Riskcare

Predicting mortgage loan delinquency status with neural networks

Ksenia Ponomareva Paul Epstein David Knight
Ksenia.Ponomareva@riskcare.com

Abstract

Monitoring of loan performance and early identification of high-risk consumers aids prevention of loan defaults and is of interest to many banks and investors. For banks especially, timely monitoring ensures regulatory compliance as well as adequately quantifying their risk, accurately calculating their capital and setting aside proper reserves. With this goal in mind, four multi-classification models have been built that take as input characteristics of a loan at inception, as well as information about the first 12 monthly payments, and predict the status of these payments over the next 12-month period.

1 Introduction and motivation

There is an extensive literature on application of machine learning algorithms to credit scoring, where a model determines the relationship between default and loan characteristics. Once built, this model is used to predict the probability of debt being repaid by the obligor. Review and comparison of these types of models are provided in [1]-[3], with [4] focusing on assessing mortgage risk using a deep net and [5] using a convolutional neural network.

As can be seen from these papers, most publications in consumer lending applications cover binary classification, where a loan either defaults or it does not. Sometimes severe delinquency, defined as having payments delayed by six months or more, is used as an alternative to the default. There are several reasons for considering a multi-classification problem instead. Firstly, a lot of information is lost if only default and non-default states are taken into account. For example, a loan with no missed payments has a different creditworthiness profile to a loan where payments are late by one month only but have occured several times during the initial term of the loan. Slightly late (between 30 and 180 days) payments can be a sign of declining financial health of the obligor. However, since the loan has not technically defaulted, it will be marked as "good" by the binary classification and remain unnoticed until the delinquency status deteriorates significantly. Differentiating between the frequency of the occurrence and the delay magnitude of the payments would allow more accurate tailoring of credit terms and conditions as well as enable early-warning signal detection - since borrowers are likely to transition through different credit profiles throughout the life of the loan.

The aim of this paper is to investigate the technical feasibility of predicting several classes, or statuses, of loan near-term delinquency and payment behaviour by considering a number of multi-classification models. In this paper residential mortgage loan data has been used, but model architectures described here can be applied to other types of loans as well - provided a suitable dataset is available for training.

The rest of the paper is organised as follows. Section 2 provides details on the construction of the final dataset and feature analysis. Section 3 specifies model architectures and provides high-level implementation details. Section 4 provides summary and analysis of results. Conclusions are drawn in Section 5.

2 Dataset and features

2.1 Dataset construction

For this paper, publicly available Fannie Mae single-family loan performance data [6] has been used. The population considered contains a subset of Fannie Mae's 30-year, fully amortizing, full documentation, single-family, conventional fixed-rate mortgages. This data provides information about mortgage loans issued between 2000 and 2016 and their performance, e.g. late payments in months, up to the fourth quarter of 2017. The original mortgage rate for each example has been scaled by the average 30-year federal funds rate for the corresponding quarter to enable comparison of mortgage loans issued over this 17-year period. Only entries with complete information were used and records where loans were terminated prior to month 24 were discarded.

The input used in this paper has 20 features. The first eight features have been selected from the numeric values in the available 24 fields characterising the loan: the original rate, the original amount, the original loan-to-value (LTV), the number of borrowers, the debt-to-income (DTI) ratio, the credit score of the borrower, the first three digits of the zip code and the mortgage insurance percentage. These features have been normalised to speed up convergence of the gradient descent and accelerate training. Please note that this static information is as of the loan initiation and is not updated throughout the considered two-year period of the loan term. Additionally, statuses of each monthly payment for the first 12 months of the loan term are utilised as the final 12 input features. This loan performance information in the Fannie Mae dataset indicates the number of days, represented in months, the obligor is delinquent as determined by the governing mortgage documents. Here 0, for example, represents a loan that is current or is less than 30 days past due and the sequence continues thereafter for every 30-day period.

The ground truth, or the true label, output has six classes with values ranging from zero to five and is based on the actual loan performance over the months 13 to 24. This loan performance information has been analysed both for the maximum number of days that payments were late and how frequently payments were late in the dedicated period. Table 1 provides a definition for each class considering 12 consecutive monthly payments over the months 13 to 24. Each class value is converted into a six-dimensional one-hot vector, where the class i vector would have zeros everywhere apart from the i^{th} value, which is equal to one.

Class	Description
0	All monthly payments are made on time or less than 30 days past due.
1	Only one monthly payment is more than 30 and less than 60 days late.
2	More than one monthly payment is more than 30 and less than 60 days late.
3	Some monthly payments are more than 60 and less than 90 days late.
4	Some monthly payments are more than 90 and less than 180 days late.
5	Some monthly payments are at least 180 days late.

Table 1: Class definitions.

Data is randomly shuffled and then split in such a way that 90% is used for training, 5% for cross-validation and 5% for testing. In the original dataset of over 5 million data points $\approx$ 93% of entries belong to class 0. Please see Figure 1 for the class distribution histogram. This finding is as expected, since most people would make payments on time in the first two years of their mortgage term. However, this dominance of class 0 poses some challenges for training, such as model would tend to predict this class all the time to achieve higher accuracy. Approaches for dealing with training an imbalanced dataset, as applied in this paper, are discussed further in section 3.5.

It is worth noting that in order to predict loan payment delinquency status it is possible to use only the first eight features discussed above. However, adding information about the payments for the first 12 months of the loan term allows comparison between very different model types and architectures.

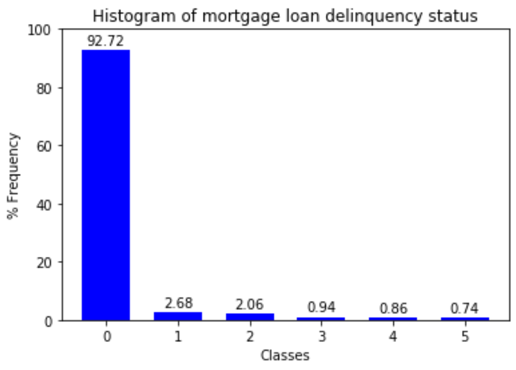


Figure 1: Class frequency distribution

2.2 Training dataset analysis

Getting to know data before starting work on building a solution is always recommended, hence, in this section training dataset is analysed first. Before attempting to train a model, it should be first determined whether or not there is a relationship between the variables of interest. There are 20 dimensions representing different features for each loan. In order to determine if there is any relationship between these loan characteristics and the future delinquency status, and to visualise structure of this data, it would be helpful to reduce the dimension to 2D.

In this paper, a t-distributed Stochastic Neighbour Embedding (t-SNE) technique is used to represent each high-dimensional data point by a point on a two-dimensional grid (please see [7] for details). These two-dimensional points tend to be close whenever the corresponding high-dimensional data points are close. This results in clusters in the two-dimensional t-SNE scatter plot forming wherever clusters are present in the original high-dimensional data. Note that PCA could also have been used for analysis, however this method does not generally handle outliers or skewed distributions as well as t-SNE.

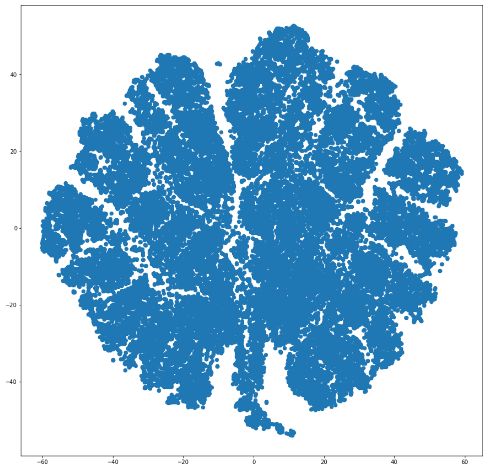


Figure 2: Sampled data

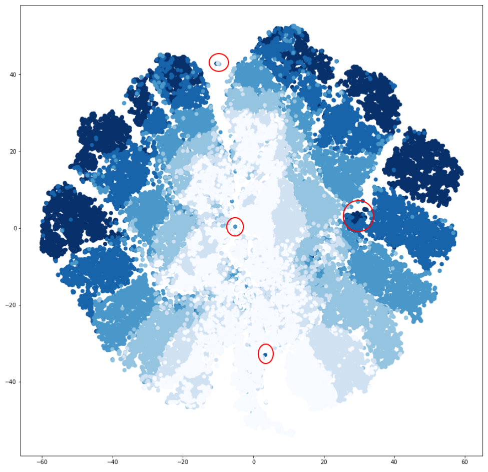


Figure 3: Sampled data colour coded by class

Ten thousand examples are randomly selected from each class in the training data set and the resulting t-SNE scatter plot is shown in the Figure 2, where each point on this graph represents a loan. Figure 3 shows the same data, however each point is coloured depending on its class, with white corresponding to class 0 and the darker the colour the higher the class. It can be observed that there are obvious groupings, although boundaries between clusters are not very clearly defined. It can be deduced that there is a relationship between loan features and class, although some of these can be very intermingled. It can also be observed that there is symmetry along the x axis and a feather-like structure spanning from the centre of the scatter plot. If colour coding is applied by a certain feature, i.e. number of borrowers, mortgage insurance percentage and original LTV, the symmetry and shape can be explained as seen in Figures 4-6.

T-SNE can also be used for data anomaly detection if the points are coloured by each feature in turn, as in [8]. Here, some examples are circled in red in Figures 3 and 6. These points could either be true anomalies, for example, unexpected personal circumstances can prevent timely mortgage payments (dark points in the middle white section of Figure 3), or can be data-entry errors (unusually low LTV values in Figure 6). Further investigation will be required to determine the true nature of these abnormalities.

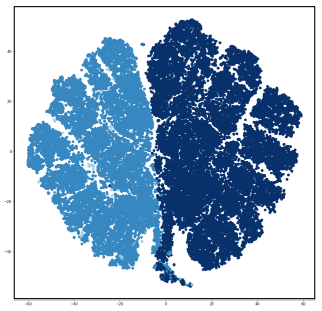


Figure 4: Number of borrowers

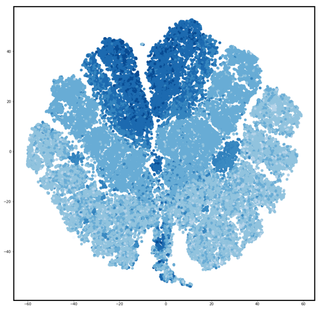


Figure 5: Mortgage insurance percentage

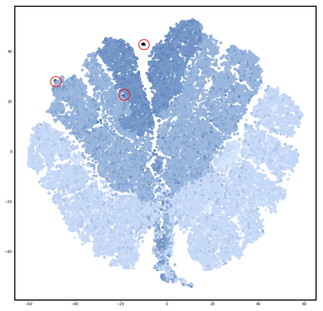


Figure 6: Original loan-to-value ratio

Now that a relationship between loan characteristics and performance and future delinquency status has been confirmed, modelling solutions can be investigated.

3 Methods

Different model architectures are considered for the mortgage payment delinquency status prediction problem. The definition of the problem and the dataset available enables comparison of four different multi-classification model architectures, which have been implemented using TensorFlow, [9]. These are a baseline model, a deep neural network, a recurrent neural network (RNN) and a one-dimensional convolutional neural network (CNN). Figures 7-10 show high-level architecture for each model. Prior to providing further model characteristics, computational details - common for all models - are discussed.

3.1 Input and output

Each model, apart from RNN, has an input, X, with 20 features (eight loan characteristics at inception and performance of the first 12 monthly payments) and an output prediction, $\hat{Y}$, probability distribution over the six classes, 0 to 5.

3.2 Softmax and cross-entropy loss

Unlike a binary prediction of whether a loan would default or not, predicting the delinquency status of a loan - in other words, mapping to one of the six classes defined in Table 1 - is a multi-classification task. Hence a softmax function is used as the activation in the output layer. The softmax function takes a multi-dimensional vector and transforms it into a vector of real numbers, $p_i \in (0, 1)$, which add up to 1. It is computed as follows

$$softmax(x_i) = p_i = \frac{e^{x_i}}{\sum_j e^{x_j}}.$$

Since the softmax function outputs a probability distribution over six classes, it is necessary to apply $argmax$, which returns an index of the largest element in an array, to determine which class has the

highest probability.
Cross-entropy loss is used as a loss function. Cross entropy indicates the distance between what the model believes the output distribution should be and what the actual distribution is. The softmax final activation function and cross-entropy loss function are used for all four models in this paper.

3.3 Metrics

In order to evaluate and compare performance of the four models implemented in this paper, several metrics are considered. These are recall, precision, F1 score and Gini impurity index. Since the dataset is highly dominated by class 0, overall accuracy will not provide enough information on how well models predict classes 1-5 and in some cases it can even be misleading (e.g. 95% accuracy due to 95% of the data belonging to one class).

First, precision and recall for each class are computed, where:

$$precision_i = \frac{TP_i}{TP_i + FP_i},\ recall_i = \frac{TP_i}{TP_i + FN_i}.$$

Here, for a class i, TP_i stands for true positives (both prediction and true label belong to class i), FP_i stands for false positives (prediction is for class i but true label is class j, $i \neq j$) and FN_i means false negatives (prediction is for class j and true label is class i, $i \neq j$). Precision is a good measure to determine when the costs of a FP are high. For example, incorrectly identifying an obligor as one with deteriorating creditworthiness and imposing stricter credit terms or even terminating the loan, will result in losing a valuable customer and is best to be avoided. Recall, on the other hand, is a useful metric when costs of a FN are high. For example, an obligor who is getting behind on monthly payments and remains undetected as he/she is not assigned to an appropriate deliquency class, can later default and cause more issues for the loan originator.

F1 score is calculated as the harmonic mean of precision and recall, as follows:

$$F1_i = 2 \times \frac{precision_i \times recall_i}{precision_i + recall_i}.$$

F1 score is a good metric when a balance between precision and recall is needed and there is an uneven class distribution. In order to compute one score for each model, F1 values for each class can either be averaged to obtain a macro F1 score or a micro F1 score, based on global precision, $\frac{TP_0 + ... + TP_5}{TP_0 + ... + TP_5 + FP_0 + ... + FP_5}$, and recall, $\frac{TP_0 + ... + TP_5}{TP_0 + ... + TP_5 + FN_0 + ... + FN_5}$, can be used instead. Since F1 scores for each class are provided separately, micro F1 score per model is chosen for this paper.

The final metric provides one number per model and is compared to the dataset used for training. Gini impurity index for six classes is calculated as:

$$G = 1 - \sum_{i=0}^{5} (p_i)^2,$$

where, for each i, p_i is the number of times that class appears in the dataset (true or predicted) divided by the dataset size. The smaller the value of the impurity G, the more skewed is the class distribution. This metric is only used for the training set, where, due to the balanced mini-batch approach (see section 3.4.1), the true label classes are equally represented and have $G = 1 - 6(\frac{1}{6})^2 = 83.33\%$.

3.4 Working with an imbalanced dataset

Imbalanced datasets, where classes are not represented equally, are common in practice. As discussed in section 2.1, it is expected that the vast majority of payments over the first two years of the mortgage term would be made on time. In the paper two of many possible solutions proposed in [10] have been implemented and discussed further in the next sections.

3.4.1 Balanced mini-batches

In order to ensure that a balanced dataset is used for training, one can either add repeated instances of the under-represented samples or delete some instances of the over-represented class. It is not usually desirable to delete any of the data and adding repeated class instances might have significant memory restrictions if the original dataset is already large. In this paper, a balanced mini-batch selection method is used, where an equal number of each class is sampled at each step and is iterated through for training. It is worth noting that over-sampling can result in overfitting to the small training dataset available for certain classes and relevant measures, such as regularization, would need to be applied to reduce overfitting.

3.4.2 Weighted loss function

Given class imbalances in the training data, as shown in Figure 1, the model would tend to predict everything as class 0 to achieve high accuracy. In order to address this issue, the loss function needs to be re-weighted to apply higher penalties for making wrong predictions for classes 1 to 5 compared to the dominant class 0. Median-frequency re-weighting, introduced in [11], is applied here and is calculated as follows:

$$\alpha_c = median\_freq/freq(c).$$

Here $freq(c)$ is the number of examples of class c present in the mini-batch divided by the total number of examples in the mini-batch and $median\_freq$ is the median of all class frequencies in the mini-batch. For all models in this paper, in the training procedure, each example's softmax cross-entropy loss is multiplied by the coefficient, α_c, according to the label's class, c. Therefore the dominant labels will be assigned the lowest weight which balances out the training process.

3.5 Models

3.5.1 Baseline

Baseline is the simplest neural network architecture considered here and has a single node, as shown in Figure 7. A linear transformation is applied to the input, X, and a softmax activation function is used to produce a prediction, $\hat{Y}$:

$$Z = WX + b,$$
$$\hat{Y} = softmax(Z).$$

This model can be considered as a multi-classification analogue of a logistic regression.

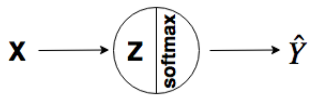


Figure 7: Baseline model architecture

3.5.2 Deep neural network

The deep neural network (DeepNN) model has five hidden layers, see Figure 8, with 200, 150, 100, 50 and 10 nodes respectively. All activations apart from the final one are ReLU, where $ReLU(z) = max(0, z)$. By introducing several hidden layers, the model is expected to capture relationships between features and classes better and hence provide some improvement to the baseline results.

3.5.3 Convolutional neural network

The convolutional neural network (CNN) model uses a one dimensional structure inspired by [5] and has two blocks of convolutional and max pooling layers, followed by a fully connected layer, as shown in Figure 9. Activation functions for layers other than final are ReLU. Input is 1x20 and output is 1x7 based on the softmax final activation function. The first block has one convolutional layer with 32 filters with size 1x5, stride 1 and same padding, then max pooling layer with filter size 1x3, stride 2 and same padding. The second block has a convolutional layer with 64 filters with size 1x3, stride 1 and same padding, then max pooling layer with filter size 1x2, stride 2 and same padding.

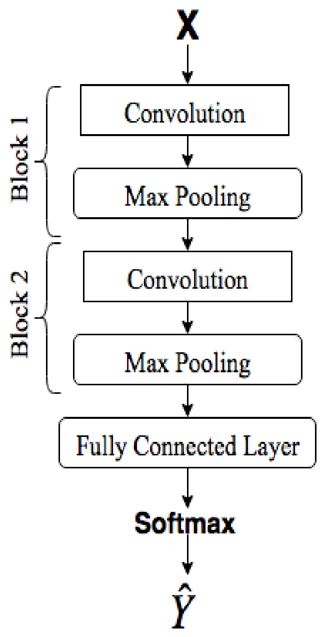


Figure 8: CNN model architecture

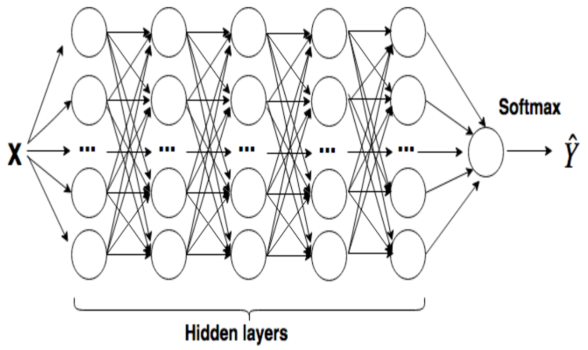


Figure 9: DeepNN model architecture

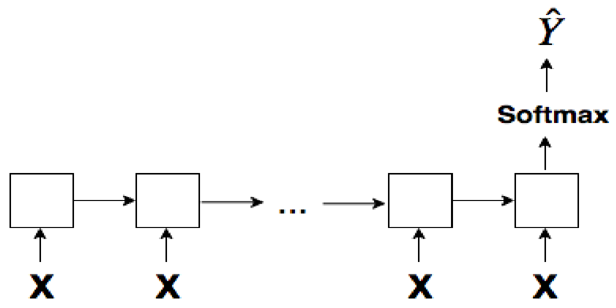


Figure 10: RNN model architecture

3.5.4 Recurrent neural network

This model is a many-to-one one-directional Long-Short Term Memory (LSTM) RNN with a softmax activation applied to the last output only, as shown on Figure 10. The input sequence has 12 items, one for each month of the observable monthly mortgage payment performance in the first year of the loan. Each of the inputs has nine features, eight loan characteristics, which are the same for each input, and one monthly payment performance for the relevant month. The LSTM hidden layer has 200 units. This model is one directional since the order of loan monthly payment statuses is very important. Many-to-one architecture is used since prediction is needed for the payment status over the months 13 to 24 of the loan term and not for each consecutive month in the first year.

4 Results

Models have been iterated over various hyperparameters and network structures to provide the best test accuracy. Batch size 36, Adam optimizer, Xavier weight initialisation have been used for all four models that were trained over ten epochs. Dropouts have been used for each layer in DeepNN and RNN models to prevent over-fitting. Results are provided in Tables 2-3 and Figures 11-12.

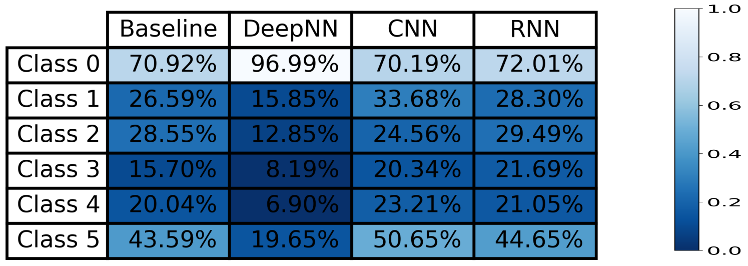
G	Baseline	DeepNN	CNN	RNN	True Label
Training dataset	81.26%	81.91%	82.07%	83.29%	83.33%

Table 2: Gini impurity index for all models.

F1	Baseline	DeepNN	CNN	RNN
Test dataset	67.74%	90.82%	67.82%	68.81%

Table 3: Micro F1 score for all models.

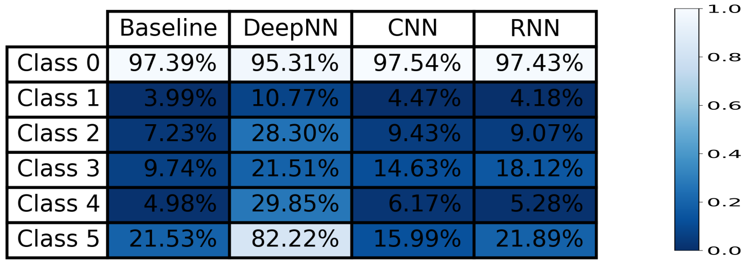
Examining results, a number of observations can be made. First of all, compared to the other models considered in this paper, RNN has the closest Gini impurity index to the true label training dataset.



	Baseline	DeepNN	CNN	RNN
Class 0	70.92%	96.99%	70.19%	72.01%
Class 1	26.59%	15.85%	33.68%	28.30%
Class 2	28.55%	12.85%	24.56%	29.49%
Class 3	15.70%	8.19%	20.34%	21.69%
Class 4	20.04%	6.90%	23.21%	21.05%
Class 5	43.59%	19.65%	50.65%	44.65%

1.0
0.8
0.6
0.4
0.2
0.0

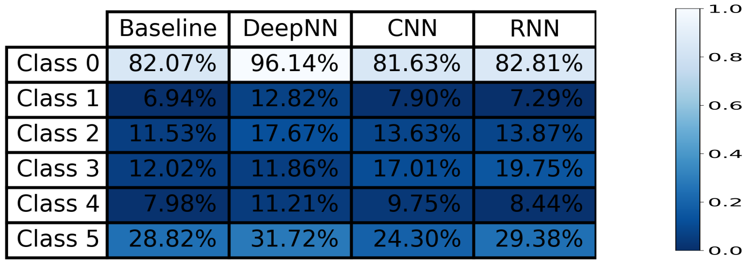
Figure 11: Recall for the test dataset



	Baseline	DeepNN	CNN	RNN
Class 0	97.39%	95.31%	97.54%	97.43%
Class 1	3.99%	10.77%	4.47%	4.18%
Class 2	7.23%	28.30%	9.43%	9.07%
Class 3	9.74%	21.51%	14.63%	18.12%
Class 4	4.98%	29.85%	6.17%	5.28%
Class 5	21.53%	82.22%	15.99%	21.89%

1.0
0.8
0.6
0.4
0.2
0.0

Figure 12: Precision for the test dataset



	Baseline	DeepNN	CNN	RNN
Class 0	82.07%	96.14%	81.63%	82.81%
Class 1	6.94%	12.82%	7.90%	7.29%
Class 2	11.53%	17.67%	13.63%	13.87%
Class 3	12.02%	11.86%	17.01%	19.75%
Class 4	7.98%	11.21%	9.75%	8.44%
Class 5	28.82%	31.72%	24.30%	29.38%

1.0
0.8
0.6
0.4
0.2
0.0

Figure 13: F1 scores for the test dataset.

Although Gini index does not provide information about how accurately class allocation is done, it shows that both RNN and CNN have a more balanced class distribution compared to the Baseline and DeepNN models, which is also reflected in training set accuracy results. The second observation is that, based on the micro F1 score alone, DeepNN, CNN and RNN models show an improvement over the Baseline, with DeepNN outperforming the other three models.

Finally, DeepNN, CNN and RNN models also show improvement over the Baseline for individual class recall, precision and F1 values. From Figure 11 it can be observed that the DeepNN model has the highest recall value for the class 0, however for classes 1-5 CNN and RNN alternate to present the highest and the second highest recall values. Whereas, Figure 12 shows that the DeepNN model significantly outperforms the other three models for precision for classes 1-5. It has a particularly high precision for class 5, 82.22%, which is impressive considering that this class has the least amount of data compared to classes 0-4. Since F1 score is a harmonic mean of recall and precision, the DeepNN model has the highest F1 values for classes 0-2 and 4-5, whereas RNN has the highest score for class 3. Overall, if precision for classes 1-5 or F1 scores are more important, DeepNN is the best model out of the four described in this paper, and if recall for classes 1-5 is important, then either CNN or RNN would be a more suitable choice.

5 Conclusion

In this paper, four multi-classification models have been built to take as input characteristics of a loan at inception, as well as information about the first 12 monthly payments, and predict the statuses of these payments over the next 12-month period. The results for all four models have been presented and recommendations for usage of each model provided, depending on the importance of each type of metric. Unlike image classification, where all the information is contained in the pixels of the image, the loan delinquency classification problem has a lot of extraneous factors (from personal circumstances to economic environment), which are not captured in the loan application details and can affect timeliness of obligor's monthly payments. Although DeepNN, CNN and RNN models outperform the Baseline, they can be further enhanced by considering more than the first 12 payments as features. Another approach could be supplementing with current/savings/credit account data as in [5] or with county-level unemployment rates, zip-code level housing prices, and lagged foreclosure and prepayment rates in zip-code as in [4] to help improve models' performance.

References

[1] García, V., Marqués, A. I., & Sánchez, J. S. (2014). An insight into the experimental design for credit risk and corporate bankruptcy prediction systems. *Journal of Intelligent Information Systems*, **44** (1), 159–189.

[2] Lessmann, S., Baesens, B., Seow, H.-V., & Thomas, L. C. (2015). Benchmarking state- of-the-art classification algorithms for credit scoring: An update of research. *European Journal of Operational Research*, **247** (1), 124–136.

[3] Yeh, I. & Lien, C., (2009) The comparisons of data mining techniques for the predictive accuracy of probability of default of credit card clients, *Elsevier Expert Systems with Applications*, **36**, 2473–2480.

[4] Sirignano, J., Sadhwani, A., & Giesecke, K. (2016). Deep Learning for Mortgage Risk. ArXiv e-prints http://adsabs.harvard.edu/abs/2016arXiv160702470S.

[5] Kvamme, H., Sellereite, N., Aas, K. & Sjursen, S. (2018) Predicting mortgage default using convolutional neural networks, *Elsevier Expert Systems with Applications*, **102**, 207-217.

[6] https://loanperformancedata.fanniemae.com/lppub/index.html.

[7] Laurens van der Maaten, Geoffrey Hinton Journal of Machine Learning Research 9 (2008) 2579-2605, Visualizing Data using t-SNE, http://www.jmlr.org/papers/volume9/vandermaaten08a/vandermaaten08a.pdf.

[8] https://medium.com/@Zelros/anomaly-detection-with-t-sne-211857b1cd00.

[9] https://www.tensorflow.org.

[10] https://machinelearningmastery.com/tactics-to-combat-imbalanced-classes-in-your-machine-learning-dataset.

[11] Eigen, D. & Fergus, R., (2015) Predicting depth, surface normals and semantic labels with a common multi-scale convolutional architecture, *arXiv:1411.4734v4*.