Network Provider Recommender System

Aniruddh Gupta
Dept. of Computer Science and Information Systems
BITS Pilani, Hyderabad Campus
Hyderabad, India
f20170149@hyderabad.bits-pilani.ac.in

Anushray Mathur
Dept. of Computer Science and Information Systems
BITS Pilani, Hyderabad Campus
Hyderabad, India
f20171570@hyderabad.bits-pilani.ac.in

Anuj Kharbanda
Dept. of Computer Science and Information Systems
BITS Pilani, Hyderabad Campus
Hyderabad, India
f20171508@hyderabad.bits-pilani.ac.in

Vidish Bharadwaj
Dept. of Mechanical Engineering
BITS Pilani, Hyderabad Campus
Hyderabad, India
f20171391@hyderabad.bits-pilani.ac.in

***Abstract*—This is the report for the project of the course Data Mining (CS F415) by Group No. 16. The report contains an introduction to the project, dataset and the operations performed on the dataset.**

I. Problem Definition

It is a widely known fact that cellular communication companies differ from each other considerably in terms of voice call quality when pitted against one another under the same circumstances. This difference can arise due to several reasons like Network Blind Spots, low bandwidth allocation due to loaded cell sites, sparsely located network towers etc. To the average person selecting a network carrier, this is a problem too complex to be solved manually, hence the need for a recommender system that takes into account various external factors and gives the most reliable network provider for the given case as output. System will recommend the best possible network provider by seeking requirements from the user such as their location, network type they want to use and use type. Predict the expected call drop quality based on network operator, use type, network type and location coordinates. We will also find relevant association rule between the attributes of the data set.

II. Data Description

The chosen dataset is taken directly from the archives of Open Govt. Data (OGD) Platform, India.

This data set was recorded monthly at various locations throughout India consisting of around 25,000 entries (per month) with the following attributes:

1. Operator
2. Use Type (In/Out/Travelling)
3. Network Type (2G/3G/4G)
4. Call Quality Rating (out of 5)
5. Call Drop Category
6. Latitude
7. Longitude
8. State Name

Our dataset includes the data for the months of April 2018 to March 2019 and all the operations done have been conducted on the data of each month individually as well as the combined data of the 12 months.

III. Data Preprocessing

To make the dataset more efficient to use and get it in a useful format, the following operations were performed.

A. Data Cleaning

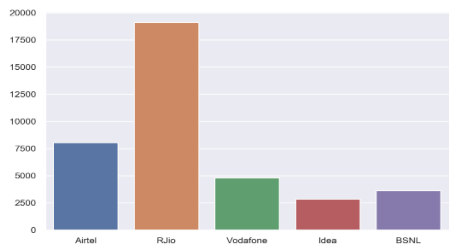
- Thousands of records for each month did not contain the state name but had the values for latitude and longitude. To make this data more usable and readable, the missing data of state name was filled up using a script which used the coordinates to fill up the state name.
- Some records in the dataset of each month had no value for state name, latitude and longitude, giving no information about the location of the record thus rendering our later operations difficult to perform. Since the number of such entries were very low as compared to the total volume of the data, these entries were removed from the dataset.

B. Data Transformation

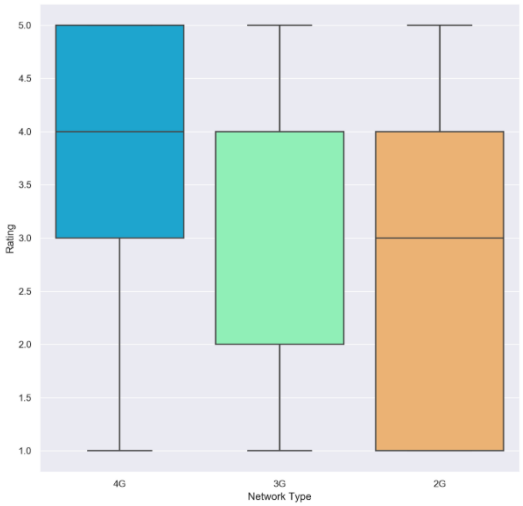
- Label Encoding – The initial dataset had the values Call Dropped, Poor Voice Quality and Satisfactory for the attribute Call Drop Category. Label Encoding was used on this attribute to transform the values from Call Dropped, Poor Voice Quality and Satisfactory to 0, 1 and 2 respectively. This was done to create an ordinal relationship between the values of the attribute and make the later operations easier to perform.

C. Data Visualization

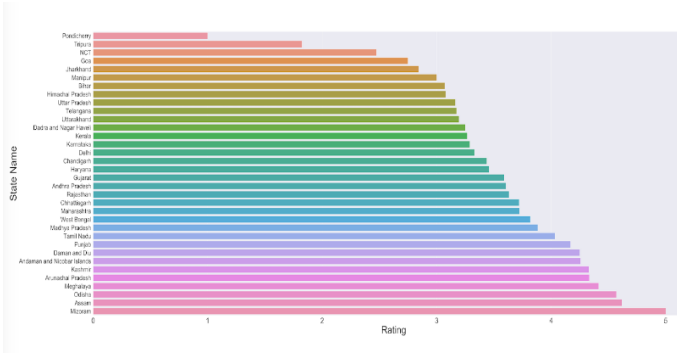
- The following plots were created during the course of data visualization of the dataset for each of the 12 months' data as well as for the combined data: -
  1. Distribution Plot – 6 Distribution plots were created namely Rating, Call Drop category, Network Type, Usage Type, States and Operator. (The distribution plot for operators has been shown below, number of records is on the y axis)



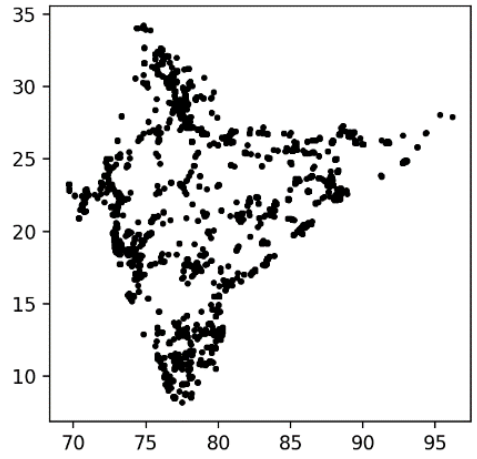
2. Box Plot – A total of 5 Box plots were created for each iteration which were Operator vs Rating, State Name vs Rating, Network Type vs Rating, Usage Type vs Rating and Call Drop Category vs Rating. (Box plot for Rating vs Network Type has been shown below)



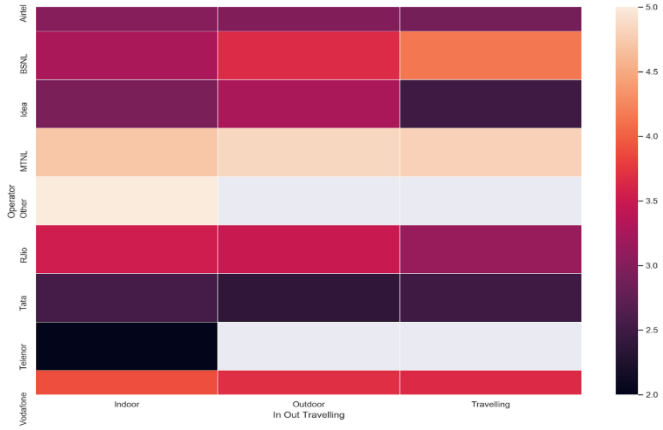
3. Bar Plot – 5 Bar Plots were created for each month which were Operator vs Call Drop Category, Operator vs Overall Rating, Operator vs Overall Call Drop Category, State vs Rating, State vs Call Drop Category. (The State Name vs Rating Bar Plot has been shown below)



4. Scatter Plot – The following scatter plot was made Latitude vs Longitude (Fig below). The x axis shows the longitude degree while y axis has the latitude degree.



5. Heat Map – The following Heat maps were created State vs Operator Based on Rating, State vs Operator Based on Call Drop Category, Operator vs Usage Type Based on Rating, Operator vs Usage Type Based on Call Drop Category, Correlation Plot. (The Operator vs In-Out Travelling Heat Map has been shown below)



D. Exploratory Data Analysis

To understand the relationship between the attributes of the dataset, certain statistical tests were performed on the dataset. The following tests were performed on the data of each of the 12 months individually as well as on the combined data of all the months: -

- Pearson Correlation – Pearson's correlation coefficient is a statistical measure of the strength of a *linear* relationship between paired data.

  Using Data Visualization (Heat Map – Correlation Plot), the only pair which might have a correlation was found to be that of Rating and Call Drop Category.

  A correlation between this pair was found in each of the 12 months' data as well as the combined data.

- Spearman Correlation – Spearman's correlation coefficient is a statistical measure of the strength of a *monotonic* relationship between paired data.

  This test too was run on the pair of attributes Rating and Call Drop Category.

This test too found a correlation between this particular pair in each of the 12 months' data as well as the combined data.

- Chi Square Test – To confirm the relationships found in the above tests, Chi Square Test was performed on each of the 12 months' data and also on the combined data.

  The Chi Square test confirmed that Rating and Call Drop Category are dependent on each other for each of the 12 months' data as well as the combined data.

IV. Data Processing Techniques

The chosen data processing techniques were applied on the pre-processed data set to get insightful results and statistical inferences about the data provided. These techniques were also deemed necessary to be able to predict the end user experience based on given attribute data for any new data entry.

The applied Data Processing Techniques were as follows:

1. **K.N.N. Classification**
2. **Association Rule Mining**
3. **Decision Tree Classification**

The process details and results of the aforementioned techniques are presented in the subsequent parts.

A. K.N.N Classification

The K.N.N. Classification model developed uses the following attributes as input to predict the call drop category:

- Operator
- In/Out/Travelling
- Network Type (2G/3G/4G)
- Location Coordinates

First the model filters the data set based on the given input and then applies K.N.N. Classification on the data set to predict the **call drop category.**

The value of K was taken from the primes between [2, 1501] based on the accuracy with which it helped to predict the target class on a small sample of testing data set. Identifying the suitable K value was a difficult task as the amount of time it takes to calculate the accuracy for each K value was huge and due to the time constraint, we had to reduce the size of the testing data set and then choose the optimal K value.

Reason for the choice of K.N.N. Classification over Clustering:
Clustering was not deemed feasible as when we plotted the training data set on the 2-D plane, no meaningful set of clusters were visible. Hence, the pursuit of Clustering was dropped in favour of going towards the K.N.N. Classification route.

B. Association Rule Mining

We used the concatenated form of the data sets for all 12 months to find the Association Rules for it. This was done in order to avoid the paucity of cases of occurrence of the rules consequently found. The location coordinates of the data set were dropped due to the continuous nature of the values.
The dataset was one-hot encoded at first, after which frequent itemsets were found. After finding the frequent itemsets, the required rules were formulated using different combinations of minimum support, minimum confidence and lift so as to find the best combinations which provided the required results.

Identifying the minimum threshold values for the support, confidence and lift was an arduous task and required considerable fine tuning. In particular, the support was kept considerably low due to the huge size of the data set (approximately 360,000 elements) and the confidence was kept high to retain the accuracy of the association rules.

C. Decision Tree and Random Forest Classification

Given all the details as input from the user, our aim was to find the best network operator for the user. This model accomplishes the same using two techniques:

- **Decision Tree**
- **Random Forest made out of many decision trees**

The **decision tree** algorithm creates a decision tree with all features (except operator) as parameters and predicts the operator. The max. depth of the decision tree is decided by trial and error method. We find the accuracy by varying maximum depth and creating a tree and then repeating the process multiple times.

It is a recursive algorithm and has 2 cases, namely, a **recursive case** (to further go deep down the decision tree) and a **base case** (to stop recursion and classify the data into categories).

1. **Base case:** The algorithm classifies data when -

   - It is pure (same class for all data points)
   - Length of remaining data is less than min_samples (to remove unnecessary computations for very minimal change in accuracy)
   - The tree's depth has reached max_depth (to remove unnecessary computations for very minimal change in accuracy)

2. **Recursive case:** In the recursive case, first it finds the potential splits in the data which can be a

combination of data from many features. If the feature is categorical then all unique values qualify as potential splits but for continuous features potential splits are the values between two unique values (averages of 2 adjacent unique values). After that we find the best split by actually splitting data for each potential split and finding the overall entropy (using entropies of both the splits). So the split with minimum overall entropy is the best split. Then we split data on that best split.

The question (node of the decision tree) is framed using the best split feature name and split value, e.g., "stn = Maharashtra". After which we recursively go on the subtree (those two splits), which are 'yes' and 'no' answers to the question respectively.

This process creates the Decision tree.

To classify data and find accuracy, the tree is traversed and at every node the question is split into feature name and feature value. Then, if our test data for that feature satisfies that question then the subtree which signifies yes answer is assigned to the answer and vice versa.
If that answer is a single value (it is leaf node) then that value is our output (classification), but if it is a dictionary then we repeat the same process till we reach the leaf node. This classifies the data. Then we compare the original class and classification to find the accuracy.

For **random forest**, we create a set of decision trees with bootstrapped data, i.e., only a percentage of data (suppose m data points) from the whole dataset and random subspace of some n (= total no. of features / 2) features among all the features. Those m data points and n features are selected randomly for every decision tree. Now the no. of decision trees is decided by trial and error method. We find the accuracy by varying the number of trees and creating forest and repeating the process multiple times. Now for classifying the data, we classify with respect to all the trees and then take the mode to find the final classification of the forest.

Initially the algorithm was run generally without giving any preference to any feature, but due to this it was taking a considerable amount of time to execute even for a single month. But, it was found that if we fix state as base criteria for the decision tree then it has to search in a smaller subset of the dataset and consequently the algorithm was now very much faster than the original one. So this was done to reduce time complexity while getting almost the same result with almost the same accuracy. So, till we match with a state at some node we recursively run the algorithm with only state name as the feature. Once we find a state, we run the algorithm on all the remaining features

V. Results and Insights

We were able to draw the following insights and conclusions from the data set given by making use of the previously mentioned data processing techniques.

A. K.N.N. Classification

Optimal K value which was used in the model was **101**. It Gave **82.03%** accuracy with 128 testing entries. Accuracy with which the model predicts the target class is **76.91±1.38 %.**

B. Association Rule Mining

The minimum support used in the model is **0.04** while the value of minimum confidence and minimum lift are **0.8** and **1.9** respectively. Using this combination, the following **12 rules** were mined:

- **['Call Dropped'] -> [1]**
- **[2] -> ['Poor Voice Quality']**
- **['4G', 'Call Dropped'] -> [1]**
- **['Indoor', 'Call Dropped'] -> [1]**
- **['4G', 2] -> ['Poor Voice Quality']**
- **['Indoor', 2] -> ['Poor Voice Quality']**
- **[4, 'RJio'] -> ['4G', 'Satisfactory']**
- **[5, 'RJio'] -> ['4G', 'Satisfactory']**
- **['RJio', 4, 'Maharashtra'] -> ['4G', 'Satisfactory']**
- **['Indoor', 4, 'RJio'] -> ['4G', 'Satisfactory']**
- **[5, 'Outdoor', 'RJio'] -> ['4G', 'Satisfactory']**
- **['Indoor', 5, 'RJio'] -> ['4G', 'Satisfactory']**

C. Decision Tree and Random Forest Classification

- The **decision tree** algorithm has an average accuracy of 0.906437 (**90.64%**) and a total accuracy (after appending all data) is 0.89493 (**89.49%**).
- The **random forest** algorithm has an average accuracy of 0.919886 (**91.99%**).

The total accuracy of the random forest wasn't calculated because of computational constraint as it took nearly a day to predict total accuracy of the decision tree and hours to predict accuracy of even one month in the random forest.

VI. References

[1] https://machinelearningmastery.com/how-to-use-correlation-to-understand-the-relationship-between-variables/

[2] https://www.statisticshowto.datasciencecentral.com/probability-and-statistics/chi-square/

[3] https://www.kaggle.com/learn/data-visualization

[4] "Introduction to Data Mining" Tan,Pang-Ning & others

[5] “Data Mining: Concepts and Techniques” Han J & Kamber M

[6] https://www.tutorialspoint.com/machine_learning_with_python/machine_learning_with_python_knn_algorithm_finding_nearest_neighbors.htm

[7] https://www.kaggle.com/datatheque/association-rules-mining-market-basket-analysis

[8] https://towardsdatascience.com/understanding-decision-trees-for-classification-python-9663d683c952

GITHUB REPOSITORY :

https://github.com/anuj-kh/Data-Mining-Project