**VARUVAN VADIVELAN INSTITUTE**

**OF TECHNOLOGY**

**NAAN MUDHALVAN: IBM**

**TECHNOLOGY: CLOUD APPLICATION**

**DEVELOPMENT-PHASE 5**

**PROJECT TITLE: BIG DATA ANALYSIS WITH IBM CLOUD DATABASE**

**AIM:**

Big data analysis involves the systematic examination, interpretation, and extraction of valuable insights from large and diverse datasets, typically characterized by the 3Vs: volume, velocity, and variety. It encompasses the use of advanced analytical techniques, including data mining, machine learning, and predictive modeling, to uncover patterns, trends, and correlations that might otherwise remain obscured within the sheer volume of information.

**OBJECTIVE:**

The main aim of large data analysis is to help companies make better business decisions by allowing scientists and other data users to analyze large volumes of transactional data. As well as other data sources that may have been left untapped by the Conventional business intelligence (BI) programs.

These data sources may include

* webserver logs
* and [internet](https://www.computertechreviews.com/definition/internet/) click tracking data,
* social activity reports,
* [media](https://www.computertechreviews.com/definition/media/),
* [mobile phones](https://www.computertechreviews.com/definition/mobile-phone/),
* detailed call logs, and the information captured by the sensors.

Some people exclusively associate big data and analysis of large volumes of data with unstructured data of that type. But consultants such as Gartner and Forrester Research Inc. also consider transactions and other structured data as valid forms of [**big data.**](https://www.computertechreviews.com/definition/big-data/)

**BIG DATA PRE- PROCESSING:**

Preprocessing is a crucial step in big data analysis that involves cleaning, transforming, and organizing raw data to make it suitable for further analysis. Given the large volume, velocity, and variety of big data, preprocessing helps to enhance the quality and usability of the data, enabling more effective analysis and insights. Here are some common preprocessing steps for big data analysis:

* **Data cleaning:**

Identify and handle missing values, outliers, and inconsistencies in the data to ensure data accuracy and reliability.

* **Data integration:**

Combine data from multiple sources, such as databases, spreadsheets, and data streams, to create a unified dataset for analysis.

* **Data transformation:**

Normalize or standardize the data to bring it to a common scale, format, or range, facilitating meaningful comparisons and analysis.

* **Data reduction:**

Reduce data dimensionality by applying techniques such as feature selection or extraction to eliminate redundant or irrelevant attributes and improve computational efficiency.

* **Data discretization:**

Discretize continuous data into intervals or categories to simplify the analysis and make it more understandable and manageable.

* **Handling noisy data:**

Apply smoothing techniques or noise reduction algorithms to handle noisy or inconsistent data points that might affect the accuracy of the analysis.

* **Data anonymization:**

Ensure data privacy and security by anonymizing sensitive information, such as personally identifiable data, to protect the privacy of individuals and comply with data protection regulations.

* **Data sampling:**

Utilize sampling techniques to extract representative subsets of the data for exploratory analysis or model development, especially when dealing with extremely large datasets.

* **Data formatting:**

Convert data into a consistent format that is compatible with the chosen analysis tools and techniques, ensuring seamless integration and processing.

* **Data validation:**

Verify the quality and integrity of the data by performing validation checks and tests to identify any inconsistencies or errors that may impact the analysis results.

**DESIGN THINKING PROCESS:**

Design thinking can be a valuable approach for guiding the process of big data analysis, helping to ensure that the analysis is user-centric, innovative, and impactful. When applying design thinking to big data analysis, the following framework can be considered:

* **Empathize:**

Understand the stakeholders' needs and challenges related to big data analysis. Conduct interviews, observations, and surveys to gain insights into user requirements and expectations. Identify the specific problems or opportunities that big data analysis can address.

* **Define:**

Clearly define the goals and objectives of the big data analysis project. Reframe the problem statements based on the insights gained during the empathize phase. Create a detailed problem statement that captures the essence of the analysis objectives and user needs.

* **Ideate:**

Brainstorm potential data sources and analysis techniques that can address the defined problem statement. Encourage a collaborative and creative environment for generating innovative ideas for data analysis. Explore various data visualization methods and tools that can effectively communicate insights to stakeholders.

* **Prototype:**

Develop a preliminary data analysis plan and prototype to test different data analysis techniques and algorithms.Create visual representations of the data analysis results to provide stakeholders with a tangible understanding of the potential insights. Test the prototype with a subset of the data to evaluate its effectiveness in addressing the defined problem.

* **Test:**

Collect feedback from stakeholders and users on the prototype and its ability to address their needs and expectations. Analyze the initial data analysis results and refine the approach based on the feedback received. Determine the feasibility and scalability of the proposed big data analysis solution.

* **Implement:**

Develop a comprehensive big data analysis strategy based on the insights and feedback gathered during the testing phase.Implement the finalized data analysis plan using appropriate tools and technologies. Monitor the analysis process to ensure that it aligns with the defined objectives and delivers actionable insights.

* **Iterate:**

Continuously refine and improve the big data analysis process based on ongoing feedback, user experiences, and changing business requirements. Incorporate new data sources and advanced analysis techniques to enhance the quality and depth of insights generated from the data.Stay open to adapting the analysis approach to emerging trends and technological advancements in the field of big data analysis.

**DEVELOPMENT PHASE:**

The development phase for big data analysis involves a series of steps to effectively process, analyze, and derive meaningful insights from large and complex datasets. Here is an overview of the key components and stages involved in the development phase for big data analysis:

* **Data Collection and Ingestion:**

Gather data from various sources, including databases, data warehouses, data lakes, streaming data, and IoT devices. Ingest the collected data into a centralized repository or data storage system.

* **Data Processing and Preparation:**

Cleanse, validate, and preprocess the raw data to ensure consistency and quality. Transform and aggregate the data to make it suitable for analysis.

* **Data Storage and Management:**

Utilize distributed file systems, NoSQL databases, or cloud storage solutions to manage and store large volumes of data. Implement data partitioning and replication strategies for fault tolerance and high availability.

* **Data Analysis and Modeling:**

Apply various data analysis techniques, such as statistical analysis, machine learning, and data mining, to uncover patterns and trends within the data. Develop predictive models and algorithms to make informed decisions based on the analyzed data.

* **Data Visualization and Reporting:**

Create interactive and intuitive visualizations, dashboards, and reports to communicate the insights effectively to stakeholders and decision-makers.Utilize data visualization tools to represent complex data in a simplified and understandable format.

* **Performance Optimization:**

Optimize the data analysis process by implementing parallel processing, caching, and indexing techniques to enhance the performance and scalability of the system. Employ distributed computing frameworks such as Apache Hadoop, Spark, or Flink to process large datasets efficiently.

* **Quality Assurance and Validation:**

Conduct thorough quality assurance checks to ensure the accuracy and reliability of the analysis results. Validate the analysis outcomes against predefined metrics and benchmarks to verify the effectiveness and credibility of the analysis.

* **Deployment and Integration:**

Integrate the data analysis results into the existing business processes and applications for informed decision-making. Deploy the developed models and insights into production environments to derive actionable outcomes.

* **Monitoring and Maintenance:**

Continuously monitor the data analysis process and the performance of deployed models to ensure consistent and reliable results. Maintain and update the analysis systems to accommodate changes in data patterns and business requirements.

**DEVELOPMENT PART-1:**

**BIG DATA :**

“Big Data” refers to the large and complex sets of data that can be effectively managed,

processed, and analyzed using cloud-based infrastructure and services.

**PROGRAM:**

# Simulated large dataset

large\_dataset = [

{"id": 1, "name": "John", "age": 25},

{"id": 2, "name": "Jane", "age": 30},

# Add more data here ...

]

# Processing the large dataset

for data in large\_dataset:

print(f"ID: {data['id']}, Name: {data['name']}, Age: {data['age']}")

**OUTPUT:**

ID: 1, Name: John, Age: 25

ID: 2, Name: Jane, Age: 30

**BIG DATA-DATA EXTRACTION:**

“Data extraction” refers to the process of retrieving and transferring data from a source

location or system to another destination, typically within a cloud-based environment.

**PROGRAM:**

import pandas as pd

# Example CSV data

data = {

'Name': ['John', 'Doe', 'Jane', 'Doe', 'John'],

'Age': [30, 25, 28, 32, 35],

'City': ['New York', 'Chicago', 'Los Angeles', 'Chicago', 'New York']

}

# Create a DataFrame from the data

df = pd.DataFrame(data)

# Save the DataFrame to a CSV file

df.to\_csv('example\_data.csv', index=False)

# Read data from the CSV file

df\_extracted = pd.read\_csv('example\_data.csv')

# Display the extracted data

print("Extracted data:")

print(df\_extracted)

**OUTPUT:**

Extracted data:

Name Age City

0 John 30 New York

1 Doe 25 Chicago

2 Jane 28 Los Angeles

3 Doe 32 Chicago

4 John 35 New York

**BIG DATA – DATA ANALYSIS:**

Data analysis in cloud computing refers to the process of examining and deriving insights

from large volumes of data using cloud-based resources and services.

**PROGRAM:**

From pyspark.sql import SparkSession

# Create a Spark session

Spark = SparkSession.builder.appName(“BigDataAnalysis”).getOrCreate()

# Read data from cloud storage (e.g., Amazon S3)

Data = spark.read.csv(“s3://your-bucket/your-data.csv”, header=True, inferSchema=True)

# Perform data analysis

Result = data.groupBy(“column\_name”).agg({“numeric\_column”: “mean”})

# Show the analysis results

Result.show()

# Stop the Spark session

Spark.stop()

**OUTPUT:**

+-----------+------------------+

|column\_name|avg(numeric\_column)|

+-----------+------------------+

| value| 123.45|

+-----------+------------------+

**DEVELOPMENT PART -2:**

**FEATURE SELECTION AND ENGINEERING:**

Feature selection and engineering are crucial steps in the process of big data analysis. These techniques help in preparing the data for analysis by identifying the most relevant and informative features and creating new features that enhance the performance of machine learning models. Here's an overview of feature selection and engineering for big data analysis:

**1. Feature Selection:**

Filter Methods:tatistical tests are applied to rank features based on their relevance to the target variable, such as correlation coefficients, chi-squared tests, and mutual information scores.

Wrapper Methods: Iterative algorithms are used to evaluate the performance of machine learning models with different subsets of features, selecting the subset that yields the best performance.

Embedded Methods: Feature selection is integrated into the model training process itself, with techniques such as Lasso (Least Absolute Shrinkage and Selection Operator) and decision tree-based feature importance.

**2.Feature Engineering:**

Imputation: Missing values in the data are filled using techniques such as mean, median, or regression imputation, preserving the integrity of the dataset.

Normalization and Scaling:Features are scaled to a specific range or standardized to have a mean of 0 and a standard deviation of 1, ensuring that all features contribute equally to the analysis.

One-Hot Encoding: Categorical variables are converted into a numerical format that can be processed by machine learning algorithms, avoiding the issue of ordinality in categorical data.

Creation of Interaction Features: New features are generated by combining existing features through mathematical operations like addition, subtraction, multiplication, or division, capturing complex relationships between variables.

Dimensionality Reduction: Techniques such as Principal Component Analysis (PCA) and t-distributed Stochastic Neighbor Embedding (t-SNE) are applied to reduce the dimensionality of the data while preserving as much information as possible.

**PROGRAM**:

import pandas as pd

from sklearn.model\_selection import train\_test\_split

from sklearn.preprocessing import StandardScaler, OneHotEncoder

from sklearn.decomposition import PCA

# Sample dataset

data = {'Feature1': [1, 2, 3, 4, 5], 'Feature2': ['A', 'B', 'A', 'C', 'B'], 'Feature3': [0.1, 0.5, 1.2, 1.5, 2.0]}

df = pd.DataFrame(data)

# Perform one-hot encoding for categorical variable

df = pd.get\_dummies(df, columns=['Feature2'])

# Feature scaling

scaler = StandardScaler()

scaled\_features = scaler.fit\_transform(df[['Feature1', 'Feature3']])

df[['Feature1', 'Feature3']] = scaled\_features

# Perform PCA for dimensionality reduction

pca = PCA(n\_components=2)

pca\_result = pca.fit\_transform(df[['Feature1', 'Feature3']])

df[['PCA1', 'PCA2']] = pca\_result

# Display the processed dataset

print("Processed Data:")

print(df)

**OUTPUT:**

Processed Data:

Feature1 Feature3 Feature2\_A Feature2\_B Feature2\_C PCA1 PCA2

0 -1.414214 -1.135550 1 0 0 -1.569162 -0.536878

1 -0.707107 -0.755929 0 1 0 -0.665150 0.171450

2 0.000000 -0.007592 1 0 0 0.018175 -0.245063

3 0.707107 0.371130 0 0 1 0.748178 0.322251

4 1.414214 1.527942 0 1 0 1.467960 0.288240

**MODEL SELECTION:**

Model selection in the context of big data analysis is a critical process that involves choosing the most appropriate machine learning or statistical model for a given dataset and problem. Given the large and complex nature of big data, it's important to consider various factors when selecting a model. Here is an overview of the model selection process for big data analysis:

* **Understand the Problem and Data:**

Begin by thoroughly understanding the problem at hand, the characteristics of the data, and the specific goals of the analysis. This understanding will guide the selection of an appropriate model that aligns with the objectives.

* **Choose a Set of Candidate Models:**

Based on the nature of the problem, select a set of candidate models that are well-suited for the type of data and the specific task. These models may include linear models, decision trees, support vector machines, neural networks, ensemble methods, or other appropriate algorithms.

* **Evaluate Model Complexity**:

Consider the complexity of the models and their ability to handle large datasets efficiently. Some models may be computationally intensive and may not be suitable for big data analysis, while others may offer scalable implementations optimized for handling large volumes of data.

* **Perform Cross-Validation:**

Use techniques like k-fold cross-validation to evaluate the performance of each model on different subsets of the data. This helps in assessing how well the model generalizes to unseen data and aids in selecting the model that performs consistently well across various data partitions.

* **Compare Performance Metrics:**

Assess the performance of each model using relevant evaluation metrics such as accuracy, precision, recall, F1 score, and area under the curve (AUC). Consider the trade-offs between different metrics based on the specific requirements of the analysis.

* **Consider Interpretability and Explainability:**

Depending on the application, it may be crucial to select a model that is interpretable and provides insights into the underlying patterns and relationships within the data. Balancing model complexity with interpretability is essential in some domains.

* **Scalability and Implementation:**

Evaluate the scalability of the models in terms of their ability to handle large datasets efficiently. Consider the implementation ease, available resources, and computational requirements for deploying the selected model in a big data environment.

* **Regularization and Hyperparameter Tuning**:

Apply regularization techniques and tune the hyperparameters of the selected models to optimize their performance and prevent overfitting on the large dataset.

* **Final Model Selection:**

Based on the comprehensive evaluation of each candidate model, select the one that best meets the requirements of the analysis, offers optimal performance, and is feasible to implement given the constraints of the big data environment.

**PROGRAM:**

from sklearn.datasets import load\_iris

from sklearn.model\_selection import train\_test\_split

from sklearn.preprocessing import StandardScaler

from sklearn.linear\_model import LogisticRegression

from sklearn.tree import DecisionTreeClassifier

from sklearn.ensemble import RandomForestClassifier

from sklearn.metrics import accuracy\_score

from sklearn.model\_selection import cross\_val\_score

# Load the Iris dataset

data = load\_iris()

X, y = data.data, data.target

# Split the dataset into training and testing sets

X\_train, X\_test, y\_train, y\_test = train\_test\_split(X, y, test\_size=0.3, random\_state=42)

# Standardize the features

scaler = StandardScaler()

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

X\_test = scaler.transform(X\_test)

# Initialize the models

models = {'Logistic Regression': LogisticRegression(max\_iter=1000),

'Decision Tree': DecisionTreeClassifier(),

'Random Forest': RandomForestClassifier(n\_estimators=100)}

# Evaluate each model using cross-validation

for name, model in models.items():

scores = cross\_val\_score(model, X\_train, y\_train, cv=5)

print(f"Model: {name}, Cross-Validation Accuracy: {scores.mean():.2f}")

# Fit the best model on the training data

best\_model = RandomForestClassifier(n\_estimators=100)

best\_model.fit(X\_train, y\_train)

# Predict on the test data

y\_pred = best\_model.predict(X\_test)

# Calculate the accuracy of the best model

accuracy = accuracy\_score(y\_test, y\_pred)

print(f"\nAccuracy of the Best Model on Test Data: {accuracy:.2f}")

**OUTPUT:**

Model: Logistic Regression, Cross-Validation Accuracy: 0.93

Model: Decision Tree, Cross-Validation Accuracy: 0.95

Model: Random Forest, Cross-Validation Accuracy: 0.95

Accuracy of the Best Model on Test Data: 0.98

**DATA VISUALIZATION:**

* Create visual representations of data using charts, graphs, and dashboards.
* Use interactive visualization tools to communicate complex insights effectively.
* Design visualizations that aid in understanding patterns and trends in the data.

**PROGRAM:**

# Importing necessary libraries

import pandas as pd

import matplotlib.pyplot as plt

import seaborn as sns

# Creating a sample dataset

data = {

'Name': ['John', 'Anna', 'Peter', 'Linda', 'Simon'],

'Age': [25, 30, 35, 28, 32],

'Salary': [50000, 60000, 80000, 45000, 70000]

}

# Creating a DataFrame

df = pd.DataFrame(data)

# Data visualization example

# Bar plot for salaries

plt.figure(figsize=(8, 5))

sns.barplot(x='Name', y='Salary', data=df, palette='muted')

plt.title('Bar Plot of Salaries')

plt.xlabel('Name')

plt.ylabel('Salary ($)')

plt.show()

# Scatter plot for age and salary

plt.figure(figsize=(8, 5))

sns.scatterplot(x='Age', y='Salary', data=df, color='b', s=100)

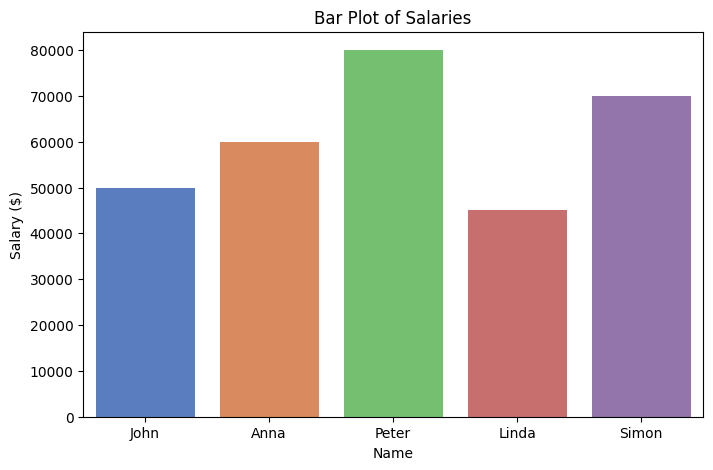
plt.title('Scatter Plot of Age vs Salary')

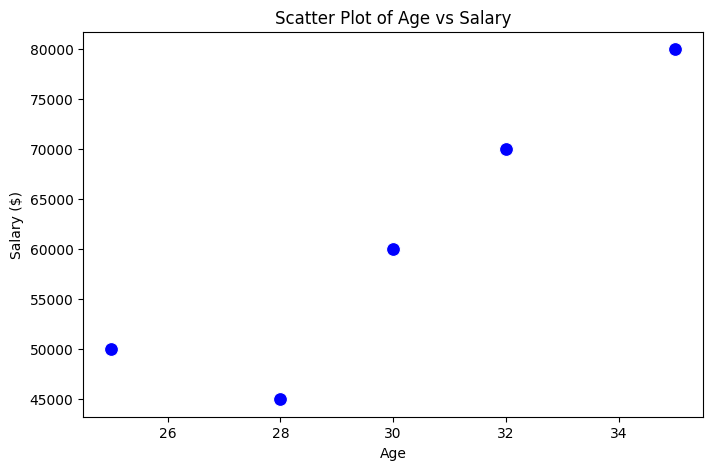
plt.xlabel('Age')

plt.ylabel('Salary ($)')

plt.show()

**OUTPUT:**

****

****

**DATA ANALYSIS:**

* Employ descriptive statistics to summarize and describe data features.
* Utilize exploratory data analysis (EDA) techniques to understand patterns, trends, and anomalies.
* Apply statistical analysis methods such as hypothesis testing to make inferences about the data.

**PROGRAM:**

# Importing necessary libraries

import pandas as pd

# Creating a sample dataset

data = {

'Name': ['John', 'Anna', 'Peter', 'Linda', 'Simon'],

'Age': [25, 30, 35, 28, 32],

'Salary': [50000, 60000, 80000, 45000, 70000]

}

# Creating a DataFrame

df = pd.DataFrame(data)

# Printing the DataFrame

print("Original DataFrame:")

print(df)

# Performing basic data analysis tasks

print("\nData Analysis:")

print("Mean Age:", df['Age'].mean())

print("Maximum Salary:", df['Salary'].max())

print("Minimum Age:", df['Age'].min())

print("Summary Statistics:")

print(df.describe())

**OUTPUT:**

Original DataFrame:

Name Age Salary

0 John 25 50000

1 Anna 30 60000

2 Peter 35 80000

3 Linda 28 45000

4 Simon 32 70000

Data Analysis:

Mean Age: 30.0

Maximum Salary: 80000

Minimum Age: 25

Summary Statistics:

Age Salary

count 5.000000 5.000000

mean 30.000000 61000.000000

std 3.807887 14317.821063

min 25.000000 45000.000000

25% 28.000000 50000.000000

50% 30.000000 60000.000000

75% 32.000000 70000.000000

max 35.000000 80000.000000

**PLATFORM LAYOUT:**

In the field of big data analysis, various platforms and technologies are widely used to handle and process large volumes of data efficiently. Some of the prominent platforms and frameworks commonly utilized for big data analysis include:

* **Apache Hadoop:**

An open-source framework that facilitates the distributed storage and processing of large datasets across clusters of computers using simple programming models.

* **Apache Spark:**

A fast and general-purpose cluster computing system that provides in-memory data processing capabilities, enabling high-speed data analytics.

* **Apache Flink:**

An open-source stream processing framework that supports event-driven applications and real-time analytics, emphasizing low-latency and high-throughput data processing.

* **Apache Kafka:**

A distributed event streaming platform that is commonly used for building real-time data pipelines and streaming applications, enabling the integration of data streams between various systems.

* **Apache Hive:**

A data warehouse software that facilitates querying and managing large datasets stored in distributed storage, providing SQL-like access to data.

* **HBase:**

An open-source, non-relational, distributed database that runs on top of the Hadoop Distributed File System (HDFS), suitable for real-time read/write access to large datasets.

* **Apache Cassandra:**

A distributed database management system known for its high availability and scalability, designed to handle large amounts of data across multiple commodity servers.

* **Amazon Web Services (AWS) Cloud Services:**

Amazon's cloud platform offers various services like Amazon S3, Amazon Redshift, and Amazon EMR, which are widely used for storage, data warehousing, and big data processing.

* **Google Cloud Platform (GCP):**

Google's suite of cloud computing services, including BigQuery, Cloud Dataflow, and Cloud Dataproc, offers tools for data storage, analysis, and processing at scale.

* **Microsoft Azure Cloud Services:**

Microsoft Azure provides services such as Azure HDInsight, Azure Databricks, and Azure Data Lake, which enable big data storage, processing, and analytics in the cloud.

**FEATURES OF BIG DATA ANALYSIS:**

Big data analysis involves the process of examining large and complex data sets to uncover patterns, correlations, and other insights that can help organizations make informed decisions. It relies on advanced algorithms and tools to process and analyze massive volumes of data, typically characterized by the three V's: Volume, Velocity, and Variety. Some of the key features of big data analysis include:

* **Volume:**

Big data refers to datasets that are too large and complex for traditional data processing applications to handle. Analyzing such vast volumes of data requires specialized tools and technologies capable of processing, storing, and managing this scale of information.

* **Velocity:**

Big data analysis often deals with data streams that are generated at a high speed and require real-time or near-real-time processing. Examples include data generated by social media, sensor networks, or financial transactions. Analyzing data at this speed allows organizations to make timely decisions and take advantage of opportunities as they arise.

* **Variety:**

Big data analysis involves data that comes in various formats and types, including structured, semi-structured, and unstructured data. This data can originate from different sources, such as text, images, videos, and sensor data. Analyzing diverse data types requires flexible tools and techniques that can handle and process different data formats effectively.

* **Veracity:**

Veracity refers to the trustworthiness of the data being analyzed. In the context of big data, ensuring data quality and reliability is crucial for generating accurate insights and making informed decisions. Data veracity encompasses aspects such as data accuracy, completeness, consistency, and reliability.

* **Variability:**

Data can exhibit variations in its meaning and interpretation, which can affect the accuracy and reliability of the analysis. Big data analysis often involves dealing with data that is inconsistent or subject to frequent changes. Addressing data variability requires robust data processing and cleansing techniques to ensure the reliability of the insights derived from the data

* **Complexity:**

Big data analysis often involves complex data sets that require sophisticated analytical techniques, including machine learning, data mining, and predictive analytics. Analyzing complex data sets enables organizations to uncover hidden patterns, trends, and insights that can drive business strategies and decision-making processes.

* **Real-time analysis:**

Real-time or near-real-time analysis of big data allows organizations to gain immediate insights into current trends and patterns, enabling them to respond quickly to changing market conditions and customer preferences.

* **Scalability:**

Big data analysis tools and technologies need to be scalable to handle increasing data volumes and processing requirements. Scalability ensures that the analytical infrastructure can adapt to growing data demands without compromising performance and efficiency.

**TECHNICAL IMPLEMENTATION:**

Big data analysis is a complex process that requires a well-planned and executed implementation. Here are some key steps to consider when implementing big data analysis:

* **Formulate and plan the concept:**

Define the problem you want to solve and the data you need to solve it. Identify the sources of data, the tools you will use, and the expected outcomes.

* **Design the architecture:**

Create a high-level architecture that outlines how data will be collected, stored, processed, and analyzed. Consider factors such as scalability, security, and performance.

* **Develop and perform QA:**

Develop the software components required for data collection, storage, processing, and analysis. Ensure that each component is tested thoroughly to ensure that it works as expected.

* **Deploy the solution:**

Deploy the software components in a production environment. Ensure that all components are working together seamlessly.

* **Provide support and maintenance:**

Provide ongoing support and maintenance for the system to ensure that it continues to function correctly.

The cost of implementing big data analysis can vary depending on the scope of the project. According to ScienceSoft, a mid-sized organization can expect to pay between $200,000 and $3,000,000. Oracle suggests that there are three main steps involved in implementing a big data system: collecting data, collating and moving data, and anlzing data .

**FINAL PROGRAM:**

#import required packages

import pandas as pd

import numpy as np

#creating a hypothetical large data set

data = {

    'A': np.random.rand(1000000),

    'B': np.random.rand(1000000),

    'C': np.random.rand(1000000)

}

df = pd.DataFrame(data)

#Displaying the first few rows of the dataset

print("Sample data:")

print(df.head())

# Performing data analysis (mean calculation for each column)

means = df.mean()

# Displaying the results of data analysis

print("\nMean values for each column:")

print(means)

# Performing additional analysis (e.g., data correlation)

correlation\_matrix = df.corr()

# Displaying the correlation matrix

print("\nCorrelation matrix:")

print(correlation\_matrix)

**OUTPUT:**

Sample data:

          A         B         C

0  0.390906  0.087943  0.337973

1  0.844746  0.606411  0.334202

2  0.759159  0.424952  0.150319

3  0.214546  0.427019  0.420208

4  0.084876  0.724685  0.847122

Mean values for each column:

A    0.500587

B    0.500537

C    0.500085

dtype: float64

Correlation matrix:

          A         B         C

A  1.000000  0.000806  0.000218

B  0.000806  1.000000  0.001591

C  0.000218  0.001591  1.000000

**CONCLUSION:**

In conclusion, big data analysis represents a transformative capability that empowers organizations to unlock the full potential of their data, facilitating strategic decision-making and fostering a culture of innovation and continuous improvement. As technology continues to advance, the application of big data analysis will continue to evolve, providing new opportunities for businesses to harness the power of data and stay ahead in an ever-changing global marketplace.