SUPERIOR UNIVERSITY

Programming For Artificial Intelligence

Assignment - 2

Name:

Ali Maqsood.

Roll no:

SU92-BSAIM-F23-050.

Department:

Software Engineering Department.

Program:

Artificial Intelligence.

Section:

BSAI-4A

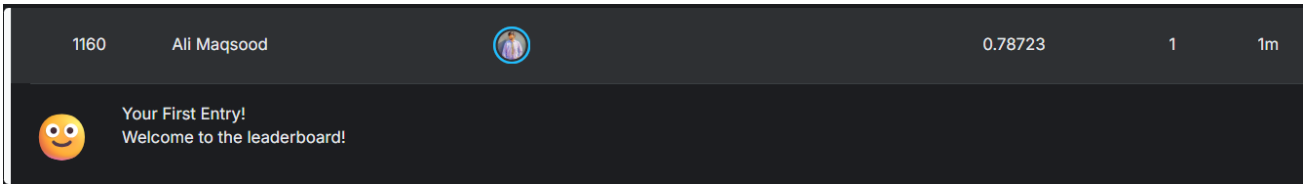
Question # 1:

Kaggle Competition: Spaceship Titanic Passenger Problem.

Documentation:

This code is solving the **Spaceship Titanic prediction problem** a Kaggle challenge using a Random Forest model. It first loads *train.csv* and *test.csv*, drops the columns that are not needed i.e., *Passenger Id* and *Name*, and handles missing values by filling *Home Planet, Cabin,* and *Destination* with "Unknown". The target column "*Transported*" is separated from the training features. The *Cabin* column is simplified to just its first letter deck as done in the last problem. Then, categorical features (*Home Planet, Cabin, Destination*) are encoded into numbers for further process using Label Encoder. A Random Forest Classifier is trained on the processed training data and used to predict whether passengers in the test set were *Transported*. In the end, the predictions are written to *submission.csv* alongside passenger IDs for identification. This file than is submitted to Kaggle for ranking.

Ranking:



Code:

```
import csv
import pandas as pd
from sklearn.preprocessing import LabelEncoder
from sklearn.ensemble import RandomForestClassifier

train=pd.read_csv("train.csv")
test=pd.read_csv("test.csv")

train.drop(columns=["PassengerId","Name"],inplace=True)
test.drop(columns=["PassengerId","Name"],inplace=True)

# Filling missing values
train["HomePlanet"].fillna("Unknown", inplace=True)
test["HomePlanet"].fillna("Unknown", inplace=True)

train["Cabin"].fillna("Unknown", inplace=True)
test["Cabin"].fillna("Unknown", inplace=True)

train["Destination"].fillna("Unknown", inplace=True)
test["Destination"].fillna("Unknown", inplace=True)

# Dropping output columns
train_x=train.drop(columns=["Transported"])
train_y=train["Transported"]

# converting the cabin into deck only
```

```
train_x["Cabin"]=train_x["Cabin"].str[0]
test["Cabin"]=test["Cabin"].str[0]

# Encoding categorical features
le_home=LabelEncoder()
le_cabin=LabelEncoder()
le_dest=LabelEncoder()
train_x["HomePlanet"]=le_home.fit_transform(train_x["HomePlanet"])
test["HomePlanet"]=le_home.transform(test["HomePlanet"])

train_x["Cabin"]=le_cabin.fit_transform(train_x["Cabin"])
test["Cabin"]=le_cabin.transform(test["Cabin"])

train_x["Destination"]=le_dest.fit_transform(train_x["Destination"])
test["Destination"]=le_dest.transform(test["Destination"])

model=RandomForestClassifier(random_state=42)
model.fit(train_x,train_y)
predictions=model.predict(test)

psid=pd.read_csv("test.csv")["PassengerId"]
with open("submission.csv","w",newline="") as f:
    writer=csv.writer(f)
    writer.writerow(["PassengerId","Transported"])
    for i in range(len(predictions)):
        writer.writerow([psid[i],predictions[i]])
```