Problem statement

Data collection

In [1]:

```
#to import libraries
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
```

In [2]:

```
df=pd.read_csv(r"C:\Users\user\Downloads\11_winequality-red.csv")[0:500]
df
```

Out[2]:

	fixed acidity	volatile acidity	citric acid	residual sugar	chlorides	free sulfur dioxide	total sulfur dioxide	density	pH	sulphates	al
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	
...	...	...	...	...	...	...	...	...	...	...	
495	10.7	0.35	0.53	2.6	0.070	5.0	16.0	0.9972	3.15	0.65	
496	7.8	0.52	0.25	1.9	0.081	14.0	38.0	0.9984	3.43	0.65	
497	7.2	0.34	0.32	2.5	0.090	43.0	113.0	0.9966	3.32	0.79	
498	10.7	0.35	0.53	2.6	0.070	5.0	16.0	0.9972	3.15	0.65	
499	8.7	0.69	0.31	3.0	0.086	23.0	81.0	1.0002	3.48	0.74	

500 rows × 12 columns

In [3]:

```
df.head()
```

Out[3]:

	fixed acidity	volatile acidity	citric acid	residual sugar	chlorides	free sulfur dioxide	total sulfur dioxide	density	pH	sulphates	alco
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	

In [4]:

```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 500 entries, 0 to 499
Data columns (total 12 columns):
 #   Column                Non-Null Count  Dtype  
---  ------                --------------  -----  
 0   fixed acidity         500 non-null    float64
 1   volatile acidity      500 non-null    float64
 2   citric acid           500 non-null    float64
 3   residual sugar        500 non-null    float64
 4   chlorides             500 non-null    float64
 5   free sulfur dioxide   500 non-null    float64
 6   total sulfur dioxide  500 non-null    float64
 7   density               500 non-null    float64
 8   pH                    500 non-null    float64
 9   sulphates             500 non-null    float64
 10  alcohol               500 non-null    float64
 11  quality               500 non-null    int64  
dtypes: float64(11), int64(1)
memory usage: 47.0 KB
```

In [5]:

```
#to display summary of statistics
df.describe()
```

Out[5]:

	fixed acidity	volatile acidity	citric acid	residual sugar	chlorides	free sulfur dioxide	total sulfur dioxide	
count	500.00000	500.000000	500.000000	500.000000	500.000000	500.000000	500.000000	5
mean	8.68640	0.533370	0.302460	2.586800	0.093962	15.041000	51.444000	
std	1.88393	0.176169	0.216569	1.382229	0.060240	9.783673	33.716947	
min	4.60000	0.180000	0.000000	1.200000	0.039000	3.000000	8.000000	
25%	7.40000	0.400000	0.107500	1.900000	0.073000	7.000000	25.000000	
50%	8.10000	0.530000	0.275000	2.200000	0.082000	12.000000	42.000000	
75%	9.82500	0.645000	0.480000	2.700000	0.093000	20.000000	67.000000	
max	15.60000	1.330000	1.000000	15.500000	0.611000	68.000000	165.000000	

In [6]:

```
#to display cloumn heading
df.columns
```

Out[6]:

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

EDA and VISUALIZATION

In [7]:

```
df1=df[['fixed acidity', 'volatile acidity', 'citric acid', 'residual sugar',
       'chlorides', 'free sulfur dioxide']]
df1
```

Out[7]:

	fixed acidity	volatile acidity	citric acid	residual sugar	chlorides	free sulfur dioxide
0	7.4	0.70	0.00	1.9	0.076	11.0
1	7.8	0.88	0.00	2.6	0.098	25.0
2	7.8	0.76	0.04	2.3	0.092	15.0
3	11.2	0.28	0.56	1.9	0.075	17.0
4	7.4	0.70	0.00	1.9	0.076	11.0
...	...	...	...	...	...	...
495	10.7	0.35	0.53	2.6	0.070	5.0
496	7.8	0.52	0.25	1.9	0.081	14.0
497	7.2	0.34	0.32	2.5	0.090	43.0
498	10.7	0.35	0.53	2.6	0.070	5.0
499	8.7	0.69	0.31	3.0	0.086	23.0

500 rows × 6 columns

In [8]:

```
df1.fillna(1)
```

Out[8]:

	fixed acidity	volatile acidity	citric acid	residual sugar	chlorides	free sulfur dioxide
0	7.4	0.70	0.00	1.9	0.076	11.0
1	7.8	0.88	0.00	2.6	0.098	25.0
2	7.8	0.76	0.04	2.3	0.092	15.0
3	11.2	0.28	0.56	1.9	0.075	17.0
4	7.4	0.70	0.00	1.9	0.076	11.0
...	...	...	...	...	...	...
495	10.7	0.35	0.53	2.6	0.070	5.0
496	7.8	0.52	0.25	1.9	0.081	14.0
497	7.2	0.34	0.32	2.5	0.090	43.0
498	10.7	0.35	0.53	2.6	0.070	5.0
499	8.7	0.69	0.31	3.0	0.086	23.0

500 rows × 6 columns

In [9]:

```
df1.info()
```

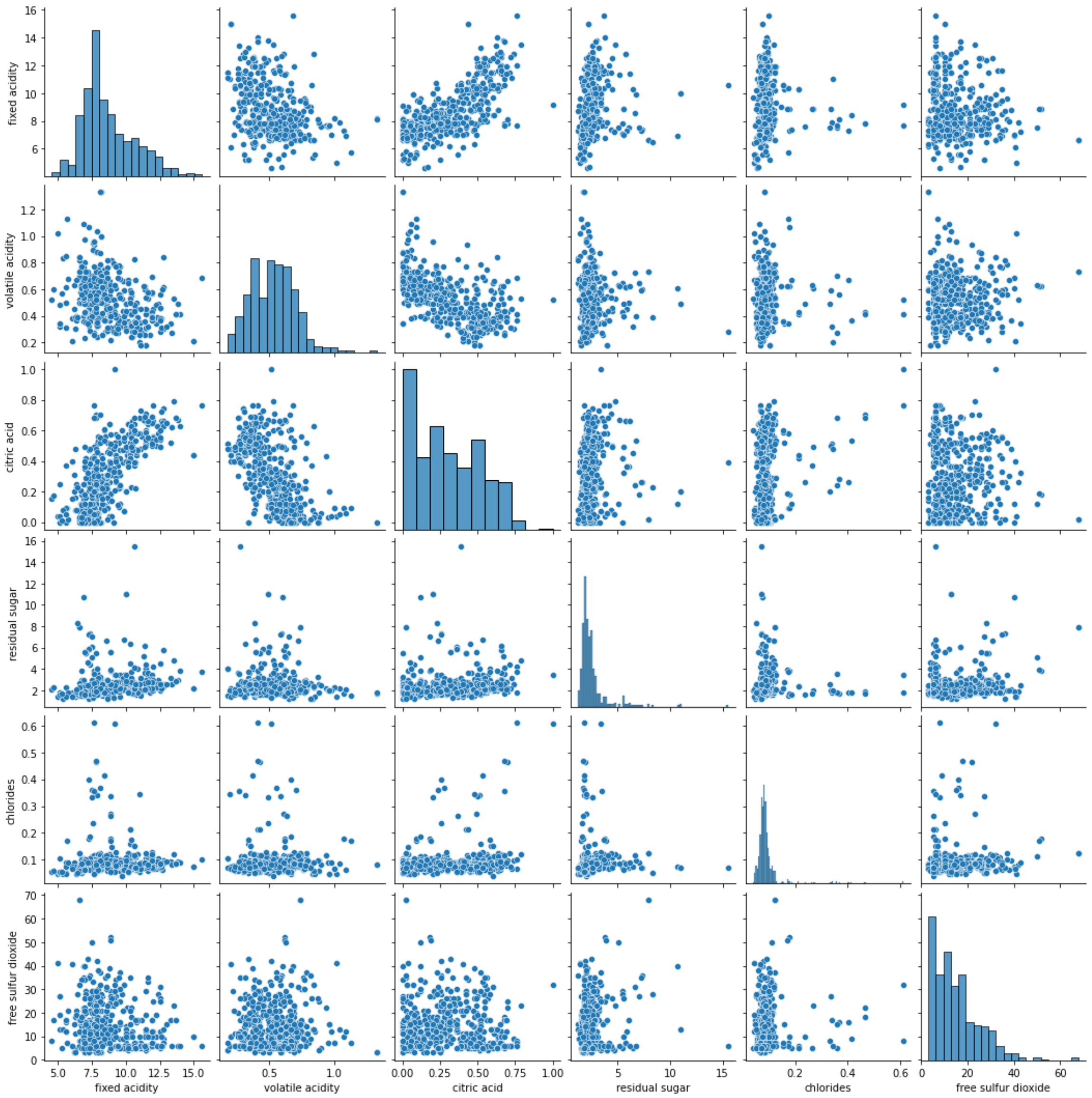
```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 500 entries, 0 to 499
Data columns (total 6 columns):
 #   Column               Non-Null Count  Dtype  
---  ------               --------------  -----  
 0   fixed acidity        500 non-null    float64
 1   volatile acidity     500 non-null    float64
 2   citric acid          500 non-null    float64
 3   residual sugar       500 non-null    float64
 4   chlorides            500 non-null    float64
 5   free sulfur dioxide  500 non-null    float64
dtypes: float64(6)
memory usage: 23.6 KB
```

In [10]:

```
sns.pairplot(df1)
```

Out[10]:

```
<seaborn.axisgrid.PairGrid at 0x1624fb70fa0>
```



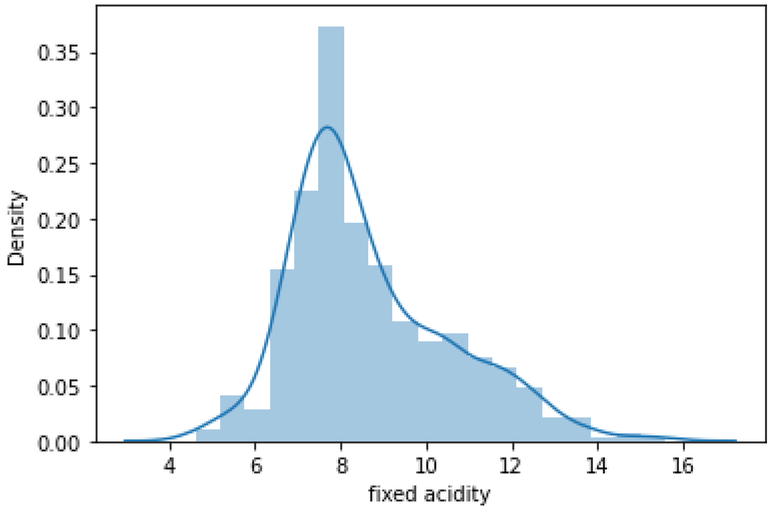
In [11]:

```
sns.distplot(df['fixed acidity'])
```

```
C:\ProgramData\Anaconda3\lib\site-packages\seaborn\distributions.py:2557:
FutureWarning: `distplot` is a deprecated function and will be removed in
a future version. Please adapt your code to use either `displot` (a figure
-level function with similar flexibility) or `histplot` (an axes-level fun
ction for histograms).
  warnings.warn(msg, FutureWarning)
```

Out[11]:

```
<AxesSubplot:xlabel='fixed acidity', ylabel='Density'>
```

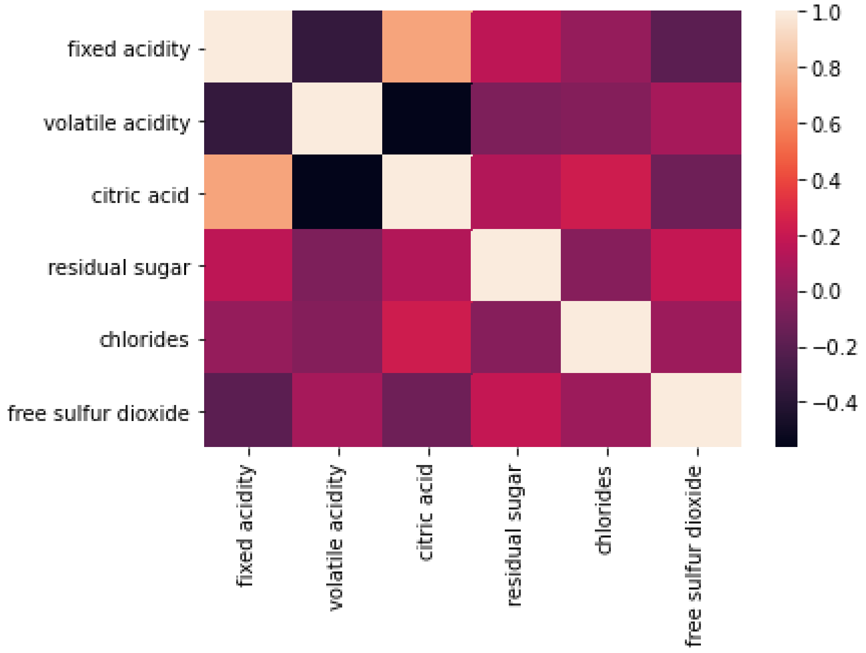


In [12]:

```
data=df1[['fixed acidity', 'volatile acidity', 'citric acid', 'residual sugar',
       'chlorides', 'free sulfur dioxide']]
sns.heatmap(data.corr())
```

Out[12]:

```
<AxesSubplot:>
```



to Train the model-Model buliding

we are going to split our data into two variable where x is a independent and y is dependent on x

In [13]:

```
x=data[['volatile acidity', 'citric acid', 'residual sugar',
       'chlorides', 'free sulfur dioxide']]
y=data['fixed acidity']
```

In [14]:

```
# to split my dataset into test and train data
from sklearn.model_selection import train_test_split
x_train,x_test,y_train,y_test = train_test_split(x,y,test_size=0.3)
```

In [15]:

```
from sklearn.linear_model import LinearRegression

lr=LinearRegression()
lr.fit(x_train,y_train)
```

Out[15]:

```
LinearRegression()
```

In [16]:

```
print(lr.intercept_)
```

```
6.691322817405667
```

In [17]:

```
coeff=pd.DataFrame(lr.coef_,x.columns,columns=['Co-effecient'])
coeff
```

Out[17]:

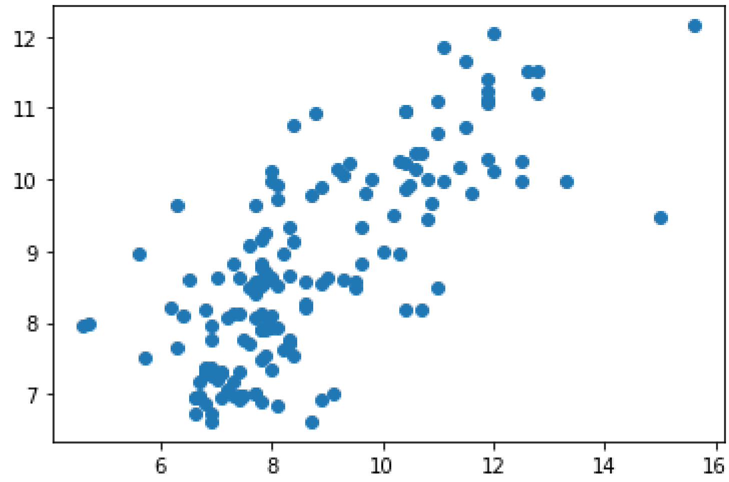
	Co-effecient
volatile acidity	0.918061
citric acid	6.653698
residual sugar	0.102520
chlorides	-4.799736
free sulfur dioxide	-0.020701

In [18]:

```
prediction=lr.predict(x_test)
plt.scatter(y_test,prediction)
```

Out[18]:

```
<matplotlib.collections.PathCollection at 0x16253449d60>
```



In [19]:

```
print(lr.score(x_test,y_test))
```

```
0.5576770170364151
```

In [ ]: