National University of Technology

Computer Science Department

Semester: Spring 2025

Program: Artificial Intelligence

Course: Machine Learning

Course Code: CS284

Telco Customer Churn Prediction: A Machine Learning Approach

Submitted To:

Lec. Hajra Ahmed

Submitted By:

Name	ID
Khurram Riaz	F23607049

Contents

Abstract

This report details the development of a machine learning model to predict customer churn for a telecommunications company. The project involved comprehensive data preprocessing, exploratory data analysis, model selection, training, and evaluation. Various classification algorithms were considered, with a LightGBM (LGBM) classifier ultimately selected as the best performing model. The tuned LGBM model achieved an F1-score of 0.8729 for the churn class on the hold-out test set, demonstrating strong predictive capabilities. Key drivers of churn identified by the model include monthly charges, tenure months, and customer lifetime value (CLTV).

1 Introduction

Customer churn, the phenomenon where customers cease doing business with a company, is a significant concern for telecommunications providers. Retaining existing customers is often more cost-effective than acquiring new ones. Therefore, accurately predicting which customers are likely to churn allows businesses to implement targeted retention strategies proactively. This project aims to build a robust predictive model for customer churn using a publicly available Telco customer dataset.

2 Dataset Description

The dataset used for this project is the "Telco Customer Churn" dataset, sourced from an Excel file. It contains various attributes for 7043 customers, including demographic information, account details, services subscribed to, and churn status.

2.1 Initial Data Overview

The raw dataset consisted of 7043 rows and 33 columns. Key information includes:

- Customer demographics (Gender, Senior Citizen, Partner, Dependents).
- Account information (Tenure Months, Contract, Paperless Billing, Payment Method, Monthly Charges, Total Charges).
- Service subscriptions (Phone Service, Multiple Lines, Internet Service, Online Security, etc.).
- Churn information (Churn Label, Churn Value, Churn Score, Churn Reason, CLTV).

The target variable for prediction is 'Churn Label', indicating whether a customer churned (Yes/1) or not (No/0).

3 Methodology

The project followed a standard machine learning pipeline, encompassing data preprocessing, feature engineering, model training, hyperparameter tuning, and evaluation.

3.1 Data Preprocessing

Data preprocessing was a critical step to prepare the data for modeling.

1. **Initial Data Cleaning:**
    - Columns deemed irrelevant for prediction, such as 'Count', 'Country', 'State', 'CustomerID', 'Lat Long', 'Latitude', and 'Longitude', were dropped.
    - The 'Churn Reason' column was dropped as it represents information available only after churn occurs, thus being a source of data leakage for predictive modeling.
    - The 'Churn Value' column, being a numerical representation of 'Churn Label', was also dropped to avoid redundancy with the chosen target variable.

2. **Handling Target Variable:** The 'Churn Label' column (Yes/No) was mapped to a numerical format (1/0).

3. **Data Type Conversion:** The 'Total Charges' column, initially an object type, was converted to a numeric type. Errors during conversion (e.g., for empty strings) were coerced to NaN.

4. **Handling High Cardinality Features:** 'City' and 'Zip Code' columns were dropped due to their high cardinality, which can complicate modeling without specialized encoding techniques.

5. **Train-Test Split:** The dataset was split into training (80%) and testing (20%) sets. Crucially, this split was performed *before* imputation, scaling, and encoding to prevent data leakage from the test set into the training process. Stratification was used based on the target variable ('Churn Label') to ensure similar class proportions in both sets.

6. **Feature Identification:** Numerical and categorical features were identified from the training set. 'Senior Citizen', initially an object type, was treated as a categorical feature.

7. **Imputation:**

    - Missing values in the 'Total Charges' column (resulting from the earlier numeric conversion) were imputed using the median value calculated from the *training set only*. This median was then applied to fill NaNs in both the training and test sets.
    - No other features had missing values after the initial cleaning steps.

8. **Feature Scaling:** Numerical features were standardized using `StandardScaler` from scikit-learn. The scaler was fit only on the training data and then used to transform both training and test sets.

9. **Categorical Encoding:** Categorical features were one-hot encoded using `OneHotEncoder` from scikit-learn. `drop='first'` was used to avoid multicollinearity, and `handle_unknown='ignore'` was set to manage new categories in the test set by creating all-zero columns for them. The encoder was fit only on the training data.

10. **Final Processed Data:** The scaled numerical features and one-hot encoded categorical features were concatenated to form the final processed feature sets for training (`X_train_processed`) and testing (`X_test_processed`).

3.2 Model Selection and Training

A range of classification algorithms were evaluated to identify the best performer for churn prediction. These included both non-ensemble and ensemble methods.

3.2.1 Non-Ensemble Models

The following non-ensemble models were initially trained and evaluated with default parameters:

- Logistic Regression
- Support Vector Machine (Linear Kernel)

- Support Vector Machine (RBF Kernel)
- K-Nearest Neighbors
- Decision Tree
- Gaussian Naive Bayes

3.2.2 Ensemble Models

Ensemble learning techniques, known for their robust performance, were also explored:

- Random Forest
- AdaBoost
- Gradient Boosting (Scikit-learn's implementation)
- XGBoost
- LightGBM (LGBM)

All models were trained on the `X_train_processed` data and initially evaluated on the `X_test_processed` data.

3.3 Hyperparameter Tuning

Based on initial performance (primarily F1-score for the churn class and ROC AUC), the LightGBM (LGBM) classifier was selected for further optimization. Hyperparameter tuning was performed using `RandomizedSearchCV` with 5-fold stratified cross-validation on the training data. The search aimed to find the combination of parameters that maximized the F1-score for the churn class. The parameter grid explored included:

- `n_estimators`: [100, 200, 300, 500]
- `learning_rate`: [0.01, 0.05, 0.1]
- `num_leaves`: [31, 50, 70]
- `max_depth`: [-1, 10, 20]
- `subsample`: [0.7, 0.8, 0.9, 1.0]
- `colsample_bytree`: [0.7, 0.8, 0.9, 1.0]

A total of 50 iterations were performed in the randomized search.

3.4 Evaluation Metrics

The models were evaluated using a suite of metrics appropriate for classification tasks, especially those with potential class imbalance:

- Accuracy
- Precision (for the churn class)
- Recall (for the churn class, also known as Sensitivity)
- F1-score (for the churn class)
- ROC AUC (Area Under the Receiver Operating Characteristic Curve)
- Confusion Matrix

The F1-score and Recall for the churn class were given particular attention, as correctly identifying churners is a key business objective.

4 Results and Discussion

4.1 Initial Model Performance

The initial evaluation of various models (with default parameters) on the test set provided insights into their baseline performance. Ensemble models, particularly LightGBM and Gradient Boosting, showed superior performance compared to non-ensemble models, especially in terms of F1-score and ROC AUC. Among non-ensemble models, Support Vector Machine (RBF kernel) and Logistic Regression performed commendably.

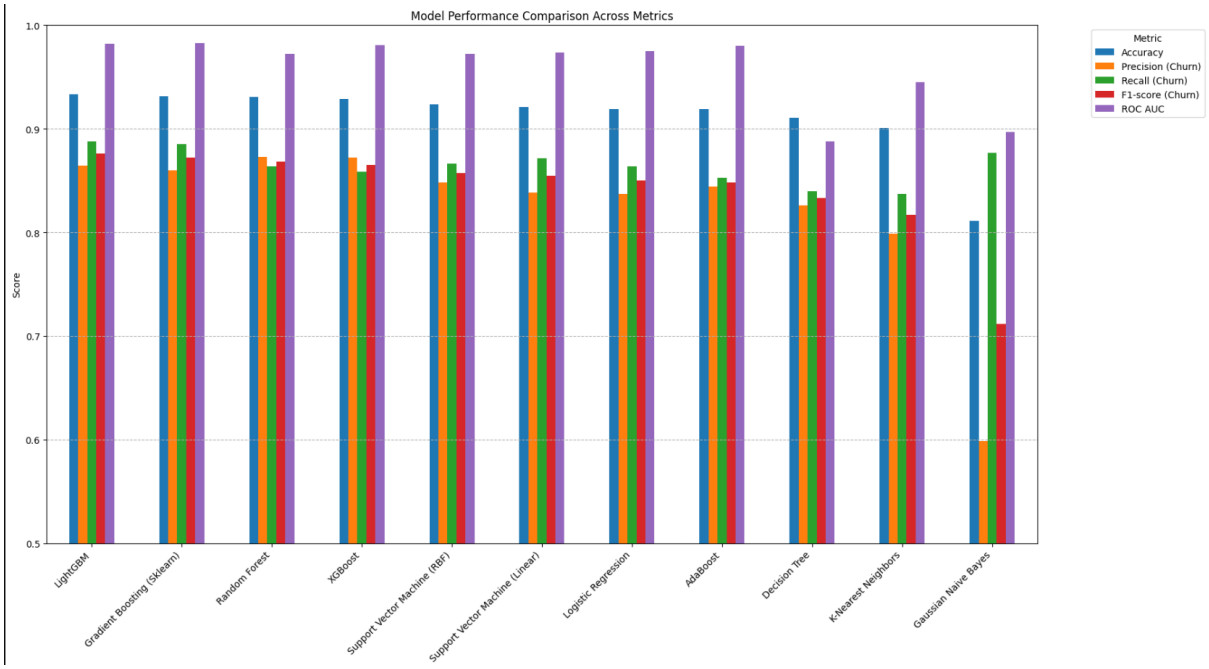

Figure 1: Initial Performance Comparison of Different Models

4.2 Cross-Validation and Hyperparameter Tuning Results

Cross-validation on the training set for the LightGBM model (before extensive tuning) yielded stable performance estimates, indicating that the model was not significantly overfitting. After hyperparameter tuning using `RandomizedSearchCV`, the best parameters found for the LGBM classifier were:

```
{'subsample': 0.7, 'num_leaves': 50, 'n_estimators': 300,
 'max_depth': 10, 'learning_rate': 0.01, 'colsample_bytree': 0.9}
```

The best F1-score achieved during the cross-validated randomized search was noted.

4.3 Final Evaluation of Tuned LightGBM Model

The tuned LightGBM model was then evaluated on the held-out test set. The performance metrics were as follows:

- **Accuracy:** 0.9312
- **ROC AUC:** 0.9814

The classification report provided a more detailed breakdown:

Table 1: Classification Report for Tuned LightGBM Model on Test Set

Class	Precision	Recall	F1-score	Support
0 (No Churn)	0.96	0.95	0.95	1035
1 (Churn)	0.86	0.89	0.87	374
Accuracy			0.93	1409
Macro Avg	0.91	0.92	0.91	1409
Weighted Avg	0.93	0.93	0.93	1409

The F1-score for the churn class (1) was **0.8729**. This indicates a good balance between precision and recall for identifying customers likely to churn.

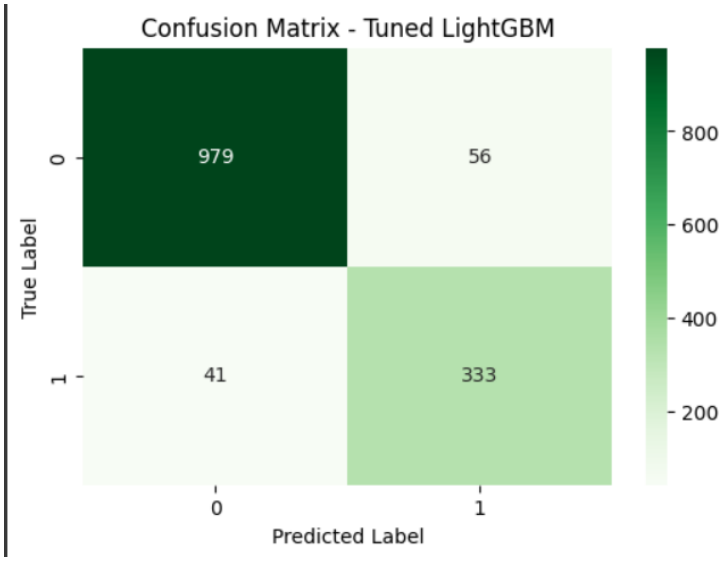

Figure 2: Confusion Matrix for the Tuned LightGBM Model on Test Set

4.4 Feature Importance

The tuned LightGBM model provided feature importances, highlighting the factors most influential in its predictions. The top key features influencing churn were identified as:

1. Monthly Charges
2. Tenure Months
3. CLTV (Customer Lifetime Value)

Other significant features likely included contract type, internet service type, and various add-on service subscriptions.

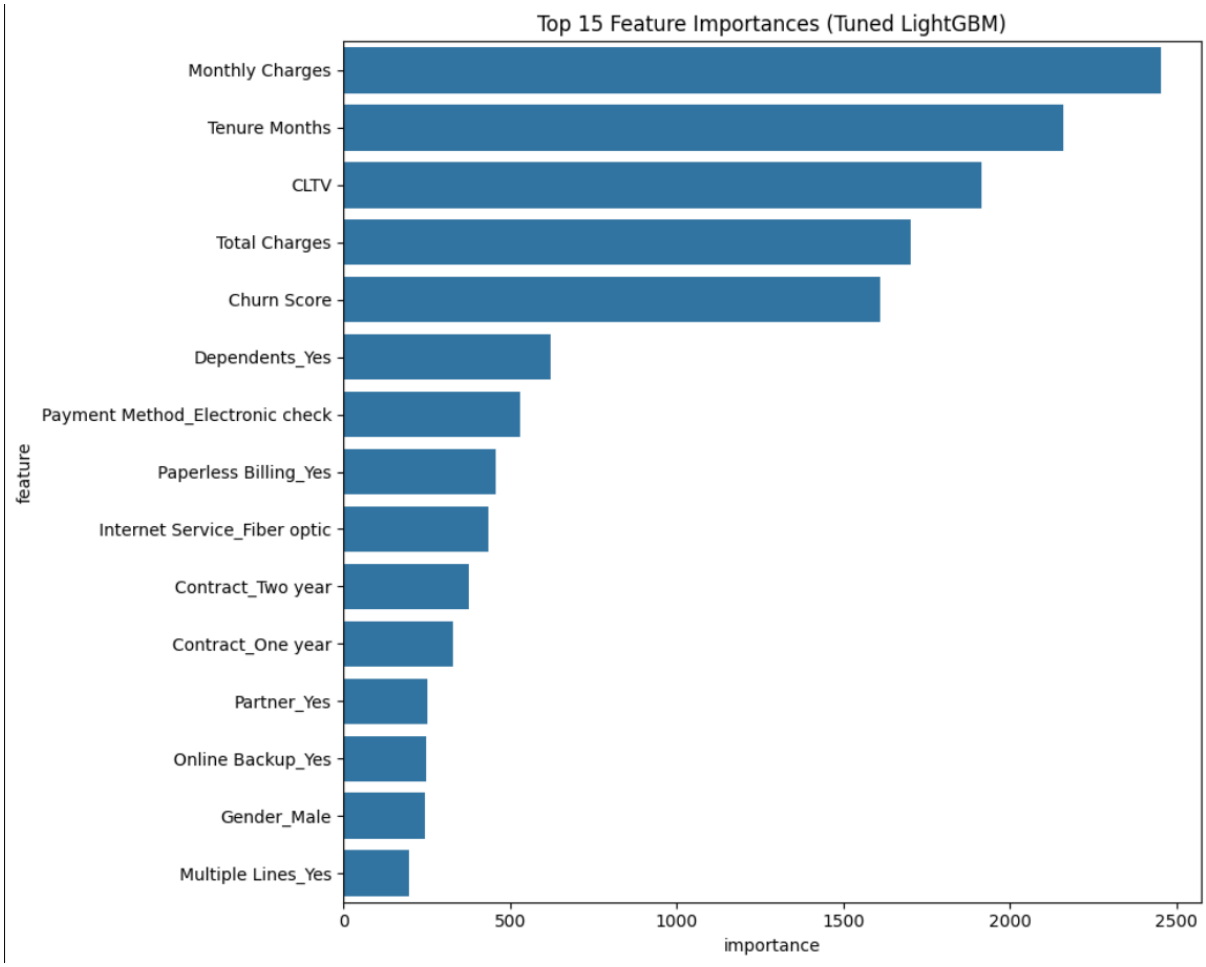

Figure 3: Top Feature Importances from the Tuned LightGBM Model

The identification of these features aligns with business intuition. Higher monthly charges might lead to churn if perceived value is low, while longer tenure and higher CLTV are typically associated with loyalty.

5 Model Deployment and Prediction Function

For practical application, such as integration into a user interface for on-demand predictions, the finalized model and its associated preprocessing components (scaler, encoder, imputation values, feature lists) were saved using `joblib`. A prediction pipeline was established, consisting of:

1. A `preprocess_new_data` function that takes raw new customer data (as a dictionary or DataFrame) and applies the exact same transformations (imputation, scaling, encoding, column ordering) as used for the training data, utilizing the saved preprocessing objects.
2. A `predict_churn` function that takes raw customer data, passes it through the preprocessing function, and then uses the loaded trained model to output a churn prediction (e.g., "Churn" or "No Churn") and the associated probability of churn.

This setup ensures consistency in preprocessing and allows the model to be used for predicting churn for new, unseen customer instances. Example predictions demonstrated the model's ability to differentiate between profiles likely to churn and those likely to remain, and to handle missing input features gracefully through predefined imputation strategies.

6 Conclusion and Future Work

The project successfully developed a LightGBM classifier capable of predicting Telco customer churn with an F1-score of 0.8729 for the churn class on the test set. The model identified 'Monthly Charges', 'Tenure Months', and 'CLTV' as primary drivers of churn. The preprocessing pipeline was carefully constructed to prevent data leakage, ensuring robust and reliable model evaluation.

Future work could involve:

- **Extensive Hyperparameter Tuning:** Applying more exhaustive tuning techniques (e.g., `GridSearchCV` or Bayesian optimization) to other top-performing models like Gradient Boosting or XGBoost.
- **Advanced Feature Engineering:** Exploring the creation of new features, such as interaction terms or aggregations (e.g., from 'City' or 'Zip Code' if more sophisticated encoding methods are used).
- **Threshold Tuning:** Adjusting the prediction threshold of the classifier to optimize for specific business objectives (e.g., maximizing recall subject to a minimum precision).
- **Deployment to UI:** Integrating the saved model and prediction function into a user-friendly interface (e.g., using Streamlit or Flask) for real-time churn predictions based on new customer data input.

Overall, the developed model provides a strong foundation for a customer churn prediction system.

THE END