Delft University of Technology

Data Science for Finance
WI3435TU

Assignment: Part IV. Neural Networks

Authors:
Dinu Gafton
Dan Sochirca

January 29, 2023

Question 1. The features selected for the classification task

By now, we have the 10 most important features found when we used logistic regression, and the 10 most important features found when we trained a random forest model (from the variable importance diagram).

This means we have **3 options**: using the features selected by logistic regression; using the ones important for the random forest model; a mix of both.

Trying out all three options on our final neural network model (in point 3 of this assignment) resulted in similar results, with the first one performing the best, meaning **we will use the most important features found with logistic regression**.

Here are the **10 most important features** we used for our neural network: **sub_grade, term, loan_amnt, dti, fico_range_low, mort_acc, mo_sin_old_rev_tl_op, earliest_cr_line,acc_open_past_24mths, int_rat.**

Question 2. Training a neural network with a single hidden layer

Here are the parameters of the trained model:

- **Architecture:** 1 hidden layer, 10 hidden neurons
- **learning rate:** 0.001
- **no. of epochs:** 500
- **activation function:** sigmoid,
- **L2 regularization term:** 0.01
- **solver:** adam (a stochastic gradient-based optimizer)

The parameters were arbitrarily chosen, some of which were kept to default values.

As usual, we computed different metrics for different probability thresholds. The results are diplayed in figure 1 below.

METRICS:

	0	1	2	3	4	5
THRESHOLD	0.5	0.6	0.7	0.75	0.8	0.85
accuracy	0.777291	0.770472	0.725175	0.680985	0.606699	0.502265
true pos rate	0.978709	0.923652	0.803057	0.70434	0.567776	0.395063
precision	0.785218	0.806576	0.834136	0.856185	0.879019	0.906081
AUC-score	0.539656	0.589747	0.633289	0.653432	0.652621	0.628744

Figure 1: Different metrics corresponding to the different probability thresholds considered for the single-layer neural network. The highest AUC score is highlighted in yellow.

The best threshold in terms of AUC is **0.75**, with **AUC=0.653432**. The **precision** is **0.856185** and **TPR= 0.70434**. **The ROC curve** and **confusion matrices** for different thresholds can be found in the **appendix**, figure 4.

The simple NN model performs a bit **worse** than our best regularization and tree models, both in terms of AUC and precision (accuracy and TPR are higher). The AUC is in the same 0.65 range as the other models, but the precision is lower compared to lasso, which has it at 0.883.

Question 3. Refining the model

Refining a neural network model is important because it can improve the model's accuracy and performance. We have decided to build multiple models with different architectures (**different number of hidden layers and hidden neurons on each layer**). The results can be seen in Figure 2.

```
ANN with  1 layers, hidden neurons on each level =  (5,)  and accuracy :  0.7775680708641602
ANN with  2 layers, hidden neurons on each level =  (5, 5)  and accuracy :  0.7765614776787961
ANN with  3 layers, hidden neurons on each level =  (5, 5, 5)  and accuracy :  0.7773667522270874
ANN with  1 layers, hidden neurons on each level =  (10,)  and accuracy :  0.7780965322864765
ANN with  2 layers, hidden neurons on each level =  (10, 10)  and accuracy :  0.777517741204892
ANN with  3 layers, hidden neurons on each level =  (10, 10, 10)  and accuracy :  0.7776938950123308
ANN with  1 layers, hidden neurons on each level =  (20,)  and accuracy :  0.7782475212642811
ANN with  2 layers, hidden neurons on each level =  (20, 20)  and accuracy :  0.7777945543308672
ANN with  3 layers, hidden neurons on each level =  (20, 20, 20)  and accuracy :  0.7782475212642811
ANN with  1 layers, hidden neurons on each level =  (30,)  and accuracy :  0.7776184005234285
ANN with  2 layers, hidden neurons on each level =  (30, 30)  and accuracy :  0.7782978509235492
ANN with  3 layers, hidden neurons on each level =  (30, 30, 30)  and accuracy :  0.7780210377975741
```

Figure 2: Performance of multiple NN models with different architectures

We have observed that all the models with 2 hidden layers have a lower accuracy than their counterparts with 1 or 3 hidden layers, but the models with 3 hidden layers have a similar or worse performance than those with 1 hidden layer. Thus, we chose the model highlighted in the picture (**1 hidden layer with 20 hidden neurons**).
Since our models are trained just on 10 features (the most important ones), a big number of hidden neurons is not needed. We have also built models with *hidden neurons* > 100 for each hidden layer, but those models brought very similar results, so we decided to work with smaller and simpler architectures that have the same performance.

The best NN model

As usual, to find the best model we need to find the probability threshold that maximizes AUC. We did so, for our refined NN model with a single hidden layer and 20 neurons, and it appears to be **0.75**, with **AUC=0.655294**.

Again, we computed different metrics and displayed them in figure 3 below. The ROC curve can be found in the **appendix**, figure 5.

```
Confusion matrix of the best model:
[[0.14562887 0.08374855]
 [0.24991192 0.52071065]]
```

	accuracy	true pos rate	precision	AUC-score
0	0.66634	0.675701	0.861449	0.655294

Figure 3: Different metrics computed for our best NN model with a probability threshold of 0.75.

The NN model performs **very** similarly to our best tree model, with the tree model slightly outperforming it by roughly 0.001-0.01 in all metrics categories. Our logistic regression model still is the best out of the three.

Therefore, we conclude that **the best NN model is with the following architecture and parameters**: single hidden layer, 20 hidden neurons, learning rate of 0.001, 500 epochs, sigmoid activation function, L2 regularization term of 0.001, and a **probability threshold of 0.75**.

Appendix

```
Confusion matrix for threshold = 0.5
 [[0.02307615 0.20630127]
 [0.01640747 0.75421511]]

Confusion matrix for threshold = 0.6
 [[0.05868438 0.17069304]
 [0.05883537 0.71178721]]

Confusion matrix for threshold = 0.7
 [[0.10632141 0.12305602]
 [0.15176909 0.61885349]]

Confusion matrix for threshold = 0.75
 [[0.13820524 0.09117218]
 [0.22784237 0.54278021]]

Confusion matrix for threshold = 0.8
 [[0.16915798 0.06021944]
 [0.33308169 0.43754089]]

Confusion matrix for threshold = 0.85
 [[0.19782073 0.0315567 ]
 [0.46617847 0.30444411]]
```

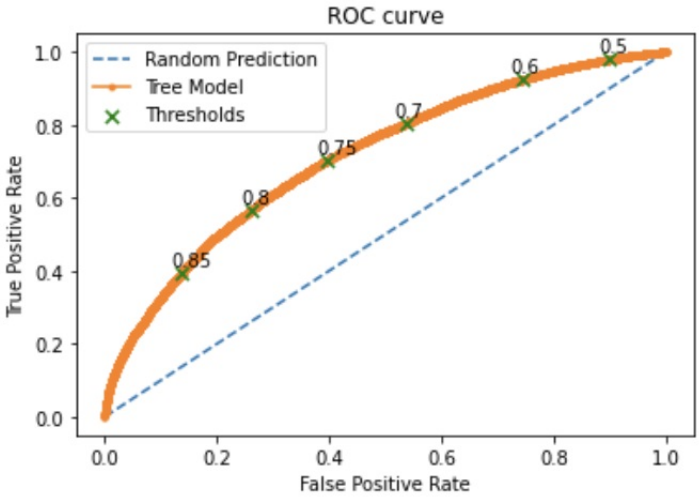

Figure 4: The confusion matrices for different thresholds and ROC curve of the single-layer neural network model for **question 2**.

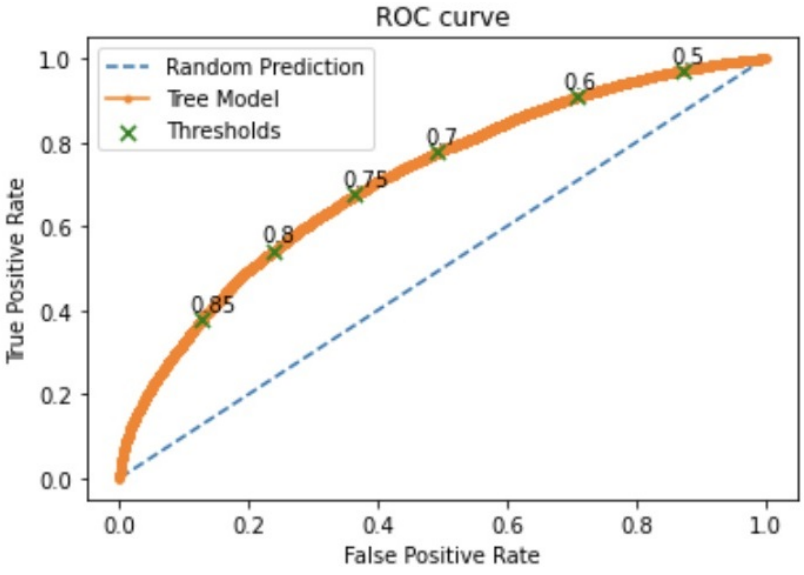

Figure 5: The ROC curve of the best NN model, which is our refined model in question 3 with a probability threshold of 0.75.