A MINI PROJECT REPORT

ON

# "IMPROVING SHOPPING MALL REVENUE BY REAL-TIME CUSTOMIZED DIGITAL COUPON ISSUANCE"

A Project submitted in partial fulfilment of the Requirement for the award of degree of

BACHELOR OF TECHNOLOGY

IN

COMPUTER SCIENCE AND ENGINEERING

BY

BHUKYA SHASHANK NAIK 21B41A0520

B. Tech IV Year I semester

**DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING**

**K S H A T R I Y A**

COLLEGE OF ENGINEERING

NH-63, CHEPUR, ARMOOR, DIST: NIZAMBAD-(T.S)

(Affiliated to JNTU Hyderabad)

2024-2025.

**Ref: ________** **Date:_________**

## DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

## CERTIFICATE

This is to certify that the Mini Project Report on

**"IMPROVING SHOPPING MALL REVENUE BY REAL-TIME CUSTOMIZED DIGITAL COUPON ISSUANCE"**

Is submitted by

**BHUKYA SHASHANK NAIK** **21B41A0520**

In partial fulfilment for the award of degree of B. Tech in **COMPUTER SCIENCE AND ENGINEERING (CSE)** from **KSHATRIYA COLLEGE OF ENGINEERING** affiliated to **JAWAHARLAL NEHRU TECHNOLOGICAL UNIVERSITY HYDERABAD**. This is a Bonafide work carried out in the college for the academic year 2024-2025.

| | | |
|---|---|---|
| Signature of the Guide | **External Examiner** | Signature of the HOD |
| **MR. VENKATESH** | | **MR. M. LAXMAN** |
| **ASST.PROF CSE** | | **HOD CSE** |

## ACKNOWLEDGEMENT

We express our gratitude to **DR. R. K. PANDEY**, Principal of Kshatriya College of Engineering Who permitted us to carry the project work as part of the Academics.

We would like to thank **MR. M. LAXMAN**, Head of the Department, Computer Science of Engineering (CSE), for his support and encouragement in completing our project.

We express our profound gratitude to our guide **MR. M. LAXMAN** APROF of Kshatriya College of Engineering, by providing required information without her guidance, cooperation and encouragement, we couldn't learn many new things during our project tenure.

We would also express my sincere thanks to all the staff members of Kshatriya College of Engineering, for their kind cooperation and timely help during our academic career.

We would also express our special thanks to all my family members, for their kind cooperation and timely help during my academic career.

**BHUKYA SHASHANK NAIK** **21B41A0520**

## DECLARATION

We do here solemnly declare that the project entitled **“IMPROVING SHOPPING MALL REVENUE BY REAL-TIME CUSTOMIZED DIGITAL COUPON ISSUANCE”** is done by us a partial fulfilment for the award of the degree B. TECH, in Computer science & Engineering during the academic year 2024 – 2025 and no part of this work in part or full is submitted to any other University.

We further declare that dissertation has not been submitted to any previous course.

| | |
|---|---|
| **BHUKYA SHASHANK NAIK** | **21B41A0520** |

# ABSTRACT

The competitive landscape of retail shopping malls necessitates innovative strategies to enhance customer engagement and drive revenue growth. This paper presents a novel approach to improving shopping mall revenue through real-time customized digital coupon issuance. By leveraging advanced data analytics, machine learning algorithms, and mobile technology, shopping malls can provide personalize product offers to shoppers based on their preferences, behaviors, and real-time location within the mall. Thismethod not only enhances the shopping experience but also boosts customer of loyaltyand spending. The proposed system integrates with existing mall infrastructure including Wi-Fi, Bluetooth beacons, and point-of-sale systems, to collect and analyse data, ensuring the delivery of relevant coupons at the optimal moment. Our demonstrated significant increases in foot traffic, average transaction value, and overall sales, highlighting the effectiveness of real-time customized in driving shopping mall revenue. This paper discusses the, data processing techniques, and the results of the pilot study, insights for mall operators looking to adopt similar strategies.

# INDEX

# LIST OF FIGURES

# 1. INTRODUCTION

# 1.INTRODUCTION:

## 1.1 GENERAL

The retail industry is undergoing significant transformation due to advancements in technology and changing consumer behaviours. Shopping malls, in particular, are seeking innovative ways to attract customers, enhance their shopping experience, and boost revenue. One effective strategy is the use of digital coupons, which can be tailored to individual shoppers' preferences and delivered in real-time. This approach leverages data analytics, machine learning, and mobile technology to create a personalized shopping experience that drives engagement and sales.

With the development of big data and deep learning technology, big data and deep learning technology have also been applied to the marketing field, which was a part of management. Also, growth in internet adoption has made digital coupons a popular promotional tool. Customized digital coupon issuance is a very important topic in online commerce. This is because maintaining existing customers is a more important business issue than acquiring new customers. Also, retaining existing customers is much more economically advantageous than acquiring new customers. In fact, the acquisition cost of new customers is known to be five to six times higher than the maintenance cost of existing customers. Companies that have effectively managed customer churn by improving customer retention are known to have a positive effect not only on the company's profitability but also on improving brand image by improving customer satisfaction. Customized coupon issuance research has traditionally been active in highly competitive and urgent sectors such as telecommunications, finance, distribution, and game industries, and has focused technology.

Also, recently, AI based marketing using big data analysis and deep learning isemerging. Such AI-driven targeting can save huge amounts of marketing costs and raise online sales provided that the targeting model succeeds in estimating customer responsiveness accurately. In particular, in the case of online shopping malls, the average purchase conversion rate is around 2%. Online shopping malls have the advantage of being easily accessed through the PC web or mobile web, but on the contrary, this advantage can be a disadvantage that it is easy to see and leave quickly. Therefore, even the slightest reduction of customer churn rate can lead to high conversions, which can lead to huge profits. The easiest and most intuitive way is to issue personalized coupons to customers in real time. By selecting customers with a high risk of real-time churn and issuing real-time customized discount coupons, it is possible to increase sales by increasing the purchase conversion rate without burdening special expenses such as promotional events. And to put these strategies into action, you needAI- powered strategy. After AI automatically learns the histories of customers, it is possible to properlyissue coupons by identifying the behaviours and tastes of individual customers. Among the AI methodologies, in particular, deep learning-based strategies can be implemented. Deep learning learns a large amount of data to make an optimal decision, and the more data, the better the result. By learning a large amount of real-time log data accumulated in an online shopping mall, it is possible to predict customer behaviour and taste. In particular, it is possible to create a more sophisticated model every day by updating and re-learning the existing model with data that accumulates everyday.AI-based customized coupon issuance methods are largely divided into three: customer segmentation, customer churn prediction, and personalized recommendation. Customer segmentation is an activity that categorizes customers according to their homogeneous customer characteristics, providing the basis for differentiated marketing activities by customer group. Machine learning models used for customer segmentation were mainly used either supervised learning models such as decision trees or unsupervised learning models such as self-organizing maps (SOMs) or K-means models. One of the key features of recent machine learning-based customer segmentation studies is that customer segmentation is being performed for related customers.

other marketing research purposes, such as customer churn prediction. Customer churn prediction is also one of the main marketing research topics based on machine learning. Not to mention the fact that effective churn prediction has been recognized as a critical research topic not only for marketing but also for enterprise-wide management strategy, with the increasing number of customer churn under a highly competitive modern business environment, many new model development studies have been conducted to successfully predict customer churn. In the past, there have been major studies to learn models using single algorithms such as decision trees, logistic regression, and artificial neural networks to predict customer deviations, however, in recent, more attempts have been made to develop ensemble models or hybrid models that interconnects different models. Meanwhile, personalized recommendation systems are also one of the most active machine learning-based marketing research topics along with churn prediction. Research on personalized recommendations applied to recommended services such as Amazon and Netflix are increasing. Personalized recommendation studies have been dominated by model development studies to enhance predictive performance itself.

On the other hand, customized coupon issuance can contribute greatly to online shopping malls. In the case of an online shopping mall, real-time performance is required compared to an offline shopping mall because a large number of users come and go in an instant. Therefore, it is inappropriate to apply the traditional offline discount coupon issuance strategy online. Also, in online, a lot of log data can be collected much more than offline. Therefore, if you use the marketing method using AI, you can establish effective marketing strategies such as discount coupon issuance strategy in real time.

In most studies, the entire customer group is regarded as a group and AI prediction models are developed at once. In fact, however, customers have different behavioural characteristics due to un explain able and different transaction patterns, so it is unreasonable to assume the entire customer as a single customer group. It will be much more powerful if AI models are established for each group who are sharing similar tendencies according to customer behaviour. In this study, applying deep learning techniques to real-time click stream data, we find customers with high chance of churning rates and issue a coupon that suits customers' preferences. This study has the following significance: First, we segmented the customer and develop a suitable model for customer churn pre- diction for each segmentation. Second, we made a click stream-based real-time customer churn risk pre-dictionmodel using deep learning models. Third, we improved the actual conversion rate by issuing customized coupons in real shopping mall website. Unlike other studies, the scientific contribution of this study was to analyse customers in real time using data collected in real time as well as going through three steps to prevent customer churn. Also, we applied our model to the actual shopping mall, demonstrating the economic effectiveness and efficiency of the three steps of our model.

## 1.2 OBJECTIVE OF THE PROJECT:

The primary objective of this project is to design and implement a system for real-time customized digital coupon issuance in shopping malls. This system aims to:

- Increase foot traffic and customer engagement within the mall.
- Enhance the shopping experience through personalized offers.
- Boost overall sales and revenue for mall retailers.
- Provide mall operators with valuable insights into customer behaviour and preference.

## 1.3 DESCRIPTION OF THE PROJECT:

The project involves developing a comprehensive system that integrates data collection, analysis, and real-time coupon delivery. The system utilizes existing mall infrastructure such as Wi-Fi, Bluetooth beacons, and point -of-sale systems to gather data on shopper behaviour and preferences. Machine learning algorithms analyse this data to generate personalized offers, which are then delivered to shoppers' mobile devices via a dedicated app. The system's effectiveness is evaluated through a pilot implementation, with metrics such as foot traffic, transaction value, and sales analysed to assess its impact.

# 2. LITERATURE SURVEY

# 2. LITERATURE SURVEY:

## LITERATURE SURVEY

The literature survey covers existing research and case studies on digital coupon systems, personalization techniques in retail, and the use od data analytics in enhancing customer engagement. It includes a review of the technological advancements that enable real-time data processing and the deployment of personalized marketing strategies in shopping malls. Studies have shown that personalized marketing significantly increases customer engagement and sales, underscoring the potential of real-time customized digital coupons.

## 2.1 FEASIBILITY STUDY

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

Three key considerations involved in the feasibility analysis are

- ECONOMIC FEASIBILITY
- TECHNICAL FEASIBILITY
- SOCIAL FEASIBILITY

### ECONOMIC FEASIBILITY:

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

### TECHNICAL FEASIBILITY:

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

**SOCIAL FEASIBILITY:**

The aspect of study is to check the level of acceptance of the system by the user. This includes the process of training the user to use the system efficiently. The user must not feel threatened by the system, instead must accept it as a neccessity. The level of acceptance by the users solely depends on the methods that are employed to educate the user about the system and to make him familiar with it. His level of confidence must be raised so that he is also able to make some constructive criticism, which is welcomed, as he is the final user of the system.

# 3.SYSTEM ANALYSIS

# 3. SYSTEM ANALYSIS:

## 3.1 EXISTING SYSTEM:

The existing systems in shopping malls often rely on traditional marketing techniques such as staticdiscounts, paper coupons, and generic promotions. These methods lack personalization and real-timerelevance, leading to lower engagement and effectiveness. Traditional coupons also face logisticalchallenges in distribution and redemption, making them less efficient in driving sales.

**The existing system has several drawbacks:**

The existing systems in shopping malls often rely on traditional marketing techniques such as staticdiscounts, paper coupons, and generic promotions. These methods have several drawbacks:

1. **Lack of Personalization:**
   - Traditional coupons are generally distributed broadly without considering individualshopper preferences or behaviors. This results in lower relevance and engagement fromcustomers who receive offers that do not match their interests.
2. **Inefficient Distribution:**
   - Paper coupons and static promotions require manual distribution, which can be time-consuming and labor-intensive. This method also often fails to reach the intended audienceeffectively, leading to wasted resources.
3. **Limited Reach:**
   - Static in-store promotions are visible only to shoppers who are already inside the mall or specific stores. This limits the ability to attract new customers or drive spontaneous visitsfrom potential shoppers who are nearby.
4. **Redemption Challenges:**
   - Paper coupons can be easily lost or forgotten, reducing the likelihood of redemption.Additionally, the manual process of redeeming paper coupons at the point of sale can becumbersome for both shoppers and store staff.
5. **Ineffective Data Utilization:**

- Existing systems often lack the capability to collect and analyze detailed shopper data. Without insights into customr behavior and preferences, mall operators cannot optimize their marketing efforts or improve the shopping experience.

## 3.2 PROPOSED SYSTEM:

The proposed system offers a significant improvement by delivering customized digital coupons in real-time. It leverages advanced data analytics and machine learning to understand shopper behavior and preferences, ensuring that the offers are relevant and timely. The system's integration with mob iletechnology allows for seamless delivery and redemption of coupons. Key features of the proposedsystem include:

**ADVANTAGES OF PROPOSED SYSTEM:**

1. **Real-Time Data Collection:**
   Utilizing Wi-Fi and Bluetooth beacons to track shopper movements and behaviors within the mall. This data provides insights into shopper traffic patterns, dwell times, and store visit frequencies.

2. **Personalization Engine:**
   Machine learning algorithms analyze collected data to create personalized offers basedon individual preferences, shopping history, and real-time behavior. This ensures that couponsare relevant and attractive to each shopper.

3. **Mobile Integration:**
   A mobile app delivers digital coupons to shoppers' smartphones, enabling easy access and redemption. Shoppers receive notifications about new offers as they move through the mall, encouraging spontaneous purchases.

4. **Dashboard:**
   A comprehensive dashboard for mall operators to monitor system performance and gain insights into shopper behavior and campaign effectiveness. This enables data-driven decision-making and continuous optimization of marketing strategies.

**5. Seamless Redemption:**

Digital coupons can be redeemed directly through the mobile app, simplifying the process for both shoppers and stor for both shoppers and store staff. Integration with point-of-sale systems ensures smooth and efficient transactions.

**6. Dynamic Targeting:**

The system can dynamically adjust offers based on real-time data, such as current shopper density in specific areas or recent purchase trends. This flexibility allows for more effective and timely promotions.

**7. Ehnaced customer Engagement**:

Personalized and timely offers enhance the overall shopping experience, increasing customer satisfaction and loyalty.

## 3.3. HARDWARE & SOFTWARE REQUIREMENTS:

### 3.3.1. SOFTWARE REQUIREMENT SPECIFICATIONS:

- Operating System:
  Windows 7, 8, 10, or Linux
- Coding Language
  Python 3.8
- Web Framework:
  Flask
  Django
- Database:
  PostgreSQL or MySQL

- Data Analytics Platform:
  Apache Spark or a similar big data processing framework

- Machine Learning Libraries:
  Scikit-learn
  TensorFlow or PyCharm

### 3.3.2. HARDWARE REQUIREMENT SPECIFICATIONS:

- System: pentium i3 processor.
- Hard Disk: 500 GB.
- Input Devices: Keyboard, Mouse
- Ram : 8GB

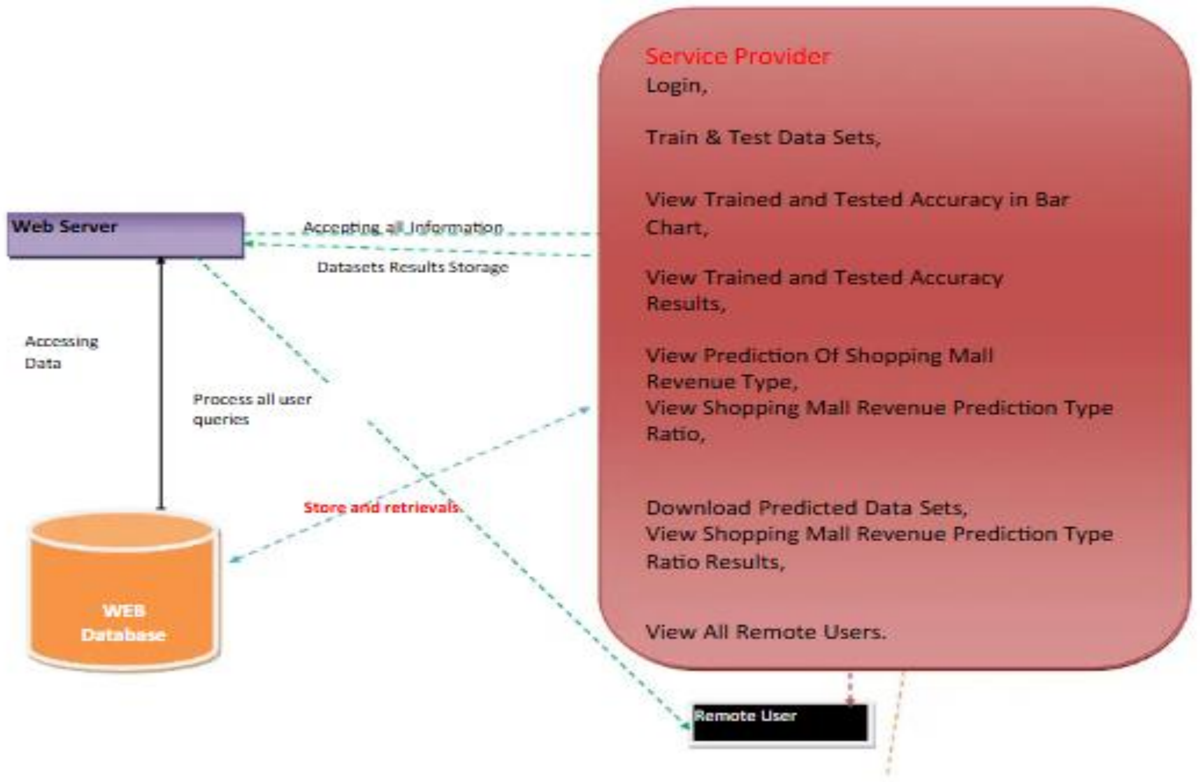
# 5. SYSTEM DESIGN

# 4. SYSTEM DESIGN:

## 4.1 SYSTEM ARCHITECTURE:

The system architecture for real-time customized digital coupon issuance consists of several keycomponents that work together to collect data, analyze shopper behavior, and deliver personalizedcoupons. The architecture is designed to be scalable, efficient, and secure, ensuring a seamless userexperience for both shoppers and mall operators.

**ARCHITECTURE DIAGRAM**


Service Provider
Login,
Train & Test Data Sets,
View Trained and Tested Accuracy in Bar Chart,
View Trained and Tested Accuracy Results,
View Prediction Of Shopping Mall Revenue Type,
View Shopping Mall Revenue Prediction Type Ratio,
Download Predicted Data Sets,
View Shopping Mall Revenue Prediction Type Ratio Results,
View All Remote Users.
Web Server
Accepting all Information
Datasets Results Storage
Accessing Data
Process all user queries
Store and retrievals
WEB Database
Remote User

REGISTER AND LOGIN,

Predict Shopping Mall Revenue Prediction Type,

VIEW YOUR PROFILE.

The architecture is divided into several key components:

**Presentation Layer:**

User Interface: Provides the graphical interface through which users interact with the system. This includes web pages for students to submit applications, upload documents, and track application status, as well as an administrative dashboard for managing applications.

**Application Layer:**

Application Server: Hosts the core business logic of the system, handling tasks such as form processing,document management, and user authentication. Technologies like Node.js, Django (Python), or Laravel (PHP)are used here.

**Data Layer:**

Database Management System (DBMS): Stores all the application data, user information, and documents.Popular DBMS options include MySQL, PostgreSQL, and MongoDB.

Data Storage: Ensures that all uploaded documents and files are securely stored, often using cloud storagesolutions for scalability and redundancy.

**Security Layer:**

Authentication and Authorization: Manages user authentication (e.g., login and registration) and authorizationto ensure that users can only access permitted resources.Data Encryption: Encrypts sensitive data both in transit (using SSL/TLS) and at rest to protect it fromunauthorized access.

**Communication Layer:**

Email Notifications: Automated email system to send notifications and updates to students regarding their application status.Real-Time Updates: Uses WebSocket's or similar technologies to provide real-time status updates and notifications to users.

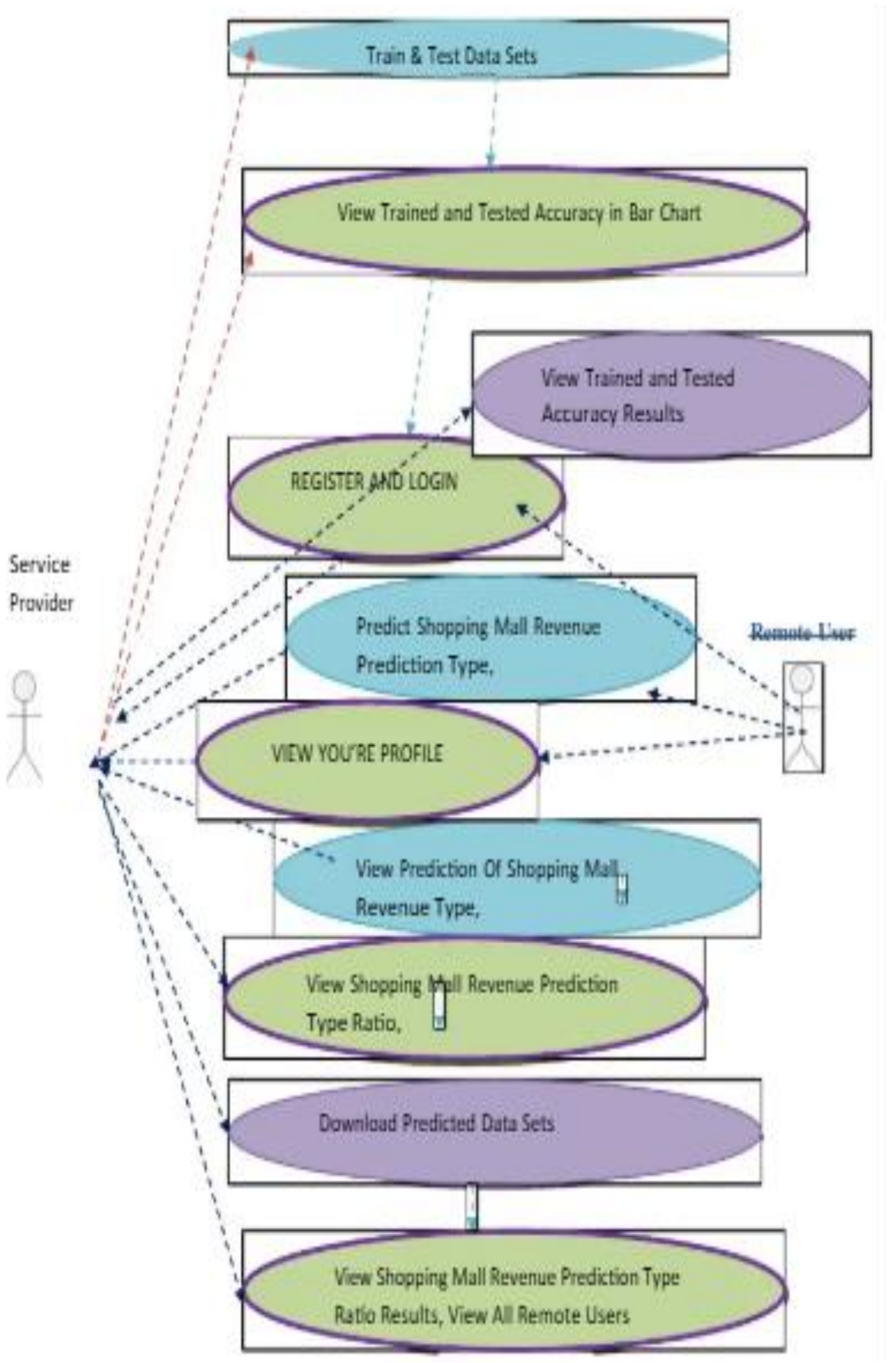
## 4.2 UML DIAGRAMS

Unified Modelling Language (UML) diagrams help in visualizing the design and structure of the system.

## Use Case Diagram:

This diagram outlines the interactions between the actors (shoppers and mall operators) and the system.

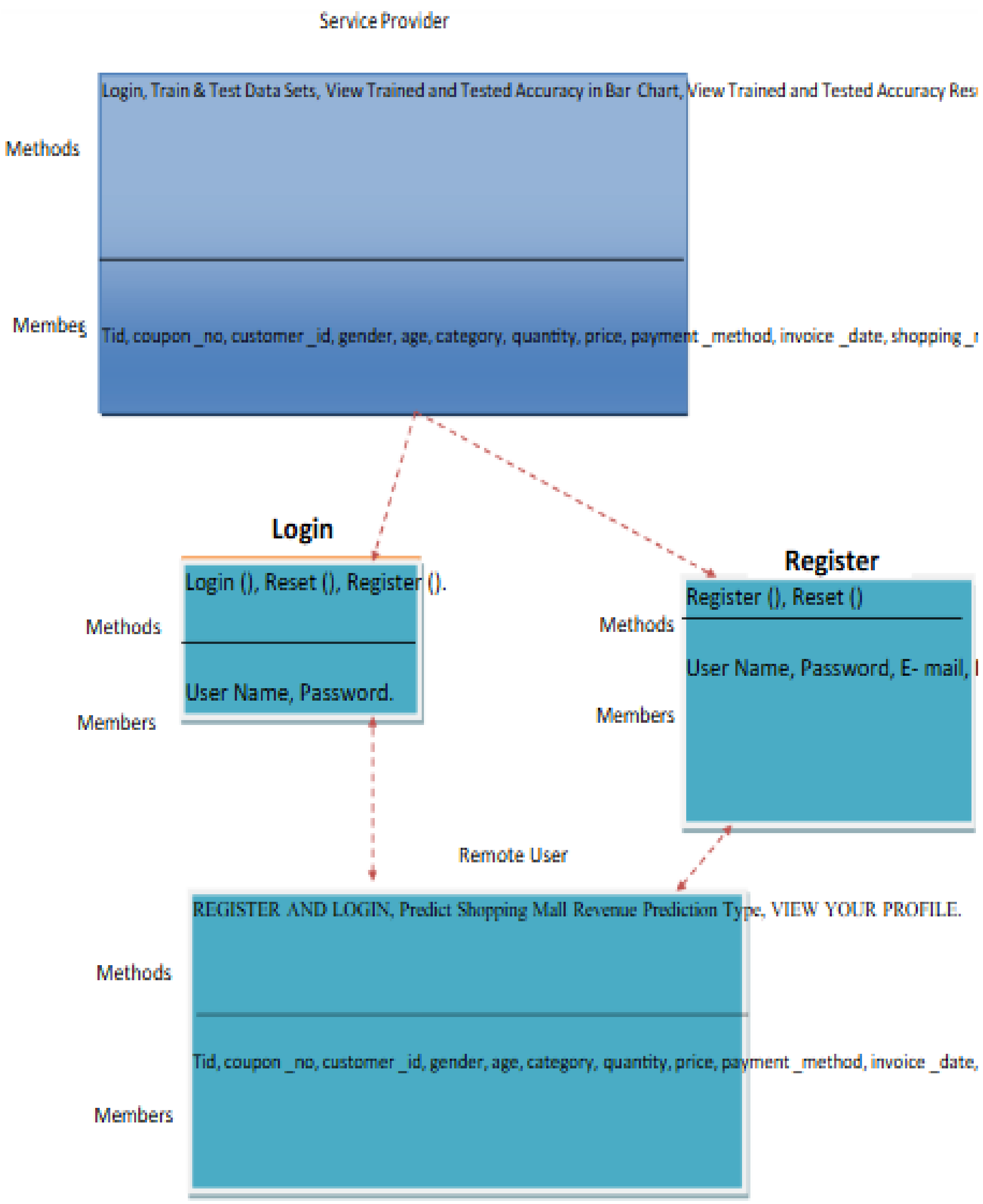
- **Use Case**


Train & Test Data Sets
View Trained and Tested Accuracy in Bar Chart
View Trained and Tested Accuracy Results
REGISTER AND LOGIN
Service Provider
Predict Shopping Mall Revenue Prediction Type,
Remote User
VIEW YOU'RE PROFILE
View Prediction Of Shopping Mall Revenue Type,
View Shopping Mall Revenue Prediction Type Ratio,
Download Predicted Data Sets
View Shopping Mall Revenue Prediction Type Ratio Results, View All Remote Users

**Class Diagram:**

This diagram shows the static structure of the system, including classes, attributes, methods, and relationships.

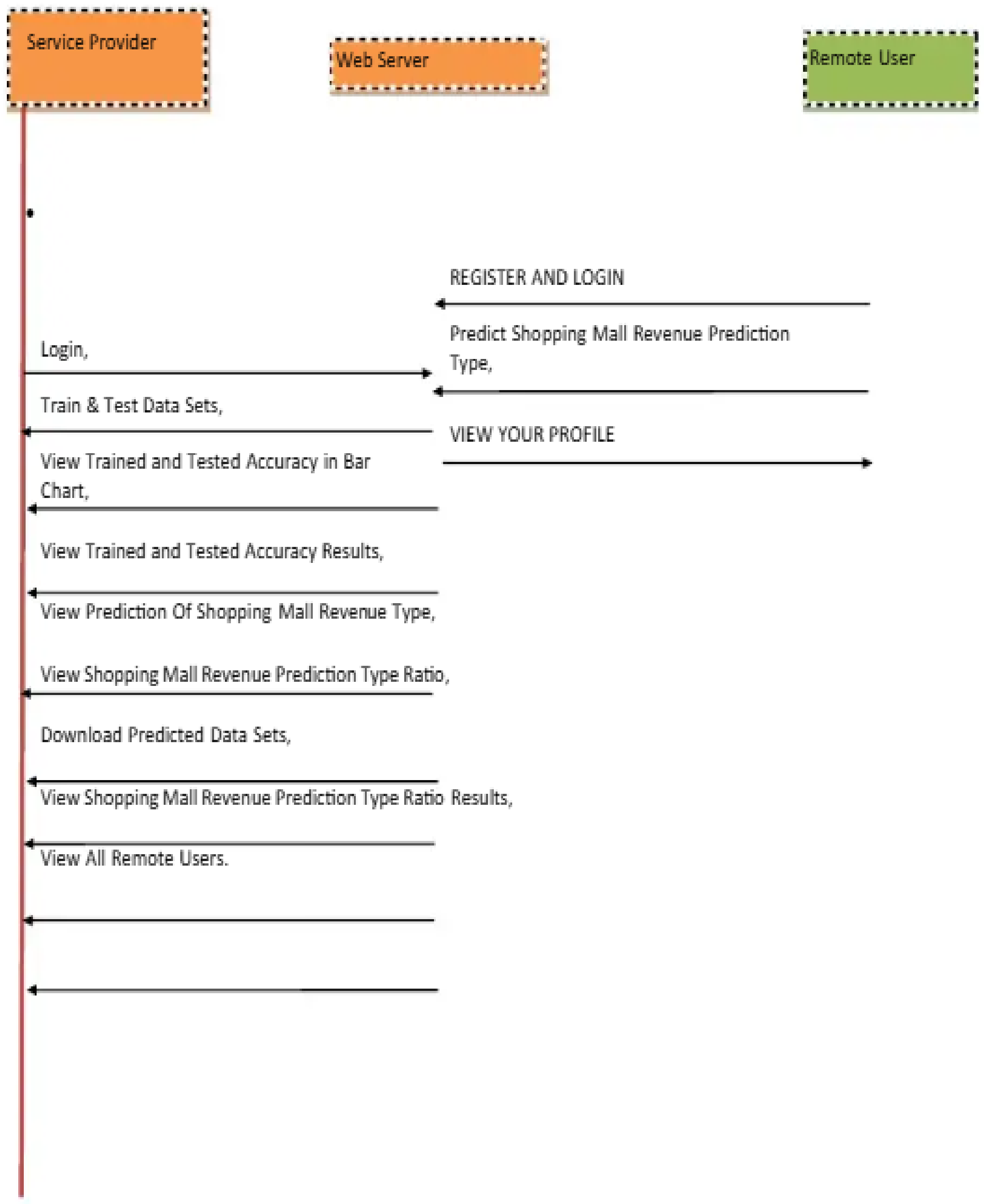
➢ **ClassDiagram:**


Service Provider
Login, Train & Test Data Sets, View Trained and Tested Accuracy in Bar Chart, View Trained and Tested Accuracy Res
Methods
Membeg
Tid, coupon _no, customer _id, gender, age, category, quantity, price, payment _method, invoice _date, shopping _
Login
Login (), Reset (), Register ().
Methods
User Name, Password.
Members
Register
Register (), Reset ()
Methods
User Name, Password, E- mail,
Members
Remote User
REGISTER AND LOGIN, Predict Shopping Mall Revenue Prediction Type, VIEW YOUR PROFILE.
Methods
Tid, coupon _no, customer _id, gender, age, category, quantity, price, payment _method, invoice _date,
Members

## SEQUENCE DIAGRAM:

This diagram illustrates how objects interact in a particular scenario of the system, showing the sequence of messages exchanged.

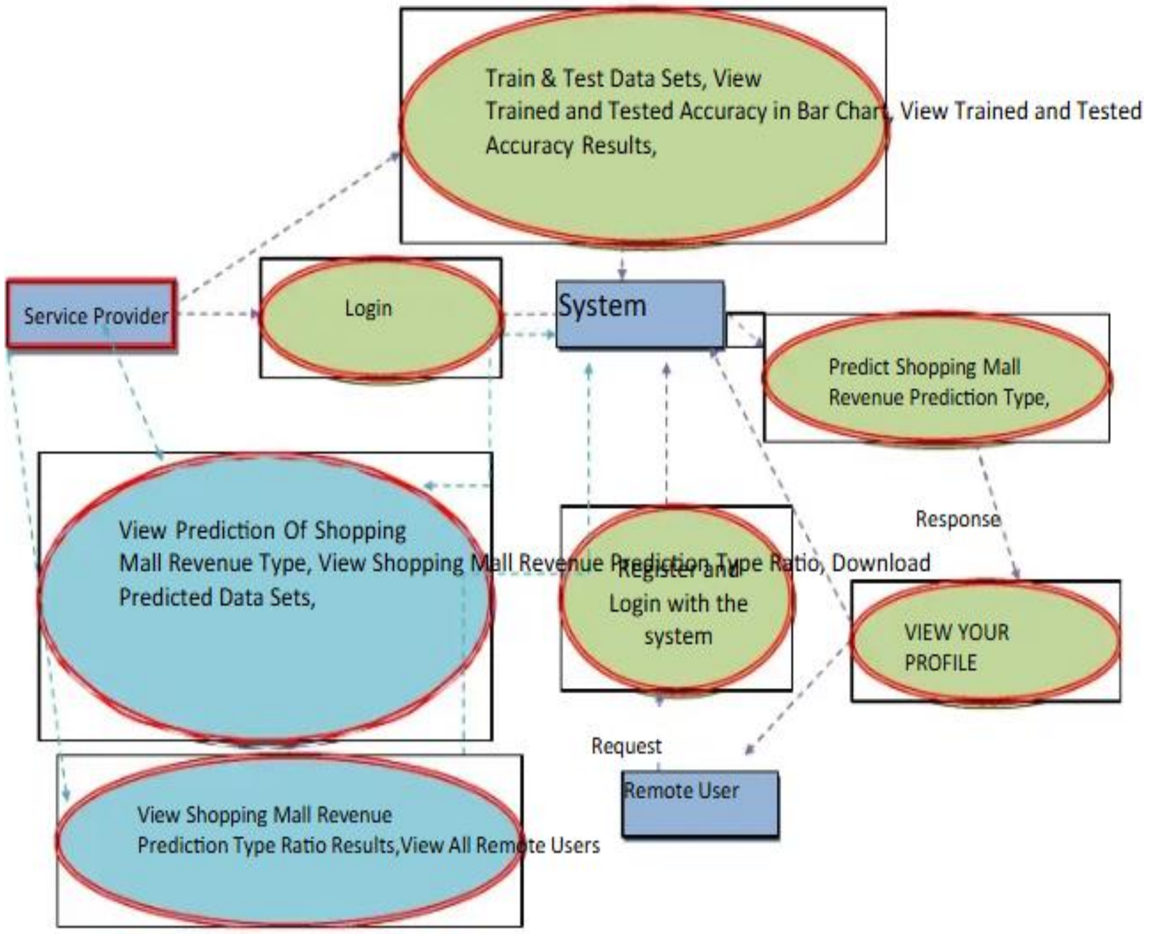
- Sequence Diagram:


Service Provider
Web Server
Remote User
REGISTER AND LOGIN
Predict Shopping Mall Revenue Prediction Type,
Login,
Train & Test Data Sets,
VIEW YOUR PROFILE
View Trained and Tested Accuracy in Bar Chart,
View Trained and Tested Accuracy Results,
View Prediction Of Shopping Mall Revenue Type,
View Shopping Mall Revenue Prediction Type Ratio,
Download Predicted Data Sets,
View Shopping Mall Revenue Prediction Type Ratio Results,
View All Remote Users.

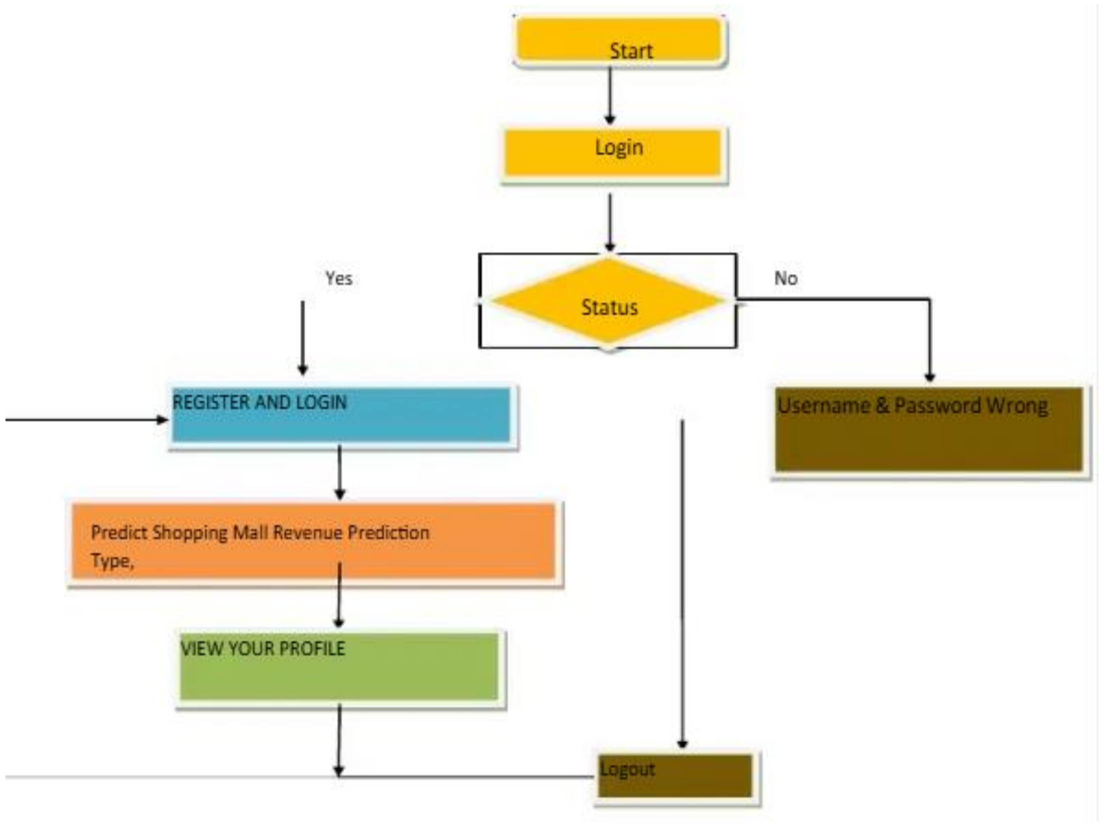
## ACTIVITY DIAGRAM:

This diagram depicts the work flow of the system, showing the activities involved in issuing and redeeming digital coupons.

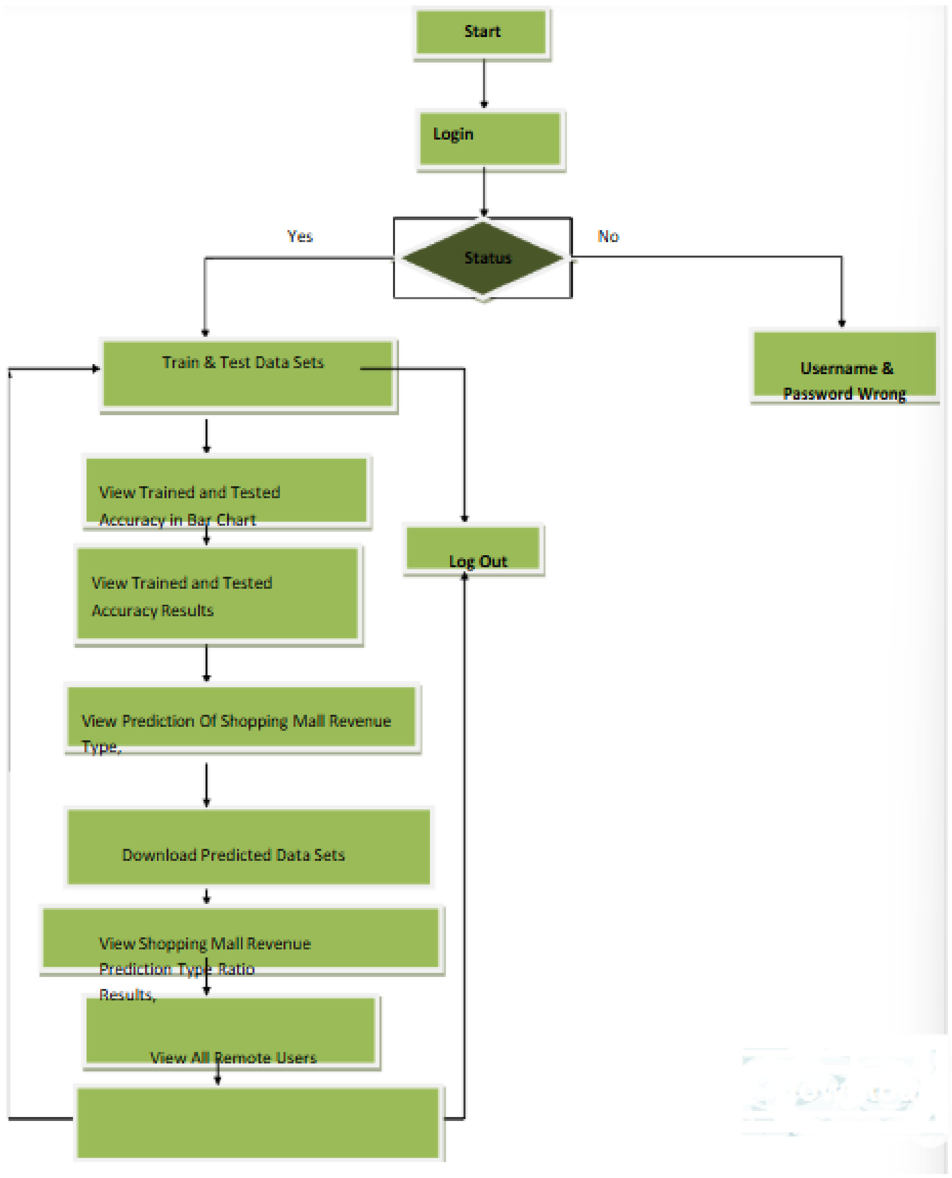
➢ **Data Flow Diagram:**


Train & Test Data Sets, View Trained and Tested Accuracy in Bar Chart, View Trained and Tested Accuracy Results,
Service Provider
Login
System
Predict Shopping Mall Revenue Prediction Type,
Response
View Prediction Of Shopping Mall Revenue Type, View Shopping Mall Revenue Prediction Type Ratio, Download Predicted Data Sets,
Register and Login with the system
VIEW YOUR PROFILE
Request
Remote User
View Shopping Mall Revenue Prediction Type Ratio Results,View All Remote Users

- **Flow Chart : Remote User**


Start
Login
Yes
Status
No
REGISTER AND LOGIN
Username & Password Wrong
Predict Shopping Mall Revenue Prediction Type,
VIEW YOUR PROFILE
Logout

- **Flow Chart : Service Provider**


Start
Login
Yes
Status
No
Train & Test Data Sets
Username & Password Wrong
View Trained and Tested Accuracy in Bar Chart
Log Out
View Trained and Tested Accuracy Results
View Prediction Of Shopping Mall Revenue Type,
Download Predicted Data Sets
View Shopping Mall Revenue Prediction Type Ratio Results,
View All Remote Users

# 5. IMPLEMENTATION

# 5. IMPLEMENTATION

## MODULES:

### Service provider

In this module, the Service Provider has to login by using valid user name and password. After login successful he can do some operations such as Train & Test Data Sets, View Trained and Tested Accuracy in Bar Chart, View Trained and Tested Accuracy in Bar Chart, View Trained and Tested Accuracy Results, View Prediction of Shopping Mall Revenue Type, View Shopping Mall Revenue Prediction Type Ratio, Download Predicted Data Sets, View Shopping Mall Revenue Prediction Type Ratio Results, View All Remote User.

### View and Authorize Users

In this module, the admin can view the list of users who all registered. In this, the admin can view the User's details such as, user name, email, address and admin authorize the users.

### Remote User

In this module, there are n numbers of users are present. User should register before doing any operations. Once user registers, their details will be stored to the database. After registration successful, he has to login by using authorized user name and password. Once Login is successful user will do some operations like REGISTER AND LOGIN Predict Shopping Mall Revenue Prediction Type, VIEW YOUR PROFILE.

## 5.1 MODULES DESCRIPTION:

The implementation of the real-time customized digital coupon issuance system involves several key modules, each responsible for specific functionalities within the system. Below is a detailed description of each module:

### 1.Data Collection Module:

- **Wi-Fi Tracking**: Captures the location of shoppers' mobile devices as they connect to the mall's Wi-Fi network.
- **Bluetooth Beacon Interaction:** Detects and records interactions between shoppers' mobile devices and strategically placed Bluetooth beacons within the mall.

- **POS Integration:** Collects transaction data and coupon redemption details from the point-of-sale systems in stores.

**2. Data Processing Module:**

- **Data Aggregation:** Collects raw data from the Wi-Fi tracking, Bluetooth beacons, and POS systems, aggregating it for further analysis.
- **Data Cleaning:** Cleans the collected data to remove any inaccuracies or inconsistencies, ensuring high-quality data for analysis.
- **Storage:** Stores the processed data in a centralized database (e.g., PostgreSQL or MySQL).

**3. Analytics and Personalization Module:**

- **Behaviour Analysis:** Uses machine learning algorithms to analyse shopper behaviour and preferences based on historical and real-time data.
- **Coupon Generation:** Generates personalized coupons for shoppers using insights derived from the behaviour analysis.
- **Recommendation Engine:** Continuously refines and updates coupon recommendations based on new data and shopper interactions.

**4. Admin Dashboard Module:**

- **Monitoring:** Allows mall operators to monitor system performance and shopper engagement metrics in real-time.
- **Analytics:** Provides visualizations and insights into shopper behaviour, coupon redemption rates, and overall system effectiveness.
- **Management:** Enables administrators to configure system settings, manage coupon campaigns, and view detailed reports.

## 5.2 SOFTWARE ENVIRONMENT

### 1.1 PYTHON

Python is a high-level, interpreted, interactive and object-oriented scripting language. Python Is designed to be highly readable. It uses English keywords frequently where as other language Use punctuation, and it has fewer syntactical constructions than other languages.

- **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 is Object-oriented:** Python supports Object- oriented style or technique of programming that encapsulates code with your programs.

- **Python is a Beginner's Language:** Python is a great language for the beginner- level programmers and supports the development of a wide range of applications from simple text processing to WWW browsers to games.

### 1.2 History of python

Python was developed by Guido van Rossum in the late eighties and early nineties at the National Research Institute for Mathematics and Computer Science in the Netherlands. Python is derived from many other languages, including ABC, Modula-3, C, C++, Algol-68, Small Talk, and Unix shell and other scripting languages. Python is copyrighted. Like Perl, Python source code is now available under the GNU General Public License (GPL). Python is now maintained by a core development team at the institute, although Guido van Rossum still holds a vital role in directing its progress.

### 1.3 Python Features

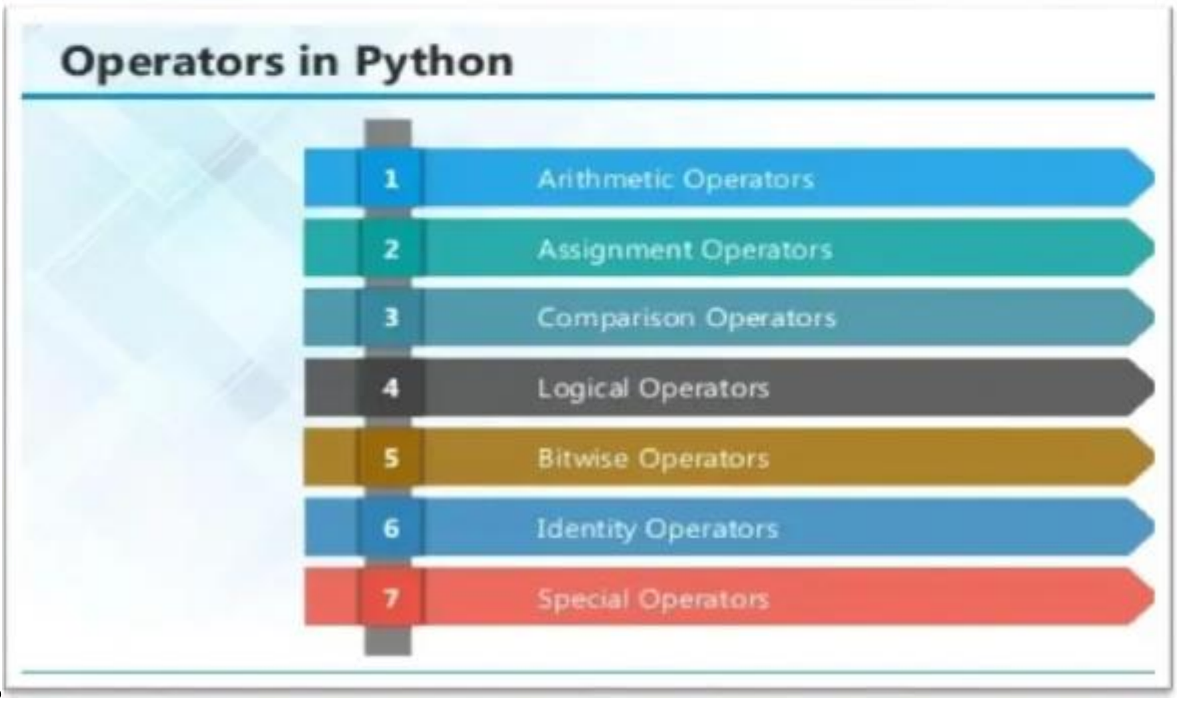
Python's features include:

- **Easy-to-learn**: Python has few keywords, simple structure, and a clearly defined syntax. This allows the student to pick up the language quickly.

- **Easy-to-read:** Python code is more clearly defined and visible to the eyes.
- **Easy-to-maintain:** Python's source code is fairly easy-to-maintain.
- **A broad standard library:** Python's bulk of the library is very portable and cross-platform compatible on UNIX, Windows, and Macintosh.
- **Interactive Mode:** Python has support for an interactive mode which allows interactive testing and debugging of snippets of code.
- **Portable:** Python can run on a wide variety of hardware platform and has the same interface on all platforms.
- **Extendable:** You can add low-level module to the Python interpreter. These modules enable programmers to add to or customize their tools to be more efficient.
- **Databases:** Python provides interfaces to all major commercial databases.
- **GUI Programming:** Python supports GUI applications that can be created and ported to many system calls, libraries and windows systems, such as windows MFC, Macintosh, and the X Window system of Unix.
- **Scalable:** Python provides a better structure and support for large programs than shell scripting.

**Python has a big list of good features:**

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



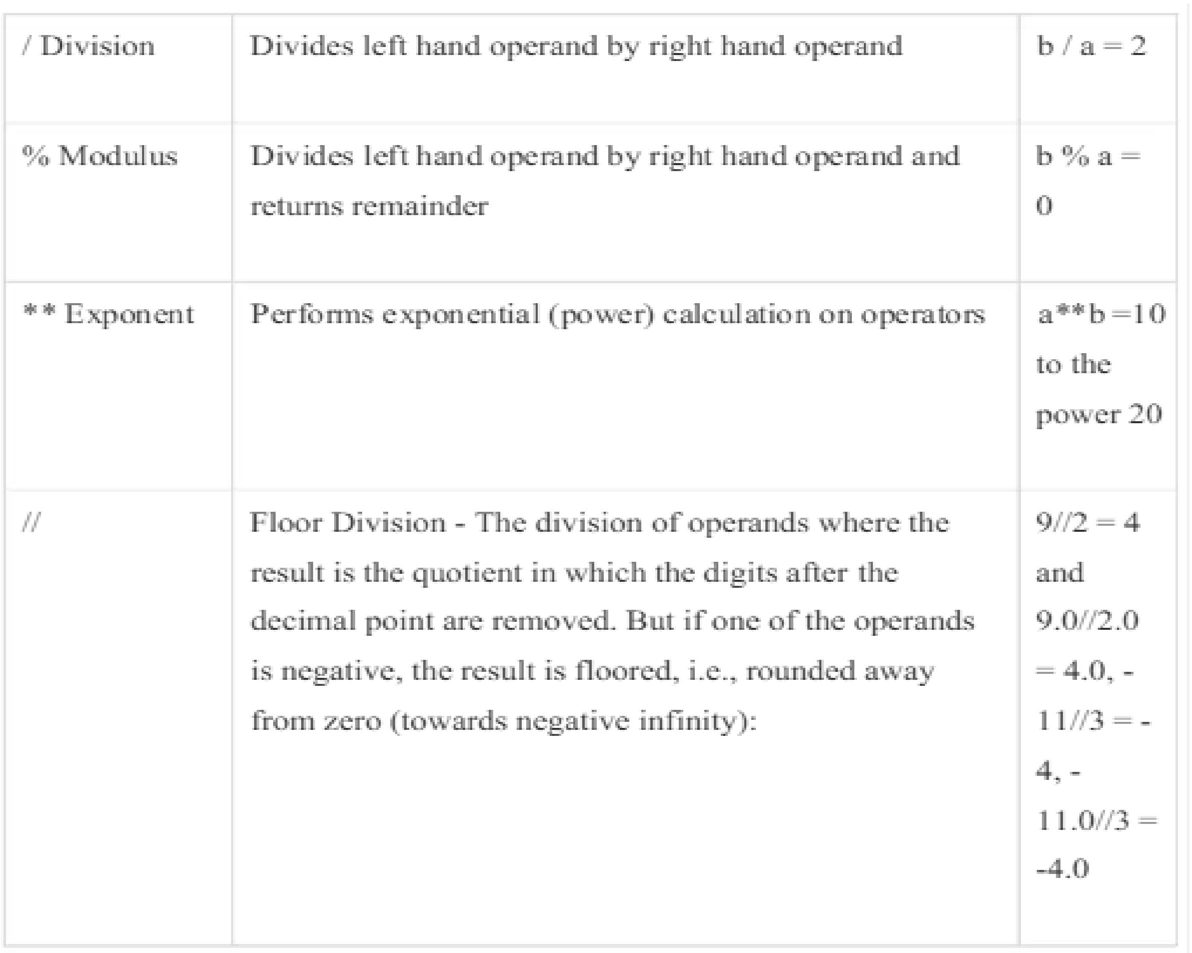
**Operators in Python**

1. Arithmetic Operators
2. Assignment Operators
3. Comparison Operators
4. Logical Operators
5. Bitwise Operators
6. Identity Operators
7. Special Operators

S

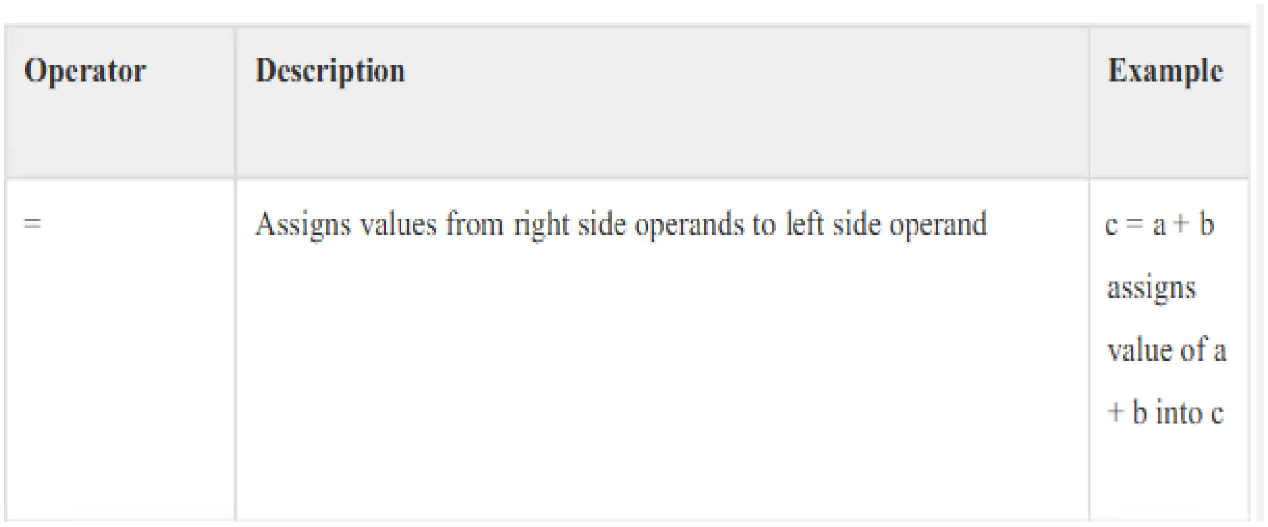
## 2.1 ARITHMETIC OPERATORS

| Operator | Description | Example |
|---|---|---|
| + Addition | Adds values on either side of the operator. | a + b = 30 |
| - Subtraction | Subtracts right hand operand from left hand operand. | a – b = -10 |
| * Multiplication | Multiplies values on either side of the operator | a * b = 200 |

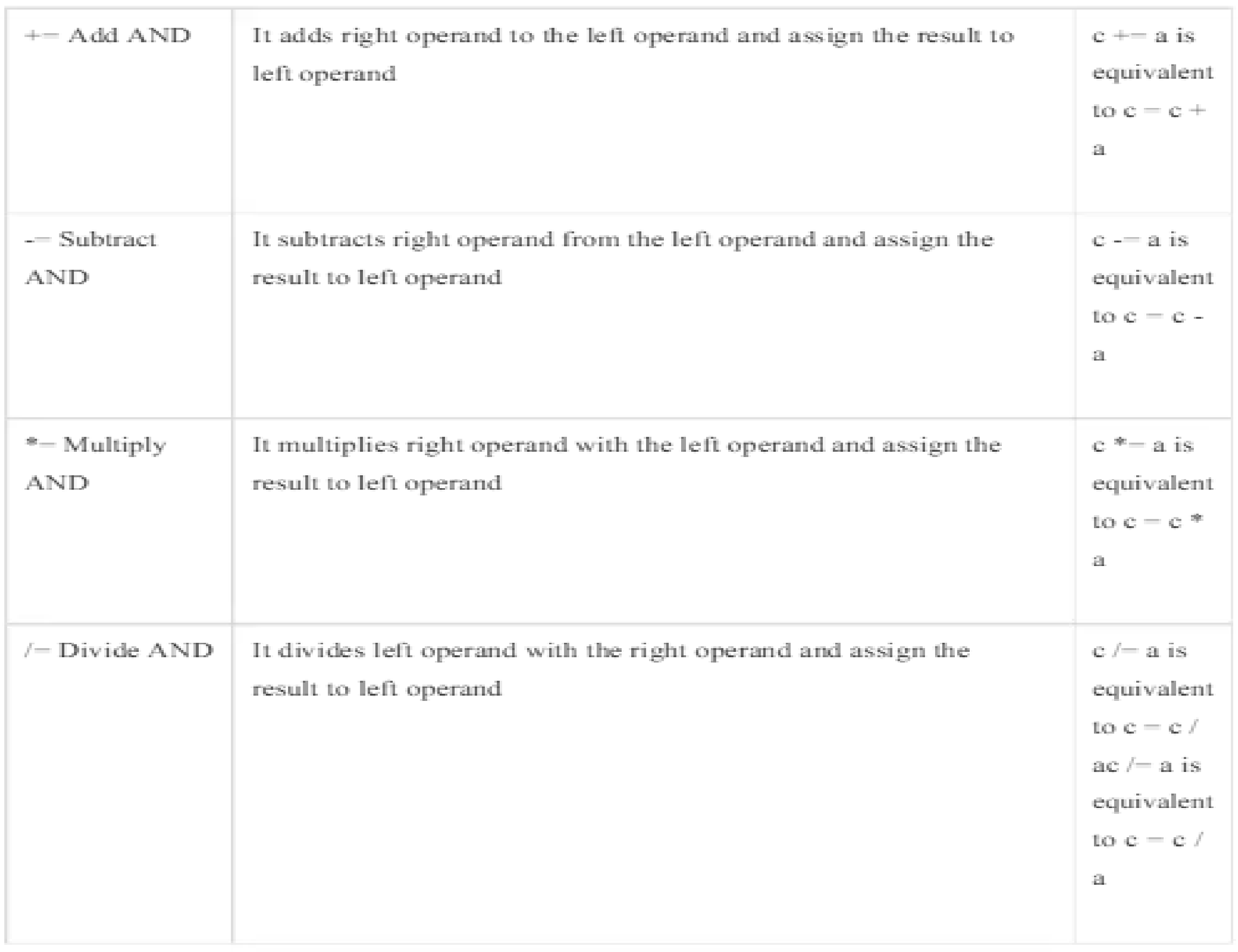


| | | |
|---|---|---|
| / Division | Divides left hand operand by right hand operand | b / a = 2 |
| % Modulus | Divides left hand operand by right hand operand and returns remainder | b % a = 0 |
| ** Exponent | Performs exponential (power) calculation on operators | a**b =10 to the power 20 |
| // | Floor Division - The division of operands where the result is the quotient in which the digits after the decimal point are removed. But if one of the operands is negative, the result is floored, i.e., rounded away from zero (towards negative infinity): | 9//2 = 4 and 9.0//2.0 = 4.0, -11//3 = -4, -11.0//3 = -4.0 |

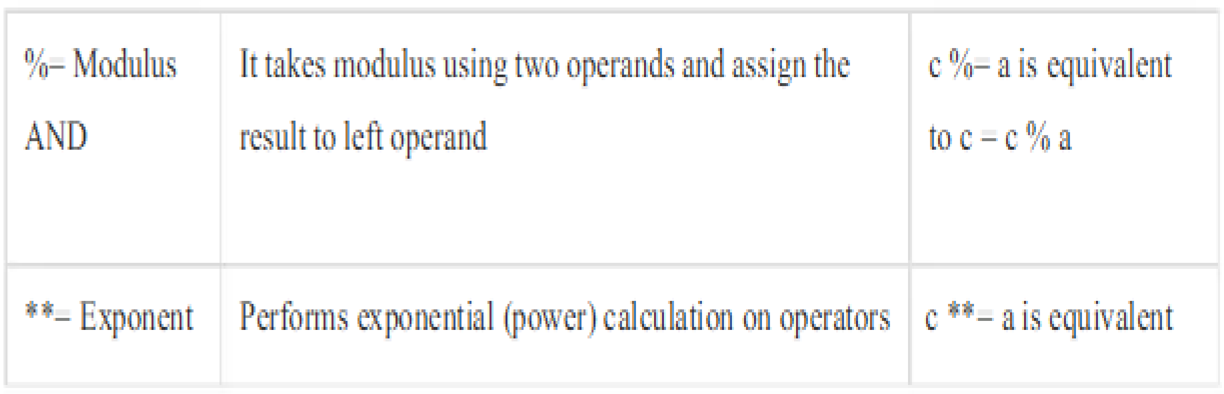
## 2.2 ASSIGNMENT OPERATOR



| Operator | Description | Example |
|---|---|---|
| = | Assigns values from right side operands to left side operand | c = a + b assigns value of a + b into c |



| | | |
|---|---|---|
| += Add AND | It adds right operand to the left operand and assign the result to left operand | c += a is equivalent to c = c + a |
| -= Subtract AND | It subtracts right operand from the left operand and assign the result to left operand | c -= a is equivalent to c = c - a |
| *= Multiply AND | It multiplies right operand with the left operand and assign the result to left operand | c *= a is equivalent to c = c * a |
| /= Divide AND | It divides left operand with the right operand and assign the result to left operand | c /= a is equivalent to c = c / ac /= a is equivalent to c = c / a |



| | | |
|---|---|---|
| %= Modulus AND | It takes modulus using two operands and assign the result to left operand | c %= a is equivalent to c = c % a |
| **= Exponent | Performs exponential (power) calculation on operators | c **= a is equivalent |

| | | |
|---|---|---|
| AND | and assign value to the left operand | to c = c ** a |
| //= Floor Division | It performs floor division on operators and assign value to the left operand | c //= a is equivalent to c = c // a |

### 2.3 IDENTITY OPERATOR

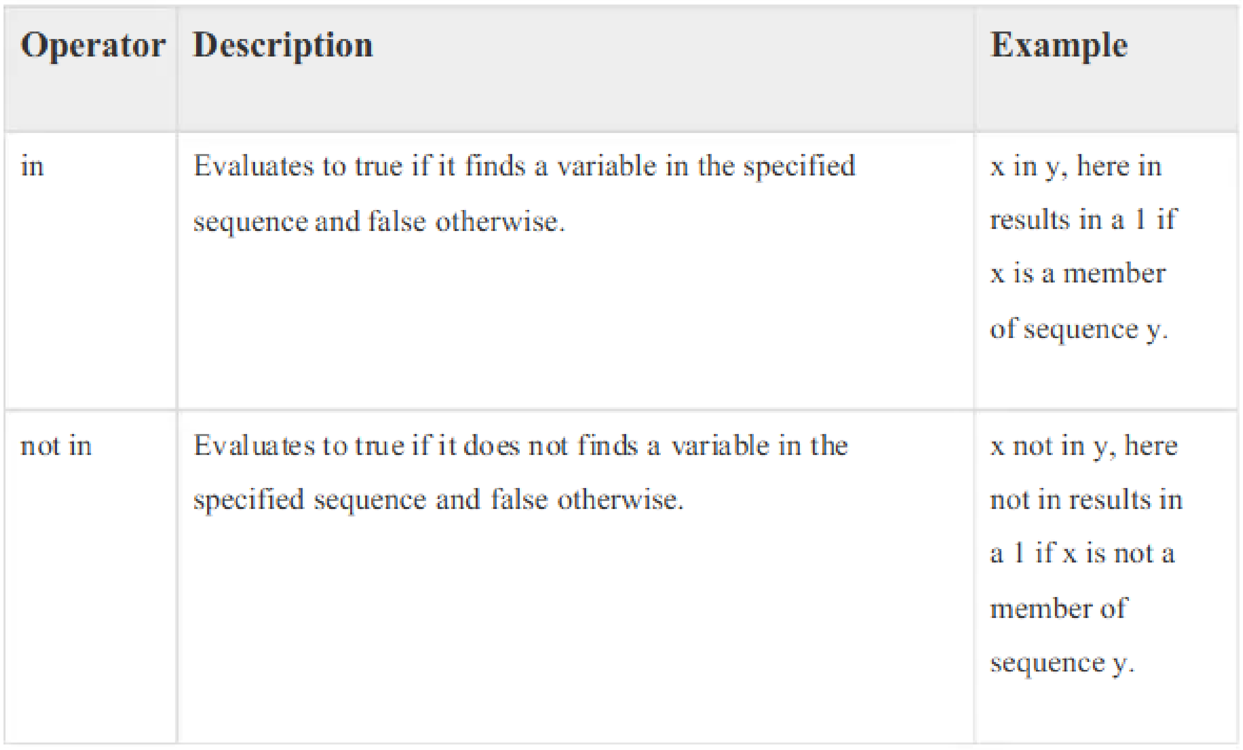
| Operator | Description | Example |
|---|---|---|
| is | Evaluates to true if the variables on either side of the operator point to the same object and false otherwise. | x is y, here **is** results in 1 if id(x) equals id(y). |
| is not | Evaluates to false if the variables on either side of the operator point to the same object and true otherwise. | x is not y, here **is not** results in 1 if id(x) is not equal to id(y |

### 2.4 COMPARISON OPERATOR

| Operator | Description | Example |
|---|---|---|
| & Binary AND | Operator copies a bit to the result if it exists in both operands | (a & b) (means 0000 1100) |
| \| Binary OR | It copies a bit if it exists in either operand. | (a \| b) = 61 (means 0011 1101) |
| ^ Binary XOR | It copies a bit if it is set in one operand but not both. | (a ^ b) = 49 (means 0011 0001) |
| ~ Binary ones complement | It is unary and has the effect of 'flipping' bits. | (~a) = 61 (means1100 0011 in 2's complement form due to a signed binary numbers. |
| << Binary Left Shift | The left operand value is moved left by the number of bits specified by the right operand. | a << 2 = 240 (means 1111 0000) |

| | | |
|---|---|---|
| >> Binary Right Shift | The left operands value is moved right by the number of bits specified by the right operand. | a >> 2 = 15 (means 0000 1111) |
| And Logical AND | If both the operands are true then condition becomes true. | (a and b) is true. |
| Or Logical OR | If any of the two operands are non-Zero then condition becomes true. | (a or b) is true. |
| Not Logical NOT | Used to reverse the logical state of its operand. | Not (a and b) is false. |

**2.6 Membership Operators**



| Operator | Description | Example |
|---|---|---|
| in | Evaluates to true if it finds a variable in the specified sequence and false otherwise. | x in y, here in results in a 1 if x is a member of sequence y. |
| not in | Evaluates to true if it does not finds a variable in the specified sequence and false otherwise. | x not in y, here not in results in a 1 if x is not a member of sequence y. |

**Python Operators Precedence**

| Operators | Description |
|---|---|
| ** | Exponentiation (raise to the power) |
| ~ + - | Complement, unary plus and minus (method names for the last two are +@ and -@ |
| * / % // | Multiply, divide, modulo and floor division |
| + - | Addition and subtraction |

| | |
|---|---|
| >> << | Right and left bitwise shift |
| & | Bitwise 'AND' |
| ^ \| | Bitwise exclusive 'OR' and regular 'OR' |
| <= < > >= | Equality operators |
| = %= /= //= -= += *=**= | Assignment operators |
| is is not | Identity operators |
| In not in | Membership operators |
| not or and | Logical operators |

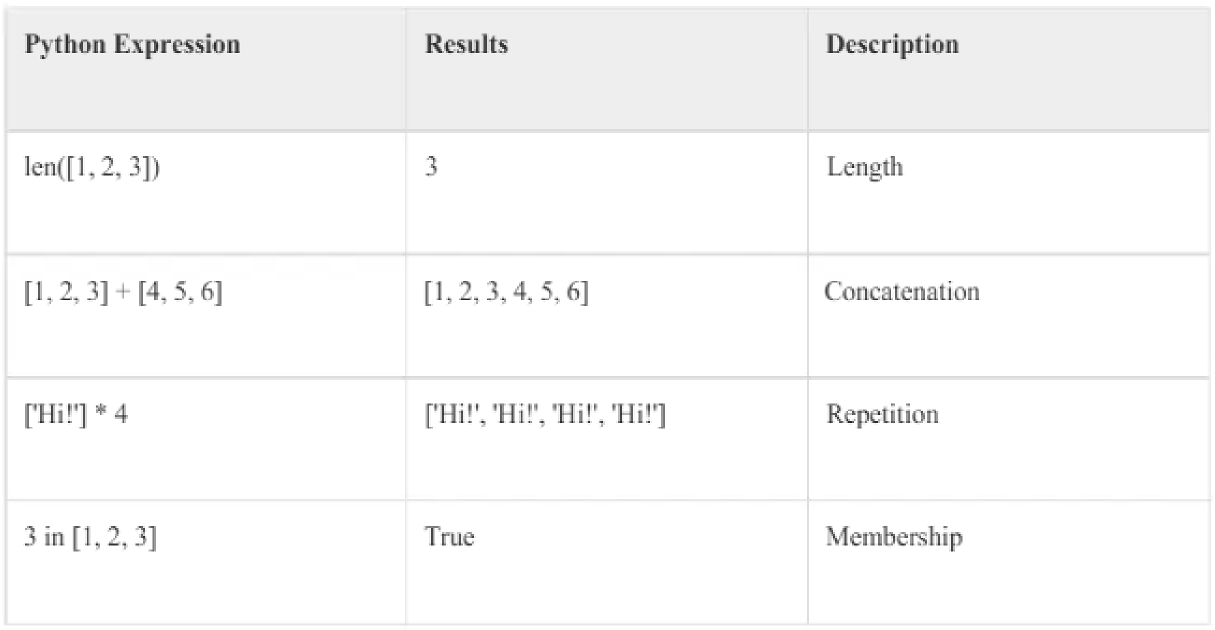
**3.1 LIST**

The list is a most versatile data type available in Python which can be written as a list of comma-separated values (items) between square brackets. Important thing about a list is that items in a list need not be of the same type.

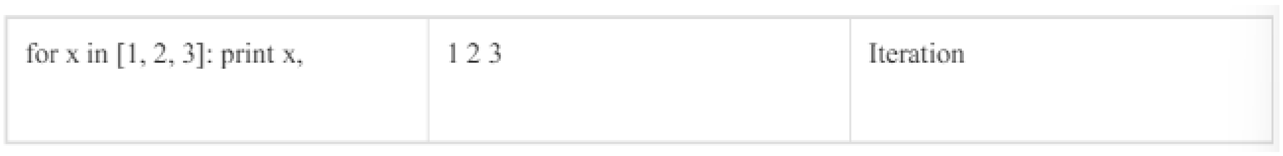
Creating a list is as simple as putting different comma-separated values between square brackets. For example –

**Basic List Operations**

Lists respond to the + and * operators much like strings; they mean concatenation and repetition here too, except that the result is a new list, not a string.



| Python Expression | Results | Description |
|---|---|---|
| len([1, 2, 3]) | 3 | Length |
| [1, 2, 3] + [4, 5, 6] | [1, 2, 3, 4, 5, 6] | Concatenation |
| ['Hi!'] * 4 | ['Hi!', 'Hi!', 'Hi!', 'Hi!'] | Repetition |
| 3 in [1, 2, 3] | True | Membership |



| | | |
|---|---|---|
| for x in [1, 2, 3]: print x, | 1 2 3 | Iteration |

**Built-in List Functions & Methods:**

Python includes the following list functions –

| SN | Function with Description |
|---|---|
| 1 | Cmp(list1, list2)<br><br>Compares elements of both lists. |
| 2 | Len(list)<br><br>Gives the total length of the list. |

| | |
|---|---|
| 3 | Max(list)<br><br>Returns item from the the list with max value |
| 4 | Min(list)<br><br>Returns item from the list with min value. |
| 5 | List(seq)<br><br>Converts a tuple into list. |

Python includes following list methods

| SN | METHODS WITH DESCRIPTION |
|---|---|
| 1 | List. Count (obj) |
| | Appends object obj to list |
| 2 | List. Count (obj)<br><br>Returns count of how many times obj occurs in list |
| 3 | list. Extends (deq)<br><br>Appends the contents of seq to list |

| | |
|---|---|
| 4 | List. Index (obj)<br><br>Returns the lowest index in list that obj appears |
| 5 | List. Insert (index. Obj)<br><br>Inserts object obj into list at offset index |
| 6 | List. pop (obj=list [-1])<br><br>Removes and returns last object or obj from list |
| 7 | List. Remove (obj)<br><br>Removes objects of list in place |
| 8 | List. Reverse ()<br><br>Reverses objects of list in place |
| 9 | List. Sort ([func])<br><br>Sorts objects of list, use compare function if given |

## 3.2 TUPLES

A tuple is a sequence of immutable Python objects. Tuples are sequences, just like lists. The differences between tuples and lists are, the tuples cannot be changed unlike lists and tuples use par entheses, whereas lists use square brackets.

Creating a tuple is as simple as putting different comma-separated values. Optionally we can put these comma-separated values between parentheses also. For example –

The empty tuple is written as two parentheses containing nothing –

To write a tuple containing a single value you have to include a comma, even though there is only one value –

Like string indices, tuple indices start at 0, and they can be sliced, concatenated, and so on.

- **Accessing Values in Tuples:**

To access values in tuple, use the square brackets for slicing along with the index or indices to obtain value available at that index. For example –

When the code is executed, it produces the following result –

tup1[0]: (physics, chemistry, 997, 2000);

print "tup1[0]: ", tup1 [0]

print "tup2[1:5]: ", tup2 [1:5]

tup2[1:5]: [2, 3, 4, 5]

**Updating Tuples:**

Tuples are immutable which means you cannot update or change the values of tuple elements. Weare able to take portions of existing tuples to create new tuples as the following example demonstrates –

When the above code is executed, it produces the following result

Tup1= (12, 34.56),

Tup2= ('abc ', 'XYZ');

(12, 34.56, 'abc', 'xyz')

Tup3= tup1 + tup2; print tup3

**Delete Tuple Elements**

Removing individual tuple elements is not possible. There is, of course, nothing wrong with putting together another tuple with the undesired elements discarded.

To explicitly remove an entire tuple, just use the del statement. For example:

Tup = ('physics', 'chemistry', 1997, 2000);

del tup;

print "After deleting tup:"

print tup

**Basic Tuples Operations:**

| Python Expression | Results | Description |
|---|---|---|
| Len ((1, 2, 3)) | 3 | Length |
| (1, 2, 3) + (4, 5, 6) | (1, 2, 3, 4, 5, 6) | Concatenation |
| (‘Hi!’,) *4 | ('Hi!', 'Hi!', 'Hi!', 'Hi!') | Repetition |
| 3 in (1, 2, 3) | True | Membership |
| For x in (1, 2, 3): print x, | 1 2 3 | Iteration |

**Built- in Tuple Functions**

| SN | Function with Description |
|---|---|
| 1 | **cmp(tuple1, tuple2):** Compares elements of both tuples. |
| **2** | **len(tuple):** Gives the total length of the tuple. |
| **3** | **max(tuple)**: Returns item from the tuple with max value. |

| | |
|---|---|
| | |
| 4 | **min(tuple):** Returns item from the tuple with min value**.** |
| **5** | **Tuple(seq):** Converts a list into a tuple. |

**3.2 DICTIONARY**

Each key is separated from its value by a colon (:); the items are separated by commas, and the whole thing is enclosed in curly braces. An empty dictionary without any items is written with just two curly braces, like this: {}.

Keys are unique within a dictionary while values may not be. The values of a dictionary can be of any type, but the keys must be of an immutable data type such as strings, numbers, or tuples.

**Accessing values in Dictionary:**

To access dictionary elements, you can use the familiar square brackets along with the key to obtain its

value. Following is a simple example –

Result –

Dict {‘Name’ : ‘Zara’, ‘Age’: 7,’class’: ‘First’}

Dict[name’]: Zara

Dict [‘Age’]: 7

Print dic[‘Name’]: “, dict[‘Name’]

Print “dict[‘Age’]:”, dict[ ‘Age’]

**Updating Dictionary**

We can update a dictionary by adding a new entry or a key-value pair, modifying an existing entry, or deleting an existing entry as shown below in the simple example –

result-

```
dict={‘Name’: ‘Zara’, ‘Age’: 7, ‘class’: ‘First’}
dict[‘Age’]=8;    # update existing entry
dict[‘school’]=”DPS School”;    # Add new
dict[‘Age’]: 8
```

**Delete Dictionary Elements**

We can either remove individual dictionary elements or clear the entire contents of a dictionary. You can also delete entire dictionary in a single operation.

To explicitly remove an entire dictionary, just use the **del** statement. Following is a simple example

```
dict= {'Name':'Zara','Age':7,'Class':'First'}
del dict['Name']; # remove entry with key 'Name’
dict.clear (); # remove all entries in dict

del dict; # delete entire dictionary
print "dict ['Age']: “, dict['Age']
 print"dict['School']: “,dict ['School']
```

**Built-in Dictionary Functions & Methods-**

Python includes the following dictionary functions-

| SN | Function with Description |
|---|---|

| | |
|---|---|
| 1 | Cmp (dict1, dict2)<br><br>Compares elements of both dict. |
| 2 | Len(dict)<br><br>Gives the total length of the dictionary. This would be equal to the number of items in the dictionary. |
| 3 | Str(dict)<br><br>Produces a printable string representation of a dictionary |
| 4 | Type(variable)<br><br>Returns the type of the passed variable. If passed variable is dictionary, then it would return a dictionary type. |

Python includes following dictionary methods-

| **SN** | **Methods with Description** |
|---|---|
| 1 | **Dict. Clear ():** Removes all elements of dictionary *dict* |
| 2 | **dict. Copy ():** Returns a shallow copy of dictionary *dict* |

| | |
|---|---|
| 3 | **dict. From keys ()**: Create a new dictionary with keys from seq and values *set* to *value.* |
| 4 | **dict. Get (key, default=None)**:<br>key, returns value or default if key not in dictionary |
| 5 | **Dict. has_ key (key):** Returns *true* if key in dictionary *dict, False* otherwise |
| 6 | **dict. Items ()**: Returns a list of *dict'*s (key, value) tuple pairs |
| 7 | **Dict. Keys ()**: Returns list of dictionary dict's keys |
| 8 | **dict. Set default (key, default=None):** Similar to get (), but will set dict [key] =default if *key* is not already in dict |
| 9 | **Dict. Update(dict2):** Adds dictionary *dict2's* key-values pairs to dict |
| 10 | **Dict. Values: Returns list of dictionary *dict's* values** |

A function is a block of organized, reusable code that is used to perform a single, related action. Functions provide better modularity for your application and a high degree of code reusing. Python gives you many built-in functions like print (), etc. but you can also create your own functions. These functions are called *user-defined functions.*

**Defining a Function**

Simple rules to define a function in Python.

- Function blocks begin with the keyword def followed by the function name and parentheses (()).
- Any input parameters or arguments should be placed within these parentheses. You can also define parameters inside these parentheses.
- The first statement of a function can be an optional statement - the documentation string of the function or *docstring.*
- The code block within every function starts with a colon (:) and is indented.
- The statement return [expression] exits a function, optionally passing back an expression to the caller. A return statement with no arguments is the same as return None.

**Calling a Function**

Defining a function only gives it a name, specifies the parameters that are to be included in the function and structures the blocks of code. Once the basic structure of a function is finalized, you can execute it by calling it from another function or directly from the Python prompt. Following is the example to call

Print me () function –

n[expression]

# Function definition is here def print me(str):

"This prints a passed string into this function" print str

Return;

# Now you can call print me function print me ("I'm first call to user defined function!")

When the above code is executed, it produces the following result –

Print me (Again second can to the same function!

I'm first call user defined function!

Again second call to the same function

**Function Arguments**

You can call a function by using the following types of formal arguments:

- Required arguments
- Keyword arguments
- Default arguments
- Variable- length arguments

**Scope of Variables**

All variables in a program may not be accessible at all locations in that program. This depends on where you have declared a variable.

The scope of a variable determines the portion of the program where you can access a particular identifier. There are two basic scopes of variables in Python −

Global variables Local variables

**Global vs Local Variables**

Variables that are defined inside a function body have a local scope, and those defined outside have a global scope.

This means that local variables can be accessed only inside the function in which they are declared,

whereas global variables can be accessed throughout the program body by
all functions. When you call a function, the variables declared inside it are brought into scope.
Following is a simple example –

Result

Total 0; # This is global

Variable. #Function definition is

Inside the function local total: 30

Here

Outside the function global total: 0

Def sum (arg1, arg2);

# Add both the parameters and return them.

Total= arg1+ arg2; # Here total is local

Variable. Print "Inside the function local total",

Total return total;

Sum (10, 20),

A module allows you to logically organize your Python code. Grouping related code into a module makes the code easier to understand and use. A module is a Python object with arbitrarily named attributes that you can bind and reference. Simply, a module is a file consisting of Python code. A module can define functions, classes and variables. A module can also include runnable code.

**Example:**

The Python code for a module named *aname* normally resides in a file named *aname.py***.** Here's an example of a simple module, support.py

Def print_ func (par):

Print "Hello:", par

**The import statement**

The *Import* has the following syntax:

When the interpreter encounters an import statement, it imports the module if the module is present in the search path. A search path is a list of directories that the interpreter searches before importing a module. For example, to import the module support.py, you need to put the following command at the top of the script −

A module is loaded only once, regardless of the number of times it is imported. This prevents the module execution from happening over and over again if multiple imports occur.

**Packages in Python**

A package is a hierarchical file directory structure that defines a single Python application environment that consists of modules and sub packages and sub-sub packages.

Consider a file *Pots.py* available in *Phone* directory. This file has following line of source code −

Similar way, we have another two files having different functions with the same name as above −

- *Phone/Isdn.py* file having function Isdn ()
- *Phone/G3.py* file having function G3()

Now, create one more file __init__.py in *Phone* directory −

- Phone/__init_.py

To make all of your functions available when you've imported Phone, to put explicit import statements in __init__.py as follows –

Rom Pots import Pots

From Isdn import Isdn

From G3 import G3

After you add these lines to_init_.py, you have all of these classes available when you import the phone package.

**RESULT:**

#Now import your phone

Package import phone

Phone: pots

I'm 3G Phone ()

I'm ISDN Phone

**Phone. Isdn**

In the above example, we have taken example of a single functions in each file, but you can keep multiple functions in your files. You can also define different Python classes in those files and then you can create your packages out of those classes.

This chapter covers all the basic I/O functions available in python.

**Printing to the screen**

The simplest way to produce output is using the print statement where you can pass zero or more expressions separated by commas. This function converts the expressions you pass into a string and writes the result to standard output as follows-

**Result:**

Print “Python is really a great language.”, “isnt it?”

**Reading keyboard Input**

Python is really a great language, isn’t it?

Python provides two built-in functions to read a line of text from standard input, which by default comes from the keyboard. These functions are

- Raw_ input.
- Input

The raw_ input Function

The raw _input([prompt/) function reads one line from standard input and returns it as a string (removing the trailing newline).

```
Str= raw_ input ("Enter your input: ");
print "Received input is: ", str
```

This prompts you to enter any string and it would display same string on the screen. When I typed "Hello Python!", its output is like this

The input function

```
Enter your input. Hello Python
Received input is: Hello python
Str input (“Enter your input.);
Print (“Received input”);
Opening closing files
```

This would produce the following result against the entered input –

```
Enter your input [ x 5 for x in range (2, 10,2);
Received input is: [10,20, 30, 40]
```

The input((prompt]) function is equivalent to raw_ input, except that it assumes the input is a valid Python expression and returns to evaluated result to you.

Until now, you have been regarding and writing to the standard input and output. Now, we will see how to use actual data files.

Python provides basic functions and methods necessary to manipulate files by default. You can do most of the file manipulation using a **file** object.

The open Function

Before you can read or write a file, you have to open it using Python's built-in open () function. This function creates a **file** object**,** which would be utilized to call other supports methods associated with it.

Syntax

File object= open (file_ name [access_ model] [,buffering])

| Modes | Description |
|---|---|
| r | Opens a file for reading only. The file pointer is placed at the beginning of the file. This is the default mode. |
| rb | Opens a file for reading only in binary format. The file pointer is placed at<br>the beginning of the file. This is the default mode. |
| r + | Opens a file for both reading and writing. The file pointer placed at the beginning of the file. |
| rb + | Opens a file for both reading and writing in binary format. The file pointer placed at the beginning of the file. |

| | |
|---|---|
| wb | Opens a file for both writing and reading in binary format. Overwrites the existing file if the file exists. If the file does not exist, creates a new file for reading and writing. |
| a | Opens a file for appending. The file pointer is at the end of the file if the file exists. That is, the file is in the append mode. If the file does not exist, it creates anew file for writing. |
| ab | Opens a file for appending in binary format. The file pointer is at the end of the file if the file exists. That is, the file is in the append mode. If the file does not exist, it creates a new file for writing. |
| a + | Opens a file for both appending and reading. The file pointer is at the end of the file if the file exists. The file opens in the append mode. If the file does not exist, it creates a new file for reading and writing. |
| ab+ | Opens a file for both appending and reading in binary format. The file pointer is at the end of the file if the file exists. The file opens in the append mode. If the file does not exist, it creates a new file for reading and writing. |

Here are parameters details:

- **file_ name:** The file_ name argument is a string value that contains the name of the file that you want to access.

- **access_ mode:** The access_ mode determines the mode in which the file has to be opened, i.e., read, write, append, etc. A complete list of possible values is given below in the table. This is optional parameter and the default file access mode is read (r).

- **buffering:** If the buffering value is set to 0, no buffering takes place. If the buffering value is l, line buffering is performed while accessing a file. If you specify the buffering value as an integer greater than 1, then buffering action is performed with the indicated buffer size. If negative, the buffer size is the system default (default behaviour**).**

  Here is a list of the different modes of opening a file −

  The *File* Object Attributes Once a file is opened and you have one *File* object, you can get various information related to that file.

| **Attribute** | **Description** |
|---|---|
| File. closed | Returns true if file is closed, false otherwise. |
| file mode | Returns access mode with which file was opened. |
| file.name | Returns name of file. |
| File. Soft space | Returns false if space explicitly required with print, true otherwise. |

Here is a list of all attributes related to file object.
Example

This produces the following result-

# open a file

Fo = open (“ foo.txt”, “wb”)

Name of the file:

Printx “Name of the file:”,

Closed of the file:

Fo. Name print “ closed or not:”,

Fo. Closed print “ opening mode:”,

Fo.mode

The close () Method

Opening mode:

The close() method of a *file* object flushes any unwritten information and closes the file object, after which no more writing can be done.Python automatically closes a file when the reference object of afile is reassigned to another file. It is a good practice to use the close() method to close a file.

Syntax

Example

fileObject.close();

Result-

# open a fiile

Reading and Writing Files

Fo= open(“ foo.txt”,” wb”)

Name of the file: foo.txt

The file object provides a set of access methods to make our lives easier. We would see how to

Print “Name of the file:”,

Use read() and write() methods to read and write files.

Fo.name # closed oened file

The write() Method

Fo.close()

The *write()* method writes any string to an open file. It is important to note that Python strings can have binary data and not just text.The write() method does not add a newline character ('\n') to the end of the string

**Syntax**

FileObject. Write(string);

Here, passed parameter is the content to be written into the opened file.

**Example**

# open a file

The above method would create *foo.txt* file and would write given content in that file and finally itwould close that file. If you would open this file, it would have following content.

Fo= open(“ foo.txt”,”wb”)

The read () Method reads a string from an open file. It is is important to note that Python strings can language. Yeah its have binary data. apart from text data.

# close opend

**Syntax**

File of close()

Fileobject.read([count],

Here passed parameter is the number of bytes to be read from the opened file. This method startsreading from the beginning of the file and if*Count* is missing, then it tries to read as much as possible,maybe until the end of file.

**Example**

**Let's take a file *foo.txt* , which we created above**

**# open a file**

**Fo= =open("foo.txt","r+")str=fo.read(10);**

**print"Read String is : ",str**

This producs the following result-

# close opend file

Fo.close()

File positions

Read String is: Python is

The *tell()* method tells you the current position within the file; in other words, the next read or writewill occur at that many bytes from the beginning of the file.

The *seek(offset[, from])* method changes the current file position. The *Offset* argument indicates the number of bytes to be moved.

If *from* is set to 0, it means use the beginning of the file as the reference position and 1 means use thecurrent position as the reference position and if it is set to 2 then the end of the file would be taken asthe reference position.

**Example**

Let us take a file *foo.txt,* , which we created above.

# open a file

Fo= open (“foo.txt”,”r+”) str=fo.read(10);

print"Read String is : ",str

# Check current

position position=fo.tell();

print"Current file position : ", position

This produces the following result –

# Reposition pointer at the beginning once

Again: position= fo seek(0.0);

Read String is. Python

Python os module; provides methods that help you perform file-processingg operations, such as renamig and deleting files.

Str=fo.read(10);

Is current file position:

Print “Again read string is:”,

10

To use this module you need to import it first and then you can call any related functions.

Str #closed opend file

To rename() Method

Fo. Close()

The rename() method takes two arguments, the current filename and the new filename.

**Syntax**

Example

Os. Rename( current_ file_name, new_file_name)

Following is the example to rename an existing file *test1.txt:*

Import os

# Rename a file from test1.txt to test2.txt

The *remove()* Method

Os. Rename ( test. Txt", "test2.txt")

You can use the *remove()* method to delete files by supplying the name of the file to be deleted as the argument.

**Syntax**

Example

Os. Remove(file_name)

Following is the example to delete an existing file *test2.txt* –

Directories in Python

#!/USI/bin/

Python import

All files are contained within various directories, and Python has no problem handling these too.

The **os** module has several methods that help you create, remove, and change directories.

The *mkdir()* Method

You can use the *mkdir()* method of the **os** module to create directories in the current directory. You need to supply an argument to this method which contains the name of the directory to be created.

# Delete file test2.txt

**Syntax**

Example

Os.mkdir(" newdir")

Following is the example to create a directory *test* in the current directory –

#!/usr/bin/ python

The chdir() Method

Import os

You can use the *chdir()* method to change the current directory. The chdir() method takes anargument, which is the name of the directory that you want to make the current directory.

# Create a directory

**Syntax**

**“test” os.mkdir(“test”)**

Example

Os.chdir(“newdir”)

Following is the example to go into "/home/newdir" directory –

The *getcwd()* Method

#!/usr/bin/ python

The *getcwd()* method displays the current working directory.

Python import

**Syntax**

Os

Os.getcwd()

Example

Following is the example to give current directory-

# changing a directory to

Import os

# This would give location of the current directory

Os.getcwd()

| EXCEPTION NAME | DESCRIPTION |
|---|---|
| Exception | Base class for all exceptions. |
| StopIteration | Raised when the next() method of an iteratordoes not point to any object. |
| SystemExit | Raised by the sys.exit() function. |
| Standard Error | Base class for all built-in exceptions exceptStopIteration and SystemExit. |
| Arithmetic Error | Base class for all errors that occur for numericcalculation. |
| Overflow Error | Raised when a calculation exceeds maximumlimit for a numeric type. |
| FloatingPoint Error | Raised when a floating point calculation fails. |

| | |
|---|---|
| ZeroDivision Error | Raised when division or modulo by zero takes place for all numeric types. |
| Assertion Errror | Raised in case of failure of theAssert statement. |
| AttributeError | Raised in case of failure of attribute referenceor assignment. |
| EOFError | Raised when there is no input from either theraw_input() or input() function and the end of file is reached. |
| ImportError Raised | Raised when an import statement fails. |
| Keyboard Interrupt. | Raised when the user interrupts programexecution, usually by pressing Ctrl+c. |
| LookupError | Base class for all lookup errors. |

| | |
|---|---|
| IndexError | Raised when an index is not found in asequence. |
| KeyError | Raised when the specified key is not found inthe dictionary. |
| NameError | Raised when an identifier is not found in thelocal or global namespace. |
| UnboundLocalError | Raised when trying to access a local variablein a function or method but no value has been assigned to it. |
| EnvironmentError | Base class for all exceptions that occur outside the Python environment. |
| IOError | Raised when an input/ output operation fails,such as the print statement or the open()function when trying to open a file that doesnot exist |
| IOError | Raised for operating system-related errors. |

| | |
|---|---|
| SyntaxError | Raised when there is an error in Python syntax. |
| IndentationError | Raised when indentation is not specified properly. |
| SystemError | Raised when the interpreter finds an internal problem, but when this error is encounteredthe Python interpreter does not exit. |
| SystemExist | Raised when Python interpreter is quit byusing the sys.exit() function. If not handled inthe code, causes the interpreter to exit. |
| TypeError | Raised when an operation or function isattempted that is invalid for the specified data type. |
| ValueError | Raised when the built-in function for a data type has the valid type of arguments, but the arguments have invalid values specified. |
| RuntimeError | Raised when a generated error does not fall into any category. |

| | |
|---|---|
| NotImplementedError | Raised when an abstract method that needs to be implemented in an inherited class is not actually implmented. |

The rmdir() Method

The rmdir() method the deletes directory, which is passed as an argument in the method.

Before removing a directory, a directory, all the contents in it should be removed.

**Syntax:**

Example:

Os.rmdir('dirname')

Following is the example to remove "/tmp/test" directory. It is required to give fuly qualified name of the directory, otherwise it would search for that directory n the current directory.

File& Directory Related Methods

Import os

There are three important sources, which provide a wide range of utility methods to handle and manipulate files& directories on windows and unix operating syste,. They are as follows-

#This would remove "/tmp/test"

Directory.os.rmdir("/tmp/test")

- File object Methods: The file object provides functions to manipulate files.
- OS Object Methods: This provides methods to process files as well as directories.

Python provides two very important features to handle any unexpected error in your Python programs and to add debugging capabilities in them –

- **Exception Handling**: This would be covered in this tutorial. Here is a list standard Exceptions available in Python: Standard Exceptions

- **Assertions:** This would be covered in Assertions in Python

List of Standard Exceptions-

**What is Exception?**

An exception is an event, which occurs during the execution of a program that disrupts the normal flow of the program's instructions. In general, when a Python script encounters a situation that it cannot cope with, it raises an exception. An exception is a Python object that represents an error.

When a Python script raises an exception, it must either handle the exception immediately otherwise it terminates and quits.

**Handling an exception**

If you have some suspicious code that may raise an exception, you can defend your program by placing the suspicious code in a **try:** block. After the try: block, include an **except:** statement, followed by a block of code which handles the problem as elegantly as possible.

The Python standard for database interfaces is the Python DB-API. Most Python database interfaces adhere to this standard.

You can choose the right database for your application. Python database API supports a wide range of database servers such as-

- GadFly
- msql
- MySQL
- PostgreSQL
- Microsoft SQL Server 2000
- Informix
- Interbase
- Oracle
- Sybase

The DB API provides rovides a minimal standard for working with databases using Python structures andsyntax wherever possible. This API includes the following:

- Importing the API module.
- Acquiring a connection with the database.
- Issuing SQL statements and stored procedures.
- Closing the connection

# 6. ALGORITHMS

# 6. ALGORITHMS

**Decision tree classifiers**

Decision tree classifiers are used successfully in many diverse areas. Their mostimportant feature is the capability of capturing descriptive decision making knowledgefrom the supplied data. Decision tree can be generated from training sets. The procedure for such generation based on the set of objects (S), each belonging to one of the classes C1, C2, …, Ck is as follows:

**Step 1:**

If all the objects in S belong to the same class, for example Ci, the decision treefor S consists of a leaf labeled with this class

**Step :** Otherwise, let T be some test with possible outcomes O1, O2,…, On. Eachobject in S has one outcome for T so the test partitions S into subsets S1, S2,… Snwhere each object in Si has outcome Oi for T. T becomes the root of the decision treeand for each outcome Oi we build a subsidiary decision tree by invoking the same procedure recursively on the set Si.

**Gradient boosting**

Gradient boosting is a machine learning technique used in regression and classification tasks, among others. It gives a prediction model in theform of anensembleof weak prediction models,

which are typicallydecision trees.When a decision tree is the weak learner, the resulting algorithm is called gradient- boosted trees; it usually outperformsrandom forest. A gradient-boosted trees model is built in a stage-wise fashion as in other boostingmethods, but it generalizes the other methods by allowing optimization of an arbitrarydifferentiable loss function.

**K-Nearest Neighbors (KNN)**

- Simple, but a very powerful classification algorithm
- Classifiers based on a similarity measure
- Non- parametric
- Lazy learning
- Does not “learn” until the tset example is given
- Whenevr we have a new data to classify, we find its K-nearest neighbors from the training data

Example

- Training dataset consists of k-closest examples in feature space
- Feature space means, space with categorization variables (non-metric variables)
- Learning based on instances, and thus also works lazily because instance close to the input vector for test or prediction may take time to occur in the training dataset

## **Logistic regression Classifiers**

Logistic regression analysis studies the association between a categorical dependent variable and a set of independent (explanatory) variables. The name logistic regression is used when the dependent variable has only two values, such as 0 and 1 or Yes and No. The name multinomial logistic regression is usually reserved for the case when the dependent variable has three or more unique values, such as Married, Single, Divorced, or Widowed. Although the type of data used for the dependent variable is different from that of multiple regression, the practical use of the procedure is similar.

Logistic regression competes with discriminant analysis as a method for analyzing categorical-response variables. Many statisticians feel that logistic regression is more versatile and better suited for modeling most situations than is discriminant analysis. This is because logistic regression does not assume that the independent variables are normally distributed, as discriminant analysis docs.

This program computes binary logistic regression and multinomial logistic regression on both numeric and categorical independent variables. It reports on the regression equation as well as the goodness of fit, odds ratios, confidence limits, likelihood, and deviance. It performs a comprehensive residual analysis including diagnostic residual reports and plots. It can perform an independent variable subset selection search, looking for the best regression model with the fewest independent variables. It provides confidence intervals on predicted values and provides ROC curves to help determine the best cutoff point for classification. It allows you to validate your results by automatically classifying rows that are not used during the analysis.

## **Naïve Bayes**

The naive bayes approach is a supervised learning method which is based on a simplistic hypothesis: it assumes that the presence (or absence) of a particular feature of a class is unrelated to the presence (or absence) of any other feature.

Yet, despite this, it appears robust and efficient. Its performance is comparable to other supervised learning techniques. Various reasons have been advanced in the literature. In this tutorial, we highlight an explanation based on the representation bias. The naive bayes classifier is a linear classifier, as well as linear discriminant analysis, logistic regression or linear SVM (support vector machine). The difference lies on the method of estimating the parameters of the classifier (the learning bias).

While the Naive Bayes classifier is widely used in the research world, it is not widespread among practitioners which want to obtain usable results. On the one hand, the researchers found especially it is very easy to program and implement it, its parameters are easy to estimate, learning is very fast even on very large databases, its accuracy is reasonably good in comparison to the other approaches. On the other hand, the final users do not obtain a model easy to interpret and deploy, they does not understand the interest of such a technique.

Thus, we introduce in a new presentation of the results of the learning process. Theclassifier is easier to understand, and its deployment is also made easier. In the first partof this tutorial, we present some theoretical aspects of the naive bayes classifier. Then,we implement the approach on a dataset with Tanagra. We compare the obtainedresults (the parameters of the model) to those obtained with other linear approachessuch as the logistic regression, the linear discriminant analysis and the linear SVM. Wenote that the results are highly consistent. This largely explains the good performanceof the method in comparison to others. In the second part, we use various tools on thesame dataset (Weka 3.6.0, R 2.9.2, Knime 2.1.1, Orange 2.0b and RapidMiner 4.6.0). We try aboveall to understand the obtained results.

## Random Forest

Random forests or random decision forests are an ensemble learning method for classification, regression and other tasks that operates by constructing a multitude of decision trees at training time. For classification tasks, the output of the random forestis the class selected by most trees. For regression tasks, the mean or average predictionof the individual trees is returned. Random decision forests correct for decision trees'habit of overfitting to their training set. Random forests generally outperform decisiontrees, but their accuracy is lower than gradient boosted trees. However, datacharacteristics can affect their performance.

The first algorithm for random decision forests was created in 1995 by Tin Kam Ho[1]using the random subspace method, which, in Ho's formulation, is a way to implementthe "stochastic discrimination" approach to classification proposed by EugeneKleinberg. An extension of the algorithm was developed by Leo Breiman and AdeleCutler, who registered "Random Forests" as a trademark in 2006 (as of 2019, owned byMinitab, Inc.). The extension combines Breiman's "bagging" idea and random selectionof features, introduced first by Ho[1] and later independently by Amit and Geman[13] in order to construct a collection of decision trees with controlled variance.Random forests are frequently used as "blackbox" models in businesses, as they generatereasonable predictions across a wide range of data while requiring little configuration.

## SVM

In classification tasks a discriminant machine learning technique aims at finding, basedon an

*independent and identically distributed* (*iid* ) training dataset, a discriminantfunction that can correctly predict labels for newly acquired instances. Unlikegenerative machine learning approaches, which require computations of conditional probability distributions, a discriminant classification function takes a data point *x* andassigns it to one of the different classes that are a part of the classification task. Less powerful than generative approaches, which are mostly used when prediction involvesoutlier detection, discriminant approaches require fewer computational resources andless training data, especially for a multidimensional feature space and when only posterior probabilities are needed. From a geometric perspective, learning a classifier isequivalent to finding the equation for a multidimensional surface that best separates thedifferent classes in the feature space.

**SVM is a discriminant :** technique, and, because it solves the convex optimization problem analytically, it always returns the same optimal hyperplane parameter—in contrast to *genetic algorithms* (*GAs*) or *perceptrons*, both of which are widely used for classification in machine learning. For perceptrons, solutions are highly dependent on the initialization and termination criteria. For a specific kernel that transforms the data from the input space to the feature space, training returns uniquely defined SVM model parameters for a given training set, whereas the perceptron and GA classifier models are different each time training is initialized. The aim of GAs and perceptrons is only tominimize error during training, which will translate into several hyperplanes' meeting this requirement.

# 7. SYSTEM TESTING

# 7. SYSTEM TESTING

The purpose of testing is to discover errors. Testing is the process of trying to discover everyconceivable fault or weakness in a work product. It provides a way to check the functionality of components, sub-assemblies, assemblies and/or a finished product It is the process of exercisingsoftware with the intent of ensuring that the Software system meets its requirements and user expectations and does not fail in an unacceptable manner. There are various types of tests. Each testtype addresses a specific testing requirement.

## 7.1TYPES OF TESTS:

### Unit Test

The purpose of testing is to discover errors. Testing is the process of trying to discover everyconceivable fault or weakness in a work product. It provides a way to check the functionality of components, sub-assemblies, assemblies and/or a finished product It is the process of exercisingsoftware with the intent of ensuring that the Software system meets its requirements and user expectations and does not fail in an unacceptable manner. There are various types of tests. Each testtype addresses a specific testing requirement.

**7.1TYPES OF TESTS:**

### Unit Testing

Unit testing involves the design of test cases that validate that the internal program logic isfunctioning properly, and that program inputs produce valid outputs. All decision branches and internal code flow should be validated. It is the testing of individual software units of theapplication .it is done after the completion of an individual unit before integration. This is a structuraltesting, that relies on knowledge of its construction and is invasive. Unit tests perform basic tests atcomponent level and test a specific business process, application, and/or system configuration. Unittests ensure that each unique path of a business process performs accurately to the documentedspecifications and contains clearly defined inputs and expected results.

### Integration Testing

Integration tests are designed to test integrated software components to determine if theyactually run as one program. Testing is event driven and is more concerned with the basic outcome of screens or fields. Integration tests demonstrate that although the components were individuallysatisfaction, as shown by successfully unit testing, the combination of components is correct andconsistent. Integration testing is specifically aimed at exposing the problems that arise from thecombination of components.

**Functional Testing**

Functional tests provide systematic demonstrations that functions tested are available as specified bythe business and technical requirements, system documentation, and user manuals.

Functional testing is centered on the following items:

Valid Input: identified classes of valid input must be accepted.

Invalid Input: identified classes of invalid input must be rejected.

Functions: identified functions must be exercised.

Output: identified classes of application outputs must be exercised.

Systems/Procedures: interfacing systems or procedures must be invoked.

Organization and preparation of functional tests is focused on requirements, key functions, or specialtest cases. In addition, systematic coverage pertaining to identify Business process flows; data fields, predefined processes,and successive processes must beconsidered for testing.

Before functional testing is complete, additional tests are identified and the effective value of current tests is determined.

**System Test**

System testing ensures that the entire integrated software system meets requirements. It tests aconfiguration to ensure known and predictable results. An example of system testing is theconfiguration-oriented system integration test. System testing is based on process descriptions andflows, emphasizing pre-driven process links and integration points.

**White Box Testing**

White Box Testing is a testing in which in which the software tester has knowledge of the inner workings, structure and language of the software, or at least its purpose. It is purpose. It is used to testareas that cannot be reached from a black box level.

**Black Box Testing**

Black Box Testing is testing the software without any knowledge of the inner workings, structure or language of the module being tested. Black box tests, as most other kinds of tests, must be written from a definitive source document, such as specification or requirements document such as specification or requirements document. It is a testing in which the software under test is treated, as a black box .you cannot “see” into it. The test provides inputs and responds to outputs without consider in how the software works.

## 7.2System Testing

**TESTING METHODOLOGIES:**

The following are the Testing Methodologies:

- Unit Testing.
- Integration Testing.
- User Acceptance Testing.
- Output Testing.
- Validation Testing.

**Unit Testing**

Unit testing focuses verification effort on the smallest unit of Software design that is the module. Unit testing exercises specific paths in a module's control structure to ensure complete coverage and maximum error detection. This test focuses on each module individually, ensuring that it functions properly as a unit. Hence, the naming is Unit Testing.

During this testing, each module is tested individually and the module interfaces are verified for the consistency with design specification. All important processing path are tested for theexpected results. All error handling paths are also tested.

**Integration Testing**

Integration testing addresses the issues associated with the dual problems of verificationand program construction. After the software has been integrated a set of high order tests are conducted. The main objective in this testing process is to take unit tested modules and builds a program structure that has been dictated by design.

The following are the types of Integration Testing:

**1)Top Down Integration**

This method is an incremental approach to the construction of program structure. Modules are integrated by moving downward through the control hierarchy, beginning with the main program module. The module subordinates to the main program module are incorporatedinto the structure in either a depth first or breadth first manner.

In this method, the software is tested from main module and individual stubs are replaced when the test proceeds downwards.

**2.Bottom-up Integration**

This method begins the construction and testing with the modules at the lowest level in the program structure. Since the modules are integrated from the bottom up, processing required for modules subordinate to a given level is always available and the need for stubs is eliminated. The bottom up integration strategy may be implemented with the following steps:

- The low-level modules are combined into clusters into clusters that perform a specific Software sub-function.
- A driver (i.e.) the control program for testing is written to coordinate test case input and output.
- The cluster is tested.
- Drivers are removed and clusters are combined moving upward in the program structure

The bottom up approaches tests each module individually and then each module is integrated with a main module and tested for functionality.

## OTHER TESTING METHODOLOGIES

### User Acceptance Testing

User Acceptance of a system is the key factor for the success of any system. The system under consideration is tested for user acceptance by constantly keeping in touch with the prospective system users at the time of developing and making changes wherever required. The system developed provides a friendly user interface that can easily be understood even by a person who is new to the system.

**Output Testing**

After performing the validation testing, the next step is output testing of the proposed system, since no system could be useful if it does not produce the required output in the specified format. Asking the users about the format required by them tests the outputs generated or displayed by the system

under consideration. Hence the output format is considered in 2 ways – one is on screen and another in printed format.

**Validation Checking**

Validation checks are performed on the following fields.

**Text Field:**

The text field can contain only the number of characters lesser than or equal to its size. The text fields are alphanumeric in some tables and alphabetic in other tables. In Correct entry always flashes and error message.

**Numeric Field:**

The numeric field can contain only numbers from 0 to 9. An entry of any character flashes an error messages. The individual modules are checked for accuracy and what it has to perform. Each module is subjected to test run along with sample data. The individually tested modules are integrated into a single system. Testing involves executing the real data information is used in the program the existence of any program defect is inferred from the output. The testing should be planned so that all the requirements are individually tested.

A successful test is one that gives out the defects for the inappropriate data and produces and output revealing the errors in the system.

**Preparation of Test Data**

Taking various kinds of test data does the above testing. Preparation of test data plays avital role in the system testing. After preparing the test data the system under study is tested using that test data. While testing the system by using test data errors are again uncovered and corrected by using above testing steps and corrections are also noted for future use.

**Using Live Test Data:**

Live test data are those that are actually extracted from organization files. After a system is partially constructed, programmers or analysts often ask users to key in a set of data from their normal activities. Then, the systems person uses this data as a way to partially test the system. In

other instances, programmers or analysts extract a set of live data from the files and have them entered themselves.

It is difficult to obtain live data in sufficient amounts to conduct extensive testing. And, although it is realistic data that will show how the system will perform for the typical processing requirement, assuming that the live data entered are in fact typical, such data generally will not test all combinations or formats that can enter the system. This bias toward typical values then does not provide a true systems test and in fact ignores the cases most likely to cause system failure.

**Using Artificial Test Data:**

Artificial test data are created solely for test purposes, since they can be generated to test all combinations of formats and values. In other words, the artificial data, which can quickly prepared by a data generating utility program in the information systems department, make possible the testing of all login and control paths through the program.

The most effective test programs use artificial test data generated by persons other than those who wrote the programs. Often, an independent team of testers formulates a testing plan, using the systems specifications.
The package “Virtual Private Network” has satisfied all the requirements specified as per software requirement specification and was accepted.

## USER TRAINING

Whenever a new system is developed, user training is required to educate them about the working of the system so that it can be put to efficient use by those for whom the system has been

primarily designed. For this purpose the normal working of the project was demonstrated the prospective users. Its working is easily understandable and since the expected users are people who have good knowledge of computers, the use of this system is very easy.

## MAINTAINENCE

This covers a wide range of activities including correcting code and design errors. To reduce the need for maintenance in the long run, we have more accurately defined the user’s requirements during the process of system development. Depending on the requirements, this system has been developed to satisfy the needs to the largest possible extent. With development in technology, it may be possible to add many more features based on the requirements in future. The coding and designing is simple and easy to understand which will make maintenance easier.

## TESTING STRATEGY:

A strategy for system testing integrates system test cases and design techniques into a well planned series of steps that results in the successful construction of software. The testing strategy must co-operate test planning, test case design, test execution, and the resultant data collection and evaluation. A strategy for software testing must accommodate low-level tests that are necessary to verify that a small source code segment has been correctly implemented as well as high level tests that validate major system functions against user requirements.

Software testing is a critical element of software quality assurance and represents the ultimate review of specification design and coding. Testing represents an interesting anomaly for the

software. Thus, a series of testing are performed for the proposed system before the system is ready for user acceptance testing.

## SYSTEM TESTING:

Software once validated must be combined with other system elements (e.g. Hardware, people, database). System testing verifies that all the elements are proper and that overall system function performance is achieved. It also tests to find discrepancies between the system and its original objective, current specifications and system documentation.

## UNIT TESTING:

In unit testing different are modules are tested against the specifications produced during the design for the modules. Unit testing is essential for verification of the code produced during the coding phase, and hence the goals to test the internal logic of the modules. Using the detailed design description as a guide, important Conrail paths are tested to uncover errors within the boundary of the modules. This testing is carried out during the programming stage itself. In this type of testing step, each module was found to be working satisfactorily as regards to the expected output from the module.

In Due Course, latest technology advancements will be taken into consideration. As part of technical build-up many components of the networking system will be generic in nature so that future projects an either use or interact with this. The future holds a lot to offer to the development and refinement of this page.

# 8. RESULT

# 8. RESULT

The real-time customized digital coupon issuance system was highly effective in enhancing the shopping experience and boosting mall performance. The system successfully integrated all components, delivering accurate and timely personalized coupons to shoppers through a user-friendly mobile app. Performance metrics showed fast response times and the ability to handle high data volumes. Shoppers reported increased engagement and convenience, leading to higher foot traffic and sales for retailers. The system also demonstrated strong reliability and security, with minimal downtime and robust data protection. Overall, the system achieved its goals of improving shopper satisfaction and increasing mall revenue.

# 9. CONCLUSION

# 9.CONCLUSION

We identified previous e-commerce marketing approaches to derive user behaviour prediction. A deep learning method for real time customer churn prediction showed an appropriate result. We applied our research to online shopping mall to raise conversion rate and sales. To check whether our experiment carry out monetary value, we developed a framework to measure the sales amount when used with segment model and personalized recommended digital coupon. We found that our model(scenario1) shows the best results. We found it is suitable for e-commerce online shopping mall to raise conversion rate and sales. Our study empirically that marketing, which was a field of management, could be solved more efficiently and quickly by applying big data and deep leaning technology.

# 10.REFERENCES

# 10. REFERENCES

[1] P. Naval and N. Pandey, ‘‘What makes a consumer redeem digital coupons? Behavioural insights from grounded theory approach,’’ *J. Promotion Manage.*, vol. 28, no. 3, pp. 205–238, 2021.

[2] C. Hung and C. F. Tsai, ‘‘Market segmentation based on hierarchical self organizing map for markets of multimedia on demand,’’ *Expert Syst. With Appl.*, vol. 34, pp. 780– 787, Jan. 2008.

[3] G. Nie, ‘‘Finding the hidden pattern of credit card holder’s churn: A case of China,’’ in *Proc. Int. Conf. Comput. Sci.* Cham, Switzerland: Springer, 2009, pp.561–569.

[4] A. D. Athanassopoulos, ‘‘Customer satisfaction cues to support market segmentation and explain switching behaviour,’’ *J. Bus. Res.*, vol. 47, no. 3, pp. 191– 207, Mar. 2000.

[5] C. Hung and C. F. Tsai, ‘‘Market segmentation based on hierarchical self organizing map for markets of multimedia on demand,’’*Expert Syst. With Appl.*, vol. 34, pp. 780– 787, Jan. 2008.

[6] H.-S. Kim and H. Seung-Woo, ‘‘A two-dimensional customer loyalty segment-based customer churn prediction methodology,’’ *Intell. Inf. Res.*, vol. 26, no 4, pp. 111-126, 2020.

[7] R.M. Gubela, S. Lessmann, and S. Jaroszewicz, ‘‘Response transformation and profit decomposition for revenue uplift modeling,’’ *Eur. J. Oper. Res.*, vol. 283, no. 2, pp. 647–661, Jun. 2020.

[8] M.-S. Chang, H. Kim, and Joong, ‘‘A customer segmentation scheme base on big data in a bank,’’ *J. Digit. Contents Soc.*, vol. 19, no. 1, pp. 85–91, 2018.

[9] N. Chang, ‘‘Improving the effectiveness of customer classification models: A pre-segmentation approach,’’ *Inf. Syst. Rev.*, vol. 7, no. 2, pp. 23– 40, 2005.

[10] C. -F. Tsai and Y.-H. Lu, ‘‘Customer churn prediction by hybrid neural networks,’ *Expert Syst. Appl.*, vol. 36, no. 10, pp. 12547–12553, Dec. 2009.

[11] Y. Xie, X. Li, E. W. T. Ngai, and W. Ying, ‘‘Customer churn prediction using improved balanced random forests,’’ *Expert Syst. Appl.*, vol. 36, no. 3, pp. 5445– 5449, Apr. 2009.

[12] S.-Y. Hung, D. C. Yen, and H.-Y. Wang, ‘‘Applying data mining to telecom churn management,’’ *Expert Syst. Appl.*, vol. 31, no. 3, pp. 515–524, Oct. 2006.

[13] J. Wen and W. Zhou, ‘‘An improved item-based collaborative filtering algorithm based on clustering method,’’ *J. Comput. Inf. Syst.*, vol. 8, no. 2, pp. 571–578, 2012.

[14] M. Pham and Cuong, ‘‘A clustering approach for collaborative filtering recommendation using social network analysis,’’ *J. Univers. Comput. Sci.*, vol. 17, pp.583–604, Feb. 2011

[15] W. Jo-Ting, L. Shih-Yen, and W. Hsin-Hung, ‘‘A review of the application of RFM model,’’

*African J. Bus. Manage.*, vol. 4, no. 19, pp. 4199–4206, 2010.

[16] J. T. Wei, S.-Y. Lin, Y.-Z. Yang, and H.-H. Wu, ‘‘The application of data mining and RFM model in market segmentation of a veterinary hospital,’’*J. Statist. Manage. Syst.*, vol. 22, no. 6, pp. 1049–1065, Aug. 2019.

[17] M. Pakyurek, M. S. Sezgin, S. Kestepe, B. Bora, R. Duzagac, and O. T. Yildiz, ‘‘Customer clustering using RFM analysis,’’ in *Proc. 26 the Signal Process. Commun. Appl. Conf. (SIU)*, May 2018, p. 2.

[18] P. A. Sarvari, A. Ustundag, and H. Takci, ‘‘Performance evaluation of different customer segmentation approaches based on RFM and demographics analysis,’’ *Kybernetes*, vol. 45, no. 7, pp. 1129–1157, Aug. 2016.

[19] F. Tian, ‘‘Learning deep representations for graph clustering,’’ in *Proc. AAAI Conf. Artif. Intell.*, 2014, pp. 1293–1299.

[20] J. Girshick and R. Farhadi, ‘‘Unsupervised deep embedding for clustering analysis,’’ in *Proc. Int. Conf. Mach. Learn.*, 2016, pp. 478–487.

[21] K. Tian, S. Zhou, and J. Guan, ‘‘Deep cluster: A general clustering framework based on deep learning,’’ in*Proc. Joint Eur. Conf. Mach. Learn. Knowl. Discovery databases*. Cham, Switzerland: Springer, 2017, pp. 809–825.

[22] S. Oh, E Lee, J. Woo, and H. K. Kim, ‘‘Constructing and evaluating a churn prediction model using classification of user types in MMORPG,’’ *KIISE Trans. Comp. Practices*, vol. 24, no. 5, pp. 220–226, May 2018.

[23] J. Kawale, A. Pal, and J. Srivastava, ‘‘Churn prediction in MMORPGs: A social influence sbased approach,’’ in *Proc. Int. Conf. Comput. Sci. Eng.*, 2009, pp. 423–428.

*[24]* S. Renjith, ‘‘B2C E-Commerce customer churn management: Churn detection using support vector machine and personalized retention using hybrid recommendations,’’ *Int. J. Future Revolution Comput. Sci. Commun. Eng.*, vol. 3, no. 11, pp. 34–39, 2017.

[25] B. Mishachandar and K. A. Kumar, ‘‘Predicting customer churn using targeted proactive retention,’’ *Int. J. Eng. Technol.*, vol. 7, no. 2, p. 69, Aug. 2018.