

PREDICTING CUSTOMER STRESS IN TELECOM INDUSTRY USING DATA MINING

Submitted in partial fulfillment of the requirements for
the award of
Bachelor of engineering degree in Computer Science and Engineering
By

V.Anji Reddy (38110034)



**DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING
SCHOOL OF COMPUTING**

SATHYABAMA
INSTITUTE OF SCIENCE AND TECHNOLOGY
(DEEMED TO BE UNIVERSITY)
Accredited with Grade "A" by NAAC

**JEPPIAAR NAGAR, RAJIV GANDHI SALAI
CHENNAI - 600 119**

APRIL 2022



SATHYABAMA

**INSTITUTE OF SCIENCE AND TECHNOLOGY
(DEEMED TO BE UNIVERSITY)**

Accredited with Grade "A" by NAAC

JEPPIAAR NAGAR, RAJIV GANDHI SALAI, CHENNAI- 600 119

www.sathyabama.ac.in



DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

BONAFIDE CERTIFICATE

This is to certify that this Project Report is the bonafide work of **V.Anji Reddy (38110034)** who carried out the project entitled **"PREDICTING CUSTOMER STRESS IN TELECOM INDUSTRY USING DATA MINING"** under our supervision from November 2021 to April 2022.

Internal Guide

Dr. AROUL CANESSANE M.E., Ph.D.,

**HEAD OF THE DEPARTMENT
Dr. L. LAKSHMANAN, M.E., Ph.D.**

Submitted for Viva voce Examination held on _____

Internal Examiner

External Examiner

DECLARATION

I, **V.Anji Reddy (38110034)** hereby declare that the Project Report entitled “**PREDICTING CUSTOMER STRESS IN TELECOM INDUSTRY USING DATA MINING**” done by us under the guidance of **Dr. AROUL CANESSANE M.E., Ph.D.**, at Sathyabama Institute of Science and Technology is submitted in partial fulfillment of the requirements for the award of **Bachelor of Engineering** Degree in **Computer Science and Engineering**.

1.

DATE:

2.

PLACE:

SIGNATURE OF THE CANDIDATES

ACKNOWLEDGEMENT

I am pleased to acknowledge my sincere thanks to **Board of Management of SATHYABAMA** for their kind encouragement in doing this project and for completing it successfully. I am grateful to them.

I convey my thanks to **Dr.T.Sasikala M.E., Ph.D, Dean**, School of Computing, **Dr.S.Vigneshwari M.E., Ph.D. and Dr.L.Lakshmanan M.E., Ph.D. , Heads** of the Department of Computer Science and Engineering for providing me necessary support and details at the right time during the progressive reviews.

I would like to express my sincere and deep sense of gratitude to my Project Guide **Dr. AROUL CANESSANE M.E., Ph.D.**, for his valuable guidance, suggestions and constant encouragement paved way for the successful completion of my project work.

I wish to express my thanks to all **Teaching and Non-Teaching** staff members the **Department of Computer Science and Engineering** who were helpful in many ways for the completion of the project.

ABSTRACT

Retaining the most valuable customers is a major problem companies face in this information age. Especially, the field of telecommunication faces complex challenges due to a number of vibrant competitive service providers. Therefore, it has become very difficult for them to retain existing customers. Since the cost of acquiring new customers is much higher than the cost of retaining the existing customers, it is the time for the telecom industries to take necessary steps to retain the customers to stabilize their market value. CRM uses data mining (one of the elements of CRM) techniques to interact with customers. This study investigates the use of a technique, supervised learning, for the management and analysis of customer-related data warehouse and information. Data mining technologies extract hidden information and knowledge from large data stored in databases or data warehouses, which supports the corporate in decision making process. Several data mining techniques have been proposed in the literature for predicting the happy and stressed customer using heterogeneous customer records. Probably, the stressed customers are in the urge of moving out to competitive service providers. This project analysis the telecom customer data available in open data set and predict the customer stress by applying supervised machine-learning algorithms mainly using Deep Neural Network , K Nearest Neighbour , Support Vector Machine and Random Forest.

TABLE OF CONTENTS

CHAPTER No.	CHAPTER NAME	PAGE No.
	ABSTRACT	V
	LIST OF FIGURES	VIII
	LIST OF ABBREVIATIONS	IX
1	INTRODUCTION	01
	1.1 OBJECTIVE	01
	1.2 EXISTING SYSTEM	01
	1.2.1 SVM ALGORITHM	02
	1.2.2 K NEAREST NEIGHBOUR	02
	1.3 PROPOSED SYSTEM	03
2	LITERATURE SURVEY	05
	2.1 CUSTOMER CHURN	05
	2.2 DATA MINING TECHNIQUES	06
	2.2.1 ARTIFICIAL NEURAL NETWORK	06
	2.2.2 SELF ORGANIZING MAPS	07
	2.2.3 HYBRID APPROCHES	08
	2.3 RELATED WORK	09
3	METHODOLOGY	10
	3.1 HARDWARE REQUIREMENTS	10
	3.2 SOFTWARE REQUIREMENTS	10
	3.3 SYSTEM FEATURES	11
	3.3.1 PYTHON	11
	3.3.2 SPYDER	12

3.3.3 KERAS	12
3.3.4 ANACONDA	13
3.4 ALGORITHMS	13
3.4.1 DEEP NEURAL NETWORK	13
3.5 DATAFLOW DIAGRAM	14
3.5.1 DFD AT THE INITIAL LEVEL	14
3.5.2 DFD LEVEL 1 FOR ENTIRE PROJECT	15
3.5.3 DFD LEVEL1 FOR DATA PREPROCESSING	15
3.5.4 DFD LEVEL 2 FOR FEATURE EXTRACTION	16
3.5.5 DFD LEVEL 2 FOR SPLITTING DATA	17
3.5.6 DFD LEVEL 2 FOR CLASSIFYING DATA	18
3.5.7 DFD LEVEL 2 FOR PREDICTION	19
3.6 UNIFIED LANGUAGE	20
3.6.1 USE CASE DIAGRAM FOR LOAD FORECASTING	20
3.6.2SEQUENCE DIAGRAM FOR LOAD FORECASTING	21
3.7 ARCHITECTURE DIAGRAM	23
3.8 MODULES	23
3.9 MODULER DESCRIPTION	25
3.9.1 DATA EXTRACTION	25
3.9.2 PREPROCESSING	25
3.9.3 LABEL ENCODING	25
3.9.4 ONE HOT CODING	25
3.9.5 STANDARD SCALE	25
3.9.6 DATA VISUVALIZATION AND DESCRIPTION	26
3.9.7 DATA SPLITTING INTO TRAINING AND TEST SETS	26
3.9.7.1 OVER FITTING	26
3.9.7.2 UNDER FITTING	26
3.9.7.3 TRAIN /TEST SPLIT	27
3.9.8 DEEP NEURAL NETWORK	27

	3.9.9 PREDICTING ADMISSIONS BASED ON THE TEST	29
4	RESULT AND DISCUSSION	31
5	CONCLUSION AND FUTURE WORK	34
	5.1 CONCLUSION	34
	5.2 FUTURE WORK	34
	REFERENCES	35
	APPENDICES	36
	A.SOURCE CODE	36
	B.SCREENSHOTS	42

LIST OF FIGURES

FIGURE No.	FIGURE NAME	PAGE No.
2.1	MULTILAYER NEURAL NETWORK	07
2.2	A 4*4 SELF KOHONENS SELF ORGANIZING MAP	08
3.1	DATA FLOW DIAGRAM LEVEL 0 FOR ENTIRE PROJECT	14
3.2	DATA FLOW DIAGRAM LEVEL 1 FOR ENTIRE PROJECT	15
3.3	DATA FLOW DIAGRAM LEVEL1 FOR PREPROCESSING DATA	16
3.4	DATA FLOW DIAGRAM LEVEL 2 FOR FEATURE EXTRACTION	17
3.5	DATA FLOW DIAGRAM LEVEL 2 FOR SPLITTING DATA	18
3.6	DATA FLOW DIAGRAM LEVEL 2 FOR CLASSIFYING DATA	18
3.7	DATA FLOW DIAGRAM LEVEL 2 FOR ACCURACY WEATHER PREDICTION	19
3.8	USE CASE DIAGRAM OF LOAD FORECASTING	20
3.9	SEQUENCE DIAGRAM OF MODULE	21
3.10	ARCHITECTURE DIAGRAM	23
3.11	DEEP NEURAL NETWORK	27
4.1	SPLITTING OF DATA	31
4.2	EPOCH COUNT FOR PREDICTION	32
4.3	ACCURACY PREDICTION FOR SUPPORT VECTOR MACHINE	32
4.4	ACCURACY PREDICTION FOR DEEP NEURAL NETWORK	33

LIST OF ABBREVIATIONS

CRM	CUSTOMER RESOURCE MANAGEMENT
DNN	DEEP NEURAL NETWORK
DFD	DATA FLOW DIAGRAM
IDE	INTEGRATED DEVELOPMENT ENVIRONMENT
KNN	K NEAREST NEIGHBOUR
RDF	RANDOM FOREST
SGD	STOCHASTIC GRADIENT DESCENT
SVM	SUPER VECTOR METHOD
UML	UNIFIED MODELING LANGUAGE

CHAPTER 1

INTRODUCTION

1.1 OBJECTIVE

Our Aim Customer Stress-ML should meet the basic requirements, i.e., predict the churning of the customer, find the accuracy of the above algorithms *and* find the efficient one. Efficient Churn model predicts the Customer churning and helps in relieving the customer from the stress and help them in retaining in the respective telecom service.

1.2 EXISTING SYSTEM

Stressed customer, tend to cease doing business with a company in a given time period which has become a significant problem for many firms. These include publishing industry, investment services, insurance, electric utilities, health care providers, credit card providers, banking, Internet service providers, telephone service providers, online services, and cable services operators.

There are numerous predictive modeling techniques for predicting customer behavior and their satisfactory level. These vary in terms of statistical technique (e.g., neural nets versus logistic regression), variable selection method (e.g., theory versus stepwise selection), and number of variables included in the model.

We have highlighted below two algorithm and its predicting techniques.

- SVM algorithm
- KNN algorithm

1.2.1 SVM ALGORITHM

SVM algorithm developed by Vapnik, is based on statistical learning theory. In some classification cases, we try to find an optimal hyper-plane that separates two classes. When the two classes of points in the training set can be separated by a linear hyper-plane, it is natural to use the hyper-plane that separates the two groups of points in the training set by the largest margin. In order to find an optimal hyper-plane, we need to minimize the norm of the vector w , which defines the separating hyper-plane. This is equivalent to maximizing the margin between two classes.

LIMITATIONS

Customer stress prediction is a problem of classification between “happy” and “stressed” customer. But when the number of the negative examples is too small, the generalization performance of SVM classifier must be weak, and the error rates is proved unsatisfactory.

1.2.2 K NEAREST NEIGHBOUR

KNN algorithm is one of the simplest classification algorithm. Even with such simplicity, it can give highly competitive results. KNN algorithm can also be used for regression problems. KNN algorithm which when implemented helps to predict whether the customer will churn or not. But the accuracy level is not so high.

LIMITATIONS

The major disadvantage in KNN is determining the K value . Distance based classifier makes a very strong assumption on the shape of your data distribution, i.e. any two features are independent given the output class.

Another problem happens due to data scarcity. For any possible value of a feature, you need to estimate a likelihood value by a frequent approach. This can result in probabilities going towards 0 or 1, which in turn leads to numerical instabilities and worse results. In this case, you need to smooth in some way your probabilities or to impose some prior on your data.

1.3 PROPOSED SYSTEM

In this paper, we review the existing works on customer stress prediction in three different perspectives: Data sets, methods, and metrics. Firstly, we present the details about the availability of public data sets and what kinds of customer details are available for predicting customer stress, which leads to churn. Secondly, we compare and contrast the various predictive modeling methods that have been used in the literature for predicting the churners using different categories of customer records, and then quantitatively compare their performances. Finally, we summarize what kinds of performance metrics have been used to evaluate the existing churn prediction methods. Analyzing all these three perspectives is very crucial for developing a more efficient churn prediction system for telecom industries.

In a business environment, the term, customer churn simply refers to the customers leaving one business service to another, which is the process of customers switching from one service provider to another anonymously. This is because of customer stress and un-satisfaction with the business. From a machine learning perspective, customer stress (probable churn) prediction is a supervised (i.e. labeled) problem defined as follows: Given a predefined forecast horizon, the goal is to predict the future churners over that horizon, given the data associated with each subscriber in the network.

Churn Prediction is a phenomenon which is used to identify the possible churners in advance before they leave the network. This helps the CRM department to prevent subscribers who are likely to churn in future by taking the required retention

policies to attract the likely churners and to retain them. Thereby, the potential loss of the company could be avoided.

The input for this problem includes the data on past calls for each mobile subscriber, together with all personal and business information that is maintained by the service provider. In addition, for the training phase, labels are provided in the form of a list of churners. After the model is trained with highest accuracy, the model must be able to predict the list of churners from the real data set which does not include any churn label. In the perspective of knowledge discovery process, this problem is categorized as predictive mining or predictive modeling.

We are analyzing the Churn data with DNN – Deep Neural Network and Random forest . As both the algorithms are a deep investigator of data , the accuracy of the prediction can be increased.

CHAPTER 2

LITERATURE SURVEY

2.1 CUSTOMER CHURN

In the today's competitive market, many companies are coming to a full realization of the importance of the customer-oriented business strategy for sustaining their competitive edge and maintaining a stable profit level. That is, companies mainly rely on the income which comes from customers. However, to create and retain customers is difficult and costly in term of marketing. As new account setup, credit searches, and advertising and promotional expenses can add up to several times the cost of efforts that might enable the firm to retain a customer (Keaveney, 1995), it is becoming an industry-wide belief that the best core marketing strategy for the future is to retain existing customers and avoid customer churn (Kim, Park, & Jeong, 2004; Kim & Yoon, 2004).

Burez and Van den Poel (2007) indicate that there are two types of targeted approaches to managing customer churn: reactive and proactive. When a company adopts a reactive approach, it waits until customers ask the company to cancel their service relationship. In this situation, the company will offer the customer an incentive to stay. On the other hand, when a company adopts a proactive approach, it tries to identify customers who are likely to churn before they do so. The company then provides special programs or incentives for these customers to keep the customers from churning. Targeted proactive programs have potential advantages of having lower incentive costs. However, these systems may be very wasteful if churn predictions are inaccurate, because companies are wasting incentive money on customers who will not churn. Therefore, it is important to build a customer-churn prediction model as accurately as possible (Burez & Van den Poel, 2007; Van den Poel & Larivie're, 2004).

2.2 DATA MINING TECHNIQUES

In order to establish effective and accurate customer-churn prediction model, many data mining methods have been recently considered (e.g. Coussement & Van den Poel, 2008; Hung, Yen, & Wang, 2006). The two primary goals of data mining in practice tend to be description and prediction. Description focuses on finding human-interpreted patterns describing the data, and prediction involves using some variables or fields in the database to predict unknown or future values of other variables of interest (Fayyad, Piatetsky, & Smyth, 1996; Fayyad & Uthurusamy, 1996). The goals of description and prediction can be achieved using a variety of particular data mining methods include classification, clustering, regression, and so on (Berry & Linoff, 2003).

2.2.1 ARTIFICIAL NEURAL NETWORK

Classification is one of the commonly used data mining methods, and called as supervised learning techniques. It calculates the value of some variables, and classifies according to results. The algorithms of classification include decision trees, artificial neural networks, and so on (Han & Kamber, 2001; Tou & Gonzalez, 1974), in which artificial neural networks are the most widely considered technique in many business problems.

Artificial neural networks (ANN) attempt to simulate biological neural systems which learn by changing the strength of the synaptic connection between neurons upon repeated stimulation's by the same impulse (Li & Tan, 2006). Neural networks can be distinguished into single-layer perception and multilayer perception (MLP). The multilayer perception consists of multiple layers of simple, two taste, sigmoid processing nodes or neurons that interact by using weighted connections. In addition, the neural network contains one or more several intermediary layers between the input and output layers. Such intermediary layers are called hidden layers and nodes embedded in these layers are called hidden nodes shown in Fig. 1. Based on prior research results (e.g. Cybenko, 1998; Hung et al.,

2006; Zhan, Patuwo, & Hu, 1998), multilayer perception is a relatively accurate neural network model.

2.2.2 SELF-ORGANIZING MAPS

Clustering is an unsupervised learning method that partitions a set of patterns into groups (or clusters). Cluster analysis refers to the grouping of a set of data object into clusters. In particular, no predefined classes are assigned (Jain, Murty, & Flynn, 1999). Kohonen (1987) proposed and demonstrated a new form of a neural network architecture called self-organizing map (SOM), which has proved extremely useful when the input data are high dimensionality and complexity. SOM is used to discover associations in a data set and cluster data according to the similarity of data (i.e., similar expression patterns) where the model creators cannot predict the nature of the classification, or they consider that there may be more than one way to categorize the characteristics of a data set (Zhang, Edwards, & Harding, 2007). Fig. 2 shows an example of a 4 x 4 SOM.

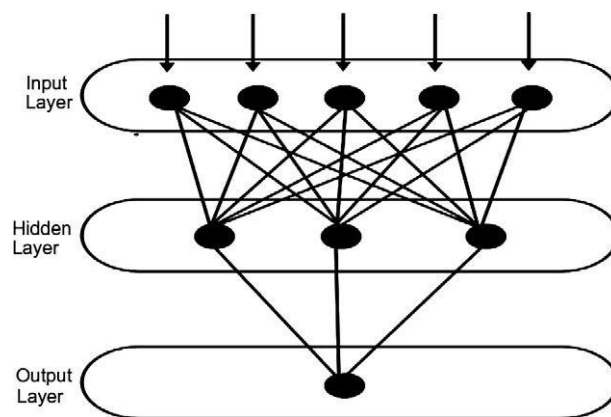


FIG 2.1 MULTILAYER NEURAL NETWORK

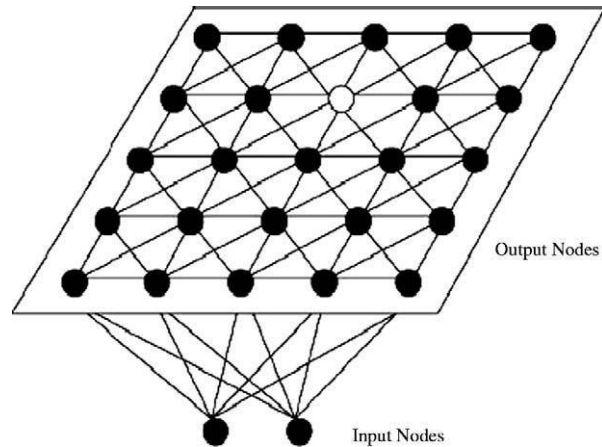


FIG 2.2 A 4×4 KOHONEN'S SELF ORGANIZING MAP

2.2.3 HYBRID APPROACHES

In general, hybrid models are based on combining two or more data mining techniques. For example, the clustering and classification techniques can be serially combined. That is, clustering can be used as a preprocessing stage to identify pattern classes for subsequent supervised prediction or classification (Jain et al., 1999). Therefore, the clustering result can be used for either pre classification of unlabeled collections or identifying major populations of a given data set.

Then, the clustering result becomes the training set to train and create a prediction model. After the model is created, it is able to classify or predict new (unknown) instances. In other words, the first component of the hybrid models can simply perform the task of outlier detection and the second for prediction. Therefore, there are two possible combination methods for hybrid models (Lenard et al., 1998) as shown in Figs. 3 and 4, respectively.

As the churn prediction task is based on supervised classification techniques, the second component or prediction model of the above mentioned two hybrid approaches are all based on classification techniques. Therefore, two hybrid models considered in this paper are the classification + classification and cluster + classification hybrid models (c.f. Section 3.2).

2.3 RELATED WORK

Building an effective customer churn prediction model has become an important topic for business and academics in recent years. In order to understand how related work constructs their prediction models, this paper reviews some of current related studies shown in Table 1. Coussement and Van den Poel (2008) applies support vector machines in a newspaper subscription context in order to construct a churn model. The predictive performance of the support vector machine model is benchmarked to logistic regression and random forests.

Burez and Van den Poel (2007) built a prediction model for European pay-TV company by using Markov chains and a random forest model benchmarked to a basic logistic model. Other studies (e.g. Kim & Yoon, 2004) only use a binomial logit regression to construct the prediction model. According to prior literature about customer churn, much related work focus on using only one data mining method such as classification or clustering to mine the customer retention data. Few studies (Hung et al., 2006) apply more than one technology which are cluster analysis and classification.

In this paper, we consider to develop hybrid data mining models by combining clustering and classification data mining techniques which may outperform one single methods. In particular, in the data mining process, the first clustering or classification technique is used during the preprocessing stage for data reduction, which aims at filtering out unrepresentative training data. Besides, we also contemplate using cross-validation which can improve the performance and reflect the 'true' performance when validating the created models for unseen data that the validation set stays untouched (Coussement & Van den Poel, 2008). As ANN and SOM are the two mostly used neural network techniques, there are two approaches to develop the hybrid prediction model for customer churn, which are SOM + ANN and ANN + ANN. The baseline model for comparisons is based on the single ANN model.

CHAPTER 3

METHODOLOGY

The system requirement is a main part in the analyzing phase of the project. The analyzer of the project has to properly analyze the hardware and the software requirements, otherwise in future the project designer will face more trouble with the hardware and software required. Below specified are the project hardware and software requirements.

3.1 HARDWARE REQUIREMENTS

The hardware requirements may serve as the basis for a contract for the implementation of the system and should therefore be a complete and consistent specification of the whole system. They are used by the software engineers as the starting point for the system design. It shows what the system does and not how it should be implemented.

- SYSTEM : PENTIUM DUAL CORE
- HARD DISK : 120 GB
- MONITOR : 15" LCD
- INPUT DEVICE : KEYBOARD, MOUSE
- RAM : 8 GB

3.2 SOFTWARE REQUIREMENTS

The software requirements document is the software specification of the system. It should include both a definition and a specification of a requirements. It is a set of what the system should do rather than how it should do it. The software requirements provide a basis for creating the software requirements specification. It is useful in estimating cost, planning team activities, performing tasks and tracking the team's progress throughout the development activity.

- OPERATING SYSTEM : WINDOWS 10
- PROGRAMMING LANGUAGE : PYTHON

- TOOLS : ANACONDA,SPYDER,PYTHON
- DATABASE : KERAS LIBRARY

3.3 SOFTWARE FEATURES

3.3.1 PYTHON

Python features a dynamic type system and automatic memory management. It supports multiple programming paradigms, including object oriented, imperative, functional and procedural, and has a large and comprehensive standard library. Python interpreters are available for many operating systems. CPython, the reference implementation of Python, is open source software and has a community-based development model, as do nearly all of Python's other implementations. Python and C Python are managed by the non-profit Python Software Foundation Python is a multi-paradigm programming language. Object-oriented programming and structured programming are fully supported, and many of its features support functional programming and aspect-oriented programming .Many other paradigms are supported via extensions, including design by contract and logic programming.

Python uses dynamic typing, and a combination of reference counting and a cycle-detecting garbage collector for memory management. It also features dynamic name resolution, which binds method and variable names during program execution.Python's design offers some support for functional programming in the Lisp tradition. It has `filter()`, `map()`, and `reduce()` functions; list comprehensions, dictionaries, and sets; and generator expressions. The standard library has two modules that implement functional tools borrowed from Haskell and Standard ML.

3.3.2 SPYDER

Spyder is an open source cross-platform integrated development environment (IDE) for scientific programming in the Python language. Spyder integrates with a number of prominent packages in the scientific Python stack, including NumPy, SciPy, Matplotlib, pandas, IPython, SymPy and Cython, as well as other open source software. It is released under the MIT license. Initially created and developed by Pierre Raybaut in 2009, since 2012 Spyder has been maintained and continuously improved by a team of scientific Python developers and the community.

Spyder is extensible with first- and third-party plugins, includes support for interactive tools for data inspection and embeds Python-specific code quality assurance and introspection instruments, such as Pyflakes, Pylint and Rope. It is available cross-platform through Anaconda, on Windows with WinPython and Python (x,y), on macOS through MacPorts, and on major Linux distributions such as Arch Linux, Debian, Fedora, Gentoo Linux, openSUSE and Ubuntu. Spyder uses for its GUI, and is designed to use either of the PyQt or PySide Python bindings. QtPy, a thin abstraction layer developed by the Spyder project and later adopted by multiple other packages, provides the flexibility to use either back end.

3.3.3 KERAS

Keras is an Open Source Neural Network library written in Python that runs on top of Theano or Tensor flow. It is designed to be modular, fast and easy to use. It was developed by François Chollet, a Google engineer. Keras doesn't handle low-level computation. Instead, it uses another library to do it, called the "Back end. So Keras is high-level API wrapper for the low-level API, capable of running on top of TensorFlow, CNTK, or Theano.

Keras High-Level API handles the way we make models, defining layers, or set up multiple input-output models. In this level, Keras also compiles our model with loss and optimizer functions, training process with fit function. Keras doesn't handle Low-

Level API such as making the computational graph, making tensors or other variables because it has been handled by the "Back end" engine.

3.3.4 ANACONDA

Anaconda is a free and open-source distribution of the Python programming languages for scientific computing that aims to simplify package management and deployment. Package versions are managed by the package management system conda. The Anaconda distribution is used by over 12 million users and includes more than 1400 popular data-science packages suitable for Windows, Linux, and MacOS.

Anaconda is a Python-based data processing and scientific computing platform. It has built in many very useful third-party libraries. Installing Anaconda is equivalent to automatically installing Python and some commonly used libraries such as Numpy, Pandas, Scipy, and Matplotlib, so it makes the installation so much easier than regular Python installation. It is painful and you need to consider compatibility, thus it is highly recommended to directly install Anaconda.

3.4 ALGORITHMS

The machine learning algorithms that has to be applied to the training data to build the models are

3.4.1 DEEP NEURAL NETWORK

Logistic regression is suitable for predicting a binary dependent variable, such as positive/negative; deceased/alive; or in this study, admit/not admit. The technique uses a logistic link function to enable the calculation of the odds of an outcome occurring. The second algorithm that was used was a random forest, specifically recursive partitioning from the RPART package . The RPART package is an implementation based on the model presented by Breiman and colleagues . This algorithm splits the data at each node based on the variable that best separates the data until either an optimal model is identified or a minimum number of observations exists in the final (terminal) nodes . The resulting tree can then be pruned to prevent

over fitting and to obtain the most accurate model for prediction . The third algorithm was a SVM which creates two vectors and tries to classify the data based on it . The last algorithm is KNN where the N defines the number of nearest neighbours and the value changes as and when the n changes .

Thus all the algorithms can increase the prediction and accuracy of the Churn prediction in Telecom Indusustry. The term “Design” is defined as the technical kernel of the software engineering process and is applied regardless of the development paradigm and area of application. Design is the first step in the development phase for any engineered product or system. The designer’s goal is to produce a model or representation of an entity that will later be built.

3.5 DATA FLOW DIAGRAM

DFDs are used to Specify Functions of the Information System and how data flow from function to function. It’s a collection of function that manipulates data. On a DFD, data items flow from an external data source or an internal data store to an internal data store or an external data sink, via an internal process. A DFD provides no information about the timing of processes, or about whether processes will operate in sequence or in parallel.

3.5.1 DATA FLOW DIAGRAM AT THE INITIAL LEVEL

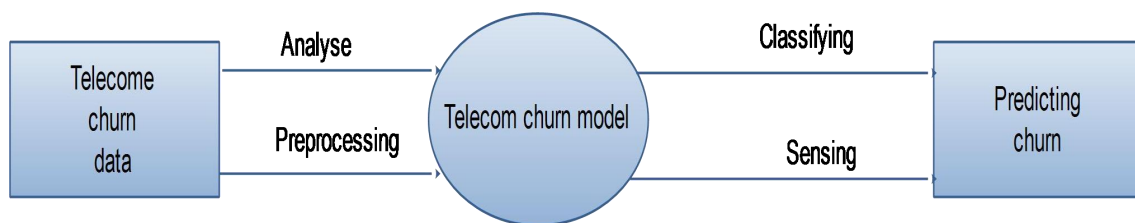


FIG 3.1 DATA FLOW DIAGRAM LEVEL 0 FOR THE ENTIRE PROJECT

Fig 3.5.1 explains the outer overall functionality of the proposed system application this indicates that have collected the Telecom Churn data set from various source and analysis and processing of data have done by fetching the data from many database .

3.5.2 DATA FLOW DIAGRAM LEVEL 1 FOR ENTIRE PROJECT

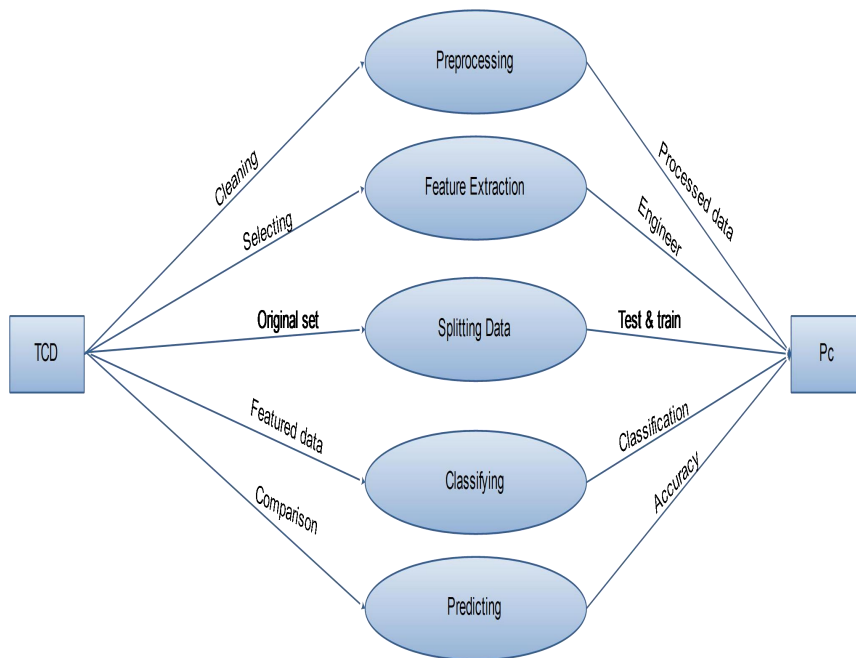


FIG 3.2 DATA FLOW DIAGRAM LEVEL 1 FOR THE ENTIRE PROJECT

Fig 3.2 above is an overall representation of each module and their functionality in this project. This Level one Data Flow Diagram shows how the various modules help in the providing the output which predicts the accurate Telecom Person information data with high frequency data set.

3.5.3 DATA FLOW DIAGRAM LEVEL 1 FOR DATA PREPROCESSING

Fig 3.3 below clearly explains the data is cleaned before loading to the Neural Network Classifier. Import the Label Encoder class from the sclera library, fit and transform the first column of the data, and then replace the existing text data with the new encoded data. One hot encoding is a process by which categorical variables are

converted into a form that could be provided to ML algorithms to do a better job in prediction. This work fitting and transforming Standard scalar method on train data.

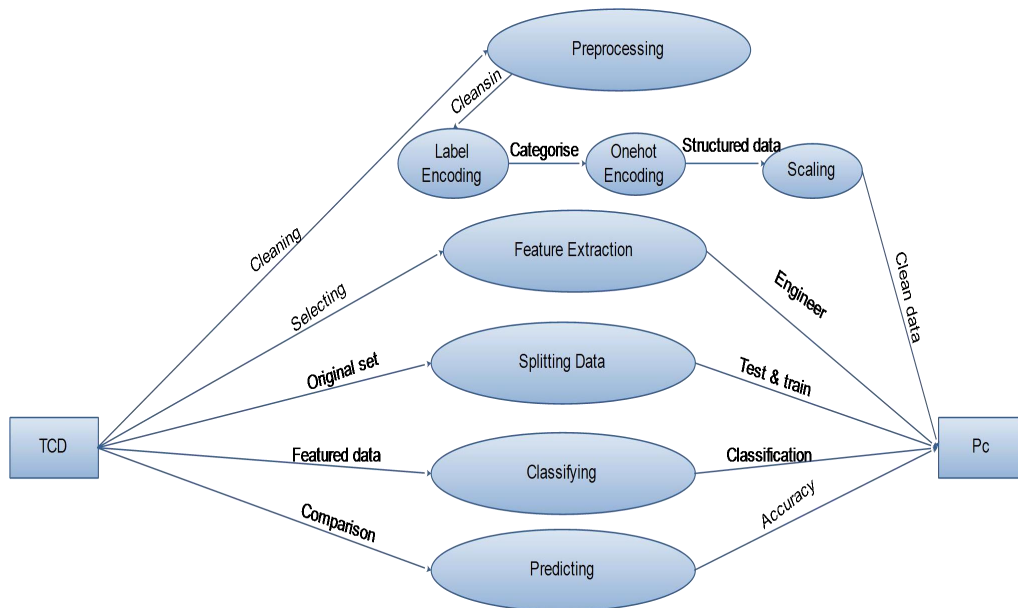


FIG 3.3 DATA FLOW DIAGRAM LEVEL 1 FOR PREPROCESSING DATA

3.5.4 DATA FLOW DIAGRAM LEVEL 2 FOR FEATURE EXTRACTION

Fig 3.4 below explains the Feature extraction by Layering of data. For feature extraction, first select the data and process data for original feature. Then reduce the feature by transform refining data. This can be done by selecting the data by CSV file and process the data for original feature and it was transformed and refining data to reduced feature, finally it predicts the accuracy of churning.

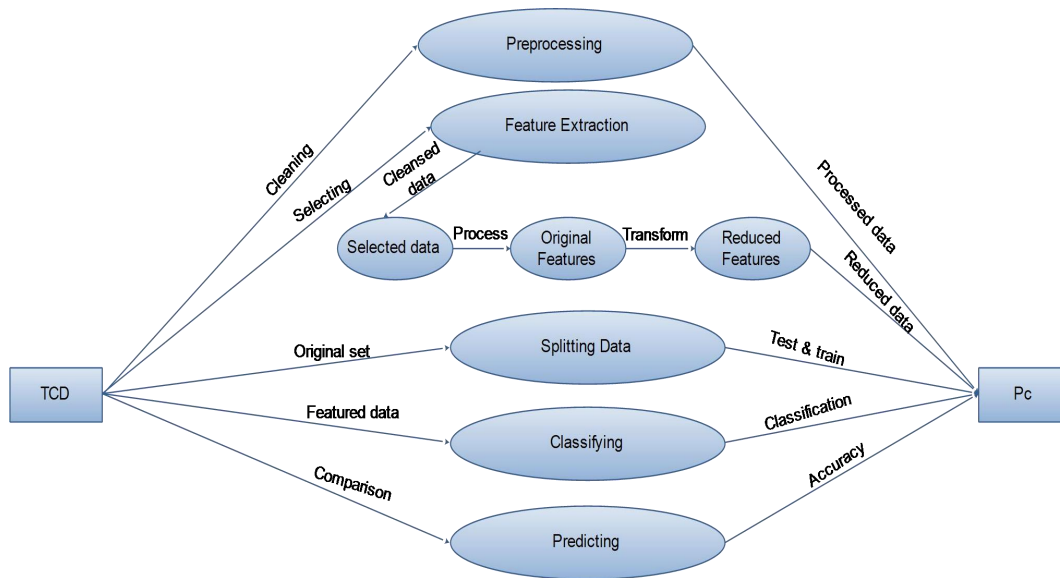


FIG 3.4 DATA FLOW DIAGRAM LEVEL 2 FOR FEATURE EXTRACTION

3.5.5 DATA FLOW DIAGRAM LEVEL 2 FOR SPLITTING DATA

Fig 3.5 below explains the Splitting data by two ways are Training data set and Testing data set. In statistics and machine learning we usually split our data into two subsets: **training data** and **testing data** (and sometimes to three: train, validate and test), and fit our model on the train data, in order to make predictions on the test data. When this work do that, one of two thing might happen: we over fit our model or we under fit our model. It don't want any of these things to happen, because they affect the predictability of our model— Work might be using a model that has lower accuracy and/or is un-generalize .

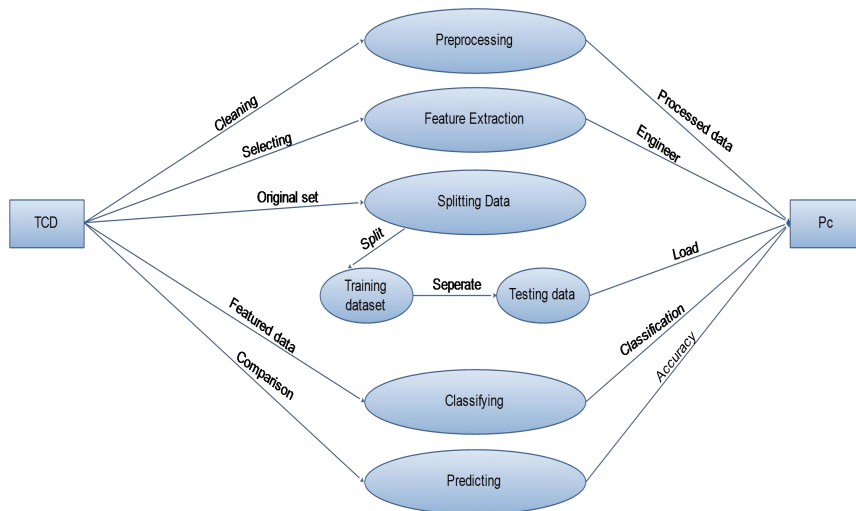


FIG 3.5 DATA FLOW DIAGRAM LEVEL 2 FOR SPLITTING DATA

3.5.6 DATA FLOW DIAGRAM LEVEL 2 FOR CLASSIFYING

DATA

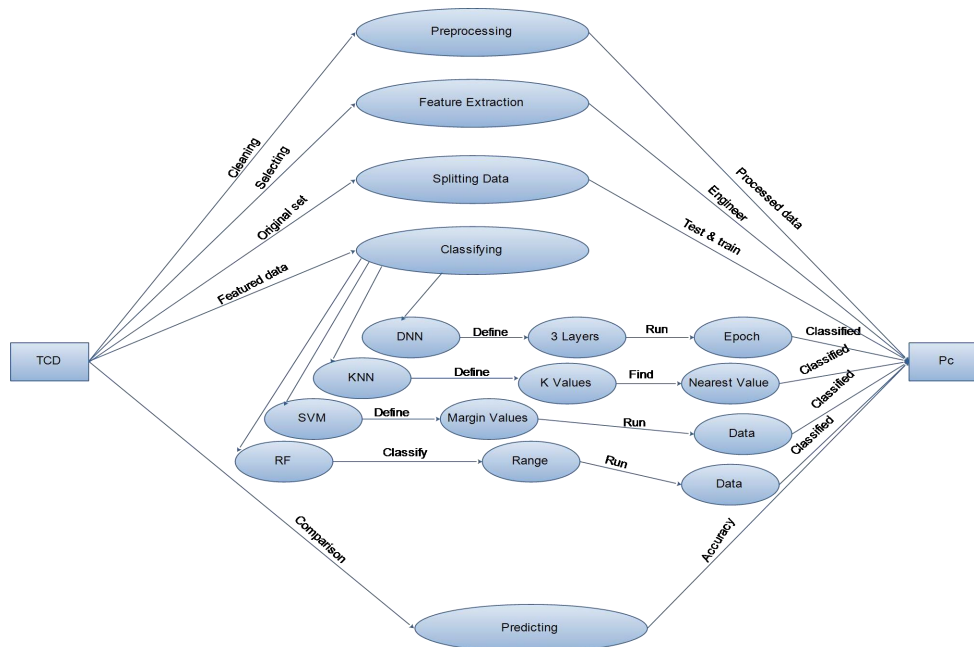


FIG 3.6 DATA FLOW DIAGRAM LEVEL 2 FOR CLASSIFYING DATA

Fig 3.5.6 above explains that Classification of data through DNN (Deep Neural Network) Algorithm Neural networks use randomness by design to ensure they

effectively learn the function being approximated for the problem. Randomness is used because this class of machine learning algorithm performs better with it than without. The most common form of randomness used in neural networks is the random initialization of the network weights. A deep neural network is a neural network with a certain level of complexity, a neural network with more than two layers. Deep neural networks use sophisticated mathematical modeling to process data in complex ways. A neural network, in general, is a technology built to simulate the activity of the human brain – specifically, pattern recognition and the passage of input through various layers of simulated neural connections. Deep neural networks as networks that have an input layer, an output layer and at least one hidden layer in between. Each layer performs specific types of sorting and ordering in a process that some refer to as “feature hierarchy.”

3.5.7 DATA FLOW DIAGRAM LEVEL 2 FOR PREDICTION

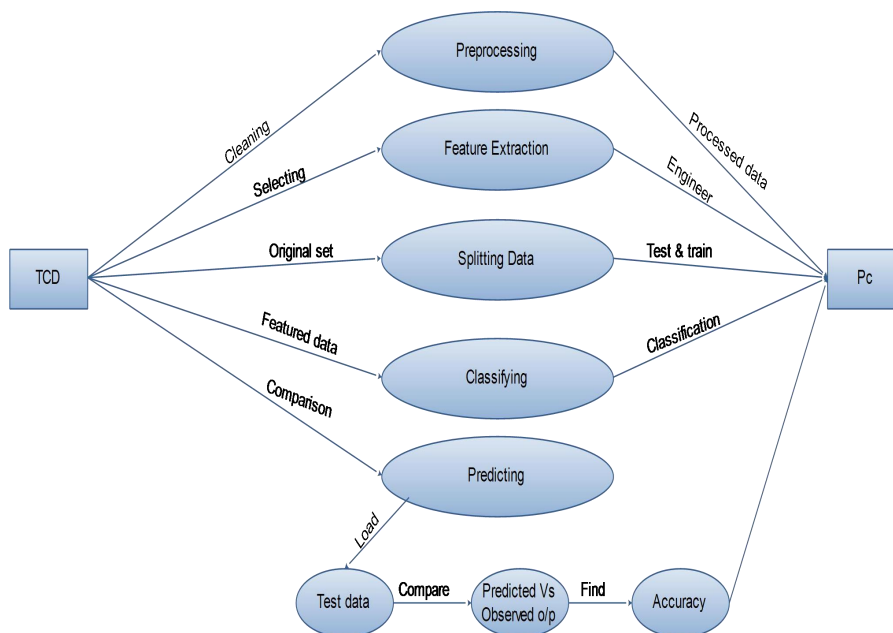


FIG 3.7 DATA FLOW DIAGRAM LEVEL 2 FOR ACCURACY WEATHER PREDICTION

Fig 3.7 above explains how the accuracy emergency department prediction occurs using the classifier, list with predicted values and plot the rainfall graph. **Predicting** the test set result. The prediction result will give you probability of customers churning the telecom industry. This will convert that probability into binary 0 and 1. 1 for Churning the Telecom Industry and 0 for not churning . This is the final step where this work evaluating our model performance. It predicts the Confusion matrix to check the accuracy of model.

3.6 UNIFIED MODELING LANGUAGE

The Unified Modeling Language (UML) is used to specify, visualize, modify, construct and document the artifacts of an object-oriented software intensive system under development. UML combines techniques from data modeling (entity relationship diagrams), business modeling (work flows), object modeling, and component modeling. It can be used with all processes, throughout the software development life cycle, and across different implementation technologies.

3.6.1 USE CASE DIAGRAM FOR LOAD FORECASTING

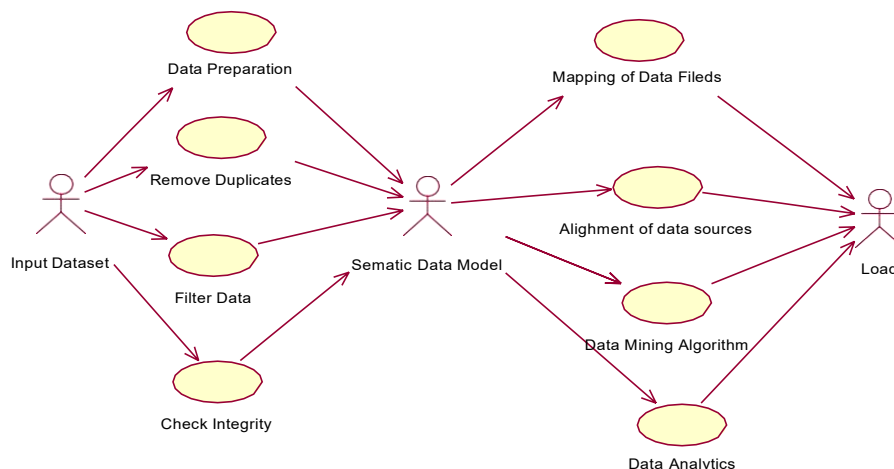


FIG 3.8 USE CASE DIAGRAM OF LOAD FORECASTING

Above the figure shows that it provides a global view of the actors involved in a system and the actions that the system performs, which in turn provide an observable result that is of value to the actors. Actors are an abstraction of the entity that interacts with the software. Here the system is partitioned simpler functions. The use case diagram of this software is as follows.

3.6.2 SEQUENCE DIAGRAM FOR LOAD FORECASTING

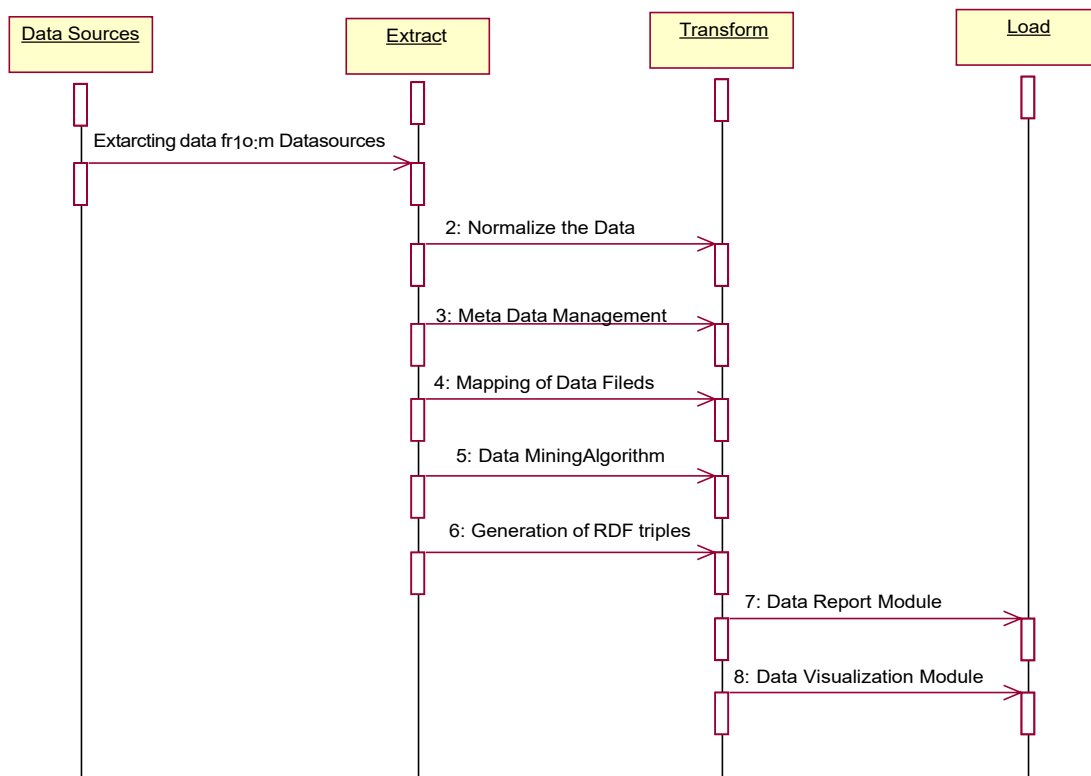


FIG 3.9 SEQUENCE DIAGRAM OF MODULE 1

In the below sequence diagram, the entities used are data sources, extract, transform, load. The data source extract data from database which normalize the data and meta data management that performs mapping of data fields. Data Mining Algorithm were used to generate RDF triple sub net data set which transforms the data rate. This data results in each module to visualize the load data set in effective

and distributed way. First argument is **Optimizer**, this is the algorithm we use to find optimal set of weights. This algorithm is Stochastic Gradient descent(SGD). Among several types of SGD algorithm the one which this will use is '**Adam**'. This go in deeper detail of SGD, we will find that SGD depends on loss thus our second parameter is **loss**. Since out dependent variable is binary, this will have to use logarithmic loss function called '**binary_crossentropy**', if our dependent variable has more than 2 categories in output then use '**categorical_crossentropy**'

IDE Setup

The Machine Learning Concepts can be well implemented through PYTHON . We have a numerous Python Tools but for the DNN implementation Anaconda can meet our needs. Spyder in Anaconda is chosen as the IDE setup . The latest Anaconda Spyder has Python 3.6 version . Python 3.6 is unstable to hold the KERAS backend . Apart from the inabilitNO and TENSORFLOW . Tensorflow can be implemented over Python 3.5 only. A cloning environment is built to build over Python 3.6 , KERAS requires two additional library packages THEA using the Anaconda Prompt . The Python 3.5 is cloned using the command .

conda create -n py35 python=3.5 anaconda

After activating the cloned environment the Spyder is installed using the command . So the Python3.5 has its own Spyder version , which does not interfere with the base version of Anaconda .

conda install spyder

The required libraries can be set up using the below commands

conda install theano

conda install tensorflow

conda install keras

3.7 ARCHITECTURE DIAGRAM

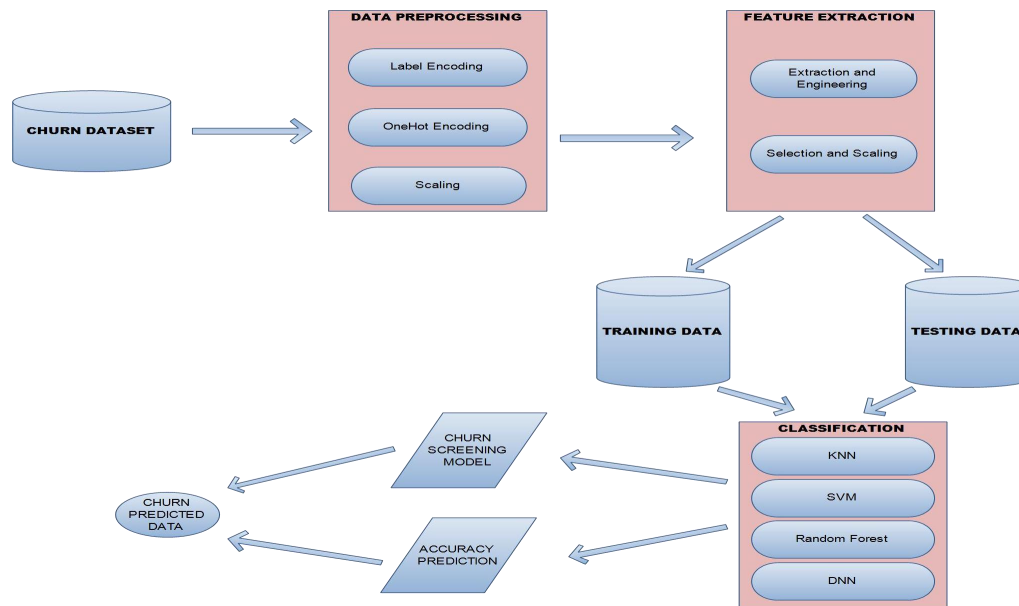


FIG 3.10 ARCHITECTURE DIAGRAM

3.8 MODULES

The modules for this study involved seven data mining tasks.

1. Data extraction
2. Data cleansing and feature engineering;
3. Data visualization and descriptive statistics;
4. Data splitting into training (80%) and test sets (20%);
5. Model tuning using the training set and 10fold cross validation repeated 5 times
6. Predicting admissions based on the test data set and
7. The evaluation of model performance based on predictions made on the test data

These steps help to ensure the models are optimal and prevent against over fitting. A range of variables were considered in the model building, with the final variables decided upon based on previous studies, significance in the models, and the impact of inclusion on the performance of the model. The final models consisted of variables describing whether the customer feels to retain in the same telecom industry or he finds it as stress and opts out of it .

Feature engineering was also carried out on the incoming , outgoing calls , revenue , number of income and outgoing calls etc. The dependent variable in all models was customer id from the telecom data set. Most of the variables included in the model are mandatory on the telecom churn system, and recorded using of drop down menus. This led to a relatively clean data set for analysis, with list wise deletion of cases with missing data.

3.9 MODULE DESCRIPTION

3.9.1 DATA EXTRACTION

We have collected the Telecom Industry Churn data set .We are using this data info to predict the churning of customer from that particular telecom industry for the current year and make appropriate preparations.

Importing the Pandas gives massive functionality to work on data . Along with Pandas importing all the required libraries are even imported.

3.9.2 PREPROCESSING

The data has to be Cleaned before loading into the Neural Classifier .

3.9.3 LABEL ENCODING

Before running a model, we need to make this data ready for the model.To convert any kind of categorical text data into model-understandable numerical data, we use the Label Encoder class, all we have to do is to label encode the first column. We import the Label Encoder class from the sklearn library, fit and transform the first column of the data, and then replace the existing text data with the new encoded data.

3.9.4 ONE HOT CODING

One hot encoding is a process by which categorical variables are converted into a form that could be provided to ML algorithms to do a better job in prediction. This is why we use one hot encoder to perform “binarization” of the category and include it as a feature to train the model.

3.9.5 STANDARD SCALE

Standard Scale is available in ScikitLearn we are fitting and transforming Standard Scale method on train data. We have to standardize our scaling so we will use the same fitted method to transform/scale test data.

3.9.6 DATA VISUALISATION AND DESCRIPTIVE STATISTICS

The extracted features are usually plotted against the output to check its relation .

3.9.7 DATA SPLITTING INTO TRAINING (80%) AND TEST SETS (20%)

In statistics and machine learning we usually split our data into two subsets: **training data** and **testing data** (and sometimes to three: train, validate and test), and fit our model on the train data, in order to make predictions on the test data. When we do that, one of two things might happen: we over fit our model or we under fit our model. We don't want any of these things to happen, because they affect the predictability of our model—we might be using a model that has lower accuracy and/or is ungeneralized (meaning you can't generalize your predictions on other data).

3.9.7.1 Over fitting

Over fitting means that model we trained has trained “too well” and is now, well, fit too closely to the training data set. This usually happens when the model is too complex (i.e. too many features/variables compared to the number of observations). This model will be very accurate on the training data but will probably be very not accurate on untrained or new data. It is because this model is not generalized (or not AS generalized), meaning we can generalize the results and can't make any inferences on other data, which is, ultimately, what we are trying to do. Basically, when this happens, the model learns or describes the “noise” in the training data instead of the actual relationships between variables in the data. This noise, obviously, isn't part in of any new data set, and cannot be applied to it.

3.9.7.2 Under fitting

In contrast to over fitting, when a model is under fitted, it means that the model does not fit the training data and therefore misses the trends in the data. It also means the model cannot be generalized to new data. As you probably guessed this is usually the result of a very simple model (not enough predictors/independent variables). It could also happen when, for example, we fit a linear model to data that

is not linear. It almost goes without saying that this model will have poor predictive ability (on training data and can't be generalized to other data).

3.9.7.3 TRAIN / TEST SPLIT

The data we use is usually split into training data and test data. The training set contains a known output and the model learns on this data in order to be generalized to other data later on. We have the test data set (or subset) in order to test our model's prediction on this subset.

3.9.8 DEEP NEURAL NETWORK

A deep neural network is a neural network with a certain level of complexity, a neural network with more than two layers. Deep neural networks use sophisticated mathematical modeling to process data in complex ways. A neural network, in general, is a technology built to simulate the activity of the human brain – specifically, pattern recognition and the passage of input through various layers of simulated neural connections.

Many experts define deep neural networks as networks that have an input layer, an output layer and at least one hidden layer in between. Each layer performs specific types of sorting and ordering in a process that some refer to as “feature hierarchy.” One of the key uses of these sophisticated neural networks is dealing with unlabeled or unstructured data. The phrase “deep learning” is also used to describe these deep neural networks, as deep learning represents a specific form of machine learning where technologies using aspects of artificial intelligence seek to classify and order information in ways that go beyond simple input/output protocols.

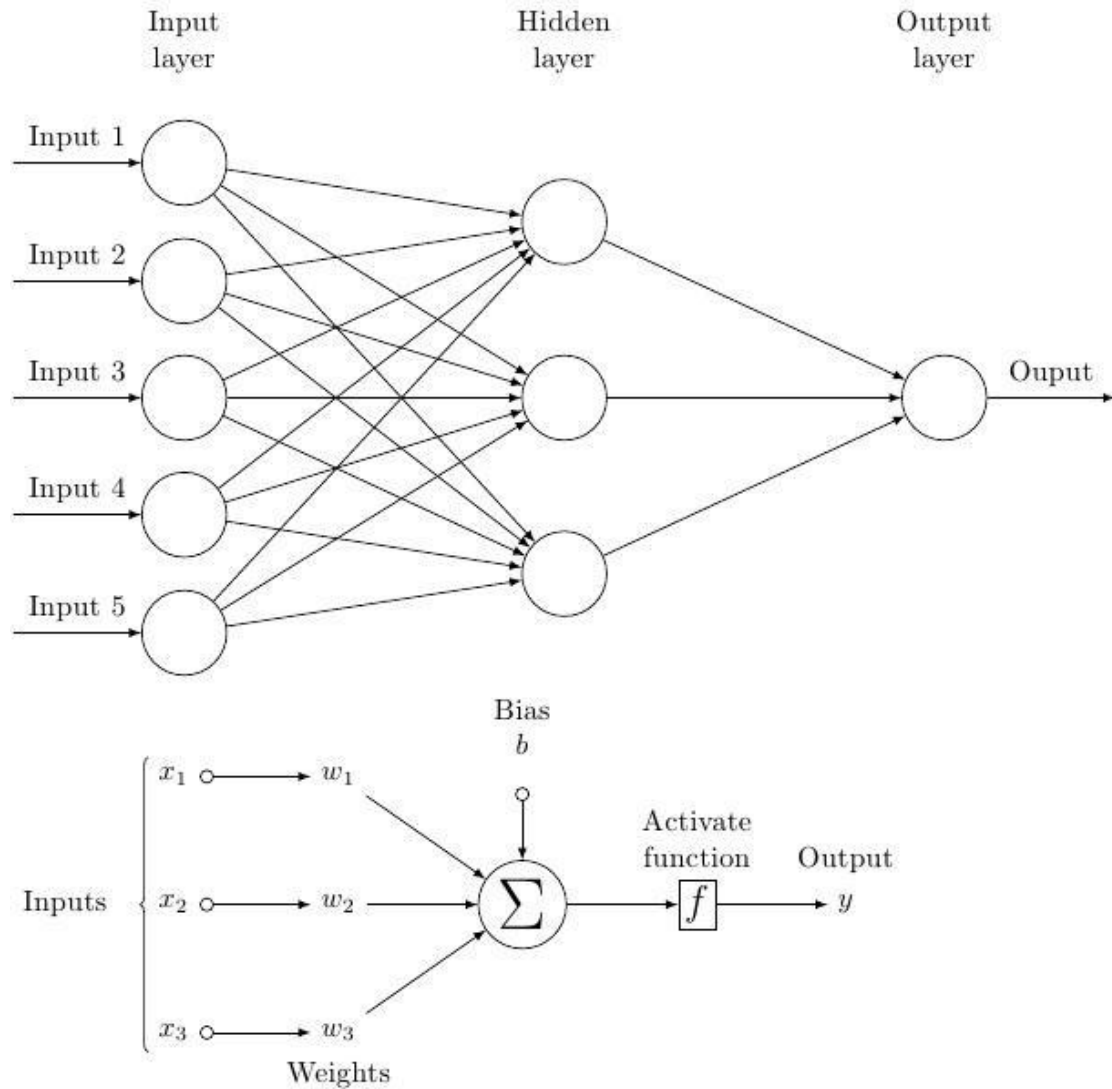


FIG 3.9.8 DEEP NEURAL NETWORK

```
# Adding the input layer and the first hidden layer
classifier.add(Dense(output_dim = 6, init = 'uniform', activation = 'relu',
input_dim = 11))
# Adding the second hidden layer
classifier.add(Dense(output_dim = 6, init = 'uniform', activation = 'relu'))
# Adding the output layer
classifier.add(Dense(output_dim = 1, init = 'uniform', activation = 'sigmoid'))
```

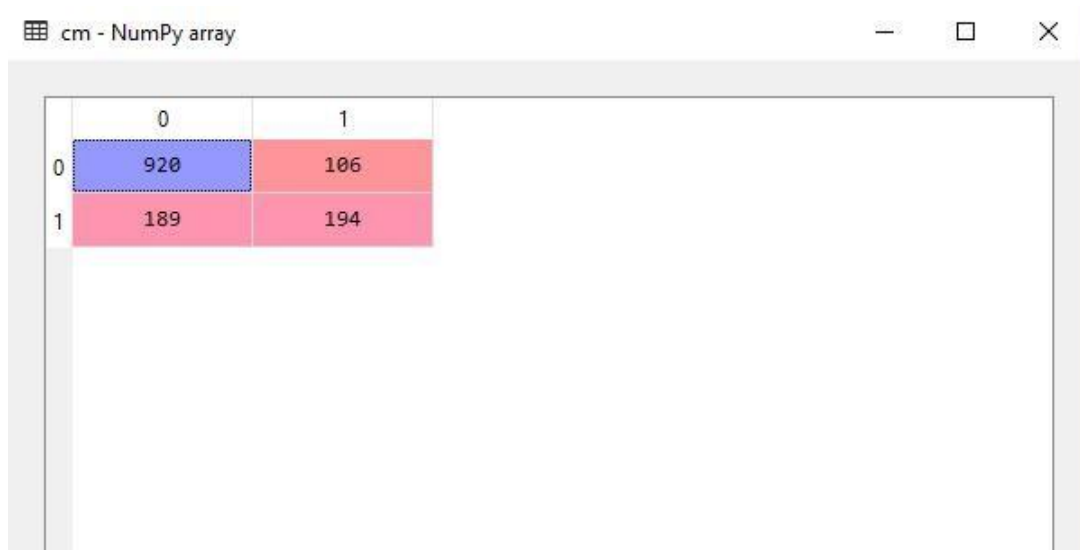
First argument is **Optimizer**, this is the algorithm we use to find optimal set of weights. This algorithm is Stochastic Gradient descent(SGD). Among several types of SGD algorithm the one which we will use is '**Adam**'. If we go in deeper detail of SGD, we will find that SGD depends on loss thus our second parameter is **loss**. Since our dependent variable is binary, we will have to use logarithmic loss function called '**binary_crossentropy**', if our dependent variable has more than 2 categories in output then use '**categorical_crossentropy**'. We want to improve performance of our neural network based on **accuracy** so add **metrics** as accuracy. We can train our model on training data. We use fit method to fit our model **Batch size** is used to specify the number of observation after which we want to update weight. **Epoch** is nothing but the total number of iterations. Choosing the value of batch size and epoch is trial and error there is no specific rule for that.

3.9.9 PREDICTING ADMISSIONS BASED ON THE TEST DATA SET

Predicting the test set result. The prediction result will give you probability of the customers churning out of the Telecom industry for the given period. We will convert that probability into binary 0 and 1.

3.9.10 THE EVALUATION OF MODEL PERFORMANCE BASED ON PREDICTIONS MADE ON THE TEST DATA

This is the final step where we are evaluating our model performance. We build Confusion matrix to check the accuracy of model.



	0	1
0	920	106
1	189	194

Accuracy= $920+194/1409=0.790631 = 80\%$

CHAPTER 4

RESULTS AND DISCUSSION

Name	Type	Size	Value
X	float64	(19289, 18)	[[1.0000e+00 0.0000e+00 0.0000e+00 ... 0.0000e+00 0.0000e+00 0.0000e+0 ...
X_test	float64	(3858, 18)	[[0.89053431 -0.25121133 -0.60962648 ... -0.53825969 -0... -0 ...
X_train	float64	(15431, 18)	[[-1.12292136 3.98071223 -0.60962648 ... -0.1270457 -0... -0 ...
dataset	DataFrame	(19289, 15)	Column names: Site ID, Condition, Plot number, Year, Month , Day , Tim ...
tmp	DataFrame	(19289, 15)	Column names: Site ID, Condition, Plot number, Year, Month , Day , Tim ...
y	float64	(19289,)	[0. 0. 0. ... nan nan nan]
y_test	float64	(3858,)	[1. 1. 1. ... 1. 1. nan]
y_train	float64	(15431,)	[0. 0. 1. ... 1. 1. 0.]

Variable explorer
File explorer
Help

FIG 4.1 SPLITTING OF DATA

```

+00
Epoch 36/100
15431/15431 [=====] - 2s 120us/step - loss: nan - acc: 0.0000e
+00
Epoch 37/100
15431/15431 [=====] - 2s 120us/step - loss: nan - acc: 0.0000e
+00
Epoch 38/100
15431/15431 [=====] - 2s 120us/step - loss: nan - acc: 0.0000e
+00
Epoch 39/100
15431/15431 [=====] - 2s 119us/step - loss: nan - acc: 0.0000e
+00
Epoch 40/100
15431/15431 [=====] - 2s 110us/step - loss: nan - acc: 0.0000e

```

FIG 4.2 EPOCH COUNT FOR PREDICTION

```

...:
[[ 3 757]
 [ 1 3097]]
      precision    recall  f1-score   support

     0       0.75       0.00       0.01       760
     1       0.80       1.00       0.89      3098

 avg / total       0.79       0.80       0.72      3858

In [69]: print('Accuracy' + str(accuracy_score(y_test, y_pred)))
Accuracy0.8035251425609123

```

FIG 4.3 ACCURACY FOR SVM ALGORITHM

```

...: print('Accuracy' + str(accuracy_score(y_test, y_pred)))
...: print(classification_report (y_test, y_pred))
...:
...:
[[ 584 176]
 [ 203 2895]]
Accuracy0.9017625712804562

```

	precision	recall	f1-score	support
0	0.74	0.77	0.76	760
1	0.94	0.93	0.94	3098
avg / total	0.90	0.90	0.90	3858

FIG 4.4 ACCURACY FOR DNN ALGORITHM

CHAPTER 5

CONCLUSION AND FUTURE WORK

5.1 CONCLUSION

Thus the DNN algorithm works far better than the other algorithm in predicting the churning of Telecom Industry.

5.2 FUTURE WORK

The future upgrade of this work is to use the propelled systems for grouping and advancement to additionally improve the exhibition.

REFERENCES

- [1] A. Jalva, "The job of enormous information in improving force framework activity and security," in Proc. IEEE IREP Symp. Rethymnon Bulk Power Syst. Dyn. Control-IX Optim. Security Control Emerging Power Grid, 2013, pp. 1–9.
- [2] D. X. Niu, H. F. Shi, and D. D. Wu, "Momentary burden determining utilizing Bayesian neural systems learned by half and half Monte Carlo algorithm," *Appl. Delicate Comput.*, vol. 12, no. 6, pp. 1822–1827, 2012.
- [3] F. Sulla, M. Koivisto, and J. Seppanen, "Factual investigation and guaging of damping in the nordic force framework," *IEEE Trans. Force Syst.*, vol. 30, no. 1, pp. 306–315, Jan. 2015.
- [4] K. B. Melody et al., "Half breed load determining strategy with investigation of temperature sensitivities," *IEEE Trans. Force Syst.*, vol. 21, no. 2, pp. 869–876, May 2006.
- [5] M. Kezunovic, L. Xie, and S. GriA. Azadeh, M. Saberi, S. F. Ghaderi, and V. Ebrahimipour, "Improved estimation of power request work by coordination of fluffy framework and information mining approach," *Energy Convers. Oversee.*, vol. 49, no. 8, pp. 2165–2177, 2008.
- [6] M. Rejc and M. Panto, "Momentary transmission-misfortune conjecture for the Slovenian transmission influence framework dependent on a fluffy rationale choice methodology," *IEEE Trans. Force Syst.*, vol. 26, no. 3, pp. 1511–15.
- [7] W. C. Hong, "Confused molecule swarm improvement calculation in a help vector relapse electric burden estimating model," *Energy Convers. Oversee.*, vol. 50, no. 1, pp. 105–117, 2009.
- [8] Y. Wang, Q. Xia, and C. Kang, "Optional anticipating dependent on deviation examination for momentary burden determining," *IEEE Trans. Force Syst.*, vol. 26, no. 2, pp. 500–507, May 2011.

APPENDICES

A.SOURCE CODE

```
import numpy as np
import pandas as pd
import matplotlib
import theano
import tensorflow
import keras

#dataset=pd.read_csv('C:\\Users\\BALAJIKRISHNARAJ\\Downloads\\SoilData
set.csv', encoding = "ISO-8859-1", header=None)
dataset=pd.read_csv('D:\\Nithya_vedha\\FinalProjects\\5.Telecom
Churn\\Telco-Customer-Churn.csv')
X = dataset.iloc[:, 0:20].values
y = dataset.iloc[:, 20].values

from sklearn.preprocessing import LabelEncoder, OneHotEncoder
labelencoder_X_0 = LabelEncoder()
X[:, 0] = labelencoder_X_0.fit_transform(X[:, 0])
labelencoder_X_2 = LabelEncoder()
X[:, 1] = labelencoder_X_2.fit_transform(X[:, 1])
labelencoder_X_2 = LabelEncoder()
X[:, 2] = labelencoder_X_2.fit_transform(X[:, 2])
labelencoder_X_3 = LabelEncoder()
X[:, 3] = labelencoder_X_3.fit_transform(X[:, 3])
labelencoder_X_4 = LabelEncoder()
X[:, 4] = labelencoder_X_4.fit_transform(X[:, 4])
labelencoder_X_5 = LabelEncoder()
X[:, 5] = labelencoder_X_5.fit_transform(X[:, 5])
```

```
labelencoder_X_6 = LabelEncoder()
X[:, 6] = labelencoder_X_6.fit_transform(X[:, 6])

labelencoder_X_7 = LabelEncoder()
X[:, 7] = labelencoder_X_7.fit_transform(X[:, 7])
labelencoder_X_8 = LabelEncoder()
X[:, 8] = labelencoder_X_8.fit_transform(X[:, 8])
labelencoder_X_9 = LabelEncoder()
X[:, 9] = labelencoder_X_9.fit_transform(X[:, 9])
labelencoder_X_10 = LabelEncoder()
X[:, 10] = labelencoder_X_10.fit_transform(X[:, 10])
labelencoder_X_11 = LabelEncoder()
X[:, 11] = labelencoder_X_11.fit_transform(X[:, 11])
labelencoder_X_12 = LabelEncoder()
X[:, 12] = labelencoder_X_12.fit_transform(X[:, 12])
labelencoder_X_13 = LabelEncoder()
X[:, 13] = labelencoder_X_13.fit_transform(X[:, 13])
labelencoder_X_14 = LabelEncoder()
X[:, 14] = labelencoder_X_4.fit_transform(X[:, 14])
labelencoder_X_15 = LabelEncoder()
X[:, 15] = labelencoder_X_15.fit_transform(X[:, 15])
labelencoder_X_16 = LabelEncoder()
X[:, 16] = labelencoder_X_16.fit_transform(X[:, 16])
labelencoder_X_17 = LabelEncoder()
X[:, 17] = labelencoder_X_17.fit_transform(X[:, 17])
labelencoder_X_18 = LabelEncoder()
X[:, 18] = labelencoder_X_18.fit_transform(X[:, 18])
labelencoder_X_19 = LabelEncoder()
X[:, 19] = labelencoder_X_9.fit_transform(X[:, 19])
onehotencoder = OneHotEncoder(categorical_features = [1])
```

```
X = onehotencoder.fit_transform(X).toarray()
X = X[:, 1:]
from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size = 0.2)
from sklearn.preprocessing import StandardScaler
sc = StandardScaler()
X_train = sc.fit_transform(X_train)
X_test = sc.transform(X_test)
import keras
from keras.models import Sequential
from keras.layers import Dense
classifier = Sequential()
classifier.add(Dense(output_dim = 6, init = 'uniform', activation = 'relu',
input_dim = 20))
classifier.add(Dense(output_dim = 6, init = 'uniform', activation = 'relu'))
classifier.add(Dense(output_dim = 1, init = 'uniform', activation = 'sigmoid'))
classifier.compile(optimizer = 'adam', loss = 'binary_crossentropy', metrics =
['accuracy'])
classifier.fit(X_train, y_train, batch_size = 10, nb_epoch = 100)
y_pred = classifier.predict(X_test)
y_pred = (y_pred > 0.5)
from sklearn.metrics import confusion_matrix
cm = confusion_matrix(y_test, y_pred)
from sklearn.discriminant_analysis import LinearDiscriminantAnalysis as LDA
lda = LDA(n_components=1)
X_train = lda.fit_transform(X_train, y_train)
X_test = lda.transform(X_test)
from sklearn.ensemble import RandomForestClassifier
classifier = RandomForestClassifier(max_depth=2, random_state=0)
classifier.fit(X_train, y_train)
y_pred = classifier.predict(X_test)
```

```

from sklearn.metrics import confusion_matrix
from sklearn.metrics import accuracy_score

cm = confusion_matrix(y_test, y_pred)
print(cm)
print('Accuracy' + str(accuracy_score(y_test, y_pred)))

""" Knn """

from sklearn import metrics
from sklearn.neighbors import KNeighborsClassifier
from sklearn.linear_model import LogisticRegression
from sklearn.model_selection import train_test_split

""" Chi Feature Extraction Start"""

from sklearn.feature_selection import SelectKBest
from sklearn.feature_selection import chi2

X = X.astype(int)
X_train= X_train.astype(int)
X_test= X_test.astype(int)
chi2_features = SelectKBest(chi2)
X_kbest_features_train = chi2_features.fit_transform(X_train, y_train)
print('Original feature number:', X.shape[1])
print('Reduced feature number:', X_kbest_features_train.shape[1])
X_kbest_features_test = chi2_features.fit_transform(X_test, y_test)
print('Original feature number:', X.shape[1])
print('Reduced feature number:', X_kbest_features_test.shape[1])

""" End """

k_range = list(range(1,26))
scores = []
for k in k_range:
    knn = KNeighborsClassifier(n_neighbors=k)
    knn.fit(X_kbest_features_train, y_train)

```

```

y_pred = knn.predict(X_kbest_features_test)
scores.append(metrics.accuracy_score(y_test, y_pred))

import matplotlib.pyplot as plt
plt.plot(k_range, scores)
plt.xlabel('Value of k for KNN')
plt.ylabel('Accuracy Score')
plt.title('Accuracy Scores for Values of k of k-Nearest-Neighbors')
plt.show()

knn = KNeighborsClassifier(n_neighbors=12)
knn.fit(X, y)

"""
    knn.predict([[0.990813,-1.50883,-0.436021,1.04317,-0.653768,-
0.426607,0.330896,-0.848089,-1.17851,1.55922,-0.728968,-
0.720422,1.55455,-0.792075,-0.788519,0.372455,0.837616,-0.537015,-
0.414516,-1.45381]])
    knn.predict([[-1.00927,-1.43637,-0.436021,-0.958612,-0.653768,-
0.954689,0.330896,-0.848089,0.175706,-0.641344,-0.728968,-0.720422,-
0.643274,-0.792075,1.2682,-0.826637,0.837616,0.397371,0.439018,1.25887]])
    knn.predict([[0,6511,0,0,0,2,1,0,1,0,0,0,0,0,0,1,2,729,925]])
    knn.predict([[1,2564,0,0,0,2,1,0,0,1,1,0,0,0,0,0,1,3,436,157]])
"""

"""SVM """
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
from sklearn.svm import SVC
svclassifier = SVC(kernel='linear')
svclassifier.fit(X_train, y_train)
y_pred = svclassifier.predict(X_test)
from sklearn.metrics import classification_report, confusion_matrix

```

```
print(confusion_matrix(y_test,y_pred))
print(classification_report(y_test,y_pred))
```

```
"""Random Forest """
```

```
from sklearn.discriminant_analysis import LinearDiscriminantAnalysis as LDA
lda = LDA(n_components=1)
X_train = lda.fit_transform(X_train, y_train)
X_test = lda.transform(X_test)
from sklearn.ensemble import RandomForestClassifier
classifier = RandomForestClassifier(max_depth=2, random_state=0)
classifier.fit(X_train, y_train)
y_pred = classifier.predict(X_test)
from sklearn.metrics import confusion_matrix
from sklearn.metrics import accuracy_score
cm = confusion_matrix(y_test, y_pred)
print(cm)
print('Accuracy' + str(accuracy_score(y_test, y_pred)))
```

B.SCREENSHOTS

FileHomeInsertPage LayoutFormulasDataReviewViewHelpTell me what you want to do

Calibri11A A

B I U

Font

Wrap Text

General

Conditional Formatting

Cell Styles

InsertDeleteFormat

AutoSum

Sort & Filter

Find & Select

PasteCopyFormat PainterClipboard

Font

Alignment

Number

Styles

Cells

Editing

A1

fx

customerID

A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	
1	customerID	gender	Seniority	CITI Partner	Depender	tenure	PhoneSer	Multiple	InternetSer	OnlineSec	OnlineBac	DevicePro	TechSupp	Streaming	Streaming	Contract	Paperless	Payment	MonthlyC	TotalCh	Churn
2	7590-VHV	Female	0	Yes	No	1	No	No	DSL	No	Yes	No	No	No	No	Month-to	Yes	Electronic	29.85	29.85	
3	5575-GNV	Male	0	No	No	34	Yes	No	DSL	Yes	No	No	No	No	No	One year	No	Mailed ch	56.95	1889.5	
4	3668-OPV	Male	0	No	No	2	Yes	No	DSL	Yes	Yes	No	No	No	No	Month-to	Yes	Mailed ch	53.85	108.15	
5	7795-CFO	Male	0	No	No	45	No	No	DSL	Yes	No	Yes	Yes	No	No	One year	No	Bank trans	42.3	1840.75	
6	9237-HQT	Female	0	No	No	2	Yes	No	Fiber opti	No	No	No	No	No	No	Month-to	Yes	Electronic	70.7	151.65	
7	9305-CDS	Female	0	No	No	8	Yes	Yes	Fiber opti	No	No	Yes	No	Yes	Yes	Month-to	Yes	Electronic	99.65	820.5	
8	1452-KIO	Male	0	No	Yes	22	Yes	Yes	Fiber opti	No	Yes	No	No	Yes	No	Month-to	Yes	Credit car	89.1	1949.4	
9	6713-KO	Female	0	No	No	10	No	No	DSL	Yes	No	No	No	No	No	Month-to	No	Mailed ch	29.75	301.9	
10	7892-POO	Female	0	Yes	No	28	Yes	Yes	Fiber opti	No	No	Yes	Yes	Yes	Yes	Month-to	Yes	Electronic	104.8	3046.05	
11	6388-TAB	Male	0	No	Yes	62	Yes	No	DSL	Yes	Yes	No	No	No	No	One year	No	Bank trans	56.15	3487.95	
12	9763-GRS	Male	0	Yes	Yes	13	Yes	No	DSL	Yes	No	No	No	No	No	Month-to	Yes	Mailed ch	49.95	587.45	
13	7469-LXB	Male	0	No	No	16	Yes	No	No	No	No	No	No	No	No	Two year	No	Credit car	18.95	326.8	
14	8091-TTV	Male	0	Yes	No	58	Yes	Yes	Fiber opti	No	No	No	Yes	Yes	Yes	One year	No	Credit car	100.35	5681.1	
15	0280-XUG	Male	0	No	No	49	Yes	Yes	Fiber opti	No	Yes	No	Yes	Yes	Yes	Month-to	Yes	Bank trans	103.7	5036.3	
16	5129-JLP	Male	0	No	No	25	Yes	No	Fiber opti	Yes	No	Yes	Yes	Yes	Yes	Month-to	Yes	Electronic	105.5	2686.05	
17	3655-SNQ	Female	0	Yes	Yes	69	Yes	Yes	Fiber opti	Yes	Yes	Yes	Yes	Yes	Yes	Two year	No	Credit car	113.25	7895.15	
18	8191-KWS	Female	0	No	No	52	Yes	No	No	No	No	No	No	No	No	One year	No	Mailed ch	20.65	1022.95	
19	9599-WOI	Male	0	No	Yes	71	Yes	Yes	Fiber opti	Yes	No	Yes	No	Yes	Yes	Two year	No	Bank trans	106.7	7382.25	
20	4190-MFU	Female	0	Yes	Yes	10	Yes	No	DSL	No	No	Yes	Yes	No	No	Month-to	No	Credit car	55.2	528.35	
21	4183-MYF	Female	0	No	No	21	Yes	No	Fiber opti	No	Yes	Yes	No	No	Yes	Month-to	Yes	Electronic	90.05	1862.9	
22	8779-GRD	Male	1	No	No	1	No	No	DSL	No	No	No	Yes	No	Yes	Month-to	Yes	Electronic	39.65	39.65	
23	1680-VDC	Male	0	Yes	No	12	Yes	No	No	No	No	No	No	No	No	One year	No	Bank trans	19.8	202.25	

Telco Customer Churn

100%

100%

100%

100%

100%

100%

100%

100%

100%

100%

100%

100%

100%

100%

100%

100%

100%

100%

100%

100%

100%

100%

100%

100%

100%

100%

100%

100%

100%

100%

100%

100%

100%

100%

100%

100%

100%

100%

100%

100%

100%

100%

100%

100%

100%

100%

100%

100%

100%

100%

100%

100%

100%

100%

100%

100%

100%

100%

100%

100%

100%

100%

100%

100%

100%

100%

100%

100%

100%

100%

100%

100%

100%

100%

100%

100%

100%

100%

100%

100%

100%

100%

100%

100%

100%

100%

100%

100%

100%

100%

100%

100%

100%

100%

100%

100%

100%

100%

100%

100%

100%

100%

100%

100%

100%

100%

100%

100%

100%

100%

100%

100%

100%

100%

100%

100%

100%

100%

100%

100%

100%

100%

100%

100%

100%

100%

100%

100%

100%

100%

100%

100%

100%

100%

100%

100%

100%

100%

100%

100%

100%

100%

100%

100%

100%

100%

100%

100%

100%

100%

100%

100%

100%

100%

100%

100%

100%

100%

100%

100%

100%

100%

100%

100%

100%

100%

100%

100%

100%

100%

100%

100%

100%

100%

100%

100%

100%

100%

100%

100%

100%

100%

100%

100%

100%

100%

100%

100%

100%

100%

100%

100%

100%

100%

100%

100%

100%

100%

100%

100%

100%

100%

100%

100%

100%

100%

100%

100%

100%

100%

100%

100%

100%

100%

100%

100%

100%

100%

100%

100%

100%

100%

100%

100%

100%

100%

100%

100%

100%

100%

100%

100%

100%

100%

100%

100%

100%

100%

100%

100%

100%

100%

100%

100%

100%

100%

100%

100%

100%

100%

100%

100%

100%

100%

100%

100%

100%

100%

100%

100%

100%

100%

100%

100%

100%

100%

100%

100%

100%

100%

100%

100%

100%

100%

100%

100%

100%

100%

100%

100%

100%

100%

100%

100%

100%

100%

100%

100%

100%

100%

100%

100%

100%

100%

100%

100%

100%

100%

100%

100%

100%

100%

100%

100%

100%

100%

100%

100%

100%

100%

100%

100%

100%

100%

100%

SCREENSHOT FOR MODULE 1

Telco-Customer-Churn																					
customerID	gender	Seniority	Partner	Depender	tenure	PhoneSer	Multiple	InternetSer	OnlineSec	OnlineBac	DevicePro	TechSupp	Streaming	Streaming	Contract	Paperless	Payment	MonthlyC	TotalCh	Churn	
1	customerID	gender	Seniority	Partner	Depender	tenure	PhoneSer	Multiple	InternetSer	OnlineSec	OnlineBac	DevicePro	TechSupp	Streaming	Streaming	Contract	Paperless	Payment	MonthlyC	TotalCh	Churn
2	7590-VHV	Female	0	Yes	No	1	No	DSL	No	Yes	No	No	No	No	Month-to	Yes	Electronic	29.85	29.85		
3	5575-GNV	Male	0	No	No	34	Yes	No	DSL	Yes	No	No	No	No	One year	No	Mailed ch	56.95	1889.5		
4	3668-OPV	Male	0	No	No	2	Yes	No	DSL	Yes	Yes	No	No	No	Month-to	Yes	Mailed ch	53.85	108.15		
5	7795-CFO	Male	0	No	No	45	No	No	DSL	Yes	No	Yes	No	No	One year	No	Bank trans	42.3	1840.75		
6	9237-HQT	Female	0	No	No	2	Yes	No	Fiber opti	No	No	No	No	No	Month-to	Yes	Electronic	70.7	151.65		
7	9305-CDS	Female	0	No	No	8	Yes	Yes	Fiber opti	No	No	Yes	No	Yes	Month-to	Yes	Electronic	99.65	820.5		
8	1452-KIO	Male	0	No	Yes	22	Yes	Yes	Fiber opti	No	Yes	No	No	Yes	Month-to	Yes	Credit car	89.1	1949.4		
9	6713-KO	Female	0	No	No	10	No	No	DSL	Yes	No	No	No	No	Month-to	No	Mailed ch	29.75	301.9		
10	7892-POO	Female	0	Yes	No	28	Yes	Yes	Fiber opti	No	No	Yes	Yes	Yes	Month-to	Yes	Electronic	104.8	3046.05		
11	6388-TAB	Male	0	No	Yes	62	Yes	No	DSL	Yes	Yes	No	No	No	One year	No	Bank trans	56.15	3487.95		
12	9763-GRS	Male	0	Yes	Yes	13	Yes	No	DSL	Yes	No	No	No	No	Month-to	Yes	Electronic	104.8	3046.05		
13	7469-LXB	Male	0	No	No	16	Yes	No	No	No	No	No	No	No	Two year	No	Credit car	18.95	326.8		
14	8091-TTV	Male	0	No	No	58	Yes	Yes	Fiber opti	No	No	Yes	No	Yes	One year	No	Credit car	100.35	5681.1		
15	0280-XUG	Male	0	No	No	49	Yes	Yes	Fiber opti	No	Yes	No	Yes	Yes	Month-to	Yes	Bank trans	103.7	5036.3		
16	5129-JLP	Male	0	No	No	25	Yes	No	Fiber opti	Yes	No	Yes	Yes	Yes	Month-to	Yes	Electronic	105.5	2686.05		
17	3655-SNQ	Female	0	Yes	Yes	69	Yes	Yes	Fiber opti	Yes	Yes	Yes	Yes	Yes	Two year	No	Credit car	113.25	7895.15		
18	8191-KWS	Female	0	No	No	52	Yes	No	No	No	No	No	No	No	One year	No	Mailed ch	20.65	1022.95		
19	9599-WOI	Male	0	No	Yes	71	Yes	Yes	Fiber opti	Yes	No	Yes	No	Yes	Two year	No	Bank trans	106.7	7382.25		
20	4190-MFU	Female	0	Yes	Yes	10	Yes	No	DSL	No	No	Yes	Yes	No	Month-to	No	Credit car	55.2	528.35		
21	4183-MYF	Female	0	No	No	21	Yes	No	Fiber opti	No	Yes	Yes	No	No	Month-to	Yes	Electronic	90.05	1862.9		
22	8779-GRD	Male	1	No	No	1	No	No	DSL	No	No	Yes	No	No	Month-to	Yes	Electronic	39.65	39.65		
23	1680-VDC	Male	0	Yes	No	12	Yes	No	No	No	No	No	No	No	One year	No	Bank trans	19.8	202.25		

SCREENSHOT FOR MODULE 2

