**BIG DATA ANALYSIS WITH CLOUD DATABASES**

Project Title: Big Data Analysis

**Problem Statement :-**

* Dive into the world of big data analysis with IBM Cloud Databases.
* Uncover hidden insights from vast datasets, from climate trends to social patterns.
* Visualize your findings and derive valuable business intelligence.
* Embark on data-driven adventures, exploring the endless possibilities of big data.

**Required Libraries:-**

1. **Pandas:**

Pandas is a powerful data manipulation library for handling structured data. It provides data structures like Data Frames and tools for data cleaning and analysis.

1. **Numpy:**

numpy is a fundamental library for numerical computing in Python. It’s crucial for performing mathematical and statistical operations on large datasets.

1. **Matpoltlib or Seaborn:**

These libraries are used for data visualization. Matplotlib is a versatile library for creating various types of plots, while Seaborn is built on Matplotlib and simplifies creating aesthetically pleasing statistical visualizations.

1. **Hadoop Ecosystem (PySpark):**

If you are working with big data, you can use PySpark, the Python library for Apache Spark, which is a popular framework for distributed data processing. This is especially useful for large-scale data analysis.

1. **Ibm-cos-sdk:**

If you’re using IBM Cloud Object Storage, you can use the ibm-cos-sdk library to interact with your COS buckets and objects.

1. **IBM Watson SDKs:**

If you are integrating with IBM Watson services or using IBM Cloud offerings like Watson Studio, you might need the relevant Watson SDKs for Python.

1. **date time**:

For working with date and time data, you can use Python’s built-in datetime module.

1. **scikit-learn:**

If your analysis includes machine learning tasks, scikit-learn provides a wide range of machine learning algorithms and tools.

1. **Ibm-watson-machine-learning:**

If you are working on machine learning projects, the ibm-watson-machine-learning Python library can help you deploy and manage models on IBM Cloud.

**10.PyArrow (for efficient data exchange):**

When working with large datasets, you may use PyArrow for efficient data exchange between different data formats and systems.

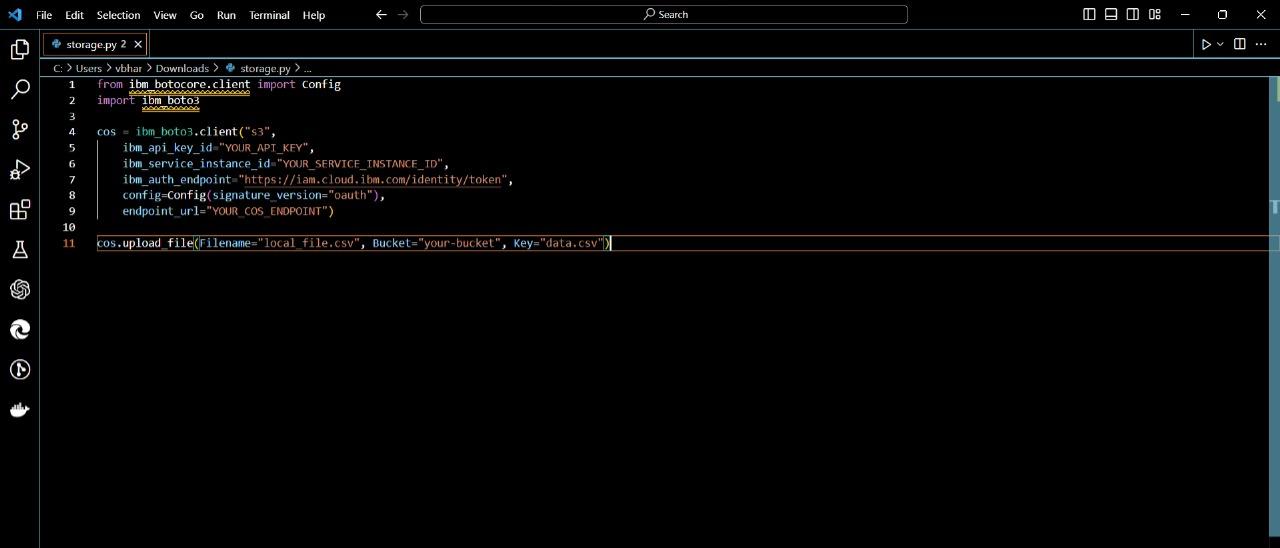
**11.IBM Cloud CLI:**

If you need to manage IBM Cloud services and resources from the command line, you’ll require the IBM Cloud Command Line Interface (CLI).

**Method of Approach with Implementation :-**

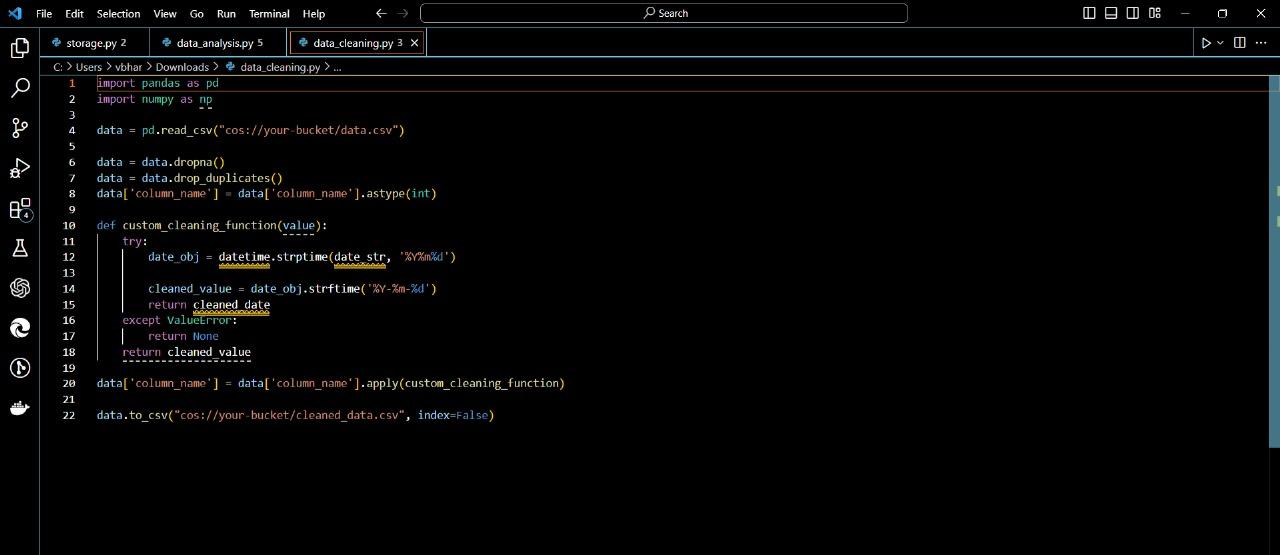
1. **Data Storage:**

Begin by storing your datasets in IBM Cloud Object Storage (COS). You can use the ibm - cos-sdk library to interact with COS and upload your data.



**2.Data Cleaning with Pandas and Numpy:**

Use Python, Pandas, and Numpy for data cleaning. Here’s an example of a cleaning function:



**3. Data Analysis:**

Utilize Pandas and Numpy for data analysis. Calculate statistics, create visualizations, and derive insights:



**4.Execute in IBM Cloud Environment:**

You can run this Python code in various IBM Cloud environments such as IBM Watson Studio or IBM Cloud Functions, depending on your specific requirements.

**5.Ensure Access and Authentication:**

Make sure you have the necessary credentials and permissions to access IBM Cloud services from your Python code. Replace placeholders with your actual credentials, service instance IDs, and endpoints.