

# Design Document for the Analysis of the Online Retail Dataset

Contact Persons:

(TEAM 6)

Himaja Vadaga

Sandeep Kumar Bethi

Bryce Brako

## Version Control:

Date	Version	Description	Changed by	Reviewed by
08/05/2016	1.0	First Draft	Bryce Brako	Himaja Vadaga Sandeep Kumar Bethi

## 1. Preface

This document is created to describe the design, development and deployment procedures taken to perform the analysis of the Online Retail Dataset. It will illustrate the design of the algorithm, the web service and the user interface in the design document.

## 2. INTRODUCTION

We researched on the company E-Commerce Analytics Inc. and found that they are into E-Commerce Analytics. On receiving the opportunity of an interview the requirements to clear this interview is performing the analytics of any e-commerce dataset and presenting it to the company. Based on the expectations and to showcase our level of expertise in analytics we decided to perform the analytics on transactional data set for a UK-based and registered non-store online retail. The company mainly sells all-occasion gifts. Many Customers of the company are wholesalers.

## 3. About the Dataset

The dataset is a transactional dataset which contains all the transactions occurring between 01/12/2010 and 09/12/2011 for a UK-based and registered non-store online retail. It has a total of **541909 rows** and **8 columns** which are mentioned below.

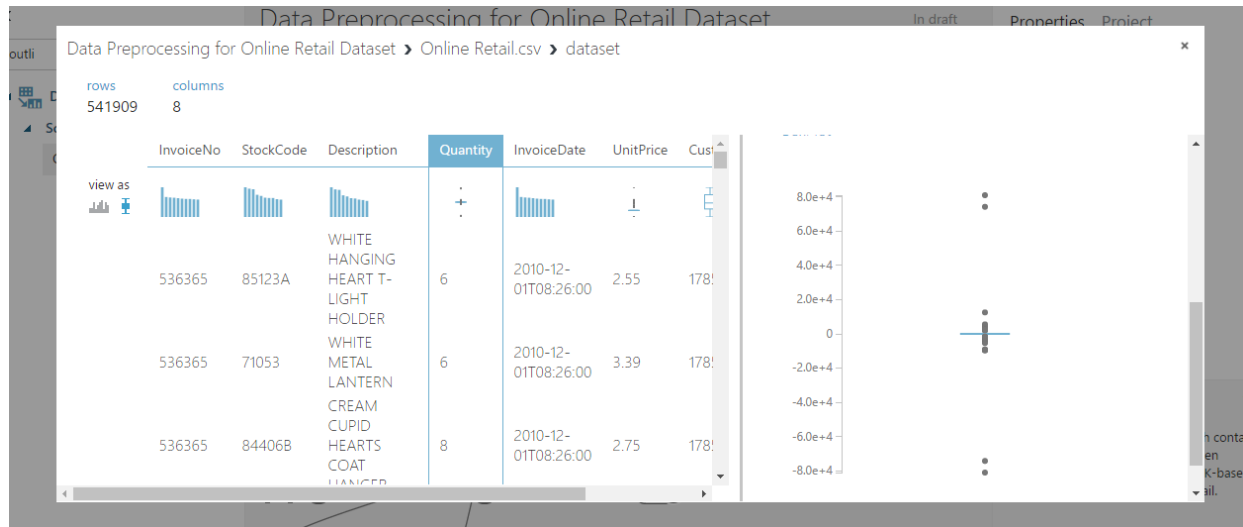
The following are the columns in the dataset and its description:

- **InvoiceNo**: Invoice number. Nominal, a 6-digit integral number uniquely assigned to each transaction. If this code starts with letter 'c', it indicates a cancellation.
- **StockCode**: Product (item) code. Nominal, a 5-digit integral number uniquely assigned to each distinct product.
- **Description**: Product (item) name. Nominal.
- **Quantity**: The quantities of each product (item) per transaction. Numeric.
- **InvoiceDate**: Invoice Date and time. Numeric, the day and time when each transaction was generated.
- **UnitPrice**: Unit price. Numeric, Product price per unit in sterling.
- **CustomerID**: Customer number. Nominal, a 5-digit integral number uniquely assigned to each customer.
- **Country**: Country name. Nominal, the name of the country where each customer resides.

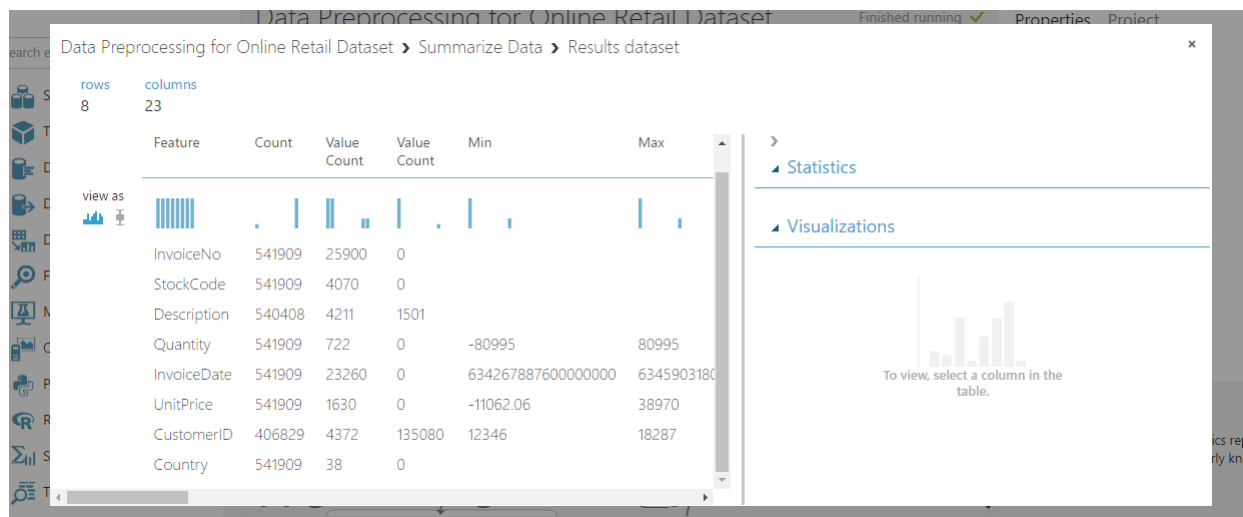
#### 4. Analysis of the Raw Data

- We analyzed the data using Microsoft Machine Learning Studio and it provided a summarized view of the data with respect to each column.

[NOTE: Since this is a free version the execution is done in a single node. Hence the performance and speed of the application would be low. But whereas if there is a standard version then the execution takes place on multiple nodes and performance is much higher.]



The above screenshot is from MLStudio that gave us information on the outliers present amongst the values of the integer columns.

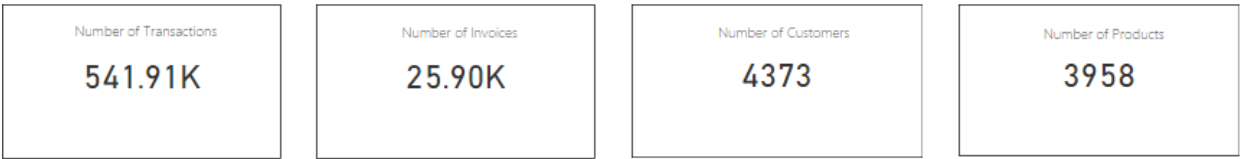


The above screenshot is from MLStudio that gave us information on the Unique Value Count, Missing Value Count and the Min-Max of the integer columns.

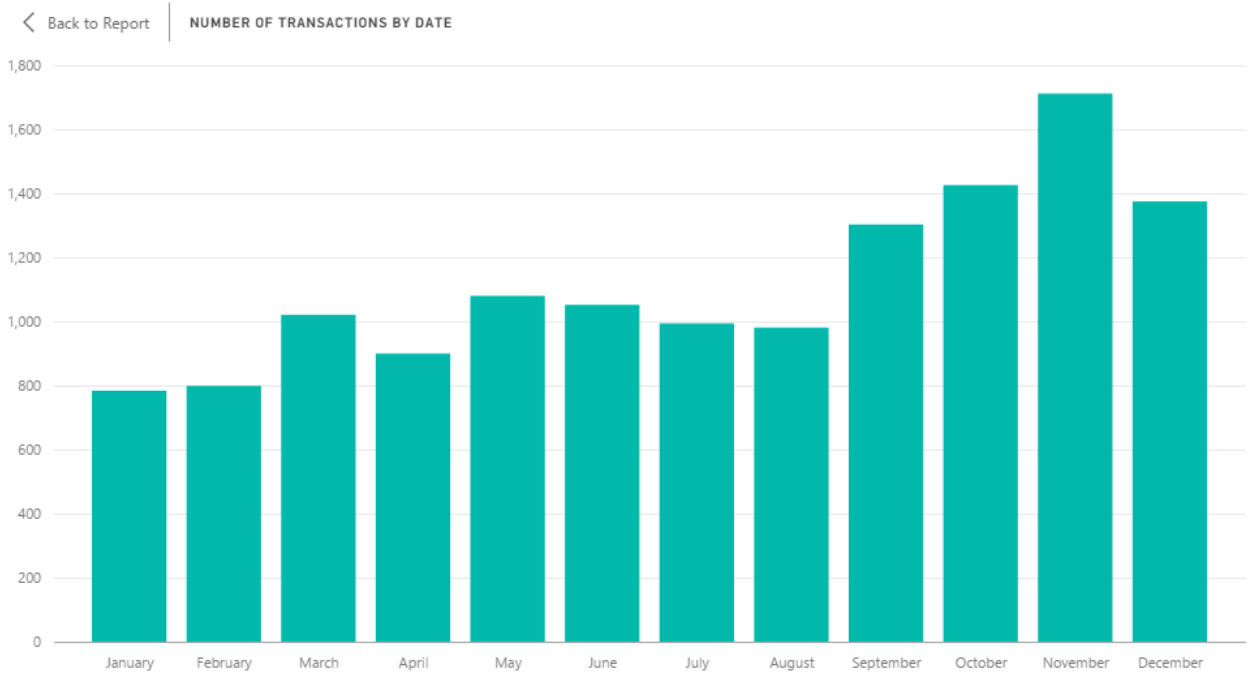
- We also used POWER BI to get a graphical view of the dataset.

The title explains what the graph describes.

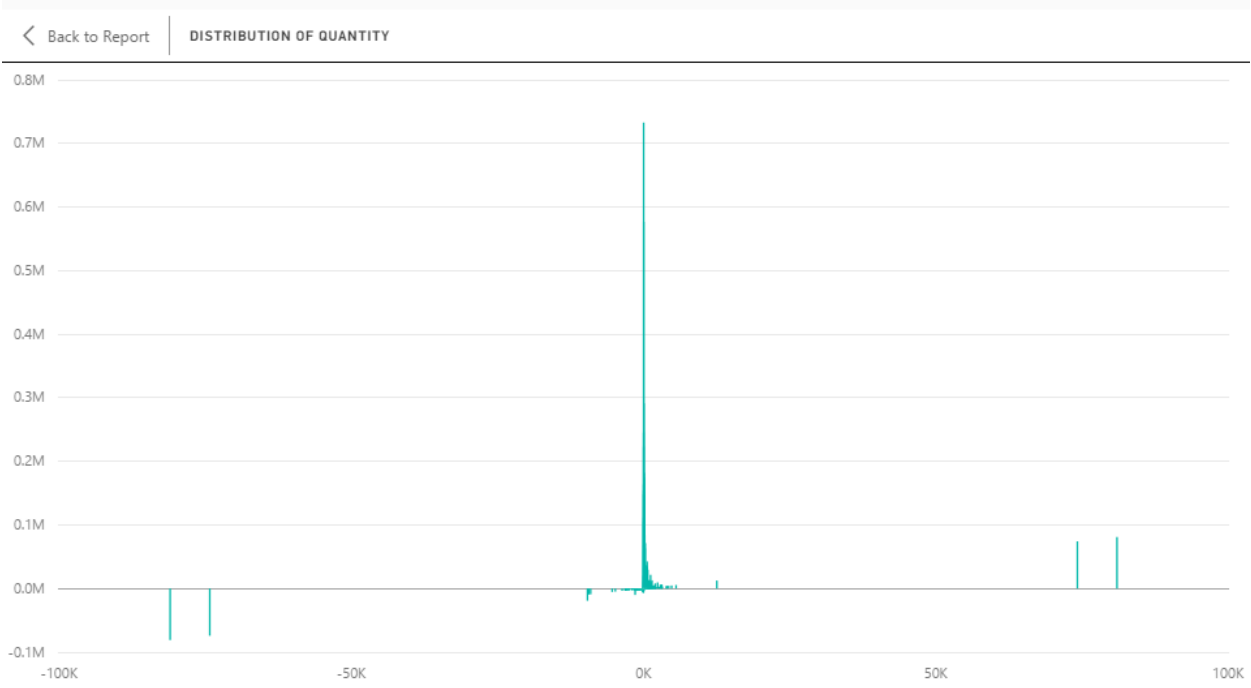
a.



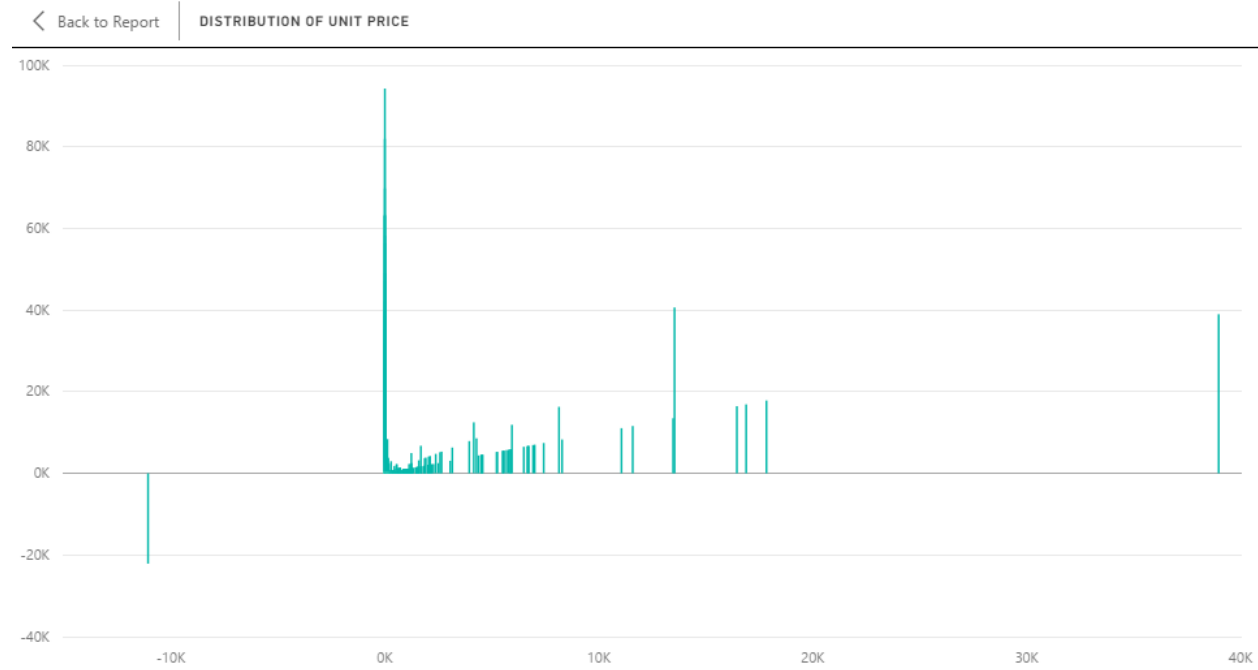
b.



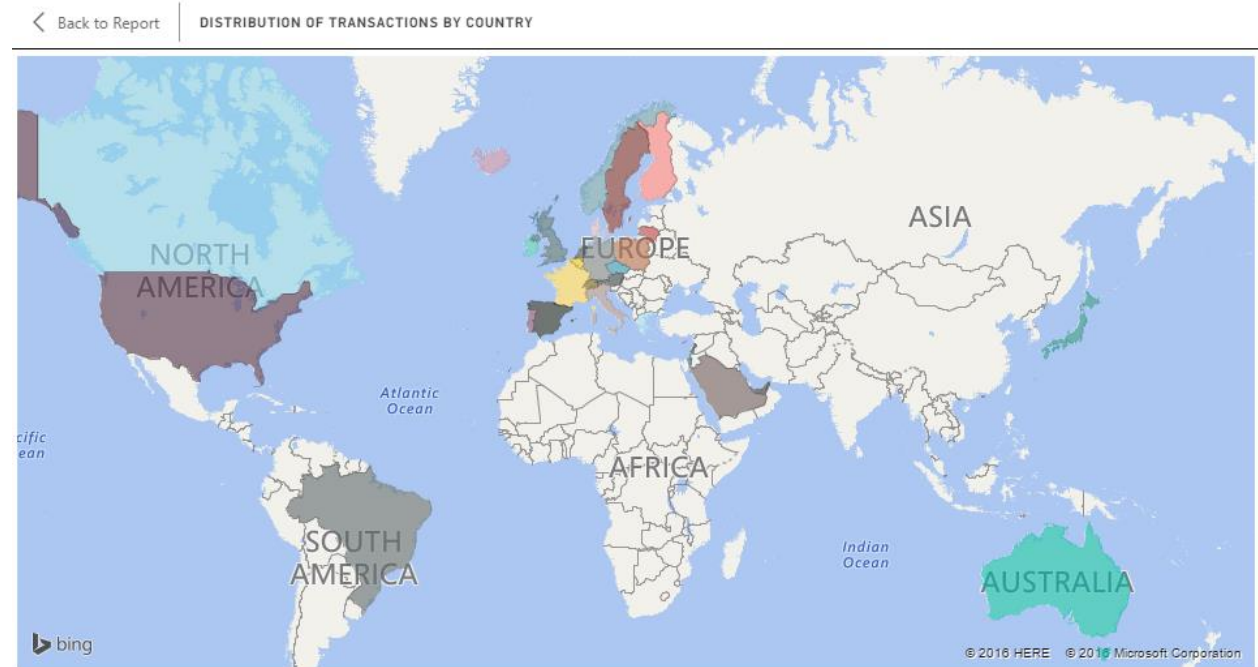
c.



d.



e.



## 5. Data Pre-Processing

For the purpose of preprocessing and cleansing the dataset we made use of the efficiency of ML Studio to perform the same. On analysis of the dataset we came across the following bad data:

- Good Practice for Pre-processing: Avoid names, values or fields with blank spaces, otherwise each word will be interpreted as a separate variable, resulting in errors that are related to the number of elements per line in your data set.
- The "Description" column had a missing value count of 1501.
- The "CustomerID" column had a missing value count of 135080.
- The "Quantity" column had negative values in it.
- The "Unit Price" column had negative values in it.
- We can perform feature enhancement by adding more features to the dataset.
- The columns Unit Price and Quantity had outliers

We resolved the above given analysis by

- Replaced the values of columns which had spaces between words with underscore.
- Deleting the rows which had a missing description
- Deleting the rows which had Quantity and Unit Price as negative values.
- Replaced the blank CustomerID 's as "guestuser" as we are assuming that the customer that placed the order is not a registered user, thus not having a CustomerID.
- Separated the InvoiceDate column which is of the DateTime format and created new features called as InvoiceDate and InvoiceTime.
- Created a new feature Sales Amount which is the multiplication of Quantity and Unit Price. Deployed feature engineering here
- Created a Classifying feature called as "Contribution\_to\_OverallSales" which defines how a particular product has contributed to the overall sales of the company. So it is of categorical feature with the values being "Good\_contributor" and "Bad\_contributor".
- Before building the feature "Contribution\_to\_OverallSales" we performed aggregation of the Sales\_Amount by ProductID, Country, and Date.
- We have done further feature enhancement by building features such as Year, Hour, and Month from Invoice Date. By doing this we could do analysis of products and customers by geography and time.
- We have also made the Year, Month and Hour features as categorical.
- We noticed that making these features as categorical our performance metrics have improved considerably while Prediction and classification
- We dealt with the outliers by using Boxplot for Unit Price and Quantity features
- We made sure to remove the outliers from Unit Price and Quantity well before Calculating the Sales\_Amount feature.
- After the cleansing we reduced the size of our Dataset by 10%

## 6. Algorithms

### 1. RECOMMENDATION:

- This section explains the implementation and the application of a Recommendation system built around the dataset.
- The main aim of a recommendation system is to recommend one or more items to users of the system.
- There are two approaches to recommender systems, content-based and collaborative filtering. To implement a recommender system, we used the Matchbox Recommender on ML Studio.
- The Matchbox recommender on ML Studio combines both the approaches, Collaborative filtering and Content based thus also called as a hybrid recommender.
- This dataset has 8 columns before preprocessing, and to build a recommendation model the structure of the dataset has to be in a format of user-item-rating triples.
- For this dataset, the CustomerID column is the user, StockCode column is the item, and for the rating column we had to infer a new column based on the data we had.
- To create a rating column, we calculated the frequency of a product bought by the customer. It created a distribution of values ranging from 1-490.  
Drawback: The matchbox recommender has a limit of 0-99 or 1-100 distribution of ratings so the model would not train.
- Thus we grouped the frequency ratings and gave ratings to range from 1-5 in a R Script in MLStudio.
- In the end, we have a dataset of the structure CustomerID-StockCode-rating.

#### **Training/Scoring/Evaluating Matchbox Recommender:**

- To train the matchbox recommender, we split the data into test and train datasets using the recommender split option in the Split data module.
- Split the data in the proportion of 75% and 25%.
- Then we Add the Train MatchBox Recommender module and connect it the training data which returns a trained Matchbox recommender.
- We then used the Score Matchbox Recommender module, which creates recommendations for the different users.



## Recommendation &gt; Score Matchbox Recommender &gt; Scored dataset

rows

columns

1008

6

User

Item 1


Item 2


Item 3


Item 4


Item 5


view as




















14702

21672

21078

22720

84978

47566

17962

17090A

22161

17021

22335

85152

18150

21906

22620

22568

22570

20970

15296

17090A

22355

20724

20723

23202

16759

23081

23080

22910

22620

85099C

13819

20727

20726

20725

21931

22326

14047

22328

22467

23207

23206

22840

18265

20914

84879

23298

47566

21539

17946

21034

85099C




20751

21385

21386

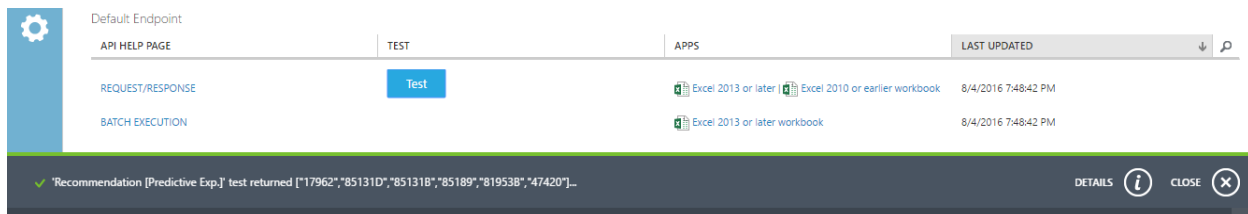
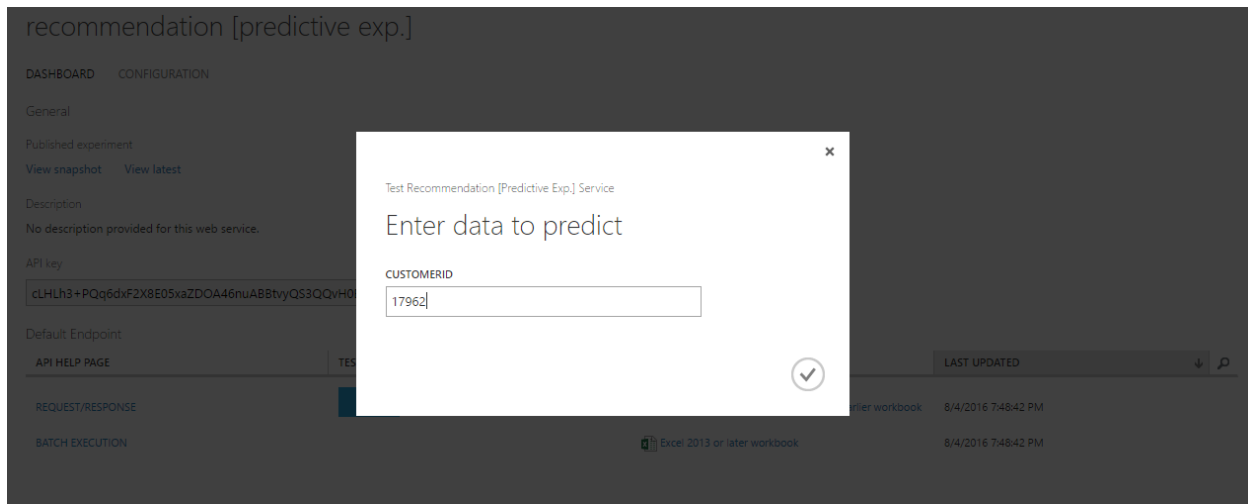
- We then added a recommender module to measure the accuracy of the recommending products to a user. It computed the accuracy as (Please find the screenshot below.) which is called as the normalized discounted cumulative gain (NDCG).

## Recommendation &gt; Evaluate Recommender &gt; Metric

rows	columns		
1	1		
		NDCG	
view as			
			
		0.822159	

**Web Service Deployment:**

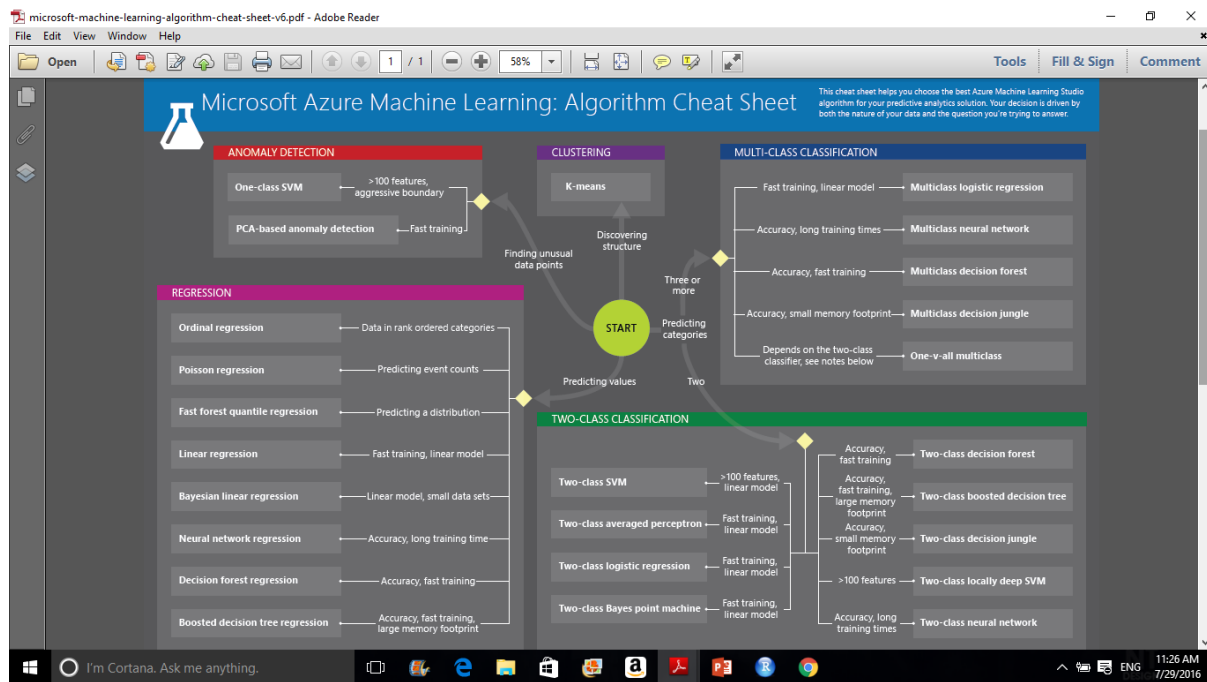
- We then set the web service and deployed it as a predictive experiment where we select the Web Service input (CustomerID) and the output will be the Recommended Items.
- The web service can then be tested by selecting the web service, click on test which will display a pop-up shown below:



**Business Value of Recommendation: By using this recommender model you can recommend appropriate products to specific customers.**

## 2. CLASSIFICATION:

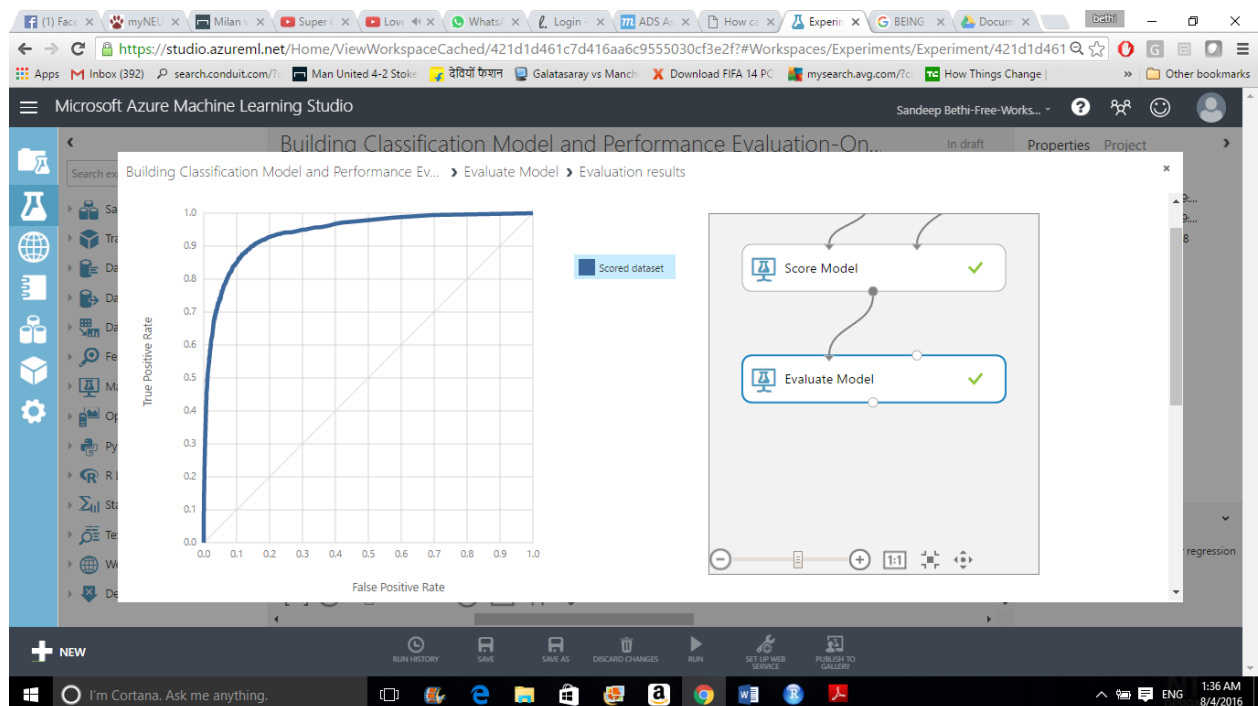
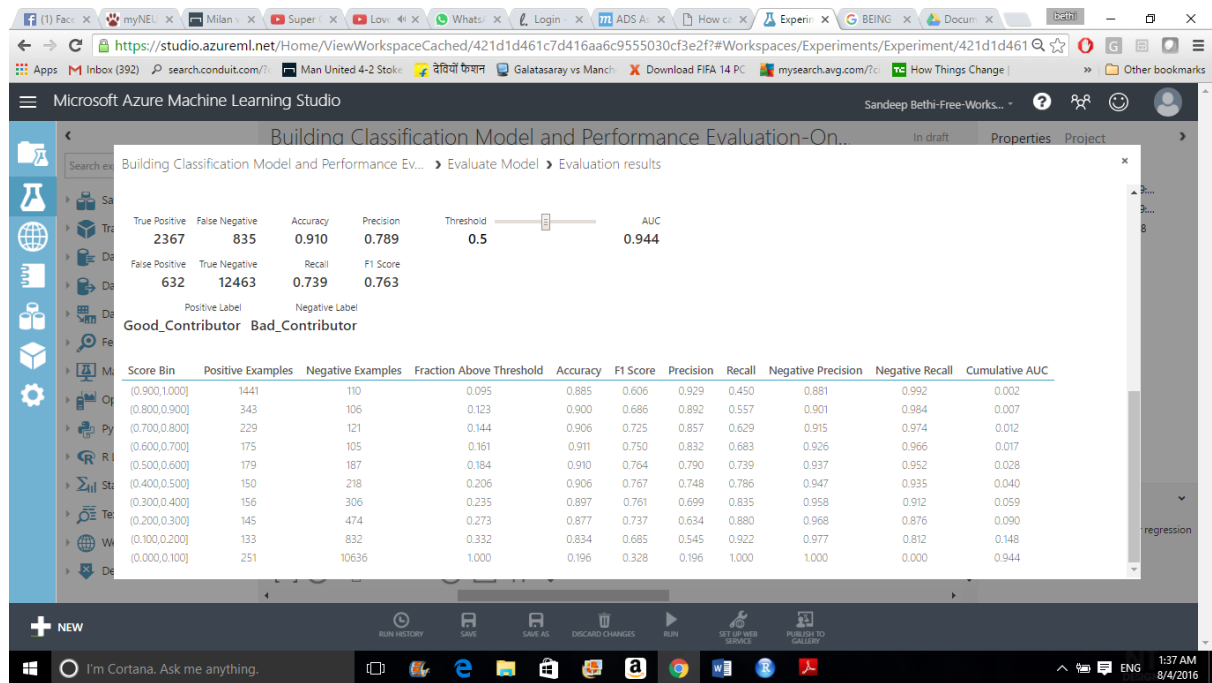
- Our Classification includes classifying whether a particular product is a good contributor or a bad contributor for the overall revenue of a company. For this we made use of the “Sales\_Amount” column as well as the “Contribution\_to\_Overallsales” column which we have built during feature engineering.
- Based on the information provided in the following link we have applied 4 algorithms for classification and evaluated their performances against each other.  
<https://azure.microsoft.com/en-us/documentation/articles/machine-learning-algorithm-choice/>



- We built four models for classification and evaluated them against each other based on their performance metrics. We selected one model for classification which has better performance metrics in going ahead and deploying the webservice.
  - Two-class logistic regression
  - Two-Class Average Perception
  - Two-class support vector machine(SVM)
  - Two-class Bayern point machine
- The above link contains information regarding How to choose algorithms for Microsoft Azure Machine Learning.
- It has Machine Learning Algorithm Cheat Sheet which tells what algorithm to be used for different type of problems like clustering, classification and regression.
- It also gives the advantages and disadvantages of each algorithm for a set of problem.
- Before building these models we have done **feature selection** by using filter based feature selection module present in Microsoft Azure ML Studio.
  - We selected this module because the features present in our dataset are of both categorical and numeric type.
  - Hence we selected this module for determining the importance of features in building the models.

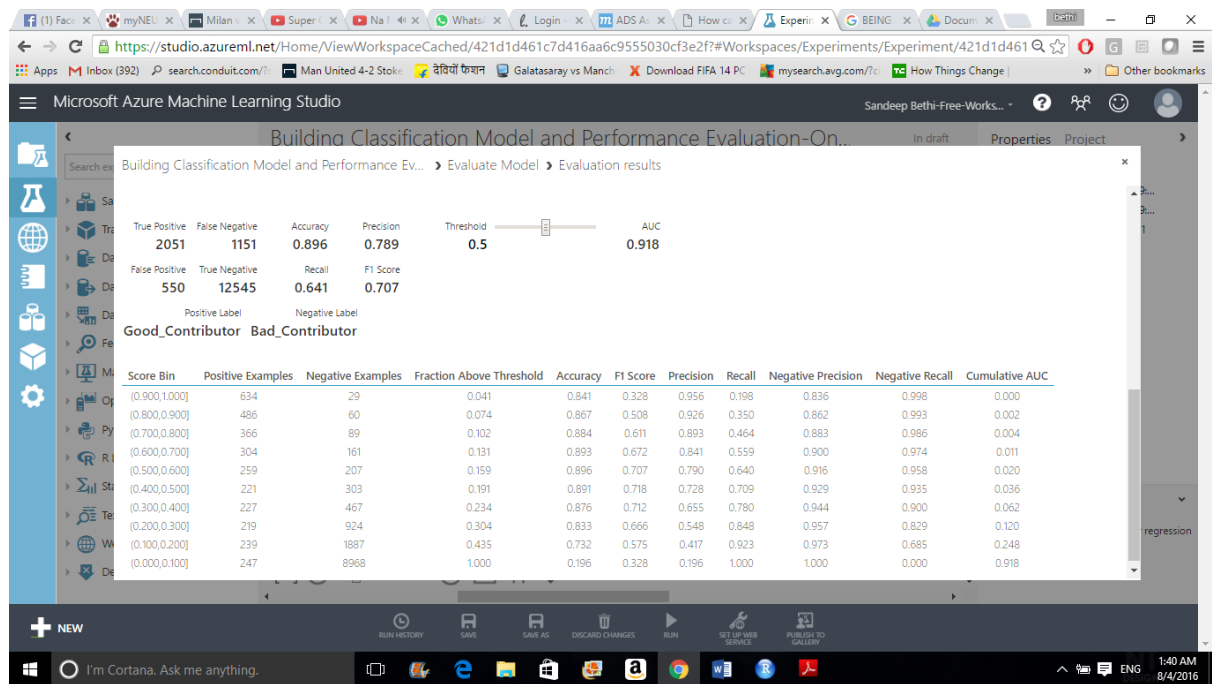
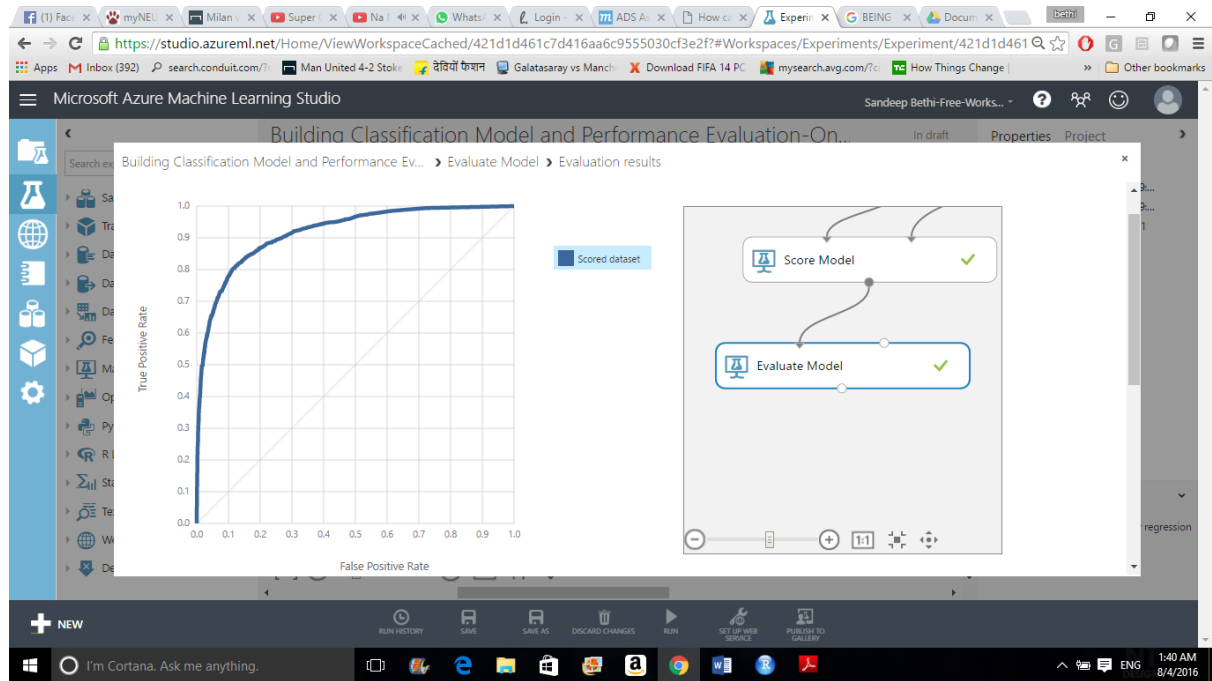
### A) Two-class logistic regression

This particular model gave us the following ROC curve and other performance metrics



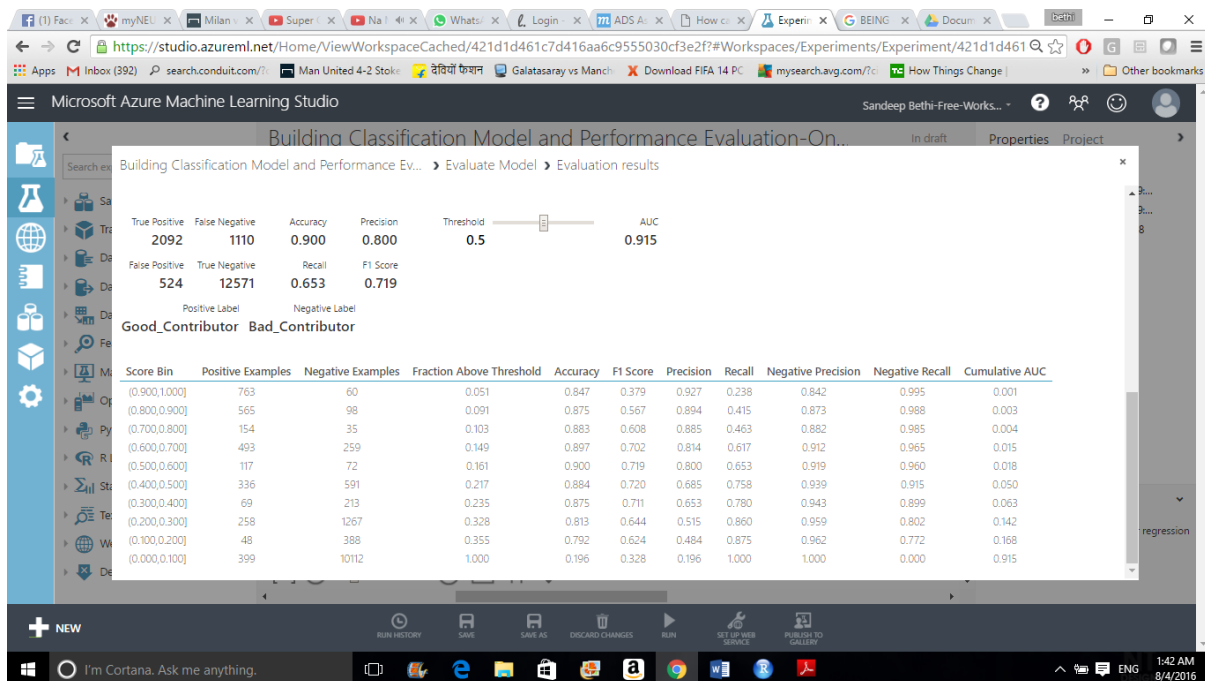
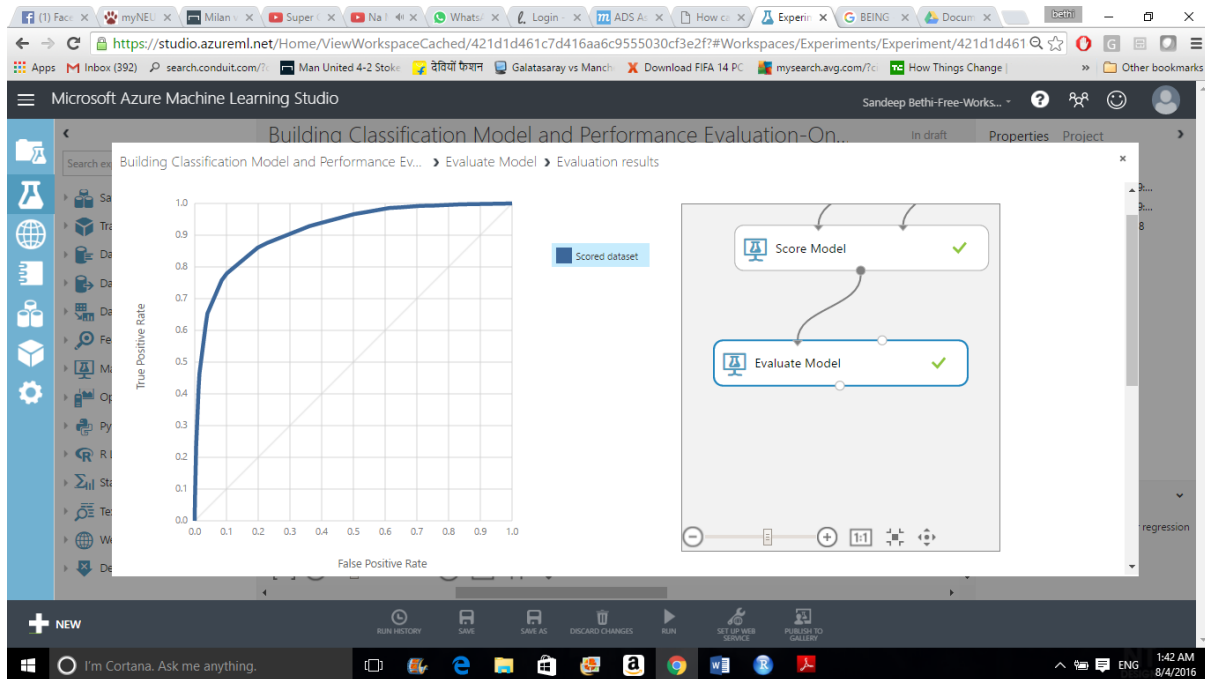
## B) Two-Class Average Perception

This particular model gave us the following ROC curve and other performance metrics



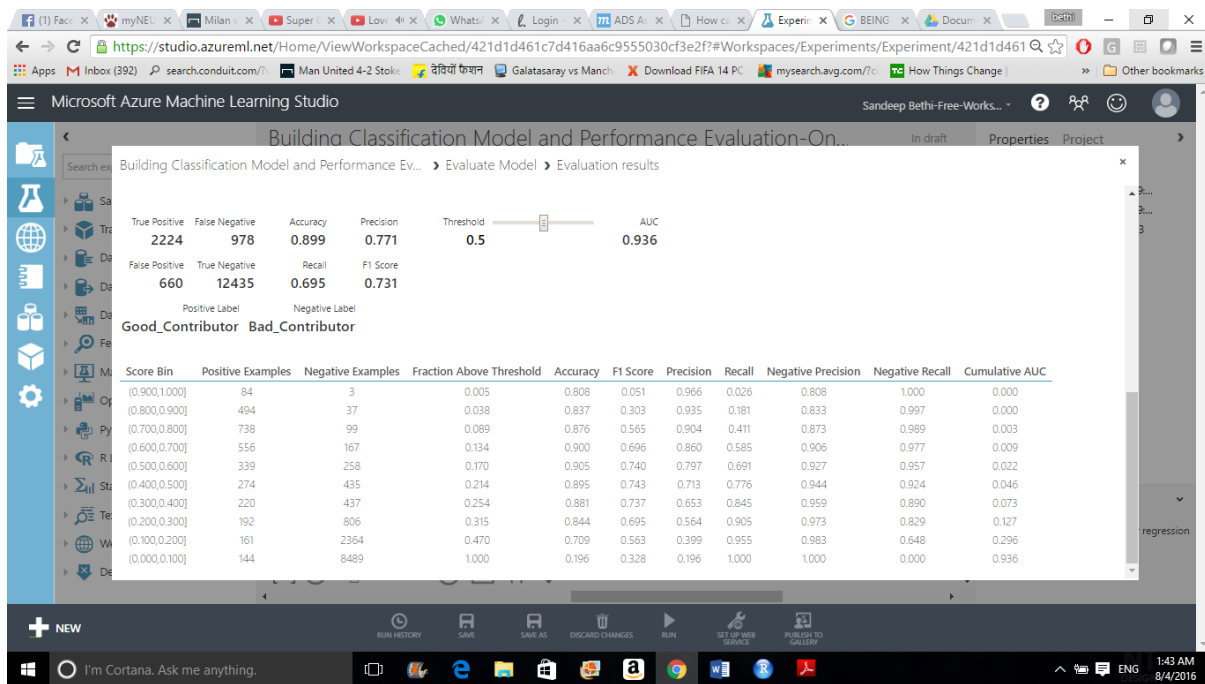
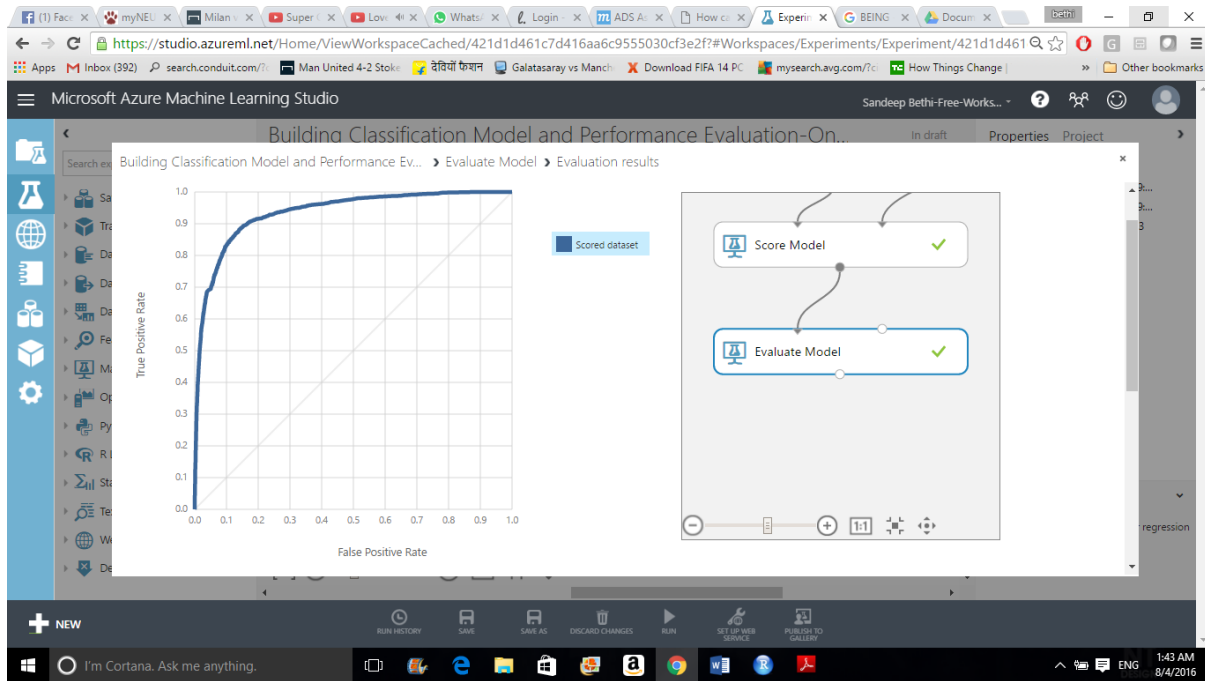
### C) Two Class support vector machine(SVM)

This particular model gave us the following ROC curve and other performance metrics



## D) Two-Class Bayes Point Machine

This particular model gave us the following ROC curve and other performance metrics



**CONCLUSION:**

This table gives the comparison of performance metrics of 4 different algorithms used in clustering

	Two-class logistic regression	Two-Class Average Perception	Two Class support vector machine(SVM)	Two-Class Bayes Point Machine
AUC	0.944	0.918	0.915	0.936
Accuracy	0.910	0.896	0.900	0.899
Precision	0.789	0.789	0.800	0.771
Recall	0.739	0.641	0.653	0.695

By looking the above table we can clearly say that the overall performance i.e AUC of **Two-class logistic regression Algorithm** is much better than the rest of the Algorithms.

Even the other performance metrics such as Accuracy, Precision and Recall are better for Two-Class logistic regression Algorithm.

Hence for the Classification of whether a particular product is a Good\_Contributor or not we use **Two-Class logistic regression** Algorithm and deployed the web service using this model.

**Business Value of Classification: By classifying whether a product is a good\_contributor or a bad contributor to the overall sales, the company can take measures and decide whether to continue selling that particular product or not.**

**3. PREDICTION:**

- In Prediction we are going to forecast the Sales Amount of any particular product based on the country and time it is sold. We are going to get the Sales Amount prediction analysis of different products by geography and time.
- We built the following five models for Prediction and evaluated them against each other based on their performance metrics. We selected one model for Prediction of Sales Amount which has better performance metrics in going ahead and deploying the web service

A) Poisson Regression

B) Linear Regression

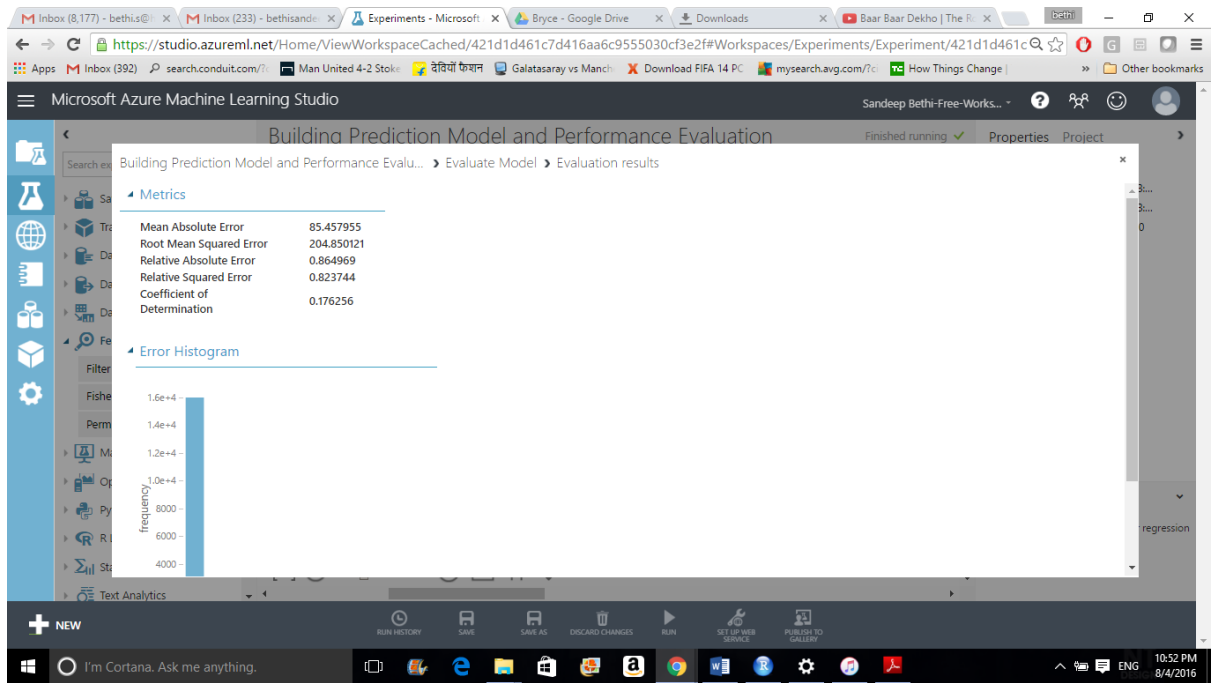
C) Neural Network Regression

D) Decision Forest Regression

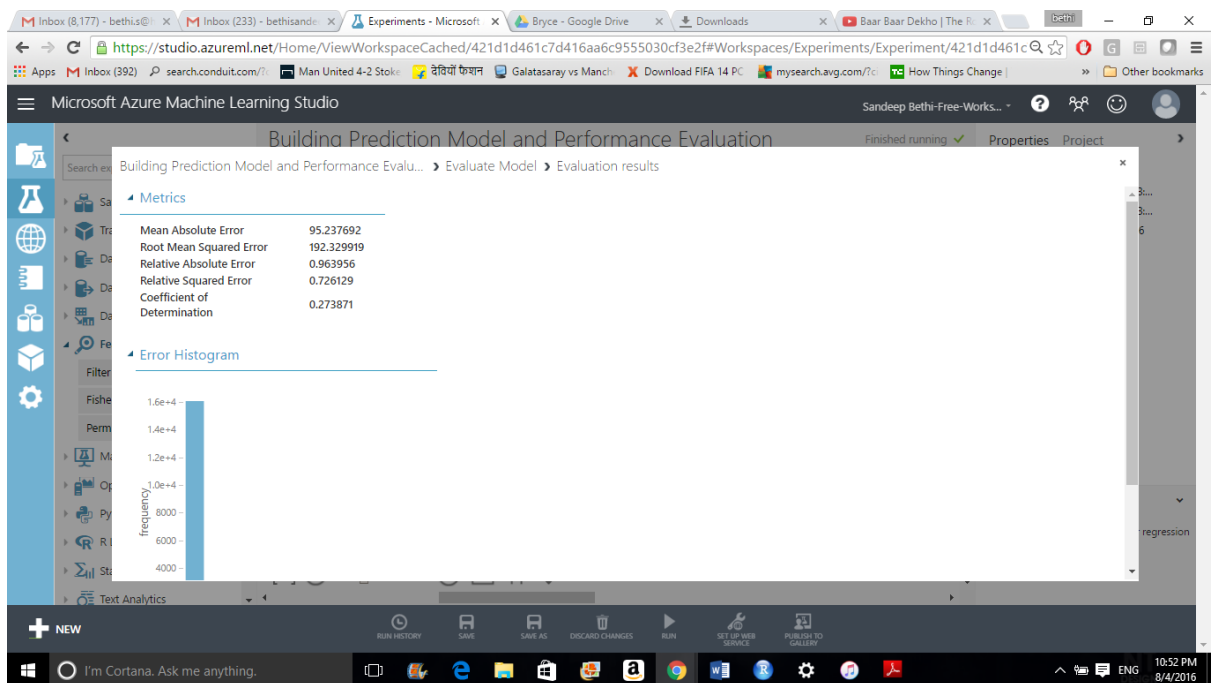
E) Boosted Decision Tree Regression



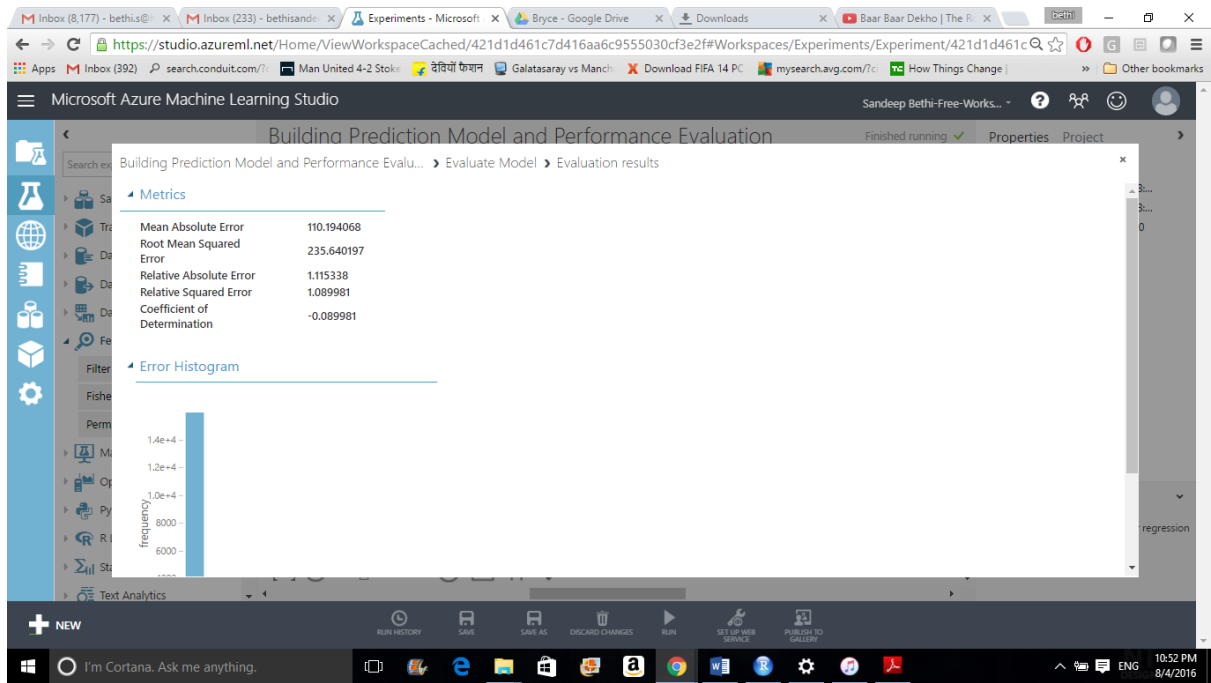
A) **POISSON REGRESSION:** This model gave the following Performance metrics



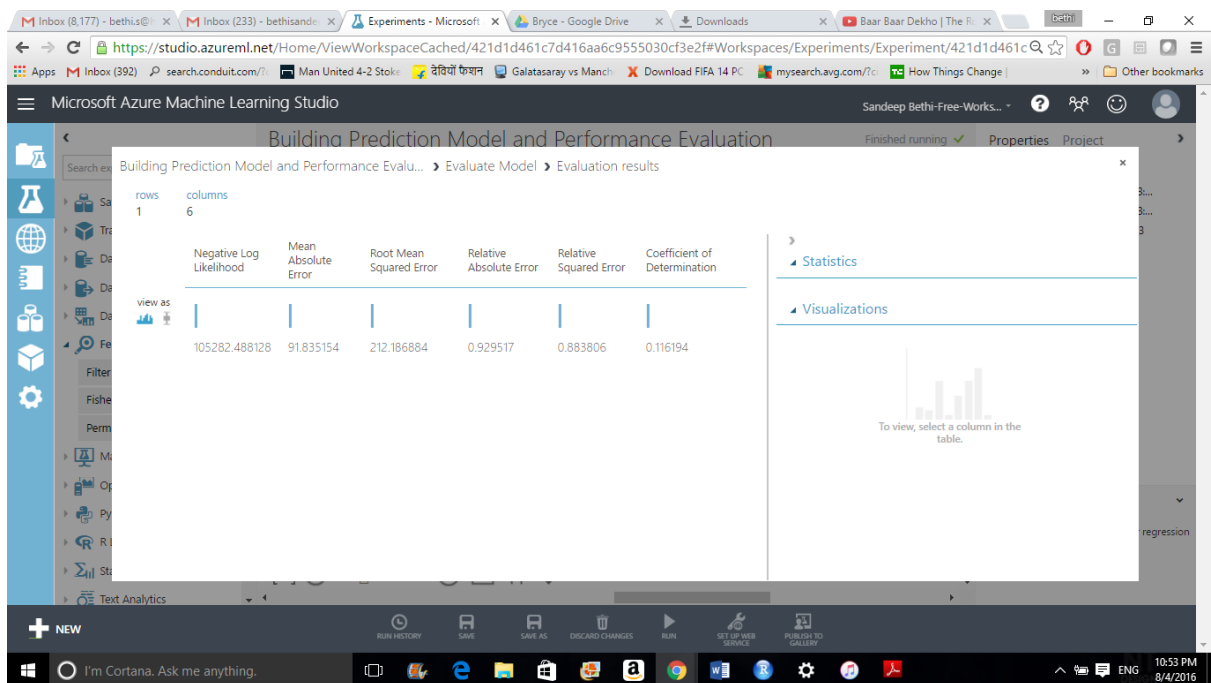
B) **LINEAR REGRESSION:** This model gave the following Performance metrics



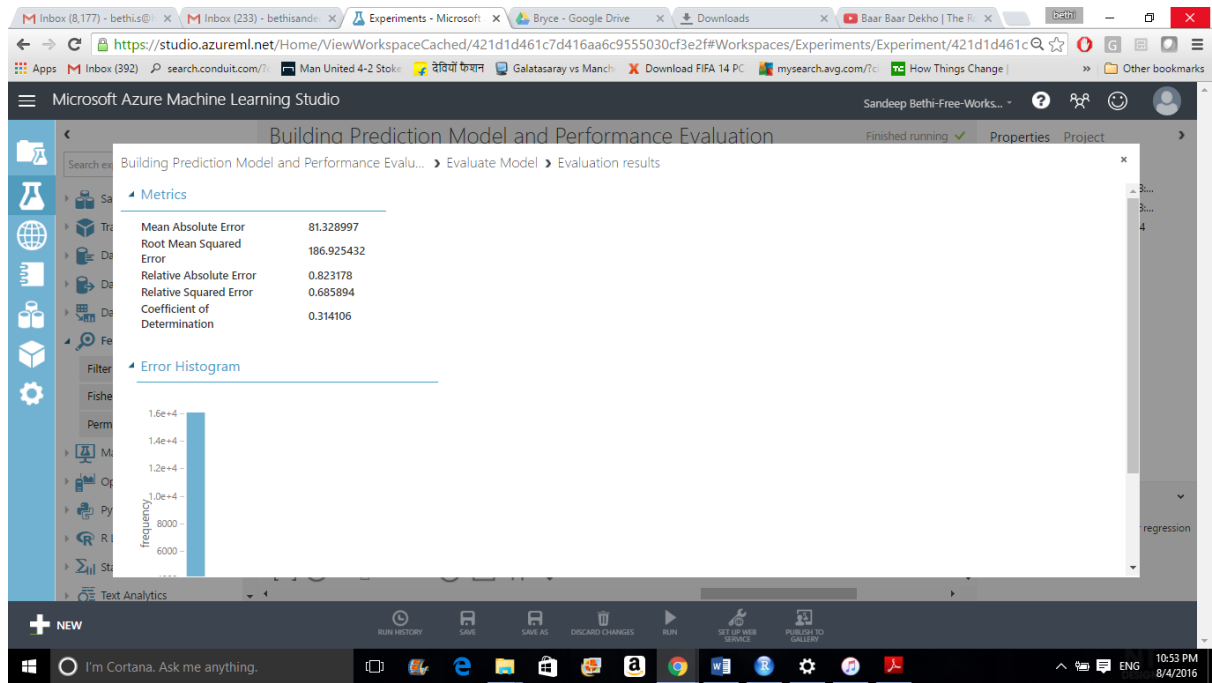
C) **NEURAL NETWORK REGRESSION:** This model gave the following Performance metrics



D) **DECISION FOREST REGRESSION:** This model gave the following Performance metrics



E) **BOOSTED DECISION TREE REGRESSION:** This model gave the following Performance metrics



## CONCLUSION

This table gives the comparison of performance metrics of 5 different algorithms used in prediction

	POISSON REGRESSION	LINEAR REGRESSION	NEURAL NETWORK REGRESSION	DECISION FOREST REGRESSION	BOOSTED DECISION TREE REGRESSION
MEAN ABSOLUTE ERROR ( <b>MAE</b> )	85.45	95.23	110.19	91.83	81.32
ROOT MEAN SQUARE ERROR ( <b>RMSE</b> )	204.85	192.32	235.64	212.18	186.92
RELATIVE ABSOLUTE ERROR ( <b>RAE</b> )	0.86	0.96	1.15	0.92	0.82
RELATIVE SQUARED ERROR ( <b>RSE</b> )	0.82	0.72	1.08	0.88	0.68
COEFFICIENT OF DETERMINATION ( <b>COD</b> )	0.17	0.27	0.08	0.11	0.31

By looking the above table we can clearly say that the overall performance of **BOOSTED DECISION TREE Algorithm** is much better than the rest of the Algorithms. Because the Mean absolute error(MAE), root mean square error(RMSE), relative absolute error(RAE), relative squared error (RSE) values are lower in this particular model compared to other models.

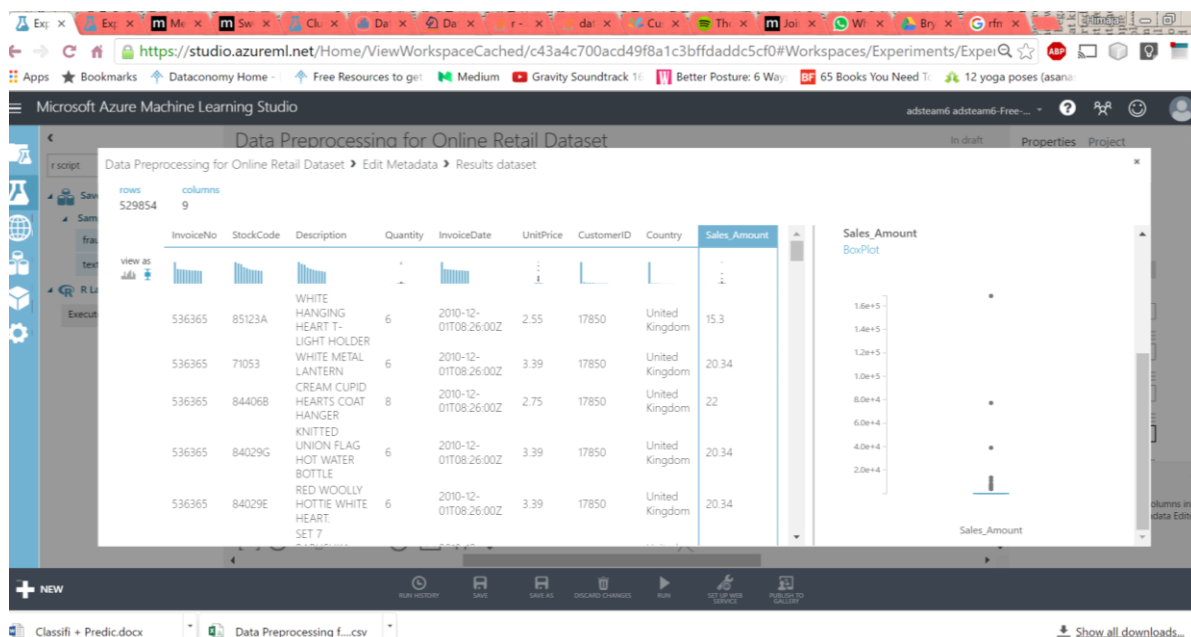
Hence for the Prediction of Sales Amount of a product in a particular time and country the **BOOSTED DECISION TREE** Algorithm is better and we choose this model for deploying the web service.

**Business Value of Prediction: By predicting the sales of the product by country and time, the company can take good measures in beforehand to allocate proper resources so that they can achieve their goals. They can also design their marketing strategy of different products in a better way.**

## 7. FUTURE SCOPE

### CLUSTERING:

- Our clustering model borrows the concept from RFM analysis.
- With the prepared target dataset we intended to identify whether consumers can be segmented meaningfully in the view of recency, frequency and monetary values. The *k*-means clustering algorithm was employed for this purpose
- RFM (recency, frequency, monetary) analysis is a marketing technique used to determine quantitatively which customers are the best ones by examining how recently a customer has purchased (recency), how often they purchase (frequency), and how much the customer spends (monetary).
- The *k*-means clustering algorithm is very sensitive to a dataset that contains outliers (anomalies) or variables that are of incomparable scales or magnitudes.



- ✓ Executed an R script to calculate the R-F-M values based on the three columns IE: CustomerId, InvoiceDate and Sales\_Amount
- ✓ Calculate RFM

- Remove the duplicate records with the same customer ID
- Find the most recent date for each ID and calculate the days to the endDate, to get the Recency data
- Calculate the quantity of transactions of a customer, to get the Frequency data
- Sum the amount of money a customer spent and divide it by Frequency, to get the average amount per transaction that is the monetary data
- ✓ Arguments for consideration
  - df - A data frame of transaction records with customer ID, dates, and the amount of money of each transaction
  - startDate - the start date of transaction, the records that happened after the start date will be kept
  - EndDate - the end date of transaction, the records that happened after the end date will be removed. It works with the start date to set a time scope
  - CustomerID - the column name which contains customer IDs in the input data frame
  - InvoiceDate- the column name which contains transaction dates in the input data frame
  - SalesAmount- the column name which contains the amount of money of each transaction in the input data frame
- ✓ The three variables *Recency*, *Frequency* and *Monetary* were chosen as input for the clustering analysis.
- ✓ Filtering was done to exclude from the analysis any instances having a rare value for any variables involved, and the minimum cut-off value for rare values was set between reasonable values of R-F-M using R script.

### **K-Means Clustering**

**K-Means Clustering** module to create an untrained K-means clustering model. K-means is one of the simplest and the best known *unsupervised* learning algorithms, and can be used for a variety of machine learning tasks, such as detecting abnormal data, clustering of text documents

Distance: Euclidian

#### **Business Prospects:**

- ✓ Cluster formation size: 5 from varying recency, frequency and monetary values
- ✓ ***Lower the recency, high frequency and high monetary -> good customer***
- ✓ Recency inversely proportional to frequency and monetary
- ✓ Cluster 1 should ideally be composed of 15-20 per cent of the whole population. This group seems to be the least profitable group as none of the customers in this group purchased anything in the second half of the year. Even for the first half of the year, the consumers didn't shop often, and the average value of frequency range would be around 1.0-1.5 (LOW).
- ✓ Cluster 5 mainly started shopping with the online retailer at the beginning of the year, and continued to the end of the year with an average value range of recency 0.5-2. They purchased quite often and as a result, spent a quite high amount of money. This group of consumers can be categorized as very low recency (HIGH), very high frequency and very high monetary with a high spending per consumer. These consumers should be contributing 25-30 per cent of the total sales in the year. This group, will be the smallest (only composed of 5-10 per cent of the whole population), seems to be the most profitable group.
- ✓ Cluster 4 contains some 10 percent consumers with a very high value for frequency and monetary, although lower than those of cluster 5. This group seems to be the second high profit group.

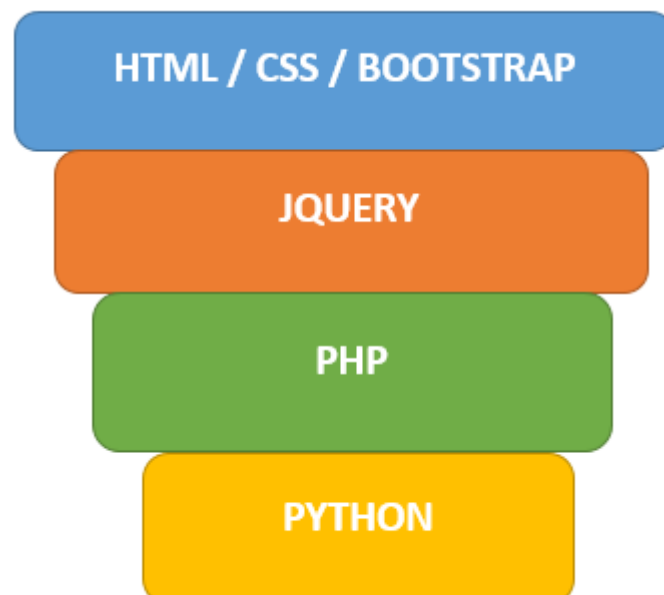
- ✓ Cluster 2 compared with clusters 4 and 5, this group of customers has a lower frequency throughout the year and a significantly smaller average value of monetary, indicating that a much smaller amount of spending per consumer. This group can be categorized as high recency (LOW), high frequency and medium monetary with a medium spending per consumer.
- ✓ Cluster 3 should be the largest-sized group with largest number of consumers. Consumers in this group have a reasonable value of frequency. Compared with clusters 2 and 4, this group has a lower but reasonable value of monetary as the group includes many newly registered consumers starting shopping with the retailer very recently. This group seems to have represented ordinary consumers and therefore has a certain level of uncertainty in terms of profitability. In the long-term view, some of the consumers might be potentially very highly profitable or unprofitable at all

#### **Enhancing clustering analysis of CLUSTER 3 using decision tree**

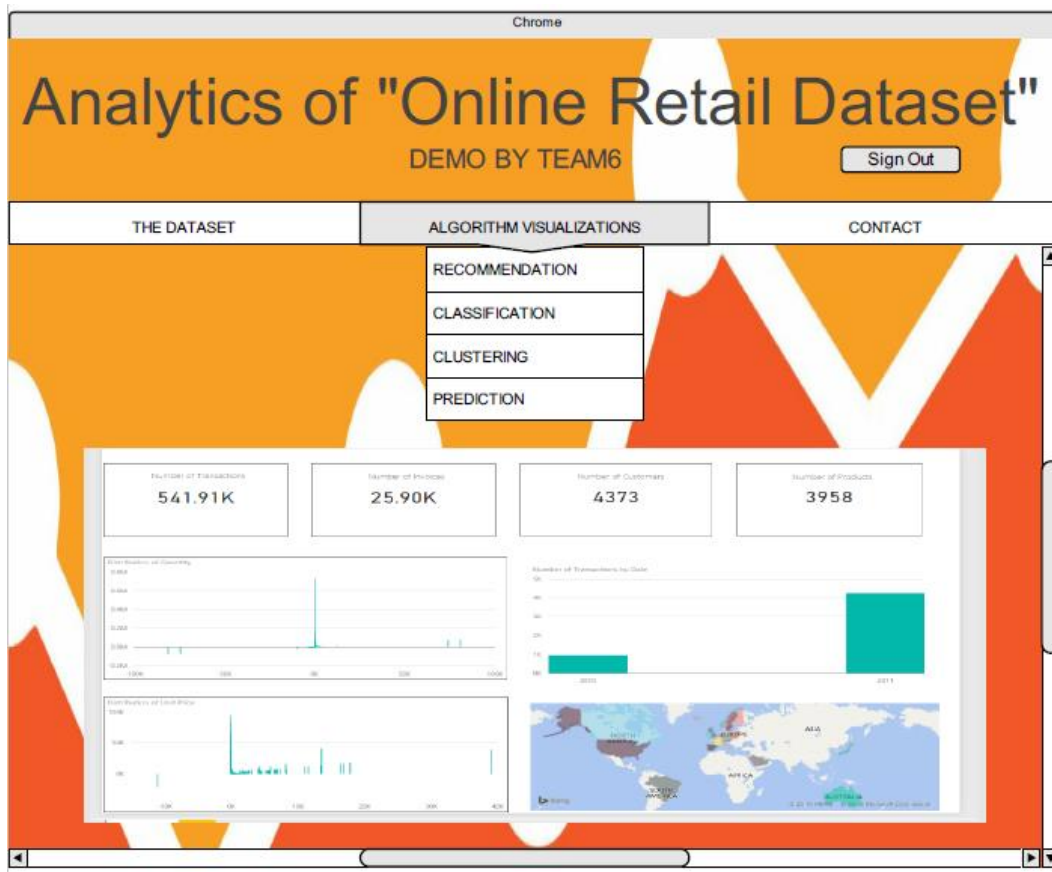
- Cluster 3 is the most diverse cluster among the five identified clusters in the sense that it contains both newly registered and old customers as well. To refine the segmentation of the instances in this cluster, a decision tree can be used to create some nested segments internally inside the cluster
- In other words, these nested segments form some sub-clusters inside cluster 3, and make it possible to categorize the consumers concerned into some sensible sub-categories.
- *Interesting insight:* the relationship between frequency and monetary seems to be a monotonic linear relationship.

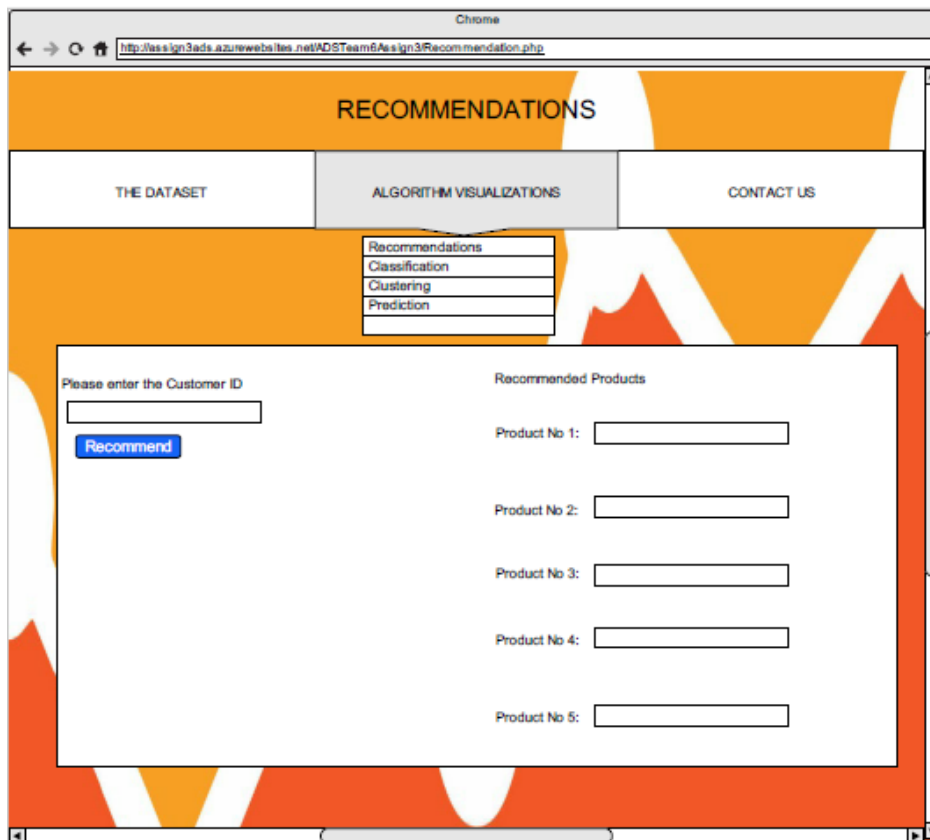
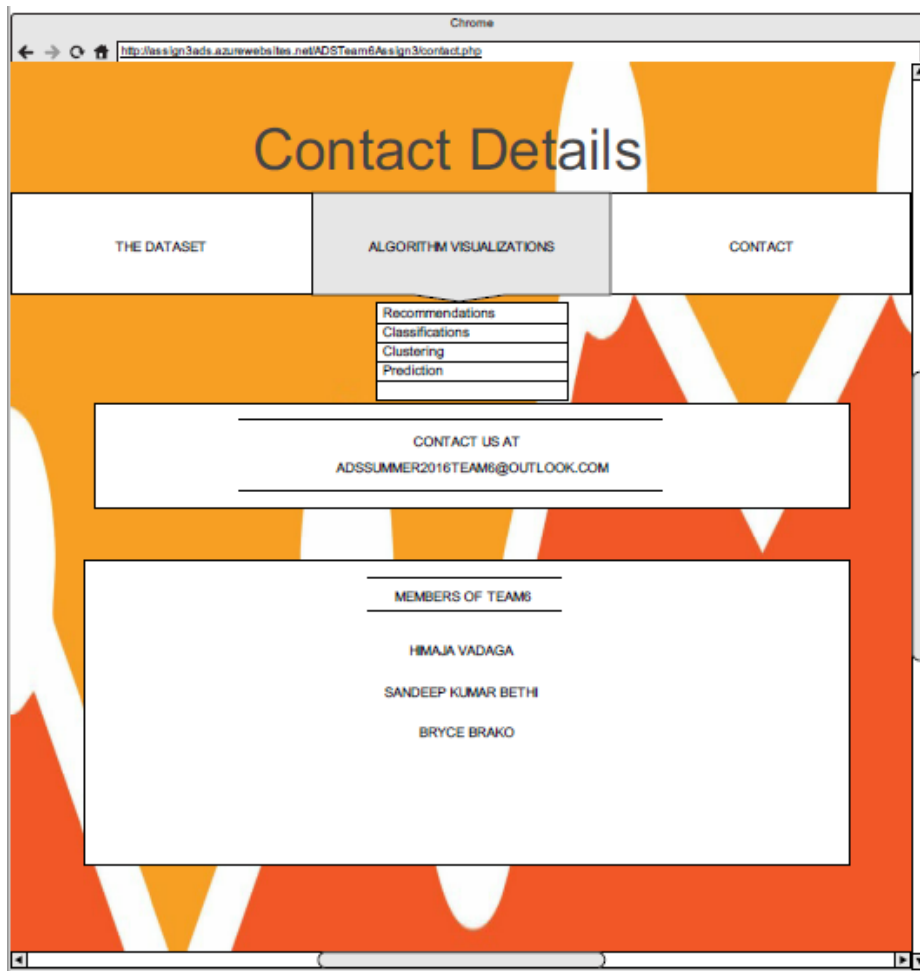
## 8. Web Site Design

- The website was designed using HTML, CSS, Bootstrap, JQuery, PHP and Python.
- The process flow design can be depicted as:



- The Main Page is loaded using HTML, CSS and JavaScript.
- The Visualization on the dashboard are implemented by embedding POWERBI into the webpage.
- Using JQuery we call a PHP page which in turn executes a python script.
- The python script invokes the webservice which retrieves the data based on the input provided.
- In order to build this website we followed the three-click rule where the user can navigate to his desired page within 3-clicks.
- We built the wire frame of our website using 'MOQUPS' and you can find the design of the web pages below:







Chrome

http://assign3ads.azurewebsites.net/ADSTeam6Assign3/Recommendation.php

## CLASSIFICATION

THE DATASET	ALGORITHM VISUALIZATIONS	CONTACT US
-------------	--------------------------	------------

Recommendations

Classification

Clustering

Prediction

Please enter the following details

YEAR:

MONTH:

COUNTRY:

PRODUCT ID:

PRODUCT NAME:

OUTPUT:

Type of contributor to overall revenue of the company:

Probability of the product being a good contributor to overall revenue of the company:

Chrome

http://assign3ads.azurewebsites.net/ADSTeam6Assign3/Recommendation.php

## CLUSTERING

THE DATASET	ALGORITHM VISUALIZATIONS	CONTACT US
-------------	--------------------------	------------

Recommendations

Classification

Clustering

Prediction

Please enter the Customer ID

OUTPUT:

TYPE OF CUSTOMER

Chrome

← → ↻ 🏠 <http://assign3ads.azurewebsites.net/ADSTeam6Assign3login.php>

## Analytics of "Online Retail Dataset"

LOGIN WINDOW

USERNAME:

PASSWORD:

Sign In

Chrome

← → ↻ 🏠 <http://assign3ads.azurewebsites.net/ADSTeam6Assign3Recommendation.php>

## PREDICTION

THE DATASET **ALGORITHM VISUALIZATIONS** CONTACT US

Recommendations  
Classification  
Clustering  
Prediction

Please enter the following details

OUTPUT:

YEAR:

SALES PREDICTION

MONTH:

COUNTRY:

PRODUCT ID:

PRODUCT NAME:

Predict

## 9. Hosting the Web Application

We chose Microsoft Azure to host the web application as it is a Microsoft product and we were already using ML Studio to perform analytics.

Microsoft DreamSpark is a free subscription within Azure which gives us the capability to host our own website under the azurewebsites.net domain easily.

We are using a FTP Client 'FileZilla' to upload the web app files to the azure cloud and navigating to the website using this URL →

<http://assign3ads.azurewebsites.net/ADSTeam6Assign3/login.php>

The configuration steps on Azure are as follows:

- a. Login in to portal.azure.com
- b. Click on the App Services tab, and click on Add.
- c. Enter your App name, 'Create new' Resource Group or 'Use existing'.
- d. Select the App and set up the deployment credentials if they are not set. Install 'FileZilla' and use the credentials to login remotely to the server to deploy your app files in the appropriate location.
- e. Additional ways to deploy your application can be found on this link:  
<https://azure.microsoft.com/en-us/documentation/articles/web-sites-deploy/>

## 10. References

- <https://archive.ics.uci.edu/ml/datasets/Online+Retail#>
- Daqing Chen, Sai Liang Sain, and Kun Guo, Data mining for the online retail industry: A case study of RFM model-based customer segmentation using data mining, Journal of Database Marketing and Customer Strategy Management, Vol. 19, No. 3, pp. 197â€“208, 2012 (Published online before print: 27 August 2012. doi: 10.1057/dbm.2012.17).
- <https://azure.microsoft.com/en-us/documentation/articles/machine-learning-algorithm-choice/>