Experiment 2: Regression

Pravesh Ganwani
T.E. I.T.
Batch B
Roll No. 2018140021

Aim:
To use linear regression for predictions on a dataset

Problem Statement:
Choose a regression dataset of your choice from any of the following Repository Links, download it:

1. Kaggle: https://www.kaggle.com/
2. UCI Machine Learning Repository: https://archive.ics.uci.edu/ml/index.php

Perform Linear Regression on the chosen dataset.

Your notebook should contain:
1. Basic EDA
2. Creation of Linear Regression Model using sklearn and subsequent training on Training set
3. Print out the coefficients and intercept after training
4. Test the model on test set by:
    a. Observing Scatter plot of actual vs predicted
    b. Using Standard evaluation metrics for regression

[***Hint***: Follow the steps in Boston Housing Linear Regression notebook uploaded on moodle under tutorial 3]

Tool/Language:
Programming language: Python
Libraries: numpy, pandas, sklearn, matplotlib, seaborn

Code with visualisation graphs:

1) Dataset Chosen: Red Wine Quality

2) Dataset Description: This dataset is related to red variants of the Portuguese "Vinho Verde" wine.

a) Input variables (based on physicochemical tests):
- fixed acidity (most acids involved with wine or fixed or non-volatile)
- volatile acidity (the amount of acetic acid in wine, which at too high of levels can lead to an unpleasant, vinegar taste)
- citric acid (found in small quantities, citric acid can add 'freshness' and flavour to wines)
- residual sugar (the amount of sugar remaining after fermentation stops, it's rare to find wines with less than 1 gram/litre)
- chlorides (the amount of salt in the wine)

- free Sulphur dioxide (the free form of SO2 exists in equilibrium between molecular SO2 (as a dissolved gas) and bisulphite ion)
- total Sulphur dioxide (amount of free and bound forms of S02)
- density (the density of water is close to that of water depending on the percent alcohol and sugar content)
- pH (describes how acidic or basic a wine is on a scale from 0 (very acidic) to 14 (very basic))
- sulphates (a wine additive which can contribute to Sulphur dioxide gas (S02) levels, which acts as an antimicrobial)
- alcohol

b) Output variable (based on sensory data):

- quality (score between 0 and 10)

3) Code:

```
from google.colab import files
uploaded = files.upload()

# Basic Pre-processing
import numpy as np
import pandas as pd
import io

# For Visualizations
import matplotlib.pyplot as plt
import seaborn as sns

# For Model Selection and Training
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LinearRegression

# For Model Evaluation
from sklearn import metrics
from sklearn.metrics import r2_score

df = pd.read_csv(io.BytesIO(uploaded['winequality-red.csv']))
df.head()
```

	fixed acidity	volatile acidity	citric acid	residual sugar	chlorides	free sulfur dioxide	total sulfur dioxide	density	pH	sulphates	alcohol	quality
0	7.4	0.70	0.00	1.9	0.076	11.0	34.0	0.9978	3.51	0.56	9.4	5
1	7.8	0.88	0.00	2.6	0.098	25.0	67.0	0.9968	3.20	0.68	9.8	5
2	7.8	0.76	0.04	2.3	0.092	15.0	54.0	0.9970	3.26	0.65	9.8	5
3	11.2	0.28	0.56	1.9	0.075	17.0	60.0	0.9980	3.16	0.58	9.8	6
4	7.4	0.70	0.00	1.9	0.076	11.0	34.0	0.9978	3.51	0.56	9.4	5

```
df.shape
```

```
(1599, 12)
```

```
df.columns
```

```
Index(['fixed acidity', 'volatile acidity', 'citric acid', 'residual sugar',
       'chlorides', 'free sulfur dioxide', 'total sulfur dioxide', 'density',
       'pH', 'sulphates', 'alcohol', 'quality'],
      dtype='object')
```

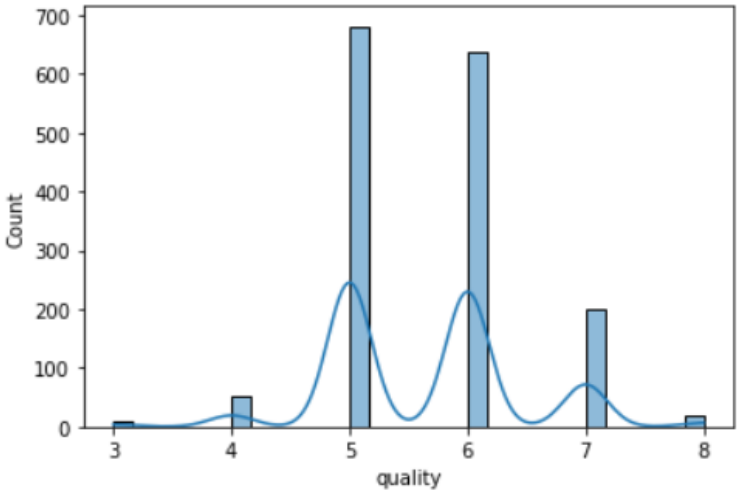
```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1599 entries, 0 to 1598
Data columns (total 12 columns):
 #   Column                Non-Null Count  Dtype  
---  ------                --------------  -----  
 0   fixed acidity         1599 non-null   float64
 1   volatile acidity      1599 non-null   float64
 2   citric acid           1599 non-null   float64
 3   residual sugar        1599 non-null   float64
 4   chlorides             1599 non-null   float64
 5   free sulfur dioxide   1599 non-null   float64
 6   total sulfur dioxide  1599 non-null   float64
 7   density               1599 non-null   float64
 8   pH                    1599 non-null   float64
 9   sulphates             1599 non-null   float64
 10  alcohol               1599 non-null   float64
 11  quality               1599 non-null   int64  
dtypes: float64(11), int64(1)
memory usage: 150.0 KB
```

```
df.describe()
```

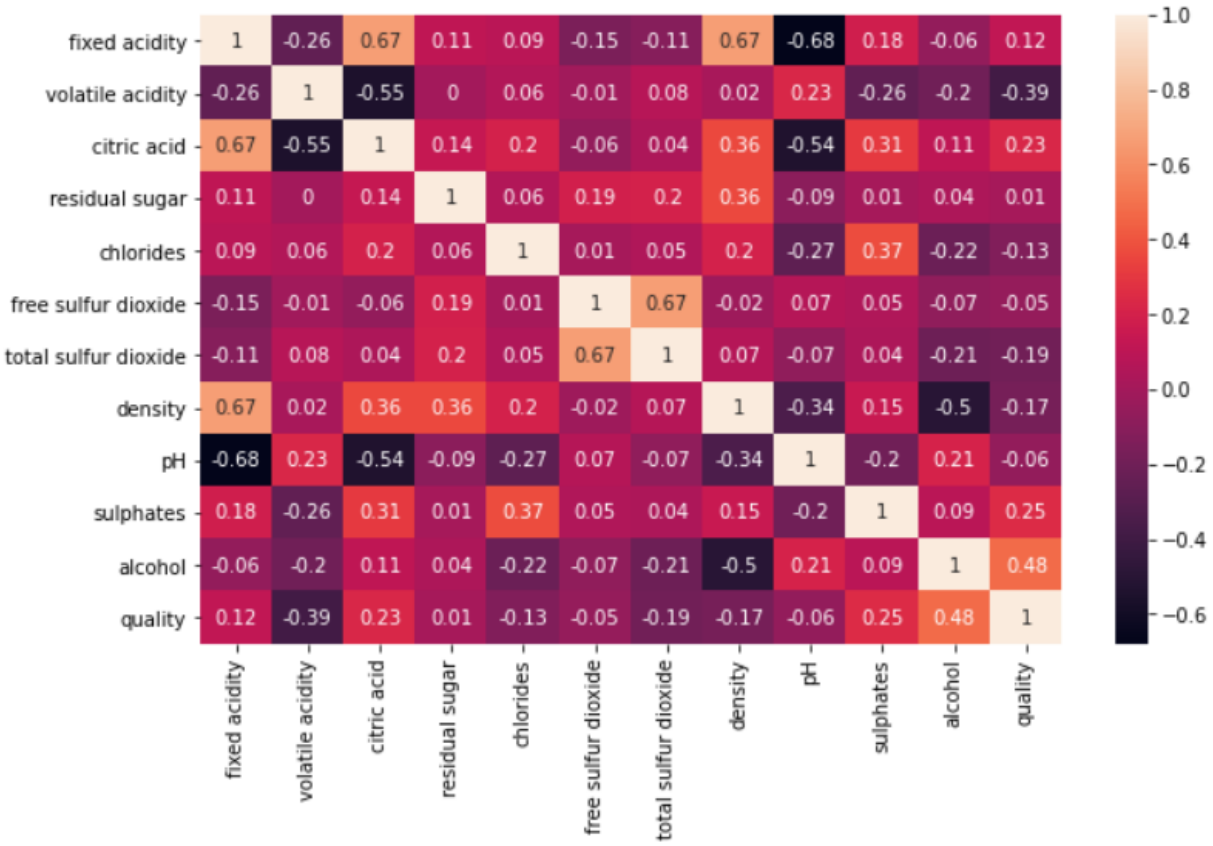
	fixed acidity	volatile acidity	citric acid	residual sugar	chlorides	free sulfur dioxide	total sulfur dioxide	density	pH	sulphates	alcohol	quality
count	1599.000000	1599.000000	1599.000000	1599.000000	1599.000000	1599.000000	1599.000000	1599.000000	1599.000000	1599.000000	1599.000000	1599.000000
mean	8.319637	0.527821	0.270976	2.538806	0.087467	15.874922	46.467792	0.996747	3.311113	0.658149	10.422983	5.636023
std	1.741096	0.179060	0.194801	1.409928	0.047065	10.460157	32.895324	0.001887	0.154386	0.169507	1.065668	0.807569
min	4.600000	0.120000	0.000000	0.900000	0.012000	1.000000	6.000000	0.990070	2.740000	0.330000	8.400000	3.000000
25%	7.100000	0.390000	0.090000	1.900000	0.070000	7.000000	22.000000	0.995600	3.210000	0.550000	9.500000	5.000000
50%	7.900000	0.520000	0.260000	2.200000	0.079000	14.000000	38.000000	0.996750	3.310000	0.620000	10.200000	6.000000
75%	9.200000	0.640000	0.420000	2.600000	0.090000	21.000000	62.000000	0.997835	3.400000	0.730000	11.100000	6.000000
max	15.900000	1.580000	1.000000	15.500000	0.611000	72.000000	289.000000	1.003690	4.010000	2.000000	14.900000	8.000000

```
hs=sns.histplot(df['quality'],kde=True)
```




```
Observation: The above graph is a histogram that plots the
distribution of the output feature.

fig = plt.figure(figsize=(10,6))
sns.heatmap(df.corr().round(2),annot=True)
```



Observation: The above heatmap plot indicated the relations between different features of the dataset. We can conclude that no 2 features are very closely related. However, we come to know that the quality of the wine depends greatly on the alcohol content in it (alcohol correlation is at 0.48 in the heatmap).

```
X = df[df.columns]
Y = df['quality']

X_train, X_test, Y_train, Y_test = train_test_split(X, Y, test_size=0.4, r
andom_state=101)

X_train.shape
```

```
(959, 11)
```

```
lm = LinearRegression()

lm.fit(X_train, Y_train)

print('Intercept: ', lm.intercept_)
print('Coefficients for All Features: ', lm.coef_)
```

```
Intercept:  11.411886829961531
Coefficients for All Features:  [ 4.31171293e-02 -1.20909413e+00 -3.79645694e-01  1.38686432e-02
 -1.35313410e+00  1.50959899e-03 -3.22985489e-03 -7.17707938e+00
 -4.91493826e-01  7.49240935e-01  2.99338069e-01]
```

```
coeff_df = pd.DataFrame(lm.coef_,X.columns,columns=['Coefficient'])
coeff_df
```

	Coefficient
fixed acidity	0.043117
volatile acidity	-1.209094
citric acid	-0.379646
residual sugar	0.013869
chlorides	-1.353134
free sulfur dioxide	0.001510
total sulfur dioxide	-0.003230
density	-7.177079
pH	-0.491494
sulphates	0.749241
alcohol	0.299338

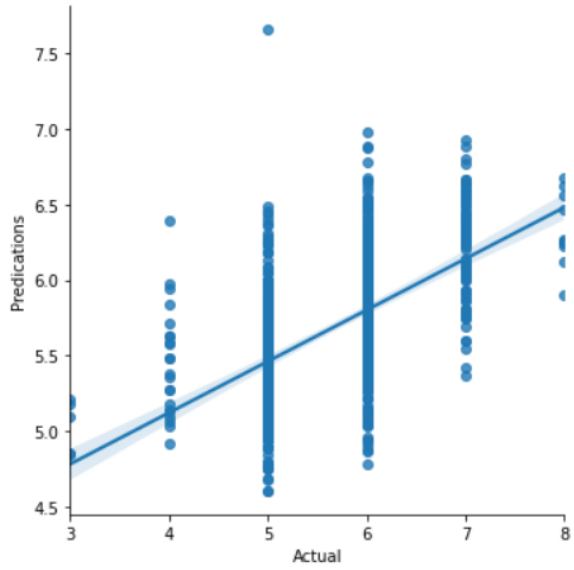
```
predictions = lm.predict(X_test)

df2 = pd.DataFrame(data=list(zip(Y_test, predictions)), columns=['Actual',
 'Predications'])
df2.head()
```

	Actual	Predications
0	5	5.017292
1	5	5.251785
2	5	5.010907
3	7	5.794375
4	6	6.207566

```
sns.lmplot(x='Actual', y='Predications', data=df2)
```

```
<seaborn.axisgrid.FacetGrid at 0x7f1e1a8e79b0>
```




```
print('MAE:', metrics.mean_absolute_error(Y_test, predictions))
```

```
MAE: 0.5291196968072086
MSE: 0.47645322202824064
RMSE: 0.6902559105348107
```

```
print('MSE:', metrics.mean_squared_error(Y_test, predictions))
print('RMSE:', np.sqrt(metrics.mean_squared_error(Y_test, predictions)))

print('R2 Score:', r2_score(Y_test,predictions))
```

```
R2 Score: 0.3161469789233594
```

Conclusion:

Thus, we learnt the basics of Linear Regression modeling and how linear data fits in a Linear Regression model. We also learnt the use of sklearn library to work with different basic Machine Learning models. We also understood the different parameters/metrics used to evaluate the predicted output of the model.