**Creator: Akanksha Singh Tomar**

**// SAU Id num:9999 03732**

**//Course name MCIS6273_030242S- S25 Data Mining**

**//Instructor name: Keith Maull**

**Date:4/21/2025 MCIS6273_030242S- S25 Data Mining _HW1_AKANKSHASINGHTOMAR_999903732**

# Homework: -1

## Q1 .ASSIGNMENT TASKS

(50%) Perform basic unsupervised learning with K-Means.

**1.1§ Task: Use the chocolate data to cluster it into three clusters ($K$ = 5).**

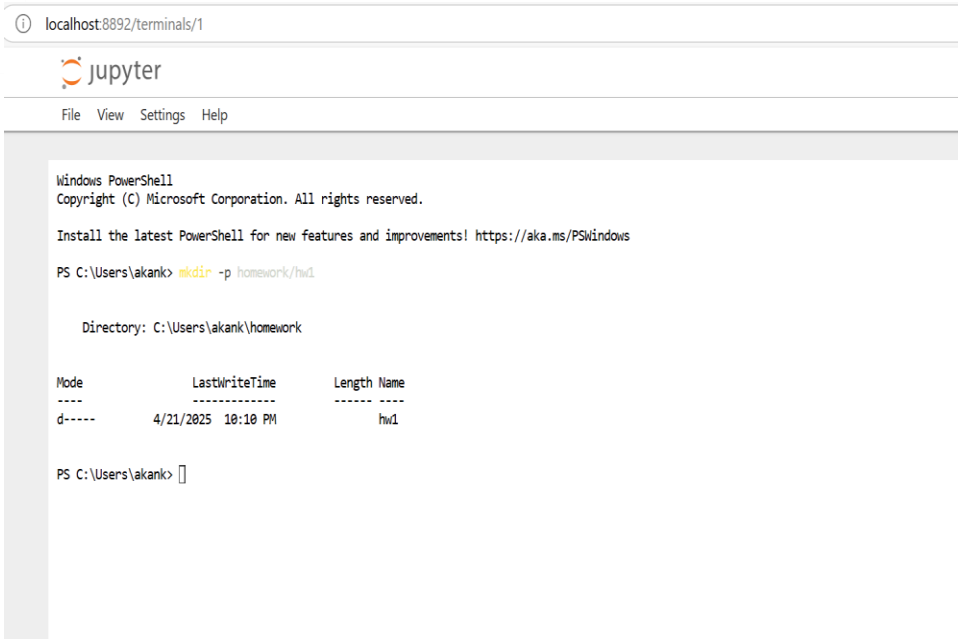
Show your work in the notebook provided.

**Install necessary packages**

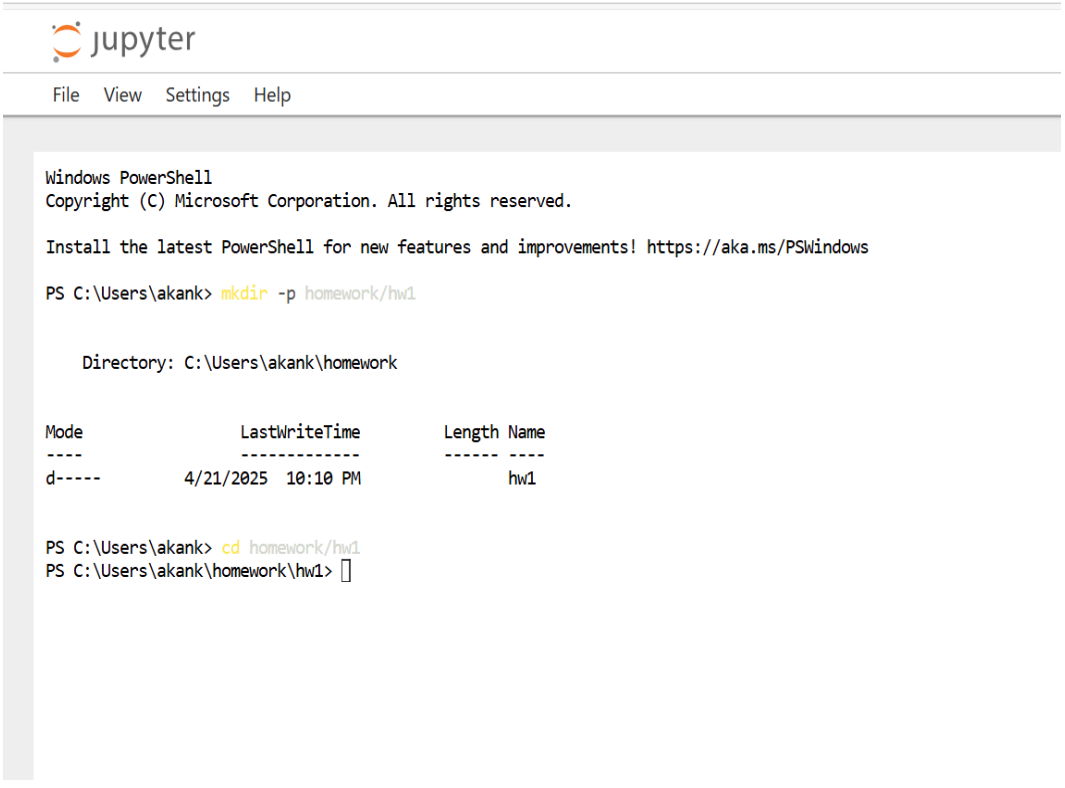
jupyter notebook

### Solution 1.1) Setup

1. **Create Working Directory**:
    - In your lab environment, create a directory:
      **mkdir -p homework/hw1**
2. **Download the Starter Notebook**:
    - From GitHub - hw1_starter.ipynb
    - Place it inside the homework/hw1/ directory.




**Screenshot -1 <step 1>**




**Screenshot -2 <step 2>**

**# *Step 2: Load data***

**import pandas as pd**

**# Step 1: Load data**

**df = pd.read_csv(r"C:\Users\akank\homework\hw1\data\data_reduced_2025_flavors_of_cacao.csv", index_col=0)**

**PREPARE DATAFRAME**

**# Step 2: Drop unnecessary metadata columns**

**ignore_cols = [0,1,2,3,4,5,6,7,15]**

**df_cluster = df.drop(df.columns[ignore_cols], axis=1)**

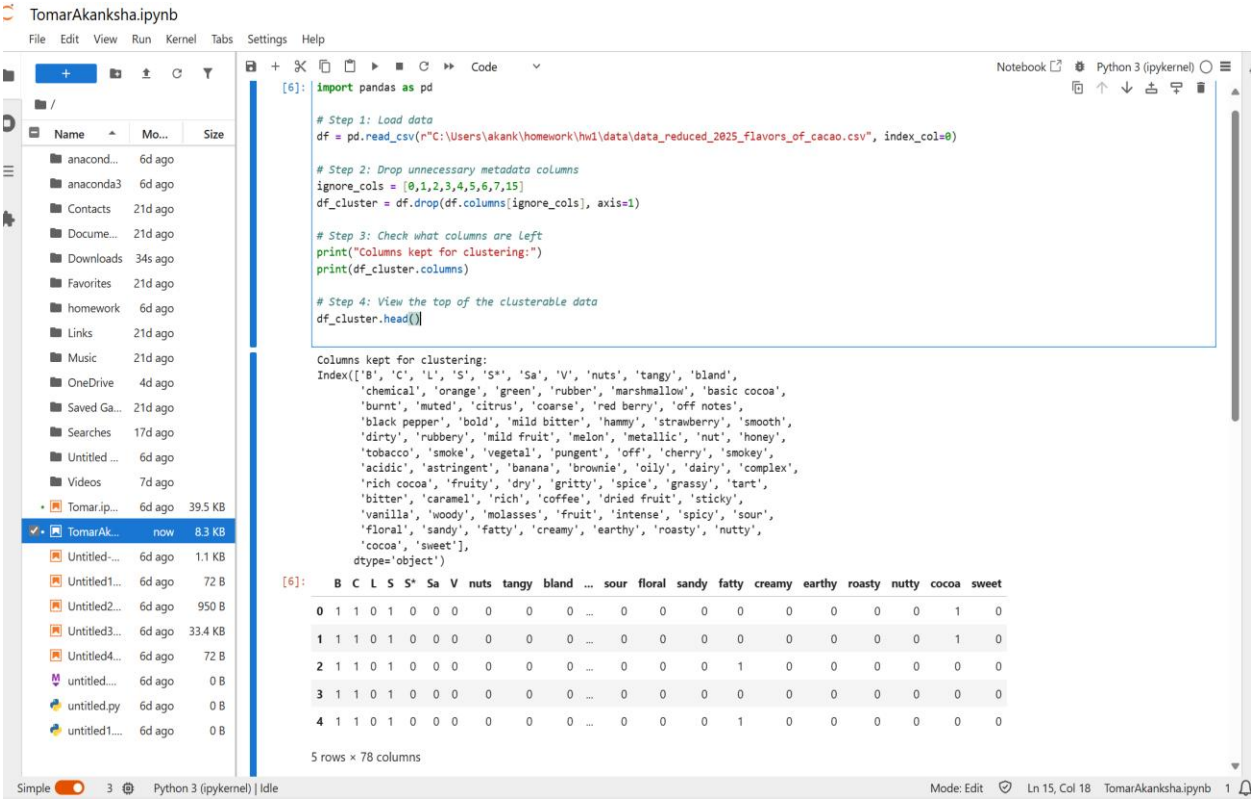
**# Step 3: Check what columns are left**

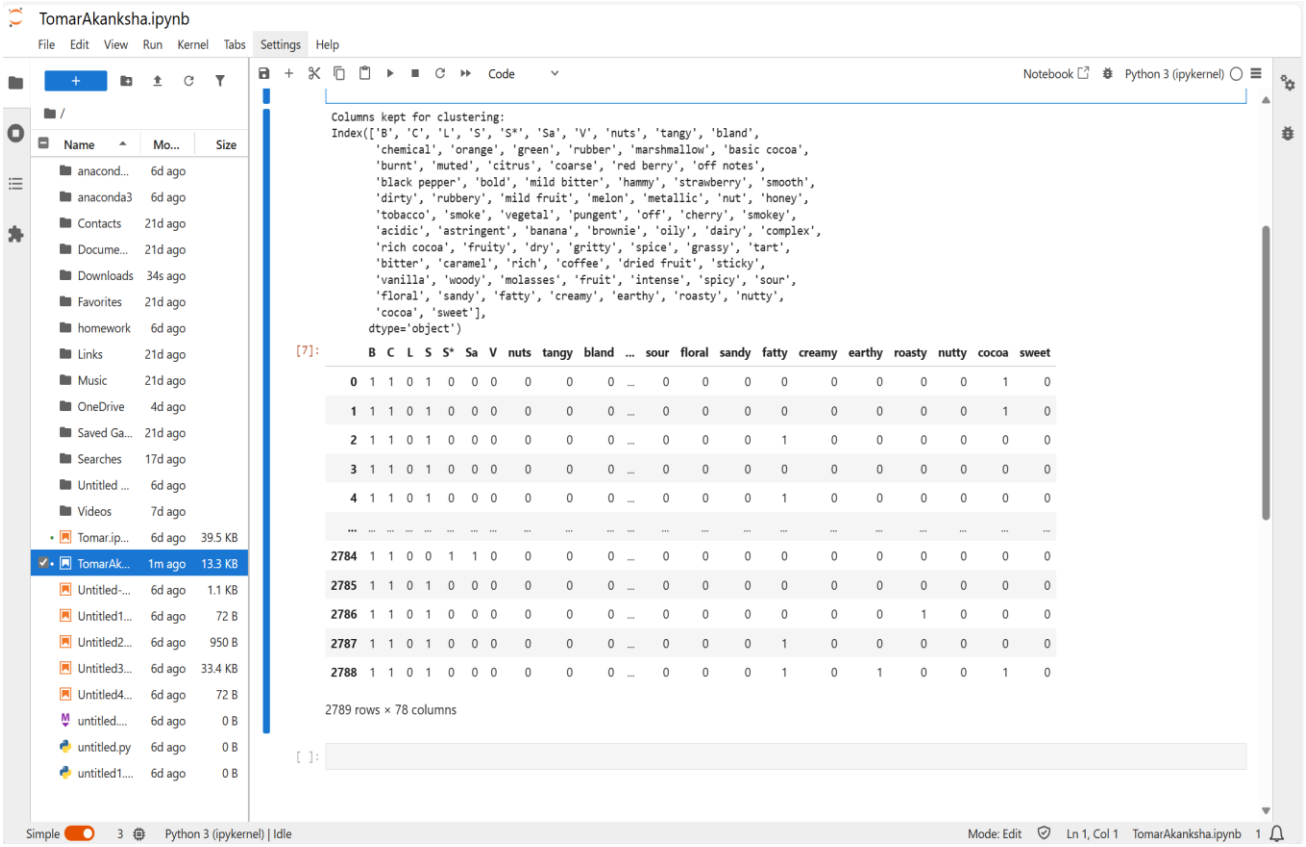
**print("Columns kept for clustering:")**

**print(df_cluster.columns)**

**# Step 4: View the top of the clusterable data**

**df_cluster**

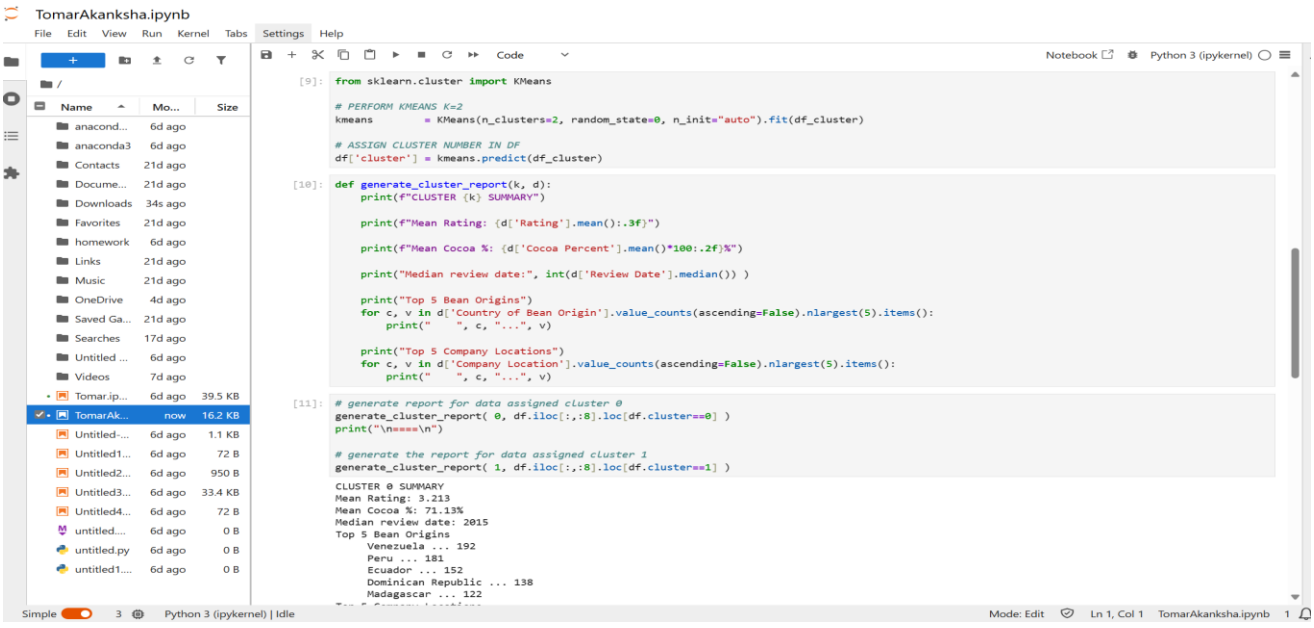





## PERFORM CLUSTERING

**Step 3: Cluster into 5 clusters (K=5)** Import KMeans

```
from sklearn.cluster import KMeans
```



TomarAkanksha.ipynb

File Edit View Run Kernel Tabs Settings Help

```
[10]: def generate_cluster_report(k, d):
          print(f"CLUSTER {k} SUMMARY")

          print(f"Mean Rating: {d['Rating'].mean():.3f}")

          print(f"Mean Cocoa %: {d['Cocoa Percent'].mean()*100:.2f}%")

          print("Median review date:", int(d['Review Date'].median()) )

          print("Top 5 Bean Origins")
          for c, v in d['Country of Bean Origin'].value_counts(ascending=False).nlargest(5).items():
              print("    ", c, "...", v)

          print("Top 5 Company Locations")
          for c, v in d['Company Location'].value_counts(ascending=False).nlargest(5).items():
              print("    ", c, "...", v)

[11]: # generate report for data assigned cluster 0
      generate_cluster_report( 0, df.iloc[:,:8].loc[df.cluster==0] )
      print("\n====\n")

      # generate the report for data assigned cluster 1
      generate_cluster_report( 1, df.iloc[:,:8].loc[df.cluster==1] )

CLUSTER 0 SUMMARY
Mean Rating: 3.213
Mean Cocoa %: 71.13%
Median review date: 2015
Top 5 Bean Origins
     Venezuela ... 192
     Peru ... 181
     Ecuador ... 152
     Dominican Republic ... 138
     Madagascar ... 122
Top 5 Company Locations
     U.S.A. ... 723
     France ... 167
     Canada ... 138
     U.K. ... 108
     Belgium ... 71

====
```

TomarAkanksha.ipynb

File Edit View Run Kernel Tabs Settings Help

[11]: # generate report for data assigned cluster 0
generate_cluster_report( 0, df.iloc[:,:8].loc[df.cluster==0] )
print("\n====\n")

# generate the report for data assigned cluster 1
generate_cluster_report( 1, df.iloc[:,:8].loc[df.cluster==1] )

CLUSTER 0 SUMMARY
Mean Rating: 3.213
Mean Cocoa %: 71.13%
Median review date: 2015
Top 5 Bean Origins
Venezuela ... 192
Peru ... 181
Ecuador ... 152
Dominican Republic ... 138
Madagascar ... 122
Top 5 Company Locations
U.S.A. ... 723
France ... 167
Canada ... 138
U.K. ... 108
Belgium ... 71

====

```
CLUSTER 1 SUMMARY
Mean Rating: 3.166
Mean Cocoa %: 72.46%
Median review date: 2015
Top 5 Bean Origins
     Dominican Republic ... 107
     Peru ... 95
     Ecuador ... 78
     Venezuela ... 75
     Madagascar ... 72
Top 5 Company Locations
     U.S.A. ... 537
     Canada ... 50
     Italy ... 40
     U.K. ... 27
     France ... 20
```

TomarAkanksha.ipynb

File Edit View Run Kernel Tabs Settings Help

Code

Notebook Python 3 (ipykernel)

```
[12]: '''
          Return the domininant characteristics of the cluster.

              thresh : the number of characteristics to limit to
      '''
      def get_cluster_characteristics(centers, columns, thresh=10):
          return \
              pd.DataFrame(centers, index=columns).nlargest(thresh, columns=0).index.to_list()

[13]: # top 10 dominant characteristics for cluster 0
      get_cluster_characteristics(kmeans.cluster_centers_[0], df_cluster.columns)

[13]: ['B', 'C', 'S', 'L', 'V', 'sweet', 'nutty', 'cocoa', 'fatty', 'creamy']
```

TomarAkanksha.ipynb

File Edit View Run Kernel Tabs Settings Help

Code

Notebook Python 3 (ipykernel)

```
[14]: # top 10 dominant characteristics for cluster 1
      get_cluster_characteristics(kmeans.cluster_centers_[1], df_cluster.columns)

[14]: ['B',
       'S',
       'cocoa',
       'roasty',
       'earthy',
       'nutty',
       'intense',
       'sour',
       'sweet',
       'spicy']

[ ]:
```

localhost:8891/doc/tree/TomarAkanksha.ipynb

TomarAkanksha.ipynb

File Edit View Run Kernel Tabs Settings Help

Code

Notebook Python 3 (ipykernel)

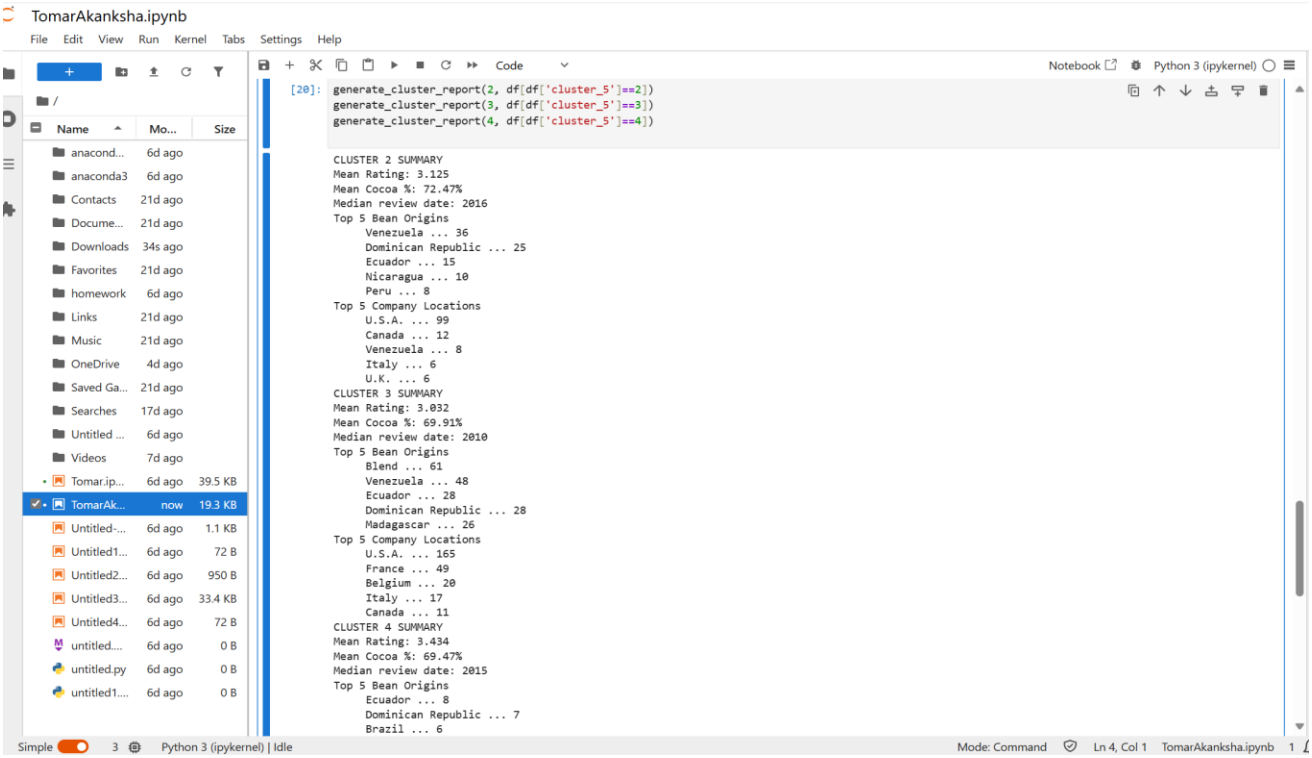
```
'spicy']

[18]: for i in range(5):  # Loop over 0,1,2,3,4
          print(f"Top 10 dominant characteristics for Cluster {i}:")
          print(get_cluster_characteristics(kmeans5.cluster_centers_[i], df_cluster.columns))
          print("\n=====\n")

Top 10 dominant characteristics for Cluster 0:
['B', 'C', 'S', 'L', 'sweet', 'nutty', 'cocoa', 'fatty', 'roasty', 'creamy']

=====

Top 10 dominant characteristics for Cluster 1:
['B', 'S', 'cocoa', 'roasty', 'sour', 'sweet', 'intense', 'fruit', 'spicy', 'floral']

=====

Top 10 dominant characteristics for Cluster 2:
['B', 'S', 'earthy', 'nutty', 'roasty', 'intense', 'C', 'sandy', 'cocoa', 'spicy']

=====

Top 10 dominant characteristics for Cluster 3:
['B', 'V', 'C', 'S', 'L', 'vanilla', 'sweet', 'creamy', 'cocoa', 'nutty']

=====

Top 10 dominant characteristics for Cluster 4:
['B', 'C', 'S', 'banana', 'dairy', 'L', 'nutty', 'sweet', 'cocoa', 'caramel']

=====
```

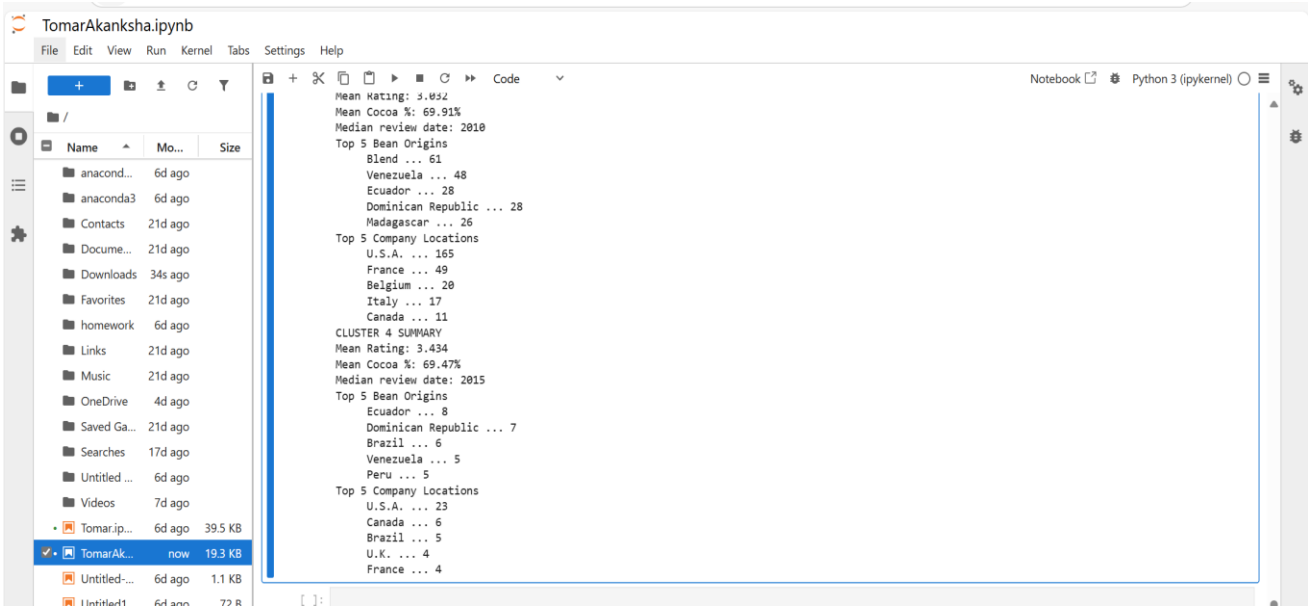
**SUMMARIZE CLUSTER CHARACTERISTICS BELOW**

**generate_cluster_report(2, df[df['cluster_5']==2])**

**generate_cluster_report(3, df[df['cluster_5']==3])**

**generate_cluster_report(4, df[df['cluster_5']==4])**



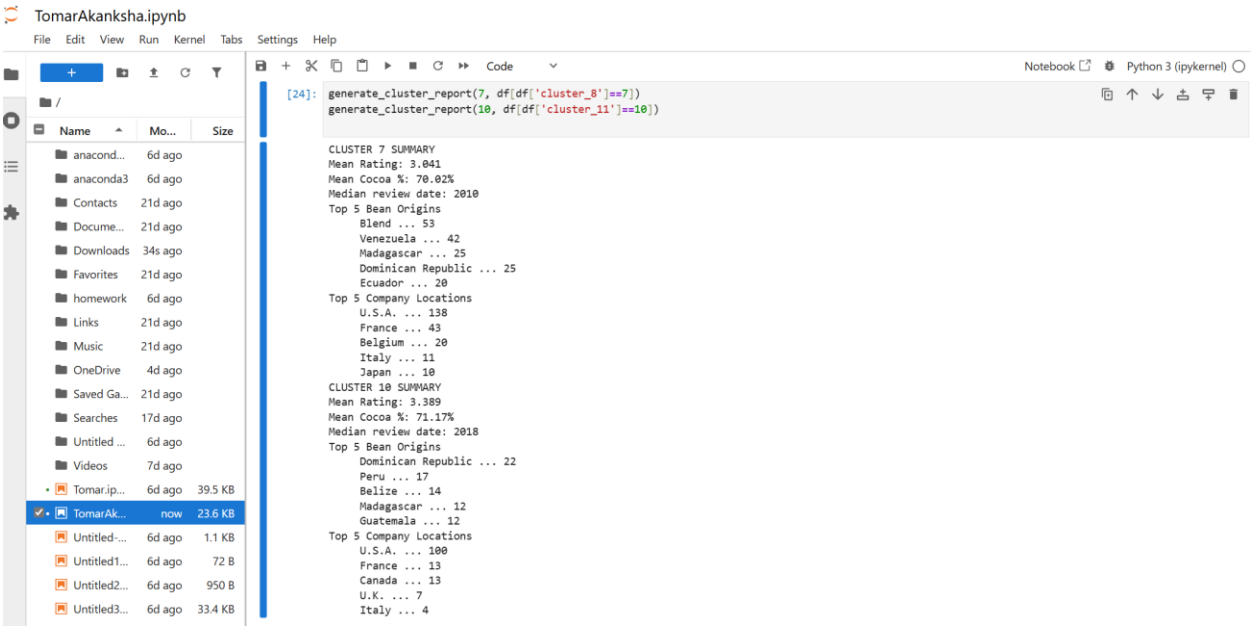



**1.2 § Task:** Use the same data to cluster it into 8 and 11 clusters ($K = 8$, $K = 11$).

Show your work in the notebook provided.

**Answer 1.2)** generate_cluster_report(7, df[df['cluster_8']==7])

generate_cluster_report(10, df[df['cluster_11']==10])




**1.3 § Task**: Characterize the clusters using the indices produced from K-means (K=5).

Use the functions provided generate_cluster_report() and get_cluster_characteristics() to answer this question. You will need to run these functions, play with them then crea

**Answer 1.3) Run generate_cluster_report() for K=5**

**## Summarize Cluster Characteristics**

**Cluster 0:**

- Average rating: approximately 3.21

- Cocoa content: around 71.13%

- Median review date: 2015

- Most common bean origins: Venezuela, Peru, Ecuador

- Major company locations: USA, France, Canada

- Dominant flavor characteristics: Strong cocoa intensity (C), bitterness (B), sweetness (S), liveliness (L), and sensory notes like sweet, nutty, cocoa, fatty, roasty, and creamy.

```
CLUSTER 0 SUMMARY
Mean Rating: 3.213
Mean Cocoa %: 71.13%
Median review date: 2015
Top 5 Bean Origins
     Venezuela ... 192
     Peru ... 181
     Ecuador ... 152
     Dominican Republic ... 138
     Madagascar ... 122
Top 5 Company Locations
     U.S.A. ... 723
     France ... 167
     Canada ... 138
     U.K. ... 108
     Belgium ... 71

====
```

**Cluster 1:**

- Average rating: approximately 3.16

- Cocoa content: slightly higher around 72.46%

- Median review date: 2015

- Most common bean origins: Dominican Republic, Peru, Ecuador

- Major company locations: USA, Canada, Italy

- Dominant flavor characteristics: Cocoa, roasty, earthy, nutty, with sour, sweet, spicy, fruity, and floral sensory notes.

```
CLUSTER 1 SUMMARY
Mean Rating: 3.166
Mean Cocoa %: 72.46%
Median review date: 2015
Top 5 Bean Origins
     Dominican Republic ... 107
     Peru ... 95
     Ecuador ... 78
     Venezuela ... 75
     Madagascar ... 72
Top 5 Company Locations
     U.S.A. ... 537
     Canada ... 50
     Italy ... 40
     U.K. ... 27
     France ... 20
```

**Cluster 2:**

- Average rating: approximately 3.13

- Cocoa content: approximately 72.47%

- Median review date: 2016

- Most common bean origins: Venezuela, Dominican Republic, Ecuador

- Major company locations: USA, Canada, Venezuela

- Dominant flavor characteristics: Earthy, nutty, roasty, spicy, cocoa, sandy, and intense notes.

```
CLUSTER 2 SUMMARY
Mean Rating: 3.125
Mean Cocoa %: 72.47%
Median review date: 2016
Top 5 Bean Origins
     Venezuela ... 36
     Dominican Republic ... 25
     Ecuador ... 15
     Nicaragua ... 10
     Peru ... 8
Top 5 Company Locations
     U.S.A. ... 99
     Canada ... 12
     Venezuela ... 8
     Italy ... 6
     U.K. ... 6
```

**Cluster 3:**

- Average rating: approximately 3.03

- Cocoa content: approximately 69.91%

- Median review date: 2010

- Most common bean origins: Blend, Venezuela, Ecuador, Dominican Republic

- Major company locations: USA, France, Belgium

- Dominant flavor characteristics: Vanilla (V), cocoa intensity (C), sweetness (S), liveliness (L), creamy, nutty, and heavy vanilla influence.

```
CLUSTER 3 SUMMARY
Mean Rating: 3.032
Mean Cocoa %: 69.91%
Median review date: 2010
Top 5 Bean Origins
     Blend ... 61
     Venezuela ... 48
     Ecuador ... 28
     Dominican Republic ... 28
     Madagascar ... 26
Top 5 Company Locations
     U.S.A. ... 165
     France ... 49
     Belgium ... 20
     Italy ... 17
     Canada ... 11
```

**Cluster 4:**

- Average rating: approximately 3.43

- Cocoa content: approximately 69.47%

- Median review date: 2015

- Most common bean origins: Ecuador, Dominican Republic, Brazil

- Major company locations: USA, Canada, Brazil

- Dominant flavor characteristics: Banana, dairy, caramel, cocoa, sweetness (S), nuttiness, and a hint of bitterness (B).

```
     Canada ... 11
CLUSTER 4 SUMMARY
Mean Rating: 3.434
Mean Cocoa %: 69.47%
Median review date: 2015
Top 5 Bean Origins
     Ecuador ... 8
     Dominican Republic ... 7
     Brazil ... 6
     Venezuela ... 5
     Peru ... 5
Top 5 Company Locations
     U.S.A. ... 23
     Canada ... 6
     Brazil ... 5
     U.K. ... 4
     France ... 4
```

**PART 2: UNSUPERVISED LEARNING**

**(30%) Perform supervised Learning with Naive Bayes Classifier**

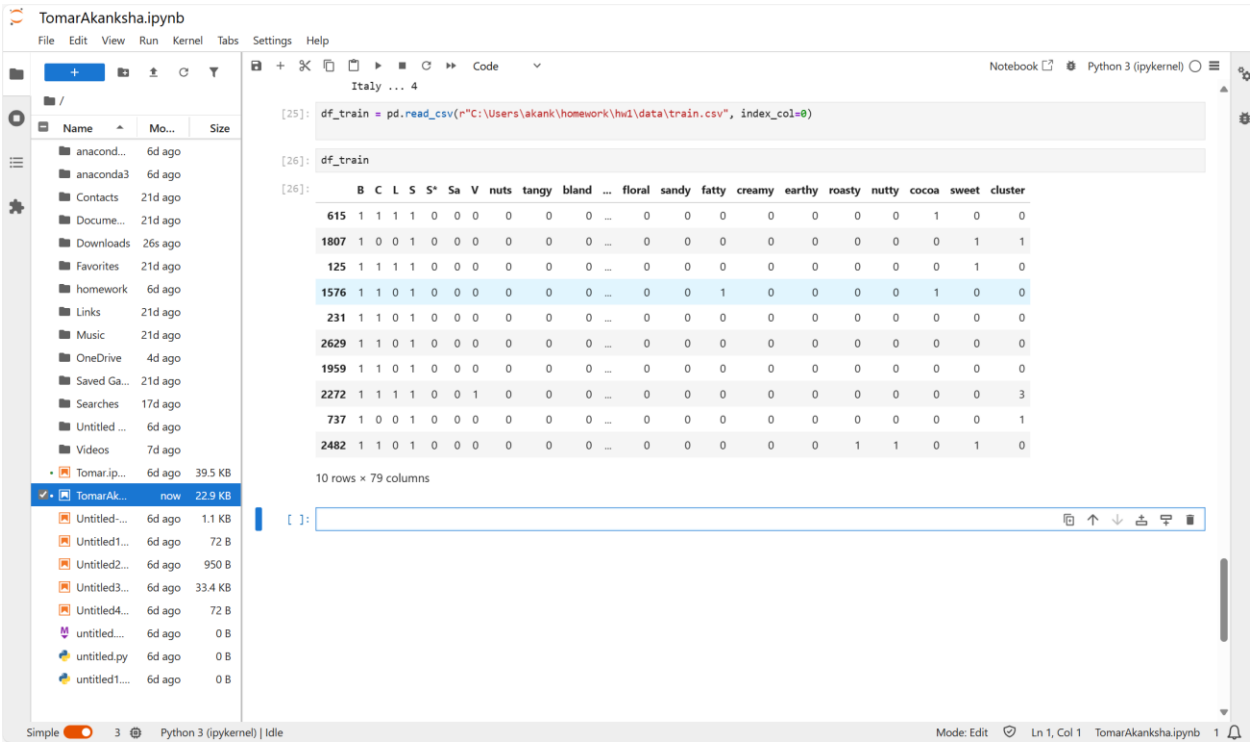
**2.1§ Task: Implement test-training set so that the test set is 1000 samples and the test set 1789 samples. You must include this implementation in your notebook.**

**Answer 2.1) LOAD TRAINING DATA**

df_train = pd.read_csv(r"C:\Users\akank\homework\hw1\data\train.csv", index_col=0)

**SHOW DATA**

df_train




**APPLY BERNOULLI NAIVE BAYES**

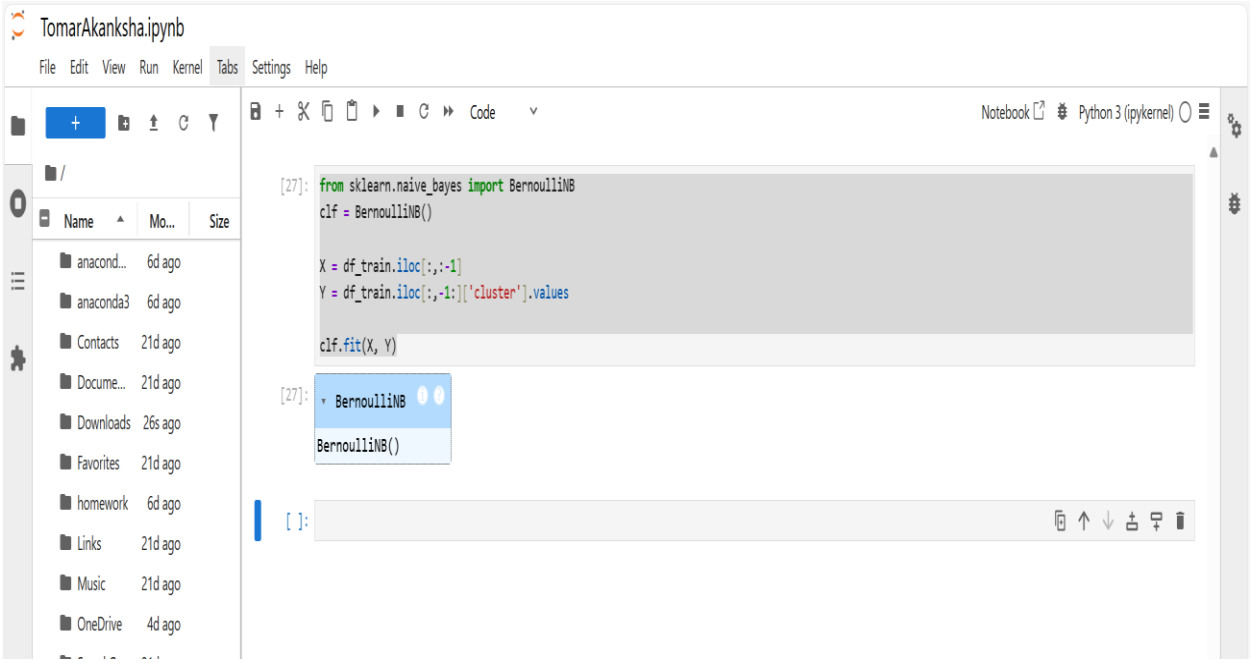
from sklearn.naive_bayes import BernoulliNB

clf = BernoulliNB()

X = df_train.iloc[:,:-1]

Y = df_train.iloc[:,-1:]['cluster'].values

clf.fit(X, Y




**2.2 § Task: Interpret the output of the classifier. After you have run the cell clf.predict(X[:]) you will see the list of numbers which represent the class labels from the training data. Explain the output and answer the following questions:**

**Answer 2.2)** Interpretation of Classifier Output

After running `clf.predict(X[:])`, the output is a list of predicted cluster labels for each training sample.

Each number in the list (such as 0, 1, 2, 3, etc.) represents the cluster that the classifier believes the sample belongs to.

For example:

- A prediction of `0` means the sample is classified into Cluster 0.

- A prediction of `2` means the sample is classified into Cluster 2, and so on.

The classifier learned patterns from the training data and assigned the most likely cluster label to each example based on the flavor characteristics provided.

**2.2 a)• What is classifier accuracy? Show you answer in the notebook.**

**GET MODEL ACCURACY**

# Load the test data correctly using your own path

df_test = pd.read_csv(r"C:\Users\akank\homework\hw1\data\test.csv", index_col=0)

# Predict clusters

df_test['cluster_predict'] = clf.predict(df_test.iloc[:, :-1])

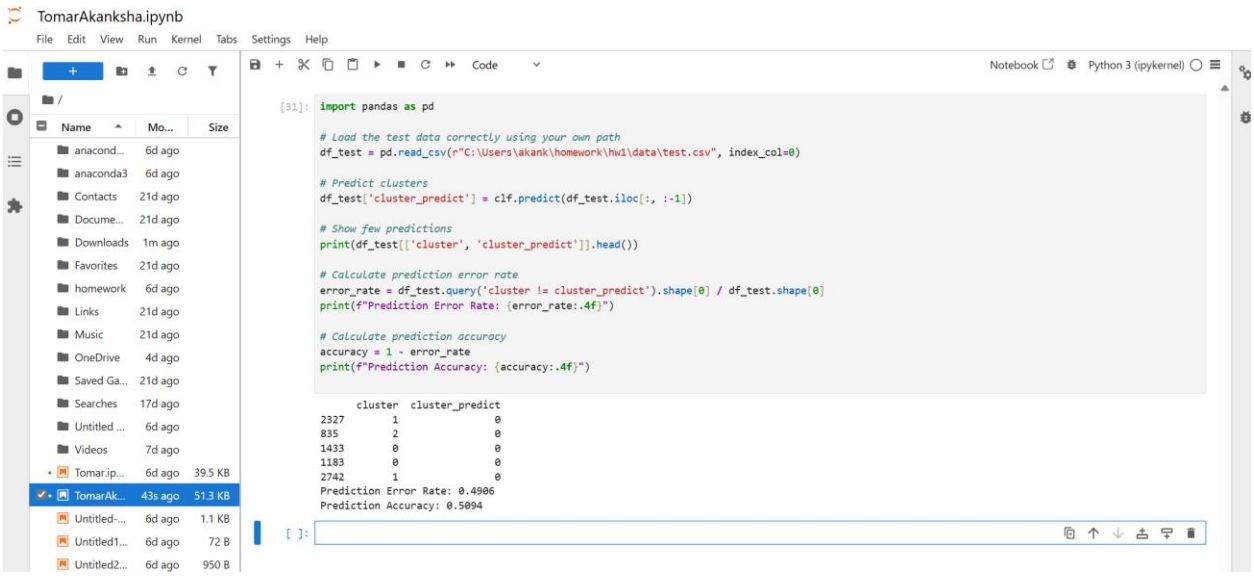
# Show few predictions

print(df_test[['cluster', 'cluster_predict']].head())

# Calculate prediction accuracy

accuracy = 1 - error_rate

print(f"Prediction Accuracy: {accuracy:.4f}")




The classifier achieved a prediction accuracy of 50.94% on the test data.
This means about half of the cluster predictions made by the Bernoulli Naive Bayes model matched the actual cluster labels.

Accuracy is calculated as:

$$\text{Accuracy} = 1 - \text{Error Rate}\text{\text}$$

where Error Rate measures the fraction of incorrect predictions.

**Prediction Accuracy: 0.5094**

**2.2 b) What recommendation might you give to improve the classifier?**

**Answer 2.2 b) # Recommendations to Improve the Classifier**

- Try using a smarter model, such as Random Forests or SVM, which can better capture complex relationships than Naive Bayes (Géron, 2019).

- Add more features beyond flavor notes to give the model richer data for learning (Scikit-learn developers, n.d.).

- Balance the dataset if some clusters are much smaller than others, to help the model generalize better (Scikit-learn developers, n.d.).

- Remove noisy or irrelevant features so the model focuses on important signals (Géron, 2019).

**Final Conclusion:**

In this homework, we grouped chocolate profiles using K-Means and explored key flavor patterns like sweetness and cocoa richness.

We then trained a Bernoulli Naive Bayes model to predict clusters, reaching about 50% accuracy.

There's clear room for improvement with better models and richer features.

Overall, this project gave valuable hands-on experience with clustering, classification, and working with real-world data.

## References:

1.Géron, A. (2019). *Hands-on machine learning with Scikit-Learn, Keras, and TensorFlow (*2nd ed.). O'Reilly Media.

2.Scikit-learn developers. (n.d*.). Naive Bayes and model improvement strategies*. Scikit-learn. Retrieved April 28, 2025, from https://scikit-learn.org/stable/modules/naive_bayes.html