In [1]:
```
from platform import python_version


print(python_version())
```

```
3.8.8
```

In [2]:
```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
```

Data Cleaning

Step 1: Importing the dataset

In [3]:
```
wine_df = pd.read_csv('Wine.csv')
```

In [4]:
```
wine_df
```

Out[4]:

	fixed acidity	volatile acidity	citric acid	residual sugar	chlorides	free sulfur dioxide	total sulfur dioxide	density	pH	sulphates	alcohol	q
0	7.4	0.700	0.00	1.9	0.076	11.0	34.0	0.99780	3.51	0.56	9.4	
1	7.8	0.880	0.00	2.6	0.098	25.0	67.0	0.99680	3.20	0.68	9.8	
2	7.8	0.760	0.04	2.3	0.092	15.0	54.0	0.99700	3.26	0.65	9.8	
3	11.2	0.280	0.56	1.9	0.075	17.0	60.0	0.99800	3.16	0.58	9.8	
4	7.4	0.700	0.00	1.9	0.076	11.0	34.0	0.99780	3.51	0.56	9.4	
...	...	...	...	...	...	...	...	...	...	...	...	
1594	6.2	0.600	0.08	2.0	0.090	32.0	44.0	0.99490	3.52	0.58	10.5	
1595	5.9	0.550	0.10	2.2	0.062	39.0	51.0	0.99512	3.52	0.76	11.2	
1596	6.3	0.510	0.13	2.3	0.076	29.0	40.0	0.99574	3.52	0.75	11.0	
1597	5.9	0.645	0.12	2.0	0.075	32.0	44.0	0.99547	3.52	0.71	10.2	
1598	6.0	0.310	0.47	3.6	0.067	18.0	42.0	0.99549	3.52	0.66	11.0	

1599 rows × 12 columns

In [5]:
```
wine_df.head()
```

Out[5]:

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

In [6]:
```
wine_df.tail()
```

Out[6]:

	fixed acidity	volatile acidity	citric acid	residual sugar	chlorides	free sulfur dioxide	total sulfur dioxide	density	pH	sulphates	alcohol	q
1594	6.2	0.600	0.08	2.0	0.090	32.0	44.0	0.99490	3.52	0.58	10.5	
1595	5.9	0.550	0.10	2.2	0.062	39.0	51.0	0.99512	3.52	0.76	11.2	
1596	6.3	0.510	0.13	2.3	0.076	29.0	40.0	0.99574	3.52	0.75	11.0	
1597	5.9	0.645	0.12	2.0	0.075	32.0	44.0	0.99547	3.52	0.71	10.2	
1598	6.0	0.310	0.47	3.6	0.067	18.0	42.0	0.99549	3.52	0.66	11.0	

Step 2: Exploring Data

In [7]:
```
wine_df.shape
```

Out[7]:
```
(1599, 12)
```

In [8]:
```
wine_df.index
```

Out[8]:
```
RangeIndex(start=0, stop=1599, step=1)
```

In [9]:
```
wine_df.columns
```

Out[9]:
```
Index(['fixed acidity', 'volatile acidity', 'citric acid', 'residual sugar',
       'chlorides', 'free sulfur dioxide', 'total sulfur dioxide', 'density',
       'pH', 'sulphates', 'alcohol', 'quality'],
      dtype='object')
```

In [10]:
```
wine_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  1598 non-null   float64
 7   density               1599 non-null   float64
 8   pH                    1598 non-null   float64
 9   sulphates             1599 non-null   float64
 10  alcohol               1599 non-null   float64
 11  quality               1598 non-null   float64
dtypes: float64(12)
memory usage: 150.0 KB
```

In [11]:
```
wine_df.describe()
```

Out[11]:

	fixed acidity	volatile acidity	citric acid	residual sugar	chlorides	free sulfur dioxide	total sulfur dioxide	
count	1599.000000	1599.000000	1599.000000	1599.000000	1599.000000	1599.000000	1598.000000	159
mean	8.319637	0.527821	0.270976	2.538806	0.087467	15.874922	46.433041	(
std	1.741096	0.179060	0.194801	1.409928	0.047065	10.460157	32.876249	(
min	4.600000	0.120000	0.000000	0.900000	0.012000	1.000000	6.000000	(
25%	7.100000	0.390000	0.090000	1.900000	0.070000	7.000000	22.000000	(
50%	7.900000	0.520000	0.260000	2.200000	0.079000	14.000000	38.000000	(
75%	9.200000	0.640000	0.420000	2.600000	0.090000	21.000000	62.000000	(
max	15.900000	1.580000	1.000000	15.500000	0.611000	72.000000	289.000000	

Step 3: Handling Missing Values

In [12]:
```
wine_df.isna().any()
```

Out[12]:
```
fixed acidity           False
volatile acidity        False
citric acid             False
residual sugar          False
chlorides               False
free sulfur dioxide     False
total sulfur dioxide     True
density                 False
pH                       True
sulphates               False
alcohol                 False
quality                  True
dtype: bool
```

In [13]:
```
wine_df.isna().sum()
```

Out[13]:
```
fixed acidity           0
volatile acidity        0
citric acid             0
residual sugar          0
chlorides               0
free sulfur dioxide     0
total sulfur dioxide    1
density                 0
pH                      1
sulphates               0
alcohol                 0
quality                 1
dtype: int64
```

In [14]:
```
wine_df[wine_df['total sulfur dioxide'].isna()]
```

Out[14]:

	fixed acidity	volatile acidity	citric acid	residual sugar	chlorides	free sulfur dioxide	total sulfur dioxide	density	pH	sulphates	alcohol	quali
9	7.5	0.5	0.36	6.1	0.071	17.0	NaN	0.9978	3.35	0.8	10.5	5

In [15]:
```
wine_df[wine_df['pH'].isna()]
```

Out[15]:

	fixed acidity	volatile acidity	citric acid	residual sugar	chlorides	free sulfur dioxide	total sulfur dioxide	density	pH	sulphates	alcohol	qu
184	6.7	0.62	0.21	1.9	0.079	8.0	62.0	0.997	NaN	0.58	9.3	

In [16]:
```
wine_df[wine_df['quality'].isna()]
```

Out[16]:

	fixed acidity	volatile acidity	citric acid	residual sugar	chlorides	free sulfur dioxide	total sulfur dioxide	density	pH	sulphates	alcohol	qu
123	8.0	0.71	0.0	2.6	0.08	11.0	34.0	0.9976	3.44	0.53	9.5	

In [17]:
```
# Fill missing values with the mean of their respective columns
for col,value in wine_df.items():
    wine_df[col] = wine_df[col].fillna(wine_df[col].mean())
```

In [18]:
```
wine_df.isna().sum()
```

Out[18]:
```
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
```

In [19]:
```
wine_df.isna().any()
```

Out[19]:
```
fixed acidity           False
volatile acidity        False
citric acid             False
residual sugar          False
chlorides               False
free sulfur dioxide     False
total sulfur dioxide    False
density                 False
pH                      False
sulphates               False
alcohol                 False
quality                 False
dtype: bool
```

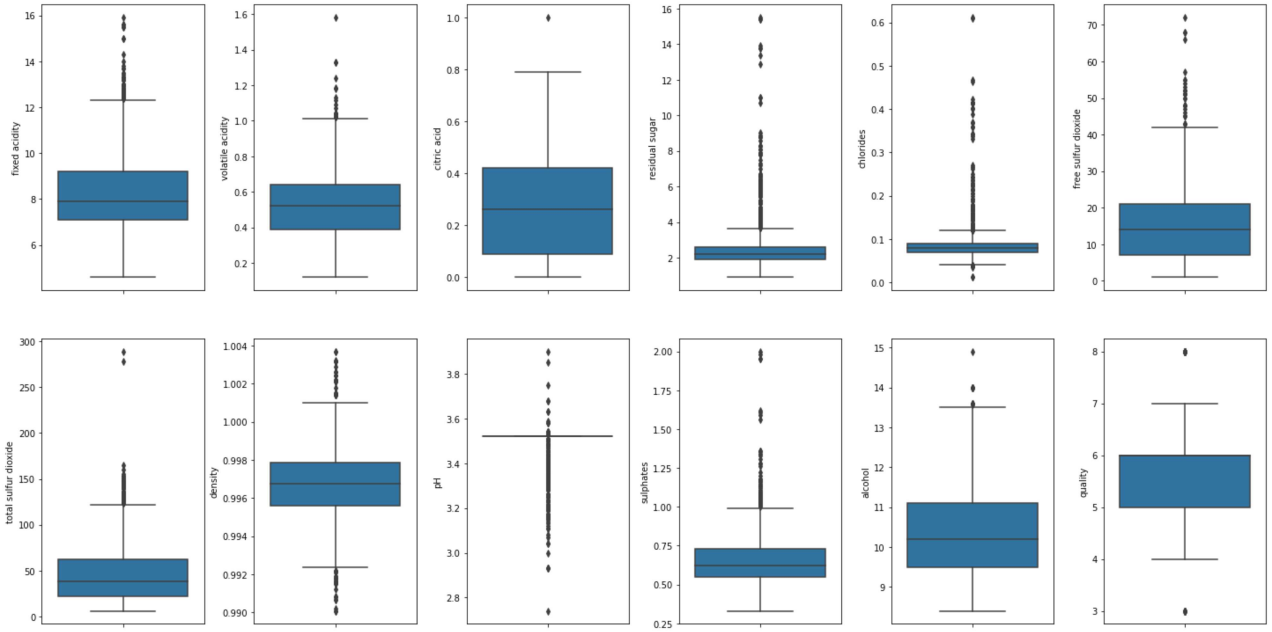
In [ ]:
```

```

Step 4: Handling outliers

In [20]:
```
fig, ax = plt.subplots(ncols=6, nrows=2, figsize=(20,10))
index = 0
ax = ax.flatten()

for col, value in wine_df.items():
        sns.boxplot(y=col, data=wine_df, ax=ax[index])
        #wine_df[col].plot(kind='box',ax=ax[index])
        index += 1
plt.tight_layout(pad=0.5, w_pad=0.7, h_pad=5.0)
```



In [21]:
```
wine_df[wine_df.columns].describe()
```

Out[21]:

	fixed acidity	volatile acidity	citric acid	residual sugar	chlorides	free sulfur dioxide	total sulfur dioxide	159
count	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	46.433041	
std	1.741096	0.179060	0.194801	1.409928	0.047065	10.460157	32.865961	
min	4.600000	0.120000	0.000000	0.900000	0.012000	1.000000	6.000000	
25%	7.100000	0.390000	0.090000	1.900000	0.070000	7.000000	22.000000	
50%	7.900000	0.520000	0.260000	2.200000	0.079000	14.000000	38.000000	
75%	9.200000	0.640000	0.420000	2.600000	0.090000	21.000000	62.000000	
max	15.900000	1.580000	1.000000	15.500000	0.611000	72.000000	289.000000	

In [22]:
```
def find_outlier_limits(col_name):
    Q1,Q3=wine_df[col_name].quantile([.25,.75])
    IQR=Q3-Q1
    low=Q1-(1.5* IQR)
    high=Q3+(1.5* IQR)
    return (high,low)
#Leaving pH  outliers as they are (it just takes 3.52
otherwise)
#Leaving quality outliers as they are as they are the target
```

```
for col,value in wine_df.items():
    if col!='pH' and col!='quality':
        high_hp,low_hp=find_outlier_limits(col)
        print('For ',col)
        print('High: ','upper limit: ',high_hp,' lower limit: ',low_hp)
        print('--------------------------------------------------')
        wine_df.loc[wine_df[col]>high_hp,col]=high_hp
        wine_df.loc[wine_df[col]<low_hp,col]=low_hp
```

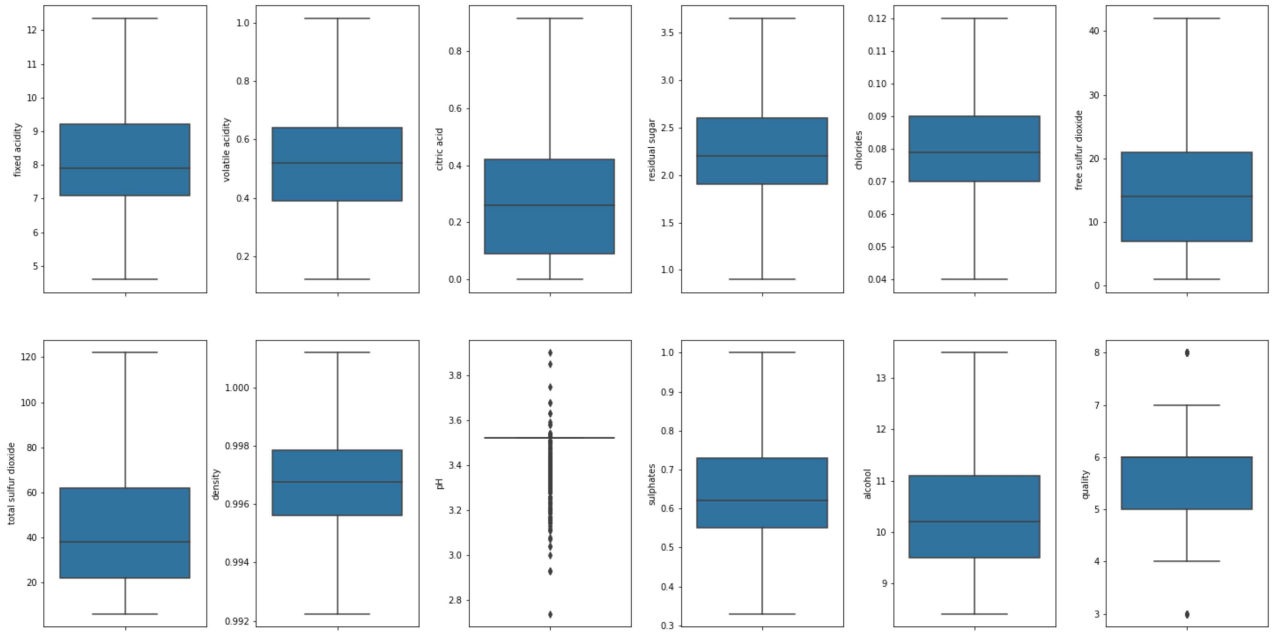
```
For  fixed acidity
High:  upper limit:  12.349999999999998  lower limit:  3.95
--------------------------------------------------
For  volatile acidity
High:  upper limit:  1.0150000000000001  lower limit:  0.015000000000000013
--------------------------------------------------
For  citric acid
High:  upper limit:  0.9149999999999999  lower limit:  -0.4049999999999999
--------------------------------------------------
For  residual sugar
High:  upper limit:  3.6500000000000004  lower limit:  0.8499999999999996
--------------------------------------------------
For  chlorides
High:  upper limit:  0.11999999999999998  lower limit:  0.04000000000000002
--------------------------------------------------
For  free sulfur dioxide
High:  upper limit:  42.0  lower limit:  -14.0
--------------------------------------------------
For  total sulfur dioxide
High:  upper limit:  122.0  lower limit:  -38.0
--------------------------------------------------
For  density
High:  upper limit:  1.0011875  lower limit:  0.9922475000000001
--------------------------------------------------
For  sulphates
High:  upper limit:  0.9999999999999999  lower limit:  0.28000000000000014
--------------------------------------------------
For  alcohol
High:  upper limit:  13.5  lower limit:  7.1000000000000005
--------------------------------------------------
```

In [23]:

```
fig, ax = plt.subplots(ncols=6, nrows=2, figsize=(20,10))
index = 0
ax = ax.flatten()

for col, value in wine_df.items():
```

```
        sns.boxplot(y=col, data=wine_df, ax=ax[index])
        #wine_df[col].plot(kind='box',ax=ax[index])
        index += 1
plt.tight_layout(pad=0.5, w_pad=0.7, h_pad=5.0)
```



Data Understanding

In [24]:

```
col_values = []
for col,values in wine_df.items():
    col_values.append(col)
correlation_df=wine_df[col_values].corr()
correlation_df
```

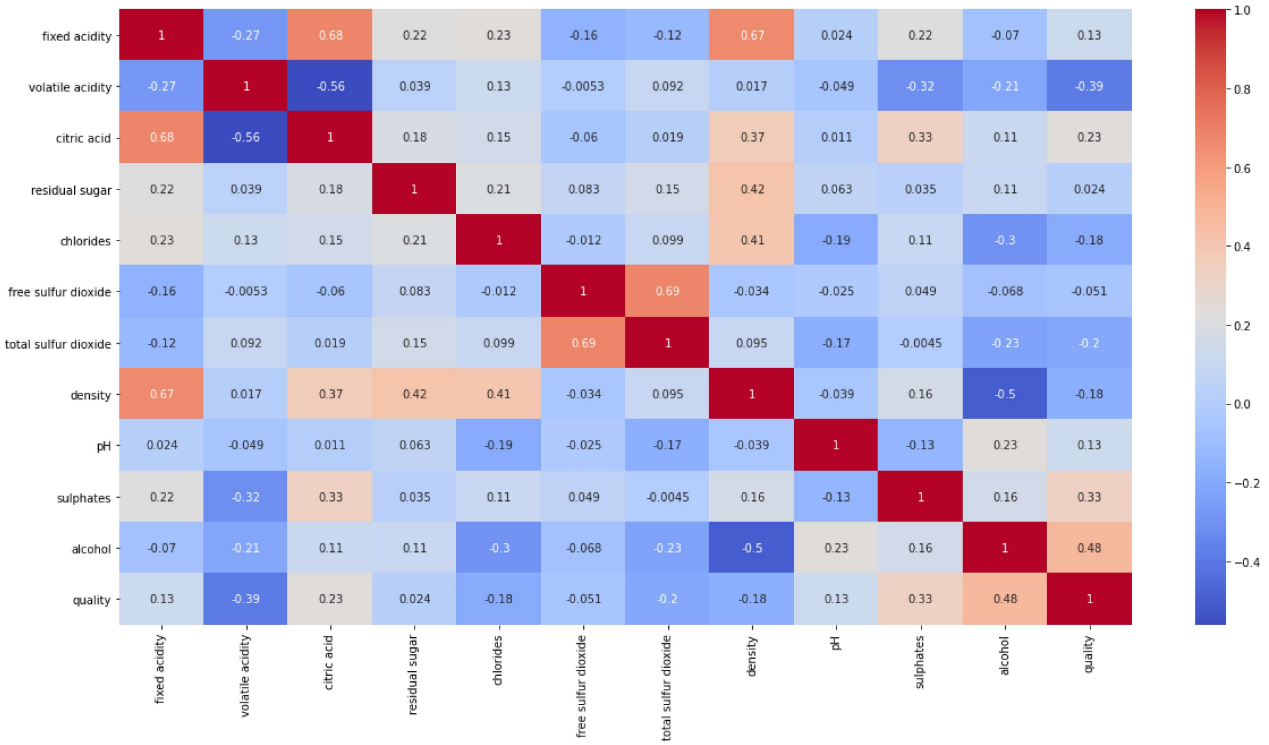
Out[24]:

	fixed acidity	volatile acidity	citric acid	residual sugar	chlorides	free sulfur dioxide	total sulfur dioxide	density	pH
fixed acidity	1.000000	-0.268153	0.678704	0.215090	0.228484	-0.157083	-0.119067	0.668076	0.02433
volatile acidity	-0.268153	1.000000	-0.560770	0.039427	0.133096	-0.005288	0.091856	0.017347	-0.04941
citric acid	0.678704	-0.560770	1.000000	0.183553	0.147668	-0.060140	0.018774	0.369893	0.01100
residual sugar	0.215090	0.039427	0.183553	1.000000	0.208471	0.082933	0.154923	0.424354	0.06293
chlorides	0.228484	0.133096	0.147668	0.208471	1.000000	-0.012169	0.098815	0.407441	-0.19222

	fixed acidity	volatile acidity	citric acid	residual sugar	chlorides	free sulfur dioxide	total sulfur dioxide	density	pH
free sulfur dioxide	-0.157083	-0.005288	-0.060140	0.082933	-0.012169	1.000000	0.687772	-0.034012	-0.02503
total sulfur dioxide	-0.119067	0.091856	0.018774	0.154923	0.098815	0.687772	1.000000	0.095303	-0.16503
density	0.668076	0.017347	0.369893	0.424354	0.407441	-0.034012	0.095303	1.000000	-0.03937
pH	0.024333	-0.049418	0.011004	0.062931	-0.192226	-0.025036	-0.165030	-0.039377	1.00000
sulphates	0.215195	-0.316181	0.333402	0.034996	0.107645	0.049086	-0.004488	0.161310	-0.13441
alcohol	-0.070242	-0.209385	0.111640	0.107114	-0.295608	-0.068099	-0.229194	-0.500237	0.23377
quality	0.125380	-0.387163	0.226502	0.024020	-0.183105	-0.050886	-0.201086	-0.176130	0.13393

In [25]:
```
#pH has very identical values
import seaborn as sns
plt.figure(figsize=(20,10))
sns.heatmap(correlation_df, cmap="coolwarm", annot=True)
```

Out[25]: <AxesSubplot:>



ML Model Building

In [26]:
```
x = wine_df.drop(columns=['quality'])
y = wine_df['quality']
print(x)
print(y)
```

```
      fixed acidity  volatile acidity  citric acid  residual sugar  chlorides  \
0               7.4             0.700         0.00             1.9      0.076
1               7.8             0.880         0.00             2.6      0.098
2               7.8             0.760         0.04             2.3      0.092
3              11.2             0.280         0.56             1.9      0.075
4               7.4             0.700         0.00             1.9      0.076
...             ...               ...          ...             ...        ...
1594            6.2             0.600         0.08             2.0      0.090
1595            5.9             0.550         0.10             2.2      0.062
1596            6.3             0.510         0.13             2.3      0.076
1597            5.9             0.645         0.12             2.0      0.075
1598            6.0             0.310         0.47             3.6      0.067

      free sulfur dioxide  total sulfur dioxide  density    pH  sulphates  \
0                    11.0                  34.0  0.99780  3.51       0.56
1                    25.0                  67.0  0.99680  3.20       0.68
2                    15.0                  54.0  0.99700  3.26       0.65
3                    17.0                  60.0  0.99800  3.16       0.58
4                    11.0                  34.0  0.99780  3.51       0.56
...                   ...                   ...      ...   ...        ...
1594                 32.0                  44.0  0.99490  3.52       0.58
1595                 39.0                  51.0  0.99512  3.52       0.76
1596                 29.0                  40.0  0.99574  3.52       0.75
1597                 32.0                  44.0  0.99547  3.52       0.71
1598                 18.0                  42.0  0.99549  3.52       0.66

      alcohol
0         9.4
1         9.8
2         9.8
3         9.8
4         9.4
...       ...
1594     10.5
1595     11.2
1596     11.0
1597     10.2
1598     11.0

[1599 rows x 11 columns]
0       5.0
1       5.0
2       5.0
3       6.0
4       5.0
       ...
1594    5.0
1595    6.0
1596    6.0
1597    5.0
1598    6.0
Name: quality, Length: 1599, dtype: float64
```

In [27]:
```
from sklearn.model_selection import train_test_split
```

```
X_train, X_test, y_train, y_test = train_test_split(x, y,
test_size = 0.2, random_state = 0)
```

In [28]:
```
from sklearn.preprocessing import StandardScaler
scale = StandardScaler()
scale.fit_transform(X_train)
scale.transform(X_test);
```

Training the model

In [29]:
```
from sklearn.linear_model import LinearRegression
reg = LinearRegression()
```

In [30]:
```
reg.fit(X_train, y_train)
```

Out[30]:
```
LinearRegression()
```

In [31]:
```
m = reg.coef_
c = reg.intercept_
m,c
```

Out[31]:
```
(array([ 8.06853464e-02, -1.12791423e+00, -3.18560663e-01,  5.29829577e-02,
        -2.53993444e+00,  1.50635712e-03, -2.00292281e-03, -5.56148484e+01,
         5.18064838e-01,  1.38420764e+00,  2.35686053e-01]),
 56.07953442350708)
```

In [32]:
```
y_pred_train = reg.predict(X_train)
```

In [33]:
```
y_pred_test = reg.predict(X_test)
```

In [34]:
```
from sklearn.metrics import r2_score
r2_S = r2_score(y_train, y_pred_train)
r2_S
```

Out[34]:
```
0.3741506138252614
```

In [35]:
```
from sklearn.metrics import r2_score
```

```
r2_S = r2_score(y_test, y_pred_test)
r2_S
```

Out[35]: 0.3340324134328466

In [47]:
```
# quality of wine depends more on the alcohol

wine={ 'fixed acidity':[8.0],'volatile acidity':[0.6],'citric
acid':[0.5],'residual sugar':[2.8],
       'chlorides':[0.025],'free sulfur dioxide':[10.0],'total
sulfur dioxide':[30.0],
       'density':[0.995],'pH':[3.25],'sulphates':
[0.6],'alcohol':[10.0]}

print(reg.predict(pd.DataFrame(wine)))
```

```
[5.46313983]
```

In [48]:
```
wine={ 'fixed acidity':[8.0],'volatile acidity':[0.6],'citric
acid':[0.5],'residual sugar':[2.8],
       'chlorides':[0.025],'free sulfur dioxide':[10.0],'total
sulfur dioxide':[30.0],
       'density':[0.995],'pH':[3.25],'sulphates':
[0.6],'alcohol':[20.0]}

print(reg.predict(pd.DataFrame(wine)))
```

```
[7.82000036]
```

In [ ]:
```

```