



Engenharia Informática

Departamento de ciencias e tecnologia

Laboratório de Projeto

Coordenador:

Professor Doutor Daniel Silvestre

Modelo de previsão e apostas desportivas

Discentes:

Cláudio Mirão – 20150766

Gonçalo Serôdio – 20150659

Francisco Trindade – 20150991

Lisboa, Setembro 2018

Resumo

Atualmente os algoritmos de *machine learning* estão bastante inseridos no dia-a-dia quer seja em filtros de spam, anúncios ou ate recomendações de filmes ou séries na aplicação Netflix.[1]

Estes algoritmos quando programados corretamente podem também ser usados para tentar prever diversos acontecimentos como por exemplo resultados desportivos e movimentações no mercado da bolsa de valores.

O intuito deste projeto passa por, com recurso a diferentes algoritmos, criar um modelo de previsão de resultados desportivos, um modelo de previsão de *odds* para diversos jogos e também um modelo inovador e pouco utilizado no mercado mas com grande vantagem para o utilizador ou apostador.

Neste projeto foram realizados três modelos de previsão para apostas desportivas.

Um modelo que com base nas estatísticas de cada equipa vai calcular a probabilidade de cada equipa vencer, ou haver um empate, um segundo modelo que vai usar as *odds* prévias de cada equipa e que vai calcular e originar *odds* novas para a vitória de cada equipa ou empate.

Por fim, um modelo inovador, não usado frequentemente online, que vai ajudar muito mais o apostador, pois permite que este tenha total controlo da equipa que ele pensa que será titular. Desta forma, o modelo vai fazer uma previsão, não com base na equipa, mas com base na performance individual dos jogadores e o impacto que cada um tem no jogo.

Qualquer um destes modelos quando aplicado individualmente já trará uma vantagem ao utilizador, contudo quando aplicados em conjunto vai permitir ter três diferentes abordagens ao mesmo jogo, lembrando ainda que uma destas abordagens tem a possibilidade de ser controlada pelo utilizador, dando a este uma ferramenta que prevê o impacto que cada jogador tem na sua equipa, resultando então numa vantagem enorme sobre outro utilizador que não tenha acesso a este modelo.

Porém, isto são modelos de previsão, que se baseiam no acontecimento mais comum, ou seja na regra, para fazer uma previsão, e por mais alta que seja a percentagem de um acontecimento, existe sempre uma exceção para a regra.

Em suma, estes três modelos, quando usados corretamente, são capazes de fornecer imensa informação a um utilizador sobre um determinado evento desportivo, esta informação rapidamente se transforma numa vantagem

importantíssima quando se trata de apostar num jogo, especialmente pela razão de que estes modelos de *machine learning* não são fáceis de fazer e também não estão disponíveis a qualquer apostador.

Abstract

Currently, machine learning algorithms are heavily embedded in day-to-day spam filters, ads, or even movie or series recommendations in the Netflix application.

These algorithms when properly programmed can also be used to try to predict several events such as sports results and movements in the stock market.

The goal of this project is to use different algorithms, to create a prediction model for sports results, a prediction model of odds for various games and also an innovative and less used model in the market but with great advantage for the user or bettor.

For this project three prediction models for sports betting were carried out.

One model is based on the statistics of each team. It will make a prediction on whether the home team wins, the away team wins or if there is draw.

The second model that will use the previous odds of each team and will calculate a new odd for the victory of the home team, the victory of the away team or a tie between both teams.

At last, a third and innovative model, not often used online that will help the bettor much more because it lets the bettor make a prediction on a matched between two teams based on the players that he thinks that will be on the starting eleven.

This model will make a prediction, not based on the team, but based on the individual performance of each player and the impact the player has had on the previous games he played.

Any of these models when applied individually will already bring an advantage to the user, however when applied together it will allow to have three different approaches to the same game, also remembering that one of these approaches has the possibility of being controlled by the user, giving the bettor a tool that predicts the impact each player has on his team, resulting in a huge advantage over another user who does not have access to this model.

These prediction models are based on the occurrence of the most common event, however the least common event is still eventually going to happen, which means that sometimes the models might fail their predictions but most of times the prediction models will be correct.

In short, these three models, when used correctly, are able to provide immense information to a user about a particular sporting event, this information quickly becomes a major advantage when it comes to betting on a game, especially for the reason that these models machine learning are not easy to make, and they are not available to any bettor.

Índice

Resumo.....	2
Abstract	4
Introdução.....	7
Objectivos.....	11
Dificuldades encontradas.....	12
Implementação do Projecto.....	13
Manual do Utilizador.....	15
Pré-requisitos de software	15
Instalações e configurações antes de correr o programa	17
Web Application.....	33
Manual do programador	42
Conclusão	64
Referências.....	65

Introdução

Este projeto foi desenvolvido no âmbito da unidade curricular “Laboratório de Projeto” do ano letivo 2017/2018 para a conclusão da licenciatura, com o apoio do Docente Doutor Daniel Silvestre, cujo intuito é criar uma plataforma de previsão de resultados desportivos e um modelo inovador que permitirá ao utilizador ter uma vantagem sobre os restantes apostadores podendo escolher o seu onze inicial.

Vão ser realizados três modelos de previsão para apostas desportivas que irão incidir sobre as 5 principais ligas europeias: Liga NOS (Portugal), Premier League (Inglaterra), La Liga (Espanha), Bundesliga (Alemanha) e Serie A (Itália).

O primeiro modelo ao qual iremos chamar “previsão de odds com base nas estatísticas” consiste em fazer-se uma recolha das estatísticas já existentes sobre as equipas que queiramos fazer uma previsão.

Esta recolha vai ser efetuada através de algoritmos de *web scraping* da biblioteca python “Selenium”[2] e os dados serão retirados do site <http://www.meusresultados.com>.

Este script vai recolher e guardar os dados que lhe sejam pedidos num ficheiro .csv que posteriormente será usado para fazer as previsões.

As previsões vão ser feitas com base num algoritmo classificador “Naive Bayes” da biblioteca *scikit learn*.

O algoritmo “Naive Bayes” é um classificador probabilístico baseado no “Teorema de Bayes”, o qual foi criado por Thomas Bayes. Por ser muito simples e rápido, possui um desempenho relativamente maior do que outros classificadores. Além disso, o Naive Bayes só precisa de um pequeno número de dados de teste para concluir classificações com uma boa precisão.

O teorema de Bayes incide-se na probabilidade condicional. Isto é, qual a probabilidade de o evento A ocorrer, dado o evento B.[3]

Uma vez que os dados do ficheiro .csv estejam agrupados e prontos a serem usados (remover dados desnecessários para o teste e transformar os dados quantitativos em dados percentuais), recorre-se ao modelo Naive Bayes, que por sua vez irá dar o *target* que é pretendido. Este *target* vai ser convertido para o formato de uma *odd*. Por fim, vão ser apresentadas *odds* de um determinado jogo entre duas equipas, para a vitória, empate ou derrota.

O segundo modelo denomina-se de “previsão de odds com base em odds prévias”. As *odds* prévias serão recolhidas de diversos sites de apostas já conhecidos.

Esta recolha vai ser efetuada através de algoritmos de *web scraping* da biblioteca python “Selenium” e os dados serão retirados do site <http://www.meusresultados.com>.

Este script vai recolher e guardar os dados que lhe sejam pedidos num ficheiro .csv, que mais à frente, será usado para fazer as previsões.

As previsões vão ser feitas com base num algoritmo classificador Naive Bayes da biblioteca scikit learn.

Uma vez que os dados do ficheiro .csv estejam agrupados e prontos a serem usados (remover dados desnecessários para o teste e transformar os dados quantitativos em dados percentuais), recorre-se ao modelo Naive Bayes que por sua vez irá dar o target que é pretendido. Este target vai ser convertido para o formato de uma *odd*. Por fim, vão ser apresentadas *odds* de um determinado jogo entre duas equipas, para a vitória, empate ou derrota.

O terceiro modelo, denominado de "onze titular", não se baseará em estatísticas da equipa (como um todo coletivo) nem em *odds* antigas, mas sim em estatísticas individuais de cada jogador, que serão recolhidas do site <http://www.meusresultados.com> através de algoritmos de *web scraping* da biblioteca python "Selenium", estes dados serão guardados num ficheiro .csv que posteriormente serão usados para a previsão.

O utilizador vai ter a vantagem de poder escolher os seus "onzes" titulares, tanto da equipa de casa, como da equipa visitante. Cada jogador vai ter as suas estatísticas, e após os 11 escolhidos, será feita uma média entre todos eles para serem analisados como uma equipa.

No final, será dada a probabilidade de cada um dos destinos de jogo, tanto para a vitória de cada um dos "onzes" como também para o empate.

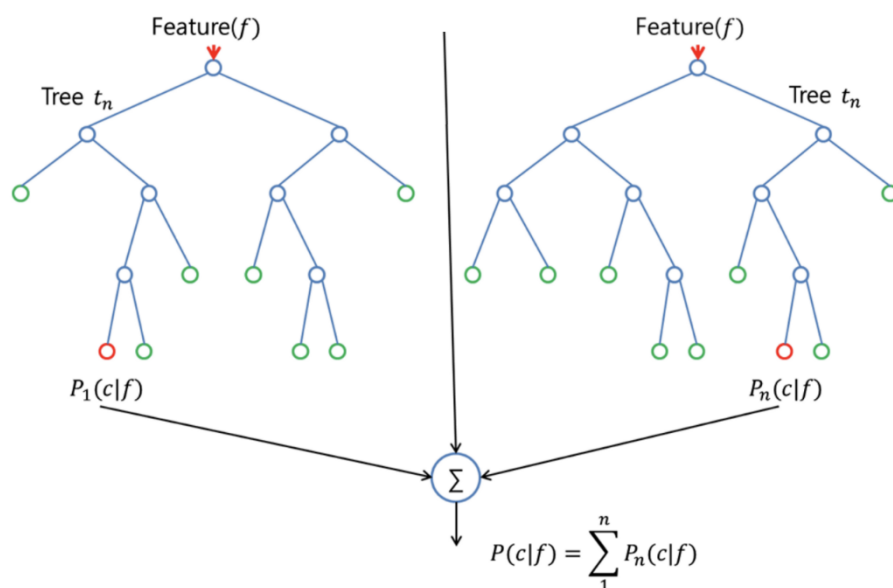
Todo este modelo utiliza o classificador Random Forest.

Random Forest é um algoritmo de aprendizagem flexível e fácil de usar que produz excelentes resultados a maioria das vezes, mesmo sem ajuste de hiper parâmetros.

Como se pode perceber pelo nome, vai ser criada uma “floresta” de um modo aleatório. A “floresta” é uma combinação de árvores de decisão, na maioria dos casos treinados com o método de *bagging*. A ideia principal deste método é que a combinação dos modelos de aprendizagem aumente o resultado geral.

Uma grande vantagem deste algoritmo é que ele pode ser utilizado tanto para tarefas de classificação como de regressão, o que representa a maioria dos sistemas de aprendizagem atuais.[4]

Abaixo, pode-se observar o algoritmo com duas árvores:



“Fig.1 – Grafico com duas árvores”

Imagem exemplificadora de como funciona o algoritmo Random Forest

O classificador usa como “train” estatísticas de jogos aleatórios, que dará uma ideia ao mesmo de como são as estatísticas de uma equipa vencedora ou perdedora. Baseando-se nestes jogos anteriores, consegue utilizar os dois novos

"onzes" e com as médias já feitas dar-nos-á uma resposta de qual deles tem uma maior probabilidade de vencer.

Para além deste três modelos, vai ser desenvolvido uma *web application*.

Nesta *web application*, para além de futuros jogos inseridos num calendário da corrente época, estarão apresentados os *outputs* dos dois primeiros modelos (*odds* de futuros jogos).

Estará disponível ainda o terceiro modelo, apresentado de forma interativa para que o utilizador consiga escolher o seu onze e desta forma poder interagir com um modelo de previsão mais complexo, criando assim uma vantagem sobre outros apostadores que na sua maioria apenas têm acesso às *odds* finais que são apresentadas nas casas de apostas.

Objectivos

Para este projeto traçámos um objetivo principal de criar um site que disponibilizasse vários modelos de previsão desportiva com o intuito de ajudar o apostador.

De modo a que não fosse apenas mais um site, quisemos introduzir algo inovador.

Temos como meta desenvolver um modelo de previsão de resultados com base nas *odds* de jogos anteriores, um segundo modelo com base nas estatísticas de cada equipa e por fim, introduzir algo novo e menos comum no mercado das apostas, um modelo que se baseia nas estatísticas dos jogadores que entrariam em campo como onze titular, e o impacto que estes tem nas suas respetivas equipas.

O site, quando composto por estes três modelos irá fornecer ao apostador uma experiência nova, tendo ao seu dispor vários modelos para o ajudar a escolher a melhor aposta.

Vamos também disponibilizar uma api que permitirá a um administrador ter acesso às bases de dados bem como alterar os respetivos dados.

Por fim temos como objetivo também criar uma aplicação para android de modo a ser possível aceder aos nossos modelos de uma forma remota, cómoda e portátil, tornando assim o acesso mais fácil para todos e em qualquer lugar.

Dificuldades encontradas

A primeira dificuldade encontrada foi ao fazer o *scrape* dos dados que iriam ser necessários.

Começámos por eleger os sites de onde queríamos fazer o *scrape*, optámos por utilizar o *beautiful soup*[5], mas depressa nos deparámos com vários obstáculos, muitos deles relacionados com problemas de interação entre o *beautiful soup* e o próprio site. Isto deve-se ao facto de bastantes sites terem uma proteção para que os dados não sejam retirados por terceiros. Após alguma reflexão e pesquisa adotámos uma estratégia diferente, que passou para utilizar o *ChromeDriver*[6] para fazer o *scrape* dos dados. O *ChromeDriver* tinha muito menos problemas de interação com os sites a nível de java e html, e também optámos por fazer vários *scrapes* com um menor número de dados e mais tarde juntar todos num *Dataframe*, ultrapassando assim o obstáculo que os sites tinham, que tornava bastante difícil a um terceiro retirar os dados que estão visíveis.

Uma vez que os classificadores usados funcionam com estatísticas de jogos que já se realizaram, e os analisam com o intuito de fazer uma previsão, alguns eventos desportivos não terão uma previsão realista. Isto acontece devido ao facto de que algumas equipas ou jogadores ainda não possuem dados suficientes para que seja feita uma previsão com precisão.

Para que as previsões estejam o mais corretas possível, é necessário que os modelos usem os dados mais recentes, isto revelou ser um grande obstáculo.

Para ultrapassarmos esta dificuldade decidimos implementar uma máquina virtual que o utilizador apenas precisa de ligar uma vez por semana e que posteriormente irá fazer o *scrape* de todos os dados novamente, permitindo assim aos modelos terem acesso aos dados mais recentes.

Tínhamos como objetivo adicionar um extra, que para além da *web application*, iríamos criar uma app para android, que tornaria o modelo ainda mais acessível.

Contudo, acabámos por não ter muito tempo para realizar esta tarefa, e então preferimos aperfeiçoar a *web application* e os modelos para que a previsão fosse a melhor possível.

Implementação do Projecto

Numa altura em que cada vez existem mais apostadores, o nosso intuito foi criar algo que pudesse ajudar todos aqueles que pretendam apostar em eventos desportivos, de modo a que pudessem fazer uma aposta com uma maior probabilidade de acertarem.

Decidimos então mostrar com este projeto, que através de algoritmos de *machine learning* podemos introduzir algo novo no mercado das apostas desportivas.

Optámos por escolher o futebol, devido a ser o desporto mais apostado tanto na Europa como em Portugal, isto não impede que os modelos desenvolvidos por nós não se possam expandir a mais desportos, antes pelo contrário, uma vez que a “base” dos modelos está criada, apenas é preciso recriar as mesmas condições para qualquer desporto que se queira implementar (Basquetebol, futsal, andebol, etc.). Ou seja, seria preciso apenas recolher a informação requerida para que os modelos possam fazer as previsões, estamos a falar de informação estatística de cada equipa e dados individuais de cada jogador.

Este conjunto de classificadores pode tanto ser usados em separado como em conjunto, com o intuito de fornecer mais previsões sobre um evento com base em diferentes features, o utilizador iria depois escolher aquela que lhe parece a mais acertada dependendo dos últimos acontecimentos com as equipas em questão.

Existem imensas casas de apostas, portanto fazer mais uma nunca foi a nossa intenção, ainda para mais porque a burocracia necessária para iniciar um negócio de apostas online é imensa e tornar-se-ia num processo desgastante.[7]

Como tal a melhor maneira de implementar estas ferramentas para que os apostadores tenham acesso a elas, seria através de um site (*web application*), independente de qualquer casa de apostas.

Decidimos então criar um site que serve de protótipo para o que estaria disponível ao público.

O utilizador tem acesso a resultados anteriores, calendário da época desportiva, bem como notícias sobre vários desportos, contudo, para ter acesso aos modelos de previsão é necessário que o utilizador se registe no site.

Após ser feito o registo o utilizador tem então acesso aos modelos, pode ainda selecionar as previsões (*odds*) feitas por um modelo sobre um determinado evento e guardá-las para futura análise.

Para que tanto os modelos como o site estivessem sempre atualizados, iria ser necessário alugar um servidor para o site, e comprar um serviço anual a uma empresa que fornece estatísticas tanto de equipas como de jogadores para todos os desportos, claro que esta manutenção iria ter custos.

Para colmatar estes custos, iria ser cobrada uma mensalidade (por exemplo de 5,00 euros) para que o utilizador pudesse ter acesso aos modelos de previsão.

Para quem não esteja dentro do “ramo” das apostas, pagar uma mensalidade pode parecer algo arriscado e sem sentido, no entanto é preciso ter em conta que uma grande maioria dos apostadores não tem conhecimentos de *machine learning*, muito menos acesso a um modelo de previsão desportiva, quanto mais três.

Atualmente existem fóruns em que apostadores mencionam o facto de terem adquirido ou feito um modelo de previsão de resultados, mas não estão dispostos a abdicar da vantagem que têm, ao partilhar o modelo com outros apostadores, mantendo assim o uso de modelos de previsão muito restrito.[8]

Com o serviço disponibilizado por nós, e por uma mensalidade acessível, o apostador teria acesso a três modelos.

Isto vai então criar uma vantagem enorme para este apostador, que possui informação, que os seus “rivals” (outros apostadores e as casas de apostas) não possuem ou não pretendem divulgar ao público, e facilmente iria conseguir não só reaver o dinheiro da mensalidade como aumentar os seus ganhos.

É importante referir que estes modelos são uma ajuda, e apesar de terem um nível de previsão bom, funciona melhor ainda quando o próprio utilizador tem conhecimento sobre o evento que vai apostar.

Manual do Utilizador

Neste módulo o utilizador pode encontrar um guia com todos os procedimentos necessários para poder aceder à plataforma.

Este guia divide-se em três módulos:

- Pré-requisitos de software – Consiste em todos os downloads e extração dos ficheiros em anexo necessários com o intuito de criar um ambiente disponível para a *Web application* e manter os dados atualizados da plataforma;
- Instalações e configurações prévias – Neste grupo o utilizador deverá fazer ajustes à própria máquina com o objetivo de remover qualquer irregularidade que comprometa a *Web application*;
- *Web Application* – Este módulo explicita todas as funcionalidades da *Web application*.

Pré-requisitos de software

O utilizador deverá de ter na sua máquina os seguintes programas e ficheiros:

- XAMPP (Download disponível em: <https://www.apachefriends.org/download.html>)
 - O utilizador deverá fazer o download da versão mais adequada e recente para a sua máquina.
 - Depois da transferência o utilizador deverá de correr uma instalação normal e a mais recomendada.
- Oracle Virtual Box (Download disponível em: <https://www.virtualbox.org/wiki/Downloads>):
 - O utilizador deverá fazer o download da versão mais adequada para a sua máquina.
 - Depois da transferência o utilizador deverá de correr uma instalação normal e a mais recomendada.
- O utilizador também deverá ter os ficheiros que seguem anexo no CD:
 - Máquina virtual
 - Pasta “bets” que contem os ficheiros necessários para o site

- Ficheiros .py para testes
- Ficheiro .sql com os dados da BD
- O utilizador também deve de instalar o *python* (Download disponível em: <https://www.python.org/downloads/release/python-360/>)
 - **Nota:** O projeto foi desenvolvido na versão 3.6 do *python*, para isso também é recomendado que o utilizador instale a mesma versão para evitar pacotes não compatíveis.
 - O utilizador deve de ir a parte inferior do site onde faz o download.

Files					
Version	Operating System	Description	MD5 Sum	File Size	GPG
Gzipped source tarball	Source release		9a080a86e1a8d85e45eee4b1cd0a18a2	22930752	SIG
XZ compressed source tarball	Source release		c3f30a0aff425dda77d19e02f420d6ba	17156744	SIG
macOS 64-bit/32-bit installer	Mac OS X	for Mac OS X 10.6 and later	c58267cab96fd291d332a2b163edd33	28060853	SIG
macOS 64-bit installer	Mac OS X	for OS X 10.9 and later	3ad13cc51c488182ed21a50050a38ba7	26954940	SIG
Windows help file	Windows		e01b52e24494611121b4a866932b4123	8139973	SIG
Windows x86-64 embeddable zip file	Windows	for AMD64/EM64T/x64	7148ec14edfcd13f42e06a14d617c921	7186734	SIG
Windows x86-64 executable installer	Windows	for AMD64/EM64T/x64	767db14ed07b245e24e10785f9d28e29	31930528	SIG
Windows x86-64 web-based installer	Windows	for AMD64/EM64T/x64	f30be4659721a0ef68e29cae099fed6f	1319992	SIG
Windows x86 embeddable zip file	Windows		b4c424de065bad238c71359f3cd71ef2	6401894	SIG
Windows x86 executable installer	Windows		467161f1e894254096f9a69e2db3302c	30878752	SIG
Windows x86 web-based installer	Windows		a940f770b4bc617ab4a308ff1e27abd6	1293456	SIG

- Deverá instalar a seguinte versão: “Windows x86-64 executable installer”(Caso o sistema operativo da maquina seja Windows).
- Durante a instalação é importante anotar que é necessário assinalar a opção “Add Python 3.6 to Path”



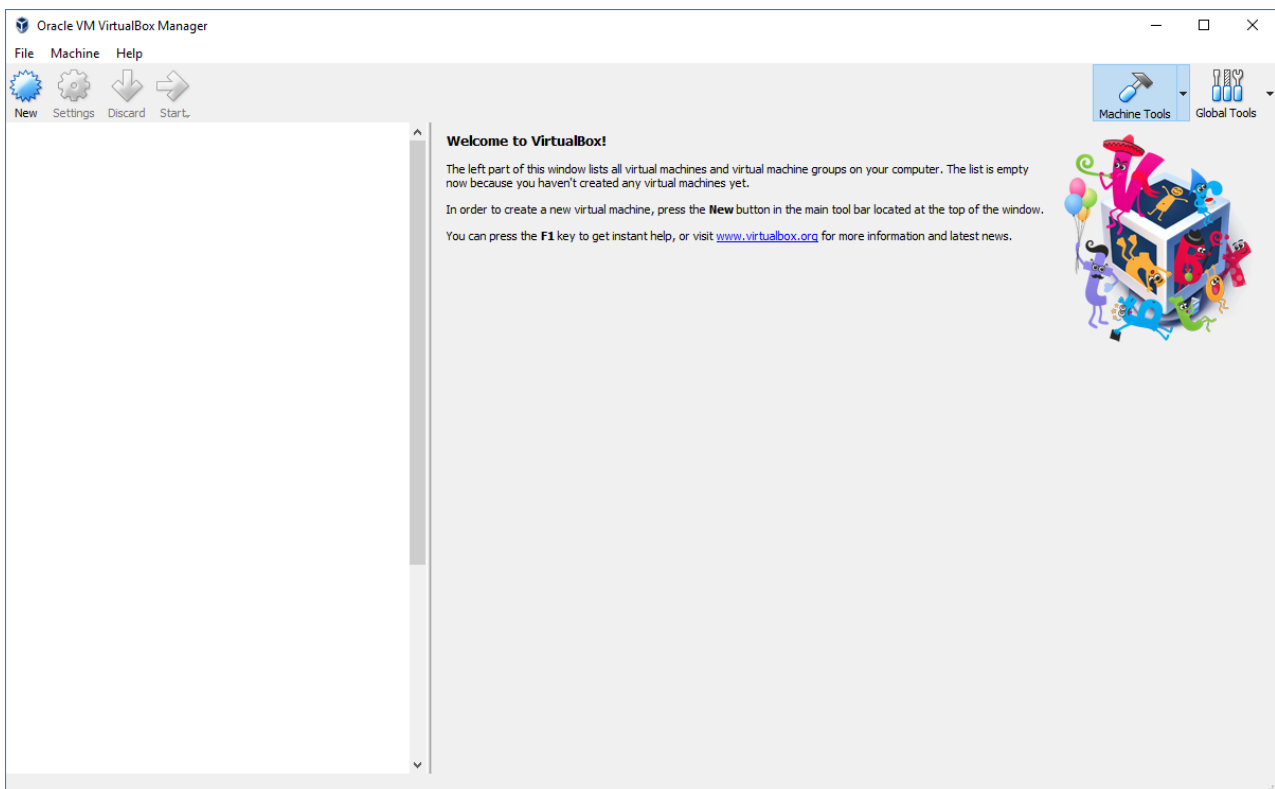
- De seguida deverá selecionar a opção “Install Now” e seguir a instalação da forma mais recomendada e normal.

Instalações e configurações antes de correr o programa

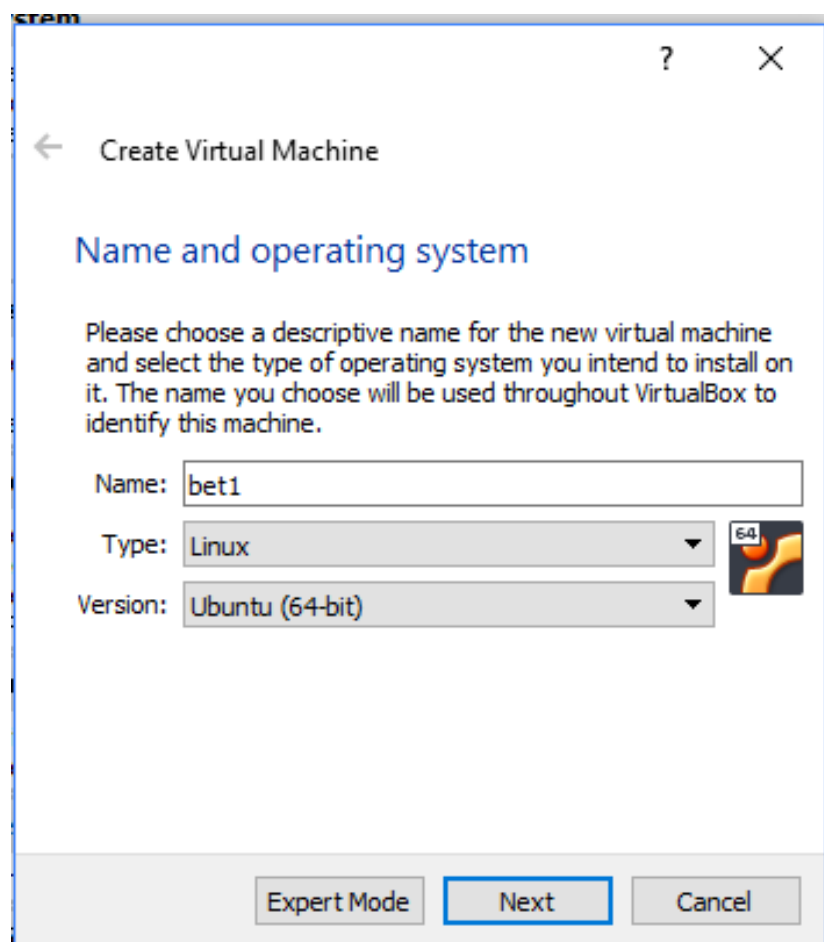
Após as instalações dos programas terem corrido sem irregularidades, o utilizador deverá fazer alguns ajustes em certos programas.

- Oracle Virtual Box

A máquina virtual é usada com o objetivo de manter os dados (próximas jornadas, resultados, notícias, estatísticas e classificações) na base de dados atualizados e os futuros cálculos das *odds* sempre atualizados



1. Deverá abrir o programa e clicar na opção “New” para criar uma nova máquina virtual de seguida escolha o nome que quer dar à máquina virtual e indique o sistema operativo da máquina, deve escolher Linux Ubuntu-64 bits;



2. De seguida indique quanta memória da sua máquina “Host” quer alocar para a máquina virtual (este passo dependa de máquina para máquina);

The screenshot shows a Windows-style dialog box titled 'Create Virtual Machine'. The 'Memory size' section is active, showing a slider and a text box for memory allocation. The recommended size is 1024 MB, and the current selection is 2048 MB. The 'Next' button is highlighted with a blue border.

Create Virtual Machine

Memory size

Select the amount of memory (RAM) in megabytes to be allocated to the virtual machine.

The recommended memory size is **1024 MB**.

4 MB 8192 MB

2048 MB

Next Cancel

3. Selecionar a opção “Do not add a virtual hard disk” e carregar em “create” e posteriormente “continue”;

?

×

← Create Virtual Machine

Hard disk

If you wish you can add a virtual hard disk to the new machine. You can either create a new hard disk file or select one from the list or from another location using the folder icon.

If you need a more complex storage set-up you can skip this step and make the changes to the machine settings once the machine is created.

The recommended size of the hard disk is **10,00 GB**.

☒ Do not add a virtual hard disk

☐ Create a virtual hard disk now

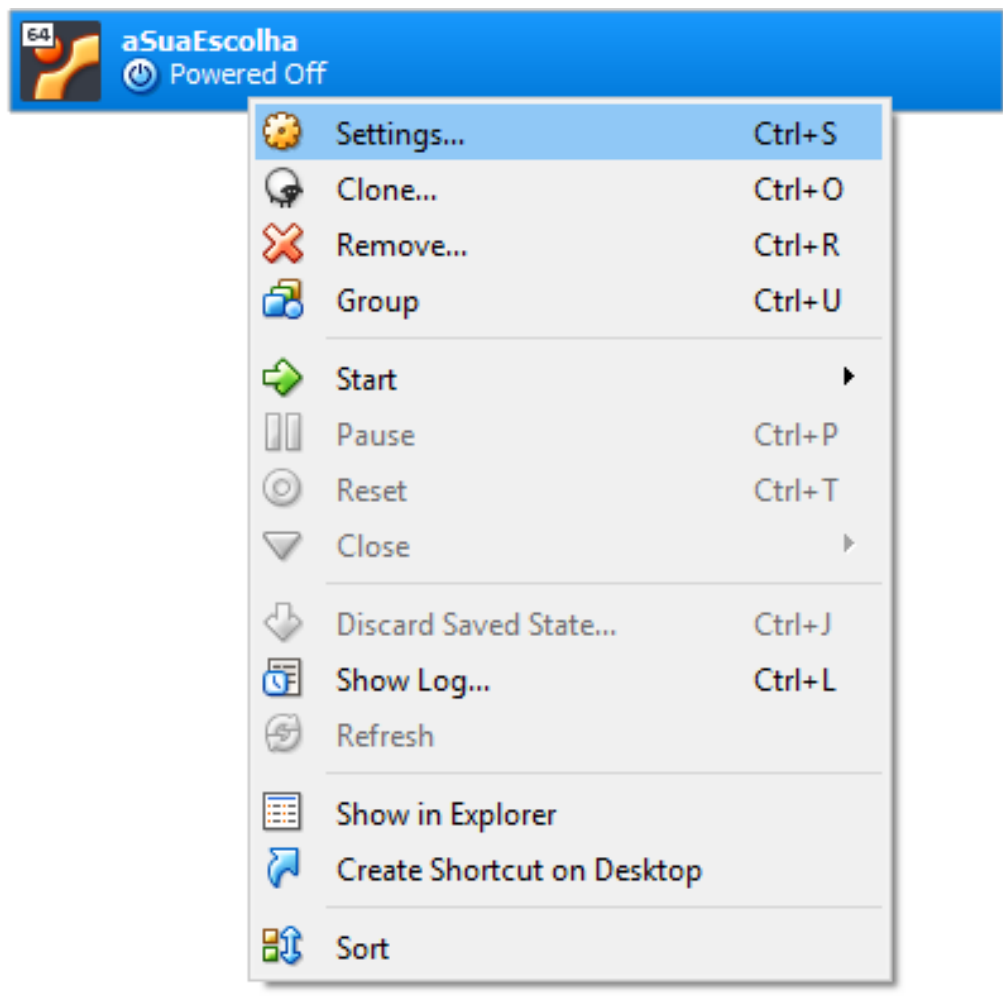
☐ Use an existing virtual hard disk file

Empty ▾ 

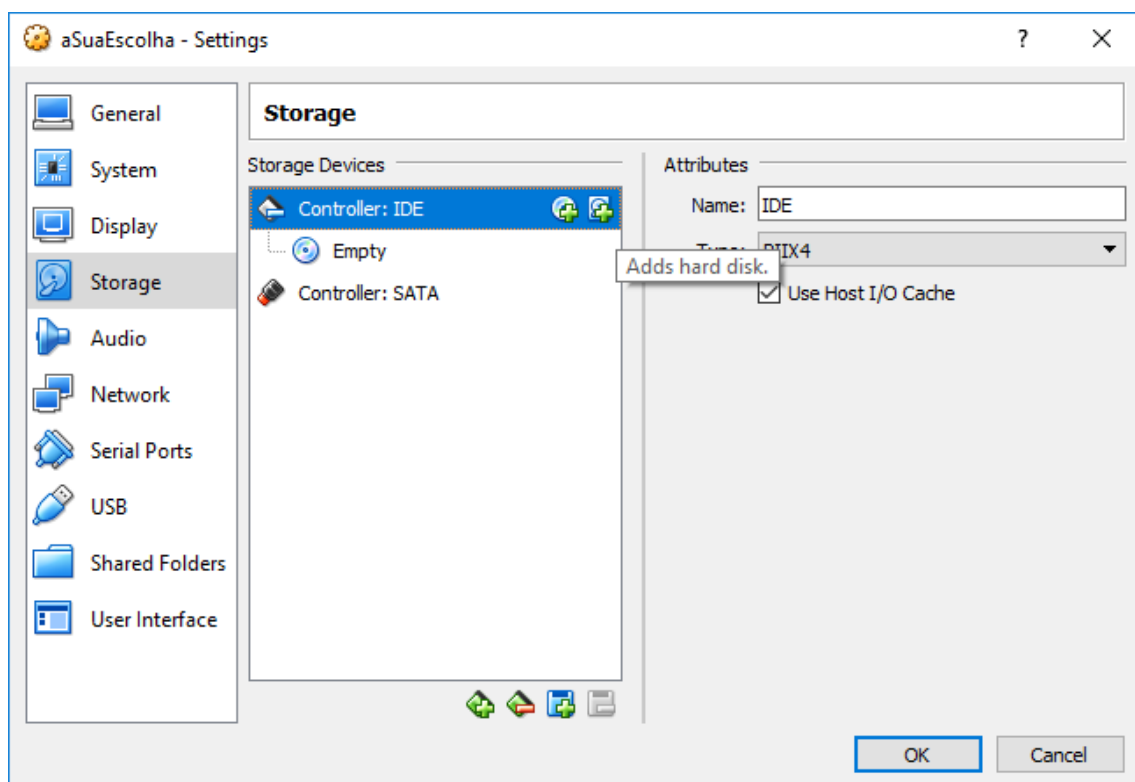
Create

Cancel

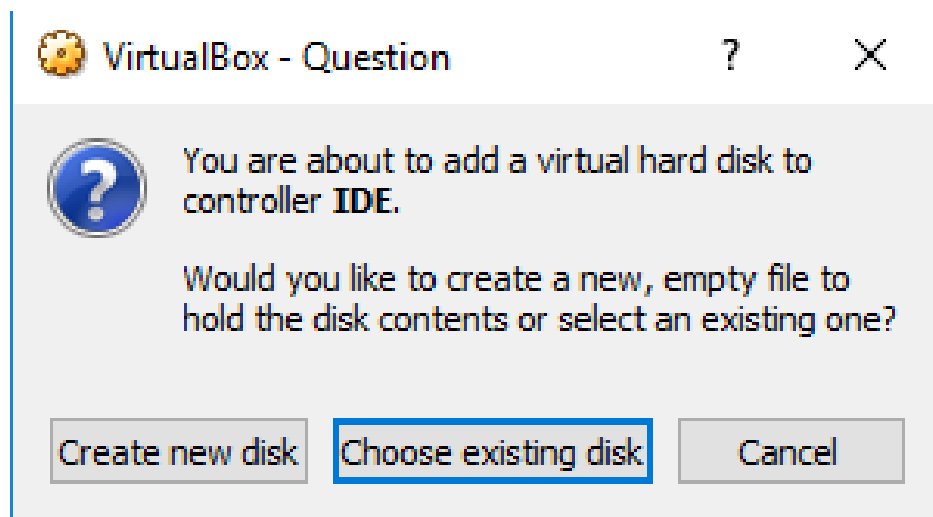
4. No passo seguinte selecione a opção “Settings” -> “Storage controller”;



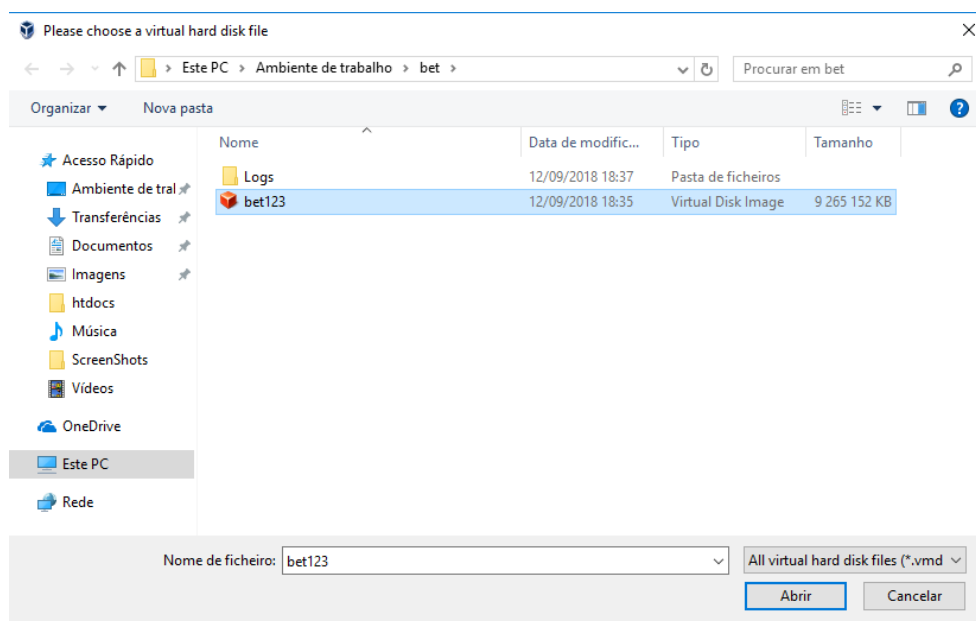
5. Selecionar “adds Hard disk”;



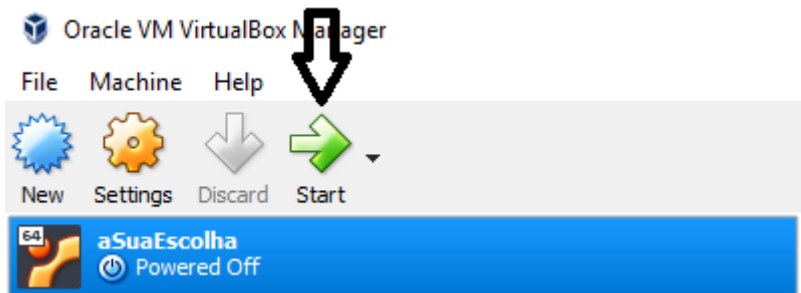
6. Escolher “choose existing disk”;



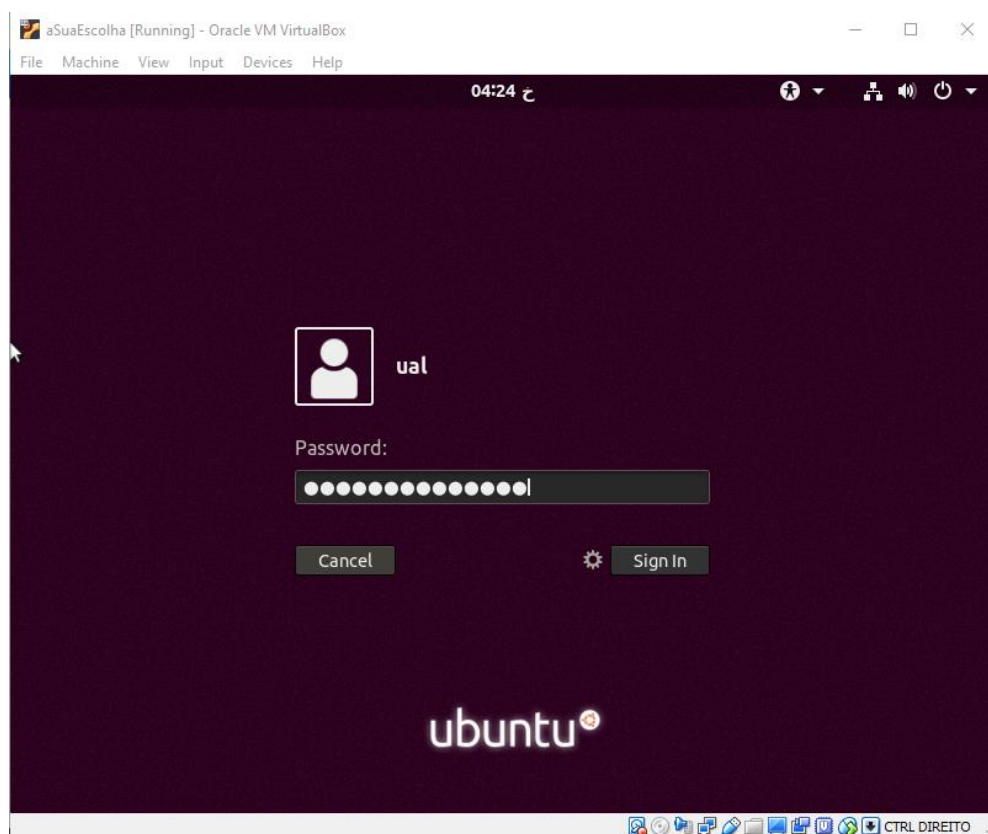
7. Escolher o ficheiro bet123;



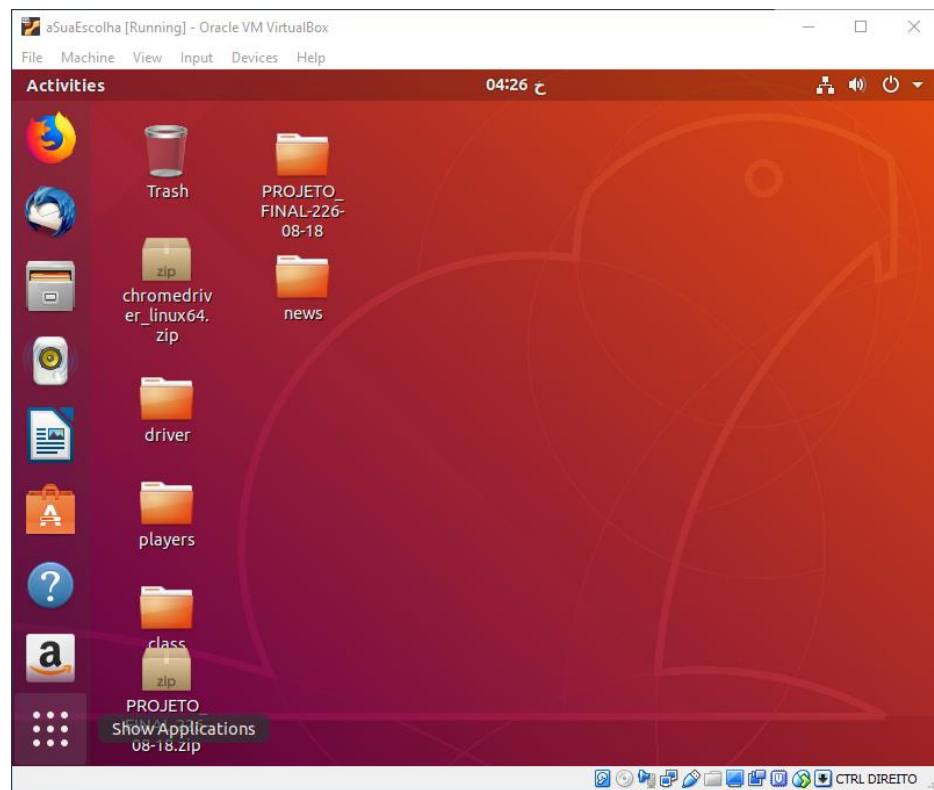
8. Iniciar a máquina virtual com o botão “start”;



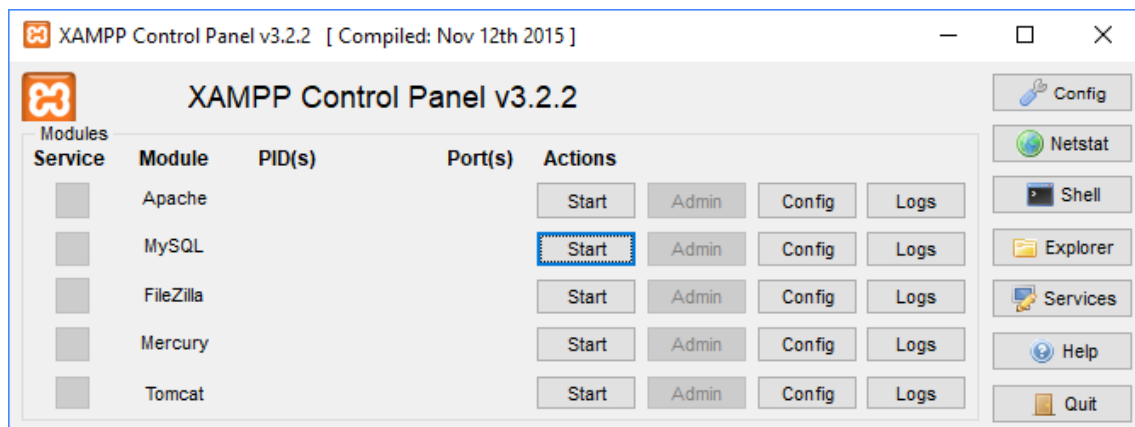
9. Selecionar o utilizador ual e inserir a palavra pass: Ual-1234567890;



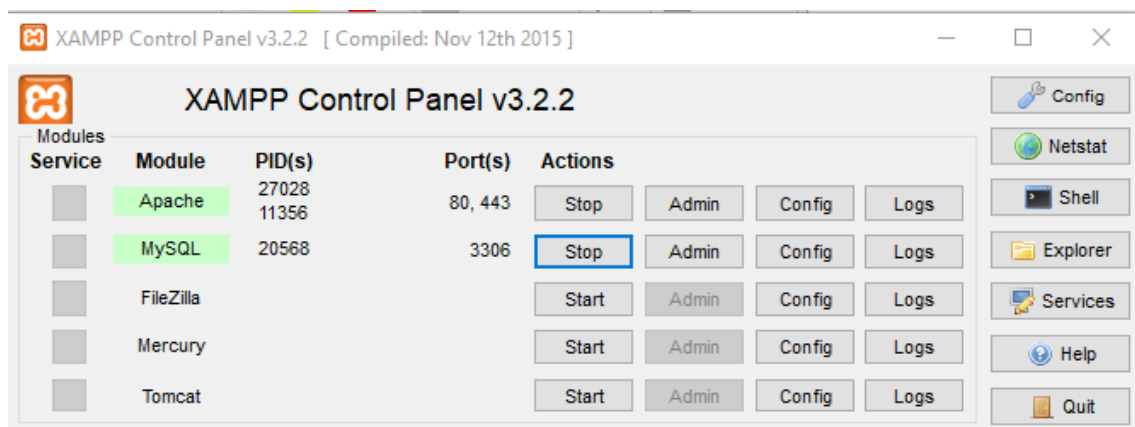
10. Este é o aspeto que a sua máquina virtual deverá ter, após efetuar o login.



- XAMPP – APACHE WEB SERVER
 1. Deverá abrir o XAMPP CONTROL PANEL e selecionar a opção “Start” das linhas referentes ao APACHE e ao MySQL



- Deverá aguardar até que os serviços estejam iniciados aparecendo da seguinte forma:

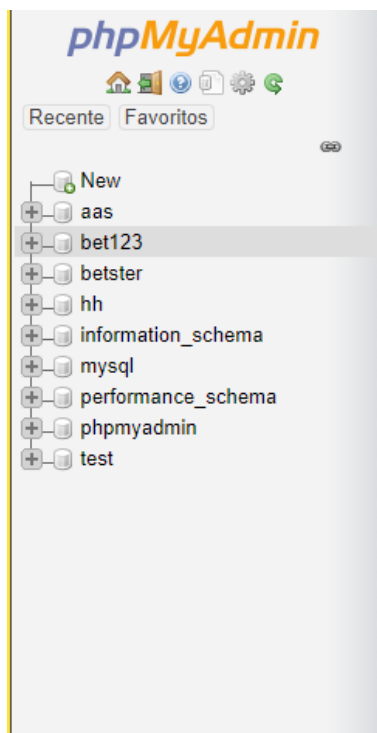


- De seguida deverá seleccionar a opção presente na aba lateral direita : “Explorer” e ir ao seguinte diretório : “C:\xampp\htdocs”(esta pasta o utilizador encontra todos os ficheiros relacionados com cada site) e deve de extrair do CD a pasta comprimida referente à *Web application* com o nome “bets”

- XAMPP – MySQL Server

1. Com os ficheiros já presentes na pasta “htdocs” o utilizador deverá agora abrir o seu navegador e ir ao endereço:
<http://localhost/phpmyadmin/>

2. Dentro do “phpMyAdmin” é necessário criar a Base de dados, para isso clique na opção “New” presente no menu lateral esquerdo



3. Dentro do menu de criação da base de dados deverá criar uma base de dados com a seguinte configuração:

Base de Dados

 Criar base de dados 

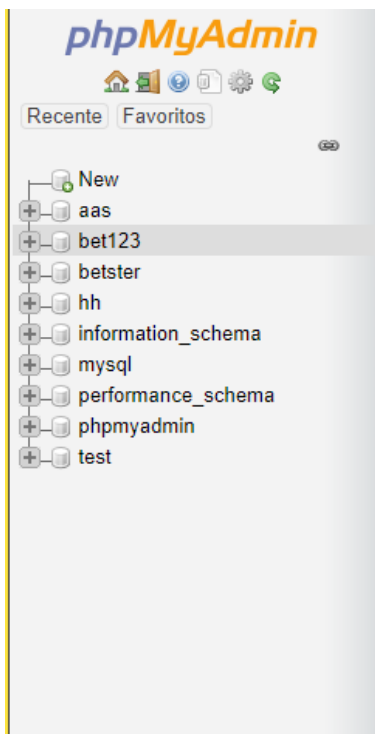




4. O nome da base de dados será obrigatoriamente “bet123” e a sua codificação “latin1_swedish_ci”

5. De seguida clique na opção “Criar”

6. O utilizador pode aceder a base de dados seleccionando no menu lateral a BD “bet123”



7. Dentro deste menu o utilizador pode visualizar as tabelas criadas e realizar algumas operações sobre elas(que após a criação não existem)

Servidor: 127.0.0.1 » Base de Dados: bet123

Estrutura SQL Pesquisar Pesquisa por formulário Exportar Importar Operações Privilégios Rotinas Eventos Mais

Filters

Containing the word:

Tabela	Acções	Registos	Tipo	Agrupamento (Collation)	Tamanho	Suspensão
☐ aposta_odd	★ Procurar Estrutura Pesquisar Inserir Limpar Eliminar	17	InnoDB	latin1_spanish_ci	16 KB	-
☐ aposta_pt	★ Procurar Estrutura Pesquisar Inserir Limpar Eliminar	78	InnoDB	latin1_spanish_ci	16 KB	-
☐ a_bola	★ Procurar Estrutura Pesquisar Inserir Limpar Eliminar	27	InnoDB	latin1_spanish_ci	80 KB	-
☐ categories	★ Procurar Estrutura Pesquisar Inserir Limpar Eliminar	0	MyISAM	latin1_spanish_ci	1 KB	-

8. Agora clique na opção “Importar” da barra de ferramentas do topo da pagina

9. Aqui deverá dar upload ao ficheiro .sql referente á base de dados contidos no CD em anexo.

File to import:

O ficheiro pode ser comprimido (gzip, bzip2, zip) ou descomprimido.
A compressed file's name must end in `.[format].[compression]`. Example: `.sql.zip`

Procurar no seu computador: Nenhum ficheiro selecionado (Tamanho máximo: 2,048KB)

You may also drag and drop a file on any page.

Configurar o Mapa de Caracteres do ficheiro:

Partial import:

☒ Allow the interruption of an import in case the script detects it is close to the PHP timeout limit. *(This might be a good way to import transactions.)*

Skip this number of queries (for SQL) starting from the first one:

Other options:

☒ Ativa a verificação de chaves estrangeiras

Formato:

10. Clique na opção “Executar”

11. Por fim, deverá dar permissões para escrita e leitura da tabela para isso o utilizador deverá saber o seu “IPv4 Address”, poderá saber abrindo o terminal e executar o comando “ipconfig” e anotar o IPV4 Address.

12. Clique na opção “SQL” presente na barra de ferramentas do topo da pagina e execute o seguinte comando:

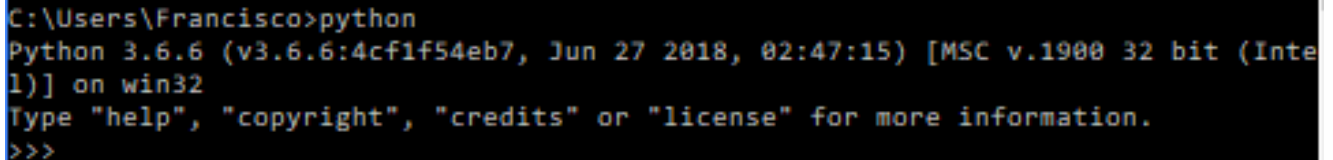
```
• GRANT ALL PRIVILEGES ON bet123.* TO root
  @ [IPV4 Address da maquina] IDENTIFIED B
  Y “ola” WITH GRANT OPTION;
```

13. Clique na opção executar

- Python - Verificação das instalações dos pacotes

O utilizador deverá testar os ficheiros .py previamente para que seja possível eles serem executados na *Web application*. Estes testes funcionam como verificação da instalação dos pacotes. Para isso o utilizador deve:

1. Abrir o terminal;
2. Digitar o comando “python”;
3. Verificar que a versão do python instalada é a 3.6.X;



```
C:\Users\Francisco>python
Python 3.6.6 (v3.6.6:4cf1f54eb7, Jun 27 2018, 02:47:15) [MSC v.1900 32 bit (Intel)] on win32
Type "help", "copyright", "credits" or "license" for more information.
>>>
```

4. De seguida digite “exit()” para sair da Shell do python;
5. Para melhor conveniência deverá ter os ficheiros .py dos testes numa pasta no ambiente de trabalho;
6. Aceder via terminal ao diretório onde estão presentes os ficheiros (ver proximo passo);
7. Caso estejam presentes numa pasta no Ambiente de trabalho correr os seguintes comandos
 - cd Desktop
 - cd [Nome_da_pasta]

8. Já dentro da pasta o utilizador vai verificar se tem todos os pacotes instalados correndo um dos ficheiros .py contidos nessa pasta para isso o utilizador deve correr o seguinte comando (Devera ocorrer um erro caso seja a primeira vez que o utilizador corre o programa):

- `python odds_pt1.py`

9. No caso de recém instalações do python o utilizador deverá correr os seguintes comandos

- `pip install -U scikit-learn`
- `pip Install pandas`
- `pip install scipy`
- `pip install SQLAlchemy`
- `pip install mysql-connector`

10. O utilizador deverá abrir o web browser, e colocar na barra de url o endereço: <http://localhost/phpmyadmin/> de seguida, na barra de ferramentas seleccionar a opção “SQL” e introduzir os seguintes comandos (em separado) e clicar no botão executar para correr o comando, uma mensagem a verde a dizer: “MySQL não retornou nenhum registo” deverá aparecer, isto indica que o comando foi inserido com sucesso.

- 1- `CREATE USER 'root'@"%" IDENTIFIED BY "ola";`
- 2- `GRANT ALL PRIVILEGES ON *.* TO 'root'@"%";`

11. Caso todas as instalações dos pacotes tenham corrido sem irregularidades correr outra vez o comando “python odds_pt.py” no cmd, e deverá ser devolvido uma probabilidade percentual para 1(vitória da equipa de casa), X (Empate) e 2(Vitória da equipa visitante).

Isto comprova que o script está a correr de forma correta e o utilizador pode iniciar a *Web application*.

12. O utilizador pode fechar o terminal

Web Application

Já com todas as transferências, configurações e instalações efetuadas o utilizador pode iniciar a *Web Application* no seu browser predefinido. Para isso é necessário:

1. Insira na barra de navegação o seguinte link:
 - <http://localhost/bets/>
2. Uma vez no site o utilizador depara-se com as seguintes funcionalidades: Calendário, Resultados, Noticias, Classificação e equipas, modelos de previsão desportiva e Login/Registo
 - Calendário
 - Dentro desta funcionalidade o utilizador pode escolher qual das ligas disponíveis pretende visualizar



- Após escolher a liga será apresentado o calendário das próximas jornadas
- Resultados
 - À semelhança do calendário, no interior da página de resultados o utilizador pode escolher qual das ligas disponíveis pretende ver

- Após escolher a liga será apresentado os resultados das jornadas passadas
- Notícias
 - Nas notícias é possível encontrar duas fontes de informação sobre futebol e ainda um canal de *Youtube* com golos e *highlights* dos jogos
- Classificação e equipas
 - Na funcionalidade Classificação e equipas o utilizador pode escolher a classificação da liga que pretende visualizar



- Após a seleção da liga é possível ver a classificação de cada equipa



UAL BETS

Calendário

Resultados

Noticias

Classificação e equipas

Modelos de previsão desportiva

	Posição	Equipa	Pontos	Jogos jogados	Vitorias	Empates	Derrotas	Golos marcados	Golos sofridos	Diferença de golos
	1	Sporting	10	4	3	1	0	7	3	4
	2	Benfica	10	4	3	1	0	10	3	7
	3	Braga	10	4	3	1	0	11	6	5
	4	FC Porto	9	4	3	0	1	13	5	8
	5	Marítimo	9	4	3	0	1	5	4	1
	6	Feirense	7	4	2	1	1	4	2	2
	7	Rio Ave	7	4	2	1	1	6	5	1
	8	V. Guimarães	6	4	2	0	2	6	6	0
	9	Santa Clara	5	4	1	2	1	9	8	1
	10	Belenenses SAD	5	4	1	2	1	4	4	0
	11	V. Setúbal	4	4	1	1	2	4	4	0
	12	Boavista	4	4	1	1	2	5	7	-2
	13	Moreirense	4	4	1	1	2	4	8	-4

- E também visualizar informação estatística sobre os jogadores dessa equipa.



UAL BETS

Calendário

Resultados

Notícias

Classificação e equipas

	Nome	Jogos	Golos marcados	Média de golos	Jogos a titular	Jogos a suplente	Minutos	Amarelos	Duplos amarelos	Vermelhos	Nacionalidade
	Alhassan Wakaso	5	1	0.2	5	-	450	3	-	-	Gana
	Tyler Boyd	5	0	0	4	1	317	1	-	-	Nova Zelândia
	João Carlos Teixeira	5	0	0	4	1	292	1	-	-	Portugal
	Davidson	5	1	0.2	2	3	236	-	-	-	Brasil
	Ola John	5	0	0	2	3	216	-	-	-	Holanda
	André André	4	2	0.5	4	-	360	-	-	-	Portugal
	João Afonso	4	0	0	4	-	360	-	-	-	Portugal
	Douglas	4	0	0	4	-	360	-	-	-	Brasil
	Pedro Henrique	3	0	0	3	-	270	-	-	-	Brasil
	Florent Hanin	3	0	0	3	-	270	1	-	-	França
	Dodô	3	0	0	3	-	270	2	-	-	Brasil
	Welthon	3	0	0	3	-	241	-	-	-	Brasil
	Tozé	3	1	0.33	1	2	162	-	-	-	Portugal

- Login/Registo
 - Para novos utilizadores é necessário o registo na plataforma, para isso clique em “Login/Registo” e de seguida em “Registe-se aqui”(NOTA: o *username* funciona como identificador único fazendo com que seja impossível haver utilizadores com o *username* idêntico)

- Uma vez que o registo esteja efetuado o utilizador pode efetuar o login com o *username* e *password* definidos



A screenshot of a 'Log In' form. The form is a light blue rectangle with a thin border, centered on a background of green grass. It features the title 'Log In' in bold black text at the top. Below the title are two white input fields: 'Username' and 'Password'. To the right of the 'Password' field is a blue button with the text 'Login' in white. Below the input fields is a blue link that says 'Registe-se aqui'.

- Modelos de previsão desportiva
 - Após o login o utilizador tem acesso aos modelos de previsão desportiva (<http://localhost/bets/Profile.php>)



A screenshot of a 'Registo' (Registration) form. The form is a light blue rectangle with a thin border, centered on a background of green grass. It features the title 'Registo' in bold black text at the top. Below the title are five white input fields stacked vertically: 'Username', 'Primeiro Nome' (First Name), 'Apelido' (Last Name), 'Email', and 'Password'. At the bottom of the form is a blue button with the text 'Registrar' in white.

- É importante notificar que o acesso desta página está dependente da autenticação do utilizador.



- Nesta página o usuário tem também acesso às previsões dos últimos jogos guardados

VISUALIZE AS SUAS ODDS MAIS RECENTES

STATS->ODDS					ODDS-> ODDS				
Data	Jornada	Home Team	Away Team	APOSTA ESCOLHIDA	Data	Jornada	Home Team	Away Team	APOSTA ESCOLHIDA
23.09.2018 16:00	5	torreda	moreirense	X	14.09.2018 19:30	3	dortmund	frankfurt	X
23.09.2018 16:00	5	portimonense	cdmora-palmascas	1	16.09.2018 17:00	3	freiburg	stuttgart	X
23.09.2018 16:00	5	benfica	atros	2	16.09.2018 14:30	3	breman	munster	X
23.09.2018 16:00	5	portimonense	cdmora-palmascas	X	15.09.2018 17:30	3	le-mans	schalke	X
23.09.2018 16:00	5	portimonense	cdmora-palmascas	3.159888977671802	15.09.2018 14:30	3	monchengladbach	schalke	X
23.09.2018 16:00	5	torreda	moreirense	1	15.09.2018 14:30	3	th-lensse	hammer	X
14.09.2018 20:00	4	lousica	vallecas	2	15.09.2018 14:30	3	wolfsburg	hertha	X
				16.75135919173116	15.09.2018	-	-	-	X

○ Modelos - “Stats”

- Neste modelo é possível encontrar as *odds* para cada acontecimento (Vitória da equipa da casa, empate, vitória da equipa visitante) para os futuros eventos desportivos, baseadas nas estatísticas das equipas.

Liga NOS	La Liga	Serie A	Premier League	Bundesliga
----------	---------	---------	----------------	------------

Data	Jornada	Home Team	Away Team	1	X	2	Guardar	Estado
23/09/2018 16:00	5	tondela	moreirense	2.207249014115214	3.4972026023184792	3.831351134453249	Submit	GUARDADO

Data	Jornada	Home Team	Away Team	1	X	2	Guardar	Estado
23/09/2018 16:00	5	portimonense	vitoria-guimaraes	1.5385689986465805	3.159888977671802	29.780886601916777	Submit	GUARDADO

- É possível ainda selecionar um acontecimento e guardar essa *odd* onde poderá ser visualizada na página “modelos de previsão desportiva”

○ Modelos - “Odds”

- Neste modelo é possível encontrar as *odds* para cada acontecimento (Vitória da equipa da casa, empate, vitória da equipa visitante) para os futuros eventos desportivos, baseadas nas *odds* anteriores disponíveis no mercado.

Liga NOS	La Liga	Serie A	Premier League	Bundesliga
----------	---------	---------	----------------	------------

Data	Jornada	Home Team	Away Team	1	X	2	Guardar	Estado
15/09/2018 14:00	4	inter	parma	1.6463163671285497	5.330964015992892	4.878048780487805	Submit	

Data	Jornada	Home Team	Away Team	1	X	2	Guardar	Estado
15/09/2018 17:00	4	napoli	fiorentina	1.9821605550049555	30.3030303030303	2.1621621621621623	Submit	

- É possível ainda selecionar um acontecimento e guardar essa *odd* onde poderá ser visualizada na página “modelos de previsão desportiva”.
- Modelos- “Onze inicial”
 - No modelo do onze inicial o utilizador tem a possibilidade de inserir o onze que vai entrar em campo.



UAL BETS

[Calendário](#)
[Resultados](#)
[Noticias](#)
[Classificação e equipas](#)
[Modelos de previsão desportiva](#)

Liga NOS

La Liga

Serie A

Premier League

HOME TEAM

Passo 1 : Verificação

Submit

Registado

Jogador1

Varela B.

Jogador2

Jardel

Jogador3

Dias R.

Jogador4

Grimaldo A.

Jogador5

Pizzi

Jogador6

Fejsa L.

Jogador7

Zivkovic A.

Jogador1

Cervi F.

Jogador9

Jimenez R.

Jogador1

Silva R.

Jogador11

Almeida A.



AWAY TEAM

Jogador1

Casillas I.

Jogador2

Marcano I.

Jogador3

Felipe

Jogador4

Pereira R.

Jogador5

Brahimi Y.

Jogador6

Herrera H.

Jogador7

Sergio Oliveira

Jogador8

Otavio

Jogador9

Marega M.

Jogador10

Soares T.

Jogador11

Telles A.

- Uma vez que os dois onzes estejam submetidos o modelo vai calcular as médias de vários acontecimentos de um jogo bem como a probabilidade de cada equipa ganhar ou de haver um empate, com base nas estatísticas individuais de cada jogador e o seu impacto na equipa em jogos anteriores.


UAL BETS

[Calendário](#)
[Resultados](#)
[Notícias](#)
[Classificação e equipas](#)
[Modelos de previsão desportiva](#)
[Login/Registo](#)

Passo 2 : Medias e Probabilidade

test

Tentativas de golo [C]	Tentativas de golo [V]	Remates a baliza[C]	Remates a baliza[V]	Remates bloqueados[C]	Remates bloqueados[V]	Foras de jogo [C]	Foras de jogo[V]
7.999916749916751	7.930652680652681	1.5824175824175823	2.994755244755245	1.9405594405594406	0.49358974358974356	1.2272727272727273	2.2727272727272725

Faltas[C]	Faltas[V]	Cartões amarelos[C]	Cartões amarelos[V]	Cartões vermelhos[C]	Cartões vermelhos[V]	Golos de casa	Golos de fora
8.12920412920413	9.604312354312356	2.9875124875124874	2.5952797202797204	0.09090909090909093	0	0.14285714285714285	0.9166666666666667

1 - [5.6000000000000005] % || x - [27.0] % || 2 - [67.4] %

Manual do programador

Import

Descrição	Para que seja possível utilizar diversas funções e métodos ao longo do código, é necessário realizar os devidos <i>imports</i> das bibliotecas onde estão localizados as funções e métodos que queremos utilizar.
------------------	---

[9]

Webdriver.Chrome

Descrição	ChromeDriver é suportado pelo projeto "Chromium". O WebDriver funciona com o Chrome através do binário do chromedriver. É preciso ter o chromedriver e uma versão do chrome instalada.
Exemplo	<pre>from selenium import webdriver from selenium.common.exceptions import TimeoutException from selenium.webdriver.support.ui import WebDriverWait # available since 2.4.0 from selenium.webdriver.support import expected_conditions as EC # available since 2.26.0 driver = webdriver.Chrome() # Vai para a homepage do google driver.get("http://www.google.com") print driver.title # Procurar um elemento cujo nome do atributo seja "q" inputElement = driver.find_element_by_name("q") # Escrever na barra de pesquisa inputElement.send_keys("cheese!") # Submeter inputElement.submit() try: WebDriverWait(driver, 10).until(EC.title_contains("cheese!")) # Deverá ser possível ver "cheese! - Google Search"</pre>

	<pre>print driver.title finally: driver.quit()</pre>
--	---

[10]

driver.implicitly_wait

Descrição	Um explicit wait vai ser usado para esperar por uma determinada condição, antes de prosseguir no código. Existem alguns métodos disponíveis que servem para parar o código apenas por um determinado tempo. O WebDriverWait em conjunto com o ExpectedCondition é uma das maneiras de conseguir esta paragem no código.
Exemplo	<pre>driver.implicitly_wait(5)</pre> Indica para esperar 5 segundos antes de prosseguir.

[11]

driver.get

Descrição	O método driver.get vai realizar uma pesquisa na página fornecida pelo url. O WebDriver vai esperar ate que a página esteja carregada completamente antes de devolver o controlo ao script. É importante ter em conta que se uma página usar demasiado código AJAX o WebDriver pode não conseguir perceber quando a página está totalmente carregada.
Exemplo	<pre>jogos = [] for k in PTTEAMS: driver.get(k)</pre> Nesta porção do código, o driver.get vai buscar a informação do que estiver associado a variável “k”.

[12]

driver.close

Descrição	Fecha a janela que estiver aberta.
------------------	------------------------------------

[13]

list(set)

Descrição	Um Set é um conjunto de itens não ordenados. Cada elemento é único (não existe duplicados) e não pode ser alterado. Contudo o Set em si pode ser alterado, por exemplo adicionar ou remover itens. Os Sets podem ser usados para realizar diversas operações matemáticas como união, intersecção entre outras.
Exemplo	Neste caso o list(set) vai transformar um array numa lista, e eliminar todos os itens duplicados.

[14]

predict_proba

Descrição	O output esperado vai ser ordenado pela ordem das classes. Para um problema com multi_class, se a função multi_class for multinomial, a função softmax vai ser usada para prever a probabilidade de cada class. Caso contrário usa uma abordagem de one-vs-rest.
Parâmetros	X : Array , forma = [n_samples, n_features] T : Array , forma = [n_samples, n_classes]
Returns	Devolve a probabilidade de uma sample para cada class no modelo

[15]

driver.find_elements_by_tag_name

Descrição	Esta função deve ser usada, quando se sabe o nome do atributo de um elemento. Desta forma, o primeiro elemento cujo atributo corresponde ao nome pelo qual foi feita a pesquisa será devolvido. Se nenhum elemento tiver uma correspondência, a exceção NoSuchElementException vai aparecer.
------------------	---

[16]

driver.find_element_by_xpath

Descrição	XPath é uma linguagem usada para localizar nodes num documento XML. Uma vez que o HTML pode ser uma implementação do XML, os utilizadores do Selenium pode usar isto como uma vantagem, de maneira a localizar mais facilmente elementos nas <i>web applications</i>. O XPath vai mais além do que uma simples maneira de localizar, por id ou nome, um determinado elemento, pois abre uma janela de possibilidades que permite ao utilizador encontrar vários detalhes numa página, como por exemplo uma checkbox.
------------------	---

[17]

.append(elem)

Descrição	Adiciona um único elemento a uma lista.
------------------	---

[18]

get_attribute

Descrição	Este método vai buscar o atributo ou propriedade de um elemento. Primeiro o método vai tentar devolver um valor de uma propriedade com um dado nome, se uma propriedade com esse nome não existir, o método devolve o valor do atributo com o mesmo nome. Caso não exista atributo também, então nada é devolvido.
------------------	---

[19]

.append(elem) e get_attribute

Exemplo	<pre>jogos.append("https://www.meusresultados.com/jogo/"+i.get_attribute('id')[4:12:1]+"/#estatisticas-de-jogo;0")</pre> Nesta linha, o nosso elemento é o link do jogo, dentro desta string existe um elemento <code>get.attribute</code> que vai identificar o componente <code>id</code>.
----------------	---

DataFrame()

Descrição	Tabela estruturada com dois eixos (colunas e linhas) que pode ser alterada. Podem ser realizadas operações aritméticas tanto em colunas como em linhas.									
Parâmetros	data : numpy ndarray (structured or homogeneous), dict, or DataFrame “Dict” pode conter Series, arrays, “constants”, ou “list-like objects” index : Index ou array Index para usar na frame final. Vai ter como default “RangeIndex” se nenhuma informacao for indexada. columns : Index ou array Colunas a usar na frame final. Vai ter como default “RangeIndex” (0, 1, 2, ..., n) se nenhuma coluna for fornecida previamente. dtype : dtype, default None Apenas um datatype é permitido copy : boolean, default False Copy data dos inputs. Apenas afecta o Dataframe									
Exemplo	<pre>>>> d = {'col1': [1, 2], 'col2': [3, 4]} >>> df = pd.DataFrame(data=d) >>> df</pre> <table><tr><th></th><th>col1</th><th>col2</th></tr><tr><td>0</td><td>1</td><td>3</td></tr><tr><td>1</td><td>2</td><td>4</td></tr></table>		col1	col2	0	1	3	1	2	4
	col1	col2								
0	1	3								
1	2	4								

[20]

text.split

Descrição	A determinada altura, poderá ser necessário separar uma <i>string</i> em pedaços mais pequenos, ou seja, várias <i>strings</i>. Isto é o contrario de concatenação, que por sua vez vai juntar ou combinar <i>string</i> numa só.
------------------	---

[21]

pd.concat

Descrição	Permite concatenar vários objetos de um eixo com outros eixos. Pode ser ainda possível atribuir um nível de hierarquia a um dos eixos, isto ajuda quando os indexes são os mesmos ou estão sobrepostos.
Parâmetros	objs : <i>Uma sequência, DataFrame ou Pannel de objects.</i> axis : <i>{0/'index', 1/'columns'}, default 0</i> Eixo onde se vai realizar a concatenação join : <i>{'inner', 'outer'}, default 'outer'</i> Como lidar com índices noutra eixo join_axes : <i>list of Index objects</i> Índices específicos a serem usados para os outros eixos n - 1 em vez de executar a lógica de conjunto interno / externo ignore_index : <i>boolean, default False</i> Se for True, não usa os valores de índice ao longo do eixo de concatenação. O eixo resultante será identificado como 0,..., n - 1. Isso será útil se estivermos a concatenar objetos em que o eixo de concatenação não possui informações de indexação significativas. keys : <i>sequence, default None</i> Se vários levels foram passados, deve conter arrays. Construir um índice hierárquico usando as keys passadas como o nível mais externo levels : <i>list of sequences, default None</i> Levels específicos (valores exclusivos) a serem usados na construção de um MultiIndex. Caso contrário, eles serão inferidos das key names : <i>list, default None</i> Nomes para os Levels no índice hierárquico resultante verify_integrity : <i>boolean, default False</i> Verifica se o novo eixo concatenado contém duplicados. Isso pode ser muito dispendioso em relação à concatenação real de dados sort : <i>boolean, default None</i> Alinha o eixo de não concatenação se ele ainda não estiver alinhado quando a junção for "outer". copy : <i>boolean, default True</i> Se for False, não copia os dados desnecessariamente.
Returns	concatenated : <i>object, type of objs</i> Ao concatenar todas as séries ao longo do índice (eixo = 0), uma série é devolvida. Quando "objs" contém pelo menos um DataFrame, um

	DataFrame é devolvido. Ao concatenar ao longo das colunas (axis = 1), um DataFrame é retornado.
--	---

[22]

str.contains

Descrição	Teste se padrão ou regex está contido numa <i>string</i> de uma série ou índice
Parâmetros	pat : str Sequência de caracteres ou expressão regular. case : bool, default True Se for True, diferencia maiúsculas de minúsculas. flags : int, default 0 (no flags) Flags para passar para o módulo re, por exemplo: re.IGNORECASE. na : default NaN Qualquer valor ausente vai ser preenchido. regex : bool, default True Se True, assume que o pat é uma expressão regular. Se False, trata o pad como uma literal string.
Returns	Série ou Índice de valores booleanos Uma série ou índice de valores booleanos indicando se o padrão fornecido está contido na string de cada elemento da série ou índice.

[23]

numpy.select

Descrição	Devolve um <i>array</i> formado com os elementos na <i>choicelist</i> , dependendo de algumas condições
Parâmetros	condlist : <i>list of bool ndarrays</i> A lista de condições que determinam de qual <i>array</i> na <i>choicelist</i> os elementos de <i>output</i> serão retirados. Quando várias condições são satisfeitas, a primeira encontrada na <i>condlist</i> é usada. choicelist : <i>list of ndarrays</i> A lista de <i>arrays</i> dos quais os elementos de <i>output</i> são retirados. Tem que ser do mesmo tamanho que <i>condlist</i> default : <i>scalar, optional</i> O elemento inserido no output quando todas as condições são avaliadas como falsas.
Returns	output : <i>ndarray</i> O output na posição <i>m</i> é o <i>m</i> elemento da matriz na <i>choicelist</i> , onde o <i>m</i> elemento da matriz correspondente na <i>condlist</i> é True.

[24]

train_test_split

Descrição	Dividir <i>arrays</i> ou matrizes em subconjuntos aleatórios de <i>train</i> e <i>test</i> .
Parâmetros	arrays : seqência de indexes com a mesma length / shape[0] Inputs permitidos são: lists, arrays numpy, matrizes esparsas ou dataframes. test_size : float, int, None, optional Se for <i>float</i> , deve estar entre 0.0 e 1.0 e representa a proporção do <i>dataset</i> que será incluído no <i>test split</i> . Se for <i>Int</i> , representa o número absoluto de amostras de teste. Se for <i>None</i> , o valor é definido para complementar o <i>train size</i> . Por norma o valor está definido para 0.25.

	Train_size : float, int, or None, default None Se for <i>float</i>, deve estar entre 0.0 e 1.0 e representa a proporção do <i>dataset</i> que será incluído no <i>train split</i>. Se for <i>Int</i>, representa o número absoluto de amostras de teste. Se for <i>None</i>, o valor é definido para complementar o <i>test size</i>. random_state : int, RandomState instance or None, optional (default=None) Se <i>Int</i>, o <i>Random_State</i> vai ser a seed usada pelo gerador de números aleatório; Se o <i>Random_State</i> for instance, o <i>random_state</i> passa a ser o gerador de números; se for <i>None</i> o gerador de números vai ser o <i>RandomState</i> instance usada pelo <i>np.random</i>. shuffle : boolean, optional (default=True) Define se a os dados serão ou não baralhos antes do split acontecer. Se for <i>false</i> então o <i>Stratify</i> tem de ser <i>None</i>. stratify : array-like or None (default is None) Se não for <i>None</i>, os dados serão divididos de maneira estratificada, usando isto como class labels.
Returns	splitting : list, length=2 * len(arrays) Lista contendo os inputs do <i>train-test split</i>.

RandomForestClassifier

Descrição	O Random Forest utiliza um conjunto de <i>Decision Trees</i> para realizar o aprendizado dos dados, o que tende a apresentar um desempenho melhor que um modelo simples
Parâmetros	n_estimators : integer, optional (default=10) Numero de arvores que seram usadas. criterion : string, optional (default="gini") Funcao que mede a qualidade do split. Os critérios suportados são "gini" para a impureza Gini e "entropy". max_features : int, float, string or None, optional (default="auto") Numero de features a considerar para ter o melhor split possivel. • Se int, considerar max_features para cada split. • Se float, entao max_features passa a percentagem e int features passam a ser consideradas para cada split. • Se "auto", então max_features=sqrt(n_features). • Se "sqrt", então max_features=sqrt(n_features) (igual a "auto"). • Se "log2", então max_features=log2(n_features). • Se None, então max_features=n_features. max_depth : integer or None, optional (default=None) Comprimento maximo para a cada arvore. Se for None, então os nodes vai se expandir ate que as leaves sejam pure ou contenham menos amostras que o minimo estipulado.

min_samples_split : int, float, optional (default=2)

O numero minimo de numeros de amostras necessarias para dividir um internal node:

- Se int, considerar min_samples_split como o numero minimo.
- Se float, considerar min_samples_split como uma percentagem e $\text{ceil}(\text{min_samples_split} * n_samples)$ é o numero minimo de amostras por cada split.

min_samples_leaf : int, float, optional (default=1)

Numero minimo de amostras precisas para estarem num leaf node:

- Se int, considerar min_samples_leaf como o numero minimo.
- Se float, considerar min_samples_leaf is a percentage e $\text{ceil}(\text{min_samples_leaf} * n_samples)$ é o numero minimo de amostras por cada node.

min_weight_fraction_leaf : float, optional (default=0.)

A fração ponderada mínima da soma total de pesos (de todas as amostras de input) necessária para estar num leaf node. As amostras têm o mesmo peso quando o sample_weight não é fornecido.

max_leaf_nodes : int or None, optional (default=None)

Aumentar o tamanho das árvores com max_leaf_nodes do estilo "best-first". Os melhores nodes são definidos como redução relativa na impureza. Se None, então os numeros de leaf nodes sao ilimitados

min_impurity_split : float,

Threshold para a interrupção do crescimento das árvores. Um node será dividido se a sua impureza estiver acima do limite, caso contrário, será uma leaf

min_impurity_decrease : float, optional (default=0.)

Um node será dividido se essa divisão induzir uma diminuição da impureza maior ou igual a esse valor.

bootstrap : boolean, optional (default=True)

Define se as bootstrap samples são usadas na construção das arvores.

oob_score : bool (default=False)

Define se são usadas out-of-bag samples para calcular a precisao

n_jobs : integer, optional (default=1)

O numero de jobs que vao correr em paralelo para o fit e para o predict. Se 1 ent o numero de jobs vai ser o mesmo numero de cores.

random_state : int, RandomState instance or None, optional (default=None)

Se int, o random_state vai ser a seed usada pelo gerador aleatorio de numeros; Se RandomState instance, entao random_state vai ser o gerador aleatorio de numeros; Se None, o gerador aleatorio de numeros vai ser o RandomState instance usado pelo np.random.

verbose : int, optional (default=0)

Controla a verbosidade do processo de construção de árvores.

warm_start : bool, optional (default=False)

	Quando for True, reutiliza a solucao anterior para ajustar e adicionar mais “estimators” ao “ensemble”, caso contrario, vai fazer “fit” para uma nova “forest”. class_weight : dict, list of dicts, “balanced”. “balanced_subsample” ou None, “optional (default=None) Weights” associados a classes na forma de {class_label: weight}. Se não for dado qualquer “weight” todas as classes devem ter weight = 1. Para problemas com mais que um output, a “list of dicts” pode ser distribuida na mesma ordem que as colunas de y.
Atributos	estimators_ : list of DecisionTreeClassifier Conjunto ajustado de “sub-estimators”. classes_ : array of shape = [n_classes] or a list of such arrays n_classes_ : int or list n_features_ : int Numero de features quando “fit” é executado. n_outputs_ : int Numero de outputs quando “fit” é executado. feature_importances_ : array of shape = [n_features] Imporancia da feature (quanto maior, mais importante é a feature). oob_score_ : float

	Resultado do <i>training dataset</i> obtido usando uma estimativa “out-of-bag”.
--	---

[26]

Fit

Descrição	Construir uma “forest” com arvores a partir do “training set (X, y)”.
Parâmetros	X : array ou matriz esparsa = [n_samples, n_features] y : array = [n_samples] or [n_samples, n_outputs] sample_weight : array, com a forma = [n_samples] ou None Sample weights. Se None, então as samples tem o mesmo peso. Splits que iriam criar “child nodes” com uma “net” de zero ou peso negativo serão ignorados quando for efectuado uma pesquisa por splits em cada “node”. No caso de classificação, os Splits são também ignorados se o seu resultado final fosse uma classe apenas com peso negativo em cada “child node”.
Returns	self : object Devolve self.

[27]

Score

Descrição	Calcula a precisão nos dados de teste que foram disponibilizados.
Syntax:	Score (X, y, sample_weight=None)
Parâmetros	X : array, forma = (n_samples, n_features) Test samples. y : array, forma = (n_samples) ou (n_samples, n_outputs) Labels verdadeiros para X.

	sample_weight : array, forma = [n_samples], optional Pesos das amostras.
Returns	score : float Devolve o valor da precisão do classificador em teste.

[28]

Groupby

Descrição	Agrupar dados considerando a característica escolhida.
Syntax:	groupby (by=None, axis=0, level=None, as_index=True, sort=True, group_keys=True, squeeze=False, observed=False)
Parâmetros	by (mapping, função, label ou lista de labels) - Usado para determinar os dados a agrupar. axis (0 ou 1) - index = 0 ; colunas = 1 level (opcional, int, nome do level) - Se a label for multiIndex (hierárquica), agrupa por um nível específico. as_index : <i>boolean, default True</i> Para output agregado, devolve o nome do "group" usado como index. sort : <i>boolean, default True</i> ordena a chave do grupo. group_keys : <i>boolean, default True</i>

	Ao aplicar, adiciona "group keys" ao index para serem identificadas. squeeze : <i>boolean, default False</i> Reduz a dimensionalidade do tipo de variável que é devolvido, se não, devolve um tipo de variável consistente. observed : <i>boolean, default False</i> Isto só é aplicado se algum dos agrupantes for categórico.
Returns	Devolve um <i>dataframe</i> agrupado pelo item escolhido.

[29]

GaussianNB

Descrição	O algoritmo “Naive Bayes” é um classificador probabilístico baseado no “Teorema de Bayes”. O teorema de Bayes incide-se na probabilidade condicional. Isto é qual a probabilidade de o evento A ocorrer, dado o evento B
Parâmetros	Priors (array) - Probabilidades anteriores das classes.
Atributos	class_prior (array) - probabilidade de cada classe class_count (array) - número de amostras de treino observadas em cada classe. theta (array) - Significado de cada característica por classe sigma (array) - Variância de cada característica por classe

[30]

fillna

Descrição	Preenche os espaços em "NA" ou "NaN" pelos valores especificado no método.
Syntax	<code>fillna(value=None, method=None, axis=None, inplace=False, limit=None, downcast=None)</code>
Parâmetros	value (scalar, dict, dataframe) - O "quadro" onde vão ser corrigidos os espaços em branco. method ('backfill', 'bfill', 'pad', 'ffill', None) - Método a usar para preencher. axis (0 ou 1) - index = 0 ; colunas = 1 inplace (boolean) - if "True", preenche no espaço. limit (opcional, int) - Número para definir o máximo de NaN's para substituir. downcast (opcional, dict) - Dependendo do tipo escolhido, se for encontrado, será eliminado também.
Return	Devolve o dataframe já sem os NaN's.
Exepmlo	<code>df99.fillna(0, inplace = True)</code> - Neste caso, substituímos os Nan's por 0, no dataframe "df99".

[31]

drop

Descrição	Remove linhas ou colunas especificando o nome do index da linha ou o nome da coluna.
Syntax	<code>drop(labels=None, axis=0, index=None, columns=None, level=None, inplace=False, errors='raise')</code>
Parâmetros	labels (linha ou lista) - Index ou coluna para remover axis (0 ou 1), para dropar labels do index será 0, para dropar colunas será 1 index, columns (linha isolada ou lista) - uma alternativa para especificar o "axis" level (opcional) - Para multi-index, nível de cada label para ser removido inplace (bool) - se True, não devolve nada errors (ignore, raise) - Se "ignore", ignora o erro e apenas as linhas existentes serão removidas.
Return	Apenas remove as linhas/colunas escolhidas.
Exemplo	<code>df.drop(['A','FS','SP','TP'],1)</code> - Neste exemplo estamos a remover as colunas mencionadas da dataframe "df", são colunas pois a axis = 1. (seriam linhas se axis = 0)

loc

Descrição	Acede a um grupo específico de linhas e colunas por tabela.
Syntax	loc[]
Return	Devolve as labels, em forma de tabela, que foram pedidas.
Exemplo	df100.loc['JUGU','Tentativas de golo[C]'] - Vai localizar, na dataframe df100, apenas o index "JUGU" e a coluna Tentativas de golo [C].

[33]

Sum

Descrição	A função sum() adiciona os itens numa iterável.
Syntax	sum(iterable, start)
Parâmetros	Iterable - iterável (ex: lista) cuja soma dos itens é para ser encontrada. Normalmente, os itens da iterável são números. Start (opcional) - Este valor é adicionado à soma dos itens da iterável. O valor default é 0
Return	A função soma devolve a soma dos itens dados pela "iterable"
Exemplo	dfcasa.sum() - somar todos os dados do dataframe "dfcasa"

[34]

str.replace

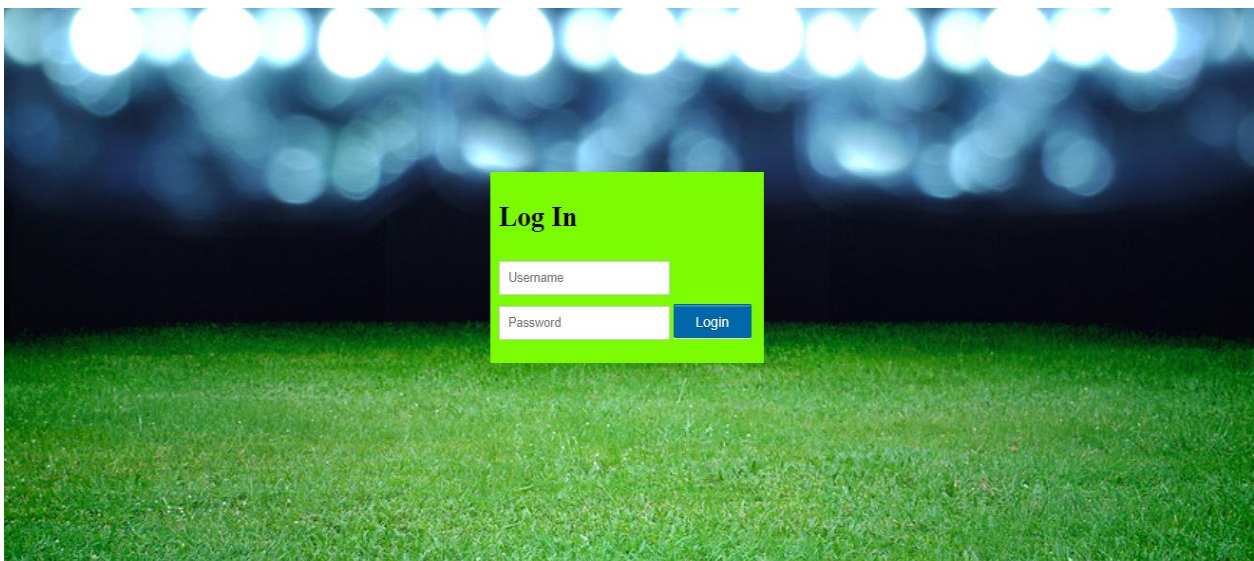
Descrição	A função sum() adiciona os itens numa iterável.
Syntax	sum(iterable, start)
Parâmetros	Iterable - iterável (ex: lista) cuja soma dos itens é para ser encontrada. Normalmente, os itens da iterável são números. Start (opcional) - Este valor é adicionado à soma dos itens da iterável. O valor default é 0
Return	A função soma devolve a soma dos itens dados pela "iterable"
Exemplo	dfcasa.sum() - somar todos os dados do dataframe "dfcasa"

[35]

API

Neste projeto foi criada uma espécie de REST API com o objetivo de efetuar operações CRUD(Create, Read, Update, Delete). Esta api não funciona como uma REST API no sentido que não é possível fazer “Http Requests”, isto é, obter correspondências dos dados apartir de um url na barra de navegação do browser.

Esta Api necessita de uma autenticação especial, para isso o administrador deve colocar um username e uma password (nesta situação, username: “admin”, password: “admin”)



De seguida o administrador é remetido para a pagina das operações CRUD onde pode realizar diversas operações sobre as tabelas da base de dados da “Web application”

O administrador pode realizar operações de inserir novos dados na BD, editar, remover e ver os mesmos dados.

OPERAÇÕES CRUD

Id	username	mail	fname	lname	Delete
28	user	trerrr@sdaasedf	antonio	rrr	
27	ola	asd	asd	asd	
26	asdsdas	asd@asdqwe	asdwqe	asd	
5	dfsdfqwe	asd@asdqwe	xcv	sdcvx	

Por fim, o administrador encontra uma barra lateral onde estão o nome das tabelas e onde poderá editar as mesmas.



Users

A bola

Youtube

Sapo

stats_odds_pt

stats_odds_ale

stats_odds_esp

stats_odds_ita

stats_odds_eng

odds_odds_pt

odds_odds_ale

odds_odds_esp

Conclusão

No início deste projeto tínhamos como objetivo mostrar o impacto que o *machine learning* pode ter no mercado das apostas desportivas.

Deparámo-nos com algumas dificuldades mas consideramos que o objetivo foi alcançado.

Elaborámos dois modelos já conhecidos no mundo do *betting* e do *machine learning*, e com os nossos conhecimentos e experiência prévia em apostas, criámos um modelo inovador.

Percebemos rapidamente que este modelo facilmente poderia ser apreciado por muitos apostadores, devido à maneira como funciona e a vantagem que traz ao apostador.

Para que estes modelos funcionassem da melhor forma, tivemos que testar varios classificadores, e acabamos por concluir que o classificador naive bayes e o random forest eram os melhores para o nosso projecto

Foi então que decidimos criar uma *web application*, não só para ser mais fácil de mostrar os nossos resultados, mas também com uma visão para o futuro de implementar esta ideia e ajudar todos os apostadores no mundo.

Referências

- [1] Using Machine Learning to Improve Streaming Quality at Netflix. Retirado em 14 de setembro, 2018 de <https://medium.com/netflix-techblog/using-machine-learning-to-improve-streaming-quality-at-netflix-9651263ef09f>
- [2] Retirado em 14 de setembro, 2018 de <https://selenium-python.readthedocs.io/getting-started.html#getting-started>
- [3] Algoritmo Naive Bayes. Retirado em 14 de setembro, 2018 de <http://artificiencia.com/aprenda/algoritmo-naive-bayes/>
- [4] Retirado em 14 de setembro, 2018 de <https://medium.com/machina-sapiens/o-algoritmo-da-floresta-aleat%C3%B3ria-3545f6babdf8>
- [5] BeautifulSoup Documentation. Retirado em 14 de setembro, 2018 de <https://www.crummy.com/software/BeautifulSoup/bs4/doc/>
- [6] ChromeDriver - WebDriver for Chrome. Retirado em 14 de setembro, 2018 de <http://chromedriver.chromium.org/>
- [7] Retirado em 14 de setembro, 2018 de <https://expresso.sapo.pt/economia/aprovada-lei-do-jogo-online-portugal-passara-a-tributar-receitas-fiscais-que-ate-agora-lhe-escapavam=f912575#gs.oSBHqqg>
- [8] Match Prediction Bot. Retirado em 14 de setembro, 2018 de https://www.reddit.com/r/MachineLearning/comments/43z5yk/match_prediction_bot_xpost_from_rsoccerbetting/
- [9] The import system. Retirado em 14 de setembro, 2018 de <https://docs.python.org/3/reference/import.html>
- [10] Retirado em 14 de setembro, 2018 de https://www.seleniumhq.org/docs/03_webdriver.jsp
- [11] Waits. Retirado em 14 de setembro, 2018 de <https://selenium-python.readthedocs.io/waits.html>

- [12] Retirado em 14 de setembro, 2018 de <https://selenium-python.readthedocs.io/getting-started.html>
- [13] WebDriver API. Retirado em 14 de setembro, 2018 de <https://selenium-python.readthedocs.io/api.html>
- [14] Python Sets. Retirado em 14 de setembro, 2018 de <https://www.programiz.com/python-programming/set>
- [15] LogisticRegression. Retirado em 14 de setembro, 2018 de http://scikit-learn.org/stable/modules/generated/sklearn.linear_model.LogisticRegression.html
- [16] Locating Elements. Retirado em 14 de setembro, 2018 de <https://selenium-python.readthedocs.io/locating-elements.html>
- [17] Retirado em 14 de setembro, 2018 de <https://selenium-python.readthedocs.io/locating-elements.html>
- [18] Python Lists. Retirado em 14 de setembro, 2018 de <https://developers.google.com/edu/python/lists>
- [19] WebDriver API. Retirado em 14 de setembro, 2018 de <https://selenium-python.readthedocs.io/api.html>
- [20] pandas.DataFrame. Retirado em 14 de setembro, 2018 de <https://pandas.pydata.org/pandas-docs/stable/generated/pandas.DataFrame.html>
- [21] How to use Split in Python. Retirado em 14 de setembro, 2018 de <https://www.pythonforbeginners.com/dictionary/python-split>
- [22] pandas.concat. Retirado em 14 de setembro, 2018 de <https://pandas.pydata.org/pandas-docs/stable/generated/pandas.concat.html>
- [23] pandas.Series.str.contains. Retirado em 14 de setembro, 2018 de <https://pandas.pydata.org/pandas-docs/stable/generated/pandas.Series.str.contains.html>
- [24] numpy.select. Retirado em 14 de setembro, 2018 de <https://docs.scipy.org/doc/numpy/reference/generated/numpy.select.html>
- [25] train_test_split. Retirado em 14 de setembro, 2018 de http://scikit-learn.org/stable/modules/generated/sklearn.model_selection.train_test_split.html

[26] RandomForestClassifier. Retirado em 14 de setembro, 2018 de <http://scikit-learn.org/stable/modules/generated/sklearn.ensemble.RandomForestClassifier.html>

[27] RandomForestClassifier. Retirado em 14 de setembro, 2018 de <http://scikit-learn.org/stable/modules/generated/sklearn.ensemble.RandomForestClassifier.html>

[28] RandomForestClassifier. Retirado em 14 de setembro, 2018 de <http://scikit-learn.org/stable/modules/generated/sklearn.ensemble.RandomForestClassifier.html>

[29] pandas.DataFrame.groupby. Retirado em 14 de setembro, 2018 de <https://pandas.pydata.org/pandas-docs/stable/generated/pandas.DataFrame.groupby.html>

[30] GaussianNB. Retirado em 14 de setembro, 2018 de http://scikit-learn.org/stable/modules/generated/sklearn.naive_bayes.GaussianNB.html

[31] pandas.DataFrame.fillna. Retirado em 14 de setembro, 2018 de <https://pandas.pydata.org/pandas-docs/stable/generated/pandas.DataFrame.fillna.html>

[32] pandas.DataFrame.drop. Retirado em 14 de setembro, 2018 de <https://pandas.pydata.org/pandas-docs/stable/generated/pandas.DataFrame.drop.html>

[33] pandas.DataFrame.loc. Retirado em 14 de setembro, 2018 de <https://pandas.pydata.org/pandas-docs/stable/generated/pandas.DataFrame.loc.html>

[34] Python sum(). Retirado em 14 de setembro, 2018 de <https://www.programiz.com/python-programming/methods/built-in/sum>

[35] Python String replace() Method. Retirado em 14 de setembro, 2018 de https://www.tutorialspoint.com/python/string_replace.htm