

BHARAT DATA SCIENCE INTERNSHIP

TASK-2: Titanic Classification

Make a system which tells whether the person will be save from sinking. What factors were most likely lead to success-socio-economic status, age, gender and more.

```
In [2]: # Import necessary libraries
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import StandardScaler
from sklearn.ensemble import RandomForestClassifier
from sklearn.metrics import accuracy_score, classification_report
```

```
In [3]: # Load the Titanic dataset
data = pd.read_csv(r"C:\Users\harsh\Downloads\titanic_dataset.csv")
```

```
In [4]: # Data Preprocessing
# Drop irrelevant columns and handle missing values
data.drop(['PassengerId', 'Name', 'Ticket', 'Cabin'], axis=1, inplace=True)
data['Age'].fillna(data['Age'].median(), inplace=True)
data['Embarked'].fillna(data['Embarked'].mode()[0], inplace=True)
data.dropna(subset=['Fare'], inplace=True)
```

```
In [5]: # Encoding categorical variables
data = pd.get_dummies(data, columns=['Sex', 'Embarked'], drop_first=True)
```

```
In [6]: # Split the data into features (X) and target variable (y)
X = data.drop('Survived', axis=1)
y = data['Survived']
```

```
In [7]: # Split the data into training and testing sets
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)
```

```
In [8]: # Feature Scaling
scaler = StandardScaler()
X_train = scaler.fit_transform(X_train)
X_test = scaler.transform(X_test)
```

```
In [9]: # Train a Random Forest Classifier
clf = RandomForestClassifier(n_estimators=100, random_state=42)
clf.fit(X_train, y_train)
```

```
Out[9]:
RandomForestClassifier
RandomForestClassifier(random_state=42)
```

```
In [10]: # Make predictions on the testing set
y_pred = clf.predict(X_test)
```

```
In [11]: # Evaluate the model
accuracy = accuracy_score(y_test, y_pred)
print(f'Accuracy: {accuracy:.2f}')
```

Accuracy: 0.82

```
In [12]: # Generate a classification report
report = classification_report(y_test, y_pred)
print('Classification Report:')
print(report)
```

```
Classification Report:
              precision    recall  f1-score   support

     0       0.83         0.87         0.85         105
     1       0.80         0.76         0.78          74

 accuracy          0.82         0.82         0.82         179
  macro avg       0.82         0.81         0.81         179
 weighted avg     0.82         0.82         0.82         179
```