Import Required Packages

In [4]:
```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
```

In [6]:
```
import sklearn
```

In [8]:
```
from sklearn.datasets import load_wine
data=load_wine()
data
```

```
Out[8]: {'data': array([[1.423e+01, 1.710e+00, 2.430e+00, ..., 1.040e+00, 3.920e+00,
                 1.065e+03],
                [1.320e+01, 1.780e+00, 2.140e+00, ..., 1.050e+00, 3.400e+00,
                 1.050e+03],
                [1.316e+01, 2.360e+00, 2.670e+00, ..., 1.030e+00, 3.170e+00,
                 1.185e+03],
                ...,
                [1.327e+01, 4.280e+00, 2.260e+00, ..., 5.900e-01, 1.560e+00,
                 8.350e+02],
                [1.317e+01, 2.590e+00, 2.370e+00, ..., 6.000e-01, 1.620e+00,
                 8.400e+02],
                [1.413e+01, 4.100e+00, 2.740e+00, ..., 6.100e-01, 1.600e+00,
                 5.600e+02]]),
         'target': array([0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
        0, 0,
                0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
                0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 1, 1, 1, 1, 1, 1,
                1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,
                1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,
                1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 2, 2,
                2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2,
                2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2,
                2, 2]),
         'frame': None,
         'target_names': array(['class_0', 'class_1', 'class_2'], dtype='<U7'),
         'DESCR': '.. _wine_dataset:\n\nWine recognition dataset\
        n------------------------\n\n**Data Set Characteristics:**\n\n:Number of Inst
        ances: 178\n:Number of Attributes: 13 numeric, predictive attributes and the
        class\n:Attribute Information:\n    - Alcohol\n    - Malic acid\n    - Ash\n
        - Alcalinity of ash\n    - Magnesium\n    - Total phenols\n    - Flavanoids\n
        - Nonflavanoid phenols\n    - Proanthocyanins\n    - Color intensity\n    - H
        ue\n    - OD280/OD315 of diluted wines\n    - Proline\n    - class:\n
        - class_0\n        - class_1\n        - class_2\n\n:Summary Statistics:\n\
        n============================= ==== ===== ======= =====\n
        Min   Max   Mean     SD\n============================= ==== ===== =======
        =====\nAlcohol:                      11.0  14.8    13.0   0.8\nMalic Acid:
        0.74  5.80    2.34  1.12\nAsh:                          1.36  3.23    2.36
        0.27\nAlcalinity of Ash:           10.6  30.0    19.5   3.3\nMagnesium:
        70.0 162.0    99.7  14.3\nTotal Phenols:                0.98  3.88    2.29
        0.63\nFlavanoids:                  0.34  5.08    2.03  1.00\nNonflavanoid Ph
        enols:        0.13  0.66    0.36  0.12\nProanthocyanins:              0.41
        3.58    1.59  0.57\nColour Intensity:             1.3  13.0     5.1   2.3\nH
        ue:                          0.48  1.71    0.96  0.23\nOD280/OD315 of diluted
        wines: 1.27  4.00    2.61  0.71\nProline:                      278  1680
        746   315\n============================= ==== ===== ======= =====\n\n:Missing
        Attribute Values: None\n:Class Distribution: class_0 (59), class_1 (71), clas
        s_2 (48)\n:Creator: R.A. Fisher\n:Donor: Michael Marshall (MARSHALL%PLU@io.ar
        c.nasa.gov)\n:Date: July, 1988\n\nThis is a copy of UCI ML Wine recognition d
        atasets.\nhttps://archive.ics.uci.edu/ml/machine-learning-databases/wine/win
        e.data\n\nThe data is the results of a chemical analysis of wines grown in th
        e same\nregion in Italy by three different cultivators. There are thirteen di
        fferent\nmeasurements taken for different constituents found in the three typ
        es of\nwine.\n\nOriginal Owners:\n\nForina, M. et al, PARVUS -\nAn Extendible
        Package for Data Exploration, Classification and Correlation.\nInstitute of P
```

```
harmaceutical and Food Analysis and Technologies,\nVia Brigata Salerno, 16147
Genoa, Italy.\n\nCitation:\n\nLichman, M. (2013). UCI Machine Learning Reposi
tory\n[https://archive.ics.uci.edu/ml]. Irvine, CA: University of Californi
a,\nSchool of Information and Computer Science.\n\n.. dropdown:: References\
n\n    (1) S. Aeberhard, D. Coomans and O. de Vel,\n    Comparison of Classif
iers in High Dimensional Settings,\n    Tech. Rep. no. 92-02, (1992), Dept. o
f Computer Science and Dept. of\n    Mathematics and Statistics, James Cook U
niversity of North Queensland.\n    (Also submitted to Technometrics).\n\n
The data was used with many others for comparing various\n    classifiers. Th
e classes are separable, though only RDA\n    has achieved 100% correct class
ification.\n    (RDA : 100%, QDA 99.4%, LDA 98.9%, 1NN 96.1% (z-transformed d
ata))\n    (All results using the leave-one-out technique)\n\n    (2) S. Aebe
rhard, D. Coomans and O. de Vel,\n    "THE CLASSIFICATION PERFORMANCE OF RD
A"\n    Tech. Rep. no. 92-01, (1992), Dept. of Computer Science and Dept. o
f\n    Mathematics and Statistics, James Cook University of North Queenslan
d.\n    (Also submitted to Journal of Chemometrics).\n',
 'feature_names': ['alcohol',
  'malic_acid',
  'ash',
  'alcalinity_of_ash',
  'magnesium',
  'total_phenols',
  'flavanoids',
  'nonflavanoid_phenols',
  'proanthocyanins',
  'color_intensity',
  'hue',
  'od280/od315_of_diluted_wines',
  'proline']}
```

In [34]:

```
input_data=data['data']
output_data=data['target']
input_cols=data['feature_names']
output_cols=data['target_names'][0]
df=pd.DataFrame(input_data,columns=input_cols)
df[output_cols]=output_data
df
```

Out[34]:

	alcohol	malic_acid	ash	alcalinity_of_ash	magnesium	total_phenols	flava
0	14.23	1.71	2.43	15.6	127.0	2.80	
1	13.20	1.78	2.14	11.2	100.0	2.65	
2	13.16	2.36	2.67	18.6	101.0	2.80	
3	14.37	1.95	2.50	16.8	113.0	3.85	
4	13.24	2.59	2.87	21.0	118.0	2.80	
...	...	...	...	...	...	...	
173	13.71	5.65	2.45	20.5	95.0	1.68	
174	13.40	3.91	2.48	23.0	102.0	1.80	
175	13.27	4.28	2.26	20.0	120.0	1.59	
176	13.17	2.59	2.37	20.0	120.0	1.65	
177	14.13	4.10	2.74	24.5	96.0	2.05	

178 rows × 14 columns

In [36]:
```
df.drop_duplicates()
```

Out[36]:

	alcohol	malic_acid	ash	alcalinity_of_ash	magnesium	total_phenols	flava
0	14.23	1.71	2.43	15.6	127.0	2.80	
1	13.20	1.78	2.14	11.2	100.0	2.65	
2	13.16	2.36	2.67	18.6	101.0	2.80	
3	14.37	1.95	2.50	16.8	113.0	3.85	
4	13.24	2.59	2.87	21.0	118.0	2.80	
...	...	...	...	...	...	...	
173	13.71	5.65	2.45	20.5	95.0	1.68	
174	13.40	3.91	2.48	23.0	102.0	1.80	
175	13.27	4.28	2.26	20.0	120.0	1.59	
176	13.17	2.59	2.37	20.0	120.0	1.65	
177	14.13	4.10	2.74	24.5	96.0	2.05	

178 rows × 14 columns

In [38]:
```
df.isnull().sum()
```

Out[38]:
```
alcohol                         0
malic_acid                      0
ash                             0
alcalinity_of_ash               0
magnesium                       0
total_phenols                   0
flavanoids                      0
nonflavanoid_phenols            0
proanthocyanins                 0
color_intensity                 0
hue                             0
od280/od315_of_diluted_wines    0
proline                         0
class_0                         0
dtype: int64
```

In [42]:
```
y=data.target
y
```

Out[42]:
```
array([0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
       0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
       0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 1, 1, 1, 1, 1, 1,
       1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,
       1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,
       1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 2, 2,
       2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2,
       2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2,
       2, 2])
```

In [48]:
```
data['target']
```

Out[48]:
```
array([0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
       0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
       0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 1, 1, 1, 1, 1, 1,
       1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,
       1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,
       1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 2, 2,
       2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2,
       2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2,
       2, 2])
```

In [56]:
```
df['class_0']
```

Out[56]:

```
0      0
1      0
2      0
3      0
4      0
      ..
173    2
174    2
175    2
176    2
177    2
Name: class_0, Length: 178, dtype: int32
```

In [58]:

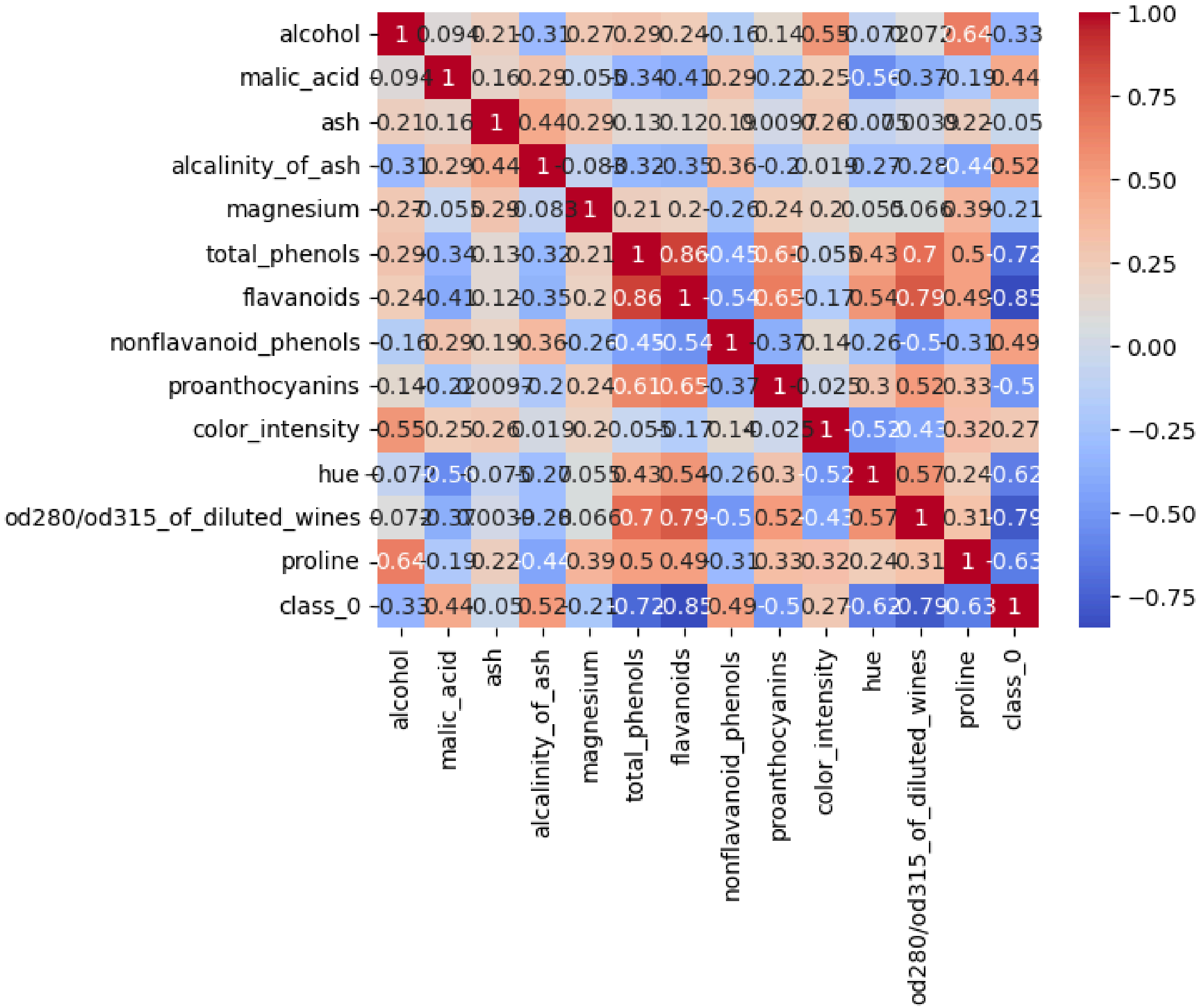
```
corre=df.corr(numeric_only=True)
corre
```

Out[58]:

	alcohol	malic_acid	ash	alcalinity_of_ash	magn
alcohol	1.000000	0.094397	0.211545	-0.310235	0.2
malic_acid	0.094397	1.000000	0.164045	0.288500	-0.0
ash	0.211545	0.164045	1.000000	0.443367	0.2
alcalinity_of_ash	-0.310235	0.288500	0.443367	1.000000	-0.0
magnesium	0.270798	-0.054575	0.286587	-0.083333	1.0
total_phenols	0.289101	-0.335167	0.128980	-0.321113	0.2
flavanoids	0.236815	-0.411007	0.115077	-0.351370	0.1
nonflavanoid_phenols	-0.155929	0.292977	0.186230	0.361922	-0.2
proanthocyanins	0.136698	-0.220746	0.009652	-0.197327	0.2
color_intensity	0.546364	0.248985	0.258887	0.018732	0.1
hue	-0.071747	-0.561296	-0.074667	-0.273955	0.0
od280/ od315_of_diluted_wines	0.072343	-0.368710	0.003911	-0.276769	0.0
proline	0.643720	-0.192011	0.223626	-0.440597	0.3
class_0	-0.328222	0.437776	-0.049643	0.517859	-0.2

In [60]:

```
sns.heatmap(corre,annot=True,cmap='coolwarm')
plt.show()
```



Short-Cut

In [62]:
```
from sklearn.datasets import load_wine
load_wine(as_frame=True).frame
```

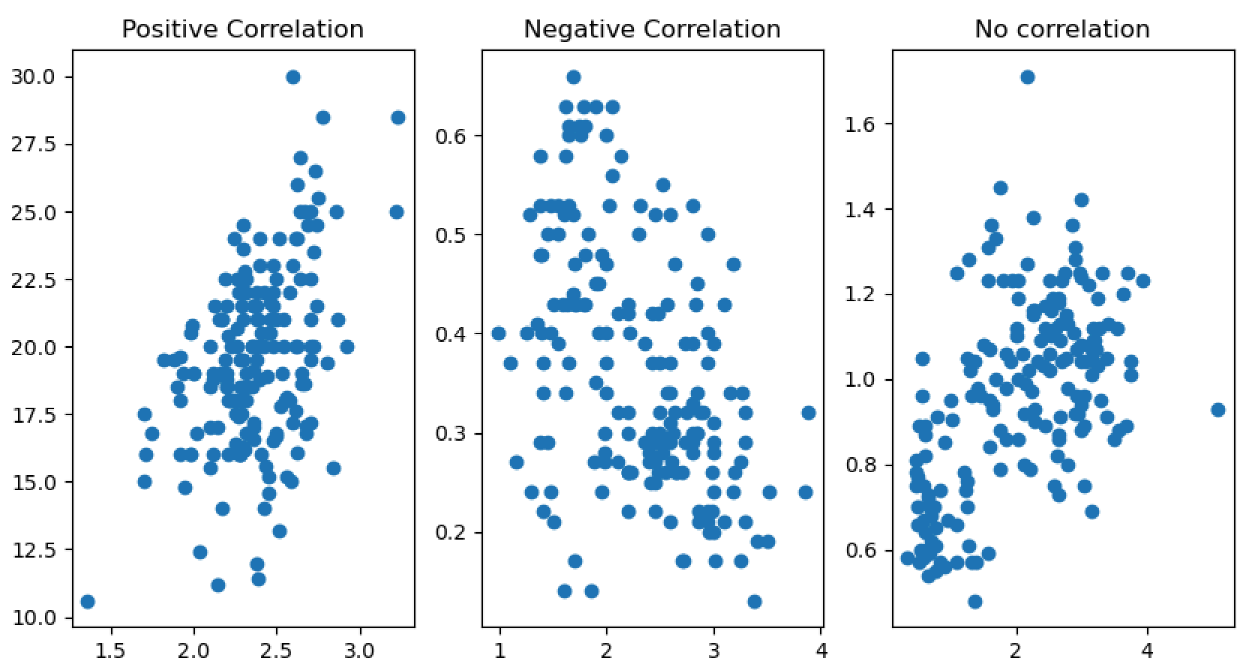
Out[62]:

	alcohol	malic_acid	ash	alcalinity_of_ash	magnesium	total_phenols	flav
0	14.23	1.71	2.43	15.6	127.0	2.80	
1	13.20	1.78	2.14	11.2	100.0	2.65	
2	13.16	2.36	2.67	18.6	101.0	2.80	
3	14.37	1.95	2.50	16.8	113.0	3.85	
4	13.24	2.59	2.87	21.0	118.0	2.80	
...	...	...	...	...	...	...	
173	13.71	5.65	2.45	20.5	95.0	1.68	
174	13.40	3.91	2.48	23.0	102.0	1.80	
175	13.27	4.28	2.26	20.0	120.0	1.59	
176	13.17	2.59	2.37	20.0	120.0	1.65	
177	14.13	4.10	2.74	24.5	96.0	2.05	

178 rows × 14 columns

In [89]:

```
plt.figure(figsize=(10,5))
plt.subplot(1,3,1).scatter(df['ash'],df['alcalinity_of_ash'])
plt.title("Positive Correlation")
plt.subplot(1,3,2).scatter(df['total_phenols'],df['nonflavanoid_phenols'])
plt.title("Negative Correlation")
plt.subplot(1,3,3).scatter(df['flavanoids'],df['hue'])
plt.title("No correlation")
plt.show()
```



In [93]:

```
from statsmodels.stats.outliers_influence import variance_inflation_factor
X=df.drop("class_0",axis=1)
vif_list=[]
for i in range(len(X.columns)):
    vif_list.append(variance_inflation_factor(X.values,i))
vif_list
```

Out[93]:

```
[206.1890565710355,
 8.925540511579005,
 165.64036999032706,
 73.14156355409301,
 67.36486845647852,
 62.78693524693161,
 35.53560246690394,
 16.63670778287286,
 17.115665485297978,
 17.022272420024827,
 45.39840748252447,
 54.539165172315194,
 16.37082766215551]
```

In [95]:

```
X.columns
```

Out[95]:

```
Index(['alcohol', 'malic_acid', 'ash', 'alcalinity_of_ash', 'magnesium',
       'total_phenols', 'flavanoids', 'nonflavanoid_phenols',
       'proanthocyanins', 'color_intensity', 'hue',
       'od280/od315_of_diluted_wines', 'proline'],
      dtype='object')
```

- 1. flavanoids
- 2. od280/od315_of_diluted_wines
- 3. color_intensity
- 4. alcohol
- 5. proline

In [101…

```
vif_data=pd.DataFrame(vif_list,index=X.columns,columns=['VIF'])
vif_data
```

Out[101…

	VIF
alcohol	206.189057
malic_acid	8.925541
ash	165.640370
alcalinity_of_ash	73.141564
magnesium	67.364868
total_phenols	62.786935
flavanoids	35.535602
nonflavanoid_phenols	16.636708
proanthocyanins	17.115665
color_intensity	17.022272
hue	45.398407
od280/od315_of_diluted_wines	54.539165
proline	16.370828

In [107…

```
vif_sorted1=vif_data.sort_values(by='VIF',ascending=True)
vif_sorted1
```

Out[107…

	VIF
malic_acid	8.925541
proline	16.370828
nonflavanoid_phenols	16.636708
color_intensity	17.022272
proanthocyanins	17.115665
flavanoids	35.535602
hue	45.398407
od280/od315_of_diluted_wines	54.539165
total_phenols	62.786935
magnesium	67.364868
alcalinity_of_ash	73.141564
ash	165.640370
alcohol	206.189057

In [109…

```
vif_sorted1[vif_sorted1['VIF']<36]
```

Out[109…

	VIF
malic_acid	8.925541
proline	16.370828
nonflavanoid_phenols	16.636708
color_intensity	17.022272
proanthocyanins	17.115665
flavanoids	35.535602

Select The features)

In [114…

```
final_cols=['flavanoids','malic_acid','color_intensity','alcohol','proline','c
final_df=df[final_cols]
final_df
```

Out[114…

	flavanoids	malic_acid	color_intensity	alcohol	proline	class_0
0	3.06	1.71	5.64	14.23	1065.0	0
1	2.76	1.78	4.38	13.20	1050.0	0
2	3.24	2.36	5.68	13.16	1185.0	0
3	3.49	1.95	7.80	14.37	1480.0	0
4	2.69	2.59	4.32	13.24	735.0	0
...	...	...	...	...	...	...
173	0.61	5.65	7.70	13.71	740.0	2
174	0.75	3.91	7.30	13.40	750.0	2
175	0.69	4.28	10.20	13.27	835.0	2
176	0.68	2.59	9.30	13.17	840.0	2
177	0.76	4.10	9.20	14.13	560.0	2

178 rows × 6 columns

In [120…

```
X=final_df.drop("class_0",axis=1)
y=final_df['class_0']

from sklearn.preprocessing import MinMaxScaler
mms=MinMaxScaler()
scaled_df=pd.DataFrame()
scaled_df[X.columns]=mms.fit_transform(X)
scaled_df['class_0']=y
scaled_df
```

Out[120…

	flavanoids	malic_acid	color_intensity	alcohol	proline	class_0
0	0.573840	0.191700	0.372014	0.842105	0.561341	0
1	0.510549	0.205534	0.264505	0.571053	0.550642	0
2	0.611814	0.320158	0.375427	0.560526	0.646933	0
3	0.664557	0.239130	0.556314	0.878947	0.857347	0
4	0.495781	0.365613	0.259386	0.581579	0.325963	0
...	...	...	...	...	...	...
173	0.056962	0.970356	0.547782	0.705263	0.329529	2
174	0.086498	0.626482	0.513652	0.623684	0.336662	2
175	0.073840	0.699605	0.761092	0.589474	0.397290	2
176	0.071730	0.365613	0.684300	0.563158	0.400856	2
177	0.088608	0.664032	0.675768	0.815789	0.201141	2

178 rows × 6 columns

Model Development

In [123…

```
X=scaled_df.drop("class_0",axis=1)
y=scaled_df['class_0']
```

In [133…

```
print(scaled_df.shape)
print(X.shape)
print(y.shape)
```

```
(178, 6)
(178, 5)
(178,)
```

In [139…

```
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,random_sta
```

In [141…

```
print(scaled_df.shape)
print(X_train.shape)
print(y_train.shape)
print(X_test.shape)
print(y_test.shape)
```

```
(178, 6)
(124, 5)
(124,)
(54, 5)
(54,)
```

In [143…

```
X_train
```

Out[143…

	flavanoids	malic_acid	color_intensity	alcohol	proline
138	0.029536	0.563241	0.377133	0.647368	0.215407
104	0.333333	0.195652	0.141638	0.389474	0.281027
78	0.318565	0.049407	0.180887	0.342105	0.336662
36	0.493671	0.177866	0.283276	0.592105	0.429387
93	0.402954	0.413043	0.074232	0.331579	0.008559
...	...	...	...	...	...
71	0.531646	0.152174	0.179181	0.744737	0.094151
106	0.356540	0.195652	0.180887	0.321053	0.165478
14	0.696203	0.223320	0.530717	0.881579	0.905136
92	0.236287	0.156126	0.151024	0.436842	0.154779
102	0.373418	0.337945	0.129693	0.344737	0.114123

124 rows × 5 columns

In [147…

```
y_train
```

Out[147…

```
138    2
104    1
78     1
36     0
93     1
      ..
71     1
106    1
14     0
92     1
102    1
Name: class_0, Length: 124, dtype: int32
```

In [149…

```
X_test
```

Out[149…

	flavanoids	malic_acid	color_intensity	alcohol	proline
19	0.567511	0.466403	0.325939	0.686842	0.404422
45	0.487342	0.652174	0.337884	0.836842	0.572040
140	0.033755	0.409091	0.283276	0.500000	0.229672
30	0.613924	0.150198	0.377133	0.710526	0.718260
67	0.350211	0.084980	0.290102	0.352632	0.165478
16	0.590717	0.233202	0.419795	0.860526	0.714693
119	0.274262	0.531621	0.000000	0.255263	0.203994
174	0.086498	0.626482	0.513652	0.623684	0.336662
109	0.544304	0.120553	0.116894	0.152632	0.286733
141	0.033755	0.359684	0.368601	0.613158	0.358060
24	0.478903	0.211462	0.191126	0.650000	0.404422
150	0.259494	0.470356	0.624573	0.650000	0.158345
41	0.493671	0.612648	0.255973	0.626316	0.539943
118	0.191983	0.531621	0.180887	0.457895	0.067047
15	0.542194	0.211462	0.513652	0.684211	0.736091
111	0.407173	0.333992	0.061433	0.392105	0.033524
113	0.352321	0.000000	0.153584	0.100000	0.111270
82	0.261603	0.077075	0.078498	0.276316	0.251070
9	0.592827	0.120553	0.506826	0.744737	0.547076
114	0.411392	0.128458	0.138225	0.276316	0.076320
18	0.757384	0.167984	0.633106	0.831579	1.000000
66	0.599156	0.053360	0.343003	0.547368	0.159772
60	0.158228	0.071146	0.169795	0.342105	0.286733
169	0.130802	0.762846	0.616041	0.623684	0.251070
171	0.035865	0.326087	0.735495	0.457895	0.136947
164	0.071730	0.399209	0.708191	0.723684	0.240371
117	0.369198	0.171937	0.066553	0.365789	0.047789
65	0.487342	0.092885	0.283276	0.352632	0.285307
90	0.244726	0.215415	0.095563	0.276316	0.144080
55	0.514768	0.195652	0.424061	0.665789	0.600571
29	0.419831	0.185771	0.291809	0.786842	0.539943

	flavanoids	malic_acid	color_intensity	alcohol	proline
128	0.445148	0.175889	0.071672	0.352632	0.045649
145	0.044304	0.559289	0.232082	0.560526	0.393723
31	0.601266	0.181818	0.479522	0.671053	0.882311
12	0.510549	0.195652	0.368601	0.715789	0.743224
42	0.679325	0.227273	0.354096	0.750000	0.582739
158	0.204641	0.185771	1.000000	0.871053	0.272468
137	0.054852	0.942688	0.317406	0.394737	0.169044
98	0.719409	0.065217	0.274744	0.352632	0.272468
159	0.160338	0.183794	0.893345	0.644737	0.243937
38	0.485232	0.150198	0.206485	0.536842	0.529244
108	0.358650	0.108696	0.121160	0.313158	0.024251
85	0.337553	0.047431	0.114334	0.431579	0.122682
68	0.202532	0.039526	0.161263	0.607895	0.336662
143	0.097046	0.832016	0.266212	0.681579	0.194009
2	0.611814	0.320158	0.375427	0.560526	0.646933
100	0.386076	0.264822	0.172355	0.276316	0.308131
122	0.377637	0.729249	0.068259	0.365789	0.062054
154	0.050633	0.108696	0.539249	0.407895	0.258203
51	0.559072	0.179842	0.368601	0.736842	0.703994
76	0.356540	0.031621	0.283276	0.526316	0.081312
56	0.561181	0.189723	0.435154	0.839474	0.493581
26	0.548523	0.203557	0.300341	0.621053	0.654066
153	0.103376	0.505929	0.788396	0.578947	0.283167

In [151…

```
y_test
```

Out[151…

19	0
45	0
140	2
30	0
67	1
16	0
119	1
174	2
109	1
141	2
24	0
150	2
41	0
118	1
15	0
111	1
113	1
82	1
9	0
114	1
18	0
66	1
60	1
169	2
171	2
164	2
117	1
65	1
90	1
55	0
29	0
128	1
145	2
31	0
12	0
42	0
158	2
137	2
98	1
159	2
38	0
108	1
85	1
68	1
143	2
2	0
100	1
122	1
154	2
51	0
76	1
56	0
26	0
153	2

```
Name: class_0, dtype: int32
```

Linear Regression

In [156…
```python
from sklearn.linear_model import LinearRegression
lr=LinearRegression()
lr.fit(X_train,y_train)
```

Out[156…
```
▾ LinearRegression
LinearRegression()
```

Model_prediciton``

In [159…
```python
y_pred=lr.predict(X_test)
y_pred
```

Out[159…
```
array([ 0.37403152,  0.29401935,  1.68070343, -0.13307837,  1.15568096,
       -0.08788719,  0.99804621,  1.75937097,  0.49134058,  1.56398371,
        0.30987212,  1.75407012,  0.31609603,  1.48053226,  0.2161892 ,
        0.91412681,  1.14766025,  0.95404212,  0.28890117,  0.98135694,
       -0.46534745,  0.62803088,  1.20611347,  1.96456159,  2.46165567,
        2.08150122,  0.95388282,  0.72511174,  1.1823732 ,  0.31812313,
        0.31490204,  0.82318408,  1.37770901, -0.13570389,  0.03440839,
       -0.12757274,  2.06765879,  1.95927474,  0.2649601 ,  2.16717327,
        0.2125327 ,  1.09996965,  0.92181186,  0.87910929,  1.57383328,
        0.08908901,  0.82004688,  1.05870024,  1.96657152, -0.02754054,
        1.12518831,  0.27562066,  0.03043692,  2.19349444])
```

In [169…
```python
import warnings
warnings.filterwarnings('ignore')
lr.predict([X_test.values[0],X_test.values[1]])
```

Out[169…
```
array([0.37403152, 0.29401935])
```

In [171…
```python
y_pred
```

Out[171…
```
array([ 0.37403152,  0.29401935,  1.68070343, -0.13307837,  1.15568096,
       -0.08788719,  0.99804621,  1.75937097,  0.49134058,  1.56398371,
        0.30987212,  1.75407012,  0.31609603,  1.48053226,  0.2161892 ,
        0.91412681,  1.14766025,  0.95404212,  0.28890117,  0.98135694,
       -0.46534745,  0.62803088,  1.20611347,  1.96456159,  2.46165567,
        2.08150122,  0.95388282,  0.72511174,  1.1823732 ,  0.31812313,
        0.31490204,  0.82318408,  1.37770901, -0.13570389,  0.03440839,
       -0.12757274,  2.06765879,  1.95927474,  0.2649601 ,  2.16717327,
        0.2125327 ,  1.09996965,  0.92181186,  0.87910929,  1.57383328,
        0.08908901,  0.82004688,  1.05870024,  1.96657152, -0.02754054,
        1.12518831,  0.27562066,  0.03043692,  2.19349444])
```

In [187…
```python
new_df=pd.DataFrame(X_test)
new_df['y_actual']=y_test
```

```
new_df['y_predicted']=y_pred
new_df['Error']=y_test-y_pred
new_df['square Error']=np.square(y_test-y_pred)
total_error=np.sum(np.square(y_test-y_pred))
total_error
```

Out[187… 3.925628324409452

In [189…
```
mean_squared_error=total_error/len(X_test)
mean_squared_error
```

Out[189… 0.07269682082239726

Mean Squared Error Method

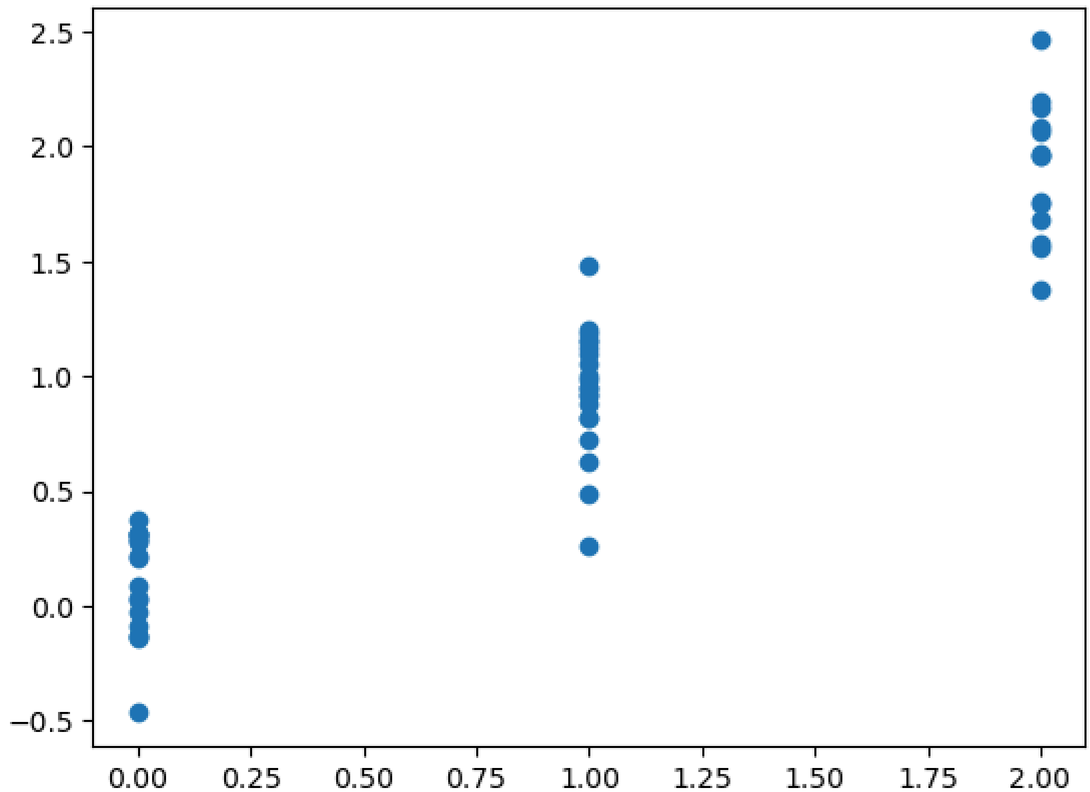
In [193…
```
from sklearn.metrics import mean_squared_error
mean_squared_error(y_test,y_pred)
```

Out[193… 0.07269682082239726

In [196…
```
rmse=np.sqrt(mean_squared_error(y_test,y_pred))
rmse
```

Out[196… 0.2696234797312676

In [198…
```
y_pred
errors=y_test-y_pred
plt.scatter(y_test,y_pred)
plt.show()
```



Model Deployment

- Using pickle

In [201…

```
import pickle
with open("lr_wine_model.pkl",'wb') as file:
    pickle.dump(lr,file)
```

In [215…

```
import pickle
with open("lr_wine_model.pkl",'rb') as file:
    model1=pickle.load(file)
model1
```

Out[215…

```
▾ LinearRegression
LinearRegression()
```

- using joblib

In [229…

```
import joblib
with open("lr_wine_model.joblib",'wb') as file:
    joblib.dump(lr,file)
```

In [231…

```
import joblib
with open("lr_wine_model.joblib",'rb') as file:
    model2=joblib.load(file)
model2
```

Out[231…

```
▾ LinearRegression
LinearRegression()
```

In [233…

```
model1.predict([[1,2,3,4,5]])
```

Out[233…

```
array([-3.96822747])
```

In [223…

```
model2.predict([[1,2,3,4,5]])
```

Out[223…

```
array([-3.96822747])
```

In [225…

```
model2.feature_names_in_
```

Out[225…

```
array(['flavanoids', 'malic_acid', 'color_intensity', 'alcohol',
       'proline'], dtype=object)
```

In [ ]: