Zhiwei He
Final Report
Data 1030

Credit Card Purchase Prediction

Zhiwei He

Data Science Institute, Brown University

Github Repository: https://github.com/WilliamHE123

Zhiwei He
Final Report
Data 1030

1.Introduction

1.1 Motivation of Problem

This project aims to address the challenge of predicting customer interest in credit card offers by leveraging machine learning techniques. The problem is significant for financial institutions aiming to optimize targeted marketing strategies, enhance customer retention, and drive growth. By accurately identifying customers who are likely to purchase a credit card, companies can deliver more personalized and relevant offers, thereby improving engagement and overall profitability.

1.2 Properties of Dataset

The dataset for this project was collected from Kaggle's "JOB-A-THON by Analytics Vidhya", comprising 245,725 datapoints and 11 features, including demographic (e.g., age, gender) and financial attributes (e.g., account balance). The dataset contains missing values with 11% of Credit Product Availability missing. 76.28% of the target variable belongs to class 0 and 23.72% of the target variable belongs to class 1. This binary classification problem predicts whether a customer is interested (class 1) or not (class 0) in a credit card offer.

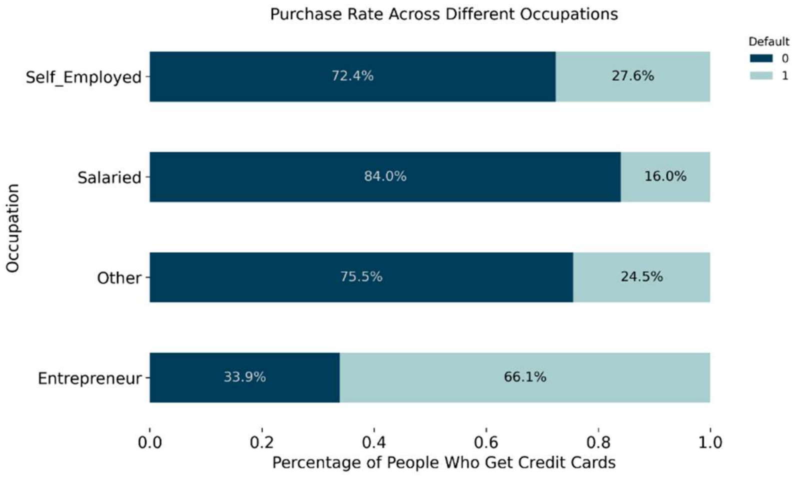
1.3 Previous Research

The data provider of this project approached the same problem by leveraging gradient boosting algorithms, including CatBoost, LightGBM , and XGBoost with pre-determined hypermeters for each model. These models achieved an average ROC-AUC score of 0.85, indicating a relatively high predictive performance in distinguishing the two classes However, our project will utilize precision as the evaluation metric since it measures the accuracy of correctly identifying class 1 predictions, ensuring more effective targeting of interested customers.

2. Exploratory Data Analysis

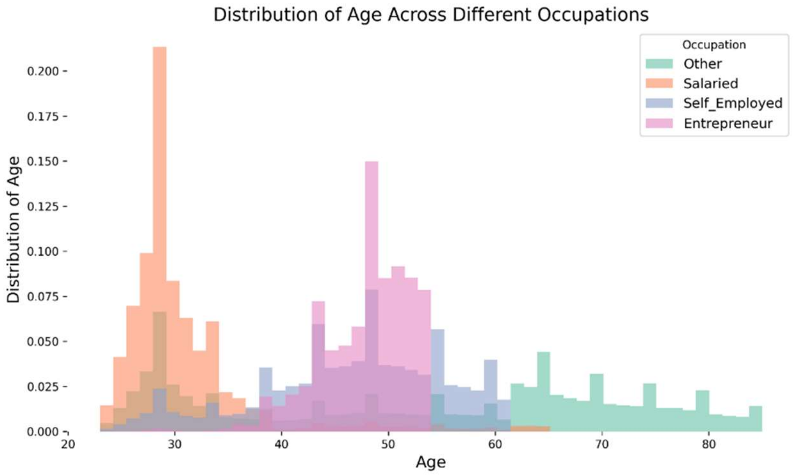
There are 3 interesting finds throughout the EDA process. They focus respectively on: (1). Credit Card Purchase Rates by Occupation, (2). Distribution of Age Across Different Occupations, (3). Distribution of Vintage Across Different Credit Product Availability.

Figure (1): Credit Card Purchase Rates by Occupation



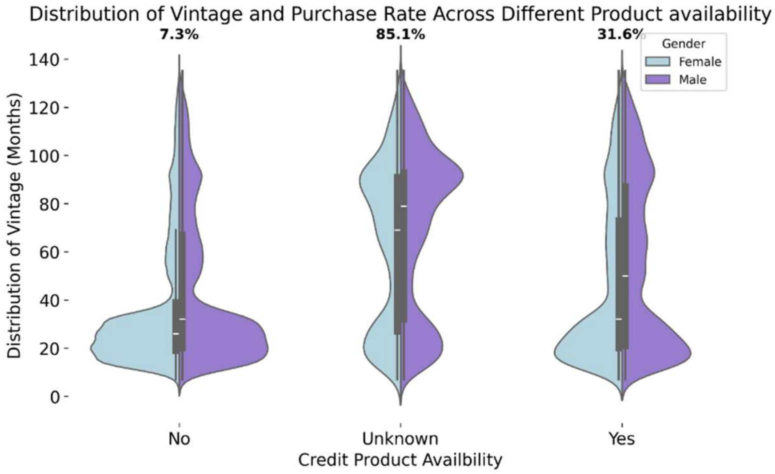
The credit card purchase rates for the occupation of self-employed individuals (27.6%), those in the "Other" category (24.5%), and salaried individuals (16.0%) are relatively close. All of them fall into a similar range and remain low in magnitude. However, there is a notable jump in the purchase rate among entrepreneurs, who exhibit a significantly higher rate of 66.1%. This sharp increase highlights that occupation is a critical categorical feature that can introduce substantial variation in prediction outcomes.

Figure (2): Distribution of Age Across Different Occupations



The age distribution graph reveals that entrepreneurs are predominantly concentrated in the 40 to 60 age range, a demographic typically associated with financial stability, established credit histories, and higher purchasing power. In contrast, salaried individuals and others who exhibited the lowest purchase rate, are more evenly distributed across younger age brackets, potentially indicating less financial need or interest in credit cards.

Figure (3): Distribution of Vintage Across Different Credit Product Availability



The distribution of vintage for customers with "Yes" and "No" credit product availability is similar, with most customers concentrated around 20–40 months. On the other hand, the "Unknown" category exhibits a noticeably broader vintage distribution with significantly longer vintage values exceeding 100 months. Furthermore, we see that “Unknown” category has the highest credit card purchase rate of 85.1 % compared to 7.3% of “No” category and 31.6% of “Yes” category. The sharp difference in purchase rate indicates a potential relationship between vintage and credit product purchase rate.

3.Methods

3.1. Splitting Method

Since the dataset is very large and hyperparameter tuning for different models can be computationally expensive, we initially used 5% of the data to streamline the process. Gradually, the data size was increased to 10% and 20%, which took approximately 8.5 hours to train. For each dataset size, the average performance of different models was recorded, and we evaluate how model performance scales with the size of the dataset. This approach balances computational efficiency and evaluation of model behavior.

The dataset was split into training-validation and testing sets using an 80-20 ratio while maintaining class balance through stratified splitting. This ensures that both sets have the same class distribution of the target variable. Additionally, k-fold cross-validation technique was employed to further validate model performance across different subsets of the data to ensure robustness of the result.

3.2 Data Preprocessing

Before training the model, we will utilize three scalers to preprocess the dataset. In addition, for the 11% of missing values in the "Credit Product Availability" feature, we fill all the missing values with "Unknown."

Table (1): Feature Scaling and Encoding Techniques

	Min-Max Scaling	**Standardize Scaling**	**One-Hot Encoding**
Features	Age, Vintage	Avg_Account_Balance	All categorical Features
Rationale	Defined ranges, minimal outliers.	High account balance leads to outliers	No categorical features have ordinal meanings

3.3 ML Pipeline

The machine learning pipeline is split into four steps:

1. **Splitting**: Split the dataset into 80% Train-Validation and 20% Test
2. **Preprocessing**: fit all three of the scalers using the training dataset, then use the scalers to transform the validation and test set.
3. **Grid Search with Cross-Validation**: Gridsearch was used to identify the optimal parameter for each model under different random state by evaluating combinations of different hyperparameters. Cross-validation ensured that the model was validated on multiple subsets, reducing the risk of overfitting.
4. **Model Evaluation:** The best-performing model from the grid search under each random state was evaluated on the test set to assess its generalization performance.’

Given the dataset's imbalance, accuracy was avoided as it can be misleading by overemphasizing the majority class. Instead, the primary metric chosen is **precision**, as it measures the proportion of correctly predicted positives (class 1) and ensure customers identified as interested in credit cards are genuinely likely to be. By optimizing precision, the bank can effectively target its marketing efforts, reducing the cost and effort wasted on customers who are not genuinely interested.

3.4 Model Parameter Tuning

There are five algorithms Implemented for the task: Random Forest, Support Vector Machine, XGBoost, K-Nearest Neighbors, and Logistic Regression.

Table (2): Model Hyperparameters

Algorithm	Type	Parameter	Value:
Random Forest	Non- Linear	Max_Feature	[0.25, 0.5,0.75,1.0]
		Max_Depth	[1, 3, 5, 10, 30, 100]
Support Vector Machine	Non-Linear	C	[1e-2, 1e-1, 1e0, 1e1, 1e2]
		Gamma	[1e-3, 1e-2, 1e-1, 1e0, 1e1, 1e2]
XGBoost • Eearly Stopping = 100 • N_Estimator = 10000 • Subsample = 0.66 • colsample_bytree = 0.9 • Learning Rate = 0.1	Non-Linear	Reg_Lambda	[0e0,1e-1, 1e0, 1e1, 1e2]
		Reg_Alpha	[0e0,1e-1, 1e0, 1e1, 1e2]
		Max_Depth	[1,3,5,10,30,100]
KNN	Non-Linear	N_Neighbor	[3, 5, 7]
		Metric	['euclidean', 'manhattan']
Logistic Regression	Linear	C	[0.01, 0.1, 1, 10,100]
		Penalty	['l1', 'l2']

3.5 Model Uncertainty

The model's performance uncertainty comes from two sources. First, **grid search cross-validation** under five random states introduces variations due to different data splits. Second, **non-deterministic models** like Random Forest and XGBoost add randomness during training (e.g., sampling data or features). This combination of random data splitting and model variability helps capture a more robust estimate of each model's performance.

4.Result

4.1 Baseline and Performance of Models

The baseline precision for the dataset is **0.2372**. All five machine learning models significantly outperform this baseline precision score by being at least 3 standard deviations above the precision baseline. **Random Forest** consistently ranks as the highest-performing algorithm across all three dataset sizes (5%, 10%, and 20%). Conversely, KNN consistently exhibits the lowest precision scores, likely due to its sensitivity to imbalance compared to other algorithms. As the dataset size increases, the standard deviation of the test precision scores decreases for all models, which reflects the robustness of the models as the dataset size grows.

Table (3): Model Performance Across Different Dataset Sizes

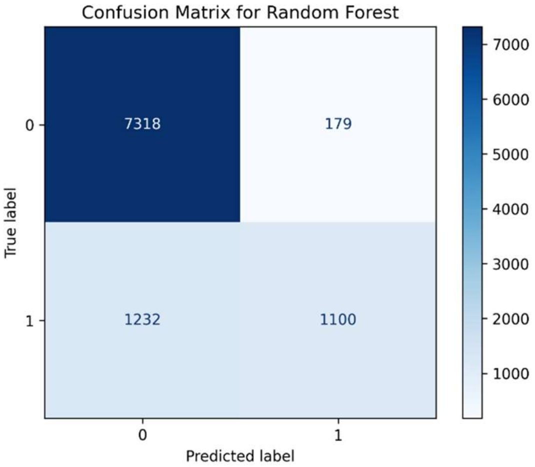
5% Data Performance					
	Random Forest	Logistic Regression	SVM	XGBoost	KNN
Precision:	**0.8503**	0.8439	0.8326	0.8279	0.7011
Std of Precision:	0.0192	0.0171	0.0256	0.0261	0.016
# of std Above Baseline	31.93	35.48	23.26	22.63	28.99
10 % Data Performance					
	Random Forest	Logistic Regression	SVM	XGBoost	KNN
Precision:	**0.8597**	0.8527	0.8496	0.8491	0.7290
Std of Precision:	0.0094	0.0113	0.0088	0.0089	0.0128
# of std Above Baseline	66.22	54.47	69.59	68.75	38.42
20 % Data Performance					
	Random Forest	Logistic Regression	SVM	XGBoost	KNN
Precision:	**0.8549**	0.8331	0.8480	0.8392	0.7174
Std of Precision:	0.00526	0.00394	0.0047	0.0104	0.0076
# of std Above Baseline	117.43	151.24	129.96	57.88	63.18

Table (4): Best Hyperparameters For Each Model

	Parameter Grid
Random Forest	'max_depth': 5,’max_features': 0.5
Logistic Regression	‘C’: 0.01, ‘penalty': 'l2'
SVM	'svc__C': 0.1, 'svc__gamma': 0.1
XGBoost	'max_depth': 1, 'reg_alpha': 100.0, 'reg_lambda': 10.0,
KNN	metric': 'manhattan', n_neighbors': 7

The final model is Random Forest with max_depth of 5 and max_feature. Although both precision and accuracy are above 85%, the recall is relatively low at approximately 47%. This indicates that while the model is effective at correctly identifying positive cases, it misses a significant portion of actual positive cases.

Figure (4): Confusion Matrix Of Random Forest Test Prediction



4.2 Global Feature Inportance

To analyze global feature importance, three different methods were used: **permutation importance**, **SHAP values**, and **Gini impurity-based feature importance**. Permutation importance measures the drop in model precision when a specific feature's values are shuffled. SHAP values quantify the average contribution of each feature to the model's predictions. Gini impurity evaluates the reduction in impurity achieved by splitting on a feature across all trees.

In all three of the analysis graphs, **Credit_Product** is considered as the most important feature, regardless of the categories of "No," "Unknown," and "Yes". The second and third most important features were **Vintage** and **Age**. Feature Credit_Product, Vintage, and Age consistently are ranked as the three most important features across all methods, reflecting their significance of predictive power.

Figure (5): Feature Importance by Permutation

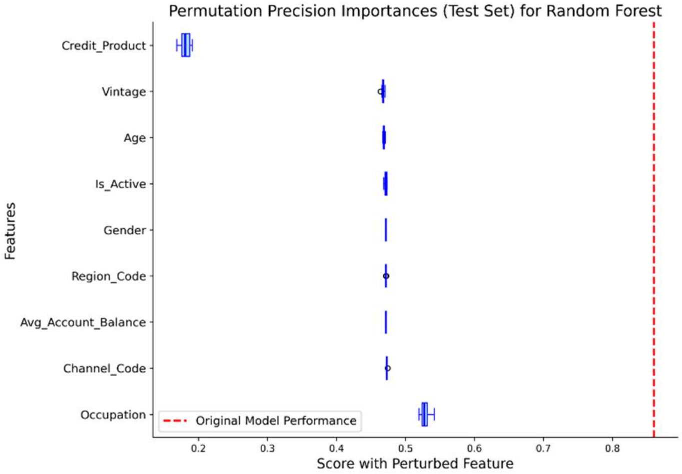


Figure (6): Feature Importance By Gini Impurity

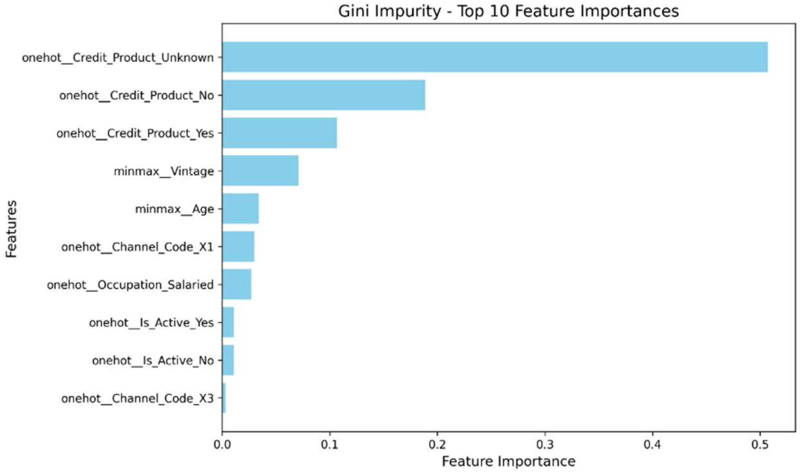
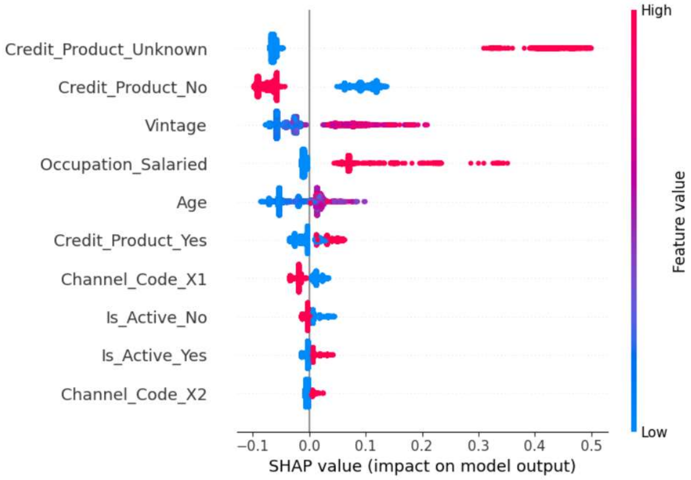


Figure (7): Feature Importance by SHAP



4.4 Local Feature Importance:

SHAP values were calculated for both random class 0 and class 1 predictions, allowing for a detailed analysis of the features' contributions to the predictions of each class. For class 0 predictions, the five least important features all were regional codes, like class 1 predictions where the least impactful features were again all Region Code value. On the other hand, for both class 0 and class 1 predictions, Credit product availability, vintage, and age are ranked as the most 5 important features.

Figure (8): Least and Most Impactful Features for A Class 0 Prediction

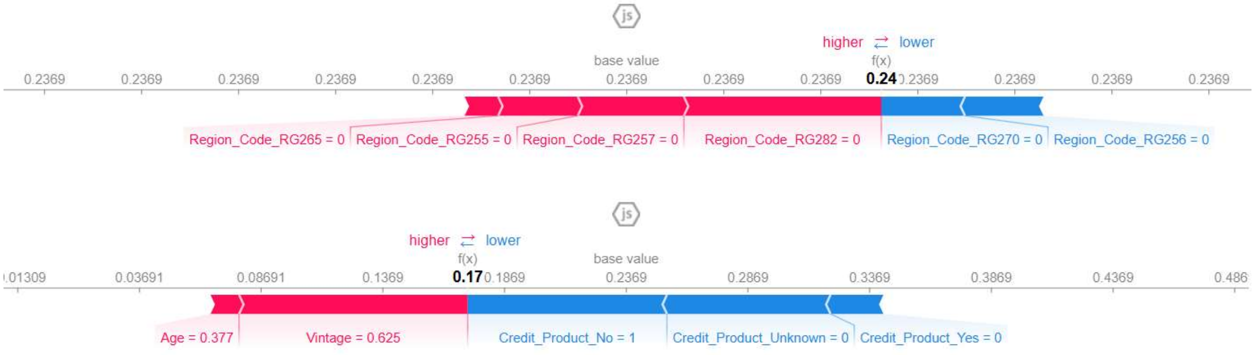
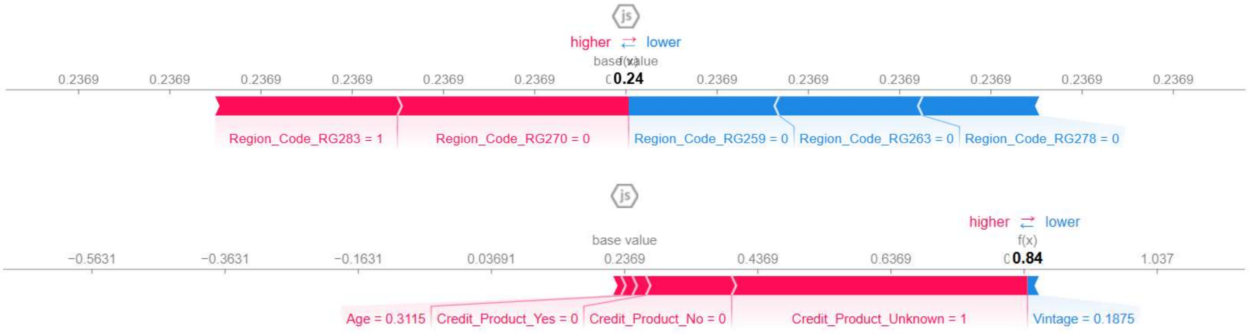


Figure (9): Least and Most Impactful Features for A Class 1 Prediction



The local feature importance results align closely with the global feature importance finding, highlighting the consistency in identifying Credit_Product, Vintage, and Age as the most critical features for predicting credit card interest. One unexpected observation is that Occupation, which appeared significant during the EDA due to its apparent influence on purchase rates (e.g., the high rate among entrepreneurs), is not ranked as one of the most impactful features in the final model. This suggests that its influence on purchase rates might come from the correlation with other significant features such as Age.

5. Outlook

5.1 Weakness of Model

One limitation of the model is that it was trained and evaluated using only 20% of the dataset, which, although computationally efficient, may limit its full potential to generalize on unseen data. Additionally, the model's recall is relatively low, indicating that it fails to identify a significant portion of actual positive cases, which could impact its effectiveness in targeting customers interested in credit card offers.

5.2 Improvement

There are three methods to improve the model: (1). weight reassigning to enhance recall by assigning higher importance to class 1 predictions to improve recall score, (2): feature engineering using PCA to focus on the most predictive features and remove redundancy, and (3). model stacking to leverage the strengths of multiple models for improved generalization. Additionally, using a stronger GPU would allow training on larger datasets, enhancing performance. Collecting additional data on customer behavior trends or spending patterns could provide more granular insights and improve predictive accuracy.

Reference:

1. Github Repository: https://github.com/WilliamHE123
2. Kaggle Dataset: https://www.kaggle.com/datasets/swastikmohanty845/jobathon-may-2021-credit-card-lead-prediction/data
3. Previous Research: https://www.kaggle.com/datasets/swastikmohanty845/jobathon-may-2021-credit-card-lead-prediction/data
4. "Feature Importance with Random Forests", https://www.geeksforgeeks.org/feature-importance-with-random-forests/ , 05 Apr 2024