Predicting Credit Card Approvals

Author: Sathish Manthani

Date: 1/9/2021

Contents

Introduction

Image source: paizen

Predicting credit card approval is a classical usecase in finance industry. Analyzing the credit card applications data for a handful of applications can be done manually but if credit institutions get a large number of requests then analyzing them manually is not feasible. The applications are going to increase with growing number of digital devices and overall digital transformation across the globe. Credit card companies would need a program to analyze the historical patterns of the data and understand the impacting factors in an application and decide whether to approve the application or not based on the risk score of the applicant.

Credit Analysis involves the statistical and qualitative measure to analyze the probability of a customer to pay back the loan to the bank in time and predict its default characteristic. Analysis focus on identifying and reducing the financial risks involved which may otherwise results in the losses incurred by the company while lending. The loss of business risk will also be there by not approving the application of eligible candidate. So, it is important to manage credit risk and

handle challenges efficiently for credit decision as it can have adverse effects on credit management. Hence, assessing credit approval is significant before granting decision on each application.

In this project, I would be creating a machine learning program to read the data for existing approval/rejection patterns of the credit card applications and build a ML algorithm to predict the approval status for future credit card applications. I used predictive analytics and machine learning algorithms like logistic regression to predict whether the application is approved or not.

Research Questions

Below are my research questions for this project:

- What are the data sources for credit card applications data. Which one works for our research.
- What are attributes available in credit card applications data. Would there be any optional attributes for which the data would be missing
- What features are influencing the approval status the most.
- What features are least related and can be removed from the dataset
- Are there any attributes negatively or positively moving with the approval status variable.
- Look at several data visualizations to understand the underlying data
- What models can work best for this dataset
- Do we have to tune any parameters in the model to make it fit the dataset

Data Sources

I looked at multiple datasets on Kaggle and other data sites for credit card applications data. There are a lot of datasets available related to credit card banking data but most of them do not have all the attributes of a typical credit card application. So I chose dataset provided by UCI Machine Learning Repository which has a limited number of observations but it covers all the features that I'm looking for.

Features available in the dataset are:

Feature Name	**Description**
Gender	Represents the applicant's gender. Nominal value (b, a).
Debt	Debt amount of the applicant
BankCustomer	Represents if the applicant is a bank customer or not. Nominal value (p, gg)
Ethnicity	Represents the Ethnicity. Nominal value (v, h, bb, j, n, z, dd, ff, o).
PriorDefault	Represents the PriorDefault. Nominal value (t, f)
CreditScore	Represents if the client is employed,

Citizen	Represents the Citizen. Nominal value (g, p, s).
Income	Income of the applicant
Age	Age of the applicant
Married	Represents the marital status of the applicant. Nominal value (u, y, l, t)
EducationLevel	It is a nominal variable. Possible values are (c, d, cc, i, j, k, m, r, q, w, x, e, aa, ff). They represent the education level.
YearsEmployed	Continuous variable and represents the YearsEmployed
Employed	Represents whether applicant is employed or not. Nominal value (t, f)
DriversLicense	Represents whether applicant has driver license or not. Nominal value (t, f)
ZipCode	Represents the ZipCode
ApprovalStatus	Represents the approval status of the applicant. Nominal values (+, -)

Exploratory Analysis

The first step is to analyze the dataset by looking at the codebook. A quick peek at the codebook and dataset shows us that all the values are converted to unusual symbols to protect the sensitivity of the applicant's data. This would still be okay for us to do the analysis to develop machine learning model.

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
0	b	30.83	0.000	u	g	w	v	1.25	t	t	1	f	g	00202	0	+
1	a	58.67	4.460	u	g	q	h	3.04	t	t	6	f	g	00043	560	+
2	a	24.50	0.500	u	g	q	h	1.50	t	f	0	f	g	00280	824	+
3	b	27.83	1.540	u	g	w	v	3.75	t	t	5	t	g	00100	3	+
4	b	20.17	5.625	u	g	w	v	1.71	t	f	0	f	s	00120	0	+

These variables did not come with Header. So, I used the codebook to name these columns in the dataframe:

Male	Age	Debt	Married	BankCustomer	EducationLevel	Ethnicity	YearsEmployed	PriorDefault	Employed	CreditScore	DriversLicense	Citizen	ZipCode	Income	Approved
b	30.83	0.000	u	g	w	v	1.25	t	t	1	f	g	00202	0	+
a	58.67	4.460	u	g	q	h	3.04	t	t	6	f	g	00043	560	+
a	24.50	0.500	u	g	q	h	1.50	t	f	0	f	g	00280	824	+
b	27.83	1.540	u	g	w	v	3.75	t	t	5	t	g	00100	3	+
b	20.17	5.625	u	g	w	v	1.71	t	f	0	f	s	00120	0	+

We can see that the data values in Approved status are '+' or '-'. It is obvious that '+' is for approved applications and '-' is for denied applications. These character symbols are still not much meaningful as is so we would have to transform them. We can turn '+' to a '1' and the '-'

to a ‘0’. This would greatly help the classification and logistic regression models later in the analysis.

We will have to do some additional transformations on the dataset like missing values. If you observe the data in the screenshot, this dataset has numerical as well as non-numerical features. So, we may have to do some preprocessing to convert the non-numeric features to numeric. Let’s examine the statistics of the dataset.

	Debt	YearsEmployed	CreditScore	Income
count	690.000000	690.000000	690.00000	690.000000
mean	4.758725	2.223406	2.40000	1017.385507
std	4.978163	3.346513	4.86294	5210.102598
min	0.000000	0.000000	0.00000	0.000000
25%	1.000000	0.165000	0.00000	0.000000
50%	2.750000	1.000000	0.00000	5.000000
75%	7.207500	2.625000	3.00000	395.500000
max	28.000000	28.500000	67.00000	100000.000000

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 690 entries, 0 to 689
Data columns (total 16 columns):
Male              690 non-null object
Age               690 non-null object
Debt              690 non-null float64
Married           690 non-null object
BankCustomer      690 non-null object
EducationLevel    690 non-null object
Ethnicity         690 non-null object
YearsEmployed     690 non-null float64
PriorDefault      690 non-null object
Employed          690 non-null object
CreditScore       690 non-null int64
DriversLicense    690 non-null object
Citizen           690 non-null object
ZipCode           690 non-null object
Income            690 non-null int64
Approved          690 non-null object
dtypes: float64(2), int64(2), object(12)
memory usage: 86.3+ KB
None
```

Descriptive Statistics and Missing Values

- The dataset contains numeric and non-numeric data. It is important to note most of the features contain non-numeric values.
- The dataset also contains values from several ranges. Few features have a value range of 0 - 28, some have a range of 2 - 67, and some have a range of 1017 - 100000. So, I think we would have to scale these features to a uniform range.
- The dataset also has some missing values which we will handling now. The missing values in the dataset are labeled with '?'. So, I replaced the '?' values with NaN initially.
- We should also note that ignoring a lot of missing values can affect the performance of the machine learning model. Because ignoring the missing values may lead to missing

out on information about the dataset that is useful for model training. So, to avoid this problem, lets replace the missing values with Mean.

- There is another problem, some of the features are non-numeric so replacing it with Mean is not a possible solution for them. So, I am going to replace missing values for non-numeric features with most frequent values.

Preprocessing

We have handled the missing values, but I like I mentioned earlier, we would still have to convert the non-numeric features to numeric features. That is the pre-processing step will be doing now. We will do this conversion to achieve speed in machine learning models computation and also, many machine learning models expects the features to be in numeric values. Later, I would scale the features to a uniform range using Min Max Scaler.

Test/Train data split

After the numeric feature conversion, we also have to split the dataset into training and test sets. Where I will use Training dataset to build the machine learning model and test dataset to test the accuracy of the machine learning model.

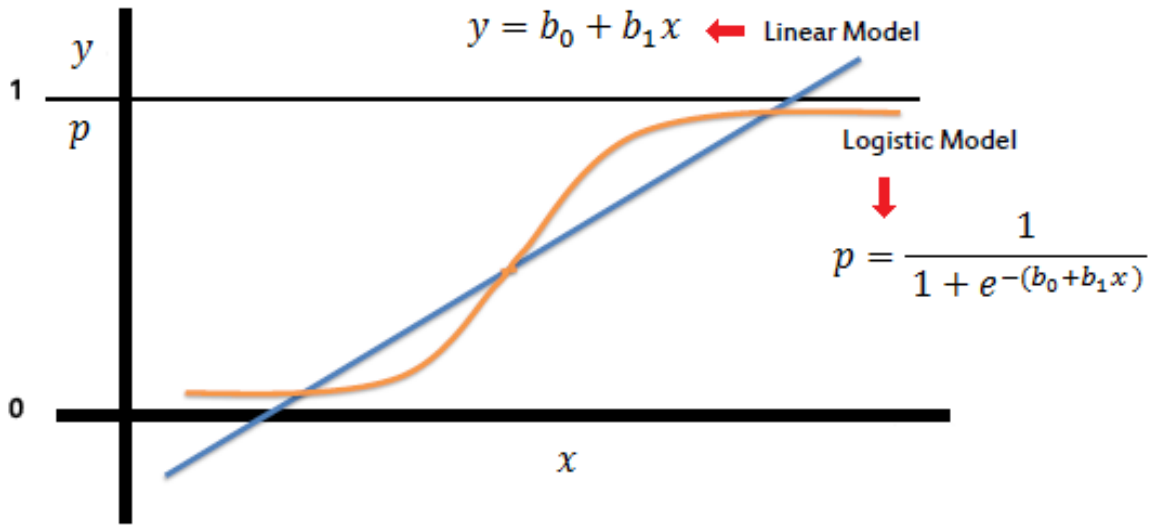
Modeling and evaluation

Regression models are useful for predicting continuous (numeric) variables. However, the target value in Approved is binary and can only be values of 1 or 0. The applicant can either be issued a credit card or denied- they cannot receive a partial credit card. We could use linear regression to predict the approval decision using threshold and anything below assigned to 0 and anything above is assigned to 1. Unfortunately, the predicted values could be well outside of the 0 to 1

expected range. Therefore, linear, or multivariate regression will not be effective for predicting the values. Instead, logistic regression will be more useful because it will produce probability that the target value is 1. Probabilities are always between 0 and 1 so the output will more closely match the target value range than linear regression.

Logistic Regression model

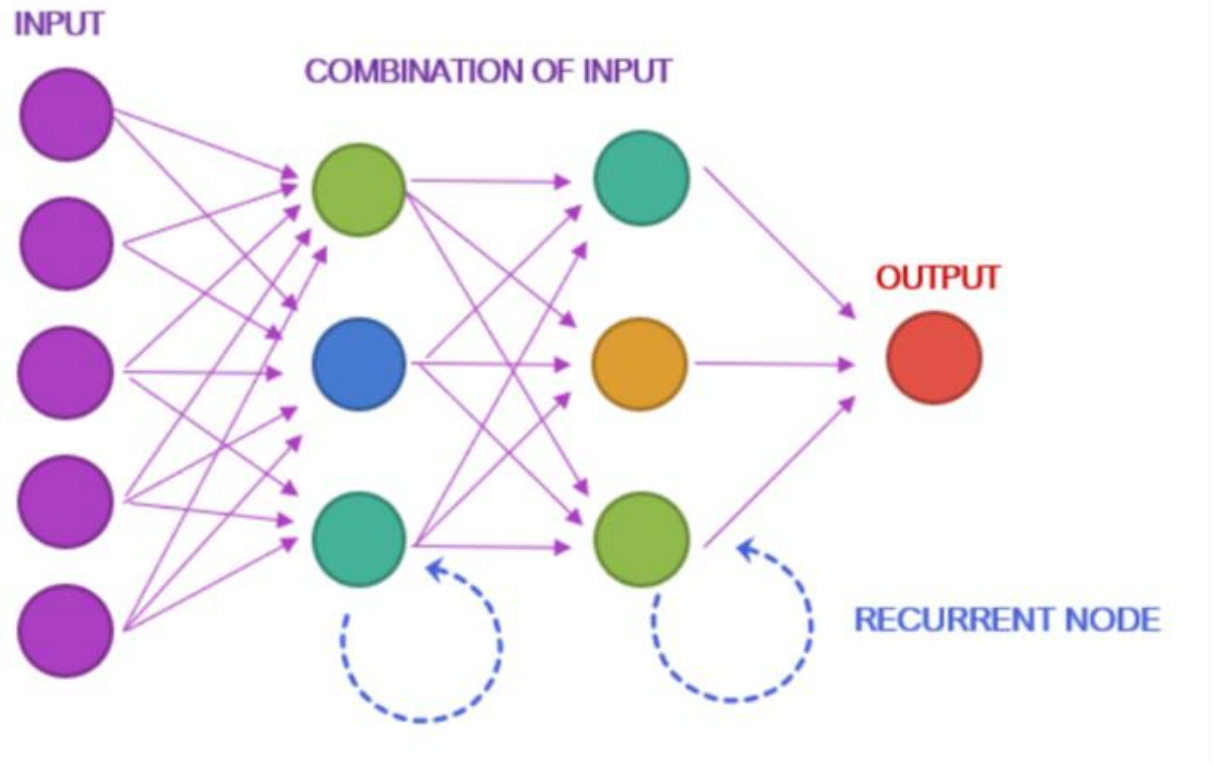
I started my modeling with logistic regression because predicting whether a credit card application is approved or not is a *classification* problem. When I analyzed the data for number of approved and denied applications in the datasets – it came out to be about 45% are approved and 55% are denied applications.



Logistic regression modeling gave me accuracy of 84% initially. So, I tuned the parameters in Gridsearch further to get better performance. I got accuracy of around 85% after performance tuning.

Neural Networks

After logistic regression model, I tried Neural network binaryclassifer on the dataset.



Neural Networks gave me similar accuracy. I tried different epoch iterations, yet the the accuracy remained same. I used Relu sigmoid activation function.

Summary/Conclusion

The goal of this project is to predict the credit card approvals using machine learning model. I started with data extraction using pandas library and did the exploratory analysis on the data. I realized there are a lot of missing values in the datasets, so I took the approach of substituting the missing values with mean values for numeric features. After looking at the descriptive statistics, I felt the need to bring the data values in few features to uniform range – so I used Min Max

Scaler to do the scaling. I split the dataset into two parts – Training set and Test Set. I used the training set to build the machine learning model. I tried Logistic regression model. I tuned the parameters in logistic regression model to improve the accuracy of the model. Accuracy of the model increased from 84% to 85%. I achieved the similar accuracy using Neural networks.

References

[1] Credit card approval analysis: https://www.researchgate.net/publication/321002603_Credit_Approval_Analysis_using_R

[2] Application of deep learning for credit card approval: https://www.researchgate.net/publication/341443586_Application_of_deep_learning_for_credit_card_approval_a_comparison_with_two_machine_learning_techniques

[3] Credit Card data: https://www.nerdwallet.com/blog/credit-card-data/

[4] Research and Statistics on Credit cards: https://www.creditcards.com/credit-card-news/research-statistics-stories/

[5] Key Technological Advancements in Finance https://learning.oreilly.com/library/view/machine-learning-applications/9781484237878/html/464968_1_En_14_Chapter.xhtml