CUSTOMER CHURN PREDICTION

Kanjula Ramakoti Reddy(22125014), Mallishetty Hima Varshith(22125020), Ram Kumar Singh(22125028).

Abstract- —In this project, we are going to predict whether a regular customer of a company is going to churn or retain using the product of that company. So, our main goal is to suggest the optimal machine learning algorithm for early client churn prediction. The analysis conducted uses a dataset of 7043 records,21 features, with preprocessing steps including one-hot encoding for the categorical variables and normalising numerical features. We used four different machine learning algorithms namely- Logistic Regression, K-Nearest Neighbors (KNN), Support Vector Machines (SVM), and Gaussian Mixture Models (GMM). Then, we are comparing the results for all models which we have taken and finding out the best model which can be used for customer churn prediction.

I. INTRODUCTION

Churning refers to the number of customers who stopped using a particular product. Every company wants that there customers' churning rate must be low. The reason behind the churning is that when the customer has got same type of product from other company with minimum cost as well as best quality. For company retaining existing customers is equally important as gathering new customers.

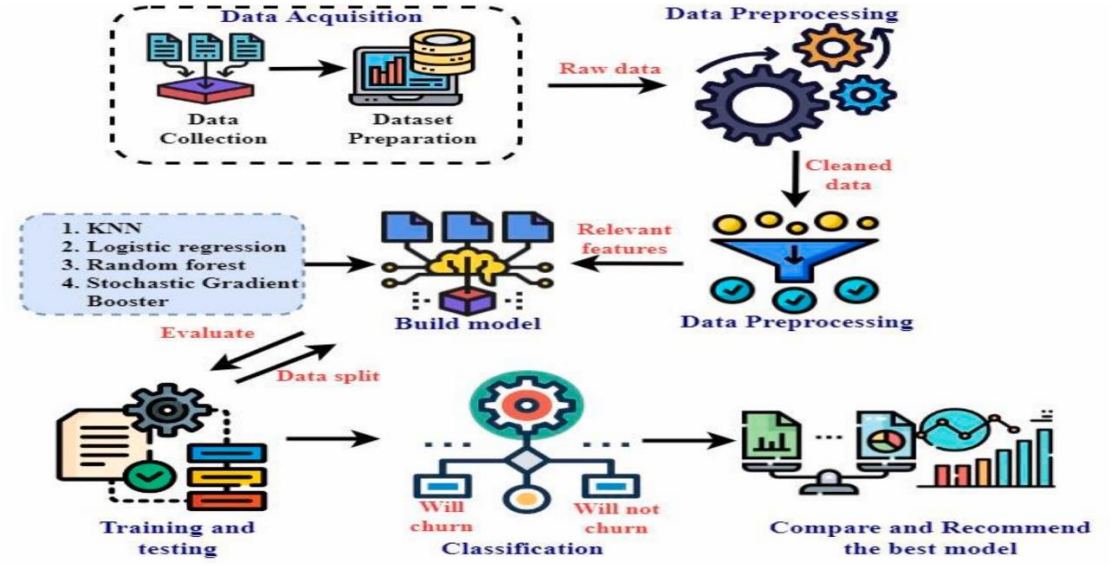
Generally customers will churn and opt for other company when they face the issues of high cost or low quality of product as compared to other company. If businesses can predict the customers who are going to churn then can segment those customers and will try to provide them better services to retain them. Hence, the churn prediction model has become compulsory for businesses in today's digitized economy.

An organization can achieve a high customer retention rate and maximize its revenue. If we can identify the possible churn early then the company can take the proactive measures to retain that customer and it will help in preventing revenue loss. Companies need to compete to attract new customers as well as to retain the existing ones.

Some of the main churn prediction algorithms which can identify the most necessary variables that are affecting the target variable are utilized in this project: Logistic Regression, K-Nearest Neighbors (KNN), Support Vector Machines (SVM), and Gaussian Mixture Models (GMM).

II. STUDY FLOW

Customer churn prediction has been performed using different methods, techniques, data processing, machine learning . The data set which we have taken is from the Kaggle with 7043 alternatives and 21 attributes was taken as input. Initially, to convert the categorical data into numerical data we have used one hot encoding as well as the label encoding and for the numerical attributes we have used knowledge MinMax Scaler to make range of all attributes [0,1]. Then, the information has been parted into two sections, train, and test set separately.



III. DATA PREPROCESSING

Effective data preprocessing is a crucial step in building a good and accurate machine learning model for customer churn prediction. The dataset used in this project consists of 7,043 records and 21 attributes, including both categorical and numerical features. As, the dataset we used doesn't have any missing data or irrelevant values. The preprocessing steps are as follows:

1. **Handling Nan or No Service Values:**
   We are replacing Nan or No Phone Services with No. If at any Data was given that it has no service it is considered as no.

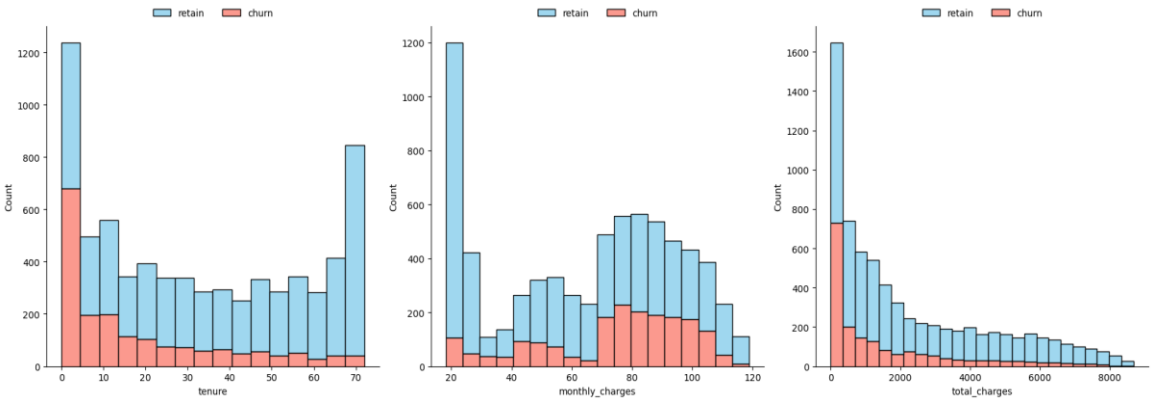
2. **Encoding Categorical Variables:**
   Categorical attributes were changed to numerical values using one-hot encoding and label encoding. This conversion was necessary because all the Algorithms used works on numerical data. One-hot encoding and label encoding enables to change categorical data to numerical data.

3. **Scaling Numerical Features:**
   To normalize the range of numerical features and avoid bias due to varying scales, the MinMaxScaler was applied. This technique scales all numerical values to a range of [0, 1], improving model performance and stability.

4. **Splitting the Dataset:**
   The data was divided into training and testing sets to evaluate model performance. A typical 75:25 split was used, ensuring that the training set contained a representative distribution of classes.

5. **Exploratory Data Analysis:**

   We checked the data and different features for the churn distribution. This is done to check if the feature is relevant or not and to have a good understanding about the all the features.



The Data Set contains the following features:

gender, senior_citizen, partner, dependents, tenure, phone_service, multiple_lines, internet_service, online_security, online_backup, device_protection, tech_support, streaming_tv, streaming_movies, contract, paperless_billing, payment_method, monthly_charges, total_charges, churn.

With tenure, monthly_charges, total_charges being numerical attributes and all the remaining being categorical attributes.

After One-hot encoding and label encoding the features are :

gender_Male, senior_citizen_1, partner_Yes, dependents_Yes, phone_service_Yes, multiple_lines_Yes, internet_service_Fiber optic, internet_service_No, online_security_Yes, online_backup_Yes, device_protection_Yes, tech_support_Yes, streaming_tv_Yes, streaming_movies_Yes, contract_One year, contract_Two year, paperless_billing_Yes, payment_method_Credit card (automatic), payment_method_Electronic check, payment_method_Mailed check.

IV. METHODS USED

The system involved in the analysis of customer churning uses different algorithms mentioned below:

(i) Logistic regression:

It is a supervised learning method which is widely used for binary classification. The output predicts the probability that a given input belongs to one of the two predefined category.

It uses the sigmoid or logistic function to transform the linear combination of the features into the probability values ranging from [0,1]. The threshold is chosen which is usually 0.5. If the predicted probability is above this threshold, the input is classified into one category; otherwise, it's classified into the other. The probability that the input belongs to a category is:

$$P(x) = \frac{1}{e^{-(\beta 0 + x * \beta 1)}}$$

Here the loss function is the cross entropy. Log loss for the k th point is :

$L_K = - y_K * \ln(p_K) - (1 - y_K) * \ln(1 - p_K)$ For the parameter estimation minimizing the Loss function w.r.t to $\beta 0$ and $\beta 1$

$$\frac{\partial L}{\partial \beta_0} = 0, \quad \frac{\partial L}{\partial \beta_1} = 0$$

Logistic regression is widely used because it's simple, efficient, and interpretable. It's particularly useful when the relationship between the independent variables and the dependent variable is linear, and the dependent variable is binary. Here there are many features so we will use a computer program to solve for coefficients.

(ii) K-Nearest Neighbors (KNN):

K-Nearest Neighbors (KNN) is an example of a instance-based learning algorithm(i.e., it remembers training data rather than learning specific patterns) in which we are taking the k nearest neighbour for the data-point

and taking the majority of the classes as its final category. For the regression tasks we take the average of all the values of k-nearest neighbour and update it as the final answer for the new data-point. To find the nearest neighbors we can use the measurements like Euclidean distance or manhattan distance. We can also take all the neighbors but along with their weights which is inversely proportional to the euclidean distance or manhattan distance between them. Let x1 … xk denote the k instances from training examples that are nearest to xq.

$$\hat{f}(x_q) \leftarrow \underset{v \in V}{\operatorname{argmax}} \sum_{i=1}^{k} \delta(v, f(x_i))$$

$$\text{where } \delta(a,b) = 1 \quad \text{if } a = b$$

$$= 0 \quad \text{otherwise}$$

The inductive bias of KNN is that all the data-points which are of same category belongs to the same cluster or the are near as compare to the data-points which are of different category.

(iii) Support vector machines (SVM)

Support Vector Machine (SVM) is a supervised learning algorithm that is primarily used for classification tasks. The objective of SVM is to find the optimal hyperplane that separates data points of two classes in a high-dimensional feature space. SVM is effective in high-dimensional spaces and is capable of handling both linear and non-linear classification problems.

For a dataset $\{(\mathbf{x}_i, y_i)\}_{i=1}^{N}$, where each feature vector $\mathbf{x}_i \in \mathbb{R}^n$ belongs to one of two classes, $y_i \in \{-1, +1\}$, the goal is to find a hyperplane defined by:

$$\mathbf{w} \cdot \mathbf{x} + b = 0$$

Where:

w is the weight vector that defines the orientation of the hyperplane.

b is the bias term that shifts the hyperplane.

SVM aims to find the hyperplane that maximizes the margin, defined as the distance between the hyperplane and the nearest data points from both classes (the support vectors). The margin M

M is given by:

$$M = \frac{2}{\|\mathbf{w}\|}$$

Thus, the larger the margin, the better the classifier is expected to generalize. To maximize this margin, we need to minimize the term $\frac{1}{2}\|\mathbf{w}\|^2$.

The optimization problem for SVM becomes:

$$\min_{\mathbf{w},b} \frac{1}{2}\|\mathbf{w}\|^2$$

Subject to the constraint that all data points are correctly classified:

$$y_i(\mathbf{w} \cdot \mathbf{x}_i + b) \geq 1 \quad \forall i$$

This ensures that each data point $\mathbf{x}_i$ lies on the correct side of the margin, meaning it is classified correctly by the hyperplane.

(iv) Gaussian Mixture Model (GMM):

Gaussian Mixture Model (GMM) is an unsupervised learning algorithm based on probabilistic clustering. It assumes that the data is generated from a mixture of several Gaussian distributions with unknown parameters. GMM uses the Expectation-Maximization (EM) algorithm to estimate these parameters iteratively.

The probability of a data point belonging to a cluster is calculated as:

$$P(x|\theta) = \sum_{k=1}^{K} \pi_k \cdot \mathcal{N}(x|\mu_k, \Sigma_k)$$

Where:

- K is the number of clusters.
- π_k is the mixing coefficient for the k^{th} Gaussian component.
- μ_k and Σ_k are the mean and covariance matrix of the k^{th} Gaussian component.

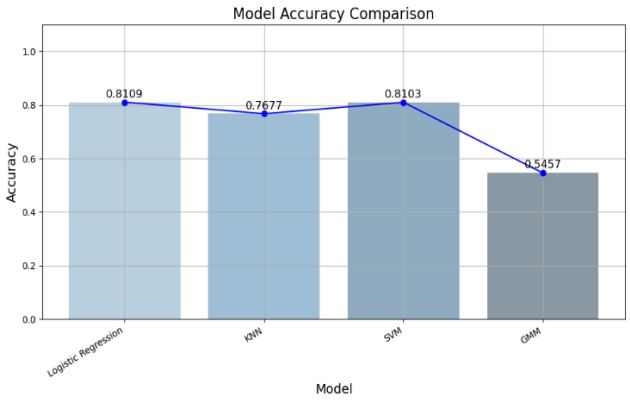
GMM is flexible and can capture complex data distributions, making it suitable for tasks such as density estimation and clustering. However, it assumes that the data follows a Gaussian distribution, which may not hold true in all scenarios.

V. RESULTS

We trained all four models using one-hot encoded data, which had 23 features after encoding. To evaluate performance, we split the dataset into training and testing sets using a 75:25 ratio. The evaluation metrics included accuracy, F1 score, precision, and recall.

The results are summarized in the table below.

model	accuracy	precision	recall	f1_score
Logistic Regression	0.810903	0.645161	0.577778	0.609613
KNN	0.767746	0.547126	0.528889	0.537853
SVM	0.810335	0.664773	0.520000	0.583541
GMM	0.545713	0.116228	0.117778	0.116998

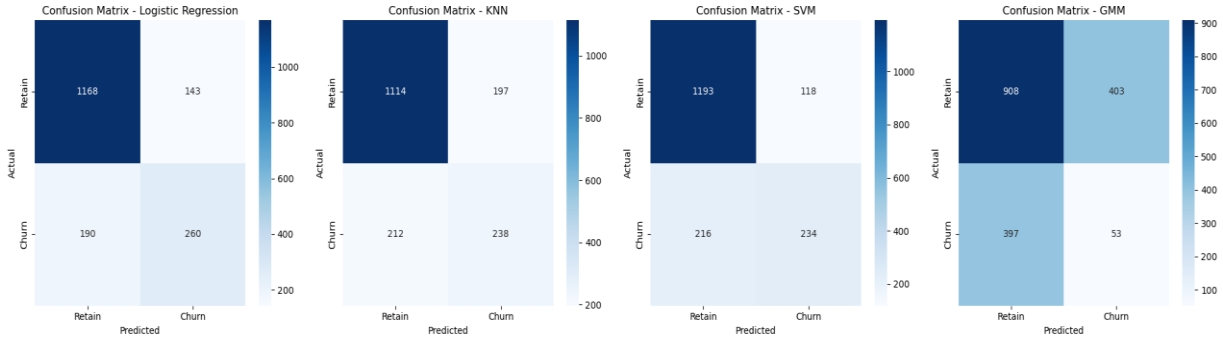


Logistic Regression is the best model overall. It achieves the highest accuracy of 81.09% and maintains a good balance between precision (64.5%) and recall (57.8%), resulting in the highest F1-score (60.96%). This balance means it correctly predicts both churn and non-churn customers fairly well.

SVM is a close second, with similar accuracy (81.03%) and slightly higher precision (66.5%), but its recall is lower (52.0%), which lowers its F1-score.

KNN performs reasonably with an accuracy of 76.77%, but both its precision and recall are lower compared to Logistic Regression and SVM, leading to a lower F1-score (53.78%).

GMM has the weakest performance across all metrics, with accuracy at 54.57% and very low precision and recall, indicating it struggles to classify the data accurately.



In summary, ***Logistic Regression*** is the most reliable choice based on these results.

VI. CONCLUSION & INSIGHTS

Since, We have kept the model simple, the following insights are helpful to improve the model.

To improve the performance of our churn prediction model, several strategies can be explored. One major challenge we addressed was data imbalance, which we can avoid by using SMOTE. This technique helps by generating synthetic data points for the underrepresented class, allowing the model to better recognize patterns in both the majority and minority classes.

Additionally, we could enhance our model by using hyperparameter tuning, particularly with L1 and L2 regularization. These techniques help to prevent overfitting by reducing the complexity of the model and ensuring it focuses on the most important features. Another area for improvement is feature engineering, where we can refine the input features by adding more meaningful ones or eliminating unnecessary ones, ultimately helping the model make better predictions.

We could also experiment with different distance metrics for KNN and apply cross-validation to ensure that our models are consistent and perform well across various data splits. By making these adjustments, we can further enhance the accuracy and reliability of our churn prediction model.

Note: The Dataset used in the project is from Kaggle.

VII. REFERENCES

[1] B. Prabadevi, R. Shalini, B.R. Kavitha, Customer churnng analysis using machine learning algorithms.International Journal of Intelligent Networks 4 (2023) 145–154.

[2] Omar Adwan, Hossam Faris, Khalid Jaradat, Osama Harfoushi, Nazeeh Ghatasheh, Predicting customer churn in telecom industry using multilayer preceptron neural networks: modeling and analysis, Life Sci. J. 11 (3) (2014).

[3] S. Babu, N.R. Ananthanarayanan, V. Ramesh, A study on efficiency of decision tree and multi layer perceptron to predict the customer churn in telecommunication using WEKA, Int. J. Comput. Appl. 140