A Pricing Mechanism Using Social Media and Web Data to Infer Dynamic Consumer Valuations

Mabin Mariam Joseph
Computer Science Department
Illinois Institute of Technology,Chicago
USA
mjoseph4@hawk.iit.edu

CS595 – Economic and Privacy Issues in Big Data

Project Report

***Abstract*—This document provides a project report on the Dynamic Pricing Model implementation done as part of CS595 Project Requirement.**

Index words – Sentimental Analysis, Twitter Sentiments, Dynamic Pricing

I. Introduction

The project explained in this document is part of the requirement for CS595(Economic and Privacy Issues in Big Data). The project focuses on the economic aspect related to Big Data and is based on [1]. The project implements the proposed Dynamic Pricing Mechanism Model that takes into account sentiment analysis as a major factor. The project goes on to show that there is a marked increase in the revenue generation using the new dynamic pricing model when compared to the existing dynamic pricing model.

In today's world we can see tides of sentiments being poured out on social media on any topic you can think of. Companies from industries such as Hospitality, Retail, Advertisement etc. classify, aggregate and use these sentiments to predict sentimental analysis that can be used for their revenue growth. By tracking the changes in sentiments towards a particular product, companies that have dynamic pricing mechanisms that take advantage of the fluctuations in customer's interest and demand.

A. *IMPORTANT TERM DEFINITIONS*

<u>Dynamic Pricing</u>

A pricing strategy in which businesses set flexible prices for products or services based on current market demands and other external factors such as competitor prices, demand and supply etc.

<u>Sentimental Analysis</u>

Also known as Opinion Mining, it refers to the use of natural language processing, text analysis and computational linguistics to identify and extract subjective information in source materials.

<u>Revenue Management</u>

Application of disciplined analytic that optimizes product availability and price to maximize revenue growth. In simple words it is to sell the right product to the right customer at the right time for the right price.

II. Project design

The project's design consists of majorly three steps – Data extraction from Twitter which makes use of Twitter's Public Stream API, Sentimental Analysis in which the data obtained from Twitter is analyzed to extract two time series that shows the aggregate sentiments of the user towards a product. This time series is then provided as an input to the Dynamic Pricing Model mechanism to forecast prices over a future selling period and to show that when the new pricing model is used there is a marked increase in the revenue generated.

A. *Data Extraction from Twitter*

In order to collect live data from Twitter, the Streaming API was used. The Streaming API gives developers low latency access to Twitter's global stream of Tweet data. A proper implementation of a streaming client will be pushed messages indicating Tweets and other events have occurred, without any of the overhead associated with polling a REST end point.

The POST statuses/filter end point was used here. This returns statuses that match one or more filter predicates.

URL : https://stream.twitter.com/1.1/statuses/filter.json

The below steps are involved in the data extraction process implemented in the project:

1) Twitter Application Creation :

An application named "MabinSentiments" was created on https://apps.twitter.com to get the API keys required for live streaming of data. Once this was created, the Consumer key, Consumer secret, Access Token and Access Token Secret were obtained from the OAuth tab.

2) Java project Creation for Data Extraction :

The Twitter Stream API allows one to connect to Twitter's global stream of tweet data, allowing messages to be pushed to us instead of pulling them. To use the Twitter Stream, a persistent connection to the server has to be maintained. In order to achieve this, a Java project was created using Maven as the build tool. To get the OAuth library, the Oauth Java library Scribe was added to the pom.xml file as shown below:

```
<dependency>
  <groupId>com.github.scribejava</groupId>
  <artifactId>scribejava-apis</artifactId>
  <version>2.3.0</version>
</dependency>
```

Once the OAuth library was set up, it was used along with OAuth keys obtained from Step 1 to connect to Twitter's public stream. The program configures these OAuth details, creates a request object, set certain Twitter headers, sends the request and creates a buffered stream reader that reads the response stream and then writes to the file of choice. This streaming of data was done for approximately 10 days(close to 30,000 tweets) extracting tweets with the keyword “electric”.

B. Sentimental Analysis

The data set obtained from the Data Extraction step was analyzed to build a time series of aggregate customer sentiments towards a particular topic, in our case “electric”.

The raw data needs to undergo pre-processing steps in order to be useful in performing Sentimental Analysis. For the pre-processing, each JSON object or tweet was parsed to obtain the relevant information.

Twitter data is stored in JSON format, a lightweight technology used for data marshaling. For processing the JSON data, the Java library JSON Simple was used. JSON Simple easily allows parsing of JSON objects with great ease. The JSON object received from Twitter is based on the standard JSON format for Twitter can be referred to from the below link:

https://dev.twitter.com/overview/api/tweets

Key parts to parse from the JSON object here are :

“created at” : String which ensures we keep a track of the period of tweet.

“text” : String which is essentially the crux of the matter. This includes the hashtag portion of the tweet as well.

“lang” : String to ensure we get only English tweets, since the lexicon used for sentimental analysis is based on English words.

Code was then written for graceful parsing of data for the purpose of sentimental analysis.

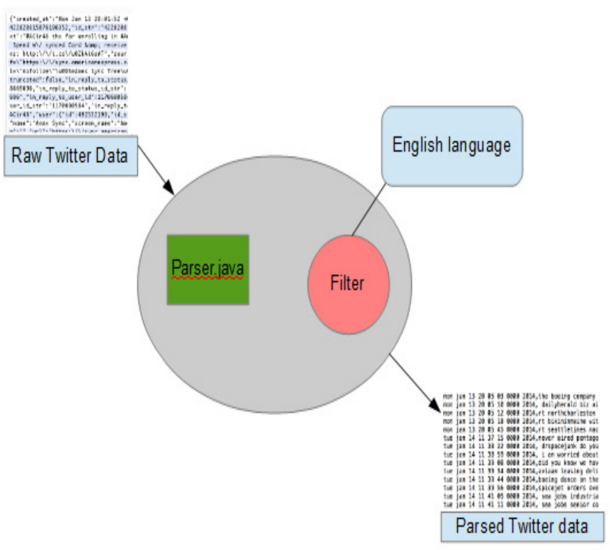

Figure 1 : Parsing of Twitter Data

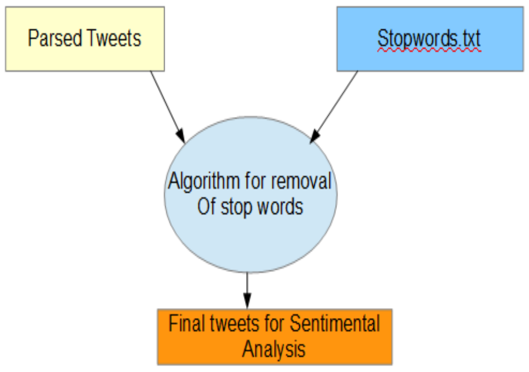
Parsed Tweets
Stopwords.txt
Algorithm for removal Of stop words
Final tweets for Sentimental Analysis

Figure 2 : Stop word removal from parsed tweets

Tweet before parsing:

{"created_at":"Thu Apr 07 17:20:58 +0000 2016","id":718126432535363584,"id_str":"718126432535363584","text":"RT @AwardsDarwin: Why you shouldn't make an electric guitar at home. https:\/\/t.co\/e5byCAQ3RU","source":"\u003ca href=\"http:\/\/twitter.com\/#!\/download\/ipad\"rel=\"nofollow\"\u003eTwitter for iPad\u003c\/a\u003e","truncated":false,"in_reply_to_status_id":null,"in_reply_to_status_id_str":null,"in_reply_to_user_id":null,"in_reply_to_user_id_str":null,"in_reply_to_screen_name":n ull,"user"{"id":177258828,"id_str":"177258828","name":"benja","screen_name":"zaino83","location":"Argentina

","url":null,"description":"Lic en relaciones internacionales, deportista muy amateur. Politicamentehuerfano","protected":false,"verified":false,"followers_count":343,"friends_count":1041,"listed_count":8,"favourites_count":7843,"statuses_councations":null},"geo":null,"coordinates":null,"place":null,"contributors":null,"is_quote_status":false,"retweet_count":0,"favorite_count":0,"entities"{"hashtags"[],"urls"[{"url":"https:\/\/t.co\/C4Vlmjxu1Z","expanded_url":"http:\/\/ift.tt\/1UL7o0g","display_url":"ift.tt\/1UL7o0g","indies":[87,110]}],"user_mentions":
[],"symbols"[]},"favorited":false,"retweeted":false,"possibly_sensitive":false,"filter_level":"low","lang":"en","timestamp_ms":"1460049654367"}

Tweet after parsing:

Thu Apr 07 17:20:58 +0000 2016,RT @AwardsDarwin: Why you shouldn't make an electric guitar at home. https://t.co/e5byCAQ3RU

Once the data has been successfully pre processed as shown above, the next step is to remove stop words from the body of the text. Stop words are words like as,at,be etc. which add unnecessary weight to the text. An appropriate stop words dictionary will accompany the sentiment algorithm to leave only important words for polarity detection. A popular stopwords.txt was obtained from Google code repository was utilized to filter the pre processed data set.

In [2], Bing Liu has made available a 6800 word sentiment word data set without polarity scores. Sentiment analysis for the pre processed twitter data set was done using this Sentiment lexicon. The Sentimental Analysis Algorithm is as below:

1) Load the positive and negative words into two different array lists.
2) Tokenize each word in the tweet.
3) For each word in the tweet, if present in the positive word lexicon, the polarity score is incremented.
4) If the word is present in the negative word lexicon, the polarity score is decremented.
5) If not found, "no match found" is printed in the output.
6) The output of the sentimental analysis program will contain Date + Text + Associated polarity

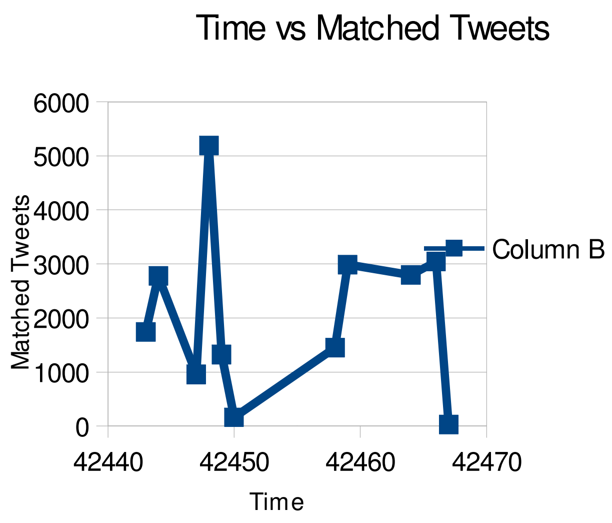

Figure 1. Time vs. Matched tweets

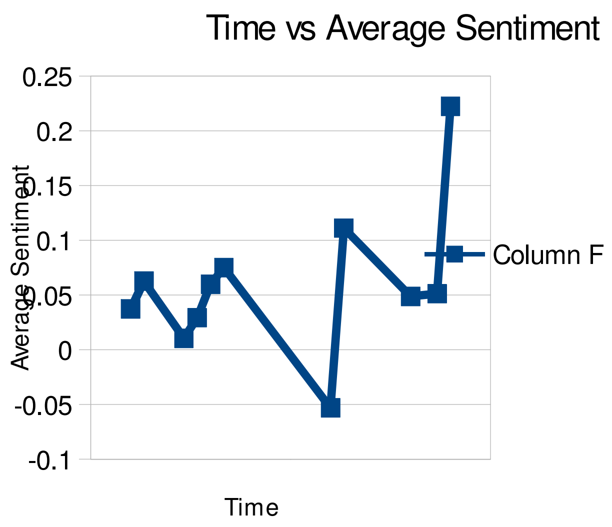

Figure 2. Time vs. Average Sentiments

The above two time series are extracted from the output of sentimental analysis. The data from these time series forms the input for the Dynamic Pricing Algorithm to follow.

Let Mt be the matched tweets for a day 't'. The average sentiment score st is given by:

$$s_t = s(M_t) = \sum_{m \in M_t} s(m)/q_t ,$$

where s(m) is the individual sentiment score and $q_t = |M_t|$.

C. Dynamic Pricing Mechanism

For the dynamic pricing mechanism, we assume that the customer's behavior follows a variant of the multinomial model, extended to use time varying arrival rate and valuations.

The ultimate aim of the proposed dynamic pricing model is to compare it to the existing dynamic pricing model and show that incorporating sentiments of a consumer to the algorithm for dynamic pricing has a higher percentage of revenue generated.

Let J(n,t) be the price generated for a set of products 'n' present in the inventory at time 't'. P is the set of permissible prices. The probability that a customer arrives is given by λ_t. The price is generated using a recursive formulation.

$$J^*(n, t) = \max(p_t \in P) \{\lambda_t P(p_t)(p + J^*(n - 1, t + 1)) + \lambda_t(1 - P(p_t))J^*(n, t + 1) + (1 - \lambda_t) J^*(n, t + 1)\}, \quad (1)$$

where J(0,t) = 0 and J(n,T) = 0 where T is the maximum time set.

The customer valuation λ_t and arrival rate a_t are informed by the sentiment dynamics. Given a baseline valuation a_{base}, the dynamic valuation a_t at time 't' is estimated by skewing the baseline by a function of the sentiment score s_t.

$$a_t = a_{\text{base}} + c_v \cdot (\sigma(s_t - s_{\text{base}}) - 0.5), \quad (2)$$

where $\sigma(\cdot)$ is the sigmoid function[1] for a constant cv ≥ 0. st is the sentiment score and sbase is the average sentiment score obtained from the Sentimental Analysis step.

Given a baseline arrival rate λ_{base}, the dynamic arrival rate λ_t at time 't' is estimated by skewing the baseline by a function of the volume of tweet v_t,

$$\lambda_t = \lambda_{\text{base}} + c_a \cdot (\sigma(v_t - v_{\text{base}}) - 0.5), \quad (3)$$

where $c_a \geq 0$ such that $1 \geq \lambda_t \geq 0$ and v_{base} is the average volume of tweets.

In Equations (2) and (3), we note that when c_v and c_a have a value of 0, it deduces to the existing dynamic pricing model.

The buying probability $P(p_t)$ or the probability that a customer that arrives at a time 't' will buy the product is given by the equation

$$P(p_t) = (e^{\kappa t/\mu} + 1)^{-1} \quad (4)$$

where $k_t = u_t - a_t + p$, with u_t being the marginal utility, a_t is the dynamic valuation obtained from (2) and p is the intrinsic price of the product. In the same line of thought $(1-P(p_t))$ is the probability that the customer does not buy the product at time 't' for the price 'p'.

Putting (4) and (3) into (1), the revenue generated is calculated.

[1]The sigmoid function is defined $\sigma(x) = 1/(1 + e^{-x})$.

The dynamic pricing was calculated using the new model as well as the old model and the two were compared to show that the new dynamic pricing model generates a higher revenue than the existing dynamic pricing model.

III. TECHNICAL SPECIFICATIONS

For Data Extraction : Twitter Stream API implemented using the Scribe Java library

For Sentimental Analysis : Parsing of raw twitter data implemented using the Simple JSON library.

IV. FILE DESCRIPTION AND DATA USED

This section gives a detailed description of the different files using in the project implementation.

File A : Project_Twitter_Data.json

This is the file into which Twitter's live stream of data is copied into. It contains the information in the raw form. The data used for this implementation contains around 30,000 tweets based on the keyword “electric”.

File B : Mabin_data_final.txt

This file is the output of the parsing program which breaks down the raw data into the form Date + Text. The filtering is done based on the “lang = en” tag to ensure only English tweets are parsed and written into this file.

File C : Stop_words.txt

This file is taken from [7] and is used in the removal of stop words from File B.

File D : No_Stop_Words_Tweet.txt

Using File B as the input, all the stop words in the tweets are removed based on File C and the final output is present in this file. This is the final file used as input for Sentimental Analysis.

File E : positive-words.txt

This file is the set of positive words provided in [1] without the polarity scores.

File F : negative-words.txt

This file is the set of negative words provided in [1] without the polarity scores.

File G : Polarity_final.txt

Using Files E and F, the polarity of each tweet is calculated and written into this file in the format Date + Tweet + Polarity score.

V. IMPLEMENTATION DETAILS

This section describes the project's code implementation. Note that the code explained here has been attached along with the project report as part of the final project submission.\

A. Data Extraction Code Implementation

(a)TwitterStreamServer.java :Starts the TwitterStreamConsumer thread for extracting data from Twitter's public stream.

(b) TwitterStreamConsumer.java : Imports the Java OAuth library by defining it as a dependency in the pom.xml file. Using this library, the Oauth consumer key, consumer secret, access token and access token secret are configured. A new OAuth request is created to the Stream URL https://stream.twitter.com/1.1/statuses/filter.json. The keywords to be considered when extracting the tweet is done using the addBodyParameter method of the OAuth request object. Using the authentication keys mentioned above, our application is now connected to the Twitter's server and reads the responses using a BufferedReader object after which the read data stream is written into File A.

(c) Parser.java : The JSON Simple Java library is used for the parsing function. File A is read and then filtered based on “lang = en”. This is to ensure that only English tweets are considered for the Sentimental Analysis. Each stream is taken as a JSON object and parsed to obtain the “created_at” and “text” fields only from the JSON object. Hash tags are not taken into consideration here since it is a part of the “text” field itself. The data is parsed and written into File B in the form Date + Text.

(d) sentimentalAnalysis.java : This program takes File C and File B as inputs. It tokenizes each word in the text field of the tweet and checks to see if the word is present in File C. If so, that word is removed from the text field. The final output of this program is File D.

(e) Sentiment.java : This program takes File D, File E and File F as inputs. The polarity score is initialized to 0 and is reset to 0 after evaluating every tweet. The set of positive and negative words are first structured into array lists. The data to be evaluated is tokenized and the text token is checked word by word and compared against the words present in Files E anf F. For every positive word present, the polarity score is incremented and for every negative word the score is decremented. The final output of this program is File G which is of the format Date + Tweet + Polarity.

(f) DynamicPricing.java : Implements the methods required to calculate revenue generated as shown in Equation (1).

The method details are as below:

1) Sigmoid Calculator : Takes in difference between the base value and value at time 't' for customer valuation 'a' and arrival rate 'λ'. It returns the result of the sigmoid function for the passed in value.

2) KtCalculator : Takes in time 't', sentiment score 'st' and average sentiment score 'sbase' as input. The result is calculated as MarginalUtility – DynamicValuation at time 't' + price at time 't'.

3) DynamicValuationCalculator : Takes in time 't', sentiment score 'st' and average sentiment score 'sbase' as input. Keeping abase as 0.2 and cv as 0.13 the result is calculated based on Equation (2) and returned.

4) DynamicArrivalRateCalculator : Takes in time 't', volume of tweet 'vt' and average volume of tweet 'vbase' as input. Keeping lambdabase as 0.2 and ca as 0.135 the result is calculated based on Equation (3) and returned.

5) BuyingProbability : Takes in time 't', sentiment score 'st' and average sentiment score 'sbase' as input. Based on Equation (4), the buying probability is calculated and returned.

(g) TestNewDynamicPricing.java : The time, average sentiments, individual sentiments, individual volume of tweets, average volume of tweets, maximum Time for which revenue is to be calculated, number of items in the inventory and the price of the product is given as the input in arrays. The program calculates the Dynamic Customer valuation and Dynamic arrival rates using the inputs given. This is then used to calculate the Revenue generated for n = 10 for a MaxTime of 10. The generated output is plotted to obtain the graphs shown under the Results section.

(h) DynamicPricingOld.java : In this program, the methods defined are the same as DynamicPricing.java except that the values of c_a and c_v are kept as 0 for the DynamicValuationCalculator and DynamicArrivalRateCalculator methods. The abase is given the average sentiment score value sbase. The lamdabase is given the average volume of tweets vbase.

(i) TestOldDynamicPricing.java : This program calcualtes the revenue generated when using the old dynamic pricing model calculations used in DynamicPricingOld.java. The generated output is used in the determination of price comparison between the two models and highlighted in the Results section of the report.

VI. RESULTS

The dynamic pricing mechanism was simulated using data from Twitter (period between March – April 2016). The tweets were filtered based on the keyword 'electric' and the language tag 'eng'. Then using the scoring mechanism described in the above sections, time series were generated that includes the volume of the tweets and the average sentiments of the matched tweets.

In the experiments, s_{base} was set to equal the average sentiment score observed in the time series. Likewise, for the influence of tweet volume on the arrival rate, v_{base} was set to average daily volume of tweets over the course of the time series. The resultant time series is shown in Figure 3a and 3b.

The dynamic valuation and arrival rates were fed into the pricing mechanism and the revenue generated using these inputs. Then to compare our pricing mechanism, we also experimented on a model which takes valuations and arrival rates as fixed from the data extracted from the time series obtained in the data extraction step.

Keeping an inventory of 10 items and for a maximum time of 10 days, the revenue generated was calculated. The results as seen in Figure 4 show that the new dynamic pricing mechanism implemented through this project yields a higher revenue when compared to the static mechanism when the valuation and arrival rates are kept fixed

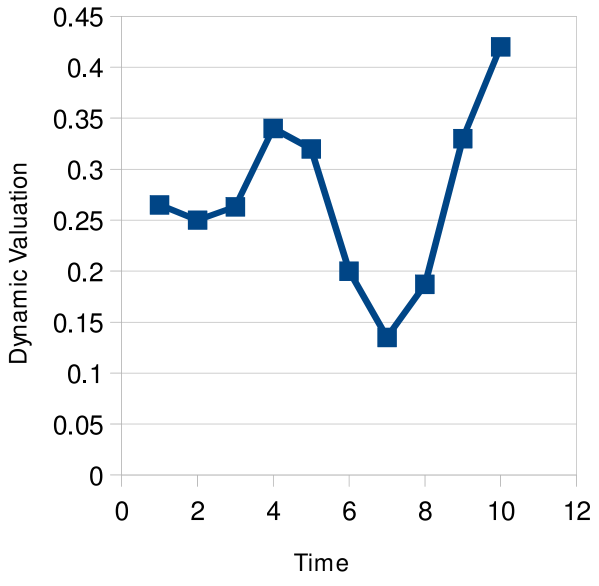

Figure 3a : Dynamic customer valuation time series

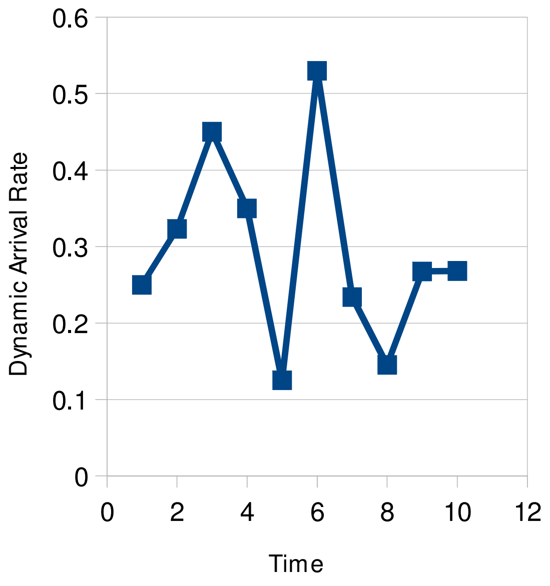

Figure 3b : Dynamic Arrival Rate time series

VII. CONCLUSION

Online social networks like Twitter play a major role in the diffusion of information by increasing the speed of novel information and diverse viewpoints. Given the impact of such information and its role in helping industries understand what people think, it is only wise if we utilize and harvest this information for industry gain.

From the project implementation and on reading several papers related to the topic of dynamic pricing, it is evident as to why many companies are turning to dynamic pricing as part of revenue management process. As per [3], many industries are viewing dynamic pricing as a major opportunity to contribute to business strategy.

Incorporating sentiments into determining the price of a product helps to give a real indication as to the demand of the product. More than the number of people who bought a product, it is important to note how the product was received by the people. And from this implementation, it has been proved that a pricing mechanism that uses sentiments to determine the dynamic price yields a higher revenue.

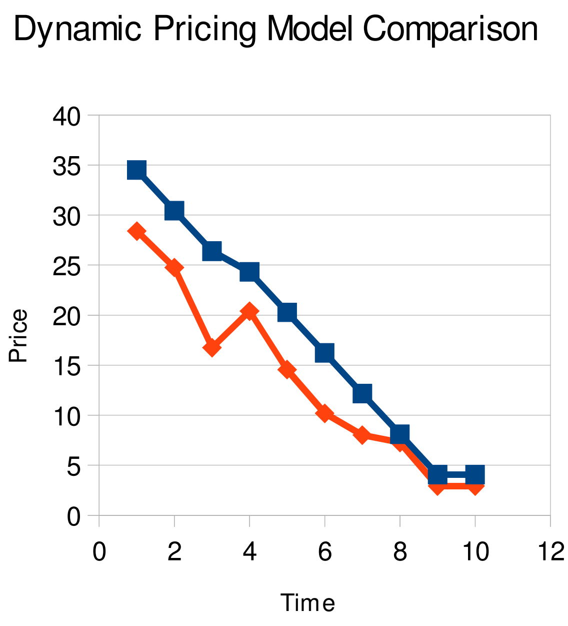

Figure 4 : Comparison of price generated using the implemented dynamic pricing mechanism model (Blue) and the pricing model that takes static arrival rate and valuation(Orange)

VIII. CHALLENGES AND LESSONS LEARNT

Throughout the course of the project implementation, several valuable lessons were learnt along with facing multiple challenges along the way. The main points in this regard is highlighted below.

A. Challenges

1) The project was implemented by a single member team, which required putting in more effort in terms of brain storming and understanding the project requirement aspects. The project was planned keeping in mind the deadline and the effort needed. Lot of time and research was put into understanding the concepts and coming up with ideas for the implementation.

2) Implementing a paper is much harder than reading and understanding it. A lot of research time had to be put it to understand why certain statements were written and how they are connected to the topic. Many papers related to dynamic pricing models are available and going through them enabled better understanding of the topic.

3) The area of sentimental analysis is an area of research in itself. Several algorithms are available, but since we are dealing with evaluating human sentiments from sentences, it is often very challenging to get the true nature of the sentiment. For example, statements that are sarcastic in nature will be interpreted as a positive statement. In today's world, along with words, sentiments are often expressed using emoticons and emoji's which further complicates this area of study.

Keeping in mind the complexity of sentimental analysis and understanding that the main reason behind the project is to understand dynamic pricing strategy, the algorithm for understanding the sentiments was kept simple – if there is a positive word, increment the score and if there is a negative word, decrement the score.

B. Lessons Learnt

1) In doing research for this project implementation, the value of the information available online was understood and how if it is harvested and analyzed, it can revolutionize the way the industries work today. People can get better products and industries can get higher revenue.

2) The data obtained from twitter contains information about the person's account and location from where the tweet occurred. This shows how much of personal information is present and available to any one who wants to use it. It is important to understand how this information can be protected.

IX. FURTHER AREA OF STUDY

1) Adding privacy protection to the data obtained from online sources :

2) There are several external and internal factors that can affect dynamic pricing apart from customer sentiments. Market demand, competitor prices, inventory availability, time of day, conversion rates, financial goals and even the weather are some of the elements that industries can incorporate into the dynamic pricing algorithm. The key to successful dynamic pricing is to know when to change and by how much to change.

3) Offering personalized dynamic pricing to customers. This is often referred to as channel price coordination which refers to a holistic set of capabilities that enable the design of pricing architectures within and across channels.

REFERENCES

[1] Samuel D. Johnson, Kang-Yu Ni, A Pricing Mechanism Using Social Media and Web Data to Infer Dynamic Consumer Valuations, 2015 IEEE International Conference on Big Data.
[2] B.Liu, Sentimental Analysis and Opinion Mining, ser. Synthesis Lectures on Human Language Technologies. Morgan & Claypool Publishers, 2012, no.16.
[3] IBM Commerce, Attracting and Retaining Customers with insights-driven dynamic pricing, January 2016
[4] A. Guille, H. Hacid, C. Favre, and D. A. Zighed, "Information diffusion in online social networks: A survey," SIGMOD Rec.,vol. 42, no. 2, pp. 17–28, Jul. 2013. [Online]. Available:http://doi.acm.org/10.1145/2503792.2503797
[5] B. Liu, M. Hu, and J. Cheng, "Opinion observer: Analyzing and comparing opinions on the web," in Proceedings of the 14th International Conference on World Wide Web, ser. WWW '05. New York, NY, USA: ACM, 2005, pp. 342–351.
[6] Gabriel Bitran, Rene Caldentey, An Overview of Pricing Models for Revenue Management, December 2002 ,[Online]. Available: http://dx.doi.org/10.1287/msom.5.3.203.16031
[7] "Stopwords.txt – Twitter-sentiment-analysis" – Twitter Sentiment Analysis Using Machine Learning Techniques – Google Project Hosting. Web. 26 Apr. 2014. <https://code.google.com/p/twitter-sentiment analysis/source/browse/trunk/files/stopwords.txt?r=51>.
[8] "Opinion Mining, Sentiment Analysis, Opinion Extraction." Opinion Mining, Sentiment Analysis, Opinion Extraction. Web. 27 Apr. 2014. <http://www.cs.uic.edu/~liub/FBS/sentiment-analysis.html#lexicon>.
[9] Json-simple – JSON.simple – A Simple Java Toolkit for JSON – Google Project Hosting." Json-simple – JSON.simple – A Simple Java Toolkit for JSON – Google Project Hosting. Web. 28 Apr. 2014. <https://code.google.com/p/json-simple/.>.