TEAM 7

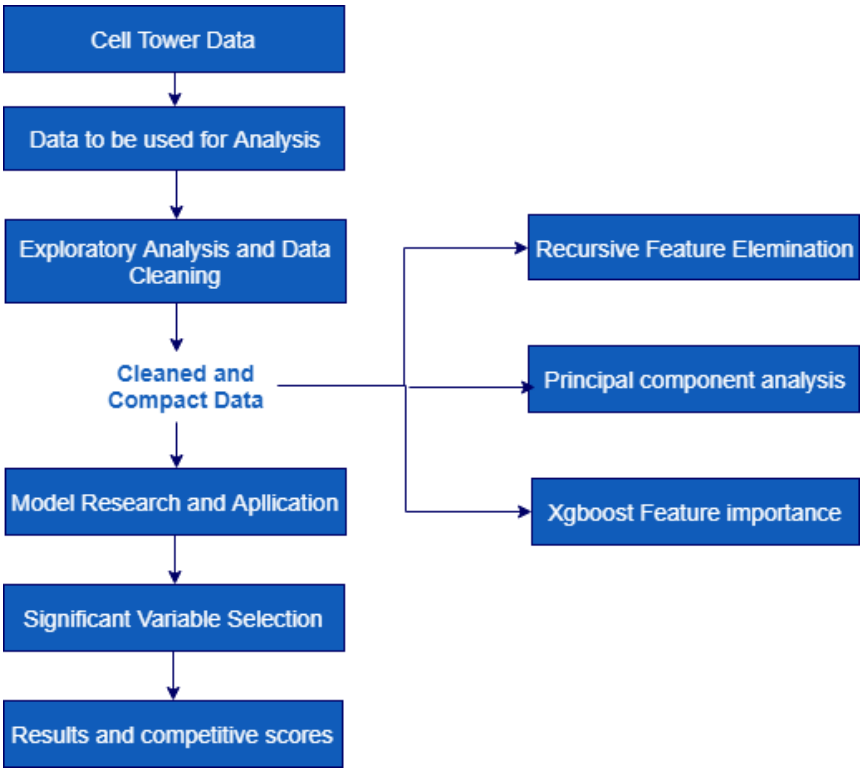
INDEX

INTRODUCTION

The report outlines the analysis performed on the cell tower statistics provided by the three network vendors namely Nokia, Ericsson and Huawei. The purpose of the analysis is to determine the type of network congestion in advance in order to take proactive actions. It has been shown that congestion, even if for smaller durations, has a negative impact on customer loyalty, especially in price-sensitive markets like the telecommunication industry. The prediction of the congestion type in advance is therefore imperative for a firm as it allows them to gain an upper hand as compared to its competitors.

The dataset given for the competition contains cell tower data usage statistics for certain time packets during the month December 2018 and the specifications of the cell tower such as beam direction, tilt, etc. This data which is primarily a merge of the Incidents Table and ESR Records, is then used for further analysing the type of congestion possible in a cell tower.

The model developed for evaluating the congestion type is utilized to design an optimal visual representation of the cell tower statistics for all the three major telecommunication firms. This visualisation exercise provides an understanding between the relation between network congestion and the statistical data at hand. The detailed flow of the approach to solve the problem is shown in the flowchart.



DATA DESCRIPTION

The subjective analysis of the variables present in the cell tower statistics for the firms Nokia, Ericsson and Huawei assist in the comprehensive analysis and study of the data with respect to the types of congestion type in the tower. The data provided for predicting the congestion type of the different cell towers consists of a merge of two datasets viz. Tower level activity data and User level activity data.

The dataset provided here is a subset of original dataset, that only has sample data for the month of December, 2018. In order to maintain discrepancy and avoid leakage of proprietary information some fields in the current dataset are anonymized; while the usage data has also been anonymously scaled or randomized by a single or constant factor.

The collected user data has been aggregated for small time buckets of duration ranging from 5 minutes to 60 minutes. While the month and year data are of no use as they are constant for the entire dataset, the day and hour data suggest the collection of data has been even for each day and the entire month.

The training data is a pretty balanced dataset containing approximately 20,000 entries for each congestion type. The user data usage is perfectly categorized into different columns and covers almost all usage types like software download, photo sharing, storage services, gaming, etc.

DATA ENGINEERING

As the dataset provided was void of any NULL values, there was no requirement of cleaning the data. The year data and the month data had to dropped as they had the same value for the entire dataset, and therefore weren't relaying any information to be used in the further steps.

In order to obtain better accuracy, we created a list of new features using the existing ones, after a proper research on what factors might affect the network congestion in a cell tower. As it was clearly evident that network congestion depended mostly on data usage, data speed and bandwidth of the channel, here are the new features we came up with:

- **day_of_week:** Feature to denote which day a date falls in a week

After exploratory analysis we observed the data usage pattern for the cell towers appeared to be cyclic after every week. Therefore, instead of going for par_day as a parameter we opted for weekday as a parameter.

$$day\_of\_week = (par\_day)\%7$$

- **hour_bucket:** Feature to group the hour parameter into different hour buckets
  As research and exploratory data analysis suggested that total data usage is varied over an entire day, we grouped the hour parameter into different buckets after applying a certain threshold for each hour bucket.

Hour Bucket 1	03:00 -09:00
Hour Bucket 2	09:00 -16:00
Hour Bucket 3	16:00 -23:00
Hour Bucket 4	24:00, 01:00, 02:00

- **avg_data_speed:** Feature containing the average data speed at which the cell tower is operating
  Network congestion occurs when the cell tower reaches its maximum allowable speed. Therefore, we calculated the average data speed for each of the customer by dividing the total data usage by minutes parameter and subscriber count parameter.

$$avg\_data\_speed = \sum(\#\#\_total\_bytes)/(1024 * 60 * par\_min * subscriber\_count)$$

- **large_data/small_data:** Features to group the data types into large and small data usage
  Network congestion is very much dependent on the bandwidth of the channel. When the user demands large chunks of data, the network becomes unable to meet the demands. We therefore classified the different data types into large data and small data depending on amount of the data consumption by applying a suitable threshold value.

Large Data	Small Data		
subscriber_count	location_services_total_bytes	marketplace_total_bytes	storage_services_total_bytes
video_total_bytes	audio_total_bytes	advertisement_total_bytes	speedtest_total_bytes
social_ntwrking_bytes	mms_total_bytes	remote_access_total_bytes	others_total_bytes
cloud_computing_total_bytes	photo_sharing_total_bytes	weather_total_bytes	web_security_total_bytes
communication_total_bytes	software_dwnld_total_bytes	voip_total_bytes	email_total_bytes
presence_total_bytes	health_total_bytes	file_sharing_total_bytes	gaming_total_bytes
web_browsing_total_bytes	media_total_bytes		

- **subscribers_per_area:** Feature to denote subscriber count in a unit area
  We also calculated the number of subscribers in a unit area of the coverage of a cell tower as with the increase of traffic in the area the probability of network congestion increases.

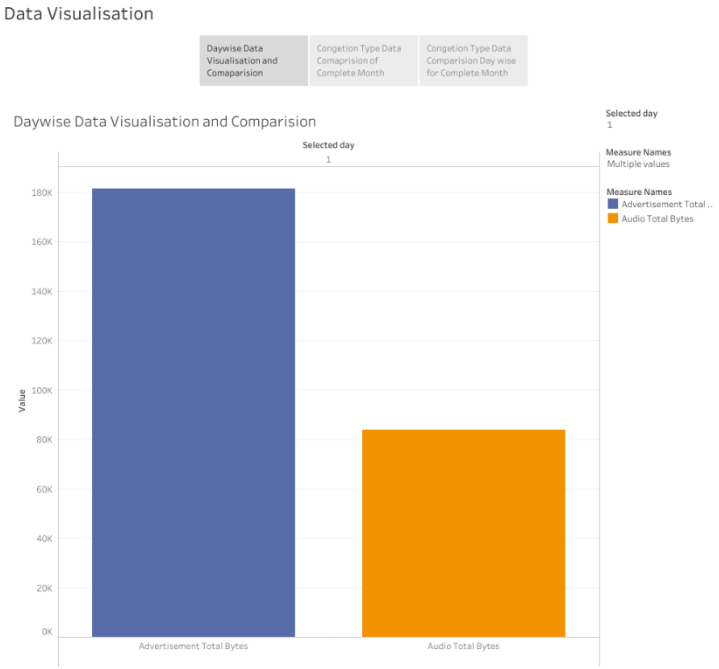
$$subscribers\_per\_area = {subscriber\_count}\Big/{(\pi * (cell_{range})^2)}$$

DATA VISUALISATION

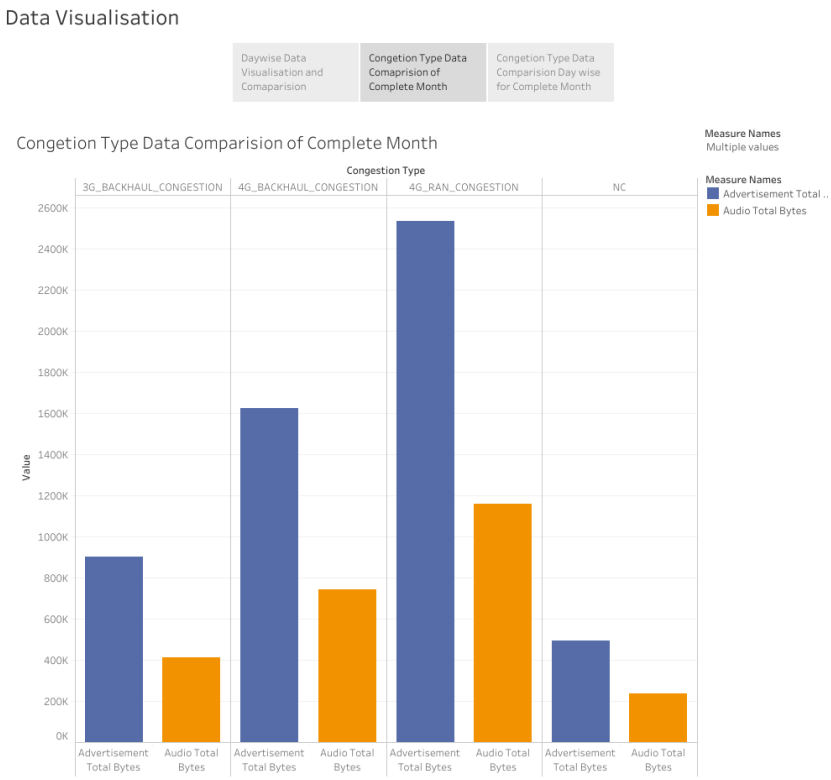
For the purpose of visualization, the Tableau Desktop has been employed which provides interactive visualization. These visualisations basically consists of 4 basic divisions.

An interactive story was created using tableau software that may be used by the vendors in the to get an idea on the dependence of various features on the cell tower congestion. Apart from getting a simple prediction from the models applied, vendors can themselves look at the generated plots to get an idea of the forces at play for causing the network congestion.

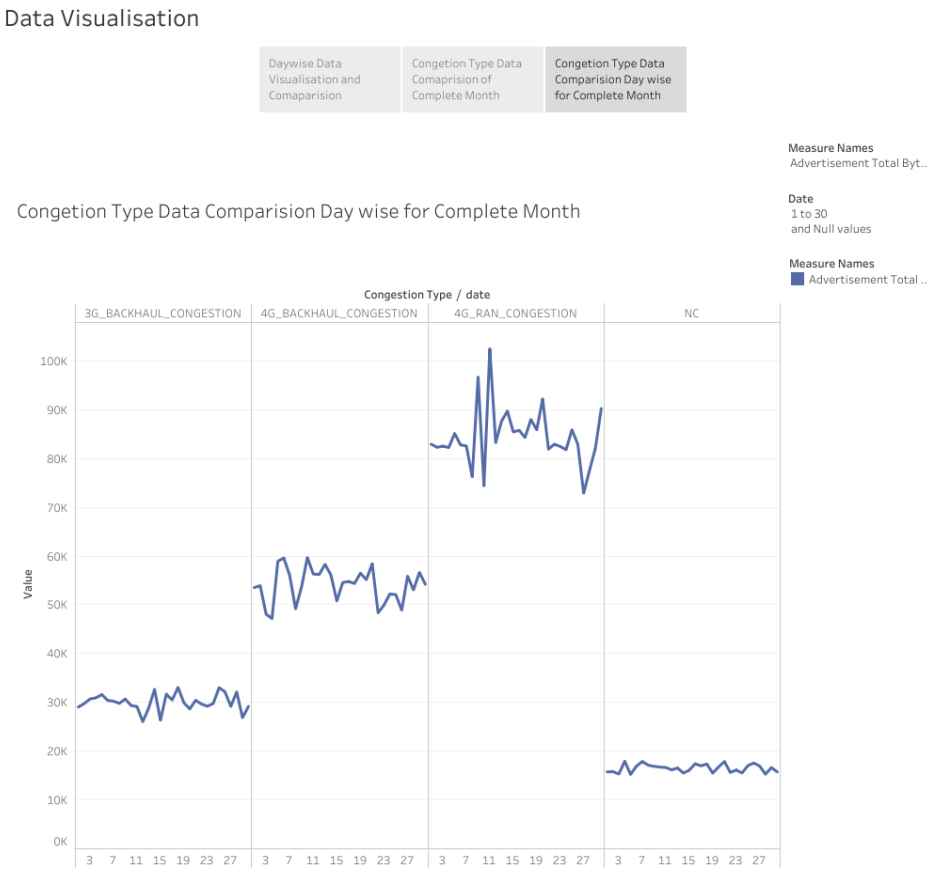
Day wise Data Visualisation and Comparison



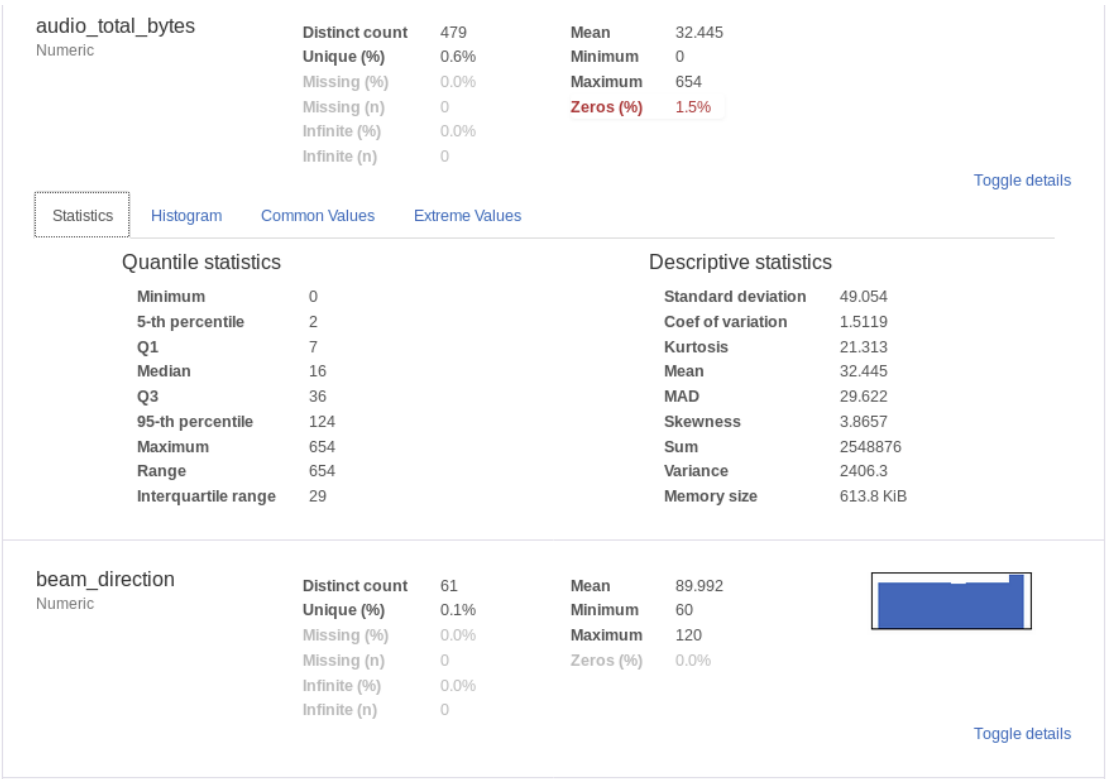
Congestion Type Data Comparison of Complete Month



Congestion Type Data Comparison Day wise for Complete Month



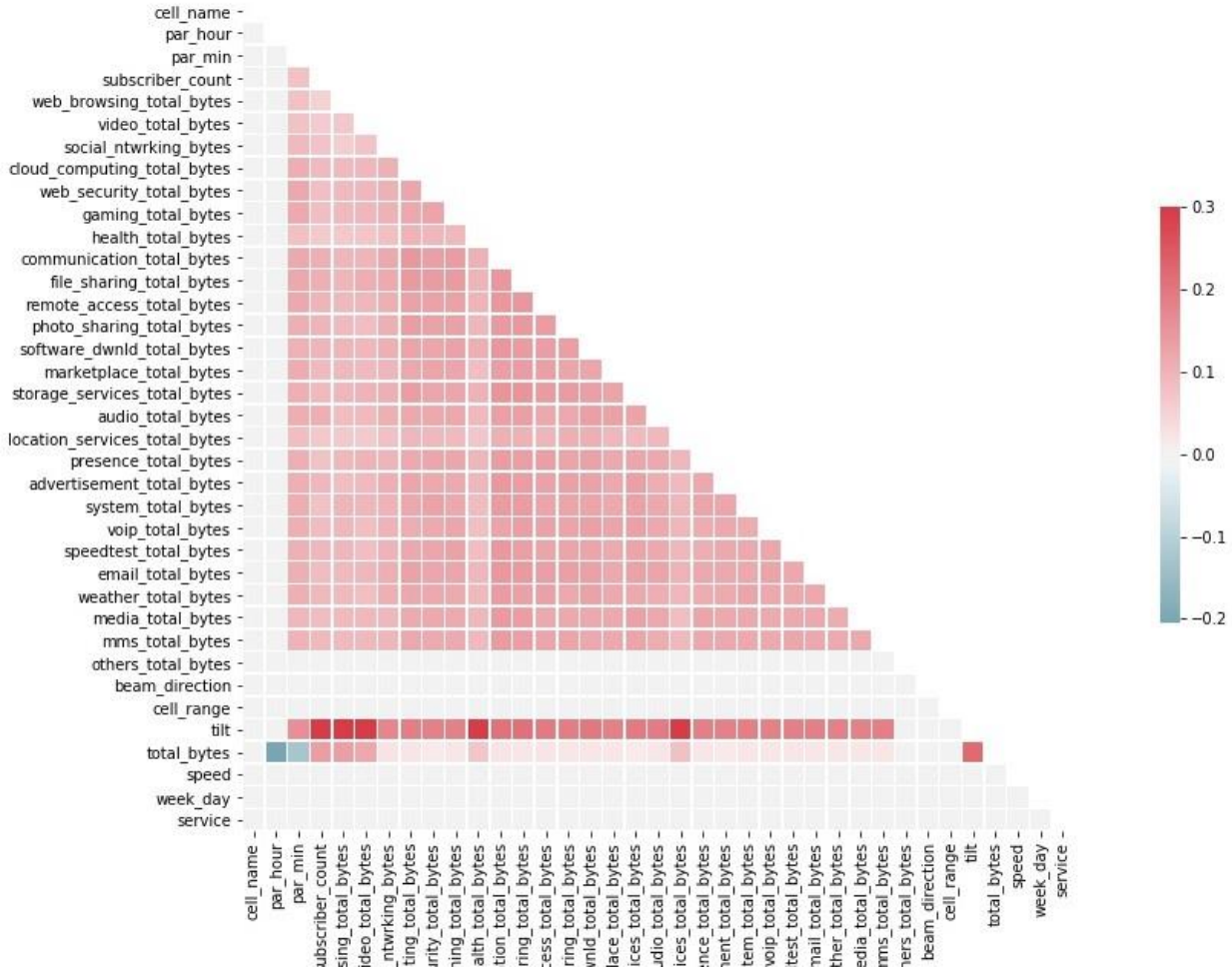
Apart from Tableau visualisations we used python pandas library to generate a complete report of all the features in the dataset. It is hosted at https://kgpresearch12.herokuapp.com/data/



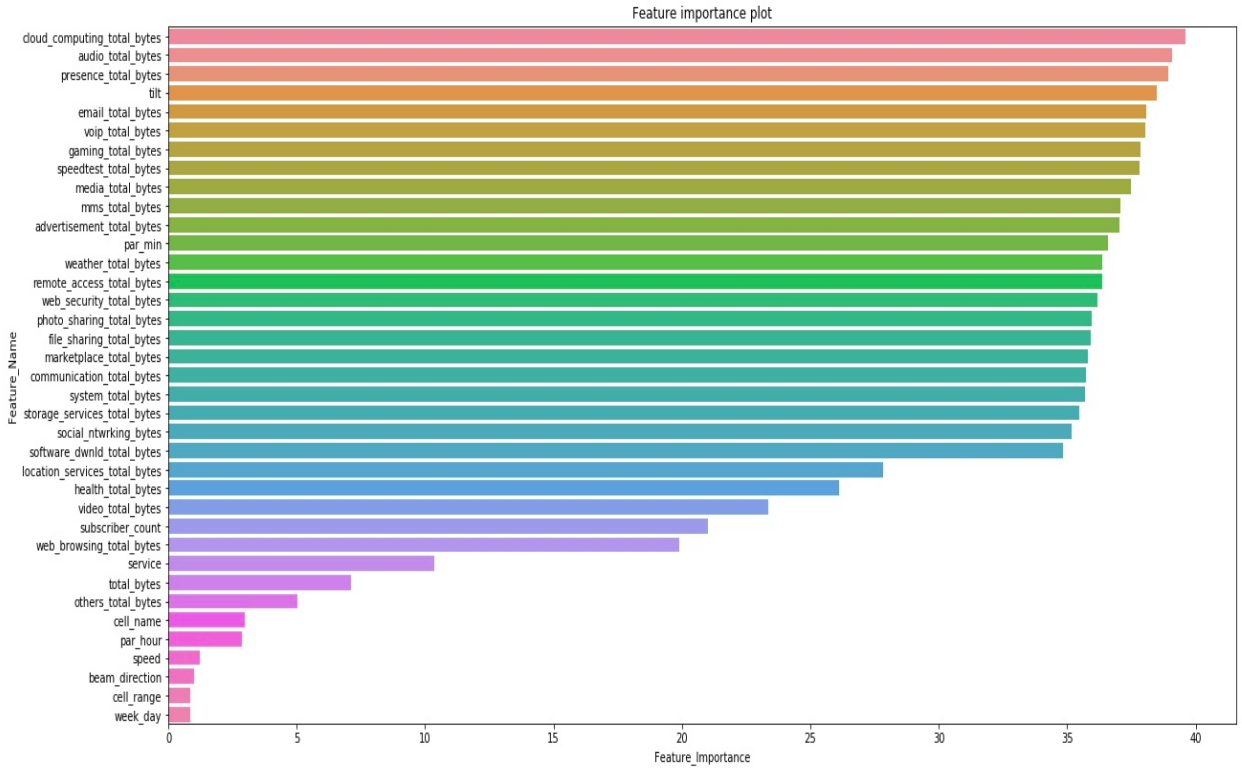
EXPLORATORY ANALYSIS

With a focus on summarizing and visualizing the important characteristics of a data set, exploratory data analysis assists in understanding the data's underlying structure and variables, developing intuition about the data set and deciding how it can be investigated with more formal statistical methods. After a detailed exploratory analysis, we gathered some significant results.

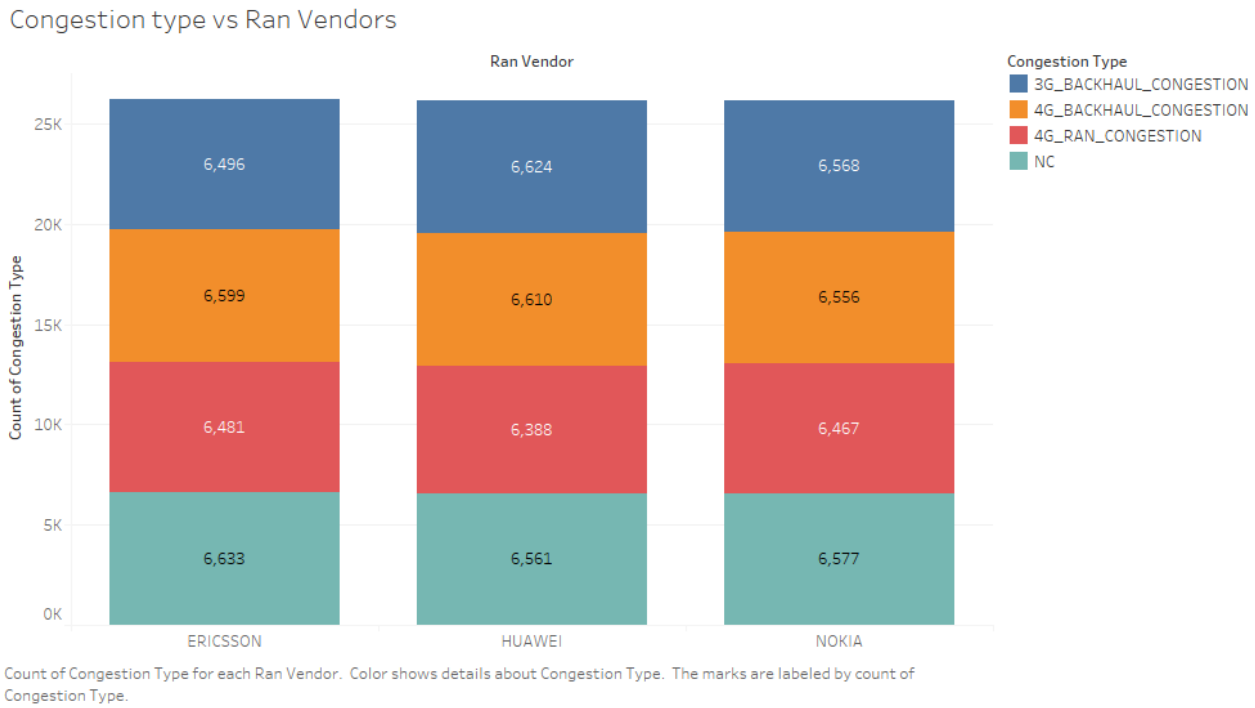
We analysed all the parameters in the cell tower data set.



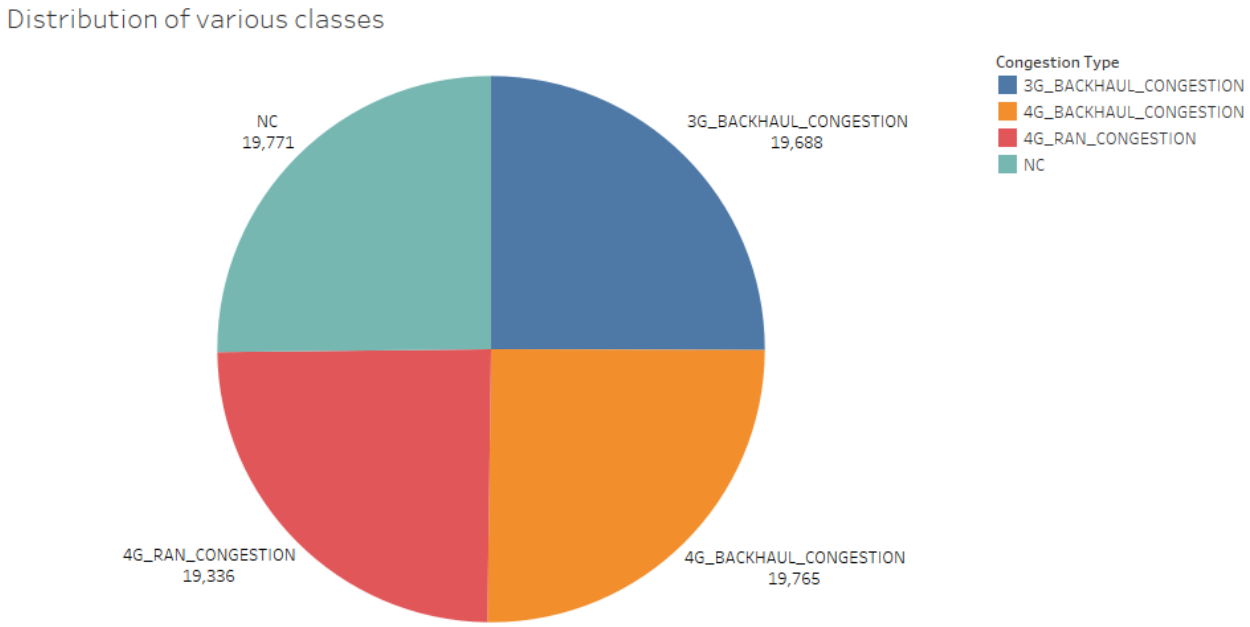
The correlation matrix was plotted in order to ensure the new features created don't overlap with each other or give erroneous results.



The feature importance plot is very much helpful in running feature analysis and choosing the features while running the machine learning models.

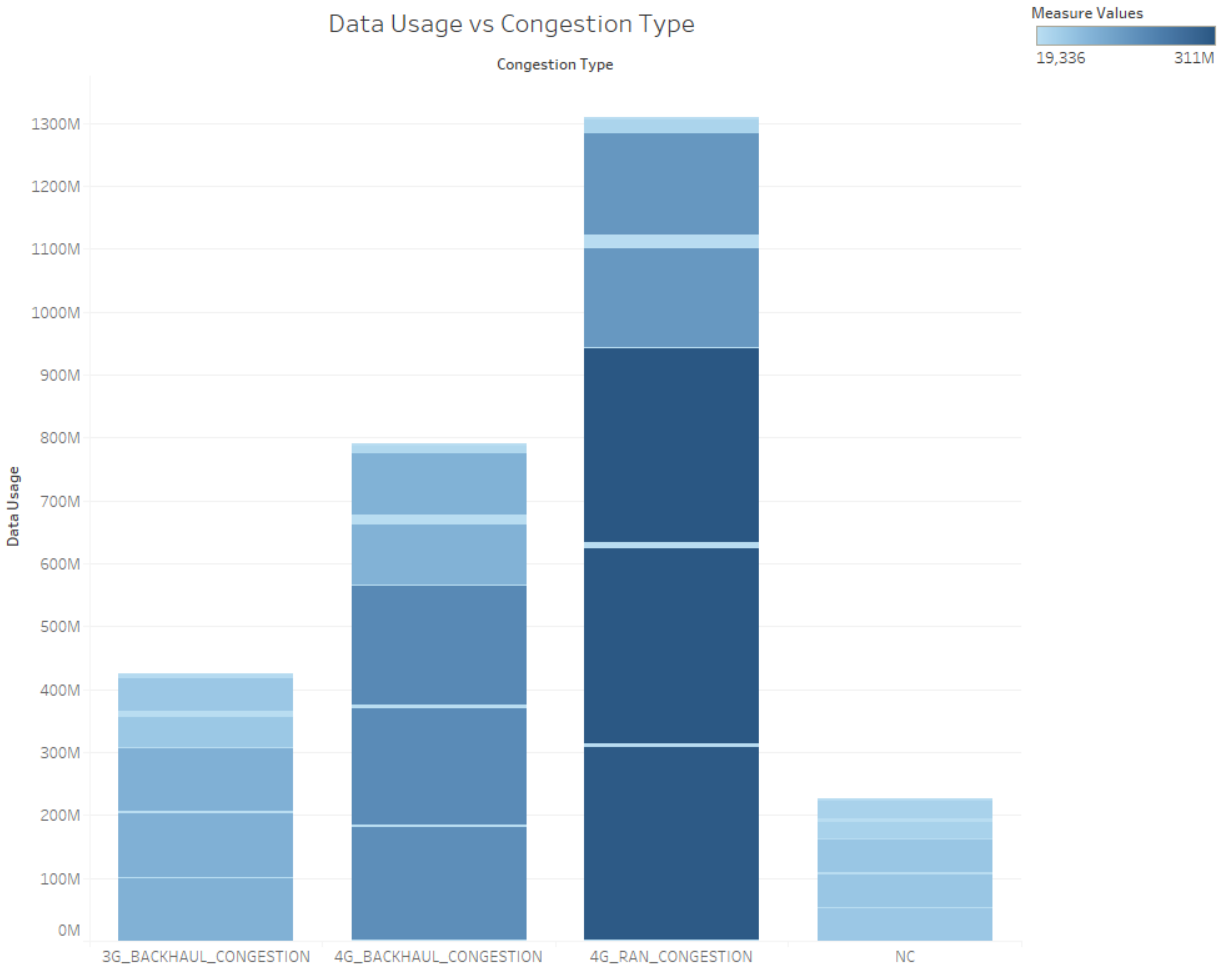


The congestion type for cell towers of each of the telecommunication firms had a uniform distribution throughout the dataset.



Congestion Type and count of Congestion Type. Color shows details about Congestion Type. Size shows count of Congestion Type. The marks are labeled by Congestion Type and count of Congestion Type.

The training dataset was perfectly balanced with almost equal number of entries for each of the Congestion Type.



From the above plot it is quite clear that the cell tower congestion is dependent on data traffic as total data consumption is highest for 4G RAN Congestion and minimum for No Congestion.

APPROACH AND MODEL

The whole exercise involving a concrete methodology to predict the congestion type for all the three telecommunication companies viz. Nokia, Ericsson and Huawei is strongly based on the structure which gives a high-level understanding of the network congestion. These network congestion parameters then become the basis for defining the probability of network congestion and ultimately gives an idea about the probable congestion type for a cell tower.

Congestion, in the context of networks, refers to a network state where a node or link carries so much data that it may deteriorate network service quality, resulting in queuing delay, frame or data packet loss and the blocking of new connections.

In the context of telecommunications industry, one of the most important issues that industry faces is network congestion. The increasing usage of data services is leading to excess usage of network resources congesting the network and deteriorating the user's data experience. It has been shown that congestion, even if for smaller durations, has a negative impact on customer loyalty, especially in price sensitive markets. To solve this problem effectively, it becomes imperative for firms to be able to predict congestion in advance and take proactive actions.

Our approach to this problem involves a two-step algorithm, in which the initial step is the detection of congestion and in the subsequent step we determine the type of congestion in the network. In the analysis performed, this algorithm has been taken into consideration after extensive research for an optimum solution. We tried our luck with the old-school single step multi-class classification problem-based approach, exhausting almost every type of machine learning algorithm at hand to classify the congestion type for each of the cell towers. Even after proper tuning and adequate feature engineering, there was a huge bias problem.

XGBoost

XGBoost is an optimized distributed gradient boosting library designed to be highly efficient, flexible and portable. It implements machine learning algorithms under the Gradient Boosting framework. Using hyperopt library we tuned the parameters of the XGboost model

Tuned parameters: 'colsample_bytree': 0.2, 'subsample': 0.6 , learning_rate =0.2, n_estimators=800,gamma=0, max_depth=2, min_child_weight=5

Matthews Correlation Coefficient: 0.7307

Support Vector Machines

A

Support Vector Machine (SVM) is a discriminative classifier formally defined by a separating hyperplane. In other words, given labelled training data (supervised learning), the algorithm outputs an optimal hyperplane which categorizes new examples.

Tuned parameters: C = 1, kernel = 'linear'
Matthews Correlation Coefficient: 0.7219

Random Forest

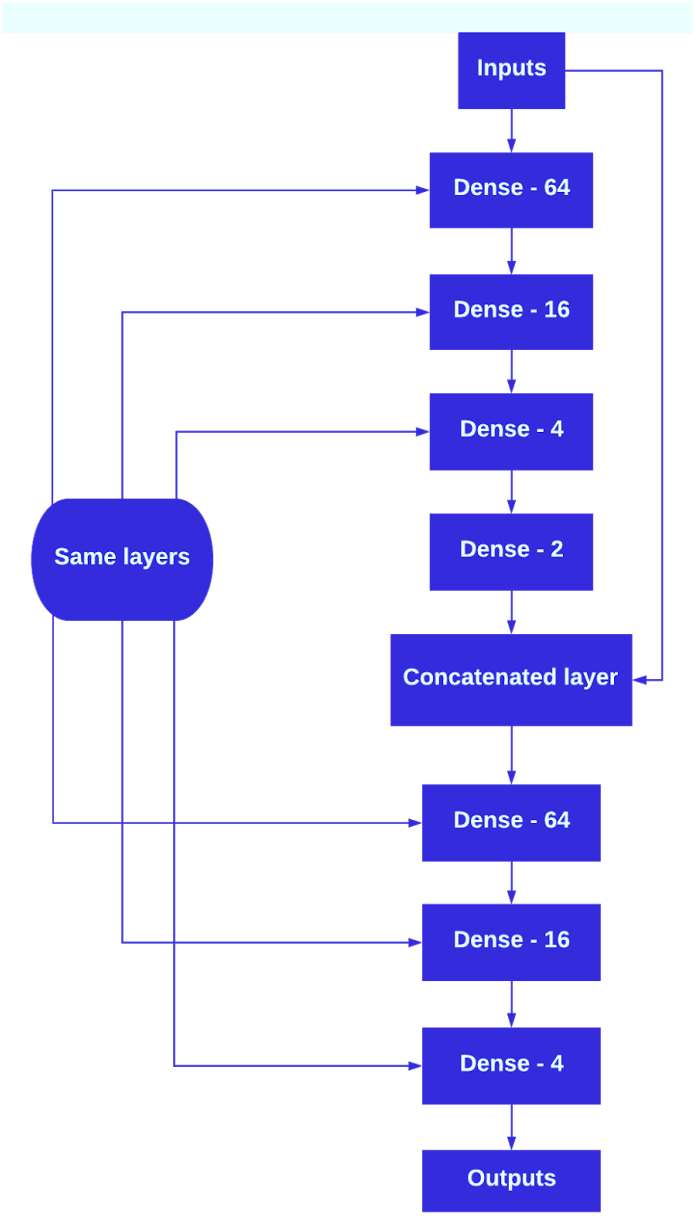
Random forests or random decision forests are an ensemble learning method for classification, regression and other tasks that operates by constructing a multitude of decision trees at training time and outputting the class that is the mode of the classes (classification) or mean prediction (regression) of the individual trees.

Tuned parameters: n_components=15, "max_depth": 10, "max_features":10, "n_estimator":600
Matthews Correlation Coefficient: 0.7221

Neural Network

Artificial neural networks are forecasting methods that are based on simple mathematical models of the brain. They allow complex nonlinear relationships between the response variable and its predictors.
We used keras library for applying neural network on the training dataset. Multiple combination of neural layers were used to get the best possible matthews correlation coefficient score.

Different models of neural networks used were:



We use shared layers to train our model. The output of dense layer having two units was concatenated with the input layer in the neural network. The reason for keeping the layer with two neurons is to capture congestion vs non-congestion in a middle step and get them as a feature for the same model.

Tuned parameters: activation - selu
Matthews Correlation Coefficient: 0.726

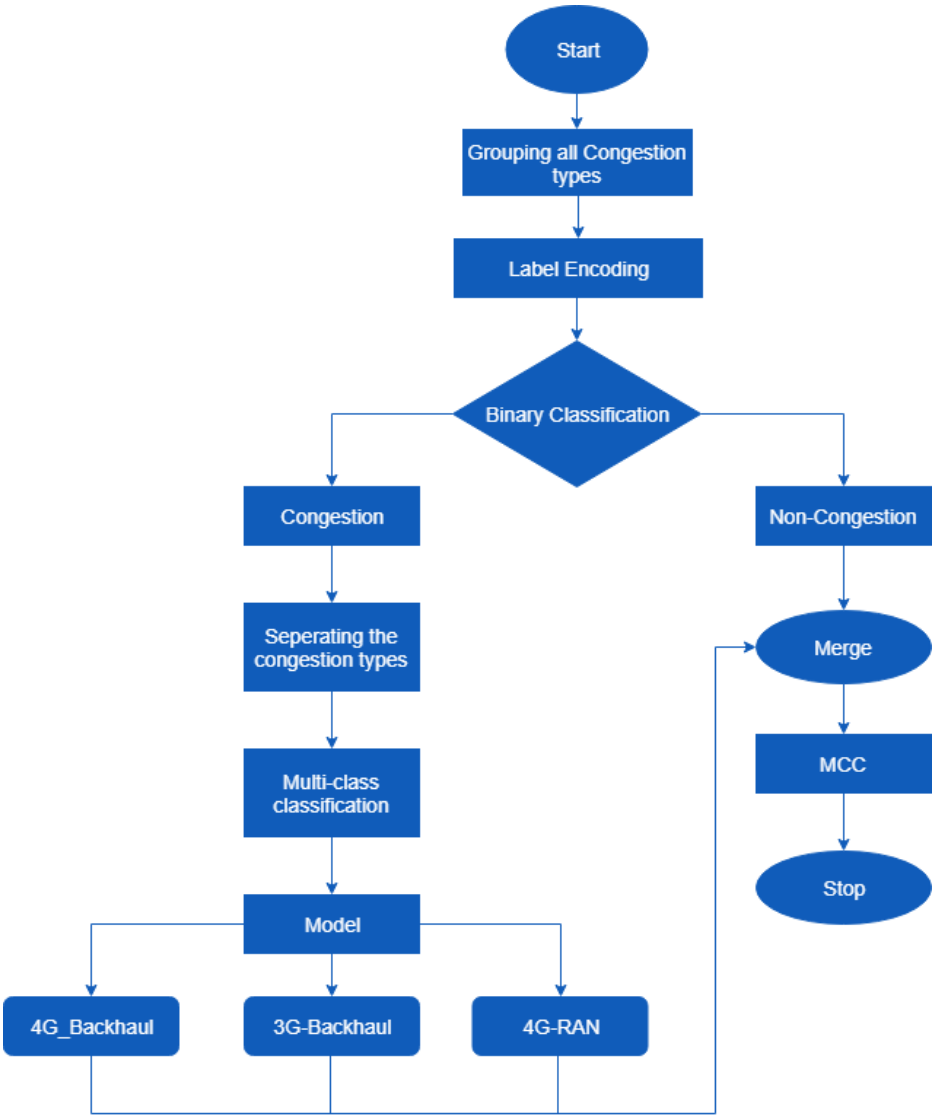
Naïve Bayes

Naive Bayes is a simple, yet effective and commonly-used, machine learning classifier. It is a probabilistic classifier that makes classifications using the Maximum A Posteriori decision rule in a Bayesian setting.

For applying naive Bayes model, the entire dataset is first normalized restrict all the values in all the feature columns to be in the range [0, 1]. After proper tuning and applying all possible

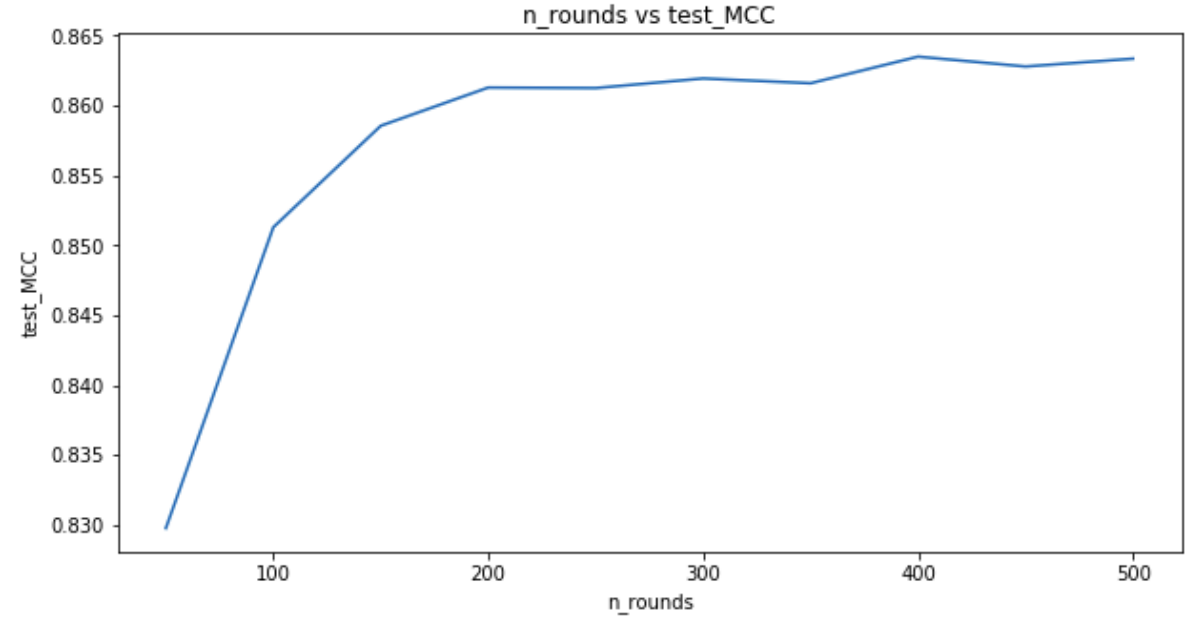
Tuned parameters: priors: None, var_smoothing:1e-9
Matthews Correlation Coefficient: 0.695

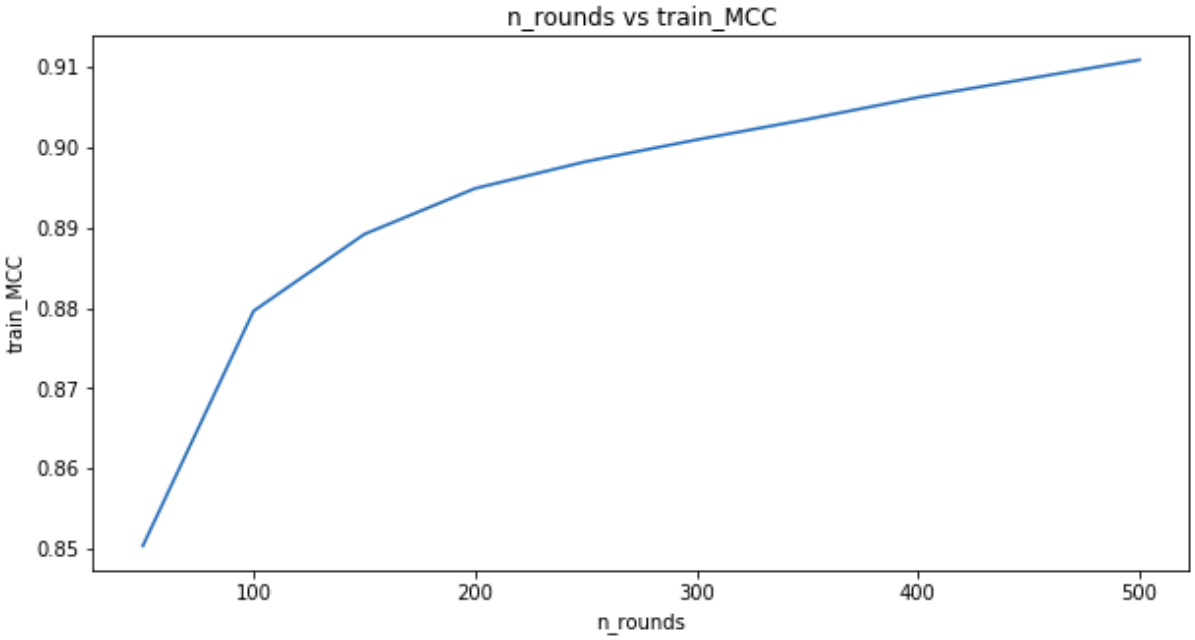
TWO STEP APPROACH



STEP 1: Binary Classification (Congestion Detection)

In this step, the congestion type for the entire training dataset is modified to Congestion and Non-Congestion by grouping all the congestion types. For this binary classification problem, we used xgboost model to predict the two classes and finally the parameters of the model are tuned.





Above plots shows the training and testing MCC variation with number of rounds respectively.

Tuned Parameters: booster='gbtree', colsample_bylevel=1, colsample_bytree=1, gamma=0, learning_rate=0.1, max_delta_step=0, max_depth=2, min_child_weight=4, missing=None, n_estimators=900, n_jobs=3, nthread=1, objective='multi:softmax', random_state=0, reg_alpha=0, reg_lambda=1, scale_pos_weight=1, seed=None, subsample=1, num_class = 2

STEP 2: Multiclass Classification (Congestion Type Detection)

The Congestion data types are then converted back to the three original classes 4G RAN Congestion, 4G Backhaul Congestion and 3G Backhaul Congestion and a multi class classification using a tuned xgboost model is used to predict the type of congestion in the network.

We used fivefold cross validation with proper tuning using xgboost cv and grid search cv for getting the best results without overfitting the data.

Tuned parameters: booster='gbtree', colsample_bylevel=1, colsample_bytree=0.2, gamma=0, learning_rate=0.1,max_delta_step=0, max_depth=2, min_child_weight=4, missing=None, n_estimators=850, n_jobs=3, nthread=1, objective='multi:softmax', random_state=0, reg_alpha=0, reg_lambda=1, scale_pos_weight=1, seed=None, subsample=0.6

Matthews Correlation Coefficient: 0.7395

FEATURE SELECTION

Principal Component Analysis

Principal Component Analysis is used to reduce the dimensionality of a data set consisting of many variables correlated with each other, either heavily or lightly, while retaining the variation present in the dataset, up to the maximum extent. The same is done by transforming the original variables to a new set of variables, which are known as the principal components and are orthogonal, ordered such that the retention of variation present in the original variables decreases as we move down in the order.

Recursive Feature Elimination

Recursive Feature Elimination is based on the idea to repeatedly construct a model and choose either the best or worst performing feature, setting the feature aside and then repeating the process with the rest of the features. This process is applied until all features in the dataset are exhausted. Features are then ranked according to when they were eliminated. As such, it is a greedy optimization for finding the best performing subset of features.

For the problem at hand, RFE algorithm is applied on the 37 selected features from the given data for the applied XGB Classifier model. Later, the feature selection is done according to rank predicted. The subscriber_count feature was scored as the best feature, implying the network congestion is mostly dependent on the traffic in the area.

HANDLING OVERFITTING

Overfitting is a modeling error which occurs when a function is too closely fit to a limited set of data points. Overfitting the model generally takes the form of making an overly complex model to explain idiosyncrasies in the data under study. In reality, the data often studied has some degree of error or random noise within it. Thus attempting to make the model conform too closely to slightly inaccurate data can infect the model with substantial errors and reduce its predictive power.

To prevent the dataset to overfit too well on the training dataset, the following measures were applied:

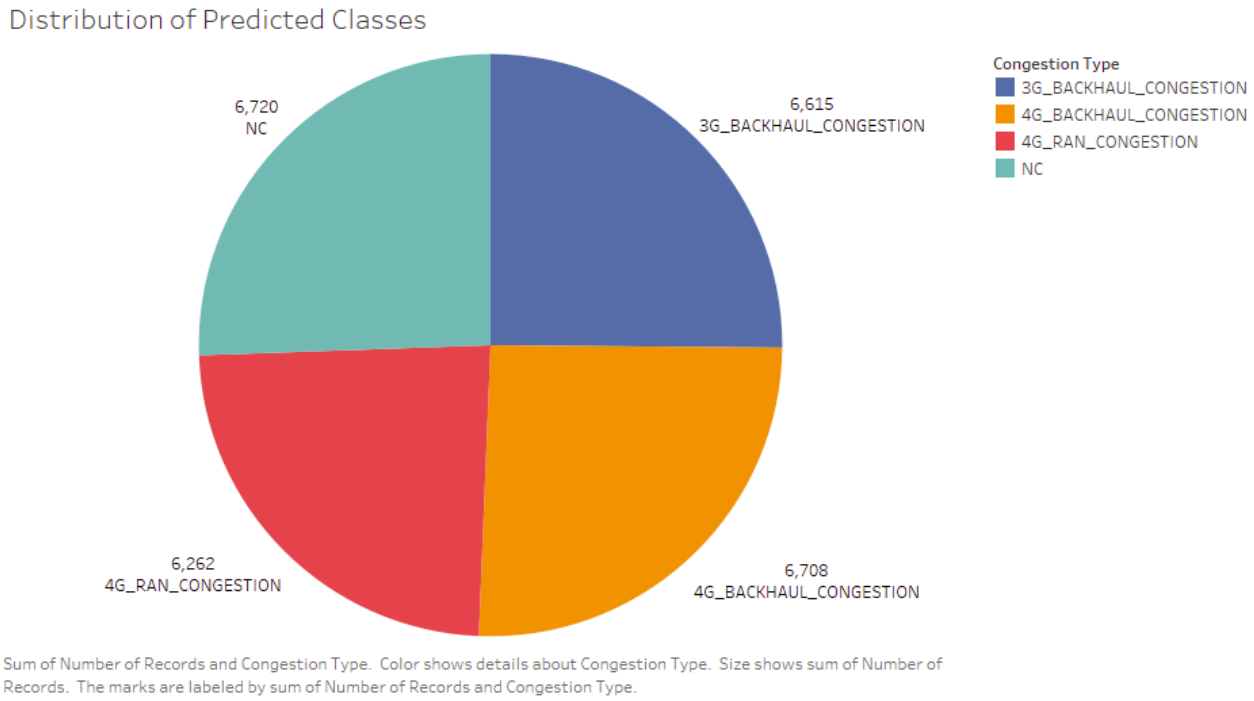
- **Feature selection**
  In order to get the best possible Matthews correlation coefficient score without overfitting the data, we applied three different metrics to calculate the feature importance and finally selected the best parameters from the list to get best balanced bias-variance tradeoff.

- **K Fold Cross validation with custom MCC**
  We applied 10-fold cross validation on the model in order to prevent the model handpicking a selected group of data entries to train on. K fold along with a custom MCC calculation was used for every model used to prevent overfitting of the model on training data.

RESULTS WITH COMPETITIVE SCORES

The two-step classification model, got us the best results with a cross validation Matthew correlation coefficient of 0.7395. The plot given below is the distribution of the congestion types on the test set given in the problem.

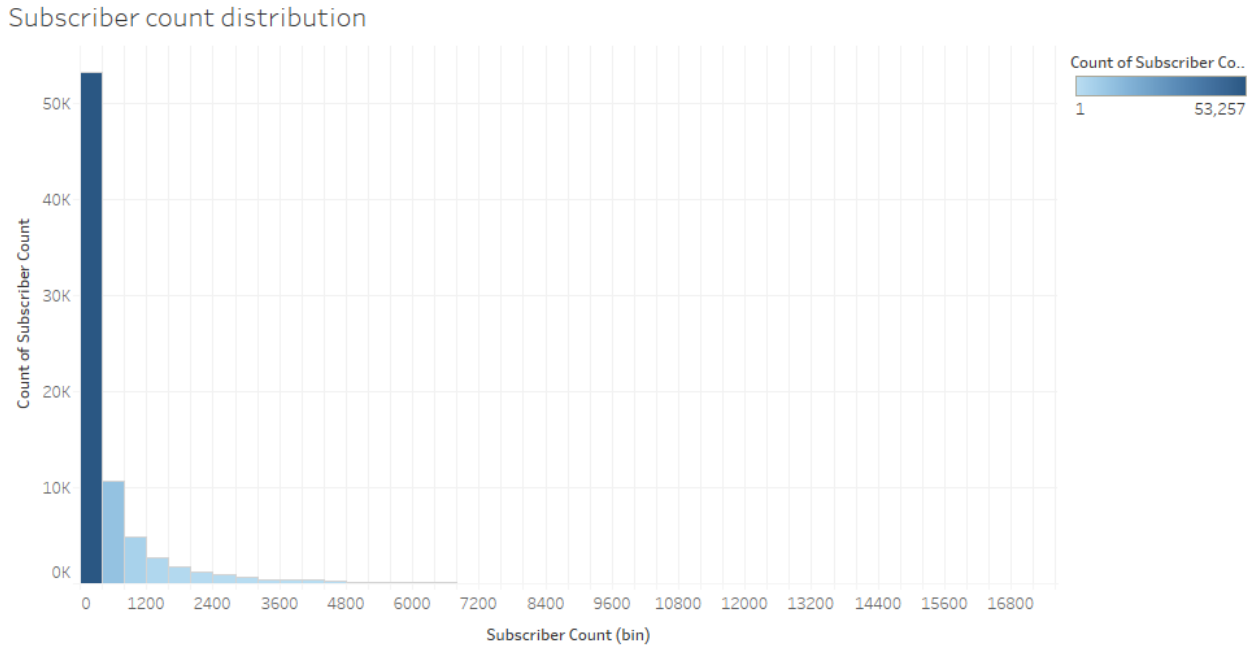
Model Used: ***Two step classification (Step 1: xgboost, Step 2: xgboost)***
Matthew Correlation Coefficient: ***0.7395***



ANNEXURE

Feature Name	**Feature Description**
cell_name	Cell tower number/name – Masked name for cell towers
4G_rat	Tower supports 3G/4G indicator
Par_year	Year under consideration (2018)
par_month	Month under consideration (December)
par_day	Day under consideration
par_hour	Hour under consideration
par_min	Minute bucket under consideration (Buckets of 5 min interval. Eg: value of 15 implies statistics are compiled/aggregated over a time period from 10-15 mins)
subscriber_count	Count of total subscribers for the cell in the specified time period
Usage data	Data usage by activity type; includes both upload and download bytes
web_browsing_total_bytes	Total web-browsing bytes
video_total_bytes	Total video bytes
social_ntwrking_bytes	Total social networking bytes
cloud_computing_total_bytes	Total Cloud computing bytes
web_security_total_bytes	Total web security bytes
gaming_total_bytes	Total gaming bytes
health_total_bytes	Total heath bytes
communication_total_bytes	Total communication bytes
file_sharing_total_bytes	Total file sharing bytes
remote_access_total_bytes	Total remote access bytes
photo_sharing_total_bytes	Total photo sharing bytes
software_dwnld_total_bytes	Total software download bytes

marketplace_total_bytes	Total marketplace bytes
storage_services_total_bytes	Total storage service bytes
audio_total_bytes	Total audio bytes
location_services_total_bytes	Total location services bytes
presence_total_bytes	Total presence bytes
advertisement_total_bytes	Total advertisement bytes
system_total_bytes	Total system bytes
voip_total_bytes	Total voip bytes
speedtest_total_bytes	Total speedtest bytes
email_total_bytes	Total email bytes
weather_total_bytes	Total weather bytes
media_total_bytes	Total media bytes
mms_total_bytes	Total mms bytes
others_total_bytes	Other bytes total
beam_direction	Beam direction of cell tower
cell_range	Cell tower range
tilt	Cell tower tilt
ran_vendor	Service Vendor
Congestion_Type	Type of congestion observed (Target Variable)
week_day	week_day = (par_day) % 7
hour_bucket	Feature to group the hour parameter into different hour buckets
avg_data_speed	avg_data_speed = ∑(##_total_bytes) / (1024 * 60 * par_min * subscriber_count)
large_data/small_data	Features to group the data types into large and small data usage
subscribers_per_area	subscribers_per_area = (subscriber_count) / (π * (cell_range)2)



The trend of count of Subscriber Count for Subscriber Count (bin). Color shows count of Subscriber Count.

REFERENCES

1. https://www.analyticsvidhya.com/blog/2018/06/comprehensive-guide-for-ensemble-models/
2. Machine Learning for Networking: Workflow, Advances and Opportunities [Mowei Wang et. al]
3. https://towardsdatascience.com/data-pre-processing-techniques-you-should-know-8954662716d6
4. https://www.kaggle.com/arthurtok/introduction-to-ensembling-stacking-in-python/notebook
5. https://scikit-learn.org/
6. https://en.wikipedia.org/wiki/Matthews_correlation_coefficient
7. https://towardsdatascience.com/pca-using-python-scikit-learn-e653f8989e60