Predicting Credit Card Approval with Machine Learning Algorithms

Ruben Solis

February 2023

Contents

List of Figures

1 Introduction

Credit card approval is a topic that affects everyone. From banks to customers, being approved for a credit card has impacted borrowers and lenders alike.

1.1 Objective:

In this paper, we will use machine learning methods to develop a prediction model to determine whether a client is approved or disapproved. We will also explore to see which features play a role in determining this decision Financial institutions will find much use of this model. For instance, banks receive a lot of applications for the issuance of credit cards. Many were rejected for high-loan balances, low-income levels, or too many inquiries on an individual's credit report. Manually analyzing these applications is error-prone and a time-consuming process. This task can be automated with machine learning, and almost every bank does so nowadays. The criteria for success will be defined as being able to identify at least two or three factors that would determine whether a client should be issued a credit card or not.

1.2 Significance:

The bank or financial institution issuing the credit card will be able to deploy the model to determine which clients should be approved for the credit card and reduces the number of delinquent debts from approved applicants.

2 Data

2.1 Data Source:

The data used for this project may be found at the following link:
`https://archive.ics.uci.edu/ml/datasets/Credit+Approval`

2.2 Data Decription:

Since the data set contains confidential information, the features have been anatomized. The author of this blog article. has given us a very good indication of what the features very well may be. In this case, we see that the features are male, `age`, `debt`, `married`, `bank customer`, `education level`, `ethnicity`, `years employed`, `prior default`, `employed`, `credit score`, `drivers license`, `citizen`, `zip code`, `income`, and `approval status`.

2.3 Wrangling and Cleaning:

The purpose of data wrangling and cleaning are:

- To ensure that all features are of the correct data type.
- To ensure that missing values are properly dealth with (imputation).
- To prepare the data set for exploratory data analysis and statistical analysis.

Our data set contains both numeric and non-numeric data (specifically data that are of float64, int64 and object types). Specifically, the features `debt`, `years_employed`, `credit_score` and `income` contain numeric values (of types `float64`, `int64`) and all the other features contain non-numeric values (of type `object`). The data set also contains values from several ranges. Some features have a value range of 0-28, some have a range of 2-67, and some have a range of 1017-100000. Finally, the data set had missing values. The missing values in the data set are labeled with "`?`", which can be seen in the last cell's output.

The correct columns and types were converted as follows:

Column Name	Data Type
male	object
age	float64
debt	float64
married	object
bank_customer	object
education_level	object
ethnicity	object
years_employed	float64
prior_default	object
employed	object
credit_score	int64
drivers_license	object
citizen	object
zip_code	object
income	float64
approval_status	int64

2.4 Incorrect Value and Missing Value Imputation

The data were relatively clean aside from a few missing values here and there. There were some incorrect place holders such as question marks that were replace with Numpy's `np.nan` to denote that we had missing values. Moreover, for `approval_status` the values in the column were initially + and - for approved and disapprove respectfully. We recoded these to be 1 and 0 for our machine learning models later. Here is how the missing values for the features were handled.

2.4.1 Numeric Features

For the numeric features, we filled the missing values with the mean of the column.

2.4.2 Categorical Features

For categorical features, we imputed missing values with the mode of the data.

3 Data Visualization and Analysis

3.1 Approval Status

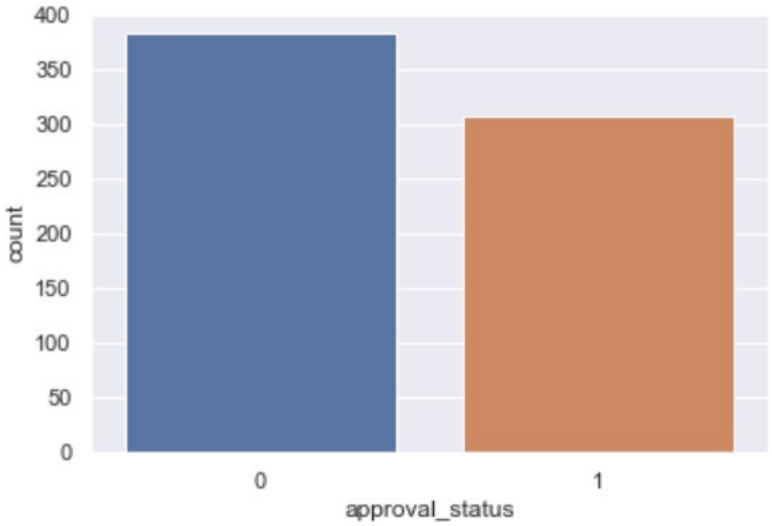

Figure 1: Approval and Disapproval Counts for Applicants

When we look at figure 1, we see that more applicants are not getting approval compared to those that are. While at the moment we are not sure why, we will investigate the possible factors that played a role.

Let's now turn our attention to viewing the applicants based on gender.

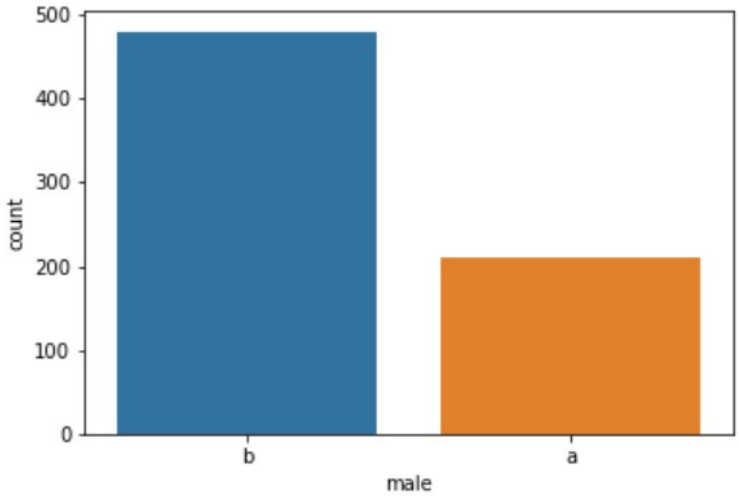

Figure 2: Application Count by Gender

Due to the Anonymity of the data, we could not definitively say whether variable `a` or `b` is male (or female), but one group did apply for more credit cards

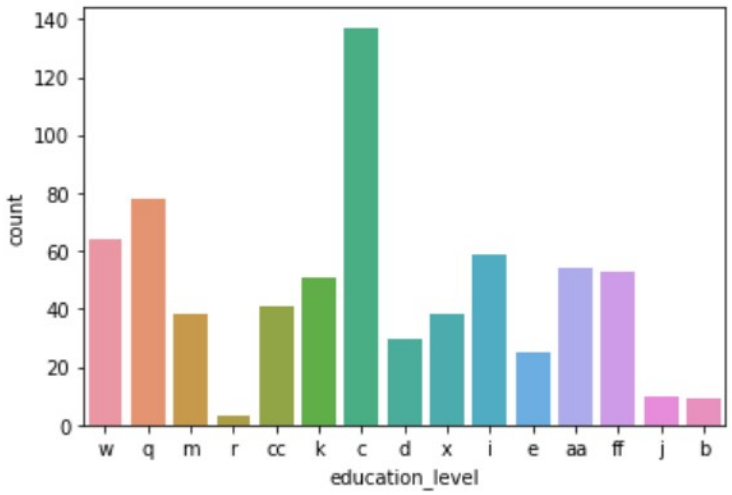

Figure 3: Education Level for Credit Card Applicants

We see that people from a wide variety of educational backgrounds applied for credit cards. This seems to be normal, as expected.

Lastly, we will look at the amount of people who applied for credit cards with prior defaults

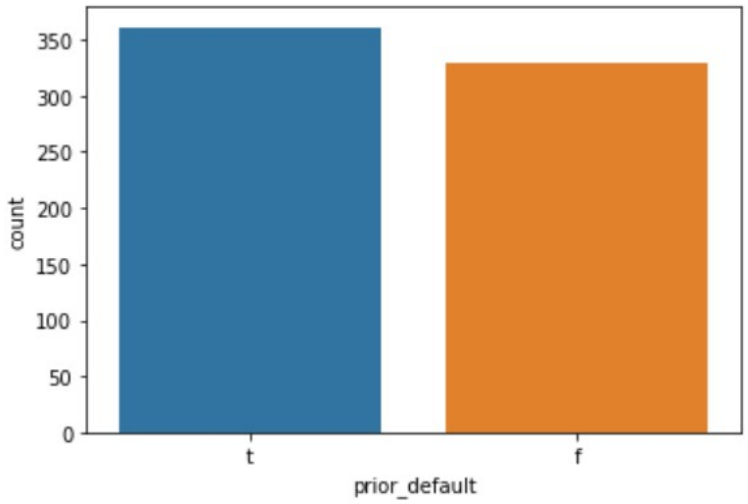

Figure 4: Prior Default of Credit Card Applicants

Surprisingly, there is not much of a difference in number of applicants with prior defaults. One would think that applicants with prior defaults would not apply for credit cards. This demonstrates why its important to be able to determine whether one should be issued a credit card. The financial institution is more at risk of losing money with applicants with prior defaults.

4 Model Development and Selection

Before we split the data, we encoded the categorical features of our data, so that it will be in the proper form when we built the model. Then, we split the data into a training and test data set. Finally, once the data was split, we applied feature scaling to our data. We used the `MinMaxScaler` function which transforms the data into a given range, in our case between zero and one.

4.1 Data Modeling

For our modeling process, five standard models were selected for comparison: Logistic Regression, Random Forest, XGBoost, KNN, and Decision Trees. The details of each model are given below.

4.1.1 Logistic Regression

The logistic regression model arises from the desire to model the posterior probabilities of the K classes via linear functions in x, while at the same time ensuring they sum to one and remain in $[0, 1]$. The model is specified in terms of $K - 1$ log-odds or logit transformations. Although the model uses the last class as the denominator in the odds-ratios, the choice of the denominator is arbitrary in that the estimates are equivariant under this choice.

4.1.2 Random Forest

In Random Forest, we build a number of deicision trees on bootstrapped training samples. But when building these decision trees, each time a split in a tree is considered, *a random sample of m predictors* is chosen as split candidates from the full list of p predictors. The split is allowed to use only one of those m predictors.

4.1.3 XGBoost

XGBoost is an implementation of Gradient Boosted decision trees. In this algorithm, decision trees are created in sequential form. Weights play an important role in XGBoost. They are assigned to all the independent variables which are then fed into the decision tree which predicts results. The weight of variables predicted wrong by the tree is increased and these variables are then fed to the second decision tree. These individual classifiers/predictors then ensemble to give a strong and more precise model. It can work on regression, classification, ranking, and user-defined prediction problems.

4.1.4 KNN

The K-Nearest Neighbors (KNN) algorithm is a simple, easy-to-implement supervised machine learning algorithm that can be used to solve both classification and regression problems. The KNN algorithm assumes that similar things exist in close proximity. In other words, similar things are near to each other. KNN captures the idea of similarity (sometimes called distance, proximity, or closeness) with some mathematics we might have learned in our childhood— calculating the distance between points on a graph.

4.1.5 Decision Trees

Decision Tree is the most powerful and popular tool for classification and prediction. A Decision tree is a flowchart-like tree structure, where each internal node denotes a test on an attribute, each branch represents an outcome of the test, and each leaf node (terminal node) holds a class label.

4.2 Performance Evaluation

We will use the confusion matrix to compute several metrics for our classification model.

	Predicted: Legitimate	**Predicted: Fradulent**
Actual: Legitimate	True Negative	False Positive
Actual: Fraudulent	False Negative	True Positive

4.2.1 Accuracy:

Accuracy is one metric for evaluating classification models. Informally, accuracy is the fraction of predictions our model got right. Formally, accuracy has the following definition:

$$\text{Accuracy} = \frac{tp + tn}{tp + tn + fp + fn}.$$

4.2.2 Precision

Precision seeks to answer the following question: "What proportion of positive identifications was actually correct?" We define it as follows:

$$\text{Precision} = \frac{tp}{tp + fp}.$$

4.2.3 Recall

Recall seeks to answer the following question: "What proportion of actual positives was identified correctly?" We define it as follows:

$$\text{Recall} = \frac{tp}{tp + fn}.$$

4.2.4 F1

The F1 score can be interpreted as a harmonic mean of the precision and recall, where an F1 score reaches its best value at 1 and worst score at 0. The relative contribution of precision and recall to the F1 score are equal.

$$\text{F1} = 2 \times \frac{\text{precision} \times \text{recall}}{\text{precision} + \text{recall}}$$

4.2.5 Cross-Validation

Cross-validation is a resampling method that uses different portions of the data to test and train a model on different iterations. It is mainly used in settings where the goal is prediction, and one wants to estimate how accurately a predictive model will perform in practice. In a prediction problem, a model is usually given a dataset of known data on which training is run (training dataset), and a dataset of unknown data (or first seen data) against which the model is tested (called the validation dataset or testing set).

We summarize the results below.

Model	Accuracy	F1	Precision	Recall	Cross-Validation Score
Logistic Regression	0.826087	0.833333	0.810811	0.857143	0.8625
Random Forest	0.847826	0.844444	0.876923	0.814286	0.8659
XGBoost	0.840580	0.840580	0.852941	0.828571	0.8497
KNN	0.811594	0.803030	0.854839	0.757143	0.8660
Decision Tree	0.782609	0.776119	0.812500	0.742857	0.8188

From the output above, we elected to go with the random forest model for our data set!

4.2.6 Selecting the Tuning Parameters

Once we elected to go with a random forest model, we then proceeded to find the optimal value for the hyper parameter. We did this by completing a grid search. Applying this method allowed us to increase the accuracy of our model to 86.23%.

5 Feature Importance

Once we determined we would be using a Random Forest model, we went on the see which features are most important when determining credit card approval. The output is shown in Figure 5.

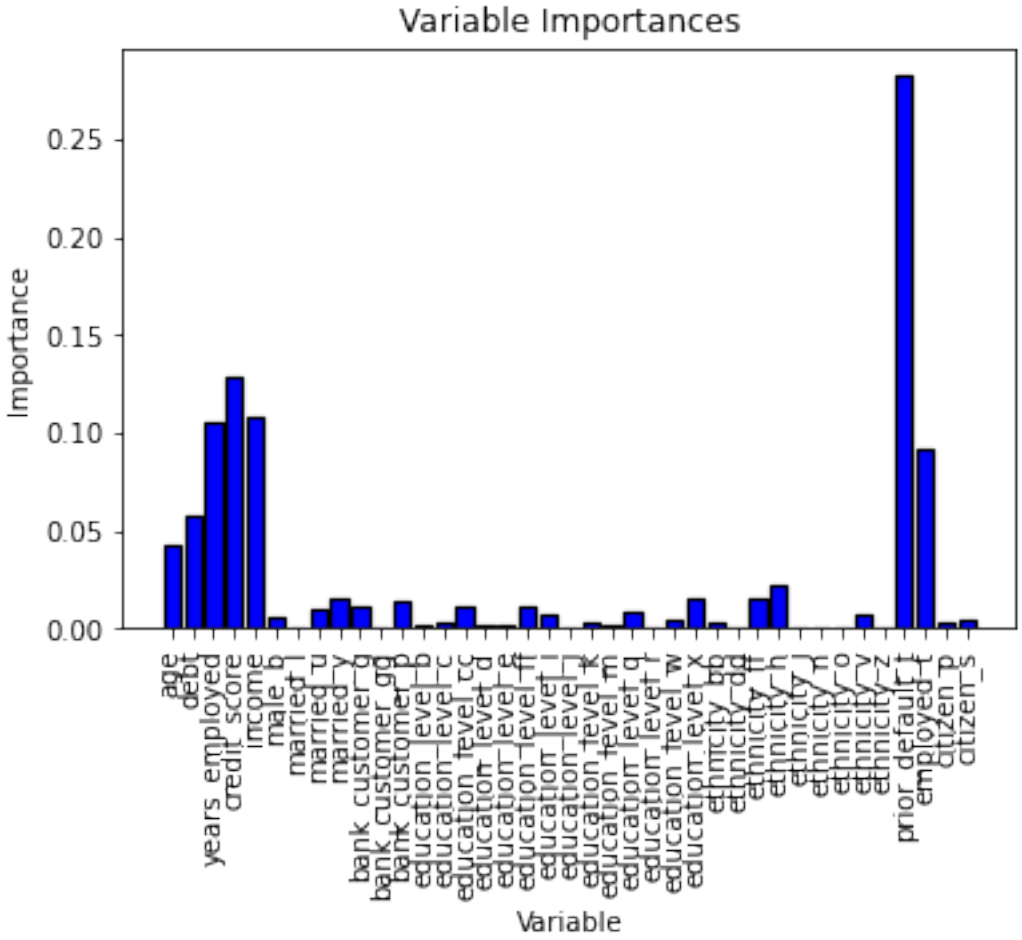

Figure 5: Random Forest Feature Importance

We see that when determining credit card approval `prior_default_t` was the most important feature. This would make sense as financial institutions don't want to give credit to customers who have no intention of paying them back. We also see that `credit_score`, `years_employed`, `income`, `employed`, and `debt` are features that are important when determining credit card approval.

Looking at Figure 6, we see that if we want to explain 95% of the variance, we should use the features up to where the green line crosses the blue line, which is `bank_customer_g`

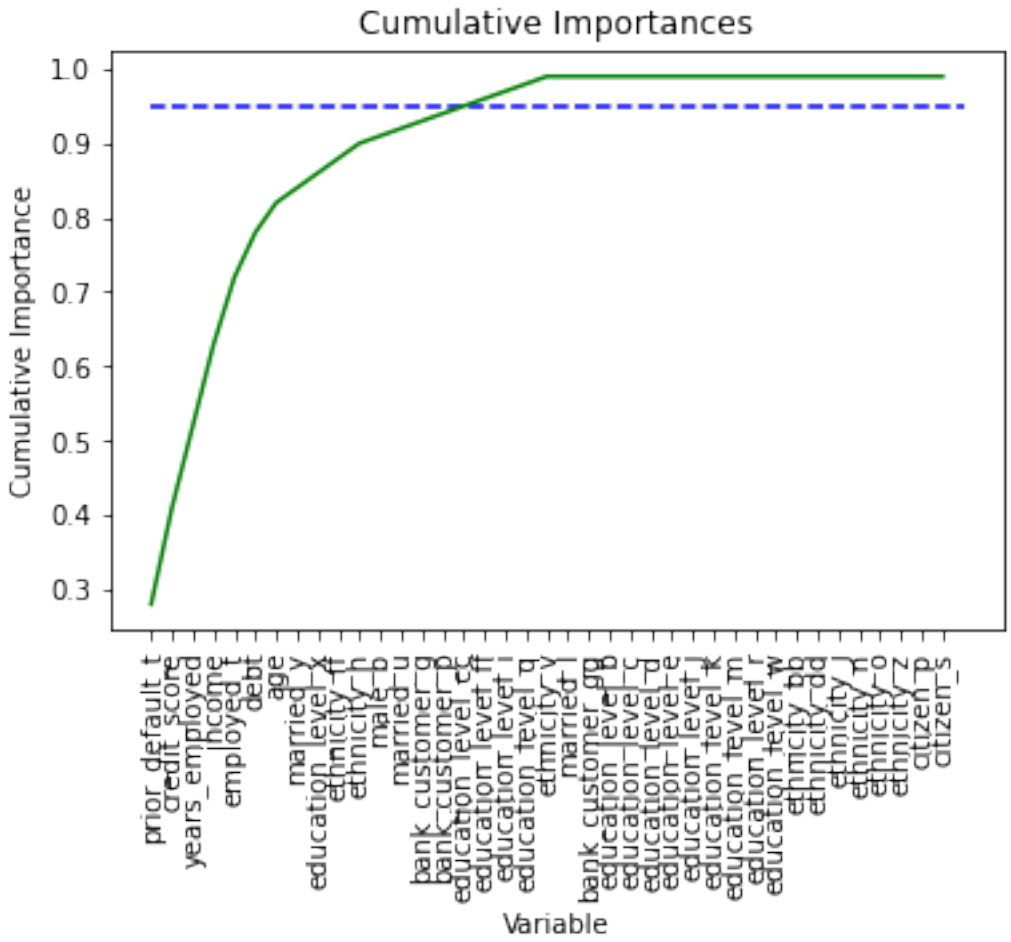

Figure 6: Cumulative Importance Features

6 Conclusion and Recommendations

In this report, we developed a model to determine credit card approval as well as determine which features play an important role in making that decision. The model accurately predicted credit card approval 86.23% of the time. Based on our findings, we have the following recommendations.

1. Prior default is the most important feature when determining credit card approval. As a result, it is not recommended that clients who have defaulted in the past be issued credit-cards, unless there are good reasons to do so.

2. Credit score, income, and years employed are very important when making this decision. These factors impacted the model more than debt, which could mean that if a client has a large amount of debt, it does not mean that they are necessarily bad candidates for credit cards. The other factors should play a factor as well.

3. As data sets become more and more readily available, it may prove to be beneficial to conduct these analysis again in the future to see if this trend is continuing or if there are other features we should consider when making credit card approval decisions.