Importing basic libraries

In [1]:
```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
%matplotlib inline
import seaborn as sns
import warnings
warnings.filterwarnings('ignore')
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import StandardScaler
from sklearn.metrics import mean_squared_error
from sklearn.metrics import mean_absolute_error, r2_score
```

```
<frozen importlib._bootstrap>:228: RuntimeWarning: scipy._lib.messagestream.MessageStream size changed, may indicate bina
ry incompatibility. Expected 56 from C header, got 64 from PyObject
```

In [2]:
```
# Read the data

df = pd.read_csv('https://raw.githubusercontent.com/subhashdixit/Support_Vector_Machines/main/SVC/Red_Wine_Quality/winequ
```

In [3]:
```
df.head()
```

Out[3]:

	Unnamed: 0	fixed acidity	volatile acidity	citric acid	residual sugar	chlorides	free sulfur dioxide	total sulfur dioxide	density	pH	sulphates	alcohol	quality
0	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	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	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	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	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 [4]:
```
df.drop(['Unnamed: 0'],axis=1,inplace=True)
df.head()
```

Out[4]:

	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

In [5]:
```
# Shape of the data

df.shape
```

Out[5]:
```
(1599, 12)
```

In [6]:
```
# Uniques values in target

df.quality.unique()
```

Out[6]:
```
array([5, 6, 7, 4, 8, 3], dtype=int64)
```

In [7]:
```
# Number of values in each category

df['quality'].value_counts()
```

Out[7]:
```
5    681
6    638
7    199
4     53
8     18
3     10
Name: quality, dtype: int64
```

Observation:- Highly Imbalanced data.

In [8]:
```
df.columns
```

Out[8]:
```
Index(['fixed acidity', 'volatile acidity', 'citric acid', 'residual sugar',
       'chlorides', 'free sulfur dioxide', 'total sulfur dioxide', 'density',
       'pH', 'sulphates', 'alcohol', 'quality'],
      dtype='object')
```

In [9]:
```
# Information about the data

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
```

In [10]:
```
# Statistical Analysis

df.describe().T
```

Out[10]:

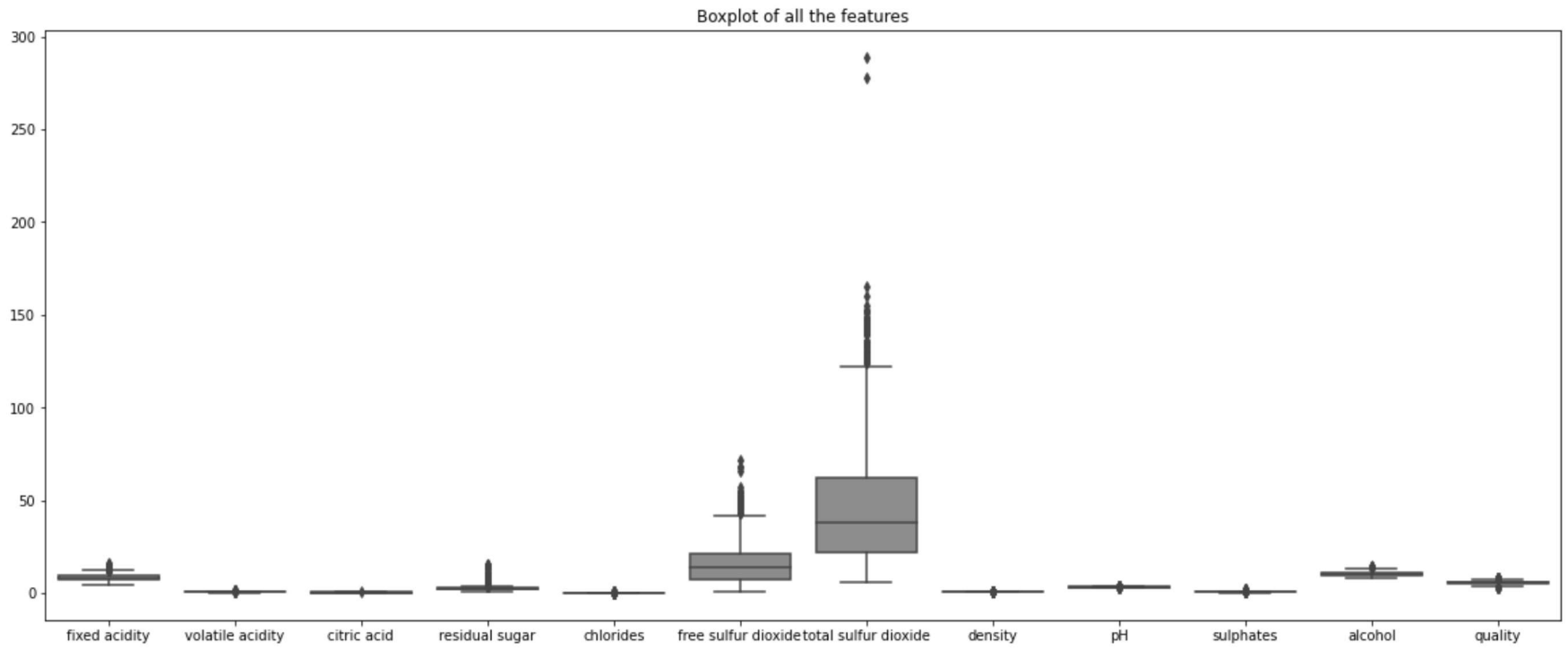
	count	mean	std	min	25%	50%	75%	max
fixed acidity	1599.0	8.319637	1.741096	4.60000	7.1000	7.90000	9.200000	15.90000
volatile acidity	1599.0	0.527821	0.179060	0.12000	0.3900	0.52000	0.640000	1.58000
citric acid	1599.0	0.270976	0.194801	0.00000	0.0900	0.26000	0.420000	1.00000
residual sugar	1599.0	2.538806	1.409928	0.90000	1.9000	2.20000	2.600000	15.50000
chlorides	1599.0	0.087467	0.047065	0.01200	0.0700	0.07900	0.090000	0.61100
free sulfur dioxide	1599.0	15.874922	10.460157	1.00000	7.0000	14.00000	21.000000	72.00000
total sulfur dioxide	1599.0	46.467792	32.895324	6.00000	22.0000	38.00000	62.000000	289.00000
density	1599.0	0.996747	0.001887	0.99007	0.9956	0.99675	0.997835	1.00369
pH	1599.0	3.311113	0.154386	2.74000	3.2100	3.31000	3.400000	4.01000
sulphates	1599.0	0.658149	0.169507	0.33000	0.5500	0.62000	0.730000	2.00000
alcohol	1599.0	10.422983	1.065668	8.40000	9.5000	10.20000	11.100000	14.90000
quality	1599.0	5.636023	0.807569	3.00000	5.0000	6.00000	6.000000	8.00000

In [11]:
```
# Check Null values in the dataset

df.isnull().sum()
```

Out[11]:
```
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 [12]:
```
# Checking for Outliers

plt.figure(figsize=(20,8))
sns.boxplot(data=df)
plt.title("Boxplot of all the features")
```

Out[12]:
```
Text(0.5, 1.0, 'Boxplot of all the features')
```



In [13]:
```
# Handling outliers

# Creating a function which will return upper and lower limit for outs
def find_boundaries(df, var, distance):
    iqr = df[var].quantile(0.75) - df[var].quantile(0.25)
    low_bound = df[var].quantile(0.25) - (iqr*distance)
    upp_bound = df[var].quantile(0.75) + (iqr*distance)
    return upp_bound, low_bound
```

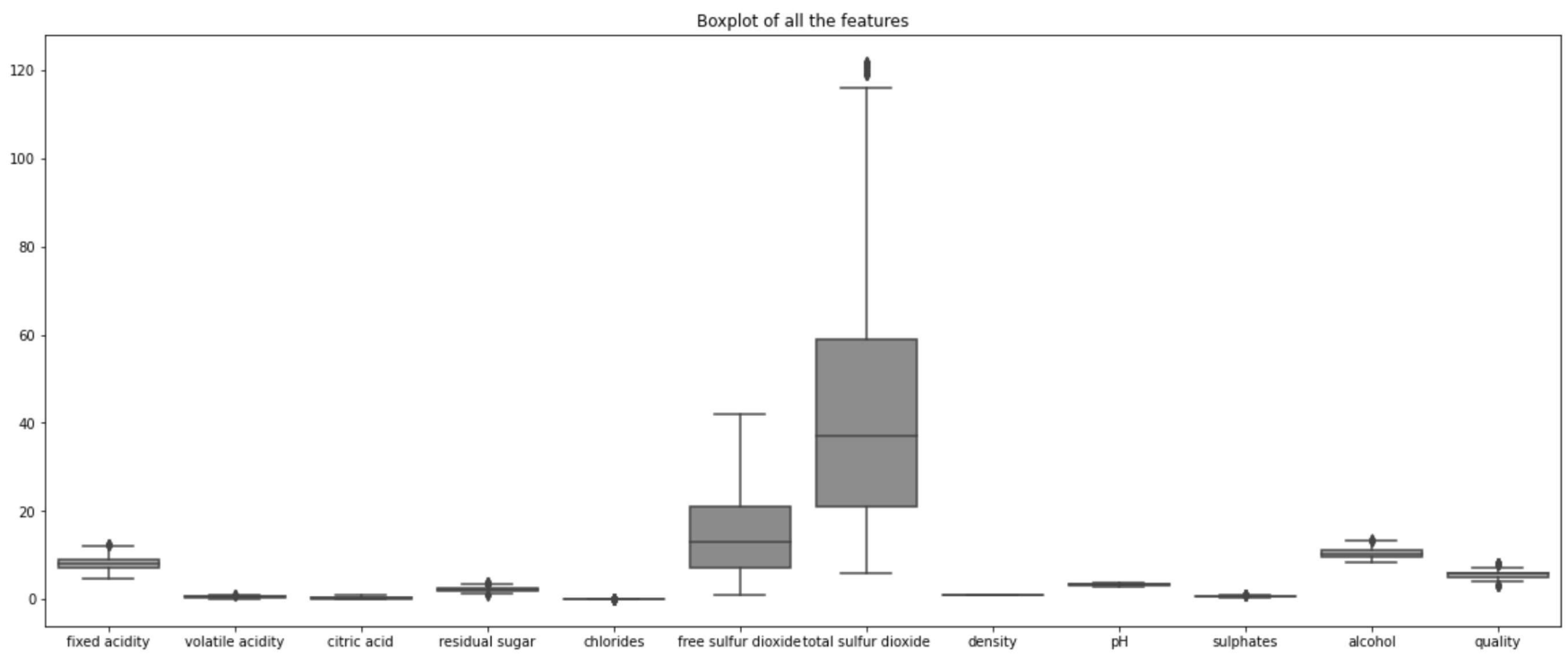
In [14]:
```
df.columns
```

Out[14]:
```
Index(['fixed acidity', 'volatile acidity', 'citric acid', 'residual sugar',
       'chlorides', 'free sulfur dioxide', 'total sulfur dioxide', 'density',
       'pH', 'sulphates', 'alcohol', 'quality'],
      dtype='object')
```

In [15]:
```
outliers_columns = ['fixed acidity', 'volatile acidity', 'citric acid', 'residual sugar',
       'chlorides', 'free sulfur dioxide', 'total sulfur dioxide', 'density',
       'pH', 'sulphates', 'alcohol']

# Removing Outliers of the dataset
for i in outliers_columns:
    upper_boundary, lower_boundary = find_boundaries(df,i,1.5)
    outliers = np.where(df[i]>upper_boundary, True, np.where(df[i]<lower_boundary,True, False))
    outliers_df = df.loc[outliers,i]
    df_trimed = df.loc[~outliers,i]
    df[i] = df_trimed
```

In [16]:
```
# Boxplot after removing the outlier

plt.figure(figsize=(20,8))
sns.boxplot(data=df)
plt.title("Boxplot of all the features")
```

Out[16]:
```
Text(0.5, 1.0, 'Boxplot of all the features')
```



In [17]:
```
for x in outliers_columns:
    df[x] = df[x].astype('float32')
```

Graphical Analysis

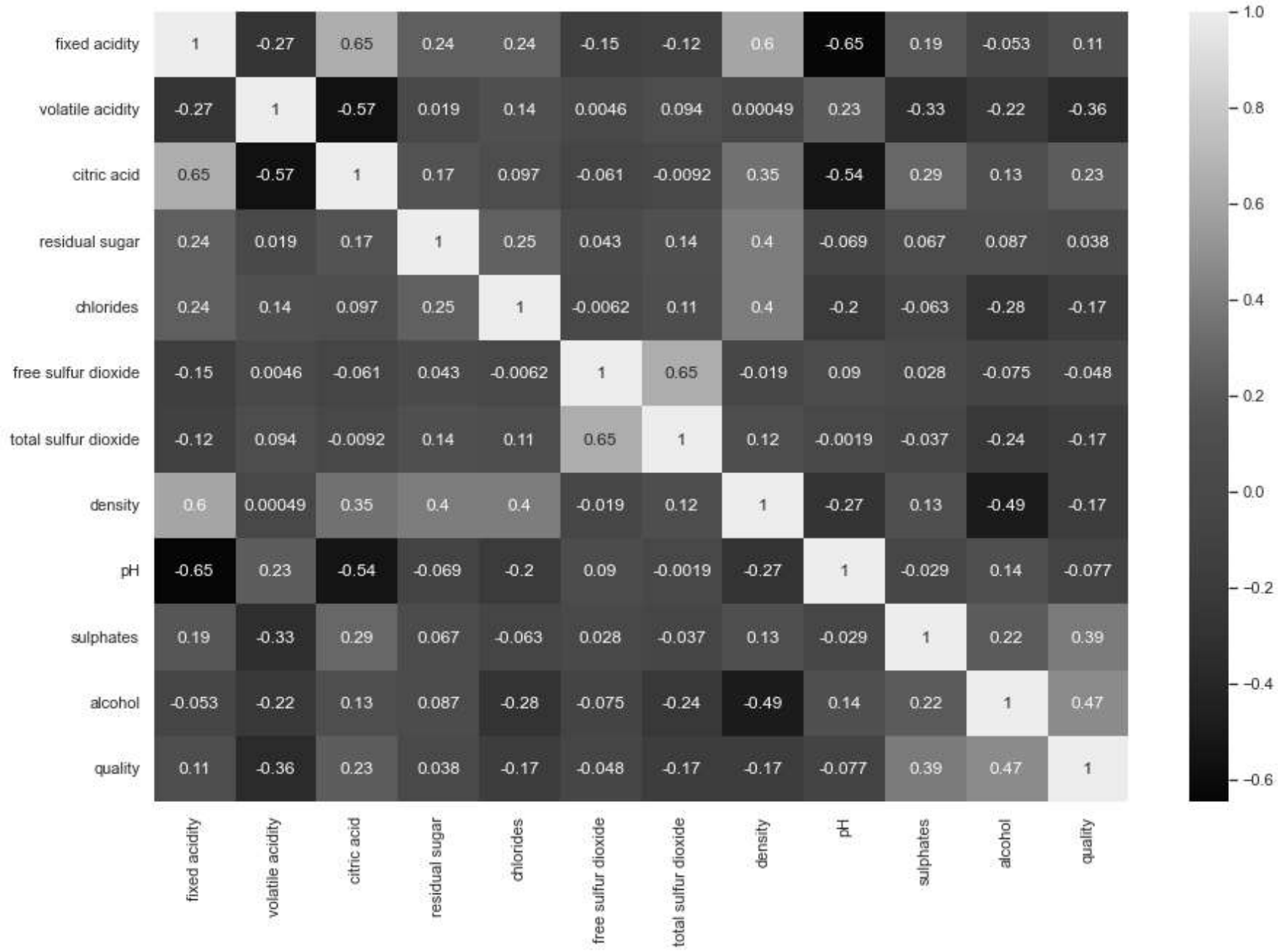
In [18]:
```
df.columns
```

Out[18]:
```
Index(['fixed acidity', 'volatile acidity', 'citric acid', 'residual sugar',
       'chlorides', 'free sulfur dioxide', 'total sulfur dioxide', 'density',
       'pH', 'sulphates', 'alcohol', 'quality'],
      dtype='object')
```

In [19]:
```
df['quality'] = df['quality'].astype('int')
```

In [20]:
```
sns.set(rc={'figure.figsize':(15,10)})
sns.heatmap(data=df.corr(),annot=True)
```

Out[20]:
```
<AxesSubplot:>
```



Segregate independent and dependent variable

In [21]:
```
X = df.drop('quality',axis=1)
y = df['quality']
```

In [22]:
```
# Train Test split

from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split(X,y, test_size=0.33, random_state=42)
```

In [23]:
```
print(X_train.shape)
print(X_test.shape)
```

```
(1071, 11)
(528, 11)
```

In [24]:
```
print(y_train.shape)
print(y_test.shape)
```

```
(1071,)
(528,)
```

Scaling the data

In [25]:
```
from sklearn.preprocessing import StandardScaler
scaler = StandardScaler()
scaler.fit(X_train)
```

Out[25]:
```
StandardScaler
StandardScaler()
```

In [26]:
```
print(scaler.mean_)
```

```
[ 8.15216554  0.5259254   0.26865421  2.17671517  0.07918374 15.39109848
 43.29961278  0.99677205  3.31215105  0.63549515 10.38250077]
```

In [27]:
```
X_train_tf = scaler.transform(X_train)
X_train_tf
```

Out[27]:
```
array([[        nan, -1.0561401 ,  1.137821  , ..., -1.3770882 ,
         0.9624082 ,  0.01728958],
       [-0.97791845,  1.345196  , -1.3296037 , ...,  1.7045898 ,
        -0.21428543,  2.3885891 ],
       [-1.0452604 ,  0.6247952 , -1.3810083 , ..., -0.8037517 ,
        -1.5590785 , -0.87194717],
       ...,
       [-0.6412081 ,  0.5647619 , -1.0725802 , ...,  1.4179225 ,
        -0.8026323 , -0.87194717],
       [-0.16981342, -1.9566411 ,  0.41815543, ...,  0.05625052,
         1.3826561 ,  1.4993515 ],
       [-1.5839969 , -1.4163405 , -0.04448672, ...,  0.55792016,
        -0.8026323 ,  3.0802178 ]], dtype=float32)
```

In [28]:
```
X_train_tf_df = pd.DataFrame(X_train_tf, columns = outliers_columns)
X_train_tf_df
```

Out[28]:

	fixed acidity	volatile acidity	citric acid	residual sugar	chlorides	free sulfur dioxide	total sulfur dioxide	density	pH	sulphates	alcohol
0	NaN	-1.056140	1.137821	0.947174	-0.012438	1.245548	0.929073	1.629694	-1.377088	0.962408	0.017290
1	-0.977918	1.345196	-1.329604	0.499640	-0.080133	0.172623	-0.408483	-0.788820	1.704590	-0.214285	2.388589
2	-1.045260	0.624795	-1.381008	NaN	0.935292	NaN	1.236349	-0.739216	-0.803752	-1.559078	-0.871947
3	1.917791	-1.356307	1.600463	-0.171664	-1.704812	-0.900302	-0.878435	0.761511	-0.015416	2.055053	0.116094
4	0.166897	1.315179	-0.815557	-0.619199	0.732207	0.065331	0.712172	-0.168695	-0.875419	1.550755	-0.773143
...	...	...	...	...	...	...	...	...	...	...	...
1066	0.638292	0.444695	-1.381008	-0.619199	-1.434033	-1.114887	-1.203786	0.575441	-0.947085	-0.046186	0.017290
1067	0.032213	0.654812	-0.866962	-0.171664	-0.418608	1.030963	0.603721	-0.243138	-0.158750	0.962408	0.511310
1068	-0.641208	0.564762	-1.072580	1.170942	-0.147828	-0.041962	1.507475	0.426628	1.417922	-0.802632	-0.871947
1069	-0.169813	-1.956641	0.418155	-1.066734	-1.704812	-0.900302	-1.023035	-1.359378	0.056251	1.382656	1.499352
1070	-1.583997	-1.416340	-0.044487	-1.066734	-1.095558	-1.329471	-1.167636	NaN	0.557920	-0.802632	3.080218

1071 rows × 11 columns

In [29]:
```
X_train_tf_df.fillna(X_train_tf_df.mean(),inplace=True)
```

Support Vector Classifier(SVC)

In [30]:
```
from sklearn.svm import SVC
svc = SVC()
```

In [31]:
```
svc.fit(X_train_tf_df,y_train)
```

Out[31]:
```
SVC
SVC()
```

In [32]:
```
# Train Accuracy
svc.score(X_train_tf_df, y_train)
```

Out[32]:
```
0.6909430438842203
```

In [33]:
```
## Test data

X_test_tf = scaler.transform(X_test)
X_test_tf
```

Out[33]:
```
array([[-0.30449775,  0.20456141, -0.9697709 , ..., -0.5170843 ,
         0.20596243, -0.7731427 ],
       [-0.23715542, -0.15563901, -0.5071288 , ...,  0.55792016,
        -1.3069298 , -0.87194717],
       [ 1.7157642 ,  0.8649289 , -0.25010538, ..., -0.2304169 ,
         2.895548  , -0.47673106],
       ...,
```

```
       [ 0.0995549 , -1.2962736 ,  0.6237741 , ..., -0.01541633,
         1.1305072 ,  2.0921764 ],
       [ 0.03221259,         nan, -1.3810083 , ...,  1.5612562 ,
        -1.2228801 ,  0.5113102 ],
       [ 0.6382916 ,  0.8048956 , -0.6099381 , ..., -0.2304169 ,
        -0.8026323 , -0.7731427 ]], dtype=float32)
```

In [34]:
```
X_test_tf_df = pd.DataFrame(X_test_tf,columns=outliers_columns)
```

In [35]:
```
X_test_tf_df.isnull().sum()
```

Out[35]:
```
fixed acidity           17
volatile acidity         7
citric acid              0
residual sugar          46
chlorides               37
free sulfur dioxide     15
total sulfur dioxide    17
density                 16
pH                      10
sulphates               18
alcohol                  3
dtype: int64
```

In [36]:
```
X_test_tf_df.fillna(X_test_tf_df.mean(),inplace=True)
```

In [37]:
```
y_pred = svc.predict(X_test_tf_df)
```

In [38]:
```
y_test
```

Out[38]:
```
803     6
124     5
350     6
682     5
1326    6
       ..
813     4
377     7
898     7
126     5
819     5
Name: quality, Length: 528, dtype: int32
```

In [39]:
```
## Test Accuracy

from sklearn.metrics import classification_report, confusion_matrix
```

In [40]:
```
print("Confusion Matrix")
print(confusion_matrix(y_test,y_pred))
print('\n')
print('Classification Report')
print(classification_report(y_test,y_pred))
```

```
Confusion Matrix
[[  0   0   2   0   0   0]
 [  0   0  11   8   0   0]
 [  0   0 166  50   1   0]
 [  0   0  73 127  13   0]
 [  0   0   6  49  15   0]
 [  0   0   0   4   3   0]]


Classification Report
              precision    recall  f1-score   support

           3       0.00      0.00      0.00         2
           4       0.00      0.00      0.00        19
           5       0.64      0.76      0.70       217
           6       0.53      0.60      0.56       213
           7       0.47      0.21      0.29        70
           8       0.00      0.00      0.00         7

    accuracy                           0.58       528
   macro avg       0.27      0.26      0.26       528
weighted avg       0.54      0.58      0.55       528
```

Lets hypertune the model

In [41]:
```
from sklearn.model_selection import GridSearchCV
param_grid = {'C': [0.1,1, 10, 100], 'gamma': [1,0.1,0.01,0.001],'kernel':['linear','rbf']}
```

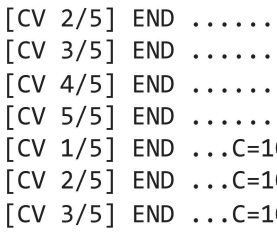
In [42]:
```
from sklearn.svm import SVC
svc = SVC()
```

In [43]:
```
grid_model = GridSearchCV(svc, param_grid,verbose=3)
grid_model.fit(X_train_tf_df, y_train)
```

```
Fitting 5 folds for each of 32 candidates, totalling 160 fits
[CV 1/5] END .....C=0.1, gamma=1, kernel=linear;, score=0.540 total time=   0.0s
[CV 2/5] END .....C=0.1, gamma=1, kernel=linear;, score=0.579 total time=   0.0s
[CV 3/5] END .....C=0.1, gamma=1, kernel=linear;, score=0.570 total time=   0.0s
[CV 4/5] END .....C=0.1, gamma=1, kernel=linear;, score=0.584 total time=   0.0s
[CV 5/5] END .....C=0.1, gamma=1, kernel=linear;, score=0.640 total time=   0.0s
[CV 1/5] END ........C=0.1, gamma=1, kernel=rbf;, score=0.433 total time=   0.5s
[CV 2/5] END ........C=0.1, gamma=1, kernel=rbf;, score=0.435 total time=   0.5s
[CV 3/5] END ........C=0.1, gamma=1, kernel=rbf;, score=0.435 total time=   0.5s
[CV 4/5] END ........C=0.1, gamma=1, kernel=rbf;, score=0.435 total time=   0.5s
[CV 5/5] END ........C=0.1, gamma=1, kernel=rbf;, score=0.430 total time=   0.5s
[CV 1/5] END ...C=0.1, gamma=0.1, kernel=linear;, score=0.540 total time=   0.0s
[CV 2/5] END ...C=0.1, gamma=0.1, kernel=linear;, score=0.579 total time=   0.0s
[CV 3/5] END ...C=0.1, gamma=0.1, kernel=linear;, score=0.570 total time=   0.0s
[CV 4/5] END ...C=0.1, gamma=0.1, kernel=linear;, score=0.584 total time=   0.0s
[CV 5/5] END ...C=0.1, gamma=0.1, kernel=linear;, score=0.640 total time=   0.0s
[CV 1/5] END ......C=0.1, gamma=0.1, kernel=rbf;, score=0.558 total time=   0.4s
[CV 2/5] END ......C=0.1, gamma=0.1, kernel=rbf;, score=0.575 total time=   0.4s
[CV 3/5] END ......C=0.1, gamma=0.1, kernel=rbf;, score=0.593 total time=   0.4s
[CV 4/5] END ......C=0.1, gamma=0.1, kernel=rbf;, score=0.612 total time=   0.4s
[CV 5/5] END ......C=0.1, gamma=0.1, kernel=rbf;, score=0.640 total time=   0.4s
[CV 1/5] END ..C=0.1, gamma=0.01, kernel=linear;, score=0.540 total time=   0.0s
[CV 2/5] END ..C=0.1, gamma=0.01, kernel=linear;, score=0.579 total time=   0.0s
[CV 3/5] END ..C=0.1, gamma=0.01, kernel=linear;, score=0.570 total time=   0.0s
[CV 4/5] END ..C=0.1, gamma=0.01, kernel=linear;, score=0.584 total time=   0.0s
[CV 5/5] END ..C=0.1, gamma=0.01, kernel=linear;, score=0.640 total time=   0.0s
[CV 1/5] END .....C=0.1, gamma=0.01, kernel=rbf;, score=0.516 total time=   0.4s
[CV 2/5] END .....C=0.1, gamma=0.01, kernel=rbf;, score=0.542 total time=   0.4s
[CV 3/5] END .....C=0.1, gamma=0.01, kernel=rbf;, score=0.565 total time=   0.4s
[CV 4/5] END .....C=0.1, gamma=0.01, kernel=rbf;, score=0.607 total time=   0.4s
[CV 5/5] END .....C=0.1, gamma=0.01, kernel=rbf;, score=0.626 total time=   0.4s
[CV 1/5] END .C=0.1, gamma=0.001, kernel=linear;, score=0.540 total time=   0.0s
[CV 2/5] END .C=0.1, gamma=0.001, kernel=linear;, score=0.579 total time=   0.0s
[CV 3/5] END .C=0.1, gamma=0.001, kernel=linear;, score=0.570 total time=   0.0s
[CV 4/5] END .C=0.1, gamma=0.001, kernel=linear;, score=0.584 total time=   0.0s
[CV 5/5] END .C=0.1, gamma=0.001, kernel=linear;, score=0.640 total time=   0.0s
[CV 1/5] END ....C=0.1, gamma=0.001, kernel=rbf;, score=0.433 total time=   0.4s
[CV 2/5] END ....C=0.1, gamma=0.001, kernel=rbf;, score=0.435 total time=   0.4s
[CV 3/5] END ....C=0.1, gamma=0.001, kernel=rbf;, score=0.435 total time=   0.4s
[CV 4/5] END ....C=0.1, gamma=0.001, kernel=rbf;, score=0.435 total time=   0.4s
[CV 5/5] END ....C=0.1, gamma=0.001, kernel=rbf;, score=0.430 total time=   0.4s
[CV 1/5] END .......C=1, gamma=1, kernel=linear;, score=0.549 total time=   0.0s
[CV 2/5] END .......C=1, gamma=1, kernel=linear;, score=0.561 total time=   0.0s
[CV 3/5] END .......C=1, gamma=1, kernel=linear;, score=0.561 total time=   0.0s
[CV 4/5] END .......C=1, gamma=1, kernel=linear;, score=0.579 total time=   0.0s
[CV 5/5] END .......C=1, gamma=1, kernel=linear;, score=0.640 total time=   0.0s
[CV 1/5] END ..........C=1, gamma=1, kernel=rbf;, score=0.600 total time=   0.4s
[CV 2/5] END ..........C=1, gamma=1, kernel=rbf;, score=0.589 total time=   0.4s
[CV 3/5] END ..........C=1, gamma=1, kernel=rbf;, score=0.612 total time=   0.4s
[CV 4/5] END ..........C=1, gamma=1, kernel=rbf;, score=0.701 total time=   0.5s
[CV 5/5] END ..........C=1, gamma=1, kernel=rbf;, score=0.654 total time=   0.4s
[CV 1/5] END .....C=1, gamma=0.1, kernel=linear;, score=0.549 total time=   0.0s
[CV 2/5] END .....C=1, gamma=0.1, kernel=linear;, score=0.561 total time=   0.0s
[CV 3/5] END .....C=1, gamma=0.1, kernel=linear;, score=0.561 total time=   0.0s
[CV 4/5] END .....C=1, gamma=0.1, kernel=linear;, score=0.579 total time=   0.0s
[CV 5/5] END .....C=1, gamma=0.1, kernel=linear;, score=0.640 total time=   0.0s
[CV 1/5] END ........C=1, gamma=0.1, kernel=rbf;, score=0.549 total time=   0.4s
[CV 2/5] END ........C=1, gamma=0.1, kernel=rbf;, score=0.589 total time=   0.4s
[CV 3/5] END ........C=1, gamma=0.1, kernel=rbf;, score=0.617 total time=   0.4s
[CV 4/5] END ........C=1, gamma=0.1, kernel=rbf;, score=0.621 total time=   0.4s
[CV 5/5] END ........C=1, gamma=0.1, kernel=rbf;, score=0.692 total time=   0.4s
[CV 1/5] END ....C=1, gamma=0.01, kernel=linear;, score=0.549 total time=   0.0s
[CV 2/5] END ....C=1, gamma=0.01, kernel=linear;, score=0.561 total time=   0.0s
[CV 3/5] END ....C=1, gamma=0.01, kernel=linear;, score=0.561 total time=   0.0s
[CV 4/5] END ....C=1, gamma=0.01, kernel=linear;, score=0.579 total time=   0.0s
[CV 5/5] END ....C=1, gamma=0.01, kernel=linear;, score=0.640 total time=   0.0s
[CV 1/5] END .......C=1, gamma=0.01, kernel=rbf;, score=0.540 total time=   0.4s
[CV 2/5] END .......C=1, gamma=0.01, kernel=rbf;, score=0.570 total time=   0.4s
[CV 3/5] END .......C=1, gamma=0.01, kernel=rbf;, score=0.570 total time=   0.4s
[CV 4/5] END .......C=1, gamma=0.01, kernel=rbf;, score=0.589 total time=   0.4s
[CV 5/5] END .......C=1, gamma=0.01, kernel=rbf;, score=0.640 total time=   0.4s
[CV 1/5] END ...C=1, gamma=0.001, kernel=linear;, score=0.549 total time=   0.0s
[CV 2/5] END ...C=1, gamma=0.001, kernel=linear;, score=0.561 total time=   0.0s
[CV 3/5] END ...C=1, gamma=0.001, kernel=linear;, score=0.561 total time=   0.0s
[CV 4/5] END ...C=1, gamma=0.001, kernel=linear;, score=0.579 total time=   0.0s
[CV 5/5] END ...C=1, gamma=0.001, kernel=linear;, score=0.640 total time=   0.0s
[CV 1/5] END ......C=1, gamma=0.001, kernel=rbf;, score=0.521 total time=   0.4s
[CV 2/5] END ......C=1, gamma=0.001, kernel=rbf;, score=0.547 total time=   0.5s
[CV 3/5] END ......C=1, gamma=0.001, kernel=rbf;, score=0.565 total time=   0.5s
[CV 4/5] END ......C=1, gamma=0.001, kernel=rbf;, score=0.607 total time=   0.5s
[CV 5/5] END ......C=1, gamma=0.001, kernel=rbf;, score=0.626 total time=   0.5s
[CV 1/5] END ......C=10, gamma=1, kernel=linear;, score=0.540 total time=   0.0s
[CV 2/5] END ......C=10, gamma=1, kernel=linear;, score=0.556 total time=   0.0s
[CV 3/5] END ......C=10, gamma=1, kernel=linear;, score=0.565 total time=   0.0s
[CV 4/5] END ......C=10, gamma=1, kernel=linear;, score=0.584 total time=   0.0s
[CV 5/5] END ......C=10, gamma=1, kernel=linear;, score=0.636 total time=   0.0s
[CV 1/5] END .........C=10, gamma=1, kernel=rbf;, score=0.605 total time=   0.4s
```

```
[CV 2/5] END .........C=10, gamma=1, kernel=rbf;, score=0.598 total time=   0.4s
[CV 3/5] END .........C=10, gamma=1, kernel=rbf;, score=0.612 total time=   0.4s
[CV 4/5] END .........C=10, gamma=1, kernel=rbf;, score=0.701 total time=   0.4s
[CV 5/5] END .........C=10, gamma=1, kernel=rbf;, score=0.668 total time=   0.4s
[CV 1/5] END ....C=10, gamma=0.1, kernel=linear;, score=0.540 total time=   0.0s
[CV 2/5] END ....C=10, gamma=0.1, kernel=linear;, score=0.556 total time=   0.0s
[CV 3/5] END ....C=10, gamma=0.1, kernel=linear;, score=0.565 total time=   0.0s
[CV 4/5] END ....C=10, gamma=0.1, kernel=linear;, score=0.584 total time=   0.0s
[CV 5/5] END ....C=10, gamma=0.1, kernel=linear;, score=0.636 total time=   0.0s
[CV 1/5] END .......C=10, gamma=0.1, kernel=rbf;, score=0.605 total time=   0.3s
[CV 2/5] END .......C=10, gamma=0.1, kernel=rbf;, score=0.584 total time=   0.3s
[CV 3/5] END .......C=10, gamma=0.1, kernel=rbf;, score=0.584 total time=   0.4s
[CV 4/5] END .......C=10, gamma=0.1, kernel=rbf;, score=0.607 total time=   0.3s
[CV 5/5] END .......C=10, gamma=0.1, kernel=rbf;, score=0.664 total time=   0.4s
[CV 1/5] END ...C=10, gamma=0.01, kernel=linear;, score=0.540 total time=   0.0s
[CV 2/5] END ...C=10, gamma=0.01, kernel=linear;, score=0.556 total time=   0.0s
[CV 3/5] END ...C=10, gamma=0.01, kernel=linear;, score=0.565 total time=   0.0s
[CV 4/5] END ...C=10, gamma=0.01, kernel=linear;, score=0.584 total time=   0.0s
[CV 5/5] END ...C=10, gamma=0.01, kernel=linear;, score=0.636 total time=   0.0s
[CV 1/5] END ......C=10, gamma=0.01, kernel=rbf;, score=0.549 total time=   0.3s
[CV 2/5] END ......C=10, gamma=0.01, kernel=rbf;, score=0.593 total time=   0.4s
[CV 3/5] END ......C=10, gamma=0.01, kernel=rbf;, score=0.612 total time=   0.4s
[CV 4/5] END ......C=10, gamma=0.01, kernel=rbf;, score=0.598 total time=   0.3s
[CV 5/5] END ......C=10, gamma=0.01, kernel=rbf;, score=0.673 total time=   0.4s
[CV 1/5] END ..C=10, gamma=0.001, kernel=linear;, score=0.540 total time=   0.0s
[CV 2/5] END ..C=10, gamma=0.001, kernel=linear;, score=0.556 total time=   0.0s
[CV 3/5] END ..C=10, gamma=0.001, kernel=linear;, score=0.565 total time=   0.0s
[CV 4/5] END ..C=10, gamma=0.001, kernel=linear;, score=0.584 total time=   0.0s
[CV 5/5] END ..C=10, gamma=0.001, kernel=linear;, score=0.636 total time=   0.0s
[CV 1/5] END .....C=10, gamma=0.001, kernel=rbf;, score=0.544 total time=   0.4s
[CV 2/5] END .....C=10, gamma=0.001, kernel=rbf;, score=0.575 total time=   0.4s
[CV 3/5] END .....C=10, gamma=0.001, kernel=rbf;, score=0.575 total time=   0.4s
[CV 4/5] END .....C=10, gamma=0.001, kernel=rbf;, score=0.593 total time=   0.4s
[CV 5/5] END .....C=10, gamma=0.001, kernel=rbf;, score=0.640 total time=   0.4s
[CV 1/5] END .....C=100, gamma=1, kernel=linear;, score=0.544 total time=   1.3s
[CV 2/5] END .....C=100, gamma=1, kernel=linear;, score=0.556 total time=   0.6s
[CV 3/5] END .....C=100, gamma=1, kernel=linear;, score=0.561 total time=   0.6s
[CV 4/5] END .....C=100, gamma=1, kernel=linear;, score=0.584 total time=   0.5s
[CV 5/5] END .....C=100, gamma=1, kernel=linear;, score=0.636 total time=   0.6s
[CV 1/5] END ........C=100, gamma=1, kernel=rbf;, score=0.605 total time=   0.4s
[CV 2/5] END ........C=100, gamma=1, kernel=rbf;, score=0.598 total time=   0.4s
[CV 3/5] END ........C=100, gamma=1, kernel=rbf;, score=0.612 total time=   0.4s
[CV 4/5] END ........C=100, gamma=1, kernel=rbf;, score=0.701 total time=   0.4s
[CV 5/5] END ........C=100, gamma=1, kernel=rbf;, score=0.668 total time=   0.4s
[CV 1/5] END ...C=100, gamma=0.1, kernel=linear;, score=0.544 total time=   1.4s
[CV 2/5] END ...C=100, gamma=0.1, kernel=linear;, score=0.556 total time=   0.5s
[CV 3/5] END ...C=100, gamma=0.1, kernel=linear;, score=0.561 total time=   0.7s
[CV 4/5] END ...C=100, gamma=0.1, kernel=linear;, score=0.584 total time=   0.5s
[CV 5/5] END ...C=100, gamma=0.1, kernel=linear;, score=0.636 total time=   0.6s
[CV 1/5] END ......C=100, gamma=0.1, kernel=rbf;, score=0.605 total time=   0.3s
[CV 2/5] END ......C=100, gamma=0.1, kernel=rbf;, score=0.579 total time=   0.3s
[CV 3/5] END ......C=100, gamma=0.1, kernel=rbf;, score=0.579 total time=   0.4s
[CV 4/5] END ......C=100, gamma=0.1, kernel=rbf;, score=0.607 total time=   0.4s
[CV 5/5] END ......C=100, gamma=0.1, kernel=rbf;, score=0.589 total time=   0.4s
[CV 1/5] END ..C=100, gamma=0.01, kernel=linear;, score=0.544 total time=   1.4s
[CV 2/5] END ..C=100, gamma=0.01, kernel=linear;, score=0.556 total time=   0.5s
[CV 3/5] END ..C=100, gamma=0.01, kernel=linear;, score=0.561 total time=   0.7s
[CV 4/5] END ..C=100, gamma=0.01, kernel=linear;, score=0.584 total time=   0.5s
[CV 5/5] END ..C=100, gamma=0.01, kernel=linear;, score=0.636 total time=   0.6s
[CV 1/5] END .....C=100, gamma=0.01, kernel=rbf;, score=0.572 total time=   0.4s
[CV 2/5] END .....C=100, gamma=0.01, kernel=rbf;, score=0.593 total time=   0.4s
[CV 3/5] END .....C=100, gamma=0.01, kernel=rbf;, score=0.584 total time=   0.4s
[CV 4/5] END .....C=100, gamma=0.01, kernel=rbf;, score=0.626 total time=   0.4s
[CV 5/5] END .....C=100, gamma=0.01, kernel=rbf;, score=0.673 total time=   0.4s
[CV 1/5] END .C=100, gamma=0.001, kernel=linear;, score=0.544 total time=   1.4s
[CV 2/5] END .C=100, gamma=0.001, kernel=linear;, score=0.556 total time=   0.6s
[CV 3/5] END .C=100, gamma=0.001, kernel=linear;, score=0.561 total time=   0.8s
[CV 4/5] END .C=100, gamma=0.001, kernel=linear;, score=0.584 total time=   0.7s
[CV 5/5] END .C=100, gamma=0.001, kernel=linear;, score=0.636 total time=   0.8s
[CV 1/5] END ....C=100, gamma=0.001, kernel=rbf;, score=0.544 total time=   0.4s
[CV 2/5] END ....C=100, gamma=0.001, kernel=rbf;, score=0.561 total time=   0.5s
[CV 3/5] END ....C=100, gamma=0.001, kernel=rbf;, score=0.570 total time=   0.5s
[CV 4/5] END ....C=100, gamma=0.001, kernel=rbf;, score=0.570 total time=   0.5s
[CV 5/5] END ....C=100, gamma=0.001, kernel=rbf;, score=0.636 total time=   0.5s
```

Out[43]:



In [44]:

```
print(grid_model.best_params_)
```

```
{'C': 10, 'gamma': 1, 'kernel': 'rbf'}
```

In [45]:

```
grid_model.score(X_train_tf_df, y_train)
```

Out[45]:

```
1.0
```

In [46]:

```
# Test Data

from sklearn.metrics import classification_report, confusion_matrix
```

In [47]:
```
grid_model_pred = grid_model.predict(X_test_tf_df)
print("Confusion Matrix")
print(confusion_matrix(y_test, grid_model_pred))
print('\n')
print("Classification report")
print(classification_report(y_test, grid_model_pred))
```

```
Confusion Matrix
[[  0   0   2   0   0   0]
 [  0   0  11   8   0   0]
 [  0   0 160  57   0   0]
 [  0   0  55 155   3   0]
 [  0   0   7  40  22   1]
 [  0   0   0   5   1   1]]


Classification report
              precision    recall  f1-score   support

           3       0.00      0.00      0.00         2
           4       0.00      0.00      0.00        19
           5       0.68      0.74      0.71       217
           6       0.58      0.73      0.65       213
           7       0.85      0.31      0.46        70
           8       0.50      0.14      0.22         7

    accuracy                           0.64       528
   macro avg       0.44      0.32      0.34       528
weighted avg       0.63      0.64      0.62       528
```

Sampling to handle imbalanced dataset

In [48]:
```
X.shape
```

Out[48]:
```
(1599, 11)
```

In [49]:
```
X.isnull().sum()
```

Out[49]:
```
fixed acidity           49
volatile acidity        19
citric acid              1
residual sugar         155
chlorides              112
free sulfur dioxide     30
total sulfur dioxide    55
density                 45
pH                      35
sulphates               59
alcohol                 13
dtype: int64
```

In [50]:
```
X.fillna(X.mean(),inplace=True)
```

from imblearn.combine import SMOTETomek from imblearn.under_sampling import TomekLinks!pip install imblearn

In [51]:
```
import imblearn
from imblearn.over_sampling import SMOTE
# Resampling the minority class. The strategy can be changed as required.
sm = SMOTE(sampling_strategy = 'minority',random_state=42)
# Fit the model to generate the data.
oversampled_X, oversampled_Y = sm.fit_resample(X,y)
oversampled = pd.concat([pd.DataFrame(oversampled_Y), pd.DataFrame(oversampled_X)], axis=1)
```

In [52]:
```
pip install -U imbalanced-learn
```

```
Requirement already satisfied: imbalanced-learn in c:\users\rosha\anaconda3\lib\site-packages (0.10.1)
Requirement already satisfied: numpy>=1.17.3 in c:\users\rosha\anaconda3\lib\site-packages (from imbalanced-learn) (1.24.
2)
Requirement already satisfied: scikit-learn>=1.0.2 in c:\users\rosha\anaconda3\lib\site-packages (from imbalanced-learn)
(1.2.1)
Requirement already satisfied: scipy>=1.3.2 in c:\users\rosha\anaconda3\lib\site-packages (from imbalanced-learn) (1.10.
1)
Requirement already satisfied: threadpoolctl>=2.0.0 in c:\users\rosha\anaconda3\lib\site-packages (from imbalanced-learn)
(3.1.0)
Requirement already satisfied: joblib>=1.1.1 in c:\users\rosha\anaconda3\lib\site-packages (from imbalanced-learn) (1.2.
0)
Note: you may need to restart the kernel to use updated packages.
WARNING: Ignoring invalid distribution -ython (c:\users\rosha\anaconda3\lib\site-packages)
WARNING: Ignoring invalid distribution -ython (c:\users\rosha\anaconda3\lib\site-packages)
WARNING: Ignoring invalid distribution -ython (c:\users\rosha\anaconda3\lib\site-packages)
WARNING: Ignoring invalid distribution -ython (c:\users\rosha\anaconda3\lib\site-packages)
WARNING: Ignoring invalid distribution -ython (c:\users\rosha\anaconda3\lib\site-packages)
WARNING: Ignoring invalid distribution -ython (c:\users\rosha\anaconda3\lib\site-packages)
```

In [53]:
```
oversampled_X.shape
```

Out[53]: (2270, 11)

In [54]:
```
oversampled_Y.shape
```

Out[54]: (2270,)

In [55]:
```
## Since our dfset shape has changed we need to perform train_test_split again
```

In [56]:
```
 X_train_sm, X_test_sm, y_train_sm, y_test_sm = train_test_split(oversampled_X,
oversampled_Y, test_size=0.25, random_state=44)
```

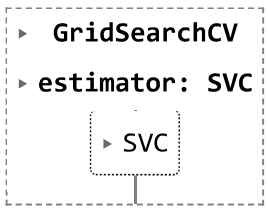
In [59]:
```
# Applying standardization to our model
scaler.fit(X_train_sm)
X_train_sm_tf = scaler.transform(X_train_sm)
X_test_sm_tf = scaler.transform(X_test_sm)
```

In [60]:
```
grid_model.fit(X_train_sm_tf, y_train_sm)
```

```
Fitting 5 folds for each of 32 candidates, totalling 160 fits
[CV 1/5] END .....C=0.1, gamma=1, kernel=linear;, score=0.619 total time=   0.0s
[CV 2/5] END .....C=0.1, gamma=1, kernel=linear;, score=0.657 total time=   0.0s
[CV 3/5] END .....C=0.1, gamma=1, kernel=linear;, score=0.641 total time=   0.0s
[CV 4/5] END .....C=0.1, gamma=1, kernel=linear;, score=0.656 total time=   0.0s
[CV 5/5] END .....C=0.1, gamma=1, kernel=linear;, score=0.674 total time=   0.0s
[CV 1/5] END ........C=0.1, gamma=1, kernel=rbf;, score=0.548 total time=   1.2s
[CV 2/5] END ........C=0.1, gamma=1, kernel=rbf;, score=0.543 total time=   1.5s
[CV 3/5] END ........C=0.1, gamma=1, kernel=rbf;, score=0.559 total time=   1.5s
[CV 4/5] END ........C=0.1, gamma=1, kernel=rbf;, score=0.515 total time=   1.3s
[CV 5/5] END ........C=0.1, gamma=1, kernel=rbf;, score=0.512 total time=   1.3s
[CV 1/5] END ...C=0.1, gamma=0.1, kernel=linear;, score=0.619 total time=   0.0s
[CV 2/5] END ...C=0.1, gamma=0.1, kernel=linear;, score=0.657 total time=   0.0s
[CV 3/5] END ...C=0.1, gamma=0.1, kernel=linear;, score=0.641 total time=   0.0s
[CV 4/5] END ...C=0.1, gamma=0.1, kernel=linear;, score=0.656 total time=   0.0s
[CV 5/5] END ...C=0.1, gamma=0.1, kernel=linear;, score=0.674 total time=   0.0s
[CV 1/5] END ......C=0.1, gamma=0.1, kernel=rbf;, score=0.654 total time=   1.0s
[CV 2/5] END ......C=0.1, gamma=0.1, kernel=rbf;, score=0.695 total time=   1.0s
[CV 3/5] END ......C=0.1, gamma=0.1, kernel=rbf;, score=0.662 total time=   1.0s
[CV 4/5] END ......C=0.1, gamma=0.1, kernel=rbf;, score=0.674 total time=   1.0s
[CV 5/5] END ......C=0.1, gamma=0.1, kernel=rbf;, score=0.700 total time=   1.0s
[CV 1/5] END ..C=0.1, gamma=0.01, kernel=linear;, score=0.619 total time=   0.0s
[CV 2/5] END ..C=0.1, gamma=0.01, kernel=linear;, score=0.657 total time=   0.0s
[CV 3/5] END ..C=0.1, gamma=0.01, kernel=linear;, score=0.641 total time=   0.0s
[CV 4/5] END ..C=0.1, gamma=0.01, kernel=linear;, score=0.656 total time=   0.0s
[CV 5/5] END ..C=0.1, gamma=0.01, kernel=linear;, score=0.674 total time=   0.0s
[CV 1/5] END .....C=0.1, gamma=0.01, kernel=rbf;, score=0.560 total time=   1.1s
[CV 2/5] END .....C=0.1, gamma=0.01, kernel=rbf;, score=0.587 total time=   1.2s
[CV 3/5] END .....C=0.1, gamma=0.01, kernel=rbf;, score=0.600 total time=   1.2s
[CV 4/5] END .....C=0.1, gamma=0.01, kernel=rbf;, score=0.565 total time=   1.2s
[CV 5/5] END .....C=0.1, gamma=0.01, kernel=rbf;, score=0.588 total time=   1.3s
[CV 1/5] END .C=0.1, gamma=0.001, kernel=linear;, score=0.619 total time=   0.0s
[CV 2/5] END .C=0.1, gamma=0.001, kernel=linear;, score=0.657 total time=   0.0s
[CV 3/5] END .C=0.1, gamma=0.001, kernel=linear;, score=0.641 total time=   0.0s
[CV 4/5] END .C=0.1, gamma=0.001, kernel=linear;, score=0.656 total time=   0.0s
[CV 5/5] END .C=0.1, gamma=0.001, kernel=linear;, score=0.674 total time=   0.0s
[CV 1/5] END ....C=0.1, gamma=0.001, kernel=rbf;, score=0.299 total time=   1.3s
[CV 2/5] END ....C=0.1, gamma=0.001, kernel=rbf;, score=0.299 total time=   1.2s
[CV 3/5] END ....C=0.1, gamma=0.001, kernel=rbf;, score=0.300 total time=   1.2s
[CV 4/5] END ....C=0.1, gamma=0.001, kernel=rbf;, score=0.300 total time=   1.2s
[CV 5/5] END ....C=0.1, gamma=0.001, kernel=rbf;, score=0.297 total time=   1.2s
[CV 1/5] END .......C=1, gamma=1, kernel=linear;, score=0.639 total time=   0.0s
[CV 2/5] END .......C=1, gamma=1, kernel=linear;, score=0.666 total time=   0.0s
[CV 3/5] END .......C=1, gamma=1, kernel=linear;, score=0.674 total time=   0.0s
[CV 4/5] END .......C=1, gamma=1, kernel=linear;, score=0.659 total time=   0.0s
[CV 5/5] END .......C=1, gamma=1, kernel=linear;, score=0.700 total time=   0.0s
[CV 1/5] END ..........C=1, gamma=1, kernel=rbf;, score=0.736 total time=   1.1s
[CV 2/5] END ..........C=1, gamma=1, kernel=rbf;, score=0.739 total time=   1.0s
[CV 3/5] END ..........C=1, gamma=1, kernel=rbf;, score=0.753 total time=   1.0s
[CV 4/5] END ..........C=1, gamma=1, kernel=rbf;, score=0.729 total time=   1.0s
[CV 5/5] END ..........C=1, gamma=1, kernel=rbf;, score=0.715 total time=   1.0s
[CV 1/5] END .....C=1, gamma=0.1, kernel=linear;, score=0.639 total time=   0.0s
[CV 2/5] END .....C=1, gamma=0.1, kernel=linear;, score=0.666 total time=   0.0s
[CV 3/5] END .....C=1, gamma=0.1, kernel=linear;, score=0.674 total time=   0.0s
[CV 4/5] END .....C=1, gamma=0.1, kernel=linear;, score=0.659 total time=   0.0s
[CV 5/5] END .....C=1, gamma=0.1, kernel=linear;, score=0.700 total time=   0.0s
[CV 1/5] END ........C=1, gamma=0.1, kernel=rbf;, score=0.701 total time=   0.8s
[CV 2/5] END ........C=1, gamma=0.1, kernel=rbf;, score=0.721 total time=   0.8s
[CV 3/5] END ........C=1, gamma=0.1, kernel=rbf;, score=0.721 total time=   0.8s
[CV 4/5] END ........C=1, gamma=0.1, kernel=rbf;, score=0.706 total time=   0.8s
[CV 5/5] END ........C=1, gamma=0.1, kernel=rbf;, score=0.753 total time=   0.8s
[CV 1/5] END ....C=1, gamma=0.01, kernel=linear;, score=0.639 total time=   0.0s
[CV 2/5] END ....C=1, gamma=0.01, kernel=linear;, score=0.666 total time=   0.0s
[CV 3/5] END ....C=1, gamma=0.01, kernel=linear;, score=0.674 total time=   0.0s
[CV 4/5] END ....C=1, gamma=0.01, kernel=linear;, score=0.659 total time=   0.0s
[CV 5/5] END ....C=1, gamma=0.01, kernel=linear;, score=0.700 total time=   0.0s
[CV 1/5] END .......C=1, gamma=0.01, kernel=rbf;, score=0.610 total time=   1.0s
[CV 2/5] END .......C=1, gamma=0.01, kernel=rbf;, score=0.657 total time=   1.0s
[CV 3/5] END .......C=1, gamma=0.01, kernel=rbf;, score=0.644 total time=   1.0s
[CV 4/5] END .......C=1, gamma=0.01, kernel=rbf;, score=0.647 total time=   1.0s
```

```
[CV 5/5] END .......C=1, gamma=0.01, kernel=rbf;, score=0.650 total time=   1.0s
[CV 1/5] END ...C=1, gamma=0.001, kernel=linear;, score=0.639 total time=   0.0s
[CV 2/5] END ...C=1, gamma=0.001, kernel=linear;, score=0.666 total time=   0.0s
[CV 3/5] END ...C=1, gamma=0.001, kernel=linear;, score=0.674 total time=   0.0s
[CV 4/5] END ...C=1, gamma=0.001, kernel=linear;, score=0.659 total time=   0.0s
[CV 5/5] END ...C=1, gamma=0.001, kernel=linear;, score=0.700 total time=   0.0s
[CV 1/5] END ......C=1, gamma=0.001, kernel=rbf;, score=0.554 total time=   1.1s
[CV 2/5] END ......C=1, gamma=0.001, kernel=rbf;, score=0.587 total time=   1.1s
[CV 3/5] END ......C=1, gamma=0.001, kernel=rbf;, score=0.606 total time=   1.1s
[CV 4/5] END ......C=1, gamma=0.001, kernel=rbf;, score=0.559 total time=   1.1s
[CV 5/5] END ......C=1, gamma=0.001, kernel=rbf;, score=0.568 total time=   1.1s
[CV 1/5] END ......C=10, gamma=1, kernel=linear;, score=0.651 total time=   0.2s
[CV 2/5] END ......C=10, gamma=1, kernel=linear;, score=0.672 total time=   0.2s
[CV 3/5] END ......C=10, gamma=1, kernel=linear;, score=0.679 total time=   0.1s
[CV 4/5] END ......C=10, gamma=1, kernel=linear;, score=0.668 total time=   0.3s
[CV 5/5] END ......C=10, gamma=1, kernel=linear;, score=0.712 total time=   0.2s
[CV 1/5] END .........C=10, gamma=1, kernel=rbf;, score=0.721 total time=   1.0s
[CV 2/5] END .........C=10, gamma=1, kernel=rbf;, score=0.742 total time=   1.0s
[CV 3/5] END .........C=10, gamma=1, kernel=rbf;, score=0.753 total time=   1.0s
[CV 4/5] END .........C=10, gamma=1, kernel=rbf;, score=0.721 total time=   1.0s
[CV 5/5] END .........C=10, gamma=1, kernel=rbf;, score=0.712 total time=   1.0s
[CV 1/5] END ....C=10, gamma=0.1, kernel=linear;, score=0.651 total time=   0.2s
[CV 2/5] END ....C=10, gamma=0.1, kernel=linear;, score=0.672 total time=   0.2s
[CV 3/5] END ....C=10, gamma=0.1, kernel=linear;, score=0.679 total time=   0.2s
[CV 4/5] END ....C=10, gamma=0.1, kernel=linear;, score=0.668 total time=   0.3s
[CV 5/5] END ....C=10, gamma=0.1, kernel=linear;, score=0.712 total time=   0.2s
[CV 1/5] END .......C=10, gamma=0.1, kernel=rbf;, score=0.724 total time=   0.8s
[CV 2/5] END .......C=10, gamma=0.1, kernel=rbf;, score=0.745 total time=   0.8s
[CV 3/5] END .......C=10, gamma=0.1, kernel=rbf;, score=0.753 total time=   0.8s
[CV 4/5] END .......C=10, gamma=0.1, kernel=rbf;, score=0.741 total time=   0.7s
[CV 5/5] END .......C=10, gamma=0.1, kernel=rbf;, score=0.715 total time=   0.8s
[CV 1/5] END ...C=10, gamma=0.01, kernel=linear;, score=0.651 total time=   0.2s
[CV 2/5] END ...C=10, gamma=0.01, kernel=linear;, score=0.672 total time=   0.2s
[CV 3/5] END ...C=10, gamma=0.01, kernel=linear;, score=0.679 total time=   0.1s
[CV 4/5] END ...C=10, gamma=0.01, kernel=linear;, score=0.668 total time=   0.3s
[CV 5/5] END ...C=10, gamma=0.01, kernel=linear;, score=0.712 total time=   0.2s
[CV 1/5] END ......C=10, gamma=0.01, kernel=rbf;, score=0.683 total time=   0.8s
[CV 2/5] END ......C=10, gamma=0.01, kernel=rbf;, score=0.716 total time=   0.9s
[CV 3/5] END ......C=10, gamma=0.01, kernel=rbf;, score=0.674 total time=   0.9s
[CV 4/5] END ......C=10, gamma=0.01, kernel=rbf;, score=0.688 total time=   0.9s
[CV 5/5] END ......C=10, gamma=0.01, kernel=rbf;, score=0.718 total time=   0.9s
[CV 1/5] END ..C=10, gamma=0.001, kernel=linear;, score=0.651 total time=   0.2s
[CV 2/5] END ..C=10, gamma=0.001, kernel=linear;, score=0.672 total time=   0.2s
[CV 3/5] END ..C=10, gamma=0.001, kernel=linear;, score=0.679 total time=   0.1s
[CV 4/5] END ..C=10, gamma=0.001, kernel=linear;, score=0.668 total time=   0.3s
[CV 5/5] END ..C=10, gamma=0.001, kernel=linear;, score=0.712 total time=   0.2s
[CV 1/5] END .....C=10, gamma=0.001, kernel=rbf;, score=0.598 total time=   0.9s
[CV 2/5] END .....C=10, gamma=0.001, kernel=rbf;, score=0.648 total time=   1.0s
[CV 3/5] END .....C=10, gamma=0.001, kernel=rbf;, score=0.635 total time=   1.0s
[CV 4/5] END .....C=10, gamma=0.001, kernel=rbf;, score=0.615 total time=   1.0s
[CV 5/5] END .....C=10, gamma=0.001, kernel=rbf;, score=0.665 total time=   1.0s
[CV 1/5] END .....C=100, gamma=1, kernel=linear;, score=0.651 total time=   2.1s
[CV 2/5] END .....C=100, gamma=1, kernel=linear;, score=0.669 total time=   1.8s
[CV 3/5] END .....C=100, gamma=1, kernel=linear;, score=0.682 total time=   1.5s
[CV 4/5] END .....C=100, gamma=1, kernel=linear;, score=0.668 total time=   2.0s
[CV 5/5] END .....C=100, gamma=1, kernel=linear;, score=0.715 total time=   2.0s
[CV 1/5] END ........C=100, gamma=1, kernel=rbf;, score=0.721 total time=   1.0s
[CV 2/5] END ........C=100, gamma=1, kernel=rbf;, score=0.742 total time=   1.0s
[CV 3/5] END ........C=100, gamma=1, kernel=rbf;, score=0.753 total time=   1.0s
[CV 4/5] END ........C=100, gamma=1, kernel=rbf;, score=0.721 total time=   1.0s
[CV 5/5] END ........C=100, gamma=1, kernel=rbf;, score=0.712 total time=   1.0s
[CV 1/5] END ...C=100, gamma=0.1, kernel=linear;, score=0.651 total time=   2.0s
[CV 2/5] END ...C=100, gamma=0.1, kernel=linear;, score=0.669 total time=   1.7s
[CV 3/5] END ...C=100, gamma=0.1, kernel=linear;, score=0.682 total time=   1.5s
[CV 4/5] END ...C=100, gamma=0.1, kernel=linear;, score=0.668 total time=   1.9s
[CV 5/5] END ...C=100, gamma=0.1, kernel=linear;, score=0.715 total time=   1.9s
[CV 1/5] END ......C=100, gamma=0.1, kernel=rbf;, score=0.727 total time=   0.7s
[CV 2/5] END ......C=100, gamma=0.1, kernel=rbf;, score=0.742 total time=   0.7s
[CV 3/5] END ......C=100, gamma=0.1, kernel=rbf;, score=0.738 total time=   0.8s
[CV 4/5] END ......C=100, gamma=0.1, kernel=rbf;, score=0.700 total time=   0.7s
[CV 5/5] END ......C=100, gamma=0.1, kernel=rbf;, score=0.718 total time=   0.8s
[CV 1/5] END ..C=100, gamma=0.01, kernel=linear;, score=0.651 total time=   2.0s
[CV 2/5] END ..C=100, gamma=0.01, kernel=linear;, score=0.669 total time=   1.7s
[CV 3/5] END ..C=100, gamma=0.01, kernel=linear;, score=0.682 total time=   1.5s
[CV 4/5] END ..C=100, gamma=0.01, kernel=linear;, score=0.668 total time=   1.9s
[CV 5/5] END ..C=100, gamma=0.01, kernel=linear;, score=0.715 total time=   1.9s
[CV 1/5] END .....C=100, gamma=0.01, kernel=rbf;, score=0.701 total time=   0.8s
[CV 2/5] END .....C=100, gamma=0.01, kernel=rbf;, score=0.713 total time=   0.8s
[CV 3/5] END .....C=100, gamma=0.01, kernel=rbf;, score=0.726 total time=   0.8s
[CV 4/5] END .....C=100, gamma=0.01, kernel=rbf;, score=0.715 total time=   0.8s
[CV 5/5] END .....C=100, gamma=0.01, kernel=rbf;, score=0.744 total time=   0.8s
[CV 1/5] END .C=100, gamma=0.001, kernel=linear;, score=0.651 total time=   2.0s
[CV 2/5] END .C=100, gamma=0.001, kernel=linear;, score=0.669 total time=   1.8s
[CV 3/5] END .C=100, gamma=0.001, kernel=linear;, score=0.682 total time=   1.6s
[CV 4/5] END .C=100, gamma=0.001, kernel=linear;, score=0.668 total time=   2.0s
[CV 5/5] END .C=100, gamma=0.001, kernel=linear;, score=0.715 total time=   1.9s
[CV 1/5] END ....C=100, gamma=0.001, kernel=rbf;, score=0.630 total time=   0.9s
[CV 2/5] END ....C=100, gamma=0.001, kernel=rbf;, score=0.674 total time=   0.9s
[CV 3/5] END ....C=100, gamma=0.001, kernel=rbf;, score=0.662 total time=   0.9s
[CV 4/5] END ....C=100, gamma=0.001, kernel=rbf;, score=0.656 total time=   0.9s
[CV 5/5] END ....C=100, gamma=0.001, kernel=rbf;, score=0.676 total time=   0.9s
```

Out[60]:



In [61]:
```
## Train Accuracy
grid_model.score(X_train_sm_tf, y_train_sm)
```

Out[61]:
```
0.9242068155111633
```

In [62]:
```
# Test Data
# Prediction on test data

y_pred_grid_sm = grid_model.predict(X_test_sm_tf)
```

In [63]:
```
# Test Accuracy
from sklearn.metrics import classification_report, confusion_matrix
```

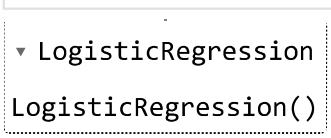
In [64]:
```
print("Confusion Matrix")
print(confusion_matrix(y_test_sm,y_pred_grid_sm))
print("Classification Report")
print(classification_report(y_test_sm,y_pred_grid_sm))
```

```
Confusion Matrix
[[179   0   0   0   0   0]
 [  0   0  11   3   0   0]
 [  3   3 126  39   1   0]
 [  1   0  45  88  15   0]
 [  0   0   5  18  24   1]
 [  0   0   1   2   3   0]]
Classification Report
              precision    recall  f1-score   support

           3       0.98      1.00      0.99       179
           4       0.00      0.00      0.00        14
           5       0.67      0.73      0.70       172
           6       0.59      0.59      0.59       149
           7       0.56      0.50      0.53        48
           8       0.00      0.00      0.00         6

    accuracy                           0.73       568
   macro avg       0.47      0.47      0.47       568
weighted avg       0.71      0.73      0.72       568
```

Logistic Regressionn Model

In [65]:
```
from sklearn.linear_model import LogisticRegression
lr = LogisticRegression()
lr.fit(X_train_tf, y_train)
```

Out[65]:



In [66]:
```
## Train Accuracy
lr.score(X_train_tf,y_train)
```

Out[66]:
```
0.6041083099906629
```

In [67]:
```
# Test Data
y_pred_lr = lr.predict(X_test_tf)
```

In [68]:
```
## Test Accuracy
from sklearn.metrics import classification_report, confusion_matrix
```

In [72]:
```
print("COnfusion Matrix")
print(confusion_matrix(y_test,y_pred_lr))
print('\n')
print("Classification report")
print(classification_report(y_test,y_pred_lr))
```

```
COnfusion Matrix
[[  0   0   1   1   0   0]
 [  0   0   9  10   0   0]
 [  0   0 165  50   2   0]
 [  0   0  82 110  21   0]
 [  0   0   5  47  18   0]
 [  0   0   0   4   3   0]]
```

```
Classification report
              precision    recall  f1-score   support

           3       0.00      0.00      0.00         2
           4       0.00      0.00      0.00        19
           5       0.63      0.76      0.69       217
           6       0.50      0.52      0.51       213
           7       0.41      0.26      0.32        70
           8       0.00      0.00      0.00         7

    accuracy                           0.55       528
   macro avg       0.26      0.26      0.25       528
weighted avg       0.51      0.55      0.53       528
```

Conclusion

In [82]:

```
from sklearn.metrics import accuracy_score
print("Final Test Accuracy Score of all the Models")
print('\n')
print('SVC : ',round(accuracy_score(y_test,y_pred),2)*100)
print('\n')
print('SVC after hyperparmeter tunning : ',round(accuracy_score(y_test,grid_model_pred)))
print('\n')
print('SVC after sampling using hypertunned model : ',round(accuracy_score(y_test_sm,y_pred_grid_sm),2)*100)
print('\n')
print('Logistic Regression : ',round(accuracy_score(y_test, y_pred_lr),2)*100)
```

```
Final Test Accuracy Score of all the Models


SVC :  57.99999999999999


SVC after hyperparmeter tunning :  1


SVC after sampling using hypertunned model :  73.0


Logistic Regression :  55.00000000000001
```

In [ ]: