5/5/2021

Philip Dowhy, Bevon Gomez, Shristee Sinha

ISYS 620 Course Project Spring 2021

Home Credit Default Risk

***Introduction***

Given the impact Covid-19 has had on our current environment, both socially and economically. We thought that it would be pertinent to explore alternative methods of determining credit worthiness and the ability for a borrower to repay their loan. In particular, we decided to focus on the dataset provided by the Home Credit Group. “Home Credit makes use of a variety of alternative data—including telco and transactional information—to predict their clients’ repayment abilities.” (Kaggle) Traditional methods such as the FICO score, Debt-to-income (DTI) Ratio, Collateral Value, Amount of assets, and loan conditions may not be sufficient to paint an accurate picture of a borrower’s ability to repay. (Corporate Finance Institute) Our goal is to use statistics, in particular various machine learning algorithms, to predict the ability to repay of borrowers who may not be properly represented when taking a more traditional approach. Given the breadth of features in our dataset, we capture many of the input variables and use a Principal Component Analysis to reduce the dimensionality down to the most representative for our target variable, loan repayment classification. Using these inputs, we then explore the Logistic Regression, Random Forest, and Neural Networks algorithms to determine which model is best suited for our purpose.

***Describing the Dataset***

<https://www.kaggle.com/c/home-credit-default-risk/data>

**Data Preprocessing:**

Linked above, the training dataset contains 307,511 rows and 122 columns. The Test dataset contains 48,744 rows and 121 columns. There were large portions of the data that had missing, null values. The initial pass of cleaning the dataset from large portions of missing information resulted in a reduction of input variables to 57. In the train dataset, there were two variables which still contained missing values. These were imputed using the input means to make the features whole for the principal component analysis. The “SK\_ID\_CURR” feature was removed from our data as it was not relevant for our prediction models. The qualitative data was then transformed using label encoding from categorical/text to numerical data while minimizing the addition of columns (Yadav,2019). The resulting dataset was highly imbalanced and later determined to be negatively affecting the prediction models as the output value of 1 (client payment difficulties) was underrepresented, a minority class. An oversampling method was used to add new samples of the minority class and restore balance. Finally, the dataset was split and scaled in preparation of our subsequent analysis.

**Clean Data Analysis:**

**Contract Type**

Cash loans 278232

Revolving loans 29279

**Income Type**

Working 158774

Commercial associate 71617

Pensioner 55362

State servant 21703

Unemployed 22

Student 18

Businessman 10

Maternity leave 5

**Family Status**

Married 196432

Single / not married 45444

Civil marriage 29775

Separated 19770

Widow 16088

Unknown 2

**Housing Type**

House / apartment 272868

With parents 14840

Municipal apartment 11183

Rented apartment 4881

Office apartment 2617

Co-op apartment 1122

**Average Income by Education Type**

Academic degree 240009.146341

Higher education 208652.053814

Incomplete higher 181563.812397

Secondary / secondary special 155158.512138

Lower secondary 130079.358491

**Repayment Difficulty by Education Type (1 = repayment difficulty, 0 = all other)**

Academic degree 0 161

1 3

Higher education 0 70854

1 4009

Incomplete higher 0 9405

1 872

Lower secondary 0 3399

1 417

Secondary / secondary special 0 198867

1 19524

***Description of Learning Algorithms***

***Principal Component Analysis (PCA)***

PCA is a statistical method that uses orthogonal transformation to convert a set of possibly correlated variables into a set of linearly uncorrelated variables called principal components. These components summarize the total input variables into a smaller number that represent whole. Each component is a best fitting line that captures that captures the most variance and is orthogonal to the previous, apart from the first which is determined by the line that captures the maximum. Given that our dataset has many features, we decided to use utilize PCA to give better insight into which one we should include in the model while still accounting for a large portion of the variance. Since there is some loss of variance(information) when reducing the dimensional space, we calculated the explained variance of the components in order to confirm that we are still capturing much of the information while reducing the storage space and eliminating and redundant features. (Galarnyk,2017)

***Logistic Regression***

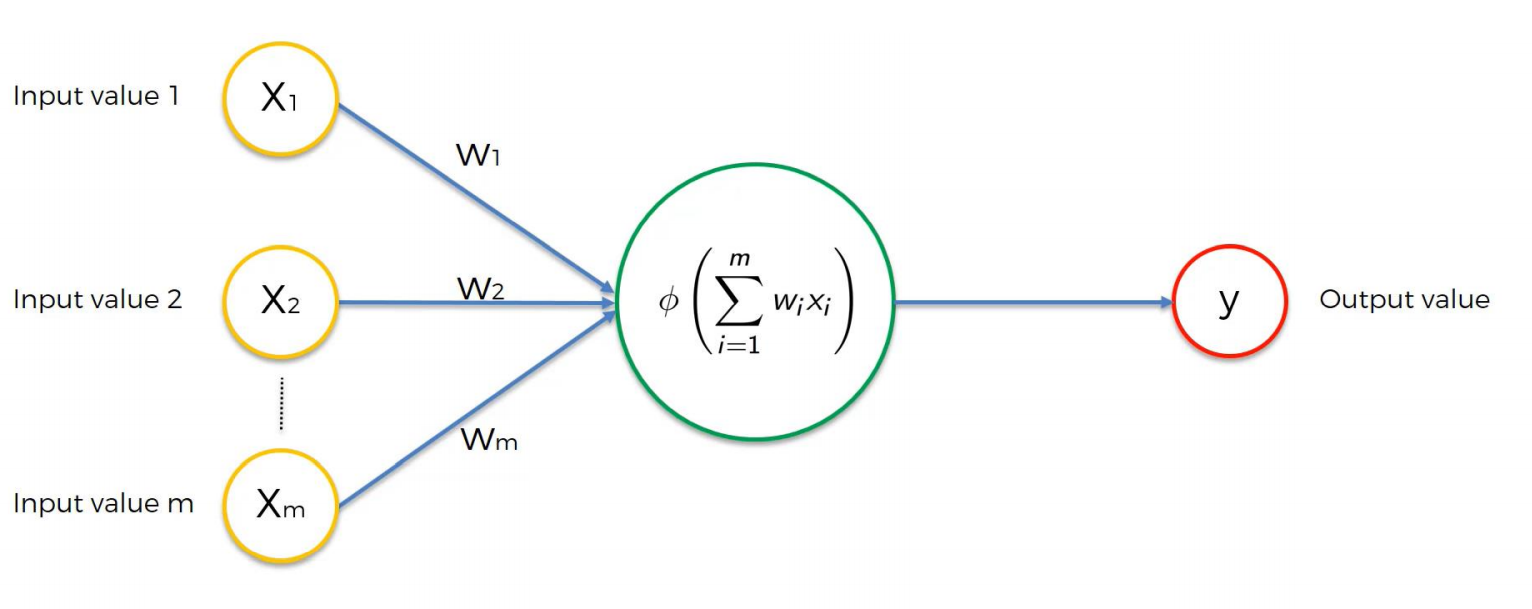
Logistic Regression is a supervised classification algorithm that is used to predict the probability of a discrete categorical value of a dependent variable. (Li,2017). The algorithm takes the linear equation and transforms it into a probability calculation using the Sigmoid Function. As stated above, our dependent variable is the binary target value of loan repayment difficulties where 1 indicates loan repayment difficulties and 0 indicates all other cases. This makes the algorithm a potential fit for achieving our desired results.

***Random Forest***

The Random Forest algorithm is a supervised classification or regression algorithm that leverages ensemble learning to combine the same decision tree algorithm multiple times and form a powerful prediction model (Malik). To understand the Random Forest algorithm, we first must understand what the Decision Trees algorithm does. The Decision Tree is a tree-structured plan of attributes that repetitively split the data to maximize the information gain, reduction in data impurity or entropy. The root node, or entire data population is divided into two or more homogeneous decision nodes and repeated until a desired terminal (decision) node. Random Forest uses a multitude of these trees concurrently to predict the output variable.

***Neural Networks***

A Neural Network is the mathematical representation of our brain cell. Each layer in a neural network is like one neuron in the brain cell. It has dendrites, a neuron, and an axon. Dendrites are the part of the cell which is responsible for getting the inputs from the receptors. Neurons are the nucleus of the cell, which is the main working block of the cell. Axons work as the output which gives out the signal out of the cell. The neural network in machine learning works the same way. The dendrite forms the input layers in the network which is followed by hidden layer which are nucleus of the cell. Lastly, axon forms the output layer to the network. The hidden layer consists of various nodes through which apply activation functions. The input is attached with weights and when the neural networks learn from backpropagation, these weights get adjusted in order to predict accurate values. Optimization in the neural network is done through gradient descent or stochastic gradient descent.



Neural Networks are supervised learning algorithm which can act as Regressor or Classifier depending on the output variable as set in the network. The hidden layers up to 2 are considered Neural network and layers more than 2 are considered to be Deep Learning algorithm.

***Discussion and Conclusions***

***Principal Component Analysis (PCA)***

The PCA was initially ran with no principal components. The sum of the variances was calculated using the numpy cumsum method and plotted to visualize the most efficient number of components while capturing a sufficient variance in the data. It was initially determined that 2 principal components captured ~ 15% of the variance, 30 components captured ~75% and 50 captured 99.6%. Considering the implications of inability to repay on the client and company, and keeping the overall model performance in mind, 50 principal components were chosen to capture over 99% of the variance in the data.

***Logistic Regression***

Sklearn.linear\_model.LosisticRegression was imported from the sklearn library to analyze the resulting data. The model produced an accuracy score of 80% which indicated that it was somewhat accurate. 68,553 true negatives, no difficulties with repayment, and 67,455 true positives, client had difficulties with repayment, were returned. There were 17,197 results returned with a type 2 error, client was incorrectly predicted to have no difficulties in repayment, and 16,407 with a type 1 error, client was incorrectly predicted to have difficulties in repayment. A grid search was subsequently used to tune the hyperparameters of the solver in the model. There was marginal improvement as an accuracy score of 80% was returned with 17,197 type 2 errors and 16,406 type 1 errors.

***Neural Network***

Keras.models.Sequential was imported from the keras library. After oversampling the unbalanced data, standardization and PCA, we considered 50 input variables to the neural network. The hidden layer was calculated to be 25((50+1)/2). The output variable is a sigmoid function as we needed to calculate a binary result where or not the client had loan repayment difficulties. The threshold for the probability was 50%. The batch size and epoch number were finalized after running the Grid Search on the model. The final batch size and epoch was 10 and 100, respectively. The model returned an accuracy of 90%. Analyzing the confusion matrix, we have 12,923 False Negatives, Type 2 error, and 4,660 False Positives, Type 2 error. According to the classification matrix, the accuracy of our neural network model is 90% which indicated that it was our most accurate model.

***Random Forest Classification***

Sklearn.ensemble.RandomForestClassifier was imported from the sklearn library. A grid search was subsequently used to optimize the number of trees in the model. The model produced an accuracy score of 88% which indicated that it was more accurate than the logistic regression model, but less accurate than the Neural Networks model. 76,875 true negatives, no difficulties with repayment, and 72,307 true positives, client had difficulties with repayment, were returned. There were 12,345 results returned with a type 2 error, client was incorrectly predicted to have no difficulties in repayment, and 8,085 with a type 1 error, client was incorrectly predicted to have difficulties in repayment.

***Conclusion***

Of the three supervised models used for predictions, the neural network returned the most promising results with an accuracy of 90% and the least amount of type 1 and type 2 errors. The accuracy of the model could be improved upon in retrospect.

In the dataset, there was additional information available in other tables that were not included in the testing. Some of this information could prove to significantly affect the model performance. In addition, there was a large portion of data that was lost in the data cleaning. A further analysis of different methods to effectively impute missing values and preserve some of the data could also have made a positive impact on the accuracy of the model. Given the size of the data and the number of columns, it would also be a fair assessment than additional computing power would aid in the optimization of the accuracy scores.

First and foremost, the goal of the exploration was to determine if it was possible to effectively determine loan repayment difficulties and client credit worthiness using nontraditional credit approval methods. Most of the features used in our models were not traditional. Independent variables used in our analysis included education level, income type, family status, current housing type, did they provide all the documents (2-21), does the client live in a highly populated area, is the clients permanent address in the same region of their contact and work addresses provided. Some of the information used in the analysis was more aligned with a traditional approach to loan repayment such income. Notwithstanding, the analysis did seem to indicate that it was possible to predict the client’s loan repayment ability using a broader scope of approval parameters. With an accuracy score of 90% and the potential for optimization, the information provides insight into opportunity Home Credit and other lenders have to garner an increased client base from a previously underrepresented population. Further analysis should be done to determine ethical and legal impacts to using some of the nontraditional variables used in ability to repay calculations to mitigate any discriminatory implications.

***Statement of Individual Contribution***

**All Individuals contributed to the dataset selection, project direction, interpretations, feedback, and report creation.**

Philip Dowhy

* Preprocessing, imputing means of missing data and Label Encoder
* Preprocessing, visualizations
* PCA Model
* Logistic Regression with Gridsearch

Shristee Sinha

* Searched and listed various Banking Datasets for the project
* Oversampling the unbalanced data with the imlearn SMOTE library
* Neural Network along with parameterizing using GridSearch using the keras library

Bevon Gomez

* Preprocessing, Data import and Cleaning using various functions and methods
* Preprocessing, Visualizations
* Radom Forest with GridSearch

***References***

Yadav, Dinesh (2019). Categorical encoding using Label-Encoding and One-Hot-Encoder. Towards Data Science. <https://towardsdatascience.com/categorical-encoding-using-label-encoding-and-one-hot-encoder-911ef77fb5bd>

Kaggle. Home Credit Default Risk. Can you Predict hoe capable each applicant is of repaying a loan? <https://www.kaggle.com/c/home-credit-default-risk/overview/description>

Corporate Finance Institute. What Are the 5 Cs of Credit? <https://corporatefinanceinstitute.com/resources/knowledge/credit/5-cs-of-credit/>

Brownlee, Jason (2020). Tune Hyperparameters for Classification Machine Learning Algorithms. <https://machinelearningmastery.com/hyperparameters-for-classification-machine-learning-algorithms/>

Galarnyk, Michael (2017). PCA using Python. <https://towardsdatascience.com/pca-using-python-scikit-learn-e653f8989e60>

Li, Susan (2017). Building a Logistic Regression in Python, Step by Step. <https://towardsdatascience.com/building-a-logistic-regression-in-python-step-by-step-becd4d56c9c8>

Malik, Usman. Random Forest Algorithm with Python and Scikit-Learn. <https://stackabuse.com/random-forest-algorithm-with-python-and-scikit-learn/>

Loy, James (2018). How to build your own Neural Network from scratch in Python. <https://towardsdatascience.com/how-to-build-your-own-neural-network-from-scratch-in-python-68998a08e4f6>

Engineero (2019). Find out the percentage of missing values in each column in the given dataset. [discussion board]. <https://stackoverflow.com/questions/51070985/find-out-the-percentage-of-missing-values-in-each-column-in-the-given-dataset>

Wijaya, Cornellius Yudha (2020). 5 SMOTE Techniques for oversampling your imbalance data.