Customer Churn Prediction

Machine Learning Project Report

Author: Mohd Aafi
Date: January 15, 2026
Project Type: Binary Classification
Tools Used: Python, Scikit-learn, TensorFlow/Keras, XGBoost, Streamlit

---

Table of Contents

---

1. Executive Summary

This project addresses the critical business challenge of **customer churn prediction** in a subscription-based service. Using machine learning techniques, we developed predictive models to identify customers at risk of discontinuing their service.

Key Findings:

Metric	Value
Dataset Size	10,000 customers
Churn Rate	49.2% (balanced dataset)
Best Model	XGBoost Classifier
Test Accuracy	~50.1%
Test Recall	~52.7%
Test F1-Score	~51.6%

Key Insights:

- The dataset exhibits **no significant correlation** between features and churn behavior
- **Class distribution is balanced** (~50/50 split), eliminating need for resampling techniques
- **No multicollinearity** detected (all VIF values < 5)
- Feature importance analysis reveals **Monthly Bill** and **Total Usage** as top predictors
- Model performance suggests the given features have **limited predictive power** for churn

---

2. Problem Statement

Business Context

In today's competitive business landscape, customer retention is paramount for sustainable growth. Customer churn—the phenomenon where customers discontinue their use of a service—leads to:

- **Revenue loss** from departing customers
- **Increased acquisition costs** to replace churned customers
- **Decline in market share** and competitive position

Objective

Develop a machine learning model that can:

1. **Accurately predict** which customers are likely to churn
2. **Identify key factors** contributing to churn behavior
3. **Enable proactive retention strategies** through early intervention

Success Criteria

- Build models with high recall to minimize false negatives (missed churners)
- Provide interpretable insights for business decision-making
- Create an interactive dashboard for model deployment and monitoring

---

3. Dataset Description

Overview

Attribute	Value
Total Records	10,000
Total Features	9
Target Variable	Churn (Binary: 0/1)
Missing Values	0 (0%)
Duplicate Records	0

Feature Dictionary

Feature	Type	Description	Range/Values
CustomerID	Identifier	Unique customer identifier	1 - 100,000
Name	Text	Customer name	Text strings
Age	Numerical	Customer age in years	18 - 70
Gender	Categorical	Customer gender	Male, Female
Location	Categorical	Customer location	Houston, Los Angeles, Miami, Chicago, New York
Subscription_Length_Months	Numerical	Duration of subscription	1 - 24 months
Monthly_Bill	Numerical	Monthly billing amount	$30 - $100
Total_Usage_GB	Numerical	Total data usage	50 - 500 GB
Churn	Binary	Target variable	0 (Retained), 1 (Churned)

Statistical Summary

Feature	Mean	Std Dev	Min	Median	Max
Age	44.13	15.27	18.00	44.00	70.00
Subscription_Length_Months	12.46	6.96	1.00	12.00	24.00

Monthly_Bill	64.81	20.23	30.02	64.67	99.99
Feature	**Mean**	**Std Dev**	**Min**	**Median**	**Max**
Total_Usage_GB	276.26	130.18	50.00	276.00	500.00

---

4. Exploratory Data Analysis

4.1 Target Variable Distribution

The churn distribution analysis reveals:

Category	Count	Percentage
Retained (0)	~5,080	50.8%
Churned (1)	~4,920	49.2%

Insight: The dataset is nearly perfectly balanced, eliminating the need for oversampling (SMOTE) or undersampling techniques.

4.2 Categorical Feature Analysis

Gender Distribution

Gender	Count	Percentage
Female	50,216	50.2%
Male	49,784	49.8%

Location Distribution

Location	Count
Houston	20,157
Los Angeles	~20,000
Miami	~20,000
Chicago	~20,000
New York	~20,000

Insight: Both gender and location are evenly distributed across the dataset.

4.3 Numerical Feature Distributions

All numerical features exhibit approximately **normal distributions**:

- **Age:** Uniform distribution between 18-70 years
- **Subscription Length:** Uniform distribution between 1-24 months
- **Monthly Bill:** Uniform distribution between $30-$100
- **Total Usage:** Uniform distribution between 50-500 GB

Skewness Analysis:

Feature	Skewness	Interpretation
Age	~0.00	Symmetric
Subscription_Length_Months	~0.00	Symmetric
Monthly_Bill	~0.00	Symmetric
Total_Usage_GB	~0.00	Symmetric

4.4 Correlation Analysis

The correlation matrix reveals **no significant correlations** between features:

Feature Pair	Correlation
Age - Subscription Length	0.0034
Age - Monthly Bill	0.0011
Age - Total Usage	0.0019
Age - Churn	0.0016
Subscription Length - Monthly Bill	-0.0053
Subscription Length - Churn	0.0023
Monthly Bill - Total Usage	0.0032
Monthly Bill - Churn	-0.0002
Total Usage - Churn	-0.0028

Critical Insight: The near-zero correlations between all features and the target variable (Churn) indicate that **the provided features have very weak predictive power** for customer churn. This is a significant finding that explains the model performance limitations.

---

5. Data Preprocessing

5.1 Data Cleaning Steps

Step	Action	Result
1	Check for missing values	0 missing values found
2	Check for duplicates	0 duplicates found
3	Remove identifier columns	Dropped CustomerID, Name
4	Verify data types	All correct

5.2 Feature Encoding

One-Hot Encoding applied to categorical variables:

- Gender → Gender_Male (drop_first=True)
- Location → Location_Houston, Location_Los Angeles, Location_Miami, Location_New York (drop_first=True, Chicago as reference)

Post-encoding features: 9 features

5.3 Train-Test Split

Split	Size	Percentage
Training Set	7,000	70%
Test Set	3,000	30%

Random State: 42 (for reproducibility)

5.4 Feature Scaling

MinMaxScaler applied to numerical features:

Feature	Original Range	Scaled Range
Age	18 - 70	0 - 1
Subscription_Length_Months	1 - 24	0 - 1
Monthly_Bill	30 - 100	0 - 1
Total_Usage_GB	50 - 500	0 - 1

Rationale: MinMaxScaler chosen because majority of features (after encoding) are binary (0/1).

---

6. Feature Engineering & Selection

6.1 Feature Importance Analysis

Using **Random Forest Feature Importance**:

Rank	Feature	Importance Score
1	Location_Miami	0.129
2	Location_Houston	0.119
3	Location_New York	0.116
4	Monthly_Bill	0.111
5	Total_Usage_GB	0.109
6	Subscription_Length_Months	0.108
7	Gender_Male	0.104
8	Age	0.103
9	Location_Los Angeles	0.100

Observation: Feature importances are relatively uniform, indicating no single feature dominates prediction.

6.2 Cumulative Importance Analysis

Number of Features	Cumulative Importance
4 (top features)	~45%
9 (all features)	100%

6.3 Multicollinearity Check (VIF)

Feature	VIF Score	Status
All features	< 5	✅ No multicollinearity

6.4 Feature Selection Experiments

Two approaches tested:

1. **All 9 features** - Original feature set
2. **Top 4 features** - Monthly_Bill, Total_Usage_GB, Age, Subscription_Length_Months

Finding: Using reduced features did not significantly improve model performance.

7. Model Development

7.1 Machine Learning Algorithms Evaluated

Algorithm	Type	Key Characteristics
Logistic Regression	Linear	Baseline, interpretable
Decision Tree	Tree-based	Non-linear, prone to overfitting
K-Nearest Neighbors	Instance-based	Distance-based classification
Naive Bayes	Probabilistic	Assumes feature independence
Support Vector Machine	Kernel-based	Effective for high-dimensional data
Random Forest	Ensemble	Bagging, reduces variance
AdaBoost	Ensemble	Boosting with weak learners
Gradient Boosting	Ensemble	Sequential boosting
XGBoost	Ensemble	Optimized gradient boosting

7.2 Neural Network Architectures

Five different architectures tested with:

- **Activation Functions:** ReLU, Sigmoid
- **Regularization:** Dropout (10-50%)
- **Optimization:** Adam optimizer
- **Loss Function:** Binary Cross-entropy
- **Callbacks:** EarlyStopping, ModelCheckpoint

Architecture Summary:

Architecture	Layers	Parameters	Regularization
I	128→64→1	Basic	None
II	32→32→16→8→1	Moderate	Dropout (25%, 50%)
III	16→8→4→1	Light	Dropout (25%, 50%)
IV	10→10→5→1	Light + BN	BatchNorm + Dropout
V	64→32→1	Moderate	Dropout (50%)

7.3 Ensemble Methods

Ensemble	Base Estimator	n_estimators
AdaBoost + RF	RandomForest(100)	50
Gradient Boosting	Decision Trees	50
XGBoost	Decision Trees	50

7.4 Dimensionality Reduction (PCA)

Components	Variance Explained
8	~98%
9	100%

Result: PCA did not improve model performance.

7.5 Hyperparameter Tuning

GridSearchCV applied to XGBoost:

Parameter	Values Tested
max_depth	3, 4, 5
learning_rate	0.1, 0.01, 0.001
n_estimators	50, 100, 150

Best Parameters: (Marginal improvement)

- Scoring Metric: Recall
- Cross-validation: 5-fold

---

8. Model Evaluation & Results

8.1 Training Performance (All Features)

Algorithm	Accuracy	Precision	Recall	F1-Score
Logistic Regression	0.500	0.500	0.500	0.500
Decision Tree	**1.000**	**1.000**	**1.000**	**1.000**
KNN	0.652	0.652	0.652	0.652
Naive Bayes	0.500	0.500	0.500	0.500
Random Forest	**1.000**	**1.000**	**1.000**	**1.000**
Gradient Boosting	**1.000**	**1.000**	**1.000**	**1.000**
XGBoost	0.526	0.526	0.526	0.526
SVM	0.500	0.500	0.500	0.500

⚠ **Warning:** Perfect training scores indicate **overfitting**.

8.2 Test Performance (All Features)

Algorithm	Accuracy	Precision	Recall	F1-Score
Logistic Regression	0.500	0.500	0.500	0.500
Decision Tree	0.495	0.495	0.495	0.495
KNN	0.495	0.495	0.495	0.495
Naive Bayes	0.500	0.500	0.500	0.500
Random Forest	0.497	0.497	0.497	0.497

Algorithm	Accuracy	Precision	Recall	F1-Score
Gradient Boosting	0.500	0.500	0.500	0.500
XGBoost	**0.501**	**0.504**	**0.527**	**0.516**
SVM	0.500	0.500	0.500	0.500

8.3 Final Model Performance (XGBoost)

Metric	Training	Test
Accuracy	50.26%	50.10%
Precision	50.44%	50.44%
Recall	52.75%	52.75%
F1-Score	51.57%	51.57%

8.4 Neural Network Performance

Metric	Value
Accuracy	50.07%
Precision	50.22%
Recall	98.08%
F1-Score	66.43%

Note: High recall but low precision indicates the model tends to predict most instances as churned.

8.5 Confusion Matrix Analysis (XGBoost @ threshold=0.5)

	Predicted: 0	Predicted: 1
Actual: 0	TN	FP
Actual: 1	FN	TP

8.6 ROC-AUC Analysis

Model	AUC Score
XGBoost	~0.50
Keras NN	~0.50

Interpretation: AUC ≈ 0.50 indicates model performance is no better than random guessing.

8.7 Threshold Optimization

Threshold	Accuracy	Precision	Recall	F1-Score
0.1	0.49	0.49	1.00	0.66
0.2	0.49	0.49	0.99	0.66
0.3	0.49	0.50	0.94	0.65
0.4	0.50	0.50	0.73	0.60
0.5	**0.50**	**0.50**	**0.53**	**0.52**

Threshold	Accuracy	Precision	Recall	F1-Score
0.6	0.50	0.51	0.29	0.37
0.7	0.49	0.51	0.12	0.20
0.8	0.49	0.52	0.04	0.08
0.9	0.49	0.49	0.01	0.01

Optimal Threshold: 0.5 (balanced trade-off)

---

9. Dashboard Implementation

9.1 Technology Stack

Component	Technology
Framework	Streamlit
Visualization	Matplotlib, Seaborn
Data Processing	Pandas, NumPy
ML Models	Scikit-learn, XGBoost, TensorFlow

9.2 Dashboard Features

Tab 1: Overview

- Dataset statistics (rows, columns, churn rate)
- Churn distribution pie chart
- Feature statistics table

Tab 2: Model Performance

- Model selection (XGBoost / Keras NN)
- Classification threshold slider
- Metrics display (Accuracy, Precision, Recall, F1)
- Confusion Matrix visualization
- ROC Curve with AUC
- Feature Importance chart

Tab 3: Visualizations

- Age distribution by churn status
- Monthly bill distribution
- Usage patterns analysis
- Correlation heatmap

Tab 4: Data Explorer

- Interactive data filtering
- CSV download functionality

9.3 Dashboard Screenshots

[Dashboard running at http://localhost:8502]

---

10. Conclusions & Recommendations

10.1 Key Findings

1. **Dataset Quality:** The dataset is clean with no missing values or duplicates.
2. **Class Balance:** Near-perfect 50/50 split eliminates class imbalance concerns.
3. **Feature-Target Relationship:** Extremely weak correlations between features and churn (~0.00) indicate that the provided features have **minimal predictive**

power.

4. **Model Performance:** All models perform at approximately **random chance level (~50%)**, regardless of:
    - Algorithm complexity (simple to ensemble)
    - Feature selection approach
    - Hyperparameter tuning
    - Neural network architecture
5. **Overfitting Observed:** Tree-based models achieve perfect training scores but fail to generalize.

10.2 Root Cause Analysis

The poor model performance is likely due to:

Factor	Evidence
Insufficient feature informativeness	Near-zero correlations with target
Random data generation	Uniform distributions, no patterns
Missing important features	Customer behavior, engagement, complaints

10.3 Business Recommendations

1. **Data Collection Enhancement:**
    - Customer interaction frequency
    - Service quality ratings
    - Customer support tickets
    - Payment history
    - Feature usage patterns
    - Customer lifetime value
2. **Feature Engineering:**
    - Create interaction features
    - Time-based aggregations
    - Customer segmentation variables
3. **External Data Integration:**
    - Market conditions
    - Competitor offerings
    - Economic indicators

10.4 Technical Recommendations

1. **For Production Deployment:**
    - Collect more informative features before deploying
    - Implement A/B testing framework
    - Set up model monitoring
2. **For Future Iterations:**
    - Acquire real customer data
    - Include behavioral features
    - Consider time-series analysis

---

11. Technical Appendix

11.1 Environment Setup

```
# Create virtual environment
python -m venv .venv

# Activate environment
.venv\Scripts\activate  # Windows

# Install dependencies
pip install -r requirements.txt

# Run dashboard
streamlit run app.py
```

11.2 Dependencies (requirements.txt)

```
pandas>=1.3
numpy>=1.21
scikit-learn>=1.0
xgboost>=1.6
joblib
streamlit>=1.10
matplotlib
seaborn
openpyxl
tensorflow>=2.6
```

11.3 Project Structure

```
ML Project/
├── app.py                          # Streamlit dashboard
├── Customer Churn Prediction.ipynb # Main analysis notebook
├── clean_code.py                   # Python script version
├── export_metrics.py               # Metrics export utility
├── ChurnClassifier.h5              # Saved Keras model
├── customer_churn_classifier.pkl   # Saved XGBoost model (if present)
├── model_metrics.json              # Pre-computed metrics
├── feature_importances_*.csv       # Feature importance data
├── requirements.txt                # Python dependencies
├── README.md                       # Project documentation
└── Customer_Churn_Prediction_Report.md  # This report
```

11.4 Model Artifacts

File	Description	Size
ChurnClassifier.h5	Keras Neural Network	~50KB
customer_churn_classifier.pkl	XGBoost Classifier	~100KB
model_metrics.json	Evaluation metrics	~1KB

11.5 Cross-Validation Results

Fold	Accuracy	Recall
1	0.501	0.528
2	0.499	0.525
3	0.502	0.530

Fold	Accuracy	Recall
4	0.498	0.524
5	0.500	0.527
Mean	**0.500**	**0.527**

References

1. Scikit-learn Documentation: https://scikit-learn.org/
2. XGBoost Documentation: https://xgboost.readthedocs.io/
3. TensorFlow/Keras Documentation: https://www.tensorflow.org/
4. Streamlit Documentation: https://docs.streamlit.io/