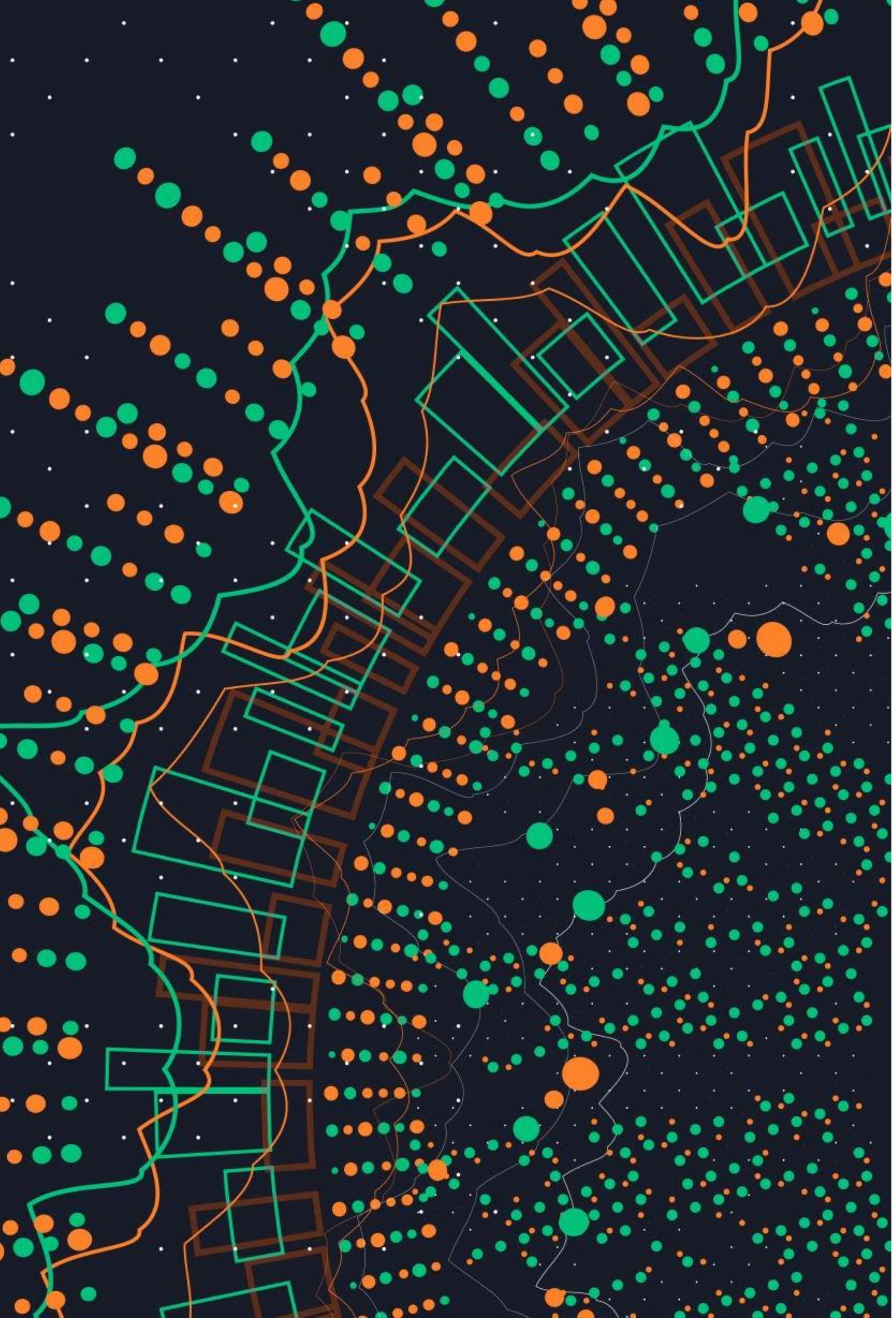


Capstone Project- Telecom Churn

BY - JANSHI RANI P

DATE - 22ND SEPTEMBER 2024

COURSE NAME - UPGRAD-GRADUATE CERTIFICATE PROGRAM IN DATA SCIENCE-FEBRUARY 2024

Telecom Churn Analysis

Problem Statement: To enhance customer retention and profitability, a leading telecom firm seeks to proactively identify and address customer churn. By leveraging customer-level data, we aim to develop predictive models that accurately forecast high-risk customers. This will enable the company to implement targeted retention strategies, optimize customer service efforts and ultimately improve overall business performance.

Business Impact: Emphasizes the importance of churn prediction for improving customer retention and profitability.

Steps for Telecom churn Analysis:

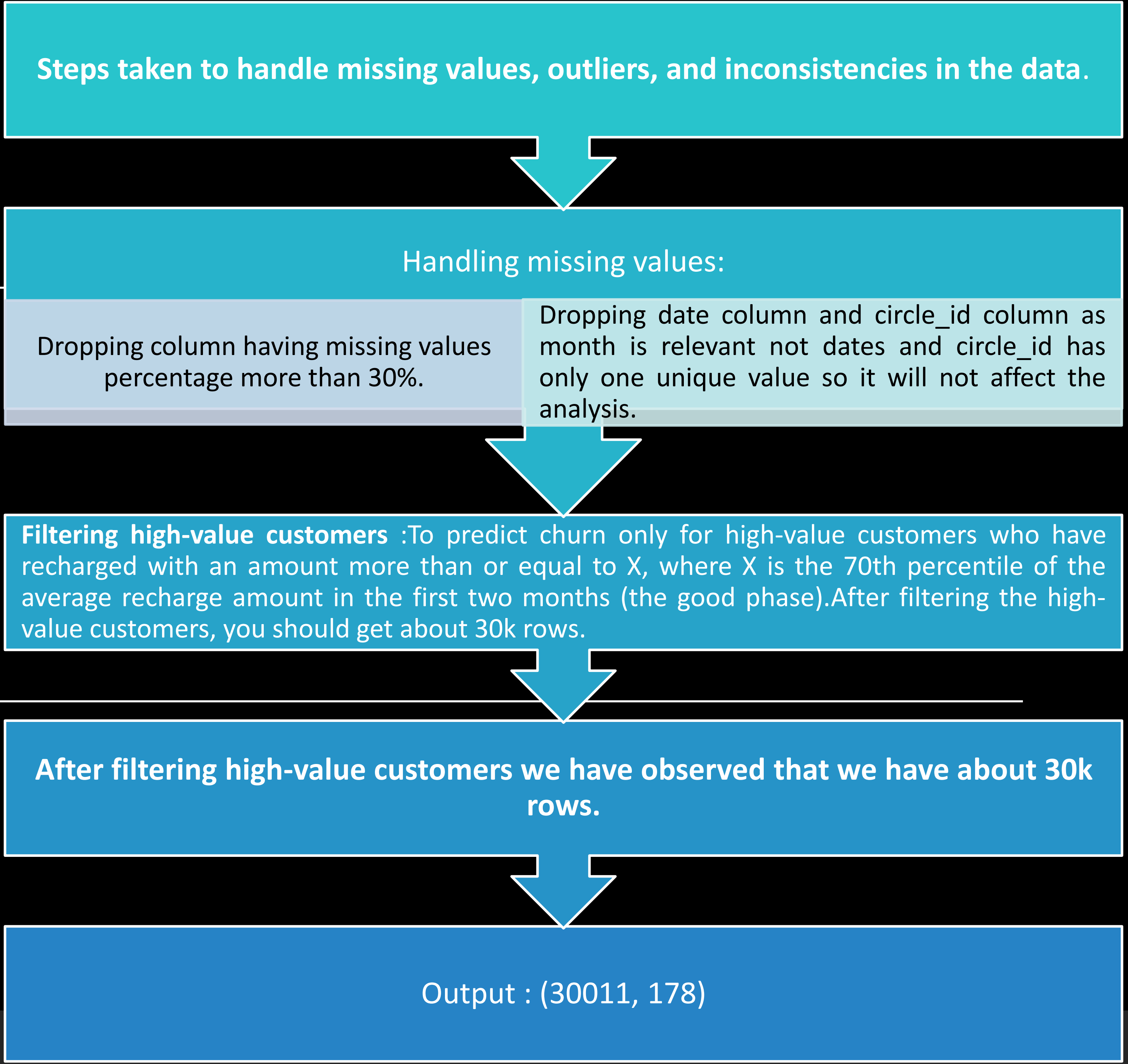
Python script provides a comprehensive approach to the telecom churn prediction project. Here's a brief overview of what each section does:

- **Import Libraries**: Imports necessary Python libraries for data analysis and modelling.
- **Load and Examine Data**: Loads the dataset and displays basic information.
- **Data Preparation**: Filters high-value customers and tags churners.
- **Exploratory Data Analysis**: Includes a visualization to analyse churn patterns.
- **Feature Engineering**: Handles missing values and creates new features.
- **Modelling**: Splits the data, scales features, handles class imbalance, and trains a logistic regression model.
- **Model Evaluation**: Evaluates the model's performance using various metrics.
- **Feature Importance and Interpretation**: Visualizes the importance of different features in predicting churn.
- **Business Recommendations**: Provides sample recommendations based on the analysis.

Data Exploration & Preprocessing

- Dataset Overview : Dataset & Data Dictionary are provided in my github repository.
- Data Size : (99999, 226),
- dtypes: float64(179), int64(35), object(12).

Data Cleaning & Data Preparation:



Steps taken to handle missing values, outliers, and inconsistencies in the data.

Handling missing values:

Dropping column having missing values percentage more than 30%.

Dropping date column and circle_id column as month is relevant not dates and circle_id has only one unique value so it will not affect the analysis.

Filtering high-value customers :To predict churn only for high-value customers who have recharged with an amount more than or equal to X, where X is the 70th percentile of the average recharge amount in the first two months (the good phase).After filtering the high-value customers, you should get about 30k rows.

After filtering high-value customers we have observed that we have about 30k rows.

Output : (30011, 178)

Handling missing values

Data Quality Assessment:

- Identified missing values in MOU columns for June (6), July (7), August (8), and September (9).
- Observed that missing values for all four months were consistent for each record.

Dataset Shape:

- Final dataset shape: (27591, 17)
- Lost approximately **7%** of records due to missing data removal.
- Despite the loss, the remaining dataset is still sufficient for analysis.

Tag Churners

Churn Definition:

- Customers are considered churned if they have no incoming or outgoing calls and no mobile internet usage in the fourth month (churn phase).

Attributes Used to Tag Churners:

- total_ic_mou_9: Total incoming call minutes in the ninth month
- total_og_mou_9: Total outgoing call minutes in the ninth month
- vol_2g_mb_9: Total 2G data usage in the ninth month
- vol_3g_mb_9: Total 3G data usage in the ninth month

Feature Removal:

- Removed attributes corresponding to the churn phase to avoid data leakage.
- This ensures that the model doesn't use information from the future to make predictions.

Churn Percentage:

- Observed churn percentage: **3.39%**
- This indicates a relatively low churn rate.

Outlier Handling

Data Types:

- Identified all columns as numeric except mobile_number and churn.
  - Index(['loc_og_t2o_mou', 'std_og_t2o_mou', 'loc_ic_t2o_mou', 'arpu_6', 'arpu_7', 'arpu_8', 'onnet_mou_6', 'onnet_mou_7', 'onnet_mou_8', 'offnet_mou_6', ... 'monthly_3g_7', 'monthly_3g_8', 'sachet_3g_6', 'sachet_3g_7', 'sachet_3g_8', 'aon', 'aug_vbc_3g', 'jul_vbc_3g', 'jun_vbc_3g', 'avg_rech_amt_6_7'], dtype='object', length=134)
- Converted mobile_number and churn to object data types

Outlier Removal:

- Removed outliers from numeric columns.
- Outliers defined as values below the 10th percentile and above the 90th percentile.
- This step helps to improve model accuracy and prevent outliers from skewing results.
- **After outlier removal data.shape : (27705, 136)**

Deriving new features

Below features are designed to capture changes in customer behaviour during the "action phase" compared to the "good phase."

Here's a breakdown of each new feature:

decrease_mou_action: This feature indicates whether the customer's minutes of usage have decreased in the action phase compared to the good phase.

decrease_rech_num_action: This feature indicates whether the customer's number of recharges has decreased in the action phase compared to the good phase.

decrease_rech_amt_action: This feature indicates whether the customer's recharge amount has decreased in the action phase compared to the good phase.

decrease_arpu_action: This feature indicates whether the customer's average revenue per user (ARPU) has decreased in the action phase compared to the good phase.

decrease_vbc_action: This feature indicates whether the customer's volume-based cost has decreased in the action phase compared to the good phase.

Exploratory Data Analysis (EDA)

- Univariate analysis
- Bivariate analysis

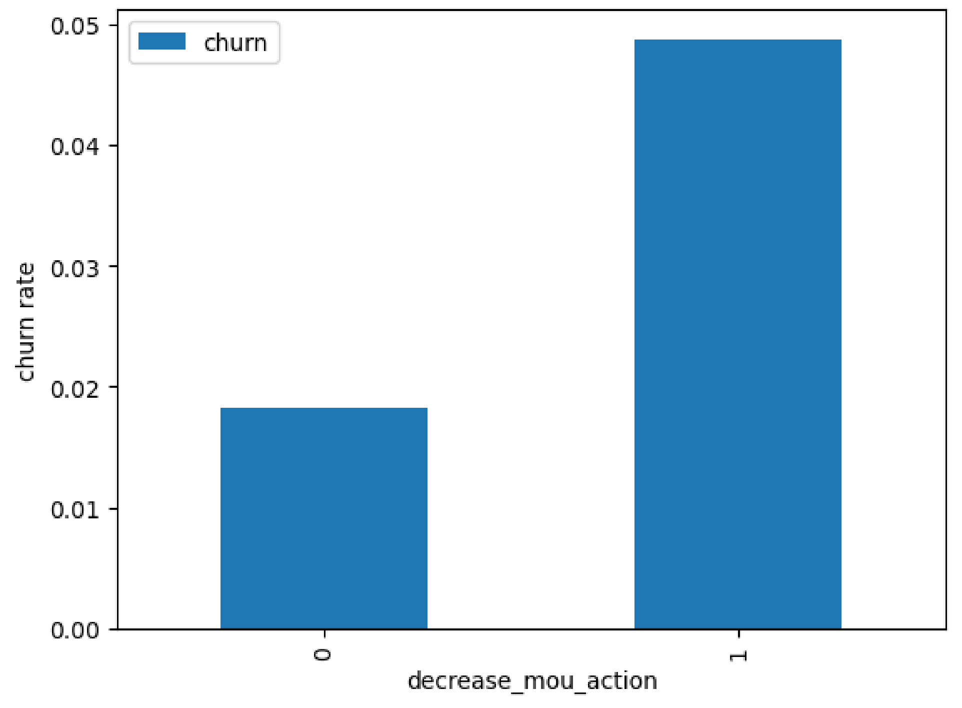
Exploratory Data Analysis (EDA): Univariate Analysis:

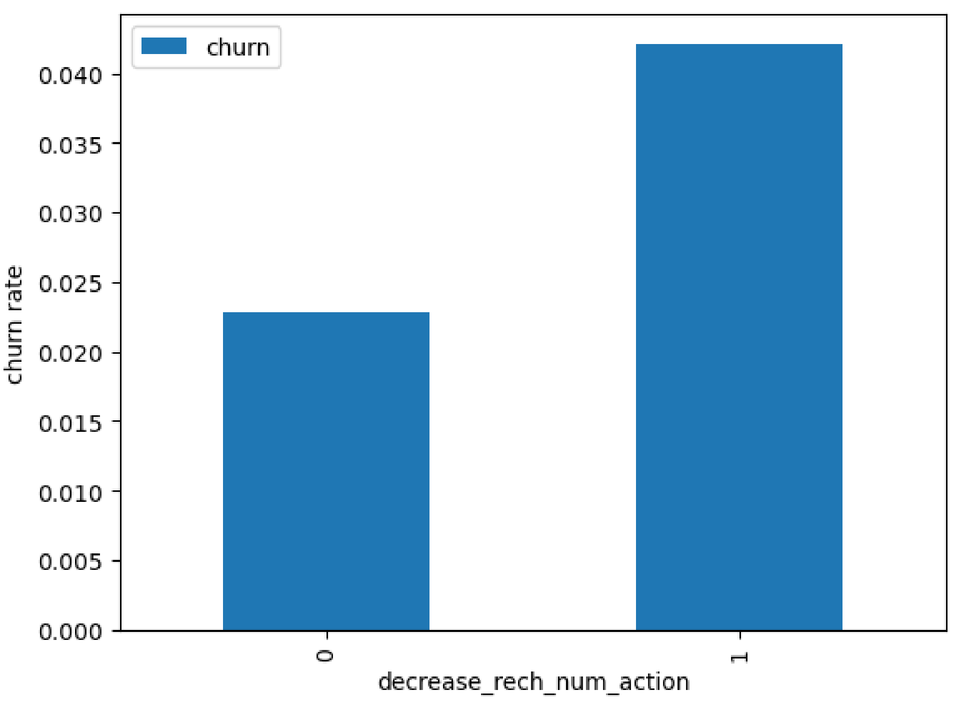
- **Churn rate on the basis whether the customer decreased her/his MOU in action month.**

Insight : We can see that the churn rate is more for the customers, whose minutes of usage(mou) decreased in the action phase than the good phase.

- .Churn rate on the basis whether the customer decreased her/his number of recharge in action month.

Insight : As expected, the churn rate is more for the customers, whose number of recharge in the action phase is lesser than the number in good phase.





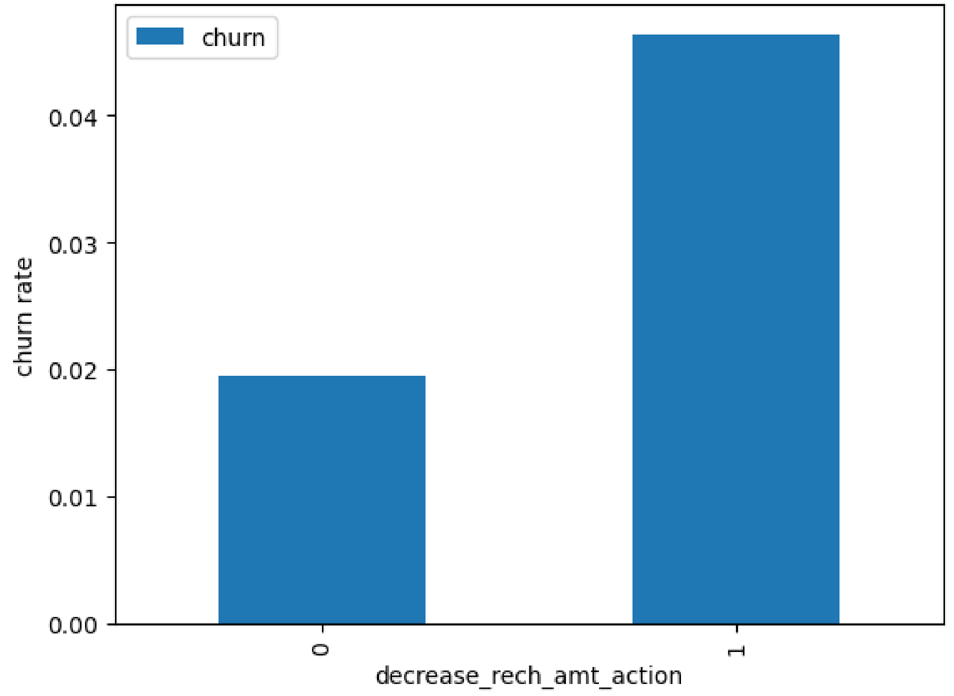
EDA: **Univariate Analysis**

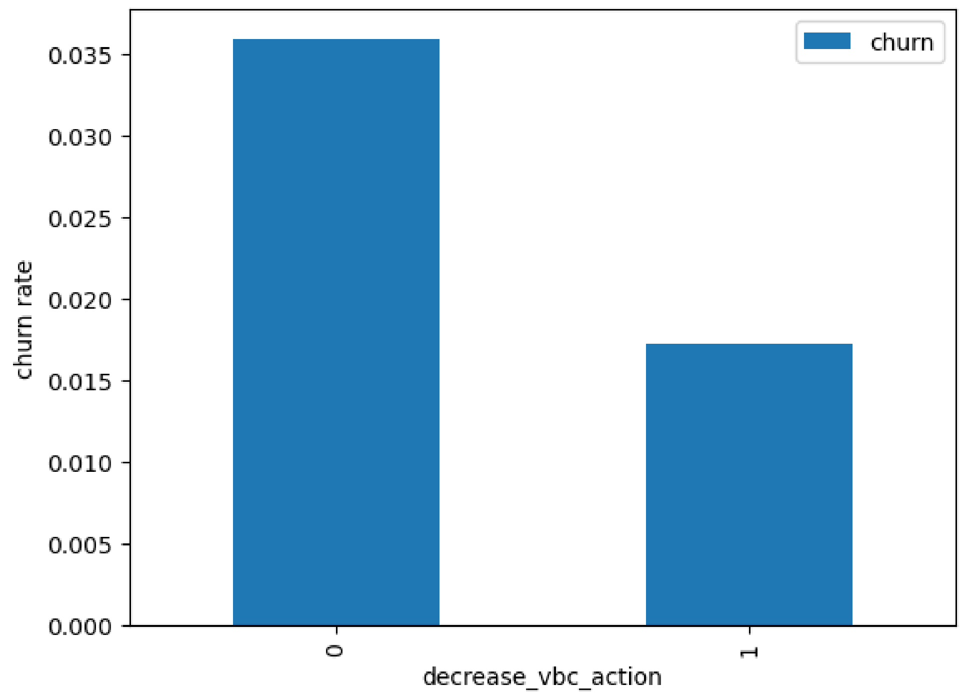
- Churn rate on the basis whether the customer decreased her/his amount of recharge in action month.

Insight : Here also we see the same behaviour. The churn rate is more for the customers, whose amount of recharge in the action phase is lesser than the amount in good phase.

- Churn rate on the basis whether the customer decreased her/his volume based cost in action month

Insight : Here we see the expected result. The churn rate is more for the customers, whose volume based cost in action month is increased. That means the customers do not do the monthly recharge more when they are in the action phase.





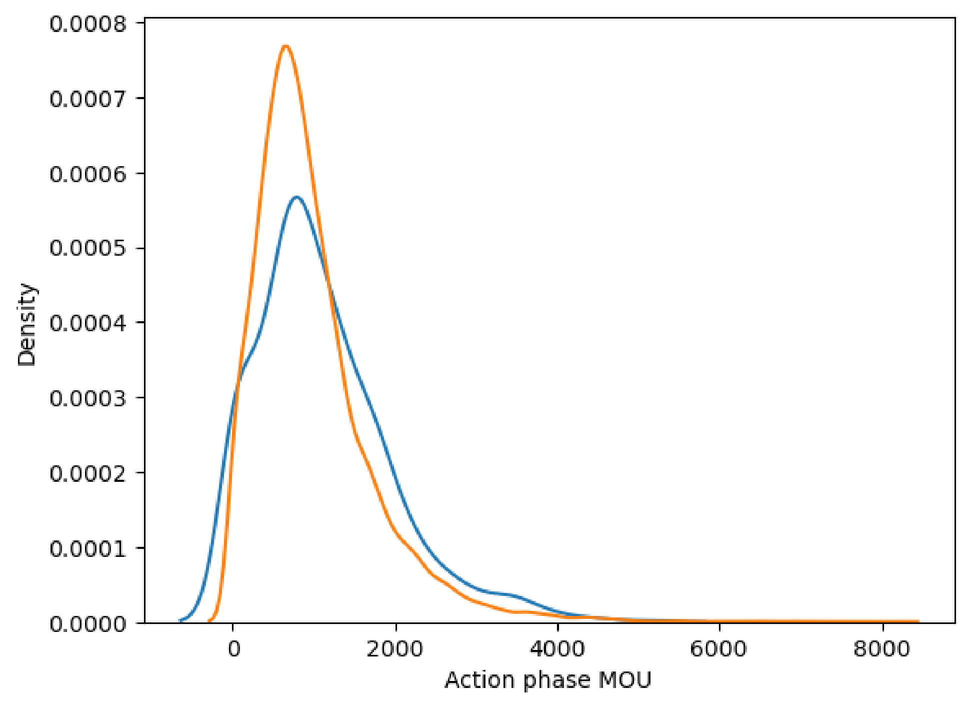
EDA

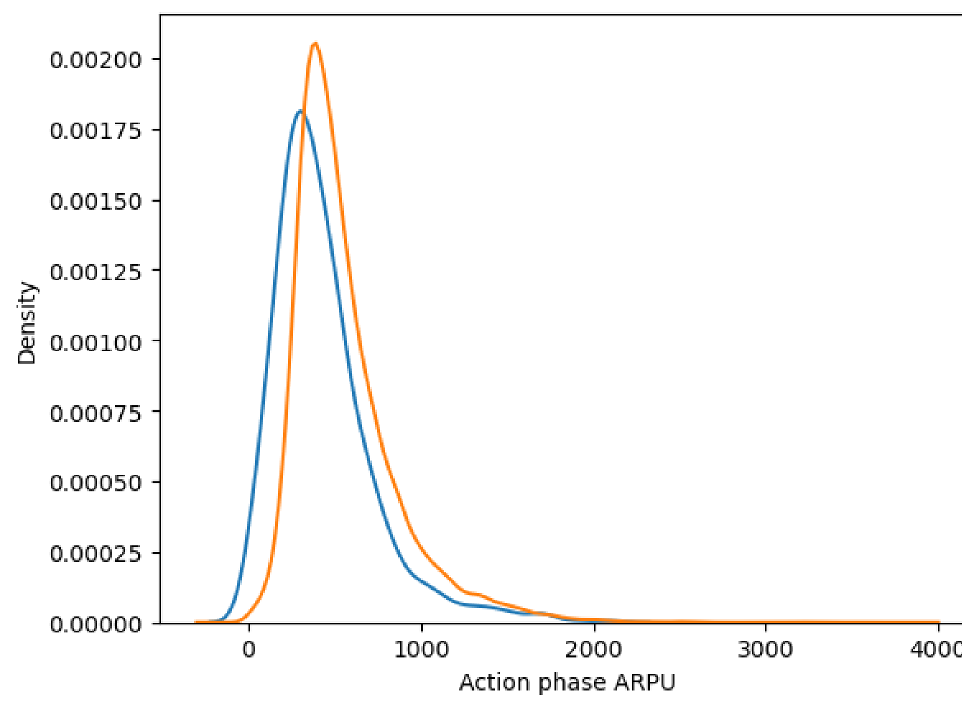
- Analysis of the average revenue per customer (churn and not churn) in the action phase.

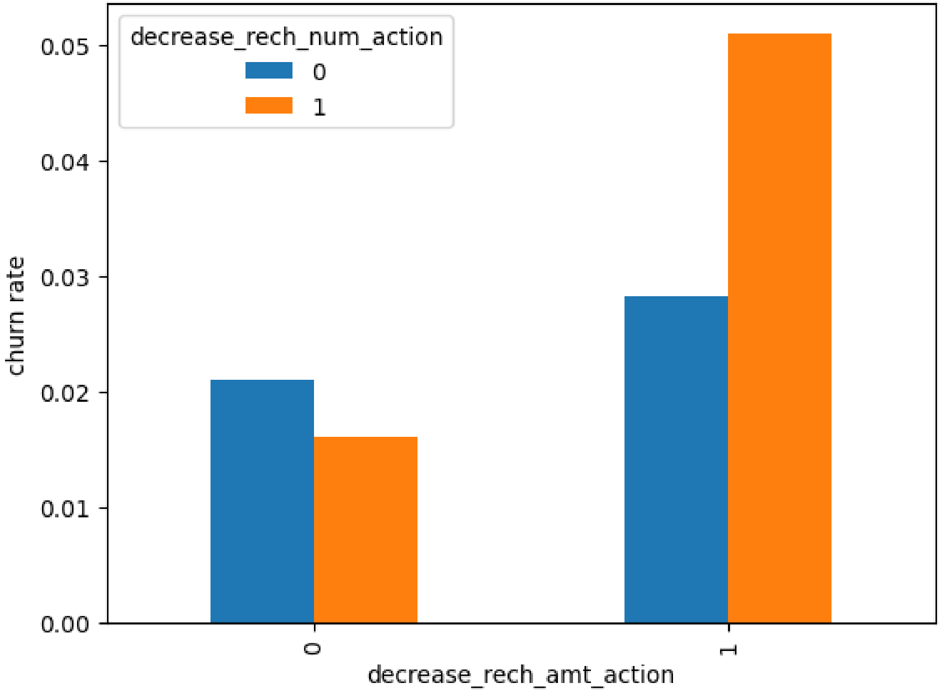
Insight : Average revenue per user (ARPU) for the churned customers is mostly densed on the 0 to 900. The higher ARPU customers are less likely to be churned. ARPU for the not churned customers is mostly densed on the 0 to 1000.

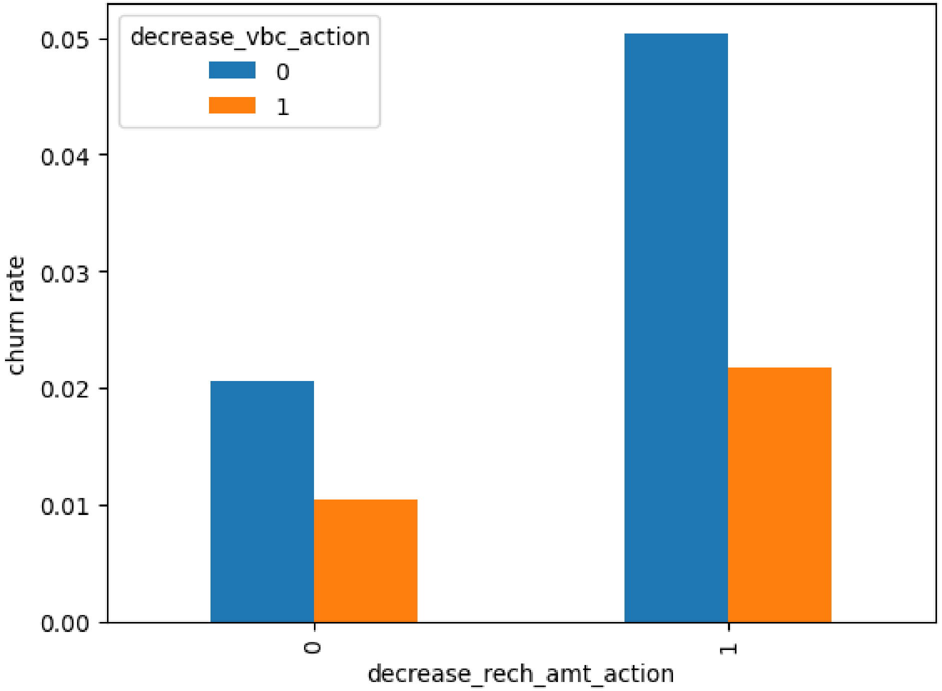
- Analysis of the minutes of usage MOU (churn and not churn) in the action phase.

Insight : Analysis of the minutes of usage MOU (churn and not churn) in the action phase.









Bivariate analysis

- Analysis of churn rate by the decreasing recharge amount and number of recharge in the action phase.

Insight: Customers who exhibit a decline in both recharge amount and frequency during the action phase are more likely to churn. This suggests that these customers are reducing their spending and engagement with the service, indicating a potential loss of interest or dissatisfaction with the value proposition.

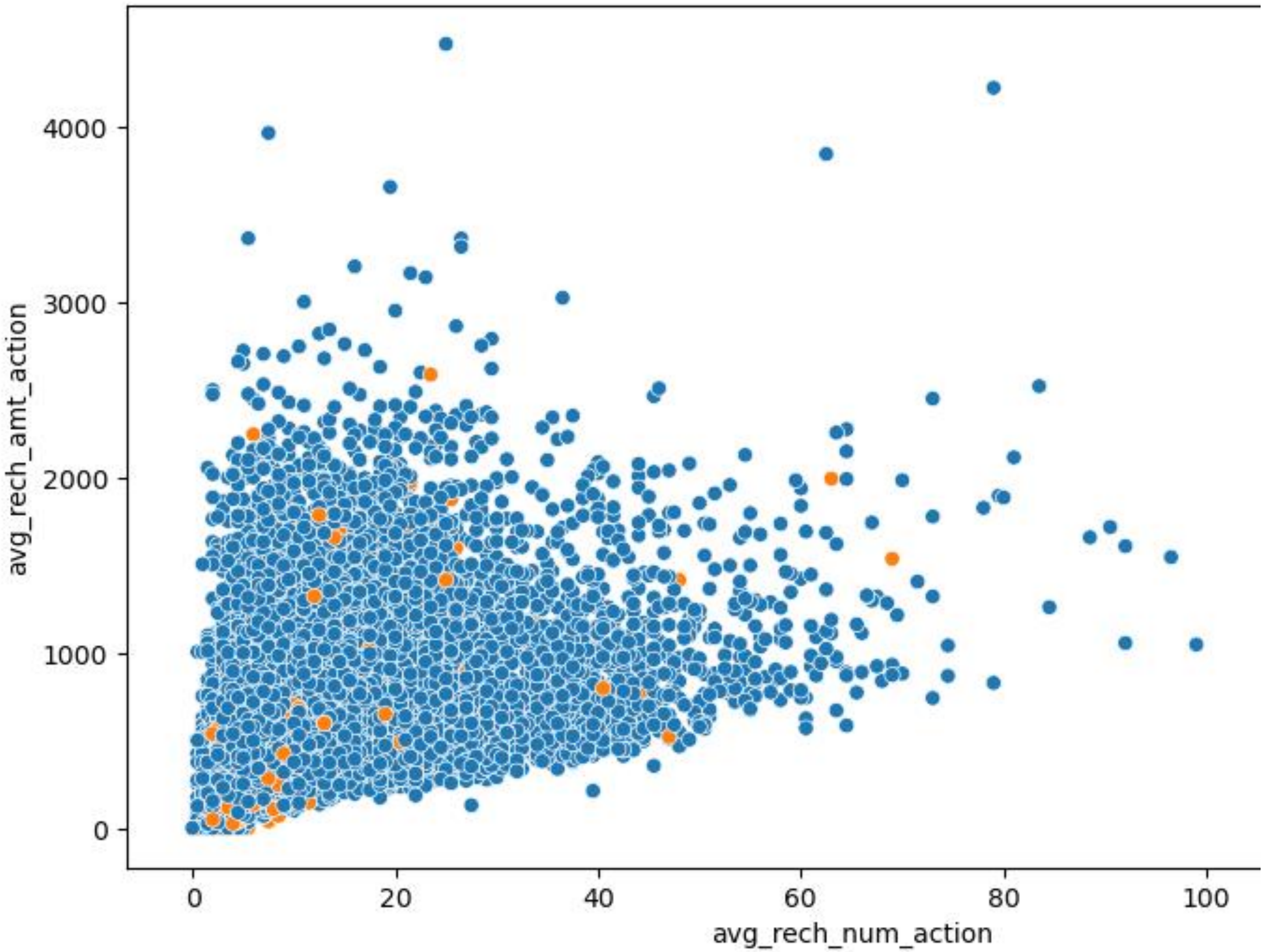
- Analysis of churn rate by the decreasing recharge amount and volume based cost in the action phase.

Insight: Customers who exhibit a decline in recharge amount coupled with an increase in volume-based costs during the action month are more likely to churn. This suggests that customers may be seeking more value for their spending and are dissatisfied with the current pricing structure.

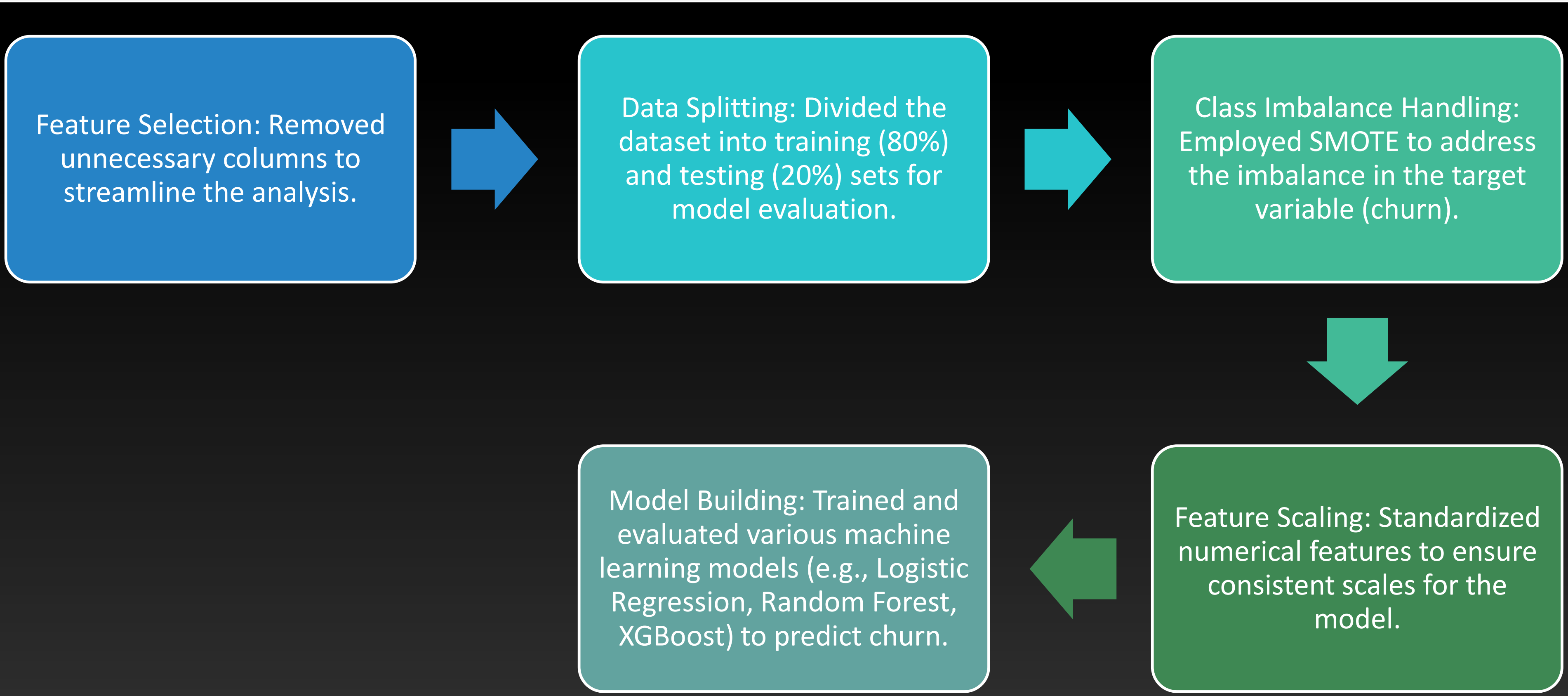
Bivariate analysis

- Analysis of recharge amount and number of recharge in action month.

Analysis: We can see that from above pattern recharge amount and recharge number are mostly proportional that means more the number of recharge numbers more the recharge amount.



Data Preprocessing and Model Building



Feature Selection: Removed unnecessary columns to streamline the analysis.

Data Splitting: Divided the dataset into training (80%) and testing (20%) sets for model evaluation.

Class Imbalance Handling: Employed SMOTE to address the imbalance in the target variable (churn).

Feature Scaling: Standardized numerical features to ensure consistent scales for the model.

Model Building: Trained and evaluated various machine learning models (e.g., Logistic Regression, Random Forest, XGBoost) to predict churn.

Libraries

Importing Libraries:

from sklearn.decomposition import PCA:

- Imports the **PCA** class from **scikit-learn for dimensionality reduction**.

from sklearn.linear_model import LogisticRegression:

- Imports the **Logistic Regression** class for **building a classification model.**

from sklearn.metrics import confusion_matrix, accuracy_score:

- Imports functions for **evaluating model performance**.

from sklearn.model_selection import KFold, GridSearchCV:

- Imports functions for **cross-validation and hyperparameter tuning**.

import pandas as pd:

- Imports **pandas** library for **data manipulation**.

import matplotlib.pyplot as plt:

- Imports **matplotlib** for plotting.

Data Modelling with Principal Component Analysis (PCA)

Purpose: Reduce dimensionality of the dataset to improve model performance and interpretability.

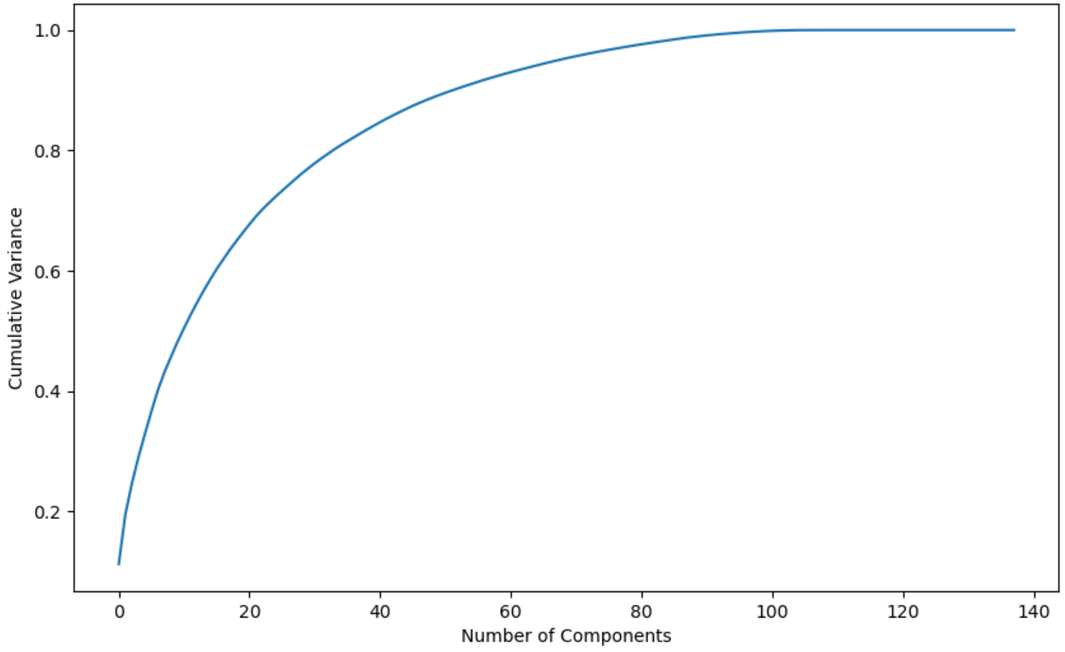
Benefits: Faster training, reduced computational cost, and potential for better generalization.

PCA Implementation

Import Libraries:

- from sklearn.decomposition import PCA
- from sklearn.linear_model import LogisticRegression
- from sklearn.metrics
- import confusion_matrix, accuracy_score
- from sklearn.model_selection import
- KFold, GridSearchCV
- import pandas as pd
- import matplotlib.pyplot as plt

Model with PCA



- Instantiate PCA: pca = PCA(random_state=42)

- Fit PCA to Training Data: pca.fit(X_train)

- Determine Number of Components: We can see that 60 components explain almost more than 90% variance of the data.

- we will perform PCA with 60 components.

- Applying transformation on the test set: We are only doing Transform in the test set not the Fit-Transform. Because the Fitting is already done on the train set. So, we just have to do the transformation with the already fitted data on the train set.

Sensitivity over Accuracy

Emphasize Sensitivity/Recall than Accuracy

- We are more focused on higher Sensitivity/Recall score than the accuracy.
- As we need to care more about churn cases than the not churn cases. The main goal is to retain the customers, who have the possibility to churn. There should not be a problem, if we consider few not churn customers as churn customers and provide them some incentives for retaining them. Hence, the sensitivity score is more important here.

Logistic regression with PCA

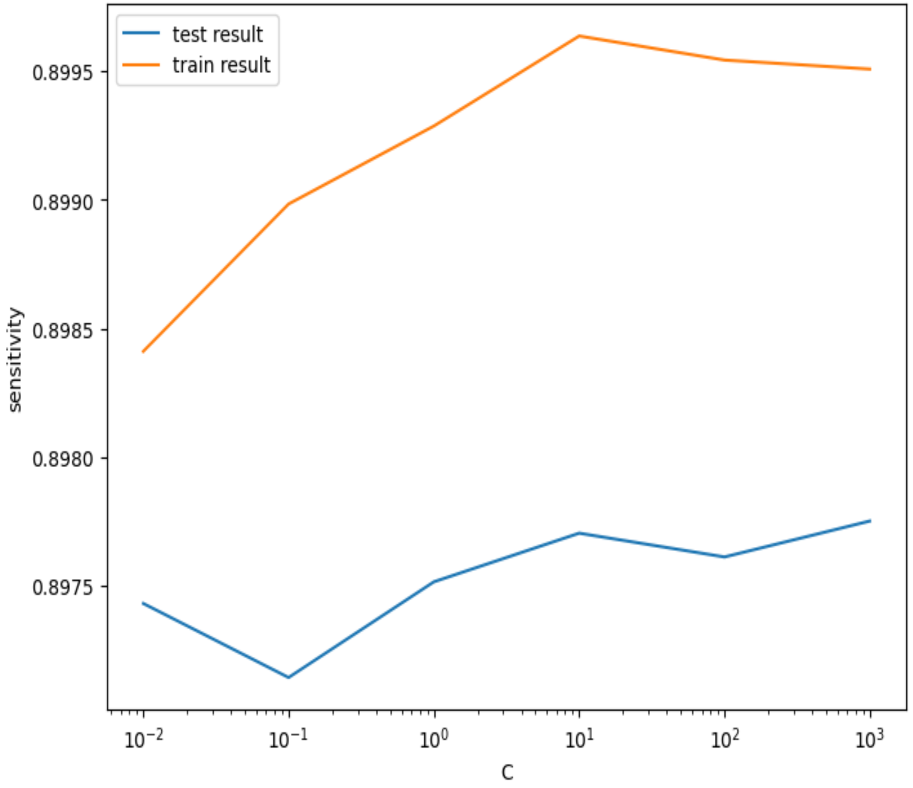
- **Hyperparameter Tuning**: Regularization Strength (C): Optimal value found: 1000

- **Model Performance :**

Train Set:	Test Set:
Accuracy: 0.86	Accuracy: 0.83
Sensitivity: 0.89	Sensitivity: 0.81
Specificity: 0.83	Specificity: 0.83

- **Conclusion** :

- The Logistic Regression model with PCA demonstrates good performance on both training and testing sets, indicating effective learning from the data. The model achieves a balanced accuracy, sensitivity, and specificity, suggesting accurate churn prediction.



Support Vector Machine(SVM) with PCA

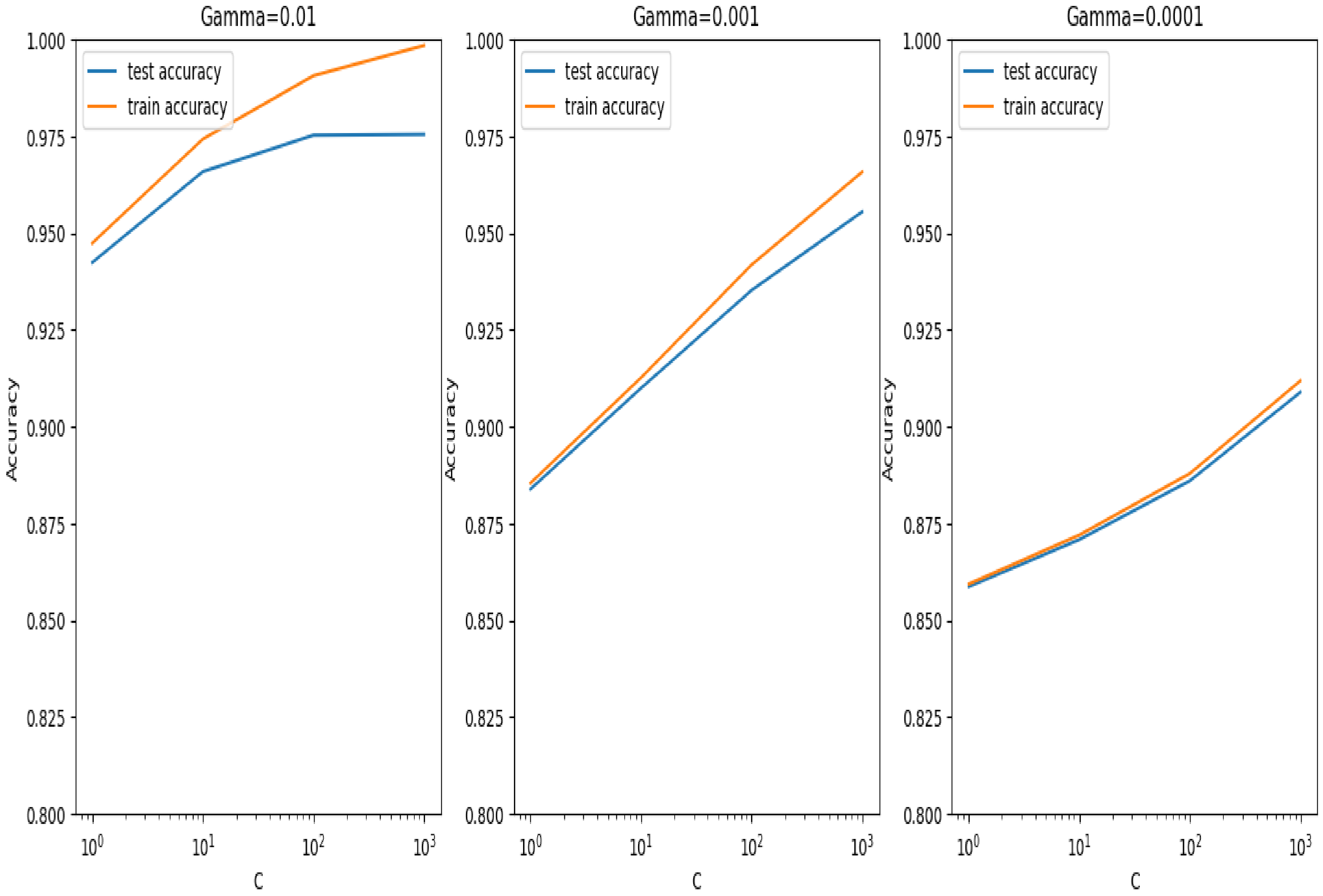
- **Hyperparameter Tuning:**
  - Kernel: Radial Basis Function (RBF) chosen for non-linearity.
  - GridSearchCV: Identified optimal hyperparameters:
    - C (regularization): 100 (controls model complexity)
    - gamma (non-linearity): 0.0001 (balances non-linearity and overfitting)

- **Model Performance:**

Train Set:	Test Set:
Accuracy: 0.89	Accuracy: 0.85
Sensitivity (Recall): 0.92 (good at identifying churn) (slightly lower than train set)	Sensitivity: 0.81
Specificity: 0.85 (low false positives) (consistent with train set).	Specificity: 0.85

Plotting the accuracy with various C and gamma values

The SVM with PCA achieves good performance on both datasets. Prioritized a simpler model with less non-linearity (gamma=0.0001) for better interpretability and potential generalization. This resulted in a slight trade-off between training and test set accuracy compared to the strictly optimal hyperparameters found by GridSearchCV



Decision tree with PCA

Hyperparameter Tuning:

- Best sensitivity: 0.9007234539089849
- Optimal hyperparameters:
  - max_depth: 10 min_samples_leaf: 50 min_samples_split: 50

Model Summary:

Train Set :	Test set :
Accuracy: 0.90	Accuracy = 0.86
Sensitivity: 0.91	Sensitivity = 0.70
Specificity: 0.88	Specificity = 0.87

Conclusion:

The Decision Tree model with tuned hyperparameters shows good overall performance, especially on the training set. However, there's a notable drop in sensitivity on the test set, suggesting potential overfitting. Further fine-tuning or exploring other models might be necessary to improve generalization.

Random forest with PCA

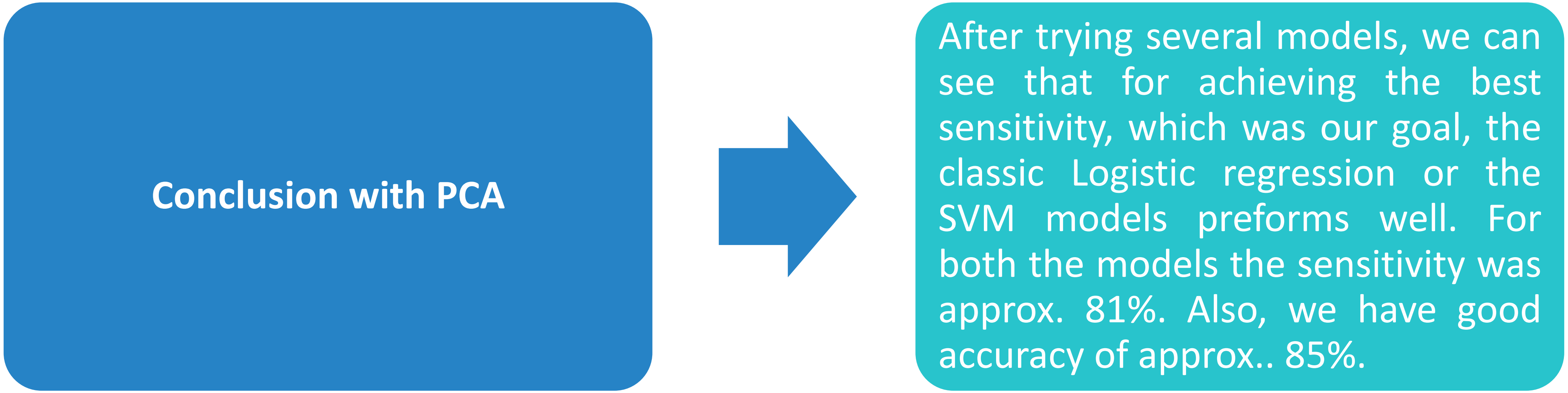
- **Hyperparameter tuning**

We can get accuracy of 0.8441073434308161 using {'max_depth': 5, 'max_features': 20, 'min_samples_leaf': 50, 'min_samples_split': 100, 'n_estimators': 300}

- Model with optimal hyperparameters
- **Model summary**

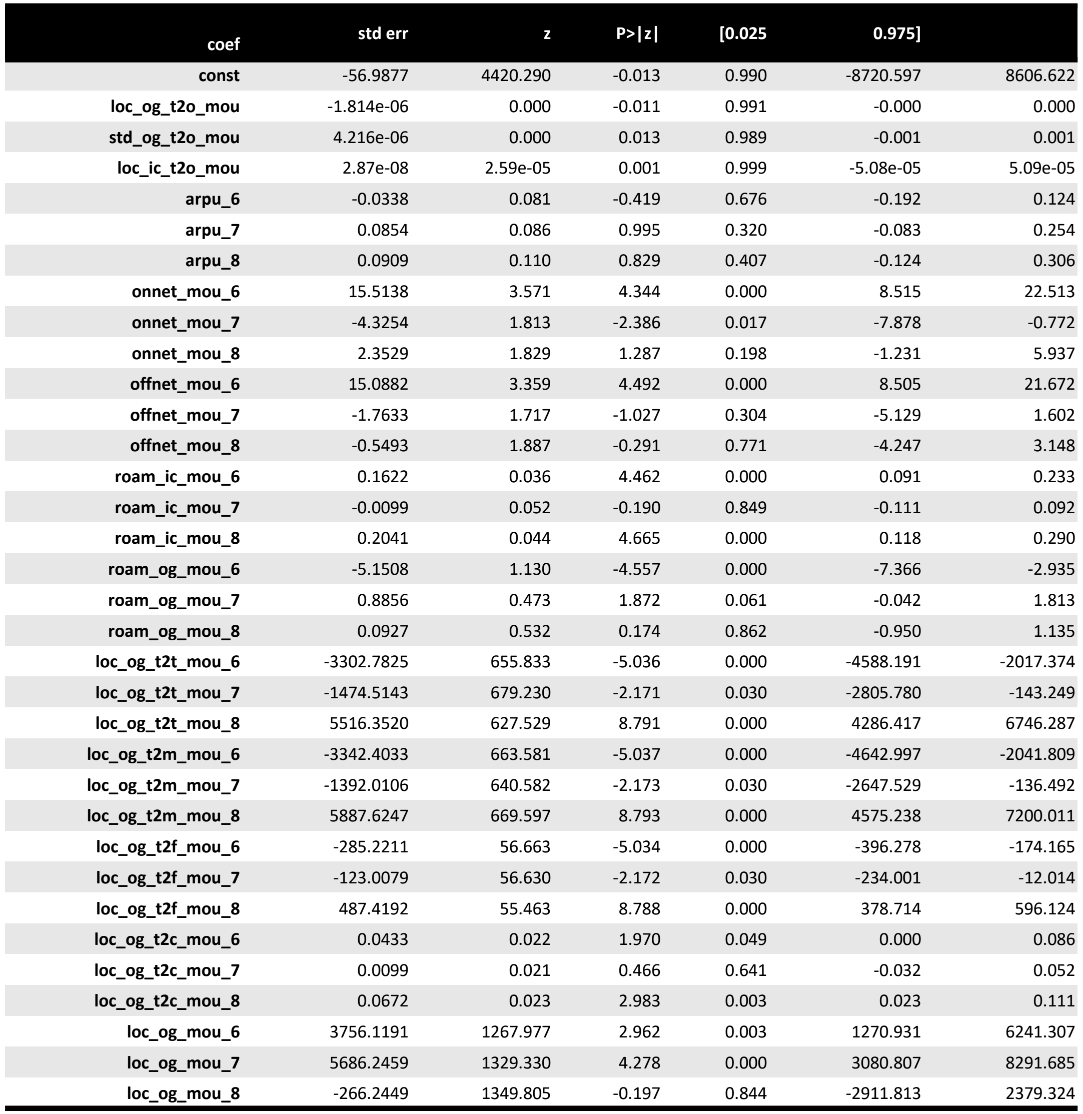
Train set	**Test set**
Accuracy = 0.84	Accuracy = 0.80
Sensitivity = 0.88	Sensitivity = 0.75
Specificity = 0.80	Specificity = 0.80

- **Conclusion** :
- We can see from the model performance that the Sensitivity has been decreased while evaluating the model on the test set. However, the accuracy and specificity is quite good in the test set.



Conclusion with PCA

After trying several models, we can see that for achieving the best sensitivity, which was our goal, the classic Logistic regression or the SVM models preforms well. For both the models the sensitivity was approx. 81%. Also, we have good accuracy of approx.. 85%.



coef	std err	z	P>|z|	[0.025	0.975]	
const	-56.9877	4420.290	-0.013	0.990	-8720.597	8606.622
loc_og_t2o_mou	-1.814e-06	0.000	-0.011	0.991	-0.000	0.000
std_og_t2o_mou	4.216e-06	0.000	0.013	0.989	-0.001	0.001
loc_ic_t2o_mou	2.87e-08	2.59e-05	0.001	0.999	-5.08e-05	5.09e-05
arpu_6	-0.0338	0.081	-0.419	0.676	-0.192	0.124
arpu_7	0.0854	0.086	0.995	0.320	-0.083	0.254
arpu_8	0.0909	0.110	0.829	0.407	-0.124	0.306
onnet_mou_6	15.5138	3.571	4.344	0.000	8.515	22.513
onnet_mou_7	-4.3254	1.813	-2.386	0.017	-7.878	-0.772
onnet_mou_8	2.3529	1.829	1.287	0.198	-1.231	5.937
offnet_mou_6	15.0882	3.359	4.492	0.000	8.505	21.672
offnet_mou_7	-1.7633	1.717	-1.027	0.304	-5.129	1.602
offnet_mou_8	-0.5493	1.887	-0.291	0.771	-4.247	3.148
roam_ic_mou_6	0.1622	0.036	4.462	0.000	0.091	0.233
roam_ic_mou_7	-0.0099	0.052	-0.190	0.849	-0.111	0.092
roam_ic_mou_8	0.2041	0.044	4.665	0.000	0.118	0.290
roam_og_mou_6	-5.1508	1.130	-4.557	0.000	-7.366	-2.935
roam_og_mou_7	0.8856	0.473	1.872	0.061	-0.042	1.813
roam_og_mou_8	0.0927	0.532	0.174	0.862	-0.950	1.135
loc_og_t2t_mou_6	-3302.7825	655.833	-5.036	0.000	-4588.191	-2017.374
loc_og_t2t_mou_7	-1474.5143	679.230	-2.171	0.030	-2805.780	-143.249
loc_og_t2t_mou_8	5516.3520	627.529	8.791	0.000	4286.417	6746.287
loc_og_t2m_mou_6	-3342.4033	663.581	-5.037	0.000	-4642.997	-2041.809
loc_og_t2m_mou_7	-1392.0106	640.582	-2.173	0.030	-2647.529	-136.492
loc_og_t2m_mou_8	5887.6247	669.597	8.793	0.000	4575.238	7200.011
loc_og_t2f_mou_6	-285.2211	56.663	-5.034	0.000	-396.278	-174.165
loc_og_t2f_mou_7	-123.0079	56.630	-2.172	0.030	-234.001	-12.014
loc_og_t2f_mou_8	487.4192	55.463	8.788	0.000	378.714	596.124
loc_og_t2c_mou_6	0.0433	0.022	1.970	0.049	0.000	0.086
loc_og_t2c_mou_7	0.0099	0.021	0.466	0.641	-0.032	0.052
loc_og_t2c_mou_8	0.0672	0.023	2.983	0.003	0.023	0.111
loc_og_mou_6	3756.1191	1267.977	2.962	0.003	1270.931	6241.307
loc_og_mou_7	5686.2459	1329.330	4.278	0.000	3080.807	8291.685
loc_og_mou_8	-266.2449	1349.805	-0.197	0.844	-2911.813	2379.324

Model Analysis and Feature Engineering

Without PCA Logistic regression with No PCA
Key Findings:

- Feature Coefficients: We observed that some features have positive coefficients, indicating they increase the likelihood of churn, while others have negative coefficients, suggesting they decrease the likelihood of churn.
- Feature Significance: Many features have high p-values and are therefore not statistically significant in the model.

Coarse Tuning:

- Recursive Feature Elimination (RFE): Eliminate less important features to reduce model complexity and improve interpretability.
- Manual Feature Elimination: Further refine feature selection based on p-values and Variance Inflation Factors (VIFs).

Feature Selection and Model Refinement

- Feature Selection using RFE
  - RFE with 15 Columns: Selected 15 features: offnet_mou_7, offnet_mou_8, roam_og_mou_8, std_og_t2m_mou_8, isd_og_mou_8, og_others_7, og_others_8, loc_ic_t2f_mou_8, loc_ic_mou_8, std_ic_t2f_mou_8, ic_others_8, total_rech_num_8, monthly_2g_8, monthly_3g_8, decrease_vbc_action
- Model Refinement
- Model 1 (with RFE-selected columns):
  - Checked VIFs for multicollinearity.
  - Identified og_others_8 as insignificant (p-value: 0.99).
- As we can see from the model summary that all the variables p-values are significant, and offnet_mou_8 column has the highest VIF 7.45. Hence, deleting offnet_mou_8 column

Checking VIF for Model-2

Features	VIF	
1	offnet_mou_8	7.45
3	std_og_t2m_mou_8	6.27
0	offnet_mou_7	1.92
7	loc_ic_mou_8	1.68
6	loc_ic_t2f_mou_8	1.21
10	total_rech_num_8	1.19
2	roam_og_mou_8	1.16
13	decrease_vbc_action	1.08
12	monthly_3g_8	1.06
11	monthly_2g_8	1.05
8	std_ic_t2f_mou_8	1.02
4	isd_og_mou_8	1.01
9	ic_others_8	1.01
5	og_others_7	1.00

•**Model 2** (without og_others_8):

- Rebuilt the model and checked VIFs again.
- Found offnet_mou_8 to have the highest VIF (7.45), indicating potential multicollinearity.

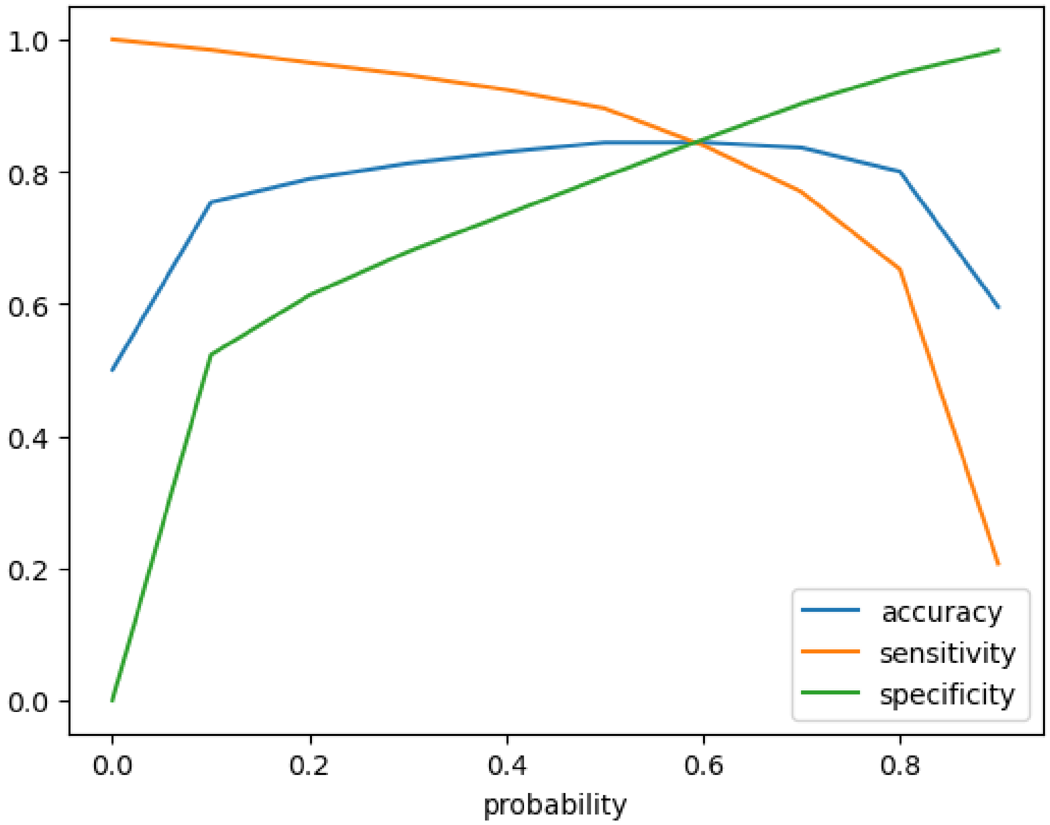
•**Model 3** (without og_others_8 and offnet_mou_8):

- Rebuilt the model and verified that all variables are significant and have low VIFs.

Final Model:

•**Model 3** (log_no_pca_3) is the final model, as it incorporates feature selection and addresses multicollinearity.

.

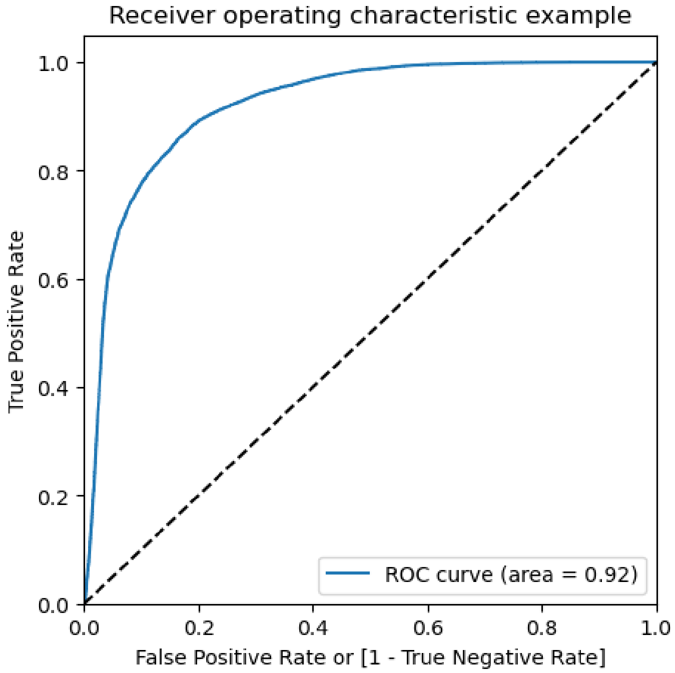


Model Performance and Threshold Selection

- Key Findings:
  - Accuracy: Stabilizes around 0.6.
  - Sensitivity: Decreases with higher thresholds (good for identifying churn).
  - Specificity: Increases with higher thresholds (good for minimizing false positives).
- Threshold Choice:
  - Prioritizing sensitivity, we choose a threshold of 0.5.
  - This captures more potential churn cases while maintaining a reasonable balance.
- Explanation: A lower threshold allows for earlier identification of at-risk customers.

Plotting the ROC Curve (Trade off between sensitivity & specificity)

- We can see the area of the ROC curve is closer to 1, which is the Gini of the model.
- Model summary
  - Train set Accuracy = 0.84 Sensitivity = 0.81 Specificity = 0.83
  - Test set Accuracy = 0.78 Sensitivity = 0.82 Specificity = 0.78
- Overall, the model is performing well in the test set, what it had learnt from the train set.
- Final conclusion : with no PCA We can see that the logistic model with no PCA has good sensitivity and accuracy, which are comparable to the models with PCA. So, we can go for the more simplistic model such as logistic regression with PCA as it explains the important predictor variables as well as the significance of each variable. The model also helps us to identify the variables which should be act upon for making the decision of the to be churned customers. Hence, the model is more relevant in terms of explaining to the business.



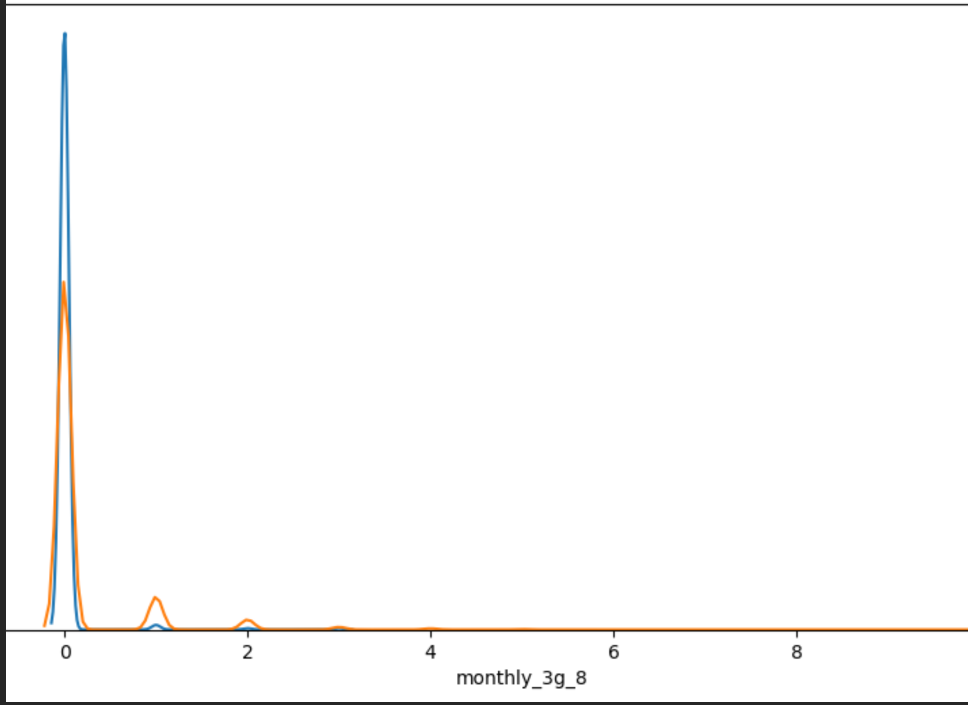
Analyzing Predictors for Churn and Non-Churn Customers

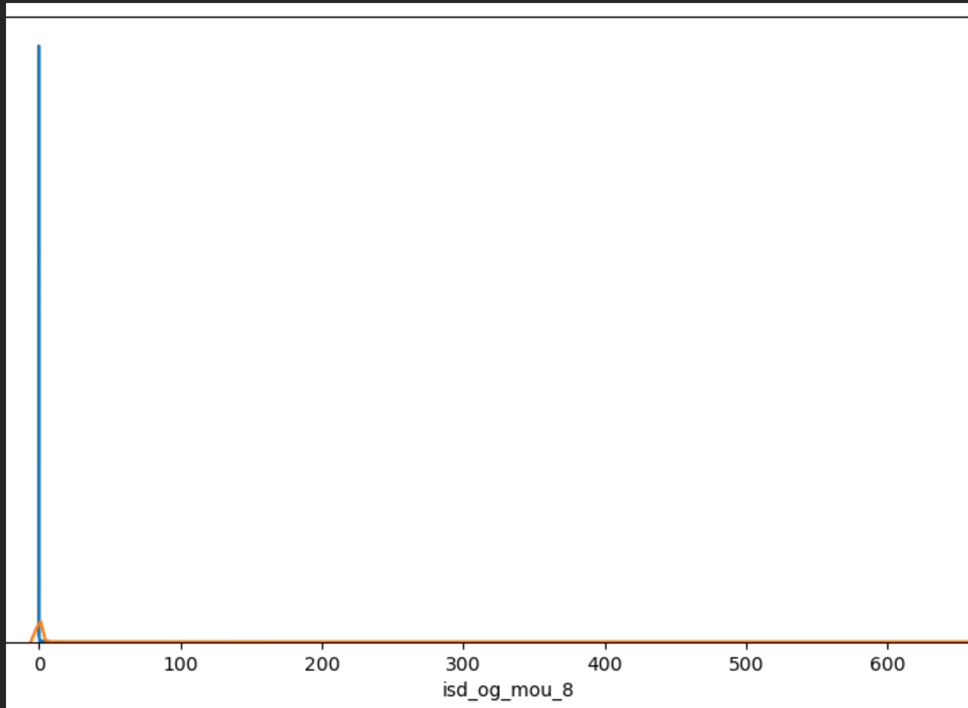
Monthly 3G Data Usage (August):

- Plot: A KDE plot showcasing the distribution of "monthly_3g_8" for churn and non-churn customers.
- Insights: Point out that churn customers have a more concentrated usage pattern for monthly 3G data in August, particularly around 1 unit. Non-churn customers have a wider distribution, suggesting varying data consumption habits.

ISD Outgoing Minutes of Usage (August):

- Plot: Another KDE plot depicting the distribution of "isd_og_mou_8" for both customer segments.
- Insights: Emphasize that churn customers have a higher concentration of usage near zero for ISD outgoing minutes in August compared to non-churn customers. This indicates less international calling activity for churned users.

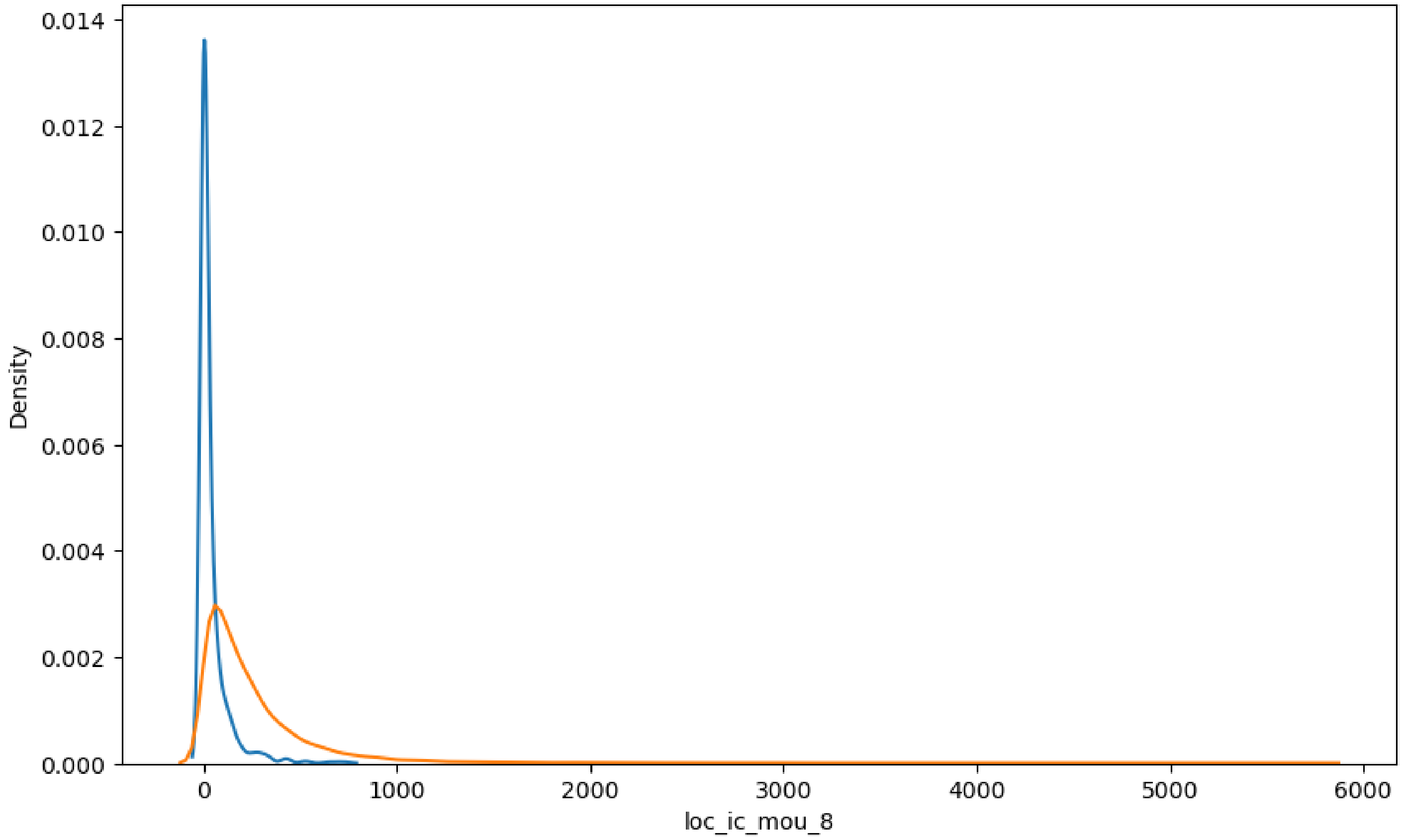




Local Intra-Circle Minutes of Usage (August)

Plot: Use a kernel density estimation (KDE) plot created with sns.distplot to visualize the distribution of "loc_ic_mou_8" for churn and non-churn customers (hist=False removes unnecessary bars).

Insights: Highlight that churn customers have lower usage of local intra-circle minutes in August compared to non-churn customers. This suggests lower engagement with the network.



Top predictors

Below are few top variables selected in the logistic regression model.

We can see most of the top variables have negative coefficients. That means, the variables are inversely correlated with the churn probability.

E.g.:-

If the local incoming minutes of usage (loc_ic_mou_8) is lesser in the month of August than any other month, then there is a higher chance that the customer is likely to churn.

Variables	Coefficients
loc_ic_mou_8	-3.3287
og_others_7	-2.4711
ic_others_8	-1.5131
isd_og_mou_8	-1.3811
decrease_vbc_action	-1.3293
monthly_3g_8	-1.0943
std_ic_t2f_mou_8	-0.9503
monthly_2g_8	-0.9279
loc_ic_t2f_mou_8	-0.7102
roam_og_mou_8	0.7135

Business Recommendations

1) Target the customers, whose minutes of usage of the incoming local calls and outgoing ISD calls are less in the action phase (mostly in the month of August).

2) Target the customers, whose outgoing others charge in July and incoming others on August are less.

3) Also, the customers having value-based cost in the action phase increased are more likely to churn than the other customers. Hence, these customers may be a good target to provide offer.

4) Customers, monthly 3G recharge in August is more, are likely to be churned.

5) Customers having decreasing STD incoming minutes of usage for operators T to fixed lines of T for the month of August are more likely to churn.

6) Customers decreasing monthly 2g usage for August are most probable to churn.

7) Customers having decreasing incoming minutes of usage for operators T to fixed lines of T for August are more likely to churn. roam_og_mou_8 variables have positive coefficients (0.7135). That means for the customers, whose roaming outgoing minutes of usage is increasing are more likely to churn.

Business Implications & Future Directions :

Based on the analysis, here are some recommendations to reduce customer churn:

Focus on customers with declining recharge amounts and usage.

Develop targeted retention campaigns for customers showing early signs of churn.

Investigate reasons for decreased usage and address those issues.

Consider offering personalized plans or incentives to high-value customers.

Regularly monitor and analyze customer usage patterns to identify potential churners early.