Carnegie Mellon University
HeinzCollege
INFORMATION SYSTEMS • PUBLIC POLICY • MANAGEMENT

QL2 Capstone Project

Final Report

Project Advisor :
Professor Beibei Li

Project Manager :
Sukriti Bharti

Team Members :
Bramantyo Danur Jati
Jiabin Chen
Raghav Sood
Shruti Karandikar
Yuhan He

Heinz College
Carnegie Mellon University
2020

TABLE OF CONTENTS

Executive Summary

QL2 Software LLC. collects real time pricing information across the travel industry including airlines and car rentals. Their clients use this data to inform their tactical and strategic pricing decisions to beat their competition and more effectively price.

A consistent theory is that these two industries' prices are correlated, based on the assumption that travelers first book flights and then rent cars. As per the theory, there is a measurable relationship between pricing trends in airlines that can inform future pricing trends in car rental.

As a part of the Price Prediction Project, we analyzed the data gathered for these two industries to understand and highlight relationships or patterns. We also performed feature engineering to transform the raw data and used Machine Learning algorithms to form the basis of our findings.

From the results obtained, we can infer that the airline prices can be instrumental in predicting the car rental prices. The accuracy of the prediction can be made better with further enhanced feature engineering, reduced aggregations, and advanced time series analysis algorithms.

The outcome of our analysis will help QL2 Software to help their clients in identifying potential changes in travel demand far earlier and adjusting their pricing and inventory decisions.

Confidentiality Disclaimer

The content of this report is confidential and intended for the recipients only. It is strictly forbidden to share any part of this report with any third party, without a written consent from QL2 Software LLC.

I. INTRODUCTION

Our Client: QL2 Software LLC.

Our client for this project, QL2 Software delivers true competitive advantage through on-demand data extraction, product matching, and actionable insights. Their comprehensive real-time analytics help clients make profitable decisions that outsmart their competition.

The data they collect is public, real-time and is not static or cached. They maintain a large data repository with advanced request analytics and analyze source traffic in real time to ensure quality and up time.

QL2's approach to business is simple. They ensure that their clients are always able to maintain or improve their market position with the quickest, most comprehensive competitive insights.

Problem Statement

At the core, QL2's value to their clients is based on their ability to help them optimize their pricing and pricing strategies. Currently, they primarily do that through delivering insights into their clients' competitors' pricing and their relative position. This limits the analysis within a given industry.

QL2 hypothesizes that the price trends in airline and car rental industries are interconnected. If this hypothesis is validated, they will be able to delve deeper into the data and provide improved insights to the clients which would help them take optimal pricing and strategy decisions.

This project goal was to validate our client's hypothesis and quantify any relationship between the two industries, justified with data.

Business Value

This analysis will be the beginning of establishing broader travel trends that are applicable across multiple verticals.

It would enable QL2's clients to identify potential changes in travel demand down to the market level rather than basing their strategy off the movement within their industry.

It will potentially identify shifts in demand far earlier and allow them to react in their pricing and inventory decisions

Objectives

The major objectives for this project were to:

1. Understand the relationship between Airfare data and Car Rental Data and then

2. Build a classification model, so that we can predict the increase or decrease in car rental price based on the airline price.

This would establish a base and form the foundation to build predictive models that would be capable of predicting the value of increase or decrease in the car rental prices using the airline prices. The business level inference of that further work could be, for instance, "If we see a 10% increase in airline ticket price 30 days before departure date, then we will see a 15% increase in car rental price 10 days before departure date"

Approach

The approach that we employed to achieve our target objectives is shown in **Fig.1.**

We initiated with extracting the data from the Snowflake instance that was provided to us by QL2. There were 2 tables that we retrieved the data from. Once we had the data, we explored it to understand the general trends in terms of locations, airlines or agencies, time of the year etc. We then drilled down to the city level analysis using the data for Los Angeles. We further looked for patterns and fluctuations for this specific location.

Once we had a fair idea of the city level data, we used Regression to come up with a couple of models and analyzed the information provided by them. Post this, we moved onto Classification models and did literature review on what all models would fit our use case. We built our baseline classification model for the car rental data, which formed the foundation for further feature engineering and improved accuracy of prediction.

The improved model was built over the baseline model by including the airline data and tuning the hyper-parameters for the different models used. This helped us to reach model accuracy that was better than a random guess based on the data distribution.

With this improved model, we were able to make inferences and future work recommendations for this project.

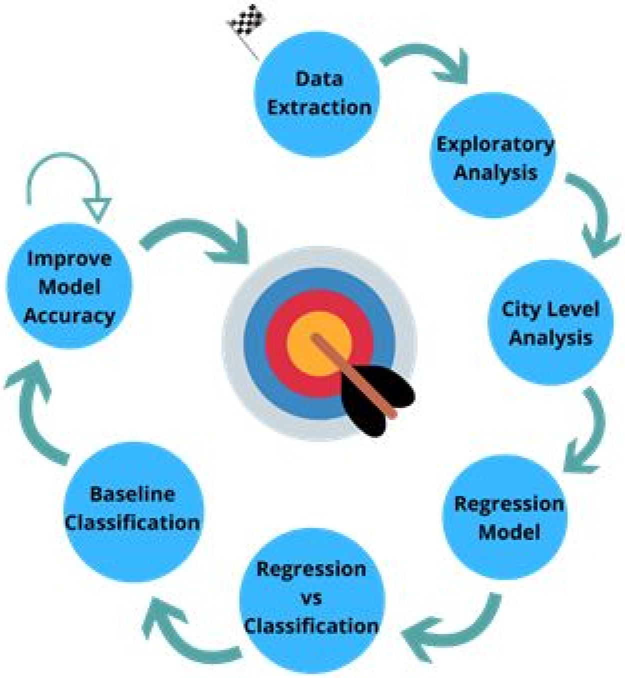
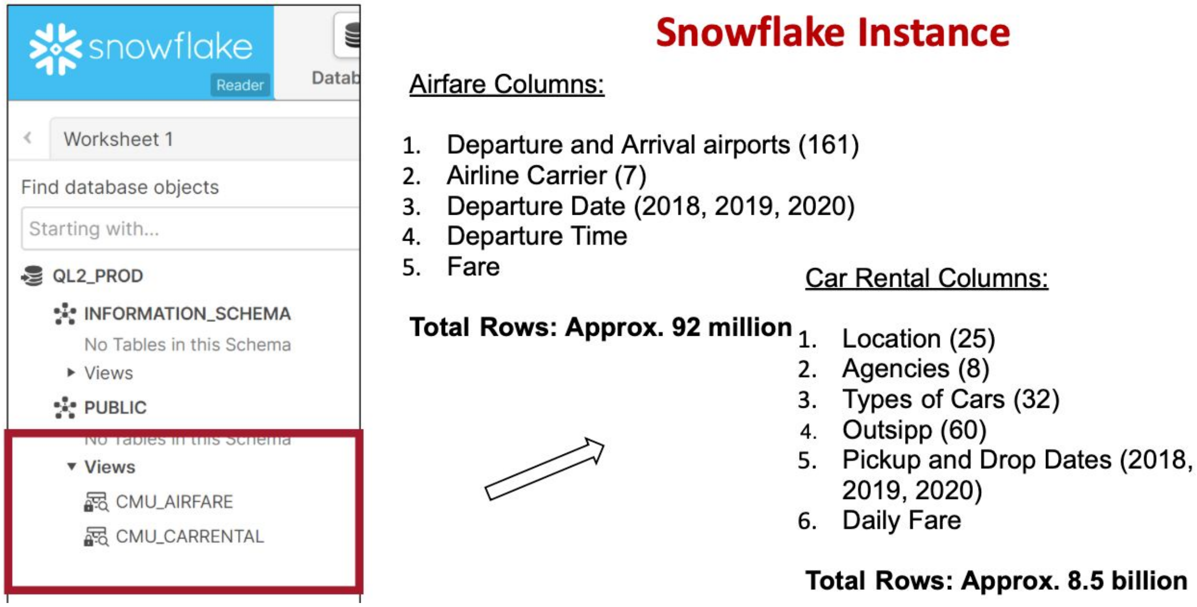


Fig. 1. Approach for Price Prediction Project

II. DATA AND EXPLORATORY ANALYSIS

Data at Hand



Snowflake Instance

Airfare Columns:

1. Departure and Arrival airports (161)
2. Airline Carrier (7)
3. Departure Date (2018, 2019, 2020)
4. Departure Time
5. Fare

Total Rows: Approx. 92 million

Car Rental Columns:

1. Location (25)
2. Agencies (8)
3. Types of Cars (32)
4. Outsipp (60)
5. Pickup and Drop Dates (2018, 2019, 2020)
6. Daily Fare

Total Rows: Approx. 8.5 billion

Two Verticals

Airfare

Airfare consists of data scraped for the flights booked to and from different airports scraped at different time intervals before departure like 15, 30, 45, 60 etc. The features here included departure and arrival airports, carriers, departure date, scraping date, time and fare.

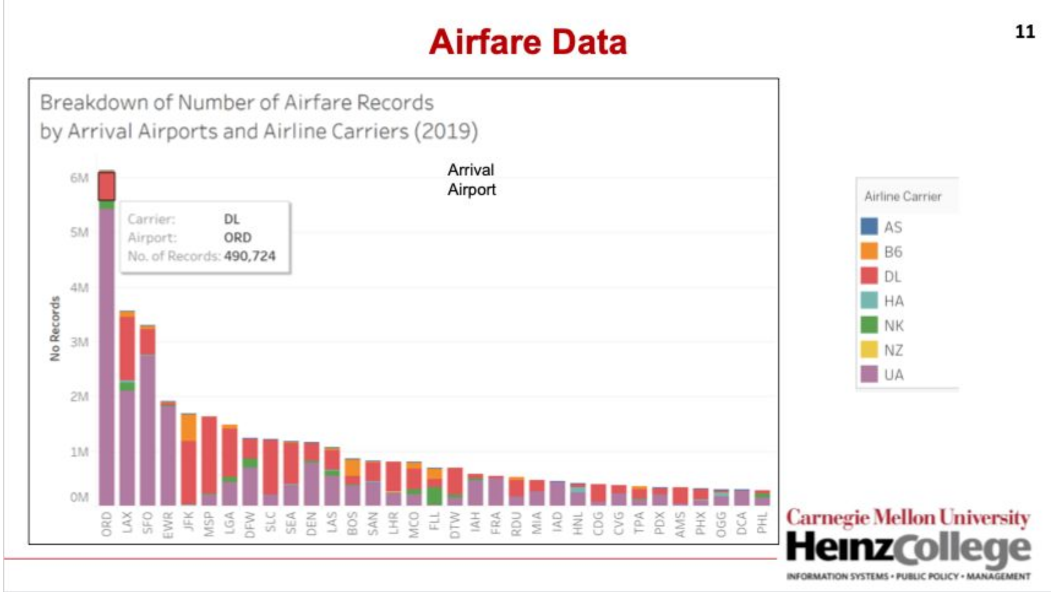
Car Rental

Car rental consists of data similar to airfare but does not consist of actual bookings but of different prices offered across the country for different dates scraped on a similar time before the actual departure. The features include Location, Agencies, Types of Car, Outsipp, pickup and drop dates and daily fare.

Exploratory Data Analysis

Visualizing both the domains to get a better understanding of the distribution

Fig 1



Airfare Data

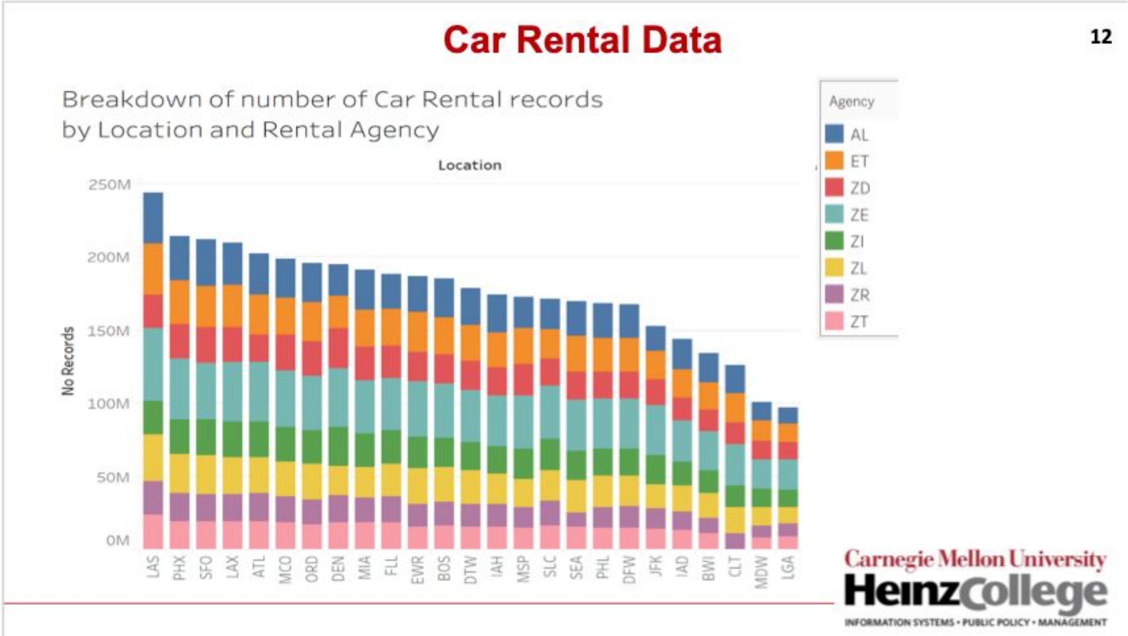
11

This chart plots the number of airfare records both incoming and outgoing flights for different cities in the US in 2019 split on carriers.

Things to notice:

- Chicago airport has the most number of records followed by Los Angeles, San Francisco etc.

Fig 2



Car Rental Data

12

Breakdown of number of Car Rental records
by Location and Rental Agency

Carnegie Mellon University
HeinzCollege
INFORMATION SYSTEMS • PUBLIC POLICY • MANAGEMENT

Similar to the previous chart, this chart plots the number of car rental records in 2019 across different cities in the US split on car agency.

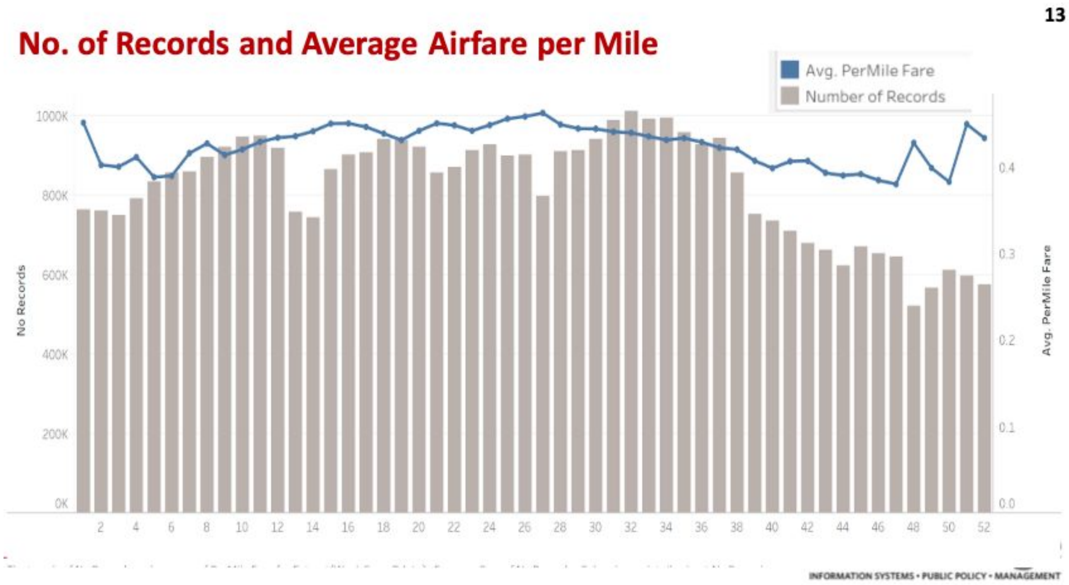
Things to notice:

- Las Vegas followed by Phoenix, San Francisco and Los Angeles has the most number of records.

Based on the above two graphs and a deep down distribution analysis we went ahead with LAX (Los Angeles) as our city of choice for further analysis as it provided a good amount of data both in car rental and airfare. It also shows an even distribution across different sets of features like air flight carriers, from airports, car agency's, and types of cars.

Going forward we will be looking at data from Los Angeles.

Fig 3



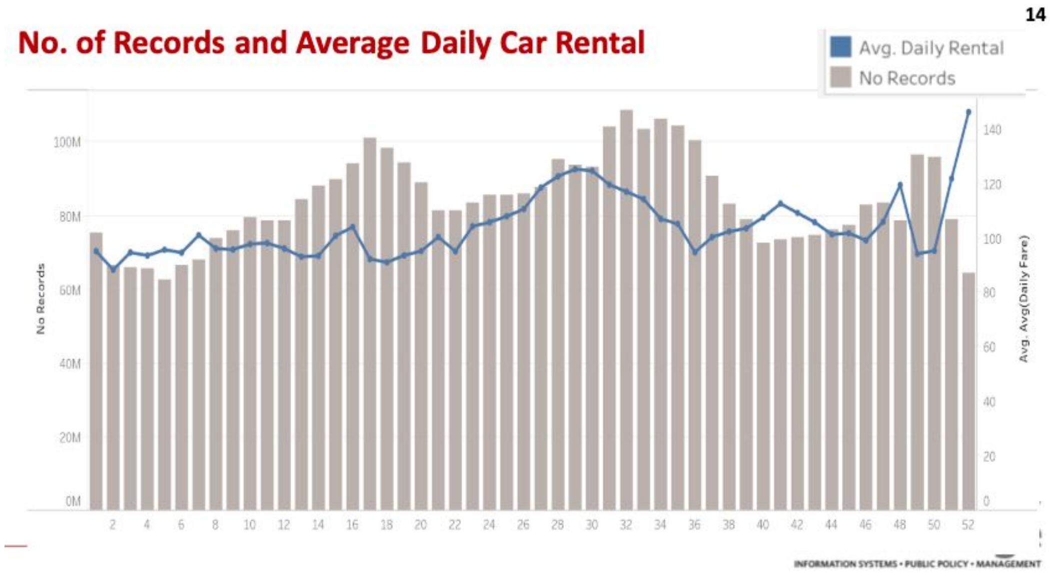
Here we analyze the air fare data for LA for all the weeks in 2019. The X-axis represents the week numbers 1-52. The bins here represent the number of records on the y1 axis and the blue trend line here represents the average Airfare price per mile for that week.

We noticed the price of a flight was clear function of distance, eg. flight from NY to LAX is more expensive than from SFO.

Things to notice:

- There is an increase in average price per mile in March i.e. spring break period(8-12) - this was common across different airports
- An expected increase in average price per mile towards the year end season (50-52) along with a decrease in number of flights towards the year end.

Fig 4



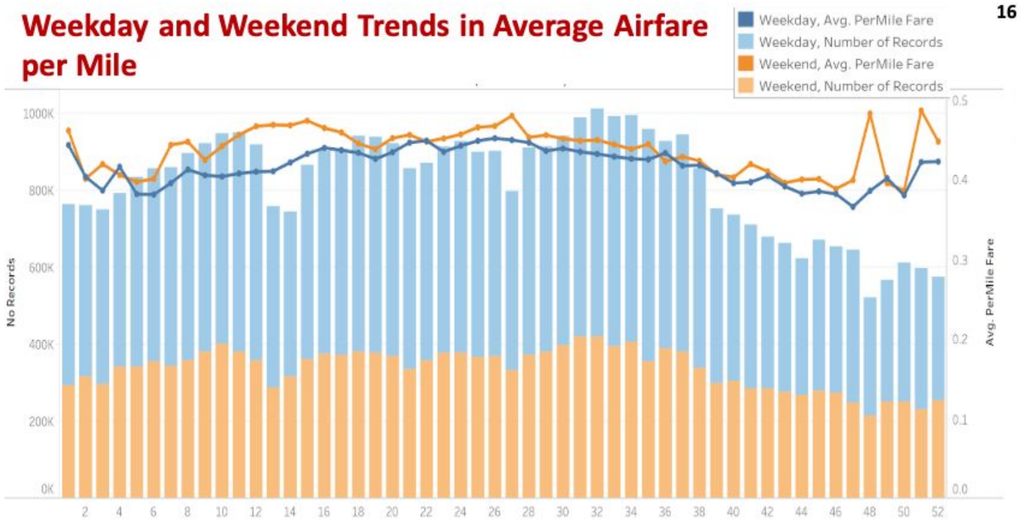
This graph plots the number of records for car rental data i.e the bins in y1 axis and average daily rental i.e the trend line in y2 axis across different weeks 1-52 in 2019.

Things to notice:

- Gradual increase in price from weeks 23-30 - upon research that showed holiday months increases demand of people renting cars
- Expected increase at the end of the year

The same graphs will now be split on weekdays vs weekends where weekend comprises Friday, Saturday and Sunday to get a better understanding of its impact.

Fig 5



This is a similar chart to what you noticed earlier for airfare data. Instead, here we have split the data on weekdays and weekends for airfare in 2019.

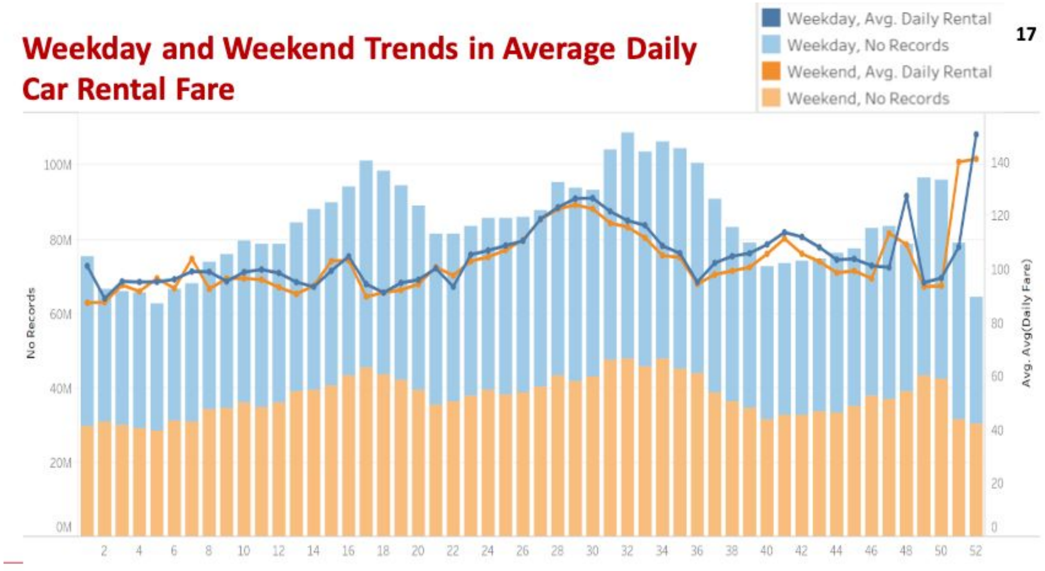
Here,

- The orange bins are weekends and the blue bins are weekdays representing the number of records
- Similarly the orange trend line is for weekends whereas the blue trend line is for weekdays.

Things to notice:

- Blue bins dominate orange throughout but that is because weekdays are more vs weekends hence counts will be more (4 vs 3)
- Trend line we see a clear domination of weekends over weekdays almost throughout the year proving our hunch that weekends tend to be costlier.

Fig 6

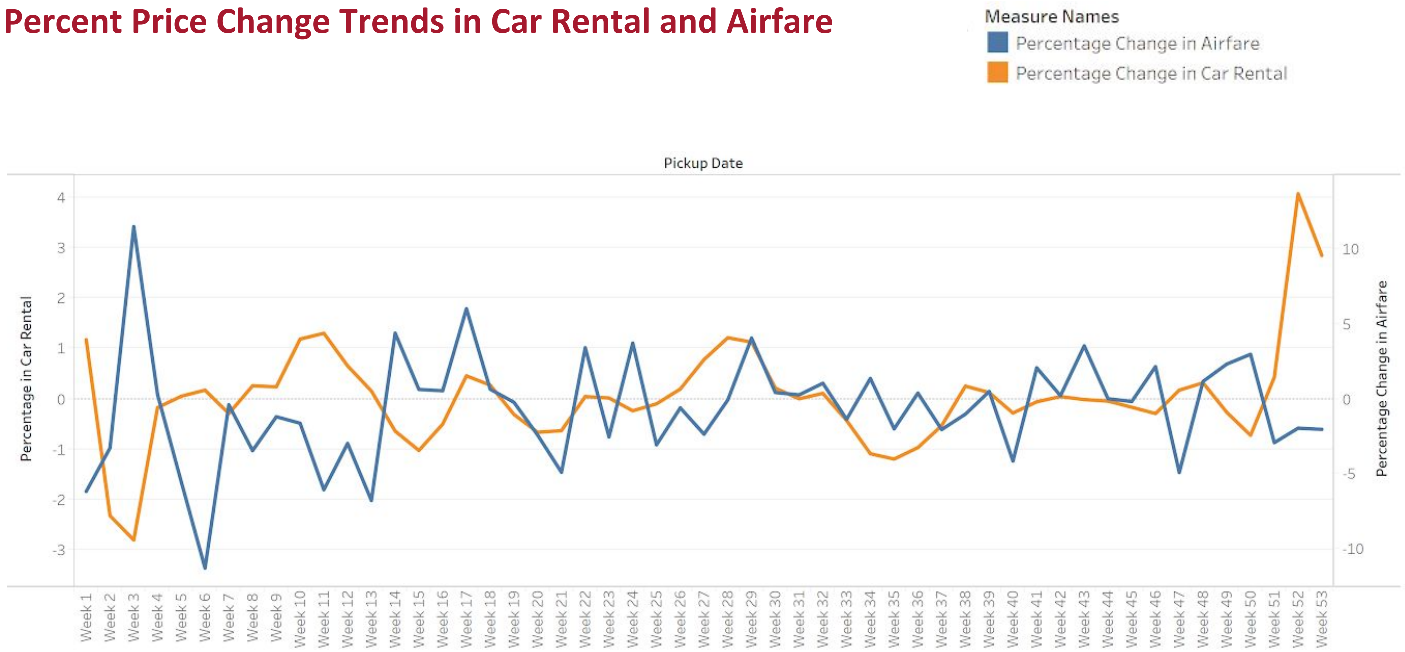


Similar breakdown for car rental data in 2019 on weekends.

Things to notice:

- There is not a clear domination for car rental price for weekends over weekdays. Upon further investigation we figured this is because a car rental unlike airfare lasts for more than a day which might or might not include a weekend thus nullifying its impact on trends.

Fig 7



Percent Price Change Trends in Car Rental and Airfare

This is a trend line for percent price change in car rental and airfare over a year

Here,

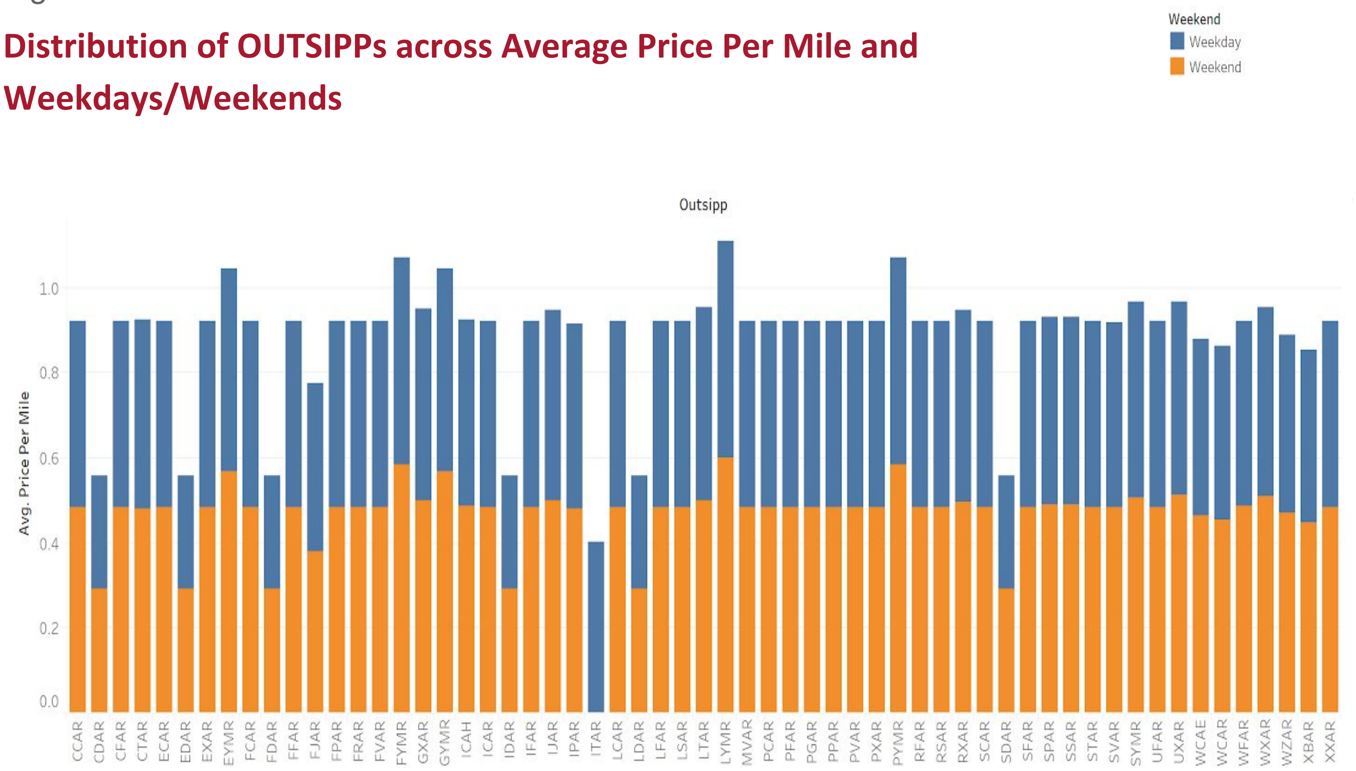
- The orange trend line represents car rental prices and blue trend line represents airfare
- Y-axis is the percent change in prices and not the absolute value for rental/airfare

Things to notice:

- Car rental price trend is smoother relative to airfare fluctuations
- Except for the first 10-12 weeks (~ First Quarter) the trends in % change go hand in hand, suggesting a correlation.

Fig 8

Distribution of OUTSIPPs across Average Price Per Mile and Weekdays/Weekends



This is a distribution of OUTSIPP v/s Average Price Per Mile across weekend and weekdays

Things to notice:

- Except for a couple of OUTSIPPs, most of the cars have similar average price per mile
- Barring a few OUTSIPPs, distribution of car types across weekday and weekends is similar[Which suggests we do not have any pattern wrt outsipp and weekday/weekend for LAX]

III. MACHINE LEARNING

Linear Regression

The first model that we tried is Linear Regression. We will do Linear Regression for car rental data only, airfare data only and on the merged model.

Airfare

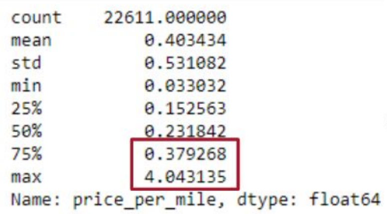
Linear Regression on the Airfare data only is trying to predict the airfare price based on some airfare features. We also want to measure the importance of each feature we were using. The data pre-processing that we did are :

- Convert Departure date into Week Number, Day of Week, and Weekend or Not.

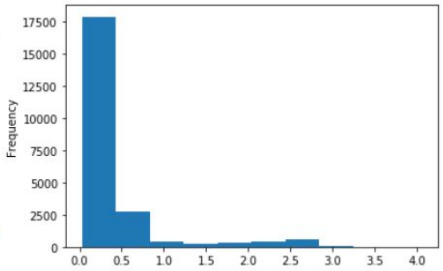
Departure Date		Week Number	Day of Week	Weekend
01-01-2019	→	1	2 (Tuesday)	0 (Not weekend)

- Calculate the price per mile by this formula. We found out from the histogram that it has a long tail, so we skip the top 5% data.

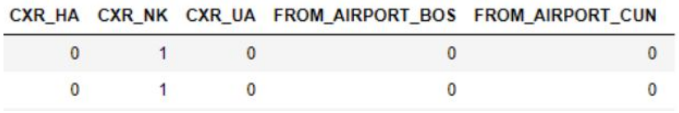
$$pricepermile(\$/mi) = \frac{airfare(\$)}{distance(mi)}$$



```
count    22611.000000
mean         0.403434
std          0.531082
min          0.033032
25%          0.152563
50%          0.231842
75%          0.379268
max          4.043135
Name: price_per_mile, dtype: float64
```



- Convert Airline Carrier and Departure Airport into one hot encoding.



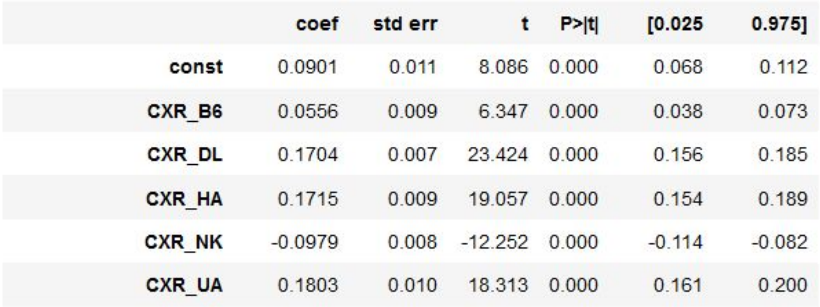
CXR_HA	CXR_NK	CXR_UA	FROM_AIRPORT_BOS	FROM_AIRPORT_CUN
0	1	0	0	0
0	1	0	0	0

The data pre-processing will results in below features :

Week_Number	: Convert to numerical (One hot encoding)
Day_of_Week	: 1 to 7, 1 is Monday, 7 is Sunday
Weekend	: If the record is weekend equals 1, otherwise it is 0
Airline_Carrier	: Convert to numerical (One hot encoding)
Departure_Airport	: Convert to numerical (One hot encoding)
Price_per_mile	: target dependent variable. Airfare($)/Distance(mil)

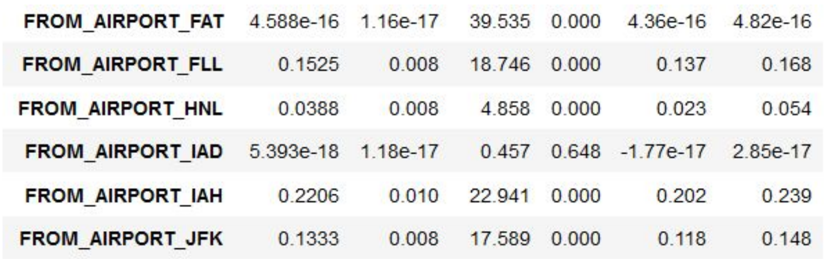
Price_per_mile = Function (Week_Number, Day_of_Week, Weekend,
Airline_Carrier, Departure_Airport)

Result
RMSE train : 0.1264129091689638
RMSE test : 0.30508142974805086



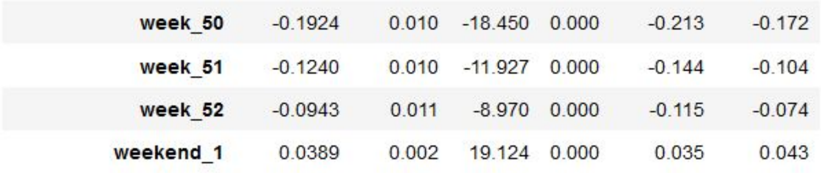
	coef	std err	t	P>|t|	[0.025	0.975]
const	0.0901	0.011	8.086	0.000	0.068	0.112
CXR_B6	0.0556	0.009	6.347	0.000	0.038	0.073
CXR_DL	0.1704	0.007	23.424	0.000	0.156	0.185
CXR_HA	0.1715	0.009	19.057	0.000	0.154	0.189
CXR_NK	-0.0979	0.008	-12.252	0.000	-0.114	-0.082
CXR_UA	0.1803	0.010	18.313	0.000	0.161	0.200

...



FROM_AIRPORT_FAT	4.588e-16	1.16e-17	39.535	0.000	4.36e-16	4.82e-16
FROM_AIRPORT_FLL	0.1525	0.008	18.746	0.000	0.137	0.168
FROM_AIRPORT_HNL	0.0388	0.008	4.858	0.000	0.023	0.054
FROM_AIRPORT_IAD	5.393e-18	1.18e-17	0.457	0.648	-1.77e-17	2.85e-17
FROM_AIRPORT_IAH	0.2206	0.010	22.941	0.000	0.202	0.239
FROM_AIRPORT_JFK	0.1333	0.008	17.589	0.000	0.118	0.148

...



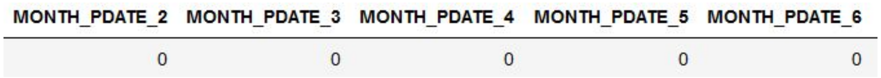
week_50	-0.1924	0.010	-18.450	0.000	-0.213	-0.172
week_51	-0.1240	0.010	-11.927	0.000	-0.144	-0.104
week_52	-0.0943	0.011	-8.970	0.000	-0.115	-0.074
weekend_1	0.0389	0.002	19.124	0.000	0.035	0.043

From the result we found out that all carriers give positive coefficients except for NK (Spirit Airlines). As for departure airport, IAD (Dulles) has the highest positive coefficients. And we know that the weekend has a higher price since it has positive coefficient.

Car Rental

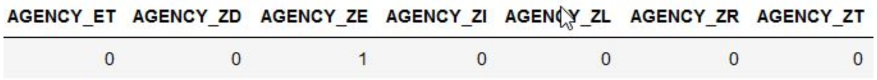
Linear Regression on the car rental data is trying to predict the car rental price based on some of the car rental features. We also want to measure the features impotence that we were using in the model. The data pre-processing that we did are :

- Extract Month from PDATE and convert it to one hot encoding.



MONTH_PDATE_2	MONTH_PDATE_3	MONTH_PDATE_4	MONTH_PDATE_5	MONTH_PDATE_6
0	0	0	0	0

- Convert Agency to one hot encoding.



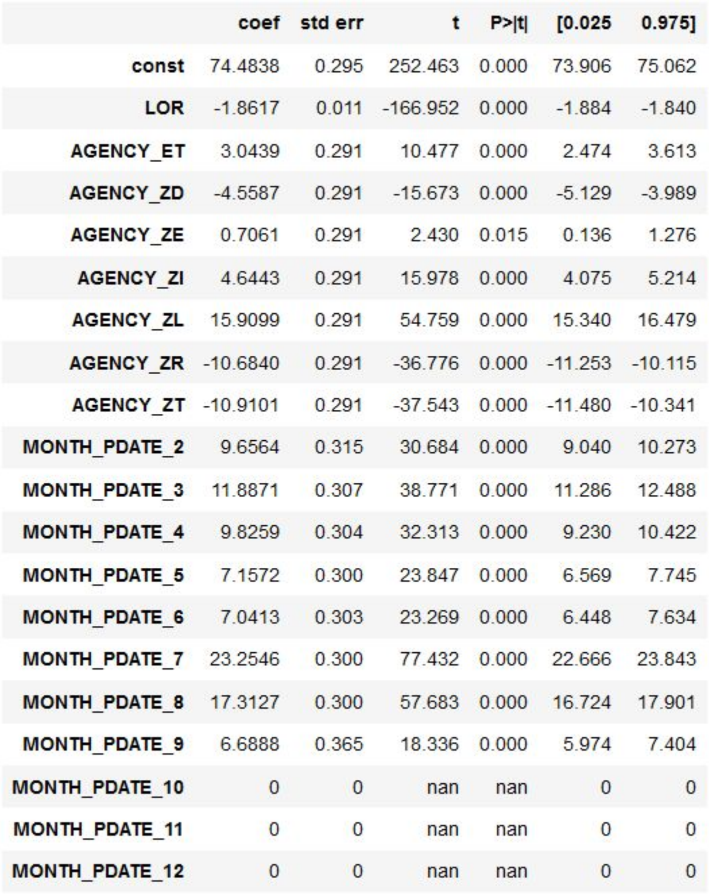
AGENCY_ET	AGENCY_ZD	AGENCY_ZE	AGENCY_ZI	AGENCY_ZL	AGENCY_ZR	AGENCY_ZT
0	0	1	0	0	0	0

The data pre-processing will results in below features :

LOR	: Length of Rental
Agency	: Convert to numerical (one hot encoding)
Month	: Convert to numerical (one hot encoding)
Price	: Target Dependent Variable

Price = Function(LOR, Agency, Month)

Result
RMSE train : 10.238060824942616
RMSE test : 18.96851552452373



	coef	std err	t	P>|t|	[0.025	0.975]
const	74.4838	0.295	252.463	0.000	73.906	75.062
LOR	-1.8617	0.011	-166.952	0.000	-1.884	-1.840
AGENCY_ET	3.0439	0.291	10.477	0.000	2.474	3.613
AGENCY_ZD	-4.5587	0.291	-15.673	0.000	-5.129	-3.989
AGENCY_ZE	0.7061	0.291	2.430	0.015	0.136	1.276
AGENCY_ZI	4.6443	0.291	15.978	0.000	4.075	5.214
AGENCY_ZL	15.9099	0.291	54.759	0.000	15.340	16.479
AGENCY_ZR	-10.6840	0.291	-36.776	0.000	-11.253	-10.115
AGENCY_ZT	-10.9101	0.291	-37.543	0.000	-11.480	-10.341
MONTH_PDATE_2	9.6564	0.315	30.684	0.000	9.040	10.273
MONTH_PDATE_3	11.8871	0.307	38.771	0.000	11.286	12.488
MONTH_PDATE_4	9.8259	0.304	32.313	0.000	9.230	10.422
MONTH_PDATE_5	7.1572	0.300	23.847	0.000	6.569	7.745
MONTH_PDATE_6	7.0413	0.303	23.269	0.000	6.448	7.634
MONTH_PDATE_7	23.2546	0.300	77.432	0.000	22.666	23.843
MONTH_PDATE_8	17.3127	0.300	57.683	0.000	16.724	17.901
MONTH_PDATE_9	6.6888	0.365	18.336	0.000	5.974	7.404
MONTH_PDATE_10	0	0	nan	nan	0	0
MONTH_PDATE_11	0	0	nan	nan	0	0
MONTH_PDATE_12	0	0	nan	nan	0	0

From the result we found out that :

- LOR is negatively correlated with price. The longer the length of rental is, the lower the price.
- ZL (National Car Rental) is the most expensive car rental, while ZT (Thrifty Car Rental) is the least expensive.

- There is a trend of price increase during July (summer holiday) and December (Christmas holiday).

Merged Dataset

Linear Regression on the merged dataset trying to predict the car rental price from some features from both verticals. For the merged datasets we took the significant features from both verticals then merged it on the Week and Day of Week columns. Here are the features that were brought from each vertical :

Source	Features	Remarks
Car Rental Data	Dummy variables for 7 Car Agencies Dummy variables for 52 weeks	One hot encoding One hot encoding
Airfare Data	Weekend or Not Average air price	Weekend = 1, Weekday = 0 Price per mile
New Features (Aggregated)	Number of flights arrived at LAX Number of car rentals at LAX	Aggregated for that date Aggregated for that date

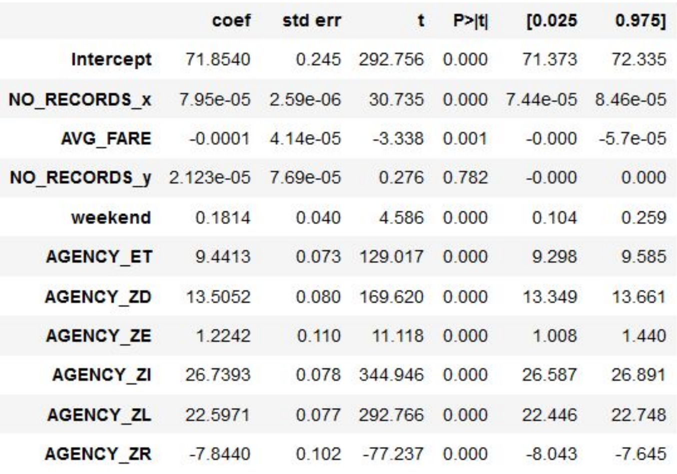
To avoid overfitting, we remove a few categorical columns such as Day of Week, From Airport, Air Carrier, and Car Type. Air Carriers were removed due to 2 out of 6 being insignificant, while the other air carriers had small coefficients. As for car type, we restricted our analysis only to the most popular car in LAX.

Variables

Agency	: Convert to numerical (one hot encoding)
Week_no	: Convert to numerical (one hot encoding)
Weekend	: If the record is weekend equals 1, otherwise it is 0
No_of_records_x	: Number of flights arriving at LAX
No_of_records_y	: Number of cars rented at LAX
Avg_daily_fare	: Target dependent variable

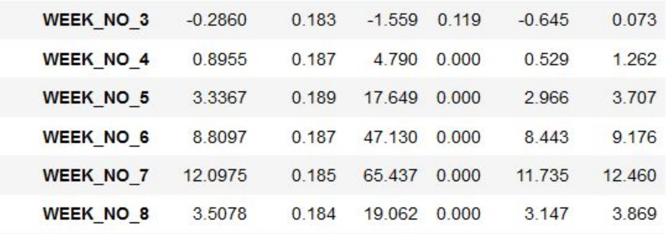
Avg_daily_fare = Function(Agency, Week_no, Weekend, No_of_records_x, No_of_records_y)

Result



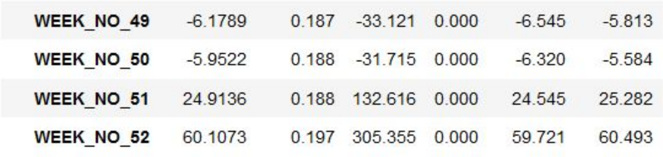
	coef	std err	t	P>|t|	[0.025	0.975]
Intercept	71.8540	0.245	292.756	0.000	71.373	72.335
NO_RECORDS_x	7.95e-05	2.59e-06	30.735	0.000	7.44e-05	8.46e-05
AVG_FARE	-0.0001	4.14e-05	-3.338	0.001	-0.000	-5.7e-05
NO_RECORDS_y	2.123e-05	7.69e-05	0.276	0.782	-0.000	0.000
weekend	0.1814	0.040	4.586	0.000	0.104	0.259
AGENCY_ET	9.4413	0.073	129.017	0.000	9.298	9.585
AGENCY_ZD	13.5052	0.080	169.620	0.000	13.349	13.661
AGENCY_ZE	1.2242	0.110	11.118	0.000	1.008	1.440
AGENCY_ZI	26.7393	0.078	344.946	0.000	26.587	26.891
AGENCY_ZL	22.5971	0.077	292.766	0.000	22.446	22.748
AGENCY_ZR	-7.8440	0.102	-77.237	0.000	-8.043	-7.645

...

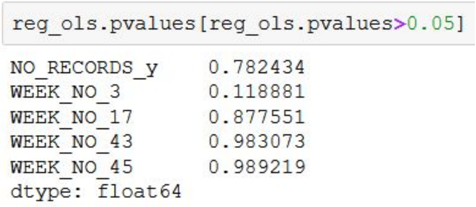


WEEK_NO_3	-0.2860	0.183	-1.559	0.119	-0.645	0.073
WEEK_NO_4	0.8955	0.187	4.790	0.000	0.529	1.262
WEEK_NO_5	3.3367	0.189	17.649	0.000	2.966	3.707
WEEK_NO_6	8.8097	0.187	47.130	0.000	8.443	9.176
WEEK_NO_7	12.0975	0.185	65.437	0.000	11.735	12.460
WEEK_NO_8	3.5078	0.184	19.062	0.000	3.147	3.869

...



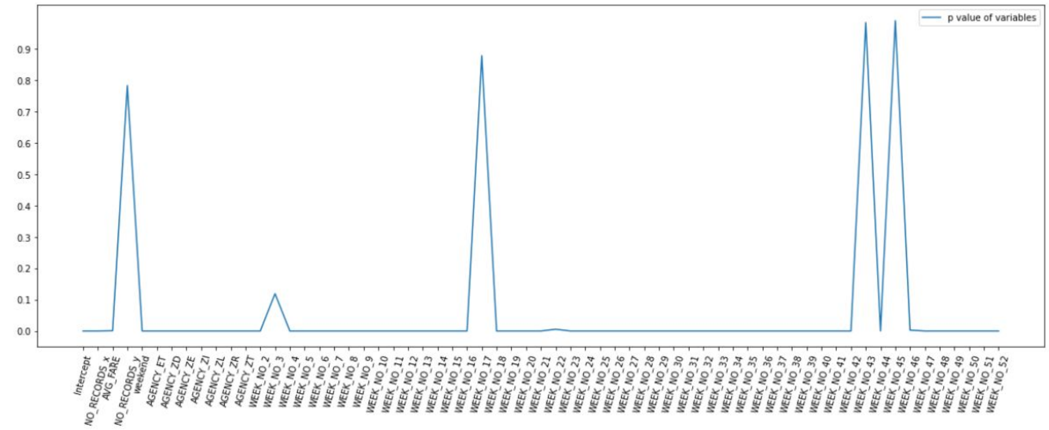
WEEK_NO_49	-6.1789	0.187	-33.121	0.000	-6.545	-5.813
WEEK_NO_50	-5.9522	0.188	-31.715	0.000	-6.320	-5.584
WEEK_NO_51	24.9136	0.188	132.616	0.000	24.545	25.282
WEEK_NO_52	60.1073	0.197	305.355	0.000	59.721	60.493

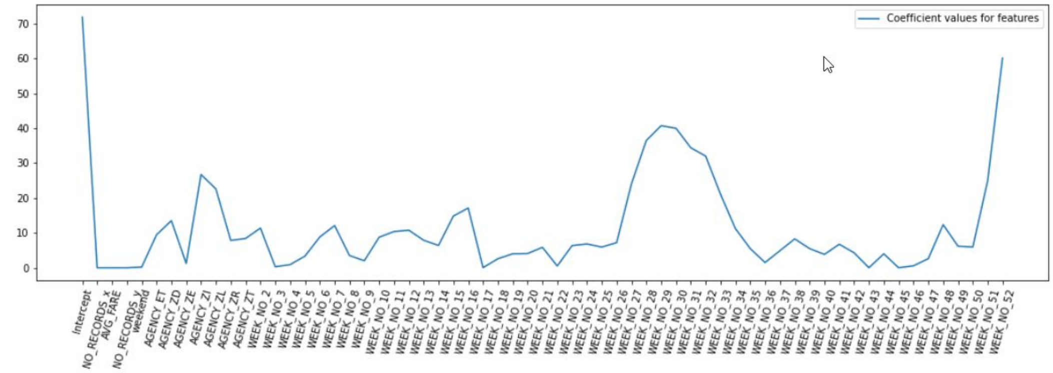


```
reg_ols.pvalues[reg_ols.pvalues>0.05]
```

```
NO_RECORDS_y    0.782434
WEEK_NO_3       0.118881
WEEK_NO_17      0.877551
WEEK_NO_43      0.983073
WEEK_NO_45      0.989219
dtype: float64
```

From the result we found out that almost all selected features are significant except a few weeks. We also found out that the number of cars rented at LAX (Number_of_records_y) turned out to be insignificant.





Due to the number of cars rented at LAX turned out to be insignificant, the Linear Regression failed to capture the relationship between car rental price and airfare price. Thus we need a better model.

Classification

Since Linear Regression failed to capture the relationship between airfare data and car rental data, we need to find better models. We have been exploring a few other models ever since, such as ARIMA, DeepAR and LSTM. But none of them fit perfectly well with our case. QL2 already tried ARIMA before and this model does not fit well with the case, DeepAR will only be effective for lots of time series data, while LSTM is an advanced model that needs great understanding of it to be efficient.

Based on the discussion with the Professor and also client feedback during the midterm presentation, we decided to move to the Classification model. Instead of predicting the actual price of car rental, now we want to classify whether the car rental is increasing, decreasing or no change related to the airfare price fluctuation.

Here is the steps we define to measure the correlation between airfare data and
car rental data :

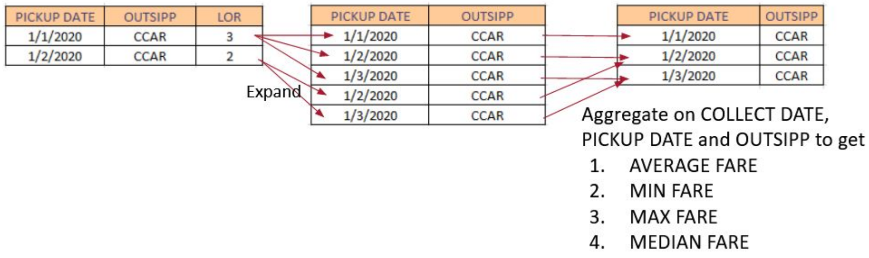
1. Create classification labels on car rental data, either its increase, decrease, or no change in price.
2. Create a baseline model using car rental data only.
3. Merge airfare data into the baseline model to create the merged model.
4. Evaluate the accuracy, precision and recall on both models. If the accuracy on the merged model is increased, then we can infer some correlation between those 2 data.

Classification on Car Rental Data

Feature Engineering

Split LOR

We find that the Length of Rental (LOR) on car rental data makes the average daily fare have different magnitude depending on how long is the LOR. In order to make them equal, we need to distribute car rental data on its LOR. So, a single car rental data with pickup date on Jan 1st, 2020 and LOR 3 days, will be expanded into 3 rows with different pickup dates running from Jan 1st to Jan 3rd.



PICKUP DATE	OUTSIPP	LOR
1/1/2020	CCAR	3
1/2/2020	CCAR	2

PICKUP DATE	OUTSIPP
1/1/2020	CCAR
1/2/2020	CCAR
1/3/2020	CCAR
1/2/2020	CCAR
1/3/2020	CCAR

PICKUP DATE	OUTSIPP
1/1/2020	CCAR
1/2/2020	CCAR
1/3/2020	CCAR

Aggregate on COLLECT DATE,
PICKUP DATE and OUTSIPP to get

1. AVERAGE FARE
2. MIN FARE
3. MAX FARE
4. MEDIAN FARE

And after we expanded all the car rental data based on its LOR, then we aggregated it on COLLECT DATE, PICKUP DATE, and OUTSIPP and count the AVERAGE_FARE, MINIMUM_FARE, MAXIMUM_FARE and MEDIAN_FARE. These resulting daily fares are having the same magnitude compared to each other.

Classification Label

We changed the daily average to 15 days moving average while keeping the same outsipp code. We only care about the price collected 30 days before the pick-up date to avoid the influence of booking time. After that, we calculated the change percentage based on the moving average.

We created three labels: increase, decrease, and no change, for the multiclass classification problem. When we defined the labels, we found the distribution of the change percentage is almost a normal distribution, so we decided to use the mean plus or minus some standard deviation. We tuned the number before standard deviation so that it could produce a balanced percentage for these three labels, and we ended up with 0.25. So if the change percentage is larger than mean + 0.25*sd, it is labeled as increase. If the change percent is smaller than mean - 0.25*sd, it is labeled as decrease. If the change percent is within the range, then it is labeled as no change. After labeling, 29% of data are increase, 31% are decrease, and 40% are no change.

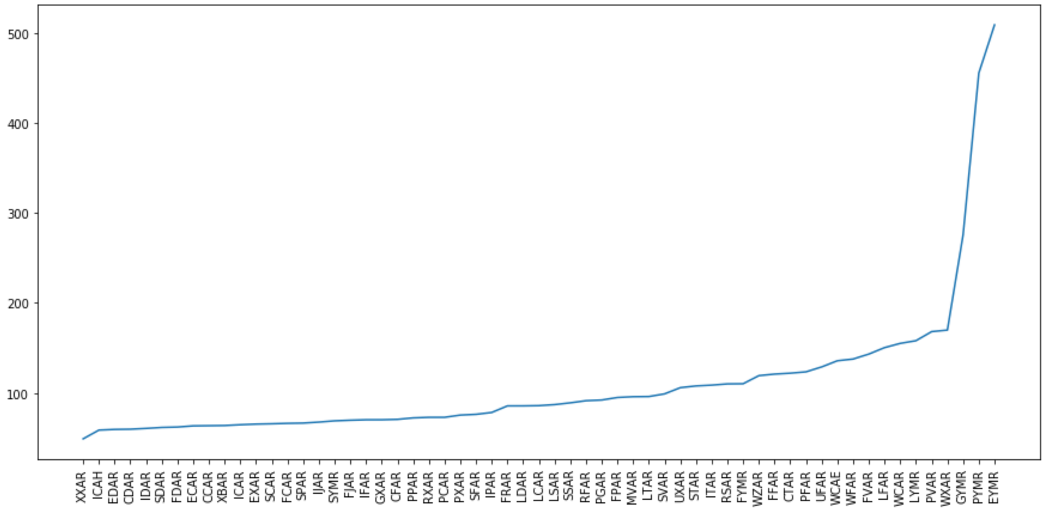
Timestamp Variable

For each pick-up date, we converted it to year, month, day, and whether it is a weekend or not. In addition to that, we added three more features: whether it is a public holiday, whether it is one day before or after a public holiday, which is used to include the influence of the long weekend and public holiday.

Baseline Classification

Reducing Dimension

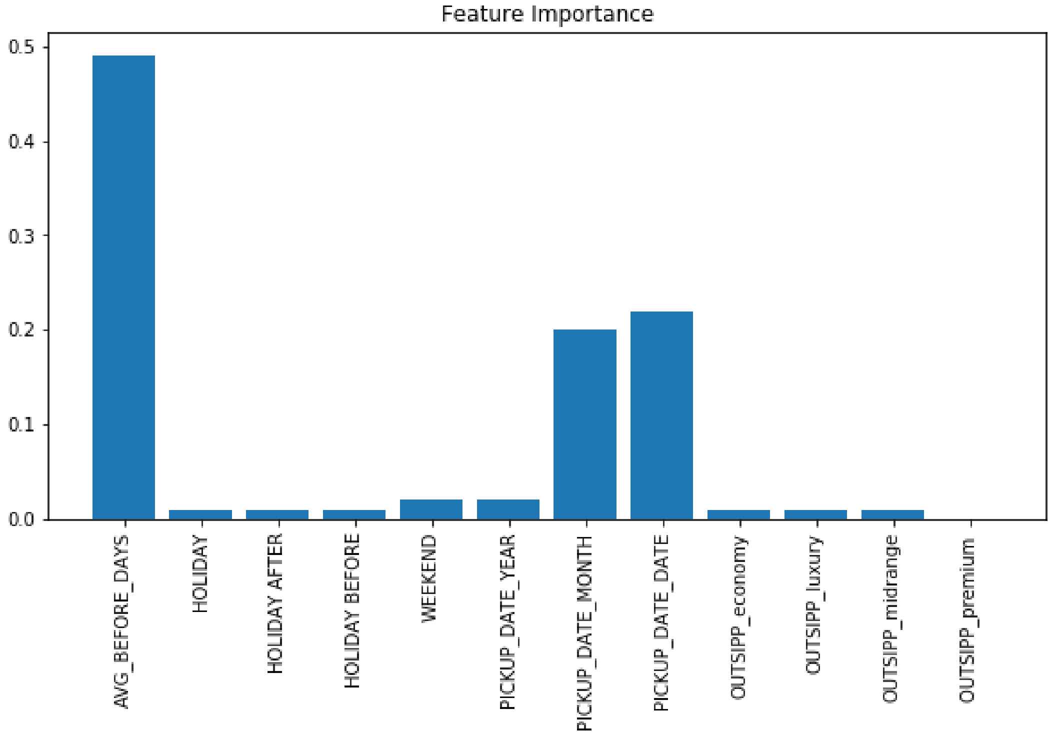
We changed car type to outsipp. Outsipp is a 4 character representation of the different car types with an industry standard. But there are 60 levels of outsipp in the data. So we manually divided them into 5 categories based on the average price and named them premium, luxury, midrange, economy, and basic. Thus, we greatly decreased the levels from 60 to 5, and also greatly decreased the dimension.



Outsipp with sorted average price

Default Random Forest

The baseline model is trained by applying default random forest. As the feature importance graph shows, the top three important features are moving average, pick-up day, pick-up month. These top features show that whether the change depends on the moving average in previous days and also some seasonal factors. We will include these features in the further improved model.



Result and Interpretation

We split the data into training data and testing data. In testing data, 24% of data are increase, 32% are decrease, and 44% are no change. Total accuracy on testing data is 41%, which is not high but still more effective than the random selection model which should only have 30% accuracy. The model yields more precise results for decrease label and no change label. In terms of recall, they are not quite balanced as the percentage of each label in testing data.

	Precision	Recall	F1-score
Decrease	0.53	0.29	0.38
Increase	0.28	0.58	0.38
No change	0.55	0.41	0.47

Classification on Merged Data

Airfare Data EDA

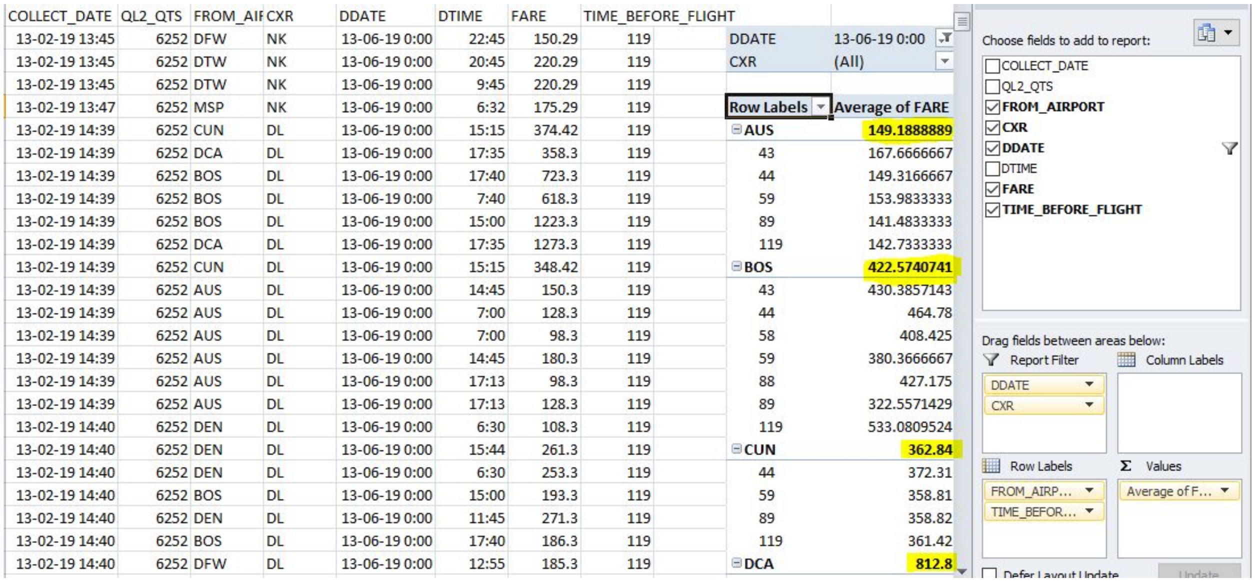
This is the second crucial piece of information in our analysis. To start with, we had 9,149,465 records with 13 columns. Basic division of the data was as below :

- Date Features : COLLECT_DATE, QL2_QTS, DDATE, DTIME
- Demographic Features : FROM_AIRPORT, TO_AIRPORT , SITE , CXR
- Flight Features : DSTP, CURRENCY, DFLIGHT, DROUTE , FARE

Based on our scope, we filtered airfare data for following :

- All the flights arriving to LAX airport
- In 2018 , 2019 , 2020 years
- Grouped by all the columns like date, carrier etc
- Averaged on price

To better understand airfare data, we did some basic EDA for one departure date -



COLLECT_DATE	QL2_QTS	FROM_AIF	CXR	DDATE	DTIME	FARE	TIME_BEFORE_FLIGHT
13-02-19 13:45	6252	DFW	NK	13-06-19 0:00	22:45	150.29	119
13-02-19 13:45	6252	DTW	NK	13-06-19 0:00	20:45	220.29	119
13-02-19 13:45	6252	DTW	NK	13-06-19 0:00	9:45	220.29	119
13-02-19 13:47	6252	MSP	NK	13-06-19 0:00	6:32	175.29	119
13-02-19 14:39	6252	CUN	DL	13-06-19 0:00	15:15	374.42	119
13-02-19 14:39	6252	DCA	DL	13-06-19 0:00	17:35	358.3	119
13-02-19 14:39	6252	BOS	DL	13-06-19 0:00	17:40	723.3	119
13-02-19 14:39	6252	BOS	DL	13-06-19 0:00	7:40	618.3	119
13-02-19 14:39	6252	BOS	DL	13-06-19 0:00	15:00	1223.3	119
13-02-19 14:39	6252	DCA	DL	13-06-19 0:00	17:35	1273.3	119
13-02-19 14:39	6252	CUN	DL	13-06-19 0:00	15:15	348.42	119
13-02-19 14:39	6252	AUS	DL	13-06-19 0:00	14:45	150.3	119
13-02-19 14:39	6252	AUS	DL	13-06-19 0:00	7:00	128.3	119
13-02-19 14:39	6252	AUS	DL	13-06-19 0:00	7:00	98.3	119
13-02-19 14:39	6252	AUS	DL	13-06-19 0:00	14:45	180.3	119
13-02-19 14:39	6252	AUS	DL	13-06-19 0:00	17:13	98.3	119
13-02-19 14:39	6252	AUS	DL	13-06-19 0:00	17:13	128.3	119
13-02-19 14:40	6252	DEN	DL	13-06-19 0:00	6:30	108.3	119
13-02-19 14:40	6252	DEN	DL	13-06-19 0:00	15:44	261.3	119
13-02-19 14:40	6252	DEN	DL	13-06-19 0:00	6:30	253.3	119
13-02-19 14:40	6252	BOS	DL	13-06-19 0:00	15:00	193.3	119
13-02-19 14:40	6252	DEN	DL	13-06-19 0:00	11:45	271.3	119
13-02-19 14:40	6252	BOS	DL	13-06-19 0:00	17:40	186.3	119
13-02-19 14:40	6252	DFW	DL	13-06-19 0:00	12:55	185.3	119

DDATE	13-06-19 0:00
CXR	(All)

Row Labels	Average of FARE
AUS	**149.1888889**
43	167.6666667
44	149.3166667
59	153.9833333
89	141.4833333
119	142.7333333
BOS	**422.5740741**
43	430.3857143
44	464.78
58	408.425
59	380.3666667
88	427.175
89	322.5571429
119	533.0809524
CUN	**362.84**
44	372.31
59	358.81
89	358.82
119	361.42
DCA	**812.8**

Choose fields to add to report:

☐ COLLECT_DATE
☐ QL2_QTS
☑ **FROM_AIRPORT**
☑ **CXR**
☑ **DDATE**
☐ DTIME
☑ **FARE**
☑ **TIME_BEFORE_FLIGHT**

Drag fields between areas below:

Report Filter: DDATE, CXR
Column Labels:
Row Labels: FROM_AIRP..., TIME_BEFOR...
Values: Average of F...

Understandings from EDA:

- Price largely depends on the FROM_AIRPORT [Farther the airports, higher the price]
- DTIME and Carrier give the same level of granularity. [Usually, there will only be one flight at a specific time by one carrier. I.e On 13 June, at 17:40, there is only one flight, which is by Delta. United may have flight on same day, but not exactly at 17:40]
- We have all prices only in USD [Hence CURRENCY adds no value]
- SITE and CARRIER are duplicate values
- We have only one stop flights [Hence DSTP, DROUTE adds no value]

Feature Engineering

Based on the above understandings and business logic, we identified 2 strategies to transform to airfare data so that we can feed it into our ML model. There are two types of transformations we used-
1) Data Aggregation 2) Data Augmentation

Data Aggregation

Primary aim behind data aggregation was to ensure that the data provided to the model is as generalized as possible. Below are the aggregations we did and their business reasoning -

Aggregation function	Dropped columns	Business Logic
Average price over single day	Collect_Date , Dtime	We are interested in average price trend over an entire day, not necessarily over specific time in a day
Aggregated price over all carriers	CXR , SITE	We would be looking at 'market' price trend
Normalized price over distance	FROM_AIRPORT	We would focus on a price per mile, this helps us remove distance bias

Data Augmentation

Goal behind adding new airfare features was to capture as much time sensitive information as possible in a single row of data. Since we are not using any time series model, it is crucial for us to make every row in training data independent of each other and have time-enriched features.

- **Time Before Flight**
    - Difference between Collect_Date and Departure_Date(DDate) of the flight
    - Will only be used to determine the airfare price trend for a particular DDate
- **Fluctuation tags**
    - Total percent price change over a time interval of last x days
    - Captures high demand, price surge or any seasonal impact on airfare
    - Used as a feature for modelling

    In order to be holistic, we have created 3 features to capture this fluctuation -

    1) Short Term Fluctuation - Captures airline surge pricing
    2) Mid-Term Fluctuation - Captures exogenic events like baseball game, concert,conference
    3) Long-Term Fluctuation - Captures environmental seasonality

    These fluctuations are calculated using the following rule, where x determines if it is long/short

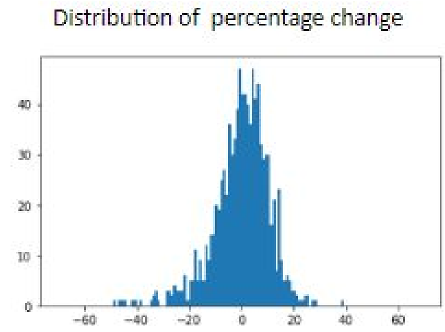
Total_fluc = sum (% change[1] in price per mile in last x days)
We have chosen the value of x using the scheme below.
Short term = total % airfare change in last 50 days
Mid term = total % airfare change in last 70 days
Long term = total % airfare change in last 120 days

Once we have extracted relevant information from the ‘Time Before Flight’(TBF) column , we can aggregate the data to 1-record-for-1-DDATE level. After this, we dropped the TBF column and averaged the price_per_mile. With this, we had 920 unique records, which means approximately 3 years worth of airfare data.

- **Rolling average**
  - This is similar to the price treatment for car rental. To get smoother trends, we have averaged over the last 10 days[2].
    average_price = mean of price_per_mile for last 10 days
  - Used as a feature in modelling

Note: The caveat in this formula is that the average is over the last 10 days of the data we have collected; which may not necessarily be the last 10 calendar days. Currently, for LAX, we haven’t observed any gaps in dates.

- **Percentage change**
  - This is similar to the price treatment for car rental. As seen from the distribution, % change is Normal around 0% change.
  - Used as a feature in modelling
  - % change heavily depends on the number of days we take average upon. For our implementation, we tried with 7 and 10. Although, no significant difference was observed in the performance of the model.



[1] % change = [(average price in last n days)-(today’s price)]/ (average price in last n days)
[2] All these hyper parameters are one click modifiable in the code.

Notes :

1. We got better results by using the actual decimal values for % change instead of using a categorical variable[Increase / Decrease / No Change] to indicate the airfare trend.
2. For fluctuation scores, 50-70-120 are chosen based on least correlation. They are found to be important features after modelling. Can try with different combinations in future.
3. For all the analysis, we have taken airfare data where Time Before Flight >=30 so that we are predicting prices 1 month in advance. But, considering the nature of data [IIDs[3]] we can skip this step and use all the information available for airfare prices because, instead of directly using available airfare features, we are creating new data points and inducing them into the model.
4. We could also use actual dollar value change instead of the % change values. Due to time constraints, we weren't able to implement and test this alternative.

Machine Learning Modelling

Merging the datasets

Merging the two datasets was done on the date on which a flight landed in LAX to the date on which a car was picked up in LAX. We did a left join on car rental data.

Performing sanity checks on all the columns/features

- Dropped duplicate columns for date
- Dropped price_per_mile column. This is the average price for airfare on that DDATE.
- Dropped avg_price column. This is the average car rental price on that DDATE.
- Checked for nulls. There were around <0.1% rows which were null, we decided to drop them.
- Divided the data into 70-30 training-test splits.
  Note: We have maintained order of the data. i.e first 70% rows are used for training and the next 30% are used for testing. This is to ensure that even if we use a time-series model ahead, we won't be required to split the data again. But, as mentioned earlier, any other scheme for having the train-test split would be okay, if we make the assumption of data being IID.

- Final list of columns on which data was trained-

Date	Pickup date, year, month, holiday flag , weekend flag
Car Rental data	Average price over last 15 days , OUTSIPP
Airfare data	Fluctuation Scores , % change in airfare
Label	Change

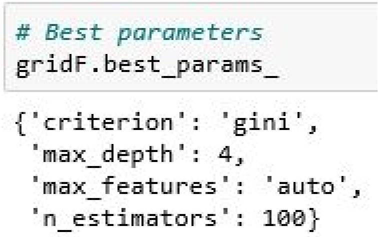
[3] Independent and Identically Distributed i.e all samples stem from the same generative process and that the generative process is assumed to have no memory of past generated samples.

Initial modelling

We decided to use RandomForest as our first model similar to our baseline.

Model tuning

We used GridSearch to optimize and hypertune the parameters. Parameters of the best model were as shown alongside.



```
# Best parameters
gridF.best_params_

{'criterion': 'gini',
 'max_depth': 4,
 'max_features': 'auto',
 'n_estimators': 100}
```

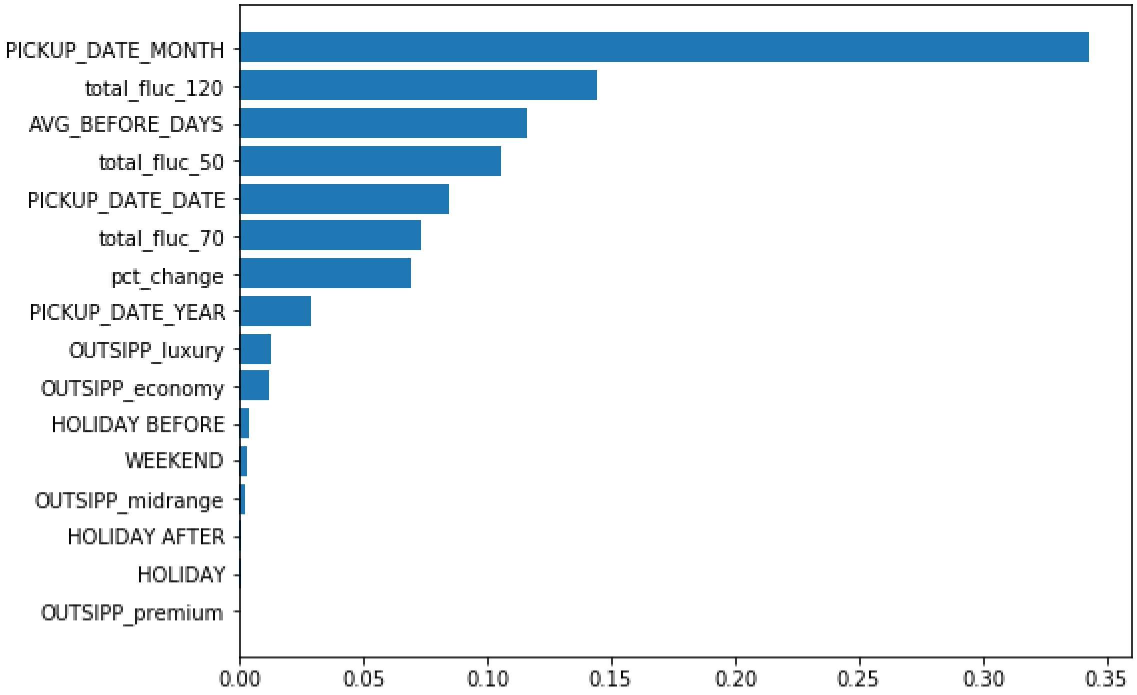
Evaluating the best model

Below are the results of the best random forest model -

Test Classification report

	Precision	Recall	F1-score
Decrease	0.53	0.16	0.25
Increase	0.47	0.29	0.36
No change	0.52	0.90	0.66

Feature Importance Graph



Based on the feature importance graph above, it is evident that the total_fluctuation_scores are important in predicting car rental prices; along with other date-time features.

IV. CONCLUSION

Business Impact and Result Evaluation

1. Model accuracy improves by adding airfare features.
   Adding ticket prices can help predict car rental prices more accurately.
2. Car rental price depends on average airfare fluctuations. (Short, Medium and Long)
   Airfare price data could be used to predict car rental prices, and the relevant results can be provided to companies in the car rental industry. Therefore, exploring the relationship between these two prices can unearth the company's potential customers.

Other Models Explored

Logistic Regression
Inference: Can not handle a large number of multi-type features or variables well; It can only handle two classification problems (softmax derived on this basis can be used for multiple classification), and must be linearly separable. Logistic regression is not suitable for our data.

KNN
Inference: When the sample is unbalanced, the prediction deviation is relatively large. For example: there are fewer samples in one category, and more samples in other categories; KNN is not suitable for our dataset.

SVM
Accuracy: 0.4263
Inference: When there are many observation samples, the efficiency is not very high; The volume of data is large, thus SVM is not a good choice.

Neural Network
Accuracy: 0.3177
Inference: The process is a black box, and the learning process between them cannot be observed. The output is difficult to interpret, which will affect the credibility and acceptability of the results; Neural Network could be used here, but parameter processing might be a challenge.

Gradient Boosting Classification
Training Accuracy: 0.4129
Test Accuracy: 0.4768

Training Classification report

	precision	recall	f1-score
Decrease	0.42	0.18	0.25
Increase	0.31	0.16	0.21
No change	0.44	0.80	0.56
accuracy			0.41

Test Classification report

	precision	recall	f1-score
Decrease	0.76	0.10	0.18
Increase	0.29	0.05	0.09
No change	0.47	0.98	0.64
accuracy			0.48

Inference: The performance of Gradient Boosting Classification is not better than random forest model.

ADA Boost

Training Accuracy: 0.5471
Test Accuracy: 0.4775

Training Classification report

	precision	recall	f1-score
Decrease	0.63	0.38	0.47
Increase	0.71	0.40	0.51
No change	0.48	0.80	0.60
accuracy			0.55

Test Classification report

	precision	recall	f1-score
Decrease	0.50	0.18	0.26

Increase	0.26	0.09	0.13
No change	0.50	0.90	0.64
accuracy			0.48

Inference: Training accuracy is larger than testing accuracy. Overfitting might be a problem here.

Future Work

1. **Aggregate by Regions/Cities**
   At present, our project mainly focuses on the Los Angeles market. In the future, markets in other parts of the United States can be explored. Cluster analysis can also be performed on each airport, for example, east coast, west coast and central area.
2. **Different time before rental**
   The time before the rental variable of the current model is set to 30 days, and the results obtained can only help predict the trend of car rental price changes after 30 days.
   For example, explore the record of renting a car 15 days in advance, 45 days in advance, and 90 days in advance. In this way, the accuracy of the model can be improved, and car rental prices at different points in the future can also be predicted.
3. **Enhanced Feature Engineering**
   The features selected by the current model are only part of the data features, and many features are not included in the model.
   Optimizing existing features is also very important. For instance, more optimized coding processing for categorical variables.
   Not only the binary variables of holidays and non-holidays are considered, but the normalized coding centering on holidays is the direction that can be explored.
4. **Explore Other Time Series Model**
   In the early stage of the project, the models studied, such as ARIMA, are not applicable to our data. In the future, other time series models such as vector autoregressive models can be explored.