University of Windsor

PROJECT REPORT

COMP-8610

NEURAL NETWORKS AND DEEP LEARNING

Credit Card Default Prediction using Neural Networks on TensorFlow

Authors:
Parth Mistry
Mohammed Maaz Khan
Kavan Mehulkumar Dave

Submitted to:
Dr. Robin Gras

February 24, 2023

Abstract

This project aims to develop a predictive model for credit card default risk using deep learning techniques. The study uses a dataset consisting of 30,000 credit card clients in Taiwan, with features such as the client's age, gender, education, marital status, and credit history. The model was developed using Keras, a high-level neural networks API, and TensorFlow, an open-source machine learning framework.

The exploratory data analysis revealed that the majority of clients in the dataset did not default on their credit card payments. Additionally, the study found that features such as education level and marital status had a low correlation with default risk, whereas features such as credit limit, bill amount, and payment amount had a higher correlation. The predictive model achieved an accuracy of 82% on the test set using a parameter grid that included batch normalization, dropout, and L2 regularization.

Contents

1 Introduction

This project aims to explore and analyze a dataset containing information about credit card users in Taiwan. The dataset [1] was obtained from the UCI Machine Learning Repository and consists of 30,000 observations and 24 variables. The variables in the dataset provide information about the credit card usage, demographic characteristics, and payment history of the credit card users.

The analysis will be conducted using various data analysis [2] and visualization techniques in order to gain insights into the credit card usage behavior and identify any trends or patterns in the data. The project will also aim to develop predictive models to help identify the factors that contribute to credit card default.

Overall, this project will provide valuable insights into credit card usage and default behavior, which can be useful for credit card companies and financial institutions to develop strategies to reduce credit card defaults and improve their risk management practices. We will be using Python [3] as the primary programming language in this project.

This report is structured to cover different stages of a data science project. Firstly, an overview of the dataset will be provided, followed by exploratory data analysis to gain insights and understanding of the data. Feature engineering techniques will be applied to enhance the predictive power of the model. Then, a deep learning model will be built and optimized using hyperparameter tuning. Finally, the results and conclusion will be presented based on the performance of the model. By following these steps, a comprehensive analysis of the dataset and the potential predictive power of the model can be achieved. The steps to setup this project can be found here.

2 Data

The UCI Credit Card dataset is a real-life dataset of credit card users in Taiwan, which was made available by the UCI Machine Learning Repository. The dataset contains 25 variables, including demographic information about the credit card holders, credit card usage, payment history, and default status.

The dataset is often used as a benchmark for credit risk assessment and prediction models, and it has been the subject of numerous studies in the field of machine learning and data mining.

The dataset contains a total of 30,000 observations, with each observation representing one credit card holder. The response variable in the dataset is a binary variable indicating whether the credit card holder defaulted on their payment in the following month. The other variables in the dataset include information about the credit card usage and payment history over the previous six months, as well as demographic variables such as age, gender, and marital status.

The UCI Credit Card dataset is a popular dataset for teaching and research purposes, as it provides a realistic and challenging dataset for developing predictive models in the field of credit risk assessment.

3 Exploratory Data Analysis

Before building the model, we conducted exploratory data analysis to gain insights into the data. We looked at the distribution of the target variable, the distribution of the features, and the correlation between the features and the target variable. We found that the data set was imbalanced, with only 22% of the clients defaulting on their credit card payments. We also found that there were several features that were highly correlated with the target variable, such as payment status and amount of bill statement.

We were able to identify some key patterns and trends in the data, such as the correlation between **Education Level** and credit card default. Additionally, we observed that certain features, such as the **Gender**, **Marital Status**, **Age**, **Amount of the previous payment** and the **Credit Limit**, may be important predictors of default. These insights have helped us better understand the factors that contribute to credit card default and can guide us in building more accurate models for predicting the default.

4 Feature Engineering

Based on the insights gained from the exploratory data analysis, we performed feature engineering to create new features that could potentially improve the performance of the model. We created dummy variables for the categorical features, scaled the numerical features, and created new features based on the payment status and amount of bill statement.

```
from sklearn.preprocessing import StandardScaler
from tensorflow.keras.utils import to_categorical

predictors = data.drop(['ID', 'default.payment.next.month'],
    axis=1).values
predictors = StandardScaler().fit_transform(predictors)

target = to_categorical(data['default.payment.next.month'])
```

First, the 'ID' and 'default.payment.next.month' columns are dropped from the dataset, as they are not used for predicting the target variable. Then, the predictors (all remaining features) are standardized using the StandardScaler from the scikit-learn [4] library, which scales the data to have a mean of 0 and a standard deviation of 1. Finally, the target variable, which indicates whether or not the customer defaulted on their credit card payment, is converted into a one-hot encoded categorical variable using the to_categorical function from the Keras library. This preprocessed data can then be used to train a neural network model for credit card default prediction.

5 Model Building

To achieve this objective, the code defines a function called build_model that constructs a neural network model with a specified architecture. The architecture consists of an input layer with 64 nodes, one hidden layers with 32 nodes, and an output layer with two nodes. The hidden layers use the ReLU activation function, while the output layer uses the sigmoid activation function. The model is compiled with the categorical cross-entropy loss function, and the Adam optimizer, with the accuracy metric.

The model is then used to create a KerasClassifier object, which is a wrapper that allows the model to be used with scikit-learn's GridSearchCV [6] function. The GridSearchCV function is used to perform a grid search over a range of hyperparameters to find the best combination of hyperparameters that yields the highest cross-validation accuracy. The hyperparameters that are tuned in this process are the optimizer, the dropout rate, the use of batch normalization, and the L2 regularization rate.

Parameter Grid	
Parameter	**Values**
optimizer	adam, rmsprop, SGD, adagrad
dropout	0.0, 0.2
batch_norm	False, True
l2_reg	0.0, 0.01

Table 1: Grid of hyperparameters for model tuning

Once the GridSearchCV object is created, the fit method is called to train the model on the data and find the best hyperparameters. It performed fitting 3 folds for each of 32 candidates, totalling 96 fits. The results of the grid search are then printed to the console, including the best hyperparameters and the corresponding cross-validation accuracy. Overall, the model building process described in the code above involves defining a neural network model, tuning its hyperparameters using cross-validation, and selecting the best hyperparameters based on their cross-validation performance. This process is essential for creating accurate and robust predictive models that can generalize well to new data.

6 Results

After tuning the hyperparameters with GridSearchCV, we achieved a test accuracy of 82% on the classification task. The best performing combination of hyperparameters was found to be batch normalization set to True, dropout rate set to 0.2, L2 regularization set to 0.0 and the optimizer set to 'rmsprop'.

Parameters	Best Value
Batch Normalization	True
Dropout	0.2
Optimizer	RMSprop
l2 Regularization	0.0

Table 2: Best parameters after tuning

It is important to note that although 82% accuracy is a reasonable result, further improvements could be made to the model performance by exploring other models and techniques such as ensemble methods or fine-tuning the model architecture. Additionally, further investigation could be made into the feature engineering process and how it may impact the model performance. Overall, the results obtained provide a good starting point for predicting credit card default risk using this dataset, but there is still room for improvement.

7 Contribution

This chapter acknowledges and highlights the contributions made by the author(s) in the report. These contributions can include data analysis, datasets, experimental setups, model building, theoretical insights and documentation.

Contribution	**Parth Mistry**	**Mohammed Maaz Khan**	**Kavan Mehulkumar Dave**
Data cleaning & anaylsis, feature engineering		✓	✓
Model building and model evaluation	✓	✓	
Project management, report writing	✓		✓

8 Conclusion

In conclusion, this project aimed to predict credit card defaults using machine learning techniques. After performing exploratory data analysis, feature engineering, and model building, we achieved a test set accuracy of 82% with a parameter grid that included batch normalization as True, dropout as 0.2, l2 regularization as 0.0, and the optimizer set as rmsprop. We found that the most important predictors of credit card default were the history of past payments and the amount of credit card bills.

Furthermore, we observed that there were imbalanced classes in the target variable, with a higher proportion of non-default accounts than default accounts. To address this issue, we used oversampling techniques during the model training process.

Overall, our results suggest that machine learning can be a useful tool for predicting credit card defaults, and can help financial institutions identify high-risk accounts and take appropriate measures to reduce the risk of defaults.

9 Future Work

In the future, this project could be extended to incorporate more advanced feature engineering techniques or explore different machine learning algorithms to further improve the model's performance. Additionally, developing a front-end and back-end system for this project would make it easier for others to use it and apply it to their own data. This would involve building a user-friendly interface that allows users to upload their data, preprocess it, and apply the trained model to make predictions. Overall, this project provides a strong foundation for future research in credit risk assessment and has the potential to be a useful tool for financial institutions looking to improve their decision-making processes.

References

[1] I.-C. Yeh and C. hui Lien, "The comparisons of data mining techniques for the predictive accuracy of probability of default of credit card clients," *Expert Systems with Applications*, vol. 36, no. 2, Part 1, pp. 2473–2480, 2009.

[2] Wes McKinney, "Data Structures for Statistical Computing in Python," in *Proceedings of the 9th Python in Science Conference* (Stéfan van der Walt and Jarrod Millman, eds.), pp. 56 – 61, 2010.

[3] T. E. Oliphant, "Python for scientific computing," *Computing in Science and Engineering*, vol. 9, no. 3, pp. 10–20, 2007.

[4] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and Édouard Duchesnay, "Scikit-learn: Machine learning in python," *Journal of Machine Learning Research*, vol. 12, no. 85, pp. 2825–2830, 2011.

[5] "GitHub - tensorflow/tensorflow: An Open Source Machine Learning Framework for Everyone — github.com." `https://github.com/tensorflow/tensorflow`. Accessed: 2023-01-27.

[6] "How to grid search hyperparameters for deep learning models in python with keras." `https://machinelearningmastery.com/grid-search-hyperparameters-deep-learning-models-python-keras/`. Accessed: 2023-01-25.

[7] "Default of credit card clients dataset." `https://www.kaggle.com/datasets/uciml/default-of-credit-card-clients-dataset`. Accessed: 2023-01-23.