INSTITUTO SUPERIOR DE ENGENHARIA DO PORTO
MESTRADO EM ENGENHARIA INFORMÁTICA




Machine Learning para previsão de resultados de jogos de Ténis

EDUARDO FILIPE SANTOS NOGUEIRA
Outubro de 2019

POLITÉCNICO
DO PORTO

Instituto Superior de Engenharia do Porto

Machine Learning para previsão de resultados de jogos de Ténis

Eduardo Filipe Santos Nogueira

Dissertação para obtenção do Grau de Mestre em Engenharia Informática, Área de Especialização em Engenharia de Software

Orientador: Carlos Manuel Abreu Gomes Ferreira

Co-orientador: Maria de Fátima Coutinho Rodrigues

Porto, Outubro 2019

Resumo

Com o crescimento do mercado das apostas desportivas a nível mundial [1] e o facto de um ténis ser um dos desportos mais populares para os apostadores [2], cresce a necessidade da existência de plataformas que ajudem os apostadores na tomada de decisão. O principal objetivo deste projeto passa pela criação de um modelo de previsão baseado em machine learning que consiga prever resultados de jogos profissionais de ténis.

Uma plataforma para apostadores que disponibiliza previsões de forma automática, e tendo como base a análise de dados das últimas dezoito épocas desportivas, irá permitir aos apostadores pouparem tempo nas suas análises sem comprometerem os seus ganhos.

Existem alguns trabalhados desenvolvidos relacionados com a previsão de resultados de jogos de ténis, alguns destes utilizam modelos de machine learning e outros utilizam apenas técnicas de análise de dados históricos, os resultados obtidos nestes trabalhos variam entre os 62.6% e os 69.9% de taxa de acerto a prever o vencedor de um jogo de ténis.

A solução proposta é constituída por três componentes, o componente chamado Deuce Brain que é responsável pelo treino e teste do modelo de previsão, o componente chamado Deuce Services que é responsável por disponibilizar previsões através de um API, e por fim, o componente chamado Deuce Application que é uma aplicação web para disponibilização de previsões a apostadores.

Durante o projeto foram feitas algumas experiências, onde se testaram modelos treinados com diferentes conjuntos de variáveis e diferentes abordagens. Foram desenvolvidas cinco experiência com conjuntos de variáveis diferentes, e para cada uma dessas experiência foram testados um modelo de regressão logística, uma rede neuronal artificial e um modelo SVM (Support-vector machine).

O modelo que no final obteve maior taxa de acerto foi um modelo de regressão logística, com os rankings dos jogadores, a categoria do torneio e a superfície do court. Este modelo conseguiu uma taxa de acerto de 68%, e um retorno do investimento de 4.32% nos jogos do US Open de 2019. No geral, os modelos de regressão logística foram os mais precisos, seguidos das redes neuronais artificiais com taxas de acerto muito semelhantes, e por último os modelos SVM com um diferença significativa.

Palavras-Chave: *Machine Learning; Classificação; Aprendizagem Supervisionada; Ténis; Apostas; Regressão logística; ANN; SVM.*

Abstract

With the growth of the sports betting market worldwide [1] and the fact that tennis is one of the most popular sports for gamblers [2], there is a growing need for platforms that help gamblers in their decision making process. The main goal of this project is the development of a prediction model based in machine learning that can predict the result for professional tennis matches.

A platform for gamblers that provides automatic predictions based on data analysis from the last eighteen sports seasons, will allow the gambler to save time in their analysis without compromising their profits.

There are some academic works related to the prediction of tennis match results, some of them use machine learning models and the others only use historical data analysis techniques, the results for these studies range from 62.6% to 69.9% of accuracy predicting the winner of a tennis match.

The proposed solution is made up of three components, a component called Deuce Brain which is responsible for training and testing the prediction model, a component called Deuce Services which is responsible for providing predictions through an API, and finally a component called Deuce Application which is a web applications for making predictions available to gamblers.

During the project were developed some experiments that tested models with different sets of variables and different approaches. It was tested five experiments with different sets of variables, and for each of these experiments it was produced a logistic regression model, an artificial neural network and a SVM (Support-vector machine) model.

The model with highest accuracy was a logistic regression model with the players rankings, the tournament category and the court surface. This model achieved 68% of accuracy, and a return of investment of 4.32% during the US Open 2019. Overall, the logistic regression models were the most accurate, followed by the artificial neural networks with very similar accuracy, and lastly the SVM models with a significant difference.

Keywords: *Machine Learning; Classification; Supervised Learning; Tennis; Betting; Logistic Regression; ANN; SVM.*

Dedicatória

Gostaria de começar por agradecer à minha família pelo apoio, e por sempre me terem dado todas as condições para que conseguisse fazer o meu percurso académico para que pudesse ter um futuro melhor.

Agradecer também à minha namorada, por ter tido uma paciência inesgotável durante o tempo todo que despendi a desenvolver este projeto, e por todos os dias que tive de condicionar os nossos planos.

Ao professor Carlos Ferreira e à professora Fátima Rodrigues por todo o apoio prestado durante o projeto, e principalmente por todas as sugestões de melhorias que foram fazendo durante o mesmo.

Por fim, tenho também de agradecer ao Instituto de Engenharia do Porto por tudo o que aprendi ao longo destes anos, por todos os amigos que fiz durante este percurso, e por muitos momentos que nunca me esquecerei.

Índice

Lista de Figuras

Lista de Tabelas

Acrónimos e Símbolos

Lista de Acrónimos

NCD	*New Concept Development*
ATP	*Association of Tennis Professionals*
WTA	*Women's Tennis Association*
ML	*Machine Learning*
ANN	*Artificial Neural Network*
AWS	*Amazon Web Services*
CRISP-DM	*Cross Industry Standard Process for Data Mining*
SVM	*Support-vector Machine*
ROI	*Return on Invesment*
CSV	*Comma-separated values*
ORM	*Object-relational mapping*
CRUD	*Create, read, update and delete*
URL	*Uniform Resource Locator*
JSON	*JavaScript Object Notation*
SPA	*Single-page application*
SASS	*Syntactically awesome style sheets*
REST	*Representational state transfer*

Lista de Símbolos

£	Libra esterlina
€	Euro

1 Introdução

Neste capítulo é feito um enquadramento das áreas em que esta dissertação se insere, é apresentado o problema, são enumerados os objetivos definidos, é apresentada a abordagem utilizada, e por fim é apresentada a estrutura deste documento.

1.1 Enquadramento

O ténis é um dos desportos mais populares do mundo, como tal, em cada jogo que é realizado existe muito dinheiro a ser movimentado em apostas. Na ultima década existiu um aumento significativo no número de apostas em eventos de ténis, sendo neste momento o segundo desporto em que mais se aposta no mercado europeu, apenas superado pelo futebol [2]. Durante a final de Wimbledon em 2013 foram movimentados £48 milhões na betfair, a maior bolsa de apostas do mundo.

Com o crescimento mundial das apostas desportivas [1], plataformas que disponibilizam previsões de resultados são uma ferramenta muito utilizada pelos apostadores. Estas plataformas, na sua maioria, são alimentadas por previsões de apostadores experientes. O facto de as previsões serem adicionadas por pessoas é limitativo, visto que não é possível a uma pessoa analisar todos os jogos e todos os dados estatísticos disponíveis. Além disso, a plataforma está sujeita a outros fatores que podem viciar as previsões, tais como, a pouca diversificação de origem dos utilizadores, a popularidade ou impopularidade de alguns jogadores, etc.

Nos últimos anos as áreas de data mining e machine learning têm visto avanços significativos, hoje em dia técnicas e algoritmos destas áreas são aplicados na resolução de problemas complexos, tais como, condução autónoma, deteção de doenças, etc. O desporto devido à quantidade de dados disponível online é um tema recorrente para tentativas de implementação de trabalhos relacionados com essas áreas [3].

1.2 Problema

Um apostador tem de despender muito tempo a analisar dados estatísticos para conseguir tomar uma decisão sobre em que jogos apostar, e em que jogador apostar, isto porque todo este processo de análise da informação é totalmente manual. Também o facto de ser uma pessoa a analisar toda esta informação introduz alguns fatores de erro inerentes à natureza humana. Um modelo de previsão que consiga analisar esta quantidade de informação e ajudar apostadores a tomar a decisão sobre quais as apostas com mais valor, seria muito valorizado no sentido de diminuir o trabalho do apostador e também para obter maior precisão, melhorando o retorno financeiro.

1.3 Objetivos

O objetivo principal deste projeto é o desenvolvimento de um modelo de previsão, que permita prever o vencedor de um jogo profissional de ténis. Além deste objetivo principal, foram também definidos os seguintes objetivos:

- Modelo de previsão final com uma taxa de acerto igual ou superior a 65%.
- Modelo de previsão final com uma taxa de retorno do investimento (ROI) igual ou superior a 3%.
- Desenvolvimento de uma plataforma web que permita aos apostadores terem acesso às previsões.

1.4 Abordagem

A informação base utilizada neste projeto foi obtida a partir de um dataset desenvolvido por um software developer que é também analista de ténis Jeff Sackmann [4]. Este dataset foi posteriormente tratado e completado para poder ser utilizado no contexto deste projeto.

Com a informação tratada e organizada foram desenvolvidos vários modelos usando combinações de variáveis diferentes, estes modelos foram posteriormente analisados, avaliados e comparados de acordo com a sua taxa de acerto na previsão de resultados. Estes modelos foram desenvolvidos e analisados de acordo com a metodologia padrão Cross Industry Standard Process for Data Mining (CRISP-DM) [5]. Os modelos foram criados usando três abordagens diferentes, a regressão logística, uma rede neuronal e um modelo SVM.

1.5 Estrutura do documento

Este documento está dividido em sete capítulos: Introdução, Contexto e Estado da Arte, Soluções existentes, Design da Solução, Solução Implementada, Avaliação da Solução e Conclusão.

O corrente capítulo designado por introdução, apresenta uma visão geral da dissertação de mestrado apresentada neste documento.

No capítulo do Contexto e Estado da Arte é detalhado o contexto onde se insere este projeto, abordando algumas áreas tecnológicas de relevo para o mesmo e alguns conceitos de ténis e apostas. É também feita uma análise de valor a este projeto e, por fim, é feito um levantamento do estado da arte com apresentação de alguns modelos desenvolvidos e os seus respetivos resultados.

No capítulo das soluções existentes, são apresentadas e analisadas algumas soluções de problemas similares implementadas em outros projetos. Com esta análise foi possível identificar várias abordagens alternativas bem como os resultados das mesmas.

No capítulo de design da solução será apresentada a proposta de solução para o problema, esta proposta será apresentada tanto a nível conceptual como a nível técnico.

No capítulo da solução implementada é apresentada a solução técnica que foi desenvolvida durante o projeto.

No capítulo da avaliação da solução é descrito o processo que será utilizado para avaliar a solução desenvolvida ao longo do projeto, aqui serão identificadas as grandezas usadas na avaliação, os testes de hipóteses utilizados, e ainda as metodologias e testes estatísticos que serão implementados. Além disso, neste capítulo serão apresentados os resultados obtidos para as várias experiências e abordagens testadas neste projeto.

Por fim, no capítulo da conclusão é feito um resumo do trabalho desenvolvido e dos seus resultados, é também identificado o trabalho que deve ser realizado no futuro.

2 Contexto e Estado da Arte

Neste capítulo será feita uma contextualização do problema, apresentando alguns conceitos de negócio e conceitos técnicos relevantes para a compreensão do mesmo e da solução implementada. É feita também uma análise de valor, de maneira a identificar os principais benefícios que este projeto trará para o seu mercado alvo, e a identificar também alguns aspetos que serão importantes para o plano de negócios deste projeto.

2.1 Ténis

O ténis é um desporto praticado com raquetes que tem duas variantes, a variante de singles que consiste em jogos de um jogador contra outro, e a variante de doubles que consiste em jogos de um par de jogadores contra outro. Neste projeto o foco será na variante de singles por questões de maior facilidade de modelação do problema.

Durante os jogos que compõe o encontro um jogador é designado para servir e o seu adversário irá receber, servir é a designação que se dá à tacada para começar um ponto, essa tacada tem que enviar a bola para uma área restrita do campo que se encontra na diagonal oposta à posição do jogador que serviu. Os jogadores jogam em lados opostos do court, o court é uma área retangular com uma rede no meio a toda a largura do mesmo. Um court pode ter diferentes tipos de piso consoante o torneio disputado, esses tipos podem ser, relva, terra batida, ou superfície dura. No ténis um jogador perde o ponto quando não consegue devolver a bola para a área de jogo do seu adversário.

O sistema de pontuação do ténis é composto por três componentes, jogo, set e encontro. O jogo consiste numa sequência de pontos com o mesmo jogador a servir, o jogo é ganho pelo primeiro jogador a ter pelo menos quatro pontos, e ter mais dois pontos que o seu adversário. O set consiste numa sequência de jogos em que o jogador que serve vai variando, um jogador vence um set quando ganha pelo menos seis jogos, e tem pelo menos mais dois jogos ganhos do que o seu adversário. O encontro consiste numa sequência de sets e é jogado à melhor de três ou à melhor de cinco dependendo do torneio ou circuito.

Um dado interessante sobre o ténis é que as regras deste desporto quase não se alteraram desde 1890, o que permite o uso de um historial de informação estatística bastante alargado sabendo que os encontros analisados foram disputados sob as mesmas regras. [6]

2.2 Apostas Desportivas

Apostas desportivas são fundamentalmente uma tentativa de previsão de algo que possa acontecer durante um evento desportivo, as apostas têm associadas um risco e quanto mais alto esse risco maior serão os ganhos do apostador caso ganhe a aposta.

2.2.1 Odd

A odd de uma aposta é o valor que representa a probabilidade da mesma, e pode ser representada em três formatos diferentes, odds decimais (formato europeu), odds fracionárias (formato do reino unido) ou moneyline odds (formato americano) [7]. No desenvolvimento deste projeto, sempre que seja necessária a utilização de odds serão utilizadas no formato decimal. As odds decimais correspondem ao rácio que nos permite calcular o valor a receber a partir do valor que será apostado, por exemplo, uma aposta com uma odd de 2.0 significa que o valor que receberemos caso a aposta seja vencedora corresponde a duas vezes o valor que apostamos, ou seja, corresponde a um lucro igual ao valor que foi apostado.

2.2.2 Casas de Apostas

Existem dois tipos de casas de apostas em que podemos fazer apostas desportivas, as casas de apostas normais ou os mercados de apostas. Nas casas de apostas normais os apostadores jogam contra a própria casa, ou seja, a casa de apostas define as odds para cada aposta, e quando os apostadores ganham a casa de apostas perde dinheiro, e vice-versa. As odds das casas de apostas normais são definidas normalmente de acordo uma análise da probabilidade do acontecimento, ao valor obtido nessa análise é retirada uma margem de segurança para que a casa de apostas seja lucrativa.

Nos mercados de apostas desportivas os apostadores jogam entre si, nestes mercados não só é possível apostar num acontecimento mas também é possível apostar contra este mesmo acontecimento, desta maneira os apostadores apostam uns contra os outros, o que significa que quando um apostador ganha uma aposta outro apostador perdeu a aposta contra. Nos mercados de apostas, a casa ganha dinheiro cobrando comissões por transação ou por levantamentos.

2.3 Análise de Valor

Neste capítulo será realizada uma análise de valor à solução proposta neste projeto, nesta análise serão identificados os cinco elementos chave do modelo NCD (New concept development model), será apresentada a proposta de valor para o cliente, e por fim será analisado o modelo de negócio Canvas desenvolvido.

2.3.1 New concept development model

O modelo NCD providencia uma linguagem comum e uma visão sobre atividades de front end, este modelo divide-se em três áreas. A primeira área é o motor, o centro do modelo, que é composto por elementos como, a visão, a estratégia e a cultura. A segunda parte define cinco atividades de front end, e a última parte consiste em fatores externos que possam influenciar o negócio. Neste subcapítulo serão apenas abordadas as cinco atividades de front end que constituem a segunda parte deste modelo.

2.3.1.1 Identificação das oportunidades

Os resultados de eventos desportivos são objeto de estudo de muitas pessoas. Existem apostadores que conseguem seguindo alguns métodos de análise de informação de eventos passados ter ganhos consistentes com as suas previsões de resultados. Mas esta análise é feita manualmente e como tal é preciso muito tempo para analisar estas quantidades de informação. Com os avanços das áreas de data mining e machine learning é hoje possível em outras áreas substituir as pessoas neste trabalho moroso de analisar grandes quantidades de informação e decidir de acordo com essa informação. Um bom exemplo disso, é o mercado da bolsa de valores onde existem bastantes implementações de modelos de previsões, estes modelos através de informação histórica de uma determinada empresa e das previsões para o futuro da mesma decide se esta é considerada um bom investimento ou não.

Concluindo, projetos semelhantes a este já são hoje em dia utilizados em outras áreas com bastante sucesso. A aplicação à área do desporto e das apostas desportivas deverá também ser um sucesso tendo em conta a quantidade de apostadores que despendem bastante tempo a analisar dados estatísticos. No futuro, este tipo de solução pode também ser usada por casas de apostas para disponibilizar preços mais atrativos para apostadores, com o menor risco possível para a própria casa de apostas. Com isto, e verificando que não existe ainda um produto relevante no mercado, este pode ser o timing ideal para este projeto.

2.3.1.2 Análise das oportunidades

O principal mercado alvo deste sistema será o mercado das apostas desportivas, este mercado tem um valor estimado de $250 mil milhões, as suas receitas correspondem a 40% do total de

receitas do mercado de jogos de azar de todo o mundo, e a previsão é que haja um crescimento anual de 8.62% entre 2018 e 2022 [8].

Em termos de concorrência, o que foi encontrado durante a investigação deste projeto foram apenas plataformas que permitem a apostadores experientes partilhar as suas previsões com outros apostadores. Existem também algumas plataformas que alegam ter algoritmos avançados para previsões, mas nenhuma destas parece neste momento ter uma posição relevante no mercado.

2.3.1.3 Geração de ideias

As ideias que serão apresentadas resultam essencialmente das necessidades da comunidade de apostadores.

- Previsão de resultados – Como primeiro ponto e o mais importante deste projeto é fornecer aos apostadores de uma forma automática previsões de resultados precisas.
- Previsão vs Odd – O segundo ponto consiste em fornecer ao apostador uma comparação entre a probabilidade calculada pelo modelo de previsão e a odd oferecida pela casa de apostas, para que este possa verificar se a odd compensa em relação ao risco da aposta.
- Alertas – Dar a possibilidade ao apostador de personalizar alertas quando forem encontradas odds / previsões que correspondam aos seus critérios.
- Previsões ao vivo – Efetuar previsões e ajustes a previsões passadas tendo em conta o que está a acontecer durante o jogo.

2.3.1.4 Seleção de ideias

Das ideias enumeradas no subcapítulo anterior, para esta fase, foram selecionadas as duas primeiras, ou seja, o desenvolvimento de um modelo de previsão de resultados e a disponibilização dessas previsões, e ainda uma ferramenta que permita a comparação de odds de casas de apostas com as probabilidades das previsões. Estas ideias foram as selecionadas tendo em conta o risco tecnológico, os custos de desenvolvimento e o benefício que trazem para os apostadores. As restantes ideias ficarão como trabalho futuro a realizar após este projeto.

2.3.1.5 Definição do conceito

O principal objetivo deste projeto é o cálculo e disponibilização de previsões precisas de resultados de jogos de ténis profissional. O momento em que nos encontramos é propício ao

lançamento deste projeto visto que o mercado alvo tem vindo a crescer nos últimos anos e prevê-se que vá continuar a crescer nos próximos anos, além disso ainda não existe concorrência relevante. A solução implementada pode vir a ser muito importante para os apostadores tendo em conta que lhes pode poupar muito trabalho na análise de jogos, e pode ajudá-los a obter mais lucros nas suas apostas. Também há a possibilidade de esta solução vir a ser usada pelas próprias casas de apostas para melhorar a sua oferta de preços sem aumentar o risco.

2.3.2 Benefícios e sacrifícios para o cliente

Os benefícios para os apostadores prendem-se essencialmente pela redução de esforço de análise inerente à sua atividade sem que isso influencie negativamente os seus ganhos. Além disso, terão acesso a uma plataforma que lhe permitirá analisar o risco de uma aposta e se o valor da mesma (odd) compensa esse risco.

Os sacrifícios que os apostadores terão de fazer serão muito poucos, estes prendem-se essencialmente pela necessidade de despenderem algum tempo a aprender como utilizar a plataforma.

2.3.3 Proposta de Valor

Uma plataforma para apostadores que permite de uma maneira fácil aceder a previsões de resultados de jogos profissionais de ténis, calculadas por algoritmos avançados que analisam dados estatísticos de todos os jogos das últimas dezoito épocas desportivas. Esta plataforma irá permitir aos apostadores pouparem muito do seu tempo de análise de jogos sem comprometerem ou até mesmo aumentarem os seus ganhos.

2.3.4 Modelo Canvas

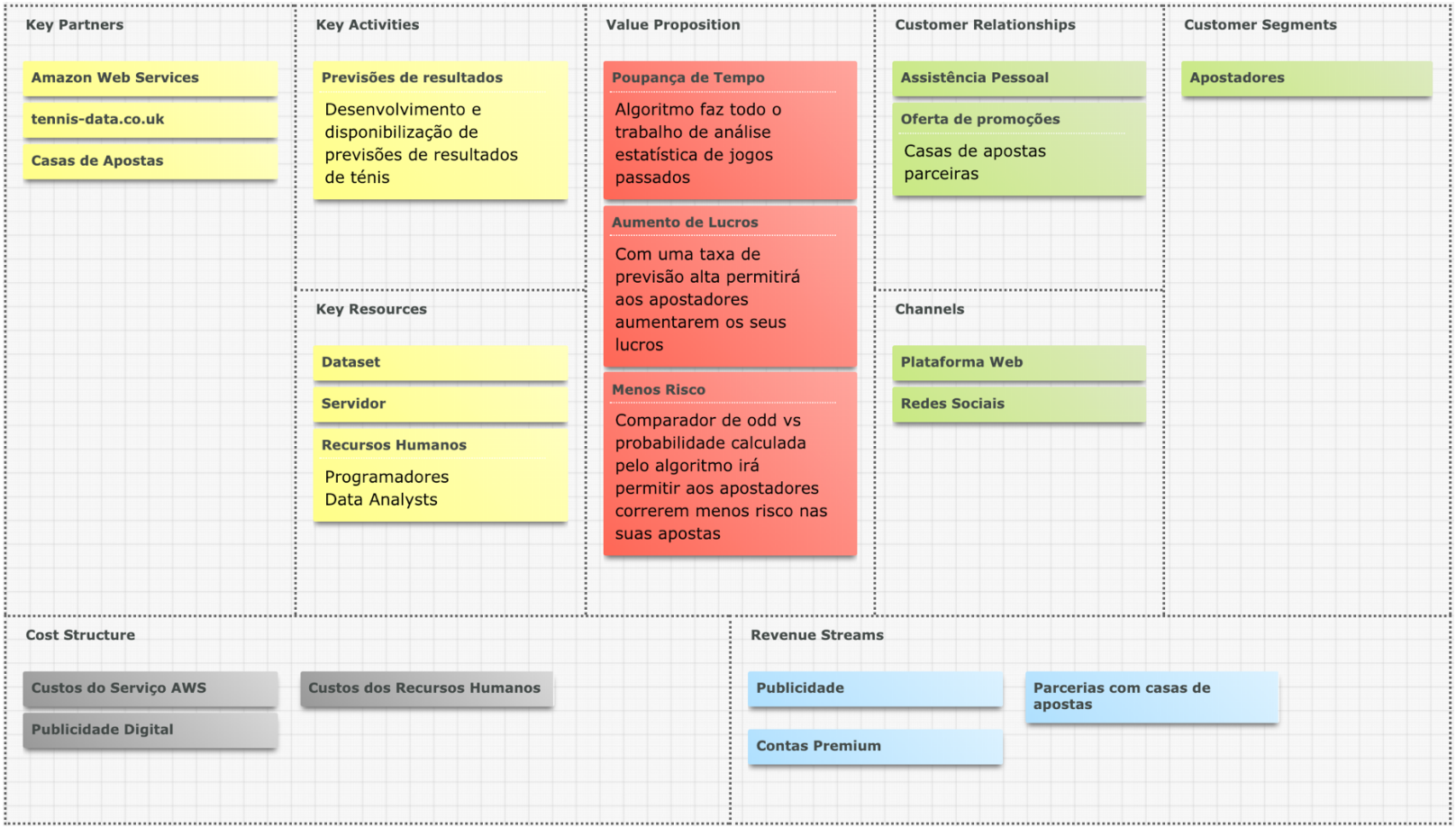


Figura 1 - Modelo Canvas

2.4 Machine Learning

A área de machine learning pretende dar resposta à questão de como desenvolver programas de computadores que melhoram automaticamente com a experiência [9]. Esta área é uma das áreas em maior crescimento na ultima década no setor tecnológico, e hoje em dia devido à mesma existem carros autónomos, motores de busca mais inteligentes, algoritmos de recomendação, etc.

2.4.1 Aprendizagem Supervisionada

Em machine learning existem três tipos de aprendizagem, a aprendizagem supervisionada, a aprendizagem não supervisionada e a aprendizagem semi-supervisionada. No contexto deste projeto iremos apenas explorar técnicas de aprendizagem supervisionada. A aprendizagem supervisionada (supervised learning) é uma forma de aprendizagem em machine learning, que é adequada para resolver problemas em que os dados nos permitem treinar um modelo, fornecendo variáveis de input e o output esperado para aquele conjunto de inputs. O modelo através da análise dos dados de treino rotulados com o respetivo output esperado, irá produzir uma função que irá ser utilizada para prever o output de novos exemplos de dados que não estão classificados. [10]

2.4.2 Feature Selection

É o processo através do qual se escolhe um subgrupo de atributos considerados relevantes, para serem usados na construção do modelo. Este processo tem um papel importante para a precisão do modelo, sendo que uma boa escolha desses atributos pode culminar num modelo mais preciso, mais simples e com melhor performance de treino. Neste processo devem ser identificados e removidos todos os atributos irrelevantes e redundantes da informação, de maneira a reduzir o ruido de informação e facilitar a aprendizagem do modelo.

A seleção por importância de uma feature é uma técnica de feature selection, que permite a partir de um modelo perceber quais são as variáveis com mais impacto no output [11]. Outra técnica utilizada para feature selection é a matriz de correlação, que é normalmente utilizada para analisar padrões de relações entre as variáveis do dataset [12].

2.4.3 Função sigmóide

A função sigmóide é uma função matemática quem tem como característica ter uma curva em forma de S. Um exemplo de uma função sigmóide bastante utilizada é a função logística [13], que é utilizada no contexto de machine learning em modelos de regressão logística. Um das principais razões para o uso desta função é porque esta toma valores entre 0 e 1. Portanto, é principalmente usada em problemas onde se quer prever a probabilidade de algo [14].

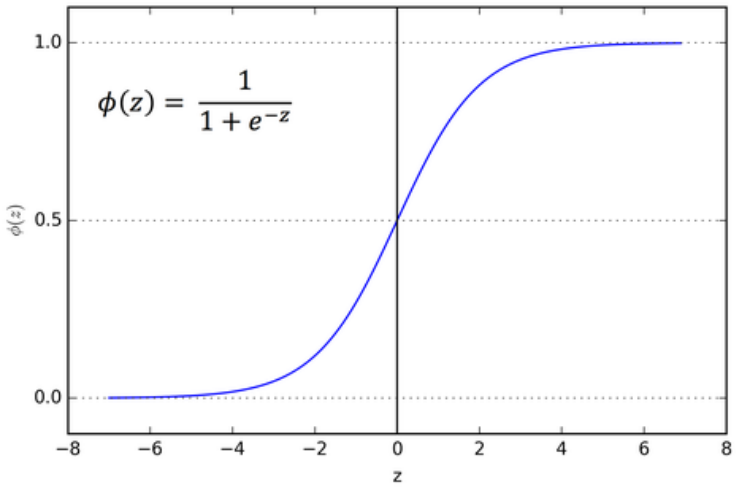


Figura 2 - Função sigmóide

Como podemos analisar na Figura 2, a função sigmóide aproxima-se de 0 quando Z tende para - ∞, aproxima-se de 1 quando Z tende para + ∞, e no ponto em que Z=0 o resultado desta função é 0.5. Esta função é um função limitada [15], diferenciável [16], e que para todos os valores reais de input tem uma derivada positiva.

2.4.4 Algoritmos de Classificação

Os algoritmos de classificação são usados para resolver problemas onde queremos identificar dentro de um conjunto finito de categorias, qual se adequa melhor ao exemplo analisado. Os algoritmos de classificação são usados no contexto de uma aprendizagem supervisionada, tendo dados de treino com as varáveis de input e o respetivo output, neste caso uma ou mais categorias.

2.4.4.1 Regressão Logística

Os algoritmos de regressão logística são apropriados para realizar uma análise de regressão quando a variável dependente toma valores binários (0 ou 1). Tal como outros tipos de regressão, a regressão logística é bastante utilizada para realizar previsões, mas ao contrário dos outros tipos, a regressão logística permite apenas obter previsões numa forma binária. O facto de o resultado ser um valor entre 0 e 1 torna estes algoritmos muito úteis quando se quer obter uma probabilidade, devido às probabilidades serem também um valor entre 0 e 1. Este modelo usa a função logística [13] que é uma função sigmóide daí a natureza binária do seu output.

No caso deste projeto a aplicação de regressão logística será viável, porque o que se quer é prever o vencedor de um jogo de ténis, portanto a previsão será sempre numa forma binária (jogador 1, jogador 2).

2.4.4.2 Redes neuronais artificiais

As redes neuronais artificiais são inspiradas nas redes neuronais biológicas, que permitem ao cérebro humano ao processar informação criar padrões que usa para tomar decisões. Uma rede neuronal é um sistema de neurónios interligados, cada neurónio é responsável por calcular um valor a partir do input que lhe foi passado, esse valor é depois passado como input para outros nerónios. ANNs são normalmente estruturadas com várias layers de neurónios, com um neurónio em cada layer conectado a todos os neurónios da layer anterior.

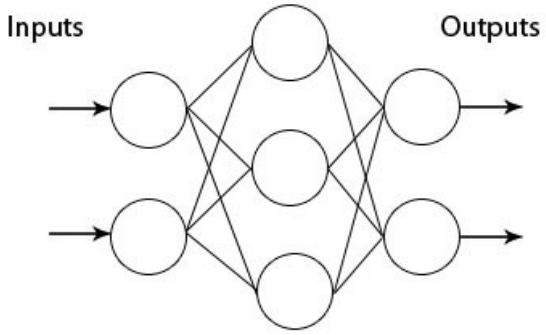


Figura 3 - Rede neuronal artificial

Na Figura 3 podemos analisar um exemplo de uma rede neuronal artificial, com uma layer de input, uma hidden layer e uma layer de output. Podemos também verificar que todos os neurónios estão conectados a todos os neurónios da layer anterior. As redes neuronais aprendem quais os cálculos que tem realizar em cada neurónio a partir da informação catalogada que lhe é passada, sempre que são adicionados mais exemplos os cálculos e valores guardados em cada neurónio são atualizados lentamente [17].

2.4.4.3 SVMs

Máquinas de vetores de suporte são modelos de aprendizagem supervisionada. Um SVM faz um mapeamento da informação que lhe é passada para pontos num espaço, e tenta encontrar um plano que separe as classes, de maneira a que a distância entre este plano e o ponto mais perto de cada uma das classes seja a maior possível [18]. Uma nova previsão pode depois ser colocada no espaço e classificada de acordo com o lado do plano em que esta é colocada.

2.4.5 Tecnologias

Nesta secção serão apresentadas algumas tecnologias que serão consideradas para as diversas componentes deste sistema. As tecnologias aqui apresentadas são de três áreas diferentes, devido a serem destinadas a componentes diferentes do sistema. Serão apresentadas tecnologias de machine learning, de backend e ainda de frontend.

2.4.5.1 Tensorflow

O TensorFlow [19] é uma framework open source, desenvolvida pela Google que permite realizar computação numérica usando grafos de fluxo de informação. Nestes grafos os nós representam operações matemáticas enquanto que as arestas representam matrizes de dados que circulam entre nós. O tensorflow pode ser usado diretamente ou pode ser usado em conjunto com outras bibliotecas que simplificam o processo de contruir um modelo. Um das principais vantagens do tensorflow é que oferece a possibilidade de correr um modelo de machine learning em quase qualquer máquina, inclusive dispositivos móveis.

2.4.5.2 PyTorch

O PyTorch [20] é uma biblioteca open source, que foi originalmente desenvolvida pela unidade de investigação de inteligência artificial do facebook. O Pytorch oferece funcionalidades similares ao tensorflow mas tenta diferenciar-se pela sua forte aceleração quando os seus tensors ou redes neuronais são treinadas com GPU.

2.4.5.3 Keras

O Keras [21] é uma API de alto nível para contruir e treinar modelos, que se destaca por ser amigável para o utilizador, modular e extensível. Em 2017, a equipa do tensorflow decidiu suportar esta API, o que fez com que a utilização das funcionalidades do tensorflow fosse facilitadada por uma API mais amigável para o utilizador, sem sacrificar a flexibilidade e performance.

2.4.5.4 Pandas

O Pandas [22] é uma biblioteca que dispõe de estruturas de dados rápidas e flexíveis, que faz com que trabalhar com datasets de informação seja mais fácil e intuitivo. Esta biblioteca facilita grande parte das operações sobre o dataset que terão de ser feitas ao longo deste projeto.

2.4.5.5 Django

O Django [23] é uma framework web em python, que tem como principais benefícios o facto de o desenvolvimento ser extremamente rápido, a incorporação por defeito de medidas de segurança standard e o facto de ser facilmente escalável.

2.4.5.6 Django REST framework

A Django REST framework [24] é uma biblioteca que deve ser usada em conjunto com a framework Django e que facilita a construção de apis REST. Esta biblioteca trás algumas funcionalidades úteis para apis REST já implementadas, tais como, politicas de autenticação, serialização a partir de um ORM, etc.

2.4.5.7 Flask

O Flask [25] é um framework web em python, que foi desenhada para ser fácil e rápido começar a desenvolver, mas com a habilidade de escalar para aplicações mais complexas. Esta framework trás menos funcionalidades incorporadas mas é muito fácil estender essas funcionalidades com bibliotecas desenvolvidas pela comunidade.

2.4.5.8 React

O React [26] é uma biblioteca javascript mantida pelo facebook, que tem como objetivo facilitar o desenvolvimento de interfaces de utilizador. No contexto do desenvolvimento de single page applications o React é uma das bibliotecas mais utilizadas pela comunidade.

2.4.5.9 Sass

O Sass [27] é uma linguagem de script que é interpretada e compilada para css, isto permite estender as funcionalidade do mesmo, adicionando por exemplo a capacidade de especificação de regras dentro de regras, em vez da repetição dos mesmos seletores como acontece no css.

2.4.5.10 Webpack

O Webpack [28] é um bundler de módulos javascript, a partir de um ponto de entrada definido por configuração ele percorre os imports, constrói um grafo de dependências para o projeto, e gera os recursos estáticos necessários.

2.4.5.11 Babel

O babel [29] é um compilador de javascript, que é normalmente utilizado para converter código javascript moderno (ES6+) em código compatível com versões anteriores de javascript. O babel permite a utilização de novas funcionalidades do javascript sem sacrificar o suporte de browsers mais antigos.

3 Soluções existentes

Neste capítulo serão abordadas algumas soluções implementadas para problemas semelhantes ao problema abordado neste projeto. A análise a estas abordagens consiste numa breve apresentação de como a solução funciona, e na apresentação dos resultados obtidos por esta solução.

3.1 Ranking based Models

Nos ranking based models os jogadores são indexados a um valor representativo da sua qualidade, e a previsão é feita tendo em conta apenas essa pontuação no ranking.

3.1.1 ATP ranking

No caso do ténis existem dois rankings bastante conhecidos, estes são o ranking ATP para jogadores masculinos e o ranking WTA para jogadores femininos. Os torneios ATP são categorizados de acordo com a sua importância (Grand Slam, ATP Finals, Masters 1000, etc), tendo em conta a categoria do torneio é atribuída uma pontuação a cada fase do mesmo (Ganhar, Final, Semi-Final, etc), o ranking ATP consiste apenas na atribuição dos pontos respetivos para a fase do torneio que o jogador conseguiu alcançar.

No trabalho desenvolvido por Marcus Nilsson a utilização apenas deste ranking para previsão de resultados, ou seja, prever que o jogador com mais pontos no ranking ATP será o vencedor do encontro, resultou numa taxa de acerto de 62.6% [30].

3.1.2 Elo ranking

O Elo ranking foi originalmente desenvolvido para classificar jogadores de xadrez, o elo é um valor numérico que varia dependendo dos resultados dos jogos disputados, neste ranking todos os jogadores começam com um elo de 1500, e o seu elo é atualizado depois de cada jogo que realizem. Se um jogador com menor elo que o seu adversário ganha o jogo então este jogador ganha mais pontos do que um jogador que ganhe a um adversário com menor elo do que o seu. No caso das derrotas o fundamento é o mesmo, se um jogador perde com um adversário com menor elo do que o seu, vai ser mais penalizado do que um jogador que perde com um adversário com maior elo do que o seu.

Stephanie Kovalchik comparou 11 modelos de previsão de resultados de ténis, e uma implementação de um modelo Elo ranking foi o mais preciso com 70% de taxa de acerto nos resultados dos jogos ATP em 2014 [31].

3.2 Regression based Models

Os modelos de regressão são utilizados quando existem dados estatísticos disponíveis, mas não existe uma relação precisa entre cada uma das variáveis e a sua influência no resultado final. Um modelo de regressão calcula uma estimativa da influência de cada variável no resultado final, e tendo em conta essas estimativas de influência e os valores que cada variável toma para um jogo, calcula as probabilidades dos resultados possíveis.

A taxa de acerto de modelos deste tipo está muito dependente da qualidade dos inputs fornecidos, os melhores modelos deste tipo apresentam resultados a rondar os 68% de taxa de acerto nos resultados dos jogos ATP em 2014 [31].

3.3 ANNs

Andre Cornman, Grant Spellman e Daniel Wright estudaram a possibilidade do uso de redes neuronais para prever resultados de jogos de ténis [32]. Para eles a taxa de acerto do modelo foi desapontante, tendo conseguido uma taxa de 65.2% utilizando validação cruzada, foi um dos modelos com menor taxa de acerto dos que foram estudados nesse trabalho. Apesar disto, no artigo é referido que era possível obter uma taxa ligeiramente maior, se continuassem a tentar melhorar os hiperparâmetros do modelo (tais como número de layers escondidas, nós por cada layer escondida, a função de ativação, etc).

3.4 SVM

Andre Cornman, Grant Spellman e Daniel Wright também estudaram a possibilidade do uso de SVMs para prever resultados de jogos de ténis [32]. Eles testaram três tipos de SVMs, o Gaussian kernel, o kernel polimonial de grau 3 e o kernel linear. Destes três tipos os dois primeiros conseguiram precisões muito baixas, 51% e 54% respetivamente, utilizando validação cruzada. Mas o modelo com kernel linear resultou numa taxa de acerto bastante boa, de 69.9%, utilizando também validação cruzada e superando assim as taxas de acerto das redes neuronais por larga margem.

4 Design da Solução

Neste capítulo será apresentada a análise do problema do ponto de vista técnico, será apresentada a solução que se pretende implementar. Na primeira secção será apresentada a arquitetura do sistema, nos subcapítulos seguintes serão apresentados alguns componentes do sistema e será feita um análise à escolha de tecnologias a utilizar.

4.1 Componentes

Neste subcapítulo serão apresentados os componentes que integram este sistema, juntamente com o divisão interna e a finalidade de cada um.

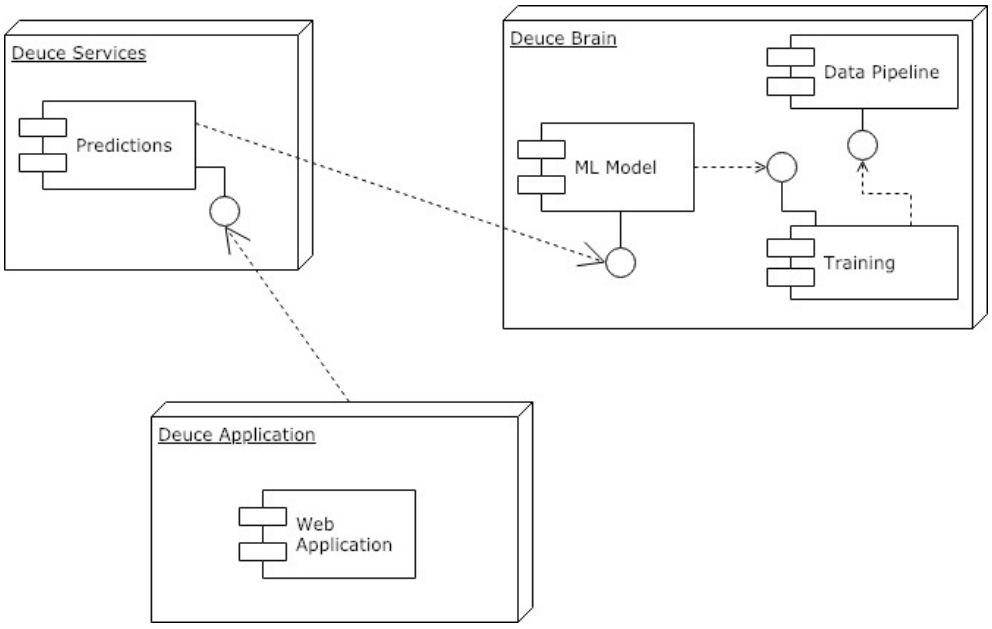


Figura 4 - Diagrama de Componentes

No diagrama de componentes podemos verificar que este sistema divide-se essencialmente em três componentes:

O componente deuce brain é sem dúvida o mais importante, neste componente está tudo o que está relacionado com o modelo de previsão de resultados e as suas fases. Este componente divide-se em três módulos, o primeiro módulo é a pipeline de dados que consiste numa série de scripts que carregam os dados de ficheiros, tratam esses dados, selecionam os dados necessários, e escrevem para ficheiros a informação tratada. O segundo módulo é o módulo de treino, este é responsável por carregar dados, utilizar esses dados para treinar o modelo, e verificar a taxa de acerto do mesmo. Por fim o módulo do modelo de previsão é responsável por para um determinado input de dados sobre um jogo, disponibilizar uma previsão do resultado do mesmo.

O segundo componente é os deuce services que é essencialmente um micro serviço que é responsável pela comunicação entre a aplicação web e o modelo de previsão. Este micro serviço recebe a informação sobre o jogo que o utilizador quer obter uma previsão, pede a previsão ao modelo de previsões e devolve para a aplicação web as previsões.

O último componente é a deuce application que é uma single page application onde serão disponibilizadas previsões para os próximos jogos de ténis aos apostadores.

4.2 Pipeline de dados

Para qualquer sistema de machine learning ter uma pipeline de dados eficaz e eficiente é extremamente importante, a pipeline de dados tem que garantir que o modelo de machine learning é treinado e testado com dados atualizados e verificados para que a precisão do modelo não possa ficar comprometida por informação desatualizada ou incorreta.

Figura 5 - Diagrama da pipeline de dados

Na Figura 5 podemos encontrar um diagrama da pipeline de dados que foi implementada para este sistema, esta pipeline é composta por quatro fases:

A primeira fase é a fase de pré-processamento e seleção dos dados, esta recebe como input os dados em bruto, obtidos do dataset original, e esses dados passam por uma série de scripts que tratam os dados de maneira a corrigir alguns erros e formatar alguns campos. O output destes scripts é uma série de ficheiros com os dados tratados e apenas com as características necessárias.

A segunda fase é a fase de carregamento em memória dos dados, esta recebe como input os ficheiros de dados tratados e carrega a informação em objetos em memória, de maneira a facilitar as operações que serão feitas com os mesmos.

A terceira fase é a fase de treino, esta recebe como input os objetos que foram carregados na fase anterior e divide-os em dois grupos, o grupo de dados de treino e o grupo de dados de teste. O grupo de dados de treino, serão os objetos que serão passados para o algoritmo de machine learning para que ele possa gerar um modelo representativo desta amostra. O output desta fase é um modelo de machine learning capaz de efetuar previsões de resultados.

A última fase desta pipeline consiste em usar o grupo de dados de teste gerados na fase anterior, testar o modelo de machine learning criado também nessa fase, e desta maneira obter a taxa de acerto do mesmo.

4.3 Escolha das tecnologias

Depois de feito o levantamento e análise dos componentes deste sistema e das suas responsabilidades é necessário escolher as tecnologias a utilizar. As tecnologias que foram consideradas para esta análise foram apresentadas na secção 2.4.5.

4.3.1 Deuce Brain

Com as tecnologias identificadas para construção e treino de modelos existem essencialmente três possibilidades. Estas são, a utilização apenas do tensorflow, a utilização do pytorch, ou ainda a utilização do tensorflow com a api do keras.

Devido à minha inexperiência com problemas de machine learning, um dos drivers desta decisão passava pela utilização de uma api amigável que abstrai-se alguma da complexidade deste processo. Além disso, foi também bastante valorizada a comunidade de maneira a que qualquer dificuldade seja mais facilmente ultrapassada com alguma pesquisa.

A decisão final recaiu sobre a utilização do tensorflow com a api do keras. Apesar da comunidade do tensorflow ser significativamente maior do que a do pytorch, a curva de aprendizagem é também mais íngreme [33]. Mas com a possibilidade de utilizar a api do keras que foi integrada no próprio tensorflow, torna-se mais fácil a utilização das funcionalidades do mesmo.

4.3.2 Deuce Services

Para que a integração entre código partilhado entre os serviços e o cérebro do sistema fosse facilitada, foi decidido que seria utilizada a mesma linguagem de programação, neste caso, o python. Esta decisão deixa-nos com essencialmente duas alternativas para as tecnologias para construção dos serviços, estas são, a utilização da framework Django com a biblioteca Django REST framework, ou a utilização de Flask.

A decisão final recaiu sobre a utilização da framework Django com a biblioteca Django REST framework, essencialmente devido à minha experiência com essa framework e ao facto desta já incluir algumas funcionalidades que serão necessárias, e que o Flask não inclui por default. Assim, esta decisão irá permitir o desenvolvimento dos serviços num tempo mais reduzido o que libertará mais tempo para o trabalho no modelo de previsão.

4.3.3 Deuce Application

A aplicação web não é o principal objetivo deste projeto, como tal, não foram consideradas alternativas e optei por escolher a stack técnica em que já tenho experiência no desenvolvimento deste tipo de aplicações.

As tecnologias escolhidas foram o React como framework de desenvolvimento da interface de utilizador, o webpack para fazer bundling da aplicação, o babel para compilar o código javascript, e sass para a escrita das folhas de estilo. Este conjunto de tecnologias é basicamente o standard hoje em dia para o desenvolvimento de uma single page applications.

5 Solução implementada

Neste capítulo será detalhada a solução que foi implementada. Este capítulo está divido por componentes que constituem a solução final, em cada um deles será abordado o seu funcionamento, as decisões tomadas, as razões que levaram a essas decisões, e ainda possíveis melhorias. Aqui será abordado o conjunto de dados base, será apresentada a base de dados do sistema, será analisada a pipeline de dados implementada, será também analisada a implementação do módulo de treino dos modelos, serão apresentadas as experiências que foram realizadas, e por fim, apresentada a implementação dos serviços e da aplicação web.

5.1 Dataset

O ténis dispõe de uma grande quantidade de informação estatística de várias épocas disponível online. O dataset base usado no desenvolvimento deste projeto está disponível no GitHub [4] e inclui todos os jogos de torneios ATP desde 1968 até setembro de 2018. A partir do ano 2000 o dataset passa a incluir mais informação, e esta informação é mais viável, pelo que neste projeto são utilizados os dados disponíveis a partir desse mesmo ano. O dataset é composto pelas entidades descritas nas próximas subsecções.

5.1.1 Torneio

Tabela 1 - Dados relativos a torneios

ID	Nome	Tipo de Piso	Número de Jogadores	Importância	Data
2016-520	Roland Garros	Clay	128	G	20160523
2016-M006	Indian Wells Masters	Hard	128	M	20160307

Na tabela 1 podemos observar dois exemplos de entradas no dataset de torneios, um torneio é caraterizado pelo seu identificador, o nome, o tipo de piso do court (distribuição no anexo 6 figura 18), o número de jogadores que disputam o torneio (distribuição no anexo 6 figura 19), a importância ou categoria do torneio (distribuição no anexo 6 figura 20), e por fim a data de inicio do torneio. O atributo tipo de piso pode tomar 4 valores diferentes, Hard para pisos duros por exemplo cimento, Clay para pisos em terra batida, Grass para pisos em relva, e Carpet para pisos em borracha ou superfícies têxteis. A importância pode tomar 5 valores diferentes, G para Grand Slams, F para ATP finals, M para Masters, A para ATP Tour 250 e D para ATP Tour 500.

5.1.2 Jogo

Tabela 2 - Dados relativos a jogos

ID	Vencedor	Derrotado	Melhor de X sets
100	104925	104229	5
127	102856	102796	3

Na tabela 2 podemos observar dois exemplos de entradas no dataset de jogos disputados, um jogo é caracterizado pelo seu identificador, o identificador do jogador vencedor, o identificador do jogador derrotado e um atributo que diz à melhor de quantos sets é que o jogo é disputado (distribuição no anexo 7 figura 21). O atributo melhor de x sets pode tomar dois valores possíveis, 5 para jogos que são disputados à melhor de 5 sets, ou 3 para jogos que são disputados à melhor de 3 sets.

5.1.3 Jogador

Tabela 3 - Dados relativos a jogadores

ID	Primeiro Nome	Último Nome	Mão de jogo	Data de Nascimento	Nacionalidade
103819	Roger	Federer	R	19810808	SUI
104745	Rafael	Nadal	L	19860603	ESP

Na tabela 3 podemos observar dois exemplos de entradas no dataset de jogadores, um jogador é caracterizado pelo seu identificador, o seu primeiro e último nome, a sua mão de jogo (distribuição no anexo 8 figura 22), a sua data de nascimento e a sua nacionalidade. O atributo mão de jogo pode tomar três valores diferentes, L para jogadores que preferem jogar com a mão esquerda, R para jogadores que preferem jogar com a mão direita, ou U para jogadores que não têm uma mão de jogo preferida.

5.1.4 Ranking

Tabela 4 - Dados relativos a rankings

Data	Posição	Jogador	Pontos
20180806	1	104745	9310
20180806	2	103819	7080

Na tabela 4 podemos observar dois exemplos de entradas no dataset de rankings, um ranking é caracterizado pela sua data, a posição do jogador no ranking, o identificador do jogador e o número de pontos que o jogador têm no ranking.

5.1.5 Análise de atributos chave

Nesta secção é feita uma análise de alguns atributos existentes neste dataset, que foram considerados chave para serem incluídos no modelo de previsão. Estes atributos foram selecionados tendo em conta o desempenho de modelos desenvolvidos que incluem estes atributos.

5.1.5.1 Rankings

Os rankings são a melhor maneira de traduzir a qualidade de um jogador num valor numérico, eles são um indicador bastante preciso de como tem sido a performance desportiva de um determinado jogador no último ano.

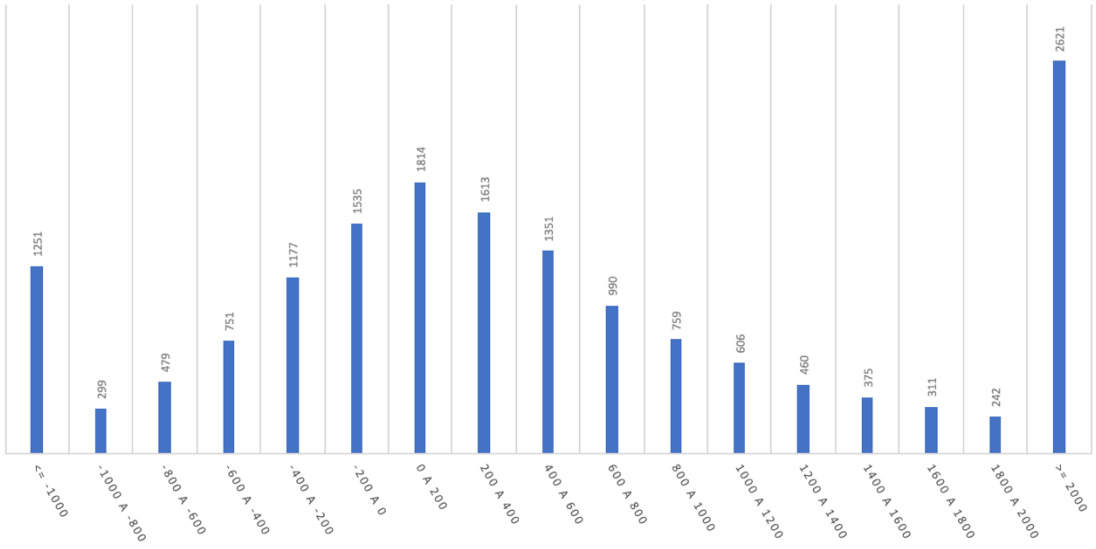


Figura 6 - Diferenças de ranking entre vencedor e vencido

Na Figura 6 é apresentada uma amostra de 16634 jogos, para os quais foi calculada a diferença de pontos dos rankings dos jogadores, e estas diferenças foram agrupadas pelas escalas

definidas abaixo do gráfico. A diferença dos rankings é calculada subtraindo ao valor do ranking do jogador vencedor o ranking do jogador vencido, como tal, os valores negativos correspondem a um resultado surpresa e os valores positivos correspondem a um resultado mais expectável. Dos 16634 jogos, 5492 jogos foram ganhos pelo jogador com menos pontos no ranking e 11142 foram ganhos pelo jogador com mais pontos no ranking.

5.1.5.2 Categoria do Torneio

A categoria do torneio é útil para perceber a importância do mesmo na época dos jogadores. Jogadores de níveis diferentes têm normalmente objetivos para a época diferentes, e o facto de um torneio ser importante ou não para os objetivos de um jogador, pode afetar a sua performance desportiva no mesmo.

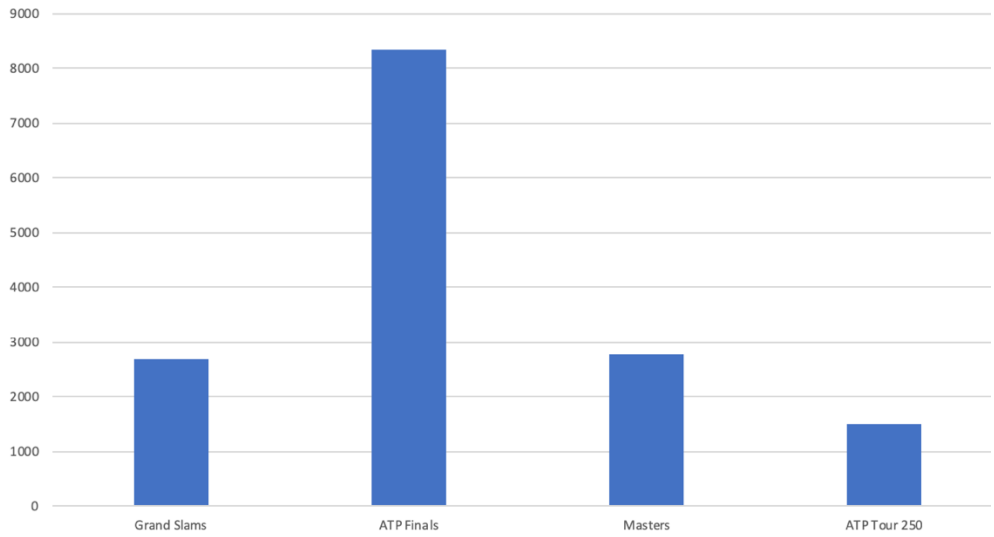


Figura 7 - Média do ranking de vencedores em função da categoria do torneio

Na Figura 7 podemos analisar o valor médio da pontuação dos jogadores que venceram jogos disputados em torneios de uma determinada categoria. Como podemos facilmente observar pelo gráfico existe uma relação entre a qualidade do torneio e a qualidade dos jogadores que o disputam, exemplo disso é a diferença entre a categoria ATP Tour 250 e as restantes categorias que são consideradas as três mais importantes do circuito mundial de ténis. As categorias Masters e Grand Slams verificam valores muito aproximados, porque são normalmente disputados pela mesma base de jogadores, os jogadores de topo mundial, isto deve-se essencialmente ao facto de os Grand Slams serem apenas quatro e haver espaço para bastantes mais torneios na época de um jogador profissional. A categoria ATP Finals, é apenas um torneio que é disputado pelos oito melhores jogadores do ano, daí a diferença bastante acentuada para as outras categorias.

5.2 Base de dados

Neste projeto foi usada uma base de dados relacional (PostgreSQL), nesta base de dados são guardados todos os dados utilizados para treinar e testar os diversos modelos de previsão.

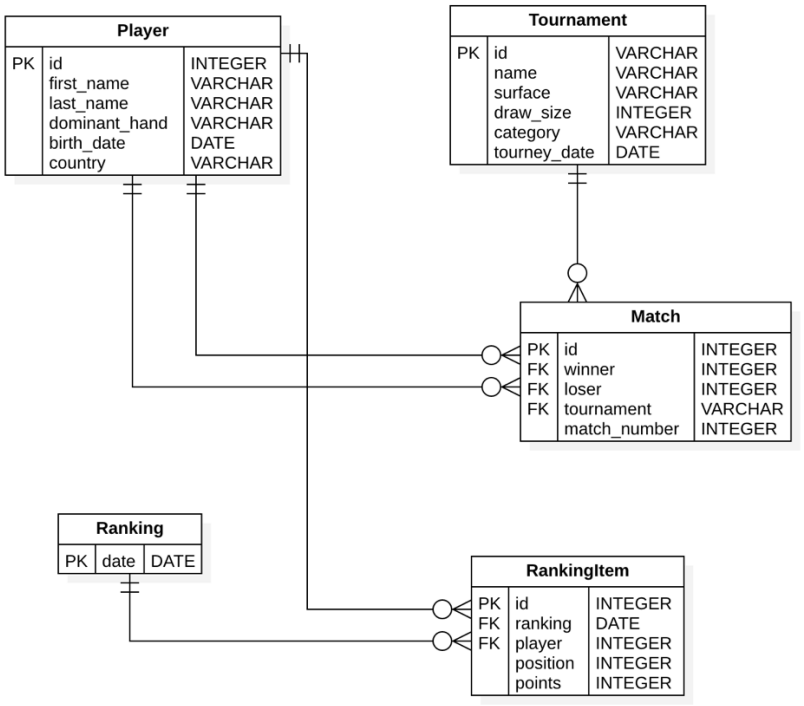


Figura 8 - Modelo de dados

A estrutura da base de dados é a seguinte:

- Player: Tabela que representa os jogadores, estes são representados pelo seu nome, mão dominante, data de nascimento e nacionalidade.
- Tournament: Tabela que representa os torneios, estes são representados pelo seu nome, superfície do court, número de participantes, categoria e data de início.
- Ranking: Tabela que representa o ranking ATP de uma determinada data.
- RankingItem: Tabela que representa cada linha dos vários rankings ATP, estas são representadas pelas chaves estrangeiras para o ranking e o jogador correspondentes, e ainda pela posição e pontos.
- Match: Tabela que representa os jogos, estes são representados pelas chaves estrangeiras para o jogador vencedor, jogador derrotado, torneio correspondente e ainda pelo número do jogo no torneio em que se insere.

5.3 Pipeline de Dados

Neste módulo residem vários scripts de importação e exportação de dados, que permitem ler informação do dataset inicial, ou então exportar ficheiros csv com a informação necessária para treinar e testar um determinado modelo.

5.3.1 Importação de dados

Este módulo do sistema é responsável por ler os dados do dataset inicial, tratá-los e carregar toda essa informação para a base de dados do sistema. O módulo é constituído essencialmente por quatro scripts, cada um destes scripts é responsável por importar os dados referentes a uma determinada entidade (jogadores, torneios, rankings ou jogos). O processo de importação começa com a leitura de um ou mais ficheiros csv, é feito o tratamento dos dados que não se encontram no formato expectável, de seguida esses dados são carregados em objetos em memória, e por fim esses objetos são armazenados na base de dados.

5.3.2 Exportação de dados

Este módulo do sistema é responsável por exportar informação da base de dados para ficheiros csv, que serão utilizados para treinar e testar modelos de previsão. O módulo é constituído por cinco scripts, cada um destes scripts é responsável por exportar os dados necessários para uma determinada experiência, sendo que uma experiência é neste sistema a geração de um modelo a partir de dados diferentes dos usados nas outras experiências. O processo de exportação começa com a identificação da experiência para a qual queremos exportar os dados, os dados são carregados da base de dados para objetos em memória, e por fim é feita a escrita de um ficheiro csv com a informação relevante para essa experiência.

5.4 Módulo de treino

Este módulo é responsável por usar os ficheiros csv que foram exportados, para treinar e testar os modelos das várias experiências realizadas. Para cada experiência são treinados e testados três modelos de previsão baseados em abordagens diferentes. Os scripts responsáveis por treinar e testar cada uma das abordagens têm em comum os passos apresentados na Figura 9.

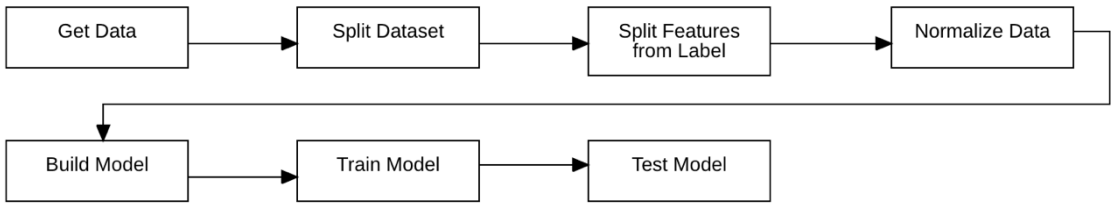


Figura 9 - Fluxo de scripts de treino e teste de modelos

- Get data: Leitura do ficheiro csv respetivo, utilizando a biblioteca pandas.

- Split dataset: O dataset é dividido entre dados para treino e dados para teste, é utilizada uma distribuição de 80% para treino e os restantes 20% para teste, esta divisão foi escolhida por ser a que proporcionou melhores resultados nas experiências realizadas.

- Split features from label: O dataset de treino é divido em dois, um com todas as variáveis necessárias para treinar o modelo, e o outro apenas com o resultado esperado.

- Data normalization: Nesta fase os dados do dataset são transformados para obedecerem à mesma escala, este processo consiste em retirar aos valores a média de todos os restantes valores da mesma coluna, e dividir o resultado pelo desvio padrão dos valores dessa coluna.

- Build model: Esta fase é diferente em cada um dos scripts dependo da abordagem que queremos testar. Aqui é utilizado o tensorflow com a API do keras para construir os modelos.

- Train model: O dataset de treino normalizado e o dataset de resultados esperados são utilizados para treinar o modelo construído no passo anterior. O treino termina quando houver um intervalo de 20 épocas em que o modelo não melhore a taxa de acerto.

- Test model: O dataset de teste normalizado é utilizado para validar a taxa de acerto do modelo treinado no passo anterior.

5.4.1 Regressão Logística

Este módulo treina e testa um modelo de regressão logística a partir de um dataset lido de um ficheiro csv. O módulo segue a estrutura apresentada anteriormente, nesta secção serão abordadas as principais funções utilizadas pelo modelo.

5.4.1.1 Função de ativação

Como função de ativação é usada a função sigmóide, uma das principais características desta função reside no seu resultado que apenas pode tomar valores entre 0 e 1. Esta função é por isso muito utilizada em modelos onde se pretende obter probabilidades, tendo em conta que a probabilidade de uma determinada coisa acontecer pode sempre ser representada por um valor entre 0 e 1.

5.4.1.2 Função de perda

A função de perda utilizada neste modelo é a função entropia cruzada binária, esta função é utilizada em problemas que envolvem decisões de sim ou não. Neste caso, podemos ver o problema como sendo "O jogador 1 vai vencer?" em que uma previsão de 1.0 seria o máximo de probabilidade do jogador 1 vencer, e 0.0 seria o máximo de probabilidade de o jogador 1 não vencer, ou seja, o jogador 2 vencer.

5.4.1.3 Método de otimização

Para otimização do modelo é usado o método gradiente descendente estocástico. O método gradiente descendente é utilizado para obter o mínimo de uma função progredindo iterativamente na direção da descida mais íngreme, que é indicada pelo negativo do gradiente [34]. Em cada passo desta otimização, os pesos das variáveis do modelo são ajustados, de modo a reduzir o resultado da função de custo. O método gradiente descendente estocástico é uma variante do método gradiente descendente, que usa algumas amostras do dataset escolhidas aleatoriamente para avaliar os gradientes, em vez de usar o dataset completo.

5.4.2 ANN

Este módulo treina e testa uma rede neuronal artificial a partir de um dataset lido de um ficheiro csv. O módulo segue também a mesma estrutura do módulo apresentado anteriormente. Nesta secção serão abordadas as principais funções utilizadas pelo modelo.

5.4.2.1 Camadas

A rede neuronal é constituída por três camadas, o número de neurónios na primeira camada é igual ao número de variáveis utilizadas para treinar o modelo. A última camada tem apenas um neurónio, o output desse neurónio será a previsão. Durante os testes realizados verificou-se que a introdução de uma camada intermédia com metade dos neurónios da primeira, aumentava ligeiramente a precisão do modelo.

5.4.2.2 Função de ativação

Nas primeiras duas camadas da rede neuronal é utilizada a função linear retificada, esta função consiste em simplesmente retornar o input caso este seja superior a 0, ou retornar 0 caso o input seja negativo ou igual a 0. Na última camada é utilizada a função sigmóide de maneira a garantir que o output desta será um valor entre 0 e 1.

5.4.2.3 Função de perda

A função de perda usada neste modelo é também a função entropia cruzada binária.

5.4.2.4 Método de otimização

Para otimização do modelo é usado o método adam, este método é uma extensão do gradiente descendente estocástico que acrescenta a possibilidade de haver taxas de aprendizagem diferentes para cada uma das variáveis do modelo, tornando o treino mais eficiente.

5.4.3 SVM

Este módulo treina e testa um modelo baseado em máquinas de vetores de suporte, a partir de um dataset lido de um ficheiro csv. O módulo segue também a mesma estrutura dos dois módulos apresentados anteriormente. Nesta secção serão abordadas as principais funções utilizadas por este modelo.

5.4.3.1 Função de ativação

Em termos das funções de ativação este modelo é muito semelhante ao anterior, a única diferença é que neste modelo temos apenas duas camadas, portanto na primeira utilizamos a função linear retificada e na última camada utilizamos a função sigmóide.

5.4.3.2 Função de perda

A função de perda utilizada neste modelo é a função hinge, esta função usa uma margem para calcular se deve penalizar uma previsão ou não, ou seja, mesmo que o modelo acerte numa previsão, este pode ser penalizado no calculo da perda se o valor dessa previsão (grau de certeza) for inferior à margem que foi definida.

5.4.3.3 Método de otimização

O método de otimização utilizado neste modelo foi o método adadelta, que é uma extensão do método adagrad, que é por sua vez um algoritmo para otimizações baseadas em gradientes, mas que adapta a taxa de aprendizagem aos parâmetros, fazendo pequenas atualizações para parâmetros que tomam valores muito recorrentes, e atualizações grandes para parâmetros que tomam valores muito diversificados. O método adadelta é mais eficiente que o adagrad porque em vez de guardar todos os gradientes quadrados, a soma dos gradientes é definida pela média decadente desses mesmos gradientes quadrados.

5.5 Experiências

Nesta secção serão descritas as várias experiências de modelos desenvolvidos durante este projeto. Consideramos uma experiência como sendo a criação de um modelo com variáveis diferentes dos outros já criados, de maneira a tentar melhorar a taxa de acerto do mesmo.

5.5.1 Experiência 1

Para a experiência 1 o objetivo foi fazer um modelo simples apenas com as variáveis consideradas serem as mais importantes para a previsão.

Tabela 5 - Variáveis usadas para treinar o modelo da experiência 1

Ranking Player 1
Ranking Player 2
Result (0 or 1)

Como podemos observar na Tabela 5, para este modelo foram apenas utilizados os rankings (pontos do ranking à data do jogo) dos dois jogadores que disputam o jogo. O ranking dos jogadores é sem dúvida a variável que mais influencia o resultado final do jogo, aplicando apenas a estratégia de escolher o jogador com ranking superior podemos conseguir precisões a rondar os 62.6% [35]. No anexo 1 foi colocado um gráfico onde podemos analisar a distribuição dos atributos desta experiência uns em função dos outros.

5.5.2 Experiência 2

Para a experiência 2 o objetivo foi tentar adicionar variáveis que em conjunto com os rankings usados na experiência anterior melhorassem a taxa de acerto do modelo.

Tabela 6 - Variáveis usadas para treinar o modelo da experiência 2

Ranking Player 1
Ranking Player 2
Tournament Category
Result (0 or 1)

Como podemos observar na Tabela 6, a variável adicionada neste modelo em relação ao anterior foi a categoria do torneio. A ideia aqui seria o modelo perceber o contexto onde o jogo se insere, muitas vezes jogadores de topo fazem participações em torneios menos prestigiados e a sua performance não é a esperada. Isto acontece muitas vezes, porque os jogadores de topo vão a torneios menos prestigiados apenas para ganharem forma, e a vitória nestes torneios não é um objetivo crucial para a suas temporadas. No anexo 2 foi colocado uma gráfico onde podemos analisar a distribuição dos atributos desta experiência uns em função dos outros.

5.5.3 Experiência 3

Para a experiência 3 o objetivo foi dar a conhecer ao modelo alguns atributos dos jogadores que disputam o jogo, de maneira a este tentar relacionar as características dos jogadores com o resultado dos jogos.

Tabela 7 - Variáveis usadas para treinar o modelo da experiência 3

Ranking Player 1
Hand Player 1
Age Player 1
Ranking Player 2
Hand Player 2
Age Player 2
Tournament Category
Result (0 or 1)

Como podemos observar na Tabela 7, para cada um dos jogadores foram adicionadas duas variáveis que representam características dos mesmos. As variáveis adicionadas foram, a mão com que o jogador prefere jogar, e a idade do mesmo. A ideia de incluir a mão preferida do jogador resulta da possível (mas não comprovada) vantagem que os jogadores esquerdinos podem ter por serem a minoria no circuito internacional, e os restantes jogadores estarem mais habituados a jogar contra destros. Por fim, a ideia de incluir a idade dos jogadores reside essencialmente na possibilidade de haver algum padrão em jogos em que a idade dos jogadores seja muito díspar, podendo haver algum benefício em experiência versos capacidade física ou vice versa. No anexo 3 foi colocado uma gráfico onde podemos analisar a distribuição dos atributos desta experiência uns em função dos outros.

5.5.4 Experiência 4

Para a experiência 4 o objetivo foi pegar no modelo da experiência 2, e tentar adicionar algum tipo de informação que permitisse ter uma noção da performance dos jogadores nos jogos imediatamente anteriores ao momento atual.

Tabela 8 - Variáveis usadas para treinar o modelo da experiência 4

Ranking Player 1
Number of wins (last 10 games) Player 1
Ranking Player 2
Number of wins (last 10 games) Player 2
Tournament Category
Result (0 or 1)

Como podemos observar na Tabela 8, em relação ao modelo da experiência 2 foi adicionado um novo campo para cada um dos jogadores. Este novo campo, é essencialmente a contagem do número de vitórias que o jogador conseguiu alcançar nos últimos dez jogos. Isto permite ao modelo ter uma noção da forma que os jogadores atravessam. Além do intervalo dos dez últimos jogos foi também testado um intervalo de cinco jogos e três jogos, recaindo a escolha sobre os dez últimos jogos devido aos resultados ligeiramente superiores. No anexo 4 foi colocado uma gráfico onde podemos analisar a distribuição dos atributos desta experiência uns em função dos outros.

5.5.5 Experiência 5

Para a experiência 5 o objetivo passou por adicionar ao modelo da experiência 2 informação sobre a superfície do court de ténis onde o jogo é disputado.

Tabela 9 - Variáveis usadas para treinar o modelo da experiência 5

Ranking Player 1
Ranking Player 2
Tournament Category
Surface
Result (0 or 1)

Como podemos observar na Tabela 9, em relação ao modelo da experiência 2 foi adicionado o campo da superfície do court onde o jogo é disputado. As diferentes superfícies de courts têm características diferentes, que têm um enorme impacto no jogo e consequentemente no seu resultado. No anexo 5 foi colocado uma gráfico onde podemos analisar a distribuição dos atributos desta experiência uns em função dos outros.

5.6 Métodos de aposta

Uma das grandezas a ser avaliada no modelo final é o retorno do investimento, devido à falta de informação respetiva a odds de casas de apostas e a necessidade atual de importar manualmente essa mesma informação para o sistema, este cálculo do ROI foi realizado tendo como base um dataset significativamente menor que o original. O dataset utilizado contêm os jogos do torneio ATP US Open de 2019. Serão utilizadas as odds oferecidas pela casa de apostas William Hill imediatamente antes do início dos respetivos jogos. Esta informação foi obtida através do site oddsportal [36]. Além disso, para que seja possível avaliar esta grandeza é também preciso definir uma estratégia de aposta a utilizar. Nesta secção serão descritos os três métodos que foram utilizados como estratégia de aposta para avaliar o retorno do modelo final.

5.6.1 Método 1

O primeiro método consiste em simplesmente apostar 1€ em cada uma das previsões disponibilizadas pelo modelo, o objetivo deste método não é obter o melhor resultado possível, mas sim identificar o resultado base a partir do qual melhorar.

5.6.2 Método 2

No segundo método o objetivo foi tentar de alguma forma relacionar o valor a apostar com o risco inerente à aposta, neste método o valor a apostar toma valores entre 0€ e 10€ de acordo com a probabilidade calculada pelo modelo, sendo que previsões com menor probabilidade terão valores de aposta menores, e previsões com mais probabilidade terão valores de aposta maiores.

5.6.3 Método 3

No terceiro método foi utilizada a fórmula de cálculo do valor a apostar desenvolvida no método 2, mas neste método apenas são considerados para apostas jogos em que a probabilidade calculada pelo modelo de previsão seja superior a 70%.

5.7 Serviço de Previsões

Para que seja possível pedir previsões ao modelo e mostrar essas previsões numa aplicação web, foi criado um web service. Este serviço foi desenvolvido usando a framework Django e a biblioteca django rest framework, que oferece politicas de autenticação e serialização automática de objetos obtidos a partir do ORM.

5.7.1 Base de dados

O serviço implementado tem uma base de dados independente da base de dados utilizada no treino e teste dos modelos de previsão. Esta escolha deve-se à tentativa de manter o serviço desacoplado do resto do sistema, de maneira a ser mais fácil de desenvolver, fazer deploy e escalar ambas as partes independentemente [37]. Além disso, o serviço precisa apenas de ter na base de dados os dados que são relevantes para o cliente que o vai consumir e os dados que necessita para pedir ao modelo escolhido as previsões.

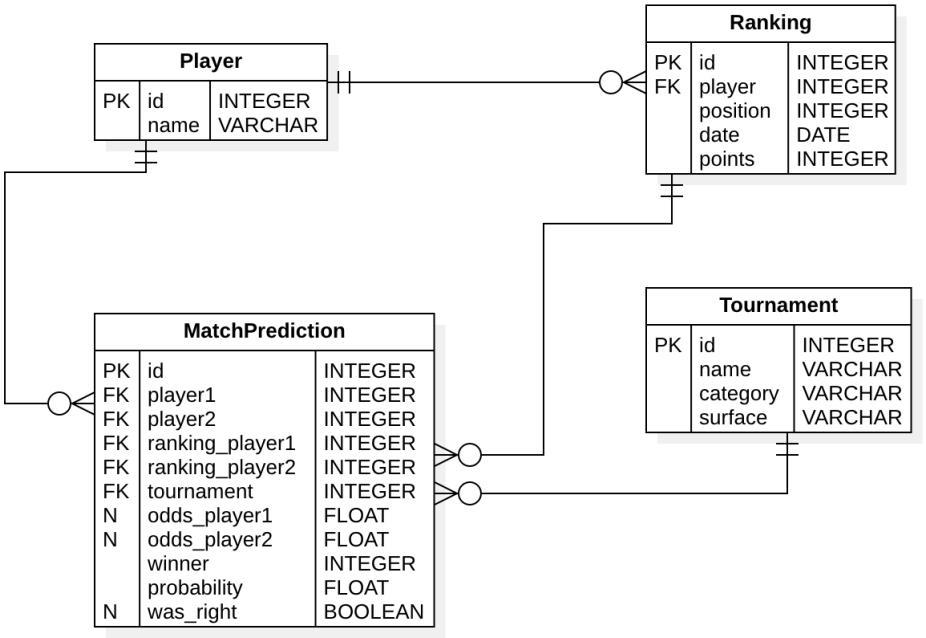


Figura 10 - Modelo de dados (serviço)

As tabelas de jogador, ranking e torneio são versões simplificadas das mesmas tabelas do sistema de treino e teste de modelos. A sincronização das mesmas é neste momento feita por um script mas o objetivo final é que esta comunicação entre o serviço e o sistema de treino e teste de modelos seja feita através de um message broker.

A tabela de previsões é exclusiva da base de dados deste serviço, e é responsável por guardar todas as previsões que foram pedidas ao modelo de previsão escolhido. Esta tabela tem duas chaves estrangeiras para a tabela de jogadores para identificar os jogadores que disputam o jogo, tem também duas chaves estrangeiras para a tabela de rankings para sabermos o ranking de cada um desses jogadores, e tem ainda uma chave estrangeira para a tabela de torneios para identificar o torneio ao qual o jogo pertence. A tabela de previsões tem também dois campos de odds, que correspondem às odds oferecidas pelas casas de apostas para cada um dos jogadores vencer o encontro. E por fim, tem três campos que caracterizam a previsão, que são o vencedor previsto, a probabilidade dessa previsão, e ainda um campo que é preenchido após o jogo para informar se a previsão se verificou ser correta ou não.

5.7.2 Pedidos HTTP

O serviço de previsões suporta os métodos http get, head, post, put, patch, delete e options. Com estes métodos, o serviço permite realizar operações CRUD sobre os modelos apresentados anteriormente. Nesta secção serão abordados apenas os métodos get e post para o modelo de previsões, que são os mais relevantes para o sistema neste momento.

5.7.3 Pedido GET

Os pedidos get de previsões podem ser feitos a dois endpoints, o endpoint "/match-predictions" devolve uma lista de previsões, enquanto que o endpoint "/match-predictions/:id" devolve uma determinada previsão definida pelo id passado no url. Os pedidos a estes endpoints são essencialmente úteis para a aplicação web, a partir destes a aplicação consegue pedir previsões para os próximos jogos, ou então, pedir um histórico de previsões de jogos que já terminaram, de maneira a mostrar se essas previsões estavam corretas ou erradas.


```
{
   "id":7,
   "player1":{
      "url":"http://127.0.0.1:8000/players/2/",
      "name":"John Isner"
   },
   "player2":{
      "url":"http://127.0.0.1:8000/players/3/",
      "name":"Jordan Thompson"
   },
   "ranking_player1":{
      "url":"http://127.0.0.1:8000/rankings/4/",
      "position":15,
      "date":"2019-08-05",
      "points":2085,
      "player":"http://127.0.0.1:8000/players/2/"
   },
   "ranking_player2":{
      "url":"http://127.0.0.1:8000/rankings/5/",
      "position":46,
      "date":"2019-08-05",
      "points":1034,
      "player":"http://127.0.0.1:8000/players/3/"
   },
   "tournament":{
      "url":"http://127.0.0.1:8000/tournaments/3/",
      "name":"Coupe Rogers",
      "category":"M",
      "surface":"Hard"
   },
   "winner":0,
   "probability":75.4892289638519,
   "odds_player1":1.44,
   "odds_player2":2.62,
   "was_right":true
}
```


Figura 11 - Exemplo de resposta a pedido GET de uma previsão

Como podemos verificar no exemplo da Figura 11 as respostas de previsões devolvem um JSON com os campos da previsão e também com os campos dos objetos relacionados (jogadores, rankings e torneio). Desta maneira, evitamos que os consumidores deste serviço tenham que fazer pedidos após receberem a resposta da previsão para completar a informação dos objetos relacionados.

5.7.4 Pedido POST

O pedido post de previsões é feito ao endpoint "/match-predictions" e permite criar uma nova previsão. Este pedido é normalmente feito por um script que a partir do calendário de próximos

jogos de ténis, vai buscar os ids dos objetos relacionados, e faz um pedido de criação de uma nova previsão.

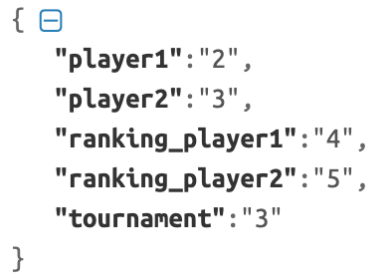


Figura 12 - Exemplo do body de um pedido POST de uma previsão

Como podemos verificar no exemplo da Figura 12, no método post é enviado um body apenas com os ids dos objetos relacionados com a previsão. Quando este pedido chega ao serviço, este vai buscar os objetos respetivos a partir dos ids. Após isso, é feita uma chamada ao modelo de previsão com a informação que este necessita, e é devolvida a previsão como um valor numérico entre 0 e 1. Este valor vai dar origem ao campo do vencedor previsto e da probabilidade a partir do seguinte método:

```
winner = 2 if prediction > 0.5 else 1
if winner == 1:
    prediction = 0.5 - prediction
if winner == 2:
    prediction -= 0.5

probability = prediction * 100 / 0.5
```

Figura 13 - Transformação do output do modelo

Como podemos analisar na Figura 13, o campo do vencedor previsto será igual a 2 (jogador 2) se a previsão do modelo for superior a 0.5, caso contrário será igual a 1 (jogador 1). No caso de a previsão do modelo ser de o jogador 1 vencer, a 0.5 retiramos o valor da previsão, isto vai garantir que quanto mais próximo de 0 for o valor devolvido pelo modelo mais próximo de 0.5 será o valor resultante. No caso de a previsão do modelo ser de o jogador 2 vencer, retiramos 0.5 ao valor devolvido pelo modelo, garantindo assim que vai também obedecer à escala de 0 a 0.5. Com isto, garantimos que a previsão neste momento será sempre um valor entre 0 (menos provável) e 0.5 (mais provável), o campo da probabilidade é então calculado com uma regra de três simples, onde multiplicamos o valor da previsão por 100 e dividimos o resultado por 0.5. Por fim, tendo os campos da previsão todos preenchidos, esta é inserida na base de dados.

5.8 Aplicação Web

A aplicação web tem como objetivo mostrar as previsões para os próximos jogos de ténis e mostrar um histórico de previsões efetuadas anteriormente. Esta aplicação é uma single page application (SPA), ou seja, quando o utilizador acede à aplicação, é lida apenas uma página html que carrega o código javascript da aplicação, a partir desse momento, qualquer navegação que ocorra é tratada no próprio browser substituindo o conteúdo da página, em vez de ser pedida uma nova página ao servidor como nas abordagens de server side rendering.

5.8.1 Browser Suporte

A lista de browsers a suportar foi definida de acordo com quota de mercado do browser, a razão principal para o uso desta estratégia deve-se essencialmente à facilidade de manutenção no futuro, sendo que à medida que o tempo passa a lista de browsers irá ajustar-se automaticamente de acordo com o mercado. A quota mínima de mercado que foi definida foi de 0.2%, ou seja, todos os browser com uma quota de mercado superior a esse valor devem ser suportados.

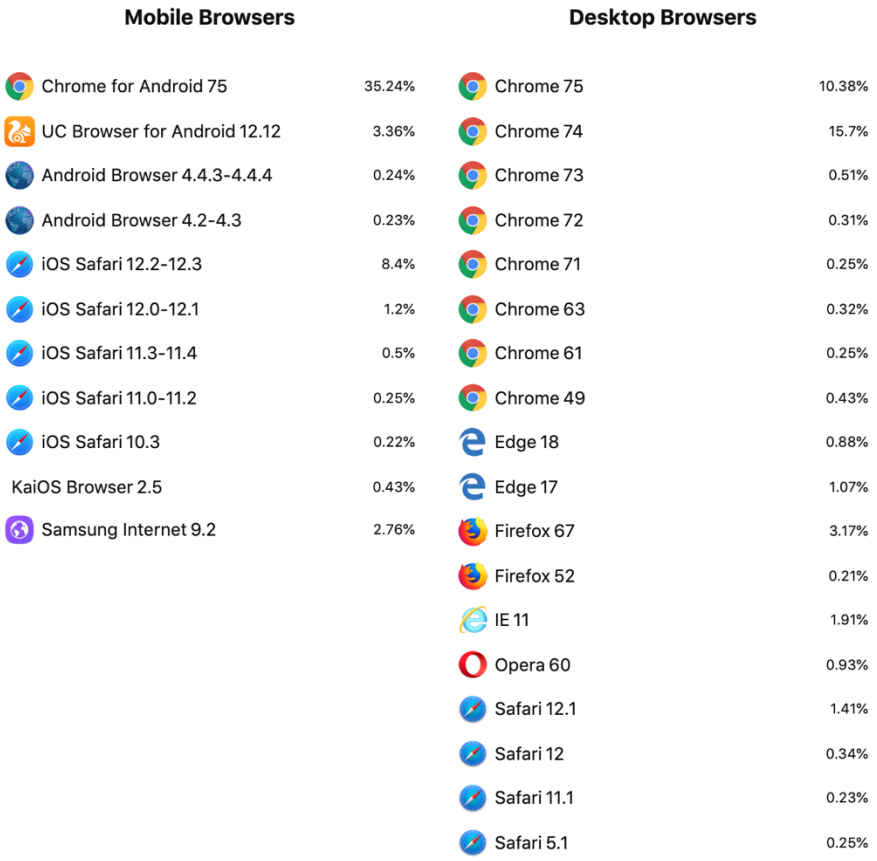


Figura 14 - Lista de browser suporte

Na Figura 14 podemos observar a lista completa dos browser suportados neste momento por esta aplicação, o conjunto destes browser corresponde a 91.37% do total do mercado neste momento [38]. À medida que novos browser ganhem uma quota de mercado superior a 0.2% e browsers mais antigos caiam abaixo dessa mesma quota, esta lista irá ajustar-se automaticamente e o código da aplicação será compilado tendo em conta a lista atualizada.

6 Avaliação da Solução

Neste capítulo será abordada a avaliação da solução implementada, primeiro será descrito o método que foi utilizado para avaliar as diferentes abordagens, e por fim serão apresentadas essas mesmas abordagens e os seus resultados.

6.1 Método de Avaliação

Nesta secção será abordado o método utilizado para avaliar as várias experiências e abordagens que serão realizadas no decorrer deste projeto. A descrição do método de avaliação passará pela enumeração das grandezas que serão utilizadas na avaliação, a enumeração das hipóteses que se pretende testar, e por fim a identificação da metodologia de avaliação usada.

6.1.1 Grandezas

6.1.1.1 Taxa de acerto

A taxa de acerto é um valor percentual que indica a percentagem do número de jogos usados para avaliar o modelo em que a previsão correspondeu ao resultado final do mesmo. O objetivo mínimo que foi definido para esta grandeza é uma taxa de acerto de 65%.

6.1.1.2 ROI

O return on investment é um valor percentual que indica qual a percentagem de retorno caso fossem colocadas apostas em todas as previsões realizadas para os jogos usados para avaliar o modelo. O objetivo mínimo que foi definido para esta grandeza é um ROI de 3%.

6.1.2 Teste de Hipóteses

Neste subcapítulo serão usadas as grandezas apresentadas no subcapítulo anterior para avaliar os modelos de acordo com um conjunto de hipóteses.

Tabela 10 - Teste de hipóteses para objetivos

Modelo	Taxa de Acerto > 65%
Regressão Logística	Verdade / Falso
Redes Neuronais	Verdade / Falso
SVM	Verdade / Falso

A Tabela 10 servirá para apresentar a comparação feita entre o resultado obtido para a taxa de acerto de cada um dos modelos de previsão, e os objetivos que foram definidos no inicio do projeto. O modelo final escolhido será também avaliado de acordo com a grandeza de retorno do investimento.

Tabela 11 - Teste de hipótese das grandezas

	Regressão Logística	Redes Neuronais	SVM
Regressão Logística	X	Maior / Menor	Maior / Menor
Redes Neuronais	Maior / Menor	X	Maior / Menor
SVM	Maior / Menor	Maior / Menor	X

A Tabela 11 servirá para apresentar a comparação feita das taxas de acerto das diferentes abordagens testadas. Desta maneira, conseguimos identificar os modelos com melhor taxa de acerto.

6.1.3 Metodologia de Avaliação

Para efeitos de avaliação não serão considerados jogos em que exista falta de informação em algum dos parâmetros utilizados no modelo. Desta maneira evita-se previsões erradas por falta de informação, e simula-se também o processo do apostador que escolhe apenas jogos em que tem toda a informação que necessita disponível. Depois de ser feita essa filtragem de jogos usados para avaliar os modelos, estes serão avaliados de acordo com o a sua taxa de acerto. Para cada modelo, a taxa de acerto será comparada com os objetivos definidos e também com os valores dos restantes modelos. Para que os valores das grandezas sejam mais precisos serão utilizadas técnicas de validação cruzada, tais como o k-fold cross validation. No final de todo o processo, será escolhido um modelo analisando as tabelas de testes de hipóteses com os valores calculados para cada um dos modelos.

6.1.4 Teste estatístico

O teste estatístico que será utilizado para avaliar cada modelo individualmente será o k-fold cross-validation com dez partições. Este teste irá dividir o dataset em dez partições de tamanhos iguais, cada uma das partições será utilizada como informação de teste para um modelo gerado, a partir do treino das partições restantes. Este processo repete-se por dez iterações de maneira a todas as partições serem usadas uma vez como informação de teste.

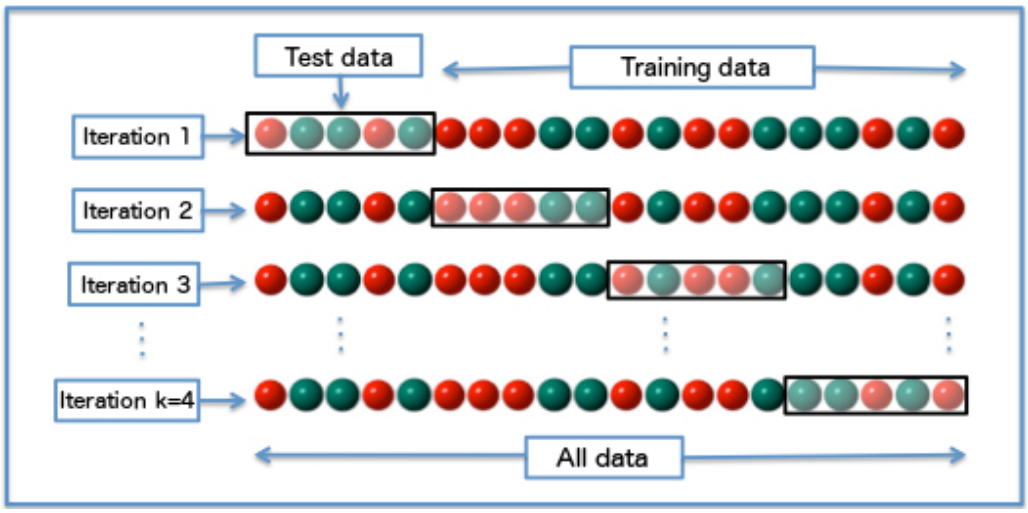


Figura 15 - k-fold cross-validation

6.2 Resultados

Nesta secção serão apresentados os resultados obtidos nas cinco experiências realizadas, em cada uma das experiências teremos três resultados diferentes que correspondem aos três tipos de modelos testados. Além disso, será feita uma análise do modelo final escolhido, onde é feita uma análise à precisão, distribuição do erro e ainda o retorno do investimento. É também relevante referir que os resultados de todos estes modelos foram alcançados utilizando o mesmo dataset de jogos, que contêm os jogos do circuito ATP entre 2010 e 2018.

6.2.1 Experiência 1

Tabela 12 - Resultados da experiência 1

Modelo	Taxa de Acerto > 65%
Regressão Logística	66.9% ✓
Redes Neuronais	66.6% ✓
SVM	55.8% ✗

A experiência 1 foi interessante para perceber a importância do ranking ATP, apenas com a pontuação dos jogadores nesse ranking consegue-se resultados bastante interessantes. As taxas de acerto do modelo de regressão logística e da rede neuronal foram superiores ao objetivo definido.

6.2.2 Experiência 2

Tabela 13 - Resultados da experiência 2

Modelo	Taxa de Acerto > 65%
Regressão Logística	67.5% ✓
Redes Neuronais	67.2% ✓
SVM	58.0% ✗

Na experiência 2 foi possível melhorar os resultados da experiência 1, adicionando a informação sobre a categoria do torneio. Nesta experiência conseguimos uma melhoria de 0.6% no modelo de regressão logística e na rede neuronal, e ainda 2.2% no modelo SVM. As taxas de acerto do modelo de regressão logística e da rede neuronal foram superiores ao objetivo definido.

6.2.3 Experiência 3

Tabela 14 - Resultados da experiência 3

Modelo	Taxa de Acerto > 65%
Regressão Logística	66.1% ✓
Redes Neuronais	66.5% ✓
SVM	58.3% ✗

Na experiência 3 os resultados foram desapontantes, a tentativa de adicionar atributos que caracterizam o jogador, tais como, a idade e a mão de jogo, pioraram as taxas de acerto do modelo de regressão logística e da rede neuronal em relação à experiência anterior. O modelo de regressão logística e a rede neuronal tiveram um decréscimo na taxa de acerto de 1.4% e 0.7% respetivamente. O modelo SVM teve uma ligeira melhoria na taxa de acerto de 0.3%. Apesar dos resultados desapontantes, o modelo de regressão logística e a rede neuronal verificaram taxas de acerto superiores ao objetivo definido.

6.2.4 Experiência 4

Tabela 15 - Resultados da experiência 4

Modelo	Taxa de Acerto > 65%
Regressão Logística	67.1% ✓
Redes Neuronais	67.0% ✓
SVM	59.5% ✗

Na experiência 4 também não foi possível melhorar os resultados da experiência 2, a tentativa de adicionar a performance dos jogadores nos últimos 10 jogos também piorou as taxas de acerto do modelo de regressão logística e da rede neuronal. As taxas de acerto do modelo de regressão logística e da rede neuronal tiveram um decréscimo em relação à experiência 2 de 0.4% e 0.2% respetivamente. O modelo SVM verificou um aumento da taxa de acerto de 1.5% em relação à experiência 2. Apesar dos resultados desapontantes, o modelo de regressão logística e a rede neuronal verificaram taxas de acerto superiores ao objetivo definido.

6.2.5 Experiência 5

Tabela 16 - Resultados da experiência 5

Modelo	Taxa de Acerto > 65%
Regressão Logística	68.0% ✓
Redes Neuronais	67.8% ✓
SVM	60.3% ✗

Na experiência 5 foi finalmente possível melhorar os resultados da experiência 2, adicionando ao modelo informação sobre a superfície do court onde o jogo é disputado. As taxas de acerto do modelo de regressão logística, da rede neuronal e do modelo SVM verificaram uma melhoria em relação à experiência 2 de 0.5%, 0.6% e 2.3% respetivamente. O modelo de regressão logística e a rede neuronal verificaram taxas de acerto superiores ao objetivo definido.

6.2.6 Comparação das abordagens

Nesta secção serão comparados os resultados obtidos pelas diferentes abordagens testadas, neste caso, o modelo de regressão logística, a rede neuronal e o modelo SVM. Os resultados da taxa de acerto apresentados correspondem à média desta grandeza para as experiências realizadas.

Tabela 17 – Comparação dos resultados das abordagens

	Regressão Logística	**Redes Neuronais**	**SVM**
Regressão Logística	X	67.12% vs 67.02% ✓	67.12% vs 58.38% ✓
Redes Neuronais	67.02% vs 67.12% ✗	X	67.02% vs 58.38% ✓
SVM	58.38% vs 67.12% ✗	58.38% vs 67.02% ✗	X

Na tabela anterior é feita uma comparação das médias da taxa de acerto nas experiências, das diferentes abordagens testadas, esta média é calculada somando as taxas de acerto dos modelos de uma determinada abordagem nas cinco experiência, e dividindo esse valor por cinco. Na comparação feita na tabela 17, em cada célula da tabela o valor da esquerda corresponde à média da abordagem analisada nessa linha, e o valor da direita corresponde à média da abordagem analisada nessa coluna. Como podemos verificar o modelo de regressão logística foi o que obteve o melhor resultado, com 67.12% de média de taxa de acerto das várias experiências, seguido da rede neuronal com uma média de 67.02%, e por fim o modelo SVM com uma média de 58.38%.

6.3 Análise do modelo final

O modelo final escolhido foi o modelo de regressão logística da experiência 5, visto ser o modelo que obteve uma taxa de acerto mais alta. Nesta secção será feita uma análise mais detalhada a esse modelo.

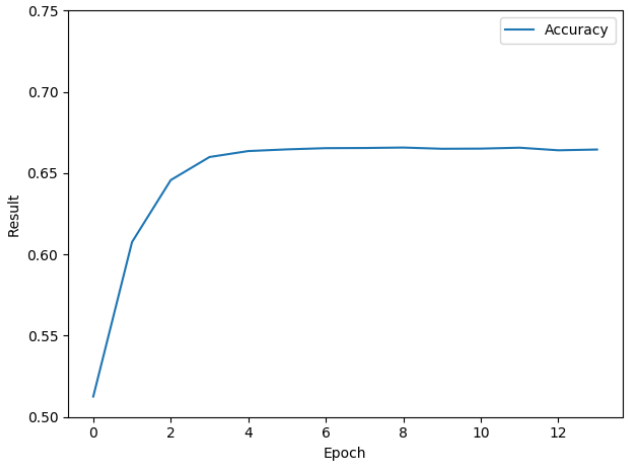


Figura 16 - Gráfico de precisão do modelo final

O gráfico da Figura 16 representa as taxas de acerto obtidas em fase de treino para o modelo final, como podemos verificar a taxa de acerto aumenta bastante nas primeiras épocas de treino, e a partir da terceira época começa a estabilizar, o modelo continua a melhorar ligeiramente a precisão até à época onze onde obteve o seu melhor resultado.

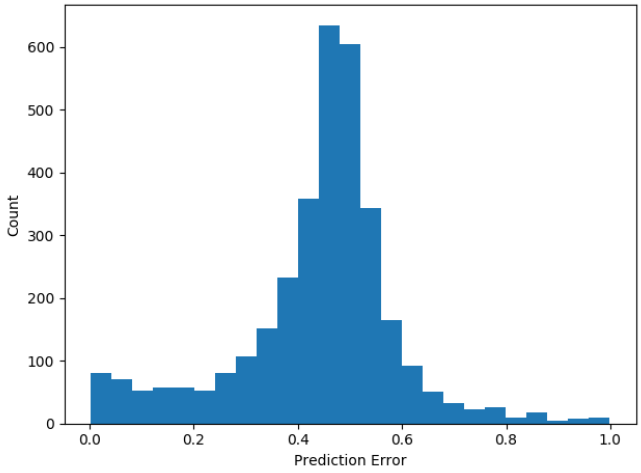


Figura 17 - Gráfico de distribuição do erro do modelo final

O gráfico da Figura 17 representa o número de ocorrências de um determinado erro numa previsão, este erro é o valor absoluto da diferença entre a previsão e o resultado esperado. Como podemos facilmente observar, a grande maioria dos erros compreende valores entre os 0.4 e 0.6. Isto deve-se aos casos em que a previsão do modelo compreende também valores entre 0.4 (jogador 1 vencer com 20% de certeza) e 0.6 (jogador 2 vencer com 20% de certeza). Os valores mais próximos das extremidades (0 ou 1) ocorrem quando o modelo tem mais certezas do vencedor, e ai podemos verificar que a contagem dos erros é significativamente menor.

6.3.1 Retorno do Investimento

Nesta secção serão analisados os resultados obtidos para o retorno do investimento no modelo de previsão final. Estes resultados foram obtidos utilizando as previsões para os jogos do torneio ATP US Open 2019 e utilizando os métodos de aposta descritos na secção 5.6.

Tabela 18 - Resultados do retorno do investimento

	ROI
Método 1	-39.62%
Método 2	-11.91%
Método 3	+4.32%

Na Tabela 18 podemos observar os resultados obtidos em cada um dos métodos, como podemos verificar os resultados são maioritariamente negativos, isto deve-se essencialmente ao facto das odds oferecidas pela casa de apostas serem bastante mais baixas do que o risco associado à aposta, de maneira a proteger os interesses da própria casa de apostas. Apesar disso, foi possível utilizando o terceiro método obter um retorno positivo superior a 4% conseguindo assim superar o objetivo definido de 3%. Contudo, deve-se realçar que este

resultado foi obtido com um dataset relativamente pequeno (US Open 2019), assim sendo não dá garantias da mesma rentabilidade durante um período de tempo mais alargado.

7 Conclusão

O mercado das apostas desportivas é um mercado em grande crescimento, em 2017 o valor deste mercado era de 45.8 mil milhões de dólares e estima-se que em 2024 este mercado tenha um valor de 94.4 mil milhões, ou seja, mais do dobro do que em 2017 [1].

O principal objetivo deste trabalho é tentar reduzir o tempo que os apostadores precisam de despender a analisar a performance dos jogadores antes de apostarem. Sendo previsões calculadas por um modelo de machine learning consegue-se disponibilizá-las para uma enorme quantidade de jogos que não seria possível a uma pessoa analisar.

O modelo de previsão final foi escolhido após várias experiências realizadas com datasets de informação e abordagens diferentes, tendo em conta a taxa de acerto do mesmo. O modelo final é um modelo de regressão logística com variáveis que caracterizam os jogadores e o torneio que estão a disputar, a taxa de acerto obtida para este modelo foi de 68%, bastante superior ao objetivo definido de 65%.

Tendo em conta que o objetivo final é a disponibilização destas previsões para apostadores, foi feito ao longo deste projeto algum trabalho com esse objetivo em mente. Foram desenvolvidos webservices responsáveis por comunicar com o modelo de previsão, armazenar as previsões e alimentar a aplicação web com essas previsões. Por fim, foi desenvolvida uma aplicação web básica, que neste momento, disponibiliza uma comparação entre as previsões do modelo e as odds oferecidas pela casa de apostas.

Os resultados com a aplicação do modelo final foram bastante positivos, e cumpriram os objetivos definidos, apesar disso foi também possível verificar que as odds oferecidas pelas casas de apostas são bastante defensivas portanto não é fácil ter ganhos consistentes apenas com a aplicação das previsões do modelo. Este modelo deve então servir apenas como mais uma ferramenta para os apostadores e não como única análise feita pelos mesmos.

7.1 Limitações e Trabalho Futuro

O trabalho que foi realizado deixa ainda algumas limitações e possíveis melhorias que podem ser exploradas no futuro, nesta secção iremos enumerar algumas que foram levantadas ao longo do desenvolvimento deste projeto.

7.1.1 Dataset e Modelo de Previsão

No que diz respeito ao dataset base um melhoria possível seria a inclusão de mais variáveis, tais como, o país onde se disputa um determinado torneio, condições meteorológicas e odds de casas de apostas. Quanto ao modelo de previsão as melhorias passariam por testar a inclusão de novas variáveis no modelo e verificar o seu impacto na taxa de acerto do mesmo, se possível, tornar todo este processo automático.

7.1.2 Serviços

Nos serviços uma melhoria possível seria a adoção de alguns padrões da arquitetura de micro serviços. Com a adoção do padrão messaging [39] de microserviços, seria possível melhorar a comunicação entre o serviço de previsões e a aplicação deuce brain que contêm o modelo de machine learning.

7.1.3 Aplicação Web

Na aplicação web existe um enorme espaço para o desenvolvimento de novas funcionalidades, tais como, alertas de novas previsões, feedback das previsões dado por apostadores, subscrições pagas, adicionar previsões à “watch list”.

7.1.4 Automatismos

No que diz respeito a automatismos é possível identificar algumas lacunas que podem ser melhoradas. Uma melhoria importante para o sistema seria a atualização do dataset, treino e teste do modelo com esse dataset automaticamente assim que um torneio terminasse.

Seria bastante importante também automatizar o processo de entrega de software, tendo uma pipeline que é despoletada pelo repositório e é responsável por correr os testes e fazer deploy do sistema.

Referências

[1] "Online gambling worldwide 2017 and 2024," [Online]. Available: https://www.statista.com/statistics/270728/market-volume-of-online-gaming-worldwide/.

[2] M. Townend, "Tennis gambling market second only to football for bookmakers after boom in online and in-play betting," 18 01 2016. [Online]. Available: https://www.dailymail.co.uk/sport/tennis/article-3405544/Tennis-gambling-market-second-football-bookmakers-boom-online-play-betting.html. [Acedido em 10 12 2018].

[3] C. Pickering, "How the Rise of Machine Learning Is Impacting Sport," [Online]. Available: https://simplifaster.com/articles/machine-learning-sports/.

[4] J. Sackmann, "JeffSackmann/tennis_atp," 13 03 2015. [Online]. Available: https://github.com/JeffSackmann/tennis_atp. [Acedido em 20 11 2018].

[5] R. Wirth e J. Hipp, "CRISP-DM: Towards a standard process model for data mining," 2000. [Online]. Available: http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.198.5133. [Acedido em 05 02 2019].

[6] "Tennis Rules," [Online]. Available: http://protennistips.net/tennis-rules/.

[7] "Odds," [Online]. Available: https://en.wikipedia.org/wiki/Odds.

[8] A. Gray, "The Size and Increase of the Global Sports Betting Market," 25 01 2019. [Online]. Available: https://www.sportsbettingdime.com/guides/finance/global-sports-betting-market/. [Acedido em 02 02 2019].

[9] T. M. Mitchell, Machine Learning, McGraw-Hill, 1997.

[10] M. Rouse, "What is supervised learning?," Setembro 2019. [Online]. Available: https://searchenterpriseai.techtarget.com/definition/supervised-learning.

[11] R. Shaikh, "Feature Selection Techniques in Machine Learning with Python," Outubro 2018. [Online]. Available: https://towardsdatascience.com/feature-selection-techniques-in-machine-learning-with-python-f24e7da3f36e.

[12] T. Bock, “What is a Correlation Matrix?,” [Online]. Available: https://www.displayr.com/what-is-a-correlation-matrix/.

[13] “Logistic function,” [Online]. Available: https://en.wikipedia.org/wiki/Logistic_function.

[14] S. Sharma, “Activation Functions in Neural Networks,” Setembro 2017. [Online]. Available: https://towardsdatascience.com/activation-functions-neural-networks-1cbd9f8d91d6.

[15] “Bounded function,” [Online]. Available: https://en.wikipedia.org/wiki/Bounded_function.

[16] “Differentiable function,” [Online]. Available: https://en.wikipedia.org/wiki/Differentiable_function.

[17] Josh, “Everything You Need to Know About Artificial Neural Networks,” Dezembro 2015. [Online]. Available: https://medium.com/technology-invention-and-more/everything-you-need-to-know-about-artificial-neural-networks-57fac18245a1.

[18] “Support-vector machine,” [Online]. Available: https://en.wikipedia.org/wiki/Support-vector_machine.

[19] “TensorFlow,” [Online]. Available: https://www.tensorflow.org/.

[20] “PyTorch,” [Online]. Available: https://pytorch.org/.

[21] “Keras,” [Online]. Available: https://keras.io/.

[22] “Pandas Data Analysis Library,” [Online]. Available: https://pandas.pydata.org/.

[23] “Django,” [Online]. Available: https://www.djangoproject.com/.

[24] “Django REST Framework,” [Online]. Available: https://www.django-rest-framework.org/.

[25] “Flask,” [Online]. Available: https://palletsprojects.com/p/flask/.

[26] “ReactJS,” [Online]. Available: https://reactjs.org/.

[27] “Sass,” [Online]. Available: https://sass-lang.com/.

[28] “Webpack,” [Online]. Available: https://webpack.js.org/.

[29] “Babel,” [Online]. Available: https://babeljs.io/.

[30] M. Nilsson, “Can a Machine Beat the Best Tennis Player in the World?,” 17 Junho 2019. [Online]. Available: https://towardsdatascience.com/can-a-machine-beat-the-best-tennis-player-in-the-world-79112b47f547.

[31] S. Kovalchik, “Searching for the GOAT of tennis win prediction,” em *Journal of Quantitative Analysis in Sports*, 2016, pp. 127-138.

[32] A. Cornman, G. Spellman e D. Wright, “Machine Learning for Professional Tennis Match Prediction and Betting,” em *Experimental Results*, Stanford University, 2018, pp. 3-4.

[33] “Tensorflow or PyTorch : The force is strong with which one?,” 24 Abril 2018. [Online]. Available: https://medium.com/@UdacityINDIA/tensorflow-or-pytorch-the-force-is-strong-with-which-one-68226bb7dab4.

[34] “Gradient Descent,” [Online]. Available: https://en.wikipedia.org/wiki/Gradient_descent.

[35] M. Nilsson, “Can a Machine Beat the Best Tennis Player in the World?,” 17 Junho 2019. [Online]. Available: https://towardsdatascience.com/can-a-machine-beat-the-best-tennis-player-in-the-world-79112b47f547.

[36] “ATP US Open Odds,” [Online]. Available: https://www.oddsportal.com/tennis/usa/atp-us-open/results/.

[37] C. Richardson, “Pattern: Database per service,” [Online]. Available: https://microservices.io/patterns/data/database-per-service.html.

[38] “Browser List,” [Online]. Available: https://browserl.ist/?q=>+0.2%25%2C+not+dead%2C+not+op_mini+all.

[39] C. Richardson, “Messaging,” [Online]. Available: https://microservices.io/patterns/communication-style/messaging.html.

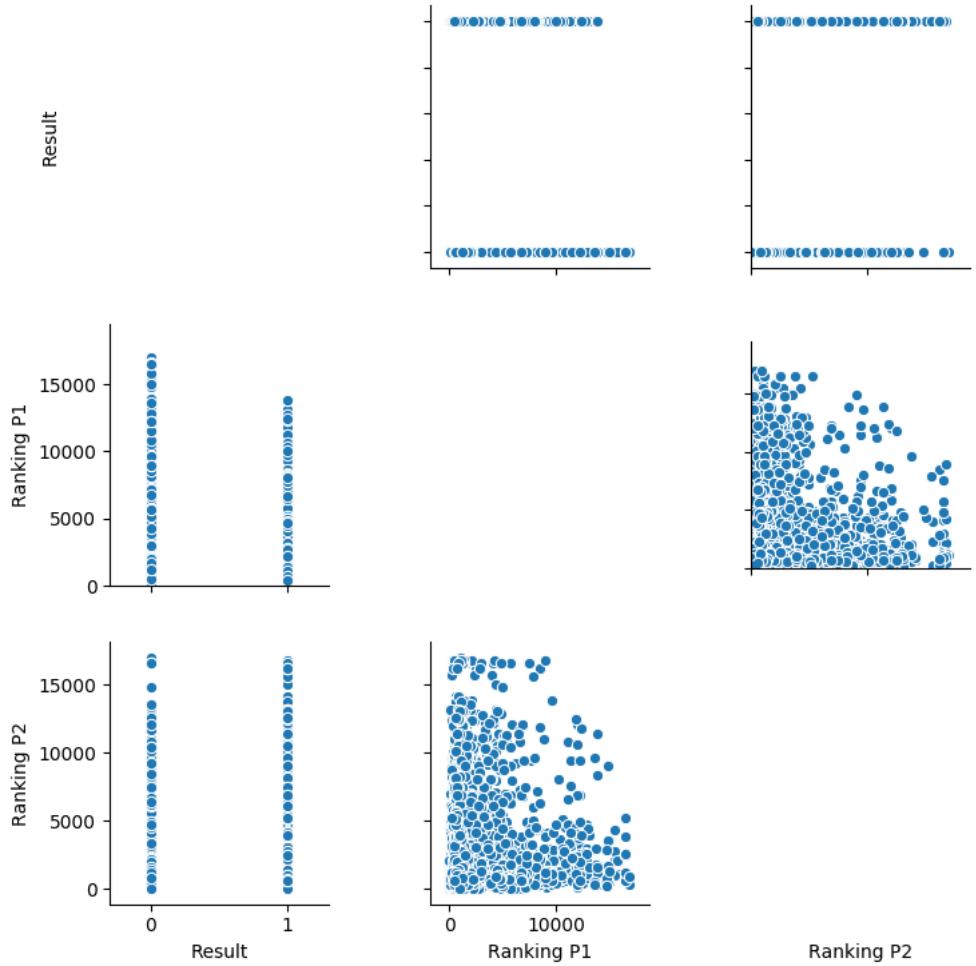
[40] M. Ingram, “An introduction to tennis modelling,” betfair.com.au, 22 11 2018. [Online]. Available: https://www.betfair.com.au/hub/an-introduction-to-tennis-modelling/. [Acedido em 04 01 2018].

[41] M. Sipko, “5.3 Artificial Neural Network,” em *Machine Learning for the Prediction of Professional Tennis Matches*, Imperial College London, 2015, pp. 37-43.

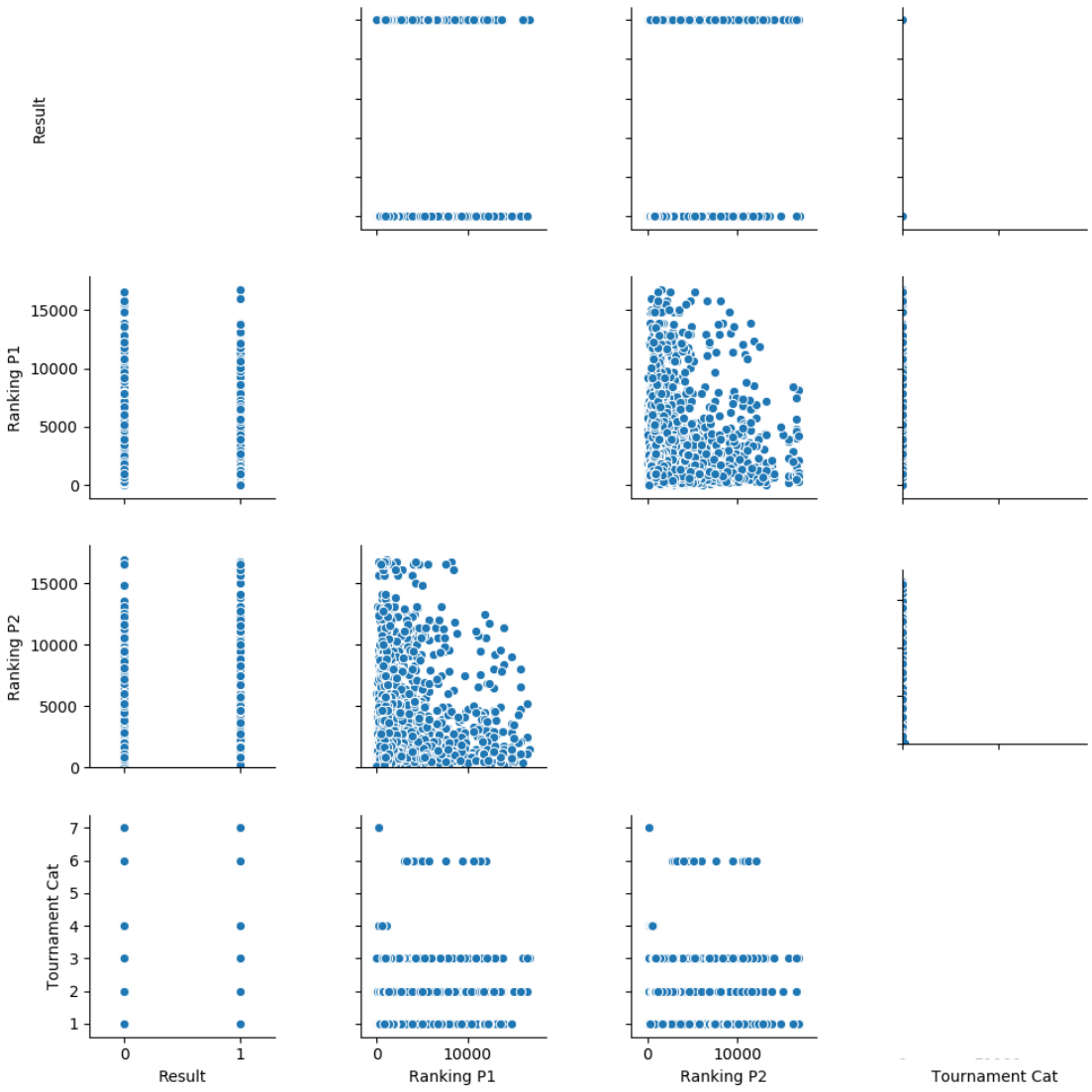
[42] S. Swaminathan, “Logistic Regression — Detailed Overview,” Março 2018. [Online]. Available: https://towardsdatascience.com/logistic-regression-detailed-overview-46c4da4303bc.

Anexos

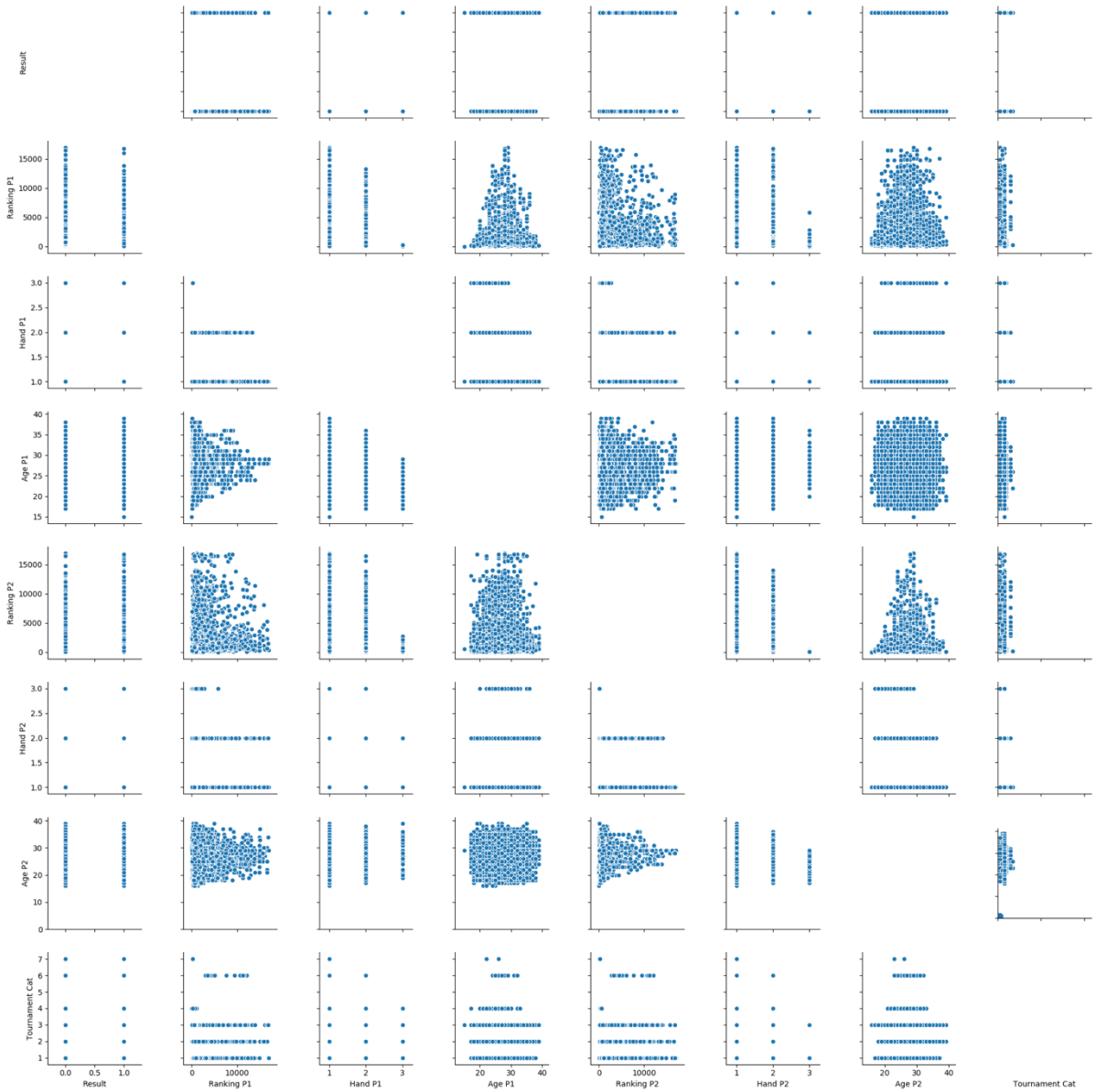
Anexo 1 – Distribuição de atributos da experiência 1



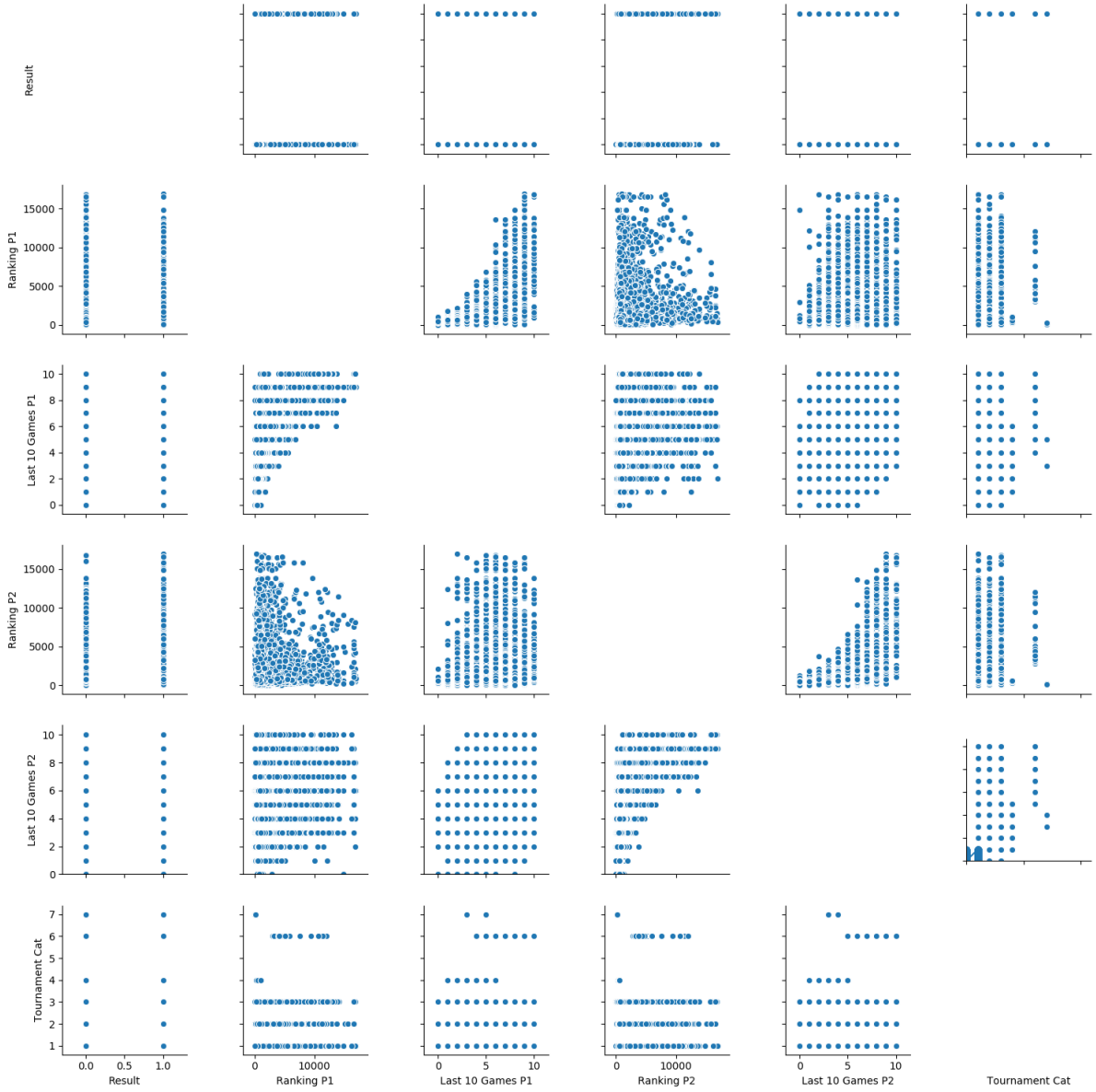
Anexo 2 – Distribuição de atributos da experiência 2



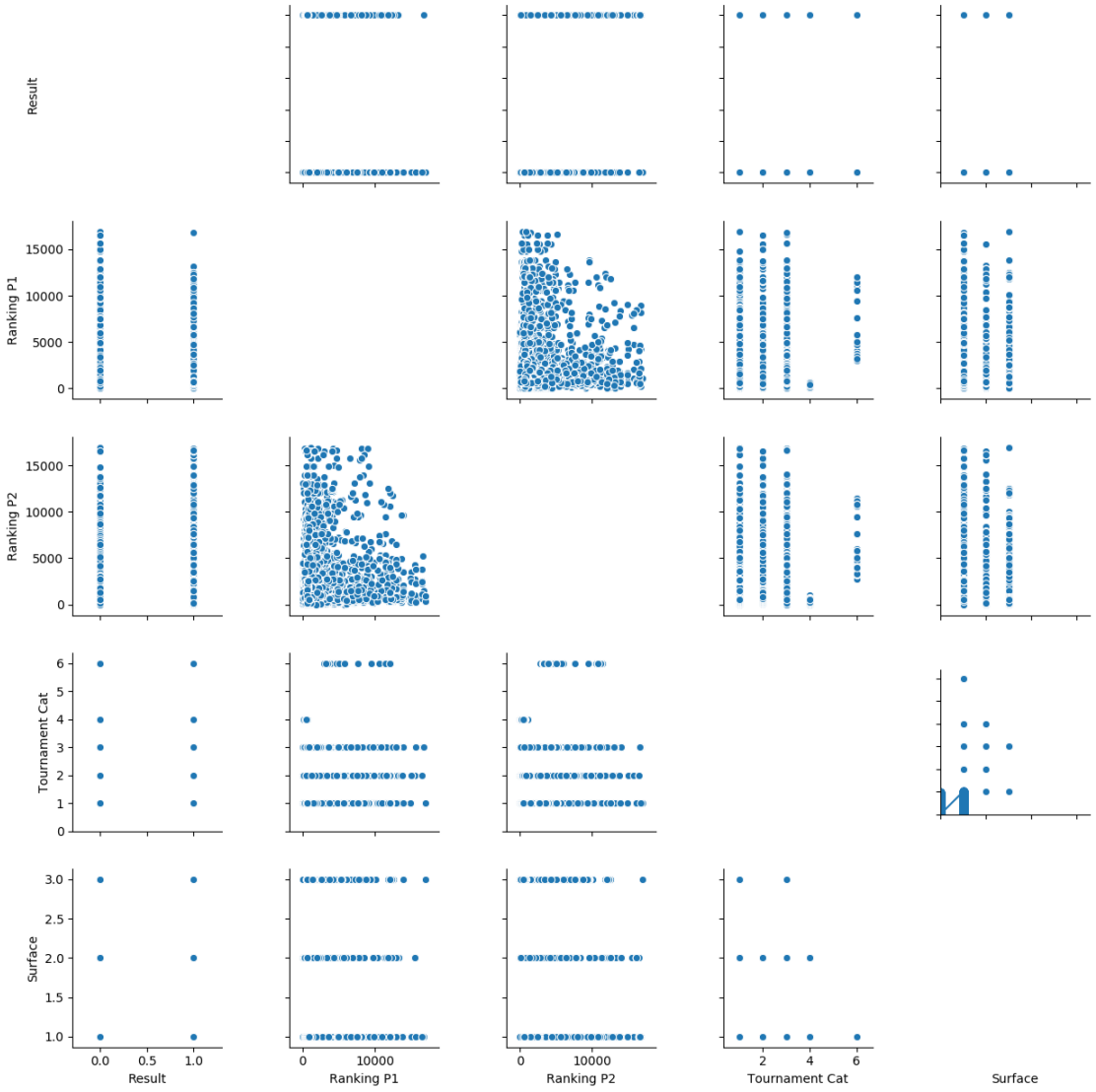
Anexo 3 – Distribuição de atributos da experiência 3



Anexo 4 – Distribuição de atributos da experiência 4



Anexo 5 – Distribuição de atributos da experiência 5



Anexo 6 – Distribuição de atributos do torneio

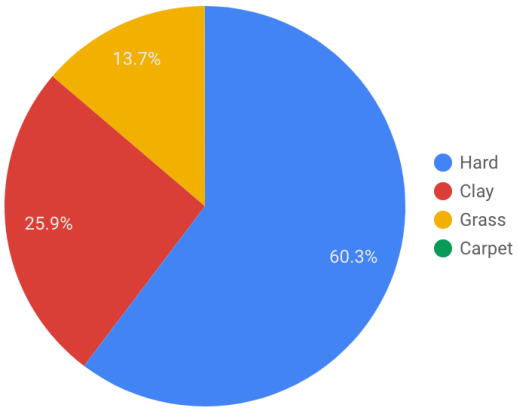


Figura 18 - Distribuição do atributo: Superfície do Court

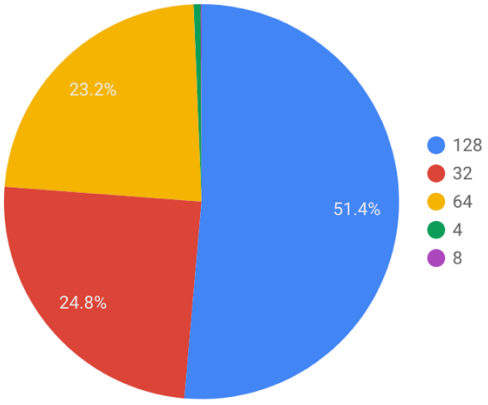


Figura 19 - Distribuição do atributo: Número de Jogadores

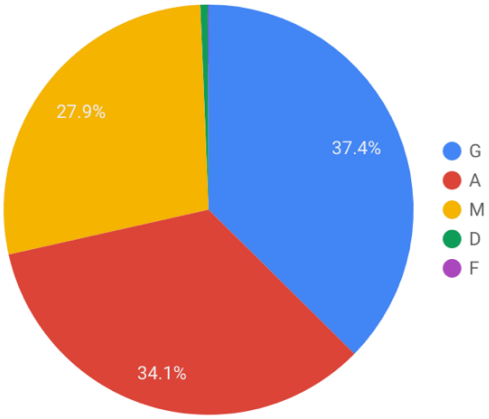


Figura 20 - Distribuição do atributo: Importância

Anexo 7 – Distribuição de atributos de um jogo

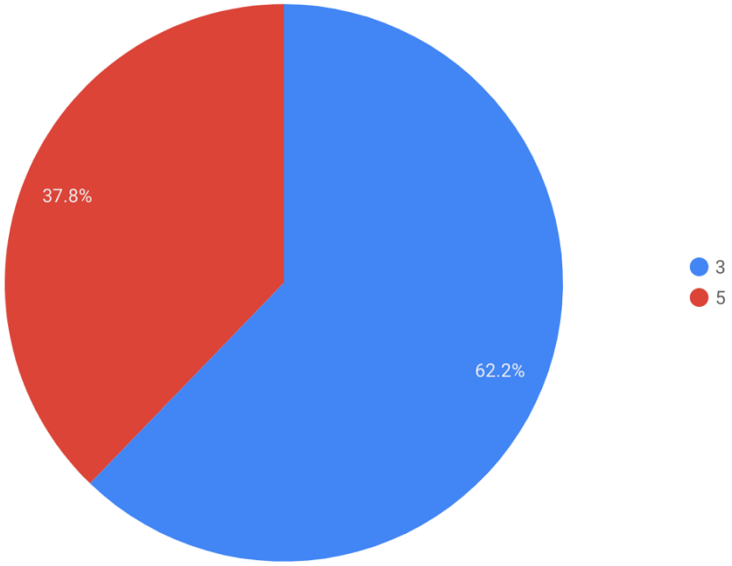


Figura 21 - Distribuição do atributo: Melhor de X Sets

Anexo 8 – Distribuição de atributos de um jogador

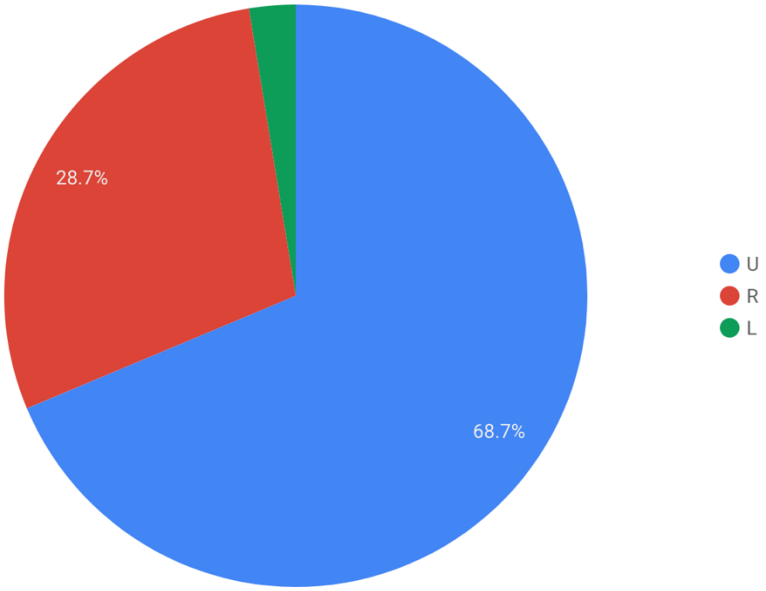


Figura 22 - Distribuição do atributo: Mão de Jogo