**Name: Muhammad Zeeshan**

**Roll no: 064**

**Section NO: BSAI-4B**

**Code:**

**import numpy as np**

**import pandas as pd**

**import os**

**from sklearn.model\_selection import train\_test\_split**

**from sklearn.preprocessing import LabelEncoder, StandardScaler**

**from sklearn.ensemble import RandomForestClassifier**

**from sklearn.metrics import accuracy\_score**

**train\_df = pd.read\_csv(r"C:\Users\ZEESHAN\Desktop\PAI-Lab\train.csv")**

**test\_df = pd.read\_csv(r"C:\Users\ZEESHAN\Desktop\PAI-Lab\test.csv")**

**encoders = {}**

**def preprocess\_data(df, is\_train=True):**

**df.fillna(method='ffill', inplace=True)**

**df[['Deck', 'Num', 'Side']] = df['Cabin'].fillna("Unknown/0/Unknown").str.split('/', expand=True)**

**df.drop(columns=['Name', 'PassengerId', 'Cabin'], inplace=True, errors='ignore')**

**categorical\_features = df.select\_dtypes(include=['object']).columns**

**for col in categorical\_features:**

**if is\_train:**

**encoders[col] = LabelEncoder()**

**df[col] = encoders[col].fit\_transform(df[col].astype(str))**

**else:**

**df[col] = df[col].astype(str).map(lambda x: encoders[col].transform([x])[0] if x in encoders[col].classes\_ else -1)**

**return df**

**train\_df = preprocess\_data(train\_df, is\_train=True)**

**test\_df = preprocess\_data(test\_df, is\_train=False)**

**X = train\_df.drop(columns=['Transported'])**

**y = train\_df['Transported']**

**X\_train, X\_val, y\_train, y\_val = train\_test\_split(X, y, test\_size=0.2, random\_state=42)**

**scaler = StandardScaler()**

**num\_features = X\_train.select\_dtypes(include=['number']).columns**

**X\_train[num\_features] = scaler.fit\_transform(X\_train[num\_features])**

**X\_val[num\_features] = scaler.transform(X\_val[num\_features])**

**test\_df[num\_features] = scaler.transform(test\_df[num\_features])**

**# Train Model**

**model = RandomForestClassifier(n\_estimators=100, random\_state=42)**

**model.fit(X\_train, y\_train)**

**# Evaluate Model**

**y\_pred = model.predict(X\_val)**

**accuracy = accuracy\_score(y\_val, y\_pred)**

**print(f'Validation Accuracy: {accuracy:.4f}')**

**# Make Predictions**

**test\_predictions = model.predict(test\_df)**

**# Create Submission File**

**submission = pd.DataFrame({**

**"PassengerId": pd.read\_csv(r"C:\Users\ZEESHAN\Desktop\PAI-Lab\test.csv")["PassengerId"],**

**"Transported": test\_predictions**

**})**

**submission.to\_csv("submission.csv", index=False)**

**print("Submission file saved successfully!")**

The code starts by loading the training and test datasets. It handles missing values by forward-filling them and extracts details from the Cabin column, splitting it into separate features. Unnecessary columns like Name, PassengerId, and Cabin are removed. Categorical features are converted into numerical values using LabelEncoder. The dataset is then split into training and validation sets, and numerical features are standardized using StandardScaler to improve model performance. A RandomForestClassifier is trained on the processed training data and tested on the validation set, with accuracy used to measure its performance. Once trained, the model makes predictions on the test dataset. The results, including passenger IDs and predicted transport status, are saved in a CSV file for submission.

Accuraccy:

