Data Science Project Report: Clustering, Association Rule Mining, Sensitivity Analysis, and Customer Analysis

Bhavana Ginuga

1 Introduction

This report presents the exploration and analysis of various datasets using four key machine learning techniques: DBSCAN clustering, Association Rule Mining (Market Basket Analysis), Sensitivity Analysis in Management Science, and Customer Analysis using K-Means. The tasks cover data preprocessing, feature engineering, algorithm application, and key insights extraction from the datasets.

2 Task 1:Custoer Analysis using K-Means Clustering

2.1 Objective

The objective of this task was to implement a custom K-Means clustering algorithm manually to segment customers based on their purchasing behavior. The dataset used for this analysis is **Mall Customer Segmentation Data**. This dataset includes demographic features such as customer age, gender, income, and spending score to explore underlying customer behavior patterns.

2.2 Data Preprocessing and Exploratory Data Analysis (EDA)

The customer dataset underwent preprocessing and exploratory analysis:

- **Feature Selection and Engineering**: Relevant features such as annual income and spending score were selected for clustering. Gender was converted to numerical form if necessary.
- **Normalization**: All selected features were normalized to ensure equal scaling, as K-Means is sensitive to feature magnitudes.
- **EDA**: Histograms and pair plots were employed to assess feature distributions and potential cluster structures before modeling.

2.3 Determining the Optimal Number of Clusters

To identify the ideal number of clusters (K), the Elbow Method was utilized. The within-cluster sum of squares (WCSS) was computed for a range of K values, and the point of inflection ("elbow") was used to estimate the optimal K.

2.4 K-Means Clustering Implementation (Custom Algorithm)

The clustering algorithm was implemented from scratch using the following steps:

- **Initialization**: Random selection of K centroids from the dataset.
- **Cluster Assignment**: Each data point was assigned to the closest centroid using the Euclidean distance metric.
- **Centroid Update**: Centroids were recalculated by taking the mean of data points assigned to each cluster.
- **Convergence Check**: The iteration process continued until centroids stabilized, i.e., no significant change occurred between iterations.

2.5 Visualizations

2.5.1 Elbow Graph

A plot of WCSS against different values of K was used to identify the optimal cluster count.

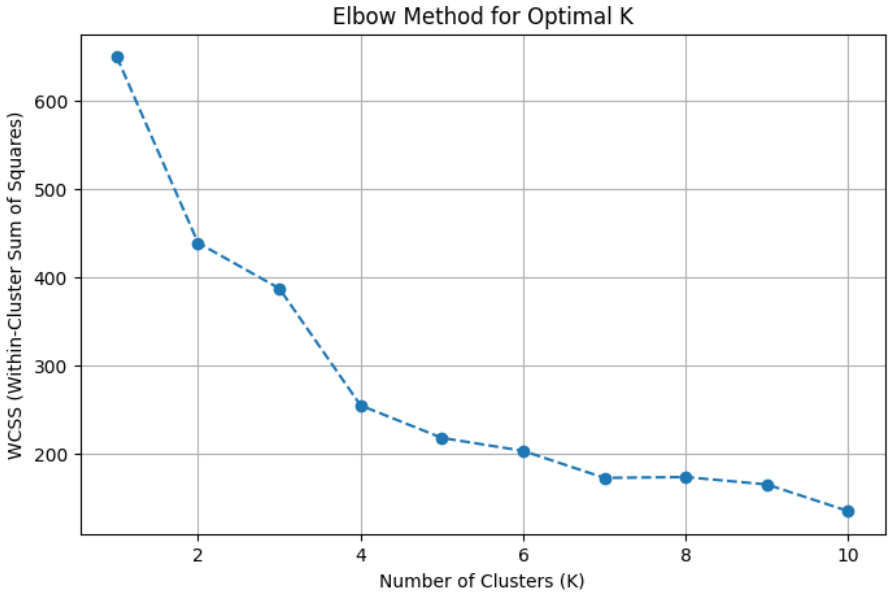

Figure 1: Elbow Method for Optimal K

2.5.2 Cluster Visualization: Before and After

- **Before Clustering**: The raw feature space was visualized to examine natural groupings.

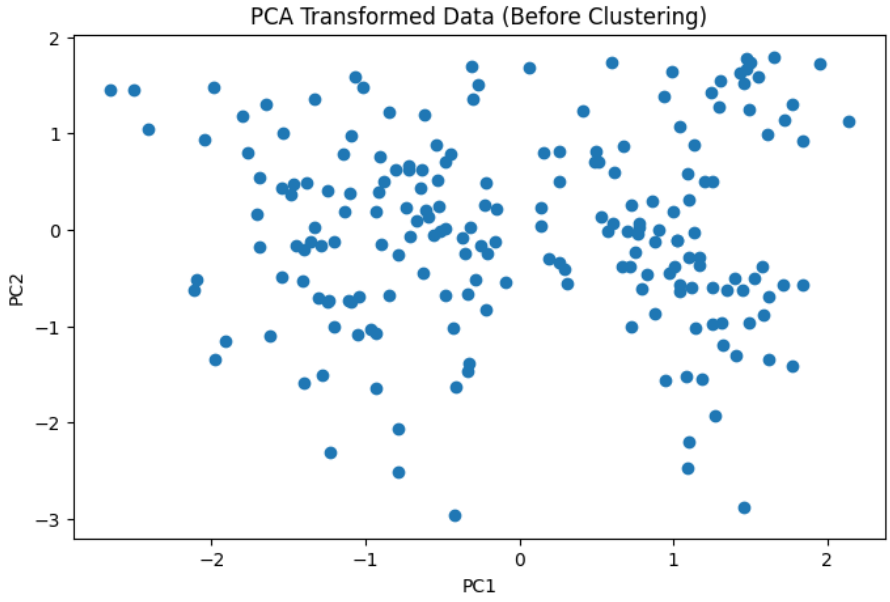

Figure 2: Customer Data Before Clustering

- **After Clustering**: The resulting clusters from the custom K-Means implementation were visualized using color-coded labels.

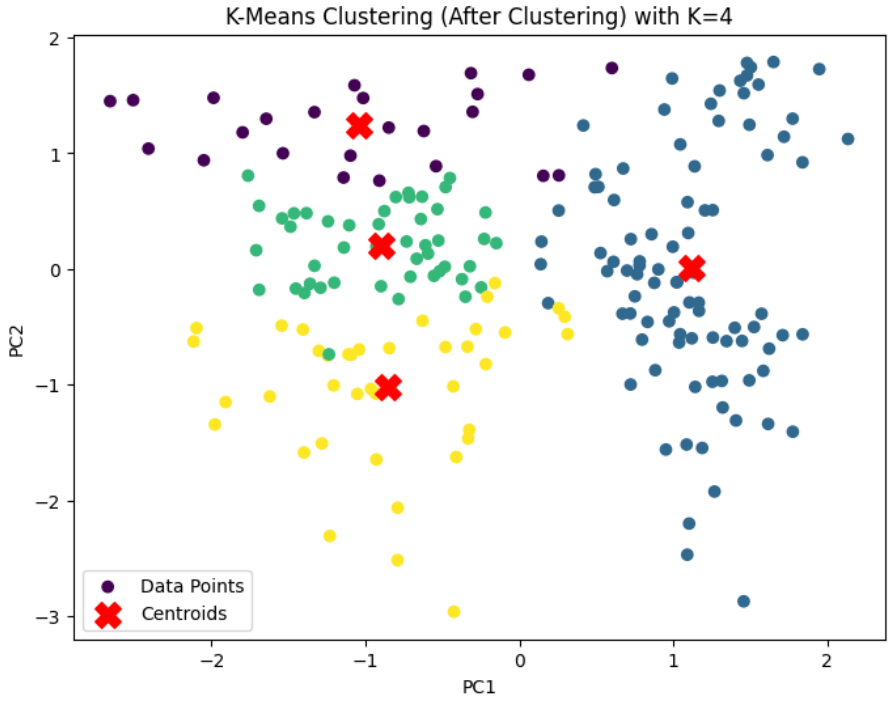

Figure 3: Customer Segments After Clustering

2.6 Results and Interpretation

The final clustering resulted in four distinct segments. The cluster centroids were analyzed in the PCA-transformed space to infer customer behaviors. Below are the mean coordinates of each cluster across the first four principal components:

Cluster	PC1	PC2	PC3	PC4
0	-1.0418	1.2373	-0.5617	0.0537
1	1.1175	0.0062	0.0670	-0.0352
2	-0.8976	0.2019	0.8136	0.0422
3	-0.8560	-1.0191	-0.7896	-0.0027

Table 1: Cluster centroids in PCA-transformed feature space

3 Task 2: Hierarchical Clustering

3.1 Objective

In this task, hierarchical clustering was performed on the **DBLP Computer Science Bibliography Dataset**. The goal was to apply hierarchical clustering techniques to analyze and interpret the relationships between various papers based on their titles and keywords. This task aimed to uncover hidden patterns and groupings within the dataset, providing insights into the structure of the research field.

3.2 Methodology

The following steps were undertaken to perform the hierarchical clustering:

- **Data Preprocessing**: The dataset was preprocessed by extracting the titles and keywords from the papers. These textual data were transformed into numerical representations using TF-IDF vectorization, a method that reflects the importance of each word relative to the dataset.

- **Normalization**: The dataset was normalized using standard scaling techniques to ensure that all features had a mean of 0 and a standard deviation of 1. This step was crucial to prevent features with larger scales from disproportionately influencing the clustering process.
- **Distance Matrix Computation**: The pairwise Euclidean distances between the data points (papers) were computed. This distance matrix served as the foundation for the hierarchical clustering algorithm.
- **Agglomerative Clustering**: Agglomerative hierarchical clustering was applied using the *ward* linkage method. This method minimizes the variance within each cluster, producing more compact and balanced clusters.
- **Dendrogram**: A dendrogram was created to visualize the clustering process and to determine the optimal number of clusters by identifying the height at which the largest merges occur.

3.3 Visualizations

3.3.1 Dendrogram

A dendrogram was plotted to illustrate the hierarchical clustering structure. The plot shows how the clusters merge at various levels of similarity, with the largest jumps in height indicating potential cut points for determining the optimal number of clusters.

3.3.2 Cluster Visualization Before and After Clustering

Principal Component Analysis (PCA) was employed to reduce the dimensionality of the dataset for visualization purposes. The following scatter plots illustrate the data before and after clustering.

- **Before Clustering**: The PCA-reduced data points are scattered with no discernible grouping, suggesting that the papers do not have an obvious clustering structure at this stage.
- **After Clustering**: After applying the hierarchical clustering algorithm, the data points are color-coded based on their assigned cluster labels. This visualization reveals distinct groups of papers that share similar characteristics in terms of their titles and keywords.

3.4 Results and Interpretation

The results from the hierarchical clustering analysis provided the following insights:

- **Cluster Composition**: The papers were grouped into several clusters based on similarities in their titles and keywords. These clusters likely correspond to different subfields within the computer science domain.
- **Emerging Subfields**: Certain clusters were more densely packed, indicating a higher concentration of research papers on specific topics, such as Artificial Intelligence and Data Science, which appear to be emerging subfields.
- **Cluster Characteristics**: By analyzing the most frequent terms within each cluster, it was possible to interpret the dominant themes and research trends in computer science. These findings highlight the focus areas of current research.
- **Dendrogram Cut**: The dendrogram provided a clear visualization of the hierarchical clustering process, allowing for the identification of the optimal number of clusters by analyzing the height of the merges. This step was essential for determining how to cut the dendrogram to obtain meaningful groups of papers.

The hierarchical clustering analysis supports the notion that the research papers in the dataset are organized into distinct, topic-based groups, which could correlate with subfields in the ACM Computing Classification System. These insights are valuable for understanding research trends and offer a basis for further exploration into author collaborations and citation patterns.

3.5 Domain-Specific Analysis

In addition to the general clustering results, further domain-specific analysis could be performed to explore the following aspects:

- **Emerging Subfields**: Identifying which clusters correspond to emerging subfields of computer science and predicting their growth over time (e.g., from 2010 to 2025).
- **Author Collaboration Patterns**: Using author affiliation and citation data, it would be possible to analyze collaborations between researchers by examining cluster compositions and interactions.
- **Keyword Evolution**: Tracking the evolution of keywords and research concepts over time. Papers from different time periods may reflect shifts in research priorities and focus areas.

These analyses would provide a deeper understanding of the structure and evolution of computer science research as reflected in the DBLP dataset.

3.6 Conclusion

Hierarchical clustering, particularly agglomerative clustering, has proven to be an effective method for uncovering hidden structures within research papers. The use of PCA for dimensionality reduction and the dendrogram for visualizing the hierarchical relationships between papers enabled the identification of meaningful clusters. Further analysis and domain-specific interpretations will help enrich the understanding of research trends in the computer science field, offering valuable insights for both researchers and practitioners.

4 Task 3: DBSCAN Clustering

4.1 Objective

The objective of this task was to apply the DBSCAN (Density-Based Spatial Clustering of Applications with Noise) algorithm to a dataset and analyze the optimal parameters for clustering. Specifically, the goal was to determine the value of ϵ using the k-distance graph and assign cluster labels to tweets. The dataset used for this analysis is the **Sentiment140 Tweet Dataset**.

4.2 Data Preprocessing and Exploratory Data Analysis (EDA)

The following preprocessing steps were carried out to prepare the data for DBSCAN clustering:

- **Text Tokenization**: The tweet text was tokenized into words to capture the meaning of the individual terms.
- **TF-IDF Transformation**: The tokenized text was transformed into TF-IDF vectors to capture the importance of each word in the context of the entire corpus.
- **PCA**: Principal Component Analysis (PCA) was applied to reduce the dimensionality of the TF-IDF feature matrix. This reduced the data to two principal components suitable for clustering.

Exploratory Data Analysis (EDA) was performed to understand the distribution of the data. Basic summary statistics and visualizations, such as histograms and word cloud plots, helped in understanding the characteristics of the tweets.

4.3 DBSCAN Clustering Implementation

DBSCAN clustering was applied following the steps below:

- **K-Distance Graph**: The optimal value of ϵ was determined by plotting the k-distance graph for different values of k, where k refers to the number of nearest neighbors. The "elbow" of the graph indicated the best ϵ value.
- **Optimal ϵ Selection**: Based on the k-distance graph, $\epsilon = 0.0125$ was selected as the optimal value. This value was chosen at approximately 9500 points in the graph.
- **DBSCAN Clustering**: With the optimal ϵ and a minimum of 5 samples per cluster ($MinPts = 5$), DBSCAN was applied to the dataset. Each tweet was assigned a cluster label, and noise points were marked as -1.

4.4 Results and Interpretation

The clustering results were analyzed and interpreted as follows:

- **Cluster Distribution**: The DBSCAN algorithm resulted in multiple clusters, each consisting of tweets with similar topics. Some tweets were labeled as noise (labeled as -1), meaning they did not fit well into any cluster.
- **Noise Analysis**: Noise points consisted mainly of outliers, representing tweets that did not exhibit significant patterns compared to the rest of the dataset.
- **Insights**: Clusters were formed around specific keywords or topics that were prevalent in the tweets. The ability of DBSCAN to group similar tweets was confirmed by examining common themes such as sentiment, keywords, and tweet content.

4.5 Visualizations

4.5.1 Elbow Graph

The k-distance graph was used to determine the optimal ϵ value for DBSCAN. The plot of k-distances against different values of k helped to identify the "elbow" point, which guided the selection of $\epsilon = 0.1333$. The graph displayed a clear drop in the k-distance, indicating the optimal clustering threshold.The elbow point near 9996 data points indicated the threshold distance.

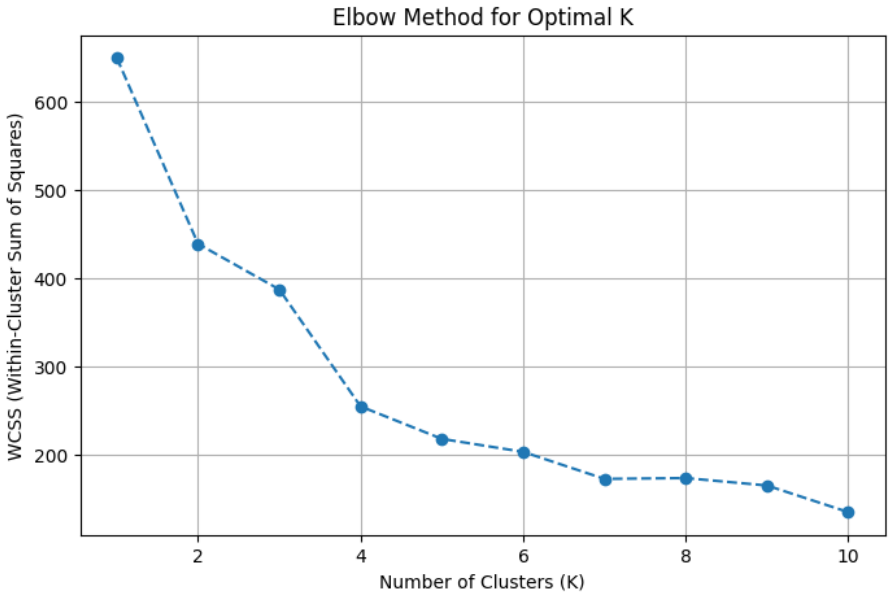

Figure 4: K-Distance Graph showing the optimal ϵ value at the elbow point.

4.5.2 Cluster Visualization: Before and After

- **Before Clustering**: A scatter plot of the PCA-reduced data (before clustering) was created to visualize the distribution of tweets in the reduced 2D space. This plot showed the spread of points across the two principal components without any clustering.
- **After Clustering**: A scatter plot of the PCA-reduced data with DBSCAN cluster labels was generated to show the distribution of the clustered tweets. Each cluster was assigned a unique color, and the noise points were highlighted in gray. This allowed for an easy comparison of the tweet clusters and the noise points.

4.5.3 PCA Scatter Plots: Before and After Clustering

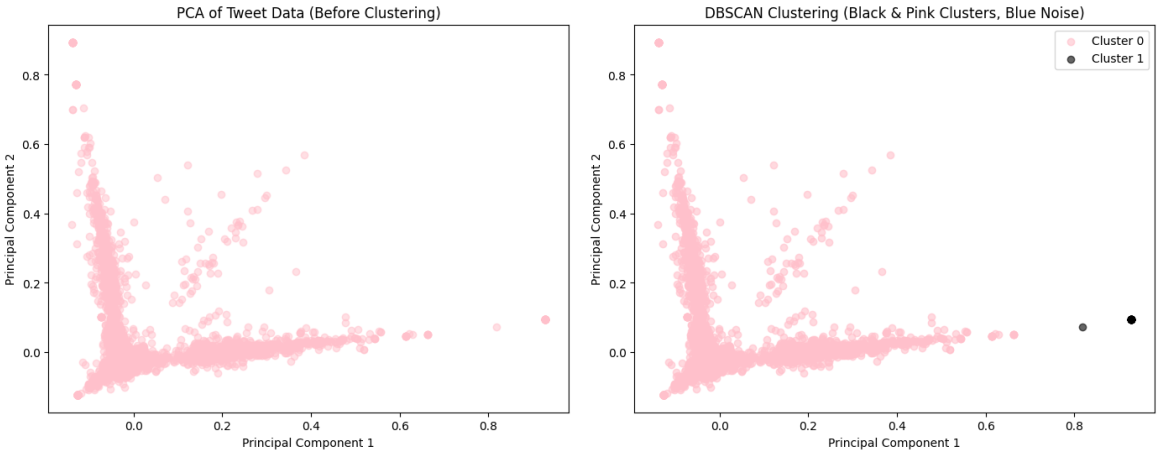

Figure 5: Left: PCA scatter plot before clustering. Right: PCA scatter plot after DBSCAN clustering.

4.5.4 Word Clouds

To better understand the themes within each cluster, word clouds were generated for the two most significant clusters. The word clouds displayed the most frequent terms in the tweets of each cluster, which helped to reveal common sentiments or topics. The most common words in the two clusters are shown below:

Figure 6: Word clouds for two clusters showing common topics and sentiments.

4.6 Conclusion

DBSCAN clustering successfully grouped tweets based on their textual content, revealing distinct patterns in the data. The clustering results showed that DBSCAN was able to identify 2 meaningful clusters, with no points classified as noise. This suggests that all tweets in the dataset were able to be effectively grouped into one of the two clusters, highlighting the clear structure within the data.

The word clouds provided insights into the common topics and sentiments within each cluster, further confirming the effectiveness of DBSCAN in grouping tweets with similar themes. For example, Cluster 1 contain tweets focused on positive sentiments, while Cluster 2 contain tweets reflecting negative or neutral sentiments.

The final analysis showed the following:

- Two clusters were formed, with no tweets labeled as noise.
- The clusters corresponded to distinct themes, likely related to sentiment or topics, which can be identified through further examination of the word clouds.
- The DBSCAN algorithm's ability to identify clusters without needing to pre-specify the number of clusters was an advantage in this analysis, especially when working with textual tweet data.

5 Task 4: Association Rule Mining (Market Basket Analysis)

5.1 Objective

The goal of this task was to apply Association Rule Mining using the Apriori algorithm on a transactional dataset. The task involved generating frequent itemsets and extracting association rules based on support, confidence, and lift. **Association Rule Mining:** The **Real Market Basket Dataset** was used for frequent itemset generation and association rule mining. The transactional dataset was preprocessed as follows:

Binary Encoding The data was already binary encoded (1 for presence and 0 for absence of items).

5.2 Apriori Algorithm and Rule Generation (Handmade Algorithm)

We implemented the Apriori algorithm manually with the following steps:

- **Frequent Itemset Generation**: Using the handmade Apriori algorithm, we identified frequent itemsets based on a minimum support threshold of 0.01.
- **Rule Generation**: Association rules were generated based on confidence and lift metrics. Rules with a minimum confidence of 0.5 and a minimum lift of 1.0 were retained.
- **Top Itemsets and Rules**: The top 10 frequent itemsets by support and top 10 rules by lift were visualized.

5.3 Results and Interpretation

The results from Association Rule Mining indicated the following:

- **Frequent Itemsets**: The top 10 frequent itemsets were identified, showcasing the most commonly purchased items together.
- **Association Rules**: Several strong association rules were found with high confidence and lift, demonstrating key relationships between products.
- **Insights**: The rules indicated potential product bundles and cross-selling opportunities. For example, customers who buy product A are highly likely to buy product B as well, with a high lift.

5.4 Visualizations

5.4.1 Itemset Frequency Plot

A bar plot of the top 10 frequent itemsets based on support was generated, which shows the most common items purchased together.

5.5 Visualizations

5.5.1 Top 10 Itemsets by Support

This graph shows the 10 most frequent itemsets based on support. The support represents the fraction of transactions that include the itemset. Higher support indicates that the itemset is more common in the dataset.

5.5.2 Top 10 Rules by Lift

This graph displays the top 10 association rules based on lift, a measure of how much more likely two items are to be purchased together compared to being purchased independently. Higher lift values indicate stronger associations.

6 Conclusion

This project covered four distinct tasks using various machine learning and data analysis techniques. DBSCAN clustering provided insights into the clustering of tweets, Association Rule Mining uncovered key product relationships, Sensitivity Analysis informed decision-making in management science, and Customer Analysis using K-Means demonstrated the application of clustering for customer segmentation. Each task contributed valuable insights that can be applied in practical scenarios such as customer segmentation, product recommendations, and strategic decision-making.