

Projeto Aplicado em Ciência de Dados I

Previsão do nº de sets para conclusão de um jogo de ténis profissional - Estados Unidos

Grupo 10

Gonçalo Mealha
123391

Jingyu Huang
123432

João Jin
123388

José Valério
112255

Tiago Fernandes
123400

Salvador Ferreira
123465

Docente:
Sérgio Moro

Índice

Introdução	4
<i>Business Understanding</i>	5
<i>Data Understanding</i>	7
<i>Data Preparation</i>	9
Recolha de dados em falta.....	9
<i>Update</i> das informações dos jogadores	9
Informações dos oponentes.....	10
Novo campo “LinkOpponent”	10
Inserção das informações dos oponentes numa nova collection “opponents_data”	10
Países dos torneios	11
Campo Opponent	11
Campo Score.....	12
Criação de novos campos na base de dados	12
Campo “TournamentId”	12
Campo “MatchId”	13
Criação de <i>Collections</i>	14
Collection de jogadores (<i>Player</i>).....	14
Collection de torneios (<i>Tournament</i>).....	14
Collection de jogos (<i>Match</i>)	15
Collection Final	15
Campo “n_set”	16
Criação de novos campos para coleção final	17
Winner_rank e loser_rank.....	17
Game_score	17
Number_of_sets.....	18
Exportação para CSV	18
Análise Exploratória de Dados.....	19
Criação de novas variáveis	27
Análise univariada com a variável alvo	32
Modelação	34

Eliminação dos NA's	34
Correlações e associações	34
Associações entre as variáveis numéricas e a variável alvo	34
Associações entre as variáveis nominais e a variável alvo	36
Correlações entre os preditores numéricos	37
Correlação entre os preditores numéricos e categóricos.....	38
Features a usar nos modelos	39
Construção da amostra treino-teste.....	40
Modelos.....	41
Regressão Logística	42
KNN	43
Árvore de Decisão	45
<i>Random Forest</i>	46
Avaliação Final	48
Implementação	49
Conclusão	50

Introdução

Neste trabalho final no âmbito da Unidade Curricular de Projeto Aplicado a Ciência de Dados I, foi-nos concedida uma base de dados com informações sobre ténis, onde o objetivo é desenvolver um modelo preditivo capaz de estimar o número de *sets* necessários para concluir uma partida, com base em diferentes variáveis relacionadas aos jogadores, torneios e características do jogo. No nosso caso, foi-nos atribuído um país em específico para trabalharmos: os Estados Unidos.

Para este projeto, foi seguida a metodologia CRISP-DM (Cross Industry Standard Process for Data Mining), conforme ilustrada na figura abaixo. Esta metodologia organiza o processo de Ciência de Dados em seis etapas principais, estruturadas de forma iterativa.

A primeira etapa, *Business Understanding*, é essencial para definir claramente os objetivos do projeto do ponto de vista do negócio e para orientar todas as decisões técnicas subsequentes.

Em seguida, a fase de *Data Understanding* permite explorar, descrever e avaliar os dados disponíveis, identificando possíveis problemas de qualidade ou limitações. Já na etapa de *Data Preparation*, procede-se à limpeza, transformação e seleção das variáveis relevantes — sendo esta uma fase crítica, pois determina a base com que os modelos serão treinados.

A etapa de *Modeling* envolve a escolha e a aplicação de algoritmos de *machine learning* adequados ao problema em questão. No entanto, a eficácia de um modelo não é garantida apenas pela sua construção: na fase de *Evaluation*, os modelos são avaliados rigorosamente com base em métricas de desempenho, assegurando que respondem aos objetivos definidos inicialmente.

Por fim, na fase de *Deployment*, os resultados obtidos são integrados no processo de tomada de decisão, seja através da entrega de um modelo funcional ou de recomendações analíticas.

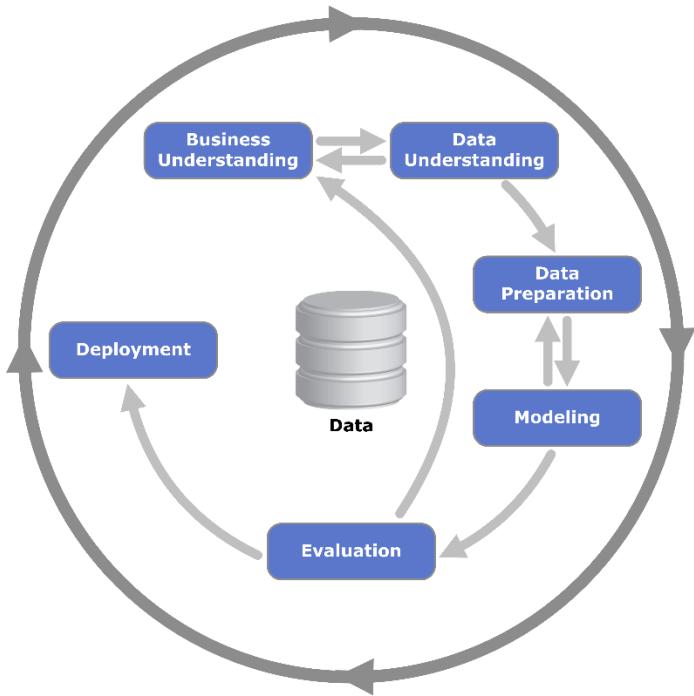


Fig. 1 – CRISP-DM

Business Understanding

O ténis é um dos desportos mais populares a nível mundial, com uma forte presença nos Estados Unidos, onde são realizados diversos torneios de grande prestígio.

Regras do Ténis Profissional

O ténis é um desporto disputado por dois (1vs1) ou quatro jogadores (2vs2), embora neste projeto apenas se considere os jogos entre dois jogadores. De forma muito simples, o seu objetivo é bater a bola com a raquete para o campo contrário, de modo que o adversário não consiga devolvê-la com sucesso, acumulando pontos até à vitória. O ténis profissional segue um formato bem definido, no qual os jogos podem ser decididos em diferentes números de sets, dependendo das regras do torneio. No circuito ATP, os jogos são normalmente disputados em formato melhor de três sets (Bo3) ou melhor de cinco sets (Bo5), sendo que este último ocorre principalmente nos torneios do *Grand Slam*.

O primeiro jogador a ganhar seis jogos vence o set, desde que tenha uma vantagem mínima de dois jogos sobre o adversário. Se for necessário, o set continuará até que se alcance essa diferença. Em caso de empate de seis jogos, joga-se um *tie-break*. Para vencer o *tie-break* (e consequentemente o set), um jogador precisa de marcar 7 pontos com uma vantagem mínima de 2 pontos sobre o adversário.

Ténis nos EUA

Os Estados Unidos têm uma história rica no ténis, sendo um dos países com maior tradição na formação de jogadores de topo e na realização de torneios de elite. Os torneios realizados em território americano, além de atraírem os melhores jogadores do mundo, também geram um impacto económico significativo, movimentando milhões de dólares em prémios, patrocínios e bilheteiras.

Além disso, a infraestrutura desportiva do país permite que os torneios ocorram em diferentes tipos de superfície, com predominância de campos rápidos (*hard court*), que influenciam o ritmo dos jogos e, consequentemente, o número de *sets* disputados.

Como jogadores, destacam-se *Pete Sampras* com 14 *Grand Slam's* conquistados e *Jimmy Connors* que foi considerado o melhor tenista do mundo em 5 anos consecutivos.

Impacto Económico e Comercial

A previsão do número de sets num jogo de ténis tem diversas aplicações económicas e comerciais.

No setor das apostas desportivas, um modelo preditivo pode ser uma ferramenta essencial para apostadores e casas de apostas na definição de probabilidades e estratégias de risco, levando a decisões mais fundamentadas.

Além disso, a duração dos jogos influencia diretamente a transmissão televisiva e a publicidade, visto que encontros mais longos podem alterar a programação e gerar maior envolvimento do público, resultando em receitas publicitárias mais elevadas. No planeamento de torneios, compreender a probabilidade de um jogo durar mais ou menos *sets* auxilia na gestão de horários, garantindo uma organização eficiente e evitando atrasos em partidas subsequentes.

Outra vertente importante é o impacto no consumo dentro dos eventos, pois jogos mais longos tendem a aumentar a venda de alimentos, bebidas e *merchandising*, maximizando o retorno financeiro para os organizadores e patrocinadores. Ao saber da duração das partidas, as empresas poderiam decidir em que eventos é que iriam investir mais recursos.

Assim, prever o número de *sets* torna-se uma ferramenta valiosa para os diferentes intervenientes do ténis profissional.

Objetivos

Com este projeto pretendemos explorar os diferentes modelos de aprendizagem supervisionada onde o principal objetivo é desenvolver um modelo capaz de prever o número de *sets* em jogos da ATP Tour realizados nos Estados Unidos. Para isso, serão analisadas diversas variáveis, incluindo características dos jogadores, jogos e torneios. A previsão do número de *sets* permitirá melhorar a compreensão do jogo e otimizar a tomada de decisões em múltiplos contextos.

Data Understanding

A fase de *Data Understanding* constitui uma etapa analítica fundamental que requer a compreensão abrangente dos dados disponíveis e do seu contexto desportivo. Esta etapa permite não apenas identificar a estrutura dos dados, mas também avaliar a sua relevância nos objetivos do estudo.

Para o presente trabalho, utilizou-se uma base de dados recolhida através do portal oficial da *Association of Tennis Professionals (ATP)*, que reúne informações detalhadas sobre jogadores, torneios, resultados, *rankings* e estatísticas desde 1973. O *dataset* inicial é composto por 1308835 instâncias de 15 variáveis diferentes organizadas em três dimensões principais: jogadores, torneios e jogos. Segue-se abaixo uma descrição dos atributos que as compõem.

Nome da variável	Descrição da variável
_id	Identificador de cada registo
PlayerName	Nome do jogador
Born	País e cidade de nascimento do jogador
Height	Altura do jogador
Hand	Golpe usado pelo jogador
LinkPlayer	Link identificador de cada jogador
Tournament	Nome do torneio
Location	Localização do torneio
Date	Data do torneio
Ground	Pavimento do jogo
Prize	Prémio do torneio
GameRound	Etapas do torneio
GameRank	Posição de oponente no ranking
Oponent	Nome do jogador oponente
WL	Jogo ganho (W) ou perdido (L)
Score	Pontuação do jogo

Tab 1 - Variáveis da base de dados e a sua respetiva descrição

Registos duplicados

Um dos problemas mais frequentes quando enfrentamos uma base de dados de tamanho considerável é a existência de valores duplicados, isto é, quando todos os elementos de uma determinada linha são exatamente iguais a todos os elementos de outra linha. Este problema compromete a qualidade e a integridade dos dados, e dificulta as análises subsequentes e, por consequência, as conclusões finais.

Logo, uma das primeiras tarefas foi verificar a existência de registos duplicados na base de dados. Foram identificados 2830 registos duplicados que foram prontamente removidos.

Jogos espelhados

Durante a análise de dados, o grupo identificou um problema específico que está relacionado com a duplicidade de informações referentes ao mesmo jogo. Ou seja, na base de dados existem casos em que o mesmo jogo é registado duas vezes, mas com perspetivas diferentes:

- Num registo, o “*PlayerName*” identificava o jogador A, e o jogador B no “*Opponent*”;
- Num outro registo, esses papéis eram invertidos: no “*PlayerName*” aparecia o jogador B e no “*Opponent*” o jogador A.

No entanto, os outros campos relacionados às informações do torneio e jogo são compartilhados nesses dois registo. Ora, tal redundância de informações é um desafio que precisa de ser resolvido durante a fase da preparação dos dados.

Informações em falta

Apesar de não haver *missing values* propriamente ditos, até por se tratar de uma base de dados JSON, deparámo-nos com uma descoberta bastante importante: a existência de nomes de jogadores “*Oponent*” que não existiam no atributo “*PlayerName*”.

A ausência destes nomes no campo “*PlayerName*” implica que não hajam registo com as informações/características sobre estes jogadores. Este é um dos grandes problemas desta base de dados, que mais tarde poderia distorcer os resultados do modelo. Portanto, tornou-se claro que seria necessário encontrar uma forma de obter os dados destes jogadores em falta. Após uma investigação mais profunda, verificámos que existiam 12731 jogadores em “*Oponent*” que não existiam no atributo “*PlayerName*”.

Caractere especial

Verificámos também que existem valores com um caractere especial (♦) em várias variáveis, o que indica possíveis problemas de codificação, provavelmente causados por ficheiros guardados com formatos que não suportam corretamente caracteres *Unicode*.

A tabela abaixo apresenta as variáveis que possuem esses caracteres especiais e respetivo número de ocorrência:

PlayerName	Born	Location	Prize	Oponent
2	36	1	234	10

Tab 2 - Número de caracteres especiais (♦) por variável

Data Preparation

Recolha de dados em falta

Update das informações dos jogadores

Para as informações dos jogadores em falta, e uma vez que seriam fundamentais para a construção dos modelos, foi desenvolvido um *script* em *Python* para recolher os dados de cada um dos jogadores diretamente do *site* da *ATP Tour*. Foi utilizada a biblioteca *Selenium* que permite realizar casos de teste automatizados em *browsers*.

Ao usar essa ferramenta, conseguimos simular a abertura de cada um dos *links* dos jogadores e obter os dados correspondentes referenciando os atributos e identificadores presentes no código fonte da página *html*. Assim, o grupo decidiu extrair: a altura (“Height”), a localização/país onde o jogador nasceu (“Born”), a sua data de nascimento (“Birth Date” - novo campo), o código desse país (“BornCountryCode” - novo campo), os tipos de golpes usados pelos jogadores (“Hand”) e o rank ATP na altura do jogo (“SglRollRank” - novo campo).

À medida que os dados iam sendo recolhidos, o *script* gerava automaticamente os comandos *MongoDB* necessários para atualizar a base de dados, armazenando-os num ficheiro que, no final do processo, continha todos os comandos de atualização correspondentes aos jogadores identificados como *PlayerName*.

Informações dos oponentes

Novo campo “LinkOpponent”

Como identificado na fase de *Data Understanding*, as informações dos “*Opponent*” que nunca apareciam como “*PlayerName*” não constavam na base de dados. A estratégia para recolher essas informações seria a mesma que foi usada para os “*PlayerName*”, só que para isso seriam necessários os *links* dos oponentes, informação que não constava na base de dados inicial.

Então, foi criado um campo “*LinkOpponent*” com o objetivo de armazenar o *link* dos jogadores do campo “*Opponent*”, não só para extrair as suas informações como mais tarde servir como um identificador único.

Com recurso ao *Python* e às suas bibliotecas, foi possível desenvolver um *script* que simulava a abertura do browser *Firefox*, e no site [duckduckgo](https://duckduckgo.com) realizava uma pesquisa “site:www.atptour.com <nome do jogador>” que depois, através de expressões regulares, extraía o primeiro *link* da página de resultados. Ao incluir o operador “site:www.atptour.com” e o nome do jogador, restringe-se a pesquisa apenas ao domínio oficial da ATP e aumenta-se a probabilidade de que o primeiro resultado seja a página oficial do perfil do jogador nesse site.

Inserção das informações dos oponentes numa nova collection “opponents_data”

A partir dos *links* obtidos pelo *script*, foi possível extrair informações como Nome, Data de nascimento, Altura, País de nascimento, Código do país e os golpes usados pelo jogador (“*Hand*”) tal como foi feito para os “*PlayerName*”. Esses dados foram inicialmente armazenados numa coleção “*opponents_data*”.

Nota: De referir que esta coleção continha os dados de todos os jogadores que apareciam como “*Opponent*”, independentemente de já existirem as suas informações quando registados como “*PlayerName*”. Posteriormente, foram encapsulados numa nova coleção chamada *Player*, onde foi adotada uma estratégia para que não fossem armazenadas informações repetidas. Esta coleção irá reunir todos os jogadores distintos presentes na base de dados (sejam eles identificados como “*PlayerName*” ou “*Opponent*”).

Países dos torneios

Visto que era fundamental saber em que países foram realizados os torneios, de forma a podermos a extrair a nossa amostra correspondente aos jogos nos Estados Unidos, decidimos criar um campo para guardar os códigos dos países referentes aos torneios, à semelhança do que foi feito para o código dos países de nascimento do jogador. Foi atribuído o nome “*TournamentCountryCode*” a este campo.

Para preencher este atributo foi criado um *script* em *Python* que teve como objetivo identificar e associar os códigos de países às localizações de torneios presentes na base de dados. Para isso, foram utilizados dois arquivos CSV: um contendo os nomes e localizações dos torneios e outro com os nomes dos países e os seus respetivos códigos.

Desenvolveu-se um algoritmo que verificou se o nome de um país estava presente na localização de um torneio, realizando a comparação de forma insensível a maiúsculas e minúsculas.

Sempre que uma correspondência era encontrada, o código do país era então usado para ser gerado o comando *MongoDB* necessário para atualizar a base de dados.

Campo Opponent

Outro problema que surgiu foi o aparecimento do valor “*bye*” como sendo um oponente. Reparámos ainda que na maioria das vezes em que o “*Opponent*” era “*bye*”, o resultado do jogo (“*WL*”) encontrava-se vazio. Porém, encontrámos 3 exceções de jogos sem “*Opponent*” (“*Opponent*” = “*bye*”) em que o resultado do jogo (“*WL*”) foi “*W*”.

Em torneios de ténis, “*bye*” significa que o jogador avança automaticamente para a próxima fase sem precisar de jogar. Estas situações acontecem maioritariamente nas fases iniciais dos torneios quando não existem jogadores suficientes para fazer o emparelhamento dos jogos. Geralmente, este estatuto de passar à próxima fase é atribuído aos jogadores melhor posicionados no *ranking*, funcionando como uma espécie de recompensa pelos seus desempenhos nas fases anteriores. Assim, fez sentido para o grupo não considerar estes jogos na criação da coleção Match, pois o “jogo” não chegou a acontecer e não faria sentido fazer uma previsão sobre o mesmo.

Campo Score

Devido a algumas inconsistências no formato da variável “Score”, foi necessário proceder à limpeza e padronização dessa mesma variável. Identificaram-se três tipos de problemas que exigiam intervenção para garantir a qualidade e uniformidade dos dados.

O primeiro problema dizia respeito a 14 registos com valores em falta (representados por *strings* vazias, “ ”), ou seja, partidas cujo resultado não estava disponível no *dataset* original. Para tentar preencher essas lacunas, foi realizada uma pesquisa manual em *sites* oficiais de ténis, com base nas informações disponíveis acerca dos jogos. No entanto, apesar dos esforços, não foi possível recuperar os dados em questão, que permaneceram como *missings* na base de dados.

O segundo problema envolvia *scores* registados com vírgulas seguidas de espaço, uma inconsistência de formatação que dificultava o seu processamento. Nestes casos, os caracteres foram removidos para tornar o valor uniforme. Por exemplo, um *score* originalmente registado como “04 14 54, 42 54” foi transformado em “04 14 54 42 54”.

Num terceiro momento, analisaram-se 16 casos em que os valores dos *scores* apareciam unidos por hífen (“-”), situação típica quando pelo menos um dos jogadores atinge 10 ou mais pontos num *set* (*tie-break*). Estes casos exigiram uma correção manual uma vez que apresentavam valores indevidamente colados, como exemplificado por “46 15-138-1062 62”. Para estes casos, realizou-se uma verificação manual no *site* oficial da ATP, obtendo-se os resultados corretos - no exemplo citado, a versão corrigida passou a “46 15-13 8-10 62 62”.

Após a limpeza, todos os valores da variável “Score” foram convertidos para o tipo *string*, assegurando a consistência no seu formato e facilitando etapas futuras de processamento e análise.

Criação de novos campos na base de dados

Campo “TournamentId”

A fim de conseguirmos distinguir os torneios diferentes e, ao mesmo tempo, facilitar a associação entre os torneios e jogos posteriormente na criação de *collections*, o grupo decidiu adicionar um novo campo na base de dados denominado “*TournamentId*”.

Este campo foi preenchido com *hash's* gerados através um *script Python*, com o objetivo de garantir que cada torneio tenha um ID único e exclusivo.

A decisão de criar “*TournamentId*” surgiu durante a análise da base de dados, onde verificámos que existiam torneios que apresentavam o mesmo nome e data de ocorrência, mas que tinham ocorrido em localizações diferentes.

A falta de um identificador único para diferenciar esses casos específicos dificultava a análise dos dados. Além disso, a similaridade entre estes torneios poderia aumentar o risco de confusão durante o processo de criação da collection “*Match*”, especificamente na associação de jogos e torneios.

Assim, para resolver estes problemas, introduzimos o “*TournamentId*”, o que permitiu que cada torneio fosse diferenciado de forma inequívoca.

Um exemplo de um ID para um torneio:

TournamentId: `e4585c63a8a1b6bd50adf42dd82343a7`

Campo “*MatchId*”

A criação do campo “*MatchId*” justificou-se principalmente pela necessidade de construir e estruturar a collection “*Match*” (desenvolvida posteriormente) de forma mais eficiente e consistente. Durante a compreensão dos dados, o grupo identificou um problema específico que estava relacionado com a duplicidade de informações referentes ao mesmo jogo (jogos espelhados).

Assim, para abordar esta questão, foi criado o “*MatchId*”. Este campo vai ser um identificador único para cada jogo, composto pelo ID do Torneio onde esse jogo foi realizado (hash criado anteriormente), a fase da competição (“*GameRound*”) e os *ID’s* dos jogadores envolvidos nesse jogo, ficando com um aspeto final como o do seguinte exemplo:

MatchId: `000cea2d9229b5c919fca319557a9c7a_Quarter-Finals_a385_b837`

É importante destacar que os *ID’s* dos jogadores envolvidos foram ordenados alfabeticamente, independentemente de quem surgir como “*PlayerName*” ou “*Opponent*”. Dessa forma, mesmo que os registos estejam invertidos (*Player A vs Player B* e *Player B vs Player A*), terão o mesmo “*MatchId*”, o que tornará mais fácil a criação da coleção dos jogos e distinção entre os mesmos.

Criação de *Collections*

Collection de jogadores (Player)

Esta *collection* tem como principal objetivo guardar todos os jogadores da base de dados presentes nos atributos “*PlayerName*” e “*Opponent*”, sendo que as informações referentes aos oponentes ficaram armazenadas numa *collection* auxiliar denominada “*opponents_data*”, conforme referido anteriormente.

No MongoDB, ao criar uma coleção, cada linha de dados é atribuída um “*_id*” que atua como um identificador único desse registo. No entanto, durante a análise dos dados, o grupo observou que o campo “*LinkPlayer*” poderia ser utilizado para esse fim. Os “*LinkPlayer*” são *links* que direcionam diretamente para a página dos jogadores no site ATP.

Notámos que uma parte desses *links* é constituída por 4 caracteres aleatórios (combinação de letras e números) e que são únicos. Por isso, para aproveitar esta vantagem, foi feito um comando *split()* no campo “*LinkPlayer*”, de forma que apenas esses 4 caracteres sejam utilizados como “*_id*” dos jogadores, garantindo a unicidade dos dados. Além disso, o campo “*Hand*” foi separado em dois, ficando “*forehand*” para o golpe predileto do jogador e “*backhand*” para o movimento secundário.

Para tal efeito, foram extraídos todos os campos que dizem respeito às características dos jogadores, tendo em conta as alterações efetuadas previamente. Os campos utilizados foram: “*Name*”, “*Born*”, “*BornCountryCode*”, “*BirthDate*”, “*Height*”, “*Forehand*”, “*Backhand*”, “*LinkPlayer*”.

Para a criação desta coleção, foram desenvolvidos três comandos *aggregate()*: Um para extrair as informações dos jogadores na *collection* “*atpplayers*” e outro para extrair as informações dos jogadores na *collection* “*opponents_data*”. Foi ainda necessário um último *aggregate* para remover os registos duplicados, isto é, os jogadores que estavam em ambas a coleções e que, por conseguinte, as suas informações iriam aparecer 2 vezes.

Collection de torneios (Tournament)

A criação da coleção “*Tournament*” foi feita de forma análoga ao processo adotado para a coleção “*Players*”, com foco na identificação dos atributos associados a cada torneio.

Começámos por identificar os atributos da base de dados que caracterizam um torneio, como “*TournamentId*”, “*Tournament*”, “*Date*”, “*Location*”, “*TournamentCountryCode*” e

“Prize”. O uso do “*TournamentId*”, criado previamente na base de dados como um identificador único, foi uma vantagem significativa nesta etapa, uma vez que garantiu que as informações referentes aos torneios fossem agrupadas de uma forma precisa e sem haver duplicados, assegurando dessa forma a integridade dos dados.

Assim, os campos a serem projetados foram: *id* (corresponde ao *TournamentId*), *name* (nome do torneio), *location* (local onde o torneio ocorreu), *location_country_code* (código do país onde o torneio ocorreu), *start_date* (data de início), *end_date* (data de fim) e *prize* (prémio associado ao torneio).

Collection de jogos (Match)

Para a criação da coleção de jogos, decidimos não considerar os jogos em que o adversário foi marcado como “*bye*”, ou seja, jogos em que um jogador não teve um oponente real.

Decidimos guardar o vencedor e o perdedor de cada jogo através da criação dos campos: *winner_id*, *winner_name* e *loser_id*, *loser_name*. Esses campos eram posteriormente preenchidos com base no campo “*WL*”, onde “*W*” significa que o “*PlayerName*” foi o vencedor (*winner_name*) e o “*Opponent*” o perdedor (*loser_name*). Quando “*WL*” é “*L*”, a situação é inversa: o vencedor é o “*Opponent*” (*winner_name*) e o perdedor o “*PlayerName*” (*loser_name*). Os ID's de ambos os jogadores foram extraídos a partir do seu respetivo *link*, num processo semelhante ao realizado na *collection Player*.

Assim, os campos a serem projetados foram: *MatchId* (*id* do jogo referido anteriormente), *TournamentId* (referência ao torneio), *GameRound* (fase do torneio), *Ground* (tipo de superfície), além dos campos *winner_id* (ID do vencedor), *loser_id* (ID do perdedor), *winner_name* (Nome do vencedor), *loser_name* (Nome do perdedor) e *Score* (Resultado do vencedor do jogo).

Para eliminar registos duplicados, é utilizado o *\$group*, que agrupa os jogos com base no “*MatchId*”. Dessa forma, para os jogos que estão registados de duas perspetivas (*Player A vs Player B* e *Player B vs Player A*), apenas um desses registos será mantido.

Collection Final

Após a criação destas três coleções, o grupo decidiu exportá-las para ficheiros CSV de forma a, posteriormente, importar esses ficheiros no SQL. Esta abordagem permitiu obter a tabela final com os torneios realizados apenas nos Estados Unidos (*Tournament*

CountryCode : "US"), com todas as variáveis necessárias para as análises subsequentes e para a construção dos modelos.

Como resultado da integração no SQL, obtivemos uma tabela que inclui as variáveis com informação sobre ambos os jogadores (id, nome, altura, país, tipo de *forehand* e data de nascimento), bem como variáveis relativas ao jogo e torneio (*match_id*, nome do torneio, datas de início e fim de torneio, tipo de pavimento, fase do torneio e prémio monetário).

No entanto, após uma reavaliação do processo e uma reflexão mais aprofundada sobre o problema, decidimos regressar ao *MongoDB* com o objetivo de adicionar novas variáveis, tais como "*winner_rank*", "*loser_rank*", "*game_score*" e "*number_of_sets*".

Antes da criação destas novas variáveis, foi filtrada a base de dados para incluir apenas os torneios realizados nos Estados Unidos chamada *atpplayers* no *MongoDB*. Dado que esta base de dados, contrariamente à coleção dos jogos, ainda apresentava o problema das partidas espelhadas (duplicados com jogadores invertidos). Para resolver esta questão, foi extraída a partir da coleção *atpplayers* uma nova coleção denominada *atpplayers1*, com base no campo "*MatchId*", de forma a manter apenas um registo por jogo (num total de 79807 jogos).

Dessa forma, as variáveis "*winner_rank*", "*loser_rank*", "*game_score*" e "*number_of_sets*" que se pretende adicionar à coleção *final* foram obtidas a partir da coleção *atpplayers1*. Através da correspondência pelo campo "*MatchId*", estas variáveis foram integradas na coleção *final*, permitindo assim complementar as informações já existentes com dados adicionais relevantes.

Campo "*n_set*"

Tendo em conta que o principal objetivo deste trabalho é desenvolver um modelo capaz de prever o número de *sets* num jogo de ténis, tornou-se necessária a inclusão de uma variável que indicasse a quantidade de *sets* disputados em cada jogo. Para tal, recorreu-se à variável "*Score*", que contém os resultados parciais de cada partida, extraíndo dessa forma a informação necessária para a criação da variável "*n_set*". Esta variável reflete o número de *sets* efetivamente disputados, sendo obtido através da contagem das sequências de resultados. Por exemplo, "64 36 36" corresponde a três *sets*.

Contudo, é importante destacar que nem todos os valores presentes na variável "*Score*" apresentam um formato convencional. Existem casos em que os resultados não contêm esse par de números, sendo classificados como casos irregulares. Estes casos refletem exceções ou interrupções nos jogos, identificadas pelas seguintes anotações:

- (**(W/O) (walkover)**): atribuição da vitória ao jogador quando o oponente não pôde comparecer, portanto, nestes casos não houve jogo;

- **(RET) (retired)**: indica que um dos jogadores desistiu, geralmente por motivos de lesão ou doença;
- **(DEF) (defaulted)**: indica que o jogo foi interrompido devido a má conduta ou violação de regras de jogo.

Nos casos em que estas anotações aparecem isoladas no campo “Score”, o número de *sets* atribuído será 0, uma vez que o jogo não foi disputado. No entanto, quando as anotações *(RET)* e *(DEF)* aparecem juntamente com pares de números, como em "67 63 34 (RET)", o número de *sets* será calculado com base nos pares de números. Neste exemplo, apesar de o último *set* não ter sido concluído por motivos excepcionais aos 3-4, serão considerados os 3 *sets*, uma vez que o *set* em questão já tinha sido começado antes da desistência. Mais tarde, em *Python*, estes casos serão analisados com mais atenção.

Criação de novos campos para coleção final

Winner_rank e *loser_rank*

Para a criação destas variáveis, foi utilizada uma função *aggregate* com a operação *\$lookup*, que permitiu a junção entre a coleção *final* e a coleção *atpplayers1*, com base no campo “*match_id*”.

A correta atribuição dos *ranks* aos jogadores exigiu uma atenção especial no campo “*WL*”, o qual indica se o “*PlayerName*” ganhou ou perdeu o jogo:

- ***winner_rank***: se o valor de “*WL*” for “*W*”, significa que o jogador principal (“*PlayerName*”) venceu e, portanto, o seu *rank* (“*SglRollRank*”) é atribuído a *winner_rank*. Caso contrário, se “*WL*” for “*L*”, então o *winner_rank* corresponde ao *rank* do oponente (“*GameRank*”).
- ***loser_rank***: aplica-se a lógica inversa. Se “*WL*” for “*W*”, ou seja, o “*PlayerName*” venceu, então *loser_rank* é “*GameRank*” do oponente. Caso “*WL*” seja “*L*”, então *loser_rank* será o “*SglRollRank*” do próprio jogador principal.

Game_score

A variável “*game_score*” foi criada, novamente, com recurso à operação *aggregate* e *\$lookup* entre as coleções *final* e *atpplayers1*, com base no campo “*match_id*”. Sempre que o “*match_id*” da coleção *final* coincide com “*MatchId*” da coleção *atpplayers1*, o valor correspondente do campo “*Score*” da *atpplayers1* é inserido na variável “*game_score*” da coleção *final*.

Number_of_sets

De forma análoga à criação da variável “*game_score*”, a variável “*number_of_sets*” foi gerada utilizando a operação *aggregate* com *\$lookup* entre as coleções *final* e *atpplayers1*, com base no campo “*match_id*”. Sempre que o “*match_id*” da coleção *final* coincidisse com o “*MatchId*” da coleção *atpplayers1*, o número de *sets* era inserido na variável “*number_of_sets*” da coleção *final*.

No final das alterações, a coleção *final* passou a ter seguinte estrutura:

```
_id: ObjectId('67fa8472ef021112a2b43aed')
winner_id : "a092"
winner_name : "Andre Agassi"
winner_height : 180
winner_country : "United States"
winner_forehand : "Right-Handed"
winner_birth_date : "1970/04/29"
loser_id : "p024"
loser_name : "Mikael Pernfors"
loser_height : 173
loser_country : "Sweden"
loser_forehand : "Right-Handed"
loser_birth_date : "1963/07/16"
match_id : "004955061247a50bbaf1c9be289047f2_Finals_a092_p024"
tournament_name : "Memphis"
tournament_start_date : "15/2/1988"
tournament_end_date : "21/2/1988"
match_ground : "Hard"
tournament_prize : "$297,500"
match_round : "Finals"
winner_rank : 18
loser_rank : 39
game_score : "64 64 75"
number_of_sets : 3
```

Fig. 2 – Estrutura final da base de dados em MongoDB

Exportação para CSV

Concluído o processo de tratamento e enriquecimento dos dados em *MongoDB*, procedeu-se à exportação final da coleção consolidada. Esta exportação resultou na criação de um ficheiro CSV intitulado “*atp1.final.csv*”, que passou a representar a versão final e estruturada da base de dados, pronta para ser utilizada nas etapas subsequentes de análise exploratória e modelação preditiva em *Python*.

Análise Exploratória de Dados

Após o tratamento dos dados realizado em *MongoDB*, deu-se início à fase de preparação e exploração do *dataset* em *Python*, como etapa prévia à construção dos modelos. Nesta fase, foi feita uma análise mais aprofundada das variáveis a serem utilizadas, permitindo aplicar os ajustes necessários para garantir a sua adequação no processo de modelação.

Dados omissos

Para começar, decidiu-se estudar a qualidade da base de dados através da construção de um gráfico que mostrasse a proporção de valores omissos em cada variável.

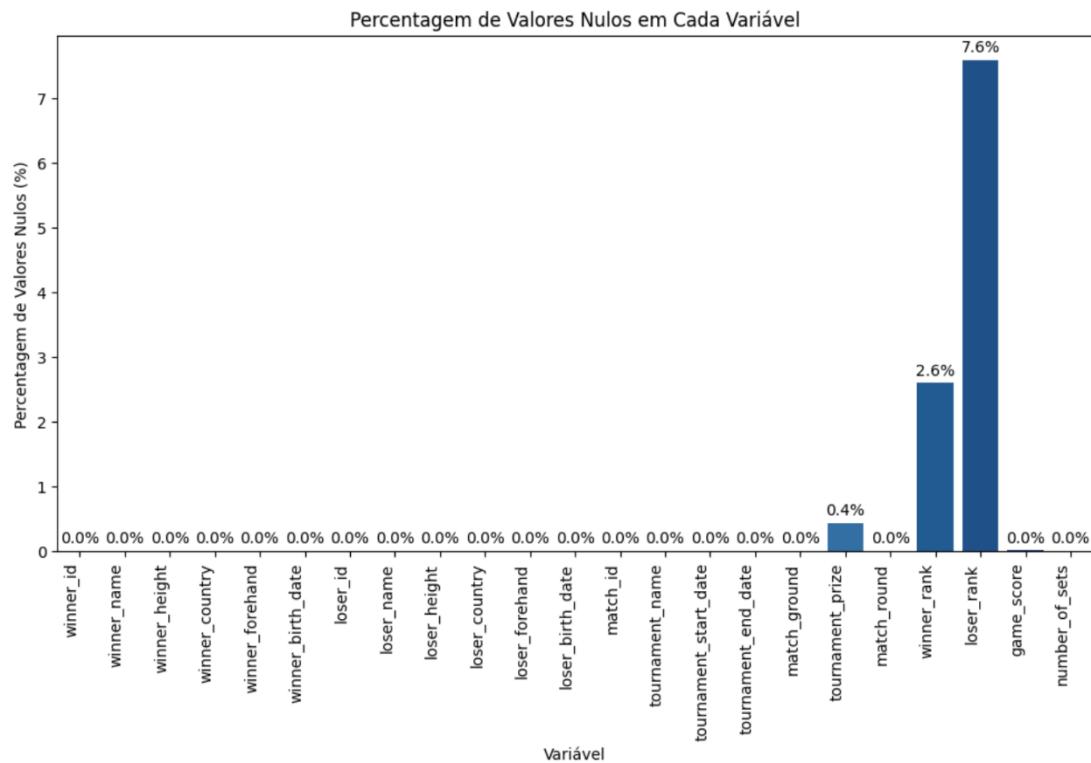


Fig. 3 – Percentagem de valores nulos por variável

Como é possível observar, as variáveis referentes ao *ranking* do jogador são as mais afetadas pelos valores omissos. Apesar do nosso esforço em recolher essas informações, o facto das mesmas não constarem no *site ATP*, dificultou a nossa tarefa. Nas restantes variáveis, a ocorrência de valores em falta é bastante reduzida.

Tournament

Nos Estados Unidos estão registados 415 torneios diferentes, incluindo jogos da *ATP Tour*, *ATP Challenger*, *ITF (International Tennis Federation)* e ainda compromissos entre seleções, isto é, jogos onde os jogadores representam o seu país de origem. No entanto, o trabalho foca-se nas partidas que contam para o *ranking ATP* e como os jogos entre seleções não são contabilizados para os pontos do *ranking*, esses registo foram removidos. Além disso, estas partidas não apresentam necessariamente o mesmo nível de competitividade dos torneios oficiais, sendo muitas vezes de carácter amigável, o que pode influenciar negativamente a motivação dos jogadores. Tal discrepância poderia enviesar a análise e comprometer a consistência dos modelos preditivos. Após esta filtragem, permaneceram 344 torneios diferentes na base de dados.

O *US Open*, um dos torneios mais antigos do mundo, destaca-se como sendo o com mais partidas nos Estados Unidos, seguido pelos *ATP Masters 1000 de Miami* e de *Cincinnati*.

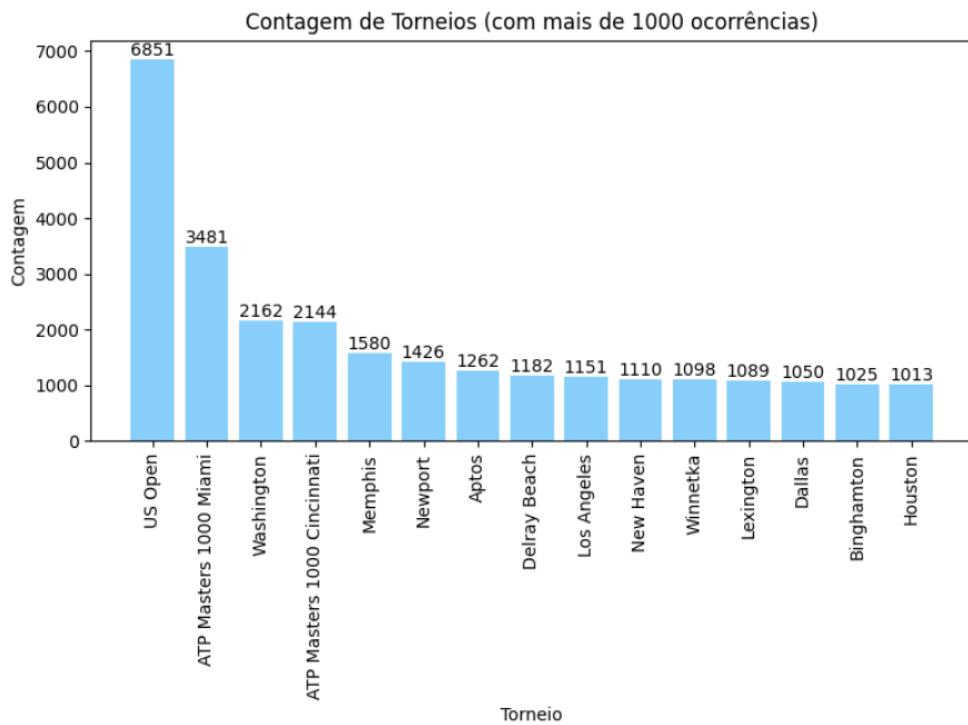


Fig. 4 – Torneios com mais jogos

Number of Sets

A variável alvo representa quantos sets foram disputados em cada partida. Inicialmente, os dados incluíam jogos com 0, 1, 2, 3, 4 e 5 sets. No entanto, partidas com 0 ou 1 sets configuram situações atípicas, geralmente associadas a desistências (*RET*), desqualificações (*DEF*) ou vitórias por ausência (*W/O*). Esses casos não refletem o

desenvolvimento completo de um jogo e, por isso, foram removidos da base de dados para garantir uma análise mais coerente e comparável.

Além disso, foram também excluídas todas as partidas que continham situações anómalas como as mencionadas anteriormente ((RET), (DEF) ou (W/O)), independentemente do número de sets, de modo a garantir que a análise fosse realizada apenas sobre jogos decorridos dentro da normalidade.

Dessa forma, a esmagadora maioria das partidas é composta por 2 ou 3 sets, sendo raros os jogos que se estendem a 4 ou 5 sets.

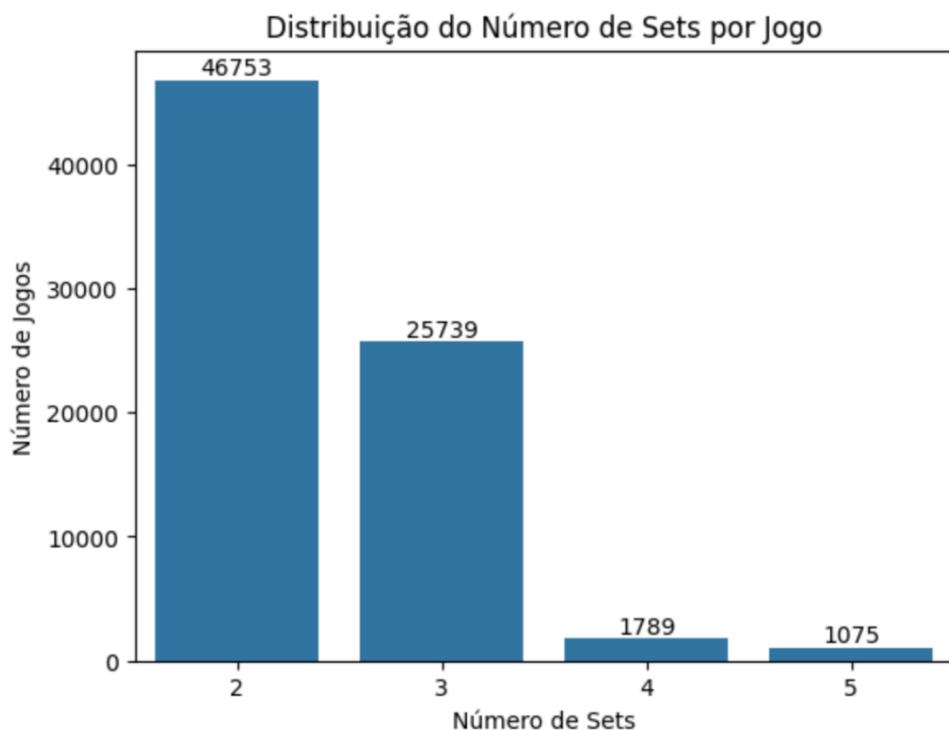


Fig. 5 - Distribuição das partidas pelo número de sets

Os jogos de ténis podem ser decididos de duas formas: à melhor de 3 sets onde o jogador precisa de ganhar 2 sets para vencer, ou à melhor de 5 sets onde o jogador precisa de ganhar 3 sets para vencer. Assim, foi possível distinguir as partidas decididas à melhor de 3 sets das decididas à melhor de 5 sets, através do resultado do jogo (game_score). O processo consistiu em criar uma função que contava quantos sets cada jogador venceu. Com base nesse número, classificava o formato do jogo: se algum dos jogadores tivesse vencido 3 sets, a partida era considerada à melhor de 5. Os jogos com 2 ou 4/5 sets eram automaticamente classificados como à melhor de 3 e 5 sets, respectivamente.

A grande maioria dos jogos foi disputada à melhor de 3 sets como é possível observar na figura abaixo, sendo as partidas à melhor de 5 sets significativamente menos frequentes.

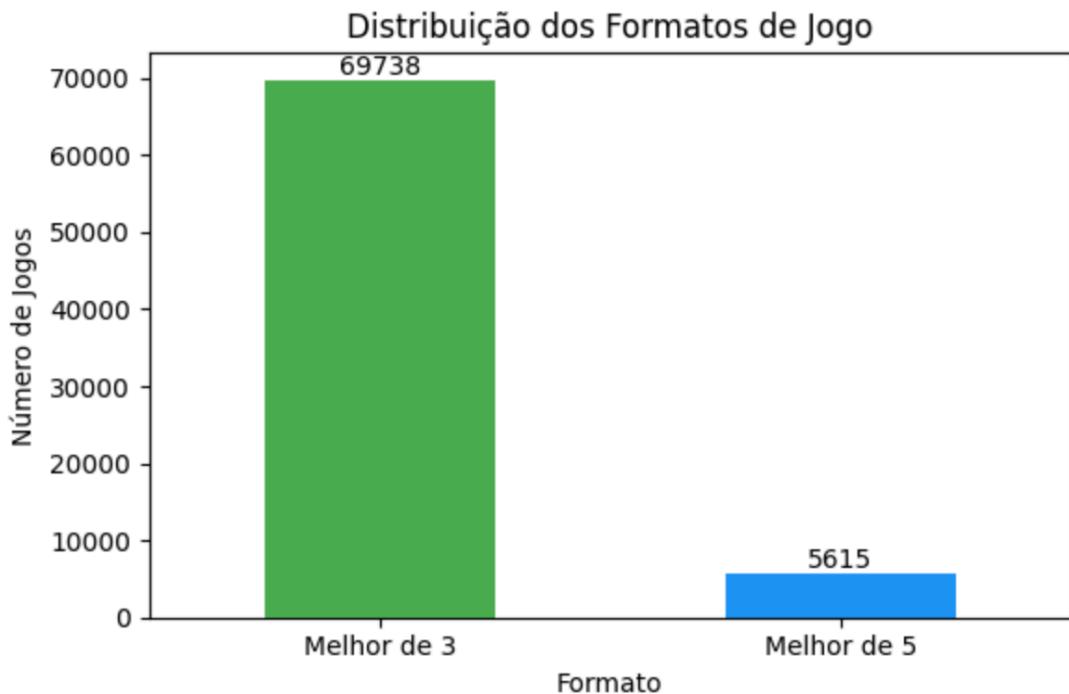


Fig. 6 - Distribuição dos formatos de jogo

Para manter a consistência na análise e facilitar a construção dos modelos, devido à diferença no formato de jogo, decidiu-se eliminar também as partidas disputadas à melhor de 5 sets, focando-nos exclusivamente em jogos à melhor de 3 sets, que representam a norma na maioria dos torneios profissionais.

No final, a base de dados fica com 69 738 partidas à melhor de 3 sets, sendo 46 753 delas decididas com apenas 2 sets enquanto 22 985 estenderam-se aos 3 sets.

Match Round

A variável *match_round* indica a fase do torneio em que a partida foi realizada. É importante referir que diferentes torneios podem ter estruturas distintas de rondas. Por exemplo, o *Round of 128* ocorre apenas em torneios de grande dimensão como é o caso do *US Open*. Observando a distribuição desta variável, destaca-se que a *Round of 32* é a fase que abrange o maior número de partidas, visto que é a primeira fase na maioria dos torneios. Este padrão é compreensível, uma vez que os torneios de ténis seguem um formato eliminatório: nas fases iniciais existem mais jogadores em competição, o que se traduz num maior número de jogos. À medida que os jogadores avançam na competição e as fases se tornam mais decisivas, o número de partidas naturalmente diminui.

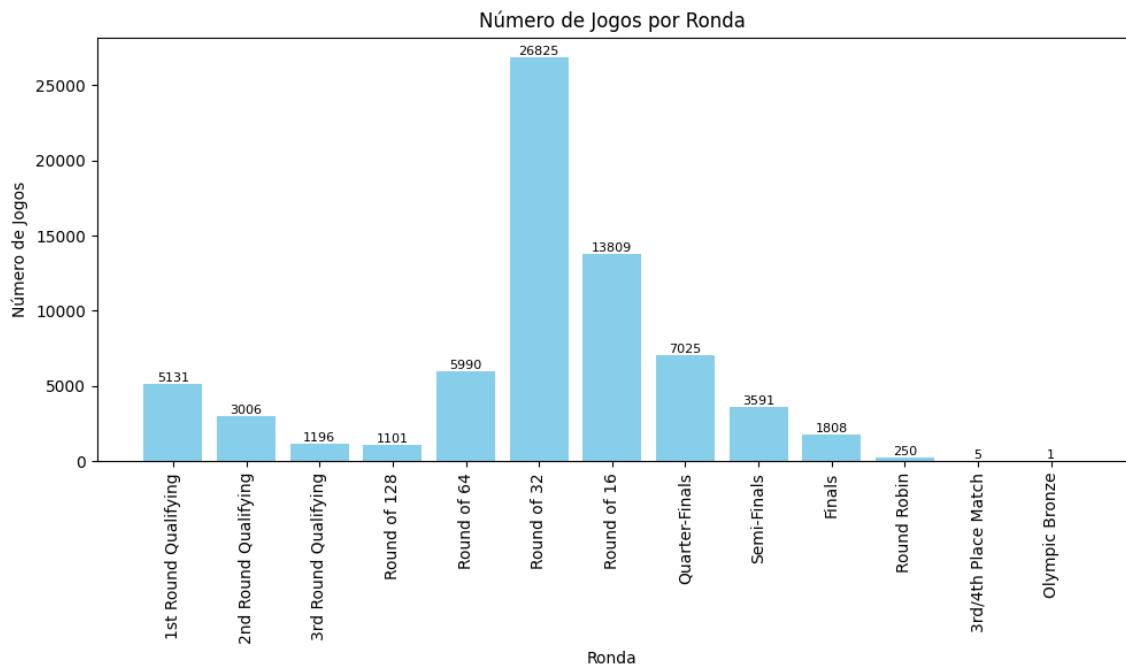


Fig. 7 - Número de jogos por ronda

Os jogos de classificação fazem parte de uma fase prévia ao quadro principal do torneio e servem como uma espécie de "filtro" para jogadores com *ranking* mais baixo que ainda não têm entrada direta no torneio. Para reduzir a dimensionalidade da variável decidimos agregar as três categorias correspondentes a essa fase (*1st Round Qualifying*, *2nd Round Qualifying* e *3rd Round Qualifying*) numa só categoria denominada "*Qualifiers*". Na mesma linha de pensamento, existem também partidas disputadas em formato de *Round Robin*, onde os jogadores se enfrentam em grupos antes de avançar às fases eliminatórias. Como essas partidas também não fazem parte do quadro eliminatório principal, optámos por agrupá-las aos jogos de qualificação, sob a mesma categoria "*Qualifiers*".

Para as categorias *3rd/4th Place Match* e *Olympic Bronze* que continham apenas 5 e 1 partidas, respetivamente, foi tomada a decisão de retirá-las da base de dados.

Ground

É importante perceber quais as características que envolvem os jogos realizados nos Estados Unidos, e para isso, foi construído um gráfico do número de jogos realizados por cada tipo de pavimento (*Hard*, *Clay*, *Carpet* e *Grass*). A superfície onde é realizada a partida pode influenciar o estilo de jogo e o desenrolar das partidas.

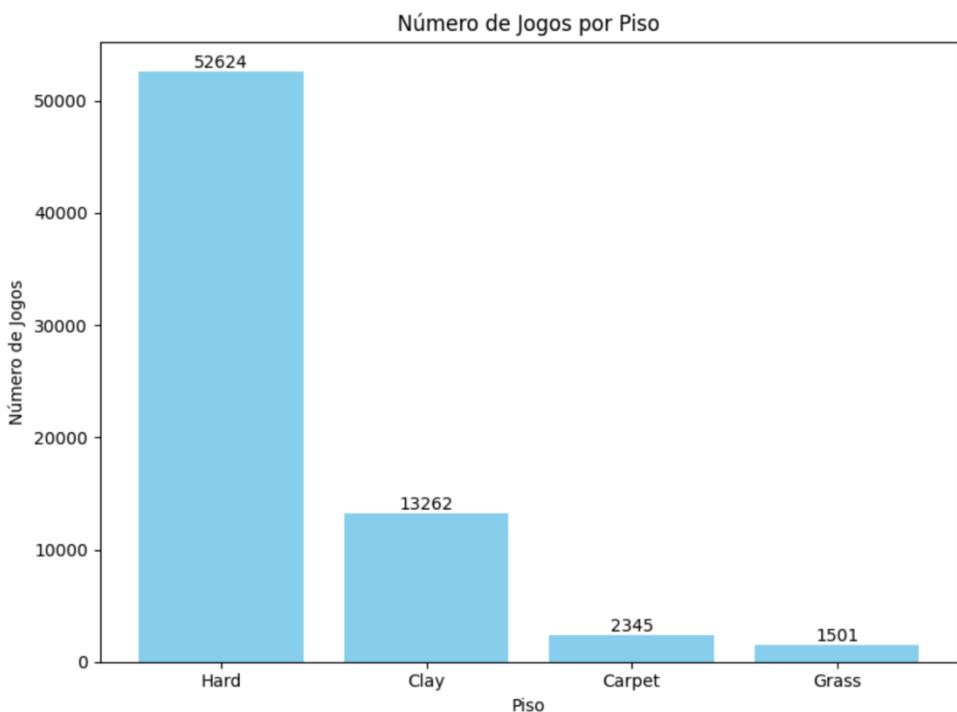


Fig. 8 - Número de jogos por tipo de pavimento

A análise revela uma clara predominância de partidas disputadas em piso duro (*Hard*), totalizando 52 624 jogos. Este domínio está alinhado com a realidade tenística norte-americana, onde a maioria dos torneios é realizado neste tipo de superfície. O piso duro é amplamente adotado por ser mais acessível, de fácil manutenção e adequado a diferentes condições climáticas.

Em contraste, o piso de terra batida (*Clay*) aparece com 13 262 jogos, um valor relativamente elevado tendo em conta que este tipo de superfície não é tradicionalmente o mais comum nos Estados Unidos. Este número poderá refletir a existência de torneios de menor escala ou torneios do circuito *Challenger*.

Já o piso de carpete (*Carpet*), atualmente extinto no circuito principal, soma 2345 jogos. A sua presença no gráfico é justificada pela grande amplitude do intervalo de tempo da base de dados que inclui jogos em que este tipo de superfície ainda era utilizado. Por fim, a relva (*Grass*), com apenas 1501 jogos, representa a superfície menos utilizada nos Estados Unidos, sendo este valor esperado dada a escassez de torneios neste tipo de piso no país, cuja tradição está sobretudo enraizada na Europa.

Hand

O estudo das preferências de golpe no ténis profissional oferece uma perspetiva interessante sobre as características motoras dos atletas e a sua possível influência no desempenho em jogo.

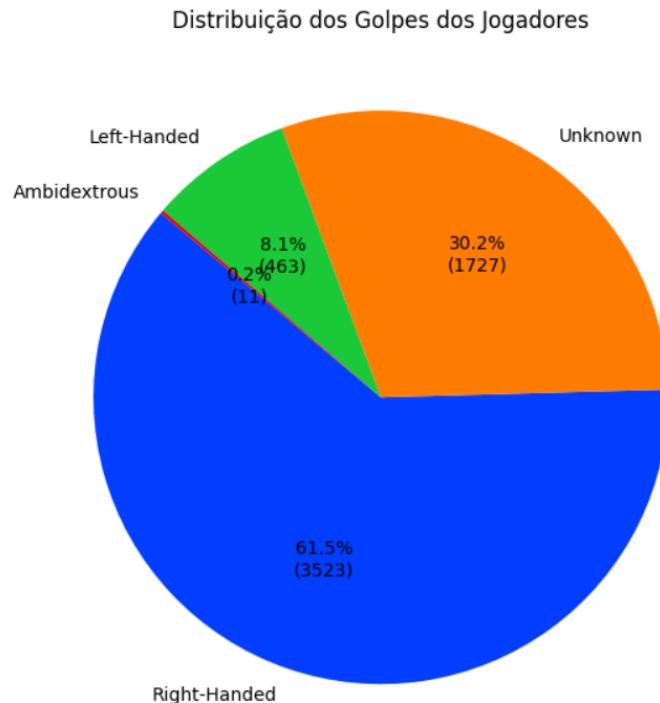


Fig. 9 – Golpes usados pelos jogadores

A distribuição dos golpes dos jogadores de ténis reflete uma característica amplamente conhecida do comportamento humano: o domínio da mão direita. A maioria dos seres humanos realiza as suas tarefas preferencialmente com a mão direita, o que se confirma também no ténis profissional. No conjunto de dados analisado, observa-se que a esmagadora maioria dos atletas é classificado como "*Right-Handed*" (destros), o que está em consonância com os estudos populacionais gerais.

Durante a construção da base de dados, houve o esforço de reunir a informação do golpe predileto de todos os jogadores. No entanto, para uma parcela significativa (1727 atletas), os dados não estavam disponíveis nas fontes oficiais, como o site da ATP, o que resultou na categorização como "*Unknown*".

Além disso, a categoria "*Ambidextrous*" (usar ambas as mãos) foi identificada em apenas 11 jogadores, sendo casos bastante raros. Por esse motivo, optou-se por remover essa categoria da análise, atribuindo um foco nos dois grupos mais representativos: destros e canhotos.

Evolução dos prémios dos torneios ao longo dos anos

Com o objetivo de facilitar a análise temporal dos dados e a criação de estatísticas agregadas por ano, foi criada uma variável correspondente ao ano do torneio

(*tournament_year*), extraída a partir da data de início de cada evento. Esta transformação revelou-se mais conveniente, especialmente na preparação do *dataset* para os modelos de *machine learning*, uma vez que permite observar padrões ao longo do tempo e realizar agrupamentos anuais de forma simples e eficaz.

Durante este processo, foi também tomada a decisão de eliminar os jogos realizados antes de 1937, dado que estes representavam apenas quatro observações no total. Além da baixa representatividade, havia ainda um grande intervalo temporal até 1968 sem qualquer registo. Assim, a exclusão destes dados antigos visa garantir uma maior coerência histórica e melhorar a qualidade geral da análise.

Graças à extração dos anos de realização dos torneios, foi possível perceber como os prémios monetários dos mesmos variaram ao longo do tempo. No entanto, comparar diretamente valores de diferentes períodos de tempo ignora o impacto da inflação — 1.000 dólares nos dias de hoje não têm o mesmo poder de compra que no passado. Então, para compreender a real evolução dos prémios monetários ao longo dos anos, foi necessário ajustar os valores históricos à inflação. Este processo de desinflação utilizou o índice de preços ao consumidor (*CPI – Consumer Price Index*) como referência, com o ano base definido como 2019 (*CPI = 255.7*). O ano de 2019 foi escolhido como ano base para o ajuste dos valores à inflação por ser o último período económico estável antes da pandemia de *COVID-19* (a partir de 2020) e o subsequente aumento generalizado da inflação. A correção teve como objetivo garantir uma comparação justa entre os valores de diferentes épocas, eliminando o efeito da inflação acumulada ao longo dos anos.

A fórmula usada para desinflacionar o valor do prémio de um certo ano foi a seguinte:

$$\text{valor ajustado} = \text{valor original} \times \frac{\text{CPI ano referência (2019)}}{\text{CPI ano do torneio}}$$

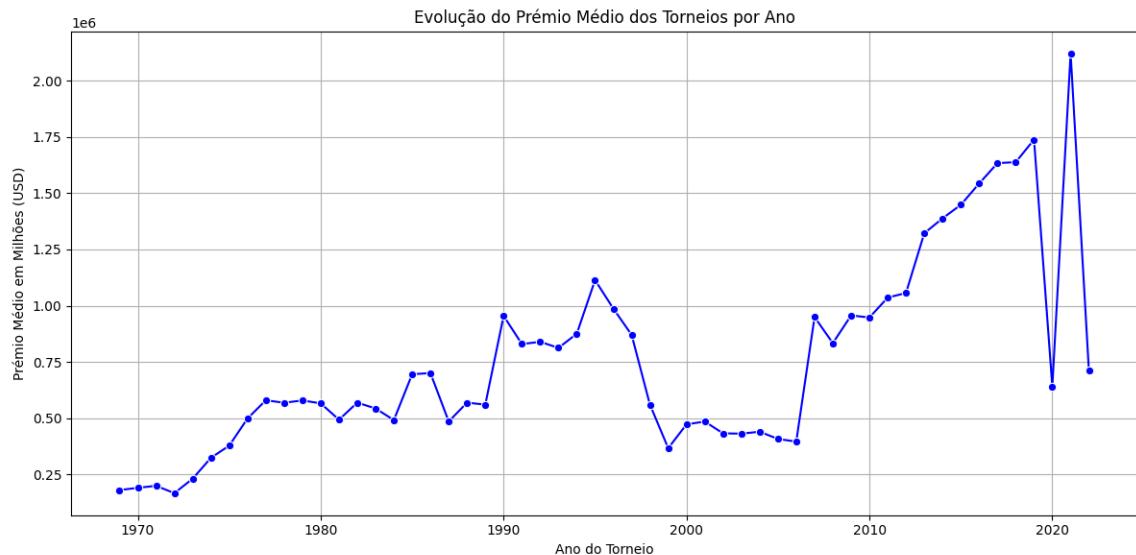


Fig. 10 - Evolução do prémio médio (em milhões de dólares) dos torneios

De forma geral, observa-se uma tendência de crescimento nos valores médios desde o final da década de 1960 até aos dias de hoje. Este aumento reflete a crescente profissionalização do circuito, a expansão global do ténis enquanto espetáculo desportivo, e o consequente reforço de receitas provenientes de patrocínios, transmissões televisivas e direitos comerciais.

Durante os anos 70 e 80, o crescimento foi relativamente moderado e constante, com os prémios médios situando-se abaixo de 1 milhão de dólares. A partir da década de 1990, começa a observar-se um incremento mais acentuado, apesar de algumas oscilações pontuais nos anos 90 e início dos anos 2000. Estes períodos de estagnação e ligeira queda são explicados pela Grande Recessão ocorrida na década de 2000 que foi marcada por diversas crises.

Após a estabilização económica em 2010 houve um crescimento mais consistente e acentuado dos prémios até à ocorrência de um novo flagelo mundial, o *COVID-19*. O impacto da pandemia, por volta de 2020, é particularmente evidente no gráfico, com uma quebra acentuada no valor médio dos prémios atribuídos. Esta descida abruta explica-se pela redução de torneios realizados, pela ausência de público e pelas dificuldades financeiras enfrentadas pelas organizações desportivas nesse período. Em 2021 regista-se uma recuperação significativa, com um pico de valorização que poderá estar associado a torneios isolados com premiações excepcionalmente elevadas. Importa ainda referir que os dados relativos ao ano de 2022 não estão completos na base de dados utilizada, o que impossibilita a sua análise.

Em suma, o gráfico reflete não só a evolução natural da indústria do ténis ao longo do tempo, mas também a sua vulnerabilidade a acontecimentos externos, sendo o ano de 2020 um claro exemplo de como fatores globais podem impactar significativamente o circuito profissional.

Criação de novas variáveis

No processo de preparação dos dados, foi decidido criar algumas variáveis com o objetivo de enriquecer o conjunto de dados e cumprir os requisitos definidos no enunciado.

Age difference

A idade de cada jogador foi calculada a partir da diferença entre a data de início do torneio e a data de nascimento do respetivo jogador.

A partir do cálculo da idade de cada jogador foi criada uma nova variável designada por *age_difference*, que representa a diferença absoluta entre as idades do vencedor e do perdedor em cada partida.

Esta variável permite analisar se o contraste de idades entre os jogadores pode ter impacto na duração dos jogos. Diferenças muito acentuadas de idade podem refletir uma desproporção ao nível da experiência, resistência física ou até maturidade tática, o que pode influenciar o desenrolar da partida e, consequentemente, o número de *sets* necessários para se encontrar um vencedor.

Rank difference

Outra variável criada foi a diferença absoluta entre os *rankings* dos dois jogadores presentes em cada partida. Esta variável permite captar o nível de desequilíbrio teórico entre os adversários — quanto maior esta diferença, maior a discrepância esperada de desempenho entre eles. Ao considerar o valor absoluto da diferença, pretende-se focar no desnível em si, independentemente de qual o jogador com o *ranking* mais elevado. Esta métrica revelou-se particularmente útil para avaliar a probabilidade de um jogo ser mais ou menos disputado, o que se pode refletir no número de *sets* da partida.

Para garantir a consistência desta variável, foi necessário tratar os casos em que um ou ambos os jogadores apresentavam valores nulos ou inexistentes. Nestas situações, foi imputado o valor da mediana dos *rankings* disponíveis na base de dados. Esta decisão visou reduzir o impacto de *outliers* e evitar distorções na distribuição dos dados, assegurando uma representação mais realista da diferença de níveis entre os jogadores em campo.

Total_matches_difference

Adicionalmente, foram criadas duas novas variáveis auxiliares que contabilizam o número total de jogos disputados por cada jogador ao longo do *dataset*, independentemente de terem vencido ou perdido: *winner_total_matches* e *loser_total_matches*.

Para garantir que o número de partidas contabilizado para cada jogador correspondesse apenas aos jogos disputados antes de cada partida (excluindo, portanto, o jogo atual), foi essencial ordenar os dados corretamente. Assim, os registo da base de dados foram ordenados cronologicamente pela data de início dos torneios e, adicionalmente, pela ronda do mesmo. Esta ordenação assegura que o cálculo acumulado respeite a

sequência temporal dos eventos, permitindo calcular com precisão o número de partidas anteriores de cada jogador até àquele momento específico da carreira.

A partir dessas variáveis, foi criada a variável principal “*total_matches_difference*” que representa a diferença absoluta no número total de jogos previamente disputados entre os jogadores de cada partida.

Esta variável tem como principal objetivo quantificar o desnível de experiência competitiva entre os atletas, partindo do princípio de que, quanto maior foi número de jogos já disputados, maior será o grau de experiência acumulada. Portanto, assume-se que quanto menor for a diferença de jogos totais já disputados entre os jogadores, maior probabilidade de o jogo ser equilibrado e, por consequente, exigir um número de *sets* maior para encontrar o vencedor.

Forehand_difference

Com o objetivo de analisar a influência da lateralidade no desenrolar dos jogos, foi decidido criar uma nova variável binária “*forehand_difference*”, a qual assume valor 1 quando ambos os jogadores apresentam a mesma mão dominante (ambos destros ou ambos canhotos) e 0 quando possuem mãos opostas (e casos de “*Unknown*”). Esta variável procura captar possíveis efeitos associados à simetria ou assimetria entre os estilos de jogo dos jogadores em cada partida.

A lateralidade é um fator que pode ter impacto no ténis: influencia significativamente o posicionamento dos jogadores no campo, a direção dos serviços e a forma como o adversário devolve a bola. Jogadores canhotos, por constituírem uma minoria nos circuitos de ATP, podem introduzir elementos de imprevisibilidade e exigir uma maior adaptação tática do seu oponente. Considerou-se, portanto, que esta diferença pode afetar o equilíbrio competitivo da partida e, eventualmente refletir em número de *sets* mais elevados.

Height Difference

Foi também criada uma nova variável “*height_difference*”, que representa a diferença absoluta entre as alturas do vencedor e do perdedor em cada partida. A hipótese por trás dessa variável é que uma diferença significativa entre os jogadores poderia influenciar a dinâmica dos jogos, afetando fatores como o alcance e a potência do serviço. Embora não existam estudos que estabeleçam uma relação direta entre a altura e o número de *sets* disputados, considerou-se pertinente analisar o eventual impacto que

pode ter em número de sets necessários para encontrar um vencedor. Especificamente, pressupõe-se que uma maior diferença de alturas pode estar associada a um menor número de sets, dada uma possível vantagem física do jogador mais alto.

Nota: Existiam vários jogadores cujas alturas eram valores ambíguos, como por exemplo 15 centímetros. Ora, é humanamente impossível uma pessoa ter apenas 15 centímetros de altura, logo esse valor seria um erro no preenchimento da base de dados que advinha do site *ATP Tour*. Para esses jogadores bem como para os que tinham alturas nulas, foi imputada a altura média dos jogadores da sua nacionalidade presentes na base de dados. Esta abordagem permitiu manter a coerência das variáveis numéricas sem recorrer à eliminação de regtos potencialmente valiosos para a análise.

Average Sets Difference

Foram criadas uma variável para cada jogador, *winner_avg_sets* e *loser_avg_sets* que representam, respetivamente, o número médio de sets disputados pelos jogadores (vencedores e perdedores) nas 10 partidas anteriores, considerando a ordem cronológica dos jogos com base na data de início do torneio e na ronda da competição. Esta abordagem garante que o histórico de cada jogador reflita apenas o seu desempenho passado até ao momento do jogo em questão.

Para o caso em que um jogador não tenha disputado 10 jogos anteriormente, é atribuído um valor padrão de 2.5, correspondente ao ponto médio entre 2 e 3 sets. Esta escolha garante que a variável mantenha um valor neutro e realista sempre que o histórico do jogador seja insuficiente para calcular a média com base empírica.

A partir destas variáveis foi criada a *avg_sets_difference*, que representa a diferença absoluta entre as médias de sets disputados por cada jogador. Esta variável visa captar o grau de semelhança no padrão de desempenho recente entre os dois jogadores. Um valor próximo de zero sugere que ambos apresentam um histórico semelhante em termos de duração dos seus jogos, o que pode indicar maior equilíbrio no confronto. Por outro lado, valores mais elevados podem sinalizar uma assimetria nas prestações, o que, teoricamente, poderá estar associado a encontros menos equilibrados e com menor probabilidade de se prolongarem até ao terceiro set.

Ratio Difference

Com o objetivo de incorporar o histórico de desempenho dos jogadores ao longo do tempo, foi criada uma nova variável designada por *ratio_difference*. Esta variável visa

medir o desequilíbrio entre os dois adversários com base na sua performance passada, mais concretamente, na proporção de vitórias em relação ao total de jogos já disputados por cada um.

O rácio de vitórias de um jogador foi definido como o quociente entre o número de jogos vencidos e o número total de jogos disputados (vitórias + derrotas). Este valor varia entre 0 (jogador que perdeu todos os jogos) e 1 (jogador invicto até ao momento). Para garantir a consistência temporal, os jogos foram previamente ordenados pela data do torneio e pela fase da ronda, de forma que o cálculo do rácio de cada jogador fosse sempre baseado apenas nos jogos anteriores à observação atual.

Dessa forma, a cada nova linha (jogo), é calculado o rácio de vitórias do jogador vencedor e do jogador derrotado com base no histórico acumulado até esse ponto. A variável *ratio_difference* resulta da diferença absoluta entre esses dois rácios. Assim, valores próximos de zero indicam que os jogadores têm um desempenho semelhante até ao momento do encontro, enquanto valores mais altos apontam para um maior desequilíbrio entre os adversários.

A hipótese subjacente é que partidas entre jogadores com desempenhos mais próximos tendem a ser mais disputadas, sendo mais provável que se prolonguem até ao terceiro *set*, ao contrário de confrontos entre adversários com históricos de performance muito distintos.

Home Game

A variável binária *home_game* foi criada com o objetivo de captar o possível efeito de “fator casa” no desempenho dos jogadores vencedores (“*winner*”) de cada partida. Esta variável assume o valor 1 quando o respetivo jogador é dos Estados Unidos — ou seja, está a competir no seu país de origem — e 0 caso contrário.

O foco exclusivo nos vencedores justifica-se pelo facto de o desempenho ser avaliado com base na vitória. Como a criação desta variável tem como objetivo em perceber se jogar em casa influencia positivamente os resultados, basta analisar os casos em que o jogador venceu a partida.

A expectativa é que, se o fator casa tiver algum impacto, os jogadores norte-americanos terão mais vantagem e, consequentemente, os jogos terão menos *sets*. Este possível efeito torna-se relevante, uma vez que competir perante o público local e em condições climáticas e culturais familiares, pode influenciar positivamente o desempenho dos jogadores. Ao introduzir estas variáveis, pretende-se, portanto, avaliar se os jogadores

norte-americanos tendem a ter um desempenho diferente quando competem nos Estados Unidos.

Análise univariada com a variável alvo

Número de sets vs rank difference

De modo a compreender a relação entre a diferença de *rank* dos jogadores e o número de *sets* disputados nos jogos, foi construído um gráfico de violino que permite visualizar simultaneamente a distribuição e a densidade das diferenças de *rank* para jogos que terminam em 2 ou 3 *sets*.

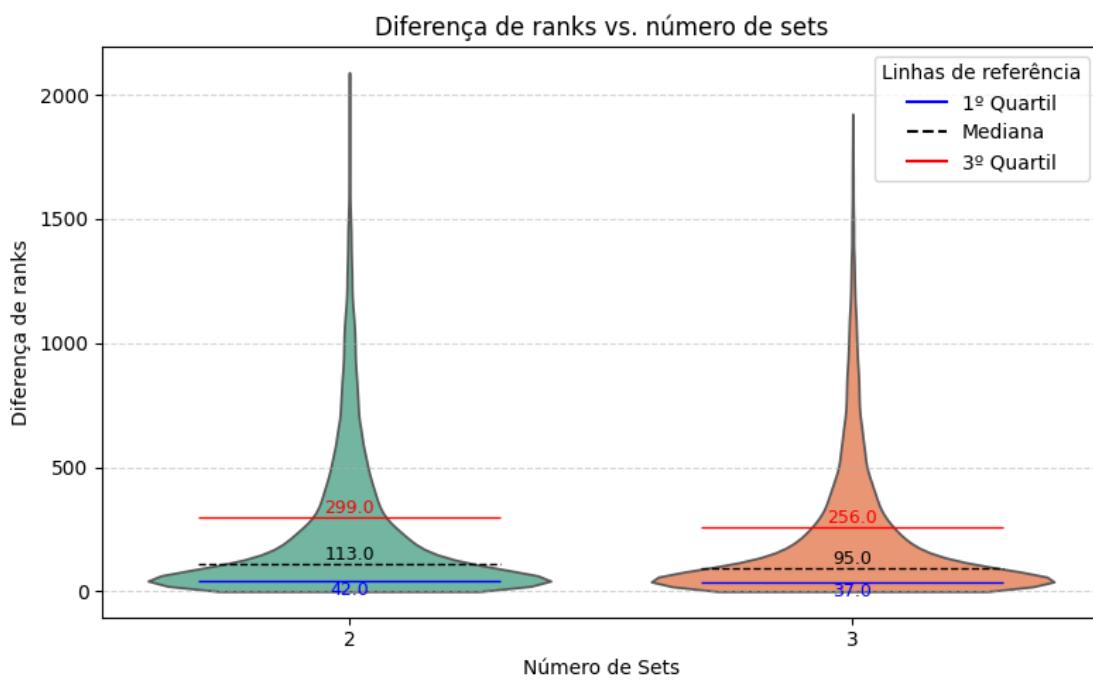


Fig. 11 - Densidade do rank_difference em relação ao número de sets

A análise do gráfico evidencia que, independentemente do número de *sets*, a maior parte dos jogos ocorre entre jogadores com diferenças de *rank* relativamente pequenas. Esta tendência é visível pelo alargamento das regiões centrais dos violinos, próximo da diferença de *rank* igual a zero, indicando uma maior concentração de jogos equilibrados.

No entanto, observa-se uma diferença subtil entre os dois casos: jogos que se prolongam até aos 3 *sets* tendem a ocorrer com maior frequência entre os jogadores com *ranks* próximos, sugerindo jogos mais equilibrados (a mediana é ligeiramente inferior). Em contraste, os jogos decididos em apenas 2 *sets* apresentam uma distribuição mais

dispersa, com uma cauda do violino mais comprida, indicando uma prevalência de encontros entre os jogadores com disparidade significativa nos *ranks*.

Esta observação está em linha com os padrões anteriormente identificados: partidas mais equilibradas, seja em termos de número de *sets* ou de proximidade no *rank*, tendem a refletir uma maior competitividade entre os jogadores, o que se traduz numa maior probabilidade de o jogo se prolongar até ao terceiro *set*.

Match Round vs number_of_sets

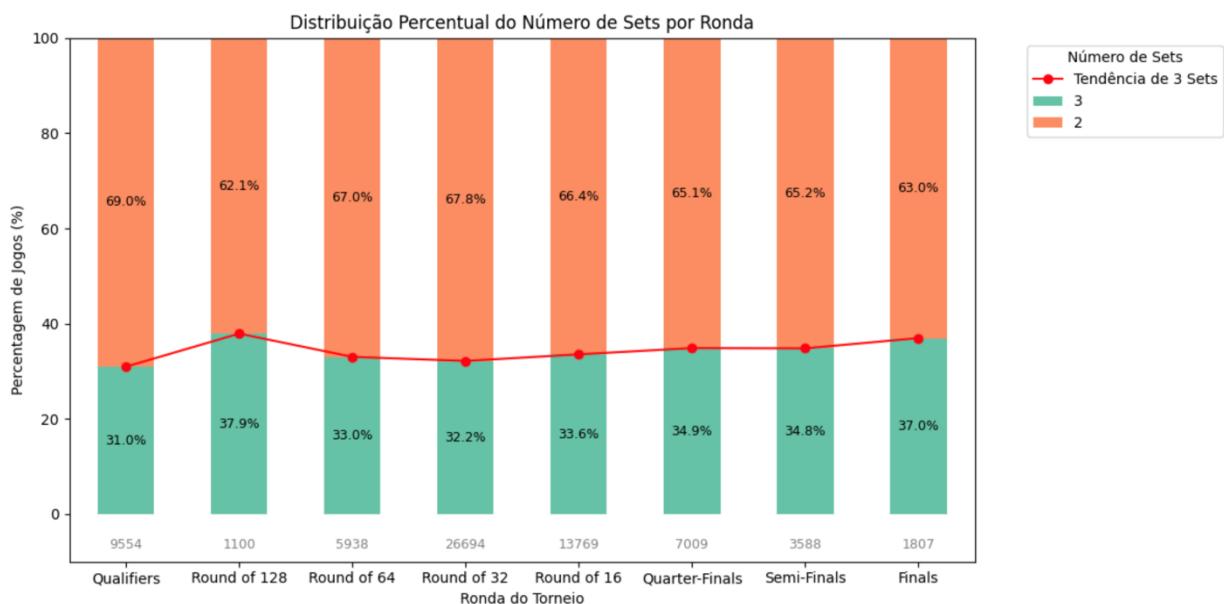


Fig. 12 - Proporção do número de sets por ronda

A figura acima ilustra a distribuição do número de *sets* (2 ou 3) em cada fase do torneio, considerando apenas partidas disputadas à melhor de 3 *sets*. Observa-se que em todas as fases, a maioria dos jogos são decididos em 2 *sets*, indicando a existência de um domínio por parte de um dos jogadores.

Seria de esperar que a proporção de jogos com 3 *sets* aumentasse à medida que se avança para as fases finais, já que nestas tende a existir maior equilíbrio e competitividade entre os jogadores. De facto, observa-se um ligeiro aumento na proporção de jogos com 3 *sets* nas fases mais avançadas, ainda que este acréscimo seja pouco expressivo e quase irrelevante em termos estatísticos.

Este resultado pode estar relacionado com o facto de, mesmo nas fases finais, alguns jogadores conseguirem impor-se de forma clara, beneficiando de fatores como o *ranking* e a experiência.

Modelação

Com o *dataset* devidamente limpo e estruturado, o próximo passo consistiu na construção de modelos preditivos com o objetivo de prever o número de *sets* necessários para concluir um jogo de ténis.

Este problema de previsão enquadra-se no âmbito da classificação binária, uma vez que o valor a prever assume apenas dois estados distintos: 2 *sets* (partidas decididas em dois *sets* resultando de vitórias consecutivas de um jogador) ou 3 *sets* (partidas que exigem o terceiro *set* para desempate).

Eliminação dos NA's

Durante o pré-processamento dos dados, foram identificados alguns registo com valores em falta (*NA*). Para garantir a consistência e a aplicabilidade dos modelos de classificação, optou-se pela remoção desses registo. No total, foram eliminadas 93 linhas com dados incompletos. Esta abordagem foi escolhida por se tratar de uma quantidade reduzida face ao volume total de dados, não comprometendo significativamente a representatividade do conjunto nem a validade das análises subsequentes.

Correlações e associações

Para garantir a qualidade e robustez dos modelos, foi conduzida uma análise das variáveis explicativas. Essa análise centrou-se na avaliação da relação entre cada variável preditora e a variável alvo, bem como na verificação da existência de colinearidade entre os próprios preditores. Este processo foi essencial para evitar redundâncias e selecionar apenas os preditores mais relevantes, assegurando que os modelos fossem capazes de capturar padrões presentes nos dados.

Associações entre as variáveis numéricas e a variável alvo

Com o objetivo de avaliar o grau de associação entre as variáveis numéricas do conjunto de dados e a variável alvo — que assume valores categóricos (2 ou 3 sets) — recorreu-se ao coeficiente *Eta*. Esta medida estatística é particularmente adequada quando se pretende analisar a relação entre uma variável quantitativa e uma variável qualitativa.

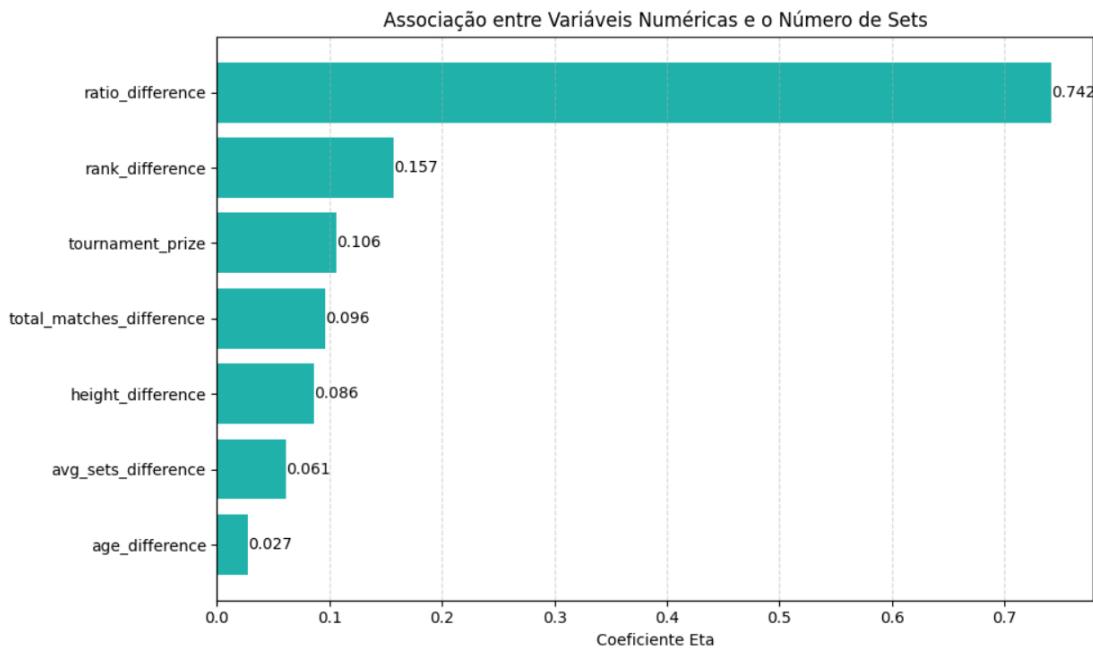


Fig. 13 - Associação entre as variáveis numéricas e a variável alvo

Com base nos coeficientes Eta calculados entre a variável alvo *number_of_sets* e as diversas variáveis numéricas preditoras, é possível avaliar o grau de associação existente entre cada uma delas e a duração de um jogo de ténis (decidido em 2 ou 3 sets). O coeficiente Eta varia entre 0 e 1, sendo que valores mais próximos de 1 indicam uma associação mais forte.

Os valores do coeficiente Eta revelaram, de forma geral, associações fracas com a variável alvo, no entanto destaca-se a variável *ratio_difference* que está fortemente associada com o número de sets (coeficiente Eta = 0.742), evidenciando que a diferença no histórico de vitórias e derrotas entre os jogadores é um forte indicador do desfecho da partida. Este resultado está alinhado com a expectativa de que jogos entre jogadores com desempenhos semelhantes tendem a ser mais equilibrados e, portanto, mais propensos a irem até ao terceiro set.

Outras variáveis com associação moderada incluem *rank_difference* (0.157), *tournament_prize* (0.106), *total_matches_difference* (0.096) e *height_difference* (0.086). Estas variáveis captam diferentes dimensões do confronto, como o desequilíbrio entre os *rankings*, a experiência acumulada, o valor monetário em jogo (potencial motivador) e diferenças físicas que podem influenciar o desenrolar do jogo.

Variáveis como a diferença de idades (0.027) e do número médio de sets (0.061) demonstraram associações mais fracas com o número de sets. Embora essas variáveis tenham relevância contextual, a sua capacidade preditiva isolada sobre o desfecho da partida, neste caso, parece ser mais limitada.

Em síntese, os resultados indicam que variáveis relacionadas com a performance histórica dos jogadores (como o rácio de vitórias/derrotas e *ranking*) são as que mais se associam à duração das partidas, sendo assim valiosas para inclusão nos modelos preditivos.

Associações entre as variáveis nominais e a variável alvo

Para as variáveis categóricas, foi usado o *V de Cramer* para medir a associação com a variável alvo bem como entre elas.

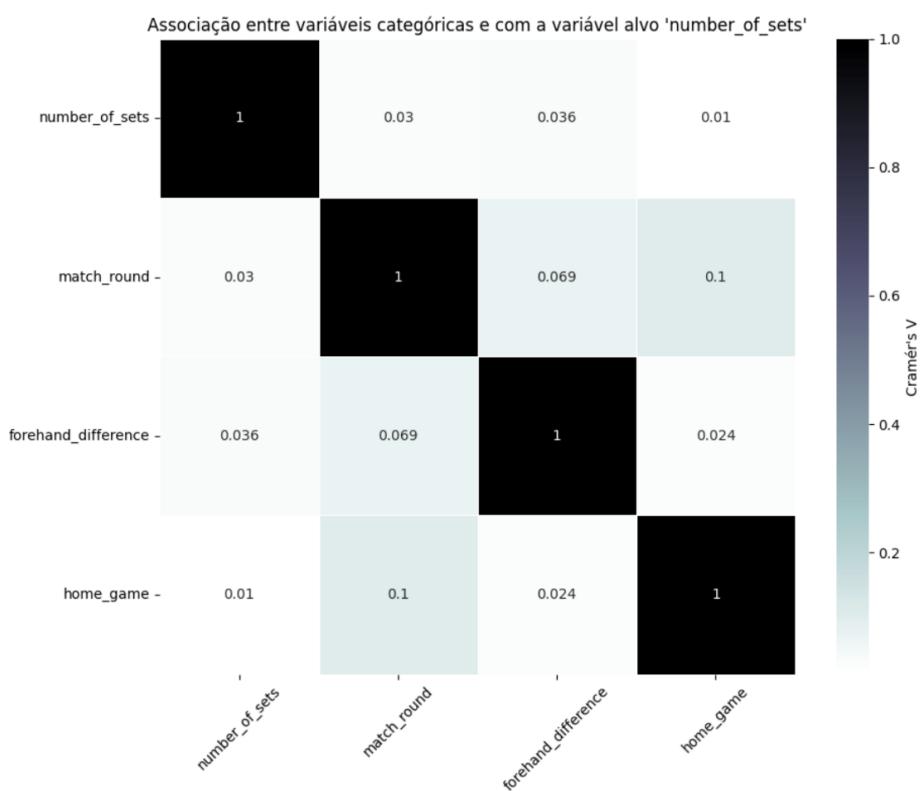


Fig. 14 - Associação entre as variáveis categóricas e a variável alvo

As associações entre as variáveis categóricas e o número de *sets* (indicadas na primeira linha ou coluna) são bastante fracas, sendo a mais elevada de apenas 0.036, registada na diferença entre os golpes dos jogadores. Este valor sugere que não existe uma relação significativa entre estas variáveis e o desfecho do jogo em termos de número de *sets*. Ainda assim, essas variáveis podem revelar-se úteis quando combinadas com outras durante a modelação.

Correlações entre os preditores numéricos

Para avaliar as correlações entre as variáveis numéricas, foi utilizado o coeficiente de correlação de *Pearson*, apropriado para variáveis contínuas. Esta análise teve como principal objetivo identificar possíveis situações de colinearidade entre os preditores.

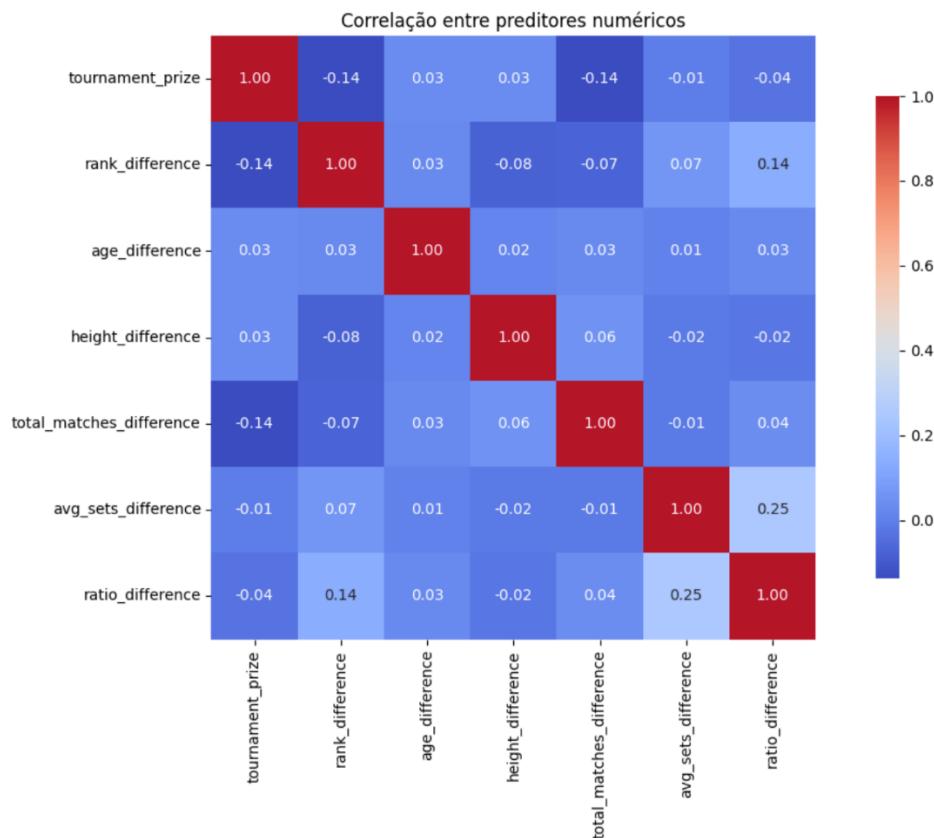


Fig. 15 - Correlação entre as variáveis preditoras numéricas

Com base na matriz de correlação entre os preditores numéricos, é possível observar que a grande maioria das correlações entre as variáveis é muito baixa (próxima de zero), indicando que os preditores são quase independentes entre si. Isso é um aspecto positivo, pois reduz o risco de multicolinearidade nos modelos.

A maior correlação observada foi entre *avg_sets_difference* e *ratio_difference* (0.25), embora ainda assim seja considerada fraca. Isso sugere que, apesar de ambas captarem dimensões de desempenho entre os jogadores, cada uma traz uma perspectiva distinta: o rácio é uma medida agregada de sucesso histórico, enquanto a média de sets disputados reflete a intensidade ou equilíbrio dos jogos mais recentes. Assim, ambas contribuem de forma complementar para a compreensão do desempenho dos atletas.

Em suma, a ausência de correlações elevadas entre as variáveis demonstra que estas podem contribuir de forma complementar para os modelos preditivos, permitindo

capturar diferentes dimensões dos dados (física, técnica, histórica e contextual) sem haver sobreposição de informação (multicolinearidade).

Correlação entre os preditores numéricos e categóricos

A análise da matriz de correlação entre as variáveis preditoras numéricas e categóricas revela, de forma geral, correlações baixas, indicando uma relação limitada entre essas dimensões. Os coeficientes foram calculados utilizando uma métrica apropriada para misturar variáveis de naturezas diferentes — numéricas e nominais — permitindo avaliar a força da associação entre elas. Neste caso, foi utilizado o coeficiente *Eta*, adequado para medir a associação entre variáveis numéricas e nominais.

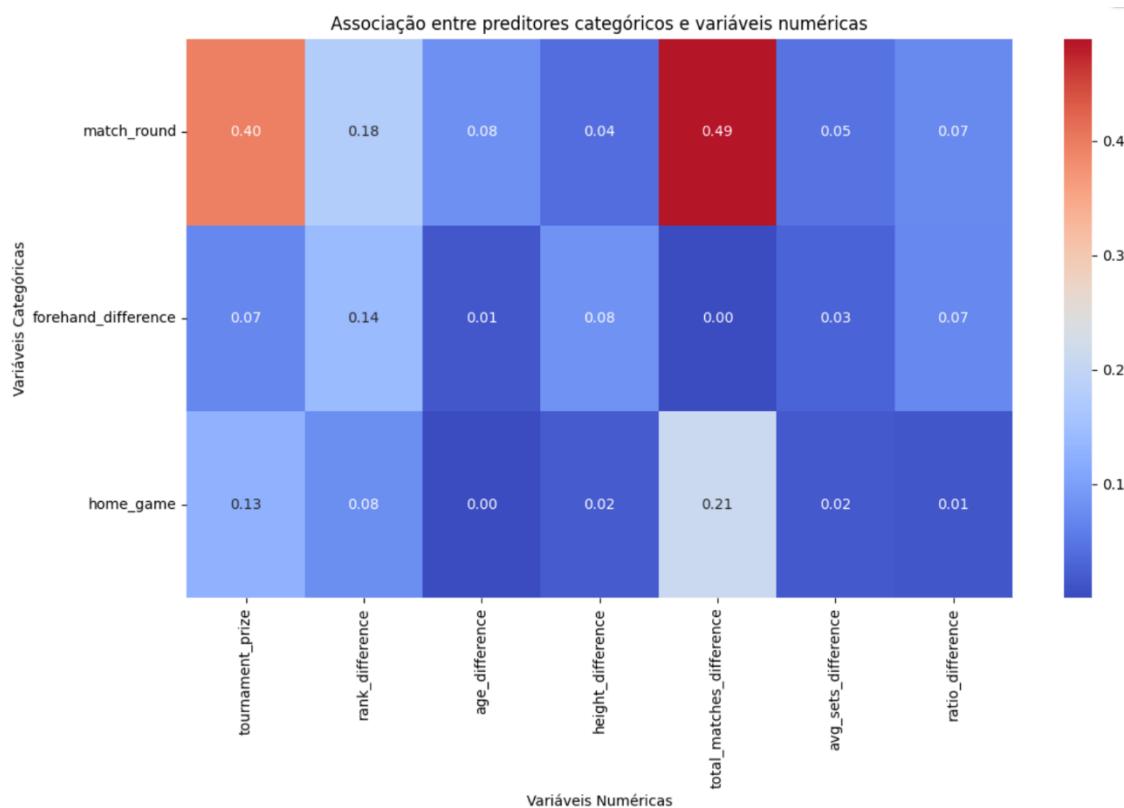


Fig. 16 - Associação entre as variáveis preditoras categóricas e variáveis numéricas

Apesar de a maioria das correlações nesta matriz de correlação serem bastante baixas, indicando uma fraca associação entre as variáveis categóricas e numéricas, há algumas exceções que merecem o destaque.

A mais significativa é uma correlação moderada (0.49) entre a variável *match_round* e *total_matches_difference*, indicando que jogos em fases mais avançadas tendem a envolver jogadores com o mesmo nível de experiência e, consequentemente, com menor diferença no número total de partidas enquanto nas fases mais precoces acontece o contrário.

Outra correlação, embora menos modesta, também merece destaque, é a relação entre *match_round* e *tournament_prize* (0.40), sugerindo que rondas mais avançadas estão relacionadas com prémios monetários mais elevados, o que é coerente com a estrutura típica dos torneios.

Features a usar nos modelos

Tendo em conta todas as análises realizadas, as variáveis que decidimos incluir no modelo são:

Feature	Descrição	Hipótese a considerar
height_difference	Diferença de alturas entre os jogadores	Diferenças significativas de altura podem dar vantagem física ao jogador mais alto (ex.: alcance e serviço), reduzindo potencialmente o número de sets.
rank_difference	Diferença de <i>ranks</i> entre os jogadores	Quanto maior a diferença entre os <i>rankings</i> , maior o desequilíbrio teórico, o que pode resultar em jogos com menos sets disputados.
age_difference	Diferença de idades entre os jogadores	Diferenças de idade refletem contraste entre juventude e experiência, podendo influenciar a duração do jogo.
total_matches_difference	Diferença do número total de jogos entre os jogadores	Um jogador com mais jogos em princípio tem mais experiência. Quanto maior for a diferença de jogos, menos renhido seria o jogo.
forehand_difference	Se os jogadores usam a mesma mão no jogo	Diferenças na lateralidade entre os jogadores (destro vs canhoto) influenciam o equilíbrio competitivo da partida, aumentando a probabilidade de a disputa estender-se a três sets.
match_round	Etapa do torneio	Rondas mais avançadas tendem a ser mais equilibradas, podendo resultar em jogos mais longos e com mais sets.
home_game	Se o vencedor é norte-americano	Jogadores que competem em casa podem ser beneficiados pelo apoio local e ambientes familiares, o que pode resultar em jogos com menor número de sets.
avg_sets_difference	Diferença entre o número médio de sets dos últimos 10	Um valor próximo de zero indica desempenhos recentes semelhantes entre os jogadores, sugerindo um

	jogos de cada jogador	maior equilíbrio nos jogos, enquanto valores elevados refletem diferenças nos históricos, podendo originar jogos menos equilibrados e com menor probabilidade de irem a três sets.
tournament_prize	Prémio monetário do torneio	Prémios mais elevados aumentam a motivação dos jogadores, levando a jogos mais renhidos e com maior número de sets.
ratio_difference	Diferença entre o rácio de vitórias/derrotas dos jogadores	Quanto mais semelhantes forem os rácios (menor <i>ratio_difference</i>), mais equilibrado se espera que seja o jogo, aumentando a probabilidade de três sets.

Tab 3 – Features a utilizar nos modelos

Nota: Para melhorar as *performances* dos modelos, as variáveis numéricas foram estandardizadas usando o *StandardScaler*, uma função da biblioteca *scikit-learn* em *Python*. O processo envolve subtrair a média de cada valor da variável e dividir pelo desvio padrão, de modo a garantir que todas as variáveis numéricas tenham a mesma escala e mesmo peso na construção dos modelos. Simultaneamente, as variáveis categóricas foram transformadas em variáveis binárias através da codificação *one-hot encoding*. Esta técnica cria $n - 1$ variáveis *dummy*, onde n é o número de categorias na variável original, permitindo que os modelos possam interpretar corretamente este tipo de informação.

Construção da amostra treino-teste

Para a construção e avaliação dos modelos preditivos, foi necessária a separação do *dataset* em dois subconjuntos: conjunto de treino (utilizado para ajustar os modelos) e conjunto de teste (usado para avaliar a performance dos mesmos). Optou-se por uma divisão de 80% para treino e 20% para teste.

Contudo, um dos desafios encontrados prendeu-se com a distribuição desproporcional da variável-alvo, *number_of_sets*. Na base de dados original, a maioria dos jogos eram decididos em 2 sets, sendo os jogos com 3 sets significativamente menos frequentes. Esta discrepância poderia levar a um enviesamento nos modelos, favorecendo a classe mais representada (jogos com 2 sets) e prejudicando a capacidade de prever corretamente jogos mais disputados (3 sets).

Para contornar este problema, foi aplicada uma técnica de balanceamento da base de dados antes da divisão treino-teste, igualando artificialmente o número de observações das duas classes (jogos com 2 sets e com 3 sets). Esta estratégia baseou-se em

undersampling, ou seja, reduziu-se intencionalmente a quantidade de observações da classe mais representada (2 sets) para igualar a da classe minoritária (3 sets). O objetivo foi criar uma amostra estratificada e equilibrada, com 50% de jogos com 2 sets e 50% com 3 sets, assegurando que os modelos tivessem igual exposição a ambas as durações das partidas e, consequentemente, uma maior robustez nas suas previsões. Assim, após o balanceamento, a amostra de treino passou a ser composta por 36584 observações e a amostra de teste por 9146 observações.

Além da avaliação no conjunto de teste, foi também implementada a técnica de validação cruzada com 10 *folds*, de forma a obter uma estimativa mais robusta e generalizável da performance dos modelos. Este processo consiste em dividir o conjunto de treino em 10 subconjuntos (*folds*) de igual dimensão: em cada iteração, o modelo é treinado com 9 desses subconjuntos e testado no subconjunto restante, repetindo-se o processo 10 vezes para garantir que cada *fold* é usado como teste exatamente uma vez. A média dos resultados obtidos nas diferentes iterações fornece uma medida mais fiável da capacidade preditiva do modelo, minimizando o risco de *overfitting* e variabilidade associada a uma única divisão treino-teste.

Modelos

Para avaliar o desempenho do modelo de regressão logística na previsão do número de sets de uma partida, fizeram-se algumas análises métricas, tal como a avaliação pela *precision*, *recall* e pela *accuracy*.

- $precision = \frac{TP}{TP + FP}$, indica a proporção de observações corretamente classificadas das que são previstas como positivas.
- $recall = \frac{TP}{TP + FN}$, indica a proporção de observações corretamente classificadas dos casos positivos.
- $accuracy = \frac{TP + TN}{TP + FP + TN + FN}$, indica a proporção total de classificações corretas, independentemente da classe.

A utilidade dos modelos pode também ser compreendida através da análise das curvas ROC, que representam graficamente a taxa de verdadeiros positivos (sensibilidade/*recall*) em função da taxa de falsos positivos, para diferentes limiares de decisão (*threshold*).

Os limiares de decisão referem-se aos valores que determinam o ponto a partir do qual uma observação é atribuída a uma determinada classe. Por exemplo, num modelo de classificação binária, se a probabilidade predita para uma observação pertencer à classe positiva for superior ao limiar (tipicamente 0.5), ela será classificada como tal. Ao variar

esse limiar, obtêm-se diferentes combinações de sensibilidade e especificidade, o que permite traçar a curva ROC.

Optou-se por gerar uma curva ROC para cada classe, utilizando a abordagem *one-vs-rest*. Isso significa que, em cada curva, uma das classes é tratada como positiva e a outra como negativa, permitindo avaliar a capacidade do modelo em distinguir especificamente cada classe do restante.

Essa estratégia é útil mesmo em contextos binários, pois oferece uma análise simétrica do desempenho: uma curva ROC foca em como o modelo identifica corretamente os jogos que terminaram em 2 sets (tratando-os como positivos), enquanto a outra analisa os que terminaram em 3 sets como positivos. Isso ajuda a perceber se o modelo está mais sensível a uma classe do que a outra, o que é relevante neste caso.

O valor da área sob a curva (AUC) quantifica esta performance: um valor de 0.5 indica uma classificação equivalente a uma aleatória, enquanto um valor de 1.0 representa uma classificação perfeita.

Regressão Logística

Como primeiro modelo usámos a Regressão Logística. Esta técnica é especialmente utilizada para prever probabilidades de um evento binário, como o nosso caso, no qual o modelo deverá prever jogos com 2 ou 3 sets.

Abaixo apresentam-se os resultados do modelo, tanto na avaliação original (conjunto de teste) como após a validação cruzada com 10 folds, permitindo uma análise mais robusta da sua *performance*:

	Original			10-Folds		
	Precision	Recall	F1-score	Precision	Recall	F1-score
2 sets	0.53	0.48	0.50	0.53	0.48	0.50
3 sets	0.52	0.57	0.55	0.52	0.57	0.55
Accuracy	0.53			0.53		

Tab. 4 - Métricas da Regressão Logística

No modelo original (sem validação cruzada), os resultados mostram uma precisão de 53% para 2 sets e 52% para 3 sets. Isto significa que, das partidas classificadas como 2 (3) sets, cerca de 53% (52%) foram corretamente identificadas. A *recall* apresenta uma ligeira variação: 48% para 2 sets e 57% para 3 sets, sugerindo que o modelo tem uma capacidade ligeiramente superior de identificar corretamente os jogos com 3 sets. O *F1-*

score, que representa o equilíbrio entre precisão e *recall*, atinge 0.50 e 0.55 para as classes 2 sets e 3 sets, respectivamente.

Com a validação cruzada (10-Folds), os resultados obtidos foram exatamente os mesmos.

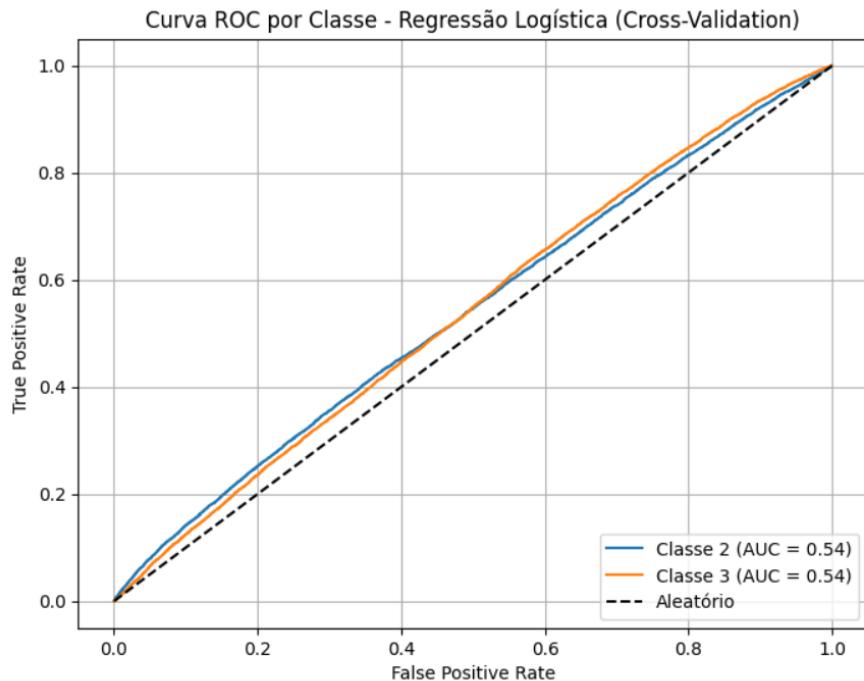


Fig. 17 – Curvas ROC (Regressão Logística)

A Regressão Logística apresenta uma AUC de 0.54 para ambas as classes (2 sets e 3 sets), indicando um desempenho ligeiramente superior ao de uma classificação aleatória (AUC = 0.50). A curva mostra uma separação mínima entre classes, o que revela uma capacidade limitada do modelo em distinguir eficazmente os dois resultados.

No geral, a Regressão Logística apresentou um desempenho equilibrado entre ambas as classes, sem mostrar qualquer enviesamento. Ainda assim, os resultados modestos — com uma acurácia média de 53% — indicam que, apesar de útil como ponto de partida, este modelo pode ser superado por algoritmos mais sofisticados

KNN

Decidiu-se também utilizar o algoritmo *KNN* (*K-Nearest Neighbors*). Este algoritmo baseia-se nos vizinhos mais próximos de cada observação e com base na sua similaridade efetua a classificação. Foram utilizados os 5 vizinhos mais próximos.

	Original			10-Folds		
	Precision	Recall	F1-score	Precision	Recall	F1-score
2 sets	0.50	0.51	0.50	0.53	0.48	0.50
3 sets	0.50	0.50	0.50	0.53	0.57	0.55
Accuracy	0.50			0.51		

Tab. 5 - Métricas do KNN

No modelo original, a *precision* tanto para a classe *2 sets* como para a *3 sets* situa-se nos 50%, ou seja, metade das previsões classificadas como jogos com 2 ou 3 *sets* correspondem efetivamente a essa classe. Portanto, existe uma certa dificuldade do modelo em distinguir as duas classes com elevada confiança.

A *recall* — que indica a proporção de acertos entre todas as ocorrências reais de cada classe — é igualmente equilibrada: 51% para *2 sets* e 50% para *3 sets*, sugerindo que o modelo consegue recuperar ambos os tipos de jogos em proporções semelhantes, embora com espaço para melhorias.

Após a aplicação da validação cruzada com 10 *Folds*, verifica-se uma leve melhoria geral. A precisão para ambas as classes sobe para 53%, e a *recall* para *2 sets* desce para 48%, enquanto para os *3 sets* sobe para 57%. O *F1-score* para a classe *3 sets*, sofre um aumento de 0.05.

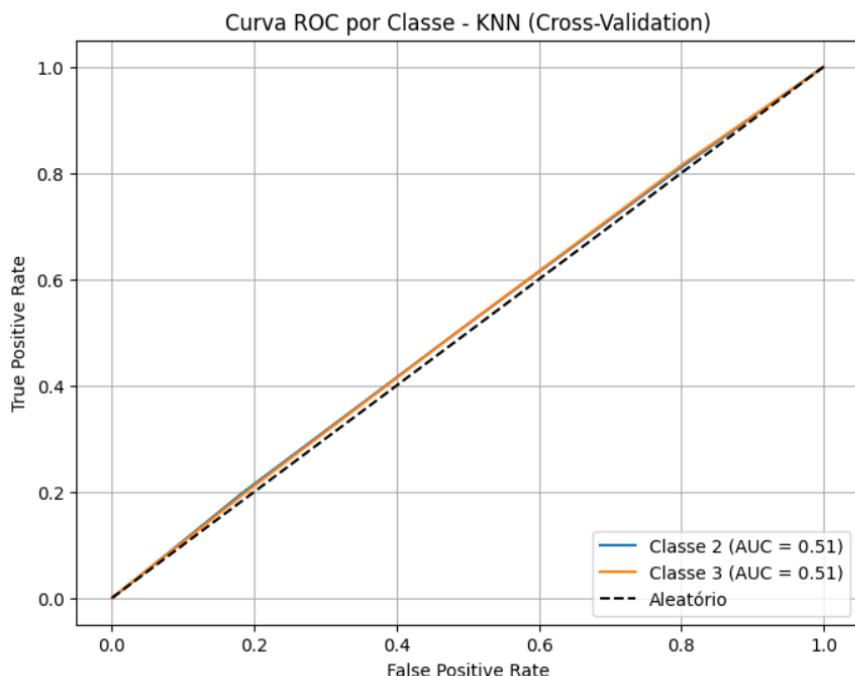


Fig. 18 – Curvas ROC (KNN)

O modelo KNN apresenta um AUC de 0.51 para ambas as classes, praticamente sobrepondo-se à linha da classificação aleatória. Isto sugere que o modelo tem pouca

capacidade discriminativa entre jogos com 2 e 3 *sets*, funcionando quase como um classificador aleatório neste contexto.

No geral, o modelo KNN mostra-se razoavelmente equilibrado entre as duas classes, sem um enviesamento acentuado para alguma delas. Ainda assim, a sua performance permanece modesta, sendo útil sobretudo como uma referência base no processo de modelação.

Árvore de Decisão

Foi também testado o modelo de Árvore de Decisão que segmenta os dados com base em regras de decisão extraídas dos preditores usados. O modelo constrói uma estrutura em forma de árvore, onde cada divisão (nó) representa uma decisão baseada numa variável.

Neste caso, foi definida uma profundidade máxima de 10, de forma a limitar a complexidade do modelo e evitar o *overfitting*. Esta restrição garante maior capacidade de generalização e torna a árvore mais interpretável, facilitando a identificação das variáveis mais relevantes para a previsão do número de *sets*.

	Original			10-Folds		
	Precision	Recall	F1-score	Precision	Recall	F1-score
2 sets	0.53	0.36	0.43	0.53	0.42	0.47
3 sets	0.51	0.67	0.58	0.52	0.62	0.57
Accuracy	0.52			0.52		

Tab. 6 - Métricas da Árvore de Decisão

No modelo original, a precisão foi de 0.53 para jogos com 2 *sets* e 0.51 para 3 *sets*, valores bastante equilibrados. No entanto, a *recall* apresentou maior disparidade: o modelo identificou corretamente apenas 36% das partidas com 2 *sets*, enquanto conseguiu identificar 67% das partidas com 3 *sets*. Esta diferença sugere que a árvore de decisão se revelou mais eficaz a detetar jogos mais longos.

Ao aplicar a validação cruzada (10-Folds), os resultados mantiveram-se estáveis, com pequenas variações: o *recall* para 2 sets aumentou para 42%, enquanto diminui para 62% em jogos com 3 sets. O *F1-score* desceu ligeiramente para 0.57 para 3 *sets*, e subiu para 0.47 nos 2 *sets*.

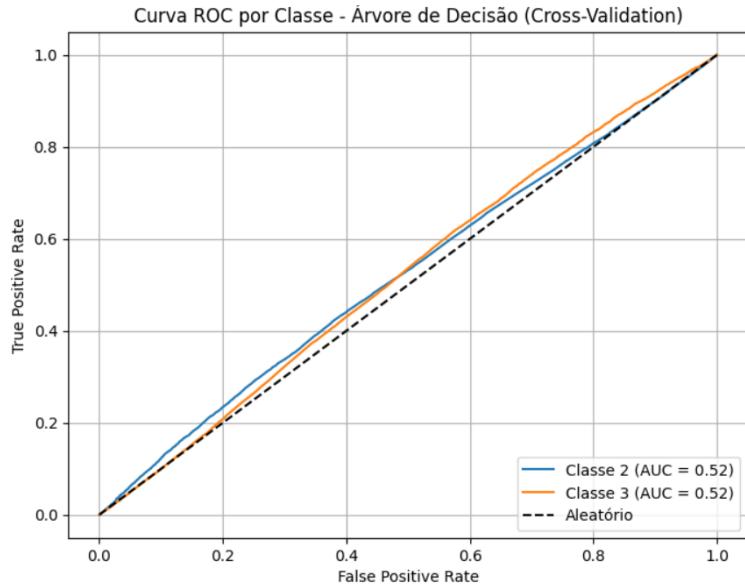


Fig. 19 – Curvas ROC (Árvore de Decisão)

A árvore de decisão demonstra uma ligeira melhoria face ao KNN, com um AUC de 0.52 para ambas as classes. Apesar de ainda estar próxima do limiar aleatório, a curva mostra uma leve separação entre classes, sugerindo uma capacidade discriminativa um pouco mais robusta, embora ainda modesta.

Em resumo, a Árvore de Decisão apresentou um desempenho mais favorável para a classe de 3 sets, mostrando uma maior sensibilidade para identificar jogos mais longos. Ainda assim, a acurácia global situou-se em 52%, sugerindo que o modelo, embora interessante, pode beneficiar de ajustes ou da combinação com outros algoritmos para melhorar o desempenho preditivo.

Random Forest

Adicionalmente, optou-se por utilizar o modelo *Random Forest*, uma técnica que combina diversas árvores de decisão para melhorar a precisão. Cada árvore é treinada com uma amostra aleatória dos dados, o que introduz diversidade no modelo e reduz o risco de *overfitting*. A decisão final é obtida através da votação da maioria entre as árvores, resultando numa classificação mais fiável.

	Original			10-Folds		
	Precision	Recall	F1-score	Precision	Recall	F1-score
2 sets	0.53	0.46	0.49	0.53	0.46	0.49
3 sets	0.52	0.59	0.55	0.52	0.60	0.56
Accuracy		0.53			0.53	

Tab. 7 - Métricas da Random Forest

Nos resultados do modelo original, observa-se uma precisão de 0.53 para jogos com 2 sets e 0.52 para 3 sets, com *recalls* de 0.46 e 0.59, respectivamente. Estes valores indicam que o modelo é ligeiramente mais eficaz a detetar jogos com 3 sets, embora com desempenho modesto para jogos mais curtos.

Com a aplicação da validação cruzada (*10-Folds*), não houve grandes alterações nos resultados: a precisão manteve-se igual para ambas as classes, o *recall* para 2 sets também manteve o mesmo valor enquanto para 3 sets baixou 0.01. Os valores do *F1-score* refletem esta estabilidade, situando-se nos 0.49 e 0.56, respectivamente.

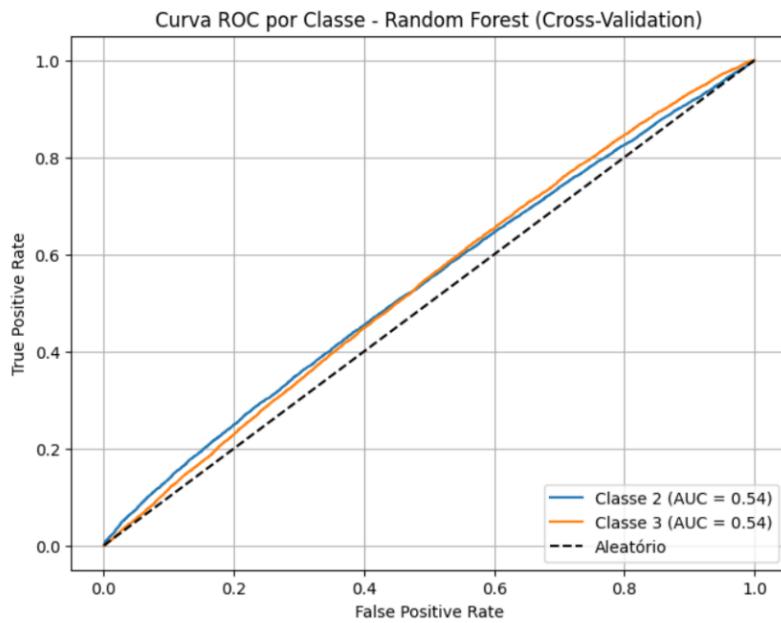


Fig. 20 – Curvas ROC (Random Forest)

O modelo *Random Forest*, tal como a Regressão Logística, apresenta um AUC de 0.54 para ambas as classes. Embora o valor continue baixo, está acima da linha aleatória, indicando que o *ensemble* de árvores consegue capturar padrões ligeiramente mais informativos nos dados, comparativamente aos outros modelos.

Em síntese, o *Random Forest* revelou-se um modelo mais robusto que a árvore de decisão isolada, mantendo uma acurácia global de 53% e demonstrando maior consistência na classificação de jogos com 3 sets, o que pode ser útil para capturar partidas mais competitivas.

Avaliação Final

Com base na aplicação dos quatro modelos - Regressão Logística, *K-Nearest Neighbors (KNN)*, Árvore de Decisão e *Random Forest* – foi possível realizar uma análise comparativa das suas capacidades preditivas relativamente ao número de sets de uma partida (2 ou 3 sets).

A regressão logística revelou um desempenho equilibrado entre as duas classes, com valores de *precision* e *recall* próximos de 52 e 53%, sem enviesamento em qualquer uma das categorias. Todavia, a sua natureza linear pode ter limitado a capacidade de identificar padrões mais complexos nos dados disponíveis. A realização de *cross-validation* também não trouxe melhorias significativas para as previsões, reforçando a ideia de que, o modelo, embora estável, não é suficientemente expressivo para capturar a complexidade do problema presente.

O modelo *KNN* teve um desempenho ainda inferior ao da Regressão Logística, com métricas globalmente mais baixas e uma maior dificuldade em distinguir as duas classes. Apesar de a *cross-validation* ter mostrado uma ligeira melhoria na sensibilidade para os jogos de 3 sets (57%), a performance global permaneceu aquém do esperado.

A árvore de decisão revelou um enviesamento notório na classificação, demonstrando um *recall* significativamente elevado para os jogos de 3 sets (67%) em comparação com os de 2 sets (36%). Este comportamento pode sugerir que o modelo pode ter capturado padrões específicos de uma classe em detrimento da outra, levando a um comportamento de desequilíbrio.

Por sua vez, a *Random Forest* foi o modelo que apresentou uma maior robustez e consistência entre todos. Mas ainda assim, as melhorias foram marginais, e a *accuracy* permaneceu nos mesmos níveis de 53%. Embora tenha sido o modelo mais estável, os resultados no geral indicaram que nenhum dos modelos consegue prever com exatidão o número de sets com base nas variáveis disponíveis.

Em termos globais, todos os modelos apresentaram uma *performance* fraca, com valores de *accuracy* compreendidos entre os 50% e 53%, aproximando-se assim, do limiar de um modelo que faz previsões aleatórias. Estes resultados mostraram que os modelos testados não foram capazes de identificar padrões significativos nos dados que lhes permitissem realizar previsões fiáveis.

Implementação

A fase de implementação tem como objetivo traduzir os resultados obtidos em ações concretas, permitindo que o conhecimento gerado ao longo do projeto possa ser aplicado de forma prática.

Ao longo deste projeto, foi construída uma base de dados robusta e detalhada, a partir de registos de jogos de ténis. Para além das variáveis originalmente disponíveis, foram criadas novas variáveis, com o objetivo de enriquecer a análise. Entre estas destacam-se, por exemplo, as diferenças entre os jogadores em termos de *ranking*, idade, altura, lateralidade e rácios de vitórias, permitindo uma caracterização mais aprofundada de cada confronto. O *dataset* encontra-se limpo, estruturado, documentado e pronto para ser reutilizado por terceiros em projetos futuros, sejam eles de análise estatística, modelação preditiva ou apenas visualização.

Foram também testados vários modelos de classificação binária (Regressão Logística, KNN, Árvore de Decisão, *Random Forest*), com a finalidade de prever se um jogo seria resolvido em dois ou três *sets*. Para assegurar uma avaliação fiável, os modelos foram treinados numa amostra balanceada (recorrendo ao *undersampling*) e avaliados com recurso à validação cruzada (10 *folds*), medindo-se o desempenho através de métricas como a acurácia, precisão, *recall*, F1-score e AUC.

Apesar de todos os esforços aplicados no tratamento dos dados, *feature engineering* e escolha de algoritmos, os modelos desenvolvidos apresentaram resultados moderados, com níveis de acurácia que limitam a sua eficácia em cenários reais. A natureza do problema e a complexidade intrínseca do jogo de ténis dificultam o desenvolvimento de previsões precisas, especialmente quando se tenta antecipar jogos decididos por margens reduzidas.

No entanto, mesmo com limitações, os modelos já apresentam alguma utilidade como um apoio para certos contextos específicos. Um exemplo é a sua integração em sistemas de apoio à decisão em casas de apostas, especificamente na estimativa da probabilidade de um jogo terminar em dois ou três *sets*. Esta informação pode ser útil na contribuição para definições de *odds* mais ajustadas, contribuindo para uma maior competitividade no mercado de apostas. Além disso, a previsão de número de *sets* de um jogo pode revelar-se particularmente útil na gestão dos torneios, partindo da estimativa da duração dos jogos, a fim de facilitar o planeamento e a organização dos encontros.

Assim, embora a base de dados represente um ativo valioso e reutilizável, a aplicação prática direta dos modelos preditivos desenvolvidos deve ser encarada com cautela. No estado atual, estes modelos podem servir como ponto de partida para investigações futuras ou aplicações exploratórias, mas requerem melhorias adicionais — como a

introdução de dados mais detalhados sobre os jogos que permitam capturar de forma mais eficaz os fatores que realmente influenciam o número de *sets* de cada jogo de ténis.

Exemplos disso incluem variáveis como a taxa de serviços bem-sucedidos de cada jogador, histórico de confrontos diretos entre os adversários ou até fatores externos como condições climatéricas no momento do jogo. A inclusão deste tipo de informação poderá aumentar substancialmente a capacidade explicativa dos modelos.

Conclusão

A abordagem estruturada da metodologia *CRISP-DM* revelou-se essencial para a condução deste projeto, desde a compreensão do problema de negócio, preparação e enriquecimento da base de dados, até à construção e avaliação dos modelos preditivos.

Apesar do rigor metodológico e da aplicação de diferentes técnicas preditivas, as análises dos resultados obtidos revelaram que os modelos desenvolvidos não atingiram as expectativas iniciais do grupo em termos do desempenho preditivo. Esta constatação, longe de representar um insucesso, constituiu como um ponto de aprendizagem valioso.

Entre os fatores que podem ter condicionado o desempenho dos modelos, destaca-se a qualidade e complexidade dos dados disponíveis. A presença de ruídos e dados em falta dificultaram a extração de padrões consistentes. Além disso, o próprio desporto em análise, o ténis, sujeito a vários fatores imprevisíveis tais como lesões e condições climáticas constituem um desafio acrescido à previsão consistente dos resultados.

Apesar de tudo, este trabalho revelou-se extremamente enriquecedor do ponto de vista académico. Foi fundamental para o desenvolvimento de competências práticas de como lidar com os dados reais, incluindo tratamento de dados não estruturados, aplicação de técnicas de *machine learning* e avaliação crítica dos modelos desenvolvidos.

Em suma, este trabalho não só permitiu consolidar os conhecimentos teóricos adquiridos ao longo da Licenciatura, como também reforçou a importância de experimentar e de adaptar aos desafios impostos pela realidade dos dados.