A Case Study Report on

Flight Fare Prediction Application

Submitted in partial fulfilment for the requirements of

B. E. (CSE) VI Semester Case Study

in

COMPUTER SCIENCE AND ENGINEERING

by

Manideep Kalyanam (160118733102)

Shanmukha Manoj Bharadwaj Korrapati (160118733115)

Department of Computer Science and Engineering

Chaitanya Bharathi Institute of Technology (Autonomous),

(Affiliated to Osmania University, Hyderabad)

Hyderabad, Telangana (India) – 500 075

CHAITANYA BHARATHI
INSTITUTE OF TECHNOLOGY (A)
Affliated to Osmania University

CERTIFICATE

This is to certify that the project entitled "**Flight Fare Prediction Application"**, submitted to the Computer Science and Engineering Department, Chaitanya Bharathi Institute of Technology, in partial fulfilment of the requirement for the course Case Study, is a bonafide record of work done by **Manideep Kalyanam (160118733102)** and **Shanmukha Manoj Bharadwaj Korrapati (160118733115)**, from February, 2021 to May 2021 under our guidance and supervision.

Mentors,

Mr. B Sateesh
Assistant Professor,
Department of CSE,
CBIT

Ms. G Kavita
Assistant Professor,
Department of CSE,
CBIT

Supervisor,

Dr. Kolla Morarjee
Associate Professor,
Department of CSE,
CBIT

ACKNOWLEDGEMENT

Firstly, we are grateful to Chaitanya Bharathi Institute of Technology for giving us this opportunity to do a Case Study. We acknowledge our sincere gratitude to our project guide Dr. Kolla Morarjee (Associate Professor, Dept. of CSE, CBIT) for providing us his valuable assistance for doing this project. We are grateful to our mentors Mr. B Sateesh (Assistant Professor, Dept. of CSE, CBIT) and Ms. G Kavita (Assistant Professor, Dept. of CSE, CBIT) for their valuable guidance and support even in their very busy schedule. Finally, we thank each of our teammates for working hard for all the days and completing this project successfully.

Table of Contents

ABSTRACT

Someone who purchase flight tickets frequently would be able to predict the right time to procure a ticket to obtain the best deal. Many airlines change airfare for their revenue management. The airline may increase the costs when the demand is to be expected to extend the capacity.

To estimate the minimum flight fare, data of different air routes in India is available in Kaggle including the features like departure time, arrival time and airways over a specific period. Features are extracted from the collected data to apply Machine Learning regression algorithms to predict the prices at a given time.

In this case study, we build an android application where the user can interact with the ML model and get the predicted airfare by giving the details like departure time, arrival time, airline, source and destination.

Chapter 1. Introduction

The flight ticket buying system is to purchase a ticket many days prior to flight take-off so as to stay away from the effect of the most extreme charge. Mostly, aviation routes don't agree this procedure. Plane organizations may diminish the value at the time, they have to create the market and at the time when the tickets are less accessible. They may maximize the costs. So, the cost may rely upon different factors. To foresee the costs this venture uses AI to exhibit the ways of flight tickets after some time. All organizations have the privilege and opportunity to change its ticket costs at any time. Explorer can set aside cash by booking a ticket at the least price. People who had travelled by flight frequently are aware of flight fare fluctuations.

The airlines use complex policies of Revenue Management for execution of distinctive evaluating systems. The evaluating system as a result changes the charge counting on time, season, and festive days to vary the header or footer on successive pages. The ultimate aim of the airways is to earn profit whereas the customer searches for the minimum costs. Customers usually try to buy the ticket well in advance of departure date so as to avoid hike in airfare as date comes closer. But actually, this is not the fact. The customer may finish up by giving quite they need to for an equivalent seat.

Thanks to mobile technology, we've everything within the palm of our hand lately. Smartphones together with app are providing people with extensive information in but a second. Today we've apps which will help us shop, order food, order a cab and far more during a blink of an eye fixed. These apps are not just available but are being used by millions all over the globe. In this context, a flight fare predictor application seems to be the most logical and convenient development.

Flight Fare Predictor is an android application which focuses on the ticket price of the user's journey. It provides the user with available airlines and their prices respectively after giving the journey particulars like date of journey, departure time and arrival time to the system.

Chapter 2. Background Information

Anyone who has booked a flight ticket knows how unexpectedly the prices vary. Airlines use sophisticated quasi-academic tactics known as "revenue management" or "yield management". The cheapest available ticket for a given date gets more or less expensive over time. This usually happens as an attempt to maximize revenue based on -

1. Time of purchase patterns (making sure last-minute purchases are expensive)

2. Keeping the flight as full as they want it (raising prices on a flight which is filling up in order to reduce sales and hold back inventory for those expensive last-minute expensive purchases)

So, if we could inform the travellers with the optimal time to buy their flight tickets based on the historic data and also show them various trends in the airline industry, we could help them save money on their travels. This would be a practical implementation of a data analysis, statistics and machine learning techniques to solve a daily problem faced by travellers.

The objectives of the project can broadly be laid down by the following questions -

1. Flight Trends: Do airfares change frequently? Do they move in small increments or in large jumps? Do they tend to go up or down over time?

2. Best Time to Buy: What is the best time to buy so that the consumer can save the most by taking the least risk? So should a passenger wait to buy his ticket, or should he buy as early as possible?

3. Verifying Myths: Does price increase as we get near to departure date? Is Indigo cheaper than Jet Airways? Are morning flights expensive?

Chapter 3. Scope of the Case Study

Methodology

The study has been carried out and the solutions has been implemented in a span of 8 weeks. The following steps are performed to complete this Case Study:

Data collection and dataset preparation: This involved collection of data related to flight journeys from Kaggle, then preprocessing is applied on dataset which will remove all the unnecessary data and extract important features from data.

Developing a predictive model for Flight Fare Prediction: In this step, regression model and Neural network are to developed, analyzed them and chose the best model which will run effectively on available data.

Building an android application: Using this machine learning model, we developed a mobile app which interacts with the user. This app asks the user for their journey details and predicts the flight fare. It also provides the fares of various airlines on the given journey.

Limitations

However, our study has several limitations on this problem. First of all, our dataset is focusing on a particular period of time (i.e., last 2-3 years). So, it is a static dataset, the prices may change in future. As we all know, the different agencies can give different prices of one specific airline, because sometimes, traveling agency might provide coupons on some specific airlines. So other agencies' prices might be more closed to the model of airline companies. Secondly, airline companies might have different price strategy on internal flights and domestic flights. Hence, our model cannot apply to international flights.

Delimitations

A lot of people use Android Operating system so we chose to develop for android for now. We assumed that once this solution goes into action, cross platform application developers can be hired to apply the same code for multiple operating systems, and that can provide this supplication for a large subset of population. With this app, traveller get the fare prediction handy using which it's easy to decide the airlines. It saves time in searching / deciding for airlines.

Chapter 4. Design and Implementation

Architecture

The system consists of two primary components:

1. Neural network: This is where the prediction part takes place. The neural network takes all the processed journey details and predicts the flight fare for that particular journey.
2. Android application: This is used to provide the interaction between user and neural network. User provide journey details to neural network through the UI provided in the android app.

Design of Software – Software Requirements Specification

Overall Description

Product Perspective

We propose a methodology using Machine Learning model namely random forest to predict the prices of flight tickets given the features. The price is estimated based on the number of features as mentioned above. The intricate details about this model on the airfare data set along with the accuracy are narrated in depth in Section V. We then deploy an android application to display our results which are capable of predicting the price of a flight ticket given so many features of it. This deployed service is a result of our work, and it incorporates the data, ML model with the features.

Product Functions

- First, we collect the data about different air routes in India including the features like departure time, arrival time and airways over a specific period, identify important features that reflect the price.
- Second, we pre-process and remove entries with NA values. Discard features that are not relevant for the prediction of the price.
- Third, we apply random forest model on the preprocessed dataset with features as inputs and the price as output.
- Finally, we deploy an android application as a service which incorporates all the features required for a flight journey and the random forest model to predict the

price of a car

- Predicts the minimum ticket price for the given journey details

User Classes and Characteristics

This application can be used as a real-time application which helps users for quick and accurate prediction of a flight ticket price for a given journey.

This application predicts the flight fare based on the following characteristics:

1.Departure time

2.Arrival time

3.Source location

4.Destination location

5.Airline

6.Number of stops

All these classes gives the minimum price for the flight journey

The features that are available to the user are:

- **Accuracy**: - The level of accuracy in the proposed system will be higher. All operation would be done correctly and it ensures that whatever information is coming from the center is accurate.
- **No Redundancy**: - In the proposed system utmost care would be that no information is repeated anywhere, in storage or otherwise. This would assure economic use of storage space and consistency in the data stored.
- **Reliability**: - The reliability of the proposed system will be high due to the above stated reasons. The reason for the increased reliability of the system is that now there would be proper storage of information.
- **Easy to Operate**: - The system should be easy to operate and should be such that it can be developed within a short period of time and fit in the limited budget of the user.

Operating Environment

The product will be operating in any android device with version higher than API 16

Android SDK version 4.1 (Jelly Bean). First, we train the dataset using the Random Forest Algorithm and create a model. Using this model, we build the application in Android Studio with all the required UI components. This application can be operated in offline mode.

Assumptions and Dependencies

The assumptions are:

- The coding should be error free.
- The application should be user-friendly so that it is easy to use for the users.
- The application should give accurate and fast results to the user.

The dependencies are:

- The specific hardware, software and libraries due to which the application will run.
- On the basis of listing requirements and specification the project will be developed andrun.
- The end users should have proper understanding of the product.
- The model should be trained to achieve accuracy.

External Interface Requirements

User Interfaces

The user is asked to enter all the details to calculate the minimum flight fare:

1.Departue time

2.Arrival time

3.Source location

4.Destination location

5.Airline

6.Number of stops

Hardware Interfaces

Device: Android device with minimum SDK version API 16: Android 4.1 (Jelly Bean)

Storage Disk (Optional): 20 MB of storage on device

Software Interfaces

Software: Anaconda, Python 3.x (3.8 or earlier)

Editor: VS Code/ PyCharm/ Sublime/ Spyder

IDE: Android Studio

Packages: pandas, numpy, sklearn, seaborn,matplotlib

Communications Interfaces

There are no external communications interface requirements, as we are using a specific data set.

System Features

Data Preprocessing: "Before Training, any model using any algorithm Data Preprocessing is the most significant step and will be the primary step. The data preprocessing contains several checkpoints (steps) such as: "

1. Import Libraries: The essential libraries for Data preprocessing we used are Pandas fordata manipulation and analysis, Numpy for numerical analysis, Matplotlib and Seaborn for better visuals and graphical stats of the data.

2. Import the Dataset: First downloaded this dataset from Kaggle, and then read the downloaded dataset using the pandas library.

3. Taking care of Missing Data in Dataset: After evaluation of this dataset, we filled missing values in the dataset.

4. Encoding categorical data: This dataset contains some Categorical values such as Source, Destination, airline, so we need to encode these categorical data into an encoded formatto better train our model, to do this we used get_Dummies() method of pandas and this converted the whole Categorical values in the dataset into binary values.

5. Splitting the Dataset into the Training set and Test Set: To split this dataset into Test and Train dataset to train our machine learning model we used the capable machine learning library of python, scikit-learn or sklearn. Using its model selection method to create testing data by picking random values from the available dataset for model prediction, or we can say Supervised Learning

Data Training and Modelling:

To train and develop a model, first of all, we need to the dependent and independent variables. Tofind these variables, first I used to find the correlation between the variables of the output and then separates my variables into two different axes we call it x and y where the x-axis contains all the independent variable and y-axis having the dependent variable, in our model it is the flight fare of a given journey. Using sklearn.model_selection library and its train_test_split function, further this dataset is distributed in the train-test dataset to find the best hyperparameters for our model prediction.

Proposed Model:

The proposed model is an application of the machine learning algorithm i.e. Random Forest Algorithm. In this model first, the dataset is loaded for further exploration. In this specific model, we used a Dataset available at Kaggle. After performing the Data preprocessing steps on this dataset such as handling missing values, one hot encoding of Categorical Values, we start training the model for distributed dataset into two 1. Training Dataset and 2. Test Dataset. This test data is picked randomly from the original dataset. Applied the machine Learning algorithm i.e. Random Forest Algorithm and done tuning of the Hyperparameters to get the best Hyper-Parameters for result prediction. Once the model predicts a result, it prints out the predicted price of flight ticket based on given parameters.

Other Nonfunctional Requirements

Performance Requirements

Performance is one of the most important aspect for this system. The system should have a highperformance such that the user can see the minimum flight ticket price.

Safety Requirements

There are no safeguards or actions that must be taken, as well as actions that must be prevented.

Security Requirements

Security must be maintained in protecting the dataset. As changes in the dataset can change theticket price. Only authorized users should be given access to the backend of the application.

Software Quality Attributes

This application is reusable even as the days pass on by changing the dataset. The end user needsto be aware of the fact that all the outputs obtained from the model may not be what it is expected. i.e., a positive review on a product might end up yielding a fairly neutral output.

Design Diagrams

ER Diagram

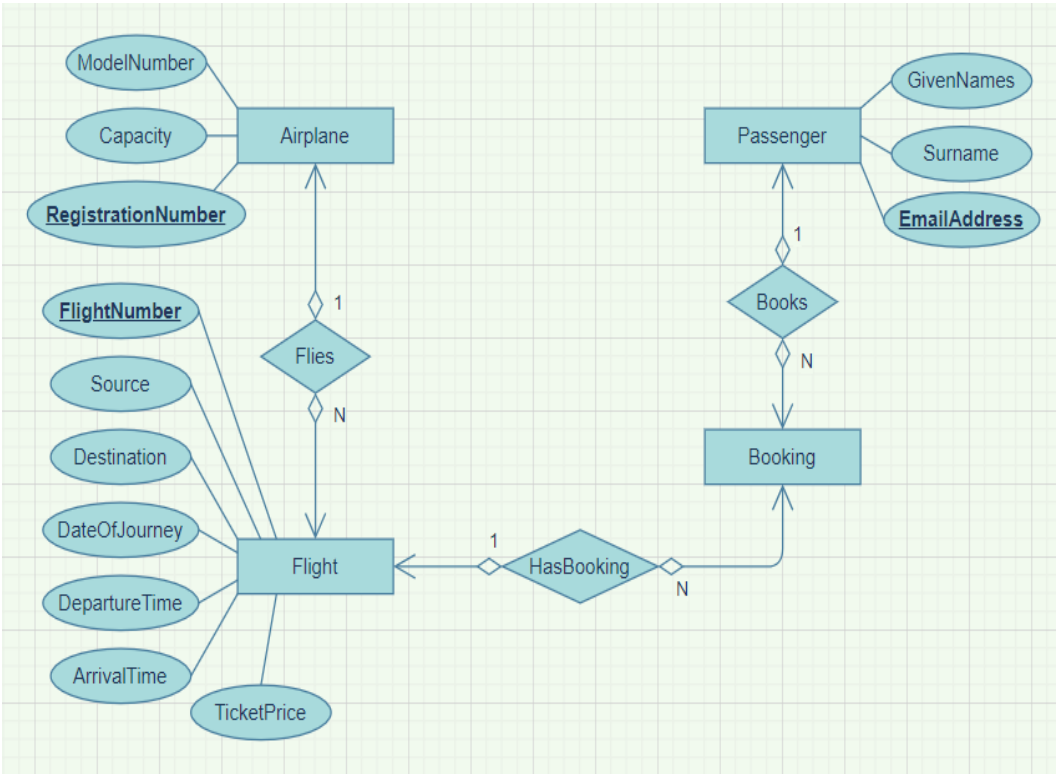


Figure 1. ER Diagram

Entity Relationship Diagram, as shown in Figure 1 shows all the relationships among various entities like Airplane, Flight, Booking and Passengers. Primary key of each entity is shown in bold and underlined format.

Use Case Diagram

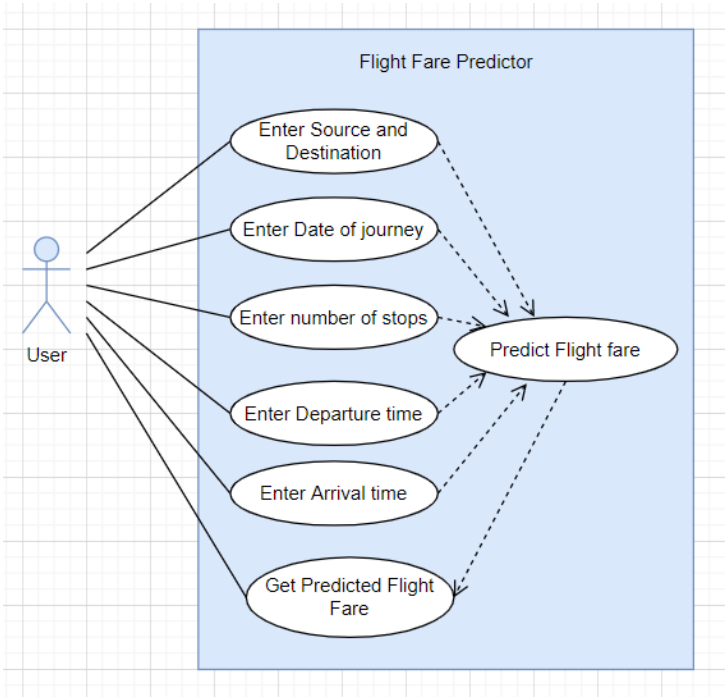


Figure 2. Use Case Diagram

Figure 2 shows the Use Case diagram for this Case Study. This Use case diagram is a graphical representation of the interactions among the elements of the system. The actor here is the user who uses this application. User will be having the use cases – Enter Source and Destination, enter date of journey, enter number of stops, enter departure and arrival time. Then the system predicts flight fare for the given details and displays output to the user, which is another use case of the user.

Sequence Diagram

Sequence diagram, as shown in Figure 3 is a type of interaction diagram because it describes how—and in what order—a group of objects works together. It shows the order of activities that take place among user, android application and neural network.

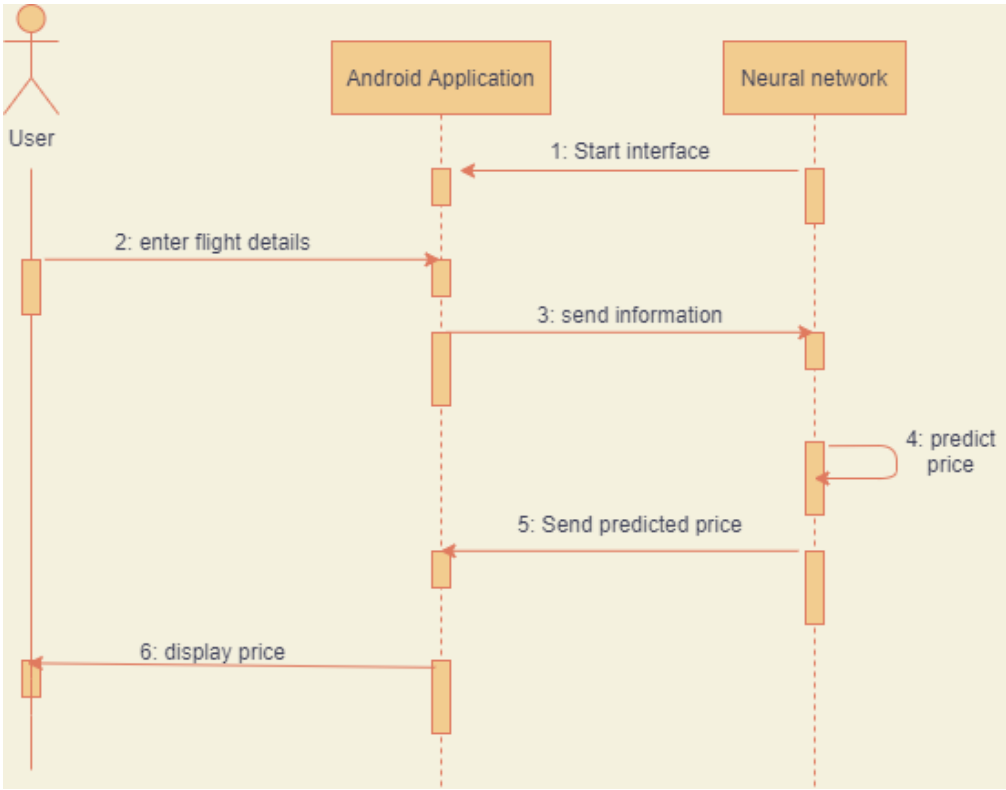


Figure 3. Sequence Diagram

Collaboration Diagram

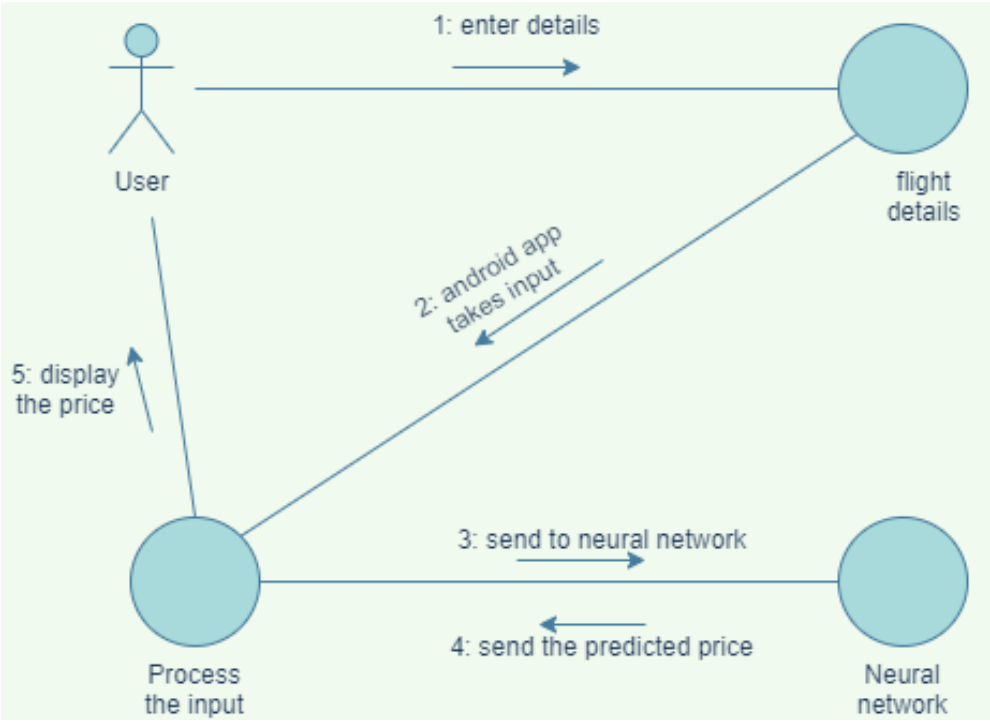


Figure 4. Collaboration Diagram

The collaboration diagram, as shown in Figure 4 is used to show the relationship between the objects in the system. Both the sequence and the collaboration diagrams represent the same information but differently. Instead of showing the flow of messages, it depicts the architecture of the object residing in the system as it is based on object-oriented programming.

Activity Diagram:

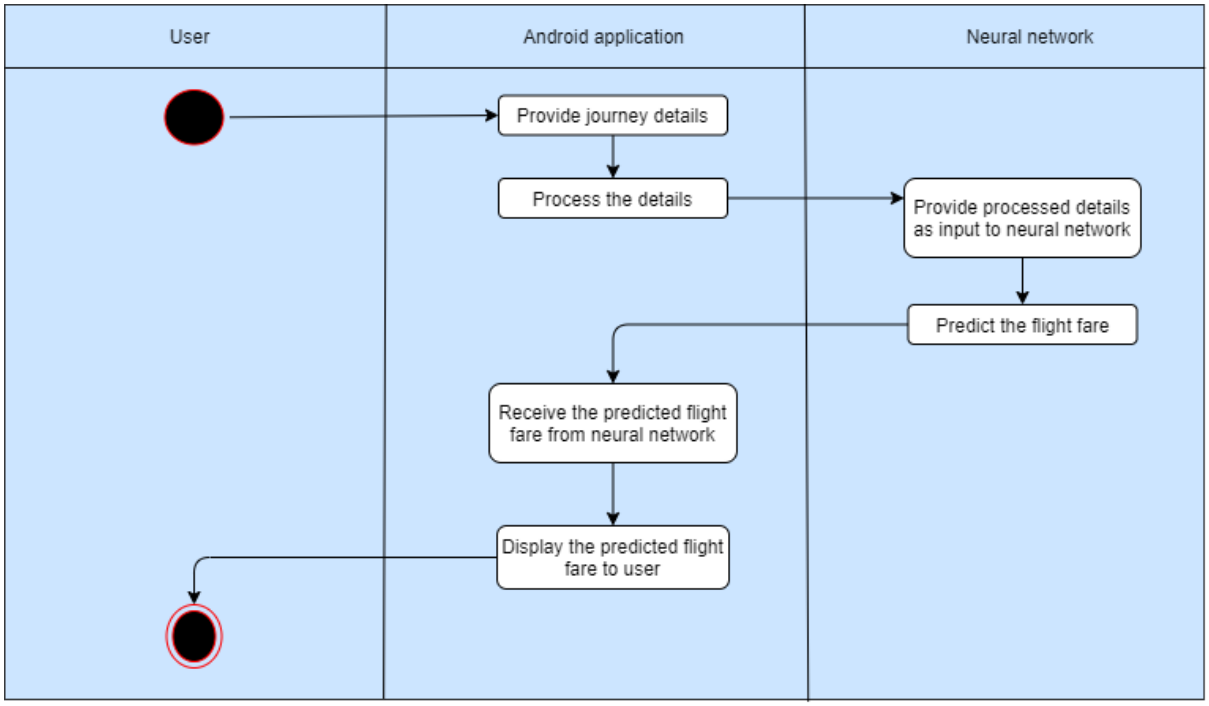


Figure 5. Activity Diagram

Activity diagram, as shown in Figure 5 is another important diagram in UML to describe the dynamic aspects of the system. Activity diagram is basically a flowchart to represent the flow from one activity to another activity. This diagram shows various activities and their flow in the system.

State Chart Diagram:

State chart diagram describes the flow of control from one state to another state. States are defined as a condition in which an object exists and it changes when some event is triggered.

The most important purpose of state chart diagram is to model lifetime of an object from creation to termination. Figure 6 shows State chart diagram for this system.

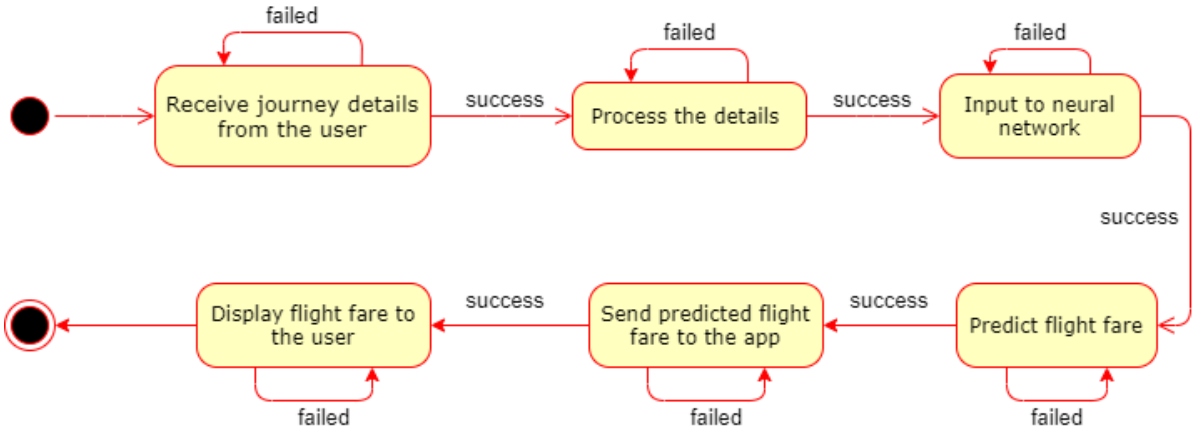


Figure 6. State Chart Diagram

Component Diagram:

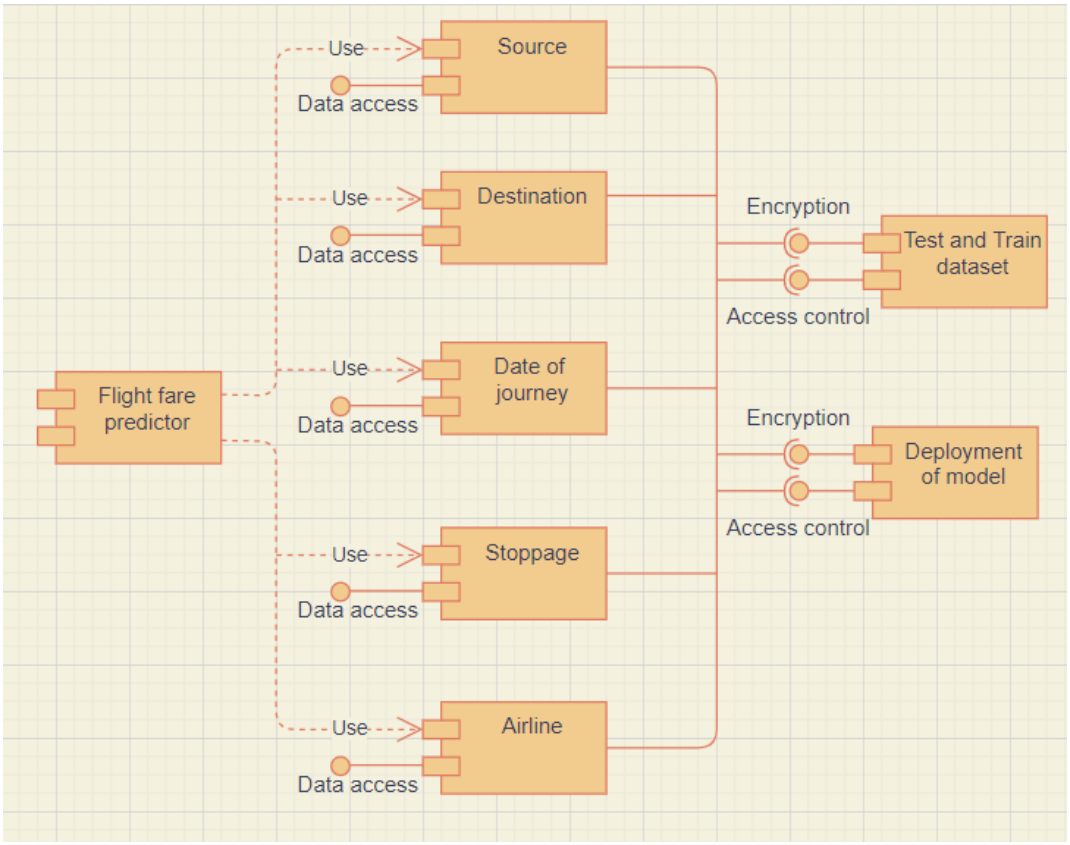


Figure 7. Component Diagram

The purpose of a component diagram is to show the relationship between different components in the system. Component refers to a module of classes that represent independent systems or subsystems with the ability to interface with the rest of the system. Figure 7 shows Component diagram for Flight fare prediction system.

Deployment Diagram

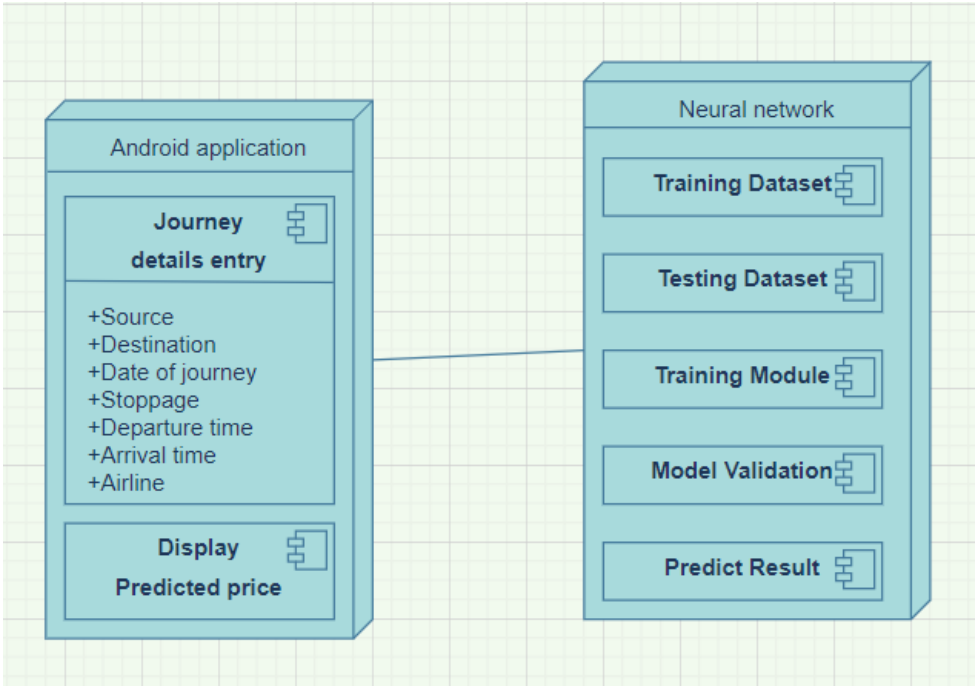


Figure 8. Deployment Diagram

Deployment diagram is used to visualize the hardware processors/ nodes/ devices of the system, the links of communication between them and the placement of software files on that hardware. Figure 8 shows deployment diagram for this Case study, where we have Android application and Neural network and a communication link between them.

Implementation of the proposed system

Dataset collection and applying ML algorithms:

Link for dataset, available in Kaggle: https://www.kaggle.com/absin7/airlines-fare-prediction

We will get two different datasets from the source. One is the Train set and another one is Test set. The information of the two datasets is given below:

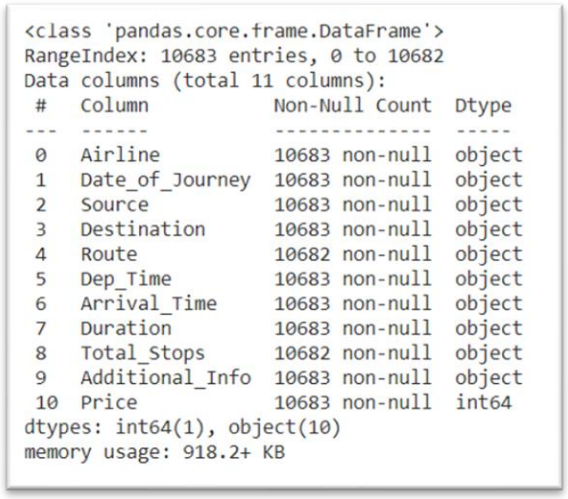


Figure 9. Train set

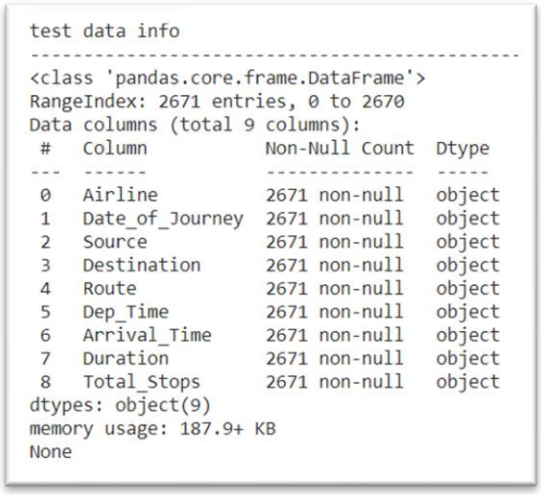


Figure 10. Test set

The datasets consist of various features as shown in Figure 9 and Figure 10. Train set consists of 10683 rows and 11 columns. Test set consists of 2671 rows and 10 columns.

We trained our model using Random Forest algorithm and then we built a neural network.

Random forest:

This is an algorithm which ensembles the less predictive model to produce better predictive models. It aggregates the base model to create a large model. The features are sampled and passed to trees without replacement to obtain the highly uncorrelated decision trees. To select the best split, it is required to have less correlation between the trees. The main concept that makes random forest different from the decision tree is aggregated uncorrelated trees

Neural network:

Deep learning is an increasingly popular subset of machine learning. Deep learning models are built using neural networks. A neural network takes in inputs, which are then processed in

hidden layers using weights that are adjusted during training. Then the model spits out a prediction. The weights are adjusted to find patterns in order to make better predictions. The user does not need to specify what patterns to look for — the neural network learns on its own.

The model type that we will be using is Sequential. Sequential is the easiest way to build a model in Keras. Keras is a user-friendly neural network library written in Python. It allows you to build a model layer by layer. Each layer has weights that correspond to the layer the follows it.

We use the 'add()' function to add layers to our model. We will add two layers and an output layer.

'Dense' is the layer type. Dense is a standard layer type that works for most cases. In a dense layer, all nodes in the previous layer connect to the nodes in the current layer.

'Activation' is the activation function for the layer. An activation function allows models to take into account nonlinear relationships. The activation function we will be using is ReLU or Rectified Linear Activation. Although it is two linear pieces, it has been proven to work well in neural networks.

Tflite model:

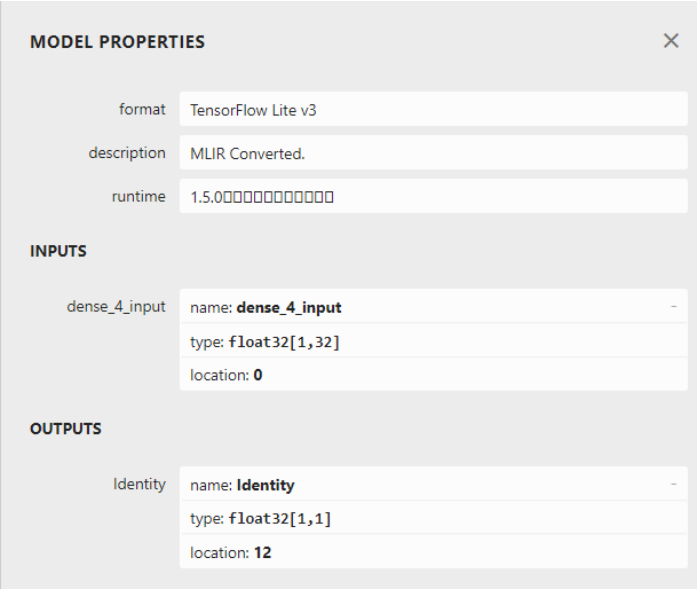
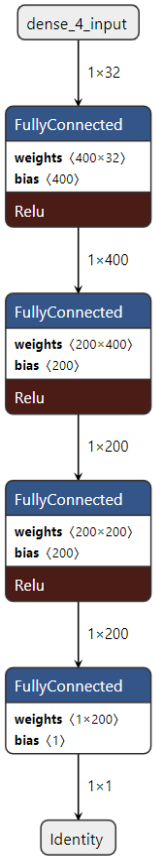


Figure 11. tflite model properties

Now, we have to use neural network in order to deploy it in android application. We have generated a **tflite** file of the model to use it in android studio. It is a quantised model, i.e., it compresses the size of the model to deploy it in another application without reduction of the prediction accuracy.



From Figure 11, we can observe the properties of tflite model. It takes a float type array containing 32 entries, and finally output is a float type array containing 1 entry (i.e., the ticket price).

Figure 12 shows the architecture of neural network, there are 3 hidden layers using Relu activation function. It takes a 2D array containing 32 entries and it processes the input through the layers and gives the predicted flight fare through the final layer as a 2D array with single element.

We add this tflite file to assets folder in android studio and access it by adding required dependencies in the build.

Figure 12 Architecture of tflite model

Android application:

We built the app in a user-friendly manner where we prompt the user to provide six journey details. Predicting the price is the area where the application of machine learning come into picture. The app takes date of journey, departure time and arrival time, etc., from the user and provides price of the ticket fare of available airlines.

We used Spinner class in Android Studio to display a drop-down menu for Source, Destination and Stoppage. We used Date Picker and Time Picker for date of journey, Departure time and Arrival time.

Figure 13. Home

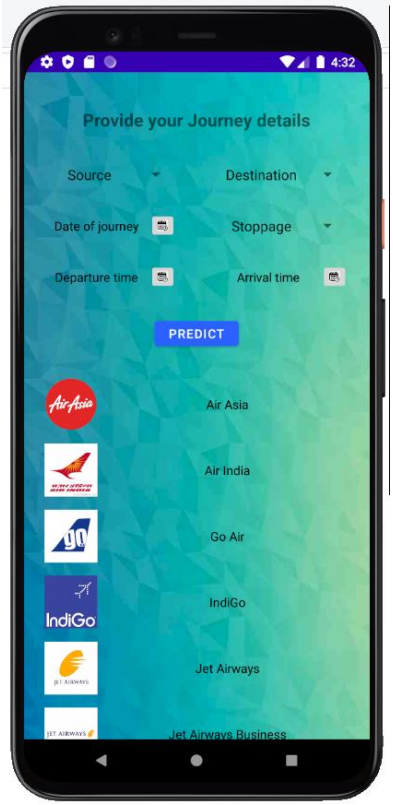


Figure 14. Main Screen

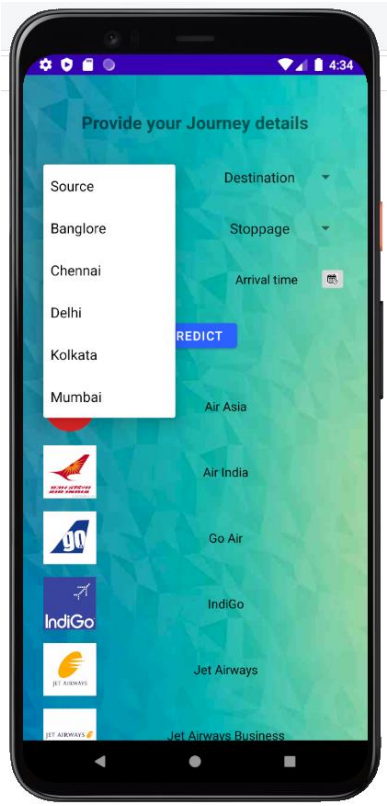


Figure 15. Drop-down menu

Figure 13 shows home page, we have a button, Let's Go! which direct us to the next screen on clicking it.

In the next screen, as shown in Figure 14, user has to provide the journey details like Source, Destination, Journey date, Number of stops, Departure time and Arrival time.

Figure 15 shows selection of Source. We provided a drop-down menu for selecting available Sources.

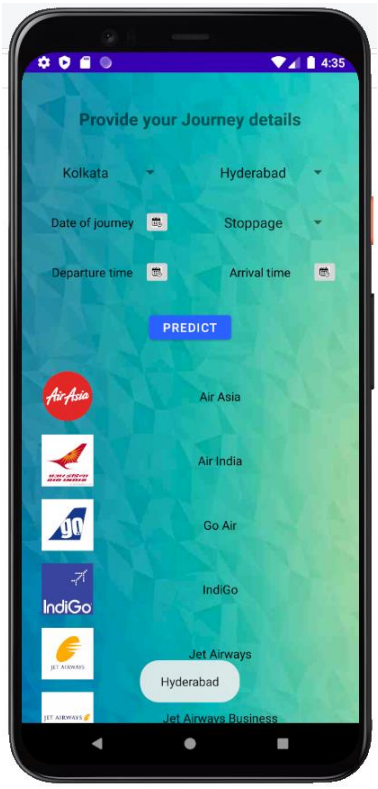


Figure 16. Displaying toast

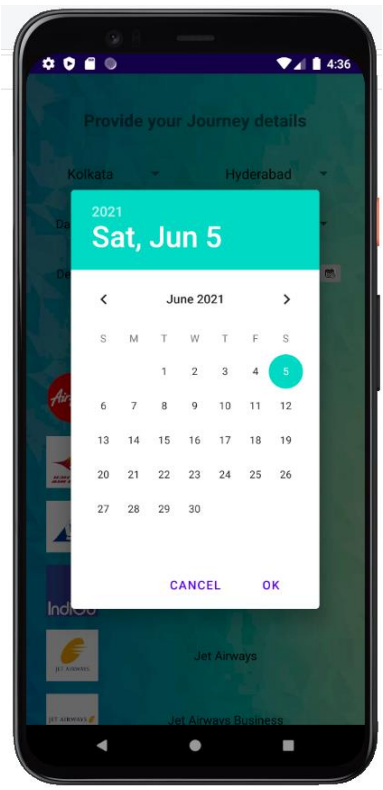


Figure 17. Date picker

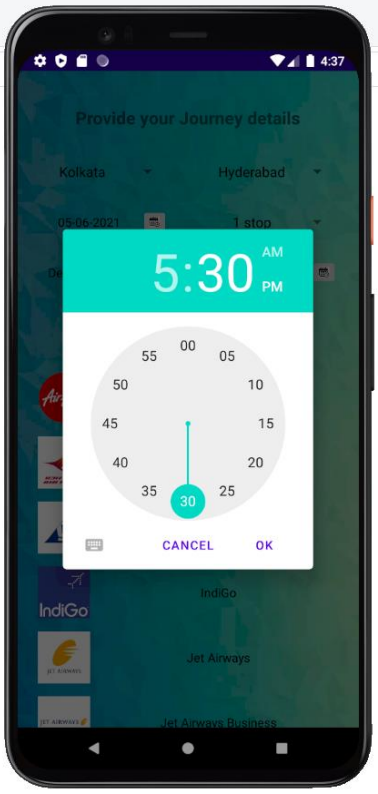


Figure 18. Time picker

Whenever user selects a source or destination or stoppage, a toast is displayed on the screen as shown in Figure 16.

Figure 17 shows Date picker, when the user clicks on the calendar icon on the screen that is available beside Date of journey, it opens the calendar and prompts the user to select the date of journey.

Figure 18 shows Time Picker, when the user clicks on the calendar icon on the screen that is available beside Departure time or arrival time, it opens the clock and prompts the user to select the time of departure and time of arrival of Flight.

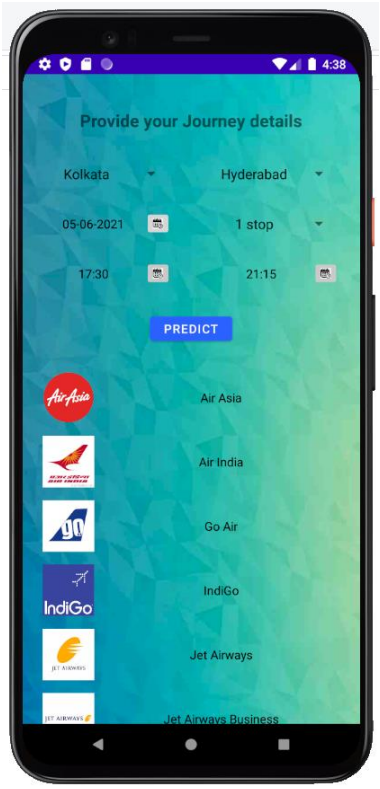


Figure 19. Click on Predict button

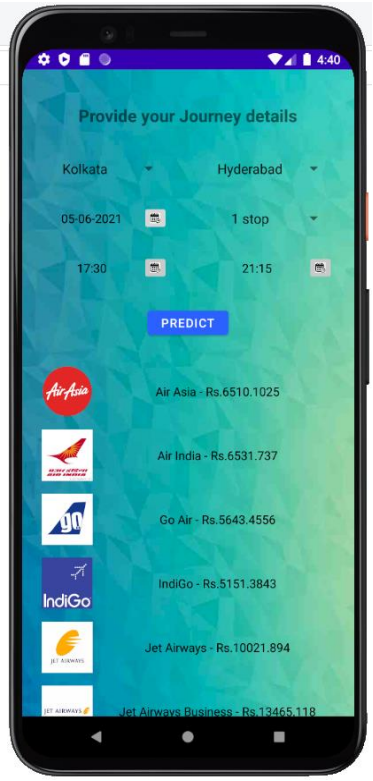


Figure 20. Display fares

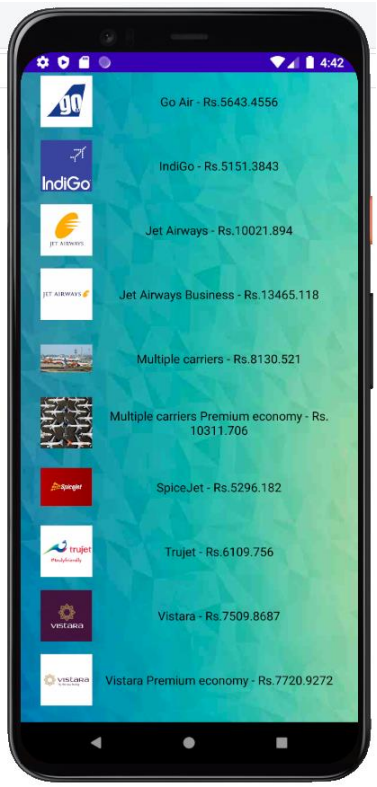


Figure 21. Display fares

Figure 19 shows the screen after filling all the journey details, now we have to click on the Predict button.

After clicking on Predict button, flight fares for various airlines on provided journey details will be displayed on the screen as shown in Figure 20 and Figure 21.

Chapter 5. Result analysis and Recommendations

The solution has been successfully implemented according to the proposed software design. The application provides fares of all available airlines for their journey. It so does that task efficiently without much resource utilization. The solution helps users in finding available flights and get an insight of fares in future. Random forest gives an accuracy of 95.6% for training set and 82.2% for test set as shown in Figure 22.

```
[ ] reg_rf.score(x_train,y_train)

    0.9561732730228009

[ ] reg_rf.score(x_test,y_test)

    0.822412855094374
```

Figure 22. Accuracy of Random Forest

Similarly, the epoch vs. loss value graph is plotted across train and test set and is shown in Figure 23.

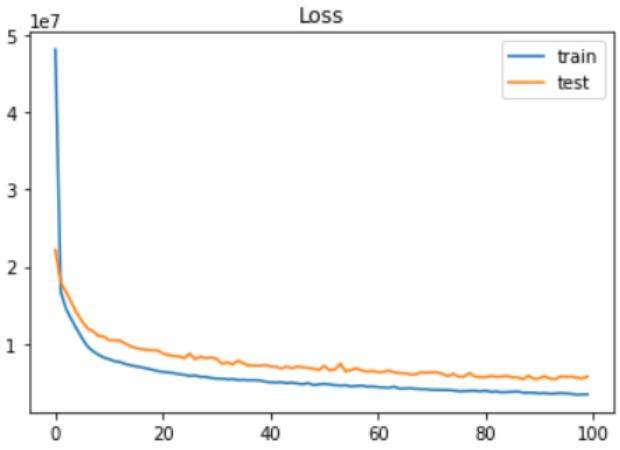


Figure 23. Epoch vs. Loss graph

People who often travel through flights will have an idea about change in flight fares as all airline companies will utilise their privilege to change their flight fares time to time to manage their revenue. But people who travel through flights once in a while will don't have much idea about variation in flight fares, so these people can make use of this application and get an idea about all available airlines with their accurate flight fares on their journey day.

Chapter 6. Conclusion and Discussions

In this project, we have practiced different machine learning techniques and different models for data training, attempting to achieve the highest accuracy of predicting the flight fare. Thus, this study settled on predicting ticket price based on given user's journey details using regression algorithm and consequently testing its accuracy.

We conclude Neural network is the preferred choice in terms of its high accuracy and computational efficiency and for the deployment of the model in android studio. However, there is no single algorithm that works best on all given problems.

So, with this project, we successfully implemented the flight fare prediction.

This project has not implemented flight ticket reservation. So, we can access location of the user and recommend nearby airports and we can also work on cancelling tickets in the last moment as per the convenience of the user with possibly less pre-booking charges. We can even add a chatbot which can interact with the user and help as a journey assist to the user.

References

1. https://www.kaggle.com/absin7/airlines-fare-prediction
2. http://www.ijstr.org/final-print/dec2019/Predicting-The-Price-Of-A-Flight-Ticket-With-The-Use-Of-Machine-Learning-Algorithms.pdf
3. https://www.researchgate.net/publication/337821411_Predicting_Flight_Prices_in_India
4. https://www.youtube.com/watch?v=y4EMEpEnElQ