

## **KMeans Clustering and PCA Analysis on Dataset**

Niranjana Rao

San Jose State University

DATA 230 - Business Intelligence and Data Visualization

November 20, 2024

## Abstract

This report outlines the implementation of KMeans clustering and dimensionality reduction using PCA (Principal Component Analysis) on the dataset. The objective is to group data into clusters and reduce its dimensionality for better visualization and understanding.

## 1. Introduction

The dataset contains various attributes such as health-related data, employment, insurance status, and other demographic information. KMeans clustering was applied to identify inherent groupings in the data, while PCA was used to reduce the dataset's dimensionality, enabling visualization and feature analysis.

## 2. Data Preprocessing

Before applying the techniques, the dataset was preprocessed to ensure compatibility with clustering and PCA algorithms. Key preprocessing steps included:

1. Dropping irrelevant columns (e.g., non-numeric and ID-like columns).
2. Handling missing values by filtering or imputing them.
3. Scaling numerical data for uniformity.

### Code Snippet:

```
file_path = 'DentalVisit-Clean.csv'

data = pd.read_csv(file_path)

# Selecting numerical columns for clustering
numerical_data = data.select_dtypes(include=['float64', 'int64']).dropna()

# Scaling the data
scaler = StandardScaler()

scaled_data = scaler.fit_transform(numerical_data)
```

## 3. KMeans Clustering

KMeans clustering was employed to group the data points into clusters based on their similarities.

### 3.1 Elbow Method

The Elbow Method was used to determine the optimal number of clusters by plotting the inertia (sum of squared distances to cluster centroids) against the number of clusters.

**Code Snippet:**

```
# Determining the optimal number of clusters using the elbow method
```

```
inertia = []
```

```
cluster_range = range(1, 11)
```

```
for k in cluster_range:
```

```
    kmeans = KMeans(n_clusters=k, random_state=42)
```

```
    kmeans.fit(scaled_data)
```

```
    inertia.append(kmeans.inertia_)
```

```
# Plotting the elbow graph
```

```
plt.figure(figsize=(8, 5))
```

```
plt.plot(cluster_range, inertia, marker='o')
```

```
plt.title('Elbow Method for Optimal Clusters')
```

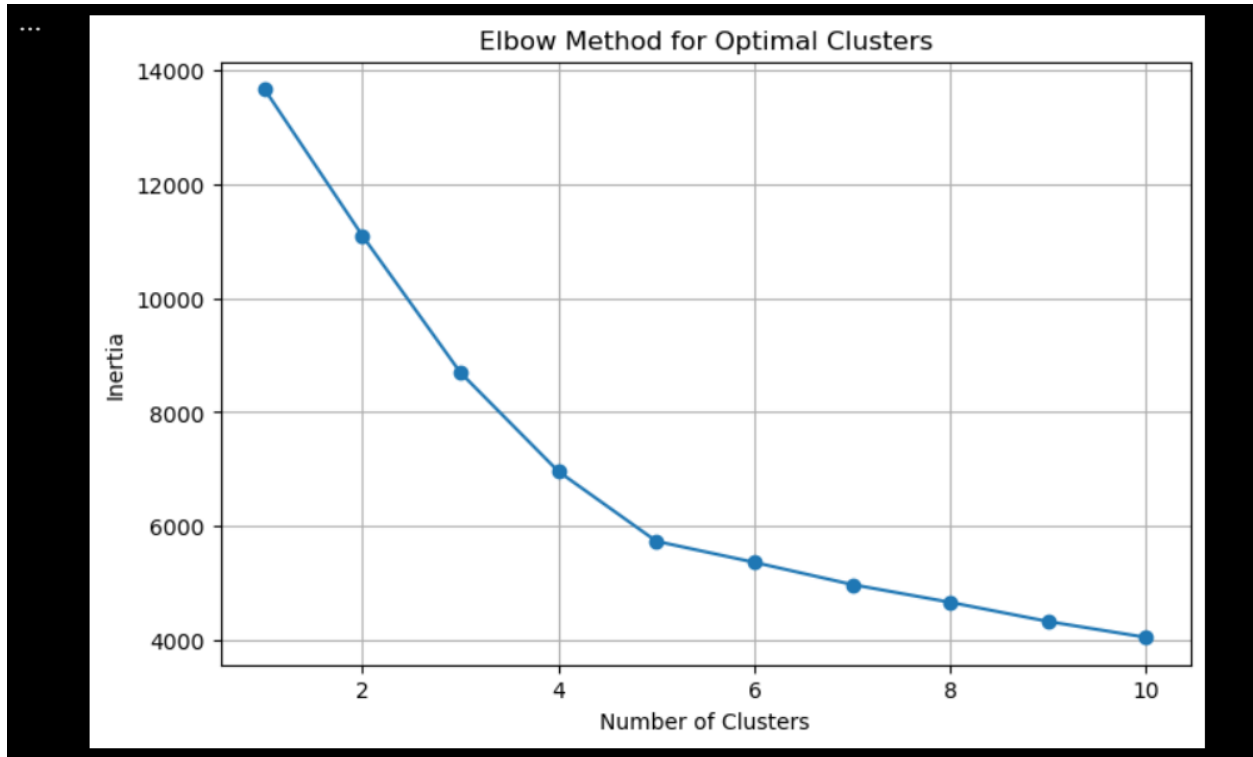
```
plt.xlabel('Number of Clusters')
```

```
plt.ylabel('Inertia')
```

```
plt.grid(True)
```

```
plt.show()
```

**Screenshot:**



### 3.2 Clustering Results

Using the optimal number of clusters determined from the Elbow Method, KMeans clustering was applied. The clusters were labeled and appended to the dataset for further analysis.

**Code Snippet:**

```
# Applying KMeans with an optimal number of clusters (e.g., 3 based on the elbow graph)
```

```
optimal_clusters = 3
```

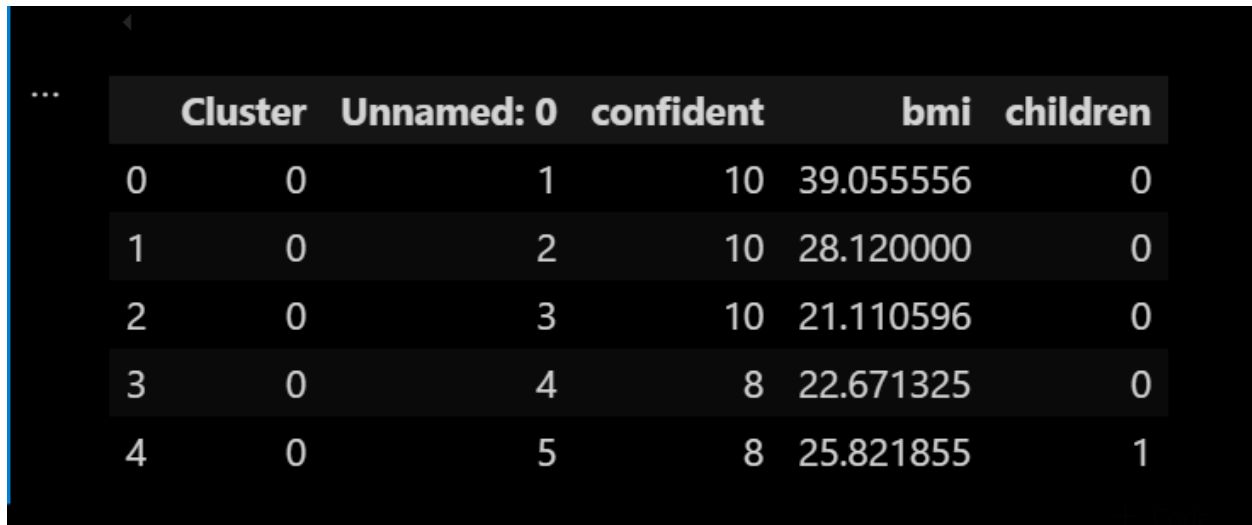
```
kmeans = KMeans(n_clusters=optimal_clusters, random_state=42)
```

```
data['Cluster'] = kmeans.fit_predict(scaled_data)
```

```
# Displaying the first few rows with cluster labels
```

```
data[['Cluster'] + numerical_data.columns.tolist()].head()
```

Screenshot:



...	Cluster	Unnamed: 0	confident	bmi	children
0	0	1	10	39.055556	0
1	0	2	10	28.120000	0
2	0	3	10	21.110596	0
3	0	4	8	22.671325	0
4	0	5	8	25.821855	1

#### 4. Dimensionality Reduction using PCA

Principal Component Analysis (PCA) was applied to reduce the dataset's dimensions, allowing for a better understanding and visualization of high-dimensional data.

##### 4.1 Explained Variance Ratio

The explained variance ratio was computed to determine the contribution of each principal component to the total variance.

##### Code Snippet:

```
# Applying PCA to the scaled data
```

```
pca = PCA(n_components=2) # Reducing to 2 dimensions for visualization
```

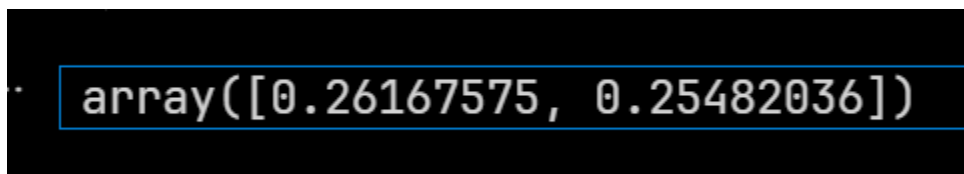
```
pca_result = pca.fit_transform(scaled_data)
```

```
# Adding PCA results back to the dataframe for visualization
```

```
data['PCA1'] = pca_result[:, 0]
```

```
data['PCA2'] = pca_result[:, 1]
```

Screenshot:



```
array([0.26167575, 0.25482036])
```

##### 4.2 2D Visualization

The data was reduced to two principal components, and the clusters identified by KMeans were visualized in a 2D scatter plot.

**Code Snippet:**

```
# Visualizing the PCA results with cluster labels
```

```
plt.figure(figsize=(10, 6))
```

```
for cluster in range(optimal_clusters):
```

```
    cluster_data = data[data['Cluster'] == cluster]
```

```
    plt.scatter(cluster_data['PCA1'], cluster_data['PCA2'], label=f'Cluster {cluster}')
```

```
plt.title('PCA Results with KMeans Clusters')
```

```
plt.xlabel('PCA1')
```

```
plt.ylabel('PCA2')
```

```
plt.legend()
```

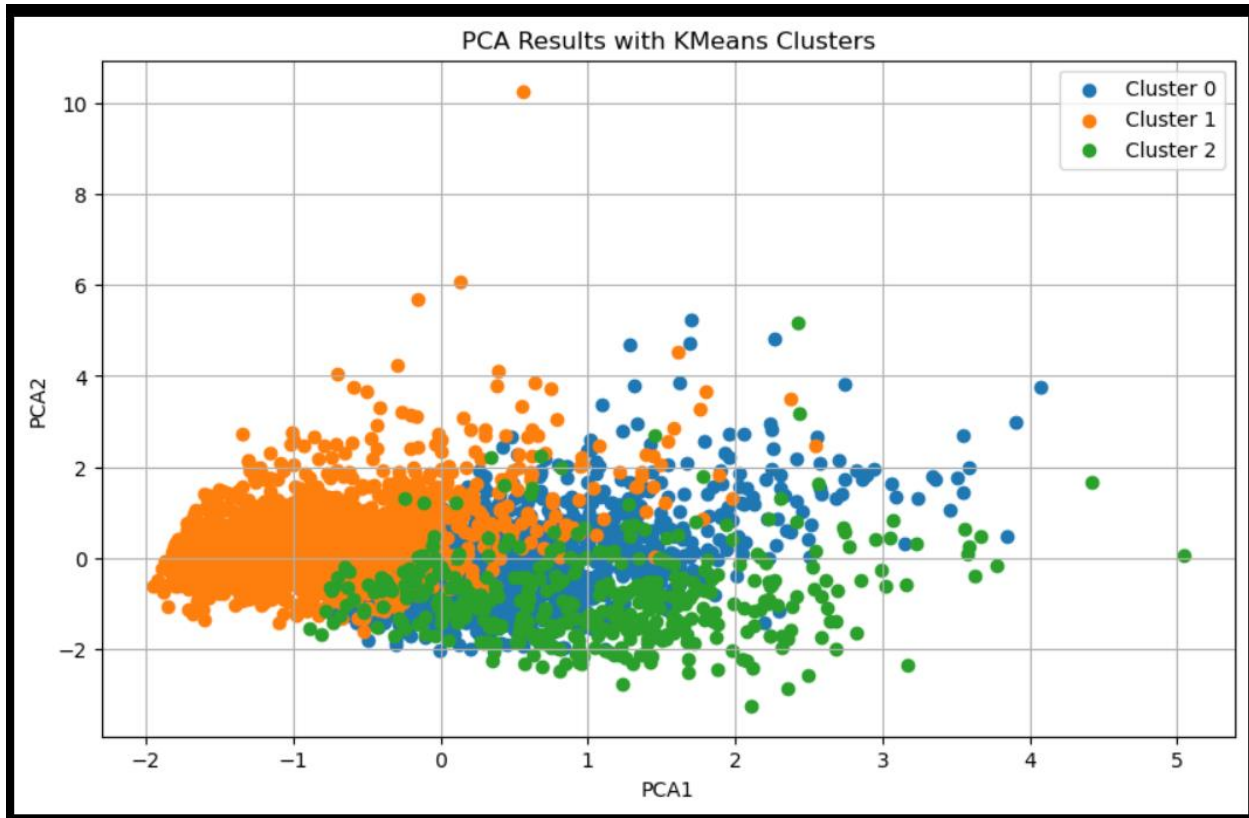
```
plt.grid(True)
```

```
plt.show()
```

```
# Explained variance ratio to understand PCA contribution
```

```
pca.explained_variance_ratio_
```

**Screenshot:**



## 5. Insights and Observations

### 1. Clustering Insights:

- The dataset was successfully grouped into distinct clusters.
- Each cluster represents a unique grouping of individuals based on the features analyzed.

### 2. PCA Insights:

- The PCA scatter plot highlights how the clusters are distributed in the reduced 2D space.
- The explained variance ratio indicates that the top two components account for a significant portion of the variance.

## 6. Conclusion

This analysis demonstrates the use of KMeans clustering to identify patterns in data and PCA to reduce dimensionality. These techniques are powerful for exploratory data analysis and understanding complex datasets.

## References

- Bishop, C. M. (2006). *Pattern Recognition and Machine Learning*. Springer.
- Jolliffe, I. T., & Cadima, J. (2016). Principal component analysis: A review and recent developments. *Philosophical Transactions of the Royal Society A*, 374(2065). <https://doi.org/10.1098/rsta.2015.0202>
- Pedregosa, F., et al. (2011). Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12, 2825–2830.