Uma imagem com texto

Descrição gerada automaticamente

Introduction to Machine Learning — 2022/2023

Projeto final

André Filipe Frade Guerra

Lucas Barrigó

Janeiro/2023

Conteúdo

[Introdução 3](#_Toc117354189)

[Classes e métodos 3](#_Toc117354190)

[Exercício 1 5](#_Toc117354191)

[Resultados exercício 1 6](#_Toc117354192)

[Exercício 2 7](#_Toc117354193)

[Resultados exercício 2 8](#_Toc117354194)

[Exercício 3 9](#_Toc117354195)

[Resultados exercício 3 11](#_Toc117354196)

[Exercício 4 12](#_Toc117354197)

[Resultados exercício 4 14](#_Toc117354198)

[Conclusão 16](#_Toc117354199)

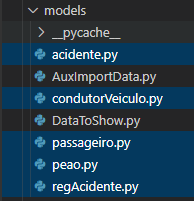
## Introdução

Neste relatório irá ser explicado resumidamente as classes criadas, funcionalidade dos métodos criados e por fim a execução e discussão dos resultados dos exercícios.

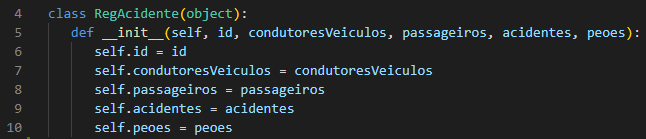
Este trabalho tem como objetivo analisar os dados sobre acidentes rodoviários em Portugal, desde o ano 2010 a 2019. Os dados foram fornecidos em formato Excel, 1 por cada ano, dos quais foram retirados dados de forma aleatória (3000 acidentes por cada ano) guardados em formato JSON que depois foram filtrados e normalizados para formato CSV, formato utilizado pelos algoritmos dos exercícios.

## Classes e métodos

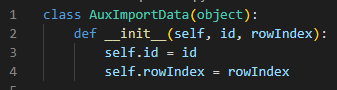
Começando pela extração de dados, foram criados objetos para cada folha dentro dos exíeis, que contém os mesmos atributos presentes nas mesmas:



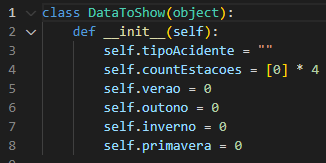
O objeto “RegAcidente”, presente na pasta dos models, representa um acidente que contem o id, condutor do veiculo (condutoresVeiculos), passageiros e peões, caso o acidente tenha esses atributos nas diferentes folhas do respetivo Excel:



O modelo “AuxImportData”, é um auxiliar para a extração dos dados, que irá ser explicada mais a frente:

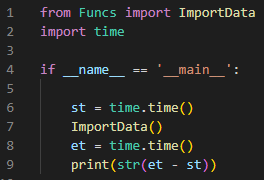


O modelo “DataToShow”, é um auxiliar para o exercício 3:



## Exercício 1

O exercício 1, presente no ficheiro exer1.py, consiste numa chamada do método que exporta aleatoriamente dados de todos os ficheiros para um formato de JSON, e mostra o tempo demorado da tarefa:

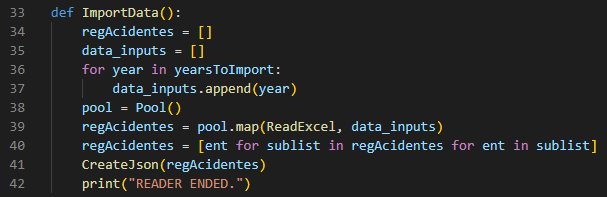


A exportação dos dados utiliza multiprocessamento para efetuar a extração mais rapidamente, tendo em conta que temos 10 exíeis para exportar, são criados 10 processos que correm em núcleos separados do CPU, caso o CPU tenho menos de 10 núcleos, o próximo Excel apenas começa a exportar quando um núcleo fica disponível. A nossa extração contem 3000 acidentes de cada Excel, um total de 30.000 acidentes, escolhidos aleatoriamente, e tendo em conta o processador usado (i7-12700H), demora cerca de 1 hora a extração.

O método que inicia a exportação de dados “ImportData” está no ficheiro “Funcs.py”, que também contem todos os outros métodos necessários para a exportação. Neste mesmo ficheiro começamos por declarar 2 variaveis que contem o numero de acidentes e os anos a serem exportados:



O método “ImportData” começa por percorrer o array com os anos a serem exportados e cria o processo que realiza a exportação, os processos partilham uma variável global “regAcidentes”, que contem uma lista de objetos (“RegAcidente”) extraídos. Por cada processo é chamado o método “ReadExcel” que lê e extrai os dados:



O método “ReadExcel” começa por ler o ficheiro do ano que recebe, e as suas folhas. De seguida cria arrays com o objeto “AuxImportData” que contem o id e o index da linha na folha, este array com o id e a posição da linha permite que a extração seja feita mais rapidamente (percorrer uma array de objetos pequenos ao invés das folhas no Excel):

Uma imagem com texto

Descrição gerada automaticamente

Uma imagem com texto

Descrição gerada automaticamente

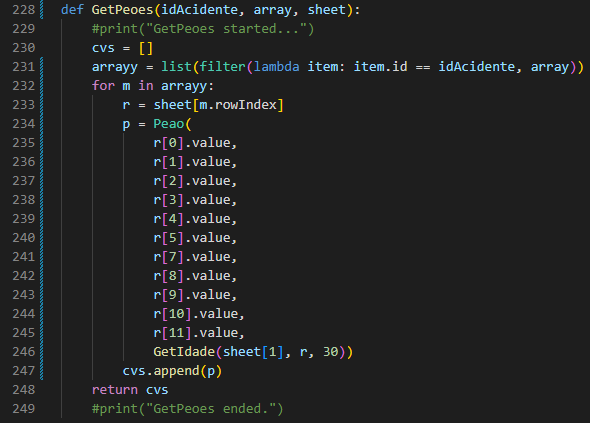
Após os arrays estarem construídos, percorremos o array 3 (ou a folha do Excel 3 “30 Dias \_ Acidentes”), até se atingir o número de dados pretendidos ou o array 3 chegar ao fim.

Esta extração pega numa posição aleatória do array, que escolhe o id do acidente a ser exportado, esse id é usado depois nos outros arrays, que depois de filtrados permite saber a posição do dado na folha de excel, essa exportação depois é feita nos métodos “GetCondutoresVeiculo”, “GetPassageiros”, “GetAcidentes” e “GetPeoes”, que cria o objeto “RegAcidente”. A cada acidente extraído, é removido dos arrays o acidente pelo id escolhido, impedindo a repetição do mesmo:

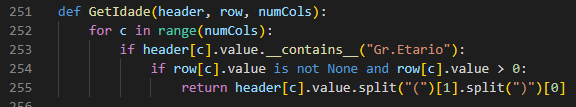
Uma imagem com texto

Descrição gerada automaticamente

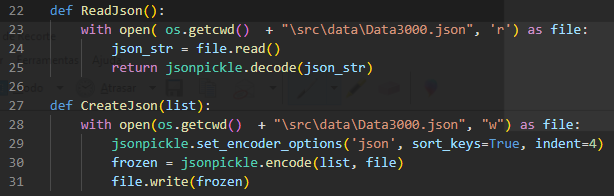
Os 4 métodos que extraem o objeto de cada folha do Excel são todos semelhantes. Explicando o “GetPeoes”, este recebe o id do acidente, o array da folha de Excel que contem os id’s e o index da posição dado presente na folha peões “30 Dias\_Peões”. O array depois de filtrado pelo id do acidente, é percorrido e extraída a informação da folha do Excel tendo em conta a posição do mesmo dado:



O método “GetIdade” retorna a idade de forma mais prática de se trabalhar mais a frente nos exercícios, retornando apenas a idade a que pertence o dado, por exemplo “40-44”:



Os métodos “ReadJson” e “CreateJson”, permitem ler e escrever em formato JSON os dados extraídos, estes dados estão presentes da pasta “data”:



## Exercício 2

Presente no ficheiro exer2.py, fazemos um tratamento dos dados exportados substituindo tudo por números e guardando tudo num ficheiro CSV, de forma a poder trabalhar com as bibliotecas:

Uma imagem com texto

Descrição gerada automaticamente

Este tratamento torna-se bastante eficiente na escolha dos dados, pois caso não seja possível converter para inteiro este é ignorado. A escolha dos dados a serem usados também é bastante simples e permite escolher apenas os que iram ser usados ou relevantes para o algoritmo:

Uma imagem com texto

Descrição gerada automaticamente

## Exercício 3

No exercício 3, não é utilizado o CSV tratado anteriormente, mas sim o JSON criado na extração. Neste exercício, para cada tipo de acidente, contamos a estação em que ocorreu:

Uma imagem com texto

Descrição gerada automaticamente

E depois alteramos o formato dos dados contabilizados para o input do gráfico, o valor do tipo de acidente “não identificada” foi filtrado, pois tem um valor muito alto e irrelevante no exercício:

Uma imagem com texto

Descrição gerada automaticamente

Print dos graficos

## Exercício 4

O exercício 4 é feito juntamente com o exercício 2.

## Exercício 5

Após ser feito o tratamento dos dados no exercício 2 é calculada a correlação e mostrada em um HeatMap.

## Conclusão

Inicialmente o código foi feito para poder trabalhar com várias cores, assim sendo possível em cada posição da pattern poderem existir até 8 cores, os exercícios anteriores foram executados sempre apenas com 2 cores possíveis.

Para podermos observar mais facilmente a relação entre a quantidade de cores e a velocidade, foi criado um rating que vai ser calculado da seguinte forma: