**Case Study -2. M.Jayasree**

1. Upload, explore, clean, and preprocess data for classification tree.

a. Create the flight\_df data frame by uploading the original data set into Python. Determine and present in this report the data frame dimensions, i.e., number of rows and columns.

A screenshot of a computer

AI-generated content may be incorrect.

**Explanation:-**

The Flight Delays dataset has been imported into a Pandas DataFrame named flight\_df. It comprises 2201 rows and 11 columns, as shown in the output.

**Understanding Dimensions:**

* Rows (2201): Each row corresponds to a distinct flight record.
* 11 columns: Each column corresponds to a specific variable or feature associated with the flight data.

b. Remove ‘DEST’ and ‘ORIGIN’ variables from the flight\_df data frame. Then, display the column data types in flight\_df, provide and briefly explain them in your report.

A screenshot of a computer code

AI-generated content may be incorrect.

**Explanation:-**

* By applying flight\_df.drop(columns=['DEST', 'ORIGIN']), the 'DEST' and 'ORIGIN' columns are removed from the DataFrame.
* When we use flight\_df.dtypes, it reveals the data types of each column.
* The output indicates that the 'CARRIER' and 'FL\_STATUS' columns have the 'object' data type, while all other columns are of the 'int64' data type.

c. You leave the outcome variable ‘FL\_STATUS’ unchanged in flight\_df. However, for the ‘CARRIER’ predictor variable, you need to convert it into the binary variables and avoid using the Boolean (‘bool’) values. Display in Python the modified column data types, provide and briefly explain them in your report.

A close-up of a list of text

AI-generated content may be incorrect.

**Explanation:**

* In the flight dataset, the 'CARRIER' predictor variable was converted into binary dummy variables to avoid using Boolean (bool) values.
* The 'CARRIER' column was transformed into 4 new columns representing each carrier.
* These new columns were explicitly converted to int64 data type using the astype() method, which ensures the binary values (0 or 1) are stored as integers instead of Booleans (True or False).
* After preprocessing, all numerical columns are of int64 type, except for the FL\_STATUS outcome variable, which remains an object to represent its categorical nature.

d. Display in Python and provide in your report the first 10 records of the modified flight\_df data frame. Briefly explain the outcome and predictors in this case.

A table with numbers and letters

AI-generated content may be incorrect.A white table with numbers and black text

AI-generated content may be incorrect.

**Explanation:**

**Predictor Variables (Features):**

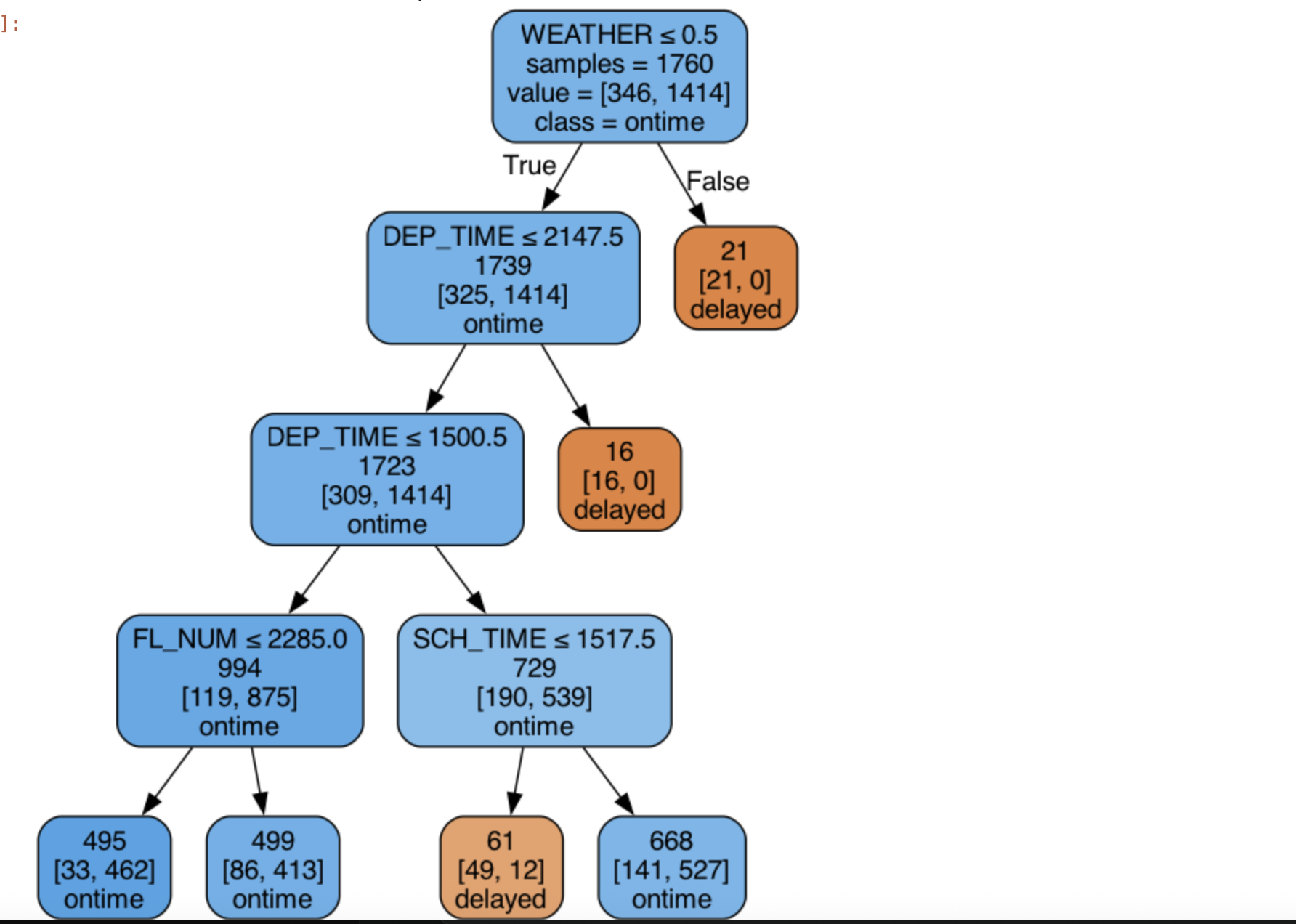
* SCH\_TIME: Scheduled departure time (e.g., 1455, 1640).
* DEP\_TIME: Actual departure time (e.g., 1455, 1640).
* DISTANCE: Distance of the flight in miles (e.g., 184 miles, 213 miles).
* FL\_NUM: Flight number (e.g., 5935, 6155).
* WEATHER: Weather condition, where 0 likely indicates good flying conditions.
* WK\_DAY: Day of the week, encoded numerically (e.g., 4 could represent a specific day).
* MTH\_DAY: Day of the month (e.g., 1, 15).
* CARRIER\_ (CARRIER\_DH, CARRIER\_DL, CARRIER\_MQ, etc.): These are binary columns indicating which carrier operated the flight. For example, CARRIER\_DH indicates whether the flight was operated by carrier 'DH' (1 for yes, 0 for no).

**Outcome Variable:**

FL\_STATUS: This is the outcome variable, representing the flight status. It can have two values:

* Ontime : The flight was on time.
* Delayed: The flight was delayed.

2. Develop a classification tree for the Flight Delays case.

a. Develop in Python the predictor variables (14 variables) and outcome variable (‘FL\_STATUS’), partition the data set (80% for training and 20% for validation partitions). Train a classification tree model using DecisionTreeClassifier() with the training data set and the following tree control parameters: (a) maximum depth (number of splits) equals 4; (b) minimum impurity decrease per split of 0.001; and (c) minimum number of node records (samples) to split equals to 30. Use plotDecisionTree() with the feature\_names and class\_names parameters to display the classification tree in Python and present it in your report.

**Explanation:**

The decision tree predicts flight status ("ontime" or "delayed") based on weather, departure time, scheduled time, and flight number. The first split checks the weather condition, where good weather predicts the flight is ontime. If the weather is not favorable, the flight is predicted as delayed. Further splits refine the prediction by departure time, where earlier departure times are more likely ontime, while later departure times are predicted as delayed. Additionally, flight number and scheduled time further refine these predictions.

b. Using the classification tree, explain the outcome (‘FL\_STATUS’) of a flight if the weather (‘WEATHER’) is in good flying condition, departure time (‘DEP\_TIME’) is 1450 (2:50 pm), and scheduled time (‘SCH\_TIME’) is 1435 (2:35 pm).

**Explanation:**

Given the conditions: Weather in good flying condition (WEATHER = 0), departure time (DEP\_TIME = 1450), and scheduled time (SCH\_TIME = 1435), the classification tree predicts the flight status as follows:

* Weather: Since WEATHER = 0, it falls into the "ontime" branch.
* Departure Time: The DEP\_TIME = 1450 is less than the threshold (2147.5), which further confirms the "ontime" prediction.
* Scheduled Time: The SCH\_TIME = 1435 is less than **1517.5**, this condition leads to the **"**delayed**"** prediction in the decision tree.

Despite initial predictions of **"**ontime**"** based on weather and departure time, the tree ultimately classifies the flight as **"**delayed**"** due to the scheduledtime condition.

c. Identify and display in Python confusion matrices for training and validation partitions. Present them in your report and comment on accuracy (misclassification) rate for both partitions and explain if there is a possibility of overfitting.

A white paper with black text

AI-generated content may be incorrect.

**Explanation:-**

**Training partition:**

* Accuracy is 0.8455
* True Negatives (TN): 86 (ontime, predicted ontime)
* False Positives (FP): 260 (ontime, predicted delayed)
* False Negatives (FN): 12 (delayed, predicted ontime)
* True Positives (TP): 1402 (delayed, predicted delayed)
* **Misclassification Rate** (Training): (1- 0.8455 = 0.1545)This means that around 15.45% of the predictions on the training set were incorrect.

**Validation Partition:**

* Accuracy: 0.8594
* True Negatives (TN): 26 (ontime, predicted ontime)
* False Positives (FP): 56 (ontime, predicted delayed)
* False Negatives (FN): 6 (delayed, predicted ontime)
* True Positives (TP): 353 (delayed, predicted delayed)
* Misclassification Rate (Validation): (1 -0.8594 = 0.1406). This means around 14.06% of the predictions on the validation set were incorrect.

**Overfitting Analysis**:

* The accuracy on the **training partition** and **validation partition** is quite close (both above 84%). This suggests that the model is generalizing well and is not overfitting.
* Overfitting would typically be indicated by a large gap between training and validation accuracy, where training accuracy is much higher than validation accuracy. Since the gap here is small (about 1.4%), the model appears to be well-regularized and is not overfitting.
* The misclassification rates are relatively low, and the slight gap between the training and validation accuracy suggests there is no overfitting.

d. Using the trained classification tree, make classification of flight status (‘delayed’ or ‘ontime’) for the following two new flight records:

A close-up of a number

AI-generated content may be incorrect.

Present and briefly explain the classification results in your report.

**Output:-**

A screen shot of a data

AI-generated content may be incorrect.

**Explanation**:

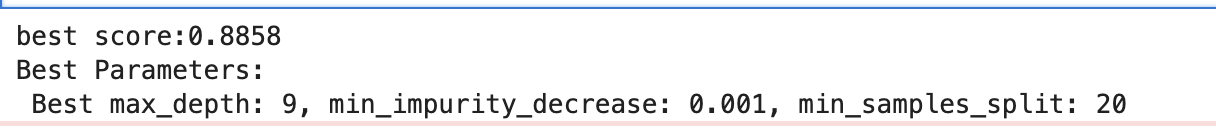
* For both flight records, the model predicts "delayed" as the flight status. This prediction is based on the features like weather, departure time, and carrier.
* Record 1 has a later scheduled and departure time with a "CARRIER\_MQ" value of 1, which suggests a delayed flight.
* Record 2 has schedule time <=1517.5 which falls to delayed with "CARRIER\_UA" and "CARRIER\_US" marked as 1, leading to a prediction of delay.

The model uses the values from the trained decision tree to classify both records as delayed.

3. Apply grid search to improve classification results.

a. Use the GridSearchCV() algorithm in Python to improve (optimize) the classification tree control parameters. Consider the following control parameters: (a) maximum depth (number of splits) in the range from 2 to 30; (b) minimum impurity decrease per split of 0, 0.0005, and 0.001; and (c) minimum number of node records (samples) to split in the range from 5 to 30. Do not use the initial guess grid search, and directly apply the improved grid search. Provide in your report the improved parameters and display inPython the associated classification tree. Display the confusion matrices for training and validation partitions for the improved classification tree.

**Improved Parameters:**



* The best score (0.8858) represents the highest accuracy achieved using the optimized parameters during cross-validation. The best parameters are max\_depth = 9, min\_impurity\_decrease = 0.001, and min\_samples\_split = 20, which were found to provide the best performance for the classification tree model.

A diagram of a flowchart

AI-generated content may be incorrect.

A screenshot of a computer screen

AI-generated content may be incorrect.

**Explanation:**

* The model achieves high accuracy, with the **training accuracy** of 89.49% and **validation accuracy** of 86.85%.
* The **misclassification rates** are reasonably low for both training and validation datasets.(Training : 1 – 0.8949 =0.1051 )(validation : 1-0.8685 = 0.1315)
* **There is a slight possibility of overfitting due to the small gap in performance between the training and validation sets, but the model is still generalizing well.**

b. Present and compare in your report the validation confusion matrices for the classification results in questions 2c and 3a. Using the accuracy value (misclassification rate), which classification tree model would you recommend using for making predictions in this case of flight status (‘delayed’ or ‘ontime’)? Briefly explain your answer.

**Confusion matrix for 2c:** **Confusion matric by gridsearchCV():**

A white paper with black text

AI-generated content may be incorrect.A screenshot of a computer

AI-generated content may be incorrect.

**Analysis of Accuracy and Misclassification Rate:**

**First Model (from Question 2c):** Validation Accuracy: 85.94%

Misclassification Rate: 1 - 0.8594 = 0.1406

**Second Model (from Question 3a):**

Validation Accuracy: 86.85%

Misclassification Rate: 1 - 0.8685 = 0.1315

**Recommendation:**

* Second Model (from Question 3a) has a slightly higher validation accuracy (86.85%) compared to the first model (85.94%).
* The misclassification rate for the second model is also lower (13.15%) compared to the first model (14.06%).

Thus, I would recommend using the second model (from Question 3a)(best parameters)for making predictions on flight status (whether delayed or ontime), as it achieves better generalization performance (higher accuracy and lower misclassification rate).