INM427 Neural Computing Individual Project

Yumi Heo (Msc Data Science / 230003122 / [yumi.heo@city.ac.uk](mailto:yumi.heo@city.ac.uk))

**A Comparison of Multilayer Perceptrons and Support Vector Machines for Bank Churn Prediction**

**Abstract**

**Description and Motivation**

Analysing customer churn in the bank is important for maintaining profitability, customer satisfaction, and competitiveness against other banks. Therefore, banks must enhance their ability to identify potential customer churn. This can be achieved using a supervised classification model based on neural networks or support vector machines (SVM), using customer and churn data. In this paper, multilayer perceptrons (MLP) and SVM have been utilised to conduct an experiment to find which of the two models better performs in classifying customer churn. By comparing and evaluating the performance of the two models, we will learn about the characteristics of the two algorithms and determine which model is appropriate for this case. Eventually, the best model will enable banks to predict future churn from customer data and proactively improve services to retain their customers.

**Initial Analysis of Dataset**

The dataset is about obtained from Kaggle [1]

165034 rows × 14 columns with the target

**Hypothesis Statement**

**Summary of the two ML methods with their pros and cons**

**Methodology**

**Choice of Parameters and Experimental Results**

**(e.g. cross-validation, choice of parameters and experimental results)**

**Analysis and Critical Evaluation of Results**

**Lessons Learned**

**Future Work**

**Reference**

**[1] kaggle**

Comparison plots: lr curve & accuracy / prevision & recall curve(https://ai-com.tistory.com/entry/ML-%EB%B6%84%EB%A5%98-%EC%84%B1%EB%8A%A5-%EC%A7%80%ED%91%9C-Precision%EC%A0%95%EB%B0%80%EB%8F%84-Recall%EC%9E%AC%ED%98%84%EC%9C%A8)

Precision(정밀도)는 얼마나 정확하게 유저 이탈이라고 예측하는지에 대한 지표입니다. Recall(재현율)은 실제 이탈자에 대해서 얼마나 정확하게 이탈자라고 예측하는지에 대한 지표입니다.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Mlp baseline(1 hidden layer / 6 hidden units) | Test Accuracy: 0. 84.38 | Precision of the best-trained model = 74.32  % | Recall of best-trained model = 48.67  % | F1 score of best-trained model = 58.82  % |
| MLP classifier(1 hidden layer / 6 hidden units) | Best Accuracy: 0.8625098466945403 |  |  |  |
| MLP classifier(2 hidden layer / 9:9 hidden units) | Best Accuracy: 0.8649336484275586  Best Hidden Units: (9, 9) |  |  |  |
| Mlp 2(2 hidden layer / 9:9 hidden units) | Test Accuracy: 0.8606 | Precision of the best-trained model = 73.40% | Recall of best-trained model = 61.06% | F1 score of best-trained model = 66.67% |
| Mlp 2(2 hidden layer / 9:9 hidden units), lr 스케줄러 적용  scheduler = lr\_scheduler.LinearLR(optimizer, start\_factor=0.33, total\_iters=4) | Test Accuracy: 0.8630 | Precision of the best-trained model = 76.74% | Recall of best-trained model = 58.41% | F1 score of best-trained model = 66.33% |
| Mlp 2(2 hidden layer / 9:9 hidden units), lr 스케줄러 적용  scheduler = lr\_scheduler.LinearLR(optimizer, start\_factor=0.33, total\_iters=100) | Test Accuracy: 0.8641 | Precision of the best-trained model = 77.65% | Recall of best-trained model = 58.41% | F1 score of best-trained model = 66.67% |
| Mlp 2(2 hidden layer / 9:9 hidden units), lr 스케줄러 적용  scheduler = lr\_scheduler.LinearLR(optimizer, start\_factor=0.33, total\_iters=100) | Test Accuracy: 0.8616 | Precision of the best-trained model = 75.61% | Recall of best-trained model = 54.87% | F1 score of best-trained model = 63.59% |

lr어떻게 변화하는지 print 뽑기

시그모이드가 왜 output layer에 쓰이는지, tanh에 대한 설명(Activation Functions: Comparison of Trends in Practice and Research for Deep Learning)

텍스트, 라인, 그래프, 스크린샷이(가) 표시된 사진

자동 생성된 설명

텍스트, 스크린샷, 소프트웨어, 멀티미디어 소프트웨어이(가) 표시된 사진

자동 생성된 설명

https://machinelearningmastery.com/data-preparation-without-data-leakage/  
k fold cross validation batches train validation test set 설명

<https://machinelearningmastery.com/learning-curves-for-diagnosing-machine-learning-model-performance/>

loss plot과 오버피팅 설명

pytorch로 모델링을 햇지만 gridsearch를 sklearn 패키지를 이요해서 햇으므로 정확성이 양 패키지 사이에 얼마나 있을지 모르겠음 future work로 냅둬

어차피 아담의 특징이 00이니 러닝레이트 스케줄러는 적용하지 않음

대신 그리드 서치 정도 적용

Support Vector Machine algorithms are not scale invariant, so **it is highly recommended to scale your data**. For example, scale each attribute on the input vector X to [0,1] or [-1,+1], or standardize it to have mean 0 and variance 1. Note that the same scaling must be applied to the test vector to obtain meaningful results. (https://scikit-learn.org/stable/modules/svm.html)