Telco Churn Prediction Project - Full Journey Documentation



Goal: Build and deploy an end-to-end Telco Customer Churn prediction tool using Machine Learning and Streamlit.

Main Components:

- Data Cleaning and Feature Engineering
- Model Training and Evaluation
- Streamlit Dashboard with Visualizations
- GitHub and Streamlit Cloud Deployment

Data Cleaning and Preprocessing

- Removed duplicates and handled missing values.
- Converted total charges to numeric and dropped NaNs.
- Encoded binary columns like 'Yes/No' and 'Male/Female'.
- One-hot encoded multi-category columns like Contract, InternetService, etc.

Model Training and Evaluation



- 1. Logistic Regression
- 2. Random Forest
- 3. XGBoost

Hyperparameter Tuning:

We performed tuning using GridSearchCV with pipelines.

```
param_grid = {
    'classifier__n_estimators': [100, 200],
    'classifier__max_depth': [3, 5, 10]
}
```

Best Model Scores and Parameters:

Model: Logistic Regression\ Best Score: 0.8022719204827831\ Best Params: {'classifier_penalty': 'l1', 'classifier_C': np.float64(0.1)}

Model: Random Forest\ Best Score: 0.7983670571529996\ Best Params: {'classifier__n_estimators': 200, 'classifier__min_samples_split': 2, 'classifier__min_samples_leaf': 4, 'classifier__max_depth': 20, 'classifier__bootstrap': False}

Model: XGBoost\ Best Score: 0.8012069577564785\ Best Params: {'classifier_subsample': 0.8, 'classifier_n_estimators': 150, 'classifier_max_depth': 3, 'classifier_learning_rate': 0.1, 'classifier_gamma': 0, 'classifier_colsample_bytree': 1.0}

Limitations of Logistic Regression

While Logistic Regression was a strong baseline, it had some key limitations:

- Low Recall for Churn class (1): It correctly identified only 53% of churned customers, which is risky in churn prediction.
- **F1-score imbalance:** The performance on the minority class (churn) was not as strong, leading to an F1-score of just 58%.
- **Linear Assumptions:** It assumes a linear relationship between features and the log-odds of the outcome, which may not capture complex patterns in the data.
- **Sensitivity to Class Imbalance:** Despite using class_weight='balanced', it still struggled to detect many churners accurately.

⊗Final Model:

We used an **Ensemble Voting Classifier** combining the strengths of:

- Logistic Regression (interpretable but struggled with class imbalance)
- Random Forest (great at handling non-linearity and imbalance)
- XGBoost (strong gradient boosting performance)

The ensemble improved performance and reduced individual model drawbacks.

Model saved with versioning:

joblib.dump(model, f"models/churn_model_{date}.pkl")

Streamlit app(streamlit_app.py)

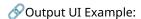
The Streamlit web app provides an interactive interface for business users to explore insights and **predict customer churn** based on input features.

1. User Input Interface

- A **sidebar form** allows users to enter customer data (e.g., contract type, monthly charges, internet service, senior citizen, etc.).
- All feature inputs match the format used during training.

2. Churn Prediction Output

- Once the user clicks **Predict**, the app:
- Runs the prediction on the ensemble model.
- Shows a **Churn Probability** and **Yes/No** prediction.



Prediction: Customer is likely to CHURN. Probability: 74.3%

3. Business Dashboard (Bonus)

In addition to prediction, the app also includes rich visualizations:

- KPI metrics for churned customers
- Churn distribution by contract, tenure, charges, services, demographics
- Actionable segment table to filter risky customers

Sections implemented:

- 1. Churn Overview (KPI Cards + Donut Chart)
- 2. Churn by Contract Type (Stacked Bar)
- 3. Churn by Tenure (Boxplot)
- 4. Churn by Monthly Charges (Scatter)
- 5. Churn by Services (Heatmap)
- 6. Churn by Demographics (Pie Charts)
- 7. Payment Method & Churn (Treemap)
- 8. Actionable Segments (Filtered Table)



ValueError: names='index' not found in DataFrame.

Fix: Changed names='index' to names=churn_counts.index in donut chart.

GitHub Setup

aGitHub was not recognized in CMD:

git : The term 'git' is not recognized

```
Fix: Installed Git and added to system PATH.
Git Init + Push:
 cd telco-churn-app
 # Initialize
 git init
 git add .
 git commit -m "Initial commit"
 # Connect to GitHub
 git remote add origin https://github.com/Amrita-DevX/telco-churn-app
 git branch -M main
 git push -u origin main
🗓 Streamlit Cloud Deployment

    Ø Created new app from GitHub repo 

    PIssue: Error while installing requirements.txt

 Issues Faced:
   1. Wrong entry:
 pip install scikit-learn==1.5.1
Fix: Removed pip install prefix — only keep:
 scikit-learn==1.5.1
   1. pywin32 Error:
 ERROR: Could not find a version that satisfies the requirement pywin32==308
Fix: Removed pywin32 from requirements.txt (Windows-only dependency).
PDeployed URL generated:\ https://smart-churn-checker.streamlit.app/
```

Let me know when you're ready — I can now export this entire doc as .docx or .pdf.

