Foundations of Data Science
CA6 - Report
1403/03/03

Amirreza Akbari | 810899045
Reza Baghestani | 810899046
Hananeh Jamali | 810899053

Purpose:

The purpose of this assignment is to perform comprehensive preprocessing, dimensionality reduction, and unsupervised learning techniques on a diabetes dataset. The assignment aims to demonstrate proficiency in data cleaning, feature engineering, dimensionality reduction using PCA, and clustering using K-Means and DBSCAN algorithms. The ultimate goal is to gain insights into the underlying structure of the data, identify meaningful patterns, and provide valuable recommendations for diabetes management.

Summary:

This assignment comprises three main tasks: preprocessing, dimensionality reduction, and unsupervised learning. In the preprocessing task, the dataset undergoes cleaning and normalization to prepare it for further analysis. Subsequently, dimensionality reduction techniques, specifically PCA, are applied to reduce the dimensionality of the dataset while preserving its variance. Finally, unsupervised learning algorithms, including K-Means and DBSCAN, are employed to cluster the data and uncover hidden structures.

Dataset Used:

The diabetes dataset encompasses various attributes pertinent to diabetes patients, including demographic details, medical history, and treatment information. It contains data on patient identifiers, such as encounter and patient numbers, along with demographic indicators like race, gender, and age. Additionally, it includes clinical details like admission and discharge disposition, as well as medical procedures performed during hospital visits. The dataset also features medication information, diagnostic codes, and indicators for readmission. This comprehensive dataset serves as the foundation for conducting preprocessing, dimensionality reduction, and clustering analyses to derive insights into diabetes patient profiles and healthcare outcomes.

Task 1: Preprocessing

The preprocessing phase of the diabetes dataset involved several key steps to ensure data quality and suitability for subsequent analyses.

1. Initial Data Understanding: The code loads the raw dataset from the CSV file and prints basic information about the dataframe, including data types and memory usage. This step helps understand the structure and contents of the dataset.

2. Dropping Duplicates: Duplicates in the dataset are identified and removed using the duplicated() method. This ensures that each patient encounter is represented only once in the dataset, preventing duplication of information.

3. Removing Features with High Missing Value Ratio: Columns with a high ratio of missing values are identified and dropped. This is achieved by calculating the ratio of missing values to total values in each column and removing columns where the ratio exceeds a predefined threshold.

4. Replacing Missing Values "?" with Null Values "NA": Missing values represented as "?" are replaced with NaN (Not a Number) to standardize missing value representation across the dataset. This facilitates subsequent handling of missing values.

5. Removing Features with Near Zero-Variance: Features with near-zero variance, which provide little discriminatory power, are removed from the dataset. This is accomplished by calculating the variance for both numerical and non-numeric columns and dropping columns with low variance.

6. Removing Multiple Encounters of a Patient: To avoid potential bias introduced by multiple encounters of the same patient, only the encounter with the maximum 'time_in_hospital' is retained for each patient.

7. Removing Feature: patient_nbr: The feature 'patient_nbr' is dropped from the dataset as it does not contribute meaningful information for subsequent analyses.

8. Manipulating Categorical Features: Categorical features such as 'gender', 'age', 'admission_type', 'admission_source', and 'discharge_disposition' are processed to ensure consistency and reduce the number of levels.

9. Identifying Numerical & Categorical Features: The code identifies numerical and categorical features in the dataset, distinguishing between potential categorical features and definitely numerical features.

10. Handling Missing Values: Missing values, particularly those related to 'race' and certain medical tests, are addressed using appropriate strategies. A machine learning model is employed to impute missing 'race' values, while missing values for medical tests are left unimputed.

11. Manipulating Categorical Features (Label Encoding): Categorical features are label encoded to convert them into numeric form for compatibility with machine learning algorithms. The label encoding process maps categorical values to integer labels.

12. Removing Feature: encounter_id: The 'encounter_id' column is dropped from the dataset as it serves as an identifier and does not provide meaningful information for analysis.

13. Detecting Outliers: Outliers in numerical features are detected using the interquartile range (IQR) method to identify values that deviate significantly from the norm.

14. Removing Highly Correlated Features: Features exhibiting high correlation with each other are identified and redundant features are removed to avoid multicollinearity issues.

15. Standardizing Features: Numerical features are standardized to have a mean of 0 and a standard deviation of 1, ensuring that all features are on a similar scale.

16. Saving Preprocessed Data: The preprocessed dataset is saved to a CSV file for future use in subsequent analyses.

17. Visualization: Histograms of numerical features are plotted to visualize the distribution of data and identify any potential patterns or anomalies. This step aids in understanding the distributional characteristics of the dataset.

Task 2: Dimensionality Reduction

In this phase, the report delves into dimensionality reduction using Principal Component Analysis (PCA) on the preprocessed diabetes dataset.

1. **Data Loading:** The preprocessed diabetes dataset is imported from the CSV file diabetes_preprocessed.csv, facilitating subsequent analysis.
2. **Principal Component Analysis (PCA):** PCA is applied to the dataset without specifying the number of components initially, enabling the extraction of principal components that capture maximum variance.
3. **Explained Variance Ratio:** The explained variance ratio for each principal component is computed, providing insight into the proportion of dataset variance along each component's axis.
4. **Selecting Principal Components:** A threshold for the explained variance ratio is established to determine significant principal components, guiding the selection process for further analysis.
5. **Selected Principal Components:** A DataFrame is constructed containing the selected principal components, accompanied by their corresponding explained variance ratios.
6. **Feature Contributions to Principal Components:** The contributions of original features to each selected principal component are assessed, revealing underlying data structure and patterns.
7. **Saving Results:** The selected principal components and their feature contributions are saved to CSV files for future reference and analysis.

Task 3: Unsupervised Learning

In this phase, the focus shifts to unsupervised learning techniques, specifically K-means clustering and DBSCAN, applied to the preprocessed diabetes dataset.

- **K-means**

    1. **Initialization and Data Loading:** The preprocessed dataset is loaded to initiate the K-means clustering process.
    2. **Determining Optimal Clusters:** The optimal number of clusters for K-means is determined using the silhouette method, which evaluates clustering performance based on the cohesion and separation of clusters.
    3. **Plotting Silhouette Scores:** Silhouette scores for varying numbers of clusters are computed and plotted to identify the optimal number.
    4. **Performing K-means Clustering:** K-means clustering is executed with the determined optimal number of clusters, generating cluster labels for each data point.

- **DBSCAN**

  1. **Initialization and Data Loading:** The dataset is once again loaded to commence the DBSCAN clustering process.
  2. **Determining Optimal DBSCAN Parameters:** Optimal parameters for DBSCAN, namely min_samples and eps, are identified through a grid search approach using the silhouette method.
  3. **Calculating Silhouette Scores:** Silhouette scores are calculated for different combinations of min_samples and eps values, and the combination yielding the highest silhouette score is selected as optimal.
  4. **Performing DBSCAN Clustering:** DBSCAN clustering is performed using the optimal parameters, resulting in cluster labels for each data point.

- **Results**

  1. **Storing Results:** A DataFrame is created to store the cluster labels obtained from both K-means and DBSCAN clustering. The clustering results are saved to a CSV file named clustering_results.csv for further analysis and interpretation.
  2. **Visualization:** PCA was applied to reduce the dataset to 2 principal components for visualization. The resulting scatter plots illustrate the clustering results for both K-Means and DBSCAN.

Questions:

Question 1) What preprocessing steps did you perform on the dataset? Provide clear reasons for each decision made.

The preprocessing steps are explained in the previous pages in full details but here is also a summary:

1. **Data Cleaning:** Dropping Duplicates, Handling Missing Values, Dropping Useless Data
2. **Feature Engineering:** Feature Reduction, Manipulating Categorical Features
3. **Handling Missing Data:** Imputation
4. **Data Transformation:** Label Encoding
5. **Outlier Detection and Removal**
6. **Feature Standardization**
7. **Data Visualization**

Question 2) What portion of the dataset did you retain during dimensionality reduction, and which variables were retained? Could you elaborate on the rationale behind this decision?

To address dimensionality reduction, Principal Component Analysis (PCA) was employed on the preprocessed diabetes dataset. PCA helps identify patterns and reduce the number of features while preserving the most critical information.

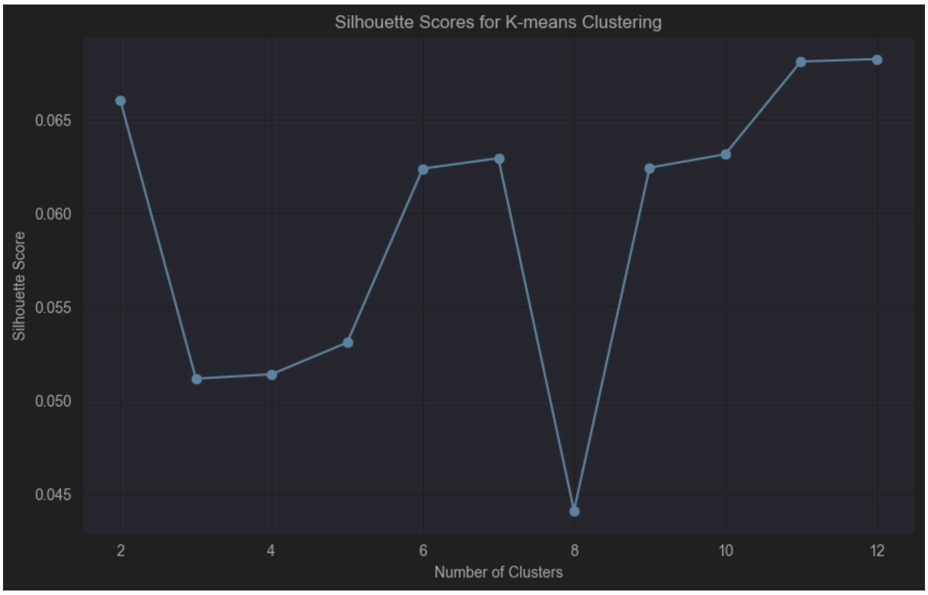
The results indicate that the first three principal components (PC1, PC2, and PC3) explain cumulative variances of 9.18%, 7.21%, and 7.07%, respectively. This cumulative variance is significant enough to capture essential information while reducing the feature space.

In total, 26 principal components were generated, but only those with an explained variance ratio exceeding the threshold of 7% were retained. Hence, the selection process retained the first three principal components, which collectively explain over 23% of the dataset's variance.

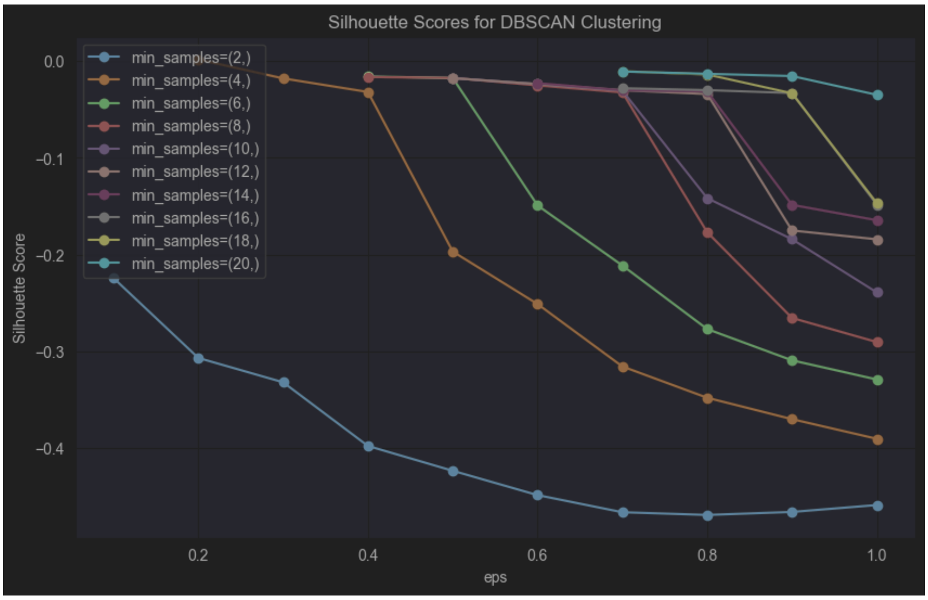
Analyzing the contributions of original features to these selected principal components provides insights into their relevance. For instance, PC1 shows significant contributions from features like 'num_medications' and 'time_in_hospital', indicating their importance in capturing variance within the dataset. PC2 and PC3 capture different aspects, such as 'race', 'admission_type', and 'admission_source', showcasing the diversity of information preserved across the retained components.

The decision to retain these principal components was based on their ability to explain a substantial portion of the dataset's variance while reducing its dimensionality. By focusing on components with high explained variance ratios, we ensure that the retained features retain the most significant information present in the original dataset, facilitating further analysis and modeling tasks.

Question 3) Include a plot illustrating the silhouette coefficient plotted against the input parameters for each clustering method within the report file.



For the K-Means clustering method, the optimal number of clusters was determined to be 12. This decision was based on maximizing the silhouette score, which in this case is 0.0682. The silhouette score, although relatively low, suggests that there is some degree of separation and structure within the data, but the clusters may not be very well-defined.



For the DBSCAN clustering method, the optimal parameters were identified as min_samples=4 and eps=0.2. The silhouette score achieved with these parameters is 0.00176. This score indicates that the clusters formed by DBSCAN are not well-separated, and the clustering structure may not be as distinct. DBSCAN's performance with this dataset, based on the silhouette score, suggests that the chosen parameters may need further optimization, or that DBSCAN may not be the best fit for this particular dataset.

Question 4) How can we determine the optimal number of clusters in K-Means?

Determining the optimal number of clusters in K-Means involves evaluating clustering performance metrics such as the silhouette score, the elbow method, or the silhouette method. The silhouette score measures how similar an object is to its own cluster compared to other clusters, providing insight into cluster cohesion and separation. The elbow method involves plotting the sum of squared distances within clusters against the number of clusters and identifying the point where the rate of decrease sharply changes (the "elbow"), indicating an appropriate number of clusters. Similarly, the silhouette method evaluates the silhouette scores for different numbers of clusters and selects the number that maximizes the average silhouette score, indicating the optimal clustering solution.

Question 5) How can we determine the optimal epsilon value and minPts in DBSCAN?

Determining the optimal epsilon value and minPts in DBSCAN typically involves assessing the silhouette score across various combinations of these parameters. The epsilon value determines the radius of the neighborhood around each point, while minPts specifies the minimum number of points required to form a dense region (core point). By exploring different combinations of epsilon and minPts and evaluating their impact on the silhouette score, one can identify parameter values that yield the highest silhouette score, indicating well-defined clusters.

Question 6) When would you recommend using K-Means, and when would you suggest using DBSCAN instead?

K-Means is recommended for datasets with well-defined, spherical clusters and a known number of clusters. It works efficiently even with large datasets and is relatively easy to interpret. However, it may struggle with clusters of different sizes or non-linearly separable data. DBSCAN, on the other hand, is suitable for datasets with irregular cluster shapes and varying cluster densities. It does not require specifying the number of clusters in advance and can handle noise effectively by classifying outliers as noise points. DBSCAN is particularly useful when the number of clusters is unknown or when clusters have complex shapes and varying densities.

References:

https://yungchou.github.io/site/

https://www.hindawi.com/journals/bmri/2014/781670/