**FASHION RECOMMENDATION SYSTEM USING ML**

Submitted By

**K.M. AKASH**

**Register No:910622301003**

**A PROJECT REPORT**

Submitted to the

FACULTY OF INFORMATION & COMMUNICATION ENGINEERING

In partial fulfillment of the requirements for the award of the degree of

### **MASTER OF COMPUTER APPLICATIONS (MCA)**

**IN**

### **K.L.N. COLLEGE OF ENGINEERING**

(An Autonomous Institution)

### **POTTAPALAYAM, SIVAGANGAI – 630 612**

**ANNA UNIVERSITY, CHENNAI-600 025 JULY 2024**

### 

### **K.L.N. COLLEGE OF ENGINEERING**

### (An Autonomous Institution)

**DEPARTMENT OF MASTER OF COMPUTER APPLICATIONS(MCA)**



### **BONAFIDE CERTIFICATE**

Certified that this project report titled “**FASHION RECOMMENDATION SYSTEM USING MACHINE LEARNING”** is the bonafide work of Mr. **K.M.AKASH (Reg.No:910622301003)** who carried out the research under my supervision. Certified further, that to the best of my knowledge the work reported herein does not form part of any other project report or dissertation based on which a degree or award was conferred on an earlier occasion on this or any other candidate.

### 

### **Dr. M.R ILANGO,MCA., M Phil., Ph.D.**

**DIRECTOR/MCA,**

**K.L.N.COLLEGE OF ENGINEERING**

### **Dr. M.R ILANGO,MCA., M Phil., Ph.D**

**DIRECTOR/MCA, INTERNAL GUIDE,**

**K.L.N.COLLEGE OF ENGINEERING**

Certified that the candidate was examined by us in the viva-voce examination held at

**K.L.N.COLLEGE OF ENGINEERING**, SIVAGANGAI on …………………………………

### **INTERNAL EXAMINER EXTERNAL EXAMINER**

**PROJECT COMPLETION CERTIFICATE**

**INFINITI MEDIA: 2023-2024 Date: 06/06/2024**

**TO WHOM SO EVER IT MAY CONCERN**

This is to certify that **Mr K M. AKASH (Reg. No: 910622301003)** student of Final year **MCA** from **K.L.N COLLEGE OF ENGINEERING, POTTAPALAYAM**. He has successfully completed his **PROJECT INTERNSHIP TRAINING** Titles on **“FASHION RECOMMENDATION SYSTEM USING MACHINE LEARNING”** in the platform of **Python** in our organization **INFINITI MEDIA** during the period of **January 2024 to June 2024**. We have that during the period he has shown keen interest in his assignment and was also regular attendance with 100%.

With best regards,

Project Coordinator

**INFINITI MEDIA**

**ACKNOWLEDGEMENT**

I would like to express my sincere thanks to our honorable Management and our respected **PRINCIPAL, Dr. A.V. RAM PRASAD M.E., Ph.D.,** for his support and wonderful opportunity for doing the project in our college.

I would like to express my sincere thanks to our Respected **DIRECTOR Dr. M.R. ILANGO MCA., M.Phil., Ph.D.,** Director of MCA Department, for his permission, guidance and support in preceding the project.

I would like to express my sincere thanks to our Respected **INTERNAL GUIDE Dr. M.R ILANGO MCA., M.Phil., Ph.D., DIRECTOR/MCA** for his guidance and support inpreceeding the project.

I would like to express my sincere thanks to my Project team **Infiniti Technologies Pvt. Ltd, Madurai,** for their immense guidance and valuable suggestion in completing my project successfully.

I thank my parents, faculty members of MCA Department for their moral support to complete the project in a successful manner.

A project is never outcome of a single person’s efforts, It is a confluence of a varied thought process harmoniously integrated into a resourceful product. It is, but natural that I feel indebted to several people for having made this project possible.

I ABSTRACT

This study presents the design and implementation of a machine learning-based chatbot system for fashion recommendation. Leveraging natural language processing (NLP) techniques, the chatbot interacts with users to understand their preferences, style preferences, and occasions, enabling personalized fashion suggestions. The system employs various machine learning algorithms, including collaborative filtering, content-based filtering, and deep learning models, to analyze user inputs and historical fashion data. By integrating these algorithms into a conversational interface, the chatbot provides tailored recommendations, taking into account factors such as user demographics, clothing attributes, and current fashion trends. The system's effectiveness is evaluated through user testing and feedback, demonstrating its capability to deliver accurate and engaging fashion advice while enhancing user satisfaction and engagement in the fashion domain. Overall, this research contributes to advancing the field of AI-driven fashion recommendation systems and showcases the potential of machine learning-powered chatbots in delivering personalized experiences in the fashion industry.

**TABLE OF CONTENTS**

|  |  |  |
| --- | --- | --- |
| **CHAPTER** | **TITLE** | **PAGE NO** |
|  | **ABSTRACT** | I |
|  | **LIST OF TABLES** |  |
|  | **LIST OF FIGURES** |  |
|  |  |  |
|  | **INTRODUCTION** | 1 |
|  | * 1. ORGANIZATION PROFILE | 1 |
|  | * 1. PROJECT DESCRIPTION | 5 |
|  | **SYSTEM** **ANALYSIS** |  |
|  | * 1. EXISTING SYSTEM | 6 |
|  | * + 1. DISADVANTAGES | 6 |
|  | * 1. PROPOSED SYSTEM | 7 |
|  | 2.2.1 ADVANTAGES | 7 |
|  | * 1. ABOUT THE PROJECT | 8 |
|  | * 1. FEASIBILITY STUDY | 8 |
|  | **SYSTEM DESIGN** |  |
|  | * 1. SYSTEM DESIGN GOALS | 10 |
|  | * 1. SYSTEM ARCHITECTURE | 13 |
|  | * 1. FLOW DIAGRAM | 14 |
|  | * 1. DATA FLOW DIAGRAM | 15 |
|  | * 1. UML DIAGRAM | 17 |
|  | **MODULES DESCRIPTION** |  |
|  | * 1. MODULES | 22 |
|  | * 1. MODULES DESCRIPTION | 22 |
|  | **SYSTEM** **REQUIREMENTS AND ANALYSIS** |  |
|  | * 1. SOFTWARE   & HARDWARE REQUIREMENTS | 25 |
|  | * 1. SOFTWARE EXPLANATION | 25 |
|  | **TESTING AND IMPLEMENTATION** |  |
|  | 6.1 INTRODUCTION | 29 |
|  | 6.2 STRATEGIC APPROACH TO SOFTWARE TESTING | 29 |
|  | 6.3 SYSTEM TESTING | 31 |
|  | 6.4 IMPLEMENTATION | 34 |
|  | **CONCLUSION** | 36 |
|  | **FUTURE** **ENHANCEMENT** | 37 |
|  | **APPENDIX** |  |
|  | 9.1 SOURCE CODE | 38 |
|  | 9.2 SCREEN SHOTS | 50 |
| **10.** | **BIBLIOGRAPHY** | 55 |

III LIST OF FIGURES

|  |  |  |
| --- | --- | --- |
| **FIGURE NO** | **TITLE** | **PAGE NO** |
| 1 | SYSTEM ARCHITECTURE | 12 |
| 2 | FLOW DIAGRAM | 14 |
| 3 | DATA FLOW DIAGRAM | 15 |
| 4 | USE CASE DIAGRAM | 17 |
| 5 | ACTIVITY DIAGRAM | 18 |
| 6 | SEQUENCE DIAGRAM | 19 |
| 7 | ER DIAGRAM | 20 |
| 8 | CLASS DIAGRAM | 21 |

**CHAPTER 1**

**INTRODUCTION**

* 1. **Organization Profile:**

**Infiniti Technologies Private Limited**, we have 22+ years of experience in developing cutting-edge technologies and data-driven systems. Infiniti Technologies, a Top Rated Data Science Consulting Company offering Data Science Services such as leveraging Artificial Intelligence and Machine Learning technologies to build higher value technical solutions worldwide. Our strength in providing the best data science consulting services, which means you can be sure that your business is in good hands. Our data services such as data science, data analytics, data engineering, data leaks. Our intelligent services such as computer vision, artificial intelligence, machine learning and deep learning.

Infiniti Technologies is one of the top data science consulting companies in India. As a multidisciplinary team of data science consultants, we mainly focus on providing practical business value with data science skills. The significant advantage of working with Infiniti Technologies is our commercially focused data science approach.

Our in-depth business experience combined with our broad range of technical skills and the domain knowledge allows us to help you focus on applying data science to the most important commercial opportunities.

**Why choose us:**

**Dedication:** We’re dedicated to our clients. Your requirements and difficulties always come first, and only with an in-depth knowledge of your business can we solve those problems.

**Credibility:** Data science is complex and also the dynamic industry that are frequently misunderstood. That’s why our credibility is a primary cornerstone of Infiniti Technologies identity.

**Efficiency:** We care about efficiency as much as our clients do. From our project management tools to our development process and to our document work, effectiveness is the main core.

**Rigour:** Each step of our process is defined by rigour. We validate results before we share them. We’re upfront with clients when we come across challenges in their data.

**Our mission:**

* Our Mission Is to Bring the Power of Data Science to Every Business:
* To create value for our customers by providing consulting services
* To leverage data and maximize ROI of being a data-driven business
* To lead a client from project inception to integration and deployment
* Flexibility to consume the information in a predictive or prescriptive manner
* Develop and implement comprehensive data strategies
* To move security in the right direction with comprehensive consulting services

**Data Science Consulting Services:**

Infiniti Technology is a Top Rated Data Science Consulting Company offering Data Science Services leveraging Artificial Intelligence and also Machine Learning technologies to develop higher value technical solutions to Data science for consulting customers worldwide. We implement end-to-end Big Data solutions with Data Science Consulting Technologies such as machine learning, Artificial Intelligence and Deep learning seamlessly baked into it. We offer result-oriented Data Science Consulting Services using the latest tools and technologies to identify and solve your most complex data science challenges strategically.

Our Data science consultants use present and historical data to build popular data science solutions-predictive modelling that identifies trends and data science patterns and uses those insights to predict future outcomes. In addition, they can use logistic regression, time series analysis, and decision trees to build unsupervised models. Besides leveraging recent machine learning, neural networks, and standard algorithms, they can build supervised predictive models whether you want domain-specific or industry-specific predictive models.

**Best Data Analytic Consulting Company**

At Infiniti Technology, we bring a fresh approach to the challenges of data management, intelligent data discovery, Data Analytic Consulting, insight and also business intelligence. Data Analytics Company helps individuals and organizations understand their data. Data analysts Consulting typically analyze raw data to look for insights and trends. Collect, transform, and organize data to draw conclusions, make predictions, and make informed decisions.

Our agile and integrated development methodology, product and service offerings, many years of experience and industry-leading software tools, recognise us to deliver advanced data analytics and business intelligence complete solutions – data solutions that give you a real edge in an ever more ambitious marketplace.

**Data Engineering Services:**

Data Engineering Services transform organizational knowledge into insights for informed and timely business decisions. Data Engineering Services is constructing raw data from data scientists and organizational levels. Infiniti Technology allows its clients to get the most utmost out of their data by offering an entire set of data engineering services and solutions that optimize your analytics, data science and also data warehouse initiatives. We connect our services with ETPL Digital Accelerators to generate solutions for companies to power up a ‘Data as a Service’ capability that is critical to allow access to data across the Enterprise. In addition, it helps to accelerate the integration of analytics into business processes as well as, reduce the time to value from digital assets. The digital natives have organised data as a source of competitive benefit by treating it as a core Business Asset. This mindset has driven a great capability in developing the technology and process infrastructure around data and contributing to the enterprise as ‘Data as a Service.

**Our Approach – Data Leaks**

Data leakage prevention solutions can assist to safeguard your most critical information assets. From enterprise plans and intellectual resources to customer data and especially identifiable data, much of your most sensitive data can simply leak via email. Sometimes it’s an accident – an employee begins the opposite recipient name when sending an email. Sometimes it’s malicious – an insider informs a file or steals sensitive information by emailing it to someone outside the company. At the same time, data loss prevention technology can help to stop leaks by checking email from leaving the organization.

* 1. **Project Description:**

A Fashion Recommendation System using Machine Learning leverages algorithms to provide personalized fashion suggestions to users based on their preferences, purchase history, and browsing behaviour. By analysing large datasets, including user interactions, product features, and trends, the system can predict and recommend clothing items, accessories, and styles that align with individual tastes. This project aims to enhance the shopping experience by offering tailored recommendations, increasing user engagement, and boosting sales for fashion retailers. The system incorporates techniques such as collaborative filtering, content-based filtering, and deep learning to continuously learn and adapt to evolving fashion trends and user preferences, ultimately delivering a highly customized and relevant fashion discovery process.

**CHAPTER 2**

**SYSTEM ANALYSIS**

**2.1 EXISTING SYSTEM:**

In existing system, Recommendation technology is an essential component of Internet of Things (IoT) services, as it enhances user experience by providing personalized and timely information. A novel recommendation model, called TCCF (Time Correlation Coefficient and an improved K-means with Cuckoo Search), has been developed to improve the accuracy and efficiency of recommendations in IoT scenarios. This model utilizes time correlation coefficients to refine its recommendations. Additionally, it incorporates a sophisticated clustering method known as CSK-means, which is an enhanced version of the traditional K-means algorithm, optimized with the Cuckoo Search technique. This clustering method effectively groups similar users, enabling quick and precise recommendations. Systematic experimental results have demonstrated the effectiveness of both the TCCF model and its enhanced version, PTCCF, in delivering accurate and efficient recommendations in IoT environments, thereby significantly improving user satisfaction.

**2.1.1 DISADVANTAGES:**

* Most users do not rate most items and hence the user rating matrix is typically very less.
* This is a problem for clustering method for recommendation system since it decreases the probability of finding a set of users with similar ratings.
  1. **PROPOSED SYSTEM:**

In the proposed system, products are recommended based on user inputs using a collaborative filtering approach. The core concept of collaborative filtering is to identify other users within the community who share similar opinions and preferences. Recommendations are then calculated based on the inputs from these like-minded users. Following this, a machine learning classification algorithm, such as a Decision Tree, is implemented to refine the recommendations. The experimental results demonstrate the system's capability to recommend products tailored to individual user IDs. Additionally, the results showcase the system's performance metrics, including accuracy, precision, recall, and F1-score, highlighting the effectiveness of the recommendation system.

**2.2.1 ADVANTAGES:**

* Collaborative filtering performs the good recommendation.
* In addition, machine learning algorithm is implemented.
* The classification of data is accurate by using the classification algorithm.
* It enhance the overall performances.
  1. **ABOUT THE PROJECT:**

This study outlines the design and implementation of a machine learning-based chatbot system for fashion recommendations. By leveraging natural language processing (NLP) techniques, the chatbot interacts with users to understand their style preferences and the occasions they are dressing for, enabling it to offer personalized fashion suggestions. The system employs a variety of machine learning algorithms, including collaborative filtering, content-based filtering, and deep learning models, to analyse user inputs and historical fashion data. Integrating these algorithms into a conversational interface, the chatbot provides customized recommendations that consider user demographics, clothing attributes, and current fashion trends. The system's effectiveness is evaluated through user testing and feedback, demonstrating its capability to deliver accurate and engaging fashion advice while enhancing user satisfaction and engagement. Overall, this research advances the field of AI-driven fashion recommendation systems and showcases the potential of machine learning-powered chatbots to provide personalized experiences in the fashion industry.

* 1. **FEASIBILITY STUDY:**

The feasibility study is carried out to test whether the proposed system is worth being implemented. The proposed system will be selected if it is best enough in meeting the performance requirements.

The feasibility carried out mainly in three sections namely.

**•** Economic Feasibility

• Technical Feasibility

• Behavioral Feasibility

**Economic Feasibility**

Economic analysis is the most frequently used method for evaluating effectiveness of the proposed system. More commonly known as cost benefit analysis. This procedure determines the benefits and saving that are expected from the system of the proposed system. The hardware in system department if sufficient for system development.

**Technical Feasibility**

This study center around the system’s department hardware, software and to what extend it can support the proposed system department is having the required hardware and software there is no question of increasing the cost of implementing the proposed system. The criteria, the proposed system is technically feasible and the proposed system can be developed with the existing facility.

**Behavioural Feasibility**

People are inherently resistant to change and need sufficient amount of training, which would result in lot of expenditure for the organization. The proposed system can generate reports with day-to-day information immediately at the user’s request, instead of getting a report, which doesn’t contain much detail.

**CHAPTER 3**

**SYSTEM DESIGN**

**3.1 SYSTEM DESIGN GOALS:**

When designing a system, several goals must be considered to ensure the system meets the needs of its users and stakeholders effectively. These goals can vary depending on the specific context and requirements, but here are some commonly considered goals in system design:

**1. Scalability**

Definition: The ability of the system to handle increased load (users, data volume, transactions) without performance degradation.

Strategies: Load balancing, horizontal scaling (adding more servers), vertical scaling (upgrading existing servers), database sharding, distributed computing.

**2. Performance**

Definition: The system's efficiency in terms of response time, throughput, and resource utilization.

Strategies: Optimizing algorithms, caching frequently accessed data, database indexing, using efficient data structures, minimizing network latency.

**3. Reliability**

Definition: The system's ability to function correctly and provide accurate results consistently over time.

Strategies: Redundancy, regular backups, failover mechanisms, robust error handling, thorough testing (unit, integration, system, user acceptance).

**4. Availability**

Definition: The proportion of time the system is operational and accessible.

Strategies: High availability architecture (HA), disaster recovery planning, clustering, load balancing, and automated failover.

**5. Maintainability**

Definition: The ease with which the system can be modified to fix defects, improve performance, or adapt to a changing environment.

Strategies: Modular design, clear documentation, adherence to coding standards, use of version control, continuous integration and deployment (CI/CD).

**6. Security**

Definition: Protecting the system against unauthorized access and ensuring data integrity and confidentiality.

Strategies: Encryption, authentication, authorization, regular security audits, intrusion detection systems (IDS), secure coding practices.

**7. Usability**

Definition: The ease with which users can learn and use the system.

Strategies: Intuitive user interface (UI) design, comprehensive user documentation, user training, feedback mechanisms, accessibility compliance.

**8. Extensibility**

Definition: The system’s ability to incorporate new features and functionalities without major redesign.

Strategies: Use of design patterns, modular architecture, plug-in architecture, open APIs.

**9. Interoperability**

Definition: The ability of the system to work with other systems or products without special effort.

Strategies: Adhering to industry standards, using common data formats (XML, JSON), APIs, middleware.

**10. Cost-effectiveness**

Definition: The balance between the system’s performance, features, and cost.

Strategies: Cost-benefit analysis, choosing appropriate technology stack, efficient resource utilization, leveraging open-source solutions.

**11. Compliance**

Definition: Ensuring the system adheres to relevant laws, regulations, and standards.

Strategies: Regular compliance audits, adherence to industry standards (e.g., GDPR for data protection), implementing compliance checks during development.

**12. Portability**

Definition: The ease with which the system can be transferred from one environment to another.

Strategies: Containerization (e.g., Docker), virtualization, use of platform-independent technologies.

**13. Adaptability**

Definition: The system’s ability to adapt to changing requirements and environments.

Strategies: Agile development practices, flexible design, regular updates and patches, continuous feedback and improvement loops.

**3.2 SYSTEM ARCHITECTURE:**

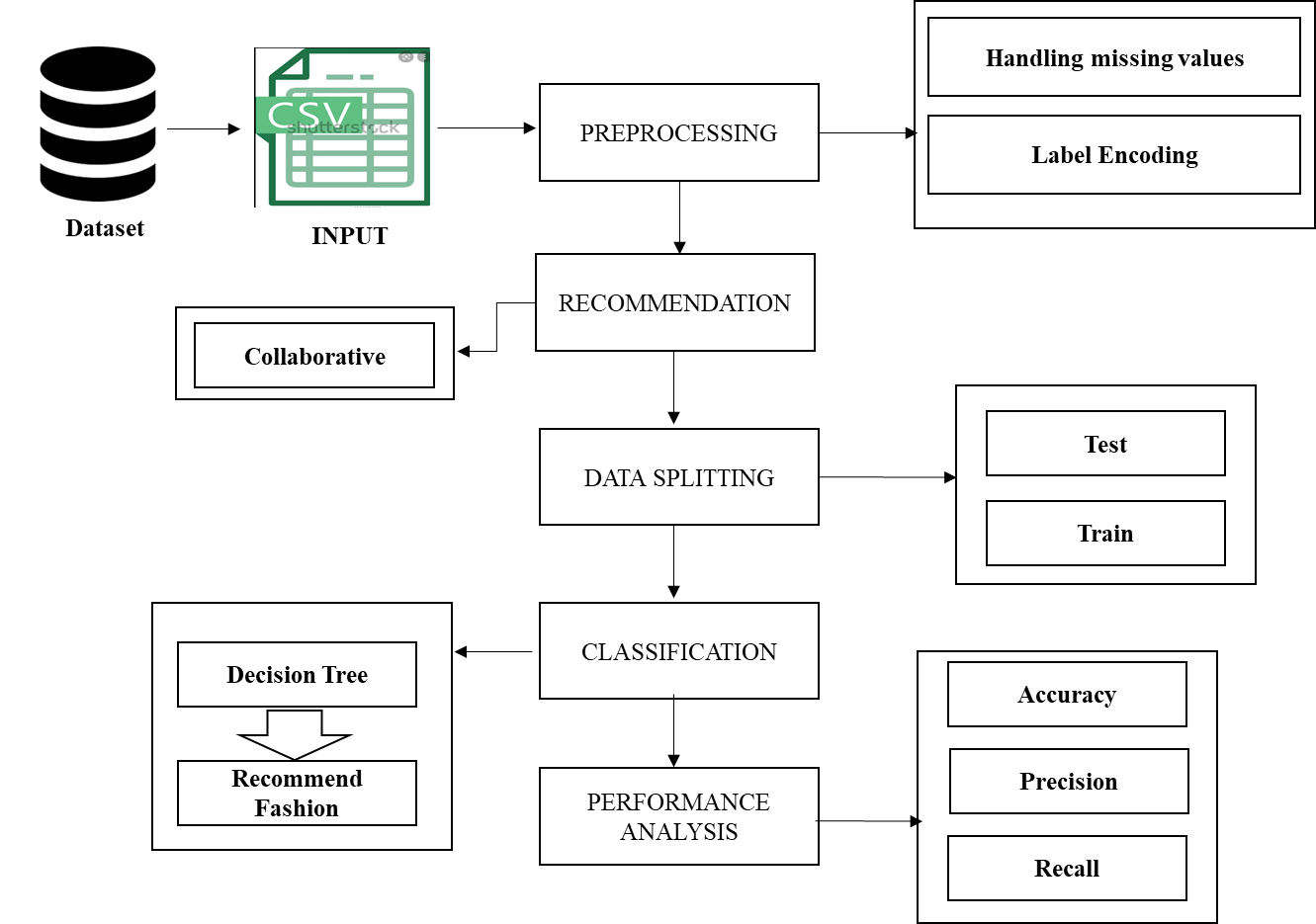
****

FIGURE 3.1: SYSTEM ARCHITECTURE

**3.2 FLOW DIAGRAM**

Input Data

Preprocessing

Recommendation

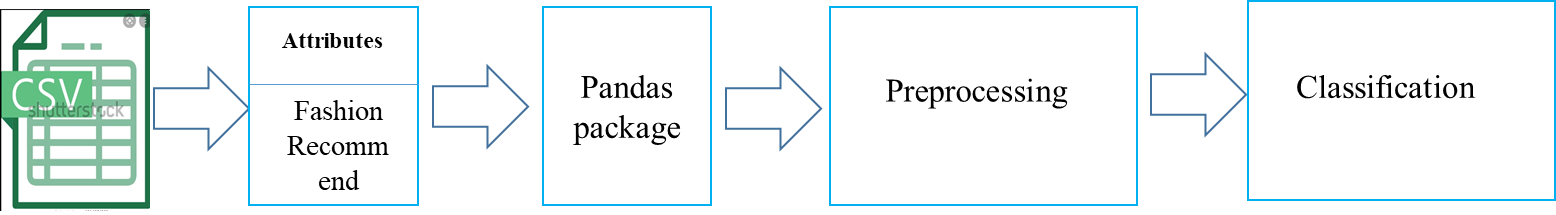
Classification

Performance analysis

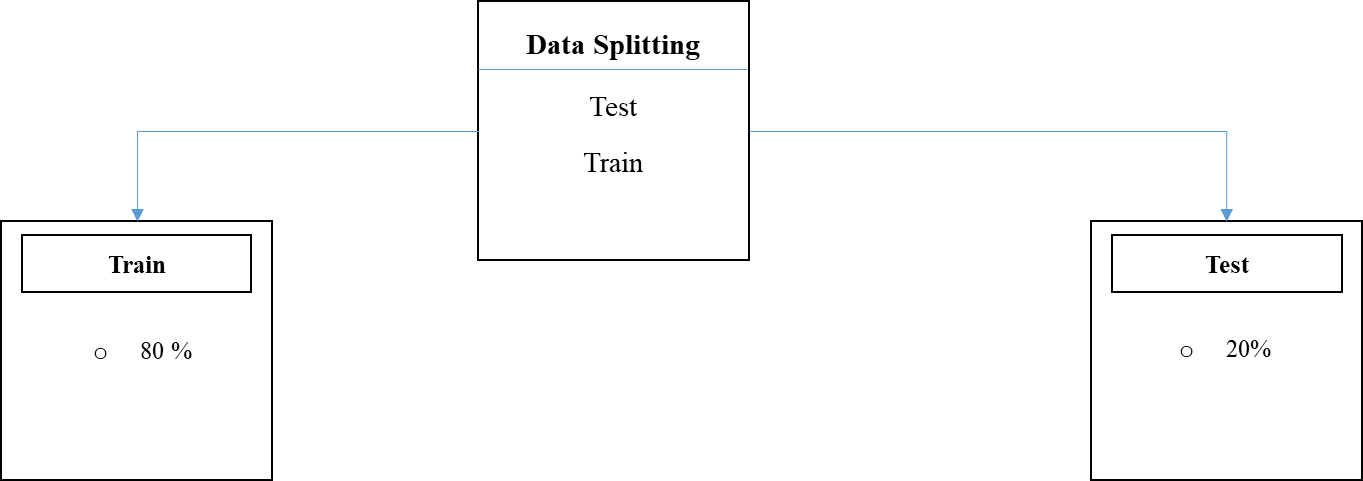
FIGURE 3.2: FLOW DIAGRAM

**3.3 DATA FLOW DIAGRAM:**

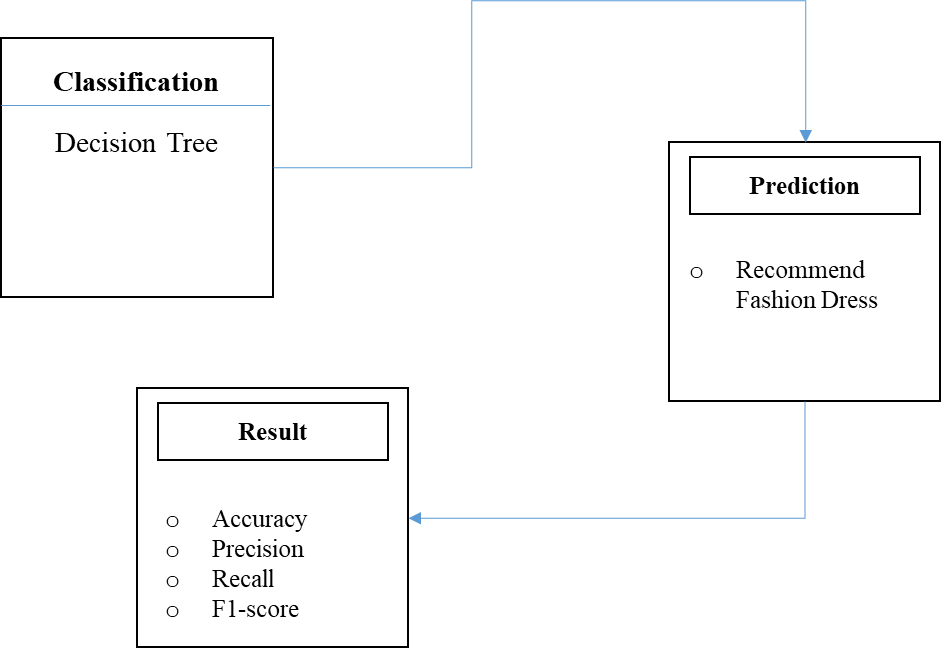
**Level 0:**

****

**Level 1:**



**Level 2:**



**3.4 UML DIAGRAMS:**

**3.4.1 USE CASE DIAGRAM:**

System

User

FIGURE 3.4.1: USE CASE DIAGRAM

**3.4.2 ACTIVITY DIAGRAM:**

Input Data

Preprocessing

Recommendation

Performance analysis

Classification

FIGURE 3.4.2: ACTIVITY DIAGRAM

**3.4.3 SEQUENCE DIAGRAM:**

Input Data

Preprocessing

Recommendation

Classification

Select data

Missing value

Collaborative

Load data

Data splitting

Decision Tree

FIGURE 3.4.3: SEQUENCE DIAGRAM

**3.4.4 ER DIAGRAM:**

Data selection

Preprocessing

Recommendation

Classification

FIGURE 3.4.4: ER DIAGRAM

**3.4.6 CLASS DIAGRAM:**

Select data ()

Load data ()

View data ()

INPUT

Collaborative ()

Recommendation

Accuracy ()

ROC ()

Performance analysis

Preprocessing

Missing values ()

Label encode ()

DT ()

Classification

FIGURE 3.4.5: CLASS DIAGRAM

**CHAPTER 4**

**MODULES DESCRIPTION**

**4.1 MODULES:**

* Data selection
* Data preprocessing
* Recommendation Score
* Data splitting
* Recommendation
* Classification
* Result Generation

**4.2 MODULES DESCRIPTION:**

**4.2.1: DATA SELECTION:**

* The dataset was collected from the dataset repository.
* The process of selecting the dataset for recommend the top products based on their users ratings.
* The dataset contains the information about the users rating, product ID, user ID and so on.
* In this step, we have to load the data with the help of panda’s packages.

**4.2.2: DATA PREPROCESSING:**

* Data pre-processing is the process of removing the unwanted data from the dataset.
* Pre-processing data transformation operations are used to transform the dataset into a structure suitable for machine learning.
* This step also includes cleaning the dataset by removing irrelevant or corrupted data that can affect the accuracy of the dataset, which makes it more efficient.
* Missing data removal
* Encoding Categorical data
* Missing data removal: In this process, the null values such as missing values and Nan values are replaced by 0.
* Missing and duplicate values were removed and data was cleaned of any abnormalities.
* Encoding Categorical data: That categorical data is defined as variables with a finite set of label values.
* That most machine learning algorithms require numerical input and output variables.

**4.2.3: RECOMMENDATION SCORE:**

* The recommendation system combines **the similarity calculated from the score value** with the similarity calculated from the user score probability and the project type to make the similarity between the users more accurate.
* The behaviour that a user gives a score to a commodity is called the user rating behaviour.
* The score value that the user gives to the commodity is called the user rating value.
* The similarity calculation of recommendation algorithm framework based on score preference and the project type.

**4.2.4 DATA SPLITTING:**

* Data splitting is the act of partitioning available data into two portions, usually for cross-validator purposes.
* One Portion of the data is used to develop a predictive model and the other to evaluate the model's performance.
* Separating data into training and testing sets is an important part of evaluating data mining models.
* Typically, when you separate a data set into a training set and testing set, most of the data is used for training, and a smaller portion of the data is used for testing.

**4.2.5 RECOMMENDATION:**

* Recommendation systems are widely used to recommend movies, items, restaurants, places to visit, items to buy, etc.
* In this step, we have to implement the collaborative filtering algorithm.
* **Collaborative filtering** is the type of recommendation algorithm that bases its predictions and recommendations on the rating or behaviour of other users in the system.

**4.2.6: CLASSIFICATION:**

**Machine learning** is a method of data analysis that automates analytical model building. It is a branch of artificial intelligence based on the idea that systems can learn from data, identify patterns and make decisions with minimal human intervention.

* **Decision Tree** Simple to understand and to interpret. Trees can be visualised. Requires little data preparation.
* Other techniques often require data normalisation, dummy variables need to be created and blank values to be removed.
* Note however that this module does not support missing values.

**4.2.5: RESULT GENERATION:**

The Final Result will get generated based on the overall classification and prediction. The performance of this proposed approach is evaluated using some measures like,

* **Accuracy**

Accuracy of classifier refers to the ability of classifier. It predicts the class label correctly and the accuracy of the predictor refers to how well a given predictor can guess the value of predicted attribute for a new data.

AC= (TP+TN)/ (TP+TN+FP+FN)

* **Precision**

Precision is defined as the number of true positives divided by the number of true positives plus the number of false positives.

Precision=TP/ (TP+FP)

* **Recall**

Recall is the number of correct results divided by the number of results that should have been returned. In binary classification, recall is called sensitivity. It can be viewed as the probability that a relevant document is retrieved by the query.

Recall=TP/ (TP+FN)

**CHAPTER 5**

**SYSTEM REQUIREMENTS AND ANALYSIS**

**5.1 SOFTWARE &HARDWARE REQUIREMENTS:**

**Software:**

* O/S : Windows 11.
* Language : Python
* Front End : Streamlit
* Software Used: : Anaconda Navigator – Spyder

**Hardware:**

* System : Pentium IV 2.4 GHz
* Hard Disk : 200 GB
* Mouse : Logitech.
* Keyboard : 110 keys enhanced
* Ram : 4GB

**5.2 SOFTWARE EXPLANATION:**

**5.2.1 Python**

Python is one of those rare languages which can claim to be both *simple* and powerful. You will find yourself pleasantly surprised to see how easy it is to concentrate on the solution to the problem rather than the syntax and structure of the language you are programming in. The official introduction to Python is Python is an easy to learn, powerful programming language. It has efficient high-level data structures and a simple but effective approach to object-oriented programming. Python's elegant syntax and dynamic typing, together with its interpreted nature, make it an ideal language for scripting and rapid application development in many areas on most platforms. I will discuss most of these features in more detail in the next section.

## **5.2.2 Features of Python**

### **Simple**

Python is a simple and minimalistic language. Reading a good Python program feels almost like reading English, although very strict English! This pseudo-code nature of Python is one of its greatest strengths. It allows you to concentrate on the solution to the problem rather than the language itself.

### **Easy to Learn**

As you will see, Python is extremely easy to get started with. Python has an extraordinarily simple syntax, as already mentioned.

### **Free and Open Source**

Python is an example of a FLOSS (Free/Libré and Open Source Software). In simple terms, you can freely distribute copies of this software, read its source code, make changes to it, and use pieces of it in new free programs. FLOSS is based on the concept of a community which shares knowledge. This is one of the reasons why Python is so good - it has been created and is constantly improved by a community who just want to see a better Python.

### **High-level Language**

When you write programs in Python, you never need to bother about the low-level details such as managing the memory used by your program, etc.

### **Portable**

Due to its open-source nature, Python has been ported to (i.e. changed to make it work on) many platforms. All your Python programs can work on any of these platforms without requiring any changes at all if you are careful enough to avoid any system-dependent features.

You can use Python on GNU/Linux, Windows, FreeBSD, Macintosh, Solaris, OS/2, Amiga, AROS, AS/400, BeOS, OS/390, z/OS, Palm OS, QNX, VMS, Psion, Acorn RISC OS, VxWorks, PlayStation, Sharp Zaurus, Windows CE and PocketPC!

You can even use a platform like [Kivy](http://kivy.org) to create games for your computer and for iPhone, iPad, and Android.

### **Interpreted**

This requires a bit of explanation.

A program written in a compiled language like C or C++ is converted from the source language i.e. C or C++ into a language that is spoken by your computer (binary code i.e. 0s and 1s) using a compiler with various flags and options. When you run the program, the linker/loader software copies the program from hard disk to memory and starts running it.

Python, on the other hand, does not need compilation to binary. You just run the program directly from the source code. Internally, Python converts the source code into an intermediate form called bytecodes and then translates this into the native language of your computer and then runs it. All this, actually, makes using Python much easier since you don't have to worry about compiling the program, making sure that the proper libraries are linked and loaded, etc. This also makes your Python programs much more portable, since you can just copy your Python program onto another computer and it just works!

### **Object Oriented**

Python supports procedure-oriented programming as well as object-oriented programming. In procedure-oriented languages, the program is built around procedures or functions which are nothing but reusable pieces of programs. In object-oriented languages, the program is built around objects which combine data and functionality. Python has a very powerful but simplistic way of doing OOP, especially when compared to big languages like C++ or Java.

### **Extensible**

If you need a critical piece of code to run very fast or want to have some piece of algorithm not to be open, you can code that part of your program in C or C++ and then use it from your Python program.

### **Embeddable**

You can embed Python within your C/C++ programs to give scripting capabilities for your program's users.

### **Extensive Libraries**

The Python Standard Library is huge indeed. It can help you do various things involving regular expressions, documentation generation, unit testing, threading, databases, web browsers, CGI, FTP, email, XML, XML-RPC, HTML, WAV files, cryptography, GUI (graphical user interfaces), and other system-dependent stuff. Remember, all this is always available wherever Python is installed. This is called the Batteries Included philosophy of Python.Besides the standard library, there are various other high-quality libraries which you can find at the Python Package Index.

**CHAPTER 6**

**TESTING AND IMPLEMENTATION**

**6.1 INTRODUCTION:**

Testing is a critical phase in the system design lifecycle that ensures the developed system meets its specified requirements and functions correctly under various conditions. It is a systematic process aimed at identifying defects, verifying functionality, and validating that the system performs as intended. The significance of testing cannot be overstated, as it directly impacts the quality, reliability, and user satisfaction of the final product. One of the primary goals of testing is to uncover defects or bugs in the system. Early identification of these issues can prevent potential system failures, reduce the cost of fixing bugs, and ensure a smoother user experience. By systematically identifying and addressing defects, testing helps maintain the integrity and performance of the system. Testing plays a crucial role in ensuring that the system adheres to predefined quality standards. It involves checking various aspects such as functionality, performance, security, and usability. Quality assurance through rigorous testing helps build trust with stakeholders and end-users by delivering a reliable and robust system. Testing serves as a means to validate that the system meets the business requirements and verify that it functions as expected. Validation ensures that the right product is being built, while verification ensures that the product is being built correctly. Together, they form the foundation of a successful system that aligns with user needs and expectations.

**6.2 STRATEGIC APPROACH TO SOFTWARE TESTING:**

**VALIDATION TESTING:**

After the culmination of black box testing, software is completed assembly as a package, interfacing errors have been uncovered and corrected and final series of software validation tests begin validation testing can be defined as many,

But a single definition is that validation succeeds when the software functions in a manner that can be reasonably expected by the customer

**USER ACCEPTANCE TESTING:**

User acceptance of the system is the key factor for the success of the system. The system under consideration is tested for user acceptance by constantly keeping in touch with prospective system at the time of developing changes whenever required.

**OUTPUT TESTING**:

After performing the validation testing, the next step is output asking the user about the format required testing of the proposed system, since no system could be useful if it does not produce the required output in the specific format. The output displayed or generated by the system under consideration. Here the output format is considered in two ways. One is screen and the other is printed format. The output format on the screen is found to be correct as the format was designed in the system phase according to the user needs. For the hard copy also output comes out as the specified requirements by the user. Hence the output testing does not result in any connection in the system.

**6.3 SYSTEM TESTING:**

System testing is the stage of implementation, which aimed at ensuring that system works accurately and efficiently before the live operation commence. Testing is the process of executing a program with the intent of finding an error. A good test case is one that has a high probability of finding an error. A successful test is one that answers a yet undiscovered error.

Testing is vital to the success of the system. System testing makes a logical assumption that if all parts of the system are correct, the goal will be successfully achieved. . A series of tests are performed before the system is ready for the user acceptance testing. Any engineered product can be tested in one of the following ways. Knowing the specified function that a product has been designed to from, test can be conducted to demonstrate each function is fully operational. Knowing the internal working of a product, tests can be conducted to ensure that “al gears mesh”, that is the internal operation of the product performs according to the specification and all internal components have been adequately exercised.

**UNIT TESTING:**

Unit testing is the testing of each module and the integration of the overall system is done. Unit testing becomes verification efforts on the smallest unit of software design in the module. This is also known as ‘module testing’.

The modules of the system are tested separately. This testing is carried out during the programming itself. In this testing step, each model is found to be working satisfactorily as regard to the expected output from the module. There are some validation checks for the fields. For example, the validation check is done for verifying the data given by the user where both format and validity of the data entered is included. It is very easy to find error and debug the system.

**INTEGRATION TESTING:**

Data can be lost across an interface, one module can have an adverse effect on the other sub function, when combined, may not produce the desired major function. Integrated testing is systematic testing that can be done with sample data. The need for the integrated test is to find the overall system performance. There are two types of integration testing. They are:

1. Top-down integration testing.
2. Bottom-up integration testing.

**TESTING TECHNIQUES/STRATEGIES:**

**WHITE BOX TESTING:**

White Box testing is a test case design method that uses the control structure of the procedural design to drive cases. Using the white box testing methods, we Derived test cases that guarantee that all independent paths within a module have been exercised at least once.

* **BLACK BOX TESTING:**

1. Black box testing is done to find incorrect or missing function
2. Interface error
3. Errors in external database access
4. Performance errors.
5. Initialization and termination errors

In ‘functional testing’, is performed to validate an application conforms to its specifications of correctly performs all its required functions. So this testing is also called ‘black box testing’. It tests the external behaviour of the system. Here the engineered product can be tested knowing the specified function that a product has been designed to perform, tests can be conducted to demonstrate that each function is fully operational.

**6.4 IMPLEMENTATION:**

Implementation of software refers to the final installation of the package in its real environment, to the satisfaction of the intended users and the operation of the system. The people are not sure that the software is meant to make their job easier.

• The active user must be aware of the benefits of using the system

• Their confidence in the software built up

• Proper guidance is impaired to the user so that he is comfortable in using the application

Before going ahead and viewing the system, the user must know that for viewing the result, the server program should be running in the server. If the server object is not running on the server, the actual processes will not take place.

**User Training**

To achieve the objectives and benefits expected from the proposed system it is essential for the people who will be involved to be confident of their role in the new system. As system becomes more complex, the need for education and training is more and more important. Education is complementary to training. It brings life to formal training by explaining the background to the resources for them. Education involves creating the right atmosphere and motivating user staff. Education information can make training more interesting and more understandable.

**Training on the Application Software**

After providing the necessary basic training on the computer awareness, the users will have to be trained on the new application software. This will give the underlying philosophy of the use of the new system such as the screen flow, screen design, type of help on the screen, type of errors while entering the data, the corresponding validation check at each entry and the ways to correct the data entered. This training may be different across different user groups and across different levels of hierarchy.

**Operational Documentation**

Once the implementation plan is decided, it is essential that the user of the system is made familiar and comfortable with the environment. A documentation providing the whole operations of the system is being developed. Useful tips and guidance is given inside the application itself to the user. The system is developed user friendly so that the user can work the system from the tips given in the application itself.

**CHAPTER 7**

**CONCLUSION**

In conclusion, our proposed system utilizes collaborative filtering to recommend products to users based on their input and preferences. By leveraging the opinions and preferences of similar users in the community, we can provide personalized recommendations that are tailored to each individual's tastes and preferences. Additionally, we employ machine learning classification algorithms, such as Decision Trees, to further enhance the recommendation process. Our experimental results demonstrate the effectiveness of our approach, showing high accuracy, precision, recall, and F1-score metrics. This indicates that our system successfully recommends relevant products to users, improving user satisfaction and engagement

**CHAPTER 8**

**FUTURE ENHANCEMENT**

As a future work, Explore advanced collaborative filtering techniques, such as matrix factorization or deep learning-based approaches, to improve recommendation accuracy and scalability. Incorporate additional user data, such as demographic information, purchase history, or browsing behavior, to enhance the personalization of recommendations

**CHAPTER 9**

**APPENDIX**

**9.1 SOURCE CODE:**

# IMPORT PACKAGES

import pandas as pd

import matplotlib.pyplot as plt

import seaborn as sns

from sklearn.model\_selection import train\_test\_split

from sklearn import preprocessing

import warnings

warnings.filterwarnings("ignore")

import streamlit as st

import base64

# --------------- BACKGROUND IMAGE

st.markdown(f'<h1 style="color:#FFFFFF;text-align: center;font-size:36px;">{"Fashion Recommendation"}</h1>', unsafe\_allow\_html=True)

def add\_bg\_from\_local(image\_file):

with open(image\_file, "rb") as image\_file:

encoded\_string = base64.b64encode(image\_file.read())

st.markdown(

f"""

<style>

.stApp {{

background-image: url(data:image/{"png"};base64,{encoded\_string.decode()});

background-size: cover

}}

</style>

""",

unsafe\_allow\_html=True

)

add\_bg\_from\_local('1.jpg')

#==================== 1.READ A INPUT DATA ===============

data=pd.read\_csv('final\_test.csv')

print("---------------------------------------")

print(" Data Selection ")

print("---------------------------------------")

print()

print(data.head(10))

print()

#================ 2.DATA PREPROCESSING ===========

#=== CHECK MISSING VALUES ===

print("---------------------------------------")

print(" Before Handling missing values ")

print("---------------------------------------")

print()

print(data.isnull().sum())

print()

print("---------------------------------------")

print(" After Handling missing values ")

print("---------------------------------------")

print()

data=data.fillna(0)

print(data.isnull().sum())

print()

#=== LABEL ENCODING ===

print("----------------------------")

print("BEFORE LABEL ENCODING")

print("----------------------------")

print()

print(data['size'].head(15))

print()

label\_size = data['size']

label\_encoder=preprocessing.LabelEncoder()

print("----------------------------")

print("AFTER LABEL ENCODING")

print("----------------------------")

print()

data['size']=label\_encoder.fit\_transform(data['size'])

print(data['size'].head(15))

print()

#================= 3. DATA SPLITTING ===============

print("---------------------------------------")

print(" Data Splitting ")

print("---------------------------------------")

print()

X=data.drop(['size'],axis=1)

Y=data['size']

X\_train, X\_test, Y\_train, Y\_test = train\_test\_split(X,Y,test\_size=0.30,random\_state=50)

print("Total no of data :",len(X))

print("Total no of Train data :",len(X\_train))

print("Total no of Test data :",len(X\_test))

#================== 4. CLASSIFICATION ================

recommend\_data1 = pd.read\_csv("styles.csv", on\_bad\_lines='skip')

recommend\_data1 = recommend\_data1[0:1000

recommend\_data = recommend\_data1[['id','gender','baseColour']]

recommend\_data.to\_csv("Finaldata.csv")

# ---- MISSING VLUES

print(recommend\_data.isnull().sum())

# ---- LABELLING

label\_gen = recommend\_data1['gender'].unique()

label\_clr = recommend\_data1['baseColour'].unique()

label\_cat = recommend\_data1['masterCategory'].unique()

label\_subcat = recommend\_data1['subCategory'].unique()

label\_article = recommend\_data1['articleType'].unique()

label\_season = recommend\_data1['season'].unique()

label\_usage = recommend\_data1['usage'].unique()

label\_prd = recommend\_data1['productDisplayName'].unique()

# ----- CONVERT STR O INT

label\_encoder=preprocessing.LabelEncoder()

recommend\_data1['gender1']=label\_encoder.fit\_transform(recommend\_data1['gender'])

recommend\_data1['baseColour1']=label\_encoder.fit\_transform(recommend\_data1['baseColour'])

recommend\_data1['masterCategory1']=label\_encoder.fit\_transform(recommend\_data1['masterCategory'])

recommend\_data1['subCategory1']=label\_encoder.fit\_transform(recommend\_data1['subCategory'])

recommend\_data1['articleType1']=label\_encoder.fit\_transform(recommend\_data1['articleType'])

recommend\_data1['usage1']=label\_encoder.fit\_transform(recommend\_data1['usage'])

recommend\_data1['productDisplayName1']=label\_encoder.fit\_transform(recommend\_data1['productDisplayName'])

recommend\_data1['season1']=label\_encoder.fit\_transform(recommend\_data1['season'])

# ------ SPLIT DATA

X1=recommend\_data1.drop(['id','gender','baseColour','masterCategory','subCategory','articleType','usage','productDisplayName','season'],axis=1)

Y1=recommend\_data1['id']

X\_train1, X\_test1, Y\_train1, Y\_test1 = train\_test\_split(X1,Y1,test\_size=0.30,random\_state=50)

print("Total no of data :",len(X1))

print("Total no of Train data :",len(X\_train1))

print("Total no of Test data :",len(X\_test1))

# ----- DECISION TREE ---

from sklearn.tree import DecisionTreeClassifier

dt = DecisionTreeClassifier()

dt.fit(X\_train1,Y\_train1)

pred\_dt = dt.predict(X\_train1)

from sklearn import metrics

acc\_dt = metrics.accuracy\_score(Y\_train1,pred\_dt) \* 100

# ----- USERS INPUT

age = st.text\_input("Enter Age ",0)

age = int(age)

weight = st.text\_input("Enter Weight",0)

weight = int(weight)

height = st.text\_input("Enter Height",0)

height = int(height)

size = st.selectbox("Choose Size", label\_size)

gender = st.selectbox("Choose Gender", label\_gen)

cat = st.selectbox("Choose Category", label\_cat)

subcat = st.selectbox("Choose Sub Category", label\_subcat)

article = st.selectbox("Choose articleType", label\_article)

clr = st.selectbox("Choose Colour", label\_clr)

season = st.selectbox("Choose Season", label\_season)

year = st.text\_input("Enter Year",2000)

year = int(year)

usage = st.selectbox("Choose Usage", label\_usage)

pdt = st.selectbox("Choose Prduct", label\_prd)

# -----

a1 = recommend\_data1[recommend\_data1['gender'] == gender]['gender1']

try:

a1 = max(a1)

except:

a1=a1

a2 = recommend\_data1[recommend\_data1['masterCategory'] == cat]['masterCategory1']

try:

a2 = max(a2)

except:

a2=a2

a3 = recommend\_data1[recommend\_data1['subCategory'] == subcat]['subCategory1']

try:

a3 = max(a3)

except:

a3=a3

a4 = recommend\_data1[recommend\_data1['articleType'] == article]['articleType1']

try:

a4 = max(a4)

except:

a4=a4

a5 = recommend\_data1[recommend\_data1['baseColour'] == clr]['baseColour1']

try:

a5 = max(a5)

except:

a5=a5

a6 = recommend\_data1[recommend\_data1['season'] == season]['season1']

try:

a6 = max(a6)

except:

a6=a6

a7 = year

a8 = recommend\_data1[recommend\_data1['usage'] == usage]['usage1']

try:

a8 = max(a8)

except:

a8=a8

a = pdt

a9 = recommend\_data1[recommend\_data1['productDisplayName'] == pdt]['productDisplayName1']

try:

a9 = max(a9)

except:

a9=a9

Data\_reg = [a1,a2,a3,a4,a5,a6,a7,a8,a9]

st.text(Data\_reg)

y\_pred\_reg=dt.predict([Data\_reg])

iddd = y\_pred\_reg[0]

# st.text(y\_pred\_reg)

# [2, 1, 27, 59, 21, 0, 2000, 0, 846]

but = st.button("Recommend")

if but:

pred\_data = pd.read\_csv("Finaldata.csv")

fin\_img = "output/" + str(iddd)+".jpg"

import matplotlib.image as mpimg

img = mpimg.imread(fin\_img)

# st.image(img)

# st.text(a)

# a9 = recommend\_data1[recommend\_data1['productDisplayName'] == pdt]['productDisplayName1']

data\_label=recommend\_data1['productDisplayName']

x1=data\_label

for i in range(0,len(data\_label)):

if x1[i]==a:

idx=i

data\_frame1\_c=recommend\_data1['id']

Req\_data\_c=data\_frame1\_c[idx]

fin\_img = "output/" + str(Req\_data\_c)+".jpg"

import matplotlib.image as mpimg

img = mpimg.imread(fin\_img)

plt.imshow(img)

st.image(img)

9.2 SCHREEN SHOTS

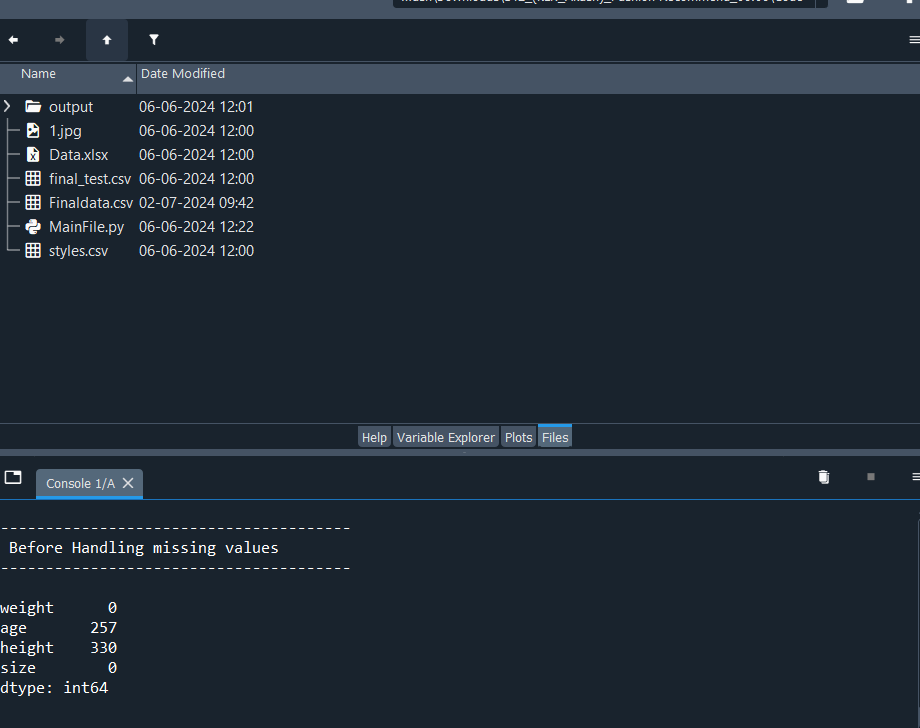
Home Page

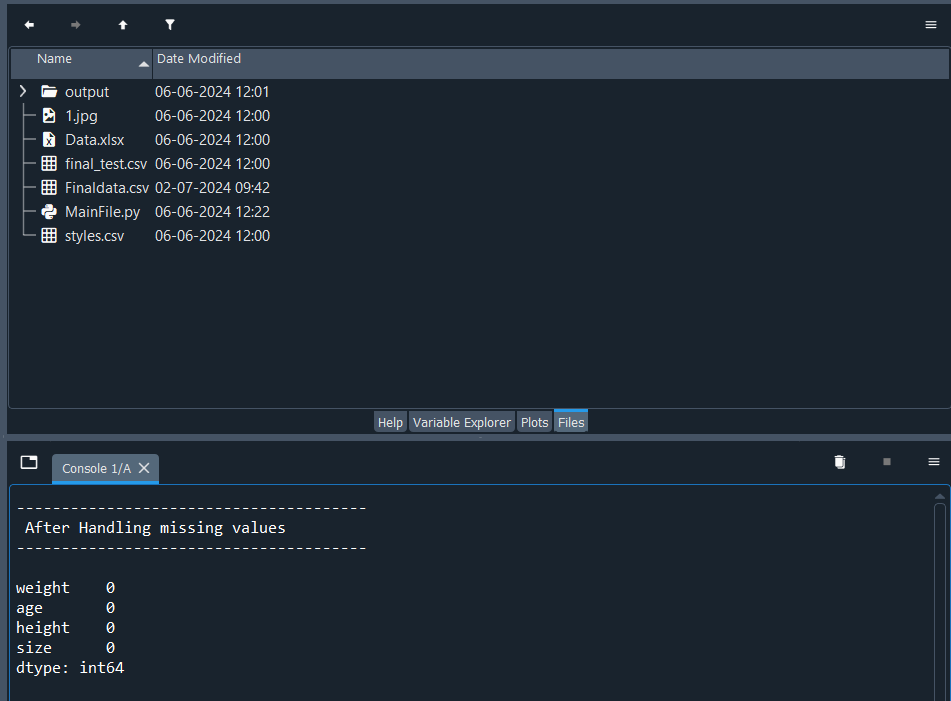


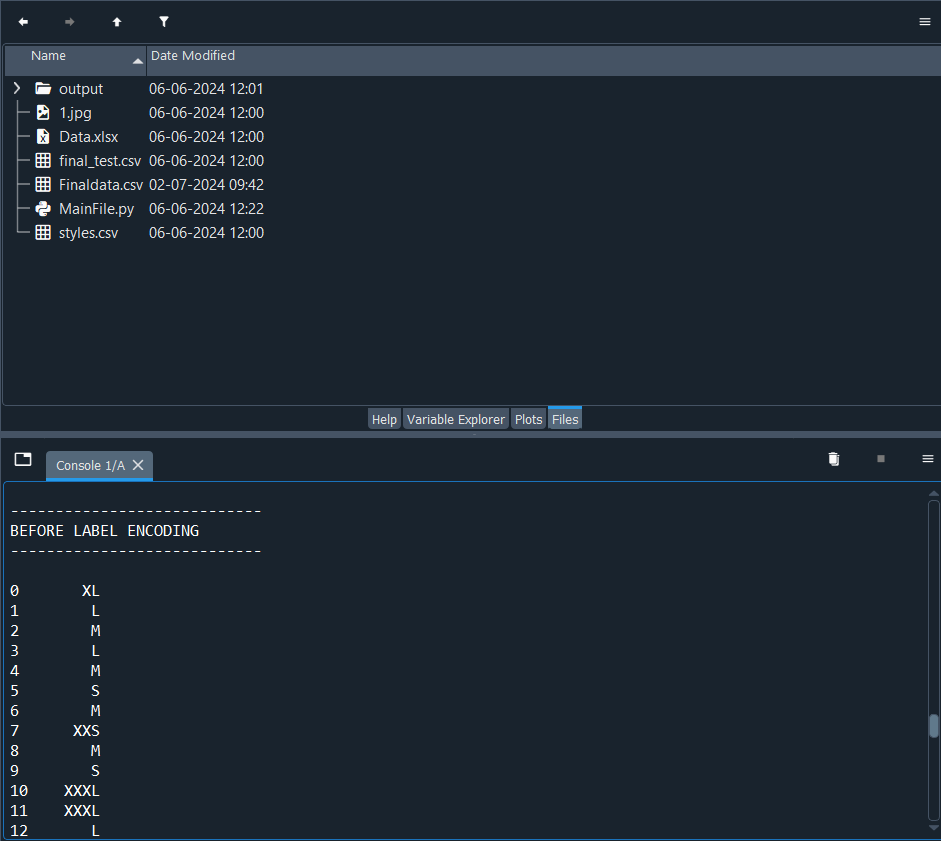
Values Insertion In Home Page



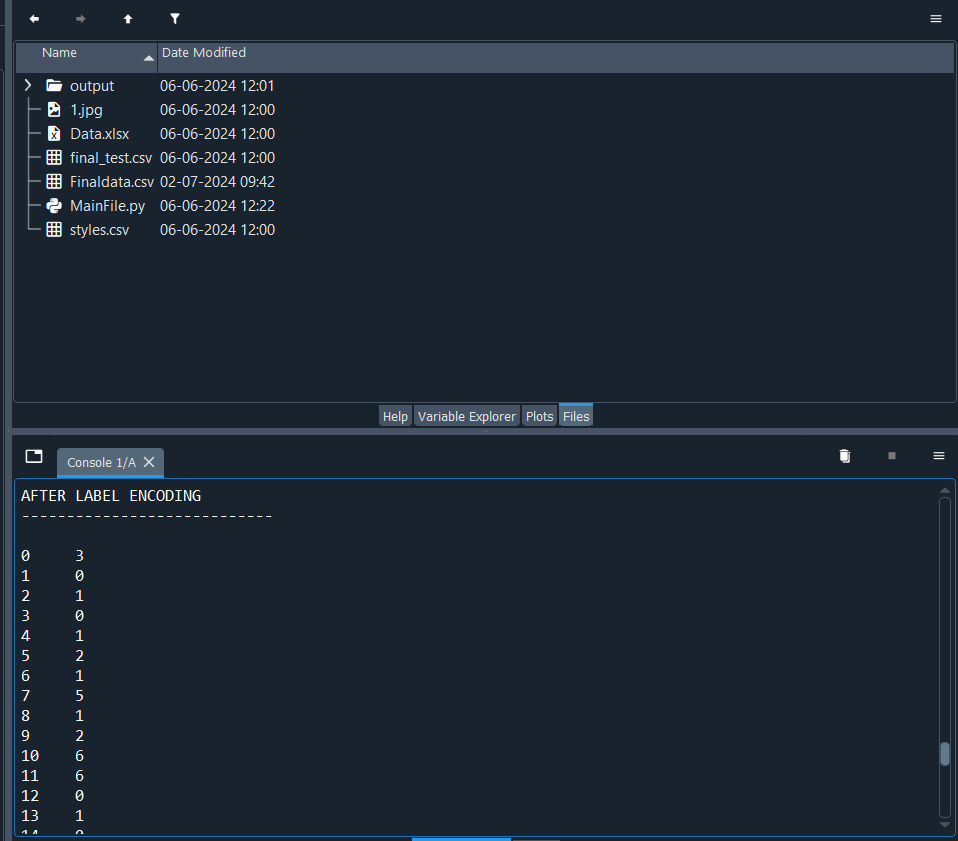
Before handling Missing Data



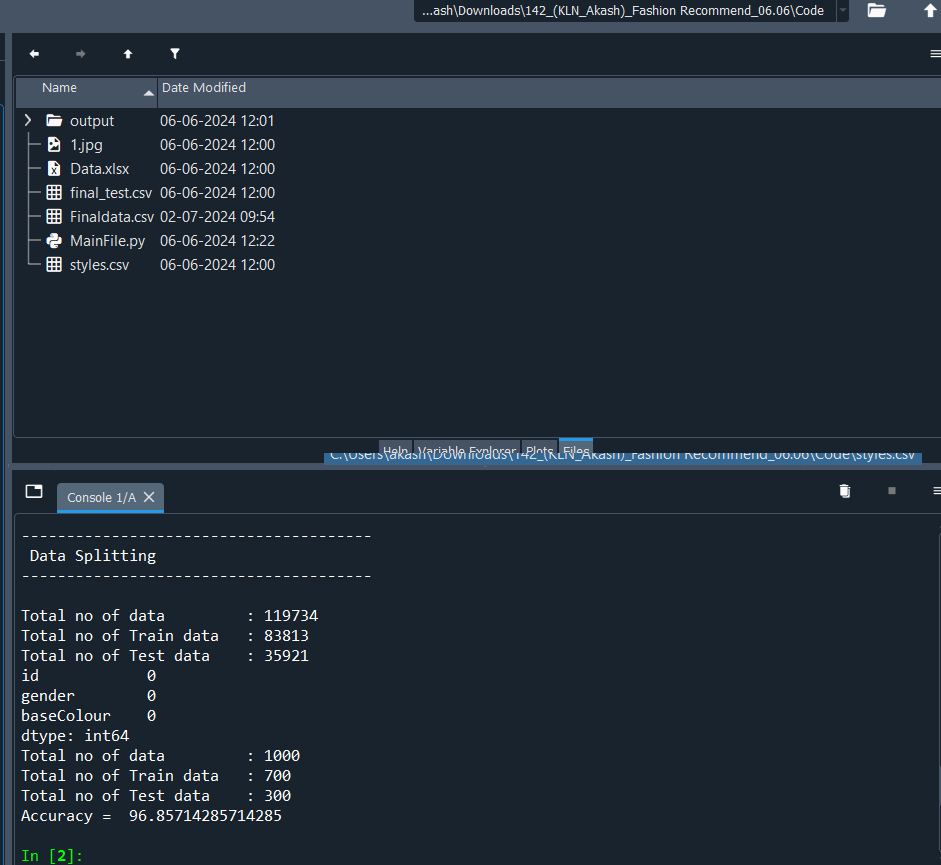
After Handling Missing Data 

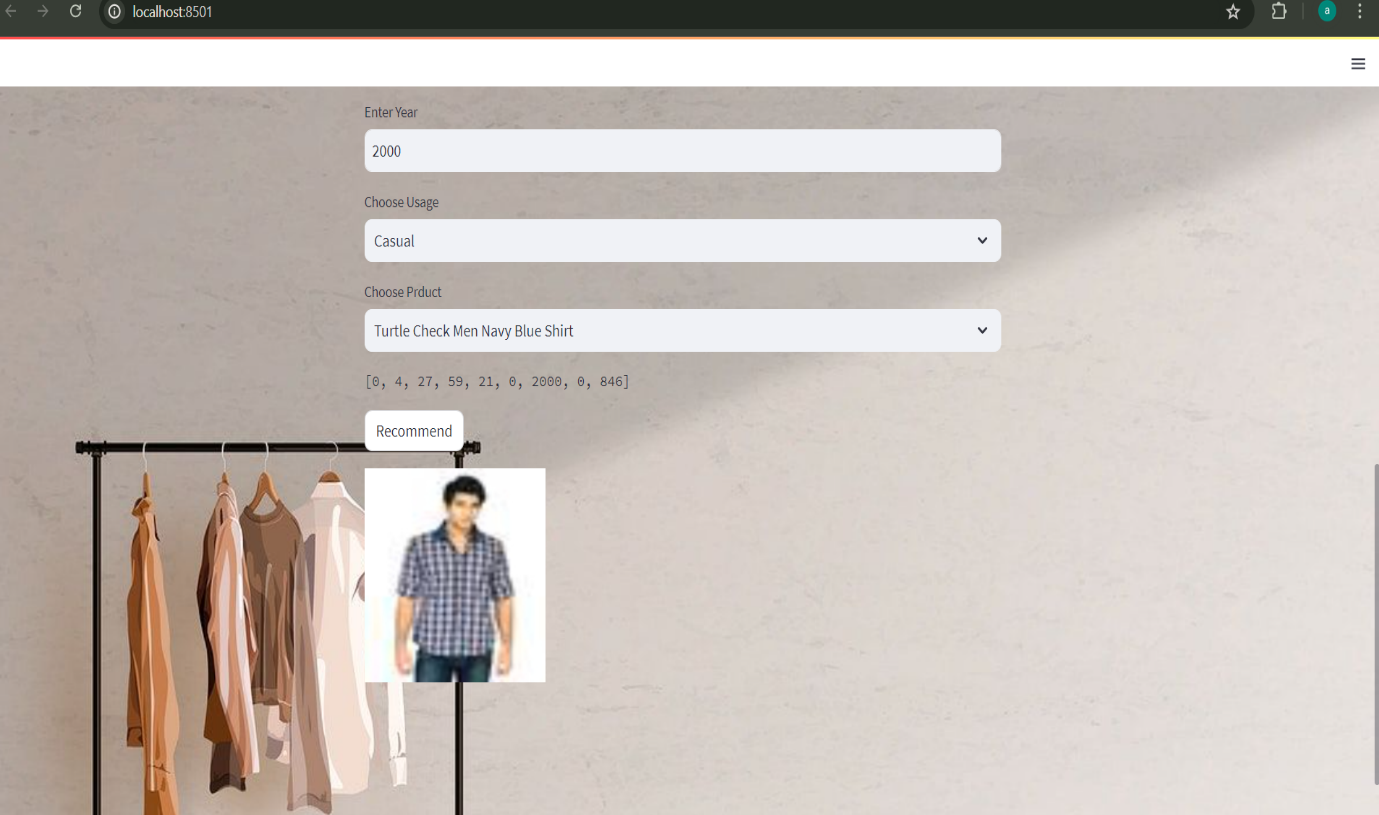
Before Label Encoding

After Label Encoding



Data Splitting

****

**Output**

**CHAPTER 10**

**BIBLIOGRAPHY**

1.Bobadilla, J., Ortega, F., Hernando, A., & Gutierrez, A. (2013). Recommender systems survey. Knowledge-Based Systems, 46, 109-132.

1. Koren, Y., Bell, R., & Volinsky, C. (2009). Matrix factorization techniques for recommender systems. Computer, 42(8), 30-37.
2. Aggarwal, C. C. (2016). Recommender Systems: The Textbook. Springer.
3. He, X., Liao, L., Zhang, H., Nie, L., Hu, X., & Chua, T. S. (2017). Neural collaborative filtering. In Proceedings of the 26th International Conference on World Wide Web (pp. 173-182).
4. Ricci, F., Rokach, L., & Shapira, B. (2011). Introduction to recommender systems handbook. In Recommender Systems Handbook (pp. 1-35). Springer.
5. Chen, L., & Chen, G. (2015). Personalized fashion recommendation on the outfit level. IEEE Transactions on Multimedia, 18(6), 1-14.
6. Zhang, S., Yao, L., Sun, A., & Tay, Y. (2019). Deep learning-based recommender system: A survey and new perspectives. ACM Computing Surveys (CSUR), 52(1), 1-38.
7. Hu, R., & Pu, P. (2011). Enhancing collaborative filtering systems with personality information. In Proceedings of the fifth ACM conference on Recommender systems (pp. 197-204).
8. McAuley, J., Pandey, R., & Leskovec, J. (2015). Inferring networks of substitutable and complementary products. In Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (pp. 785-794).
9. Sedhain, S., Menon, A. K., Sanner, S., & Xie, L. (2015). Autorec: Autoencoders meet collaborative filtering. In Proceedings of the 24th International Conference on World Wide Web (pp. 111-112).
10. Johnson, R., & Zhang, T. (2014). Effective use of word order for text categorization with convolutional neural networks. In Proceedings of the 2014 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (pp. 103-112).
11. Hinton, G. E., & Salakhutdinov, R. R. (2006). Reducing the dimensionality of data with neural networks. Science, 313(5786), 504-507.
12. Karatzoglou, A., Smirnova, E., & Amatriain, X. (2010). Collaborative music recommendation with context-aware information. In Proceedings of the fourth ACM conference on Recommender systems (pp. 195-202).
13. Sun, M., & Zhang, X. (2018). Conversational recommender system. In The 41st International ACM SIGIR Conference on Research & Development in Information Retrieval (pp. 235-244).
14. LeCun, Y., Bengio, Y., & Hinton, G. (2015). Deep learning. Nature, 521(7553), 436-444.