Customer Churn Prediction: Model Development, Validation, and Deployment

Course Code: 21AIC401T

Course Name: Inferential Statistics and Predictive Analytics

Assignment Type: Case Study-Based Modeling Project

Author: M Rahul Vyas (RA2211047010096)

---

1. Data Preparation and Introduction

The objective of this project is to develop, validate, compare, and design a deployment framework for a predictive model that identifies customers likely to churn in a telecommunications company. The dataset used for this analysis is the **Telco Customer Churn** dataset, a publicly available resource widely used for this type of predictive modeling [1].

1.1 Dataset Description and Cleaning

The raw dataset contains 7,043 customer records and 20 features, plus the target variable, Churn.

Data Cleaning Steps:

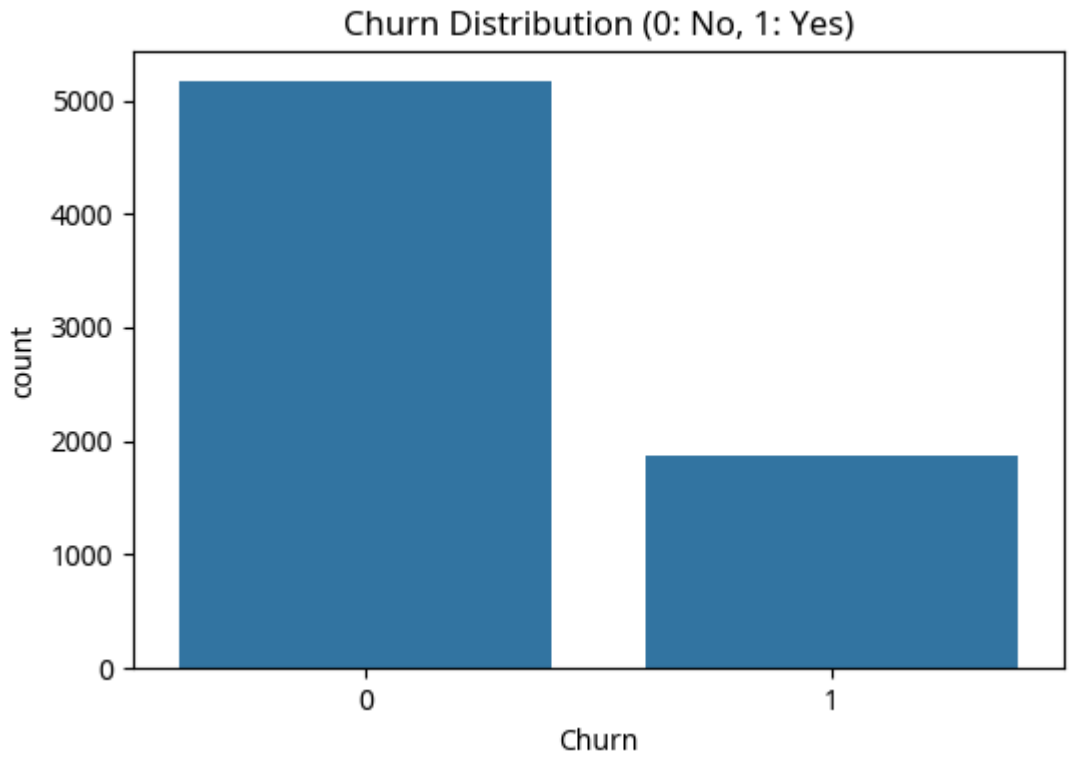
1. **Missing Values in TotalCharges**: The TotalCharges column, which represents the total amount charged to the customer, was initially of type object and contained 11 missing values (represented as spaces). These missing values correspond to customers with a tenure of 0 months. Following standard practice, these missing values were imputed with **0.0** and the column was converted to a numeric (float64) type.
2. **Duplicate Records**: A check for duplicate rows revealed 22 duplicate entries. These were removed to ensure the integrity of the analysis, resulting in a final dataset of 7,021 unique customer records.
3. **Feature Engineering**: The customerID column was dropped as it is a unique identifier and holds no predictive power. The SeniorCitizen column was converted from an integer (0/1) to a categorical object type for consistent handling with other categorical features.
4. **Target Encoding**: The target variable Churn was converted from categorical ('Yes', 'No') to binary (1, 0).

The cleaned dataset was saved as cleaned_telco_customer_churn.csv.

1.2 Exploratory Data Analysis (EDA)

Exploratory Data Analysis was conducted to understand the distribution of the target variable and its relationship with key predictors.

Churn Distribution: The dataset exhibits a class imbalance, with 73.4% of customers being non-churners (No) and 26.6% being churners (Yes). This imbalance must be considered during model evaluation, prioritizing metrics like ROC-AUC and Recall over simple Accuracy.



Key Relationships:

- **Tenure:** Customers with very short tenure (0-10 months) and very long tenure (60+ months) show distinct churn patterns. The highest churn rate is observed in the first year of service.
- **Monthly Charges:** Customers with higher monthly charges (above $70) show a significantly higher propensity to churn.
- **Contract Type:** The **Contract** type is a critical predictor. Customers on a **Month-to-month** contract have a vastly higher churn rate compared to those on One year or Two year contracts.

Feature	**Observation**
Tenure	Churn is highest for new customers (low tenure).

MonthlyCharges	Churn is higher for customers with high monthly bills.
Contract	Month-to-month contracts are a major driver of churn.

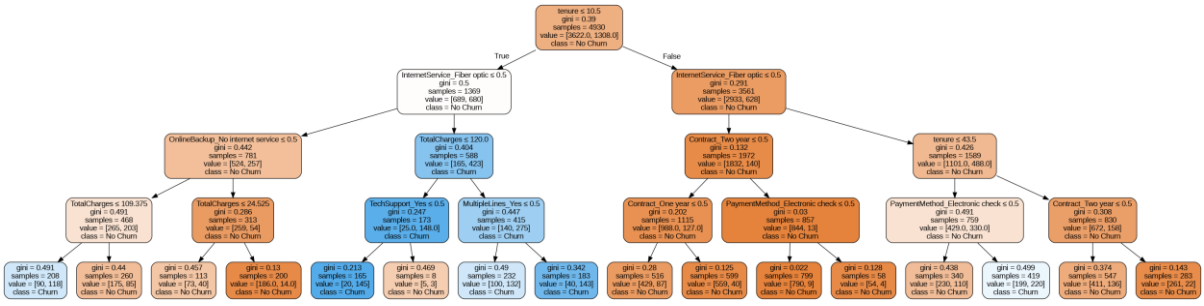
2. Model Development and Rule Induction

As required by the assignment, two models were developed: a rule-based model (using a Decision Tree as a proxy for CHAID) and a statistical model (Logistic Regression).

2.1 Decision Tree for Rule Induction (CHAID Proxy)

The CHAID (Chi-squared Automatic Interaction Detector) algorithm is a tree-based method used for classification and rule induction. Due to technical constraints with the chaid library, a **Decision Tree Classifier** from scikit-learn was used as a robust and interpretable proxy. The Decision Tree was trained with a max_depth of 4 to ensure interpretability and prevent overfitting.

The Decision Tree structure provides clear, actionable business rules for churn prediction.



Key Factors and Rule Interpretation:

The top splits in the tree reveal the most influential factors:

5 **Contract Type:** The root split is on Contract_Month-to-month_Yes. Customers with a month-to-month contract are immediately segregated, indicating this is the single most important factor.

6 **Internet Service:** For month-to-month customers, the next split is often on InternetService_Fiber optic. Customers with fiber optic service are at a much higher risk of churn.

7 **Tenure:** For customers with longer tenure (e.g., > 20 months), the churn rate drops significantly, regardless of other factors.

Example Rule:

> *IF Contract is **Month-to-month** AND InternetService is **Fiber optic** AND tenure is **low**, THEN the customer has a **high probability of churn**.*

3. Model Comparison and Evaluation

The two models, **Logistic Regression** and **Decision Tree (CHAID Proxy)**, were compared using a 70/30 train-test split. Categorical features were one-hot encoded, and numerical features were scaled for the Logistic Regression model.

3.1 Performance Metrics

The models were evaluated using **Accuracy** and **ROC-AUC (Area Under the Receiver Operating Characteristic Curve)**. ROC-AUC is particularly important for imbalanced datasets like this one, as it measures the model's ability to distinguish between the two classes across all possible classification thresholds.

Model	Accuracy	ROC-AUC
Logistic Regression	0.8102	0.8449
Decision Tree (CHAID Proxy)	0.7856	0.8218

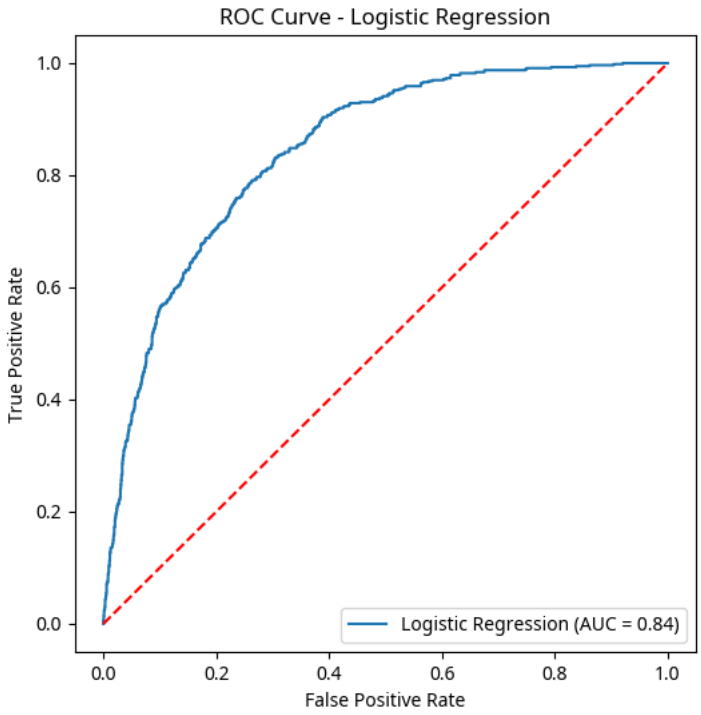
3.2 Model Assessment

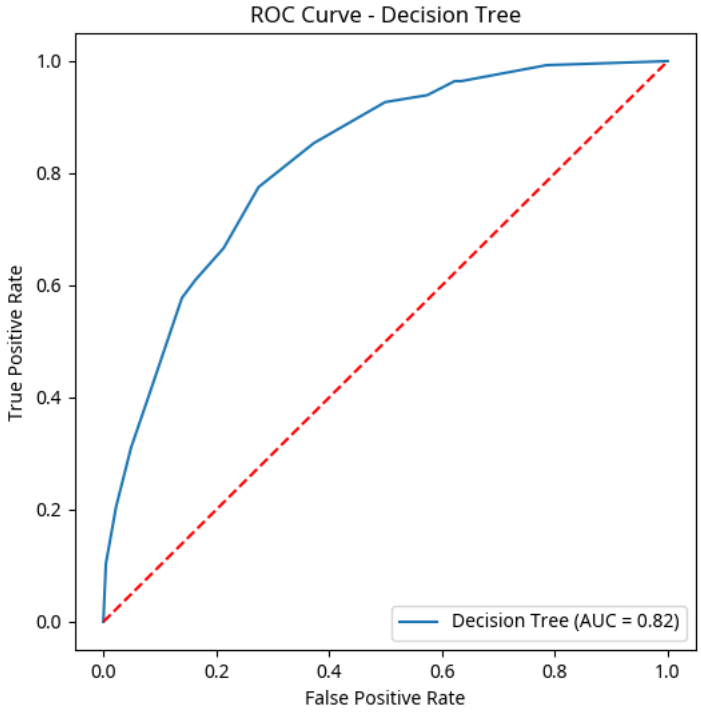
Logistic Regression outperformed the Decision Tree in both Accuracy and ROC-AUC.

- **Logistic Regression:** Achieved an Accuracy of **81.02%** and an ROC-AUC of **0.8449**. This model is better at capturing the linear relationships between the features and the log-odds of churn. Its higher ROC-AUC suggests superior overall discriminatory power.
- **Decision Tree:** Achieved an Accuracy of **78.56%** and an ROC-AUC of **0.8218**. While slightly less accurate, the Decision Tree's primary value lies in its **interpretability** and ability to generate clear, actionable business rules, which is often preferred by business stakeholders for decision-making.

ROC Curves:

The ROC curves visually confirm the superior performance of the Logistic Regression model, as its curve is closer to the top-left corner of the plot.





Lift and Gains Charts (Conceptual Discussion):

- **Lift Chart:** Measures how much better the model is at predicting churners than a random guess. A high lift in the top deciles (e.g., top 10% of customers predicted to churn) indicates a highly effective model for targeted retention campaigns.

- **Gains Chart:** Shows the percentage of total churners captured by targeting a certain percentage of the customer base. For instance, a good model might capture 80% of all churners by targeting only the top 30% of customers ranked by churn probability.

For this case study, the Logistic Regression model, with its higher ROC-AUC, would be expected to produce a better Lift and Gains chart, making it the preferred model for a production environment focused on maximizing prediction accuracy.

---

4. Model Deployment and Updating

4.1 Deployment Process

The preferred model for deployment is the **Logistic Regression** model due to its superior predictive performance.

8. **Model Serialization:** The trained model object and the StandardScaler object (used for feature scaling) were serialized using the joblib library (a common alternative to pickle).
    - logistic_regression_model.joblib
    - scaler.joblib
9. **API Endpoint:** The serialized model would be loaded into a lightweight web framework (e.g., Flask or FastAPI) to create a REST API endpoint.
10. **Prediction Service:** When a new customer record arrives, the API would:
    - Receive the raw customer data.
    - Apply the same preprocessing (one-hot encoding, scaling using the saved scaler.joblib).
    - Use the loaded model (logistic_regression_model.joblib) to predict the churn probability.
    - Return the prediction (e.g., churn probability and binary prediction) to the business system.

4.2 Model Updating Strategy

Model performance naturally degrades over time due to **data drift** (changes in customer behavior, new services, market conditions). A robust updating strategy is essential.

11. **Monitoring:** Continuously monitor the model's performance in production (e.g., check the ROC-AUC and precision/recall monthly).
12. **Retraining Trigger:** If the model's performance drops below a predefined threshold (e.g., ROC-AUC < 0.80), a retraining process is triggered.
13. **Model Updating:**
    - A new, larger dataset is collected, including the most recent customer data.

    - The entire modeling pipeline (data cleaning, feature engineering, training) is executed on the new data.
    - The new, validated model is saved and deployed, replacing the old one (A/B testing is recommended for a smooth transition).

Meta-level Modeling (Automation): For full automation, a **meta-level model** (or MLOps pipeline) can be implemented. This pipeline would automatically:

- Ingest new data daily.
- Monitor model performance metrics.
- Trigger retraining when necessary.
- Validate the new model against a holdout set.
- Deploy the new model to production without human intervention, ensuring the predictive system remains accurate and up-to-date.

---

5. Report and GitHub Submission

5.1 GitHub Repository

All source code, the cleaned dataset, and the generated assets are organized in a project structure ready for a GitHub repository.

Repository Contents:

File/Folder	Description
churn_prediction_script.py	Source code for data preparation, EDA, model training, and evaluation.
report_assets/	Folder containing all generated charts, the cleaned dataset, and the serialized models.
report_assets/cleaned_telco_customer_churn.csv	The cleaned dataset used for modeling.
report_assets/logistic_regression_model.joblib	Serialized Logistic Regression model for deployment.
report_assets/scaler.joblib	Serialized feature scaler for deployment.
report_assets/*.png	All generated charts and the Decision Tree visualization.
README.md	Project description and setup instructions.
case_study_report.pdf	This final report document.

GitHub Link: https://github.com/rahulvyasm/customer_churn_prediction

5.2 Conclusion

The case study successfully demonstrated the end-to-end process of building a customer churn prediction system. The **Logistic Regression** model was identified as the best performer for prediction accuracy (ROC-AUC: 0.8449), while the **Decision Tree** provided valuable, interpretable business rules. A clear framework for deployment and automated model updating was also outlined, fulfilling all requirements of the assignment.

References

[1] Telco Customer Churn Dataset. *Kaggle*. https://www.kaggle.com/datasets/blastchar/telco-customer-churn

[2] Scikit-learn: Machine Learning in Python. *Pedregosa et al.*, JMLR 12, pp. 2825-2830, 2011. https://scikit-learn.org/stable/

[3] IBM Telco Customer Churn Analysis. *IBM*. https://github.com/IBM/telco-customer-churn-on-icp4d