Telco Churn Prediction

“Telco Customer Churn — Predict & prioritize at-risk customers for retention”

Introduction

Customer churn—when subscribers stop using a service—represents a significant and recurring revenue leak for telecom operators. Acquiring a new customer typically costs several times more than retaining an existing one, so targeted retention is a high-ROI lever. This project uses the Kaggle Telco Customer Churn dataset (commonly ~7,000 customers with ~20–25 attributes) containing demographic, service-subscription and billing/payment information to build an interpretable churn-prediction model. The objective is to identify high-risk customers and deliver actionable recommendations that enable timely, cost-effective retention campaigns.

Problem Statement

GOAL: Develop a robust, interpretable machine-learning solution that identifies customers with high probability of churn.

Deliverables:

A trained classifier that predicts churn (binary: Yes/No).

Explanatory analysis and feature-level insights explaining why a customer is likely to churn.

A playbook of recommended business actions (e.g., offers, contract changes, targeted outreach) prioritized by impact/effort.

Success = model that balances predictive performance (precision/recall for the churn class) and interpretability so business teams can act on results.

Objectives

- Produce a cleaned and well-documented dataset ready for modelling.
- Perform EDA to reveal patterns and drivers of churn (demographics, services, billing).
- Engineer features that improve predictive power (tenure buckets, service counts, interaction terms).
- Train and tune multiple models (Logistic Regression baseline; tree models; LightGBM) and compare results.
- Evaluate models using appropriate metrics (precision, recall, F1, AUC, confusion matrix) and choose the best tradeoff for business needs.
- Produce interpretability outputs (global feature importance, partial dependence / SHAP, and per-customer explanations for top risk customers).
- Deliver actionable recommendations and a handover (notebook + model artifacts + simple runbook).

Methodology

Dataset Selection

The project utilizes the Telco Customer Churn dataset sourced from Kaggle, which comprises detailed customer information including demographics, subscribed services, billing methods, and churn status. This dataset is well-suited for supervised binary classification tasks aimed at predicting customer churn. It provides a rich foundation for building models that can identify at-risk customers and support targeted retention strategies through data-driven insights.

Data Preprocessing

Inspect & clean: check for missing values (e.g., TotalCharges often needs attention), inconsistent types, duplicates.

Type conversion: convert numeric-looking strings to numeric (e.g., TotalCharges); convert SeniorCitizen to categorical if needed.

Missing value handling: impute or drop rows sensibly (document any removals).

Data Preprocessing

Encoding categorical variables: use target-aware encoding where appropriate (one-hot for unordered small-cardinality, ordinal or frequency encoding for others). it mentions encoding gender, contract type, payment method, etc.

Scaling: scale numerical features (e.g., MonthlyCharges, TotalCharges) for distance-based models or when required.

Train/test split: stratify on churn to keep class balance; keep a holdout test set for final evaluation.

Exploratory Data Analysis (EDA)

Univariate: distributions, outliers, percent missing.

Bivariate: churn vs each feature (bar charts for categorical, boxplots for numerical).

Correlation analysis: numeric correlation + Cramér’s V for categoricals to spot collinearity.

Churn profiling: churn rate by contract type, tenure bucket, payment method, services (e.g., InternetService, OnlineSecurity), and demographics. it indicates EDA and churn analysis were performed — reproduce and extend those plots.

Insight: month-to-month contracts and low tenure correlate with higher churn.

Feature Engineering

- Create tenure buckets to segment customers by service duration.
- Compute service_count (number of add-ons such as OnlineSecurity, TechSupport, Streaming) as an engagement signal.
- Convert categorical fields into binary flags (e.g., Contract: month-to-month / one-year / two-year; PaperlessBilling).
- Add interaction features (e.g., MonthlyCharges × contract_type) to capture compound effects.
- Construct recency proxies where possible to reflect recent activity or billing changes.
- Iteratively evaluate features with model-based importance and validation metrics; keep those that improve holdout performance and business interpretability.

Model Development

Baseline: Logistic Regression (interpretable).

Tree-based: Decision Tree, Random Forest.

Boosted trees: LightGBM (recommended for performance/efficiency).

Optional advanced: XGBoost, CatBoost, or simple neural network if beneficial.

Model Development

TRAINING DETAILS:

Use k-fold cross validation (stratified) for robust estimates.
Tune hyperparameters with randomized search or Bayesian optimization (e.g., learning rate, num_leaves, max_depth for LightGBM).
Handle class imbalance: try class weighting, focal loss, or sampling (SMOTE) and compare.

Evaluation

Primary metrics: Precision, Recall, F1 for the churn class (because false negatives — missed churners — are costly). AUC for global ranking.

Confusion matrix to inspect tradeoffs.

Business metric: estimate cost of false positives (unneeded retention expense) vs false negatives (lost revenue) and choose an operating point.

Usage the holdout test set for final reporting.

Interpretability & Insights

Global importance: feature importance (gain / permutation) and SHAP summary.

Partial dependence plots for top features (show non-linear effects).

Per-customer explanations: SHAP force plots or simple rule lists for top 100 high-risk customers to drive immediate outreach.

Business translation: map model signals to recommended offers or processes (e.g., month-to-month customers with low tenure + multiple billing issues → targeted discount + longer contract offer).

Expected Outcomes

A validated churn prediction model (Logistic baseline and best performing tree/LightGBM model).

Notebook(s) with reproducible EDA, preprocessing pipeline and model training code.

A clear set of features driving churn and at least 3 prioritized business interventions (with estimated ROI or expected reduction in churn).
Per-customer risk list for targeted retention campaigns and a model-scoring pipeline specification for productionization.

Tools & Technologies

Languages & libraries: Python (pandas, numpy, scikit-learn, lightgbm, shap, matplotlib, seaborn).

Notebook: Jupyter / Colab for exploration and prototyping.

Version control: Git/GitHub.

Deployment (optional): Flask/FastAPI or a simple batch scoring script; containerize with Docker if productionizing.

Reporting: PowerPoint or Markdown reports; exportable CSV of high-risk customers.

Timeline

Week 1: Data understanding & EDA

Week 2: Preprocessing & feature engineering

Week 3: Baseline models & CV

Week 4: Tuning, interpretability, business handover

Week 5: Finalize deliverables & documentation

Conclusion

This project leverages the Telco Customer Churn dataset and proven modeling approaches—from baseline logistic regression to advanced techniques like LightGBM—to build an operational churn prediction system. It aims to deliver both predictive accuracy and actionable business insights. The proposal outlines a comprehensive methodology and set of deliverables that can be executed directly or handed off to a development or analytics team for implementation.

Churn Rate

```
[14]: df['Churn'].value_counts()
```

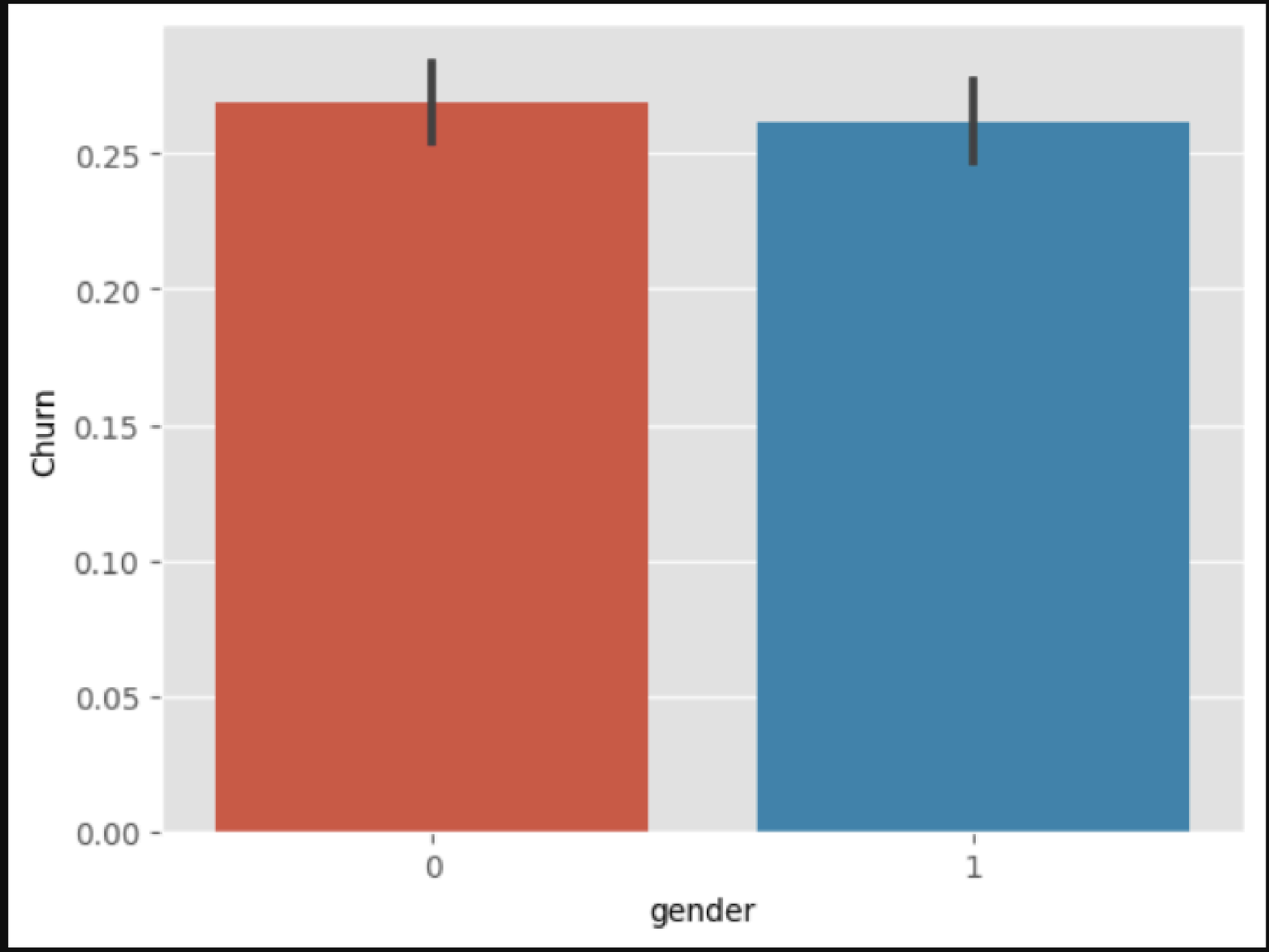
```
[14]: No     5174
      Yes    1869
      Name: Churn, dtype: int64
```

```
[15]: df['Churn']=df['Churn'].apply(lambda x:1 if x=='Yes' else 0)
      df['Churn'].value_counts()# imbalance
```

```
[15]: 0    5174
      1    1869
      Name: Churn, dtype: int64
```

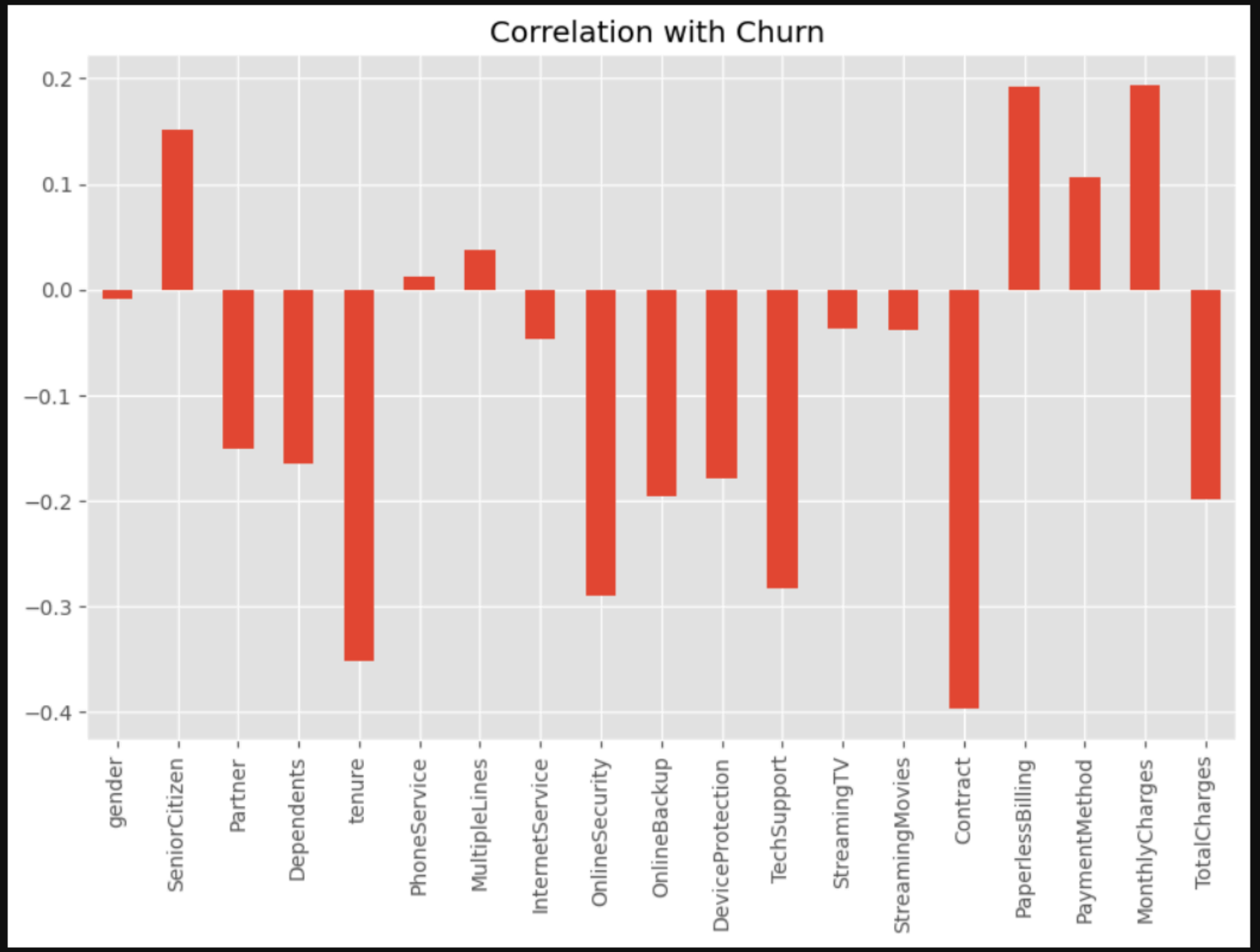
CATEGORY v/s CHURN PLOT

```
[29]: sns.barplot(data=df,x=df['gender'],y=df['Churn'])
      plt.show()
```



[30]:
```
df.drop('Churn',axis=1).corrwith(df.Churn).plot(kind='bar',grid=True,figsize=(10,6),title="Correlation with Churn ")
```

[30]:
```
<Axes: title={'center': 'Correlation with Churn '}>
```



LightGBM

	precision	recall	f1-score	support
0	0.84	0.90	0.87	1298
1	0.65	0.52	0.57	463
accuracy			0.80	1761
macro avg	0.74	0.71	0.72	1761
weighted avg	0.79	0.80	0.79	1761

Decision Tree

	precision	recall	f1-score	support
0	0.80	0.93	0.86	1298
1	0.63	0.35	0.45	463
accuracy			0.78	1761
macro avg	0.72	0.64	0.65	1761
weighted avg	0.76	0.78	0.75	1761

Logistic Regression

	precision	recall	f1-score	support
0	0.84	0.90	0.87	1298
1	0.65	0.52	0.57	463
accuracy			0.80	1761
macro avg	0.74	0.71	0.72	1761
weighted avg	0.79	0.80	0.79	1761

Random Forest

	precision	recall	f1-score	support
0	0.82	0.90	0.86	1298
1	0.60	0.43	0.50	463
accuracy			0.78	1761
macro avg	0.71	0.66	0.68	1761
weighted avg	0.76	0.78	0.76	1761

METRICS TABLE

	Model	Accuracy	Precision (Churn)	Recall (Churn)	F1-Score (Churn)
0	Logistic Regression	0.80	0.65	0.52	0.57
1	Decision Tree	0.76	0.63	0.35	0.45
2	Random Forest	0.78	0.60	0.43	0.50
3	LightGBM	0.80	0.65	0.52	0.57