

Intelligent Network Data Analytics Function in Telecom Customer Churn using Machine Learning

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

by

YAMANABOINA PRAVALLIKA (38110660)
ARNIMA CHATTERJEE (38117001)



**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

MARCH - 2022



SATHYABAMA

INSTITUTE OF SCIENCE AND TECHNOLOGY

(DEEMED TO BE UNIVERSITY)

Accredited "A" Grade by NAAC | 12B Status by UGC | Approved by AICTE

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 YAMANABOINA PRAVALLIKA (38110660), ARNIMA CHATTERJEE (38117001) who carried out the project entitled “ **Intelligent Network Data Analytics Function in Telecom Customer Churn using Machine Learning**”

under my supervision from October 2021 to May 2022

InternalGuide

(Dr. M SELVI .M.E(Ph.D))

Head of the Department

Submitted for Viva voce Examination held on_____

Internal Examiner

External Examiner

DECLARATION

I YAMANABOINA PRAVALLIKA, ARNIMA CHATTERJEE hereby declare that the Project Report entitled **Intelligent Network Data Analytics Function in Telecom Customer Churn using Machine Learning** done by me under the guidance of **Dr. M SELVI .M.E(Ph.D)** (Internal) is submitted in partial fulfillment of the requirements for the award of Bachelor of Engineering / Technology degree In COMPUTER SCIENCE AND TECHNOLOGY.

DATE:

PLACE:
CANDIDATE

SIGNATURE OF THE

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. L. Lakshmanan M.E., Ph.D. , and **Dr.S.Vigneshwari 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. M SELVI .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 of the **Department of Computer Science and Engineering** who were helpful in many ways for the completion of the project.

Abstract

The fifth generation of cellular mobile communication called 5G will be publicly available in the near future to connect more than 8.4 billion devices. 5G has been designed to improve the network capacity and speed significantly. Various providers and Android cell phone producers have contributed to this huge project and have successfully tested their networks and devices. Additionally, 5G is a revolutionizing technology in which smart devices, including Android phones, are capable of utilizing artificial intelligence software applications. In addition, other emerging technologies, including autonomous driving that rely highly on network communications, will be positively affected. The business aspect of 5G is also promising, as were the business aspects of the previous generations. 5G will be financially beneficial, as step by step, users will update their devices, or they will have to purchase a device that is compatible to 5G, which will produce a high financial turnover for the telecom companies. This review briefly addresses the technical background and 5G key technologies. In the second section, the cutting-edge development and state-of-the-art 5G related technologies will be discussed. Currently, 5G is gradually being adopted as the future of the telecommunication industry promising network reliability and faster download speeds, with immense opportunities both for consumers and businesses. Service providers across the world are in the development phase for 5G networks with the expectation of increased implementation of 5G in 2020.

CONTENT

Chapter	List of content	Page no.
I	Abstract	i
ii	List of figures	iii
1	Introduction	1
2	Literature Survey	10
3	System Requirements	13
	3.1 Hardware Requirements	13
	3.2 Software Requirements	13
	3.3 Language Specification	13
	3.4 History of python	13
	3.5 Application of python	14
	3.6 Features of python	15
	3.7 Feasibility of study	15
	3.7.1 Economical feasibility	16
	3.7.2 Technical feasibility	16
	3.7.3 Social feasibility	17
4	System Analysis	18
	4.1 Purpose	18
	4.2 Scope	18
	4.3 Exsisting System	18
	4.4 Proposed system	19
5	System Design	21
	5.1 Input design	21
	5.2 Output design	21
	5.3 Data Flow Diagram	22
6	Modules Description	26
7	System implementation	28
	7.1 System architecture	28
8	System testing	29
	8.1 Test plan	29
	8.2 Verification	29
	8.3 Validation	29
	8.4 Basics of software testing	29
	8.5 Black box testing	30
	8.6 White box testing	30
	8.7 Types of testing	30

	8.7.1	Unit testing	30
	8.7.2	Integration testing	31
	8.7.3	Functional testing	31
	8.7.4	System testing	31
	8.7.5	Stress testing	31
	8.7.6	Performance testing	32
	8.7.7	Usability testing	32
	8.7.8	Acceptance testing	32
	8.7.9	Regression testing	32
9		Result and Discussion	36
iv		References	56

LIST OF FIGURES

FIGURE NO	FIGURE NAME	PAGE NO
1	Working Algorithm of RD algorithm	4
2	Random Forest Classifier	6
3	Example image of RD algorithm	7
4	Level 1	24
5	Level 2	24
6	Level 3	25
7	Level 4	25
8	Level 5	25
9	Architecture Diagram	28

CHAPTER 1

INTRODUCTION

Lately, the use of data analytics to support the operation of 5G cellular networks has emerged as a hot research topic. The application of data analytics covers a wide range of network operations such as network management and control as well as policy enforcement. Figure 1 illustrates how this is achieved. More specifically, data are collected from a number of network components and may include a variety of information fields such as the quality of the wireless channel, the network load, accounting information, configuration and fault indications, the profile of the subscribers, etc. This data is stored and updated regularly. When collected, it is preprocessed, and any necessary transformation, discretization, normalization, outlier detection and dimensionality reduction is applied. The result of this process is then provided to a data analysis procedure which builds a model that caters for knowledge extraction from the processed data. For example, the result of this process may be the identification of situations where the occurrence of some specific events (e.g., a significant increase of the number of high mobility users). Causes a given result (e.g., increase of the handover blocking probability). The knowledge model may also include some solutions for specific situations (e.g., force the network components to place high moving users to macro cells). The knowledge discovery results can then be communicated to either policy, management or control modules. These modules can employ this information in order to optimize the operation of the network and improve its performance. In this paper we provide a complete framework on how data analytics can be used to support Radio Access Technology (RAT) selection and traffic steering.

The 5th generation (5G) of cellular networks is being designed to satisfy the massive growth in capacity demand, number of connections and the evolving use cases of a connected society for 2020 and beyond [2]. In particular, 5G networks target the following KPIs: (i) very high throughput, in the order of 1 Gbps or more, to enable virtual reality applications and high-quality video streaming; (ii) ultralow latency, possibly smaller than 1 ms on the wireless link, to support autonomous control applications; (iii) ultrahigh reliability; (iv) low energy consumption; and (v) high availability of robust connections [3], [4]. In order to meet these requirements, a new approach in the design of the network is required, and new paradigms have recently emerged [4]. First, the densification of the network will increase the spatial reuse and, combined with the usage of mmWave frequencies, the available throughput. On the other hand, this will introduce new challenges related to mobility management [5]. Second, with Mobile Edge Cloud (MEC), the content will be brought closer to the final users, in order to decrease the end-to-end latency [4]. Third, a higher level of automation will be introduced in cellular networks, relying on ML techniques and Software Defined Networking (SDN), in order to manage the increased complexity of 5G networks. The usage of ML and Artificial Intelligence (AI) techniques to perform autonomous operations in cellular networks has been widely studied in recent years, with use cases that range from optimization of video flows [6] to energy-efficient networks [7] and resource allocation [8]. This trend is coupled with the application of big-data analytics that leverage the huge amount of monitoring data generated in mobile networks to provide more insights on the behavior of networks at scale [9]. In the domain of mobile networks, these two technological components can empower costs savings, but also new applications, as we will show in this paper. However, despite

the importance of this topic, the state of the art lacks considerations on how it is possible to effectively deploy machine learning algorithms and intelligence in cellular networks, and an evaluation of the gains of a data-driven approach with real large-scale network datasets.

1.1 Scope

This document defines a unified architecture for Machine Learning in Fifth Generation and future networks. The unified logical architecture establishes a common vocabulary and nomenclature for ML functions and their interfaces. By applying (or superimposing) this logical architecture to a specific technology, like 3GPP or transport networks, the corresponding technology specific realization is derived. Such an overlay architecture for ML allows standardization and achieves interoperability for ML functions in 5G and future networks. In addition, the key issues for integration of such an overlay on 5G and future networks are studied and presented in this document.

1.1 PROPOSED ALGORITHMS

RANDOM FOREST ALGORITHM

RANDOM FOREST ALGORITHM

Random forest algorithm can use both for classification and the regression kind of problems. In this you are going to learn, how the **random forest algorithm** works in machine learning for the classification task.

Random Forest is a popular machine learning algorithm that belongs to the supervised learning technique. It can be used for both Classification and Regression problems in ML. It is based on the concept of **ensemble learning**, which is a process of combining multiple classifiers to solve a complex problem and to improve the performance of the model.

A random forest algorithm consists of many decision trees. The 'forest' generated by the random forest algorithm is trained through bagging or bootstrap aggregating. Bagging is an ensemble meta-algorithm that improves the accuracy of machine learning algorithms.

As the name suggests, "Random Forest is a classifier that contains a number of decision trees on various subsets of the given dataset and takes the average to improve the predictive accuracy of that dataset." Instead of relying on one decision tree, the random forest takes the prediction from each tree and based on the majority votes of predictions, and it predicts the final output.

The below diagram explains the working of the Random Forest algorithm:

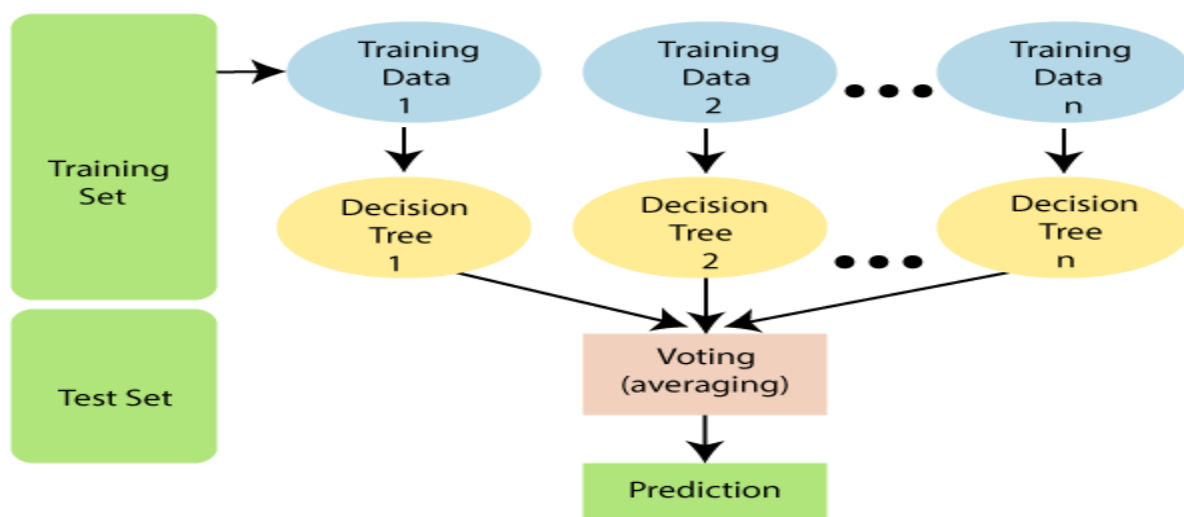


Fig 1: Explaining the working algorithm of the Random Forest algorithm

Below are some points that explain why we should use the Random Forest algorithm:

- It takes less training time as compared to other algorithms.
- It predicts output with high accuracy, even for the large dataset it runs efficiently.
- It can also maintain accuracy when a large proportion of data is missing.

Features of a Random Forest Algorithm

- It's more accurate than the decision tree algorithm.
- It provides an effective way of handling missing data.
- It can produce a reasonable prediction without hyper-parameter tuning.
- It solves the issue of overfitting in decision trees.
- In every random forest tree, a subset of features is selected randomly at the node's splitting point.

Classification in random forests

Classification in random forests employs an ensemble methodology to attain the outcome. The training data is fed to train various decision trees. This dataset consists of observations and features that will be selected randomly during the splitting of nodes.

A rain forest system relies on various decision trees. Every decision tree consists of decision nodes, leaf nodes, and a root node. The leaf node of each tree is the final output produced by that specific decision tree. The selection of the final output follows the majority-voting system. In this case, the output chosen by the majority of the decision trees becomes the final

output of the rain forest system. The diagram below shows a simple random forest classifier.

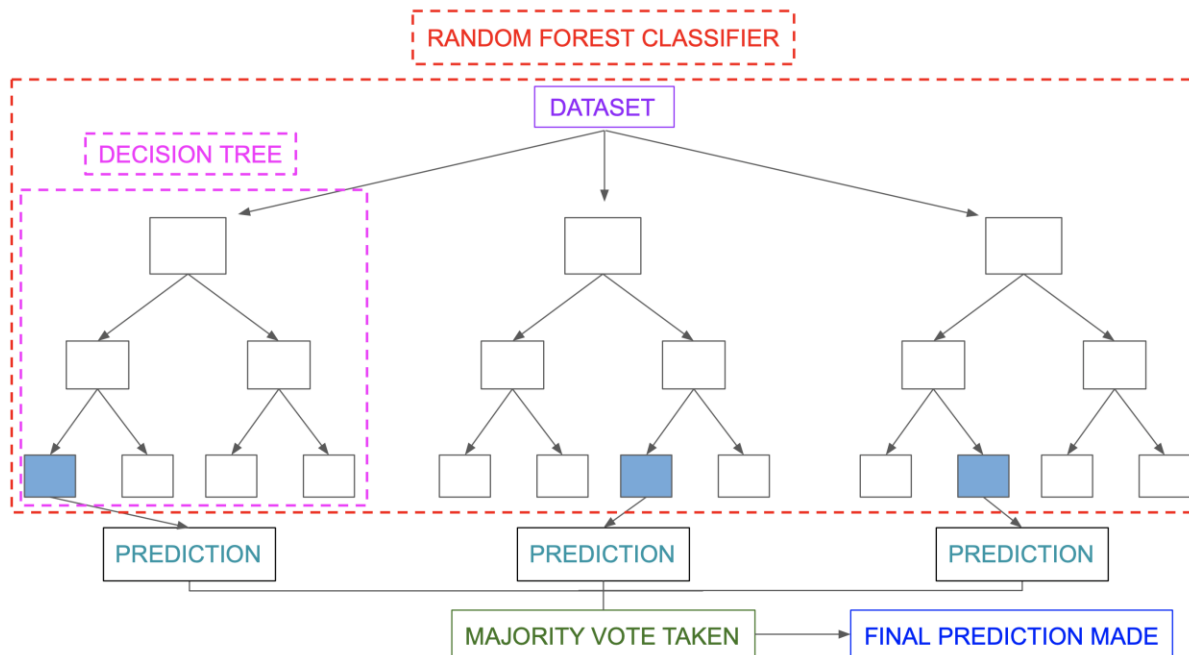


Fig 2: Explaining the Random Forest Classifier

Random Forest Steps

1. Randomly select “k” features from total “m” features.
 1. Where $k \ll m$
2. Among the “k” features, calculate the node “d” using the best split point.
3. Split the node into daughter nodes using the best split.
4. Repeat 1 to 3 steps until “l” number of nodes has been reached.
5. Build forest by repeating steps 1 to 4 for “n” number times to create “n” number of trees.

The beginning of random forest algorithm starts with randomly selecting “k” features out of total “m” features. In the image, you can observe that we are randomly taking features and observations.

The working of the algorithm can be better understood by the below example:

Example: Suppose there is a dataset that contains multiple fruit images. So, this dataset is given to the Random forest classifier. The dataset is divided into subsets and given to each decision tree. During the training phase, each decision tree produces a prediction result, and when a new data point occurs, then based on the majority of results, the Random Forest classifier predicts the final decision. Consider the below image:

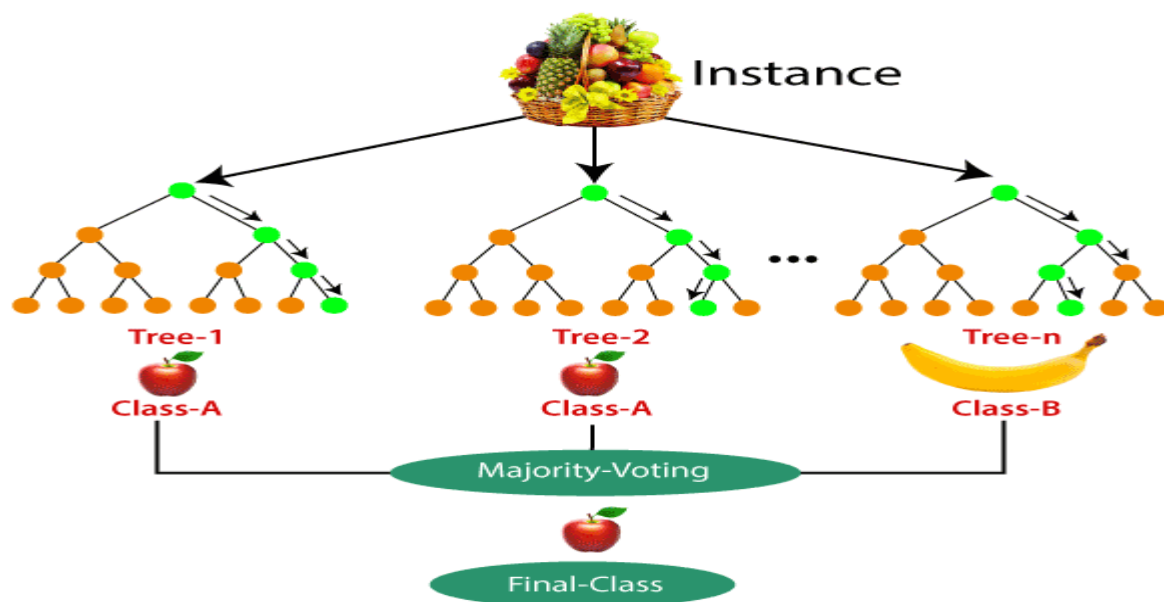


Fig 3: Explaining the Random Forest Classifier algorithm with example

APPLICATIONS OF RANDOM FOREST

There are mainly four sectors where Random forest mostly used:

1. **Banking:** Banking sector mostly uses this algorithm for the identification of loan risk.
2. **Medicine:** With the help of this algorithm, disease trends and risks of the disease can be identified.
3. **Land Use:** We can identify the areas of similar land use by this algorithm.
4. **Marketing:** Marketing trends can be identified using this algorithm.

Advantages of Random Forest

Random Forest is capable of performing both Classification and Regression tasks.

- It is capable of handling large datasets with high dimensionality.
- It enhances the accuracy of the model and prevents the overfitting issue.

Disadvantages of Random Forest

- Although random forest can be used for both classification and regression tasks, it is not more suitable for Regression tasks.

Python Implementation of Random Forest Algorithm

Now we will implement the Random Forest Algorithm tree using Python. For this, we will use the same dataset "user_data.csv", which we have used in previous classification models. By using the same dataset, we can compare the Random Forest classifier with other classification models such as Decision tree Classifier, KNN, SVM, Logistic Regression, etc.

Implementation Steps are given below:

- Data Pre-processing step
- Fitting the Random forest algorithm to the Training set
- Predicting the test result
- Test accuracy of the result (Creation of Confusion matrix)
- Visualizing the test set result.

CHAPTER 2

LITERATURE SURVEY

Optical access network technology for 5G wireless front/backhaul network

This paper describes trends in 5G mobile networks and RAN technologies. Cloud RAN, which uses switches to enable flexible MFH networking, and two promising candidates of MFH in Cloud RAN are discussed.

Optimal Slice Allocation in 5G Core Networks

5G network slicing is essential to providing flexible, scalable, and on-demand solutions for the vast array of applications in 5G networks. Two key challenges of 5G network slicing are function isolation (intra-slice) and guaranteeing end-to-end delay for a slice. In this letter, we address the question of optimal allocation of a slice in 5G core networks by tackling these two challenges. We adopt and extend the work by Dietrich et al. (see introduction) to create a model that satisfies constraints on end-to-end delay as well as isolation between components of a slice for reliability.

Virtualising and orchestrating a 5G evolved packet core network

In this paper, the design, construction, and testing of a fully-functional 10ireless10at mobile core network is outlined. Lessons learned and recommendations for future improvements are provided. The presented work uses open-source software for virtual network function infrastructure

control (OpenStack), network flow programming (OpenDaylight), and network orchestration (OpenBaton) to 11ireless11a a commercial software evolved packet core solution deployed on common off-the-shelf hardware. The findings presented in this paper prove the concept of function 11ireless11ation for mobile networks, and paves the way towards future mobile core network function flexibility as required for 5G networks. The paper provides researchers and network operators with first-hand experience to help build similar virtual mobile network infrastructures, and highlights the challenges to tackle and the issues to address to harness the power of 11ireless11ation in 5G networks.

An11ireless11ation of energy efficiency in 5G wireless networks

In next generation wireless networks along with the overwhelming demand of high data rate and network capacity, the user demands ubiquitous connectivity with the network. In order to fulfill the demand of anywhere at any time data services, the network operators have to install more and more base stations that eventually leads towards high power consumption. For this, the potential solution is derived from 5G network that proposes a heterogeneous environment of wireless access networks. More particularly, deployment of Femto and Pico cell under the umbrella of Macro cell base stations (BS). Such networking strategy will result high network capacity and energy efficiency along with better network coverage. In this article, an analysis of energy efficiency has been carried out by using two-tier and three tier network configurations. The simulation results demonstrate that rational deployment of small cells improves the energy efficiency of wireless network.

5G network challenges and realization insights

5G technologies are a new paradigm strongly supported by the European Commission (EC) to overcome the challenges of next generation networks and aiming to tackle the novel and manifold business requirements of vertical sectors such as Industry 4.0, Smart Grids, Smart Cities, and eHealth. Extraordinarily high speeds and capacity, multi-tenancy, fixed and wireless access network convergence, self-X, unconventional resource allocation, on-demand service-oriented resource allocation and automated management of resources are only few examples of the complex demands that 5G aims to undertake. The accommodation of such features requires multipronged efforts in different network technology domains at various levels. In this paper, we investigate how the relative 5G network features challenge the current generation of access networks, both fixed and wireless, and data centres, from an end-to-end network service point of view. Furthermore, we provide some insights into how emerging technologies such as Network Function Virtualisation (NFV) and Software Defined Networks (SDN) can be utilized for the accommodation of the above mentioned features and realization of 5G networks.

CHAPTER 3

SYSTEM ARCHITECTURE

3.1 HARDWARE REQUIREMENTS:

- System : Pentium Dual Core.
- Hard Disk : 120 GB.
- Monitor : 15” LED
- Input Devices : Keyboard, Mouse
- Ram : 4 GB.

3.2 SOFTWARE REQUIREMENTS:

- Operating system: Windows 10.
- Coding Language : Python

3.3 LANGUAGE SPECIFICATION

Python is a general-purpose interpreted, interactive, object-oriented, and high-level programming language. It was created by Guido van Rossum during 1985- 1990. Like Perl, Python source code is also available under the GNU General Public License (GPL). This **tutorial** gives enough understanding on **Python programming** language.

3.4. HISTORY OF PYTHON

Python was developed by Guido van Rossum in the late eighties and early nineties at the National Research Institute for Mathematics and Computer Science in the Netherlands.

Python is derived from many other languages, including ABC, Modula-3, C, C++, Algol-68, SmallTalk, and Unix shell and other scripting languages.

Python is copyrighted. Like Perl, Python source code is now available under the GNU General Public License (GPL).

Python is now maintained by a core development team at the institute, although Guido van Rossum still holds a vital role in directing its progress.

3.5. APPLICATION OF PYTHON

- **Easy-to-learn** – Python has few keywords, simple structure, and a clearly defined syntax. This allows the student to pick up the language quickly.
- **Easy-to-read** – Python code is more clearly defined and visible to the eyes.
- **Easy-to-maintain** – Python's source code is fairly easy-to-maintain.
- **A broad standard library** – Python's bulk of the library is very portable and cross-platform compatible on UNIX, Windows, and Macintosh.
- **Interactive Mode** – Python has support for an interactive mode which allows interactive testing and debugging of snippets of code.
- **Portable** – Python can run on a wide variety of hardware platforms and has the same interface on all platforms.
- **Extendable** – You can add low-level modules to the Python interpreter. These modules enable programmers to add to or customize their tools to be more efficient.

- **Databases** – Python provides interfaces to all major commercial databases.
- **GUI Programming** – Python supports GUI applications that can be created and ported to many system calls, libraries and windows systems, such as Windows MFC, Macintosh, and the X Window system of Unix.
- **Scalable** – Python provides a better structure and support for large programs than shell scripting.

3.6 FEATURES OF PYTHON

- It supports functional and structured programming methods as well as OOP.
- It can be used as a scripting language or can be compiled to byte-code for building large applications.
- It provides very high-level dynamic data types and supports dynamic type checking.
- It supports automatic garbage collection.
- It can be easily integrated with C, C++, COM, ActiveX, CORBA, and Java.

3.7 FEASIBILITY STUDY

The feasibility of the project is analyzed in this phase and business proposal is put forth with a very general plan for the project and some cost estimates. During system analysis the feasibility study of the proposed system is to be carried out. This is to ensure that the proposed system is not a burden to the company. For feasibility analysis, some understanding of the major requirements for the system is essential.

The feasibility study investigates the problem and the information needs of the stakeholders. It seeks to determine the resources required to provide an information systems solution, the cost and benefits of such a solution, and the feasibility of such a solution.

The goal of the feasibility study is to consider alternative information systems solutions, evaluate their feasibility, and propose the alternative most suitable to the organization. The feasibility of a proposed solution is evaluated in terms of its components.

3.7.1 ECONOMICAL FEASIBILITY

This study is carried out to check the economic impact that the system will have on the organization. The amount of fund that the company can pour into the research and development of the system is limited. The expenditures must justified. Thus the developed system as well within the budget and this was achieved because most of the technologies used are freely available. Only the customized products had to be purchased.

3.7.2 TECHNICAL FEASIBILITY

This study is carried out to check the technical feasibility, that is, the technical requirements of the system. Any system developed must not have a high demand on the available technical resources. This will lead to high demands on the available technical resources. This will lead to high demands being placed on the client. The developed system must have a modest requirement, as only minimal or null changes are required for implementing this system.

3.7.3 SOCIAL FEASIBILITY

The aspect of study is to check the level of acceptance of the system by the user. This includes the process of training the user to use the system efficiently. The user must not feel threatened by the system, instead must accept it as a necessity.

CHAPTER 4

SYSTEM ANALYSIS

4.1 PURPOSE

The purpose of this document is intelligent network data analytics function in 5G cellular networks using machine learning algorithms. In detail, this document will provide a general description of our project, including user requirements, product perspective, and overview of requirements, general constraints. In addition, it will also provide the specific requirements and functionality needed for this project - such as interface, functional requirements and performance requirements.

4.2 SCOPE

The scope of this SRSdocument persists for the entire life cycle of the project. This document defines the final state of the software requirements agreed upon by the customers and designers. Finally at the end of the project execution all the functionalities may be traceable from the SRSto the product. The document describes the functionality, performance, constraints, interface and reliability for the entire life cycle of the project.

4.3 EXISTING SYSTEMS

4G, some of the prime objectives or demands that need to be addressed are increased capacity, improved data rate, decreased latency, and better quality of service. To meet these demands, drastic improvements need to be made in cellular network architecture. This paper presents the results of a detailed survey on the fifth generation (5G) cellular network architecture and

some of the key emerging technologies that are helpful in improving the architecture and meeting the demands of users.

DISADVANTAGES OF EXISTING SYSTEM

- Old devices can't support this- Hardware replacement will required.
- At present, 5G technology is theoretical, it may take longer than expected to come into existence.

4.4 PROPOSED SYSTEM

In the current technologically advanced and interconnected world, both success and survival rely on adaptation to changes by evolving with time. One of the most significant changes witnessed in the telecommunication industry is the ever-increasing demand for high-speed connectivity and faster protocols for data transfer. The rapidly increasing demand for high-speed connectivity has stretched 4G to its limit rendering it ineffective to enable the next phase in telecommunication. The 5G network architecture is user-centric, while 4G is application-centric. 5G offers high-speed connectivity needed to increase data access. Ultra-low latency connection offered by 5G is expected to increase the adoption of Internet by enabling the interconnection of sensors and devices. Internet applications, such as asset and data tracking, forecasting trends, and industrial automation, require ultra-low latency to attain the highest level of functionality.

ADVANTAGES OF PROPOSED SYSTEM

- High data rates per area and massive a large number of connected devices. Thus, the interference among transmitters should be minimized.
- Congestion control and prioritization of data packets

- Reliability and cross-layering between network and transport layers
- Secure localization
- Extremely high data rates per device.

CHAPTER 5

SYSTEM DESIGN

5.1 INPUT DESIGN

The input design is the link between the information system and the user. It comprises the developing specification and procedures for data preparation and those steps are necessary to put transaction data in to a usable form for processing can be achieved by inspecting the computer to read data from a written or printed document or it can occur by having people keying the data directly into the system. The design of input focuses on controlling the amount of input required, controlling the errors, avoiding delay, avoiding extra steps and keeping the process simple. The input is designed in such a way so that it provides security and ease of use with retaining the privacy. Input Design considered the following things:

- What data should be given as input?
- How the data should be arranged or coded?
- The dialog to guide the operating personnel in providing input.
- Methods for preparing input validations and steps to follow when error occur.

5.2 OUTPUT DESIGN

A quality output is one, which meets the requirements of the end user and presents the information clearly. In any system results of processing are communicated to the users and to other system through outputs. In output design it is determined how the information is to be displaced for immediate need and also the hard copy output. It is the most important and direct source

information to the user. Efficient and intelligent output design improves the system's relationship to help user decision-making.

The output form of an information system should accomplish one or more of the following objectives.

- Convey information about past activities, current status or projections of the
- Future.
- Signal important events, opportunities, problems, or warnings.
- Trigger an action.
- Confirm an action

5.3 DATA FLOW DIAGRAM

1. The DFD is also called as bubble chart. It is a simple graphical formalism that can be used to represent a system in terms of input data to the system, various processing carried out on this data, and the output data is generated by this system.
2. The data flow diagram (DFD) is one of the most important modeling tools. It is used to model the system components. These components are the system process, the data used by the process, an external entity that interacts with the system and the information flows in the system.
3. DFD shows how the information moves through the system and how it is modified by a series of transformations. It is a graphical technique that depicts information flow and the transformations that are applied as data moves from input to output.

4. DFD is also known as bubble chart. A DFD may be used to represent a system at any level of abstraction. DFD may be partitioned into levels that represent increasing information flow and functional detail.
5. The DFD is also called as bubble chart. It is a simple graphical formalism that can be used to represent a system in terms of input data to the system, various processing carried out on this data, and the output data is generated by this system.
6. The data flow diagram (DFD) is one of the most important modeling tools. It is used to model the system components. These components are the system process, the data used by the process, an external entity that interacts with the system and the information flows in the system.
7. DFD shows how the information moves through the system and how it is modified by a series of transformations. It is a graphical technique that depicts information flow and the transformations that are applied as data moves from input to output.
8. DFD is also known as bubble chart. A DFD may be used to represent a system at any level of abstraction. DFD may be partitioned into levels that represent increasing information flow and functional detail.

FIG 4:LEVEL-1

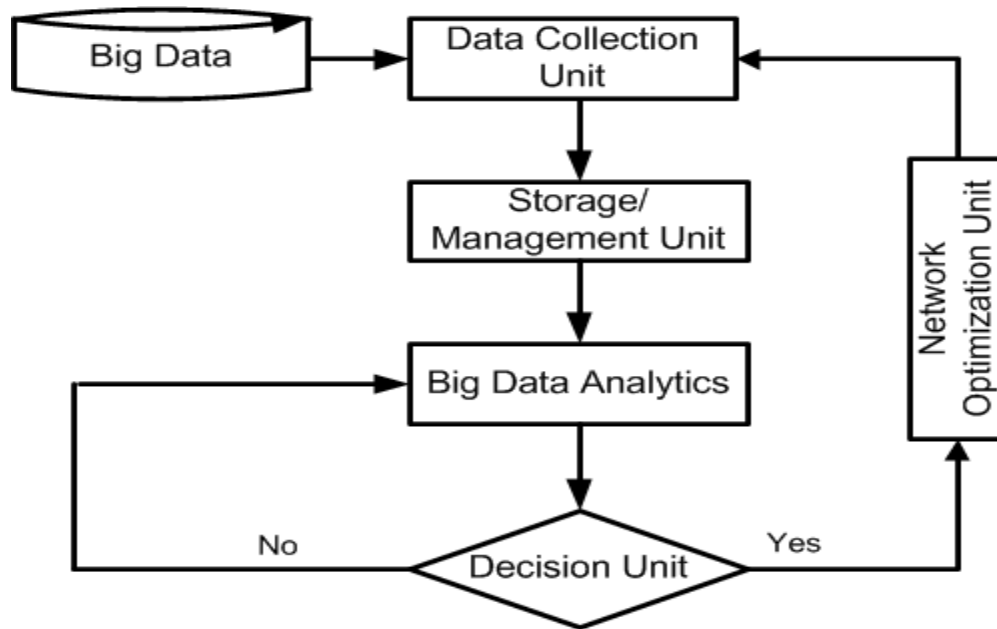


FIG 5:LEVEL-2

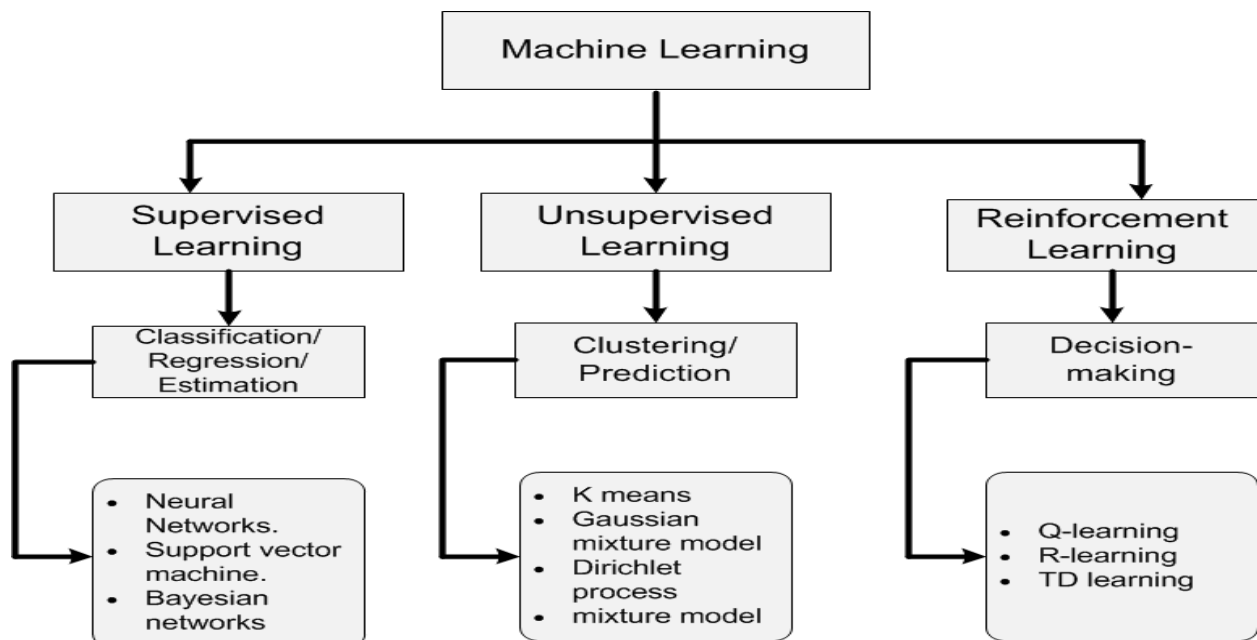


FIG 6:LEVEL -3



FIG 7:LEVEL-4

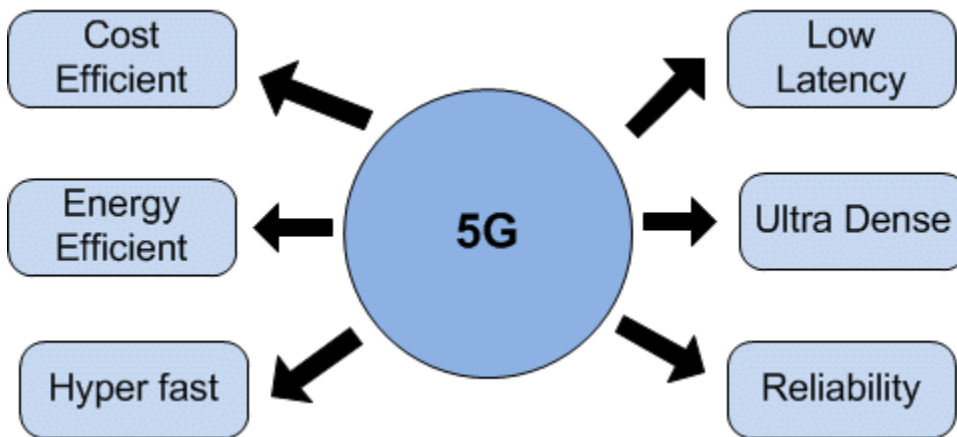
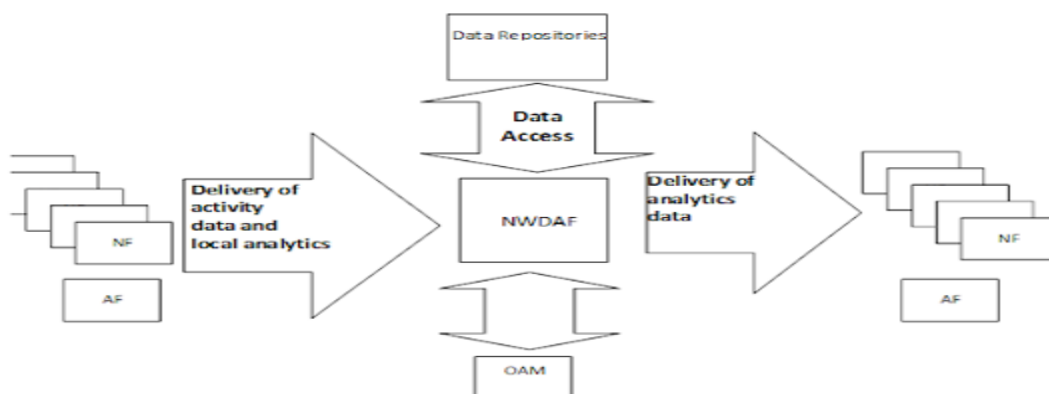


FIG 8

FIG 8:LEVEL-5



CHAPTER 6

6.1 MODULES

- ML in cellular networks
- Data Preprocessing
- Data analytics
- A Context Information Preprocessing module

MODULES DESCRIPTION

6.1.1 ML in cellular networks:

The application of ML techniques to cellular networks is a topic that has gained a lot of attention recently, thanks to the revived importance of ML and AI throughout all facets of the industry. The surveys in [1] present some recent results on how it is possible to apply regression techniques to mobile and cellular scenarios in order to optimize the network performance. The paper gives an overview of how machine learning can play a role in next-generation 5G cellular networks, and lists relevant ML techniques and algorithms. The usage of bigdata-driven analytics for 5G is considered in, with a discussion of how data-driven approaches can empower self-organizing networks. However, none of these papers provides results based on real operators datasets at large scale that show the actual gains of data-driven and machine learning based approaches.

6.1.2 Data Preprocessing

The performance analysis presented in this Section is based on the Sample dataset. We sampled the number of users in each base station with a time

step $T_s = 5$ minutes, and divided the dataset into a training set (which will be used for k-fold cross validation) and a test set.

6.1.3 Data analytics

Data analytics in simple terms refers to the analysis of data. The process involves continuous extraction of data, analyzing the extracted information and forming a behavioral pattern. The analyzed information helps to improve the network/system performance, capacity and availability. The process can vary from Business-to-Business based on the requirements.

6.1.4 A Context Information Preprocessing module

One of the most critical aspects of network context-based systems is the vast volumes of data. that need to be aggregated and transmitted towards a specific network analytics function deployed in the core network. The CIP mechanism performs context data aggregation, filtering and preprocessing techniques. The module comprises aggregating and compressing mobile network-related context information, as well as techniques related to identifying and discarding user profile-redundant or unnecessary context data, before any transmission to the network data analytics function

CHAPTER 7

SYSTEM IMPLEMENTATION

7.1 SYSTEM ARCHITECTURE

Describing the overall features of the software is concerned with defining the requirements and establishing the high level of the system. During architectural design, the various web pages and their interconnections are identified and designed. The major software components are identified and decomposed into processing modules and conceptual data structures and the interconnections among the modules are identified. The following modules are identified in the proposed system.

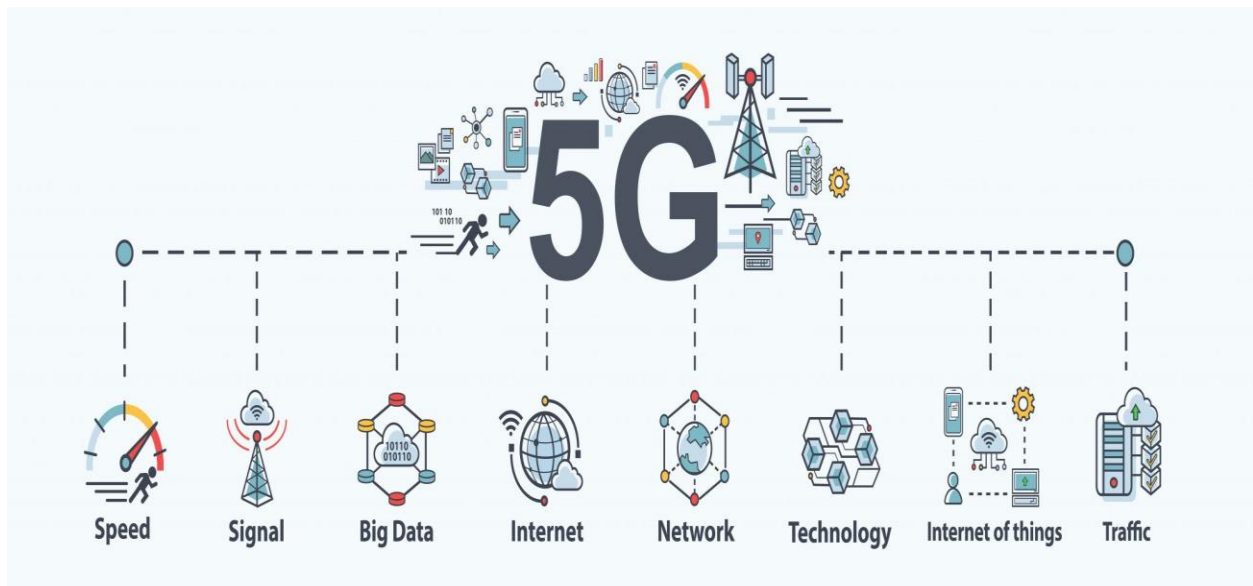


FIG 9: Architecture Diagram

CHAPTER 8

SYSTEM TESTING

8.1 Test plan

Software testing is the process of evaluation a software item to detect differences between given input and expected output. Also to assess the feature of a software item. Testing assesses the quality of the product. Software testing is a process that should be done during the development process. In other words software testing is a verification and validation process.

8.2 Verification

Verification is the process to make sure the product satisfies the conditions imposed at the start of the development phase. In other words, to make sure the product behaves the way we want it to.

8.3 Validation

Validation is the process to make sure the product satisfies the specified requirements at the end of the development phase. In other words, to make sure the product is built as per customer requirements.

8.4 Basics of software testing

There are two basics of software testing: black box testing and white box testing.

8.5 Black box Testing

Black box testing is a testing technique that ignores the internal mechanism of the system and focuses on the output generated against any input and execution of the system. It is also called functional testing.

8.6 White box Testing

White box testing is a testing technique that takes into account the internal mechanism of a system. It is also called structural testing and glass box testing. Black box testing is often used for validation and white box testing is often used for verification.

8.7 Types of testing

There are many types of testing like

- Unit Testing
- Integration Testing
- Functional Testing
- System Testing
- Stress Testing
- Performance Testing
- Usability Testing
- Acceptance Testing
- Regression Testing
- Beta Testing

8.7.1 Unit Testing

Unit testing is the testing of an individual unit or group of related units. It falls under the class of white box testing. It is often done by the programmer to test that the unit he/she has implemented is producing expected output against given input.

8.7.2 Integration Testing

Integration testing is testing in which a group of components are combined to produce output. Also, the interaction between software and hardware is tested in integration testing if software and hardware components have any relation. It may fall under both white box testing and black box testing.

8.7.3 Functional Testing

Functional testing is the testing to ensure that the specified functionality required in the system requirements works. It falls under the class of black box testing.

8.7.4 System Testing

System testing is the testing to ensure that by putting the software in different environments (e.g., Operating Systems) it still works. System testing is done with full system implementation and environment. It falls under the class of black box testing.

8.7.5 Stress Testing

Stress testing is the testing to evaluate how system behaves under unfavorable conditions. Testing is conducted at beyond limits of the specifications. It falls under the class of black box testing.

8.7.6 Performance Testing

Performance testing is the testing to assess the speed and effectiveness of the system and to make sure it is generating results within a specified time as in performance requirements. It falls under the class of black box testing.

8.7.7 Usability Testing

Usability testing is performed to the perspective of the client, to evaluate how the GUI is user-friendly? How easily can the client learn? After learning how to use, how proficiently can the client perform? How pleasing is it to use its design? This falls under the class of black box testing.

8.7.8 Acceptance Testing

Acceptance testing is often done by the customer to ensure that the delivered product meets the requirements and works as the customer expected. It falls under the class of black box testing.

8.7.9 Regression Testing

Regression testing is the testing after modification of a system, component, or a group of related units to ensure that the modification is

working correctly and is not damaging or imposing other modules to produce unexpected results. It falls under the class of black box testing

REQUIREMENT ANALYSIS

Requirement analysis, also called requirement engineering, is the process of determining user expectations for a new modified product. It encompasses the tasks that determine the need for analyzing, documenting, validating and managing software or system requirements. The requirements should be documentable, actionable, measurable, testable and traceable related to identified business needs or opportunities and define to a level of detail, sufficient for system design.

FUNCTIONAL REQUIREMENTS

It is a technical specification requirement for the software products. It is the first step in the requirement analysis process which lists the requirements of particular software systems including functional, performance and security requirements. The function of the system depends mainly on the quality hardware used to run the software with given functionality.

Usability

It specifies how easy the system must be use. It is easy to ask queries in any format which is short or long, porter stemming algorithm stimulates the desired response for user.

Robustness

It refers to a program that performs well not only under ordinary conditions but also under unusual conditions. It is the ability of the user to cope with errors for irrelevant queries during execution.

Security

The state of providing protected access to resource is security. The system provides good security and unauthorized users cannot access the system there by providing high security.

Reliability

It is the probability of how often the software fails. The measurement is often expressed in MTBF (Mean Time Between Failures). The requirement is needed in order to ensure that the processes work correctly and completely without being aborted. It can handle any load and survive and survive and even capable of working around any failure.

Compatibility

It is supported by version above all web browsers. Using any web servers like localhost makes the system real-time experience.

Flexibility

The flexibility of the project is provided in such a way that is has the ability to run on different environments being executed by different users.

Safety

Safety is a measure taken to prevent trouble. Every query is processed in a secured manner without letting others to know one's personal information.

NON- FUNCTIONAL REQUIREMENTS

Portability

It is the usability of the same software in different environments. The project can be run in any operating system.

Performance

These requirements determine the resources required, time interval, throughput and everything that deals with the performance of the system.

Accuracy

The result of the requesting query is very accurate and high speed of retrieving information. The degree of security provided by the system is high and effective.

Maintainability

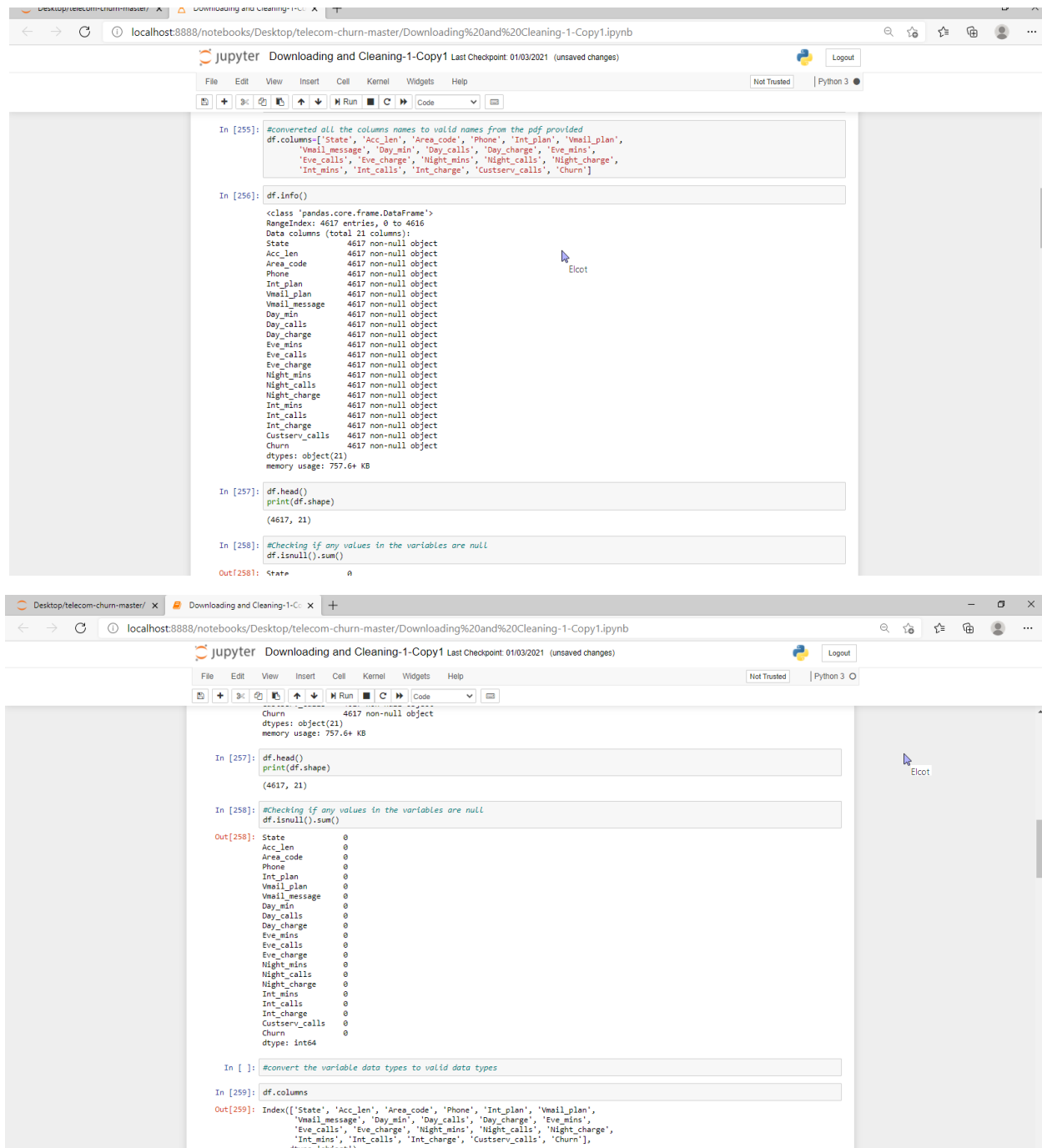
Project is simple as further updates can be easily done without affecting its stability. Maintainability basically defines that how easy it is to maintain the system. It means that how easy it is to maintain the system, analyse, change and test the application. Maintainability of this project is simple as further updates can be easily done without affecting its stability.

CHAPTER 9

RESULT AND DISCUSSION

OUTPUT SCREENSHOTS

Downloading and Cleaning the Dataset



```
In [255]: #converted all the columns names to valid names from the pdf provided
df.columns=['State', 'Acc_len', 'Area_code', 'Phone', 'Int_plan', 'Vmail_plan',
            'Vmail_message', 'Day_min', 'Day_calls', 'Day_charge', 'Eve_mins',
            'Eve_calls', 'Eve_charge', 'Night_mins', 'Night_calls', 'Night_charge',
            'Int_mins', 'Int_calls', 'Int_charge', 'Custserv_calls', 'Churn']

In [256]: df.info()

<class 'pandas.core.frame.DataFrame'>
RangeIndex: 4617 entries, 0 to 4616
Data columns (total 21 columns):
 State          4617 non-null object
 Acc_len        4617 non-null object
 Area_code      4617 non-null object
 Phone          4617 non-null object
 Int_plan       4617 non-null object
 Vmail_plan     4617 non-null object
 Vmail_message  4617 non-null object
 Day_min        4617 non-null object
 Day_calls      4617 non-null object
 Day_charge     4617 non-null object
 Eve_mins       4617 non-null object
 Eve_calls      4617 non-null object
 Eve_charge     4617 non-null object
 Night_mins     4617 non-null object
 Night_calls    4617 non-null object
 Night_charge   4617 non-null object
 Int_mins       4617 non-null object
 Int_calls      4617 non-null object
 Int_charge     4617 non-null object
 Custserv_calls 4617 non-null object
 Churn          4617 non-null object
dtypes: object(21)
memory usage: 757.6+ KB

In [257]: df.head()
print(df.shape)

(4617, 21)

In [258]: #Checking if any values in the variables are null
df.isnull().sum()

Out[258]: State          0
Acc_len          0
Area_code        0
Phone            0
Int_plan         0
Vmail_plan       0
Vmail_message    0
Day_min          0
Day_calls        0
Day_charge       0
Eve_mins         0
Eve_calls        0
Eve_charge       0
Night_mins       0
Night_calls      0
Night_charge     0
Int_mins         0
Int_calls        0
Int_charge       0
Custserv_calls   0
Churn            0
dtype: int64

In [ ]: #convert the variable data types to valid data types

In [259]: df.columns

Out[259]: Index(['State', 'Acc_len', 'Area_code', 'Phone', 'Int_plan', 'Vmail_plan',
                'Vmail_message', 'Day_min', 'Day_calls', 'Day_charge', 'Eve_mins',
                'Eve_calls', 'Eve_charge', 'Night_mins', 'Night_calls', 'Night_charge',
                'Int_mins', 'Int_calls', 'Int_charge', 'Custserv_calls', 'Churn'],
                dtype='object')
```

```
Desktop/telecom-churn-master/ x Downloading and Cleaning-1-Copy1 x +
localhost:8888/notebooks/Desktop/telecom-churn-master/Downloading%20and%20Cleaning-1-Copy1.ipynb
jupyter Downloading and Cleaning-1-Copy1 Last Checkpoint: 01/03/2021 (autosaved)
File Edit View Insert Cell Kernel Widgets Help Not Trusted Python 3
In [233]: df.info()
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 4617 entries, 0 to 4616
Data columns (total 21 columns):
State          4617 non-null object
Acc_len        4617 non-null int64
Area_code      4617 non-null int64
Phone          4617 non-null object
Int_plan       4617 non-null object
Vmail_plan     4617 non-null object
Vmail_message  4617 non-null int64
Day_mins       4617 non-null float64
Day_calls      4617 non-null int64
Day_charge     4617 non-null float64
Eve_mins       4617 non-null float64
Eve_calls      4617 non-null int64
Eve_charge     4617 non-null float64
Night_mins     4617 non-null float64
Night_calls    4617 non-null int64
Night_charge   4617 non-null float64
Int_mins       4617 non-null float64
Int_calls      4617 non-null int64
Int_charge     4617 non-null float64
Custserv_calls 4617 non-null int64
Churn          4617 non-null object
dtypes: float64(8), int64(8), object(5)
memory usage: 757.6+ KB

In [261]: #dropping phone field -not required since it is unique
df.drop(columns='Phone',inplace=True)
print(df.shape)
(4617, 20)

In [262]: #saving the file before proceeding further
df_bfre_enc= df.to_csv('df_bfre_enc.csv',index=False)

In [ ]:
```

```
Search results - gtscse.chennai/ x Downloads/telecom-churn-master/ x Model selection-4-Copy1 - Jupyter x EDA-2-Copy1 - Jupyter Notebo... x Downloading and Cleaning-1-Copy1 x +
localhost:8888/notebooks/Downloads/telecom-churn-master/telecom-churn-master/Downloading%20and%20Cleaning-1-Copy1.ipynb
jupyter Downloading and Cleaning-1-Copy1 Last Checkpoint: 02/21/2021 (autosaved)
File Edit View Insert Cell Kernel Widgets Help Not Trusted Python 3
In [265]: df.head()
df_after_enc=df.to_csv('df_after_enc.csv',index=False)

In [ ]: # a column like state needs to be done one hot encoding since the values in it are huge and each value is unique -Learn about one

In [266]: df.State.value_counts()
#plt.bar(df.State,height=0.8,width=0.6)
#plt.show()

Out[266]: WV 149
AL 114
NH 112
ID 110
OH 110
VA 109
NY 109
OR 107
WY 106
NJ 106
TX 105
UT 104
WI 99
MI 98
PA 97
KS 96
VT 95
ME 95
CT 92
MD 92
HT 92
NH 91
```

localhost:8888/notebooks/Downloads/telecom-churn-master/telecom-churn-master/Downloading%20and%20Cleaning-1-Copy1.ipynb

Jupyter Downloading and Cleaning-1-Copy1 Last Checkpoint: 02/21/2021 (autosaved)

File Edit View Insert Cell Kernel Widgets Help Not Trusted Python 3

```

NC 83
DC 82
AZ 81
ND 80
OK 80
HI 79
NE 79
IL 79
AR 77
GA 77
SD 76
LA 73
AK 69
PA 68
IA 65
CA 44
Name: State, dtype: int64

to reduce the reduce complexity using the pandas method to convert categorical variable into dummy/indicator variables with the get_dummies function

In [245]: df.State=pd.get_dummies(df.State,dtype='int64')

In [246]: df.State.value_counts()
Out[246]: 0    4548
          1      69
          Name: State, dtype: int64

In [248]: #df.Area_code.value_counts()

```

Exploratory Data Analysis

localhost:8888/notebooks/Downloads/telecom-churn-master/telecom-churn-master/EDA-20-2-Copy1.ipynb

Jupyter EDA -2-Copy1 (autosaved)

File Edit View Insert Cell Kernel Widgets Help Not Trusted Python 3

```

In [123]: import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
%matplotlib inline

In [ ]: #importing dataset after cleaning and encoding the categorical variables

In [124]: df=pd.read_csv('cleaned_dataset.csv')

In [114]: df.head()
Out[114]:
   State  Acc_len  Area_code  Int_plan  Vmail_plan  Vmail_message  Day_min  Day_calls  Day_charge  Eve_mins  Eve_calls  Eve_charge  Night_mins  Night_calls
0    0    128    415    0    1    25    265.1    110    45.07    197.4    99    16.78    244.7    8
1    0    107    415    0    1    26    161.6    123    27.47    195.5    103    16.62    254.4    10
2    0    137    415    0    0    0    243.4    114    41.38    121.2    110    10.30    162.6    10
3    0    84    408    1    0    0    299.4    71    50.90    61.9    88    5.26    196.9    8
4    0    75    415    1    0    0    166.7    113    28.34    148.3    122    12.61    186.9    12

In [ ]: #analysing data like
#GOAL : 2 classes Churners and non-Churners
#which features are the driving factors for churn

Do churners call customer service more?

```

Search results - gtsscse.chennai X Downloads/telecom-churn-master X EDA-2-Copy1 - Jupyter Notebo X +

localhost:8888/notebooks/Downloads/telecom-churn-master/telecom-churn-master/EDA%20-2-Copy1.ipynb

Jupyter EDA-2-Copy1 (autosaved) Logout

File Edit View Insert Cell Kernel Widgets Help Not Trusted Python 3

Do churners call customer service more?

```
In [118]: #taking a part of the dataframe and analysing
ff=df.loc[:,['Custserv_calls','Vmail_message','Int_plan','Churn']]
#using groupby()
#df.groupby(['Churn']).mean()
ff.head()
```

Out[118]:

	Custserv_calls	Vmail_message	Int_plan	Churn
0	1	25	0	0
1	1	26	0	0
2	0	0	0	0
3	2	0	1	0
4	3	0	1	0

```
In [119]: ff.groupby(['Churn']).std()
```

Out[119]:

	Custserv_calls	Vmail_message	Int_plan
Churn			
0	1.163901	13.848080	0.247683
1	1.822432	11.418575	0.451058

```
In [120]: ff.groupby(['Churn']).mean()
#the mean and std of customer services calls for the churners is high compared to non-churners
```

Search results - gtsscse.chennai X Downloads/telecom-churn-master X EDA-2-Copy1 - Jupyter Notebo X +

localhost:8888/notebooks/Downloads/telecom-churn-master/telecom-churn-master/EDA%20-2-Copy1.ipynb

Jupyter EDA-2-Copy1 (autosaved) Logout

File Edit View Insert Cell Kernel Widgets Help Not Trusted Python 3

```
In [120]: ff.groupby(['Churn']).mean()
#the mean and std of customer services calls for the churners is high compared to non-churners
```

Out[120]:

	Custserv_calls	Vmail_message	Int_plan
Churn			
0	1.455693	8.385761	0.065640
1	2.239329	4.614329	0.283537

```
In [29]: #Churning in some area_codes is more compared to others--
gg=pd.read_csv('df_after_enc.csv')
gg.groupby('Area_code')['Churn'].value_counts()
```

Out[29]:

Area_code	Churn	
408	0	996
	1	165
415	0	1981
	1	318
510	0	984
	1	173

Name: Churn, dtype: int64

```
In [ ]:
```

EDA with visualization

```
In [28]: df.head()
```

Out[28]:

State	Acc_len	Area_code	Int_plan	Vmail_plan	Vmail_message	Day_min	Day_calls	Day_charge	Eve_mins	Eve_calls	Eve_charge	Night_mins	Night_calls
-------	---------	-----------	----------	------------	---------------	---------	-----------	------------	----------	-----------	------------	------------	-------------

Search results - gtsscse.chennai... Downloads/telecom-churn-master... EDA -2-Copy1 - Jupyter Notebo... +

localhost:8888/notebooks/Downloads/telecom-churn-master/telecom-churn-master/EDA%20-2-Copy1.ipynb

Jupyter EDA -2-Copy1 (autosaved) Logout

File Edit View Insert Cell Kernel Widgets Help Not Trusted Python 3

In []:

EDA with visualization

In [28]: df.head()

Out[28]:

	State	Acc_len	Area_code	Int_plan	Vmail_plan	Vmail_message	Day_min	Day_calls	Day_charge	Eve_mins	Eve_calls	Eve_charge	Night_mins	Night_calls
0	0	128	415	0	1	25	265.1	110	45.07	197.4	99	16.78	244.7	9
1	0	107	415	0	1	26	161.6	123	27.47	195.5	103	16.62	254.4	10
2	0	137	415	0	0	0	243.4	114	41.38	121.2	110	10.30	162.6	10
3	0	84	408	1	0	0	299.4	71	50.90	61.9	88	5.26	196.9	8
4	0	75	415	1	0	0	166.7	113	28.34	148.3	122	12.61	186.9	12

In [30]: #checking the distribution of variables-normal distribution(-1.96 to +1.96)
from scipy import stats

In [49]: list=['Acc_len', 'Area_code', 'Vmail_message', 'Day_min', 'Day_calls', 'Day_charge', 'Eve_mins',
'Eve_calls', 'Eve_charge', 'Night_mins', 'Night_calls', 'Night_charge',
'Int_mins', 'Int_calls', 'Int_charge', 'Custserv_calls']
for i in list:
print('skewness and Kurtosis for '+ str(i)+' : '+str(stats.skew(df[i]))+' and '+ str(stats.kurtosis(df[i])))

skewness and Kurtosis for Acc_len: 0.10621316744599318 and -0.0937513108437864
skewness and Kurtosis for Area_code: 1.134964594925911 and -0.6865952826512816
skewness and Kurtosis for Vmail_message: 1.3263030509972922 and 0.12209308548663644
skewness and Kurtosis for Day_min: -0.002947284016695239 and -0.04365222612737929

Search results - gtsscse.chennai... Downloads/telecom-churn-master... EDA -2-Copy1 - Jupyter Notebo... +

localhost:8888/notebooks/Downloads/telecom-churn-master/telecom-churn-master/EDA%20-2-Copy1.ipynb

Jupyter EDA -2-Copy1 (autosaved) Logout

File Edit View Insert Cell Kernel Widgets Help Not Trusted Python 3

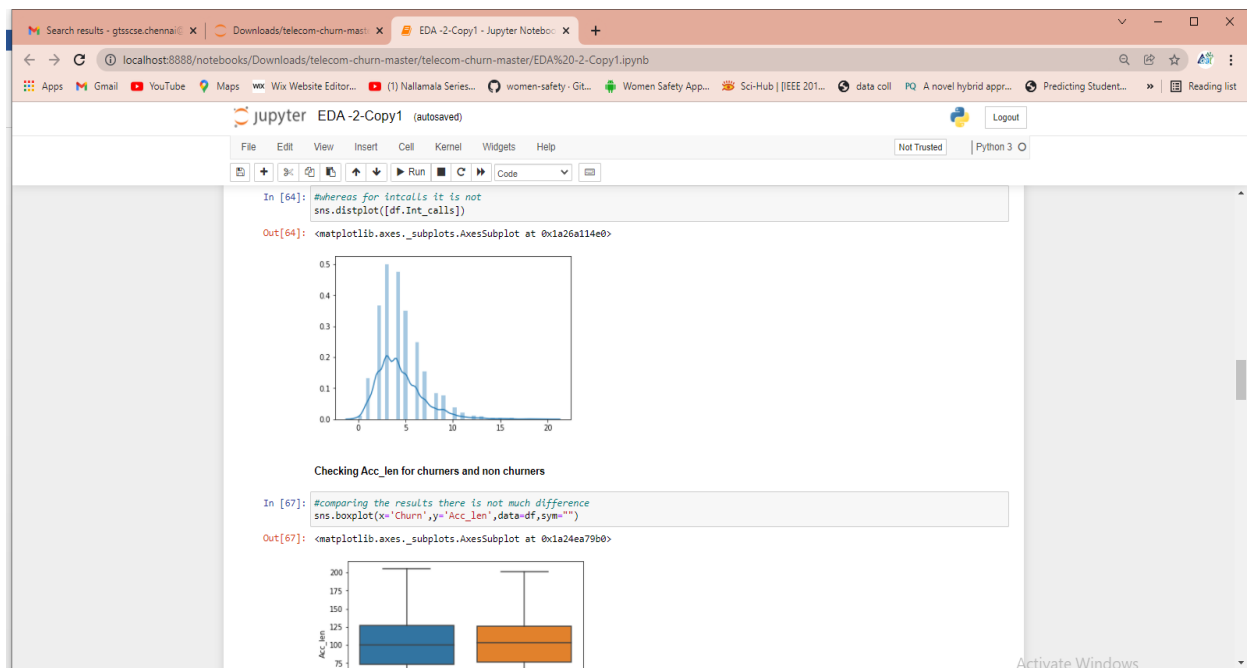
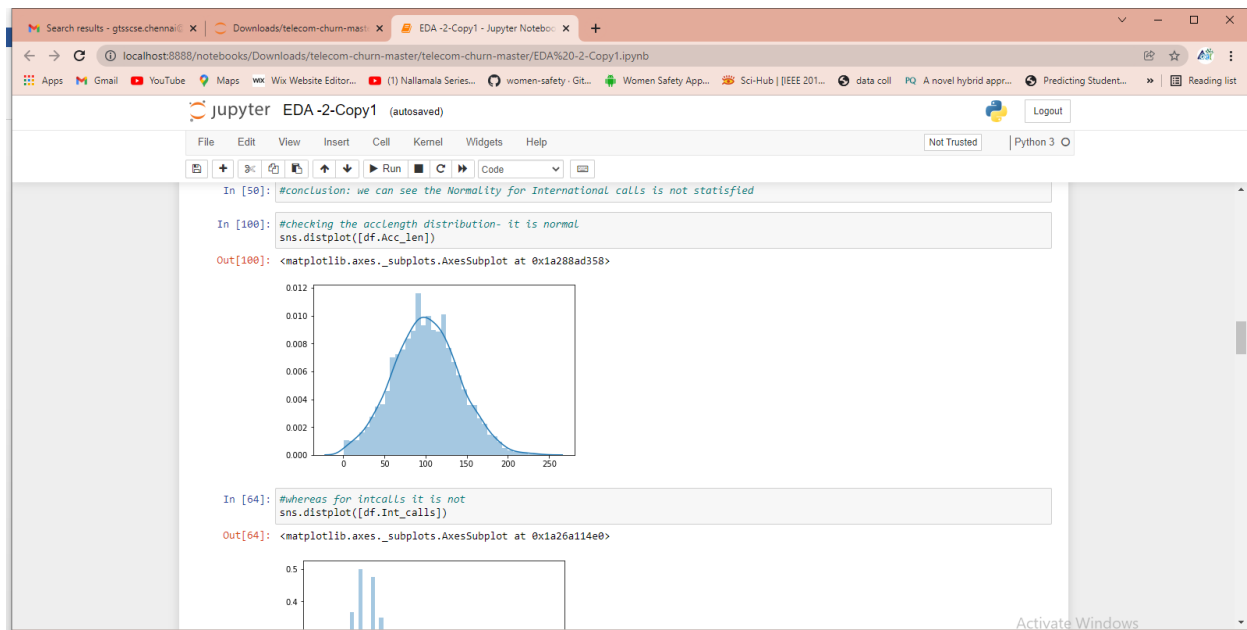
In [49]: list=['Acc_len', 'Area_code', 'Vmail_message', 'Day_min', 'Day_calls', 'Day_charge', 'Eve_mins',
'Eve_calls', 'Eve_charge', 'Night_mins', 'Night_calls', 'Night_charge',
'Int_mins', 'Int_calls', 'Int_charge', 'Custserv_calls']
for i in list:
print('skewness and Kurtosis for '+ str(i)+' : '+str(stats.skew(df[i]))+' and '+ str(stats.kurtosis(df[i])))

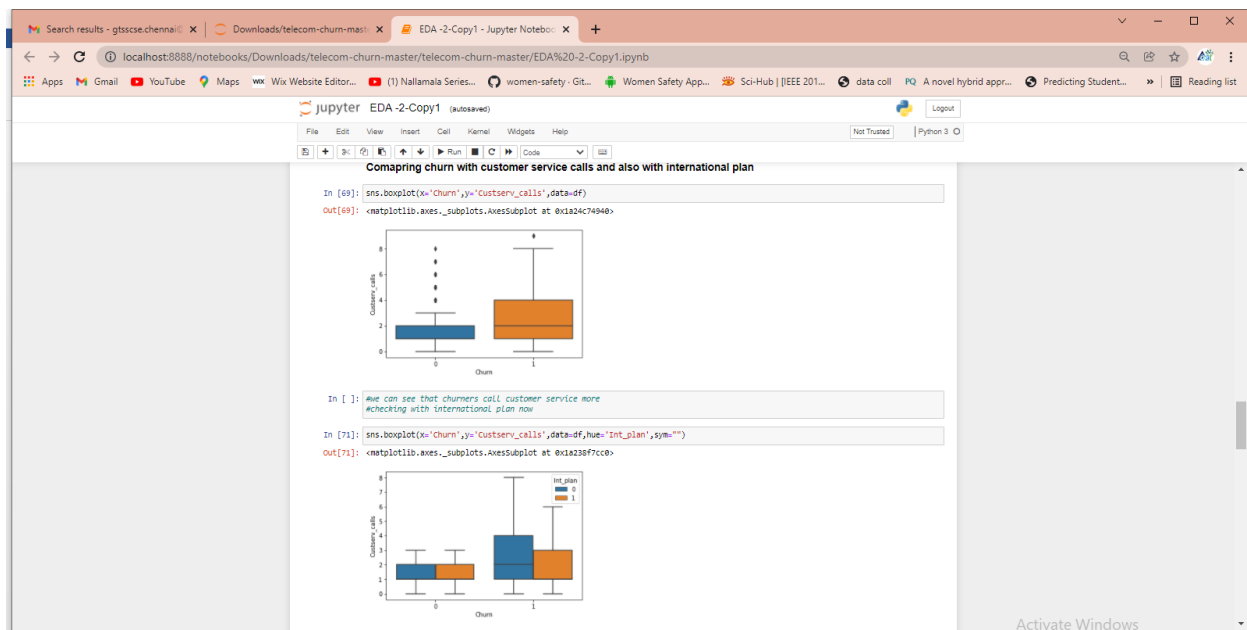
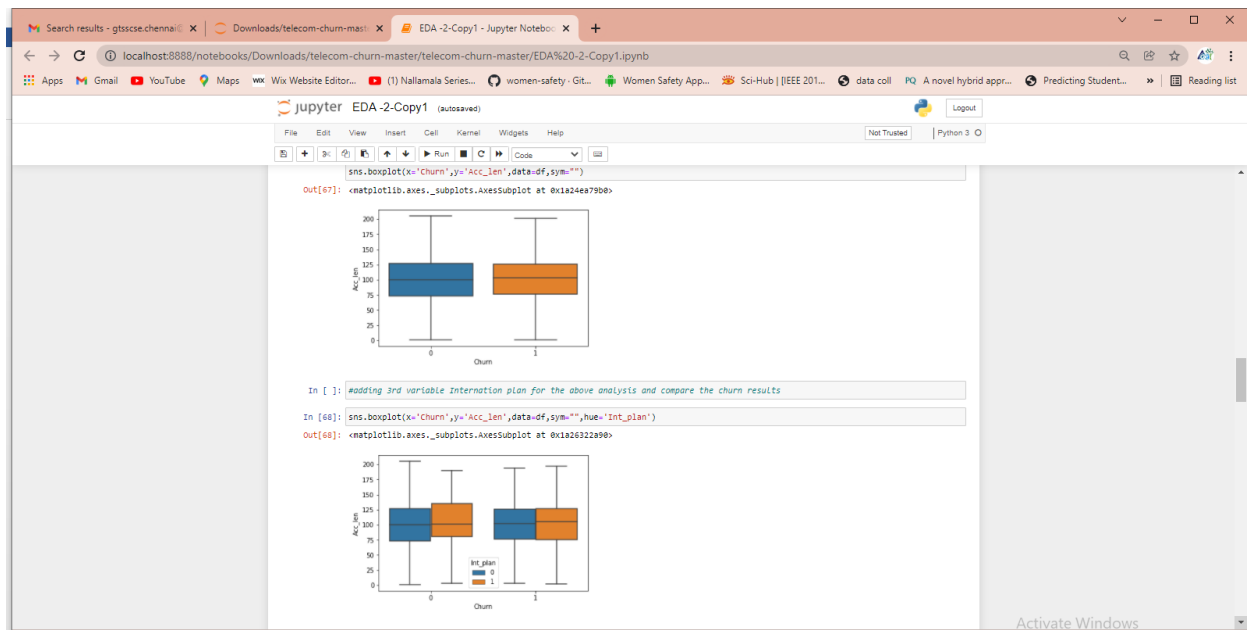
skewness and Kurtosis for Acc_len: 0.10621316744599318 and -0.0937513108437864
skewness and Kurtosis for Area_code: 1.134964594925911 and -0.6865952826512816
skewness and Kurtosis for Vmail_message: 1.3263030509972922 and 0.12209308548663644
skewness and Kurtosis for Day_min: -0.002947284016695239 and -0.04365222612737929
skewness and Kurtosis for Day_calls: -0.00898712687907403 and 0.19463514964618556
skewness and Kurtosis for Day_charge: -0.0029508087758798626 and -0.04351742102672107
skewness and Kurtosis for Eve_mins: -0.0052781500846280525 and 0.042283656150823745
skewness and Kurtosis for Eve_calls: -0.01754836130740179 and 0.13453389767071755
skewness and Kurtosis for Eve_charge: -0.005250110884106289 and 0.042175304917122514
skewness and Kurtosis for Night_mins: 0.02050841390755843 and 0.06804343141054003
skewness and Kurtosis for Night_calls: 0.030807626932619969 and 0.06744140412465867
skewness and Kurtosis for Night_charge: 0.020508316821170136 and 0.06801463464630463
skewness and Kurtosis for Int_mins: -0.22081808952003746 and 0.6695756138404381
skewness and Kurtosis for Int_calls: 1.3659756765088018 and 3.2980521818291813
skewness and Kurtosis for Int_charge: -0.22127033177212463 and 0.6704905004546904
skewness and Kurtosis for Custserv_calls: 1.04646081992374216 and 1.512086924300303

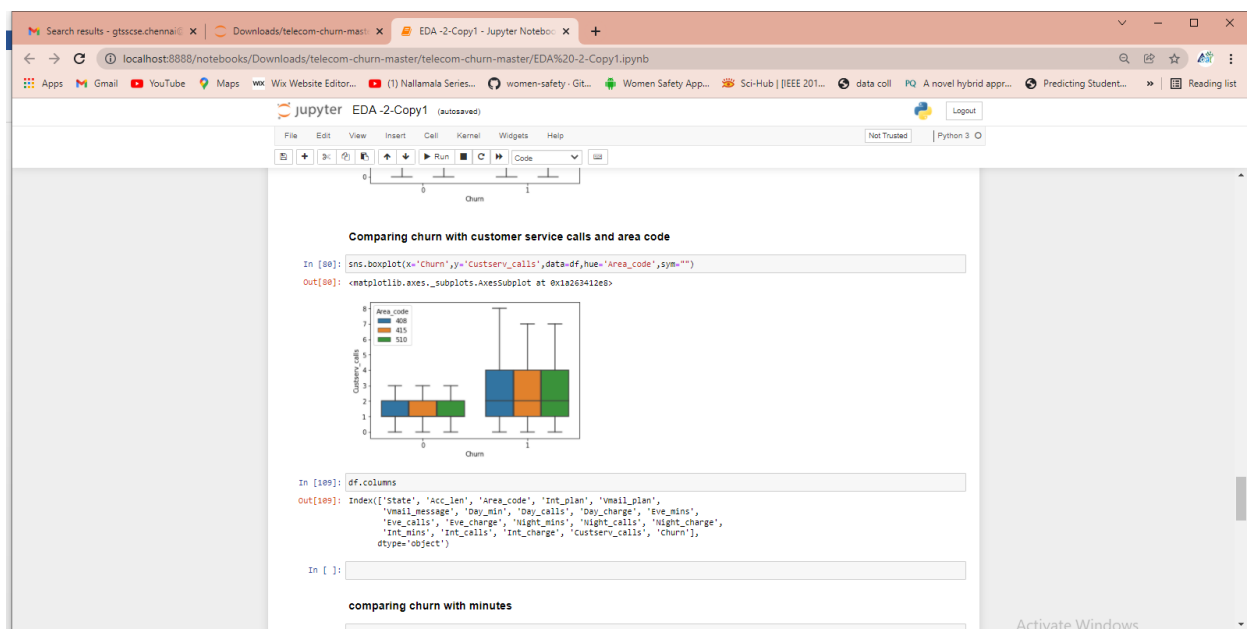
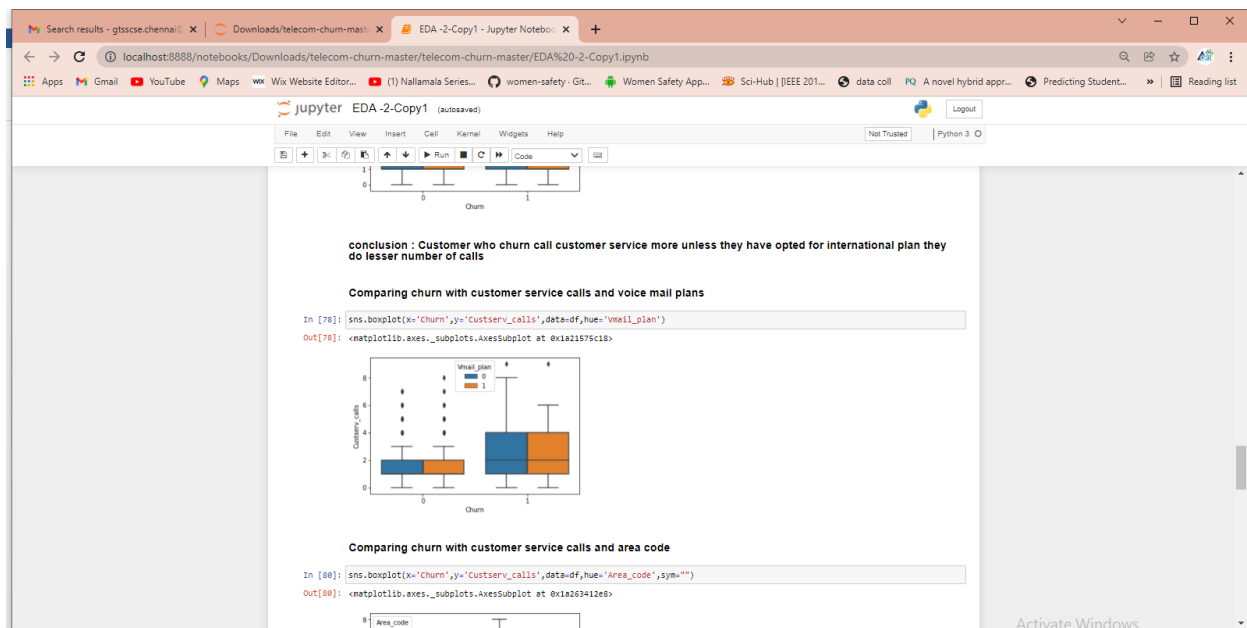
In [50]: #conclusion: we can see the Normality for International calls is not satisfied

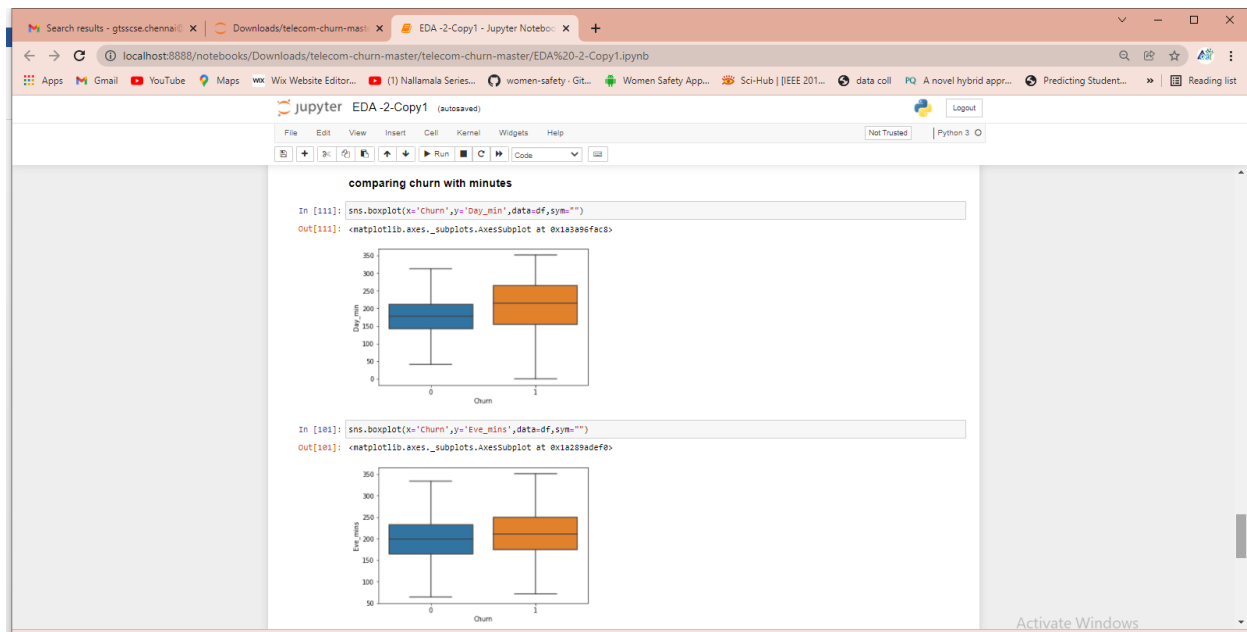
In [100]: #checking the accLength distribution- it is normal
sns.distplot([df.Acc_len])

Out[100]: <matplotlib.axes._subplots.AxesSubplot at 0x1a288ad358>

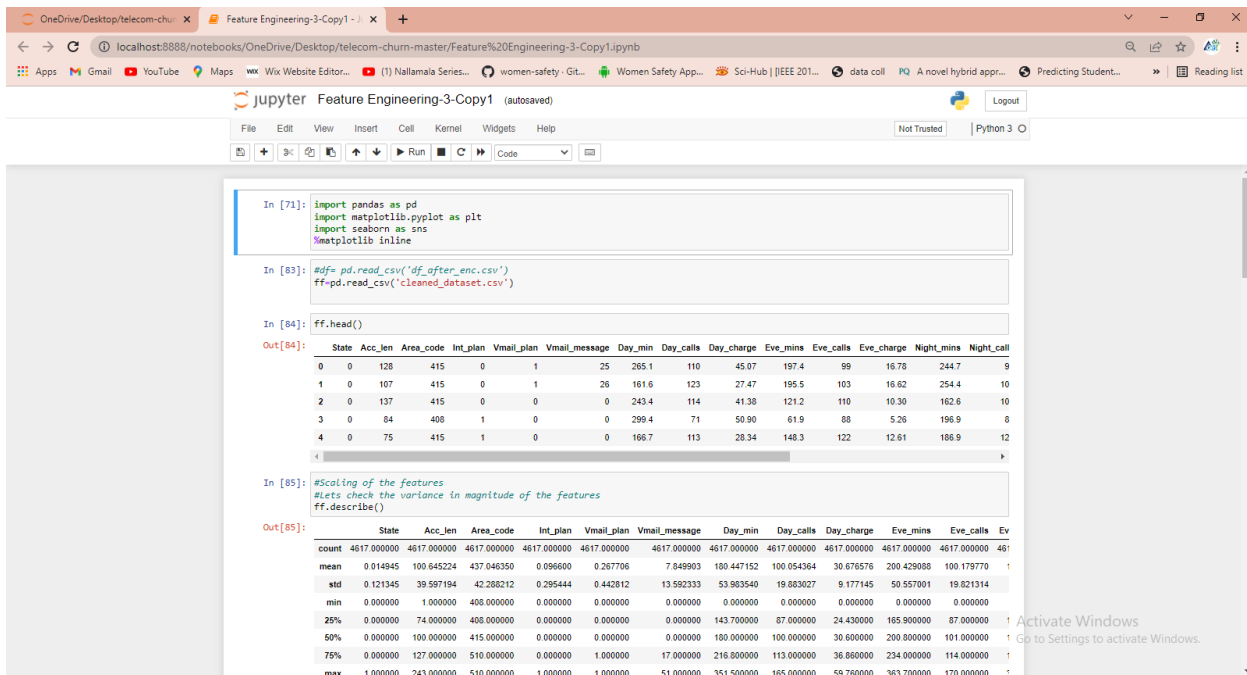








Feature Selection



OneDrive/Desktop/telecom-churn-master/Feature Engineering-3-Copy1 - X

localhost:8888/notebooks/OneDrive/Desktop/telecom-churn-master/Feature%20Engineering-3-Copy1.ipynb

Feature Engineering-3-Copy1 (autosaved)

File Edit View Insert Cell Kernel Widgets Help

Not Trusted Python 3

```

In [85]: #scaling of the features
#lets check the variance in magnitude of the features
ff.describe()

Out[85]:
```

	State	Acc_len	Area_code	Int_plan	Vmail_plan	Vmail_message	Day_min	Day_calls	Day_charge	Eve_mins	Eve_calls	Churn
count	4617.000000	4617.000000	4617.000000	4617.000000	4617.000000	4617.000000	4617.000000	4617.000000	4617.000000	4617.000000	4617.000000	4617.000000
mean	0.014945	100.645224	437.046350	0.096600	0.267706	7.849903	100.447152	100.054364	30.676576	200.429088	100.179770	0.014945
std	0.121345	39.597194	42.288212	0.295444	0.442812	13.592333	53.983540	19.883027	9.177145	50.557001	19.821314	0.121345
min	0.000000	1.000000	408.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
25%	0.000000	74.000000	408.000000	0.000000	0.000000	0.000000	143.700000	87.000000	24.430000	165.900000	87.000000	0.000000
50%	0.000000	100.000000	415.000000	0.000000	0.000000	0.000000	100.000000	100.000000	30.600000	200.800000	101.000000	0.000000
75%	0.000000	127.000000	510.000000	0.000000	1.000000	17.000000	216.800000	113.000000	36.860000	234.000000	114.000000	0.000000
max	1.000000	243.000000	510.000000	1.000000	1.000000	51.000000	351.500000	185.000000	59.750000	363.700000	170.000000	0.000000

```

In [68]: #to scale import standardizer
from sklearn.preprocessing import StandardScaler

In [86]: #scaling on the quantitative data
list=['Acc_len', 'Vmail_message', 'Day_min', 'Day_calls', 'Day_charge', 'Eve_mins',
      'Eve_calls', 'Eve_charge', 'Night_mins', 'Night_calls', 'Night_charge',
      'Int_mins', 'Int_calls', 'Int_charge', 'Custserv_calls']

#list=['State', 'Acc_len', 'Area_code', 'Int_plan', 'Vmail_plan', 'Vmail_message', 'Day_min', 'Day_calls', 'Day_charge', 'Eve_mins',
#      'Eve_calls', 'Eve_charge', 'Night_mins', 'Night_calls', 'Night_charge',
#      'Int_mins', 'Int_calls', 'Int_charge', 'Custserv_calls']

ff[list] = StandardScaler().fit_transform(ff[list])

/Users/muralija/anaconda3/lib/python3.7/site-packages/sklearn/preprocessing/data.py:645: DataConversionWarning: Data with 1
input dtype int64, float64 were all converted to float64 by StandardScaler.
return self.partial_fit(X, y)
/Users/muralija/anaconda3/lib/python3.7/site-packages/sklearn/base.py:464: DataConversionWarning: Data with input dtype int
64, float64 were all converted to float64 by StandardScaler.
return self.fit(X, **fit_params).transform(X)

In [87]: ff.info()

```

Activate Windows
Go to Settings to activate Windows.

OneDrive/Desktop/telecom-churn-master/Feature Engineering-3-Copy1 - X

localhost:8888/notebooks/OneDrive/Desktop/telecom-churn-master/Feature%20Engineering-3-Copy1.ipynb

Feature Engineering-3-Copy1 (autosaved)

File Edit View Insert Cell Kernel Widgets Help

Not Trusted Python 3

```

return self.fit(X, **fit_params).transform(X)

In [87]: ff.info()

<class 'pandas.core.frame.DataFrame'>
RangeIndex: 4617 entries, 0 to 4616
Data columns (total 20 columns):
State                4617 non-null int64
Acc_len              4617 non-null float64
Area_code            4617 non-null int64
Int_plan             4617 non-null int64
Vmail_plan           4617 non-null int64
Vmail_message        4617 non-null float64
Day_min              4617 non-null float64
Day_calls            4617 non-null float64
Day_charge           4617 non-null float64
Eve_mins             4617 non-null float64
Eve_calls            4617 non-null float64
Eve_charge           4617 non-null float64
Night_mins           4617 non-null float64
Night_calls          4617 non-null float64
Night_charge         4617 non-null float64
Int_mins             4617 non-null float64
Int_calls            4617 non-null float64
Int_charge           4617 non-null float64
Custserv_calls       4617 non-null float64
Churn                4617 non-null int64
dtypes: float64(15), int64(5)
memory usage: 721.5 KB

In [88]: ff.Churn=ff.Churn.astype('int64')

In [89]: ff.describe()

Out[89]:
```

	State	Acc_len	Area_code	Int_plan	Vmail_plan	Vmail_message	Day_min	Day_calls	Day_charge	Eve_mins	Eve_calls	Churn
count	4617.000000	4.617000e+03	4617.000000	4617.000000	4617.000000	4.617000e+03	4.617000e+03	4.617000e+03	4.617000e+03	4.617000e+03	4.617000e+03	4617.000000
mean	0.014945	-1.186450e-16	437.046350	0.096600	0.267706	-7.147557e-16	2.507560e-17	-5.183955e-17	-2.786980e-17	-2.332021e-16	-1.0717	0.014945
std	0.121345	1.000108e+00	42.288212	0.295444	0.442812	1.000108e+00	1.000108e+00	1.000108e+00	1.000108e+00	1.000108e+00	1.000108e+00	0.121345
min	0.000000	-2.516744e+00	408.000000	0.000000	0.000000	-5.775869e-01	-3.342995e+00	-5.032694e+00	-3.343076e+00	-3.964847e+00	-5.0546	0.000000

Activate Windows
Go to Settings to activate Windows.

```
OneDrive/Desktop/telecom-churn-master/Feature%20Engineering-3-Copy1.ipynb
localhost:8888/notebooks/OneDrive/Desktop/telecom-churn-master/Feature%20Engineering-3-Copy1.ipynb

jupyter Feature Engineering-3-Copy1 (autosaved)
File Edit View Insert Cell Kernel Widgets Help
Run Not Trusted Python 3

Churn 4617 non-null int64
dtypes: float64(15), int64(5)
memory usage: 721.5 KB

In [88]: ff.Churn=ff.Churn.astype('int64')

In [89]: ff.describe()

Out[89]:
```

	State	Acc_len	Area_code	Int_plan	Vmail_plan	Vmail_message	Day_min	Day_calls	Day_charge	Eve_mins	Eve_calls
count	4617.000000	4.617000e+03	4617.000000	4617.000000	4617.000000	4.617000e+03	4.617000e+03	4.617000e+03	4.617000e+03	4.617000e+03	4.617000e+03
mean	0.014945	-1.186450e-16	437.046350	0.096600	0.267706	-7.147557e-16	2.507560e-16	-5.183956e-17	-2.786980e-17	-2.332021e-16	-1.0717
std	0.121345	1.000108e+00	42.288212	0.295444	0.442812	1.000108e+00	1.000108e+00	1.000108e+00	1.000108e+00	1.000108e+00	1.0001
min	0.000000	-2.516744e+00	408.000000	0.000000	0.000000	-5.775069e-01	-3.342995e+00	-5.032694e+00	-3.343076e+00	-3.964847e+00	-5.05481
25%	0.000000	-6.729798e-01	408.000000	0.000000	0.000000	-5.775069e-01	-6.807840e-01	-6.566293e-01	-6.807402e-01	-6.830474e-01	-6.6500
50%	0.000000	-1.629646e-02	415.000000	0.000000	0.000000	-5.775069e-01	-8.284011e-03	-2.734503e-03	-0.345078e-03	7.337303e-03	4.1385
75%	0.000000	6.656439e-01	510.000000	0.000000	1.000000	6.732537e-01	6.734791e-01	6.511603e-01	6.735581e-01	6.640930e-01	6.9731
max	1.000000	3.585462e+00	510.000000	1.000000	1.000000	3.174935e+00	3.168954e+00	3.266739e+00	3.169450e+00	3.228792e+00	3.5228

```


In [97]: #FINDING THE CORRELATION between the variables to drop the redundant columns
data_cor= ff.corr()

In [99]: data_cor.Churn.sort_values()

Out[99]:
```

	Vmail_plan	Vmail_message	Int_calls	State	Night_calls	Eve_calls	Area_code	Day_calls	Acc_len	Night_charge	Night_mins	Int_charge
	-0.118149	-0.096884	-0.045844	-0.024561	-0.006671	0.001943	0.011995	0.016268	0.018846	0.043260	0.043262	0.064728
	0.001943	0.011995	0.016268	0.018846	0.043260	0.043262	0.064728	0.064731	0.064731	0.064731	0.064731	0.064731

```
OneDrive/Desktop/telecom-churn-master/Feature%20Engineering-3-Copy1.ipynb
localhost:8888/notebooks/OneDrive/Desktop/telecom-churn-master/Feature%20Engineering-3-Copy1.ipynb

jupyter Feature Engineering-3-Copy1 (autosaved)
File Edit View Insert Cell Kernel Widgets Help
Run Not Trusted Python 3

Night_charge 0.043260
Night_mins 0.043262
Int_charge 0.064728
Int_mins 0.064731
Eve_charge 0.087036
Eve_mins 0.087042
Day_charge 0.262434
Day_min 0.262439
Custserv_calls 0.209351
Int_plan 0.257524
Churn 1.000000
Name: Churn, dtype: float64

In [ ]:

In [ ]: #saving file after only scaling
df_after_scale = ff.to_csv('df_after_scale.csv',index=False)

conclusion we can see that day mins-day charge / eve_mins-eve charge / night mins -night charge/ Int mins and Int charge are highly correlated

In [ ]: # we can add new features as well - TBD LATER

In [92]: df = ff.drop(['Day_charge','Eve_charge','Night_charge','Int_charge'],axis=1)

In [93]: df.head()

Out[93]:
```

	State	Acc_len	Area_code	Int_plan	Vmail_plan	Vmail_message	Day_min	Day_calls	Eve_mins	Eve_calls	Night_mins	Night_calls	Int_mins	Int_calls
0	0	0.690901	415	0	1	1.261885	1.568293	0.500261	-0.059821	-0.059827	0.872135	-0.448712	-0.101301	-0.583487
1	0	0.160503	415	0	1	1.335463	-0.349166	1.154156	-0.097506	0.142298	1.064069	0.153308	1.240707	-0.583487
2	0	0.918214	415	0	0	-0.577587	1.166275	0.701460	-1.567294	0.495402	-0.752381	0.203477	0.696649	0.230398
3	0	-0.420409	408	1	0	-0.577587	2.203741	-1.461423	-2.740354	-0.614545	-0.073686	-0.549048	-1.334498	1.044284
4	0	-0.647723	415	1	0	-0.577587	-0.254882	0.651180	-1.031207	1.000966	-0.271556	1.056338	-0.065031	-0.583487

```


In [94]: df.info()
```

OneDrive/Desktop/telecom-churn-master/Feature Engineering-3-Copy1 - Jupyter

localhost:8888/notebooks/OneDrive/Desktop/telecom-churn-master/Feature%20Engineering-3-Copy1.ipynb

Jupyter Feature Engineering-3-Copy1 (autosaved)

File Edit View Insert Cell Kernel Widgets Help

Not Trusted Python 3

```

2 0 0.918214 415 0 0 -0.577587 1.166275 0.014609 -1.567294 0.495492 -0.752381 0.203477 0.998849 0.230390
3 0 -0.420409 408 1 0 -0.577587 2.203741 -1.461423 -2.740354 -0.614545 -0.073686 -0.549048 -1.334486 1.044284
4 0 -0.647723 415 1 0 -0.577587 -0.254682 0.651160 -1.031207 1.100966 -0.271556 1.056338 -0.065031 -0.583487

```

```

In [94]: df.info()
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 4617 entries, 0 to 4616
Data columns (total 16 columns):
State                4617 non-null int64
Acc_len              4617 non-null float64
Area_code            4617 non-null int64
Int_plan             4617 non-null int64
Vmail_plan           4617 non-null int64
Vmail_message        4617 non-null float64
Day_min              4617 non-null float64
Day_calls             4617 non-null float64
Eve_mins              4617 non-null float64
Eve_calls             4617 non-null float64
Night_mins           4617 non-null float64
Night_calls          4617 non-null float64
Int_mins              4617 non-null float64
Int_calls             4617 non-null float64
Custserv_calls       4617 non-null float64
Churn                 4617 non-null int64
dtypes: float64(11), int64(5)
memory usage: 577.2 KB

In [95]: df.to_csv('df_scale_corr.csv', index=False)

In [ ]: # saved file after scaling and dropping highly correlated variables.

```

Activate Windows
Go to Settings to activate Windows.

Model selection

OneDrive/Desktop/telecom-churn-master/Model selection-4-Copy1 - Jupyter

localhost:8888/notebooks/OneDrive/Desktop/telecom-churn-master/Model%20selection-4-Copy1.ipynb

Jupyter Model selection-4-Copy1 Last checkpoint: 02/21/2021 (autosaved)

File Edit View Insert Cell Kernel Widgets Help

Not Trusted Python 3

```

In [240]: import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
%matplotlib inline

since the output variable is Goal- churn or no churn TRIAL AND ERROR between logistic regression ,SVM ,Randomforest

In [241]: from sklearn.linear_model import LogisticRegression
from sklearn.svm import SVC
from sklearn.ensemble import RandomForestClassifier
from sklearn.model_selection import train_test_split
from collections import Counter
from imblearn.over_sampling import SMOTE
from sklearn.metrics import accuracy_score, classification_report, confusion_matrix, recall_score, precision_score

In [242]: df = pd.read_csv('df_after_scale.csv')

In [243]: X = df.drop('Churn', axis=1)

In [192]: X.head()

Out[192]:
   State  Acc_len  Area_code  Int_plan  Vmail_plan  Vmail_message  Day_min  Day_calls  Day_charge  Eve_mins  Eve_calls  Eve_charge  Night_mins  Night_c
0  0  0.690901    415      0      1      1.261885  1.568293  0.500261  1.568569 -0.059921 -0.059527 -0.059742  0.872135 -0.448
1  0  0.160503    415      0      1      1.335463 -0.349166  1.154156 -0.349447 -0.097506  0.142298 -0.096878  1.064069  0.153
2  0  0.918214    415      0      0     -0.577587  1.166275  0.701460  1.166439 -1.567294  0.495492 -1.567818 -0.752381  0.203
3  0 -0.420409    408      1      0     -0.577587  2.203741 -1.461423  2.203911 -2.740354 -0.614545 -2.740765 -0.073686 -0.549
4  0 -0.647723    415      1      0     -0.577587 -0.254682  0.651160 -0.254636 -1.031207  1.100966 -1.030217 -0.271556  1.056

```

```

In [193]: y = df.Churn

In [194]: X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.3, random_state=10)

```

Activate Windows
Go to Settings to activate Windows.

```
2 0 0.918214 415 0 0 -0.577587 1.166275 0.701460 1.166439 -1.567294 0.495492 -1.567818 -0.752381 0.203
3 0 -0.420409 408 1 0 -0.577587 2.203741 -1.461423 2.203911 -2.740354 -0.614545 -2.740765 -0.073886 -0.549
4 0 -0.647723 415 1 0 -0.577587 -0.254682 0.651160 -0.254636 -1.031207 1.100966 -1.030217 -0.271556 1.056

In [193]: y=df.Churn

In [194]: X_train,X_test,y_train,y_test = train_test_split(X,y,test_size=0.3,random_state=10)

In [195]: print(X_train.shape)
print(X_test.shape)
(3231, 19)
(1366, 19)

In [196]: print(Counter(y_train))
print(Counter(y_test))
Counter({0: 2761, 1: 470})
Counter({0: 1200, 1: 186})

In [ ]:

In [ ]: ### SHOTING REQUIRED -since the ratio of churners and non-churners is not equal

In [197]: sm = SMOTE()

In [198]: X_train_sm,y_train_sm = sm.fit_sample(X_train,y_train)

In [199]: # now the churners and non churners ratio is same in the output
print(Counter(y_train_sm))
print(Counter(y_test))

Counter({0: 2761, 1: 2761})
Counter({0: 1200, 1: 186})
```

Logistic Regression

```
Logistic Regression

In [ ]:

In [200]: clf=LogisticRegression()
clf.fit(X_train_sm,y_train_sm)

/Users/muralisajula/anaconda3/lib/python3.7/site-packages/sklearn/linear_model/logistic.py:433: FutureWarning: Default solver w
ill be changed to 'lbfgs' in 0.22. Specify a solver to silence this warning.
FutureWarning)

Out[200]: LogisticRegression(C=1.0, class_weight=None, dual=False, fit_intercept=True,
intercept_scaling=1, max_iter=100, multi_class='warn',
n_jobs=None, penalty='l2', random_state=None, solver='warn',
tol=0.0001, verbose=0, warm_start=False)

In [201]: y_pred=clf.predict(X_test)

In [ ]: #checking accuracy score

In [202]: accuracy_score(y_test,y_pred)

Out[202]: 0.7741702741702742

In [203]: recall_score(y_test,y_pred)

Out[203]: 0.7795698924731183

In [204]: precision_score(y_test,y_pred)

Out[204]: 0.34772182254196643

In [205]: confusion_matrix(y_test,y_pred)

Out[205]: array([[1928, 272],
[ 41, 145]])

In [206]: cm=confusion_matrix(y_test,y_pred)
```

OneDrive/Desktop/telecom-churn x Model selection-4-Copy1 - Jupyter x +

localhost:8888/notebooks/OneDrive/Desktop/telecom-churn-master/Model%20selection-4-Copy1.ipynb

jupyter Model selection-4-Copy1 Last Checkpoint: 02/21/2021 (autosaved)

File Edit View Insert Cell Kernel Widgets Help

Not Trusted Python 3

```
In [202]: accuracy_score(y_test,y_pred)
Out[202]: 0.7741702741702742

In [203]: recall_score(y_test,y_pred)
Out[203]: 0.7795698924731183

In [204]: precision_score(y_test,y_pred)
Out[204]: 0.34772182254196643

In [205]: confusion_matrix(y_test,y_pred)
Out[205]: array([[1928, 272],
               [ 41, 145]])

In [207]: pd.crosstab(y_test,y_pred)
Out[207]:
```

	0	1
Churn		
0	928	272
1	41	145

```
In [206]: print(classification_report(y_test,y_pred))
```

	precision	recall	f1-score	support
0	0.96	0.77	0.86	1200
1	0.35	0.78	0.48	186
micro avg	0.77	0.77	0.77	1386
macro avg	0.65	0.78	0.67	1386
weighted avg	0.88	0.77	0.81	1386

For the LR though the recall is 78% the precision score is very less

Activate Windows
Go to Settings to activate Windows.

Support Vector Machine

OneDrive/Desktop/telecom-churn x Model selection-4-Copy1 - Jupyter x +

localhost:8888/notebooks/OneDrive/Desktop/telecom-churn-master/Model%20selection-4-Copy1.ipynb

jupyter Model selection-4-Copy1 Last Checkpoint: 02/21/2021 (autosaved)

File Edit View Insert Cell Kernel Widgets Help

Not Trusted Python 3

Support vector Machine

```
In [208]: svc= SVC()

In [209]: svc.fit(X_train_sm,y_train_sm)
/Users/muralijajula/anaconda3/lib/python3.7/site-packages/sklearn/svm/base.py:196: FutureWarning: The default value of gamma will change from 'auto' to 'scale' in version 0.22 to account better for unscaled features. Set gamma explicitly to 'auto' or 'scale' to avoid this warning.
  "avoid this warning.", FutureWarning)

Out[209]: SVC(C=1.0, cache_size=200, class_weight=None, coef0=0.0,
              decision_function_shape='ovr', degree=3, gamma='auto_deprecated',
              kernel='rbf', max_iter=1, probability=False, random_state=None,
              shrinking=True, tol=0.001, verbose=False)

In [210]: y_pred_svc= svc.predict(X_test)

In [211]: accuracy_score(y_test,y_pred_svc)
Out[211]: 0.8787878787878788

In [212]: print(classification_report(y_test,y_pred_svc))
```

	precision	recall	f1-score	support
0	0.95	0.91	0.93	1200
1	0.54	0.71	0.61	186
micro avg	0.88	0.88	0.88	1386
macro avg	0.74	0.81	0.77	1386
weighted avg	0.90	0.88	0.89	1386

```
In [213]: confusion_matrix(y_test,y_pred_svc)
Out[213]: array([[1086, 114],
               [ 54, 132]])
```

Activate Windows
Go to Settings to activate Windows.

Model selection-4-Copy1 Last Checkpoint 02/21/2021 (autosaved)

File Edit View Insert Cell Kernel Widgets Help

Not Trusted Python 3

```

/Users/murati13guls/anaconda3/lib/python3.7/site-packages/sklearn/svm/base.py:196: FutureWarning: The default value of gamma will change from 'auto' to 'scale' in version 0.22 to account better for unscaled features. Set gamma explicitly to 'auto' or 'scale' to avoid this warning.
  "avoid this warning.", FutureWarning)

Out[209]: SVC(C=1.0, cache_size=200, class_weight=None, coef0=0.0,
            decision_function_shape='ovr', degree=3, gamma='auto_deprecated',
            kernel='rbf', max_iter=1, probability=False, random_state=None,
            shrinking=True, tol=0.001, verbose=False)

In [210]: y_pred_svc= svc.predict(X_test)

In [211]: accuracy_score(y_test,y_pred_svc)

Out[211]: 0.8787878787878788

In [212]: print(classification_report(y_test,y_pred_svc))

              precision    recall  f1-score   support

     0       0.95      0.91      0.93       1200
     1       0.54      0.71      0.61        186

   micro avg       0.88      0.88      0.88       1386
   macro avg       0.74      0.81      0.77       1386
  weighted avg       0.90      0.88      0.89       1386

In [213]: confusion_matrix(y_test,y_pred_svc)

Out[213]: array([[1086, 114],
                [ 54, 132]])

In [214]: pd.crosstab(y_test,y_pred)

Out[214]: col_0  0   1
          0  928 272
          1   41 145

```

Activate Windows
Go to Settings to activate Windows.

Random Forest

Model selection-4-Copy1 Last Checkpoint 02/21/2021 (autosaved)

File Edit View Insert Cell Kernel Widgets Help

Not Trusted Python 3

Random forest

```

In [233]: model = RandomForestClassifier(random_state=5,n_estimators=100,max_depth=15)

In [234]: model.fit(X_train_sm,y_train_sm)

Out[234]: RandomForestClassifier(bootstrap=True, class_weight=None, criterion='gini',
            max_depth=15, max_features='auto', max_leaf_nodes=None,
            min_impurity_decrease=0.0, min_impurity_split=None,
            min_samples_leaf=1, min_samples_split=2,
            min_weight_fraction_leaf=0.0, n_estimators=100, n_jobs=None,
            oob_score=False, random_state=5, verbose=0, warm_start=False)

In [239]: model.feature_importances_

Out[239]: array([0.00067071, 0.0212972 , 0.00580704, 0.17522528, 0.03849519,
            0.02408966, 0.11243647, 0.0219473 , 0.1229068 , 0.04745361,
            0.02094816, 0.04968137, 0.03437586, 0.02169135, 0.03184385,
            0.02962555, 0.04814574, 0.0277985 , 0.16564815])

In [235]: y_pred_rf= model.predict(X_test)

In [236]: print(classification_report(y_test,y_pred_rf))

              precision    recall  f1-score   support

     0       0.97      0.97      0.97       1200
     1       0.81      0.81      0.81        186

   micro avg       0.95      0.95      0.95       1386
   macro avg       0.89      0.89      0.89       1386
  weighted avg       0.95      0.95      0.95       1386

In [237]: accuracy_score(y_test,y_pred_rf)

Out[237]: 0.948051948051948

In [238]: confusion_matrix(y_test,y_pred_rf)

```

Activate Windows
Go to Settings to activate Windows.

```
OneDrive/Desktop/telecom-churn x Model selection-4-Copy1 - Jupyter x +
localhost:8888/notebooks/OneDrive/Desktop/telecom-churn-master/Model%20selection-4-Copy1.ipynb
Apps Gmail YouTube Maps Wix Website Editor... (1) Nallamala Series... women-safety - Git... Women Safety App... Sci-Hub | IEEE 201... data coll PQ A novel hybrid appr... Predicting Student... Reading list

jupyter Model selection-4-Copy1 Last Checkpoint: 02/21/2021 (autosaved)
File Edit View Insert Cell Kernel Widgets Help Not Trusted Python 3 O

max_depth=15, max_features='auto', max_leaf_nodes=None,
min_impurity_decrease=0.0, min_impurity_split=None,
min_samples_leaf=1, min_samples_split=2,
min_weight_fraction_leaf=0.0, n_estimators=100, n_jobs=None,
oob_score=False, random_state=5, verbose=0, warm_start=False)

In [239]: model.feature_importances_
Out[239]: array([0.00067071, 0.0212972 , 0.00580704, 0.17522526, 0.03849519,
0.02408966, 0.11243647, 0.0219473 , 0.1229068 , 0.04745381,
0.02094816, 0.04060127, 0.03437586, 0.02169135, 0.03184385,
0.02962555, 0.04814574, 0.0277985 , 0.16564815])

In [239]: y_pred_rf= model.predict(X_test)

In [236]: print(classification_report(y_test,y_pred_rf))
precision recall f1-score support
0 0.97 0.97 0.97 1200
1 0.81 0.81 0.81 186
micro avg 0.95 0.95 0.95 1386
macro avg 0.89 0.89 0.89 1386
weighted avg 0.95 0.95 0.95 1386

In [237]: accuracy_score(y_test,y_pred_rf)
Out[237]: 0.948051948051948

In [238]: confusion_matrix(y_test,y_pred_rf)
Out[238]: array([[1164, 36],
[ 36, 150]])

Random forest gives best results for all the metrics as compared to LR and SVM

In [ ]:
```

Random Forest Analysis

```
OneDrive/Desktop/telecom-churn x Random Forest optimized Model-5-Copy1 x +
localhost:8888/notebooks/OneDrive/Desktop/telecom-churn-master/Random%20Forest%20optimized%20Model-5-Copy1.ipynb
Apps Gmail YouTube Maps Wix Website Editor... (1) Nallamala Series... women-safety - Git... Women Safety App... Sci-Hub | IEEE 201... data coll PQ A novel hybrid appr... Predicting Student... Reading list

jupyter Random Forest optimized Model-5-Copy1 (autosaved)
File Edit View Insert Cell Kernel Widgets Help Not Trusted Python 3 O

In [2]: import pandas as pd
import numpy as np
from sklearn.linear_model import LogisticRegression
from sklearn.ensemble import RandomForestClassifier
from imblearn.over_sampling import SMOTE
from collections import Counter
from sklearn.model_selection import GridSearchCV, RandomizedSearchCV
from sklearn.metrics import classification_report, confusion_matrix, accuracy_score, recall_score, precision_score, roc_curve, roc_auc
from sklearn.model_selection import train_test_split
import matplotlib.pyplot as plt

In [44]: df= pd.read_csv('cleaned_dataset.csv')

In [45]: df.head()
Out[45]:
State Acc_len Area_code Int_plan Vmail_plan Vmail_message Day_min Day_calls Day_charge Eve_mins Eve_calls Eve_charge Night_mins Night_call
0 0 128 415 0 1 25 265.1 110 45.07 197.4 99 16.78 244.7 9
1 0 107 415 0 1 26 161.6 123 27.47 195.5 103 16.62 254.4 10
2 0 137 415 0 0 0 243.4 114 41.38 121.2 110 10.30 162.6 10
3 0 84 408 1 0 0 299.4 71 50.90 61.9 88 5.26 196.9 8
4 0 75 415 1 0 0 166.7 113 28.34 148.3 122 12.61 186.9 12

In [46]: df.columns
Out[46]: Index(['State', 'Acc_len', 'Area_code', 'Int_plan', 'Vmail_plan',
'Vmail_message', 'Day_min', 'Day_calls', 'Day_charge', 'Eve_mins',
'Eve_calls', 'Eve_charge', 'Night_mins', 'Night_calls', 'Night_charge',
'Int_mins', 'Int_calls', 'Int_charge', 'Custserv_calls', 'Churn'],
dtype='object')

In [47]: df = df.astype('float')
```

```
OneDrive/Desktop/telecom-churn-master Random Forest optimized Model-5-Copy1 (autosaved)
localhost:8888/notebooks/OneDrive/Desktop/telecom-churn-master/Random%20Forest%20optimized%20Model-5-Copy1.ipynb

In [46]: df.columns
Out[46]: Index(['State', 'Acc_len', 'Area_code', 'Int_plan', 'Vmail_plan',
              'Vmail_message', 'Day_min', 'Day_calls', 'Day_charge', 'Eve_mins',
              'Eve_calls', 'Eve_charge', 'Night_mins', 'Night_calls', 'Night_charge',
              'Int_mins', 'Int_calls', 'Int_charge', 'Custserv_calls', 'Churn'],
             dtype='object')

In [47]: df = df.astype('float')

In [48]: df.info()
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 4617 entries, 0 to 4616
Data columns (total 20 columns):
#   Column             Non-Null Count  Dtype
---  --
0   State              4617 non-null   float64
1   Acc_len            4617 non-null   float64
2   Area_code          4617 non-null   float64
3   Int_plan           4617 non-null   float64
4   Vmail_plan         4617 non-null   float64
5   Vmail_message      4617 non-null   float64
6   Day_min            4617 non-null   float64
7   Day_calls          4617 non-null   float64
8   Day_charge         4617 non-null   float64
9   Eve_mins           4617 non-null   float64
10  Eve_calls          4617 non-null   float64
11  Eve_charge         4617 non-null   float64
12  Night_mins         4617 non-null   float64
13  Night_calls        4617 non-null   float64
14  Night_charge       4617 non-null   float64
15  Int_mins           4617 non-null   float64
16  Int_calls          4617 non-null   float64
17  Int_charge         4617 non-null   float64
18  Custserv_calls     4617 non-null   float64
19  Churn              4617 non-null   float64
dtypes: float64(20)
memory usage: 721.5 KB

In [49]: X = df.drop('Churn',axis=1)
```

```
OneDrive/Desktop/telecom-churn-master Random Forest optimized Model-5-Copy1 (autosaved)
localhost:8888/notebooks/OneDrive/Desktop/telecom-churn-master/Random%20Forest%20optimized%20Model-5-Copy1.ipynb

memory usage: 721.5 KB

In [49]: X = df.drop('Churn',axis=1)
         y=df.Churn

In [50]: X_train,X_test,y_train,y_test = train_test_split(X,y,test_size=0.3,random_state=10)

In [51]: print(X_train.shape)
         print(X_test.shape)
         print(Counter(y_train))
         print(Counter(y_test))

(3231, 19)
(1386, 19)
Counter{(0.0: 2761, 1.0: 478)}
Counter{(0.0: 1200, 1.0: 186)}

In [52]: sm = SMOTE()
         X_train_sm,y_train_sm = sm.fit_sample(X_train,y_train)

In [53]: # now the chuners and non chuners ratio is same in the output
         print(Counter(y_train_sm))
         print(Counter(y_test))

Counter{(0.0: 2761, 1.0: 2761)}
Counter{(0.0: 1200, 1.0: 186)}

In [ ]:

In [54]: model = RandomForestClassifier(n_estimators=9,
         max_features='sqrt',
         max_depth=9,
         criterion='gini',
         bootstrap=False)

         model.fit(X_train_sm,y_train_sm)

Out[54]: RandomForestClassifier(bootstrap=False, max_depth=9, max_features='sqrt',
                                n_estimators=9)
```

```
OneDrive/Desktop/telecom-churn-master/Random Forest optimized Model-5-Copy1.ipynb
localhost:8888/notebooks/OneDrive/Desktop/telecom-churn-master/Random Forest optimized Model-5-Copy1.ipynb

jupyter Random Forest optimized Model-5-Copy1 (autosaved)
File Edit View Insert Cell Kernel Widgets Help Not Trusted Python 3

Counter((0.0: 2761, 1.0: 2761))
Counter((0.0: 1200, 1.0: 186))

In [ ]:

In [54]: model = RandomForestClassifier(n_estimators=9,
max_features='sqrt',
max_depth=9,
criterion='gini',
bootstrap=False)
model.fit(X_train_sm, y_train_sm)

Out[54]: RandomForestClassifier(bootstrap=False, max_depth=9, max_features='sqrt',
n_estimators=9)

In [55]: y_pred_rf = model.predict(X_test)

In [56]: print(classification_report(y_test, y_pred_rf))
precision    recall  f1-score   support

0.0   0.96   0.95   0.96   1200
1.0   0.72   0.77   0.74    186

accuracy          0.93   1386
macro avg         0.84   0.86   0.85   1386
weighted avg      0.93   0.93   0.93   1386

In [57]: # got a recall of 82% and precision of 80% at a particular trial

In [58]: accuracy_score(y_test, y_pred_rf)
Out[58]: 0.9285714285714286

In [59]: confusion_matrix(y_test, y_pred_rf)
Out[59]: array([[1144,  56],
[ 43, 143]])
```

```
OneDrive/Desktop/telecom-churn-master/Random Forest optimized Model-5-Copy1.ipynb
localhost:8888/notebooks/OneDrive/Desktop/telecom-churn-master/Random Forest optimized Model-5-Copy1.ipynb

jupyter Random Forest optimized Model-5-Copy1 (autosaved)
File Edit View Insert Cell Kernel Widgets Help Not Trusted Python 3

accuracy          0.93   1386
macro avg         0.84   0.86   0.85   1386
weighted avg      0.93   0.93   0.93   1386

In [57]: # got a recall of 82% and precision of 80% at a particular trial

In [58]: accuracy_score(y_test, y_pred_rf)
Out[58]: 0.9285714285714286

In [59]: confusion_matrix(y_test, y_pred_rf)
Out[59]: array([[1144,  56],
[ 43, 143]])

Grid Search Method

In [60]: params = {'n_estimators': np.arange(1, 10),
'criterion': ['entropy', 'gini'],
'max_features': ['auto', 'sqrt', 'log2'],
'bootstrap': [True, False]}

In [61]: model_cv = RandomizedSearchCV(RandomForestClassifier(), params)
#model_cv = GridSearchCV(RandomForestClassifier(), params)

In [62]: model_cv.fit(X_train_sm, y_train_sm)

Out[62]: RandomizedSearchCV(estimator=RandomForestClassifier(),
param_distributions={'bootstrap': [True, False],
'criterion': ['entropy', 'gini'],
'max_features': ['auto', 'sqrt', 'log2'],
'n_estimators': array([1, 2, 3, 4, 5, 6, 7, 8, 9])})

In [63]: model_cv.best_params_
```

OneDrive/Desktop/telecom-churn x Random Forest optimized Model-5-Copy1 x

localhost:8888/notebooks/OneDrive/Desktop/telecom-churn-master/Random%20Forest%20optimized%20Model-5-Copy1.ipynb

jupyter Random Forest optimized Model-5-Copy1 (autosaved)

File Edit View Insert Cell Kernel Widgets Help Not Trusted Python 3

```
In [59]: confusion_matrix(y_test,y_pred_rf)
Out[59]: array([[1144,  56],
               [ 43, 143]])
```

Grid Search Method

```
In [60]: params= {'n_estimators':np.arange(1,10),
                  'criterion': ['entropy', 'gini'],
                  'max_features': ['auto', 'sqrt', 'log2'],
                  'bootstrap': [True, False]
                }

In [61]: model_cv=RandomizedSearchCV(RandomForestClassifier(),params)
#model_cv = GridSearchCV(RandomForestClassifier(),params)

In [62]: model_cv.fit(X_train_sm,y_train_sm)

Out[62]: RandomizedSearchCV(estimator=RandomForestClassifier(),
                             param_distributions={'bootstrap': [True, False],
                                                  'criterion': ['entropy', 'gini'],
                                                  'max_features': ['auto', 'sqrt',
                                                                'log2'],
                                                  'n_estimators': array([1, 2, 3, 4, 5, 6, 7, 8, 9])})

In [63]: model_cv.best_params_

Out[63]: {'n_estimators': 8,
          'max_features': 'auto',
          'criterion': 'entropy',
          'bootstrap': False}

In [64]: model_cv.best_estimator_

Out[64]: RandomForestClassifier(bootstrap=False, criterion='entropy', n_estimators=8)
```

Activate Windows
Go to Settings to activate Windows.

OneDrive/Desktop/telecom-churn x Random Forest optimized Model-5-Copy1 x

localhost:8888/notebooks/OneDrive/Desktop/telecom-churn-master/Random%20Forest%20optimized%20Model-5-Copy1.ipynb

jupyter Random Forest optimized Model-5-Copy1 (autosaved)

File Edit View Insert Cell Kernel Widgets Help Not Trusted Python 3

ROC CURVE

```
In [65]: #checking for the ROC - CHECK FOR EFFICIENCY OF THE MODEL
model.predict_proba(X_test)

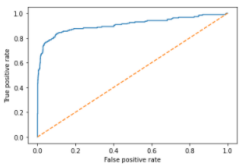
Out[65]: array([[0.97275532, 0.02724468],
               [0.15512411, 0.84487589],
               [0.77468549, 0.22531451],
               ...,
               [0.08874944, 0.11125056],
               [0.3216353 , 0.6783647 ],
               [0.32079094, 0.67920906]])

In [66]: #EXTRACTING ONLY the 2nd column values as it the predict proba values for the churners
y_pred_prob=model.predict_proba(X_test)[:,1]
print(y_pred_prob)

[0.02724468 0.84487589 0.22531451 ... 0.11125056 0.6783647 0.67920906]

In [67]: fpr, tpr , thresholds = roc_curve(y_test,y_pred_prob)

In [68]: plt.plot(fpr,tpr)
plt.xlabel("False positive rate")
plt.ylabel("True positive rate")
plt.plot([0,1],[0,1],"--")
plt.show()



In [69]: # The default probability threshold is 0.5
```

Activate Windows
Go to Settings to activate Windows.

OneDrive/Desktop/telecom-churn x Random Forest optimized Model-5-Copy1 x

localhost:8888/notebooks/OneDrive/Desktop/telecom-churn-master/Random%20Forest%20optimized%20Model-5-Copy1.ipynb

Jupyter Random Forest optimized Model-5-Copy1 (autosaved)

File Edit View Insert Cell Kernel Widgets Help Not Trusted Python 3

False positive rate

```
In [69]: ## The default probability threshold is 50%

In [70]: # checking the area under the curve - for a default random model AUC = 0.5
#So, we make sure we get better AUC score for a better performing model

In [71]: auc = roc_auc_score(y_test,y_pred_prob)

In [72]: print(auc)
0.9076022401433691
```

FROM THE Above result we can see the area under the curve is 90% which indicates the model is pretty good for the predictions further

```
In [73]: importances= model.feature_importances_
sort= np.argsort(importances)
label= X.columns[sort]

In [74]: X.shape
ss= range(X.shape[1])
print(ss)
range(0, 19)

In [75]: plt.barh(ss,importances[sort],tick_label=label)
Out[75]: <BarContainer object of 19 artists>
```

Activate Windows
Go to Settings to activate Windows.

OneDrive/Desktop/telecom-churn x Random Forest optimized Model-5-Copy1 x

localhost:8888/notebooks/OneDrive/Desktop/telecom-churn-master/Random%20Forest%20optimized%20Model-5-Copy1.ipynb

Jupyter Random Forest optimized Model-5-Copy1 (autosaved)

File Edit View Insert Cell Kernel Widgets Help Not Trusted Python 3

FROM THE Above result we can see the area under the curve is 90% which indicates the model is pretty good for the predictions further

```
In [73]: importances= model.feature_importances_
sort= np.argsort(importances)
label= X.columns[sort]

In [74]: X.shape
ss= range(X.shape[1])
print(ss)
range(0, 19)

In [75]: plt.barh(ss,importances[sort],tick_label=label)
Out[75]: <BarContainer object of 19 artists>
```

The above figure shows the feature importances for the particular model and which features drive churn the most

```
In [76]: import joblib

In [77]: joblib.dump(model,"Telecom_trained_model.m1")
Out[77]: ['Telecom_trained_model.m1']

In [78]: #to Load the trained model
```

Activate Windows
Go to Settings to activate Windows.

CONCLUSIONS

Machine learning, software-defined networks and edge cloud will be key components of the next generation of cellular networks. In this paper we investigated how these three elements can be jointly used in the system design for 5G networks, providing insights and results based on a dataset collected from hundreds of base stations of a sample cellular network in two different cities for more than a month. We described a clustering solution that limits the interactions among different controllers to minimize the need for inter-controller synchronization and reduce the control plane latency, and evaluated the performance of the proposed approach using data from a real network. Then, we outlined a second possible application enabled by our architecture, providing an extensive set of results related to the prediction accuracy of the number of users in base stations, using one month of data collected from the San Francisco base stations. In particular, we showed how the usage of the cluster-based architecture proposed in this paper can reduce the prediction error. With respect to a solution in which each base station tries to perform the regression based solely on its own data, as realized by a completely distributed architecture (e.g., in LTE), the controller-based design makes it possible to aggregate data from multiple neighboring base stations, and to predict a vector with the number of users in the nodes associated to the controller. This captures the spatial correlation given by the mobility of users, and, especially when increasing the temporal horizon of the prediction, reduces the RMSE by up to 53%. Finally, we also described some prediction-enabled use cases, either to control the network itself, or to offer innovative predictive services to network users, for example by recommending different driving itineraries to improve the user experience in

the network. We illustrated a real example in the San Francisco area, showing how the fastest route does not necessarily yield the best throughput, or the minimum outage, and that the best itinerary according to these metrics (which we derive from the number of users in each base station) may differ according to the departure time, so that a prediction-based approach is useful. We believe that this paper addresses for the first time several issues related to the practical deployment of machine learning techniques in 5G cellular networks, providing results and conclusions based on a real-network dataset. As future work, we will test different prediction algorithms (e.g., neural networks) to understand if it is possible to improve even more the prediction accuracy, and will extend the regression to other relevant metrics in the network (e.g., the number of handovers, the utilization), to verify the limits of what can be actually predicted in a cellular network.

REFERENCE

- [1] A. Gupta and R. K. Jha, "A survey of 5G network: Architecture and emerging technologies," *IEEE Access*, vol. 3, pp. 1206–1232, July 2015.
- [2] M. Agiwal, A. Roy, and N. Saxena, "Next generation 5G wireless networks: A comprehensive survey," *IEEE Commun. Surveys & Tuts.*, vol. 18, no. 3, pp. 1617–1655, 3rd Quart. 2016.
- [3] "Mobile visual networking index (VNI)," accessed: 2019-12-05. [Online]. Available: https://www.cisco.com/c/m/en_us/solutions/service-provider/visual-networking-index.html
- [4] H. Shariatmadari, R. Ratasuk, S. Iraji, A. Laya, T. Taleb, R. Jäntti, and A. Ghosh, "Machine-type communications: Current status and future perspectives toward 5G systems," *IEEE Commun. Mag.*, vol. 53, no. 9, pp. 10–17, Sept. 2015.
- [5] J. G. Andrews, S. Buzzi, W. Choi, S. V. Hanly, A. Lozano, A. C. Soong, and J. C. Zhang, "What will 5G be?" *IEEE J. Selected Areas Commun.*, vol. 32, no. 6, pp. 1065–1082, June 2014.
- [6] "Non-standalone and standalone: Two standards-based paths to 5G," accessed: 2019-12-05. [Online]. Available: <https://www.ericsson.com/en/blog/2019/7/standalone-and-non-standalone-5g-nr-two-5g-tracks>
- [7] 3GPP, "Architecture enhancements for 5G System (5GS) to support network data analytics services," 3rd Generation Partnership Project (3GPP), Technical Specification (TS 23.288), Sept. 2019, version 16.1.0. [Online]. Available: <https://portal.3gpp.org/desktopmodules/Specifications/SpecificationDetails.aspx?specificationId=3579>
- [1] A. Gupta and R. K. Jha, "A survey of 5G network: Architecture and emerging technologies," *IEEE Access*, vol. 3, pp. 1206–1232, July 2015.

- [2] M. Agiwal, A. Roy, and N. Saxena, "Next generation 5G wireless networks: A comprehensive survey," *IEEE Commun. Surveys & Tuts.*, vol. 18, no. 3, pp. 1617–1655, 3rd Quart. 2016.
- [3] "Mobile visual networking index (VNI)," accessed: 2019-12-05. [Online]. Available: https://www.cisco.com/c/m/en_us/solutions/service-provider/visual-networking-index.html
- [4] H. Shariatmadari, R. Ratasuk, S. Irajli, A. Laya, T. Taleb, R. Jäntti, and A. Ghosh, "Machine-type communications: Current status and future perspectives toward 5G systems," *IEEE Commun. Mag.*, vol. 53, no. 9, pp. 10–17, Sept. 2015.
- [5] J. G. Andrews, S. Buzzi, W. Choi, S. V. Hanly, A. Lozano, A. C. Soong, and J. C. Zhang, "What will 5G be?" *IEEE J. Selected Areas Commun.*, vol. 32, no. 6, pp. 1065–1082, June 2014.
- [6] "Non-standalone and standalone: Two standards-based paths to 5G," accessed: 2019-12-05. [Online]. Available: <https://www.ericsson.com/en/blog/2019/7/standalone-and-non-standalone-5g-nr-two-5g-tracks>
- [7] 3GPP, "Architecture enhancements for 5G System (5GS) to support network data analytics services," 3rd Generation Partnership Project (3GPP), Technical Specification (TS 23.288), Sept. 2019, version 16.1.0. [Online]. Available: <https://portal.3gpp.org/desktopmodules/Specifications/SpecificationDetails.aspx?specificationId=3579>
- [8] A. Gupta and R. K. Jha, "A survey of 5G network: Architecture and emerging technologies," *IEEE Access*, vol. 3, pp. 1206–1232, July 2015.
- [9] M. Agiwal, A. Roy, and N. Saxena, "Next generation 5G wireless networks: A comprehensive survey," *IEEE Commun. Surveys & Tuts.*, vol. 18, no. 3, pp. 1617–1655, 3rd Quart. 2016.

[9] “Mobile visual networking index (VNI),” accessed: 2019-12-05. [Online]. Available: visual-networking-index.html

[10] H. Shariatmadari, R. Ratasuk, S. Iraj, A. Laya, T. Taleb, R. Jäntti, and A. Ghosh, “Machine-type communications: Current status and future perspectives toward 5G systems,” *IEEE Commun. Mag.*, vol. 53, no. 9, pp. 10–17, Sept. 2015.

[11] J. G. Andrews, S. Buzzi, W. Choi, S. V. Hanly, A. Lozano, A. C. Soong, and J. C. Zhang, “What will 5G be?” *IEEE J. Selected Areas Commun.*, vol. 32, no. 6, pp. 1065–1082, June 2014.

[12] “Non-standalone and standalone: Two standards-based paths to 5G,” accessed: 2019-12-05. [Online]. Available: <https://www.ericsson.com/en/blog/2019/7/standalone-and-non-standalone-5g-nr-two-5g-tracks>

[13] 3GPP, “Architecture enhancements for 5G System (5GS) to support network data analytics services,” 3rd Generation Partnership Project (3GPP), Technical Specification (TS 23.288), Sept. 2019, version 16.1.0. [Online].

[14] 3GPP, “System architecture for the 5G System (5GS),” 3rd Generation Partnership Project (3GPP), Technical Specification (TS 23.501), Sept. 2019, version 16.2.0. [Online]. Available: <https://portal.3gpp.org/desktopmodules/Specifications/SpecificationDetails.aspx?specificationId=3144>

[15] 3GPP, “5G System; Network data analytics services; Stage 3,” 3rd Generation Partnership Project (3GPP), Technical Specification (TS 29.520), September 2019, version 16.1.0. [Online]. Available:

<https://portal.3gpp.org/desktopmodules/Specifications/SpecificationDetails.aspx?specificationId=3355>

[16] 3GPP, “5G System; Unified data management services; Stage 3,” 3rd Generation Partnership Project (3GPP), Technical Specification (TS 29.503), Sept. 2019, version 16.1.0. [Online]. Available: <https://portal.3gpp.org/desktopmodules/Specifications/SpecificationDetails.aspx?specificationId=3342>

[17] C. Hernandez-Chulde and C. Cervello-Pastor, “Intelligent optimization and machine learning for 5G network control and management,” in Proc. PAAMS, June 2019, pp. 339–342.