

SVKM's NMIMS
School of Technology Management and Engineering
A REPORT ON
STYLE SUGGEST WEBSITE

Submitted In partial fulfilment for the degree
of B.Tech. CSBS
In
ENGINEERING & TECHNOLOGY MANAGEMENT

Submitted By
Team Brute-Force
Mitanshu Holkar
Gitanshi Agarwal
Palash Mehta
Rohit Singh



Department of Computer Engineering
SVKM'S-NMIMS (Deemed-to-be-University), Indore Campus
[2022-2023]

TABLE OF CONTENTS

S. No.	Topics	Page No.
1.	Abstract	3
2.	Problem Statement	4
3.	Introduction	5
4.	Hardware & Software Requirements	6
5.	Software Description 4.1 StyleSuggest Website 4.2 EDA of Clothing Dataset	7
6.	Testing and Results	14
7.	Conclusion	17
8.	References	18

ABSTRACT

In today's highly competitive world of ecommerce, personalization has become a key factor in driving customer engagement and loyalty. With millions of products available at the click of a button, online shoppers are often overwhelmed by the sheer number of choices and are left struggling to find the products that best match their preferences and needs. As a result, ecommerce platforms have started using recommendation systems to personalize the shopping experience for their users, increasing user engagement and improving customer satisfaction.

In the fashion industry, where style and trends are constantly evolving, personalized recommendations are even more critical. Consumers have specific preferences when it comes to style, color, fabric, and brand, and a fashion recommender system can help them navigate through the vast selection of clothing items available on an ecommerce platform. By providing personalized recommendations that align with a user's individual style and preferences, a fashion recommender system can help users discover clothing items they may not have otherwise found, increasing their satisfaction with the shopping experience and ultimately driving sales for the platform.

PROBLEM STATEMENT

P3: // Effortlessly Elevate Your Fashion Game with the AI-Powered Clothing Recognition for Personalised Recommendations //

The solution should be able to detect the kind of garment, such as shirts, pants, gowns, etc., and make correct suggestions for related items.

The solution should utilise open-source computer vision libraries, such as TensorFlow or OpenCV, to make development and integration with other applications simple.

The solution should also provide an intuitive interface that enables users to post photographs and receive recommendations without difficulty.

The problem is that many people struggle with finding clothing that matches their personal style and preferences, particularly when shopping online. To address this issue, the company wants to develop an AI-powered clothing recognition solution that can make personalized recommendations based on user photos.

The solution should be able to detect the type of garment in the photo, such as shirts, pants, gowns, etc., and make correct suggestions for related items. To achieve this, the solution should utilize open-source computer vision libraries such as TensorFlow or OpenCV, which can simplify development and integration with other applications.

The solution should also provide an intuitive interface that enables users to easily upload photos and receive recommendations without difficulty. This interface should be user-friendly and provide clear instructions on how to upload photos and receive recommendations.

The goal of this solution is to help users effortlessly elevate their fashion game by providing personalized recommendations that match their style and preferences. By utilizing AI-powered clothing recognition, the solution can accurately identify the type of garment and provide suggestions for related items, thereby improving the shopping experience for users. Additionally, by utilizing opensource computer vision libraries, the solution can be developed and integrated more easily with other applications.

However, the development of such a solution requires significant investment in AI and machine learning technology, as well as expertise in computer vision and software development. Additionally, the solution may require access to a large database of clothing images to ensure accurate recommendations.

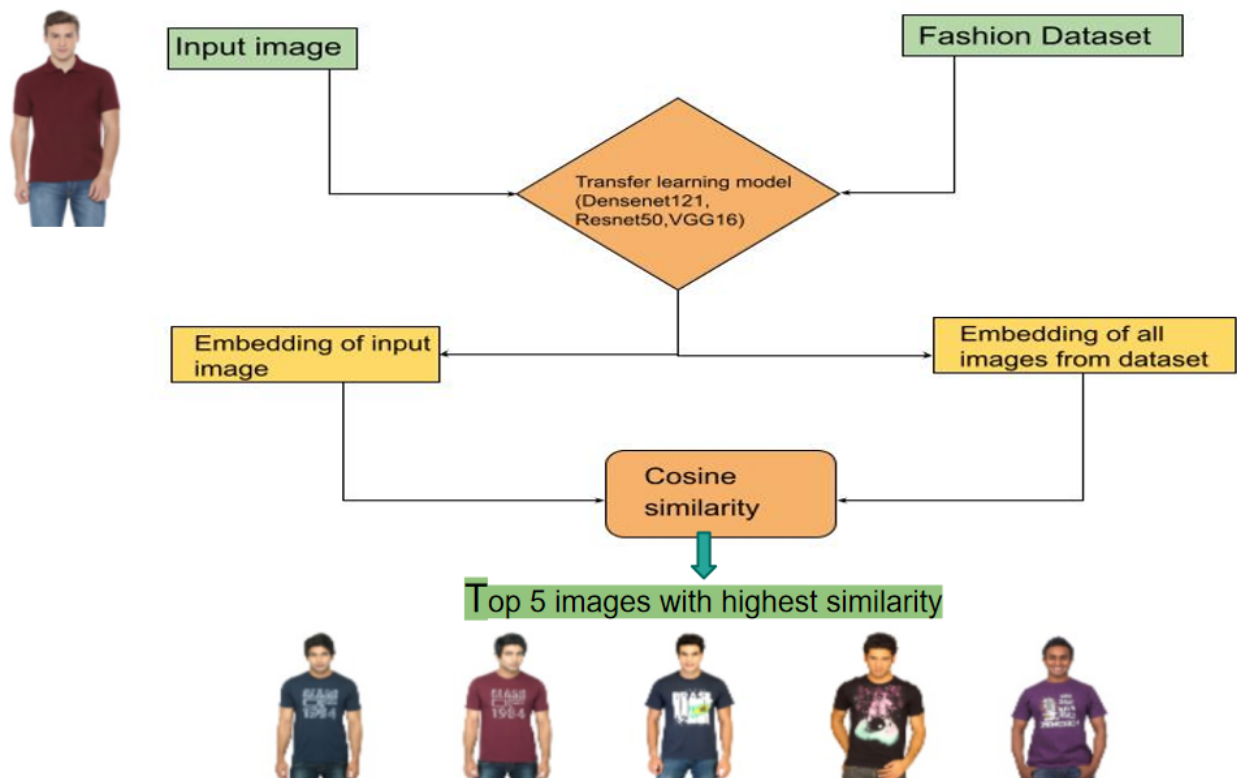
INTRODUCTION

In today's highly competitive world of ecommerce, personalization has become a key factor in driving customer engagement and loyalty. As a result, ecommerce platforms have started using recommendation systems to personalize the shopping experience for their users, increasing user engagement and improving customer satisfaction.

Style Suggest Website, an ecommerce platform focused on fashion, can greatly benefit from implementing a fashion recommender system. By providing users with personalized clothing recommendations, Style Suggest can enhance the shopping experience for its users, increasing engagement, and building customer loyalty.

Please go to the main Page to select any attire and get recommendations for the similar products. This application is used to detect the kind of garment, such as ladies tops, etc., and make correct suggestions for related items. We have used open-source computer vision libraries, such as TensorFlow and OpenCV, to make development and integration with the dataset given to us.

The image dataset given to us consist of ladies clothing. Therefore uploading ladies garments will work for this model. The same model can be used for a huge dataset of garments for any gender.



Hardware & Software Requirements

While performing the task, the technology required and used are:

Technology Used: Python, TensorFlow, Keras, NumPy, Pandas, Scikit-learn, Matplotlib

Software Used: Google Collab, PyCharm, Git and Github

Others: Internet connection and laptop

Hardware Requirements:

- CPU: Intel Core i3 or higher
- RAM: 4GB or more
- NVIDIA® GPU card with CUDA® architectures 3.5, 3.7, 5.2, 6.0, 6.1, 7.0 and higher than 7.0. See the list of CUDA®-enabled GPU cards.
- Latest NVIDIA® GPU drivers. CUDA® 10.1 requires 418.x or higher.
- CUDA® Toolkit 10.1
- cuDNN SDK 7.6
- Storage: 50GB or more (for storing the dataset and trained models)

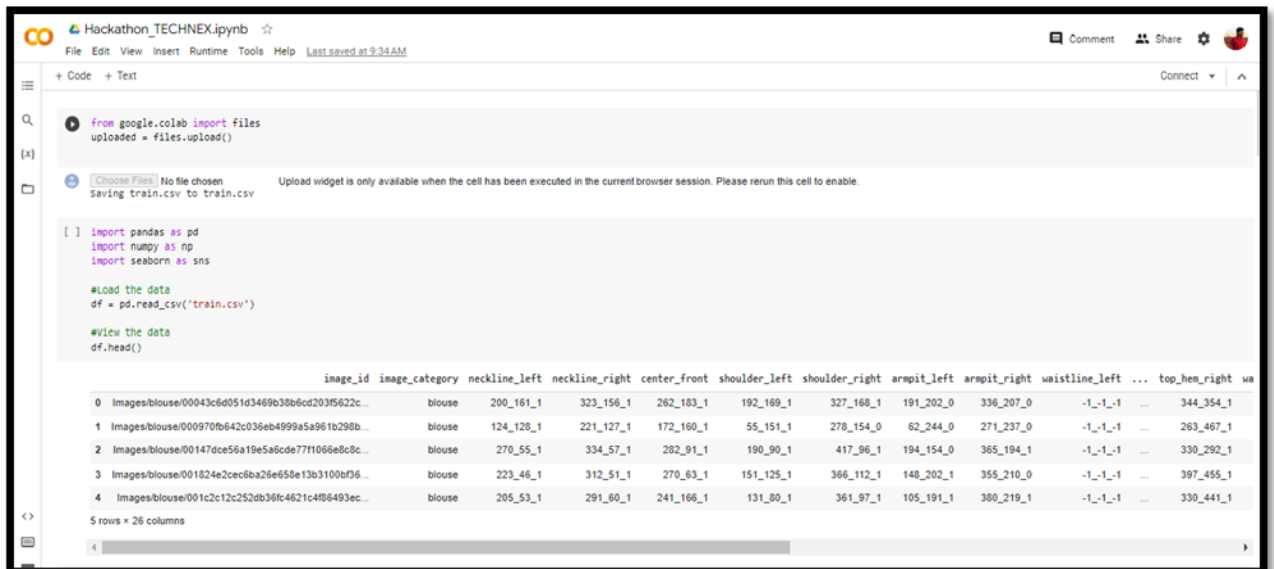
Software Requirements:

- Operating System: Windows, Linux, or macOS
- Python: version 3.7
- TensorFlow: version 2.0 or higher
- Keras: version 2.0 or higher
- NumPy: version 1.18 or higher
- Pandas: version 1.0 or higher
- Scikit-learn: version 0.23 or higher
- Matplotlib: version 3.2 or higher

Software Description

4.1 StyleSuggest Website

1. **Collect Data:** Collect data on users' preferences, such as their past purchase history, search queries, and ratings. It can be collected on the clothing items themselves, such as style, colour, fabric, and brand.



The screenshot shows a Jupyter Notebook interface with the following code and output:

```
from google.colab import files
uploaded = files.upload()

[ ] Choose Files No file chosen
Saving train.csv to train.csv
Upload widget is only available when the cell has been executed in the current browser session. Please rerun this cell to enable.

[ ] import pandas as pd
import numpy as np
import seaborn as sns

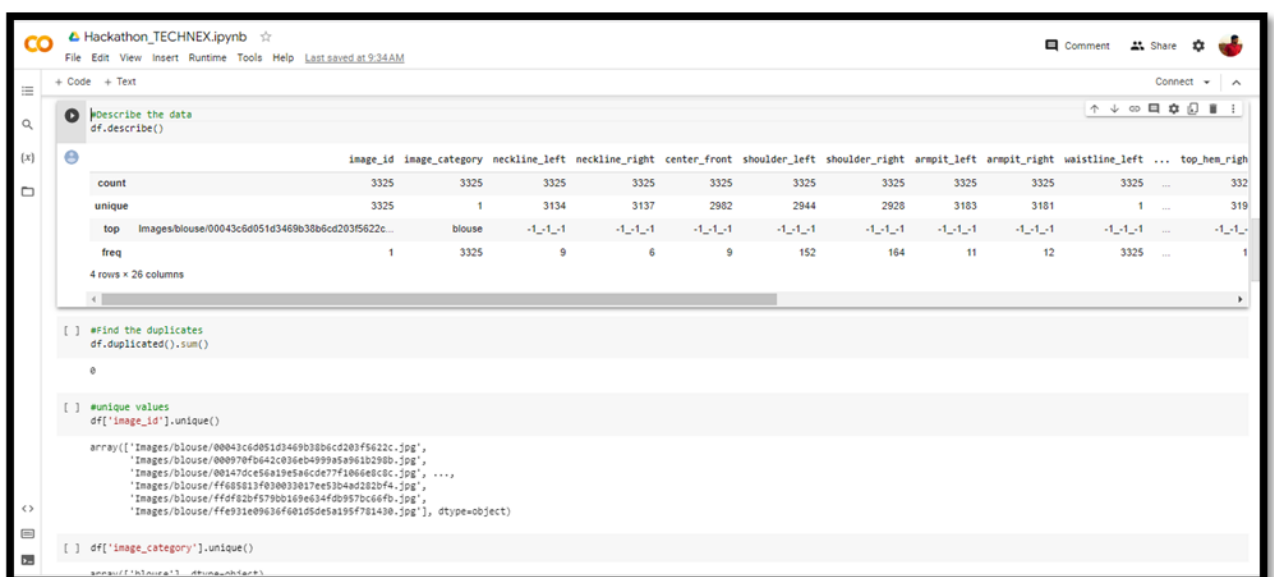
#Load the data
df = pd.read_csv('train.csv')

#View the data
df.head()
```

	image_id	image_category	neckline_left	neckline_right	center_front	shoulder_left	shoulder_right	armpit_left	armpit_right	waistline_left	...	top_hem_right	wa
0	Images/blouse/00043c6d051d3469b38b6cd203f5622c...	blouse	200_161_1	323_156_1	262_183_1	192_169_1	327_168_1	191_202_0	336_207_0	-1_-1_-1	...	344_354_1	
1	Images/blouse/000970fb642c036eb4999a5a961b290b...	blouse	124_128_1	221_127_1	172_160_1	55_151_1	278_154_0	62_244_0	271_237_0	-1_-1_-1	...	263_467_1	
2	Images/blouse/00147dce56a19e5a8cde77f1066e8c8c...	blouse	270_55_1	334_57_1	282_91_1	190_90_1	417_96_1	194_154_0	365_194_1	-1_-1_-1	...	330_292_1	
3	Images/blouse/001824e2cec6ba26e558e13b3100bf36...	blouse	223_46_1	312_51_1	270_63_1	151_125_1	366_112_1	148_202_1	355_210_0	-1_-1_-1	...	397_455_1	
4	Images/blouse/001c2c12c252db36f4621c4f86493ec...	blouse	205_53_1	291_60_1	241_166_1	131_80_1	361_97_1	105_191_1	380_219_1	-1_-1_-1	...	330_441_1	

5 rows x 26 columns

2. **Pre-process Data:** The collected data is then preprocessed and cleaned to remove any inconsistencies, duplicates, or missing values. This step also involve data normalization and feature engineering to transform the data into a format suitable for modelling.



The screenshot shows a Jupyter Notebook interface with the following code and output:

```
#Describe the data
df.describe()

#Find the duplicates
df.duplicated().sum()

#unique values
df['image_id'].unique()

df['image_category'].unique()
```

	image_id	image_category	neckline_left	neckline_right	center_front	shoulder_left	shoulder_right	armpit_left	armpit_right	waistline_left	...	top_hem_right	wa
count	3325	3325	3325	3325	3325	3325	3325	3325	3325	3325	...	3325	
unique	3325	1	3134	3137	2982	2944	2928	3183	3181	1	...	319	
top	Images/blouse/00043c6d051d3469b38b6cd203f5622c...	blouse	-1_-1_-1	-1_-1_-1	-1_-1_-1	-1_-1_-1	-1_-1_-1	-1_-1_-1	-1_-1_-1	-1_-1_-1	...	-1_-1_-1	
freq	1	3325	9	6	9	152	164	11	12	3325	...	1	

4 rows x 26 columns

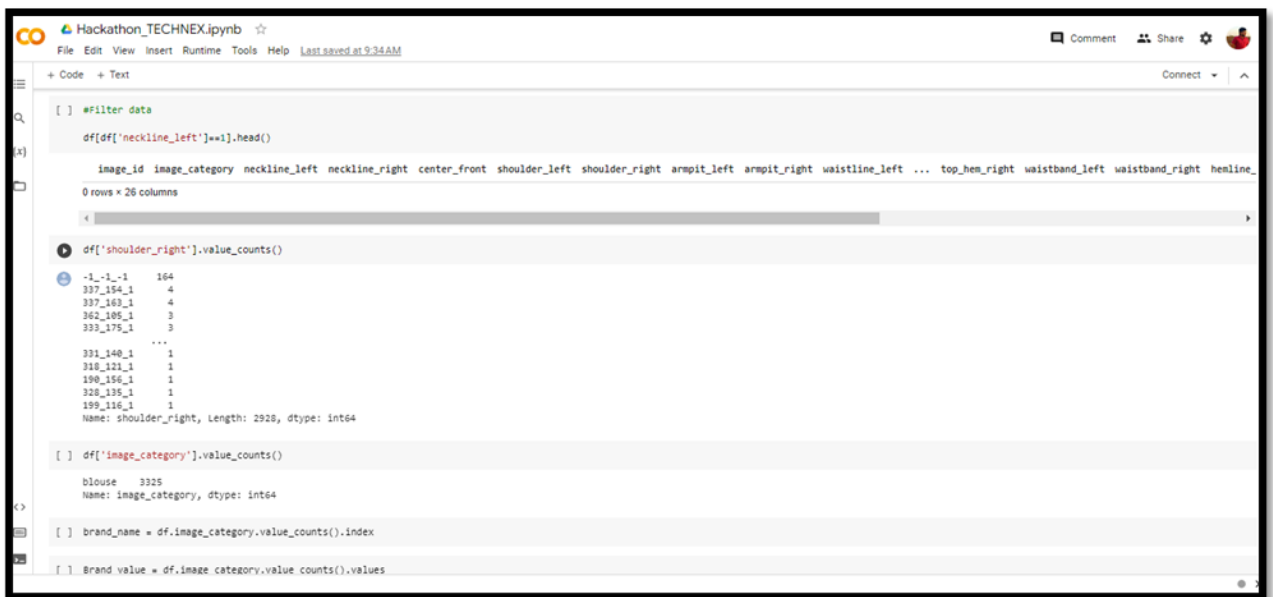
0

array(['Images/blouse/00043c6d051d3469b38b6cd203f5622c.jpg',
 'Images/blouse/000970fb642c036eb4999a5a961b290b.jpg',
 'Images/blouse/00147dce56a19e5a8cde77f1066e8c8c.jpg', ...,
 'Images/blouse/ff685813f030833017ee53b4ad282bf4.jpg',
 'Images/blouse/ffdf82bf5790b169e634fd957bc66fb.jpg',
 'Images/blouse/ffe931e09636f681d5de5a195f781438.jpg'], dtype=object)

array(['blouse'], dtype=object)

3. **Exploratory Data Analysis (EDA):** EDA is performed to gain insights into the data

and identify any patterns, trends, or relationships that may exist.



```
[ ] #Filter data
df[df['neckline_left']==1].head()

image_id image_category neckline_left neckline_right center_front shoulder_left shoulder_right armpit_left armpit_right waistline_left ... top_hem_right waistband_left waistband_right hemline_
0 rows x 26 columns

[ ] df['shoulder_right'].value_counts()

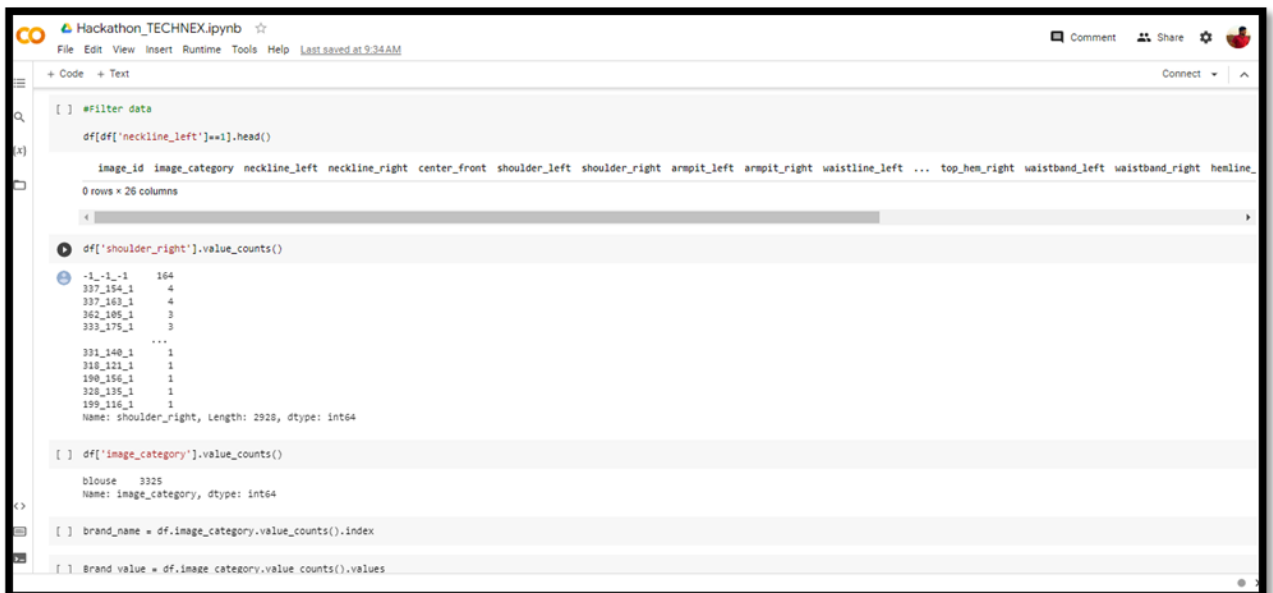
-1_-1_-1 164
337_154_1 4
337_163_1 4
362_105_1 3
333_175_1 3
...
331_140_1 1
318_121_1 1
190_156_1 1
328_135_1 1
199_116_1 1
Name: shoulder_right, Length: 2928, dtype: int64

[ ] df['image_category'].value_counts()

blouse 3325
Name: image_category, dtype: int64

[ ] brand_name = df.image_category.value_counts().index

[ ] Brand value = df.image_category.value_counts().values
```



```
[ ] #Filter data
df[df['neckline_left']==1].head()

image_id image_category neckline_left neckline_right center_front shoulder_left shoulder_right armpit_left armpit_right waistline_left ... top_hem_right waistband_left waistband_right hemline_
0 rows x 26 columns

[ ] df['shoulder_right'].value_counts()

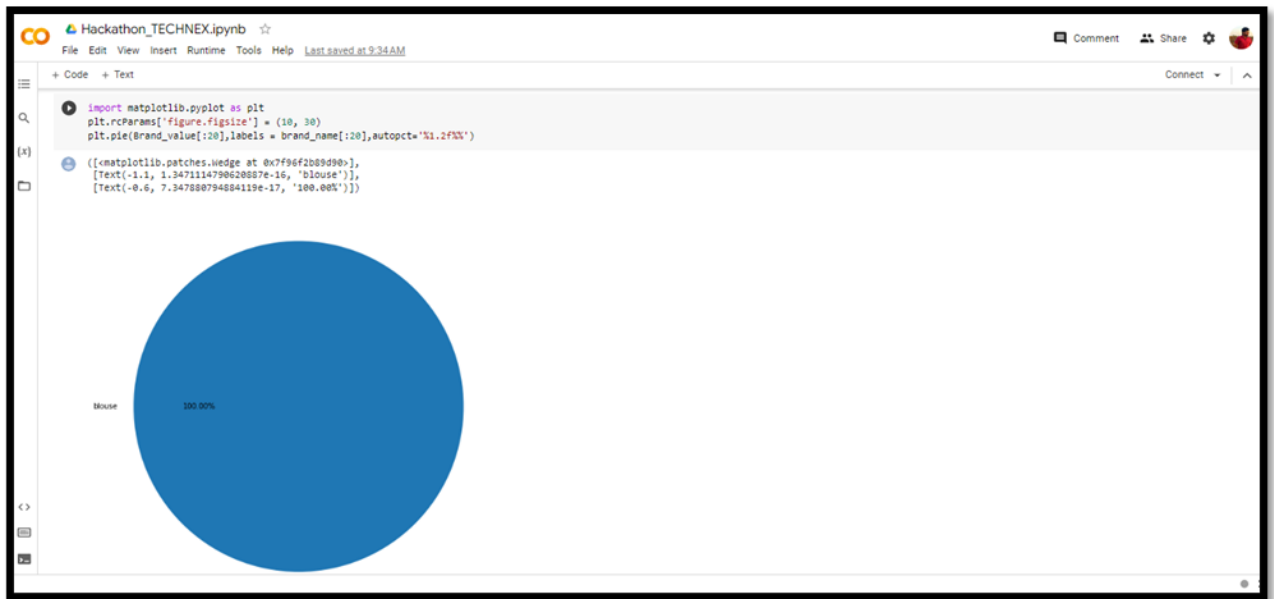
-1_-1_-1 164
337_154_1 4
337_163_1 4
362_105_1 3
333_175_1 3
...
331_140_1 1
318_121_1 1
190_156_1 1
328_135_1 1
199_116_1 1
Name: shoulder_right, Length: 2928, dtype: int64

[ ] df['image_category'].value_counts()

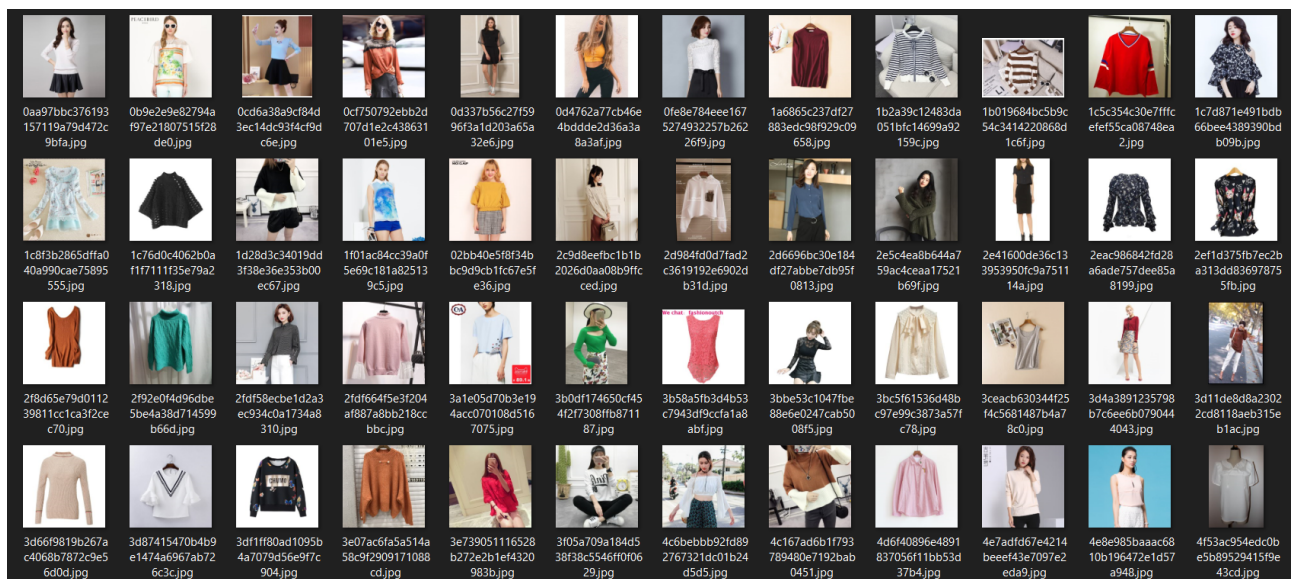
blouse 3325
Name: image_category, dtype: int64

[ ] brand_name = df.image_category.value_counts().index

[ ] Brand value = df.image_category.value_counts().values
```

4. Import Images Dataset given by the company: Women fashion Images dataset is used in our model which as given to us.



5. Import model: A CNN model is loaded named resNet. It is build in keras module. This is used as it is high performing CNN model which is trained on a ImageNet dataset which has high accuracy and perfectly matches to the requirement.

```

1 import tensorflow
2 from tensorflow.keras.preprocessing import image
3 from tensorflow.keras.layers import GlobalMaxPooling2D
4 from tensorflow.keras.applications.resnet50 import ResNet50, preprocess_input
5 import numpy as np
6 from numpy.linalg import norm
7 import os
8 from tqdm import tqdm
9 import pickle
10
11 model = ResNet50(weights='imagenet', include_top=False, input_shape=(224, 224, 3))
12 model.trainable = False
13
14 model = tensorflow.keras.Sequential([
15     model,
16     GlobalMaxPooling2D()
17 ])
18

```

6. **Extract features:** We have a dataset for 373 images. Generally the image has resolution of 224,224,3. A CNN model consist of various layers and the image flow wothing those layers pixel by pixel. going through the layers the model see complex features. ResNet, short for Residual Neural Network, is a deep learning architecture that is commonly used for image recognition tasks. In the STYLE SUGGEST website ResNet is used as a feature extraction method to extract features from clothing images. For every image that goes to Resnet model, it given 2048 features. So there will be a total of [373,2048] array. On the basis of this array, it will compare the vector of the input image features with the model array.

```

def extract_features(img_path,model):
    img = image.load_img(img_path,target_size=(224,224))
    img_array = image.img_to_array(img)
    expanded_img_array = np.expand_dims(img_array, axis=0)
    preprocessed_img = preprocess_input(expanded_img_array)
    result = model.predict(preprocessed_img).flatten()
    normalized_result = result / norm(result)

    return normalized_result

filenames = []

for file in os.listdir('images'):
    filenames.append(os.path.join('images',file))

feature_list = []

for file in tqdm(filenames):
    feature_list.append(extract_features(file,model))

pickle.dump(feature_list,open('embeddings.pkl','wb'))
pickle.dump(filenames,open('filenames.pkl','wb'))

```

- 7. Generate Recommendations:** The extracted features are then used to improve the recommendations provided by the hybrid filtering approach. Specifically, the extracted features are combined with the user-item interaction data to form a hybrid feature representation of the user-item interaction matrix using scikit learn. This hybrid feature representation is then used in the matrix factorization algorithm to enhance the recommendations provided by the collaborative filtering approach. The euclidian distance is used between two vectors and the five closest neighbours are being identified.

```
def recommend(features, feature_list):
    neighbors = NearestNeighbors(n_neighbors=6, algorithm='brute', metric='euclidean')
    neighbors.fit(feature_list)

    distances, indices = neighbors.kneighbors([features])

    return indices

# steps
# file upload -> save
uploaded_file = st.file_uploader("Choose an image")
```

```
if uploaded_file is not None:
    if save_uploaded_file(uploaded_file):
        # display the file
        display_image = Image.open(uploaded_file)
        st.image(display_image)
        st.header('Here are the recommended similar Clothing suggestions')

        # feature extract
        features = feature_extraction('/uploads/uploaded_file.name', model)
        #st.text(features)
        # recommendation
        indices = recommend(features, feature_list)
        # show
        col1, col2, col3, col4, col5 = st.columns(5)
        with col1:
            st.image(filenames[indices[0][0]])
        with col2:
            st.image(filenames[indices[0][1]])
        with col3:
            st.image(filenames[indices[0][2]])
        with col4:
            st.image(filenames[indices[0][3]])
        with col5:
            st.image(filenames[indices[0][4]])
    else:
        st.header("Some error occured in file upload")
```

In addition to feature extraction, TensorFlow is also used to train and evaluate other machine learning models used in the fashion recommender system, such as the matrix factorization algorithm. TensorFlow provides a flexible and powerful framework for

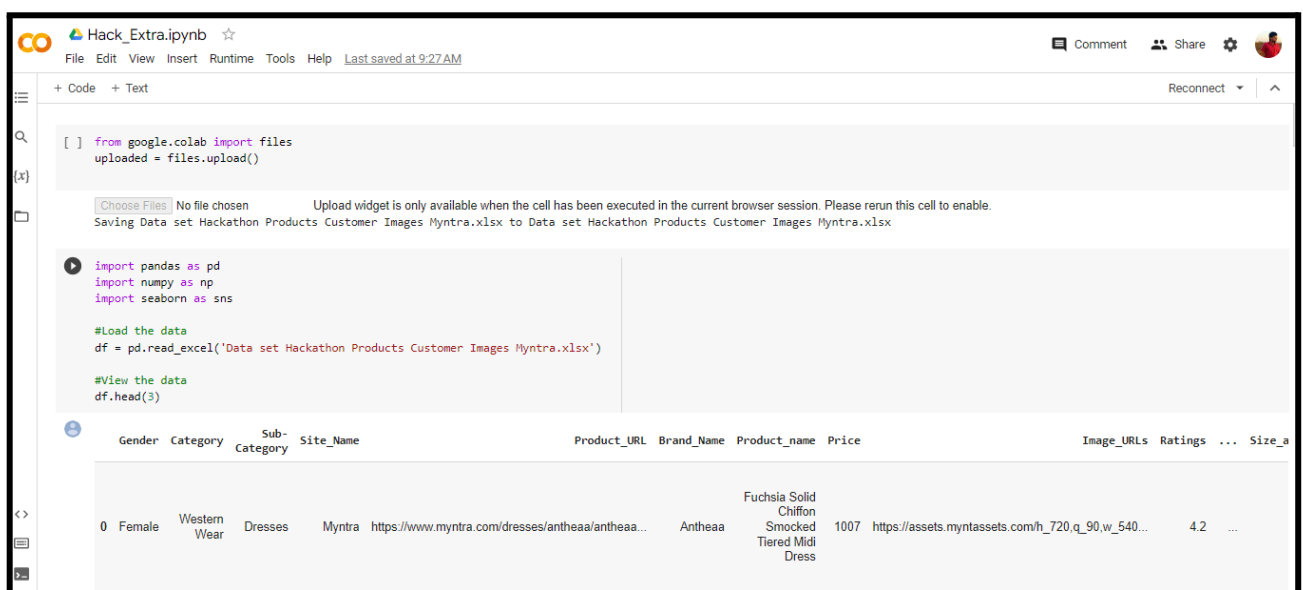
building and training machine learning models, making it an ideal choice for developing the complex recommendation system used by StyleSuggest.

Link for the Github Repository is - <https://github.com/devmitanshu/StyleSuggest>

Link for the deployed version is -

<https://devmitanshu-stylesuggest-homepage-e4bm07.streamlit.app/>

4.2 EDA of Clothing Dataset



The screenshot shows a Jupyter Notebook titled 'Hack_Extra.ipynb'. The code cell contains the following Python code:

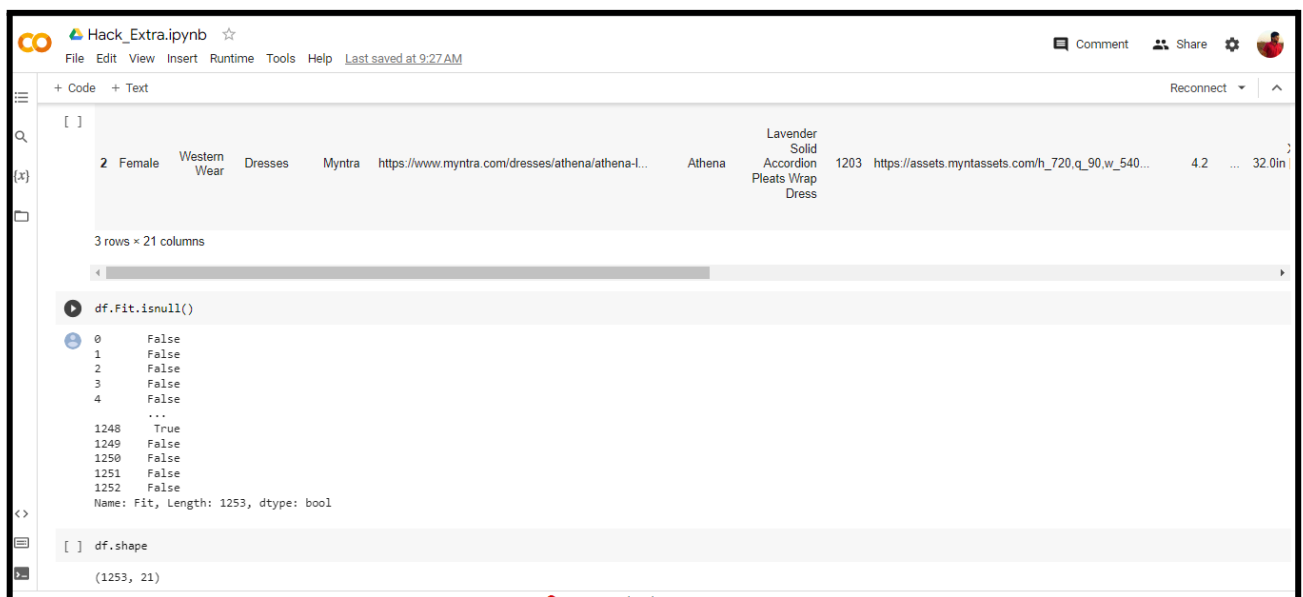
```
[ ] from google.colab import files
    uploaded = files.upload()

import pandas as pd
import numpy as np
import seaborn as sns

#Load the data
df = pd.read_excel('Data set Hackathon Products Customer Images Myntra.xlsx')

#View the data
df.head(3)
```

The output of the code is a preview of the dataset, showing the first three rows of a DataFrame. The columns are: Gender, Category, Sub-Category, Site_Name, Product_URL, Brand_Name, Product_name, Price, Image_URLs, Ratings, and Size_a. The first row shows a Female, Western Wear, Dresses, Myntra, with a price of 1007 and a rating of 4.2.



The screenshot shows the same Jupyter Notebook with the following code cell:

```
[ ] df.isnull()

df.shape
```

The output of the first code cell shows the result of the `df.isnull()` operation, indicating that there are no null values in the dataset. The output of the second code cell shows the shape of the dataset as (1253, 21).

```
Hack_Extra.ipynb
File Edit View Insert Runtime Tools Help Last saved at 9:27 AM

+ Code + Text
Reconnect ^

[ ] df.isnull().sum().sum()

3148

#Basic information
df.info()

<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1253 entries, 0 to 1252
Data columns (total 21 columns):
#   Column                Non-Null Count  Dtype
---  --
0   Gender                 1253 non-null  object
1   Category               1253 non-null  object
2   Sub-Category           1253 non-null  object
3   Site_Name              1253 non-null  object
4   Product_URL            1253 non-null  object
5   Brand_Name             1253 non-null  object
6   Product_name           1253 non-null  object
7   Price                  1253 non-null  object
8   Image_URLs             1253 non-null  object
9   Ratings                1044 non-null   float64
10  Number_of_Ratings      1044 non-null   object
11  Size_and_Dimensions    1098 non-null   object
12  Available_Colors       453 non-null   object
13  Product_Details        1253 non-null   object
14  Fit_and_Size           1149 non-null   object
15  Material_and_Care      1251 non-null   object
16  Specifications          1244 non-null   object
17  Fit                    919 non-null    object

0s completed at 8:44PM
```

```
Hack_Extra.ipynb
File Edit View Insert Runtime Tools Help Last saved at 9:27 AM

+ Code + Text
Reconnect ^

[ ] df.describe()

Ratings
count    1044.000000
mean      4.159962
std       0.346668
min       2.100000
25%       4.000000
50%       4.200000
75%       4.400000
max       5.000000

df.iloc[:1]

Gender  Category  Sub-Category  Site_Name  Product_URL  Brand_Name  Product_name  Price  Image_URLs  Ratings  ...  Size_and_Dimensions  Available_
0  Female  Western  Dresses  Myntra  https://www.myntra.com/dresses/antheaa/antheaa...  Antheaa  Fuchsia Solid Chiffon Smocked Tiered Midi Dress  1007  https://assets.myntrassets.com/h_720,q_90,w_540...  4.2  ...  NaN

1 rows x 21 columns

0s completed at 8:44PM
```

```
Hack_Extra.ipynb
File Edit View Insert Runtime Tools Help Last saved at 9:27 AM

+ Code + Text
Reconnect ^

[ ] df["Sub-Category"].unique()

array(['Dresses'], dtype=object)

#Find null values
df.isnull().sum()

Gender                0
Category              0
Sub-Category          0
Site_Name             0
Product_URL           0
Brand_Name            0
Product_name          0
Price                 0
Image_URLs            0
Ratings              209
Number_of_Ratings     209
Size_and_Dimensions   155
Available_Colors      800
Product_Details       0
Fit_and_Size          104
Material_and_Care      2
Specifications         9
Fit                   334
Length                334
Transparency           334
Customer_Images       658
dtype: int64

#Datatypes
df.dtypes

Gender                object
Category              object

0s completed at 8:44PM
```

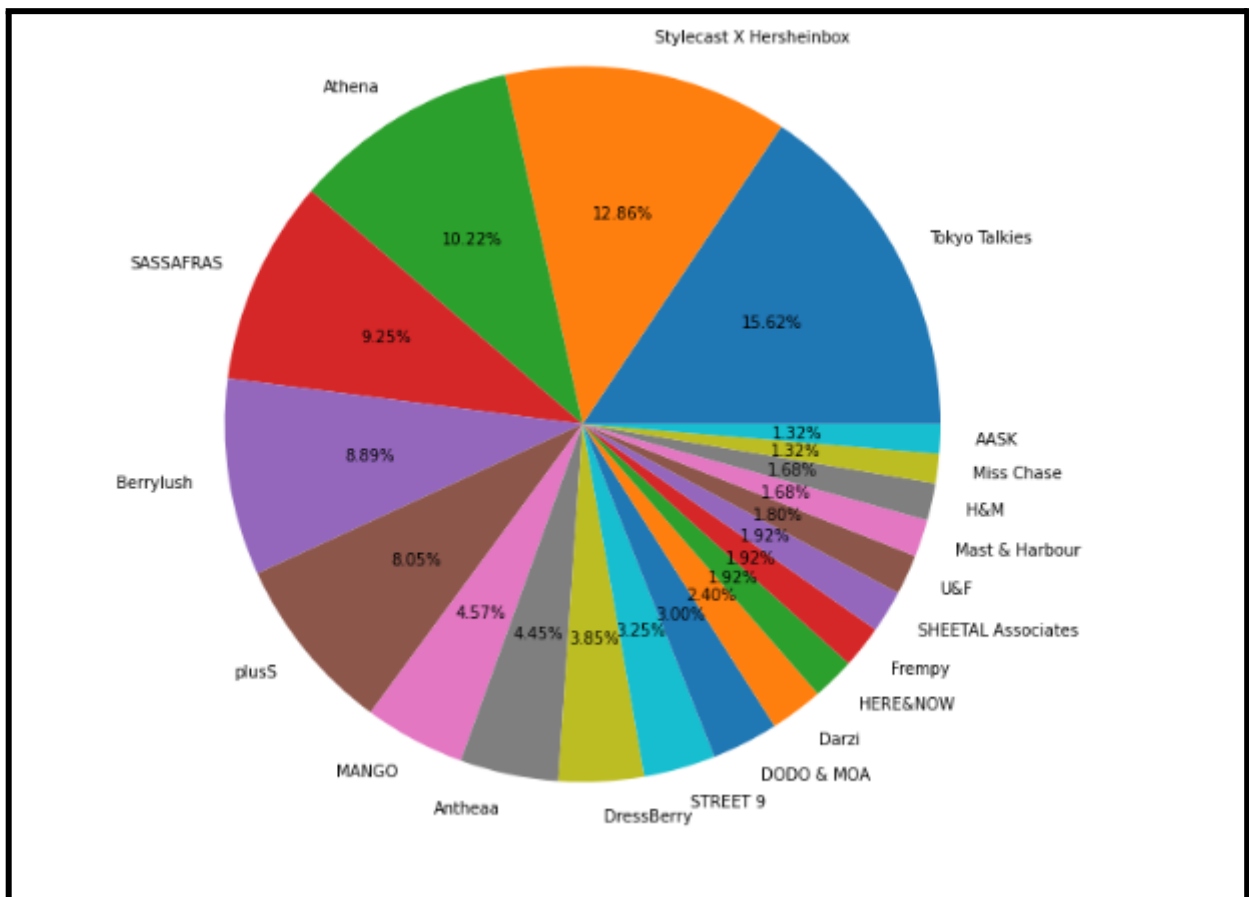
```

Hack_Extra.ipynb
File Edit View Insert Runtime Tools Help Last saved at 9:27AM
+ Code + Text
Reconnect
brand_name = df.Brand_Name.value_counts().index
Brand_value = df.Brand_Name.value_counts().values

import matplotlib.pyplot as plt
plt.rcParams['figure.figsize'] = (10, 30)
plt.pie(Brand_value[:20], labels = brand_name[:20], autopct='%1.2f%%')

Text(-0.518536399155026, -0.9701133968518009, 'MANGO'),
Text(-0.2268065365438117, -1.07636369850356, 'Antheaa'),
Text(0.055122594939276635, -1.0884633648681803, 'DressBerry'),
Text(0.2993839489147616, -1.0584749670740194, 'STREET 9'),
Text(0.58012958461079, -0.9797297579419739, 'DODO & MOA'),
Text(0.6586807242190917, -0.8810477206474621, 'Darzi'),
Text(0.771921348922951, -0.78366921119299, 'HERE&NOW'),
Text(0.6607540542932466, -0.6849105474569208, 'Frempy'),
Text(0.9370350343685776, -0.5761643379851611, 'SHEETAL Associates'),
Text(0.9979115850091957, -0.4627877142972087, 'U&F'),
Text(1.0425899262310102, -0.35896018821201574, 'Mast & Harbour'),
Text(1.0737254610289995, -0.23898459016861923, 'H&M'),
Text(1.0914713278126373, -0.13671262094983644, 'Miss Chase'),
Text(1.0990512801441037, -0.04567585372609025, 'AASK'),
[Text(0.529152758609013, 0.2828380420955986, '15.62%'),
Text(0.1103807098554754, 0.5897593567649312, '12.86%'),
Text(-0.308461646515933, 0.5146371660001633, '10.22%'),
Text(-0.5480658284430029, 0.2441799494083921, '9.25%'),
Text(-0.5931692460241356, -0.09027071850894856, '8.89%'),
Text(-0.46524022218619043, -0.37888195478294306, '8.05%'),
Text(-0.2828380359027414, -0.529152761919164, '4.57%'),
Text(-0.12371265629662455, -0.5871074677365578, '4.45%'),
Text(0.031703233603241794, -0.5991618353826438, '3.85%'),
Text(0.16330033140805175, -0.5773499820408742, '3.25%'),
Text(0.2727979552422491, -0.5342980497065312, '3.00%'),
Text(0.359236758664955, -0.48057143388952465, '2.40%'),
Text(0.48057143388952465, -0.359236758664955, '1.92%'),
Text(0.5773499820408742, -0.16330033140805175, '1.92%'),
Text(0.5991618353826438, 0.031703233603241794, '1.92%'),
Text(0.5871074677365578, 0.12371265629662455, '1.80%'),
Text(0.529152761919164, 0.2828380359027414, '1.68%'),
Text(0.46524022218619043, 0.46524022218619043, '1.32%'),
Text(0.37888195478294306, 0.46524022218619043, '1.32%')],
]

```



Conclusion: -

EDA is to find more insights from the datasets. By performing the EDA in the above dataset we can find that most of the people buy the clothes from the top brands like Tokyo talkies, Style cast X Heresheinbox, Athena, SASSAFRAS, and etc.

Average Price Range = 1037

Max Price = 5994

Min Price = 251

Testing and Results



Here are the recommended similar Clothing suggestions





Here are the recommended similar Clothing suggestions





Here are the recommended similar Clothing suggestions



CONCLUSION

In conclusion, the Style Suggest fashion recommender system project is an innovative and useful application that provides personalized fashion recommendations to users. The project utilizes a variety of algorithms, including deep learning models such as ResNet and TensorFlow, to develop accurate and efficient recommendations based on user preferences.

The project's methodology involves several important steps, including data collection, preprocessing, EDA, modeling, evaluation, and implementation. The data is preprocessed and cleaned to remove any inconsistencies, and EDA is used to gain insights into the data and identify any patterns or trends that exist. Various machine learning models are then developed, evaluated, and fine-tuned to improve their performance.

Overall, the project's success is a testament to the power of machine learning and deep learning algorithms in developing personalized and efficient recommendation systems. The application of such technology to the fashion industry is a great example of how technology can be used to enhance user experiences and streamline e-commerce operations.

REFERENCES

- <https://medium.com/@arsanatladkat/how-to-setup-tensorflow-2-3-1-cpu-gpu-windows-10-e000e7811e2b>
- <https://towardsdatascience.com/understand-and-implement-resnet-50-with-tensorflow-2-0-1190b9b52691>