# MO432 - Trabalho 1

Jônatas Trabuco Belotti
RA: 230260
jonatas.t.belotti@hotmail.com

## I. Introdução

O objetivo desse trabalho é realizar a leitura de uma base de dados, realizar o preprocessamento dos dados e realizar uma regressão linear com uma validação cruzada. A base de dados utilizada na regressão é uma base com dados com atributos de carros e o objeto é prever o preço de venda de cada carro. Disponibilizada através de um arquivo *csv*, a base contém 301 amostras com 8 atributos de entrada e 1 valor de saída e pode ser acessada através do link `https://www.ic.unicamp.br/~wainer/cursos/1s2020/432/car-data.csv`.

Todos os códigos deste trabalho foram escritos utilizando a linguagem de programação **Python 3.6.9** com as seguintes bibliotecas: **Numpy 1.18.2**, **Scikit-Learn 0.22.2** e **Matplotlib 3.2.2**. Sendo estas obrigatórias para execução dos códigos. Todo o código desenvolvido juntamente com os arquivos utilizados para a Regressão Linear podem ser acessados no repositório do GitHub `https://github.com/jonatastbelotti/mo432-trab1`.

## II. Execução

Para a execução do trabalho foi proposta uma lista de atividades a serem cumpridas. São elas:

- Leitura do arquivo *csv*;
- Conversão dos atributos categóricos;
- Centering and Scaling;
- Redução de dimensionalidade;
- Validação cruzada e regressão linear.

As próximas seções descrevem em detalhes cada um dos passos mencionados.

### A. Leitura do arquivo csv

O arquivo com os dados de entrada foi lido através da classe **DictReader** do pacote **csv**. Os atributos são:

- **Year -** Ano de fabricação do carro;
- **Present_Price -** Preço atual do carro;
- **Kms_Driven -** Quilômetros rodados;
- **Owner -** Quantos donos esse carro já teve;
- **Car_Name -** Modelo do carro;
- **Fuel_Type -** Tipo de combustível;
- **Seller_Type -** Tipo de vendedor (Concessionária ou pessoa);
- **Transmission -** Tipo de transmissão do carro;
- **Selling_Price -** Preço de venda do carro (atributo de saída).

Nesse ponto os dados foram separados em dados de entrada e dados de saída.

### B. Conversão dos atributos categóricos

Alguns dos atributos de entrada são dados não numéricos (categóricos), portanto, se faz necessário converter esses dados para transformá-los em dados numéricos. Os atributos categóricos são: Car_Name, Fuel_Type, Seller_Type e Transmission.

Para realizar a conversão dos atributos categóricos foi utilizada a classe **OneHotEncoder** do pacote **Scikit-Learn**. Após a conversão os 8 atributos de entrada se transformaram em 109.

### C. Centering and Scaling

A padronização dos dados é uma etapa fundamental, nesse trabalho foram aplicadas as técnicas de *Centering* e *Standard Scaling* para todos os atributos de entrada. Essas duas técnicas foram aplicadas através da classe **StandardScaler** do pacote **Scikit-Learn**.

### D. Redução de dimensionalidade

A redução de dimensionalidade ajuda a diminuir o tempo de treinamento dos modelos de previsão. Nesse trabalho foi utilizada a Análise de componentes principais (PCA, do inglês *Principal Component Analysis*) como técnica para estipular quantos atributos são necessários para manter certa taxa da variância das amostras.

O gráfico da Fígura 1 relaciona a taxa de variância das amostras que é mantida com cada quantidade de atributos dos dados de treinamento.

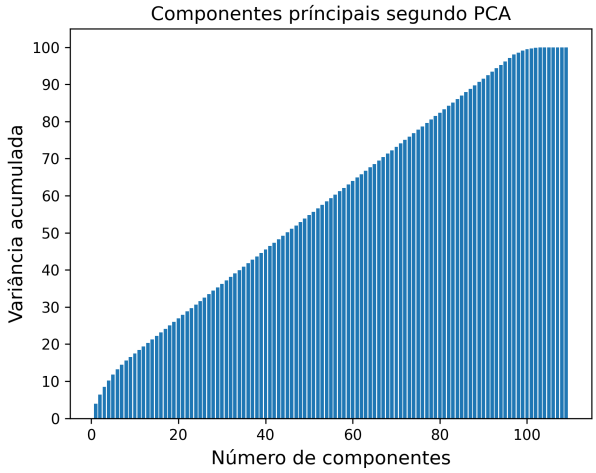


Fig. 1: Número de componentes principais segundo PCA.

Como pode ser observado no gráfico da Fígura 1 para se alcançar uma variância nas amostras de $90\%$ são necessários $89$ atributos dos dados de entrada. Entretanto, como descrito pelo professor, a utilização desses $89$ atributos na Regressão Linear resulta em uma previsão muito ruim. Portanto, as $109$ dimensões dos dados de entrada foram reduzidas para $10$.

### *E. Validação cruzada e regressão linear*

Para o treinamento e a execução da Regressão Linear os dados foram divididos em $2$ conjuntos, o conjunto de treinamento com $70\%$ dos dados e o conjunto de teste com $30\%$. Também foram aplicadas $5$ repetições de Validação Cruzada, onde em cada repetição os conjuntos de treinamento e teste são gerados de forma aleatória. Foram medidos os valores de RMSE (*Root Mean Square Error*) e MAE (*Mean Absolute Error*) para as previões realizadas no conjunto de teste.

Para montar os conjuntos de treinamento e teste em cada repetição da validação cruzada foi utilizada a função **train_test_split** e como algoritmo de Regressão Linear foi utilizada a classe **LinearRegression**, ambos do pacote **Scikit-Learn**.

A Tabela I contém os valores de RMSE e MAE para cada uma das $5$ repetições da Validação Cruzada. Além disso, a tabela também mostra a média desses valores.

TABELA I: Resultados para o conjunto de teste em cada repetição da validação cruzada.

| Repetição | RMSE | MAE |
|---|---|---|
| **1** | 2.599543 | 1.662949 |
| **2** | 1.527589 | 1.169432 |
| **3** | 2.732607 | 1.878950 |
| **4** | 2.334098 | 1.405526 |
| **5** | 1.886773 | 1.417125 |
| **Média** | 2.216122 | 1.506796 |

Como pode ser visto nos dados da Tabela I, os menores valores de RMSE e MAE foram alcançados na iteração $2$. Sendo a média do RMSE de $2.216122$ dólares na previsão de cada carro.