NLP in recommendation Engine

A Project Report

Submitted in partial fulfillment of the

Requirements for the award of the Degree of

MASTER OF SCIENCE (INFORMATION TECHNOLOGY)

By

Mr. Vishal Kashyap & Mr. Akshay Oza

Under the esteemed guidance of

Ms. Manali Patil

DEPARTMENT OF INFORMATION TECHNOLOGY

CHIKITSAK SAMUHA'S

S.S & L.S PATKAR COLLEGE OF ARTS & SCIENCE

&

V. P. VARDE COLLEGE OF COMMERECE & ECONOMICS.

(Affiliated to University of Mumbai)

S.V. ROAD, GOREGAON (W), MUMBAI – 400 062

MAHARASHTRA

2020-2021

PNR No.: …………………… Roll no: 25, 40

1. Name of the Student
Vishal Kashyap, Akshay Oza

2. Title of the Project
NLP in Recommendation Engine

3. Name of the Guide
Ms. Manali Patil

4. Teaching experience of the Guide?

5. Is this your first submission? Yes No

Signature of the Student Signature of the Guide

Date: ……………………. Date: ……………………..…

Signature of the Coordinator

Date: …………………….

ACKNOWLEDGEMENT

We are extremely delighted for getting an opportunity to express our gratitude for all those good souls who were more than guiding stars in helping us to complete the project.

Our first and foremost thanks goes to **Patkar-Varde College** and our faculties from whom we have learnt a lot.

We are deeply indebted to our principal **Dr. Shrikant B. Sawant**, our Chief Education Officer **Dr. (Mrs.) Mala Kharkar** and our coordinator as well as our project guide **Ms. Manali Patil**. Their valuable feedback helped us tremendously to improve our project in many ways.

The infectious chain of gratitude would be far from complete without our friends who were with us through the difficult times and who taught us many things that matter in improving our project.

DECLARATION

We hereby declare that the project entitled, **“NLP in Recommendation Engine”** done at **Patkar-Varde College,** has not been in any case duplicated to submit to any other university for the award of any degree. To the best of our knowledge other Than us, no one has submitted to any other university.

The project is done in partial fulfilment of the requirements for the award of degree of **MASTER’S OF SCIENCE (INFORMATION TECHNOLOGY)** to be submitted as final semester project as part of our curriculum.

Mr. Vishal Kashyap, Mr. Akshay Oza

Signature……………………….

INDEX

Chapter 1

NLP in Recommendation Engine

1.1 Background

Recommendation engine is most commonly used in retail business or ecommerce websites to increase sales and generate revenue out of it. A recommendation engine is a system that suggests products, services, information to users based on analysis of data. Notwithstanding, the recommendation can derive from a variety of factors such as the history of the user and the behaviour of similar users. Recommendation systems are quickly becoming the primary way for users to expose to the whole digital world through the lens of their experiences, behaviors, preferences and interests. And in a world of information density and product overload, a recommendation engine provides an efficient way for companies to provide consumers with personalized information and solutions. Most of the recommendation engine algorithms use content based or collaborative filtering (User – User/Item - Item) or Hybrid approach (both). And in all these approaches, rating is used as a feedback variable. As we know some of the customers give their feedback as a rating and some as a comment. In the current approach most of the companies are only using rating as a feedback variable because it is in numeric format, and to run any algorithm, the data need to be in numeric format. People don't use comments as a feedback variable because it is in a non-numeric format. But using NLP technique we can convert this non numeric data into numeric format, prior to which we can use this in our recommendation collaborative filtering approach.

Introduction

In a very general way, recommender systems are algorithms aimed at suggesting relevant items to users (items being movies to watch, text to read, products to buy or anything else depending on industries).

Recommender systems aim to predict users' interests and recommend product items that quite likely are interesting for them. They are among the most powerful machine learning systems that online retailers implement in order to drive sales.

Data required for recommender systems stems from explicit user ratings after watching a movie or listening to a song, from implicit search engine queries and purchase histories, or from other knowledge about the users/items themselves.

1.2 Objectives

In this project, the main objective is to use NLP techniques to convert the unstructured data into structured data to enrich our dataset and use that data into recommendation engine algorithm to improve the performance our algorithm, which in return will improve the accuracy of our algorithm and will predict correct recommendation to our customers so they can buy that products.
These systems use information filtering techniques to process information and provide the user with potentially more relevant items by numerically estimating the users' preferences for unseen items or to provide users with item lists ranked in accordance to the estimated

1.3 Purpose, Scope and Applicability

1.3.1 Purpose

The purpose of a recommender system is to suggest relevant items to users. To achieve this task, there exist two major categories of methods : collaborative filtering methods and content based methods. Before digging more into details of particular algorithms, Taking briefly on Collaborative filtering.

- Collaborative methods for recommender systems are methods that are based solely on the past interactions recorded between users and items in order to produce new recommendations. These interactions are stored in the so-called “user-item interactions matrix”.

1.1.1 Scope

Firstly, we will need unstructured data like comments, Ratings, and then we will apply Natural language processing technique to convert unstructured data like comments into structured data on the scale of 0 to 5. Then we have to fill the missing values in rating variables, from the data which we converted into structured data and use it in our recommendation engine algorithm for better prediction.

1.1.2 Applicability

This algorithm is applicable to any organization, who collects unstructured data like comments and are not using their recommendation engine algorithm to enrich data. Almost any business can benefit from a recommendation system. There are two important aspects that determine how much a business benefits from a recommendation system:

- **Breadth of data**: A business serving only a handful of customers that behave in different ways will not receive much benefit from an automated recommendation system. Humans are still much better than machines in the area of learning from a few examples. In such cases, your employees will use their logic, qualitative and quantitative understanding of customers to make accurate recommendations.
- **Depth of data**: Having a single data point on each customer is also not helpful to recommendation systems. Deep data about customers online activities and if possible offline purchases can guide accurate recommendations.

Chapter 2: Description of the problems/topics

2.1 Problem and State why it matters

Most common problem which comes up when building a recommendation engine is ‘High percentage of Data Sparsity’. Sparsity and density are terms used to describe the percentage of cells in a database table that are not populated and populated, respectively. The sum of the sparsity and density should equal 100%. A table that is 10% dense has 10% of its cells populated with non-zero values. It is therefore 90% sparse – meaning that 90% of its cells are either not filled with data or are zeros. This is a huge problem in building recommendation engines and needs to be solved. Because if we recommend the wrong product to customers they might not get proper service, which may lead to customer loss.

2.2 Description of work

Recommendation systems collect customer data and auto analyze this data to generate customized recommendations for your customers. These systems rely on both implicit data such as browsing history and purchases and explicit data such as ratings provided by the user.

Content based filtering and collaborative filtering are two approaches commonly used to generate recommendations.

In this project we’ve implemented item-to-item(Collaborative filtering) and it is model based(Matrix based).

Now to reduce the percentage of Data Sparsity and to increase the percentage of Data density, we have to use the data which we already collect but we don’t use it in our recommendation algorithm. Here, in this project we’ve used unstructured data like comments which the people in industry ideally don’t use for building recommendation engines

2.3 Propose solution and benefits of proposed solution

To solve the problem of High data sparsity, and to increase the percentage of data density, we are using NLP techniques to convert the unstructured data into structured data. We are going to normalize the comments of customers on the scale of 0-5. Wherever the rating is missing, we will use the normalized values which we have generated using NLP techniques

Benefits of proposed solution would be,

- Reduced Churn.
- Increased sales/conversion
- Increased loyalty and share of mind

2.4 Summarizing the problem and solution

Solving the problem of Data Sparsity in the current recommendation engine using NLP techniques for improving the performance and accuracy of algorithms

Chapter 3: Status of the research/knowledge in the field and literature review

3.1 Literature Survey

We have referred these research paper:

- Scienstein: A Research Paper Recommender System
- A unified context-free grammar and n-gram model for spoken language processing
- Collaborative Document Evaluation: An Alternative Approach to Classic Peer Review.
- Comparative study of existing system.
- NLP BASED SERVICE RECOMMENDATION SYSTEM
- Recommender Systems: An Overview, Research Trends, and Future Directions

3.2 Survey of Technology

Requirement Analysis

- Frontend : HTML, CSS, JavaScript
- Backend : Python, Flask, Sklearn, Pandas
- Libraries used :
  - Flask : Flask is a micro web framework written in Python
  - Jinja2 : Jinja is a fast, expressive, extensible templating engine
  - Numpy : NumPy is a library for the Python programming language, adding support for large, multi-dimensional arrays and matrices
  - Openpyxl : openpyxl is a Python library to read/write Excel 2010 xlsx/xlsm/xltx/xltm files.
  - Pandas : pandas is a fast, powerful, flexible and easy to use open source data analysis
  - Requests : requests library is the de facto standard for making HTTP requests in Python
  - scikit-learn : Simple and efficient tools for predictive data analysis
  - Scipy : SciPy (pronounced “Sigh Pie”) is a Python-based ecosystem of open-source software for mathematics, science, and engineering.
  - Sklearn : Simple and efficient tools for predictive data analysis
  - WTForms : WTForms is a flexible forms validation and rendering library for Python web development
  - Xlrd : xlrd is a library for reading data and formatting information from Excel files in the historical .xls format

3.3 Technology Used

a. PYTHON - Python is an interpreted, high-level, general-purpose programming language. Python's design philosophy emphasizes code readability with its notable use of significant whitespace. Its language constructs and object-oriented approach aim to help programmers write clear, logical code for small and large-scale projects. Python is dynamically typed AND supports multiple programming paradigms, including procedural, object-oriented, and functional programming.

b. MACHINE LEARNING - Machine learning is the scientific study of algorithms and statistical models that computer systems use in order to perform a specific task effectively without using explicit instructions, relying on patterns and inference instead. It is seen as a subset of artificial intelligence. Machine learning algorithms build a mathematical model based on sample data, known as "training data", in order to make predictions or decisions without being explicitly told.

Chapter 4: Description of the methodology/approach.

4.1 Methodological approach

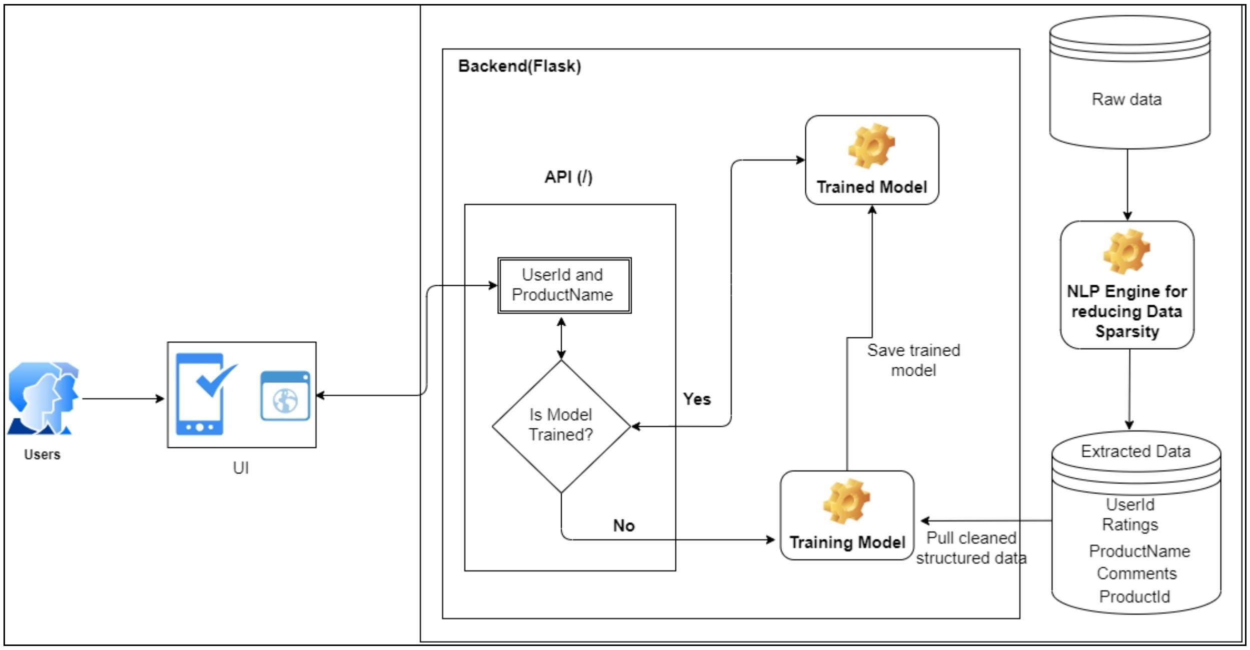
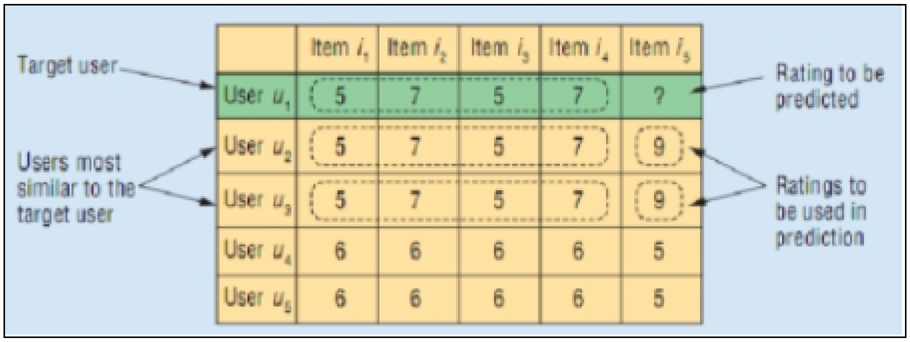


Fig: 1 System Diagram for Proposed Method

In the approach shown, the primary focus is to prepare the good quality of data by reducing Data Sparsity using NLP techniques and eventually increase the accuracy of recommendation system.


Target user
Users most similar to the target user
Rating to be predicted
Ratings to be used in prediction

	Item i_1	Item i_2	Item i_3	Item i_4	Item i_5
User u_1	5	7	5	7	?
User u_2	5	7	5	7	9
User u_3	5	7	5	7	9
User u_4	6	6	6	6	5
User u_5	6	6	6	6	5

Fig: 2 Item-to-item filtering matrix

In this project, Onlick(button click) is the event upon which Products are recommended on UI, but in other e-commerce website, pageload is the event upon which other products are recommended.

4.2 Planning and Scheduling

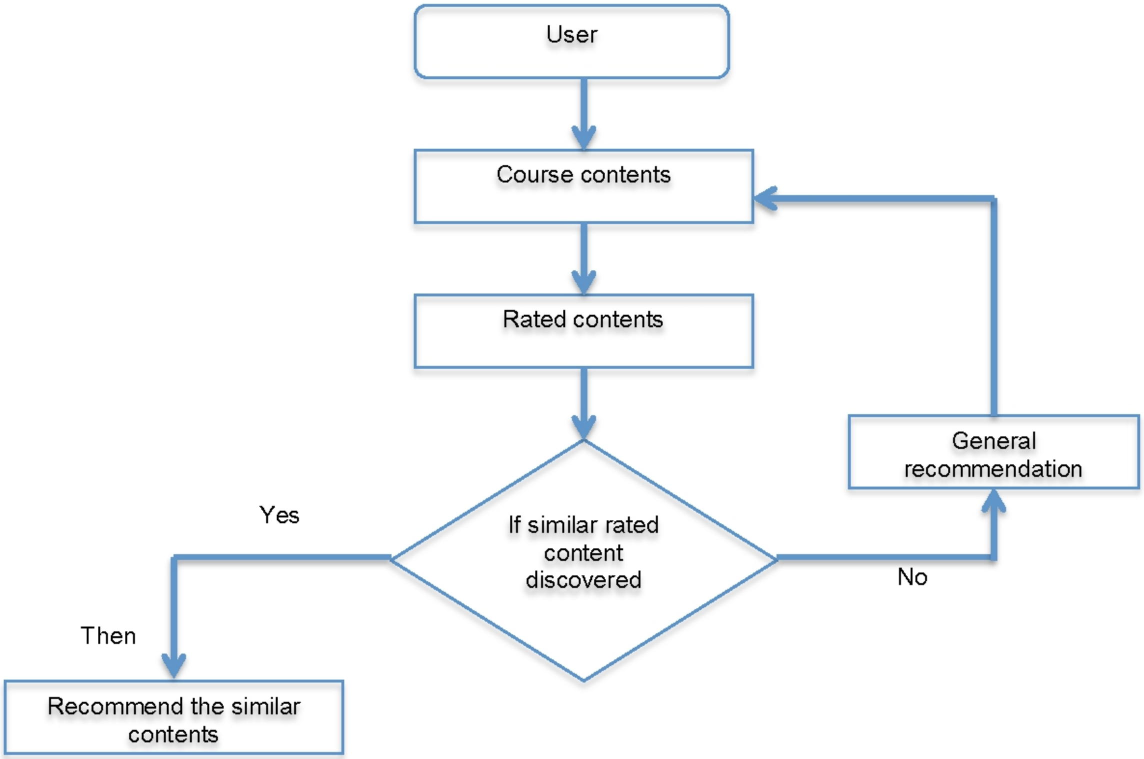
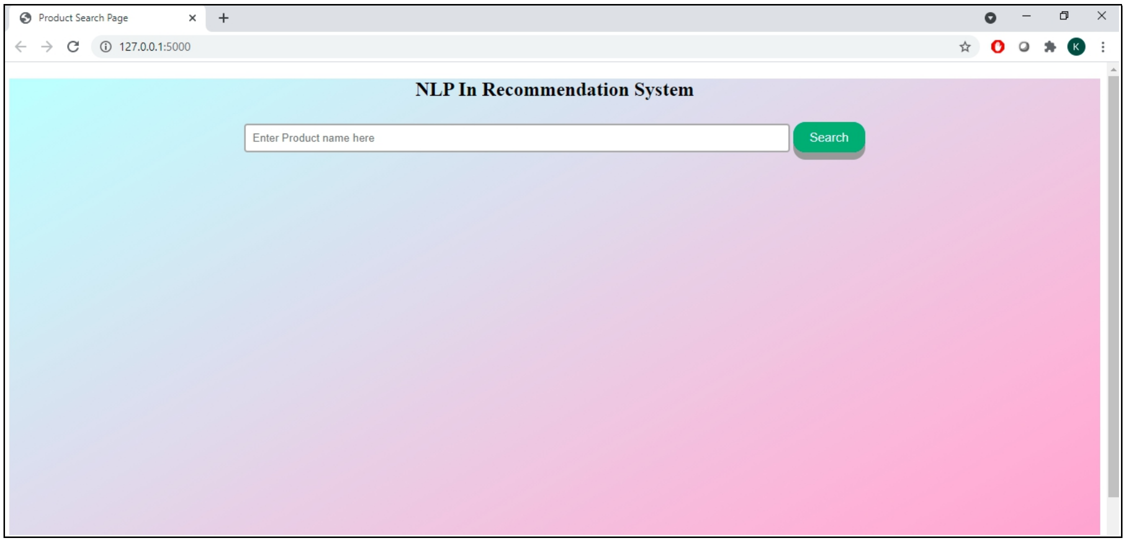
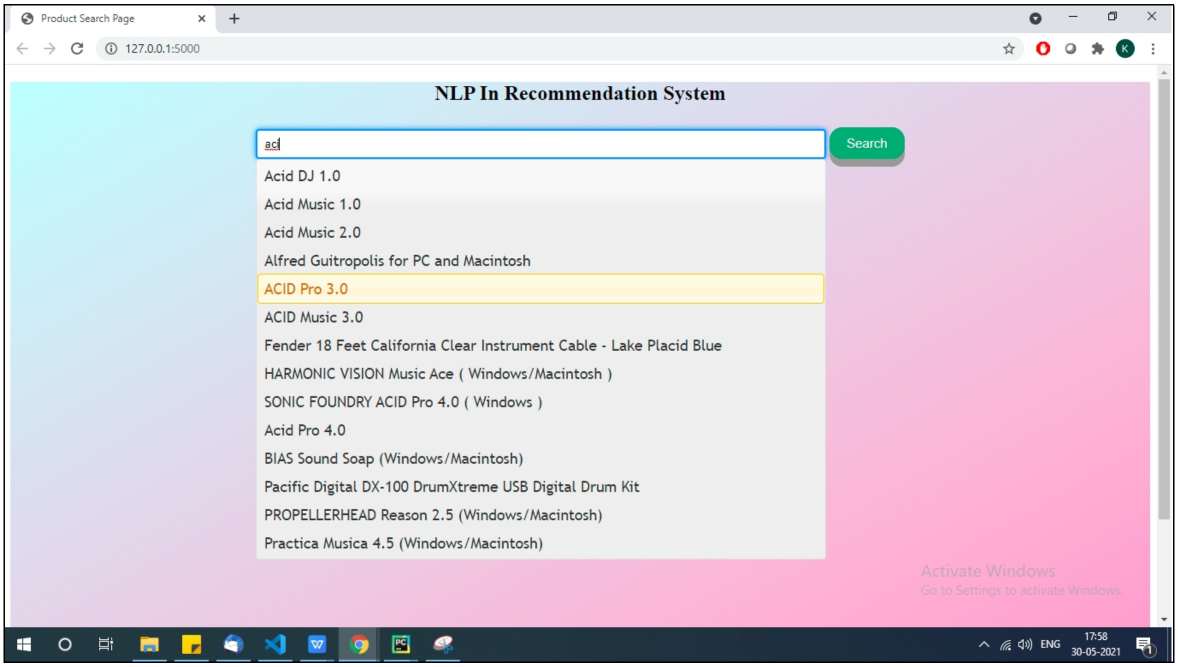

User
Course contents
Rated contents
General recommendation
Yes
If similar rated content discovered
No
Then
Recommend the similar contents

Fig 3. FLOWCHART

4.3 User Interface

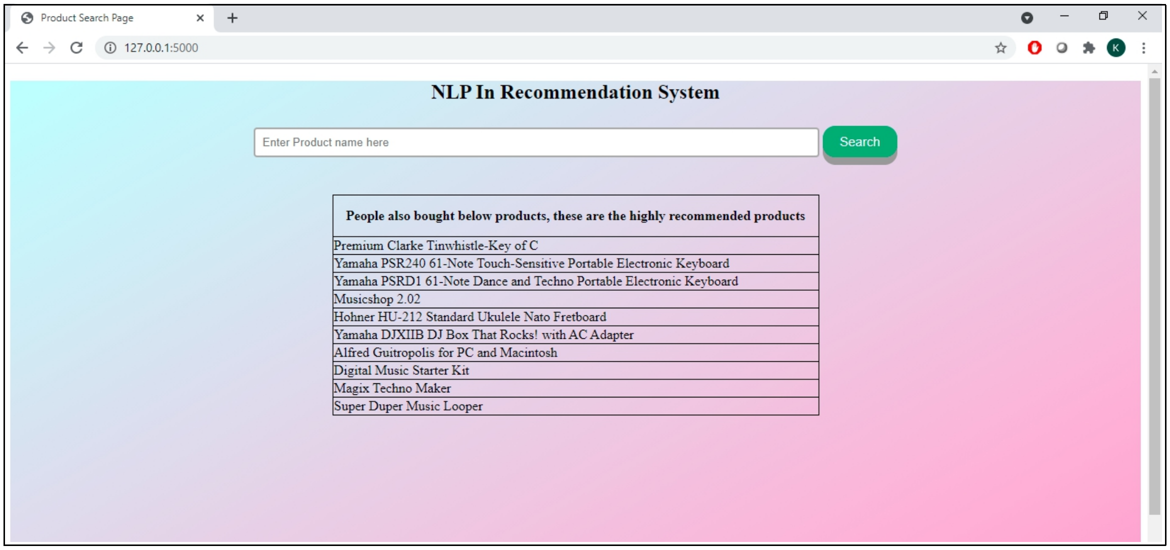




NLP In Recommendation System

ac

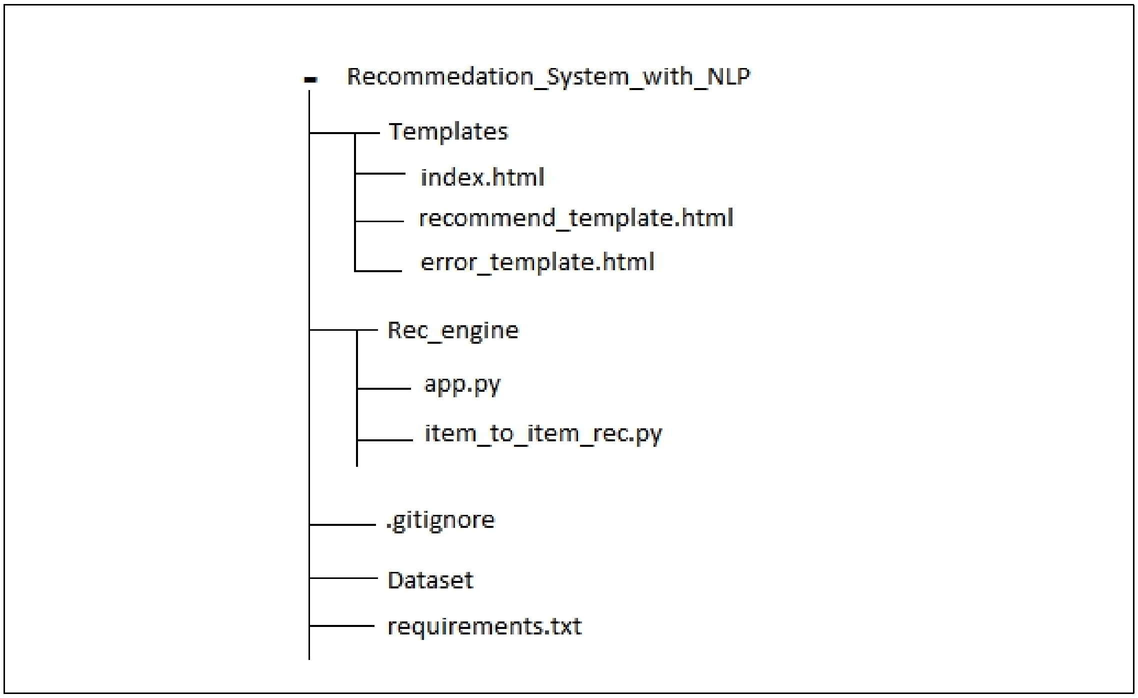
Acid DJ 1.0
Acid Music 1.0
Acid Music 2.0
Alfred Guitropolis for PC and Macintosh
ACID Pro 3.0
ACID Music 3.0
Fender 18 Feet California Clear Instrument Cable - Lake Placid Blue
HARMONIC VISION Music Ace (Windows/Macintosh)
SONIC FOUNDRY ACID Pro 4.0 (Windows)
Acid Pro 4.0
BIAS Sound Soap (Windows/Macintosh)
Pacific Digital DX-100 DrumXtreme USB Digital Drum Kit
PROPELLERHEAD Reason 2.5 (Windows/Macintosh)
Practica Musica 4.5 (Windows/Macintosh)



NLP In Recommendation System

People also bought below products, these are the highly recommended products
Premium Clarke Tinwhistle-Key of C
Yamaha PSR240 61-Note Touch-Sensitive Portable Electronic Keyboard
Yamaha PSRD1 61-Note Dance and Techno Portable Electronic Keyboard
Musicshop 2.02
Hohner HU-212 Standard Ukulele Nato Fretboard
Yamaha DJXIIB DJ Box That Rocks! with AC Adapter
Alfred Guitropolis for PC and Macintosh
Digital Music Starter Kit
Magix Techno Maker
Super Duper Music Looper

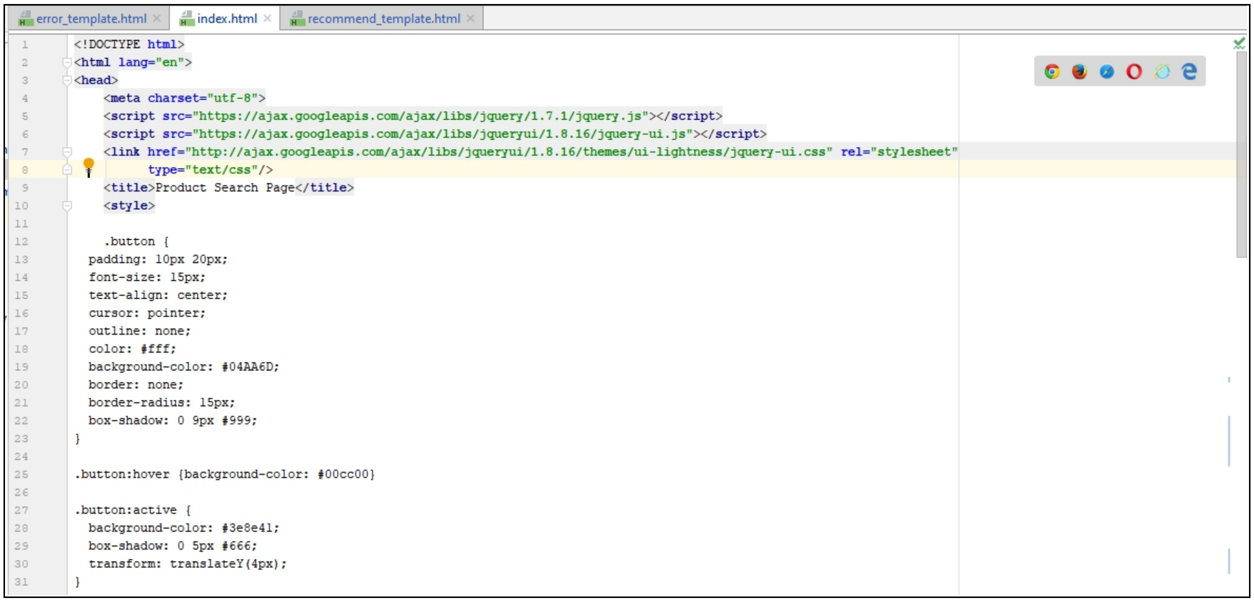
4.4 Coding details



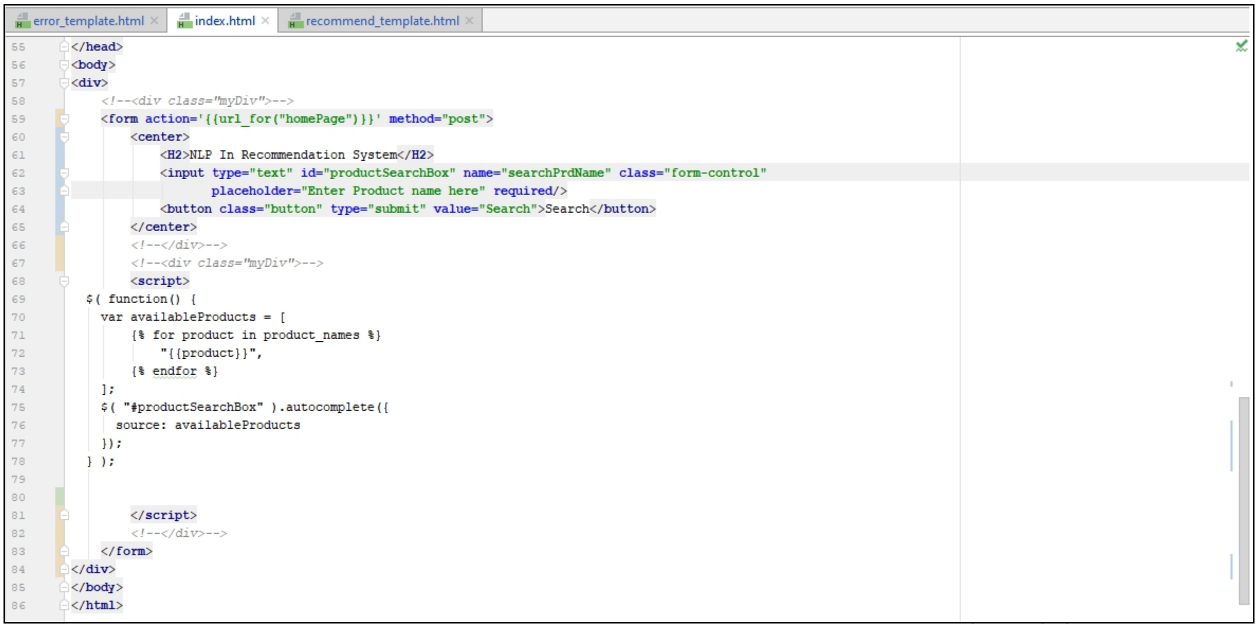
Project Structure

All code is been stored on github : [***https://github.com/vishalkashyap95/recommendation_engine_item_to_item.git***](https://github.com/vishalkashyap95/recommendation_engine_item_to_item.git)

- templates - This folder contains the static UI pages(index.html, error_template.html, recommend_template.html)



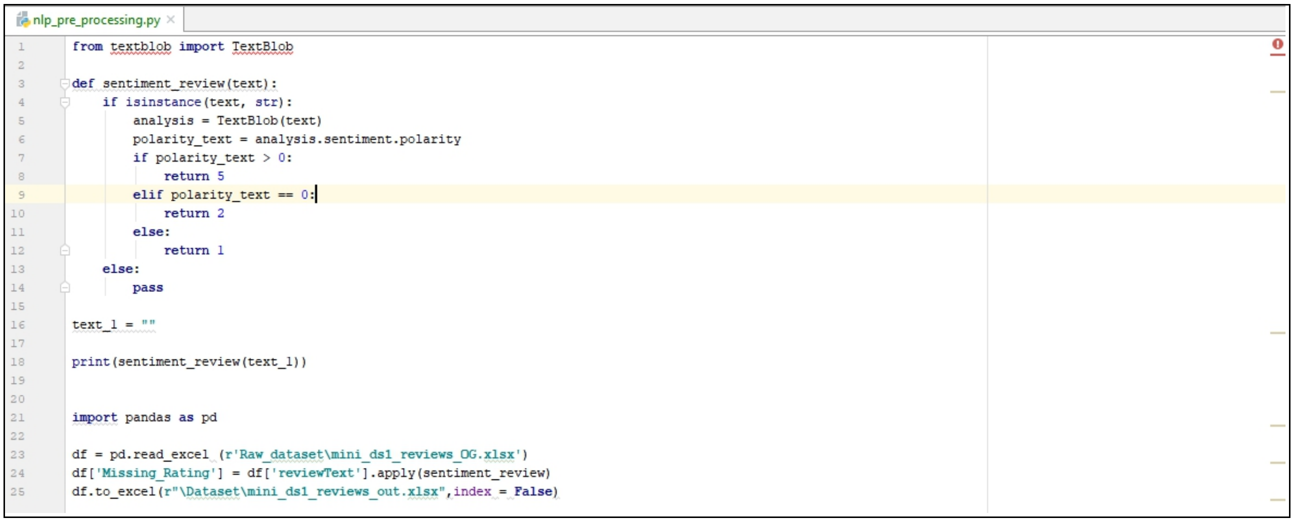
```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="utf-8">
    <script src="https://ajax.googleapis.com/ajax/libs/jquery/1.7.1/jquery.js"></script>
    <script src="https://ajax.googleapis.com/ajax/libs/jqueryui/1.8.16/jquery-ui.js"></script>
    <link href="http://ajax.googleapis.com/ajax/libs/jqueryui/1.8.16/themes/ui-lightness/jquery-ui.css" rel="stylesheet"
          type="text/css"/>
    <title>Product Search Page</title>
    <style>

    .button {
  padding: 10px 20px;
  font-size: 15px;
  text-align: center;
  cursor: pointer;
  outline: none;
  color: #fff;
  background-color: #04AA6D;
  border: none;
  border-radius: 15px;
  box-shadow: 0 9px #999;
}

.button:hover {background-color: #00cc00}

.button:active {
  background-color: #3e8e41;
  box-shadow: 0 5px #666;
  transform: translateY(4px);
}
```



error_template.html × index.html × recommend_template.html ×

```html
</head>
<body>
<div>
    <!--<div class="myDiv">-->
    <form action='{{url_for("homePage")}}' method="post">
        <center>
            <H2>NLP In Recommendation System</H2>
            <input type="text" id="productSearchBox" name="searchPrdName" class="form-control"
                   placeholder="Enter Product name here" required/>
            <button class="button" type="submit" value="Search">Search</button>
        </center>
        <!--</div>-->
        <!--<div class="myDiv">-->
        <script>
  $( function() {
    var availableProducts = [
        {% for product in product_names %}
            "{{product}}",
        {% endfor %}
    ];
    $( "#productSearchBox" ).autocomplete({
      source: availableProducts
    });
  } );


        </script>
        <!--</div>-->
    </form>
</div>
</body>
</html>
```

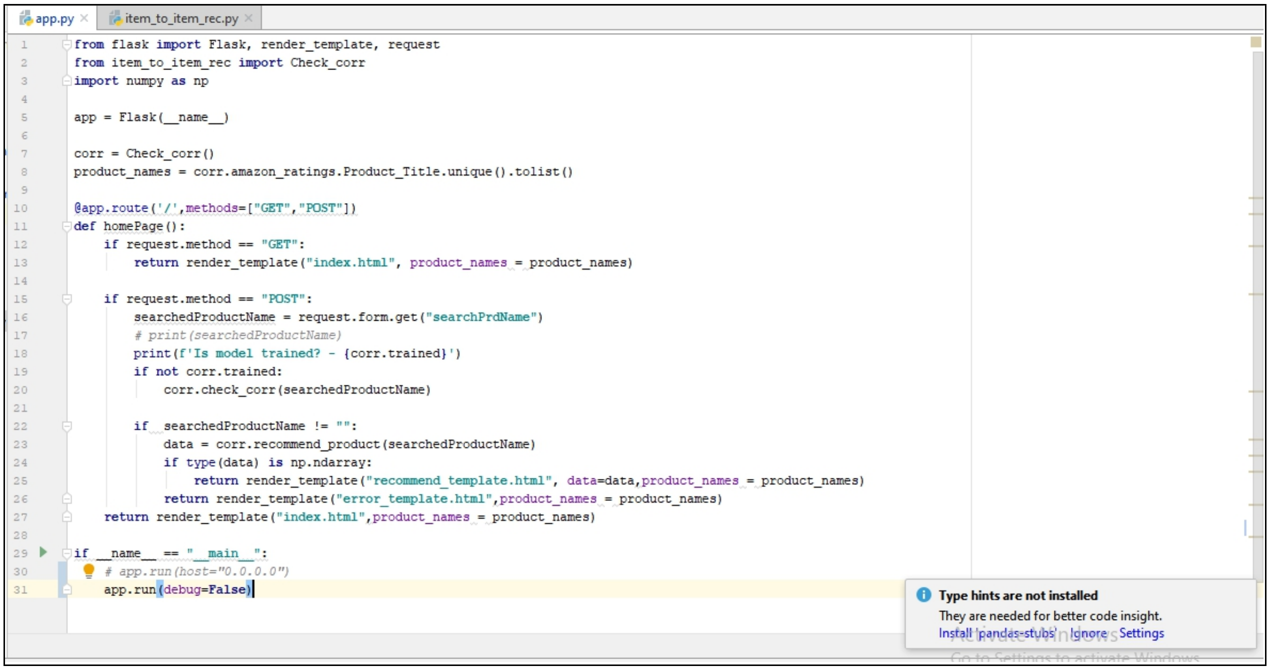
- nlp_pre_processing.py - This file is used to pre-process the raw data. Preprocessing steps are:
    - Reading the comments
    - Converting those comments into numerical(ratings) form using TextBlob library.
    - Droping unwanted columns



nlp_pre_processing.py ×

```python
from textblob import TextBlob

def sentiment_review(text):
    if isinstance(text, str):
        analysis = TextBlob(text)
        polarity_text = analysis.sentiment.polarity
        if polarity_text > 0:
            return 5
        elif polarity_text == 0:
            return 2
        else:
            return 1
    else:
        pass

text_1 = ""

print(sentiment_review(text_1))


import pandas as pd

df = pd.read_excel (r'Raw_dataset\mini_ds1_reviews_OG.xlsx')
df['Missing_Rating'] = df['reviewText'].apply(sentiment_review)
df.to_excel(r"\Dataset\mini_ds1_reviews_out.xlsx",index = False)
```

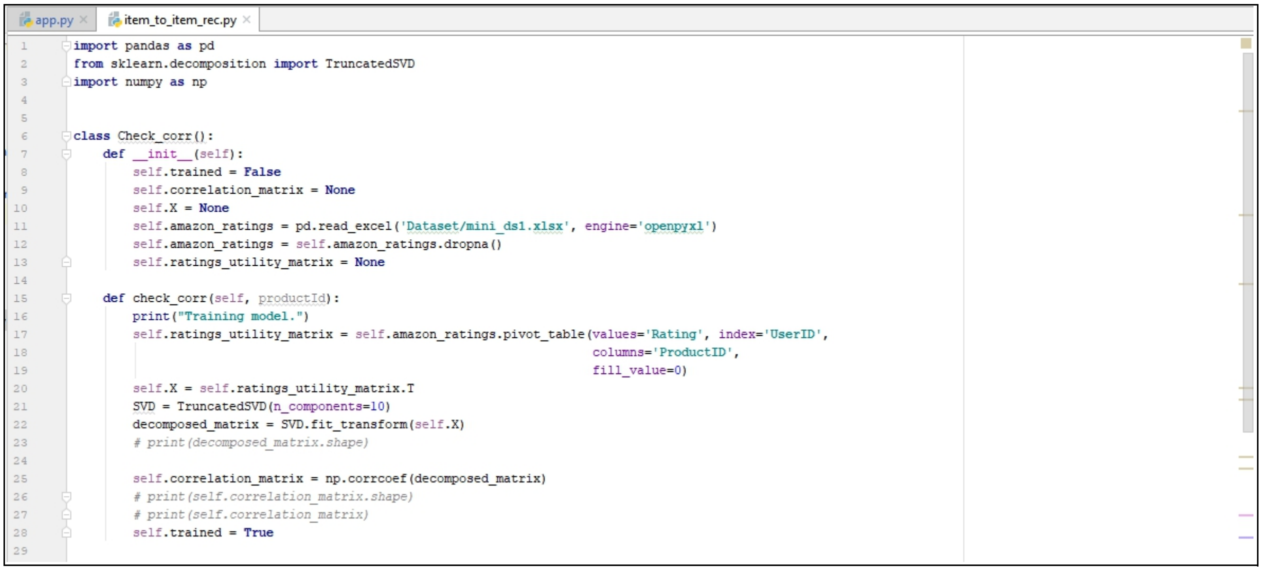
- app.py - This file creates an flask app and a route **("/"),** whenever user visit localhost:8080, **"/"** route will receive the request and return the predicted output.



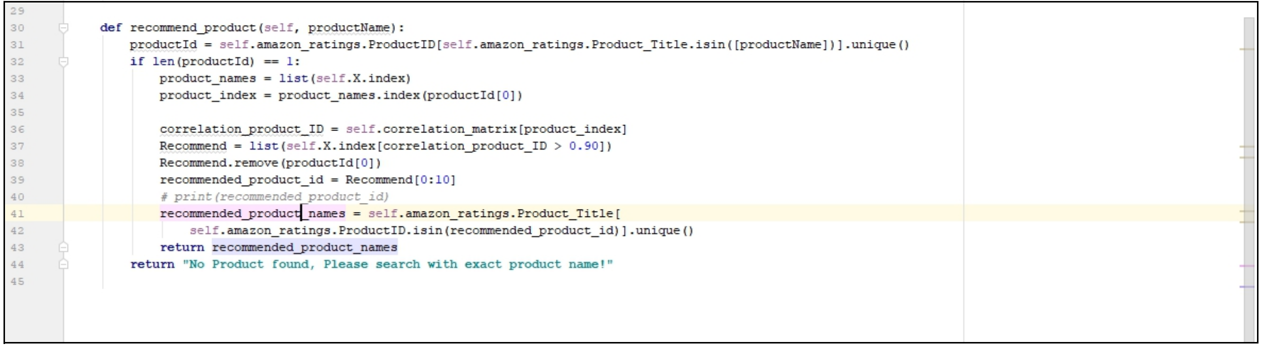
```python
from flask import Flask, render_template, request
from item_to_item_rec import Check_corr
import numpy as np

app = Flask(__name__)

corr = Check_corr()
product_names = corr.amazon_ratings.Product_Title.unique().tolist()

@app.route('/',methods=["GET","POST"])
def homePage():
    if request.method == "GET":
        return render_template("index.html", product_names = product_names)

    if request.method == "POST":
        searchedProductName = request.form.get("searchPrdName")
        # print(searchedProductName)
        print(f'Is model trained? - {corr.trained}')
        if not corr.trained:
            corr.check_corr(searchedProductName)

        if  searchedProductName != "":
            data = corr.recommend_product(searchedProductName)
            if type(data) is np.ndarray:
                return render_template("recommend_template.html", data=data,product_names = product_names)
            return render_template("error_template.html",product_names = product_names)
    return render_template("index.html",product_names = product_names)

if __name__ == "__main__":
    # app.run(host="0.0.0.0")
    app.run(debug=False)
```

Type hints are not installed
They are needed for better code insight.
Install 'pandas-stubs' Ignore Settings

- Item_to_item_rec.py - This file contains the actual model training and prediction part.



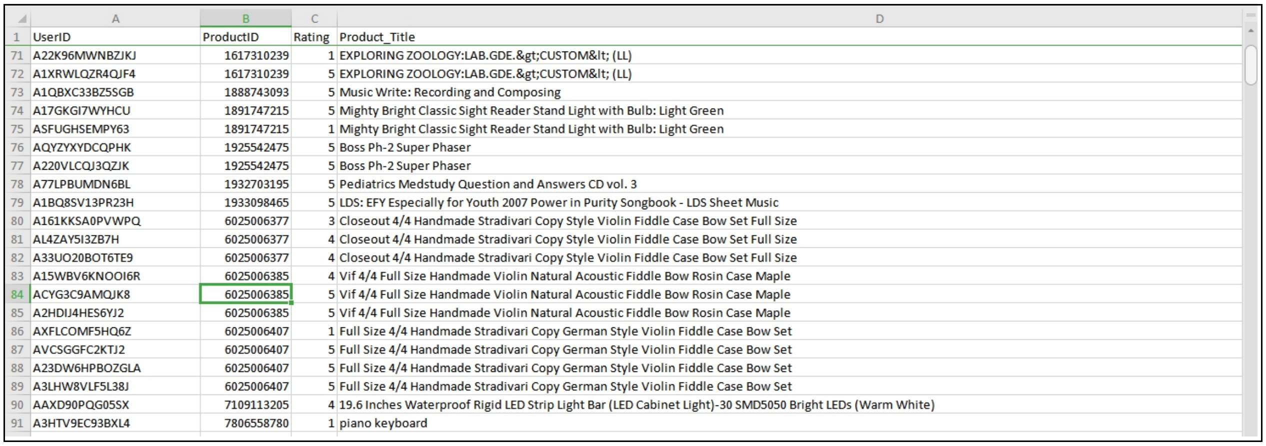
```python
import pandas as pd
from sklearn.decomposition import TruncatedSVD
import numpy as np


class Check_corr():
    def __init__(self):
        self.trained = False
        self.correlation_matrix = None
        self.X = None
        self.amazon_ratings = pd.read_excel('Dataset/mini_ds1.xlsx', engine='openpyxl')
        self.amazon_ratings = self.amazon_ratings.dropna()
        self.ratings_utility_matrix = None

    def check_corr(self, productId):
        print("Training model.")
        self.ratings_utility_matrix = self.amazon_ratings.pivot_table(values='Rating', index='UserID',
                                                                      columns='ProductID',
                                                                      fill_value=0)

        self.X = self.ratings_utility_matrix.T
        SVD = TruncatedSVD(n_components=10)
        decomposed_matrix = SVD.fit_transform(self.X)
        # print(decomposed_matrix.shape)

        self.correlation_matrix = np.corrcoef(decomposed_matrix)
        # print(self.correlation_matrix.shape)
        # print(self.correlation_matrix)
        self.trained = True

```



```python

    def recommend_product(self, productName):
        productId = self.amazon_ratings.ProductID[self.amazon_ratings.Product_Title.isin([productName])].unique()
        if len(productId) == 1:
            product_names = list(self.X.index)
            product_index = product_names.index(productId[0])

            correlation_product_ID = self.correlation_matrix[product_index]
            Recommend = list(self.X.index[correlation_product_ID > 0.90])
            Recommend.remove(productId[0])
            recommended_product_id = Recommend[0:10]
            # print(recommended_product_id)
            recommended_product_names = self.amazon_ratings.Product_Title[
                self.amazon_ratings.ProductID.isin(recommended_product_id)].unique()
            return recommended_product_names
        return "No Product found, Please search with exact product name!"

```

- requirement.txt - This file contains all the libraries which is been used in entire project with a version number.

- Dataset - This folder contains the actual cleaned data.



	A	B	C	D
1	UserID	ProductID	Rating	Product_Title
71	A22K96MWNBZJKJ	1617310239	1	EXPLORING ZOOLOGY:LAB.GDE.&gt;CUSTOM&lt; (LL)
72	A1XRWLQZR4QJF4	1617310239	5	EXPLORING ZOOLOGY:LAB.GDE.&gt;CUSTOM&lt; (LL)
73	A1QBXC33BZ5SGB	1888743093	5	Music Write: Recording and Composing
74	A17GKGI7WYHCU	1891747215	5	Mighty Bright Classic Sight Reader Stand Light with Bulb: Light Green
75	ASFUGHSEMPY63	1891747215	1	Mighty Bright Classic Sight Reader Stand Light with Bulb: Light Green
76	AQYZYXYDCQPHK	1925542475	5	Boss Ph-2 Super Phaser
77	A220VLCQJ3QZJK	1925542475	5	Boss Ph-2 Super Phaser
78	A77LPBUMDN6BL	1932703195	5	Pediatrics Medstudy Question and Answers CD vol. 3
79	A1BQ8SV13PR23H	1933098465	5	LDS: EFY Especially for Youth 2007 Power in Purity Songbook - LDS Sheet Music
80	A161KKSA0PVWPQ	6025006377	3	Closeout 4/4 Handmade Stradivari Copy Style Violin Fiddle Case Bow Set Full Size
81	AL4ZAY5I3ZB7H	6025006377	4	Closeout 4/4 Handmade Stradivari Copy Style Violin Fiddle Case Bow Set Full Size
82	A33UO20BOT6TE9	6025006377	4	Closeout 4/4 Handmade Stradivari Copy Style Violin Fiddle Case Bow Set Full Size
83	A15WBV6KNOOI6R	6025006385	4	Vif 4/4 Full Size Handmade Violin Natural Acoustic Fiddle Bow Rosin Case Maple
84	ACYG3C9AMQJK8	6025006385	5	Vif 4/4 Full Size Handmade Violin Natural Acoustic Fiddle Bow Rosin Case Maple
85	A2HDIJ4HES6YJ2	6025006385	5	Vif 4/4 Full Size Handmade Violin Natural Acoustic Fiddle Bow Rosin Case Maple
86	AXFLCOMF5HQ6Z	6025006407	1	Full Size 4/4 Handmade Stradivari Copy German Style Violin Fiddle Case Bow Set
87	AVCSGGFC2KTJ2	6025006407	5	Full Size 4/4 Handmade Stradivari Copy German Style Violin Fiddle Case Bow Set
88	A23DW6HPBOZGLA	6025006407	5	Full Size 4/4 Handmade Stradivari Copy German Style Violin Fiddle Case Bow Set
89	A3LHW8VLF5L38J	6025006407	5	Full Size 4/4 Handmade Stradivari Copy German Style Violin Fiddle Case Bow Set
90	AAXD90PQG05SX	7109113205	4	19.6 Inches Waterproof Rigid LED Strip Light Bar (LED Cabinet Light)-30 SMD5050 Bright LEDs (Warm White)
91	A3HTV9EC93BXL4	7806558780	1	piano keyboard

Requirement Analysis

- Frontend : HTML, CSS, JavaScript
- Backend : Python, Flask, Sklearn, Pandas
- Libraries used :
  - Flask : Flask is a micro web framework written in Python
  - Jinja2 : Jinja is a fast, expressive, extensible templating engine
  - Numpy : NumPy is a library for the Python programming language, adding support for large, multi-dimensional arrays and matrices
  - Openpyxl : openpyxl is a Python library to read/write Excel 2010 xlsx/xlsm/xltx/xltm files.
  - Pandas : pandas is a fast, powerful, flexible and easy to use open source data analysis
  - Requests : requests library is the de facto standard for making HTTP requests in Python
  - scikit-learn : Simple and efficient tools for predictive data analysis
  - Scipy : SciPy (pronounced “Sigh Pie”) is a Python-based ecosystem of open-source software for mathematics, science, and engineering.
  - Sklearn : Simple and efficient tools for predictive data analysis
  - WTForms : WTForms is a flexible forms validation and rendering library for Python web development
  - Xlrd : xlrd is a library for reading data and formatting information from Excel files in the historical .xls format

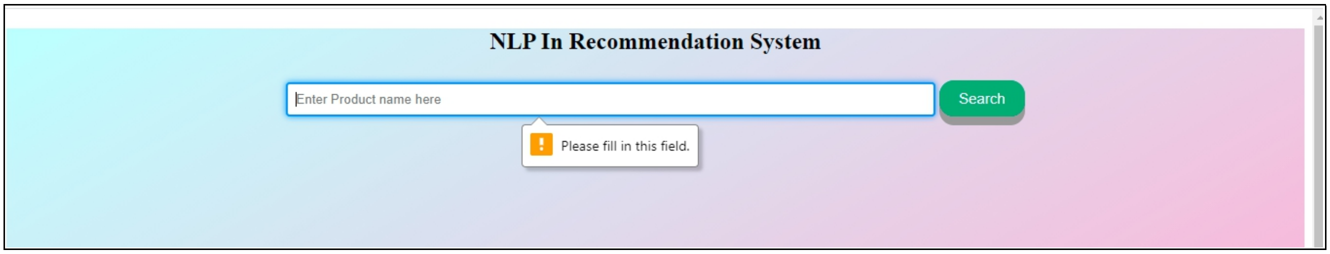
4.5 Test Cases

Test case ID	Module	Test case	Expected output	status
TC_01	Index page	Validation of input field	should not be able to search if the field is empty	Pass
TC_02	Index page	Enter random product and check the resut	Should show empty table and should suggest to select product from suggestion in the dropdown	Pass
TC_03	Index page	Check product autofill functionality	Should suggest product if the input keywords are matching with the product in the database	Pass
TC_04	Index page	Search valid product and check the recommendation output	Should get the top 10 highly correlated products list in return	Pass

Chapter 5: Results

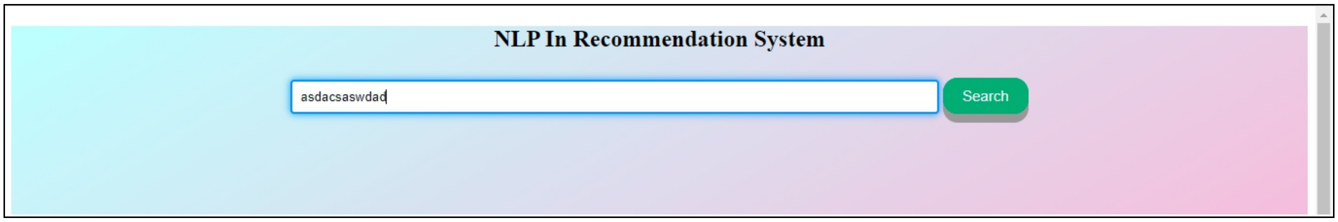
5.1 Test Reports

Test case ID	Module	Test case	Expected output	status
TC_01	Index page	Validation of input field	should not be able to search if the field is empty	Pass



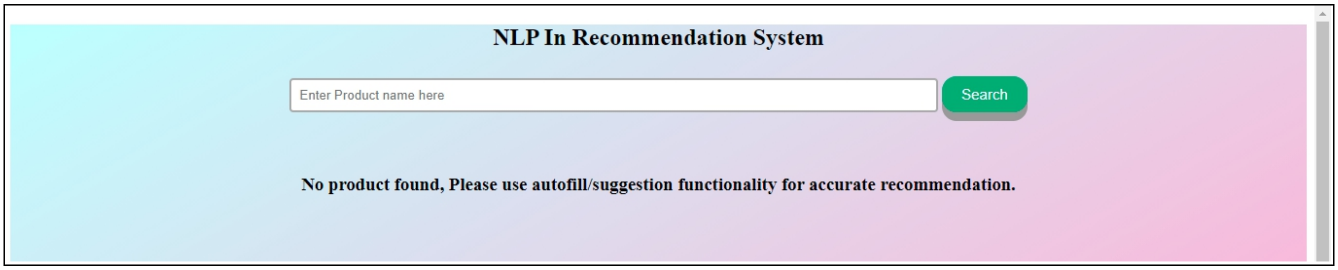
NLP In Recommendation System

Test case ID	Module	Test case	Expected output	status
TC_02	Index page	Enter random product and check the resut	Should show empty table and should suggest to select product from suggestion in the dropdown	Pass



NLP In Recommendation System

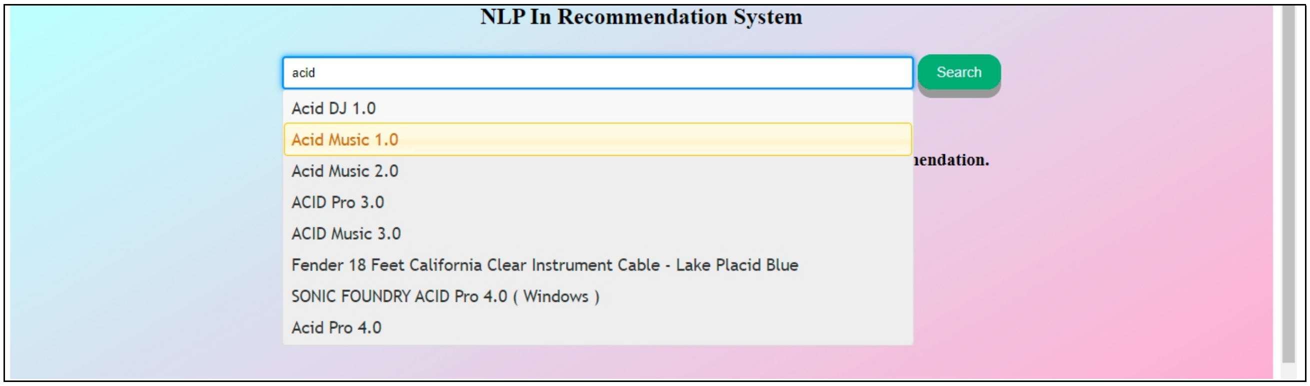
asdacsaswdad



NLP In Recommendation System

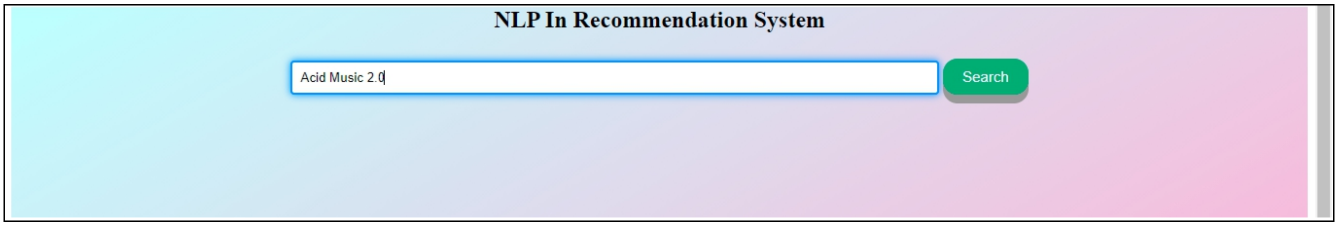
No product found, Please use autofill/suggestion functionality for accurate recommendation.

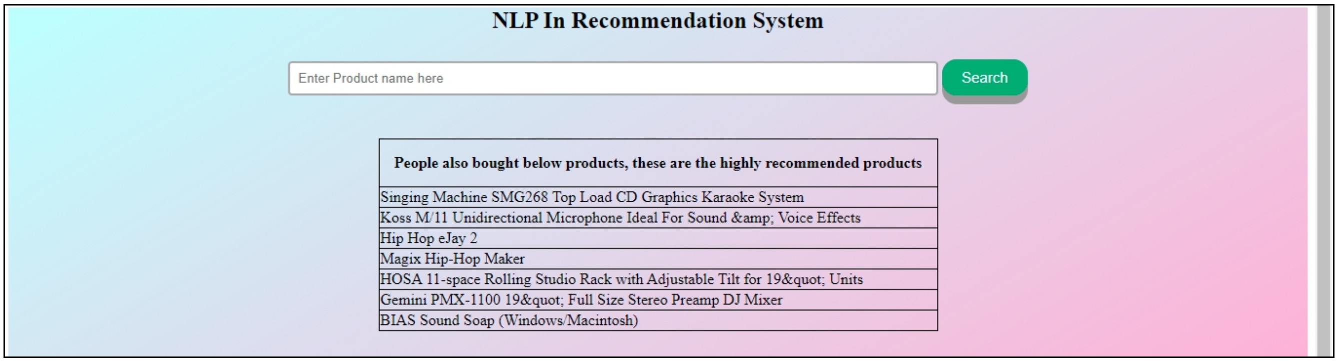
Test case ID	Module	Test case	Expected output	status
TC_03	Index page	Check product autofill functionality	Should suggest product if the input keywords are matching with the product in the database	Pass



Acid DJ 1.0
Acid Music 1.0
Acid Music 2.0
ACID Pro 3.0
ACID Music 3.0
Fender 18 Feet California Clear Instrument Cable - Lake Placid Blue
SONIC FOUNDRY ACID Pro 4.0 (Windows)
Acid Pro 4.0

Test case ID	Module	Test case	Expected output	status
TC_04	Index page	Search valid product and check the recommendation output	Should get the top 10 highly correlated products list in return	Pass





People also bought below products, these are the highly recommended products
Singing Machine SMG268 Top Load CD Graphics Karaoke System
Koss M/11 Unidirectional Microphone Ideal For Sound &amp; Voice Effects
Hip Hop eJay 2
Magix Hip-Hop Maker
HOSA 11-space Rolling Studio Rack with Adjustable Tilt for 19&quot; Units
Gemini PMX-1100 19&quot; Full Size Stereo Preamp DJ Mixer
BIAS Sound Soap (Windows/Macintosh)

5.2 User Documentation

1. Working of Software

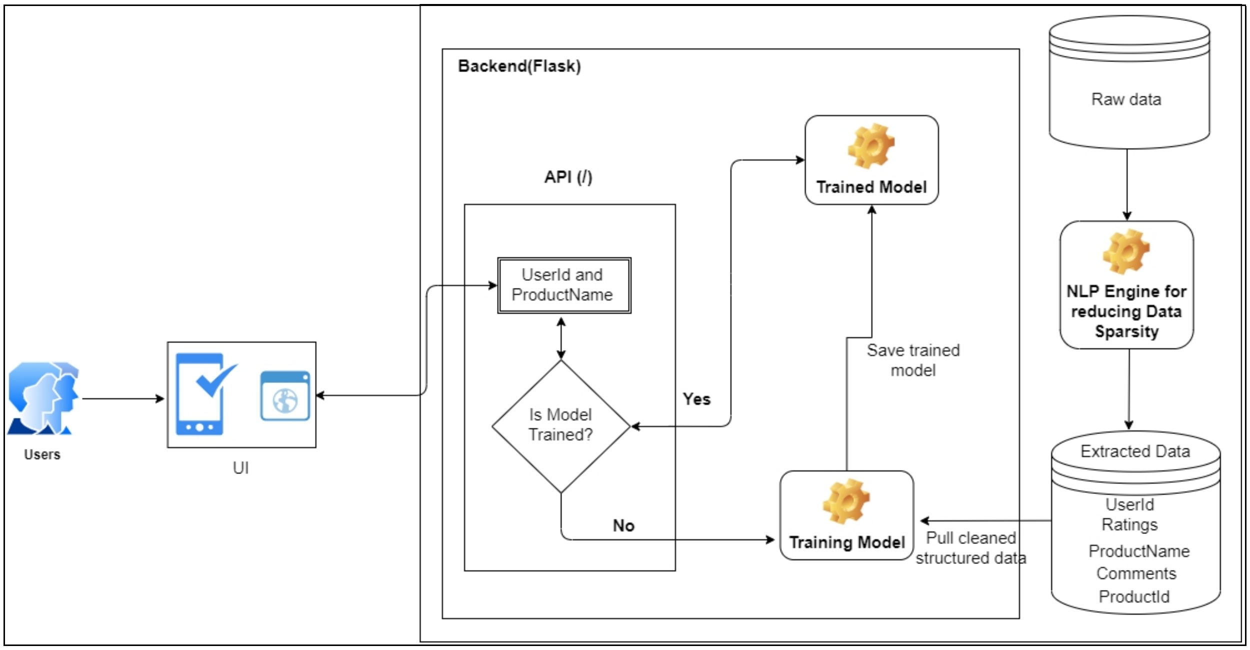


Fig 4: Flow of recommendation system

Step 1 - User search the product from User Interface(UI)

Step 2 - When user search product on UI, the request goes to backend("/" route) and the input/search product is processed.

Step 3 - API internally call the method(**homePage()**) and this method check If the model is trained or not.

Step 4 - After we launch the application, for the 1st request model won't be trained, So method(**check_corr()**) will be called and model will get trained and it will be stored onto the RAM(like a variable, which holds the value).

Step 5 - While training the model, method(**nlp_prepocessing()**) will be called and this method reads raw data and clean and convert it as per the requirement. Once the cleaning and conversion is done, it export it to a excel file.

Step 6 - If the model is trained, then the method(**recommend_product()**) will be called and that method will return the list of products.

Chapter 6: Conclusions and proposals for the future work

6.1 Conclusion:

1. Most of the recommendation engine uses content based (Item – Item), collaborative filtering (User – User) or Hybrid approach (both). And in all these approaches, rating is used as a feedback variable. As we know some of the customer give their feedback as a rating and some as a comments. In current approach most of the companies are only using rating as a feedback variable because it is in numeric format, because all the algorithm runs on numeric data. But in our research paper, we have used comments also as a feedback variable. To use comments we have converted it into numeric format between the scale of 0 – 5 (similar to rating) and use it in our algorithm. This will solve **Data Sparsity** to some extend and will improves the accuracy of algorithm. The more the data the better will be the prediction accuracy of recommendation engine.

6.2 Limitation of the System:

1. Need high computation power and sparse data for accurate prediction.
2. Very difficult to measure the performance of the algorithm at the testing phase.

6.3 Future Scope

1. To build a technique to measure the performance of the recommender system algorithm.

6.4 Bibliography

1. *[Mining Skewed and Sparse Transaction Data for Personalized Shopping. Recommendation]http://link.springer.com/article/10.1023/B:MACH.0000035471.28235.6d.
2. *[Evaluation of Item-Based Top-N Recommendation Algorithms]https://www.researchgate.net/profile/George-Karypis-2/publication/2440594_Evaluation_of_Item-Based_Top-N_Recommendation_Algorithms/links/549ee9c80cf267bdb8fdb8df/Evaluation-of-Item-Based-Top-N-Recommendation-Algorithms.pdf
3. Anand Shanker Tewari, Jyoti Prakash Singh. "Generating Top-N Items Recommendation Set Using Collaborative, Content Based Filtering and Rating Variance." International Conference on Computational Intelligence and Data Science (ICCIDS 2018)
4. Tewari, A.S., and A.G. Barman. (2018) "Sequencing of items in personalized recommendations using multiple recommendation techniques." An International Journal: Expert Systems with Applications97: 70-8
5. *[customer reviews use nlp to gain insights from your data] https://medium.com/artefact-engineering-and-data-science/customer-reviews-use-nlp-to-gain-insights-from-your-data-4629519b518e.
6. https://towardsdatascience.com/detecting-bad-customer-reviews-with-nlp-d8b36134dc7e