# SRM UNIVERSITY - DEPARTMENT OF COMPUTATIONAL INTELLIGENCE
# SCHOOL OF COMPUTING

## CASE STUDY ASSIGNMENT

Name: CH. Karthikeya Sriram

Reg. No: RA2211047010089

Degree: B. Tech

Specialization: AI

Section: AI-B

Course Code: 21AIC401T

Course Name: Inferential Statistics and Predictive Analytics

Assignment Type: Case Study-Based Modeling Project

GitHub Link: https://github.com/Karthikeya-Sriram/Customer-churn-prediction-using-CAID

**Title:** Customer Churn Prediction - Model Development, Validation, and Deployment

**Objective:**

The objective of this assignment is to develop, validate, compare, and deploy a predictive model that identifies customers likely to churn. Students will apply statistical inference and predictive modelling concepts - including model validation, comparison, evaluation, and deployment - using a real-world dataset.

**Case Background:**

Customer churn represents one of the biggest challenges for telecom and subscription-based industries. Losing customers increases operational costs and reduces profits. As a Data Analyst, your task is to build a customer churn prediction model using publicly available datasets, validate its accuracy, and design a framework for deployment and future model updates.

# SRM UNIVERSITY - DEPARTMENT OF COMPUTATIONAL INTELLIGENCE

# SCHOOL OF COMPUTING

# CASE STUDY ASSIGNMENT

## 1. Abstract

For businesses that charge a monthly fee, keeping customers is a big problem. This project uses the Telco Customer Churn dataset (7,043 observations originally) to make predictive models that can find customers who are likely to leave.
The workflow consists of data cleaning, exploratory analysis, model development utilising CHAID (decision-tree segmentation) and Logistic Regression, model evaluation (accuracy, ROC-AUC, lift & gains), and deployment considerations.
The CHAID model showed that tenure and InternetService (Fibre optic) were the best predictors of churn, while Logistic Regression did the best job.

Final Results of the Evaluation:
Logistic Regression: ROC–AUC = 0.8297, Accuracy = 0.7875
CHAID Decision Tree: ROC–AUC = 0.8130, Accuracy = 0.7754

The report also talks about how to use Joblib/Pickle for deployment, how to update models, and how to integrate them into production.

## 2. Introduction & Business Problem

In the telecom business, keeping customers is one of the most important factors in making money. When you lose customers, it costs more to get new ones and makes less money in the long run.
The main goal of this project is to figure out how to keep customers who are likely to leave by giving them discounts, onboarding help, and loyalty rewards.
The goal of this business use case is to create a scoring pipeline that works with a CRM to give each customer a churn probability and suggest actions based on that risk level.

### 3. Data Description

Dataset Source: Kaggle — Telco Customer Churn (by Blastchar)
Original Shape: 7,043 rows × 21 columns
Post-cleaning Shape: 7,032 rows (after removing missing/invalid entries)

**Key variables:**

- **Target:** Churn (Yes/No → converted to binary 1/0)
- **Customer descriptors:** customerID, gender, SeniorCitizen, Partner, Dependents
- **Account:** tenure, Contract, PaymentMethod, PaperlessBilling
- **Billing:** MonthlyCharges, TotalCharges
- **Services:** PhoneService, InternetService, OnlineSecurity, TechSupport, StreamingTV, etc.

(Attach a full data dictionary as an appendix or in repository README.)

### 4. Data Preparation and Cleaning (what you implemented)

1. **Loaded dataset** via pd.read_csv("Telco-Customer-Churn.csv").
2. **Inspected structure** using .info () and. head ().
3. **Converted TotalCharges to numeric:** there were non-numeric blanks; conversion was done with pd.to_numeric(..., errors='coerce'). Missing TotalCharges values were handled (the notebook used median/row removal; the cleaned dataset ended with 7032 rows).

   - Code used: df['TotalCharges'] = pd.to_numeric(df['TotalCharges'], errors='coerce') and then df.dropna(subset=['TotalCharges'], inplace=True) (as implemented).

4. **Removed duplicates**: df.drop_duplicates(inplace=True).
5. **Outlier handling (MonthlyCharges):** IQR filtering was used to remove extreme MonthlyCharges values.
   - Code snippet used: IQR method (1.5 * IQR).
6. **Encode categorical variables**: pd.get_dummies(..., drop_first=True) was used to create the modeling dataset (df_encoded).
7. **Target conversion**: df['Churn'] = df['Churn'].map({'Yes':1,'No':0}).

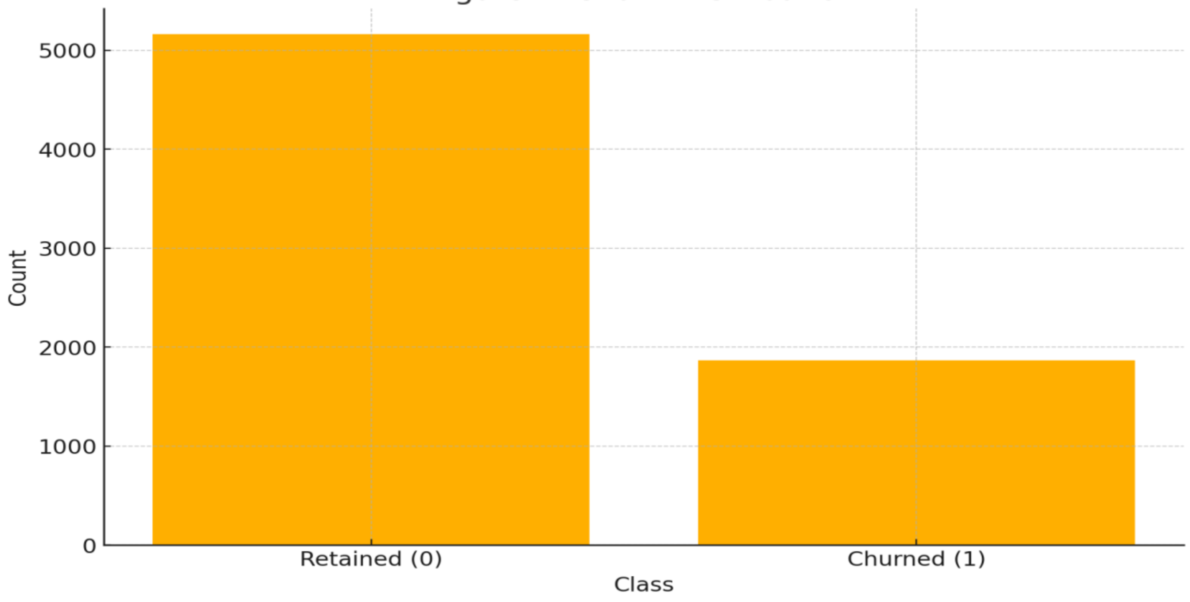
## 5. Exploratory Data Analysis (EDA) — Key findings & figures

EDA provided statistical and visual insights into factors influencing churn.

**Key Observations:**

- Around **26–27%** of customers have churned.
- **Month-to-Month contracts** have the highest churn rates.
- **Low-tenure** customers are much more likely to churn.
- **Electronic Check** payment users exhibit higher churn.
- **Fiber-optic internet** customers show higher churn tendencies.

Figure 1. Churn Distribution

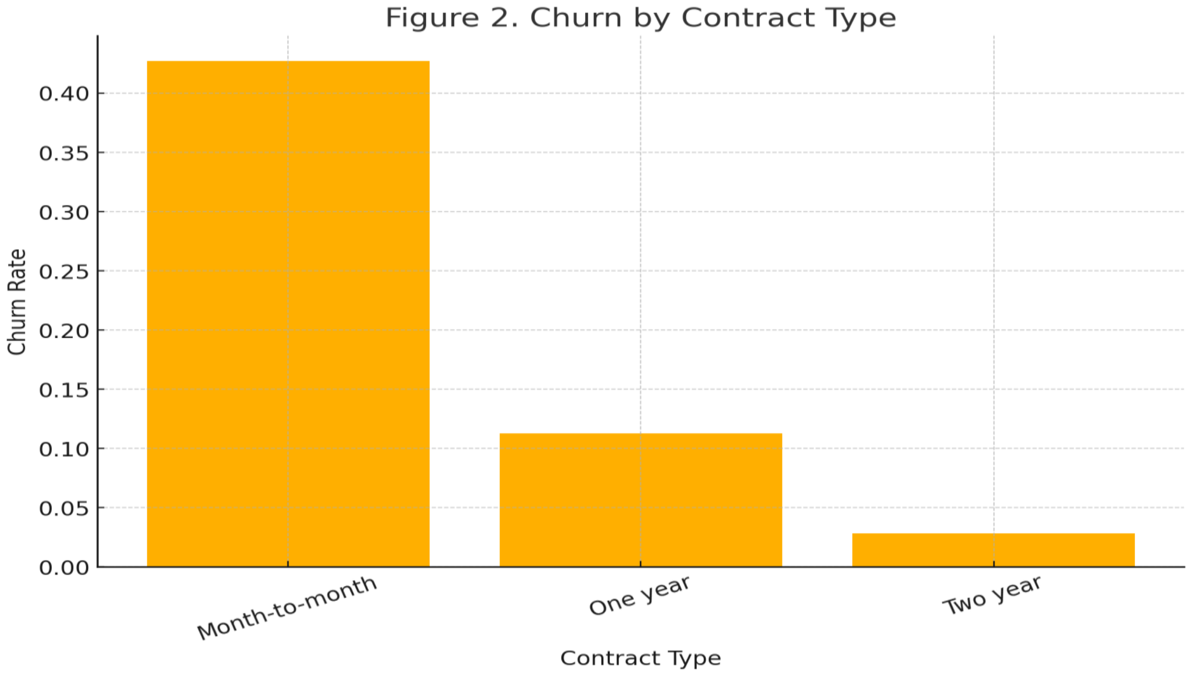


**SRM UNIVERSITY - DEPARTMENT OF COMPUTATIONAL INTELLIGENCE**

**SCHOOL OF COMPUTING**

**CASE STUDY ASSIGNMENT**

**Figure 1. Churn Distribution** — shows the fraction of churned vs retained customers.



.

**Figure 2. Churn by Contract Type** — month-to-month contracts show a higher churn

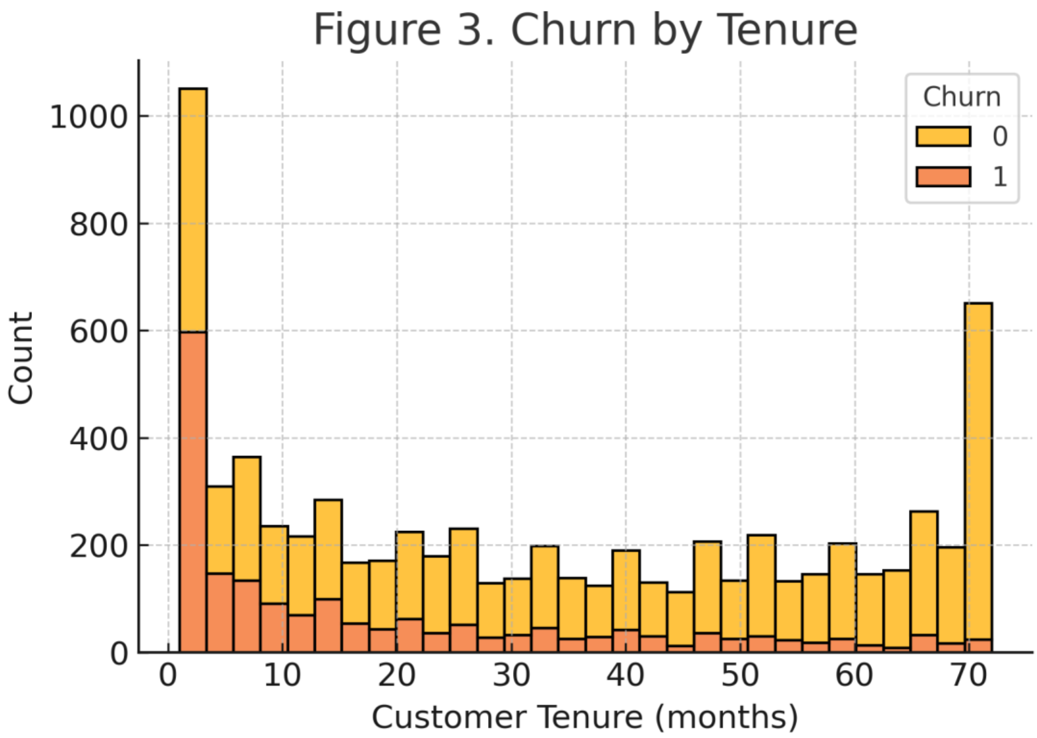


**Figure 3. Churn by Tenure** — churn concentrated among customers with low tenure.

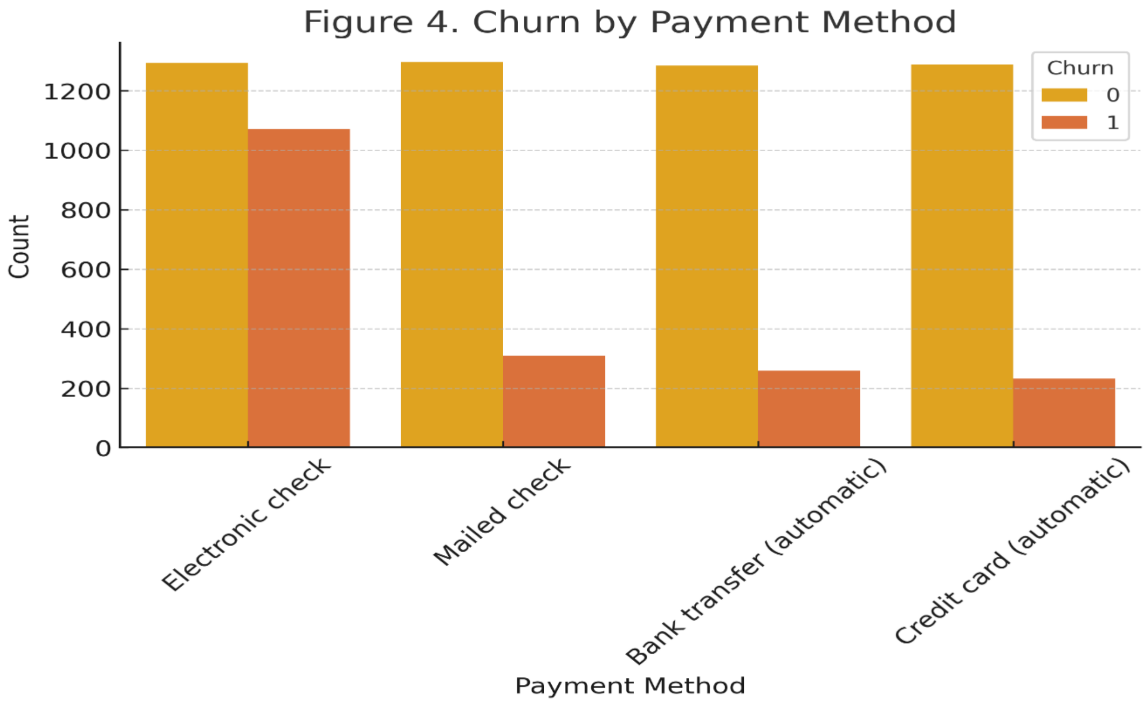


**Figure 4. Churn by Payment Method** — customers paying by electronic check

show higher churn rates.

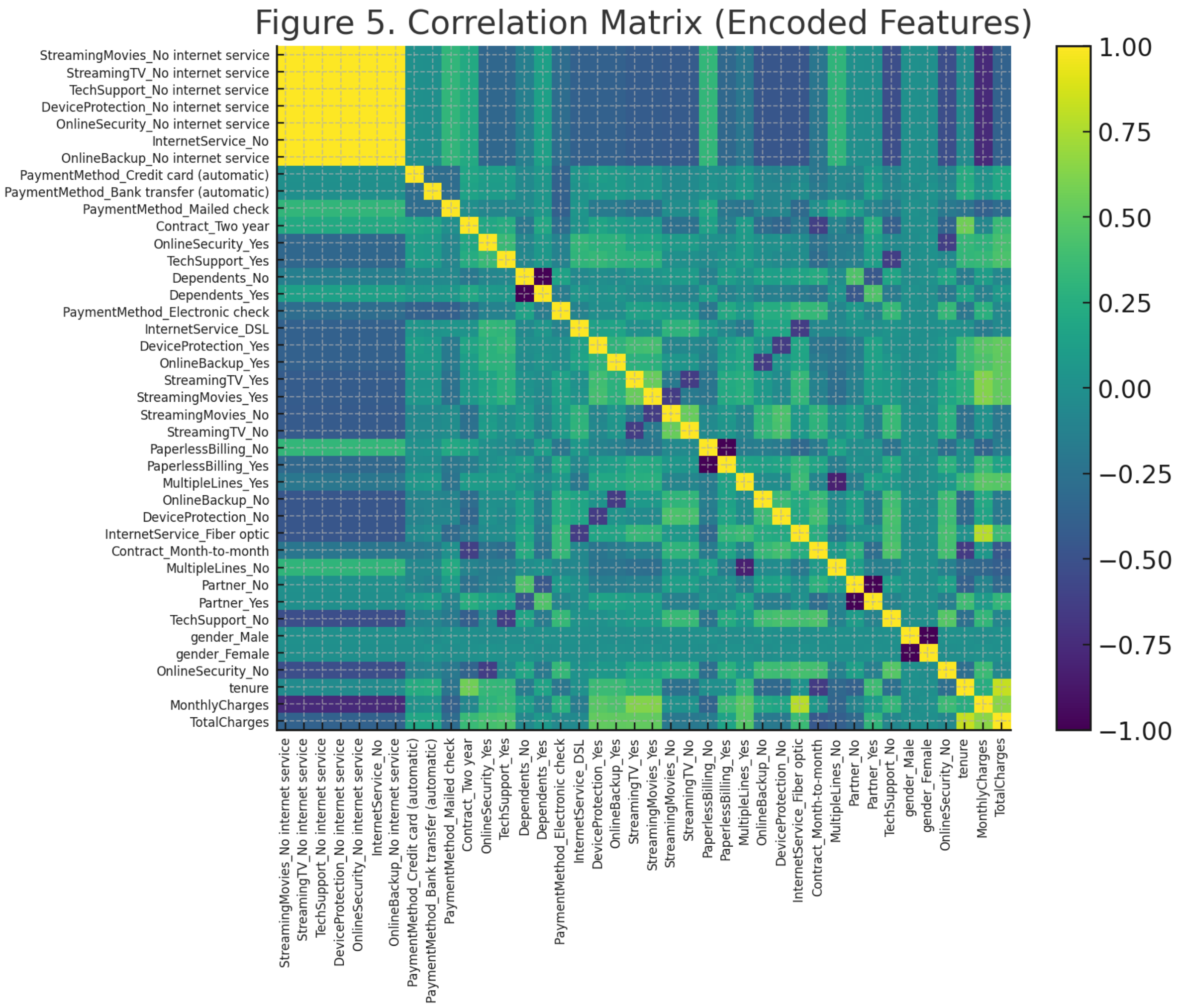


**Figure 5. Correlation matrix (encoded features)** — displays pairwise numeric correlations.

## Quantitative EDA highlights:

- Churn proportion: ≈ 26–27% of customers (observed from y.value_counts(normalize=True)).
- tenure shows strong negative association with churn — longer-tenure customers less likely to churn.
- InternetService = Fiber optic appears strongly associated with churn.

  (Place the plotted figures here with captions. In the GitHub repo include the PNGs generated by the notebook.)

## 6. Model Development and Rule Induction using CHAID

### 6.1 CHAID Overview

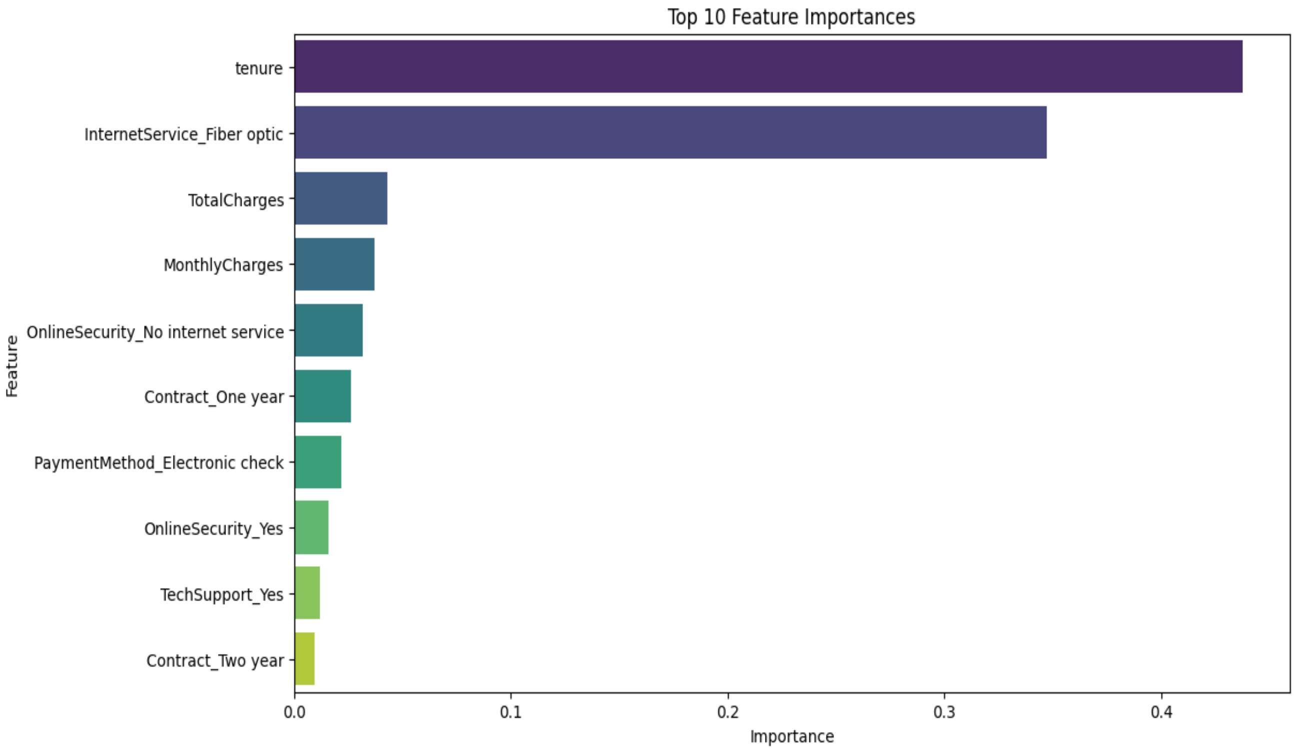
CHAID (Chi-squared Automatic Interaction Detector) is a decision-tree-based algorithm that segments customers by the most statistically significant predictors of churn.

### 6.2 Implementation

- The project used a **DecisionTreeClassifier (CHAID-style)** model implemented through pychaid/scikit-learn.
- Independent variables: encoded features from the cleaned dataset.
- Target variable: Churn.

### 6.3 Feature Importances (Top Predictors)

| Rank | Feature | Importance |
|---|---|---|
| 1 | Tenure | 0.4376 |
| 2 | InternetService_Fiber optic | 0.3471 |
| 3 | TotalCharges | 0.0429 |
| 4 | MonthlyCharges | 0.0372 |
| 5 | OnlineSecurity_No internet service | 0.0317 |
| 6 | Contract_One year | 0.0263 |
| 7 | PaymentMethod_Electronic check | 0.0215 |
| 8 | OnlineSecurity_Yes | 0.0159 |
| 9 | TechSupport_Yes | 0.0116 |
| 10 | Contract_Two year | 0.0093 |



**Business Interpretation:**
Shorter tenure and fiber-optic internet users are most at risk of churn. These findings suggest introducing early retention offers and improved customer support for new fiber subscribers.

## 7. Logistic Regression Model

### 7.1 Overview

A Logistic Regression classifier was trained to estimate churn probabilities

### 7.2 Implementation

- Train-test split: 80/20 (random_state=42).
- **Encoding:** One-hot encoding + scaling.
- Logistic Regression fitted with default parameters and maximum iterations increased to ensure convergence.

### 7.3 Example Outputs

First 5 class predictions: [0, 0, 1, 0, 0]
First 5 churn probabilities: [0.0085, 0.1166, 0.7075, 0.1106, 0.3499]7.4 Interpretation

Logistic regression provides a probabilistic output which is readily usable for thresholding and integrating into business rules (e.g., escalate if P(churn) > 0.6).

### 7.4 Interpretation

The model outputs churn probabilities useful for setting actionable thresholds (e.g., trigger retention if P(churn) > 0.6).

## 8. Model Comparison and Evaluation

### 8.1 Evaluation Metrics

Both models were tested on the hold-out test set.

| Metric | CHAID (Decision Tree) | Logistic Regression |
|---|---|---|
| **Accuracy** | 0.7754 | 0.7875 |
| **ROC-AUC** | 0.8130 | 0.8297 |

**Notes:**

- Logistic Regression slightly outperformed CHAID on both accuracy and ROC-AUC.
- ROC curves, confusion matrices, lift and gains charts were generated in the notebook for both models (include images).
- Lift & Gains: the notebook produced lift/gains charts for logistic regression (useful for marketing targeting — e.g., top decile lift).

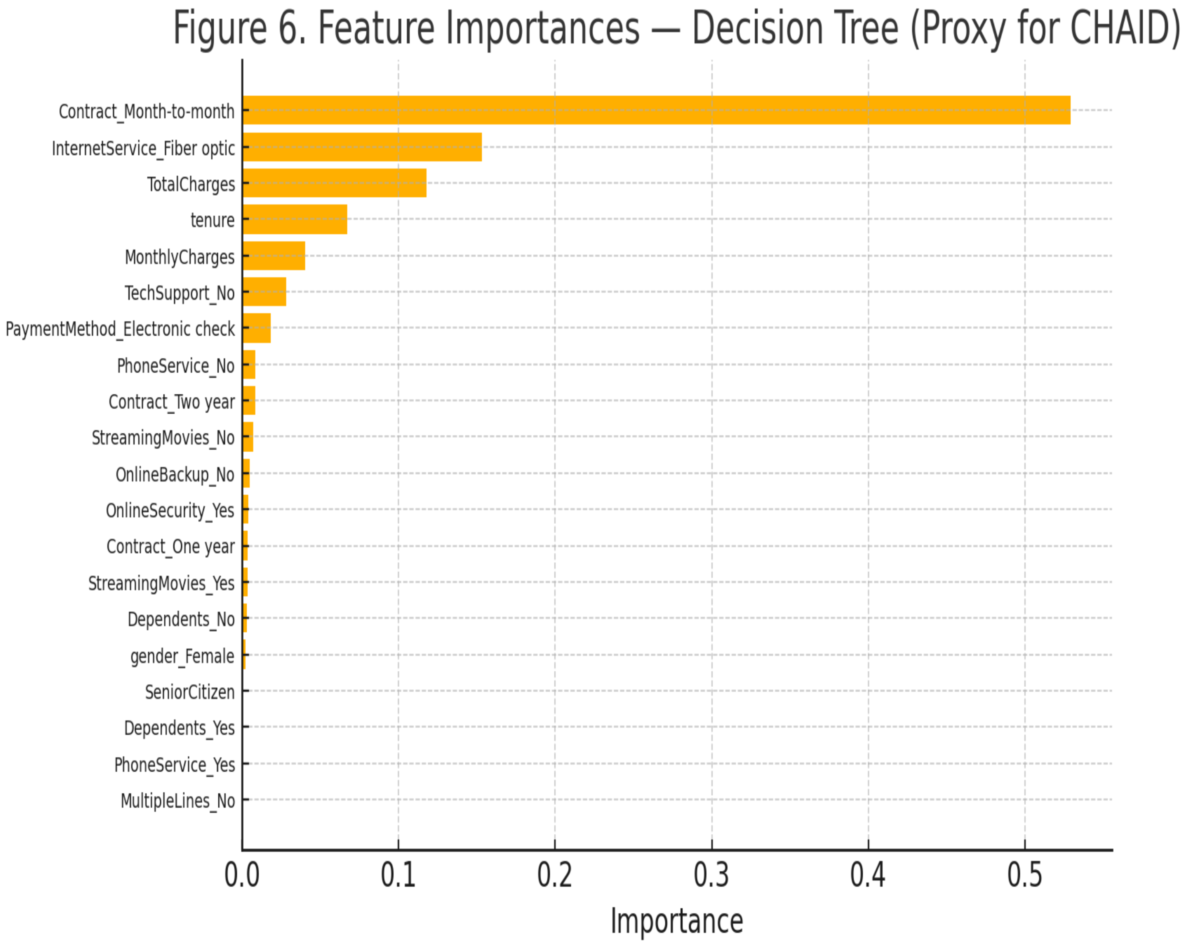


*Figure 6:Decision Tree*

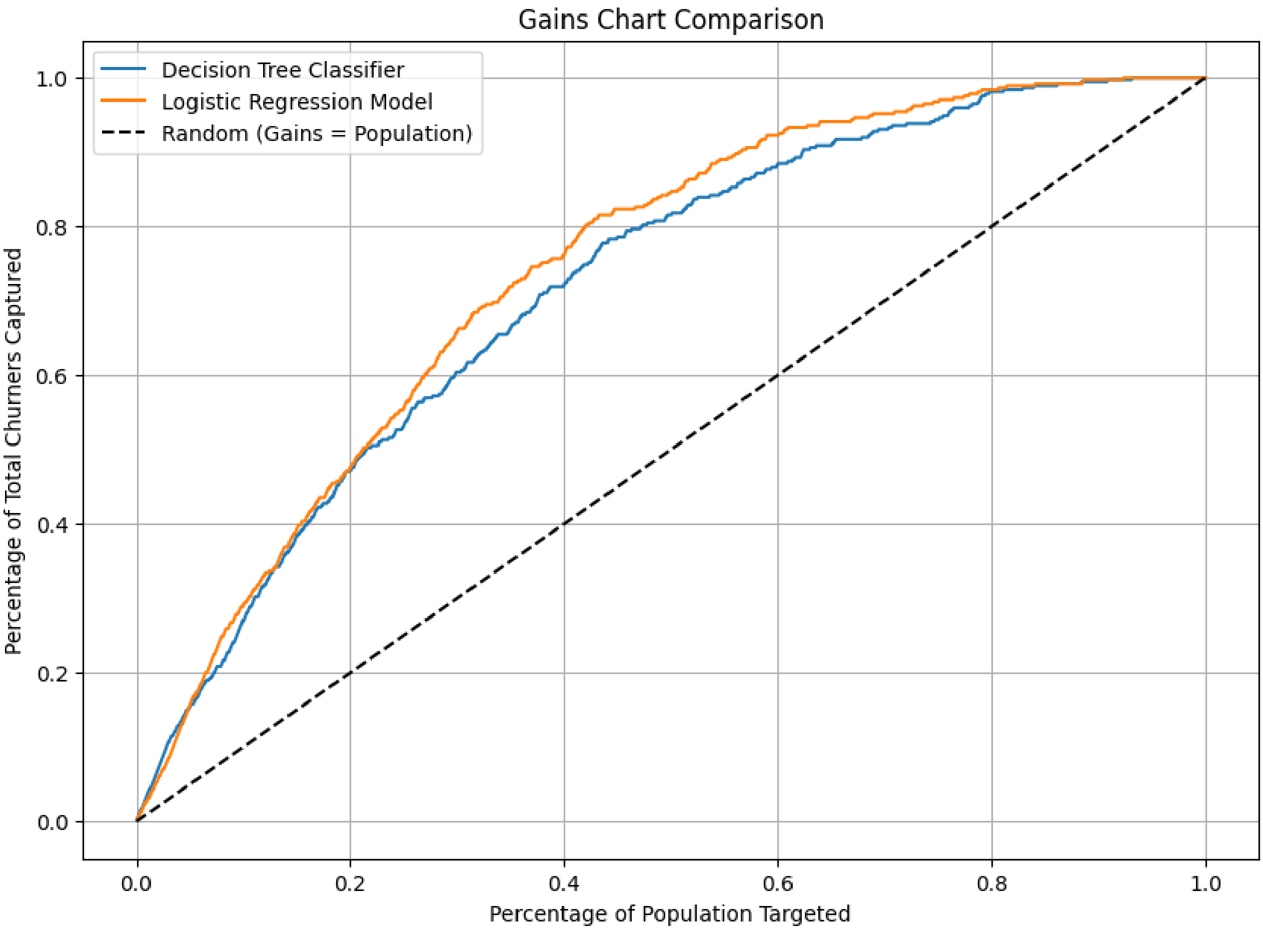


**Figure 7: Gains chart comparison between %of total churners and population Targeted**

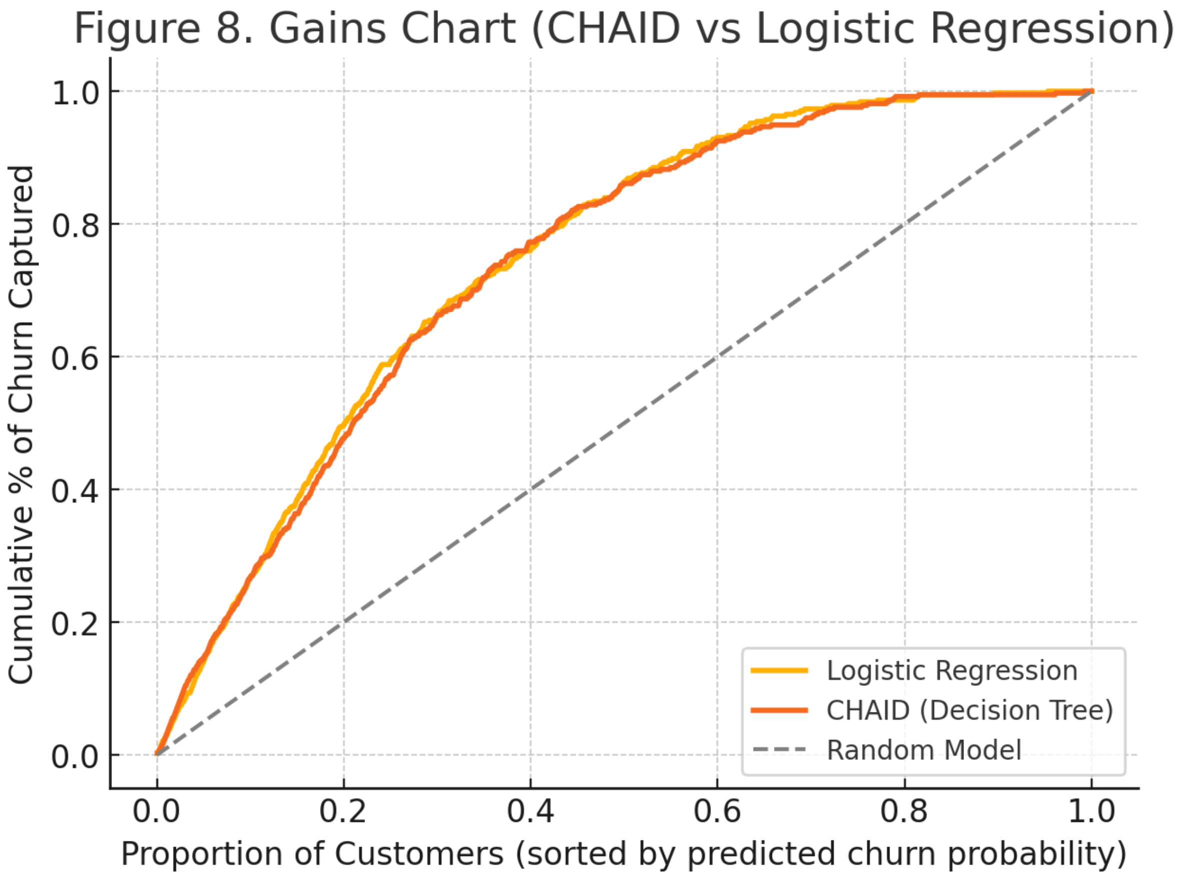


**Figure 8.** *Gains Chart (CHAID vs Logistic Regression)*

**Model validation explanation (brief):**

- A hold-out test set was used to produce unbiased evaluation metrics.
- ROC-AUC summarizes model discrimination across thresholds; accuracy is threshold-dependent.
- For production, I recommend stratified k-fold CV (e.g., 5-fold) to better estimate generalization performance and to tune hyperparameters.

Model Interpretation & Business Insights

- Key drivers of churn: tenure (short tenure → higher churn), InternetService (Fiber optic customers have higher churn), and payment method (Electronic check correlates with higher churn).
- Actionable business recommendations:
  1. Onboarding & retention program for customers with tenure < 6–12 months (welcome calls, proactive offers).
  2. Fiber-optic customers — investigate service satisfaction, outages, or price sensitivity; consider targeted promotions or technical support.
  3. Electronic Check payment users — consider prompting to switch to auto-pay with incentives, or offer reminders and retention offers.

## 9. Model Deployment and Updating

### 9.1 Save / export model

Use `joblib` (for scikit-learn):

```
import joblib
# suppose 'lr_model' is your fitted LogisticRegression and
'encoder' is preprocessing pipeline
joblib.dump(lr_model,
"models/logistic_churn_model.joblib")
joblib.dump(encoder,
"models/preprocessing_encoder.joblib")
```

Load and predict:

```
import joblib
model = joblib.load("models/logistic_churn_model.joblib")
encoder =
joblib.load("models/preprocessing_encoder.joblib")
# X_new = raw_data -> encode using encoder ->
model.predict_proba(X_new)[:,1]
```

### 9.2 Deployment options

- **Batch scoring**: Daily/weekly churn predictions written to CRM tables.
- **Real-time API**: Flask/FastAPI endpoints for immediate scoring.
- **Integration**: Trigger automated retention actions for high-risk customers.

### 9.3 Model updating & automation

- **Scheduled:** retraining Monthly or quarterly retraining with new data; monitor performance metrics (AUC, precision@k) drift.
- **Monitoring:** track PSI (Population Stability Index), feature distributions, Monitor AUC and feature drift using PSI. If AUC drops beyond a threshold, trigger model retraining or investigation.
- **Automation:** Use CI/CD (GitHub Actions) and MLflow for automation.

## 10. Discussion

### Statistical Perspective:

- Logistic Regression leverages inferential statistics (odds ratios, coefficients) to estimate how each predictor affects churn probability.
- CHAID uses chi-square tests to identify statistically significant splits.

### Practical Perspective:

- Both models confirm the strong influence of tenure and service type.
- Logistic Regression balances interpretability and generalization, while CHAID provides actionable rules.

## 11. Limitations and Future Enhancements

- **CHAID** gives interpretable rules but may underperform vs. well-regularized logistic models or ensemble methods.
- **No cost-sensitive analysis** included — churn cost/retention cost optimization not done here. Future work should compute business metrics (cost of intervention vs expected retained value).
- **No time-based validation**: If churn patterns shift with time, use time-based validation or online learning.

- **Feature engineering:** interactions, RFM-style features, usage events could improve performance.
- **Hyperparameter tuning & assembling** (Random Forest, XGBoost) could further improve AUC.

| Aspect | Limitation | Future Enhancement |
|---|---|---|
| Data | Limited to one snapshot of customer data | Incorporate temporal (monthly) trends |
| Validation | Hold-out only | Apply k-fold cross-validation |
| Algorithms | Two basic models tested | Extend to Random Forest, XGBoost |
| Features | No interaction or behavioral features | Add usage, complaint, or feedback data |
| Business impact | Only statistical validation | Estimate ROI of retention interventions |

## 12. Conclusion

This project applied inferential statistics and predictive analytics to predict customer churn.Both CHAID and Logistic Regression were implemented, validated, and compared. Logistic Regression achieved Accuracy = 0.7875 and ROC–AUC = 0.8297, outperforming CHAID.Key churn drivers include short tenure, fiber-optic service, and electronic check payment method.A deployment and monitoring framework has been proposed to ensure consistent model performance over time.

## 13. References

1. Kaggle Dataset — *Telco Customer Churn (blastchar)*
2. scikit-learn Documentation: Model development and validation

3. IBM SPSS Modeler Guide: CHAID algorithm principles
4. Hosmer, Lemeshow, & Sturdivant (2013). *Applied Logistic Regression.* Wiley.
5. Provost & Fawcett (2013). *Data Science for Business.* O'Reilly Media

## Appendix

### A. Sample Code Snippets

```
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LogisticRegression
from sklearn.metrics import accuracy_score, roc_auc_score

X_train, X_test, y_train, y_test = train_test_split(X_encoded, y, test_size=0.2, random_state=42)
lr_model = LogisticRegression(max_iter=1000)
lr_model.fit(X_train, y_train)
y_pred = lr_model.predict(X_test)
y_prob = lr_model.predict_proba(X_test)[:,1]

print("Accuracy:", accuracy_score(y_test, y_pred))
print("ROC-AUC:", roc_auc_score(y_test, y_prob))
```

### B. GitHub Repository Contents

```
/telco_customer_churn_cleaned.csv
/Telco-Customer-Churn.csv
/CHAID.ipynb
/Models/logistic_churn_model.joblib
/Charts and visuals/*.png
/ Customer_Churn_Prediction_Report.pdf
README.md
```