

Business Analysis Using Demographic Data Modelling and Visualisation

# Submitted By: Submitted To:

## Aman Parmar (16103221) ​ Kirti Agarwal

## Sameer Bairwa (16103309)

## Siddhartha Singhal (16803010)

# Abstract

Business Analysis is the practice of enabling change in an organizational context, by defining needs and recommending solutions that deliver value to stakeholders.

This is a project which mainly focuses on the techniques of Data Preprocessing, Data Visualization, Data Analytics, and Machine Learning algorithms. This project had several different dimension of each category mentioned above ranging from data retrieval through Twitter API to filtration of data i.e. removal of unnecessary details in Data Preprocessing, plotting count of variables to geographical location plotting in Data Visualization along with the use of Shiny package for demographic plotting of sentiment histograms, pattern finding to predict the possible responses and outcomes in Machine Learning.

The challenges associated with the project included the learning and implementation of R language in all the three categories, followed by the objective of working upon the pre-existing datasets available in the public domain as well as learning about fetching data through various APIs and then the compilation of entire project under one package.

# Introduction

In today's world where social media usage is growing exponentially, there exists a need to organize, analyze and predict the contents shared over it. Twitter which receives almost 6000 tweets per seconds and has a sum total of 232 million active users has become one of the fastest growing social platforms for information sharing and awareness.

Business Analysis Using Demographic Data Modelling and Visualisation is a project which mainly focuses on the techniques of Data Preprocessing, Data Visualization and Machine Learning algorithms with the future scope of implementation of real-time extraction of informative measures from a dataset and Database Connectivity for the future references. This project had several different dimension of each categories mentioned above ranging from data retrieval through Twitter API to filtration of data i.e. removal of unnecessary details in Data Preprocessing, plotting count of variables to geographical location plotting in Data Visualization along with the use of Shiny package for plotting sentiment histograms, pattern finding to predicting the possible responses and outcomes in Machine Learning. The challenges associated with the project included the learning and implementation of the R language in all the three categories, followed by the objective of working upon the pre-existing datasets available in the public domain and then the compilation of entire project under one package. Upon implementation of the of all the techniques mentioned above the project had a well-developed model which is capable enough of making the raw data made available ready for effective use and visualization. After this the project model can be utilized to dig up some existing data patterns hidden inside the dataset and find the relation amongst different variables, collect some most informative information and can later on can be trained well enough to predict the response and conclusion on a particular or a group of inputs using the machine learning techniques. The future aspects upon implementation will make the model more effective and stable on the grounds of data analysis and sentiment prediction.

# Goals

1. Get a proper insight into the target product/element.
2. Proper visualization for a better understanding of the situation about the general public.
3. Getting real-time tweets and information about the crawled data for analysis
4. Getting proper visualizations for easing out the process of framing marketing and advertising strategies.

# Data Source and Datasets:

## Data Source​ :

* + <https://www.kaggle.com>

## Dataset​ :

* + Sentiment classification of tweets
    - Number Of Rows: 14641
    - Number Of Columns: 13

## Dataset Link​ :

* + https://www.kaggle.com/valencar/sentiment-classification-of-tweets

## Real-time extraction​ of data using the ​Twitter API.

**Background Study and Findings**

Twitter is a social networking service which allows the user to send and read the short message of 140 characters called “tweets”. There are two types of users for twitter account.

One is registered users who can only read the tweets and another are registered users who can read and post the tweets. It is a public platform for all the people of different

age categories all over the world. Data generated by twitter is heterogeneous in terms of content because user can post a **text and image.**

In the research paper, we found that using kmeans clustering over the twitter dataset leads to some flaws while making the predictions. Being a basic clustering model kmeans provides us with highly overlapped clusters.

For analyzing the word pattern of the tweets wordcloud was used as a visualisation technique. Naïve Bayes being a popular algorithm in sentiment analysis problems is used along with an ensemble of decision trees, random forest and K Means. Background study showed us that an average of about 84% accuracy is obtained while doing sentiment analysis over text data. An accuracy greater than 84% is a sign of overfitting.

Due to lack of interrelated attributes, an ensemble of algorithms is used instead of a single one so as to provide the best predicting model possible.

Live tweets, on a particular keyword of our interest, were crawled using the twitter API incorporated in Twittr Package of R which also covers a proper user authentication method. To get some extra insights from the tweets we used an additional dataset, having multiple attributes, collected Kaggle.

# Model Assumptions

* Data Source and Dataset :

● Data Source : https://www.kaggle.com

● Dataset : Sentiment classification of tweets

➢ Number Of Rows : 14641 ➢ Number Of Columns : 13

● DatasetLink:

https://www.kaggle.com/valencar/sentiment -classification-of-tweets

* Tools/Techniques/Libraries :
  + **Tools :** 
    - R Language
    - RStudio
  + **Techniques :** 
    - Data Preprocessing
    - Data Visualization
    - Machine Learning
    - Geographical plotting
    - Twitter Crawling
* **Libraries :** 
  + Rtexttools :
    - RTextTools is a machine learning package for automatic text classification.
  + E1071 :
    - Functions for latent class analysis, short time Fourier transform, fuzzy clustering, support vector machines, shortest path computation, bagged clustering, naive Bayes classifier
  + Dplyr :
    - dplyr is an R package for working with structured data both in and outside of R. dplyr makes data manipulation for R users.
  + Tidytext :
    - In this package, we provide functions and supporting data sets to allow conversion of text to and from tidy formats and to switch seamlessly between tidy tools and existing text mining packages.
  + Nbclust :
    - NbClust package provides 30 indices for determining the number of clusters and proposes to the user the best clustering scheme from the different results obtained by varying all combinations of the number of clusters, distance measures, and clustering methods.
  + Sentimentr:
    - Use for calculate sentiment value of a tweet. sentimentr is designed to quickly calculate text polarity sentiment at the sentence level and optionally aggregate by rows or grouping variables.
  + Tokenizers:
    - Convert natural language text into tokens. It includes tokenizers for shingled n-grams, skip n-grams, words, word stems, sentences, paragraphs, characters, shingled characters, lines, tweets, Penn Treebank, and regular expressions, as well as functions for counting characters, words, and sentences, and a function for splitting longer texts into separate documents, each with the same number of words.
  + Stopwords:
    - R package providing “one-stop shopping” for stopword lists in R, for multiple languages and sources. No longer should text analysis or NLP packages bake in their own stopword lists or functions, since this package can accomodate them all, and is easily extended.
  + snowballC :
    - An R interface to the C libstemmer library that implements Porter's word stemming algorithm for collapsing words to a common root to aid comparison of vocabulary.
  + Wordcloud :
    - Use for generating pretty word clouds.
  + Rcolorbrewer:
    - Provides color schemes for maps (and other graphics) designed by Cynthia Brewer.
  + Stringr:
    - Character manipulation, Whitespace tools to add, remove, and manipulate whitespace, Locale sensitive operations, Pattern matching functions.
  + Caret:
    - The caret package (short for **Classification And REgression Training**) is a set of functions that attempt to streamline the process for creating predictive models. The package contains tools for data splitting.
  + Tm :
    - Text mining package
  + Rpart
    - Recursive Partitioning and Regression Trees
    - Recursive partitioning for classification, regression and survival trees. An implementation of most of the functionality of the 1984 book by Breiman, Friedman, Olshen, and Stone
  + Rpart.plot
    - Plot an [rpart](https://www.rdocumentation.org/link/rpart?package=rpart.plot&version=3.0.7) model, automatically tailoring the plot for the model's response type. This function is a simplified front-end to [prp](https://www.rdocumentation.org/link/prp?package=rpart.plot&version=3.0.7), with only the most useful arguments of that function, and with different defaults for some of the arguments.
  + Knitr:
    - **Knitr** is an engine for dynamic report generation with R. It is a package in the statistical programming language R that enables the integration of R code into LaTeX, LyX, HTML, Markdown, AsciiDoc, and reStructuredText documents.
  + Magrittr:
    - The **magrittr** (to be pronounced with a sophisticated French accent) is a package with two aims: to decrease development time and to improve readability and maintainability of code.
  + NbClust:
    - NbClust package provides 30 indices for determining the number of clusters and proposes to the user the best clustering scheme from the different results obtained by varying all combinations of a number of clusters, distance measures, and clustering methods.
  + Cluster:
    - Methods for Cluster analysis. Much extended the original from Peter Rousseeuw, Anja Struyf, and Mia Hubert, based on Kaufman and Rousseeuw (1990) "Finding Groups in Data".
  + Factoextra:
    - Provides some easy-to-use functions to extract and visualize the output of multivariate data analyses, including 'PCA' (Principal Component Analysis), 'CA' (Correspondence Analysis), 'MCA' (Multiple Correspondence Analysis), 'FAMD' (Factor Analysis of Mixed Data), 'MFA' (Multiple Factor Analysis) and 'HMFA' (Hierarchical Multiple Factor Analysis) functions from different R packages. It contains also functions for simplifying some clustering analysis steps and provides 'ggplot2' - based elegant data visualization.
  + Class:
    - Various functions for classification, including k-nearest neighbor, Learning Vector Quantization and Self-Organizing Maps.
  + Ggmap:
    - A collection of functions to visualize spatial data and models on top of static maps from various online sources (e.g Google Maps and Stamen Maps). It includes tools common to those tasks, including functions for geolocation and routing.
  + Maptools:
    - Set of tools for manipulating geographic data. It includes binary access to 'GSHHG' shoreline files. The package also provides interface wrappers for exchanging spatial objects with packages such as 'PBSmapping', 'spatstat', 'maps', 'RArcInfo', and others.
  + Maps:
    - Display of maps. Projection code and larger maps are in separate packages ('mapproj' and 'mapdata').
  + randomForest:
    - **R** - **Random Forest**. In the **random forest** approach, a large number of **decision** trees are created. Every observation is fed into every **decision** tree. The most common outcome for each observation is used as the final output.
  + Httr:
    - The aim of httr is to provide a wrapper for the curl package, customized to the demands of modern web APIs. ... Response content is available with content() as a raw vector ( as = "raw" ), a character vector ( as = "text" ), or parsed into an R object ( as = "parsed" ), currently for HTML, XML, JSON, png and jpeg .
  + Ggplot2:
    - ggplot2 allows you to create graphs that represent both univariate and multivariate numerical and categorical data in a straightforward manner.
  + twitteR:
    - provides access to the Twitter API. Most functionality of the API is supported, with a bias towards API calls that are more useful in data analysis as opposed to daily interaction.
  + Shiny:
    - makes it easy to build interactive web apps straight from R. You can host standalone apps on a webpage or embed them in R Markdown documents or build dashboards.
* **Methods and Algorithms :** 
  + KMeans
    - ***k*-means clustering** is a method of [vector quantization](https://en.wikipedia.org/wiki/Vector_quantization), originally from [signal processing](https://en.wikipedia.org/wiki/Signal_processing), that is popular for [cluster analysis](https://en.wikipedia.org/wiki/Cluster_analysis) in [data mining](https://en.wikipedia.org/wiki/Data_mining). *k*-means clustering aims to [partition](https://en.wikipedia.org/wiki/Partition_of_a_set) *n* observations into *k* clusters in which each observation belongs to the [cluster](https://en.wikipedia.org/wiki/Cluster_(statistics)) with the nearest [mean](https://en.wikipedia.org/wiki/Mean), serving as a prototype of the cluster.
    - Clustering is done on the basis of three parameters:
      * Latitude
      * Longitude
      * sentiment
    - After preprocessing the data, we used the NbClust package to find out the optimal number of cluster using the elbow method.
    - Elbow was found at k=6 and k=26
    - Using proper analysis, the optimal number of clusters was found to be 26.
    - Using kmeans function of the NbClust package, proper clustering of 855 data values is done.
    - Using the factorextra package the clusters are plotted.
    - On further analysis of the clusters formed and the processing of the data received, we calculated the number of data points in each cluster and found the following extra parameters:
      * Number of positive points in the cluster
      * Number of negative points in the cluster
      * Number of neutral points in the cluster
      * Coordinate set of all the points of the cluster
      * Total points in the cluster
      * Mean coordinates of the cluster
    - Further geographic plotting of the coordinates was done using the maptools and maps package for R.
  + Naive Bayes
    - **Naive Bayes classifiers** are a family of simple "[probabilistic classifiers](https://en.wikipedia.org/wiki/Probabilistic_classifier)" based on applying [Bayes' theorem](https://en.wikipedia.org/wiki/Bayes%27_theorem) with strong (naive) [independence](https://en.wikipedia.org/wiki/Statistical_independence) assumptions between the features. Naive Bayes has been studied extensively since the 1960s. It was introduced (though not under that name) into the [text retrieval](https://en.wikipedia.org/wiki/Information_retrieval) community in the early 1960s,[[1]](https://en.wikipedia.org/wiki/Naive_Bayes_classifier#cite_note-1) and remains a popular (baseline) method for [text categorization](https://en.wikipedia.org/wiki/Text_categorization), the problem of judging documents as belonging to one category or the other (such as [spam or legitimate](https://en.wikipedia.org/wiki/Spam_filtering), sports or politics, etc.) with [word frequencies](https://en.wikipedia.org/wiki/Bag_of_words) as the features.
    - Finding the required attributes from the dataset
    - Preprocessing the tweets to remove
      * Special characters
      * Punctuations
      * Endline characters
      * Converting the tweets to lower case
      * Substitutions of various special words
    - Creating the training dataset of 6000 rows and testing dataset of 700 rows
    - Creating the sparse matrix of all the words versus the train data tweets
      * Sparsity = 48000+/ 3cr 80lkh
    - Creating the sparse matrix of all the words versus the test data tweets
      * Sparsity = 1120+/2100+ = 60%(approx)
    - Predicting the results and finding the accuracy
    - Accuracy of about 81%
    - Dumping the model for future use
  + Decision Tree
    - A **decision tree** is a decision support tool that uses a tree-like graph or model of decisions and their possible consequences, including chance event outcomes, resource costs, and utility. It is one way to display an algorithm that only contains conditional control statements.
    - Required attributes of the target dataset were:
      * Sentiment value
      * Airline
      * Timezone
    - Due to the discrete nature of sentiment value for all the data points, it was unusable yet most important for the classification.
    - To resolve the problem we provided the data points 5 different classes based on their sentiment values:
      * A: [-100 , -50)
      * B: [-50, 0)
      * C: 0
      * D: (0, 50]
      * E: (50, 100]
    - New levels of the decision tree:
      * Range
      * Airline
      * Timezone
    - Due to less computation power, training was done on 450 rows for now and testing over 150 rows.
    - So as to remove the “no level found error” while testing, we filtered the test data using the unique values of the test data
    - On testing a total accuracy of about 95%.
  + Random Forest
    - **Random forests** or **random decision forests** are an [ensemble learning](https://en.wikipedia.org/wiki/Ensemble_learning) method for [classification](https://en.wikipedia.org/wiki/Statistical_classification), [regression](https://en.wikipedia.org/wiki/Regression_analysis) and other tasks that operates by constructing a multitude of [decision trees](https://en.wikipedia.org/wiki/Decision_tree_learning) at training time and outputting the class that is the [mode](https://en.wikipedia.org/wiki/Mode_(statistics)) of the classes (classification) or mean prediction (regression) of the individual trees.[[1]](https://en.wikipedia.org/wiki/Random_forest#cite_note-ho1995-1)[[2]](https://en.wikipedia.org/wiki/Random_forest#cite_note-ho1998-2) Random decision forests correct for decision trees' habit of [overfitting](https://en.wikipedia.org/wiki/Overfitting) to their [training set](https://en.wikipedia.org/wiki/Test_set).
    - Same preprocessing of the data as done in the decision tree
    - Creating the ensemble of 100 decision trees
    - Predicting and finding the accuracy of the model formed
    - Accuracy of about % obtained
    - Dumping the model for future use.
  + Word Cloud
    - A word cloud is a popular visualization of words typically associated with Internet keywords and text data. They are most commonly used to highlight popular or trending terms based on the frequency of use and prominence. A word cloud is a beautiful, informative image that communicates much in a single glance.
    - First, we selected the target attribute i.e tweets from the dataset
    - Steps involved in the preprocessing of the tweets obtained were:
      * Converting tweets to plain text
      * Converting to lower case
      * Removing numericals from tweets
      * Removing English stopwords
      * Removing punctuations
      * Stripping whitespaces
      * Stemming the document
    - Generating the wordcloud using the wordcloud package
    - Analyzing the most frequent words from the wordcloud for proper further work.

# Experimental Outcomes :

* **Experimental Values :** 
  + **KMeans:**
    - Number of clusters: 26
    - Size of the clusters:
      * 19, 16, 66, 11, 32, 31, 14, 18, 16, 35, 17, 25, 60, 50, 28, 39, 21, 35, 15, 9, 50, 13, 57, 74, 53, 51
    - Cluster Means:

|  |  |
| --- | --- |
| **LATITUDE** | **LONGITUDE** |
| 33.11794 | -116.4258 |
| 40.43057 | -117.3423 |
| 33.92034 | -117.0313 |
| 21.22247 | -117.0313 |
| 22.97069 | -78.95089 |
| 38.97358 | -79.67756 |
| 4.029701 | 105.5749 |
| 40.8594 | -73.87987 |
| 46.28686 | -117.9593 |
| 40.67832 | -74.1902 |
| 52.40635 | 1.388745 |
| 42.51464 | -72.09149 |
| 38.67979 | -113.299 |
| 40.65391 | -73.81336 |
| 34.89202 | -117.9691 |
| 32.89045 | -88.93558 |
| 26.24224 | -85.37158 |
| 39.66238 | -75.84788 |
| 41.20251 | -111.8531 |
| 42.60943 | -89.36921 |
| 41.50657 | -86.35637 |
| 41.698 | -88.9608 |
| 35.03239 | -83.25523 |
| 30.70627 | -93.45472 |
| 40.95409 | -78.6516 |
| 39.56214 | -76.16754 |

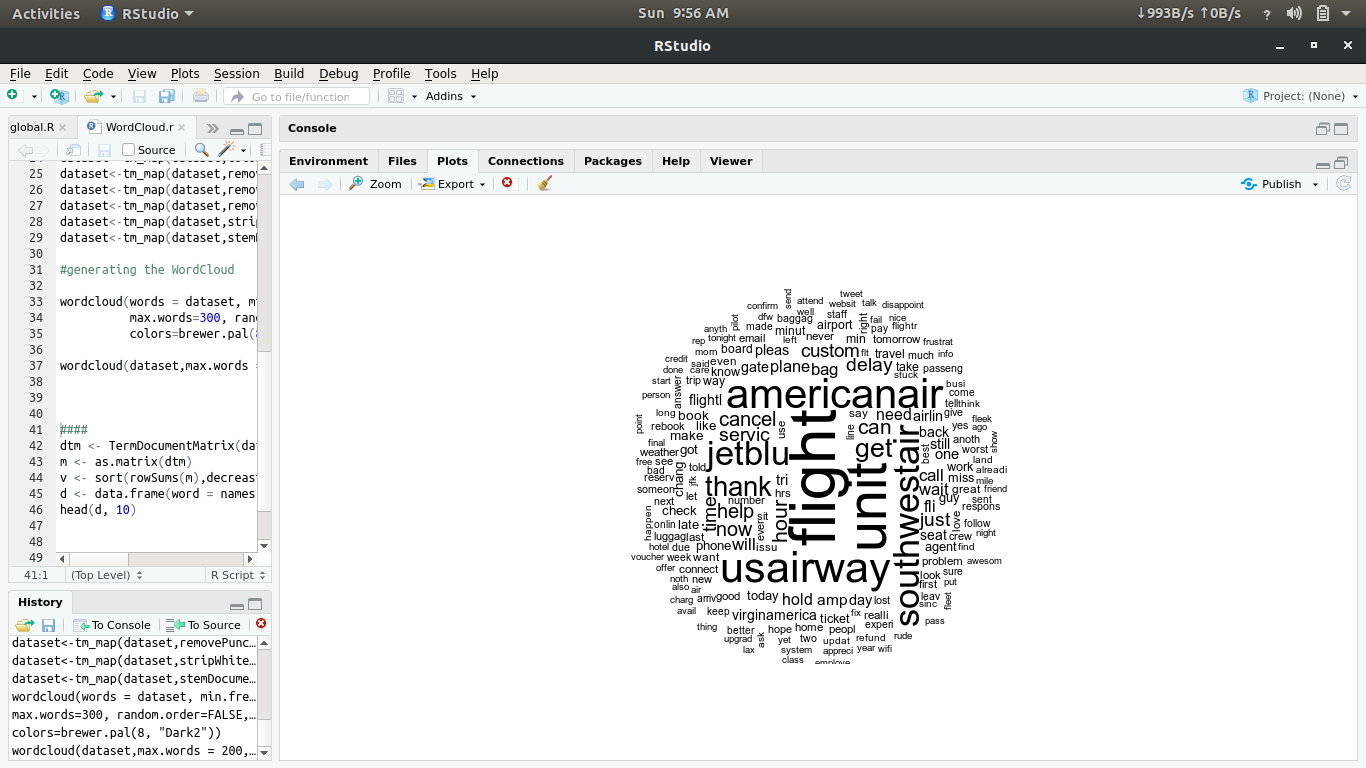
* + Naive Bayes:
    - Training Set Size : 4500
    - Test Set Size : 700
    - Accuracy: 81.86%
    - Confusion Matrix:

|  |  |  |  |
| --- | --- | --- | --- |
|  | NEGATIVE | NEUTRAL | POSITIVE |
| NEGATIVE | 0 | 0 | 0 |
| NEUTRAL | 127 | 149 | 0 |
| POSITIVE | 0 | 0 | 424 |

* + Decision Tree :
    - Training Set Size: 500
    - Test Set Size: 150
    - Accuracy: 42.3%
  + Random Forest:
    - Training Set Size: 5513
    - Test Set Size: 1834
    - Accuracy: 66.83%
  + WordCloud:
    - Dataset size: 14641
    - Most used words :
      * Flight- 4839 occurrences
      * Unit- 4127 occurrences
      * Usairways- 3040 occurrences
      * Americanair- 2941 occurrences
      * Southwestair- 2445 occurrences

# Results:

* Word Cloud:



# 

# horizontal line

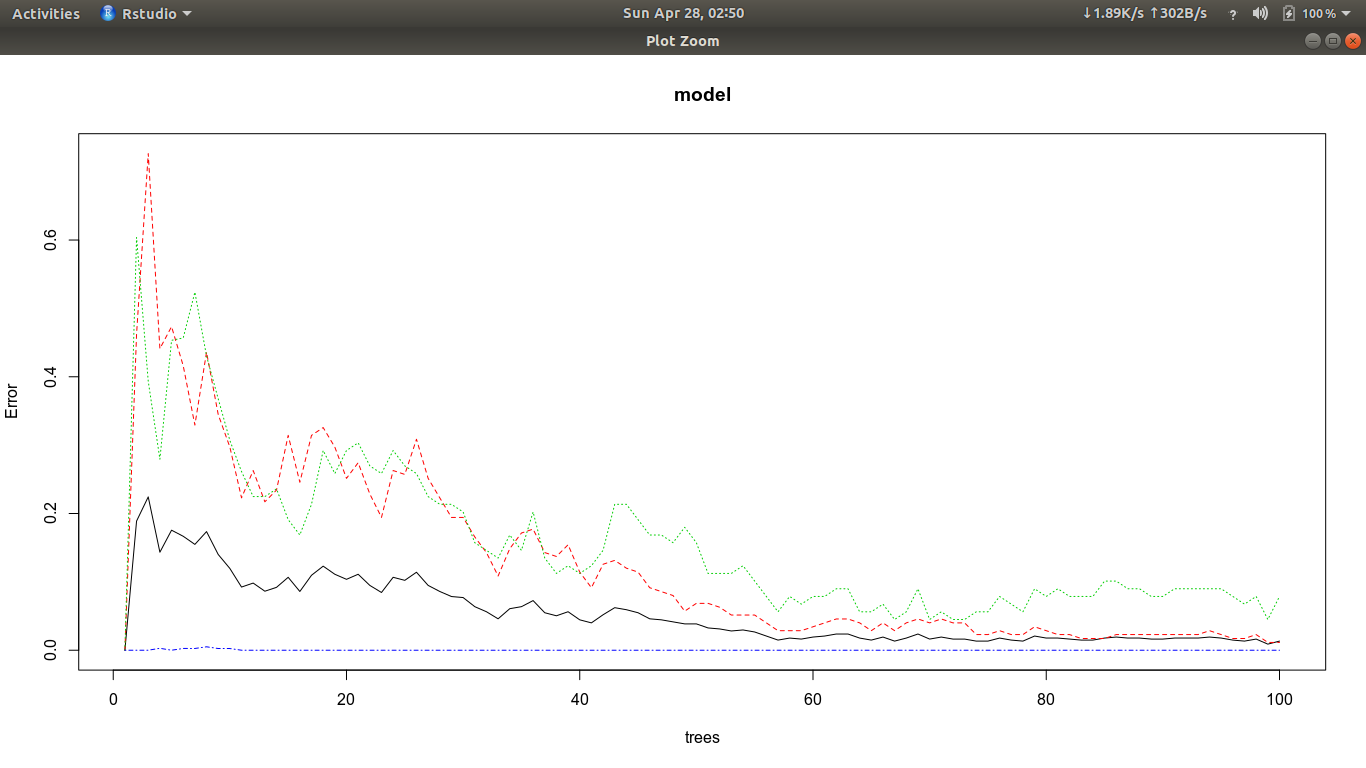
# 

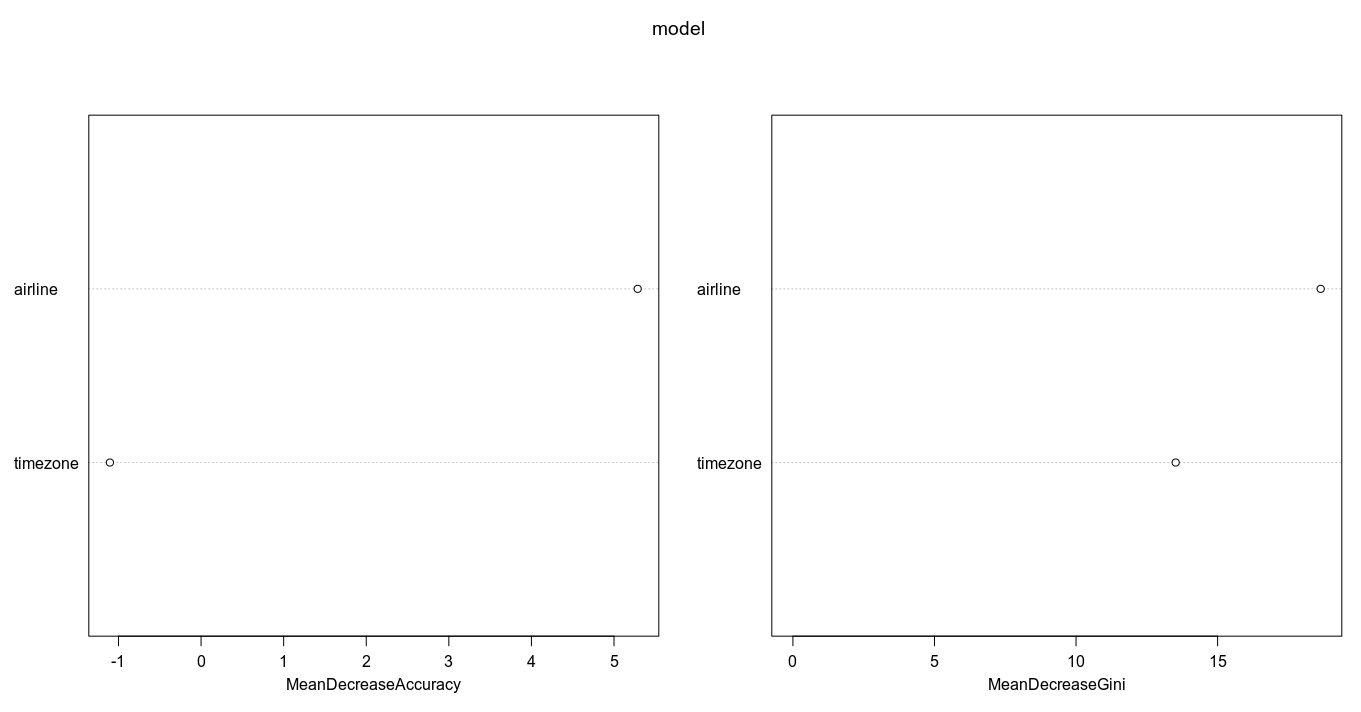
* Decision Tree



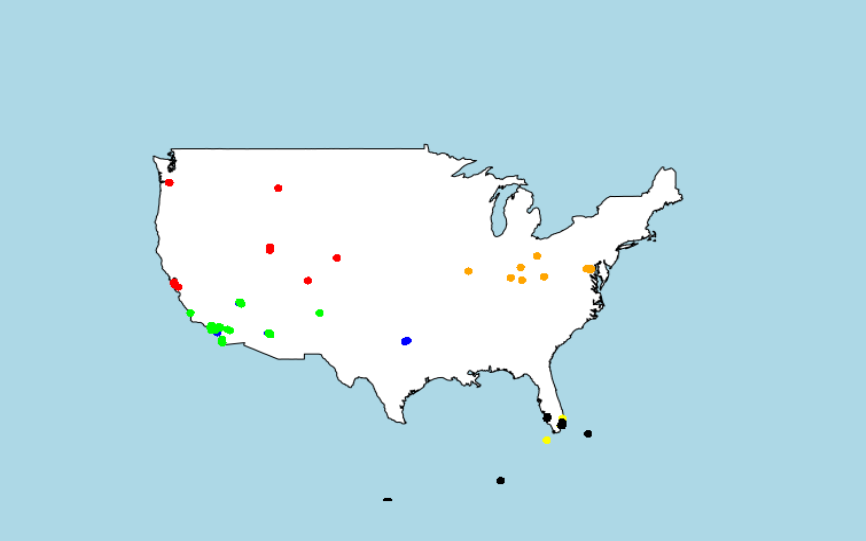
# 

* Random forest





* K-Means



# 

# Code Screenshots:

## Calculate Sentiment and Sentiment Values

## Formation of Word Cloud

## Fetching live tweets from Twitter

## Naive Bayes Classification

## Preprocessing Data for K-means

## K -Means Clustering

## Decision Tree Classification

## Preprocessing Data for Decision Tree

## Random Forest Classification

## Shiny App for Naive Bayes

## Global file for Shiny App for Naive Bayes

## Preprocessing the initial data for adding additional attributes

## Filtering Test data for decision tree

## Shiny App for Word Cloud

## Shiny App for Naive Bayes

# Conclusions:

* Conclusive Remarks :
  + Business Analysis Using Demographic Data Modelling and Visualisation is a project which mainly focuses on the techniques of Data Preprocessing, Data Visualization, and Machine Learning algorithms
* Field and techniques worked upon :
* Data Preprocessing:
  + Dataset collection
  + Data refining (Noise reduction, NA values etc)
  + Data filtration
  + Data generation based on required attributes
* Data Visualization:
  + Plots
  + Histograms
  + Geographical Plotting
  + Clusters
  + Elbow graph
  + Decision Tree
* Machine Learning:
  + KMeans
  + Random Forest
  + Naive Bayes
  + Decision Tree
  + Word Cloud
* Future Aspect :
  + Future Work Directions :
    - This project upon worked ahead can lead to the following achievements :
      * Making the prediction in real time.
      * Feeding the data into several models leading to analysis and prediction.
      * Use of database to maintain the records.
      * Database handling platforms like MongoDB can be used in order to store the data along with the results for the future reference.
      * Records available of the previous findings might be useful in witnessing the pattern followed by the phenomenon. By phenomenon here, we are referring to several different situations like Tsunami, Earthquake, Flood, Diseases, etc.

# References

* [1] M. Sarnovsky, P. Butka, A. Huzvarova, “Twitter data analysis and visualizations using the R language on top of the Hadoop platform”, IEEE 15th International Symposium on Applied Machine Intelligence and Informatics, vol. 15, pp. 327-331, 2017.
* [2] M. Bashir, R. Kusumaningrum, “Sentiment Analysis Using Latent Dirichlet Allocation and Topic Polarity Wordcloud Visualization”, 2017 Fifth International Conference on Information and Communication Technology (ICoICT), vol. 5, 2017.
* [3] N. Garg, R. Rani, “Analysis and Visualization of Twitter Data using k-means Clustering”, International Conference on Intelligent Computing and Control Systems ICICCS 2017, 2017.
* [4] P. Manivannan, Dr. P. Isakki, Dengue Fever Prediction Using K-Means Clustering Algorithm”, 2017 IEEE INTERNATIONAL CONFERENCE ON INTELLIGENT TECHNIQUES IN CONTROL, OPTIMIZATION AND SIGNAL PROCESSING, 2017.
* [5] Introduction to visualizing spatial data in R Robin Lovelace (R.Lovelace@leeds.ac.uk), James Cheshire, Rachel Oldroyd, and others 2017-03-23.
* [6] Wickham, H. (2014). Tidy data. The Journal of Statistical Software, 14(5), Retrieved from http://www. jstatsoft.org/v59/i10.
* [7] Cheshire, J., & Lovelace, R. (2015). Spatial data visualization with R. In C. Brunsdon & A. Singleton (Eds.), Geocomputation (pp. 1–14). SAGE Publications.