Utilizing Machine Learning to project the financial outcomes of reconnecting with potential customers of the same industry

by

Shoumik Hossain
17101322
Quazi Fahmiduzzaman
17101307
Nehrin Siddique Payel
17101508
Mohammad Shahriar Hossain
17101239
Nabil Hossain
16301134

A thesis submitted to the Department of Computer Science and Engineering
in partial fulfillment of the requirements for the degree of
B.Sc. in Computer Science

Department of Computer Science and Engineering
Brac University
January 2021

Declaration

It is hereby declared that

1. The thesis submitted is our own original work while completing degree at Brac University.
2. The thesis does not contain material previously published or written by a third party, except where this is appropriately cited through full and accurate referencing.
3. The thesis does not contain material which has been accepted, or submitted, for any other degree or diploma at a university or other institution.
4. We have acknowledged all main sources of help.

Student's Full Name & Signature:

Shoumik Hossain
17101322

Quazi Fahmiduzzaman
17101307

Nehrin Siddique Payel
17101508

Mohammad Shahriar Hossain
17101239

Nabil Hossain
16301134

Approval

The thesis/project titled "Utilizing Machine Learning to project the financial outcomes of reconnecting with potential customers of the same industry" submitted by

1. Shoumik Hossain (17101322)
2. Quazi Fahmiduzzaman (17101307)
3. Nehrin Siddique Payel(17101508)
4. Mohammad Shahriar Hossain (17101239)
5. Nabil Hossain (16301134)

Of Fall, 2020 has been accepted as satisfactory in partial fulfillment of the requirement for the degree of B.Sc. in Computer Science on January 15, 2021.

Examining Committee:

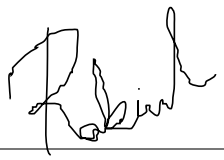
Supervisor:
(Member)

Moin Mostakim
Lecturer
Computer Science and Engineering
Brac University

Co-supervisor:
(Member)

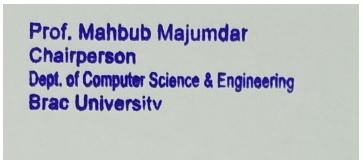
Mahbub Alam Majumdar
Professor and Dean
School of Data and Sciences
Brac University

Program Coordinator:
(Member)



Dr. Md Golam Robiul Alam
Associate Professor
Computer Science and Engineering
Brac University

Head of Department:
(Dean)



Prof. Mahbub Alam Majumdar
Dean
School of Data and Sciences
Brac University

Abstract

In competitive markets, it is costly to attract new customers in the business as they already have a wide customer base; that being the case, businesses spend a healthy budget to bring back customers who have once been with them. Despite the business investing heavily in trying to retain their customers, the re-engagement of the customers is not satisfactory as many businesses use intuition, experience, and traditional methods for marketing literature. Moreover, there is a global pandemic (COVID-19) that is hampering businesses everywhere. While the majority of the businesses are operating on a loss, a few small businesses are already being shut down. Thus, there is a need to increase the profitability of the businesses and retain their customer base. To increase customer turnover and re-engagement, this paper focuses on the implementation of intelligent business practices using machine learning and neural networks. The paper focuses on analyzing the customer behavior based on their purchase behavior and transactional values for customer retention. The classification is done to identify the customers who are profitable to the business and customers who are likely to churn due to various reasons. In this paper, we proposed a Multi-layer perceptron (MLP) to segment the customers according to RFM methodology and customer profitability index. The result of MLP was compared with K-Nearest Neighbors (KNN) and Support Vector Machine (SVM) as the later models have been widely used. Additionally, Bidirectional Long Short-Term Memory (LSTM) models have been implemented for primary customer classification and sales prediction. The prediction model is an attempt to reduce financial loss on marketing campaigns for re-engagement of customers in the business.

Keywords: Customer Segmentation, Long Short-Term Memory (LSTM) Networks, Deep Learning, Multi-layer Perceptron, K-nearest Neighbor (KNN) Algorithm.

Acknowledgement

Firstly, all praise to the Almighty Allah for whom our thesis have been completed without any major interruption.
Secondly, to our co-advisor Mr. Moin Mostakim sir for his kind support and advice in our work. He helped us whenever we needed help.
Thirdly, to Cardinal Care BD Ltd. for providing us with the data set we used for manipulation and model training.
And finally to our parents without their throughout support it would not be possible. With their kind support and prayer we are now on the verge of our graduation.

Table of Contents

List of Figures

List of Tables

Chapter 1

Introduction

Every business adapts a marketing policy to promote their products or services to reach their targeted potential customers in the market. Nevertheless, the core purpose of any marketing policy is to ensure that the revenue generated is more than the expenses incurred to maximize profit. This research aims to forecast business policies that can bring in more revenue and predict the steps a company should take for profit maximization. The following research will be based on identifying customer purchase behavior patterns to recommend offers and promotions for customer retention using machine learning and neural network algorithms. The main focus of the research problem is to mitigate the cost of the business incurred due to unsuccessful marketing campaigns as a great proportion of the budget is spent on advertisements and promotional offers for customers based on traditional marketing policies or intuition.

However, with the research model developed using statistical tools and neural network algorithms, we plan to see how a particular offer might work in the future and what offers might be perfect for the business to implement next. Additionally, this research will also help businesses to control their marketing budgets when needed. For national or economic crises, our models can be helpful to recognize the purchase patterns of the customers and how it will affect a business and its sales turnover. For instance, the pandemic outbreak of Corona virus disease has stalled the world economy and businesses are forced to stop their operation during such pressing times. With almost zero revenue these companies are having to pay their overhead costs and are operating at a loss. When the situation improves, these businesses might need to change their selling price and develop strong marketing schemes to attract more customers to compensate for the losses by boosting their sales.

However, not all businesses will be able to re-engage with their customers. With available customer information and past sales records, a series of results using models to understand how exactly the customers react to various marketing strategies and predict which type of strategy might be the most beneficial for the business owners will be implemented for sales boosting.

Chapter 2

Related Works

Business in modern times is more service-oriented and thus, businesses focus on customer satisfaction and customer feedback to improve their service. According to Di GangiWasko (2009), businesses should use customer engagement tactics like customer loyalty rewards, membership cards, or discounts to encourage the existing customers to purchase more services or products from that company which will give the business a competitive edge. Additionally, Thakur Summey (2010) have stated that customer re-engagement can be a good factor to predict the future revenue model and performance of the business.

To identify regular, loyal and profitable customers CRM (Customer Relationship Management) is used. Traditional CRMs would evaluate the records of the business to detect patterns in the data; however, the outcomes were not always successful, profitable and accurate. With the advancement of technology, machine learning tools and algorithms can be used to create intelligent business practices such as prediction models to reduce churn and increase re-engagement of the customer.

Based on the research conducted by Aluri, Mcintyre and Price (2018), it was established that implementation of machine learning in CRM can provide individualistic data of customers that can assist in personalized marketing and help to create a specific customer profile. Analyzing customer profiles can help the business to implement re-engagement techniques to retain existing customers. This can be done by updating the database with the behavioral patterns of the customers. The data will then be used by machine learning algorithms in real-time to improve the individual data stored about each customer so that a better-customized offer based on the profile can be given. This novel feature will not only help the business generate more revenue but will also help to identify regular customers. Hence, using this, customer churn management can be done. The profit generated from an individual can be determined using a model that has been designed to calculate values related to this task specifically.

To date, various models have been implemented to categorize clients depending on the pattern of computation required. Some examples are artificial 4 neural networks (ANNs) [4,5,6], expert systems [8], decision trees [7] etc. At the same time, ANNs have been implemented as well to predict any potential risks and for identifying the consumer's capability of being able to pay back loans. It has been observed that ANNs have easily solved issues that traditional models failed to do [9]. But ANNs have a specific set of drawbacks associated with their behavior. Extremely frequent anomalous results are generated which is a major issue as it may portray a bad

client as a creditworthy one due to the errors in calculations due to wrong results and this may lead to improper investments [10]. Moreover, ANNs are considered to be a black box as understanding the output of this model is quite difficult [3].
Lim and Sohn have stated that to differentiate between clients more precisely, using only one algorithm is not sufficient [11]. They suggested that a combination of algorithms will be more suitable for classifying clients with more precision so that the final decision contains fewer errors as possible. Thus, this research introduced Long Short-Term Memory (LSTM) networks forsales prediction and classification of the customers. For the segmentation of the customers into different groups this paper has utilized models that implement Support Vector Machine, Multilayer Perceptron and K Nearest Neighbor to generate multiples results of various accuracy. After proper comparisons, the best results will be chosen which will ensure the reliability of the output. Although this will require higher computational power and will increase the time complexity, the accuracy of the result will be beneficial in saving thousands of dollars in expense for business owners which are generally wasted on reconnecting with previous customers.
RNNs could have been used instead of LSTMs but the idea of implementing RNNs had to be discarded due to the vanishing gradient problem. This is because RNNs have no cell states as compared to LSTMs [1]. The use of a cell state is crucial in utilizing outputs of previous modules as an input of modules which exist along further iterations. A special type of LSTM known as Bidirectional LSTM has also been used which can simultaneously process information from two different directions [2]. Models of KNN, SVM and MLP has been implemented alongside all these to gain further insight on the output and compare them accordingly.

Chapter 3

Work Flow Diagram

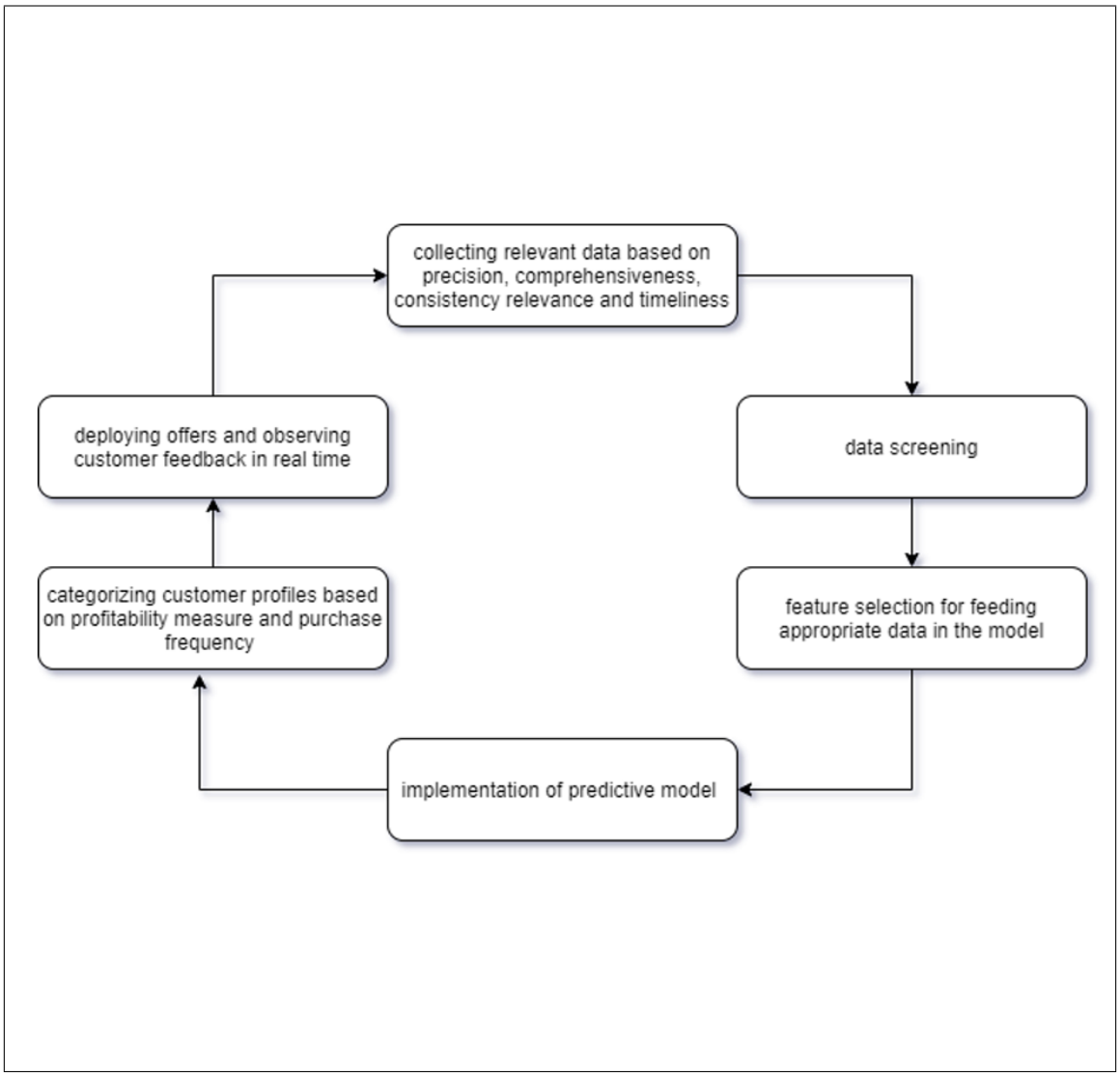


Figure 3.1: Work Flow Diagram

Chapter 4

Data Pre Processing

Data Screening

It is not feasible to waste marketing resources to retain customers who are not going to add any value to the business, hence, data screening is of utmost importance. Initially, we found the percentiles, mean, max and standard deviation of the columns relating to the transactional details. Any value less than the 1st quartile value of the column was dropped.

Aggregation

As we collected raw data, multiple transactions relating to a particular CustomerID is present in the dataset. To avoid irrelevant and multiple segmentation of the same customer, data was grouped based on Customer ID to retain all the necessary information in a single column. This will also help us to understand the purchase behavior of a particular customer.

Imputation

Imputation is handling for null or missing values. Initially, the raw database having any null value relating to the column of purchase information indicates that the customer has made no purchase and is a potential customer. However, since our focus is on retaining existing customers, we drop the rows with missing values.

Extracting date

From the Invoice Date column, dates were extracted to find the monthly information such as total sales, total number of customers served, total quantity sold.

Figure 4.1: Monthly Sales

Figure 4.2: Number of Products Sold every Month

Figure 4.3: Number of Customer in a month

Data Subset

In order to find the RFM (Recency, Frequency, monetary) values for customers, the labels named "Invoice No", "Unit Price", "Quantity", "Invoice Date" were used. "Invoice No" column has a unique number for each invoice which is a proven statement of the purchase of a customer from the business.

Discretization

Afterwards, discretization was performed and the values of the RFM columns were transformed into ordinal values of 1,2,3,4 from continuous ones.

Feature Transformation and feature scaling

It is important for more distributed and comparable column values as different types of columns can have different range of values.

Standardization

It is used to convert or rescale the different types of data into a set, such that, the mean of the values is 0 and the standard deviation is 1. This is done individually according to each feature. The data goes through a process of centering followed by scaling which also called center scaling. The values of input are at first subtracted by the mean and then divided by the standard deviation to end up with the standardized value of the input.

MinMaxScaler()

Used to scale the features on the training set, such that the data set is in a given range. $(X - X_{min})/(X_{max} - X_{min})$

Customer Segmentation

Customer segmentation is done based on RFM methodology. RFM segmentation is a process which marketers use in order to focus on a certain group of customers and communicate with them in the hopes of getting a positive response from the majority of this customer base. The cluster of customers targeted is also expected to show more loyalty and good will towards the marketers. RFM segmentation is an effective method of targeting potential customers that can ensure high profitability for businesses and organizations. In order to implement RFM segmentation effectively marketers are required to have information on customers' interests, recent purchases, demographics etc.
Customers are categorized into two categories:

1. Possibility of Customer churn
2. Regular Customer

The two categories are divided into another 2 subcategories:

1. Sub-divisions of churn: High risk of churning (segmentation one) and Occasional buyers (segmentation two)
2. Sub-divisions of regular customer: Repeat customers (segmentation three) and Loyal customers (segmentation four)

To find the frequency of purchase of a customer, the Invoice numbers were counted for each customer. Date of invoice issue is available, from that available data, the date when the customer last purchased was identified and converted into days. For Monetary value, the total purchase of each customer was calculated by multiplying Quantity and Unit Price columns.

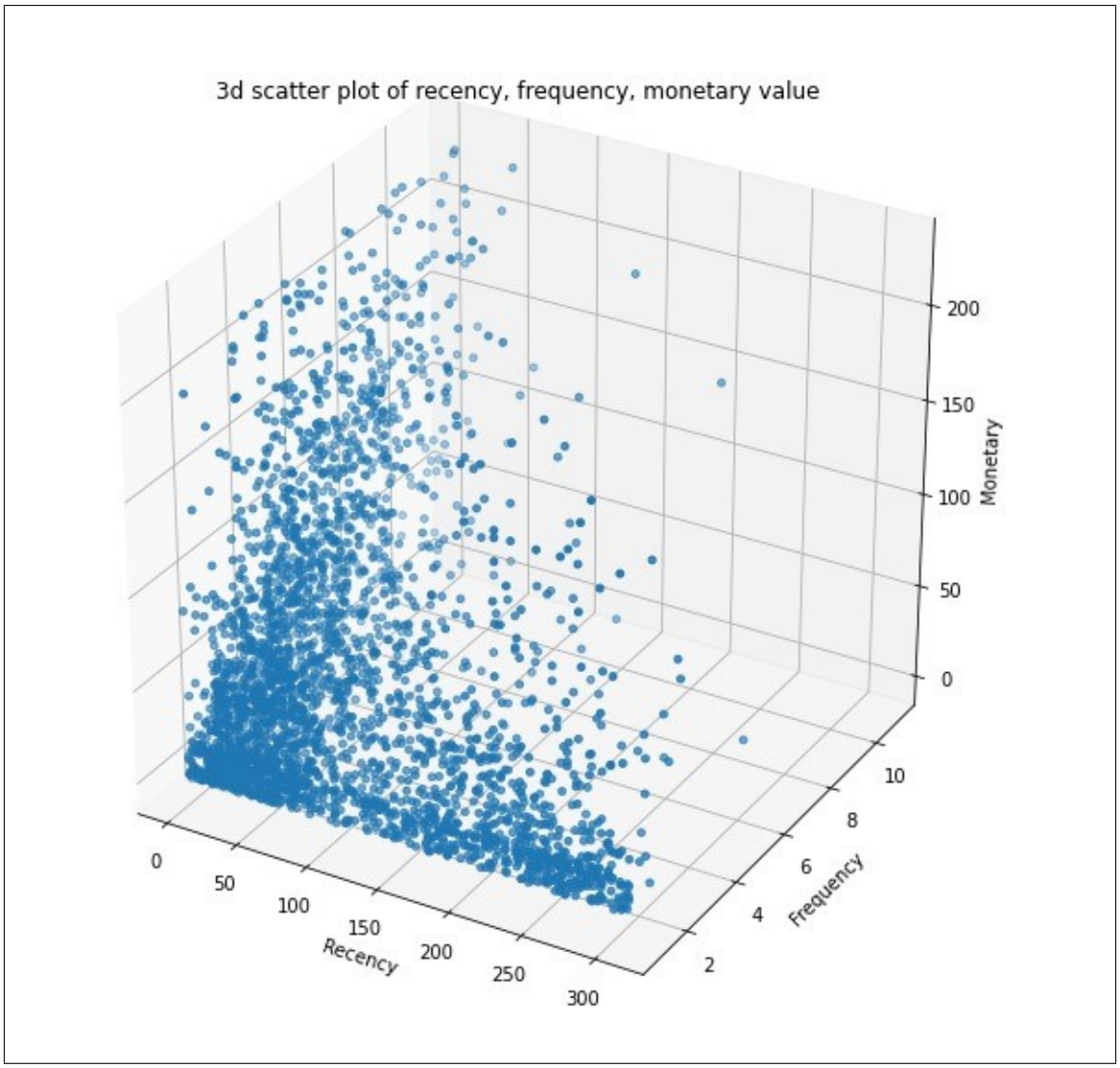


Figure 4.4: 3D scatter diagram of RFM columns

3D scatter plot of recency, frequency, monetary RFM calculated on customer purchase dataset. Recency range 0-350. Frequency range 1-11. Monetary range 0-250.

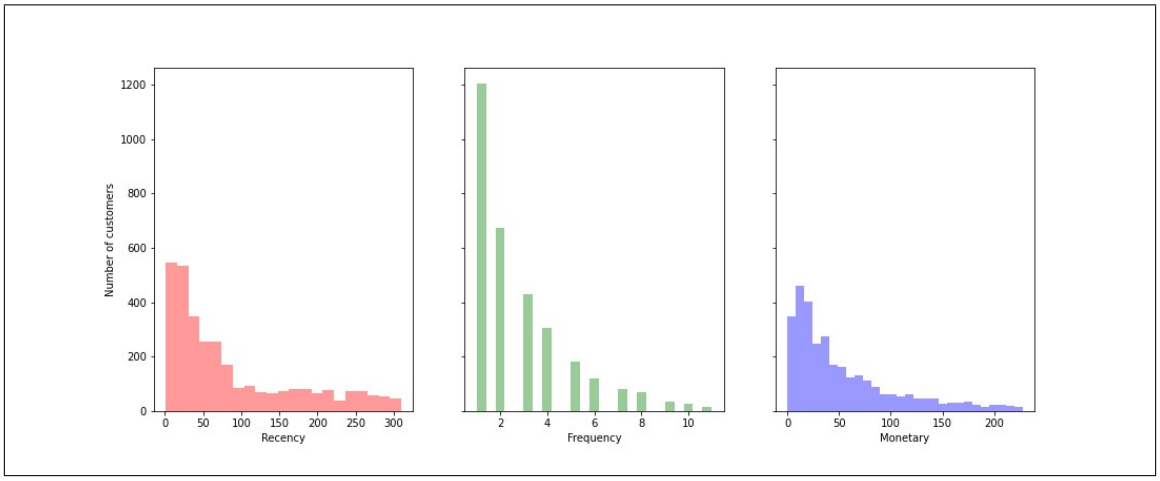


Figure 4.5: Histogram of RFM columns VS number of customers

Histograms of recency, frequency, monetary RFM calculated on customer purchase dataset. All RFM values are set over a twelve-month period.

Chapter 5

Customer Screening

5.1 Recurrent Neural Network (RNN)

When a human performs a task, he/she needs to follow a series of procedures which needs to be accomplished sequentially in order to obtain a reliable output of the task given. If there are 8 steps required to complete a task then in order to finish task 'n' the task 'n-1' needs to be completed and its output needs to be forwarded to the next task and so on [1]. This implies that there has to be a memory space present to carry information of subsequent tasks.
Traditional neural networks might be very efficient in providing an output but their drawbacks arise when their objectives are divided into a sequence and has to be completed serially. The absence of a memory space has led to such an issue. To deal with such a problem Recurrent Neural Networks (RNNs) have been introduced. Such a network consists of a loop which enables information to be persistent.

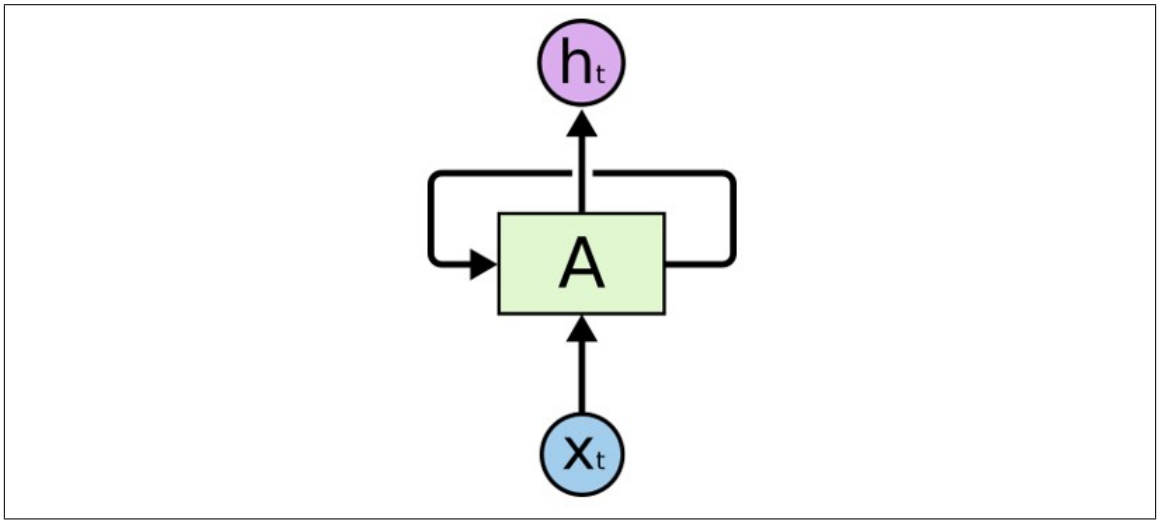


Figure 5.1: Recurrent Neural Network

The following diagram represents how an input xt entered into the neural network R provides an output h_t and also has a loop to transfer the results from one process to another. Thus, a series of these networks connected together and transferring the output of each module to the next module results in forming a recurrent neural network. Forming such a network allows RNNs to overcome the issue related to

having a memory space which was unavailable in traditional neural networks [1]. This network can be implemented in various fields such as language modeling, speech recognition etc. In addition to this, the implementation of LSTMs has solved problems that has been an obstacle for recurrent neural networks.

5.2 Drawbacks of RNN

We know RNNs to be an excellent network for creating a connection between several networks and for using loops. However, the depth of this memory stored in quite low. If there is a huge number of a recurrent neural network present within a system then there is a high chance that the 8th neuron will not be able to obtain the output of the 2nd neuron. Problems like these arise when two distant neurons require the output of either one of them. This problem is addressed by the term "vanishing gradient problem" and is a major issue of the recurrent neural network.

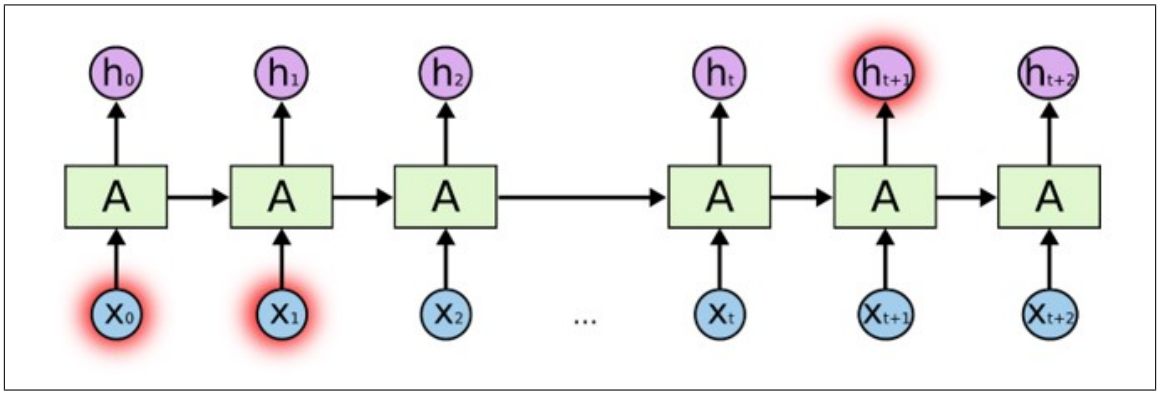


Figure 5.2: Long Short Term Memory

Although theoretically, such problems can be handled using RNNs, this is not always the case during practice. Thus, in order to resolve such issues LSTMs have been introduced. LSTM is a certain kind of RNN which specializes in learning dependencies that last for a longer period of time. Hochreiter and Schmidhuber (1997) proposed the idea of LSTMs and it has been enhanced by further researchers over the years. LSTMs are generally designed to form a series of modules that are involved with repetitions within the neural network.

5.3 Difference between LSTM and RNN

The internal architecture of a RNN is simpler than the structure of an LSTM. The repeating modules are connected together and each module has one tanh layer. On the other hand LSTMs have repeating modules but with a more complex structure having one neural network layer, four layers are present inside them.

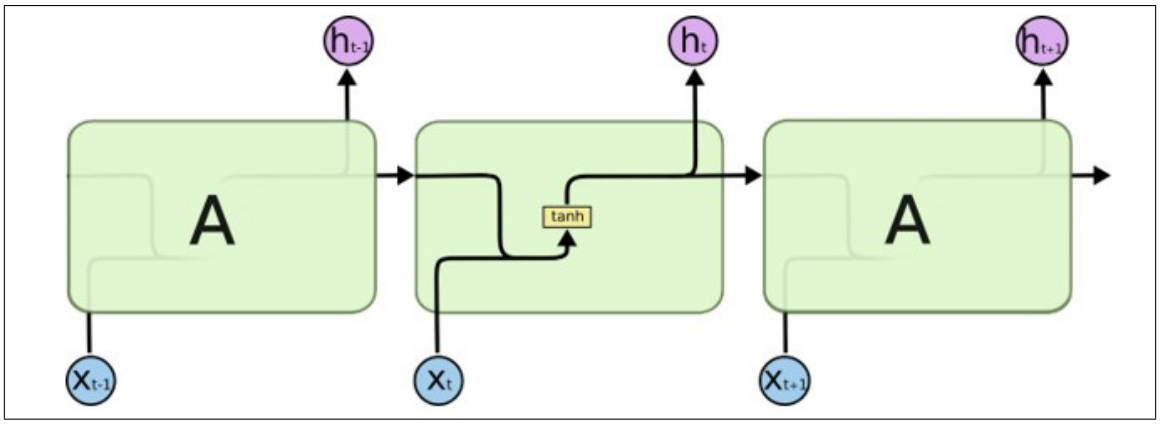


Figure 5.3: Repeating module comprising of a single layer

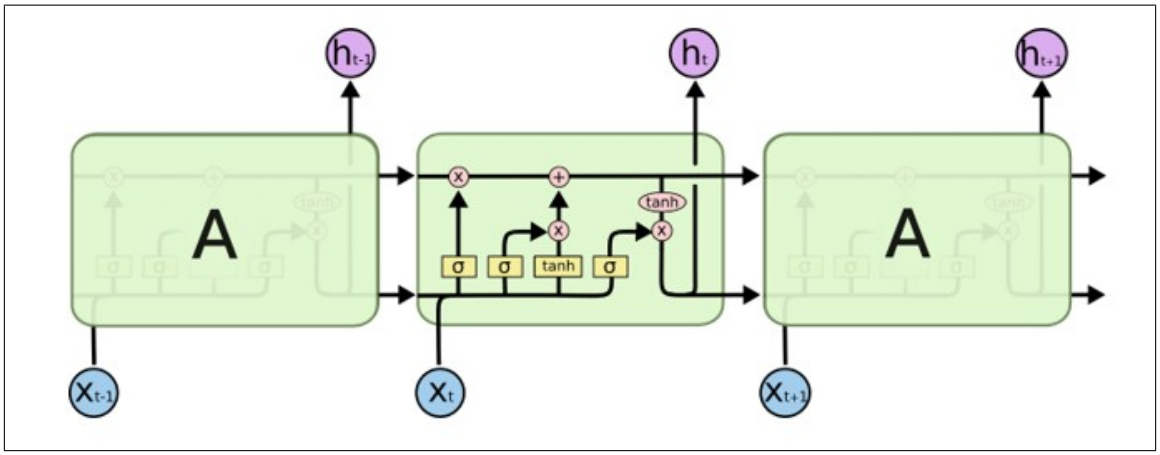


Figure 5.4: Repeating module of an LSTM comprising of four interacting layers

The figures above shows that LSTMs have a cell state which is absent inside an RNN. The cell state is the line that passes horizontally through the top portion of the module structure and is the primary factor that differentiates an LSTM from a RNN. Using gates, an LSTM can control the availability of the information present in the cell state. The gates are comprised of a sigmoid layer with a pointwise multiplication operation which results in a number from zero to one indicating the percentage of data that the gate will allow to pass through it. Zero depicts that no data will be allowed to pass through while one means all data will be transferred through the gate. Thus the sigmoid layer is also referred to as a "forget gate layer".

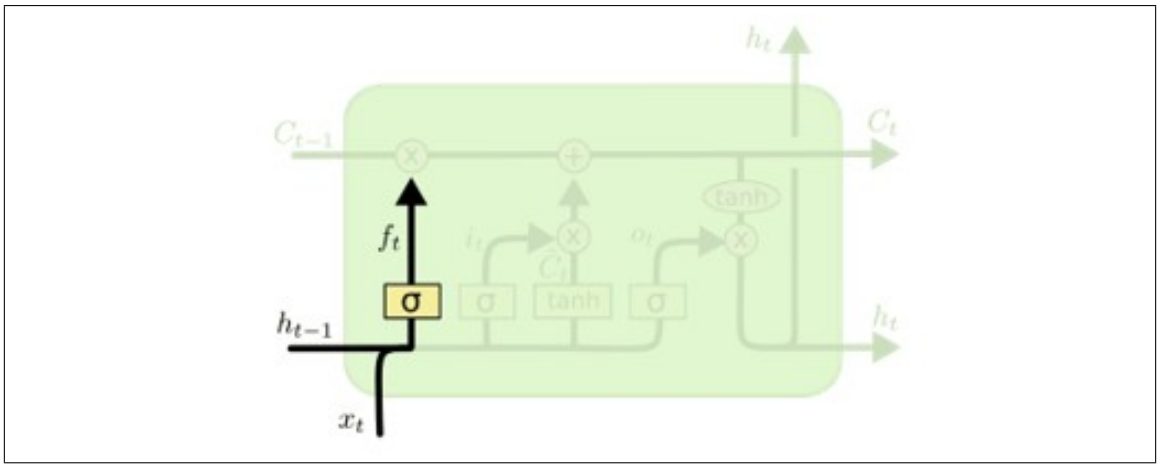


Figure 5.5: The Forget Gate Function

$$f_t = \sigma\left(W_f \cdot [h_{t-1}, x_t] + b_f\right) \tag{5.1}$$

After a new module is selected with new data, the old data is screened and the required information is passed through. For that, the following function mentioned above is selected to determine the amount of data to be preserved within the cell state. The sigmoid layer and the tanh layer results in an updated set of information, $C\ t$, which is then combined with the new data.

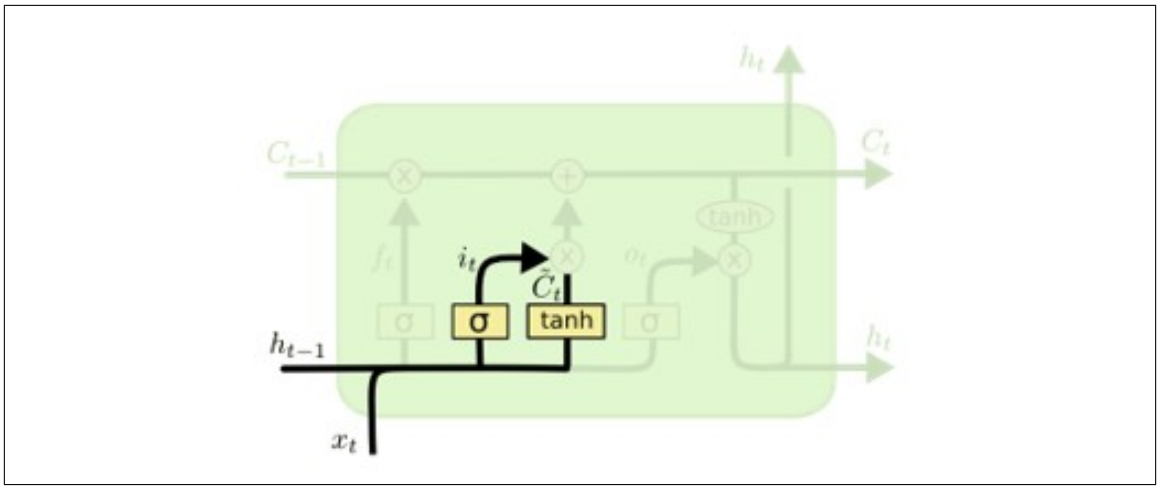


Figure 5.6: Updated Cell State and Input Function

$$i_t = \sigma\left(W_i \cdot [h_{t-1}, x_t] + b_i\right) \tag{5.2}$$

$$\tilde{C}_t = \tanh\left(W_C \cdot [h_{t-1}, x_t] + b_C\right) \tag{5.3}$$

The product of the value of the previous state and the function f_t added with the product of the updated value and the input leads to a new cell state which is presented as C_t.

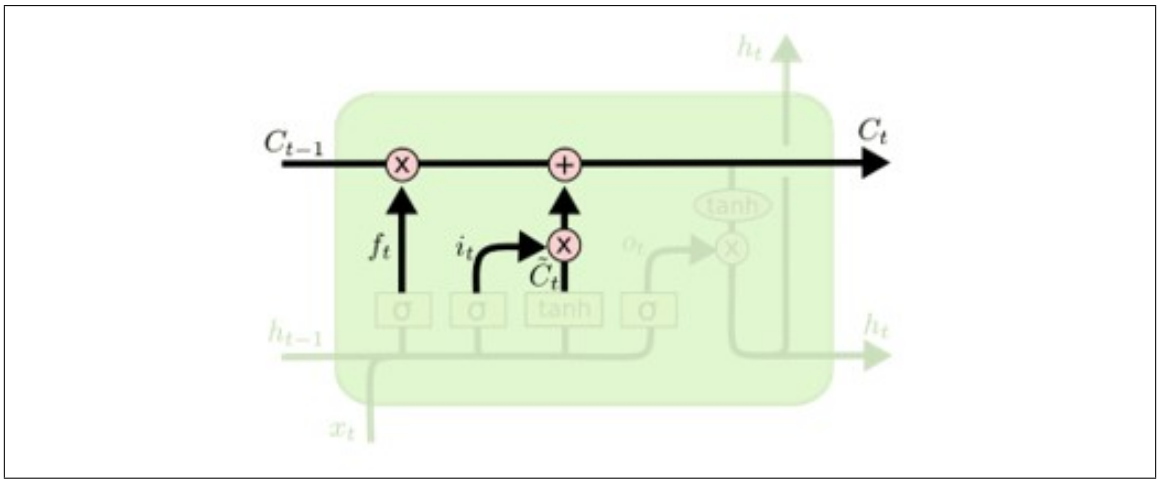


Figure 5.7: New cell state

$$C_t = f_t * C_{t-1} + i_t * \tilde{C}_t \tag{5.4}$$

This value is then screened by passing it through a sigmoid layer that determines the portion of the cell state, o_t that will be transferred to the next module. It is then multiplied with the value of the new cell state passed through the tanh layer to obtain h_t as the output.

5.4 Bidirectional LSTM

This is a special form of LSTM that is involved with the processing of information from two different directions [2]. It is composed of two hidden layers where one of the layers handles operations of the sequential input in the forward direction while the other layer handles the operations of the input in the backward direction (Di Wang and Eric Nyberg, 2015). The resulting value is obtained by adding the hidden vectors of the two layers [16].

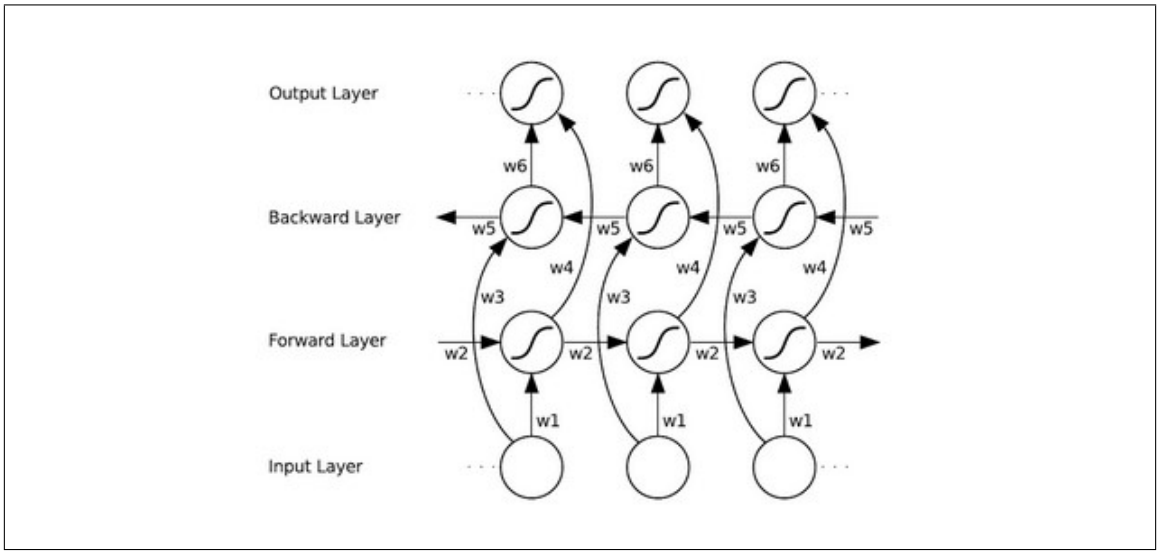


Figure 5.8: Bidirectional LSTM

Due to limitations of RNN discussed earlier, Bidirectional LSTM is used for binary classification of the customers in the initial stage. Binary classification is to classify customers into two distinct groups. The classification is done to reconnect with the customers who purchased from the business a long time ago and are likely to churn. Business needs to continuously find ways to promote their services and products to new and existing customers so that their product can capture market share in today's competitive market. However, marketing and promotion for product awareness can be costly. As discussed earlier, it is easier to retain existing customers than to attract new customers to increase the revenue of the business. As a result, business should focus on reconnecting with the existing customers for a better relationship in future. Hence, primary customer classification is important.
In the data preprocessing section we have discussed how we have calculated the RFM values and how the values were made discrete. Another further step conducted for binary classification is to binarization. From the RFM values another feature extracted into binary form of 0 and 1. '0' indicates that the business needs to implement the reconnecting measures as the customers are likely to churn and '1' indicates that the customers are less likely to churn in the short term.

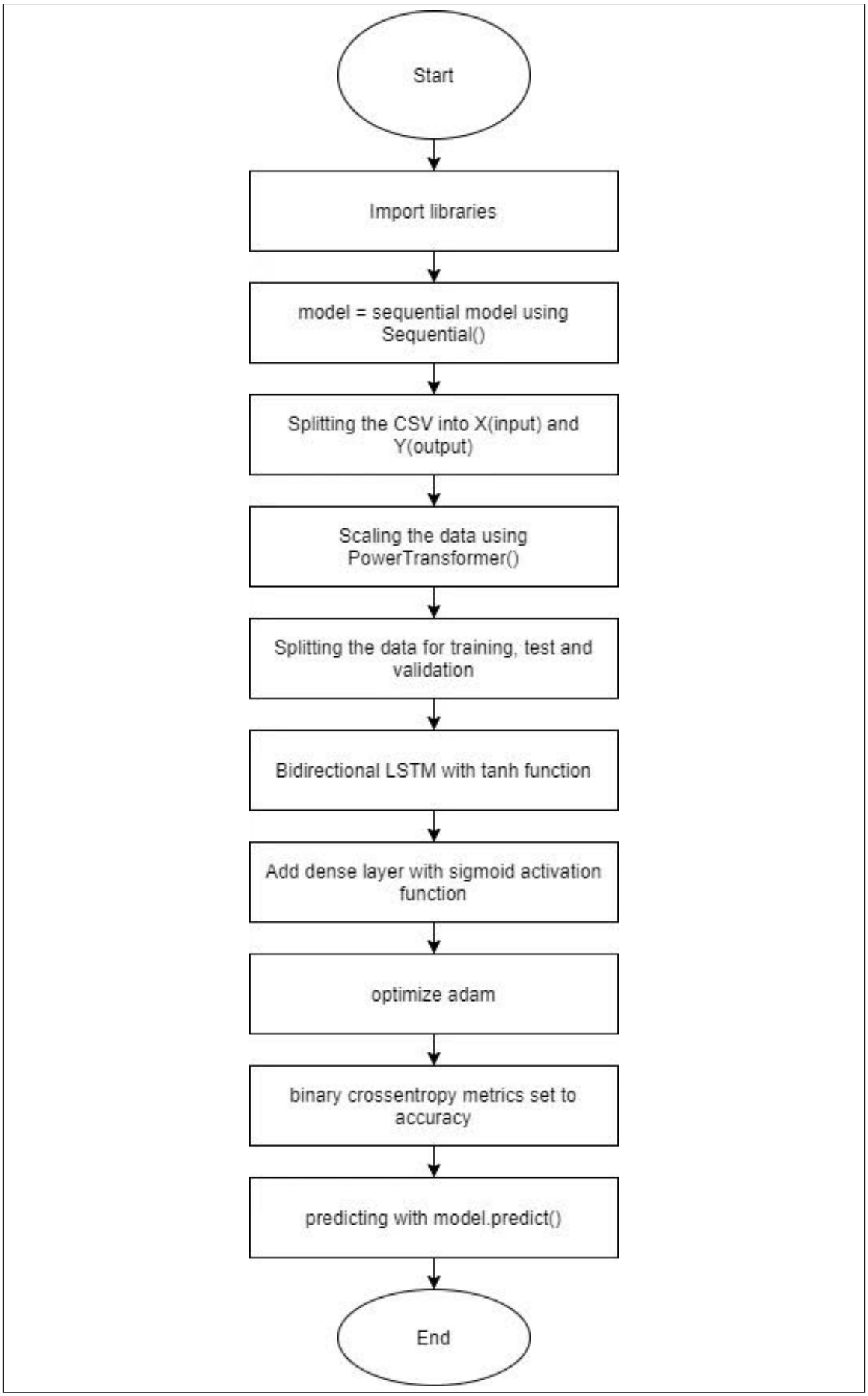


Figure 5.9: Work Flow Diagram for Customer Classification

```
Model: "sequential"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
bidirectional (Bidirectional (None, 2940, 100)         21600
_________________________________________________________________
dense (Dense)                (None, 2940, 1)           101
=================================================================
Total params: 21,701
Trainable params: 21,701
Non-trainable params: 0
```

Figure 5.10: Model summary of bidirectional LSTM for binary classification

```
103/103 [==============================] - 5s 13ms/step - loss: 0.6029 - accuracy: 0.8576 - val_loss: 0.3375 - val_accuracy: 0.9558
Epoch 2/100
103/103 [==============================] - 0s 3ms/step - loss: 0.2813 - accuracy: 0.9682 - val_loss: 0.1761 - val_accuracy: 0.9558
Epoch 3/100
103/103 [==============================] - 0s 3ms/step - loss: 0.1512 - accuracy: 0.9732 - val_loss: 0.1264 - val_accuracy: 0.9580
Epoch 4/100
103/103 [==============================] - 0s 4ms/step - loss: 0.1074 - accuracy: 0.9745 - val_loss: 0.1074 - val_accuracy: 0.9592
Epoch 5/100
103/103 [==============================] - 0s 3ms/step - loss: 0.0859 - accuracy: 0.9741 - val_loss: 0.0988 - val_accuracy: 0.9603
Epoch 6/100
103/103 [==============================] - 0s 3ms/step - loss: 0.0876 - accuracy: 0.9617 - val_loss: 0.0949 - val_accuracy: 0.9603
Epoch 7/100
103/103 [==============================] - 0s 3ms/step - loss: 0.0758 - accuracy: 0.9652 - val_loss: 0.0932 - val_accuracy: 0.9580
Epoch 8/100
103/103 [==============================] - 0s 3ms/step - loss: 0.0704 - accuracy: 0.9726 - val_loss: 0.0925 - val_accuracy: 0.9592
Epoch 9/100
103/103 [==============================] - 0s 3ms/step - loss: 0.0602 - accuracy: 0.9774 - val_loss: 0.0929 - val_accuracy: 0.9592
Epoch 10/100
103/103 [==============================] - 0s 3ms/step - loss: 0.0605 - accuracy: 0.9770 - val_loss: 0.0928 - val_accuracy: 0.9580
Epoch 11/100
103/103 [==============================] - 0s 3ms/step - loss: 0.0577 - accuracy: 0.9785 - val_loss: 0.0931 - val_accuracy: 0.9580
Epoch 12/100
103/103 [==============================] - 0s 3ms/step - loss: 0.0600 - accuracy: 0.9724 - val_loss: 0.0938 - val_accuracy: 0.9569
Epoch 13/100
103/103 [==============================] - 0s 3ms/step - loss: 0.0623 - accuracy: 0.9681 - val_loss: 0.0938 - val_accuracy: 0.9603
```

Figure 5.11: Epoch value with measure of loss and accuracy

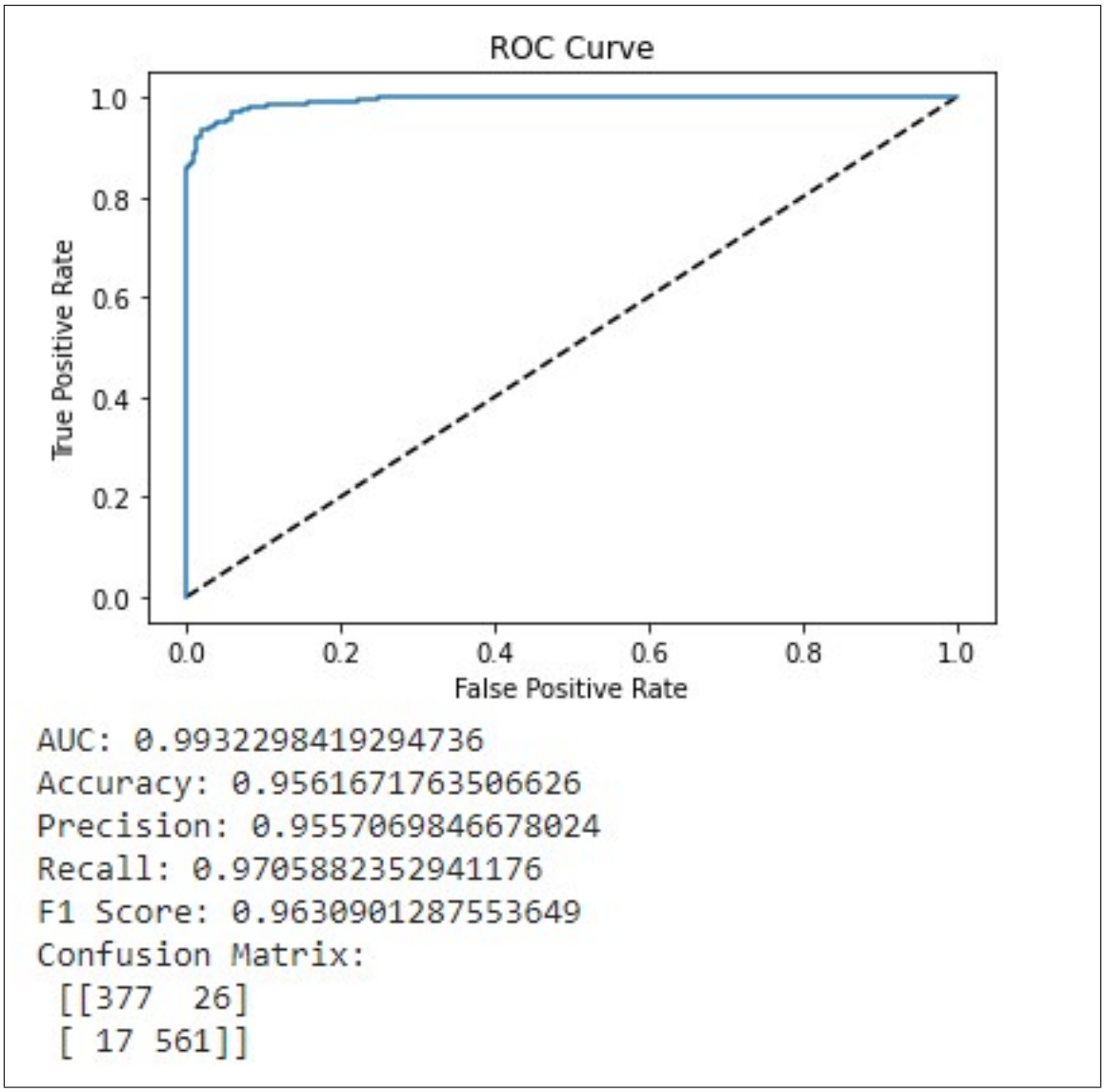


Figure 5.12: ROC curve

Chapter 6

Customer Segmentation

6.1 K Nearest Neighbour

K-nearest neighbor is used for classification and regression predictive problems. KNN is mostly used for classification problems. Important aspect of KNN is ease to interpret output, calculation time, predictive power.
K nearest neighbor is a classification approach where it groups objects that are nearby to the test value. To classify an unlabeled object, the distance between this object and labelled object is computed and it is K nearest neighbors are identified. Classification accuracy mainly depends on the chosen value of K and will be better than that of using the nearest neighbor classifier. For big data sets, K can be larger to lower the error rate. Choosing K can be done randomly, where a number of patterns taken out from the training set can be classified using the remaining training patterns for different values of k. The value of K which will give the least error in classification will be chosen. If same class is shared between several of K-nearest neighbors, then per-neighbor weights of that class are added together, and the resulting weighted sum is used as the likelihood score of that class with respect to the test dataset. A ranked list is obtained for the test document by sorting the scores of candidate classes. Decision rule for KNN can be written as:

$$\sum_{d_j \in KNN(d)} \mathrm{Sim}\,(d, d_j)\,\delta\,(d_j, c_i) \tag{6.1}$$

Where d is the test document, c_i indicates the classes of KNN which is used by the system to find K-nearest neighbors among training documents, $KNN(d)$ is the set of K-nearest neighbors of document $d, d(d_j, c_i)$ is the classification for document d_j with respect to class c_i , that is the value of $\delta(d_j, c_i)$ Will be 1 if d_j is an element of class c_i , Else it will be 0 For the test document d, it should be assigned the class that has highest resulting weighted sum. The classification of KNN is easy to understand and implement and it can perform well in many situations. It is also scalable to new modifications as it is possible to eliminate many of the stored data objects, but still retain the classification accuracy of the KNN classifier. If K is too small then result can be sensitive to noise points whereas if for large value of K, the neighborhood may include too many points from other classes. The choice to distance measure is another important consideration. Although various measures can be used to compute the distance between two points, but smaller distance between two objects does not always implies a greater likelihood of having the same class.

Based on the distance value between customers, KNN classifier is used to segment customers based on recency, frequency, and monetary (RFM). The columns values are transformed into ordinal values by converting the value to distinct labels of one, two, three and four based on the quartile threshold of each column.
Higher value for recency indicates that the customer purchased from the business quite a long time ago; this is considered to be a bad factor and thus, it is segmented as 4. If the value of the recency column is greater than or equal to the 1st quartile value of the column, then it is classified as 1 indicating the customer has bought from the business recently and is a good factor.
Higher value of frequency and monetary value indicates that the customer has been purchasing from the business multiple times and has contributed to an increase in revenue for the business; as a result, it is a considered to be a good indicator and is labeled as 1 if the value of frequency is above the 3rd quartile of the data of the column.

CustomerID	recency	frequency	monetary_value	new_recency	new_frequency	new_moneytary value
12346.0	325	1	77183.60	4	4	1
12747.0	2	103	4196.01	1	1	1
12748.0	0	4596	33719.73	1	1	1
12749.0	3	199	4090.88	1	1	1
12820.0	3	59	942.34	1	2	2

Figure 6.1: RFM to segment the customers

Based on the above factors, another column named “segmentation” was introduced to label customers as loyal, regular, occasional, and churned customers.

CustomerID	recency	frequency	monetary_value	Segmentation
12346.0	325	1	77183.60	Ocassional buyers
12747.0	2	103	4196.01	Loyal customers
12748.0	0	4596	33719.73	Loyal customers
12749.0	3	199	4090.88	Loyal customers
12820.0	3	59	942.34	Repeat customers
12821.0	214	6	92.72	High risk of churning
12822.0	70	46	948.88	Ocassional buyers
12823.0	74	5	1759.50	Ocassional buyers
12824.0	59	25	397.12	Ocassional buyers
12826.0	2	91	1474.72	Repeat customers

Figure 6.2: Segmentation of Customers using KNN

The business data we used sells normal goods and the retail price of the good is quite low. According to the above table, the customer is segmented as occasional buyer because it bought once and maybe purchased bulk amounts. However, loyal customers have high RFM values. Thus, all three RFM columns are essential for segmentation of customers.
The segmentation column is then changed into labels 1,2,3,4 to feed into the model.

The workflow diagram of KNN:

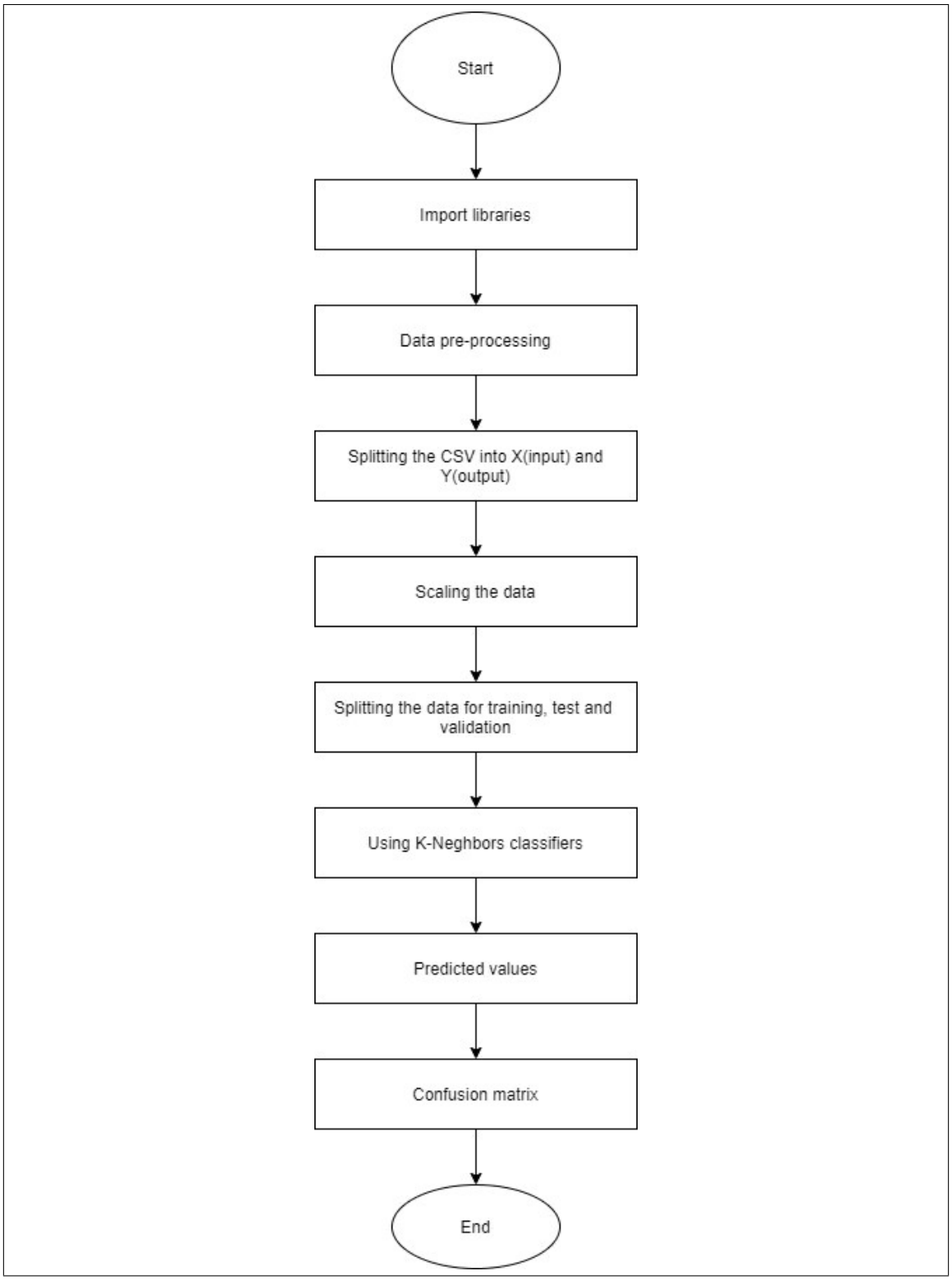


Figure 6.3: Workflow Diagram of KNN

```
[[107  11   0   0]
 [ 18 312  14   0]
 [  0  18 335  18]
 [  0   0  20 324]]
              precision    recall  f1-score   support

         1.0       0.86      0.91      0.88       118
         2.0       0.91      0.91      0.91       344
         3.0       0.91      0.90      0.91       371
         4.0       0.95      0.94      0.94       344

    accuracy                           0.92      1177
   macro avg       0.91      0.91      0.91      1177
weighted avg       0.92      0.92      0.92      1177
```

Figure 6.4: Results obtained from the training and testing of the data

6.2 Support Vector Machine

SVM or Support Vector Machine is a supervised machine learning classification algorithm. This algorithm is slightly different from the other algorithms in machine learning in terms of how it operates. When we wish to linearly separate data, most machine learning algorithms will look for and find a boundary that will separate the data points such that the misclassification error is low.
However, SVM places the decision boundary in a special way, such that, the boundary has the maximum distance from the nearest points in each class or segment. This is regarded as the optimal decision boundary with the maximum margin from the nearest class. In SVM, the decision boundary is also called the maximum margin classifier or the maximum hyper plane.
An SVM constructs a hyperplane in a high or infinite dimensional plane that can be used for classifications. If any value falls on one side of the plane the classifier will give one result and if it falls on the other side the classifier gives a different result.

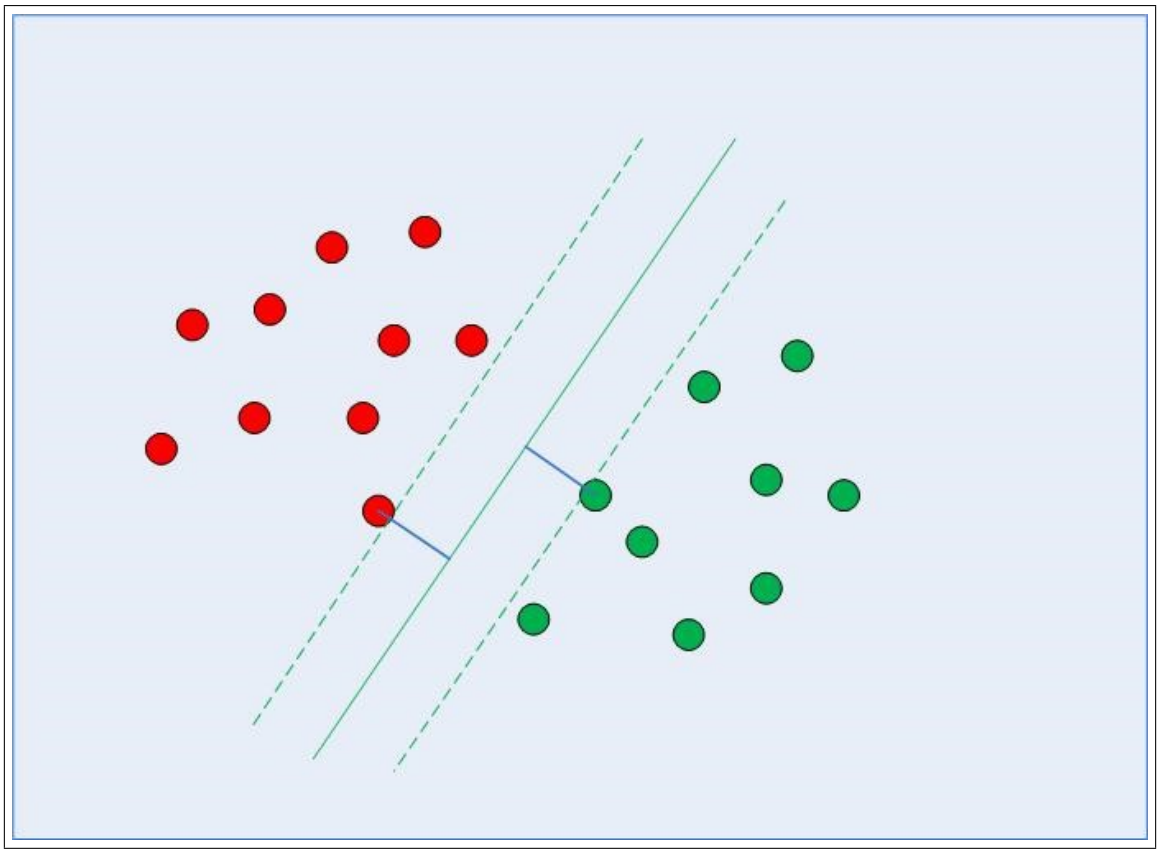


Figure 6.5: How a Simple SVM takes the Maximum Margin Classifier

The simple Support Vector Machine is used to linearly classify data into different classes. If we wish to work with non-linear data and classify it, then a more advanced version of SVM is used which is known as kernel SVM. Here, the data is projected from a lower dimension to higher dimension, linearly separable data in a way so that the different classes occupy different dimensions.
In our paper, we have used Kernel SVM to classify customers into different groups. We used three types of kernels to run the algorithm and see how the accuracy varies and determined which particular kernel we would suggest is used in the future. The three kernels we have used are:

1. Gaussian Radial Basis Function Kernel
2. Linear Kernel
3. Sigmoid Kernel

Kernel SVM

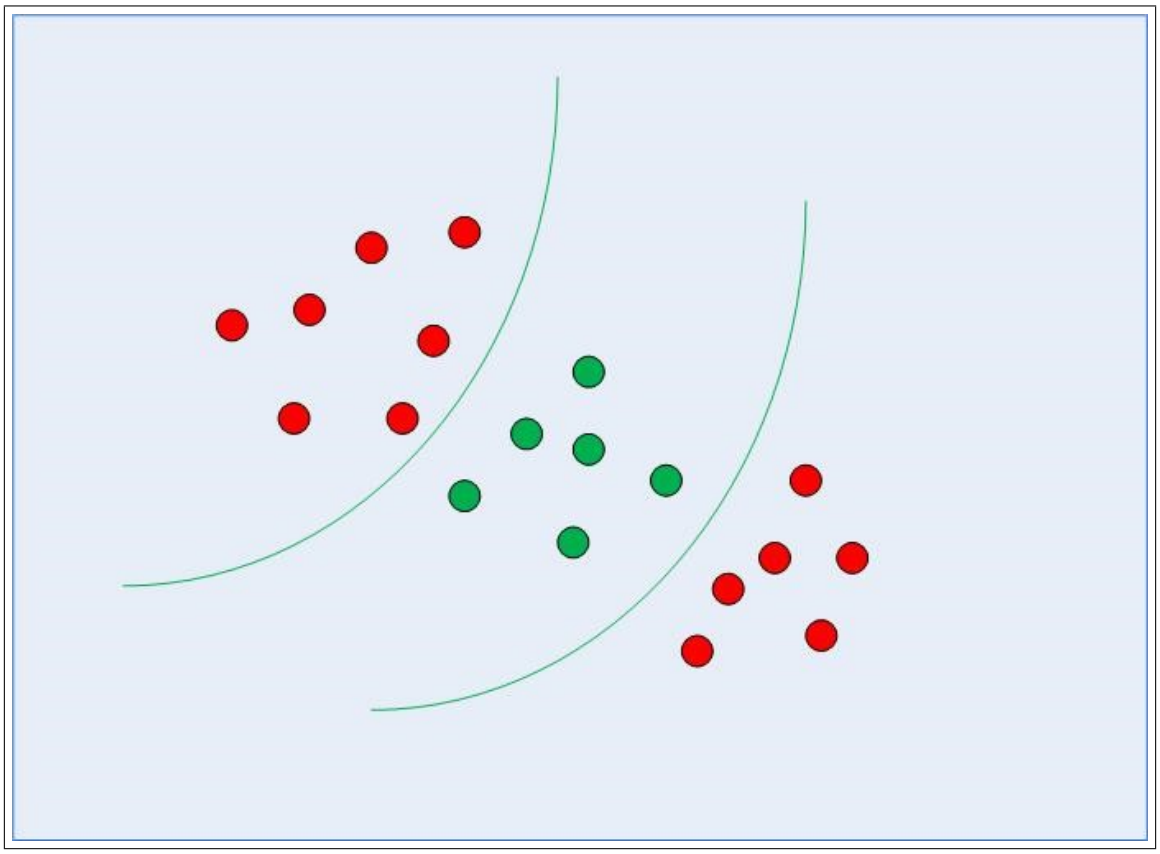


Figure 6.6: How a Kernel SVM Functions

Since the SVM only returns two classes, we slightly modified our data so it could be run. We originally had 4 distinct classes into which we could divide the customers. We first subtracted 1 from the number of classes to bring the range down to 0-3. Next, we converted the class numbers to binary in the range of 00-11. We then split the two bits into separate result fields and ran the kernel SVM algorithm with sigmoid as the kernel. After doing all the preprocessing, we still realized that the accuracy was quite low. The reason why this might have happened is that when we tried to find out which class the model predicts our customer is in, we had to concatenate two predictions from separate SVM models and then convert it to decimal and add a 1 to it. When we concatenate a correct value with an incorrect value, we essentially lose a correct value that our model has predicted. So, the accuracy value we have received from our model is slightly higher than the true value of accuracy which we later found out when we visualized the predicted values vs real values in a pie chart.
The accuracy that we have received from the support vector machine algorithm that we have ran are:

1. Gaussian Radial Basis Function Kernel: 77.80%
2. Linear Kernel: 69.60%
3. Sigmoid Kernel: 65.45%

A linear kernel is used when the data is easily dividable by a straight line. It is faster than any other kernels and it is the most commonly used kernel. However, in our experiment we have seen that the accuracy given by the linear kernel is not the highest so it is not a part of the suggested list of kernels to use with the SVM algorithm.
The Hyperbolic Tangent or Sigmoid Kernel is one such kernel which is also referred to as the Multilayer Perceptron (MLP) Kernel. This kernel comes from the Neural Networks field where we often see that bipolar sigmoid functions are being used as activation functions for artificial neurons. A notable fact is that the SVM model with a sigmoid kernel function can also be treated as a two-layer perceptron neural network. Study shows that it has performed well in a lot of cases but unfortunately in our case the SVM model with Hyperbolic Tangent kernel performed the worst, making it the most inaccurate model of the lot.
The following equation is used for Sigmoid Kernels:

$$K(X_1, X_2) = \tanh(\alpha x^T y + c) \tag{6.2}$$

The gaussian radial basis function is the most generalized way of kernelization. It uses sigma (variance) as the hyperparameter and the Euclidean distance between two points x_1 and x_2 is used as a part of the equation. If the Euclidean distance is d_12 is close to 0 we can say the points are similar and might belong to the same class. When they are far apart it is regarded that the points are in different classes. The hyperparameter will determine how large the region of similarity is. The larger the value of σ, the larger the region of similarity.
The following equation is used for Gaussian Radial Basis Function:

$$K(X_1, X_2) = e^{-\frac{d_{12}}{2\sigma^2}} \tag{6.3}$$

Our suggested kernel to use, if SVM is used to classify the customers, is to use the Gaussian Radial Basis Function ("rbf") function due to the higher level of accuracy even though it takes a while to compute.

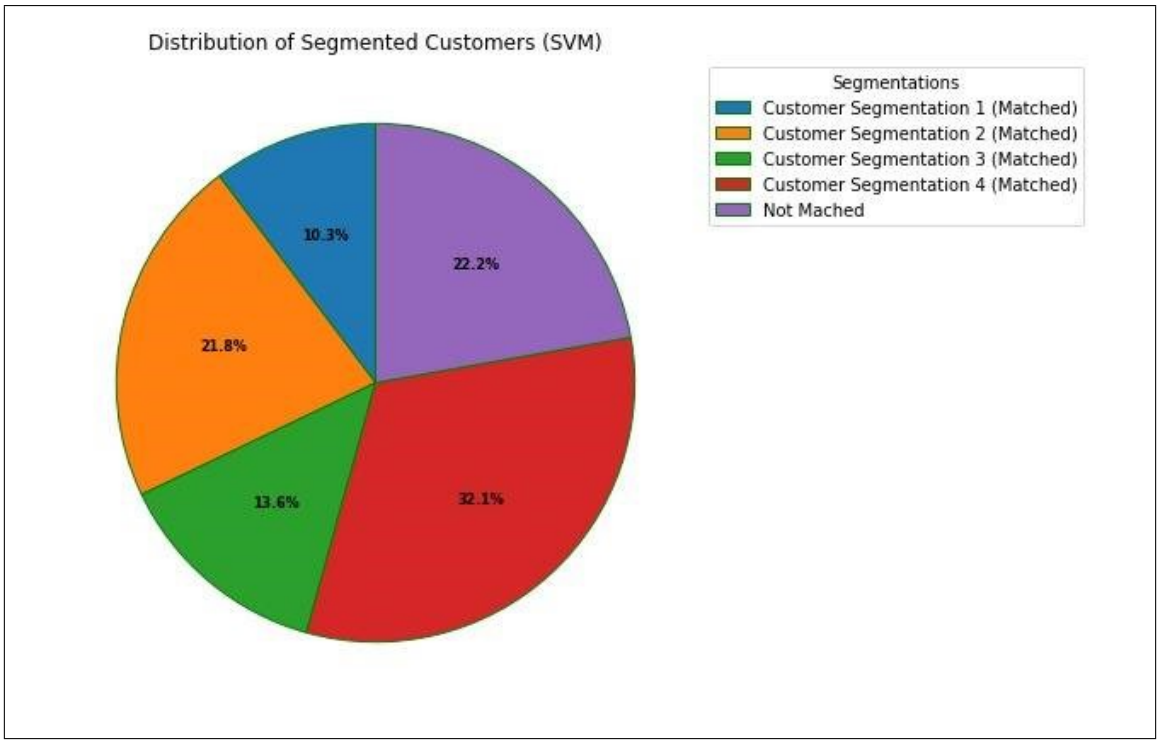


Figure 6.7: Distribution of Segmented Customers

6.3 Multilayer Perceptron

Feedforward neural networks are considered to be the base structure of the various deep learning models that are available to us till date. A few examples of such networks can be CNNs and RNNs which are used to utilize machine learning in predicting results that have a target function which is known before-hand. A learning of such type is referred to as supervised machine learning.
A feedforward neural network is expected to generate a function f* which will perform mapping on an input value x resulting in a new value y. The purpose of a feedforward network is to process the information it has been fed to define a mapping of $y = f(x; \theta)$ where θ is the parameter the network uses to generate a reliable output function. This network consists of three layers - the input layer, the hidden layer and the output layer.

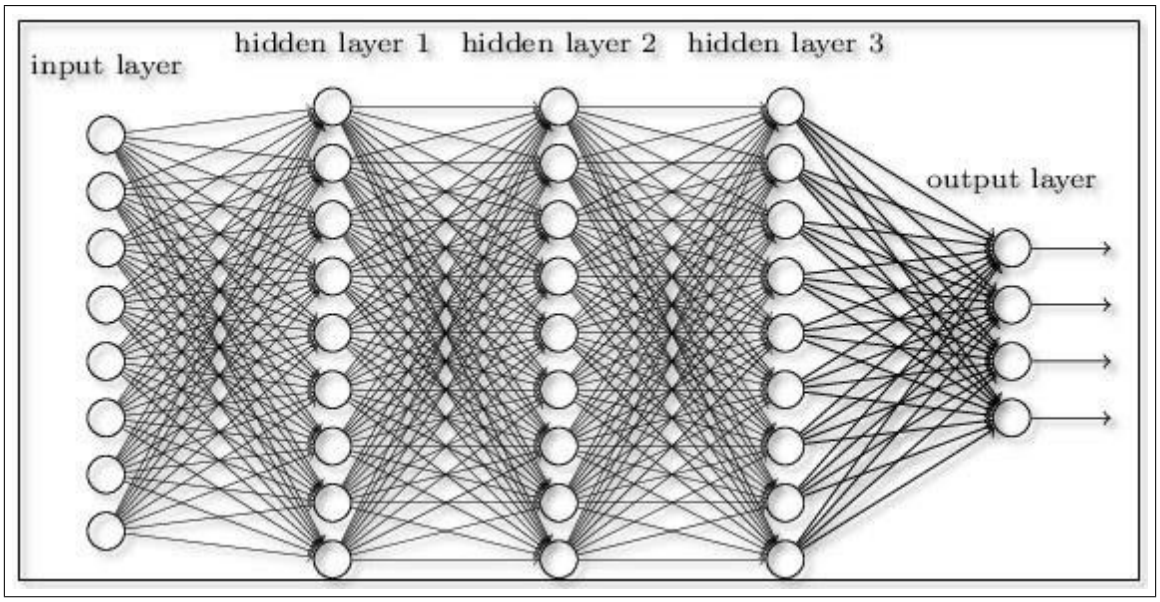


Figure 6.8: How MLP Works

The hidden layers existing between the input and output layers are responsible for generating unique outputs using the inputs, weights and biased values from their predecessor neurons. The results produced by the hidden layers are not displayed as their outputs are used as inputs for the output layer and thus these layers are referred to as hidden layers [12]. A network can consist of multiple hidden layers depending on the complexity of its functions.

Functionalities of MLP

A multilayer perceptron consists of layers of perceptrons stacked together to determine the solution of problems that require complex calculations. Every perceptron on the input layer uses different weights (and biases, if any) to transmit signals to the next perceptron. The result is displayed by the perceptrons in the output layer. Every layer can have a significant number of perceptrons and there can be various layers thus making the overall system extremely complex. The benefits of an MLP over normal neural networks is that it can utilize activation functions other than the decision function (it is a step function) and the output generated can be between 0 and 1 or between -1 and 1 (in a normal neural network the output is always binary) [13].
Perceptrons in different layers all undergo these simple procedures to generate the output:

1. The perceptrons take inputs and calculate the sum using the product of their weights and inputs

2. A bias factor is sometimes associated with the weight to move the activation function up, down, left, or right on the number graph

3. The calculated sum is passed through the activation function which later maps the values entered in the input to output values

4. Steps 1,2 and 3 are repeated until the output layer is reached which generates the final result and then displays it

5. A loss function will calculate the deviation of the predicted value (generated output) from the expected output (real output) to determine whether backpropagation (a method that uses backpropagation algorithm) to update weights of the previous neurons as required to obtain a reliable output.

MLPs have enriched the computing power of computers to solve problems that require classification and regression by being able to understand complex functions and generating reliable results. Thus, this helps modern day computers to overcome most of their limitations and generate effective and reliable values of complex problems.

We have used the Multilayer Perceptron or MLP to segment our customers into the 4 categories. The MLP algorithm we have decided to use has 3 layers. The input layer uses an activation function of ReLU. The second layer also has the activation function of ReLU. The third and final output layer has an activation function of Sigmoid. We have chosen to use the Rectified linear Unit activation function mainly due to its computational efficiency. The sigmoid function was selected as it gives us a normalized output and makes sure there are no sudden changes in the output values.

When compiling the model, we have opted for a loss function of binary cross entropy and the adam optimizer. We ran the model with 100 epochs to get our desired segmented outputs.

Since we have used the binary cross entropy loss function, we needed to restructure our dataset to make sure that our model predicts a binary value. We opted for one hot encoding here. We subtracted 1 from the segment numbers to get a range of 0-3 from 1-4. We, then converted the values to binary and dis-concatenated the values (ranging from 00-11) to get individual columns of labels with only binary values. Next, we ran the MLP algorithm twice with the same hyper parameters and received two sets of output. We concatenated the output values to form the 2-bit binary result which we then converted to decimal. A 1 was added to this decimal value result to end up with the desired output of segmented customers.

Our predicted output had an accuracy of about 89% which was a good result because it could give us such an accuracy with just 100 epochs in a relatively short amount of time.

Work Flow Diagram of MLP

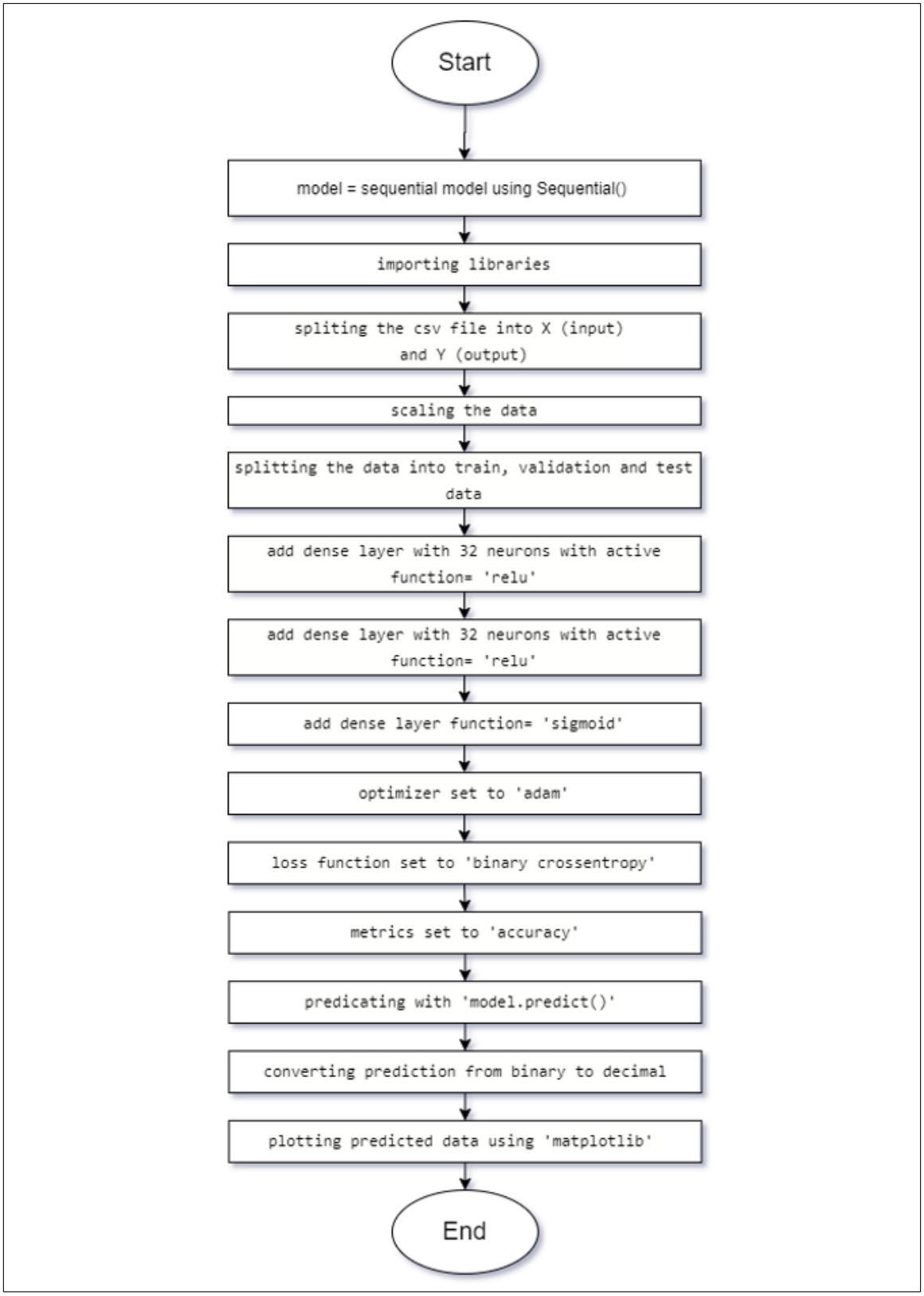


Figure 6.9: Work Flow Diagram of MLP

Below are some screenshots of the MLP algorithm running:

```
Epoch 1/100
86/86 [==============================] - 1s 7ms/step - loss: 0.6220 - accuracy: 0.6463 - val_loss: 0.5175 - val_accuracy: 0.8299
Epoch 2/100
86/86 [==============================] - 0s 2ms/step - loss: 0.5009 - accuracy: 0.8042 - val_loss: 0.4231 - val_accuracy: 0.8265
Epoch 3/100
86/86 [==============================] - 0s 2ms/step - loss: 0.4388 - accuracy: 0.8011 - val_loss: 0.3826 - val_accuracy: 0.8282
Epoch 4/100
86/86 [==============================] - 0s 3ms/step - loss: 0.3994 - accuracy: 0.8167 - val_loss: 0.3698 - val_accuracy: 0.8146
Epoch 5/100
86/86 [==============================] - 0s 2ms/step - loss: 0.3736 - accuracy: 0.8301 - val_loss: 0.3465 - val_accuracy: 0.8486
Epoch 6/100
86/86 [==============================] - 0s 2ms/step - loss: 0.3776 - accuracy: 0.8101 - val_loss: 0.3303 - val_accuracy: 0.8656
Epoch 7/100
86/86 [==============================] - 0s 2ms/step - loss: 0.3677 - accuracy: 0.8244 - val_loss: 0.3184 - val_accuracy: 0.8639
Epoch 8/100
86/86 [==============================] - 0s 2ms/step - loss: 0.3384 - accuracy: 0.8382 - val_loss: 0.3045 - val_accuracy: 0.8776
Epoch 9/100
86/86 [==============================] - 0s 2ms/step - loss: 0.3296 - accuracy: 0.8517 - val_loss: 0.2849 - val_accuracy: 0.9014
Epoch 10/100
86/86 [==============================] - 0s 2ms/step - loss: 0.3207 - accuracy: 0.8660 - val_loss: 0.2691 - val_accuracy: 0.9099
Epoch 11/100
86/86 [==============================] - 0s 2ms/step - loss: 0.2892 - accuracy: 0.8898 - val_loss: 0.2631 - val_accuracy: 0.9065
Epoch 12/100
86/86 [==============================] - 0s 2ms/step - loss: 0.2787 - accuracy: 0.8949 - val_loss: 0.2354 - val_accuracy: 0.9405
Epoch 13/100
86/86 [==============================] - 0s 2ms/step - loss: 0.2487 - accuracy: 0.9291 - val_loss: 0.2313 - val_accuracy: 0.9354
Epoch 14/100
86/86 [==============================] - 0s 2ms/step - loss: 0.2346 - accuracy: 0.9400 - val_loss: 0.2081 - val_accuracy: 0.9218
Epoch 15/100
86/86 [==============================] - 0s 2ms/step - loss: 0.2270 - accuracy: 0.9314 - val_loss: 0.1936 - val_accuracy: 0.9371
Epoch 16/100
86/86 [==============================] - 0s 2ms/step - loss: 0.2154 - accuracy: 0.9329 - val_loss: 0.1819 - val_accuracy: 0.9371
Epoch 17/100
86/86 [==============================] - 0s 2ms/step - loss: 0.2027 - accuracy: 0.9389 - val_loss: 0.1757 - val_accuracy: 0.9456
Epoch 18/100
86/86 [==============================] - 0s 2ms/step - loss: 0.1877 - accuracy: 0.9431 - val_loss: 0.1690 - val_accuracy: 0.9303
Epoch 19/100
86/86 [==============================] - 0s 2ms/step - loss: 0.1709 - accuracy: 0.9467 - val_loss: 0.1629 - val_accuracy: 0.9388
Epoch 20/100
86/86 [==============================] - 0s 2ms/step - loss: 0.1624 - accuracy: 0.9498 - val_loss: 0.1632 - val_accuracy: 0.9439
Epoch 21/100
86/86 [==============================] - 0s 2ms/step - loss: 0.1645 - accuracy: 0.9491 - val_loss: 0.1568 - val_accuracy: 0.9439
```

Figure 6.10: MLP Running (1)

```
Epoch 75/100
86/86 [==============================] - 0s 2ms/step - loss: 0.4698 - accuracy: 0.7747 - val_loss: 0.5622 - val_accuracy: 0.7075
Epoch 76/100
86/86 [==============================] - 0s 2ms/step - loss: 0.4861 - accuracy: 0.7715 - val_loss: 0.4551 - val_accuracy: 0.7738
Epoch 77/100
86/86 [==============================] - 0s 2ms/step - loss: 0.4837 - accuracy: 0.7805 - val_loss: 0.4501 - val_accuracy: 0.7823
Epoch 78/100
86/86 [==============================] - 0s 2ms/step - loss: 0.4513 - accuracy: 0.7745 - val_loss: 0.4470 - val_accuracy: 0.7874
Epoch 79/100
86/86 [==============================] - 0s 2ms/step - loss: 0.4683 - accuracy: 0.7695 - val_loss: 0.4770 - val_accuracy: 0.7500
Epoch 80/100
86/86 [==============================] - 0s 2ms/step - loss: 0.4523 - accuracy: 0.7865 - val_loss: 0.4484 - val_accuracy: 0.8095
Epoch 81/100
86/86 [==============================] - 0s 2ms/step - loss: 0.4649 - accuracy: 0.7769 - val_loss: 0.4410 - val_accuracy: 0.7857
Epoch 82/100
86/86 [==============================] - 0s 2ms/step - loss: 0.4328 - accuracy: 0.7794 - val_loss: 0.4536 - val_accuracy: 0.7976
Epoch 83/100
86/86 [==============================] - 0s 2ms/step - loss: 0.4262 - accuracy: 0.7967 - val_loss: 0.4515 - val_accuracy: 0.7925
Epoch 84/100
86/86 [==============================] - 0s 2ms/step - loss: 0.4253 - accuracy: 0.8122 - val_loss: 0.6667 - val_accuracy: 0.7483
Epoch 85/100
86/86 [==============================] - 0s 3ms/step - loss: 0.5427 - accuracy: 0.7437 - val_loss: 0.4763 - val_accuracy: 0.7755
Epoch 86/100
86/86 [==============================] - 0s 2ms/step - loss: 0.4641 - accuracy: 0.7707 - val_loss: 0.4223 - val_accuracy: 0.7993
Epoch 87/100
86/86 [==============================] - 0s 2ms/step - loss: 0.4399 - accuracy: 0.7918 - val_loss: 0.4894 - val_accuracy: 0.7534
Epoch 88/100
86/86 [==============================] - 0s 2ms/step - loss: 0.4425 - accuracy: 0.7850 - val_loss: 0.4182 - val_accuracy: 0.7942
Epoch 89/100
86/86 [==============================] - 0s 2ms/step - loss: 0.4149 - accuracy: 0.7958 - val_loss: 0.4113 - val_accuracy: 0.7976
Epoch 90/100
86/86 [==============================] - 0s 2ms/step - loss: 0.4210 - accuracy: 0.8040 - val_loss: 0.5316 - val_accuracy: 0.7415
Epoch 91/100
86/86 [==============================] - 0s 2ms/step - loss: 0.4598 - accuracy: 0.7822 - val_loss: 0.4187 - val_accuracy: 0.8112
Epoch 92/100
86/86 [==============================] - 0s 2ms/step - loss: 0.4114 - accuracy: 0.8157 - val_loss: 0.4063 - val_accuracy: 0.8214
Epoch 93/100
86/86 [==============================] - 0s 2ms/step - loss: 0.3970 - accuracy: 0.8178 - val_loss: 0.4008 - val_accuracy: 0.8452
Epoch 94/100
86/86 [==============================] - 0s 2ms/step - loss: 0.3920 - accuracy: 0.8361 - val_loss: 0.4070 - val_accuracy: 0.7925
Epoch 95/100
86/86 [==============================] - 0s 2ms/step - loss: 0.3992 - accuracy: 0.8077 - val_loss: 0.4051 - val_accuracy: 0.8503
Epoch 96/100
86/86 [==============================] - 0s 2ms/step - loss: 0.4233 - accuracy: 0.8062 - val_loss: 0.3972 - val_accuracy: 0.8163
Epoch 97/100
86/86 [==============================] - 0s 2ms/step - loss: 0.3902 - accuracy: 0.8330 - val_loss: 0.3930 - val_accuracy: 0.8367
Epoch 98/100
86/86 [==============================] - 0s 2ms/step - loss: 0.3934 - accuracy: 0.8171 - val_loss: 0.3847 - val_accuracy: 0.8316
Epoch 99/100
86/86 [==============================] - 0s 2ms/step - loss: 0.3835 - accuracy: 0.8313 - val_loss: 0.3875 - val_accuracy: 0.8537
Epoch 100/100
86/86 [==============================] - 0s 2ms/step - loss: 0.3926 - accuracy: 0.8268 - val_loss: 0.3850 - val_accuracy: 0.8095
```

Figure 6.11: MLP Running (2)

6.4 Comparison Between the Algorithms

Table 6.1: Accuracy of the Algorithms

Algorithm Used	**Accuracy**
KNN	92%
SVM (rbf)	77.80%
SVM (Linear)	69.60%
SVM (Sigmoid)	65.45%
MLP	89%

We can see from the above table how the accuracy of the different algorithms are varying. The interesting part about our thesis is that we have used MLP as one of our algorithms to segment the customers and this has not been done in any of researches we have looked into.
Even though SVM and KNN are very widely used algorithms for classification and segmentation they do have some demerits which we noticed while carrying out our experiments. KNN and SVM are both quite sensitive to noise in the data set. Noise is essentially the overlap of the target classes which might mislead the algorithm while it segments the data. KNN cannot function accurately when we pass very large data sets through it and suffers from over-fitting. SVMs also have the problem of over-fitting data, however the biggest draw back of SVM is the extremely high time complexity. In our experimentation, Gaussian Radial Basis Function kernel SVM took the longest time to compute compared to all the other algorithms.
MLP on the other hand out performed the others by not only giving a decent accuracy value of around 89% but it also executed with a lower time complexity. Also, we know that MLPs can handle a lot of data which will make it suitable to tackle any sort of data sets generated from any given range of time.
Thus, we can suggest based on our experimental outcomes that MLP would be the best algorithm to use here to segment the customers into their respective categories.

Chapter 7

Sales Prediction

Sales Forecasting

Sales forecasting is a method that is used in the prediction of sales that will be made by a company or business in the upcoming future. It is an important tool for companies as it helps in taking effective decisions in advance which will assist the company to enhance its performance both in the near and distant future. Predictions can be made from data that is already available like past transactions and sales.
In making such predictions, companies get a better understanding of their current state and can take sensible decisions regarding their recruitment procedure, setting objectives, incoming and outgoing cash etc. Thus, sales forecasting is a key factor in ensuring successful operations for a company.

Time Series Forecasting

A time series can be modeled using a stochastic procedure. A series of variables, which are generally random, are used to forecast a time setting. If the Stochastic process has a function Y(t) and has a time t, then time series forecasting will predict the value of Y(t+h) using available data of time t.
The time series forecasting model is considered accurate depending on how accurately it displays the performance of a company in the future with some concrete evidence and justification.

Sales prediction is based on time and it is a continuous process for which we need a stable pattern of sales history. We calculated the difference between each month to find the difference and plotted a graph.

Figure 7.1: Difference in sales every month

From the above figure we can see that there is a similar pattern in the graph from February to October. There is a sudden increase in the difference value in November and a sudden increase in December.
The monthly net revenue is fed into the model and values for the last 6 months are predicted.

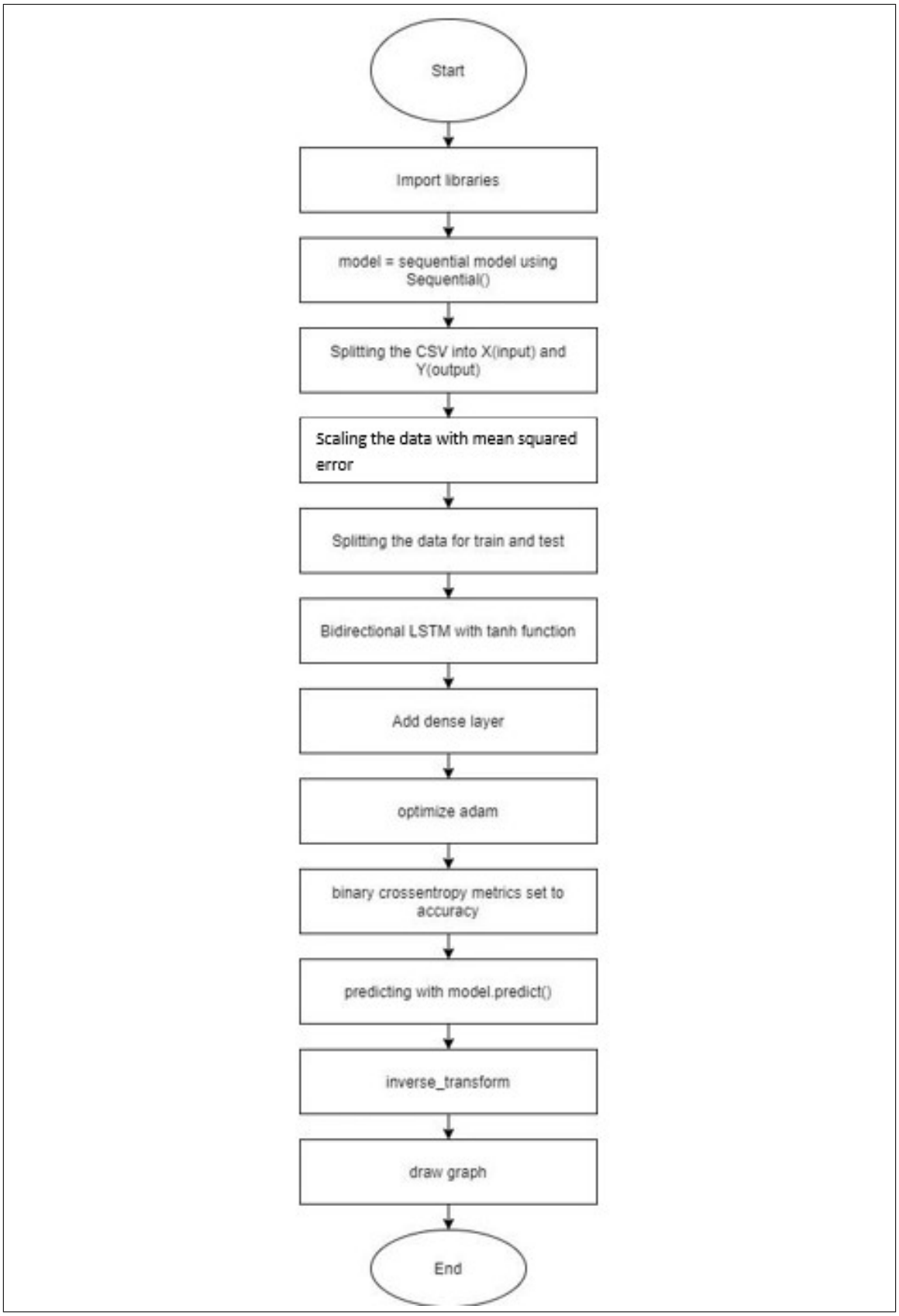


Figure 7.2: Work Flow Diagram for Predicting Sales

```
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
bidirectional (Bidirectional (1, 256)                  144384
_________________________________________________________________
dense (Dense)                (1, 1)                    257
=================================================================
Total params: 144,641
Trainable params: 144,641
Non-trainable params: 0
```

Figure 7.3: Model summary of the Bidirectional LSTM

```
Epoch 1/500
6/6 [==============================] - 3s 4ms/step - loss: 1.3317 - accuracy: 0.0000e+00
Epoch 2/500
6/6 [==============================] - 0s 3ms/step - loss: 1.2554 - accuracy: 0.0000e+00
Epoch 3/500
6/6 [==============================] - 0s 4ms/step - loss: 1.2026 - accuracy: 0.0762
Epoch 4/500
6/6 [==============================] - 0s 3ms/step - loss: 1.1532 - accuracy: 0.0762
Epoch 5/500
6/6 [==============================] - 0s 3ms/step - loss: 1.1047 - accuracy: 0.0762
Epoch 6/500
6/6 [==============================] - 0s 3ms/step - loss: 1.0563 - accuracy: 0.0762
Epoch 7/500
6/6 [==============================] - 0s 4ms/step - loss: 1.0078 - accuracy: 0.0762
Epoch 8/500
6/6 [==============================] - 0s 4ms/step - loss: 0.9595 - accuracy: 0.0762
Epoch 9/500
6/6 [==============================] - 0s 5ms/step - loss: 0.9120 - accuracy: 0.0762
Epoch 10/500
6/6 [==============================] - 0s 4ms/step - loss: 0.8662 - accuracy: 0.0762
Epoch 11/500
6/6 [==============================] - 0s 4ms/step - loss: 0.8230 - accuracy: 0.0762
Epoch 12/500
6/6 [==============================] - 0s 4ms/step - loss: 0.7832 - accuracy: 0.0762
Epoch 13/500
6/6 [==============================] - 0s 4ms/step - loss: 0.7473 - accuracy: 0.0762
Epoch 14/500
6/6 [==============================] - 0s 4ms/step - loss: 0.7155 - accuracy: 0.0762
Epoch 15/500
6/6 [==============================] - 0s 4ms/step - loss: 0.6879 - accuracy: 0.0762
```

Figure 7.4: Model Running, Epoch: 1-15

```
Epoch 281/500
6/6 [==============================] - 0s 5ms/step - loss: 0.5164 - accuracy: 0.0762
Epoch 282/500
6/6 [==============================] - 0s 5ms/step - loss: 0.5164 - accuracy: 0.0762
Epoch 283/500
6/6 [==============================] - 0s 4ms/step - loss: 0.5164 - accuracy: 0.0762
Epoch 284/500
6/6 [==============================] - 0s 4ms/step - loss: 0.5164 - accuracy: 0.0762
Epoch 285/500
6/6 [==============================] - 0s 4ms/step - loss: 0.5164 - accuracy: 0.0762
Epoch 286/500
6/6 [==============================] - 0s 4ms/step - loss: 0.5164 - accuracy: 0.0762
Epoch 287/500
6/6 [==============================] - 0s 4ms/step - loss: 0.5164 - accuracy: 0.0762
Epoch 288/500
6/6 [==============================] - 0s 5ms/step - loss: 0.5164 - accuracy: 0.0762
Epoch 289/500
6/6 [==============================] - 0s 4ms/step - loss: 0.5164 - accuracy: 0.0762
Epoch 290/500
6/6 [==============================] - 0s 5ms/step - loss: 0.5164 - accuracy: 0.0762
Epoch 291/500
6/6 [==============================] - 0s 5ms/step - loss: 0.5164 - accuracy: 0.0762
Epoch 292/500
6/6 [==============================] - 0s 5ms/step - loss: 0.5164 - accuracy: 0.0762
Epoch 293/500
6/6 [==============================] - 0s 4ms/step - loss: 0.5164 - accuracy: 0.0762
Epoch 294/500
6/6 [==============================] - 0s 5ms/step - loss: 0.5164 - accuracy: 0.0762
```

Figure 7.5: Model Running, Epoch: 281-294

Initially the accuracy is very low. However, as the model is trained, it learns from the previous value and the accuracy increases to 76%.

Figure 7.6: The prediction of the sales value

The model could identify a pattern and predict values quite close to the original ones apart from the December month, which showed a completely different pattern. Since sufficient data was not available after December, it could predict a decrease in value but not the sudden drastic drop. This could be due to various internal and external task environments of the business and other environmental factors that were not taken into account.

Chapter 8

Activation Functions, Loss Functions and Optimizer

Activation Functions

An activation function is used to generate the final output by the network model. It determines the percentage of accuracy and the effectiveness of the model. Activation functions play a major role in identifying the capability of a model to converge and the speed with which this convergence process is done. A series of mathematical equations are implemented as activation functions. The equations are added to every neuron within a network which later observes the sum computed by the neurons from their weights and inputs, and determines whether the neuron should be activated or not. Moreover, it normalizes the output of the neurons within a range of 0 and 1 or -1 and 1. The activation function acts as a gate between the input being processed by a neuron and then generating the output.
The activation functions that we have implemented are the ReLU and sigmoid function. Their attributes have been discussed below:

1. **ReLU:** ReLU is an extremely efficient activation function for performing computations and this allows convergence within the network to occur much more quickly. ReLU has a derivative function and it can back propagate when it is necessary.

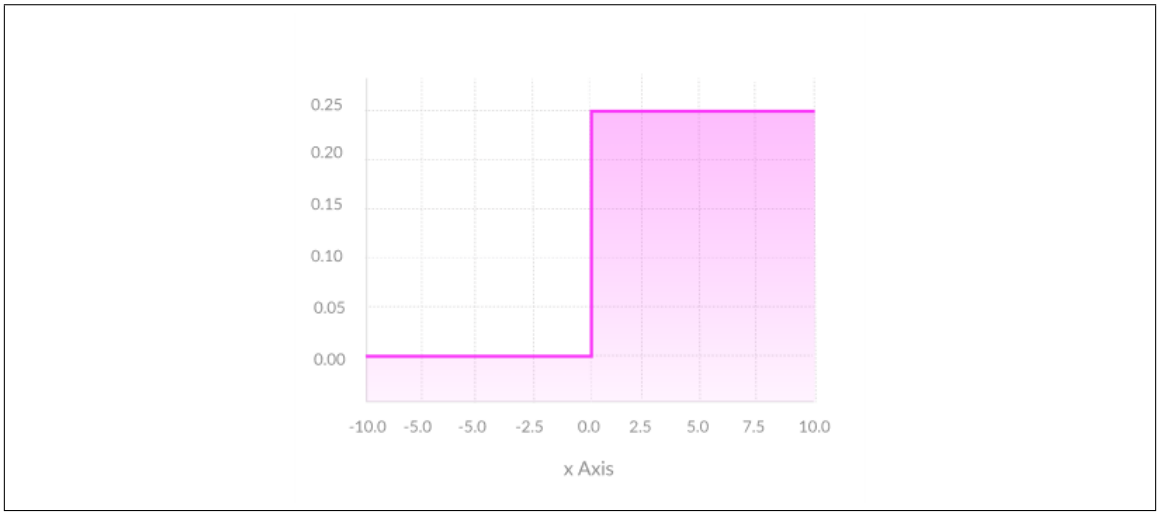


Figure 8.1: ReLu

2. **Sigmoid:** Sigmoid functions prevent any sudden changes of an output value. Moreover, the output values of each neuron are bounded within the range of 0 and 1 thus normalizing it. As the values of Y coordinates are brought close to the edge of the curve (near to 1 or 0) for values of X coordinates under -2 or above 2, proper predictions can be made [12].

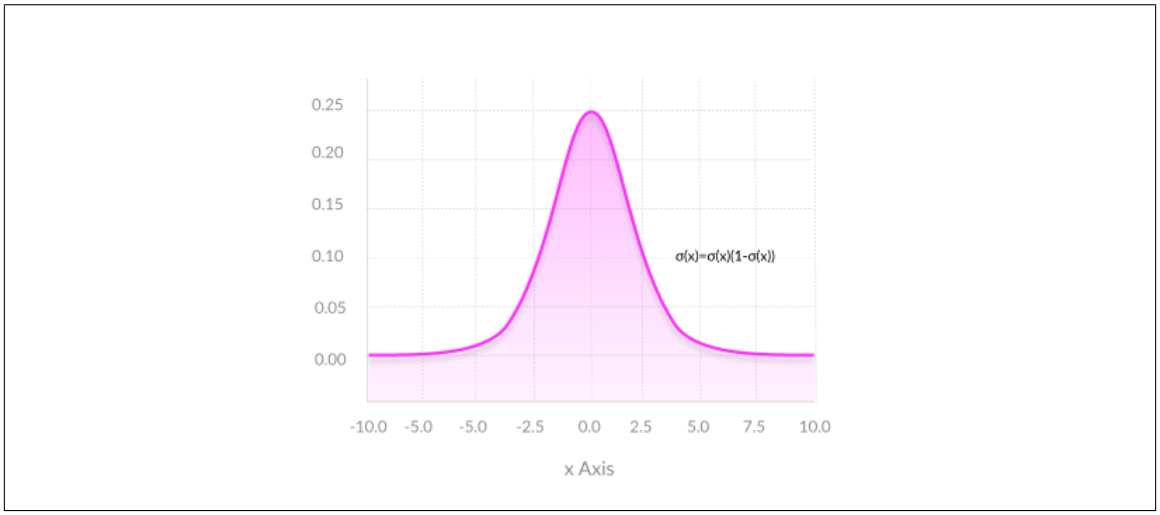


Figure 8.2: Sigmoid

3. **Tanh:**A Tanh activation function is categorized as a non-linear activation function and its main purpose is to represent the various data into a generalized form as well as present comparisons between the output. Tanh is usually used for classification purposes.

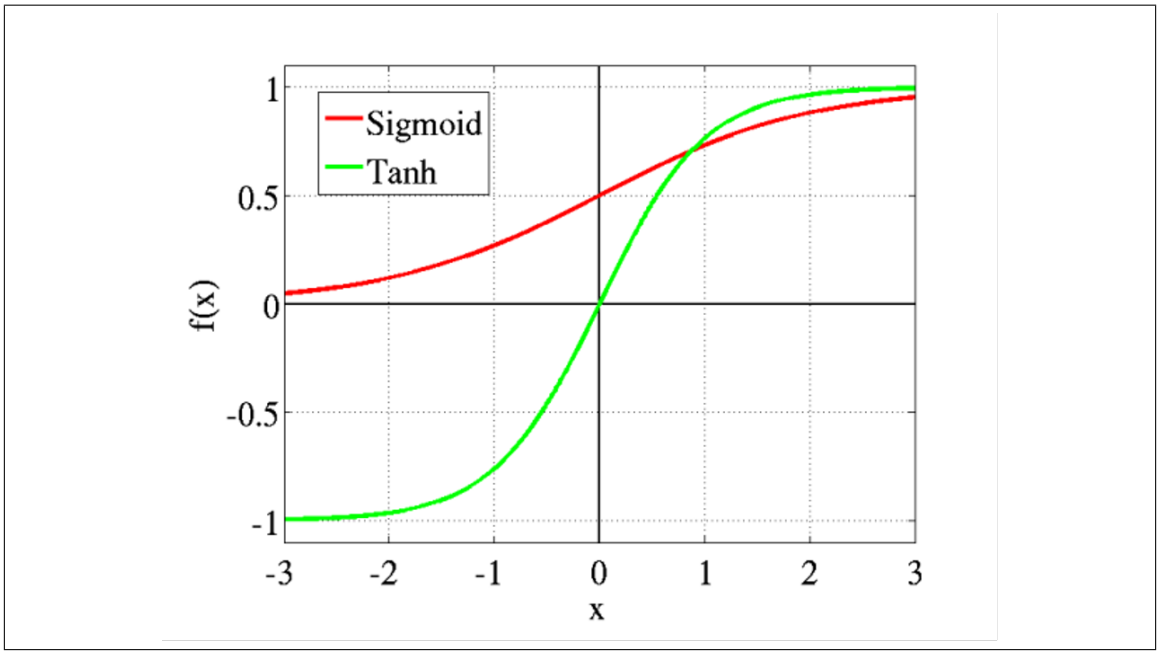


Figure 8.3: Tanh

The Tanh activation function is sigmoidal and its range is between -1 to 1 as shown in the diagram above. Negative inputs are generally mapped as strongly

negative due to the Tanh function. A benefit of the Tanh activation function is its differentiability. Moreover, this function is monotonic but its derivative is not.

Loss Functions

The deviation of the output data from the original result is represented using the loss function. If the deviation is high, the loss function generates a very large value which can be decreased using the optimization function. Loss functions vary in terms of functionality and implementation which means different scenarios require different loss functions. Different machine learning algorithms have different complexities and features which means each of them have separate criteria to select the appropriate loss function for them. The processes needed to calculate the derivatives mainly act as the primary factors that determine which loss function is needed for the algorithm [1]

Regression Loss Function

L1/Absolute Error Loss L2/Squared Error Loss Function: The square of the difference between the generated value and actual value is the Mean square error (MSE). It is basically the mean of the absolute difference between the generated output and actual result. L1 or LAD (more commonly known as Least Absolute Deviations) and L2 or LS (more commonly known as the least square errors) are the two loss functions which are widely used for error minimization where L1 function reduces the errors in calculations by summing up the differences between output and real value while L2 reduces the error by summing up the squared difference between generated value and actual value.

$$\text{L1LossFunction} = \sum_{i=1}^{n} |y_{\text{true}} - y_{\text{predicted}}| \tag{8.1}$$

$$\text{L2LossFunction} = \sum_{i=1}^{n} (y_{\text{true}} - y_{\text{predicted}})^2 \tag{8.2}$$

Mean Squared Error Loss

The Mean square error (MSE) represents the similarity index of the predicted values to the actual value. This means that if the MSE generates a low value, the forecast is close to the real value. Thus, regression models implement MSE to evaluate their models and if the value of MSE is low it means the fitness is proper. This helps to eliminate outlier predictions. The equation has a section where the value is squared to magnify the error which usually arises due to poor forecasts.

Binary Classification Loss

The classification of an object into two separate classes is done using binary classification. A rule applied to the input feature vector classifies the object.
Generally, an output p is obtained when the binary cross-entropy loss function is used for classification models. The mathematical expression of Binary Cross Entropy Loss is as follows:

$$\text{CrossEntropyLoss} = -\left(y_i \log\left(\hat{y}_i\right) + \left(1 - y_i\right) \log\left(1 - \hat{y}_i\right)\right) \tag{8.3}$$

Cross-entropy displays the comparisons between two probability distributions. While it somewhat different from the KL divergence, it aids in the calculation of the total entropy amongst the distributions.

Optmizer

The optimizer that we have used for our model is the ADAM optimizer. This is a broadly used optimizer algorithm that is an update or extension to the classic stochastic gradient descent. Stochastic gradient descent has a fixed learning rate which makes it quite slow to iterate over all the data. Adam makes use of two extensions of Stochastic gradient descent, Adaptive Array Algorithm (AdaGrad) and Root Mean Square Propagation (RMSProp). It has been observed that Adam performs at a lower cost (faster) than the algorithms mentioned above while training for the algorithms we have used in our research.

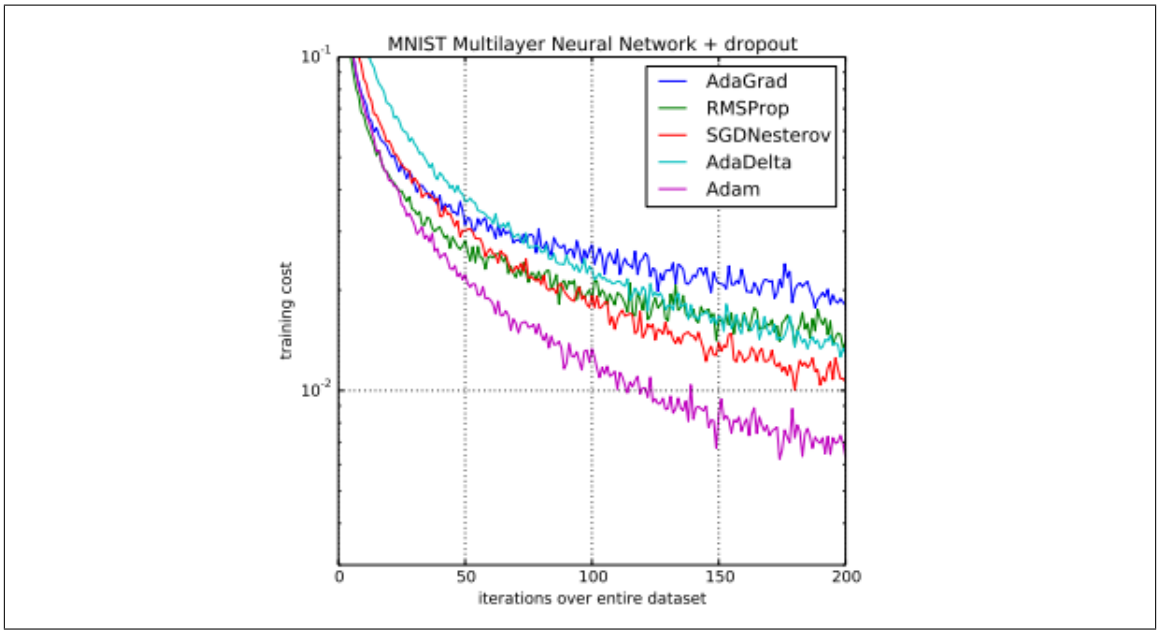


Figure 8.4: Optimizer

Chapter 9

Future Works

Our research for the past one year has allowed us to successfully achieve a lot of the objectives that we have set out to achieve. However, this is not the end of our research and we feel that there is still a lot of potential ground to cover with this research and as a team we would like to continue doing it. The prediction models described in the research paper have all generated outputs of various accuracies based on the data which has been gathered from an E-commerce business. It can be safe to presume that the following models will be able to generate reliable outputs if they were trained with data obtained from other business sectors as well. However, the unavailability of information on particular business sectors will be a major obstacle. To explain some of the key aspects that we would like to focus on in the future is the sales prediction. Up until now the sales that we can predict from our model is slightly limited in terms of how accurately and how much into the future we can make a correct prediction. We wish to improve upon our model with new data obtained after the global pandemic to make a more accurate assumption of how much the sales might be in the future. Moreover, net sales prediction was solely predicted based difference in sale value in month without taking any internal, external task environment into account and also no environmental factors were considered. We plan to research further to understand how it will influence the sales and take into account.

Profitability of a business can be determined by comparing the net revenue with total expense incurred. According to the profitability of the business, further decision about marketing campaigns and policies can be taken. And by doing that, we can propose a comprehensive marketing budget which can predict exactly how much sales will be generated.

Alongside that, these steps will help business get out of the problems they are facing due to the pandemic and once the pandemic is over not only will they have a successful re-connection with their old customers, but the marketing strategies they will implement will highly likely generate a solid profit. And finally, we will work on ways to suggest offers that can be deployed in order to make sure that the old customers reconnect and the business remains profitable. We hope that if we can achieve the above-mentioned tasks, we will be able to add an edge to our research.

Chapter 10

Conclusion

It is very difficult to build customer relationships in the E-commerce industry due to a lack of personal engagement and direct interaction with the customers. Although there were many types of research in many different fields regarding retaining existing customers, with our research, it is going to be a complete marketing solution for any business. Additionally, the impact of COVID-19 has forced majority of the businesses all over the world to operate in losses. Thus, effective marketing strategies to broaden the customer base of businesses have paramount importance in the market.

It is considered that a small proportion of customers are users of E-commerce platforms compared to the banking sectors or stock market; hence it is considered to be a slow growth sector. However, the sector is booming and is expected to grow in large proportion within a few years. According to Financial Express, the growth of the industry is 72% each month. As of 2018, Ecommerce business sites were 2,500 and E-commerce business pages were 150,000. During the pandemic, online business sales increased by 70–80% than usual. The net worth of E-commerce market was 1,648 million USD in 2019 which increased by 26% in 2020 and is expected to rise to 3077 million USD by 2023 [15].

This research can ensure customer engagement, build customer relationships, provide excellent customer service and most importantly use accessible and available marketing resources for personalized marketing. All these will translate more purchase intentions to actual purchases, increasing the revenue of the business.

In future, we plan to develop a mechanism to project marketing budget, personalized marketing strategy implementation based on customer profile and predict future sales keeping external environmental factors in mind.

Bibliography

[1] https://colah.github.io/posts/2015-08-Understanding-LSTMs/

[2] Brownlee, J. (2020). A Gentle Introduction to Long Short-Term Memory Networks by the Experts. Long Short-Term Memory Networks, 1. https://machinelearningmastery.com/gentle-introduction-long-short-term-memory-networks-experts/

[3] H. A. Abdou and J. Pointon, "Credit scoring, statistical techniques and evaluation criteria: a review of the literature," Intelligent Systems in Accounting, Finance and Management, vol. 18, no.2-3, pp.59–88, 2011.

[4] B. Baesens, R. Setiono, C. Mues, and J. Vanthienen, "Using neural network rule extraction and decision tables for credit risk evaluation," Management Science, vol.49, no.3, pp.312–329, 2003.

[5] E. I. Altman, G. Marco, and F. Varetto, "Corporate distress diagnosis: comparisons using linear discriminant analysis and neural networks," Journal of Banking Finance, vol. 18, no. 3, pp.505–529, 1994.

[6] A. F. Atiya, "Bankruptcy prediction for credit risk using neural networks: a survey and new results," IEEE Transactions on Neural Networks and Learning Systems, vol. 12, no. 4, pp. 929– 935, 2001.

[7] C. Carter and J. Catlett, "Assessing Credit Card Applications Using Machine Learning," IEEE Expert-Intelligent Systems and their Applications, vol.2, no.3, pp.71–79, 1987.

[8] K. J. Leonard, "Detecting credit card fraud using expert systems," Computers Industrial Engineering, vol.25, no.1-4, pp. 103–106, 1993.

[9] T. Bellotti and J. Crook, "Support vector machines for credit scoring and discovery of significant features," Expert Systems with Applications, vol.36, no.2, pp.3302–3308, 2009.

[10] B. S. Trinkle and A. A. Baldwin, “Research opportunities for neural networks: the case for credit,” Intelligent Systems in Accounting, Finance and Management, vol. 23, no. 3, pp. 240– 254,2016.

[11] B. S. Trinkle and A. A. Baldwin, “Research opportunities for neural networks: the case for credit,” Intelligent Systems in Accounting, Finance and Management, vol. 23, no. 3, pp. 240– 254,2016.

[12] Upadhyay, Yash. Introduction to FeedForward Neural Networks, Towards Data Science, 2019.

[13] Gupta, Tushar. Deep Learning: Feedforward Neural Network, Towards Data Science, 2017.

[14] Implementing SVM and Kernel SVM with Python's Scikit-Learn. (2021). Retrieved 12 January 2021, from https://stackabuse.com/implementing-svm-and-kernel-svm-with-pythons-scikit-learn/.

[15] Islam, M. (2021). E-commerce sales to reach $3b in 4 years. Retrieved 12 January 2021, from https://www.thedailystar.net/business/news/ecommerce-sales-reach-3b-4-years-1841428

[16] Wang, D., Nyberg, E. (2015). A Long Short-Term Memory Model for Answer Sentence Selection in Question Answering.

Cardinal Care

+8801878086002
cardinalcareltd@gmail.com
/cardinalcarebd

4th September, 2020

To whom it may concern,

A thesis group from BRAC University has approached Cardinal Care Ltd. to collect some sales and customer data for their research purpose. We were delighted to provide them with a set of data which they can use in their thesis research.

The details of the team and the research is as follows:

Title: Utilizing Machine Learning to project the financial outcomes of reconnecting with potential customers of the same industry

Research team:

1. Shoumik Hossain (ID: 17101322)
2. Quazi Fahmiduzzaman (ID: 17101307)
3. Nehrin Siddique Payel (ID: 17101508)
4. Mohammad Shahriar Hossain (ID: 17101239)
5. Nabil Hossain (ID: 16301134)

Supervised by: Moin Mostakim, Lecturer, Department of Computer Science and Engineering, BRAC University.

With Regards,

Aslam Beg
Chairman
Cardinal Care Limited

Aslam Beg Sayem
Chairman, Cardinal Care Ltd.
Contact No: +8801878086002

15, Bailey Road, Dhaka-1217, Beside Nawabi Bhoj Restaurant