Trip recommendation system

Edoardo Berardi Vittur
Ali Yassine
Ethan Greene

TABLE OF CONTENTS

Destinations opportunities:

Recommendation systems based on:

- Generic personal information
- Money restrictions
- Dates and duration

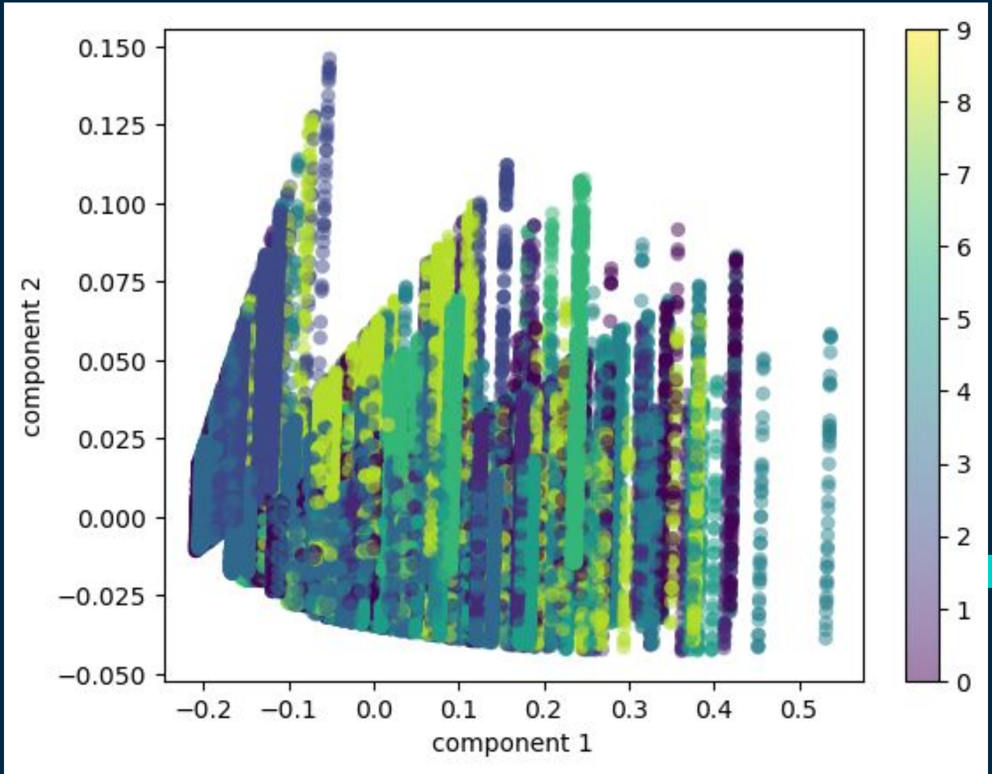
	from	flightType	price_flight	days	price_hotel_x_day	gender	age	month
0	Recife (PE)	1	1434.38	4	313.02	male	21	9

How the data look like?

Principal Component analysis

Dimensionality reduction to plot the data in 2 dimensions.

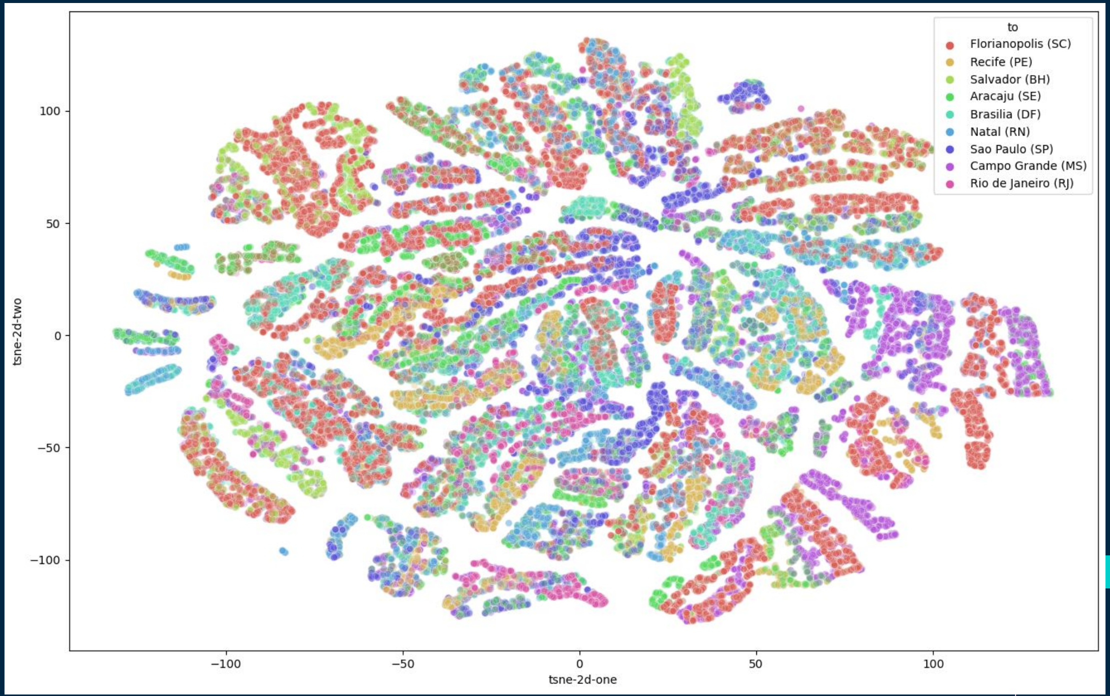
- Clusters follow a pattern but are not recognizable from this graph.



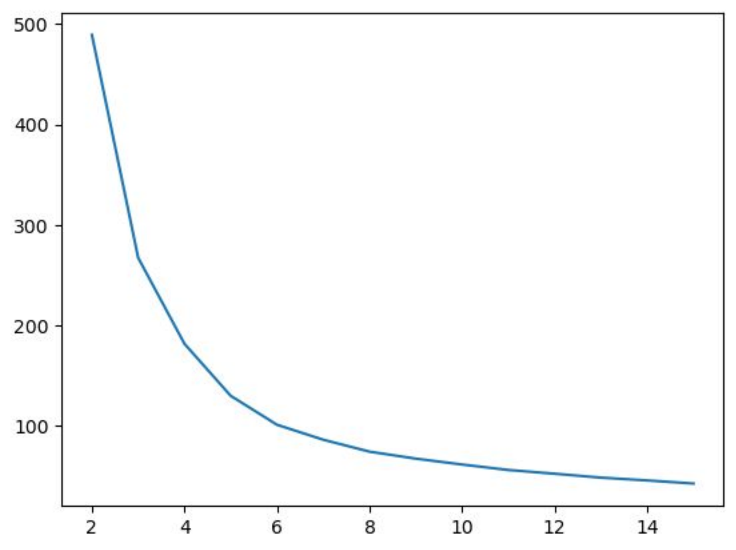
How the data looks like?

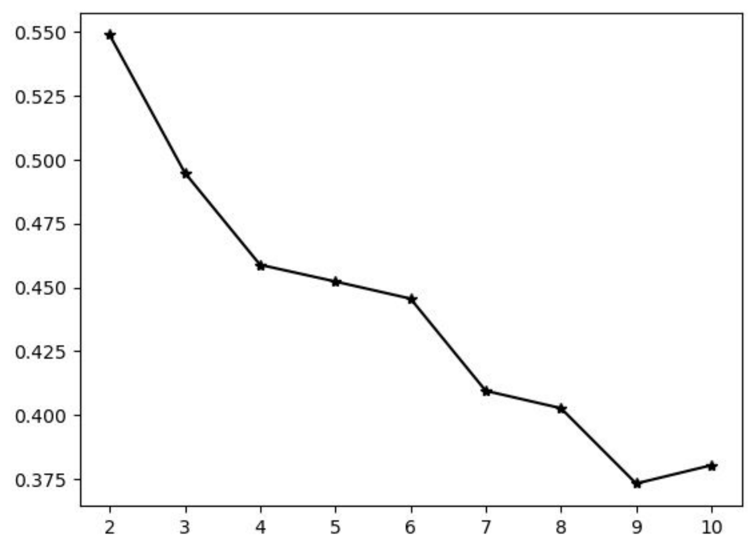
Many points are also grouped correctly in two dimensions --> similarity of data for certain destinations.

T-SNE



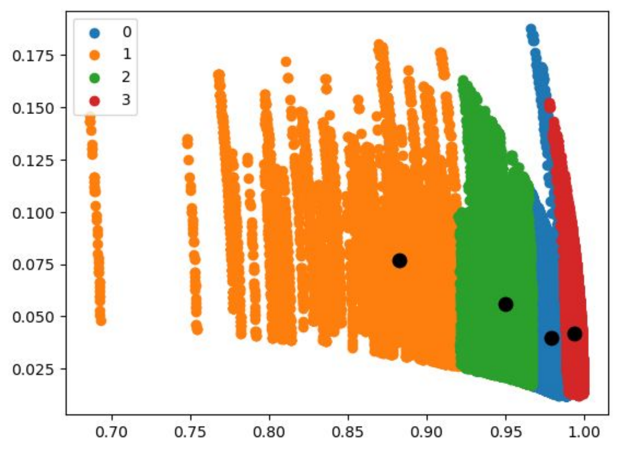
Number of clusters for Kmean:





Destination results

	from	fligthType	price_flight	days	price_hotel_x_day	gender	age	month
0	Rio de Janeiro (RJ)	economic	300	5	80	male	25	9



Florianopolis (SC)	4646
Brasilia (DF)	3938
Recife (PE)	3050
Campo Grande (MS)	2573
Sao Paulo (SP)	2349

Hotel Recommendation System

- Imports
- Functions
- EDA
  - Cleaning
  - Reshaping
  - Visualizing
    - KMeans-Elbow Method
    - KMeans_Silhouette Method
      - K=2
      - K=2 clusters visualized
      - K=3
      - K=3 clusters visualized
    - Bar Graphs
- User Input

- Algorithms
  - Cosine Simlarity
    - Recommended(Acc to description)
    - Recommendation(Acc to description+scores)
  - Dimension Reduction-PCA
  - **Final Result**

Cleaning +Reshaping

```
df1.Hotel_Address[1]
```

```
' s Gravesandestraat 55 Oost 1092 AA Amsterdam Netherlands'
```

- Parsed the address to extract country and city and create a column for each

Cleaning +Reshaping

- New column called 'review' that is the difference between total number of words in negative comments and that in positive comments.

Review_Total_Negative_Word_Counts	Total_Number_of_Reviews	Positive_Review	Review_Total_Positive_Word_Counts
397	0.817947	Only the park outside of the hotel was beauti...	11
0	0.817947	No real complaints the hotel was great great ...	105
42	0.817947	Location was good and staff were ok It is cut...	21

Cleaning +Reshaping

- Rescale all numeric column (0->10)
- Grouped by Hotel Name to get one row for each unique hotel
- Added a contract price column

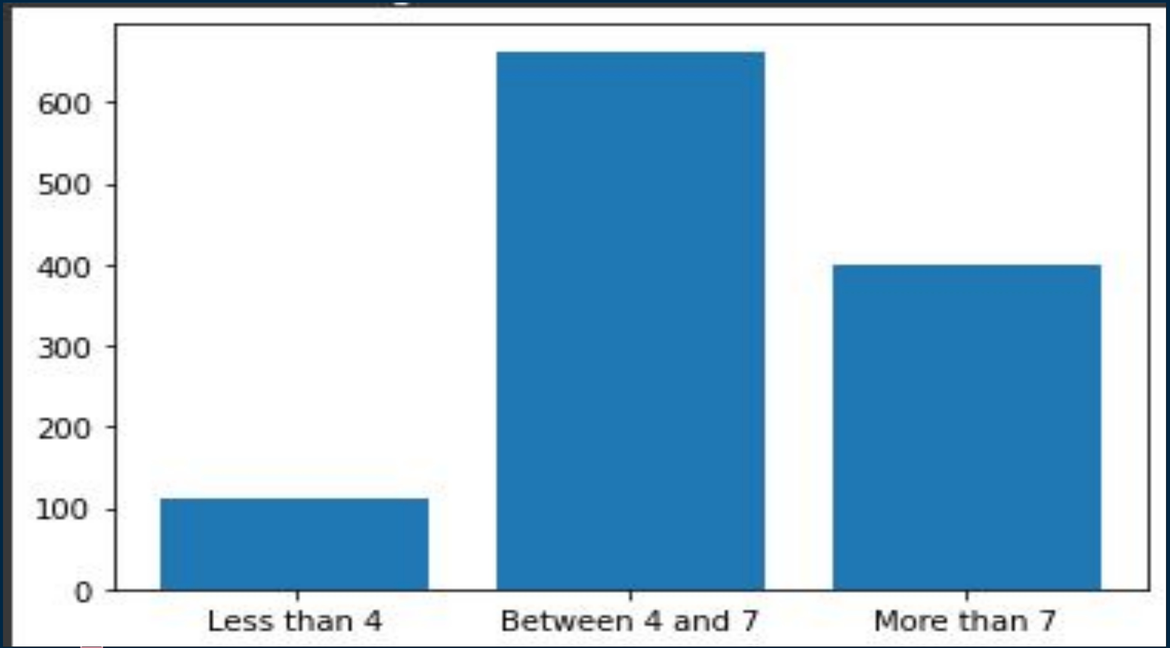
```
[34] price_list=[10000,20000,30000,40000,50000]

[35] contract_price=[]
     for i in range(len(data)):
       num=random.choice(price_list)
       contract_price.append(num)

[36] data['Contract_Price']=contract_price
```

Final Data

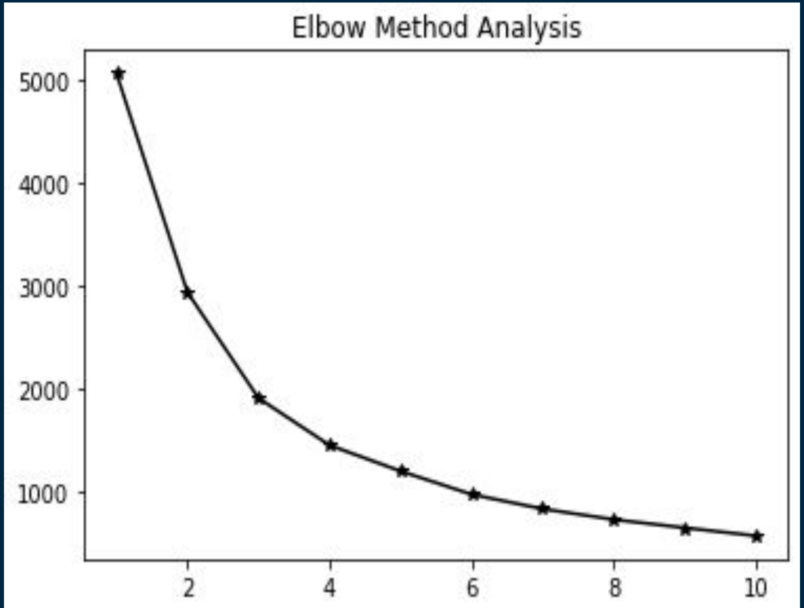
```
[39] data.head()
```

	Hotel_Name	country	city	review	Average_Score	Total_Number_of_Reviews	Tags	Contract_Price
0	11 Cadogan Gardens	Kingdom	United	5.171076	7.608696	0.210501	Leisure trip Couple Superior Queen Room Staye...	50000
1	1K Hotel	France	Paris	4.995788	5.434783	0.372887	Leisure trip Couple Superior M Double Room St...	50000
2	25hours Hotel beim MuseumsQuartier	Austria	Vienna	5.187664	7.826087	2.574728	Leisure trip Solo traveler Standard Double Ro...	10000
3	41	Kingdom	United	5.323378	9.565217	0.120888	Leisure trip Couple Executive King Room with ...	20000
4	45 Park Lane Dorchester Collection	Kingdom	United	5.175391	9.130435	0.015036	Leisure trip Solo traveler Executive Queen Ro...	20000

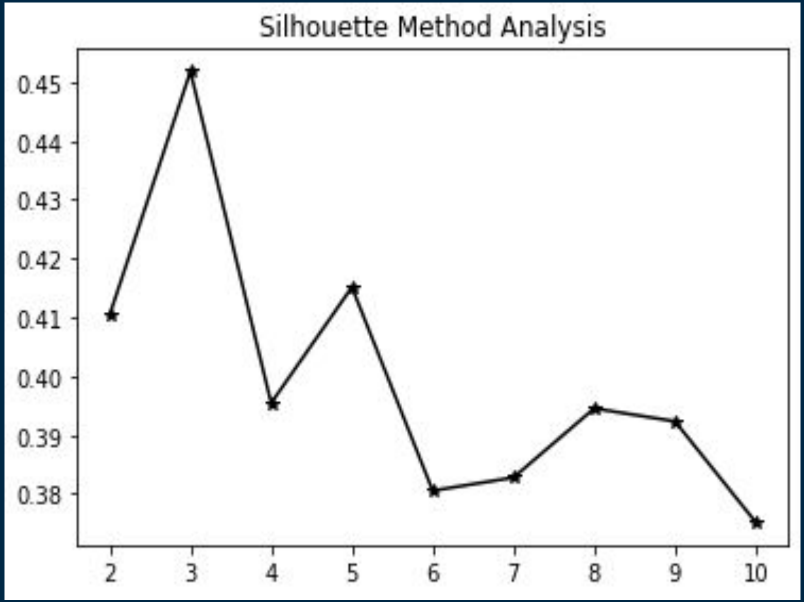
Investigating on the data



Clustering

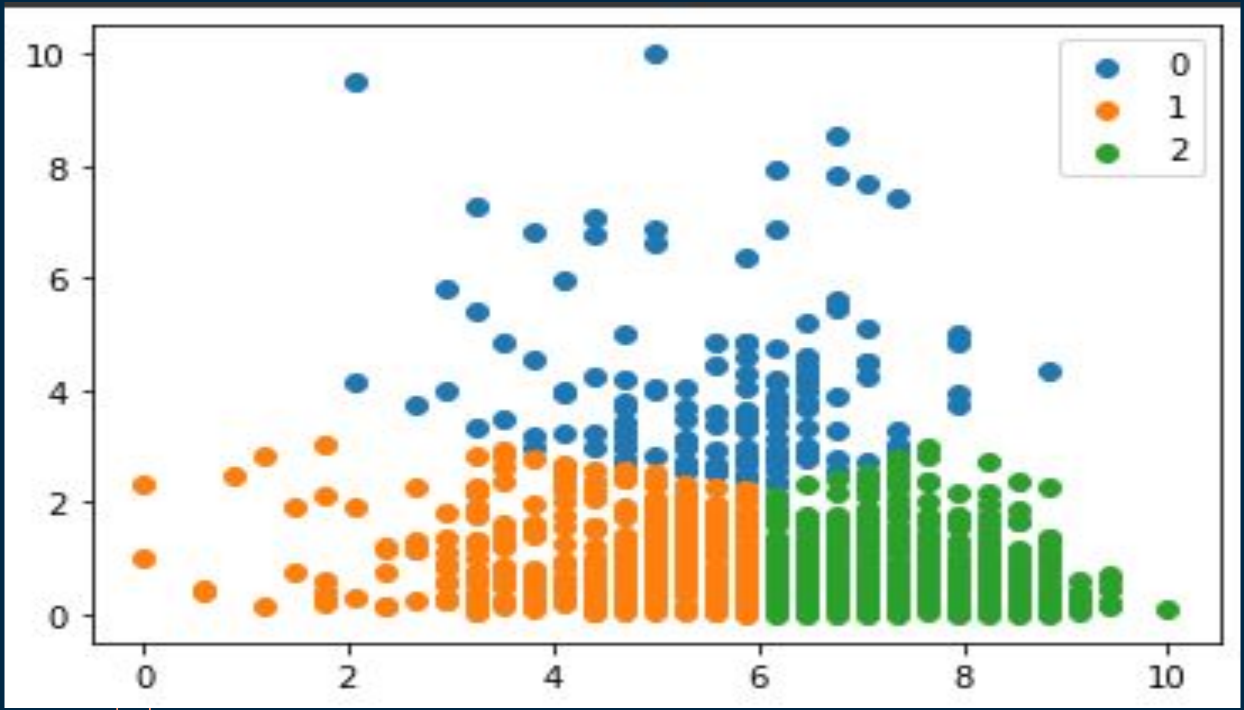


Elbow Method Analysis
5000
4000
3000
2000
1000
2
4
6
8
10



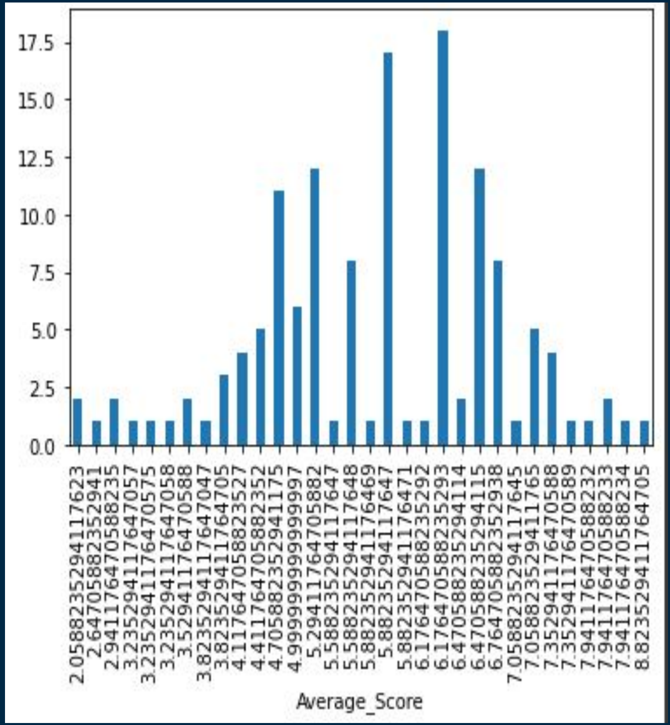
Silhouette Method Analysis
0.45
0.44
0.43
0.42
0.41
0.40
0.39
0.38
2
3
4
5
6
7
8
9
10

Clustering K=3

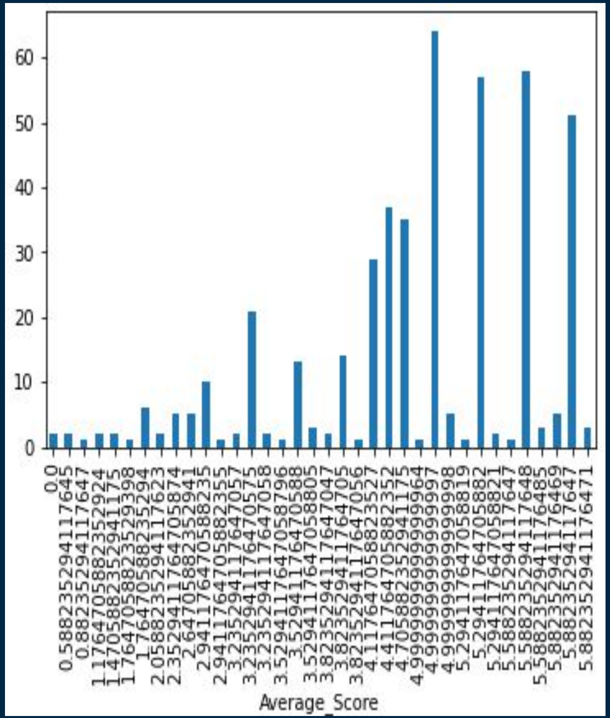


Clustering K=3

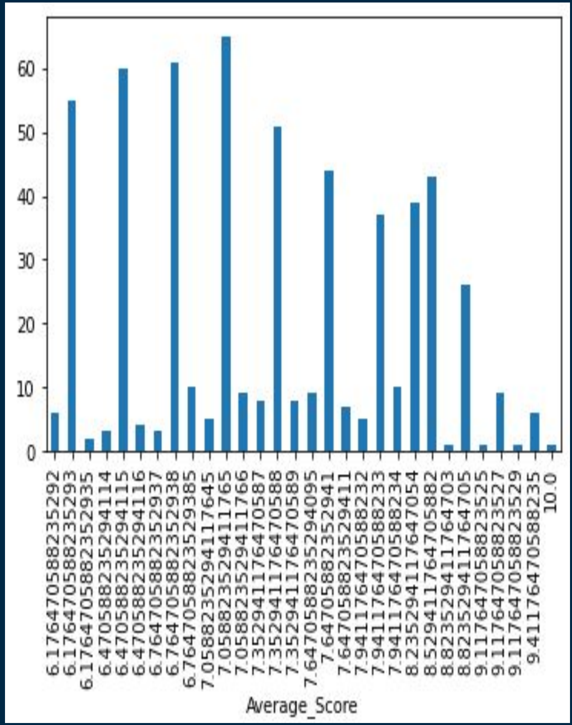
Label 0 (3->7)



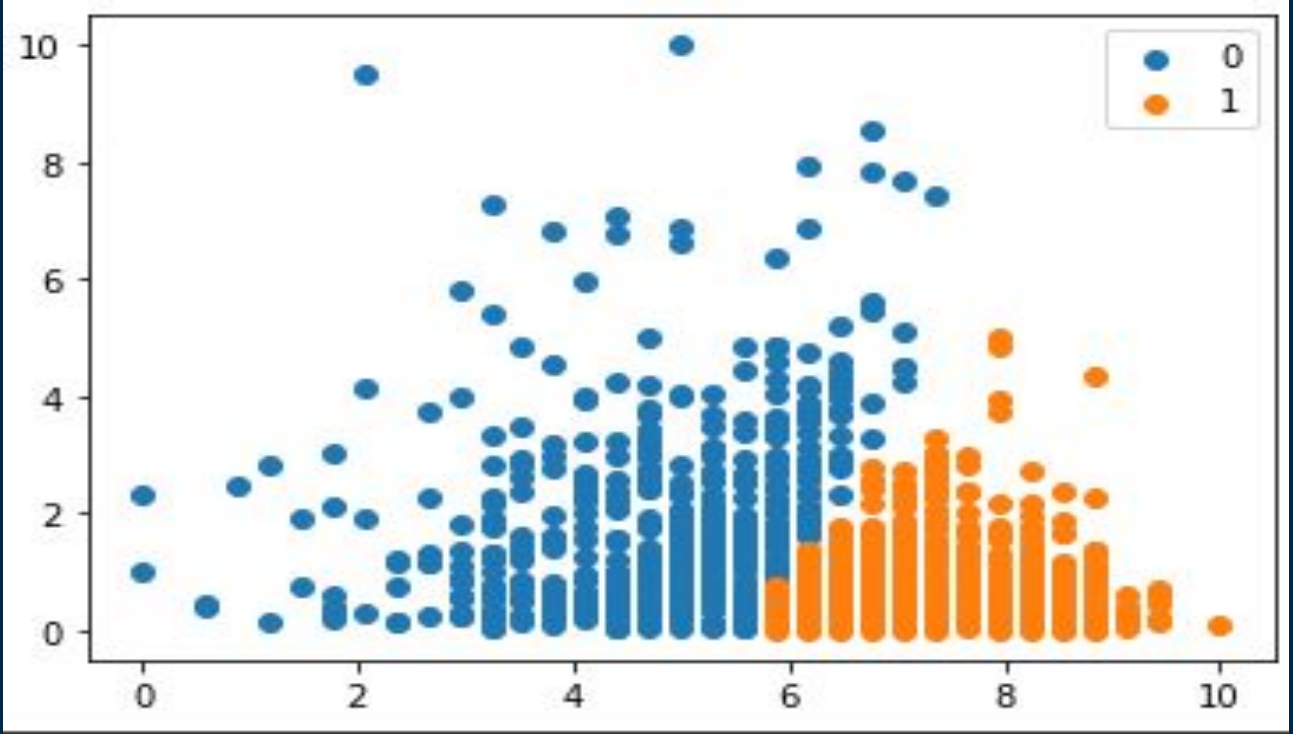
Label 1 (3->5)



Label 2 (6->9)

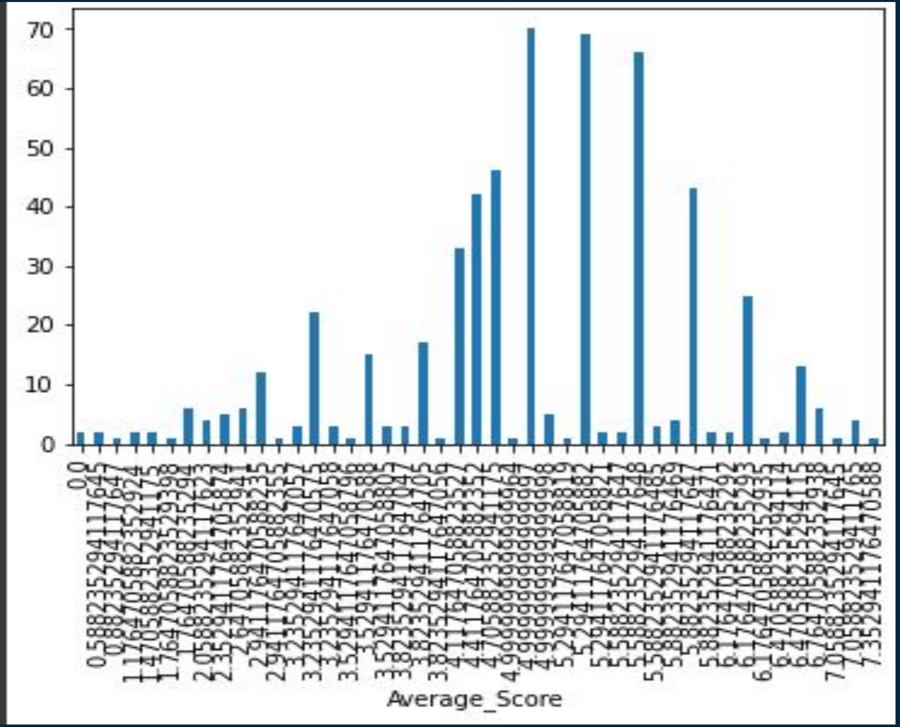


Clustering K=2

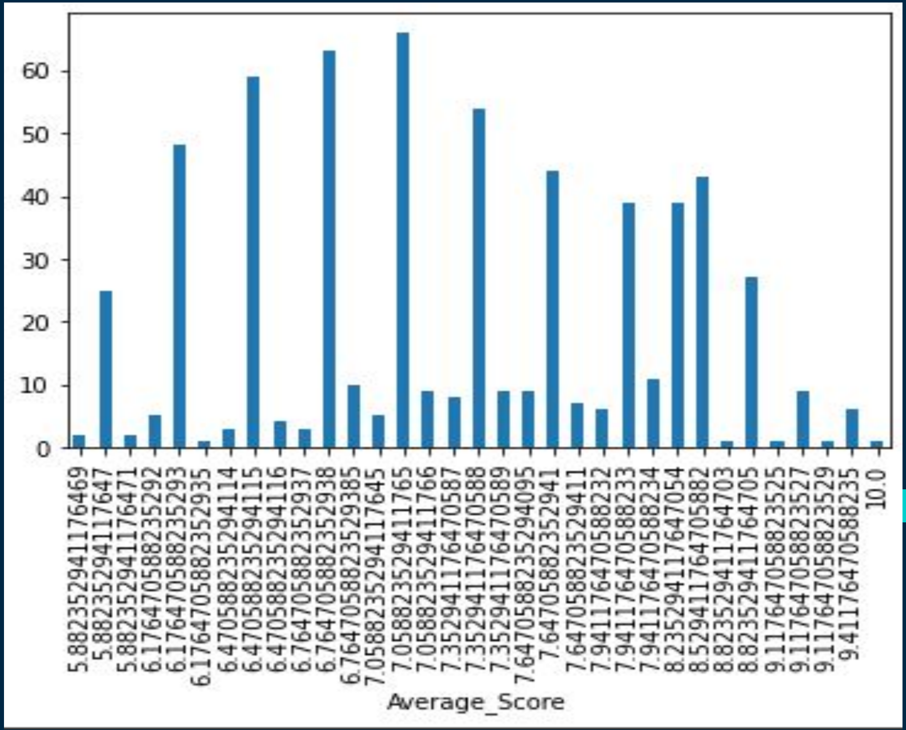


Clustering K=2

Label 0 (3->6)



Label 1(6->9)



User Input

User Input

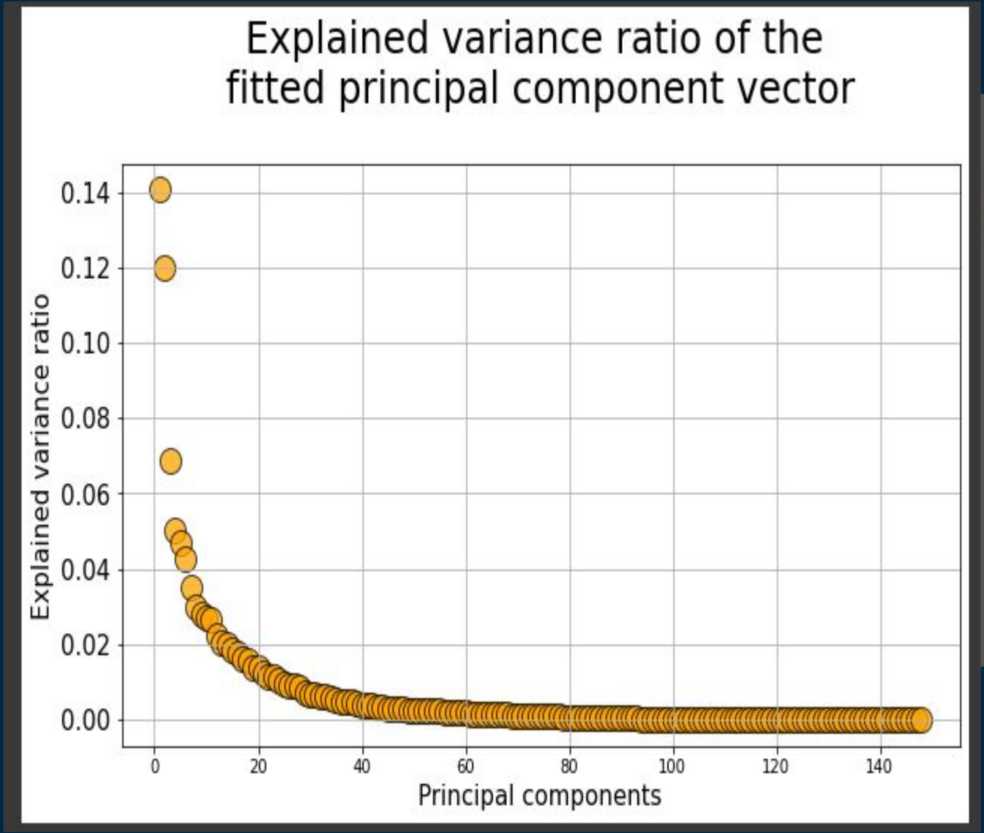
```
[ ] add_user_input(data,"austria","vienna","business")
```

	Hotel_Name	country	city	review	Average_Score	Total_Number_of_Reviews	Tags	Contract_Price	id
0	user_choice	austria	vienna	10.000000	10.000000	10.000000	business	0	0
3	25hours hotel beim museumsquartier	austria	vienna	5.187664	7.826087	2.574728	leisure trip solo traveler standard double ro...	10000	3
19	arcotel kaiserwasser superior	austria	vienna	5.084699	7.391304	0.730138	solo traveler 2 rooms stayed nights submitted...	40000	19
20	arcotel wimberger	austria	vienna	5.112625	6.521739	1.108438	leisure trip family with young children 2 roo...	30000	20
21	azimut hotel vienna	austria	vienna	5.106380	6.521739	0.611656	leisure trip group standard double room staye...	20000	21

Cosine Similarity

- Using CounterVectorizer each keyword in tags became a dimensions and if the hotel has it then the value in the column is a 1 and 0 otherwise.
- Since the tags are very important for the user, we first recommend get the hotels with best fit for these tags disregarding other factors.
- We then use cosine similarity on this list to get the best option when also considering the other factors such as reviews, average scores....

PCA




```
total=0
for i in range(len(dfx_pca.explained_variance_ratio_)):
    total=total+dfx_pca.explained_variance_ratio_[i]
    if total>0.9:
     n=i
     print(n)
     break
```

PCA

	0	1	2	3	4	5	6	7	8	9	...	
0	8.935204	3.253622	2.793380	1.103115	2.410415	-0.903439	1.084676	0.379189	-0.419739	-0.462841	...	0.
1	1.768751	0.191977	-0.540381	-0.674409	0.113401	-0.965103	-0.693144	0.234126	0.652246	-0.122305	...	-0.
2	0.205814	-0.394670	-0.765924	0.291291	-1.158145	1.294623	-1.017946	0.582162	-0.675942	-0.546161	...	0.
3	0.411302	-1.039892	-0.180343	-0.050216	-0.733820	0.960374	-0.928122	-0.006086	0.679524	-0.681028	...	0.
4	-0.403669	-1.110787	0.361837	-0.516304	0.249397	-0.363648	0.160275	0.275966	0.134922	-0.825257	...	0.

5 rows × 58 columns

Order of Hotels – without considering prices

```
[142] ordered_data.insert(9, 'order_factor', (data.Total_Number_of_Reviews/data.review)*data.Average_Score)
```

```
[143] ordered_data=ordered_data.sort_values(by=['order_factor'],ascending=False)
```

```
[267] print("Recommended Hotels without price paid considered")
      print(*result1,sep='\n')

      Recommended Hotels without price paid considered
      leonardo hotel vienna
      austria trend parkhotel sch nbrunn wien
      melia vienna
      steigenberger hotel herrenhof
      arion cityhotel vienna und appartements
      best western plus amedia wien
      intercityhotel wien
      austria trend hotel bosei wien
      hotel schani wien
      hotel viennart am museumsquartier
      mercure josefshof wien
      nh collection wien zentrum
      fleming s conference hotel wien
      the levante parliament a design hotel
      hotel wandl
      hotel mailberger hof
      nh wien city
      fourside hotel vienna city center
```

Ordering-with pricing

```
print("Recommended Hotels with price paid considered")
print(*result2,sep='\n')
```

```
Recommended Hotels with price paid considered
austria trend hotel bosei wien
hotel viennart am museumsquartier
fleming s conference hotel wien
hotel mailberger hof
austria trend parkhotel sch nbrunn wien
leonardo hotel vienna
melia vienna
steigenberger hotel herrenhof
arion cityhotel vienna und appartements
hotel schani wien
nh collection wien zentrum
best western plus amedia wien
intercityhotel wien
mercure josefshof wien
the levante parliament a design hotel
fourside hotel vienna city center
hotel wandl
nh wien city
```

Restaurant Recommender

Restaurant data

Restaurant Id: Unique id of every restaurant across various cities of the world
• Restaurant Name: Name of the restaurant
• Country Code: Country in which restaurant is located
• City: City in which restaurant is located
• Address: Address of the restaurant
• Locality: Location in the city
• Locality Verbose: Detailed description of the locality
• Longitude: Longitude coordinate of the restaurant's location
• Latitude: Latitude coordinate of the restaurant's location
• Cuisines: Cuisines offered by the restaurant
• Average Cost for two: Cost for two people in different currencies
• Currency: Currency of the country
• Has Table booking: yes/no
• Has Online delivery: yes/ no
• Is delivering: yes/ no
• Switch to order menu: yes/no
• Price range: range of price of food
• Aggregate Rating: Average rating out of 5
• Rating color: depending upon the average rating color
• Rating text: text on the basis of rating of rating
• Votes: Number of ratings casted by people

Standardizing

- Average Cost for two: Cost for two people in different currencies
- Currency: Currency of the country
- Country Code: Country in which restaurant is located
- Price range: range of price of food
- Aggregate Rating: Average rating out of 5

Expense data frames

Restaurant Name	rating	votes	price_for_oneUSD	price_rating
Le Petit Souffle	4.8	314	41.40	3
Izakaya Kikufuji	4.5	591	45.16	3
Heat - Edsa Shangri-La	4.4	270	150.54	4
Ooma	4.9	365	56.45	4
Sambo Kojin	4.8	229	56.45	4
Din Tai Fung	4.4	336	37.64	3
Buffet 101	4.0	520	75.27	4
Vikings	4.2	677	75.27	4
Spiral - Sofitel Philippine Plaza Manila	4.9	621	225.82	4
Locavore	4.8	532	41.40	3
Silantro Fil-Mex	4.9	1070	30.11	3
Mad Mark's Creamery & Good Eats	4.2	488	33.87	3
Silantro Fil-Mex	4.8	294	30.11	3
Guevarra's	4.2	458	37.64	3
Sodam Korean Restaurant	4.3	223	26.35	3

Separating the cuisines

Cuisines
French, Japanese, Desserts
Japanese
Seafood, Asian, Filipino, Indian

→

cat1	cat2	cat3	cat4	cat5	cat6	cat7
Desserts	French	Japanese	N/A	N/A	N/A	N/A
Japanese	N/A	N/A	N/A	N/A	N/A	N/A
Asian	Filipino	Indian	Seafood	N/A	N/A	N/A
Japanese	Sushi	N/A	N/A	N/A	N/A	N/A
Japanese	Korean	N/A	N/A	N/A	N/A	N/A
Chinese	N/A	N/A	N/A	N/A	N/A	N/A

Separate into multiple categories?

```python
def uniqueList(data):
    myCuis = df["Cuisines"].unique()
    new_list = []
    for i in myCuis:
        if i != i:
            continue
        listy = i.split(", ")
        for j in listy:
            new_list.append(j)

    best_list = []
    for food in new_list:
        found = False
        for cuisine in best_list:
            if food == cuisine:
                found = True
        if(not found):
            best_list.append(food)
    return best_list
```

```python
def makeScore(data,restrauntCats):
    perRestraunt = []
    catScore = []
    index = 0
    found = True
    for cuisineList in data['Cuisines']:
        #print(cuisineList)
        if cuisineList != cuisineList:
            cuisineList = "None"
        cuisines = cuisineList.split(", ")
        catScore = []
        for restrauntCat in restrauntCats:
            found = False
            #print(restrauntCat)
            for cuisine in cuisines:
                if(found):
                    continue
                #print("Cuis and restCuis")
                #print(cuisine)
                #print(restrauntCat)
                if restrauntCat == cuisine:
                    catScore.append(1)
                    found = True
                    #print(catScore)
            if(not found):
                catScore.append(0)
                #print(catScore)
        perRestraunt.append(catScore)
    return perRestraunt
```

What I've been looking for

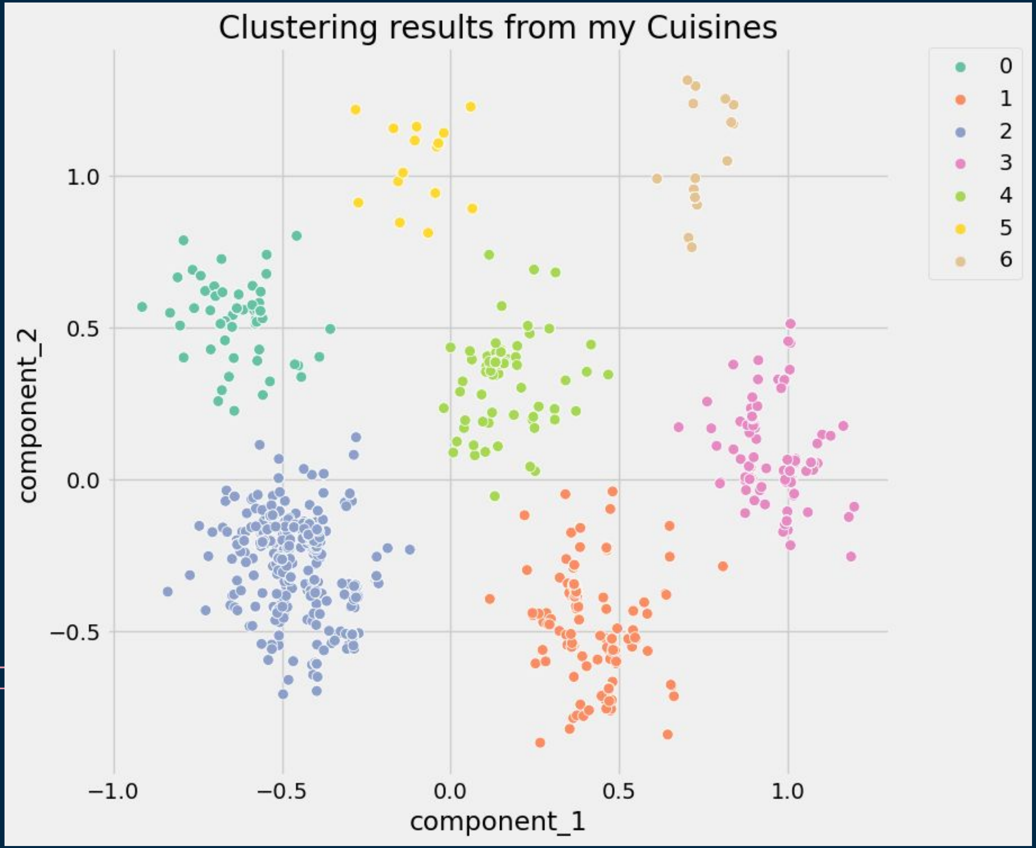
Restaurant Name	French	Japanese	Desserts	Seafood	Asian	Filipino	Indian	Sushi	Korean	Chinese	...	Drinks Only	Oriya	Bihari	Assamese	Andhra	Mangalorean
Le Petit Souffle	1	1	1	0	0	0	0	0	0	0	...	0	0	0	0	0	0
Izakaya Kikufuji	0	1	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0
Heat - Edsa Shangri-La	0	0	0	1	1	1	1	0	0	0	...	0	0	0	0	0	0
Ooma	0	1	0	0	0	0	0	1	0	0	...	0	0	0	0	0	0
Sambo Kojin	0	1	0	0	0	0	0	0	1	0	...	0	0	0	0	0	0
Din Tai Fung	0	0	0	0	0	0	0	0	0	1	...	0	0	0	0	0	0
Buffet 101	0	0	0	0	1	0	0	0	0	0	...	0	0	0	0	0	0
Vikings	0	0	0	1	1	1	0	0	0	0	...	0	0	0	0	0	0
Spiral - Sofitel Philippine Plaza Manila	0	0	0	0	1	0	1	0	0	0	...	0	0	0	0	0	0
Locavore	0	0	0	0	0	1	0	0	0	0	...	0	0	0	0	0	0

PCA and silhouette score

```
In [150]: kmeans = KMeans(n_clusters=7)

          kmeans.fit(scaled_features)


          kmeans_silhouette = silhouette_score(
              scaled_features, kmeans.labels_
          ).round(2)
          kmeans_silhouette

Out[150]: 0.08
```

```
pipe.fit(df_cuisines)

preprocessed_data = pipe["preprocessor"].transform(df_cuisines)

predicted_labels = pipe["clusterer"]["kmeans"].labels_

silhouette_score(preprocessed_data, predicted_labels)

Out[92]: 0.77518093602708
```

Visualizing our clusters



Recommending from this:

```
def recc_restraunts(X, new_data_point):
```

```
new_data_point = makeScore(["Ice Cream, Pizza"], uniqueCuis)
recc_restraunts(X, new_data_point)
```

```
Baskin Robbins
KB's Kulfi & Icecream
KB's Kulfi & Icecream
Konetto Pizza
Hot N Fresh Pizza
Da Pizza Corner
Kwality wall's swirl's
Amul Ice-Cream Parlour
```