In [1]:

```
import pandas as pd
```

In [2]:

```
df1 = pd.read_csv('train.csv')
```

In [3]:

```
df1.head()
```

Out[3]:

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

In [4]:

```
df1.drop(['PassengerId','Name','Ticket','Cabin','Embarked','SibSp','Parch'], axis=1, inplac
```

In [5]:

```
df1.head()
```

Out[5]:

	Survived	Pclass	Sex	Age	Fare
0	0	3	male	22.0	7.2500
1	1	1	female	38.0	71.2833
2	1	3	female	26.0	7.9250
3	1	1	female	35.0	53.1000
4	0	3	male	35.0	8.0500

In [6]:

```
df1.isnull().sum()
```

Out[6]:

```
Survived      0
Pclass        0
Sex           0
Age         177
Fare          0
dtype: int64
```

In [7]:

```
df1['Age'].describe()
```

Out[7]:

```
count    714.000000
mean      29.699118
std       14.526497
min        0.420000
25%       20.125000
50%       28.000000
75%       38.000000
max       80.000000
Name: Age, dtype: float64
```

In [8]:

```
df1['Age'].fillna(df1['Age'].mean(),inplace=True)
```

In [9]:

```
df1.isnull().sum()
```

Out[9]:

```
Survived    0
Pclass      0
Sex         0
Age         0
Fare        0
dtype: int64
```

In [10]:

```
l_sex_dummies=pd.get_dummies(df1['Sex'],drop_first=True)
```

In [11]:

```
df1= pd.concat([df1,l_sex_dummies],axis=1)
```

In [12]:

```
df1.head()
```

Out[12]:

	Survived	Pclass	Sex	Age	Fare	male
0	0	3	male	22.0	7.2500	1
1	1	1	female	38.0	71.2833	0
2	1	3	female	26.0	7.9250	0
3	1	1	female	35.0	53.1000	0
4	0	3	male	35.0	8.0500	1

In [13]:

```
df1.drop(['Sex'], axis=1, inplace=True )
```

In [14]:

```
df1.head()
```

Out[14]:

	Survived	Pclass	Age	Fare	male
0	0	3	22.0	7.2500	1
1	1	1	38.0	71.2833	0
2	1	3	26.0	7.9250	0
3	1	1	35.0	53.1000	0
4	0	3	35.0	8.0500	1

In [15]:

```
from sklearn.preprocessing import StandardScaler
sts =StandardScaler()
```

In [16]:

```
feature_scale = ['Age','Fare']
df1[feature_scale] = sts.fit_transform(df1[feature_scale])
```

In [17]:

```
df1.head()
```

Out[17]:

	Survived	Pclass	Age	Fare	male
0	0	3	-0.592481	-0.502445	1
1	1	1	0.638789	0.786845	0
2	1	3	-0.284663	-0.488854	0
3	1	1	0.407926	0.420730	0
4	0	3	0.407926	-0.486337	1

In [18]:

```
X=df1.drop(['Survived'],axis=1)
y=df1['Survived']
```

In [19]:

```
from sklearn.model_selection import GridSearchCV
from sklearn.tree import DecisionTreeClassifier
from sklearn.neighbors import KNeighborsClassifier
from sklearn.svm import SVC
```

In [20]:

```
#create param
model_param = {
    'DecisionTreeClassifier':{
        'model':DecisionTreeClassifier(),
        'param':{
            'criterion': ['gini','entropy']
        }
    },
        'KNeighborsClassifier':{
        'model':KNeighborsClassifier(),
        'param':{
            'n_neighbors': [5,10,15,20,25]
        }
    },
        'SVC':{
        'model':SVC(),
        'param':{
            'kernel':['rbf','linear','sigmoid'],
            'C': [0.1, 1, 10, 100]

        }
    }
}
```

In [21]:

```
scores =[]
for model_name, mp in model_param.items():
    model_selection = GridSearchCV(estimator=mp['model'],param_grid=mp['param'],cv=5,return
    model_selection.fit(X,y)
    scores.append({
        'model': model_name,
        'best_score': model_selection.best_score_,
        'best_params': model_selection.best_params_
    })
```

In [22]:

```
df_model_score = pd.DataFrame(scores,columns=['model','best_score','best_params'])
df_model_score
```

Out[22]:

	model	best_score	best_params
0	DecisionTreeClassifier	0.773360	{'criterion': 'gini'}
1	KNeighborsClassifier	0.803616	{'n_neighbors': 5}
2	SVC	0.811481	{'C': 100, 'kernel': 'rbf'}

In [23]:

```
model_svc = SVC( C= 100,kernel='rbf')
```

In [24]:

```
model_svc.fit(X, y)
```

Out[24]:

```
SVC(C=100)
```

In [25]:

```
df2 = pd.read_csv('test.csv')
```

In [26]:

```
df2.head()
```

Out[26]:

	PassengerId	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare	Cabin	Emba
0	892	3	Kelly, Mr. James	male	34.5	0	0	330911	7.8292	NaN	
1	893	3	Wilkes, Mrs. James (Ellen Needs)	female	47.0	1	0	363272	7.0000	NaN	
2	894	2	Myles, Mr. Thomas Francis	male	62.0	0	0	240276	9.6875	NaN	
3	895	3	Wirz, Mr. Albert	male	27.0	0	0	315154	8.6625	NaN	
4	896	3	Hirvonen, Mrs. Alexander (Helga E Lindqvist)	female	22.0	1	1	3101298	12.2875	NaN	

In [27]:

```
df3=df2.drop(['PassengerId','Name','Ticket','Cabin','Embarked','SibSp','Parch'], axis=1 )
```

In [28]:

```
df3.isnull().sum()
```

Out[28]:

```
Pclass     0
Sex        0
Age       86
Fare       1
dtype: int64
```

In [29]:

```
df3['Age'].fillna(df3['Age'].mean(),inplace=True)
df3['Fare'].fillna(df3['Fare'].mean(),inplace=True)
```

In [30]:

```
l_sex_dummies=pd.get_dummies(df3['Sex'],drop_first=True)
df3= pd.concat([df3,l_sex_dummies],axis=1)
df3.drop(['Sex'], axis=1, inplace=True )
```

In [31]:

```
df3.head()
```

Out[31]:

	Pclass	Age	Fare	male
0	3	34.5	7.8292	1
1	3	47.0	7.0000	0
2	2	62.0	9.6875	1
3	3	27.0	8.6625	1
4	3	22.0	12.2875	0

In [32]:

```
df3[feature_scale] = sts.fit_transform(df3[feature_scale])
```

In [33]:

```
df3.head()
```

Out[33]:

	Pclass	Age	Fare	male
0	3	0.334993	-0.498407	1
1	3	1.325530	-0.513274	0
2	2	2.514175	-0.465088	1
3	3	-0.259330	-0.483466	1
4	3	-0.655545	-0.418471	0

In [34]:

```
y_predicted = model_svc.predict(df3)
```

In [35]:

```
submission = pd.DataFrame({
        "PassengerId": df2['PassengerId'],
        "Survived": y_predicted
    })
```

In [36]:

```

submission.to_csv('titanic_submission.csv', index=False)
```

In [ ]:

```

```