CS 491/591
Introduction to Machine Learning

Assignment 1

Due by 11:59PM, Friday, **November 10**

NOTE: Assignment is to be done in your assigned teams. Do not look at another team's answer and do not allow another team to look at your answer. Doing so will result in charges of academic misconduct. Email your submission to me (atkison@cs.ua.edu). Only one submission is required per team.

Welcome to the year 2920! Your expertise is urgently required to decipher a space-age mystery. We've intercepted a signal from a distance of four and a half lightyears, indicating a concerning situation.

The Galactic Voyager, an intergalactic cruiser, set out on its first journey a few weeks prior. Transporting around 13,000 settlers, its mission was to move citizens from our galaxy to a trio of newly discovered planets surrounding distant suns.

While navigating near Alpha Centauri on its way to its primary destination, the torrid LHS 3844 b, the unsuspecting Galactic Voyager encountered a spacetime distortion concealed within a nebula. Tragically, its journey echoed that of a legendary vessel from a millennium ago. Although the Voyager remained unharmed, a significant number of its travelers found themselves shifted to a different realm!

To assist the search teams and recover the missing travelers, you are tasked with forecasting which individuals were displaced by the disturbance using data salvaged from the ship's compromised database.

Dataset Description:

In this assignment, your task is to predict whether a traveler was displaced to this alternate dimension during the Galactic Voyager's encounter with the spacetime disturbance. To guide your predictions, you're provided with a collection of personal records salvaged from the ship's compromised database.

- **train.csv**: Contains personal details of approximately two-thirds (around 8700) of the travelers, intended for training purposes.
    - **PassengerId**: A distinct identifier for every traveler. The format is gggg_pp, where 'gggg' denotes a group the traveler is associated with, and 'pp' signifies their position within that group. Group members are often related but not always.
    - **HomePlanet**: Represents the planet from which the traveler embarked, usually their primary residence.
    - **CryoSleep**: Specifies if the traveler chose suspended animation for the journey's duration. Those in cryosleep remain in their rooms.
    - **Cabin**: The room number assigned to the traveler. The format is deck/num/side, with 'side' being either 'P' (Port) or 'S' (Starboard).
    - **Destination**: The planet the traveler intends to disembark at.
    - **Age**: The traveler's age.
    - **VIP**: Indicates if the traveler availed the exclusive VIP services during the journey.
    - **RoomService, FoodCourt, ShoppingMall, Spa, VRDeck**: Represents the billed amount by the traveler at each of the Galactic Voyager's luxury facilities.
    - **Name**: The traveler's full name.
    - **Transported**: Determines if the traveler was shifted to a different dimension. This is the prediction target.
- **test.csv**: Contains personal details of the remaining one-third (around 4300) of the travelers, intended for testing. Your objective is to forecast the 'Transported' value for these travelers.
- **sample_submission.csv**: A template for the submission file.
    - **PassengerId**: Identifier for each traveler in the test dataset.
    - **Transported**: The prediction target. For every traveler, forecast either 'True' or 'False'.

Submission Guidelines:

Upon completing your analysis and predictions, you are required to submit the following:

1. **sample_submission_format.csv**: This file should contain your predictions. Ensure that the format matches the provided sample submission file. It should have the **PassengerId** and the corresponding **Transported** prediction for each passenger.

2. **Your Analysis Notebook (.ipynb)**: This should containing:
    - **Introduction**: A brief overview of the problem statement and your approach to solving it.
    - **Data Exploration**: Visualizations and insights from the initial dataset.
    - **Data Preprocessing**: Steps taken to clean and prepare the data for modeling.
    - **Model Selection and Training**: Explanation of why you chose a particular model, training process, and any challenges faced.
    - **Model Evaluation**: Metrics used to evaluate the model's performance and results from the evaluation.
    - **Feature Importance Analysis**: Insights on which features were most influential in the model's predictions.
    - **Conclusions**: Summary of findings, challenges, and any recommendations for future analysis.
    - **References**: Any sources or references you used during your analysis.

Note: Ensure that your notebook is well-structured with clear headings, sub-headings, and comments. It should be easy to follow, with explanations for each code block. Remember, the clarity and structure of your notebook are as important as the analysis itself. It should tell a clear story of your data science journey from understanding the problem to making predictions.

You are required to use **Google Colab** for this assignment. Additionally, it's mandatory to use **TensorFlow** and specifically the **Random Forest model** (**TFDF**).

Extra Hints for analyzing lab assignment:

1. **Introduction to Decision Forests**: Decision Forests are a collection of tree-based models, including Random Forests and Gradient Boosted Trees. They are particularly effective for tabular data and can serve as a strong baseline before diving into more complex models like neural networks.
2. **Data Loading and Exploration**:
    - Use pandas to load the dataset.
    - Explore the dataset's structure, dimensions, and basic statistics.
    - Visualize the distribution of the target variable (**Transported**) and other numerical features.
3. **Data Preprocessing**:
    - Handle missing values: While TensorFlow Decision Forests (TFDF) can natively handle missing values in categorical columns, you might need to handle missing values in numerical or boolean columns.
    - Convert boolean columns to integer format.
    - Extract information from composite columns. For example, split the **Cabin** column into **Deck**, **Cabin_num**, and **Side**.
4. **Dataset Splitting**: Divide the dataset into training and validation sets. This helps in evaluating the model's performance on unseen data.
5. **Model Selection and Training**:
    - Choose a tree-based model. For starters, a Random Forest model is recommended.
    - Train the model on the training dataset.
    - Visualize the trained decision trees to understand the decision-making process.
6. **Model Evaluation**:
    - Evaluate the model's performance using Out-of-Bag (OOB) data. OOB evaluation provides an estimate of the model's accuracy on unseen data.
    - Additionally, evaluate the model on the validation set to get a sense of its generalization capability.
7. **Feature Importance**:
    - Understand which features are most influential in making predictions. TFDF provides various methods to compute feature importance.
8. **Predictions and Submission**:
    - Prepare the test dataset in a similar manner as the training dataset.
    - Make predictions using the trained model.
    - Format the predictions as per the submission requirements.

Steps for the Lab Assignment:

1. **Setup and Data Loading**:
    - Import necessary libraries.
    - Load the dataset using pandas.
2. **Data Exploration**:
    - Check the first few rows of the dataset.
    - Visualize the distribution of the target variable and other key features.
3. **Data Preprocessing**:
    - Handle missing values.
    - Convert boolean columns to integers.
    - Extract and create new features from existing columns.
4. **Model Selection, Training, and Evaluation**:
    - Convert the pandas dataframe to TensorFlow datasets format.
    - Choose a tree-based model and train it.
    - Evaluate the model's performance on both OOB data and the validation set.
5. **Feature Importance Analysis**:
    - Identify and visualize the most important features influencing the predictions.
6. **Predictions**:
    - Prepare the test dataset.
    - Make predictions using the trained model.
    - Format and save the predictions for submission.