Ansh Awasthi IT A 3rd Year 2100290139002

In [1]:
```
import pandas as pd
import numpy as np
from sklearn import preprocessing
from  sklearn.naive_bayes  import  GaussianNB
from sklearn.tree import DecisionTreeClassifier
from sklearn.neighbors import KNeighborsClassifier
from sklearn.metrics import accuracy_score
from sklearn.model_selection import train_test_split
```

In [2]:
```
# Load the dataset
#df=pd.read_csv('D:/datasets_ML/gender_submission.csv')
df=pd.read_csv('E:/Downloads/TITANIC/train.csv')
df.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 [3]:
```
df.describe()
```

Out[3]:

	PassengerId	Survived	Pclass	Age	SibSp	Parch	Fare
count	891.000000	891.000000	891.000000	714.000000	891.000000	891.000000	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

In [4]:
```
# Preprocess the data
df = df.drop(['PassengerId', 'Name', 'Ticket', 'Cabin'], axis=1)
df = pd.get_dummies(df, columns=['Sex', 'Embarked'], drop_first=True)
df = df.fillna(df.mean())
```

In [5]:
```
# Split the data into training and testing sets
X = df.drop('Survived', axis=1)
y = df['Survived']
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)
```

In [6]:
```
# Naive Bayes classifier
nb = GaussianNB()
nb.fit(X_train, y_train)
nb_pred = nb.predict(X_test)
nb_acc = accuracy_score(y_test, nb_pred)
print('Naive Bayes accuracy:', nb_acc)
```

```
Naive Bayes accuracy: 0.770949720670391
```

In [7]:
```
# J48 classifier
j48 = DecisionTreeClassifier()
j48.fit(X_train, y_train)
j48_pred  =  j48.predict(X_test)
j48_acc = accuracy_score(y_test, j48_pred)
print('J48 accuracy:', j48_acc)
```

```
J48  accuracy:  0.7821229050279329
```

In [8]:
```
# KNN classifier
knn = KNeighborsClassifier(n_neighbors=5)
knn.fit(X_train, y_train)
knn_pred = knn.predict(X_test)
```

```
        knn_acc = accuracy_score(y_test, knn_pred)
        print('KNN accuracy:', knn_acc)
```

```
KNN  accuracy:  0.7039106145251397
```

```
In [9]: print('Naive Bayes accuracy:', nb_acc)
        print('J48 accuracy:', j48_acc)
        print('KNN accuracy:', knn_acc)
```

```
Naive Bayes accuracy: 0.770949720670391
J48 accuracy: 0.7821229050279329
KNN accuracy: 0.7039106145251397
```