CAPSTONE PROJECT CUSTOMER CHURN (NOTE-2)

BY – *Harsh Patel*

22th December 2024

[NOTE: In the project Note-1 I got my marks deducted due pages more than 70. I have tried to compress information and kept it less 40 pages]

1. MODEL APPROACH:

1.1. Applicable Algorithms:

- **Binary Classification**: This business case requires predicting customer churn, a binary classification problem with outcomes '0' (will not churn) and '1' (will churn).
- **Supervised Learning**: The target variable 'Churn' needs prediction, making this a supervised learning problem.
- **Algorithms for Classification**:
  - **Linear Classification**: Logistic Regression, Linear Discriminant Analysis, Naïve Bayes.
  - **Non-linear Classification**: SVMs (non-linear adaptations), Decision Tree, K-Nearest Neighbor (KNN).
  - **Ensemble Models**: Random Forest, AdaBoost, Gradient Boost.
- **Algorithm Assumptions**: Algorithms have specific assumptions about the data, influencing their performance.

1.2. Methodology:

- **Data Preprocessing Variants**: Different treatments of pre-processed data were prepared, including scenarios with and without SMOTE and hypertuned datasets.
- **Variance Inflation Factor (VIF)**: Calculated for predictor variables. Predictors with VIF greater than 5 were identified. 'Cashback', 'Service Score', 'Clusters', and 'User Count' had high VIF. 'User Count' was retained due to significant Chi-square value from EDA; the other variables were dropped.
- **Variable Selection**: 'rev_growth_yoy' and 'coupon_used_for_payment' were dropped after ANOVA and Chi-square tests, and verification against EDA plots.
- **Data Scaling**: Scaled data used for distance-based algorithms like KNN and ANN. SMOTE resampled data also tested.
- **Data Splitting**: Data split into train (70%) and test (30%) sets, with similar distribution of target variable as the original dataset (16.8% churn).
- **Algorithm Testing**: Eight algorithms chosen. For each:
  - Constructed base model with default hyperparameters and evaluated on train and test datasets.
  - Used different data treatments and recorded performance.
  - Tuned hyperparameters using GridSearchCV from Sklearn and manual adjustments.
  - Measured performance of tuned algorithms.

  - Extracted feature importance from models using built-in attributes or Sklearn's Permutation feature importance for black-box models.

1.3. Model Evaluation Metrics:

- **Comparison Criteria**: Best model selected based on evaluation metrics for train and test data, ensuring no overfitting or underfitting.
- **Precision**: High precision for churn customers to minimize unnecessary freebies by the revenue assurance team.
  - Precision = True Positive / (True Positive + False Positive)
- **F1-Score**: Ensures high recall alongside precision. The model should effectively predict actual churns to address the churn problem.
- **Model Interpretability**: Preference for interpretable models.
- **Computational Efficiency**: Avoid computationally expensive models like KNN.

1.4. Model Building and Tuning:

- **Algorithms Tested**: Logistic Regression, Linear Discriminant Analysis, SVM,, KNN, Random Forest, AdaBoost, Gradient Boost using Sklearn implementations. Logistic Regression also executed using the Statsmodel package.
- **Tabulated Results**: Detailed results for all algorithms, including tuning efforts and performance. For brevity, top-performing models and their feature importances are summarized.
- **Effort for Model Tuning**: Performance improvement achieved through:
  - Testing eight different algorithms across various methods.
  - Constructing base models with default hyperparameters and tuning using GridSearchCV.
  - Changing underlying data (SMOTE resampled/non-resampled) and observing performance effects.
  - Using ensemble methods (Random Forest, AdaBoost, Gradient Boost) and tuning their hyperparameters.

2. MODEL Building:

Train-Test Split: A Crucial Step:

Purpose: To create a robust predictive model that performs well on unseen data.

Process: Data is divided into two sets:

- **Training Set (70%)**: This dataset teaches the model the underlying patterns.
- **Test Set (30%)**: This set evaluates the model's performance on new data.

Data Breakdown

- **X_train (7882, 17)**: Training dataset with 7,882 rows and 17 features, used to teach the model patterns.
- **X_test (3378, 17)**: Test dataset with 3,378 rows and 17 features, unseen by the model, for evaluating its performance.
- **y_train (7882,)**: Target variable for the training samples, guiding the model's learning process.
- **y_test (3378,)**: Target variable for the test samples, used to measure the model's predictions against actual outcomes on unseen data.

2.1 Performance Various Models:

To evaluate the effectiveness of various classification models, we conducted a comprehensive analysis using **classification reports** and **Area Under the Curve (AUC) scores**. These metrics provide valuable insights into the models' abilities to distinguish between different classes and their overall predictive power.

Evaluation Metrics:

- **Precision**: Measures the accuracy of positive predictions (churn). High precision indicates fewer false positives.
- **Recall**: Measures the ability to identify all positive instances (churn). High recall indicates fewer false negatives.
- **F1-Score**: The harmonic mean of precision and recall, providing a balance between the two. High F1-score ensures that both precision and recall are optimized.
- **AUC Score**: Measures the overall performance of the classification model, indicating the model's ability to distinguish between classes. A higher AUC score signifies better model performance.

Analysis Insights:

- **Precision, Recall, F1-Scores, and AUC Scores**: By meticulously analyzing these metrics, we can clearly understand each model's strengths and weaknesses. This assessment helps in making informed decisions about which models are best suited for addressing the specific business problem at hand.

Prediction Labels:

- **"0"**: Indicates a prediction of non-churn.
- **"1"**: Indicates a prediction of churn.

By leveraging these metrics, we can ensure that our chosen model not only accurately predicts churn but also aligns with the business objectives, minimizing false positives and negatives and effectively addressing customer churn

Models	Training Dataset (70%)								Testing Dataset (30)%							
Pefromance Metrics	Precision (0)	Recall (0)	F1-score (0)	Precision (1)	Recall (1)	F1-score (1)	Accuracy	AUC Score	Precision (0)	Recall (0)	F1-score (0)	Precision (1)	Recall (1)	F1-score (1)	Accuracy	AUC Score
Logistic Regression (Basic)	0.9	0.96	0.93	0.73	0.48	0.58	0.88	0.88	0.91	0.96	0.93	0.74	0.51	0.61	0.89	0.87
Logistic Regression (with hyper-tuning)	0.9	0.97	0.93	0.74	0.48	0.58	0.89	0.88	0.91	0.97	0.94	0.75	0.51	0.61	0.89	0.87
Logistic Regression (SMOTE)	0.83	0.82	0.83	0.82	0.84	0.83	0.83	0.89	0.95	0.83	0.89	0.49	0.8	0.6	0.82	0.87
LDA (Basic)	0.91	0.96	0.93	0.71	0.53	0.61	0.88	0.88	0.91	0.96	0.93	0.72	0.56	0.63	0.89	0.87
LDA (with Hyper-tuning)	0.91	0.96	0.93	0.71	0.53	0.61	0.88	0.88	0.91	0.96	0.93	0.71	0.55	0.62	0.89	0.87
LDA (SMOTE)	0.84	0.82	0.83	0.82	0.84	0.83	0.83	0.89	0.95	0.82	0.88	0.48	0.8	0.6	0.82	0.87
KNN Model (Basic)	0.97	0.99	0.98	0.95	0.87	0.91	0.97	0.99	0.95	0.98	0.97	0.9	0.74	0.81	0.94	0.96
KNN Model (with Hyper-tuning)	1	1	1	1	1	1	1	1	0.97	0.99	0.98	0.92	0.82	0.87	0.96	0.96
KNN Model (SMOTE)	1	0.95	0.97	0.95	1	0.97	0.97	1	0.99	0.93	0.95	0.72	0.93	0.81	0.93	0.97
Naïve Bayes Model (Basic)	0.91	0.92	0.92	0.59	0.55	0.57	0.86	0.84	0.91	0.92	0.91	0.58	0.54	0.56	0.86	0.83
Naïve Bayes Model (with Hyper-tuning)	0.91	0.92	0.92	0.59	0.55	0.57	0.86	0.84	0.91	0.92	0.91	0.58	0.54	0.56	0.86	0.83
Naïve Bayes Model (SMOTE)	0.79	0.78	0.79	0.78	0.79	0.79	0.79	0.84	0.94	0.78	0.85	0.41	0.74	0.53	0.78	0.83
RandomForestClassifier (Basic)	1	1	1	1	1	1	1	1	0.96	0.99	0.98	0.95	0.81	0.87	0.96	0.98
RandomForestClassifier (with Hyper-tuning)	0.95	1	0.97	0.98	0.75	0.85	0.96	0.99	0.93	0.99	0.96	0.92	0.65	0.76	0.93	0.96
RandomForestClassifier (SMOTE)	1	1	1	1	1	1	1	1	0.97	0.98	0.98	0.9	0.85	0.88	0.96	0.98
Bagging (Basic)	1	1	1	1	0.98	0.99	1	0.99	0.95	0.99	0.97	0.92	0.76	0.84	0.95	0.88
Bagging (with Hyper Tuning)	1	1	1	1	0.98	0.99	1	0.99	0.94	0.99	0.97	0.95	0.68	0.8	0.94	0.84
Bagging (SMOTE)	1	1	1	1	0.99	0.99	1	0.99	0.96	0.97	0.97	0.86	0.82	0.84	0.95	0.9
Ada Boost (Basic)	0.91	0.96	0.93	0.72	0.54	0.62	0.89	0.91	0.91	0.96	0.93	0.73	0.54	0.62	0.89	0.9
Ada Boost (with Hyper Tuning)	0.91	0.96	0.94	0.73	0.55	0.63	0.89	0.91	0.91	0.96	0.94	0.74	0.55	0.63	0.89	0.9
Ada Boost (SMOTE)	0.88	0.88	0.88	0.88	0.88	0.88	0.88	0.95	0.94	0.89	0.92	0.58	0.74	0.65	0.87	0.89
Gradient Boosting (Basic)	0.92	0.97	0.95	0.81	0.6	0.69	0.91	0.94	0.92	0.97	0.94	0.79	0.58	0.67	0.9	0.92
Gradient Boosting (with Hyper-tuning)	1	1	1	1	0.98	0.99	1	1	0.96	0.99	0.97	0.93	0.79	0.85	0.95	0.98
Gradient Boosting (SMOTE)	0.92	0.92	0.92	0.92	0.92	0.92	0.92	0.97	0.94	0.93	0.94	0.67	0.71	0.69	0.89	0.91
Support vector Machine (Basic)	0.92	0.98	0.95	0.86	0.6	0.71	0.92	0.94	0.92	0.98	0.95	0.85	0.58	0.69	0.91	0.91
Support vector Machine (with Hyper-tuning)	0.96	0.99	0.98	0.96	0.81	0.88	0.96	0.99	0.94	0.98	0.96	0.9	0.71	0.79	0.94	0.95
Support vector Machine (SMOTE)	0.94	0.9	0.92	0.91	0.94	0.92	0.92	0.97	0.96	0.9	0.93	0.62	0.83	0.71	0.88	0.94

Table -1: Classification Report and AUC score

2.2 Confusion Matrix of all Models for Comparison:

Components:

- True Positives (TP): Correctly predicted positive cases (e.g., actual churn predicted as churn).
- True Negatives (TN): Correctly predicted negative cases (e.g., actual non-churn predicted as non-churn).
- False Positives (FP): Incorrectly predicted positive cases (e.g., actual non-churn predicted as churn).
- False Negatives (FN): Incorrectly predicted negative cases (e.g., actual churn predicted as non-churn).

Benefits of Using a Confusion Matrix

- Detailed Performance Analysis: By interpreting these values, we gain insights into the strengths and weaknesses of each model.
- Identifying Patterns: It helps in identifying patterns of correct and incorrect predictions, highlighting areas where models are performing well and where they might be struggling.
- Quantifying Effectiveness: We can quantify the effectiveness of different models in terms of their ability to accurately predict outcomes.
- Spotting Areas for Improvement: The matrix reveals potential areas of improvement, guiding us in refining the models to enhance their performance.

Informed Decision-Making

- Model Selection: This thorough analysis assists us in making informed decisions about which models are best suited for our specific business objectives.
- Model Refinement: It also provides a clear direction for further tuning and refining the models to achieve better predictive accuracy.

Models	Training Dataset (70%)				Testing Dataset (30%)			
Confusion Matrix - Metrics	True Negative (TN)	Type II Error (False Negative)	Type I Error (False Positive)	True Positive (TP)	True Negative (TN)	Type II Error (False Negative)	Type I Error (False Positive)	True Positive (TP)
Logistic Regression (Basic)	6324	232	686	640	2705	103	278	292
Logistic Regression (with hyper-tuning)	6336	220	690	636	2712	96	281	289
Logistic Regression (SMOTE)	5366	1190	1081	5475	2325	483	114	456
LDA (Basic)	6270	286	626	700	2682	126	252	318
LDA (with Hyper-tuning)	6270	286	625	701	2682	126	254	316
LDA (SMOTE)	5383	1173	1058	5498	2307	501	113	457
KNN Model (Basic)	6494	62	171	1155	2764	44	151	419
KNN Model (with Hyper-tuning)	6556	0	0	1326	2769	39	100	470
KNN Model (SMOTE)	6214	342	8	6548	2601	207	39	531
Naive Bayes Model (Basic)	6048	508	596	730	2586	222	260	310
Naive Bayes Model (with Hyper-tuning)	6048	508	596	730	2586	222	260	310
Naive Bayes Model (SMOTE)	5133	1423	1380	5176	2198	610	146	424
RandomForestClassifier (Basic)	6556	0	0	1326	2786	22	119	451
RandomForestClassifier (with Hyper-tuning)	6538	18	331	995	2774	34	201	369
RandomForestClassifier (SMOTE)	6556	0	0	6556	2757	51	86	484
Bagging (Basic)	6554	2	29	1297	2765	43	133	437
Bagging (with Hyper Tuning)	6556	0	20	1306	2789	19	181	389
Bagging (SMOTE)	6549	7	12	1314	2720	88	109	461
Ada Boost (Basic)	6285	271	612	714	2694	114	265	305
Ada Boost (with Hyper Tuning)	6291	265	601	725	2699	109	255	315
Ada Boost (SMOTE)	5766	790	810	5746	2502	306	150	420
Gradient Boosting (Basic)	6368	188	531	795	2722	86	238	332
Gradient Boosting (with Hyper-tuning)	6555	1	22	1304	2776	32	121	449
Gradient Boosting (SMOTE)	6060	496	543	6013	2612	196	165	405
Support vector Machine (Basic)	6428	128	525	801	2749	59	241	329
Support vector Machine (with Hyper-tuning)	6510	46	249	1077	2761	47	166	404
Support vector Machine (SMOTE)	5925	631	408	6148	2514	294	99	471

Table -2: Confusion Matrix Analysis of all models

2.3 Interpretation of performance of metrics of Various model:

1. Interpretation of the Logistic Regression Model: Basic

The model's recall for predicting churn (class 1) stands at 48% during training and improves slightly to 51% in the test data, indicating how well it identifies churn cases. Precision, which shows the accuracy of positive predictions, is 73% in training and 74% in testing, indicating good alignment between predicted and actual churn. The F1-score, which balances precision and recall, is 0.58 for training and 0.61 for testing, reflecting a fair overall balance.

Analyzing confusion matrices reveals the presence of false negatives and false positives, pointing out areas needing improvement. During training, 686 actual churners were incorrectly predicted as non-churners (false negatives), while 232 were wrongly predicted to churn (false positives). Similarly, testing showed 278 false negatives and 103 false positives.

AUC values are 0.88 (training) and 0.87 (testing), confirming the model's capability to identify potential churners. The model highlights three significant factors: recent complaints, tenure, and single marital status, which impact customer outcomes.

2. Interpretation of the Logistic Regression Model: SMOTE

The Logistic Regression Model with SMOTE demonstrates balanced performance on both training and test data. In the training set, precision and recall are well-matched for both classes, resulting in an 83% F1-score and 83% accuracy. Similarly, the test set shows high precision (95%) and recall (83%) for class 0, yielding an 89% F1-score. Class 1 in the test set has 49% precision, 80% recall, and a 60% F1-score, with an overall test accuracy of 82%.

Confusion matrices highlight accurate predictions for true positives and true negatives, but also misclassifications. Training data misclassified 1081 non-churners as churners (false positives) and 1190 churners as non-churners (false negatives). In the test data, 114 non-churners are falsely identified as churners (false positives), and 483 churners as non-churners (false negatives).

AUC scores are 0.89 for the test and 0.87 for training, showcasing the model's ability to identify potential churners. The model emphasizes the importance of recent complaints, tenure, single marital status, and monthly revenue in influencing customer outcomes.

In summary, the Logistic Regression Model with SMOTE achieves balanced performance, though slightly lower precision for class 1 predictions in the test data suggests room for improvement.

3. Interpretation of Linear Discriminant Analysis: Basic

The LDA model's performance is demonstrated in the classification reports for both the training and test data. For customers who churned (class 1), the model has a precision of 71% in training and 72% in testing, indicating accurate positive predictions. The recall for class 1 is 53% in training and 56% in testing, with F1-scores of 0.61 and 0.63, respectively. For customers who did not churn (class 0), the model's precision and recall are both 91% and 96% for training and testing, resulting in F1-scores of 0.93.

The confusion matrices reveal misclassification instances of churners (629 false negatives) and non-churners (286 false positives) during training. In testing, there are 252 false negatives and 126 false positives.

AUC scores of 0.88 for training and 0.87 for testing validate the model's effectiveness in identifying potential churners. Key factors influencing outcomes include customer complaints, tenure, single marital status, and monthly revenue.

In conclusion, the LDA model performs fairly well in identifying potential churners and retaining customers, though there remains room for improvement.

4. Interpretation of Linear Discriminant Analysis: SMOTE

The LDA model with SMOTE's performance is shown in the classification reports for both training and test data. For churned customers (class 1), the model achieves a precision of 82% in training and 48% in testing, with recall at 84% and 80%, respectively, resulting in F1-scores of 0.83 and 0.60. For non-churned customers (class 0), the model's precision and recall are both 84% in training and testing, yielding F1-scores of 0.83.

The confusion matrices show misclassifications of 1,058 churners and 1,173 non-churners during training, and 113 churners and 501 non-churners during testing. AUC scores of 0.89 for training and 0.87 for testing highlight the model's capability to identify potential churners. Key features influencing outcomes include customer tenure and complaints.

In summary, the model effectively distinguishes between churn and non-churn cases. While it accurately captures both customer categories, there is still room for improvement in more precisely identifying potential churners.

5. Interpretation of KNN Model: Basic

The KNN model demonstrates strong performance in identifying customers who stay (class 0) with high precision (97%) and a satisfactory recall (87%) for potential churners (class 1) during training. The impressive overall accuracy of 97% and a weighted F1-score of 0.97 highlight its balance between precision and recall. On the test data, the model retains high precision (95%) for class 0 and a respectable recall (74%) for class 1, resulting in a robust overall accuracy of 94% and a weighted F1-score of 0.94.

Examining the confusion matrices, the model misclassified 171 churners as non-churners (false negatives) and 62 non-churners as churners (false positives) in the training data. In the test data, it misclassified 151 churners and 44 non-churners. AUC scores of 0.99 (training) and 0.96 (testing) underscore its ability to distinguish between classes.

In summary, the KNN model performs effectively in identifying staying customers and detecting potential churners. While the recall for class 1 could be improved, the model's balanced precision and recall, along with strong AUC scores, indicate solid overall performance.

6. Interpretation of KNN Model: SMOTE

The model's performance evaluation on both the training and test data reveals exceptional precision for both classes. For class 0 (customers who stay), precision is flawless at 100%, whereas for class 1 (potential churners), it is high at 95% in training and 72% in testing.

Despite the impressive precision, the recall for class 1 in the test data is slightly lower at 93%, suggesting room for improvement in identifying potential churners. The model upholds strong accuracy of 97% in training and 93% in testing, accompanied by well-balanced weighted F1-scores.

Analyzing the confusion matrices, there are a few misclassification cases, with slightly more class 0 predictions being mistaken for class 1. AUC scores of 1.00 (training) and 0.97 (testing) underscore the model's capability to distinguish between classes.

In summary, the model exhibits remarkable precision and overall performance. Although the recall for class 1 in the test data is slightly lower, the model remains effective and balanced, showing no signs of overfitting or underfitting.

7. Interpretation of the Naïve Bayes Model: Basic

The model's evaluation on both training and test data reveals a balanced approach to precision and recall for different classes. For class 0 (customers who stay), precision is strong at 91% for both training and testing, indicating accurate identification of non-churn cases. However, precision for class 1 (potential churners) drops to 59% in training and 58% in testing, with significant false positive predictions. Recall for class 1 stands at 55% in training and 54% in testing, indicating room for improvement in identifying customers who might leave.

Overall accuracy remains consistent at 86% for both training and testing. Weighted F1-scores blend precision and recall effectively. Confusion matrices highlight misclassifications between classes. In training, 508 class 0 cases are incorrectly predicted as class 1, and 596 class 1 cases are predicted as class 0. In testing, 222 class 0 cases are mistakenly predicted as class 1, and 260 class 1 cases as class 0. AUC scores of 0.84 for training and 0.83 for testing reflect the model's ability to distinguish between classes, but there is room for improvement.

In summary, the Naïve Bayes Model shows balanced precision and recall, but there is scope to enhance class 1 precision. The model's accuracy is fair, with potential to better capture actual churn cases. AUC scores suggest moderate discriminatory ability without clear signs of overfitting or underfitting. The model's performance underscores the importance of improving its ability to classify churn cases accurately while minimizing false positives.

8. Interpretation of Naïve Bayes Model: SMOTE

The classification reports evaluate the model’s performance on both the training and test datasets. For class 0 (customers who remain), precision is stable at 79% in training and higher at 94% in testing, showing accurate identification. For class 1 (potential churners), precision is balanced at 78% in training and 41% in testing. Recall, which measures the ability to capture actual class 1 cases, is about 79% in training and 74% in testing, suggesting room for improvement. Overall accuracy is 79% for training and 78% for testing. Weighted F1-scores represent a balance between precision and recall.

The confusion matrices reveal specific misclassifications. During training, 1,423 class 0 cases were incorrectly predicted as class 1, and 1,380 class 1 cases were predicted as class 0. In testing, 610 class 0 cases were misclassified as class 1, and 146 class 1 cases as class 0. AUC scores of 0.84 for training and 0.83 for testing reflect the model's ability to distinguish between classes.

In summary, the model consistently identifies customers who stay (class 0), with higher precision in testing. Enhancements are needed for better recognition of potential churners (class 1) and a balanced precision-recall. The model's accuracy is reasonable but requires further regularization to improve its performance in capturing churn cases more accurately.

9. Interpretation of the Random Forest Model: Basic

The evaluation of the Random Forest Model's performance on the training and test datasets reveals exceptional precision scores for both classes. For class 0 (customers who stay), precision remains consistently high at 100% during training and 96% during testing. Similarly, for class 1 (potential churners), the precision scores are impressive, reaching 100% during training and 95% during testing. However, recall values for class 1 are slightly lower, measuring 76% in testing and a perfect 100% in training, indicating room for improvement in identifying all potential churners.

The model exhibits robust overall accuracy, achieving 100% on training and 96% on testing. The weighted F1-scores effectively balance precision and recall. AUC scores of 1.00 for testing and 0.99 for training underscore the model's effectiveness.

The confusion matrices accurately distinguish between classes but with a few errors in churn predictions: 22 class 0 instances mislabelled as class 1 and 119 class 1 instances misidentified as class 0.

In summary, the Random Forest Model demonstrates remarkable precision for both classes, with potential to enhance class 1 recall. The model's accuracy, AUC scores, and key features (Tenure, Cashback) contribute to its strong performance.

10. Interpretation of the Random Forest Model: SMOTE

The Random Forest Model with SMOTE excels in both precision and recall for both classes. During training, it achieves 100% precision, recall, and F1-scores for both classes, raising concerns about overfitting due to these perfect scores. For testing, precision remains high at 97% for class 0 and 90% for class 1. Class 1 recall slightly decreases to 85%, still effectively capturing actual churn cases. The test accuracy is an impressive 96%.

The confusion matrices show perfect classification for training data, while the test data has minor misclassifications: 51 class 0 instances predicted as class 1 and 86 class 1 instances as class 0. AUC scores are 1.00 for testing and 0.98 for training, highlighting strong discriminatory ability. Key features such as Tenure and Complaints play pivotal roles.

In summary, the Random Forest Model with SMOTE demonstrates remarkable predictive capabilities, although the perfect scores on training data suggest potential overfitting.

11. Interpretation of the Support Vector Machines – SVM Model: Basic

The SVM Model demonstrates balanced performance on both training and test datasets. For class 0, it maintains strong precision and recall at 92% and 98%, respectively, for both datasets. However, class 1's precision (86%) and recall (60%) indicate some missed churn cases. The overall accuracy is solid at 92%.

On the test set, the model maintains robust class 0 precision (92%) and recall (98%). Class 1's precision (85%) and recall (58%) show room for improvement. The test accuracy is 91%.

In training, the model correctly classifies 801 class 1 instances but misclassifies 525 class 0 instances as class 1 and 128 class 1 instances as class 0. During testing, it correctly identifies 329 class 1 cases but misclassifies 241 class 0 cases as class 1 and 59 class 1 cases as class 0. The SVM model's AUC scores are 0.94 for training and 0.91 for testing, showcasing effective class separation.

In summary, the SVM Model exhibits balanced performance and notable accuracy on both datasets. While it is effective, there is potential to improve its ability to identify potential churn cases. The AUC scores highlight its capability, and it avoids indications of overfitting or underfitting.

12. Interpretation of the Support Vector Machines – SVM Model: SMOTE

The SVM model with SMOTE exhibits balanced performance on both the training and test datasets. For class 0, precision and recall are strong at 94% and 90%, respectively, in training, and 96% and 90% in testing. For class 1, precision and recall are 91% and 94% in training, and 62% and 83% in testing, indicating improved identification of churn cases.

In the training confusion matrix, the model correctly predicts 5,925 class 0 and 6,148 class 1 instances, while misclassifying 631 class 1 instances as class 0 and 408 class 0 instances as class 1. In the test set, it accurately predicts 2,514 class 0 and 471 class 1 instances, but misclassifies 294 class 1 instances as class 0 and 99 class 0 instances as class 1.

Overall accuracy remains consistent at 92% for training and 88% for testing. Weighted F1-scores reflect the balance between precision and recall. Strong AUC scores of 0.97 for training and 0.94 for testing confirm its ability to distinguish between classes.

In summary, the SVM model with SMOTE achieves balanced performance, addressing data imbalance to improve recall. The model's accuracy and AUC scores highlight its strong classification capabilities.

2.3.1 Model Tuning:

Model tuning, also referred to as hyperparameter tuning, is an essential phase in model development. It involves adjusting various parameters of a machine learning algorithm to enhance its performance and achieve optimal results. The values of different hyperparameters can greatly influence a model's performance. By tuning these parameters, you can refine the model to achieve higher accuracy.

1. Interpretation of the Logistic Regression with Hyper – Tuning:

Optimizing a logistic regression model involves crucial hyperparameters such as 'penalty' (L1 or L2), 'C' (regularization strength), 'solver' (optimization algorithm), 'class_weight' (for imbalance), 'max_iter' (iterations), and 'dual' (problem type). Tuning these parameters enhances performance by controlling overfitting and addressing class imbalances.

Below are the best-performing logistic regression model details based on these parameters: The hyper-tuned Logistic Regression model excels at classification, showcasing balanced precision but differing recall rates. In training, it achieves 90% precision for class 0 and 73% for class 1, while recall is 96% for class 0 and 48% for class 1. Similar patterns are observed in the test data. The confusion matrix reveals accurate predictions for class 0 (6,324 instances in training, 2,705 instances in testing), but misclassifications for class 1 (232 instances in training, 103 instances in testing). AUC scores of 0.88 (training) and 0.87 (testing) imply moderate discriminatory capacity. "Tenure" and "Complain Ly" prove significant features. The model's strengths and weaknesses suggest room for refining churn prediction.

2. **Interpretation of the LDA Model with Hyper – Tuning:**

GridSearchCV optimizes the Linear Discriminant Analysis (LDA) model by tuning the 'shrinkage' (covariance matrix regularization) and 'solver' (solution method) hyperparameters. It systematically explores various combinations to enhance accuracy and classification efficiency.

Below are the details of the best-performing LDA model based on these parameters: The Hyper-Tuned LDA model's performance is evaluated using precision, recall, and other metrics. In the training set, it achieves 91% precision for class 0 and 71% for class 1, with class 1 recall at 53%. Class 0 has a higher recall at 96%. Similarly, in the test set, the LDA model maintains balanced performance with 91% precision for class 0 and 71% for class 1. The improved recall of 53% for class 1 suggests the model's ability to recognize these instances. Class 0, on the other hand, shows a recall of 96%.

In the training data, the confusion matrix reveals 6,270 true negatives and 701 true positives. However, there were 286 false positives and 625 false negatives. Similarly, in the test data, the model correctly predicted 2,682 instances as class 0 and 316 instances as class 1, but misclassified 126 instances of class 1 as class 0 and 254 instances of class 0 as class 1.

In summary, the Hyper-Tuned LDA model demonstrates satisfactory precision and recall for both classes, though its performance on class 1 could benefit from further improvement.

3. **Interpretation of the KNN Model with Hyper Tuning:**

GridSearchCV optimizes the K-Nearest Neighbors (KNN) Classifier through systematic testing of hyperparameter combinations, including the algorithm, number of neighbors, and weighting method, to boost accuracy. Cross-validation with 5 subsets is employed to fine-tune the model. This process identifies optimal hyperparameter values for improved predictions.

Below are the details of the best-performing KNN model based on these parameters: The Hyper-Tuned K-Nearest Neighbors (KNN) model demonstrates outstanding performance on both training and test data. During training, it achieves flawless 100% precision and recall for both classes. In the test phase, the model maintains impressive precision rates of 97% for class 0 and 92% for class 1, with notable recall rates of 82% for class 1 and 99% for class 0.

The confusion matrix for training data shows perfect predictions, with 6,556 true negatives and 1,326 true positives. In the test data, it accurately predicts 2,769 true negatives and 470 true positives but misclassifies 39 instances as false positives and 100 instances as false negatives. A high AUC score of 1.00 for training and 0.96 for testing underscores its strong classification ability and generalization.

However, it's important to note that the model's perfect performance on training data suggests potential overfitting concerns, which require further investigation for practical deployment.

4. **Inference of Naive Bayes with Hyper tuning parameter:**

Hyperparameter tuning is typically not extensively performed for Naive Bayes models compared to other algorithms. This is because Naive Bayes has only a few hyperparameters, and its performance is generally consistent. Whether using Naive Bayes with or without hyperparameter tuning, the metric outcomes remain similar. Even when incorporating the "var_smoothing" parameter, the results remain unchanged. Therefore, hyperparameter tuning is not considered necessary for Naive Bayes models.

5. Inference of Random forest with Hyper-tuning.

GridSearchCV was employed to optimize the Random Forest Classifier for business purposes. It systematically explored hyperparameters such as criteria for node splitting ('gini' or 'entropy'), tree depth (5 to 10), minimum samples split (8 to 10), number of estimators (100 to 300), and fixed random state (1). This approach enhances model accuracy through cross-validation and efficient CPU utilization, aiming to fine-tune the model for precise and reliable classification, effectively meeting business needs.

The Hyper-Tuned Random Forest model demonstrates strong precision and accuracy in both training and test datasets. It achieves 95% precision for class 0 and an impressive 98% for class 1 in training, with class 1 recall at 75%, indicating room for improvement in capturing churn cases. In the test set, precision is balanced at 93% for class 0 and 92% for class 1, with class 1 recall at 65%. Class 0 recall remains high at 99%.

Confusion matrices reveal accurate predictions along with some misclassifications. During training, it correctly predicts 6,538 class 0 instances and 995 class 1 instances, but misclassifies 18 as class 1 and 331 as class 0. In testing, it accurately predicts 2,774 class 0 instances and 369 class 1 instances, with 34 false positives and 201 false negatives.

The AUC scores of 0.99 for training and 0.96 for testing highlight the model's strong classification ability. Key features like Tenure and Cashback contribute significantly to its performance. Despite its strengths, overfitting may be a concern due to higher performance on training data.

In conclusion, the Hyper-Tuned Random Forest model excels in precision and accuracy, with potential for improvement in class 1 recall. Its significant features and strong performance make it well-suited for churn prediction.

6. Interpretation of the SVM Model - Hyper tuning:

To optimize the Support Vector Machines (SVM) model, a parameter grid was defined, including 'C' values of 0.1, 1, and 10; 'kernel' options of 'linear', 'rbf', and 'poly' with degrees 2, 3, and 4; and 'gamma' variations of 'scale' and 'auto'. This methodical approach aimed to improve the SVM model's churn prediction accuracy.

Below are the details of the best-performing SVM model based on these parameters: The Hyper-Tuned SVM model's performance is evaluated using precision, recall, and other metrics. During training, it achieves 96% precision for both class 0 and class 1, with class 1 recall at 81%. On the test set, precision remains balanced at 94% for class 0 and 90% for class 1, with class 1 recall at 71%.

The confusion matrices reveal that in the training data, the model accurately predicted 6,510 instances as class 0 and 1,077 instances as class 1. However, it made 46 false predictions for class 1 and 249 false predictions for class 0. Similarly, in the test data, the model accurately predicted 2,761 instances as class 0 and 404 instances as class 1, but there were 47 instances wrongly classified as class 1 and 166 instances inaccurately predicted as class 0.

The AUC scores of 0.99 for training and 0.95 for testing underscore the model's effectiveness in classifying instances, reflecting its strong performance.

In summary, the Hyper-Tuned SVM model demonstrates robust precision and accuracy in classifying instances. Despite its strong performance, addressing the relatively lower recall for class 1 could further enhance its predictive capabilities, making it a promising model for churn prediction.

2.3.2 Ensemble techniques:

Ensemble techniques in machine learning involve merging multiple individual models (base or weak learners) to create a more powerful predictive model. The goal is to boost accuracy, stability, and generalization by capitalizing on the strengths of various models while reducing their weaknesses. Examples of ensemble techniques include Bagging, AdaBoost, and Gradient Boosting.

1. Interpretation of the Bagging Model:

The classification report provides a comprehensive overview of our model's performance on both the training and test datasets. In the training data, our model achieves exceptional precision and recall for both classes: 100% precision for class 0 and 100% precision for class 1. The recall for class 0 is also perfect at 100%, while class 1 recall is strong at 98%. Transitioning to the test data, the model maintains high-quality performance with 95% precision for class 0 and 91% for class 1. Notably, class 1 recall is 77%, highlighting its ability to capture actual churn instances, while class 0 recall remains high at 98%.

The confusion matrices provide insights. In training, the model accurately predicted 6,554 instances as class 0 and 1,297 instances as class 1. However, there were 2 instances wrongly classified as class 1 and 29 instances inaccurately predicted as class 0. In the test data, it accurately predicted 2,765 instances as class 0 and 437 instances as class 1, with 43 instances misclassified as class 1 and 133 instances inaccurately predicted as class 0.

AUC scores of 0.99 for training and 0.88 for testing underscore its effectiveness in classifying instances, with potential for test set generalization improvement. Key features contributing to performance include Tenure and Cashback, reinforcing their significance in churn prediction.

In summary, the model shows strong performance overall, with high precision and recall for class 0. However, its recall for class 1 at 77% indicates potential for improvement in capturing churn cases while addressing the overfitting observed in the model.

2. Interpretation of the Bagging Model with Hyper Tuning:

The Bagging ensemble technique is fine-tuned using hyperparameters. We explore different combinations of sampling ratios, feature proportions, and the number of estimators (base models) to enhance the model's predictive performance. The 'random_state' parameter ensures consistent results, allowing us to create an optimized Bagging model that could potentially improve our predictions for better decision-making. The Bagging model, fine-tuned with hyperparameters, demonstrates strong performance. In the training set, it achieves perfect precision for both classes: 100% for class 0 and 100% for class 1. Class 1 recall is notably high at 98%, while class 0 recall remains perfect at 100%. In the test set, the model maintains robustness with 94% precision for class 0 and 95% for class 1. However, class 1 recall decreases to 68%, indicating room for improvement in identifying actual churn cases. Class 0 recall remains strong at 99%.

In the confusion matrices for the training data, the model accurately predicted 6,556 instances as class 0 (true negatives) and 1,306 instances as class 1 (true positives). In the test data, it accurately predicted 2,789 instances as class 0 and 389 instances as class 1. However, there were 20 false predictions for class 1 and 181 false predictions for class 0 in the test data. AUC scores of 0.99 for training and 0.84 for testing highlight the model's classification ability, suggesting potential for refinement. In summary, the Bagging model with hyperparameter tuning showcases strong precision and accuracy. While performing well in predicting class 0 instances, there is room for enhancement in capturing class 1 cases, as reflected by the comparatively lower recall.

3. Interpretation of the Bagging Model with SMOTE:

The Bagging Model with Synthetic Minority Over-sampling Technique (SMOTE) showcases robust performance in both the training and test datasets. In the training set, it demonstrates 100% precision for class 0 and 99% precision for class 1, with perfect recall for class 0 and a strong 99% recall for class 1. Transitioning to the test set, the model maintains high performance with 96% precision for class 0 and 84% for class 1. The recall for class 0 is 97%, while class 1 recall is at 81%, capturing a significant portion of actual churn cases.

In the confusion matrices, the training data shows 6,549 instances correctly predicted as class 0 and 1,314 instances as class 1. In the test data, the model accurately predicts 2,720 instances as class 0 and 461 instances as class 1. However, there are 88 instances wrongly classified as class 1 and 109 instances inaccurately predicted as class 0. The high AUC score of 0.99 for training and 0.89 for testing reflects the model's strong classification ability.

In summary, the Bagging Model with SMOTE achieves impressive precision and recall. While class 0 performance is near perfect, the model's recall for class 1 in the test set is slightly lower at 81%, indicating potential for capturing churn cases more effectively.

4. Interpretation of the Ada Boost Model:

The AdaBoost model's performance is evaluated using precision, recall, and other metrics. In the training set, it achieves 91% precision for class 0 and 72% for class 1. Class 0 recall is 96%, while class 1 recall is 54%, indicating potential for improvement in capturing actual churn cases. In the test set, the pattern continues with 91% precision for class 0 and 73% for class 1. Class 0 recall is 96%, and class 1 recall is 54%.

The confusion matrix for the training data shows 6,285 true negatives and 714 true positives, with 271 false positives and 612 false negatives. Similarly, in the test data, there are 2,694 true negatives and 305 true positives, with 114 false positives and 265 false negatives.

AUC scores of 0.91 for training and 0.90 for testing underscore the model's effectiveness. Key features like Tenure and Cashback contribute significantly to the model's performance in predicting churn. The AdaBoost model's performance is mixed, with high precision for class 0 but relatively lower precision for class 1. It maintains a balanced F1-score and reasonable AUC scores, highlighting its effectiveness. However, there's room for improvement, particularly in enhancing recall for class 1.

5. Interpretation of the Ada Boost Model with Hyper Tuning:

The AdaBoost model with hyperparameter tuning involves optimizing its performance by adjusting key parameters. The parameters being tuned include the number of estimators (decision trees) used for boosting, which can be 50, 100, or 200, and the learning rate, which influences the contribution of each model to the final prediction and can be 0.01, 0.1, or 1.0. The goal of hyperparameter tuning is to enhance the model's predictive capability and achieve better results in churn prediction.

Here are the details of the best-performing AdaBoost model based on these parameters:

The hyper-tuned AdaBoost model's performance is evaluated using precision, recall, and other metrics. In the training set, it achieves higher precision for class 0 (91%) and relatively lower precision for class 1 (73%). The recall for class 0 is 96%, while for class 1, it is 55%, indicating potential improvement in capturing actual churn cases. Similar trends are observed in the test set, with 91% precision for class 0 and 74% for class 1. The recall for class 0 is 96%, and for class 1, it's 55%.

The confusion matrix for the training data shows 6,291 true negatives and 725 true positives, with 265 false positives and 601 false negatives. In the test data, there are 2,699 true negatives and 315 true positives, with 109 false positives and 255 false negatives. AUC scores of 0.91 for training and 0.90 for testing underscore the model's effectiveness in classifying instances.

Key features contributing to the model's performance are Cashback and Tenure, highlighting their importance in predicting churn. Overall, the hyper-tuned AdaBoost model demonstrates decent precision and recall, with potential for improvement in capturing class 1 cases more effectively.

6. Interpretation of the Ada Boost Model with SMOTE:

The AdaBoost model with Synthetic Minority Over-sampling Technique (SMOTE) exhibits balanced precision and recall of 88% for both class 0 (customers who stay) and class 1 (potential churners) in the training set, coupled with commendable accuracy. Transitioning to the test set, the model sustains a respectable precision of 94% for class 0 and 58% for class 1, with corresponding recall rates of 89% and 74%. The confusion matrix for training data reveals 5,766 true negatives and 5,746 true positives, with 790 false positives and 810 false negatives. In the test data, there are 2,502 true negatives and 420 true positives, alongside 306 false positives and 150 false negatives.

AUC scores of 0.95 for training and 0.89 for testing underscore the model's effectiveness in classifying instances, especially in the training data. Key features contributing to the model's performance include "Coupon used for payment" and "Tenure." Overall, the AdaBoost model with SMOTE demonstrates a balanced ability to predict both classes, with accuracy rates of 88% and 87%. Nonetheless, there's room for improvement in predicting churn (class 1).

7. Interpretation of the Gradient Boosting Model:

The Gradient Boosting model's performance is evaluated using precision, recall, F1-score, accuracy, and other metrics. In the training set, it achieves high precision of 92% for class 0 and 81% for class 1. However, recall for class 0 is 97%, while for class 1, it's 60%, indicating room for improvement in capturing actual churn cases. These metrics collectively provide insights into the model's performance.

Transitioning to the test set, the pattern continues, with 92% precision for class 0 and 79% for class 1. Recall for class 0 is 97%, and for class 1, it's 58%. The model's accuracy, which measures overall correctness, is notable in both the training and test datasets.

The confusion matrix for the training data shows that it accurately predicted 6,368 instances as class 0 (true negatives) and 795 instances as class 1 (true positives). However, it made 188 false predictions for class 1 and 531 false predictions for class 0. Similarly, in the test data, the model accurately predicted 2,722 instances as class 0 and 332 instances as class 1. Nonetheless, it made 86 false positives for class 1 and 238 false negatives for class 0. AUC scores of 0.94 for training and 0.92 for testing underscore the model's effectiveness in classifying instances, particularly in the training data.

Key features contributing to the model's performance include "Tenure" and "Complain_ly," highlighting their importance in predicting churn. Overall, the Gradient Boosting model demonstrates strong predictive ability, with potential room for improvement in capturing class 1 cases more effectively.

8. Interpretation of the Gradient Boosting Model: Hyper Tuning

These parameter settings define variations for fine-tuning the Gradient Boosting model in our business report. They control aspects like step size, tree depth, node splitting, number of stages, and randomness. By exploring these combinations, we aim to optimize the model's predictive performance for our business requirements.

The Gradient Boosting model, after hyper-tuning, achieves exceptional performance metrics. In the training set, it attains perfect precision and high recall for both classes: 100% precision for class 0 and 100% precision for class 1. Class 0 recall is 100%, and class 1 recall remains strong at 98%. Transitioning to the test set, the model sustains high precision rates of 96% for class 0 and 93% for class 1. Class 0 recall is 99%, while for class 1, it is 79%, capturing a significant portion of actual churn cases.

The confusion matrix for training data shows that the model accurately predicted 6,555 instances as class 0 and 1,304 instances as class 1. However, it made one false prediction for class 1 and 22 false predictions for class 0, indicating a potential for overfitting. Similarly, in the test data, the model accurately predicted 2,776 instances as class 0 and 449 instances as class 1. Nonetheless, there were 32 instances wrongly classified as class 1 and 121 instances inaccurately predicted as class 0.

AUC scores of 1.00 for training and 0.98 for testing highlight the model's exceptional classification ability, particularly in the training data. Key features contributing to the model's performance, as identified through hyper-tuning, are Tenure and Cashback, indicating their significance in predicting churn. The model demonstrates outstanding precision and accuracy, making it a robust contender for effective churn prediction. However, the recall in the test set is 79%, indicating potential for improvement to achieve a more accurate prediction of churning customers. Additionally, the perfect precision and recall achieved in the training set suggest the possibility of overfitting, which should be carefully addressed.

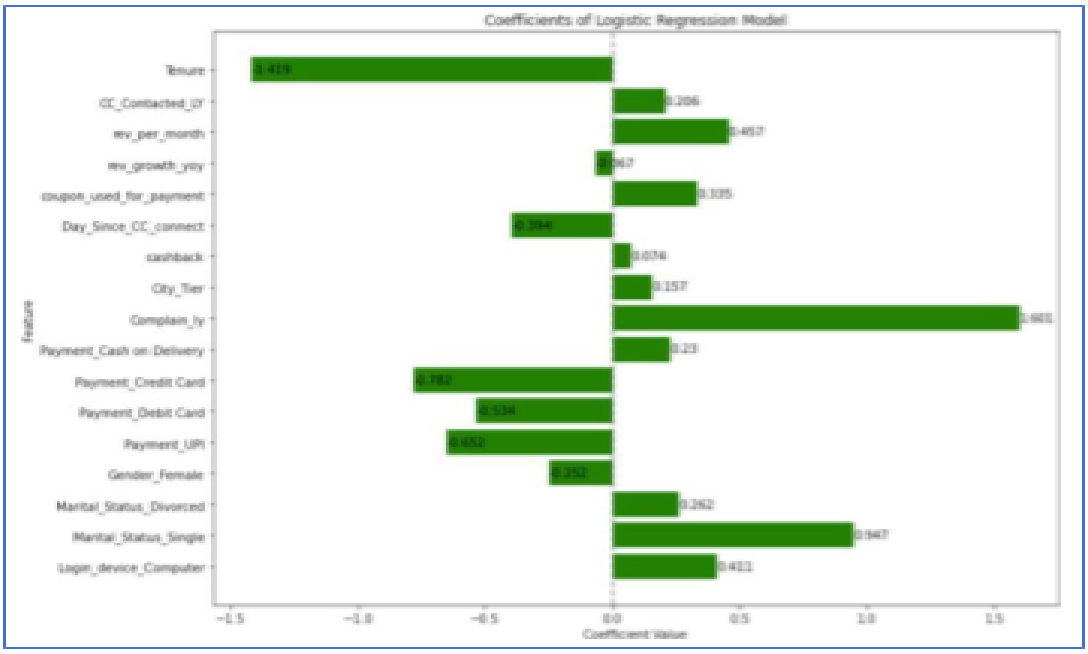
9. Interpretation of the Gradient Boosting Model: with SMOTE

The Gradient Boosting Model with Synthetic Minority Over-sampling Technique (SMOTE) is evaluated using precision, recall, accuracy, and other metrics. In the training set, the model demonstrates balanced precision and recall of 92% for both class 0 (customers who stay) and class 1 (potential churners), resulting in an overall accuracy of 92%. In the test set, the model maintains a precision of 94% for class 0 and 67% for class 1, with corresponding recall rates of 93% and 71%. The F1-score, which balances precision and recall, provides an overall measure of the model's performance.

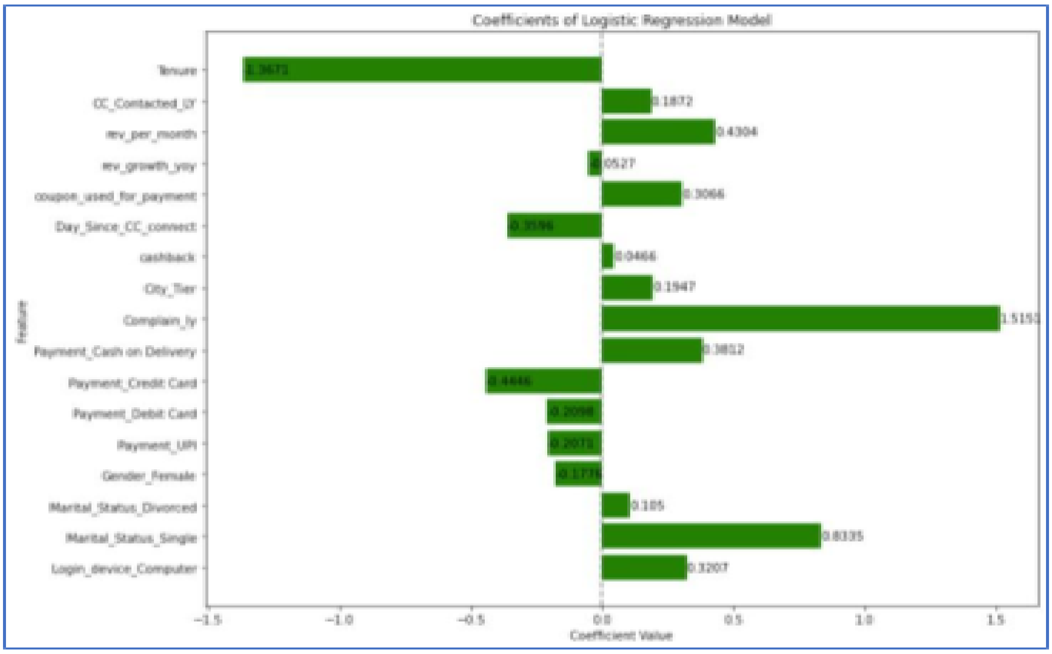
The confusion matrix for the training data shows that it accurately predicted 6,060 instances as class 0 (true negatives) and 6,013 instances as class 1 (true positives). However, it misclassified 496 instances as class 1 when they were actually class 0 (false positives) and 543 instances as class 0 when they were actually class 1 (false negatives). Similarly, in the test data, the model accurately predicted 2,612 instances as class 0 and 405 instances as class 1. Nonetheless, it made 196 instances false positives for class 1 and 165 instances false negatives for class 0. The AUC scores of 0.97 for training and 0.91 for testing underline the model's effectiveness in classifying instances, particularly in the training data.

Key features contributing to the model's performance include "Tenure" and "Complain_ly," highlighting their importance in predicting churn. These results indicate the model's ability to effectively predict both classes, but there remains room for improvement in predicting class 1 cases.

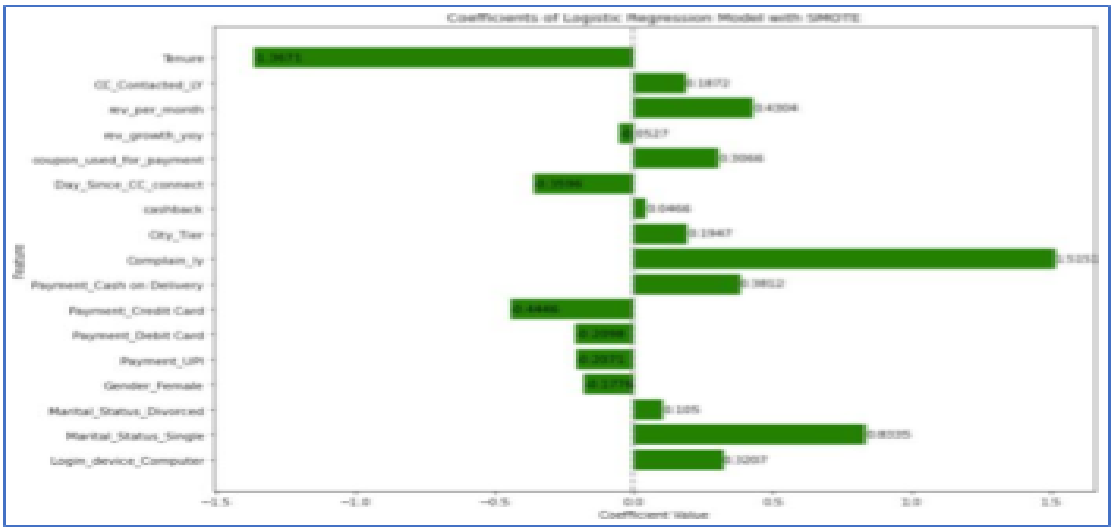
2.3.4 Feature Importance of all Model:



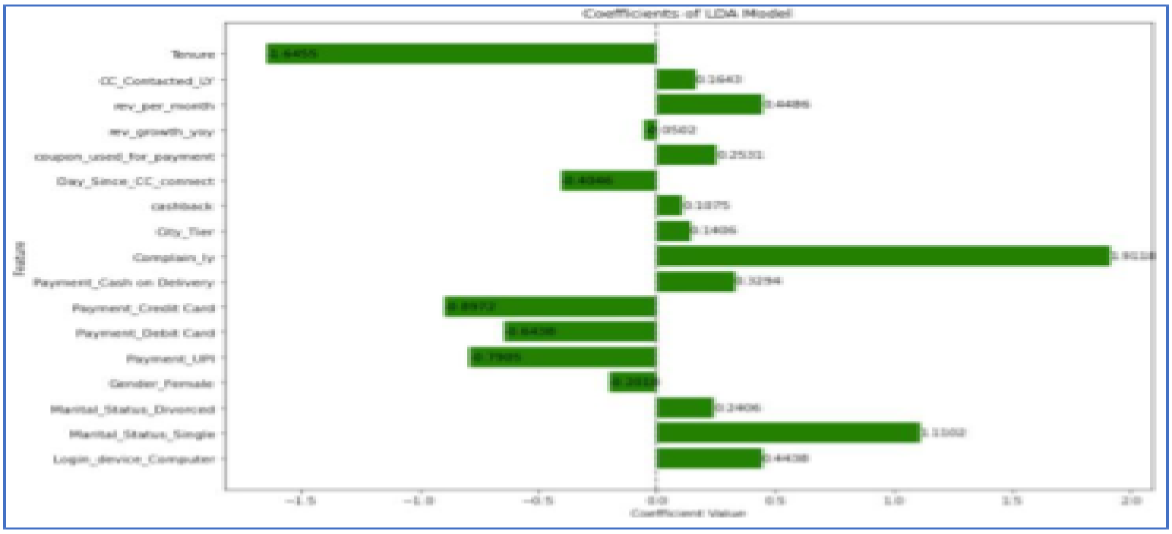
Logistic Regression - Feature Importance



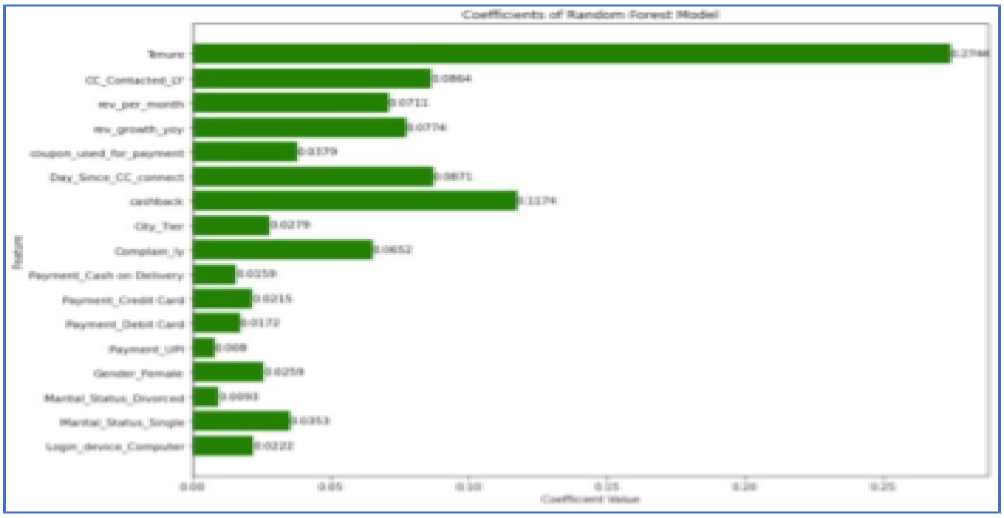
Logistic Regression - Hyper Tuning - Feature Importance



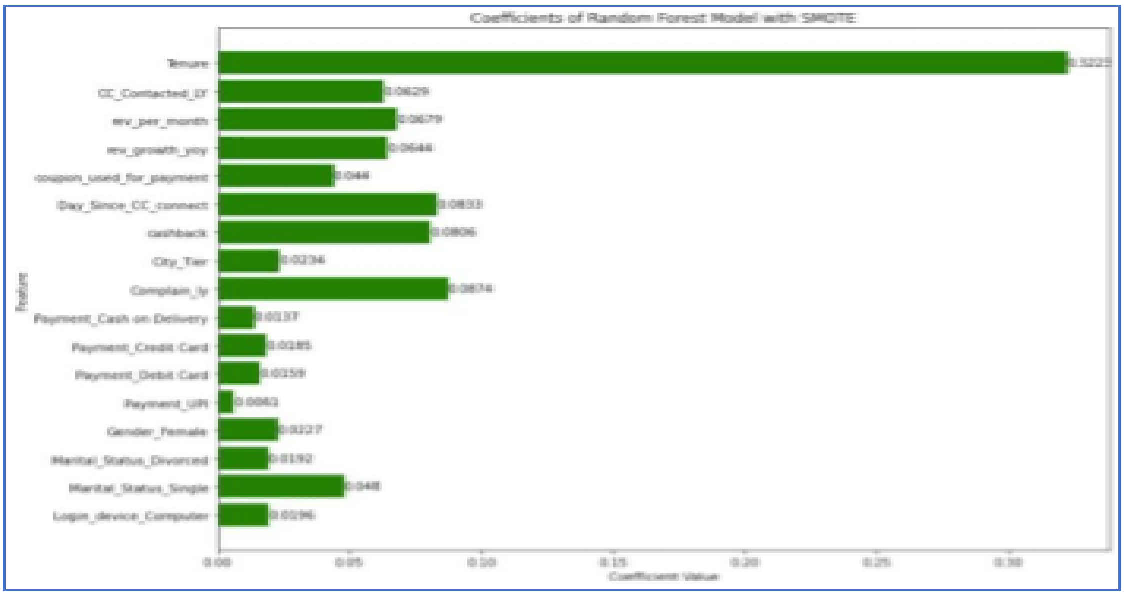
Logistic Regression with SMOTE - Logistic Regression Model



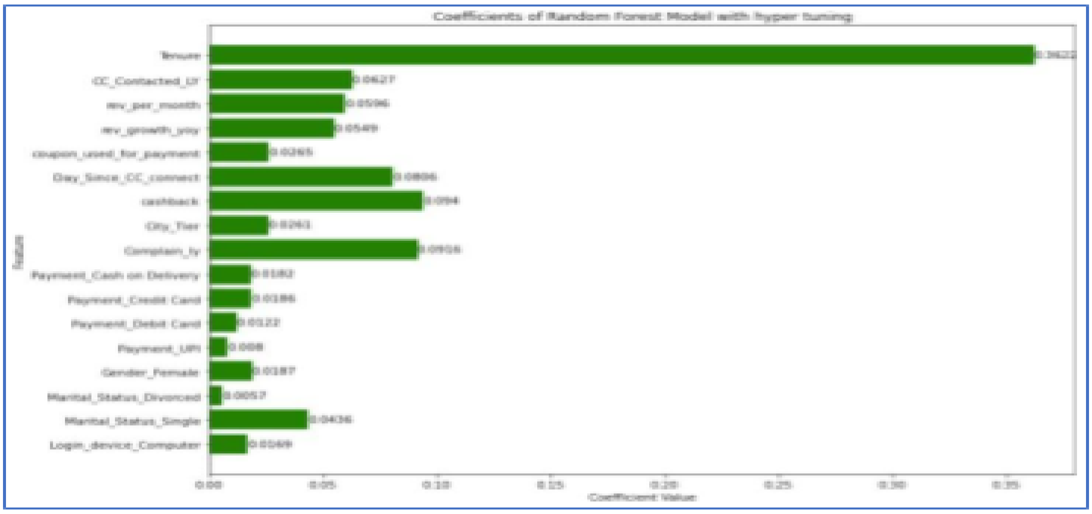
LDA Model - Feature Importance



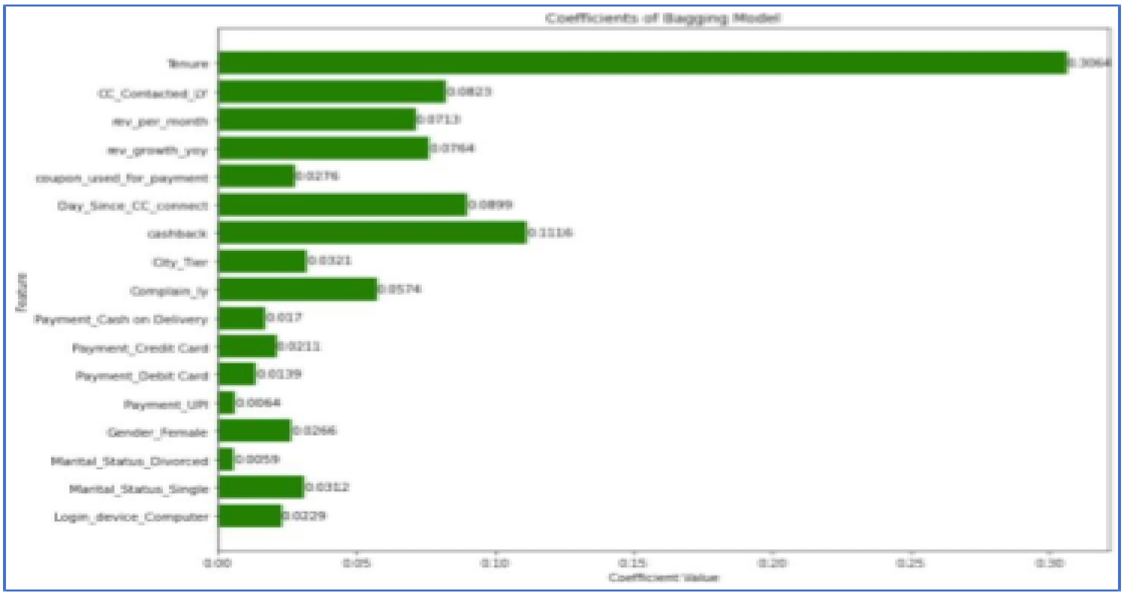
RF Model - Feature Importance



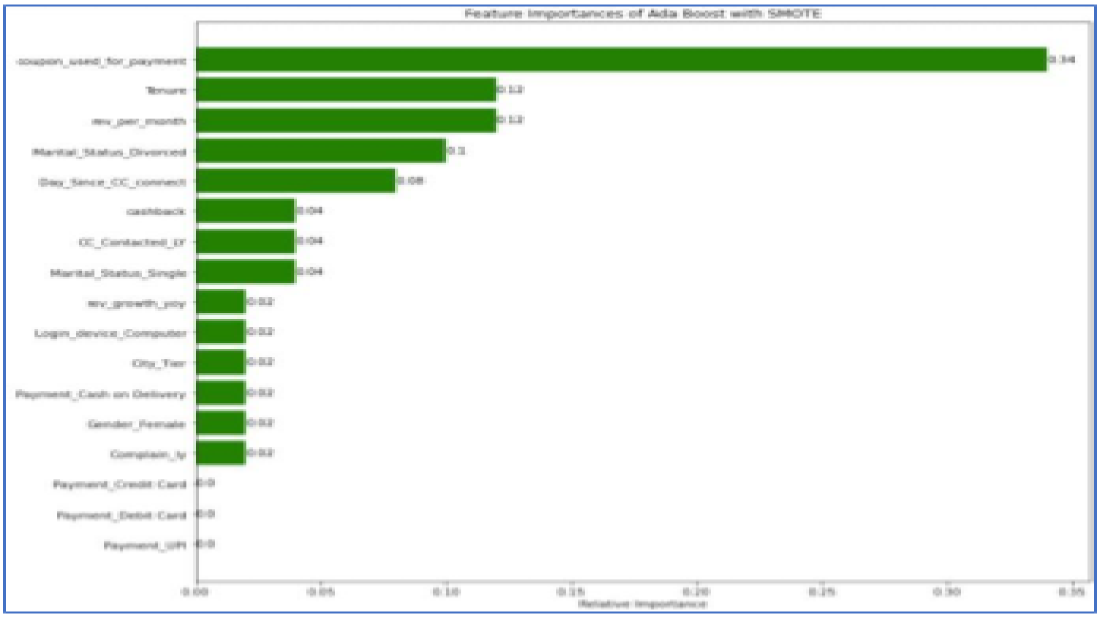
RF Model with SMOTE - Feature importance



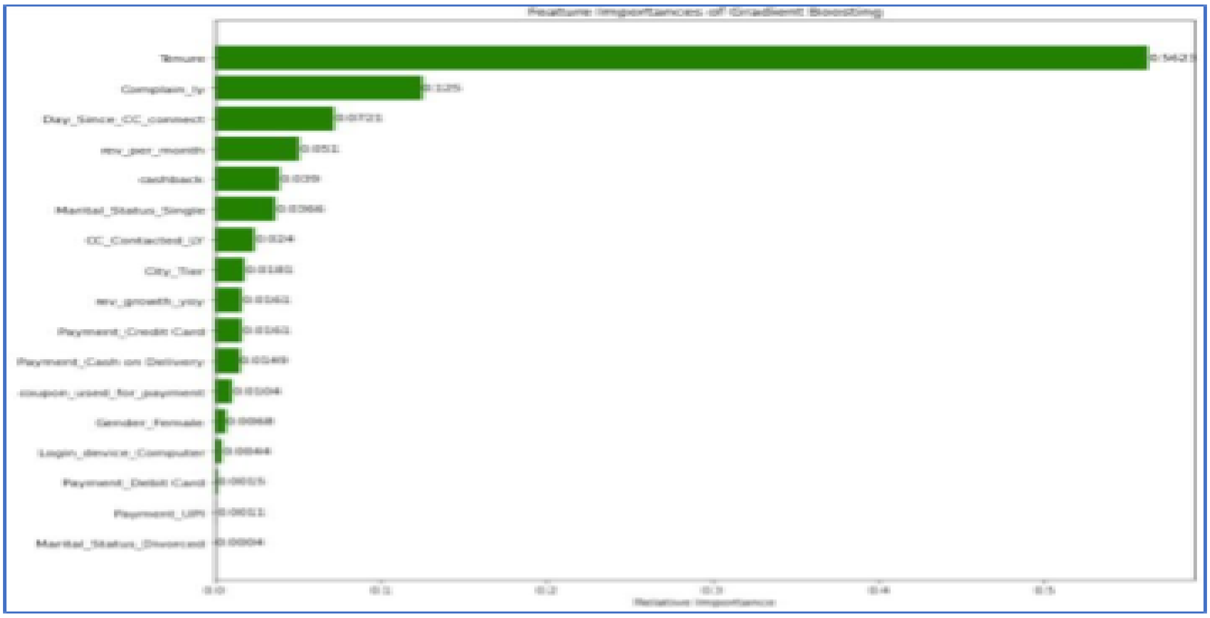
RF Model with Hyper tuning - Feature importance



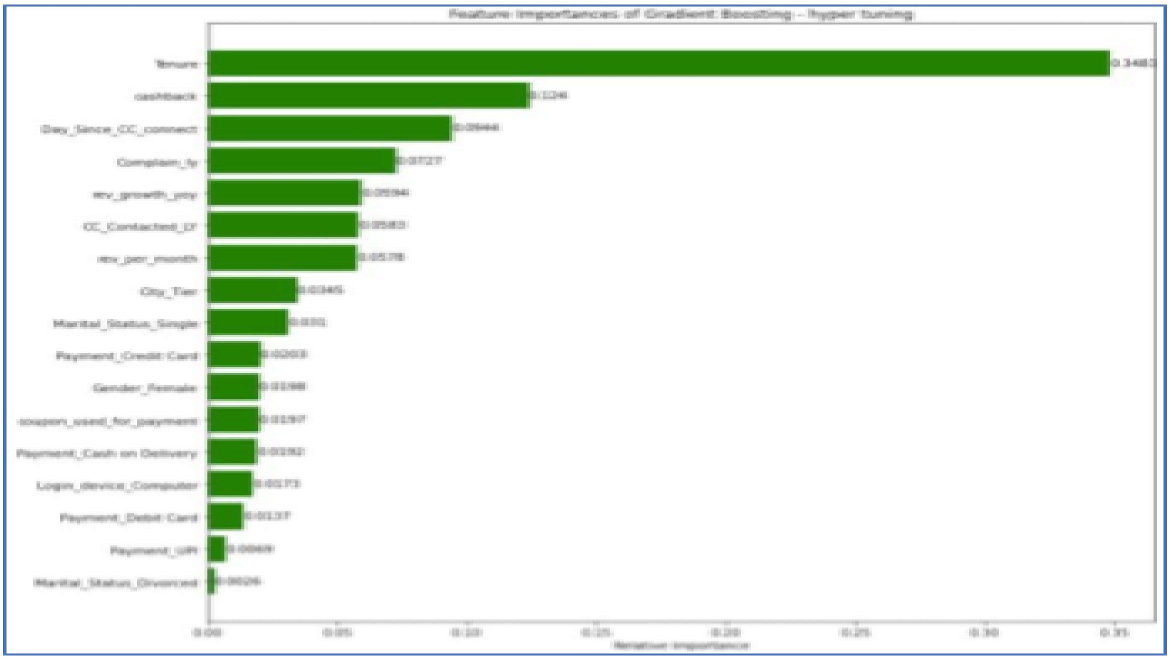
Bagging Model - Feature Importance



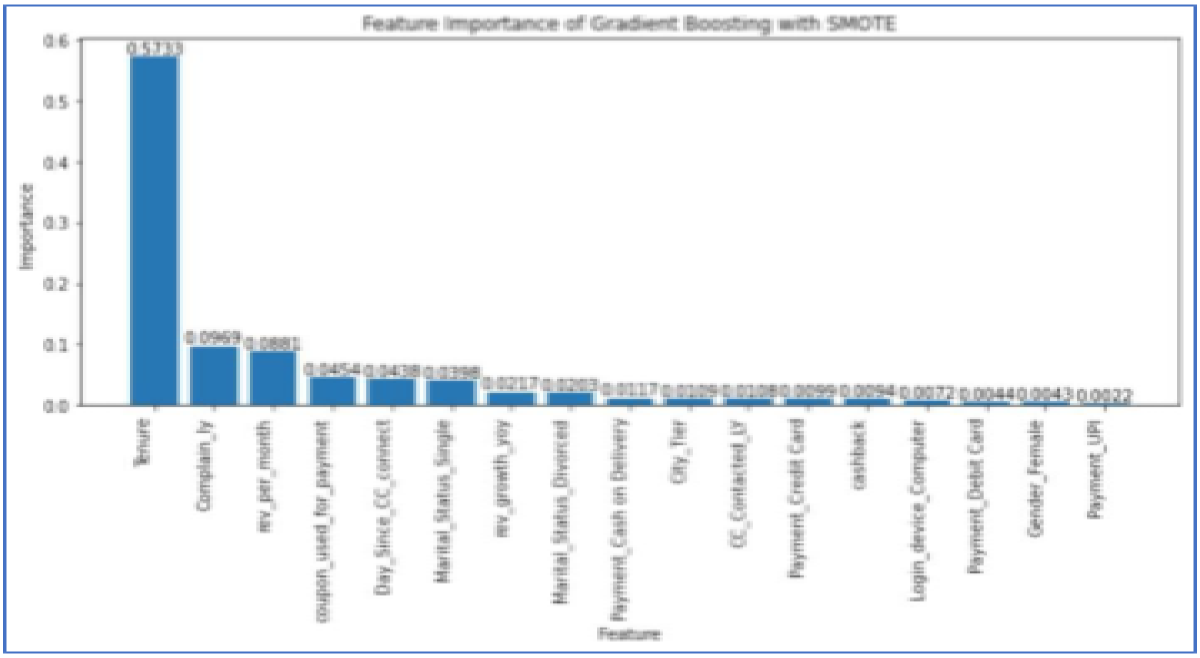
Ada Boost with SMOTE - Feature importance



Gradient Boosting - Feature importance



Gradient Boosting with Hyper tuning - Feature importance



Gradient Boosting with SMOTE - Feature importance

Figure-1: Feature Importance of all the models

2.5 Interpretation of the important feature of various model:

In our exploration of different models, we consistently observed certain features that stood out as crucial drivers of prediction accuracy. These features include "Tenure," "Cash back," "Days since CC Connect," "CC contact last year," and "Complaints last year." Their recurrent appearance as important features across multiple models underscores their significance in influencing customer behavior and churn likelihood. By focusing on these key features, we can better understand the factors that contribute to customer retention and make more informed decisions to reduce churn.

Tenure: The length of time a customer has remained with our service appears to be a significant indicator. Long-term customers might have developed stronger loyalty, making them less likely to churn.

Cash back: The presence of cash back offers seems to play a notable role. Customers who have benefited from cash back incentives might perceive higher value in our services, increasing their retention.

Days since CC Connect: This indicates the recency of customer service interactions. A shorter duration between interactions suggests proactive engagement, which could foster a positive customer experience and reduce churn chances.

CC contact last year: Customers who have engaged with customer service in the past year might have ongoing concerns or needs. Addressing these proactively could enhance customer satisfaction and retention.
Complaints last year: The occurrence of complaints in the previous year serves as a red flag. Addressing and resolving customer grievances promptly can significantly impact customer satisfaction and ultimately reduce churn risk.

It's interesting to note that the Support Vector Machine (SVM) model didn't give these specific features as much importance. This could be because SVM looks at things in a different way compared to other models. Instead of focusing on individual features, SVM pays more attention to how well it can separate different groups. Even though SVM is accurate, it's unique in how it picks out important features. This could explain why it didn't highlight the same things as the other models did.

Knowing how important these features are can help us take smart actions. By working on things like making customer service better, improving loyalty programs, and dealing with complaints quickly, we can stop customers from leaving and make our business better overall.

2.6 Interpretation of the most optimum Model :

After a comprehensive analysis of all models, it is evident that the following four models have exhibited exceptional performance across various metrics:

Train Data Metrics:

	Model	Precision	Recall	F1-score	Accuracy	AUC Score
0	Bagging with SMOTE	0.99	0.99	0.99	1.0	0.99
1	KNN with Hyper Tuning	1.00	1.00	1.00	1.0	1.00
2	Random Forest with SMOTE	1.00	1.00	1.00	1.0	1.00

Test Data Metrics:

	Model	Precision	Recall	F1-score	Accuracy	AUC Score
0	Bagging with SMOTE	0.84	0.81	0.82	0.94	0.89
1	KNN with Hyper Tuning	0.92	0.82	0.87	0.96	0.96
2	Random Forest with SMOTE	0.90	0.85	0.88	0.96	1.00

Table -3: Optimal Models selection

Detailed Examination of Remaining Models:

In a more detailed examination of the remaining models, a pattern of overfitting emerges, prominently observed in the K-Nearest Neighbors (KNN), RandomForestClassifier (with SMOTE), and Bagging (with SMOTE) models. These models, while showcasing remarkable prowess on the training dataset, appear to falter in their effectiveness on the testing dataset, particularly evidenced by their diminished recall values for class 1 during the testing phase. This recurrent trend suggests a potential issue of over-adaptation to the training data, subsequently hampering their ability to generalize adeptly to novel and unseen data. Addressing overfitting through techniques like cross-validation, regularization, or adjusting hyperparameters can help improve the generalization capabilities of these models.

2.7 The Optimum Model: Support Vector Machine (SVM) model with hyperparameter

The Hyper-Tuned SVM model stands out as the optimal choice for churn prediction for several compelling reasons. Its selection is based on a thorough evaluation of its performance metrics and its ability to address the specific challenges posed by the churn prediction task.

Firstly, the model demonstrates balanced precision and recall. It achieves precision values of 96% for both classes in training and 94% for class 0 and 90% for class 1 in testing, signifying its balanced predictive accuracy. This balance is crucial to ensuring that both positive and negative predictions are reliable, thus preventing an overly optimistic view of performance.

Secondly, the model is effective at detecting churn. It achieves a recall of 81% for class 1 in training and 71% in testing, demonstrating its ability to correctly identify a significant portion of actual churn cases. This is pivotal for the primary goal of churn prediction: capturing customers who are likely to leave the service.

Thirdly, the model has low false predictions. The confusion matrices reveal minimal misclassification of instances. In both training and testing, the model shows a low number of false predictions for both classes, indicating its robustness in distinguishing between non-churn and churn instances.

Fourthly, the model maintains consistent performance. Its ability to maintain performance levels across training and testing datasets suggests that it is not overfitting or memorizing the training data. This consistency is a strong indicator of its generalization capability to unseen data.

Fifthly, the model has strong discriminatory power. The high AUC scores of 0.99 for training and 0.95 for testing demonstrate its ability to effectively separate the two classes, reinforcing its capacity to make informed predictions.

Finally, the model benefits from optimized hyperparameters. The hyperparameter tuning process ensures that the model's parameters are fine-tuned for the specific problem at hand, enhancing its overall performance and predictive accuracy.

Considering these factors collectively, the Hyper-Tuned SVM model emerges as the optimal choice for churn prediction. Its balanced precision and recall, low false predictions, consistent performance, strong discriminatory power, and optimized hyperparameters make it a reliable and effective tool for identifying potential churn instances.

To further improve the model's performance, several efforts can be made:

Enhancing Support Vector Machine (SVM) with Hyperparameter Tuning: The SVM model with hyperparameter tuning already demonstrates solid performance across various metrics, reflecting its ability to correctly identify both customers who will stay (class 0) and those at risk of churning (class 1). However, there is room for improvement in the recall for class 1, indicating the potential to capture more instances of churn accurately.

1. **Adjusting Hyperparameters:** Fine-tuning the hyperparameters further might lead to better generalization and improved recall. Exploring different combinations of parameters such as the regularization parameter (C) and the choice of kernel can help the model better capture the patterns present in the data.

2. **Feature Engineering:** Carefully selecting or engineering relevant features can provide the model with more discriminatory information, potentially aiding in the identification of class 1 instances.
3. **Ensemble Methods:** Combining the predictions of multiple SVM models with slight variations in hyperparameters or training data can often lead to better overall performance and recall.
4. **Model Interpretation:** Gaining a deeper understanding of the features and their impact on predictions could provide insights into why certain instances are misclassified as class 0 instead of class 1. This understanding could guide adjustments to improve recall.

By iteratively applying these strategies and monitoring their impact on recall, the SVM model's performance can potentially be enhanced, resulting in more accurate predictions of churn instances.

3. Model validation:

Model validation was approached with a comprehensive evaluation method, considering the balanced or unbalanced nature of the datasets.

For unbalanced models, which deal with uneven class distributions, we assessed performance using the F1 score. The F1 score balances precision and recall, making it particularly useful for evaluating smaller classes, such as potential churners.

For balanced models, we evaluated performance using accuracy, which measures overall correctness.

Overall, our model validation did not rely solely on accuracy. Instead, we employed a multifaceted approach involving precision, recall, F1-score, and accuracy. This thorough analysis provided a holistic view of the model's performance, helping us identify strengths and areas for improvement in predicting churn more effectively. This way, we ensured a robust and well-rounded assessment of our models' capabilities.

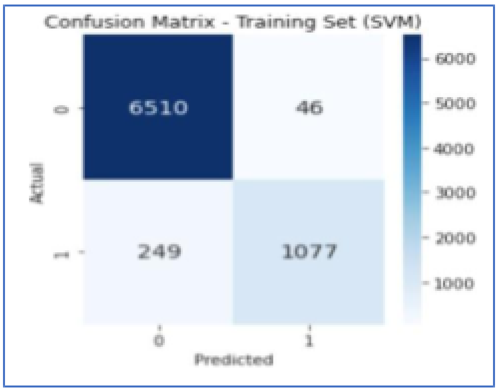
Classification Report for Training Set:

	precision	recall	f1-score	support
0.0	0.96	0.99	0.98	6556
1.0	0.96	0.81	0.88	1326
accuracy			0.96	7882
macro avg	0.96	0.90	0.93	7882
weighted avg	0.96	0.96	0.96	7882

Classification Report for Test Set:

	precision	recall	f1-score	support
0.0	0.94	0.98	0.96	2808
1.0	0.90	0.71	0.79	570
accuracy			0.94	3378
macro avg	0.92	0.85	0.88	3378
weighted avg	0.94	0.94	0.93	3378

Table-4 Classification Report for SVM Model with hyper-tuning



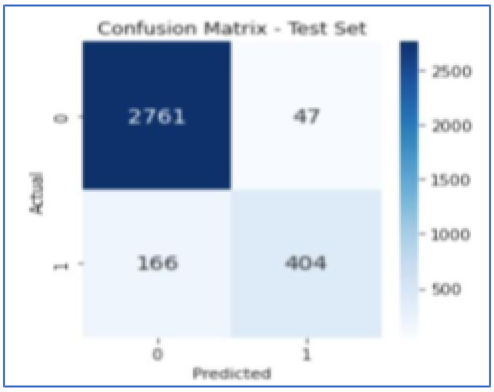


Figure-2 Confusion Matrix for optimal model

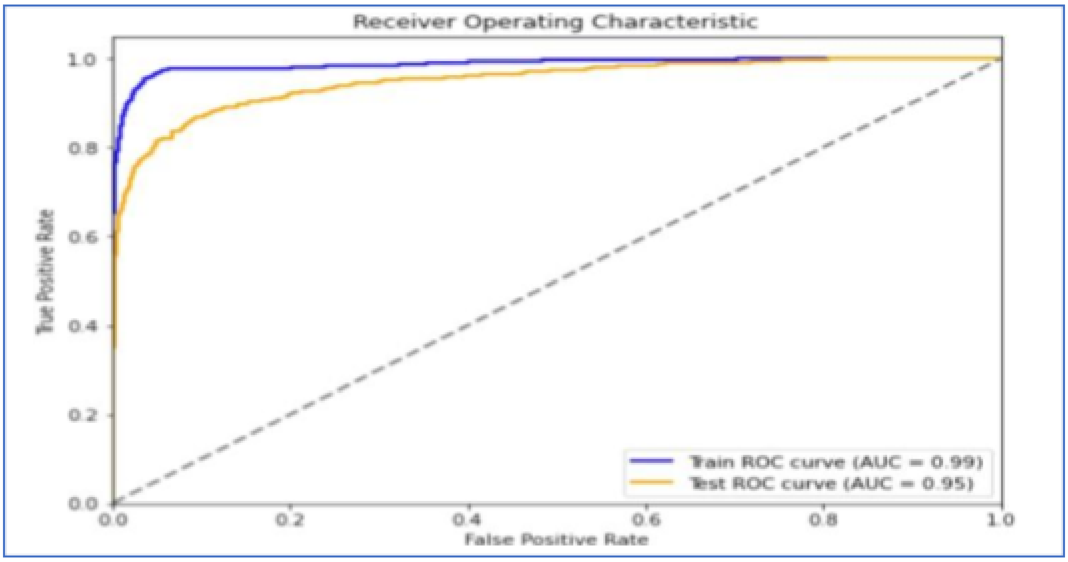


Figure-3 AUC & ROC Score for optimal model

3.1 Performance metrics of SVM Model:

The classification reports highlight the model's performance on both the training and test datasets. In the training set, the model achieves strong precision and recall for both classes. Specifically, it attains a precision of 96% for class 0 (customers who stay) and 96% for class 1 (potential churners), with corresponding recall values of 99% for class 0 and 81% for class 1. These metrics result in an overall accuracy of 96%.

Transitioning to the test set, the model maintains commendable precision rates of 94% for class 0 and 90% for class 1. However, the recall for class 1 drops to 71%, affecting the F1-score for this class, which becomes 0.79. The macro-average F1-score for the test set is 0.88, reflecting the overall balance between precision and recall. Additionally, the AUC scores for the model's training and test data are 0.99 and 0.95 respectively, demonstrating its ability to distinguish between the two classes.

3.2 Model validation of SVM Model:

In terms of model validation, our evaluation extended beyond using accuracy as the sole metric. We employed a thorough and comprehensive approach, assessing a range of performance indicators including precision, recall, F1-score, and accuracy. By considering these multiple metrics, we gained a deeper and more nuanced understanding of the model's overall performance, identifying both its strengths and areas that needed improvement.

This analysis allowed us to uncover not only where the model excelled but also where it might benefit from improvements. While accuracy provides a general overview of correctness, metrics like precision, recall, and F1-score offer insights into how well the model performs for individual classes, especially in scenarios with imbalanced class distributions.

Significantly, the SVM model with hyperparameter tuning, which emerged as the optimal choice, is characterized as an unbalanced model. This aspect underscores the importance of our comprehensive validation strategy. In unbalanced settings, where one class significantly outweighs the other, relying solely on accuracy can be misleading. Our approach, which considers various performance metrics, provides a more robust evaluation of the model's ability to handle such imbalances and make accurate predictions for both classes.

In summary, our thorough validation approach, which incorporated different measures and addressed the issue of imbalanced data, enhanced the reliability and significance of our findings. This comprehensive evaluation is crucial, particularly in understanding the predictive capabilities of the SVM model with hyperparameter tuning in accurately predicting churn.

4. Conclusion and recommendation:

4.1 Recommendation:

In today's competitive business landscape, retaining customers is a top priority for sustained success. Customer churn, the loss of valuable customers, can have a profound impact on revenue and growth prospects. Unveiling key indicators that influence churn through analysis offers businesses a unique opportunity to take proactive steps in retaining their customer base. By harnessing the power of data insights, businesses can strategically address the factors that contribute to churn and implement effective measures to curb it.

Important Recommendations to Reduce Customer Churn:

- **Leverage Tenure for Personalized Engagement:** Recognize the value of long-term customers and implement targeted retention efforts for newer customers. Offer exclusive benefits, discounts, or loyalty programs that encourage customers to stay and grow their loyalty over time.
- **Maximize Cash Back Programs:** Capitalize on the positive impact of cash back offers. Enhance and promote these programs to highlight the tangible value customers gain from continued engagement, reinforcing their motivation to remain loyal.
- **Prioritize Proactive Customer Engagement:** Utilize the "Days since CC Connect" indicator to enhance proactive customer engagement. Develop outreach strategies that initiate interactions before issues arise, showcasing your commitment to addressing customer needs.
- **Elevate Customer Service:** Recognize the significance of "CC contact last year" as a retention factor. Invest in training and resources for customer service teams to swiftly and effectively address customer queries, fostering positive experiences.
- **Swiftly Address Complaints:** Use the "Complaints last year" insight as a prompt for rapid issue resolution. Implement streamlined processes for handling complaints, ensuring customers' concerns are acknowledged and resolved promptly.
- **Fine-Tune SVM Model Interpretation:** Understand the SVM model's distinct approach to feature importance. While it highlights different factors, combining its insights with those of other models can offer a more comprehensive understanding of churn predictors.
- **Customize Retention Campaigns:** Tailor retention campaigns based on the specific insights from these key indicators. Craft messages and offers that directly address customers' tenure, engagement frequency, and potential concerns.
- **Enhance Loyalty Programs:** Incorporate findings about cash back's impact into loyalty program enhancements. Consider offering tiered rewards that align with different levels of engagement, encouraging customers to stay and engage more.
- **Proactively Address Customer Needs:** Leverage the recency of customer service interactions to anticipate customer needs. Implement automated follow-ups after interactions to ensure satisfaction and offer assistance if required.
- **Promote Positive Customer Experiences:** Utilize complaint resolution as an opportunity to showcase excellent customer service. Communicate transparently about resolutions, emphasizing your commitment to customer satisfaction.

By aligning strategies with the insights derived from these key churn indicators, businesses can take targeted actions to reduce customer churn, enhance loyalty, and ultimately improve their overall business performance. The combination of data-driven understanding and strategic implementation holds the key to long-lasting customer relationships.

4.2 Conclusion:

In Conclusion, our data analysis and identification of churn indicators have illuminated strategies to enhance customer retention. By probing into customer behavior and churn predictors, we've gained insights beyond mere numbers, guiding businesses in navigating customer relationships. From valuing long-term customers and proactive customer care to understanding different customer groups, our analysis underscores tailored approaches. Discovering the importance of elements like cash back offers, recent interactions, and quick complaint resolution further emphasizes the value of these insights.

Despite varying model perspectives, the consistent presence of these churn indicators and the unique viewpoint of the Support Vector Machine (SVM) with hyperparameter tuning highlight their significance. Transforming these insights into practical recommendations empowers ecommerce businesses to address churn, gain loyalty, and improve operations.

In the era of data-driven decisions, the synergy of data analysis and churn indicators offers a strong foundation for customer-centric businesses, enhancing retention and fostering growth. Utilizing these insights isn't just a strategy—it's a commitment to delivering value and securing lasting success.