Implementing Predictive Analytics to Optimize Customer Churn Prediction

A Predictive Analytics Project for Customer Retention

Name of Intern: Adithya Vinod

Internship Program: HEPro Business Analytics Internship

Duration: November 2025 - December 2025

Submission Date: 17/12/2025

---

2. Table of Contents

3. Executive Summary

This project focuses on developing a **Customer Churn Prediction System** using machine learning techniques to identify customers at risk of leaving a telecommunications company. The primary objective was to build a predictive model that enables proactive customer retention strategies, ultimately reducing revenue loss and improving customer lifetime value.

The approach involved comprehensive data preprocessing, exploratory data analysis of 7,043 customer records with 21 attributes, and training multiple classification algorithms including Logistic Regression, Decision Tree, Random Forest, and Gradient Boosting. After rigorous cross-validation and hyperparameter tuning, the **Random Forest classifier** was selected as the final model.

The final deliverable includes a CRM-ready scoring system that categorizes customers into four risk tiers (Critical, High, Moderate, Low) with corresponding recommended actions, enabling the business to prioritize retention efforts effectively. The estimated net impact of implementing this model is **$8,110** in retained customer value.

4. Introduction

4.1 What is Customer Churn?

Customer churn, also known as customer attrition, refers to the phenomenon where customers discontinue their relationship with a company by canceling their subscriptions, switching to competitors, or simply ceasing to purchase products or services. In subscription-based industries such as telecommunications, SaaS, and streaming services, churn is measured as the percentage of customers who leave within a specific time period.

The formula for churn rate is:

```
Churn Rate = (Customers Lost During Period / Total Customers at Start of Period) × 100
```

4.2 Why is Churn Prediction Important for Businesses?

Churn prediction is critically important for several reasons:

Factor	Impact
Customer Acquisition Cost	Acquiring new customers costs 5-25x more than retaining existing ones
Revenue Protection	A 5% increase in retention can increase profits by 25-95%
Competitive Advantage	Proactive retention creates stronger customer relationships
Resource Optimization	Targeted interventions are more cost-effective than broad campaigns

4.3 Business Impact of Reducing Churn

For a telecommunications company, reducing churn directly impacts:

1. **Monthly Recurring Revenue (MRR):** Each retained customer continues contributing to stable, predictable revenue streams.
2. **Customer Lifetime Value (CLV):** Longer customer relationships translate to higher cumulative value per customer.
3. **Brand Reputation:** Satisfied, long-term customers often become brand advocates, driving organic growth.
4. **Operational Efficiency:** Reduced customer service costs associated with cancellations and win-back campaigns.

4.4 Problem Statement

> ***"Given historical customer data including demographics, service subscriptions, billing information, and usage patterns, develop a machine learning model that can accurately predict which customers are likely to churn. The model should provide actionable risk scores that enable the customer retention team to prioritize outreach efforts and implement targeted intervention strategies."***

Key Requirements:

- Achieve high precision to minimize false positives (avoid wasting resources on non-churners)
- Maintain acceptable recall to capture a significant portion of actual churners
- Provide interpretable results for business stakeholders
- Generate CRM-ready output for operational deployment

5. Project Goals and Objectives

Primary Goal

Develop an end-to-end machine learning pipeline for predicting customer churn with high accuracy and business applicability.

Specific Objectives:

#	Objective	Success Criteria
1	**Data Quality Assessment**	Identify and handle missing values, duplicates, and data inconsistencies
2	**Exploratory Analysis**	Discover key patterns and relationships between features and churn
3	**Feature Engineering**	Create meaningful features and remove data leakage columns
4	**Model Development**	Train and compare multiple classification algorithms
5	**Hyperparameter Tuning**	Optimize model performance using cross-validation
6	**Threshold Optimization**	Select optimal decision threshold for business requirements

7	**Model Interpretability**	Identify top predictive features using permutation importance
8	**Segmentation Analysis**	Evaluate model performance across customer segments
9	**Business Deployment**	Generate CRM-ready risk scores with actionable recommendations
10	**Documentation**	Create comprehensive stakeholder guidance and monitoring reports

Target Metrics

- **ROC-AUC Score:** ≥ 0.80
- **Precision:** ≥ 0.70 (to minimize false positives)
- **Interpretability:** Top 10 feature importance rankings

6. Daily Work Log

Week 1: Project Setup and Data Understanding

Day	Activities	Deliverables
Day 1	• Project initialization and environment setup • Library installation (pandas, scikit-learn, matplotlib) • Directory structure creation (data/, models/, reports/, figures/)	Project folder structure, requirements.txt
Day 2	• Dataset acquisition and initial loading • Understanding data schema and column meanings • Initial data profiling	Raw data loaded, column dictionary
Day 3	• Data quality assessment • Missing value analysis• Duplicate detection • Data type verification	Data quality report (data_quality.txt)
Day 4	• Column standardization and renaming • Target variable encoding (Yes/No → 1/0) • Numeric conversion for TotalCharges	Clean dataset with standardized columns
Day 5	• Leakage column identification	Feature list, training/test sets

	• Feature selection (19 features finalized) • Train-test split (80/20 stratified)	

Week 2: Exploratory Data Analysis

Day	Activities	Deliverables
Day 6	• Target variable distribution analysis • Class imbalance assessment (26.5% churn rate) • Baseline metrics calculation	Target balance visualization
Day 7	• Univariate analysis of categorical features • Contract type vs churn relationship • Internet service impact on churn	Bar charts, categorical insights
Day 8	• Numerical feature distributions • Tenure, monthly charges, total charges analysis • Correlation analysis	Histograms, correlation heatmap

Day 9	• Bivariate analysis • Feature interactions with target • Risk by contract type visualization	Risk by contract visualization
Day 10	• EDA documentation • Key insights compilation • Hypothesis formulation for modeling	EDA summary report

Week 3: Model Development and Evaluation

Day	Activities	Deliverables
Day 11	• Preprocessing pipeline creation • Numerical: median imputation + StandardScaler • Categorical: most frequent imputation + OneHotEncoder	ColumnTransformer pipeline
Day 12	• Baseline model (DummyClassifier) • Logistic Regression implementation • Cross-validation setup (5-fold stratified)	Baseline metrics, LR results
Day 13	• Decision Tree implementation • Random Forest with RandomizedSearchCV	DT and RF models, best params

	• Hyperparameter tuning (15 iterations)	
Day 14	• Gradient Boosting implementation • Model comparison across all metrics • Holdout validation on test set	All models trained, comparison table
Day 15	• Final model selection (Random Forest) • Threshold optimization for precision target • Confusion matrix and classification report	Final model, optimal threshold

Week 4: Deployment and Documentation

Day	Activities	Deliverables
Day 16	• Permutation importance analysis • Feature ranking (top 25 features) • Segment performance analysis	Feature importance report, segment CSV
Day 17	• Risk tier creation (Low/Moderate/High/Critical) • Recommended action mapping • CRM-ready output generation	churn_scored_for_crm.csv

Day 18	• Model serialization (joblib) • Batch scoring script creation • PSI drift monitoring setup	Saved model, run_scoring.py
Day 19	• Metrics summary JSON • Stakeholder guide creation • ROI estimation ($8,110 net impact)	metrics_summary.json, stakeholder_guide.md
Day 20	• Final visualizations • Project report compilation • Code documentation and cleanup	Complete project report

7. Project Development Phase

7.1 Dataset Description

The dataset used for this project is the **Telco Customer Churn Dataset** containing information about telecommunications customers and their service subscriptions.

Dataset Overview:

Attribute	Value
Source	data/raw/customer_churn.csv
Total Records	7,043
Total Features	21 (after cleaning: 19 used for modeling)
Target Variable	churn (binary: 0 = No, 1 = Yes)
Baseline Churn Rate	26.5%

Feature Categories:

1. Customer Demographics

Feature	Description	Type
customer_id	Unique customer identifier	ID (excluded)

gender	Customer gender	Categorical
senior_citizen	Whether customer is senior (65+)	Binary
Partner	Whether customer has a partner	Categorical
Dependents	Whether customer has dependents	Categorical

2. Service Subscriptions

Feature	Description	Type
phone_service	Phone service subscription	Categorical
multiple_lines	Multiple phone lines	Categorical
internet_service	Internet service type (DSL/Fiber/None)	Categorical
online_security	Online security add-on	Categorical
online_backup	Online backup add-on	Categorical
device_protection	Device protection add-on	Categorical
tech_support	Tech support add-on	Categorical
streaming_tv	Streaming TV add-on	Categorical

streaming_movies	Streaming movies add-on	Categorical

3. Account Information

Feature	Description	Type
tenure	Months as customer	Numerical
contract_type	Contract duration (Month-to-month/1yr/2yr)	Categorical
autopay	Paperless billing status	Binary
payment_method	Payment method used	Categorical
monthly_charges	Monthly bill amount	Numerical
total_charges	Total amount charged	Numerical

7.2 Data Preprocessing

7.2.1 Data Loading and Initial Cleaning

```
# Load data and standardize column names
DATA_PATH = "data/raw/customer_churn.csv"
df = pd.read_csv(DATA_PATH)
df.columns = df.columns.map(lambda c: str(c).strip())

# Standardize column names for consistency
rename_map = {
    "customerID":"customer_id", "SeniorCitizen":"senior_citizen",
    "PhoneService":"phone_service", "MultipleLines":"multiple_lines",
```

```
    "InternetService":"internet_service", "Contract":"contract_type",
    "MonthlyCharges":"monthly_charges", "TotalCharges":"total_charges",
    "Churn":"churn"
    # ... additional mappings
}
df = df.rename(columns={k:v for k,v in rename_map.items() if k in
df.columns})
```

7.2.2 Target Variable Encoding

The target variable "Churn" was converted from categorical (Yes/No) to binary (1/0):

```
if not pd.api.types.is_numeric_dtype(df["churn"]):
    df["churn"] = df["churn"].astype(str).str.strip().str.lower().map({
        "yes":1, "no":0, "1":1, "0":0
    })
```

7.2.3 Handling Missing Values

```
# Convert TotalCharges to numeric (some were blank strings)
df["total_charges"] = pd.to_numeric(df.get("total_charges"),
errors="coerce")

# Fill missing total_charges with calculated value
df["total_charges"] = df["total_charges"].fillna(
    df["monthly_charges"] * df["tenure"]
)
```

Missing Values Summary:

Column	Missing Count	Treatment
total_charges	11	Filled with monthly_charges × tenure
Other columns	0	No treatment needed

7.2.4 Data Leakage Prevention

Columns that could leak future information were identified and excluded:

```
denylist_exact = {
    "cancel_date", "cancellation_flag", "termination_reason",
    "end_date", "churn_date", "churn score", "churn reason",
    "churn value", "churn label", "cltv", "count"
```

```
}
leak_cols = [c for c in df.columns if c.lower() in denylist_exact]
# Result: No leakage columns found in this dataset
```

7.2.5 Final Data Cleaning

```
df = df.dropna(subset=["churn"]).drop_duplicates().copy()
df["churn"] = df["churn"].astype(int)

# Final shape
print("Shape:", df.shape)  # (7043, 21)
print("Baseline churn:", round(df["churn"].mean(), 3))  # 0.265
```

7.2.6 Preprocessing Pipeline

A robust preprocessing pipeline was created using scikit-learn's ColumnTransformer:

```
from sklearn.pipeline import Pipeline
from sklearn.compose import ColumnTransformer
from sklearn.preprocessing import OneHotEncoder, StandardScaler
from sklearn.impute import SimpleImputer

# Numerical preprocessing
num_pipe = Pipeline([
    ("imputer", SimpleImputer(strategy="median")),
    ("scaler", StandardScaler())
])

# Categorical preprocessing
cat_pipe = Pipeline([
    ("imputer", SimpleImputer(strategy="most_frequent")),
    ("onehot", OneHotEncoder(handle_unknown="ignore", sparse_output=True))
])

# Combined preprocessor
preprocessor = ColumnTransformer([
    ("num", num_pipe, selector(dtype_include=np.number)),
    ("cat", cat_pipe, selector(dtype_exclude=np.number))
])
```

Preprocessing Summary:

Data Type	Imputation	Transformation
Numerical	Median	StandardScaler (z-score normalization)
Categorical	Most Frequent	One-Hot Encoding

7.3 Exploratory Data Analysis (EDA)

7.3.1 Target Variable Distribution - Visualization: Churn Class Distribution

```
sns.countplot(x="churn", data=df)
plt.title("Churn class balance")
plt.tight_layout()
plt.savefig("figures/target_balance.png")
```

Interpretation:

- **Non-Churners (0):** 5,174 customers (73.5%)
- **Churners (1):** 1,869 customers (26.5%)
- The dataset exhibits **moderate class imbalance** with approximately 3:1 ratio
- This imbalance was addressed using `class_weight="balanced"` in model training

7.3.2 Contract Type vs Churn Analysis

Key Insight: Contract Type is the Strongest Predictor

Contract Type	Avg. Churn Risk Score	Churn Rate
Month-to-month	0.42	High
One year	0.18	Medium

Two year	0.08	Low

Interpretation:

- Customers on **month-to-month contracts have 5x higher churn risk** compared to two-year contracts
- This indicates that contract flexibility, while attractive to customers, correlates strongly with attrition
- **Business Recommendation:** Incentivize customers to migrate to longer-term contracts through discounts or added benefits

7.3.3 Tenure Distribution and Churn Relationship

Key Finding: Tenure Shows Strong Negative Correlation with Churn

Tenure Range	Churn Rate	Observation
0-12 months	~45%	New customers at highest risk
13-24 months	~30%	Moderate risk period
25-48 months	~20%	Relationship stabilizing
48+ months	~10%	Loyal customers, low risk

Interpretation:

- The **first year is critical** for customer retention
- Customers who stay beyond 12 months are significantly less likely to churn
- **Business Recommendation:** Focus retention efforts on the first 12 months with onboarding programs and early engagement

7.3.4 Monthly Charges Analysis

Risk Distribution by Monthly Charges:

```
seg = pd.qcut(X_test["monthly_charges"], q=3, labels=["low","mid","high"])
```

Charge Segment	Count	Precision	Recall
Low ($18-35)	~470	0.65	0.38
Medium ($35-70)	~470	0.72	0.45
High ($70-118)	~470	0.71	0.52

Interpretation:

- Customers with **higher monthly charges show slightly higher churn rates**
- The model performs consistently across all charge segments
- High-value customers churning represents greater revenue loss per customer

7.3.5 Internet Service Type Impact

Internet Service	Churn Risk	Observation
Fiber Optic	High	Fastest growth, highest churn
DSL	Medium	Stable customer base
No Internet	Low	Phone-only customers rarely churn

Interpretation:

- **Fiber optic customers churn more** despite having the premium service
- This may indicate competitive pressure in the fiber market or service quality issues
- **Business Recommendation:** Investigate service quality and pricing for fiber customers

7.3.6 Churn Risk Score Distribution

Visualization: Overall Risk Score Distribution

```
sns.histplot(out["churn_risk_score"], bins=30)
plt.title("Churn risk score distribution")
```

Interpretation:

- Distribution shows clear separation between low-risk (left peak) and high-risk (right tail) customers
- Majority of customers fall in the low-risk category (score < 0.3)
- A distinct high-risk segment exists with scores > 0.7

7.3.7 Top Feature Importance (Permutation Importance)

Rank	Feature	Importance Score
1	**contract_type**	0.0586
2	**tenure**	0.0254
3	**internet_service**	0.0173
4	**total_charges**	0.0137
5	**online_security**	0.0056

6	**tech_support**	0.0042
7	**payment_method**	0.0017
8	**streaming_movies**	0.0011
9	**multiple_lines**	0.0010
10	**streaming_tv**	0.0008

Interpretation:

- **Contract type dominates** with more than 2x the importance of the next feature
- **Tenure** confirms that customer relationship length is crucial
- **Service add-ons** (online security, tech support) have protective effects against churn
- Demographic features (gender, senior_citizen) have minimal impact

7.4 Model Development

7.4.1 Train-Test Split

```
X_train, X_test, y_train, y_test = train_test_split(
    X_full, y_full,
    test_size=0.2,
    stratify=y_full,
    random_state=42
)
print("Training set:", X_train.shape)  # (5634, 19)
print("Test set:", X_test.shape)       # (1409, 19)
```

Dataset	Samples	Features	Churn Rate
Training	5,634	19	26.5%

Test	1,409	19	26.5%

Stratification ensured equal churn distribution across splits.

7.4.2 Cross-Validation Strategy

```
cv = StratifiedKFold(n_splits=5, shuffle=True, random_state=42)
scoring = ["roc_auc", "precision", "recall", "f1"]
```

- **5-Fold Stratified Cross-Validation** to maintain class balance
- Multiple metrics tracked for comprehensive evaluation

7.4.3 Models Implemented

1. Baseline Model (DummyClassifier)

```
dummy = Pipeline([
    ("prep", preprocessor),
    ("model", DummyClassifier(strategy="stratified", random_state=42))
])
```

Purpose: Establish minimum performance threshold

2. Logistic Regression

```
logreg = Pipeline([
    ("prep", preprocessor),
    ("model", LogisticRegression(
        max_iter=400,
        class_weight="balanced",
        solver="liblinear",
        random_state=42
    ))
])
```

Rationale: Simple, interpretable baseline classifier

3. Decision Tree

```
dt = Pipeline([
    ("prep", preprocessor),
    ("model", DecisionTreeClassifier(
        max_depth=None,
```

```
        min_samples_leaf=2,
        random_state=42,
        class_weight="balanced"
    ))
])
```

Rationale: Captures non-linear relationships, fully interpretable

4. Random Forest (with Hyperparameter Tuning)

```
rf_base = Pipeline([
    ("prep", preprocessor),
    ("model", RandomForestClassifier(
        class_weight="balanced",
        n_jobs=-1,
        random_state=42
    ))
])

param_dist = {
    "model__n_estimators": np.arange(250, 601, 50),
    "model__max_depth": np.arange(4, 21, 2),
    "model__min_samples_leaf": [1,2,3,4,5,8],
    "model__max_features": ["sqrt","log2", None]
}

rf_search = RandomizedSearchCV(
    rf_base, param_dist,
    n_iter=15, cv=cv,
    scoring="roc_auc",
    n_jobs=-1, random_state=42
)
```

Rationale: Ensemble method with reduced overfitting, tuned for optimal performance

5. Gradient Boosting

```
gb = Pipeline([
    ("prep", preprocessor),
    ("model", GradientBoostingClassifier(random_state=42))
])
```

Rationale: Sequential ensemble with strong predictive power

7.4.4 Why These Models Were Chosen

Model	Reason for Selection
Logistic Regression	Interpretable coefficients, good for establishing linear baseline
Decision Tree	Visual interpretability, captures non-linear patterns
Random Forest	Robust ensemble, handles mixed data types well
Gradient Boosting	State-of-the-art performance for tabular data

7.5 Model Evaluation

7.5.1 Cross-Validation Results

Model	ROC-AUC	Precision	Recall	F1 Score
Dummy (Baseline)	0.507	0.275	0.278	0.276
Logistic Regression	0.846	0.517	0.801	0.628
Decision Tree	0.693	0.480	0.593	0.530
Random Forest	**0.847**	-	-	-
Gradient Boosting	0.848	0.662	0.529	0.588

Key Observations:

- All models significantly outperform the random baseline
- **Gradient Boosting achieves the highest CV AUC (0.848)**
- **Logistic Regression offers the best recall (0.801)** but with lower precision
- Random Forest and Gradient Boosting are closely matched

7.5.2 Holdout Validation Results

```
holdout_auc = {
    "LogisticRegression": 0.843,
    "DecisionTree": 0.721,
    "RandomForest": 0.844,
    "GradientBoosting": 0.841
}
```

Model	Holdout AUC	Generalization
LogisticRegression	0.843	Good
DecisionTree	0.721	Overfitting detected
RandomForest	**0.844**	**Best**
GradientBoosting	0.841	Good

Selected Model: Random Forest

- Highest holdout AUC (0.844)
- Consistent performance between CV and holdout
- Good balance of interpretability and predictive power

7.5.3 Threshold Optimization

The default threshold of 0.5 was optimized to meet a **target precision of 0.70**:

```
target_precision = 0.7
feasible = np.where(prec[:-1] >= target_precision)[0]
chosen_thr = float(thr[feasible[0]]) if len(feasible) else float(best_thr)
# Optimal threshold: 0.736
```

7.5.4 Final Model Performance (at threshold = 0.736)

Metric	Value	Interpretation
ROC-AUC	0.844	Excellent discrimination
PR-AUC	0.652	Good for imbalanced data
Precision	0.702	70% of predicted churners actually churn
Recall	0.460	46% of actual churners are identified
F1 Score	0.556	Balanced precision-recall tradeoff
Accuracy	0.805	Overall correctness

7.5.5 Confusion Matrix

```
             Predicted
              No    Yes
Actual  No   962    73
        Yes  202   172
```

Metric	Count	Business Meaning
True Negatives	962	Correctly identified loyalists
False Positives	73	Unnecessary retention efforts
False Negatives	202	Missed churners (revenue loss)
True Positives	172	Successful churn predictions

7.5.6 Classification Report

```
              precision    recall  f1-score   support

           0      0.826     0.929     0.875      1035
           1      0.702     0.460     0.556       374

    accuracy                          0.805      1409
   macro avg      0.764     0.695     0.715      1409
weighted avg      0.793     0.805     0.790      1409
```

7.5.7 Model Comparison Summary

Rank	Model	AUC	Precision	Recall	F1	Selection Rationale
1	**Random Forest**	0.844	0.702*	0.460*	0.556*	Best AUC, robust generalization
2	Gradient Boosting	0.841	0.662	0.529	0.588	Close second, slower training
3	Logistic Regression	0.843	0.517	0.801	0.628	High recall, low precision
4	Decision Tree	0.721	0.480	0.593	0.530	Overfitting, poor generalization

*At optimized threshold of 0.736

Final Model Selected: Random Forest

7.6 Final Insights & Recommendations

7.6.1 Key Predictive Factors

Based on permutation importance analysis, the following factors are most predictive of churn:

Priority	Factor	Recommendation

1	Contract Type	Incentivize migration from month-to-month to annual contracts
2	Tenure	Focus on first-year customer experience and onboarding
3	Internet Service	Investigate fiber optic service quality and pricing
4	Total Charges	Monitor high-value customers for retention opportunities
5	Online Security	Bundle security add-ons to increase stickiness
6	Tech Support	Promote tech support services to reduce frustration

7.6.2 Risk Tier Framework

The model outputs are categorized into actionable tiers:

Risk Tier	Score Range	% of Customers	Recommended Action
Critical	0.95 - 1.00	5%	Immediate outreach + tailored offer
High	0.85 - 0.95	10%	Proactive call + targeted discount
Moderate	0.60 - 0.85	25%	Email check-in + survey
Low	0.00 - 0.60	60%	Nurture campaigns

7.6.3 ROI Estimation

```
cost_per_contact = $2.00
benefit_retained = $50.00

True Positives (TP) = 172
False Positives (FP) = 73

Net Impact = (172 × $50) - (245 × $2) = $8,600 - $490 = $8,110
```

Estimated Net Impact: $8,110 per scoring cycle on the test set

Annualized projection (assuming monthly scoring):

- **Annual Net Benefit: ~$97,320**

7.6.4 Business Recommendations

Immediate Actions:

1. **Deploy CRM integration** using `churn_scored_for_crm.csv`
2. **Train retention team** on risk tier prioritization
3. **Create retention scripts** for Critical and High tiers
4. **Set up automated email campaigns** for Moderate tier

Strategic Initiatives:

1. **Contract migration incentives** - Primary lever for churn reduction
2. **Enhanced onboarding program** - Target first 90 days
3. **Service quality review** - Focus on fiber optic customers
4. **Add-on bundling strategy** - Security and tech support packages

Model Governance:

1. **Monthly performance monitoring** using holdout metrics
2. **Quarterly retraining** with fresh data
3. **Drift detection** using PSI reports (reports/psi_top_shifts.csv)

7.7 References

1. Scikit-learn Documentation - https://scikit-learn.org/stable/
2. Telco Customer Churn Dataset - Kaggle
3. Python Data Science Handbook - Jake VanderPlas
4. Introduction to Machine Learning with Python - Andreas Müller & Sarah Guido
5. Customer Churn Prediction Best Practices - Towards Data Science
6. Precision-Recall Analysis for Imbalanced Classification - Machine Learning Mastery
7. Feature Importance Techniques - Scikit-learn User Guide

8. Project Conclusion

This project successfully developed and deployed a **Customer Churn Prediction System** that meets all defined objectives. The key accomplishments include:

Technical Achievements

- ✅ Built an end-to-end ML pipeline with robust preprocessing
- ✅ Evaluated 4 classification algorithms with rigorous cross-validation
- ✅ Achieved **ROC-AUC of 0.844** (exceeding 0.80 target)
- ✅ Achieved **Precision of 70.2%** (meeting 0.70 target)
- ✅ Identified contract type and tenure as primary churn drivers
- ✅ Created interpretable risk tiers for business stakeholders

Business Deliverables

- ✅ CRM-ready scoring file with risk tiers and recommended actions
- ✅ Stakeholder guidance document for operationalization
- ✅ Batch scoring script for ongoing predictions
- ✅ Data drift monitoring framework
- ✅ Estimated $8,110 net impact per scoring cycle

Model Sustainability

The solution includes:

- Serialized model (`models/final_churn_model.joblib`)
- PSI-based drift monitoring (`reports/psi_top_shifts.csv`)
- Retraining recommendations in stakeholder guide
- Segment performance tracking for continuous improvement

The Random Forest model strikes an optimal balance between predictive accuracy, interpretability, and operational simplicity, making it suitable for production deployment in a customer retention workflow.

9. Challenges and Solutions

Challenge	Description	Solution Implemented
Missing Values in TotalCharges	11 records had blank TotalCharges values that couldn't be directly converted to numeric	Imputed using calculated value: `monthly_charges × tenure`, which represents the logical total for those customers
Class Imbalance	26.5% churn rate created bias toward predicting "No Churn"	Used `class_weight="balanced"` parameter in all classifiers to give equal importance to both classes
Data Leakage Risk	Potential for post-churn features to leak future information	Created a denylist of suspicious columns and systematically excluded ID columns before modeling
Threshold Selection	Default 0.5 threshold didn't meet precision requirements	Analyzed precision-recall curve to find threshold (0.736) meeting 70% precision target
Model Overfitting	Decision Tree showed poor generalization (CV: 0.69,	Selected Random Forest which uses bagging to reduce overfitting; applied min_samples_leaf constraints

	Holdout: 0.72 AUC drop)	
Hyperparameter Tuning Complexity	Full grid search on Random Forest would require excessive computation	Used RandomizedSearchCV with 15 iterations to efficiently explore parameter space
Feature Interpretability	Black-box nature of ensemble models	Applied permutation importance to identify key features without relying on model internals
Categorical Variable Encoding	High cardinality categoricals could create sparse features	Used OneHotEncoder with `handle_unknown="ignore"` to gracefully handle unseen categories in production
Production Deployment Readiness	Model needed to be deployable without Jupyter notebook	Created standalone `run_scoring.py` script and serialized complete pipeline including preprocessing
Performance Across Segments	Concern about model fairness across customer segments	Conducted segment analysis by monthly_charges to verify consistent performance

10. Learnings and Key Takeaways

Technical Learnings

1. **Pipeline Architecture is Essential**
    - Encapsulating preprocessing with the model ensures consistent transformations in production
    - ColumnTransformer elegantly handles mixed data types
2. **Cross-Validation Prevents Overfitting**
    - The Decision Tree's poor holdout performance demonstrated the importance of CV
    - Stratified K-Fold maintained class balance across splits
3. **Threshold Optimization is Business-Critical**
    - Default thresholds rarely align with business objectives
    - Precision-recall curves enable data-driven threshold selection
4. **Feature Importance Varies by Method**
    - Permutation importance provides model-agnostic insights
    - Contract type emerged as dominant factor across all analyses
5. **Ensemble Methods Excel on Tabular Data**
    - Random Forest and Gradient Boosting consistently outperformed simpler models
    - Class weighting effectively addressed imbalance

Business Learnings

1. **Churn is Predictable**
    - With the right features, ~85% discrimination is achievable
    - Behavioral signals (tenure, contract) outweigh demographics
2. **Early Intervention Matters**
    - First-year customers show highest churn risk
    - Onboarding quality directly impacts retention
3. **Contract Structure Drives Loyalty**
    - Month-to-month flexibility comes at retention cost
    - Incentivizing longer commitments pays dividends
4. **ROI Quantification Enables Buy-In**
    - Translating predictions to dollar impact resonates with stakeholders
    - Even conservative estimates show significant value

Professional Learnings

1. **Documentation is Deliverable**
    - Stakeholder guides bridge technical and business audiences
    - Clear recommendations accelerate adoption
2. **Monitoring Sustains Value**
    - PSI tracking catches data drift before model degradation
    - Quarterly retraining maintains performance
3. **Reproducibility Requires Discipline**
    - Random seeds ensure consistent results

    - Directory structure organizes artifacts
4. **End-to-End Thinking is Valuable**
    - Scoring scripts demonstrate production readiness
    - CRM integration shows practical application