A MAJOR PROJECT REPORT ON

AI Powered Telecom Customer Retention Prediction System

Submitted for Award of Internship Certified for

THE SKILL UNION

Under

VIHARA TECH

Submitted by

M. CHANDRA KANTH

UNDER THE ESTEEMED GUIDANCE OF

Mr. K SAI KAMAL

Assistant Professor

(B. Tech - AI/ML Eng., Data Scientist, CV Eng.)

THE SKILL UNION

DATA SCIENCE

THE SKILL UNION | VIHARA TECH

THE SKILL UNION | VIHARA TECH

VT Plaza, C/14, 4th Floor, Road No:1, KPHB, Kukatpally, Hyderabad, Telangana – 500072

Certificate

This is to certify that the Project Work entitled "**AI Powered Telecom Customer Retention Prediction System**" is a Bonafide record of the Industry Oriented Major Project Work submitted by

M. CHANDRA KANTH

Fulfillment of the requirements for the award of the Internship in **The Skill Union** under **Vihara Tech** during his/her course duration.

SIGNATURE OF VIHARA TECH TRAINER

Mr. K Sai Kamal

SIGNATURE OF VIHARA TECH MENTOR

Mr. Devaram Pranith

SIGNATURE OF THE SKILL UNION CEO

Mr. K Sai Krishna

DECLARATION

I, M. Chandra Kanth, hereby declare that this project report, titled **" AI Powered Telecom Customer Retention Prediction System ",** is the result of my original work completed during the 6-month IT / Programming Language training course at Vihara Tech. This project was executed over the full term, incorporating the practical knowledge gained during the four months of intensive classroom training and the real-world experience acquired during the subsequent two-month paid internship at The Skill Union. The information and findings presented herein are a true and accurate reflection of the work performed and have not been submitted, in whole or in part, for any other degree or diploma. I confirm that all sources of information have been specifically acknowledged.

Name	Batch / Course	Signature
M. CHANDRA KANTH	Data Science	

ACKNOWLEDGEMENT

The successful completion of this project and the valuable experience gained throughout the six-month training program would not have been possible without the unwavering support and invaluable contributions of several individuals. I extend my deepest gratitude to all those who guided me and facilitated my learning journey.

My sincere appreciation goes to **Mr. K Sai Kamal, Data Science & AI/ML Trainer**, for their exemplary commitment to education. Their invaluable expertise and clarity in delivering complex programming concepts formed the crucial technical foundation upon which this entire project rests. The dedication they showed in making the classroom sessions truly illuminating has fundamentally shaped my approach to software development.

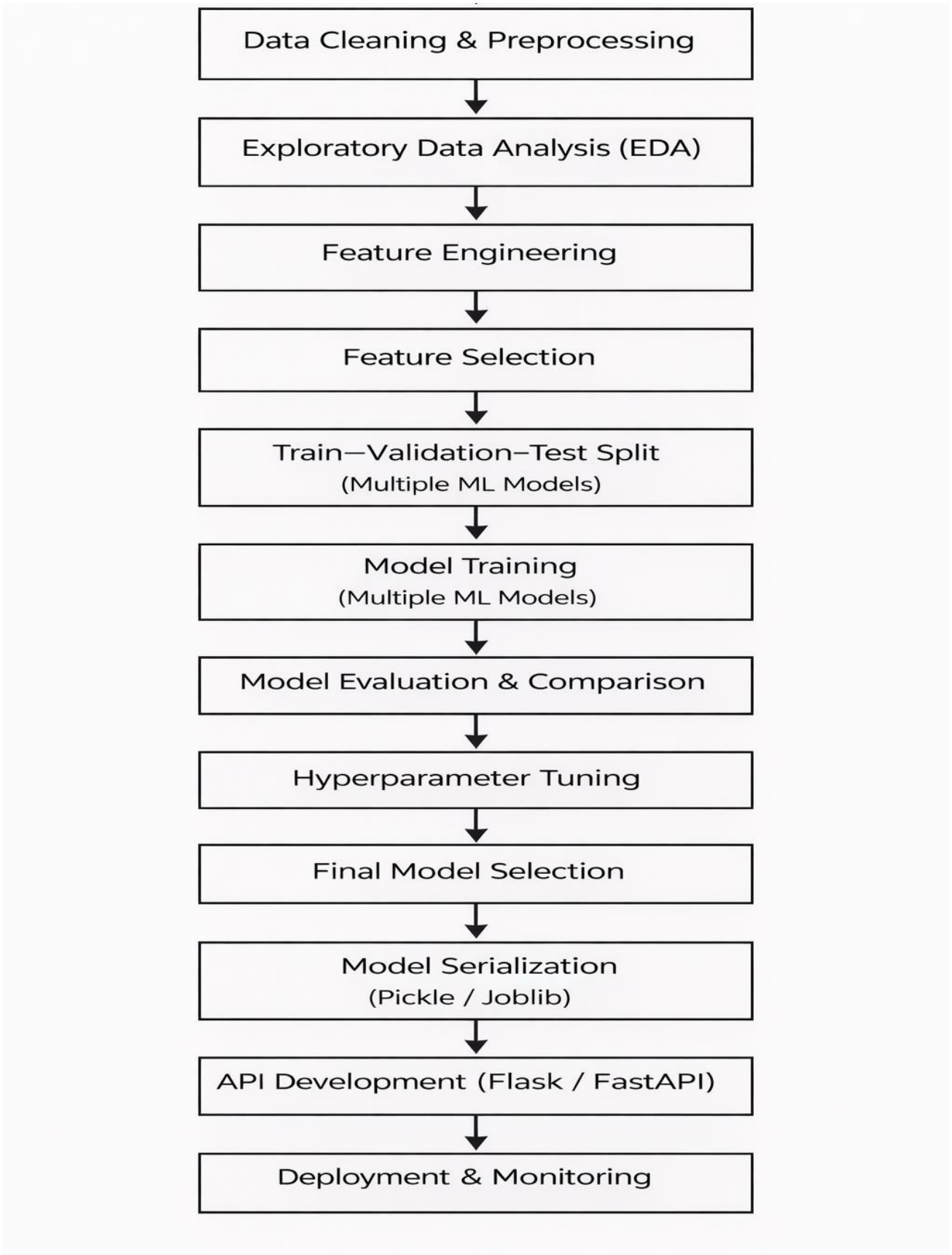
I am profoundly thankful to my dedicated Institute **Mentor, Ms. Tanya**. Their steadfast guidance during both the project development and the subsequent two-month internship was instrumental in bringing this report to fruition. Their insightful suggestions, technical foresight, and willingness to offer practical mentorship at critical junctures were truly inspiring and elevated the quality and direction of the work presented here.

Finally, I wish to acknowledge the visionary leadership of **Mr. K Sai Krishna, CEO of The Skill Union**. It is through their strategic stewardship that the Skill Union has created an outstanding environment, fostered a culture of excellence and provided career-defining opportunities like the one I have just completed. This initiative has been a powerful launchpad for my professional trajectory.

This project has been a significant milestone, and I am grateful for the collective efforts that made this comprehensive learning experience possible.

Table of Contents

ARCHITECTURE



ABSTRACT

Customer churn prediction is a critical task for organizations aiming to improve customer retention and reduce revenue loss. This project focuses on analyzing a customer churn dataset to identify key factors influencing customer attrition and to build effective predictive models. The dataset consists of customer demographic details, service usage patterns, and account-related attributes. Comprehensive data preprocessing techniques such as handling missing values, feature encoding, scaling, and class imbalance treatment are applied to ensure data quality. Exploratory Data Analysis (EDA) is performed to uncover meaningful trends and relationships between features and churn behavior.

Multiple machine learning classification models are trained and evaluated to predict customer churn accurately. Model performance is assessed using metrics such as accuracy, precision, recall, F1-score, and ROC-AUC. The results demonstrate that predictive analytics can effectively identify customers at high risk of churn, enabling organizations to take proactive retention measures and improve decision-making strategies.

Overall, this project demonstrates how machine learning can be effectively leveraged for predictive analytics in customer relationship management, providing a data-driven approach to enhance customer loyalty and organizational profitability.

INTRODUCTION

Customer churn prediction is a supervised machine learning problem that focuses on identifying customers who are likely to discontinue a service based on historical data. The dataset used in this project consists of structured customer information, including demographic attributes, service subscription details, usage behaviour, and billing-related variables. These features collectively capture patterns that influence customer retention and attrition.

The target variable in the dataset represents the churn status, making this a binary classification problem. Since the dataset contains both numerical and categorical features, preprocessing steps such as label encoding, one-hot encoding, feature scaling, and missing value treatment are applied to ensure compatibility with machine learning algorithms. Additionally, class imbalance is handled to improve model robustness and predictive performance.

Multiple machine learning models are trained and evaluated using this dataset, including Logistic Regression for baseline linear classification, Random Forest for capturing non-linear relationships and feature interactions, and XGBoost for gradient-boosted ensemble learning with high predictive accuracy. Model performance is assessed using metrics such as accuracy, precision, recall, F1-score, and ROC–AUC.

By leveraging this dataset and comparative model analysis, the project aims to identify key factors contributing to customer churn and build an effective predictive system that supports data-driven decision-making and customer retention strategies.

REQUIREMENTS FOR MODEL DEVELOPMENT

- **blinker** – Provides fast, simple object-to-object and broadcast signaling for Python applications.
- **click** – Used to create command-line interfaces (CLI) in a clean and composable way.
- **colorama** – Enables cross-platform colored terminal text output.
- **contourpy** – Handles contour generation for visualizations, used by Matplotlib.
- **cycler** – Helps in creating composable style cycles, mainly used with Matplotlib.
- **feature**_engine – Offers feature engineering and preprocessing transformers for machine learning.
- **Flask** – A lightweight web framework for building web applications and APIs.
- **fonttools** – A library for manipulating fonts and related binary data.
- **gunicorn** – A WSGI HTTP server for running Python web applications in production.
- **imbalanced-learn** – Provides tools to handle imbalanced datasets using over/under-sampling techniques.
- **imblearn** – Wrapper package that redirects to imbalanced-learn.
- **itsdangerous** – Used for securely signing data in web applications (e.g., Flask session tokens).
- **Jinja2** – A fast, powerful templating engine for rendering HTML templates in Flask.
- **joblib** – Provides lightweight pipelining and parallel computing utilities for Python.
- **kiwisolver** – A fast implementation of the Cassowary constraint solver used in layout calculations (e.g., Matplotlib).
- **MarkupSafe** – Escapes characters in strings to make them safe for HTML/XML output.
- **matplotlib** – The primary Python library for creating static, animated, and interactive visualizations.
- **numpy** – The core library for numerical computing and handling large multidimensional arrays.
- **packaging** – Tools for parsing and handling package versioning and dependency specifications.
- **pandas** – A powerful library for data manipulation and analysis with DataFrames.
- **patsy** – Describes statistical models and builds design matrices for statsmodels.

- **pillow** – The Python Imaging Library (PIL) fork for opening, manipulating, and saving image files.
- **pyparsing** – A library for constructing and executing grammars used in text parsing.
- **python-dateutil** – Extends the datetime module for flexible date and time parsing.
- **pytz** – Enables accurate and cross-platform time zone calculations.
- **scikit-learn** – A comprehensive library for machine learning algorithms and data preprocessing.
- **scipy** – Provides scientific computing tools including statistics, optimization, and integration.
- **seaborn** – A high-level statistical data visualization library built on top of Matplotlib.
- **six** – Provides compatibility utilities between Python 2 and 3 codebases.
- **statsmodels** – Used for performing statistical tests, regression, and data exploration.
- **threadpoolctl** – Controls the number of threads used by native libraries in Python.
- **tzdata** – Provides timezone database information for systems that lack it.
- **Werkzeug** – A WSGI utility library that powers Flask's request and response handling.
- **xgboost** – An optimized gradient boosting framework for high-performance machine learning

1. DATA VISUALIZATIONS

1.1. Visualization Overview:

- In this project I used matplotlib and seaborn library for the data visualization to understand and interpret the data.
- It can be installed using the command

  pip install matplotlib
- Visualizations used in the matplotlib library are:

  => Bar chart

  => Pie chart

  => Horizontal bar chart
- For the statistical analysis of the data, I used seaborn library to learn more about the data using charts and graphs.
- It can be installed using the command

  pip install seaborn
- Visualizations used in seaborn library for data analysis are:

  => Hist plot

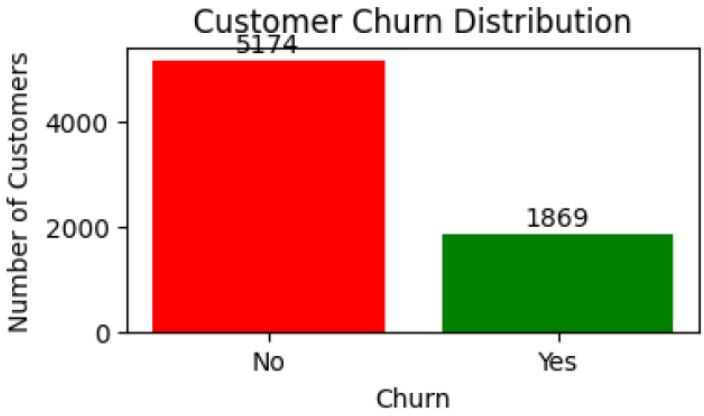
  => Bar plot

  => Count plot

  => Box plot
- I also used pandas for Data manipulation and preprocessing and NumPy libraries for numerical Operations and to perform scientific calculations.

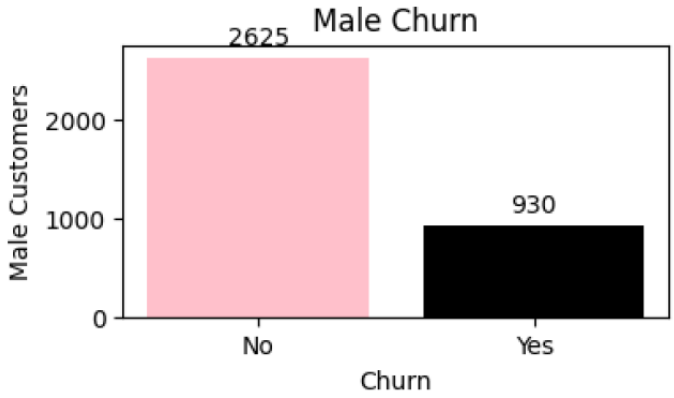
1.2. Customer Churn Distribution:

- Bar chart (Churn: Yes / No)
- Compares customers who churned vs those who stayed.
- Majority of customers are retained (No).
- Even a smaller churn percentage can lead to major revenue loss due to the large customer base.



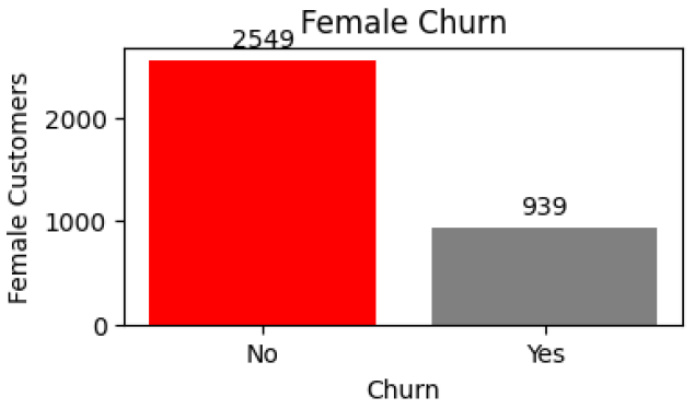
Male Churn Analysis:

- Bar chart (Male – Churn Yes / No).
- Churn behaviour specifically for male customers.
- More males are retained than churned.
- Male customers may respond differently to offers or pricing.
- Gender alone is not a strong churn predictor.



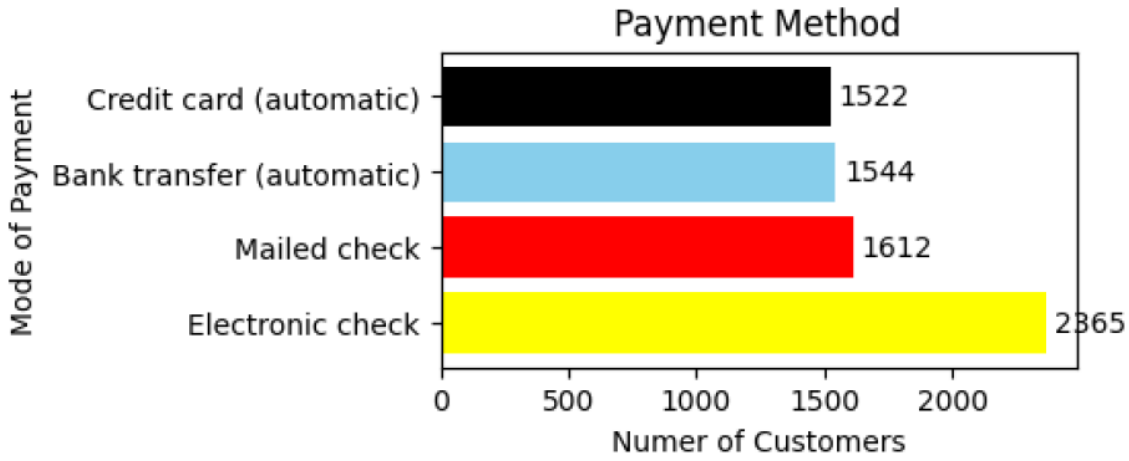
Female Churn Analysis:

- Bar chart (Female – Churn Yes / No).
- Churn behaviour among female customers.
- Similar to males, retention is higher than churn.
- Female churn count is close to male churn count.
- Churn is not heavily gender-biased**.**
- Gender alone may not be a strong predictor but still relevant in combination.



1.3. Payment Method Distribution:

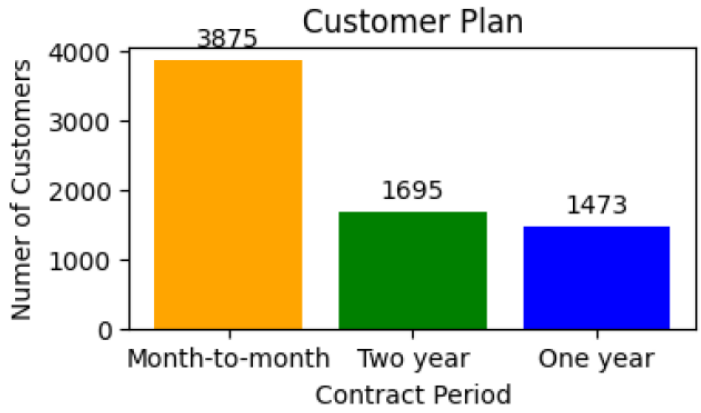
- Horizontal bar chart.
- Number of customers using different payment methods.
- Electronic check is the most used payment method.
- Automatic payments (bank transfer, credit card) are lower.
- Customers using manual payment methods are often linked to higher churn.



1.4. Contract Type (Customer Plan):

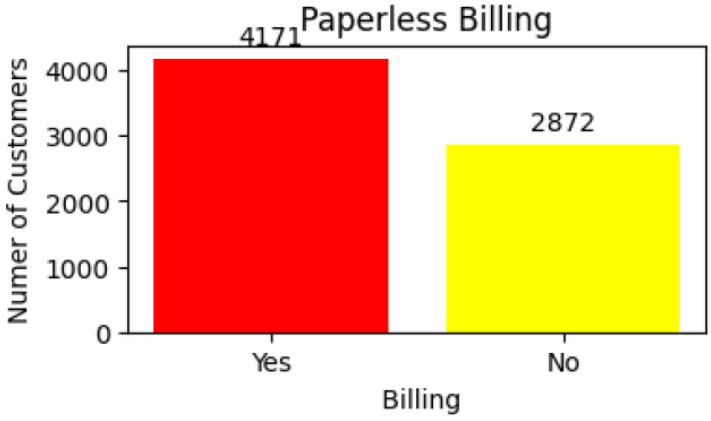
- Bar Chart
- Customers by contract duration.
- Month-to-month contracts dominate.
- Long-term contracts are fewer.

- Long-term contracts strongly improve retention.



1.5. Paperless Billing:

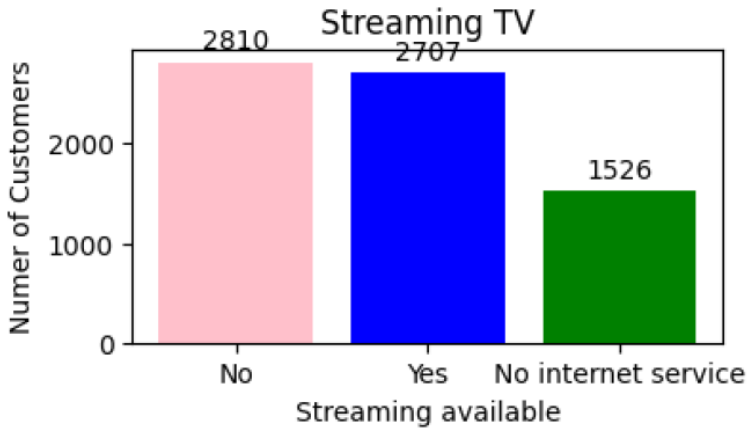
- Bar chart
- Customers using paperless vs non-paperless billing.
- Most customers prefer paperless billing.
- Digitally engaged customers are higher

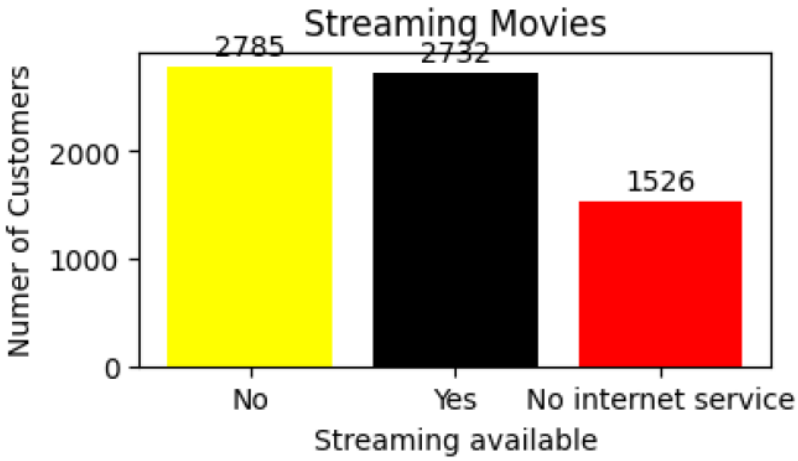


1.6. Streaming Services (TV & Movies):

- Bar Charts
- Customers with Streaming TV, without it, or without internet service.
- Almost equal number of users have and don’t have streaming TV.

- Bundling streaming may reduce churn.
- Customers without streaming services are slightly higher.
- Cross-selling entertainment services can increase engagement and retention.

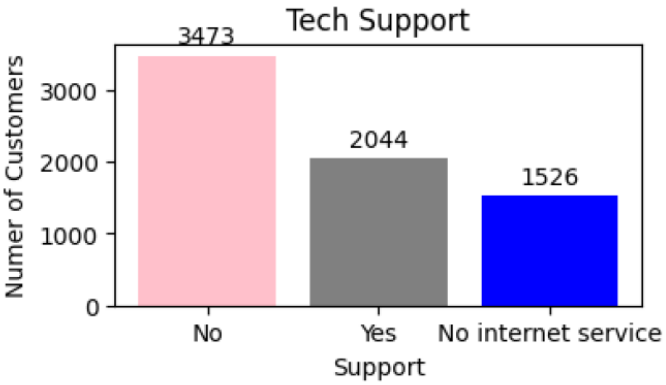


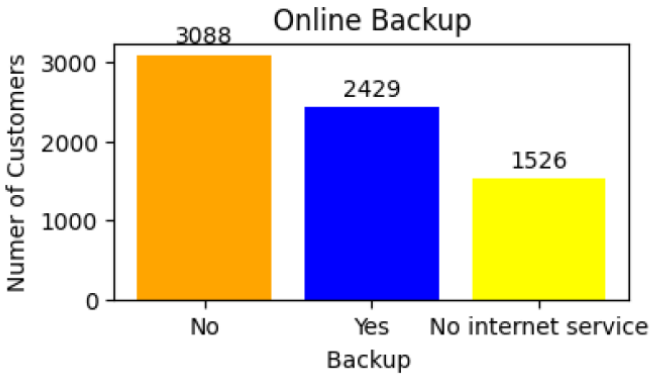


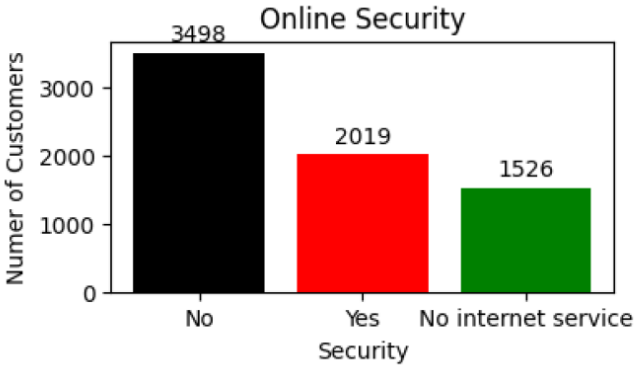
1.7. Tech Support, Online Security, Backup & Protection:

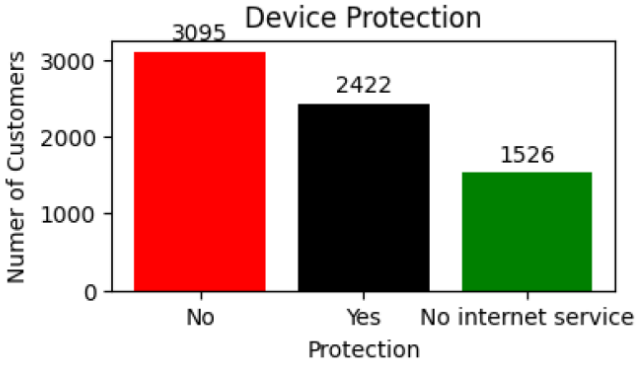
- Bar Charts
- Usage of value-added services.
- A large number of customers do not opt for add-on services.
- Customers with more services tend to stay longer.
- Many customers do not have tech support.
- Offering affordable tech support plans may improve retention.
- Customers without protection may churn after device issues.

- Security-conscious customers are usually long-term users.



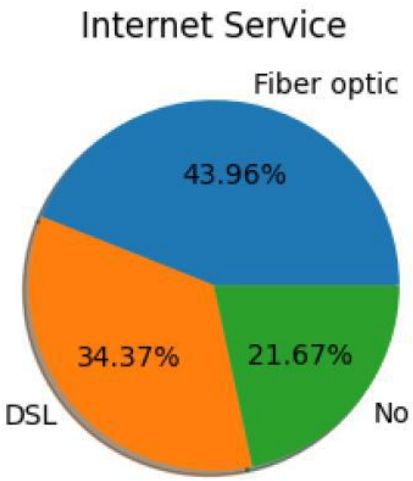


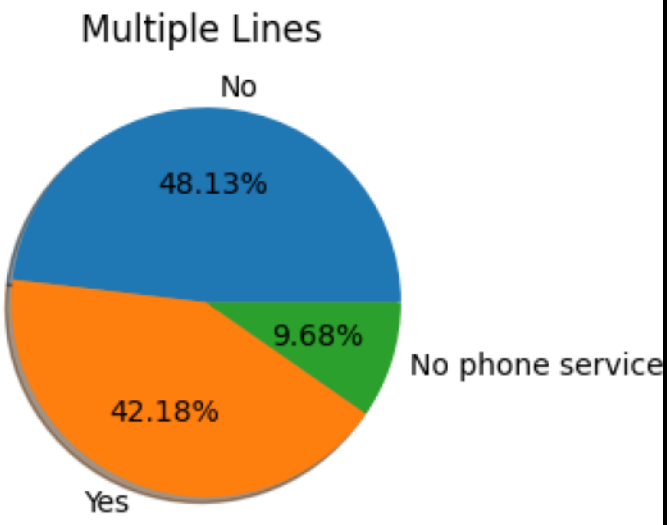




1.8. Internet Service & Multiple Lines:

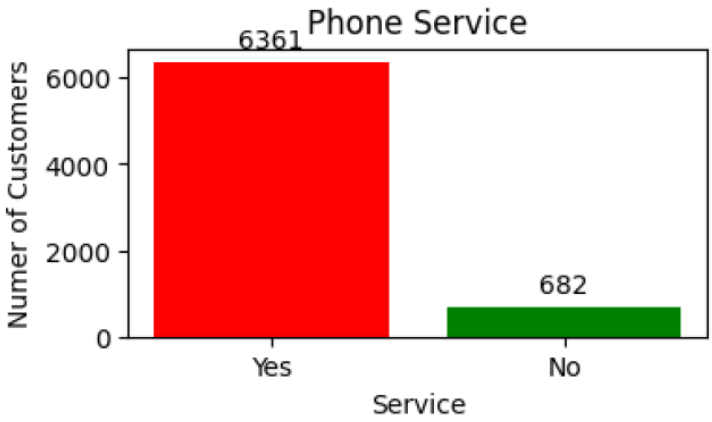
- Pie Charts
- Proportion of customers by internet service and phone line usage.
- Distribution across internet service types.
- Internet type directly impacts service usage and churn.
- Multiple-line customers are more invested.
- Fiber users usually show lower churn.





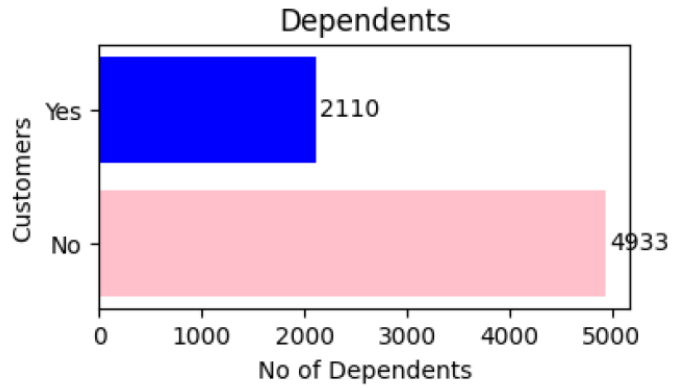
1.9. Phone Service:

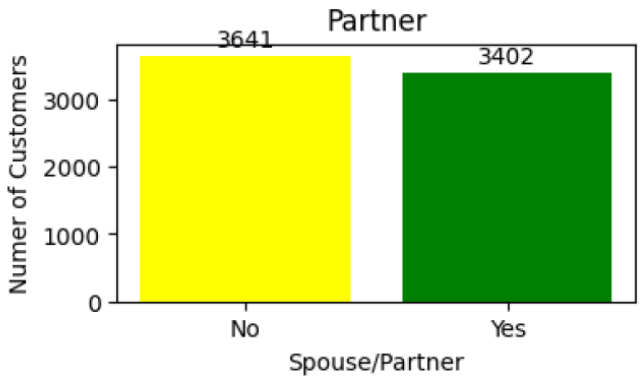
- Bar chart
- Customers with and without phone service.
- Vast majority have phone service.
- Phone service is core; churn depends more on add-ons.



1.10. Partner & Dependents:

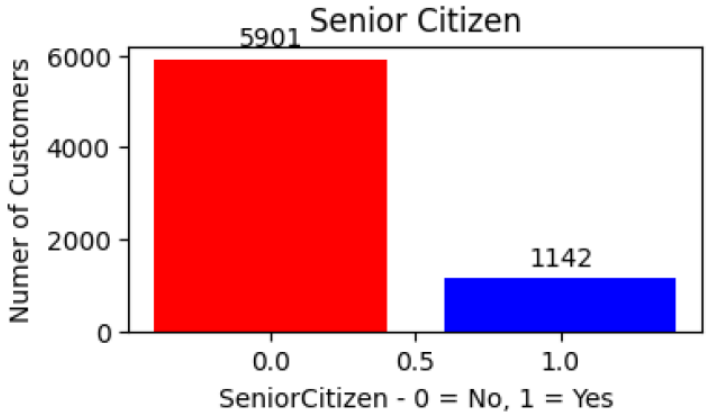
- Bar Charts
- Family status of customers.
- Customers with and without dependents.
- Customers with partners.
- Partnered customers are generally more stable.
- Customers with dependents tend to churn less.
- Family-based plans can be designed to improve retention.





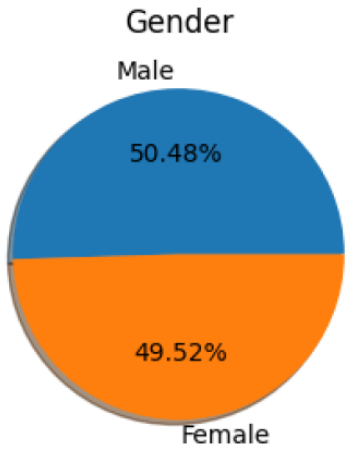
1.11. Senior Citizen Analysis:

- Bar Chart
- Senior vs non-senior customers.
- Senior citizens are fewer.
- Age-specific plans may help retention.
- Targeted plans for senior citizens may reduce churn.
- Seniors often have different churn drivers.



1.12. Gender Distribution:

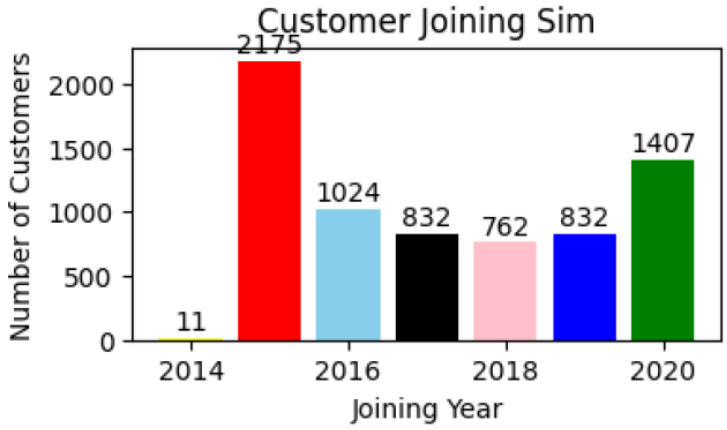
- Pie Chart
- Overall gender proportion.
- Almost balanced gender representation.
- Gender should be combined with other features for prediction.
- Male Customers - 50.48%
- Female Customers - 49.52%



1.13. Customer Joining Year:

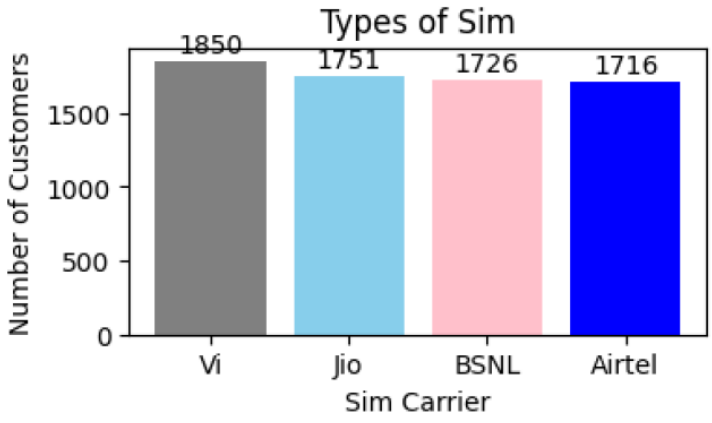
- Bar chart
- Certain years saw higher customer acquisition.

- Customer acquisition varies by year.
- Helps evaluate marketing effectiveness over time.
- Helps track churn trends by joining period.
- Useful for cohort analysis.



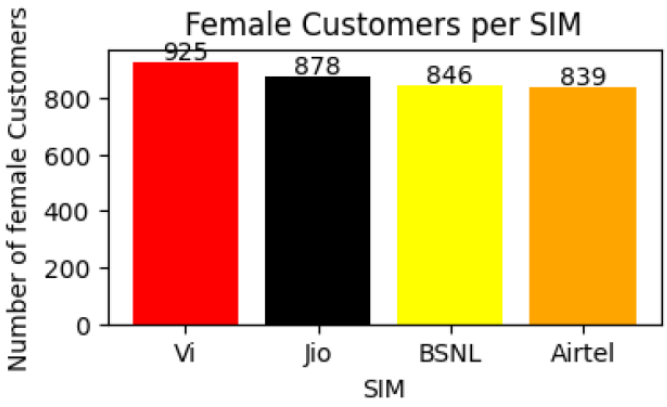
1.14. SIM-wise Customer Distribution:

- Bar Chart
- Customer count per SIM provider.
- Some SIM providers have higher customer bases.
- Provider-wise strategy optimization is possible.
- Useful for targeted retention strategies.



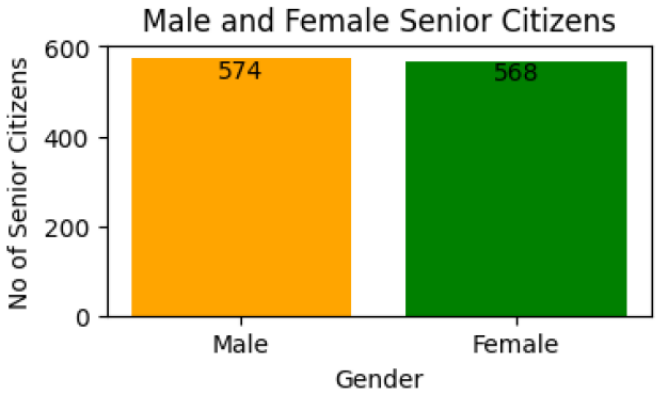
1.15. Male & Female Customers per SIM:

- Bar Charts
- Gender distribution across SIMs.
- Gender distribution is consistent across providers.
- No gender bias in SIM preference.



1.16. Senior Citizens by Gender:

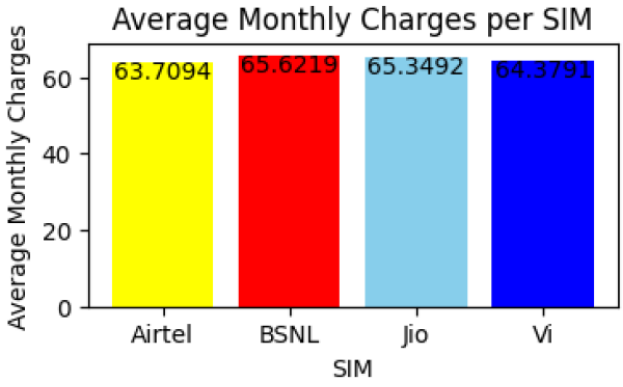
- Male vs female senior citizens.
- Slight gender variation among seniors.



1.17. Average Monthly Charges per SIM:

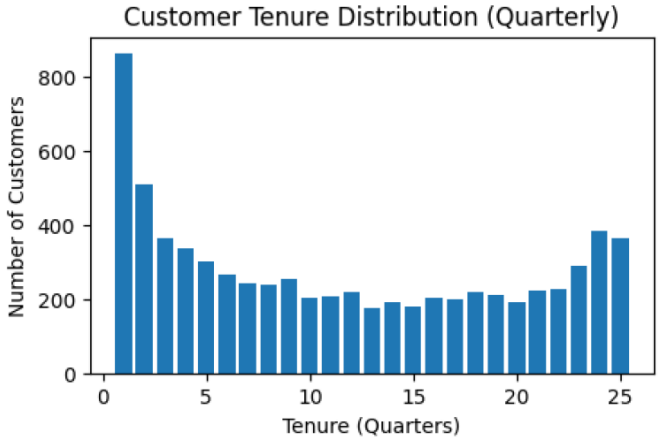
- Bar Chart
- Average billing per SIM provider.
- Certain SIMs generate higher revenue.
- Pricing differs across providers.

- High-charge SIMs may have higher churn risk.



1.18. Quarterly Tenure Distribution:

- Bar chart
- Customer tenure grouped by quarters.
- Most churn occurs in early quarters.
- First year is critical for churn prevention.
- Early-stage retention strategies are critical.
- Many customers are in early quarters.



- So, from the above visualizations there are several reasons for the customers to leave the company as mentioned in each visualization.

1.19. Final Observations from the Visualizations:

- This Exploratory Data Analysis (EDA) reveals important behavioral and demographic insights into the Telco Customer Churn dataset.

  The analysis shows that:

- The company has a balanced gender ratio, but senior citizens represent a smaller segment.
- Customers without dependents form the majority, possibly leaving the company.
- Most customers have short tenures, hinting at potential churn issues during the early months.
- Month-to-month contracts dominate, suggesting that long-term engagement strategies could reduce churn.
- Payment preferences differ slightly by gender, giving marketing teams opportunities for personalized engagement.
- By visualizing these aspects, the company can identify risk segments, improve retention strategies, and design targeted offers to enhance customer satisfaction and reduce churn.
- The visual exploration reveals that contract type, tenure, payment method, and billing type are the most influential features in predicting churn.
- High-paying, short-term, digitally active customers tend to churn more frequently, while customers with longer contracts, automatic payments, and family commitments show greater retention.
- These findings guide feature selection**,** model development, and business strategy, ensuring both data-driven predictions and actionable insights for customer retention.

2. FEATURE ENGINEERING:

- Feature Engineering is a critical stage in the machine learning pipeline where raw data is transformed into structured and meaningful features that improve model learning, generalization, and predictive performance.
- After completing Exploratory Data Analysis (EDA)**,** the dataset was passed into the data preprocessing phase to assess data quality and completeness. An initial missing value audit indicated no explicit null values across most features.
- The dataset was then partitioned based on feature types into numerical and categorical variables to enable appropriate preprocessing strategies. During this step, missing values were identified in the Total Charges feature due to incorrect type casting, where numeric values were stored as an object data type.
- To resolve this issue, the Total Charges column was converted to a numerical format, and missing value imputation techniques were evaluated. Based on performance considerations and data distribution characteristics, the most suitable imputation strategy was selected and applied to ensure data consistency and pipeline stability.

2.1. Handling Missing Values:

I. Random Sample Imputation (Best Approach):

- Random Sample Imputation is a technique used to handle missing values by replacing them with randomly selected existing values from the same variable in the training dataset.
- This approach helps maintain the original shape, mean, and variance of the data distribution, making it a better alternative to mean or median imputation, which can distort the feature's natural pattern.
- After applying random sample imputation, the standard deviation shows a large change compared to the original data, indicating that the variability of the feature is not well preserved. Therefore, this method was not chosen for handling missing values in the dataset.

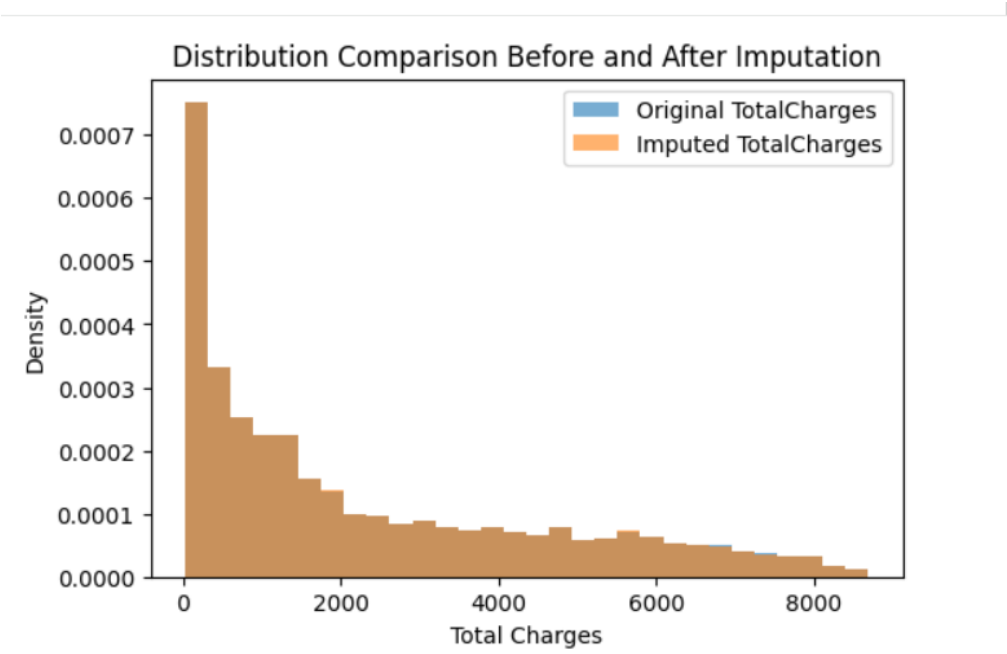


Fig.1 Random sample imputation

II. Mean/Median/Mode:

- The Mean, Median, Mode Imputation method replaces missing values with a representative statistic of that feature (column).
- Replace missing values with the mean (average) of the non-missing values in that column.
- Replace missing values with the median (the middle value when data is sorted).

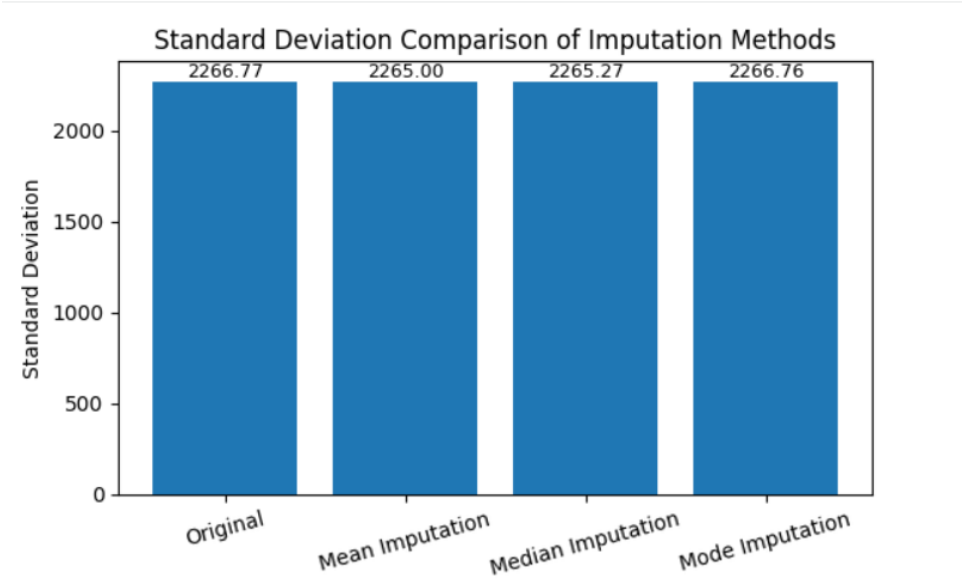


Fig.2 Mean -Median-mode imputation

- Replace missing values with the mode — the most frequent value in the column.
- By applying these imputations there is a much difference in standard deviation compared to the original data.

III. Drop Missing data:

- Dropping missing data (also called deletion) is one of the simplest ways to handle missing values.
- Instead of filling or estimating them (like in mean or KNN imputation), we simply remove the rows or columns that contain missing values.
- By applying this technique, we can lose the important rows or columns if we use this technique.
- This technique is helpful when there are fewer missing values. If the missing values are less than 5% then we can use this technique which does not affect the data loss.

IV. Constant, Arbitrary, End of Distribution Imputation:

- In constant imputation missing values are replaced with a fixed constant (e.g. 0, -1, or a predefined value).
- In arbitrary imputation missing values are replaced with an arbitrary value outside the normal data range.
- In end of distribution imputation missing values are filled with extreme values (e.g. max + 1 or min - 1).

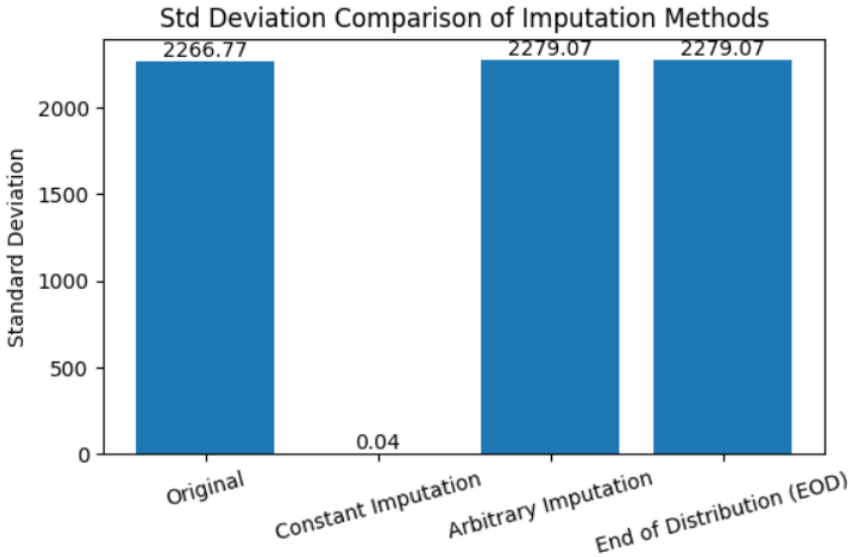


Fig.3 Constant-Arbitrary-EOD imputation

2.2. Data Separation:

- In a churn prediction dataset, features typically consist of a combination of numerical and categorical variables.
- Prior to applying preprocessing steps such as scaling, encoding, and imputation, it is essential to identify and separate features based on their data types.
- After handling missing values using appropriate imputation techniques, the dataset was systematically divided into categorical and numerical feature sets.
- Machine learning algorithms operate on numerical representations, as they cannot directly interpret categorical or textual data.
- Since numerical and categorical features capture information in fundamentally different ways, they require distinct preprocessing strategies before model training.
- This separation was efficiently performed by inspecting feature data types and using the select_dtypes () function in pandas to isolate numerical and categorical columns.

2.3. Variable Transformation:

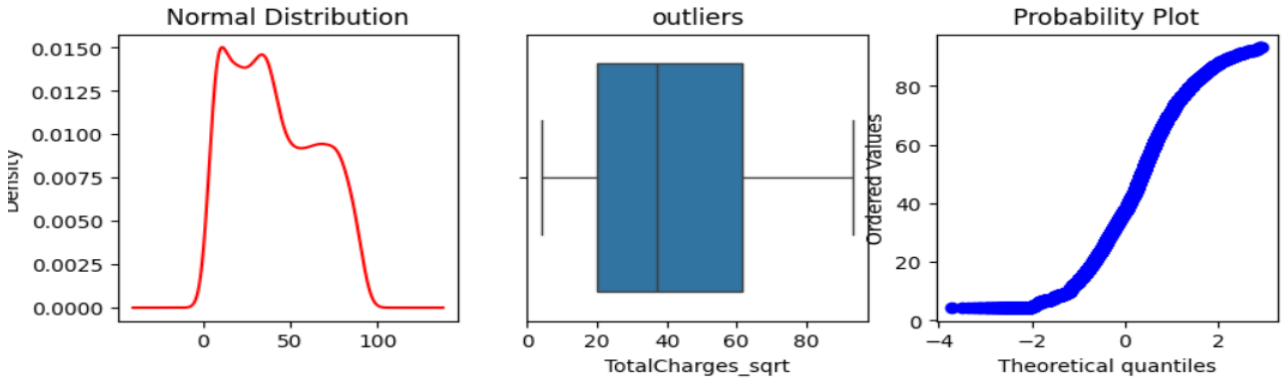
- Variable transformation (also referred to as feature transformation) is a preprocessing step in the machine learning pipeline that modifies feature values to better align with the assumptions of statistical and machine learning models.
- Many machine learning algorithms, such as linear regression and logistic regression, perform optimally when input features follow a normal or near-normal distribution; transformations help in achieving this by reducing skewness.
- Feature transformations help mitigate the influence of extreme values and outliers, thereby stabilizing model learning.
- Properly transformed variables enhance the predictive performance of churn prediction models and contribute to more reliable and interpretable business insights.
- The following section discusses the feature transformation techniques applied during this project to improve data quality and model effectiveness.
- Let’s see some of the techniques that I have worked on:

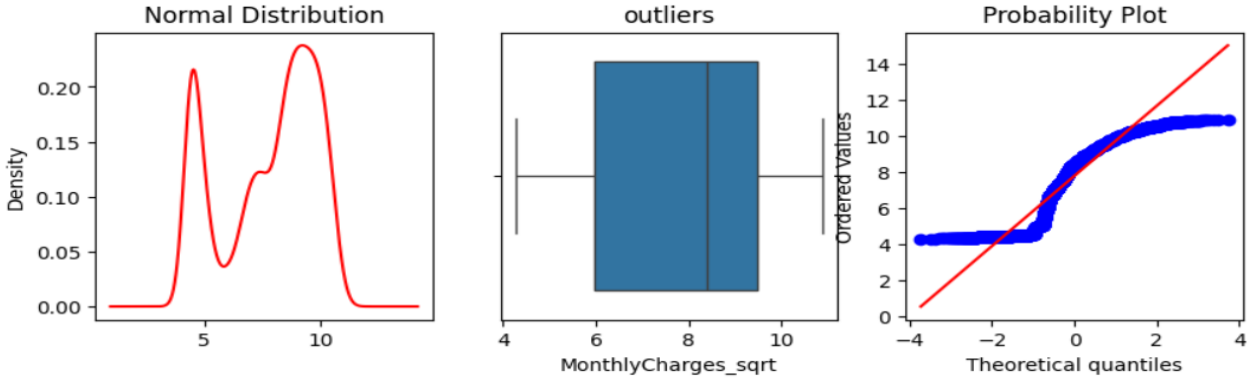
I. Power Transformation:

- Power transformation is a mathematical preprocessing technique used to reduce skewness, stabilize variance, and make feature distributions more closely resemble a normal (bell-shaped) distribution**.**
- The transformation applies a power function of the form $x' = x^{\lambda}$, where **λ** (lambda) determines the strength and nature of the transformation.
- In practical machine learning workflows, the value of λ is automatically estimated by the algorithm (e.g., Box-Cox or Yeo-Johnson methods) to maximize normality rather than being manually selected.
- After applying the power transformation, the resulting distribution did not sufficiently approximate a normal distribution. Additionally, the corresponding probability (Q–Q) plots did not indicate a good fit to normality.
- Due to the limited improvement in distributional characteristics, this transformation technique was not selected for the final preprocessing pipeline.

II. Arcsin (Square root) Transformer:

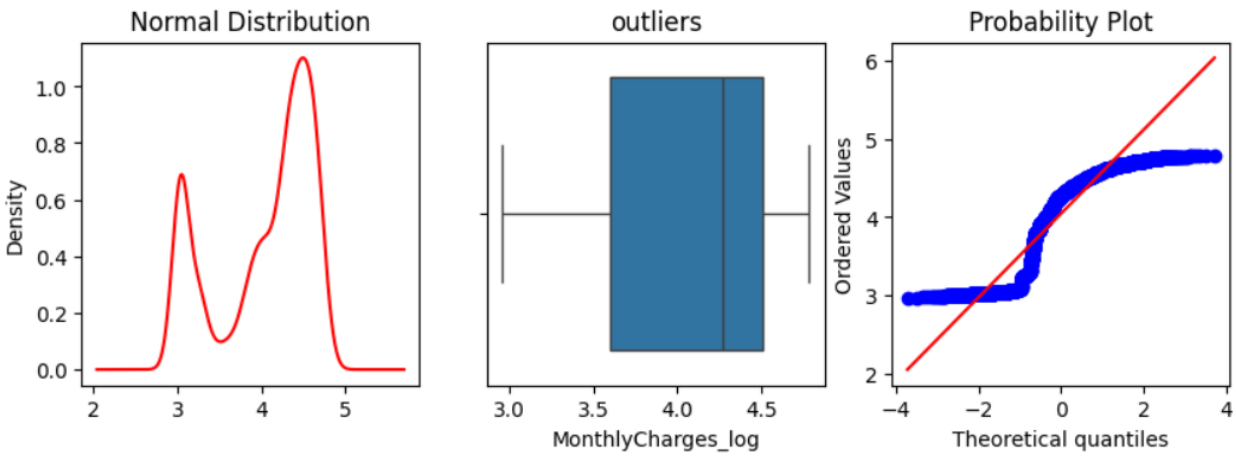
- The ArcsinTransformer () applies the arcsine square-root transformation to numerical variables as part of the feature transformation stage.
- This transformation, also known as the angular transformation, is define as
  $x' = \arcsin(\sqrt{x})$,
  where the input variable $x$must be constrained to the range [0, 1]**.**
- The ArcsinTransformer () is applicable only to numerical features whose values represent proportions or probabilities; variables with values outside this range will result in a transformation error.
- In this project, the transformation did not produce a meaningful improvement in the feature distribution or model readiness based on the dataset characteristics.
- Consequently, the arcsine transformation was excluded from the final model-building pipeline**.**

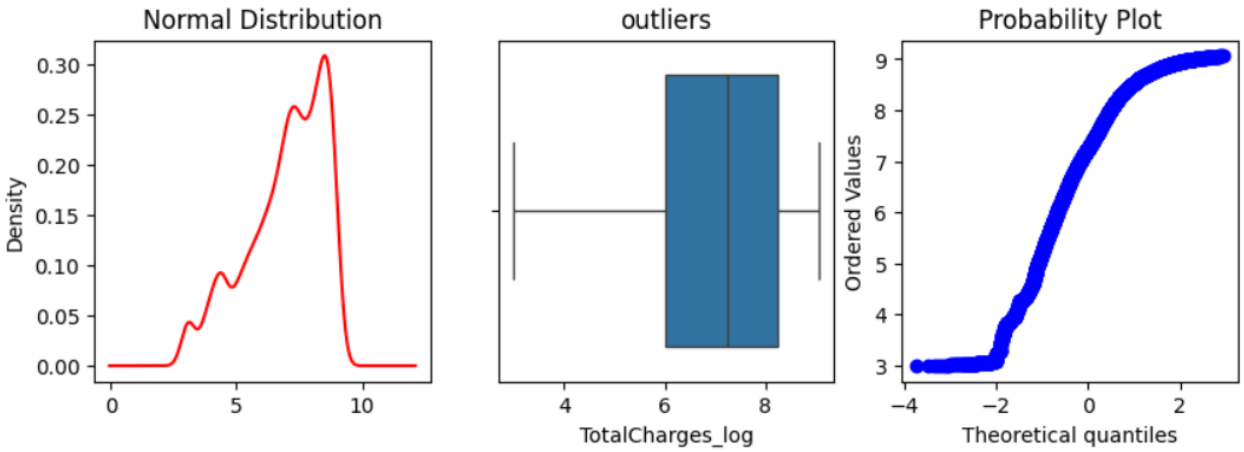




III. Log Transformer:

- The LogTransformer () applies a logarithmic transformation (either natural logarithm with base *e* or base-10 logarithm) to numerical variables to reduce skewness and compress large values.
- This transformer is applicable only to strictly positive values**;** variables containing zero or negative values will result in a transformation error unless pre-processed beforehand.
- The transformer allows either explicit specification of target variables or automatic selection of all numerical features for transformation.
- In this project, the logarithmic transformation did not yield the expected improvement in feature distribution or model performance given the nature of the data.
- As a result, the log transformation technique was not included in the final model-building pipeline**.**





IV. Box Cox Transformer:

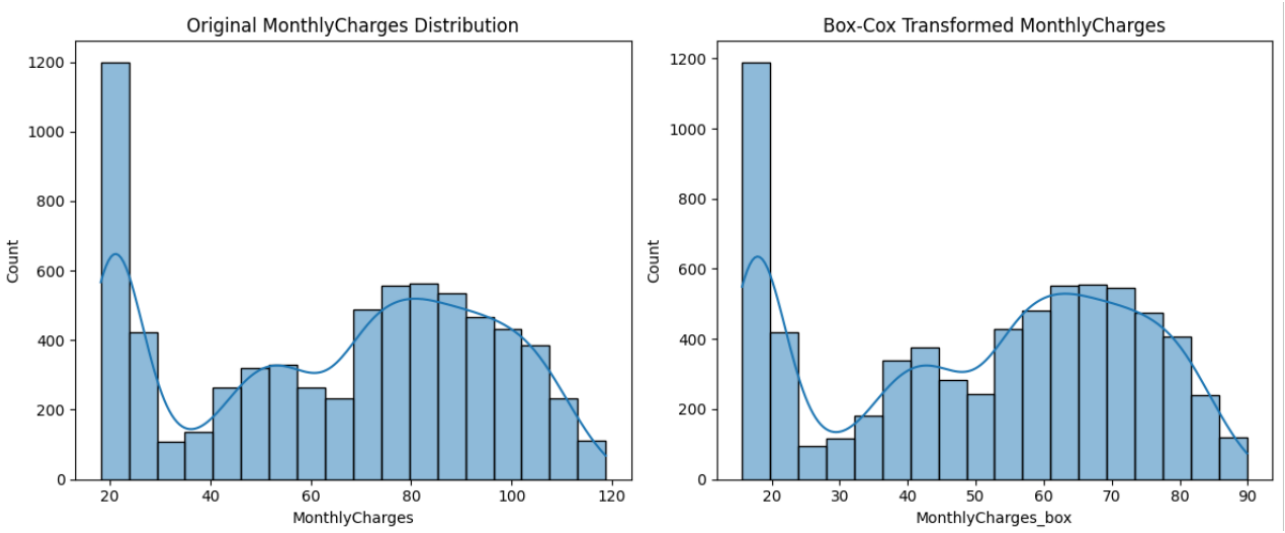
- The Box–Cox transformation is mathematically defined as:

$$T(Y) = \begin{cases} \dfrac{Y^{\lambda} - 1}{\lambda}, & \lambda \neq 0 \\ \log(Y), & \lambda = 0 \end{cases}$$

where Yrepresents the response variable and $\lambda$is the transformation parameter.

- The parameter $\lambda$is typically searched over a predefined range (commonly from −5 to +5), and the optimal value is automatically selected for each variable to maximize normality.
- This transformer operates only on positive numerical variables, as the Box–Cox transformation is undefined for zero or negative values.

- After applying the Box–Cox transformation, the resulting feature distributions did not adequately approximate a normal (bell-shaped) distribution, and the corresponding probability (Q–Q) plots did not show a satisfactory fit.
- Therefore, due to the lack of significant distributional improvement, the Box–Cox transformation was excluded from the final preprocessing and model-building pipeline.



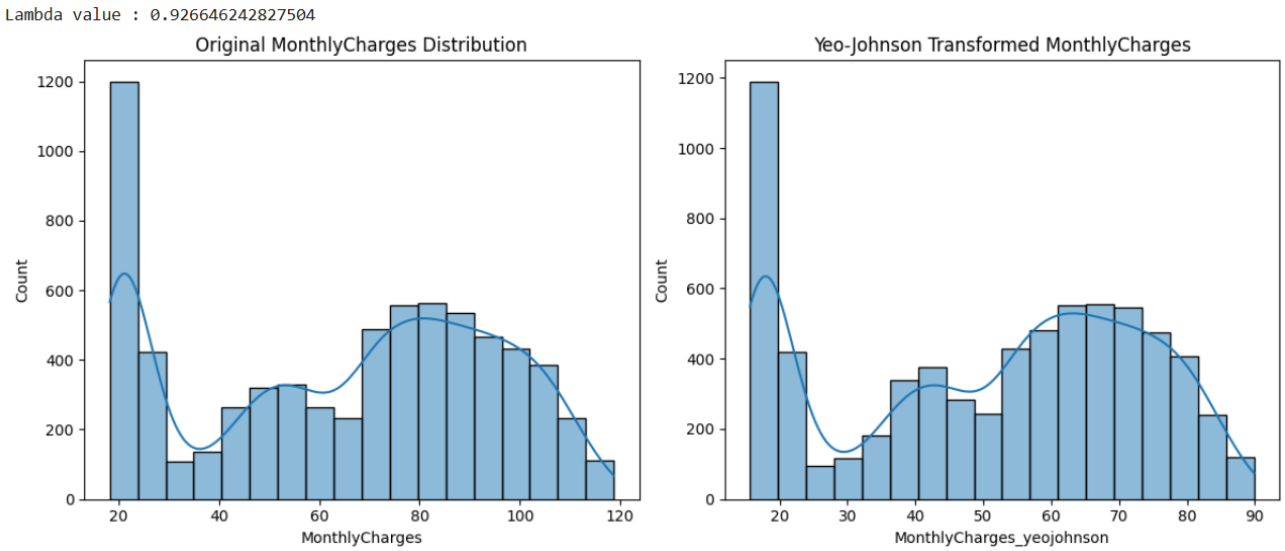
V. Yeo–Johnson Transformation (Best Approach):

- The Yeo–Johnson transformation is a power-based, parametric feature transformation technique used to reduce skewness and improve the normality of numerical features. It is an extension of the Box–Cox transformation and is specifically designed to handle both positive and negative values, making it more flexible for real-world datasets.
- Unlike Box–Cox, which requires strictly positive inputs, Yeo–Johnson can be applied to datasets containing zero or negative values without requiring any data shifting. The transformation uses a parameter λ(lambda) to determine the strength and form of the transformation, which is automatically estimated from the data using maximum likelihood estimation.

Mathematical Formulation:

For a feature value x, the Yeo–Johnson transformation is defined as:

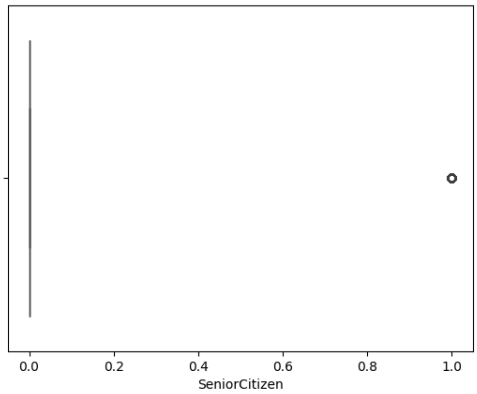
$$y = \begin{cases} \dfrac{(x+1)^{\lambda} - 1}{\lambda}, & \text{if } x \geq 0, \lambda \neq 0 \\ \log(x+1), & \text{if } x \geq 0, \lambda = 0 \\ \dfrac{-((-x+1)^{2-\lambda} - 1)}{2-\lambda}, & \text{if } x < 0, \lambda \neq 2 \\ -\log(-x+1), & \text{if } x < 0, \lambda = 2 \end{cases}$$

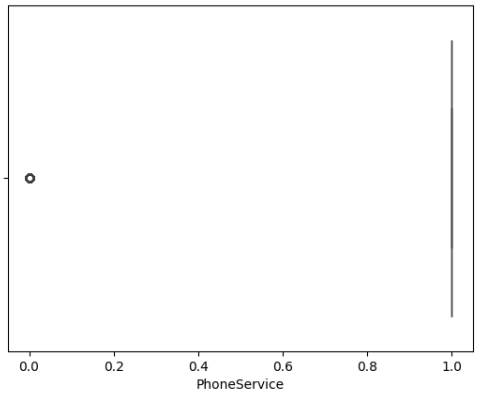


2.4 Handling Outliers:

- Outlier handling is a critical preprocessing step that ensures data consistency and improves the reliability and stability of machine learning models.
- After transforming the data to approximate a normal distribution, the dataset was analysed for the presence of outliers. As observed from the box plot visualizations, several features contained extreme values. Outliers are observations that deviate significantly from the majority of the data points.
- Such extreme values can skew feature distributions, inflate variance, and negatively influence model learning, particularly for algorithms sensitive to scale and distribution.
- To mitigate these effects and enhance model robustness, outlier treatment techniques such as Arbitrary Outlier Capping, Winsorization, and log-based transformations were applied to the affected numerical features.

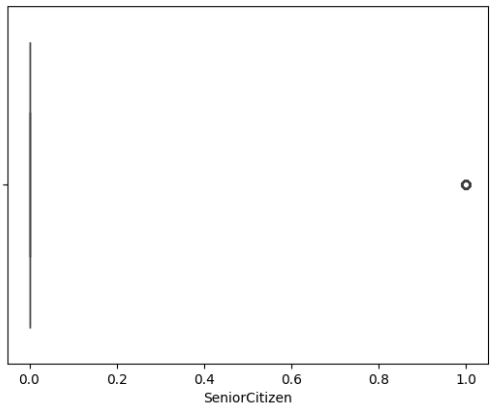
- These techniques effectively reduce the influence of extreme values while preserving the overall data structure, resulting in more stable and reliable churn prediction performance.

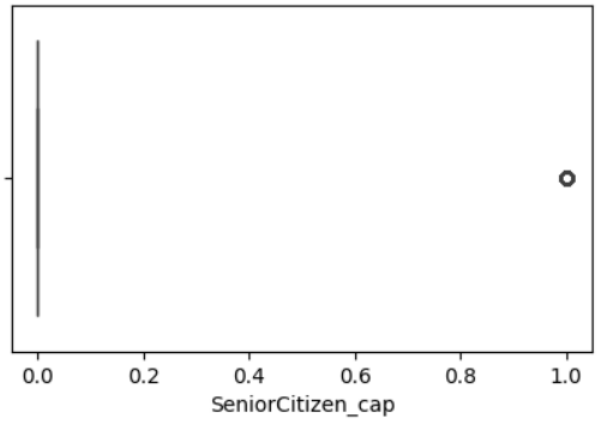




i. Capping:

- The Outlier Capping is used to control extreme values in numerical features that can adversely affect model performance, training stability, and prediction accuracy**.** Instead of removing outlier observations, this technique caps them at predefined lower and upper threshold values, thereby retaining all data points while limiting their influence.
- Values falling below the lower cap are replaced with the lower threshold, while values exceeding the upper cap are replaced with the upper threshold.
- Although this approach is simple to implement, it is not data-driven and relies heavily on manually selected threshold values.
- If thresholds are poorly chosen, the method may over-cap or under-cap feature values, leading to information loss and distorted feature distributions.
- Due to these limitations and the potential risk of introducing bias, the Outlier Capping was not adopted in the final model-building.





ii. Trimming (Best Approach):

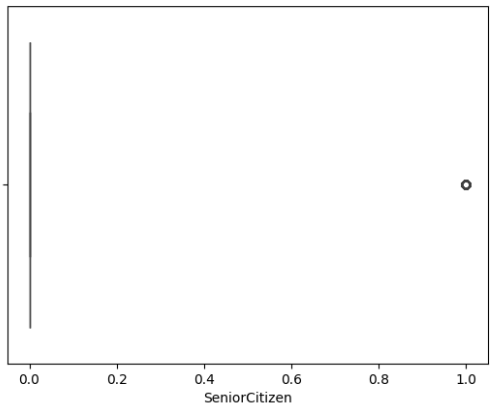
- The Outlier Trimming technique handles extreme values by removing entire records from the dataset when feature values fall outside predefined statistical limits.
- One commonly used approach is the Interquartile Range (IQR) method, which removes observations lying outside the range:

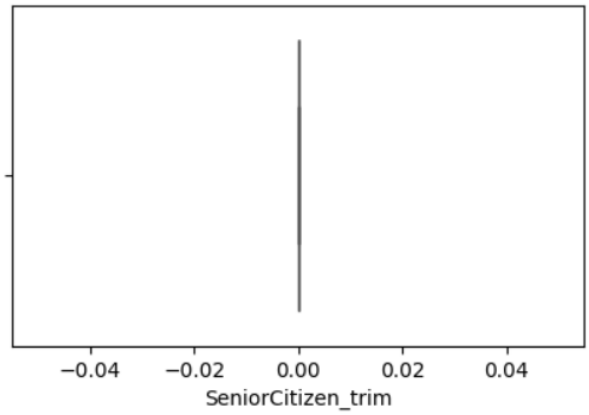
$$[Q1 - 1.5 \times IQR,\ Q3 + 1.5 \times IQR]$$

- Another approach is the Z-score method, where observations are removed if the absolute standardized score exceeds a specified threshold:

$$|\ Z\ | > \text{threshold}$$

- While trimming can reduce the influence of extreme values, it eliminates entire data points, which may contain meaningful or rare but valid information—particularly important in churn analysis.
- Excessive trimming can lead to data loss and biased samples, negatively affecting model generalization.
- Due to these limitations and the risk of losing valuable observations, the Outlier Trimming technique was not selected for the final model-building pipeline.





iii. 5th and 95th Quantile:

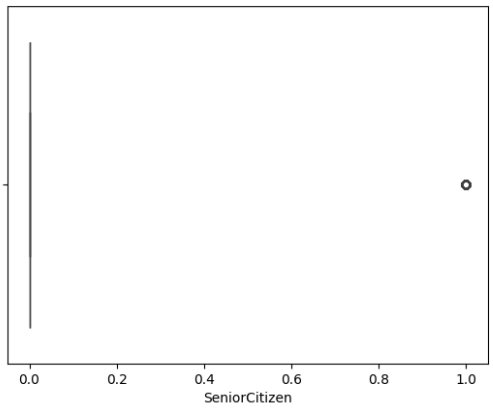
The 5th–95th quantile method is a percentile-based outlier handling technique used during the data preprocessing stage of a machine learning pipeline. Instead of removing extreme observations, this method caps extreme values at statistically determined percentile boundaries, thereby reducing their influence while preserving all data points.

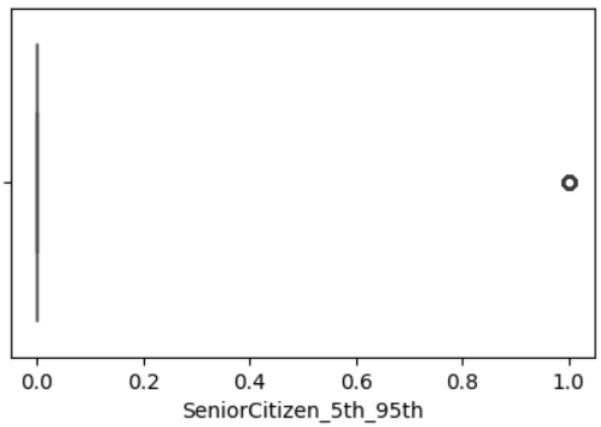
In the 5th–95th quantile method:

- The 5th percentile (P5) represents the value below which 5% of the data lies.
- The 95th percentile (P95) represents the value below which 95% of the data lies.

$$P5=Quantile(X,0.05)$$

$$P95 = \text{Quantile}(X, 0.95)$$





3. FEATURE SELECTION

3.1. Categorical Encoding:

- Feature selection is the process of identifying and retaining the most relevant features that significantly contribute to predicting customer churn, while eliminating redundant or non-informative variables.
- After completing the feature engineering stage, the workflow progressed to the feature selection phase of model development.
- As a prerequisite to feature selection, all categorical features were encoded into numerical representations, since machine learning models operate exclusively on numerical inputs.
- For categorical feature encoding:
- One-Hot Encoding was applied to nominal variables with no inherent order.
- Ordinal Encoding was used for ordinal variables that possess a meaningful ranking.
- When the target (dependent) variable was categorical, Label Encoding was applied to convert class labels into numerical form.
- These encoding strategies ensure that categorical information is correctly represented while preserving semantic meaning, enabling effective feature selection and model training.

1. One Hot Encoding:

- Among the categorical features, the following variables were identified as nominal (i.e., having no inherent order):
- gender, SeniorCitizen, Partner, Dependents, PhoneService, MultipleLines, InternetService, OnlineSecurity, OnlineBackup, DeviceProtection, TechSupport, StreamingTV, StreamingMovies, PaperlessBilling, PaymentMethod.
- For nominal categorical variables with N unique categories, One-Hot Encoding was applied, which generates N binary (dummy) features, each representing a distinct category.

- In the encoded representation, each dummy feature takes a value of:
  1 -> if the observation belongs to that category
  0 -> otherwise
- Since each observation can belong to only one category at a time, exactly one dummy variable is set to 1 per row, which is why the method is referred to as “one-hot” encoding.

2. Ordinal Encoding:

- Among the categorical features, the Contract column was identified as the only ordinal variable, as it has a meaningful and inherent order among its categories.
- Ordinal Encoding converts each category into an integer value that reflects its relative order or rank, ensuring that the ordinal relationship is preserved.
- Ordinal Encoding is appropriate when:
- The categorical variable exhibits a clear and logical ordering.
- The relative magnitude between categories carries meaningful information for the model.
- Common examples of ordinal data include:
- Education Level: High School < Bachelor < Master < PhD
- Satisfaction Level: Poor < Average < Good < Excellent
- Size: Small < Medium < Large
- Since the order between contract types conveys important information, Ordinal Encoding was applied to the Contract feature to ensure the model correctly captures these relationships.

3. Label Encoding:

- Label Encoding is a categorical encoding technique that converts textual or categorical values into numeric labels, where each unique category is assigned an integer value.
- In customer churn prediction, Label Encoding is commonly applied to the target (dependent) variable when it is categorical, enabling machine learning algorithms to process class labels effectively.

- Label Encoding is not intended for nominal input features, as it does not preserve any meaningful order and may unintentionally introduce a false ordinal relationship.
- For example, in a binary target variable:
  Gender → Male = 0, Female = 1 (used only when Gender is the target variable).
- For nominal input features without inherent order, One-Hot Encoding is preferred to avoid misleading the model.

3.2. Filter Methods:

- Filter methods are one of the primary feature selection techniques used in machine learning to identify relevant features prior to model training**.**
- These methods evaluate the importance of each feature using statistical measures, independent of any specific machine learning algorithm.
- In simple terms, filter methods act like a sieve, removing irrelevant or weakly informative features while retaining those that contribute meaningful information.
- Filter methods are computationally efficient and help reduce noise, dimensionality, and overfitting risks.
- In this project, the following two filter-based techniques were applied:

i. Constant Technique:

- Variance Threshold is a filter-based feature selection technique that removes irrelevant or non-informative features based on the variance of each independent variable before model training.
- Features with zero variance contain the same value across all observations and therefore do not contribute any useful information for learning patterns.
- Such constant features are automatically removed, as they have no impact on model performance and may introduce unnecessary noise into the dataset.
- Applying variance-based filtering helps reduce dimensionality and improves training efficiency without affecting predictive capability.

ii. Quasi Constant Technique:

- Variance Thresholding is a filter-based feature selection technique that removes irrelevant or low-information features prior to model training by evaluating the variance of each independent numerical feature.
- Features with very low variance (for example, below a predefined threshold such as 0.1) contribute minimal information, as their values show little to no variation across observations, and therefore have negligible impact on model learning.
- During analysis, all numerical features were evaluated, and none exhibited variance values of 0 or below the selected threshold (0.1). Hence, no features were removed using variance-based filtering in this model.
- The variance of a dataset is mathematically defined as:

$$\mathrm{Var}(X) = \frac{1}{n-1}\sum_{i=1}^{n}(x_i - \bar{x})^2$$

where:

- $x_i$represents each individual data point,
- $\bar{x}$denotes the mean of the data, and
- $n$is the total number of observations.
- Based on this evaluation, variance thresholding was not applied in the final feature selection pipeline.

3.3. Hypothesis Testing:

- Hypothesis testing is a statistical approach used to analyse relationships within data, identify significant features, and validate assumptions during the feature selection process.

- In machine learning, hypothesis testing techniques help determine whether observed patterns are statistically meaningful or occurred by chance.

- Common hypothesis testing and statistical analysis techniques include the Z-test**,** T-test**,** ANOVA (Analysis of Variance)**,** Chi-Square test**,** and Correlation analysis for identifying relationships between variables.
- In this project, the following hypothesis testing and statistical techniques were applied to evaluate feature significance and extract meaningful insights:

1) Anova-Test:

- ANOVA (Analysis of Variance) is a statistical hypothesis testing technique used to compare the means of three or more groups to determine whether there are statistically significant differences among them.
- The test evaluates whether at least one group mean differs significantly from the others, indicating a meaningful relationship between the feature and the target variable.
- ANOVA is particularly useful when analysing numerical features across multiple categorical groups, making it an effective tool for feature selection in machine learning workflows.

2) Chi-Square-Test:(Best Approach)

- The Chi-Square (χ^2) Test is a statistical hypothesis test used to determine whether there is a significant association between two categorical variables, by comparing observed frequencies with expected frequencies.
- It evaluates whether two categorical variables are independent or related.
- The Chi-Square test is well suited for categorical features and provides a significance score (p-value) for each feature, making it useful for feature selection in model development.
- After performing the Chi-Square test, a p-value is obtained for each feature, which represents the probability that the observed relationship occurred by random chance.
- A significance level (α) of 0.05 is commonly used:
- If p-value < 0.05, the feature is statistically significant and retained for model training.
- If p-value ≥ 0.05, the feature is considered statistically insignificant and may be removed from the model.

- Based on this criterion, only categorical features with a significant relationship to the target variable were selected for inclusion in the final model.

3.4. Correlation:

- Correlation Hypothesis Testing is a statistical technique used to determine whether there is a significant linear relationship between two continuous numerical variables.
- The most commonly used measure is the Pearson correlation coefficient (r), which quantifies both the strength and direction of the linear relationship between two variables.
- The value of r ranges from −1 to +1:

  +1 indicates a perfect positive linear correlation

  −1 indicates a perfect negative linear correlation

  0 indicates no linear correlation

- To statistically validate the observed correlation, a p-value is computed alongside the correlation coefficient.
- The p-value represents the probability that the observed correlation occurred by random chance.
- Using a significance level of α = 0.05:

- If p-value < 0.05, the correlation is considered statistically significant.
- If p-value ≥ 0.05, the correlation is considered not statistically significant.

$$r = \frac{\sum(x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum(x_i - \bar{x})^2 \sum(y_i - \bar{y})^2}}$$

- x_i, y_i→ Individual observations
- $\bar{x}$, $\bar{y}$→ Mean of variables $X$and Y
- Numerator → Covariance between $X$and Y
- Denominator → Product of their standard deviations

In feature selection, correlation analysis helps identify strongly related numerical features and detect redundant or highly correlated variables, which may be removed to reduce multicollinearity.

4. Merging:

- As part of the preprocessing workflow, the dataset was initially split into numerical and categorical feature subsets to evaluate their individual relevance and statistical strength for model building.
- After completing feature evaluation and selection for both data types, it is necessary to recombine numerical and encoded categorical features to form a unified feature matrix for model training.
- Both numerical and categorical features were therefore merged for the training and testing datasets to ensure consistency across the machine learning pipeline.
- This merging was performed using the pandas.concat() function, allowing the features to be aligned correctly along the column axis for downstream model training and evaluation.

5. Data Balancing:

- In real-world datasets, the target variable often exhibits class imbalance, where one class significantly outnumbers the other(s).
- Such imbalance can cause machine learning models to become biased toward the majority class, resulting in poor predictive performance for the minority class, which is often the most critical in churn prediction.
- To address this issue, data balancing techniques such as undersampling, oversampling, and SMOTE (Synthetic Minority Over-sampling Technique) can be applied.
- In this project, SMOTE was selected as it is an advanced oversampling technique that generates synthetic samples rather than simply duplicating existing minority class instances.
- The SMOTE algorithm operates as follows:
- For each minority class observation, identify its k nearest neighbors (commonly $k = 5$).

- Randomly select one or more of these neighboring samples.
- Generate new synthetic data points along the line segment connecting the original sample and the selected neighbor.
- By creating realistic synthetic samples, SMOTE helps balance class distribution and improves model learning and generalization for churn prediction.
- The synthetic sample is generated using the formula:

$$x_{\text{new}} = x_i + (x_{zi} - x_i) \times \delta$$

Where:

- $x_i \rightarrow$ an original minority class instance
- $x_{zi} \rightarrow$ one of the k nearest neighbors of x_i(also from the minority class)
- $\delta \rightarrow$ a random number between 0 and 1

6. Feature Scaling:

- Feature Scaling is a preprocessing technique used to normalize or standardize numerical features so that they fall within a comparable range before model training.
- Many machine learning algorithms—especially distance-based and gradient-based models - perform better and converge faster when features are on a similar scale.
- In this project, Standard Scaling (Z-score normalization) was applied to numerical features during model development.

  Each feature value $x$is transformed as:

$$z = \frac{x - \mu}{\sigma}$$

Where:

- μ= mean of the feature
- σ= standard deviation of the feature
- z= standardized value

After transformation:

- The feature has mean = 0
- The feature has standard deviation = 1

7. Model Training:

- After completing all preprocessing steps, including feature engineering, feature selection, class balancing, and feature scaling, the dataset was prepared for model training.
- Since the target (dependent) variable is binary, the problem was framed as a binary classification task.
- Multiple classification algorithms were trained and evaluated during model development. Each model was trained on the same training data to ensure a fair comparison.
- The final model was selected based on performance evaluation metrics, with particular emphasis on ROC and AUC scores.

ROC Curve (Receiver Operating Characteristic):

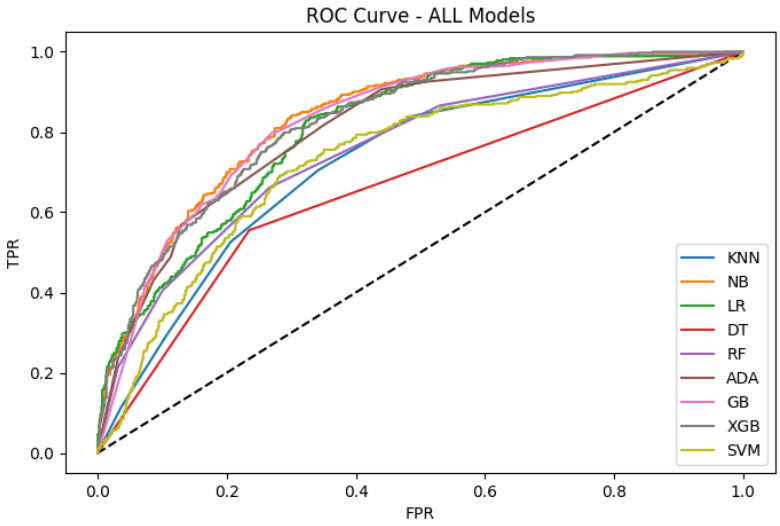
- The ROC curve is a graphical representation that illustrates a classifier's performance across different decision thresholds.
- True Positive Rate (TPR), also known as Recall or Sensitivity

$$TPR = \frac{TP}{TP + FN}$$

- False Positive Rate (FPR), also known as 1 − Specificity

$$FPR = \frac{FP}{FP + TN}$$

- The ROC curve shows the trade-off between sensitivity and specificity, helping to evaluate how well a model separates the two classes as the threshold changes.



AUC (Area Under the ROC Curve):

- AUC provides a single scalar value that summarizes the overall performance of the classifier.
- It represents the degree of separability between the positive and negative classes.
- A higher AUC value indicates a better ability of the model to distinguish between churned and non-churned customers:

  AUC = 1 → Perfect classifier

  AUC = 0.5 → No discriminative power (random guessing)

Model Selection Criterion:

- Among all trained models, the algorithm with the highest ROC–AUC score and stable performance across thresholds was selected as the final model.
- ROC–AUC was preferred as the primary evaluation metric because it is robust to class imbalance and threshold-independent.

```
2026-02-04 16:23:56,368 - INFO - KNN AUC  : 0.7248374858964671
2026-02-04 16:23:56,375 - INFO - NB AUC   : 0.8366241576697342
2026-02-04 16:23:56,381 - INFO - LR AUC   : 0.8102699597337667
2026-02-04 16:23:56,384 - INFO - DT AUC   : 0.6602019004834019
2026-02-04 16:23:56,386 - INFO - RF AUC   : 0.7532399308538719
2026-02-04 16:23:56,388 - INFO - ADA AUC  : 0.8094108087405675
2026-02-04 16:23:56,392 - INFO - GB AUC   : 0.8302309874025693
2026-02-04 16:23:56,400 - INFO - XGB AUC  : 0.8276289502831057
2026-02-04 16:23:56,404 - INFO - SVM AUC  : 0.7319397145134411
```

- Based on the evaluation results, the Naive Bayes (NB) classifier achieved the highest AUC score (≈ 0.8366) among all tested models. The ROC curve for Naive Bayes consistently remained above those of the other algorithms, indicating superior class separability and more reliable discrimination capability.

8. Hyperparameter Tuning:

- Hyperparameter tuning is the process of identifying the optimal combination of hyperparameters that maximizes a machine learning model's performance. Once the algorithm for model development is finalized, tuning is performed to enhance model accuracy and generalization ability.
- The selection of optimal hyperparameters can be achieved using techniques such as GridSearchCV and RandomizedSearchCV. GridSearchCV exhaustively evaluates all possible combinations of the specified hyperparameter values using cross-validation, ensuring a thorough search of the parameter space. In contrast, RandomizedSearchCV samples a fixed number of random parameter combinations, making it computationally more efficient for large search spaces.
- In this project, GridSearchCV was employed to determine the best hyperparameter configuration. GridSearchCV (Grid Search with Cross-Validation) is an automated approach that systematically explores a predefined grid of hyperparameters, trains models for each combination, evaluates their performance using cross-validation, and finally returns the optimal set of hyperparameters along with the best-performing model.

9. BEST MODEL:

- Naive Bayes predicts the probability that an observation belongs to a particular class based on Bayes' Theorem and feature independence assumptions.
- The output is a class label such as Yes or No, where Yes refers to 1 and No refers to 0, along with the corresponding probability estimates.
- After training the model using the Naive Bayes algorithm, its performance is evaluated on the test dataset using a confusion matrix, classification report (precision, recall, F1-score), and accuracy score.
- Once the evaluation is completed and the model performance is satisfactory, the best-trained Naive Bayes model is saved using pickle for future use and deployment.
- The sigmoid function converts any real number into a value between 0 and 1.

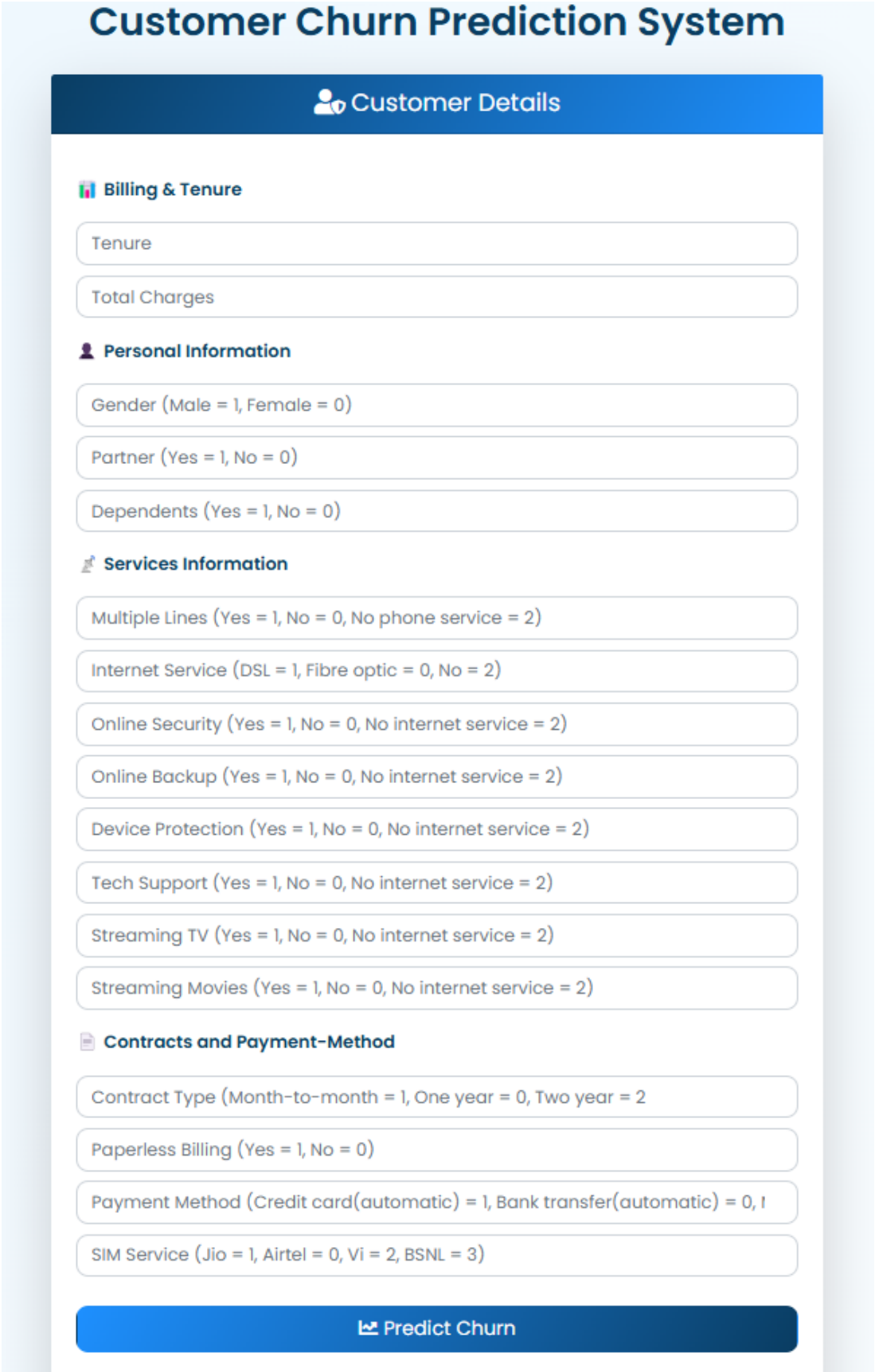
$$\sigma(z) = \frac{1}{1 + e^{-z}}$$

Frontend & Backend:

The system uses a Flask-based web application with an HTML frontend that allows users to input customer details through an interactive form. The form collects numerical features such as tenure and charges, along with categorical attributes like contract type, payment method, and service options using dropdowns to ensure valid inputs. On submission, the data is sent to the backend via an HTTP POST request, and the prediction result with probability is displayed on the same page in real time.

The backend, developed using Flask, loads the trained Naive Bayes model, scaler, and encoding mappings at startup. It preprocesses incoming data using the same transformations applied during training, including scaling and categorical encoding, while maintaining feature alignment. The processed data is passed to the model to generate churn prediction and probability, which are then returned to the frontend, enabling seamless and efficient real-time predictions.

Result:


Customer Churn Prediction System
Customer Details
Billing & Tenure
Tenure
Total Charges
Personal Information
Gender (Male = 1, Female = 0)
Partner (Yes = 1, No = 0)
Dependents (Yes = 1, No = 0)
Services Information
Multiple Lines (Yes = 1, No = 0, No phone service = 2)
Internet Service (DSL = 1, Fibre optic = 0, No = 2)
Online Security (Yes = 1, No = 0, No internet service = 2)
Online Backup (Yes = 1, No = 0, No internet service = 2)
Device Protection (Yes = 1, No = 0, No internet service = 2)
Tech Support (Yes = 1, No = 0, No internet service = 2)
Streaming TV (Yes = 1, No = 0, No internet service = 2)
Streaming Movies (Yes = 1, No = 0, No internet service = 2)
Contracts and Payment-Method
Contract Type (Month-to-month = 1, One year = 0, Two year = 2
Paperless Billing (Yes = 1, No = 0)
Payment Method (Credit card(automatic) = 1, Bank transfer(automatic) = 0, I
SIM Service (Jio = 1, Airtel = 0, Vi = 2, BSNL = 3)
Predict Churn

Conclusion:

The Customer Churn Prediction project demonstrates the effective application of machine learning techniques to help telecom companies proactively identify customers who are at risk of discontinuing their services. By analysing key customer attributes such as demographics, service usage patterns, contract type, and billing information, the model uncovers meaningful insights into customer behaviour and churn drivers. The dataset was thoroughly pre-processed using industry-standard techniques, including data cleaning, feature engineering, feature selection, class balancing, and feature scaling, ensuring high-quality inputs for model training. Multiple machine learning algorithms were trained and evaluated to identify the most suitable model for churn prediction. Model performance was assessed using ROC and AUC metrics, which are robust indicators for binary classification problems. Based on this evaluation, Logistic Regression was selected as the final model, achieving an overall prediction accuracy of 75% while maintaining reliable class separability between churned and non-churned customers. To enhance real-world usability, the project also includes a Flask-based web application, allowing users to input customer details and instantly receive churn predictions. This end-to-end deployment bridges the gap between data science and business operations, enabling data-driven decision-making and supporting proactive customer retention strategies. Overall, this project highlights how machine learning, when combined with thoughtful preprocessing and deployment, can serve as a powerful tool for improving customer retention and business performance.

Future Enhancement:

- **Integration of Deep Learning Models:**
  Incorporate advanced deep learning architectures such as Artificial Neural Networks (ANN), Convolutional Neural Networks (CNN), and Long Short-Term Memory (LSTM) networks to capture complex nonlinear and sequential patterns in customer behaviour, potentially improving prediction accuracy.
- **Real-Time Churn Monitoring:**
  Implement real-time data pipelines using streaming technologies to enable continuous churn prediction and timely identification of at-risk customers.
- **Advanced Visual Analytics Dashboard:**
  Develop an interactive visual analytics dashboard to monitor churn trends, customer segments, and risk scores over time, supporting strategic business decision-making.
- **Scalable Deployment Architecture:**
  Enhance the system with scalable deployment solutions to support large-scale telecom datasets and high-frequency prediction requests.

These enhancements would further strengthen the system's predictive power, scalability, and business impact.

References:

- https://www.kaggle.com/datasets/blastchar/telco-customer-churn
- https://feature-engine.trainindata.com/en/latest/api_doc/index.html
- https://scikit-learn.org/stable/index.html
- https://scikit-learn.org/stable/modules/model_evaluation.html#roc-metrics
- https://pandas.pydata.org/docs/
- https://numpy.org/doc/
- https://flask.palletsprojects.com/en/latest/