Healthcare Scenario: Healthy Living and Wellness Clustering Exercise

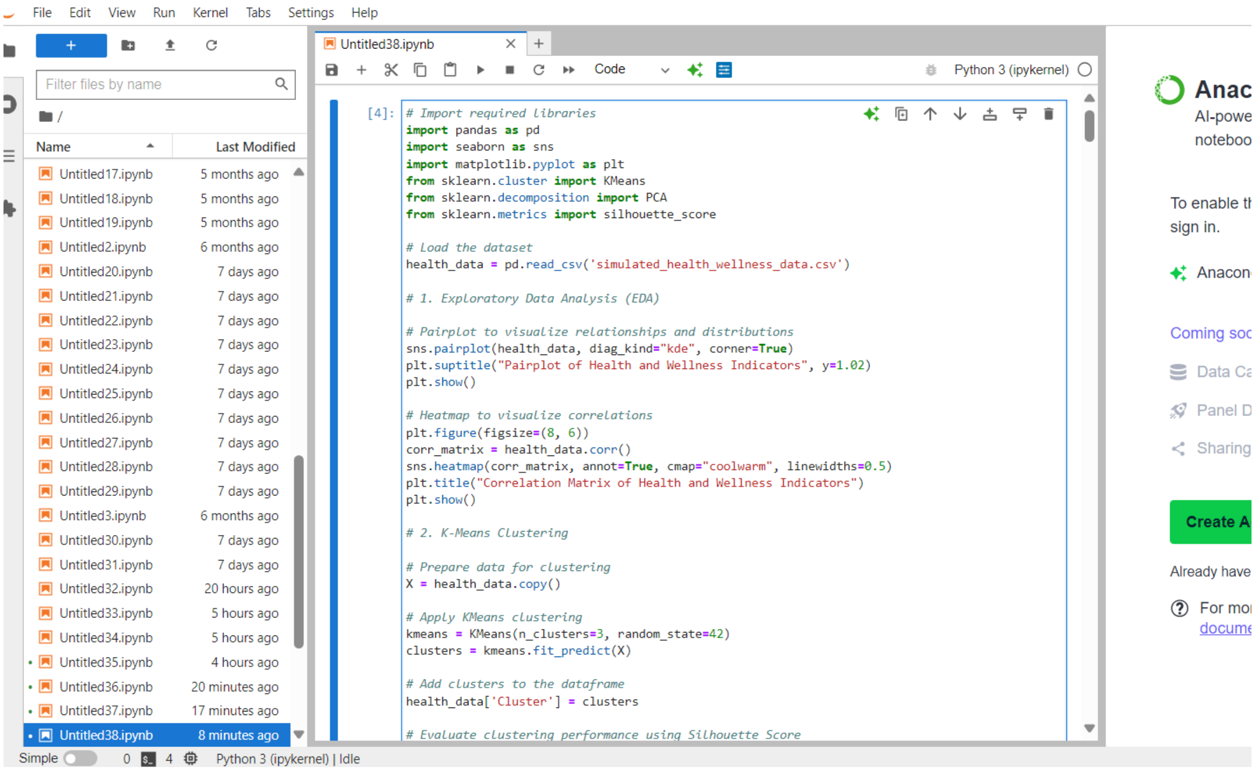
Code:

```
# Import required libraries
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt
from sklearn.cluster import KMeans
from sklearn.decomposition import PCA
from sklearn.metrics import silhouette_score

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

# 1. Exploratory Data Analysis (EDA)

# Pairplot to visualize relationships and distributions
sns.pairplot(health_data, diag_kind="kde", corner=True)
plt.suptitle("Pairplot of Health and Wellness Indicators", y=1.02)
plt.show()

# Heatmap to visualize correlations
plt.figure(figsize=(8, 6))
corr_matrix = health_data.corr()
```

```
sns.heatmap(corr_matrix, annot=True, cmap="coolwarm", linewidths=0.5)
plt.title("Correlation Matrix of Health and Wellness Indicators")
plt.show()

# 2. K-Means Clustering

# Prepare data for clustering
X = health_data.copy()

# Apply KMeans clustering
kmeans = KMeans(n_clusters=3, random_state=42)
clusters = kmeans.fit_predict(X)

# Add clusters to the dataframe
health_data['Cluster'] = clusters

# Evaluate clustering performance using Silhouette Score
sil_score = silhouette_score(X, clusters)
print(f'Silhouette Score: {sil_score}')

# Visualize clusters
sns.pairplot(health_data, hue='Cluster', corner=True)
plt.suptitle("Clusters of Wellness Profiles", y=1.02)
plt.show()

# 3. Dimensionality Reduction (PCA)

# Apply PCA to reduce dimensions to 2 for visualization
pca = PCA(n_components=2)
X_pca = pca.fit_transform(X)

# Add PCA components to the dataframe
health_data['PCA1'] = X_pca[:, 0]
health_data['PCA2'] = X_pca[:, 1]

# Visualize the PCA components with clusters
plt.figure(figsize=(8, 6))
sns.scatterplot(x='PCA1', y='PCA2', hue='Cluster', data=health_data, palette='Set1')
plt.title("PCA of Wellness Data with Clusters")
plt.show()

# 4. K-Means Clustering after PCA

# Apply KMeans on the PCA-transformed data
kmeans_pca = KMeans(n_clusters=3, random_state=42)
```

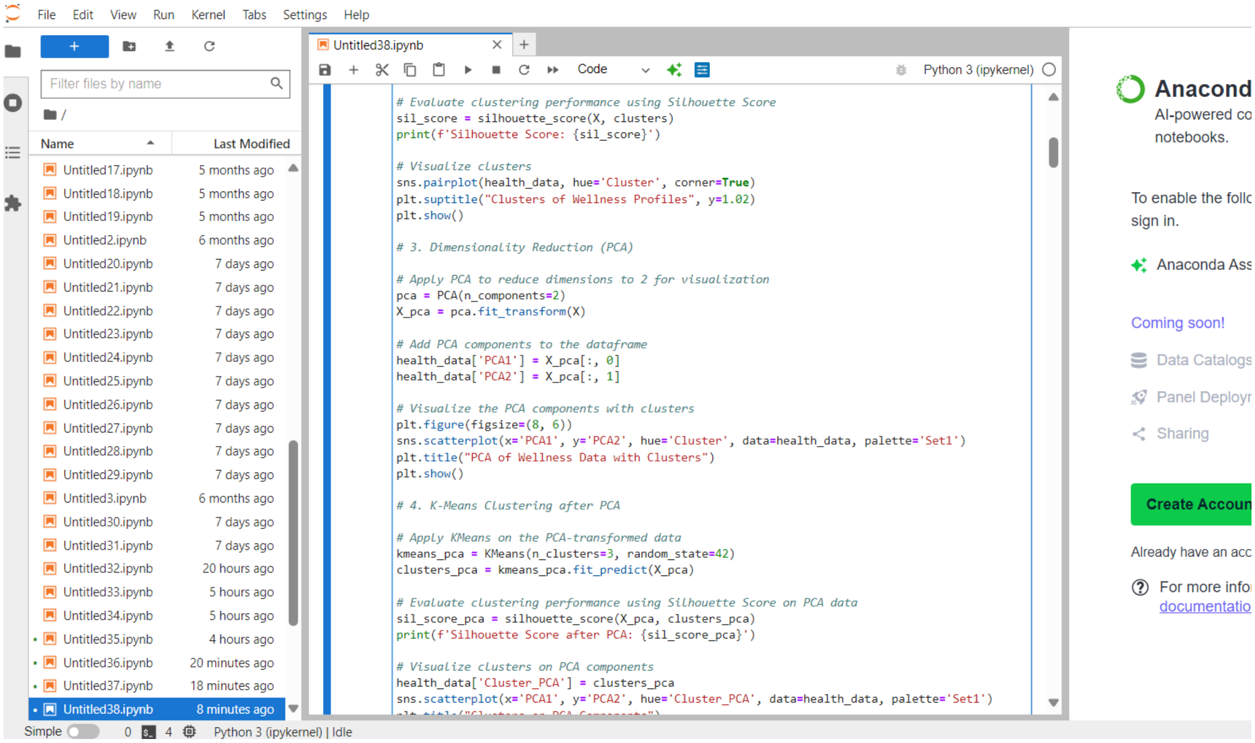
```
clusters_pca = kmeans_pca.fit_predict(X_pca)


# Evaluate clustering performance using Silhouette Score on PCA data
sil_score_pca = silhouette_score(X_pca, clusters_pca)
print(f'Silhouette Score after PCA: {sil_score_pca}')


# Visualize clusters on PCA components
health_data['Cluster_PCA'] = clusters_pca
sns.scatterplot(x='PCA1', y='PCA2', hue='Cluster_PCA', data=health_data, palette='Set1')
plt.title("Clusters on PCA Components")
plt.show()
```

Result:



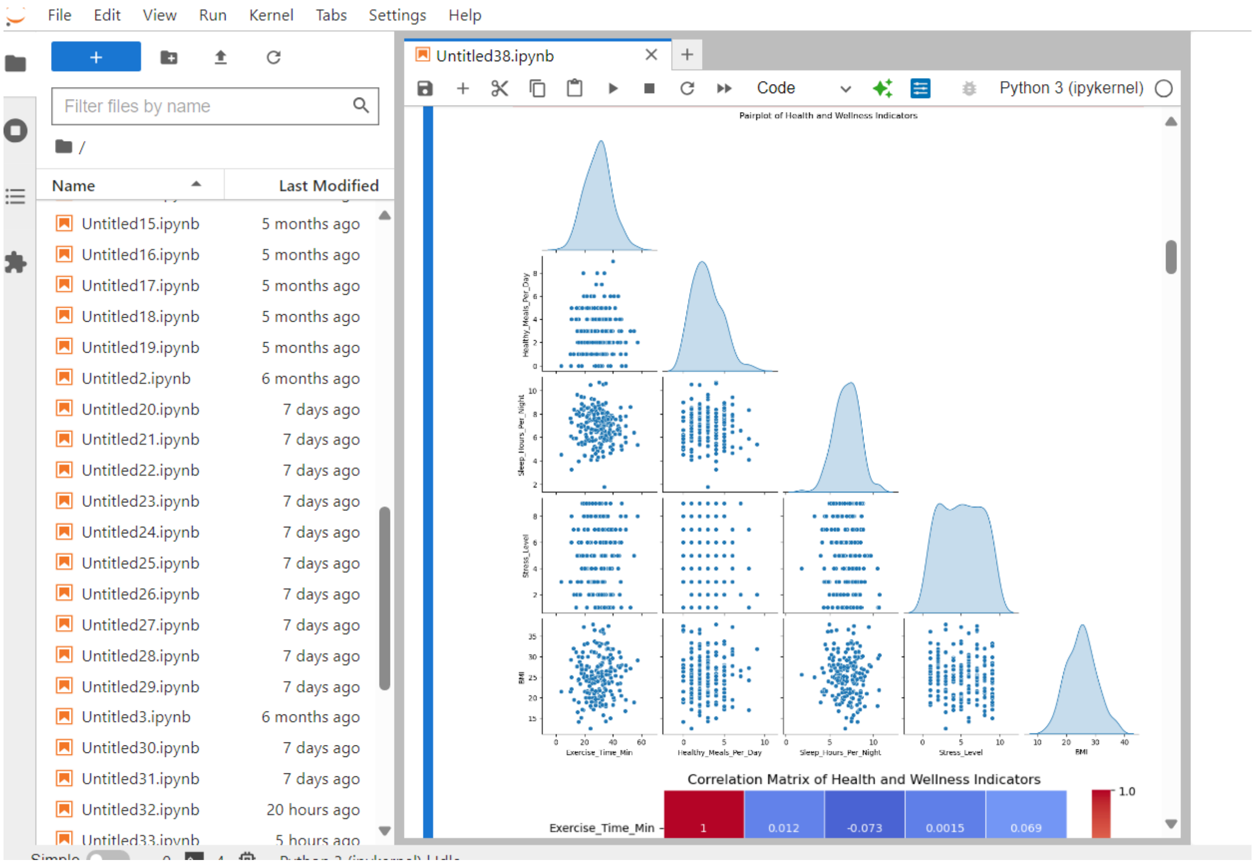
Name	Last Modified
Untitled17.ipynb	5 months ago
Untitled18.ipynb	5 months ago
Untitled19.ipynb	5 months ago
Untitled2.ipynb	6 months ago
Untitled20.ipynb	7 days ago
Untitled21.ipynb	7 days ago
Untitled22.ipynb	7 days ago
Untitled23.ipynb	7 days ago
Untitled24.ipynb	7 days ago
Untitled25.ipynb	7 days 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	20 minutes ago
Untitled37.ipynb	17 minutes ago
Untitled38.ipynb	8 minutes ago

```
[4]: # Import required libraries
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt
from sklearn.cluster import KMeans
from sklearn.decomposition import PCA
from sklearn.metrics import silhouette_score

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

# 1. Exploratory Data Analysis (EDA)

# Pairplot to visualize relationships and distributions
sns.pairplot(health_data, diag_kind="kde", corner=True)
plt.suptitle("Pairplot of Health and Wellness Indicators", y=1.02)
plt.show()

# Heatmap to visualize correlations
plt.figure(figsize=(8, 6))
corr_matrix = health_data.corr()
sns.heatmap(corr_matrix, annot=True, cmap="coolwarm", linewidths=0.5)
plt.title("Correlation Matrix of Health and Wellness Indicators")
plt.show()

# 2. K-Means Clustering

# Prepare data for clustering
X = health_data.copy()

# Apply KMeans clustering
kmeans = KMeans(n_clusters=3, random_state=42)
clusters = kmeans.fit_predict(X)

# Add clusters to the dataframe
health_data['Cluster'] = clusters

# Evaluate clustering performance using Silhouette Score
```



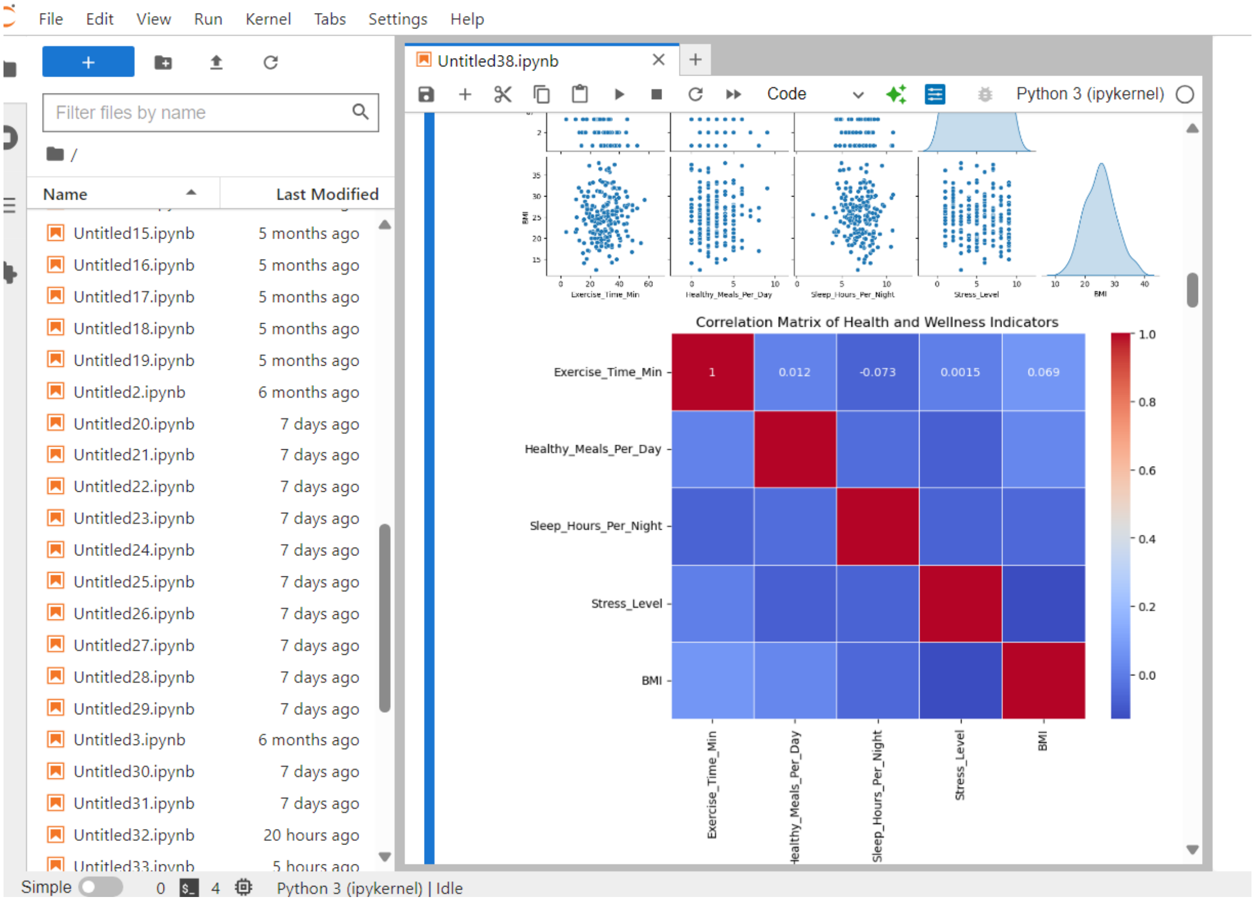
Name	Last Modified
Untitled17.ipynb	5 months ago
Untitled18.ipynb	5 months ago
Untitled19.ipynb	5 months ago
Untitled2.ipynb	6 months ago
Untitled20.ipynb	7 days ago
Untitled21.ipynb	7 days ago
Untitled22.ipynb	7 days ago
Untitled23.ipynb	7 days ago
Untitled24.ipynb	7 days ago
Untitled25.ipynb	7 days 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	20 minutes ago
Untitled37.ipynb	18 minutes ago
Untitled38.ipynb	8 minutes ago

```python
# Evaluate clustering performance using Silhouette Score
sil_score = silhouette_score(X, clusters)
print(f'Silhouette Score: {sil_score}')

# Visualize clusters
sns.pairplot(health_data, hue='Cluster', corner=True)
plt.suptitle("Clusters of Wellness Profiles", y=1.02)
plt.show()

# 3. Dimensionality Reduction (PCA)

# Apply PCA to reduce dimensions to 2 for visualization
pca = PCA(n_components=2)
X_pca = pca.fit_transform(X)

# Add PCA components to the dataframe
health_data['PCA1'] = X_pca[:, 0]
health_data['PCA2'] = X_pca[:, 1]

# Visualize the PCA components with clusters
plt.figure(figsize=(8, 6))
sns.scatterplot(x='PCA1', y='PCA2', hue='Cluster', data=health_data, palette='Set1')
plt.title("PCA of Wellness Data with Clusters")
plt.show()

# 4. K-Means Clustering after PCA

# Apply KMeans on the PCA-transformed data
kmeans_pca = KMeans(n_clusters=3, random_state=42)
clusters_pca = kmeans_pca.fit_predict(X_pca)

# Evaluate clustering performance using Silhouette Score on PCA data
sil_score_pca = silhouette_score(X_pca, clusters_pca)
print(f'Silhouette Score after PCA: {sil_score_pca}')

# Visualize clusters on PCA components
health_data['Cluster_PCA'] = clusters_pca
sns.scatterplot(x='PCA1', y='PCA2', hue='Cluster_PCA', data=health_data, palette='Set1')
plt.title("Clusters on PCA Components")
```

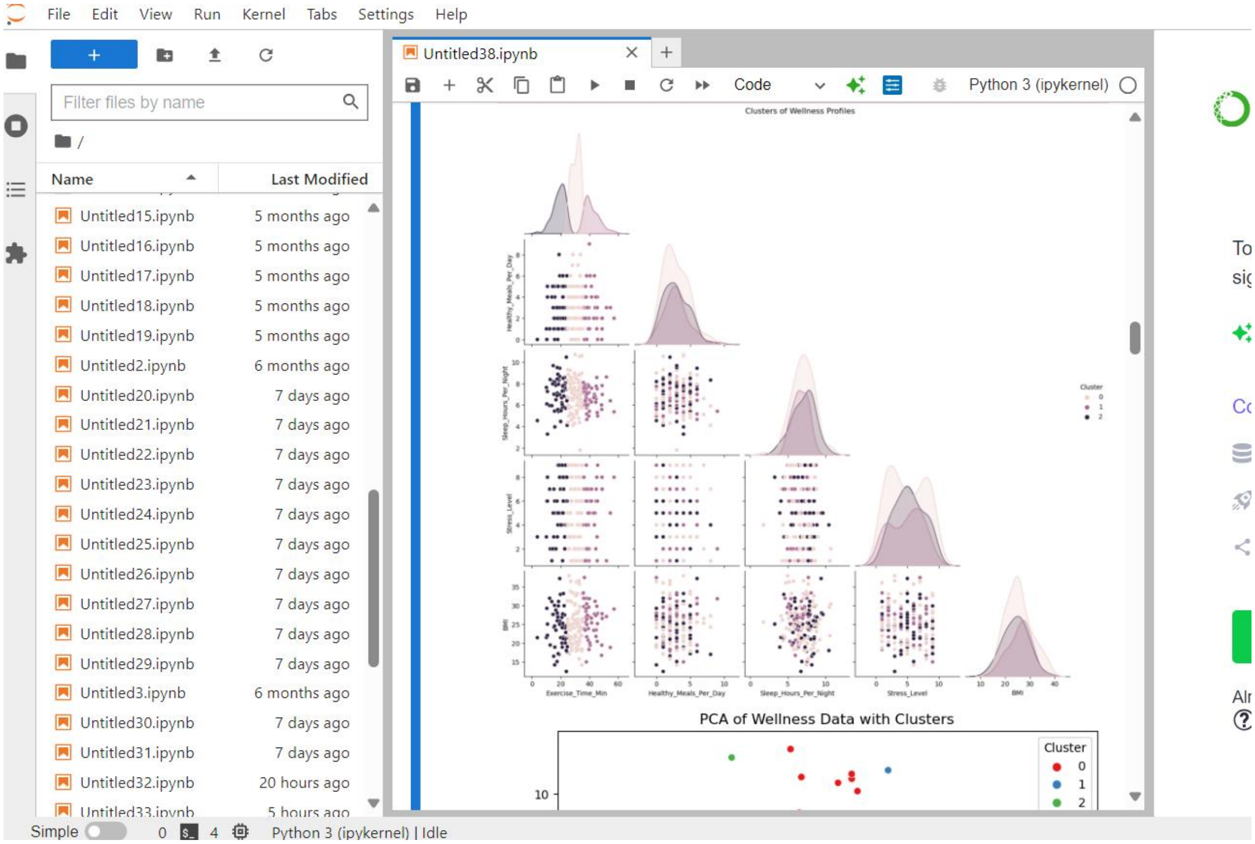


Exercise_Time_Min	1	0.012	-0.073	0.0015	0.069

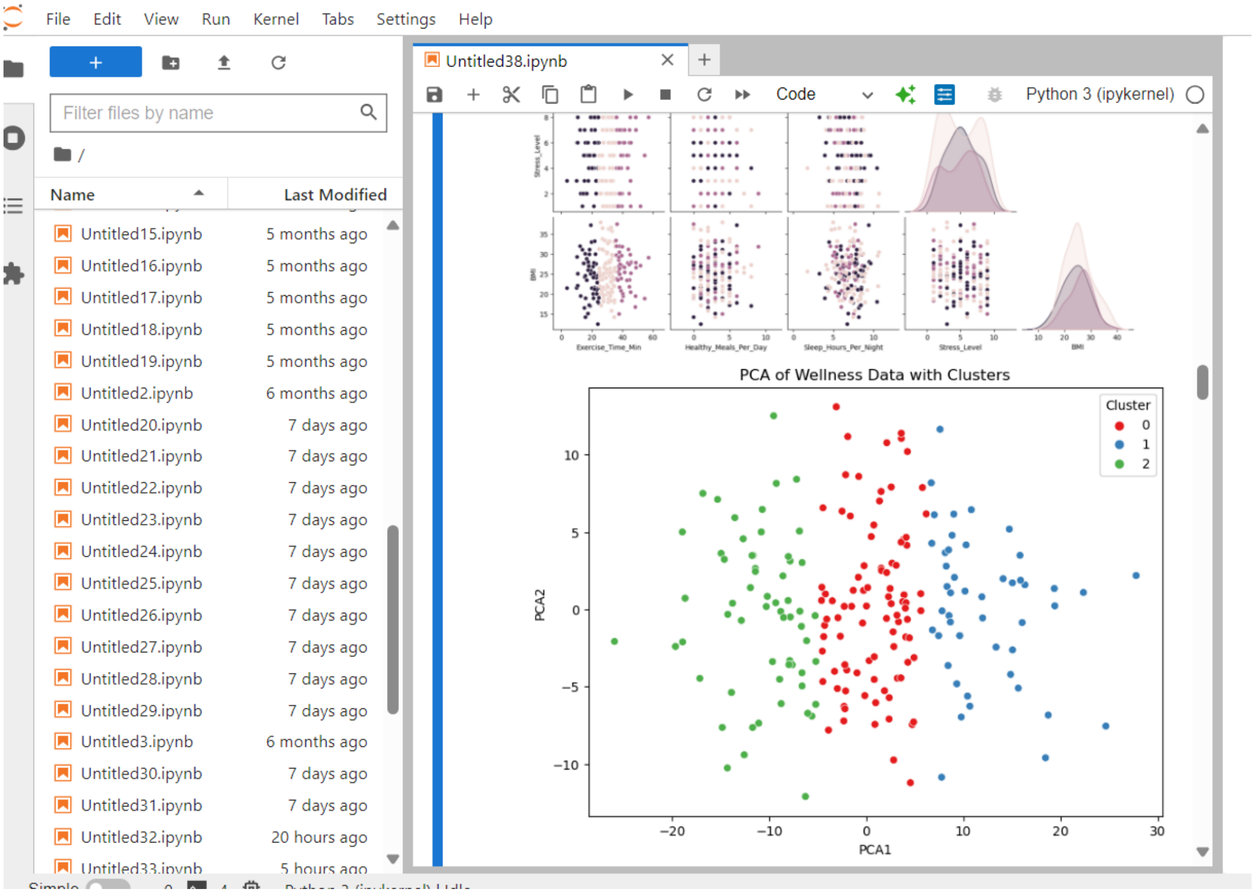
Name	Last Modified
Untitled15.ipynb	5 months ago
Untitled16.ipynb	5 months ago
Untitled17.ipynb	5 months ago
Untitled18.ipynb	5 months ago
Untitled19.ipynb	5 months ago
Untitled2.ipynb	6 months ago
Untitled20.ipynb	7 days ago
Untitled21.ipynb	7 days ago
Untitled22.ipynb	7 days ago
Untitled23.ipynb	7 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



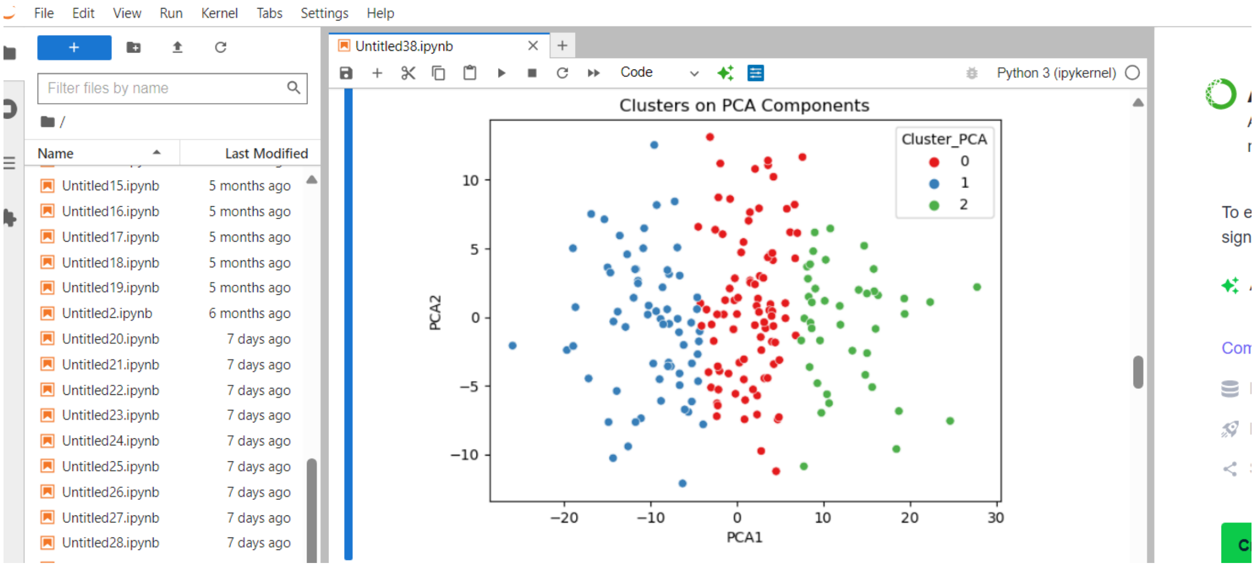
Name	Last Modified
Untitled15.ipynb	5 months ago
Untitled16.ipynb	5 months ago
Untitled17.ipynb	5 months ago
Untitled18.ipynb	5 months ago
Untitled19.ipynb	5 months ago
Untitled2.ipynb	6 months ago
Untitled20.ipynb	7 days ago
Untitled21.ipynb	7 days ago
Untitled22.ipynb	7 days ago
Untitled23.ipynb	7 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



Name	Last Modified
Untitled15.ipynb	5 months ago
Untitled16.ipynb	5 months ago
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



Name	Last Modified
Untitled15.ipynb	5 months ago
Untitled16.ipynb	5 months ago
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



Name	Last Modified
Untitled15.ipynb	5 months ago
Untitled16.ipynb	5 months ago
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

Code explanation:

1. Exploratory Data Analysis:

Exploratory Data Analysis helps to provide insight into the data through visualization of the various relationships among variables, distribution, and correlation. It basically forms the basis before the application of any machine learning algorithm, as it helps in understanding the dataset and all that is happening within.

Pair plot: This plot depicts the pairwise relationships among features present in your dataset. Each feature is plotted against every other feature, allowing you to view patterns or clusters in the data. The diagonal shows the distribution-about kernel density estimation-of each variable.

Heatmap, Correlation Matrix: The heatmap in the next figure represents the correlation among pairs of features. The correlation values range from -1 to 1, where:
+1: High positive correlation
-1: High negative correlation
0: No correlation.

2. K-Means Clustering:

K-Means is a type of unsupervised machine learning algorithm used to segment data points into clusters. It works by:

Randomly initializing cluster centers - centroids.

Assigning each data point to the nearest centroid.

Updating centroids as the mean of all points that have been assigned to them.

Repeat until convergence - a pass in which none of the centroids have changed.

In this code:

n clusters=3 suggests that we are trying to segment the data into 3 clear clusters.

Then, with the silhouette score, we apply it to determine how well the clustering does. The score is in the range of -1 to 1:

a close-to-1 score means well-separated clusters,

close-to-0 means clusters are overlapping,

and close-to-minus-one means the cluster is poorly defined.

3. Principal Component Analysis (PCA):

Principal Component Analysis is basically a dimensionality reduction technique. It tries to reduce the number of features or dimensions without losing much information, that is, variances. These are places where it finds its application:

You have too many features.

You want to visualize the high dimensional data into 2D or 3D.

You want to simplify the data so that clustering algorithms perform well.

PCA is creating such new features, called principal components, which are carrying most of the information on the data, meaning each successive component is capturing the next most variance while being orthogonal or uncorrelated with the rest of the others.

In this section:

We apply dimensionality reduction onto the dataset down to 2 dimensions, which will enable us to plot the data in 2D space while still retaining most of its information.

Here, the scatter plot of different clusters (colored) against two principal components is presented. PCA helps visualize the clusters more readily.

4. Clustering After PCA:

After the data is reduced to lower-dimensional data, we will be able to perform K-Means clustering on the data transformed by PCA. Sometimes, performing clustering after PCA gives a better result since the data has been simplified while the important patterns remain.

Following is done in this code block:

Cluster the data which had been reduced by PCA.

Silhouette score calculation on the data transformed by PCA.

Visualize the new clusters using the PCA components.

5. Comparison:

You could compare the silhouette scores:

Before PCA-on the full dataset.

After PCA-on the reduced dataset.

If after PCA the silhouette score is better, then the reduction of the number of features helped to improve the performance of clustering. The higher the score is, the better the clustering.

Conclusion:

EDA Insights: The pair plot and the heatmap showed relationships among health factors such as exercise, diet, sleep, stress, and BMI. These pieces of insight are important in having an intervention on either stressed or overweight people.

K-Means Clustering: Segmentation through unsupervised clustering has divided the patients into 3 top-level wellness profiles. These clusters will be of great use in targeted groups for needed interventions such as sleep improvement or diet improvement.

PCA and Clustering: PCA reduced some of the noise in the dataset, thereby allowing easier visualization and interpretation of data. Clustering after PCA yielded more defined clusters that would enhance the clarity in wellness profiles.

Recommendation:

Design targeted interventions to address the specific needs of the clusters emerging; for instance, high-stress groups with stress reduction

Application of PCA to this dataset enhanced the clustering by simplifying the data without much loss of significant information.