Flight Price Prediction or FPP

Yazan Kbaili, Hamada Salhab, Vladislav Lopatovskii

08.05.2024

Introduction

Predicting the base fare of flight tickets is crucial for airlines to maximize their revenue. By developing an accurate machine learning model, airlines can gain valuable insights into optimal pricing strategies. The base fare, which makes up a significant portion of the total ticket price, is the key focus of this project. Through careful analysis and modeling, the aim is to create a tool that will enable airlines to effectively set prices and adapt to market conditions, ultimately leading to increased profitability.

Data Description

The dataset comprises purchased flight tickets extracted from Expedia between the dates of Aprial 16, 2022, and October 5, 2022. These tickets facilitate travel to and from a selection of major airports, identified by their three-character IATA airport codes:

- ATL (Hartsfield-Jackson Atlanta International Airport)
- DFW (Dallas/Fort Worth International Airport)
- DEN (Denver International Airport)
- ORD (O'Hare International Airport)
- LAX (Los Angeles International Airport)
- CLT (Charlotte Douglas International Airport)
- MIA (Miami International Airport)
- JFK (John F. Kennedy International Airport)
- EWR (Newark Liberty International Airport)
- SFO (San Francisco International Airport)
- DTW (Detroit Metropolitan Wayne County Airport)
- BOS (Boston Logan International Airport)
- PHL (Philadelphia International Airport)
- LGA (LaGuardia Airport)
- IAD (Washington Dulles International Airport)
- OAK (Oakland International Airport)

Column definitions:

1. legId: Unique identifier for the flight.
2. searchDate: Date (YYYY-MM-DD) when the entry was retrieved from Expedia.
3. flightDate: Date (YYYY-MM-DD) of the scheduled flight.

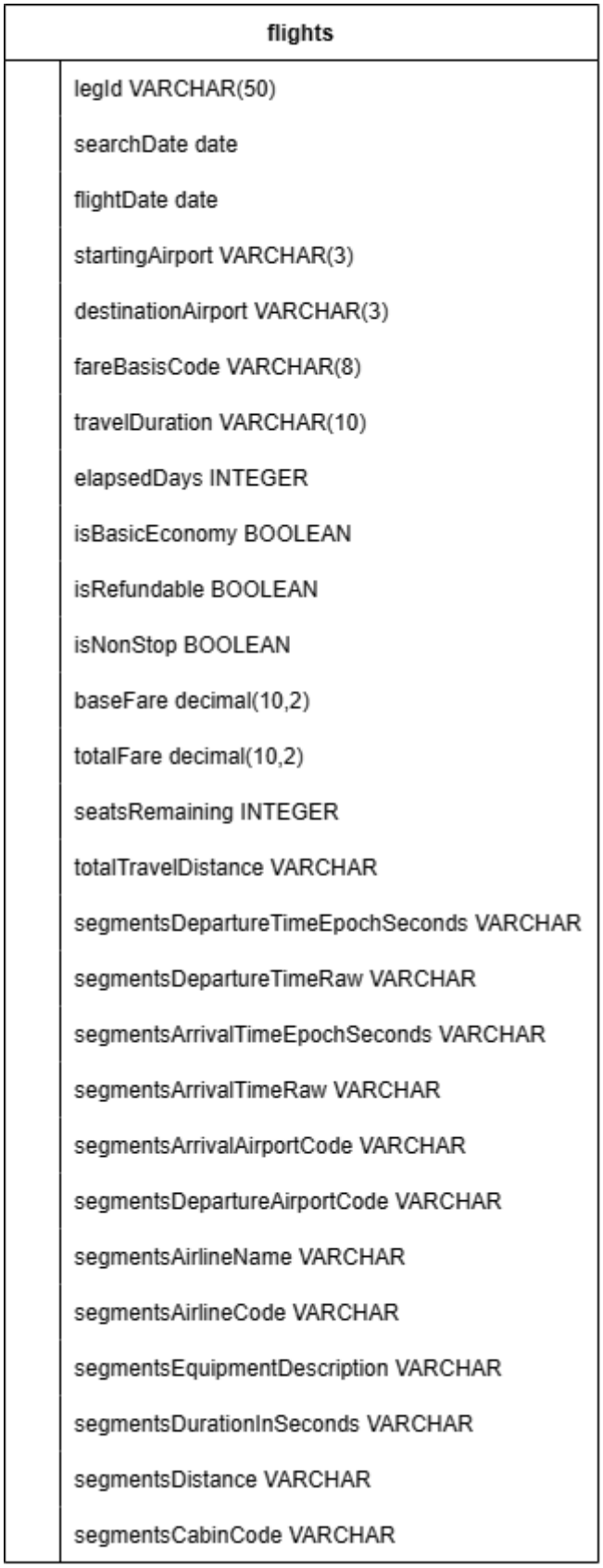
4. startingAirport: Three-character IATA code for the departure airport.
5. destinationAirport: Three-character IATA code for the arrival airport.
6. fareBasisCode: Fare basis code associated with the ticket.
7. travelDuration: Duration of travel in hours and minutes.
8. elapsedDays: Number of elapsed days (usually 0).
9. isBasicEconomy: Boolean indicating whether the ticket is for basic economy class.
10. isRefundable: Boolean indicating whether the ticket is refundable.
11. isNonStop: Boolean indicating whether the flight is non-stop.
12. baseFare: Price of the ticket in USD.
13. totalFare: Total price of the ticket including taxes and fees in USD.
14. seatsRemaining: Number of available seats remaining.
15. totalTravelDistance: Total travel distance in miles. (Some entries may be missing this data)
16. segmentsDepartureTimeEpochSeconds: Departure time (Unix time) for each leg of the trip. Entries are separated by '||'.
17. segmentsDepartureTimeRaw: Departure time (ISO 8601 format) for each leg of the trip. Entries are separated by '||'.
18. segmentsArrivalTimeEpochSeconds: Arrival time (Unix time) for each leg of the trip. Entries are separated by '||'.
19. segmentsArrivalTimeRaw: Arrival time (ISO 8601 format) for each leg of the trip. Entries are separated by '||'.
20. segmentsArrivalAirportCode: IATA code for the arrival airport for each leg of the trip. Entries are separated by '||'.
21. segmentsDepartureAirportCode: IATA code for the departure airport for each leg of the trip. Entries are separated by '||'.
22. segmentsAirlineName: Name of the airline servicing each leg of the trip. Entries are separated by '||'.
23. segmentsAirlineCode: Two-letter airline code for each leg of the trip. Entries are separated by '||'.
24. segmentsEquipmentDescription: Type of airplane used for each leg of the trip. Entries are separated by '||'.
25. segmentsDurationInSeconds: Duration of flight in seconds for each leg of the trip. Entries are separated by '||'.
26. segmentsDistance: Distance traveled in miles for each leg of the trip. Entries are separated by '||'.
27. segmentsCabinCode: Cabin class for each leg of the trip. Entries are separated by '||'.

Architecture of Data Pipeline

Stage I
Input: Raw dataset
Output: Postgresql Database
Stage II
Input: Postgresql Database
Output: Optimized Hive Table, EDA(5 charts)
Stage III
Input: Optimized Hive Table
Output: Training Testing Dataset split, Predictions, Best Params and Evaluation of GBTRegressor model, Predictions, Best Params and Evaluation of RandomForestRegressor model, Best GBTRegressor, Best RandomForestRegressor
Stage IV
Input: Stage I Stage II & Stage III Hive tables (Raw dataset for Stage I, Data Analysis results for Stage II, and Predictive Data Analysis for Stage III)
Output:Apache Superset Dashboard with Data Description, Data Insights and Prediction Results

Data Preparation

ER Diagram:


flights
legId VARCHAR(50)
searchDate date
flightDate date
startingAirport VARCHAR(3)
destinationAirport VARCHAR(3)
fareBasisCode VARCHAR(8)
travelDuration VARCHAR(10)
elapsedDays INTEGER
isBasicEconomy BOOLEAN
isRefundable BOOLEAN
isNonStop BOOLEAN
baseFare decimal(10,2)
totalFare decimal(10,2)
seatsRemaining INTEGER
totalTravelDistance VARCHAR
segmentsDepartureTimeEpochSeconds VARCHAR
segmentsDepartureTimeRaw VARCHAR
segmentsArrivalTimeEpochSeconds VARCHAR
segmentsArrivalTimeRaw VARCHAR
segmentsArrivalAirportCode VARCHAR
segmentsDepartureAirportCode VARCHAR
segmentsAirlineName VARCHAR
segmentsAirlineCode VARCHAR
segmentsEquipmentDescription VARCHAR
segmentsDurationInSeconds VARCHAR
segmentsDistance VARCHAR
segmentsCabinCode VARCHAR

Sample from the data:

legid	**65a8a691c4d77487a439d6f44c219a39**
searchdate	2022-04-17
flightdate	2022-04-21
startingairport	LGA
destinationairport	DFW
farebasiscode	UA3NA0BQ
travelduration	PT5H45M
elapseddays	0
isbasiceconomy	True
isrefundable	False
isnonstop	False
basefare	110.70
totalfare	142.60
seatsremaining	9
totaltraveldistance	1487.0
segmentsdepartureti meepochseconds	1650542400||1650554580
segmentsdepartureti meraw	2022-04-21T08:00:00.000-04:00||2022-04-21T11:23:00.000-04:00
segmentsarrivalti meepochseconds	1650551340||1650563100
segmentsarrivalti meraw	2022-04-21T10:29:00.000-04:00||2022-04-21T12:45:00.000-05:00
segmentsarrivalairportcode	ATL||DFW
segmentsdepartureairportcode	LGA||ATL
segmentsairlinename	Delta||Delta
segmentsairlinecode	DL||DL
segmentsequipmentdescription	Airbus A321||Airbus A321
segmentsdurationinseconds	8940||8520

legid	65a8a691c4d77487a439d6f44c219a39
segmentsdistance	762||725
segmentscabincode	coach||coach

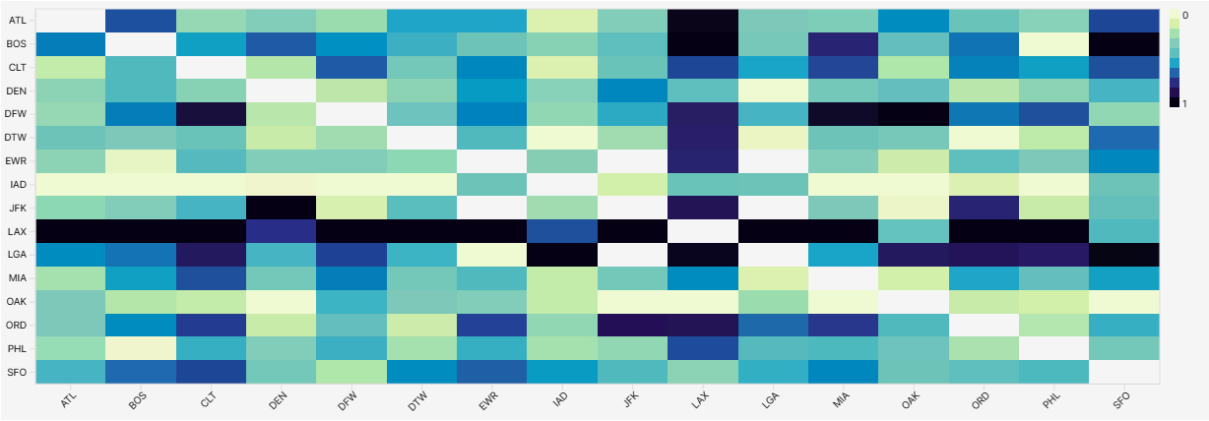
Creating Hive Tables and preparing the data for analysis:

1. We sampled the original dataset and took only 5% of it, so that the cluster will be able to handle it.
2. Optimized Hive table was created: partiotioned by *startingairport* and clustered by *legid*.

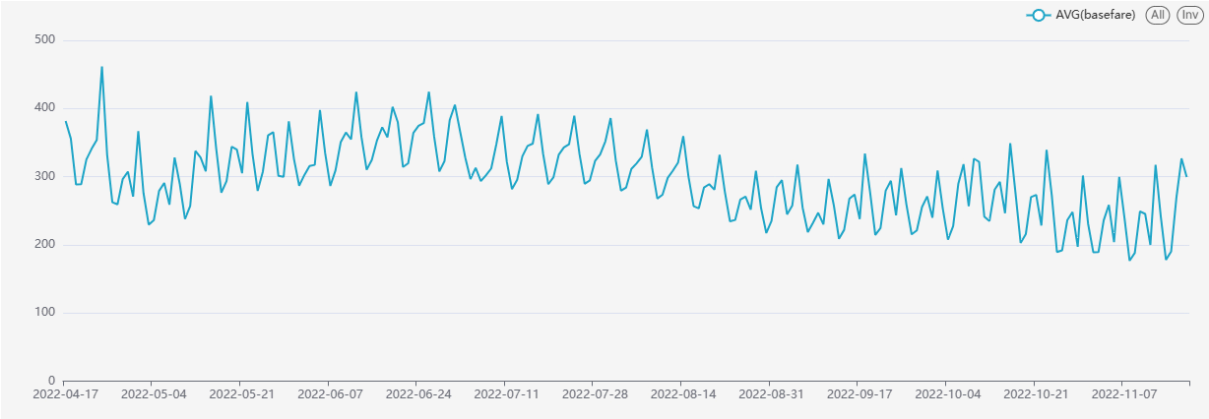
Data Analysis

Charts

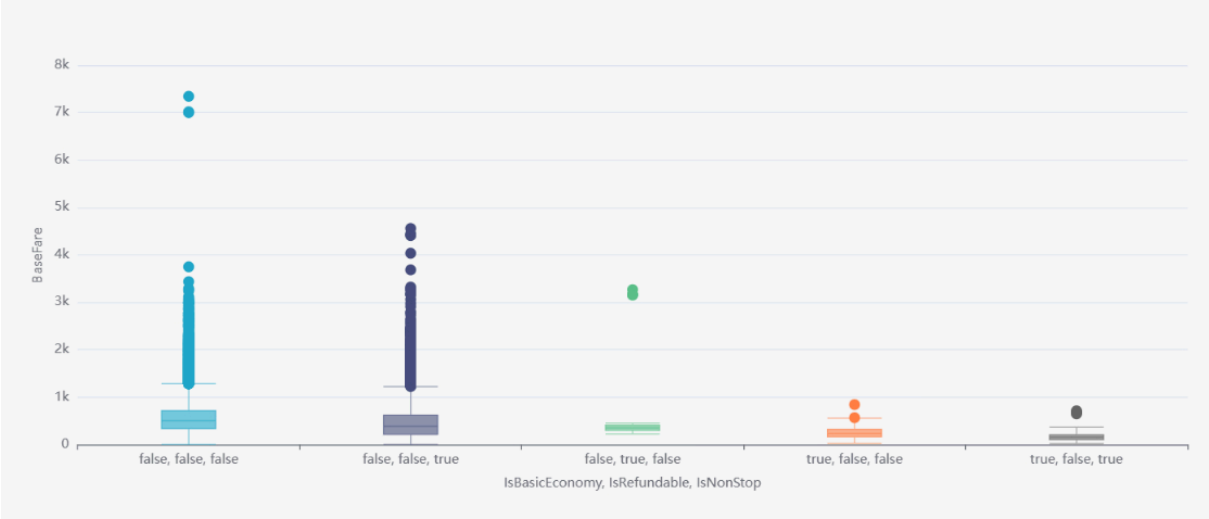
1. HeatMap of Airports



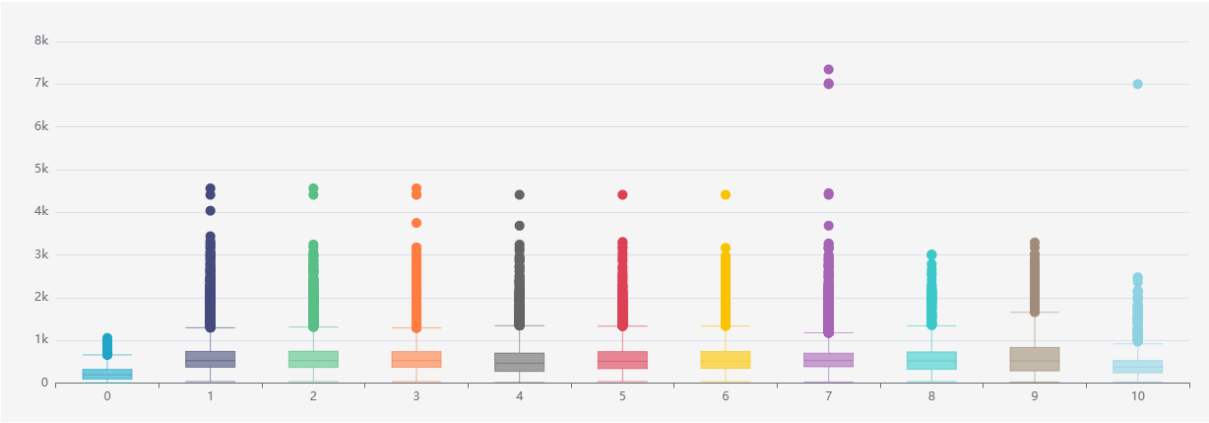
2. Historical average price



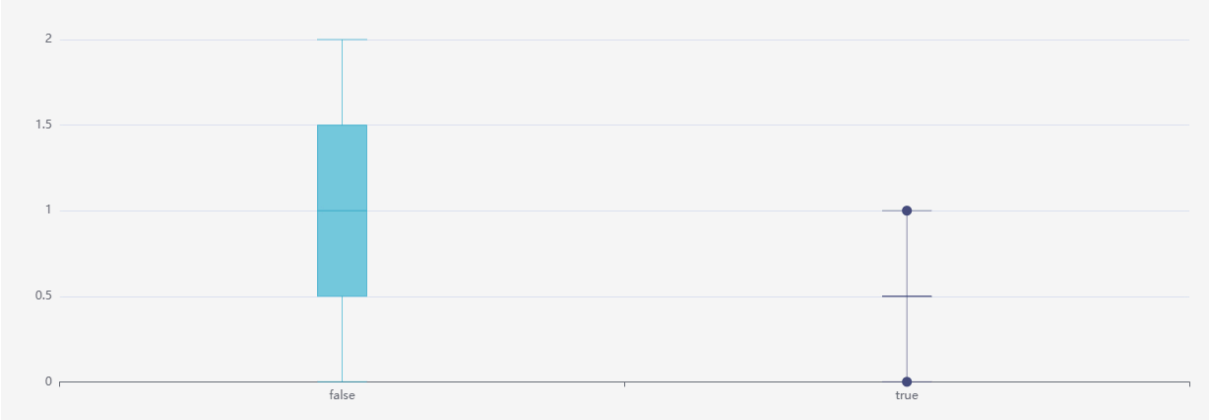
3. Boolean Features



4. Remaining seats



5. IsNonStop vs ElapsedDays



Interpretations

1. This chart shows the popularity of the airports as destination or starting. The darker color means more popularity.
2. This chart shows how changes the average price of tickets during the 2022 year.
3. This chart shows the distributions of prices based on the different combination of following characterstics: IsBasicEconomy, IsRefundable and IsNonStop.
4. This chart shows the distributions of prices based on the number of remaining seats(after the ticket is bought).

5. This chart shows the distributions of number of days based on whether the flight is non stop or not.

Conclusions

1. The most popular destionations are LA, NY, Boston, Dallas, Fort Worth
2. Weekend flights more in depand, therefore, they are more expensive
3. Some combinations are not present in the dataset. Also, unexpectedly, non-stop flights are cheaper.
4. The last seat ticket is cheaper, but also less seats left, higher the prices because it implies that the flights is in demand
5. As expected, there is a relation between whether the flight is non-stop and the number of days. If the flight is non-stop, it takes less time.

ML Modeling

Feature extraction and data preprocessing

Column Elimination:
Superfluous columns were removed to streamline the dataset for subsequent analyses. The eliminated columns included identifiers, time stamps, and other non-essential attributes such as legId, searchDate, elapsedDays, totalFare, segmentsArrivalAirportCode, segmentsDepartureAirportCode, segmentsEquipmentDescription, fareBasisCode, segmentsDepartureTimeRaw, segmentsArrivalTimeRaw, segmentsAirlineName, segmentsDepartureTimeEpochSeconds, segmentsArrivalTimeEpochSeconds, and segmentsDistance.
Feature Engineering:
The totalTravelDistance column was targeted for feature engineering, where missing values were imputed using the mean of existing data.
Time Duration Parsing:
The travelDuration attribute was parsed to convert formatted strings into a unified measure of total seconds, thereby standardizing the data for more effective processing and analysis.
Categorical Variable Encoding:
Categorical variables such as startingAirport, destinationAirport, segmentsCabinCode, and segmentsAirlineCode were transformed using one-hot encoding.
Cyclic Feature Encoding:
Temporal data from the flightDate column was encoded into cyclic features using sine and cosine transformations. This method captures the periodic nature of the data, facilitating the model's ability to recognize and utilize seasonal trends and patterns.
Imputation of Missing Values:

Missing values in categorical data were addressed using an imputation method that replaces nulls with the most frequent value observed in the column.

Numerical Feature Scaling:

A StandardScaler was employed to normalize the numerical features, which equalizes the influence of each feature on the model, enhancing the predictive performance and stability.

Data Segmentation:

The dataset was partitioned into training and testing subsets, with 80% of the data allocated for training and 20% reserved for testing.

Training and fine-tuning

1. GBTRegressor (Gradient-Boosted Tree Regressor)

How It Works:

Gradient-Boosted Trees (GBTs) are an ensemble learning method where new models are created that predict the residuals or errors of prior models and then added together to make the final prediction. It's a form of boosting because it combines multiple weak models (usually decision trees) into a stronger model, typically in a sequential process. Each tree in the sequence focuses on correcting the mistakes of the tree before it. The lossType parameter dictates how the prediction errors are calculated and minimized, while maxDepth controls the maximum depth of each tree, limiting the complexity to prevent overfitting.

2. RandomForestRegressor

How It Works:

Random Forest is another ensemble learning technique, but unlike gradient boosting, it relies on bagging (bootstrap aggregating). It builds multiple decision trees and merges them together to get a more accurate and stable prediction. Each tree in the forest is built from a sample drawn with replacement (i.e., a bootstrap sample) from the training set. Furthermore, when splitting each node during the construction of a tree, the best split is found either from all input features or a random subset of them. The subsamplingRate controls the size of the dataset used to train each tree, and numTrees specifies the number of trees in the forest.

Evaluation

1. RMSE: 119.9516
   R^2: 0.5718
   The lower RMSE and higher R^2 indicate that the GBTRegressor has done a reasonably good job of capturing the variance in the data, with about 57.18% of variability in the dependent variable being explained by the model.
2. RMSE: 129.0482
   R^2: 0.5007
   The RandomForestRegressor's performance is somewhat weaker than the

GBTRegressor's, with a higher RMSE and lower R^2, which implies that it captures around 50.07% of the variability in the target variable.

Conclusion

The project aimed to develop a machine learning model to predict the base fare of flight tickets, crucial for airlines to optimize revenue. Utilizing flight ticket data from Expedia, the project focused on base fare analysis, data preparation, and ML modeling. The data pipeline involved stages of data processing, exploration, and visualization, leading to insights such as airport popularity and pricing dynamics based on factors like flight duration and refundability. ML modeling encompassed feature extraction, preprocessing, and segmentation, emphasizing column elimination, feature engineering, time duration parsing, categorical variable encoding, cyclic feature encoding, imputation of missing values, and numerical feature scaling. The model achieved the goal of accurately predicting base fares, empowering airlines with actionable pricing strategies to enhance profitability amidst fluctuating market conditions.

Reflections on Own Work

Challenges and difficulties

In the second stage, we encountered problems in parsing some features and had to move to simpler queries.
In the third stage, we faced several challenges, mainly related to the capacity of our cluster and the huge size of the dataset. One significant challenge was the limited computing power of our cluster, which could not efficiently process the huge amount of data. This limitation not only slowed down the data processing tasks, but also prevented us from performing comprehensive hyperparameter optimization, which is very important for fine-tuning our machine learning models.

The table of contributions

Project Tasks	Yazan Kbaily	Hamada Salhab	Vladislav Lopatovskii	Deliverables	Actu Hour Spen
Sample 5% of the data	1				1
Downloading the dataset to the cluster		1			1
Creating tables, sql script			1		0.3
"import_data.sql"			1		0.3
"test_database.sql"			1		0.3
Importing the dataset to the cluster, Python script "build_projectdb.py"		0.5	0.5		1.25
Sqoop	0	0	1		0.5
put github account on server and push stage1		1			0.33
Create a Hive database			1		0.5
Hive optimizations & bucketing		1			1
EDA (extract 5 different insights) using hive queries	0.33	0.33	0.33		3
The whole Stage III	0.5	0.25	0.25		16
Create a dashboard	0	1	0		3
Composing the report	0.15	0.15	0.7	pdf file	3