Group40 Report

Deep Learning Extracted Features in Credit Scoring Models

1. Members

資財 0513460 胡明秀

資工 0513407 黃正昕

資財 0513404 姜林寬

2. Introduction / Motivation

▼ Traditionally, banks use non time dependency models, such as linear models, to deal with credit scoring problems. However, the models would not be robust for more complicated real-world data and applications. Thus, we would like to leverage **time-related deep learning models** to provide better insights for this problem.

▼ Dataset:

▼ We use Default of Credit Cards Clients Dataset collected by UCI Machine Learning Repository. This dataset contains information on default payments,demographic factors, credit data, history of payment, and bill statements of credit card clients in Taiwan from April 2005 to September 2005. **We would like to understand the features' insights before we put into the deep learning models**. Thus, with features' names for each columns, we believe this dataset would provide us more insights than most of other datasets that only have anonymous features.

▼ We have 30,000 rows of data and 23 features. The problem is formed as a **binary classification problem,** with client default = 1 and no client default = 0. In order to understand our features better, we divided 23 features into two categories:

▼ **Static Features**:

- Amount of given credit in NT dollars (X1)
- Demographics: Sex (X2), Age (X3), Education (X4), Marriage (X5)

▼ **Dynamic Features**:

- Customers’ past 6-month credit card payment history•
  - Repayment Status (X6-X11)
  - Amount of Bill Statement (X12-X17)
  - Amount of Previous payment (X18-X23)

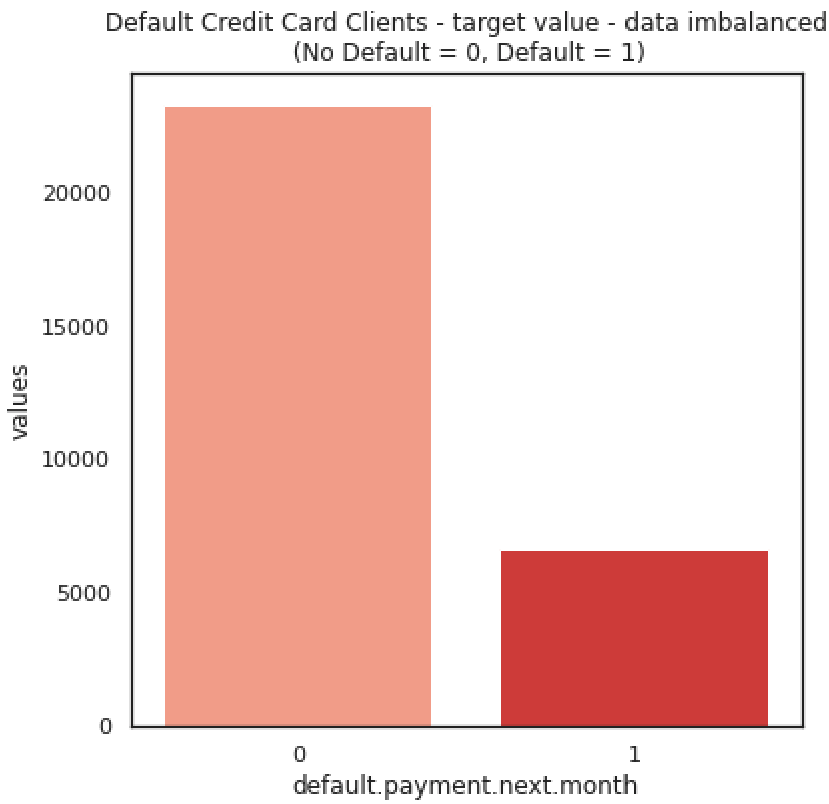
▼ The reason why we divided the features into these two categories is that we would like to know what some of the features are time-correlated.

▼ Exploratory Data Analysis

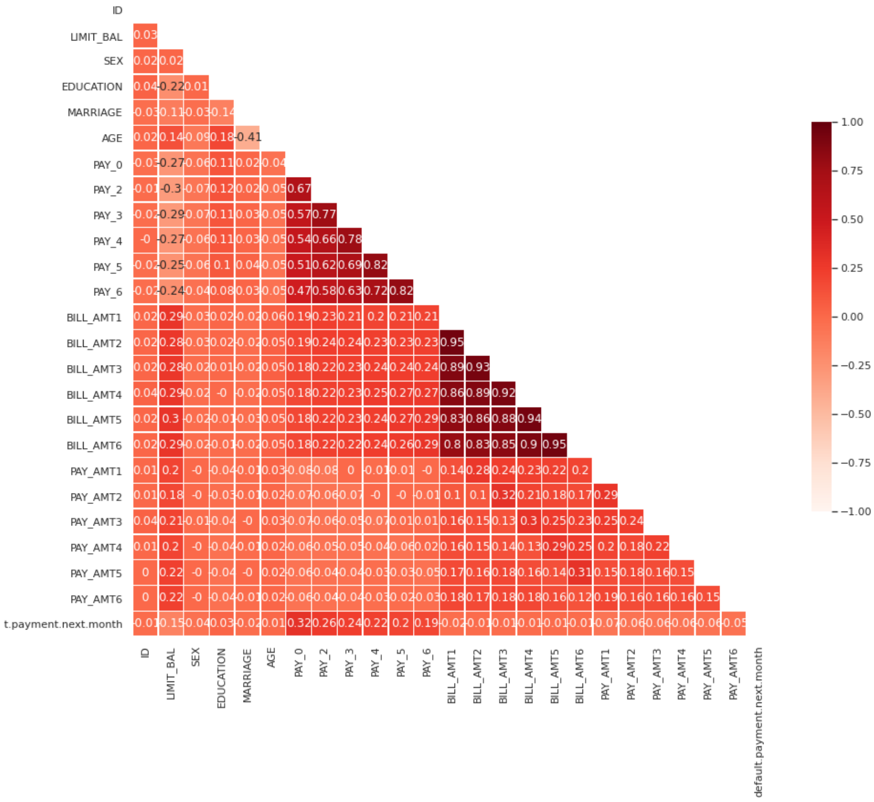
▼ Before we construct our models, we would like to do some EDA (Exploratory Data Analysis) to check if there some features' insights we can leverage.

▼ We first check the target for the data set (0 and 1) and get the following results.

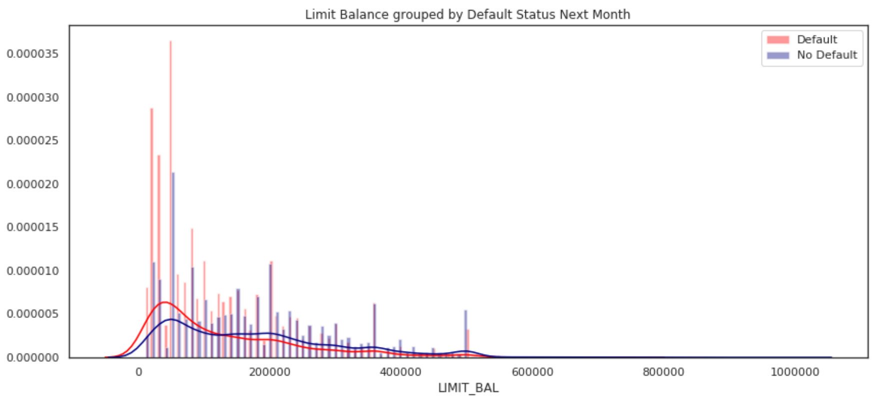
Clearly, the target suffers from an **imbalance data problem**. As a result, we would try to solve this problem by adjusting the weights in the loss function later in the model implementation part.



▼ Then, we plot the heatmap for all the features to check the correlations between each feature. We would examine the heatmap by two categories that stated above (static features and dynamic features). Apparently, **some of the dynamic features are highly correlated with time** (the darker part in the heatmap). Thus, we would try to leverage this part by applying some of the sequential deep learning models such as RNN/LSTM later in the model implementation part.



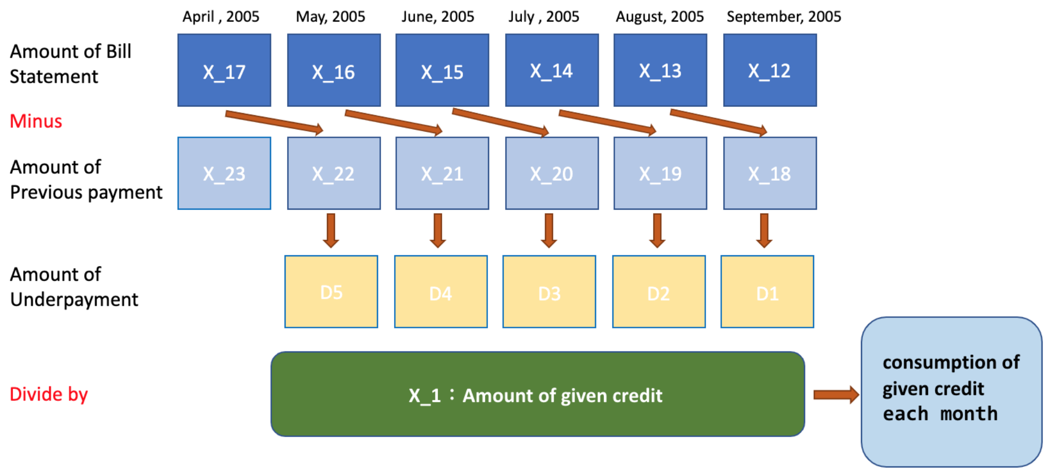
▼ As for the limit balance grouped by label (the default status next month), we plot the histogram and KDE to find out the trend for this feature. We can see that most of the defaults are for credit limits between NT0 - NT100,000. It is intuitive that those with lower limit balance, such as the poor, are more likely to default than those with high limit balance.



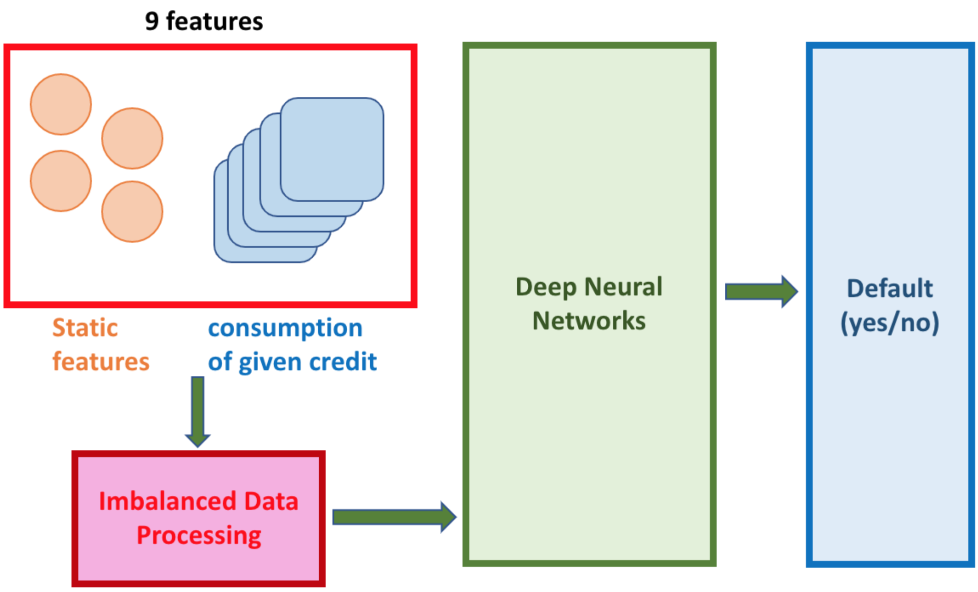
3. Technical Part

▼ Model Implementation:

▼ Following figure is another data preprocessing architecture. The amount of previous payment mean the payment of last month. Therefore, we subtracted amount of bill statement from amount of payment in same month and get amount of underpayment. Moreover, we divided amount of underpayment by amount of given credit to get the consumption of given credit each month.



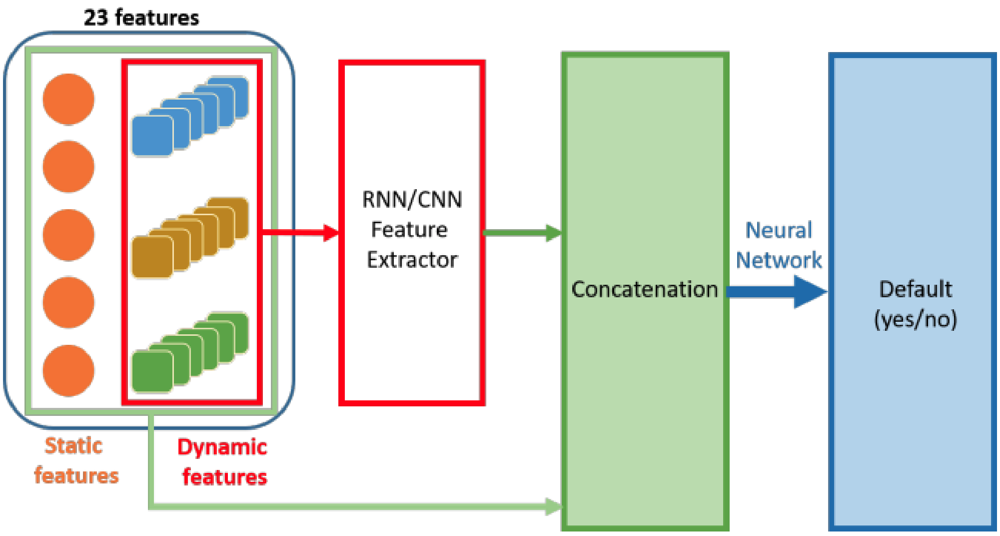
Then, We concatenated five consumptions of given credit with static features such as amount of given credit, gender, education, marriage and age. After normalization and imbalanced data processing, we passed them into a fully-connected neural network for the binary classification. The structure of model is shown in the following figure:



▼ Hyperparameters Setting

- node: [9, 36, 108, 2]
- number of layers = 3
- EPOCH = 1000
- learning rate = 0.0001
- batch_size = 256
- Activation Function = ReLU

▼ Following the second part, we know that some of dynamic features are time-correlated. Thus, we would like to use RNN/LSTM/1D-CNN to extract the latent from 3 dynamic features first. Then, we concatenate the 6 static features with the latent together. In the end, we passed them into a fully-connected neural network for the binary classification. The structure of proposed model is shown in the following figure:



▼ Time-correlated Features Extraction (LSTM)

▼ We arrange the 3 dynamic features in the following format. Then, we set time-steps = 4 months. Later, we extract latent features using LSTM cell. After extracting the latent features, we then concatenate with the static features and feed into a fully-connected DNN to acquire the predicted results.

▼ Hyperparameters Setting for LSTM Extraction:

- input dimension = 1
- hidden dimension = 512
- number of layers = 2
- EPOCH = 1000
- learning rate = 0.00001

▼ Loss function for LSTM Extraction

- We use nn.BCEWithLogitsLoss from pytorch
- This loss combines a Sigmoid later and the BCELoss into one single class
- Besides, we also pass in tensor pos_weight = [4] into the loss function to solve the imbalance data problem. This approach would simply

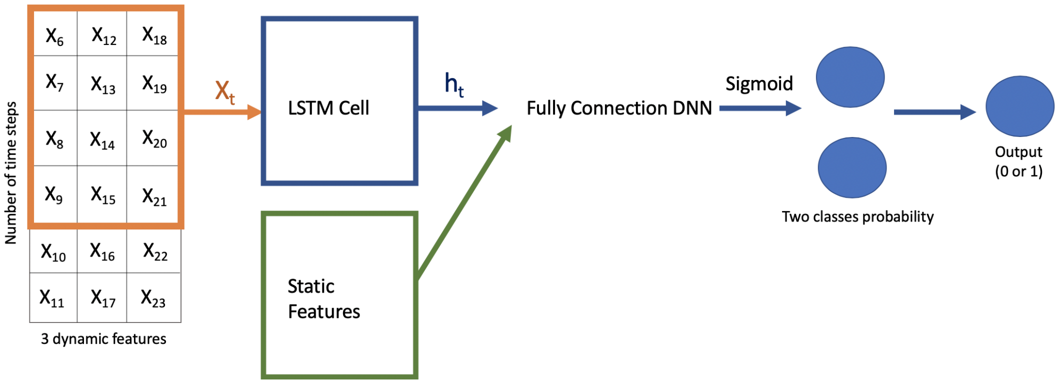
adjust the weight, or you can say the gradient, of the model to put more weight on the insufficient label.

$$l(x,y) = L = \{l_1, l_2, l_3 ... l_N\}^T,$$

$$l_n = -w_n[y_n * log\sigma(x_n) + (1 - y_n) * log(1 - \sigma(x_n)]$$

$$l(x,y) = mean(L)$$

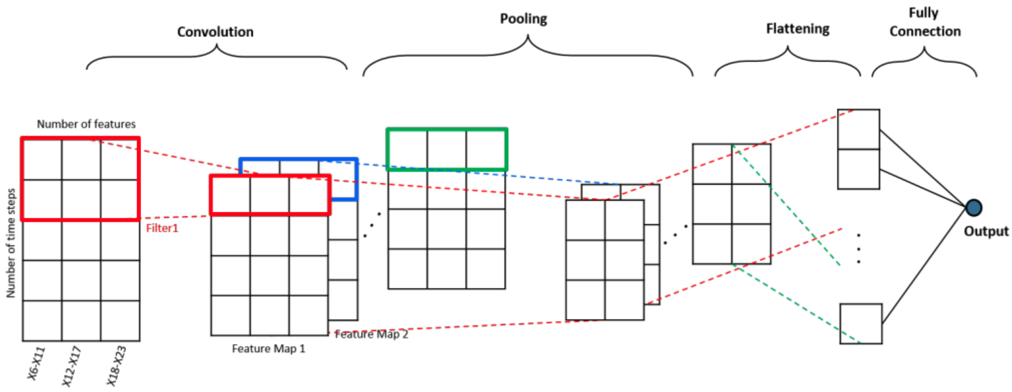
▼ The diagram of using LSTM to extract latent from dynamic features



▼ Time-correlated Features Extraction (1D-CNN)

▼ The following is the architecture of using 1D-CNN as Time-correlated Features Extraction. We use the following hyperparameters to extract time-correlated features from the dynamic features. After the extraction, we concatenate with the static features and feed into a DNN, just as the model architecture stated above.

- Conv1d (in feature = 3, out feature = 16, kernel size = 2)
- Conv1d (in feature = 16, out feature = 32, kernel size = 4)
- batch size = 128
- EPOCH = 1000
- learning rate = 0.0001
- Activation Function = ReLU



4. Experiment

▼ The traditional prediction, such as logistic regression, can be improved by using our data-preprocessing method and time-correlated features extraction model of LSTM+DNN.

Training Results

Model	Accuracy	Precision	Recall	F1-Score	AUC
Logistic Regression	0.63	0.31	0.57	0.41	0.64
Vanilla DNN	0.65	0.59	0.45	0.51	0.61
1D-CNN+DNN	0.65	0.31	0.45	0.36	0.58
LSTM+DNN	0.84	0.86	0.87	0.86	0.78

Testing Results

Model	Accuracy	Precision	Recall	F1-Score	AUC
Logistic Regression	0.62	0.31	0.56	0.40	0.62
Vanilla DNN	0.64	0.60	0.44	0.50	0.61
1D-CNN+DNN	0.66	0.30	0.44	0.36	0.58
LSTM+DNN	0.79	0.84	0.72	0.77	0.79

5. Reference

[1] A. F. Atiya. Bankrupcy prediction for credit risk using neural networks: A survey and new results. IEEE Transactions on neural networks, 12(4):929–935, 2001.

[2] L. C. Thomas J. N. Crook, D. B. Edelman. Recent developments in consumer credit risk assessment. European Journal of Operational Research, 183(3):1447–1465, 2007.

[3] B. Hassani D. Guegan, P. Addo. Credit risk analysis using machine and deep learning models. Risks, 6(2):38, 2018.

[4] Florentin Butaru, Qingqing Chen, Brian Clark, Sanmay Das, Andrew Lo, and Akhtar Siddique. Risk and risk management in the credit card industry. Journal of Banking Finance, 72:218–239, 2016.

[5] Hsinchun Chen Chia-Jung Hsu Wun-Hwa Chen and Soushan Wu Huang, Zan. Credit rating analysis with support vector machines and neural networks: A market comparative study. Decision Support Systems, 37:543–558, 2004.

[6] Adlar J. Kim Khandani, Amir E. and Andrew W. Lo. Consumer credit-risk models via machine-learning algorithms. Journal of Banking and Finance, 34:2767–2787, 2010.

[7] Cuicui Luo, Desheng Wu, and Dexiang Wu. A deep learning approach for credit scoring using credit default swaps. Engineering Applications of Artificial Intelligence, 65, 2016.

[8] F. Shen J. Chao and J. Zhao. Forecasting exchange rate with deep belief networks.

[9] T. Liu C.M. Kuan. Forecasting exchange rates using feedforward and recurrent neural networks. Journal of Applied Econometrics, 10:347–364, 1995.

[10] David A. Hsieh. Testing for nonlinear dependence in daily foreign exchange rates. The Journal of Business, 62:339–368, 1989.

[11] R. A. Meese and K. Rogoff. Empirical exchange rate models of the seventies: do they fit out of sample? Journal of International Economics, 14:3–24, 1983.

[12] Ekaterini Panopoulou and Ioannis Souropanis. The role of technical indicators in exchange rate forecasting. Journal of Empirical Finance, 53, 2019.

[13] Richard A.Meese and Kenneth Rogoff. Noisy time series prediction using recurrent neural networks and grammatical inference. Journal of International Economics, 14:3–24, 1983.

[14] A. Mitra A. K. Nag. Forecasting daily foreign exchange rates using genetically optimized neural networks. Journal of Forecasting, 21:501–511, 2002.

[15] H. White and J. Racine. Statistical inference, the bootstrap, and neural-network modeling with application to foreign exchange rates. IEE Transactions on Neural Networks, 12:657–673, 2001.
[16] E.F. Georgopoulos C. Dunis Theofilatos, A. Karathanasopoulos. Forecasting foreign exchange rates with adaptive neural networks using radial-basis functions and particle swarm optimization. European Journal of Operational Research, 225:528–540, 2013.
[17] Peter Zhang and Michael Hu. Neural network forecasting of the british pound/us dollar exchange rate. Omega, 26:495–506, 1998.
2

[18] Thomas Fischer and Christopher Krauss. Deep learning with long short-term memory networks for financial market predictions. European Journal of Operational Research, 270, 2017.
[19] Eunsuk Chong, Chulwoo Han, and Frank Park. Deep learning networks for stock market analysis and prediction: Methodology, data representations, and case studies. Expert Systems with Applications, 83.
[20] Jigar Patel, Sahil Shah, Priyank Thakkar, and Ketan Kotecha. Predicting stock and stock price index movement using trend deterministic data preparation and machine learning techniques. Expert Systems with Applications, 42:259–268, 2015.
[21] Michel Ballings, Dirk Van den Poel, Nathalie Hespeels, and Ruben Gryp. Evaluating multiple classifiers for stock price direction prediction. Expert Systems with Applications, 42, 2015.
[22] Xiaodong Li, Haoran Xie, Li Chen, Jianping Wang, and Xiaotie Deng. News impact on stock price return via sentiment analysis. Knowledge-Based Systems, 69, 2014.
[23] Sreelekshmy Selvin, Vinayakumar R, E. A Gopalakrishnan, Vijay Menon, and Soman Kp. Stock price prediction using lstm, rnn and cnn-sliding window model. pages 1643–1647, 2017.
[24] T.U. Daim E. Guresen, G. Kayakutlu. Using artificial neural network models in stock market index prediction', journal =.
[25] E.W. Saad ; D.V. Prokhorov ; D.C. Wunsch. Comparative study of stock trend prediction using time delay, recurrent and probabilistic neural networks. IEEE, Transactions on Neural Networks and Learning Systems, 9:1456 – 1470, 1998.
[26] V.N. Sastry A.M. Rather, A. Agarwal. Recurrent neural network and a hybrid model for prediction of stock returns. Expert systems with applications, 42, 2015.

[27] L. Wu Y. Zhang. Stock market prediction of sp 500 via combination of improved bco approach and bp neural network. Expert systems with applications, 36:8849–8854, 2009.
[28] A. L. Perrone and G. Basti. A new criterion of nn structure selection for financial forecasting. 6:3898 – 3903 vol.6, 1999.