recurso a esta base de dados adicional, para se perceber o impacto das transformações efetuadas.

	Missing	%	Mode	Missings after imputation	Mode after imputation
player name country	8897	10,4%	United States (26653)	1916	United States (30186)
oponent backhand	11865	13,8%	Unknown Backhand (33733)	0	Unknown Backhand (45638)
oponent forehand	11865	13,8%	Right- Handed (57596)	0	Right- Handed (57596)
oponent country	23413	27,3%	United States (20090)	5154	United States (30836)

Tabela 3 - Sumário de missings em variáveis categóricas

De seguida, e apesar dos esforços para obter o máximo de informação possível e preencher os valores em falta, alguns ainda permaneceram. Nesse sentido, para tratar estes missing values de forma eficaz, utilizámos o método do KNN Imputer, da biblioteca do sklearn, que permite imputar os valores em falta com base nas similaridades entre os dados existentes. Abaixo, descreve-se detalhadamente o processo implementado para tratar os missing values:

Primeiramente, converteram-se as colunas de datas para o formato datetime, assegurámos que as datas estivessem no formato correto para as operações subsequentes e definiu-se uma data de referência (1900-01-01) que seria utilizada para converter as datas de nascimento dos jogadores em dias desde essa data de referência, simplificando assim o processo de imputação. Note-se que o ano escolhido é inferior a 1927 (data mínima presente no dataset) para evitar que existam valores negativos. Já as variáveis categóricas, tais como os países dos jogadores (player_name_country e oponent_country), foram codificadas utilizando Label Encoding. Este processo transformou as categorias em valores numéricos, para que fosse possível aplicar algoritmos de imputação baseados em distâncias, como o KNN Imputer (que não lida com variáveis categóricas).

As colunas selecionadas para a imputação incluíram variáveis críticas como o país e a altura dos jogadores, além das datas de nascimento convertidas em dias. O subset do DataFrame contendo apenas estas colunas foi então preparado para a imputação (df_to_impute) e com recurso ao KNN Imputer com 5 vizinhos (n_neighbors=5), o que significa que cada valor em falta foi imputado com base nos valores dos 5 vizinhos mais próximos. Este método utiliza a média dos valores dos vizinhos mais próximos para preencher os missing values, proporcionando uma imputação baseada em similaridades. Após isto, os resultados foram convertidos de volta para os tipos de dados que tinham antes da imputação e ajustados no DataFrame.

Data selection

Conforme visto na fase de Business Understanding e Data Understanding, e uma vez que estamos a trabalhar com torneios dos EUA, sendo o nosso objetivo prever o número de sets jogados, é importante considerar que nos EUA há dois grandes tipos de torneios: os grand slams, que são disputados à melhor de 5 sets e os outros torneios que são disputados à melhor de 3 sets. A distribuição destes dois tipos de torneios pode ser vista na Figura 21.

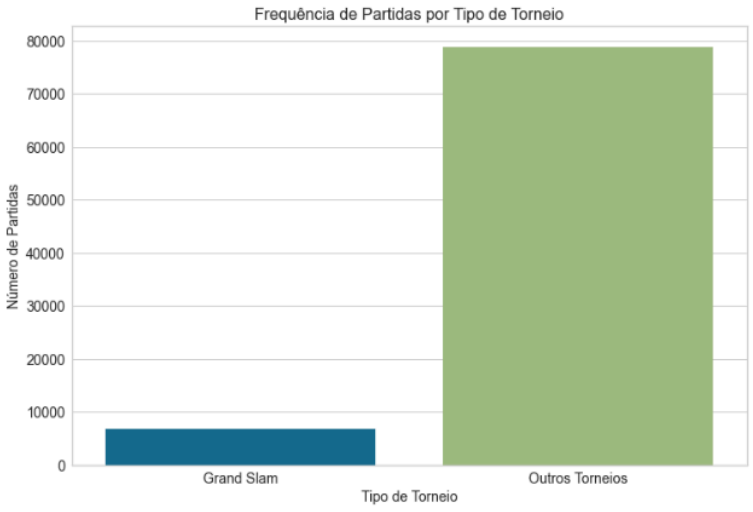


Figura 21 - Distribuição de torneios grand slams e outros torneios

Nesse sentido, daqui em diante vamos focar-nos apenas nos torneios disputados à melhor de três sets. A distribuição do número de sets nos EUA (excluindo Grand Slams) pode ser vista na Fig. 22.

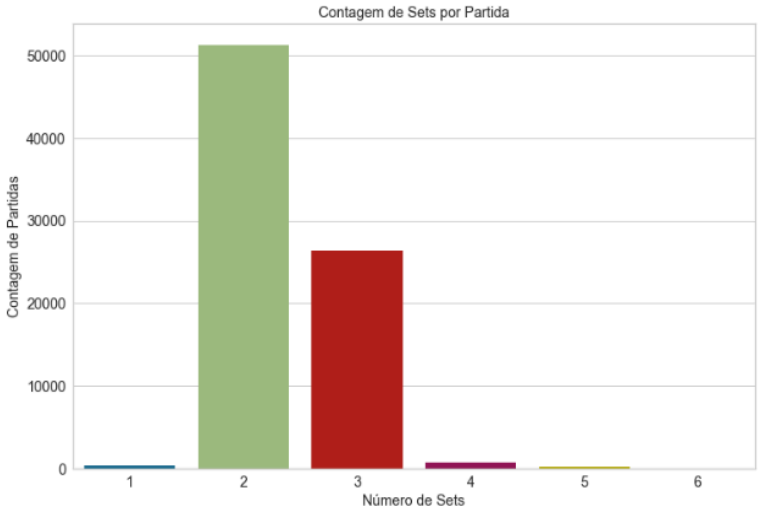


Figura 22 - Distribuição do número de sets nos torneios (excluindo Grand Slam)

Apesar da maioria dos jogos ser de 2 ou 3 sets, verificámos que havia alguns jogos onde eram jogados 1, 4 ou 5 sets. Após analisar esses casos, chegámos a algumas conclusões: Em alguns dos jogos disputados somente em um set, ocorreu o caso de walkover, esse termo indica que um jogador recebeu uma vitória automática e avançou para a próxima fase do torneio sem a necessidade de disputar o jogo, podendo acontecer devido a uma lesão, desistência ou outra circunstância que impeça a realização da partida. Ainda para jogos com apenas um set, verificou-se que o torneio de Denver de 1970 adotou uma abordagem diferente tendo o formato de 'round robin'. Este formato consiste em cada jogador jogar contra todos os outros uma única vez, num único set. O jogador com mais vitórias passa para a fase seguinte. Esta estrutura de jogo é uma exceção à norma e, portanto, os jogos desse torneio não podem ser considerados diretamente comparáveis aos padrões tradicionais dos outros torneios com que estamos a trabalhar. Aconteceu também alguns casos de erro de formatação dos dados e outros exemplos, como os torneios da Davis Cup World Group, o Key Biscayne e o WCT Finals, que adotavam um formato à melhor de cinco sets.

Decidimos proceder à eliminação desses jogos, uma vez que não se encontravam de acordo com a estrutura dos torneios atuais.

Análise de correlações

Na matriz de correlação de Pearson, representada na Fig. 23, é possível observar relações significativas entre algumas variáveis numéricas, no entanto, na maioria das variáveis a correlação entre variáveis é relativamente baixa, o que nos indica que não há

multicolinearidade entre variáveis. Ainda assim, a correlação entre a duração do torneio e do prémio é positiva (0.55), sugerindo que torneios mais longos, geralmente, tem um maior prémio associado. A correlação entre o prémio e diferença de ranking tem uma correlação negativa (-0.30), o que demonstra que quanto maiores as disparidades no ranking entre jogadores estão associadas a um menor prémio. As restantes variáveis exibem correlações mais baixas, sugerindo relações menos diretas entre elas.

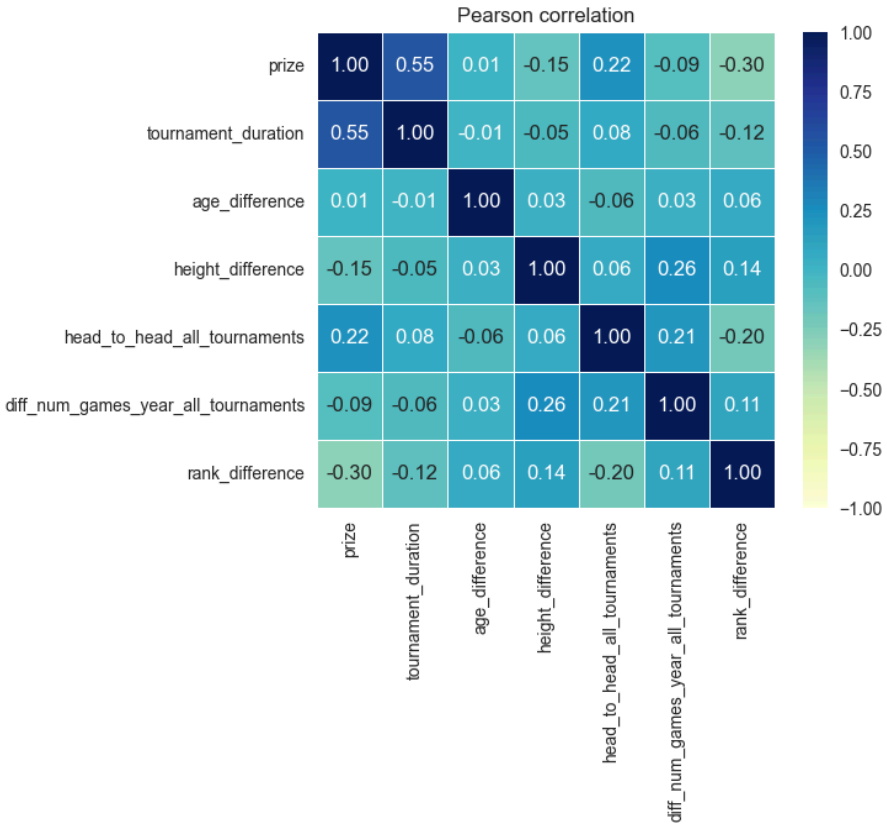


Figura 23 - Matriz de correlação de Pearson para variáveis numéricas

Na mapa de calor referente a correlação do V de Crámer, representado na Fig. 24, é possível observar que, além das correlações fortes entre os tipos de piso, as restantes variáveis categóricas apresentam uma baixa correlação. Como já referido, as maiores correlações estão relacionadas com o tipo de piso, nomeadamente, entre o piso “hard”e o “clay”, esta alta correlação (0.85), é explicada devido ao alto número de observações em torneios realizados nessas superfícies, sugerindo que os jogadores jogam frequentemente nestes dois tipos de superfícies influenciando, assim, a correlação entre estas variáveis. Existe, também, uma correlação alta (0.66) entre as rondas “Mid Rounds” e “Early Rounds”, isto acontece, novamente, pela mesma razão acima mencionada, uma vez que, as “Early Rounds” e as “Mid Rounds” tem um alto número de observações em comparação com as outras rondas.

Concluindo, este padrão de baixas correlações observado no mapa de calor indica que as variáveis analisadas têm influências complexas e multifatoriais nos torneios de ténis,

sugerindo que a previsão do número de sets jogados não pode ser explicada por apenas alguns fatores isolados. Assim, é importante reforçar que esta previsão é o resultado da combinação de múltiplas variáveis, o que demonstra a complexidade envolvida nesta análise, exigindo uma abordagem mais abrangente e detalhada.

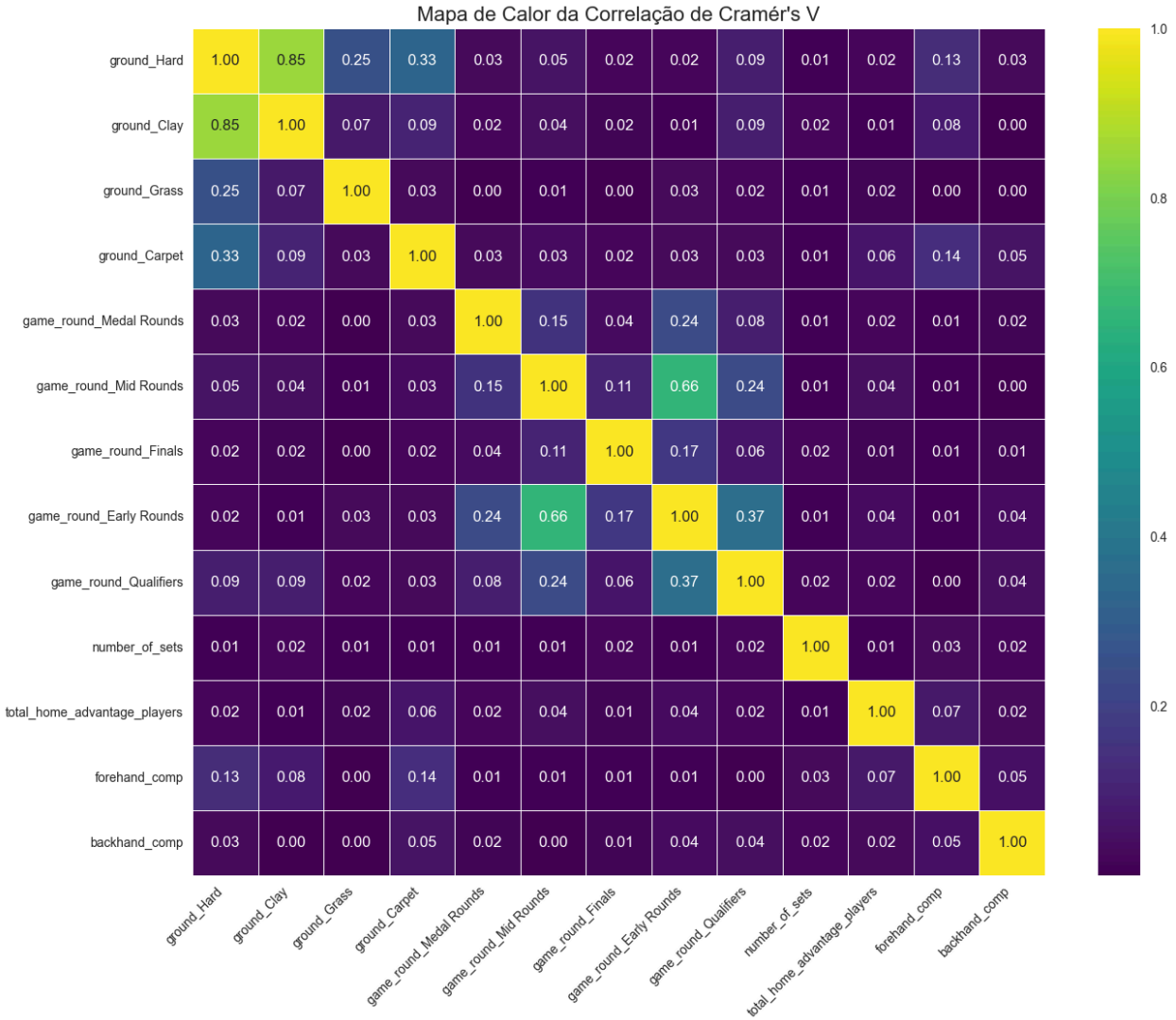


Figura 24 - Matriz de correlação de V de Cramér para variáveis categóricas

O VIF (Variation Inflaction Factor) é uma medida que permite observar o grau de multicolinearidade de variáveis independentes de um modelo de regressão. Desta forma, valores de VIF acima de 5 sugerem uma multicolinearidade forte, o que pode tornar as previsões menos precisas. Como é possível observar na Tabela 4, todos os valores das variáveis independentes estão abaixo de 5 o que indica baixa multicolinearidade, sugerindo que as variáveis utilizadas não apresentam redundância excessiva, podendo assim ser utilizadas nos nossos modelos.

Variable	VIF
prize	1.635115
tournament_duration	1.439548
head_to_head_all_tournaments	1.156359
rank_difference	1.153649
diff_num_games_year_all_tournaments	1.140428
height_difference	1.104325
height_difference	1.009633

Tabela 4 - Análise do VIF

MODELING

Com os dados devidamente pré-processados, passamos para a fase de modelação. Nesta etapa, utilizaram-se modelos de classificação para prever o número de sets nos jogos. A escolha de modelos de classificação é motivada pela natureza categórica do target, número de sets, que pode variar entre diferentes valores discretos (2 ou 3 sets - no caso dos torneios dos EUA, excluindo US Open, por se tratar de um torneio de Grand Slam, disputado à melhor de 5 sets). Note-se que em todo este capítulo focamo-nos apenas nos torneios dos EUA à melhor de 3 sets. A Fig. 25 sumariza as etapas a seguir na etapa de Modeling.

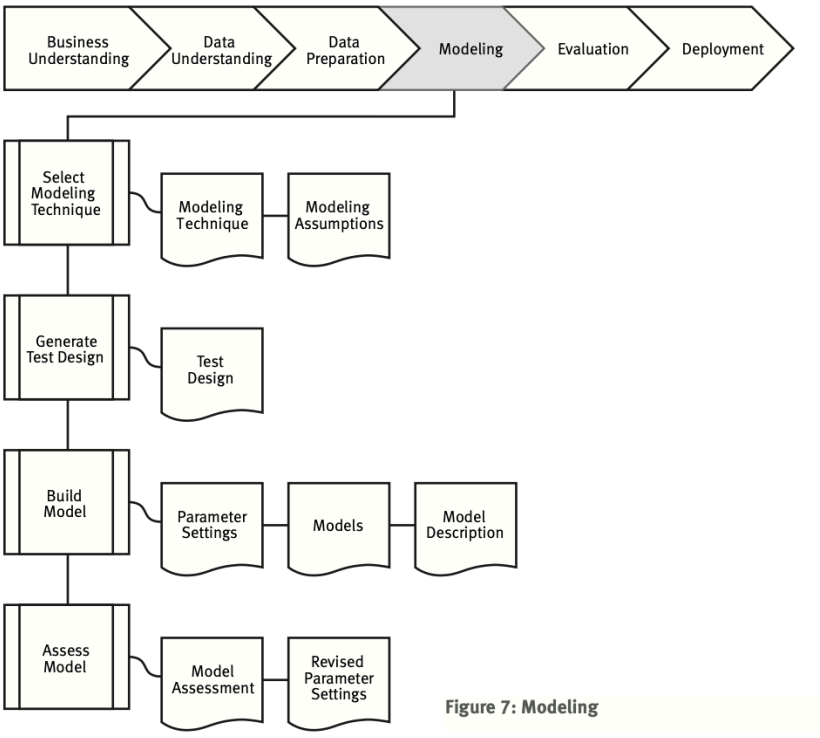


Figura 25 - Descrição das tarefas associadas à fase de Modeling

Para simplificar a modelação, os valores associados ao número de sets foram codificados em valores binários: **0 para jogos com 2 sets e 1 para jogos com 3 sets**. Será utilizada tanto a validação cruzada (cross-validation) quanto a divisão de treino e teste (train-test split) para avaliar o desempenho dos modelos. Na validação cruzada, o conjunto de dados é dividido em k folds, onde o modelo é treinado em k-1 folds e testado no fold restante. Isto permite-nos avaliar o desempenho do modelo de forma mais robusta, pois testamos o modelo em diferentes conjuntos de dados. Já na divisão de treino e teste, o conjunto de dados é dividido em um conjunto de treino, usado para treinar o modelo, e um conjunto de teste, usado para avaliar o seu desempenho. Nas abordagens que envolvem cross-validation, foram sempre utilizados 10 folds porque é uma escolha comum e equilibrada que fornece uma boa estimativa do desempenho do modelo sem exigir um tempo de computação

excessivamente longo. A utilização de 10 folds permite uma divisão do conjunto de dados em 10 partes iguais, onde o modelo é treinado em 9 partes e testado na parte restante, repetindo o processo 10 vezes. Esta abordagem proporciona uma avaliação robusta do desempenho do modelo, reduzindo a variabilidade nos resultados.

Quanto à abordagem de treino e teste split, optámos por utilizar 80% dos dados para treino e 20% para teste porque esta divisão permite treinar o modelo em uma quantidade significativa de dados enquanto ainda mantém uma quantidade suficiente de dados de teste para avaliação. Além disso, foi aplicado um split estratificado devido à natureza desbalanceada dos dados, o que significa que a distribuição das classes no conjunto de treino e teste reflete a distribuição original das classes.

Ao analisar a distribuição do target no conjunto de treino (para a abordagem de treino e teste split), verificou-se um **desequilíbrio significativo** entre as classes. Existem 40551 jogos com 2 sets (representados pelo valor 0) e 20219 partidas com 3 sets (representados pelo valor 1), conforme se pode ver na Figura 26. Este desequilíbrio pode dificultar a performance dos modelos de classificação, que tendencialmente vão acabar a favorecer a classe maioritária.

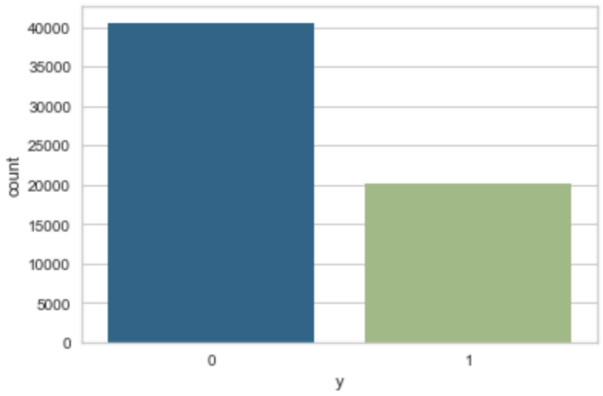


Figura 26 - Distribuição do target no conjunto de treino

Para lidar com este desequilíbrio, procurou-se aplicar técnicas de oversampling e undersampling. Oversampling consiste em aumentar o número de exemplos da classe minoritária (neste caso, jogos com 3 sets) replicando-os. Por outro lado, o undersampling consiste em reduzir o número de exemplos da classe maioritária (jogos com 2 sets). Estas técnicas ajudam a ajustar a distribuição das classes no dataset, permitindo que os modelos aprendam a identificar com maior precisão as características que diferenciam jogos com 2 sets de jogos com 3 sets. De qualquer forma, importa reforçar que já se antecipa que os resultados não vão ser muito promissores, já que a análise de correlações, elaborada no final do capítulo anterior, entre variáveis numéricas e categóricas, mostrou relações com pouca força e pouco significativas.

A aplicação de oversampling foi realizada por intermédio do SMOTE (Synthetic Minority Oversampling Technique), que gera exemplos sintéticos da classe minoritária em vez de simplesmente replicar os que já existem, enquanto que a aplicação de undersampling foi realizada ao remover exemplos da classe maioritária de forma aleatória, de forma a garantir

que o dataset resultante não perdesse a diversidade necessária para a generalização dos modelos. A Figura 27 ilustra a distribuição do target no training set após a aplicação do SMOTE (para a aplicação de treino e test split). Note-se que como temos também abordagens baseadas em validação cruzada, as técnicas de oversampling e undersampling são aplicadas a todo o dataset, traduzindo-se assim em estimativas mais otimistas, quando consideramos as métricas associadas à performance dos vários modelos treinados e testados.

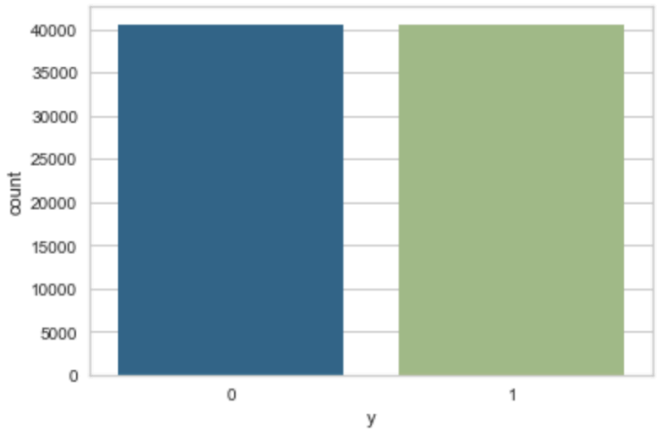


Figura 27 - Distribuição do target no conjunto de treino após a aplicação do SMOTE

Ao aplicar estas técnicas de balanceamento, procurou-se melhorar a capacidade dos modelos de classificação em identificar padrões nos dados, resultando em previsões mais precisas e equilibradas. Modelos treinados com datasets balanceados são geralmente mais robustos e têm um desempenho melhor no que respeita às métricas de precisão, recall e F1-score para ambas as classes. Desta forma, procurou-se aumentar a "equidade" dos modelos, prevenindo que favorecessem excessivamente jogos com 2 sets, e proporcionando previsão mais "justa" e "completa" dos jogos com 3 sets.

Importa também reforçar que antes do treino dos modelos, foi feita a normalização dos dados por intermédio do método Min-Max que ajudou a colocar todas as features nas mesmas escalas e a garantir uma amplitude similar entre todas para que contribuíssem equitativamente para o treino dos modelos. No caso específico do método Min-Max, cada feature foi transformada de forma a que o seu intervalo de valores fosse mapeado para um intervalo entre 0 e 1.

Modelos desenvolvidos

i. Regressão Logística (Minimum Viable Model)

O primeiro modelo desenvolvido seguiu a abordagem de um Minimum Viable Model (MVM) e tratou-se de uma regressão logística, já que o MVM representa uma abordagem que se foca num modelo funcional com o mínimo de recursos necessários para resolver o problema específico em questão. A regressão logística trata-se de um modelo de classificação binária amplamente utilizado em problemas, onde o target é categórico e

binário, ou seja, existem apenas duas classes. Neste caso, a regressão logística é aplicada para prever se um jogo terá dois sets (classe 0) ou três sets (classe 1) com base nas features explanadas no final da etapa de Data Preparation.

A regressão logística é uma técnica estatística que procura modelar a relação entre uma variável dependente binária e várias variáveis independentes, ao estimar as probabilidades associadas a cada classe de saída utilizando uma função logística e, em seguida, aplicando um limite de decisão para classificar as observações numa das duas classes. Para garantir a otimização e pesquisa eficiente pelos hiperparâmetros do modelo, optámos por utilizar neste caso o Randomized Search (com cv=5 e nº de iterações=10), que permitiu explorar/testar uma ampla gama de valores de hiperparâmetros de forma eficiente, garantindo um tempo de tuning razoável, que foi crucial para a implementação do MVM. A tabela 5 sumariza os valores testados e os hiperparâmetros definidos.

Hiperparâmetro	Descrição	Valores testados	Valor ótimo
C	Inverso da força de regularização (valores mais baixos representam regularizações mais fortes)	0.001 / 0.01 / 0.1 / 1 / 10	0.1
penalty	Tipo de regularização	'l1' / 'l2'	'l2'
solver	Algoritmo de otimização	'liblinear' / 'saga'	'liblinear'

Tabela 5 - Tabela de hiperparâmetros

ii. XGBoost

O XGBoost é uma técnica de aprendizagem supervisionada que utiliza um modelo de boosting de gradientes. Este modelo baseia-se no conceito de boosting, ou seja, aprimorar um modelo fraco combinando-o com outros modelos fracos de forma a produzir um modelo coletivamente forte. O boosting de gradientes é uma extensão deste conceito, onde o processo de gerar modelos fracos de forma aditiva é estruturado como um algoritmo de descida de gradiente aplicado a uma função objetivo. Assim, este modelo define metas de desempenho para o próximo modelo na tentativa de minimizar os erros, com as metas a serem baseadas no gradiente do erro em relação às previsões anteriores.

Desta forma, e sendo o XGBoost uma técnica que utiliza o modelo descrito acima, este baseia-se na construção sequencial de árvores de decisão, onde cada nova árvore é criada de forma a corrigir os erros da anterior. Esta técnica inclui, ainda, um procedimento de regularização para controlar a complexidade do modelo, que contribui para evitar o

sobreajuste, o que garante um melhor desempenho do modelo (NVIDIA, n.d.). No entanto, contrariamente à regressão logística, foi utilizado o GridSearchCV de forma a garantir uma pesquisa eficiente pelos hiperparâmetros do modelo, ao executar uma pesquisa exaustiva sobre um espaço de parâmetros especificados, sendo que, cada combinação de parâmetros é avaliada usando validação cruzada, e a melhor combinação é selecionada com base no desempenho (neste caso definiu-se a ROC AUC, porque tanto queremos captar a classe positiva (3 sets) como a classe negativa (2 sets)). A tabela 6 sumariza os valores testados e os hiperparâmetros definidos.

Hiperparâmetro	Descrição	Valores testados	Valor ótimo
objective	Define a função objetivo usada para treinar o modelo	'binary:logistic	'binary:logistic'
n_estimators	Número de árvores a serem construídas no modelo.	[50, 100, 150, 200]	200
max_depth	Profundidade máxima de cada árvore.	[2, 3, 4]	4
learning_rate	Taxa de aprendizagem que controla o peso das correções em cada etapa do boosting	[0.05, 0.1, 0.2, 0.3]	0.3
subsample	Fração das amostras a serem usadas para treinar cada árvore	0.8	0.8
colsample_bytree	Fração de colunas (features) usadas por cada árvore	0.8	0.8
min_child_weight	Soma mínima dos pesos das amostras necessárias num nó filho	[1, 2, 3, 4]	3

Tabela 6 - Tabela de hiperparâmetros

iii. MLP Classifier

O MLP (Multi-Layer Perceptron) é uma técnica de aprendizagem supervisionada baseada em redes neurais artificiais, frequentemente utilizada para problemas de classificação e regressão (Edpuganti, n.d.). A principal característica do MLP é a sua capacidade de aprender e modelar relações complexas e não lineares entre as variáveis de entrada e saída.

O MLP consiste em três tipos principais de layers:

- Input layer: Recebe os dados de entrada. Cada neurónio nesta camada representa uma característica (feature) dos dados.
- Hidden layer(s): Uma ou mais layers localizadas entre a input layer e a output layer. Estas layers realizam a maior parte do processamento e da aprendizagem, capturando inclusivé relações não lineares nos dados. Cada neurónio numa hidden layer aplica uma função de ativação à soma ponderada das suas entradas.
- Output layer: Produz as previsões finais. O número de neurónios nesta layer corresponde ao número de classes (para problemas de classificação), como é o nosso caso, ou a uma única saída (para problemas de regressão).

Assim como no XGBoost, foi utilizado o GridSearchCV de forma a garantir uma pesquisa eficiente pelos hiperparâmetros do modelo. Cada combinação de parâmetros foi avaliada usando validação cruzada (para ROC-AUC), e a melhor combinação foi selecionada com base no desempenho. A tabela 7 sumariza os valores testados e os hiperparâmetros definidos.

Hiperparâmetro	**Descrição**	**Valores testados**	**Valor ótimo**
hidden_layer_sizes	Número de neurônios em cada camada oculta	[(50,), (100,), (150,)]	(100,)
activation	Função de ativação para as camadas ocultas	['relu']	['relu']
solver	Algoritmo para otimização dos pesos	['adam']	['adam']
alpha	Parâmetro de regularização	[0.0001, 0.001, 0.01]	0.0001
learning_rate	Taxa de aprendizado	['constant']	['constant']
learning_rate_init	Taxa de aprendizado inicial	[0.05, 0.1, 0.2, 0.3]	0.05
max_iter	Número máximo de iterações	[50, 100, 150, 200]	50

Tabela 7 - Tabela de hiperparâmetros

iv. Random Forest

A Random Forest é uma técnica de aprendizagem supervisionada baseada em ensemble learning, que é frequentemente utilizada para problemas de classificação. A principal característica da Random Forest é a sua capacidade de construir múltiplas árvores de decisão durante o treino e de produzir a classe que é a moda das classes de cada árvore individual. Para o efeito utilizou-se o RandomForestClassifier e o tuning dos parâmetros foi feito também com recurso a Grid Search CV. A tabela 8 sumariza os valores testados e os hiperparâmetros definidos.

Hiperparâmetro	Descrição	Valores testados	Valor ótimo
n_estimators	Número de árvores na floresta	[10, 20, 30, 40, 80, 100]	100
max_depth	Profundidade máxima permitida para cada árvore	[2, 3, 4, 6, 8]	8
min_samples_split	Número mínimo de amostras necessário para dividir um nó	[2, 4, 6, 10, 20]	2

Tabela 8 - Tabela de hiperparâmetros

EVALUATION

Para a avaliação dos modelos, conforme já referido na secção anterior, procedeu-se ao treino e validação de três formas distintas: por um lado com treino e teste split + oversampling; por outro com cross-validation + oversampling; e por fim cross-validation + undersampling. A Figura 28 ilustra as etapas que seguimos para avaliação dos modelos com a configuração descrita no capítulo de Modeling nos três cenários indicados.

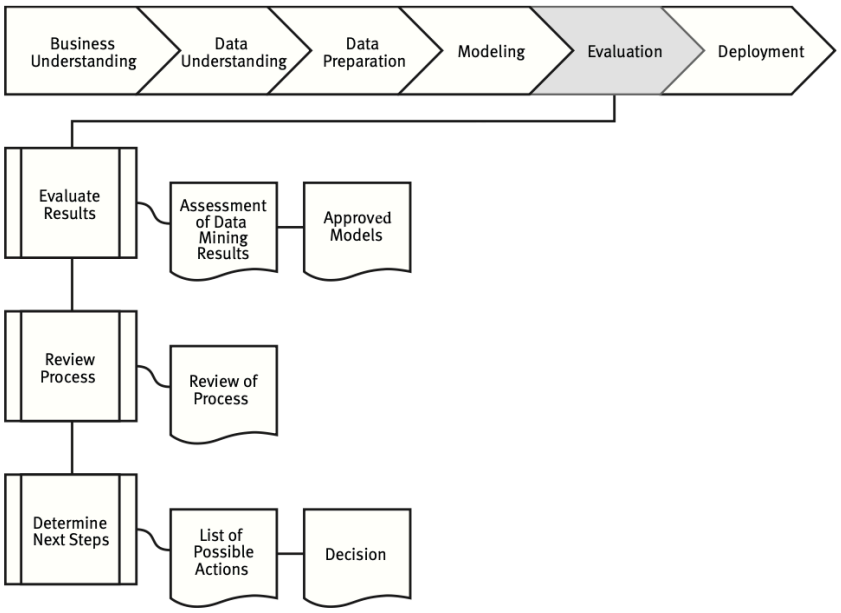


Figura 28 - Descrição das tarefas associadas à fase de Evaluation

i. Treino e test split (c/ oversampling)

A avaliação dos modelos Regressão Logística, XGBoost, MLP Classifier e Random Forest que usaram SMOTE para lidar com o desbalanceamento da variável target ("number_of_sets"), revela diferenças significativas em termos de desempenho entre os conjuntos de treino e teste, desempenho este que pode ser observado na tabela 9.

	Train set				Test set			
Model	**Accuracy**	**Precision**	**Recall**	**F1 Score**	**Accuracy**	**Precision**	**Recall**	**F1 Score**
Regressão Logística	0.674940	0.903848	0.391532	0.546381	0.655433	0.371429	0.051434	0.090356

XGBoost	0.714557	0.874424	0.501073	0.637079	0.638584	0.358257	0.109001	0.167147
MLP Classifier	0.675754	0.830566	0.441592	0.576613	0.641743	0.359827	0.098516	0.154682
Random Forest	0.679453	0.777812	0.502429	0.610503	0.625485	0.368530	0.176063	0.238286

Tabela 9 – Análise comparativa dos modelos com oversampling (treino e teste split)

As métricas revelam uma queda acentuada na precisão e no recall nos dados de teste, indicando problemas de generalização, embora tenha sido feito oversampling. A Regressão Logística apresenta uma queda drástica na precisão de 0.904 no treino para 0.371 no teste, com baixo recall em ambos. O XGBoost segue a mesma tendência, mostrando um ligeiro overfit, com a precisão a cair de 0.874 no treino para 0.358 no teste, e um recall baixo. O MLP Classifier também apresenta uma alta precisão no treino (0.831), mas cai para 0.360 no teste, com um recall extremamente baixo (0.099). Todos os modelos têm dificuldades em identificar corretamente amostras da classe minoritária (jogos de 3 sets), mas o Random Forest apresenta um desempenho ligeiramente melhor em recall e F1 Score nos dados de teste, com valores de 0.176 e 0.238, respetivamente. O XGBoost também mostra um desempenho relativamente melhor em recall (0.109) e F1 Score (0.167) quando comparado aos outros modelos, exceto pelo Random Forest.

ii. Validação cruzada (cross-validation)

Após a realização de uma abordagem com treino e teste split, que revelou algum overfit, foi feita uma outra abordagem utilizando Cross-Validation com oversampling, de forma a perceber melhor o comportamento dos modelos em várias iterações/rondas de validação e compreender as capacidades de generalização dos modelos treinados. Neste processo, os dados foram divididos em dez folds igualmente distribuídos, de seguida, foi feito um ciclo de validação cruzada, onde em cada ciclo um fold é usado como teste enquanto os outros nove folds são usadas para treino, este processo de iteração de ciclos repete-se até cada um dos fold ter sido utilizado como set de teste. De salientar que, a decisão de usar dez folds resultou de um processo de experimentação onde foram testados até trinta folds. Desta forma, foi possível garantir a melhor configuração para que os resultados fossem os melhores possíveis. Assim, os resultados obtidos estão presentes na tabela 10.

	Train set				Test set			
Model	**Accuracy**	**Precision**	**Recall**	**F1 Score**	**Accuracy**	**Precision**	**Recall**	**F1 Score**
Regressão Logística	0.673236	0.543941	0.900096	0.389732	0.673134	0.543784	0.899853	0.389631
XGBoost	0.711157	0.631630	0.871608	0.495271	0.692192	0.607088	0.839119	0.475606
MLP Classifier	0.674158	0.568470	0.846265	0.430931	0.674130	0.568716	0.845969	0.431178
Random Forest	0.676933	0.602478	0.782961	0.489644	0.667393	0.590827	0.767542	0.480321

Tabela 10 – Análise comparativa de todos os modelos (validação cruzada com oversampling)

Analisando a tabela, a regressão logística apresenta uma grande estabilidade entre os dados de treino e teste, com variações mínimas nas métricas de precisão e recall, o que indica uma generalização adequada. O XGBoost, apesar de um leve decréscimo na precisão e recall mantém um bom desempenho geral. O MLP Classifier também apresenta consistência entre treino e teste, com pequenas flutuações em todas as métricas, o que reflete uma boa capacidade de generalização. O Random Forest mostra um bom equilíbrio entre precisão e recall, com desempenho estável entre treino e teste, embora com um recall ligeiramente inferior ao XGBoost. No entanto, todos os modelos enfrentam desafios em equilibrar precisão e recall, especialmente para a classe minoritária (3 sets), ainda assim, o XGBoost é ligeiramente melhor que os outros dois modelos.

De seguida, foi seguida a mesma abordagem com Cross-Validation mas desta vez com undersampling. Desta forma, os resultados estão presentes na seguinte tabela 11.

	Train set				Test set			
Model	**Accuracy**	**Precision**	**Recall**	**F1 Score**	**Accuracy**	**Precision**	**Recall**	**F1 Score**
Regressão Logística	0.531161	0.556890	0.527917	0.589240	0.530704	0.556113	0.527570	0.587997
XGBoost	0.617930	0.635043	0.607820	0.664833	0.527301	0.547563	0.525060	0.572131

MLP Classifier	0.521287	0.501996	0.558654	0.646231	0.517132	0.497674	0.517493	0.640560
Random Forest	0.582034	0.620958	0.568081	0.684793	0.533790	0.576692	0.528083	0.635238

Tabela 11 – Análise comparativa de todos os modelos (validação cruzada com undersampling)

Com a análise da tabela é possível concluir que a regressão logística demonstra consistência entre os conjuntos de treino e teste, com variações bastante pequenas em precisão e recall, o que indica uma generalização adequada. O XGBoost, apesar de um pior desempenho no conjunto de teste em relação ao conjunto de treino, apresenta um bom F1 Score, o que demonstra uma boa capacidade de generalização. O MLP Classifier mostra variações mínimas nas métricas, mantendo uma boa generalização. Por fim, o Random Forest mostra um desempenho consistente entre conjuntos de treino e teste. No entanto, tal como na tabela anterior, tanto a abordagem com oversampling como a abordagem com undersampling, os modelos utilizados continuam com dificuldade na previsão de jogos com 3 sets (classe minoritária). Apesar disso, o XGBoost continua a destacar-se ligeiramente dos outros modelos em termos de desempenho geral.

iii. Performance do Minimum Viable Model

A matriz de confusão da regressão logística, com treino e teste split, está ilustrada na imagem 29 (test set) e mostra que o modelo não está com uma boa performance no que respeita à previsão de 3 sets (classe 1) com base nas features disponíveis. O baixo recall indica que o modelo não identifica muitos dos casos positivos reais (3 sets previstos e reais). Adicionalmente, a precisão relativamente baixa sugere também que muitas das previsões positivas (sets=3 | classe 1) do modelo são incorretas.

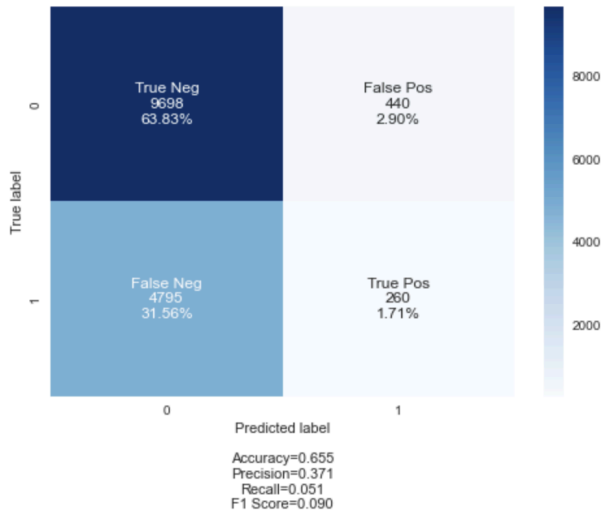


FIgura 29 - Matriz de confusão da regressão logística

A ROC-AUC (Area Under the Receiver Operating Characteristic Curve) é uma métrica que avalia o desempenho do nosso MVM em todos os limiares de decisão possíveis. Uma pontuação de 0,5 sugere que o modelo é tão bom quanto uma previsão aleatória, enquanto uma pontuação acima de 0,5 indica um desempenho melhor do que aleatório e abaixo de 0,5 indica um desempenho pior do que aleatório. Uma pontuação de 0,52 indica que este modelo tem dificuldade em distinguir as classes positiva (3 sets) e negativa (2 sets) de forma significativa, sendo quase aleatório, conforme se pode ver pela análise da figura 30.

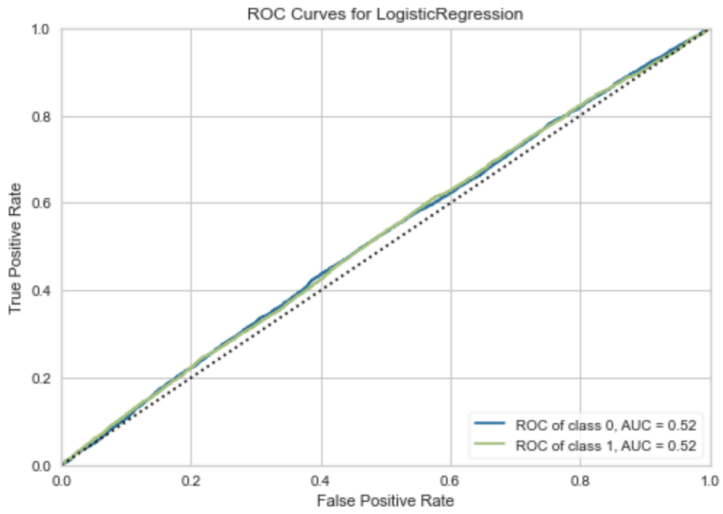


Figura 30 - ROC AUC da regressão logística

iv. Performance do melhor modelo selecionado

O melhor modelo, pela análise das métricas já discutidas anteriormente, foi o XGBoost com oversampling devido à melhor capacidade de generalização e métricas associadas face aos restantes modelos. A matriz de confusão do conjunto de teste do modelo, ilustrada na figura 31, mostra que o modelo permite identificar corretamente 4606 instâncias como sendo de 2 sets e 2411 instâncias como sendo de 3 sets. No entanto, cometeu erros ao classificar 462 instâncias incorretamente como 3 sets e 2658 instâncias como sendo de 2 sets. As métricas de desempenho são as seguintes: a accuracy é de 0.692, a precision é de 0.839, o recall é de 0.476 e o F1 Score é de 0.607. Estes resultados indicam que, apesar de o modelo ter uma boa precision, o recall é relativamente baixo, sugerindo que o modelo perde muitas instâncias reais de 3 sets, mas sendo mais eficaz a evitar falsos 3 sets.

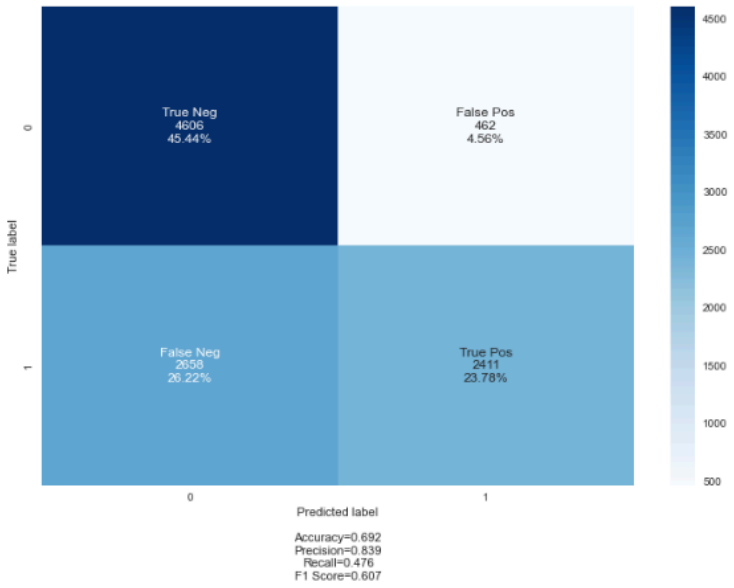


Figura 31 - Matriz de confusão do XGBoost com oversampling (cross-validation)

A figura 32 ilustra a feature importance das variáveis no modelo. Observa-se que a variável mais importante é "game_round_Qualifiers", seguida de "game_round_Medal Rounds" e "game_round_Mid Rounds". Outras variáveis significativas incluem "game_round_Early Rounds", "ground_Clay" e "game_round_Finals", indicando que tanto a fase do torneio quanto o tipo de superfície do campo (e logo a seguir o fator casa) influenciam significativamente as previsões do número de sets.

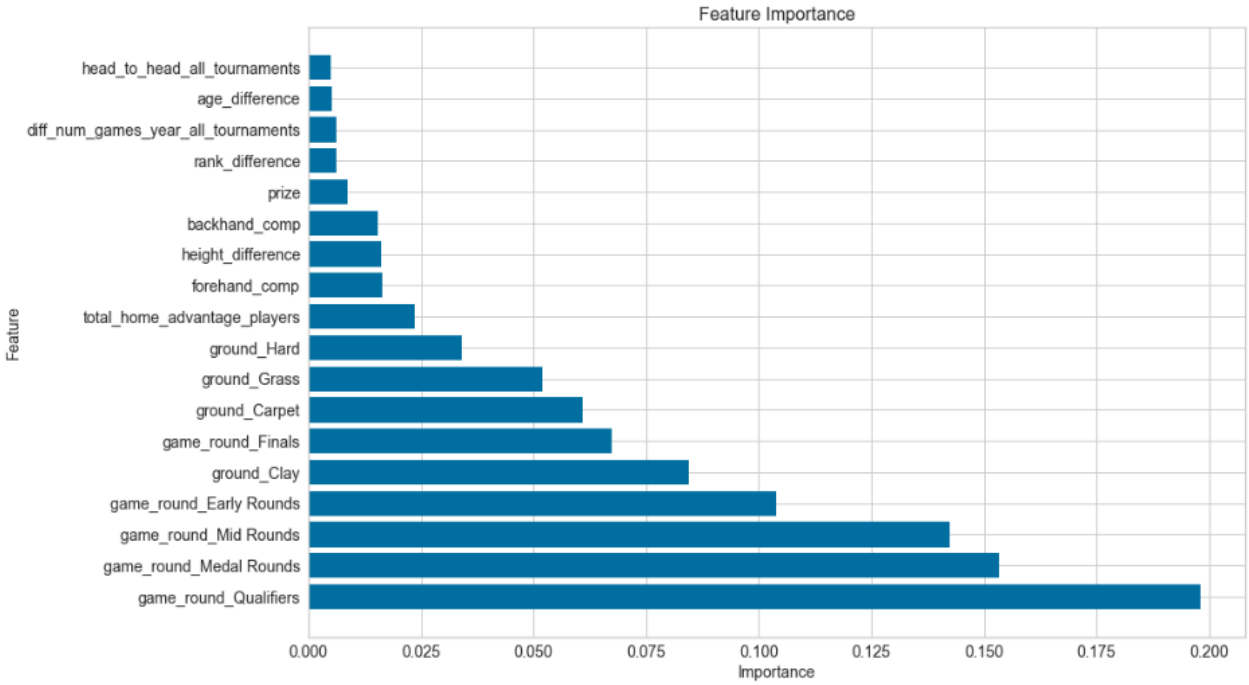


Figura 32 - Feature importance do XGBoost com oversampling

DEPLOYMENT

Com base no objetivo de negócio de desenvolver um modelo preditivo para prever o número de sets em partidas de torneios realizados nos Estados Unidos, a implementação dos modelos pode fornecer benefícios significativos para os treinadores, organizadores de torneios e transmissões televisivas. Ainda que o deployment esteja fora do scope deste projeto, e que os modelos não tenham atingido as capacidades de previsão desejadas, definiram-se duas possíveis linhas de implementação dos modelos desenvolvidos.

Implementação 1: Plataforma de Gestão de Torneios

Uma aplicação prática dos modelos desenvolvidos seria uma plataforma online destinada a facilitar a gestão de torneios de ténis nos Estados Unidos. Essa plataforma poderia oferecer:

- Previsão de Duração dos Jogos: Utilização dos modelos preditivos para prever o número de sets em cada partida, auxiliando os organizadores na programação dos jogos e evitando atrasos e sobreposições nos horários.
- Otimização de Recursos: Capacidade de planear a distribuição eficiente dos campos de jogo e do pessoal necessário com base nas previsões de duração dos jogos, garantindo um fluxo suave do torneio.
- Gestão de Transmissões Televisivas: Fornecimento de previsões precisas sobre a duração dos jogos para os organizadores das transmissões televisivas, permitindo uma agenda mais precisa e uma gestão eficiente dos intervalos comerciais.
- Análise do Desempenho Desportivo: Integração de informações detalhadas sobre o desempenho dos atletas, incluindo estatísticas históricas e fatores que influenciam o número de sets jogados, como diferença de ranking, histórico de confrontos diretos e características físicas dos jogadores.

Essa plataforma seria uma ferramenta útil para os organizadores de torneios, oferecendo insights para o planeamento e a gestão eficiente de todos os aspectos do evento.

Implementação 2: Aplicação de Apoio a Treinadores

Outra aplicação prática dos modelos seria o desenvolvimento de uma aplicação para treinadores de ténis nos Estados Unidos. Essa aplicação poderia oferecer:

- Previsão de Sets em Tempo Real: Utilização dos modelos preditivos para prever o número de sets em partidas em tempo real, permitindo aos treinadores ajustar as suas estratégias durante o jogo.
- Análise de Desempenho Histórico: Fornecimento de informações detalhadas sobre o desempenho histórico dos desportistas, incluindo estatísticas de jogos anteriores e padrões de desempenho que influenciam a duração das partidas.
- Ferramentas de Planeamento de Treino: Capacidade de planejar treinos e estratégias de jogo com base nas previsões de sets e nas análises de desempenho dos jogadores.

- Comparação de Desempenho: Possibilidade de comparar o desempenho do jogador com o adversário em termos de estatísticas relevantes, permitindo uma análise detalhada das áreas a serem melhoradas.

Essa aplicação seria uma ferramenta interessante para os treinadores, oferecendo insights para o planeamento de treinos e estratégias de jogo.

A implementação dos modelos preditivos desenvolvidos neste projeto numa plataforma de gestão de torneios ou numa aplicação de apoio a treinadores tem potencial para trazer benefícios significativos para todos os envolvidos nos torneios ATP dos Estados Unidos. Ao oferecer previsões precisas e análises detalhadas do desempenho dos jogadores, essas ferramentas podem contribuir para a melhoria da gestão, do planeamento e da análise do desporto, contribuindo para o sucesso e a eficiência de torneios futuros.

CONCLUSÃO

Neste trabalho procurou-se prever o número de sets de um jogo de ténis à melhor de três para torneios realizados nos EUA, seguindo o framework metodológico do CRISP-DM. Estas previsões são relevantes para diversas áreas como a publicidade, planeamento tático para o jogo ou organização dos torneios.

Inicialmente, procuramos entender o funcionamento do jogo e perceber as suas regras, de forma a termos um conhecimento base para a realização deste trabalho. Também definimos o nosso objetivo de negócio e analisamos trabalhos semelhantes para entender a "*state of the art*" deste campo. De seguida, foi feita a compreensão dos dados, onde se fez a descrição dos dados presentes no dataset utilizado, a verificação da qualidade destes mesmos dados e a análise exploratória examinando os atributos comparativos entre os jogadores.

Já a preparação dos dados iniciou-se no MongoDB, realizando a gestão e armazenamento de grande volumes de dados facilitando a sua utilização. Estes foram transferidos para MySQL para serem limpos e reorganizados, passando-os para python de modo a realizar o feature engineering, tratar dos outliers, missing values e fazer a seleção dos dados para o modelo.

A seguir, avançamos para a fase de modelação, essencial para alcançar o nosso objetivo de prever o número de sets em jogos de ténis dos EUA. Utilizamos uma variedade de modelos de classificação, escolhidos devido à natureza categórica da variável alvo e à necessidade de prever valores discretos, como dois ou três sets. Durante este processo, desenvolvemos e avaliamos vários modelos, desde a regressão logística até algoritmos mais avançados como XGBoost, MLP Classifier e Random Forest.

A avaliação dos modelos foi realizada utilizando técnicas como divisão de treino e teste, juntamente com oversampling para lidar com o desequilíbrio de classes, e validação cruzada para garantir a generalização dos resultados. No entanto, mesmo com essas estratégias, enfrentámos desafios em equilibrar precisão e recall, especialmente para prever

partidas com três sets, que representam a classe minoritária no nosso conjunto de dados. Pela análise da feature importance do melhor modelo (XGBoost com oversampling) conseguimos perceber que as rondas em que os jogos decorreram têm um impacto grande nas previsões que são feitas, bem como o tipo de piso. Estas características assumem maior importância do que os atributos físicos, propriamente ditos, como a diferença de alturas ou de idades dos jogadores, o que revela a importância da preparação dos atletas em diferentes tipos de campo, bem como mostra a necessidade de uma maior preparação da componente psicológica de cada jogador para conseguir jogar sob pressão e em qualquer ambiente.

Em trabalhos futuros poderia considerar-se fazer web scraping de outras variáveis que podem ser relevantes para a previsão do número de sets nos EUA, como a % de Aces no serviço de cada jogador, a quantidade de duplas faltas no serviço, a % de break points convertidos, entre outras, que estão disponíveis na página do ATP de cada jogador.

BIBLIOGRAFIA

De Seranno, A. (2020). Predicting tennis matches using machine learning (Master's dissertation, Ghent University). Supervisors: L. Martens, & T. de Pessemier. Counsellors: T. de Pessemier, & K. Vanhecke.

International Tennis Federation. (2024). Rules of tennis.

Henry, R. (n.d.). The importance of match play quantity. Voyager Tennis. The importance of match play quantity | Voyager Tennis

Barbosa, I. N. (2016). Perfil das Capacidades Físicas e Performance em Tenistas Portugueses de Alto Rendimento (Dissertação de mestrado). Universidade do Porto, Faculdade de Desporto, Porto, Portugal. Perfil das Capacidades Físicas e Performance em Tenistas Portugueses de Alto Rendimento

Genevois, C., Reid, M., Rogowski, I., & Crespo, M. (2015). Performance factors related to the different tennis backhand groundstrokes: a review. *Journal of sports science & medicine, 14*(1), 194–202.

Koning, R. H. (2011). Home advantage in professional tennis. *Journal of Sports Sciences, 29*(1), 19-27. Home advantage in professional tennis

Gu, W., & Saaty, T. L. (2019). Predicting the Outcome of a Tennis Tournament: Based on Both Data and Judgments. *Journal of Systems Science and Systems Engineering, 28*(2), 317–343. Predicting the Outcome of a Tennis Tournament: Based on Both Data and Judgments | Journal of Systems Science and Systems Engineering

Reid, M., McMurtrie, D., & Crespo, M. (2010). The relationship between match statistics and top 100 ranking in professional men's tennis. *International Journal of Performance Analysis in Sport, 10*(2), 131-138. https://doi.org/10.1080/24748668.2010.11868509

Reid, M., Crespo, M., Lay, B., & Berry, J. (2007). Skill acquisition in tennis: Research and current practice. *Journal of Science and Medicine in Sport, 10*(1), 1-10. Skill acquisition in tennis: Research and current practice - ScienceDirect

Sipko, M. (2015). Machine learning for the prediction of professional tennis matches (Master's dissertation, Imperial College London). Supervisor: W. Knottenbelt.

Mathers, J. F. (2016). Professional tennis on the ATP Tour: A case study of mental skills support. *The Sport Psychologist*. Advance online publication. School of Sport, University of Stirling, Stirling, United Kingdom.

Crim, J. (n.d.). Bye in Tennis. *TennisCompanion*. Bye in Tennis | Definition, Examples, and Common Questions About The Bye

NVIDIA. (n.d.). XGBoost – What is it and why does it matter? XGBoost – What Is It and Why Does It Matter?

Edpuganti, A. (n.d.) Multi-Layer Perceptron (MLP): A Basic Understanding. *OpenGenus IQ*. Multi-Layer Perceptron (MLP): A Basic Understanding

Khan Academy. (n.d.). Identifying outliers with the 1.5xIQR rule. Identifying outliers with the 1.5xIQR rule (article) | Khan Academy