FORECASTING E – COMMERCE SALES USING DATA SCIENCE

A Project report submitted to

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING WITH DATA SCIENCE

In partial fulfilment of the requirements for the award of the Degree of

BACHELOR OF TECHNOLOGY
IN
COMPUTER SCIENCE & ENGINEERING WITH DATA SCIENCE

By

MENTA GOWTHAM	(Y20CDS038)
EDA VIJAYA SAMBI REDDY	(L21CDS065)
BATTU HARI KRISHNA	(Y20CDS007)
VEMA SAI THIRUPATHI RAO	(Y20CDS063)
ADINA YUGANDHAR BABU	(Y20CDS001)

Under the Guidance of
Mrs.V. AMANI, M. Tech
ASSISTANT PROFESSOR

DEPARTMENT OF COMPUTER SCIENCE &ENGINEERING WITH DATA SCIENCE

CHALAPATHI INSTITUTE OF ENGINEERING AND TECHNOLOGY (AUTONOMOUS)
(Accredited by NAAC with 'A' grade, accredited by NBA, Approved by A.I.C.T.E,
Affiliated To Acharya Nagarjuna University)
Guntur-522034

2023-2024

CHALAPATHI INSTITUTE OF ENGINEERING AND TECHNOLOGY (AUTONOMOUS)

(Accredited by NAAC with 'A' grade, accredited by NBA, Approved by A.I.C.T.E, Affiliated To Acharya Nagarjuna University)

CHALAPATHI NAGAR, LAM, GUNTUR

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING WITH DATA SCIENCE

CERTIFICATE

This is to certify that the project work entitled as **"Forecasting E-Commerce Sales using Data Science"** submitted by **Menta Gowtham(Y20CDS038), Eda Vijaya Sambi Reddy(L21CDS065), Battu Hari Krishna(Y20CDS007), Vema Sai Thirupathi Rao(Y20CDS063)** and **Adina Yugandhar Babu(Y20CDS001)** in partial fulfilment for the award of the Degree of **BACHELOR OF TECHNOLOGY IN COMPUTER SCIENCE AND ENGINEERING WITH DATA SCIENCE** is a record of bonafied work carried out under my guidance and supervision.

GUIDE

Mrs. V.AMANI, M.TECH
Assistant Professor

HEAD OF THE DEPARTMENT

Mrs.K.ARUNA KUMARI, M.TECH, (Ph.D)
CSDS-HOD

ACKNOWLEDGEMENT

We express our sincere thanks to our beloved Chairman sir, **Sri. Y. V. ANJANEYULU** for providing support and stimulating environment for developing the project.

We express deep sense of reverence and profound gratitude to **Dr. M. CHANDRA SEKHAR, Ph. D,** Principal for providing us the great support in carrying out the project.

It plunges us in exhilaration in taking privilege in expressing our heartfelt gratitude to **MRS.K.ARUNA KUMARI** , **M. Tech**, **(Ph. D),** HOD-CSDS for providing us every facility and for constant supervision.

We are thankful to our guide **MRS.AMANI , M. Tech,** Assistant professor, Dept. of CSDS for his constant encouragement, suggestions, supervision, and abundant support throughout the project.

Thanks to all the teaching and non-teaching staff and lab technicians for their support and also to our team mates for their valuable Co-operation.

By

MENTA GOWTHAM (Y20CDS038)
EDA VIJAYA SAMBI REDDY (L21CDS065)
BATTU HARI KRISHNA (Y20CDS007)
VEMA SAI THIRUPATHI RAO (Y20CDS063)
ADINA YUGANDHAR BABU (Y20CDS001)

CONTENTS

INDEX

LIST OF FIGURES

ABBREVATIONS

TF-IDF: - TERM FREQUENCY-INVERSE DOCUMENT FREQUENCY

NLP: - NATURAL LANGUAGE PROCESSING

UML: -UNIFIED MODELING LANGUAGE

ML: - MACHINE LEARNING

DL: -DEEP LEARNING

ABSTRACT

In the world of online shopping, accurately predicting how much stuff people will buy is really important. This helps businesses make smart choices about what to stock up on, how much to buy, and how to keep their business going strong. To do this, companies look at past sales info and use fancy computer programs to guess what future sales might look like. There are different ways they do this, like looking at trends over time, using math formulas, and even teaching computers to learn from the data. They also think about things like when people tend to buy more (like around holidays), what people like to buy, and what's happening in the market. This helps them optimize inventory, improve customer satisfaction, and stay competitive in the market. But it's not always easy - sometimes the data isn't great, or the computer guesses too much or too little. Still, by getting good at this guessing game, businesses can make better decisions, keep their customers happy, and stay ahead in the online shopping world. Still, mastering sales forecasting is essential for ecommerce success.

KEYWORDS : *E-Commerce, Demand forecasting, Big data analysis, Machine learning.*

THE PROBLEM STATEMENT

The e-commerce industry faces significant challenges in accurately forecasting sales due to issues related to data quality, complex market dynamics, seasonal fluctuations, demand volatility, and inadequate integration with operational processes. Despite the abundance of data, many e-commerce businesses struggle to develop reliable forecasting models that can effectively predict future sales trends, leading to suboptimal decision-making and resource allocation. Addressing these challenges requires innovative approaches to data collection, preprocessing, and modeling, as well as closer alignment between forecasting efforts and operational decision-making processes, to enable businesses to anticipate market trends, optimize inventory management, and maintain competitiveness in the dynamic e-commerce landscape.

CHAPTER - 1

INTRODUCTION

1.1BACKGROUND WORK

Today, e-commerce is in more and more increasing competition, so companies must adopt strategic planning and make the right marketing decision. The first step in the planning and decision-making process is to predict the future demand for products and thus the resources can be adjusted in time to meet the demand. The significance of forecasting for corporations has been extensively discussed by various authors and experts. For instance, utilizing machine learning models for sales forecasting based on extensive historical data has proven instrumental for companies like Walmart. This approach aids in optimizing various aspects of their business operations, including cash flow, inventory management, production, and financial planning. Additionally, it helps in reducing uncertainty and enables proactive responses to market changes

Numerous machine learning (ML) and deep learning models, such as Support Vector Machine (SVM), Random Forest (RF), Gradient Boosting (GB), and Extreme Gradient Boosting (XGBoost), have demonstrated commendable performance in sales forecasting. However, these models typically necessitate a substantial volume of data to establish meaningful correlations and develop accurate predictive functions. Consequently, they are often trained on months or even years of data, making it challenging for researchers to achieve satisfactory performance with only a few days' worth of data for sales forecasting.

Regarding sales forecasting methodologies, many companies rely on simplistic approaches utilizing single sale attributes, basic promotional plans, and seasonal factors. Fewer companies delve into refining discount information, often unaware of its precise impact. Various types of discounts, such as bundle reductions, direct reductions, and coupon offers, significantly influence consumer purchasing decisions. Despite the recognized importance of discount information in marketing strategy formulation and consumer attraction, the distinct value of different types of discounts in sales forecasting remains relatively unexplored in academic research.

CHAPTER - 2

SYSTEM ANANYSIS

2.1. EXISTING SYSTEM

In our previous Dialog flow is a powerful natural language understanding platform that allows developers to build conversational interfaces for websites, mobile apps, messaging platforms, and IoT devices.

2.1.1 DISADVANTAGES:

- Limited Customization
- Limited Offline Functionality

2.2 PROPOSED SYSTEM:

The proposed system for chatbot development aims to address the limitations of existing platforms by offering a highly customizable, cost-effective, and privacy-conscious solution. Leveraging cutting-edge machine learning techniques, the system provides developers with extensive customization options to build sophisticated conversational agents tailored to specific use cases. It prioritizes user privacy and data security through robust encryption mechanisms and compliance with stringent data protection regulations.

2.2.1 ADVANTAGES:

- Flexibility Fosters.

2.3. SYSTEM STUDY

2.3.1. FEASIBILITY STUDY

A feasibility study for a chatbot development system involves assessing various aspects to determine its viability and potential success.

Three key issues concerned within the FEASIBILITY analysis area unit

- ECONOMICAL FEASIBILITY
- TECHNICAL FEASIBILITY
- SOCIAL FEASIBILITY

ECONOMICAL FEASIBILITY: This study is a critical aspect of assessing the viability a chatbot development system. It involves analysing the financial implications and benefits associated with developing, deploying, and maintaining the system

TECHNICAL FEASIBILITY: This study is Evaluate the technical requirements and capabilities needed to develop the chatbot system, including programming languages, frameworks, and infrastructure.

Assess the availability of necessary resources such as skilled developers, tools, and technologies.

Determine if the proposed system can integrate with existing platforms, databases, and APIs effectively.

SOCIAL FEASIBILITY: The facet of study is to assess the social implications of deploying chatbots, including their impact on employment, user satisfaction, and inclusivity.
Identify opportunities to enhance social responsibility and ethical practices in chatbot development and deployment.

2.4. LITERATURE SURVEY

Every newspaper publisher faced with the problem of determining the number of copies of newspaper and distributing them to the retail traders. Two aspects need to be balanced out in order to optimize the economical success which is the number of unsold copies should be minimal to reduce the cost of production, and the sell-out rate should be minimal to maximize the number of sold copies. Thus, a good sales rate prediction is necessary to optimize both antagonistic aspects. This paper utilized artificial neural network to predict newspaper sales for one vendor in the area of Sungai Petani, Malaysia. The predicted sales value can help the company to optimize their sales. The main objective is to develop a prototype that apply artificial neural network so that it can predict the future trend as well as the future daily sale. The network will consist of three layer which is input layer, one hidden layer and output layer. The input layer will have six input node where this will be the factor that will affect the output which is the number of copies that sold. The network will be trained with history data of a one year records of data. The output produced has the error value as low as 1.24% while the correlation coefficient between prediction and actual value is 0.1197.

Sales forecasting is an important aspect of different companies engaged in retailing, logistics, manufacturing, marketing and wholesaling. It allows companies to efficiently allocate resources, to estimate achievable sales revenue and to plan a better strategy for future growth of the company. In this paper, prediction of sales of a product from a particular outlet is performed via a two-level approach that produces better predictive performance compared to any of the popular single model predictive learning algorithms. The approach is performed on Big Mart Sales data of the year 2013. Data exploration, data transformation and feature engineering play a vital role in predicting accurate results. The result demonstrated that the two-level statistical approach performed better than a single model approach as the former provided more information that leads to better prediction.

Baseline prediction is an important to devise marketing strategy for a consumer goods product. Simulation techniques, time series algorithms are often used to generate baseline for the future. However the algorithm that fits a particular point of sales (POS) data varies according to the datasets. Sample set of point of sales data were simulated under different conditions and constraints incorporating seasonal and non seasonal trends. This study has compared the performance of two time series models namely Winters model and linear exponential smoothening on the simulated datasets.Winters model was found to be a better fit for the point of sales data that were used for testing.

CHAPTER - 3

SYSTEM REQUIREMENTS

3.1 SOFTWARE REQUIREMENTS

The functional requirements or the overall description documents include the product perspective and features, operating system and operating environment, graphics requirements, design constraints and user documentation.
The appropriation of requirements and implementation constraints gives the general overview of the project in regards to what the areas of strength and deficit are and how to tackle them.

• Python idle 3.7 version (or) • Jupiter (or) • Google Collab

3.2 HARDWARE REQUIREMENTS

Minimum hardware requirements are very dependent on the particular software being developed by a given Enthought Python / Google collab. Applications that need to store large arrays/objects in memory will require more RAM, whereas applications that need to perform numerous calculations or tasks more quickly will require a faster processor.

- **Operating system : windows, Linux**
- **Processor : minimum intel i5**
- **Ram : minimum 4gb**
- **Hard disk : minimum 250gb**

CHAPTER - 4

SYSTEM DESIGN

4.1. SYSTEM ARCHITECTURE

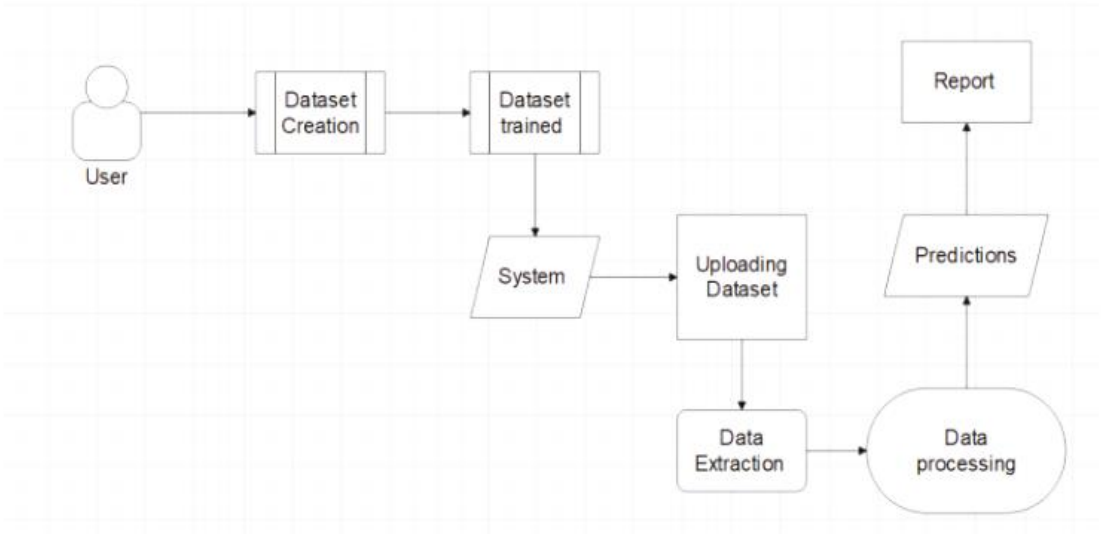


Fig 4.1 System Architecture for Forecasting E-commerce Sales

4.2. UML DIAGRAMS

- UML remains for Unified Modeling Language. UML is an institutionalized broadly useful displaying dialect in the field of protest situated programming designing. The standard is overseen, and was made by the Object Management Group. The objective is for UML to end up a typical dialect for making models of protest arranged PC programming. In its present shape UML is contained two noteworthy segments: a Meta-display and a documentation. Later on, some type of technique or process may likewise be added to; or connected with UML.

- The Unified Modeling Language is a standard dialect for indicating, Visualization, Constructing and recording the antiques of programming framework, and additionally for business displaying and other non-programming frameworks.

- The UML speaks to an accumulation of best building rehearses that have demonstrated effective in the displaying of vast and complex frameworks.

- The UML is an imperative piece of creating articles arranged programming and the product improvement prepare. The UML utilizes for the most part graphical documentations to express the plan of programming tasks.

GOALS

The Primary objectives in the plan of the UML are as per the following:

1. Provide clients a prepared to-utilize, expressive visual displaying Language with the goal that they can create and trade important models.
2. Provide extendibility and specialization instruments to develop the center ideas.
3. Be free of specific programming dialects and improvement handle.
4. Provide a formal reason for comprehension the displaying dialect.
5. Encourage the development of OO devices showcase.
6. Support more elevated amount improvement ideas, for example, coordinated efforts, systems, examples and parts.
7. Integrate best practices.

4.2.1. USE CASE DIAGRAM

The use case diagram is a technique used in the development of a software or system to capture the functional requirements of a system. The use case diagram is used to construct behavioral things in a model, since the use case diagrams can explain the interactions that occur between the users and the system itself. A use case diagram can define functionality and software features from user's perspective. The design of sales forecasting application that adopt RFM concept is done by modeling the use case diagram as shown in Figure.

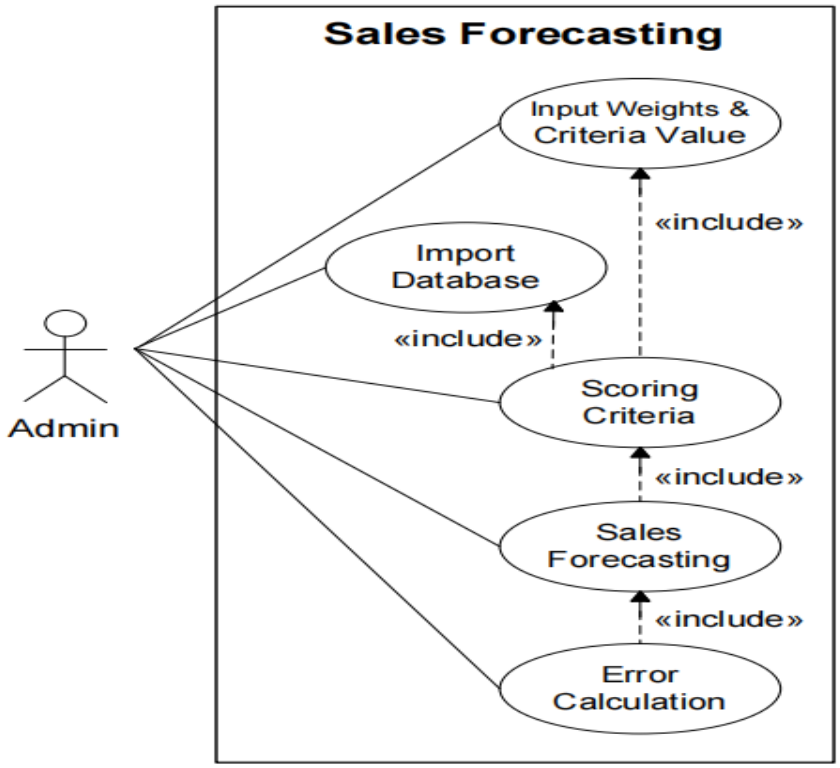


Fig: 4.2.1 Use Case Diagram for Sales Forecasting Application

The use case diagram can be used as the initial stage of development of this system In this use case diagram, you can see five use cases and one actor. Each use case is self explanatory and can represent the interaction between functions of the sales forecasting application. Meanwhile, the actor describes someone who interacts with the system. There are five use cases used in the system, the first is the function to input the value of each weight and criteria that most influence the sales volume based on the evaluation of the best-worst method. Second is the function to import data from the database of a case company. The third use case is the function for projecting the customer value into numbers. The fourth use case is sales forecasting that serves to forecast the sales volume of each product for the coming month, and the last use case is the calculation of sales forecasting errors (accuracy).

4.2.2. CLASS DIAGRAM

The class diagram is one of the main diagrams of UML that describes a class or blueprint object on a system. The analysis of class diagram formation is a core activity that greatly influences the software architecture designed up to the coding stage. UML Class Diagrams describe the static structure of classes within a system, which shows what interacts but not what happens when they interact. The class diagrams contain the system's classes, attributes, operations and relationships between classes.

Fig. 4.2 is a class diagram design that still requires customization when it will be used as a guide in building sales forecasting application at a later stage.

The class diagram in Fig. 2 shows that the class „Forecast" is the main class of the system. It coordinates with the „Scoring" class and the „Realization" class. Meanwhile, the „Scoring" class and the „Realization" class coordinate with the „Transaction" class. The „Error" class coordinates with the „Forecast" class and the „Realization" class.

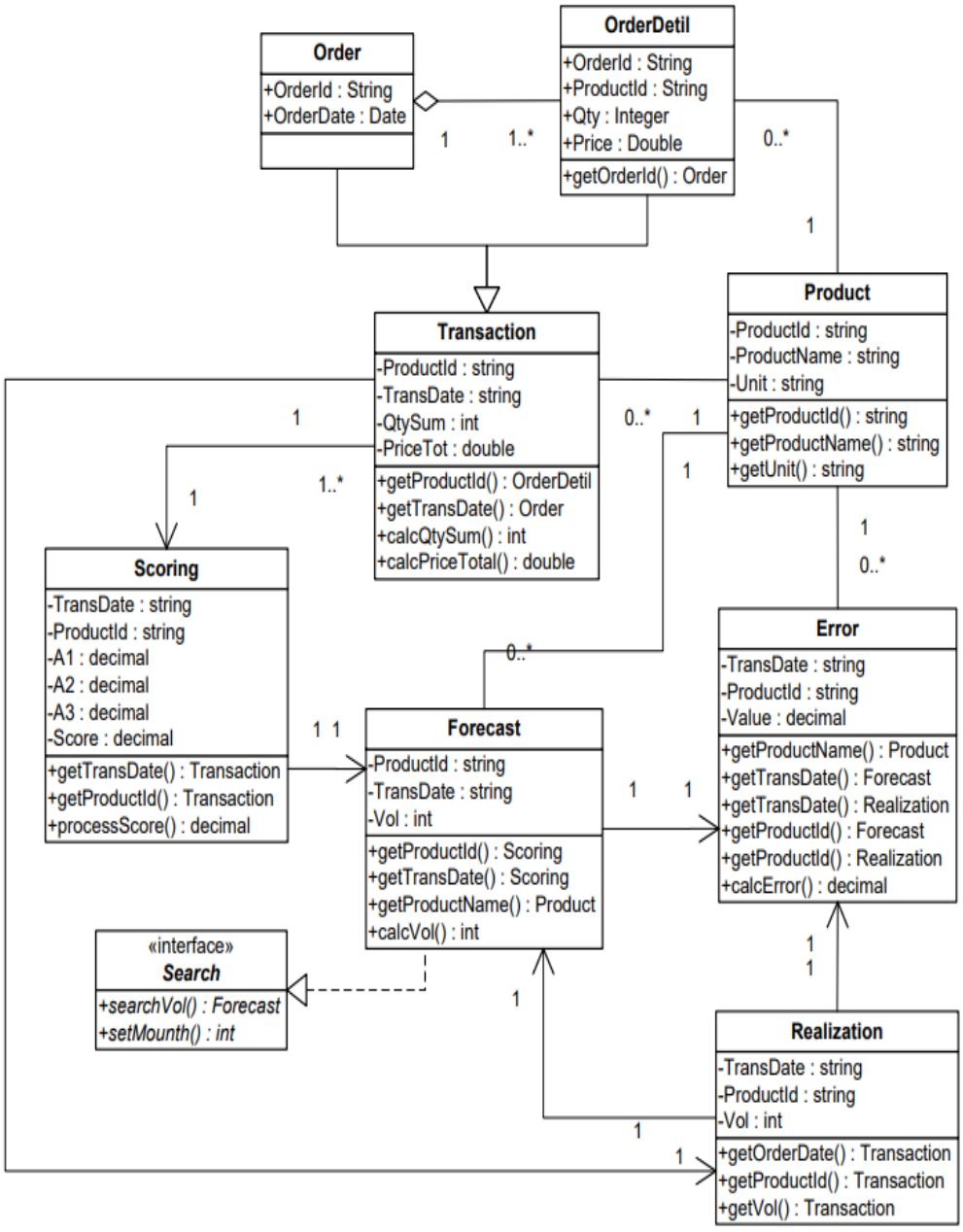


Fig: 4.2.2 Class Diagram for Forecasting E-commerce Sales

4.2.3. SEQUENCE DIAGRAM

A sequence diagram in Unified Modeling Language (UML) may be a quite interaction diagram that shows however processes operate with each other and in what order. it's a construct of a Message Sequence Chart. Sequence diagrams ar generally known as event diagrams, event situations, and temporal order diagrams.

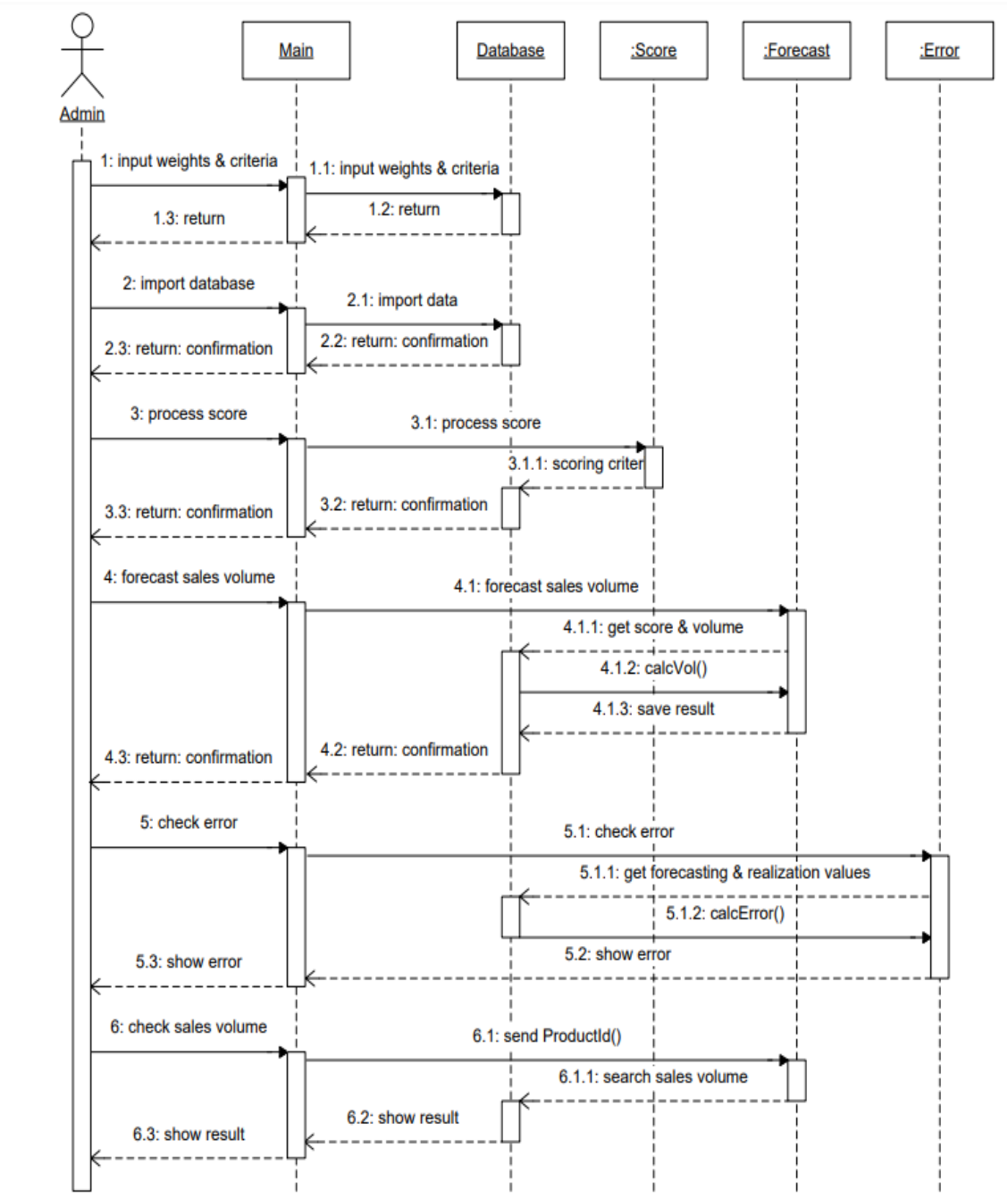


Fig: 4.2.3. Sequence Diagram representing the design of Forecasting Sales

CHAPTER - 5

SYSTEM IMPLEMENTATION

5.1. SYSTEM MODEL

Here collect 89 queries issued by the subjects, and name them as "User Q". As this approach might induce a bias towards topics in which lists are more useful than general web queries, we further randomly sample another set of 105 English queries from a query log of a commercial search engine, and name this set of queries as "Rand Q". We first ask a subject to manually create facets and add items that are covered by the query, based on his/her knowledge after a deep survey on any related resources (such as Wikipedia, Freebase, or official web sites related to the query).

5.2. MODULE DESCRIPTION

1) Question Module: The question module is responsible for parsing user queries, identifying the user's intent, and extracting relevant information. It utilizes natural language processing techniques to understand the context and structure of the question, enabling the chatbot to generate appropriate responses. Additionally, the question module may incorporate machine learning algorithms to improve accuracy and adapt to user input over time.
2) Answer Module: Using this answer module generates responses based on the input received from the user, utilizing predefined rules, templates, or machine learning algorithms. It selects the most appropriate response considering the context, user intent, and available information. Additionally, the answer module may incorporate natural language generation techniques to create human-like responses for improved user interaction.

5.3 SOFTWARE ENVIRONMENT

5.1.1. What is Python: -

Below are some facts about Python.

Python is currently the most widely used multi-purpose, high-level programming language.

Python allows programming in Object-Oriented and Procedural paradigms. Python programs generally are smaller than other programming languages like Java.

Programmers have to type relatively less and indentation requirement of the language, makes them readable all the time.

Python language is being used by almost all tech-giant companies like – Google, Amazon, Facebook, Instagram, Dropbox, Uber… etc.

The biggest strength of Python is huge collection of standard libraries which can be used for the following –

- Machine Learning
- GUI Applications (like Kivy, Tkinter, PyQt, etc.)
- Web frameworks like Django (used by YouTube, Instagram, Dropbox)
- Image processing (like OpenCV, Pillow)
- Web scraping (like Scrapy, Beautiful Soup, Selenium)
- Test frameworks
- Multimedia

5.1.1.1 Advantages of Python: -

Let's see how Python dominates over other languages.

1.1.1.1 1. Extensive Libraries

Python downloads with an extensive library and it ***contain code for various purposes like regular expressions, documentation-generation, unit-testing, web browsers, threading, databases, CGI, email, image manipulation, and more.*** So, we don't have to write the complete code for that manually.

1.1.1.2 2. Extensible

As we have seen earlier, Python can be **extended to other languages**. You can write some of your code in languages like C++ or C. This comes in handy, especially in projects.

1.1.1.3 3. Embeddable

Complimentary to extensibility, Python is embeddable as well. You can put your Python code in your source code of a different language, like C++. This lets us add **scripting capabilities** to our code in the other language.

1.1.1.4 4. Improved Productivity

The language's simplicity and extensive libraries render programmers **more productive** than languages like Java and C++ do. Also, the fact that you need to write less and get more things done.

1.1.1.5 5. IOT Opportunities

Since Python forms the basis of new platforms like Raspberry Pi, it finds the future bright for the Internet of Things. This is a way to connect the language with the real world.

1.1.1.6 6. Simple and Easy

When working with Java, you may have to create a class to print **'Hello World'**. But in Python, just a print statement will do. It is also quite **easy to learn**, **understand**, and **code**. This is why when people pick up Python, they have a hard time adjusting to other more verbose languages like Java.

1.1.1.7 7. Readable

Because it is not such a verbose language, reading Python is much like reading English. This is the reason why it is so easy to learn, understand, and code. It also does not need curly braces to define blocks, and **indentation is mandatory**. This further aids the readability of the code.

1.1.1.8 8. Object-Oriented

This language supports both the **procedural and object-oriented** programming paradigms. While functions help us with code reusability, classes and objects let us model the real world. A class allows the **encapsulation of data** and functions into one.

1.1.1.9 9. Free and Open-Source

Like we said earlier, Python is **freely available**. But not only can you **download Python** for free, but you can also download its source code, make changes to it, and even distribute it. It downloads with an extensive collection of libraries to help you with your tasks.

1.1.1.10 10. Portable

When you code your project in a language like C++, you may need to make some changes to it if you want to run it on another platform. But it isn't the same with Python. Here, you need to **code only once**, and you can run it anywhere. This is called **Write Once Run Anywhere (WORA)**. However, you need to be careful enough not to include any system-dependent features.

1.1.1.11 11. Interpreted

Lastly, we will say that it is an interpreted language. Since statements are executed one by one, **debugging is easier** than in compiled languages.
Any doubts till now in the advantages of Python? Mention in the comment section.

5.1.1.2 Advantages of Python Over Other Languages

1.1.2.1 1. Less Coding

Almost all of the tasks done in Python requires less coding when the same task is done in other languages. Python also has an awesome standard library support, so you don't have to search for any third-party libraries to get your job done. This is the reason that many people suggest learning Python to beginners.

1.1.2.2 2. Affordable

Python is free therefore individuals, small companies or big organizations can leverage the free available resources to build applications. Python is popular and widely used so it gives you better community support.

The 2019 Git hub annual survey showed us that Python has overtaken Java in the most popular programming language category.

1.1.2.3 3. Python is for Everyone

Python code can run on any machine whether it is Linux, Mac or Windows. Programmers need to learn different languages for different jobs but with Python, you can professionally build web apps, perform data analysis and **machine learning**, automate things, do web scraping and also build games and powerful visualizations. It is an all-rounder programming language.

5.1.1.3 Disadvantages of Python

So far, we've seen why Python is a great choice for your project. But if you choose it, you should be aware of its consequences as well. Let's now see the downsides of choosing Python over another language.

1.1.3.1 1. Speed Limitations

We have seen that Python code is executed line by line. But since Python is interpreted, it often results in **slow execution**. This, however, isn't a problem unless speed is a focal point for the project. In other words, unless high speed is a requirement, the benefits offered by Python are enough to distract us from its speed limitations.

1.1.3.2 2. Weak in Mobile Computing and Browsers

While it serves as an excellent server-side language, Python is much rarely seen on the **client-side**. Besides that, it is rarely ever used to implement smartphone-based applications. One such application is called **Carbonnelle**.
The reason it is not so famous despite the existence of python is that it isn't that secure.

1.1.3.3 3. Design Restrictions

As you know, Python is **dynamically-typed**. This means that you don't need to declare the type of variable while writing the code. It uses **duck-typing**. But wait,

what's that? Well, it just means that if it looks like a duck, it must be a duck. While this is easy on the programmers during coding, it can **raise run-time errors**.

1.1.3.4 4. Underdeveloped Database Access Layers

Compared to more widely used technologies like **JDBC (Java Data Base Connectivity)** and **ODBC (Open Data Base Connectivity)**, Python's database access layers are a bit underdeveloped. Consequently, it is less often applied in huge enterprises.

1.1.3.5 5. Simple

No, we're not kidding. Python's simplicity can indeed be a problem. Take my example. I don't do Java, I'm more of a Python person. To me, its syntax is so simple that the verbosity of Java code seems unnecessary.

This was all about the Advantages and Disadvantages of Python Programming Language.

History Of Python: -

What do the alphabet and the programming language Python have in common? Right, both start with ABC. If we are talking about ABC in the Python context, it's clear that the programming language ABC is meant. ABC is a general-purpose programming language and programming environment, which had been developed in the Netherlands, Amsterdam, at the CWI (Centrum Wickenden & Informatica). The greatest achievement of ABC was to influence the design of Python. Python was conceptualized in the late 1980s. Guido van Rossum worked that time in a project at the CWI, called Amoeba, a distributed operating system. In an interview with Bill Venners[1], Guido van Rossum said: "In the early 1980s, I worked as an implementer on a team building a language called ABC at Centrum poor Wickenden Informatica (CWI). I don't know how well people know ABC's influence on Python. I try to mention ABC's influence because I'm indebted to everything I learned during that project and to the people who worked on it." Later on in the same Interview, Guido van Rossum continued: "I remembered all my experience and some of my frustration with ABC. I decided to try to design a simple scripting language that possessed some of ABC's better properties, but without its problems. So I started typing. I created a simple virtual machine, a simple parser, and a simple runtime. I made my own version of the various ABC parts that I liked. I created a basic syntax, used indentation for statement grouping instead of curly braces or begin- end blocks, and developed a small number of powerful data types: a hash table (or dictionary, as we call it), a list, strings, and numbers."

What Is Machine Learning: -

Before we take a look at the details of various machine learning methods, let's start by looking at what machine learning is, and what it isn't. Machine learning is often categorized as a subfield of artificial intelligence, but I find that categorization can often be misleading at first brush. The study of machine learning certainly arose from research in this context, but in the data science application of machine learning methods, it's more helpful to think of machine learning as a means of ***building models of data***.

Fundamentally, machine learning involves building mathematical models to help understand data. "Learning" enters the fray when we give these models ***tunable parameters*** that can be adapted to observed data; in this way the program can be considered to be "learning" from the data. Once these models have been fit to previously seen data, they can be used to predict and understand aspects of newly observed data. I'll leave to the reader the more philosophical digression regarding the extent to which this type of mathematical, model-based "learning" is similar to the "learning" exhibited by the human brain. Understanding the problem setting in machine learning is essential to using these tools effectively, and so we will start with some broad categorizations of the types of approaches we'll discuss here.

Categories of Machine Learning:

At the most fundamental level, machine learning can be categorized into two main types: supervised learning and unsupervised learning.
Supervised learning involves somehow modeling the relationship between measured features of data and some label associated with the data; once this model is determined, it can be used to apply labels to new, unknown data. This is further subdivided into ***classification*** tasks and ***regression*** tasks: in classification, the labels are discrete categories, while in regression, the labels are continuous quantities. We will see examples of both types of supervised learning in the following section.

Unsupervised learning involves modeling the features of a dataset without reference to any label, and is often described as "letting the dataset speak for itself." These models include tasks such as ***clustering*** and ***dimensionality reduction.*** Clustering algorithms identify distinct groups of data, while dimensionality reduction algorithms search for more succinct representations of the data. We will see examples of both types of unsupervised learning in the following section.

5.1.2 Need for Machine Learning

Human beings, at this moment, are the most intelligent and advanced species on earth because they can think, evaluate and solve complex problems. On the other side, AI is still in its initial stage and haven’t surpassed human intelligence in many aspects. Then the question is that what is the need to make machine learn? The most suitable reason for doing this is, “to make decisions, based on data, with efficiency and scale”.

Lately, organizations are investing heavily in newer technologies like Artificial Intelligence, Machine Learning and Deep Learning to get the key information from data to perform several real-world tasks and solve problems. We can call it data-driven decisions taken by machines, particularly to automate the process. These data-driven decisions can be used, instead of using programming logic, in the problems that

cannot be programmed inherently. The fact is that we can't do without human intelligence, but other aspect is that we all need to solve real-world problems with efficiency at a huge scale. That is why the need for machine learning arises.

5.1.3 Challenges in Machines Learning: -

While Machine Learning is rapidly evolving, making significant strides with cybersecurity and autonomous cars, this segment of AI as whole still has a long way to go. The reason behind is that ML has not been able to overcome number of challenges. The challenges that ML is facing currently are −

Quality of data − Having good-quality data for ML algorithms is one of the biggest challenges. Use of low-quality data leads to the problems related to data preprocessing and feature extraction.

Time-Consuming task − Another challenge faced by ML models is the consumption of time especially for data acquisition, feature extraction and retrieval.

Lack of specialist persons − As ML technology is still in its infancy stage, availability of expert resources is a tough job.

No clear objective for formulating business problems − Having no clear objective and well-defined goal for business problems is another key challenge for ML because this technology is not that mature yet.

Issue of overfitting & underfitting − If the model is overfitting or underfitting, it cannot be represented well for the problem.

Curse of dimensionality − Another challenge ML model faces is too many features of data points. This can be a real hindrance.

Difficulty in deployment − Complexity of the ML model makes it quite difficult to be deployed in real life.

5.1.4 Applications of Machines Learning: -

Machine Learning is the most rapidly growing technology and according to researchers we are in the golden year of AI and ML. It is used to solve many real-world complex problems which cannot be solved with traditional approach. Following are some real-world applications of ML −

- Emotion analysis
- Sentiment analysis
- Error detection and prevention
- Weather forecasting and prediction
- Stock market analysis and forecasting
- Speech synthesis
- Speech recognition

- Customer segmentation
- Object recognition
- Fraud detection
- Fraud prevention
- Recommendation of products to customer in online shopping

5.2.1. How to Start Learning Machine Learning?

Arthur Samuel coined the term **"Machine Learning"** in 1959 and defined it as a **"Field of study that gives computers the capability to learn without being explicitly programmed".**
And that was the beginning of Machine Learning! In modern times, Machine Learning is one of the most popular (if not the most!) career choices. According to Indeed, Machine Learning Engineer Is the Best Job of 2019 with a ***344%*** growth and an average base salary of **$146,085** per year.
But there is still a lot of doubt about what exactly is Machine Learning and how to start learning it? So this article deals with the Basics of Machine Learning and also the path you can follow to eventually become a full-fledged Machine Learning Engineer. Now let's get started!!!

5.2.1.1 How to start learning ML?

This is a rough roadmap you can follow on your way to becoming an insanely talented Machine Learning Engineer. Of course, you can always modify the steps according to your needs to reach your desired end-goal!

5.2.1.2 Step 1 – Understand the Prerequisites

In case you are a genius, you could start ML directly but normally, there are some prerequisites that you need to know which include Linear Algebra, Multivariate Calculus,
Statistics, and Python. And if you don't know these, never fear! You don't need a Ph.D. degree in these topics to get started but you do need a basic understanding.

5.2.1.2.1 *(*a) Learn Linear Algebra and Multivariate Calculus

Both Linear Algebra and Multivariate Calculus are important in Machine Learning. However, the extent to which you need them depends on your role as a data scientist. If you are more focused on application heavy machine learning, then you will not be that heavily focused on math as there are many common libraries available. But if you want to focus on R&D in Machine Learning, then mastery of Linear Algebra and Multivariate Calculus is very important as you will have to implement many ML algorithms from scratch.

5.2.1.2.2 (b) Learn Statistics

Data plays a huge role in Machine Learning. In fact, around 80% of your time as an ML expert will be spent collecting and cleaning data. And statistics is a field that handles the collection, analysis, and presentation of data. So, it is no surprise that you need to learn it!!!
Some of the key concepts in statistics that are important are Statistical Significance, Probability Distributions, Hypothesis Testing, Regression, etc. Also, Bayesian Thinking is also a very important part of ML which deals with various concepts like Conditional Probability, Priors, and Posteriors, Maximum Likelihood, etc.

5.2.1.2.3 (c) Learn Python

Some people prefer to skip Linear Algebra, Multivariate Calculus and Statistics and learn them as they go along with trial and error. But the one thing that you absolutely cannot skip is Python! While there are other languages you can use for Machine Learning like R, Scala, etc. Python is currently the most popular language for ML. In fact, there are many Python libraries that are specifically useful for Artificial Intelligence and Machine Learning such as Pandas, TensorFlow, Scikit-learn, etc.
So, if you want to learn ML, it's best if you learn Python! You can do that using various online resources and courses such as **Fork Python** available Free on GeeksforGeeks.

5.2.1.3 Step 2 – Learn Various ML Concepts

Now that you are done with the prerequisites, you can move on to actually learning ML

(Which is the fun part!!!) It's best to start with the basics and then move on to the more complicated stuff. Some of the basic concepts in ML are:

5.2.1.3.1 (a) Terminologies of Machine Learning

- **Model –** A model is a specific representation learned from data by applying some machine learning algorithm. A model is also called a hypothesis.
- **Feature –** A feature is an individual measurable property of the data. A set of numeric features can be conveniently described by a feature vector. Feature vectors are fed as input to the model. For example, in order to predict a fruit, there may be features like colour, smell, taste, etc.
- **Target (Label) –** A target variable or label is the value to be predicted by our model.

For the fruit example discussed in the feature section, the label with each set of input would be the name of the fruit like apple, orange, banana, etc.

- **Training –** The idea is to give a set of inputs(features) and it's expected outputs(labels), so after training, we will have a model (hypothesis) that will then map new data to one of the categories trained on.
- **Prediction –** Once our model is ready, it can be fed a set of inputs to which it will provide a predicted output(label).

5.2.1.3.2 (b) Types of Machine Learning

- **Supervised Learning –** This involves learning from a training dataset with labeled data using classification and regression models. This learning process continues until the required level of performance is achieved.
- **Unsupervised Learning –** This involves using un labelled data and then finding the underlying structure in the data in order to learn more and more about the data itself using factor and cluster analysis models.
- **Semi-supervised Learning –** This involves using un labelled data like Unsupervised Learning with a small amount of labelled data. Using labelled data vastly increases the learning accuracy and is also more cost-effective than Supervised Learning.
- **Reinforcement Learning –** This involves learning optimal actions through trial and error. So, the next action is decided by learning behaviours that are based on the current state and that will maximize the reward in the future.

5.2.1.4 Advantages of Machine learning: -

5.2.1.4.1 1. Easily identifies trends and patterns -

Machine Learning can review large volumes of data and discover specific trends and patterns that would not be apparent to humans. For instance, for an e-commerce website like Amazon, it serves to understand the browsing behaviours and purchase histories of its users to help cater to the right products, deals, and reminders relevant to them. It uses the results to reveal relevant advertisements to them.

5.2.1.4.2 2. No human intervention needed (automation)

With ML, you don't need to babysit your project every step of the way. Since it means giving machines the ability to learn, it lets them make predictions and also improve the algorithms on their own. A common example of this is anti-virus software; they learn to filter new threats as they are recognized. ML is also good at recognizing spam.

5.2.1.4.3 3. Continuous Improvement

As **ML algorithms** gain experience, they keep improving in accuracy and efficiency. This lets them make better decisions. Say you need to make a weather forecast model. As the amount of data you have keeps growing, your algorithms learn to make more accurate predictions faster.

5.2.1.4.4 4. Handling multi-dimensional and multi-variety data

Machine Learning algorithms are good at handling data that are multi-dimensional and multi-variety, and they can do this in dynamic or uncertain environments.

5.2.1.4.5 5. Wide Applications

You could be an e-tailer or a healthcare provider and make ML work for you. Where it does apply, it holds the capability to help deliver a much more personal experience to customers while also targeting the right customers.

5.2.1.5 Disadvantages of Machine Learning: -

5.2.1.5.1 1. Data Acquisition

Machine Learning requires massive data sets to train on, and these should be inclusive/unbiased, and of good quality. There can also be times where they must wait for new data to be generated.

5.2.1.5.2 2. Time and Resources

ML needs enough time to let the algorithms learn and develop enough to fulfill their purpose with a considerable amount of accuracy and relevancy. It also needs massive resources to function. This can mean additional requirements of computer power for you.

5.2.1.5.3 3. Interpretation of Results

Another major challenge is the ability to accurately interpret results generated by the algorithms. You must also carefully choose the algorithms for your purpose.

5.2.1.5.4 4. High error-susceptibility

Machine Learning is autonomous but highly susceptible to errors. Suppose you train an algorithm with data sets small enough to not be inclusive. You end up with biased predictions coming from a biased training set. This leads to irrelevant advertisements being displayed to customers. In the case of ML, such blunders can set off a chain of errors that can go undetected for long periods of time. And when they do get noticed, it takes quite some time to recognize the source of the issue, and even longer to correct it.

Python Development Steps: -

Guido Van Rossum published the first version of Python code (version 0.9.0) at alt. sources in February 1991. This release included already exception handling, functions, and the core data types of lists, dict, str and others. It was also object oriented and had a module system.

Python version 1.0 was released in January 1994. The major new features included in this release were the functional programming tools lambda, map, filter and reduce, which Guido Van Rossum never liked. Six and a half years later in October 2000, Python 2.0 was introduced. This release included list comprehensions, a full garbage collector and it was supporting Unicode. Python flourished for another 8 years in the versions 2.x before the next major release as Python 3.0 (also known as "Python 3000" and "Py3K") was released. Python 3 is not backwards compatible with Python 2.x. The emphasis in Python 3 had been on the removal of duplicate programming constructs and modules, thus fulfilling or coming close to fulfilling the 13th law of the Zen of Python: "There should be one -- and preferably only one -- obvious way to do it." Some changes in Python 7.3:

- Print is now a function
- Views and iterators instead of lists

- The rules for ordering comparisons have been simplified. E.g. a heterogeneous list cannot be sorted, because all the elements of a list must be comparable to each other.
- There is only one integer type left, i.e. int. long is int as well.
- The division of two integers returns a float instead of an integer. "//" can be used to have the "old "behaviour.
- Text Vs. Data Instead of Unicode Vs. 8-bit

Purpose: -

We demonstrated that our approach enables successful segmentation of intra-retinal layers—even with low-quality images containing speckle noise, low contrast, and different intensity ranges throughout—with the assistance of the ANIS feature.

Python

Python is an interpreted high-level programming language for general-purpose programming. Created by Guido van Rossum and first released in 1991, Python has a design philosophy that emphasizes code readability, notably using significant whitespace.

Python features a dynamic type system and automatic memory management. It supports multiple programming paradigms, including object-oriented, imperative, functional and procedural, and has a large and comprehensive standard library.

- Python is Interpreted − Python is processed at runtime by the interpreter. You do not need to compile your program before executing it. This is similar to PERL and PHP.
- Python is Interactive − you can actually sit at a Python prompt and interact with the interpreter directly to write your programs.

Python also acknowledges that speed of development is important. Readable and terse code is part of this, and so is access to powerful constructs that avoid tedious repetition of code. Maintainability also ties into this may be an all but useless metric, but it does say something about how much code you have to scan, read and/or understand to troubleshoot problems or tweak behaviours. This speed of development, the ease with which a programmer of other languages can pick up basic Python skills and the huge standard library is key to another area where Python excels. All its tools have been quick to implement, saved a lot of time, and several of them have later been patched and updated by people with no Python background - without breaking.

Modules Used in Project: -

Num.py

Num.py is a general-purpose array-processing package. It provides a high-performance multidimensional array object, and tools for working with these arrays.

It is the fundamental package for scientific computing with Python. It contains various features including these important ones:

- A powerful N-dimensional array object

- Sophisticated (broadcasting) functions
- Tools for integrating C/C++ and Fortran code
- Useful linear algebra, Fourier transform, and random number capabilities Besides its obvious scientific uses, Num.py can also be used as an efficient multi-dimensional container of generic data. Arbitrary data-types can be defined using Num.py which allows Num.py to seamlessly and speedily integrate with a wide variety of databases.

Pandas

Pandas is an open-source Python Library providing high-performance data manipulation and analysis tool using its powerful data structures. Python was majorly used for data munging and preparation. It had very little contribution towards data analysis. Pandas solved this problem. Using Pandas, we can accomplish five typical steps in the processing and analysis of data, regardless of the origin of data load, prepare, manipulate, model, and analyse. Python with Pandas is used in a wide range of fields including academic and commercial domains including finance, economics, Statistics, analytics, etc.

Matplotlib

Matplotlib is a Python 2D plotting library which produces publication quality figures in a variety of hardcopy formats and interactive environments across platforms. Matplotlib can be used in Python scripts, the Python and Python shells, the Jupiter Notebook, web application servers, and four graphical user interface toolkits. Matplotlib tries to make easy things easy and hard things possible. You can generate plots, histograms, power spectra, bar charts, error charts, scatter plots, etc., with just a few lines of code. For examples, see the sample plots and thumbnail gallery.

For simple plotting the pyplot module provides a MATLAB-like interface, particularly when combined with Python. For the power user, you have full control of line styles, font properties, axes properties, etc, via an object-oriented interface or via a set of functions familiar to MATLAB users.

Scikit – learn

Scikit-learn provides a range of supervised and unsupervised learning algorithms via a consistent interface in Python. It is licensed under a permissive simplified BSD license and is distributed under many Linux distributions, encouraging academic and commercial use.

Python

Python is an interpreted high-level programming language for general-purpose programming. Created by Guido van Rossum and first released in 1991, Python has a design philosophy that emphasizes code readability, notably using significant whitespace.

Python features a dynamic type system and automatic memory management. It supports multiple programming paradigms, including object-oriented, imperative, functional and procedural, and has a large and comprehensive standard library.

- Python is Interpreted − Python is processed at runtime by the interpreter. You do not need to compile your program before executing it. This is similar to PERL and PHP.
- Python is Interactive − you can actually sit at a Python prompt and interact with the interpreter directly to write your programs.

Python also acknowledges that speed of development is important. Readable and terse code is part of this, and so is access to powerful constructs that avoid tedious repetition of code. Maintainability also ties into this may be an all but useless metric, but it does say something about how much code you have to scan, read and/or understand to troubleshoot problems or tweak behaviours. This speed of development, the ease with which a programmer of other languages can pick up basic Python skills and the huge standard library is key to another area where Python excels. All its tools have been quick to implement, saved a lot of time, and several of them have later been patched and updated by people with no Python background - without breaking.

Install Python Step-by-Step in Windows and Mac:

Python a versatile programming language doesn't come pre-installed on your computer devices. Python was first released in the year 1991 and until today it is a very popular high- level programming language. Its style philosophy emphasizes code readability with its notable use of great whitespace.

The object-oriented approach and language construct provided by Python enables programmers to write both clear and logical code for projects. This software does not come pre-packaged with Windows.

5.2.2 How to Install Python on Windows and Mac:

There have been several updates in the Python version over the years. The question is how to install Python? It might be confusing for the beginner who is willing to start learning Python but this tutorial will solve your query. The latest or the newest version of Python is version 3.7.4 or in other words, it is Python 3.

Note: The python version 3.7.4 cannot be used on Windows XP or earlier devices.

Before you start with the installation process of Python. First, you need to know about your **System Requirements**. Based on your system type i.e. operating system and based processor, you must download the python version. My system type is a **Windows 64-bit operating system**. So the steps below are to install python version 3.7.4 on Windows 7 device or to install Python 3. Download the Python Cheat sheet here. The steps on how to install Python on Windows 10, 8 and 7 are **divided into 4 parts** to help understand better.

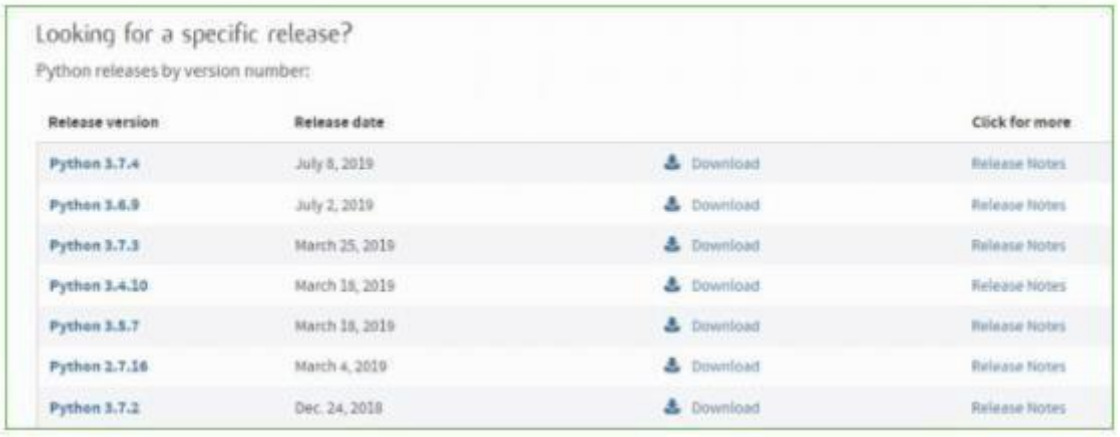
5.2.2.1 Download the Correct version into the system

Step 1: Go to the official site to download and install python using Google Chrome or any other web browser. OR Click on the following link: https://www.python.org

Now, check for the latest and the correct version for your operating system.

Step 2: Click on the Download Tab.

Step 3: You can either select the Download Python for windows 3.7.4 button in Yellow Colour or you can scroll further down and click on download with respective to their version.
Here, we are downloading the most recent python version for windows 3.7.4



Looking for a specific release?

Python releases by version number:

Release version	Release date		Click for more
Python 3.7.4	July 8, 2019	Download	Release Notes
Python 3.6.9	July 2, 2019	Download	Release Notes
Python 3.7.3	March 25, 2019	Download	Release Notes
Python 3.4.10	March 18, 2019	Download	Release Notes
Python 3.5.7	March 18, 2019	Download	Release Notes
Python 2.7.16	March 4, 2019	Download	Release Notes
Python 3.7.2	Dec. 24, 2018	Download	Release Notes

Step 4: Scroll down the page until you find the Files option.

Step 5: Here you see a different version of python along with the operating system.

Files

Version	Operating System	Description	MD5 Sum	File Size	GPG
Gzipped source tarball	Source release		6811167e5b2dfb4aef7b9ab019f0f9be	23017663	SIG
XZ compressed source tarball	Source release		d33e4aae660970511c2eca45ee3604803	17131432	SIG
macOS 64-bit/32-bit installer	Mac OS X	for Mac OS X 10.6 and later	6428b4fa7583daff1a442cbabcee08e6	34898416	SIG
macOS 64-bit installer	Mac OS X	for OS X 10.9 and later	5dd605c38217a45773bf5e4a936b2437	28982845	SIG
Windows help file	Windows		d6399957 3a2c06b2ac56cade6b4f7cd2	8131761	SIG
Windows x86-64 embeddable zip file	Windows	for AMD64/EM64T/x64	9b00c8cf6df9ec0b6abe02164a40729a2	7504391	SIG
Windows x86-64 executable installer	Windows	for AMD64/EM64T/x64	a702b4b6ad76deb6d0c304a683e563e00	26680368	SIG
Windows x86-64 web-based installer	Windows	for AMD64/EM64T/x64	28c91c6088bd73ae8e53a3bd3516b4bd2	1362904	SIG
Windows x86 embeddable zip file	Windows		9fab38d1f8841b79fda94113574139d8	6741626	SIG
Windows x86 executable installer	Windows		33cc802942a54446a3d04514763947f89	25663848	SIG
Windows x86 web-based installer	Windows		1b670cfa5d117df82c3093ea371d87c	1324608	SIG

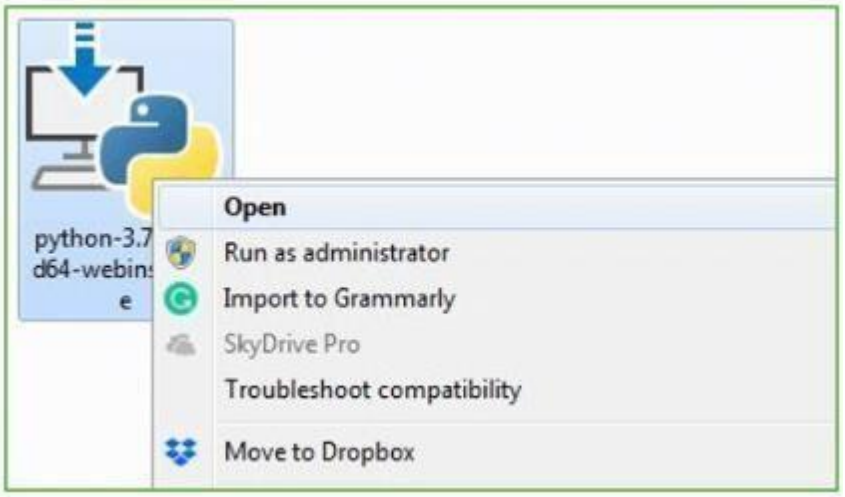
- To download **Windows 32-bit python**, you can select any one from the three options: Windows x86 embeddable zip file, Windows x86 executable installer or Windows x86 web-based installer.
- To download **Windows 64-bit python**, you can select any one from the three options: Windows x86-64 embeddable zip file, Windows x86-64 executable installer or Windows x86-64 web-based installer.

Here we will install Windows x86-64 web-based installer. Here your first part regarding which version of python is to be downloaded is completed. Now we move ahead with the second part in installing python i.e. Installation

Note: To know the changes or updates that are made in the version you can click on the Release Note Option.

5.2.2.2 Installation of Python

Step 1: Go to Download and Open the downloaded python version to carry out the installation process.



Step 2: Before you click on Install Now, make sure to put a tick on Add Python 3.7 to PATH.

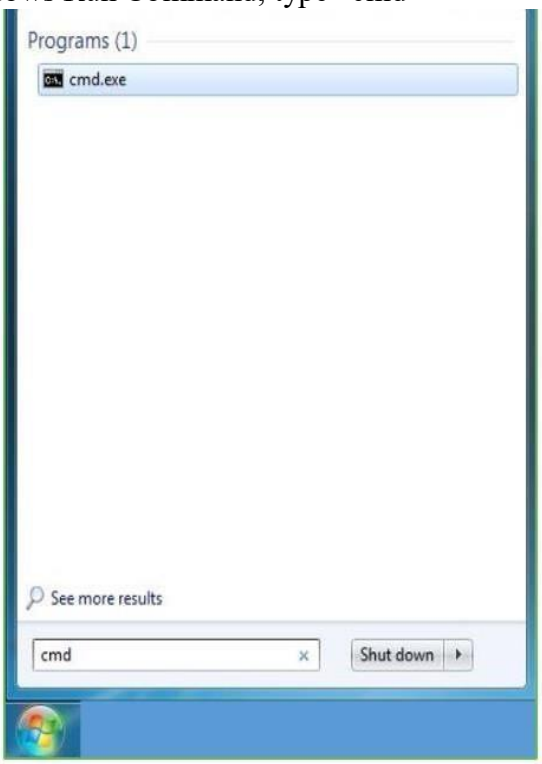
Step 3: Click on Install NOW After the installation is successful. Click on Close.

With these above three steps on python installation, you have successfully and correctly installed Python. Now is the time to verify the installation. **Note:** The installation process might take a couple of minutes.

5.2.2.3 Verify the Python Installation

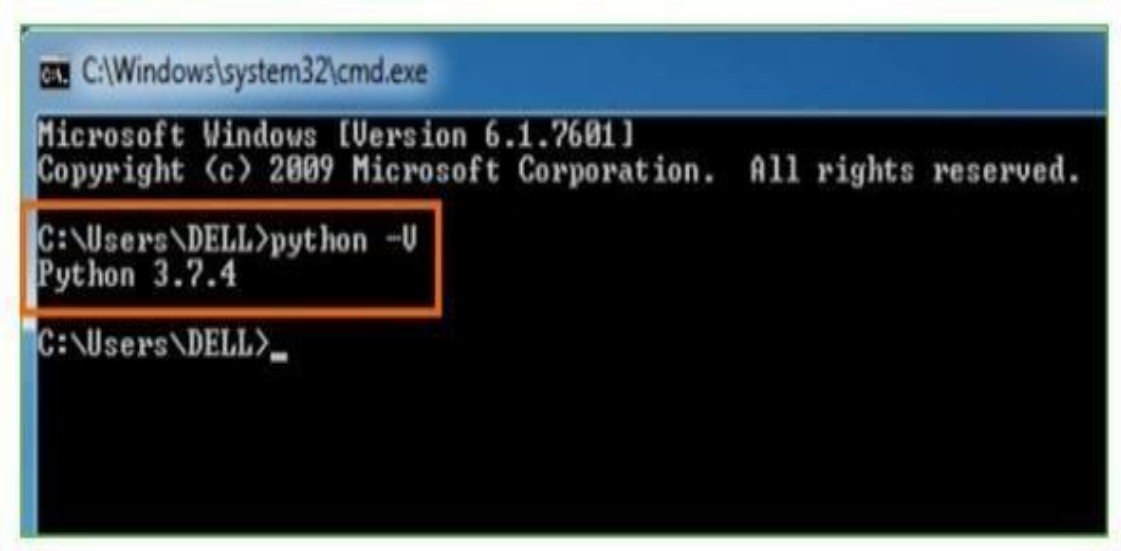
Step 1: Click on Start

Step 2: In the Windows Run Command, type "cmd"



Step 3: Open the Command prompt option.

Step 4: Let us test whether the python is correctly installed. Type **python –V** and press Enter.



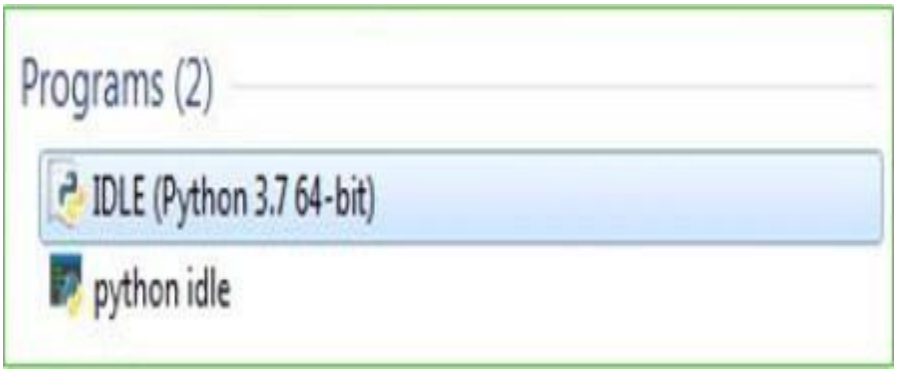
Step 5: You will get the answer as 3.7.4

Note: If you have any of the earlier versions of Python already installed. You must first uninstall the earlier version and then install the new one.

5.2.2.4 Check how the Python IDLE works

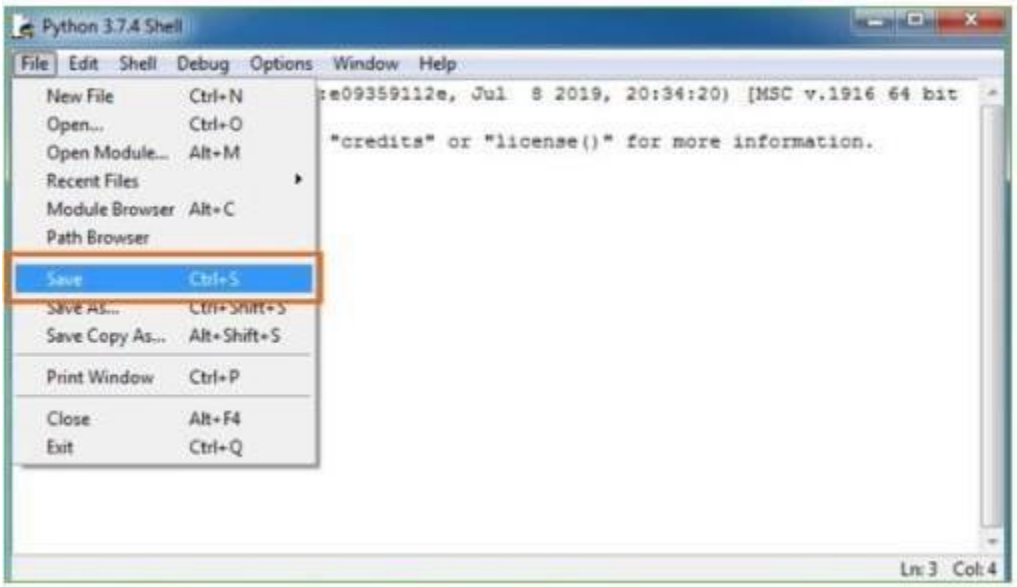
Step 1: Click on Start

Step 2: In the Windows Run command, type "python idle"

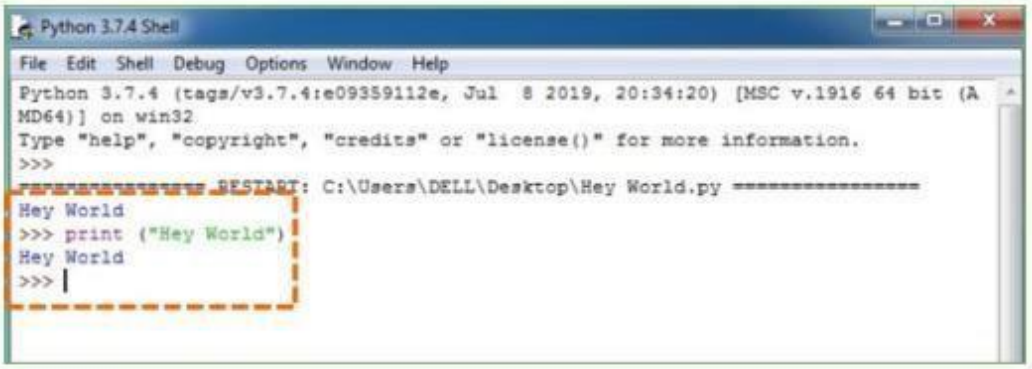


Step 3: Click on IDLE (Python 3.7 64-bit) and launch the program

Step 4: To go ahead with working in IDLE you must first save the file. **Click on File > Click on Save**



Step 5: Name the file and save as type should be Python files. Click on SAVE. Here I have named the files as Hey World.
Step 6: Now for e.g. **enter print ("Hey World")** and Press Enter.



You will see that the command given is launched. With this, we end our tutorial on how to install Python. You have learned how to download python for windows into your respective operating system.

5.2.3 **Jupiter Notebook**

The Jupiter Notebook is an open-source web application that you can use to create and share documents that contain live code, equations, visualizations, and text. Jupiter Notebook is maintained by the people at Project Jupiter.

Jupiter Notebooks are a spin-off project from the Python project, which used to have an Python Notebook project itself. The name, Jupiter, comes from the core supported programming languages that it supports: Julia, Python, and R. Jupiter ships with the Python kernel, which allows you to write your programs in Python, but there are currently over 100 other kernels that you can also use

5.4 Google colab

1. *Access Google Colab*: Go to the Google Colab website and sign in with your Google account.

2. *Create a New Notebook*: Click on the "New Notebook" button to create a new Colab notebook. This notebook will serve as your project workspace.

3. *Set Runtime Type*: Choose the desired runtime type (CPU, GPU, or TPU) by navigating to "Runtime" -> "Change runtime type". GPUs and TPUs can accelerate computation for machine learning tasks.

4. *Import Libraries*: Import any necessary libraries for your project by adding code cells and using commands like! pip install to install libraries not already available in the Colab environment.

5. *Write Code*: Write your project code in individual code cells. You can add text descriptions or explanations using Markdown cells.

6. *Execute Code*: Execute each code cell by clicking the play button or pressing Shift+Enter. Colab will execute the code and display the output.

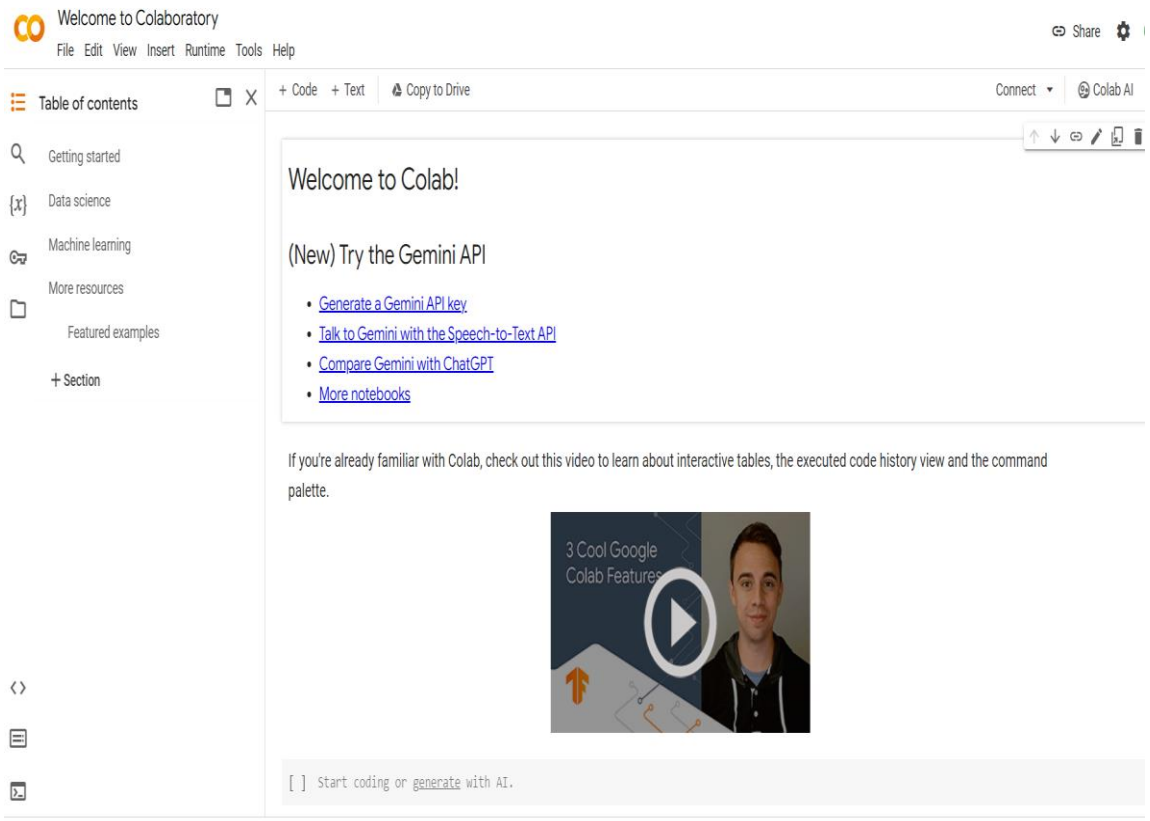
7. *Access Data*: If your project requires data, you can upload it directly to Colab from your local machine using the file upload feature or access data from Google Drive.

8. *Save and Share*: Save your progress by clicking on "File" -> "Save" or using Ctrl+S. You can share your project with collaborators by providing them with the notebook's shareable link.

9. *Export Results*: Once your project is complete, you can export the results by downloading the notebook as an IPython (. ipynb) file or saving visualizations and outputs to your Google Drive or local machine.

10. *Shutdown Notebook*: Remember to shut down your notebook when you're finished to release resources. You can do this by selecting "Runtime" -> "Shutdown all runtimes".

Following these steps will enable you to efficiently conduct your project in Google Colab, leveraging its cloud-based infrastructure for computational tasks.



5.5 RESULTS:

5.5.1 SOURCE CODE

5.5.1.1: Importing Datasets:

Importing the dataset using the below code:

```
import pandas as pd
import numpy as np

df = pd.read_csv('May-2022.csv')
df1=pd.read_csv('Amazon Sale Report.csv')
df2=pd.read_csv('International sale Report.csv')
s=df.copy()
```

```
<ipython-input-14-7b960d04416e>:5: DtypeWarning: Columns (21,23) have mixed types. Specify dtype option on import or set low_memory=False.
  df1=pd.read_csv('Amazon Sale Report.csv')
```

Checking for head values:

```
df1.head()
```

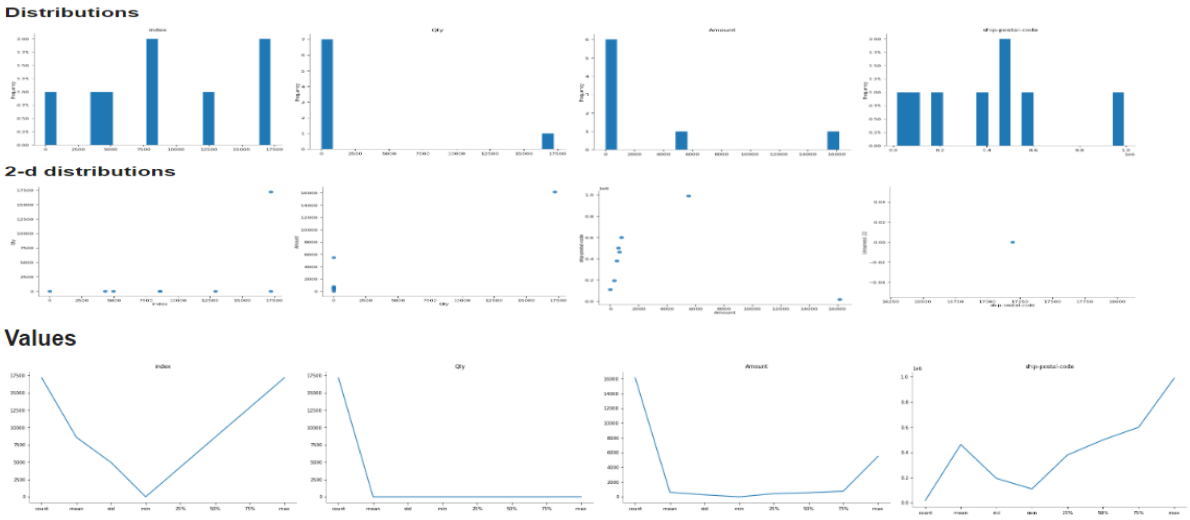
	index	Order ID	Date	Status	Fulfilment	Sales Channel	ship-service-level	Style	SKU	Category	...	currency	Amount	ship-city	ship-state
0	0	405-8078784-5731545	04-30-22	Cancelled	Merchant	Amazon.in	Standard	SET389	SET389-KR-NP-S	Set	...	INR	647.62	MUMBAI	MAHARASHTRA
1	1	171-9198151-1101146	04-30-22	Shipped - Delivered to Buyer	Merchant	Amazon.in	Standard	JNE3781	JNE3781-KR-XXXL	kurta	...	INR	406.00	BENGALURU	KARNATAKA
2	2	404-0687676-7273146	04-30-22	Shipped	Amazon	Amazon.in	Expedited	JNE3371	JNE3371-KR-XL	kurta	...	INR	329.00	NAVI MUMBAI	MAHARASHTRA
3	3	403-9615377-8133951	04-30-22	Cancelled	Merchant	Amazon.in	Standard	J0341	J0341-DR-L	Western Dress	...	INR	753.33	PUDUCHERRY	PUDUCHERRY

ship-postal-code	ship-country	promotion-ids	B2B	fulfilled-by	Unnamed: 22
400081.0	IN	NaN	False	Easy Ship	NaN
560085.0	IN	Amazon PLCC Free-Financing Universal Merchant ...	False	Easy Ship	NaN
410210.0	IN	IN Core Free Shipping 2015/04/08 23-48-5-108	True	NaN	NaN
605008.0	IN	NaN	False	Easy Ship	NaN

Checking for describe:

```
df1.describe()
```

	index	Qty	Amount	ship-postal-code	Unnamed: 22
count	17200.000000	17199.000000	16163.000000	17192.000000	0.0
mean	8599.500000	0.900343	622.710565	462592.747092	NaN
std	4965.356651	0.349038	271.931388	194376.249692	NaN
min	0.000000	0.000000	0.000000	110001.000000	NaN
25%	4299.750000	1.000000	435.000000	380015.000000	NaN
50%	8599.500000	1.000000	568.000000	500020.000000	NaN
75%	12899.250000	1.000000	759.000000	600018.000000	NaN
max	17199.000000	15.000000	5495.000000	989898.000000	NaN



To identify info:

```
[ ]  df1.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 3991 entries, 0 to 3990
Data columns (total 24 columns):
 #   Column              Non-Null Count  Dtype  
---  ------              --------------  -----  
 0   index               3991 non-null   int64  
 1   Order ID            3991 non-null   object 
 2   Date                3990 non-null   object 
 3   Status              3990 non-null   object 
 4   Fulfilment          3990 non-null   object 
 5   Sales Channel       3990 non-null   object 
 6   ship-service-level  3990 non-null   object 
 7   Style               3990 non-null   object 
 8   SKU                 3990 non-null   object 
 9   Category            3990 non-null   object 
 10  Size                3990 non-null   object 
 11  ASIN                3990 non-null   object 
 12  Courier Status      3804 non-null   object 
 13  Qty                 3990 non-null   float64
 14  currency            3718 non-null   object 
 15  Amount              3718 non-null   float64
 16  ship-city           3988 non-null   object 
 17  ship-state          3988 non-null   object 
 18  ship-postal-code    3988 non-null   float64
 19  ship-country        3988 non-null   object 
 20  promotion-ids       2570 non-null   object 
 21  B2B                 3990 non-null   object 
 22  fulfilled-by        1057 non-null   object 
 23  Unnamed: 22         0 non-null      float64
dtypes: float64(4), int64(1), object(19)
memory usage: 748.4+ KB
```

To identify null values:

```
df1.isnull().sum()
```

```
index                    0
Order ID                 0
Date                     0
Status                   0
Fulfilment               0
Sales Channel            0
ship-service-level       0
Style                    0
SKU                      0
Category                 0
Size                     0
ASIN                     0
Courier Status        6872
Qty                      0
currency              7795
Amount                7795
ship-city               33
ship-state              33
ship-postal-code        33
ship-country            33
promotion-ids        49153
B2B                      0
fulfilled-by         89698
Unnamed: 22          49050
```

After applying dropna we get:

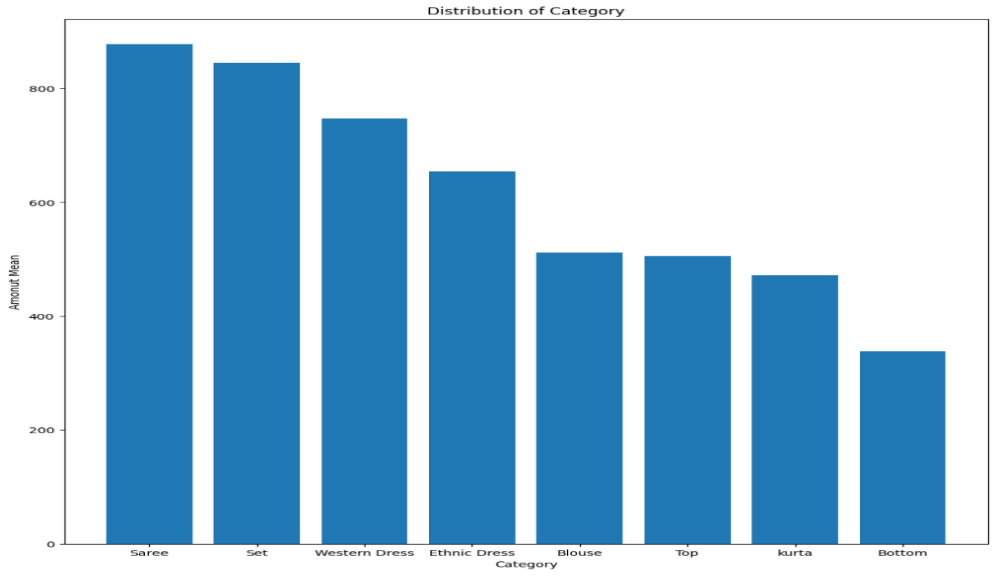
```
df1 = df1.dropna()
df.isnull().sum()
```

```
index             0
Sku               0
Style Id          0
Catalog           0
Category          0
Weight            0
TP                0
MRP Old           0
Final MRP Old     0
Ajio MRP          0
Amazon MRP        0
Amazon FBA MRP    0
Flipkart MRP      0
Limeroad MRP      0
Myntra MRP        0
Paytm MRP         0
Snapdeal MRP      0
dtype: int64
```

Visualizing Distribution of Category Mean in Data Frame:

```
[17] category_mean = df1.groupby("Category")["Amount"].mean().sort_values(ascending=False)
```

```
[18] import matplotlib.pyplot as plt
     plt.figure(figsize=[12,12])
     plt.bar(category_mean.index, category_mean.values)
     plt.title('Distribution of Category')
     plt.xlabel('Category')
     plt.ylabel('Amonut Mean')
     plt.show()
```



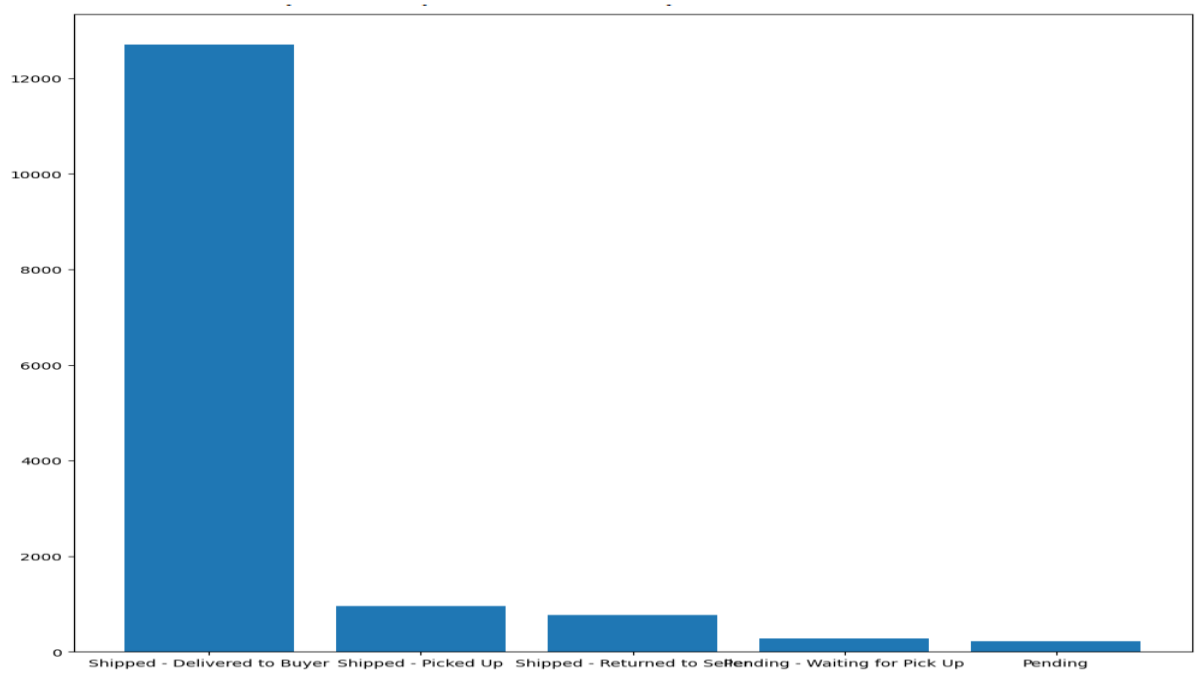
Identifying Unique Values:

```
df1.nunique()
```

```
Order ID                17993
Date                       61
Status                     10
Fulfilment                  1
Sales Channel               1
ship-service-level          1
Style                    1025
SKU                      3880
Category                    8
Size                       11
ASIN                     3878
Courier Status              2
Qty                         5
currency                    1
Amount                    594
ship-city                3080
ship-state                 52
ship-postal-code         4566
ship-country                1
promotion-ids            4012
B2B                         2
fulfilled-by                1
Unnamed: 22                 1
dtype: int64
```

Visualizing Top 5 Status Distribution in Data Frame with a Bar Chart:

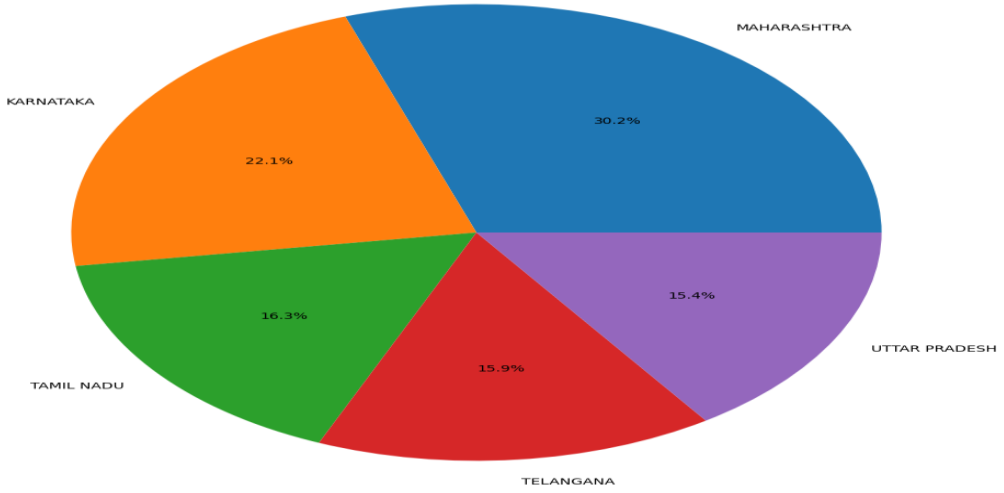
```
[ ]  status_counts = df1['Status'].value_counts().head()
     plt.figure(figsize=[12, 12])
     plt.bar(status_counts.index, status_counts.values)
```



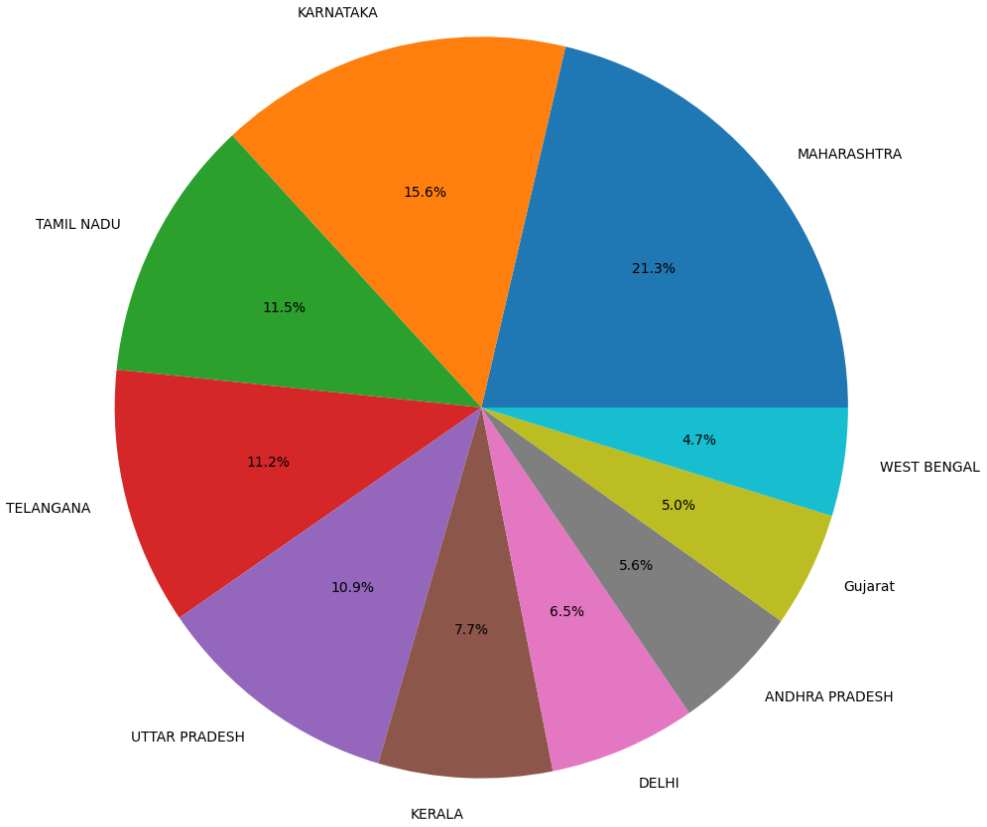
Visualizing Top 5 Shipping States Distribution with a Pie Chart:

```
[ ]  state_counts = df1['ship-state'].value_counts().head(5)
     plt.figure(figsize=[12, 12])
     plt.pie(state_counts.values, labels=state_counts.index, autopct="%1.1f%%")
```

```
([<matplotlib.patches.Wedge at 0x784d82ba9570>,
  <matplotlib.patches.Wedge at 0x784d82ba9450>,
  <matplotlib.patches.Wedge at 0x784d82baa140>,
  <matplotlib.patches.Wedge at 0x784d82baa7d0>,
  <matplotlib.patches.Wedge at 0x784d82baae60>],
 [Text(0.6396747132272632, 0.8948833785793648, 'MAHARASHTRA'),
  Text(-0.9396073604066562, 0.5719597960255917, 'KARNATAKA'),
  Text(-0.8692776677270277, -0.674059594094661, 'TAMIL NADU'),
  Text(0.1113652751060096, -1.0943481052665842, 'TELANGANA'),
  Text(0.9734653936152555, -0.5122158992392718, 'UTTAR PRADESH')],
 [Text(0.3489134799421435, 0.4881182064978353, '30.2%'),
  Text(-0.5125131056763579, 0.31197807055941357, '22.1%'),
  Text(-0.4741514551238333, -0.3676688695061787, '16.3%'),
  Text(0.06074469551236887, -0.5969171483272278, '15.9%'),
  Text(0.5309811237901393, -0.2793904904941482, '15.4%')])
```

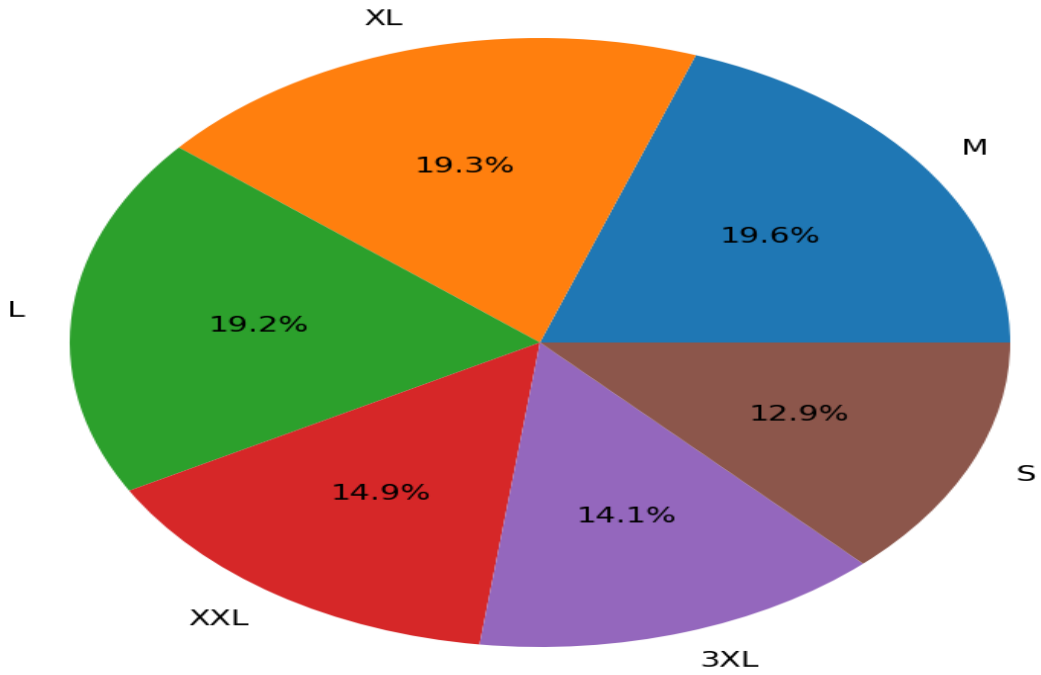



```
state_counts = df1['ship-state'].value_counts().head(10)
plt.figure(figsize=[12, 12])
plt.pie(state_counts.values, labels=state_counts.index, autopct="%1.1f%%")
```



Visualizing Top 6 Sizes Distribution in DataFrame with a Pie Chart:

```
# prompt: size pie chart

state_counts = df1['Size'].value_counts().head(6)
plt.figure(figsize=[10, 10])
plt.pie(state_counts.values, labels=state_counts.index, autopct="%1.1f%%", textprops={'fontsize': 15})
plt.show()
```



Checking for head values:

```
df.head()
```

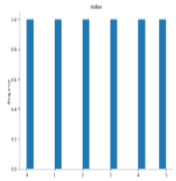
	index	Sku	Style Id	Catalog	Category	Weight	TP	MRP Old	Final MRP Old	Ajio MRP	Amazon MRP	Amazon FBA MRP	Flipkart MRP	Limeroad MRP	Myntra MRP	Paytm MRP	Snapdeal MRP
0	0	Os206_3141_S	Os206_3141	Moments	Kurta	0.3	538	2178	2295	2295	2295	2295	2295	2295	2295	2295	2295
1	1	Os206_3141_M	Os206_3141	Moments	Kurta	0.3	538	2178	2295	2295	2295	2295	2295	2295	2295	2295	2295
2	2	Os206_3141_L	Os206_3141	Moments	Kurta	0.3	538	2178	2295	2295	2295	2295	2295	2295	2295	2295	2295
3	3	Os206_3141_XL	Os206_3141	Moments	Kurta	0.3	538	2178	2295	2295	2295	2295	2295	2295	2295	2295	2295
4	4	Os206_3141_2XL	Os206_3141	Moments	Kurta	0.3	538	2178	2295	2295	2295	2295	2295	2295	2295	2295	2295

Checking for head values:

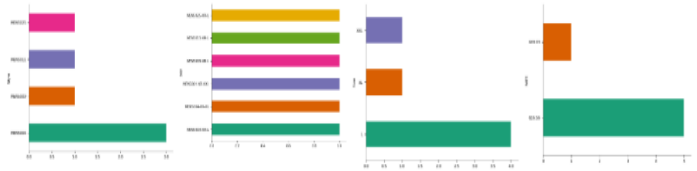
```
df2.head(6)
```

	index	DATE	Months	CUSTOMER	Style	SKU	Size	PCS	RATE	GROSS AMT
0	0	06-05-21	Jun-21	REVATHY LOGANATHAN	MEN5004	MEN5004-KR-L	L	1.00	616.56	617.00
1	1	06-05-21	Jun-21	REVATHY LOGANATHAN	MEN5004	MEN5004-KR-XL	XL	1.00	616.56	617.00
2	2	06-05-21	Jun-21	REVATHY LOGANATHAN	MEN5004	MEN5004-KR-XXL	XXL	1.00	616.56	617.00
3	3	06-05-21	Jun-21	REVATHY LOGANATHAN	MEN5009	MEN5009-KR-L	L	1.00	616.56	617.00
4	4	06-05-21	Jun-21	REVATHY LOGANATHAN	MEN5011	MEN5011-KR-L	L	1.00	616.56	617.00
5	5	06-05-21	Jun-21	REVATHY LOGANATHAN	MEN5025	MEN5025-KR-L	L	1.00	649.03	649.00

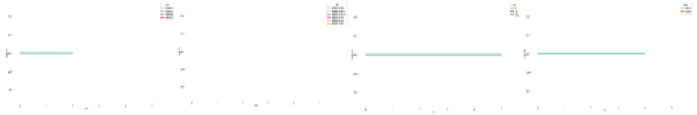
Distributions



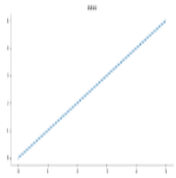
Categorical distributions



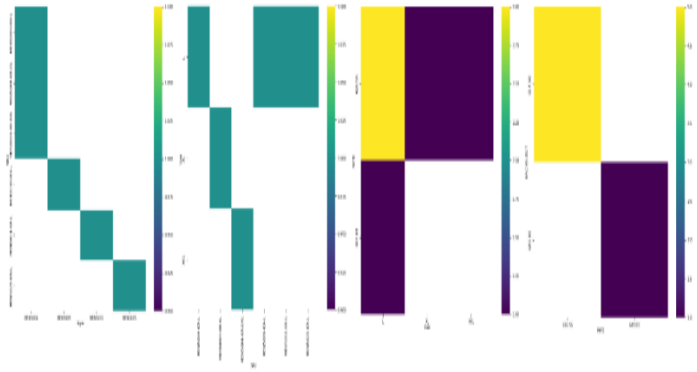
Time series



Values



2-d categorical distributions

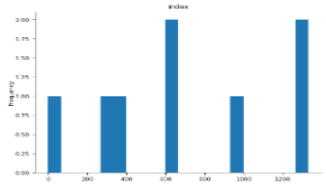


Checking for describe:

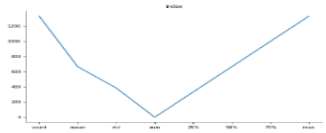
```
df.describe()
```

	index
count	1330.000000
mean	664.500000
std	384.082239
min	0.000000
25%	332.250000
50%	664.500000
75%	996.750000
max	1329.000000

Distributions



Values



Displaying Data Types of Columns:

```
[ ] df.dtypes
```

```
index            int64
Sku             object
Style Id        object
Catalog         object
Category        object
Weight          object
TP              object
MRP Old         object
Final MRP Old   object
Ajio MRP        object
Amazon MRP      object
Amazon FBA MRP  object
Flipkart MRP    object
Limeroad MRP    object
Myntra MRP      object
Paytm MRP       object
Snapdeal MRP    object
dtype: object
```

Identifying tail Values:

```
[ ] df.tail()
```

	index	Sku	Style Id	Catalog	Category	Weight	TP	MRP Old	Final MRP Old	Ajio MRP	Amazon MRP	Amazon FBA MRP	Flipkart MRP	Limeroad MRP	Myntra MRP	Paytm MRP	Snapdeal MRP
1325	1325	Os326_M	Os326	Mix	Kurta	0.3	467	1878	1995	1995	1995	1995	1995	1995	1995	1995	1995
1326	1326	Os326_L	Os326	Mix	Kurta	0.3	467	1878	1995	1995	1995	1995	1995	1995	1995	1995	1995
1327	1327	Os326_XL	Os326	Mix	Kurta	0.3	467	1878	1995	1995	1995	1995	1995	1995	1995	1995	1995
1328	1328	Os326_2XL	Os326	Mix	Kurta	0.3	467	1878	1995	1995	1995	1995	1995	1995	1995	1995	1995
1329	1329	Os326_3XL	Os326	Mix	Kurta	0.3	467	1878	1995	1995	1995	1995	1995	1995	1995	1995	1995

Identifying Unique Values:

```
df.nunique()
```

```
index             1330
Sku               1330
Style Id           254
Catalog              9
Category             5
Weight               4
TP                  93
MRP Old             67
Final MRP Old       53
Ajio MRP            52
Amazon MRP          53
Amazon FBA MRP      53
Flipkart MRP        52
Limeroad MRP        52
Myntra MRP          51
Paytm MRP           52
Snapdeal MRP        52
dtype: int64
```

Converting Categorical Variables to Numerical Codes in DataFrame 'df':

```
[ ] for c in df.columns:
      if df[c].dtype == 'object':
        df[c]=df[c].astype('category')
        df[c]=df[c].cat.codes
    df.head()
```

	index	Sku	Style Id	Catalog	Category	Weight	TP	MRP Old	Final MRP Old	Ajio MRP	Amazon MRP	Amazon FBA MRP	Flipkart MRP	Limeroad MRP	Myntra MRP	Paytm MRP	Snapdeal MRP
0	0	823	161	5	1	1	48	20	17	17	17	17	17	17	18	17	17
1	1	822	161	5	1	1	48	20	17	17	17	17	17	17	18	17	17
2	2	821	161	5	1	1	48	20	17	17	17	17	17	17	18	17	17
3	3	824	161	5	1	1	48	20	17	17	17	17	17	17	18	17	17
4	4	819	161	5	1	1	48	20	17	17	17	17	17	17	18	17	17

Displaying Data Types of Columns:

```
[ ] df.dtypes
```

```
index            int64
Sku              int16
Style Id         int16
Catalog           int8
Category          int8
Weight            int8
TP                int8
MRP Old           int8
Final MRP Old     int8
Ajio MRP          int8
Amazon MRP        int8
Amazon FBA MRP    int8
Flipkart MRP      int8
Limeroad MRP      int8
Myntra MRP        int8
Paytm MRP         int8
Snapdeal MRP      int8
dtype: object
```

Identifying correlation matrix of a DataFrame:

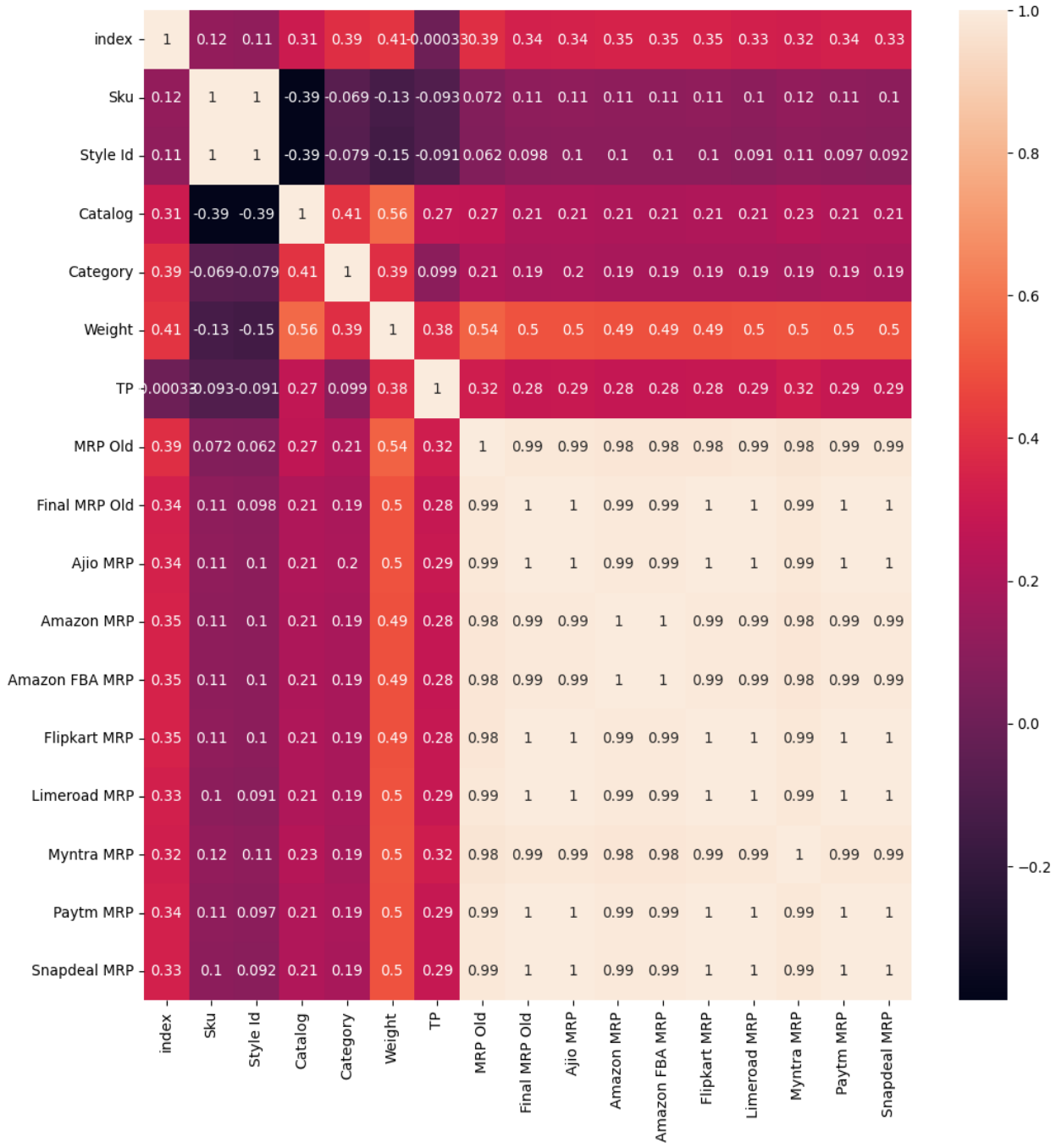
```
cor = df.corr()
cor
```

	index	Sku	Style Id	Catalog	Category	Weight	TP	MRP Old	Final MRP Old	Ajio MRP	Amazon MRP	Amazon FBA MRP	Flipkart MRP	Limeroad MRP	Myntra MRP	Paytm MRP
index	1.000000	0.121390	0.112100	0.311095	0.388027	0.413943	-0.000330	0.388367	0.338122	0.338623	0.348791	0.348791	0.346647	0.332776	0.324951	0.336559
Sku	0.121390	1.000000	0.998198	-0.385325	-0.068612	-0.131123	-0.093075	0.072113	0.107586	0.110764	0.108951	0.108951	0.110851	0.099863	0.118450	0.105976
Style Id	0.112100	0.998198	1.000000	-0.388441	-0.079234	-0.145485	-0.090871	0.061997	0.098245	0.101544	0.099857	0.099857	0.101738	0.090788	0.108682	0.096797
Catalog	0.311095	-0.385325	-0.388441	1.000000	0.405856	0.562801	0.272484	0.265417	0.210726	0.210272	0.207882	0.207882	0.212173	0.213499	0.234051	0.215000
Category	0.388027	-0.068612	-0.079234	0.405856	1.000000	0.389720	0.098874	0.214763	0.192689	0.198362	0.189549	0.189549	0.188187	0.187940	0.185726	0.191923
Weight	0.413943	-0.131123	-0.145485	0.562801	0.389720	1.000000	0.378175	0.539426	0.500852	0.500413	0.493782	0.493782	0.493521	0.496443	0.501165	0.499320
TP	-0.000330	-0.093075	-0.090871	0.272484	0.098874	0.378175	1.000000	0.317106	0.284603	0.290038	0.279014	0.279014	0.284713	0.286681	0.315541	0.285720
MRP Old	0.388367	0.072113	0.061997	0.265417	0.214763	0.539426	0.317106	1.000000	0.985802	0.986256	0.979535	0.979535	0.983610	0.985644	0.978042	0.985975
Final MRP Old	0.338122	0.107586	0.098245	0.210726	0.192689	0.500852	0.284603	0.985802	1.000000	0.998704	0.991429	0.991429	0.995935	0.998351	0.986688	0.999036
Ajio MRP	0.338623	0.110764	0.101544	0.210272	0.198362	0.500413	0.290038	0.986256	0.998704	1.000000	0.991662	0.991662	0.996147	0.998489	0.987523	0.999176
Amazon MRP	0.348791	0.108951	0.099857	0.207882	0.189549	0.493782	0.279014	0.979535	0.991429	0.991662	1.000000	1.000000	0.994017	0.990914	0.980793	0.991759
Amazon FBA MRP	0.348791	0.108951	0.099857	0.207882	0.189549	0.493782	0.279014	0.979535	0.991429	0.991662	1.000000	1.000000	0.994017	0.990914	0.980793	0.991759
Flipkart MRP	0.346647	0.110851	0.101738	0.212173	0.188187	0.493521	0.284713	0.983610	0.995935	0.996147	0.994017	0.994017	1.000000	0.995930	0.986013	0.996672
Limeroad MRP	0.332776	0.099863	0.090788	0.213499	0.187940	0.496443	0.286681	0.985644	0.998351	0.998489	0.990914	0.990914	0.995930	1.000000	0.986905	0.999177
Myntra MRP	0.324951	0.118450	0.108682	0.234051	0.185726	0.501165	0.315541	0.978042	0.986688	0.987523	0.980793	0.980793	0.986013	0.986905	1.000000	0.987520
Paytm MRP	0.336559	0.105976	0.096797	0.215000	0.191923	0.499320	0.285720	0.985975	0.999036	0.999176	0.991759	0.991759	0.996672	0.999177	0.987520	1.000000

Visualizing Correlation Matrix with Seaborn Heatmap:

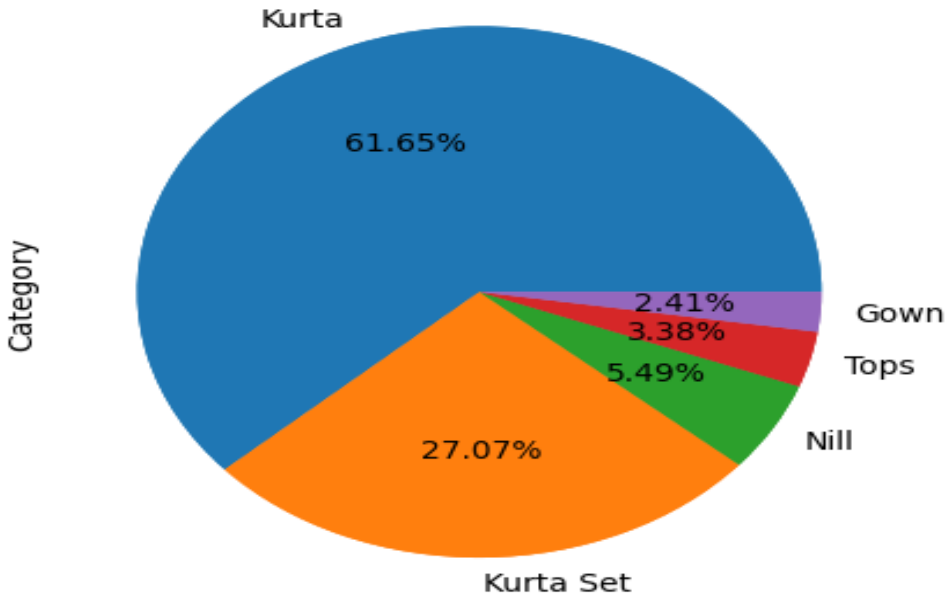
```
import seaborn as sns
import matplotlib.pyplot as plt

plt.figure(figsize=[12,12])
sns.heatmap(cor,annot=True)
```

f



Comparison of Category Distribution in DataFrame 's' with a Pie Chart:

```
plt.pie(s['Category'].value_counts(),autopct="%2.2f%%",rotatelabels=Tru
e)
s['Category'].value_counts().plot(kind='pie')
```

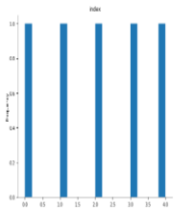


Checking for head values:

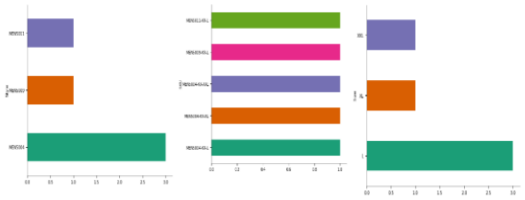
```
df2.head()
```

	index	DATE	Months	CUSTOMER	Style	SKU	Size	PCS	RATE	GROSS AMT
0	0	06-05-21	Jun-21	REVATHY LOGANATHAN	MEN5004	MEN5004-KR-L	L	1.00	616.56	617.00
1	1	06-05-21	Jun-21	REVATHY LOGANATHAN	MEN5004	MEN5004-KR-XL	XL	1.00	616.56	617.00
2	2	06-05-21	Jun-21	REVATHY LOGANATHAN	MEN5004	MEN5004-KR-XXL	XXL	1.00	616.56	617.00
3	3	06-05-21	Jun-21	REVATHY LOGANATHAN	MEN5009	MEN5009-KR-L	L	1.00	616.56	617.00
4	4	06-05-21	Jun-21	REVATHY LOGANATHAN	MEN5011	MEN5011-KR-L	L	1.00	616.56	617.00

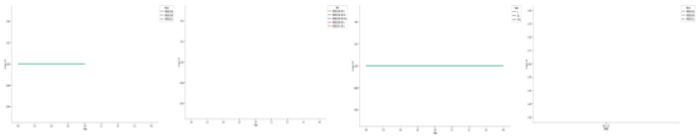
Distributions



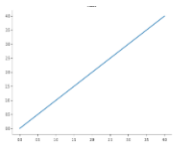
Categorical distributions



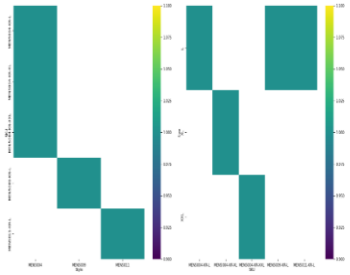
Time series



Values



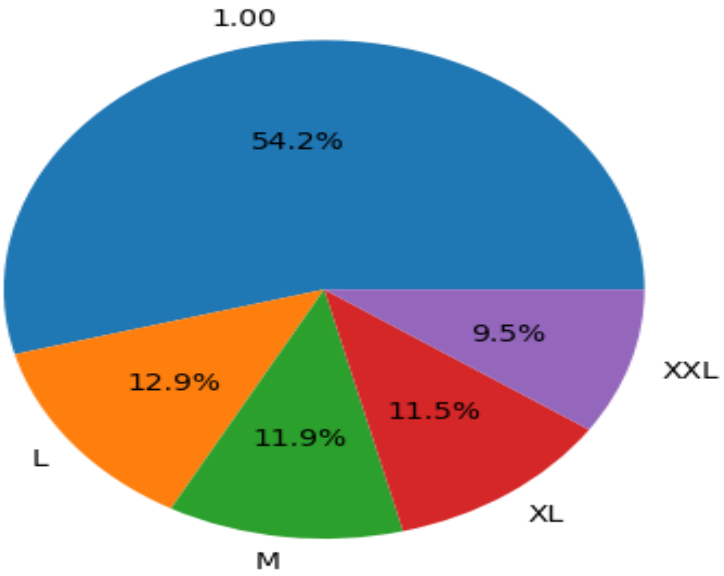
2-d categorical distributions



Visualizing Size Distribution in DataFrame df2 with a Pie Chart:

```
sizes = df2['Size'].value_counts().head()
total_stock = sizes.sum()
#print(sizes.head())
sizes_percentage = sizes / total_stock * 100
plt.pie(sizes, labels=sizes.index, autopct='%1.1f%%')
```

```
([<matplotlib.patches.Wedge at 0x784d79a3ae90>,
  <matplotlib.patches.Wedge at 0x784d79a3ad70>,
  <matplotlib.patches.Wedge at 0x784d79a3ba30>,
  <matplotlib.patches.Wedge at 0x784d79a6c130>,
  <matplotlib.patches.Wedge at 0x784d79a6c7c0>],
 [Text(-0.1441147267451587, 1.0905186589578229, '1.00'),
  Text(-0.8638921257460618, -0.6809481588740444, 'L'),
  Text(-0.13683194659348338, -1.0914563749373762, 'M'),
  Text(0.6313629244622001, -0.9007668164483738, 'XL'),
  Text(1.0513223151945028, -0.3236068441397223, 'XXL')],
 [Text(-0.07860803277008656, 0.5948283594315397, '54.2%'),
  Text(-0.47121388677057907, -0.3714262684767515, '12.9%'),
  Text(-0.07463560723280911, -0.5953398408749324, '11.9%'),
  Text(0.3443797769793819, -0.49132735442638564, '11.5%'),
  Text(0.5734485355606379, -0.17651282407621216, '9.5%')])
```



Checking for null values:

```
df2.isnull().sum()
```

```
index         0
DATE          1
Months       25
CUSTOMER   1040
Style      1040
SKU        2474
Size       1040
PCS        1040
RATE       1040
GROSS AMT  1040
dtype: int64
```

```
[ ] df2 =df2.dropna()
```

```
[ ] df2.isnull().sum()
```

```
index      0
DATE       0
Months     0
CUSTOMER   0
Style      0
SKU        0
Size       0
PCS        0
RATE       0
GROSS AMT  0
dtype: int64
```

```
[ ] df2.dtypes
```

```
index        int64
DATE        object
Months      object
CUSTOMER    object
Style       object
SKU         object
Size        object
PCS         object
RATE        object
GROSS AMT   object
dtype: object
```

To identify info:

```
[ ] df2.info()
```

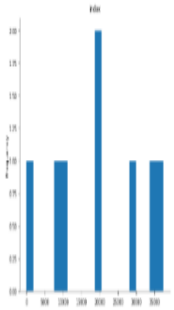
```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 34958 entries, 0 to 37431
Data columns (total 10 columns):
 #   Column     Non-Null Count  Dtype 
---  ------     --------------  ----- 
 0   index      34958 non-null  int64 
 1   DATE       34958 non-null  object
 2   Months     34958 non-null  object
 3   CUSTOMER   34958 non-null  object
 4   Style      34958 non-null  object
 5   SKU        34958 non-null  object
 6   Size       34958 non-null  object
 7   PCS        34958 non-null  object
 8   RATE       34958 non-null  object
 9   GROSS AMT  34958 non-null  object
dtypes: int64(1), object(9)
memory usage: 2.9+ MB
```

Checking for describe:

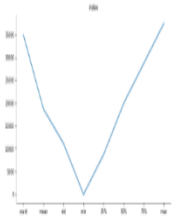
```
df2.describe()
```

	index
count	34958.000000
mean	18734.072430
std	11174.559554
min	0.000000
25%	8750.250000
50%	19935.500000
75%	28685.750000
max	37431.000000

Distributions



Values

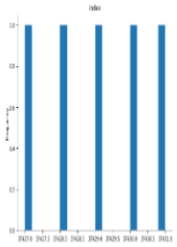


To identify tail value:

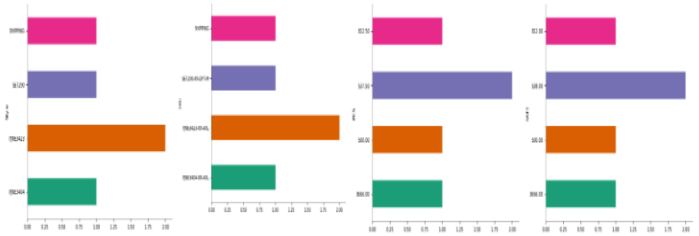
```
df2.tail()
```

	index	DATE	Months	CUSTOMER	Style	SKU	Size	PCS	RATE	GROSS AMT
37427	37427	AVIN	03-31-22	Mar-22	PJNE3423	PJNE3423-KR-4XL	1.00	537.50	538.00	4.00
37428	37428	AVIN	03-31-22	Mar-22	PJNE3404	PJNE3404-KR-4XL	1.00	500.00	500.00	5.00
37429	37429	AVIN	03-31-22	Mar-22	PJNE3423	PJNE3423-KR-4XL	1.00	537.50	538.00	4.00
37430	37430	AVIN	03-31-22	Mar-22	SET290	SET290-KR-DPT-M	1.00	812.50	812.00	7.00
37431	37431	AVIN	03-31-22	Mar-22	SHIPPING	SHIPPING	1.00	3666.00	3666.00	0.00

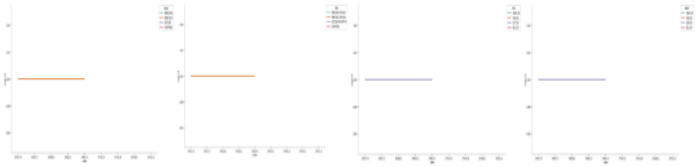
Distributions



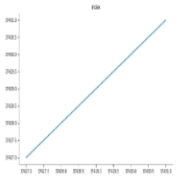
Categorical distributions



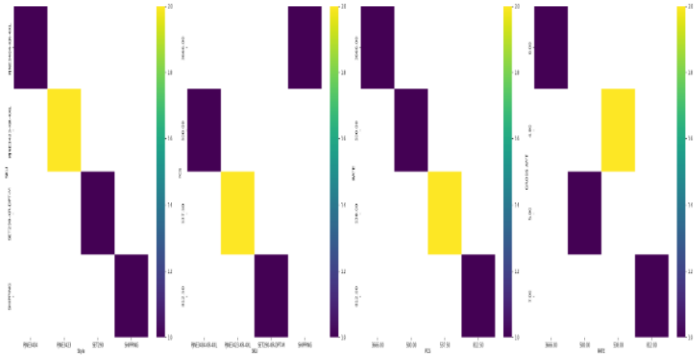
Time series



Values



2-d categorical distributions



To identify unique values:

```
[ ] df2.nunique()
```

```
index       34958
DATE          264
Months        149
CUSTOMER      136
Style         996
SKU          4598
Size           30
PCS           658
RATE         1288
GROSS AMT    1118
dtype: int64
```

CHAPTER - 6

SYSTEM TESTING

Testing is that the debugging program is one amongst the leading crucial aspects of the pc programming triggers, while not programming that works, the system would ne'er turn out relate in Nursing output of the at it had been designed. Testing is best performed once user development is asked to help in characteristic all errors and bugs. The sample knowledge is used for testing. It is not amounting however quality of the information used the matters of testing. Testing is aimed toward guaranteeing that the system was accurately relate in Nursing with efficiency before live operation commands.

Testing objectives: The most objective of testing is to uncover a bunch of errors, consistently and with minimum effort and time. Stating formally, we can say, testing may be a method of corporal punishment a program with intent of finding miscalculation.

1. A productive check is one that uncovers Associate in Nursing hitherto undiscovered error.
2. A decent legal action is one that has likelihood of finding miscalculation, if it exists.
3. The check is insufficient to find probably gift errors.
4. The code additional or less confirms to the standard and reliable standards.

6.1. TYPES OF TESTING

6.1.1. UNIT TESTING

Unit testing, we have a tendency to test every module separately and integrate with the general system. Unit testing focuses verification efforts on the littlest unit of code style within the module. this is often conjointly called module testing.

The module of the system is tested individually. as an example, the validation check is completed for variable the user input given by the user that validity of the information entered. it's terribly straightforward to search out error rectify the system. Every Module will be tested victimization the subsequent 2 Strategies: recording machine Testing and White Box Testing.

6.1.2. BLACK BOX TESTING

Recording machine checking may be a code testing technique during which practicality of the code below test (SUT) is tested while not staring at the interior code structure, implementation details and data of internal ways of the code. This type of testing is predicated entirely on the code needs and specifications. In recording machine Testing we have a tendency to simply concentrate on inputs and output of the package while not bothering concerning internal data of the code program. The on top of recording machine will be any package you wish to check. For example, Associate in Nursing software like Windows, a web site like Google, a information like Oracle or maybe your own custom

application. Under recording machine testing, you can check these applications by simply that specialize in the inputs and outputs while not knowing their internal code implementation.

Types of Black Box Testing

There are many varieties of recording machine Testing however following ar the outstanding ones.

- **Functional testing:** This recording machine testing kind is said to purposeful needs of a system; it's done by code testers.
- **Non-Functional testing:** This sort of recording machine testing isn't associated with testing of a selected practicality, however non-functional needs like performance, measurability, usability.
- **Regression testing:** Regression testing is completed once code fixes, upgrades or the other system maintenance to visualize the new code has not affected the prevailing code.

6.1.3. WHITE BOX TESTING

White Box Testing is that the testing of a code solution's internal committal to writing and infrastructure. It focuses totally on Traffic Redundancy Elimination theming security, the flow of inputs and outputs through the applying, and rising style and value. White box testing is additionally called clear, open, structural, and glass box testing. It is one amongst 2 elements of the "box testing" approach of code testing.

System Testing:

Once the individual module testing is completed, modules are assembled and integrated to perform as a system. The top-down testing, that began from higher level to lower-level module, was allotted to visualize whether or not the whole system is playacting satisfactorily. There are 3 main types of System testing: Alpha Testing, Beta Testing, Acceptance Testing.

Alpha Testing: This refers to the system checking that's allotted by the test team with the Organization.

Beta Testing: This refers to the system testing that's performed by a particular cluster of friendly customers.

Acceptance Testing: This refers to the system testing that's performed by the client to see whether or not or to not settle for the delivery of the system.

6.2. TEST STRATEGY AND APPROACH

Field testing will be performed manually and functional tests will be written in detail.

Test objectives

- All field entries must work properly.
- Pages must be activated from the identified link.
- The entry screen, messages and responses must not be delayed.

Features to be tested

- Verify that the entries are of the correct format

- No duplicate entries should be allowed • All links should take the user to the correct page.

Test Scenarios

General Scenarios

- All mandatory fields should be validated and indicated by asterisk (*) symbol
  • Validation error messages should be displayed properly at correct position
  • All error messages should be displayed in same CSS style (e.g. using red colour)
- General confirmation messages should be displayed using CSS style other than

error messages style (e.g. using green colour)

- Dropdown fields should have first entry as blank or text like _Select ‘
- Delete functionality for any record on page should ask for confirmation

GUI and Usability Test Scenarios

- All fields on page (e.g. text box, radio options, dropdown lists) should be aligned properly.
- Scroll bar should be enabled only when necessary
- Description text box should be multi-line
- User should be able to submit the form again by correcting the errors
- Default radio options should be pre-selected on page load
- Check all pages for broken images

Test Scenarios for a Window

- Check if default window size is correct
- Check if child window size is correct
- Check if child windows are getting closed on closing parent/opener window
- Check window minimize, maximize and close functionality
- Check if window is re-sizable

Database Testing Test Scenarios

- Check if correct data is getting saved in database upon successful page submit
- Check values for columns which are not accepting null values
- Check for data integrity. Data should be stored in single or multiple tables based on design
- For every database add/update operation log should be added
- Required table indexes should be created

Security Testing Test Scenarios

- Secure pages should use HTTPS protocol
- Check application logout functionality
- Check for Brute Force Attacks

- Password should not be stored in cookies
- Test for Denial-of-Service attacks

CHAPTER - 7

CONCLUSION

Businesses can understand how smoothly their operations are running and where they can make improvements. It helps them see if orders are being processed and delivered without issues, where their sales are coming from, what products are popular, how customers want their orders shipped, where their customers are located, whether they're dealing with businesses or individuals, how much they can expect to sell in the future, and how well delivery services are performing.

FUTURE SCOPE

This algorithm is still inadequate in enhancing conversational abilities through deeper understanding of context, emotions, and user preferences, while also integrating emerging technologies like AI-driven personalization and multimodal interaction for richer user experiences., So future research will focus on solving these problems.

REFERENCES

[1] Adamopoulou, Eleni, and Lefteris Moussiades. “An overview of chatbot technology.” IFIP International Conference on Artificial Intelligence Applications and Innovations. Springer, Cham, vol. 584, 2020.

[2] C. Ouaddi, L. Benaddi, I. Khriss, and A. Jakimi, "Developing Conversational Agent Using Deep Learning Techniques," in Computer Sciences & Mathematics Forum, 2023, vol. 6, no. 1: MDPI, p. 3.

[3] Anupam Singh Computer Engineering, M. H. Saboo Siddik Polytechnic Mumbai, India. “Intelligent Chatbot.” International Journal of Engineering Research & Technology (IJERT) ISSN: 2278-0181 NREST - 2021.

[4] Rohit Tamrakar, Niraj Wani Mechanical Engineering Department. “Design and Development of CHATBOT: A Review”-2021

[5] S. Pérez-Soler, E. Guerra, and J. de Lara, "Creating and migrating chatbots with conga," in 2021 IEEE/ACM 43rd International Conference on Software Engineering: Companion Proceedings (ICSE-Companion), 2021: IEEE, pp. 37-40. a

[6] G. Daniel, J. Cabot, L. Deruelle, and M. Derras, "Xatkit: multimodal low-code chatbot development framework," IEEE Access, vol. 8, pp. 15332-15346, 2020.

[7] D. Al-Ghadhban and N. Al-Twairesh, “Nabiha: An Arabic dialect chatbot,” International Journal of Advance Computer Science Applications, vol. 11, no. 3, 2020.

[8] The 4th International Workshop on the Advancements in Model Driven Engineering & software engineering (AMDE 2024) November 7-9, 2023, Almaty, Kazakhstan Towards a Software Factory for Developing the Chatbots in Smart Tourism Mobile Applications.

[9] P. Chittò, M. Baez, F. Daniel, and B. Benatallah, "Automatic generation of chatbots for conversational web browsing," in Conceptual Modeling: 39th International Conference, ER 2020, Vienna, Austria, November 3–6, 2020, Proceedings 39, 2020: Springer, pp. 239-249.

[10] V. Vashisht and P. Dharia, “Integrating chatbot application with qlik sense business intelligence (BI) tool using natural language processing (NLP),” in Micro-Electronics and Tele communication Engineering, pp. 683–692, Springer, Singapore, 2020