Predicting Customer Attrition: A Machine Learning Approach

Enterprise Data Science & ML in Production I

WINTER 2024

SECTION 078
Group 01

Presented to Prof. Fatih Nayebi

The Hidden Cost of Customer Turnover

Pathways to Increased Profits

5x - 25x

Acquiring new customers can be **five to 25 times** more expensive than retaining existing ones

Profit Uplift

5% increase in customer retention can boost profits by 25% to 95%

Project Objectives

01

ML Model

Develop an ML model with reasonable accuracy in predicting customer churn using provided sample data.

02

Insights

Identify key features contributing to churn predictions, providing initial insights into potential risk factors.

03

Potential Impacts

Estimate the potential impact of churn prediction model

Agenda

01

Business Implication & Objectives

02

Team Introduction

03

EDA

04

Preprocessing Steps

05

Feature Selection

06

Model Selection & Comparison

07

Feature Importance

08

Conclusion Q&A

Meet Our Team

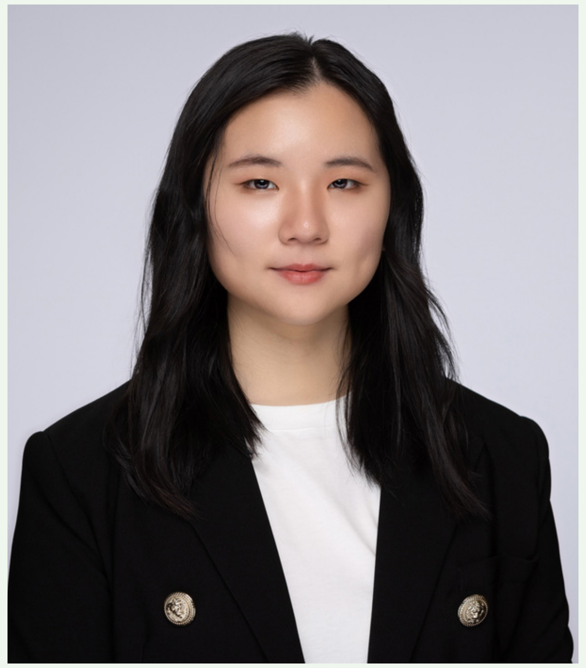
Roles & Contributions

Nandani Yadav

Data Scientist
@YadavNandani

Farah Hoque

Business Analyst
@hoquefarah



Chiara Lu

Data Scientist
@LuChiara

Yichen Yu

Data Scientist
@YichenYU

Meriem Mehri

Data Analyst
@MeriemMehri

David Gao

Data Project Manager
@GaoDavid

Data Sources & Dictionary

X: Predictors

Variable	Description
Customer_Age	Customer's Age in Years
Income_Category	Annual Income Category
Card_Category	Type of Card
Credit_Limit	Credit Limit on the Credit Card

Y

Target Variable

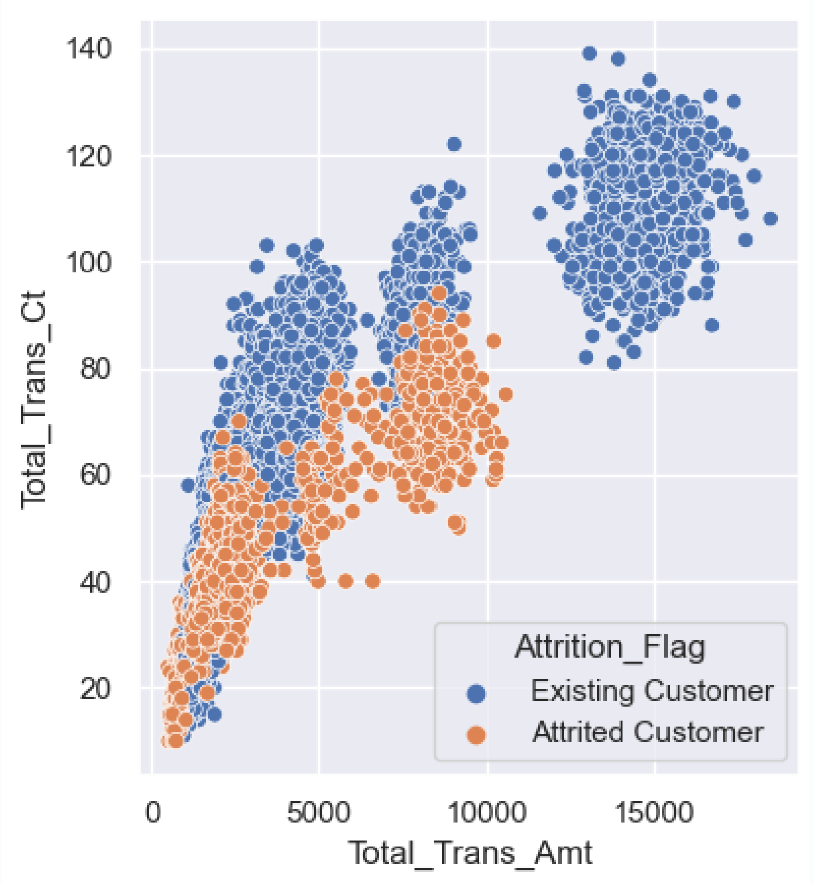
Attrition_Flag

Column Name	Description
CLIENTNUM	Client number ID
Attrition_Flag	Customer activity, Attrited or existing
Customer_Age	Customer's Age in Years
Gender	Customer's gender, male or female
Dependent_count	Number of dependents
Education_Level	Educational Qualification of the account holder
Marital_Status	Married, Single, Divorced, Unknown
Income_Category	Annual Income Category of the account holder
Card_Category	Type of Card (Blue, Silver, Gold, Platinum)
Months_on_book	Period of relationship with bank
Total_Relationship_Count	Total number of products held by the customer
Months_Inactive_12_mon	Number of months inactive in the last 12 months
Contacts_Count_12_mon	Number of Contacts in the last 12 months
Credit_Limit	Credit Limit on the Credit Card
Total_Revolving_Bal	Total Revolving Balance on the Credit Card
Avg_Open_To_Buy	Open to Buy Credit Line (Average of last 12 months)
Total_Amt_Chng_Q4_Q1	Change in Transaction Amount (Q4 over Q1)
Total_Trans_Amt	Total Transaction Amount (Last 12 months)
Total_Trans_Ct	Total Transaction Count (Last 12 months)
Total_Ct_Chng_Q4_Q1	Change in Transaction Count (Q4 over Q1)
Avg_Utilization_Ratio	Average Card Utilization Ratio
Naive_Bayes_Classifier_Attrition_...._mon_1	Naive Bayes
Naive_Bayes_Classifier_Attrition_...._mon_1	Naive Bayes

Source:

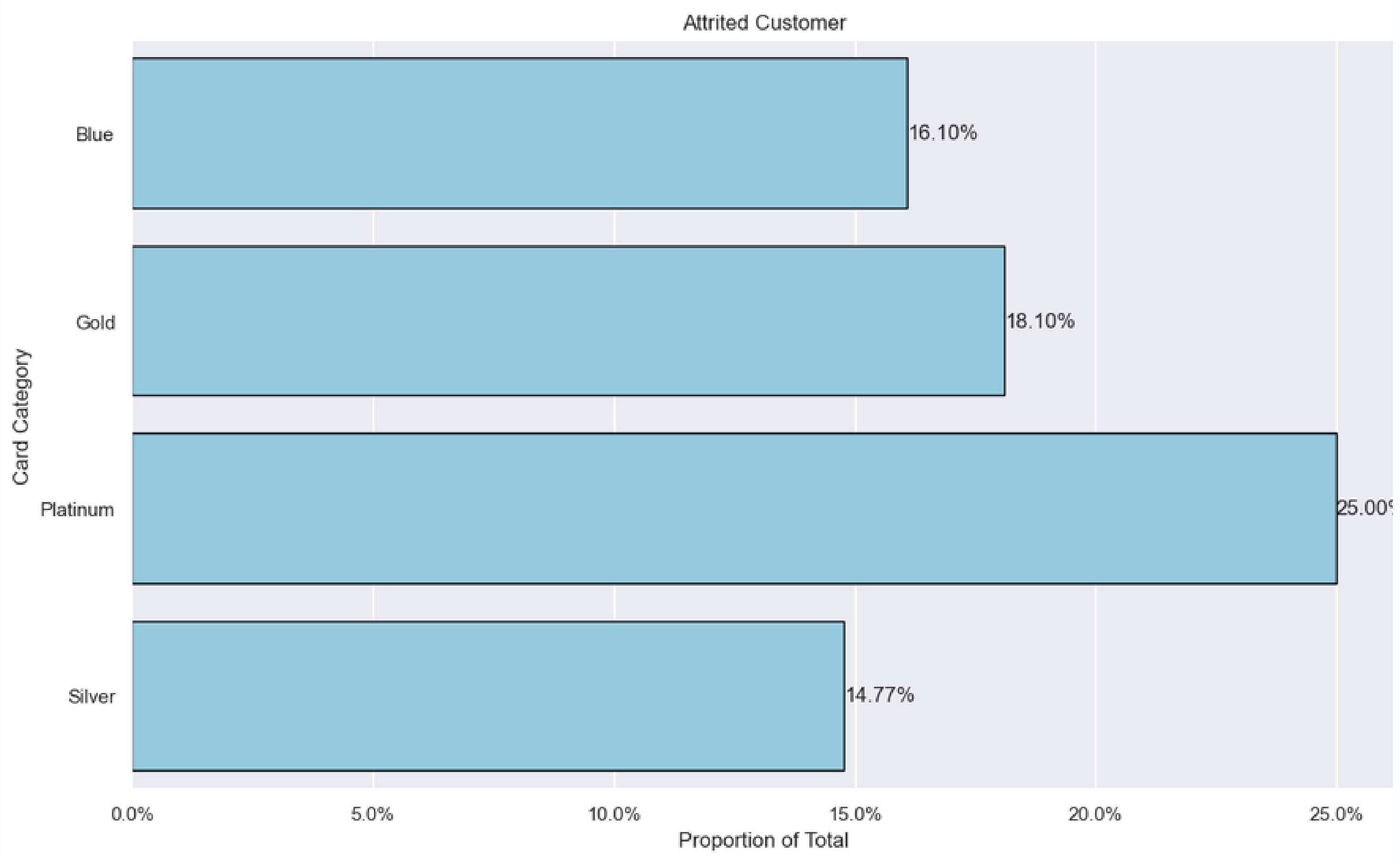
EDA-Churned Customers

Uncovering Insights from Churned Customer Data



Trx Amount

- Higher spenders are less likely to churn.

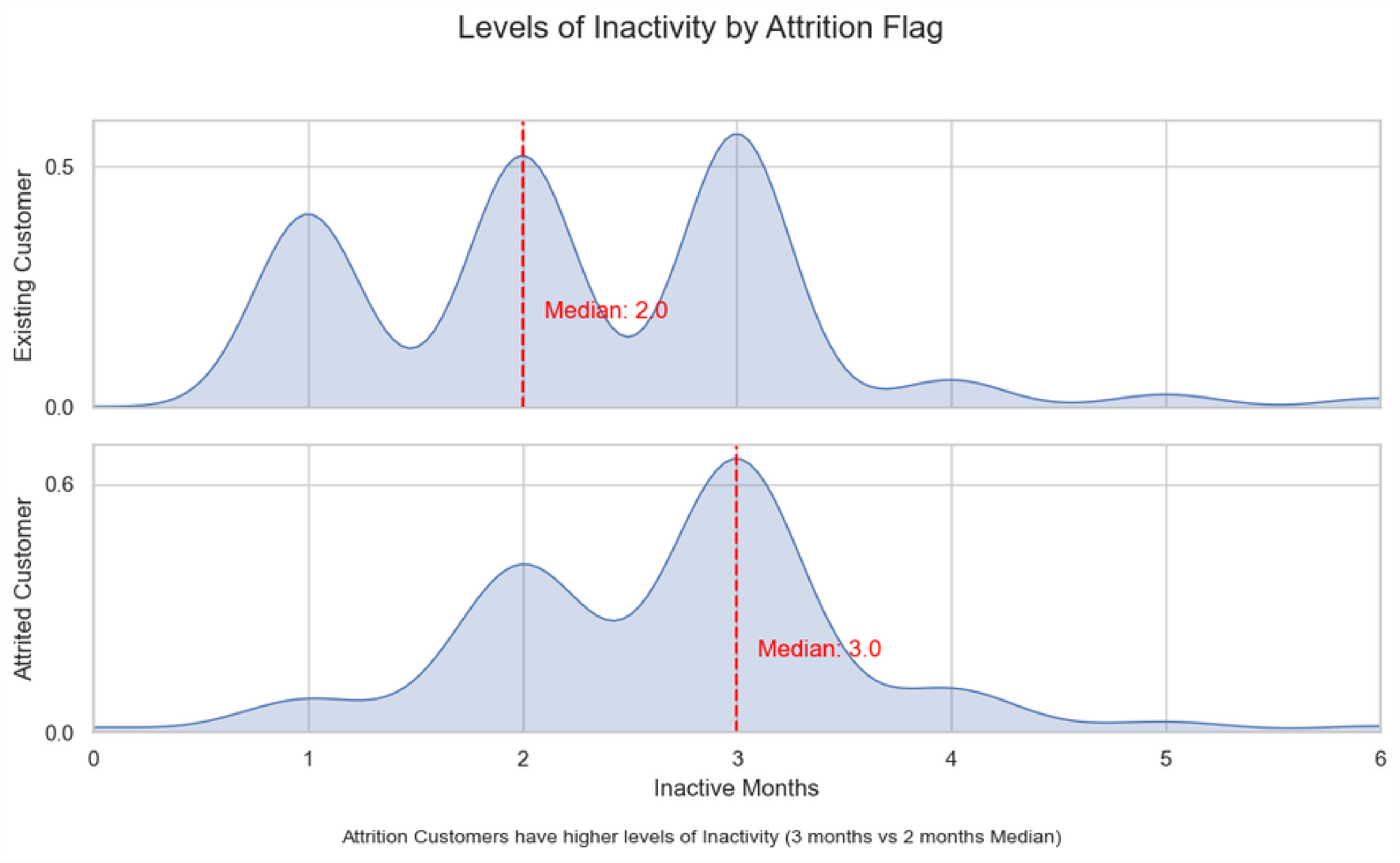


Card Category

- The highest percentage of attrition are coming from platinum and gold card users.

EDA-Churned Customers

Uncovering Insights from Churned Customer Data

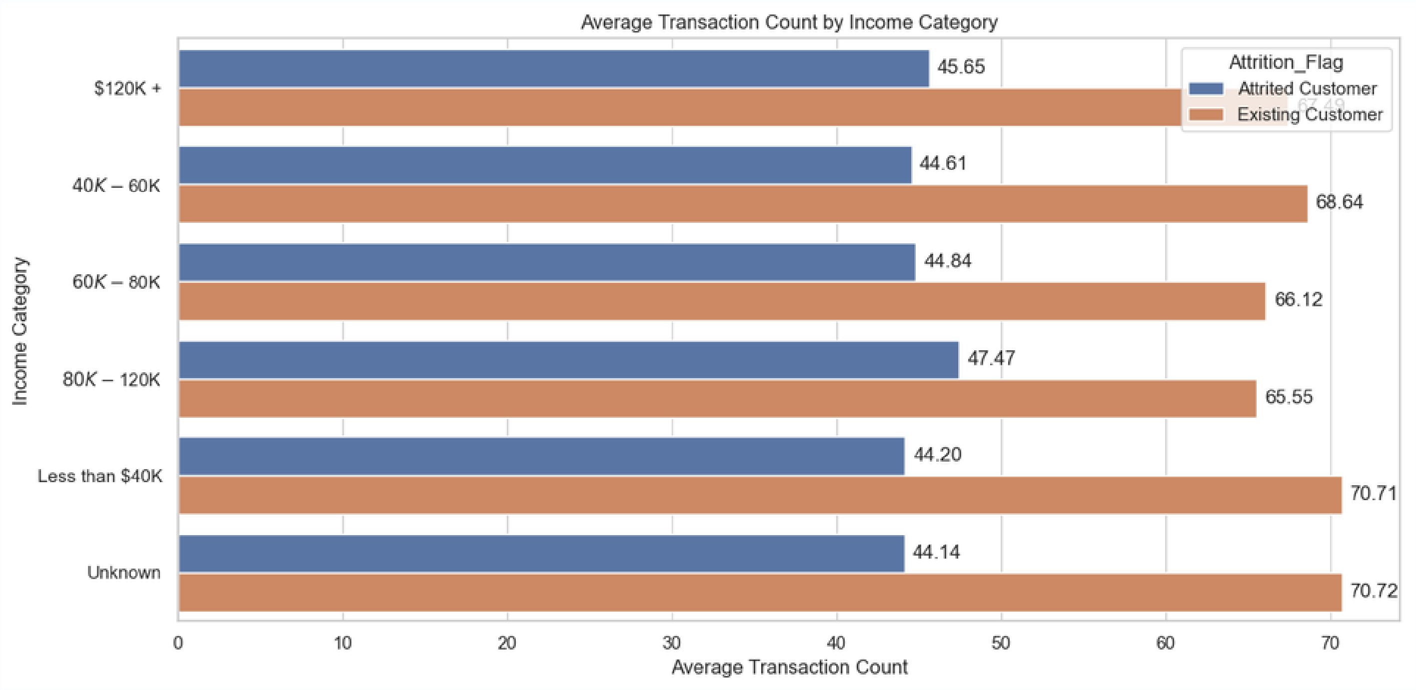


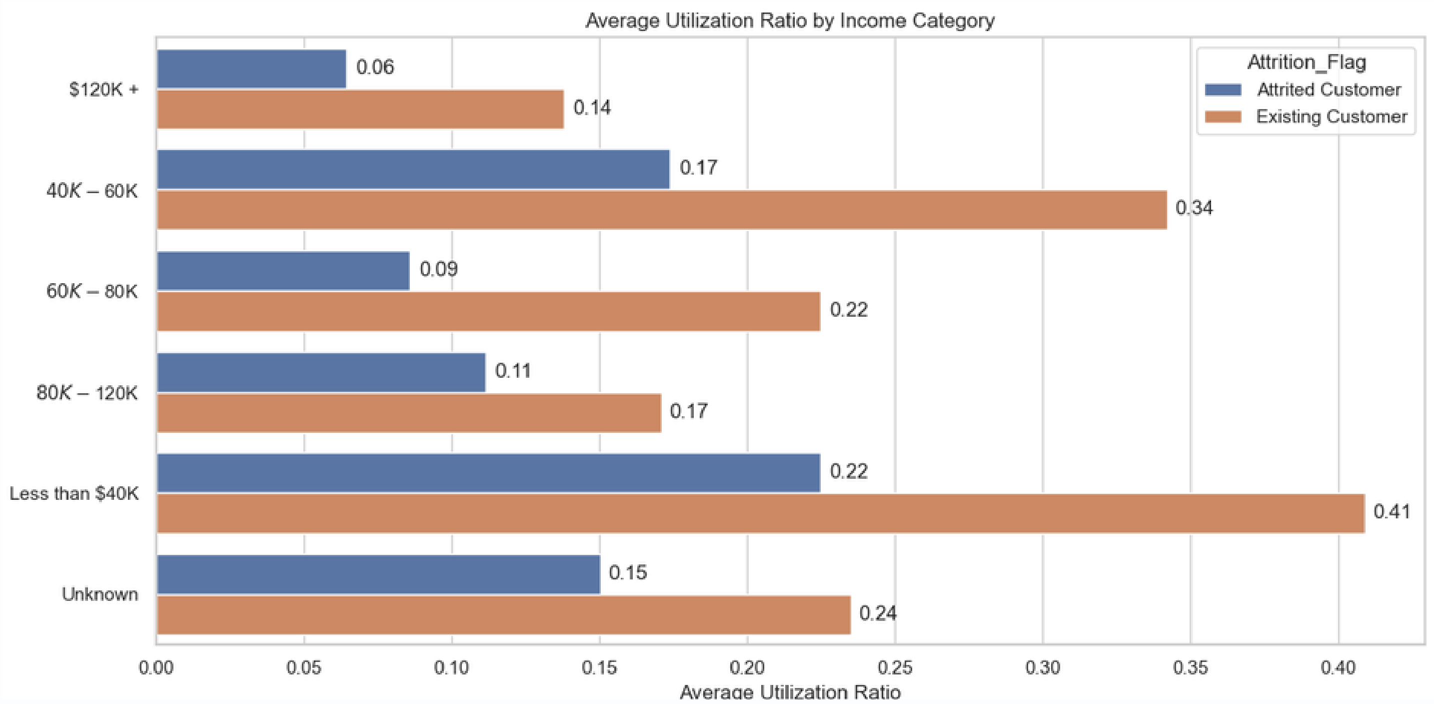
Level of Activity

- When the level of inactivity starts getting “beyond” 2-month threshold, then there is a higher chance that the person will decide to leave the organization.

EDA-Churned Customers

Uncovering Insights from Churned Customer Data





Income Level & Utilization Ratio

The lower income category has:

- **Higher attrition rate**
- **Slightly larger utilization ratio**

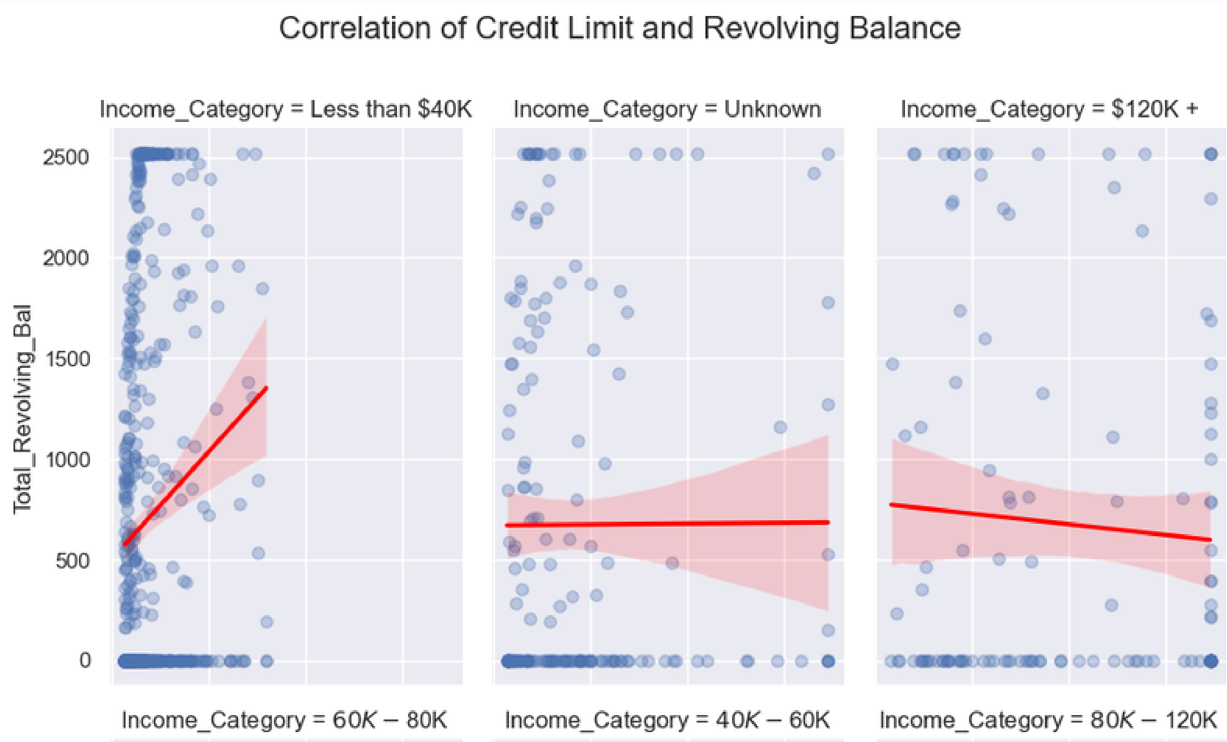
EDA-Churned Customers

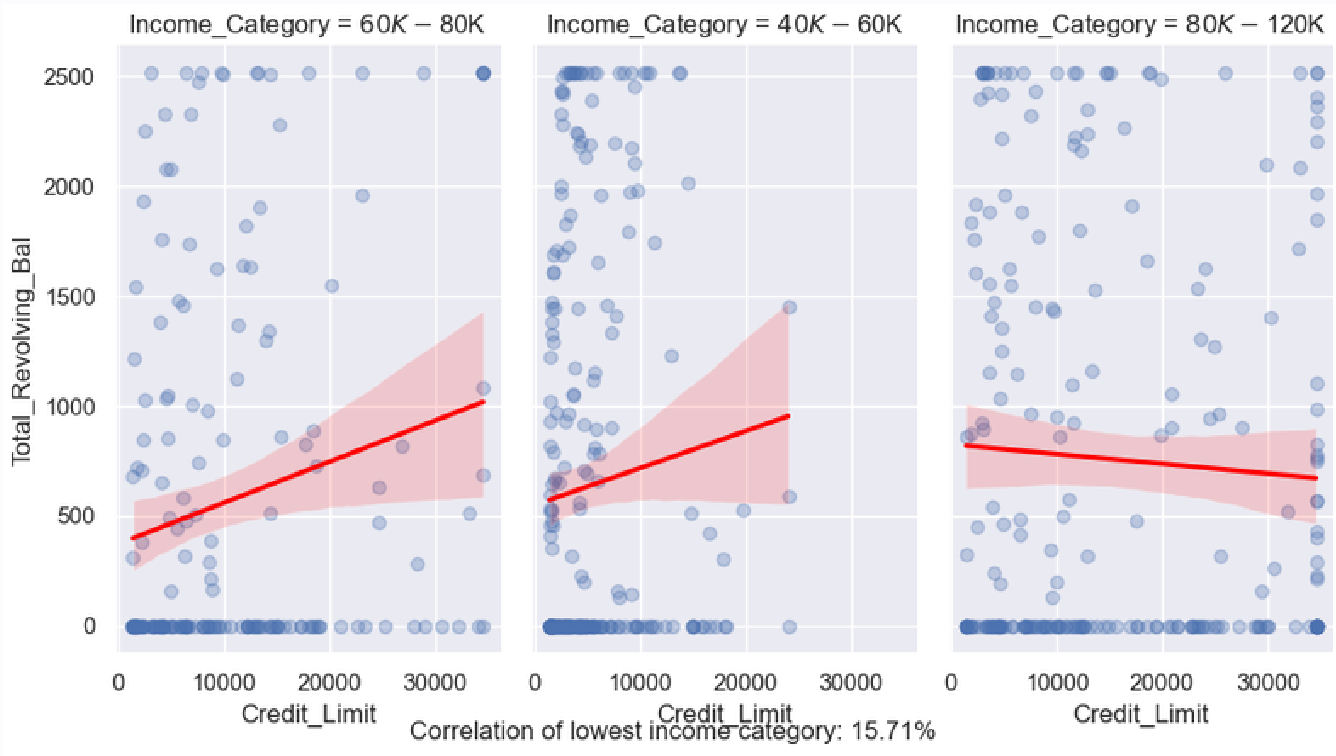
Uncovering Insights from Churned Customer Data

Credit Limit & Revolving Balance

The lower income category has

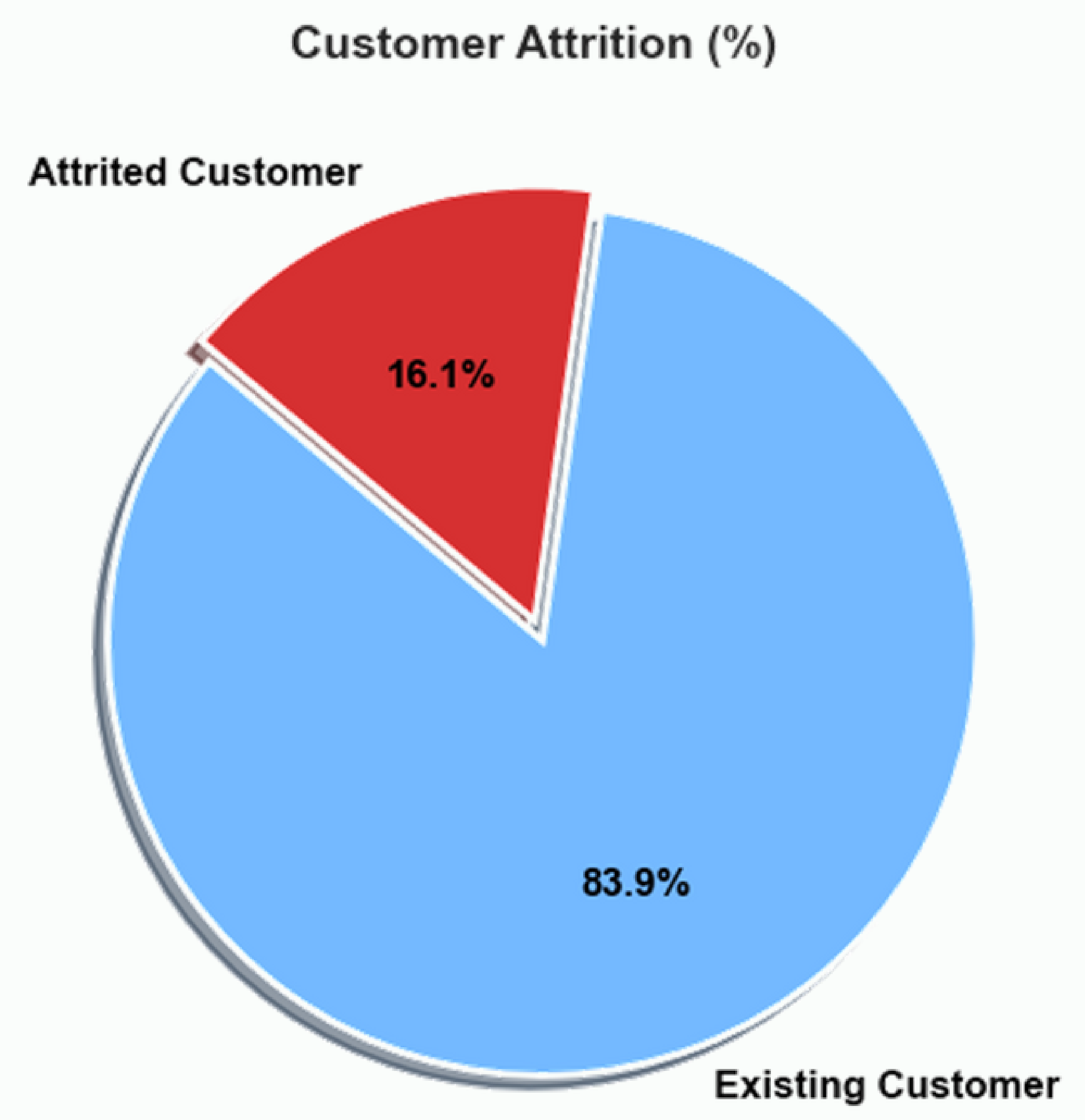
- Low credit limit
- High revolving balance





EDA-Churned Customers

Uncovering Insights from Churned Customer Data



Data Imbalance

- The data is distributed 6 to 1

Preprocessing

Laying the Foundation for Data Analysis

2

Categorical Encoding

- ***Attrition_Flag:*** *Label encoding*
- ***Gender:*** *One-hot encoding*
- ***Education_Level:*** *Frequency encoding*
- ***Marital_Status:*** *Frequency encoding*
- ***Income_Category:*** *Frequency encoding*
- ***Card_Category:*** *One-hot encoding*

1

Train, Test, Validation

Split dataset into train, test, and validation set

3

Oversampling

The dataset is imbalanced - To solve this issue, over sampling was used.

Feature Engineering & Selection

Optimizing Data for Predictive Modeling

Autofeat

Increased features & same accuracy.

Not Implemented

Scaling

Accuracy not improved.

Not Implemented

Implemented

Correlation Analysis

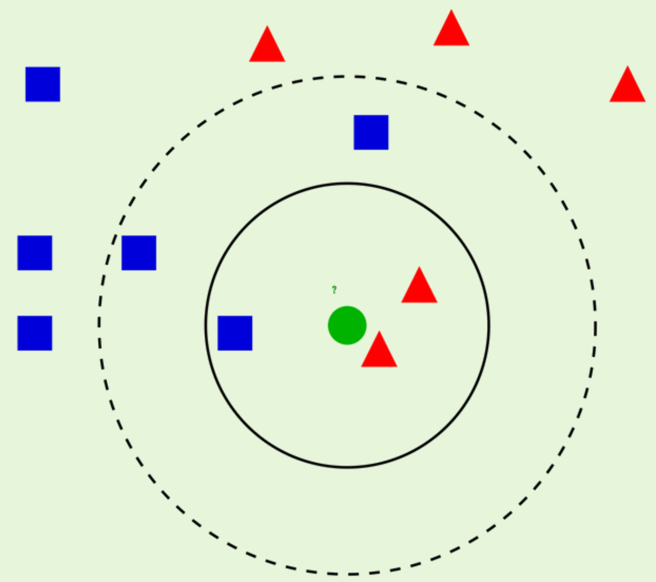
Removed strongly correlated features.

Mutual Information Method

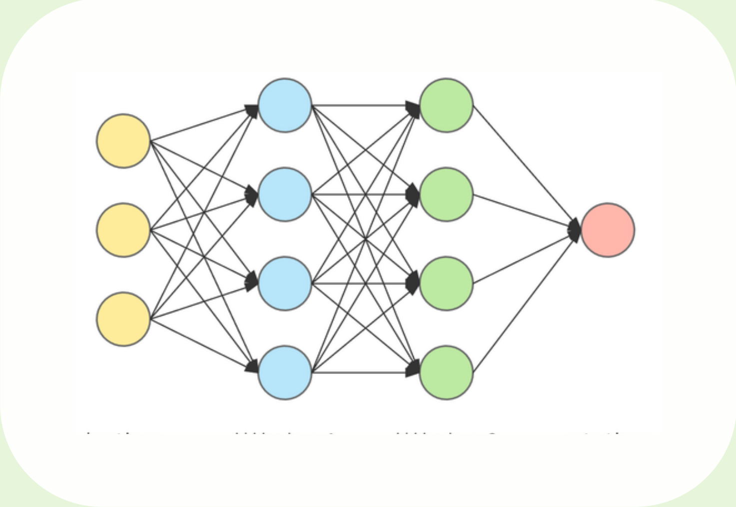
Dropped the least significant features

Model Selection

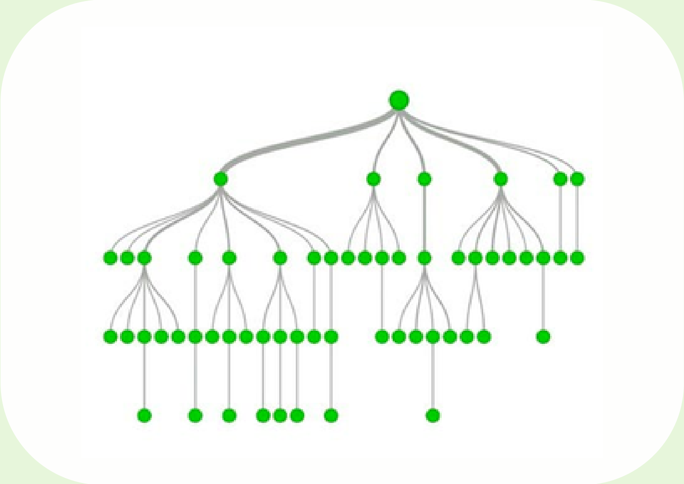
Algorithms for Predictive Success



KNN

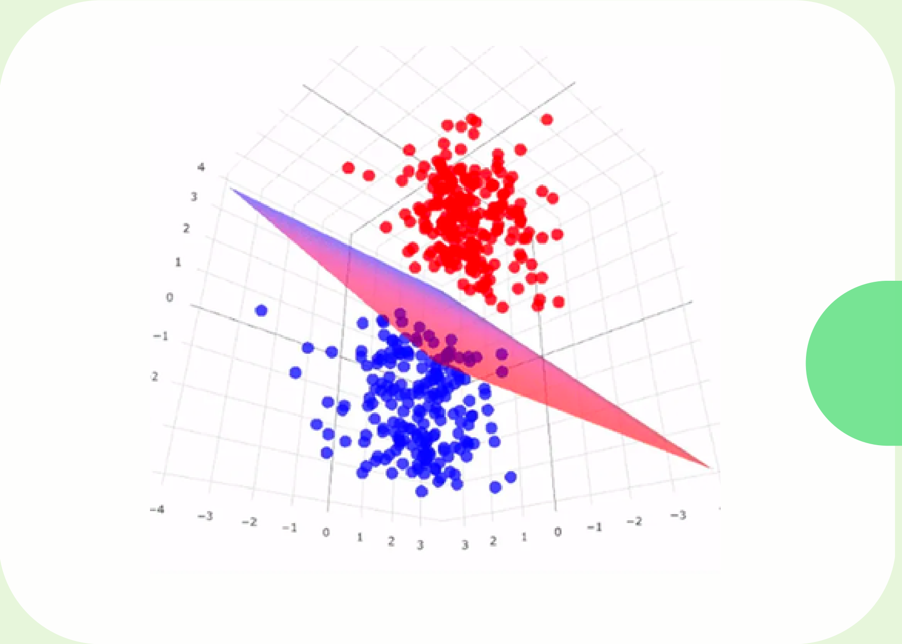


ANN

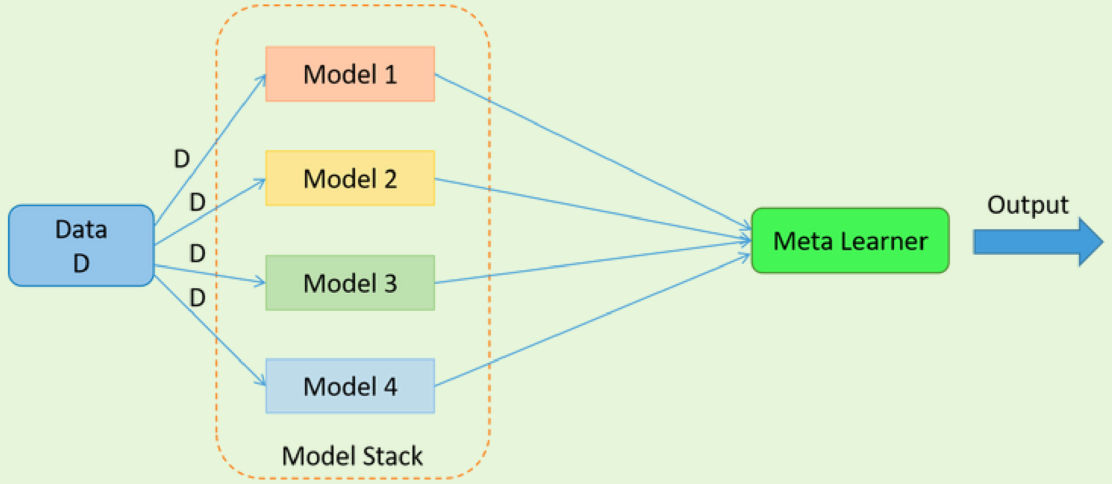


Tree-Based Models

- Decision tree
- Random Forest
- CatBoost



SVM



Stacking

Model Comparison

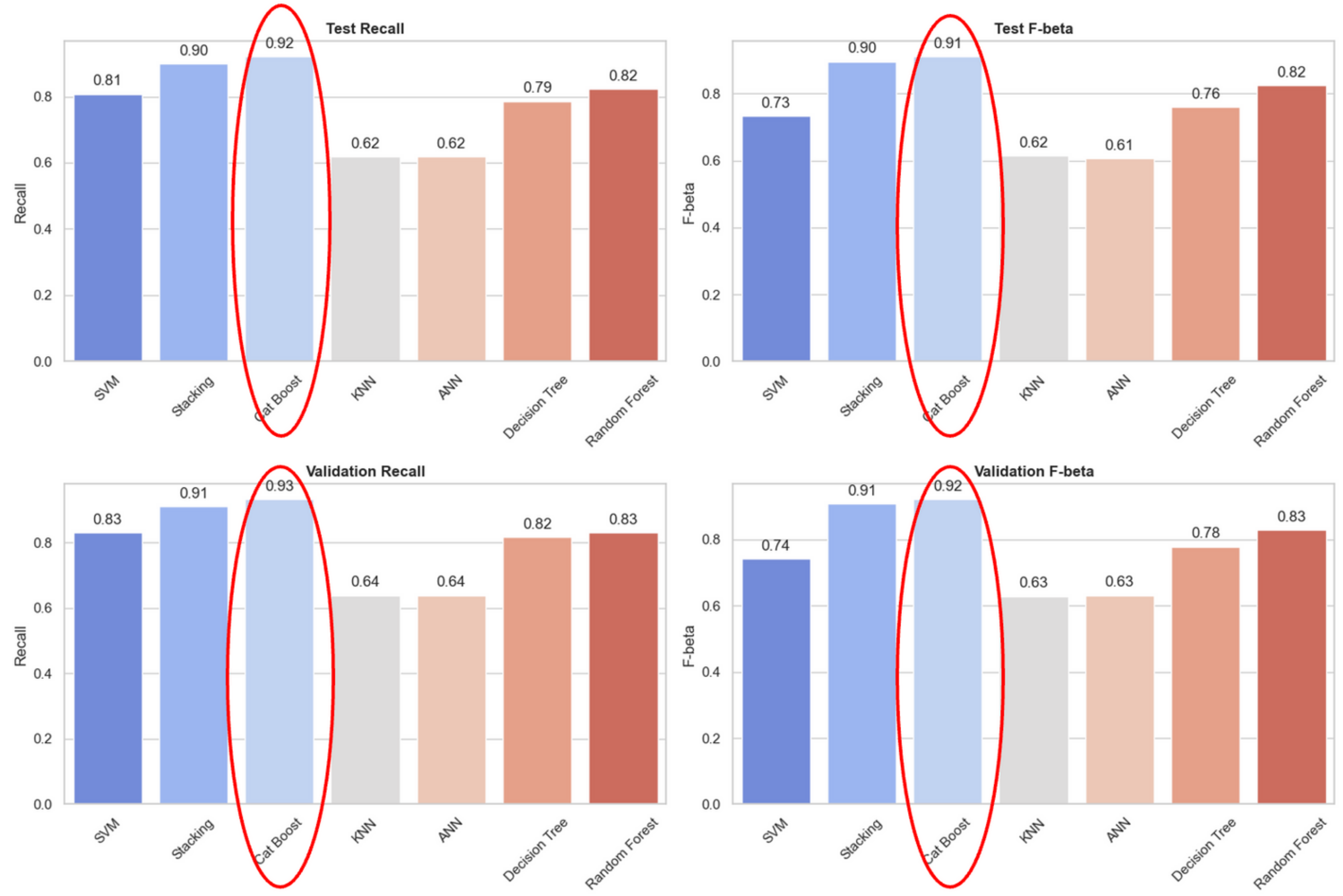
Evaluating Predictive Power Across Different Models

MODEL	Description	Pros	Cons	Validation
KNN	Classifies based on the majority vote of its nearest neighbors.	Simple to implement and understand.	Sensitive to the scale of the data and irrelevant features.	Recall: 63.83% F-beta: 62.87%
ANN	Processing data through interconnected nodes.	Highly flexible and capable of learning complex patterns.	Computationally expensive and may require a lot of data.	Recall: 63.82% F-beta: 63.15%
Tree-Based Method	Splits data into branches to make predictions, forming a tree-like structure.	Reduces overfitting risk and handles unbalanced data well.	Prone to overfitting, especially with many features.	Decision Tree Recall: 81.70% Decision Tree F-beta: 77.86% Random Forest Recall: 82.98% Random Forest F-beta: 82.91%
SVM	Support vector classifier plots a hyperplane to classify observations in a multidimensional space	Robust to overfitting & Memory efficient	Computationally intensive for convex problems	Recall: 80.73% F-beta: 73.29%
Stacking	Ensemble of base models in this list, using Logistic regression as meta learner	Handles complex relationships & combines base models' strengths	Requires tuned base models and can overfit	Recall: 89.91% F-beta: 89.63%
CatBoost	CatBoost is a high-performance, open-source gradient boosting ibrary for decision trees, designed to handle categorical data efficiently.	reduces overfitting with its advanced algorithms, and provides fast and accurate results.	model interpretability can be challenging,	Recall: 93.19% F beta: 92.17%

Performance Comparison

Assessing and Benchmarking Model Effectiveness

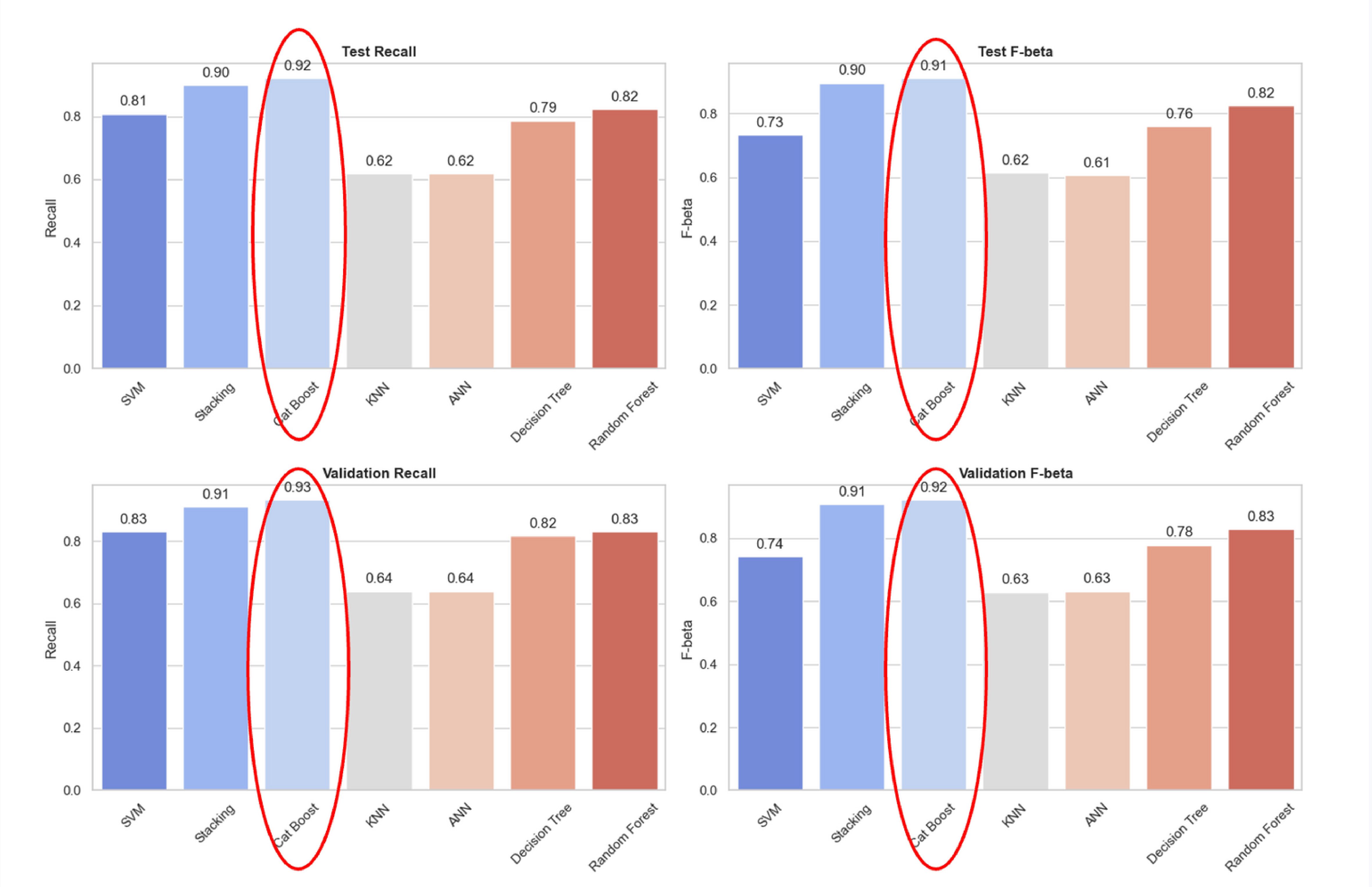
$$F_\beta = (1+\beta^2)\cdot\frac{precision \cdot recall}{(\beta^2 \cdot precision) + recall}$$



- Recall score
- F-beta score

Performance Comparison

Assessing and Be



Hyperparameter Tuning

Approaches to Optimal Model Performance through Hyperparameter Adjustments

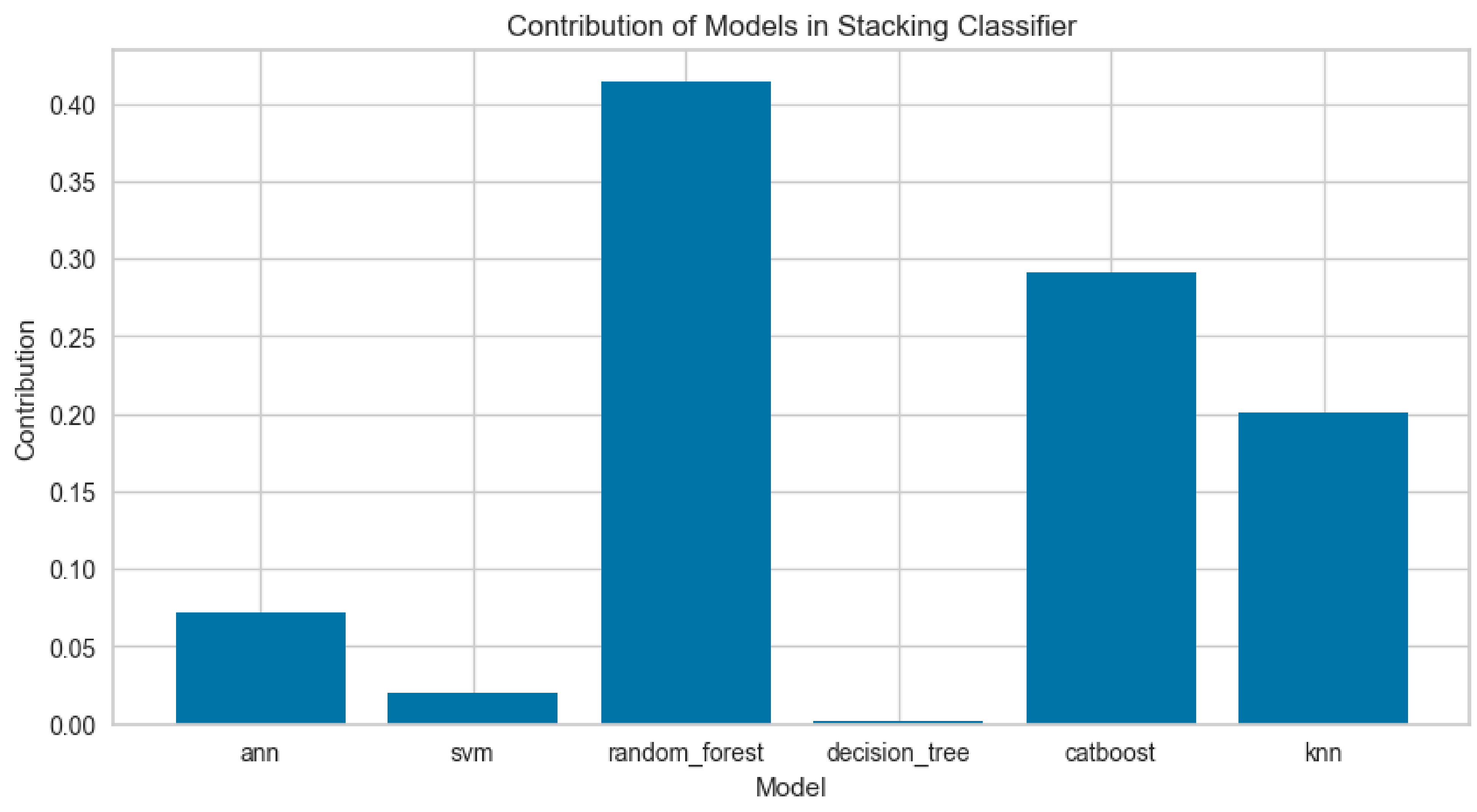
GridSearchCV

&

Optuna

Stacking

Model Contribution & Approaches to Maximizing Predictive Power



The importance of each individual model is shown. To compare importance, we normalize the coefficients, and we obtain the contribution of each model.

- **ANN:** 0.0723
- **SVM:** 0.0199
- **Random Forest:** 0.4143
- **Decision Tree:** 0.0014
- **CatBoost:** 0.2913
- **KNN:** 0.2006

Stacking

Model Contrib

Final Model

The Culmination of Our Predictive Modeling Journey

Education level	Attrition Flag	Predictions	Residuals	Leaf Output
High School **0.05**	1	0	1	0
Graduate **0.05**	0	0	0	
Graduate **0.025**	1	0	1	
High School **0.525**	1	0	1	
Uneducated **0.05**	0	0	0	

Steps for CatBoost:

- Target Encoding : $\text{Ordered Target Encoding} = \frac{\text{OptionCount} + 0.05}{n+1}$
- Residuals : Observed - Predicted
- To build second or more trees :

New Prediction = Prediction + (Learning Rate × Leaf Output)

Education level
0.025
0.05
0.05
0.05
0.525

0.04

0.29

ach

t: 0.4143

0.0014

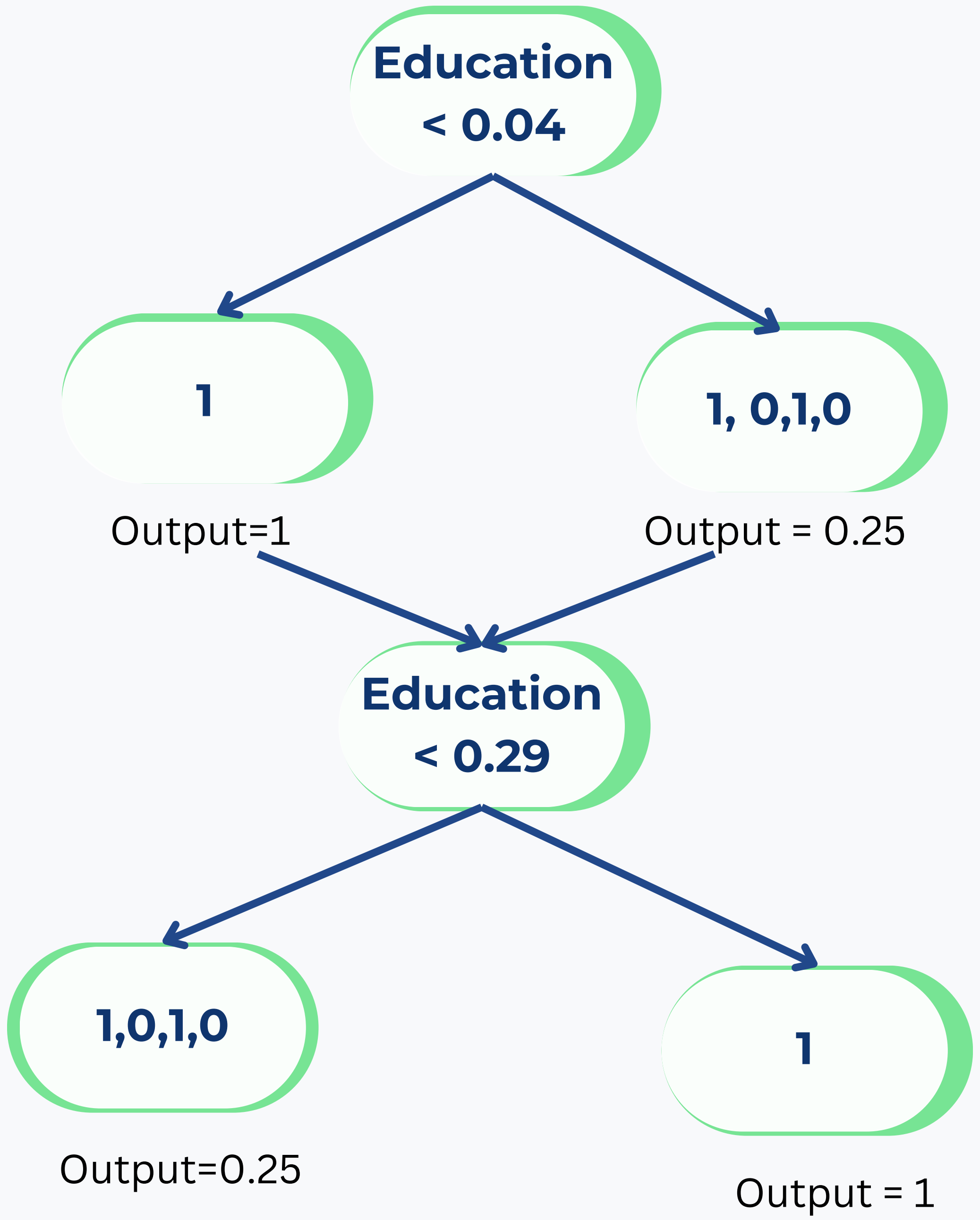
13

Final Model

The Culmination of Our Predictive Modeling Journey

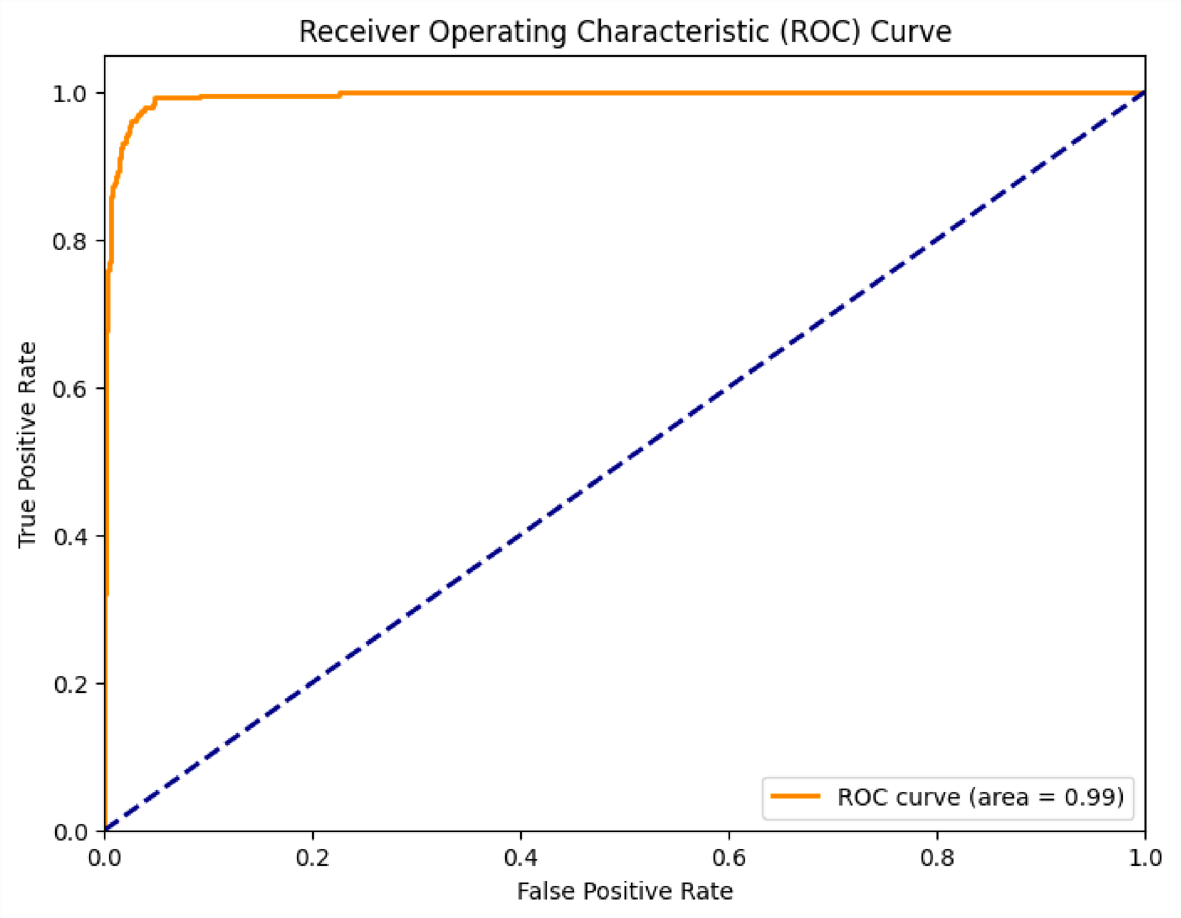
$$\text{Cosine Similarity} = \frac{\sum_{i=1}^{n} A_i B_i}{\sqrt{\sum_{i=1}^{n} A_i^2}\sqrt{\sum_{i=1}^{n} B_i^2}}$$

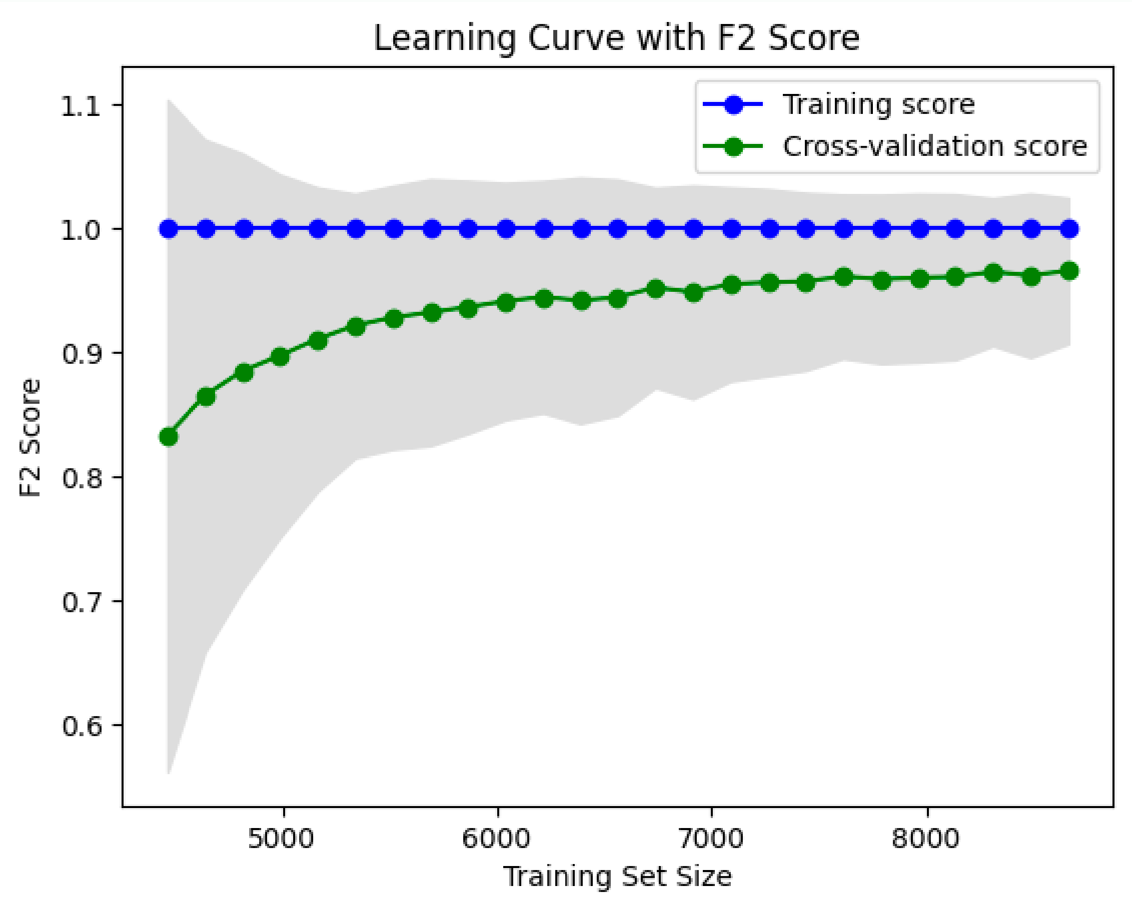
Education level	Attrition Flag	Predictions	Residuals	Leaf Output
High School **0.05**	1	0	1	0
Graduate **0.05**	0	0	0	1
Graduate **0.025**	1	0	1	0
High School **0.525**	1	0	1	0.5
Uneducated **0.05**	0	0	0	1



ROC and Learning Curve

Evaluating Model Performance: Insights from Relevant Curves





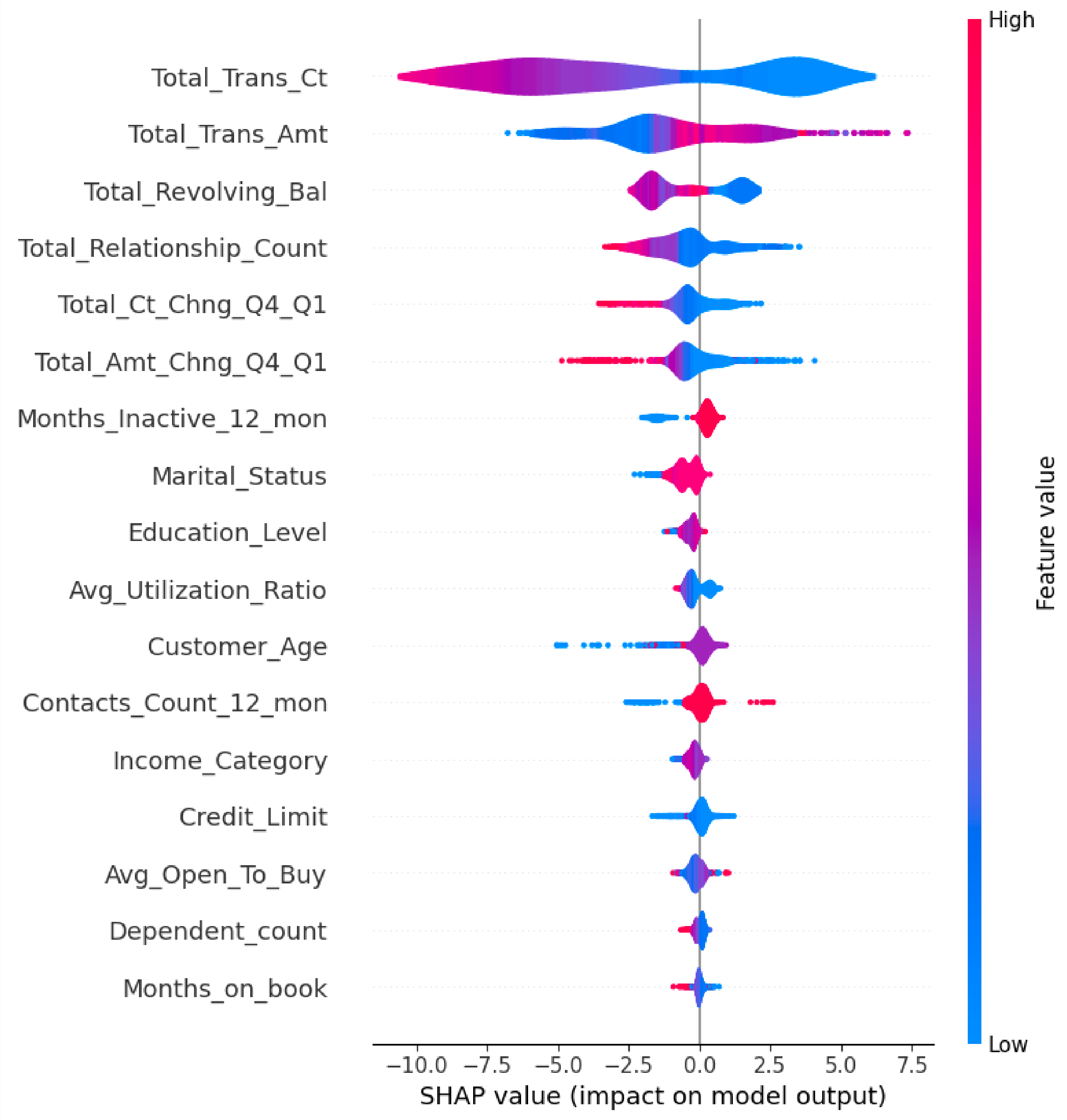
Feature Importance

Deciphering Key Predictors with SHAP Value Analysis

Top 5 important Features

- *Total_Trans_Ct:* *Higher transaction counts (indicated by the pink color) significantly reduce the likelihood of churn.*
- *Total_Trans_Amt:* *higher transaction amounts also contribute to a lower likelihood of churn, with a notable positive impact on customer retention.*
- *Total_Revolving_Bal:* *This feature shows a bimodal distribution, indicating that for some values it increases the likelihood of churn, while for others it decreases it.*
- *Total_Relationship_Count:* *A higher count is associated with a lower likelihood of churn, suggesting that customers engaged with multiple products are less likely to leave.*
- *Months_Inactive_12_mon:* *More months of inactivity strongly suggest an increased risk of churn, as indicated by the SHAP values leaning towards the positive side.*

Decreasing order of Feature Importance

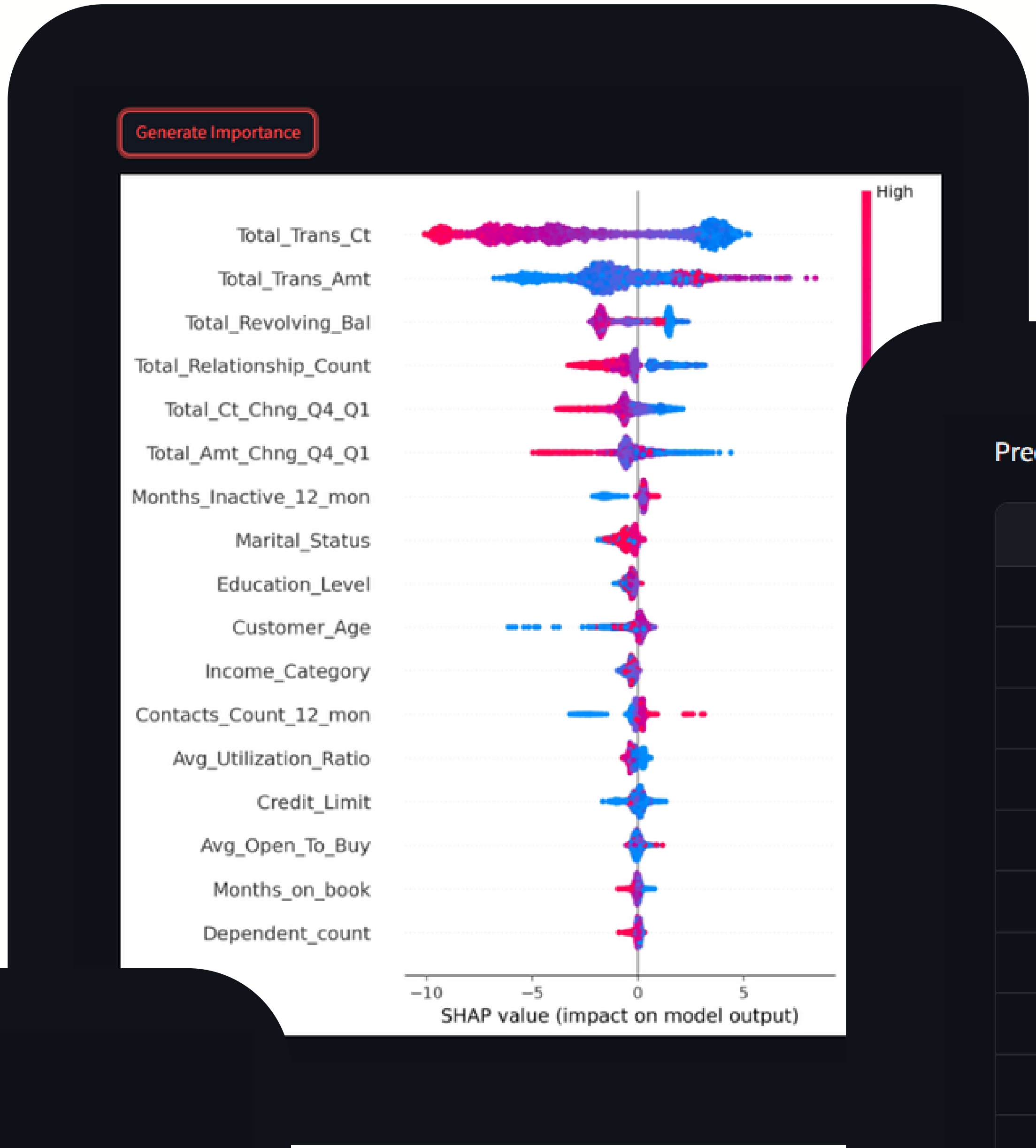


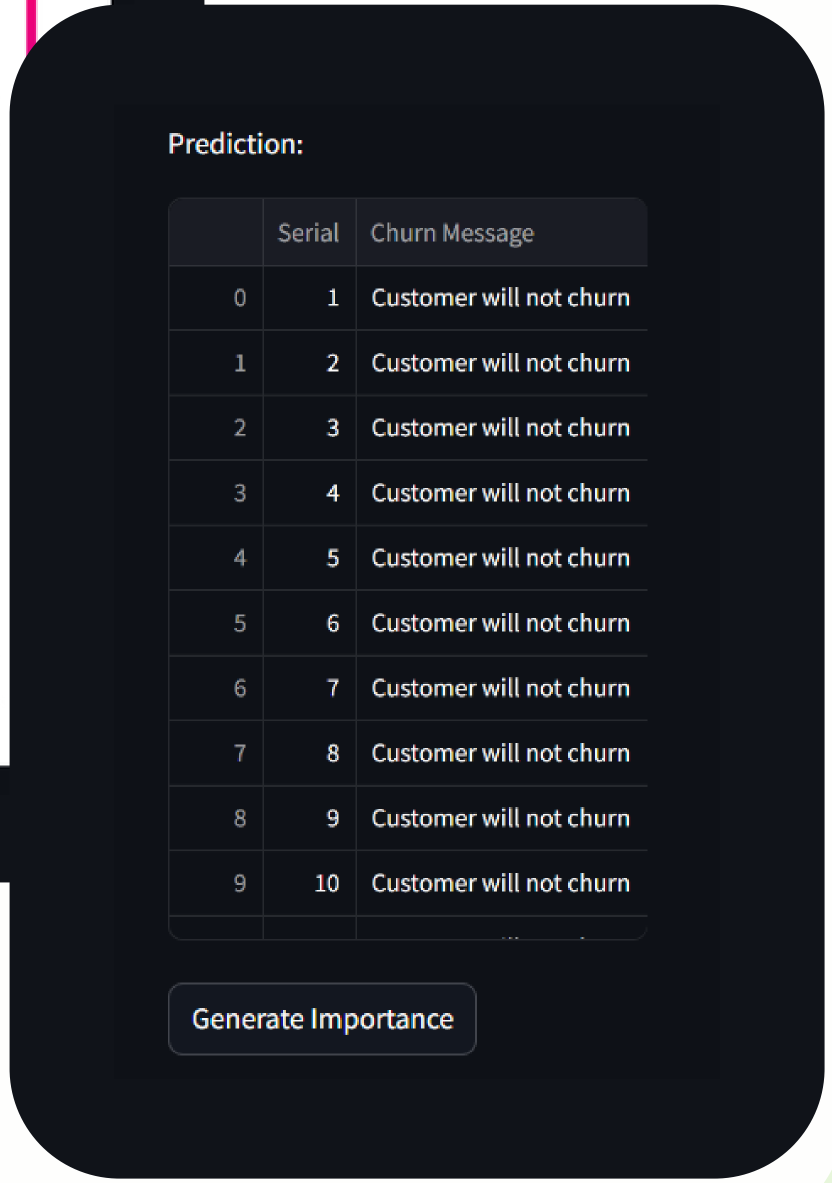
UI Demo

Showcasing Interface Design & Functionality

UI Demo

Showcasing Interface Design & Functionality




Prediction:

	Serial	Churn Message
0	1	Customer will not churn
1	2	Customer will not churn
2	3	Customer will not churn
3	4	Customer will not churn
4	5	Customer will not churn
5	6	Customer will not churn
6	7	Customer will not churn
7	8	Customer will not churn
8	9	Customer will not churn
9	10	Customer will not churn

Generate Importance

Access our repo here!

Resources

seaborn
pandas
SHAP
matplotlib
jupyter
Streamlit
scikit learn

Q&A

Appendices

Stacking Code

```
models = [
    ('ann', MLPClassifier(hidden_layer_sizes=(11,), max_iter=1000, random_state=0)),
    ('svm', best_svm), #{'C': 15, 'gamma': 1e-07, 'kernel': 'rbf'}
    ('random_forest', RandomForestClassifier(max_depth=None, min_samples_leaf=1, min_samples_split=2, n_estimators=300)),
    ('decision_tree', DecisionTreeClassifier(max_depth=10, max_features=None, min_samples_leaf=1, min_samples_split=2)),
    ('catboost', cb.CatBoostClassifier(iterations=1000, learning_rate=0.01, depth=8, verbose = 0)),
    ('knn', KNeighborsClassifier(n_neighbors=2))
]

stacking = StackingClassifier(estimators=models, cv=5)
```

- **The meta learner is Logistic Regression (by default).**