Nama : Tri Hesti Wahyuningsih

“ Tugas Sertifikasi AIBIZ (SIB BLOCKCHAIN)”

RED WINE QUALITY USING LINIER REGRESION

The two datasets are related to red and white variants of the Portuguese "Vinho Verde" wine. These datasets can be viewed as classification or regression tasks. The classes are ordered and not balanced (e.g. there are much more normal wines than excellent or poor ones).

Importing the necessary libraries

```
[ ]  1  import numpy as np
     2  import pandas as pd
     3  import matplotlib.pyplot as plt
     4  %matplotlib inline
     5  import seaborn as sns
```

Importing the dataset

```
     1  from google.colab import drive
     2  drive.mount('/content/drive')
```

```
Mounted at /content/drive
```

```
[ ]  1  #Mengimport data
     2  df = pd.read_csv('/content/drive/MyDrive/AI BLOKCHAIN/Course mandiri/Red Wine Quality/win
     3  df.head()
```

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

Checking for Null Values

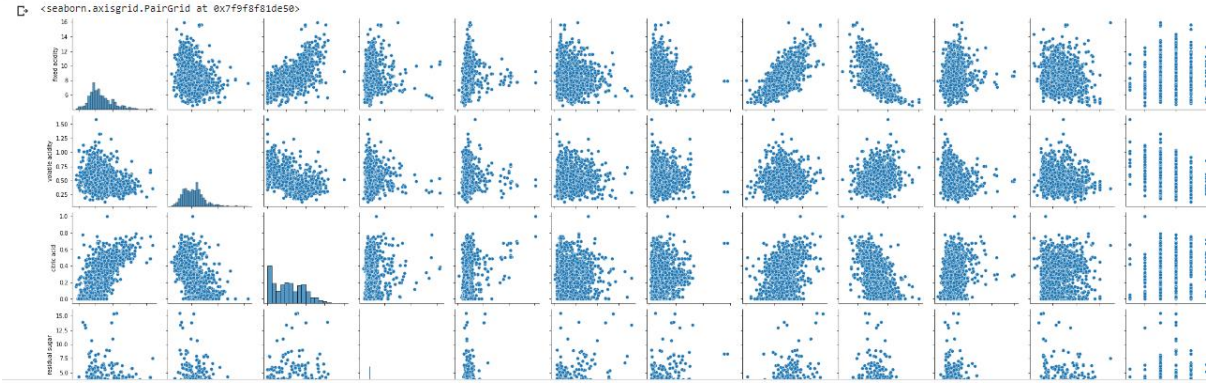
```
[ ]  1  df.isnull().sum()
```

```
fixed acidity           0
volatile acidity        0
citric acid             0
residual sugar          0
chlorides               0
free sulfur dioxide     0
total sulfur dioxide    0
density                 0
pH                      0
sulphates               0
alcohol                 0
quality                 0
dtype: int64
```

Some Eksploratory Data Analysis

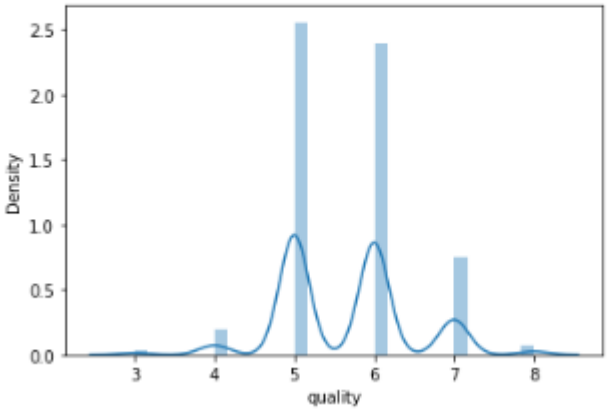
```
1  sns.pairplot(df)
```

```
<seaborn.axisgrid.PairGrid at 0x7f9f8f81de50>
```



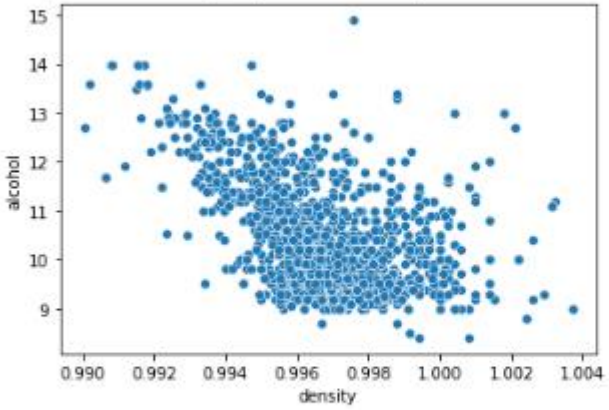
```
[ ]  1  sns.distplot(df['quality'])
```

```
/usr/local/lib/python3.8/dist-packages/seaborn/distributions.py:2619:
  warnings.warn(msg, FutureWarning)
<matplotlib.axes._subplots.AxesSubplot at 0x7f9f89884670>
```



```
1  sns.scatterplot(x='density', y='alcohol', data=df)
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x7f9f89e95d60>
```




```
[ ]  1  x = df.drop('quality',axis=1)
     2  y = df['quality']
```

Linier regression

```
[ ]  1  from sklearn.linear_model import LinearRegression
```

```
[ ]  1  lm = LinearRegression()
```

```
1  from sklearn.model_selection import train_test_split
2  x_train, x_test, y_train, y_test = train_test_split(x, y, test_size=0.3, random_state=101)
```

```
[ ]  1  lm.fit(x_train,y_train)
```

```
LinearRegression()
```

```
[ ]  1  print(lm.intercept_)
```

```
13.424238071571002
```

```
[ ]  1  pred_train = lm.predict(x_train)
```

```
[ ]  1  pred_test = lm.predict(x_test)
```

```
[ ]  1  from sklearn import metrics
```

```
[ ]  1  print('MAE:', metrics.mean_absolute_error(y_train, pred_train))
     2  print('MSE:', metrics.mean_squared_error(y_train, pred_train))
     3  print('RMSE:', np.sqrt(metrics.mean_squared_error(y_train, pred_train)))
```

```
MAE: 0.4894920791026185
MSE: 0.388883165843459
RMSE: 0.623604975800754
```

```
[ ]  1  print('MAE:', metrics.mean_absolute_error(y_test, pred_test))
     2  print('MSE:', metrics.mean_squared_error(y_test, pred_test))
     3  print('RMSE:', np.sqrt(metrics.mean_squared_error(y_test, pred_test)))
```

```
MAE: 0.5330378570424547
MSE: 0.4908886154893248
RMSE: 0.700634437841393
```