Statistical Data Mining
(EAS 506)

Project Report

ANALYSIS AND MODELING OF CUSTOMER CHURN IN TELECOM INDUSTRY

By:

Madhavan Rangarajan(UBID : 50442679,mrangara@buffalo.edu)

K. Bindu Harshita (UBID : 50479393,binduhar@buffalo.edu)

Bysani Sai Ramya Sree (UBID : 50455444,sbysani2@buffalo.edu)

Sreeja Manchikanti (UBID : 50442935,sreejama@buffalo.edu)

Nikhitha Mekala (UBID : 50461360, nmekala@buffalo.edu)

Git Integration: https://github.com/madhavanr29/Project_Group_3_telecom

ABSTRACT

Customers in the telecom industry can actively switch between operators and have access to a wide range of service providers. In this intensely competitive sector, the telecoms industry experiences a high churn rate. Due to the fact that acquiring new customers is 5 to10 times more expensive than maintaining existing ones, client retention has now exceeded customer acquisition in importance.For many established operators, keeping highly profitable customers is their primary business goal.To lower customer turnover, telecom companies must identify the customers who are most likely to depart. The main objective is to look at customer-level data from a leading telecom business and prediction models to identify consumers most likely to depart(customers who are more likely to leave the service), and analyzing the results to make decisions accordingly. Companies are seeking statistical analysis and machine learning solutions to reduce churn. The main focus of this study is on analysis of the data for a telecom chain through Exploratory Data Analysis, Feature Extraction and Predictive Modelling. Through the process the factors that impact churn and also understand the pattern of customers who are at high risk of churn are known.

MOTIVATION

The two most popular invoicing types in the telecom sector are pre-paid(customers pay/recharge with a certain amount in advance and then use the services) and post-paid(customers pay a monthly/annual bill after using the services). Pre-paid clients discontinue using the service abruptly,they don't have to give any notice if they are leaving, in contrast to post-paid users who must inform the service provider before switching. 90% of buyers in the South Asian market select the prepaid option. Therefore, it is crucial to comprehend how prepaid clients behave,that is, trends of the customers over a period of time ,in order to make further business decisions and change the existing decisions . If we can predict the behavior of prepaid clients , churn prediction will take on new directions, and telecom companies ability to make commercial decisions will be enhanced.

DATA UNDERSTANDING

Using the data (features) from the first three months, the business goal is to estimate the churn in the most recent month. Understanding the normal consumer behavior during churn will help with this process.Using the data gathered over a 4-month period, customer churn is estimated. The first two months emphasize the favorable period of consumer behaviorThe next two months can be divided into two periods: the action phase and the churn phase. There are 226 features in the dataset. There are 99999 entries in the dataset.

The source of the dataset can be found at
https://www.kaggle.com/code/gauravduttakiit/telecom-customer-churn-eda/data

	Acronyms	Descriptions
0	MOBILE_NUMBER	Customer phone number
1	CIRCLE_ID	Telecom circle area to which the customer belongs to
2	LOC	Local calls - within same telecom circle
3	STD	STD calls - outside the calling circle
4	IC	Incoming calls
5	OG	Outgoing calls
6	T2T	Operator T to T, i.e. within same operator (mobile to mobile)
7	T2M	Operator T to other operator mobile
8	T2O	Operator T to other operator fixed line
9	T2F	Operator T to fixed lines of T
10	T2C	Operator T to it's own call center
11	ARPU	Average revenue per user
12	MOU	Minutes of usage - voice calls
13	AON	Age on network - number of days the customer is using the operator T network
14	ONNET	All kind of calls within the same operator network
15	OFFNET	All kind of calls outside the operator T network
16	ROAM	Indicates that customer is in roaming zone during the call
17	SPL	Special calls
18	ISD	ISD calls
19	RECH	Recharge
20	NUM	Number
21	AMT	Amount in local currency
22	MAX	Maximum
23	DATA	Mobile internet
24	3G	3G network
25	AV	Average
26	VOL	Mobile internet usage volume (in MB)
27	2G	2G network
28	PCK	Prepaid service schemes called - PACKS
29	NIGHT	Scheme to use during specific night hours only
30	MONTHLY	Service schemes with validity equivalent to a month
31	SACHET	Service schemes with validity smaller than a month
32	*.6	KPI for the month of June
33	*.7	KPI for the month of July
34	*.8	KPI for the month of August
35	*.9	KPI for the month of September
36	FB_USER	Service scheme to avail services of Facebook and similar social networking sites
37	VBC	Volume based cost - when no specific scheme is not purchased and paid as per usage

LIFE CYCLE OF CUSTOMER BEHAVIOR

Customers typically opt to switch to another competition over time rather than in an instant (this is especially applicable to high-value customers). In churn prediction, we presume that the client lifetime comprises three stages:

a) Clean/Good Phase:

The service makes the customer happy and will stay with the provider.

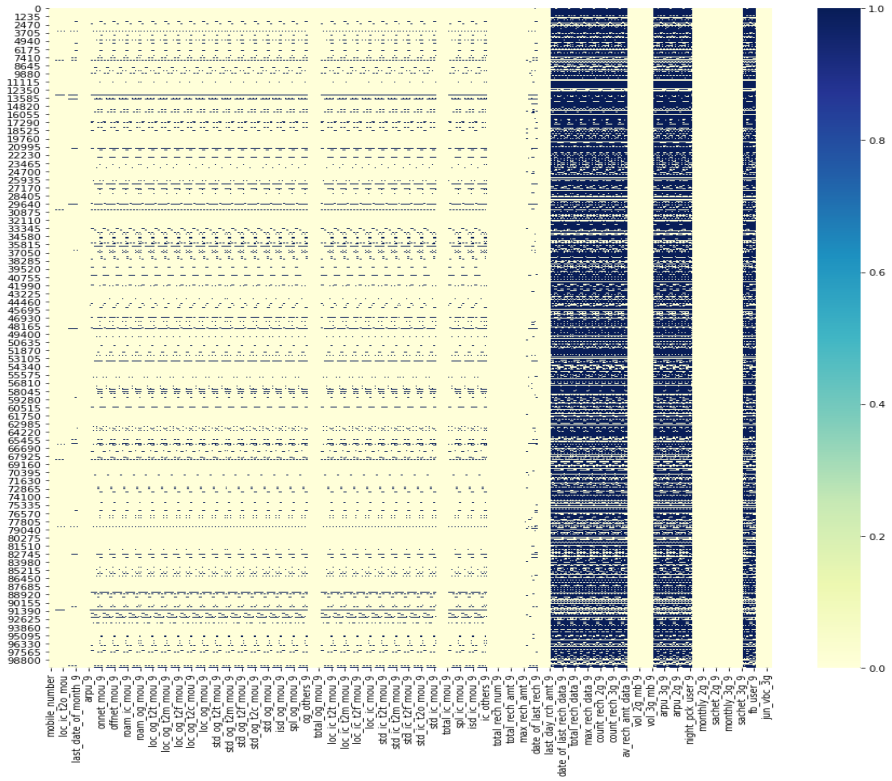
b) Action Phase:

In this stage, the customer's experience starts to suffer; for instance, he or she receives an alluring offer from a rival, pays unfair fees, is dissatisfied with the level of service, etc. The client typically behaves differently during this time than during the "good" months. Additionally, as some remedial actions (such matching the competitor's offer/improving the service quality etc.) can be implemented at this stage, it is vital to identify high-churn-risk clients during this period.

c) Churn Phase:

The consumer is considered to have churned during this time. Based on this stage, we define churn. It's also vital to keep in mind that you cannot foresee using this data at the moment (i.e., the action months). You then discard all the data associated with this phase after marking churn as 1/0 depending on this phase.

DATA PRE-PROCESSING



The above figure represents the missing values in the whole dataset. The blue parts of the above heatmap are the areas where we have null values. Also, visually we can see that there are many rows where the same features are missing (indicated by blue stripes).

a) There were multiple features with more than 50% values missing. However, imputing these features with Mean/ Median may introduce bias. Thus, the best way to handle them is to remove them. So, we have chosen to remove features with more than 50% missing values.

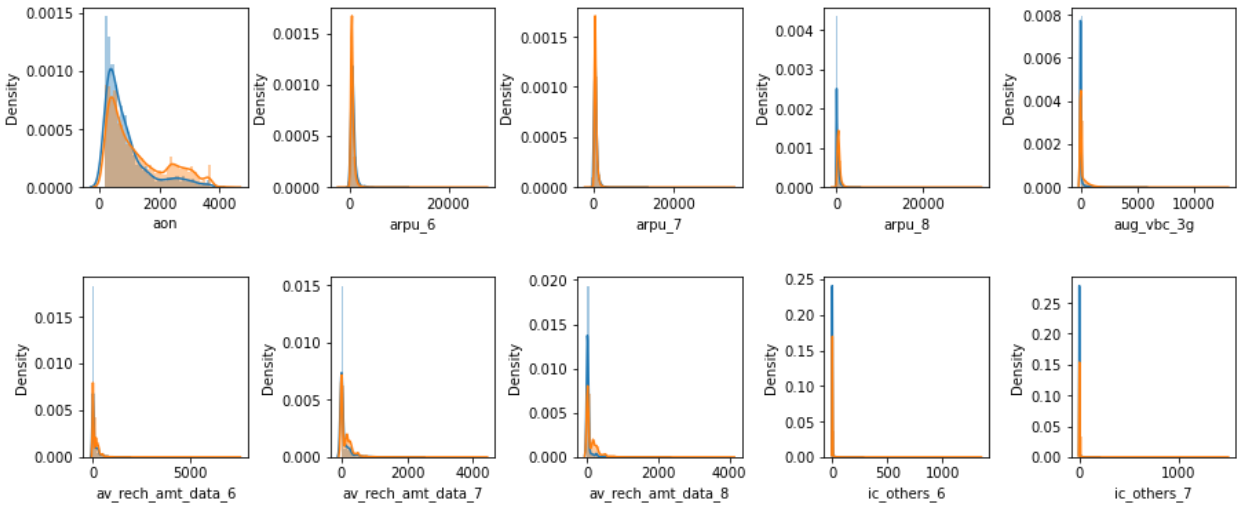
	arpu_2g_6	arpu_2g_7	arpu_2g_8	arpu_2g_9	arpu_3g_6	arpu_3g_7	arpu_3g_8	arpu_3g_9	av_rech_amt_data_6	av_rech_amt_data
count	25153.000000	25571.000000	26339.000000	25922.000000	25153.000000	25571.000000	26339.000000	25922.000000	25153.000000	25571.0000
mean	86.398003	85.914450	86.599478	93.712026	89.555057	89.384120	91.173849	100.264116	192.600982	200.9812
std	172.767523	176.379871	168.247852	171.384224	193.124653	195.893924	188.180936	216.291992	192.646318	196.7912
min	-35.830000	-15.480000	-55.830000	-45.740000	-30.820000	-26.040000	-24.490000	-71.090000	1.000000	0.5000
25%	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	82.000000	92.0000
50%	10.830000	8.810000	9.270000	14.800000	0.480000	0.420000	0.880000	2.605000	154.000000	154.0000
75%	122.070000	122.070000	122.070000	140.010000	122.070000	119.560000	122.070000	140.010000	252.000000	252.0000
max	6433.760000	4809.360000	3483.170000	3467.170000	6362.280000	4980.900000	3716.900000	13884.310000	7546.000000	4365.0000

8 rows × 36 columns

b) From the above stats if we look carefully we have columns with negative minimum values such as 'average revenue per user 2g and 3g' where missing values may not indicate zero. But if we look at some other columns we see that the columns are in absolute scale but their minimum values are not zero. This indicates the null values present in those columns represent 'no activity' . Thus, we have imputed the missing values in those columns with zero.

c) Once our data was pre-processed significantly, there were still some variables such as ‘days_last_rech_month_8’ and ‘spl_og_mou_6’ where less than 2% values were missing. For such variables we have used “KNN Imputer” as part of the “sklearn.impute” package to impute these missing values based on the nearest neighbor. The reason to do so is that Nearest Neighbour based imputation techniques have been shown to work efficiently compared to heuristic imputation techniques such as mean / mode imputation.

EXPLORATORY DATA ANALYSIS



The above plot shows the distribution between churners and non-churners for various numerical variables. We can see that the data is not interpretable because we have a lot of outliers in the top tail for most of the columns. Even transformation of variables in log scale will not solve the problem if the data is skewed to such an extent. We have realized that first we have to handle outliers as algorithms like Logistic Regression and SVM are not robust to outliers.

```
def percent_out(x):

    q1 = x.quantile(0.25)
    q3 = x.quantile(0.75)

    iqr = q3 - q1

    lower_bound = q1 -(1.5 * iqr)
    upper_bound = q3 +(1.5 * iqr)

    len_filter = len(x[x.between(lower_bound, upper_bound, inclusive=True)])
    percent_gone = 100 - ( (100 *len_filter) /len(x))
    return(round(percent_gone,2))
```

The above function is used to check the percentage of data we lose if we remove outliers using the interquartile range method. Below, we can see the amount of data we lose in each variable if we choose to remove outliers using the IQR method.

column_name	%_data_loss
aon	0.09
og_others_7	0.52
og_others_8	0.61
max_rech_data_8	1.64
max_rech_data_6	1.68
max_rech_data_7	1.69
last_day_rch_amt_8	3.78
last_day_rch_amt_7	4.46
total_rech_amt_8	4.48
total_og_mou_7	4.62
last_day_rch_amt_6	4.67
total_og_mou_6	4.88
arpu_8	4.89
total_rech_amt_6	4.96
av_rech_amt_data_6	4.98

If we choose to remove data points from the columns we will lose extensive data. In case if we use it to build our linear model it will affect the predictions. Thus, we have chosen to drop all columns which have high outliers (% data loss > 12). Rest of the features we have capped the data using IQR method.

MODEL PREPARATION

Dummy Variable Creation: All the categorical variables including some time based derived features were preprocessed using dummy encoding to make it adaptable to a machine readable format.

Stratified Train and Test Split: After the preprocessing of the data was done, a train test split was performed by stratifying on the target variable with an intention to maintain equal ratio of positive and negative samples in both training and testing set. The split between training and test set had a ratio of 75:25.

Data Normalization: A MinMaxScaler was used on the training data to fit the training data in the range 0 to 1. Although algorithms like DecisionTree do not require data to be on the same scale, we have also used linear models which have scaling assumptions. The scaler was also used to transform the range of the variables of the test data.

GOALS OF STUDY

There are 2 goals of this case study:

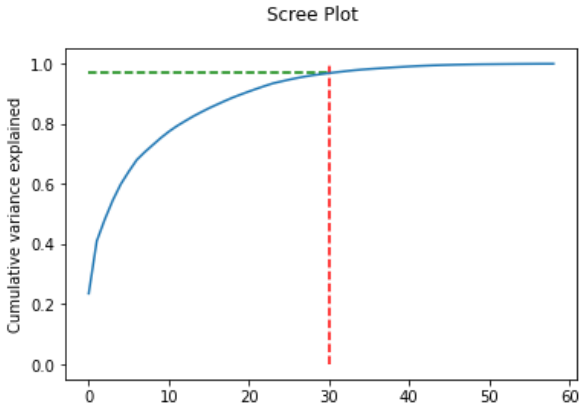
a) To build a Predictive model which will predict if a high value customer will churn or not in the future

b) Build an Interpretable model to detect the variables which are strong predictors of churn

So we need to build a predictive model with good metrics(recall being the major criteria) and an interpretable model which will be useful in analyzing strong influencers of churn.

PCA FOR DIMENSIONALITY REDUCTION

As there are a lot of features , executing ML algorithms and tuning hyperparameters on top of it is not a good idea. It may become computationally expensive. Lets use PCA as a dimensionality reduction technique (**on continuous features**) to reduce the feature space. As there are few derived dummy features (0 or 1) let's choose to add those features back **after executing PCA on continuous features.**



30 principal components capture up to 97% variation in the data. So let's use IncrementalPCA to fit and transform on the first 30 principal components. We will use these 30 principal components to build **predictive models.**

PREDICTIVE MODELS

Some common aspects in all the predictive models built:

1) **Stratified K-Fold Cross Validation:**

   Cross Validation using Stratified K-Fold can be utilized for Hyperparameter tuning by maintaining target proportion even during the Cross Validation Process.

2) **Hyperparameter Tuning:**

   We have used Random Forest, XGBoost , etc as part of this study which has multiple parameters and it is computationally expensive to tune all of these parameters together. Thus, we have chosen to tune parameters iteratively at some instances. The hyperparameter tuning was done using Grid Search CV.

3) **Evaluation Metrics:**

We have used Accuracy, Sensitivity, F1-Score and ROC-AUC as evaluation metrics to determine the performance of the model. However, Sensitivity (Recall) will be the major metric based on our project.

Accuracy:

ML model accuracy is the measurement used to determine which model is best at identifying relationships and patterns between variables in a dataset based on the input, or training, data.

Sensitivity:

The sensitivity is calculated as the ratio between the number of Positive samples correctly classified as Positive to the total number of Positive samples. The sensitivity measures the model's ability to detect positive samples. The higher the recall, the more positive samples detected.

F1 Score:

The F1 score is the harmonic mean of precision and recall. It is a very effective measure in case of an imbalanced classification problem.

AUC:

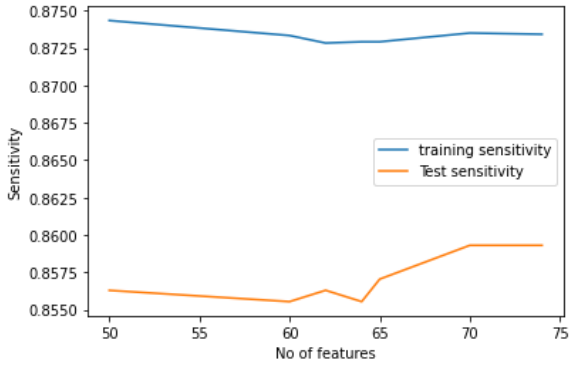
AUC is the primary metric in this study as this data source is from a competition where the primary evaluation criteria are auc_roc. Also, AUC is the measure of the ability of a classifier to distinguish between classes and is used as a summary of the ROC curve.

a) **Logistic Regression with Recursive Feature Elimination (RFE) :**

   Logistic Regression is a simple linear classification algorithm that works when the trend of the data is linear.

   Tuned Parameters: “n_features_to_select”

   Recursive Feature Selection with Grid Search CV is used to pick the number of features to select.



Note: In the above figure Test sensitivity means Cross Validation

70 Features seemed optimum and thus a Logistic Regression model was built using these 70 features.

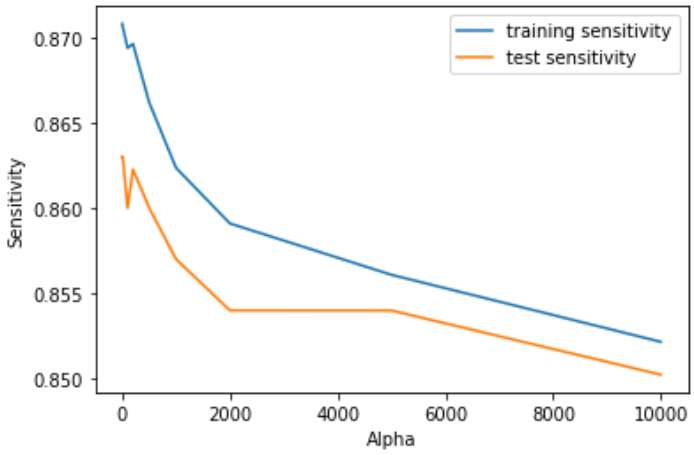
```
rfe1 = RFE(lr,70)
rfe1.fit(X_train_pca,y_train)
```

b) Ridge Classifier:

Ridge Classifier is a technique that is used to penalize the model coefficients to prevent overfitting. Especially in high dimensionality cases and at times where we are using a lot of redundant variables to build our model we can utilize Ridge Classifier.

Tuned Parameters: "alpha"

Alpha controls that regularization rate. If alpha is large the model is regularized less and if alpha is small the model regularizes more.



Note: In the above figure Test sensitivity means Cross Validation

c) Decision Tree:

A decision tree is a simple rule based algorithm that is optimized through cost by splitting each node based on the best information gain it receives in order by performing the split.

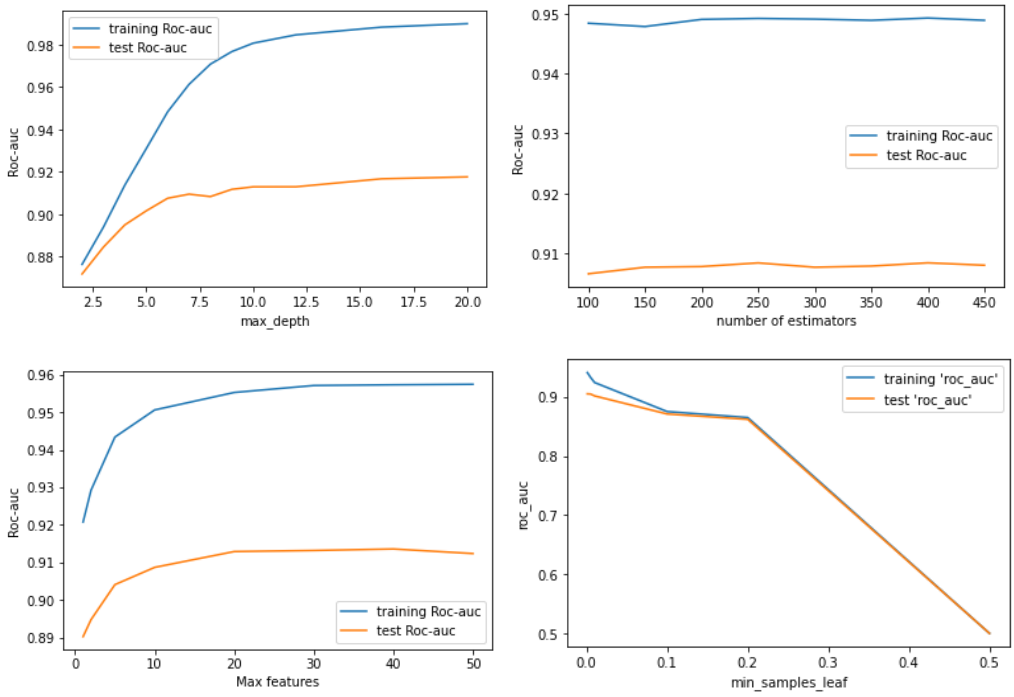
Tuned Parameters: 'max_depth', 'min_samples_leaf' , 'min_samples_split' and 'criterion'.

Max_depth controls depth of Decision tree and min_samples_leaf and min_samples_split are used to control the split properties and criterion is the criteria to use i.e [gini , entropy].

d) Random Forest:

When creating random forests, a unique ensemble technique known as bagging is used. The term "bagging" refers to Bootstrap Aggregation. Making bootstrap samples from a given data collection is referred to as bootstrapping. By uniformly and with replacement sampling of the provided data set, a bootstrap sample is produced.

Tuned Parameters: 'max_depth', 'n_estimators', 'max_features', 'min_samples_leaf' and 'min_samples_split'



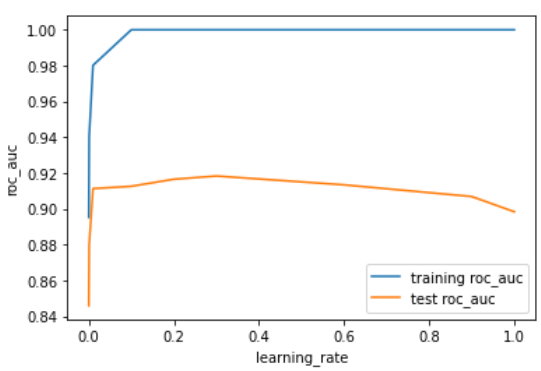
Final Random Forest Model:

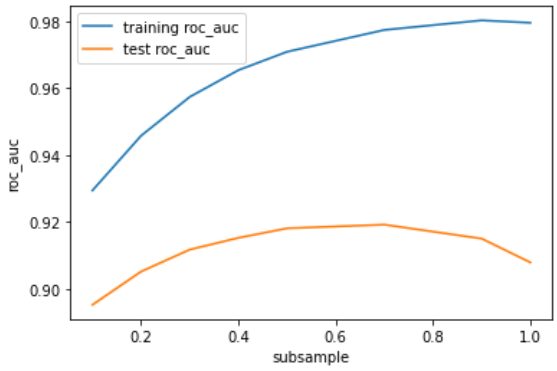
```
rf1 = RandomForestClassifier( class_weight= 'balanced' , max_depth = 6 , n_estimators = 200 ,max_features= 5,
                             min_samples_leaf= 0.1, min_samples_split= 0.01 ,bootstrap= True)
rf1.fit(X_train_pca,y_train)
```

e) XGBoost Model:

XGBoost is an algorithm that is widely used in industry because of its efficacy. Also, it has been widely used in many Machine Learning contests and hackathons and has been the top-contender algorithm in multiple domains. They work on the concept of boosting weak learners and learning from residuals. Unlike Bagging algorithms, XGBoost works well even on smaller sample sets. It is an improvised version of the original Bagging algorithm.

Tuned Parameters: 'learning_rate', 'subsample'



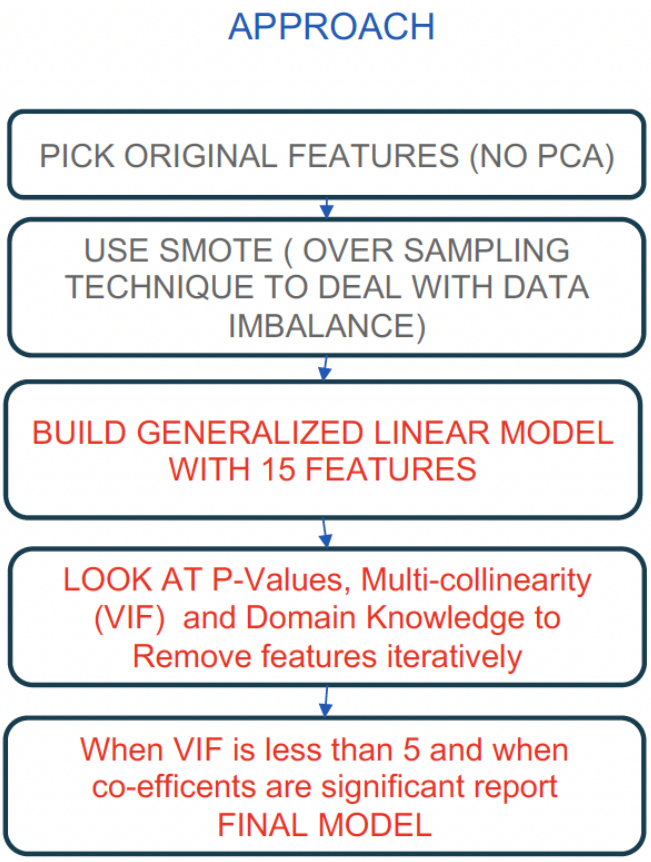


Final XGB Classifier

```
xgb_model = XGBClassifier(n_estimators= 300,max_depth = 6 , learning_rate= 0.001, subsample= 0.1
                          ,scale_pos_weight= 92.271 , max_features = 5,min_samples_leaf= 0.1, min_samples_split= 0.01)
xgb_model.fit(X_train_pca,y_train)
```

```
XGBClassifier(learning_rate=0.001, max_depth=6, max_features=5,
              min_samples_leaf=0.1, min_samples_split=0.01, n_estimators=300,
              scale_pos_weight=92.271, subsample=0.1)
```

INTERPRETABLE MODELS (IDENTIFYING CHURN INDICATORS)



To build generalized linear models and interpret coefficients, p values, and log-likelihood we have used statsmodels library in Python which has same summary metric report as R.

a) INITIAL MODEL:

Generalized Linear Model Regression Results

Dep. Variable:	churn	**No. Observations:**	31732
Model:	GLM	**Df Residuals:**	31716
Model Family:	Binomial	**Df Model:**	15
Link Function:	logit	**Scale:**	1.0000
Method:	IRLS	**Log-Likelihood:**	-10272.
Date:	Wed, 30 Nov 2022	**Deviance:**	20545.
Time:	08:50:42	**Pearson chi2:**	4.96e+04
No. Iterations:	7		
Covariance Type:	nonrobust		

	coef	std err	z	P>|z|	[0.025	0.975]
const	2.1475	0.073	29.376	0.000	2.004	2.291
arpu_7	1.6016	0.119	13.467	0.000	1.368	1.835
av_rech_amt_data_8	-3.3361	0.138	-24.099	0.000	-3.607	-3.065
last_day_rch_amt_8	-3.3864	0.103	-32.985	0.000	-3.588	-3.185
max_rech_amt_8	-2.7248	0.175	-15.586	0.000	-3.067	-2.382
offnet_mou_6	2.6756	0.208	12.892	0.000	2.269	3.082
onnet_mou_6	2.7489	0.194	14.194	0.000	2.369	3.129
std_og_mou_8	3.0461	0.268	11.379	0.000	2.521	3.571
total_ic_mou_7	1.6595	0.149	11.174	0.000	1.368	1.951
total_ic_mou_8	-5.1991	0.216	-24.123	0.000	-5.622	-4.777
total_og_mou_6	-3.3786	0.279	-12.102	0.000	-3.926	-2.831
total_og_mou_8	-5.5148	0.349	-15.785	0.000	-6.200	-4.830
total_rech_amt_8	2.1911	0.245	8.925	0.000	1.710	2.672
total_rech_num_8	-2.5194	0.130	-19.348	0.000	-2.775	-2.264
days_last_rech_month_8	2.6598	0.082	32.515	0.000	2.499	2.820
8th_month_decrease_roam_og_mou	-1.6138	0.049	-32.808	0.000	-1.710	-1.517

VIF (variance inflation factor) - Assigning cutoff as 3

5]:
```
from statsmodels.stats.outliers_influence import variance_inflation_factor
vif = pd.DataFrame()
vif['Features'] = X_train_log[rfe_col].columns
vif['VIF'] = [variance_inflation_factor(X_train_log[rfe_col].values, i) for i in range(X_train_log[rfe_col].shape[1])]
vif['VIF'] = round(vif['VIF'], 2)
vif = vif.sort_values(by = "VIF", ascending = False)
vif
```

5]:

	Features	VIF
9	total_og_mou_6	36.70
4	offnet_mou_6	16.83
11	total_rech_amt_8	16.25
10	total_og_mou_8	13.86
5	onnet_mou_6	13.58
6	std_og_mou_8	6.70
3	max_rech_amt_8	6.12
8	total_ic_mou_8	5.86
7	total_ic_mou_7	5.35
12	total_rech_num_8	4.64
0	arpu_7	4.43
14	8th_month_decrease_roam_og_mou	3.66
2	last_day_rch_amt_8	2.87
13	days_last_rech_month_8	2.62
1	av_rech_amt_data_8	2.15

b) FINAL MODEL:

Generalized Linear Model Regression Results

Dep. Variable:	churn	**No. Observations:**	31732
Model:	GLM	**Df Residuals:**	31723
Model Family:	Binomial	**Df Model:**	8
Link Function:	logit	**Scale:**	1.0000
Method:	IRLS	**Log-Likelihood:**	-11683.
Date:	Wed, 30 Nov 2022	**Deviance:**	23365.
Time:	08:50:56	**Pearson chi2:**	2.61e+05
No. Iterations:	6		
Covariance Type:	nonrobust		

	coef	std err	z	P>|z|	[0.025	0.975]
const	1.0133	0.048	21.231	0.000	0.920	1.107
av_rech_amt_data_8	-4.2457	0.121	-35.116	0.000	-4.483	-4.009
days_last_rech_month_8	2.5272	0.074	33.980	0.000	2.381	2.673
last_day_rch_amt_8	-3.1694	0.080	-39.775	0.000	-3.326	-3.013
offnet_mou_6	0.6471	0.071	9.163	0.000	0.509	0.786
onnet_mou_6	0.8987	0.068	13.307	0.000	0.766	1.031
std_og_mou_8	-1.1764	0.087	-13.453	0.000	-1.348	-1.005
total_ic_mou_8	-6.0017	0.135	-44.555	0.000	-6.266	-5.738
total_rech_num_8	-1.1894	0.092	-12.864	0.000	-1.371	-1.008

We can see here that all P-values are significant and also VIF were less than 2. Thus, let's see the variables influencing churn linearly.

RESULTS (PREDICTIVE MODELS)

Metrics of various predictive models on train data

```
round(algo_data,2) * 100
```

	Accuracy	Sensitivity	F1-score	ROC-auc
LogisticRegression	86.0	87.0	50.0	87.0
RidgeClassifier	81.0	86.0	42.0	83.0
DecisionTreeClassifier	81.0	89.0	42.0	85.0
RandomForestClassifier	77.0	82.0	35.0	79.0
AdaBoostClassifier	85.0	83.0	46.0	84.0
XGBClassifier	77.0	86.0	37.0	81.0

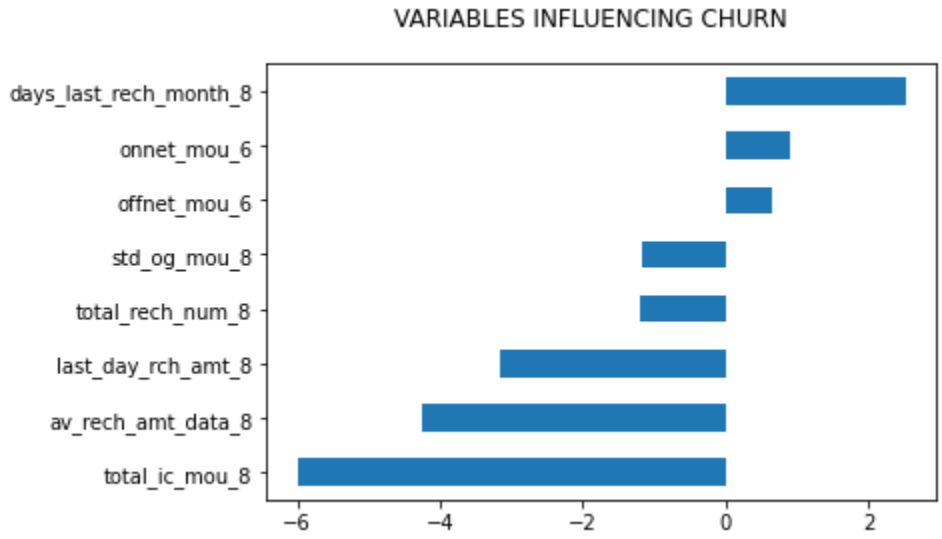
Metrics of various predictive models on test data

```
round(algo_test,2) * 100
```

	Accuracy	Sensitivity	F1-score	ROC-auc
LogisticRegression	86.0	85.0	48.0	85.0
RidgeClassifier	82.0	85.0	42.0	83.0
DecisionTreeClassifier	81.0	82.0	39.0	81.0
RandomForestClassifier	77.0	81.0	36.0	79.0
AdaBoostClassifier	85.0	81.0	45.0	83.0
XGBClassifier	77.0	84.0	36.0	80.0

Based on the comparison of scores of various models on the train-data and unseen data we see that logistic regression generalizes well. The metrics across the train and test-set are almost similar and also the goal of the case study is to predict churners . So,sensitivity is an important and major factor. Also, Logistic Regression generalizes well. Taking these factors into consideration lets report the LogisticRegression model for deployment

INFERENCE AND RECOMMENDATIONS



From the interpretable model we have built, it is clearly evident that most of the customers show different behavior(churn behavior) in the action months compared to good months. During this period the customer may have become unhappy with the service quality or there might be better offers from other network providers in the market.

Recommendations

a) Constantly provide offers and value added services to the customer

b) Adjust pricing dynamics by looking at the variation in pricing of competitors

c) Pricing is directly proportional to the amount of calls the customer makes. So, try to reduce STD rates as it may increase the STD minutes of usage

d) 'days_last_rech_month_8' indicates no of days ago relative to the last day of the month the customer has made his last recharge. So, people who generally recharge early on in the month generally churn in the next month. So, the company can focus on such customers by providing good services and offers to avoid churn

e) Monitor high value customers who recharge data packs comparatively lesser than the previous months

AUTHOR CONTRIBUTIONS

Madhavan Rangarajan	**50442679**	**Building Predictive and Interpretable ML Models and Report Writing**
K. Bindu Harshita	**50479393**	**Exploratory Data Analysis, PCA and Report Writing**
Bysani Sai Ramya Sree	**50455444**	**Data Understanding and Data Pre-processing**
Sreeja Manchikanti	**50442935**	**Building Predictive Models**
Nikhitha Mekala	**50461360**	**Presentation**

REFERENCES:

https://www.kaggle.com/code/gauravduttakiit/telecom-customer-churn-eda/notebook

https://xgboost.readthedocs.io/en/stable/

https://www.analyticsvidhya.com/blog/2020/03/beginners-guide-random-forest-hyperparameter-tuning/

https://www.mygreatlearning.com/blog/gridsearchcv/

https://scikit-learn.org/stable/modules/generated/sklearn.model_selection.GridSearchCV.html