Customer Churn Prediction for VeriTel

One of the Largest Telecom Provider

FINAL PROJECT REPORT

11-OCT-2020

Prepared and Submitted by:

Group 10

As part of Final Evaluation

Towards fulfillment of APDS 03 LDP at IIM Calcutta

Project Guide -

Prof Soumyakanti Chakraborty

Associate Professor, MIS

IIM Calcutta

Group 10 -

Anant Krishna

Bharath Ayanampudi

Chinmay Majee

Debapratim Ghosh

Kunal Chandra

Santosh Srivastava

Satish Chilloji

Thejovardhan Pammi

TABLE OF CONTENTS

Contents

TABLE OF CONTENTS

Project Introduction and Scope

INTRODUCTION

Telecom customer churn prediction is to estimate subscribers who may cancel subscription. In recent years, estimating churners before they leave has become valuable in the environment of increased competition among companies.

Customer churn has become very important because of increasing competition among companies, increased value of marketing strategies and conscious behavior of customers in the recent years. Customers can move toward alternative services. Companies must develop strategies to such possible trends, depending on the services they provide

LITREATURE REVIEW

Many researches have been conducted to increase the prediction rates of costumer churns in the telecommunication industry.

Some researchers used two different hybrid models to develop a customer churn prediction model. The developed hybrid model may be a combination of two neural networks and the second hybrid model is a combination of self-organizing maps and artificial neural networks. First models are used for data reduction and second models are used for actual classifier [5].

Some used attribute derivation process to improve correct prediction rate [2]. Bayesian Belief Network method is tried in a study [1]. Others increased the accuracy by using two different rules extraction method [6]. And, rotation-based ensemble classifiers. These are Rotation Forest and Adaboosts [7]. Yet another is decision tree and genetic programming [8]. We can also use one class support vector machine to increase the performance [3]. Other hybrid model consider is by combining neural network, tree models and fuzzy modeling [9].

PROJECT BACKGROUND

VeriTel is the second largest telecom provider in the world with operations in over 15 countries directly and in other 21 countries with a partner. The company is a leader in providing telecom services in both B2B and B2C space. Although the company has been doing well in last 3-4 years, the revenue of the company seems to have been almost plateau like and stagnating. The CEO of VeriTel hypothesizes that this stagnation is mostly due to a large number of customers churning out of their subscriptions. Given the fact that the telecommunications industry experiences an average of 30-35 percent annual churn rate and it costs 5-10 times more to recruit a new customer than to retain an existing one, customer retention has now become even more important than customer acquisition. In wireless network telecommunication industry, the monthly rate of customer churn is 2.2% and the annual rate of customer churn is 27% [2]. The yearly cost of customer churn is 4 billion dollars in Europe and America, and it is 10 billion dollars in the entire world [2]. We may suppose that 1.5 million customers would stay in the same company by increasing the correct prediction at the rate of 1%. This may yield to 54 million dollars benefit for the companies annually [3]. In the Churn model once we know *Who are the customers that are going to churn*, the next question would be *Why these customers are churning*. Over the period of time, it would be important to know the root cause for the churn. In order to manage churn reduction, not only do we need to predict which customers are at high risk of churn, but also what are the reasons why the customers are churning out. Therefore, the company can optimize their marketing intervention resources to prevent as many customers as possible from churning. The CEO wants to deploy retention strategies in synchronizing programs and

processes to keep customers longer by providing them with tailored products and services. With retention strategies in place, the CEO wants to include churn reduction as one of their business goals.

PROJECT OBJECTIVES

- Predict with reasonable accuracy, the probability of a customer churning out using classification techniques
- To compare performance and Accuracy of different Machine Learning models
- To find out important variables that contribute to model prediction and highlight features that have a propensity to churn using an Explainable technique like SHAP

SCOPE OF PROJECT

We are following the traditional ML approach to find out who are the customers that will churn out. The data is divided into 70-30 split and the model is trained on the 70% of the data and predicts on 30% of the data. As we have limited data options, we have performed feature engineering to boost the accuracy of ML model to improve the reliability of the prediction. Post selection of the ML Model, we used one of the explainability technique to find out which features are impacting model prediction also explains reasons for that particular model's specific prediction. We used different techniques like Logistic regression, boosting, bagging and neural networking to develop a customer churn prediction model. Firstly, we used various approaches for data preparation. Data cleansing approaches such as type checking and consistency of attributes, identifying missing values for features, removing unused features, normalizing the features. We also used attribute derivation process to increase the correct prediction rate. We used various feature engineering techniques to boost the dataset and evaluated classifier performance based on the feature rich dataset. Depending on the outcome, we re-engineered the features, and at times, eliminated a few features and/or retained some others.

Data Collection Method and Sources	The data is sourced from a kaggle dataset Link : https://www.kaggle.com/abhinav89/telecom-customer
Data Size	100,000 rows with 100 customer attributes for each customer till a particular year
Statistical Methodology	**For Customer Churn Prediction:** Classification techniques like Logistic Regression, Random Forest, Light GBM, Neural Networks etc. **For ML Model Eplainable technique:** SHAP
Tools to be used for analysis	Github, Tableau, R, Python Jupyter Notebook, MS Excel

OUT OF SCOPE

- The current dataset does not include the time-series features. So, we will not be able to look at temporal aspects and find out how soon customers will churn out.
- The final ML model selected for predicting churn will not be an Ensemble Model because we will be expecting the VeriTel CEO to observe the ML model Performance and its expalinability of the individual models and then select the one that is more sensible to business
- Customer segments that are vulnerable to churn and recommendations for Customer segments

ANTICIPATED OUTCOMES

- ***Who ?*** - Gauge the probability of new customers to churn out or cancel their subscriptions
- ***Why ?*** – Determine Important features that impact the Customers Churn Prediction

Visualizing and Analyzing the Data

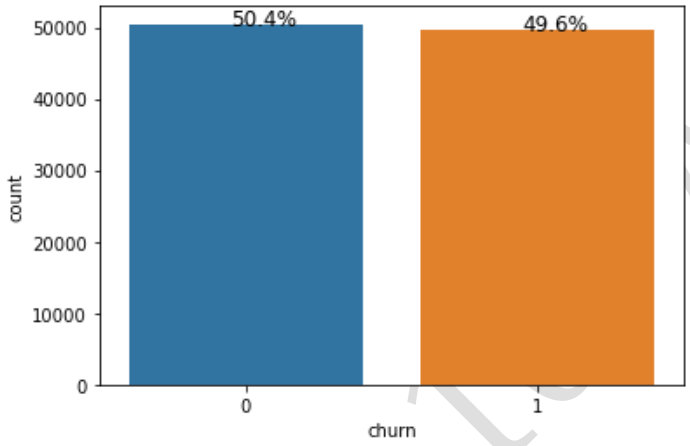
The following Table provides a summary of the data for telecom churn customers for various fields

STRUCTURE OF DATA

Data Type	No. of Columns	No. of Rows
Numerical Columns	79	100,000
Categorical Columns	21	100,000
Total	100	100,000

DISTRIBUTION OF CHURN AND NOT CHURNED CUSTOMERS

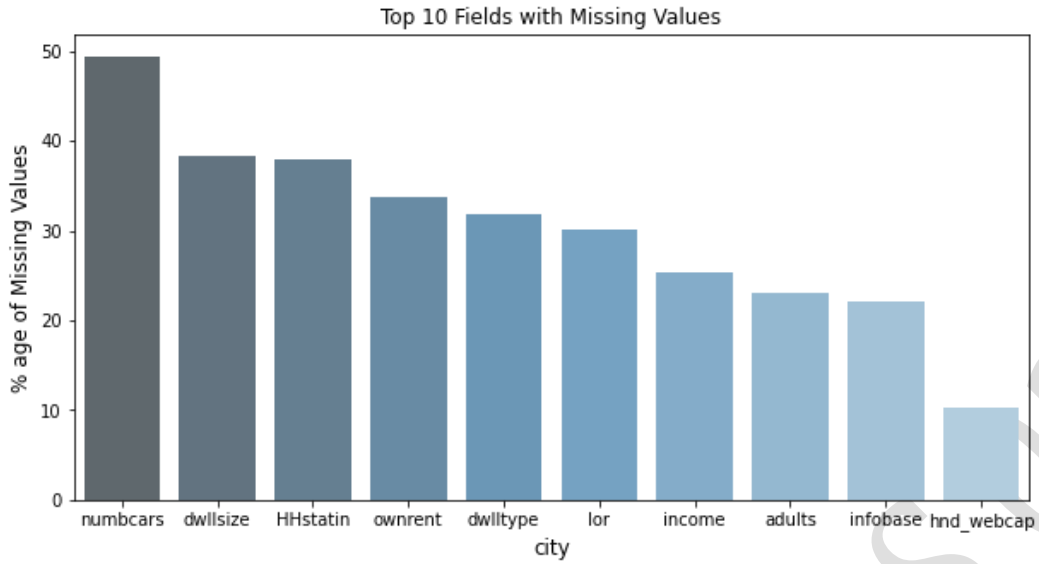
From below Figure, we can ascertain that the dataset is a balanced one with ~50% of customers churning and the remaining not churning. This implies that we don't have to employ any sampling technique (under sampling or oversampling) to drive performance gains.



MISSING VALUES IN THE COLUMNS

From below Figure, it can be surmised that the most of the missing values is in the Personal customer attributes like number of cars, dwelling size, length of residence etc. Among the top 10 variables with missing values, 3 are numerical variables while the rest are categorical. Usually, Personal information of the customers are not updated often with the telecom records or customers do not provide this data at all to the telecom companies. As a result, most of the data might be incomplete due to the usage of sparse and disparate data sources. So in order to proceed with model building, we would need to replace the missing values.

Different ways can be explored while imputing or replacing the missing values in the fields. For Numerical variables, simply imputing the missing values with the corresponding mean of the field can drive performance gain significantly. Another method of imputation would be to use a simple or iterative imputer. Similarly for categorical variables, a simple replacement with the mode would suffice and help in improving performance. Additionally, using package based imputers might also help in imputing values.



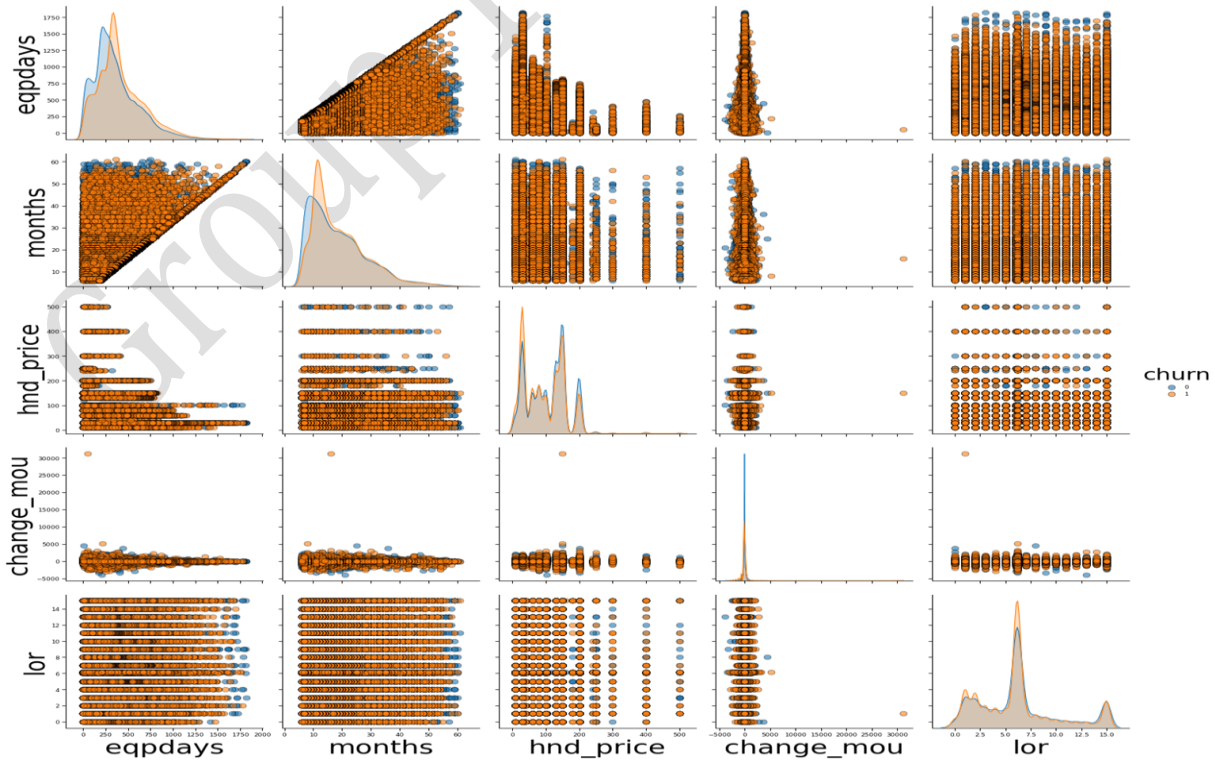
SCATTER PLOT OF RELEVANT NUMERICAL VARIABLES AND CORRELATION ANALYSIS

Numerical Variables

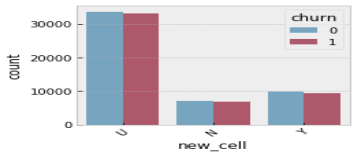
From below Figure, it can be seen that the age of current equipment and number of months of service are somewhat normally distributed. Additionally, we can see that there is a linear relationship between the age of equipment and months in service which is self-explanatory.

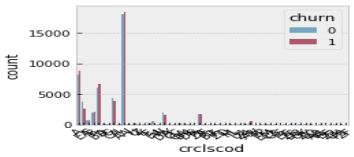
Categorical Variables

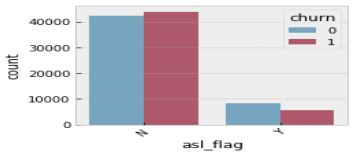
Looking at the categorical variable's relationship with churn, it is evident that the proportion of churned customers is equal to not churned customers for each value of the categorical variables. This confirms the hypothesis that there is no special concentration of churned customers across categorical values.

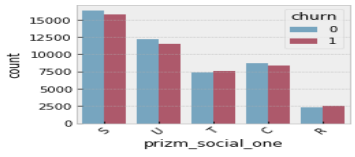


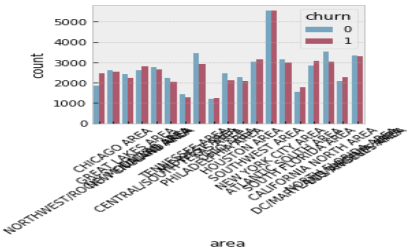
RELATIONSHIP OF CATEGORICAL VALUES WITH CHURN

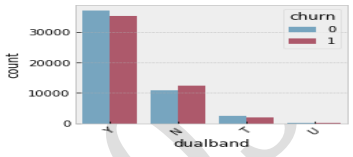


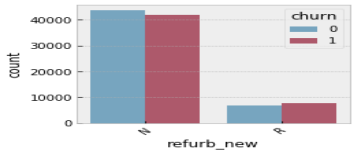


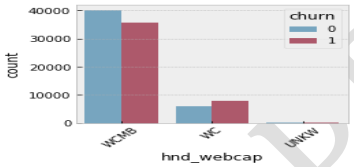


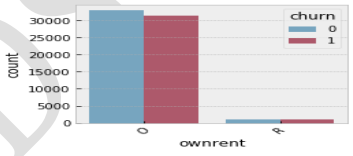


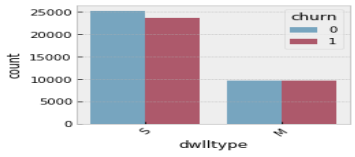


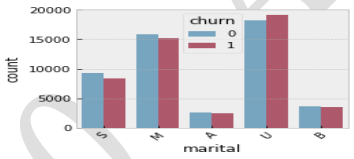


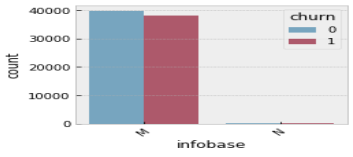


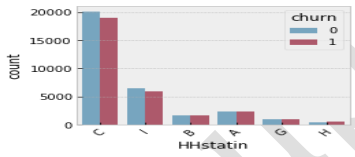


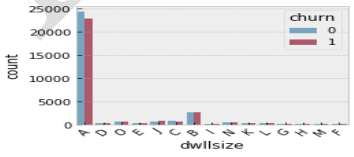


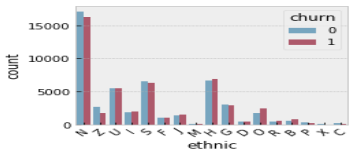


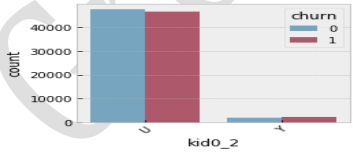


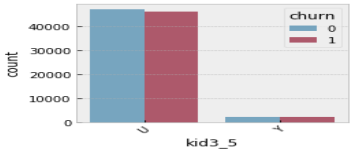


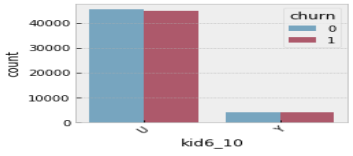


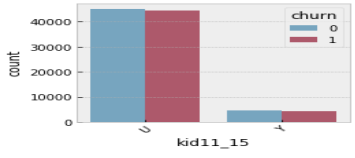


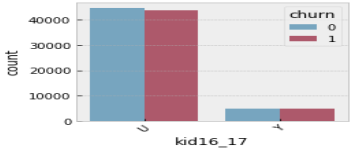


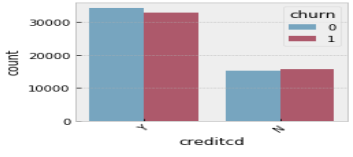






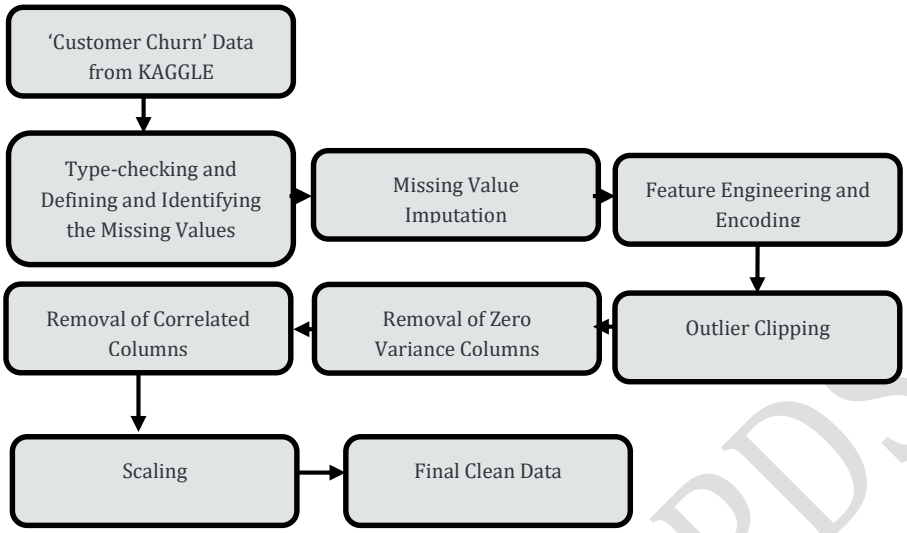






Data Preparation, Analysis and Manipulation

Our data preparation steps are outlined in the following diagram:



DATA CLEANING APPROACHES FOR THE CUSTOMER CHURN DATASET

TYPE CHECKING AND CONSISTENCY OF THE FIELDS

The very first step was validating the data sanity. No field with inconsistent data type was found in our base data set. Values across the columns were consistent in the sense that there was no type mismatch between the values in the same column.

DEFINING AND IDENTIFYING THE MISSING VALUES

We programmatically identified the missing values in the columns. We found almost 1.57% of the cell values of the entire dataset as missing or 'NA'. We imputed the missing values using various algorithm as explained later in this section. Data Imputation for Missing Value

DATA IMPUTATION FOR MISSING VALUES

Imputation is the process of estimating or deriving values for fields where data is missing. There is a vast body of literature available on various imputation techniques. For our data-preparation, we took scenario-based approaches for each field/feature that had missing values. Below Figure schematically outlines our data imputation techniques that we adopted.

Data Imputation for Numerical Variables

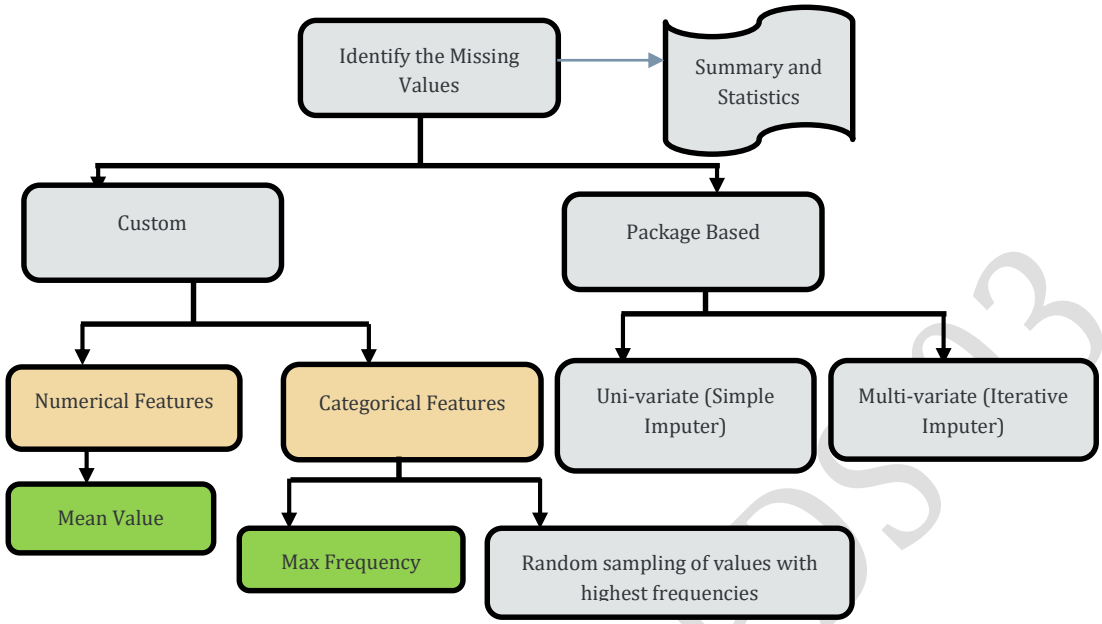
Mean: For numerical variables, we used 'mean' of the non-missing values in the feature dimension to impute the missing values.

Data Imputation for Categorical Variables

For categorical variables we used the value with highest frequency and at times randomly sampled from the values with highest frequencies to impute the missing values.

Package Based Data Imputation

We also selectively used the simple imputer and iterative imputers from Python Sci-kit learn packages for imputation of missing values for uni-variate and multivariate features respectively.



•Uni-variate Feature Imputation: The Simple Imputer class from sci-kit learn package provides basic strategies for imputing missing values leveraging a uni-variate algorithm. Missing values can be imputed with a provided constant value, or using the statistics (mean, median or most frequent) of each column in which the missing values are located.

•Multivariate Feature Imputation: The Iterative Imputer class from sci-kit learn package imputes missing values leveraging a multivariate algorithm. A multivariate feature imputation algorithm models each feature with missing values as a function of other features, and uses that estimate for imputation. It does so in an iterated round-robin fashion: at each step, a feature column is designated as output y and the other feature columns are treated as inputs X. A regressor is fit on (X, y) for known y. Then, the regressor is used to predict the missing values of y.

FEATURE ENGINEERING

We used various feature engineering techniques to boost the dataset and evaluated classifier performance based on the feature rich dataset. Depending on the outcome, we re-engineered the features, and at times, eliminated a few features and/or retained some others

FEATURE ENGINEERING FOR NUMERICAL FEATURES

For numerical features, we adopted an iterative approach to identify and retain the best features that we believed will enrich our dataset and help in classification. We leveraged following 3-steps approach:

Deriving new features based on existing features using aggregate functions and statistics

We tried to derive new features based on the existing features that presumably could enrich the feature set. We created new features with the statistics aggregated by the parent variables for all numeric features. Each observation of the parent variable will have one row in the dataset with the parent variable as the index followed by removal of duplicate values. The derived features were then fed to an iterative flow of

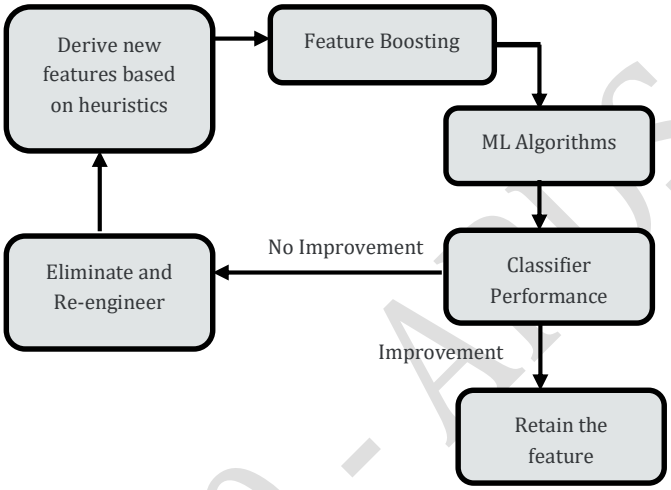
classifier performance evaluation and decision making as to whether the feature will be retained or discarded. An example of how we derived a few custom features is shown below:

Evaluate the Classifier Performance based on the new features

We then evaluated the classifier performance based on the input dataset enriched with the new features and checked whether there is any marked improvement, and in case there was no marked improvement we discarded the feature as redundant

Finalize the Feature Set

We iteratively evaluated the classifier performance and finalized the incremental feature set based on the classifier performance in each iteration.



FEATURE ENGINEERING FOR CATEGORICAL FEATURES

Encoding

We first encoded the categorical features using following encoding schemes for categorical variables

Frequency Encoding: For some of the nominal features, we used frequency encoding.

One-Hot encoding: For non-ordinal categorical variables, we used one-hot encoding to split columns with multiple categorical values into multiple columns with values '0' and/or '1'

Feature Extraction based on Feature Interactions

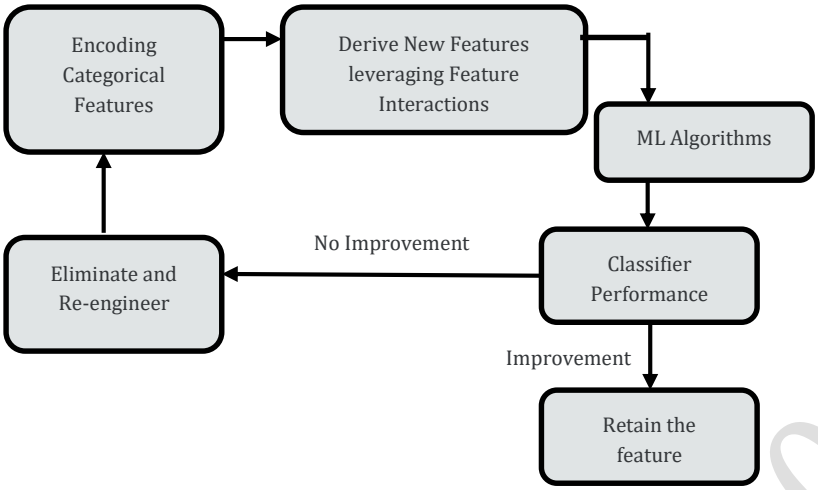
We attempted to derive some new categorical variables based on feature interaction between encoded categorical variables

Evaluate the Classifier Performance based on the new features

Similar to numerical feature engineering, we evaluated the classifier performance based on the enriched dataset to determine whether we would retain the categorical feature or drop based on the improvement/deterioration of the classifier performance level.

DERIVED FEATURES

Based on our feature engineering approach elucidated earlier, we have derived following new features to enrich our dataset. The detailed derivation is listed out in Appendix-2.



Mean Talktime per unit Charge	Mean Extra Amount that Customer Pays	Mean Monthly Profit	Mean Allocated Calls
Mean Failed/Dropped Calls	Percentage of successful calls	Unrounded to rounded completed voice call difference	Unrounded to rounded completed received voice call difference
Mean total revenue per call	Mean Total minutes of use per call	Mean total charge per call	Total Revenue Adjustment
Total minutes of use adjusted	Total calls adjusted	Average revenue per call	Average minute of use per call
Average minute of use per call	Average charge per call		

OUTLIER CLIPPING

We removed outlier outside a threshold for some identified features to further normalize the dataset. The standard threshold that we have used is 5% on both sides for most of the numerical features.

```
def data_outlier(df,low_cut,high_cut):
    numerics = ['int16', 'int32', 'int64', 'float16', 'float32', 'float64']
    numeric_col=df[df.columns.drop(list(['churn','Customer_ID']))].select_dtypes(include=numerics).columns
    i=0
    for i in range(len(numeric_col)):
        column=numeric_col[i]
        df.loc[df[column]>df[column].quantile(high_cut),column]=df[column].quantile(high_cut)
        df.loc[df[column]<df[column].quantile(low_cut),column]=df[column].quantile(low_cut)
```

REMOVAL OF ZERO VARIANCE COLUMNS

We programmatically inspected the data set for any zero variance columns, as zero variance column will not add any feature in determining the output classification. However, we didn't find any column having near zero variance (within a reasonable threshold). So, no column was eliminated out of this step.

```
# Removal of Zero Variance Columns
def rem_var(df,ther):
    numeric_df=df.select_dtypes(exclude='object')
    numeric_df=numeric_df[numeric_df.columns.drop(list(['churn','Customer_ID']))]

    #Remove near zero variance columns
    selector=VarianceThreshold(ther)
    selector.fit(numeric_df)

    numeric_df_1=df[numeric_df.columns[selector.get_support(indices=True)]]

    print ("Your selected dataframe has " + str(numeric_df.shape[1]) + " columns.\n"
           "Your selected dataframe has " + str(numeric_df.shape[1]) + " Numerical columns.\n"
            "There are " + str(numeric_df_1.shape[1]) +" columns has variance greater than " + str(ther) + " Percentage")

    object_df = df.select_dtypes(include='object')

    df_1 = pd.concat([df[['Customer_ID','churn']],df[object_df.columns],numeric_df_1], axis=1)

    return df_1
```

REMOVAL OF CORRELATED COLUMNS

We created the correlation matrix for all numerical fields to check if there is any significant correlation present between the fields and removed the fields that had significant correlation. Correlated features can impact the performance of the classifier, and removing these fields, help removing the redundancies from the feature set. We applied this step only for numerical variables.

```
#Removing the correlated Columns.
def correlated(df,val):

    #Only Numerical columns
    numeric_df=df.select_dtypes(exclude='object')
    numeric_df=numeric_df[numeric_df.columns.drop(list(['churn','Customer_ID']))]

    # Create correlation matrix
    corr_matrix = numeric_df.corr()
    # Select upper triangle of correlation matrix
    upper = corr_matrix.where(np.triu(np.ones(corr_matrix.shape), k=1).astype(np.bool))
    to_drop = [column for column in upper.columns if any(upper[column] > val)]
    numeric_df_1 = numeric_df.drop(to_drop, axis=1)

    #Categorical Columns
    object_df = df.select_dtypes(include='object')
    df_1 = pd.concat([df[['Customer_ID','churn']],df[object_df.columns],numeric_df_1], axis=1)

    gc.enable()
    del corr_matrix
    gc.collect()

    print ("Your selected dataframe has " + str(df.shape[1]) + " columns.\n"
           "In that dataframe has " + str(numeric_df.shape[1]) + " Numerical columns.\n"
           "There are " + str(len(to_drop)) +" columns has correlation greater than " + str(val) + " Percentage")

    print(numeric_df_1.shape)
    return df_1
```

SCALING

Various scaling mechanism are in use for Machine learning input dataset. The most commonly used ones are: Min-max normalization, Mean Normalization, Z-score normalization and Scaling to unit length. For our dataset, we used min-max normalization: $x' = \{ x - min(x) \}/\{ max(x) - min(x) \}$

Machine Learning Model Building

To select a suitable Churn Prediction model, we have explored following ML techniques:

LOGISTIC REGESSION

As the dependent variable is Binary, we started with Logistic Regression as the first ML technique as it is good in describing the data and explaining the relationship between Binary dependent variable and one or more Nominal/Ordinal/ Numerical independent variable. Simple Logistic regression is easy to interpret and statistically allows us to conduct the analysis.

```
#Logistic Regression Clasifier
logreg = LogisticRegression(solver='lbfgs',
                            class_weight='balanced',
                            penalty='l2')
logreg.fit(X_train, y_train)
```

Parameters

solver: lbfgs, Algorithm to use in the optimization problem.

class_weight: balanced, mode uses the values of y to automatically adjust weights inversely proportional to class frequencies in the input data.

penalty: l2, used to specify the norm used in the penalization.

RANDOM FOREST

As it is one of the best Bagging technique and often considers homogenous weak learners, it learns them independently from each other in parallel and combines them following some kind of deterministic average.

```
#Random Forest

rf_clf =RandomForestClassifier(random_state=42, oob_score = True, n_jobs= -2,
                               class_weight  = 'balanced', max_depth = 30,
                               n_estimators = 800, criterion = 'gini',
                               max_features = 'auto', verbose = 1)
rf_clf.fit(X_train, y_train)
```

Parameters

max_depth: The maximum depth of the tree.

n_estimators: The number of trees in the forest.

class_weight: balanced, mode uses the values of y to automatically adjust weights inversely proportional to class frequencies in the input data.

criterion: gini, The function to measure the quality of a split. Supported criteria are “gini” for the Gini impurity and “entropy” for the information gain.

max_features: The number of features to consider when looking for the best split.

NEURAL NETWORK

```
#create the neural network. Architecture - 3 hidden layers with 10, 5 and 3 neurons respectively
nn <- neuralnet(f1_nn,data=train_cln,hidden=c(10,5,3), threshold = 0.5)
```

Parameters

Based on the initial Logistic Regression with all fields, we identified the features that have P value <0.01 (ie. Significant at 99%), and took some of those features for building our Neural Network. We used only those features in training the Neural Network. With the entire feature set, the Neural Network training was not converging in a reasonable time frame and hence we used the reduced set of features. We used 'neuralnet' package (Fritsch, Guenther, Wright, Suling & Mueller) from CRAN to deploy the Neural Network. The no. of input features were 27. We used 3 hidden layers with 10, 5 and 3 neurons respectively. We tried various architectures so that the training converges in a reasonably good timeframe. The input dataset was split with 70:30 for training and test dataset.

churn ~ rev_Mean + totmrc_Mean + ovrmou_Mean + roam_Mean + drop_vce_Mean + threeway_Mean + iwylis_vce_Mean + months + uniqsubs + totcalls + hnd_price + phones + lor + eqpdays + fe_mean_per_minute_charge + fe_tot_revenue_per_call + fe_tot_mou_per_call + fe_tot_revenue_adj + asl_flag_N + crclscod_BA + crclscod_EA + crclscod_ZA + ethnic_O + kid16_17_U + prizm_social_one_Missing + prizm_social_one_R + prizm_social_one_T

LIGHT GBM

As it is one of the best and most recent Boosting technique developed and often considers homogenous weak learners, it also learns them sequentially in a very adaptive way i.e. Base model learns from the previous model's weak learners and improvises it and combines them using a deterministic strategy.

```
#Light GBM

#Complete Model

lgb_clf = LGBMClassifier(
        nthread=4,
        boosting_type='dart',
        dart_subsample= 0.79,
        n_estimators=2000,
        learning_rate=0.015,
        num_leaves=43,
        max_depth=18,
        reg_alpha=0.08,
        reg_lambda=0.36,
        is_unbalance=False,
        silent=-1,
        verbose=-1,
        )


lgb_clf.fit(X_train, y_train,eval_metric= 'auc', verbose= 50, early_stopping_rounds= 100)
```

Parameters

boosting_type: dart, Dropouts meet Multiple Additive Regression Trees.

dart_subsample: Subsample ratio of the training instance

num_leaves: Maximum tree leaves for base learners.

max_dept: Maximum tree depth for base learners,

learning_rate: Boosting learning rate.

n_estimators: Number of boosted trees to fit.

is_unbalance: False, balanced, mode uses the values of y to automatically adjust weights inversely proportional to class frequencies in the input data.

reg_alpha: L1 regularization term on weights

reg_lambda: L2 regularization term on weights

Machine Learning Model Evaluation

We have focused Performance metrics used for Classification problems like AUC-ROC and Accuracy for our ML Models.

ROC-AUC Curve

The AUC curve is plotted with TPR against the FPR where TPR is on y-axis and FPR is on the x-axis.

Accuracy

Accuracy in classification problems is the number of correct predictions made by the model over all kinds of predictions made.

$$\text{Accuracy} = \frac{TP+TN}{TP+FP+FN+TN}$$

Precision

Precision is a measure that tells us what proportion of customers that we predicted as churn, has actually had churn.

$$\text{Precision} = \frac{TP}{TP+FP}$$

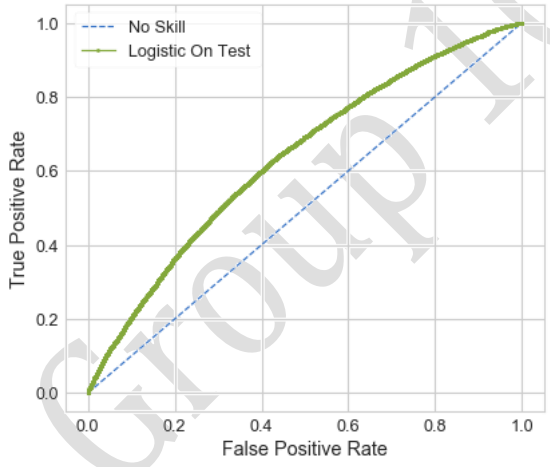
Recall

Recall is a measure that tells us what proportion of customers that actually had churn and how much the algorithm is predicted as churn.

$$\text{Recall} = \frac{TP}{TP+FN}$$

SIMPLE LOGISTIC REGESSION

```
telco_test = pd.DataFrame()
telco_test['Customer_ID']=id_test['Customer_ID']
telco_test['prob']=logreg.predict_proba(X_test)[:,1]
auc_curve(telco_test,y_test,'On Test ','Logistic ')
```

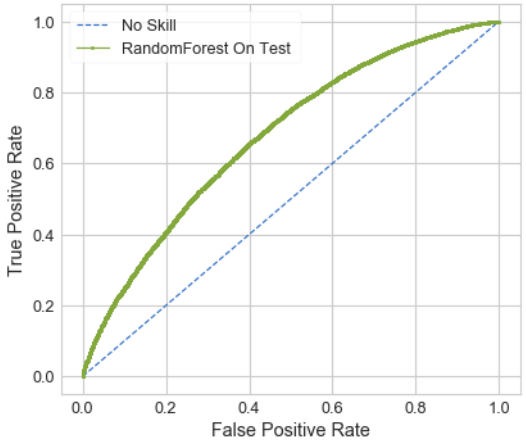



```
On Test Logistic ROC-AUC score is:  63.56862072045395
```

Logistic Regression		Actual Values	
		Positive (1)	Negative (0)
Predicted Vaues	Positive (1)	9092	6059
	Negative (0)	8893	5956

RANDOM FOREST

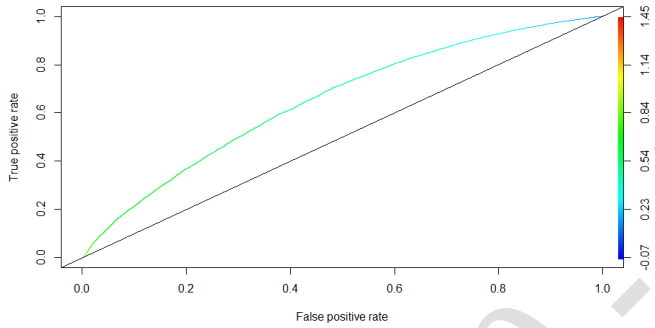
```
telco_test = pd.DataFrame()
telco_test['Customer_ID']=id_test['Customer_ID']
telco_test['prob']=rf_clf.predict_proba(X_test)[:,1]
auc_curve(telco_test,y_test,'On Test ','RandomForest ')
```



On Test RandomForest ROC-AUC score is: 67.82404232780186

Random Forest		Actual Values	
		Positive (1)	Negative (0)
Predicted Vaues	Positive (1)	9870	6287
	Negative (0)	8911	4932

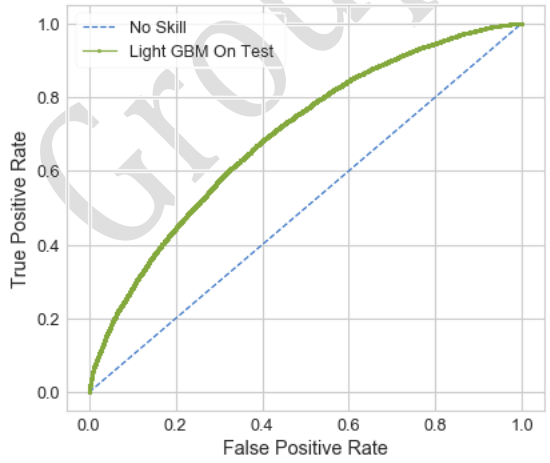
NEURAL NETWORK



Neural Network		Actual Values	
		Positive (1)	Negative (0)
Predicted Vaues	Positive (1)	9096	6126
	Negative (0)	5679	9099

LIGHT GBM

```
telco_test = pd.DataFrame()
telco_test['Customer_ID']=id_test['Customer_ID']
telco_test['prob']=lgb_clf.predict_proba(X_test,num_iteration=lgb_clf.best_iteration_)[:,1]
auc_curve(telco_test,y_test,'On Test ','Light GBM ')
```



On Test Light GBM ROC-AUC score is: 69.61633636660903

Light GBM		Actual Values	
		Positive (1)	Negative (0)
Predicted Vaues	Positive (1)	9790	5752
	Negative (0)	9340	5118

COMPARATIVE ANALYSIS OF EXPIREMENTED ML TECHNIQUES

Evaluation of ML Models				
ML Technique	Logistic Regression	Random Forest	Neural Network	Light GBM
AUC-ROC Score	63.57	67.82	65.10	69.62
Prediction Accuracy	60%	63%	61%	64%
Precision for Churn	60%	61%	60%	63%
Recall for Churn	60%	67%	62%	66%

Observations

- Logistic Regression and Neural Network have slightly limited success in predicting the Churn of VeriTel Customers, if compared to Random Forest and Light GBM
- All the ML models are able to provide an average level of accuracy in determining if the Customers will churn or will be retained. Particularly, Random Forest and Light GBM have predicted the churn slightly more consistently and accurately as compared to other models.
- Random Forest is the best among these for predicting customers that will churn
- From the test results, it was observed that overall performance of Light GBM ML model is best for predicting Churn because of its boosting technique

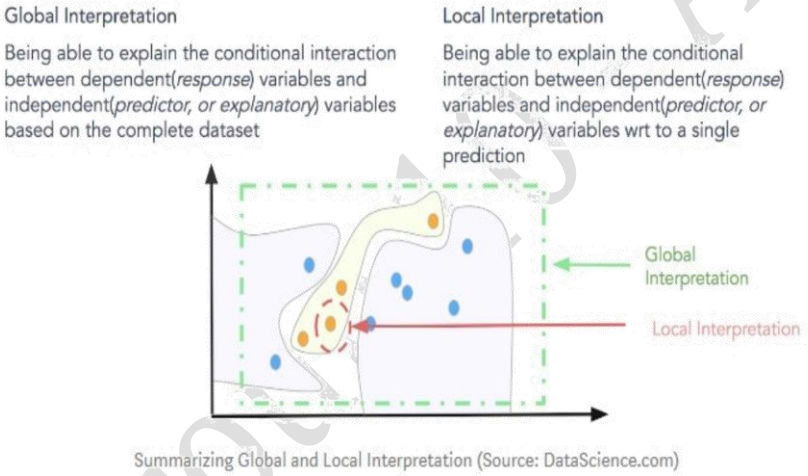
Explainable Machine Learning

Now we know who all the customers that are going to churn, the next question that the end user looks for what is reason behind this high-risk customer? Why they are going to churn? Over the period of time this will be important to know the root cause for the churn.

In machine learning complex model has big issue with transparency, we don't have any strong prove why model give that prediction and which feature are impacting the model prediction, which features are strongly contributing, and which are negative contribution for model prediction. By feature importance graph we can see which features importance by passing complete training and test dataset, but for single row of features or for any given instance it is very difficult to understand why and how model predict output. And, we will be answering to this question through the SHAP (Shapley Additive explanations).

SHAP

SHAP goal is to explain the prediction of a given instance X by computing the contribution of each feature to the prediction. The feature values of a data instance act as players in a coalitional game theory. SHAP prediction output is a fair distribution of all the feature Shapley values. Shapely value is actually distribution, it's an average of model contribution made by each player (features) over all permutation of player (features). The baseline for Shapley values is the average of all predictions. In the plot, each Shapley value is an arrow that pushes to increase (positive value) or decrease (negative value) the prediction.



Summarizing Global and Local Interpretation (Source: DataScience.com)

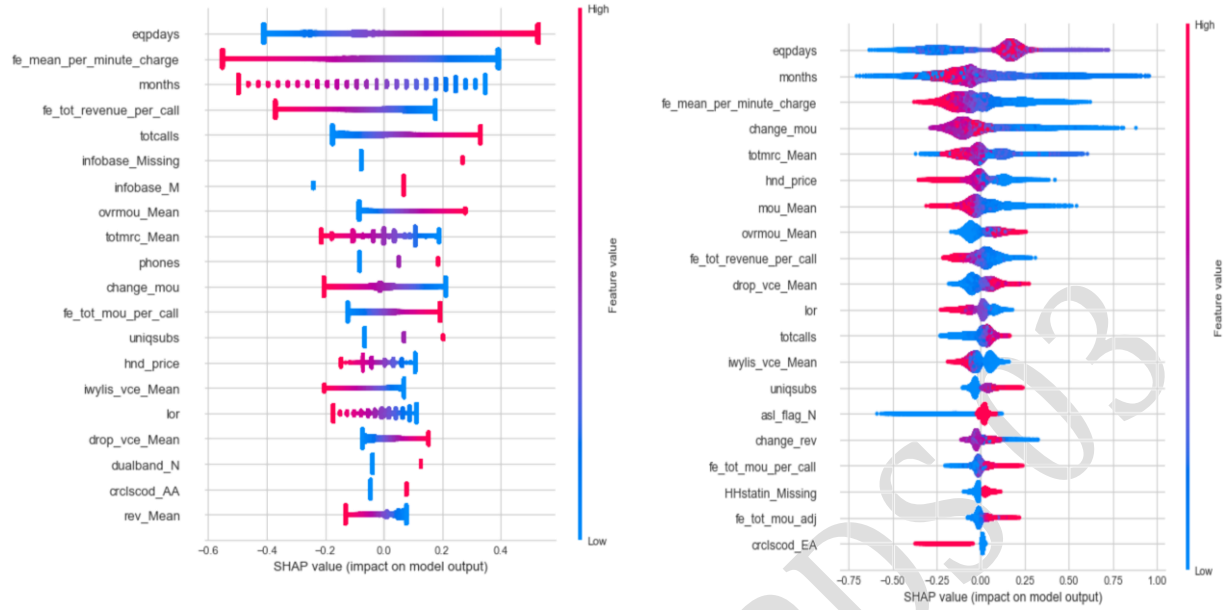
GLOBAL VS LOCAL INTERPRETATION

Here we will understand the model at both Global and Local level. At Global level we are explaining the interaction of our churn variable with Independent variables on the complete dataset. At local level we are explaining the interaction of churn variable on each single prediction.

Global Interpretation

At a global level, the below graphs are summarizing the effects of all the explanatory variables on the model output, colour coded to show the direction of the impact (red means an increase, while blue shows a decrease), with SHAP value more far away from zero meaning a bigger impact. It is also visually easy to see which variables have the strongest relationship with the target variable. In this way, SHAP can also be used as a tool for variable selection.

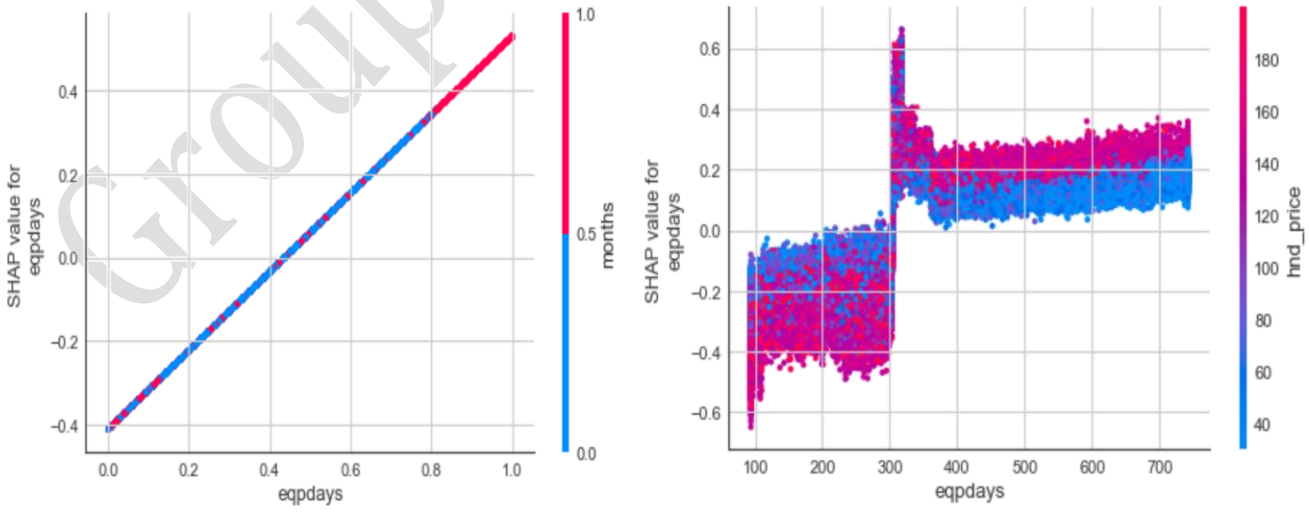
Logistic Regression **Light GBM**



In both models, the eqpdays (Number of days (age) of current equipment) variable has the most predictive power. The order of importance, further on, is slightly different, given that the regression is constrained in fitting the relationship to a linear one, while the Light GBM can use non-linear components to describe it. This is also apparent in the single variable graphs, which, in addition to showing the positive/negative relationships to the target, are also showing the form of the relationship.

The plots below represent the change in propensity score of churn as the most predictive variable (eqpdays) changes. Vertical dispersion at a single value of eqpdays (more visible on the Light GBM plot) represents interaction effects with other features. To help reveal these interactions, the plot automatically selects another feature for coloring (see right vertical axis).

Logistic Regression **Light GBM**



The Light GBM model reveals a more complex relationship being captured, and it better explains interactions between variables.

Local Interpretation

Another great aspect of SHAP is that it determines a separate set of values for each observation in the dataset. This feature can have multiple usages:

• It allows to explain why model output takes given value for each observation (in case of case, each risk/non risk customers can be explained).

• It can determine the observations where a certain variable or a set of variables are more/less predictive, and thus it aids in segmentation.

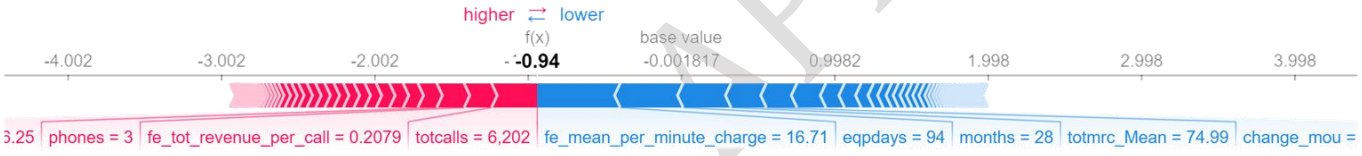
• It can help in optimizing the model by removing outliers (observations where SHAP values are low for a big number/ all variables)

• It can help in explaining interdependencies between variables at a local/ segment level

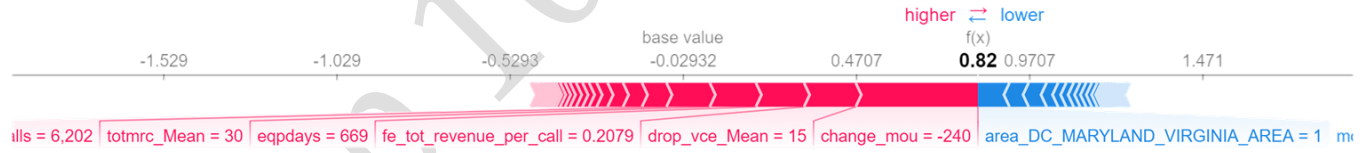
• It can help with model exclusions, as missing features have no attributed impact to the model parameters.

For a local view which makes it clearer which way each variable is 'pushing' the model output towards, the following plots can be used, selecting the row/observation we want to show:

Logistic Regression



Light GBM



The above graphs are showing the base value (The average model output over the training dataset) compared to the model output. Shown in red are the variables pushing the prediction higher, while the opposite holds true for the variables in blue.

Conclusion

We started with the CEO's problem statement of who are the customers that are going to churn out and why they will churn.

Outcomes

Who ? - To arrive at the suitable ML model that can reliably predict the probability of Customers that will churn out, we found that Light GBM ML model is more suitable given the limited dataset that we have.

It fulfills our objective of identifying those customers that will churn with reasonably good accuracy and can be used by the VeriTel CEO for strategizing targeted marketing interventions and retain those identified customers.

Why ? – To arrive at the business reasons which can explain the reasons for why the Customer will churn.

The CEO can identify actionable Reasons that can be prioritized for Customer churn at global level.

The Sales/Marketing department can identify actionable reasons that are impacting individual customers for more tailored interventions.

The business can identify non-actionable reasons and not allocate additional efforts or resources thus steering problem solving that is meaningful.

SHAP Value and reason selection			
#	Feature / reason	Interpretation	Possible action
1	**eqp_days** | Age of Equipment since it has been part of VeriTel network	Customers with Higher values of this feature have higher propensity to churn i.e. the customers who have been customers of Veritel from the same equipment have tendency to cancel their subscription or move to a new service provider after the contract ends	VeriTel should consider assisting long-time customers to move to new equipment when connectivity contracts are renewed by offering exchange policy for old equipment
2	**Mean Talktime per unit Charge** | (Mean monthly minutes of Use / Monthly Charge)	Customers with Lower values of this feature have higher propensity to churn i.e. the customers who spend less time talking and have higher monthly bills (eg: Customers of ISD Plans) or those who are getting less talktime/$ have tendency to cancel their subscription or move to a new service provider after the contract ends	VeriTel should consider offering competitive ISD Plans or offer more talktime/$ to those customers that are long-time subscribers
3	**hnd_price** | mean Price of Handset	Customers with Lower values of this feature have higher propensity to churn i.e. the customers who use low end equipment (< $400) have a tendency to to cancel their subscription or move to a new service provider after the contract	VeriTel should consider exploring if competitor service providers are offering higher value equipment (>400) and if those low-end equipment users that have been loyal customers of VeriTel can be offered new/high-end equipment at reduced pricing/contract at time of renewal

Future Recommendations

- The CEO of VeriTel can task his Business experts to consolidate a multi-year data set to arrive at how soon a customer will churn
- Data features pertaining to Connectivity at the Customer's region can be used to segment the customer and suggest possible reasons for Churn
- The current dataset only has values for call details for Customer support but data about service requests can be included.
- If data related to response of a customers to marketing campaigns (email, sms etc.) can be included, customer segmentation can be explored that will provide more insights to the CEO/ business on certain actions

Appendices

APPENDIX 1 – GITHUB PROJECT REPOSITORY

https://github.com/SatishChilloji/iim_apds_churn_prjct Access the files/folders below to refer files		
Data Source File	iim_apds_churn_prjct/data/Telecom_customer_churn.csv	The data source file accessed from Kaggle
Data Dictionary	iim_apds_churn_prjct/data/telecom_customer_churn_data_dictionary.xlt	Data Dictionary for the Variables in the data source file
Numerical Variable Missing Values	iim_apds_churn_prjct/data/numerical_var_missing_values.xlsx	Analysis of Numerical Variables with missing values
Categorical Variable Missing Values	iim_apds_churn_prjct/data/categorical_var_missing_values.xlsx	Analysis of Categorical Variables with missing values
Data Visualization	churn_eda_1.ipynb churn_eda_2.ipynb	Python Jupyter Notebook for Data Visualization
Data Preparation	data_prep.ipynb	Python Jupyter Notebook for Data Preparation
Logistic Regression and Light GBM with SHAP + Random Forest	model_build.ipynb	Python Jupyter Notebook for Logistic Regression and Light GBM Codes with SHAP + Random Forest Code
Neural Network	iim_apds_churn_prjct/src/Chinmay/Customer_Churn_V0.2.R	R Code for Neural Network with ROC Visualization

APPENDIX 2 – DERIVED FEATURES LIST

Mean Talktime per unit Charge: The new feature represents the mean of charge for the calls per minute. This value has been derived based on numerical features: 'Mean number of monthly minutes of use' and 'Mean monthly revenue (charge)' and has been derived as, *Mean per Minute Charge = Mean number of monthly minutes of use/ Mean monthly revenue (charge)*

Mean Extra Amount that Customer Pays: The new feature represents the average extra amount customer pays per month. This field is derived as follows: *Mean Extra Amount that Customer Pays = Mean monthly revenue (charge) - Mean total monthly recurring charge*

This feature provides an insight of how much extra charge a customer is paying and hence eventually can be a determinant for a customer churn.

Mean Allocated Calls: This feature signifies total minutes of calls per month and is derived from two other features: 'Mean number of monthly minutes of use' and 'Mean overage minutes of use'. The field is derived as follows: *Mean Allocated Calls = Mean number of monthly minutes of use + Mean overage minutes of use*

Mean Monthly Profit: This feature signifies total monthly profit of the telecom company and may impact the customer churn. The field is derived as: *Mean Monthly Profit = Mean overage revenue + Mean revenue of voice overage + Mean revenue of data overage*

Mean Failed/Dropped Calls: This feature signifies total monthly profit of the telecom company and may impact the customer churn. The field is derived as: *Mean Failed/Dropped Calls = Mean number of dropped (failed) voice calls + Mean number of dropped (failed) data calls + Mean number of blocked (failed) voice calls + Mean number of blocked (failed) data calls*

This field is an important indicator of overall service quality.

Percentage of successful calls: This feature is an important indicator of the service quality. The field is derived as: *Percentage of successful calls = Mean number of completed voice calls / Mean number of attempted voice calls placed*

Unrounded to rounded completed voice call difference: This feature is an important indicator of the service quality. The field is derived as: *Unrounded to rounded completed voice call difference = Mean unrounded minutes of use of completed voice calls- Mean number of completed voice calls*

Unrounded to rounded completed received voice call difference: This feature is also an important indicator of the service quality. The field is derived as: *Unrounded to rounded completed received voice call difference = Mean unrounded minutes of use of received voice calls - Mean number of received voice calls*

Mean total revenue per call: This feature signifies mean charge per call. The feature is derived as: *Mean total revenue per call = Total revenue/Total number of calls over the life of the customer*

Mean Total minutes of use per call: This feature signifies mean total minutes per call and is an important feature identifying usage. The feature is derived as: *Mean total revenue per call = Total minutes of use over the life of the customer/Total number of calls over the life of the customer*

Mean total charge per call: This field is derived out of two other derived features using the following formula: *Mean total charge per call = Mean total revenue per call * Mean Total minutes of use per call*

Total Revenue Adjustment: This field is derived as: *Total Revenue Adjustment = Total Revenue - Billing adjusted total revenue over the life of the customer*

Total minutes of use adjusted: This field is derived as: *Total minutes of use adjusted = Total minutes of use over the life of the customer - Billing adjusted total minutes of use over the life of the customer*

Total calls adjusted: This field is derived as: *Total calls adjusted = Total calls - Billing adjusted total number of calls over the life of the customer*

Average revenue per call: This field is derived as: *Average Revenue per call = Average monthly revenue over the life of the customer/Average monthly number of calls over the life of the customer*

This field is an important indicator of the average charge the customer is incurring.

Average minute of use per call: This field is derived as: *Average minute of use per call = Average monthly minutes of use over the life of the customer/Average monthly number of calls over the life of the customer*

This field is an important indicator with respect the usage.

Average charge per call: This field is derived out of two derived features using the following formula: *Average charge per call = Average revenue per call * Average minute of use per call*

APPENDIX 2 – REFERENCES

1	P. Kisioglu and I. Y. Topcu, "Applying Bayesian belief network approach to customer churn analysis: a case study on the telecom industry of Turkey." Expert Systems with Applications 38, 2010, pp. 7151-7157
2	B. Huang, M. T. Kechadi, and B. Buckley," Customer churn prediction in telecommunications." Expert Systems with Applications, 39(1), 2012, pp. 1414-1425
3	Y. Zhao, B. Li, and X. Li, "Customer churn prediction using improved one-class support vector machine." Lecture Notes in Artificial Intelligence, 3584, 2005, pp. 300—306
4	J. J. Rodrigez, L. l. Kuncheva, and J. A. Carlos, "Rotation Forest; A New Classifier Ensemble Method." IEEE Transactions on Pattern Analysis and Machine Intelligence, 28(10), 2006, pp. 1619–1630
5	C. F. Tsai and Y. H. Lu, "Customer churn prediction by hybrid neural networks." Expert Systems Application, 36(10), 2009, pp. 12547—12553
6	W. Verbeke, D. Martens, C. Mues, and B. Baesens, "Building comprehensible customer churn prediction models with advanced rule induction techniques." Expert Systems with Applications, 38, 2011, pp. 2354—2364
7	K. W. Bock and D. V. Poel, "An empirical evaluation of rotation-based ensemble classifiers for customer churn prediction." Expert Systems with Applications, 38(10), 2011 pp. 12293-–12301
8	V. Yeshwanth, V. V. Raj, and M. Saravanam, "Evolutionary churn prediction in mobile networks using hybrid learning", Proc. of XXIV Florida Artificial Intelligence Research Society Conference, 2011, pp. 471– 476
9	A. Ghorbani, F. Taghiyareh, and C. Lucas, "The application of the locally linear model tree on customer churn prediction." Proceedings of the International Conference of Soft Computing and Pattern Recognition (SOCPAR'09), Malaysia, 2009, pp. 472—477
10	D. Larose,(2005). Discovering knowledge in data:An introduction to data mining. New Jersey, USA: Wiley
11	https://journalofbigdata.springeropen.com/articles/10.1186/s40537-019-0191-6
12	**Matplotlib:** https://matplotlib.org/users/whats_new.html#figure-and-axes-creation-management
13	**Seaborn:** https://seaborn.pydata.org/tutorial/distributions.html
14	**Simple Imputer:** https://scikit-learn.org/stable/modules/generated/sklearn.impute.SimpleImputer.html
15	**Iterative Imputer:** https://scikit-learn.org/stable/modules/generated/sklearn.impute.IterativeImputer.html
16	**Feature Importance:** https://scikit-learn.org/stable/auto_examples/ensemble/plot_forest_importances.html
17	**Seaborn:** https://seaborn.pydata.org/tutorial/distributions.html
18	Edwin de Jonge, Mark van der Loo: An introduction to data cleaning with R, Statistics Netherlands
19	https://www.kdnuggets.com/2018/05/packt-tackle-common-data-cleaning-issues-r.html
20	https://blog.dominodatalab.com/manual-feature-engineering/
21	https://en.wikipedia.org/wiki/Feature_engineering
22	https://en.wikipedia.org/wiki/Data_transformation
23	https://scikit-learn.org/stable/modules/generated/sklearn.impute.SimpleImputer.html
24	https://cran.r-project.org/web/packages/neuralnet/neuralnet.pdf
25	https://scikit-learn.org/stable/modules/generated/sklearn.linear_model.LogisticRegression.html
26	https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.RandomForestClassifier.html
27	https://www.onclick360.com/interpretable-machine-learning-with-lime-eli5-shap-interpret-ml/
28	https://parker-fitzgerald.com/wp-content/uploads/2019/12/ML-Interpretability-SHAP-example.pdf