Telco Customer Churn Report (Lab 10)

Submitted by: Samruddhi Yogesh Kurhe

Project: Telco churn prediction using ensemble methods (Random Forest, XGBoost).

1. Abstract

We built an end-to-end pipeline to predict customer churn for a telecom dataset. The solution uses in-pipeline feature engineering, Random Forest and XGBoost models (with hyperparameter tuning), and ensemble strategies (stacking and soft voting). A small Streamlit app demonstrates model inference where advanced features are computed automatically or optionally overridden by the user.

2. Problem statement & motivation

Customer churn (the event of a customer leaving the service) is a high-impact business problem for telcos: acquiring new customers is typically costlier than retaining existing ones. Predicting churn enables targeted retention actions and improves lifetime value. The task is framed as binary classification (Churn = Yes/No).

3. Dataset

Source: Telco Customer Churn CSV (place in data/telco_churn.csv).
Size & format: ~7k rows, ~20 raw columns (customer demographics, services, billing, tenure, charges).
Target: churn (mapped to binary: 1 = Yes, 0 = No).
Class balance: roughly 26% churners vs 74% non-churners (imbalanced).

4. Preprocessing & feature engineering

Goal: keep the app simple (ask user only for raw fields) while the model can compute or accept advanced fields.

- Minimal raw cleaning (src/data.py): normalize column names, coerce totalcharges to numeric, normalize “No internet service” → No.
- **FeatureEngineer** (src/features.py): scikit-learn transformer that:
    - Adds tenure_group (binned tenure), lifetime_value = monthlycharges * tenure, and avg_charge_per_month = totalcharges / tenure (fallback to monthly).

- **Respects user input**: if the user supplies an engineered field, the transformer does not overwrite it.

- Pipeline design: Pipeline([('feat', FeatureEngineer()), ('pre', ColumnTransformer(...)), ('clf', <estimator>)]).
- Missing values: numeric imputed with median; categorical imputed with most frequent; categorical features one-hot encoded.

This design ensures the Streamlit UI can ask for a few basic raw fields; the pipeline computes advanced features internally when missing.

5. Models & training

- Models trained:
  - RandomForestClassifier (sklearn)
  - XGBoost (XGBClassifier)
- Hyperparameter tuning: RandomizedSearchCV with stratified cross-validation (4 folds) for both models over sensible ranges (n_estimators, max_depth, learning_rate, subsample, etc.).
- Ensembles:
  - **Stacking**: base estimators = RF, XGB; meta-estimator = RandomForest.
  - **Voting (soft)**: weighted average of RF and XGB probabilities.
- Saved artifacts: tuned pipelines and ensembles serialized with joblib into models/; feature_meta.joblib saves raw feature metadata for the app.

6. Evaluation methodology

- Data split: stratified train/test split (80/20).
- CV metrics reported during tuning: ROC-AUC (primary), with accuracy, precision, recall and F1 examined for thresholded decisions.
- Final evaluation on hold-out test set using: Accuracy, Precision, Recall, F1-score, ROC-AUC, and confusion matrix.

7. Results

Model Performance on Test Set

Model	Accuracy	Precision	Recall	F1	ROC-AUC
Random Forest (tuned)	**0.7963**	**0.68**	**0.44**	**0.53**	**0.8479**
XGBoost (tuned)	**0.8020**	**0.67**	**0.51**	**0.58**	**0.8474**
Stacking	**0.7651**	**0.56**	**0.50**	**0.53**	**0.8030**
Voting (soft)	**0.8013**	**0.68**	**0.48**	**0.56**	**0.8484**

8. Interpretability & insights

- Feature importance (from RF/XGB) identifies top predictors such as contract type, tenure, monthlycharges, and totalcharges or lifetime_value.
- SHAP analysis (recommended) reveals how features push predictions toward churn/not-churn for specific customers.
- Business insight examples: Short-tenure, month-to-month contract customers with high monthly charges are at elevated churn risk.

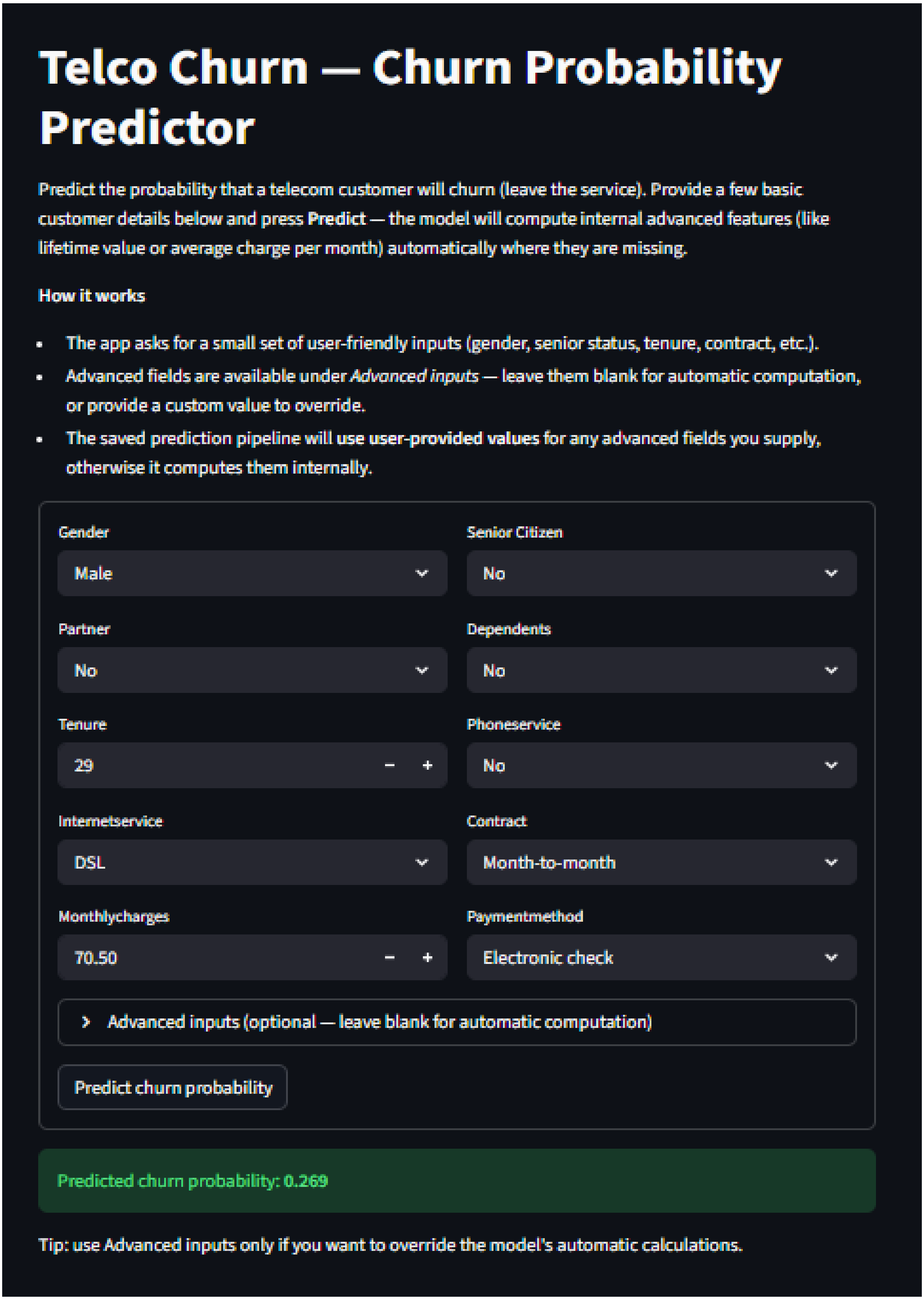
  Customers with longer tenure and multi-year contracts are far less likely to churn.

9. Limitations & ethics

- **Data limitations:** historic customer data; covariate shift may occur if product/pricing/market changes.
- **Imbalanced classes:** model evaluation should prioritize recall/precision trade-offs related to business costs (false positives may waste retention resources; false negatives lose customers).
- **Ethics:** Retention policies must avoid discriminatory practices (e.g., biased offers toward or against protected groups). Interpretability is recommended before automating interventions.

10. Project code repository - https://github.com/samruddhikurhe/telco_churn_prediction.git

11. Demo Screenshots- predicted churn without the input of advanced & engineered features,



Advanced & Engineered features if given input then predicted churn probability is as follows,