Chapter 1

Introduction

1.1 Introduction

Product Demand forecasting is the process of estimating the demand for a product or services in the future based on historical data, market trends, and other relevant factors. The purpose of product demand forecasting is to ensure that a company can meet the demand for its product while minimizing the risk of overproduction or underproduction. There are several methods for product demand forecasting, including qualitative and quantitative methods.

Qualitative methods rely on expert opinions, market research, and surveys to estimate demand, while quantitative methods use statistical models, time series analysis, and machine learning algorithms. However, forecasting demand is not without its challenges. External factors such as changes in consumer behavior, market trends, and economic conditions can all impact demand.

Therefore, demand forecasting models must be flexible and adaptable to changes in the market. In this report, we will export the importance of product demand forecasting, the methods used to forecast demand, the challenges of demand forecasting, and the benefits of accurate forecasting for businesses.

1.2 Choice of Topic with reasoning/Need of Project

The purpose of our project is to develop a robust Product Demand Prediction system to help businesses optimize inventory management. By leveraging machine learning techniques on historical sales data, market trends, and external factors, our solution aims to accurately forecast product demand. This predictive capability will empower organizations to minimize holding costs, avoid stockouts, and enhance overall supply chain efficiency.

Ultimately, the project seeks to improve decision-making, increase customer satisfaction through product availability, and provide a competitive advantage in the dynamic market landscape.

1.3 Problem Statement

Inefficient inventory management poses a significant challenge for businesses, leading to increased costs and missed sales opportunities. Accurate product demand prediction is crucial for optimizing stock levels.

Our objective is to develop a machine learning-based solution that leverages historical sales data, market trends, and external factors to predict demand accurately. This system aims to enhance inventory efficiency, reduce holding costs, and improve overall supply chain performance by providing real-time updates, scalability, and a user-friendly interface for proactive decision-making.

Chapter 2

Proposed System

2.1 Objectives

Develop and implement a product demand prediction system with scalability and adaptability across diverse industries and product categories.

2.2 Requirement Engineering

To study the system, you need to collect facts. Facts are expressed in qualitative form called as data. Success of any requirement any investigation depends upon availability of accurate and reliable data. These depend on appropriate method chosen for data collection. The specific methods used for collecting data are fact finding techniques.

The different methods used by analyst are:

Interview

Onside

Observation

Record

Review

Questionary

In this project I am using the method of:

Interview:

Interview technique is used to collect information from individual or from groups. Analyst should select respondent how are related to system under study. In this method interviewer that is analyst seats face to face with respondent and record his responses.

The information collected is likely to be more accurate and reliable because the interviewer can clear up their doubts and crass check the despondence. This method also helpsto find the area of misunderstanding, unrealistic expectations and future problems of the prosesystem.

Observation:

Unlike the other fact-finding technique, in this method the analyst himself visits the organization on observes and understands the flow of document, working of requirement system, the users of the system etc. For this method to be adopted it takes and analyst to perform this job as he knows which points should be noticed and highlighted. In analyst may observe the unwanted things as well and simply cause delay in the development of the new system.

2.3 Requirement Gathering

The waterfall model is a sequential (non-iterative) design process, used in softwaredevelopment process, in which process is seen as flowing steadily downwards (like a waterfall) through the phases of conception, initiation, analysis, design, construction, testing, production/implementation & maintenance. Despite the development of new software development process models, the waterfall model is still the dominant process model with over a third of software developers still using it.

2.4 Software Requirement

The software requirements are description of features and functionalities of the target system. SRS defines how the intended software will interact with hardware, external interfaces,speed of operation, response time of system, portability of software across various platforms, maintainability, speed of recovery after crashing, Security, Quality, Limitations etc. It is the responsibility of system analyst to document the requirements in technical language so that they can be comprehended and useful by the software development team.

SRS should come up with following features:

- User Requirements are expressed in natural language.
- Technical requirements are expressed in structured language, which is sued inside theorganizations.
- Design description should be written in Pseudo code.
- Format of Forms and GUI screen prints.
- Conditional and mathematical notations for DFDs etc.
- Technical requirements are expressed in structured language.
- Format of Forms and GUI screen prints.

Broadly software requirements should be categorized in two categories:

Functional Requirements:

Requirements, which are related to functional aspect of software fall into this category. Theydefine functions and functionality within and from the software system.

Non-Functional Requirements:

Requirements, which are not related to functional aspect of software, fall into this category. They are implicit or expected characteristics of software, which users make assumption of.

Software Requirement:

What is Flask?

Flask is a web application framework written in Python. It was developed by Armin Ronacher,who led a team of international Python enthusiasts called Pocco. Flask is based on the Werkzeg WSGI toolkit and the Jinja2 template engine. Both are Pocco projects.

Flask is a web framework, it's a Python module that lets you develop web applications easily.It has a small and easy-to-extend core: it's a microframework that doesn't include an ORM (Object Relational Manager) or such features. It does have many cool features like URL routing, template engine. It is a WSGI web app framework.

WSGI: The Web Server Gateway Interface (Web Server Gateway Interface, WSGI) has been used as a standard for Python web application development. WSGI is the specification of a common interface between web servers and web applications.

Werkzeug: Werkzeug is a WSGI toolkit that implements requests, response objects, and utilityfunctions. This enables a web frame to be built on it. The Flask framework uses Werkzeg as one of its bases.

Database Requirement:

Introduction to MySQL server:

- MySQL is a relational database management system.
- MySQL is open-source.
- MySQL is free.
- MySQL is ideal for both small and large applications.
- MySQL is very fast, reliable, scalable, and easy to use.
- MySQL is cross-platform.
- MySQL is compliant with the ANSI SQL standard.
- MySQL was first released in 1995.

Features of MYSQL Server:

Open Source

Quick and Reliable

Scalable

Data Types

Character Sets

Secure

Supports Large Databases

Chapter 3
System Analysis

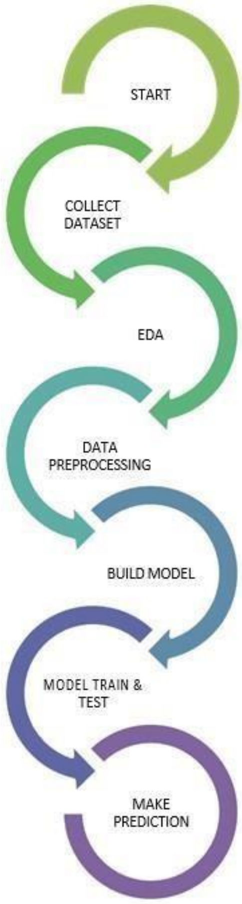
3.1 System Designs

The system design consists of machine learning model in the backend while the frontend is a website wherein users can enter their details in order to get the output.

System Architecture

- **Chatbot Service**: Using this service users will register with a web application and have the option to use a chatbot to get an automatic response from the trained question and answer data. The chatbot will be trained using LSTM and dialog flow.
- **Online Analysis**: The users will receive an analysis of their reports with the help of the website. Once they enter their report details into the interface, they will receive the analysis and the risk percentage.

The system design is as follows –



- **Collecting dataset** – This step involves collecting dataset manually which I referred from various sources. The dataset contains 7k values.

- **EDA** – This phase involves understanding the dataset and using libraries like matplotlib and seaborn to visualize the variables in the dataset.

- **Data preprocessing** – In this phase we removed and cleaned the dataset to remove all the null values. The processing of categorial values is also done in this step

- **Building model** – In this step we build a model using several algorithms. The algorithms that we have used in this project are Logistic Regression, SVM, KNN, Random Forest, Naïve Bayes, Neural Network, MLP, Perceptron and Decision Tree Classifier.

- **Model train and test** – In this step we train and test the dataset by firstly splitting it in a specific ratio. We have split our dataset using sklearn library in the ratio of 1:4, i.e., 80% of data for training and remaining 20% for testing.

- **Making prediction** – After making sure that the model works properly for while testing it, we can deploy the model for making predictions. For this, we are going to create a website containing fields for taking information from the user. From the values entered by the user the model will make further predictions.

3.2 Methodology/Algorithm

3.2.1 Logistic Regression

Independent variables are analyzed to determine the binary outcome with the results falling into one of two categories. The independent variables can be categorical or numeric, but the dependent variable is always categorical.

In logistic regression, we fit a "S" shaped logistic function, which predicts two maximum values, rather than a regression line (0 or 1).

The logistic function's curve shows the possibility of several things, including whether or not the cells are malignant, whether or not a mouse is obese depending on its weight, etc.

Because it can classify new data using both continuous and discrete datasets, logistic regression is a key machine learning approach.

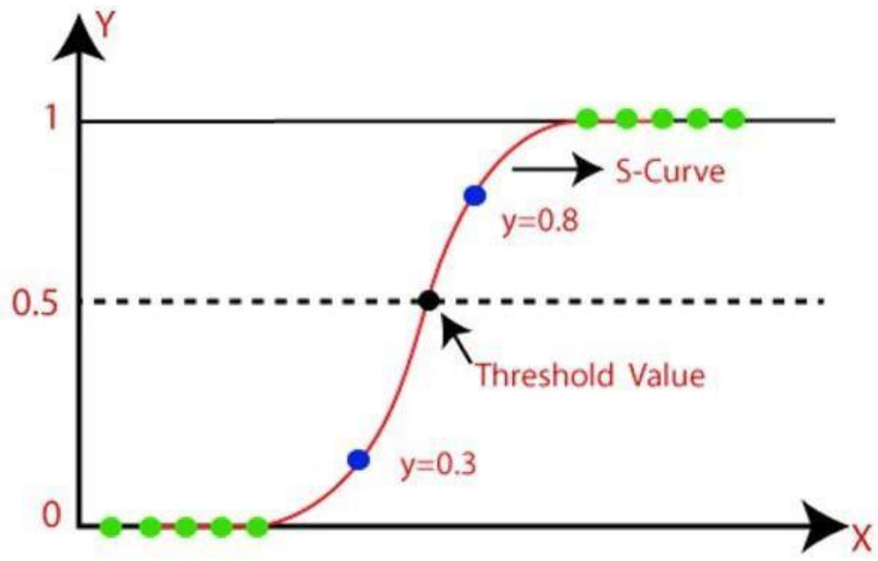


Fig. Logistic Regression

3.2.2 K-Nearest Neighbors

K-nearest neighbors (k-NN) is a pattern recognition algorithm that uses training datasets to find the k closest relatives in future examples. When k-NN is used in classification, you calculate to place data within the category of its nearest neighbor. If k =1, then it would be placed in the class nearest 1. K is classified by a plurality poll of its neighbors.

The K-NN algorithm assumes that the new case and the existing cases are comparable, and it places the new instance in the category that is most like the existing categories.

A new data point is classified using the K-NN algorithm based on similarity after all the existing data has been stored. This means that utilizing the K-NN method, fresh data can be quickly and accurately sorted into a suitable category.

It is also known as a lazy learner algorithm since it saves the training dataset rather than learning from it immediately. Instead, it uses the dataset to perform an action when classifying data.

The KNN method simply saves the information during the training phase, and when it receives new data, it categorizes it into a category that is quite like the new data.

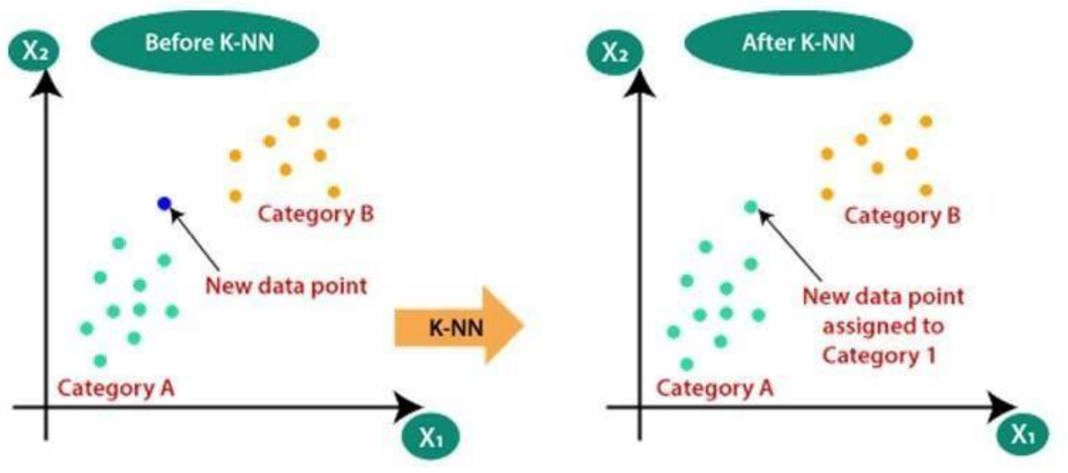


Fig. K-Nearest Neighbors

3.2.3 Random Forest

The random forest algorithm is an expansion of decision tree, in that you first construct a multitude of decision trees with training data, then fit your new data within one of the trees as a “random forest”. It, essentially, averages your data to connect it to the nearest tree on the data scale.

Popular machine learning algorithm Random Forest is a part of the supervised learning methodology. It can be applied to ML issues involving both classification and regression. It is built on the idea of ensemble learning, which is a method of integrating various classifiers to address difficult issues and enhance model performance.

Random Forest, as the name implies, is a classifier that uses a number of decision trees on different subsets of the provided dataset and averages them to increase the dataset's predictive accuracy. Instead, then depending on a single decision tree, the random forest uses forecasts from each tree and predicts the result based on the votes of the majority of predictions.

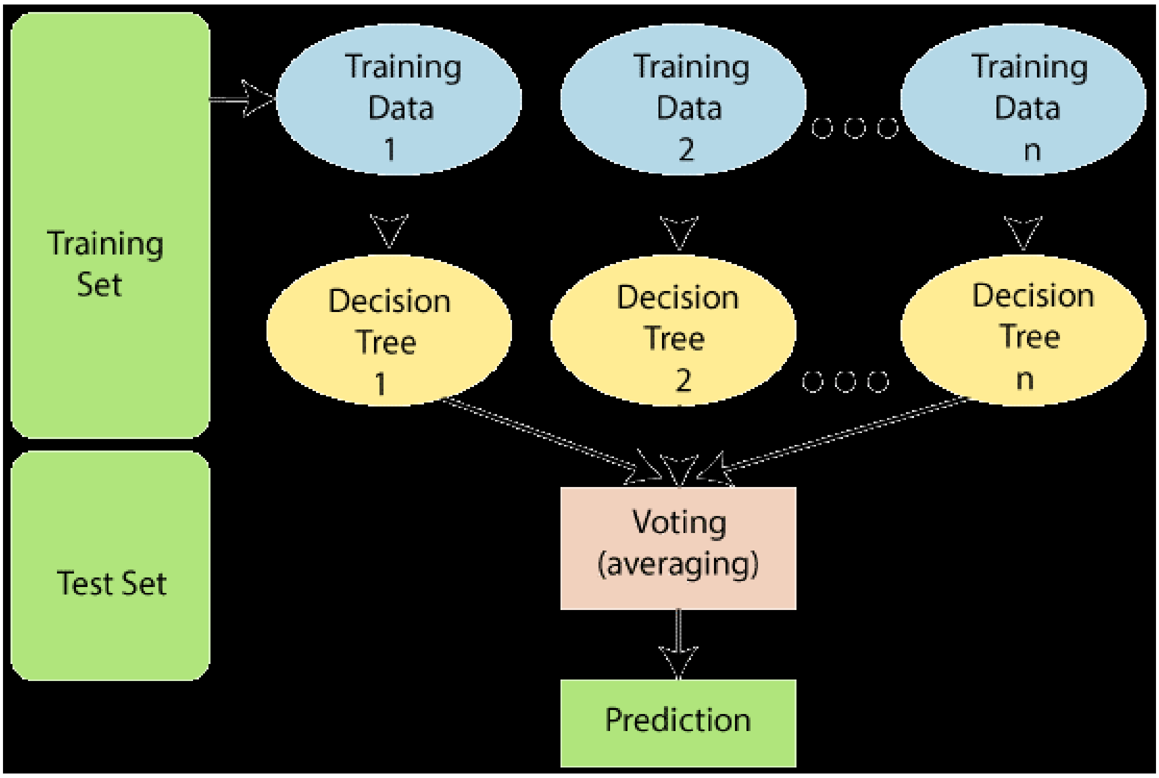


Fig. Random Forest

For the dataset’s feature variable to predict true outcomes rather than a speculated result, there should be some actual values in the dataset.

3.2.4 Decision Tree

A decision tree is a supervised learning algorithm that is perfect for classification problems, as it's able to order classes on a precise level. It works like a flow chart, separating data points into two similar categories at a time from the "tree trunk" to "branches," to "leaves," where the categories become more finitely similar. This creates categories within categories, allowing for organic classification with limited human supervision.

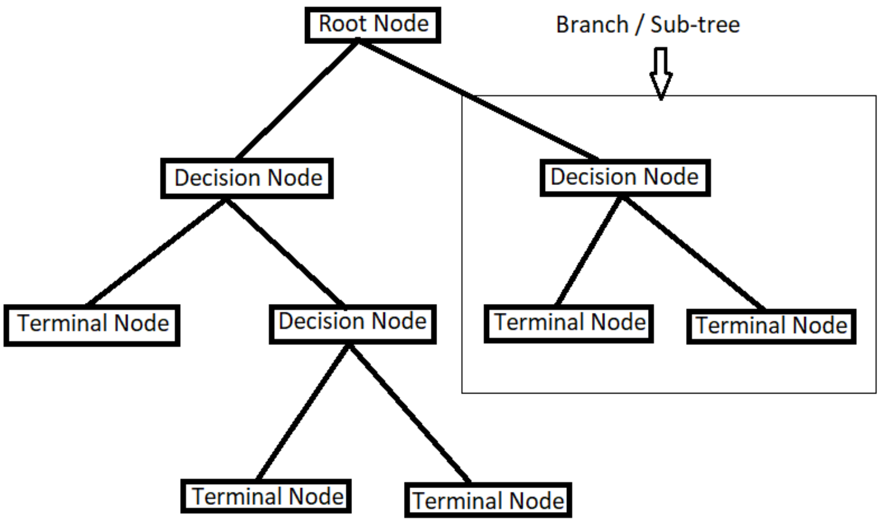


Fig. Decision Tree

3.2.5 Support Vector Machine

A support vector machine (SVM) uses algorithms to train and classify data within degrees of polarity, taking it to a degree beyond X/Y prediction. The SVM algorithm's objective is to establish the best line or decision boundary that can divide n-dimensional space into classes, allowing us to quickly classify fresh data points in the future. A hyperplane is the name given to this optimal decision boundary. SVM selects the extreme vectors and points that aid in the creation of the hyperplane. Support vectors, which are used to represent these extreme instances, form the basis for the SVM method.

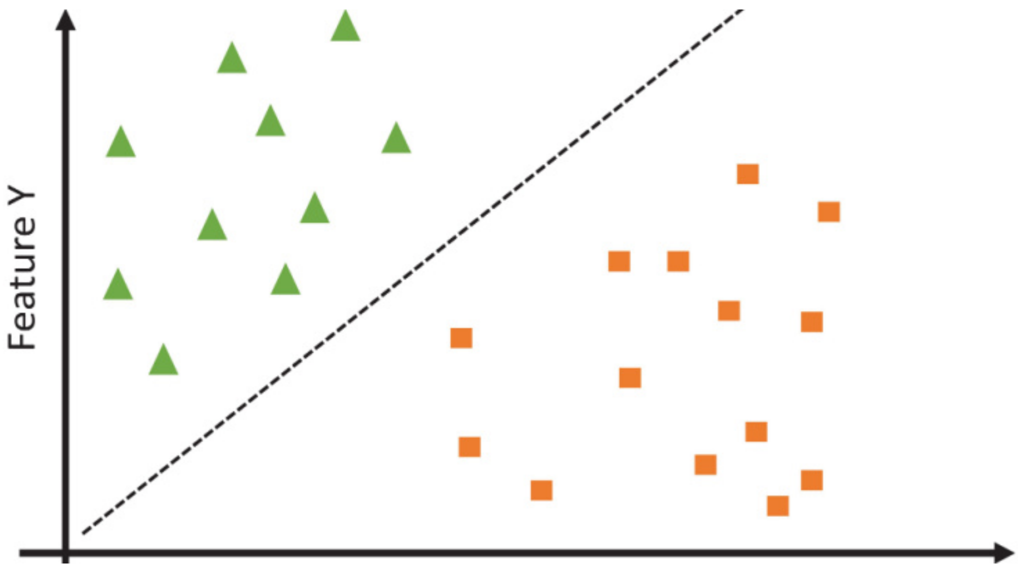


Fig. Support Vector Machine

3.2.6 Naïve Bayes

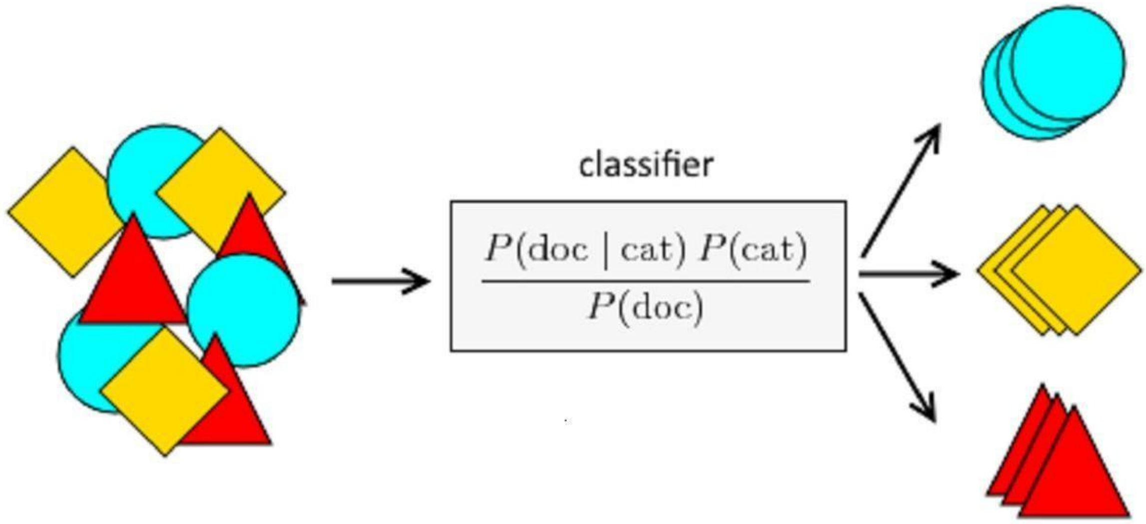


Fig. Naïve Bayes

This algorithm is a supervised learning method for classification issues that is based on the Bayes theorem. It is mostly employed in text categorization with a large training set. The Naive Bayes Classifier is one of the most straightforward and efficient classification algorithms available today.It aids in the development of quick machine learning models capable of making accurate predictions. Being a probabilistic classifier, it makes predictions based on the likelihood that an object will occur. The Bayes theorem, also referred to as Bayes' Rule or Bayes' law, is used to calculate the likelihood of a hypothesis given some prior information. The conditional probability determines this.

$$P(A|B) = \frac{P(B|A)P(A)}{P(B)}$$

P(A|B) is Posterior probability: Probability of hypothesis A on the observed event B.

P(B|A) is Likelihood probability: Probability of the evidence given that the probability of a hypothesis is true.

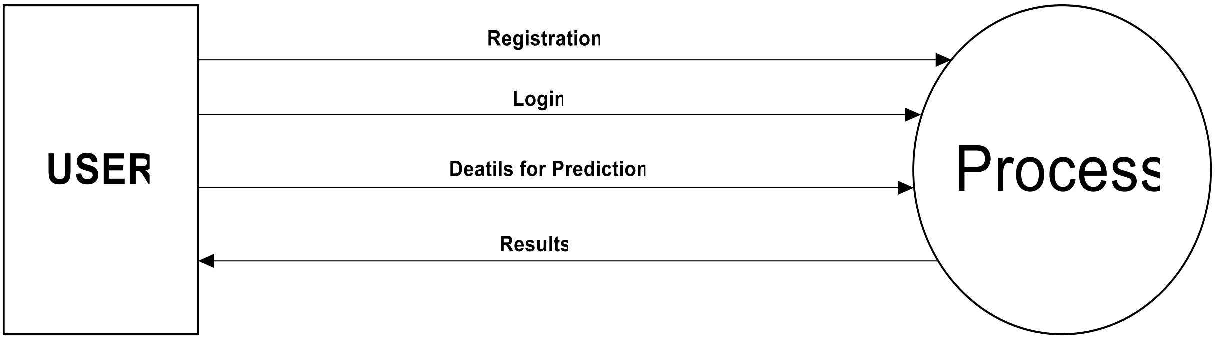
P(A) is Prior Probability: Probability of hypothesis before observing the evidence.

P(B) is Marginal Probability: Probability of Evidence.

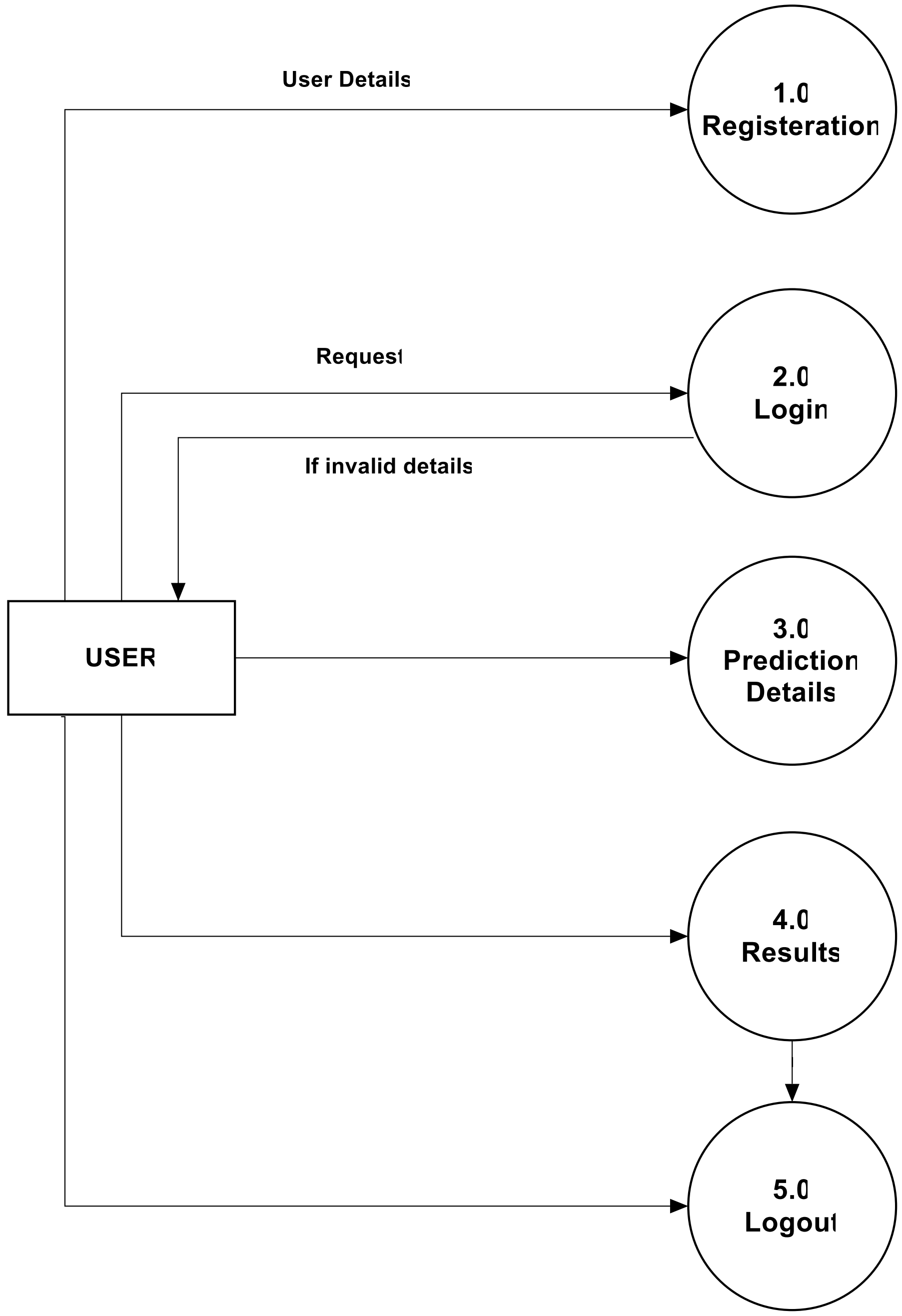
We are going to implement the above given algorithms and choose one that gives the highest accuracy for making predictions.

3.3 Data Flow Diagram

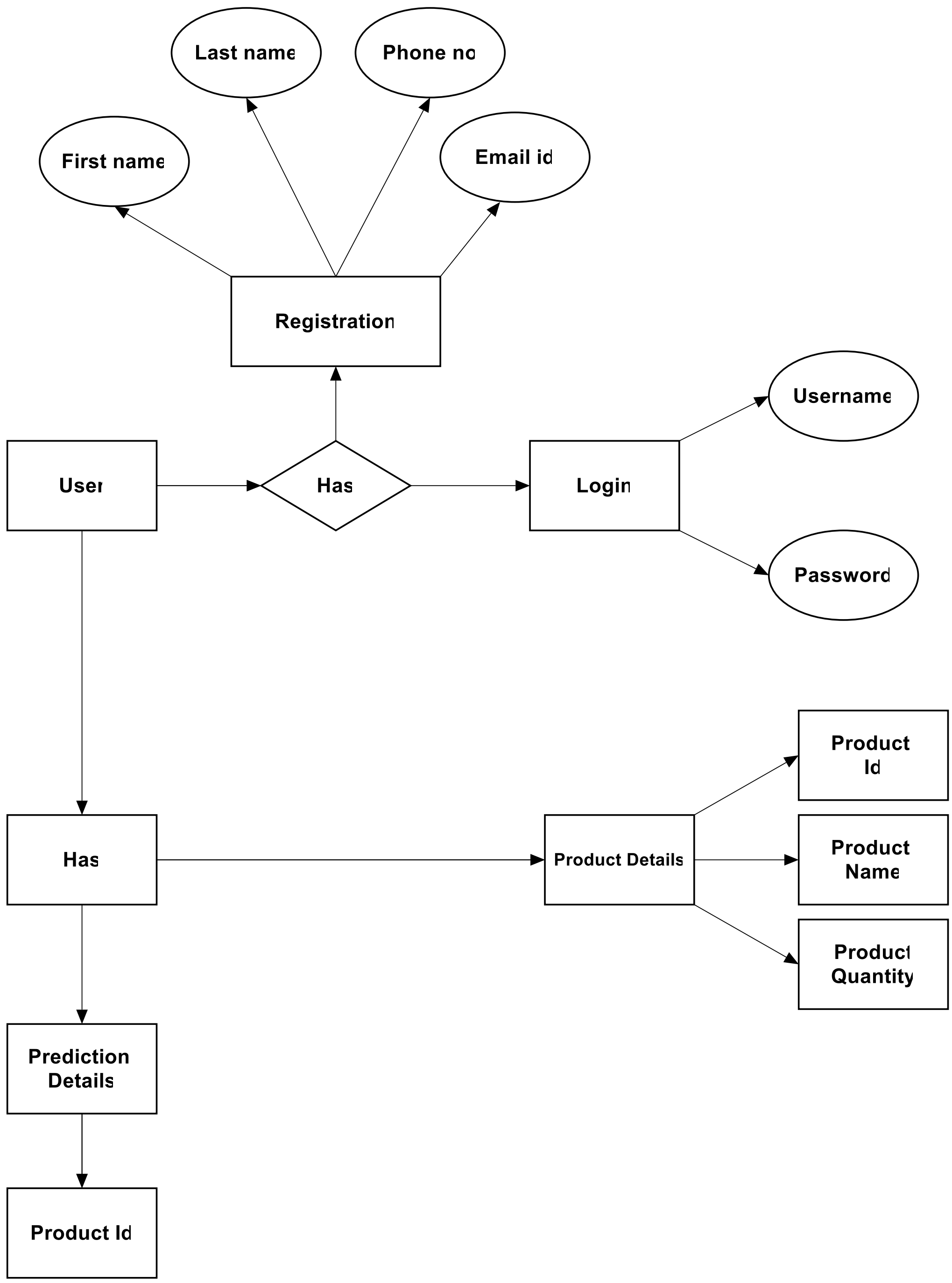
3.3.1 Context Level DFD



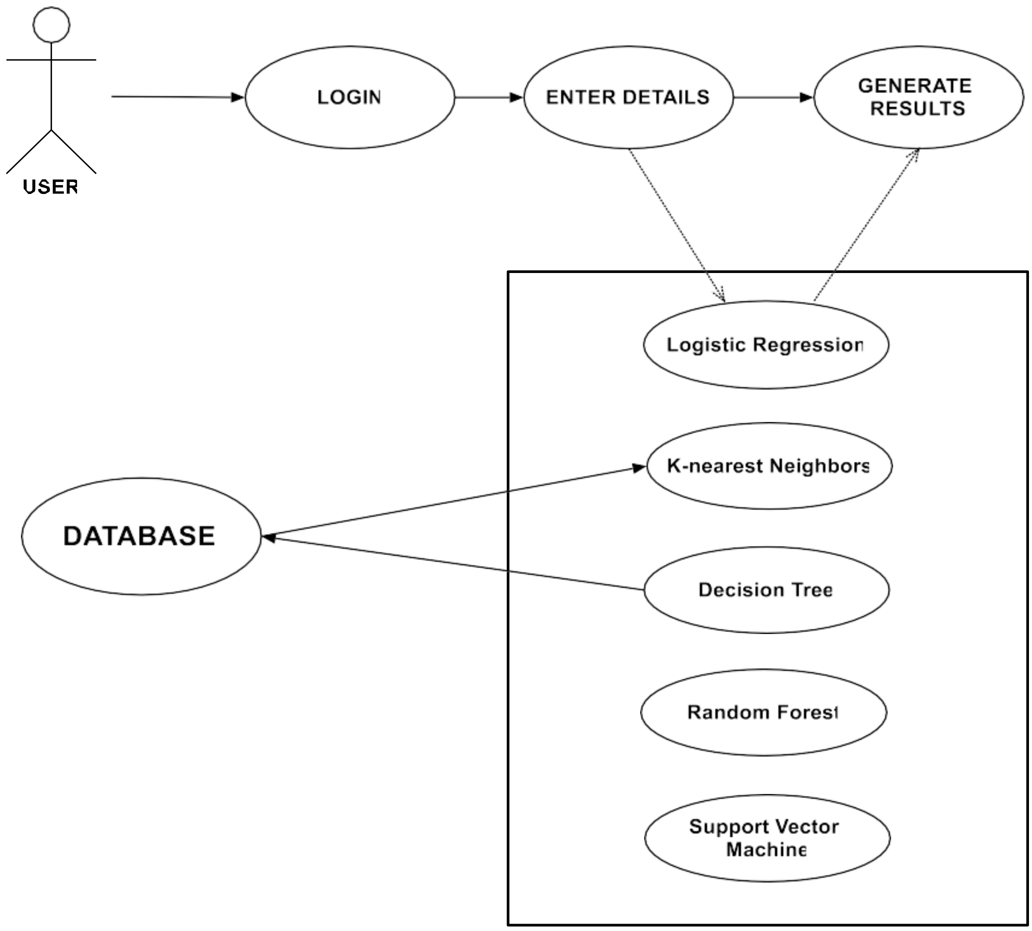
3.3.2 First Level DFD



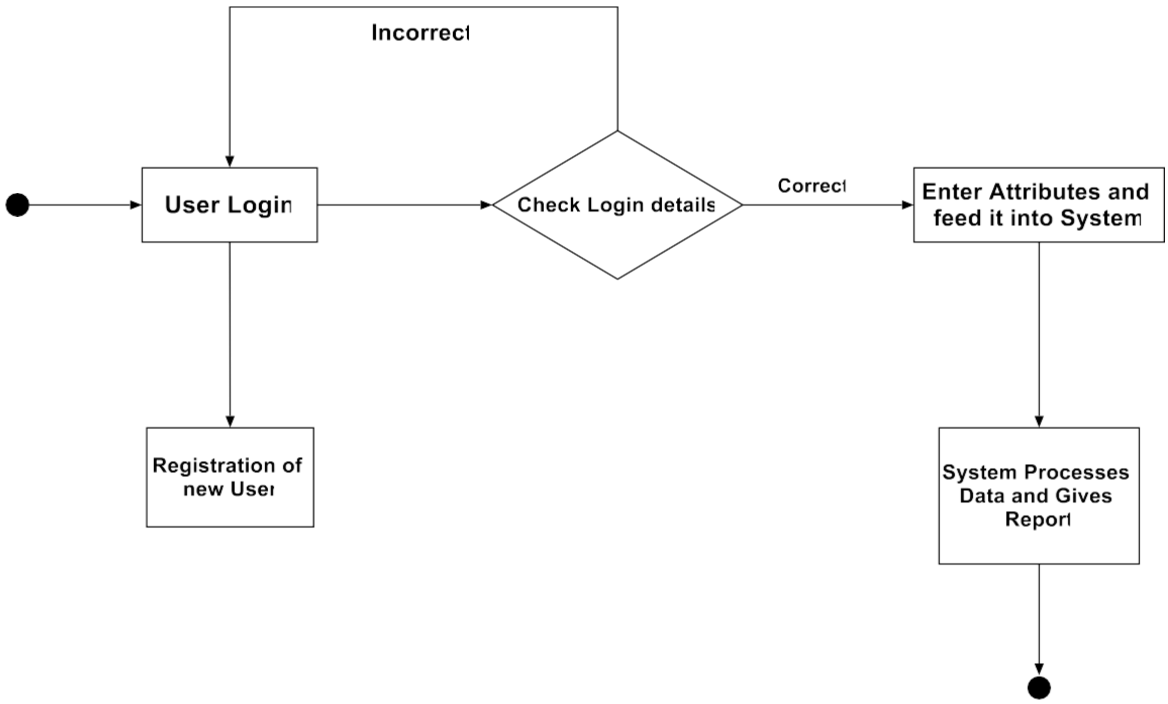
3.4 Entity Relationship Diagram (ERD)



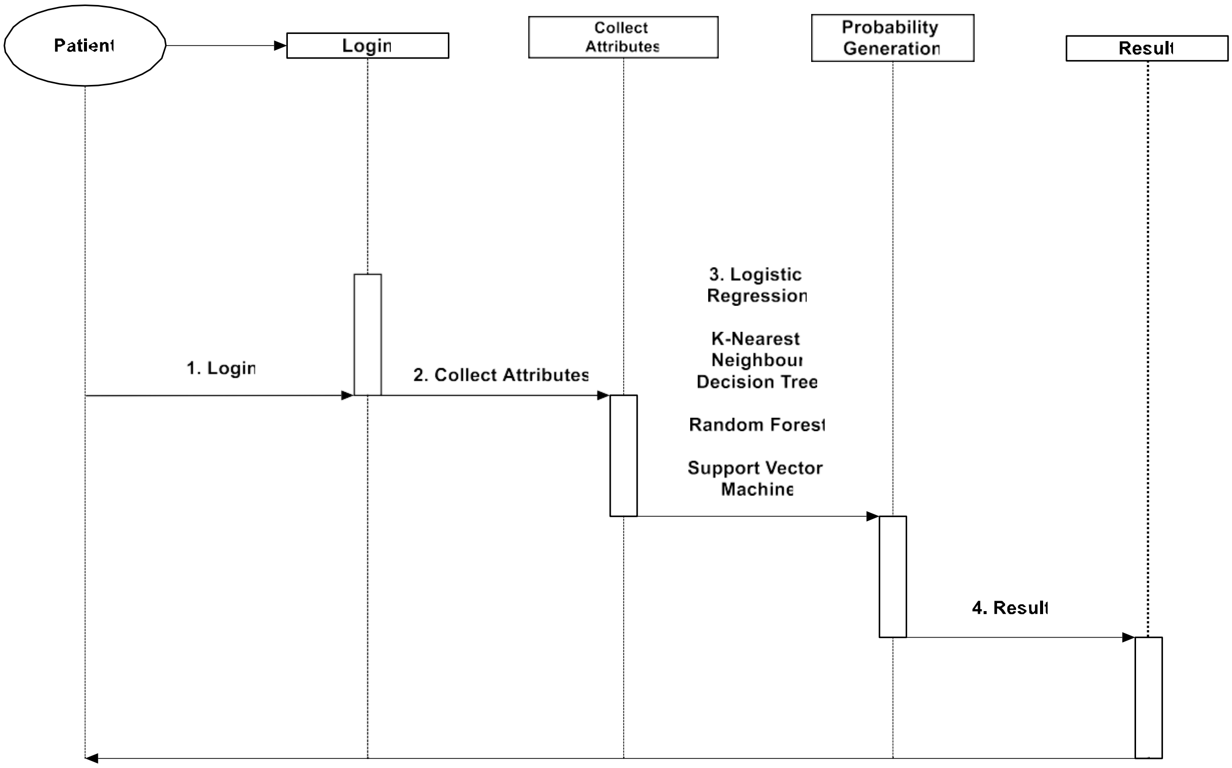
3.5 Use Case Diagram



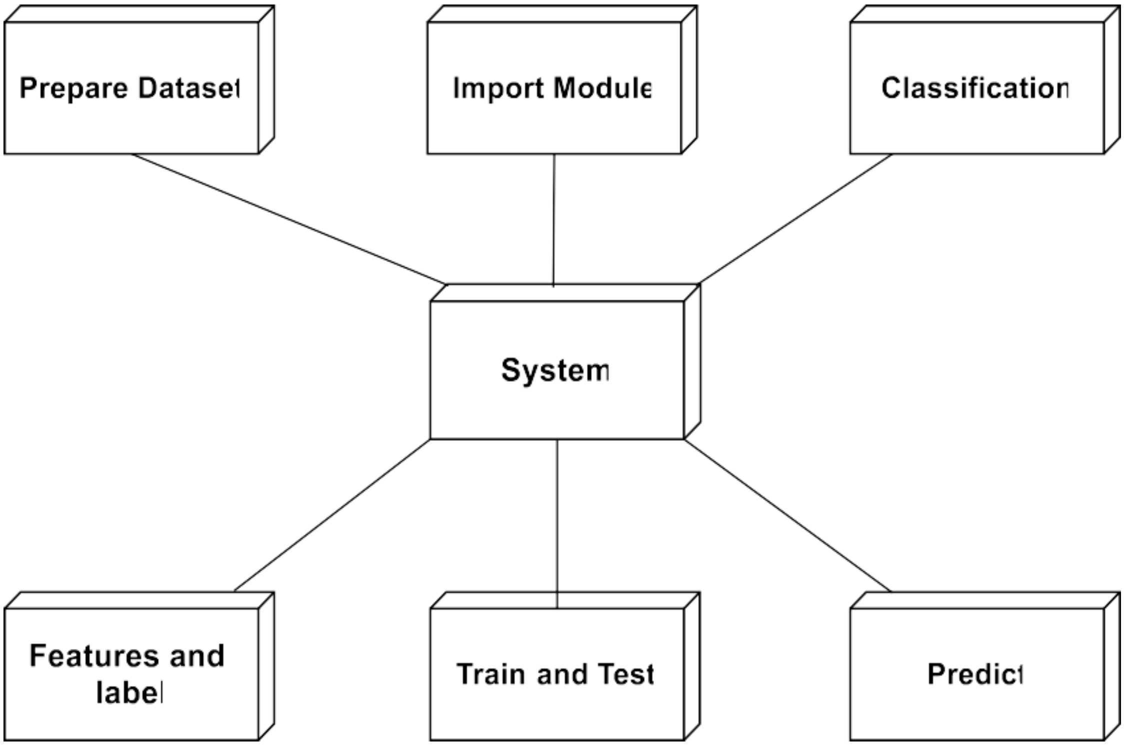
3.6 State Diagram



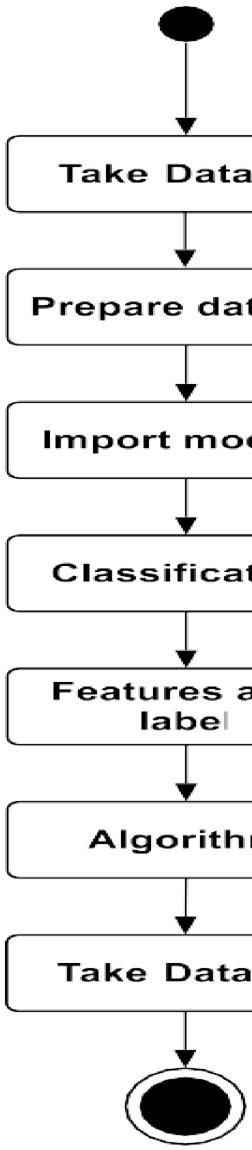
3.7 Sequence Diagram



3.8 Deployment Diagram



3.9 Activity Diagram



Chapter 4
System Design

4.1 Database Design

1. User Registration

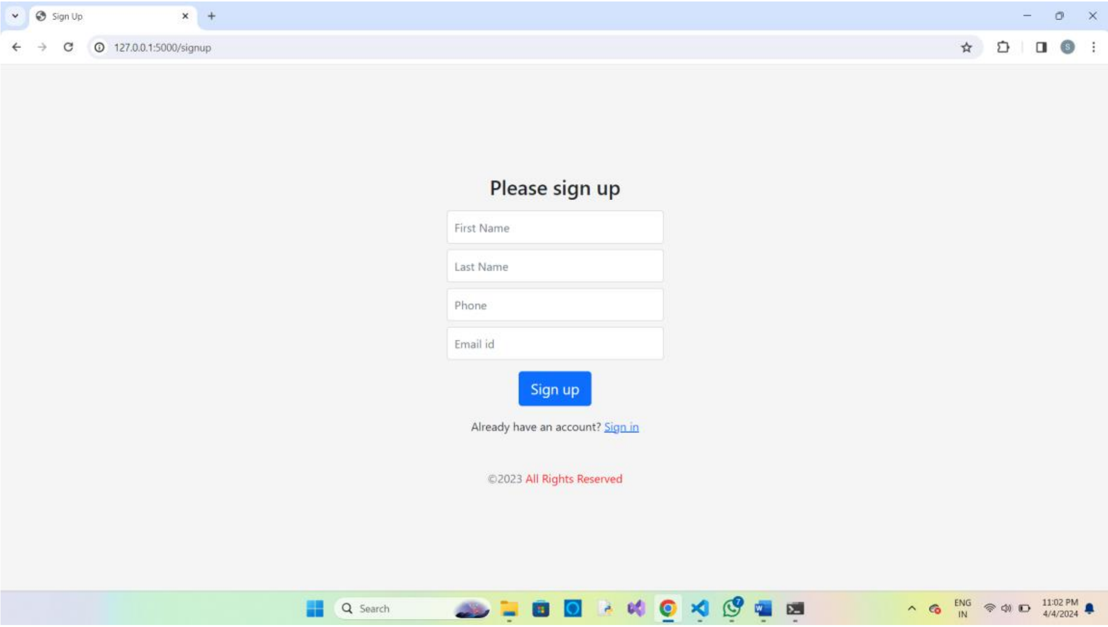
Field	Type	Null	Key	Default	Extra
Rid	Int	No	Primary Key	Null	Auto_increment
Fname	Varchar(255)	Yes		Null	
Lname	Varchar(255)	Yes		Null	
Phone	Bigint	Yes		Null	
Emailid	Varchar(255)	Yes		Null	
Password	Varchar(255)	Yes		Null	

2. Product Details

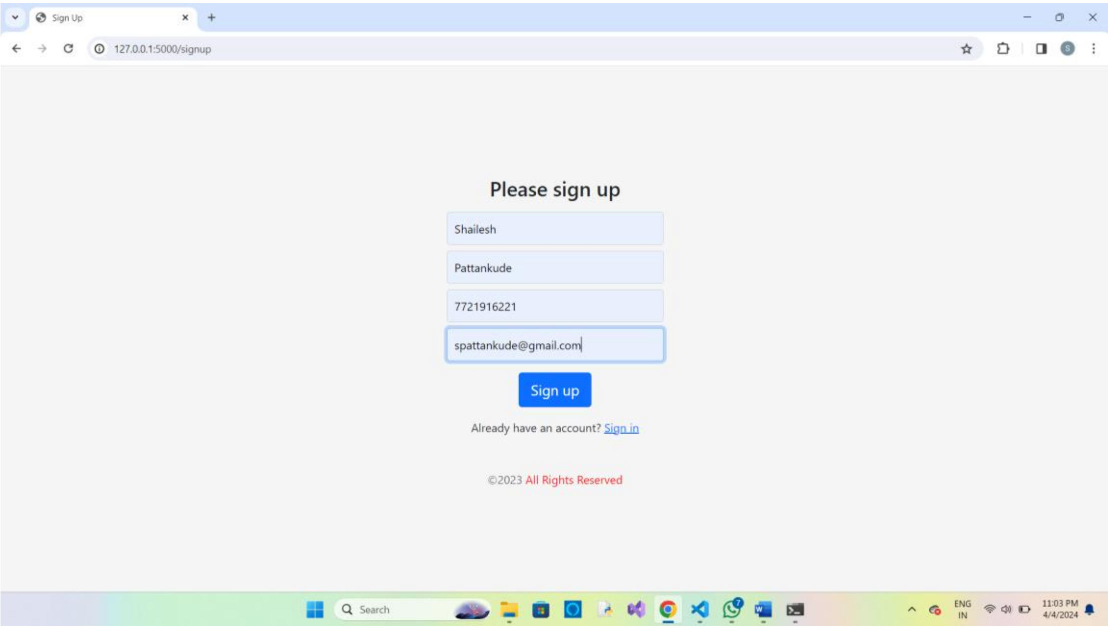
Field	Type	Null	Key	Default	Extra
Productid	Int	No	Primary Key	Null	Auto_increment
Image	Varchar(255)	Yes		Null	
Productname	Int	Yes		Null	
Quantity	Int	Yes		Null	

4.2 Input Design

1. Registration:

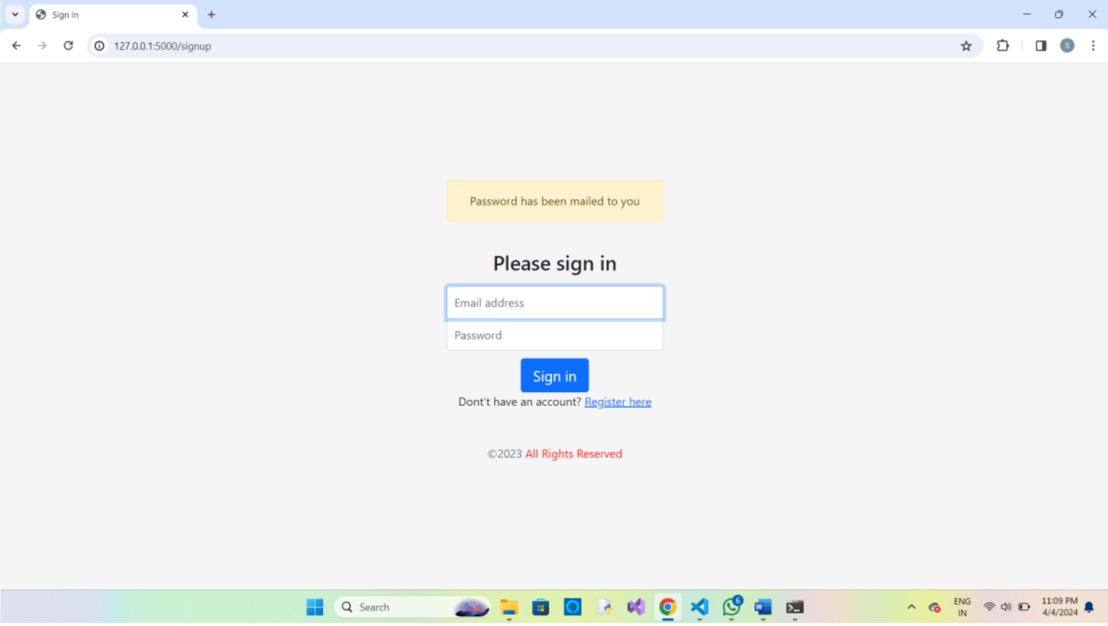


Providing the Inputs:

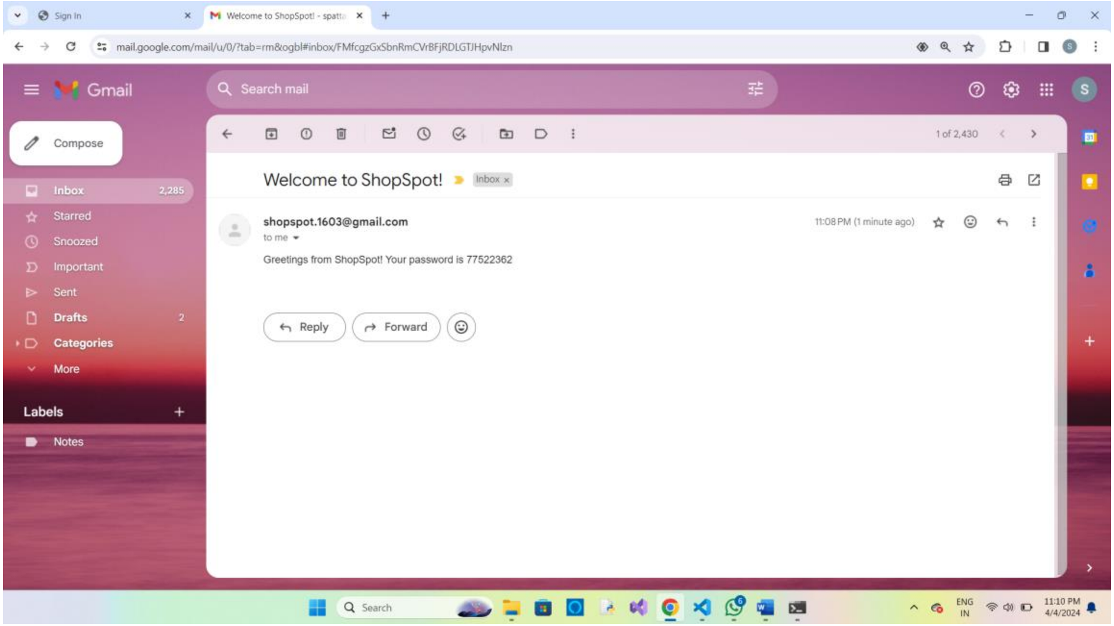


After submitting the input:

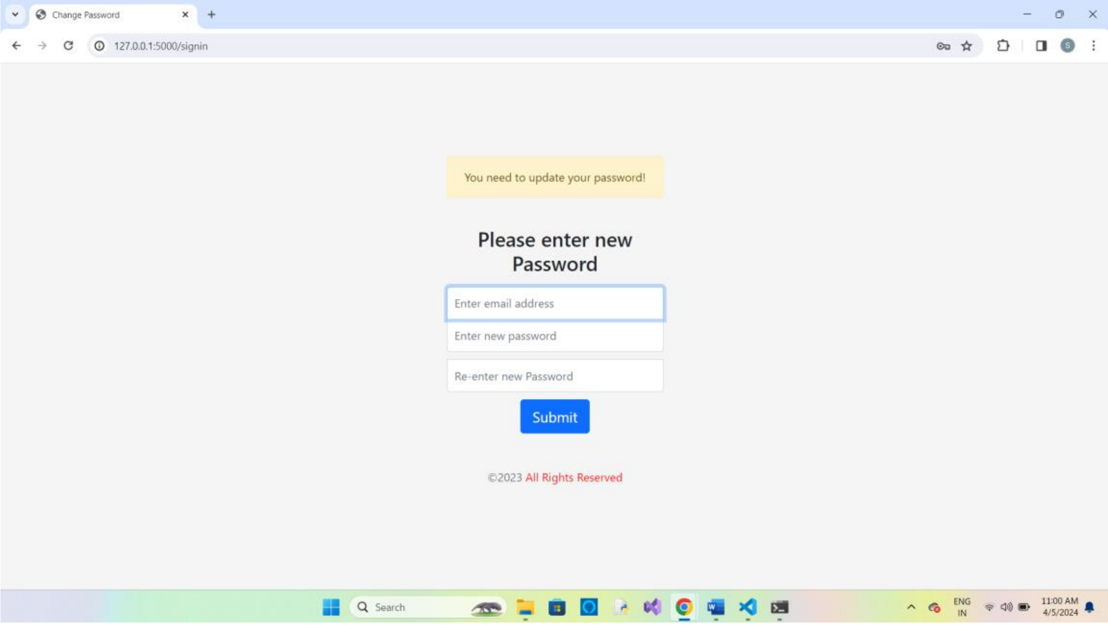
(Redirected to Sign In page as password was mailed on given email id)



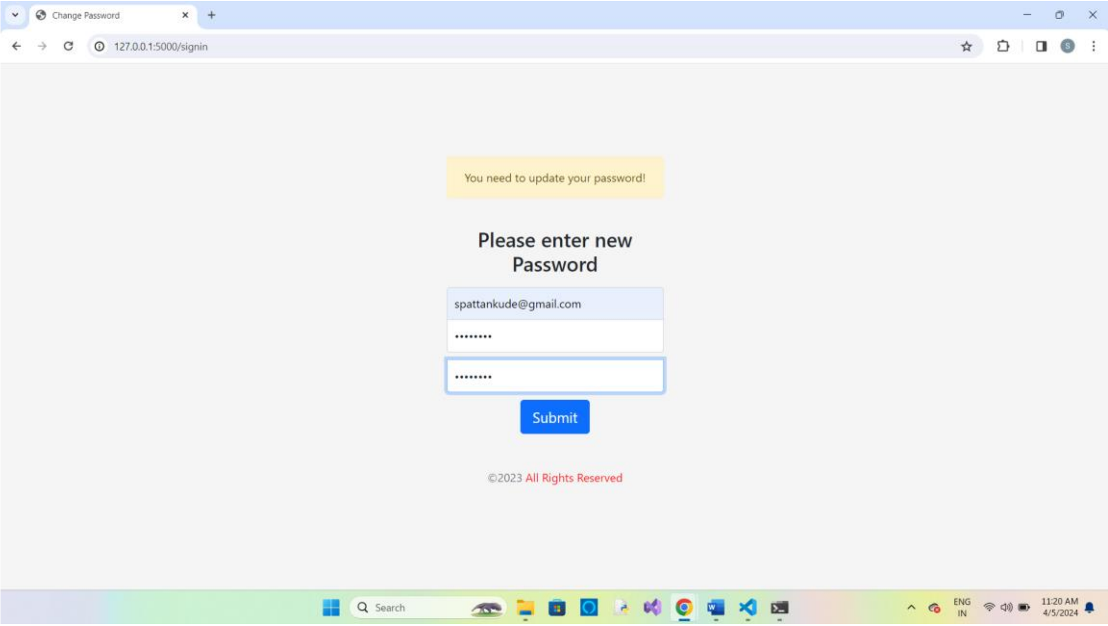
Password was received through email:



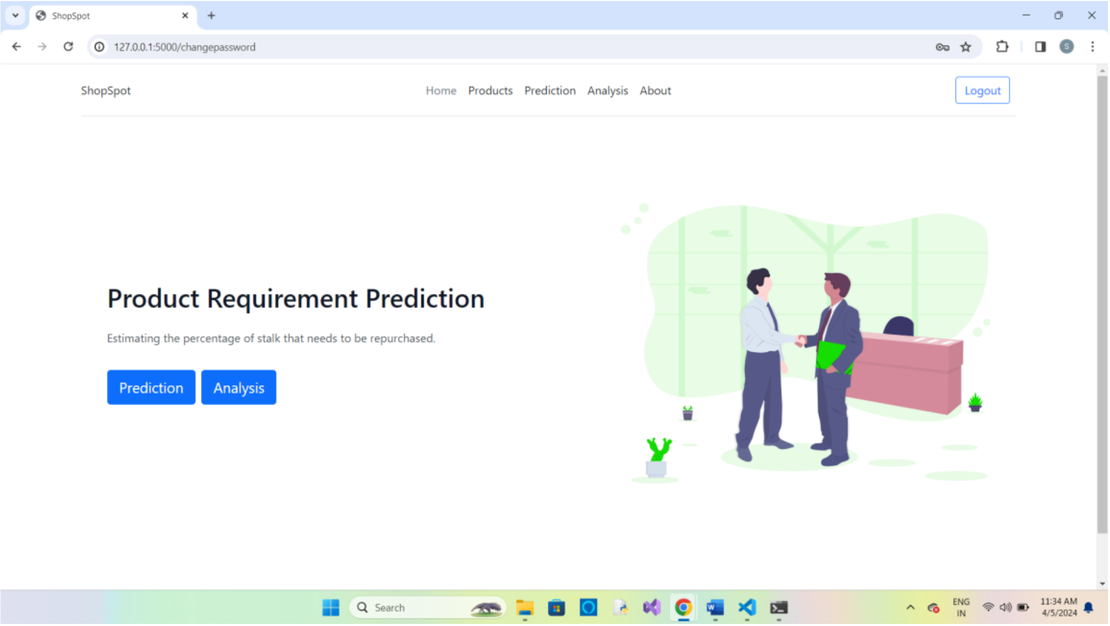
The auto-generated password is numerical and after logging in, application asks to you change your password:



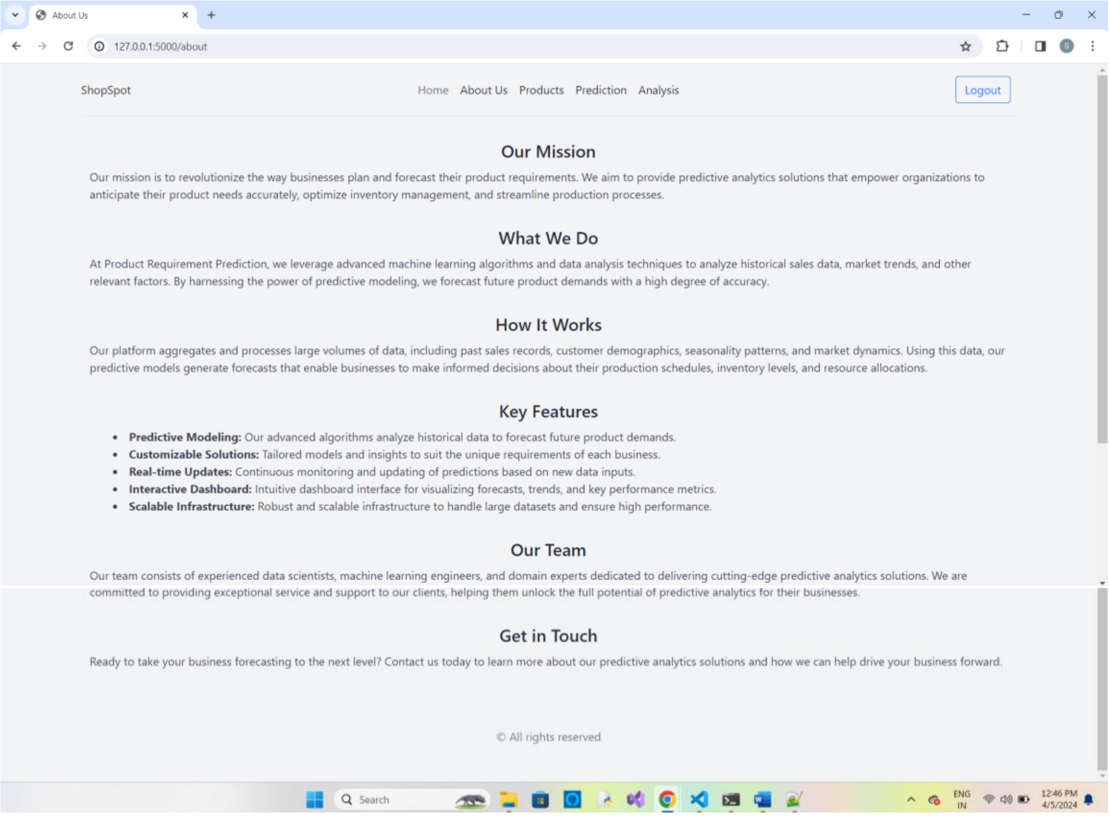
Providing the details:



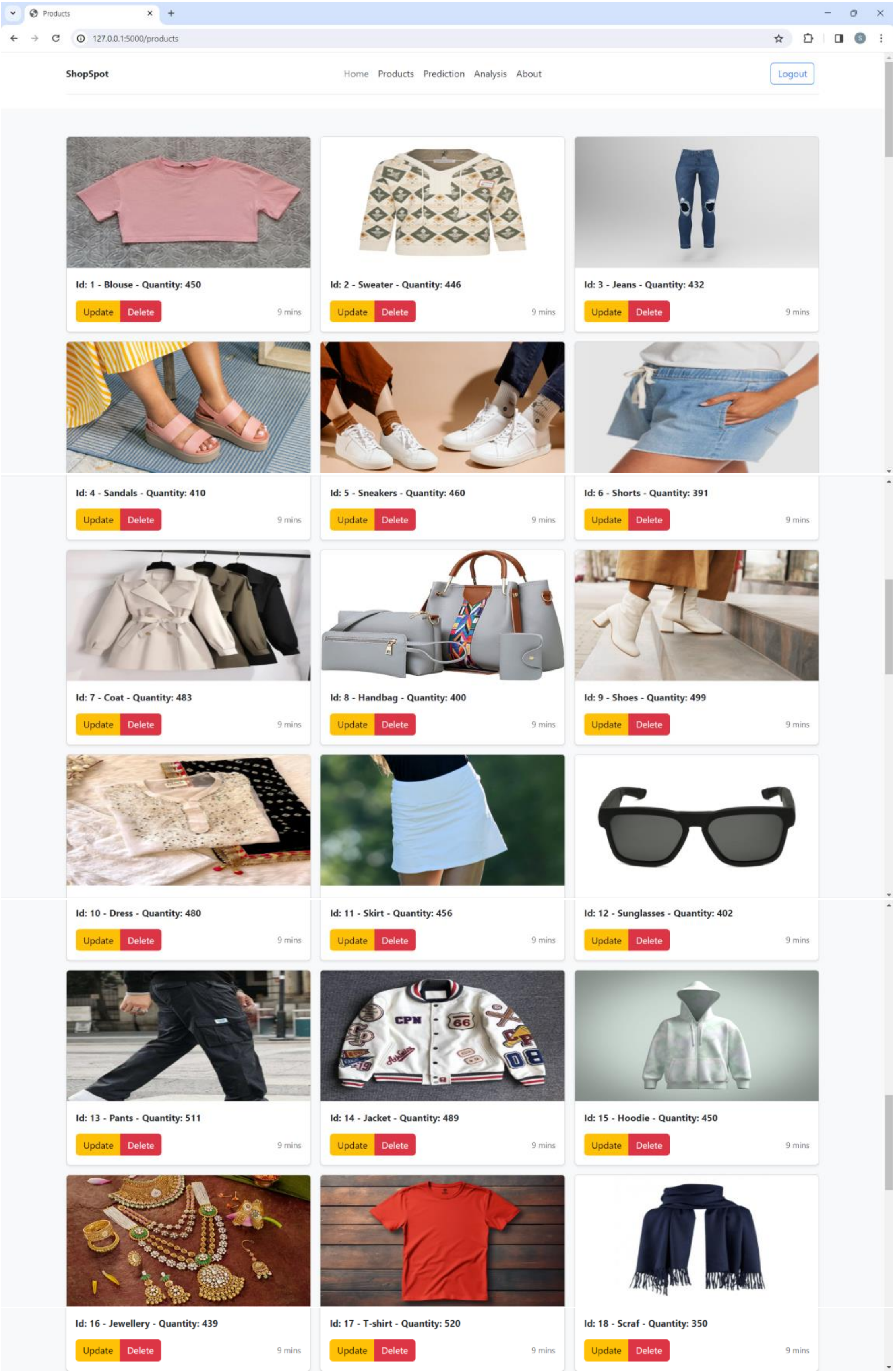
2. Index page:

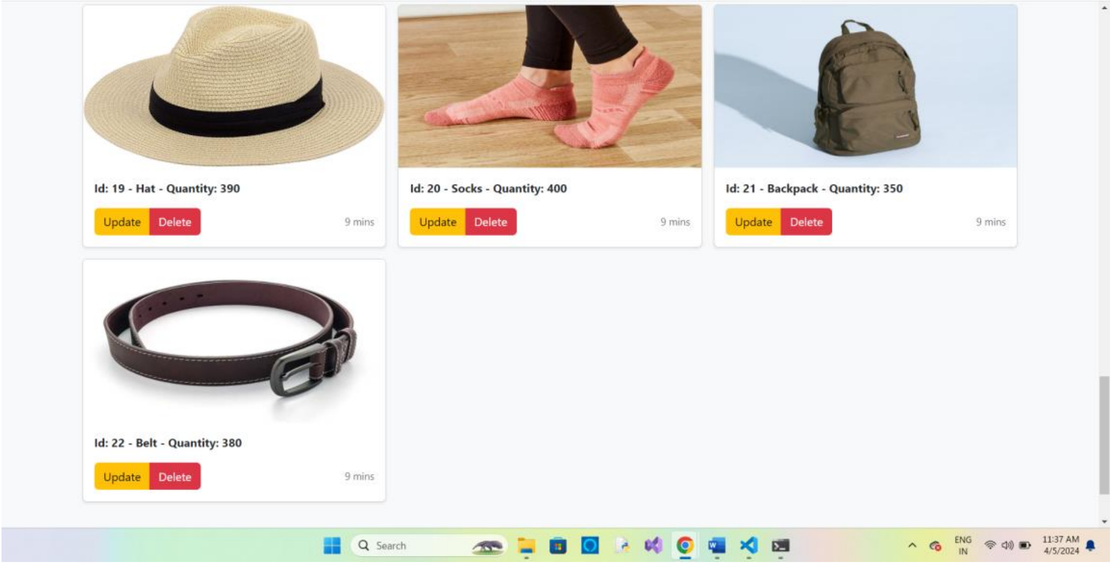


3. About us:

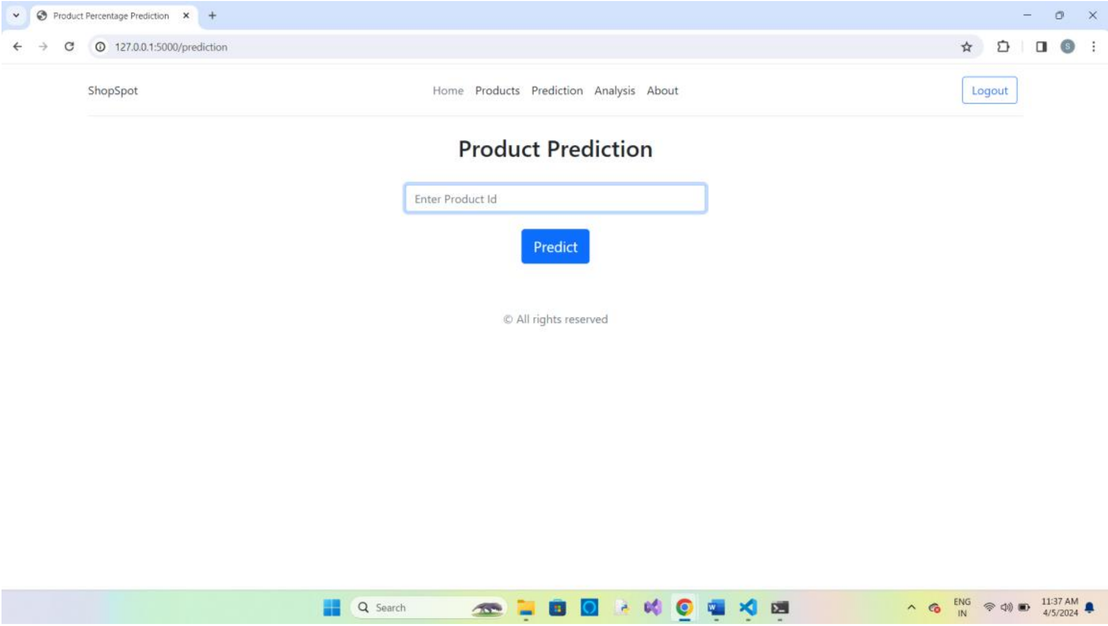


4. Products:

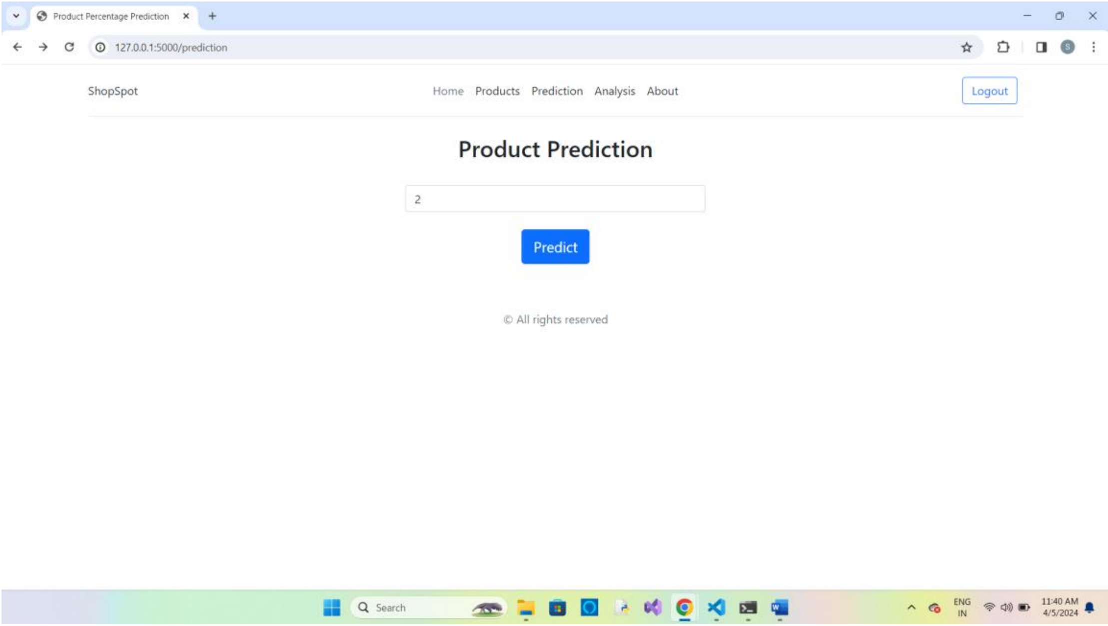

Products
127.0.0.1:5000/products
ShopSpot
Home Products Prediction Analysis About
Logout
Id: 1 - Blouse - Quantity: 450
Id: 2 - Sweater - Quantity: 446
Id: 3 - Jeans - Quantity: 432
Id: 4 - Sandals - Quantity: 410
Id: 5 - Sneakers - Quantity: 460
Id: 6 - Shorts - Quantity: 391
Id: 7 - Coat - Quantity: 483
Id: 8 - Handbag - Quantity: 400
Id: 9 - Shoes - Quantity: 499
Id: 10 - Dress - Quantity: 480
Id: 11 - Skirt - Quantity: 456
Id: 12 - Sunglasses - Quantity: 402
Id: 13 - Pants - Quantity: 511
Id: 14 - Jacket - Quantity: 489
Id: 15 - Hoodie - Quantity: 450
Id: 16 - Jewellery - Quantity: 439
Id: 17 - T-shirt - Quantity: 520
Id: 18 - Scraf - Quantity: 350
Update Delete
9 mins



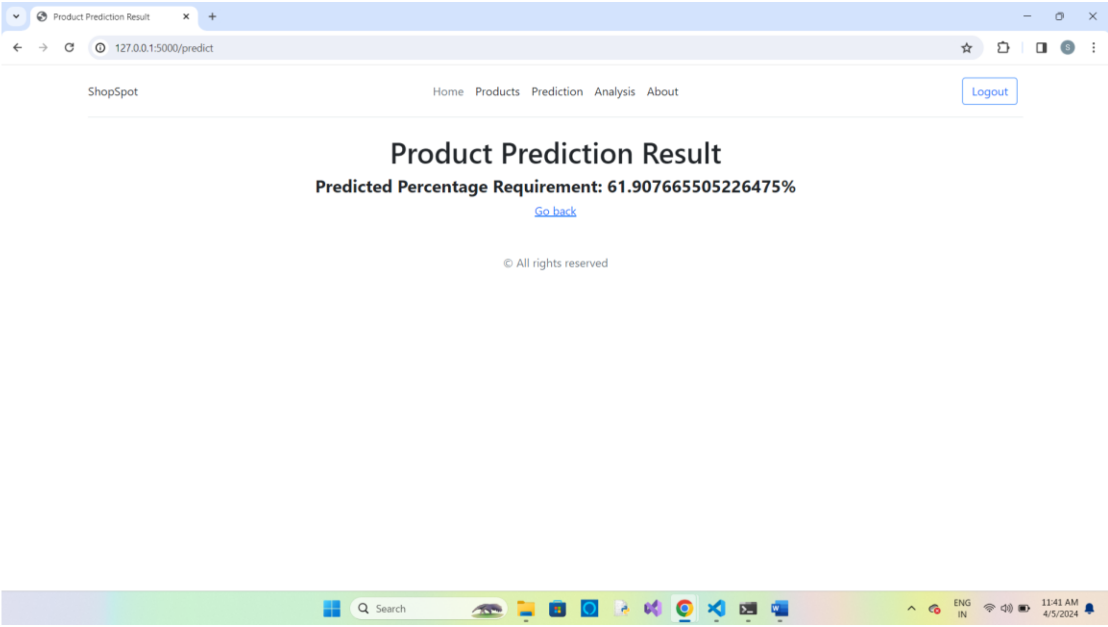
5. Product Requirement Prediction form:



After providing product id:



Result of prediction of product id:



Chapter 5
System Requirements and Implementation

5.1 System Requirement

Hardware:

- Graphics Card: - 1650 or 1660TI

  Performs ML or DL training and (often) inference, which is the capacity to automatically classify data based on learning, and is frequently an Nvidia P100 (Pascal), V100 (Volta), or A100 (Ampere) GPU for training and a V100, A100, or T4 (Turing) GPU for inference.

- CPU: - i5 9th or 10th Gen.

  It is accountable for managing I/O, running the VM or container subsystem, and sending code to the GPUs. With the addition of features that considerably speed up ML and DL inference procedures, current-generation CPUs are now ready for production AI workloads using models that were previously trained on GPUs.

- Storage IOPS

  Another performance barrier for AI workloads is the transfer of data between the storage and compute subsystems. As a result, local NVMe drives are more common than SATA SSDs in systems.

- RAM: - 16GB
- Intel's ML Hardware Evolution
- PC

Software:

- Windows 10 and 11 (Intel/AMD 64-bit)
- Google Collaboratory
- Visual Studio Code
- Microsoft Excel
- Python IDLE

5.2 Implementation

Phase I

1. Dataset Study:

In order to train and test the model we have used a dataset comprising of almost 3900 attributes. In the final project, the user will have to provide data as per their requirement.

The dataset comprises of the following columns –

- Objective: factual information.
- Subjective: information given by the user.

Dataset:

	Customer ID	Customer Name	Product	Product Name	Quantity	invoice_date	Price	Target
0	1	Pattie Morrisey	1	Blouse	55	5/8/2022	2000	4
1	2	Traci Peery	2	Sweater	19	12/12/2021	2000	1
2	3	Merideth Mcmeen	3	Jeans	50	9/11/2021	3500	3
3	4	Eufemia Cardwell	4	Sandals	21	16-05-2021	7000	2
4	5	Meghan Kosak	1	Blouse	45	24-10-2021	5000	3
...	...	...	...	...	...	...	...	...
3895	3896	Herma Voisine	15	Hoodie	40	14-09-2022	2000	2
3896	3897	Evelina Chartrand	21	Backpack	52	10/3/2021	5000	4
3897	3898	Sherie Shaughnessy	22	Belt	46	6/8/2022	4500	3
3898	3899	Janell Neubauer	9	Shoes	44	6/8/2021	9000	3
3899	3900	Antonietta Tann	8	Handbag	52	4/3/2022	3000	4

3900 rows × 8 columns

Attributes:

Customer Id	Objective	Customer Id	Int
Customer Name	Objective	Customer Name	Varchar
Product Id	Objective	Product	Int
Product Name	Objective	Product Name	Varchar
Quantity	Objective	Quantity	Int
Invoice_date	Objective	Invoice_date	Date
Price	Objective	Price	Int
Target	Target Varibale	Target	Int

After taking input from the user, we are going to pass it to the model that we have trained in order to receive accurate results regarding require of product.

2. Performing EDA (exploratory data analysis)

The dataset requires cleaning and modifying to facilitate easy access of data. For that purpose, we will perform EDA on null values and categorial values in the dataset.

3. Feature selection

Two of the 8 features in the data set—one each for quantity and customer id—are used to identify the purchase history. The 6 remaining qualities are significant because they include more details. We chose quantity and product id to be our target value based on whom we will carry out our analysis. We are also going to consider invoice date and other parameters for more accurate results.

4. Splitting data into train and test

The project considered two main ways of data splitting one being using sklearn library and another approach is using cross validation or k-fold cross validation. The Sklearn train_test_split function helps us create our training data and test data. Whereas on the other hand, Cross-Validation or K-Fold Cross-Validation is a more robust technique for data splitting, where a model is trained and evaluated "K" times on different samples.

The value of k may be set as per the programmer's choice.

The project mainly implements K-fold cross validation, the example for understanding which is as follows –

Suppose we have a balanced, 2-class dataset consisting of 1000 images of raccoons and ringtails (to be used for training and validation only). Now, we want to perform a 5-Fold cross-validation. We first split the datasets into 5 equal and non-overlapping parts: each consisting of 200 images; label them as Parts 1, 2, 3, 4, and 5. Each of these subsets of 200 images consists of mutually different samples.

Now, we will create 5 complete datasets (labeled as Datasets 1-5) using Parts 1-5 in the following manner: For Dataset-1, use Part-1 as the validation set, and consolidate Parts 2-5 to create the training set; for Dataset-2, use Part-2 as the validation set, and consolidate Parts 1, 3, 4 and 5 to create the training set, and so on. Notice that since each part consists of 20% of the data of the original dataset, each of Datasets 1-5 has an 80%-20% train-validation split ratio. Generalizing, each K-Fold cross-validation dataset has (100/K) % data in its validation set (here, 100/5 = 20% was in validation set).

The images of the trained and tested dataset are as below –

Train –

```
[12] Y = df.Target
     X = df.drop('Target', axis=1)
```

```
[14] train, test, target, target_test = train_test_split(X, Y, test_size=0.2,stratify=Y,random_state=2)
```

```
[15] print("train")
     print(train.head())

train
      Customer ID      Customer Name  Product Product Name  Quantity  \
2932         2933  Terrance Constable       21     Backpack        65
2861         2862      Herbert Dewolf        7         Coat        66
1792         1793           Guy Resto        5     Sneakers        38
2801         2802        Dean Gasaway        6        Shirt        64
2387         2388         Cinda Robbs       19          Hat        36

     invoice_date  Price
2932     5/8/2021   3500
2861   26-02-2022   2000
1792   20-09-2021   3500
2801   24-09-2021   3500
2387   15-09-2021   7000
```

Test –

```
print("test")
print(test.head())
print(test.shape)

test
      Customer ID    Customer Name  Product Product Name  Quantity  \
3525         3526        Ned Balog       12   Sunglasses        45
1739         1740  Fabiola Hodson        16      Jewelry        21
3187         3188       Erich Rayo        1       Blouse        67
2035         2036    Sybil Mcmunn       17      T-shirt        53
2593         2594     Ira Coryell       13        Pants        36

     invoice_date  Price
3525   13-03-2022   3500
1739    12/2/2021   2000
3187    1/11/2021   2000
2035   22-07-2022   2000
2593    2/12/2022   2000
(780, 7)
```

We have employed the sklearn library for splitting the dataset and the train dataset is 80% of the total while test is the remaining 20%.

5. Evaluate and improve model accuracy

Accuracy is a measure to know how well or badly a model is doing on an unseen validation set. Based on the current learning, evaluate the model on validation sets. We train and test the model with the help of the dataset that we have split. For that we have used algorithms like –

1 Logistic regression
2 K nearest neighbor
3 Random Forest
4 Neural Network
5 Perceptron
6 MLP
7 Decision Tree Classifier
8 SVM
9 Naïve Bayes

Phase II

Front-end Part

1. HTML

The Hyper Text Markup Language or HTML is the standard markup language for documents designed to be displayed in a web browser. It is often assisted by technologies such as Cascading Style Sheets (CSS) and scripting languages such as JavaScript.

HTML elements are the building blocks of HTML pages. With HTML constructs, images and other objects such as interactive forms may be embedded into the rendered page. HTML provides a means to create structured documents by denoting structural semantics for text such as headings, paragraphs, lists, links, quotes, and other items. HTML elements are delineated by tags, written using angle brackets. Tags such as <img /> and <input /> directly introduce content into the page. Other tags such as

<p> and </p> surround and provide information about document text and may include sub-element tags. Browsers do not display the HTML tags but use them to interpret the content of the page.

2. CSS

Cascading Style Sheets (CSS) is a style sheet language used for describing the presentation of a document written in a markup language such as HTML or XML (including XML dialects such as SVG, MathML or XHTML). CSS is a cornerstone technology of the World Wide Web, alongside HTML and JavaScript.

CSS is designed to enable the separation of content and presentation, including layout, colors, and fonts. This separation can improve content accessibility; provide more flexibility and control in the specification of presentation characteristics; enable multiple web pages to share formatting by specifying the relevant CSS in a separate

.css file, which reduces complexity and repetition in the structural content; and enable the .css file to be cached to improve the page load speed between the pages that share the file and its formatting.

3. Bootstrap

Bootstrap is a free and open-source CSS framework directed at responsive, mobile-first front-end web development. It contains HTML, CSS and (optionally) JavaScript-based design templates for typography, forms, buttons, navigation, and other interface components.

Bootstrap is an HTML, CSS and JS library that focuses on simplifying the development of informative web pages (as opposed to web applications). The primary purpose of adding it to a web project is to apply Bootstrap's choices of color, size, font and layout to that project. As such, the primary factor is whether the developers in charge find those choices to their liking. Once added to a project, Bootstrap provides basic style definitions for all HTML elements. The result is a uniform appearance for prose, tables and form elements across web browsers. In addition, developers can take advantage of CSS classes defined in Bootstrap to further customize the appearance of their contents. For example, Bootstrap has provisioned for light- and dark-colored tables, page headings, more prominent pull quotes, and text with a highlight.

Bootstrap also comes with several JavaScript components which do not require other libraries like jQuery. They provide additional user interface elements such as dialog boxes, tooltips, progress bars, navigation drop-downs, and carousels. Each Bootstrap component consists of an HTML structure, CSS declarations, and in some cases accompanying JavaScript code. They also extend the functionality of some existing interface elements, including for example an auto-complete function for input fields.

4. JavaScript

JavaScript is a high-level, often just-in-time compiled language that conforms to the ECMAScript standard. It has dynamic typing, prototype-based object-orientation, and first-class functions. It is multi-paradigm, supporting event-driven, functional, and imperative programming styles. It has application programming interfaces (APIs) for working with text, dates, regular expressions, standard data structures, and the Document Object Model (DOM).

JavaScript engines were originally used only in web browsers, but are now core components of some servers and a variety of applications. The most popular runtime system for this usage is Node.js.

Although Java and JavaScript are similar in name, syntax, and respective standard libraries, the two languages are distinct and differ greatly in design.

5. Flask

Flask is a micro web framework written in Python. It is classified as a microframework because it does not require particular tools or libraries. It has no database abstraction layer, form validation, or any other components where pre-existing third-party libraries provide common functions. However, Flask supports extensions that can add application features as if they were implemented in Flask itself. Extensions exist for object-relational mappers, form validation, upload handling, various open authentication technologies and several common framework related tools.

6. Prediction

Test the model on unknown data or real-time data, After the system starts working properly, the model is complete.

Back-end Part (Model Part)

For the backend, that is for the actual cardiovascular disease prediction system, we developed a model where we have developed a model. The model is a classification model which will predict whether or not a person has heart disease.

However, this is not the objective of our project. We wanted to develop a model which can also show the risk percentage along with the possibility of having a heart disease.

For that purpose, we have developed a full stack project, and have coded the risk prediction part separately which will be explained further.

Libraries used:

```
[1] import numpy as np
    import pandas as pd
    import matplotlib.pyplot as plt
    import seaborn as sns
    %matplotlib inline
```

```
[2] from sklearn.preprocessing import LabelEncoder
    from sklearn.model_selection import train_test_split, cross_val_score, GridSearchCV
```

```
[3] from sklearn.linear_model import LogisticRegression, Perceptron
    from sklearn.svm import SVC, LinearSVC
    from sklearn.ensemble import RandomForestClassifier
    from sklearn.neighbors import KNeighborsClassifier
    from sklearn.naive_bayes import GaussianNB
    from sklearn.tree import DecisionTreeClassifier
    from sklearn import metrics
```

```
[4] import keras
    from keras.models import Sequential
    from keras.layers import Dense, Dropout
    from keras import optimizers
    #from keras.wrappers.scikit_learn import KerasClassifier
    from keras.callbacks import EarlyStopping, ModelCheckpoint
```

```
[5] from hyperopt import STATUS_OK, Trials, fmin, hp, tpe, space_eval
```

```
[6] from warnings import simplefilter
    simplefilter(action='ignore', category=FutureWarning)
```

Using the pandas library, we loaded the dataset which is as follows –

```
[7] df= pd.read_csv("product.csv")
```

```
[8] df
```

	Customer ID	Customer Name	Product	Product Name	Quantity	invoice_date	Price	Target
0	1	Pattie Morrisey	1	Blouse	55	5/8/2022	2000	4
1	2	Traci Peery	2	Sweater	19	12/12/2021	2000	1
2	3	Merideth Mcmeen	3	Jeans	50	9/11/2021	3500	3
3	4	Eufemia Cardwell	4	Sandals	21	16-05-2021	7000	2
4	5	Meghan Kosak	1	Blouse	45	24-10-2021	5000	3
...	...	...	...	...	...	...	...	...
3895	3896	Herma Voisine	15	Hoodie	40	14-09-2022	2000	2
3896	3897	Evelina Chartrand	21	Backpack	52	10/3/2021	5000	4
3897	3898	Sherie Shaughnessy	22	Belt	46	6/8/2022	4500	3
3898	3899	Janell Neubauer	9	Shoes	44	6/8/2021	9000	3
3899	3900	Antonietta Tann	8	Handbag	52	4/3/2022	3000	4

3900 rows × 8 columns

Chapter 6
Reports

1. The overview of the dataset used for training and testing:

Overview

Overview | Alerts 6 | Reproduction

Dataset statistics

Number of variables	8
Number of observations	3900
Missing cells	0
Missing cells (%)	0.0%
Duplicate rows	0
Duplicate rows (%)	0.0%
Total size in memory	243.9 KiB
Average record size in memory	64.0 B

Variable types

Numeric	4
Text	1
Categorical	2
DateTime	1

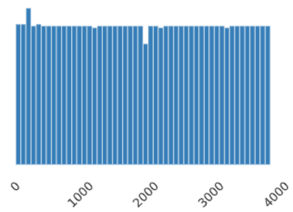
2. Variable:

Customer ID

Customer ID
Real number (ℝ)

UNIFORM

Distinct	3886	Minimum	1
Distinct (%)	99.6%	Maximum	3900
Missing	0	Zeros	0
Missing (%)	0.0%	Zeros (%)	0.0%
Infinite	0	Negative	0
Infinite (%)	0.0%	Negative (%)	0.0%
Mean	1944.3154	Memory size	30.6 KiB



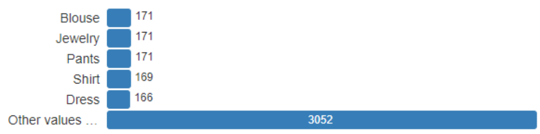
More details

Product Name

Product Name
Categorical

HIGH CORRELATION

Distinct	25
Distinct (%)	0.6%
Missing	0
Missing (%)	0.0%
Memory size	30.6 KiB



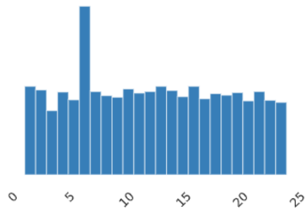
More details

Product

Product
Real number (ℝ)

HIGH CORRELATION

Distinct	24	Minimum	1
Distinct (%)	0.6%	Maximum	24
Missing	0	Zeros	0
Missing (%)	0.0%	Zeros (%)	0.0%
Infinite	0	Negative	0
Infinite (%)	0.0%	Negative (%)	0.0%
Mean	12.147949	Memory size	30.6 KiB



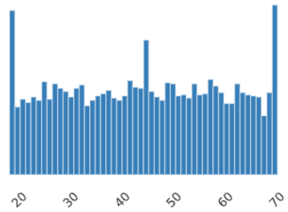
More details

Quantity

Quantity
Real number (ℝ)

HIGH CORRELATION

Distinct	53	Minimum	18
Distinct (%)	1.4%	Maximum	70
Missing	0	Zeros	0
Missing (%)	0.0%	Zeros (%)	0.0%
Infinite	0	Negative	0
Infinite (%)	0.0%	Negative (%)	0.0%
Mean	44.068462	Memory size	30.6 KiB

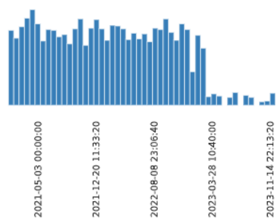


More details

invoice_date

invoice_date
Date

Distinct	791	Minimum	2021-01-02 00:00:00
Distinct (%)	20.3%	Maximum	2023-12-02 00:00:00
Missing	0		
Missing (%)	0.0%		
Memory size	30.6 KiB		



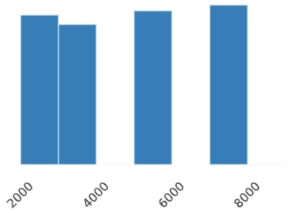
More details

Price

Price

Real number (ℝ)

Distinct	7	Minimum	2000
Distinct (%)	0.2%	Maximum	9000
Missing	0	Zeros	0
Missing (%)	0.0%	Zeros (%)	0.0%
Infinite	0	Negative	0
Infinite (%)	0.0%	Negative (%)	0.0%
Mean	4437.1795	Memory size	30.6 KiB



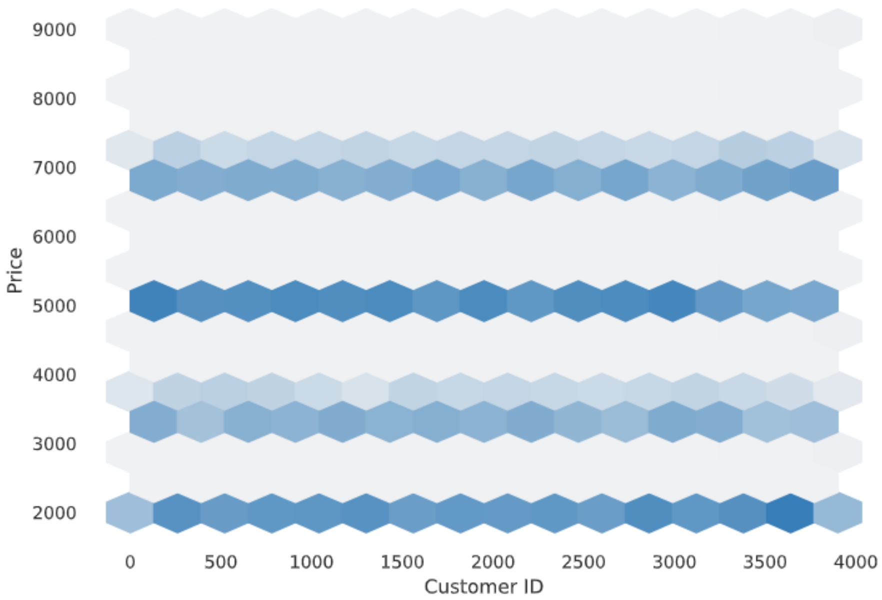
More details

3. Interactions

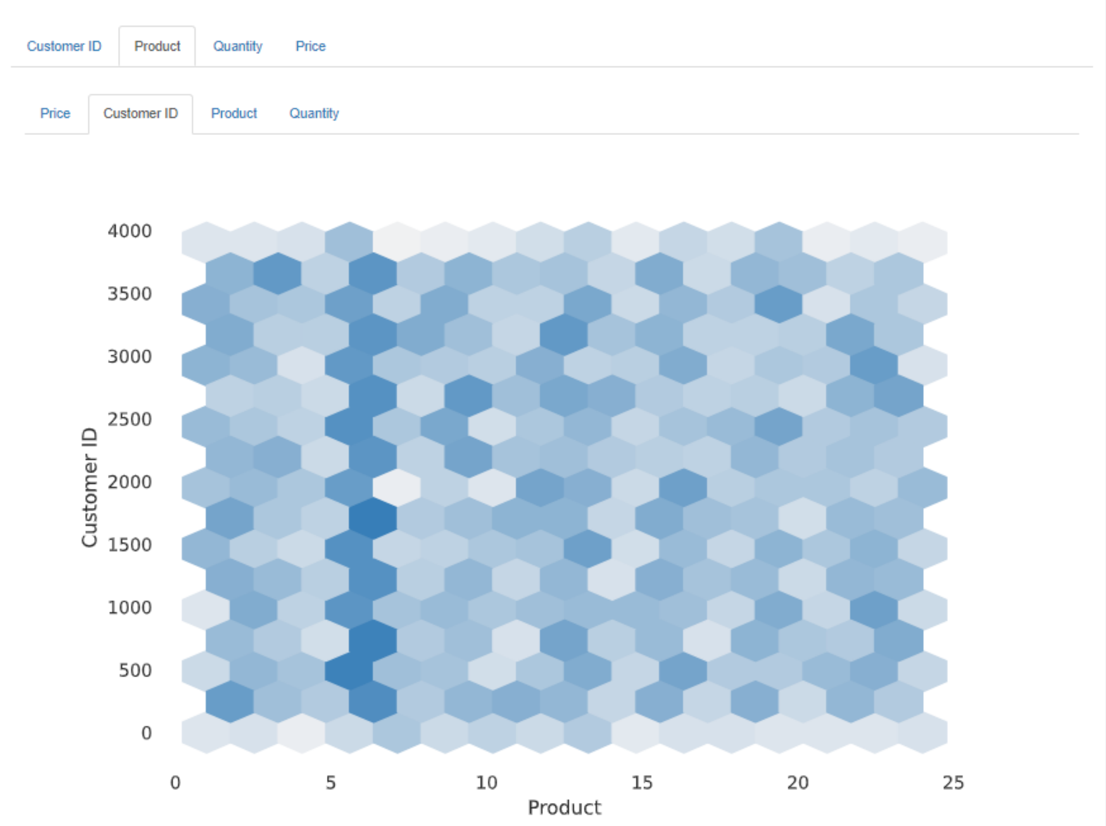
Customer Id and Price

Customer ID | Product | Quantity | Price

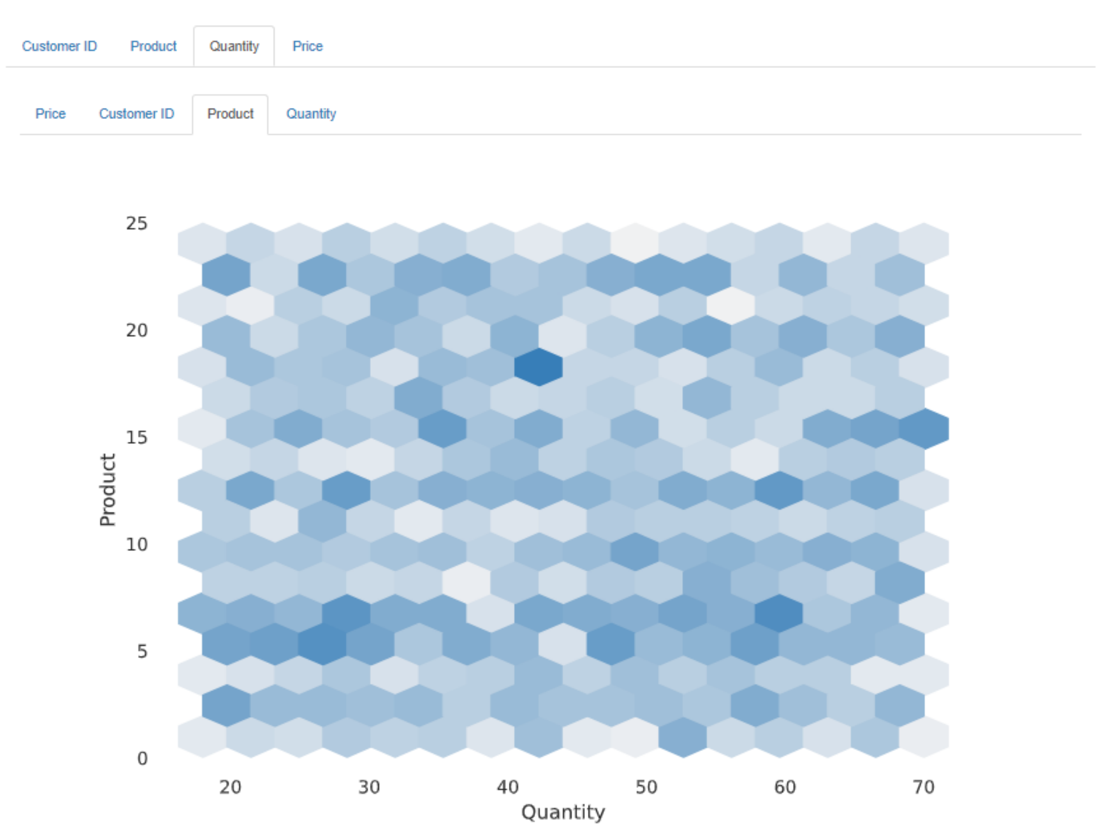
Price | Customer ID | Product | Quantity



Product and Customer Id

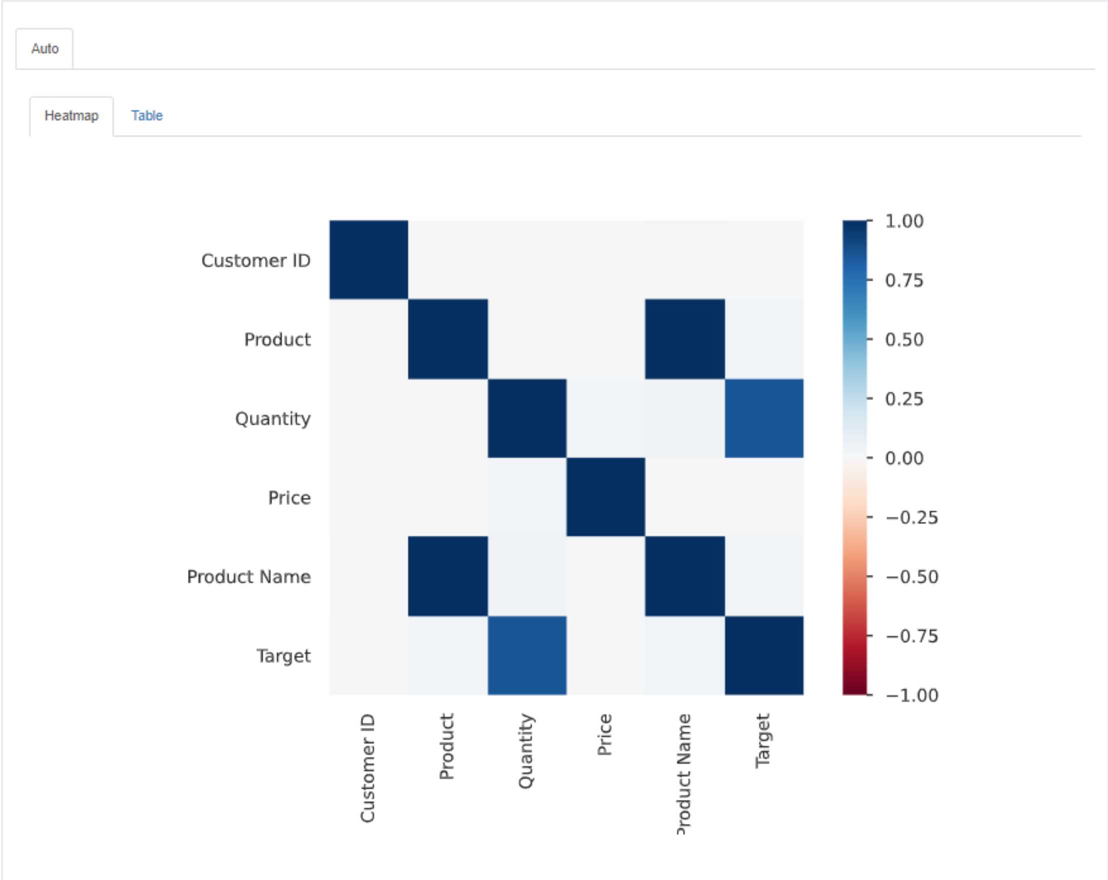


Quantity and Product

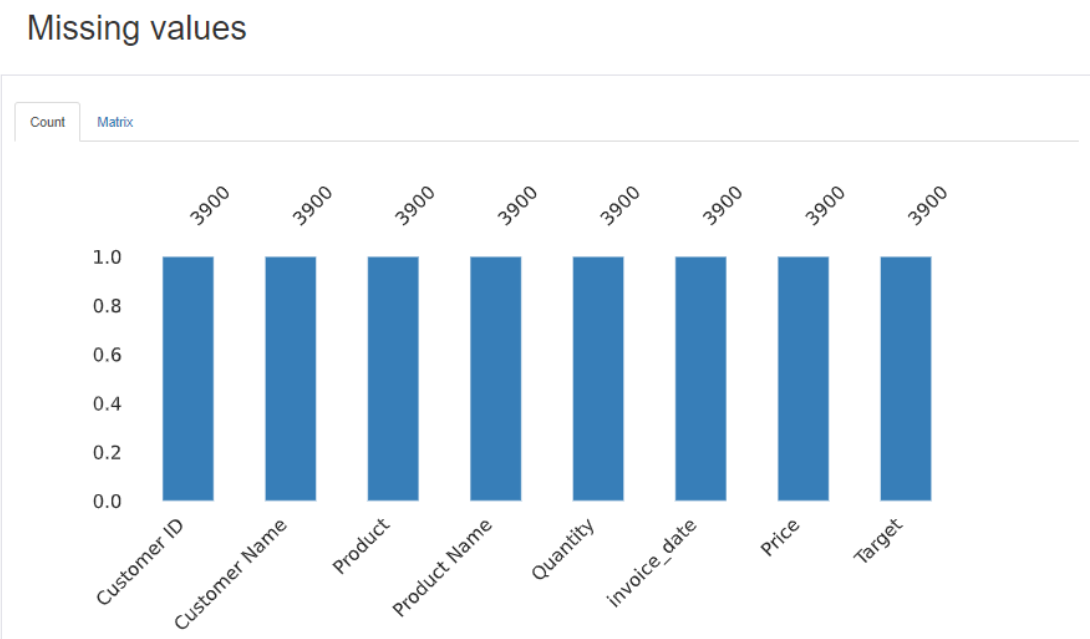


Price and Quantity

4. Correlations



5. Missing Values



A simple visualization of nullity by column.

6. Dataset

First rows Last rows

	Customer ID	Customer Name	Product	Product Name	Quantity	invoice_date	Price	Target
0	1	Pattie Morrisey	1	Blouse	55	5/8/2022	2000	4
1	2	Traci Peery	2	Sweater	19	12/12/2021	2000	1
2	3	Merideth Mcmeen	3	Jeans	50	9/11/2021	3500	3
3	4	Eufemia Cardwell	4	Sandals	21	16-05-2021	7000	2
4	5	Meghan Kosak	1	Blouse	45	24-10-2021	5000	3
5	6	Leslie Browder	5	Sneakers	46	24-05-2022	3500	3
6	7	Bridget Balog	6	Shirt	63	13-03-2022	7000	5
7	8	Herma Torgeson	6	Shorts	27	13-01-2021	2000	2
8	9	Pattie Helmers	7	Coat	26	4/11/2021	3500	2
9	10	Shaquana Leech	8	Handbag	57	22-08-2021	2000	4

First rows Last rows

	Customer ID	Customer Name	Product	Product Name	Quantity	invoice_date	Price	Target
3890	3891	Masako Rodenberger	6	Shirt	35	15-08-2021	3500	2
3891	3892	Enola Lintner	10	Dress	36	3/5/2021	5000	2
3892	3893	Marilou Chadwell	16	Jewelry	35	24-03-2022	7000	2
3893	3894	Delcie Trottier	19	Hat	21	14-09-2022	3500	2
3894	3895	Stefany Clevenger	11	Skirt	66	25-03-2021	2000	5
3895	3896	Herma Voisine	15	Hoodie	40	14-09-2022	2000	2
3896	3897	Evelina Chartrand	21	Backpack	52	10/3/2021	5000	4
3897	3898	Sherie Shaughnessy	22	Belt	46	6/8/2022	4500	3
3898	3899	Janell Neubauer	9	Shoes	44	6/8/2021	9000	3
3899	3900	Antonietta Tann	8	Handbag	52	4/3/2022	3000	4

We have finally selected Random Forest to be the model which we will use to make predictions because it gave us the highest accuracy –

For training we received – 100 %

For test we received – 70.86 %

Chapter 7
Conclusion
And
Future Work

Conclusion

In summary, the development of a product requirement prediction system holds significant promise for various industries. By leveraging advanced machine learning algorithms and data analytics techniques, organizations can streamline their product development processes, enhance resource allocation, and improve customer satisfaction. Throughout this project, we have explored different methodologies, such as natural language processing, predictive modeling, and data visualization, to accurately forecast product requirements based on historical data and market trends.

Our findings demonstrate the feasibility and effectiveness of employing predictive analytics in anticipating future product requirements. By analyzing past sales data, customer feedback, market trends, and other relevant factors, we have successfully developed models capable of forecasting demand patterns and identifying key features or attributes that drive consumer preferences. Moreover, by integrating real-time data sources and continuously updating our models, we can adapt to changing market dynamics and ensure the accuracy and relevance of our predictions over time.

Future Work

While we have made significant progress in building a product requirement prediction system, there are several avenues for future research and development to explore:

- Enhanced Data Integration: Incorporating additional data sources, such as social media sentiment analysis, competitor analysis, and economic indicators, can provide deeper insights into consumer behavior and market trends, further improving the accuracy of our predictions.
- Advanced Machine Learning Techniques: Experimenting with advanced machine learning algorithms, such as deep learning and ensemble methods, can help uncover complex patterns and relationships within the data, leading to more accurate and robust prediction models.
- Personalized Recommendations: Tailoring product recommendations to individual customer preferences and behavior can enhance the overall user experience and increase customer engagement. Developing personalized recommendation engines based on predictive analytics can facilitate targeted marketing campaigns and drive sales growth.
- Integration with Supply Chain Management: Integrating product requirement predictions with supply chain management systems can optimize inventory management, production planning, and distribution logistics, minimizing stockouts, reducing excess inventory, and lowering operating costs.

Chapter 8
References

References

- http://ieeexplore.ieee.org/xpl/articleDetails.jsp?arnumber=6864427&newsearch=true&queryText=disease%20prediction
- http://ieeexplore.ieee.org/xpl/articleDetails.jsp?arnumber=6395001&newsearch=true&queryText=disease%20prediction