SUPERIOR UNIVERSITY

Programming For Artificial Intelligence

Assignment - 1

Name:

Ali Maqsood.

Roll no:

SU92-BSAIM-F23-050.

Department:

Software Engineering Department.

Program:

Artificial Intelligence.

Section:

BSAI-4A

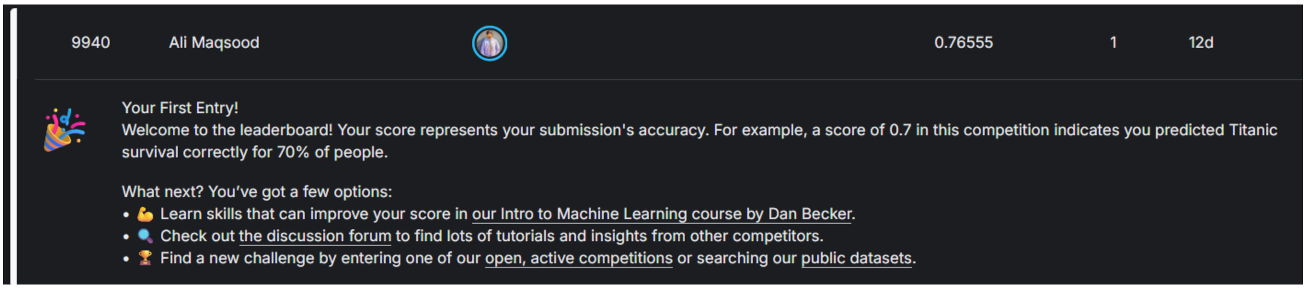
Question # 1:

Kaggle Competition: Titanic Passenger Survival Prediction.

Documentation:

This code is solving the **Titanic Passenger Survival Prediction** using a Random Forest model and dataset taken from **Kaggle Competition**. It first loads the *train.csv* and *test.csv* files, removes unnecessary columns like *Passenger Id, Ticket, Name*, and separates features “x_train” from the target Survived as “y_train”. Missing values are handled by filling *Embarked* with the most common value and *Age* with the median, while *Cabin* is simplified into just its deck letter with missing ones labelled “Missing”. Categorical features (*Embarked*, *Cabin Deck*, *Sex*) are encoded into numbers using Label Encoder. A Random Forest Classifier is then trained on the processed training data and used to predict survival for the test data. In the end, the predictions are written to *submission.csv* along with each passenger’s ID for submission and the submitted on the Kaggle competition for ranking.

Ranking:



9940 Ali Maqsood 0.76555 1 12d

Your First Entry!
Welcome to the leaderboard! Your score represents your submission's accuracy. For example, a score of 0.7 in this competition indicates you predicted Titanic survival correctly for 70% of people.

What next? You’ve got a few options:

- 💪 Learn skills that can improve your score in our Intro to Machine Learning course by Dan Becker.
- 🔍 Check out the discussion forum to find lots of tutorials and insights from other competitors.
- 🏆 Find a new challenge by entering one of our open, active competitions or searching our public datasets.

Code:

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

train=pd.read_csv("train.csv")
train=train.drop(["PassengerId","Ticket","Name"],axis=1)
test=pd.read_csv("test.csv")
test=test.drop(["PassengerId","Ticket","Name"],axis=1)

y_train=train["Survived"]
x_train=train.drop(["Survived"],axis=1)
x_test=test.copy()

# Filling missing values in Embarked
x_train["Embarked"]=x_train["Embarked"].fillna(x_train["Embarked"].mode()[0])
x_test["Embarked"]=x_test["Embarked"].fillna(x_test["Embarked"].mode()[0])

# Filling missing values in Age
x_train["Age"]=x_train["Age"].fillna(x_train["Age"].median())
x_test["Age"]=x_test["Age"].fillna(x_test["Age"].median())

# Changing Cabin to Cabin desk and filling missing values
x_train["CabinDeck"]=x_train["Cabin"].astype(str).str[0]
x_train["CabinDeck"]=x_train["CabinDeck"].replace("n","Missing")
x_train=x_train.drop("Cabin",axis=1)
```

```
x_test["CabinDeck"]=x_test["Cabin"].astype(str).str[0]
x_test["CabinDeck"]=x_test["CabinDeck"].replace("n","Missing")
x_test=x_test.drop("Cabin",axis=1)

le=LabelEncoder()
# Encoding categorical features
x_train["Embarked"]=le.fit_transform(x_train["Embarked"])
x_test["Embarked"]=le.transform(x_test["Embarked"])

x_train["CabinDeck"]=le.fit_transform(x_train["CabinDeck"])
x_test["CabinDeck"]=le.transform(x_test["CabinDeck"])

x_train["Sex"]=le.fit_transform(x_train["Sex"])
x_test["Sex"]=le.transform(x_test["Sex"])

rf_model=RandomForestClassifier(random_state=42)
rf_model.fit(x_train,y_train)
prediction=rf_model.predict(x_test)

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