In [1]:

```
import pandas as pd
import seaborn as sb
from sklearn.model_selection import train_test_split, GridSearchCV
from sklearn.preprocessing import MinMaxScaler
from sklearn.linear_model import LogisticRegression
from sklearn.neighbors import KNeighborsClassifier
from sklearn.svm import SVC
from sklearn.ensemble import RandomForestClassifier
```

In [2]:

```
df_train = pd.read_csv('../datasets/Titanic train.csv')
df_test = pd.read_csv('../datasets/Titanic test.csv')
```

In [3]:

```
df_train.head()
```

Out[3]:

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

In [4]:

```
df_train.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 891 entries, 0 to 890
Data columns (total 12 columns):
 #   Column       Non-Null Count  Dtype  
---  ------       --------------  -----  
 0   PassengerId  891 non-null    int64  
 1   Survived     891 non-null    int64  
 2   Pclass       891 non-null    int64  
 3   Name         891 non-null    object 
 4   Sex          891 non-null    object 
 5   Age          714 non-null    float64
 6   SibSp        891 non-null    int64  
 7   Parch        891 non-null    int64  
 8   Ticket       891 non-null    object 
 9   Fare         891 non-null    float64
 10  Cabin        204 non-null    object
 11  Embarked     889 non-null    object
dtypes: float64(2), int64(5), object(5)
memory usage: 83.7+ KB
```

In [5]:

```
df_train.isna().sum()
```

Out[5]:

```
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 [6]:

```
df_train.columns
```

Out[6]:

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

In [7]:

```
selected_cols = [ 'Pclass', 'Sex', 'Age', 'SibSp', 'Parch', 'Fare' ]
```

In [8]:

```
df_train[selected_cols].info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 891 entries, 0 to 890
Data columns (total 6 columns):
 #   Column  Non-Null Count  Dtype  
---  ------  --------------  -----  
 0   Pclass  891 non-null    int64  
 1   Sex     891 non-null    object 
 2   Age     714 non-null    float64
 3   SibSp   891 non-null    int64  
 4   Parch   891 non-null    int64  
 5   Fare    891 non-null    float64
dtypes: float64(2), int64(3), object(1)
memory usage: 41.9+ KB
```

In [9]:

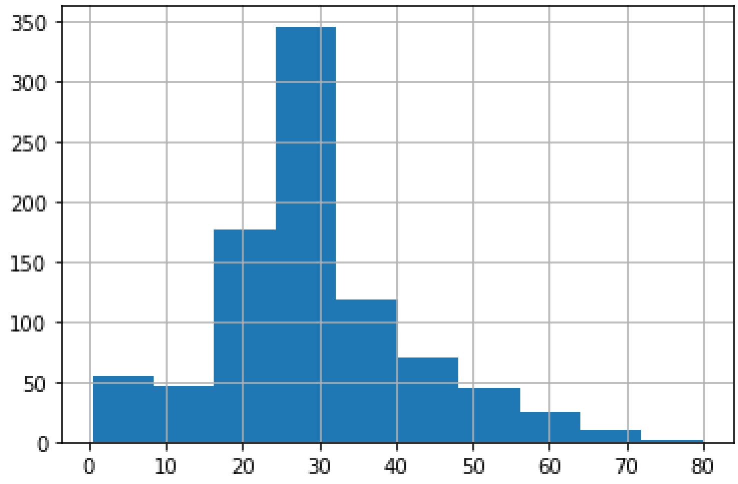
```
df_test[selected_cols].info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 418 entries, 0 to 417
Data columns (total 6 columns):
 #   Column  Non-Null Count  Dtype  
---  ------  --------------  -----  
 0   Pclass  418 non-null    int64  
 1   Sex     418 non-null    object 
 2   Age     332 non-null    float64
 3   SibSp   418 non-null    int64  
 4   Parch   418 non-null    int64  
 5   Fare    417 non-null    float64
dtypes: float64(2), int64(3), object(1)
memory usage: 19.7+ KB
```

In [36]:

```
df_train['Age'].hist()
```

Out[36]: <AxesSubplot:>



In [10]:

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

In [11]:

```
df_train['Fare'].fillna(df_train['Fare'].mean(), inplace=True)
df_test['Fare'].fillna(df_train['Fare'].mean(), inplace=True)
```

In [12]:

```
df_train[selected_cols].isna().sum()
```

Out[12]:

```
Pclass    0
Sex       0
Age       0
SibSp     0
Parch     0
Fare      0
dtype: int64
```

In [13]:

```
df_test[selected_cols].isna().sum()
```

Out[13]:

```
Pclass    0
Sex       0
Age       0
SibSp     0
Parch     0
Fare      0
dtype: int64
```

In [14]:

```
df_train['Sex']=df_train['Sex'].map({'male' :1, 'female' :0})
df_test['Sex']=df_test['Sex'].map({'male' :1, 'female' :0})
```

In [15]:

```
df_train[selected_cols].info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 891 entries, 0 to 890
Data columns (total 6 columns):
 #   Column  Non-Null Count  Dtype  
---  ------  --------------  -----  
 0   Pclass  891 non-null    int64  
 1   Sex     891 non-null    int64  
 2   Age     891 non-null    float64
 3   SibSp   891 non-null    int64  
 4   Parch   891 non-null    int64  
 5   Fare    891 non-null    float64
dtypes: float64(2), int64(4)
memory usage: 41.9 KB
```

In [16]:

```
df_test[selected_cols].info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 418 entries, 0 to 417
Data columns (total 6 columns):
 #   Column  Non-Null Count  Dtype
---  ------  --------------  -----
 0   Pclass  418 non-null    int64
 1   Sex     418 non-null    int64
 2   Age     418 non-null    float64
 3   SibSp   418 non-null    int64
 4   Parch   418 non-null    int64
 5   Fare    418 non-null    float64
dtypes: float64(2), int64(4)
memory usage: 19.7 KB
```

In [17]:

```
X = df_train[selected_cols]
y = df_train['Survived']
```

In [18]:

```
X_train, X_val, y_train, y_val=train_test_split(X,y,test_size= .20,random_state=100)
```

In [19]:

```
scaler = MinMaxScaler()
scaler.fit(X_train)
X_train_scaler = scaler.transform(X_train)
X_val_scaler = scaler.transform(X_val)
```

In [20]:

```
model_LR = LogisticRegression()
model_KNN = KNeighborsClassifier(n_neighbors=5)
model_lin = SVC(kernel= 'linear')
model_poly = SVC(kernel= 'poly')
model_rbf = SVC(kernel= 'rbf')
model_rf = RandomForestClassifier(n_estimators=10,random_state=1)
```

In [24]:

```
models = {'LR':model_LR,'KNN':model_KNN,'SVM_Lin':model_lin,'SVM_Poly':model_poly,
          'SVM_RBF':model_rbf, 'RF':model_rf}

for name,model in models.items():
    model.fit(X_train_scaler,y_train)
    print(name,round(model.score(X_train_scaler,y_train),2), round(model.score(X_val
```

```
LR 0.8 0.8
KNN 0.86 0.8
SVM_Lin 0.79 0.79
SVM_Poly 0.82 0.8
SVM_RBF 0.82 0.8
RF 0.97 0.77
```

In [25]:

```
params = {'n_estimators':[10,20,30,40,50], 'max_depth':[2,3,4,5,6],'min_samples_leaf
grid_cv = GridSearchCV(RandomForestClassifier(),param_grid=params, cv=5, n_jobs=-1)
```

In [26]:

```
grid_cv.fit(df_train[selected_cols],df_train['Survived'])
```

Out[26]:

```
GridSearchCV(cv=5, estimator=RandomForestClassifier(), n_jobs=-1,
                        param_grid={'max_depth': [2, 3, 4, 5, 6],
                                    'min_samples_leaf': [2, 3, 4, 5, 6],
                                    'n_estimators': [10, 20, 30, 40, 50]})
```

In [27]:

```
model_final = grid_cv.best_estimator_
model_final.fit(df_train[selected_cols],df_train['Survived'])
```

Out[27]:

```
RandomForestClassifier(max_depth=6, min_samples_leaf=2, n_estimators=20)
```

In [28]:

```
RandomForestClassifier(max_depth=6, min_samples_leaf=2, n_estimators=30)
```

Out[28]:

```
RandomForestClassifier(max_depth=6, min_samples_leaf=2, n_estimators=30)
```

In [29]:

```
y_pre = model_final.predict(df_test[selected_cols])
```

In [30]:

```
df_submit = pd.DataFrame({'PassengerId':df_test['PassengerId'], 'Survived':y_pre})
```

In [31]:

```
df_submit.head()
```

Out[31]:

	PassengerId	Survived
0	892	0
1	893	0
2	894	0
3	895	0
4	896	1

In [32]:

```
df_submit.to_csv('submit1.csv', index=False)
```

In [ ]: