Predictive Modeling for Flight Fare

Tom Takeuchi

2020-12-11

Overview

Real-world question

Many people have used planes as a mean of transportation. Air transportation is a very fast and innovative way of traveling a long distance, but what determines the fare? For example, when I search for the flights going to my home country, Japan, some flights have a very cheap fare while other flights have very expensive fares. I was inspired by this YouTube video. As the video says, day of the week is an important variable to determine the flight fare, but what else? I would like to determine other factors that influence the price other than what the video talks about.

This topic is interesting for me since I am from Japan and whenever I go back there, I use a plane. Predicting the fare would be helpful to plan future trips.

Data Source

The data comes from the Bureau of Transportation Statistics. The datasets I downloaded show the numerous flights that were operated in 2018. They contain data such as flight fare, distance, and origin airport name. According to their website, the datasets I downloaded are a 10% sample of the total airline tickets from reporting carriers collected by the Office of Airline Information of the Bureau of Transportation Statistics.

Approach

Problem Description

As I described in the previous section, I would like to find what variables influence the flight fare. The data I have shows various variables for predicting the flight fare including distance, origin, destination, round trip status, operating carrier, coupon type, and single carrier indicator. A more detailed description for each variable is given in a later section. With the variables in the data I have, I am going to create several predictive models that use different variables to predict the flight fare. Then, I would like to use cross-validation to calculate the mean absolute error for each of the models. Finally, I will compare the mean absolute errors and determine how the performance gets better if some variables are added.

Data

There are two datasets that I downloaded from BTS for this project. There are two because each dataset misses some variables that are indispensable for making predictive modeling. For example, the first dataset does not have the variable that shows the price of the tickets while the other one has it. Each dataset is supposed to combined with the other one after it is wrangled.

Provenance

Bureau of Transportation Statistics (BTS) is a part of the Department of Transportation and is the preeminent source of statistics on various kinds of transportation activities in the United States. Surveys are a major source of the data collection method that BTS uses. In addition to their own data collection activities, they also compile data and statistics from a wide range of sources. The datasets I downloaded are the aggregated reports of numerous flights in 2018 from various airline carriers.

I found the dataset from Kaggle that has variety of useful variables that the data from BTS does not have. However, I decided not to use the dataset because it gives no provenance information, which means it is disconnected from the real world. On the other hand, the data from Bureau of Transportation Statistics has clear a provenance.

Structure

Both of the datasets have 1048575 flights recorded in 2018, and the tables below shows the first six rows. The first dataset has 12 columns and the other one has 5 rows.

First Dataset

itin_id	origin_airport_id	origin	origin_state_fips	origin_state_abr	dest_airport_id	dest	dest_state_fips	dest_state_abr	coupon_type	operating_carrier	distance_group
202013	10135	ABE	42	PA	10397	ATL	13	GA	A	9E	2
202014	10135	ABE	42	PA	10397	ATL	13	GA	A	9E	2
202015	10135	ABE	42	PA	10397	ATL	13	GA	A	9E	2
202016	10135	ABE	42	PA	10397	ATL	13	GA	A	9E	2
202017	10135	ABE	42	PA	10397	ATL	13	GA	A	9E	2
202018	10135	ABE	42	PA	10397	ATL	13	GA	A	9E	2

Second Dataset

itin_id	roundtrip	online	itin_fare	distance
202013	0	1	133	692
202014	0	1	151	692
202015	0	1	163	692
202016	0	1	164	692
202017	0	1	168	692
202018	0	1	180	692

Description for each of the columns

- **itin_id**: ID given for individual flights
- **origin_airport_id**: ID of individual origin airports
- **origin**: name of the abbreviated origin airport name
- **origin_state_fips**: state FIPS of origin airport
- **origin_state_abr**: abbreviated state name of the origin airport
- **dest**: name of the abbreviated destination airport name
- **dest_state_fips**: state FIPS of destination airport
- **dest_state_abr**: abbreviated state name of the destination airport
- **coupon_type**: type of coupon (There are A and D, which I am not sure what they represent)
- **operating_carrier**: operating carrier. distance_group: number of distance group per 500 miles (For example, if the distance is 700, then the distance_group will be 700 / 50 = 12)

- **number_of_flight**: number of flights that each travel has
- **roundtrip**: 0 for non-round trip, and 1 for round trip. online: single carrier indicator (0 for no and 1 for yes)
- **itin_fare**: itinerary fare per person. distance: distance traveled.

Nominal data: itin_id, origin_airport_id, origin, origin_state_fips, origin_state_abr, dest_airport_id, dest, dest_state_fips, dest_state_abr, coupon_type, operating_carrier, roundtrip, online.

Numeric data: distance, distance_group, number_of_flight, itin_fare.

Appropriateness for task

The datasets contain 1048575 rows, and thus, this is large enough for the task. In addition to that, the data is coming from BTS, which is a principal agency of the U.S. Federal Statistical System. Thus, the data is reliable. In fact, there are many researchers and institutions that use data from BTS.

However, there are two big problems to overcome in order to make the predictive model more accurate. The first problem is that the first dataset have some rows with the same itinerary ID while the other dataset does not. Specificity, each row with the same itinerary ID has the same values as each other. Thus, the total number of the rows with the same columns is equivalent to the number of flights within the trip. For example, if there are 5 rows with the same itinerary ID in the first dataset, it means that the trip contains 5 flights. On the other hand, the second dataset does not take into the account of that aspect. Instead, it has the column 'roundtrip' which shows 0 for non-round trip flights, and 1 for round trip flights. This is a big problem to solve in order to join the tables by using the same itinerary ID.

The next problem is the outliers. Taking a close look at the datasets, I realized that there were many outliers. For example, there are some flights whose fares are more than $20,000, which are extremely expensive for a domestic flight.

Data Wrangling

To solve the first problem of the same itin_id, I made a new dataset called data_new. I used group_by function to group the rows with the same itin_id. I also made a new column called 'number_of_flight' in the new dataset. It shows how many trips that each trip includes by adding the number of the rows with the same itin_id. I made this column because I thought it would be useful to know later when making predictive modeling for the flight fare. The process is shown below.

```
data_new <- data1 %>%
  group_by(itin_id) %>%
  mutate(number_of_flight = sum(itin_id) / itin_id)
```

Now, each itin_id in the data_new corresponds to that of data2, and it is ready to be joined. I used left_joint to combine the two tables. The table below shows the first six rows of the combined data.

itin_id	origin_airport_id	origin	origin_state_fips	origin_state_abr	dest_airport_id	dest	dest_state_fips	dest_state_abr	coupon_type	operating_carrier	roundtrip	online	number_of_flight	itin_fare	distance
20201300135	ABE	42	PA	10397	ATL	13	GA	A	9E	2	1	0	1	133	692
20201400135	ABE	42	PA	10397	ATL	13	GA	A	9E	2	1	0	1	151	692
20201500135	ABE	42	PA	10397	ATL	13	GA	A	9E	2	1	0	1	163	692
20201600135	ABE	42	PA	10397	ATL	13	GA	A	9E	2	1	0	1	164	692
20201700135	ABE	42	PA	10397	ATL	13	GA	A	9E	2	1	0	1	168	692
20201800135	ABE	42	PA	10397	ATL	13	GA	A	9E	2	1	0	1	180	692

The combined data has 1048575 rows. Since distance_group and distance are basically the same, I decided to filter out the distance_group. Also, there are some columns that show the same information about the

destination or origin airports, and thus, I decided to filter them out as well. The cleaned dataset looks like below.

itin_id	origin	dest	coupon_type	operating_carrier	distance	distance_group	number_of_flights	roundtrip	online	itin_fare
202013	ABE	ATL	A	9E	692	2	1	0	1	133
202014	ABE	ATL	A	9E	692	2	1	0	1	151
202015	ABE	ATL	A	9E	692	2	1	0	1	163
202016	ABE	ATL	A	9E	692	2	1	0	1	164
202017	ABE	ATL	A	9E	692	2	1	0	1	168
202018	ABE	ATL	A	9E	692	2	1	0	1	180

Modeling Question and Approach

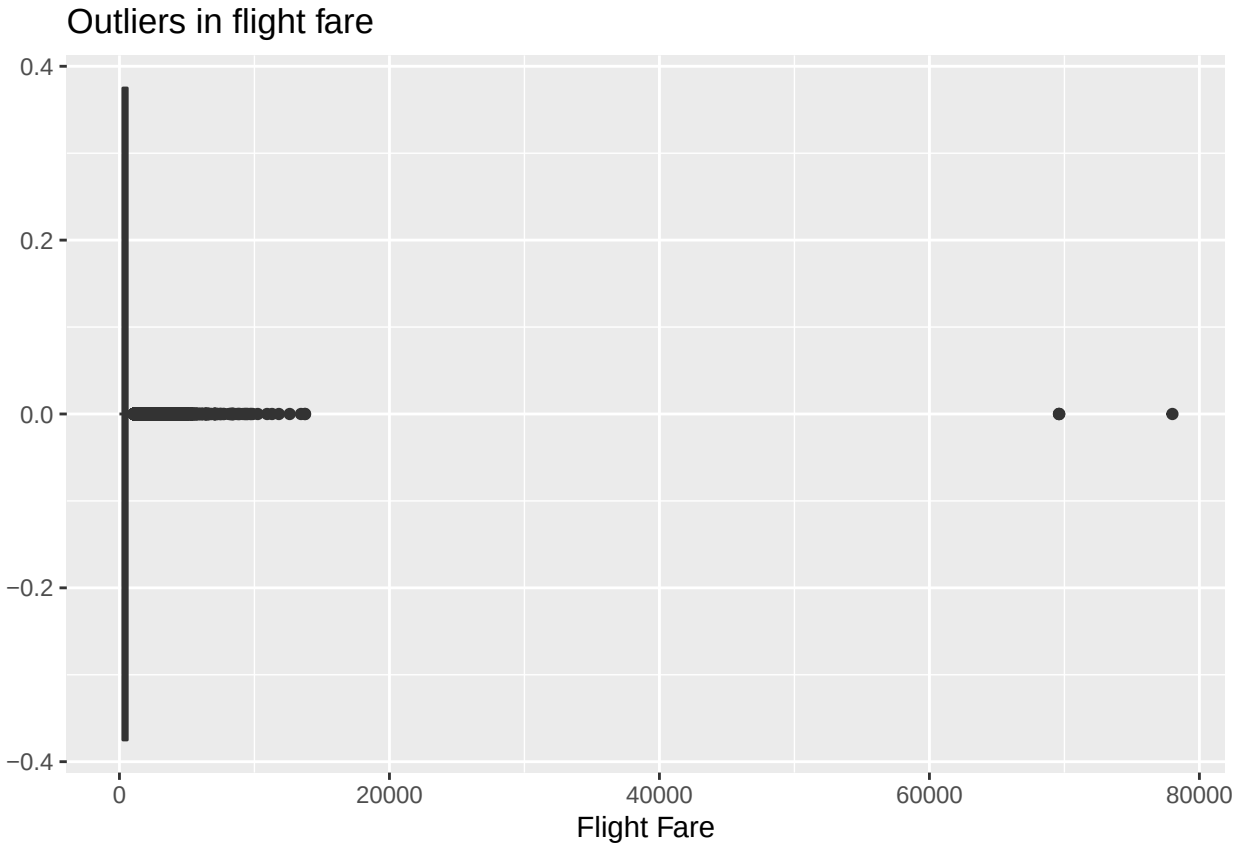
The question that has to be solved before making a prediction is whether there is a correlation between flight fare and other variables such as distance. If there is, linear regression will be appropriate for this model. To examine this, I am going to make some plots that show the relationships. I would also like to solve the problems with the outliers in this section.

Explanatory Data Analysis

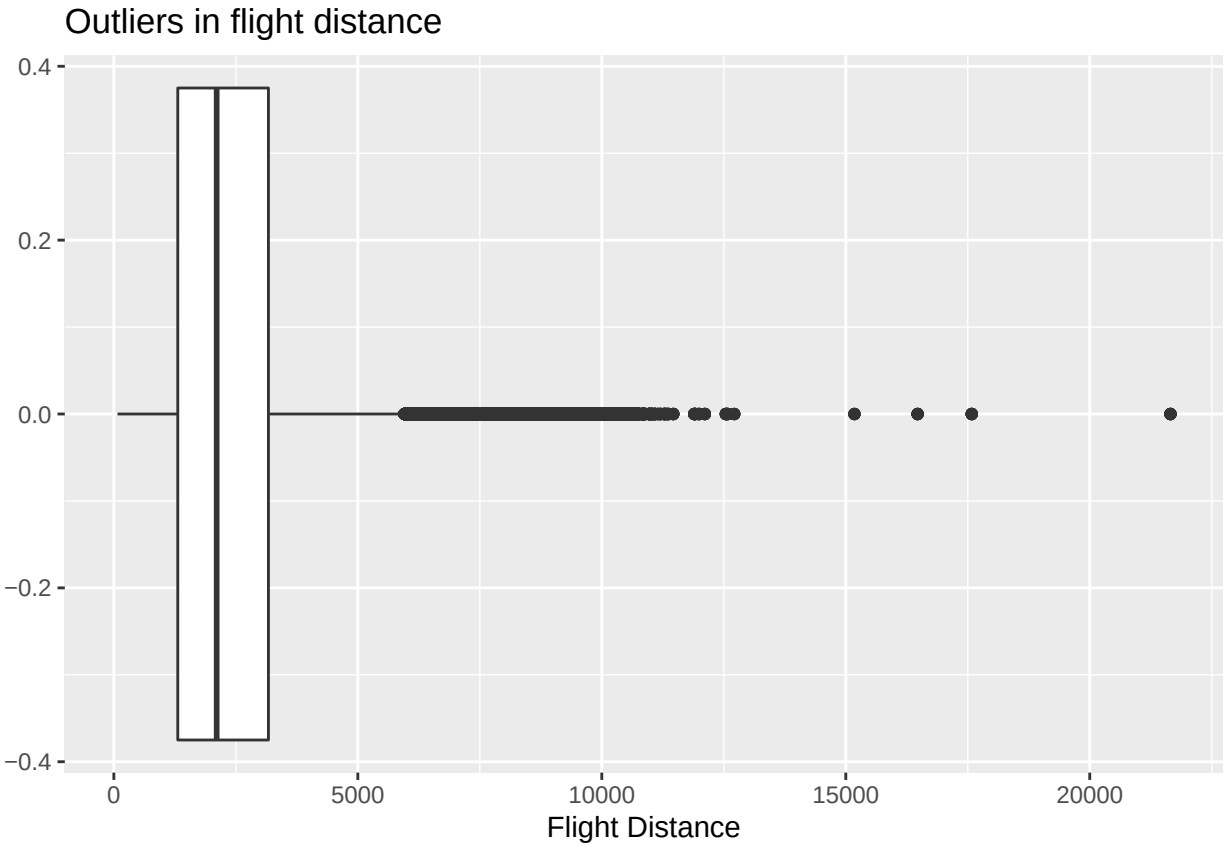
Individual Variables

As I described as a problem, there are outliers in both flight fare and distance. In this section, I would like to visualize them by making plots and handle them.

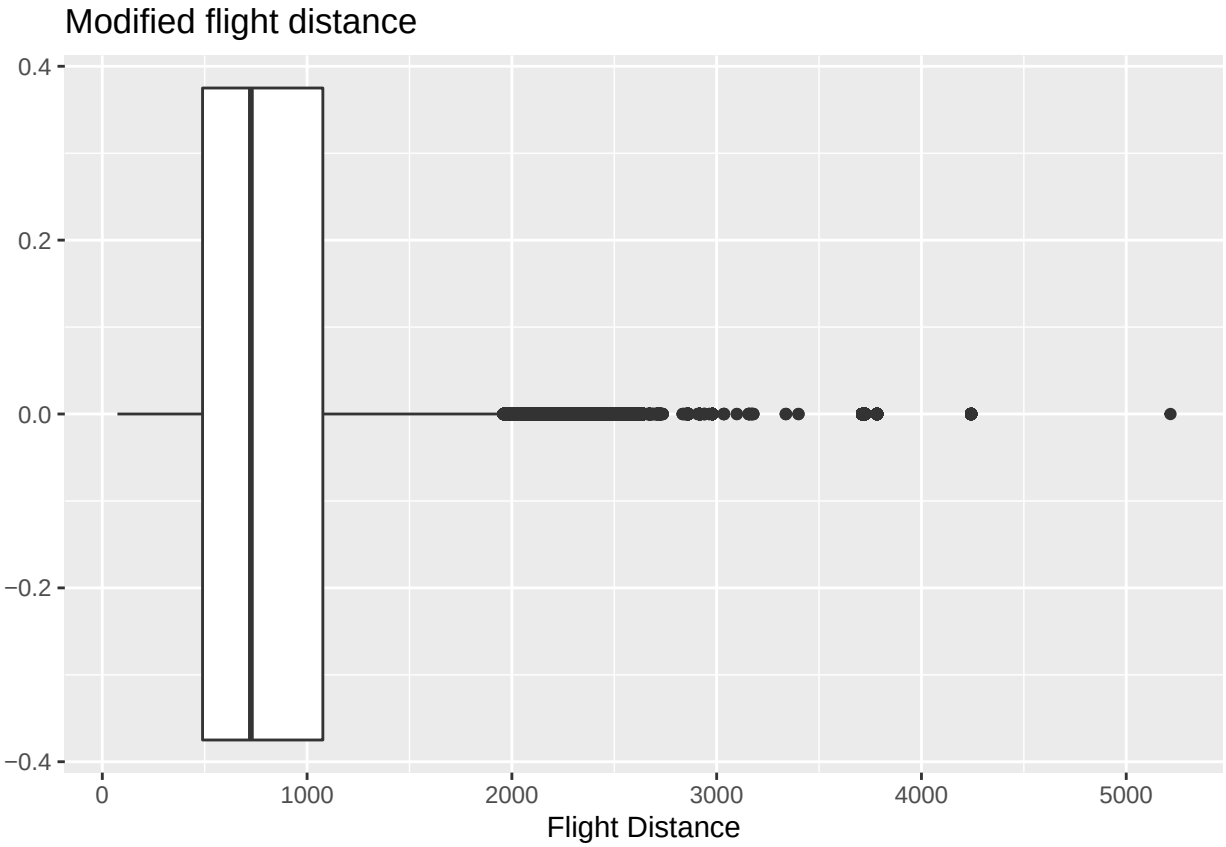
As this first plot shows there are some extremely expensive flights. For example, as far as we can see, there are two flights that cost around $70,000 - 80,000, and numerous flights that cost around $10,000. Since those are domestic flights, those prices are just too expensive.



The next plot shows the outliers in flight distance. Looking at the distance, there are extremely long travels as well. For example, there are many flights that traveled around 10,000 miles. A distance of 10,000 miles is about as long as the trip from New York to Sydney, Australia. Thus, it is certain that a distance of 10,000 miles is way too long for domestic flights.



First, to deal with the ridiculously long flight distance, I used the column "num_of_flight" in the data_new dataset. Again, it simply shows the number of flights. I divided distance by number of flights that each trip has. This turned out to solve the problem pretty well. There are some flights with more than 4000 miles of traveled distance, but as the table shows, most of the flights are going or coming from Hawaii to the mainland of the U.S. except for the flight from MSP to MIA with 5,216 miles traveled. I am not sure why the flight has such a long distance. However, since dividing the total distance traveled by the number of flight worked pretty well for the other flights, I decided to leave it.



itin_id	origin	dest	distance_per_flight
20201611596	MSP	MIA	5216
20201394325	HNL	ORD	4243
20201394326	HNL	ORD	4243
20201394327	HNL	ORD	4243
20201394328	HNL	ORD	4243
20201394329	HNL	ORD	4243
20201394330	HNL	ORD	4243
20201394331	HNL	ORD	4243
20201394332	HNL	ORD	4243
20201394333	HNL	ORD	4243

In terms of the extremely expensive flight fares, I was assuming if other factors such as bulk fare or number of passenger would cause such expensive flight fares. In the BTS website, I could also download the variables "Passenger", which, according to the description, shows the number of passengers, but it contains the very low values such as 1 in most of the rows. Since it did not seem correct, I decided not to use the variable "Passenger."

For the very cheap flights, I assumed that bulk fare is the reason of it. I downloaded the "BulkFare" from the same website, but all the rows had only 0, which could not explain this problem.

Since the outliers of flight fare defies the explanations by using the available data, I decided to filter out those outliers for the better accuracy in predictive modeling. To determine the outliers, I used the website from Penn State. According to the website, outliers can be defined by determining the fences. The way of determine the fences is take 1.5 times the IQR and subtract the value from Q1 and add the value to Q3.

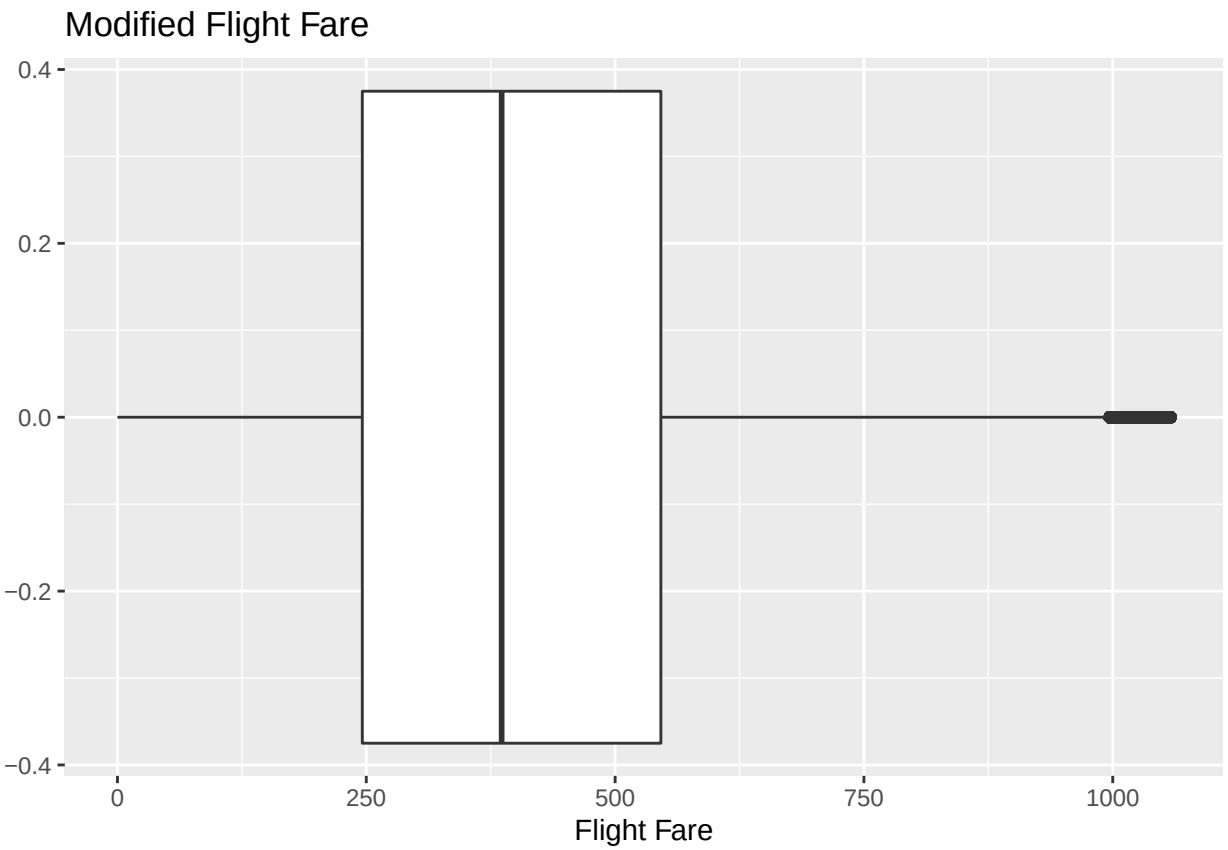
I had R compute those values, and IQR is 322, Q1 is 254, and Q3 is 576. From this, the lower fence for the

outlier is 254 - 322 * 1.5 = -299, and the upper fence is 576 + 322 * 1.5 = 1059. Therefore, I filtered out the flights that are outside the fences.

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##     0.0   254.0   398.0   456.4   576.0 78000.0
```

```
## [1] 322
```

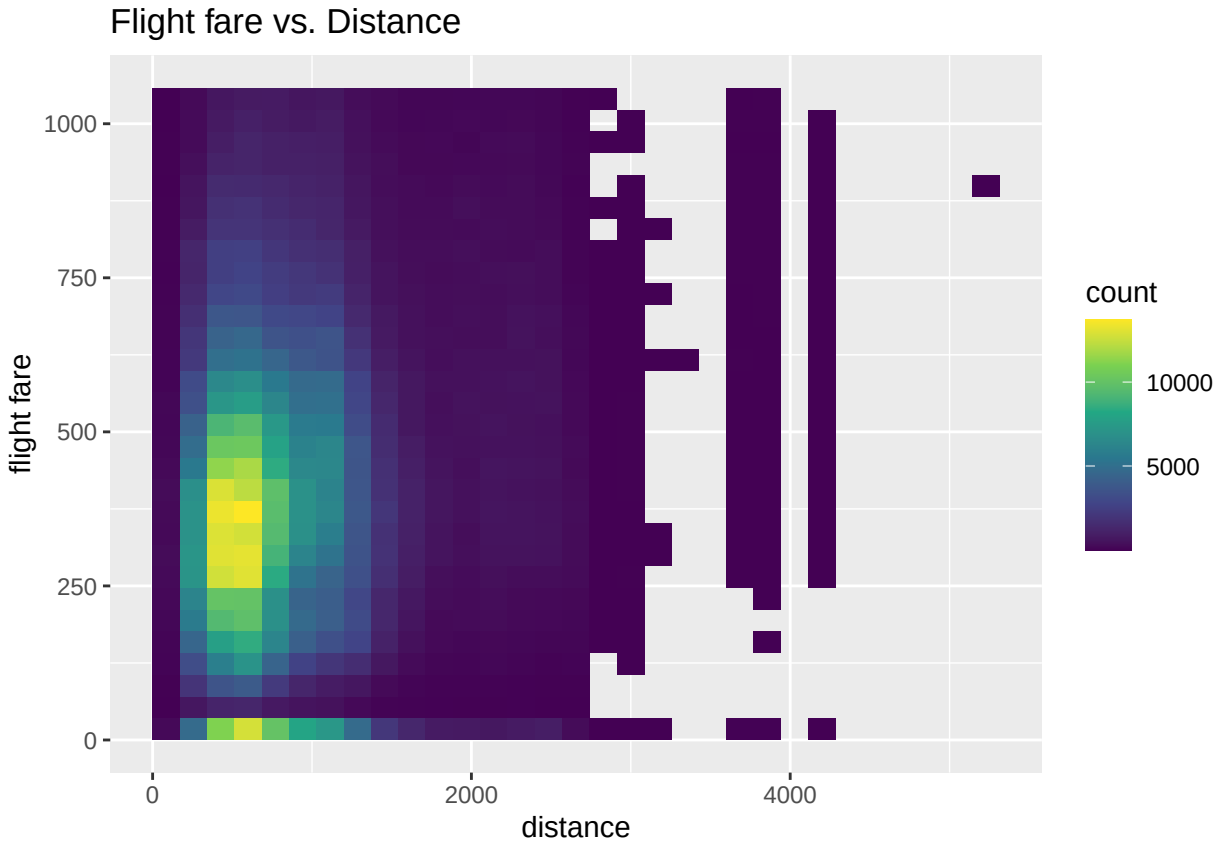
```
flight <- flight %>%
  filter(itin_fare < 1059)
```



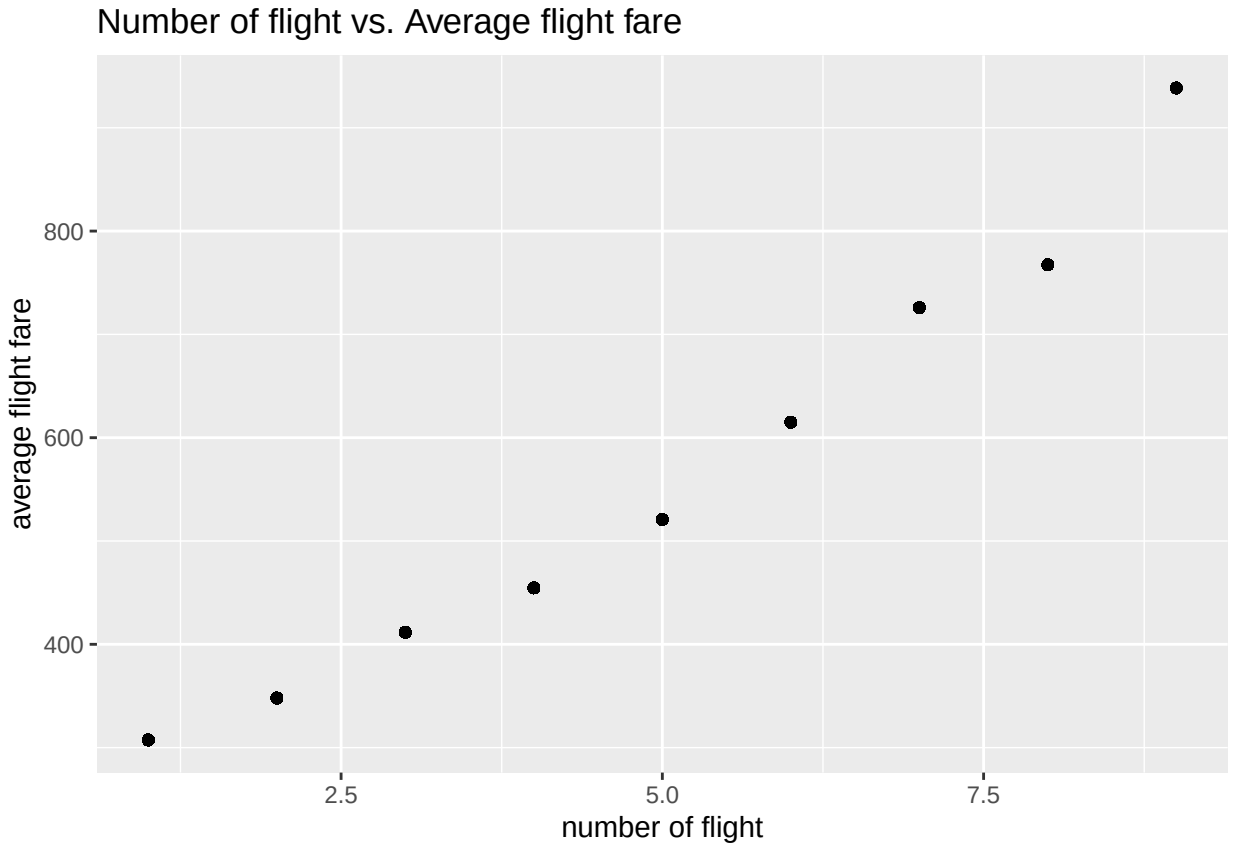
Now that the outliers are eliminated, and the boxplot for flight fare is visible.

Combination of Variables

For this part, I would like to make the bivariate plots to examine the relationships between flight fare and different variables. If I could find the correlation between them, it enables me to use a linear regression model for this project.



This density plot shows the relationship between flight fare and distance. I used density plot because there were too many points and when showing them with a scatter plot, overplot happens. To avoid it, I used the density plot instead. As the plot shows, it is obvious that there is a positive correlation between them. Specifically, flights with longer distance tend to have the more expensive fare.



The second bivariate plot shows the relationship between the number of flight and its average flight fare. To make this plot, I used group_by and mutate functions to make a new column that shows the mean flight fare for each number of the flights. A clear positive correlation can be observed between them.

Modeling

Baseline

I decided to use linear regression model for the predictive modeling for this project. Based on the charts from the previous steps, there are positive correlations between distance and flight fare, and number of flight and flight fare. Due to this, linear regression model is more appropriate for this project compared to decision tree or classification. From using the linear regression, I expect to get somewhat accurate prediction ($< \$100$) because of the large amount of data and positive correlations I observed.

The codes for making predictive modeling for this project were retrieved and edited from Lab10 created by Professor Kenneth Arnold at Calvin University.

Model1

The target variable for this model is the flight fare. First, I would like to use distance and number_of_flight to predict it. After making the model, I would like to measure the accuracy by calculating the mean absolute error by using cross-validation. Since cross-validation can be used when measuring accuracy on unseen data, it is appropriate to use for my model.

id	.metric	.estimator	.estimate	.config
Fold01	mae	standard	168.7873	Preprocessor1_Model1

id	.metric	.estimator	.estimate	.config
Fold02	mae	standard	170.2185	Preprocessor1_Model1
Fold03	mae	standard	168.6211	Preprocessor1_Model1
Fold04	mae	standard	170.3376	Preprocessor1_Model1
Fold05	mae	standard	166.8328	Preprocessor1_Model1
Fold06	mae	standard	169.3298	Preprocessor1_Model1
Fold07	mae	standard	168.4466	Preprocessor1_Model1
Fold08	mae	standard	168.2758	Preprocessor1_Model1
Fold09	mae	standard	168.6182	Preprocessor1_Model1
Fold10	mae	standard	168.3642	Preprocessor1_Model1

.metric	.estimator	mean	n	std_err	.config
mae	standard	168.7832	10	0.319357	Preprocessor1_Model1

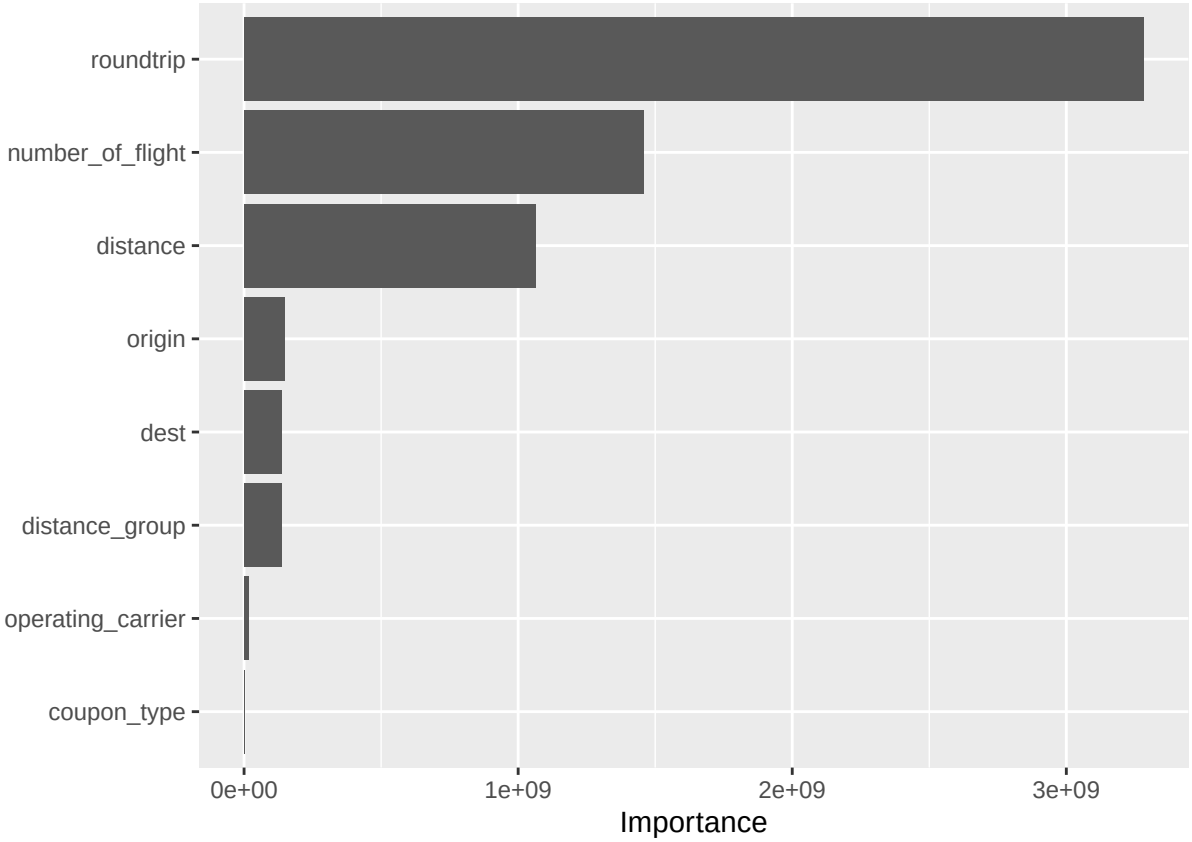
The average MAE for the first model is 168.6, and I would not say this is a good model. The average error of $168 is quite big for the flight fare. Therefore, I would add more variables to make the model better in the next step.

Refined

In this step, I am going to make a plot that shows which variables are more important. This gives me the idea of what variables affect the flight fare greatly.

The codes for making this graph was retrieved and edited from this lecture by Professor Kenneth Arnold at Calvin University.

```
## Warning: `pull_workflow_fit()` was deprecated in workflows 0.2.3.
## Please use `extract_fit_parsnip()` instead.
```



This chart shows that whether the flight is a round trip or not is the most important factor followed by number of flights, and distance. Therefore, for the new model, I am going to add roundtrip to the first model and see how much the new model improves.

Model2

The target variable for this model is the flight fare. I would like to use distance, number_of_flight, and roundtrip to predict the flight fare. After making the model, I would like to measure the accuracy by calculating the mean absolute error by using cross-validation just like what I did for model1.

id	.metric	.estimator	.estimate	.config
Fold01	mae	standard	165.5722	Preprocessor1_Model1
Fold02	mae	standard	166.7104	Preprocessor1_Model1
Fold03	mae	standard	164.9572	Preprocessor1_Model1
Fold04	mae	standard	166.7198	Preprocessor1_Model1
Fold05	mae	standard	163.6012	Preprocessor1_Model1
Fold06	mae	standard	165.8281	Preprocessor1_Model1
Fold07	mae	standard	165.2067	Preprocessor1_Model1
Fold08	mae	standard	164.7800	Preprocessor1_Model1
Fold09	mae	standard	164.9207	Preprocessor1_Model1
Fold10	mae	standard	164.9668	Preprocessor1_Model1

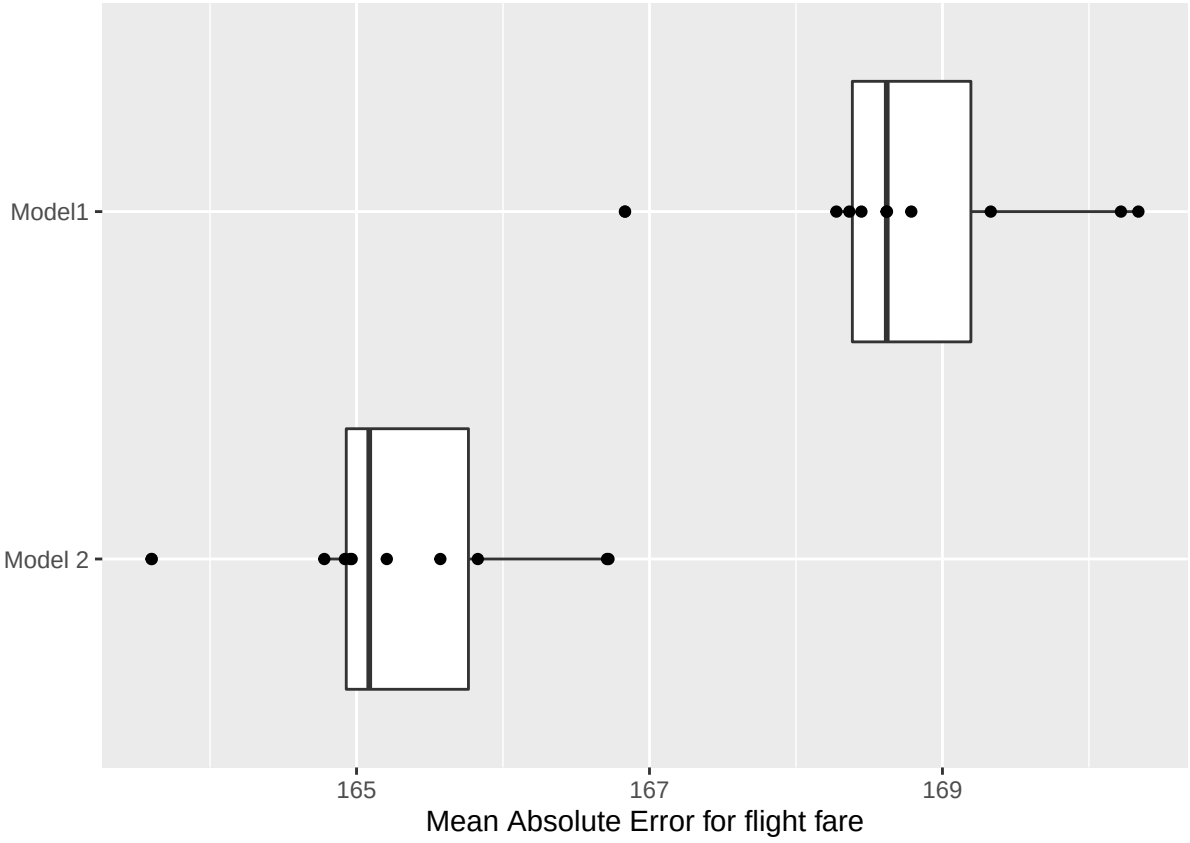
.metric	.estimator	mean	n	std_err	.config
mae	standard	165.3263	10	0.2960634	Preprocessor1_Model1

The average MAE for the new model is 165.1, which is slightly better than the first model. However, the error is still big.

Summary

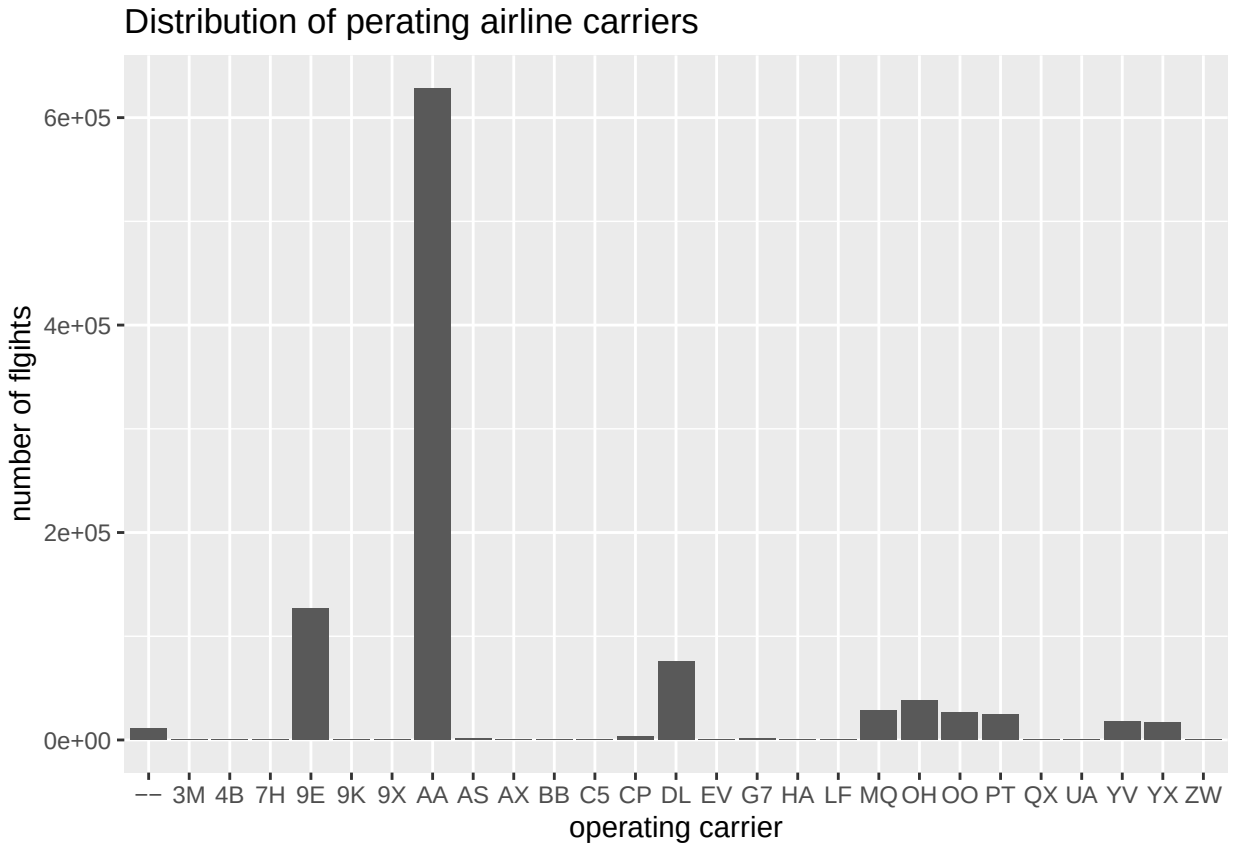
Discussion of Findings

The most important variable to determine the flight price is whether the flight is a round trip or not. It was surprising that it is almost two times more important than distance traveled. As the chart below shows, the first predictive model has the MAE of 168.6, and the second model has that of 165.1. The MAE for the new model decreased by 3.5 by adding roundtrip. However, I still think the MAE of 165.1 is pretty big. I assume this is because of the lack of the useful variables. Overall, considering that my model depends only on the distance, number of flight, and round trip status, I think my model did a good job predicting the flight fare.



Limitations and Ethical Considerations

One limitation comes from the lack of diversity of flight carriers. As the chart below shows, there are some airlines with many records while others have a small number of records. For example, American Airlines (AA) has 628,560 flights, and Endeavor Air (9E) has 126,717 flights reported. However, other carriers have very small numbers of flights compared to those. I believe the actual proportional shares of the flights in the real world are not the same as the chart shows. Since the predictive modeling for this project is greatly influenced by the number of records, it is biased toward the airlines with a large number of records.



Future Directions

For the future research, I would like to add more variables such as number of passengers, day of the week, and number of in-flight meals to make the model more accurate. After this, I would like to apply the model to international flight data to see if there are any different patterns between domestic and international flights. For example, the most important variable for domestic fights is whether it is a round trip or not, so I want to find out if this is the same for international flights as well.

Appendix

```
knitr::opts_chunk$set(echo = FALSE)
library(tidyverse)
library(tidymodels)
library(knitr)
data1 <- read.csv("data/data1.csv") %>%
  janitor::clean_names()
data2 <- read.csv("data/data2.csv") %>%
  janitor::clean_names()
data1_head <- head(data1)
kable(data1_head)
data2_head <- head(data2)
kable(data2_head)
data_new <- data1 %>%
  group_by(itin_id) %>%
```

```
  mutate(number_of_flight = sum(itin_id) / itin_id)
data_combined <- data_new %>%
  inner_join(data2, by = c("itin_id"))

combined_head <- head(data_combined)
kable(combined_head)
flight <- data_combined %>%
  select(itin_id, origin, dest, coupon_type, operating_carrier, distance, distance_group, number_of_flig

flight_head <- head(flight)
kable(flight_head)
flight %>%
  ggplot(aes(x = itin_fare)) +
  geom_boxplot() +
  labs(title = "Outliers in flight fare", x = "Flight Fare")
flight %>%
  ggplot(aes(x = distance)) +
  geom_boxplot() +
  labs(title = "Outliers in flight distance", x = "Flight Distance")
flight %>%
  ggplot(aes(x = distance/number_of_flight)) +
  geom_boxplot() +
  labs(title = "Modified flight distance", x = "Flight Distance")
table <- flight %>%
  mutate(distance_per_flight = distance / number_of_flight) %>%
  select(origin, dest, distance_per_flight) %>%
  filter(distance_per_flight > 4000) %>%
  arrange(-distance_per_flight) %>%
  head(10)

kable(table)
fare = flight$itin_fare
summary(fare)
IQR(fare)
flight <- flight %>%
  filter(itin_fare < 1059)
flight %>%
  ggplot(aes(x = itin_fare)) +
  geom_boxplot() +
  labs(title = "Modified Flight Fare", x = "Flight Fare")
flight %>%
  ggplot(aes(x = distance/number_of_flight, y = itin_fare)) +
  geom_bin2d() +
  scale_fill_continuous(type = "viridis") +
  labs(title = "Flight fare vs. Distance", x = "distance", y = "flight fare")
flight %>%
  group_by(number_of_flight) %>%
  mutate(mean_fare = mean(itin_fare)) %>%
  ggplot(aes(x = number_of_flight, y = mean_fare)) +
  geom_point() +
  labs(title = "Number of flight vs. Average flight fare", x = "number of flight", y = "average flight f
# First, divide the data to train set and test set
set.seed(10)
```

```r
flight_split <- initial_split(flight, prop = 2/3)
flight_train <- training(flight_split)
flight_test <- testing(flight_split)
# Declear cross validation
flight_resamples <- flight_train %>%
  vfold_cv(v = 10)

modeling_results <- tibble()
append_new_results <- function(modeling_results, model_name, spec, cv_results) {
  if (nrow(modeling_results) > 0 && model_name %in% modeling_results$model_name)
    stop(paste0(
      "There's already results for a model with the name ", model_name,
      ". Did you forget to change the name?"))

  bind_rows(
    modeling_results %>% mutate(model_name = as.character(model_name)),
    tibble(
      model_name = model_name,
      spec = list(spec),
      cv_results %>% select(-.estimator)
    )
  ) %>%
    mutate(model_name = as_factor(model_name)) # Ensure that factor level matches insertion order.
}

get_spec_for_model <- function(model_name) {
  modeling_results %>% filter(model_name == !!model_name) %>% purrr::chuck("spec", 1)
}

add_predictions <- function(data, ...) {
  imap_dfr(
    rlang::dots_list(..., .named = TRUE),
    function(model, model_name) {
      model %>%
        predict(data) %>%
        bind_cols(data) %>%
        mutate(model = !!model_name)
    }
  )
}

sweep_model_examples <- function(model, dataset, vars_to_sweep, examples = slice_sample(dataset, n = 10)
  X <- map_dfr(vars_to_sweep, function(v) {
    var_to_sweep <- rlang::as_label(v)
    sweep_min <- min(dataset[[var_to_sweep]])
    sweep_max <- max(dataset[[var_to_sweep]])
    expand_grid(
      examples %>% select(-!!var_to_sweep) %>% mutate(.idx = row_number()),
      !!enquo(var_to_sweep) := seq(sweep_min, sweep_max, length.out = 500)) %>%
      mutate(sweep_var = var_to_sweep, .sweep_val = .data[[var_to_sweep]])
  })
  model %>%
    predict(X) %>%
```

```
    bind_cols(X)
}

linear_reg <- function(engine = "lm", ...) {
  parsnip::linear_reg(...) %>% set_engine(engine)
}

decision_tree <- function(mode = "regression", engine = "rpart", ...) {
  parsnip::decision_tree(mode = "regression", ...) %>%
    set_engine(engine)
}
spec <- workflow() %>%
  add_recipe(
    recipe(itin_fare ~ distance + number_of_flight, data = flight_train)
  ) %>%
  add_model(
    linear_reg()
  )
# Compute the cross-validation assessment scores
cv_results <- spec %>%
  fit_resamples(resamples = flight_resamples, metrics = metric_set(mae)) %>%
  collect_metrics(summarize = FALSE)

# Get the sumarized result
cv_summary <- spec %>%
  fit_resamples(resamples = flight_resamples, metrics = metric_set(mae)) %>%
  collect_metrics(summarize = TRUE)


kable(cv_results)
kable(cv_summary)
# Collect these results into our `modeling_results` data frame.
modeling_results <- modeling_results %>%
  append_new_results(
    model_name = "Model1",
    spec = spec,
    cv_results = cv_results
  )
# Create an importance graph
regression_workflow <- workflow() %>% add_model(decision_tree(mode = "regression") %>% set_engine('rpar

model <- regression_workflow %>%
  add_recipe(recipe(itin_fare ~ ., data = flight_train)) %>%
  fit(data = flight_train)

model %>% pull_workflow_fit() %>% vip::vip(num_features = 15L)
spec <- workflow() %>%
  add_recipe(
    recipe(itin_fare ~ distance + number_of_flight + roundtrip, data = flight_train)
  ) %>%
  add_model(
    linear_reg()
  )
```

```r
# Compute the cross-validation assessment scores
cv_results <-
  spec %>%
  fit_resamples(resamples = flight_resamples, metrics = metric_set(mae)) %>%
  collect_metrics(summarize = FALSE)

# Get the summarized result
cv_summary <- spec %>%
  fit_resamples(resamples = flight_resamples, metrics = metric_set(mae)) %>%
  collect_metrics(summarize = TRUE)

kable(cv_results)
kable(cv_summary)
# Collect these results into our `modeling_results` data frame
modeling_results <- modeling_results %>%
  append_new_results(
    model_name = "Model 2",
    spec = spec,
    cv_results = cv_results
  )
modeling_results %>%
  ggplot(aes(y = fct_rev(model_name), x = .estimate)) +
    geom_boxplot() +
    geom_point() +
    labs(totle = "Comparison of model performance",  x = "Mean Absolute Error for flight fare", y = "")
carrier <- flight %>%
  group_by(operating_carrier) %>%
  summarize(n = n())

carrier %>%
  ggplot(aes(x = operating_carrier, y = n)) +
  geom_bar(stat="identity") +
  labs(title = "Distribution of perating airline carriers", x = "operating carrier", y = "number of flg
```