Modules and Their Version

In [1]:
```
import sys
print("Python Version :",sys.version)
```

```
Python Version : 3.8.1 (tags/v3.8.1:1b293b6, Dec 18 2019, 22:39:24) [MS
C v.1916 32 bit (Intel)]
```

In [2]:
```
import numpy as np
print("Numpy Version :",np.__version__)
```

```
Numpy Version : 1.19.1
```

In [3]:
```
import pandas as pd
print("Pandas version :",pd.__version__)
```

```
Pandas version : 1.1.0
```

In [4]:
```
import sklearn
print("Sklearn Version :",sklearn.__version__)
```

```
Sklearn Version : 0.23.2
```

In [5]:
```
import matplotlib
print("Matplotlib Version :",matplotlib.__version__)
```

```
Matplotlib Version : 3.3.1
```

In [6]:
```
import scipy as sc
print("Scipy Version :",sc.__version__)
```

```
Scipy Version : 1.5.2
```

In [7]:
```
import matplotlib.pyplot as plt
from pandas.plotting import scatter_matrix
from sklearn.model_selection import train_test_split
from sklearn.metrics import accuracy_score
from sklearn.metrics import confusion_matrix
from sklearn.metrics import classification_report
from sklearn.neighbors import KNeighborsClassifier
```

Load The Dataset Using Pandas(titanic-dataset)

In [8]:
```
pd.set_option('display.max_columns',10,'display.width',1000)
titanic_dataset = pd.read_csv("E:\\java\\titanic.csv")
titanic_dataset.head(5)
```

Out[8]:

	PassengerId	Survived	Pclass	Name	Sex	...	Parch	Ticket	Fare	Cabin	Emba
0	1	0	3	Braund, Mr. Owen Harris	male	...	0	A/5 21171	7.2500	NaN	
1	2	1	1	Cumings, Mrs. John Bradley (Florence Briggs Th...	female	...	0	PC 17599	71.2833	C85	
2	3	1	3	Heikkinen, Miss. Laina	female	...	0	STON/O2. 3101282	7.9250	NaN	
3	4	1	1	Futrelle, Mrs. Jacques Heath (Lily May Peel)	female	...	0	113803	53.1000	C123	
4	5	0	3	Allen, Mr. William Henry	male	...	0	373450	8.0500	NaN	

5 rows × 12 columns

In [9]:
```
titanic_dataset.tail(5)
```

Out[9]:

	PassengerId	Survived	Pclass	Name	Sex	...	Parch	Ticket	Fare	Cabin	Embarke
886	887	0	2	Montvila, Rev. Juozas	male	...	0	211536	13.00	NaN	
887	888	1	1	Graham, Miss. Margaret Edith	female	...	0	112053	30.00	B42	
888	889	0	3	Johnston, Miss. Catherine Helen "Carrie"	female	...	2	W./C. 6607	23.45	NaN	
889	890	1	1	Behr, Mr. Karl Howell	male	...	0	111369	30.00	C148	
890	891	0	3	Dooley, Mr. Patrick	male	...	0	370376	7.75	NaN	

5 rows × 12 columns

In [10]:
```
print("Shape of Data :",titanic_dataset.shape)
```

```
Shape of Data : (891, 12)
```

In [11]:
```
print("Missing Values : \n",titanic_dataset.isna().sum())
```

```
Missing Values :
 PassengerId      0
Survived         0
```

```
Pclass           0
Name             0
Sex              0
Age            177
SibSp            0
Parch            0
Ticket           0
Fare             0
Cabin          687
Embarked         2
dtype: int64
```

In [12]:
```
print(titanic_dataset.groupby('Embarked').size())
```

```
Embarked
C    168
Q     77
S    644
dtype: int64
```

In [13]:
```
print(titanic_dataset.describe())
```

```
       PassengerId    Survived      Pclass         Age       SibSp
  Parch        Fare
count   891.000000  891.000000  891.000000  714.000000  891.000000  89
1.000000  891.000000
mean    446.000000    0.383838    2.308642   29.699118    0.523008
0.381594   32.204208
std     257.353842    0.486592    0.836071   14.526497    1.102743
0.806057   49.693429
min       1.000000    0.000000    1.000000    0.420000    0.000000
0.000000    0.000000
25%     223.500000    0.000000    2.000000   20.125000    0.000000
0.000000    7.910400
50%     446.000000    0.000000    3.000000   28.000000    0.000000
0.000000   14.454200
75%     668.500000    1.000000    3.000000   38.000000    1.000000
0.000000   31.000000
```

```
max     891.000000    1.000000    3.000000   80.000000    8.000000
6.000000  512.329200
```

Removing Cabin,Name and Ticket column

In [14]:
```
titanic_dataset=titanic_dataset.drop(columns=['Cabin','Name','Ticket'])
```

In [15]:
```
print("Missing Values : \n",titanic_dataset.isna().sum())
```

```
Missing Values :
 PassengerId      0
Survived          0
Pclass            0
Sex               0
Age             177
SibSp             0
Parch             0
Fare              0
Embarked          2
dtype: int64
```

Removing two missing Embarked row

In [16]:
```
titanic_dataset = titanic_dataset.dropna(subset=['Embarked'])
```

In [17]:
```
print("Missing Values : \n",titanic_dataset.isna().sum())
titanic_dataset.shape
```

```
Missing Values :
 PassengerId      0
Survived          0
Pclass            0
Sex               0
Age             177
SibSp             0
Parch             0
Fare              0
```

```
Embarked         0
dtype: int64
```

Out[17]:
```
(889, 9)
```

Remove rows having NA Age value

In [49]:
```
titanic_dataset = titanic_dataset.dropna()
```

In [21]:
```
print("Missing Values : \n",titanic_dataset.isna().sum())
titanic_dataset.shape
```

```
Missing Values :
 PassengerId    0
Survived       0
Pclass         0
Sex            0
Age            0
SibSp          0
Parch          0
Fare           0
Embarked       0
dtype: int64
```

Out[21]:
```
(712, 9)
```

Preprocessing titanic_dataset

In [22]:
```
df2=titanic_dataset
```

In [23]:
```
from sklearn.preprocessing import LabelEncoder
labelencoder=LabelEncoder()
df2.Sex=labelencoder.fit_transform(df2.Sex)
```

In [24]:
```
df2
```

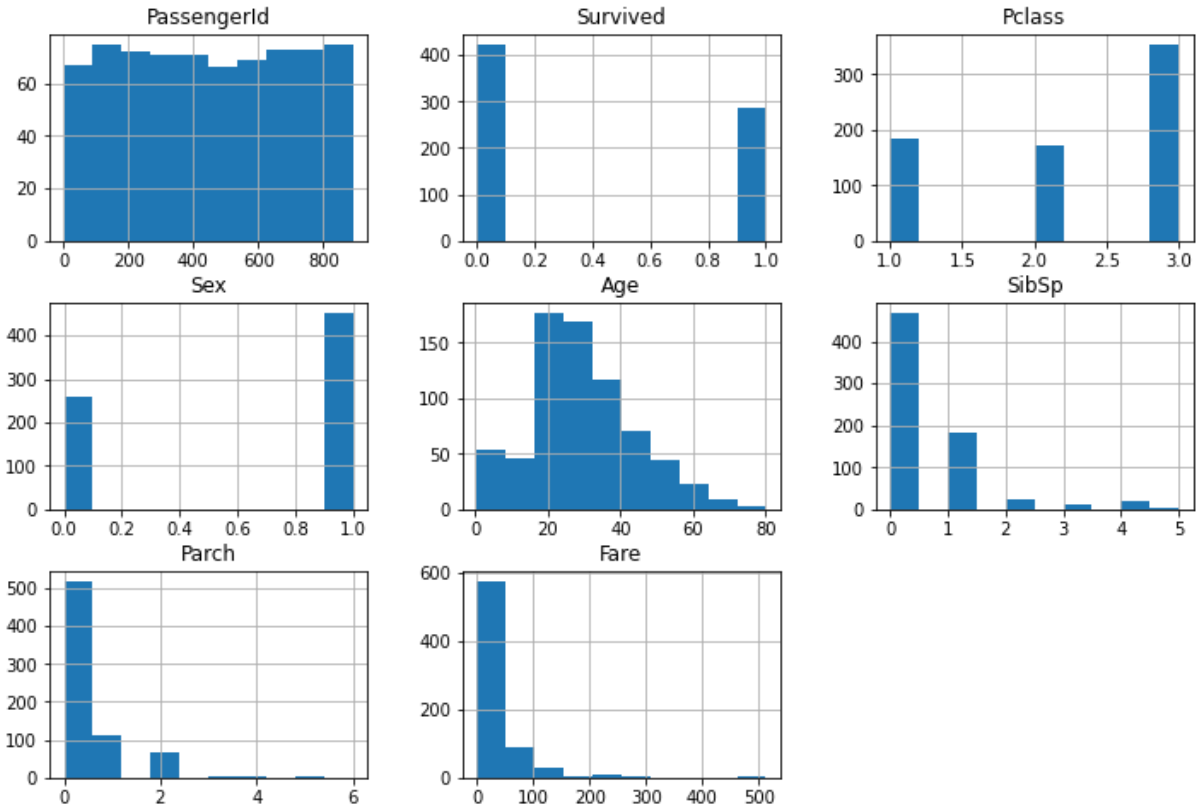
Out[24]:

	PassengerId	Survived	Pclass	Sex	Age	SibSp	Parch	Fare	Embarked
0	1	0	3	1	22.0	1	0	7.2500	S
1	2	1	1	0	38.0	1	0	71.2833	C
2	3	1	3	0	26.0	0	0	7.9250	S
3	4	1	1	0	35.0	1	0	53.1000	S
4	5	0	3	1	35.0	0	0	8.0500	S
...	...	...	...	...	...	...	...	...	...
885	886	0	3	0	39.0	0	5	29.1250	Q
886	887	0	2	1	27.0	0	0	13.0000	S
887	888	1	1	0	19.0	0	0	30.0000	S
889	890	1	1	1	26.0	0	0	30.0000	C
890	891	0	3	1	32.0	0	0	7.7500	Q

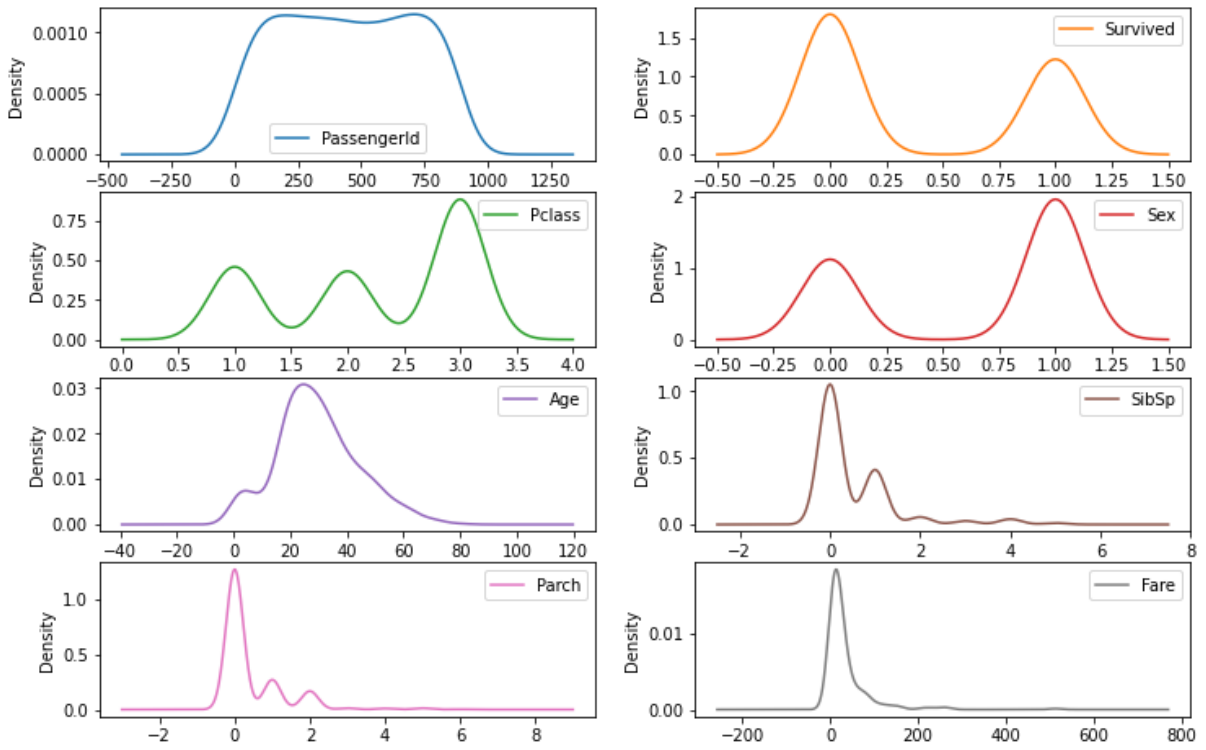
712 rows × 9 columns

Data Visulaization

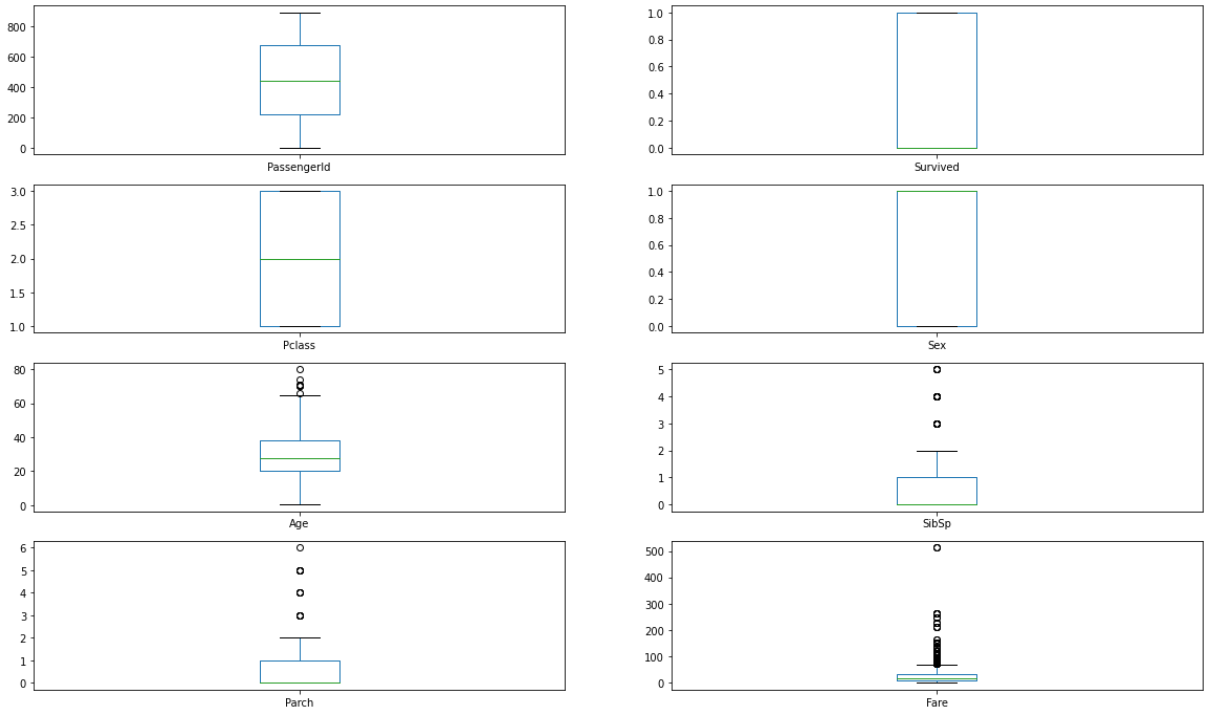
In [25]:
```
titanic_dataset.hist(figsize=(12,8),sharex=False)
plt.show()
```



In [26]:
```
titanic_dataset.plot(kind='density',subplots=True,layout=(4,2),sharex=F
alse,figsize=(12,8))
plt.show()
```



In [27]:
```
titanic_dataset.plot(kind='box',subplots=True,layout=(4,2),sharex=False
, figsize=(20,12))
plt.show()
```



In [28]:
```
correlation=titanic_dataset.corr()
```

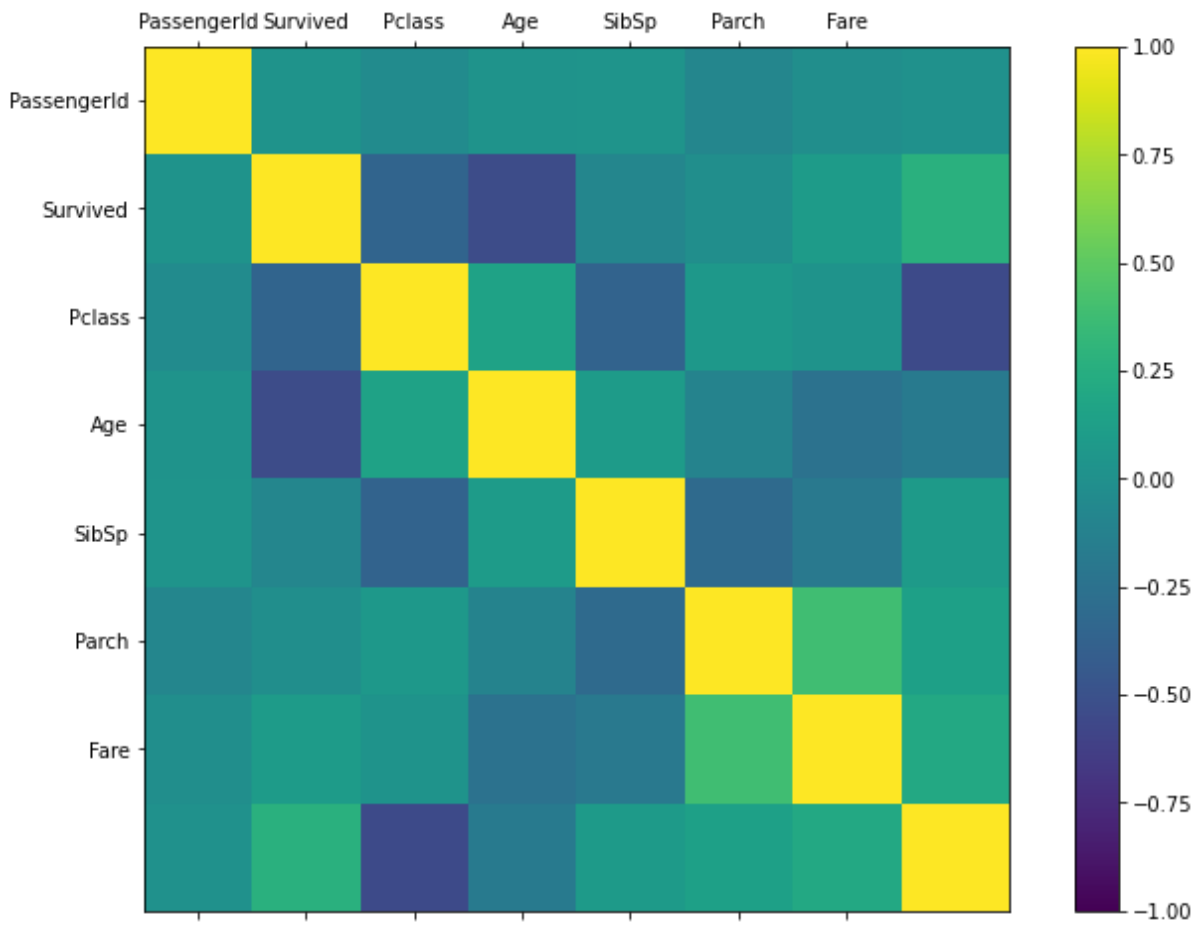
In [29]:
```
correlation
```

Out[29]:

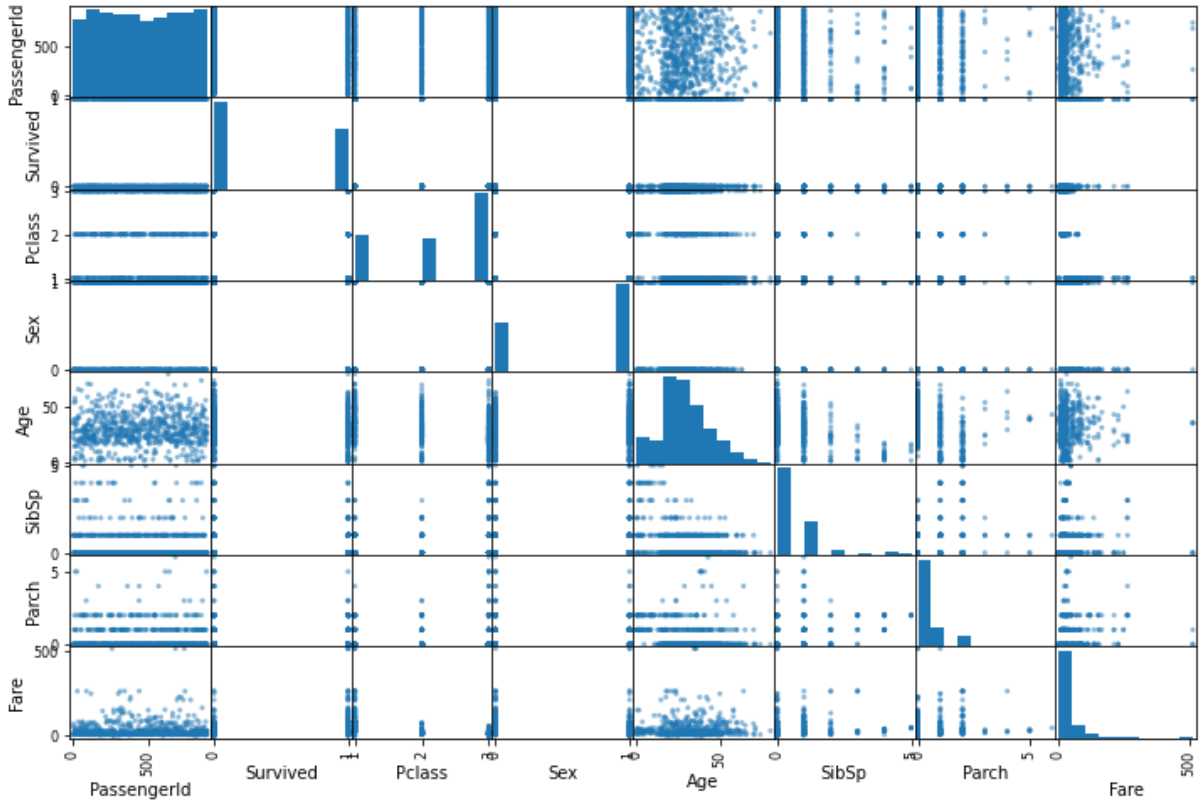
	PassengerId	Survived	Pclass	Sex	Age	SibSp	Parch	F
PassengerId	1.000000	0.029526	-0.035609	0.024674	0.033681	-0.082704	-0.011672	0.009
Survived	0.029526	1.000000	-0.356462	-0.536762	-0.082446	-0.015523	0.095265	0.266
Pclass	-0.035609	-0.356462	1.000000	0.150826	-0.365902	0.065187	0.023666	-0.552
Sex	0.024674	-0.536762	0.150826	1.000000	0.099037	-0.106296	-0.249543	-0.182
Age	0.033681	-0.082446	-0.365902	0.099037	1.000000	-0.307351	-0.187896	0.093
SibSp	-0.082704	-0.015523	0.065187	-0.106296	-0.307351	1.000000	0.383338	0.139

	PassengerId	Survived	Pclass	Sex	Age	SibSp	Parch	F
Parch	-0.011672	0.095265	0.023666	-0.249543	-0.187896	0.383338	1.000000	0.206
Fare	0.009655	0.266100	-0.552893	-0.182457	0.093143	0.139860	0.206624	1.000

In [30]:
```
fig=plt.figure(figsize=(12,8))
ax = fig.add_subplot(111)
cx= ax.matshow(correlation, vmax=1, vmin=-1)
ticks=np.arange(7)
labels =['Passengerld','Survived','Pclass','Age','SibSp','Parch','Fare'
]
ax.set_xticks(ticks)
ax.set_yticks(ticks)
ax.set_xticklabels(labels)
ax.set_yticklabels(labels)
fig.colorbar(cx)
plt.show()
```



In [31]:
```
scatter_matrix(titanic_dataset,figsize=(12,8))
plt.show()
```



In [32]:
```
x=titanic_dataset.iloc[:,:-1].values
y=titanic_dataset.iloc[:,-1:].values
```

In [33]:
```
X_train,X_test,Y_train,Y_test =  train_test_split(x,y,test_size=0.3,ran
dom_state=7)
```

Logistic regression model

In [34]:
```
from sklearn.linear_model import LogisticRegression
model5 = LogisticRegression()
model5.fit(X_train,Y_train)
Y_pred5 = model5.predict(X_test)
```

```
c:\users\hp\appdata\local\programs\python\python38-32\lib\site-packages
\sklearn\utils\validation.py:72: DataConversionWarning: A column-vector
y was passed when a 1d array was expected. Please change the shape of y
to (n_samples, ), for example using ravel().
  return f(**kwargs)
c:\users\hp\appdata\local\programs\python\python38-32\lib\site-packages
\sklearn\linear_model\_logistic.py:762: ConvergenceWarning: lbfgs faile
d to converge (status=1):
STOP: TOTAL NO. of ITERATIONS REACHED LIMIT.

Increase the number of iterations (max_iter) or scale the data as shown
in:
    https://scikit-learn.org/stable/modules/preprocessing.html
Please also refer to the documentation for alternative solver options:
    https://scikit-learn.org/stable/modules/linear_model.html#logistic-
regression
  n_iter_i = _check_optimize_result(
```

In [35]:
```
print("Accuracy by LogisticRegression :",accuracy_score(Y_test,Y_pred5
))
```

```
Accuracy by LogisticRegression : 0.7710280373831776
```

In [ ]:

In [ ]:

KNN model

In [36]:
```
model=KNeighborsClassifier(n_neighbors=10)
```

In [37]:
```
model.fit(X_train,Y_train)
```

```
<ipython-input-37-ffa49499a3bf>:1: DataConversionWarning: A column-vect
or y was passed when a 1d array was expected. Please change the shape o
f y to (n_samples, ), for example using ravel().
  model.fit(X_train,Y_train)
```

Out[37]:
```
KNeighborsClassifier(n_neighbors=10)
```

In [38]:
```
Y_pred=model.predict(X_test)
print("Accuracy by KNN :",accuracy_score(Y_test,Y_pred))
```

```
Accuracy by KNN : 0.7570093457943925
```

In [39]:
```
confusion_matrix(Y_test,Y_pred)
```

Out[39]:
```
array([[  0,   0,  37],
       [  0,   0,   9],
       [  6,   0, 162]], dtype=int64)
```

In [40]:
```
print(classification_report(Y_test,Y_pred))
```

```
              precision    recall  f1-score   support

           C       0.00      0.00      0.00        37
           Q       0.00      0.00      0.00         9
           S       0.78      0.96      0.86       168

    accuracy                           0.76       214
   macro avg       0.26      0.32      0.29       214
weighted avg       0.61      0.76      0.68       214
```

```
c:\users\hp\appdata\local\programs\python\python38-32\lib\site-packages
\sklearn\metrics\_classification.py:1221: UndefinedMetricWarning: Preci
sion and F-score are ill-defined and being set to 0.0 in labels with no
predicted samples. Use `zero_division` parameter to control this behavi
or.
  _warn_prf(average, modifier, msg_start, len(result))
```

support vector model

In [41]:
```
from sklearn.svm import SVC
model2=SVC()
```

```
model2.fit(X_train,Y_train)
Y_pred2=model2.predict(X_test)
```

```
c:\users\hp\appdata\local\programs\python\python38-32\lib\site-packages
\sklearn\utils\validation.py:72: DataConversionWarning: A column-vector
y was passed when a 1d array was expected. Please change the shape of y
to (n_samples, ), for example using ravel().
  return f(**kwargs)
```

In [42]:
```
print("Accuracy by SVM :",accuracy_score(Y_test,Y_pred2))
```

```
Accuracy by SVM : 0.7850467289719626
```

Naive bayes model

In [43]:
```
from sklearn.naive_bayes import GaussianNB
model3 = GaussianNB()
model3.fit(X_train,Y_train)
Y_pred3=model3.predict(X_test)
```

```
c:\users\hp\appdata\local\programs\python\python38-32\lib\site-packages
\sklearn\utils\validation.py:72: DataConversionWarning: A column-vector
y was passed when a 1d array was expected. Please change the shape of y
to (n_samples, ), for example using ravel().
  return f(**kwargs)
```

In [44]:
```
print("Accuracy score by Naive Bayes :",accuracy_score(Y_test,Y_pred3))
```

```
Accuracy score by Naive Bayes : 0.7570093457943925
```

In [45]:
```
from sklearn.tree import DecisionTreeClassifier
model4=DecisionTreeClassifier()
model4.fit(X_train,Y_train)
Y_pred4 = model4.predict(X_test)
```

In [46]:
```
print("Accuracy score by Decision tree :",accuracy_score(Y_test,Y_pred4
))
```

```
Accuracy score by Decision tree : 0.7570093457943925

-------->>>>>>>> Best accuracy I got by SVM <<<<<<<<<----------
```

In [47]:
```
print("Accuracy by SVM :",accuracy_score(Y_test,Y_pred2))
```

```
Accuracy by SVM : 0.7850467289719626
```