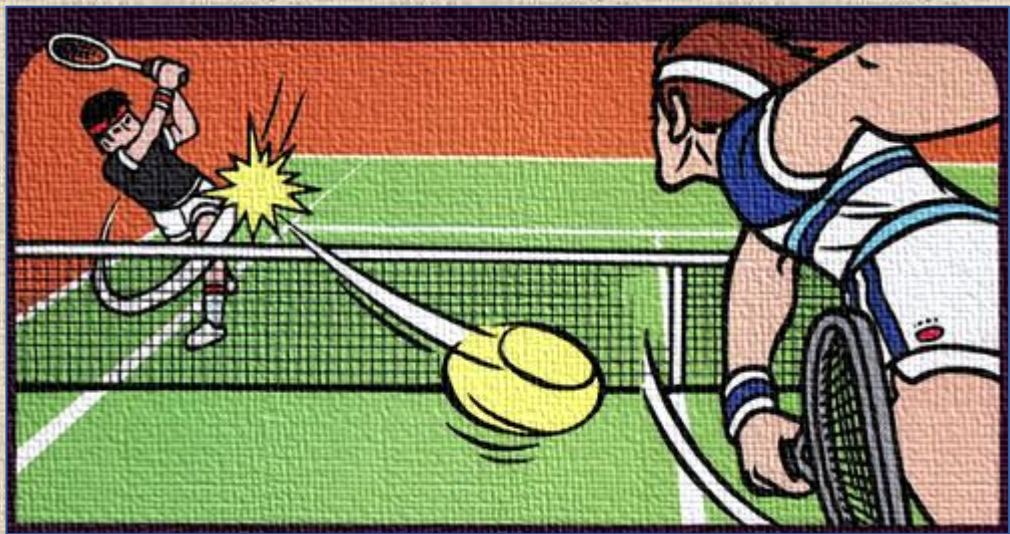

PROJETO APLICADO A CIÊNCIA DOS DADOS I



Ténis na Bélgica

PREVER O NÚMERO DE SETS

PACD1

TRABALHO REALIZADO POR:

- ALLAN KARDEC, Nº 103380, CDB1
- DIOGO FREITAS, Nº 104841, CDB1
- JOÃO FRANCISCO BOTAS, Nº 104782, CDB1
- RICARDO ÂNGELO, Nº 104826, CDB1



Índice

| | |
|--|----|
| Introdução | 2 |
| CRISP-DM | 3 |
| Business Understanding..... | 4 |
| Data Understanding | 6 |
| Tratamento de partidas espelhadas | 7 |
| Dados omissos | 8 |
| Variáveis quantitativas..... | 9 |
| Variáveis qualitativas..... | 11 |
| Data Preparation | 14 |
| Dados dos oponentes | 14 |
| WebScrapping..... | 14 |
| Tratamento de variáveis..... | 17 |
| Features a utilizar na fase de modelação..... | 22 |
| Modeling & Evaluation | 23 |
| Correlação e associação das variáveis preditoras entre si..... | 24 |
| Partição dos dados e alguns modelos básicos | 26 |
| Variável “GroundNumerico” e impacto da mesma | 27 |
| Gradient Boosting e hiperparâmetros | 28 |
| Alteração dos Hiperparâmetros..... | 29 |
| Deployment..... | 32 |
| Referências | 35 |
| Anexos | 36 |



Introdução

Neste trabalho tentar-se-á prever o número de *sets* por partida de ténis à melhor de três. Em específico, trabalhar-se-á com partidas relativas à Bélgica. Neste trabalho explicar-se-á em que medida tal previsão poderá ser útil, abordando áreas que poderão ser beneficiadas com tal previsão. Para realizar tais previsões, limpar-se-ão os dados originais de forma que propulsionem de forma positiva a capacidade preditiva dos modelos escolhidos. Para tal, abordaremos algumas técnicas que visam colmatar dificuldades e barreiras iniciais, como dados omissos e dados rudimentares. Criar-se-ão variáveis que terão por base não só os dados originais, como também os dados obtidos através de *WebScrapping*. Após isso ter-se-ão em conta as variáveis mais pertinentes para a previsão e explicar-se-ão as razões para tal, de forma a não só dar a entender a importância dos dados em si, como também dar a perceber quais as características mais importantes numa partida de ténis, em específico, quantos *sets* a mesma terá, tendo em conta os dados dos participantes bem como dados sobre as condições da partida. No final, serão comentados os resultados, com a devida interpretação destes, onde será realizada uma possível aplicação destes valores no quotidiano.



CRISP-DM

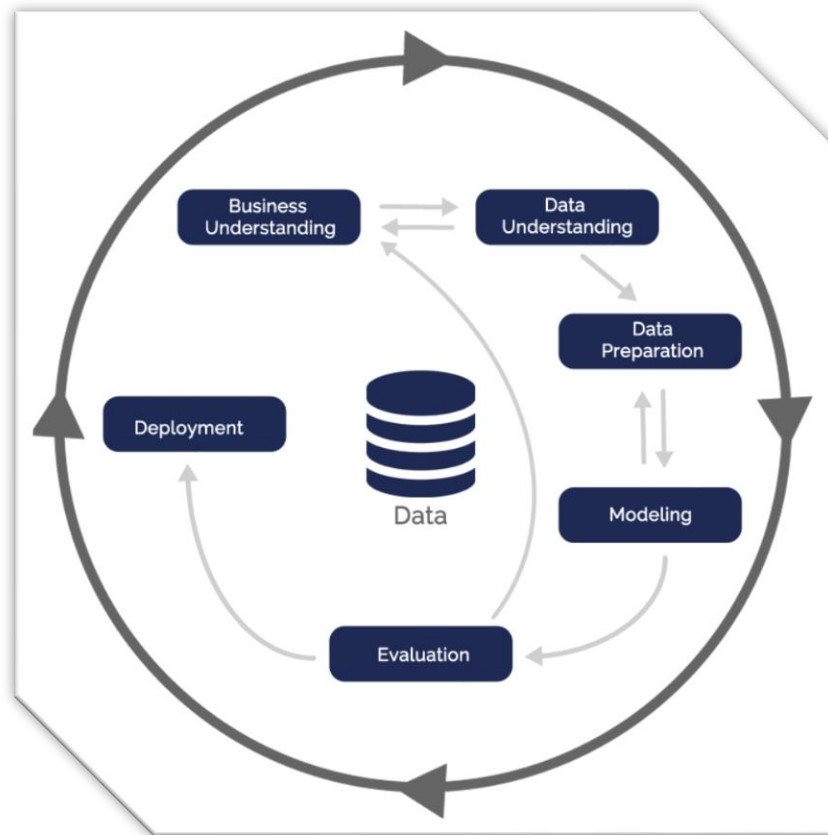


Figura 1 - CRISP-DM

- *Business Understanding* – Consiste em conhecer e perceber o conteúdo dos objetivos do trabalho. Conhecendo – e percebendo – o conteúdo dos objetivos do trabalho torna-se mais fácil perceber os dados que se irá utilizar bem como os resultados derivados do estudo dos mesmos;
- *Data Understanding* – Consiste em fazer análises – maioritariamente simples – aos dados disponíveis. Tais análises permitem evitar problemas inesperados na fase seguinte - *Data Preparation*;
- *Data Preparation* – Esta é a fase onde, regra geral, se despende mais tempo. Nesta fase: criam-se subamostras; juntam-se dados; criam-se novas variáveis; escolhem-se as variáveis que serão usadas para previsão; removem-se e/ou substituem-se valores omissos; criam-se conjuntos de treino e de teste para facilitar as fases seguintes;
- *Modeling* – Nesta fase usam-se os dados previamente preparados em um ou mais modelos com base em *machine learning*. É comum voltar atrás e preparar os dados de diferentes formas. Tal é feito com o objetivo de ter os dados o mais bem preparados para os modelos em que serão utilizados;
- *Evaluation* – Nesta fase avalia-se a capacidade dos modelos previamente desenvolvidos;
- *Deployment* – Nesta fase tiram-se conclusões sobre os dados resultantes do *Modeling* e vê-se a utilidade do modelo.



Business Understanding

Nesta etapa inicial, é importante possuir, com alguma clareza, um rumo e um objetivo em relação ao problema que se pretende resolver. Para isso, neste caso, é necessário saber os básicos de como o ténis funciona, dando uma maior preferência às características do mesmo em relação ao país em estudo – Bélgica.

Como funciona uma partida de ténis?

Antes de mais, é necessário frisar que os torneios de ténis têm várias características que influenciam a mudança do *ranking* ao longo do ano. Alguns destes fatores, que influenciam o *ranking*, são: o tipo de solo, se o jogo é jogado *outdoor* ou *indoor*, as condições climatéricas, entre outros. Logo, os resultados dos *sets* e das partidas podem ser determinados nas condições presentes, existindo jogadores melhores em certas categorias do que outros.

O foco principal neste projeto é a previsão do número de *sets* por partida. O vencedor de um *set* é determinado a partir do primeiro jogador a chegar a 6 jogos. Em caso de empate, o resultado do *set* é definido através do *tie-break*¹. O número de jogos ganhos é sempre transmitido da forma: jogador-oponente, ou seja, se o oponente tiver vencido o *set* com 6 jogos contra 3 do jogador, diz-se que o *set* ficou 3-6. O vencedor da partida é definido à melhor de 3 *sets* ou, em alguns casos, à melhor de 5 *sets*.

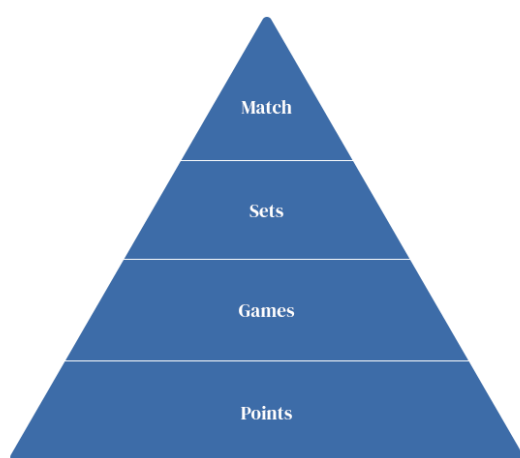


Figura 2 - Funcionamento de uma Partida de Ténis

Existe uma grande aposta no ténis por parte da Bélgica?

Bélgica não é extremamente conhecida por sediar os mais célebres torneios de ténis do ATP, porém, existem algumas edições ATP realizadas neste país, tais como o *European Open Antwerp* ou o Torneio *Ethias*, em *Mons*. Estas competições descritas são jogadas *indoor* e em solo duro, tal como a maioria dos torneios ATP na Bélgica.

¹ Um *tie-break* no ténis é um jogo especial usado para quebrar um empate em um *set* (5-5). Os jogadores competem para alcançar sete pontos primeiro (ou liderar por dois pontos, se o placar estiver empatado em 6-6). O vencedor do *tie-break* ganha o *set*.



Em termos de jogadores, a Bélgica não possui muitos atletas no topo do *ranking* atual da ATP, no entanto, tem grandes jogadores presentes no *top* 100 e 200, como é o caso de David Goffin, Zizou Bergs, Kimmer Coppejans, entre outros. Os jogadores no *top* 200 e que são da Bélgica têm entre 20 e 30 torneios jogados, de acordo com o *site* da ATP.

De que forma o ténis influencia negócios e economia na Bélgica

Como foi dito anteriormente, neste trabalho o objetivo é prever o número de *sets* de uma partida de ténis, mas isso poderá ser útil para o quê? Após deliberação, chegou-se à conclusão de que a previsão do número de *sets* pode ser útil quando se tem em mente os seguintes temas:

- a) Consumo dos espectadores;
- b) Publicidade e *merchandising*;
- c) Apostas.

Consumo dos espectadores

Suponha-se que, antes de uma partida começar, tinha-se previsto que esta seria maior. Assim sendo, a probabilidade de um espectador consumir algo é maior do que numa partida mais pequena. Consequentemente, isso poderia influenciar as decisões sobre quando fazer promoções de produtos alimentares, ou de equipamentos desportivos, a fim de se conseguir o máximo de lucro possível em relação ao consumo dos espectadores da partida.

Publicidade e merchandising

Saber o número de *sets* por partida também pode ser vantajoso no que diz respeito a *merchandising* pois, regra geral, os espectadores preferem ver partidas mais longas devido à maior intensidade. É nestas partidas que os espectadores estão mais predispostos a comprar camisolas dos jogadores da partida em questão, como recordação. Tendo isso em conta, apostar em *merchandising* em partidas cujo número de *sets* previsto é elevado é mais vantajoso.

No que diz respeito a publicidade, se dada empresa quiser ter anúncios sobre si mesma, em dada partida de ténis, convém pagar para ser publicitada em partidas com maior número de *sets* pois, pelo mesmo preço, têm maior exposição e, consequentemente, dá melhor uso ao seu investimento. Também é de salientar que fazer publicidade sobre partidas que serão mais intensas dá um carácter mais competitivo e emocionante ao torneio.

Apostas

No que diz respeito a apostas, saber o número de *sets* de dada partida pode ser vantajoso, pois se se tem conhecimento prévio da performance de ambos os jogadores em relação ao número de *sets* por partida, apostar-se-á de forma mais fundamentada, o que poderá levar a maior taxa de acerto, i.e., a melhores apostas.



Data Understanding

Nesta fase, será importante entender o que é que cada variável significa, sendo necessário estudá-las e usar aquelas que, eventualmente, serão mais úteis para atingir o objetivo. É necessário ter em atenção que estas serão apenas tratadas e trabalhadas na fase seguinte. Num primeiro momento, viu-se a proporção de casos a analisar para o país atribuído, a Bélgica, em relação à base de dados inicial. Cerca de 1% dos jogos, 10996 linhas, correspondem a jogos que contêm “Belgium” na coluna da **Location**, sem qualquer tratamento. Uma percentagem muito baixa face à quantidade existente na base de dados original. Abaixo é possível visualizar a **Tabela 1** com uma breve descrição de cada uma das variáveis, para os dados da Bélgica.

Tabela 1 - Descrição das Variáveis

| | |
|-------------------|--|
| PlayerName | Nome do jogador. |
| Born | O local onde o jogador nasceu. |
| Height | A altura do jogador, em centímetros. |
| Hand | Mão dominante do jogador, bem como a sua <i>backhand</i> . |
| LinkPlayer | Um <i>link</i> que leva direto à página <i>web</i> do jogador, tendo todas as suas características bem como o seu histórico de jogos. |
| Tournament | O nome do torneio do qual a partida pertence. |
| Location | A localização do torneio (serão só torneios localizados em Bélgica). |
| Date | A data em que foi decorrido o torneio. Varia entre 1968 e 2021 nos nossos dados, referentes à Bélgica. |
| Ground | O material do chão do campo, podendo ser um dos seguintes: Clay, Hard, Carpet |
| Prize | O prémio que o jogador ganhou no torneio. |
| GameRound | A ronda que a partida representa no torneio. As rondas existentes nos torneios realizados em Bélgica são Round of 32, Round of 16, Quarter-Finals, Semi-Finals, Finals. |
| GameRank | O <i>Rank</i> atribuído à partida, tendo como análise o torneio, os jogadores, o prémio, a localização do torneio, e várias outras características. |
| Oponent | O nome do oponente do jogador. |
| WL | W se o jogador venceu partida ou L caso o jogador tenha perdido. |
| Score | Registo do resultado de cada <i>set</i> de partida, em que cada <i>set</i> é representado por dois números. O número da esquerda representa o total de jogos no <i>set</i> que o jogador venceu; o número da direita representa o total de jogos no <i>set</i> que o jogador perdeu. |

Em relação às características das variáveis, é de notar que apenas existem 3 variáveis quantitativas de origem (**Height**, **Prize** e **GameRank**). As restantes variáveis são todas qualitativas.



Tratamento de partidas espelhadas

Esta fase – *Data Understanding* – é uma fase de análise, onde é necessário conhecer o contexto das variáveis e o que os dados de cada uma significam. No entanto, nesta fase também se tratará de um problema específico presente na base de dados: a existência de partidas espelhadas. Por outras palavras, a existência da mesma partida, mas com pontos de vista diferentes, sendo um deles do ponto de vista do jogador (que pode ter vencido a partida), ou do ponto de vista do oponente (que pode ter perdido a partida). Este pormenor, caso ignorado, poderia causar conclusões imprecisas no que toca a gráficos, às características de cada uma das variáveis, de dados omissos e do objetivo pretendido desta fase.

Para solucionar isto foram criadas duas colunas, em que uma representa o nome de quem ganhou a partida e a outra o nome de quem perdeu a partida. Para a criação destas variáveis foi utilizada a informação da coluna **WL**, que indica se o jogador, que aparece no campo **PlayerName**, venceu ou perdeu a partida. Através desta informação foi possível descartar todas as partidas espelhadas da base de dados, mantendo todas as partidas, pertencentes ao conjunto que continha as partidas espelhadas e que tinham W como valor na coluna **WL**. Para criar o conjunto de partidas espelhadas, teve-se em conta todas as partidas que tinham dados iguais nas colunas **Tournament**, **Location**, **Date**, **Ground**, **Prize** e **GameRound**. Foram consideradas estas colunas, pois é impossível dois jogadores enfrentarem-se mais que uma vez nas condições acima referidas.

Após estas transformações passou-se de 10996 linhas para 5982 linhas no *dataset*. Tendo isto em conta, é de notar que grande parte das partidas estão repetidas, cerca de 46,6% do valor original de partidas. Somado a isto, é importante referir que existiam Oponentes intitulados de “Bye” e, por se tratar de casos em que a ronda não foi jogada (“passada a frente por questões de *seed* nos torneios”), optou-se por apagar estas linhas.

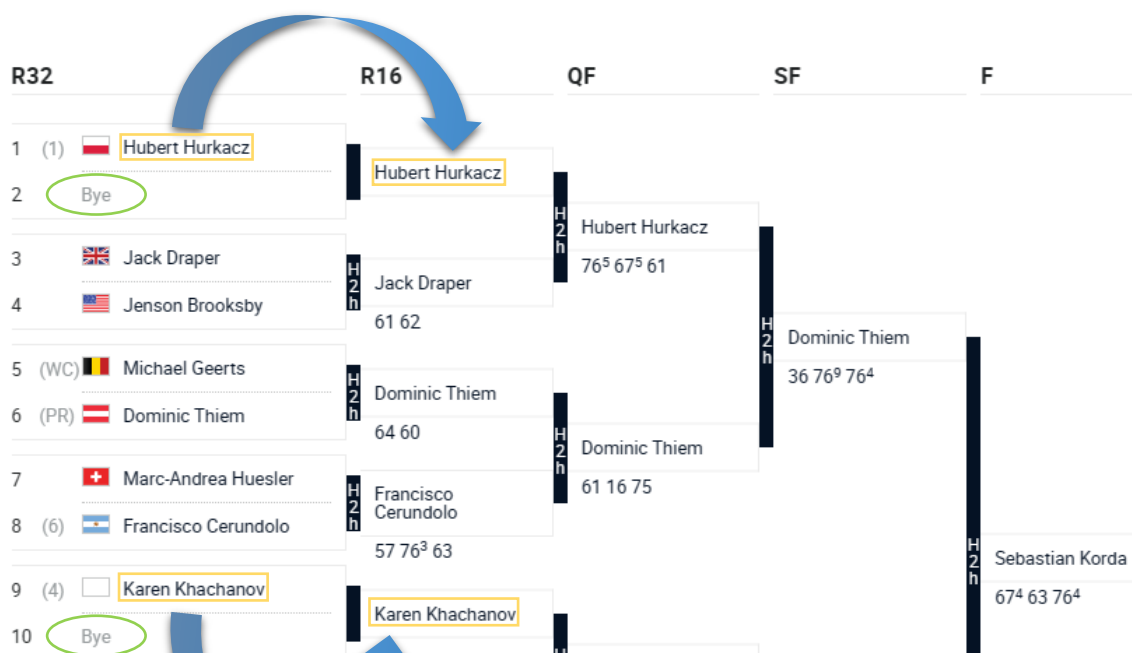


Figura 3 - Visualização do "bye"



Dados omissos

O gráfico abaixo mostra o total de valores omissos para cada variável.

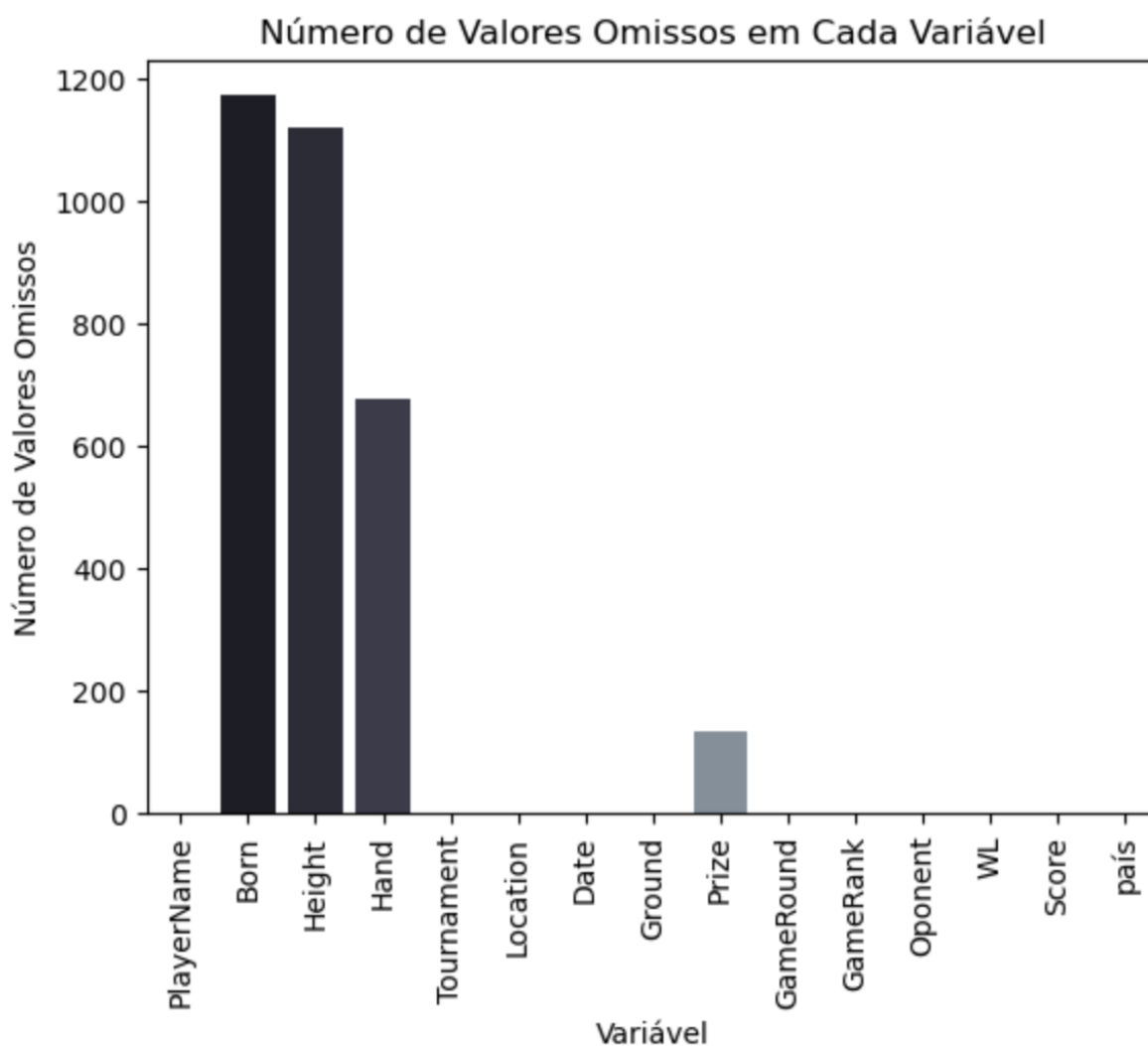


Figura 4 - Dados Omissos

Tendo em conta a informação acima, as colunas **Born**, **Height**, **Hand** e **Prize** têm uma quantidade notável de valores omissos. A cada coluna do gráfico com dados omissos está atribuída uma cor. Quanto mais escura, maior o número absoluto de dados omissos. Analisando com atenção, é perceptível que a coluna **Born** é a coluna que possui maior número de dados omissos, enquanto a coluna **Prize** - tendo em conta as colunas que possuem dados omissos - é a que possui menos dados em falta.

Não se sabe qual a razão específica para cada valor omisso, mas, para as colunas **Born**, **Height** e **Hand**, assume-se que é MAR (*missing at random*). Esses dados são, maioritariamente, relativos a jogadores que aparecem poucas vezes no *site* da ATP.

Sobre os dados omissos da coluna **Prize**, embora não sejam necessariamente todos, a maior parte são estruturalmente omissos, pois são relativos a partidas cujo prémio era \$0, i.e., não tinham prémio, e, portanto, não foi atribuído nenhum valor.



O gráfico abaixo mostra a respetiva percentagem desses dados omissos, por coluna.

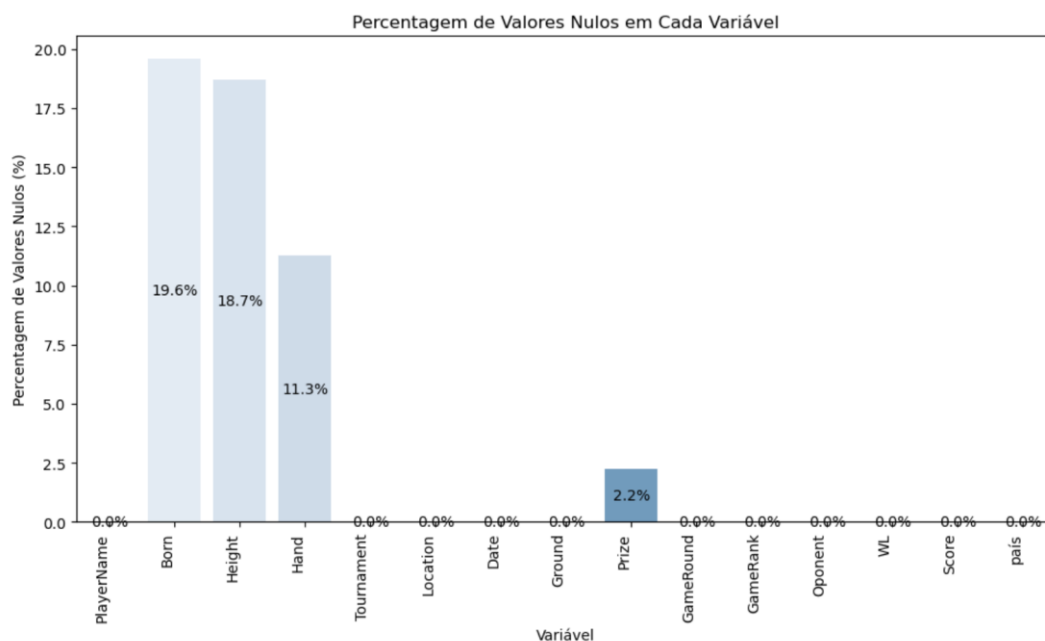


Figura 5 - Percentagem de valores omissos

Variáveis quantitativas


GameRank

O valor de **GameRank** para dado jogo tem em conta o torneio, bem como a sua fase, e, também, o *ranking* dos jogadores que dele fazem parte. Existem vários valores respetivos a esta coluna. Jogos diferentes podem ter o mesmo valor de **GameRank**.

Jogos com o valor de **GameRank** mais próximo de 0 estão associados a jogos de maior intensidade, quer seja pelo torneio que fazem parte, quer seja pela fase de tal torneio. Esta variável também tem em conta o *ranking* dos respetivos jogadores da partida. Quanto menor o valor do **GameRank**, melhor é o *rank* dos respetivos jogadores, contribuindo assim para uma partida muito mais intensa.

Prize

O valor de **Prize** representa o valor do prémio do torneio em dólares americanos ou em euros. Nem todos os jogos têm um prémio associado. Em relação a esta variável, será necessário limpá-la, retirando os caracteres indesejados e uniformizando todos os preços para uma moeda padrão, com a finalidade de tornar o tipo desta coluna em inteiro efetivamente - está como categórica devido aos caracteres. É importante referir que existe histórico de jogos que ocorreram antes do euro existir.

Após uma breve análise, reparou-se que as linhas que tinham esta variável com o caracter especial , correspondiam à moeda euro. Devido a isso, será necessário converter o valor do prémio para uma única



moeda. Mais à frente será decidida a moeda a usar, explicando o processo que se usou para respeitar a taxa de câmbio da forma mais correta possível.

Height

Para cada linha, o valor desta coluna corresponde à altura do jogador em centímetros. Esta coluna possui uma grande taxa de valores omissos, algo referido anteriormente. Pode-se verificar no *boxplot* abaixo como se encontra a dispersão dos dados pelos quantis.

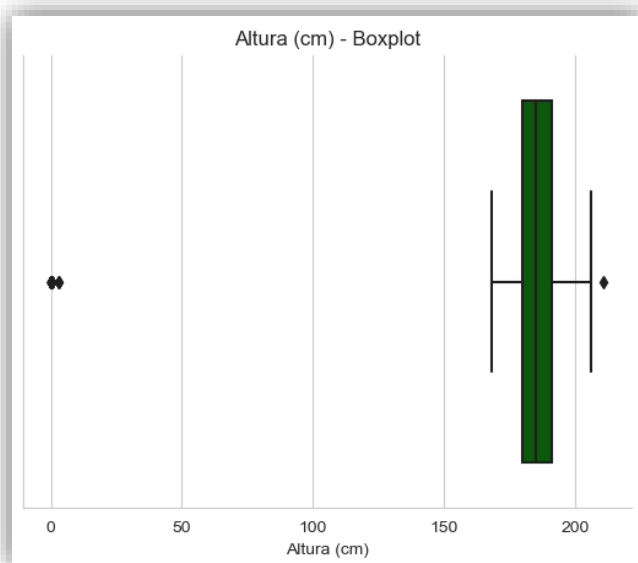


Figura 6 - Boxplot Altura

Observando atentamente para o *boxplot* da **Figura 6** é nítida a presença de *outliers* extremos inferiores nesta variável. Estudou-se estes pontos extremos e concluiu-se que estes correspondiam aos valores de 0, 3 e 71 cm de altura, que não correspondem a alturas de pessoas adultas profissionais de ténis, como é evidente. Estes valores podem ter surgido por erros de digitação ou ainda de desconhecimento da altura do jogador específico, no caso do valor 0.

Assim, ter-se-á cuidado na fase seguinte relativamente a esta *feature* e aos

valores que não correspondem a valores reais, mas sim a erros.

É possível visualizar na **Figura 7** um histograma com a distribuição das alturas e, analisando com atenção, é possível reparar, não só na existência de *outliers*, já comentados anteriormente, como também perceber que maior parte dos jogadores têm uma altura entre os 175cm e 200cm - ter em atenção que estes dados são as alturas dos jogadores que se encontram na coluna *PlayerName*, existindo a possibilidade de algumas se repetirem.

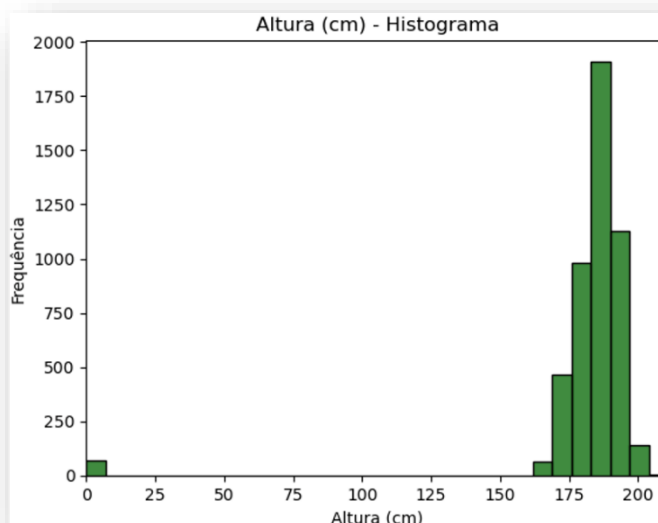


Figura 7 - Histograma Altura



Variáveis qualitativas

Ground

Para verificar o tipo de **Ground**, i.e., piso nos torneios belgas será realizado um gráfico circular, pois já se tinha visto anteriormente que esta variável tem poucas classes.

Observando o gráfico na **Figura 8**, consegue-se detetar que na Bélgica só existem 3 tipos de torneios diferentes, no que toca ao solo em que é jogado. Numa grande maioria - aproximadamente 72% - vê-se que os jogos são jogados em “Clay”, ou terra batida em português. As restantes 2 classes são de torneios jogados em solo duro ou tapete. Não há torneios realizados em relva na Bélgica, de acordo com a base de dados fornecida.

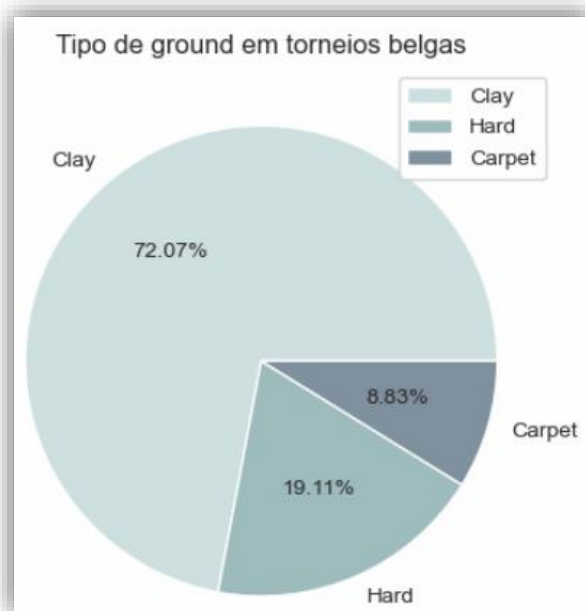


Figura 8 – Gráfico Circular com o tipo de chão

PlayerName

PlayerName é a variável que tem como valor o nome do jogador de dada partida. Antes de se realizar a limpeza desta variável, reparou-se que existem 1200 jogadores diferentes registados em jogos realizados na Bélgica.

Born

A variável **Born** tem como registo o local de nascimento do jogador, cujo nome está referido em **PlayerName**. Os registos desta variável podem variar de diferentes formas, podendo estar em vários formatos diferentes, referenciando o país e a cidade, ou só o país, etc...

Hand

A variável **Hand** guarda os dados sobre a mão dominante e sobre a *backhand* do jogador cujo nome está em **PlayerName**. O gráfico da **Figura 9** mostra as diferentes combinações de mão dominante e *backhand* registadas nos jogos realizados em Bélgica.

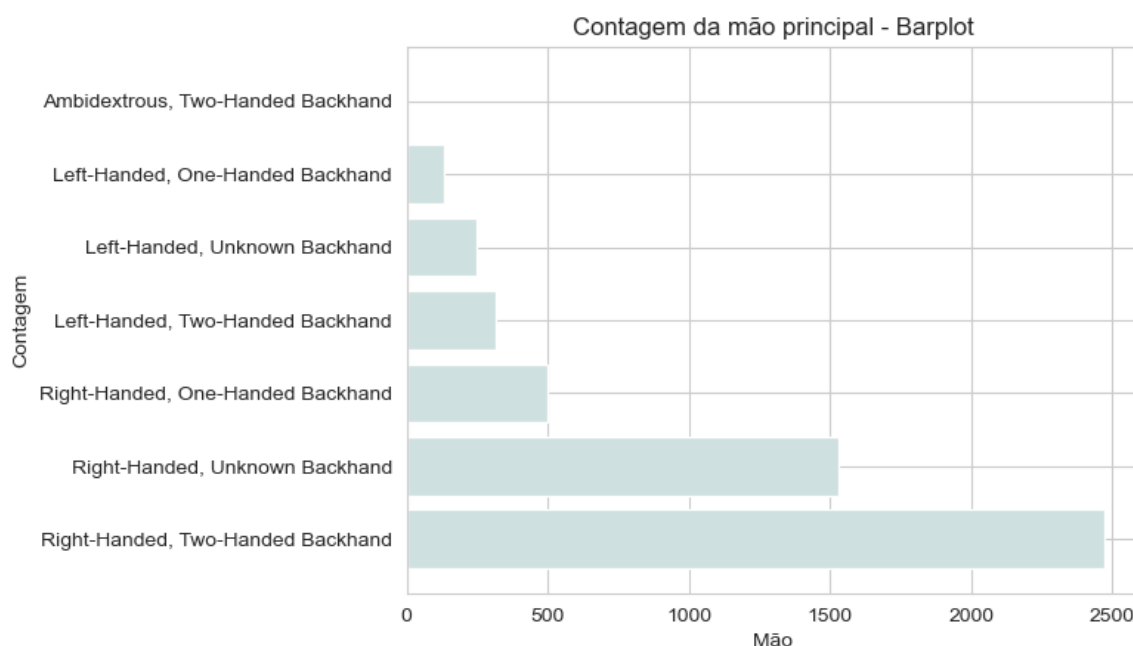


Figura 9 - Barplot com a mão principal

Mais à frente, estas duas serão separadas, a fim de se conseguir visualizar melhor as combinações possíveis e de tentar não usar valores omissos. Como é possível visualizar na **Figura 9**, existe *Unknown Backhand*, desconhecendo a *backhand* do jogador.

Tournament

Tournament é a variável que guarda o nome de cada um dos torneios realizados na Bélgica.

Na Bélgica estão registados 73 torneios diferentes e, na **Figura 10**, pode-se visualizar os torneios onde ocorreram mais de 200 jogos, podendo assim supor-se que, possivelmente, tratam-se dos torneios mais importantes. Pode-se visualizar que os três torneios com um maior número de partidas foram: **Mons**, **Belgium F2** e **Antwerp**.

No entanto, é difícil de retirar muita

informação desta variável, uma vez que só corresponde a um dado informativo, só para nomear o torneio. Isto leva a pensar se realmente esta variável é importante no que diz respeito à previsão do número de *sets* porque esta apenas dá uma *name tag* ao torneio.

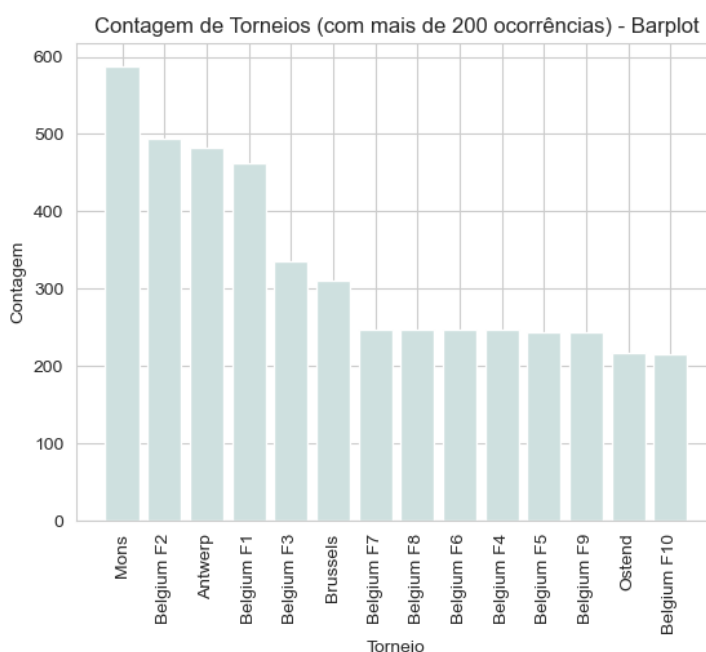


Figura 10 - Barplot com os torneios com mais de 200 ocorrências



Location

A coluna **Location** guarda o local exato onde foi realizado a partida de determinado torneio. Neste trabalho só foram consideradas as partidas realizadas na Bélgica e, por isso, foi logo realizado um filtro para apenas ter informações sobre este país.

Date

Date tem como valores as datas em que os torneios – e as partidas destes – foram realizadas. Na Bélgica, estão registadas partidas desde 1968 até 2021.

GameRound

GameRound é a coluna que representa a ronda da partida de um determinado torneio. É de se notar que os torneios possuem rondas diferentes. O gráfico da **Figura 11** mostra a quantidade de vezes que cada tipo de ronda ocorreu.

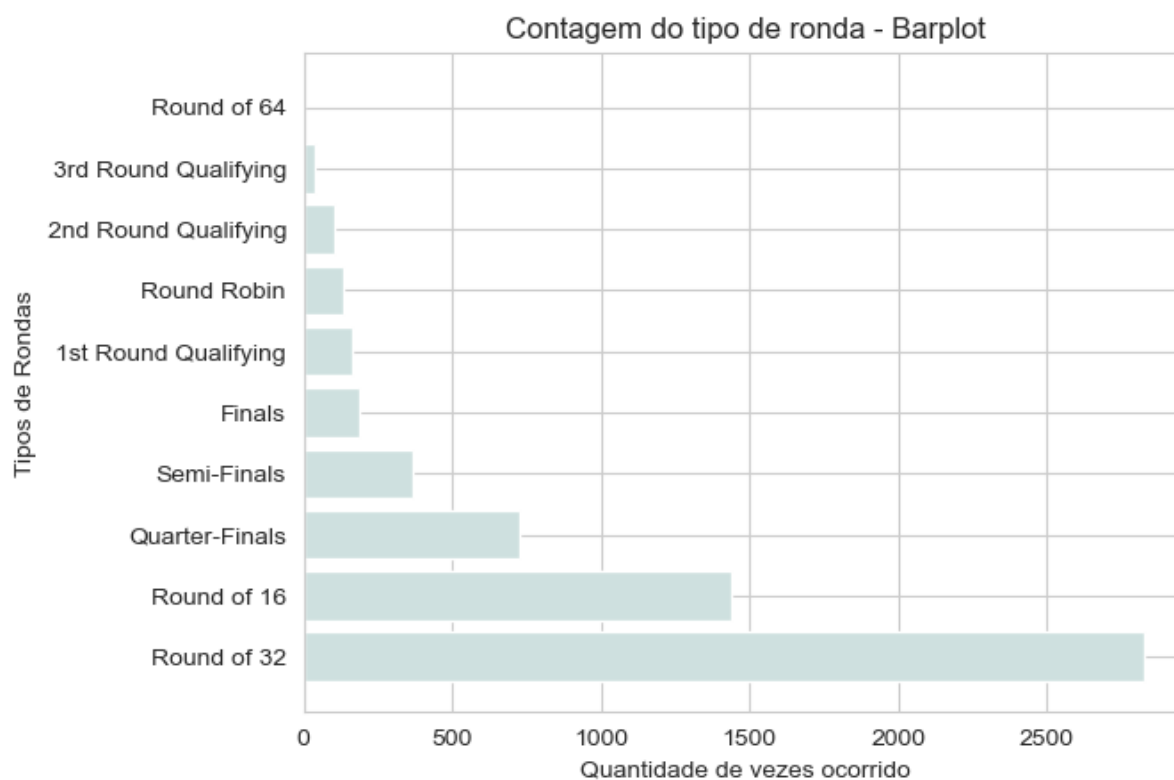


Figura 11 - Barplot com o tipo de ronda

Oponent

Oponent tem como valor o nome do oponente de dada partida, semelhante aos dados de **PlayerName**. Existem 1920 oponentes diferentes nas partidas realizadas na Bélgica.

WL

WL é a variável que diz se o jogador, cujo nome está em **PlayerName**, ganhou ou perdeu a partida. Quando o jogador ganha, **WL** assume o valor W (*Win*); quando o jogador perde, **WL** assume o valor L (*Lose*).



Score

Score é a variável que guarda o resultado de cada *set* de dada partida. Cada par de números corresponde aos resultados que o **PlayerName** e o **Oponent** obtiveram em um dado *set*. O número à esquerda representa a pontuação do **PlayerName** naquele dado *set*; o número à direita representa a pontuação do **Oponent** naquele dado *set*. Nem todos os pares de números, i.e., *sets*, tem um 6 ou um 7. Tais pares que não têm nenhum desses números são *sets* irregulares. Existem duas razões para uma partida ter tais *sets* irregulares:

- “(W/O)” - significa que um dos jogadores não compareceu e, portanto, a partida não se realizou;
- “(RET)” - significa que um dos jogadores desistiu da partida. A desistência pode ter ocorrido devido a uma lesão, por exemplo;
- “(DEF)” - significa que a partida foi interrompida por má conduta.

Data Preparation

Nesta fase, que corresponde à terceira fase do CRISP-DM, limpar-se-ão e preparar-se-ão os dados. É de notar que o problema dos jogos espelhados já foi anteriormente resolvido.

Esta fase do CRISP-DM possui, em especial, um elevado grau de importância, pois é aqui que são feitas as transformações dos dados para que estes possam ser introduzidos num algoritmo de *Machine Learning*. Estes algoritmos são, por vezes, bastante ineficazes - e às vezes disfuncionais - devido à existência de valores omissos e/ou *outliers*. Como solução para estes problemas, optou-se por realizar *WebScraping*, a fim de não só conseguir extrair os dados dos oponentes, que são escassos, como também sobre os *outliers* que prejudicam, de forma grave, a capacidade preditiva do modelo. Após ter-se realizado *WebScraping*, apercebeu-se da existência de alguns *outliers* e de valores omissos, pois, nem sempre existem dados de todos os jogadores, sendo que alguns destes podem estar errados. Mais à frente, serão abordados todos os tópicos referidos anteriormente, como a explicação das novas variáveis que foram criadas.

Dados dos oponentes

É pedido que sejam utilizados dados referentes aos oponentes, no que diz respeito às partidas da base de dados. Por esta razão foi criada uma tabela que contivesse todos os dados dos jogadores para que, caso um jogador numa partida (**PlayerName**) fosse o oponente noutra (**Oponent**), seja possível obter esses dados, sem se realizar uma pesquisa mais aprofundada.

WebScraping

Nesta fase, será explicado como foi realizado o processo de *WebScraping*. Os *sites* que foram utilizados para se retirarem os dados foram o [ATPTour](#) e o [TennisExplorer](#)



Relativamente às informações retiradas, optou-se por retirar o máximo proveito das informações dos jogadores, tendo recolhido os seguintes dados sobre estes: o seu país de nascimento, a sua altura, a mão que o jogador usa para jogar, o seu *rank*, o seu peso, a sua data de nascimento e o seu treinador. Graças a esta recolha dos dados, não só se conseguiu resolver alguma parte do problema dos valores omissos, como também se conseguiu recolher dados novos que contribuirão para uma possível melhor previsão. Todos os novos dados que foram recolhidos foram selecionados após deliberação realizada anteriormente, a fim de verificar se estes poderiam ou não ser úteis, como, por exemplo, o peso, que será utilizado com a altura para calcular o Índice de Massa Corporal (**IMC**) para depois se realizar uma comparação entre os 2 jogadores de dada partida, pois um jogador com um físico melhor tem uma maior tendência para vencer e, provavelmente, fazê-lo de forma breve, caso o seu físico seja substancialmente melhor que o do seu oponente; também, o treinador, pois nem todos os jogadores possuem treinador e, possivelmente, os jogadores que possuem serão melhores do que os restantes.

Apesar desta fase poder parecer que foi fácil e simples, obtiveram-se vários problemas, que serão comentados abaixo.

Tempo de demora e limitações

A recolha dos dados não foi nem rápida nem fácil, pelo contrário, foi muito demorada e intensiva. A recolha dos dados demorou, aproximadamente, 2 semanas para estar completamente finalizada. Neste processo, ocorreram vários erros na execução do *WebScraping*: ou o *site* bloqueava a recolha dos dados (ou ocorria um erro), fazendo com que se perdesse tudo, tendo de voltar ao início; havia jogadores que não tinham link, então foi necessário programar para que fosse pesquisado no *site* o respetivo nome do jogador para recolher os dados. Somado a isto, quando se deixava o *WebScraping*, não se podia mexer no dispositivo que estava a correr o código, tendo de deixar o dispositivo ligado a noite inteira.

Foram então criados vários ficheiros *csv*, que continham a informação de todos os jogadores, quer estes fossem da coluna **PlayerName** ou **Oponent**. Com estes dados recolhidos num único *csv*, foi então realizado um *merge* na base de dados original, podendo assim obter os dados sobre todos os jogadores de forma atualizada e, também, continuar o processo de tratamento destes. Ter em atenção que este *merge* mantinha todos os dados da base de dados atribuída inicialmente para este trabalho, substituindo os nulos que esta possui pelos valores do *dataframe* do *WebScraping*. Se fosse nulo nos dois *dataframes*, continuaria nulo. É importante lembrar que, mesmo assim, existiam dados errados e omissos.

Curiosidade: foi utilizado o AWS ao executar o código do *Webscraping* e demorou cerca de 36 horas para executar as várias partes dos ficheiros na integra, ao gerar os *csv*'s.

Limitações da recolha dos dados

Mesmo após a recolha dos dados, reparou-se em várias limitações. A imagem abaixo mostra o perfil de um oponente no *site* [TennisExplorer](#). Visualizando com atenção, é possível reparar que este perfil não possui um *rank* atual (Corresponde ao *Current/Highest rank - singles*). Muitos jogadores desta base de dados



já não jogam ténis de forma profissional e, devido a isso, eles não possuem um *rank* atual. Devido a esta limitação, optou-se por recolher o *rank* mais alto que esse jogador teve, pois esse é o único dado que tivemos acesso a.

Zelba Marious - profile



Zelba Marious
 Country: Australia
 Age: 33 (15. 5. 1990)
 Current/Highest rank - singles: - / 1144
 Current/Highest rank - doubles: - / 786
 Sex: man
 Plays: right

Figura 12 - Jogadores sem *rank* atual


Jogadores com nomes diferentes

Na recolha dos dados, apercebeu-se que existiam, aproximadamente, 800 oponentes que não estavam na base de dados original. Estranhou-se tal acontecimento, pois os dados dos jogadores foram recolhidos pelo nome de cada um destes. Após uma breve análise, apercebeu-se que poderia haver um problema.

Abaixo, é possível visualizar 2 imagens (**Figura 13**), uma que mostra a respetiva linha do *WebScrapping* do jogador e outra com uma página de um jogador com o **nome muito parecido**.

| | | | | | | |
|------|--------------------|-----|-----|--------------|-----|-----|
| 2000 | Alexandr Dolgoplov | NaN | NaN | Right-Handed | NaN | NaN |
|------|--------------------|-----|-----|--------------|-----|-----|

Dolgoplov Aleksandr - profile



Dolgoplov Aleksandr
 Country: Ukraine
 Height / Weight: 180 cm / 71 kg
 Age: 34 (7. 11. 1988)
 Current/Highest rank - singles: - / 13
 Current/Highest rank - doubles: - / 42
 Sex: man
 Plays: right

Figura 13 - Jogadores com nomes distintos

Dando uma breve explicação, no [TennisExplorer](#) existe um jogador com um nome muito parecido a um nome da base de dados. No entanto, não existia um nome exatamente igual ao da base de dados. Devido a isto, surgiram duas questões: Será este jogador a mesma pessoa da base de dados? Os outros 800 jogadores possuem o mesmo problema, o que se há de fazer com eles?

Após deliberação, optou-se por não incluir estes jogadores, tendo perdido 800 linhas que poderiam prejudicar o modelo (seriam muitos nulos e, após a imputação, os dados ficariam “sintéticos”). Somado a



isso, não havia maneira de comprovar que eram os mesmos jogadores, só tendo o nome como método de comparação e, devido a isso, preferiu-se não usar estes jogadores no modelo final, a fim de só usar dados coerentes e corretos.

Tratamento de variáveis

Depois da fase de *Data Understanding* ficou nítido que algumas colunas da base de dados teriam de ser transformadas de forma a, ou unificar dados, ou a alterar o tipo do mesmo. Para além dessas mudanças, seria necessário adicionar e tratar a informação recolhida do **webscrapping**. Abaixo estão explicadas todas as variáveis que se decidiu utilizar, justificando a escolha de cada uma delas.

Altura e peso - IMC

Relativamente à altura e ao peso, mesmo após retirar os dados dos *sites* referidos anteriormente, observou-se que continuavam a existir valores omissos e, consequentemente, procedeu-se à imputação destes pela regressão (através da biblioteca [fancyimpute](#)), de forma a tentar imputar os dados de forma a não induzir os dados.

Estas duas variáveis à partida iriam ter muita correlação entre si, o que poderia provocar multicolinearidade no que diz respeito ao modelo. Com isto, a solução que se encontrou foi conjugar estas variáveis para o cálculo do índice de massa corporal, de forma a transformar estas 2 variáveis em uma única.

$$\text{IMC} = \frac{\text{PESO}}{\text{ALTURA}^2}$$

Figura 14 - Cálculo do IMC (Índice de Massa Corporal)

Para saber como se calculou os valores desta nova variável, basta visualizar a fórmula da mesma na **Figura 14**:

Para o cálculo do **IMC**, transforma-se o peso em KG e a altura em metros. Abaixo está explicado como se interpretam os valores desta variável, pois, como esta foi usada no

modelo, é necessário entender o que os seus valores significam.

Na **Figura 15** é possível visualizar uma referência que permite entender o que os valores obtidos significam, podendo assim interpretar-se os dados da nova variável:

Analisando com atenção, percebe-se que um bom **IMC** se situa entre 18,5 e 24,9. Por se tratar de jogadores de um desporto com um grande desgaste físico, é necessário

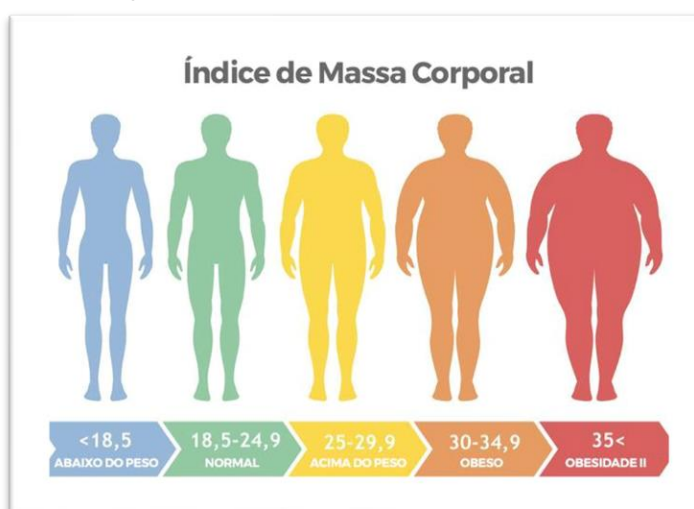


Figura 15 - Escala do IMC



que estes tenham um **IMC** equilibrado. Caso os atletas tenham maior ou menor **IMC** que os limites acima referidos, estes não de ter pior aptidão física, algo que é bastante capaz de se refletir nas partidas em que participam.

Para se fazer um método de comparação entre os dois jogadores de dada partida, escolheu-se usar a diferença entre os **IMC** do **PlayerName** e o **Oponent** (relembrar que o **PlayerName** aparece primeiro e é o vencedor da partida devido à forma de como foram removidos os jogos espelhados). Caso o valor da diferença seja positivo, o **IMC** do **PlayerName** é superior; caso o valor da diferença seja negativo, o **IMC** do **PlayerName** é inferior.

Nota: Relativamente aos *outliers* do [Height](#), estes foram corrigidos através do *Webscrapping*

País de naturalidade - Born

O local de nascimento dos jogadores é algo bastante complicado de manipular a fim de conseguir torná-lo numa *feature* útil e que tenha algum impacto no modelo. Crê-se em tal, devido ao facto de num certo país poderem existir jogadores muito bem classificados, com grandes feitos e muito premiados e existirem, também, jogadores com *rank*s não tão bons que perdem a maioria dos jogos. Somado a isso, devido à existência de um elevado número de nacionalidades diferentes, achou-se útil simplificar a informação original, pelo que se decidiu criar uma variável binária que indicasse se o **PlayerName** e/ou o **Oponent** eram belgas. Escolheu-se criar tal variável pois crê-se que o fator casa pode significar em maior apoio por parte da plateia, algo que poderia influenciar de forma positiva quem está a jogar em casa, o que, em princípio, resultaria ter numa partida mais curta. A variável foi feita de forma que os seus valores sejam ou 0, 1 ou 2:

- 0, caso nem o **PlayerName** nem o **Oponent** sejam belgas;
- 1, caso somente um dos jogadores seja belga;
- 2, quer o **PlayerName**, quer o **Oponent** sejam belgas.

Mão dominante – Hand

Nesta variável foram excluídos dados relativos à *backhand* quer do jogador, quer do oponente, pois não só existia um elevado número de omissos, como também nos *sites* usados no *Webscrapping* não existiam dados sobre essa variável. Com isso em mente, escolheu-se somente usar os dados relativos à mão dominante dos jogadores, o que, em termos práticos, significa se o jogador era destro ou canhoto.

Para realizar tal transformação foram criadas duas variáveis binárias, uma relativa ao **PlayerName** e outra relativa ao **Oponent**, que verificavam se estes eram destros – o mais comum - ou não. A variável implementada no modelo

Combinação das mãos nos jogos

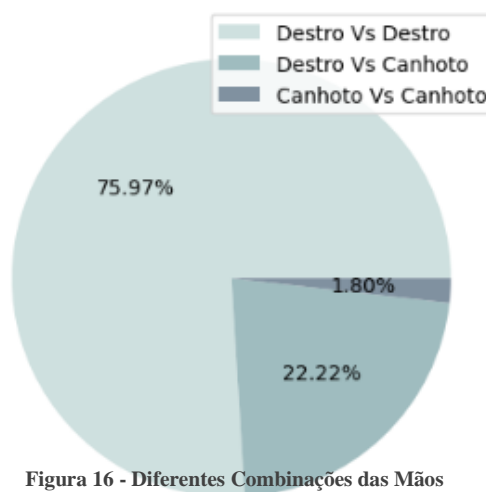


Figura 16 - Diferentes Combinações das Mãos



tinha como valor a soma dos valores destas duas variáveis binárias, dando assim a entender a quantidade de jogadores destros que jogaram naquela partida (que tinham o número 1).

Na **Figura 16**, é notório que existe um elevado número de partidas entre 2 jogadores destros, quando comparado com as outras 3 combinações possíveis, podendo assim perceber que estes podem criar tiques e hábitos. Tendo já esta ideia em mente, podemos referir que os canhotos podem possuir uma vantagem contra os destros, pois, não só existe um menor número de jogadores canhotos, como os jogadores estão mais habituados a jogar contra destros, sendo assim muito interessante incluir esta variável no modelo.

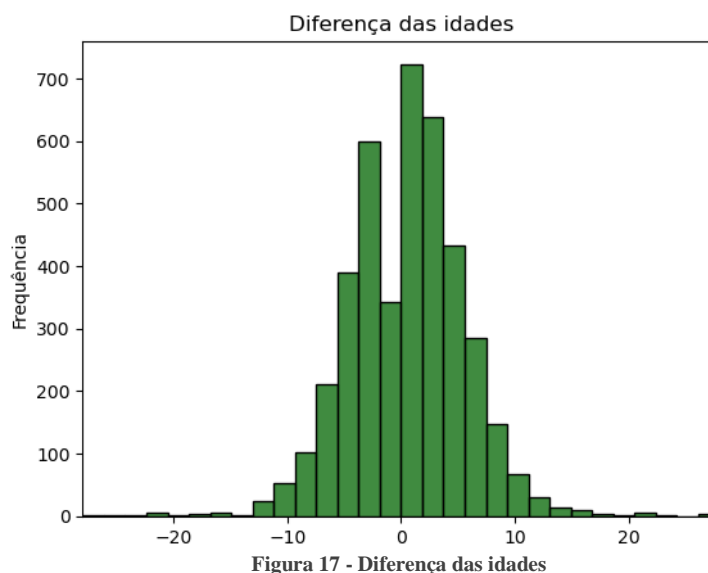
Date

Na base de dados, estão presentes jogos que ocorreram entre 1968 e 2021. Os torneios costumam ter várias rondas em dias diferentes, isto é, um torneio não ocorre num único dia. De forma geral, os torneios têm uma duração de 6 dias, ou seja, não durante muito tempo.

Apesar da duração do torneio, optou-se por criar uma variável que continha o ano em que este ocorreu, pois, o desporto tem evoluído gradativamente em questões técnico-táticas e, por isso, a altura em que ocorreu o torneio/partida pode ser flagrante em termos de previsão do número de *sets*. Nas partidas mais recentes há mais análises prévias às partidas, bem como mais preparação, do que o que havia na década de 70, por exemplo.

PlayerAge

Relativamente ao **PlayerAge**, foram usadas 2 variáveis para a criar, sendo essas, a data de nascimento do jogador e a data do torneio, nomeadamente o ano de cada uma. Os valores desta *feature* são obtidos através da subtração do ano do torneio pelo ano de nascimento dos jogadores, a fim de se obter a idade que o jogador tinha quando jogou aquele jogo. Os valores da variável usada no modelo são obtidos através da subtração da idade do **Oponent** à idade do



PlayerName. Quando o resultado é positivo o **PlayerName** é o mais velho dos dois jogadores; quando o resultado é negativo o **PlayerName** é o mais novo dos dois. Analisando a **Figura 17**, é possível verificar que, de forma geral, os jogadores possuem a mesma idade, havendo alguns casos em que existe uma diferença de 10 anos entre estes e, em alguns casos mais raros, uma diferença de 20 anos.

As performances dos jogadores dependem das suas características físicas. Jogadores com maior idade têm uma maior chance de se lesionarem e, em princípio, têm também, uma maior tendência para terem



piores performances – no que diz respeito à agilidade e, também, a tempo de reação, por exemplo – em relação a jogadores mais novos. Com isto, não se quer dizer que ser mais novo é em tudo melhor. É importante lembrar que jogadores mais velhos possuem mais experiência que os jogadores mais novos, algo que pode ser decisivo no resultado de dada partida.

Ground

O tipo de chão, como já foi visto anteriormente, só tem três categorias em torneios belgas. Tendo isso em conta, decidiu-se utilizar esta variável categórica, mas de forma diferente, em que os valores possíveis foram codificados por números. Decidiu-se que se iria usar os números 1, 2 e 3, sendo o 1 atribuído a “Clay”, o 2 a “Hard” e o 3 a “Carpet”. Optou-se por esta ordem na atribuição dos valores por classe, pois queria-se que quanto mais presente fosse uma classe na base de dados, menor seria o valor que lhe seria atribuída. Tal como se pode observar em [Ground](#).

Treinador

O treinador foi uma das variáveis retiradas através de **WebScrapping** que se concluiu que seria importante para o modelo. Como esta variável continha bastantes valores omissos, em proporção, reparou-se que só tinham treinador os jogadores que tinham um nome mais célebre no cenário do ténis. Daí, surgiu a ideia de fazer uma *feature* que avaliava se o jogador tem, ou não, um treinador conhecido, baseando-se na informação dos *sites* referidos anteriormente. A informação existente dos *sites* dá a entender que o treinador é, não só importante, como também relevante para o término de um jogo de ténis, principalmente nas pausas ou na preparação prévia dos jogos, que pode influenciar em quantos *sets* o jogo termina.

No entanto, de forma a colocar esta informação numa só coluna, teve-se em conta se, quer o jogador, quer o oponente, tinham um treinador conhecido, transformando cada uma das variáveis em variáveis binárias em que 1 significa que tem treinador e 0 significa que não tem treinador. A variável usada no modelo tem como valores possíveis os valores 0, 1 e 2. Tais valores são obtidos através da soma dos valores das duas colunas iniciais referidas anteriormente. O significado de cada um dos valores possíveis está descrito abaixo.

- 0 - Ambos os jogadores não tiverem treinador conhecido;
- 1 - Apenas um dos dois tiver treinador conhecido;
- 2 - Ambos os jogadores tiverem treinador conhecido.

Prize

Relativamente ao [prize](#), foi necessário fazer muitas alterações. Primeiro é importante referir que esta variável possuía 2 tipos de moeda diferentes, sendo essas, o euro e o dólar. Decidiu-se que, para esta variável ter valores coerentes, seria necessário converter todos os prémios para dólares ou para euros. Após um breve estudo, reparou-se que existiam jogos localizados na Bélgica antes do euro existir. Na **Figura 18** está demonstrado um exemplo disso:



Figura 18- Prémio em dólares no site do ATP, antes de 1999

Assim sendo, optou-se por converter tudo para dólares, devido não só à impossibilidade de converter os prémios das partidas realizadas antes de 1999 - criação do euro - para euros, como também pelo site ATP possuir os prémios das partidas realizadas antes de 1999 em dólares.

Para esta conversão, seria necessário descobrir a taxa de câmbio correta para se aplicar. Para a aplicar da forma o mais fidedigna possível, decidiu-se utilizar os dados do [site do Banco de Portugal](#) de forma recolher a taxa de câmbio pretendida. Após a recolha, notou-se que a taxa de câmbio mudava a cada dia e, para solucionar tal problema, optou-se por realizar uma média por cada ano, a fim de se conseguir converter tudo para dólares, visto que não há uma alteração tão brusca durante o ano.

Analisando com maior atenção, reparou-se que existe uma ronda intitulada de *Round Robin* que não possuía prémios e, devido a isso, transformou-se os nulos em 0, para simbolizar que não se recebe prémio neste tipo de ronda.

A utilidade desta variável foi que torneios que tenham jogos com maior premiação, possam criar uma ânsia maior dos jogadores de ganhar efetivamente o jogo e contribuir que o jogo tenha mais *sets*.

Rank

Esta variável tem como valor a diferença entre o *rank* do **PlayerName** com o *rank* do **Oponent**. Crê-se que esta variável é de grande importância para se implementar no modelo, pois seria uma comparação direta entre os dois jogadores. Feita a subtração dos dois *ranks*, caso este seja positiva, significa que o *rank* do **PlayerName** é superior que o *rank* do **Oponent** (o inverso também ocorre). Quanto mais próximo de 0, mais próximo será o *rank* dos dois jogadores.

É importante referir que, apesar da realização do **Webscrapping**, ainda existiam valores omissos e, para resolver isso, foi realizado uma imputação através da regressão para tentar não induzir o modelo em erro.

Feita a imputação, escolheu-se usar esta variável, pois a uma maior diferença entre os *ranks* dos participantes de dada partida, maior é a chance de esta ter menor número de *sets*, pois há uma diferença notável entre a qualidade de cada jogador.



Final

Relativamente ao [GameRound](#), surgiu de imediato uma ideia de como poderia ser útil preparar esta variável para a fase de modelação.

Normalmente, em qualquer competição, os melhores jogadores costumam chegar às últimas rondas das competições, fazendo com que nas finais haja um confronto entre os melhores, provocando partidas mais intensas e renhidas e, provavelmente, mais longas. Com esta ideia em mente, decidiu-se criar uma variável que teria como valores 0 e 1. A variável teria 1 como valor caso a **GameRound** possuísse a palavra **Final** no nome e teria 0 caso contrário. Alguns exemplos de **GameRound's** com **Final** no nome são as semifinais, os quartos de final e a própria final.

Variável Target – Número de Sets

A variável cujos valores pretende-se prever, número de *sets*, foi uma variável em que foi necessário ter vários cuidados durante o tratamento dos dados. A *target* foi retirada da variável [Score](#), tendo sido necessário uniformizá-la - colocar um único espaço entre cada pontuação de cada um dos *sets* - para poder então conseguir contar o número de *sets* que ocorreram em cada partida.

De seguida, foram apagadas todas as linhas que possuíam (W/O), (RET) e (DEF) da coluna do [Score](#), pois estas partidas foram ocasiões especiais. Abaixo está descrito de forma breve o que cada uma destas ocasiões especiais significa:

- (RET) - Um dos jogadores lesionou-se;
- (W/O) - Um dos jogadores não compareceu à partida;
- (DEF) - Má conduta.

Também é importante referir que foi feita uma separação das partidas à **melhor de 3** das partidas à **melhor de 5**. Para esta divisão, foi feita uma contagem do número de *sets* que o vencedor venceu. Caso fosse igual a 2, seria uma partida à **melhor de 3** e, caso fosse superior a 2, seria à **melhor de 5**.

Relembrando mais uma vez, as partidas à **melhor de 3** possuem 3 *sets* no máximo. Este trabalho tem como foco as partidas à **melhor de 3**, mas em anexo está uma análise muito superficial sobre as partidas à **melhor de 5**.

Features a utilizar na fase de modelação

Tendo em conta o que foi referido anteriormente, obtiveram-se várias *features*, que estão listadas na **Tabela 2**.



Tabela 2- Variáveis transformadas para modelação

| Feature | Explicação da Variável |
|-----------------------|--|
| <i>GroundNumerico</i> | Variável que tem como valor, de forma codificada, o tipo de chão da partida |
| <i>Prize</i> | Variável que tem como valor o prémio do torneio |
| <i>TournamentYear</i> | Variável que tem como valor o ano em que o torneio ocorreu |
| <i>DiffIMC</i> | Variável que tem como valor a subtração do IMC do Oponent ao IMC do PlayerName |
| <i>DiffAge</i> | Variável que tem como valor a subtração da idade do Oponent à idade do PlayerName |
| <i>DiffRank</i> | Variável que tem como valor a subtração do <i>rank</i> do Oponent ao <i>rank</i> do PlayerName |
| <i>QTDBelgas</i> | Variável tem como valor a quantidade de jogadores belgas na partida |
| <i>Finals</i> | Variável que refere se a ronda da partida possui a palavra Final |
| <i>SumMainHand</i> | Variável que tem como valor a quantidade de jogadores destros na partida |
| <i>SumTrainer</i> | Variável que tem como valor a quantidade de jogadores na partida que possuem um treinador |
| NumeroSets | Variável Target |

Modeling & Evaluation

Nesta fase será feita não só a criação dos modelos, como também será realizada uma breve análise das associações das diferentes variáveis quer com a variável *target*, quer entre si, a fim de visualizar se existe correlação entre as *features*.

Ver-se-ão a correlação de *Pearson*, o *ETA Squared* e, também, o *V de Cramer*, a fim de conseguir realizar análises de forma correta.



Correlação e associação das variáveis preditoras entre si

Coeficiente de correlação de Pearson

Primeiramente, medir-se-á o coeficiente de correlação de *Pearson* para conseguir visualizar a correlação entre as variáveis quantitativas. Como a variável *target* ***NumeroSets*** não é quantitativa, mas sim qualitativa, esta fase servirá para verificar se existe multicolinearidade entre as variáveis quantitativa existentes. Na **Figura 19** está a matriz de correlação de *Pearson*:

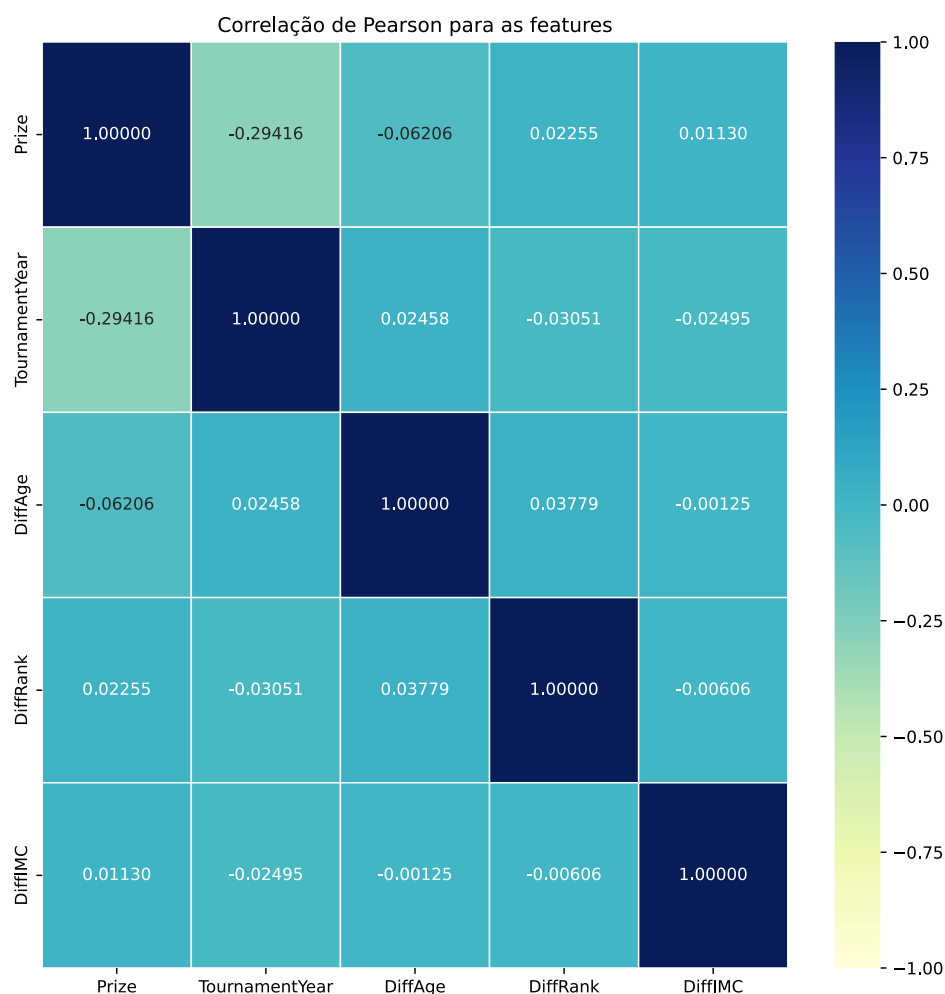


Figura 19- Correlação de Pearson para as features

Tendo em conta os valores da matriz, é de notar a correlação entre o ***Prize*** e o ***TournamentYear***, cujo valor é, aproximadamente, -0.29. Uma possível justificação para o valor desta correlação poderá ser a diferença entre valor atual do dólar com o valor que tinha há anos. Para uma melhor compreensão pode-se realizar cálculos no [Website](#), que faz uma comparação entre o poder de compra do Dólar, através do IPC (Índice de Preços de Consumidor). Analisando com atenção, pode-se concluir que o antigo poder de compra do dólar era superior ao poder de compra atual do mesmo.

Tendo isto em conta, faz sentido que a correlação seja negativa, i.e., em termos práticos, quanto menor o ano do torneio, maior será o valor do prémio. Tal faz sentido pois não se consegue afirmar que ano



base o ATP utilizou para colocar os prémios, ou até se teve cuidado com a inflação de cada ano; é importante relembrar que se converteu euros para dólares com uma média anual da taxa de câmbio, o que poderá provocar valores não totalmente reais.

Não obstante o que foi dito acima, considerou-se o risco de multicolinearidade - no que diz respeito a estas variáveis - baixo, pelo que decidimos usar todas estas variáveis.

V de Cramer

Relativamente ao coeficiente de associação *V* de *Cramer*, é importante referir que este mede associação entre variáveis nominais e, como a *target* é nominal, pode-se visualizar se existe uma boa correlação entre as *features* com a *target*, como também se pode visualizar se existe multicolinearidade. Abaixo está uma matriz com os valores de *V* de *Cramer*:

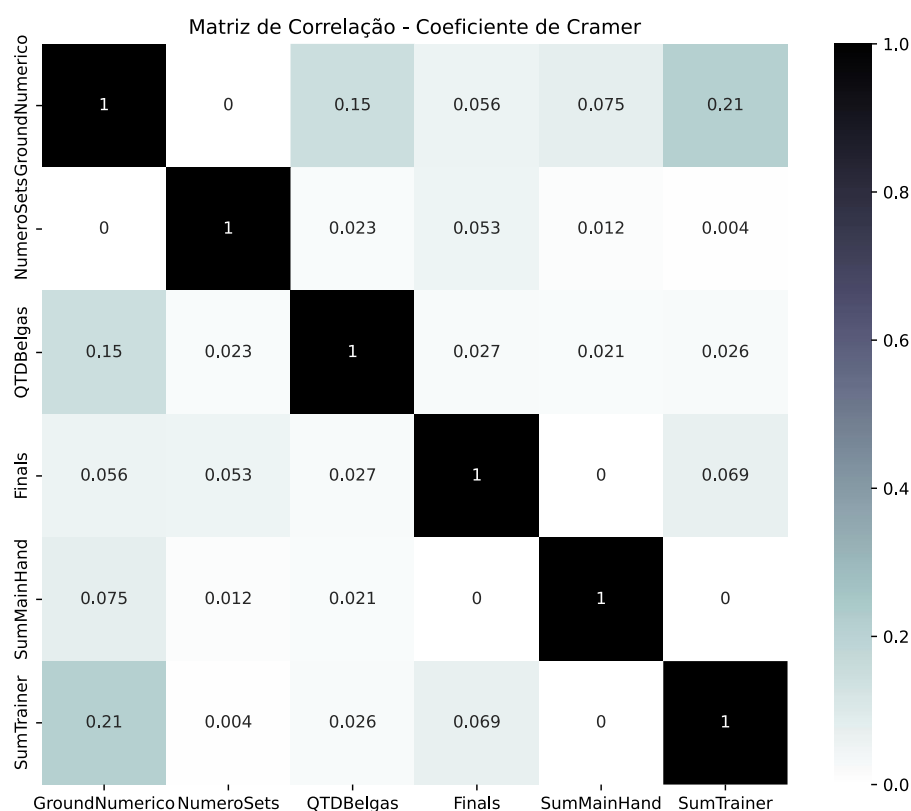


Figura 20- V de Cramer para as features

Tendo em conta os valores acima, conclui-se que a chance de multicolinearidade é bastante reduzida. No entanto, a associação destas variáveis com *target* é muito fraca, sendo *Finals* a variável com associação mais forte com a *target*, com apenas 5% de associação.

É importante referir que o *GroundNumerico* não tem nenhuma associação com *target* e, devido a isso, optou-se por testar mais à frente como esta variável se comportava no modelo.



ETA Squared

Esta é a última medida de associação que falta referir. Esta medida é utilizada para medir a associação entre variáveis quantitativas com variáveis qualitativas. Tem-se como objetivo visualizar a correlação entre as variáveis quantitativas com a *target*.

Em baixo podem-se então visualizar os valores obtidos:

- Coeficiente Eta entre *NumeroSets* e *Prize*: 0.075
- Coeficiente Eta entre *NumeroSets* e *TournamentYear*: 0.122
- Coeficiente Eta entre *NumeroSets* e *DiffAge*: 0.121
- Coeficiente Eta entre *NumeroSets* e *DiffRank*: 0.680
- Coeficiente Eta entre *NumeroSets* e *DiffIMC*: 0.765

As associações do *target* com *DiffRank* e com *DiffIMC* são substancialmente superiores às associações do *target* com as restantes variáveis. *DiffRank* e *DiffIMC* têm associações muito elevadas com a variável *target*, sendo elas, aproximadamente, 0.68 e 0.77, respetivamente.

Pode-se então concluir que o cálculo do *IMC*, e os arranjos da variável *rank*, foram benéficos em termos de aperfeiçoamento do modelo, pois, em princípio, devido à alta associação, estas serão as variáveis que mais irão aperfeiçoar o modelo. Pode-se então concluir que a capacidade física possui uma grande importância numa partida de ténis.

Partição dos dados e alguns modelos básicos

Inicialmente teve de efetuar-se a divisão entre conjunto de treino e teste e, para tal, estabeleceu-se que seria 70% para treino e 30% para teste. É de se referir que se passará por vários modelos, desde modelos mais simples até modelos mais complexos, sendo que no final será dado um maior destaque ao que apresentar, não só maior robustez, como também os melhores resultados. Com a partição proposta foram utilizados **2823** registos no conjunto de treino e **1210** registos no conjunto de teste, tendo cada um dos conjuntos 11 colunas.

Começando por modelos mais básicos, que serviriam apenas para verificar se as colunas dos dados de treino e teste estavam bem separados, aplicaram-se dois modelos com *cross-validation* de $k=5$. Os dois modelos usados foram Regressão Logística (*LR*) e *K-Nearest-Neighbour* (*KNN*) e foram testados utilizando a variável *GroundNumerico*, variável esta que, como visto anteriormente, não tem grande associação com a variável *target*. A métrica testada para estes modelos mais simples foi a *accuracy* média nos 5 *folds*, que registou um resultado de aproximadamente 0.60 para o *KNN* e 0.68 para o de *LR*. Não se elaborou mais na conceção destes modelos, pois o número de linhas no conjunto de teste é consideravelmente pequeno e, também, devido às classes do *NumeroSets* estarem bastante desbalanceadas (2703 partidas com 2 sets e 1330 partidas com 3 sets na base de dados), contribuindo para o enviesamento dos modelos. Como estes modelos utilizam probabilidades ou distâncias é difícil trabalhá-los com as condições acima mencionadas.



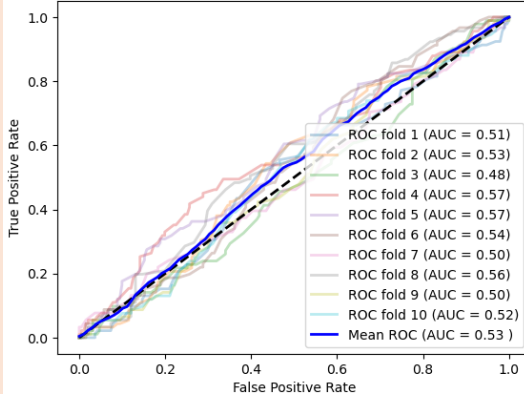
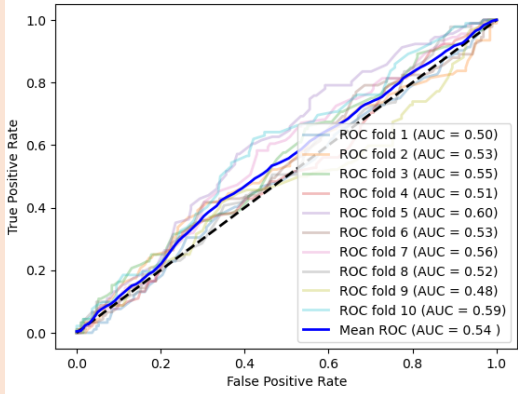
Quanto à discrepância das classes, nota-se que poderá ter um impacto negativo nos modelos, devido a estes tenderem mais para um determinado número de sets do que para outro. Uma justificação por não ter sido feito algo acerca disto, poderá ser, por exemplo, a aplicação da *Data Augmentation* - processo de incrementar dados para um modelo -, técnica que utiliza dados repetidos e artificiais, contribuindo para o *overfitting*, menor diversidade e mais enviesamento do modelo. No entanto, para colmatar o problema associado à discrepâncias das classes, realizar-se-á a estratificação K-fold para reduzir não só a aleatoriedade do modelo, como também tornar menos evidente o desbalanceamento de classes.

Variável “GroundNumerico” e impacto da mesma

Para verificar como a *feature* de **GroundNumerico** influencia no modelo, testou-se através de uma *Random Forest* (apenas porque foi o primeiro modelo e por ser robusto) para ver como esta variável poderia mudar as métricas e, se calhar, melhorar as previsões do modelo. Para tal, testou-se quer com a partição estabelecida em cima, quer com validação cruzada por *k-fold partition*, que dividia o conjunto total de dados à melhor de 3 *sets*, em k=10 partições. Foi discutido este valor, pois apresenta um número suficiente de *folds*, reduzindo a variância e não sendo computacionalmente intensivo. Assim, o modelo é favorecido com uma robustez maior e com vários conjuntos diferentes, ajudando numa maior representatividade dos dados de teste.

A **Tabela 3** mostra a diferença dos modelos com e sem a *feature*, demonstrando a sua influência:

Tabela 3- Impacto GroundNumerico num modelo de Random Forest

| RFC | <u>Com a Variável GroundNumérico</u> | <u>Sem a Variável GroundNumérico</u> |
|-----------------|--|---|
| Partição 70/30 | <p>Métricas</p> <ul style="list-style-type: none"> Accuracy: 0.633 Recall: 0.896 Precision: 0.664 AUC: 0.516 | <p>Métricas</p> <ul style="list-style-type: none"> Accuracy: 0.635 Recall: 0.992 Precision: 0.663 AUC: 0.512 |
| K-Fold com k=10 | <p>K-Fold Validation Random Forest</p>  <p>Mean AUC = 0.53</p> | <p>K-Fold Validation Random Forest (sem o Ground)</p>  <p>Mean AUC = 0.54</p> |

Analisando a tabela acima, pode-se visualizar que existiram algumas mudanças não muito significativas relativamente aos resultados.



Começando pela partição 70% treino e 30% teste, de forma geral, todas as métricas mantiveram o mesmo valor, exceto o *recall*, que teve um aumento significativo após a exclusão da variável *GroundNumerico*. No entanto, o modelo parece que se tornou um pouco menos aleatório com a retirada desta variável, devido ao pequeno aumento do valor do AUC.

No que diz respeito aos gráficos, estes foram obtidos através da partição *k-fold* e, para um melhor entendimento de como esta partição funciona, é necessário explicá-la:

- No *k-fold cross-validation*, os dados disponíveis são divididos em *k* partes (chamadas "*folds*") de tamanho aproximadamente igual. O modelo é treinado e avaliado *k* vezes, usando *k-1 folds* como conjunto de treino e o *fold* restante como conjunto de teste, fazendo com que cada *fold* seja usado como conjunto de teste uma única vez.
- Ao final das *k* iterações, são obtidos *k* resultados de desempenho diferentes que podem ser juntos, através da média, para fornecer uma estimativa mais robusta do desempenho geral do modelo.
- O *k-fold cross-validation* ajuda a mitigar o problema da variância nos resultados de avaliação, fornecendo uma medida mais confiável do desempenho do modelo. Também é útil para ajudar a detetar problemas como *overfitting* ou *underfitting*, pois o modelo é avaliado em diferentes conjuntos de dados durante cada iteração.

No que diz respeito à análise dos resultados, é possível visualizar que, quando retirada a variável *GroundNumerico*, o modelo ligeiramente melhor. Como *k-fold* é uma técnica de divisão mais confiável que a original 70% treino, 30% teste, conclui-se que seria melhor não incluir esta variável.

Na **Figura 21** é possível visualizar que o *GroundNumerico* não tem quase impacto nenhum na previsão da variável *target*. Estes valores correspondem ao modelo *Random Forest*, referido anteriormente, quando se utilizou a variável *GroundNumerico*. Analisando os valores, conclui-se que esta variável possui uma importância extremamente redundante.

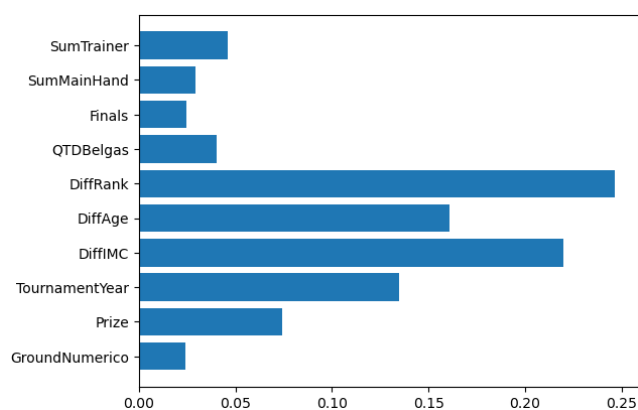


Figura 21- Importância das features no modelo de Random Forest

Devido à associação nula com a variável *target* e à pouca importância que tem no modelo, foi decidido retirar esta variável para os modelos seguintes.

Gradient Boosting e hiperparâmetros

Após aplicar o *Random Forest*, tentou-se encontrar outro modelo que conseguisse ser melhor, i.e., que apresentasse melhores resultados em termos de previsão e também no que diz respeito às métricas de classificação. Optou-se por utilizar *Gradient Boosting Tree Classifier*, modelo que, teoricamente, apresentava ser melhor que o *Random Forest*. É de referir que se fizeram testes com uma divisão de 70% para o treino e 30% para o teste.



Pode-se visualizar abaixo os resultados obtidos:

- *Accuracy*: 0.651
- *Recall*: 0.970
- *Precision*: 0.660
- *AUC*: 0.535

Comparativamente ao *Random Forest*, os resultados aparentam ser superiores. Assim sendo, decidiu-se escolher o *Gradient Boosting Tree Classifier* como o modelo principal deste trabalho, devido à sua robustez na escolha de parâmetros e de número de estimadores, e também porque é mais complexo na sua execução.

Alteração dos Hiperparâmetros

Após ter-se escolhido o modelo principal, tentou-se otimizá-lo ao máximo, de forma a obter melhores resultados. Para isso, foi necessário realizar alguns ajustes nos hiperparâmetros.

Para descobrir os melhores valores para os hiperparâmetros, realizou-se algumas iterações, com valores já predefinidos, a fim de não “estourar” a escala dos hiperparâmetros, sendo feita uma comparação entre os modelos alterados (*random search* através de um dicionário que tem várias listas de valores pré-estabelecidos para os hiperparâmetros). O resultado a que se chegou foi o seguinte:

- *subsample* = 0.95
- *n_estimators* = 1250
- *min_samples_split* = 3
- *min_samples_leaf* = 3
- *max_features* = 4
- *max_depth* = 3
- *learning_rate* = 0.005
- *random_state* = 772

De forma a entender um pouco melhor como estes parâmetros funcionam dar-se-á atenção ao que cada um destes significa:

- **subsample**: revela a fração das amostras a serem utilizadas na base de previsão;
- **n_estimators**: quantidade de treino a aplicar nas fases de *boosting* (quanto maior, melhor é a performance);
- **min_samples_split**: o número mínimo de entradas necessárias para separar um nó das árvores;
- **min_samples_leaf**: o número mínimo de entradas necessárias para constituir um nó folha;
- **max_features**: representa o número máximo de variáveis a considerar na melhor separação dos nós;
- **max_depth**: máximo de profundidade (número de nós na árvore) nos estimadores de regressão individuais;



- **learning rate**: corresponde ao quão depressa - ou devagar - o modelo aprende com os dados. Deve haver um equilíbrio para que não se aprenda nem muito rápido, nem muito lentamente, pois tal leva a previsões incoerentes e forçadas;
- **random state**: invoca uma *seed* para geração de resultados semelhantes e consistentes.

Finalmente aplicado o modelo, pode-se visualizar as métricas que foram obtidas com a divisão 70% treino e 30% teste:

- *Accuracy*: 0.653
- *Recall*: 0.987
- *Precision*: 0.657
- *AUC*: 0.542

Este modelo apenas prevê 15 jogos com 3 sets, dos quais apenas 5 foram corretamente previstos. Ocorreu um problema em que o modelo apenas previa a moda. Este problema será descrito com um maior número de pormenores mais à frente do relatório, nomeadamente, nos [Extras](#).

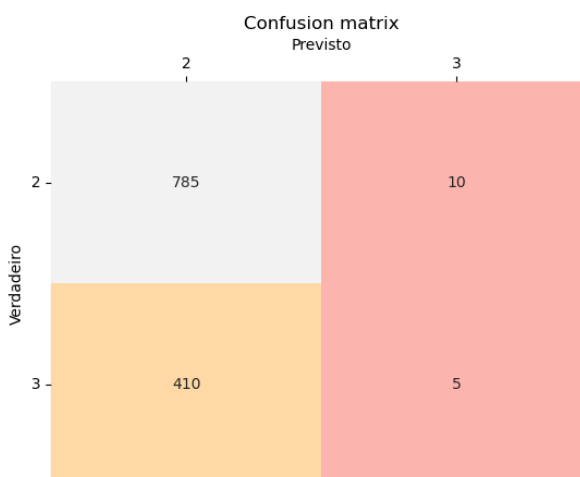


Figura 22- Matriz de confusão para o modelo GBTC com partição 70/30

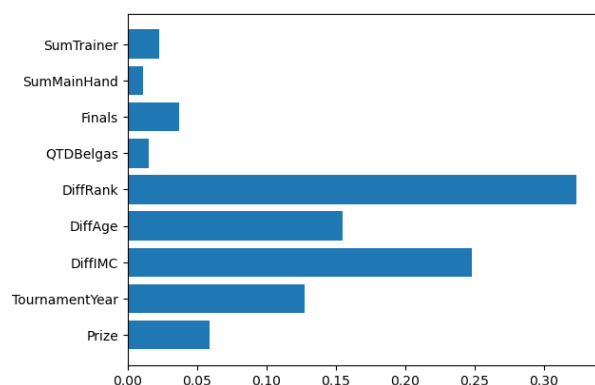


Figura 23- Importância das features no modelo de GBTC

Relembrando que *k-fold* é uma técnica de divisão mais fidedigna, esta também foi realizada, com $K=10$. Observando a **Figura 24**, é possível visualizar que o modelo teve um valor médio de AUC de 0.54, sendo um pouco melhor que um modelo completamente aleatório. É de notar que no *k-fold* 6 teve um AUC de 0.63, um valor consideravelmente positivo.

Tendo já visualizado a AUC, uma métrica de grande importância, passar-se-á a analisar outras métricas de classificação que permitem conhecer melhor os resultados do modelo:

Na **Figura 23** é possível visualizar um gráfico que compara o quão cada variável impactou no modelo. Analisando com atenção, é notório que as 3 variáveis que mais tiveram destaque foram: As diferenças de *rank*s, idades e *IMC*, concentrando quase 70% do modelo nelas.

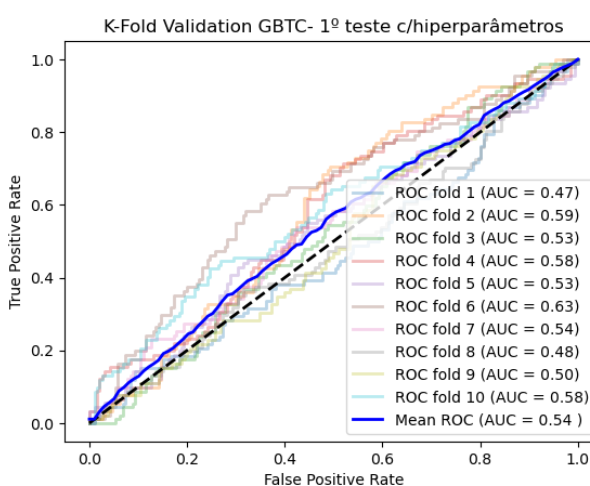


Figura 24- Partição k-fold para o modelo de GBTC (k=10)



- *Mean Accuracy*: 0.674
- *Mean Precision*: 0.677
- *Mean Recall*: 0.990
- *Mean F1-Score*: 0.804

Visualizando os resultados obtidos, caso se realize uma comparação com os resultados obtidos através da partição 70% treino e 30% teste, os resultados foram substancialmente melhores. Analisando a *Mean Accuracy* e a *Mean Precision*, observa-se que estes valores quase atingem 0.70, valores altos quando se tem em conta que se está a tratar de um problema com dados reais.

O *Mean Recall* foi excessivamente “bom”, em comparação com as duas métricas referidas anteriormente, sendo necessário interpretar este resultado com mais precaução. O *Recall* fornece informações sobre a capacidade do modelo em identificar corretamente os exemplos positivos e, tendo isto em conta, é importante lembrar que se possui classes desbalanceadas e, caso o modelo preveja todos da classe 2 *Sets* (classe modal), o valor da *Mean Recall* terá tendência a ser maior. Apesar de ter sido usado *k-Folds*, que tem como objetivo combater este problema, é importante lembrar que houve valores da AUC bastante baixos, não tendo como confirmar se nestes apenas era previsto a moda, contribuindo assim para este valor do *Mean Recall*.

Relativamente ao *F1-Score*, era esperado um valor alto, tendo em conta que este combina as informações da *Mean Precision* e *Mean Recall* em um único valor (média harmónica das duas).

Para mais informação sobre os resultados, basta visualizar o tópico das [Fórmulas](#) que se encontra nos Anexos, onde é fornecida uma melhor explicação sobre cada uma das métricas.

Extras – Problemas Relativos a Hiper parâmetros

Durante a execução e análise dos resultados existiram bastantes incongruências com as métricas de *output* dadas nos modelos. Por essa mesma razão e, como foi referido anteriormente, o modelo prevê quase a classe modal. Os testes realizados passaram, sobretudo, por utilizar a métrica do *F1-Score*, de forma a retirar os melhores valores dos hiperparâmetros abordados acima. Contudo, não existiu qualquer tipo de sucesso, devido ao facto de o modelo prever quase a moda, neste caso o 2 em *NumeroSets*, mas com resultados piores face aos anteriores (em termos de AUC).

- *subsample* = 0.95
- *n_estimators* = 1250
- *min_samples_split* = 8
- *min_samples_leaf* = 1
- *max_features* = 2
- *max_depth* = 6
- *learning_rate* = 0.005
- *random_state* = 772



Com estes parâmetros obtiveram-se métricas muito idênticas à moda, mas de lembrar que estas foram ajustadas depois de executar o código para escolher os parâmetros através do *RandomSearch*, com o dicionário. Os modelos anteriores a este e com um parâmetro a mais (*min_weight_fraction_leaf*) tendiam para previsões que basicamente eram a classe modal, estando completamente enviesados e incoerentes, optando-se por retirar do *RandomSearch*.

Deployment

Sendo o *Deployment* a última fase do *CRISP-DM*, não esquecendo que é uma parte fundamental do mesmo, é necessário abordar todas as conclusões, problemas e discussões referidas ao longo do trabalho, tendo que, de alguma forma, imaginar como seria aplicar este modelo em um problema num contexto real.

Para todos os pontos serem referidos, foram definidos vários tópicos dos temas abordados neste trabalho, tendo em conta as suas especificidades, problemas e soluções arranjadas.

Infraestruturas de Implementação

Apesar do trabalho já estar quase concluído, faltando a implementação desta fase, é importante lembrar que, para a realização deste, foram utilizados 4 computadores pessoais, tendo sido usado o *AWS* para a concretização do *WebScrapping*. Apesar do uso do *AWS*, o computador que estivesse a realizar o *WebScrapping* deixava de estar “funcional”, sendo necessário deixá-lo imóvel para que este conseguisse retirar os dados de forma correta e coerente. Somado a isso, ocorreram vários problemas, que levaram a que um membro do grupo não pudesse utilizar o mesmo.

Conclui-se este tópico a referir que, se não fosse pela perspicácia, força de vontade e pela existência de 4 computadores pessoais, um por cada membro, nunca teria sido possível realizar o *WebScrapping*, o que contribuiria para um trabalho mais limitado e simples, não permitindo que o grupo conseguisse explorar outros pontos de vista.

Integração de Dados

Relativamente aos dados em si, estes foram não só trabalhosos, como também problemáticos, tendo sido necessário procurar por várias soluções e abdicar de vários pontos de vista.

Comparativamente a outros países, a Bélgica não possui muitas partidas realizadas, possivelmente devido à sua fraca fama no mundo do ténis, o que, em termos práticos na resolução do trabalho, resulta em poucas linhas na base de dados, sendo muito difícil retirar alguma conclusão destas. Somado a isto, estes dados vinham com alguns problemas que foram necessários resolver, como *outliers* e valores omissos. É de lembrar que, também, foi necessário realizar a limpeza dos dados retirados do *WebScrapping*, pois estes não só não vinham tratados, como também foram retirados de um *Website* diferente do original,



possuindo dados novos, tais como o peso e o treinador de cada um dos jogadores. A solução final foi a realização da imputação por regressão, tal como já foi mencionado.

Antes de prosseguir para o próximo tópico, fica aqui referido, mais uma vez, que o tratamento dos dados não foi perfeito, tendo já sido explicado ao longo do trabalho.

Implementação do Modelo

Esta fase é, possivelmente, a mais frágil de todas, sendo necessário saber que modelo implementar e como o implementar, podendo alterar os hiperparâmetros para melhores resultados. Visualizando os resultados obtidos, é de fácil visualização que os resultados obtidos foram muito fracos, havendo modelos que eram piores que escolher aleatoriamente. Com a escolha de modelos mais robustos, e atualização dos próprios hiperparâmetros, surgiu um novo problema, o modelo só previa a moda. Esta dificuldade pode ter existido não só devido ao desequilíbrio das classes, como também devido à exigência que esta variável proporciona, usando apenas características dos jogadores e dos torneios para a prever.

Com um consenso dos membros do grupo, foi decidido apresentar um modelo que não previsse apenas a moda, chegando ao modelo que foi apresentando, analisado e comentado anteriormente ([Link](#)). Os resultados obtidos foram bastante impressionantes para o grupo, tendo em conta as limitações de *hardware* e de dados existentes, lembrando que se trata de um problema com dados reais.

Primeiramente, o grupo tinha grandes expectativas para as variáveis: *Finals* e *SumTrainer*, pois acreditou-se que nas finais os jogos costumam ser mais renhidos, como também jogadores com treinadores têm uma maior tendência para jogar melhor, pois possuem alguém que os corrige e dá suporte de forma direta. Acreditamos que o modelo não conseguiu captar tal informação, devido ao reduzido número de linhas existente na base de dados, contribuindo para que estas variáveis quase não tivessem importância.

De seguida, também é de grande interesse comentar 2 variáveis que surpreenderam o grupo de forma muito positiva, sendo essas: *DiffIMC* e *DiffRank*. Estas 2 variáveis foram as que tiveram um maior peso no modelo, demonstrando que o modelo deu uma maior importância à capacidade física dos atletas, como também à sua experiência no jogo.

Utilidade do Modelo

Relativamente ao porquê de se tentar prever o número de sets, podem existir muitas vantagens ao fazê-lo. Algumas dessas vantagens são:

- **Apostas desportivas e marketing:** Ao prever o número de *sets* numa partida, conseguimos ter uma influência parcial na tomada de decisões das pessoas que apostam, fazendo várias pessoas obterem grandes lucros. Ao saber a duração dos jogos, as empresas poderiam saber em quais jogos investir mais, aplicar o merchandising e publicidade



- **Análise e Estratégias Táticas:** Com base nas previsões de *sets*, vários jogadores e treinadores poderiam desenvolver técnicas e táticas para derrotar adversários específicos.
- **Transmissões e Comentários ao Vivo:** As previsões de número de *sets* podem adicionar um elemento adicional de emoção e engajamento durante a transmissão de partidas de ténis ao vivo. Os comentadores podem discutir as previsões, compará-las com o desempenho real dos jogadores e fornecer *insights* e análises aos telespectadores.
- **Planeamento de Eventos e Logística:** Em torneios de ténis, prever o número de *sets* tem uma influência direta no planeamento dos jogos, podendo usar essas previsões para determinar a duração destes, a fim de conseguir gerenciar melhor o calendário dos jogos.
- **Complementar informação:** Com esta previsão, será possível completar bases de dados que não possuem tal informação, sendo assim possível utilizar métricas para avaliar o desempenho dos jogadores, permitindo comparar o desempenho deles, como também identificar padrões e tendências.



Referências

Mendes, D., Moro, S., 2023, Moodle: Projeto Aplicado a Ciência de Dados I, scripts das semanas

Prezi para apresentações semanais

European open, Antwerp prize money | 2022 breakdown & historicals, 2023, 19 Janeiro, TennisCompanion, <https://tenniscompanion.org/prize-money/antwerp/>

ATP rankings|Pepperstone ATP rankings (Singles), ATP tour, Tennis,. <https://www.atptour.com/en/rankings/singles?rankRange=1-5000&countryCode=BEL&rankDate=2023-04-17>

Nytimes.com, 2022, The New York Times - Breaking News, US News, World News and Videos; <https://www.nytimes.com/2022/09/10/business/dealbook/tennis-is-a-failing-business.html>

IBM documentation. IBM - United States. <https://www.ibm.com/docs/en/spss-modeler/saas?topic=guide-business-understanding>

Tennis Explorer: Tennis Stats, WTA & ATP Tennis Betting. <https://www.tennisexplorer.com>

Plot types — Matplotlib 3.7.1 documentation, Matplotlib — Visualization with Python. https://matplotlib.org/stable/plot_types/index

Taxas de Câmbio, Banco de Portugal. <https://www.bportugal.pt/taxas-cambio?mlid=828>

Simulador: Calcular inflação em dólar, Clube dos Poupadores. <https://clubedospoupadores.com/simulador-inflacao-dolar>

Fancyimpute, PyPI. <https://pypi.org/project/fancyimpute/>

Terminology & procedures, News | USTA Mississippi. <https://mstennis.com/content/terminology-procedures>

Sklearn.ensemble.GradientBoostingClassifier, Scikit-learn.

<https://scikitlearn.org/stable/modules/generated/sklearn.ensemble.GradientBoostingClassifier.html>



Anexos

Modelo 0 - Introductório

Tabela 4 - Modelo 0

| Feature | Explicação do porquê da sua criação/utilização |
|--------------------|---|
| IMC Jogador | Procura-se classificar a aptidão física do jogador |
| IMC Oponente | Procura-se classificar a aptidão física do oponente |
| Belga Jogador | Fator casa deve influenciar a performance do jogador, aumentando a sua performance e, consequentemente, contribuir para jogos mais curtos |
| Belga Oponente | Fator casa deve influenciar a performance do oponente, aumentando a sua performance e, consequentemente, contribuir para jogos mais curtos |
| MainHand Jogador | Para diferentes jogadores, associados a diferentes mãos dominantes estão diferentes probabilidades em relação ao número de sets |
| MainHand Oponente | Para diferentes oponentes, associados a diferentes mãos dominantes estão diferentes probabilidades em relação ao número de sets |
| Idade Jogador | Para diferentes jogadores, a diferentes idades estão associadas diferentes probabilidades em relação ao número de sets (idade no jogo) |
| Idade Oponente | Para diferentes oponentes, a diferentes idades estão associadas diferentes probabilidades em relação ao número de sets (idade no jogo) |
| Treinador Jogador | Jogadores com treinador conhecido, em princípio, têm maior chance de participar em jogos com menor número de sets |
| Treinador Oponente | Oponentes com treinador conhecido, em princípio, têm maior chance de participar em jogos com menor número de sets |
| Rank Jogador | Quanto maior a diferença entre os ranks dos participantes (jogador e oponente), maior a chance de a partida ser mais curta |
| Rank Oponente | Quanto maior a diferença entre os ranks dos participantes (jogador e oponente), maior a chance de o jogo ser mais curto |
| Torneio Nome | Torneios diferentes têm diferentes probabilidades de total de sets, por partida |
| Ano Torneio | Torneios em datas diferentes podem estar associados a diferentes números de sets, por partida |
| Piso Numérico | A diferentes pisos estão associadas diferentes probabilidades em relação ao número de sets |
| Prémio | A jogos com prémios maiores estão associados jogadores com melhores rankings, o que deve proporcionar um jogo intenso e, grande parte das vezes, jogos com maior número de sets |
| Nome com Final | Jogos que são finais (quartos-de-final, meias-finais e finais, por exemplo) têm maior chance de serem jogos com maior número de sets |
| Número de Sets | Variável target |

Este foi um modelo apresentado ao longo das semanas e que continha a informação de todas as variáveis duplicadas, quer em jogador, quer em oponente. Por essa mesma razão e pelo medo de estarmos a sobrecarregar o modelo com informações repetidas, decidiu-se não analisar este modelo, mas fica informado de como foi feito o processo e que nos permitiu chegar às *features* que temos agora.

As suas métricas foram as seguintes, com divisão 70% treino e 30% teste:

- *Accuracy*: 0.65
- *Precision*: 0.85
- *Recall*: 0.70
- AUC: 0.55



Parecem ter valores consistentes face aos modelos realizados, mas tem os “crontas” de estarem a ser utilizadas variáveis repetidas.

Fórmulas utilizadas

$Accuracy = \frac{TP+TN}{TP+FN+FP+TN} = \frac{\text{diagonal principal}}{\text{total de obs.}}$: corresponde ao total de observações corretamente classificadas

$Precision = \frac{TP}{TP+FP}$: 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

$F1 - Score = \frac{TP}{TP + \frac{1}{2}(FP+FN)}$: média harmónica do *precision* e do *recall*, penalizando valores extremos de métricas

Sets à melhor de 5

Após fazermos a modelação para jogos à melhor de três, decidimos explorar um pouco os *sets* que foram à melhor de 5 na Bélgica separadamente, visto que foram muito poucos. Para este modelo foi utilizada uma partição de 70% de dados de treino e 30% de teste apenas por uma conveniência, resultando em apenas 45 observações no conjunto de teste. Antes de passar para o modelo, serão apresentadas as correlações das variáveis, a fim de demonstrar que estas eram muito fracas:

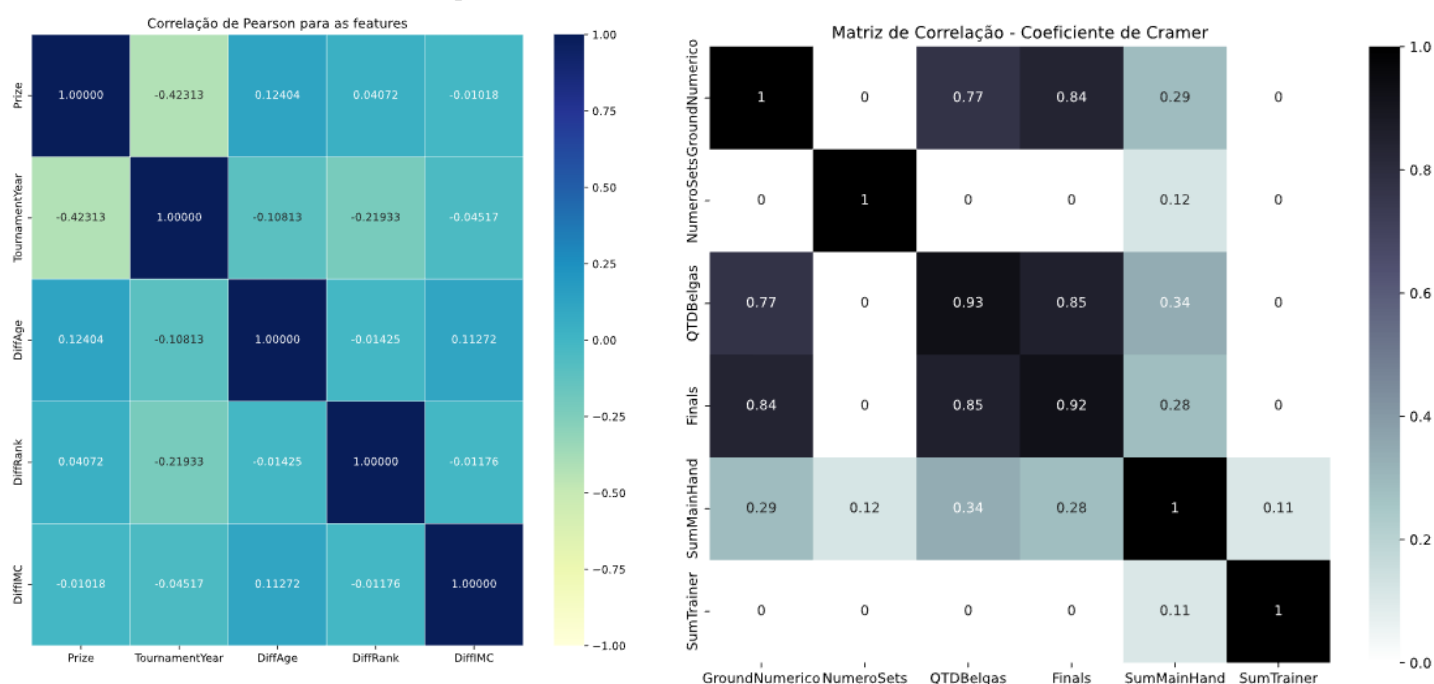


Figura 25 – Pearson e Coeficiente de Cramer



Coeficiente de Eta entre o NumeroSets e Prize: 0.06998289002575753
 Coeficiente de Eta entre o NumeroSets e TournamentYear: 0.08429682124000855
 Coeficiente de Eta entre o NumeroSets e DiffAge: 0.050787450018337
 Coeficiente de Eta entre o NumeroSets e DiffRank: 0.1858870899082243
 Coeficiente de Eta entre o NumeroSets e DiffIMC: 0.17890693161744534

Figura 26 - Coeficiente de ETA

Em seguida, foram realizados dois modelos de aprendizagem supervisionada, um de *Random Forest* e outro de *Gradient Boosting*, da biblioteca do *scikit-learn*. Os resultados e métricas utilizados foram a *accuracy*, *recall*, *precision* e AUC. De um modo geral, o modelo de *Random Forest* comportou-se melhor, atingindo um *accuracy* de cerca de 0.58 e um AUC de 0.67, mas com valores de *precision* e *recall* bastante críticos. Já o outro modelo, de *Gradient Boosting*, obteve piores métricas, mas também temos de considerar que é difícil de fazer previsão num modelo com tão poucas observações, daí não termos abordado muito profundamente esta previsão de *sets* à melhor de 5. Fica no relatório apenas de curiosidade.

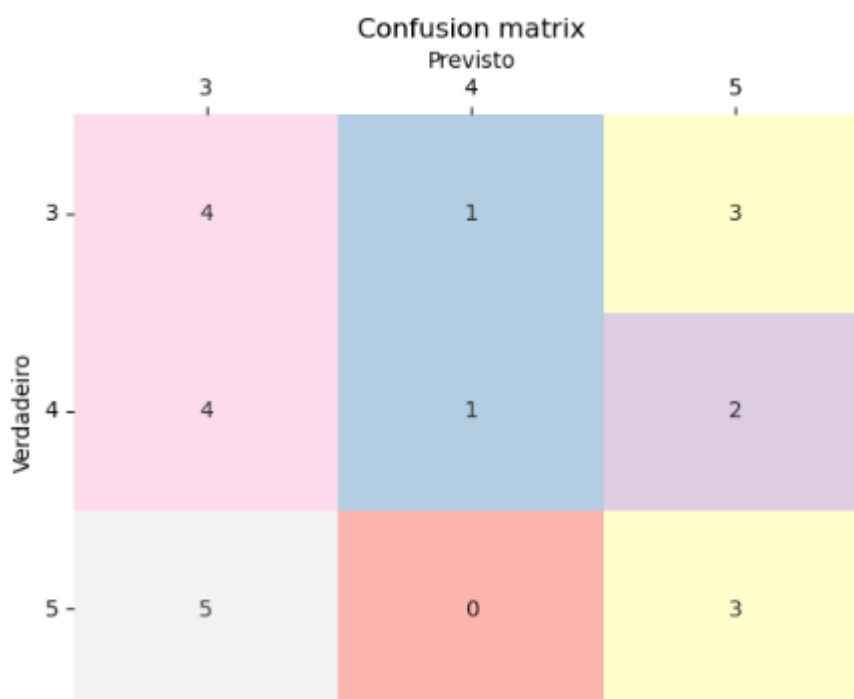


Figura 27 - Confusion Matrix melhor de 5