****

***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

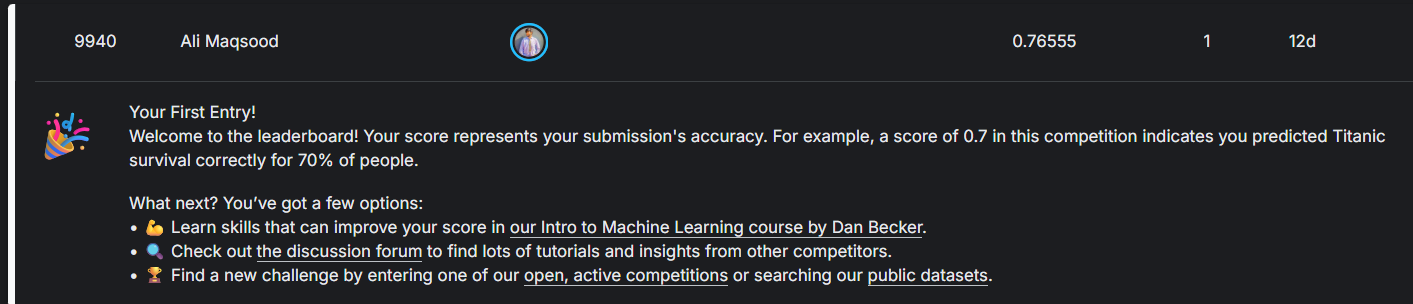
**Question # 1:**

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:**



**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]])