```python
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns

from sklearn.model_selection import train_test_split
from sklearn.ensemble import RandomForestClassifier
from sklearn.metrics import accuracy_score, classification_report,
confusion_matrix

#Dataset
df = pd.read_csv("tested.csv")


print("Initial Data Overview:\n")
print(df.info())
print("\nMissing Values:\n", df.isnull().sum())
print("\nSample Rows:")
display(df.head())
```

```
Initial Data Overview:

<class 'pandas.core.frame.DataFrame'>
RangeIndex: 418 entries, 0 to 417
Data columns (total 12 columns):
 #   Column       Non-Null Count  Dtype
---  ------       --------------  -----
 0   PassengerId  418 non-null    int64
 1   Survived     418 non-null    int64
 2   Pclass       418 non-null    int64
 3   Name         418 non-null    object
 4   Sex          418 non-null    object
 5   Age          332 non-null    float64
 6   SibSp        418 non-null    int64
 7   Parch        418 non-null    int64
 8   Ticket       418 non-null    object
 9   Fare         417 non-null    float64
 10  Cabin        91 non-null     object
 11  Embarked     418 non-null    object
dtypes: float64(2), int64(5), object(5)
memory usage: 39.3+ KB
None

Missing Values:
 PassengerId      0
Survived          0
Pclass            0
Name              0
Sex               0
Age              86
```

```
SibSp            0
Parch            0
Ticket           0
Fare             1
Cabin          327
Embarked         0
dtype: int64

Sample Rows:

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

                                          Name     Sex   Age  SibSp
Parch  \
0                              Kelly, Mr. James    male  34.5      0
0
1              Wilkes, Mrs. James (Ellen Needs)  female  47.0      1
0
2                     Myles, Mr. Thomas Francis    male  62.0      0
0
3                              Wirz, Mr. Albert    male  27.0      0
0
4  Hirvonen, Mrs. Alexander (Helga E Lindqvist)  female  22.0      1
1

    Ticket     Fare Cabin Embarked
0   330911   7.8292   NaN        Q
1   363272   7.0000   NaN        S
2   240276   9.6875   NaN        Q
3   315154   8.6625   NaN        S
4  3101298  12.2875   NaN        S
```

```
df['Age'] = df['Age'].fillna(df['Age'].median())
df['Fare'] = df['Fare'].fillna(df['Fare'].median())

df['Sex'] = df['Sex'].map({'male': 0, 'female': 1})
df['Embarked'] = df['Embarked'].map({'S': 0, 'C': 1, 'Q': 2})

df.drop(['PassengerId', 'Name', 'Ticket', 'Cabin'], axis=1,
inplace=True)

print("\nData after cleaning and preprocessing:")
display(df.head())
```

```
Data after cleaning and preprocessing:
```

```
   Survived  Pclass  Sex   Age  SibSp  Parch     Fare  Embarked
0         0       3    0  34.5      0      0   7.8292         2
1         1       3    1  47.0      1      0   7.0000         0
2         0       2    0  62.0      0      0   9.6875         2
3         0       3    0  27.0      0      0   8.6625         0
4         1       3    1  22.0      1      1  12.2875         0
```

```
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)

model = RandomForestClassifier(n_estimators=100, random_state=42)
model.fit(X_train, y_train)
```

```
RandomForestClassifier(random_state=42)
```

```
y_pred = model.predict(X_test)

print(" Model Accuracy:", accuracy_score(y_test, y_pred))
print("\n Classification Report:\n", classification_report(y_test,
y_pred))
print("\n Confusion Matrix:\n", confusion_matrix(y_test, y_pred))
```

```
 Model Accuracy: 1.0

 Classification Report:
               precision    recall  f1-score   support

           0       1.00      1.00      1.00        50
           1       1.00      1.00      1.00        34

    accuracy                           1.00        84
   macro avg       1.00      1.00      1.00        84
weighted avg       1.00      1.00      1.00        84


 Confusion Matrix:
 [[50  0]
 [ 0 34]]
```