**A white and blue background

Description automatically generated**

**Capstone – Project**

**Table of Content**

[Problem Statement 1 Project Notes 1 6](#_Toc204015580)

[1.1 Business Context 6](#_Toc204015581)

[1.1 Business Objective 6](#_Toc204015582)

[1.2 Data Landscaping 6](#_Toc204015583)

[1.3 Data Overview 7](#_Toc204015584)

[1.4 Descriptive statistics 8](#_Toc204015585)

[1.5 Exploratory data analysis 10](#_Toc204015586)

[1.5.1 Univariate analysis 10](#_Toc204015587)

[1.5.1.1 Churn 10](#_Toc204015588)

[1.5.1.2 City tier 11](#_Toc204015589)

[1.5.1.3 Payment 12](#_Toc204015590)

[1.5.1.4 Gender 13](#_Toc204015591)

[1.5.1.5 Service score 14](#_Toc204015592)

[1.5.1.6 Account User Count 15](#_Toc204015593)

[1.5.1.7 Customer complaint 16](#_Toc204015594)

[1.5.1.8 Tenure 17](#_Toc204015595)

[1.5.1.9 Customers contact with customer care 18](#_Toc204015596)

[1.5.1.10 Y-o-Y Revenue Growth 19](#_Toc204015597)

[1.5.1.11 Account segments Vs Y-o-Y revenue growth 20](#_Toc204015598)

[1.5.1.12 Tenure Vs Y-o-Y revenue growth 21](#_Toc204015599)

[1.5.2 Multivariate Analysis 22](#_Toc204015600)

[1.5.2.1 Heatmap 22](#_Toc204015601)

[1.6 Removal of unwanted variables 24](#_Toc204015602)

[1.7 Missing value treatment 24](#_Toc204015603)

[1.8 Outlier treatment 25](#_Toc204015604)

[1.9 Business insights from EDA 25](#_Toc204015605)

[1.10 Model Evaluation Criteria 25](#_Toc204015606)

[1.11 Pre-Defined Functions 26](#_Toc204015607)

[1.12 Model Building 26](#_Toc204015608)

[1.12.1 Cross-validation performance across multiple models 26](#_Toc204015609)

[1.13 Individual Models 28](#_Toc204015610)

[1.13.1 Decision Tree Classifier 28](#_Toc204015611)

[1.13.1.1 Base model 28](#_Toc204015612)

[1.13.1.2 Tuned decision tree classifier 29](#_Toc204015613)

[1.13.2 Random Forest Classifier 30](#_Toc204015614)

[1.13.2.1 Base model 30](#_Toc204015615)

[1.13.2.2 Tuned random forest classifier 31](#_Toc204015616)

[1.13.2.3 Tuned random forest classifier on balanced dataset 31](#_Toc204015617)

[1.13.3 XGBoost Classifier 33](#_Toc204015618)

[1.13.3.1 Base model 33](#_Toc204015619)

[1.13.3.2 Tuned XGB classifier 34](#_Toc204015620)

[1.13.3.3 Tuned XGB classifier on the balanced dataset 36](#_Toc204015621)

[1.13.4 Performances of the individual models 38](#_Toc204015622)

[1.13.4.1 Training data 38](#_Toc204015623)

[1.13.4.2 Validation data 39](#_Toc204015624)

[1.14 Stacking Classifiers 41](#_Toc204015625)

[1.14.1 Stacking classifier combination #1 41](#_Toc204015626)

[1.14.2 Stacking classifier combination #2 43](#_Toc204015627)

[1.14.3 Stacking classifier combination #3 45](#_Toc204015628)

[1.14.4 Performances of the stacking classifiers 47](#_Toc204015629)

[1.14.4.1 Training data 47](#_Toc204015630)

[1.14.4.2 Validation data 48](#_Toc204015631)

[1.15 Best Model – Performance on the Testing Data 49](#_Toc204015632)

[1.16 Insights 50](#_Toc204015633)

[1.17 Recommendations 50](#_Toc204015634)

**Data Tables**

[TABLE 1. Descriptive statistics of the data 8](#_Toc204015635)

[TABLE 2. Missing values per feature 24](#_Toc204015636)

**Charts & Figures**

[FIG. 1 Countplot for Churn 10](#_Toc204015637)

[FIG. 2 Countplot for city tier vs. churn 11](#_Toc204015638)

[FIG. 3 Countplot of payment vs churn 12](#_Toc204015639)

[FIG. 4 Countplot of genders vs churn 13](#_Toc204015640)

[FIG. 5 Countplot for service score vs churn 14](#_Toc204015641)

[FIG. 6 Countplot of account segment vs churn 15](#_Toc204015642)

[FIG. 7 Countplot of customer complaint vs churn 16](#_Toc204015643)

[FIG. 8 Histogram for tenure 17](#_Toc204015644)

[FIG. 9 Histogram for CC\_Contacted\_LY 18](#_Toc204015645)

[FIG. 10 Histogram for Y-o-Y Revenue Growth 19](#_Toc204015646)

[FIG. 11 Boxplots for account segments vs Y-o-Y revenue growth 20](#_Toc204015647)

[FIG. 12 Boxplots for tenure vs Y-o-Y revenue growth 21](#_Toc204015648)

[FIG. 13 Heatmap 22](#_Toc204015649)

[FIG. 14 Cross-validation performance 26](#_Toc204015650)

[FIG. 15 Cross-validation model performance 27](#_Toc204015651)

[FIG. 16 Base decision tree classifier performance 28](#_Toc204015652)

[FIG. 17 Base decision tree classifier confusion matrix 28](#_Toc204015653)

[FIG. 18 Tuned decision tree classifier performance 29](#_Toc204015654)

[FIG. 19 Tuned decision tree classifier confusion matrix 29](#_Toc204015655)

[FIG. 20 Base random forest classifier performance 30](#_Toc204015656)

[FIG. 21 Base random forest classifier confusion matrix 30](#_Toc204015657)

[FIG. 22 Tuned random forest classifier performance 31](#_Toc204015658)

[FIG. 23 Tuned random forest classifier performance 31](#_Toc204015659)

[FIG. 24 Performance of the tuned random forest classifier on balanced dataset 32](#_Toc204015660)

[FIG. 25 Confusion matrix for the tuned random forest classifier on balanced dataset 32](#_Toc204015661)

[FIG. 26 Base XGB classifier performance 33](#_Toc204015662)

[FIG. 27 Base XGB classifier confusion matrix 33](#_Toc204015663)

[FIG. 28 Tuned XGB classifier performance 34](#_Toc204015664)

[FIG. 29 Tuned XGB classifier confusion matrix 34](#_Toc204015665)

[FIG. 30 Feature importances for the tuned XGB classifier model 35](#_Toc204015666)

[FIG. 31 Performance of the tuned XGB classifier on the balanced dataset 36](#_Toc204015667)

[FIG. 32 Confusion matrix for the tuned XGB classifier on the balanced dataset 36](#_Toc204015668)

[FIG. 33 Feature importances for the tuned XGB classifier model on the balanced dataset 37](#_Toc204015669)

[FIG. 34 Individual model performance on training data 38](#_Toc204015670)

[FIG. 35 Individual model performance on validation data 39](#_Toc204015671)

[FIG. 36 39](#_Toc204015672)

[FIG. 37 Stacking classifier combination #1 41](#_Toc204015673)

[FIG. 38 Stacking classifier#1 performance 41](#_Toc204015674)

[FIG. 39 Stacking classifier #1 confusion matrix 41](#_Toc204015675)

[FIG. 40 Stacking classifier combination #2 43](#_Toc204015676)

[FIG. 41 Stacking classifier #2 performance 43](#_Toc204015677)

[FIG. 42 Stacking classifier #2 confusion matrix 43](#_Toc204015678)

[FIG. 43 Stacking classifier combination #3 45](#_Toc204015679)

[FIG. 44 Stacking classifier #3 performance 45](#_Toc204015680)

[FIG. 45 Stacking classifier #3 confusion matrix 45](#_Toc204015681)

[FIG. 46 Stacking classifier performance on training data 47](#_Toc204015682)

[FIG. 47 Stacking classifier performance on validation data 48](#_Toc204015683)

[FIG. 48 Performance on testing data – Stacking classifier #1 49](#_Toc204015684)

[FIG. 49 Confusion matrix for testing data – Stacking classifier #1 49](#_Toc204015685)

1. Project Notes 1
   1. Business Context

An E Commerce company or DTH (you can choose either of these two domains) provider is facing a lot of competition in the current market and it has become a challenge to retain the existing customers in the current situation. Hence, the company wants to develop a model through which they can do churn prediction of the accounts and provide segmented offers to the potential churners. In this company, account churn is a major thing because 1 account can have multiple customers. hence by losing one account the company might be losing more than one customer.

* 1. Business Objective

You have been assigned to develop a churn prediction model for this company and provide business recommendations on the campaign. Your campaign suggestion should be unique and be very clear on the campaign offer because your recommendation will go through the revenue assurance team. If they find that you are giving a lot of free (or subsidized) stuff thereby making a loss to the company; they are not going to approve your recommendation. Hence be very careful while providing campaign recommendation.

* 1. Data Landscaping
* AccountID - account unique identifier
* Churn - account churn flag (Target)
* Tenure - Tenure of account
* City\_Tier - Tier of primary customer's city
* CC\_Contacted\_L12m - How many times all the customers of the account has contacted customer care in last 12months
* Payment - Preferred Payment mode of the customers in the account
* Gender - Gender of the primary customer of the account
* Service\_Score - Satisfaction score given by customers of the account on service provided by company
* Account\_user\_count - Number of customers tagged with this account
* account\_segment - Account segmentation on the basis of spend
* CC\_Agent\_Score - Satisfaction score given by customers of the account on customer care service provided by company
* Marital\_Status - Marital status of the primary customer of the account
* rev\_per\_month - Monthly average revenue generated by account in last 12 months
* Complain\_l12m - Any complaints has been raised by account in last 12 months
* rev\_growth\_yoy - revenue growth percentage of the account (last 12 months vs last 24 to 13 month)
* coupon\_used\_l12m - How many times customers have used coupons to do the payment in last 12 months
* Day\_Since\_CC\_connect - Number of days since no customers in the account has contacted the customer care
* cashback\_l12m - Monthly average cashback generated by account in last 12 months
* Login\_device - Preferred login device of the customers in the account
  1. Data Overview
* The dataset consists of 11,260 rows and 19 columns.
  1. Descriptive statistics

1. Descriptive statistics of the data

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Column | count | mean | std | min | 25% | 50% | 75% | max |
| AccountID | 11260 | 25629.5 | 3250.63 | 20000 | 22814.75 | 25629.5 | 28444.25 | 31259 |
| Churn | 11260 | 0.17 | 0.37 | 0 | 0 | 0 | 0 | 1 |
| Tenure | 11042 | 11.03 | 12.88 | 0 | 2 | 9 | 16 | 99 |
| City\_Tier | 11148 | 1.65 | 0.92 | 1 | 1 | 1 | 3 | 3 |
| CC\_Contacted\_LY | 11158 | 17.87 | 8.85 | 4 | 11 | 16 | 23 | 132 |
| Service\_Score | 11162 | 2.9 | 0.73 | 0 | 2 | 3 | 3 | 5 |
| Account\_user\_count | 10816 | 3.69 | 1.02 | 1 | 3 | 4 | 4 | 6 |
| CC\_Agent\_Score | 11144 | 3.07 | 1.38 | 1 | 2 | 3 | 4 | 5 |
| rev\_per\_month | 10469 | 6.36 | 11.91 | 1 | 3 | 5 | 7 | 140 |
| Complain\_ly | 10903 | 0.29 | 0.45 | 0 | 0 | 0 | 1 | 1 |
| rev\_growth\_yoy | 11257 | 16.19 | 3.76 | 4 | 13 | 15 | 19 | 28 |
| coupon\_used\_for\_payment | 11257 | 1.79 | 1.97 | 0 | 1 | 1 | 2 | 16 |
| Day\_Since\_CC\_connect | 10902 | 4.63 | 3.7 | 0 | 2 | 3 | 8 | 47 |
| cashback | 10787 | 196.24 | 178.66 | 0 | 147.21 | 165.25 | 200.01 | 1997 |

* Only 17% of customers have churned, suggesting relatively good customer retention in the current service environment.
* Median tenure is 9 months, indicating that a significant portion of the customer base is relatively new.
* Most customers are from Tier 1 cities, which may reflect better service availability or urban-focused marketing strategies.
* Median of 16 contacts yearly shows moderate customer service engagement; some customers contact far more frequently.
* Median score of 3 suggests average service quality; room for improvement to enhance satisfaction.
* Most accounts are shared by 3–4 users, indicating multi-user subscriptions or family/shared plans.
* Median agent performance score is 3, showing average service from customer care representatives.
* Median monthly revenue is 5, but with high variance, indicating presence of both low and high-spending customers.
* Only 29% of customers have raised complaints, suggesting general satisfaction but room to reduce dissatisfaction further.
* Average growth of 16.19% shows steady improvement in business performance.
* Most customers use 1–2 coupons for payments, showing moderate usage of promotional offers.
* Median of 3 days suggests fairly recent engagement with customer care across users.
* Average cashback is around 196, indicating the business provides significant cashback, possibly to boost engagement or loyalty.
* Variables like ‘rev\_per\_month’ and ‘cashback’ have notable missing values, which may affect modeling and should be handled.
* Maximum values in ‘rev\_per\_month’ (140) and ‘cashback’ (1997) suggest the presence of outliers that may skew analysis.
  1. Exploratory data analysis
     1. Univariate analysis
        1. Churn

|  |
| --- |
| 1. Countplot for Churn |
|  |

* The Churn feature shows a significantly imbalanced distribution with approximately 83% observations in the favor of non-churning customers.
  + - 1. City tier

|  |
| --- |
| 1. Countplot for city tier vs. churn |
|  |

* Majority of the customers originate from tier 1 cities, with a significant difference between the churning and non-churning customers.
* Tier 3 cities have a high number of churning customers when compared to the total customers in the category.
  + - 1. Payment

|  |
| --- |
| 1. Countplot of payment vs churn |
|  |

* More than 50-60 % of the customers belong to the debit and credit card categories.
* Both of these categories have a significant distribution of churning customers.
  + - 1. Gender

|  |
| --- |
| 1. Countplot of genders vs churn |
|  |

* The distribution between the male and the female customers is slightly biased in the favor of male customers having the higher count.
* The original data contained of 4 classes with male and female categories named differently twice. We have renamed these classes to ensure uniformity by creating two classes ‘F’ for female and ‘M’ for male.
  + - 1. Service score

|  |
| --- |
| 1. Countplot for service score vs churn |
|  |

* Majority of the customers have given a satisfaction score of 3 on the basis of the services provided by the organization.
* Almost no customers have given a rating of 5 highlighting the urgent need for significant improvement in the quality of service being offered along with the customer support.
  + - 1. Account User Count

|  |
| --- |
| 1. Countplot of account segment vs churn |
|  |

* Close to 50% of the customers belong to the Super segment among which majority of them are non-churning customers.
* The customers in this variable have been segmented on the basis of their spending.
  + - 1. Customer complaint

|  |
| --- |
| 1. Countplot of customer complaint vs churn |
|  |

* Almost 30% of the customers registered a complaint in the last 12 months.
* Among which few customers have churned, indicating a substantial need for good quality customer service.
  + - 1. Tenure

|  |
| --- |
| 1. Histogram for tenure |
|  |

* The tenure feature is significantly right-skewed with the presence of some outliers in the data.
  + - 1. Customers contact with customer care

|  |
| --- |
| 1. Histogram for CC\_Contacted\_LY |
|  |

* The CC\_Contacted\_LY feature denotes the number of times the customers belonging to an account contacted customer care within the duration of 12 months.
* The data is highly right skewed with the presence of some outliers, with majority of the customers having contacted customer care for about 20-30 times.
  + - 1. Y-o-Y Revenue Growth

|  |
| --- |
| 1. Histogram for Y-o-Y Revenue Growth |
|  |

* The variable data depicts a left skewed distribution, with almost no values at the lower end of the data.
* This graph indicates that the revenue from the customers increases with time, but is almost negligent in the initial stages.
  + - 1. Account segments Vs Y-o-Y revenue growth

|  |
| --- |
| 1. Boxplots for account segments vs Y-o-Y revenue growth |
|  |

* The boxplots for the various account segments versus the Y-o-Y revenue growth show very minimal variation, with the means for each of the category around 15%.
* Only a few outliers are detected indicating no necessity for treatment.
  + - 1. Tenure Vs Y-o-Y revenue growth

|  |
| --- |
| 1. Boxplots for tenure vs Y-o-Y revenue growth |
|  |

* The significant extension of the lines towards the right end is due to the outliers in the data.
* Majority of the data is captured in the left portion of the lines as depicted in the histograms.
* Non-churning customers show a more stable revenue growth as compared to the churned customers.
  + 1. Multivariate Analysis
       1. Heatmap

|  |
| --- |
| 1. Heatmap |
|  |

* Only weak negative correlation (-0.23) exists between churn and tenure, implying longer-tenured users are slightly less likely to churn.
* Churn shows a weak positive correlation (0.25) with complaints, suggesting customers who complain more are slightly more likely to churn.
* Service score positively correlates (0.32) with account user count, indicating better service may be provided to accounts with more users.
* Coupon use correlates moderately (0.36) with time since last customer care contact, possibly reflecting re-engagement or incentive-driven communication.
* Tenure and days since last customer care contact show a weak positive correlation (0.12), suggesting older customers interact more recently.
* Churn has almost no correlation with most variables, suggesting churn may depend on latent factors not captured in this dataset.
* Revenue per month shows almost no correlation with other features, possibly indicating spending patterns vary independently of service metrics.
* Cashback has very weak correlations across the board, suggesting it doesn't directly relate to churn, complaints, or other key behaviors.
* City tier has negligible correlations, implying city category does not strongly influence customer behavior or service interaction metrics.
* Customer care contacted last year does not correlate strongly with churn or service score, indicating volume doesn't always reflect issues.
* Account ID has weak correlations with other variables, confirming it's a unique identifier and not a meaningful predictor.
  1. Removal of unwanted variables
* The ‘AccountID’ variable was removed as it was redundant.
  1. Missing value treatment

1. Missing values per feature

|  |  |
| --- | --- |
|  | Cross-Validation Cost |
| AccountID | 0 |
| Churn | 0 |
| Tenure | 218 |
| City\_Tier | 112 |
| CC\_Contacted\_LY | 102 |
| Payment | 109 |
| Gender | 108 |
| Service\_Score | 98 |
| Account\_user\_count | 444 |
| account\_segment | 97 |
| CC\_Agent\_Score | 116 |
| Marital\_Status | 212 |
| rev\_per\_month | 791 |
| Complain\_ly | 357 |
| rev\_growth\_yoy | 3 |
| coupon\_used\_for\_payment | 3 |
| Day\_Since\_CC\_connect | 358 |
| cashback | 473 |
| Login\_device | 221 |

* The above table depicts the number of missing values per variable in the dataset.
* The missing values for the categorical variables were imputed using the mode for the respective variable.
* The missing values for the continuous variables were imputed using the mean of the respective variable.
  1. Outlier treatment
* The ‘cashback’ and the ‘rev\_per\_month’ features were identified to have a significant number of outliers.
* The outliers for these variables were treated using the IQR method, wherein the outlier values were replaced with the nearest non-outlier observation in the respective variable.
  1. Business insights from EDA
* The dependent variable ‘Churn’ shows a significant imbalance of around 83% to 17% in the favor of non-churning customers.
* We will perform SMOTE to treat the data imbalance during the feature engineering phase, before the model building process.
* The model building process could also be performed on two different datasets including the original slightly imbalanced dataset and the balanced dataset (SMOTE) to compare the model performances, and evaluate the impact of data imbalance on various models.
* The imbalance further indicates a good factor for the business stating the most of the customers are loyal to the organization and only a few customers have churned over the given period.
* Our aim is to significantly reduce the number of churned customers number to ensure higher performance of the organization revenues.
* Most variables have low correlation with churn, indicating weak linear relationships.
* Linear models (like logistic regression) may not perform well without additional feature engineering.
* Churn prediction likely depends on non-linear patterns or interactions between multiple variables.
* Tree-based models (e.g., Random Forest, XGBoost) or neural networks are better suited for capturing complex relationships.
* Feature engineering (e.g., creating ratios, flags, or combined features) can improve model performance.
* Important churn drivers may be missing from the dataset (e.g., customer sentiment, competitor influence).
* Cross-validation and careful tuning are important to ensure robust model performance and avoid overfitting.
  1. Model Evaluation Criteria
* The model can make the following incorrect predictions:
  + Predicting that the customer will churn but in reality the customer will not.
  + **Predicting that the customer will not churn but in reality the customer will churn.**
* Since our objective is to **minimize the churn rate** for the organization, we will **focus more on the 2nd incorrect prediction.**
* Therefore, our **primary evaluation metric will be the Recall score**, i.e. higher the recall score, higher the chances of correctly predicting customers that are expected to churn.
  1. Pre-Defined Functions
* Pre-defined functions are functions used to automate repetitive commands passed during the model building and evaluation process.
* Some of the functions used in the model building and evaluation process in the project include **confusion matrix, model predictions, evaluation metrics, and performance comparison**.
  1. Model Building
     1. Cross-validation performance across multiple models

|  |
| --- |
| 1. Cross-validation performance |
|  |

* Cross-validation was performed across multiple models as shown in the figure above. Number of splits was set to 5, with the shuffle argument as True.
* The performance of the models across the scores are uniform, except the logistic regression model which displays a slight improvement in performance on the validation data.
* Based models were used for training and predictions without any hyperparameters.
* Additionally, the comparison of the models on the basis of its recall score is given in the figure below.
* XGBoost is returning the highest mean cross-validated recall followed by RandomForst, and DecisionTree
* XGBoost also has the highest recall score on the validation data set
* No major outliers are present, indicating consistency among the models
* We will further, hypertune and stack the models to enhance their performance

|  |
| --- |
| 1. Cross-validation model performance |
|  |

* 1. Individual Models
     1. Decision Tree Classifier
        1. Base model

|  |
| --- |
| 1. Base decision tree classifier performance |
|  |

|  |
| --- |
| 1. Base decision tree classifier confusion matrix |
|  |

* Our primary focus is on the recall score to properly predict the possibility of customers that will churn, and for this model the recall score is average.
* The model has overfit on the training data, which is a natural characteristic of a decision tree model without max\_depth defined.
* Although the model has a high accuracy of 93% on the validation data, the other metrics have average score. We can possibly improve the performance of the model through hypertuning.
  + - 1. Tuned decision tree classifier

|  |
| --- |
| 1. Tuned decision tree classifier performance |
|  |

|  |
| --- |
| 1. Tuned decision tree classifier confusion matrix |
|  |

* The tuned decision tree classifier has a weaker performance as compared to the base model.
* The recall score decreased by 8% in comparison to the base decision tree model, as well as, both the precision score significantly reduced to 52%.
  + 1. Random Forest Classifier
       1. Base model

|  |
| --- |
| 1. Base random forest classifier performance |
|  |

|  |
| --- |
| 1. Base random forest classifier confusion matrix |
|  |

* The base random forest model has a very high accuracy score of 96% and a precision score of 98%, but in the case of recall, the model has a recall score of 79% which is average.
* We can further tune the model to get better overall scoring performance.
  + - 1. Tuned random forest classifier

|  |
| --- |
| 1. Tuned random forest classifier performance |
|  |

|  |
| --- |
| 1. Tuned random forest classifier performance |
|  |

* We can see that the performance of the model has dipped significantly in comparison to the base model.
* The recall and the F1 score especially are significantly low in comparison to the base model.
* Although, we have hypertuned the model, it is possible that the slight imbalance in the dataset might be causing the low performance scores.
* Hence, we will try to train a tuned model on the balanced dataset to check for any improvements in the performance.
  + - 1. Tuned random forest classifier on balanced dataset

|  |
| --- |
| 1. Performance of the tuned random forest classifier on balanced dataset |
|  |

|  |
| --- |
| 1. Confusion matrix for the tuned random forest classifier on balanced dataset |
|  |

* A significant difference is seen across all metrics except the accuracy score between the performance of the model on the training data and the validation data.
* The model has generalized better on the validation data in terms of the recall and F1 scores of 78% and 77% percent respectively.
  + 1. XGBoost Classifier
       1. Base model

|  |
| --- |
| 1. Base XGB classifier performance |
|  |

|  |
| --- |
| 1. Base XGB classifier confusion matrix |
|  |

* Compared to all the other models so far, we can see that the XGBoost classifier has the best recall score of 82%.
* The model has also performed well across other metrics, with a high accuracy of 96% and a high precision of 93%.
  + - 1. Tuned XGB classifier

|  |
| --- |
| 1. Tuned XGB classifier performance |
|  |

|  |
| --- |
| 1. Tuned XGB classifier confusion matrix |
|  |

* The tuned version of the XGB classifier has almost similar performance to the base model with slight improvements in the precision and the F1 score.
* Overall, the tuned model is by far the best performing model for performing churn classification on our given dataset.

|  |
| --- |
| 1. Feature importances for the tuned XGB classifier model |
|  |

* Almost all the features played a significant role in the prediction of the target variable using the tuned XGB classifier model.
* The ‘tenure’ and the ‘complain\_ly’ feature contributed the most to the prediction of our target variable.
  + - 1. Tuned XGB classifier on the balanced dataset

|  |
| --- |
| 1. Performance of the tuned XGB classifier on the balanced dataset |
|  |

|  |
| --- |
| 1. Confusion matrix for the tuned XGB classifier on the balanced dataset |
|  |

* A 1% increase in recall from 82% in the hyper-tuned XGB model trained on the original data and the hyper tuned XGB model tuned on the balanced data.
* The rest of the performance metrics remain similar.

|  |
| --- |
| 1. Feature importances for the tuned XGB classifier model on the balanced dataset |
|  |

* In contrast to the previous feature importances for the hypertuned XGB on original training data, in the current case the ‘complain\_ly’ feature has the highest contribution to the prediction of the target variable.
  + 1. Performances of the individual models
       1. Training data

|  |
| --- |
| 1. Individual model performance on training data |
|  |

* Basic Decision Tree and Random Forest models show perfect performance across all metrics (Accuracy, Recall, Precision, F1), which is a strong indicator of overfitting, as such flawless performance is highly unlikely on real-world data.
* The Hypertuned Decision Tree significantly improves Precision (0.52) and F1-score (0.60) compared to an untuned model on possibly imbalanced data, but it still underperforms in Recall (0.71), indicating many false negatives.
* Hypertuned Random Forest shows a balanced improvement with 0.96 Accuracy and a high Precision (0.97), suggesting it is particularly good at correctly identifying positive predictions with fewer false positives.
* After applying data balancing to the Hypertuned Random Forest, Recall jumps to 0.95 from 0.80, indicating a notable reduction in false negatives and better handling of class imbalance.
* All XGBoost variants, including hypertuned and those trained on balanced data, show perfect training scores (1.0 across all metrics), suggesting extremely high model complexity and a risk of severe overfitting.
  + - 1. Validation data

|  |
| --- |
| 1. Individual model performance on validation data |
|  |

* The Decision Tree shows a notable drop in Recall (from 1.0 to 0.78) and Precision (from 1.0 to 0.81), revealing that it was likely overfitting on the training data and is unable to maintain that performance on unseen samples.
* Hypertuning the Decision Tree does not lead to major improvements on validation data, as seen by a Recall of 0.70 and a low Precision of 0.52, suggesting that even after tuning, the model struggles with false positives and false negatives.
* Random Forest generalizes well, maintaining a high validation Accuracy of 0.96 and strong Precision (0.98), though its Recall of 0.79 indicates it may still be missing some positives despite overall excellent predictive power.
* Hypertuning the Random Forest surprisingly reduces Recall (from 0.79 to 0.67) and slightly lowers F1-score, indicating that the tuning might have caused the model to become more conservative in its positive predictions.
* Applying data balancing to the Hypertuned Random Forest raises Recall to 0.78, suggesting that balancing helped recover some of the lost sensitivity, although Precision (0.75) took a hit due to more false positives.
* XGBoost models maintain high Accuracy (0.96) and F1-scores (~0.87–0.89) across all variants, showing strong generalization and robustness, unlike their overfitted training counterparts which scored perfect 1.0s.
* Hypertuned XGBoost improves Precision from 0.93 to 0.96 while maintaining a solid Recall (0.82), achieving a high F1 of 0.89—demonstrating that hyperparameter tuning enhanced both confidence in predictions and balanced performance.
  1. Stacking Classifiers
     1. Stacking classifier combination #1

|  |
| --- |
| 1. Stacking classifier combination #1 |
|  |
| 1. Stacking classifier#1 performance |
|  |
| 1. Stacking classifier #1 confusion matrix |
|  |

* The model is highly reliable in identifying negatives (Specificity is very high), given the very low false positive count.
* Overall balance in TP and TN shows the model is not biased toward any class, which is a good sign, especially if the dataset was imbalanced initially.
* Combining tree-based models captures both low-variance (Decision Tree) and low-bias (Random Forest) perspectives, offering a robust feature abstraction layer.
* Using XGBoost as the meta-learner is a powerful choice, as it can learn non-linear interactions from the outputs of the base models and refine them efficiently.
* Stacking enables leveraging complementary strengths of each model, helping improve generalization while mitigating individual weaknesses.
* Perfect training scores (1.00 across the board) suggest overfitting, especially since validation recall dropped to 0.86.
* Validation precision (0.92) remains high, indicating the model is confident and accurate in its positive predictions.
* An F1-score of 0.89 on validation shows a good balance, but the drop from perfect training metrics indicates that some generalization error exists.
  + 1. Stacking classifier combination #2

|  |
| --- |
| 1. Stacking classifier combination #2 |
|  |
| 1. Stacking classifier #2 performance |
|  |
| 1. Stacking classifier #2 confusion matrix |
|  |

* This stacking ensemble strategically combines a hypertuned Random Forest and an XGBoost model, with another XGBoost as the meta-learner, enabling layered learning of complex patterns.
* Using two gradient-boosted models (XGBoost as base and meta learner) may increase expressive power, but also raises the risk of overfitting if not properly regularized.
* The inclusion of a Random Forest trained on balanced data ensures that the model accounts for class imbalance, potentially improving minority class recall.
* Perfect training scores again indicate overfitting, as the model has learned the training data exactly, which often doesn't generalize well to unseen data.
* A validation accuracy of 96% is excellent and shows that despite overfitting in training, the model maintains strong generalization capabilities.
* A high precision of 0.91 on the validation set confirms that the model is highly confident in its positive predictions, with very few false positives.
* An F1-score of 0.88 on validation reflects a well-balanced trade-off between recall and precision, indicating the model handles both types of errors reasonably well.
  + 1. Stacking classifier combination #3

|  |
| --- |
| 1. Stacking classifier combination #3 |
|  |
| 1. Stacking classifier #3 performance |
|  |
| 1. Stacking classifier #3 confusion matrix |
|  |

* This stacking classifier combines a Hypertuned Decision Tree and a Hypertuned Random Forest, feeding into an XGBoost final estimator, making it a deep ensemble that learns both simple and complex decision patterns.
* While both base learners are tree-based models, their tuning allows diversity in decision boundaries, and XGBoost as the meta-learner helps refine predictions from the base models for better generalization.
* Training accuracy is 97%, indicating that the model fits the training data well but not perfectly, which can be a good sign for avoiding overfitting.
* The recall of 0.70 on validation shows that the model misses 30% of actual positive cases, which may be a serious concern in high-stakes classification problems.
* The high training precision of 0.98 and validation precision of 0.89 suggest that most predicted positives are correct, but fewer positives are detected overall.
* A validation F1-score of 0.78 reflects the imbalance between precision and recall, revealing that the model is skewed toward precision, potentially underperforming in identifying all true positives.
* The small drop between training and validation scores (e.g., F1 drops from 0.90 to 0.78) indicates moderate overfitting, but not as severe as in some previous stacking setups.
  + 1. Performances of the stacking classifiers
       1. Training data

|  |
| --- |
| 1. Stacking classifier performance on training data |
|  |

* Stacking Classifier 1
  + All training metrics being 1.0 suggests that the model has perfectly memorized the training data, which is a strong indication of overfitting.
  + Such perfect performance across accuracy, recall, precision, and F1-score implies very low bias but extremely high variance, which may hurt generalization.
* Stacking Classifier 2
  + Like Classifier 1, perfect training scores show the model has zero training error, pointing again to overfitting due to excessive model complexity.
  + This level of performance suggests the ensemble may be too finely tuned to training noise or class distributions and could fail in real-world settings.
* Stacking Classifier 3
  + Unlike the other two, this model has a training accuracy of 0.97 and F1-score of 0.90, suggesting it generalizes slightly better with some learning bias.
  + The recall of 0.83 shows the model doesn’t catch all positives even during training, which could be the result of a more conservative threshold.
    - 1. Validation data

|  |
| --- |
| 1. Stacking classifier performance on validation data |
|  |

* Stacking Classifier 1
  + With 0.97 accuracy and 0.88 recall, this model strikes a strong balance between correctly identifying both positive and negative classes on unseen data.
  + A validation precision of 0.95 shows that most of the predicted positives are truly positive, making the model highly dependable in high-precision applications.
  + The F1-score of 0.91 confirms the best harmonic mean among all models, indicating that Classifier 1 achieves the best overall validation performance.
* Stacking Classifier 2
  + Validation accuracy drops slightly to 0.96, and recall to 0.85, indicating this model misses more positives than Classifier 1 but still performs well overall.
  + Its precision of 0.91 suggests it maintains a high standard for positive predictions, but lower recall results in a lower F1-score of 0.88.
  + This classifier represents a safer balance but sacrifices some sensitivity, which may or may not be acceptable depending on the business context.
* Stacking Classifier 3
  + Validation recall falls significantly to 0.70, meaning the model misses nearly 30% of actual positives, which could be critical depending on the application.
  + The relatively lower precision of 0.89, while still decent, combined with low recall, leads to the weakest F1-score of 0.78 among all three models.
  + Although its accuracy is still strong at 0.94, this model’s conservative nature reduces its utility in domains where capturing positives is essential.
  1. Best Model – Performance on the Testing Data

|  |
| --- |
| 1. Performance on testing data – Stacking classifier #1 |
|  |

|  |
| --- |
| 1. Confusion matrix for testing data – Stacking classifier #1 |
|  |

* Given the business goal of maximizing customer retention through proactive churn intervention, Stacking Classifier #1 is the best choice. It achieves high recall, excellent precision, and generalizes strongly on unseen data.
* An accuracy of 98% on the test set indicates excellent overall classification ability, showing the model is successfully capturing both churners and non-churners in unseen data.
* A recall of 92% on test data means the model successfully identifies 92% of all actual churners, which is crucial in churn prediction, where missing potential churners can result in business loss.
* A precision of 95% ensures that most predicted churners are indeed real churners, meaning marketing and retention resources will be used effectively, avoiding unnecessary outreach.
* With a high F1-score, the model balances recall and precision extremely well, indicating it’s equally good at identifying churners and avoiding false alarms.
* Consistently strong performance on training, validation, and now test data demonstrates this stacking model’s robustness.
  1. Insights
* Customers who experience frequent signal interruptions or unresolved technical issues are significantly more likely to churn. This indicates a strong relationship between service quality and customer retention.
* Single-user or two-user accounts are more likely to churn, possibly due to a lack of shared cost benefit or emotional investment in the service, unlike larger family/group accounts.
* The model reveals spikes in churn after tariff increases, especially where no additional channels, features, or viewing quality improvements are offered in return.
* Accounts where subscribers view fewer hours of content per week or skip premium features (like on-demand or catch-up TV) are often those that churn next.
* Accounts that contact support multiple times in a short window or express dissatisfaction via calls or chat logs are highly likely to churn within 60 days.
  1. Recommendations
* Rather than giving flat discounts, offer retention packages that scale based on customer value and churn likelihood. High ARPU customers with medium churn risk could get content-based rewards (like free pay-per-view movie coupons) instead of monetary discounts.
* For accounts likely to churn due to travel or seasonal use, offer an option to pause the service for 1–3 months at a nominal hold fee instead of canceling. This keeps the account alive and reactivatable at minimal cost.
* Use past viewing data to offer free trial weekends of underutilized genres or language packs to low-engagement accounts. This boosts retention without permanent discounts and encourages content discovery.
* Instead of giving free months, offer customers the chance to lock in their current plan rate for 6–12 months if they commit now. This secures future revenue while appealing to price-sensitive users.
* Introduce group-based retention programs where if one user in a multi-user account refers another household/family to subscribe, they all get temporary add-on features. This leverages social stickiness rather than monetary cost.
* Assign service reps to call or message users predicted to churn soon and offer resolution for known complaints or conduct a quick satisfaction check. The cost is low, but perceived care significantly improves retention.
* The model predictions should alert customer support teams in real-time so they can intervene with at-risk accounts immediately post-complaint, preventing churn before it happens.
* Give users small, high-perceived-value bundles (e.g., 2 movie rentals + 1 month of regional pack) instead of cutting core plan prices. These feel rewarding while preserving pricing integrity.
* Build a churn-risk loyalty program where customers earn points for continued usage and on-time payments. Points can be redeemed for value-added content, not monetary benefits, protecting revenue margins.
* Since one account can have multiple users, campaigns should target overall household engagement. Encourage family watch parties, group contests, or family plans to build emotional and habitual attachment to the service.