**Regression With Credit Cards**

**Multiple Linear Regression using R and Python**

Peyton Fosnough, [pfosnough@bellarmine.edu](mailto:pfosnough@bellarmine.edu)

Zachary Pilbean, [zpilbean@bellarmine.edu](mailto:zpilbean@bellarmine.edu)

**ABSTRACT**

For our project, we selected a continuous dataset containing information on credit card users and using both R and Python we applied multiple linear regression to it to predict certain results. Our goal was to develop a model that could accurately predict credit card spending based on demographic information, such as age, income, and education level. We began by exploring and cleaning the dataset to ensure its accuracy and completeness. Next, we used exploratory data analysis to gain insights into the relationships between the variables and trained the data in order to build an accurate predictive model.

1. **INTRODUCTION**

We used a credit card dataset that contained all the details of each credit card user including gender, age, debt, married, bank customer, industry, ethnicity, years employed, prior default, employment status, credit score, drivers license, citizenship, zip code, income, and their approval status. The person who gathered this data was a manager at a bank who wanted to see how to accurately predict if someone should get approved for a credit card using a variety of variables. We used many elements of MLR to try and find an accurate predictive model.

1. **BACKGROUND**
   1. *Data Set Description*

We found this dataset on Kaggle, which is a website that is host to a variety of different datasets that can be posted by a variety of users. We collected this data from a user who worked at a bank and wanted to have a model to be able to predict whether or not an applicant should be approved for a credit card based on a variety of factors shown below. The dataset contains information on 690 customers with 16 columns of data.

* 1. *Machine Learning Model*

Multiple Linear Regression (MLR) is a statistical technique that helps explain the relationship between a dependent variable and multiple independent variables. The goal of MLR is to create a linear equation that accurately predicts the value of the dependent variable based on the values of the independent variables. To achieve this, MLR uses the least squares method to find the coefficients of each independent variable that best fit the data and minimize the sum of the squared differences between the predicted and actual values of the dependent variable.

In summary, MLR is a powerful tool that enables researchers to identify the relative contributions of multiple independent variables to the overall value of a dependent variable. Constructing a linear equation that describes the relationship between the variables enables researchers to make accurate predictions and better understand the underlying mechanisms that drive the dependent variable.

1. **EXPLORATORY ANALYSIS**

The dataset that we used had 690 samples with 16 columns of various data types, which are listed below, along with the type of variable they were zero null counts in the entire dataset which was incredibly helpful when we were cleaning the data. We found that most of the samples fell between 25 and 35 years old. We also found that the debt generally fell between 2 and 5, and most people had a prior default. The majority of the samples do not get approved for a credit card. All of the data was fairly normal.

**Table 1: Data Types**

|  |  |
| --- | --- |
| *Variable Name* | *Data Type* |
| Gender | Categorical- Int64 |
| Age | Quantitative- Float64 |
| Debt | Quantitative- float644 |
| Married | Categorical- Int64 |
| BankCustomer | Categorical- Int64 |
| Industry | Categorical- object |
| Ethnicity | Categorical- Object |
| YearsEmployed | Quantitative- float64 |
| PriorDefault | Categorical- Int64 |
| Employed | Categorical- Int64 |
| CreditScore | Quantitative- Int64 |
| DriversLicense | Categorical- Int64 |
| Citizen | Categorical- Object |
| ZipCode | Quantitative- Int64 |
| Income | Quantitative- Int64 |
| Approved | Categorical- Int64 |

1. **METHODS**
   1. *Data Preparation*

Describe how you prepared the data for your model. For example, you might need to normalize the data, so variables with wider ranges of values don’t overshadow variables with smaller ranges. If you decide to drop variables from the model or create variables from existing columns, explain the process and the reasoning behind those decisions.

* 1. *Experimental Design*

Table 2: Experiment Parameters

|  |  |
| --- | --- |
| **Experiment Number** | **Parameters** |
| 1 | 80/10/10 split for train, validate, and test |
| 2 | 70/15/15 split for train, validate, and test |
| 3 | 60/20/20 split for train, validate, and test |

* 1. *Tools Used*

For the analysis, two software tools were used: R and Python. The R programming language was used in the R Studio environment. The Tidyverse package in R was utilized for data preparation, which includes data cleaning, data transformation, and data visualization. Tidyverse is a collection of packages that provides a consistent approach to data manipulation and transformation, making it easier to work with data. The catools package in R is used for sorting and sampling datasets.

Python was used in the Anaconda environment, which is a platform that provides easy access to a variety of scientific computing packages. Pandas, a Python library, was used to clean and transform data. Matplotlib and seaborn were used to create data visualizations. Sklearn provides a wide range of tools and we used it to model the relationship between our dependent variable and the independent variables.

1. **RESULTS**
   1. *Mean square Error and R-Square calculation*

MSE (Mean Squared Error) is a commonly used metric to evaluate the performance of a regression model. It measures the average squared difference between the predicted values and the actual values in the dataset. The formula is shown below:

Text

Description automatically generated with medium confidence

Where n is the number of data points, yi is the actual value of the i-th data point, and y^i is the predicted value of the i-th data point.

R-square (R2) is a statistical measure that represents the proportion of variance in the dependent variable that is explained by the independent variables in a regression model. It ranges from 0 to 1, with a higher value indicating a better fit of the model. The formula for R2 is:

A picture containing chart

Description automatically generated

Where RSS is the sum of squared residuals and TSS is the total sum of squares.

In our experiments done in R Studio, we got the following results:

|  |  |  |
| --- | --- | --- |
| **Experiment Number** | **MSE Value** | **R^2 Value** |
| 1 | 0.096 | 0.609 |
| 2 | 0.096 | 0.609 |
| 3 | 0.096 | 0.609 |

And in our Experiment on Python, we got:

|  |  |  |
| --- | --- | --- |
| **Experiment Number** | **MSE Value** | **R^2 Value** |
| 1 | 0.13 | 0.48 |
| 2 | 0.13 | 0.46 |
| 3 | 0.18 | 0.28 |

* 1. *Discussion of Results*

From these results, we have concluded that the tests and experiments done in R are our best and most accurate models, due to the results of them having the lowest MSE value, indicating that the model is best at predicting the dependent variable based on the independent variables. Another factor we took into consideration was the R^2 value, of which the experiments we did in R Studio were the highest. R2 measures the proportion of variance in the dependent variable that is explained by the independent variables in the model. A higher R2 value indicates a better fit of the model to the data. Our worst experiment by these standards was our third experiment in python, which had the highest MSE value and the lowest R^2 value.

* 1. *Problems Encountered*

One of the biggest problems that we encountered was which model to use. Because we knew that linear regression follows the assumption that there is a linear relationship between the variables, we knew that there might not be a relationship between the two and in which case the predictions would not be accurate. Another problem we ran into was we thought our dataset might be a little too complex for a regression model. We have a relatively small sample size in comparison to other datasets (only 690 samples) and we have 16 columns of data for each sample, which means there was a risk of the model fitting certain fluctuations in the data instead of showing us the pattern in it.

* 1. *Limitations of Implementation*

There are several reasons why this might not be the best way to model the data. First and foremost, we are assuming that the relationship between variables is linear, so if it is not our model wouldn’t be accurate. Also, there is a possibility that our independent variables are correlated with each other, which often times causes problems with the model and makes it difficult to interpret results. There are several other models we could use to bypass certain limitations, including a logistic regression model.

* 1. *Improvements/Future Work*

I think adding a different dataset would be interesting, specifically something significantly bigger than what we worked with. I think having a larger dataset to manipulate would be helpful as the results may be more accurate and representative of a larger population, and I think in the future it might be wiser to do more experiments with the data which could help ensure us that the relationship between variables is linear and help assess the significance of different variables.

1. **CONCLUSION**

In our project we used a continuous dataset containing information on credit card users, and applied multiple linear regression using both R Studio and Python to predict credit card spending based on a variety of factors. Exploratory data analysis was used to gain insights into the relationships between the variables and train the data to build an accurate predictive model. We used a machine learning model called Multiple Linear Regression (MLR) to create a linear equation that accurately predicts the value of the dependent variable based on the values of the independent variables. We found that our experiments done in R prove to be our best model due to the consistency throughout the experiments as well as having the highest R^2 value and lowest MSE value in comparison to the other experiments. These two values were what we used to evaluate the performance of each regression model.

**REFERENCES**

None