Case Study 2 - Flight Delays

Predict Delayed or On time flights
with
Classification Trees

Adriel Naranjo
ed2149

Data Mining - BAN 620
Instructor: Dr. Zinovy Radovilsky

1. Upload, explore, clean, and preprocess data for classification trees.
   a. Below we can see the dimensions (rows, cols) of the Flight Delays dataframe.

```
# Determine and present in this report the data frame dimensions, i.e., number of rows and columns.
print(f'The dimensions of the Flight Delays dataset is {flight_df.shape}')
print(f'{flight_df.shape[0]} rows and {flight_df.shape[1]} columns.')
```

```
The dimensions of the Flight Delays dataset is (2201, 11)
2201 rows and 11 columns.
```

   b. After removing the DEST and ORIGIN fields from the Flight Delays dataset there are 9 variables left that describe flights. CARRIER and FL_STATUS are of type 'object' and exhibit categories and the rest of the variables are of type 'int64' and are listed as follows: 'SCH_TIME', 'DEP_TIME', 'DISTANCE', 'FL_NUM', 'WEATHER', 'WK_DAY', 'MTH_DAY'.

```
flight_df = flight_df.drop(['DEST','ORIGIN'], axis=1)
```

```
flight_df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 2201 entries, 0 to 2200
Data columns (total 9 columns):
 #   Column     Non-Null Count  Dtype 
---  ------     --------------  ----- 
 0   SCH_TIME   2201 non-null   int64 
 1   CARRIER    2201 non-null   object
 2   DEP_TIME   2201 non-null   int64 
 3   DISTANCE   2201 non-null   int64 
 4   FL_NUM     2201 non-null   int64 
 5   WEATHER    2201 non-null   int64 
 6   WK_DAY     2201 non-null   int64 
 7   MTH_DAY    2201 non-null   int64 
 8   FL_STATUS  2201 non-null   object
dtypes: int64(7), object(2)
```

   c. Change the 'CARRIER' data type from 'object' to 'category', and then convert this categorical variable into dummy variables. New list of variables and their data types.

```
flight_df.CARRIER = flight_df.CARRIER.astype('category')
```

```
flight_df = pd.get_dummies(flight_df, columns=['CARRIER'], prefix_sep='_',
                           drop_first=True)
```

```
Index(['SCH_TIME', 'DEP_TIME', 'DISTANCE', 'FL_NUM', 'WEATHER', 'WK_DAY',
       'MTH_DAY', 'FL_STATUS', 'CARRIER_DH', 'CARRIER_DL', 'CARRIER_MQ',
       'CARRIER_OH', 'CARRIER_RU', 'CARRIER_UA', 'CARRIER_US'],
      dtype='object')
```

```
flight_df.info()
```

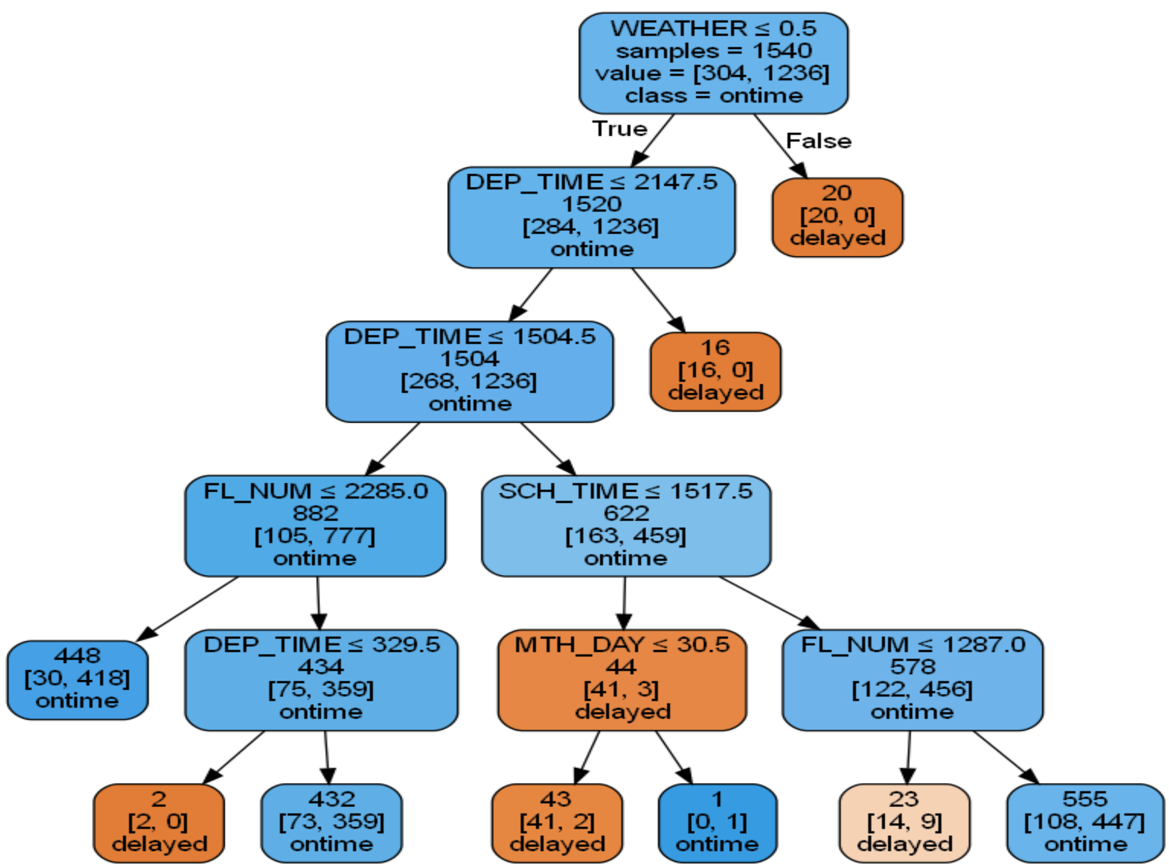
```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 2201 entries, 0 to 2200
Data columns (total 15 columns):
 #   Column      Non-Null Count  Dtype 
---  ------      --------------  ----- 
 0   SCH_TIME    2201 non-null   int64 
 1   DEP_TIME    2201 non-null   int64 
 2   DISTANCE    2201 non-null   int64 
 3   FL_NUM      2201 non-null   int64 
 4   WEATHER     2201 non-null   int64 
 5   WK_DAY      2201 non-null   int64 
 6   MTH_DAY     2201 non-null   int64 
 7   FL_STATUS   2201 non-null   object
 8   CARRIER_DH  2201 non-null   uint8 
 9   CARRIER_DL  2201 non-null   uint8 
 10  CARRIER_MQ  2201 non-null   uint8 
 11  CARRIER_OH  2201 non-null   uint8 
 12  CARRIER_RU  2201 non-null   uint8 
 13  CARRIER_UA  2201 non-null   uint8 
 14  CARRIER_US  2201 non-null   uint8 
dtypes: int64(7), object(1), uint8(7)
```

d. First 10 records of the modified flight_df dataframe

	SCH_TIME	DEP_TIME	DISTANCE	FL_NUM	WEATHER	WK_DAY	MTH_DAY	FL_STATUS	CARRIER_DH	CARRIER_DL	CARRIER_MQ	CARRIER_OH	CARRIER_RU	CARRIER_UA	CARRIER_US
0	1455	1455	184	5935	0	4	1	ontime	0	0	0	1	0	0	0
1	1640	1640	213	6155	0	4	1	ontime	1	0	0	0	0	0	0
2	1245	1245	229	7208	0	4	1	ontime	1	0	0	0	0	0	0
3	1715	1709	229	7215	0	4	1	ontime	1	0	0	0	0	0	0
4	1039	1035	229	7792	0	4	1	ontime	1	0	0	0	0	0	0
5	840	839	228	7800	0	4	1	ontime	1	0	0	0	0	0	0
6	1240	1243	228	7806	0	4	1	ontime	1	0	0	0	0	0	0
7	1645	1644	228	7810	0	4	1	ontime	1	0	0	0	0	0	0
8	1715	1710	228	7812	0	4	1	ontime	1	0	0	0	0	0	0
9	2120	2129	228	7814	0	4	1	ontime	1	0	0	0	0	0	0

The outcome that we will predicting in this work is whether or not a flight status is delayed or on time (FL_STATUS) based on the following predictors: scheduled time (SCH_TIME), departure time (DEP_TIME), distance of the flight (DISTANCE), weather presence (WEATHER), which day of the week it is (WK_DAY), which month it is (MTH_DAY), which carrier is performing the flight, and the flight number (FL_NUM).

2. Develop a classification tree for the Flight Delays case.
   a. A trained classification tree with max_depth=5, min_impurity_decrease=0.001, min_samples_split=10



b. For a flight with good flight conditions (WEATHER = 0) we would first traverse the left side of the decision tree since the first node of the tree splits left if weather condition is <= 0.5. If departure time is at 1605 (4:05 PM) we would continue down the tree to the right node that is at depth 3, which splits on schedule time <= 1517.5. Since the schedule time is 1510 (3:10 pm) which is less, we go left to the node that splits on month day <= 30.5. Since day of the month is 28 and is less, we end at the leaf node that states that this particular flight will be delayed.

c.

Training Partition	Validation Partition
Confusion Matrix (Accuracy 0.8558) Prediction Actual 0 1 0 93 211 1 11 1225	Confusion Matrix (Accuracy 0.8427) Prediction Actual 0 1 0 39 85 1 19 518

The classification accuracy for the training partition is .8558 and the classification accuracy for the test partition is .8427. Although the predictions on the training partition are higher, the difference is not significant enough to conclude that there is any overfitting taking place. I also calculated the precision and recall for each of these confusion matrices and there were also no significant differences between these measures convincing that overfitting is not noticeably present.

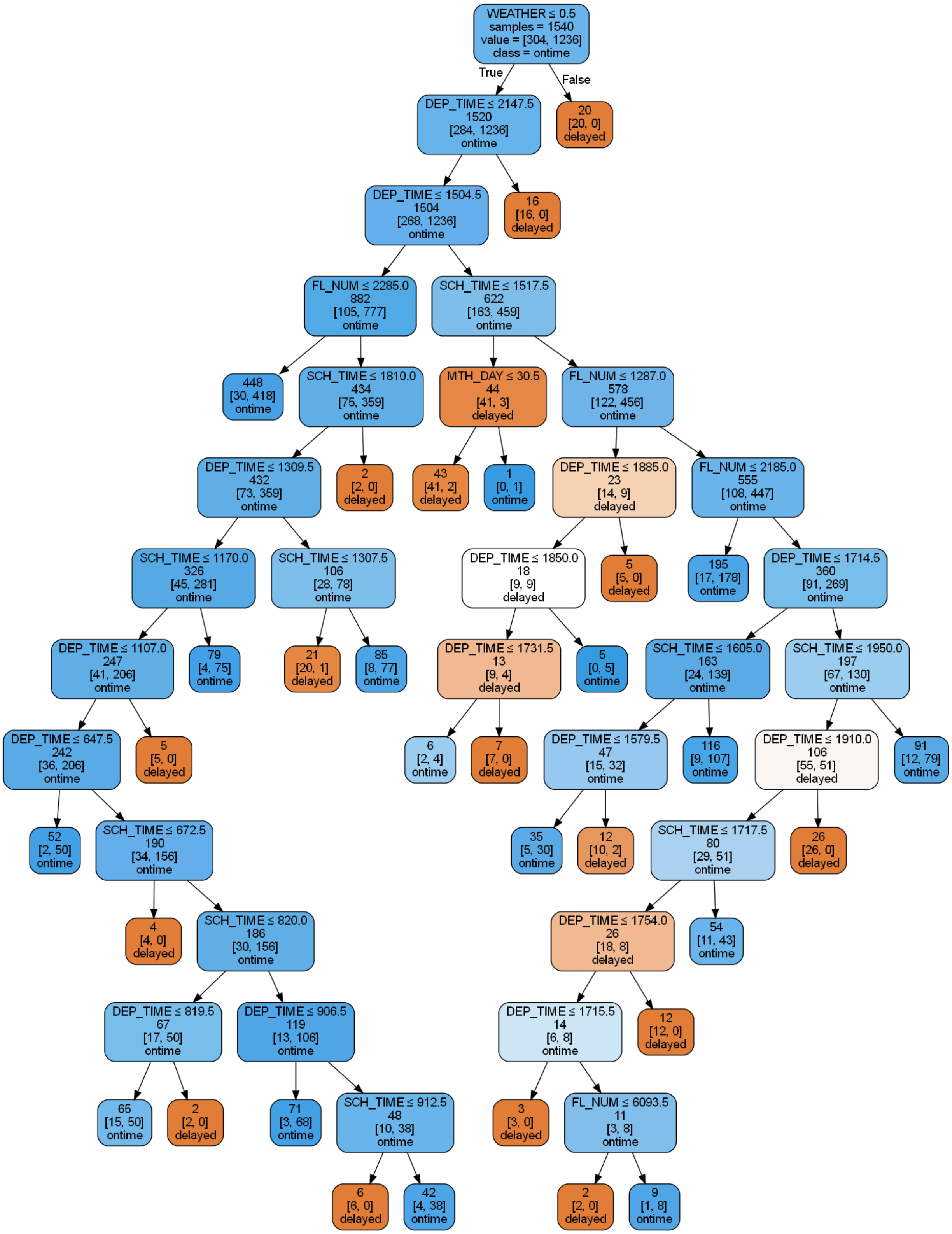
d. When presented with two new data samples, we can see the Classification Tree classified the first sample as “delayed” and the second as on time. The first sample has poor weather conditions which may have ended its journey down the classification tree a bit too early since the first split is concerning this variable. The second sample is classified as on time not only because weather conditions are good but also because its schedule time, department time, and flight number were the right values that lead to an “on time” leaf.

```
New Flight Data and Classifications for New Data
   SCH_TIME  DEP_TIME  DISTANCE  FL_NUM  WEATHER  WK_DAY  MTH_DAY  CARRIER_DH  \
0      1230      1240       214     808        1       4       20           0
1      2050      2105       199    4976        0       5       30           0

   CARRIER_DL  CARRIER_MQ  CARRIER_OH  CARRIER_RU  CARRIER_UA  CARRIER_US  \
0           0           0           0           0           1           0
1           1           0           0           0           0           0

  Classification
0        delayed
1         ontime
```

3. Apply grid search and ensemble trees to improve classification results.
    a.



The Classification tree above was optimized and trained with the following parameters:

max_depth=18, min_impurity_decrease=0.001, min_samples_split=7

Training Partition for Optimized Tree

```
Confusion Matrix (Accuracy 0.9169)

       Prediction
Actual    0    1
     0  181  123
     1    5 1231
```

Validation Partition for Optimized Tree

```
Confusion Matrix (Accuracy 0.8941)

       Prediction
Actual   0   1
     0  69  55
     1  15 522
```

b. When comparing the accuracy of the unoptimized classification tree (0.8427) and the optimized classification tree, we can see that the optimized classification tree (0.8941) out-performs the unoptimized tree by over five percent. Since the unoptimized classification tree used hyperparameters that did not capture the structure of the data, this led to overfitting to the training data and capturing noise rather than the underlying patterns. As a result, the validation accuracy for the unoptimized tree is sub-optimal. When applying GridSearchCV for optimization, the goal is to fine-tune the tree's parameters to reduce overfitting and improve its generalization ability. Using GridSearchCV to produce hyperparameters that captures the structure of the data is why the optimized tree performed better on the validation set and would be the model to use to predict flight status.

Validation Partition for Preliminary Tree

```
Confusion Matrix (Accuracy 0.8427)

       Prediction
Actual   0   1
     0  39  85
     1  19 518
```

Validation Partition for Optimized Tree

```
Confusion Matrix (Accuracy 0.8941)

       Prediction
Actual   0   1
     0  69  55
     1  15 522
```

Model Choice: Optimized Decision Tree