Study on Customer Segmentation using Multi-Layer Perceptron (MLP)

Pravin Patil[1], Pooja Chindarkar[2], Pratik Raut[3], Sakharam Zore[4]

[1]Professor, Dept. of Information Technology, Vasantdada Patil Pratishthan's College Of Engineering, Mumbai, Maharashtra, India

[2-4]Student, Dept. of Information Technology, Vasantdada Patil Pratishthan's College Of Engineering, Mumbai, Maharashtra, India

---

Abstract – *Customer Segmentation is a part of Customer Relationship Management (CRM), broadly also known as Market Segmentation, is important in business and software as it directly related to customer satisfaction. Here, in this study, we have segmented the customers into two distinct groups, the regular one, and the special customer on the basis of statistical data, which is used to train and test a neural network-based machine learning model, namely Multilayer Perceptron (MLP). Once the features are taken into consideration, and hyperparameters are tuned, by deploying a grid search algorithm, the model achieved the good classification of our customer segmentation technique, hence a good accuracy is observed within few epochs. This approach can be integrated into any company, who are looking forward to classifying their customer. This approach will frequently analyze customer data and can decide whether to promote customers as special ones or not. This algorithm helps businesses to make proper marketing decisions and improve customer satisfaction.*

Key Words: **Artificial Neural Network, Customer Segmentation, Machine Learning, Multi-Layer Perceptron, Market Segmentation, Customer Satisfaction.**

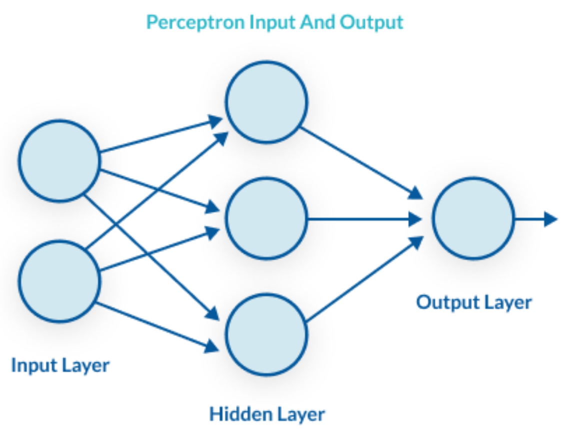
1. INTRODUCTION

Customer segmentation is the process of dividing customers into groups based on common characteristics so companies can market to each group effectively and appropriately.

Segmentation allows marketers to better tailor their marketing efforts to various audience subsets [7]. Those efforts can relate to both communications and product development. Specifically, segmentation helps a company for having more satisfied users.

1.1 Multi-Layer Perceptron

A multilayer perceptron (MLP) is a class of feedforward artificial neural network (ANN). An MLP consists of at least three layers of nodes: an input layer, a hidden layer and an output layer. Except for the input nodes, each node is a neuron that uses a nonlinear activation function.



Here, we are performing a binary classification and labelling the customers with two different labels such as"regular" and"special".

1.2 Previous Work

The segmentation has been performed manually, or through clustering techniques, which have lots of outliers. [1][2][4].

Same problem was addressed in a recent work [5], where the proposed method compares various clustering techniques to divide the customers into many groups.

The segmentation of shoppers can be done by partitioning the client into many gatherings, such as the shoppers can be based on their living types, demands, thinking behavior, economic conditions and many more[3]

1.3 Our Approach

In this study, a state-of-art artificial neural network model, namely multi-layer perceptron (MLP), is used to solve the same problem.

In further section, detailed information about the data used and what all data can be used in this model is discussed, also third section briefs about the application of neural network in segmenting by means of multilayer perceptron. The last section gives results of used data by means of plots which are plotted against the number of epochs and accuracy. Also loss is plotted for better understanding of trained and tested data

2. IMPLEMENTATION OF MLP

In this section, implementation details of the system are given. Data preparation is the first important step. When the data is ready, the necessary system parameters, i.e. hyper parameters, are optimized to achieve the best result.

2.1 Data Preparation

The details about the data are given. It is the first and foremost step in any machine learning algorithm. When data is ready and cleaning, the necessary system parameters, called as hyperparameters are optimized to give better results. The data used in this study is made in excel considering attributes such as Customer id, age (years), gender (Male=0, Female=1), income (in K), and spending score (0-100, 0 being lowest). This data consist of total 2000 rows of each column, of which 128 are our special customer labelled as '1' and rest of 1872 are regular customers.

Since the amount of special customers are nearly one tenth of the total customers, this makes it difficult for a regression based learning algorithm to achieve high accuracy because of the unbalanced nature of the sample distribution. However, using neural networks makes it possible to generalize a successful model.

Table -1: Data Set

	Customer Segmentation Data Set				
Customer ID	Age	Gender	Income (K)	Spending Score(0-100)	Label
1	26	1	146	34	0
2	12	1	96	93	0
3	29	0	86	33	0
4	48	0	148	10	0

2.2 Hyperparameter Tuning

Once the data is ready, the model can be trained, by default, we except the neural network to produce good results

As to train the model effectively, the next step, after data preparation is to tune the hyperparameters. This is the most crucial step, as it directly affects the accuracy of the system.

Various components of a machine learning model can be treated as hyper parameters, which can be optimized. In this study, components such as learning rate, batch size, activation function, number of epochs, optimization algorithm and dropout rate are selected as hyper parameters.

As traditionally performed, hyperparameters are searched by imposing a grid search where the best combination of hyperparameters are found by testing different values. In Table II selected hyper parameters with their reasonable value sets and their optimized values, in the last column, can be seen.

According to the optimized hyper parameters, the model has given the best accuracy of 93.60%. Moreover, the best learning rate is found as 0.001. Together with the optimized batch size value, softplus is the activation function used in the input and hidden network layers. Since this is a binary classification, the sigmoid function is applied at the output node of the network.

Table -2: Data Set

	Hyperparameter Tuning	
Hyperparameter	Search Value	Optimized value
Batch Size	[10, 20, 40, 60, 80, 100]	40
Epoch	[10, 50, 100]	100
Optimizer	[SGD, RMSprop, Adagrad, Adadelta, Adam, Adamax, Nadam]	Nadam
Learning Rate	[0.001, 0.01, 0.1, 0.2, 0.3]	0.001
Momentum	[0.0, 0.2, 0.4, 0.6, 0.8, 0.9]	0.2
Activation Function	[softmax, softplus, softsign, relu, tanh, sigmoid, hard_sigmoid, linear]	softplus

```
Best: 0.928518 using {'batch_size': 40, 'epochs': 100}
0.884527 (0.078438) with: {'batch_size': 10, 'epochs': 10}
0.752093 (0.265721) with: {'batch_size': 10, 'epochs': 50}
0.892523 (0.067132) with: {'batch_size': 10, 'epochs': 100}
0.885527 (0.069723) with: {'batch_size': 20, 'epochs': 10}
0.778582 (0.234625) with: {'batch_size': 20, 'epochs': 50}
0.780079 (0.243353) with: {'batch_size': 20, 'epochs': 100}
0.798569 (0.191523) with: {'batch_size': 40, 'epochs': 10}
0.869035 (0.049171) with: {'batch_size': 40, 'epochs': 50}
0.928518 (0.039086) with: {'batch_size': 40, 'epochs': 100}
```

Fig -1: Batch Size and Epoch Tuning

```
Best: 0.933005 using {'optimizer': 'Nadam'}
0.883527 (0.079851) with: {'optimizer': 'SGD'}
0.781587 (0.233662) with: {'optimizer': 'RMSprop'}
0.793572 (0.159619) with: {'optimizer': 'Adagrad'}
0.094983 (0.027508) with: {'optimizer': 'Adadelta'}
0.915012 (0.019562) with: {'optimizer': 'Adam'}
0.882528 (0.084492) with: {'optimizer': 'Adamax'}
0.933005 (0.010165) with: {'optimizer': 'Nadam'}
```

Fig -2: Optimizing Algorithm

```
Best: 0.936001 using {'learn_rate': 0.001, 'momentum': 0.2}
0.745597 (0.275967) with: {'learn_rate': 0.001, 'momentum': 0.0}
0.936001 (0.005772) with: {'learn_rate': 0.001, 'momentum': 0.2}
0.936001 (0.005772) with: {'learn_rate': 0.001, 'momentum': 0.4}
0.935501 (0.006463) with: {'learn_rate': 0.001, 'momentum': 0.6}
0.928005 (0.016987) with: {'learn_rate': 0.001, 'momentum': 0.8}
0.838050 (0.144161) with: {'learn_rate': 0.001, 'momentum': 0.9}
```

Fig -3: Learning rate and Momentum

```
Best: 0.948998 using {'activation': 'softplus'}
0.936001 (0.005772) with: {'activation': 'softmax'}
0.948998 (0.007463) with: {'activation': 'softplus'}
0.932503 (0.010655) with: {'activation': 'softsign'}
0.933003 (0.025387) with: {'activation': 'relu'}
0.936001 (0.005772) with: {'activation': 'tanh'}
0.936001 (0.005772) with: {'activation': 'sigmoid'}
0.936001 (0.005772) with: {'activation': 'hard_sigmoid'}
0.903517 (0.051587) with: {'activation': 'linear'}
```

Fig -4: Activation Function

2.3 Optimization

The model uses binary cross entropy to calculate model loss. Moreover, Nadam is the chosen optimizer. Learning rate is the most important parameter of this optimizer. The hyper parameter search in the previous section has given 0.001 as the optimal learning rate within the given list of learning rates

3. RESULTS & CONCLUSION

The success of the model can also be verified by plotting model accuracy vs epoch and model loss vs epoch graphs.

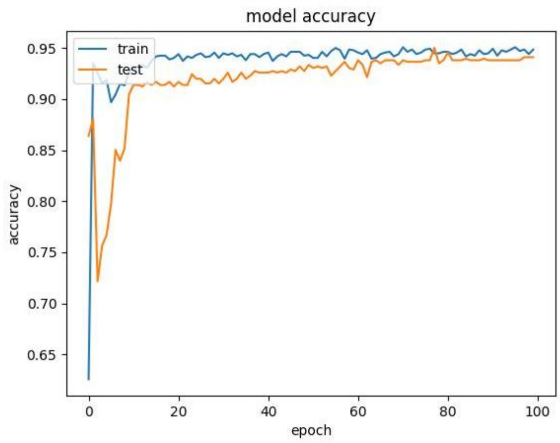


Chart -1: Model Accuracy

Chart-1 depicts the model accuracy over the training and the test sets. Chart-2 depicts the model loss over the training and the test sets. Optimal hyper parameter values in Table II are used.

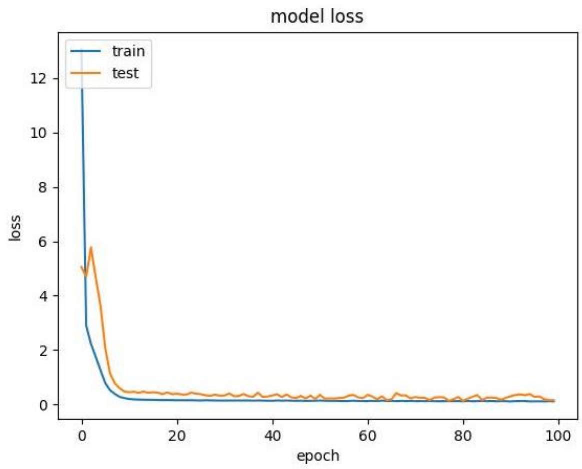


Chart -1: Model Loss

The performance of method can further be improved by making data augmentation and trying different ANN structures. Even performance of different activation functions can also be measured together with the hyper parameters. It is also possible to try different search methods other than grid search method.

As a future work, customer behavioral analysis by using machine learning techniques will be considered. This includes not only the statistical information, like the payment information of a customer, given in this study but also the sentiment analysis of communication between customers and customer representatives. Incorporating these information with machine learning can predict the probability of losing a customer at any time and enables the company to take some preemptive actions when necessary.

REFERENCES

[1] Dedi, Muhammad Iqbal Dzulhaq, Karthika Wulan Sari, Syaipul Ramdhan, Rahmat Tullah, Sutarman "Customer Segmentaion Based on RFM Value Using K-means Algorithm" STMIK Bina Sarana Global Tangerang, Indonesia, 2019.

[2] Ivan Derevitskii, Oksana Severiukhin, klavdiya Bochenina "Clustering Interest Graphs for Customer Segmentation Problems" 2019 Sixth International Conferences on Social Networks Analysis, Management and Security (SNAMS).National Centre for Cognitive Research, ITMO University Saint Petersburg, Russia.

[3] Pradnya Paramita Pramono, Isti Surjandari, Enrico Loha "Estimating Customer Segmentation Based on Customer Lifetime Value Using Two-Stage Clustering Method" Department of Industrial engineering Faculty of Engineering , Universitas Indonesia Depok, Indonesia, 2019.

[4] Lu siyue, Zhang BaoQun, Zhang Lu, Xu Hui, Zhang Jianxi, Ma Longfei, Wang Peiyi "Prediction Of Business User Segmentation Model Based On Customer Value " 2019 IEEE 4th International Conference on Cloud Computing and Big Data Analytics. State Grid Beijing Electric Power Research Institute. Beijing, China.

[5] Sarya Chugh, Vanshita R Baweja "Data Mining Application in Segmentation Customers With Clustering" 2020 International Confrence on Emerging Trenda in Information Technology and Engineering (ic-ETITE). Department of Computer Science Chikkara University, Rajapura, India.

[6] S¸ukr¨u Ozan¨, Leonardo O. Iheme∗, "Artificial Neural Network in Customer Segmentation", 2019, IEEE

[7] Brian Garda Muchardie, Annetta Gunawan, Billy Aditya, "E-Commerce Market Segmentation Based On The Antecedents Of Customer Satisfaction and Customer Retention", 2019, IEEE