Using Machine Learning to Predict Flight Delays : Decision Trees and Random Forests

Which Factors Relate to the Timeliness of Flights?

John Karuitha

1/16/23

Table of Contents

1 Background **2**

2 Significance of the Analysis **3**

3 Summary of Results **4**

4 Data **5**

5 Exploring the Data **6**

6 Machine Learning Models **8**
- 6.1 **Creating Training Set and Testing Sets** . . . 8
- 6.2 **Handling Class Imbalance** . . . 8
- 6.3 **Creating the First Decision Tree** . . . 9
  - 6.3.1 **Prunning the (Decision) Tree** . . . 10
- 6.4 **Extended Decision Tree** . . . 11
- 6.5 **Prunning the Extended Decision Tree** . . . 12
- 6.6 **Random Forest Model** . . . 13
- 6.7 **Comparing Model Performance** . . . 13

7 Variable Importance **17**

Conclusion **19**

Chapter 1

Background

Flight delays are a significant concern in the airline industry. Apart from the inconvenience caused to travelers, delays also affect the reputation of airlines, negatively impacting market share. In this analysis, I utilize data for flights between New York and Washington DC. The central questions in the analysis are;

- Which factors have a significant relationship to flight delays?
- Can machine learning be useful in predicting flight delays?

Read More of my Work

Please visit my rpubs site to see more data projects. Alternatively, copy and paste the link https://www.rpubs.com/Karuitha into your browser.
My data visualizations projects are available in my Tableau Public profile page or copy and paste the link https://public.tableau.com/app/profile/john.karuitha.
My Shiny web apps are available on this site. You can copy-paste this web address instead https://karuitha.shinyapps.io/

Tools Utilized & Skills Applied

R (R Core Team 2022), Decision Tree Model, Random Forest Model, Quarto, Data Science, Machine Learning.

Chapter 2

Significance of the Analysis

If airlines could accurately forecast delays, then they could mitigate the effects of the delays on the consumers. This intervention may save airlines substantial costs, and especially the cost related to consumer churn.

Chapter 3

Summary of Results

Chapter 4

Data

The file `FlightDelays.csv` contains information on all commercial flights departing the Washington, DC area and arriving at New York during January 2004.

```r
## read the data and clean names
flights <- read_csv("FlightDelays.csv") %>%
    clean_names() %>%
    mutate(
        carrier = factor(carrier),
        day_week = factor(day_week),
           weather = factor(weather))
```

The data consists of 2201 rows and and 9 variables. For each flight (row of data), there is information on the distance of the route, the scheduled time and date of the flight, and so on.Table 2 describes variables in this file.

Table 2. Description of variables for Flight Delays example

Variable	Definition
CRS_DEP_TIME	Scheduled departure time
CARRIER	The airline
DEP_TIME	Actual departure time
DISTANCE	Flight distance in miles
FL_DATE	Flight date
Weather	Whether the weather is inclement (1) or not (0)
DAY_WEEK	Day of week (1= Mon, 2=Tus, 3=Wed….)
DAY_OF_MONTH	Day of month (1= the first day of month; 2= the second day of month….)
Flight_Status	Whether the flight was delayed or on time (defined as arriving within 15 min of scheduled time)

Figure 4.1: Flights data

The variable that we are trying to predict is whether or not a flight is delayed (Fight_Status).

Chapter 5

Exploring the Data

```
flights %>%
    sapply(is.na) %>%
    colSums()
```

```
crs_dep_time       carrier      dep_time      distance       fl_date
           0             0             0             0             0
     weather      day_week  day_of_month flight_status
           0             0             0             0
```

```
flights %>%
    select(-fl_date) %>%
    GGally::ggpairs(mapping = aes(color = flight_status, fill = flight_status)) +
    scale_fill_manual(values = c("skyblue", "gray80")) +
    scale_color_manual(values = c("skyblue", "gray80"))
```

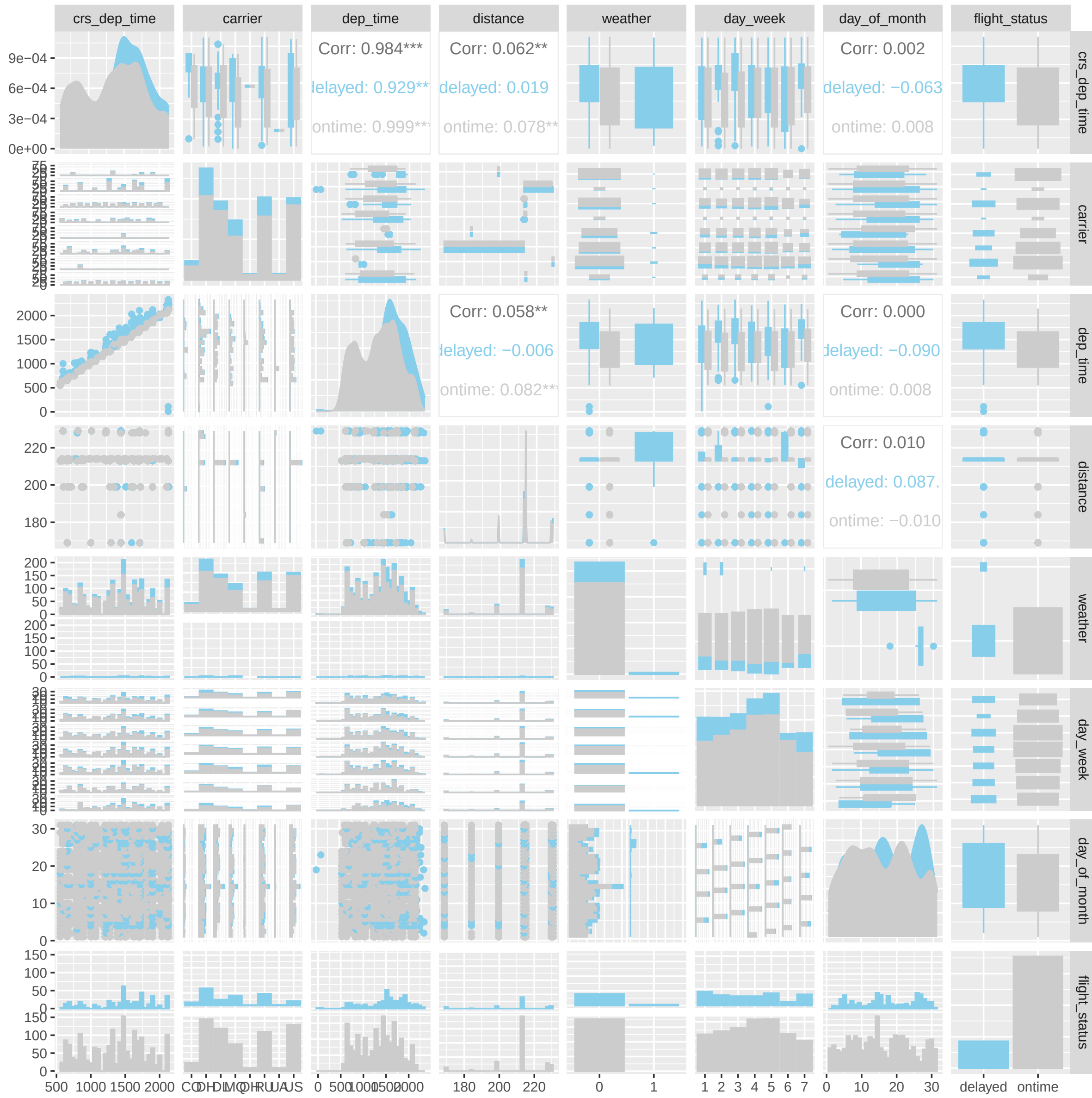


Figure 5.1: Exploring the Data

Chapter 6

Machine Learning Models

In this section, I train a pair of models.

- The Decision Tree Model.
- The Random Forest Model.

But first, I split the data into training set and testing set.

6.1 Creating Training Set and Testing Sets

In this section, I partition the data into 60% for training and 40% for validation.

```
## Split the data into training and testing set
set.seed(300, sample.kind = "Rounding")
flights_split <- initial_split(flights, prop = 0.6, strata = flight_status)


flights_training <- flights_split %>% training()
flights_testing <- flights_split %>% testing()
```

6.2 Handling Class Imbalance

It is notable that there are 428 delays against 1773 ontime departures. This level of class imbalance has an adverse effect on machine learning models. To correct this anomaly, I upsample the training set such that it has a degree of class balance.

```
my_recipe <- recipes::recipe(flight_status ~ carrier + distance + weather + day_week + day_of_month
    themis::step_upsample(over_ratio = 1) %>%
    step_dummy(weather, day_week) %>%
```

```
    prep(training = flights_training)


## Apply to training data
flights_training <- my_recipe %>%
    bake(new_data = NULL)

## Apply to testing data
flights_testing <- my_recipe %>%
    bake(new_data = flights_testing)
```

The training set is now balanced. The models can pick the signal for the previously under-represented class from the training set.

Next, we train the model using the training set and test the models using the validation or testing set.

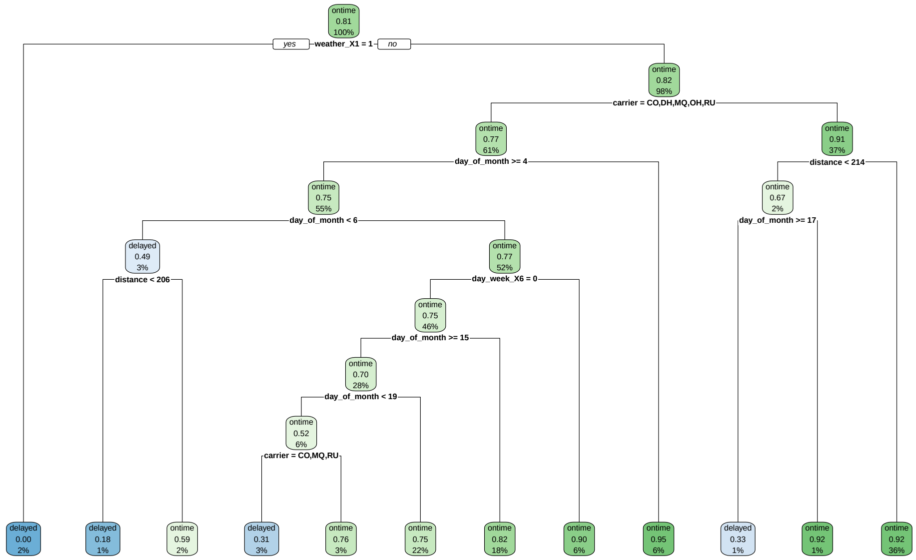
6.3 Creating the First Decision Tree

I fit a classification tree to the flight delay variable using all the relevant predictors in FlightDelays.csv on training sets with maximum of 8 levels and set up cp =0.001 and then plot the tree.

Note: `cp` refers to complexity parameter.

I then fit the classification tree.

```
## Fitting the regression tree on training data
flights_tree <- rpart(flight_status ~ .,
                      data = flights_training,
                      method = 'class',
                      control = rpart.control(cp = 0.001,
                                              maxdepth = 8))
## Plotting the tree
rpart.plot(flights_tree)
```



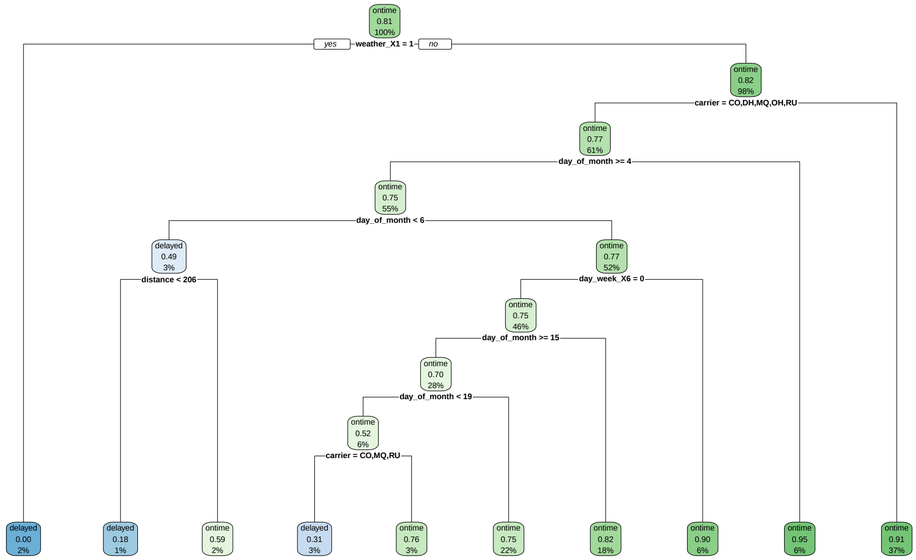
6.3.1 Prunning the (Decision) Tree

In the setting of decision tree, there is a technique called pruning the tree. Pruning is a data compression technique for reducing the size of decision trees by removing non-critical and redundant sections of the tree.

The purpose of pruning is to reduce the complexity of the classifier. Pruning also helps improves predictive accuracy reducing of over-fitting.

In this section, I prune the tree we grew in section 3 above. In pruning this tree, I raise the complexity parameter by a factor of 10 to 0.1.

```
## Prunning the tree
pruned_tree <- prune(flights_tree, cp = 0.01)

## Plot the prunned tree
rpart.plot(pruned_tree)
```



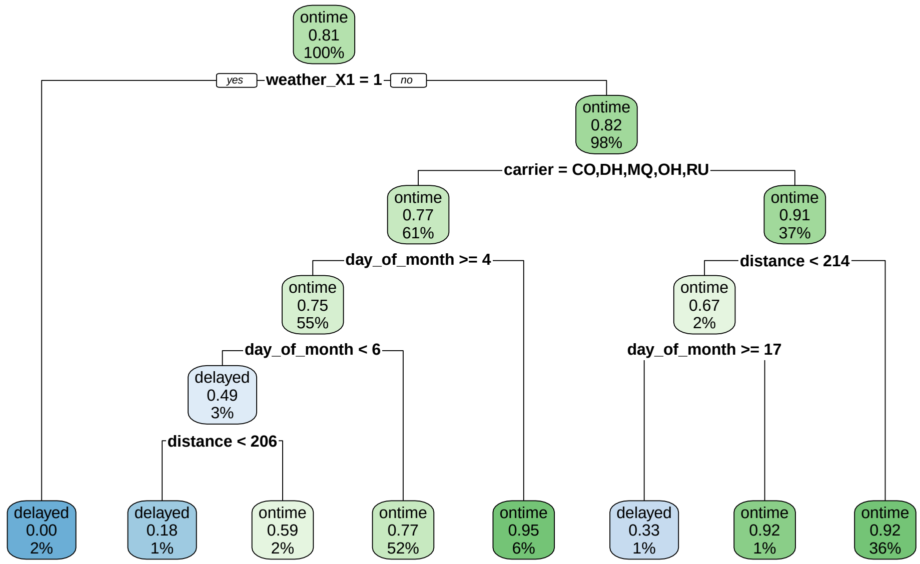
This pruned tree suggests that the sole driver of flight delays is the weather.

6.4 Extended Decision Tree

Fit a new classification tree to the flight delay variable using all the relevant predictors on training sets, excluding the Weather predictor. Set cp=0.001 and maximum =6. Plot this new classification tree. (10 points)

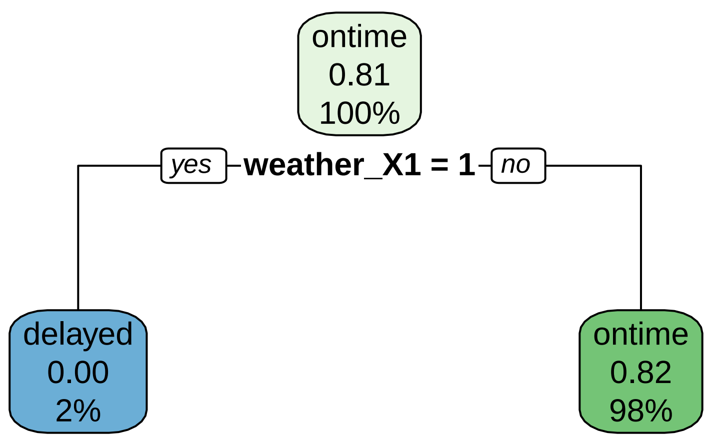
In this section, I create a new classification tree with cp = 0.001 and maximum depth of 6. I then plot the tree.

```
## Create another classification tree, cp = 0.001, depth = 6
another_flights_tree <- rpart(flight_status ~ ., data = flights_training, method = 'class',
                              control = rpart.control(cp = 0.001,
                                                      maxdepth = 6))

## Plot the tree
rpart.plot(another_flights_tree)
```



6.5 Prunning the Extended Decision Tree

```
## Prunning the tree
another_prunned_tree <- prune(another_flights_tree, cp = 0.01)

## Plot the prunned tree
rpart.plot(another_prunned_tree)
```



6.6 Random Forest Model

```
rand_model <- randomForest::randomForest(factor(flight_status) ~ .,
                                         data = flights_training, importance = TRUE, proximity = TI

summary(rand_model)
```

```
                Length  Class  Mode
call                  5 -none- call
type                  1 -none- character
predicted          1319 factor numeric
err.rate           1500 -none- numeric
confusion             6 -none- numeric
votes              2638 matrix numeric
oob.times          1319 -none- numeric
classes               2 -none- character
importance           40 -none- numeric
importanceSD         30 -none- numeric
localImportance       0 -none- NULL
proximity       1739761 -none- numeric
ntree                 1 -none- numeric
mtry                  1 -none- numeric
forest               14 -none- list
y                  1319 factor numeric
test                  0 -none- NULL
inbag                 0 -none- NULL
terms                 3 terms  call
```

6.7 Comparing Model Performance

Based on the extended decision tree model, I do predictions for both training and validations sets and report their confusion matrix respectively and other model performance metrics.

I now do the predictions on the test set and likewise, report the confusion matrix.

```
## Prediction on the testing set
train_prediction_test <- predict(another_flights_tree, newdata = flights_testing, type = "class")

## Predictions for the random forest model
rand_predictions <- predict(rand_model, newdata = flights_testing)

## Confusion matrix on the testing set
predictions <- flights_testing %>%
    select(flight_status) %>%
```

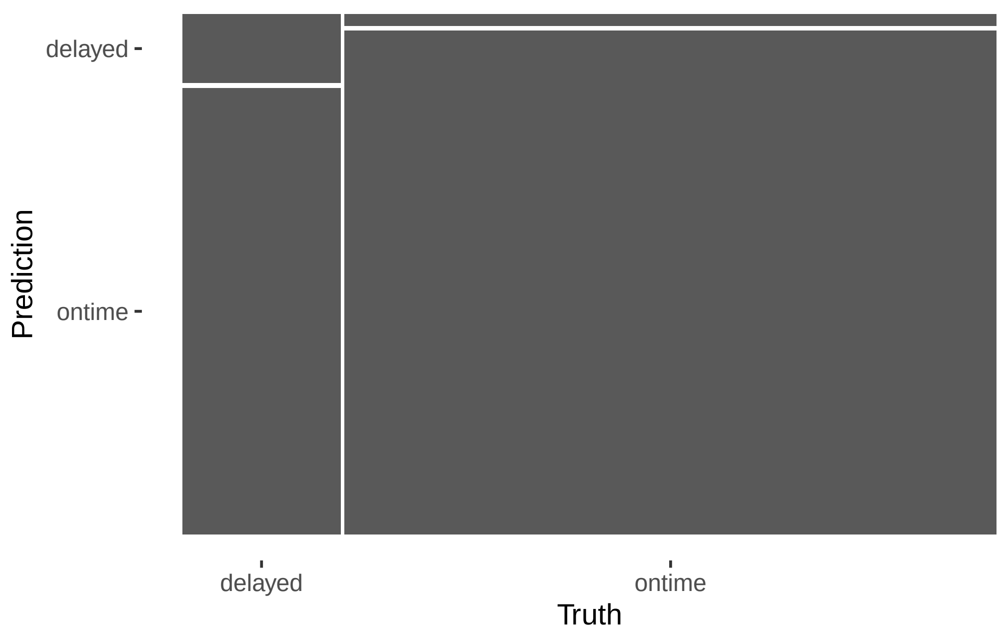
```r
    mutate(flight_status = factor(flight_status, labels = c("delayed", "ontime"))) %>%
    bind_cols(train_prediction_test) %>%
    bind_cols(rand_predictions) %>%
    set_names(c("flight_status", "decision_tree", "rand_forest"))
```

For the decision tree model, the metrics are as follows:

```r
predictions %>%
    conf_mat(truth = flight_status, estimate = rand_forest)
```

```
          Truth
Prediction delayed ontime
   delayed      23     16
   ontime      149    694
```

```r
predictions %>%
    conf_mat(truth = flight_status, estimate = rand_forest) %>%
    autoplot()
```




```r
predictions %>%
    conf_mat(truth = flight_status, estimate = rand_forest) %>%
    summary()
```

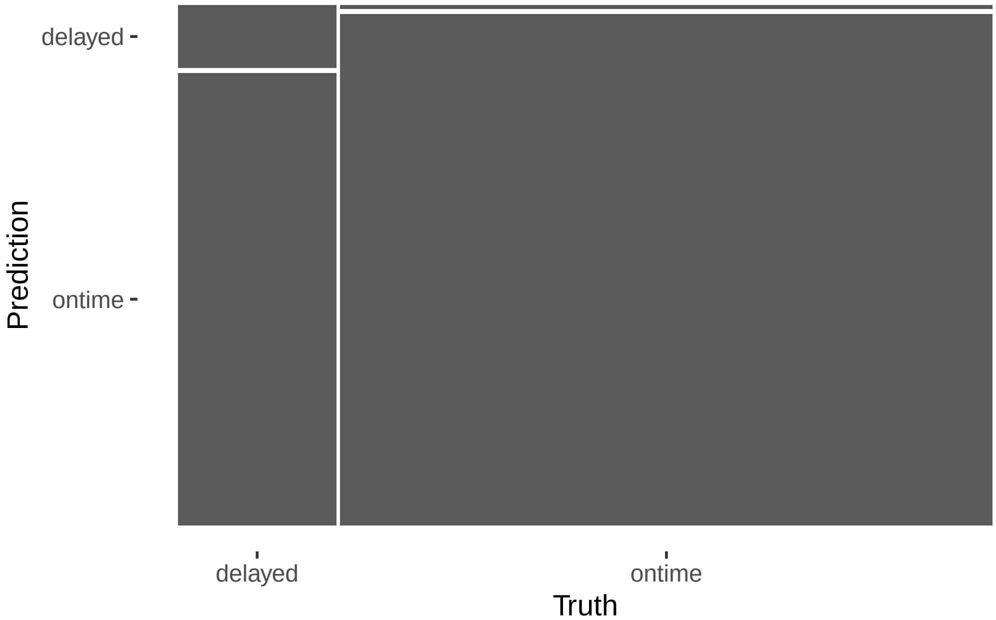
```
# A tibble: 13 x 3
   .metric              .estimator .estimate
   <chr>                <chr>          <dbl>
 1 accuracy             binary        0.813
 2 kap                  binary        0.157
 3 sens                 binary        0.134
 4 spec                 binary        0.977
 5 ppv                  binary        0.590
 6 npv                  binary        0.823
 7 mcc                  binary        0.214
 8 j_index              binary        0.111
 9 bal_accuracy         binary        0.556
10 detection_prevalence binary        0.0442
11 precision            binary        0.590
12 recall               binary        0.134
13 f_meas               binary        0.218
```

The metrics for the random forest model are as follows:

```r
predictions %>%
    conf_mat(truth = flight_status, estimate = decision_tree)
```

```
          Truth
Prediction delayed ontime
   delayed      21      5
   ontime      151    705
```

```r
predictions %>%
    conf_mat(truth = flight_status, estimate = decision_tree) %>%
    autoplot()
```



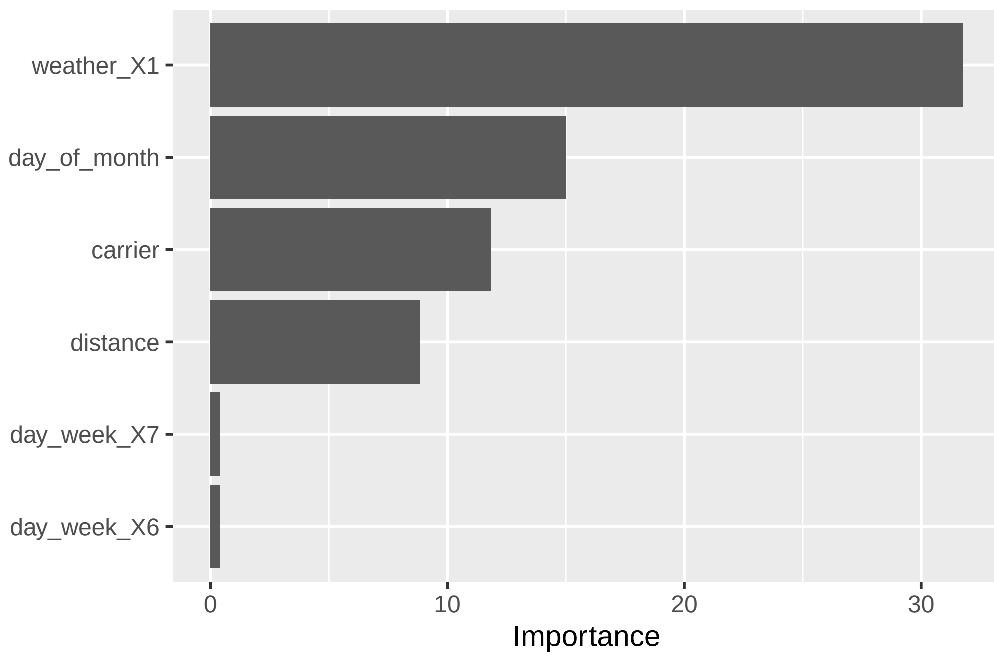

```
predictions %>%
    conf_mat(truth = flight_status, estimate = decision_tree) %>%
    summary()
```

```
# A tibble: 13 x 3
   .metric              .estimator .estimate
   <chr>                <chr>          <dbl>
 1 accuracy             binary        0.823
 2 kap                  binary        0.170
 3 sens                 binary        0.122
 4 spec                 binary        0.993
 5 ppv                  binary        0.808
 6 npv                  binary        0.824
 7 mcc                  binary        0.270
 8 j_index              binary        0.115
 9 bal_accuracy         binary        0.558
10 detection_prevalence binary        0.0295
11 precision            binary        0.808
12 recall               binary        0.122
13 f_meas               binary        0.212
```

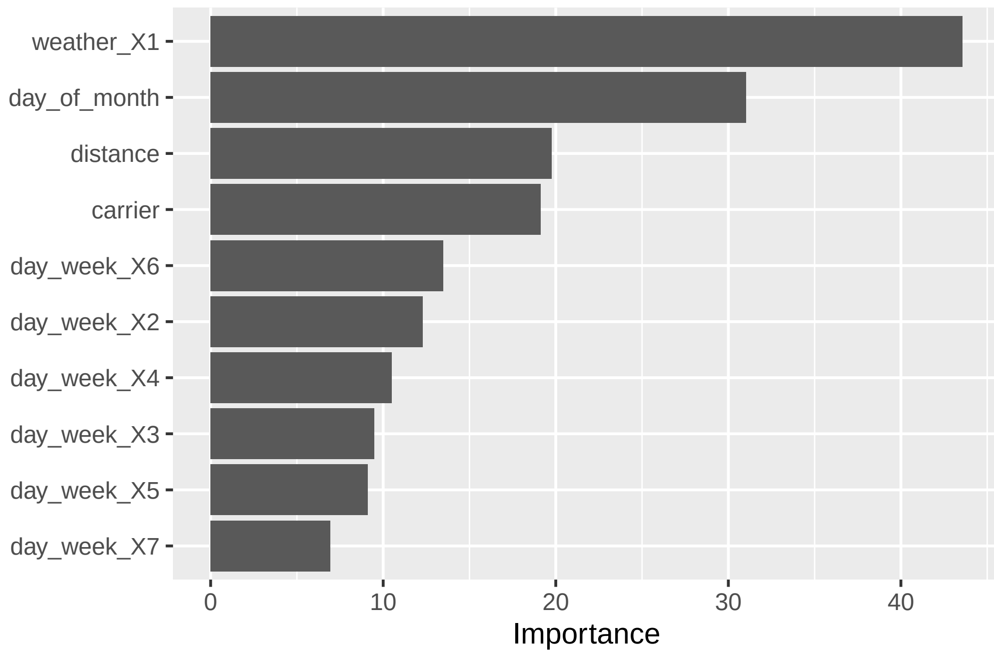
Chapter 7

Variable Importance

```
vip(another_flights_tree)
```




```
vip(rand_model)
```


weather_X1
day_of_month
distance
carrier
day_week_X6
day_week_X2
day_week_X4
day_week_X3
day_week_X5
day_week_X7
0
10
20
30
40
Importance

Conclusion

R Core Team. 2022. *R: A Language and Environment for Statistical Computing*. Vienna, Austria: R Foundation for Statistical Computing. https://www.R-project.org/.