UNIVERSIDADE FEDERAL
DO RIO DE JANEIRO

Aluno: Paulo Victor Innocencio

DRE: 116213599

Professor: Heraldo L. S. Almeida

Disciplina: Aprendizado de Máquina (EEL891)

Regressão Multivariável

Objetivo

Estimar o valor de um diamante a partir de suas características tais como dimensões geométricas, peso, cor, grau de pureza e qualidade do corte.

Dados fornecidos

Foi fornecido um conjunto de dados contendo informações a respeito de 53940 diamantes comercializados no mercado. Os dados foram separados em ***conjunto de treinamento*** e ***conjunto de teste*** (o que iria ser previsto), o primeiro com 33940 amostras e o segundo com 20000 amostras

Adaptação dos dados

Para a ser realizada a predição dos preços dos diamantes, todos os atributos devem ter forma de números para serem alocados no algoritmo matemático. Alguns do atributos da tabela se encontravam na forma de texto, como podemos ver abaixo:

- **cut - qualidade da lapidação, em uma escala categórica ordinal com os seguinte valores:**
  "Fair" = aceitável (classificação de menor valor)
  "Good" = boa
  "Very Good" = muito boa
  "Premium" = excelente
  "Ideal" = perfeita (classificação de maior valor)
- **color - cor, em uma escala categórica ordinal com os seguintes valores:**
  "D" = excepcionalmente incolor extra (classificação de maior valor)
  "E" = excepcionalmente incolor
  "F" = perfeitamente incolor
  "G" = nitidamente incolor
  "H" = incolor
  "I" = cor levemente perceptível
  "J" = cor perceptível (classificação de menor valor)
- **clarity - pureza, em uma escala categórica ordinal com os seguintes valores:**
  "I1" = inclusões evidentes com lupa de 10x (classificação de menor valor)
  "SI2" e "SI1" = inclusões pequenas, mas fáceis de serem visualizadas com lupa de 10x
  "VS2" e "VS1" = inclusões muito pequenas e difíceis de serem visualizadas com lupa de 10x
  "VVS2" e "VVS1" = inclusões extremamente pequenas e muito difíceis de serem visualizadas com lupa de 10x
  "IF" = livre de inclusões (classificação de maior valor)

***Todas as informações acima foram retiradas da descrição do problema no site Kaggle.**

Como podemos ver na imagem abaixo, foi necessário a criação de um dicionário para podermos trabalhar com esses atributos.

```
# Criando Dicionário
cut_class_dict = {"Fair": 1, "Good": 2, "Very Good": 3, "Premium": 4, "Ideal": 5}
clarity_dict = {"I1": 1, "SI2": 2, "SI1": 3, "VS2": 4, "VS1": 5, "VVS2": 6, "VVS1": 7, "IF": 8}
color_dict = {"J": 1,"I": 2,"H": 3,"G": 4,"F": 5,"E": 6,"D": 7}
```

Mapeando a informação para o dataframe de treinamento e excluindo a coluna referente ao ***id*** de cada diamante, pois a mesma segue uma ordem crescente de incremento de 1, o que poderia causar um erro na predição final. Por exemplo, um diamante de ***id*** igual a 20000 não pode ter mais relevância em comparação a outro diamante de ***id*** de valor 20, assim por diante. Abaixo podemos ver o mapeamento da informação no dataframe de treinamento.

```
df['cut'] = df['cut'].map(cut_class_dict)
df['clarity'] = df['clarity'].map(clarity_dict)
df['color'] = df['color'].map(color_dict)
df = df.drop("id",axis=1) #Minha proposta

df.head()
```

O mesmo procedimento ocorre no dataframe de teste:

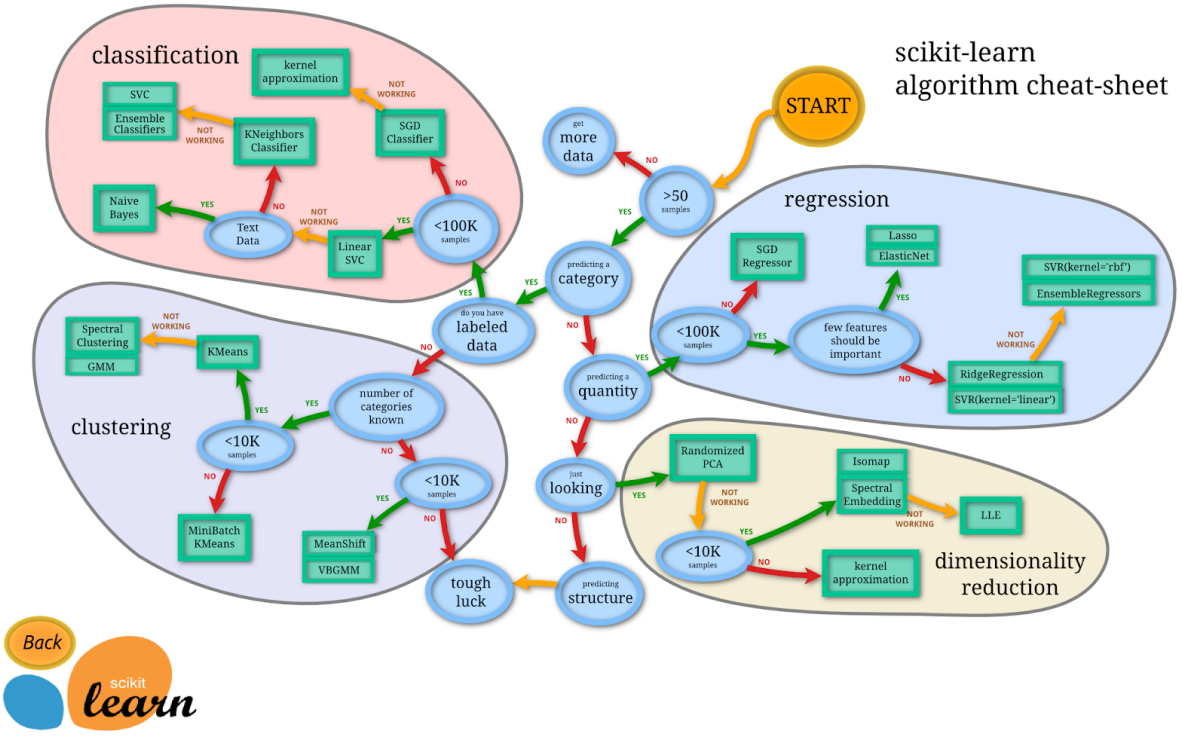
```
df2['cut'] = df2['cut'].map(cut_class_dict)
df2['clarity'] = df2['clarity'].map(clarity_dict)
df2['color'] = df2['color'].map(color_dict)
df2 = df2.drop("id",axis=1) #Minha proposta
df2['price'] = 0

df2.head()
```

O comando ***shuffle*** deveria ser usado se os dados de treinamento seguissem algum tipo de ordenamento, o que poderia manipular o resultado final. Felizmente, não foi necessário utilizar essa linha de código. (Segue como comentário no código fonte)

```
# Embaralhar os dados seria necessário se os mesmos seguissem algum tipo de ordem.
# Poderia influenciar negativamente os resultados finais
df = sklearn.utils.shuffle(df)
```

Seleção do algoritmo de regressão

Para a escolha do algoritmo que auxiliasse na predição dos preços das amostras referentes aos diamantes, foi seguido o seguinte esquema:



Seguindo o esquema desde seu ponto inicial e como queremos predizer uma quantidade em valor numérico, chegamos a opção de ***SVR(kernel='linear').***
Abaixo podemos observar a importação das bibliotecas e módulos necessários para regressão e classificação do algoritmo:

```
import sklearn
from sklearn import svm, preprocessing
```

Support Vector Machine é uma classe de algoritmos supervisionados para ambas, classificação e regressão

Ao final de todo processo de regressão e classificação, o algoritmo nos determinou alguns preços com valor negativo, algo que não condiz com o mundo real. Com isso foi necessário a utilização do ***"SVR(kernel='rbf')"***, que mesmo possuindo um ***Score*** de predição um pouco mais baixo comparado com seu modelo anterior, corrigia

todos os problemas com valores abaixo de zero.Também possuía um tempo maior de execução.
Na figura abaixo podemos observar a separação dos dados para aplicação no algoritmo de predição:

```
X = df.drop("price", axis=1).values
X = preprocessing.scale(X)
y = df['price'].values

X2 = df2.drop("price", axis=1).values
X2 = preprocessing.scale(X2)
y2 = df2['price'].values

X_train = X
y_train = y

X_test = X2
y_test = y2
```

Chamada do algoritmo de predição utilizado:

```
clf = svm.SVR(kernel="rbf")
clf.fit(X_train, y_train)

print(clf.score(X_train, y_train))
```

Tabela com a prévia das 10 primeiras predições:

id	**price (US Dollars)**
0	6433
1	5075
2	2128
3	4681
4	2736
5	2668
6	89
7	6137

8	4657
9	1965

Salvando em arquivo texto as predições encontradas:

```
# Criando arquivo texto com as predicoes
file_builder = open("predicted2.txt", "w+")
for X,y in zip(X_test, y_test):
    clf.predict([X])[0]= round((clf.predict([X])[0]),2)
    file_builder.write(f"{clf.predict([X])[0]} \n")
    print(f"Model:{clf.predict([X])[0]}")

file_builder.close()
```

Bibliografia e Referências

- https://pt.wikipedia.org/wiki/Scikit-learn
- https://pythonprogramming.net/machine-learning-python3-pandas-data-analysis/
- https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.RandomForestRegressor.html
- https://paulovasconcellos.com.br/28-comandos-%C3%BAteis-de-pandas-que-talvez-voc%C3%AA-n%C3%A3o-conhe%C3%A7a-6ab64beefa93
- Slides e códigos fontes utilizados em aula
- As ilustrações presentes neste relatório não são de minha autoria e foram retiradas da internet