MIT | Academy of Engineering

(An Autonomous Institute Affiliated to Savitribai Phule Pune University)

LAPTOP RECOMMENDATION SYSTEM

SY B.Tech. Minor Project Report

SUBMITTED BY

Sairaj Kalyan Dalavi	**[S214031]**
Swapanil Mattoo	**[S214028]**
Viraj Fale	**[S214020]**

GUIDED BY

Prof. Minakshi Vharkate

SCHOOL OF COMPUTER ENGINEERING AND TECHNOLOGY

MIT ACADEMY OF ENGINEERING, ALANDI (D), PUNE-412105

MAHARASHTRA (INDIA)

MAY, 2022

MIT | Academy of Engineering

(An Autonomous Institute Affiliated to Savitribai Phule Pune University)

LAPTOP RECOMMENDATION SYSTEM

SY B.Tech. Minor Project Report

submitted in partial fulfilment of the
requirements for the award of the degree

of

Bachelor of Technology

in

COMPUTER ENGINEERING

BY

Sairaj Dalavi , Swapanil Mattoo, Viraj Fale

SCHOOL OF COMPUETR ENGINEERING & TECHNOLOGY

MIT ACADEMY OF ENGINEERING, ALANDI (D), PUNE-412105

MAHARASHTRA (INDIA)

MAY, 2022

MIT | Academy of Engineering

(An Autonomous Institute Affiliated to Savitribai Phule Pune University)

CERTIFICATE

It is hereby certified that the work which is being presented in the SY B.Tech. Minor Project Report entitled “ Laptop Recommendation System ”, in partial fulfillment of the requirements for the award of the **Bachelor of Technology in Computer Engineering** and submitted to the **School of Computer Engineering and Technology of MIT Academy of Engineering, Alandi(D), Pune, Affiliated to Savitribai Phule Pune University (SPPU), Pune** is an authentic record of work carried out during an Academic Year 2020-2021, under the supervision of **Mrs. Minakshi Vharkate, School of Computer Engineering and Technology.**

Sairaj Kalyan Dalavi	**PRN No. 0120200141**	**Exam Seat No. S214031**
Swapanil Mattoo	**PRN No. 0120200114**	**Exam Seat No. S214028**
Viraj Fale	**PRN No. 0120200089**	**Exam Seat No. S214020**

Date: May 2022

Signature of Project Advisor

Project Adviser

School of Computer Engineering and Technology,
MIT Academy of Engineering, Alandi(D), Pune

Signature of Dean

Dean

School of Computer Engineering and Technology,
MIT Academy of Engineering, Alandi(D), Pune

(STAMP/SEAL)

Signature of Internal examiner/s
Name………………………………
Affiliation…………………………

Signature of External examiner/s
Name………………………………
Affiliation…………………………

ACKNOWLEDGEMENT

We want to express our gratitude towards our respected project guide Prof. Minakshi Vharkate forher constant encouragement and valuable guidance during the completion of this project work. We also want to express our gratitude towards respected School Dean Mrs. Ranjana Badre for her continuous encouragement.

We would be failing in our duty if we do not thank all the other staff and faculty members for their experienced advice and evergreen co-operation

1. **Sairaj Dalavi**
2. **Swapanil Mattoo**
3. **Viraj Fale**

LIST OF FIGURES

Fig No.	Fig. Name	Page No.
Fig3.1	Block Diagram	16
Fig3.2	Architecture Diagram	17
Fig3.3	Use CaseDiagram	19
Fig3.4	ActivityDiagram	20
Fig 4.1	Home Page Of System	22
Fig 4.2	Project Code	23
Fig 4.3	Project Code	23
Fig 4.4	Specification Page	24
Fig 4.5	Final Output of the project	25

CONTENTS

Acknowledgements			i
Abstract			ii
List of Figures			iii
List of Tables			iv
1.	Introduction		1
	1.1	Motivation for the project	
	1.2	Problem Statement	
	1.3	Objectives and Scope	
2.	Literature Survey		4
3.	System Design		7
	3.1	Block diagram	
	3.2	Arcitecture Diagram	
	3.3	Use Case Diagram	
	3.4	Activity Diagram	
	3.5	Hardware and Software Requirements	
	3.6	Methodology	
4.	Implementation and Results		22
	4.1 Algorithm and flowcharts		
	4.2 Results		

5.	Conclusion and Future scope	25
References		26
Appendix		

ABSTRACT

The ever increasing number of E-commerce sites on the Internet has brought about information overload. This has made it difficult for consumers of certain products to find information about such products in an attempt to purchase products that best satisfies them. It has equally reduced the volume of product sales in the E-commerce domain.

Hence, we propose a laptop recommendation system. The proposal system intelligently mines information about the features of laptop and provides professional services to potential buyers by recommending optimal products based on their personal needs. This will help the customer to get the best laptop according to his needs and the customer would be independent select as he need not to be depended on the shopkeeper or the salesman.

Be it a teacher , who wants to use a laptop for conducting classes, webinars, or other day to day activities or a student , who besides using the product for studying, demands high performance for gaming to run smoothly. Choosing a product which perfectly suits the needs of the buyer can become a very long and confusing task for the customer. As a matter of fact the World we are living in, people are more aware about investing their money in the ”perfect product” that they search for. However, sometimes it gets really tough without the prior knowledge regarding the products and can lead to being misguided.

1. INTRODUCTION

The incessant growth of the Web has led to rapid expansion of e-commerce among other things. The large amount of product information on the Web poses great challenges to both customers and online business. More customers are turning towards online shopping because it is relatively convenient , reliable and fast. Online business have often been overwhelmed by the rich data they have collected and find it difficult to promote products appropriate to specific customers. There is also the problem of ineffective utilization of the available large amount of product information from online transactions to support better decision making by both buyers and sellers. To address these information overload problems ,e-commerce stores are now applying mass customization principles not to the products but to their presentation in the online store. One way to achieve mass customization in e- commerce is the use of recommender system.

Personalized recommendation systems enable consumers to easily access information about products they are interested in, and save time of reading through electronic documents. Moreover, enterprises can get to know customers buying behaviors better, and develop efficient marketing strategies to attract different customers. Customer's satisfaction, and loyalty can thus be increased. A good personalized recommendation system should be able improve user satisfaction , a key attribute to customer loyalty and continued use.

1.1 Motivations

Nowadays, people used to buy products online more than from stores. Previously people used to buy products based on reviews given by relatives or friends but now as the options have increased and we can buy anything digitally we need to assure people that the product is good and they will like it. To give confidence in buying the products we are building a recommendation system.

1.2 Problem Statement

Implementing an interface to provide suggestions and recommendations regarding various choices of laptops to people.

1.3 Objectives and Scope

- To create a personalized recommendation system which would easily access information about laptops.
- Save time reading through electronic documents.
- Achieving a simple user interface.
- To make decisions easy for people.

2. LITERATURE SURVEY.

In [1] the paper ' Fuzzy logic based personalized recommendation system ' published in 'International journal of computer science and Information Technology and Security.2012 ', the authors Ojokho, B.A.Samule, Omisore, Ogunniyi speak about the Recommender system and fuzzy logic system also known as the nonlinear mapping of an input data set to a scalar output data set is used. Fuzzy near compactness concept is employed to measure the similarity between consumer needs and product features in order to recommend optimal products to potential buyers. Fuzzy sets have attracted growing attention growing attention in modern information technology and data analysis.

In [2] the paper ‘ Content Recommendation System Based on Private Dynamic User profile.’ Published in ‘ IEEE , Proceedings of Sixth International Conference of machine learning and cybernetics , Hongkong August 2007’ the authors ‘ Ting Chen, Wei-Li Han, Hai-Dong Wang , Bin Xu ’ speak about the Agent based personalized Recommendation method called Content based Recommendation Algorithm based on private Dynamic user profile. Here the content based recommendation algorithm collects the private data of the user such as schedules, favorite websites and personal emails which creates a big issue regarding the privacy of the user. Though we get good and helpful recommendations but the privacy is breached.

In [3] the paper ‘Ontology based Conversational Recommender System for recommending laptop’ Published in “,International Conference on Data & Information Science : Journal of Physics. 1192.2019.” the authors ‘M.S.Ayundhita , Z.K.A.Baizal , Y Siboconi’ speak about the Conversational recommender system (CRS) , Interaction generate method and Ontology. Here the CRS interacts with the iterative conversation with users to find out customer needs and to provide the most accurate recommendation. In this Conversational Recommendation system, the system interacts or asks questions which acts the same like a professional seller. The user involved in this test shows that a recommendation system that prioritizes functional requirements is more helpful in the product selection than a recommendaer system commonly used in e- commerce.

In [4] the paper ‘A Survey of Recommendation System : Research challenges’ published in ‘International Journal of Engineering Trends & Technology. Volume 4 Issue5 – May 2013’ the authors ‘Lalita Sharma, Anju Gera’ speak about Recommendation System techniques used – 1) Collaboratice filtering process. 2) Conteent Based process. 3) hybrid process. Here the author has penned down the problems which are challenging for the present recommendation system with respect to recommendation quality and privacy aspects.

In [5] the paper ‘Gadget Recommendation System using Data Science’ published in ‘IEEE, Proceedings of Third International conference on Intelligent Sustainable Sytems [ICISS 2020]’ the authors ‘A Pushpalatha, Harish Sanmugam , J Jeya , Pradeep K , Madhu Bala S.’ speak about the AI based recommendation system, K-means algorithm. K means the algorithm is specially tuned to match the user input using a wide range of extracted data which was pre-processed.

In [6] the paper ‘Implementation of Topsis Method In Web Based System Recommendations For Students Laptop Selection’ published in ‘TIJNMT. Vol. IV. No.1 | June 2017’, the authors Adhi Kusnadi and Edwin Kumiawan speak about the TOPSIS - This method is widely used to solve decision making because the concept is simple, easy to understand, computationally efficient, and has the ability to measure the relative performance of decision alternatives. In making a recommendation system, using the TOPSIS method is the recommended one because the concept is simple, easy to understand, efficient, and has the ability to measure the relative performance of alternative decisions. According to the implementation, the recommendation system with the TOPSIS method has a 70% accuracy rate. The system accuracy rate of 70% of the data that has been tested and compared to manual calculation. And the satisfaction level of respondents on the system recommendation with an average rate value is enough and good.

In [7] the paper ‘Opinion Observer: Recommendation System on E Commerce Website’ published in ‘International Journal of Computer Applications (0975 - 8887) Volume 105 - No.14, November 2014’, the authors Mohammad Daoud, S.K Naqvi, and Asad Ahmad speak about the Demographic Technique, Sorting Algorithm - Demographic recommenders systems intend to categorize the user based on personal attributes and make recommendations based on demographic classes. They easily refined the user query and found the result. In this paper, we used a text mining approach to mine product features, opinions and their semantic similarity from Web opinion sources. The consumer can clearly see the strengths and Weaknesses of each product in the minds of existing consumer's opinion.

In [8] the paper ‘A proactive personalised mobile recommendation system using analytic hierarchy process and Bayesian network’ published in ‘Springer 20 July 2012’, the authors Kam Fung Yeung, Yanyan Yang, and David Ndzi speak about the Bayesian Network Algorithm, Analytic Hierarchy Process – A Bayesian network is a probabilistic graphical model that combines the advantage of CF and CBF. AHP-MCR approach provides the flexibility to add/remove contextual criteria in different scenarios and domains. An analytic hierarchy process based multi-criteria ranking (AHP-MCR) approach has been developed and used to rate recommendations in a variety of domains. Additionally, a Bayesian network algorithm is applied to solve the cold-start problem in recommendation systems. The weights of the various contexts (criteria) are automatically adjusted using individual-based and/or group-based (group decision making) assignments. Additionally, a Bayesian Network algorithm has been applied to solve the cold-start problem inherent in recommendation systems.

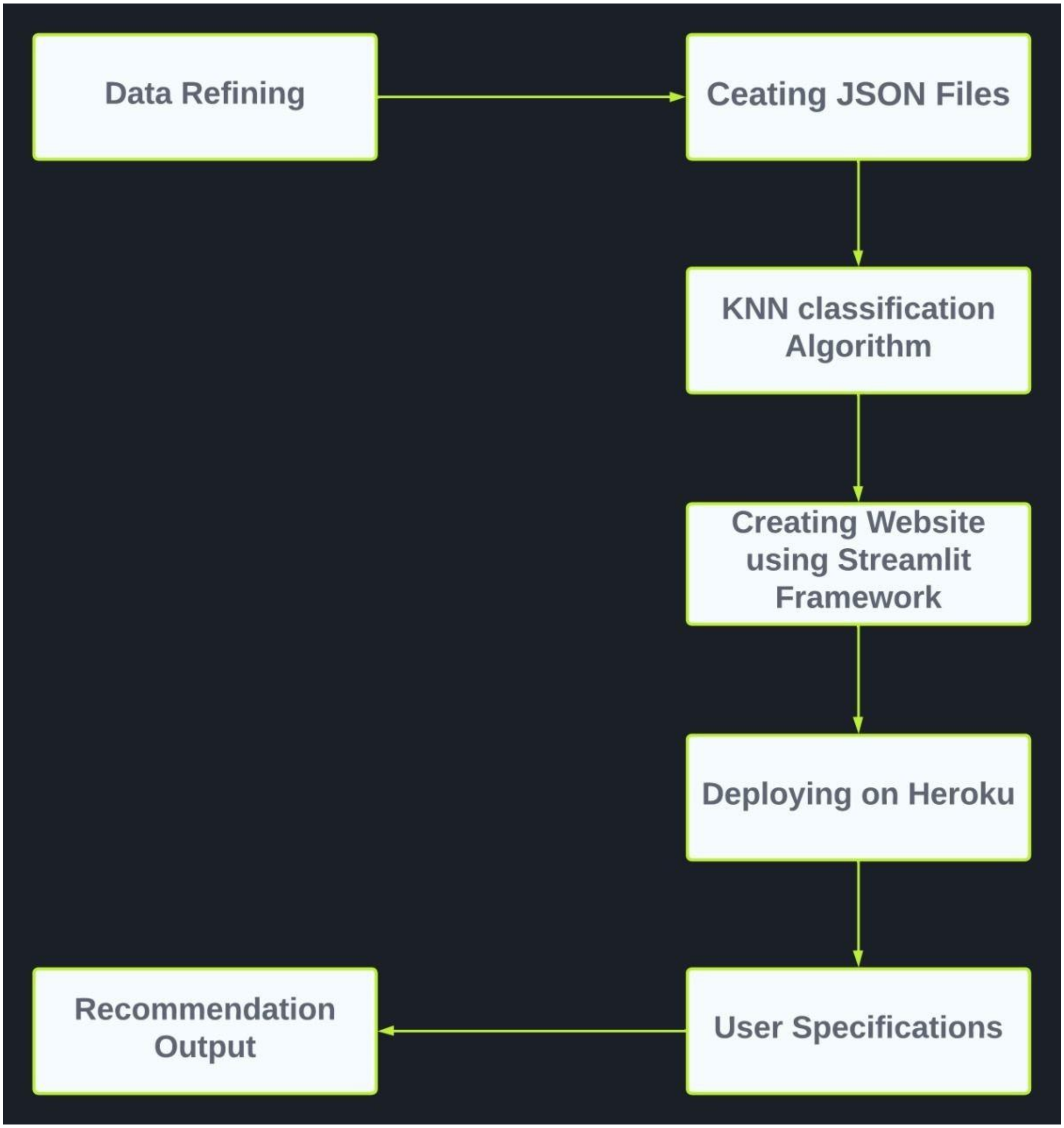
In [9] the paper ‘Contextual Sentiment Based Recommender System to Provide Recommendation in the Electronic Products Domain’ published in ‘International Journal of Machine Learning and Computing, Vol. 9, No. 4, August 2019’ the authors N.A. Osman, S A. M. Noah, and M. Darwich speak about Collaborative Filtering, Sentiment Analysis - which filters information by exploiting the recommendation of other similar users. This technique recommend items to a user based on similarities between the past behavior of the user and that of like minded people sentiment analysis is used to determine the words or sentences that have sentiment value.

We present a contextual information I sentiment based model for the recommender system by making use of user comments and preferences to provide a recommendation. The purpose of this approach is to avoid term ambiguity which is a so -called domain sensitivity problem in recommendation.

In [10] the paper ‘Decision Support System for Laptop Selection Using AHP Method and Profile Matching’ published in ‘IJCCS (Indonesian Journal of Computing and Cybernetics Systems) Vol. 15, No.3, July 2021, pp. 307~316’, the authors Muhammad I Mukharir and Retantyo Wardoyo speak about the AHP Method, PM Method, SAW Method - HP is a multi-criteria decision-making technique in which decision makers set priorities and determine decisions by making pairwise comparisons I between criteria to obtain priorities in each hierarchy. I PM method is used to assess criteria that are I close to the ideal value desired by decision makers. To calculate the final score recommendation of a laptop product I used for the SAW method. To calculate the final score recommendation of a laptop product used for the SAW method. The input values for the SAW method are the value of the weighted criteria for the weighted calculation of the AHP method, the value of the calculation results of the PM method, and the value for calculating the cost and benefit criteria using linear interpolation. The output of the SAW method is the recommendation score of a laptop product.

3. SYSTEM DESIGN

3.1 Block diagram



3.1 Block Diagram

3.2 Architecture Diagram

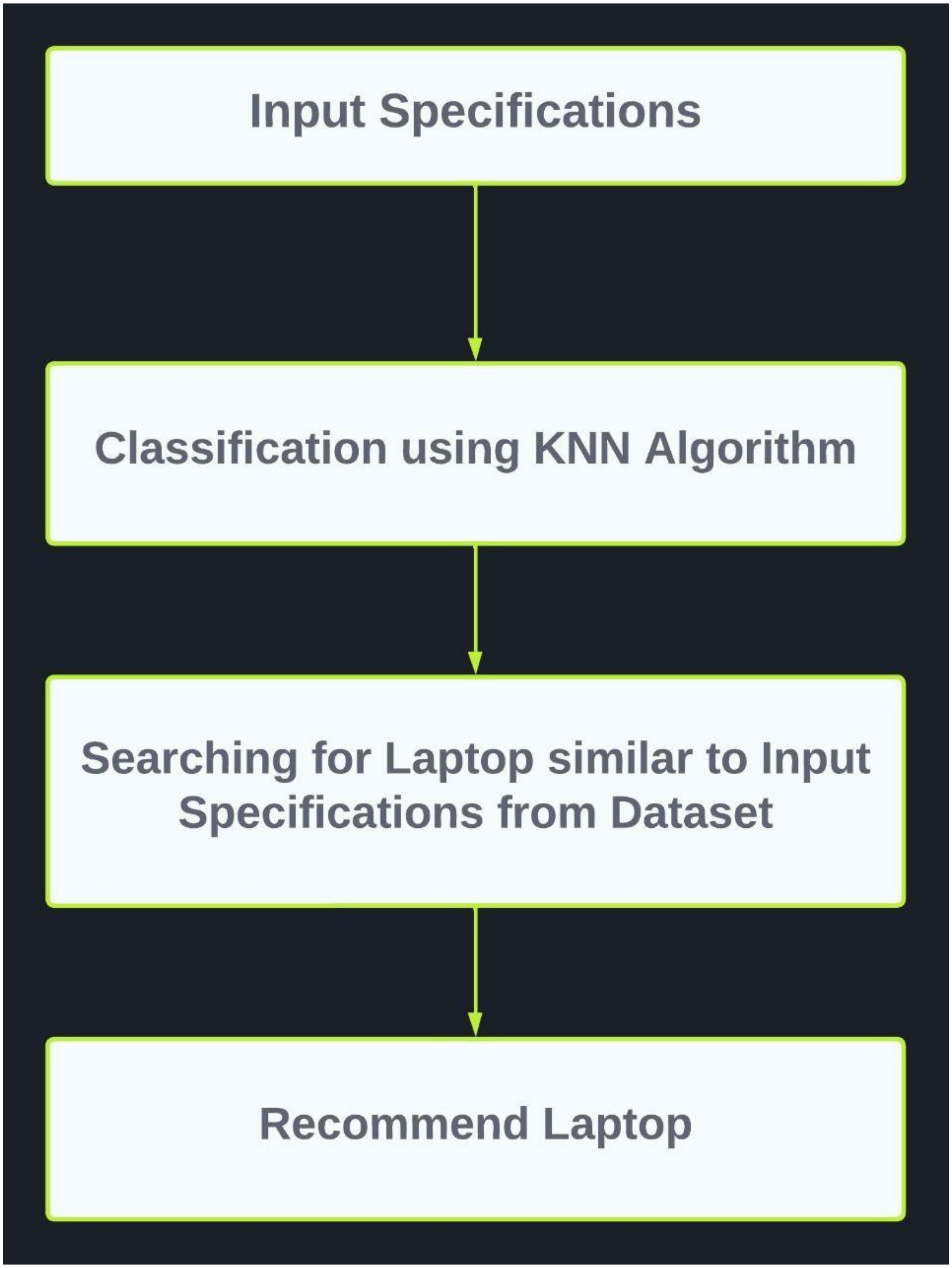

Input Specifications
Classification using KNN Algorithm
Searching for Laptop similar to Input Specifications from Dataset
Recommend Laptop

Figure 3.2: Architecture Diagram

3.3 Use Case Diagram

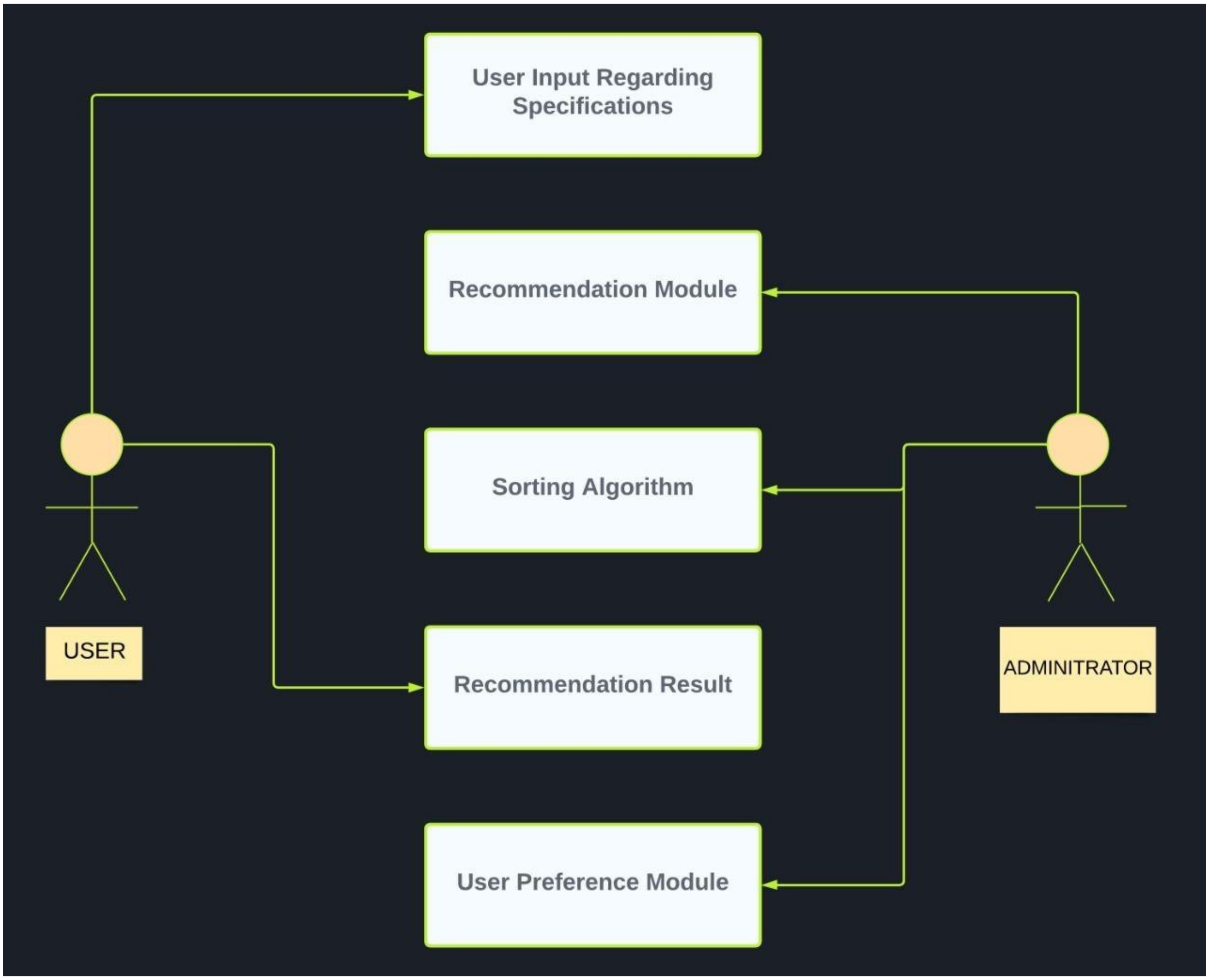


Figure 3.3:Use CaseDiagram

3.4 Activity Diagram

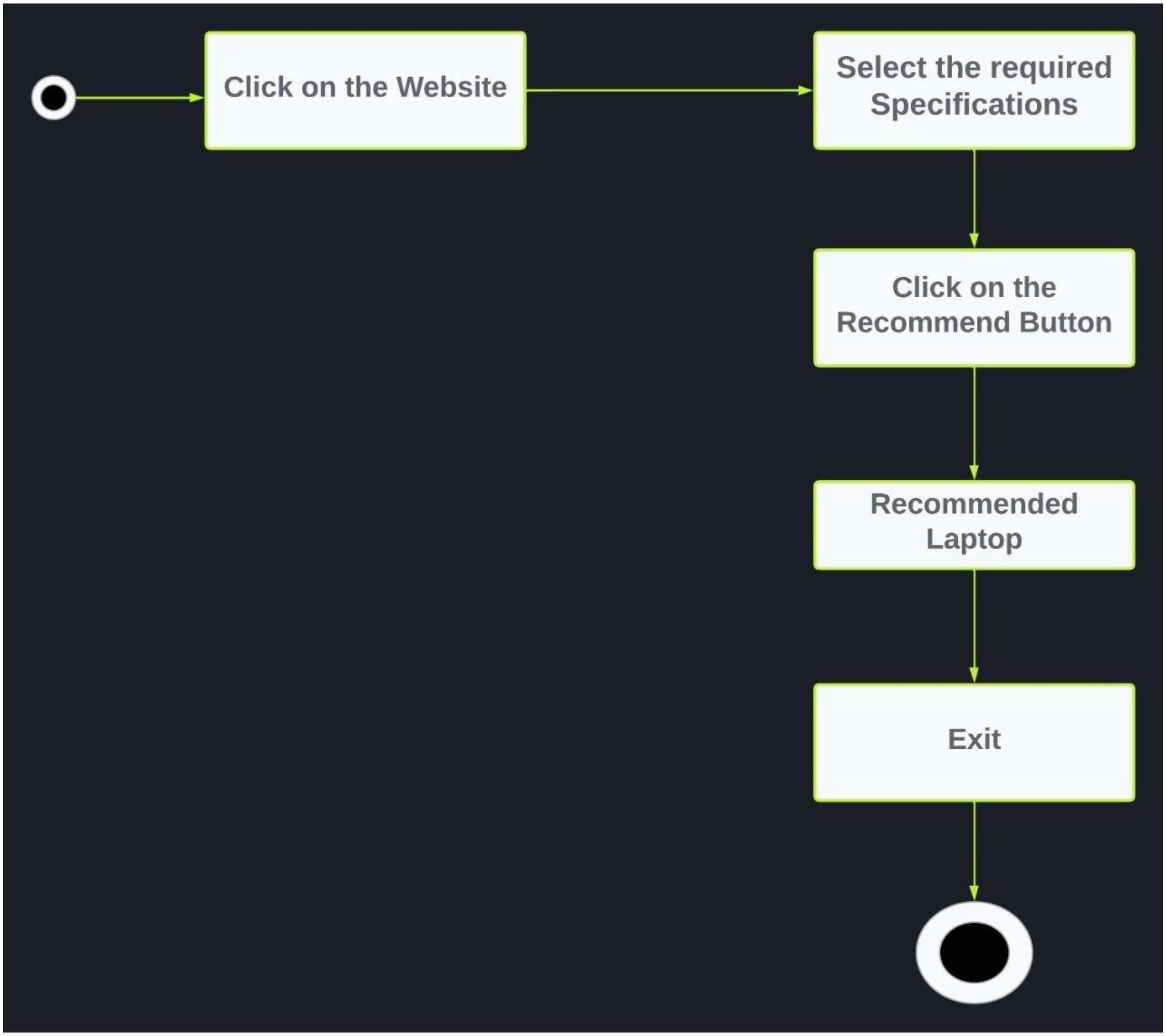

Click on the Website
Select the required Specifications
Click on the Recommend Button
Recommended Laptop
Exit

Figure 3.4:ActivityDiagram

3.5 Hardware and Software Requirements.

Software

1. Jupiter notebook

2. Python

Hardware

Laptop / Computer

3.6 Methodology

- First we will take the dataset, refine it in jupiter notebook and then create the lists and then create the JSON files by dumping it.
- Then we will write the KNN algorithm in the VScode.
- Now the framework will be build by using the Streamlit as the total coding part is in python.
- As the framework is done. We will deploy the framework on Heroku to build the website.
- Now running the code, after running it one needs to give the specifications according to his requirement.
- Output displayed. The laptop will be recommended.

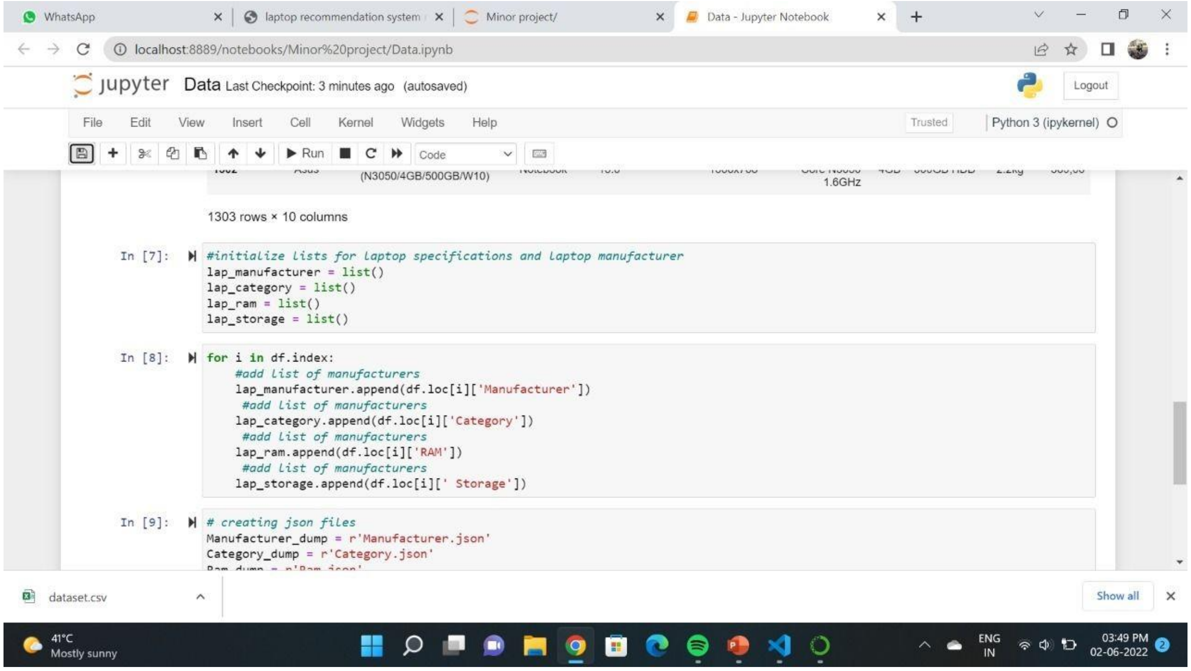
4. IMPLEMENTATION DETAILS

4.1 K-nearest neighbor Algorithm

1) The algorithm is used to solve the classification model problems. K-nearest neighbor basically creates an imaginary boundary to classify the data.
2) When the new data points come in, the algorithm will try to predict that to the nearest of the boundary line.
3) Therefore, largest k value means smother curves of separation resulting in less complex models.
4) Whereas, smaller k value tends to overfit the data and resulting in complex modes.

4.2 Screen shot of software prototype and Results

Figure 4.1: Home Page Of System

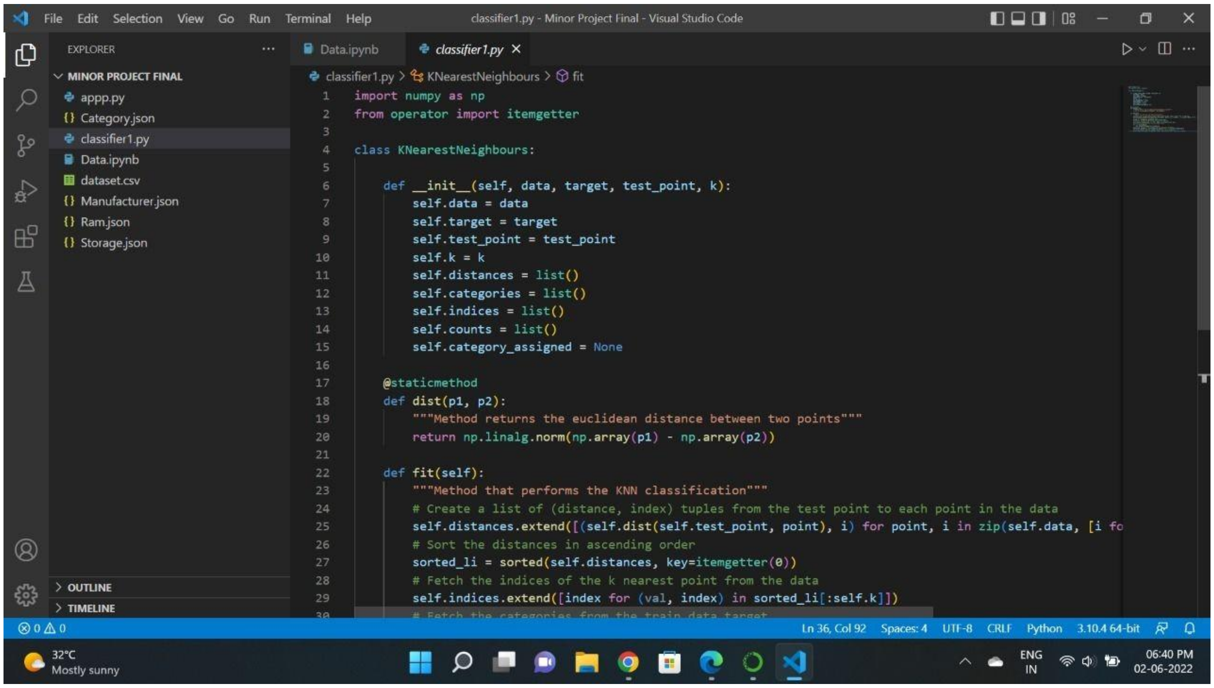


(N3050/4GB/500GB/W10) 1.6GHz

1303 rows × 10 columns

```
In [7]:  #initialize lists for laptop specifications and laptop manufacturer
         lap_manufacturer = list()
         lap_category = list()
         lap_ram = list()
         lap_storage = list()

In [8]:  for i in df.index:
             #add list of manufacturers
             lap_manufacturer.append(df.loc[i]['Manufacturer'])
              #add list of manufacturers
             lap_category.append(df.loc[i]['Category'])
              #add list of manufacturers
             lap_ram.append(df.loc[i]['RAM'])
              #add list of manufacturers
             lap_storage.append(df.loc[i][' Storage'])

In [9]:  # creating json files
         Manufacturer_dump = r'Manufacturer.json'
         Category_dump = r'Category.json'
         Ram_dump = r'Ram.json'
```

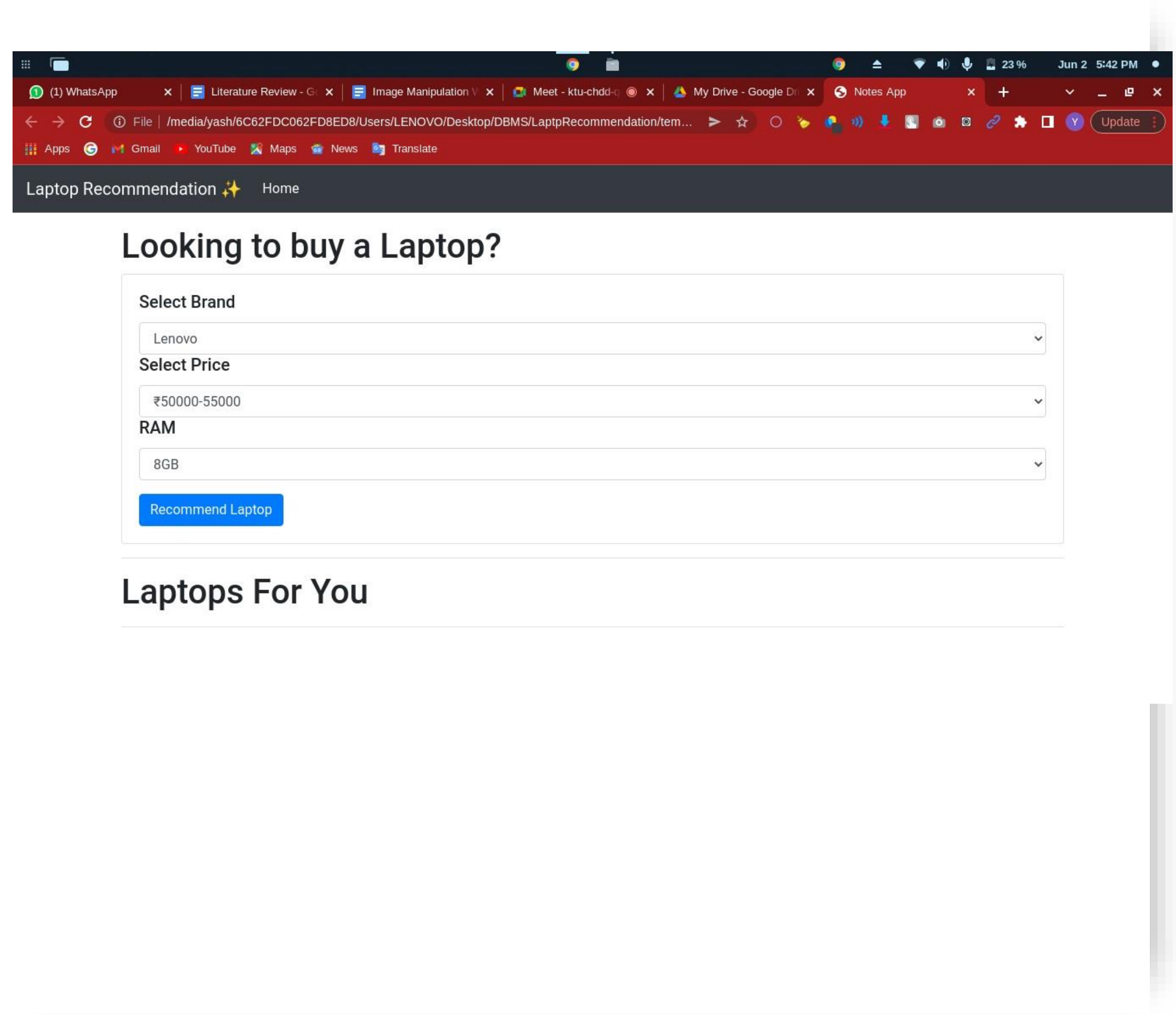
Figure 4.2: Project Code



```
import numpy as np
from operator import itemgetter

class KNearestNeighbours:

    def __init__(self, data, target, test_point, k):
        self.data = data
        self.target = target
        self.test_point = test_point
        self.k = k
        self.distances = list()
        self.categories = list()
        self.indices = list()
        self.counts = list()
        self.category_assigned = None

    @staticmethod
    def dist(p1, p2):
        """Method returns the euclidean distance between two points"""
        return np.linalg.norm(np.array(p1) - np.array(p2))

    def fit(self):
        """Method that performs the KNN classification"""
        # Create a list of (distance, index) tuples from the test point to each point in the data
        self.distances.extend([(self.dist(self.test_point, point), i) for point, i in zip(self.data, [i fo
        # Sort the distances in ascending order
        sorted_li = sorted(self.distances, key=itemgetter(0))
        # Fetch the indices of the k nearest point from the data
        self.indices.extend([index for (val, index) in sorted_li[:self.k]])
        # Fetch the categories from the train data target
```

Figure 4.3: Project Code

Specification page



Laptop Recommendation ✨ Home

Looking to buy a Laptop?

Select Brand

Lenovo

Select Price

₹50000-55000

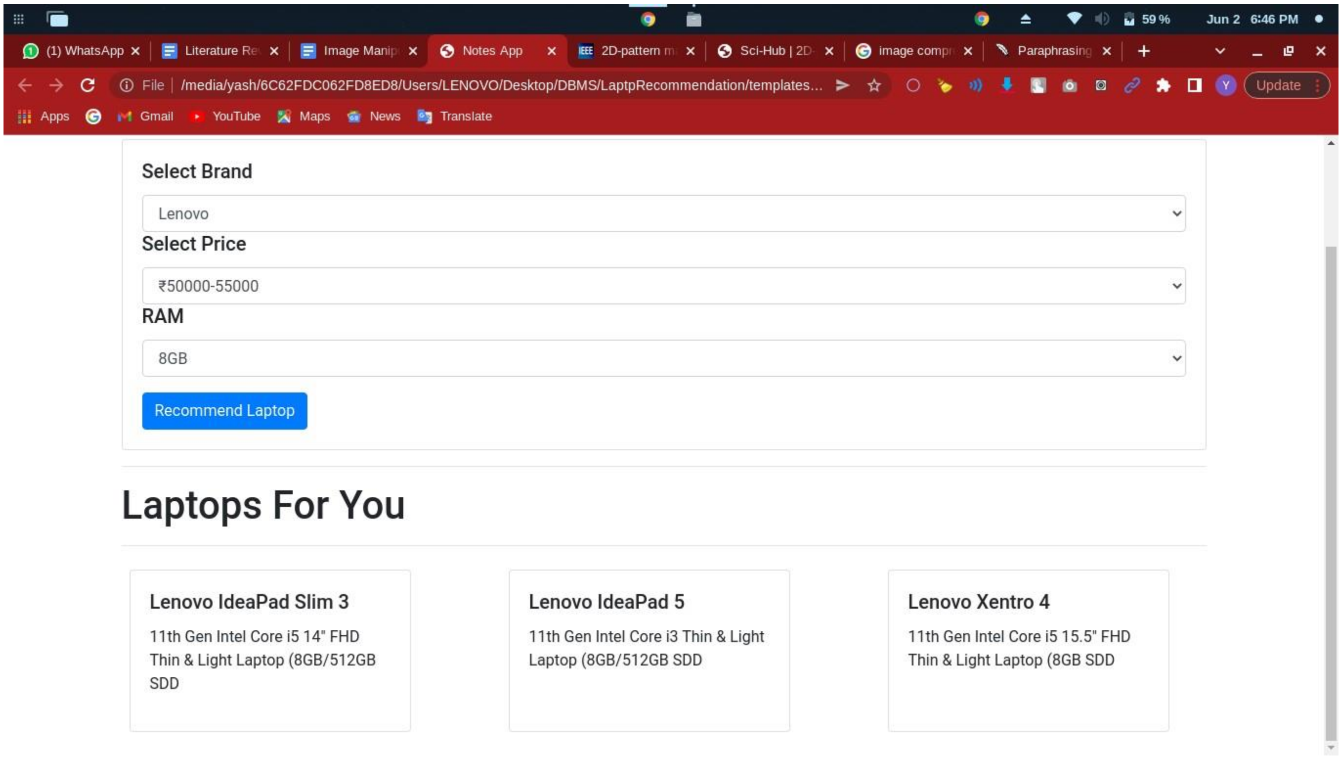
RAM

8GB

Recommend Laptop

Laptops For You

Figure 4.4:Specification page



Select Brand

Lenovo

Select Price

₹50000-55000

RAM

8GB

Recommend Laptop

Laptops For You

Lenovo IdeaPad Slim 3

11th Gen Intel Core i5 14" FHD Thin & Light Laptop (8GB/512GB SDD

Lenovo IdeaPad 5

11th Gen Intel Core i3 Thin & Light Laptop (8GB/512GB SDD

Lenovo Xentro 4

11th Gen Intel Core i5 15.5" FHD Thin & Light Laptop (8GB SDD

Figure 4.5:Final Output of the project

5. CONCLUSION AND FUTURE SCOPE

5.1 Conclusion

We propose a personalized attribute based recommender system as a solution to less frequently purchase products. Our proposed system incorporates a set of techniques for mining the requirements of customers and the attributes of laptop products, in order to recommend optimal products to prospective buyers of laptop computers. The system is able to provide online buyers with information on the products that could best meet their individual needs. The system also has the potential of increasing sales for online business, thereby making online shopping more interesting and profitable to both buyers and sellers.

5.2 Future Scope and Challenges

In the interest of further analysing other factors that increase/decrease laptop price and understandreview affect on laptop price, it would be of interest to:

1) Scrap the average rating and review a summary of each laptop and use sentimental analysis rank each of laptop.
2) Create more feature engineering relating to different specifications, such as whether the laptop contains a USB-C port or what version of Bluetooth support.
3) Use and test different types of scalars on the data-set and understand its effect on the model prediction.

REFERENCES

[1] Ojokho, B.A.Samule , Omisore, Ogunniyi, “Fuzzy logic Based Personalised recommender system”, International Journal of Computer Science and Information Technology and Security . 2012.

[2] Ting Chen, Wei-Li Han, Hai-Dong Wang , Bin Xu,”Content Recommender System Based on Private Dynamic user profile”, IEEE, Proceedings of Sixth International Conference of Machine learning and Cybernetic, Hongkong. August 2007.

[3] M.S.Ayundhita , Z.K.A.Baizal , Y Siboconi.” Ontology based Conversational Recommender System for recommending laptop” ,International Conference on Data & Information Science : Journal of Physics. 1192.2019.

[4] Lalita Sharma, Anju Gera, “A Survey of Recommendation System : Research challenges”, International Journal of Engineering Trends & Technology. Volume 4 Issue5 – May 2013.

[5] A Pushpalatha, Harish Sanmugam , J Jeya , Pradeep K , Madhu Bala S. “ Gadget Recommendation System using Data Science” , IEEE, Proceedings of Third International conference on Intelligent Sustainable Sytems [ICISS 2020].

[6] Adhi Kusnadi and Edwin Kurnaiawan, “ Implementation of Topsis Method In Web Based System Recommendations for students laptop selection” ,IJNMT, Vol.IV , No.1 | June 2017.

[7] Mohammad Doud, S.K Naqvi , and Asad Ahmad, “Opinion Observer : Recommendation System on E-commerce Website “, International Journal of Computer Applications (0975-8887) Volume 105 – No. 14, November 2014. Moniter-social-distancing- 180975059/

[8] Kam Fung Yeung , Yanyan yang and David Ndzi, “A proactive personalized mobile recommendation system using hierarchy process and Bayesian Network: , Springer 20 July 2012.

[9] N.A.Osman , S.A.M. Noah , and M . Darwich , “Decision Support System for Laptop Selection Using AHP Method and Profile Matching” , International Journal of Machine Learning and Computing , Vol.9 , No. 4 , August 2019.

[10] Muhammad Mukharir and Retantoy Wardoyo , “ Decision Support System for Laptop Selection Using AHP Method and Profile Matching” , IJCCS (Indonesial Journal of Computing and Cybernetics Systems) Vol .15 , No.3 , July 2021, pp.307~316

CODE:

Jupiter notebook code :

```
import json
import pandas as pd
#load dataset
data = pd.read_csv('dataset.csv')
data.head()
# created new data frame with relevent colums only

df = data[['Manufacturer','Model Name', 'Category','Screen Size','Screen','CPU','RAM',
Storage','Weight','Price (Euros)']]
df

#initialize lists for laptop specifications and laptop manufacturer
lap_manufacturer = list()
lap_category = list()
lap_ram = list()
lap_storage = list()

for i in df.index:

#add list of manufacturers
    lap_manufacturer.append(df.loc[i]['Manufacturer'])
    #add list of manufacturers
    lap_category.append(df.loc[i]['Category'])
     #add list of manufacturers
    lap_ram.append(df.loc[i]['RAM'])
    #add list of manufacturers
    lap_storage.append(df.loc[i][' Storage'])
```

```
# creating json files
Manufacturer_dump = r'Manufacturer.json'
Category_dump = r'Category.json' Ram_dump
= r'Ram.json'
Storage_dump = r'Storage.json'

with open (Category_dump, 'w+', encoding='utf-8') as f:
    json.dump(lap_category , f)
with open (Manufacturer_dump, 'w+', encoding='utf-8') as f:
    json.dump(lap_manufacturer ,f)
with open (Ram_dump, 'w+', encoding='utf-8') as f:
    json.dump(lap_ram ,f)
with open (Storage_dump, 'w+', encoding='utf-8') as f:
    json.dump(lap_storage ,f)
```

Code of classifier i.e KNN algorithum

```
import numpy as np

from operator import itemgetter
class KNearestNeighbours:

    def___init__(self, data, target, test_point, k):
        self.data = data
        self.target = target
        self.test_point = test_point
        self.k = k
        self.distances = list()
        self.categories = list()
        self.indices = list()
        self.counts = list()
self.category_assigned = None

@staticmethod def
    dist(p1, p2):
    """Method returns the euclidean distance between two points"""
        return np.linalg.norm(np.array(p1) - np.array(p2))
    def fit(self):

        """Method that performs the KNN classification"""

        # Create a list of (distance, index) tuples from the test point to each point in the data
        self.distances.extend([(self.dist(self.test_point, point), i) for point, i in zip(self.data,
[i for i in range(len(self.data))])])

        # Sort the distances in ascending order

        sorted_li = sorted(self.distances, key=itemgetter(0))

        # Fetch the indices of the k nearest point from the data
        self.indices.extend([index for (val, index) in sorted_li[:self.k]]) #
        Fetch the categories from the train data target
        for i in self.indices:
            self.categories.append(self.target[i])
        # Fetch the count for each category from the K nearest neighbours
        self.counts.extend([(i, self.categories.count(i)) for i in set(self.categories)])# Find
        the highest repeated category among the K nearest neighbours
        self.category_assigned = sorted(self.counts, key=itemgetter(1), reverse=True)[0][0]
```

Code of application building

```
from unicodedata import category
import streamlit as st
import json
from Classifier import KNearestNeighbours
from operator import itemgetter

# Load data and movies list from corresponding JSON files
with open(r'data.json','r+',encoding='utf-8') as f:
    data = json.load(f)
with open(r'Model.json','r+',encoding='utf-8') as f:
    laptop_models = json.load(f)
with open(r'Category.json','r+',encoding='utf-8') as f:
    category = json.load(f)

def knn(test_point, k):
    # create dummy variables for the KNN classifier
    target = [0 for item in laptop_models]
    # Instantiate object for the classifier
    model = KNearestNeighbours(data,target,test_point,k=k)
    # Run the algorithum
    model.fit()
    # Distances to most distant movie
    max_dist = sorted(model.distances, key = itemgetter(0))[-1]
    # Print list of 10 recommendations
    table = list()
    for i in model.indices:
        # Return back laptop model
        table.append([laptop_models[i][0],laptop_models[i][2]])
    return table

if _name_ == '_main_':
```

```
Manufacturer

= ['Acer','Apple','Asus','Chuwi','Dell','Fujitsu','Google','HP','Huawei','LG','Lenovo',
   'MSI','Mediacom','Microsoft','Razer','Samsung','Toshiba','Vero','Xiaomi']
   laptops = [model[0] for model in laptop_models]
   st.header('Laptop Recommendation System')
   apps = ['---Select---','Company based', 'Category based' ]
   app_options = st.selectbox('Select application :',apps)
   super1 = ['Workstation',
'Netbook',
'Ultrabook',
'2 in 1 Convertible',
'Notebook',
'Gaming']
   if app_options == 'Category based':
       laptop_select = st.selectbox('Select Category :',['--Select--'] + super1)
       if laptop_select == '--Select--':
           st.write('Select a Category')
       else:
           n = st.number_input('Number of laptops :',min_value = 5 , max_value=20,step=1)
           Manufacturer = data[category.index(laptop_select)]
           test_point = Manufacturer
           table = knn(test_point, n)
           for laptop, CPU in table:
               # Display model name with ram
               st.markdown(f"[{laptop}]({CPU})")

   elif app_options == apps[1]:
       options = st.multiselect('Select Manufactuere :',Manufacturer)
       if options:
           RAM = st.slider('RAM :',4 , 64, 8)
           n = st.number_input('Number of laptops :', min_value=5, max_value=20, step=1)
```

```
test_point = [1 if manufacturer in options else 0 for manufacturer in Manufacturer ]
          test_point.append(RAM)
          table = knn(test_point,n)
          for laptop , CPU in table:
              # Display laptop model with CPU
              st.markdown(f"[{laptop}]({CPU})")

      else:
          st.write(" ")

  else:
      st.write('Select option')
```