Customer Churn Prediction Project Documentation

1. Project Overview

Customer churn refers to customers who stop using a company's products or services. Predicting churn helps businesses take proactive actions to retain customers. This project builds an end-to-end machine learning pipeline to **predict customer churn** using historical customer data.

The project covers: - Data preprocessing - Feature engineering - Outlier handling - Encoding and scaling - Model building and evaluation

---

2. Objective

The primary objective of this project is to: - Predict whether a customer is likely to churn (`Yes` or `No`) - Use machine learning to identify churn patterns - Provide a scalable and production-ready preprocessing pipeline

---

3. Dataset Description

The dataset (`customer_churn.csv`) contains customer-level information such as:

Feature	Description
customerID	Unique customer identifier
gender	Gender of the customer
SeniorCitizen	Whether customer is a senior citizen
tenure	Number of months customer stayed
MonthlyCharges	Monthly billing amount
TotalCharges	Total amount billed
Contract	Contract type
PaymentMethod	Payment method
Churn	Target variable (Yes / No)

---

4. Technology Stack

- **Programming Language:** Python 3.14
- **Libraries Used:**

- pandas
- numpy
- scikit-learn

---

5. Data Preprocessing Steps

5.1 Data Loading

The dataset is loaded using pandas:

```
pd.read_csv('customer_churn.csv')
```

5.2 Data Cleaning

- Dropped `customerID` (non-informative feature)
- Converted `Churn` to binary format:
- Yes → 1
- No → 0
- Removed rows with missing target values

5.3 Handling Missing Values

- Numerical columns: Filled with **median**
- Categorical columns: Filled with **mode**
- Fully empty categorical columns filled with `'Unknown'`

This ensures robustness and prevents runtime errors.

---

6. Outlier Handling

Outliers in `MonthlyCharges` were handled using the **Interquartile Range (IQR) method**.

Instead of removing rows, values were **capped**: - Prevents data loss - Maintains dataset size - Improves model stability

Formula used:

```
Lower Bound = Q1 - 1.5 × IQR
Upper Bound = Q3 + 1.5 × IQR
```

---

7. Feature Engineering

7.1 Feature and Target Split

- Features (`X`): All columns except `Churn`
- Target (`y`): `Churn`

7.2 Feature Categorization

- **Numerical Features:** Scaled using `StandardScaler`
- **Categorical Features:** Encoded using `OneHotEncoder`

---

8. Machine Learning Pipeline

A **Pipeline** was used to ensure: - No data leakage - Clean and reproducible workflow

Pipeline components: 1. ColumnTransformer 2. Feature scaling 3. One-hot encoding 4. Logistic Regression classifier

Class imbalance was handled using:

```
class_weight='balanced'
```

---

9. Model Selection

Logistic Regression

Chosen because: - Interpretable - Fast and efficient - Suitable for binary classification

Hyperparameters: - `max_iter = 1000` - `class_weight = balanced`

---

10. Model Training

- Dataset split: 80% training, 20% testing
- Stratified split to maintain churn ratio

---

11. Model Evaluation

The model was evaluated using:

11.1 Confusion Matrix

Shows true positives, false positives, true negatives, and false negatives.

11.2 Classification Report

Metrics used: - Precision - Recall - F1-score - Accuracy

These metrics help assess model performance and churn detection quality.

---

12. Results

- Model successfully predicts customer churn
- Handles real-world data issues safely
- Suitable as a baseline churn prediction system

---

13. Limitations

- Logistic Regression assumes linear relationships
- Performance depends on feature quality
- Dataset-specific tuning may be required

---

14. Future Enhancements

- Use advanced models (Random Forest, XGBoost)
- Add SMOTE for imbalance handling
- Feature importance & SHAP explainability
- Model persistence using joblib
- Deployment using Streamlit or Flask

---

15. Conclusion

This project demonstrates a complete, production-ready machine learning workflow for customer churn prediction. The preprocessing pipeline is robust, scalable, and suitable for real-world datasets, making it an excellent foundation for further enhancements and deployment.

End of Documentation