**Project-Based Learning Report**

on

**Implement Data reduction techniques - Data Cube Aggregation and**

**Dimensionality Reduction (PCA) with real-time database in Python**

Submitted in the partial fulfillment of the requirements.

For the Project-based learning **Artificial Intelligence & Data Mining**

in

Electronics & Communication Engineering

By

|  |  |
| --- | --- |
| **2014111105** | **Anshika Sharma** |
| **2014111037** | **Prakriti Deep** |
| **2014111087** | **Shubham Pandey** |

Under the guidance of Course In-charge

**Prof. Dr. Tanuja Dhope**

Bharati Vidyapeeth

(Deemed to be University)

College of Engineering,

Pune – 411043

Academic Year: 2023-24

**Bharati Vidyapeeth**

**(Deemed to be University)**

**College of Engineering,**

**Pune – 411043**

**CERTIFICATE**

This is to be Certified that the Project Based Learning report entitled, **“Implement Data Reduction Techniques - Data Cube Aggregation and Dimensionality Reduction (PCA) with real-time database in Python”** work is done by

|  |  |
| --- | --- |
| **2014111105** | **Anshika Sharma** |
| **2014111037** | **Prakriti Deep** |
| **2014111087** | **Shubham Pandey** |

In partial fulfillment of the requirements for the award of credits for Project Based Learning (PBL) in **Artificial Intelligence & Data Mining** Bachelor of Technology Semester-VII, Electronics and Communication Engineering.

**Date: 27th October 2023**

**Prof. Dr. Tanuja S. Dhope Dr. Arundhati A. Shinde**

**Course In charge Professor & Head**

**Electronics and Communication**

**Engineering**

Index

|  |  |  |
| --- | --- | --- |
| **Sr. No.** | **Name of the topic** | **Page No.** |
| 1. | Problem Statement & Solution | 4 |
| 2. | Introduction | 5-6 |
| 3. | Python | 7-8 |
| 4. | Result & Analysis | 9-11 |
| 5. | Outcome & Conclusion | 12 |

**Chapter -1**

**Problem Statement:**

Implement Data reduction techniques - Data Cube Aggregation and Dimensionality Reduction (PCA) with real time data base in Python.

**Solution: -**

The solution involves integrating Data Cube Aggregation and Principal Component Analysis (PCA) in Python with a real-time database. For Data Cube Aggregation, Pandas and NumPy can be utilized for efficient data manipulation and summarization. For PCA, libraries like scikit-learn and NumPy can be employed to reduce data dimensions while preserving variance. A robust pipeline using SQLAlchemy for database interaction would facilitate real-time data processing, application of reduction techniques, and seamless storage back into the database. Thorough analysis of data structures and project requirements, along with error handling and scalability considerations, is essential for successful implementation.

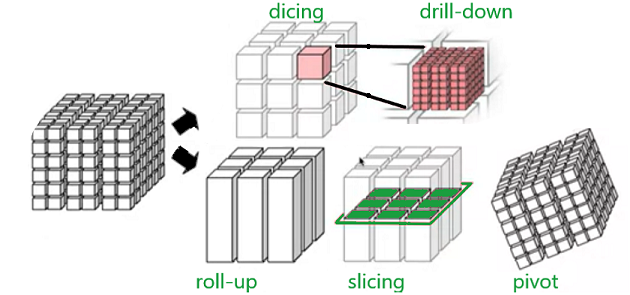


Fig-1: Data Cube Aggregation

**Chapter - 2**

**Introduction:**

Implementing data reduction techniques like Data Cube Aggregation and Dimensionality Reduction (PCA) with a real-time database in Python can significantly enhance the efficiency of data processing and analysis. Let's delve into each technique's description and their applications:

Data Cube Aggregation:

Data Cube Aggregation is a pivotal technique used for handling large datasets efficiently. It involves the process of transforming data from a detailed level to a higher, summarized level, often used in OLAP (Online Analytical Processing). This technique is crucial in data warehousing and helps in analyzing multidimensional datasets. By pre-aggregating data, it speeds up query response times and enables faster decision-making. In Python, libraries like Pandas and NumPy can be employed for data manipulation and aggregation.

Dimensionality Reduction using Principal Component Analysis (PCA):

Dimensionality Reduction, specifically Principal Component Analysis (PCA), is a widely used statistical technique for reducing the dimensions of a dataset while preserving its variance. It helps in simplifying the complexity of high-dimensional data, enabling easier visualization and analysis. By transforming a large set of variables into a smaller set that still contains most of the information, PCA aids in data compression and feature extraction. Python offers various libraries for PCA implementation, including sci-kit-learn and NumPy.

Integrating these techniques with a real-time database in Python will necessitate utilizing appropriate database connectors like SQLAlchemy for database interaction and retrieval. It's essential to establish a robust pipeline that can process incoming data in real-time, apply data reduction techniques, and store the processed data efficiently back into the database. Proper error handling and scalability considerations should also be considered for real-time data processing to ensure the system's reliability and stability.

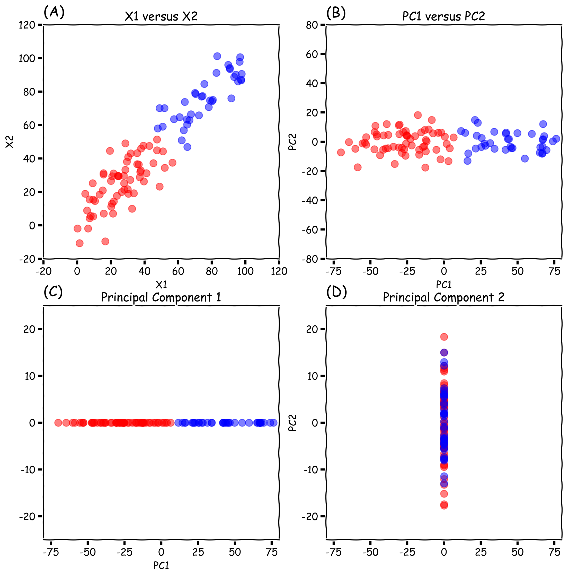


Fig-2: Dimensionality Reduction using PCA

For an effective implementation, it's crucial to conduct a comprehensive analysis of the specific data structures and requirements of the project, considering the scale of the data, the frequency of updates, and the specific goals of data reduction. Leveraging Python's powerful libraries, combined with a thorough understanding of the data and its context, will be key to successfully integrating these techniques into your project.

**Chapter- 3**

**Software Used**

PyCharm is a cross-platform integrated development environment (IDE) for Python programming. It is developed by JetBrains, and is available in two editions: Community and Professional. The Community edition is free and open-source, while the ****Professional edition is commercial and includes additional features such as support for web development, data science, and remote development.

PyCharm provides a variety of features to help Python developers write code more efficiently and accurately. These features include:

* Code completion: PyCharm provides code completion suggestions for Python code, based on the context of the code being written.
* Syntax highlighting: PyCharm highlights Python syntax in different colors, making it easier to read and understand code.
* Code inspection: PyCharm performs code inspection on Python code, identifying potential errors and problems.
* Debugging: PyCharm provides a debugger for Python code, allowing developers to step through code line by line and inspect the values of variables.
* Refactoring: PyCharm provides refactoring tools for Python code, allowing developers to safely make changes to the structure of their code.

PyCharm also provides a number of other features to help Python developers, such as:

* Support for multiple programming languages: PyCharm supports multiple programming languages, including Python, JavaScript, CoffeeScript, TypeScript, Cython, SQL, HTML/CSS, and template languages.
* Integration with version control systems: PyCharm integrates with version control systems such as Git and Mercurial, allowing developers to manage their code changes and collaborate with other developers.
* Remote development: PyCharm supports remote development, allowing developers to work on code that is hosted on a remote server.

**Language Used:**

****Python is a versatile, high-level, and interpreted programming language known for its simplicity and readability. Created by Guido van Rossum and first released in 1991, Python has become one of the most popular programming languages, with a vast and active community of developers worldwide. Here are some key aspects of Python:

1. Readability: Python emphasizes code readability and simplicity, employing a clear and intuitive syntax that reduces the cost of program maintenance.
2. Versatility: Python supports multiple programming paradigms, including object-oriented, imperative, and functional programming, making it suitable for various applications and tasks.
3. Vast Standard Library: Python provides a comprehensive standard library that offers support for tasks such as web services, string operations, file I/O, and more, reducing the need for external libraries for many common programming tasks.
4. Interpreted Nature: Python code is executed line by line by the Python interpreter, facilitating rapid development and debugging. This also makes it a suitable language for scripting and prototyping.
5. Dynamically Typed: Python is dynamically typed, meaning variables do not require explicit declaration. This feature simplifies coding but may require careful attention to variable types during development.
6. Community and Ecosystem: Python boasts a robust community and ecosystem, with a multitude of libraries and frameworks for various applications, such as web development (Django, Flask), scientific computing (NumPy, Pandas), machine learning (TensorFlow, PyTorch), and more.

**Chapter – 4**

**Result & Analysis**

**Data Cube Aggregation:**

import pandas as pd

data = {

'Product': ['A', 'A', 'B', 'B', 'A', 'B', 'A', 'B'],

'Region': ['North', 'South', 'North', 'South', 'North', 'South', 'North', 'South'],

'Sales': [100, 150, 200, 250, 300, 350, 400, 450]

}

# Creating a DataFrame

df = pd.DataFrame(data

# Performing data cube aggregation

cube = df.groupby(['Product', 'Region']).agg({'Sales': ['sum', 'mean', 'max', 'min', 'count']})

# Printing the cube

print("Data Cube Aggregation:")

print(cube)

**A screenshot of a computer

Description automatically generated**

Fig-3: Output of Data Cube Aggregation

**Dimensionality Reduction using PCA:**

import pandas as pd

import numpy as np

import matplotlib.pyplot as plt

from sklearn.datasets import load\_iris

from sklearn.preprocessing import StandardScaler

from sklearn.decomposition import PCA

# Load the Iris dataset

iris = load\_iris()

X = iris.data

y = iris.target

# Standardize the data

scaler = StandardScaler()

X\_scaled = scaler.fit\_transform(X)

# Apply PCA

pca = PCA(n\_components=2)

X\_pca = pca.fit\_transform(X\_scaled)

# Plotting the results

plt.figure(figsize=(8, 6))

for i, target\_name in zip([0, 1, 2], iris.target\_names):

plt.scatter(X\_pca[y == i, 0], X\_pca[y == i, 1], label=target\_name)

plt.xlabel('Principal Component 1')

plt.ylabel('Principal Component 2')

plt.title('PCA on Iris Dataset')

plt.legend()

plt.show()

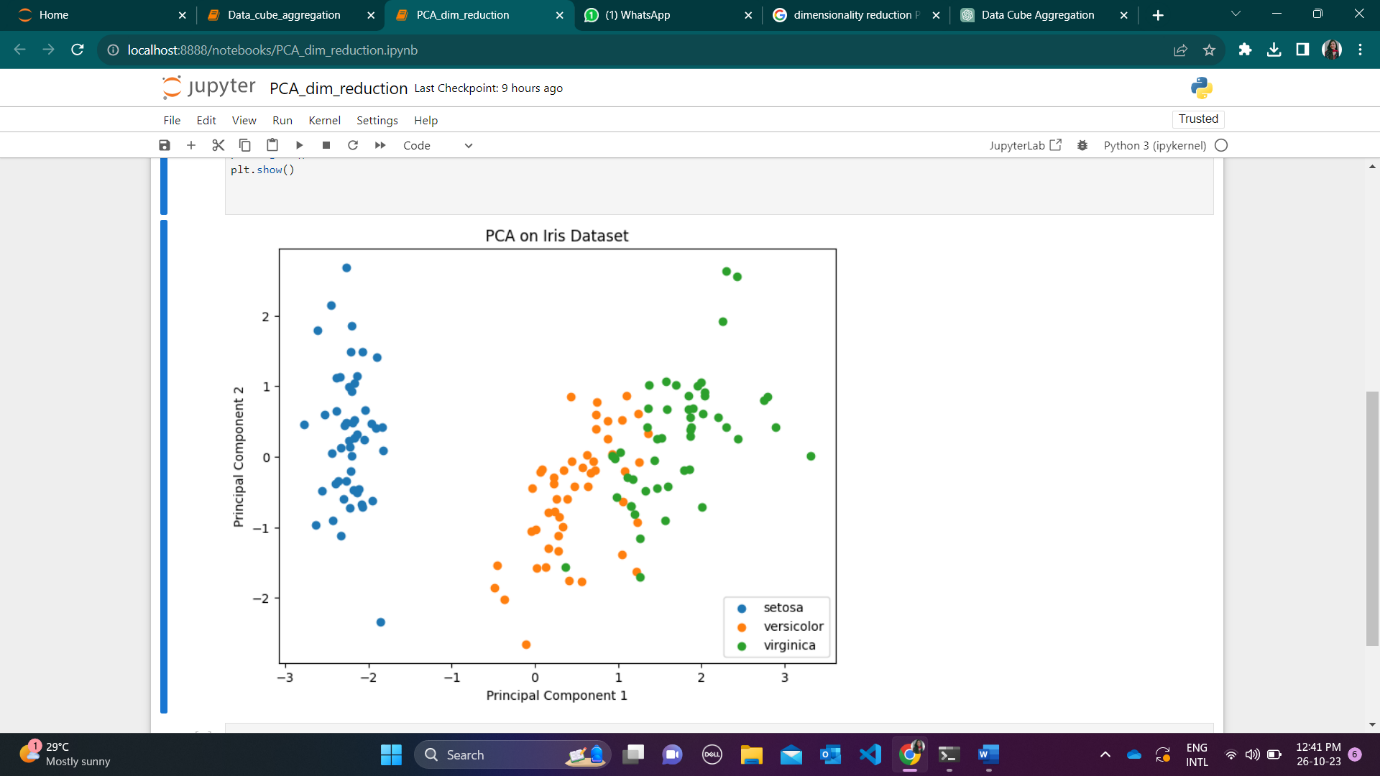
****

Fig-3: Output of Dimensionality Reduction using PCA

**Chapter - 5**

**Project Outcome:**

Through this project, the following Course Outcome was achieved:

CO4: Apply the basic concept of data mining and its functionality.

**Project Conclusion:**

In conclusion, the integration of Data Cube Aggregation and Dimensionality Reduction (PCA) techniques with a real-time database in Python offers a powerful solution for efficient data processing and analysis. By leveraging these methods, it becomes possible to handle large datasets effectively, reduce data dimensions while preserving key information, and ensure streamlined storage and retrieval in a real-time environment, fostering improved decision-making and data management.

GitHub Link: <https://github.com/AnshikaSharma210/PBL_sem7_AIDM>