Exercício 1

Neste exercício, você vai construir um modelo usando a técnica de mínimos quadrados para prever o preço de carros usados usando os dados do *Used Cars Dataset*. Este banco de dados é uma enorme lista com 957 MB de informações de veículos anunciados no site *Craigslist.org*, incluindo diversos tipos de veículos.

Ao trabalhar com dados reais, muitas vezes é necessário realizar algumas etapas de filtragem e limpeza para usar os dados para construir um modelo. No caso desse banco de dados, é necessário restringir os tipos de veículos que serão considerados, para obter melhores resultados com o modelo.

Como o foco principal do exercício é usar a técnica de mínimos quadrados para construir o modelo, você vai trabalhar com um conjunto de dados limpo, que é um pequeno subconjunto do banco de dados original. Dessa forma, antes de começar a trabalhar no código, certifique-se de ter baixado o arquivo CSV de dados limpos do repositório de materiais.

Para carregar um arquivo CSV e processar os dados, você pode usar a biblioteca Pandas. Após a instalação da biblioteca e o download do arquivo CSV, é possível carregar os dados com os seguintes comandos:

```
import pandas as pd
cars_data = pd.read_csv("vehicles_cleaned_train.csv")
```

Com esses comandos, você vai criar um *DataFrame* do Pandas chamado `cars_data` contendo os dados do arquivo CSV. A partir deste *DataFrame*, você pode gerar os *arrays* NumPy que serão usados para calcular os parâmtros do modelo de regressão linear.

Para ver algumas linhas do banco de dados, você pode usar o método `.head()`:

```
cars_data.head()
```

	price	year	condition	cylinders	fuel	odometer	transmission	size	type
0	10400	2011	excellent	4 cylinders	gas	81300	automatic	mid-size	sedan
1	6900	2007	excellent	6 cylinders	gas	79000	automatic	full-size	sedan
2	18900	2018	like new	4 cylinders	gas	5000	automatic	full-size	sedan
3	6000	2010	excellent	4 cylinders	gas	97600	automatic	mid-size	hatchback
4	19995	2013	good	6 cylinders	gas	95782	automatic	mid-size	sedan

Conforme você pode notar pela saída do comando acima, o banco de dados tem 9 colunas. A descrição dos dados é a seguinte:

Coluna	Descrição
`price`	O preço do carro. É o dado que você deseja obter com o modelo.
`year`	O ano do carro.
`condition`	Variável categórica que indica a condição do carro. Pode ter os valores `good`, `fair`, `excellent`, `like new`, `salvage`, ou `new`.
`cylinders`	Variável categórica que indica o número de cilindros do motor. Pode ter os valores `4 cylinders` ou `6 cylinders`.
`fuel`	Variável categórica que indica o combustível do carro. Pode ter os valores `gas` ou `diesel`.
`odometer`	Valor registrado no odômetro, em milhas.
`transmission`	Variável categórica que indica o tipo de transmissão. Pode ter os valores `automatic` ou `manual`.
`size`	Variável categórica que indica o tamanho do carro. Pode ter os valores `compact`, `mid-size`, `sub-compact` ou `full-size`.
`type`	Variável categórica que indica o tipo do carro. Pode ter os valores `sedan`, `coupe`, `wagon`, or `hatchback`.

Para usar esses dados para construir o modelo, você vai precisar representar os dados categóricos de forma numérica. Na maioria dos casos, cada variável categórica é transformada em um conjunto de variáveis *dummy*, que são variáveis que podem assumir o valor 0 ou 1. Como exemplo dessa transformação, considere a coluna `fuel`, que pode assumir os valores `gas` ou `diesel`. Nesse caso, a variável categórica pode ser transformada em uma variável *dummy* chamada `fuel_gas` que assume o valor 1, quando `fuel` for igual a `gas` e 0, quando `fuel` for igual a `diesel`. Vale notar que você vai precisar de apenas uma variável *dummy* para representar uma variável categórica que pode assumir dois valores diferentes. Da mesma forma, para uma variável categórica que pode assumir N valores distintos, você precisará de $N-1$ variáveis *dummy*, pois um dos valores será assumido como o padrão.

Além disso, lembre-se que você também pode criar outras variáveis, a partir de transformações e combinações das variáveis originais. Por exemplo, você poderia calcular o $\log$ ou a raiz quadrada de alguma variável numérica, ou ainda calcular uma nova variável que é igual ao produto de duas variáveis originais. Uma prática comum é considerar uma variável adicional, sempre igual a 1, o que faz com que o modelo inclua um termo constante, independente dos dados de entrada, conhecida como *intercepto* na terminologia de estatística ou *bias* no contexto de redes neurais.

A proposta do exercício é que você construa um modelo de regressão linear para prever o valor da coluna `price` a partir dos dados das demais variáveis. Para tanto, você pode gerar novas variáveis a partir das originais e/ou descartar variáveis caso julgue que não contribuam para o modelo.

Em resumo, para obter o vetor $\mathbf{w}_o$ com os coeficientes do modelo de regressão linear, você deve seguir os seguintes passos:

1. Selecionar o conjunto de variáveis originais que você vai utilizar no modelo. Lembre-se que a variável `price` não pode ser utilizada pois é a variável que você deseja prever com o modelo;
2. Substituir cada variável categórica de sua seleção por um conjunto de variáveis *dummy*, conforme descrito anteriormente. Para tanto, você pode usar a função get_dummies() do Pandas ou escrever a sua própria função para fazer a transformação;
3. Transformar as variáveis originais de sua seleção e / ou incluir combinações, caso julgue necessário;
4. A partir de sua seleção de dados, obter a matriz $\mathbf{X}$ e o vetor $\mathbf{d}$, que podem ser representados como *arrays* do NumPy. Caso você tenha usado um *DataFrame* do Pandas para organizar a sua seleção de variáveis, você pode obter um *array* do NumPy usando o método .to_numpy().
5. Usando a matriz $\mathbf{X}$ e o vetor $\mathbf{d}$, calcular o vetor $\mathbf{w}_o$ conforme mostrado na aula.

Após obter os coeficientes $\mathbf{w}_o$ do modelo de regressão linear, você vai utilizá-los para prever o valor de venda de carros de um conjunto de dados de teste disponível em um arquivo CSV do repositório de materiais. Após fazer o download do arquivo, repita as transformações que você fez no conjunto de treinamento, obtendo a matriz $\mathbf{X}_{\text{teste}}$ e o vetor $\mathbf{d}_{\text{teste}}$.

Para cada carro do banco de dados de teste, calcule o valor predito pelo seu modelo e o erro em relação ao valor da coluna `price`. Calcule também o erro quadrático médio considerando todo o banco de dados de teste.

Ao final do exercício, você deverá apresentar:

1. Uma descrição das variáveis de entrada que você utilizou como entrada e as justificativas para descartar variáveis ou utilizar transformações e combinações;
2. Os códigos utilizados para calcular o vetor $\mathbf{w}_o$ e o erro quadrático médio de seu modelo, considerando os dados de teste;
3. O valor obtido para o erro quadrático médio de seu modelo, considerando os dados de teste. Esse valor será utilizado para fazer um *ranking* dos melhores modelos.

A sugestão é que seja apresentado um Jupyter Notebook usando a linguagem Python, já que essas são as ferramentas que estamos utilizando nesta parte do curso. No entanto, isso não é obrigatório e você pode usar outra linguagem de programação, caso queira.

De Magno T. M. Silva e Renato Candido