** Kinnaird College for Women Lahore**

**Assignment No:**

Term Project (Customer Segmentation)

**Submitted By:**

**Mishaal Arbab**

**Hibba Mehmood**

Major: BSCS (Semester 6)

Course: Data Science with Python

Department: Computer Science

Kinnaird College for Women Lahore

**Submitted To:**

**Ma’am Nimra Zaheer**

Department: Computer Science

Date: January 1, 2020

Table of Contents

[Abstract: 2](#_Toc60299028)

[Introduction: 2](#_Toc60299029)

[What is Customer Segmentation? 2](#_Toc60299030)

[Need of Customer Segmentation: 3](#_Toc60299031)

[Proposed Model: 3](#_Toc60299032)

[Techniques: 3](#_Toc60299033)

[Dataset Description: 4](#_Toc60299034)

[Classification Problem: 4](#_Toc60299035)

[Label Encoder: 4](#_Toc60299036)

[Results: 4](#_Toc60299037)

[Underfitting/Overfitting: 4](#_Toc60299038)

[Learning Rate: **Error! Bookmark not defined.**](#_Toc60299039)

[NN (Neural Network) Classifier: 4](#_Toc60299040)

[Default case: 4](#_Toc60299041)

[Confusion Matrix: 4](#_Toc60299042)

[NN with Variations: 5](#_Toc60299043)

[Hidden Layer: 5](#_Toc60299044)

[Solver: 5](#_Toc60299045)

[Activation Function: 6](#_Toc60299046)

[Confusion Matrix: 6](#_Toc60299047)

[Comparison with other ML Algorithms: 7](#_Toc60299048)

[Justification: 7](#_Toc60299049)

[Other Feature Prediction: 8](#_Toc60299051)

[Annual Income (Regression): 9](#_Toc60299052)

[Conclusion: 10](#_Toc60299053)

[References: 10](#_Toc60299054)

Customer Segmentation

# Abstract:

This project aims to identify the potential customer base for selling the product. So, by predicting the correct class of the customer using various machine learning algorithms such as neural network, SVM, Gaussian Naïve Bayes, Decision Trees, Random Forests, K- Nearest Neighbor, and logistic regression, we would achieve that. We would develop this model using supervised learning. Our main goal is to identify the customer segment/class in order to sell the product and reap maximum revenue.

# Introduction:

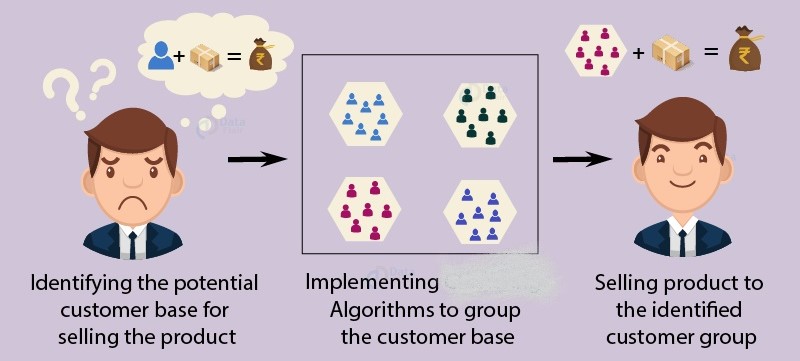
Whenever you need to find your best customer, customer segmentation is the ideal methodology.

## What is Customer Segmentation?

“Customer Segmentation is the process of division of customer into several groups or categories that share a similarity in different ways that are relevant to marketing such as gender, age, interests, and miscellaneous spending habits” [1].

Companies that deploy customer segmentation are eager to find their potential customer in order to generate maximum profit. Companies aim to gain a deeper approach of the customer they are targeting. Every customer has its own requirements and preferences therefore marketing strategies accompanied with machine learning are designed in such a way that would cater the customer of every segment and thus products are designed and developed according to the targeted audience. In this way companies would reap maximum profit in every domain.

Many machine learning algorithms have been applied to effectively forecast the class of the customer. We propose to apply machine learning techniques i.e., Neural Network, SVM, Gaussian Naïve Bayes, Decision Trees, Random Forests, Multi-Layer Perceptron Classification, K- Nearest Neighbor, and logistic regression. It is crucial to correctly predict the class of the customer in order to maximize profit.



# Need of Customer Segmentation:

* It will help in identifying the most potential customers.
* It will help managers to easily communicate with a targeted group of the audience.
* It improves the quality of service, loyalty, and retention.
* Improves customer relationship via better understanding needs of segments.
* It will help managers to design special offers for targeted customers, to encourage them to buy more products.
* It helps companies to stay a step ahead of competitors.
* It also helps in identifying new products that customers could be interested in.

# Proposed Model:

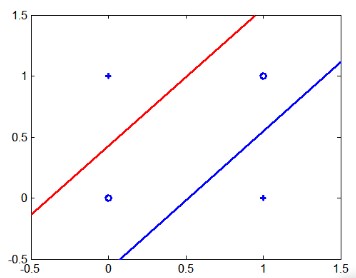
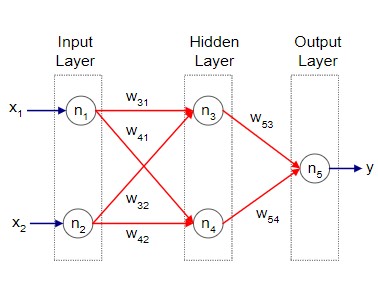


Fig. 1. Multi -Layer Perceptron NN Fig. 2. Graph of Multi-Layer Neural Network

# Techniques:

# Dataset Description:

Our dataset consists of 6 columns(features) which include age, customer Id, gender, annual income, spending score and class. Customer id is not considered a good feature as these are just Id’s that have a zero contribution to the problem. Apparently, annual income and spending score look pretty good features as; the more money you got the more you spend which in turn increase the spending score so these have some relation between them whereas gender and age appears to have a moderate affect to the problem. Maybe children and senior citizens are not much shopaholic as much as youngsters are.

## Classification Problem:

It’s purely a classification problem as the thing(label/Y) we are to predict consists of categorical data. We are predicting in which class does the potential customer lie out of these three classes Rich, Poor and Middle Class, given the features(X) age, gender, spending score, income and customer Id etc.

## Label Encoder:

Since the classifier only deal with numeric data; therefore, we use label encoder to convert our categorical data into numeric codes (0,1,2). Column names that include categorical data are

* Class (Rich, Poor, Middle Class)
* Gender (Male, Female)

# Results:

Here we are implementing algorithms to group the customer base.

## Underfitting/Overfitting:

There is no such difference between the training accuracy and the testing accuracy. They are pretty much close to each other for every algorithm that we use therefore we conclude that our results are very much generalized.

## NN (Neural Network) Classifier:

### Default case:

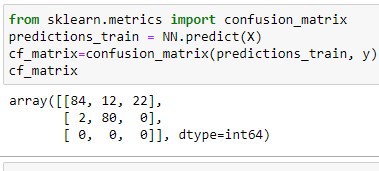
First, we calculate the accuracy at the default state that comes out to be 90% then we change the parameters (hidden layers, activation function, solver) and observe the fluctuations that we see in our accuracy and record them in the form of a table given in figure 2.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | Solver | Activation Function | Hidden Layers | Accuracy |
| Default state | Adam | Relu | 100 | 82% |

Fig. 3. Default case

## Confusion Matrix:

In the following confusion matrix, the model has not correctly classified all classes. It says class 0 as 0, 84 times and wrongly predict it of 2 times. It correctly predicts class 1 as 1, 80 times and wrongly predict it 12 times. It then predicts class 2 as 2, none times and wrongly predict it 22 times. So, it is not a very good model up till now.



## NN with Variations:

|  |  |  |  |
| --- | --- | --- | --- |
| Activation Function | Hidden Layers | Solver | Accuracy |
| relu | 100 | **sgd** | 87% |
| relu | 100 | **lbfgs** | 94% |
| relu | **100** | **adam** | **82%** |
| tanh | 100 | adam | 92% |
| logistic | 100 | adam | 91% |
| identity | 100 | adam | 87% |
| relu | **800** | adam | 93% |
| relu | **1500** | adam | 95% |
| relu | **70** | adam | 77% |
| relu | **5** | adam | 39% |

Fig. 2. NN with Variations

### Hidden Layer:

Every hidden layer represents a level of abstraction. Complex features are compositions of simpler features. As we know number of layers is known as depth of ANN therefore, deeper networks express complex hierarchy of features. We observe sometimes it happens if there comes a sudden jump or drastic increase in the number of hidden layers then it would have a negative impact to our accuracy means that it will decrease in that case whereas if we increase it sequentially in a very slow-moderate pace, not a straight jump or peak then it would significantly add some value to our accuracy and increase it [3]. Although when we generally used this parameter then our observation is as follows accuracy increases if we increase the number of hidden layers and decreases if we decrease the number of hidden layers. As we clearly see in the table given above when we increase the layers to 1500 then we are achieving the maximum accuracy and the lowest accuracy is achieved in case of 5 hidden layers that is 39%.

### Solver:

In general, a solver to optimization is similar as an engine is to driving. The default solver ‘Adam’ works pretty well on relatively large datasets with thousands of training samples or more in terms of both training and accuracy score. For small datasets, however, ‘lbfgs’ can converge faster and perform better. As our dataset is not very dense and diverse therefore lbfgs perform better in our case. Whereas sgd refers to stochastic gradient descent that updates the weight for every instance. It measures errors across all training points. It increases the accuracy to 87% when we used it but lbfgs is giving far better results than this solver [4]. Out of all three solvers lbfgs perform better and give highest accuracy of 94% whereas Adam performs least and give lowest accuracy of 82% as it is not a perfect fit for small datasets.

### Activation Function:

We observe that when we change activation function to tanh then our accuracy is increased and then when we use logistic and identity functions respectively then in case of logistic our accuracy is slightly dropped (91%) whereas when we use identity then also there can be see a decrease (87%) in the accuracy of the model. So, we conclude that these two are not contributing well whereas tanh is the star here as it is contributing in increasing the accuracy of the model. Therefore, tanh is a great function for such problems [5]. Out of all four functions tanh increases the accuracy up to 92% whereas relu provides the lowest accuracy of 82%.

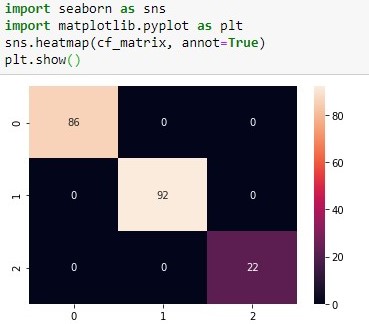
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | Solver | Activation Function | Hidden Layers | Accuracy |
| Best Fit | lbfgs | tanh | 500 | 100% |

Fig. 4. Main Result

We attain best result when solver is set to lbfgs, activation function is set to tanh and hidden layers is set to 500.

## Confusion Matrix:

In the following confusion matrix as all non-diagonal entries are zero therefore, we conclude that the model has correctly classified all classes followed by 86 times class 0, 92 times class 1- and 22-times class 2 respectively.



## Comparison with other ML Algorithms:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Algorithms | F1-Score | Recall | Precision | Accuracy |
| SVM | 0.97 | 0.97 | 0.98 | 0.97 |
| K-Nearest Neighbors | 0.97 | 0.95 | 0.98 | 0.95 |
| Gaussian Naïve Bayes | 0.99 | 0.99 | 1.0 | 0.99 |
| Decision Trees | **1.0** | **1.0** | **1.0** | **1.0** |
| Random Forests | **1.0** | **1.0** | **1.0** | **1.0** |
| Logistic Regression | 0.92 | 0.91 | 0.93 | 0.93 |
| Neural Network | **1.0** | **1.0** | **1.0** | **1.0** |

Fig. 5. Our Results

It is to be noted that performance of machine learning classifiers can yield varying results depending on the shape and structure of the data. Decision trees, random forests and neural network performed well. They are giving the highest accuracy that is considered to be the best accuracy which is 100% whereas logistic regression is giving the lowest accuracy that is 93%.

### Justification:

Our problem is not a binary classification problem. Binary classification problem is a problem in which making a prediction when the thing to predict can be one of just two possible values. For example, you might want to predict if a person is male (0) or female (1) based on age, annual income, height, weight, and so on.

Logistic regression is a technique that can be used for binary classification. A neural network is more complex than logistic regression. And, logistic regression is a subset of a neural network classifier. The moral of the story is that, in principle, anything you can do with logistic regression you can do with a neural network. Therefore, theoretically, a neural network is always better than logistic regression, or more precisely, a neural network can do no worse than logistic regression [6].

Logistic Regression and trees differ in the way that they generate decision boundaries i.e., the lines that are drawn to separate different classes. To illustrate this difference, let’s look at the results of the two model types on the following 2-class problem:

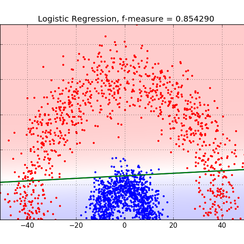
**[](https://blog.bigml.com/2016/09/28/logistic-regression-versus-decision-trees/lr_boundary_radial/)**

Fig.6. Logistic Regression

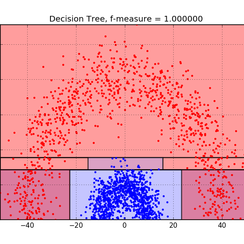
**[](https://blog.bigml.com/2016/09/28/logistic-regression-versus-decision-trees/model_boundary_radial/)**

Fig. 7. Decision Trees

Decision Trees bisect the space into smaller and smaller regions, whereas Logistic Regression fits a single line to divide the space exactly into two. A single linear boundary can sometimes be limiting for Logistic Regression. In this example where the two classes are separated by a decidedly non-linear boundary, we see that trees can better capture the division, leading to superior classification performance [7]. Therefore, we can conclude from here that as the decision trees bisects the data more precisely and accurately in more than one smaller, smaller chunks rather on just dividing the data by drawing a single line thus providing better accuracy than decision trees.

For the cases of complex datasets, linear-based algorithms may not be sufficient in segmenting the class labels, leading to poor accuracies. More sophisticated algorithms may then be required like random forest, which can learn a non-linear decision boundary and thus can achieve higher accuracy scores. For instance, a toy dataset is shown below in Figure 8 consisting of concave and convex shapes. As illustrated in this figure, logistic regression (left) poorly segments the two classes while the more flexible decision boundary learned from the random forest model produces a higher classification accuracy.

## 

Fig. 8. Decision boundary between binary classes for Logistic Regression (left) and Random Forest (right) with complex data structures (e.g., concave and convex)

Logistic Regression makes a prediction for the probability using a direct functional form whereas Naive Bayes figures out how the data was generated given the results. Naive Bayes has a higher bias but lower variance compared to logistic regression. If the data set follows the bias then Naive Bayes will be a better classifier.[8]

## Other Feature Prediction:

### Annual Income (Regression):

When we implement regression on our data set [mall customer segmentation] we select the x, y features first. Here we choose multiple features for x and predict y on the bases of features we select for x variable.

* **X** = Customer ID, Gender, Age, Spending Score (1-100), Class
* **Y**=Annual Income (k$)

We choose y variable as annual income because we want to solve it through regression, and for regression we need numbers which are random numbers that’s why I choose y as annual income so that we can predict annual income on the bases of another variable which we take as X.

First, we encode the class and gender because it is categorical data assign 1 to poor class, assign 0 to middle class and assign 2 to Rich class. For Gender assign 0 to female and 1 to male. For regression problem on the bases of Customer ID, Gender, Age, Spending Score (1-100) we predict Annual Income (k$).

* The shape of X variable is (200,5) 🡪 200 rows and 5 columns
* The shape of y variable is (200,) 🡪 200 rows and single column

Divide data 20%for testing and 80% for training after training and testing for X variable shape of variable change.

* X train. Shape= (160,5)
* X test. shape (40,5)

Apply linear regression and fit X train, y train in it and find the values of intercept or coefficient for linear equation y=mx+c

**from sklearn.linear\_model import LinearRegression**

**regressor = LinearRegression ()**

11.086454116590623

[ 4.53052856e-01 6.16679800e-01 -2.41100849e-03 2.42628831e-04

5.55500969e+00]

**regressor. Fit(X\_train, y\_train)**

**print (regressor. intercept\_) #constant**

**print(regressor.coef\_)**

multiple (5) values of coef(m) because we choose multiple variables for X

now we predict y (annual income) on the bases of x test instances 🡪

y\_pred=regressor. predict(X\_test)

Now find the errors for identify that how much percent model predict wrong and a decision is it good fitting model or not?

* **Mean Absolute Error: 2.542099990869061**
* **Mean Squared Error: 11.21449767384932**
* **Root Mean Squared Error: 3.3488054099707436**

Errors means how many percent model predict wrong annual income. Error must be less wide range of error shows that the model predicts not correctly. When we see MAE then it is good fit model means when we need predict values near to mean values. When we see MSE then it is neither good neither bad fit model because approximately 11.21% error occur. When we see RMSE then it is also good fit model because 3.34% error occur. RMSE used for getting difference between the values.

# Conclusion:

In this project, we went through the customer segmentation model. We developed this using a type of machine learning known as supervised learning. Specifically, we made use of neural network algorithm. We also trained our model through other machine learning algorithms. We analyzed and visualized the data and then proceed to implement the algorithm. We calculated results and recorded accuracy and thus conclude that our model is best fitted for the given problem with the accuracy of 100% and is trained well. Our goal that is selling product to the identified customer segment/category is very much achieved. We can also predict other features as in age, gender, spending score etc. and thus decide accordingly either it is a classification or regression problem and fit our model.

# References:

[1]<https://www.kaggle.com/vjchoudhary7/customer-segmentation-tutorial-in-python>

[2] <https://machinelearningmastery.com/understand-the-dynamics-of-learning-rate-on-deep-learning-neuralnetworks/#:~:text=A%20learning%20rate%20that%20is,the%20process%20to%20get%20stuck.&text=If%20you%20have%20time%20to,hyperparameter%2C%20tune%20the%20learning%20rate>.

[3]<https://scikit-learn.org/stable/modules/generated/sklearn.neural_network.MLPClassifier.html>

[4]<https://scikit-learn.org/stable/modules/generated/sklearn.linear_model.LinearRegression.html>

[5]<https://data-flair.training/blogs/r-data-science-project-customer-segmentation/>

[6] <https://towardsdatascience.com/everything-you-need-to-know-about-activation-functions-in-deep-learning-models-84ba9f82c253>

[7] <https://scikit-learn.org/stable/modules/sgd.html>

[8] https://medium.com/@sangha\_deb/naive-bayes-vs-logistic-regression-a319b07a5d4c#:~:text=Naive%20Bayes%20also%20assumes%20that,will%20be%20a%20better%20classifier.