# 111-2 DBMS Final Project: ECSQL

Access databases with one button

LIN, BO-YONG
NTU BEBI
First Grade

KUO, TING-YI
NTU BEBI
First Grade

TSENG, YU-HSUAN
NTU BEBI
First Grade

ZHANG, YU-JIE
NTU BEBI
First Grade

## ABSTRACT

The aim of this process is to offer users a seamless and efficient method for managing and analyzing their data. By utilizing Principal Component Analysis (PCA) and k-Nearest Neighbors (kNN) algorithms, the input data undergoes a transformation, resulting in a lower-dimensional representation that preserves the crucial characteristics of the data. Following this, the kNN algorithm is employed to classify data instances and assign clustering indices, enabling the identification of data groups with similar characteristics.

This process empowers users to gain valuable insights from their data by providing a streamlined approach to data handling and analysis. Through the application of PCA, the input data is condensed into a concise representation that captures the most significant aspects. Subsequently, the kNN algorithm is employed to categorize the data instances and assign clustering indices, simplifying the identification of data clusters with shared characteristics.

By adopting this approach, users can effectively analyze their data, make informed decisions, and uncover meaningful patterns. The utilization of PCA and kNN algorithms enhances the efficiency and accuracy of data analysis, enabling users to extract valuable knowledge from their datasets.

In summary, this process offers users an optimized methodology for data management and analysis. Leveraging PCA and kNN algorithms enables efficient data transformation, classification, and clustering, ultimately facilitating the identification of similar data groups and enhancing the understanding of the underlying patterns in the dataset.

## CCS CONCEPTS

•Information systems~Data management systems~Data structures~Data access methods

## KEYWORDS

Database, User Interface, Data Analysis, Principle Components, k-Nearest Neighbor, Clustering, Non-cluster Index

## 2 PRIOR WORK

### 2.1 EXCEL

Managing large datasets in Excel has long been a significant challenge for users, prompting the exploration of various approaches to improve efficiency. One common strategy is to split the data into multiple worksheets or workbooks, allowing for better organization and easier manipulation. Another technique involves utilizing pivot tables and filtering options, enabling users to analyze and summarize large datasets more effectively. Additionally, the use of extensions and add-ins has become popular, providing users with additional functionalities and tools to enhance data management and analysis capabilities.

While these prior works have made significant contributions in addressing the challenges of working with large datasets in Excel, there is ongoing research in this field to further enhance data management capabilities, improve performance, and provide more seamless integration with databases.

### 2.2 PCA

PCA is widely used and well-established, previous work has also recognized certain limitations. For example, the interpretability of the resulting principal components and their relationship to the original features has been a region of interest. People have tried to develop techniques to better understand and interpret the meaning and contribution of individual features in reduced dimensional space.

The state of the art has recognized the need to explore alternative dimensionality reduction techniques beyond PCA. Many research teams have investigated methods such as t-SNE, LLE, ISOMAP, etc. to address specific challenges.

### 2.3 KNN

The k-Nearest Neighbor (kNN) algorithm is a widely used supervised learning algorithm in machine learning for classification and regression tasks. It operates by identifying the k closest labeled training examples to an unlabeled data point and predicting its class based on the majority class of its neighbors. This algorithm finds its application in datasets where data naturally forms distinct clusters, enabling accurate classification of new input even in the absence of prior knowledge.

Although kNN is effective, it has certain limitations. One of the key challenges lies in computing accurate distances between data points, which directly affects the algorithm's accuracy. Additionally, determining the optimal value of k, the number of neighbors to consider, can be non-trivial. The choice of distance

metric and feature selection also impact the algorithm's performance, posing further challenges in its implementation.

## 2.4 Clustering Index & Non-cluster Index

In database systems, clustering index and non-cluster index are two different types of indexes used to optimize data retrieval operations.

1. Clustering Index:

   A clustering index in a database determines the physical order of data rows in a table based on one or more columns, usually the primary key. It organizes the data on disk to match the order of the index, resulting in improved performance for queries accessing consecutive rows or performing range-based searches. This indexing approach allows for efficient retrieval of related data. It's important to note that each table can have only one clustering index.

2. Non-cluster Index:

   A non-cluster index, also called a secondary index, is a data structure that provides a separate lookup mechanism for data in a table. Unlike a clustering index, it doesn't dictate the physical order of the data on disk. Instead, it maps the indexed column(s) to the corresponding rows in the table using a separate structure.

   Non-cluster indexes are useful for optimizing queries that involve searching for specific values or performing equality checks on the indexed column(s). When a query references the indexed column(s), the non-cluster index allows the database to locate the relevant rows more efficiently, reducing the need for a full table scan.

   It's important to note that a table can have multiple non-cluster indexes, each targeting different columns or combinations of columns. This allows for efficient access to data based on different search criteria.

# 3 SOLUTION

## 3.1 EXCEL

We propose a user-friendly solution to facilitate non-programmers in accessing data from databases through a user interface designed in Excel. The solution involves the following functions:

1. Database Connection: Users input the relevant information of the server and establish a connection to the desired database.

2. Download and Upload Data: Users can easily download data from or upload data to a specific table within the chosen database. The downloaded data is automatically saved in a worksheet named after the table.

3. Create New Tables: To prevent users from uploading data that does not exist in the database, we have implemented a feature that allows users to input the information of the table they want to create. This includes specifying the table name, attributes, and attribute types.

By providing an intuitive user interface within Excel, our solution empowers non-programmers to interact with databases efficiently. Users can seamlessly retrieve and manipulate data, as well as create new tables with the assurance of data integrity. This approach bridges the gap between non-programmers and databases, enabling easier access to valuable data resources.

## 3.2 PCA

We used Principal Component Analysis (PCA), a statistical method widely used for dimensionality reduction in data analysis, to address the challenges associated with high-dimensional data, which often bring up several difficulties, including feature correlation, computational cost, and overfitting problems.

Feature correlation is a common problem in high-dimensional data, where certain features have strong correlations with each other, leading to problems such as multiple solution ambiguity and redundancy. To solve this problem, PCA examines multi-variable data, identifies correlations between these variables, and determines the optimal combination of values that effectively captures the differences in the results. By using these combined feature values, PCA facilitates the construction of a more concise feature space.

In addition, computational costs can be a significant issue when dealing with large sample sizes and a high number of features. Too many features require more storage, which reduces operational efficiency. PCA addresses this challenge by reducing the dimensionality of the data, allowing for more efficient storage and processing.

High-dimensional data can make training models more prone to overfitting problems. With many complex features, models tend to capture too much noise and detail, blocking their ability to generalize to new data.

So, we applied initially consisted of 13 dimensions through PCA, and are able to retain three principal components that captured the maximum variability in the original data. The concept of explained variance ratio was used to measure the contribution of each principal component to the total variability of the original data. This will allow us to better understand its underlying structure and facilitates subsequent analysis and interpretation.

## 3.3 KNN

We implement the k-nearest neighbors (kNN) classification algorithm to classify a dataset that has been transformed using principal component analysis (PCA).

The kNN classification algorithm consists of several steps:

1. Data Splitting: The dataset is split into a training set and a test set.

2. Learning Step: The training data is used to construct a kNN classifier.

3. Hyperparameter Tuning: Grid search and cross-validation are performed on the kNN classifier to find the best model and hyperparameters.
4. Prediction and Accuracy Calculation: The best model is used to predict the classes of the training set and test set. The accuracy of the predictions is calculated.
5. DataFrame Construction: The principal component features, corresponding class labels, and predicted results of the training set and test set are combined to create DataFrames.
6. Optional Visualization: Optional visualization steps can be performed, such as scatter plots for classification results or heatmaps for confusion matrices.

For each new unlabeled data point, the kNN algorithm performs the following operations:

1. Calculate the Euclidean distance between the test sample and each specified training sample.
2. Find the k nearest neighbors based on the calculated distances.
3. Assigns the class that contains the maximum number of nearest neighbors to the new data point, thereby determining its classification.

The input variable 'k' determines the number of neighbors to consider.

In summary, we implement the k-nearest neighbors (kNN) classifier for classifying a dataset that has been transformed using principal component analysis (PCA). After reducing the dimensionality of the original dataset, we then use kNN to predict and classify the transformed data. The result of classifying the PCA-transformed dataset is the assignment of each data point to a specific class based on the classes of its closest neighbors.

# 4 RESULT

## 4.1 EXCEL

In our study, we developed a user interface that facilitates easy data input and management. The user interface, depicted in Figure 1 and 2, features a left column where users can input relevant information of the server. A dropdown menu displays all available databases and tables, allowing users to select a specific database and table for data upload. Once a database and table are chosen, the corresponding attributes are shown under the button. Users can then record data based on the displayed attributes.

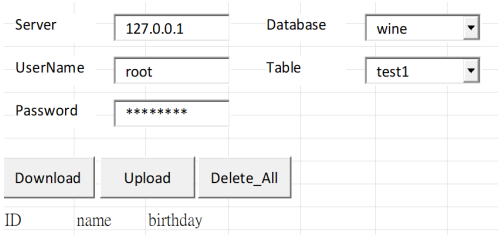


Figure 1: Database connection / Download and upload data

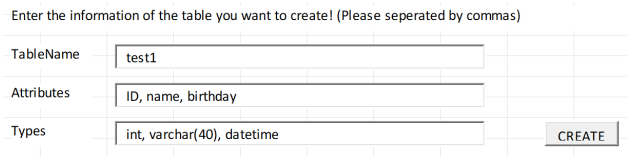


Figure 2: Create new tables

Upon pressing the Upload button, the data undergoes a two-step processing procedure involving PCA and the kNN algorithm. PCA is applied to reduce the dimensionality of the data and extract meaningful features. Subsequently, the reduced-dimensional data is classified using the kNN method. The kNN algorithm assigns each data point to a specific class based on its proximity to neighboring data points in the feature space.

Following the classification step, the data is stored in the database. We leverage the classification results to organize the data into different tables based on their assigned classes. This approach enables faster retrieval of relevant data by querying specific tables based on the desired classification. By employing PCA and kNN in the data processing pipeline, we enhance the efficiency and accuracy of data storage and retrieval within the database.

## 4.2 PCA

Analysis of Table 1 shows that a significant proportion of the variability in the data (99.81%) can be effectively captured by the first principal component alone. The second and third principal components, although less easy to interpret, show variations that are specific to different datasets, making them valuable and reserved for future use.

In addition, scatterplot matrices serve as intuitive visualization tools, allowing the observation of correlations between variables, scatter patterns, and potential trends. Scatterplot matrix analysis further supports the effectiveness of principal components. In particular, combinations that include the first principal component tend to provide better discrimination between the three different data labels when presented in a two-dimensional plot. However, combinations that do not include the first principal component fail to achieve a clear resolution of the different data labels.

These results emphasize the importance of the first principal component in capturing the most significant variability within the data. Using this principal component in combination appropriately, can enhance the distinction between data labels and facilitates a comprehensive understanding of the underlying structure of the dataset.

**Table 1. The explained variance ratio is the percentage of variance that is attributed by each of the selected components**

| Principal Components | PC1 | PC2 | PC3 |
|---|---|---|---|
| Explained Variance Ratio (%) | 99.81 | 0.17 | 0.01 |

### 4.3 KNN

The kNN classifier was trained and tested on the dataset. The best hyperparameters, determined through grid search and cross-validation, resulted in a k value of 5 for the number of neighbors. The training accuracy of the kNN model was 0.81, indicating a high level of accuracy in predicting the classes of the training set. The testing accuracy of the model was 0.67, showing a moderate level of accuracy performance on unlabeled data.

The scatterplot matrix plot provides a visual representation of the predicted and actual class distributions in the testing set. This plot helps to understand how well the predictions align with the ground truth labels.

The heatmap, representing the confusion matrix, visualizes the classification results of the kNN model on the testing set. Each cell in the heatmap shows the count or percentage of instances that were predicted as a particular class ('Predicted') while comparing them to the actual class labels ('Actual'). The heatmap enables the assessment of how well the predicted classes match the true classes, highlighting any misclassifications or patterns in the predictions.

**Table 2. Performance of kNN Classifier on Training and Testing Data**

| | Training Accuracy | Testing Accuracy |
|---|---|---|
| kNN | 0.81 | 0.67 |

## 5 CRITIQUE

### 5.1 EXCEL

One notable strength of our approach is the seamless integration of Excel with a database through our user interface. By connecting Excel to a database, we provide users with the familiar Excel environment and its extensive range of functionalities, while also enabling them to interact with and manipulate database data. This integration eliminates the need for users to switch between different software applications or learn new tools, making it convenient and efficient for users who are already proficient in Excel.

However, it is important to acknowledge that our approach has certain limitations. One of the main limitations is the restricted scope of database manipulation offered by our user interface. Although it provides basic functionality for data input, retrieval, and storage, it may not fulfill the requirements of users who need to perform more advanced operations commonly found in dedicated database management systems. Complex queries, custom data structures, and advanced data processing algorithms are beyond the capabilities of our interface, which could limit the flexibility and sophistication of data manipulation.

### 5.2 PCA

We provided a clear and concise explanation of PCA and its importance in reducing the dimensionality of data while retaining maximum variability. We effectively highlighted the specific challenges of feature correlation, computational time cost, and overfitting problems in high-dimensional datasets, providing a strong foundation for using PCA as a solution.

We also communicated our decision to retain three principal components from the original 13 dimensions of the data was adequately justified, as well as used the concept of explained variance ratio to measure the contribution of each principal component to the overall variability of the data. By incorporating these details, we demonstrated a robust approach to dimensionality reduction and the selection of informative components.

### 5.3 KNN

We successfully implemented the k-nearest neighbors (kNN) classification algorithm on a dataset transformed using principal component analysis (PCA). The algorithm demonstrated the ability to classify data points based on their nearest neighbors. With a training accuracy of 0.81, the kNN model achieves a high level of accuracy in predicting the classes of the training set. However, the model's performance on unlabeled data is moderate, as indicated by a testing accuracy of 0.67.

Despite achieving moderate accuracy on the testing data, there is room for improvement in the kNN classifier's performance. The accuracy level of 0.67 indicates that the model's predictions are correct for approximately two-thirds of the unlabeled data instances. It suggests that further exploration and refinement of the algorithm, such as exploring alternative distance metrics or feature selection techniques, could potentially enhance its accuracy. Additionally, considering other classification algorithms and comparing their performance could provide valuable insights for future improvements.

Overall, our implementation successfully applied the kNN classification algorithm to a PCA-transformed dataset, providing valuable classification results and highlighting areas for future enhancement.

## 6 POSSIBLE EXTENSION

Some potential extensions that can be explored based on the findings and methodology presented in this project are:

1. In order to provide users with more advanced functionality and enhance their experience with the database, possible extensions to our user interface include integrating additional features such as advanced querying capabilities, data visualization tools, and data mining algorithms. By incorporating these extensions, users will be able to perform complex data analysis, generate insightful visualizations, and uncover valuable patterns and trends within their data.
2. Feature importance and interpretability: While PCA effectively reduces the dimensionality of the data, the

resulting principal components may lack direct interpretability. To address this, future research could focus on exploring techniques to assess the importance of individual features within each principal component and their contribution to overall variability. Methods such as feature loading analysis or correlation analysis between the original features and the principal components could be used to gain insight into the significance and interpretability of the features in the reduced dimensional space.

3. Application to different datasets: Extending analysis to different datasets and domains will provide a broader perspective on the applicability and validity of classification. Examining how our tasks can be performed on different datasets with different characteristics and complexities will highlight their strengths and limitations in different contexts, such as healthcare, finance or image analysis, and evaluate their comprehension performance in terms of information retention and subsequent classification.

4. Explore different distance metrics in the k-nearest neighbors (kNN) algorithm. While the Euclidean distance is commonly used, other distance measures such as Manhattan distance, Minkowski distance, or cosine similarity may provide better performance for specific datasets. By experimenting with different distance metrics and evaluating their impact on the classification accuracy, we can potentially improve the overall performance of the kNN classifier.

5. Handle imbalanced datasets can be valuable. Imbalanced datasets, where the number of instances in different classes is significantly skewed, can lead to biased models. Techniques such as oversampling minority classes, undersampling majority classes, or using advanced algorithms like SMOTE (Synthetic Minority Over-sampling Technique) can help address the class imbalance issue and improve the kNN classifier's performance.

## ACKNOWLEDGMENTS

We would like to thank our teammates and professors who helped make this work possible. We are grateful for the guidance of the professor of this course who provided valuable feedback and advice in the early stages of selecting a topic. We would also like to thank our teammates who were always willing to collaborate and provide their insights and work together on the final report.

## REFERENCES

[1] Microsoft. (2022). Excel performance - tips for optimizing performance obstructions. Excel performance - Tips for optimizing performance obstructions | Microsoft Learn. https://learn.microsoft.com/en-us/office/vba/excel/concepts/excel-performance/excel-tips-for-optimizing-performance-obstructions.

[2] Zoomer Analytics. (2023a). API Reference - xlwings Documentation. https://docs.xlwings.org/zh_TW/latest/api/index.html

## WORK ASSIGNMENT TABLE

| Member | Work |
|---|---|
| LIN, BO-YONG | Organize meetings and project information, implement PCA method, map workflows |
| KUO, TING-YI | Establish connection between Excel and database, design an intuitive user interface, introduction |
| TSENG, YU-HSUAN | Create table after analysis, the search query function |
| ZHANG, YU-JIE | Conducting k-nearest neighbors (kNN) analysis |