A Comparative Study in Customer Churn Prediction through Multilayer Perceptrons and Support Vector Machines

Morteza Layegh Mirhosseini -- Student ID: 220003166

Abstract:

Companies must find and retain customers who are about to leave their services to avoid the costs of losing customers, build loyal customers, and improve their overall customer experience. In this study, we aimed to develop a Multilayer Perceptron (MLP) model and a Support Vector Machine (SVM) model to predict which customers may leave a credit card company. Our goal was to critically evaluate both models, compare their algorithms, and determine which model is better suited for this task. We used grid search cross-validation to fine-tune the hyperparameters. Our findings suggest that the MLP model takes longer to train but has slightly higher accuracy. Moreover, it performs better in identifying customers who are at risk of leaving as compared to the SVM model.

1. Introduction

Credit card services are an essential revenue stream for banks, but retaining customers can be a significant challenge. Customers may choose to cancel their credit cards for various reasons. It's commonly acknowledged that it's less expensive to keep current customers than to acquire new ones. Therefore, it is essential for banks to identify those who are about to leave their service and proactively target them with different services to try to convince them to stay.

In this report, we are trying to develop and critically evaluate two models: a Support Vector Machine (SVM) and a Multi-Layer Perceptron (MLP), to predict whether a customer of a credit card company will leave the company or not (a binary classification problem). We will try different configurations for both models to find the best configuration for each of them.

2. Summary of the two methods with their pros and cons

In this section we will give a brief description of each method and focus on their pros and cons:

1.1 Multi-Layer Perceptron

Multi-layer perceptrons, also called Artificial Neural Networks, are computational models that were inspired by the structure and functionality of biological neural networks in the brain [1]. They are composed of an input layer, one or more hidden layers, and an output layer, and the neurons in each layer are connected to those in the previous layer. The network aims to learn patterns and relationships in the data through an algorithm called backpropagation, where errors are propagated back through the layers to adjust the weights of the connections.

MLPs are powerful tools for discovering nonlinear relationships in data. They can be adapted for both classification and regression tasks, and are capable of processing various types of data, including images and texts.

However, one of the major drawbacks of MLPs is their lack of interpretability, where the internal workings of the model are often difficult to understand[2]. Additionally, due to the high number of configurations, it can be challenging to find the optimal configuration for the network[3].

1.2 Support vector machines:

Support Vector Machines (SVMs) are a type of machine learning model used for both classification and regression tasks. SVMs separate classes by mapping the data to a higher

dimension using the kernel trick and fitting a hyperplane between the classes. This enables SVMs to handle nonlinear relationships in the data and make the samples linearly separable[3].

Support Vector Machines (SVMs) work really well when the number of dimensions is greater than the number of samples in the dataset, and they are also very efficient in terms of memory usage. However, SVMs do not perform well when classes are overlapped. Additionally, SVMs do not provide probabilistic explanations for the classification results[4].

3. Hypothesis Statement

Based on the results achieved by Sabbeh et al. in their comparative study [5] on a similar problem and dataset, we hypothesize that the MLP model will achieve higher accuracy compared to the final SVM model.

4. Dataset

The dataset utilized in this analysis was obtained from Kaggle and consists of 10127 rows and 23 columns[6]. These columns include variables such as age, marital status, income, and other relevant information.

To preprocess the data, we first inspected the dataset and identified five columns that were irrelevant to the analysis. These columns were removed, leaving us with 18 independent features and one dependent feature (target). Five categorical features were present in the dataset, and these were encoded using the one-hot encoder from the scikit-learn library. It is worth noting that the dataset contained no missing values. However, it was observed that the dataset was imbalanced, with 16.1% of churned customers (class 0 - left the company) and 83.9% of existing customers (class 1- stayed at the company). To address this imbalance, we employed the SMOTE (Synthetic Minority Over-sampling Technique) method[7].

Subsequently, the data set was standardized using the StandardScaler from the scikit-learn library. It is worth mentioning that the last two steps of the preprocessing (balancing and standardizing) were implemented after splitting the dataset into training and testing sets to preserve the integrity of the training set and prevent information leakage.

5. Methods

This section outlines the specifics of how the training, validation, and testing processes were carried out, along with an explanation of the architecture and hyperparameters used in constructing the MLP and SVM models. To ensure the selection of appropriate initial hyperparameters for the models, we referred to prior research that has worked on similar problems and datasets [3, 5, 8].

5.1. Methodology

The dataset was divided into a test set (20%) and a train set (80%)[5]. The train set was used for the training and validation of both SVM and MLP models. To ensure a fair comparison, the same test and train sets were used throughout the entire process for both models.

The process of hyperparameter tuning for the Multilayer Perceptron (MLP) involves a two-step procedure. Firstly, the architecture of the model needs to be designed, which includes identifying the optimal number of hidden layers and their corresponding neurons. To determine the range of values for the number of neurons in the hidden layer for cross-validation, we used rules of thumb[9]. Secondly, appropriate learning parameters need to be selected. Both of these stages require hyperparameter tuning, which was performed using two separate grid searches. The study used a 5-fold cross-validation technique during the grid search to train and validate each model. An early stopping technique was also used to prevent overfitting[10].

Similarly, A 5-fold cross-validated grid search was employed to select optimal parameters for the SVM model. The final models (SVM and MLP) were trained on the entire training set using the best sets of hyperparameters. The performance of the final models was evaluated and compared using a holdout test set

5.2. MLP choice of parameters and experimental results:

The study considered hyperparameters such as the number of hidden layers, the number of neurons in each hidden layer, the learning rate, the dropout rate, and the optimizer's weight decay. The final MLP model used in the study had three fully connected hidden layers with 64, 32, and 16 neurons, respectively. The ReLU activation function was applied to each neuron to capture the nonlinear relationship between the input and output.

To prevent overfitting, a dropout rate of 0.3 was applied to each layer's output. The study tackled a binary classification problem, so a sigmoid activation function was chosen for the output layer's one neuron to provide class probability. The binary cross-entropy loss function was used to calculate the error between the predicted and true values of each input sample. To optimize the model, the Adam optimizer was used with a learning rate of 0.001. Furthermore, early stopping was employed as a callback mechanism to prevent overfitting by terminating training if the validation loss fails to improve over 10 epochs.

5.3. SVM choice of parameters and experimental results:

In this study, we performed hyperparameter tuning for the Support Vector Machines (SVM) model, focusing on two crucial hyperparameters: the box constraint value and the gamma parameter of the RBF kernel. The kernel determines the type of transformation used to map the input data to a higher-dimensional space where a linear decision boundary can be found. Based on the study by Sabbeh et al.[5], we selected the radial basis function (RBF) kernel for our model. After experimentation, we selected a gamma value of 0.1, which determines how tightly the decision boundary fits around the data points.

We tuned the box constraint value, which controls the trade-off between achieving a smaller margin and allowing more training examples to violate the margin. After the grid search, we selected the value of 10 for the box constraint.

6. Analysis and critical evaluation of results :

Firstly, we will focus on the process of selecting hyperparameters and the outcomes of the grid search conducted for both MLP and SVM models.

In order to determine the optimal number of hidden layers, we conducted three experiments while keeping the other parameters constant. We fixed the learning rate at 0.01 and the number of epochs at 150 with early stopping, and set 32 neurons in each layer. The results indicated that three layers achieved the best performance(Figure 1).

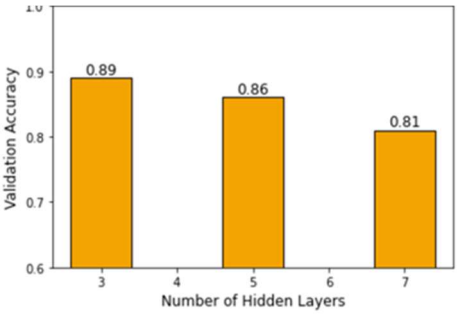

Figure 1 Validation Accuracy vs Number of Hidden Layers

We conducted a grid search to find the best number of neurons in the hidden layers. The results showed that changes in the number of neurons did not have a significant impact on the model's accuracy(Figure 2). Therefore, it is recommended to use a simpler architecture with 16-16-16 neurons. However, for the purposes of exploring the model's capabilities, We chose a more complex architecture with 64-32-16 neurons, which resulted in the highest validation accuracy.

Figure 3 shows the results of a grid search performed on the learning parameters(learning rate, dropout rate, and weight decay) after selecting the optimal architecture. The graph reveals a pattern where a decrease in accuracy is observed when the weight decay is set to 0.01 for the Adam optimizer, indicating that this penalty term's magnitude is too large. This leads to underfitting of the model, where the model is too simple to capture the complexity of the data. Our final model was selected based on these findings, using a learning rate of 0.001 and a dropout rate of 0.3 at each layer, with the weight decay parameter set to zero.

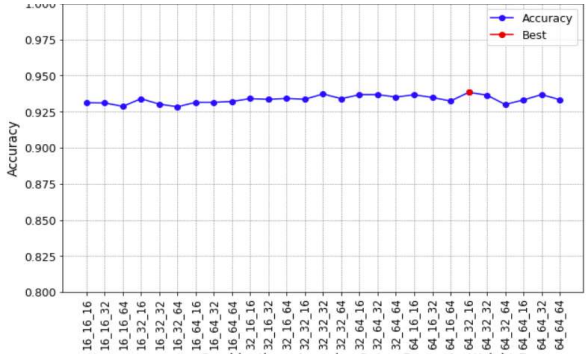

Figure 2. Hyperparameter combination: Number of neurons in 3 layers

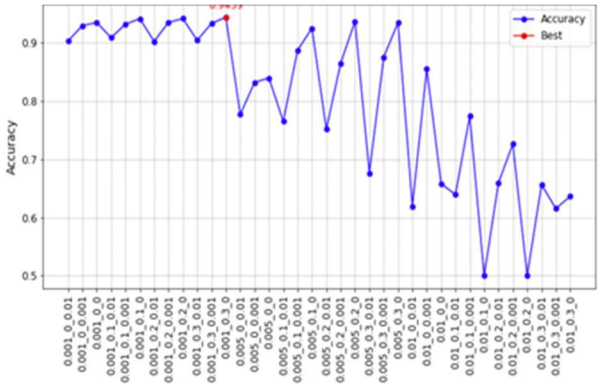

Figure 3. Hyperparameter combination: Learning Rate - Dropout - Weight decay

Figure 4 displays the hyperparameters (box constraint and gamma) combinations used in the grid search for the SVM model. No significant pattern is apparent in the graph. The optimal values for the box constraint and gamma parameters are 20 and 0.1, respectively, as determined through 5-fold cross-validation, achieving an average accuracy of 97%.

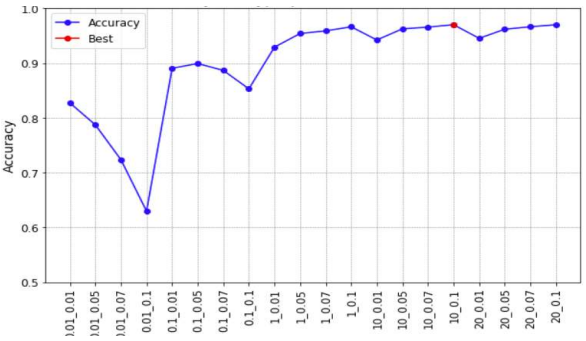

Figure 4. Hyperparameter combination: Box Constraint - Gamma

Figure 5 shows the training and validation losses of the final MLP model during training on the entire train set. Initially, there were high losses for both training and validation, but after a few epochs, the model started to learn, resulting in a significant reduction in losses. The validation losses decreased along with the training losses, indicating that the model learned effectively without overfitting. The training process stopped at 120 epochs using the early stopping callback to prevent overfitting.

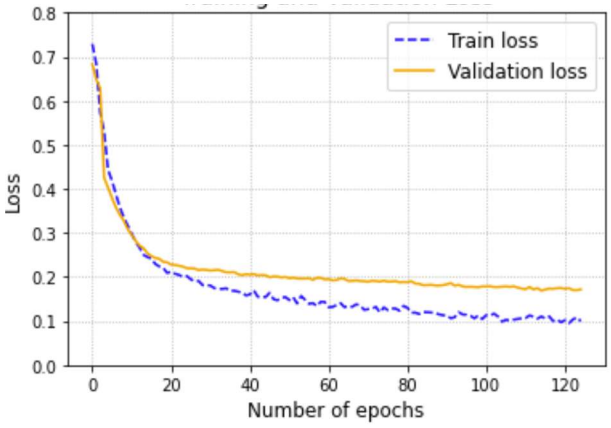

Figure 5. Loss vs Number of epochs

Table 1 displays the average training time for each model during the grid search. The average time to fit each SVM model across five folds is roughly 20 seconds, while MLP models require over 130 seconds on average to train across five folds. This indicates that training MLP models is usually much more computationally demanding than training SVM models.

Subsequently, our focus will be directed toward the optimal models discovered through the grid search. The multi-layer perceptron (MLP) model exhibits a test accuracy of 91%, which is 3% less than the validation accuracy. On the other hand, The support vector machine (SVM) model demonstrates a test accuracy of 90%, which is 7% less than the cross-validation accuracy. This suggests that the SVM model, apart from being slightly less accurate, might be marginally overfitted.

Table 1. Classification Results for Final MLP and SVM Models

MLP	Measure	SVM
130.4	Average Train Time (s)	19.7
0.94	Validation Accuracy	0.97
0.91	Test Accuracy	0.90
0.67	Precision (minority class)	0.77
0.86	Recall (minority class)	0.51
0.89	AUC	0.74

Figures 6 and 7 show the confusion matrices generated for the final model testing for SVM and MLP respectively. It is evident that each model exhibits unique strengths and weaknesses. Focusing on the model's classification errors, the MLP model displays a higher number of misclassifications of the positive class, i.e., existing customers(stayed), in comparison to the Support Vector Machine (SVM) model. Specifically, the MLP model incorrectly identifies 135 customers who did not leave the company as customers who left the company, while the SVM model misclassifies only 50 such customers.

On the other hand, the MLP model exhibits better performance in predicting the zero class, i.e., churned customers(left), as compared to the SVM model. The MLP model has a lower number of false positives compared to the SVM model, indicating that it makes fewer mistakes in identifying the customers who left the company's services.

Considering the importance of identifying customers who are likely to leave (class 0) for the company, it may be acceptable for the MLP model to make errors in misclassifying some existing customers (class 1) as potential churners. Therefore, we can conclude that the MLP classifier may be better suited for this study's task

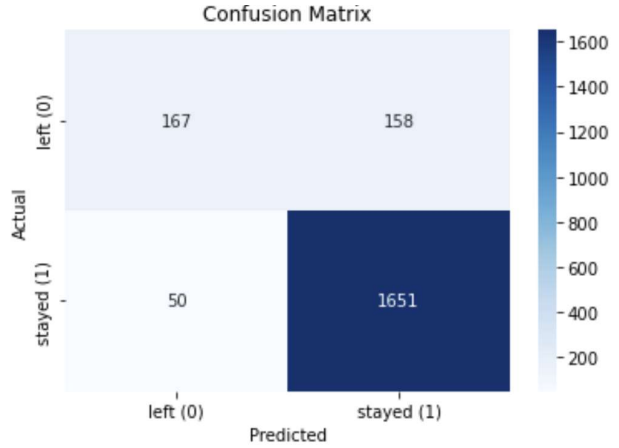

Figure 6. Confusion Matrix: SVM Model

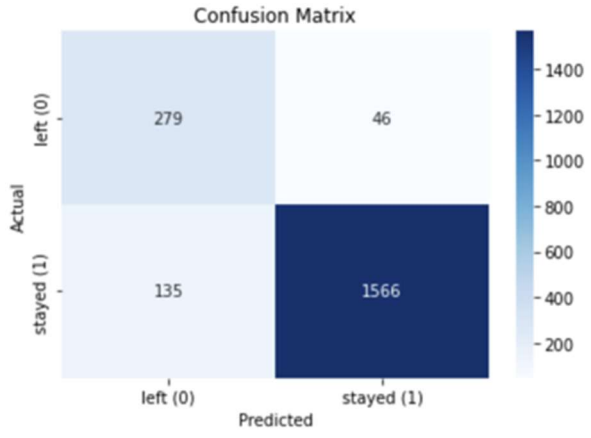

Figure 7. Confusion Matrix: MLP Model

Upon examining other metrics in Table 1, we can confirm our previous conclusion. For example, the recall score for churned customers is 0.86 for the MLP model, compared to only 0.51 for SVM. This result indicates that in the MLP model, 86% of actual class 0 instances were accurately predicted as class 0. The recall is a crucial metric when the cost of a false negative (predicting a

negative when the actual class is positive) is high. As our study aims to avoid missing the prediction of customers who are likely to leave the company, the recall score indicates that the MLP classifier is a more suitable choice for this task.

7. Lessons Learned and Future Work:

Several critical points have emerged during this process. Firstly, it is crucial to utilize rules of thumb and studies that have worked on similar data and problems to obtain an idea about the range of hyperparameters for each model. Doing so can help save time and lead to better results. Additionally, preprocessing steps, such as feature scaling and dealing with class imbalance, are crucial for training both models and can significantly impact the results.

For future work, it may be beneficial to consider using an ensemble model that includes both the MLP and SVM models, given their different strengths. Furthermore, other hyperparameters for these models could be fine-tuned, such as increasing the number of hidden layers with varying numbers of neurons for the MLP model and trying alternative kernel functions, such as polynomial, for the SVM model. These modifications can potentially improve the performance of both models and lead to better results.

8. Conclusion

This study aimed to develop, compare and critically evaluate the performance of two models, namely the Multilayer Perceptron (MLP) and Support Vector Machine (SVM), for predicting customer churn in a credit card company. The grid search cross-validation method was utilized to fine-tune the parameters of each model. After analyzing the final MLP and SVM models, it was found that the SVM model had a faster training time. However, the accuracy of the MLP model was slightly higher, which supported our hypothesis. Moreover, the MLP model proved to be a more suitable classifier for this task, as it outperformed the SVM model in identifying customers who are likely to leave the credit card company.

References

1. Mijwel, M.M., *Artificial neural networks advantages and disadvantages.* Retrieved from LinkedIn https//www. linkedin. com/pulse/artificial-neuralnet Work, 2018.
2. Tu, J.V., *Advantages and disadvantages of using artificial neural networks versus logistic regression for predicting medical outcomes.* Journal of clinical epidemiology, 1996. **49**(11): p. 1225-1231.
3. Kim, S., K.-s. Shin, and K. Park. *An application of support vector machines for customer churn analysis: Credit card case*. in *Advances in Natural Computation: First International Conference, ICNC 2005, Changsha, China, August 27-29, 2005, Proceedings, Part II 1*. 2005. Springer.
4. K, D. *Top 4 advantages and disadvantages of Support Vector Machine or SVM*. 2019; Available from: https://dhirajkumarblog.medium.com/top-4-advantages-and-disadvantages-of-support-vector-machine-or-svm-a3c06a2b107.
5. Sabbeh, S.F., *Machine-learning techniques for customer retention: A comparative study.* International Journal of advanced computer Science and applications, 2018. **9**(2).
6. GOYAL, S. *Credit Card customers*. Available from: https://www.kaggle.com/datasets/sakshigoyal7/credit-card-customers.
7. Chawla, N.V., et al., *SMOTE: synthetic minority over-sampling technique.* Journal of artificial intelligence research, 2002. **16**: p. 321-357.
8. Domingos, E., B. Ojeme, and O. Daramola, *Experimental analysis of hyperparameters for deep learning-based churn prediction in the banking sector.* Computation, 2021. **9**(3): p. 34.

9. Heaton, J. *The Number of Hidden Layers*. 2017; Available from: https://www.heatonresearch.com/2017/06/01/hidden-layers.html.
10. Caruana, R., S. Lawrence, and C. Giles, *Overfitting in neural nets: Backpropagation, conjugate gradient, and early stopping.* Advances in neural information processing systems, 2000. **13**.

Appendix 1 -- implementation details

During the cross-validation grid search phase for tuning the learning parameters (learning rate, dropout, and weight decay) of the MLP model, we obtained some initial results. Figure XI presents the outcomes of the grid search after selecting the optimal architecture. A discernible pattern emerges from the graph, indicating that certain parameter settings yield more favorable outcomes. Upon closer examination, we observed that when the weight decay is set to 0.1, the model's accuracy drops sharply to a mere 50%, which is equivalent to a basic model that classifies at random. Therefore, we decided to remove this parameter from the range of weight decay values.

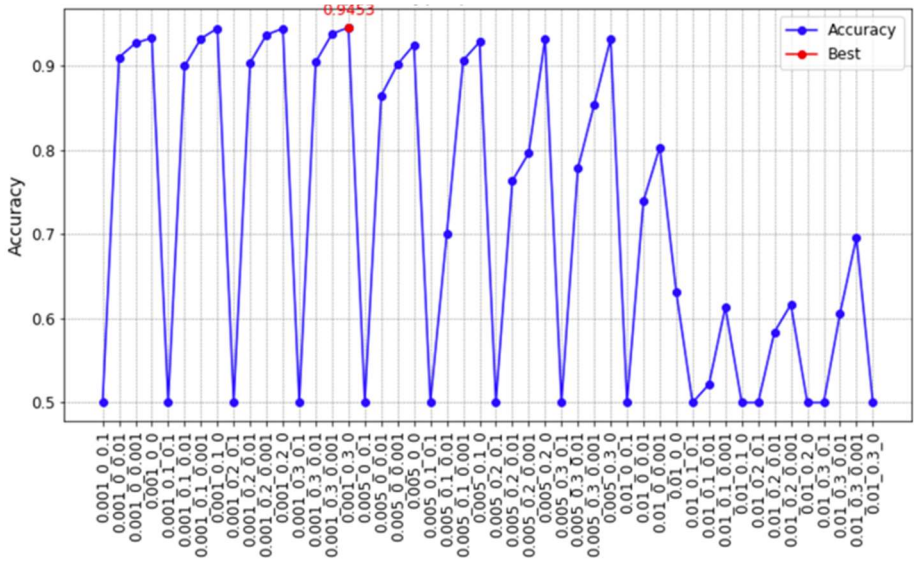

Figure 8 Hyper parameter combinations: learning Rate - Dropout - Weight decay

Appendix 2 -- Glossary:

- SMOTH

The SMOTE technique is used to deal with datasets that have an imbalanced target label. This approach increases the number of data points in the minority class by creating synthetic data points based on the existing data in the minority class. It does this by clustering blends of the data points that are near each other in the feature space.

- Grid Search:

Grid search is a method for hyperparameter tuning, which involves training a model on every possible combination of the list of hyperparameters that have been defined. The main aim is to find the best combination of hyperparameters in order to achieve the best performance metrics for the machine learning model.

- K-fold cross-validation

K-fold cross-validation is a technique that allows us to see how well a machine learning model generalizes to new data. This method divides the dataset into k folds, trains the model on k-1 folds, and uses the last fold for validation. This process is repeated until all folds have been used as test sets. The average performance metric across all k iterations is used as the performance of the model.

- Early Stopping

Early stopping is a regularization technique used in neural networks to prevent overfitting. It involves monitoring the validation loss during training and stopping the training process when the validation loss starts to increase, as this indicates that the model is starting to overfit to the training data. This helps to improve the generalization performance of the model.

- Dropout rate

Dropout is a regularization technique used in neural networks to prevent overfitting. During training, a fraction of randomly selected neurons are temporarily dropped out, or set to zero, which forces the network to learn more robust features. Dropout helps to improve the generalization performance of the model by reducing its reliance on any one particular set of neurons.