Journal of Network Communications and Emerging Technologies (JNCET) www.jncet.org
Volume 5, Special Issue 2, December (2015)

# Instigating Random Forests for Prefiguring Air Fares under a Constrained Environment

Anusha Vijay Kumar, Hamsashree Reddy R, Niranjan M Bhat, Manikantha

**Abstract –This paper presents comprehensive understanding of how the flight prices fluctuate depending on various factors taken into consideration over time using Random Forests. Random forests are ensemble methods for classification that operates by huge amount of decision trees that are been constructed during the training time and finally resulting the type of classes. This method coalesces bagging feature and the random selection of features (hence the name).Bagging refers to creating each classifier's training data set by randomly drawing samples with replacement from the given elements. Our model suggests whether the customer or user must buy the ticket right now or has to wait. The decision trees are constructed using C4.5 algorithm. According to our model an efficiency of 75% is achieved.**

**Keywords-Random forests, decision trees, C4.5 algorithm, Bagging, Classification**

This paper is presented at International Conference on Recent Trends in Computer and information Technology Research on 25th & 26th September (2015) conducted by B. S. Anangpuria Institute of Technology & Management, Village-Alampur, Ballabgarh-Sohna Road, Faridabad.

## 1. INTRODUCTION

Airlines make proper decisions on tickets using dynamic pricing, and finally rely on how to make decisions in regard to pricing on demand assessment models [3]. The purpose for such intricate prototypes is based on so many factors, some of which are stated below [2] -

- Competition en route
- Demand for seats
- Distance for the journey
- Seat availability
- Fuel prices
- Airline taxes and security fees
- Regularity of flights

Added to these, carbon emission quotas were also obligatory on airline adding cost to already burdened industry. One more factor that comes into play is the type of customer. In airline industry there are two types of passengers [5] - business travelers and leisure passengers.

Business travelers are totally comfortable with the airfare but not on dates. Leisure travelers aren't comfortable on high airfares (the cheaper the better) but are fine with any date. Airlines are trying their maximum to bring both of these passengers on the same page so that would be economically successful

Airlines estimate something called as a load factor [6]; this basically means the percentage of seats sold on a flight. They expect this number to be as high as possible. Airlines tend to manage this load factor by continuously varying the cost of the flight in order to fit in the sales of the seats and get utmost returns. So by this we could conclude that the airlines work with the ideology i.e. if the load factor is low and the demand is low, the airlines will increase the availability of cheap fare, whereas if the load factor is high and the demand is high, the airlines will automatically raise the rates on the flight.

The key intent of this project is to understand how the airline ticket expenses fluctuates over time, considering certain features and warn the customer on correct time to acquire the ticket.

We have chosen Random Forests [7] after profoundly analyzing its merits and consequences. It is effectual way to classify our dataset, thereby attaining accurate results. Rest of the paper is organized as follows: Section 2 emphasizes the related work and literature analysis in the field. Section 3 throws light on proposed approach. Section 4 discusses about the results obtained by applying proposed approach and followed by conclusion and future enhancement in section 5.

## 2. RELATED WORK

There has been some foremost works on structuring prognostic model for air charges using machine practices. These works have been accomplished using various algorithms and their respective accuracies have been dogged. There are also a range of software's that foresee the air fares over time like Bing travel (not available to general public), Sky scanner and so on.

Sub topic should be written as given below:

- To Buy or Not to Buy: Mining Airfare Data to Minimize Ticket Purchase Price - Oren Etzioni, Craig A. Knoblock, Rattapoom Tuchinda, and Alexander Yates.
- An economic guide to ticket pricing in the entertainment industry - Pascal Courty.
- Predicting Airfare Prices - Manolis Papadakis.

In paper [4], authors report on a pilot study in the sphere of airline ticket outlays where they documented over 12,000

price observations over a 41 day interval. When trained on this data, Hamlet data mining algorithm spawned a predictive model that saved 341 pretend passengers $198,074 by recommending them when to buy and when to stall ticket purchases. Remarkably, an intuitive algorithm with comprehensive knowledge of impending costs could save at most $320,572 in replication, thus Hamlet's savings were 61.8% of optimal. The algorithm's savings of $198,074 denotes a standard reserves of 23.8% for the 341 passengers for whom savings are possible. Overall, Hamlet saved 4.4% of the ticket amount averaged over the all-inclusive set of 4,488 virtual commuters.

Authors of [3] deal with the study of the pricing practices that are been observed in the ticket markets. Another important goal of this paper is to know how well they have understood things about pricing practices. Section1 briefs about the basics regarding to the entertainment industries. The rest of the paper reviews the most crucial ideas of the literature. Section2 describes about the theoretical literature on ticket pricing which is broadly divided into three board subsections, first subsection inspects at the pricing of unique seats for the equivalent performance. The second subsection briefs on pricing of tickets under demand uncertainty.
The third, reviews the pricing of tickets when the producer offers several performances.

In [1], the authors have used two approaches: the first factor is that they deal with the study of the factors that influence the average ticket price or those that sway the price of the certain aircraft in certain days till the date of departure. This particular distinctions is been depicted in the definition of this model.
The data is been collected from different websites using a tool called KVS extraction tool. The historical ticket price they have taken from the customer travel websites and air travel diagnostic establishments, but can also be assembled physically over a period of time. The latter can be extricated from statistics advertised by certain air travel administrating bodies.

## 3. PROPOSED SYSTEM

Software architecture basically depicts the whole system and helps us in understanding how it works.

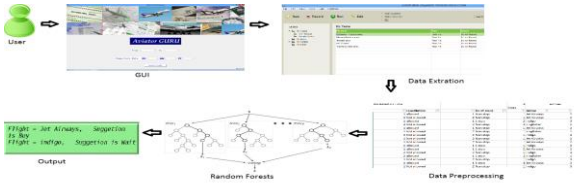


***Fig: 1** Structural design of the system.*

The above figure 1 gives the basic flow of the entire system. The user interface takes certain inputs (source, destination, date) and this is passed to the model. The model uses random forests classifier and produces the result as shown in the same figure.
The following steps are followed for the proposed system:

1. Data Collection

- The first step is that we carried out while developing the essential model was to "collect data". The data that is required by us was not accessible freely (as datasets), so we had to extract data.
  Data extraction was done using data extraction software that were freely available (we used - "automation anywhere" software). This can be done using any appropriate software. There are huge numbers of different sources in order to fetch the data of airfares on web, which in turn is used to train our proposed system.
  The data that is hauled out is done so, from well-known websites like cleartrip.com, goibibi.com, yatra.com and so on any website having suitable information can be chosen).
  The tool used to extract data, would just pick up the required values from the mentioned website and place it in an excel sheet.
- Now, this data cannot be directly used to train the model, it has to be pre-processed first.

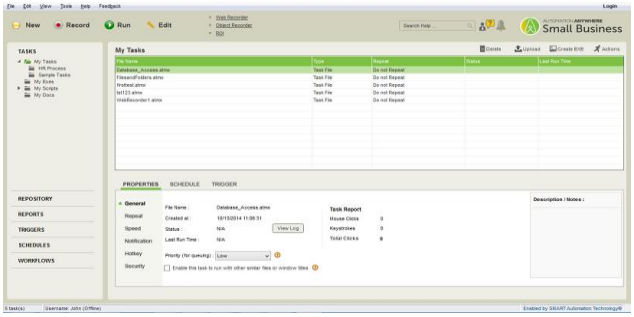


***Fig: 2** Screenshot of the data extraction tool used*

2. Preprocessing of data

The data which is now stored in excel must be pre-processed before utilizing it for training the model. This step was done manually.
We ordered the excel sheet so as to contain the attributes in a particular order, with each attribute having its maximum and minimum valued specified at the top. We assigned code values for all the values that an attribute can take in the tree, and were mentioned at the top of the excel sheet.
Lastly, we add the attribute "class label" which can take only two values- BUY or WAIT. This was decided based on the price and airline attributes.

By the end of this process, we get organized, processed data that is ready to be used to train the model.

## 3. Proposed Model

Once the data cleaned and formatted, it is used to build decision trees. The trees are built using C4.5 algorithm, coded in Python. As stated earlier Random forests build series of decision trees, where each tree predicts a classification. The Random forests predict the classification predicted by most trees. The number of trees that must be built to achieve high performance is the major concern.

It is observed that more the number of trees we use, the better the results get. But, the improvement decreases as the trees increases, i.e. at a certain time the prediction performance from learning more trees will be lower than cost of computation time for learning these additional trees. In our model, we have built 4 decision trees to obtain the results. The attributes for each of the tree is randomly chosen.

Once these trees are built, we perform a process called "voting", in order to find the final result. Here, the votes for each class label are calculated, and whichever gets the maximum votes will be treated as the final result obtained from the Random forest. The pseudo code for Random forests is shown below [7].

**Algorithm 2** Random forest

1 For b=1 to B; **do**

(a) Draw a bootstrap sample Z* of size N from the training data.

(b) Grow the random forest tree Tb to the bootstrapped data, by recursively repeating the following steps for each terminal node of the tree, until the minimum node size is reached

(i) Select m variables at random from p variables.

(ii) Pick the best variable split among the m.

(iii) Split the node into two daughter nodes.

**End for**

2 Output obtained is a collection of trees.

```
def Tree(ColList,SortGainListInput):
    global GainX1,GainX2,GainX3
    print "Tree-----------------------------------"
    AttributeCol1 = ColList[0]    # departure time columnn
    GainX1 = GainCal(rating, ClassCol,AttributeCol1)
    AttributeCol2 = ColList[1]    # Stops columnn
    ##TestClass = Test_Worksheet.cell_value(j,ClassCol)
    GainX2 = GainCal(rating, ClassCol,AttributeCol2)
    AttributeCol3 = ColList[2]    # Airline columnn
    GainX3 = GainCal(rating, ClassCol,AttributeCol3)
    SortGain = [[GainX1,AttributeCol1],[GainX2,AttributeCol2],[GainX3,AttributeCol3]]
    SortGain.sort(reverse=True)
    SortGainList = SortGainListInput
    EqualOnes ,Icount,Buy ,Wait, UpToYou =0, 0 ,0,0,0
    ##          print "SortGainList = ", SortGainList
    for TestRow in range (HeadderRows,Test_Nrows):
        TrainDataList = []

        for non, AttributeCol in SortGain:
            Attri1 = Test_Worksheet.cell_value(TestRow,AttributeCol)
            TrainDataList.append(Attri1)
##          print "TrainDataList = ",TrainDataList
        ans = comp(SortGainList, TrainDataList)
##      print "ans = ", ans
        if ans == "equal":
            EqualOnes += 1
##          print "TrainDataList = ",TrainDataList
            Temp = Test_Worksheet.cell_value(TestRow,ClassCol)
            if Temp == 1:
                Buy += 1
            elif Temp == 2:
                Wait += 1
            elif Temp == 3:
                UpToYou += 1
```

***Fig: 3*** *shows code snippet for creating trees using c4.5 algorithm*

## 4. RESULTS AND DISCUSSIONS

We have performed two types of testing for the system, which are-

Unit testing: A small testable part called unit of the application like functions, procedures were tested to determine if they were fit to use.

Integration Testing: After all the modules have been individually tested, we carried out integration testing to see whether they work in sync with each other and produce results as expected. Here we tested for interaction between the modules and interfaces between the components.

The following figures show the snap shots of the test cases that were used to test the model. It also shows the resulting output of the model and the actual outputs.

Fig: 3 shows the test cases for back end of the built model. As can be seen from among the 8 instances used, our model correctly predicted 6 instances, which gives an overall efficiency of around 75%.

| TEST CASE NO | SOURCE | DESTINATION | DATE | FLIGHT | EXPECTED O/P | ACTUAL OP | STATUS |
|---|---|---|---|---|---|---|---|
| 1 | Delhi | Benglore | 28-4-2015 | Jet airways | Wait | Wait | Pass |
| 2 | Delhi | Banglore | 28-4-2015 | Indigo | Wait | Wait | pass |
| 3 | Channai | Delhi | 22-4-2015 | Kingfisher | Buy | Wait | fail |
| 4 | Channai | Delhi | 22-4-2015 | Indigo | Wait | Wait | Pass |
| 5 | Delhi | Mumbai | 26-4-2015 | Kingfisher | Wait | Wait | Pass |
| 6 | Delhi | Mumbai | 26-4-2015 | Indigo | Wait | Wait | pass |
| 7 | Benglore | Mumbai | 27-7-2015 | Kingfisher | Wait | Buy | Fail |
| 8 | Benglore | Mumbai | 27-7-2015 | Indigo | Wait | Wait | pass |

***Fig: 4*** *shows the test cases for back end of the system, which checks if the input is adequately obtained from the user.*

| Test case no | Current Date | Date of Boarding | Expected Output | Actual Output | Status |
|---|---|---|---|---|---|
| 1 | 28-4-2015 | 5-5-2015 | Display Result | Display Result | Pass |
| 2 | 28-4-2015 | 15-4-2015 | Display Error | Display Error | Pass |
| 3 | 28-4-2015 | 27-4-2015 | Display Error | Display Error | Pass |
| 4 | 28-4-2015 | 29-4-2015 | Display Result | Display Result | Pass |
| 5 | 10-5-2015 | 5-5-2015 | Display Error | Display Error | Pass |

***Fig: 5*** *shows the test cases for the front end of the system.*

## 5. CONCLUSION AND FUTURE ENHANCEMENTS

This application is aimed at analyzing the variation in ticket pricing using Random Forests, which has proven to be very effective.

This can be further augmented to function for international flights and more locations can be added for customers to choose for various journeys.

The work can be extend to the real world utilization since this particular model is a research based model and it can be extended as an application where in the customer can use our particular model in order to make direct booking once the air fare decision is made and provide a secure interface for the customer about the booking.

The model can be enhanced for the customer to make decisions not only in the economic class but also in business classes and so on.We can also ask customer for certain conditions based on which we can carry out the classification process.

## ACKNOWLEDGMENT

We would like to thank Dr. Arti Arya, Professor and Head MCA, PESIT-Bangalore South Campus for her consistent support, encouragement and guidance towards the project, of which this paper is the outcome.

## REFERENCES

[1] Oren Etzioni, Craig A. Knoblock, Rattapoom Tuchinda, and Alexander Yates, "To Buy or Not to Buy: Mining Airfare Data to Minimize Ticket Purchase Price", SIGKDD 2003.
[2] Pascal Courty, "An economic guide to ticket pricing in the entertainment industry", Recherches Économiques de Louvain – Louvain Economic Review 66(1), 2000.\
[3] Manolis Papadakis,"Predicting Airfare Prices".
[4] R. Preston McAfee and Vera te Velde,"Dynamic Pricing in Airline Industry"
[5] Leo Breiman,"Random Forests", January 2001
[6] Bryan Matthews_ Santanu Dasy Kanishka Bhadur et al.," Discovering Anomalous Aviation Safety Events using Scalable Data Mining Algorithms".
[7] Vrushali Y Kulkarni and Dr Pradeep K Sinha," Random Forest Classifiers: A Survey and Future Research Directions", International Journal of Advanced Computing, ISSN: 2051-0845, Vol. 36, Issue. 1.
[8] Manish Mehta, Rakesh Agrawal and Jorma Rissanen, "SLIQ: A Fast Scalable Classifier for Data Mining".
[9] Rick Seaney. (2011, January 7). Understanding Airline Ticket Prices: Why Your Seatmate's Airfare Cost More (or Less) More Than Yours [Online]
[10] Available:http://www.farecompare.com/travel-advice/understanding-airline-ticket-prices-why-your-seatmates-airfare-cost-more-or-less-than-yours/
[11] Matt Kepnes. Why Plane Tickets Cost So Much (and How You Can Still Get a Deal) [Online]. Available: http://lifehacker.com/why-plane-tickets-cost-so-much-and-how-you-can-still-g-485767079.
[12] NomadicMatt. 2013, April 25). Why Your Airplane Ticket is so Expensive Nowadays [Online]. Available: http://www.nomadicmatt.com/travel-blogs/expensive-airfare/.
[13] Nando De Freitas. (2013, February 14). "Machine Learning- Random Forests" [Online]. Available: https://www.youtube.com/watch?v=3kYujfDgmNk