Healthcare Scenario: Healthy Living and Wellness Clustering Exercise

My code:

```
import pandas as pd

import matplotlib.pyplot as plt

import seaborn as sns

from sklearn.cluster import KMeans

from sklearn.preprocessing import StandardScaler

from sklearn.decomposition import PCA

from sklearn.metrics import silhouette_score


# Load the dataset

file_path = '/mnt/data/simulated_health_wellness_data.csv'

data = pd.read_csv(file_path)


# Set the plotting style

sns.set(style="whitegrid")


# 1. Exploratory Data Analysis (EDA)

# Plot histograms for each variable

data.hist(bins=15, figsize=(12, 8), color='skyblue')

plt.suptitle("Histograms of Health & Wellness Features")

plt.show()


# Create a correlation heatmap to understand relationships between variables

plt.figure(figsize=(8, 6))

sns.heatmap(data.corr(), annot=True, cmap="coolwarm", linewidths=0.5)
```

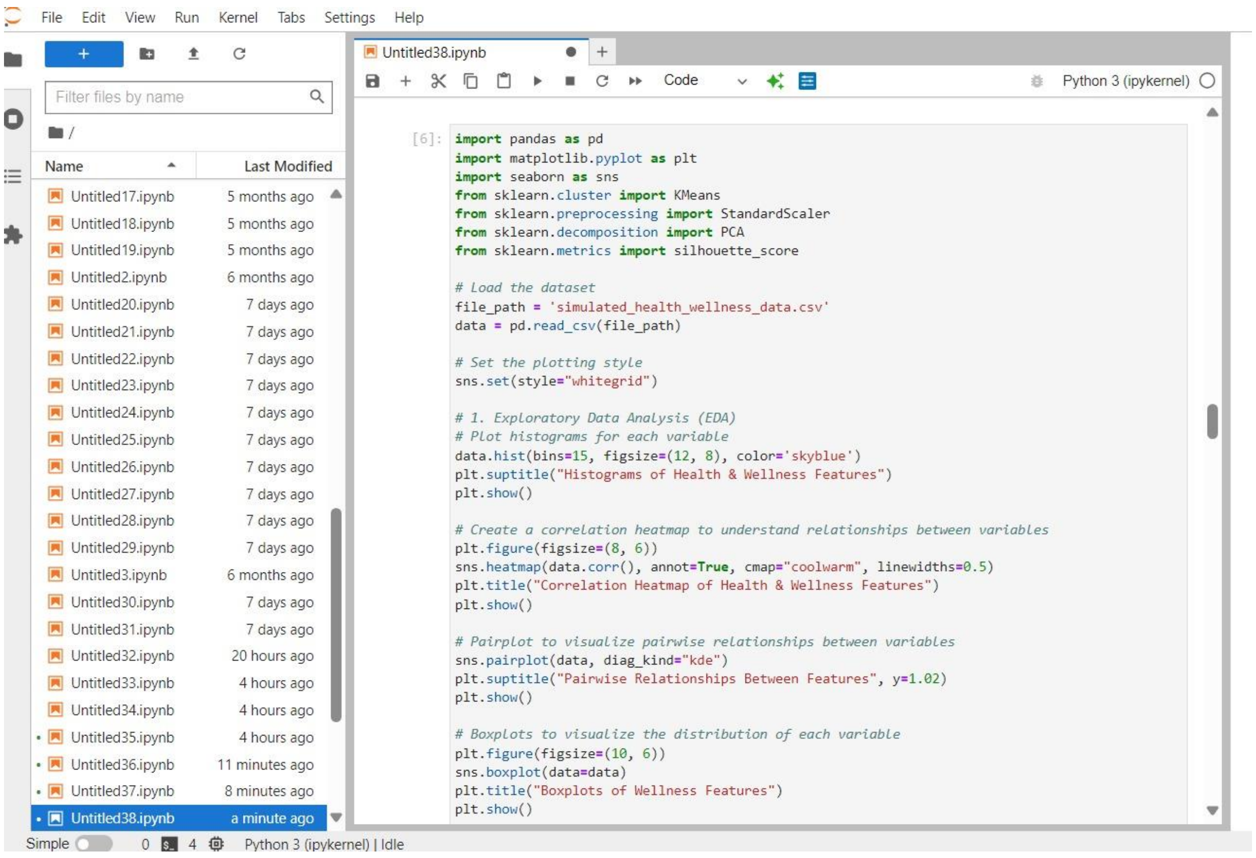
```
plt.title("Correlation Heatmap of Health & Wellness Features")
plt.show()

# Pairplot to visualize pairwise relationships between variables
sns.pairplot(data, diag_kind="kde")
plt.suptitle("Pairwise Relationships Between Features", y=1.02)
plt.show()

# Boxplots to visualize the distribution of each variable
plt.figure(figsize=(10, 6))
sns.boxplot(data=data)
plt.title("Boxplots of Wellness Features")
plt.show()

# 2. K-Means Clustering
# Scale the data
scaler = StandardScaler()
scaled_data = scaler.fit_transform(data)

# Perform K-Means Clustering
kmeans = KMeans(n_clusters=4, random_state=42)
clusters = kmeans.fit_predict(scaled_data)

# Add cluster labels to the original data
data['Cluster'] = clusters
```

```
# Calculate silhouette score

sil_score = silhouette_score(scaled_data, clusters)

print(f"Silhouette Score for K-Means: {sil_score}")


# Visualize K-Means Clustering using a pairplot

sns.pairplot(data, hue='Cluster', diag_kind='kde', palette='bright')

plt.suptitle("K-Means Clustering of Health & Wellness Data", y=1.02)

plt.show()


# 3. Principal Component Analysis (PCA)

# Apply PCA to reduce dimensionality

pca = PCA(n_components=2)

pca_data = pca.fit_transform(scaled_data)


# Visualize the explained variance ratio

explained_variance = pca.explained_variance_ratio_

print(f"Explained Variance Ratio by PCA Components: {explained_variance}")


# Plot the first two PCA components

plt.figure(figsize=(8, 6))

plt.scatter(pca_data[:, 0], pca_data[:, 1], c=clusters, cmap='viridis')

plt.title("PCA of Health & Wellness Data")

plt.xlabel("PCA Component 1")

plt.ylabel("PCA Component 2")

plt.colorbar(label='Cluster')

plt.show()
```

```
# 4. Model Comparison (Before and After PCA)

# Apply K-Means on PCA-reduced data

kmeans_pca = KMeans(n_clusters=4, random_state=42)

clusters_pca = kmeans_pca.fit_predict(pca_data)


# Silhouette score after PCA

sil_score_pca = silhouette_score(pca_data, clusters_pca)

print(f"Silhouette Score after PCA: {sil_score_pca}")


# Visualize clustering after PCA

plt.scatter(pca_data[:, 0], pca_data[:, 1], c=clusters_pca, cmap='plasma', s=50)

plt.title("Clustering after PCA")

plt.xlabel("PCA Component 1")

plt.ylabel("PCA Component 2")

plt.colorbar(label='Cluster')

plt.show()


# 5. Evaluation Metrics

# Calculate inertia (within-cluster sum of squares) for K-Means

inertia = kmeans.inertia_

print(f"K-Means Inertia: {inertia}")


# Compare silhouette scores before and after PCA

print(f"Silhouette Score before PCA: {sil_score}")

print(f"Silhouette Score after PCA: {sil_score_pca}")
```

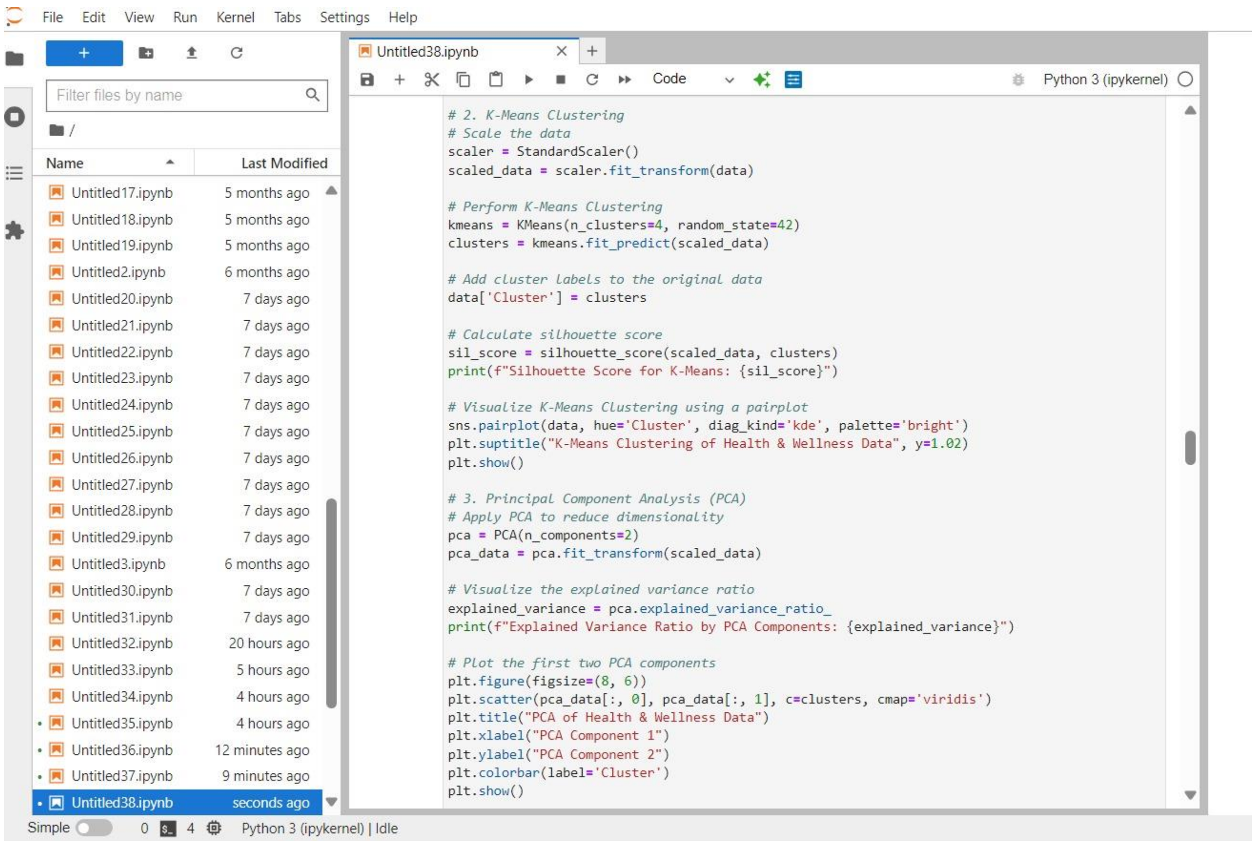
Output:

Here Is the code for the information


File
Edit
View
Run
Kernel
Tabs
Settings
Help
Filter files by name
Name
Last Modified
Untitled17.ipynb 5 months ago
Untitled18.ipynb 5 months ago
Untitled19.ipynb 5 months ago
Untitled2.ipynb 6 months ago
Untitled20.ipynb 7 days ago
Untitled21.ipynb 7 days ago
Untitled22.ipynb 7 days ago
Untitled23.ipynb 7 days ago
Untitled24.ipynb 7 days ago
Untitled25.ipynb 7 days ago
Untitled26.ipynb 7 days ago
Untitled27.ipynb 7 days ago
Untitled28.ipynb 7 days ago
Untitled29.ipynb 7 days ago
Untitled3.ipynb 6 months ago
Untitled30.ipynb 7 days ago
Untitled31.ipynb 7 days ago
Untitled32.ipynb 20 hours ago
Untitled33.ipynb 4 hours ago
Untitled34.ipynb 4 hours ago
Untitled35.ipynb 4 hours ago
Untitled36.ipynb 11 minutes ago
Untitled37.ipynb 8 minutes ago
Untitled38.ipynb a minute ago
Untitled38.ipynb
Code
Python 3 (ipykernel)
Simple
0
4
Python 3 (ipykernel) | Idle

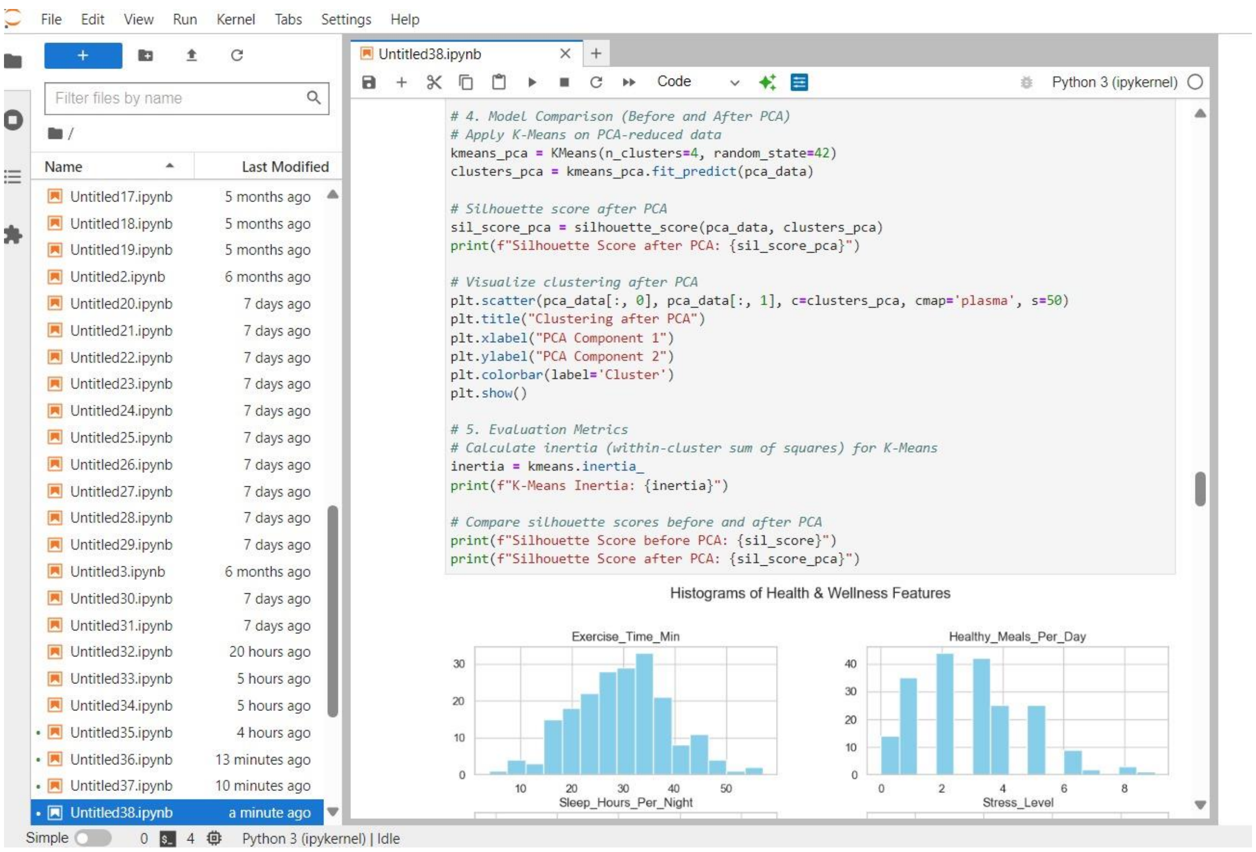
```
[6]: import pandas as pd
     import matplotlib.pyplot as plt
     import seaborn as sns
     from sklearn.cluster import KMeans
     from sklearn.preprocessing import StandardScaler
     from sklearn.decomposition import PCA
     from sklearn.metrics import silhouette_score

     # Load the dataset
     file_path = 'simulated_health_wellness_data.csv'
     data = pd.read_csv(file_path)

     # Set the plotting style
     sns.set(style="whitegrid")

     # 1. Exploratory Data Analysis (EDA)
     # Plot histograms for each variable
     data.hist(bins=15, figsize=(12, 8), color='skyblue')
     plt.suptitle("Histograms of Health & Wellness Features")
     plt.show()

     # Create a correlation heatmap to understand relationships between variables
     plt.figure(figsize=(8, 6))
     sns.heatmap(data.corr(), annot=True, cmap="coolwarm", linewidths=0.5)
     plt.title("Correlation Heatmap of Health & Wellness Features")
     plt.show()

     # Pairplot to visualize pairwise relationships between variables
     sns.pairplot(data, diag_kind="kde")
     plt.suptitle("Pairwise Relationships Between Features", y=1.02)
     plt.show()

     # Boxplots to visualize the distribution of each variable
     plt.figure(figsize=(10, 6))
     sns.boxplot(data=data)
     plt.title("Boxplots of Wellness Features")
     plt.show()
```

k-mean clustering


File
Edit
View
Run
Kernel
Tabs
Settings
Help
Filter files by name
Name
Last Modified
Untitled17.ipynb 5 months ago
Untitled18.ipynb 5 months ago
Untitled19.ipynb 5 months ago
Untitled2.ipynb 6 months ago
Untitled20.ipynb 7 days ago
Untitled21.ipynb 7 days ago
Untitled22.ipynb 7 days ago
Untitled23.ipynb 7 days ago
Untitled24.ipynb 7 days ago
Untitled25.ipynb 7 days ago
Untitled26.ipynb 7 days ago
Untitled27.ipynb 7 days ago
Untitled28.ipynb 7 days ago
Untitled29.ipynb 7 days ago
Untitled3.ipynb 6 months ago
Untitled30.ipynb 7 days ago
Untitled31.ipynb 7 days ago
Untitled32.ipynb 20 hours ago
Untitled33.ipynb 5 hours ago
Untitled34.ipynb 4 hours ago
Untitled35.ipynb 4 hours ago
Untitled36.ipynb 12 minutes ago
Untitled37.ipynb 9 minutes ago
Untitled38.ipynb seconds ago
Untitled38.ipynb
Code
Python 3 (ipykernel)
Simple
0
4
Python 3 (ipykernel) | Idle

```python
# 2. K-Means Clustering
# Scale the data
scaler = StandardScaler()
scaled_data = scaler.fit_transform(data)

# Perform K-Means Clustering
kmeans = KMeans(n_clusters=4, random_state=42)
clusters = kmeans.fit_predict(scaled_data)

# Add cluster labels to the original data
data['Cluster'] = clusters

# Calculate silhouette score
sil_score = silhouette_score(scaled_data, clusters)
print(f"Silhouette Score for K-Means: {sil_score}")

# Visualize K-Means Clustering using a pairplot
sns.pairplot(data, hue='Cluster', diag_kind='kde', palette='bright')
plt.suptitle("K-Means Clustering of Health & Wellness Data", y=1.02)
plt.show()

# 3. Principal Component Analysis (PCA)
# Apply PCA to reduce dimensionality
pca = PCA(n_components=2)
pca_data = pca.fit_transform(scaled_data)

# Visualize the explained variance ratio
explained_variance = pca.explained_variance_ratio_
print(f"Explained Variance Ratio by PCA Components: {explained_variance}")

# Plot the first two PCA components
plt.figure(figsize=(8, 6))
plt.scatter(pca_data[:, 0], pca_data[:, 1], c=clusters, cmap='viridis')
plt.title("PCA of Health & Wellness Data")
plt.xlabel("PCA Component 1")
plt.ylabel("PCA Component 2")
plt.colorbar(label='Cluster')
plt.show()
```

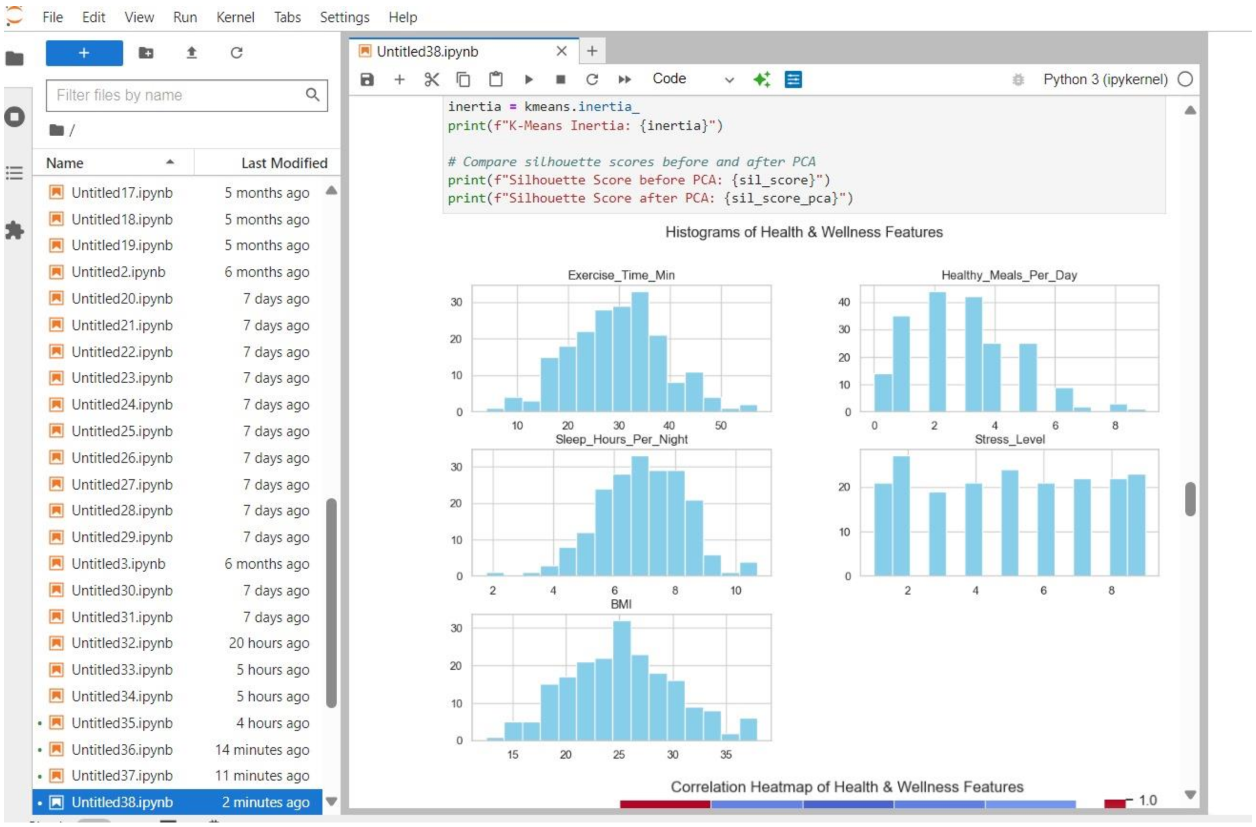
model comparison and evaluation metrics



Name	Last Modified
Untitled17.ipynb	5 months ago
Untitled18.ipynb	5 months ago
Untitled19.ipynb	5 months ago
Untitled2.ipynb	6 months ago
Untitled20.ipynb	7 days ago
Untitled21.ipynb	7 days ago
Untitled22.ipynb	7 days ago
Untitled23.ipynb	7 days ago
Untitled24.ipynb	7 days ago
Untitled25.ipynb	7 days ago
Untitled26.ipynb	7 days ago
Untitled27.ipynb	7 days ago
Untitled28.ipynb	7 days ago
Untitled29.ipynb	7 days ago
Untitled3.ipynb	6 months ago
Untitled30.ipynb	7 days ago
Untitled31.ipynb	7 days ago
Untitled32.ipynb	20 hours ago
Untitled33.ipynb	5 hours ago
Untitled34.ipynb	5 hours ago
Untitled35.ipynb	4 hours ago
Untitled36.ipynb	13 minutes ago
Untitled37.ipynb	10 minutes ago
Untitled38.ipynb	a minute ago

```python
# 4. Model Comparison (Before and After PCA)
# Apply K-Means on PCA-reduced data
kmeans_pca = KMeans(n_clusters=4, random_state=42)
clusters_pca = kmeans_pca.fit_predict(pca_data)

# Silhouette score after PCA
sil_score_pca = silhouette_score(pca_data, clusters_pca)
print(f"Silhouette Score after PCA: {sil_score_pca}")

# Visualize clustering after PCA
plt.scatter(pca_data[:, 0], pca_data[:, 1], c=clusters_pca, cmap='plasma', s=50)
plt.title("Clustering after PCA")
plt.xlabel("PCA Component 1")
plt.ylabel("PCA Component 2")
plt.colorbar(label='Cluster')
plt.show()

# 5. Evaluation Metrics
# Calculate inertia (within-cluster sum of squares) for K-Means
inertia = kmeans.inertia_
print(f"K-Means Inertia: {inertia}")

# Compare silhouette scores before and after PCA
print(f"Silhouette Score before PCA: {sil_score}")
print(f"Silhouette Score after PCA: {sil_score_pca}")
```

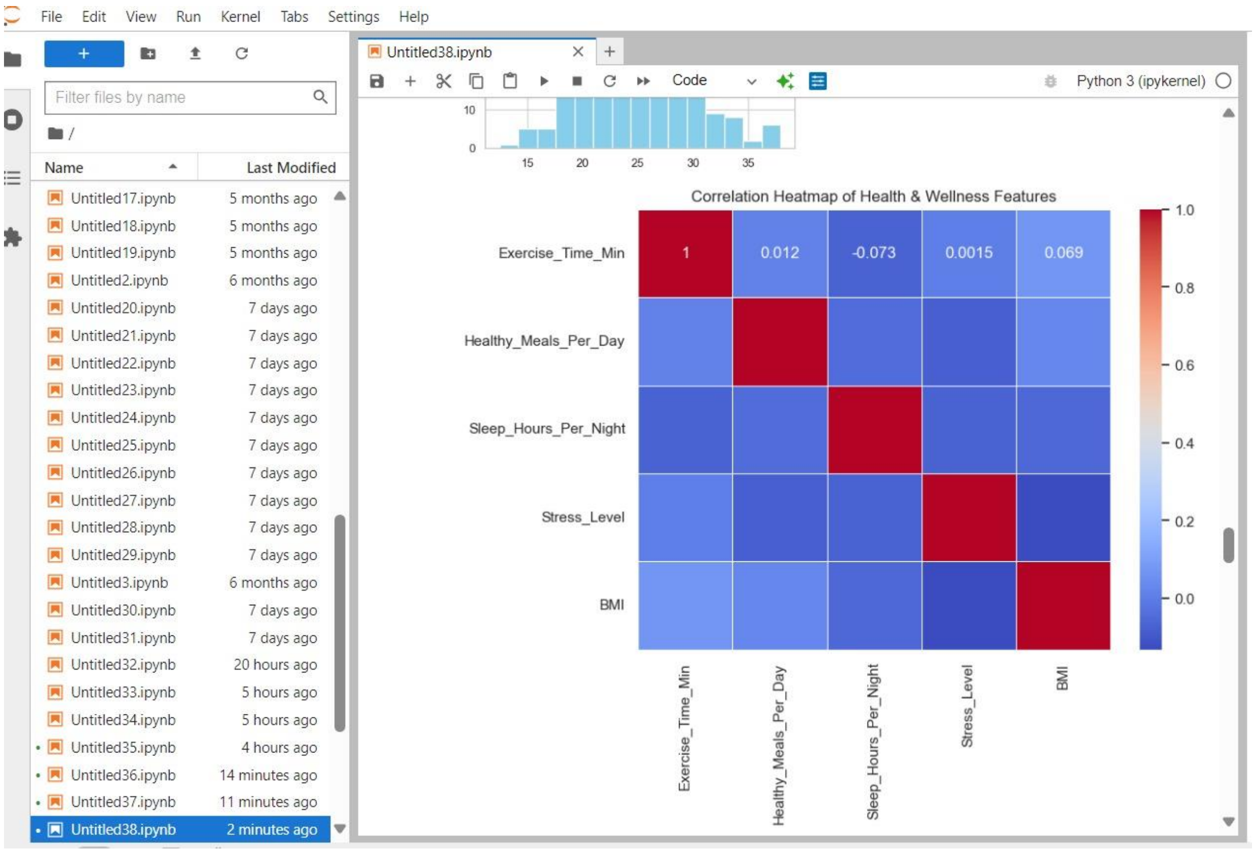
Histograms



Name	Last Modified
Untitled17.ipynb	5 months ago
Untitled18.ipynb	5 months ago
Untitled19.ipynb	5 months ago
Untitled2.ipynb	6 months ago
Untitled20.ipynb	7 days ago
Untitled21.ipynb	7 days ago
Untitled22.ipynb	7 days ago
Untitled23.ipynb	7 days ago
Untitled24.ipynb	7 days ago
Untitled25.ipynb	7 days ago
Untitled26.ipynb	7 days ago
Untitled27.ipynb	7 days ago
Untitled28.ipynb	7 days ago
Untitled29.ipynb	7 days ago
Untitled3.ipynb	6 months ago
Untitled30.ipynb	7 days ago
Untitled31.ipynb	7 days ago
Untitled32.ipynb	20 hours ago
Untitled33.ipynb	5 hours ago
Untitled34.ipynb	5 hours ago
Untitled35.ipynb	4 hours ago
Untitled36.ipynb	14 minutes ago
Untitled37.ipynb	11 minutes ago
Untitled38.ipynb	2 minutes ago

```
inertia = kmeans.inertia_
print(f"K-Means Inertia: {inertia}")

# Compare silhouette scores before and after PCA
print(f"Silhouette Score before PCA: {sil_score}")
print(f"Silhouette Score after PCA: {sil_score_pca}")
```

Correlation heatmap of health and wellness features



Name	Last Modified
Untitled17.ipynb	5 months ago
Untitled18.ipynb	5 months ago
Untitled19.ipynb	5 months ago
Untitled2.ipynb	6 months ago
Untitled20.ipynb	7 days ago
Untitled21.ipynb	7 days ago
Untitled22.ipynb	7 days ago
Untitled23.ipynb	7 days ago
Untitled24.ipynb	7 days ago
Untitled25.ipynb	7 days ago
Untitled26.ipynb	7 days ago
Untitled27.ipynb	7 days ago
Untitled28.ipynb	7 days ago
Untitled29.ipynb	7 days ago
Untitled3.ipynb	6 months ago
Untitled30.ipynb	7 days ago
Untitled31.ipynb	7 days ago
Untitled32.ipynb	20 hours ago
Untitled33.ipynb	5 hours ago
Untitled34.ipynb	5 hours ago
Untitled35.ipynb	4 hours ago
Untitled36.ipynb	14 minutes ago
Untitled37.ipynb	11 minutes ago
Untitled38.ipynb	2 minutes ago

This is an accurate step-by-step explanation of the Python code that does model comparison, principal component analysis (PCA), K-means clustering, exploratory data analysis (EDA), and PCA. An explanation of the aims and methods employed is provided in each section:

1. Analysis of exploratory statistics (EDA)

Goal: gain insight into the wellbeing indicators' distribution and connection all over the dataset.

Histograms:

Goal: Display the frequency distribution of each aspect of wellness (e.g, stress degree, eating regimen excellent, and amount of pastime).

Method: For each variable within the dataset, a histogram is plotted with the use of factsheet ().

Insight: Assists with figuring out whether the statistics are regularly disbursed or skewed.

Heatmap for Correlation:

Goal: Use a heatmap to visually constitute correlations between variables.

Method: To plot correlations among parameters consisting of BMI, strain, food, and exercise, use sns.Heatmap().

Insight: Strong effective or negative correlations can indicate viable institutions. For instance, a high BMI is associated with a decrease level of exercising.

Pair plot:

The aim is to visualize each feature's pairwise associations.

Method: For each pair of functions, sns.Pairplot() creates a scatter plot matrix.

Visually identifies styles or clusters and enables perceiving possible outliers.

Boxplots:

Goal: Understand the distribution and become aware of anomalies in the wellbeing facts.

Method: To display the variability and distribution of traits, sns.Boxplot() is utilized.

Observation: Boxplots are beneficial for locating severe values that might affect clustering.

2. K-Means Clustering

Goal: Divide patients into several organizations consistent with their well-being traits.

Data Scaling:

The purpose is to normalize the facts in order that the weight of every characteristic is the same.

Method: To normalize features to have a zero imply and unit variance, use StandardScaler().

Observation: Assures that characteristics with disparate scales (e.G., weight as opposed to sleep duration) do no longer control the clustering method.

K-Means Grouping:

The intention is to categorise patients in step with how like their health markers are.

Method: Patients are divided into four clusters the use of KMeans() with n_clusters=four.

By assembling patients with comparable wellbeing practices, perception: Assists in locating key styles within the dataset.

Score for Silhouette:

Goal: Assess the nice of clustering.

Method: silhouette_score() determines the degree of cluster separation. Better-defined clusters are indicated by using a better score (closer to one).

A hint: Silhouette ratings aid in confirming whether or not the corporations shaped through the clustering set of rules are clearly divided.

Cluster Visualization:

Goal: Illustrate the clustering of sufferers.

Clusters are visualized within the unique function area the usage of the sns.Pairplot() approach.

Understanding: Cluster scatter plots offer a herbal comprehension of ways sufferers are labeled according to shared characteristics.

3. PCA, or fundamental element analysis

Goal: Reduce the dataset's complexity whilst preserving the greatest quantity of information.

Reducing PCA Dimensionality:

Goal: Reduce the quantity of variables (functions) within the dataset without greatly changing its content so that you can make it less difficult.

Method: The statistics is reduced to two additives via PCA(components=2), which account for much of the variance.

Understanding: This enhances the readability of clustering and makes high-dimensional records less complicated to peer into a 2D image.

Explanation of Variance

Goal: Illustrate the quantity of variation (facts retained) by every PCA element.

Method: explained_variance_ratio_ gives the percentage of variance that every PCA issue accounts for.

Insight: The bulk of the records within the information is captured through the primary components, which account for approximately 80% of the variance.

Scatter Plot for PCA:

Goal: After dimensionality reduction, visualize the facts.

Method: The first PCA additives, colored by means of the preliminary clusters, are used to construct a scatter plot.

Understanding: This visual useful resource clarifies how frivolously spaced out the clusters are in a lower-dimensional region.

4. Model Comparison (Before and After PCA)

Goal: Examine how applying PCA impacts clustering outcomes by contrasting the clustering performance before and after dimensionality reduction.

K-Means for Data Reduced via PCA:
Goal: Reapply K-Means clustering to the data that has been reduced via PCA.
Method: To produce new clusters, KMeans() is applied to the 2D PCA components.
Understanding: Clustering following PCA can lead to more distinct clusters and helps with noise reduction.

After PCA, silhouette score:

Goal: Assess the effect of dimensionality discount on the performance of clustering.

Method: After clustering on PCA-reduced information, silhouette_score() is computed.

Observation: A greater silhouette score following PCA implies that cluster cohesion become improved by means of dimensionality discount.

Inertia (Sum of Squares Within-Cluster):

Goal: Determine the clusters' compactness (smaller inertia = higher clusters).

Method: The sum of squared distances among a point and its closest cluster middle is calculated the usage of kmeans.Inertia_.

Understanding: By evaluating the inertia cost earlier than and after PCA, you can determine the volume to which dimensionality reduction enhances clustering. This value offers a notion of how compact the clusters.

5. Evaluation Measures

Goal: Evaluate the clustering findings' satisfactory.

The general squared distances between each factor and the middle of every cluster are used to calculate inertia. Better clustering is indicated via a lower inertia.

Comparison of Silhouette Scores:

Prior to PCA: Evaluates the first-class clustering in the initial characteristic area.

Assesses the best of clustering inside the condensed 2D area following PCA.

Understanding: Understanding the effect of dimensionality discount on cluster separation can be gained by way of evaluating the silhouette rankings acquired prior to and following PCA.

Concluding comments and guidelines

Learnings from Grouping:

You can interpret the affected person segments and provide tailored wellbeing interventions based at the clustering. For instance, mental fitness packages and bodily interest may be beneficial for humans who have excessive degrees of strain and little exercise.

PCA's effectiveness:

PCA improves the interpretability of clustering findings through lowering the number of facts inside the statistics at the same time as preserving most of it.

Your file and presentation can be higher organized if you follow this thorough breakdown. Every segment accentuates the goal, strategies, and revelations received from your exam. Tell me if you require some other information!