# GetMyFlight

Cheapest Flight Rates in No Time

**Group No - 3**
Aastha Grover
Ankur Bag
Neelesh Saxena
Tushar K

# Problem Statement

- Unpredictability and uncertainty of flight fares.
- ‘The earlier you book, the cheaper you get’ - is always not true.
- Flight Reservation websites are in a rush to provide their customers with the cheapest flights.
- Nobody is bothered how the rates will vary in the future. This is where our research is centered.

***Our Proposal/Suggestion:***

- Build a flight recommendation system -
  - To predict the flight rates for a particular destination during the proposed dates,
  - To predict the date on which flight rates may fall or increase

# Actor/Use Cases

***Actor***

User is the sole actor of the system.

***Use Case***

- ➔ User will register and Login into the application.
- ➔ User will input Preferred dates and Source-Destination.
- ➔ System will predict the cheapest flights flight recommender.

# Milestones

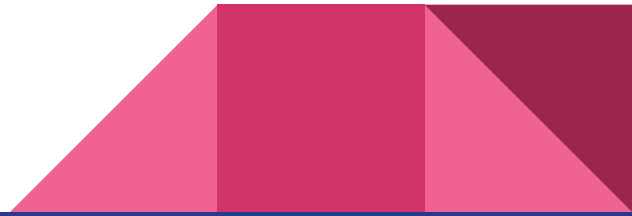
| Key Milestones | Start Date | End Date |
|---|---|---|
| A. Discusión on Relevant Factors ( Feature Selection).<br>B. Dataset Created with Relevant Factors.<br>C. Decide the architecture of the application. | 11/3/2016 | 11/10/2016 |
| A. Data Analysis - Data Visualization, Data Cleansing / Manipulation<br>B. Decide on the algorithm suitable for our application. | 11/10/2016 | 11/17/2016 |
| A. Build the Predictive algorithm/Model. | 11/17/2016 | 11/24/2016 |
| A. Application Integration , Backend Complete | 11/24/2016 | 12/1/2016 |
| A. Application Test Run Executed, FrontEnd Completed | 12/1/2016 | 12/8/2016 |

# Probable Factors

- Source - Destination
- Distance from source to destination.
- Number of unsold seats (and recent fluctuations on that number), as a measure of demand.
- State of competing options on the same route (number of available seats on similar flights, current price for those flights, recent price fluctuations) .
- Date/time of booking, especially days left until departure.
- Recent price for the same ticket, and recent fluctuations of the price
- Oil Rates

# Scope

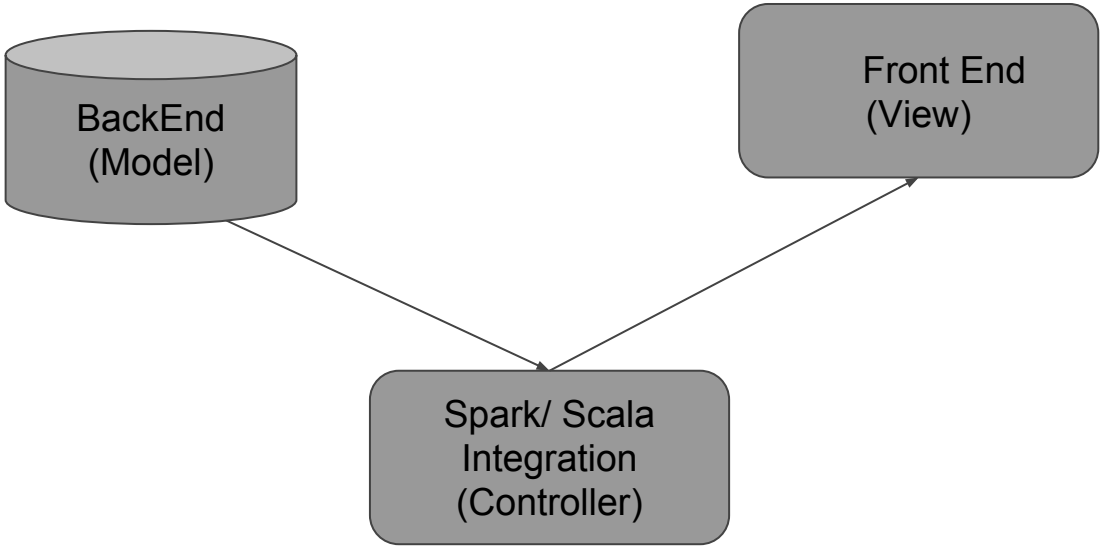
- Number of Rows - 100,000
- Round Trip/One-Way



# Acceptance Criteria

- Number Of Carrier (4)
- Number Of Cities to be Analysed(6)
- 3 successful predictions out of 5.

# Architecture



# Code Repository

https://github.com/ankurbag/CSYE7200_Scala_Project_Group3

# Programming In Scala

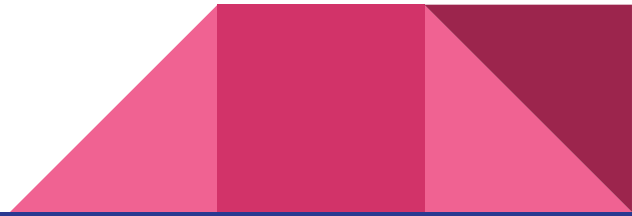
Apache Spark - data parsing, data cleaning, data manipulation

Zeppellin - scala notebook for data ingestion and collaboration

Play - create an application based on scala

Spark mlib - use machine learning algorithms supported by scala

*These is according to our current planning model. As we go ahead from here, this section might increase.



*Thank You :)*