KAGGLE COMPETITION

Andrew-ID: parmenin

DATA INFERENCE AND APPLIED MACHINE LEARNING (18-785)

12/9/22

Niyomwungeri Parmenide ISHIMWE

I, the undersigned, have read the entire contents of the syllabus for course 18-785 (Data Inference and Applied Machine Learning) and agree with the terms and conditions of participating in this course, including adherence to CMU's AIV policy.

Signature: **Niyomwungeri Parmenide ISHIMWE**
Andrew ID: **parmenin**
Full Name: **Niyomwungeri Parmenide ISHIMWE**

The libraries used:

- import numpy as np
- import pandas as pd
- import seaborn as sb
- import statsmodels.api as sm
- from sklearn.preprocessing import LabelEncoder
- from sklearn.linear_model import LinearRegression, LogisticRegression
- from sklearn.neighbors import KNeighborsClassifier
- from sklearn.ensemble import RandomForestClassifier, RandomForestRegressor
- from pathlib import Path
- from sklearn.tree import DecisionTreeClassifier

KAGGLE COMPETITION REPORT

Here are the steps taken to find the best model for the Kaggle competition.

- Reading of both datasets train.csv, and test.csv
- Filling the nan values with the mean of the age because it is the only one with missing data in the training dataset.
- Changing the male into 1 and the female into 0 in column “Sex” using LabelEncoder () function.
- Using the P-class, Sex, and Age as the training and testing sets for X.
- Using the Survived as the training set on Y.
- Fitting the models (LogisticRegression, KNeighborsClassifier, RandomForestClassifier, DecisionTreeClassifier, RandomForestRegressor, using the xTrain and the yTrain datasets.
- Predict using the xTest.
- Converting the prediction results into a data frame and adding the PassengerId.
- Generating an output CSV file to submit on the Kaggle website.

In detail, the random forest regressor gives a score of 0, the decision tree model gives a 0.77033 score, the KNN model gives 0.71291, the Logistic regression gives 0.75837, and finally, the random forest gives a score of 0.78468 which is the best score compared to other models. Therefore the random forest classifier model (**RandomForestClassifier(n_estimators = 500, max_depth = 5, max_features = 5, n_jobs = 1).fit(xTrain, yTrain))** would be the best model for classifying survival on the Titanic.

Here is the results snippet from the Kaggle website.

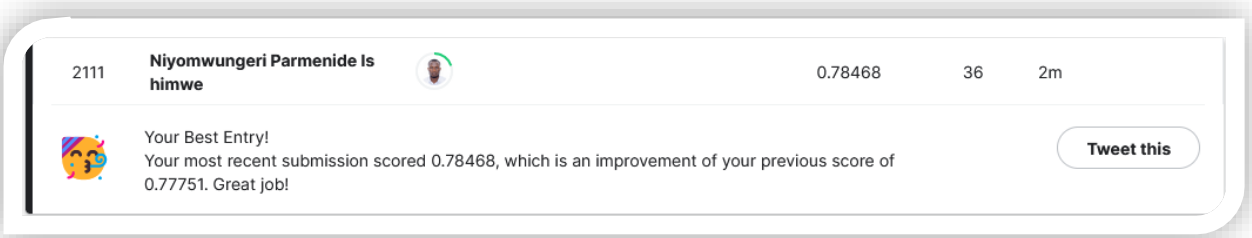

Figure: Final score for the random forest model