

Dr BR Ambedkar National Institute of Technology, Jalandhar.



Minor Project on:
Flight Ticket Price Prediction

Guided by:
Dr Narendra Kumar

Presentation by:
Ayush Singh

ACKNOWLEDGMENT

- I would like to take this opportunity to express our profound gratitude and deep regard to **Dr. Narendra Kumar**, for his exemplary guidance, valuable feedback and constant encouragement throughout the duration of the project. His valuable suggestions were of immense help throughout our project work. His perceptive criticism kept us working to make this project in a much better way. Working under him was an extremely knowledgeable experience for us. I ensure that this project was done by me and is not copied.
- Ayush Singh
- 18113019

Abstract

Flight ticket fare is the most fluctuating data which varies every day. Depending on the various factors that affect it directly or indirectly, we cannot say that the price of flight ticket fare remains the same or not. It is quite a tough task to predict the flight ticket fare. It may change throughout the week, month or some days, but it can be predicted nearly accurate to the actual flight ticket fare.

The prime objective of my project “ Flight Price Prediction ” is to make a prediction of the flight ticket fare for the future flights. The proposed approach is using machine learning algorithm and using supervised learning.

The regression model which I have selected for my prediction is “Random Forrest”. In this approach I have developed this project in python language using Jupyter Notebook also after training the model a flask app has been made and it is deployed on HEROKU app where we can predict the prices.

Introduction

- Nowadays, the airline corporations are using complex strategies for the flight ticket fare calculations. This highly complicated methods makes the flight ticket fare difficult to guess for the customers, since the fare changes dynamically.
- My project **Flight Price Prediction** which resolve this problem and provide a facility where people will be able to predict the flight-ticket price before purchasing the ticket.

Objective

The prime objective of this project is to use machine learning techniques to model the behavior of flight ticket prices over the time and predict the price of the flight-ticket.

The goal of this project is to study how the machine learning models will predict the prices by using feature engineering and doing the operation of Exploratory Data Analysis and to show what factors are highly correlated to price of tickets and at last deploy it creating a web app on HEROKUAPP.

INTRODUCTION TO REGRESSION

- **What Is Regression?**
- Regression is a statistical method used in finance, investing, and other disciplines that attempts to determine the strength and character of the relationship between one dependent variable (usually denoted by Y) and a series of other variables (known as independent variables).
- **Regression Explained**
- The two basic types of regression are simple linear regression and multiple linear regression, although there are non-linear regression methods for more complicated data and analysis. Simple linear regression uses one independent variable to explain or predict the outcome of the dependent variable Y , while multiple linear regression uses two or more independent variables to predict the outcome.

GENERAL FORMS OF REGRESSION

- The general form of each type of regression is:
- **Simple linear regression:** $Y = a + bX + u$
- **Multiple linear regression:** $Y = a + b_1X_1 + b_2X_2 + b_3X_3 + \dots + b_tX_t + u$
- Where:
- Y = the variable that you are trying to predict (dependent variable).
- X = the variable that you are using to predict Y (independent variable).
- a = the intercept.
- b = the slope.
- u = the regression residual.

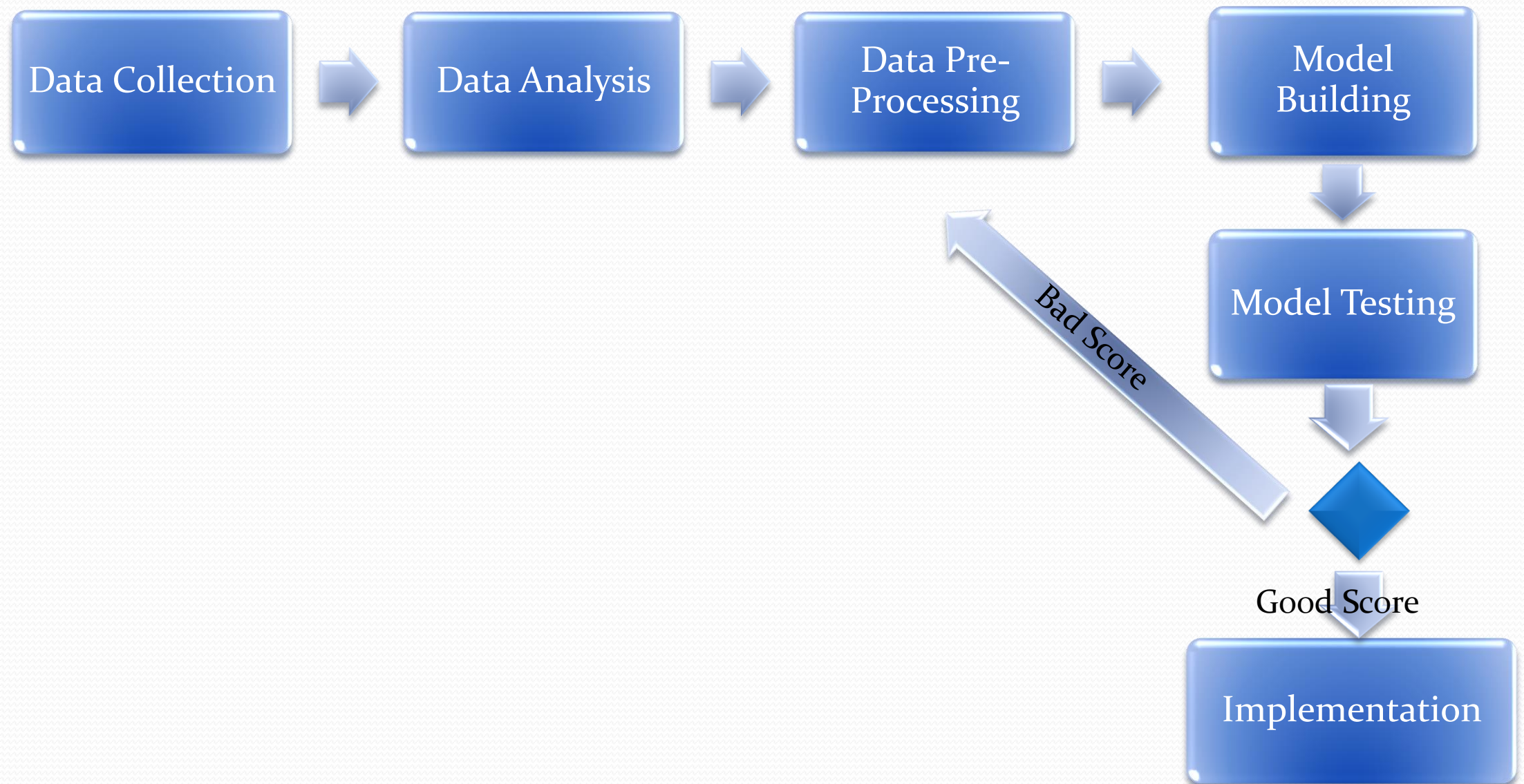
Exploratory Data Analysis (EDA)

- As its name suggests, the main aim of the **exploratory analysis** is to explore. Prior to it, there's still no notion of the relationship between the data and the variables. Once the data is investigated, the exploratory analysis enables you to find connections and generate hypotheses and solutions for specific problems. A typical area of application for exploratory analysis is data mining.
- **• Data Preparation:** This is a good time to visualize your data and check if there are correlations between the different characteristics that we obtained. It will be necessary to make a selection of characteristics since the ones you choose will directly impact the execution times and the results.
- **• Model Building :** There are several models that you can choose according to the objective that you might have: you will use algorithms of classification, prediction, linear regression, clustering, i.e. k-means or K-Nearest Neighbor, Deep Learning, i.e. Neural Networks, Bayesian, etc. There are various models to be used depending on the data you are going to process such as images, sound, text, and numerical values.

Splitting the Data into Training and Testing sets:

- You will need to train the datasets to run smoothly and see an incremental improvement in the prediction rate. Remember to initialize the weights of your model randomly -the weights are the values that multiply or affect the relationships between the inputs and outputs which will be automatically adjusted by the selected algorithm the more you train them.
- **Making Predictions** : You are now ready to use your Machine Learning model inferring results in real-life scenarios.
- **Model Evaluation**: machine created against your evaluation data set that contains inputs that the model does not know and verify the precision of your already trained model. If the accuracy is less than or equal to 50%, that model will not be useful since it would be like tossing a coin to make decisions. If you reach 90% or more, you can have good confidence in the results that the model gives you.

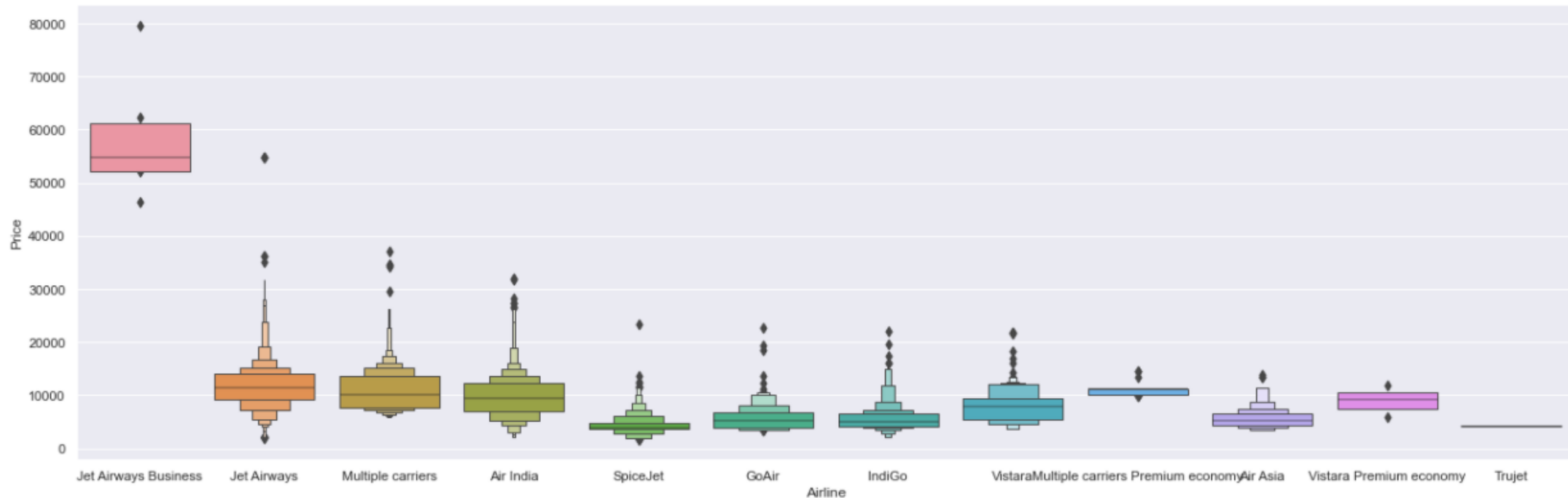
Flow chart



From graph we can see that **Jet Airways Business** have the highest Price
Apart from the first Airline almost all are having similar median

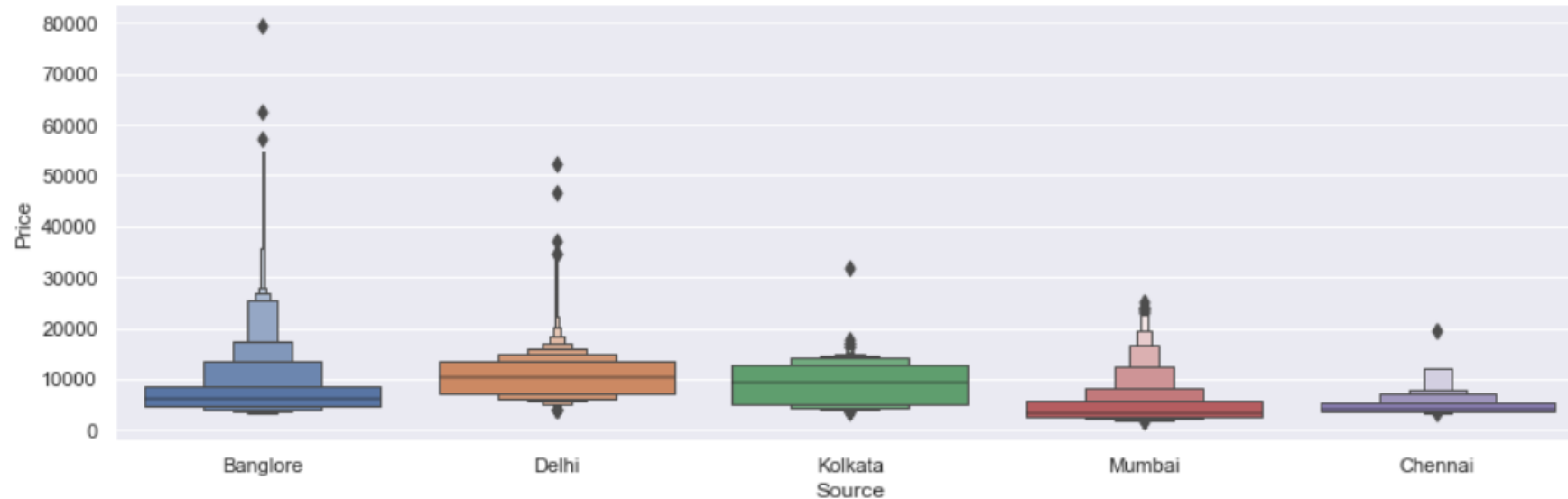
```
# Airline vs Price
```

```
sns.catplot(y = "Price", x = "Airline", data = train_data.sort_values("Price", ascending = False), kind="boxen", height = 6, aspect_ratio = 1.5, plt.show())
```



Source vs price- This feature basically tells us about how source is affecting the price. This clarifies that when a person is travelling to Bangalore or Mumbai which price is high.

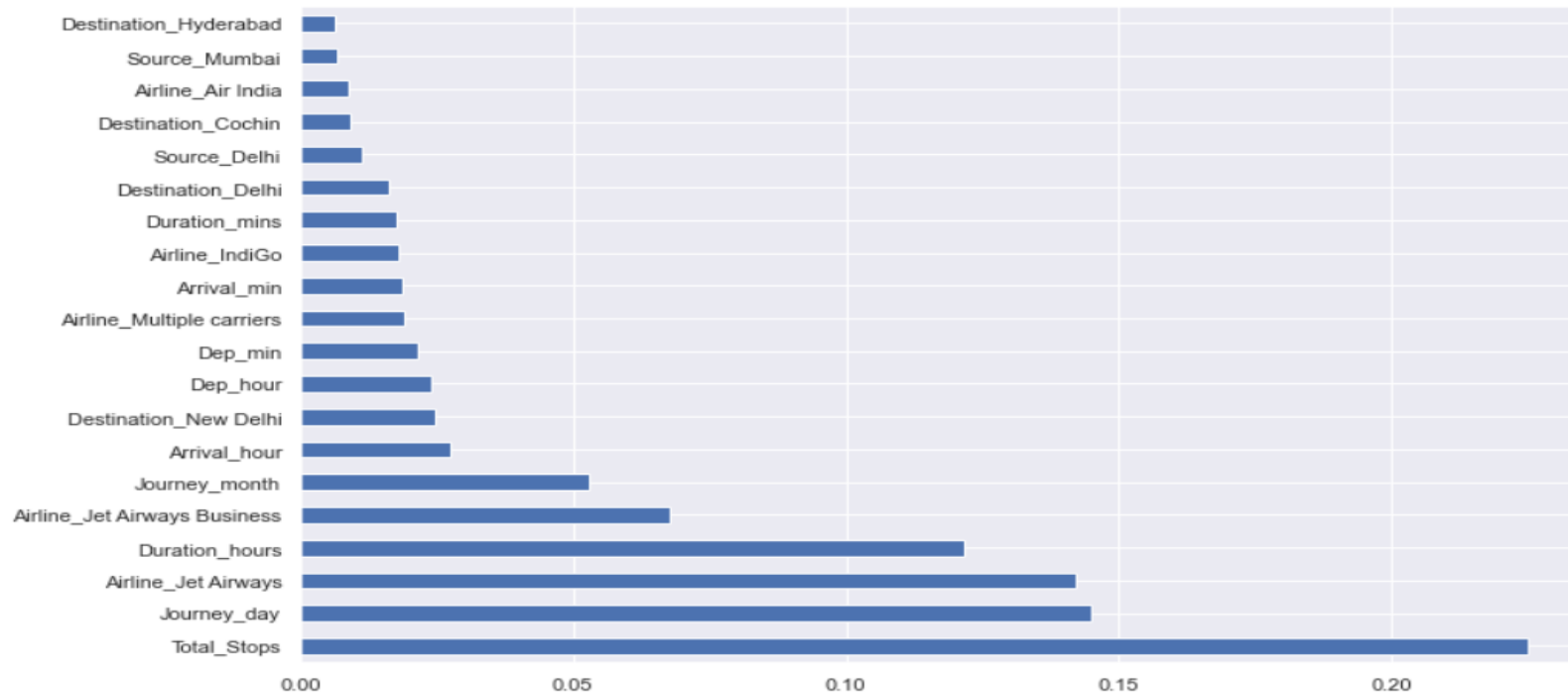
```
sns.catplot(y = "Price", x = "Source", data = train_data.sort_values("Price", ascending = False), kind="boxen", height = 4, aspect = 4, plt.show())
```



This graph shows that **Bangalore** has more outliers than others hence Bangalore's price is high.

This Bar-Graph Indicates that which features are more important and how much they are significant to the prediction model

```
plt.figure(figsize = (12,8))  
feat_importances = pd.Series(selection.feature_importances_, index=X.columns)  
feat_importances.nlargest(20).plot(kind='barh')  
plt.show()
```



- It clearly shows that **Total Stops** is the most important feature and 2nd most important feature is **Journey Day** etc

Fitting Model using Random Forrest

- Split dataset into train and test set in order to prediction w.r.t X_{test}
- If needed do scaling of data
 - Scaling is not done in Random forest
- Import model
- Fit the data
- Predict w.r.t X_{test}
- In regression check **RSME** Score
- Plot graph

```
In [61]: from sklearn.model_selection import train_test_split  
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size = 0.2, random_state = 42)
```

```
In [62]: from sklearn.ensemble import RandomForestRegressor  
reg_rf = RandomForestRegressor()  
reg_rf.fit(X_train, y_train)
```

```
Out[62]: RandomForestRegressor()
```

```
In [63]: y_pred = reg_rf.predict(X_test)
```

```
In [64]: reg_rf.score(X_train, y_train)
```

```
Out[64]: 0.953341916559342
```

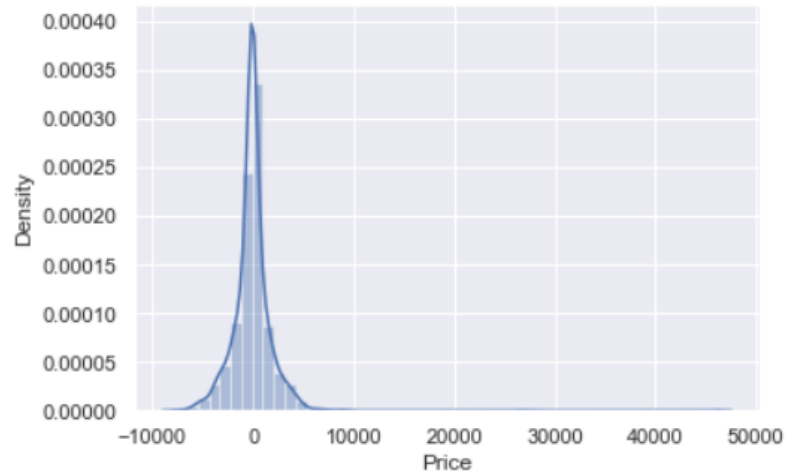
```
In [65]: reg_rf.score(X_test, y_test)
```

```
Out[65]: 0.7975387038344752
```

Plotting y_{test} and y_{pred} .

```
In [66]: sns.distplot(y_test-y_pred)  
plt.show()
```

```
c:\python37\lib\site-packages\seaborn\distributions.py:2551: FutureWarning: `distplot` is a deprecated function and will be removed in a future version. Please adapt your code to use either `displot` (a figure-level function with similar flexibility) or `histplot` (an axes-level function for histograms).  
warnings.warn(msg, FutureWarning)
```



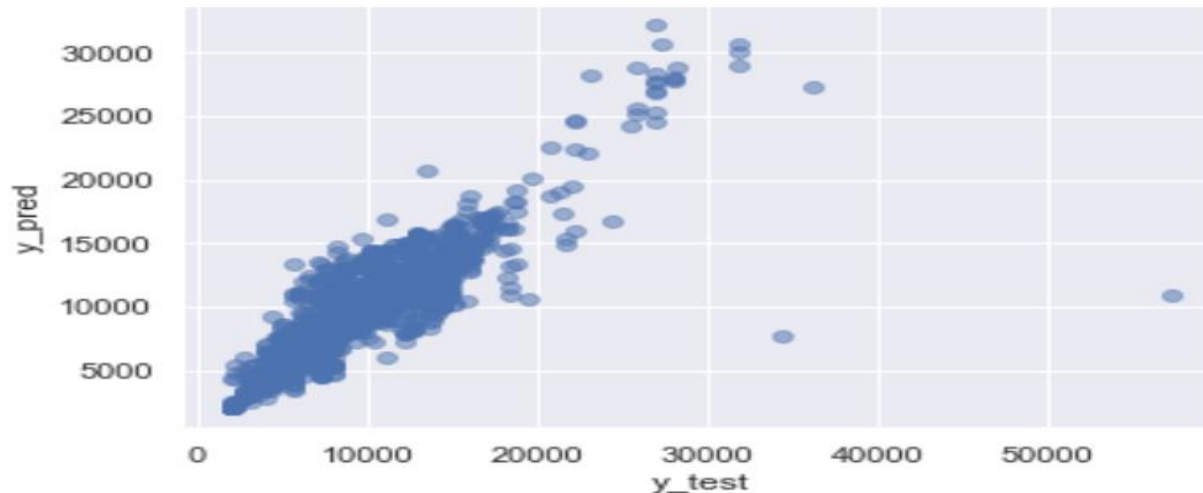
- The graph between y_{test} and y_{pred} is forming a Gaussian Distribution that means our result is very good itself.

Scatter Plot

- A scatter plot uses dots to represent values for two different numeric variables. The position of each dot on the horizontal and vertical axis indicates values for an individual data point. Scatter plots are used to observe relationships between variables.
- From the graph we can clearly see that independent and dependent variable are highly correlated as it is showing strong , positive and linear relation.

In [67]:

```
plt.scatter(y_test, y_pred, alpha = 0.5)  
plt.xlabel("y_test")  
plt.ylabel("y_pred")  
plt.show()
```



Hyperparameter Tuning:

- In machine learning, hyperparameter optimization or tuning is the problem of choosing a set of optimal hyperparameters for a learning algorithm. A hyperparameter is a parameter whose value is used to control the learning process. By contrast, the values of other parameters (typically node weights) are learned.
- Choose following method for hyperparameter tuning
 1. **RandomizedSearchCV** --> Fasst
 2. **GridSearchCV**
- Assign hyperparameters in form of dictionary
- Fit the model
- Check best parameters and best score

Randomized SearchCV .

- Scikit-learn provides **RandomizedSearchCV** class to implement random search. It requires two arguments to set up: an estimator and the set of possible values for hyperparameters called a *parameter grid* or *space*.

```
In [72]: from sklearn.model_selection import RandomizedSearchCV
```

```
In [73]: #Randomized Search CV

# Number of trees in random forest
n_estimators = [int(x) for x in np.linspace(start = 100, stop = 1200, num = 12)]
# Number of features to consider at every split
max_features = ['auto', 'sqrt']
# Maximum number of levels in tree
max_depth = [int(x) for x in np.linspace(5, 30, num = 6)]
# Minimum number of samples required to split a node
min_samples_split = [2, 5, 10, 15, 100]
# Minimum number of samples required at each leaf node
min_samples_leaf = [1, 2, 5, 10]
```

```
In [74]: # Create the random grid

random_grid = {'n_estimators': n_estimators,
               'max_features': max_features,
               'max_depth': max_depth,
               'min_samples_split': min_samples_split,
               'min_samples_leaf': min_samples_leaf}
```

```
In [75]: # Random search of parameters, using 5 fold cross validation,
# search across 100 different combinations
rf_random = RandomizedSearchCV(estimator = reg_rf, param_distributions = random_grid, scoring='neg_mean_squared_error', n_iter = 100)
```

```
In [76]: rf_random.fit(X_train,y_train)
```

Fitting 5 folds for each of 10 candidates, totalling 50 fits

After Using hyperparameter tuning technique:

- By using this technique we can see that y_{test} prediction is also showing a Gaussian distribution and scatter plot between y_{test} and y_{predict} is also highly correlated as it shows strong, positive and linear relationship and most important use of this tuning is it gives increased value of **R-Square**.

Save the model to reuse it again

```
In [90]: import pickle
# open a file, where you want to store the data
file = open('flight_rf.pkl', 'wb')

# dump information to that file
pickle.dump(rf_random, file)
```

```
In [91]: model = open('flight_rf.pkl', 'rb')
forest = pickle.load(model)
```

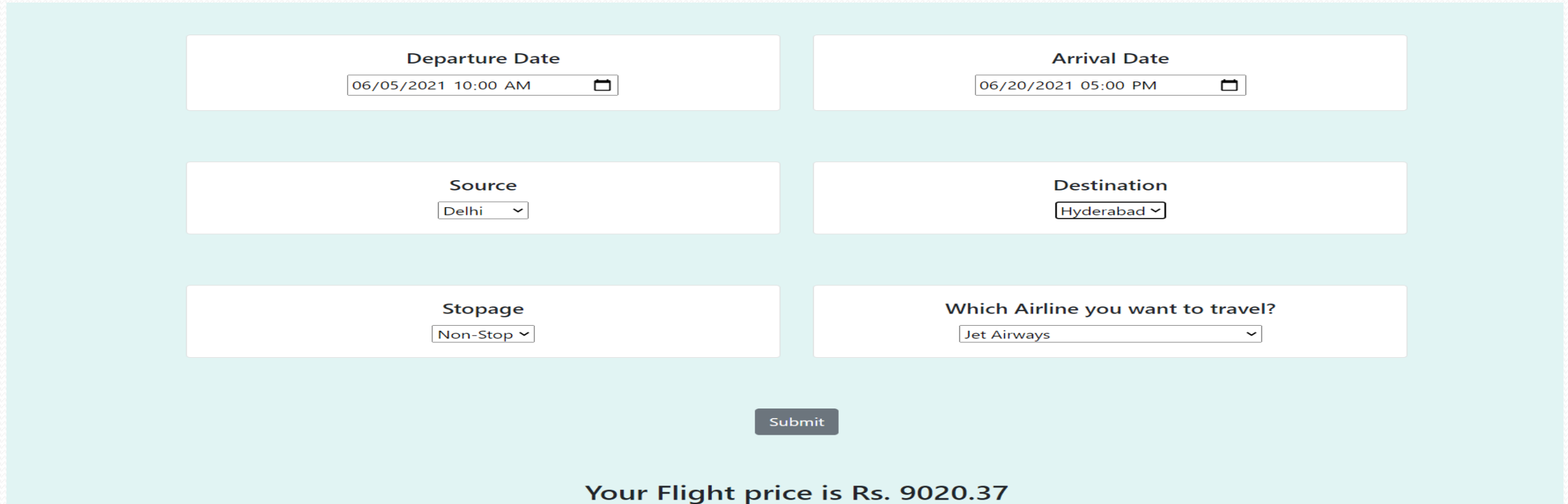
```
In [92]: y_prediction = forest.predict(X_test)
```

```
In [93]: metrics.r2_score(y_test, y_prediction)
```

```
Out[93]: 0.8114531314016973
```

Finally after applying machine learning model and calculated the predictions a web app has been made and it is deployed on HEROKUapp where we can predict ticket price for future

- Output Screen:



The screenshot displays a web application interface for predicting flight prices. It features a light blue background with a white grid of input fields. The fields are arranged in two columns and three rows. The first row contains 'Departure Date' and 'Arrival Date' fields, both showing dates and times. The second row contains 'Source' and 'Destination' dropdown menus, with 'Delhi' and 'Hyderabad' selected. The third row contains 'Stopage' and 'Which Airline you want to travel?' dropdown menus, with 'Non-Stop' and 'Jet Airways' selected. A 'Submit' button is centered below the input fields. At the bottom, a message states 'Your Flight price is Rs. 9020.37'.

Field	Value
Departure Date	06/05/2021 10:00 AM
Arrival Date	06/20/2021 05:00 PM
Source	Delhi
Destination	Hyderabad
Stopage	Non-Stop
Which Airline you want to travel?	Jet Airways

Submit

Your Flight price is Rs. 9020.37

Future Scope:

- More routes can be added and the same analysis can be expanded to major airports and travel routes in India.
- The analysis can be done by increasing the data points and increasing the historical data used. That will train the model better giving better accuracies and more savings.
- More rules can be added in the Rule based learning based on our understanding of the industry, also incorporating the offer periods given by the airlines.
- Developing a more user friendly interface for various routes giving more flexibility to the users.

CONCLUSION

- The trend of flight prices vary over various months and across the holiday
- The airfare varies depending on the time of departure, making timeslot used in analysis an important parameter.
- Airfare varies according to the day of the week of travel. It is higher for weekends and Monday and slightly lower for the other days.



Thank You.