In [1]:
```
import pandas as pd
```

In [2]:
```
import numpy as np
```

In [3]:
```
from sklearn import preprocessing
```

In [4]:
```
from sklearn.model_selection import train_test_split
```

In [5]:
```
from sklearn.metrics import accuracy_score
```

In [6]:
```
from sklearn.metrics import confusion_matrix
```

In [104]:
```
from sklearn import svm
```

In [7]:
```
df=pd.read_csv(r'C:\Users\siyad\AppData\Local\Temp\Temp1_Dataset-20200813T141334Z-001.zip\Dataset\train.csv')
```

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

Out[8]:
```
Index(['PassengerId', 'Survived', 'Pclass', 'Name', 'Sex', 'Age', 'SibSp',
       'Parch', 'Ticket', 'Fare', 'Cabin', 'Embarked'],
      dtype='object')
```

In [19]:
```
df.head()
```

Out[19]:

	PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare	Cabin	Embarked
0	1	0	3	Braund, Mr. Owen Harris	1	22.0	1	0	A/5 21171	7.2500	NaN	S
1	2	1	1	Cumings, Mrs. John Bradley (Florence Briggs Th...	0	38.0	1	0	PC 17599	71.2833	C85	C
2	3	1	3	Heikkinen, Miss. Laina	0	26.0	0	0	STON/O2. 3101282	7.9250	NaN	S
3	4	1	1	Futrelle, Mrs. Jacques Heath (Lily May Peel)	0	35.0	1	0	113803	53.1000	C123	S
4	5	0	3	Allen, Mr. William Henry	1	35.0	0	0	373450	8.0500	NaN	S

In [35]:
```
le=preprocessing.LabelEncoder()
```

In [36]:
```
le.fit(df['Sex'])
```

Out[36]:
```
LabelEncoder()
```

In [37]:
```
print(le.classes_)
```

```
[0 1]
```

In [38]:
```
df['Sex']=le.transform(df['Sex'])
```

In [39]:
```
le.fit(df['Embarked'])
```

Out[39]:
```
LabelEncoder()
```

In [40]:
```
print(le.classes_)
```

```
[0 1 2]
```

In [41]:
```
df['Embarked']=le.transform(df['Embarked'])
```

In [42]:
```
from sklearn import neighbors
```

In [51]:
```
y=df['Pclass']
```

In [52]:
```
X=df.drop(['Pclass','PassengerId','Cabin','Name','Ticket'],axis=1)
```

In [53]:
```
X.head()
```

Out[53]:

	Survived	Sex	Age	SibSp	Parch	Fare	Embarked
0	0	1	22.0	1	0	7.2500	2
1	1	0	38.0	1	0	71.2833	0
2	1	0	26.0	0	0	7.9250	2
3	1	0	35.0	1	0	53.1000	2
4	0	1	35.0	0	0	8.0500	2

In [54]:
```
X_train,X_test,y_train,y_test=train_test_split(X,y,test_size=0.3,random_state=0)
```

In [55]:
```
X_train.head()
```

Out[55]:

	Survived	Sex	Age	SibSp	Parch	Fare	Embarked
350	0	1	45.0	0	0	35.0000	2
124	1	1	12.0	1	0	11.2417	0
577	0	0	45.0	1	0	14.4583	0
422	0	0	28.0	1	1	14.4000	2
118	0	0	2.0	4	2	31.2750	2

```
In [59]: y_train.head()

Out[59]: 350    1
         124    3
         577    3
         422    3
         118    3
         Name: Pclass, dtype: int64
```

```
In [57]: knn=neighbors.KNeighborsClassifier(n_neighbors=3)
```

```
In [60]: knn.fit(X_train,y_train).score(X_test,y_test)

Out[60]: 0.8651685393258427
```

```
In [61]: y_pred=knn.predict(X_test)
```

```
In [62]: confusion_matrix(y_test,y_pred)

Out[62]: array([[ 63,   5,   2],
                [  7,  28,  14],
                [  3,   5, 140]], dtype=int64)
```

```
In [101]: knn=neighbors.KNeighborsClassifier(n_neighbors=267)
```

```
In [98]: knn.fit(X_train,y_train).score(X_test,y_test)

Out[98]: 0.8651685393258427
```

```
In [99]: y_pred=knn.predict(X_test)
```

```
In [100]: confusion_matrix(y_test,y_pred)

Out[100]: array([[ 67,   2,   1],
                 [  8,  37,   4],
                 [  4,  17, 127]], dtype=int64)
```

```
In [105]: X_train.head()
```

Out[105]:

	Survived	Sex	Age	SibSp	Parch	Fare	Embarked
350	0	1	45.0	0	0	35.0000	2
124	1	1	12.0	1	0	11.2417	0
577	0	0	45.0	1	0	14.4583	0
422	0	0	28.0	1	1	14.4000	2
118	0	0	2.0	4	2	31.2750	2

In [107]:
```
X_test.head()
```

Out[107]:

	Survived	Sex	Age	SibSp	Parch	Fare	Embarked
14	0	0	14.0	0	0	7.8542	2
158	0	1	45.0	8	2	69.5500	2
762	1	0	36.0	1	2	120.0000	2
740	0	1	36.0	1	0	78.8500	2
482	1	0	63.0	0	0	9.5875	2

In [108]:
```
clf=svm.SVC(gamma=0.01,C=100)
```

In [109]:
```
clf.fit(X_train,y_train)
```

Out[109]:
```
SVC(C=100, gamma=0.01)
```

In [110]:
```
y_pred=clf.predict(X_test)
```

In [112]:
```
accuracy_score(y_test,y_pred,normalize=True)
```

Out[112]:
```
0.9026217228464419
```

In [113]:
```
confusion_matrix(y_test,y_pred)
```

Out[113]:
```
array([[ 64,   4,   2],
       [  4,  36,   9],
       [  1,   6, 141]], dtype=int64)
```

In [ ]: