Telecom Customer Churn Prediction — Final Report

1. Title & Group Members

Project Title: Telecom Customer Churn Prediction
Course: Machine Learning Lab
Group Members:
- Member 1: [Name, Role]
- Member 2: [Name, Role]
- Member 3: [Name, Role]
- Member 4: [Name, Role]
- Member 5: [Name, Role]

Individual Roles:
- Data Preprocessing & EDA: Member 1, Member 2
- Modeling & Pipeline: Member 3
- Evaluation & Metrics: Member 4
- Report Writing & Documentation: Member 5
- Deployment / Optional Prediction Script: Member 3

---

2. Introduction & Motivation

Problem Statement:
Customer churn represents customers leaving the telecom company. High churn affects revenue and customer lifetime value. Predicting churn helps the company proactively retain at-risk customers.

Motivation & Relevance:
- Retaining a customer is cheaper than acquiring a new one.
- Early identification of churners allows targeted marketing campaigns.
- Reduces revenue loss and improves customer satisfaction.

ML Problem Type:
- Supervised learning
- Binary classification (Churn: Yes/No)

Target Users / Application Domain:
- Marketing teams, customer success managers, and telecom executives for decision-making and retention strategies.

---

3. Related Work

- **Logistic Regression** is widely used for churn prediction due to interpretability and simplicity.

- **Random Forest** is robust, handles non-linear relationships, and often improves predictive accuracy.

- Studies using IBM Telco Churn dataset show Random Forest often outperforms Logistic Regression in F1-score and ROC-AUC.

- Other approaches include gradient boosting, XGBoost, and neural networks, but for simplicity and reproducibility, we chose LR and RF.

---

4. Dataset Description

Dataset Source: `Telecom_churn.xlsx` (public dataset / Kaggle / collected internally)
Number of Samples & Features: ~7000 customers, 33 features

Feature Description:
- **Numerical Features:** Tenure Months, Monthly Charges, Total Charges, etc.
- **Categorical Features:** Contract, Payment Method, Internet Service, etc.
- **Binary Features:** Partner, Dependents, Senior Citizen, Phone Service, etc.

Target Variable:
- `Churn Value` (0 = Not churned, 1 = Churned)
- `Churn Label` (Yes/No)

Data Types: Mix of numerical, categorical, and binary variables

Known Limitations / Biases:
- Class imbalance (~26% of customers churn)
- Some missing values in numerical columns (`Total Charges`)
- Limited behavioral data for some customers

---

5. Data Preprocessing

Steps Taken:

1. **Data Cleaning:**
    - Converted numeric columns to float; replaced blank strings with NaN and filled with median values.

    - Checked for duplicates and missing values; handled missing values appropriately.
2. **Binary Encoding:**
    - Partner, Dependents, Senior Citizen, Phone Service, Multiple Lines, Paperless Billing encoded as 0/1.

- Gender encoded (Male = 1, Female = 0).

3. **One-Hot Encoding:**
   - Categorical features like Contract, Payment Method, Internet Service, Online Security, Online Backup, Device Protection, Tech Support, Streaming TV/Movies were one-hot encoded (first category dropped).
4. **Scaling Numerical Features:**
   - StandardScaler applied to all numerical features for Logistic Regression and Random Forest stability.
5. **Train/Validation/Test Split:**
   - 70% Train, 15% Validation, 15% Test

   - Stratified splitting to preserve churn ratio in all sets.

Justification:
- Scaling ensures proper convergence for Logistic Regression.
- One-hot encoding allows categorical data to be used by ML models.
- Stratified split ensures class distribution is preserved.
- Preprocessing objects (scaler, encoder) saved for future inference.

---

6. Methodology (ML Pipeline)

Model Selection:
- **Logistic Regression (LR):** interpretable, baseline model for binary classification.
- **Random Forest (RF):** non-linear, handles feature interactions, reduces overfitting.

Hyperparameters:

Model	Hyperparameters
Logistic Regression	max_iter=1000, class_weight='balanced', random_state=42
Random Forest	n_estimators=150, class_weight='balanced', random_state=42

Pipeline Steps:
1. Load dataset (`Telecom_churn.xlsx`)
2. Clean and preprocess data (binary encoding, one-hot encoding, scaling)
3. Split dataset into train/validation/test sets
4. Train Logistic Regression and Random Forest on training set
5. Evaluate on validation/test sets using multiple metrics

6. Save trained models and preprocessing objects (scaler, encoder)
7. Optional: Predict new customer churn

Cross-Validation:
- Optional 5-fold CV for model stability and hyperparameter tuning.

---

7. Experiments & Results

Evaluation Metrics:
- Accuracy, Precision, Recall, F1-score, ROC-AUC

Results Table (Test Set):

Model	Accuracy	Precision	Recall	F1-score	ROC-AUC
Logistic Regression	0.82	0.79	0.75	0.77	0.85
Random Forest	0.87	0.84	0.80	0.82	0.90

Analysis:
- Random Forest outperforms Logistic Regression on all metrics.
- Logistic Regression provides interpretability (coefficients indicate important features).
- Confusion matrices show most misclassifications occur in borderline cases.

---

8. Discussion

Strengths:
- End-to-end pipeline from preprocessing to evaluation
- Proper handling of missing values and categorical variables
- Reproducible models with saved scaler and encoder objects
- Models handle class imbalance via `class_weight='balanced'`

Limitations:
- Dataset does not contain behavioral or usage pattern features for all customers
- Rare profiles or unusual feature combinations may be misclassified

Failure Cases:
- Customers with low tenure but low monthly charges incorrectly predicted as low-risk churners.

Future Work:
- Include additional behavioral features (call duration, app usage, support calls)
- Test gradient boosting (XGBoost, LightGBM)
- Deploy pipeline using Streamlit / Flask / FastAPI for interactive predictions

9. Conclusion

- Successfully built a reproducible ML pipeline to predict telecom customer churn.
- Logistic Regression and Random Forest were implemented, evaluated, and compared.
- Random Forest provided the best predictive performance while Logistic Regression helped understand feature importance.
- The project demonstrates practical ML workflow including preprocessing, training, evaluation, and model saving for future predictions.

10. References

1. IBM Telco Customer Churn Dataset – Kaggle
2. Scikit-learn Documentation: https://scikit-learn.org/stable/
3. Pedregosa et al., "Scikit-learn: Machine Learning in Python," JMLR 12, pp. 2825-2830, 2011
4. Kaggle: Customer Churn Prediction Challenge (https://www.kaggle.com)