A MAJOR PROJECT REPORT ON

AI-Powered Customer Retention Prediction System

Submitted for Award of Internship Certified for

THE SKILL UNION

Under

VIHARA TECH

Submitted by

B. L. Archana

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 Customer Retention Prediction System**" is a Bonafide record of the Industry Oriented Major Project Work submitted by

B. L. Archana

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	SIGNATURE OF VIHARA TECH MENTOR
Mr. K Sai Kamal	**Mr. Pranith**

SIGNATURE OF THE SKILL UNION
CEO

Mr. K Sai Krishna

DECLARATION

I, B.L.Archana, hereby declare that this project report, titled **" AI-Powered 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
B.L.Archana	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.

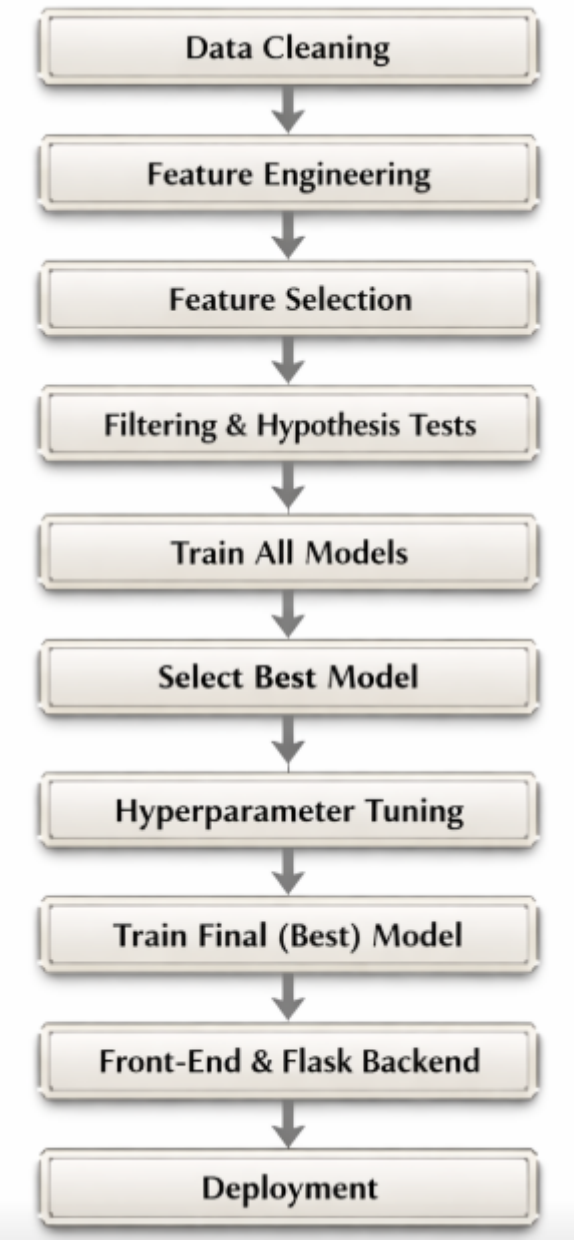
I am profoundly thankful to my dedicated Institute **Mentor, Mr. Pranith**. Their steadfast guidance during both the project development and the subsequent two-month internship was instrumental in bringing this report to fruition. Their insightful suggestions, technical foresight, and willingness to 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 retention is a crucial aspect of business growth and profitability, especially in highly competitive markets such as telecommunications, banking, and e-commerce. Customer churn occurs when a customer stops using a company's services, directly impacting revenue and long-term sustainability. This project aims to develop a Customer Churn Prediction Model using Machine Learning techniques to identify customers who are likely to leave the service in the future.

The proposed model follows a structured pipeline comprising data cleaning, feature engineering, feature selection, statistical hypothesis testing, and model training. The dataset is preprocessed to handle missing values, outliers, categorical encoding, and feature scaling. SMOTE is applied to balance the data distribution. Multiple machine learning algorithms — including Logistic Regression, Decision Tree, Random Forest, KNN, Naive Bayes — are trained and compared based on performance metrics such as accuracy, precision, recall, F1-score, and ROC-AUC.

After evaluation, the best-performing model is fine-tuned through hyperparameter optimization to enhance predictive accuracy. The final model is integrated into a Flask-based web application, allowing users to input customer details and instantly obtain churn predictions. The system provides actionable insights for businesses to implement targeted retention strategies, reduce customer attrition, and improve decision-making.

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

In today's competitive business environment, retaining existing customers is just as important as acquiring new ones. Organizations across various industries — such as telecommunications, banking, insurance, and retail — face a common challenge known as customer churn, which refers to the phenomenon where customers discontinue using a company's products or services. High churn rates can significantly impact revenue, increase acquisition costs, and reduce long-term profitability. Therefore, identifying potential churners before they leave is critical for developing effective retention strategies.

With the rapid growth of data availability and computational power, Machine Learning (ML) has emerged as a powerful tool for analyzing customer behavior and predicting churn. By learning from historical customer data, ML models can detect complex patterns and relationships between customer attributes, service usage, and churn behavior. These predictive insights enable businesses to proactively engage at-risk customers through personalized offers, improved services, or loyalty programs.

This project demonstrates how machine learning techniques can be effectively utilized to enhance customer retention strategies, reduce churn rates, and improve overall business performance. By integrating predictive analytics into decision-making, organizations can move from reactive to proactive customer management, fostering long-term loyalty and sustainable growth.

Requirements of the model development

- **blinker** – Enables signal-based communication between different parts of a Python application.
- **click** – Used to create clean and user-friendly command-line interfaces.
- **colorama** – Provides colored text output in the terminal across platforms.
- **contourpy** – Generates contour plots used in data visualizations.
- **cycler** – Manages repeating style patterns like colors in plots.
- **feature_engine** – Provides feature engineering and preprocessing tools for machine learning.
- **Flask** – A lightweight framework for building web applications and APIs.
- **fonttools** – Used for handling and manipulating font files.
- **gunicorn** – A production-ready server for running Flask applications.
- **imbalanced-learn** – Helps handle imbalanced datasets using resampling techniques.
- **imblearn** – Acts as an alias for the imbalanced-learn package.
- **itsdangerous** – Secures data such as session cookies using cryptographic signing.
- **Jinja2** – Used to create dynamic HTML pages in Flask applications.
- **joblib** – Saves trained models and supports parallel processing.
- **kiwisolver** – Solves layout constraints in visualization libraries.
- **MarkupSafe** – Prevents HTML injection by escaping unsafe characters.
- **matplotlib** – Creates static and interactive data visualizations.
- **numpy** – Performs numerical computations on multi-dimensional arrays.
- **packaging** – Handles package versioning and dependency management.
- **pandas** – Used for data manipulation and analysis using DataFrames.
- **patsy** – Builds statistical model formulas for regression analysis.
- **pillow** – Used for image processing and manipulation.
- **pyparsing** – Supports parsing and interpreting structured text.
- **python-dateutil** – Extends datetime functionality for flexible date handling.
- **pytz** – Provides accurate time zone calculations.
- **scikit-learn** – Offers machine learning algorithms and pre-processing tools.
- **scipy** – Provides scientific computing tools like statistics and optimization.
- **seaborn** – Creates advanced statistical visualizations.
- **six** – Ensures compatibility between Python 2 and Python 3.
- **statsmodels** – Performs statistical modeling and hypothesis testing.
- **threadpoolctl** – Controls CPU thread usage for performance optimization.
- **tzdata** – Supplies time zone database information.
- **Werkzeug** – Handles HTTP requests and responses in Flask applications.

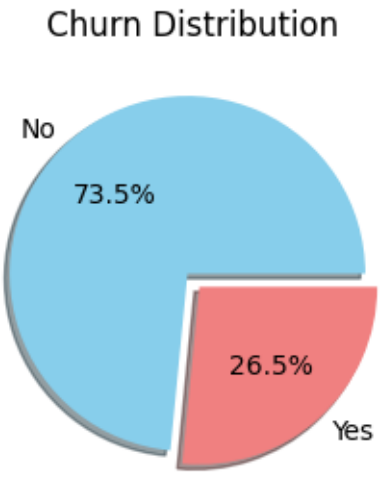
1. Data Visualization

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
- Tools used in the matplot library are:
  =>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
- Tools used in seaborn library for data analysis are:
  =>Hist plot
  =>Bar plot
- I also used pandas for Data manipulation and preprocessing and Numpy libraries for numerical Operations and to perform scientific calculations.

1.2. Churn comparision

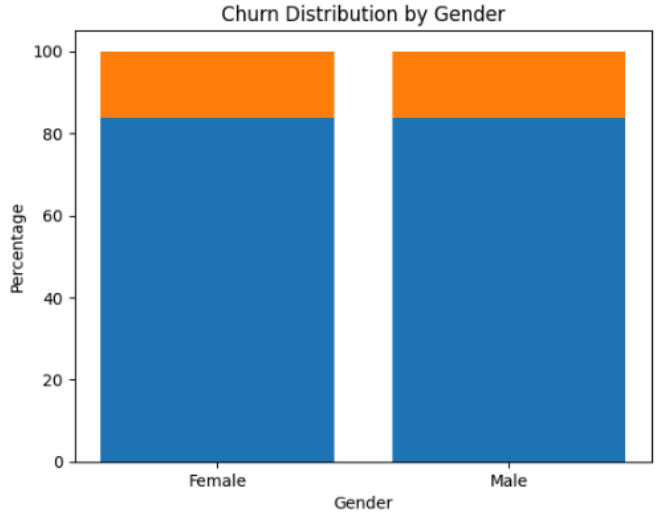
- From the data churn percentage is
  - Yes =>26.5
  - No =>73.5
- Even though the churn percentage (Yes) is smaller, it represents a critical business problem, because losing even 20–25% of customers can significantly impact revenue.
- Therefore, Churn = Yes/No serves as the foundation of the entire churn prediction process.



- Most customers (73.5%) did not churn, while 26.5% of customers churned, indicating class imbalance in the dataset.

No (Not Churned)	-	73.5%
Yes (Churned)	-	26.5%

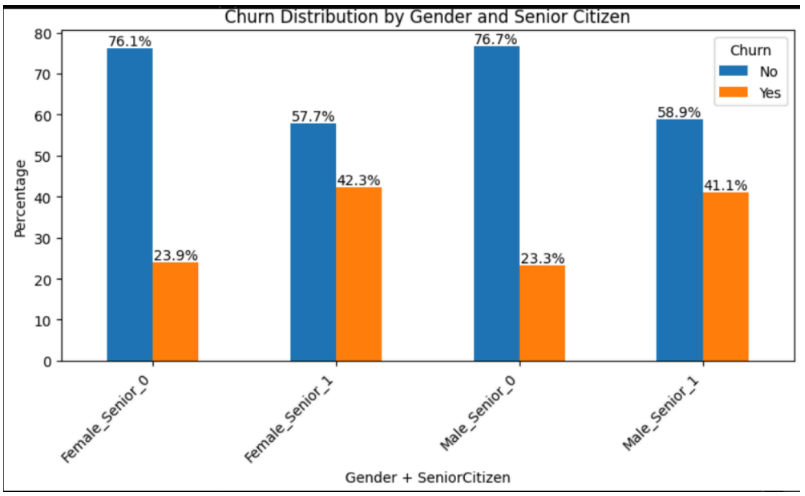
1.3 Churn Distribution by Gender:



- when we compare gender with senior citizens female senior citizens are more likely to leave.

Female	0	83.715596
	1	16.284404
Male	0	83.853727
	1	16.146273

1.4 Churn distribution by Gender and Senior Citizen:



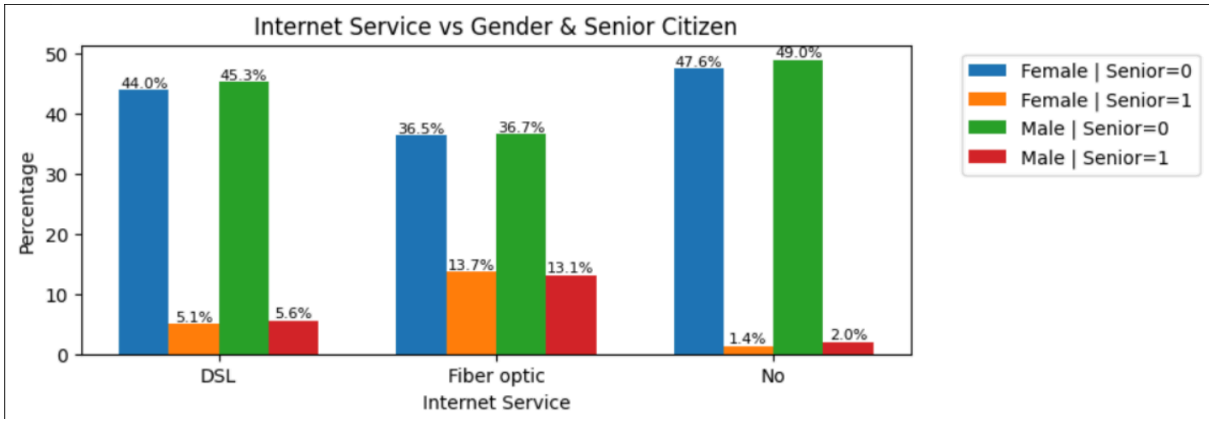
- The dependents who are likely to leave are senior citizens of the Female gender.

	No	Yes
Female, Senior = 0	76.1	23.9
Female, Senior = 1	57.7	42.3
Male, Senior = 0	76.7	23.3
Male, Senior = 1	58.9	41.1

1.5. Internet Service Usage by Gender and Senior Citizen:

	DSL	Fiber Optics	No Internet
Female (Senior = 0)	44.0%	36.5%	47.6%
Female (Senior = 1)	5.1%	13.7%	1.4%
Male (Senior = 0)	45.3%	36.7%	49.0%
Male (Senior = 1)	5.6%	13.1%	2.0%

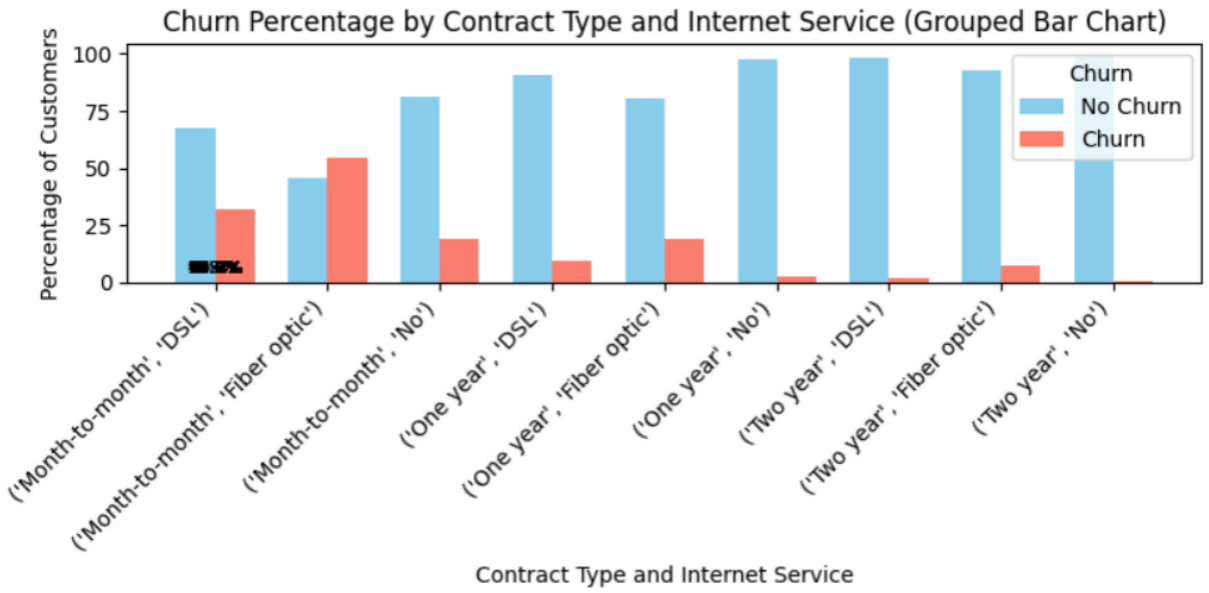
- Senior citizens using fiber optic internet show higher churn, while customers with no internet service show the lowest churn across all groups.



1.6. Churn Percentage by Contract Type and Internet Service:

Contract Type	Internet Service	No Churn	Churn
Month-to-Month	DSL	68.0	32.0
Month-to-Month	Fiber Optic	45.0	55.0
Month-to-Month	No Internet	82.0	18.0
One Year	DSL	91.0	9.0
One Year	Fiber Optic	81.0	19.0
One Year	No Internet	98.0	2.0
Two Year	DSL	99.0	1.0
Two Year	Fiber Optic	92.0	8.0
Two Year	No Internet	99.5	0.5

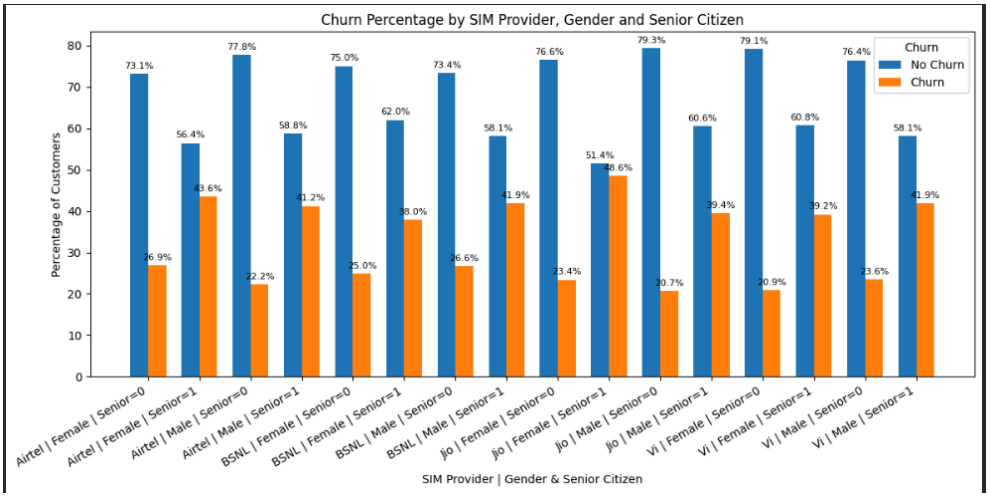
- Customers with **month-to-month contracts**, especially those using **fiber optic internet**, show the **highest churn rates**.



1.7. Churn Percentage by SIM Provider, Gender, and Senior Citizen:

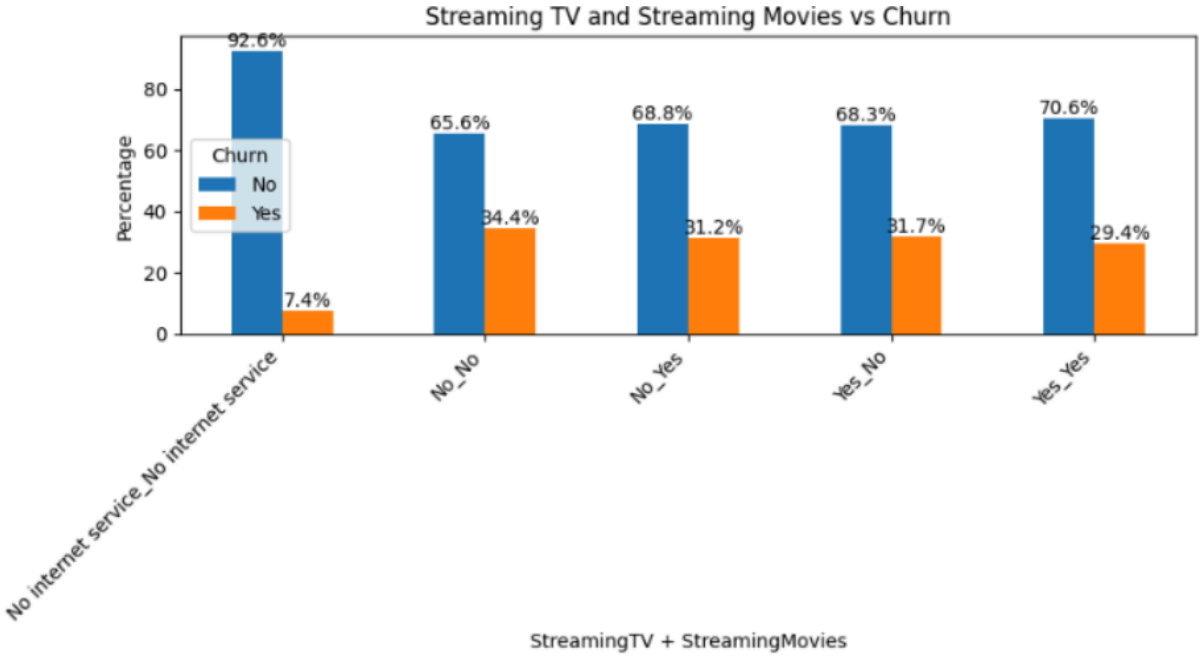
SIM Provider	Gender	Senior Citizen	No Churn (%)	Churn (%)
Airtel	Female	0	73.1	26.9
Airtel	Female	1	56.4	43.6
Airtel	Male	0	77.8	22.2
Airtel	Male	1	58.8	41.2
BSNL	Female	0	75.0	25.0
BSNL	Female	1	62.0	38.0
BSNL	Male	0	73.4	26.6
BSNL	Male	1	58.1	41.9
Jio	Female	0	76.6	23.4
Jio	Female	1	51.4	48.6
Jio	Male	0	79.3	20.7
Jio	Male	1	60.6	39.4
Vi	Female	0	79.1	20.9
Vi	Female	1	60.8	39.2
Vi	Male	0	76.4	23.6

Vi	Male	1	58.1	41.9



- Senior citizens show higher churn across all SIM providers, irrespective of gender.

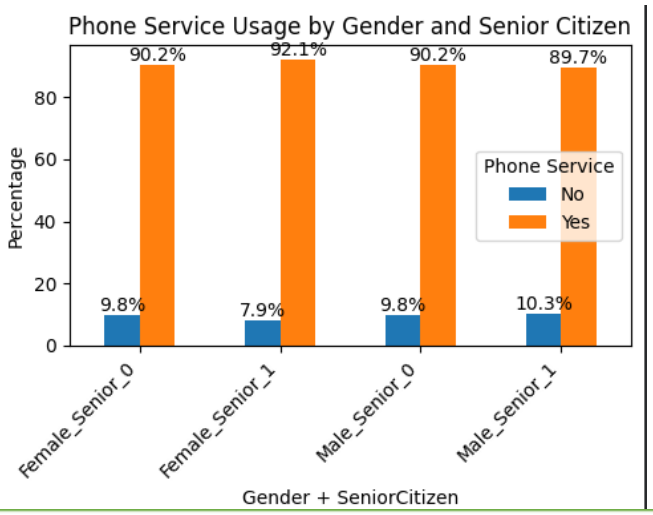
1.8. Streaming TV and Streaming Movies vs Churn:



Streaming TV	Streaming Movies	No Churn (%)	Churn (%)
No Internet	No Internet	92.6	7.4
No	No	65.6	34.4
No	Yes	68.8	31.2
Yes	No	68.3	31.7
Yes	Yes	70.6	29.4

- Customers with **no internet service have the lowest churn**, while customers using streaming services show higher churn.

1.9. Phone Service Usage by Gender and Senior Citizen:

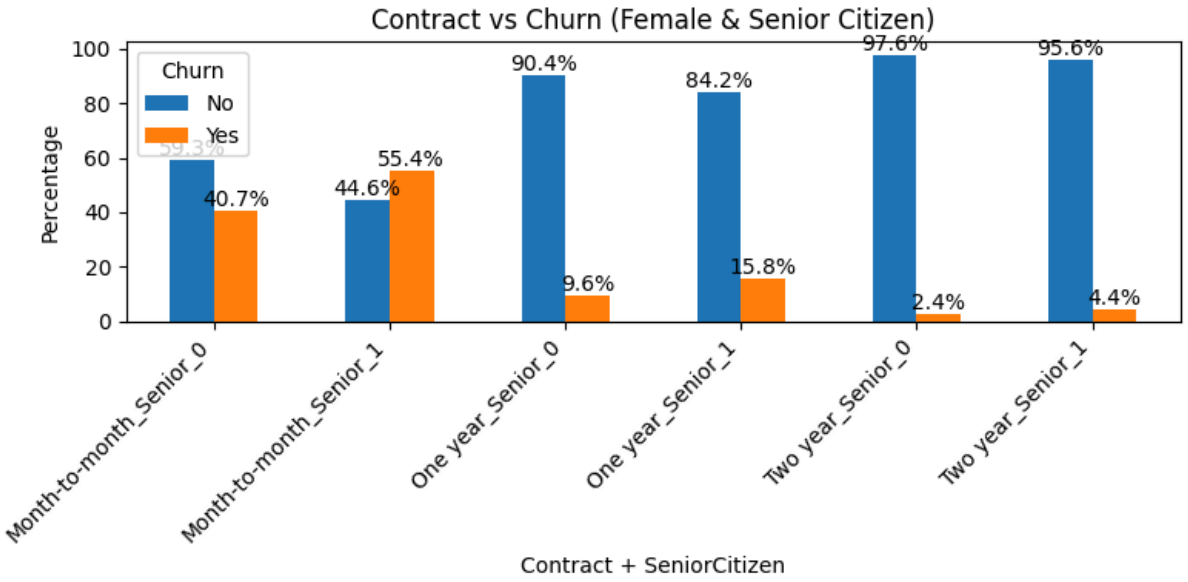


Gender	Senior Citizen	Phone Service = No (%)	Phone Service = Yes (%)
Female	0	9.8	90.2
Female	1	7.9	92.1
Male	0	9.8	90.2
Male	1	10.3	89.7

- Most customers, irrespective of gender and senior status, actively use phone services.

1.10. Contract vs Churn (Female & Senior Citizen):

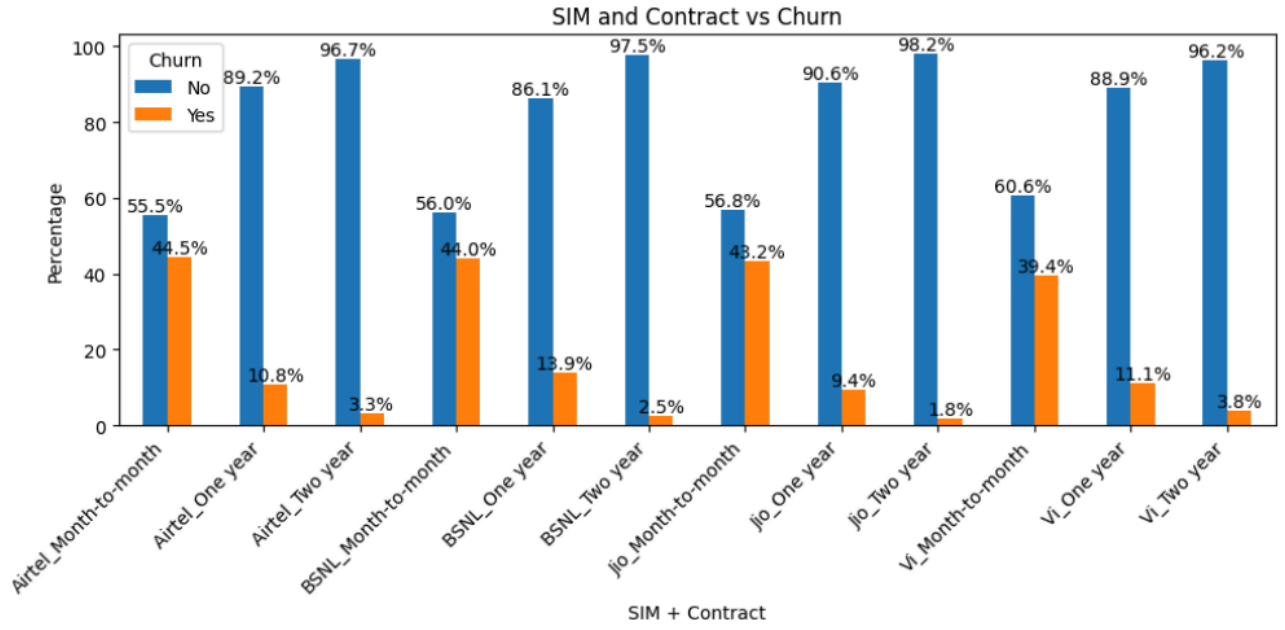
Contract Type	Senior Citizen	No Churn (%)	Churn (%)
Month-to-Month	0	59.3	40.7
Month-to-Month	1	44.6	55.4
One Year	0	90.4	9.6
One Year	1	84.2	15.8
Two Year	0	97.6	2.4
Two Year	1	95.6	4.4



- Female customers with **month-to-month contracts**, especially senior citizens, show the **highest churn**.
-

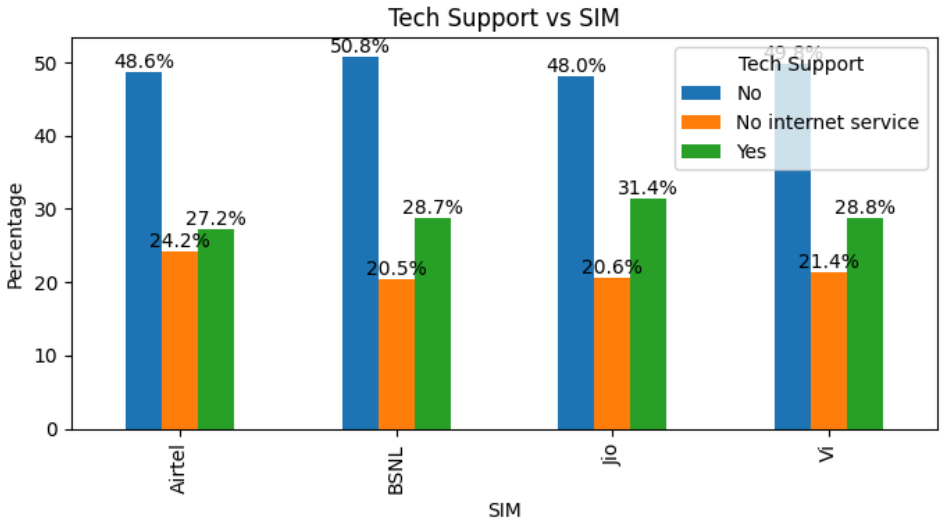
1.11. SIM and Contract vs Churn:

SIM Provider	Contract Type	No Churn (%)	Churn (%)
Airtel	Month-to-Month	55.5	44.5
Airtel	One Year	89.2	10.8
Airtel	Two Year	96.7	3.3
BSNL	Month-to-Month	56.0	44.0
BSNL	One Year	86.1	13.9
BSNL	Two Year	97.5	2.5
Jio	Month-to-Month	56.8	43.2
Jio	One Year	90.6	9.4
Jio	Two Year	98.2	1.8
Vi	Month-to-Month	60.6	39.4
Vi	One Year	88.9	11.1
Vi	Two Year	96.2	3.8



- Across all SIM providers, month-to-month contracts show the highest churn, while long-term contracts show strong retention.

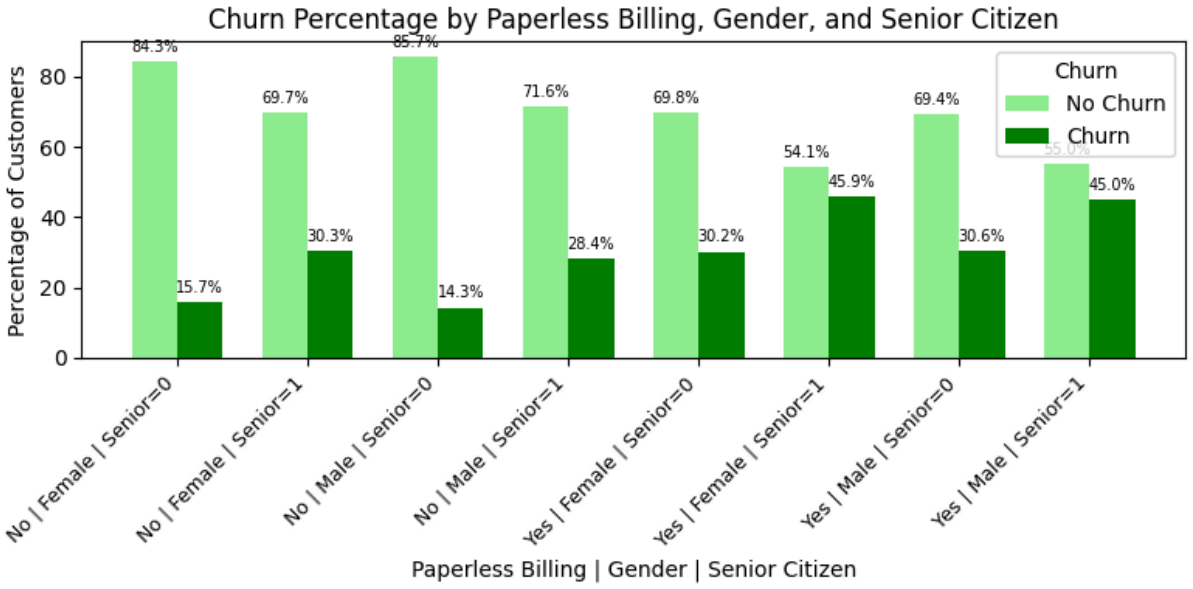
1.12. Tech Support vs SIM:



SIM Provider	Tech Support = No (%)	No Internet Service (%)	Tech Support = Yes (%)
Airtel	48.6	24.2	27.2
BSNL	50.8	20.5	28.7
Jio	48.0	20.6	31.4
Vi	49.3	21.4	28.8

- A large portion of customers across all SIM providers do not use tech support, while usage is slightly higher for Jio users.

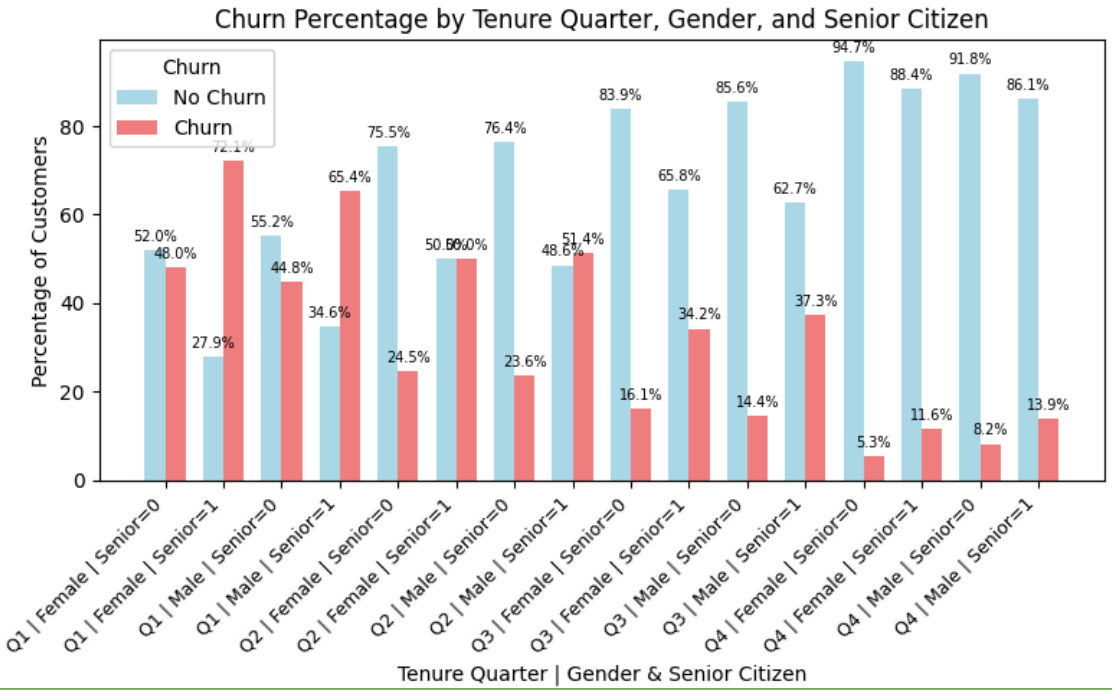
1.13. Paperless Billing vs Churn (Gender & Senior Citizen):



Paperless Billing	Gender	Senior Citizen	No Churn (%)	Churn (%)
No	Female	0	84.3	15.7
No	Female	1	69.7	30.3
No	Male	0	85.7	14.3
No	Male	1	71.6	28.4
Yes	Female	0	69.8	30.2
Yes	Female	1	54.1	45.9
Yes	Male	0	69.4	30.6
Yes	Male	1	55.0	45.0

- Customers using **paperless billing**, especially **senior citizens**, show significantly higher churn compared to non-paperless users.

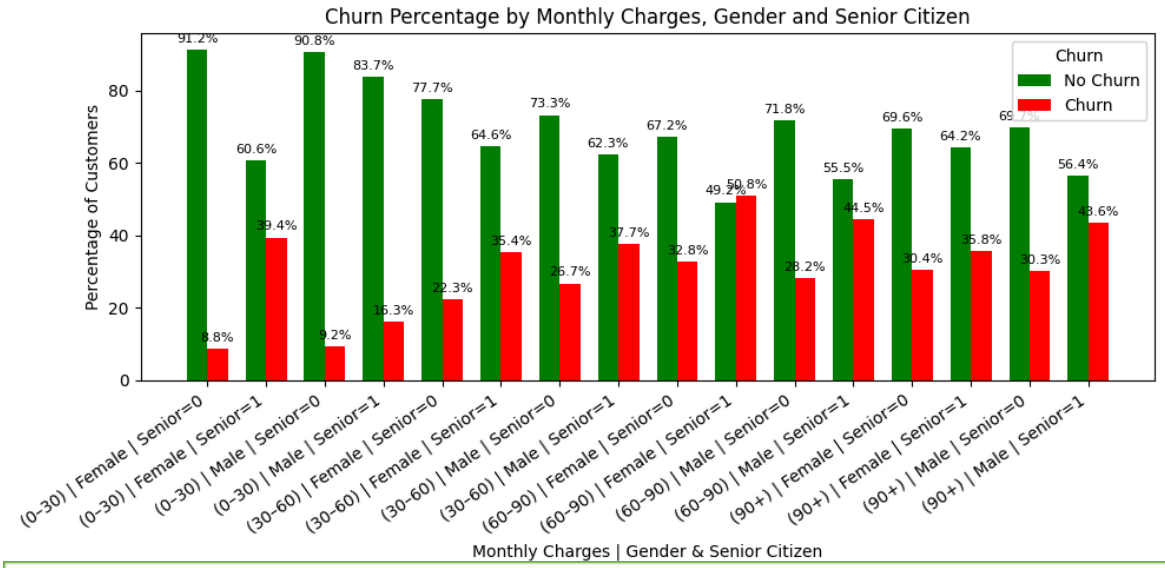
1.14. Churn by Tenure Quarter (Gender & Senior Citizen):



Tenure Quarter	Gender	Senior Citizen	No Churn (%)	Churn (%)
Q1	Female	0	52.0	48.0
Q1	Female	1	27.9	72.1
Q1	Male	0	55.2	44.8
Q1	Male	1	34.6	65.4
Q2	Female	0	75.5	24.5
Q2	Female	1	50.0	50.0
Q2	Male	0	76.4	23.6
Q2	Male	1	48.6	51.4
Q3	Female	0	83.9	16.1
Q3	Female	1	65.8	34.2
Q3	Male	0	85.6	14.4
Q3	Male	1	62.7	37.3
Q4	Female	0	94.7	5.3
Q4	Female	1	88.4	11.6
Q4	Male	0	91.8	8.2
Q4	Male	1	86.1	13.9

- Churn is **highest in the early tenure (Q1)**, especially for **senior citizens**, and consistently decreases as tenure increases.

1.15. Churn by Monthly Charges (Gender & Senior Citizen):



Monthly Charges	Gender	Senior Citizen	No Churn (%)	Churn (%)
0–30	Female	0	91.2	8.8
0–30	Female	1	60.6	39.4
0–30	Male	0	90.8	9.2
0–30	Male	1	83.7	16.3
30–60	Female	0	77.7	22.3
30–60	Female	1	64.6	35.4
30–60	Male	0	73.3	26.7
30–60	Male	1	62.3	37.7
60–90	Female	0	67.2	32.8
60–90	Female	1	49.2	50.8
60–90	Male	0	71.8	28.2
60–90	Male	1	55.5	44.5
90+	Female	0	69.6	30.4
90+	Female	1	64.2	35.8
90+	Male	0	69.7	30.3
90+	Male	1	56.4	43.6

- Churn increases as **monthly charges rise**, with **senior citizens** showing consistently higher churn across all charge ranges.

1.16. Final Observations from the Visualizations:

- Most customers do not churn, but a noticeable portion still leaves.
- Senior citizens churn much more than non-senior customers in almost every case.
- Gender has very little impact on churn compared to other factors.
- Month-to-month contracts show the highest churn.
- One-year and two-year contracts greatly reduce churn.
- New customers (low tenure) churn more than long-term customers.
- High monthly charges lead to higher churn, especially for seniors.
- Customers using fiber optic internet and streaming services churn more.
- Customers with no internet service churn the least.
- Electronic check and mailed check payments have high churn.
- Automatic payment methods have low churn.
- Phone service usage is very high and does not strongly affect churn.
- Tech support usage is moderate and slightly higher for internet users.
- Across all SIM providers, patterns are similar; SIM provider alone is not a strong churn factor.

2. Feature Engineering

- Feature Engineering is the process of transforming raw data into meaningful features that better represent the underlying patterns in the dataset, thereby improving model performance.
- After the clear visualization on the data I have moved to the feature engineering phase where at first I have seen whether the missing values present in the data or not. After performing operations on the data I have found no null values in the data.
- Then I separated the data to categorical and numerical data. After the separation I have found the missing values in the ‘Total Charges’ column due to object data type in the Total Charges column as the values in the column are numbers.

- So for handling these missing values there are several methods. After checking some of the best methods I have finalized the best method in handling those missing values.

2.1. Handling Missing Values

I. Random Sample Imputation:

- 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.

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).
- 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. Constant Imputation Technique:

- **Constant Imputation** is a technique used to handle missing values by replacing them with a fixed constant value such as `0`, `-1`.

- This approach is simple to implement and allows the model to **explicitly identify missing values** as a separate category, which can be useful for categorical features.
- However, constant imputation can **distort the original data distribution** and introduce bias, as the constant value does not reflect the natural behavior of the feature; therefore, this method was **not selected** for handling missing values in the dataset.

IV. Arbitrary Technique:

- Arbitrary Value Imputation is a technique used to handle missing values by replacing them with a predefined arbitrary value such as -999 or 999, which lies outside the normal range of the feature.
- This method helps the model clearly distinguish missing values from valid data points and is often used when missingness itself carries information.
- However, using extreme arbitrary values can significantly distort the feature distribution and may mislead the model; therefore, this technique was not selected for handling missing values in the dataset.

V. End of Distribution:

- End of Distribution Imputation is a technique used to handle missing values by replacing them with extreme values from the distribution, such as the minimum or maximum value.
- This method allows the model to recognize missing values as unusual observations while keeping them within the feature's numerical range.
- However, adding extreme values can skew the distribution and impact model learning**;** therefore, this technique was not selected for handling missing values in the dataset.

VI. Forward Fill Imputation :

- Forward Fill Imputation is a method where missing values are filled using the previous available value in the dataset.
- This approach is useful for time-series or sequential data where past values influence future observations.
- Since the dataset is not time-dependent, forward fill may introduce incorrect patterns; therefore, it was not selected.

VII. Backward Fill Imputation:

- Backward Fill Imputation replaces missing values with the next available value in the dataset.
- It works well in ordered or time-based datasets where future values are relevant.
- As the dataset lacks a natural order, backward fill can cause data leakage**,** so this method was not chosen.

VIII. Best Selected Method :

- In this project, the best missing value handling technique was selected automatically by comparing the standard deviation of different imputation methods.
- Median imputation was chosen because it preserved the original data distribution, was robust to outliers, and produced the minimum deviation from the original feature statistics.

2.2. Data Separation

- In any churn prediction dataset, the features (columns) are usually a mix of numerical and categorical variables.
- Before applying transformations such as scaling, encoding, or imputation, it's essential to separate them.
- Once the missing values are handled with the best techniques I have gone through the categorical and numerical data separation.
- Machine learning algorithms understand numbers, not categories or text.
- However, numerical and categorical data represent information very differently, so we handle them separately before training.
- So this had done simply by checking their data type and separated to categorical and numerical data using select_dtypes() pandas function.

2.3. Variable Transformation

- variable transformation (also called feature transformation) is a data preprocessing step used to modify the values of variables (features) to make them more suitable for statistical analysis or machine learning models.

- Many ML algorithms (like linear regression, logistic regression, etc.) assume data is normally distributed. Transformations make data more symmetric.
- Transformations reduce the effect of extreme values.
- It improves the predictive capability of churn models and leads to more reliable business insights**.**
- Let's see some of the techniques that I have worked on.

2.3.1. Log Transformation:

- Log Transformation is a data transformation technique used to reduce the effect of extreme values by applying a logarithmic function to numerical features.
- It helps in converting skewed distributions into more normal distributions, which improves the performance of many machine learning models.
- Log transformation stabilizes variance, reduces right skewness, and makes relationships between variables more linear.
- It is especially useful for features like charges, income, or tenure, where values grow exponentially.
- Log transformation should be applied only to positive values, as logarithms of zero or negative numbers are undefined.

2.3.2. Reciprocal Transformation:

- Reciprocal Transformation is a data transformation technique used to reduce the impact of large values by transforming a feature using the reciprocal function 1/x1/x1/x.
- It is effective in handling right-skewed distributions by compressing large values more aggressively than log transformation.
- This transformation helps in reducing the influence of extreme outliers and stabilizing variance.
- Reciprocal transformation is suitable only for non-zero and positive values, as division by zero is undefined.
- It is mainly applied when the feature has very large values and a strong skewness.

2.3.3. Square root Transformation:

- **Square Root Transformation** is a data transformation technique used to reduce skewness by applying the square root function to numerical features.
- It is effective for **moderately right-skewed data** and helps stabilize variance without being too aggressive.
- This transformation reduces the influence of **larger values** while keeping the overall data distribution interpretable.
- Square root transformation can be applied to **zero and positive values**, unlike log and reciprocal transformations.
- It is commonly used for features that represent **counts or charges**.

2.3.4. Exponential Transformation:

- Exponential Transformation is a data transformation technique where numerical features are transformed using the exponential function e^x .
- It is mainly used to increase differences between values, especially when the data is left-skewed or compressed.
- This transformation amplifies large values, making patterns more distinguishable for certain models.
- Exponential transformation can increase skewness and variance, so it must be applied carefully.
- It is less commonly used compared to log or square root transformations.

2.3.5. Box Cox Transformer:

- class feature_engine.transformation.BoxCoxTransformer (variables=None)
- The BoxCoxTransformer() applies the BoxCox transformation to numerical variables.
  The Box-Cox transformation is defined as:

- T(Y) = (Y exp(λ)−1) /λ if λ!=0
- log(Y) otherwise

- where Y is the response variable and λ is the transformation parameter. λ varies, typically from -5 to 5. In the transformation, all values of λ are considered and the optimal value for a given variable is selected.

2.3.6. Yeo Johnson Transformer:

- **Yeo–Johnson Transformation** is a power transformation technique used to make data more normally distributed.
- Unlike log and Box-Cox transformations, it can be applied to **both positive and negative values**, including zero.

- It reduces **skewness**, stabilizes **variance**, and improves linear relationships between features.
- The transformation automatically selects the **optimal lambda (λ)** value for each feature.
- It is especially useful when datasets contain **mixed-sign numerical values**.
- In this project, Yeo–Johnson transformation was applied to handle skewed numerical features and **improved model stability and performance**.

2.3.7. Best Selected method:

- After applying this method I got the bell curve which indicates the data is normally distributed and also probability graph has not given proper outcome. So I have choosen this method for transformation in the model development.

2.4 Handling Outliers

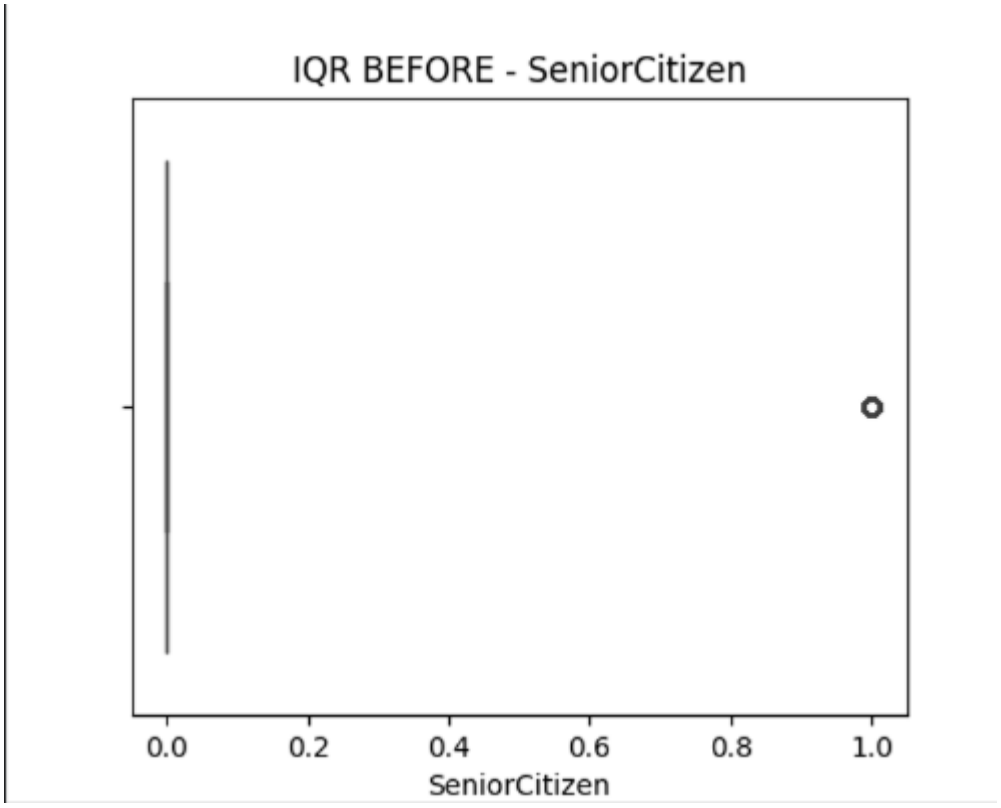
- Handling outliers is a crucial preprocessing step to ensure data consistency and model reliability.
- After the data is normally distributed we will check whether the outliers are present in the dataset. As we saw in the above box plot graphs there are outliers present in the dataset. Outliers are data points that differ significantly from most of the other observations in a dataset.
- Outliers can distort the data distribution and mislead machine learning models. So now we need to handle the outliers for better model training.
- Techniques like z-score, Trimming and IQR Capping ,Percentile Capping to effectively minimize the influence of extreme values in the dataset in which columns we find the outliers leading to more robust churn prediction results**.**

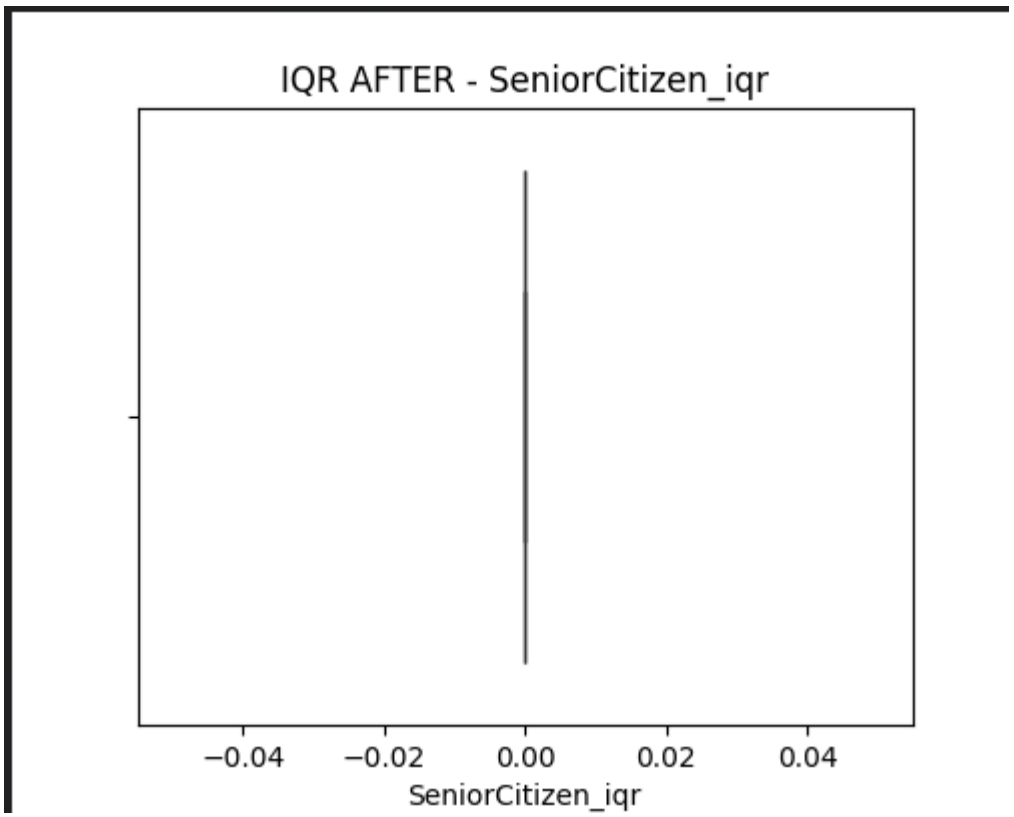
2.4.1. Z- Score Outlier Method:

- Removes rows where |Z|>threshold
- Trimming directly removes entire data points, which may include important or rare observations.
- As this technique is not so effective this technique is also not used in the model building.

2.4.2. IQR Capping :

- Outlier handling is an essential preprocessing step to ensure model stability and reliability. In this project**,** IQR Capping was used to treat extreme values present in numerical features.
- The Interquartile Range (IQR) method identifies outliers based on the spread of the middle 50% of the data. The first quartile (Q1) represents the 25th percentile, and the third quartile (Q3) represents the 75th percentile. The IQR is calculated as the difference between Q3 and Q1.
- Lower and upper limits are computed using the formula:
- Lower Bound = Q1 − 1.5 × IQR
- Upper Bound = Q3 + 1.5 × IQR
- Instead of removing data points outside these limits, IQR capping replaces extreme values with the corresponding lower or upper bound. This approach controls the influence of outliers while preserving all observations in the dataset.





2.4.3. Trimming :

- Trimming is an outlier handling technique in which extreme values are removed from the dataset instead of being modified or capped. This method focuses on eliminating observations that lie far outside the normal range of the data and may negatively affect model performance.
- In trimming, outliers are typically identified using statistical methods such as the Interquartile Range (IQR) or Z-score. Once the lower and upper thresholds are defined, all data points falling outside these limits are permanently deleted from the dataset.
- This technique helps improve data quality by removing noisy or abnormal observations that can distort model learning, especially for algorithms sensitive to extreme values.
- However, trimming leads to data loss, which can be a major drawback when the dataset size is limited or when outliers carry meaningful information. For this reason, trimming must be applied carefully and only when outliers are confirmed to be erroneous or irrelevant.
- In this project, trimming was analysed but not selected, as removing outliers could reduce valuable customer information. Instead, capping techniques were preferred to preserve dataset size while controlling the influence of extreme values.

2.4.4. Percentile Capping:

- Percentile Capping is an outlier handling technique in which extreme values are limited based on predefined percentile thresholds instead of removing them from the dataset. This method ensures that values beyond a certain range are capped to the nearest acceptable percentile.
- In this technique, lower and upper percentile limits are selected, commonly the 1st and 99th percentiles or 5th and 95th percentiles. Any data point below the lower percentile is replaced with the lower percentile value, and any data point above the upper percentile is replaced with the upper percentile value.
- Percentile capping is effective in controlling the influence of extreme outliers while preserving the overall dataset size. It is particularly useful when the data distribution is skewed and contains extreme values that can adversely affect model performance.
- In this project, percentile capping was considered as an alternative outlier handling method. However, it was not selected as the primary approach because IQR capping provided more robust boundaries for skewed features and better preserved the natural distribution of the data.

2.4.5. Best Selected Technique:

- IQR capping was selected because it is robust to skewed data, prevents information loss, and maintains the overall data distribution. This technique improved model performance by reducing the impact of extreme values without reducing the dataset size.

2.4. Categorical Encoding

- Categorical encoding is the process of converting categorical (text or label-based) variables into numerical values so that machine learning models can understand and process them.
- From the data for the categorical column conversions we will use onehot encoding technique for the nominals and for ordinals we will use ordinal encoding.If the dependent columns are categorical we will use Label encoding which are the best techniques.

2.5.1. One Hot Encoding:

- From the categorical columns the Nominals we are having are gender, Partner, Dependents, PhoneService,MultipleLines,OnlineSecurity, Online Backup, DeviceProtection,TechSupport,StreamingTV,StreamingMovies,PaperlessBilling , Sim_Providers.
- For a categorical variable with *N* unique categories, One-Hot Encoding creates *N* new columns (dummy variables).Each column corresponds to one category, and takes a value of:

  1 → if the record belongs to that category

  0 → otherwise
- Only one of these columns will have a 1 per row that's why it's called **"one-hot"** encoding.

2.5.2. Ordinal Encoding:

- From the categorical columns the ordinals we are having is Internet Service, Contract, Payment Method column.
- Ordinal Encoding replaces each category with an integer value that represents its order or rank.
- Ordinal Encoding should be used when:
- The categorical variable has a clear order or ranking (ordinal data).
- The order between categories has meaningful significance for the model.

3. Feature Selection

- In this project, feature selection was applied only to numerical features**.** Statistical techniques and variance-based analysis were used to evaluate the importance of numerical variables and remove less informative features.
- Categorical features were not directly subjected to feature selection. Instead, they were handled through categorical encoding techniques to convert them into numerical format for model training.
- This approach was chosen to avoid incorrect statistical assumptions on categorical data and to maintain interpretability and stability in the model.

3.1. Filter Methods:

- Filter methods are one of the main types of feature selection techniques used in machine learning**.**
- In simple terms, they filter out irrelevant or less important features before training the model — just like a sieve that removes noise from useful information.
- We have two techniques in filter methods**:**

i. Constant Technique:

- This technique filter out irrelevant or less important features before training the model based on the variance of every independent column.
- So if the variance of the column is 0 we will remove that column as it does not have any affect on the model data**.**

ii. Quasi Constant Technique:

- This technique filter out irrelevant or less important features before training the model based on the variance of every independent column.
- So if the variance of the column is 0.1 we will remove that column as it does not have any impact on the model data.

Note: After checking the columns, as the variance of every individual numeric column is not either 0 or 0.1. So I have not implemented any filter technique in this model.

The variance of a dataset is defined as:

$$Var(X)=n-1/\sum(xi-\bar{x})2$$

- Xi->each data value
- $\bar{x}$->mean of the data
- n->number of observations

4.Merging

- As we know for the above processes we have splitted the data to categorical and numerical for checking the strength of the data for the model building.
- As we completed the checking the strength of the data, inorder to train the model we need to combine both numerical and categorical data for training and testing.
- So we will merge both numerical categorical data for both training and testing using pandas concat function.

5. Data Balancing

- In real-world datasets, the target variable often has unequal class distributions.
- Such imbalance can cause models to become biased toward the majority class, leading to poor performance on the minority class — which is often the most important one.
- So to solve this data imbalance problem we will use some of the techniques to balance the data like oversampling, undersampling and smote(synthetic minority over sampling technique).
- In this project we will use the smote technique which is an advanced technique of oversampling.
- The SMOTE algorithm works as follows:
- For each minority class sample, find its k nearest neighbors (usually k=5).
- Randomly choose one or more of these neighbours.
- Generate a new synthetic sample somewhere between the chosen sample and its neighbour.

Mathematically:

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

Where:

- x_i = minority instance
- x_{zi} = one of its k nearest neighbors
- δ = random number between 0 and 1

6. Feature Scaling

- Feature Scaling is a data preprocessing technique used to normalize or standardize the range of independent variables or features of data before training a model.
- Machine learning algorithms perform better and converge faster when numerical features are on a similar scale.
- For feature scaling I have used standard scalar technique which is also known as z-score normalization used in model development.

Each feature value x is transformed using:

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

Where:

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

7. Model Training

- Finally the data is ready for model training.
- Since the dependent column in the data is having binary data we will use Binary classification.
- For the classification we have several algorithms(KNN, Naive Bayes, Random Forest ,Decision Tree, Logistic Regression).so we will use all the classification algorithms in the model training and train them .finally the algorithm which gives the best result will be taken as the final algorithm and build the model with that algorithm.
- Using Auc and Roc curves we will finalize the best algorithm.
- ROC curve shows how your model's performance changes as the decision threshold changes.
- AUC gives a single score to summarize how well the model distinguishes positive from negative classes.
- AUC (Area Under the Curve) represents the degree or measure of separability between the classes.
- It tells us how well the model can distinguish between churned and non-churned customers.
- The ROC curve is a graphical plot that illustrates the performance of a classification model at various thresholds.

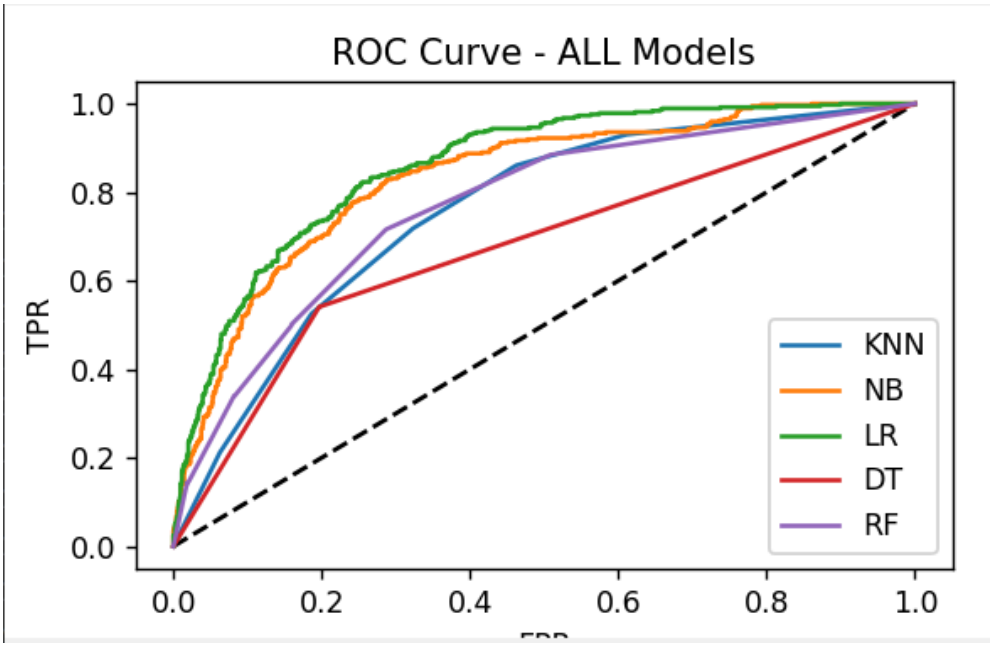
It plots:

- **True Positive Rate (TPR)** — also known as **Sensitivity or Recall**

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

- **False Positive Rate (FPR)** — also known as **1 - Specificity**

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



- Since the Auc score is having more for Logistic regression.we will build the model using Logistic Regression.Before tuning we got 75 % of accuracy.

8. Hyperparameter Tuning

- Hyperparameter tuning is the process of the selecting best combination of hyperparameters that maximize model performance.
- After the algorithm is finalized for model building we will find the best parameters of the algorithm for better performance of the model.
- Finding the best parameters can be done using Grid search cv and Randomized search cv.
- Grid Search tests all possible combinations of specified hyperparameter values and selects the combination that performs best using cross-validation**.**

- In Randomized search cv instead of trying all combinations, it randomly samples a fixed number of parameter combinations from the grid.
- So for our model for finding the best parameters I used Grid Search cv.
- GridSearchCV (Grid Search with Cross-Validation) is an automated method to:
- Search through a grid (set) of hyperparameter combinations
- Train models for each combination
- Evaluate performance using cross-validation
- Return the best parameters and the best model

9. Best Model

- Logistic Regression predicts the probability that an observation belongs to a particular class.
- The output is between 0 and 1 or Yes or No. Where, Yes refers to 1 and No refers to 0.
- Once the model is trained with this algorithm we will evaluate the algorithm on the test data set using confusion matrix and classification report and accuracy of the model.
- After tuning the model accuracy is 84 %.
- Once it is done the best model is saved using pickle for model deployment.

Formula:

$$P(Y=1|X) = \frac{1}{1+e^{-(\beta_0+\beta_1 X_1+\beta_2 X_2+\ldots+\beta_n X_n)}}$$

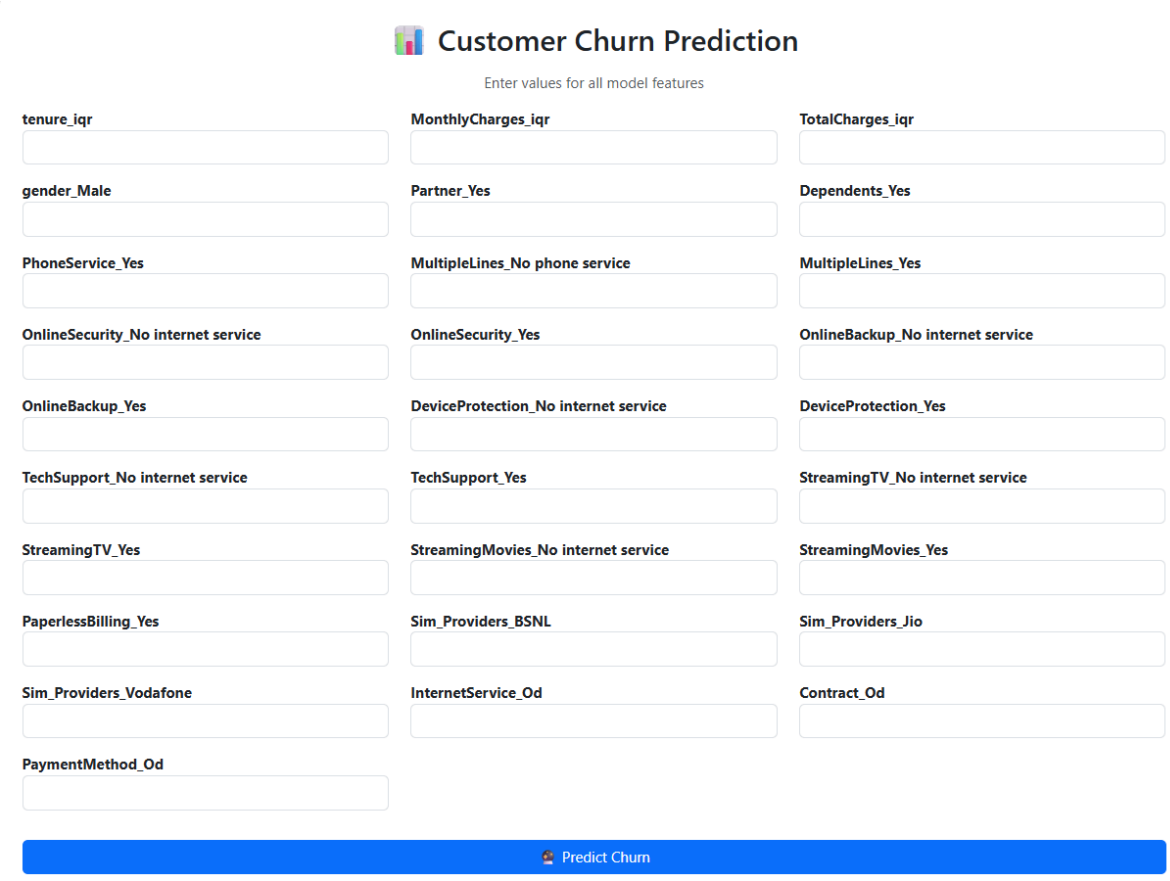
Where:

- $P(Y=1|X)$ → Probability that Y = 1 given X
- $\beta_0, \beta_1, \beta_2, \ldots$ → Model coefficients
- e → Exponential constant (≈ 2.718)

- The sigmoid function converts any real number into a value between 0 and 1.

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

10. Result



11. Conclusion

The Customer Churn Prediction project successfully demonstrates how machine learning can be leveraged to help telecom companies identify customers at risk of leaving their services. By analyzing key customer attributes — such as demographics, service usage patterns, contract type, and billing information — the model provides valuable insights into customer behavior and churn tendencies.The data is cleaned with all the possible best techniques and given the data to the model.After giving the data to the model we will train the model with all the machine learning algorithms.Using Auc and Roc curves I have finalized the model with the Logistic Regression algorithm. Now the model is predicting the output as churn w.r.to Yes or No with the accuracy of 84 percentage.

The project also includes a Flask-based web dashboard, providing an intuitive interface where users can input customer details and instantly view churn predictions. This integration bridges the gap between machine learning models and practical business decision-making.

12. Future Enhancement:

- Incorporate deep learning models (ANN) for higher accuracy.
- Integrate real-time data pipelines for continuous churn monitoring.
- Develop a visual analytics dashboard to track churn trends over time.
- Implementation with Deep Learning architectures like ANN, LSTM, or CNN to detect nonlinear and sequential patterns in customer data.

13. 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://scikitlearn.org/stable/modules/generated/sklearn.metrics.roc_auc_score.html
- https://medium.com/@dlikhitha99/all-methods-for-balancing-imbalanced-data-decfe1f4048d
- https://scikitlearn.org/stable/modules/generated/sklearn.preprocessing.StandardScaler.html
- https://scikitlearn.org/stable/modules/generated/sklearn.linear_model.LogisticRegression.html