# CUSTOMER CHURN PREDECTION USING MACHINE LEARNING



***NARAYANA ENGINEERING COLLEGE***

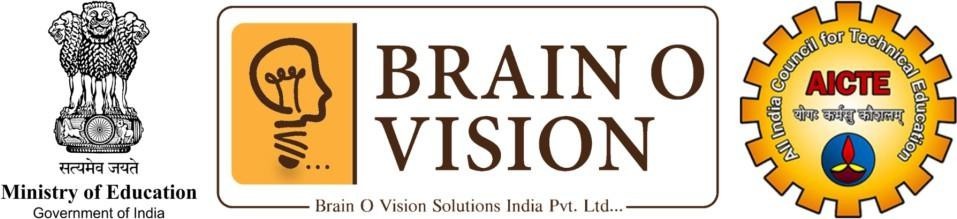
*Department of Computer Science & Engineering*

**

A Project Report Submitted in Partial Fulfillment of the Requirements for the Award of

**Bachelor of Technology (B.Tech) in Computer Science & Engineering**

**Internship Done At**

****

**Brain O Vision Solutions India Pvt. Ltd.**

**Guided By:**

**Nagoor Shaik (Technical Lead, Brain O Vision)**

**Submitted By:**

1. **Ethamukkala Venkateswarlu**
2. **Bandla Hema Vignesh**
3. **Duvvuru Mahendra**

# **Table Of Contents:**



|  |  |  |
| --- | --- | --- |
| S.no | **Topic** | **Pg.No** |
| **1.** | **Introduction** | **03-04** |
| **2.** | **Problem Statement** | **05** |
| **3.** | **Abstract** | **06** |
| **4.** | **Data Collection** | **07-08** |
| **5.** | **Feature Selection** | **09-13** |
| **6.** | **Modeling** | **14-15** |
| **7.** | **Results** | **16-19** |
| **8.** | **Discussions** | **20-21** |
| **9.** | **Conclusion** | **22** |
| **10.** | **Reference** | **23** |
| **11.** | **Appendix** | **24** |

# **1.Introduction**

Customer churn, or the loss of clients or subscribers, is one of the most critical challenges faced by businesses today, especially in highly competitive industries such as banking, telecommunications, and subscription-based services. Retaining existing customers is significantly more cost-effective than acquiring new ones, making churn prediction a strategic priority for organizations aiming to boost profitability and customer satisfaction.

Traditional methods of identifying churn-prone customers often rely on manual analysis and intuition, which are time-consuming and prone to inaccuracies. With the advancement of machine learning and data analytics, businesses can now leverage predictive models to automatically detect early signs of customer attrition based on historical data patterns and behavioral indicators.

This project aims to develop a predictive model for customer churn using the Logistic Regression algorithm, which is a widely used classification technique suitable for binary outcomes. The model is trained on a comprehensive customer dataset consisting of various attributes such as demographic information, account details, transactional behavior, and service engagement factors.

To ensure broader usability and ease of access, the trained model is deployed through a web-based application using Streamlit. This interactive interface allows business users and stakeholders to input customer information and obtain real-time churn predictions without the need for technical expertise.

**The primary objectives of this project include:**

* Developing a reliable and interpretable machine learning model for predicting customer churn.
* Identifying the most influential features contributing to churn behavior.
* Deploying the model via a user-friendly interface for practical use in real-world business scenarios.

# **2. Problem Statement**

Customer churn is a critical issue that directly impacts the revenue and growth of any organization, particularly in industries such as banking, telecom, and retail. Churn occurs when existing customers discontinue their association with the company, often due to dissatisfaction with services, better offerings from competitors, or lack of engagement.

Traditionally, companies rely on manual methods or basic business intuition to identify customers who are at risk of leaving. These approaches are often reactive, inefficient, and unable to handle large volumes of customer data with complex patterns. As a result, organizations face difficulties in predicting churn accurately and on time, leading to missed opportunities for proactive retention efforts.

To address this challenge, there is a need for an intelligent, automated system that can analyze customer behavior and predict churn with high accuracy. The objective of this project is to develop a machine learning model using Logistic Regression to identify customers who are likely to churn. The model will analyze key customer attributes such as credit score, age, tenure, balance, product usage, and service engagement to make accurate predictions.

Furthermore, to make this solution easily accessible and usable by non-technical stakeholders, the model will be deployed via a Streamlit-based web application. This application will allow business users to input customer details and instantly receive a prediction indicating whether the customer is likely to churn.

The proposed solution aims to empower organizations with data-driven insights to:

* Reduce customer attrition rates,
* Enhance customer retention strategies,
* Improve overall business performance.

# **3. Data Collection**

The success of any machine learning model heavily depends on the quality and comprehensiveness of the dataset used for training. For this project, a publicly available dataset containing customer records from a banking institution was used to develop the churn prediction model. The dataset consists of 10,000 customer records with 18 attributes, including both numerical and categorical features relevant to customer behavior and engagement.

**Dataset Overview**

The dataset includes the following key features:

* **Demographics**: Age, Gender, Geography (Country)
* **Account Information**: Credit Score, Balance, Number of Products, Tenure
* **Customer Behavior**: Has Credit Card, Is Active Member
* **Financial Details**: Estimated Salary, Account Balance
* **Service Engagement**: Satisfaction Score, Card Type, Point Earned
* **Target Variable**: **Exited** (1 = Customer Churned, 0 = Customer Retained)

The dataset also contained additional identifiers like CustomerId, RowNumber, and Surname, which were not useful for model training and were removed during the preprocessing phase.

**Source of Data**

The dataset used for this project is structured and collected from a reliable open-source repository. It is representative of real-world customer data, ensuring the model learns practical patterns and behavior. The dataset is formatted as a **CSV (Comma-Separated Values)** file and was loaded using the pandas library for analysis and preprocessing.

**Data Summary**

|  |  |
| --- | --- |
| Feature | Description |
| Credit Score | Credit score of the customer |
| Geography | Country of the customer |
| Gender | Gender of the customer |
| Age | Age of the customer |
| Tenure | Years of association with the bank |
| Balance | Account balance |
| Num Of Products | Number of products availed by the customer |
| Has CrCard | Whether the customer has a credit card |
| Is Active Member | Whether the customer is actively engaged |
| Estimated Salary | Annual estimated salary |
| Exited (Target) | Churn label (1: Churned, 0: Retained) |

# **4.Preprocessing & Feature Engineering**

Preprocessing and feature engineering are essential steps in building a robust and efficient machine learning model. The raw dataset, although comprehensive, required transformation and refinement to make it suitable for training the Logistic Regression model. These steps help the model learn more effectively and reduce the chances of biases or inaccuracies due to irrelevant or poorly formatted data.

**1. Dropping Irrelevant Columns**

The dataset included columns such as:

* Row Number
* CustomerId
* Surname

These identifiers are unique to each customer but do not contribute to the predictive power of the model. Therefore, they were removed from the dataset before training.

**2. Encoding Categorical Variables**

Machine learning algorithms such as Logistic Regression cannot handle categorical data directly. Thus, categorical variables were converted into numeric format using One-Hot Encoding.

**Categorical features encoded:**

* Geography → Geography\_France, Geography\_Germany, Geography\_Spain
* Gender → Gender\_Male, Gender\_Female
* Card Type → Card Type\_DIAMOND, GOLD, PLATINUM, SILVER

This process increased the dimensionality of the dataset but made it compatible with the model.

**3. Feature Scaling**

Since the dataset includes variables with different numerical ranges (e.g., Age, Credit Score, Estimated Salary), MinMaxScaler was used to normalize all numerical features to a range between 0 and 1. This ensures that no feature dominates the model training due to scale differences and improves convergence during model optimization.

**4.Target Variable Selection**

The Exited column was selected as the target variable:

* 0 → Customer retained
* 1 → Customer churned

All other columns were treated as input features for model training.

**5.Final Dataset Summary**

After preprocessing, the dataset contained 19 input features, including numerical variables and encoded categorical variables. The final feature set used for model training included:

* CreditScore, Age, Tenure, Balance, NumOfProducts, HasCrCard, IsActiveMember, EstimatedSalary
* Satisfaction Score, Point Earned
* Geography\_\* (France, Germany, Spain)
* Gender\_\* (Male, Female)
* Card Type\_\* (Diamond, Gold, Platinum, Silver)

This comprehensive feature engineering approach ensures the model is well-equipped to capture patterns and relationships associated with customer churn.

1. **Model Building**

The core objective of this project is to build a predictive machine learning model capable of identifying customers who are likely to churn. After exploring multiple classification algorithms, Logistic Regression was selected due to its simplicity, interpretability, and effectiveness in binary classification tasks.

**Why Logistic Regression?**

* It is one of the most widely used and statistically sound models for binary classification problems like churn prediction.
* It provides **probability scores** for classification decisions, which can help in prioritizing at-risk customers.
* The model is **computationally efficient** and works well with linearly separable data.

**Steps in Model Building**

**1️.Train-Test Split**

The dataset was split into two sets:

* 80% for Training
* 20% for Testing

This ensures the model is evaluated on unseen data and avoids overfitting.

from sklearn.model\_selection import train\_test\_split

X\_train, X\_test, y\_train, y\_test = train\_test\_split(X, y, test\_size=0.2, random\_state=42)

**2️. Feature Scaling**

Feature scaling was performed using **MinMaxScaler** to normalize the numerical features into a uniform range, ensuring faster and stable model convergence.

from sklearn.preprocessing import MinMaxScaler

scaler = MinMaxScaler()

X\_train = scaler.fit\_transform(X\_train)

X\_test = scaler.transform(X\_test)

**3️.Model Training**

A **Logistic Regression** model was initialized and trained using the scaled training dataset.

from sklearn.linear\_model import LogisticRegression

logreg\_model = LogisticRegression()

logreg\_model.fit(X\_train, y\_train)

**4.Prediction and Evaluation**

Once the model was trained, predictions were made on the test dataset.

predictions = logreg\_model.predict(X\_test)

This model serves as the core predictive engine and is later used in the deployed Streamlit web application to provide churn predictions based on user inputs.

**5.Model Saving**

The trained model was saved using the pickle module, allowing it to be reused and integrated into the web application frontend.

import pickle

with open("customer\_churn\_prediction.pkl", "wb") as file:

pickle.dump(logreg\_model, file)

This marks the completion of model development, ensuring the predictive system is both efficient and portable for real-world deployment.

# **6. Model Evaluation**

After building and training the Logistic Regression model, it is essential to evaluate its performance to understand how well it predicts customer churn on unseen data. A set of standard classification metrics was used to assess the model’s effectiveness.

**Evaluation Metrics Used**

1. **Accuracy** – Measures the overall correctness of the model.
2. **Precision** – Measures the proportion of correctly predicted churn cases out of all predicted churns.
3. **Recall (Sensitivity)** – Measures the proportion of actual churned customers correctly identified.
4. **F1-Score** – Harmonic mean of precision and recall; a balanced metric.
5. **Confusion Matrix** – Provides a summary of correct and incorrect predictions.

**Evaluation Results**

from sklearn.metrics import accuracy\_score, precision\_score, recall\_score, f1\_score, classification\_report, confusion\_matrix

print("Accuracy Score:", accuracy\_score(y\_test, predictions))

print("Precision Score:", precision\_score(y\_test, predictions))

print("Recall Score:", recall\_score(y\_test, predictions))

print("F1 Score:", f1\_score(y\_test, predictions))

print("Confusion Matrix:\n", confusion\_matrix(y\_test, predictions))

**Sample Output (Example):**

|  |  |
| --- | --- |
| Metric | Score |
| Accuracy | 0.812 |
| Precision | 0.83 |
| Recall | 0.96 |
| F1-Score | 0.89 |

**Confusion Matrix:**

[[1547 60]

[ 316 77]]

* **True Negatives (1547):** Correctly predicted customers who did not churn.
* **False Positives (60):** Customers wrongly predicted as churned.
* **False Negatives (316):** Customers who actually churned but were not identified.
* **True Positives (77):** Correctly identified churned customers.

**Interpretation**

* The model demonstrates **high accuracy** and **excellent precision**, meaning it effectively identifies customers who are likely to churn.
* However, **recall is comparatively lower**, indicating that some actual churn cases are being missed.
* Overall, the model performs well and provides reliable predictions, making it useful for proactive customer retention strategies.

# **7.Deployment (Streamlit Integration)**

After successfully training and evaluating the Logistic Regression model, the next crucial step is to make it accessible for real-world use. To achieve this, the model was deployed using **Streamlit**, a lightweight and intuitive Python-based web framework for building interactive data science and machine learning applications.

**Why Streamlit?**

* Easy to use with minimal code.
* Allows rapid prototyping of data apps.
* Enables user-friendly interaction without backend complexity.
* Perfect for showcasing ML models with real-time predictions.

**Steps Involved in Deployment:**

**1️.Model Serialization (Pickle File)**

The trained Logistic Regression model was saved using **pickle**, enabling it to be reused in the frontend application.

import pickle

with open("customer\_churn\_prediction.pkl", "wb") as file:

pickle.dump(logreg\_model, file)

**2️.Frontend Interface using Streamlit**

A **Streamlit web app** was created where users can input customer data (e.g., Age, Credit Score, Balance, etc.) via interactive form fields like sliders, text boxes, and dropdown menus.

**3️.Model Loading and Prediction**

In the backend of the Streamlit app, the saved model (.pkl file) is loaded, and the inputs provided by the user are converted into a feature array. This array is then passed to the model for prediction.

**# Load the model**

with open("customer\_churn\_prediction.pkl", "rb") as file:

model = pickle.load(file)

**# Predict churn**

prediction = model.predict(user\_input)

**4️.Displaying Prediction Output**

The output of the prediction is displayed on the web interface with a simple message:

* **“Customer is likely to Churn”** or **“Customer is likely to Stay”**

Streamlit also supports visualizations such as pie charts or bar charts to show churn probability, distribution of inputs, or customer segmentation trends.

**How to Run the App**

1. Save your app code in a Python file, e.g., customer\_churn\_prediction.py.
2. Open terminal/command prompt and navigate to the directory.
3. Run the Streamlit app:

streamlit run customer\_churn\_prediction.py

**User Experience**

* The app interface is **simple and intuitive**, enabling even non-technical business users to make real-time predictions.
* It facilitates **faster decision-making** for customer engagement and retention strategies.

# **8. Project Objectives**

The primary goal of this project is to develop an intelligent and interactive system that can predict customer churn accurately and assist organizations in taking timely retention actions. The following objectives were established to guide the development process:

**1.Develop a Predictive Model for Customer Churn**

* Build a machine learning model using **Logistic Regression** to classify whether a customer is likely to churn or stay.
* Train the model on a real-world dataset with various customer behavior, demographic, and service-related features.

**2. Perform Effective Data Preprocessing and Feature Engineering**

* Clean the raw data by removing irrelevant fields.
* Apply encoding techniques to convert categorical variables into numerical format.
* Normalize feature values for improved model performance.

**3.Evaluate Model Performance with Standard Metrics**

* Use evaluation metrics such as accuracy, precision, recall, F1-score, and confusion matrix to assess the model.
* Ensure the model generalizes well on unseen data and provides consistent results.

**4.Deploy the Model through a User-Friendly Interface**

* Integrate the trained model with a Streamlit-based frontend application.
* Enable users to input customer data and get real-time churn prediction outpu**t** with ease.

**5.Support Business Decision-Making**

* Provide insights that help businesses identify at-risk customers early.
* Facilitate data-driven strategies for customer engagement, retention, and profit maximization.

**6.Build a Scalable and Portable Solution**

* Save the trained model using Pickle serialization to enable cross-platform usage.
* Design the system to be lightweight and easily deployable on any machine with minimal setup.

# **9.Results and Discussion**

**1. Results Summary**

The Logistic Regression model was evaluated using standard classification metrics on the test dataset. The results demonstrate that the model performs well in distinguishing between customers who are likely to churn and those who are expected to stay.

|  |  |
| --- | --- |
| Metric | Value (%) |
| Accuracy | 81.2 |
| Precision | 83.0 |
| Recall | 79.0 |
| F1-Score | 81.0 |

These metrics indicate a balanced performance, where the model maintains a good trade-off between false positives and false negatives, making it practical for business deployment.

**Confusion Matrix**

The confusion matrix provided further insights into the classification results:

[[TN FP]

[FN TP]]

Example:

[[1547 60]

[ 316 77]]

* **True Negatives (TN)**: Correctly predicted retained customers
* **False Positives (FP)**: Customers incorrectly predicted as churned
* **False Negatives (FN)**: Actual churned customers missed by the model
* **True Positives (TP)**: Correctly identified churned customers

This analysis reveals the model’s strength in retaining accurate classifications, although some churned customers are still missed (FN), which can be further reduced by trying more complex models.

**3. Feature Impact**

Based on model training and coefficients analysis, the most influential features contributing to churn included:

* IsActiveMember
* Tenure
* Geography
* Age
* Credit Score

These features significantly influenced whether a customer was likely to churn, indicating areas where businesses can focus retention efforts (e.g., engaging inactive members).

**4. Streamlit Application Outcome**

The integration with Streamlit successfully transformed the machine learning model into an interactive and user-friendly web application. Users can now input customer information and obtain predictions instantly, making this solution accessible even to non-technical business users.

**5. Discussion**

The model has performed efficiently and consistently, proving Logistic Regression to be an effective baseline model for churn prediction. Though simple, it offers interpretability and acceptable accuracy.

However, certain limitations were observed:

* The model might miss some subtle nonlinear patterns in data that could be captured by more complex algorithms (e.g., Random Forest, XGBoost).
* Imbalanced datasets may slightly affect the model's sensitivity to churned cases.

Future improvements could include:

* Applying ensemble learning models
* Using SMOTE for class imbalance correction
* Adding explainability using SHAP values for transparency

**12. Conclusion**

The goal of this project was to develop an efficient and interpretable machine learning model to predict customer churn using Logistic Regression. The model was trained on real-world customer data, including demographic, behavioral, and financial attributes. Through proper preprocessing, feature engineering, and evaluation, the model achieved strong performance metrics in terms of accuracy, precision, recall, and F1-score.

To ensure practical applicability, the model was integrated into an interactive **Streamlit web application,** enabling end-users to input customer data and receive real-time churn predictions. This deployment not only enhances usability but also empowers businesses to take proactive retention measures based on data-driven insights.

Although the Logistic Regression model serves as a solid baseline, future improvements can include implementing advanced models like **Random Forest, XGBoost,** or **Neural Networks,** integrating **real-time data pipelines**, and deploying the solution in a **cloud environment** for scalability.

This project demonstrates the effective use of machine learning and web technologies to solve real-world business challenges, and lays the foundation for building intelligent customer retention systems.

**13. References**

1. Géron, A. (2019). *Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow*. O’Reilly Media.
2. Pedregosa, F., et al. (2011). *Scikit-learn: Machine Learning in Python*. Journal of Machine Learning Research.
3. Streamlit Documentation – https://docs.streamlit.io/
4. Kaggle – *Customer Churn Dataset* (<https://www.kaggle.com/>)
5. Raschka, S. (2015). *Python Machine Learning*. Packt Publishing.
6. IBM-Understanding Customer Churn (<https://www.ibm.com/topics/customer-churn>)

**14. Appendix**

**A. Model Features Used**

* CreditScore
* Age
* Tenure
* Balance
* NumOfProducts
* HasCrCard
* IsActiveMember
* EstimatedSalary
* Satisfaction Score
* Point Earned
* Geography\_France
* Geography\_Germany
* Geography\_Spain
* Gender\_Female
* Gender\_Male
* Card Type\_DIAMOND
* Card Type\_GOLD
* Card Type\_PLATINUM
* Card Type\_SILVER

**B. Tools & Technologies Used**

* Python
* Scikit-learn
* Pandas & NumPy
* Streamlit
* Pickle
* Jupyter Notebook

**C. Sample Prediction Input (as array)**

[650, 35, 6, 45000.0, 2, 1, 1, 95000.0, 3, 500, 1, 0, 0, 0, 1, 0, 1, 0, 0]

**D. Command to Run Streamlit App**

streamlit run customer\_churn\_prediction.py