ANALYSIS AND MODELLING OF CUSTOMER CHURN IN TELECOM INDUSTRY

PROJECT GROUP 3

Ramya Bysani
Bindu Harshita Kudithi
Sreeja Manchikanti
Nikhitha Mekala
Madhavan Rangarajan

University at Buffalo
Department of Computer Science and Engineering
School of Engineering and Applied Sciences

Contents

Problem Statement

- In the telecom industry, the customer churn rate is usually high.
- The cost of retaining an existing customer is 1/10th of the cost of acquiring a new customer.
- Retention of existing customers is far more critical than acquiring customers
- Can we use statistical analysis and data mining techniques to reduce churn?
- If so, what are the factors that impact churn and is there a pattern between customers who exhibit high risk of churn?

Motivation

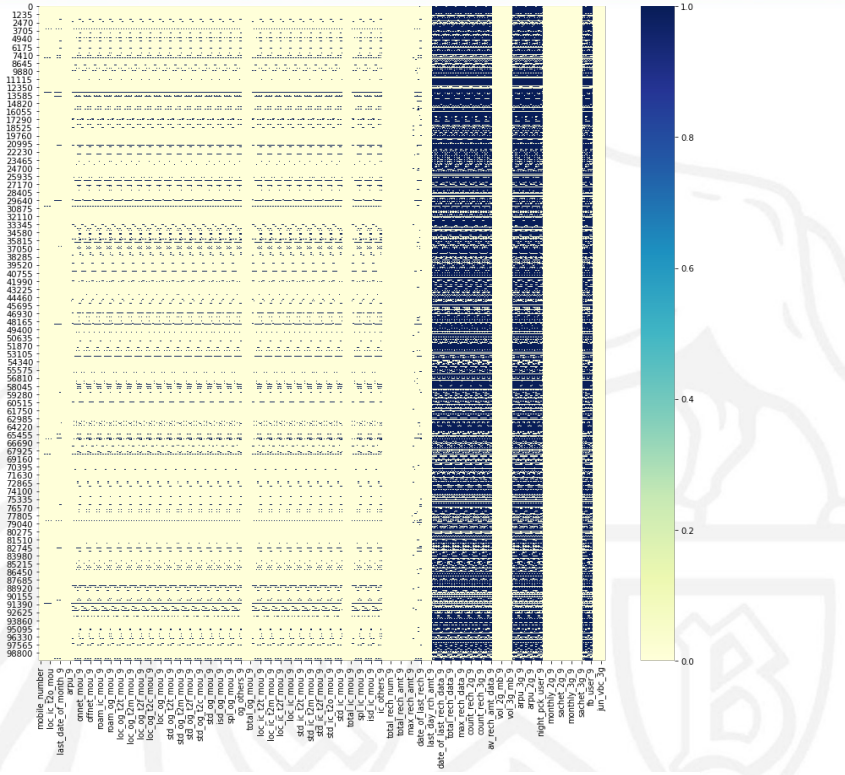
- Prepaid and post-paid are the two widely known billing options in the telecom industry.
- Unlike the post-paid option where customers contact service provider before switching the service, the pre-paid customers stop using the service without any prior notice.
- Thus, understanding behaviour for prepaid customers is most critical.
- The success of this study will give new directions in churn prediction and can improve business decision making for telecom companies

Data Understanding

- Predict the customer churn based on the data collected for 4 months i.e (June, July, August and September respectively)
- First 2 months describe the customer behaviour in the good phase
- Next 2 months can be attributed to action phase and churn phase respectively
- The dataset contains 226 features
- The dataset contains 99999 entries

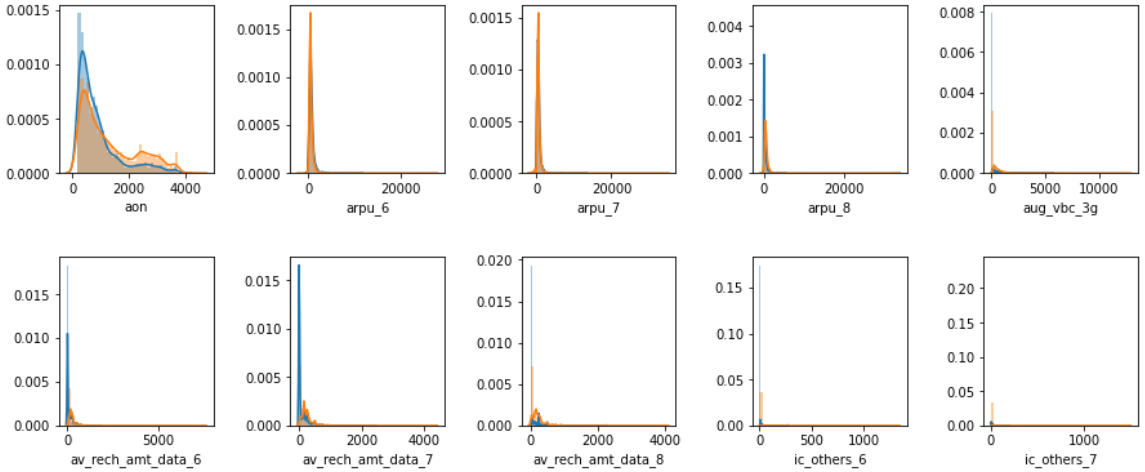
Data Pre-Processing

- Some features where missing values are present actually indicate zero. So imputed those features with zero.
- Deleted features with missing values > 50%
- Derived features based on existing features
- Imputed features with low missing values < 2% using KNN Imputer
- Created dummy variables (0/1) for categorical variables
- Removing features which are redundant



```
count_rech_2g_6              74.846748
date_of_last_rech_data_6     74.846748
count_rech_3g_6              74.846748
av_rech_amt_data_6           74.846748
max_rech_data_6              74.846748
                                 ...
last_date_of_month_8          1.100011
loc_ic_t2o_mou                1.018010
std_og_t2o_mou                1.018010
loc_og_t2o_mou                1.018010
last_date_of_month_7          0.601006
Length: 166, dtype: float64
```

Exploratory Data Analysis



Distribution of variables across ‘Churners’ and ‘Non Churners’

Outlier Treatment

```
for x in col_to_rem_outlier:

    q1 = high_impute[x].quantile(0.25)
    q3 = high_impute[x].quantile(0.75)

    iqr = q3 - q1

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

    out_treat = out_treat[out_treat[x].between(lower_bound, upper_bound, inclusive=True)]
```

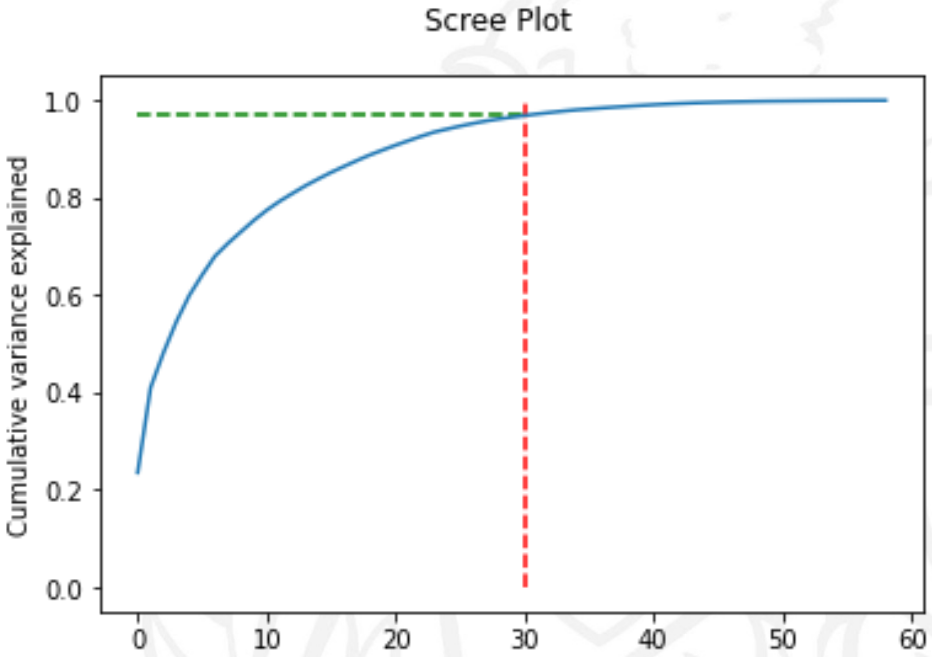
```
len(out_treat) * 100/len(high_impute)
```

```
76.55347423954055
```

We have lost about 25% of the data after removing outliers based on IQR

Principal Component Analysis

- As there are lot of features , building algorithms and tuning hyper parameters is not a good idea.
- It might become computationally expensive
- PCA as a dimensionality reduction technique (on numerical features) to reduce the feature space.
- Build statistical models on first 30 principal components.



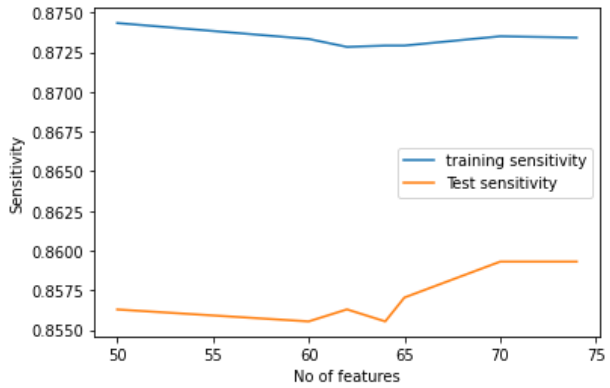
30 PC's capture upto 97% variation in the data

Predictive Modelling with Hyperparameter Tuning

Logistic Regression

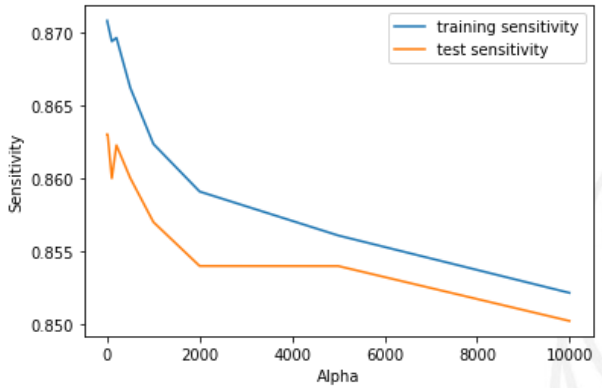
```
from sklearn.feature_selection import RFE
cv = StratifiedKFold(n_splits=10 ,shuffle = True, random_state=1)
hyper_params = [{'n_features_to_select': [50,60,62,64,65,70,len(X_train_pca.columns)]}]
```

```
lr = LogisticRegression( class_weight = 'balanced' , max_iter= 1000 )
lr.fit(X_train_pca,y_train )

rfe = RFE(lr)

model_cv = GridSearchCV(estimator = rfe,
                        param_grid = hyper_params,
                        scoring= make_scorer(recall_score),
                        cv = cv,
                        verbose = 1,
                        return_train_score=True , n_jobs= -1)
model_cv.fit(X_train_pca,y_train)
```

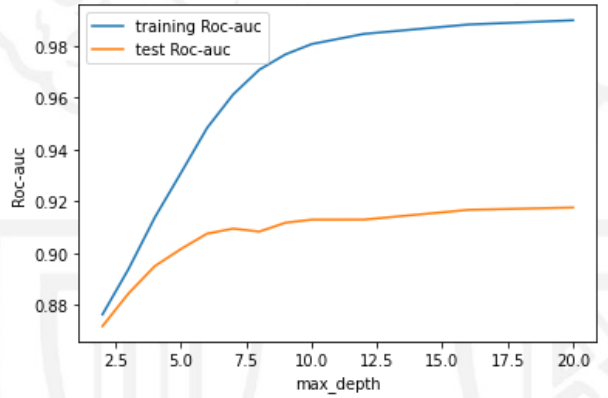


70 features looks optimum

Ridge Classifier



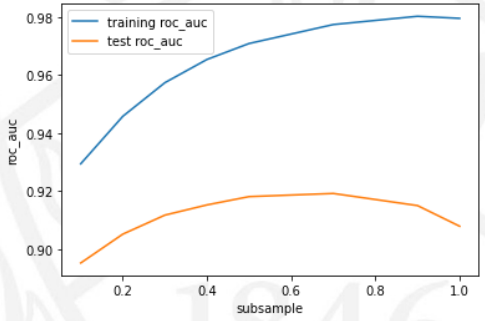
Random Forest



Decision Tree

```
grid = {
    'max_depth': range(1,50,5),
    'min_samples_leaf': range(1, 500, 50),
    'min_samples_split': range(1, 500, 50),
    'criterion': ["entropy", "gini"]
}
```

XGBoost



Interpretable model (Identifying churn indicators)

APPROACH

PICK ORIGINAL FEATURES (NO PCA)

↓

USE SMOTE (OVER SAMPLING TECHNIQUE TO DEAL WITH DATA IMBALANCE)

↓

BUILD GENERALIZED LINEAR MODEL WITH 15 FEATURES

↓

LOOK AT P-Values, Multi-collinearity (VIF) and Domain Knowledge to Remove features iteratively

↓

When VIF is less than 5 and when co-efficents are significant report FINAL MODEL

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

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

	Features	VIF
7	total_rech_num_8	2.73
2	last_day_rch_amt_8	1.99
5	std_og_mou_8	1.98
6	total_ic_mou_8	1.96
3	offnet_mou_6	1.86
1	days_last_rech_month_8	1.82
4	onnet_mou_6	1.67
0	av_rech_amt_data_8	1.59

Results

Predictive Models

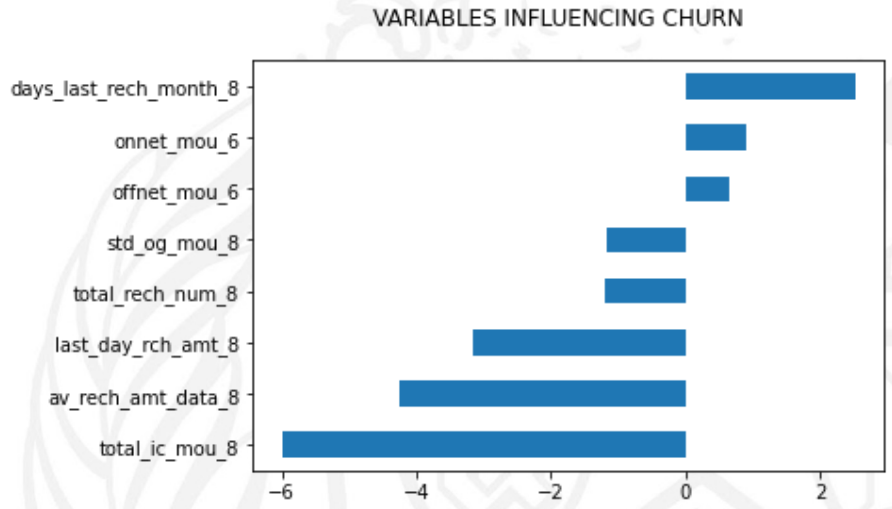
	Accuracy	Sensitivity	F1-score	ROC-auc
LogisticRegression	0.864903	0.872084	0.499461	0.868193
RidgeClassifier	0.814423	0.856283	0.416316	0.833600
DecisionTreeClassifier	0.813841	0.888638	0.424591	0.848107
RandomForestClassifier	0.768712	0.819413	0.353859	0.791939
AdaBoostClassifier	0.848502	0.833710	0.459656	0.841726
XGBClassifier	0.768596	0.860798	0.365087	0.810835

Metrics on Train Data

	Accuracy	Sensitivity	F1-score	ROC-auc
LogisticRegression	0.858514	0.846501	0.480461	0.853010
RidgeClassifier	0.817341	0.848758	0.418010	0.831734
DecisionTreeClassifier	0.805129	0.817156	0.393265	0.810639
RandomForestClassifier	0.774424	0.814898	0.358313	0.792966
AdaBoostClassifier	0.845255	0.810384	0.447352	0.829280
XGBClassifier	0.769016	0.839729	0.359768	0.801411

Metrics on Test Data

Interpretable Model



Train Accuracy - 0.856
Train Sensitivity - 0.856

Test Accuracy - 0.857
Test Sensitivity - 0.810

Recommendations

- Based on the comparison of scores of various models on the train-data and unseen data we see that ,logistic regression generalizes well.
- The goal of the case study is to predict churners. Thus, sensitivity is an important metric
- From the interpretable model we have built, it is clearly evident that most of the customers show different behaviour during last 2 months (churn phase)
- '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, before churning the customer might not recharge.