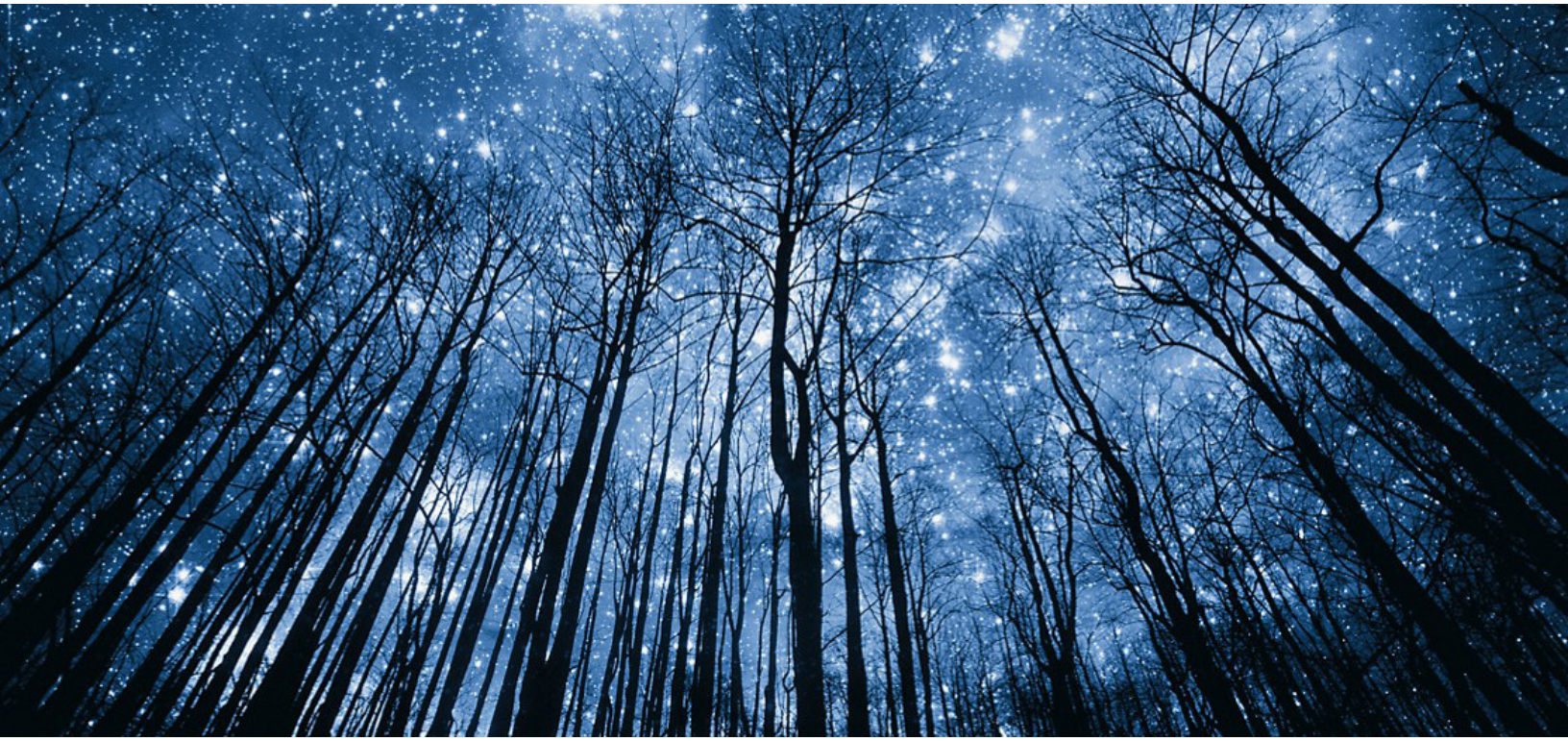
DELL Technologies

AUTOML FOR IMPROVED CUSTOMER CHURN PREDICTION



Shriya Avasthi
Associate Sales Engineering Analyst
Dell Technologies

Dr. Partha Sarathi Mangipudi
Professor
Amity Centre for Artificial Intelligence (ACAI)
Amity University, Noida

DELL Technologies
Proven Professional

Knowledge Sharing Article

The Dell Technologies Proven Professional Certification program validates a wide range of skills and competencies across Dell's multiple technologies and products with both skill and outcome-based certifications.

Proven Professional exams cover concepts and principles which enable professionals working in or looking to begin a career in IT. With training and certifications aligned to the rapidly changing IT landscape, learners can take full advantage of the essential skills and knowledge required to drive better business performance and foster more productive teams.

Proven Professional certifications include skills and solutions such as:

- Data Protection
- Converged and Hyperconverged Infrastructure
- Cloud and Elastic Cloud
- Networking
- Security
- Servers
- Storage
- …and so much more.

Courses are offered to meet different learning styles and schedules, including self-paced On Demand, remote-based Virtual Instructor-Led and in-person Classrooms.

Whether you are an experienced IT professional or just getting started, Dell Technologies Proven Professional certifications are designed to clearly signal proficiency to colleagues and employers.

Table of Contents

Introduction

The word ‘churn’ is defined as the rate of change in the number of customers a particular business acquires or loses over a period, due to various parameters. When referring to customer churn in commercial industries, it is defined as the percentage of subscribers switching from one specific company to another to avail their services, considering the differences in their product features, services, pricing, and many other factors. It is the main concern for most of the active companies in today’s industry and they try to reduce their customer churn rate to a minimum, considering the stringent competition from other companies offering similar products and services.

Customer retention is more valuable than customer acquisition for most companies. Technical companies have the highest churn rate when compared to any other sector, at 13.2%. There are many ways in which concerned industries tackle the problem of churning - creating predictive churning models and analyzing previous data of customers is one of them. It helps companies to realize the causes for the churning of customers and helps them retain their customers by improving existing technology, introducing new technology, and even by introducing exclusive beneficial offers. This helps in boosting their revenue, strengthening their brand name and in building a meaningful relationship with their customers, which enables the companies to retain their old customers while creating new leads. To do so, data analytics plays a crucial role to help companies understand the latest trends and requirements of the customers for a positive business impact. Research shows significant work is being done in customer churn prediction for commercial industries with the help and insights provided by artificial intelligence, using various statistical and data mining methods.

AdNan et al. [1] proposed a combination of Genetic Programming with Adaboost (Adaptive Boost), aimed at the evolution of multi programs for every class in which the weighted sum of the Genetic Programming outputs were the criteria for final predictions. Huang et al. [2] has considered the churn problem for the platform of big data to prove that big data can be used to greatly enhance the process of churn prediction. He has dealt with big data, using Random Forest algorithm, and used AUC as criteria for model performance evaluation. Brandusoiu et al. [3] used a dataset consisting of call details related to 3333 customers with 21 features for churn prediction using an advanced data mining methodology. This dataset is comprised of no missing values.

The author used Neural Networks, Support Vector Machine and Bayesian Network algorithms for churn prediction, after using Principal Component Analysis (PCA) for dimensionality reduction. Makhtar et al. [4] have proposed rough set theory, and it was shown that their algorithm outperformed several algorithms in their research paper, for implementing a churn prediction model. Mahreen et al. [5] used multi-classifier system and compared it to other existing classifiers, for the prediction of churn. Two datasets, the second being South Asian Churn dataset [6] were used for experimentation and an accuracy of 97.2% and 86.3% was achieved for the two datasets respectively and the multi-classifier system outperformed the previously analyzed and implemented models.

Automated Machine Learning (AutoML), as the name suggests, aims at completely automating the workflow from pre-processing of a raw dataset to implementation of models for prediction, i.e., the entire end-to-end process or some parts of the workflow. It may be implemented using various techniques and some of the most popular AutoML tools are TPOT (Treebased Pipeline Optimization Tool), auto-keras, auto-sklearn, RoBO etc. Randal et al. [7] used TPOT vo.3 and analyzed its performance on 150 supervised datasets for classification. They have shown that the process is automated in several aspects and discovered that the model pipelines selected by TPOT outperformed basic classifiers on several benchmarks. Marin et al. [8] has created customer churn prediction models using the RapidMiner AutoModel which is a semi-automated AutoML tool that requires human input at several stages. The tool suggested filtration of 88 out of 205

attributes of the dataset. Deep learning, logistic regression and gradient boosted tree algorithms were used to attain an accuracy of 87.31%, 88.86% and 88.48% respectively. The paper implemented a two-layer architecture model of AutoML and worked on datasets of Orange telecom and cell2cell and achieved an accuracy of 63% and 89% respectively, for the customer churn prediction of the two datasets. Baligh [9] proposed a model for churn prediction for the customer Churn Dataset of Orange telecom, by using PySpark. Six classifiers, out of which, Decision tree, Random Forest and Gradient-boosted tree models outperformed others in his training model. Decision Tress classifier gave the maximum accuracy of 89.40% on test dataset.

Considering the emergence of new machine learning automation tools that would greatly benefit the users by assisting them save their time and effort, in this paper we have compared the accuracy results derived by using traditional machine learning algorithms to those obtained through TPOT, to better understand how efficient the particular AutoML tool is.

The task was performed on two publicly available customer churn datasets [6] [10]. To gain more insights from the data, to analyze the reason for customer churn and to understand relationship between the different features, for dropping correlated features, handling missing values and dropping Nan values, exploratory data has been performed by visualizing the relationship between unique features. Accuracy results obtained by using traditional machine learning algorithms and TPOT suggested pipeline are then compared for both the cleaned datasets.

The proceedings of this paper are organized as follows: First, we will introduce the materials and methods used in churn prediction systems. Then we describe our experimental set-up and results, which include the techniques of feature extraction, selection, models, and churn prediction. Finally, a comparative result of performance of TPOT against existing algorithms in discussions has been drawn as a part of the conclusion.

Materials and Methods

Exploratory Data Analysis (EDA)

EDA is the first step where we begin with the analysis of any given data - developed by “John Turkey” in 1970s. As the name suggests, it helps us explore our data to summarize the main characteristics and derive information about the relationship between various features of the dataset so that we may infer meaningful results and observations through visualization.

The various techniques and tools that prove to be helpful in exploratory data analysis are various graphing techniques (histograms, multi-vary chart, scatter plot, box plot etc.), dimensionality reduction (multidimensional scaling, principal component analysis (PCA), multilinear and non-linear dimensionality reduction) and the typical quantitative techniques (median polish, teimean and ordination). The procedural components of EDA are as follows:

- **Data description** - Helps in generating different descriptive statistics for numeric and object data that display a summary of central tendency, dataset’s distribution shape and dispersion excluding the null values in the dataset.

- **Data cleansing** - This procedure helps us in detection of incomplete, inaccurate, irrelevant, or incorrect data in our dataset and then in its removal, correction, or modification accordingly so that we may improve the performance metrics.

If we have some missing data in our dataset, we can handle it using three different techniques:

- **Drop ‘Nan’/ ‘Null’ or missing values** - This technique can be used to simply drop/remove the rows with missing values and is the easiest. It should be implemented only if those values are not huge in

number and their removal will not affect the performance metrics negatively.

- **Using Machine Learning (ML) algorithm to predict missing values** - This technique is very efficient but takes much more time and effort to be implemented, hence we recommend using it only for small datasets. The rows with missing values are considered as a test dataset and the rest of the rows are used as training dataset. The columns with no missing data are fed as input features (excluding the real output column of the data set) into the training model and the column with missing values is considered as the output of the model and hence the 'Nan' values are predicted using the model.
- **Statistical methods** - It is the most common and best technique to handle missing data in big datasets. We replace the 'Nan' values in every feature/column of our dataset by either the mean value (average of data values of the particular feature), median value (the central value achieved after sorting the data values of our feature in ascending order), or the mode value (data value with highest frequency in the feature).
- **Handling outliers** - Outliers are values in data that are completely different and unique from the crowd and are generally an indication of data variance or mistake occurred in data collection. They can be easily detected by plotting a box plot, scatter plot, or by calculating either Z-score or the interquartile range.
- **Visualization of relationships through plots** - EDA is mandatory before we start modeling our data because it lets us visually analyze the key features to be used for modeling and the features that may be dropped. The heat maps, histograms, scatter plots, graphs etc. that we derive from EDA can be used in providing deep insights into our data and hence are especially useful throughout the process.

Handling Missing Values

The missing values in a dataset need to be handled before using it for training or testing of a dataset. One of the several techniques mentioned below can be used, depending on the type of features.

- Encoding - As the name suggests, encoding of data is done to convert it into the desired format to serve different purposes of compiling a program, executing a program, storing data, transmitting data, or compressing/decompressing it. It is an easy and efficient way to archive original data and arrange it in an automated manner. While creating predictive churn models, most of the time there arises a need to encode categorical values (values which have two or more categories without an intrinsic order) into numeric data. There are several encoding techniques that may be used according to which one is the most apt.
- Replacing values - In this technique, we have the freedom to select the numerical value we want to assign to the categorical values of the feature we are encoding just by creating a directory containing mapped numbers for each category, according to our preference. This type of hard coded library is efficient when the number of categories is less. For a larger number, dictionary comprehensions must be used where category names are stored in a list and then zipped to a sequence of numbers, to be iterated over it.
- Label Encoding - This is the technique used in the program that has been used twice in this report. It is legit and intuitive and may help us improve our model's performance, but at the same time the numerical values may be misinterpreted by the algorithm we are using. If you want to encode a column/feature with n categories, label encoding will assign 0-(n-1) numerical labels to the categorical values and convert them to a number.
- One-hot encoding - It follows a strategy that converts each categorical value into (True/False), i.e., assigns 1 or 0 value to the column. It solves the problem of uneven and unjustified assigning of weights to categories of a feature but can only be used when there are lesser categories in the feature.
- Binary encoding - This technique proceeds by first converting the categorical data into ordinal, converting those integers into a binary code, and then splitting digits from the binary string into

separate columns.

- Backward difference encoding - This technique is generally more suited for nominal or ordinal variables. In it, the mean of a dependent variable in a certain level is compared to the mean of a dependent variable just before its level and hence a feature with n categories enters a regression as a sequence of n-1 dummy variables.
- Miscellaneous Features - When we come across categorical features specifying a range of values, we usually either replace them with some measure, such as the means of the values of the range or can split the range values into two different columns.

Models Used in our Research

Machine learning is an application of AI and gives the ability to systems to learn from their experience without being explicitly programmed. Below are some machines' learning algorithms documented in our paper:

- **AdaBoost** - A Meta and boosting algorithm that helps to combine all the decision stumps, i.e., all the weak classifiers (decision trees with single split), into a single strong classifier. It puts less emphasis on the classifiers that are handled well while putting more weight on the difficult ones. It can be used for regression as well as classification problems.
- **Gradient boosting** - A prediction model in the form of an ensemble of weak classifiers is produced from this algorithm as; in an iterative fashion it combines all weak classifiers into one strong classifier. It may be used for regression and classification problems.
- **Support vector machines** - A supervised learning method that can be used to analyze data both for regression and classification problems. It sorts the data into one or two categories giving the output of the same while separating the categories as far apart as possible, using margins.
- **Stochastic gradient descent** - Helps in decreasing the errors in a machine learning network by making small adjustments iteratively in the network configuration. It helps in finding the best parameter configuration for a machine learning algorithm.
- **Logistic regression** - A regression analysis. This algorithm is used to explain the relationship between a dependent dichotomous variable and other independent variables that may be ordinal, ratio-level, or interval. It is most appropriate when the dependent variable is binary.
- **Random Forest** -Comprises of many ensembles operating, individual decision trees which split out a class prediction, making the class with the maximum number of predictions the model's prediction.
- **Decision Tree** - Its structure is like a flow chart with internal node representing a test on any attribute, with its branches representing the test outcomes and leaf nodes representing the decision taken after processing all the attributes, i.e., the class label.
- **Gaussian naive bayes** - A probabilistic classifier based on bayes theorem which explains the probability of occupancy of an event using prior knowledge of conditions relating it to event conditions. This algorithm is known to yield great accuracy with much less effort.
- **K-nearest neighbors** - Finds distances between all the examples and queries in the data and selects the specified number of examples that are closest to the query. After which it votes for the most averages or the most frequent label.

Tree-based Pipeline Optimization Tool (TPOT)

TPOT is one of the first Python based AutoML tools based on scikit-learn. It is the abbreviation for Tree-based Pipeline Optimization Tool and, being open source, is under constant development. It was developed at the Computational Genetics' laboratory by Dr. Randal Olson, postdoctoral, along with Dr. Jason H. Moore. It is a library that explores the best pipeline out of thousands of possible pipelines for a dataset [7]. It acts as a data science assistant and uses genetic programming for optimizing machine learning pipelines by selecting the best models based on natural selection or survival of the fittest, which saves you ample time for data gathering and preparation.

Considering it is based on genetic programming, which is a subset of machine learning and a type of Evolutionary Algorithm (EA), it has these basic properties which can be considered steps of how the genetic programming or TPOT works. These are as follows:

- **Selection** - Finds the best and fittest model among the rest using a fitness function for evolution at each iteration.
- **Crossover** - Breeds the best and the finest models to get a new generation and population.
- **Mutation** - The best models from crossover are then mutated and the best and fittest model is selected from the mutated offspring of the new generation.

Data collection and data cleaning are not included in the pipeline flow of TPOT. It assumes that any data that it is working on is numeric in nature and hence any categorical value must be encoded to numeric. Missing values and data cleaning have to be handled outside TPOT. It expects a clean dataset. It internally applies PCA, min/max scalar if required, for feature pre-processing for the dataset. It automates the process of model selection and hyper parameter optimization using genetic programming.

The pipeline can be analyzed from the below figure:

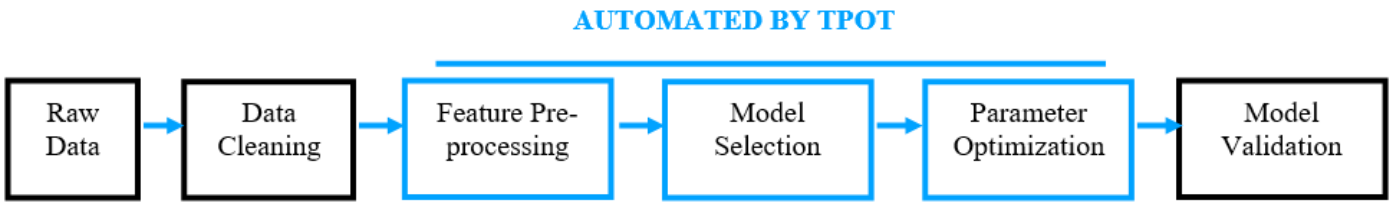


Figure 1 Machine learning TPOT pipeline

TPOT can be installed by using the “pip install tpot” command in Python and has its dependencies on scikit-learn and NumPy, hence these libraries need to pre-installed before using it. All the AutoML frameworks are compute-intensive, so while implementing, either the processes need to be distributed or multiple cores from the Central Processing Unit (CPU) must be used.

Considering this, in our paper, the multiprocessing function was imported to create a multi-processing package which assisted us in running TPOT in multiple CPU cores of our hardware system. In our case, this was MacBook Pro embedded with Appel’s M1 chip. It has a 32-core processor with its CPU consisting of a total of four,high performance and efficiency cores, making the computation run smoothly. To proceed with the implementation, the forkserver method was used to create and run a total of 20 jobs simultaneously.

TPOT uses a genetic algorithm with the generation and population size being two particularly important parameters. In our paper, parameters with similar adjustments, namely generations, population size, scoring, verbosity, n jobs and random sate have been used for both the datasets, while considering their approximate similarity in size, structure, and less complex nature.

Generations is set as 100 by default and sets the total number of iterations that TPOT will perform. Population size determines the number of models and the sample hyper-parameters that a user wants to evaluate leading to the creation of models from the first generation. The top and best performing models are selected from the first generation and crossover and the mutation of those selected models is performed to generate new model types for the second generation, out of which top performing models are selected and

the process is repeated until the set number of generations are achieved.

The scoring function is set as accuracy by default and is used for the performance analysis of the pipeline suggested automatically, while verbosity provides information for each epoch for the training process based on the level chosen. N jobs is set to 1 by default and indicates the total amount of tasks and processes running in parallel during the process of automating the TPOT pipeline and random sate, set as 'none' by default, which ensures uniformity of result given by TPOT on the same dataset.

Experimental Set-Up:

Data analytics and predictive analytics have been performed on two datasets. Accuracy is the criterion for performance evaluation in this paper - the performance of all these models is compared with the performance of the model suggested by TPOT. The datasets were first downloaded from the South Asian Churn Dataset: https://www.kaggle.com/mahreen/sato2015 and Telecom Churn Dataset: https://bml-data.s3.amazonaws.com/churn-bigml-80.csv & https://bml-data.s3.amazonaws.com/churn-bigml-20.csv in CSV format and then uploaded on separate Google Colab notebooks, to be worked upon. Required libraries were downloaded, while keeping in mind the dependency of TPOT, scikit-learn and NumPy, and required libraries for EDA and predictive analytics. The methodology of analytics flow for both datasets is as follows:

Datasets

In our paper, two different datasets have been used to build and analyze predictive models. The first dataset is the "South Asian churn dataset" [6] It is a public dataset that was collected from a major wireless telecom operator in South Asia and uploaded on Kaggle. This dataset gives us information of 2000 customers, collected for the months of August and September of the year 2015. The dataset specifies the values of 13 features of the operator, as it is related to the customer and gives us the output information of whether the particular customer churned or stayed active. The dataset contains 2000 rows, each row containing data about each customer and the feature values mentioned in the 13 columns.

The second dataset is the "Telecom Churn Dataset" [10]. It is a publicly available dataset for Orange Telecom which is updated annually. The dataset consists of 3333 customer records, and it comprises of the cleaned customer activities included in a total of 20 features

South Asian Churn Dataset

Feature Extraction

The descriptive detail about the features is as follows:

- Network Age (network_age): The lifespan in days, of the customer and operator.
- Aggregate of Total Revenue (Aggregate_Total_Rev): Aggregate monthly revenue earned from the customers, in rupees for the months of august and september,2015.
- Aggregate of SMS Revenue (Aggregate_SMS_R): Aggregate revenue earned through the SMS service used by the customer in the mentioned time interval.
- Aggregate of Data Revenue (Aggregate_Data_R): Aggregate revenue earned through the data service used by the customer in the mentioned time interval.
- Aggregate of Data Volume (Aggregate_Data_V): Total volume of data used by the customer through data service in the mentioned time interval.
- Aggregate calls (Aggregate_Calls): Total number of calls made by the customer in the mentioned time interval.
- Aggregate of On Net Revenue (Aggregate_ONNET_REV): The revenue earned by intra network calls, etc. made by the customer in the mentioned time interval.

- Aggregate of Off Net Revenue (Aggregate_OFFN): The revenue earned by inter network calls, etc. made by the customer in the mentioned time interval.
- Aggregate of Complaint Count (Aggregate_compl): Total complaints made by customers in the mentioned time interval.
- FavUser Type (aug_user_type, sep_user_type): Tells us if the user is subscribed to a 2G, 3G or some other service.
- Favorite Other Network (aug_fav_a, sep_fav_a): Tells us, which other operator does the subscribers made the most of the calls to in the mentioned time interval.

Exploratory data analysis:

- Data description - Numeric data from the dataset was then described to study the count, mean, standard deviation, maximum and minus of the values of the features as well as 25 (lower), 50 (median) and 75 (upper) percentiles of the features.

- Data Cleansing - The data set was then checked for missing or Nan values and the count for Nan values turned out to be 245 for aug_user_type, 206 for sep_user_type, 1 each for aug_fav_a and sep_fav_a and to visualize it, a heat map was plotted.

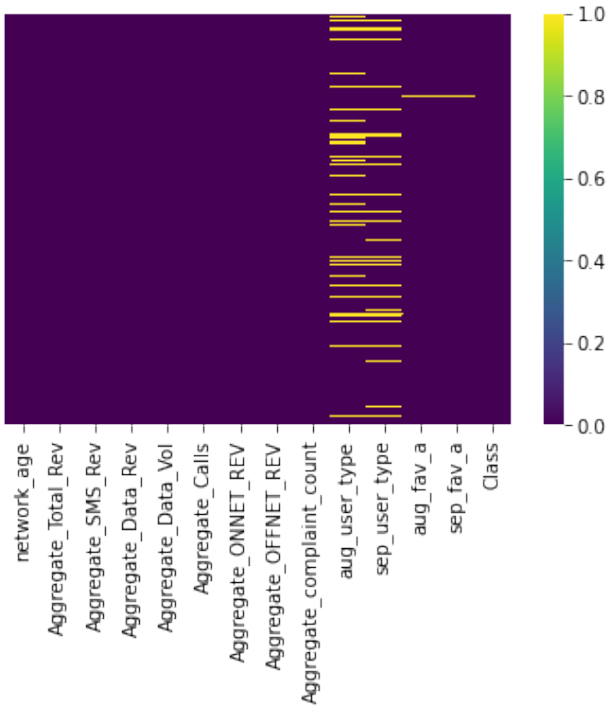


Figure 2 Heat Map-Nan Values

The single missing value in aug and sep_fav_a lies in the same row and removal of one row wouldn't have affected our data analytics and prediction model because this is a huge dataset. Hence, that row was removed (drop Nan or missing values)

Visualization of Relationships through Plots
For data analytics and data cleansing, relationships between different features with themselves and the

output were visualized to infer meaningful results.

Observations and Data

Observation 1: Fig 3. tells us that out of 1999 subscribers, 1000 churned and switched to a different operator and the churn rate is almost 50% for two months while the normal churn rate for telecom operators is 1.9-2.1% monthly and 10-67% annually. So, the statistics of churn rate don't seem so well for the company.

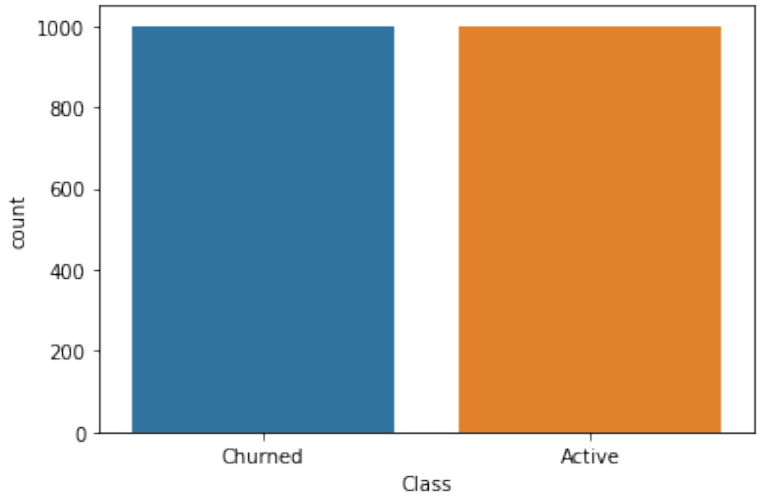


Figure 3 Count plot visualization of churned and active users

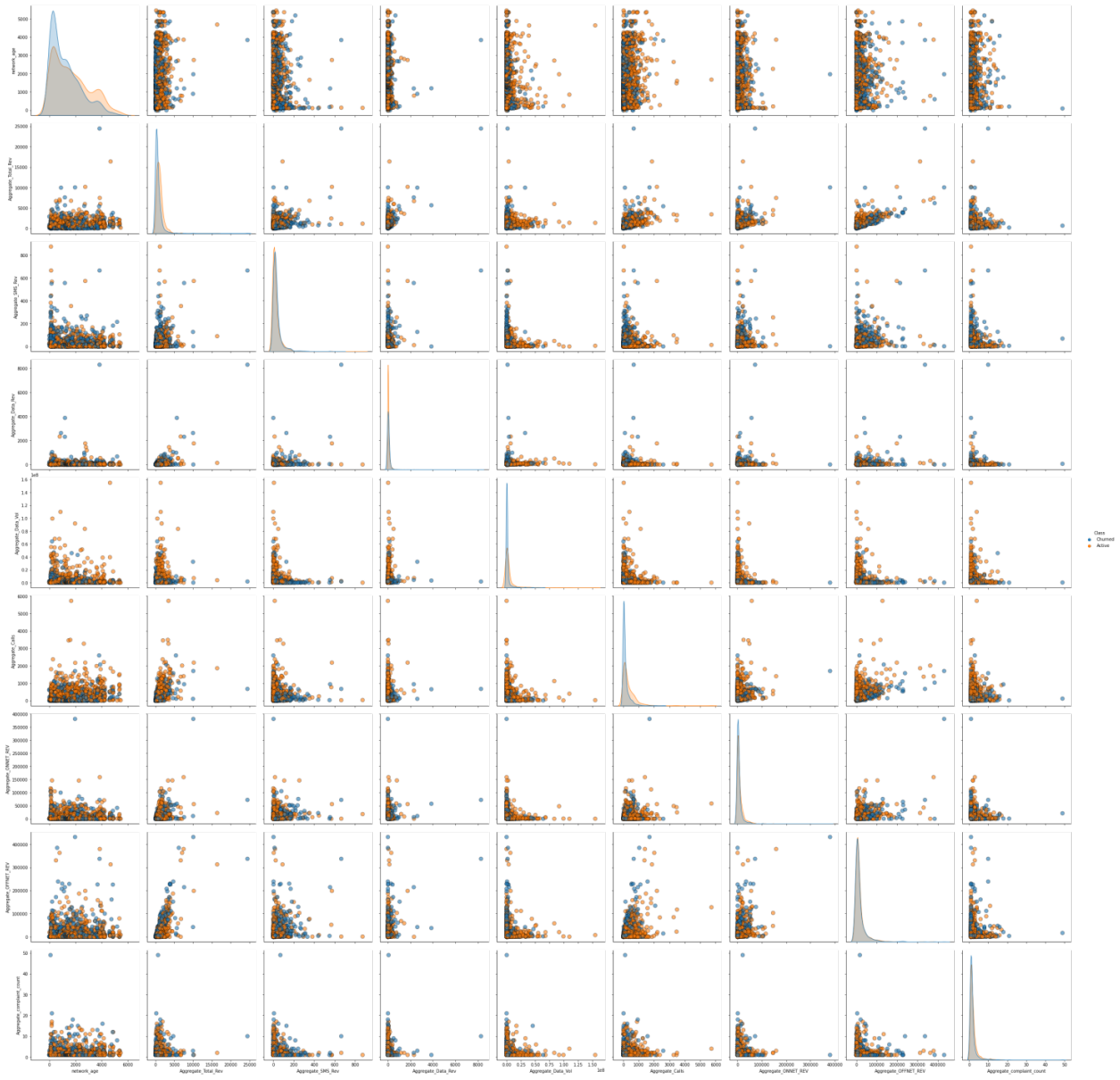


Figure 4 Visualization of relation between numerical type features

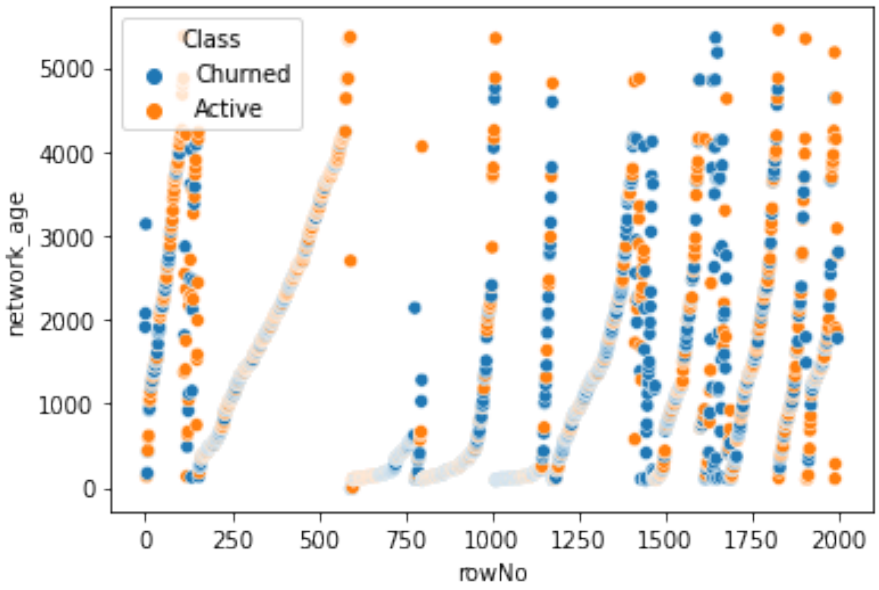
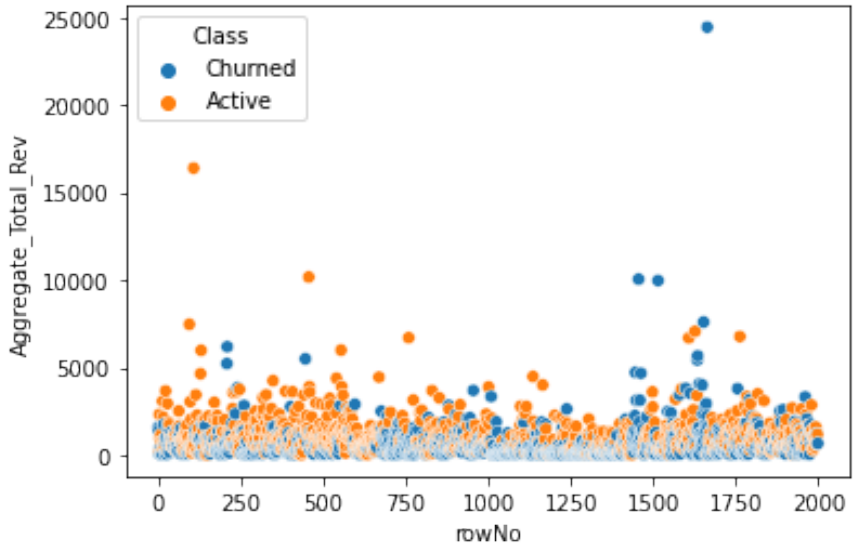


Figure 5 Scatter Plot visualization of relation between class and network age

Observation 2: From Fig 4. A clear inference can be made that the comparatively new subscribers who made more inter-operator calls, i.e., generated more offset revenue churned during the given interval. The reason could most likely have been convenience. While no clear inference can be derived from Fig 5. and seems like network age alone was not a huge deciding factor for the company's churn rate.

Observation 3: From Fig 4., Fig 6., Fig 7., it is observed that there is a very small impact of aggregate total revenue, aggregate Data revenue, and aggregate SMS revenue on the Class.

Observation 4: From Fig 4. and Fig 7. It is observed that the customers are quite satisfied with the data services of the operator. The data services of the operator and the people using the maximum volume of data are still actively subscribed to the operator and have the minimum complaint counts.



6Figure 7 Scatter plot visualization of relation between class and aggregate total revenue

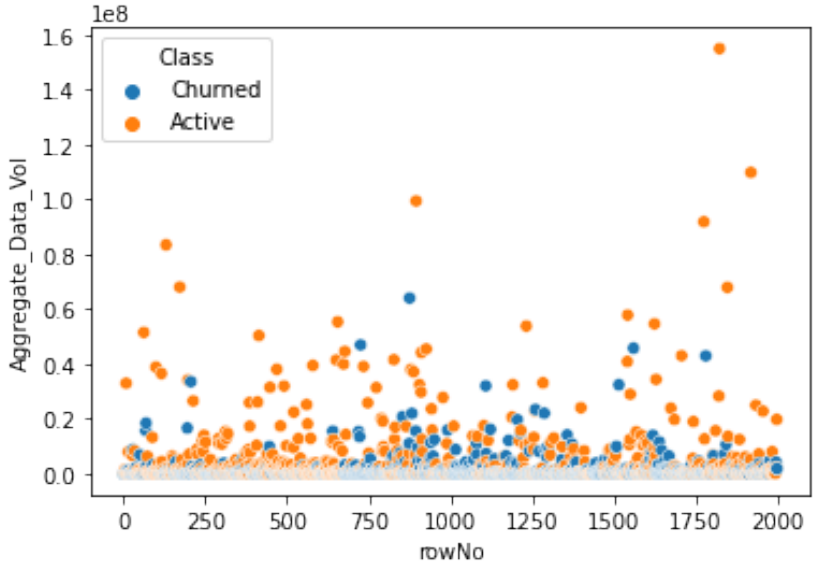


Figure 8 Scatter plot visualization of relation between class and aggregate data revenue

Observation 5: From Fig 4. and Fig 8. It is inferred that the offset revenue of the company might be a bit high for some specific operators. The subscribers who made comparatively fewer calls, but were charged high offset charges, are observed to have churned for more.

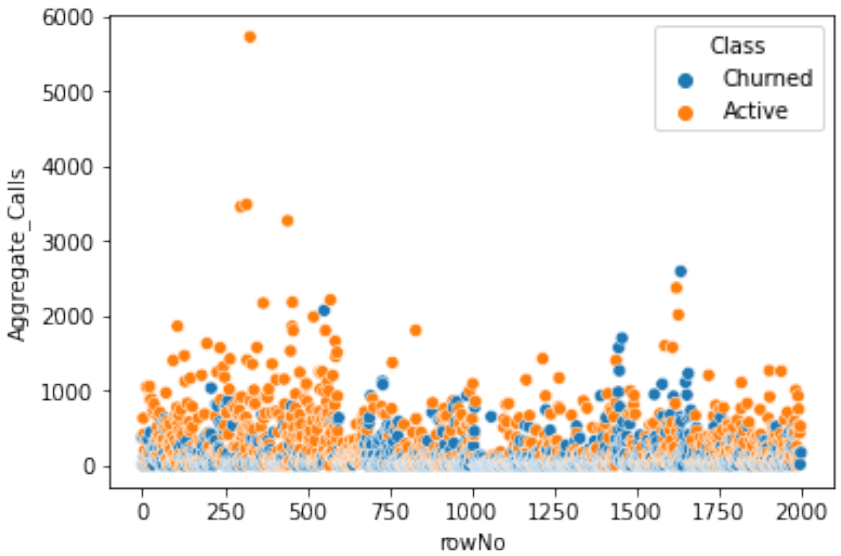


Figure 9 Scatter plot visualization of relation between class and aggregate calls

Observation 6: From Fig 4. and Fig 9. it is seen that most of the people who churned had a greater number of aggregate complaints when compared to the active users. There were a higher number of complaints that came from comparatively new users.

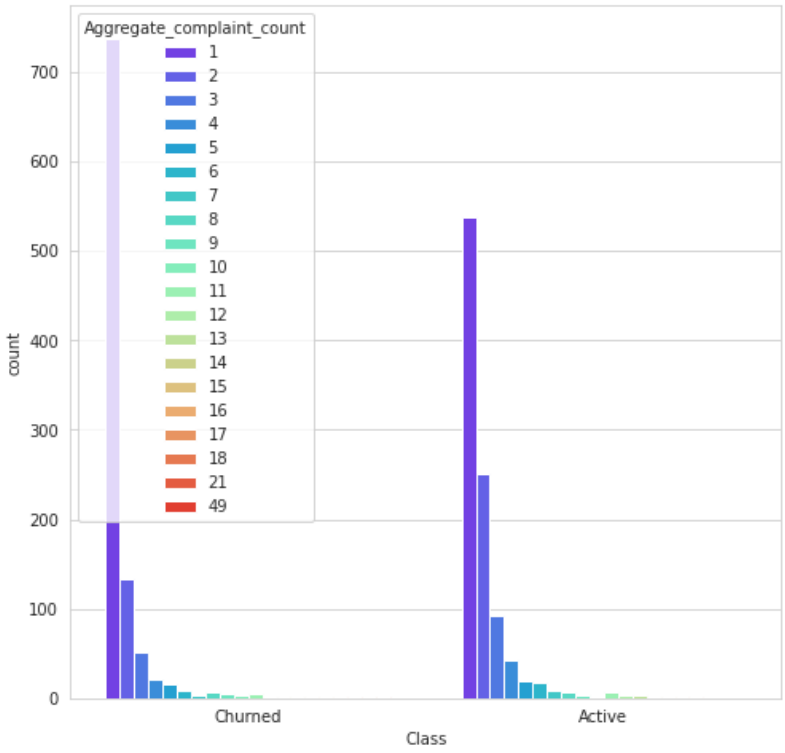


Figure 9 Count plot visualization of relation between class and aggregate complaint count

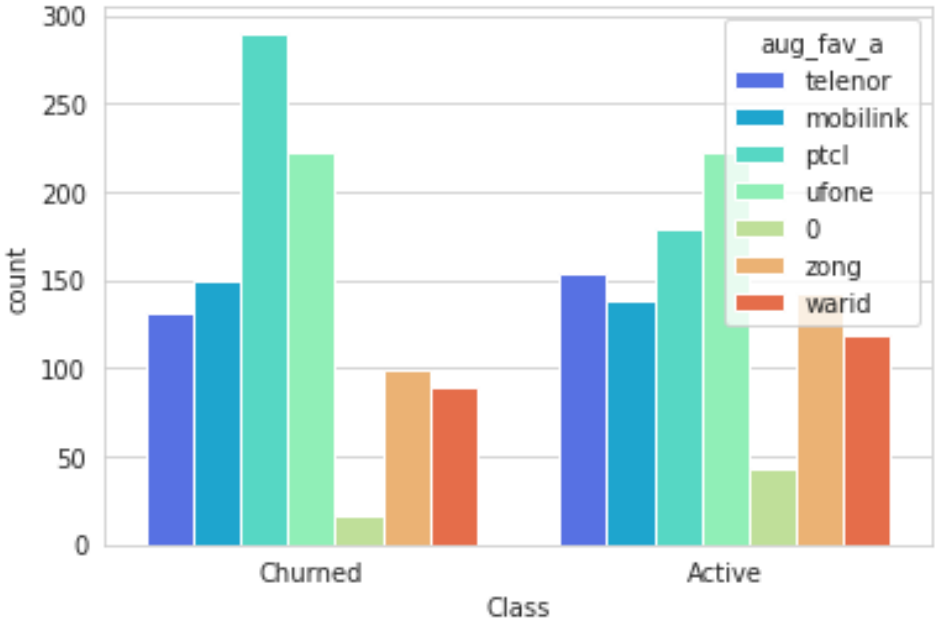


Figure 10 Count plot visualization of relation between class and favorite August operator

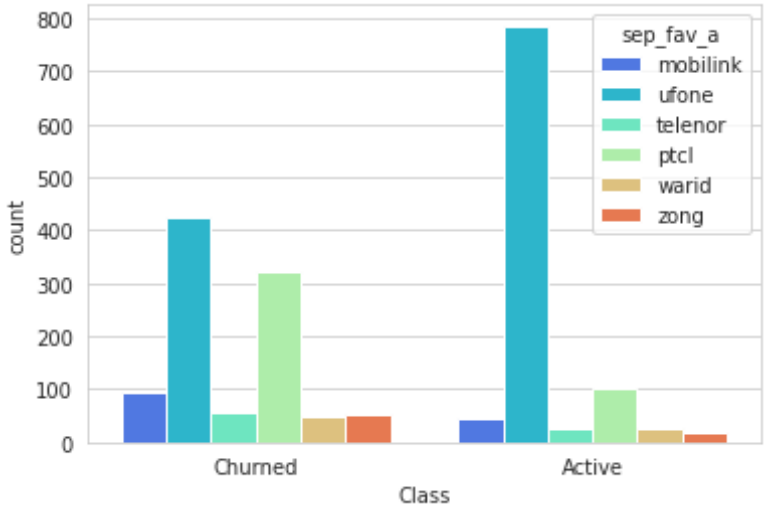


Figure 11 Count plot visualization of between class and favorite September operator

Observation 7: The inference made from plots in Fig 4, Fig 10. and Fig 11. is that major changes happened for the operator in the month of August in which ptcl and ufone were the top two other operators that were the favorites of their current subscribers. Maximum churn was also observed in these two categories potentially because of the high offnet charges of the company or it could be perhaps another reason.

By September, ufone had become the favorite of its subscribers and the churning rate, which was significantly high considering this feature, had significantly dropped. The case was not the same for ptcl, as the churning rate of subscribers had this as their favorite other operator and had significantly increased compared to the previous month. Inferences can be drawn that the operator introduced special plans in collaboration with ufone for their subscribers.

Observation 8: From Fig 4. and Fig 12. Another cause of customer churn for this operator can be inferred. Most of the subscribers who churned were generating more offset revenue than onset revenue and hence they might have churned to save money.

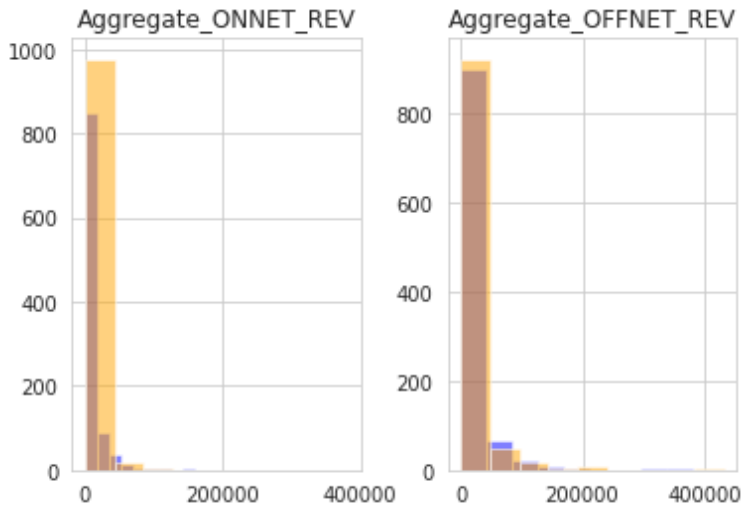


Figure 12 Histogram visualization of relation between aggregate offset and onset revenue

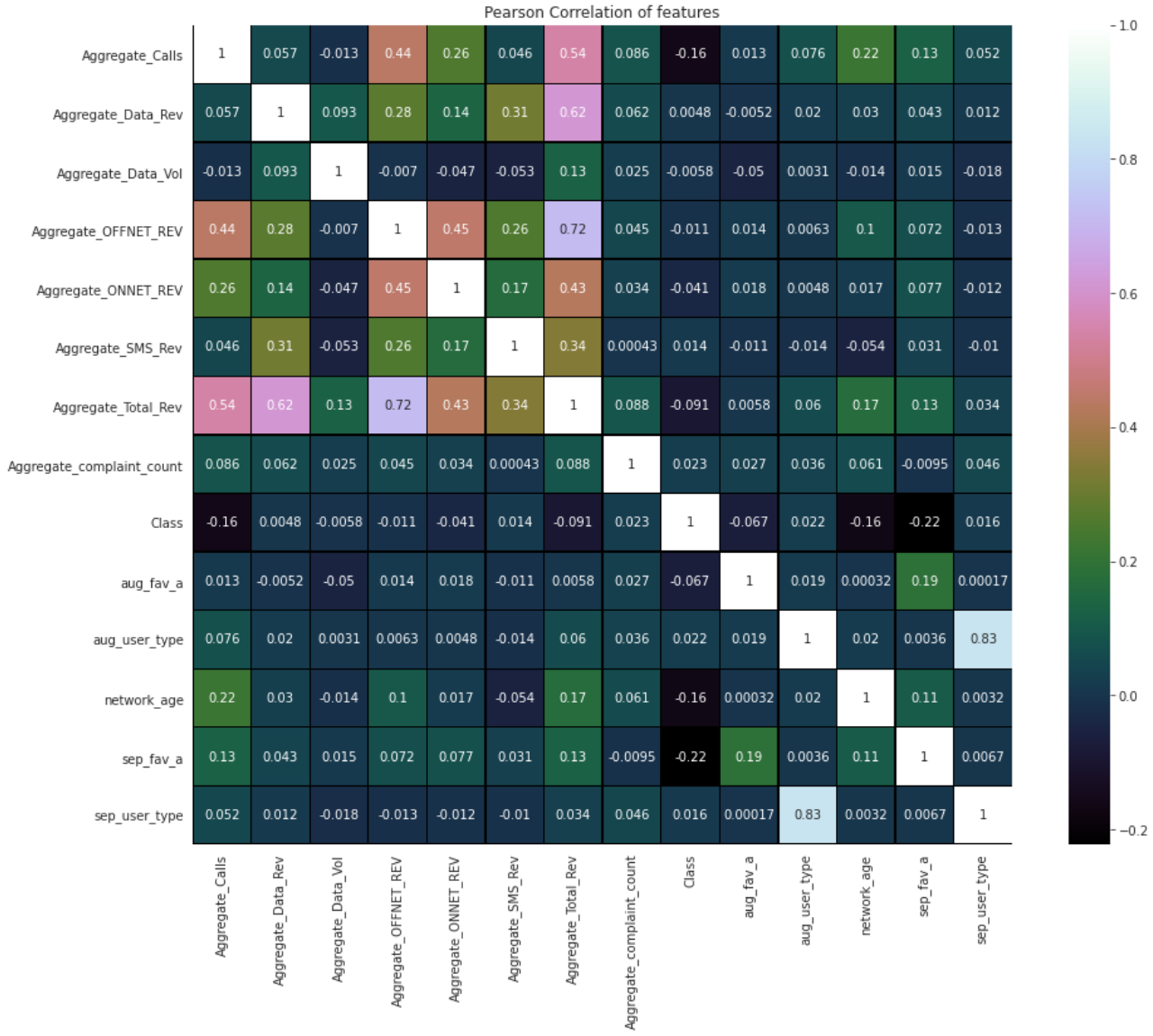


Figure 13 Pearson correlation of features

Observation 9: Observing the heat map implemented to derive Pearson correlation of features, which is a number used to indicate the extent of linear relation of two variables, ranging from -1 to 1 (Fig 13.), aggregate calls and aggregate complaint count are the two features that were most related to the features, August and September user type. The pattern of numerical data provided in the aggregate_complaint count feature and aggregate calls was best suited to be used to handle the missing values of the two features of dataset through statistical methods and hence their relation was checked.

Observation 10: August user type and September user type have a high correlation and one of the features can be dropped.

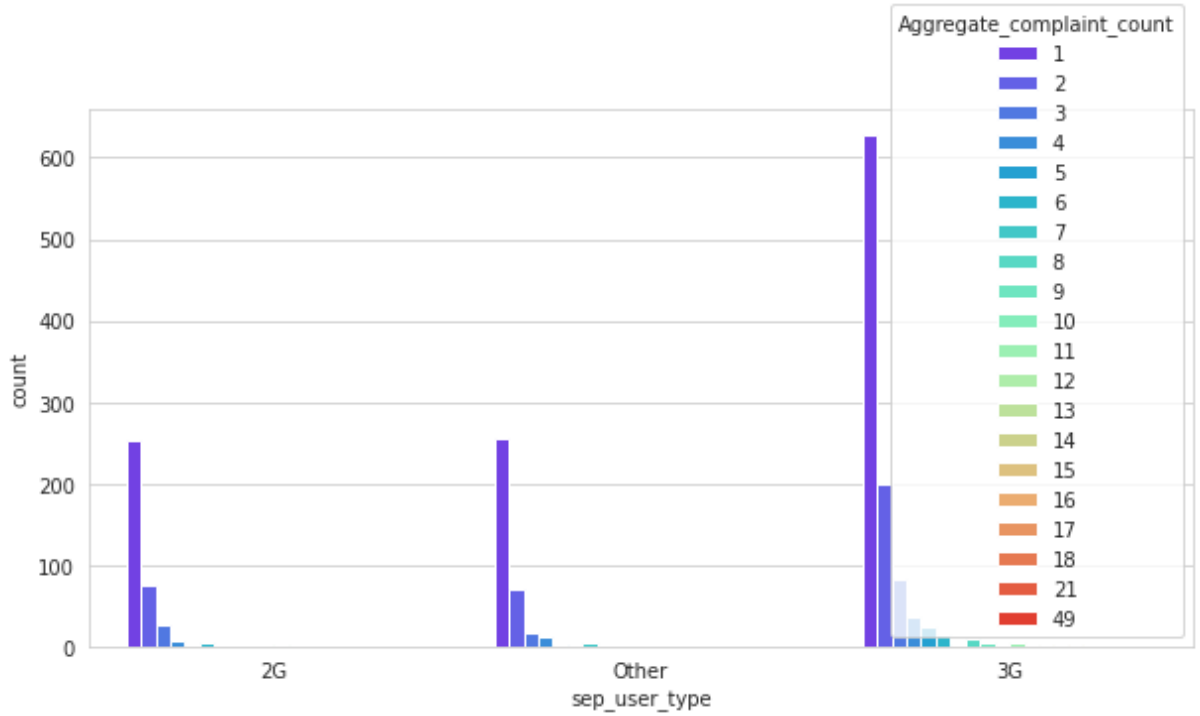


Figure 14 Count plot visualization of relation between September user type and aggregate complaint count

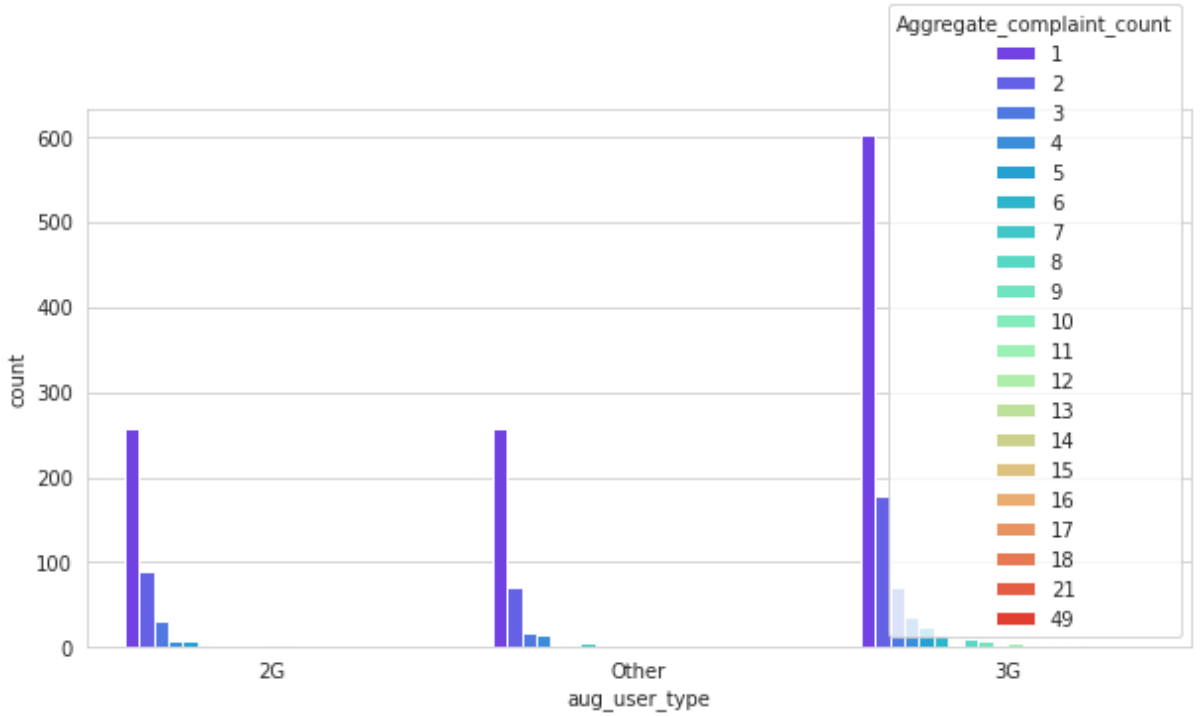


Figure 15 Count plot visualization of relation between August user type and aggregate complaint count

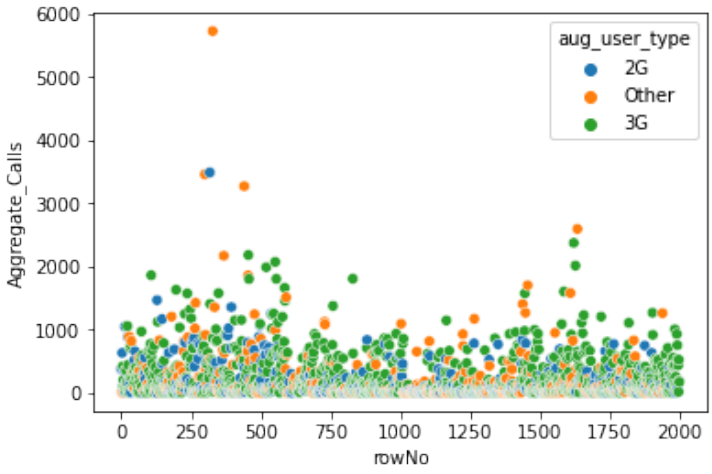


Figure 16 Scatter plot visualization of relation between August user type and aggregate calls

Handling Missing Values

The missing values in the dataset were in two columns aug_user_type and sep_user_type. The sep_user_type column was dropped considering observation 10, made from Fig 13. The aug_user_type feature had data stored as object type as 2G, 3G, and Other. To execute a program to handle the missing data through statistical approach, the data values in column had to be converted into integer type. It was done by first converting object type data into categorical type and then it was encoded into integers using label encoding assigning ‘0 for 2G’, ‘1 for 3G’ and ‘3 for others.

The problem arises when Nan/null values were encoded too as -1. They had to be replaced again by Nan/null. No clear inference could be made by Fig 16. By analyzing Fig 15., an estimation of mode was used to fill in the missing data values and to make sure that the missing values had been handled, a heat map was plotted (Fig 17.).

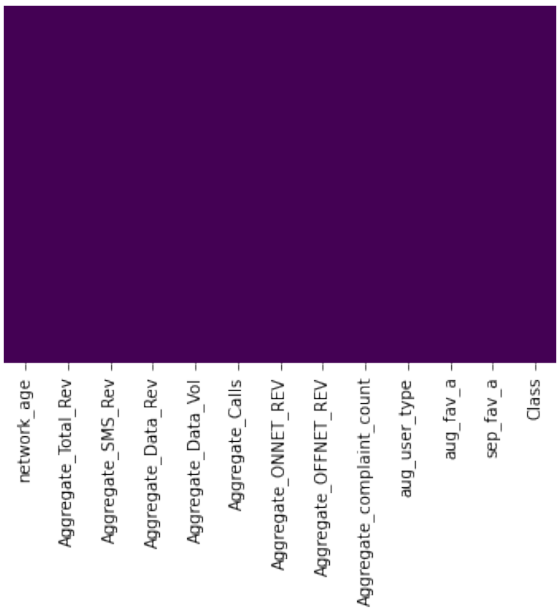


Figure 17 Heat map indicating that all the missing data has been handled

Model Implementation

- **Before implementing models** - It was ensured that the dataset comprised of only numerical values. This being a prerequisite for model implementation was ensured and implemented and now the input dataset was a 1999 x13 matrix after dropping the sep_user_type, feature column.

- **Using traditional classifiers for churn prediction model preparatio**n - To proceed, integer location-based indexing of input features and the output were performed using iloc (integer-location) indexer from pandas and then the data was split into 70% training and 30% test data. The machine learning algorithms that were used to build different models were AdaBoost, Logistic Regression, Gradient Boosting, Support Vector Machine (SVM), Random Forest, Stochastic Gradient descent, Gaussian Naive Bayes, K- Nearest Neighbors (KNN), and Decision Tree - used to build different models, so that we may compare the accuracy of every model with one another and the TPOT recommended model to analyze.

The accuracy of all the algorithms was calculated, shown in Table 1, where the scoring factor is the accuracy of the models. It was observed that Gradient Boosting algorithm outperformed others with an accuracy of 74.91%.

Classifiers	Accuracy (%)
Gradient Boosting	74.91
AdaBoost	74.40
Random Forest	71.91
Decision Tree	67.11
Logistic Regression	62.25
Gaussian Naïve Bayes	62.04
Support Vector Machine	58.04
Kneighbours	57.39
Stochastic Gradient Descent	52.25

Table 1 Accuracy of all the models implemented using different classifiers

- **Using TPOT** - Generations, population size, scoring, verbosity, n jobs and random sate are the parameters used for our pipeline. The number of generations was set as five because our datasets are small and not complex. A higher number of generations can be considered for more complex datasets. The population size was 30 in our case, hence TPOT generated 30 random models and sample hyper-parameters for the first generation while 20 jobs ran in parallel - as n jobs was set as 20, with the random state being 50.

The scoring feature for TPOT is accuracy by default and an accuracy of 99.92% (Figure 18) and cross validation score of 1 was obtained using an Extreme Gradient Boost (XGB) classifier model pipeline (Figure 19) suggested by TPOT, which constituted parameters automatically set to attain the maximum accuracy.

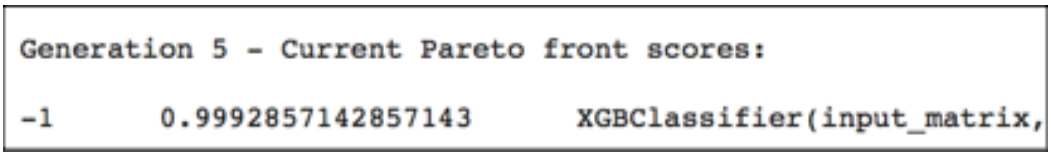


Figure 18 Accuracy of TPOT generated model pipeline

```
Pipeline(memory=None,
         steps=[('xgbclassifier',
                 XGBClassifier(base_score=0.5, booster='gbtree',
                               colsample_bylevel=1, colsample_bynode=1,
                               colsample_bytree=1, gamma=0, gpu_id=-1,
                               importance_type='gain',
                               interaction_constraints='', learning_rate=0.5,
                               max_delta_step=0, max_depth=4,
                               min_child_weight=7, missing=nan,
                               monotone_constraints='()', n_estimators=100,
                               n_jobs=1, num_parallel_tree=1,
                               objective='multi:softprob', random_state=50,
                               reg_alpha=0, reg_lambda=1, scale_pos_weight=None,
                               subsample=0.9500000000000001,
                               tree_method='exact', use_label_encoder=True,
                               validate_parameters=1, verbosity=0))],
         verbose=False)
```

Figure 19 Pipeline of TPOT generated model

Telecom Churn dataset

Feature Extraction

The detail about the features is as follows:

- State (String): There are 51 unique states in this dataset. All are from the United States of America.
- Account Length (integer): The length of account of customers.
- Area Code (integer): Three area codes, 415, 408 and 510 for San Francisco, San Jose, and the city of Oakland, respectively.
- International Plan (string): Subscription of international plan indicated by Yes/No indicators.
- Voice Mail Plan (string): Subscription of voice mail plan indicated by Yes/No indicators.
- Number vmail messages (integer): 0-50 range of total voice mail messages sent by the customer.
- Total day/eve/night minutes (double): Total usage of operator's services by the customer in the morning/evening/night, calculated in minutes.
- Total day/eve/night calls (integer): Total calls made by the customer in morning/evening/nighttime.
- Total day/eve/night charge (double): Total service charge added in the bill for the customer in the morning/evening/nighttime.

Exploratory Data Analysis

The Telecom churn dataset is already clean and does not have any missing values. The churn rate is 14% which is considered to be comparatively high. Visualization of relation between features was implemented for understanding of the dataset, to gather useful insights.

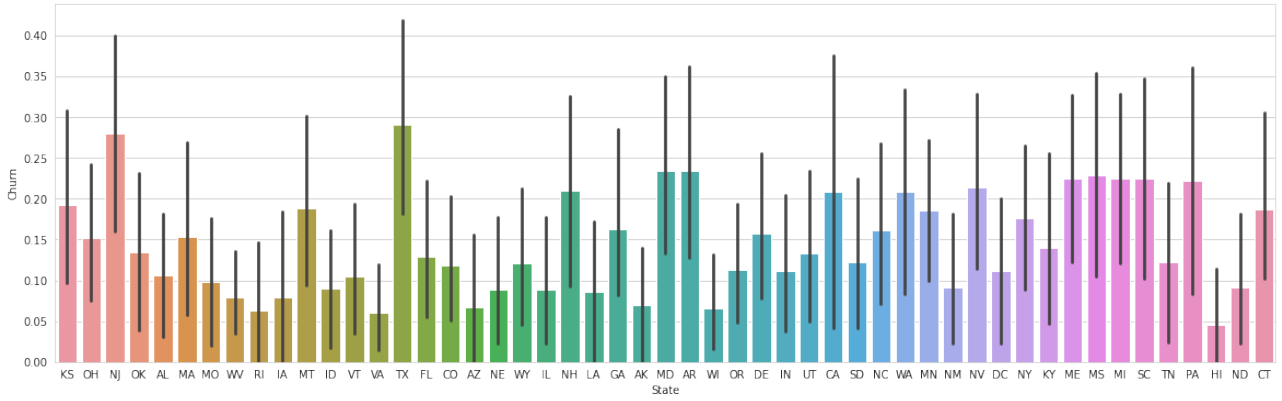


Figure 20 Bar plot visualization of relation between state and churn

Observations and Data

Observation 1: It can be analyzed (Fig 20.) that customers from Texas have churned the most and switched to other operators.

Observation 2: (Figure 21) The churned customers have called for customer service more than the customers who did not churn. The operator may work on improving its customer care services and providing prompt resolution.

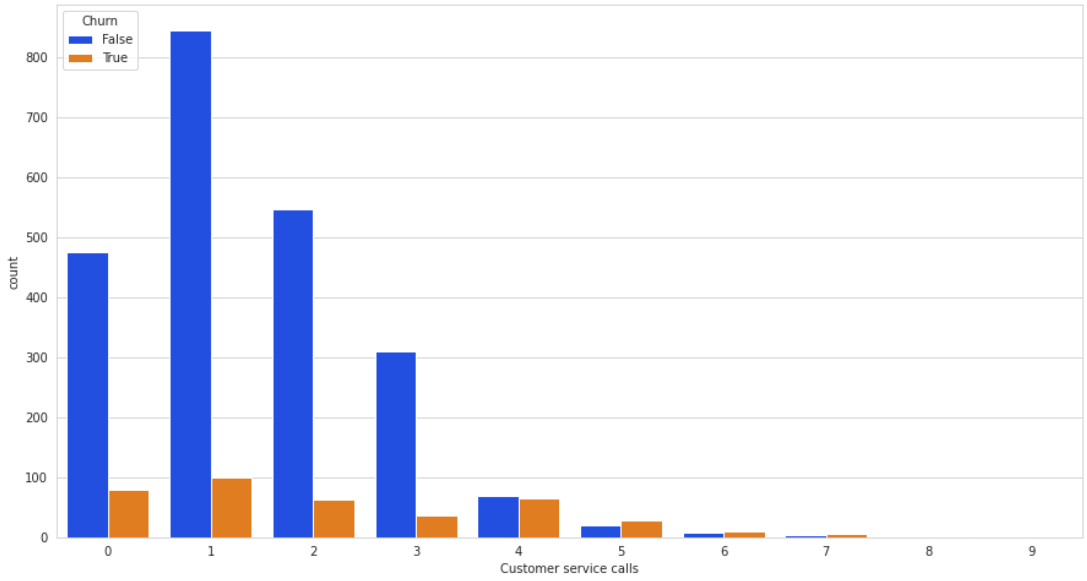


Figure 21 Barplot visualization of relation between customer service calls and churn

Observation 3: (Fig 22.) Account length, which is the duration for which the customers have been using the operator's services, is evenly distributed and not much can be inferred from it.

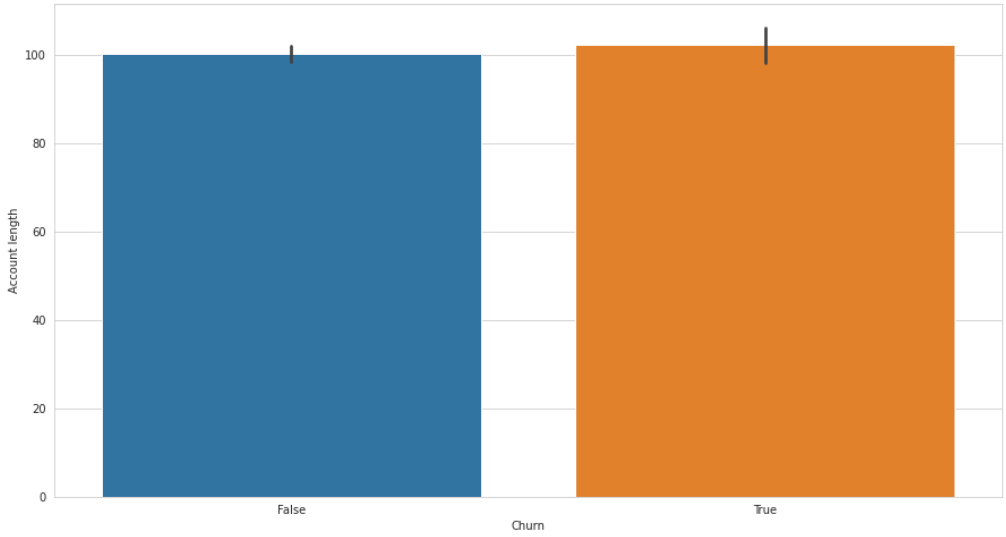


Figure 22 Barplot visualization of relation between account length and churn

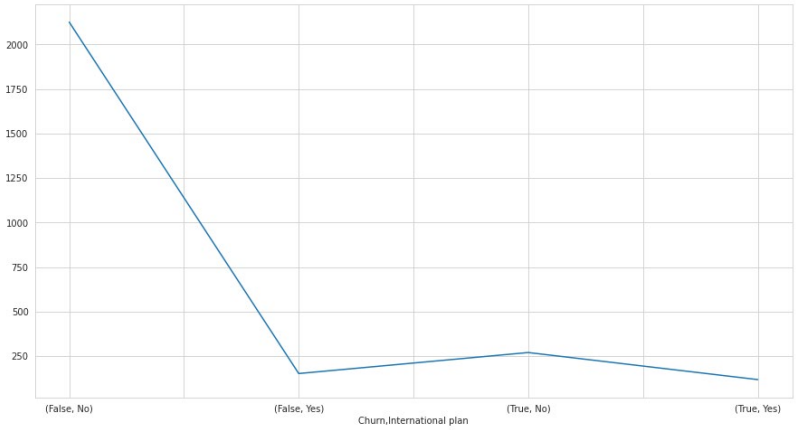


Figure 23 Plot visualization of relation between international plan and churn

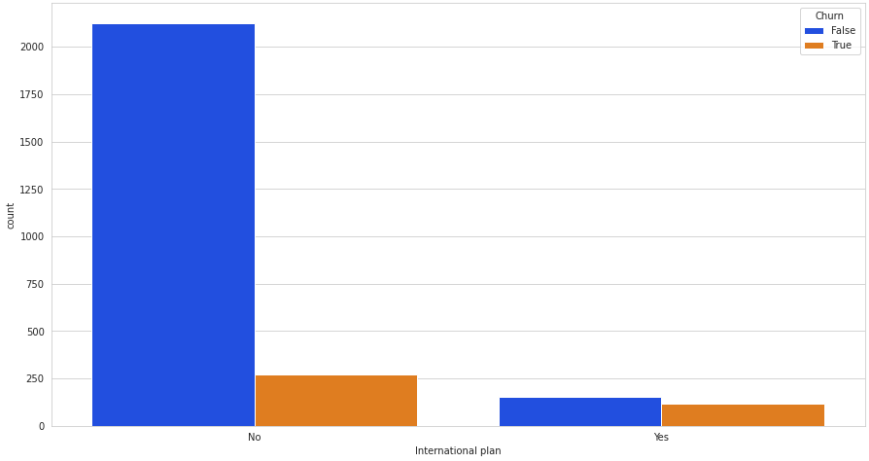


Figure 24 Countplot visualization of relation between international plan and churn

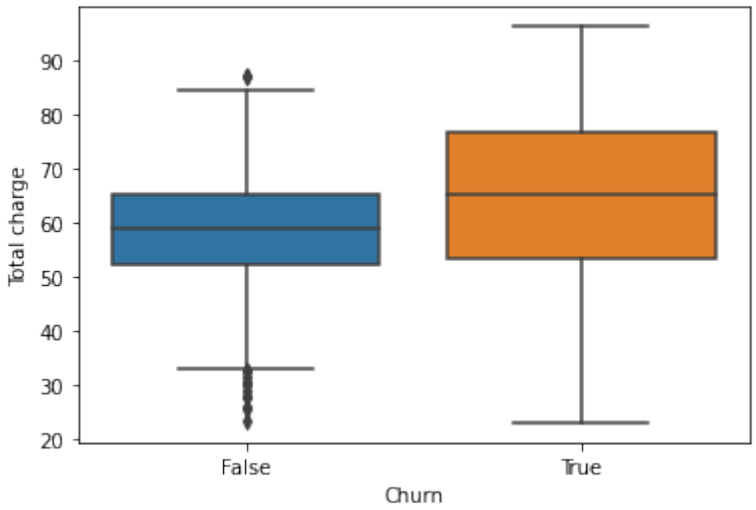


Figure 25 Boxplot visualization of relation between total charge and churn

Observation 5: Most of the customers who churned (Fig 25), were paying high charges for the operator's services. The operator may facilitate services and new offers for his customers.

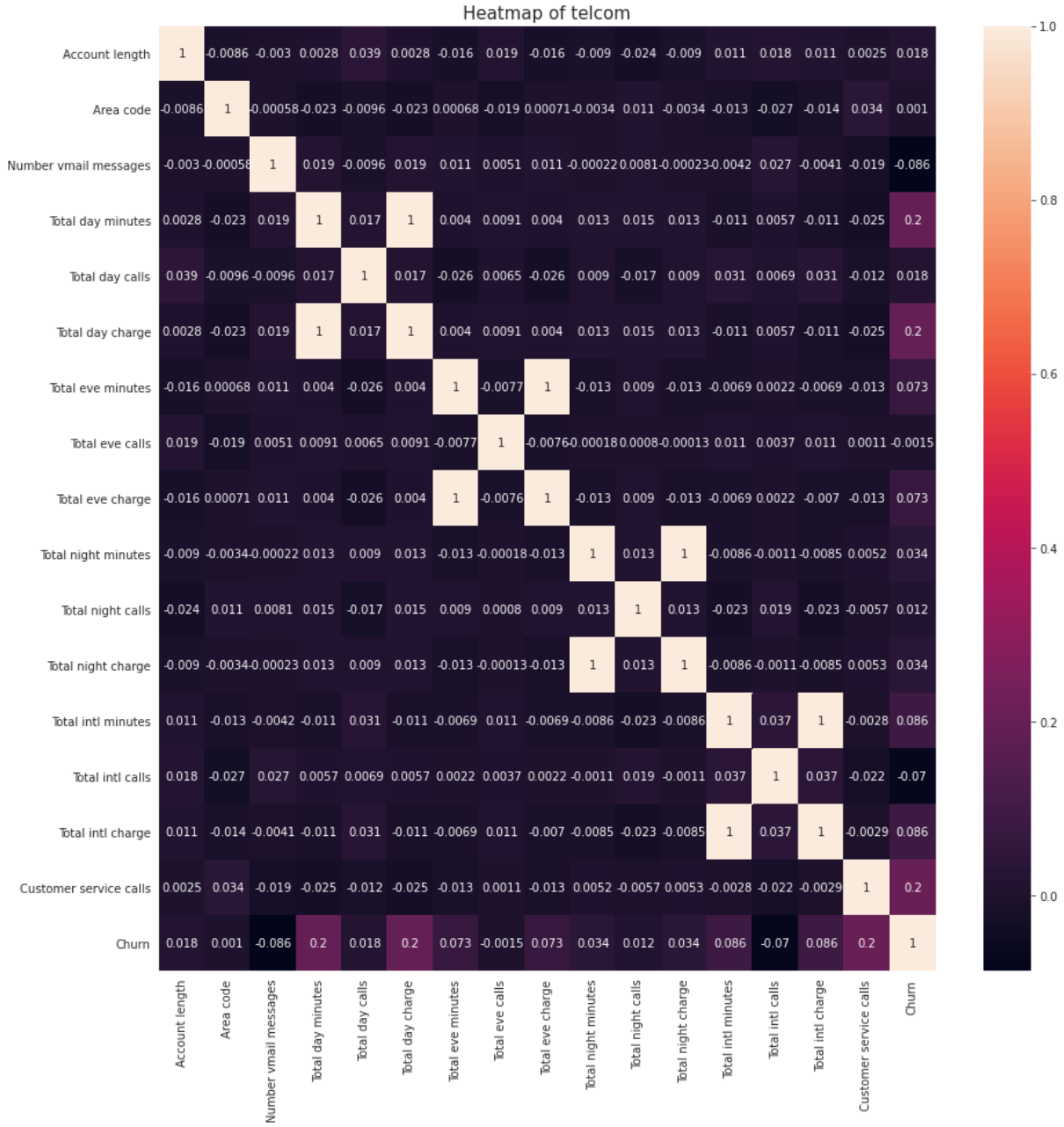


Figure 26 Heatmap for Pearson correlation of features

Observation 6: The features - total day charge, total eve charge, total night charge, and total intl charge are highly correlated (Fig 26).

Data Preprocessing

The features - state, area code, total day charge, total eve charge, total night charge, and total intl charge were either highly correlated or not especially useful and were dropped. All the features were label encoded for successful conversion into numeric values, this being a prerequisite for training models.

The 80% test and 20% train datasets were combined for duplication of columns, for multi value columns and were then separated again. All the numerical values were scaled original values dropped for the purpose of

merging scaled values for numerical columns. Our dataset was then ready to be used for the model implementation.

Model Implementation

- **Before implementing models -** it was ensured that the dataset comprised of only numerical values as this is a prerequisite for model implementation. After implementation, the training dataset was a 2666 x 14 matrix, and the testing dataset was a matrix of 667 x 14 after data preprocessing.

- **Using traditional classifiers for churn prediction model preparation** -The nine classifier models were implemented following the exact same procedure as that for The South Asian Churn Dataset. The maximum accuracy of 95.01% was again achieved using Gradient Boosting algorithm (Table 2).

CLASSIFIERS	ACCURACY (%)
GradientBoosting	95.01
AdaBoost	87.13
Random Forest	92.68
Decision Tree	90.50
Logistic Regression	86.30
Gaussian Naive Bayes	86.30
Support Vector Machines	90.28
Kneighbours	87.95
Stochastic Gradient Decent	85.40

Table 2 Accuracy of all the models implemented using different classifiers

- **Using TPOT** - Again, an exactly similar model for TPOT was implemented as for The South Asian Churn Dataset, by running a total of 20 jobs in parallel with 5 generations for a total population of 30. The best performing model pipeline for stacking estimator, while the estimator being Random Forest Classifier, with an accuracy of 95.18% (Fig 27) and cross validation score of 95.95% was suggested by TOPT (Fig 28).
-

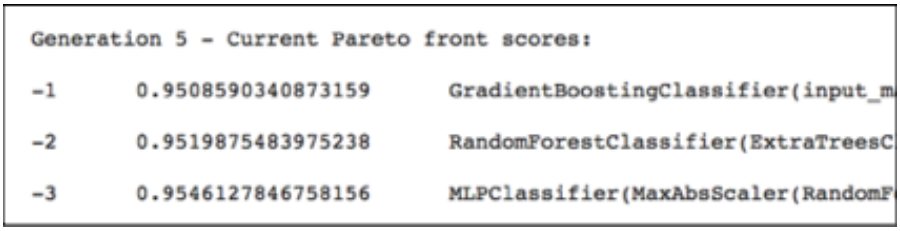


Figure 27 Accuracy of TPOT generated model pipelines

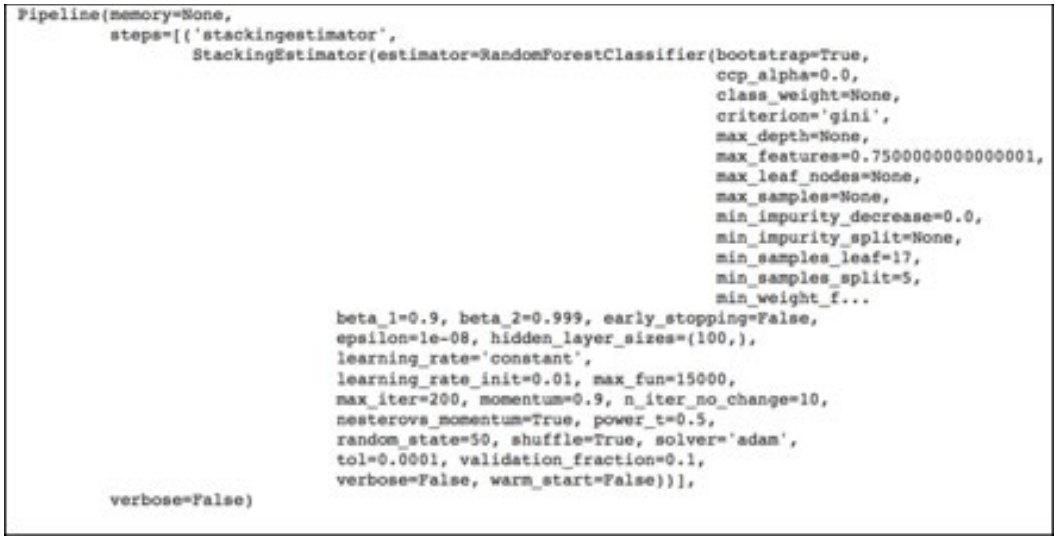


Figure 28 TPOT suggested model pipeline

Conclusion

Data analytics and predictive analytics were performed for both the datasets. It was observed that the models implemented and suggested by TPOT outperformed the traditional machine learning models that we used for prediction, and even various techniques used by several authors in various research, concluding that TPOT and other AutoML tools may be a better choice if the user is aiming to achieve results with maximum accuracy and efficiency for customer churn prediction datasets.

A maximum accuracy of 99.92% was obtained using the XGB Classifier pipeline suggested by TPOT for the South Asian Churn Dataset, whereas a maximum accuracy of 95.19% was obtained using the stacking estimator RandomForest Classifier pipeline model, suggested by TPOT for the Telecom Churn dataset.

Table 3. and Table 4. show comparison between various models implemented in our research for South Asian Churn dataset and Telecom Churn Dataset, respectively. Table 5. and Table 6. depict the comparison between the results achieved by different authors and us, for the South Asian Churn dataset and Telecom Churn Dataset, respectively.

CLASSIFIERS	ACCURACY (%)
GradientBoosting	74.91
AdaBoost	74.40
Random Forest	71.91
Decision Tree	67.11
Logistic Regression	62.25
Gaussian Naive Bayes	62.04
Support Vector Machines	58.04
Kneighbours	57.39
Stochastic Gradient Decent	52.25
TPOT: XGB pipeline	99.92

Table 3 Results for TPOT vs other classifiers for South Asian Churn Dataset

CLASSIFIERS	ACCURACY (%)
GradientBoosting	95.01
AdaBoost	87.13
Random Forest	92.68
Decision Tree	90.50
Logistic Regression	86.30
Gaussian Naive Bayes	86.30
Support Vector Machines	90.28
Kneighbours	87.95
Stochastic Gradient Decent	85.40
TPOT: Stacking Estimator Random Forest	95.19

Table 4 Results for TPOT vs other classifiers for Telecom Churn Dataset

[Reference] Year	Model Implementation	Accuracy (%)
[11] 2010	Decision Tree	75
[12] 2011	Neural Network	72
[13] 2012	Random Forest	68
[14] 2014	Hybrid model of Voted Perceptron and Logistic Regression	59
[15] 2014	Multi-Layer Perceptron	73
[16] 2015	Boosted Support Vector Machine	70
[5] 2017	Multiple Classifier System	86.3
2021	Tree-based pipeline optimization tool	99.92

Table 5 TPOT vs previously proposed models for South Asian Churn Dataset

Reference/Year	Model Implementation	Accuracy (%)
[17] 2018	Random Forest	87.47
[9] 2019	Decision Tree	89.40
2021	Tree-based pipeline optimization tool	95.19

Table 6 TPOT vs previously proposed models for Telecom Churn Dataset

Considering the performance of TPOT, for future work - various other AutoML models can be put into use for improved customer churn prediction results. The notebooks for code implementation of both the datasets can be found in the following repository: https://github.com/ShriyaAvasthi?tab=repositories .

Bibliography

[1] Idris, A., Khan, A., Lee, Y. S. (2012). Genetic Programming and Adaboosting based churn prediction for Telecom. 2012. IEEE International Conference on Systems, Man, and Cybernetics (SMC), pp. 1328–1332.
[2] Huang, F., Zhu, M., Yuan, K., Deng, E.O. (2015). Telco churn prediction with big data. In: ACM SIGMOD International Conference on Management of Data, pp. 607–18.
[3] Brandusoiu, I., Toderean, G., Ha, B. (2016). Methods for churn prediction in the prepaid mobile telecommunications industry. International Conference on Communications (COMM)., pp. 97-100.

[4] Makhtar, M., Nafis, S., Mohamed, M., Awang, M., Rahman, M., Deris, M. (2017). Churn classification model for local telecommunication company based on rough set theory. Journal of Fundamental and Applied Sciences. 9(6), pp. 854 -868.

[5] Ahmed, M., Afzal, H., Siddiqi, I., Khan, B. (2017). MCS: Multiple Classifier System to predict the churners in the telecom industry. Intelligent Systems Conference (IntelliSys), pp. 678-683.

[6] Mahreen Ahmed, South Asian Churn Dataset: Balanced dataset for predicting churners in telecom industry, 2017, Kaggle, v1, https://www.kaggle.com/mahreen/sato2015/metadata

[7] Olson R.S., Moore J.H. (2019) TPOT: A Tree-based Pipeline Optimization Tool for Automating Machine Learning. In: Hutter F., Kotthoff L., Vanschoren J. (eds) Automated Machine Learning. The Springer Series on Challenges in Machine Learning. Springer, Cham, pp. 151-160.

[8] Mandić,M.,Kraljević, G. (2020). Two-Layer architecture of churn Auto-Ml. 31st DAAAM International Symposium on intelligent manufacturing and automation. 0788-0792.

[9] Mnassri, B., Churn Prediction with PySpark. [Accessed: 08-2022] Available at: https://github.com/mnassrib/churn-prediction-with-pyspark/blob/master/churn-prediction.ipynb

[10] Baligh Mnassri, Telecom Churn Dataset: Cleaned Orange Telecom Customer Churn Dataset, 2019, Kaggle, V1, https://bml-data.s3.amazonaws.com/churn-bigml-80.csv&https://bml-data.s3.amazonaws.com/churn-bigml-20.csv

[11] Kraljević, G., Gotovac, S. (2010). Modeling Data Mining Applications for Prediction of Prepaid Churn in Telecommunication Services. Automatika. 51(3), pp. 275-283.

[12] Sharma, A., Panigrahi, P. K. (2011). A Neural Network based Approach for Predicting Customer Churn in Cellular Network Services. International Journal of Computer Applications. 27(11), pp. 26-31.

[13] Xiao, J., Xiao, Y., Huang, A., Liu, D., Wang, S. (2015). Feature-selection-based dynamic transfer ensemble model for customer churn prediction. Knowledge and Information Systems. 43, pp. 29-51.

[14] Olle, G., Cai, S. (2014). A Hybrid Churn Prediction Model in Mobile Telecommunication Industry. International Journal of e-Education, e-Business, e-Management and e-Learning.4(1), pp.55-62.

[15] Brandusoiu, I.B., Toderean G. (2014). A Neural Networks Approach for Churn Prediction Modeling in Mobile Telecommunications Industry. Annals of the University of Craiova. 11(1), pp. 9-16.
[16] Vafeiadis, T., Diamantaras, K.I., Sarigiannidis, G., Chatzisavvas, K.C. (2015). A Comparison of Machine Learning Techniques for Customer Churn Prediction. Simulation Modelling Practice and Theory. pp. 55, 1-9.

[17] Karim, M. R., 2018. Scala Machine Learning Projects: Build Real-world Machine Learning and Deep Learning Projects with Scala.