Customer Churn in Telecom Industry Using Machine Learning

Group Number: 7
Names: Mariyam Member (100867858)

***Abstract*—Customer churn is a major issue in the telecom industry, where retaining customers is more cost-effective than acquiring new ones. In this project, I used machine learning to predict customer churn using the Telco Customer Churn dataset from Kaggle. After preprocessing, I trained five models: Logistic Regression, Random Forest, SVM, XGBoost, and a Deep Learning model using Keras. Evaluation was based on accuracy, precision, recall, F1-score, and AUC. The Deep Learning model achieved the best performance, with an accuracy of 80.2%, recall of 50.3%, and F1-score of 57.5%. Feature importance analysis showed that tenure, contract type, and monthly charges were the most significant predictors of churn. The results show that deep learning can effectively identify at-risk customers, offering telecom providers valuable insights to support retention strategies.**

I. Introduction

In the highly competitive telecom industry, customer retention plays a crucial role in long-term profitability. One of the biggest challenges companies face is customer churn — the rate at which users cancel their service. High churn directly affects revenue and increases customer acquisition costs, which are often significantly higher than the cost of retaining an existing user. As a result, predicting churn in advance allows telecom providers to take proactive steps in improving customer satisfaction and offering targeted retention strategies.

Machine learning has become a powerful tool for churn prediction, as it can detect patterns in customer behavior and identify users who are at risk of leaving. In this project, I aimed to build and compare multiple machine learning models to predict churn using a real-world dataset from Kaggle. The dataset included over 7,000 telecom customers with various service, demographic, and billing features.

I experimented with five models: Logistic Regression, Random Forest, Support Vector Machine (SVM), XGBoost, and a Deep Learning model built using Keras. Each model was evaluated based on accuracy, precision, recall, F1-score, and AUC. Our goal was to identify which algorithm performs best in terms of balancing overall prediction quality and minimizing false negatives — the customers who churn but go undetected.

The rest of this report is organized as follows: Section II discusses related work on churn prediction. Section III presents the dataset and preprocessing steps. Section IV explains the methodology and the machine learning models used. Section V presents and analyzes the results. Section VI outlines the interpretation and practical insight. Finally, Section VII concludes the report.

II. Background / Related Works

Customer churn prediction has been a widely studied problem in both academic research and industry, especially in sectors like telecommunications, banking, and e-commerce. Many organizations rely on churn prediction models to identify at-risk customers and take action before they leave.

Previous studies have shown that machine learning techniques can be highly effective for churn prediction. Logistic Regression is often used as a baseline model due to its simplicity and interpretability. More advanced models, such as Decision Trees, Random Forest, and Gradient Boosting, are frequently used because they can capture complex relationships in the data and often perform better on structured datasets. For example, Idris et al. (2019) used Gradient Boosting and achieved high accuracy on a telecom churn dataset [1], showing its effectiveness in identifying patterns in customer behavior.

Deep Learning has also gained attention in recent years for churn prediction tasks. While traditional models rely heavily on feature engineering, deep learning models can automatically learn complex patterns from data with minimal preprocessing. A simple neural network outperformed several traditional models when predicting churn in a large telecom dataset [2].

Despite these advancements, one of the ongoing challenges in churn prediction is dealing with imbalanced data — where the number of churned customers is much smaller than the number of retained ones. To handle this, many studies have emphasized the importance of evaluation metrics like recall and F1-score over plain accuracy [3], especially when the goal is to minimize false negatives (customers who churn but are missed by the model).

In this project, I aimed to apply and compare a range of these models — from traditional approaches like Logistic Regression and Random Forest, to more advanced ones like XGBoost and Deep Learning — using a real-world telecom dataset. The goal was to find the most effective model for accurately predicting churn while handling class imbalance and practical business concerns.

III. Dataset

For this project, I used the Telco Customer Churn dataset from Kaggle [4], a popular open-source data science platform. This dataset contains information about 7,043 customers from a telecom company and is commonly used for churn prediction tasks.

The dataset includes a total of 21 features, which represent customer demographics, service subscriptions, and billing information. The target variable is Churn, which indicates whether a customer has left the company (Yes) or stayed (No).

The features can be categorized into the following groups:

- Demographic Features: gender, SeniorCitizen, Partner, Dependents
- Account Info: tenure, Contract, PaymentMethod, PaperlessBilling
- Services Signed Up: PhoneService, InternetService, OnlineSecurity, StreamingTV, etc.
- Billing Information: MonthlyCharges, TotalCharges

TABLE I. FEATURE CATEGORIES AND EXAMPLES FROM THE TELCO DATASET

Feature Group	**Example Feature**
Demographics	Gender, Partner
Services	InternetService, Streaming TV
Billing Info	Monthly Charges, Total Charges

Before training the models, I performed several preprocessing tasks:

1. Handling Missing Values:

I replaced blank entries in the TotalCharges column with NaN and removed those rows. This affected a small portion of the data.

```
[ ]  df.replace(" ", np.nan, inplace=True)
     df.dropna(inplace=True)
     df.drop(['customerID'], axis=1, inplace=True)
```

2. Encoding Categorical Features:

I used LabelEncoder from Scikit-learn [6] to convert text-based categorical features (like Contract and PaymentMethod) into numerical values.

```
[ ]  le = LabelEncoder()
     for col in df.select_dtypes(include=['object']).columns:
         df[col] = le.fit_transform(df[col])
```

3. Feature Scaling:

I applied StandardScaler to scale continuous numeric features such as tenure, MonthlyCharges, and TotalCharges. This was important for models like SVM and the deep learning model.

```
[ ]  X = df.drop('Churn', axis=1)
     y = df['Churn']
     X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)
```

4. Train-Test Split:

The dataset was split into 80% training and 20% testing to evaluate model performance on unseen data

```
[ ]  scaler = StandardScaler()
     X_train_scaled = scaler.fit_transform(X_train)
     X_test_scaled = scaler.transform(X_test)
```

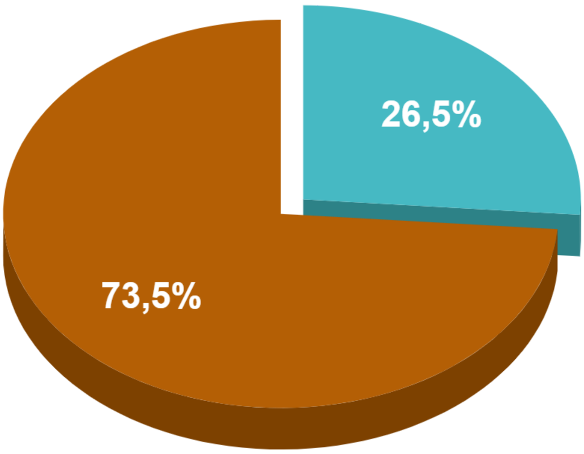


Figure 1. Churn Distribution in the Dataset. 26.5% churned, 73.5% did not

IV. METHODOLOGY

To predict customer churn, I implemented and evaluated five different machine learning models: Logistic Regression, Random Forest, Support Vector Machine (SVM), XGBoost, and a Deep Learning model built using Keras. Each model was trained on the same preprocessed dataset and evaluated using a consistent set of performance metrics.

A. Data pre-processing

Before training the models, I cleaned and transformed the data. The dataset was first checked for missing values — blank entries in the TotalCharges column were replaced with NaN and dropped. Then, I used LabelEncoder to convert categorical variables (such as Contract, PaymentMethod, and InternetService) into numerical values. Finally, continuous features like tenure, MonthlyCharges, and TotalCharges were scaled using StandardScaler to normalize the data. The data was split into training and testing sets using an 80/20 split, where 80% of the data was used for training and 20% was reserved for testing and evaluation.

B. Model Used

<u>Logistic Regression</u>
This model was used as a baseline. It's simple, interpretable, and provides quick insights into feature importance based on coefficients. It works best when the relationship between features and target is mostly linear.

<u>Random Forest</u>
An ensemble method based on decision trees. It helps capture non-linear patterns and is less prone to overfitting. It also provides feature importance scores, which are useful for interpretation.

Support Vector Machine (SVM)
SVM is effective in high-dimensional spaces and works well with datasets where the decision boundary between classes is not linear. I used the radial basis function (RBF) kernel in this case.

XGBoost
A gradient boosting model that is known for its performance on structured data. XGBoost also handles imbalanced data well [7] and usually outperforms simpler tree-based models due to its regularization features.

Deep Learning (Keras)
I built a simple feedforward neural network using the Keras Sequential API [5]. The model consisted of three hidden layers with ReLU activation and dropout to prevent overfitting. The final output layer used a sigmoid function for binary classification.

- Loss Function: Binary Crossentropy
- Optimizer: Adam
- Epochs: 20
- Batch Size: 32
- Dropout: 0.3 in each hidden layer

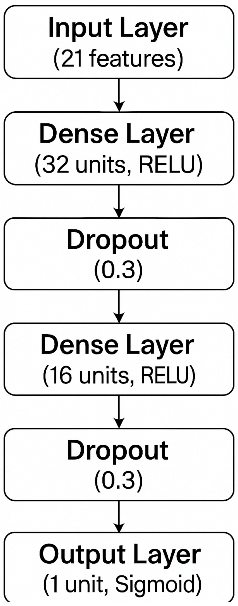


Figure 2. Deep Learning Model Architecture

C. *Evaluation Method*

To compare models fairly, I used multiple evaluation metrics:

- Accuracy: Overall correctness of predictions
- Precision: How many predicted churns were actually churn
- Recall: How many actual churns were correctly predicted
- F1-Score: Harmonic mean of precision and recall
- AUC (Area Under Curve): Measures how well the model distinguishes between classes

Since churn is an imbalanced classification problem, recall and F1-score were given more weight in final model selection.

V. Results and Analysis

The procedures of deep learning are different from the steps of machine learning, the main one is that deep learning can automatically extract features, so it does not need feature extraction or dimension reduction. This is because in CNN models, the convolutional layers serve as the filer of features.

TABLE II. Model Performance Comparison

Model	Accuracy	Precission	Recall	F1-Score	AUC
Logistic Regression	0.7889	0.6332	0.4893	0.5520	0.8300
Random Forest	0.7874	0.6383	0.4625	0.5364	0.8155
SVM	0.7938	0.6735	0.4358	0.5292	0.7891
XGBoost	0.7718	0.5874	0.4759	0.5258	0.8060
Deep Learning	0.8024	0.6714	0.5026	0.5749	0.8284

A. *Deep Learning Performance Overview*

Among all models tested, the Deep Learning model achieved the highest accuracy (80.2%), along with the best recall (0.5026) and F1-score (0.5749). That's important because in churn prediction, we care most about catching as many actual churners as possible — and recall directly reflects that.

Even though logistic regression had the highest AUC (0.8300), deep learning came super close (0.8284) and still performed better in the rest of the metrics. That's why I chose it as the final model. It was able to balance identifying actual churn cases while also avoiding too many false alarms.

B. *Confusion Matrix Analysis*

To break down the deep learning model's results, I looked at the confusion matrix:

TABLE III. Deep Learning Confusion Matrix Table

	Predicted: No	Predicted: Yes
Actual: No	941 (TN)	92 (FN)
Actual: Yes	186 (FN)	188 (FP)

- True Positives (TP): 188 customers who actually churned and were correctly identified by the model.

- True Negatives (TN): 941 customers who did not churn and were correctly classified as such.

- False Positives (FP): 92 customers who were incorrectly predicted to churn but did not.

- False Negatives (FN): 186 customers who churned but were incorrectly predicted to stay.

The total number of incorrect predictions (false positives and false negatives) amounts to 278 out of 1,407 predictions, which equates to approximately 19.8% error rate. While the model achieved a high accuracy (80.2%), this breakdown illustrates the trade-off between precision and recall in churn prediction tasks.

Notably, the number of false negatives (186) is slightly higher than false positives (92), which is critical in churn modeling, as failing to identify actual churners can lead to revenue loss. This further emphasizes the importance of optimizing recall and F1-score, rather than relying solely on accuracy, especially in the context of imbalanced datasets.

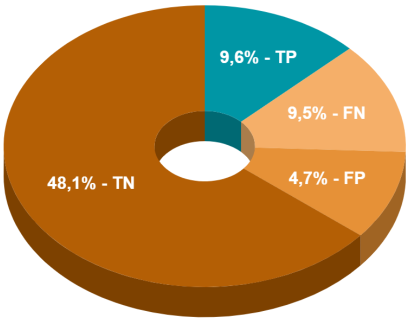


Figure 3. Confusion matrix breakdown for Deep Learning Model

C. Discussion of Results

The performance evaluation of all five machine learning models was based on key classification metrics: accuracy, precision, recall, F1-score, and area under the ROC curve (AUC). These metrics provide a comprehensive understanding of how well each model predicted customer churn, particularly considering the imbalanced nature of the dataset.

As shown in Table II, the Deep Learning model (Keras) demonstrated the best overall performance across most evaluation metrics. It achieved the highest accuracy (0.8024) and F1-score (0.5749), indicating its strong capability in correctly identifying both churn and non-churn cases. Moreover, its recall (0.5026) was notably higher than all other models, making it the most effective at detecting customers who were actually likely to churn — a critical requirement in churn prediction problems where false negatives can be costly.

The Logistic Regression model, although simpler and less complex, performed competitively with a high AUC of 0.8300, slightly higher than the deep learning model. This indicates that Logistic Regression maintained a good balance between sensitivity and specificity, despite lower recall.

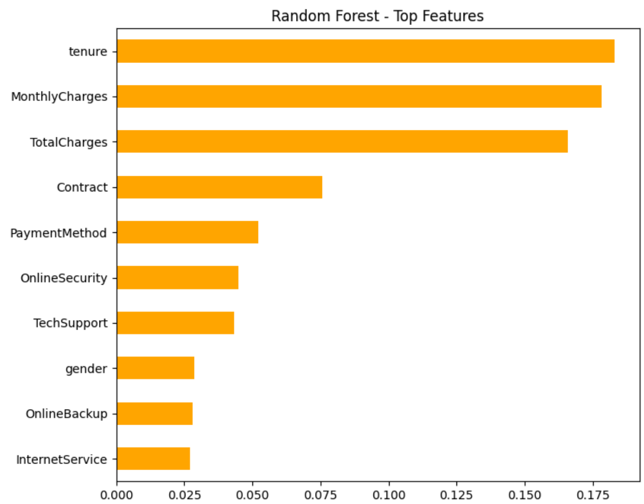
Random Forest and XGBoost also produced respectable results, with Random Forest showing slightly better precision and AUC than XGBoost. However, their recall values were lower than desired, meaning these models missed more actual churn cases. This tradeoff between precision and recall may lead to fewer false positives but at the cost of failing to identify churners.

The Support Vector Machine (SVM) model achieved strong precision (0.6736), which suggests it was conservative in its predictions and less likely to produce false positives. However, its recall (0.4358) and F1-score (0.5292) were among the lowest, showing a limitation in identifying true churn cases.

Overall, while all models showed reasonable performance, the Deep Learning approach was best suited for this classification task due to its superior ability to generalize, especially in the presence of complex, non-linear relationships within the data. It provided a balanced performance across all metrics and was particularly advantageous in handling the class imbalance, making it the most suitable model for predicting telecom customer churn.

D. Feature Importance Analysis

To gain further insights into what factors influence customer churn, I analyzed feature importance using both the Random Forest and XGBoost models. These models provide built-in importance scores for each feature based on how frequently and effectively they are used in decision splits.



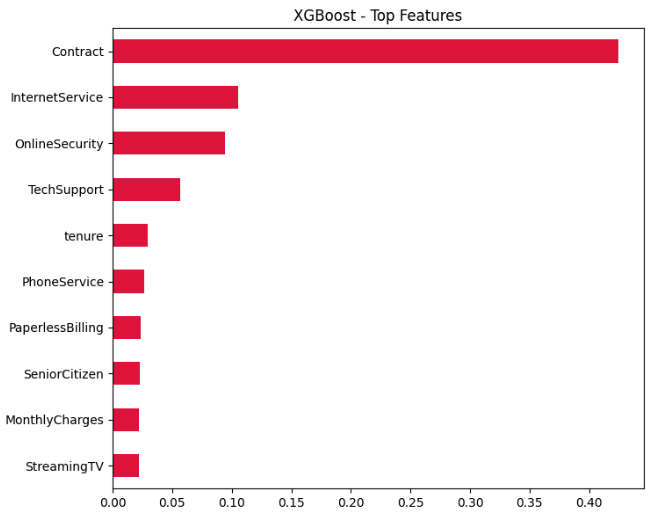


Figure 4. Top 10 Most Important Features Based on Random Forest and XGBoost

For the Random Forest model, the most influential features were:

1. Tenure, indicating how long a customer has been with the provider,
2. MonthlyCharges, representing the amount customers pay monthly, and
3. TotalCharges, the overall amount paid during the customer's lifecycle.

These three features suggest that customers with shorter tenure and higher ongoing costs are more likely to churn. The Contract Type feature also ranked highly, reflecting the likelihood of churn among users with monthly contracts versus long-term agreements.

In the XGBoost model, Contract Type emerged as the most significant predictor by a large margin, followed by InternetService, OnlineSecurity, and TechSupport. These results highlight that service-related attributes — especially whether customers receive technical support or security features — play a key role in retention.

Notably, while there was overlap in some key predictors (e.g., Contract, Tenure, MonthlyCharges), each model also revealed distinct patterns. This kind of interpretability is essential for informing business strategies, such as offering discounted rates to high-risk customers or promoting longer contract commitments.

VI. Analysis and Limitations

The results of this project demonstrate the strengths and limitations of various machine learning models in the context of predicting customer churn in the telecom industry. A comprehensive comparison of five models — Logistic Regression, Random Forest, SVM, XGBoost, and Deep Learning — revealed clear patterns in terms of predictive performance, model interpretability, and practical usability.

The Deep Learning model (Keras) outperformed traditional models in most evaluation metrics, particularly in recall (0.5026) and F1-score (0.5749). These metrics are especially important in churn prediction because they reflect the model's ability to identify customers who are actually at risk of leaving. In imbalanced datasets like this one, where churners are the minority class, optimizing for recall helps minimize false negatives — a priority when the cost of failing to retain a customer is high. While the deep learning model is more complex and less interpretable, its predictive strength makes it a suitable option for large-scale implementations where performance outweighs transparency.

Logistic Regression, despite being a simpler algorithm, achieved the highest AUC (0.8300) and a competitive F1-score. This suggests it struck a good balance between sensitivity and specificity, making it a strong baseline model for churn detection — especially when computational efficiency and interpretability are priorities.

Random Forest and XGBoost models offered moderate performance. Random Forest showed strong precision and general robustness, while XGBoost emphasized performance on structured data but lagged in recall. These models were especially useful for feature importance analysis, helping to identify critical churn indicators like contract type, tenure, and monthly charges — as visualized in Figure 4.

The SVM model had the lowest recall and F1-score, indicating that while it was precise, it missed a significant portion of actual churners. This may be due to the model's sensitivity to imbalanced data and the challenge of separating complex non-linear patterns in high-dimensional feature spaces.

Overall, the choice of model depends on the trade-offs between accuracy, interpretability, and resource availability. While deep learning delivered the best raw performance, simpler models still offer valuable insights — especially when paired with visual tools like feature importance plots and confusion matrices to explain decision-making.

VII. Conclusion

In this project, I built and compared five different machine learning models to predict customer churn in the telecom industry. After preprocessing the Kaggle dataset by handling missing values, encoding categorical features, and scaling numerical ones, I trained Logistic Regression, Random Forest, SVM, XGBoost, and a Deep Learning model using Keras.

Through evaluation, I found that the Deep Learning model performed the best overall. It achieved the highest accuracy (0.8024), recall (0.5026), and F1-score (0.5749), making it the most effective at identifying customers likely to churn. This was especially important given the class imbalance in the dataset. While simpler models like Logistic Regression also performed well in terms of AUC, they lacked the same level of recall, which is critical in churn prediction.

Feature importance analysis also highlighted key churn indicators such as tenure, contract type, and monthly charges. These insights can directly help telecom providers take action to retain high-risk customers by adjusting pricing strategies, offering long-term contracts, or targeting personalized promotions.

Overall, this project showed how machine learning — especially deep learning — can provide strong predictive power for real-world business problems like customer churn. It also gave me hands-on experience with the full data science pipeline, from data cleaning to model evaluation and interpretation.

References

[1] A. Idris, A. Khan, and Y. S. Lee, "Intelligent churn prediction in telecom: employing mRMR feature selection and rotBoost based ensemble classification," Applied Intelligence, vol. 39, no. 3, pp. 659–672, 2013.

[2] A. Keramati, R. Jafari-Marandi, R. Aliannejadi, H. Ahmadi, A. Mozaffari, and M. Abbasi, "Improved churn prediction in telecommunication industry using data mining techniques," Applied Soft Computing, vol. 24, pp. 994–1012, 2014.

[3] S. Burez and D. Van den Poel, "Handling class imbalance in customer churn prediction," Expert Systems with Applications, vol. 36, no. 3, pp. 4626–4636, 2009.

[4] Kaggle, "Telco Customer Churn," [Online]. Available: https://www.kaggle.com/datasets/blastchar/telco-customer-churn. [Accessed: April 2025].

[5] F. Chollet et al., Keras Documentation. [Online]. Available: https://keras.io

[6] Scikit-learn Developers, Scikit-learn Documentation, [Online]. Available: https://scikit-learn.org/stable/documentation.html

[7] [7] T. Chen and C. Guestrin, "XGBoost: A Scalable Tree Boosting System," in Proc. 22nd ACM SIGKDD Int. Conf. on Knowledge Discovery and Data Mining, San Francisco, CA, 2016, pp. 785–794.