

BIGDATA ANALYTICS WITH IBM CLOUD DATABASES

COLLEGE: UNITED INSTITUTE OF TECHNOLOGY

COLLEGE CODE :7145

BATCHMATES: GOKULNATH S

LAKSHANA.A

MASANAM M

VISHAL R

RATHISH Y

VIGNESH S

MARISH P

GUIDE:Mrs.NISHANTHI

DOMAIN:CLOUD COMPUTING

GUIDE:Mrs.NISHANTHI

DOMAIN:CLOUD COMPUTING

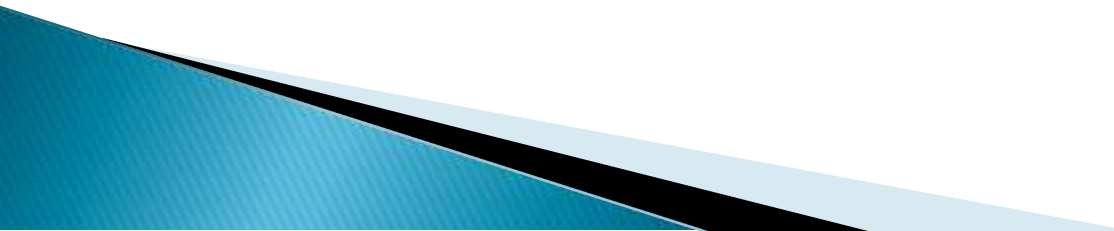
Innovation Challenges

Abstract

Big data' is massive amounts of information that can work wonders. It has become a topic of special interest for the past two decades because of a great potential that is hidden in it. Various public and private sector industries generate, store, and analyze big data with an aim to improve the services they provide. In the healthcare industry, various sources for big data include hospital records, medical records of patients, results of medical examinations, and devices that are a part of internet of things. Biomedical research also generates a significant portion of big data relevant to public healthcare.

That is exactly why various industries, including the healthcare industry, are taking vigorous steps to convert this potential into better services and financial advantages. With a strong integration of biomedical and healthcare data, modern healthcare organizations can possibly revolutionize the medical therapies and personalized medicine.

One such special social need is healthcare. Like every other industry, healthcare organizations are producing data at a tremendous rate that presents many advantages and challenges at the same time. In this review, we discuss about the basics of big data including its management, analysis and future prospects especially in healthcare sector.



The data overload

Every day, people working with various organizations around the world are generating a massive amount of data.

The term “digital universe” quantitatively defines such massive amounts of data created, replicated, and consumed in a single year. International Data Corporation (IDC) estimated the approximate size of the digital universe in 2005 to be 130 exabytes (EB).

The digital universe in 2017 expanded to about 16,000 EB or 16 zettabytes (ZB). IDC predicted that the digital universe would expand to 40,000 EB by the year 2020.

To imagine this size, we would have to assign about 5200 gigabytes (GB) of data to all individuals.

Defining big data

big data’ represents large amounts of data that is unmanageable using traditional software or internet-based platforms.

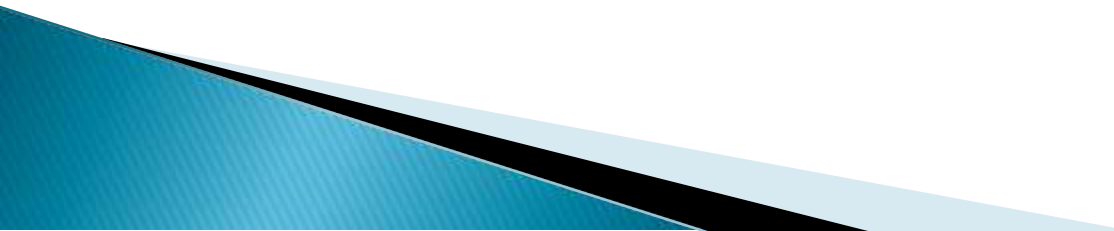
It surpasses the traditionally used amount of storage, processing and analytical power. Even though a number of definitions for big data exist, the most popular and well-accepted definition was given by Douglas Laney.

big data’ represents large amounts of data that is unmanageable using traditional software or internet-based platforms.

It surpasses the traditionally used amount of storage, processing and analytical power.

Even though a number of definitions for big data exist, the most popular and well-accepted definition was given by Douglas Laney.

In addition, visualization of big data in a user-friendly manner will be a critical factor for societal development.



Healthcare as a big-data repository

Healthcare is a multi-dimensional system established with the sole aim for the prevention, diagnosis, and treatment of health-related issues or impairments in human beings.

The major components of a healthcare system are the health professionals (physicians or nurses), health facilities (clinics, hospitals for delivering medicines and other diagnosis or treatment technologies), and a financing institution supporting the former two.

The health professionals belong to various health sectors like dentistry, medicine, midwifery, nursing, psychology, physiotherapy, and many others.

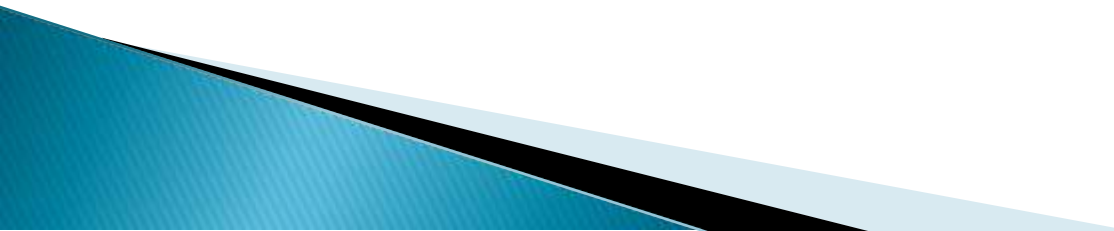
Healthcare is required at several levels depending on the urgency of situation. Professionals serve it as the first point of consultation (for primary care), acute care requiring skilled professionals (secondary care), advanced medical investigation and treatment (tertiary care) and highly uncommon diagnostic or surgical procedures (quaternary care).

Electronic health records

It is important to note that the National Institutes of Health (NIH) recently announced the “All of Us” initiative that aims to collect one million or more patients’ data such as EHR, including medical imaging, socio-behavioral, and environmental data over the next few years. EHRs have introduced many advantages for handling modern healthcare related data. Below, we describe some of the characteristic advantages of using EHRs.

The information includes medical diagnoses, prescriptions, data related to known allergies, demographics, clinical narratives, and the results obtained from various laboratory tests.

The recognition and treatment of medical conditions thus is time efficient due to a reduction in the lag time of previous test results.



Big data and analytics service ecosystem on cloud

A service provider can offer big data and analytics capabilities as a service ecosystem on a cloud, with organizations selecting their desired capabilities from a service catalog.

This big data and analytics capability service adoption model is suitable for most organizations, whether they are new to big data or have been working with it for years.

The model can be applied whether an organization needs only a few big data capabilities or many big data capabilities, whether or not it has in-house expertise, and even whether it wants to outsource the big data and analytics capability implementation.

The following list shows the most common big data capabilities that can be provided as services in a cloud model:

Extract, transform, and load (ETL) as a service


Visualization and search as a service **Analytics as a service**

Hadoop NoSQL as a service (analytics infrastructure)

Data warehousing as a service **Reporting as a service**

Entity analytics as a service

Policy and data governance as a service **Database as a service**



IBM SoftLayer for big data and analytics

With its SoftLayer cloud, IBM provides unique capabilities that enable companies to quickly deploy a public big data and analytics solution. The following key aspects of SoftLayer clouds differentiate them from other public cloud offerings:

Large, fast networks Companies offering cloud capabilities need significant bandwidth. A couple of servers in a single data center are not sufficient. The SoftLayer network and capacity are considerable. Thirteen separate data centers house about 200,000 servers at locations around the world (including the United States, Singapore, the United Kingdom, and the Netherlands), each connected to the others with dedicated 20 Gbps private network fiber links. SoftLayer has a triple network architecture (Public, Management, and Private).

Virtual servers for convenience, bare metal for performance A virtual server is a virtual machine running on a hypervisor (an operating system) and hosted on a physical server. The same physical server can host multiple virtual servers. So, one server running a Linux operating system can host additional virtual servers running Microsoft Windows or Linux. SoftLayer offers virtual servers that can be set up in minutes and scaled as needed.

In this kind of service hosting model, the service provider provides the service so that the subscribing organizations can focus on **their overall business and IT goals**, for example:

When the government intelligence agencies have an ad hoc need to perform **social media analytics** to assess and identify threats during national political events, they can subscribe to analytics as a service.

Financial organizations that need to generate quarterly and yearly reports can subscribe to reporting as a service to accomplish those goals without having to invest in technologies that will only be used occasionally.

Cyber security and crime-fighting agencies that must collaborate in their work can avoid the need for multiple software and hardware infrastructures by subscribing to information sharing as a service, where they can each have a common portfolio of services and systems to provide consistent results.

Several other SoftLayer features specifically support **Hadoop**:

- **Deeply integrated and tuned analytics** stack with best-of-breed cluster management
- Hadoop rack-awareness
- Exchange compression and anti-colocation of reduce allotments
- Increased **Data Input phase redundancy**
- HDFS buffer size tuning
- Modifiable exchange sort implementation

DEVELOPMENT PART

1

It can provide with a high-level overview of the steps need to follow to build a big data analysis solution using IBM Cloud Database:

1. Create an IBM Cloud Account:

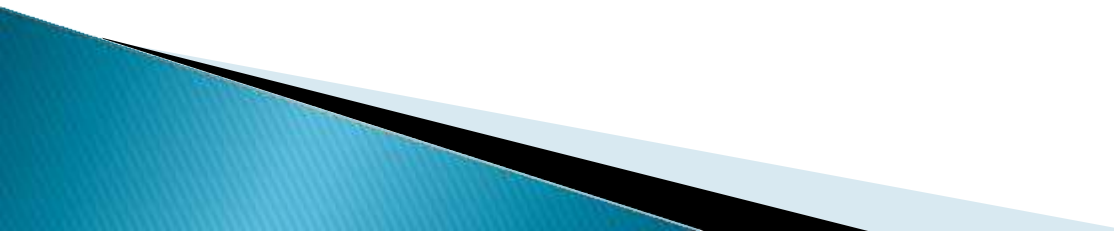
visit the IBM Cloud website and sign up for an account.

2. Choose the Database Service:

Determine which database service on IBM Cloud you want to use based on your specific requirements. You mentioned Db2 and MongoDB as options, but there are others like IBM Db2 on Cloud, IBM Cloud Databases for PostgreSQL, etc. Select the one that suits your needs best.

3. Set Up a Database Instance:

Create a database instance within your chosen service. You'll need to configure the database instance with specifications such as storage capacity, region, and access controls.



4. Import Your Dataset

Upload your dataset to the database instance. Depending on the database service you choose, this process may vary. For instance, if you're using MongoDB, you'd import JSON or BSON documents.

5. Develop Queries or Scripts:

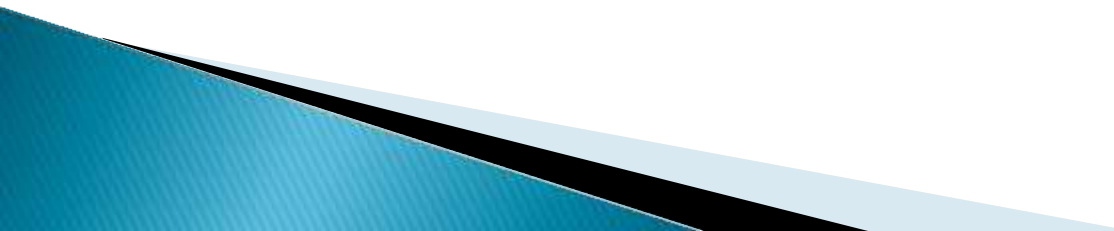
Write queries or scripts to explore and analyze your dataset. The specific queries will depend on the analysis you want to perform. For example, you might use SQL for relational databases like Db2 or MongoDB queries for NoSQL databases.

6. Data Cleaning and Transformation:

Implement data cleaning and transformation processes as needed. This might involve removing duplicates, handling missing values, and reshaping data to suit your analytical needs.

7. Perform Data Analysis:

Use the queries and scripts you developed to perform the actual data analysis. This could involve generating reports, creating visualizations, or running machine learning models, depending on your goals.



8. Optimize Performance

Monitor the performance of your database and queries. Make optimizations as necessary to ensure your analysis runs efficiently.

9. Security and Compliance:

Ensure that your solution complies with security and compliance standards. IBM Cloud provides tools and features to help with this.

10. Scale as Needed:

If your data and analysis requirements grow, scale your database instance accordingly. IBM Cloud allows for easy scalability.

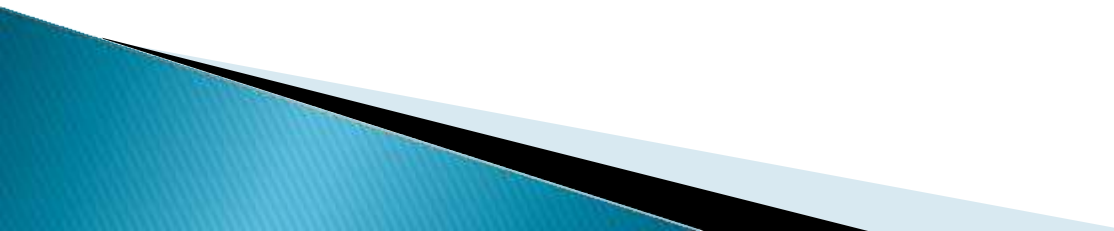


Sample Dataset:

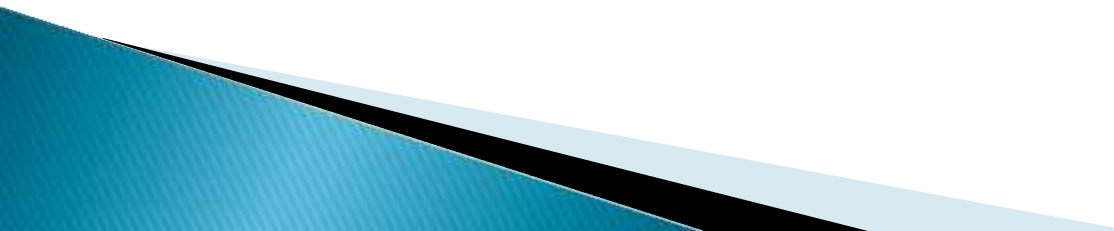
Assuming you have a MongoDB instance set up with a database named "ecommerce" and a collection named "orders." Here's a sample dataset:

json

```
[  
  {  
    "_id": 1,  
    "customer_name": "Alice",  
    "product": "Widget",  
    "quantity": 5,  
    "order_date": "2023-10-10"  
  },  
  {  
    "_id": 2,  
    "customer_name": "Bob",  
    "product": "Gadget",  
    "quantity": 3,  
    "order_date": "2023-10-12"
```



```
},  
{  
  "_id": 3,  
  "customer_name": "Alice",  
  "product": "Widget",  
  "quantity": 2,  
  "order_date": "2023-10-15"  
}  
]
```



Python Code to Analyze the Dataset:

Assuming the `pymongo` library installed, you can use Python to connect to your MongoDB instance and perform basic data analysis.

```
from pymongo import MongoClient
```

```
# Connect to the MongoDB instance
```

```
client = MongoClient('mongodb://your-mongodb-connection-url')
```

```
db = client['ecommerce']
```

```
collection = db['orders']
```

```
# Calculate total sales by product
```

```
pipeline = [
```

```
{
```

```
    "$group": {
```

```
        "_id": "$product",
```

```
        "total_sales": {"$sum": {"$multiply": ["$quantity", 10]}}
```

```
    }
```

```
}
```

```
]
```

```
sales_by_product = list(collection.aggregate(pipeline))
```

```
print("Total Sales by Product:")
```

```
for product in sales_by_product:
```

```
print(f'{product['_id']}: ${product['total_sales']})
```

```
# Find the most recent order date
```

```
latest_order = collection.find_one(sort=[('order_date', -1)])
```

```
print(f'The most recent order was placed on {latest_order['order_date']} by  
{latest_order['customer_name']})
```

Output for the given Python code with the sample dataset could be:

Output for the given Python code with the sample dataset could be:

```
Total Sales by Product:
```

```
Widget: $70
```

```
Gadget: $30
```

```
The most recent order was placed on  
2023-10-15 by Alice
```

conclusion output:

Output for the given Python code with the sample dataset could be:

Total Sales by Product:

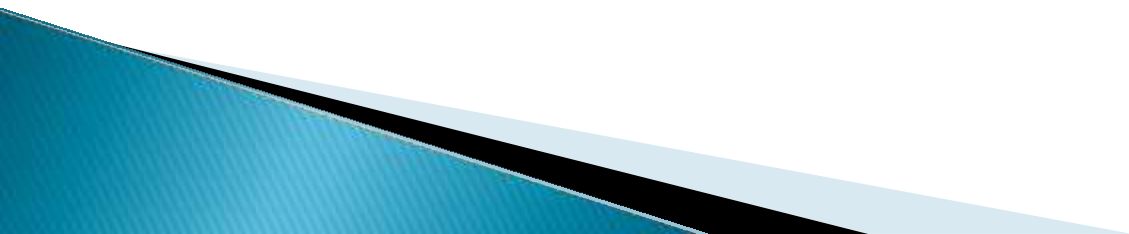
Widget: \$70

Gadget: \$30

The most recent order was placed on 2023-10-15 by Alice

In this output:

1. The first part of the output shows the total sales by product, where "Widget" generated \$70 in sales, and "Gadget" generated \$30 in sales.
2. The second part of the output tells you that the most recent order was placed on October 15, 2023, by Alice.



BIG DATA ANALYSIS WITH IBM CLOUD DATABASES

PHASE :4 DEVELOPEMENT NO 2

PROBLEM STATEMENT:

- Continue building the big data analysis solution by applying advanced analysis techniques and visualizing the results.
- Apply more complex analysis techniques, such as machine learning algorithms, time series analysis, or sentiment analysis, depending on the dataset and objectives.
- Create visualizations to showcase the analysis results. Use tools like Matplotlib, Plotly, or IBM Watson Studio for creating graphs and charts.

SOLUTION:

Certainly, building a big data analysis solution that incorporates advanced techniques and visualizations is essential for deriving meaningful insights from your data. Let's continue with the process:

1. Select Appropriate Analysis Techniques:

Depending on the nature of your dataset and specific objectives, consider various advanced analysis techniques:

Machine Learning Algorithms: Use supervised or unsupervised machine learning algorithms like decision trees, random forests, support vector machines, or clustering algorithms for predictive modeling or pattern recognition.

Time Series Analysis: If your data involves time-based data points, use time series analysis techniques to identify trends, seasonality, and forecast future values.

Sentiment Analysis: Apply natural language processing techniques to extract sentiment from text data, useful for social media or customer reviews analysis.

2. Data Preprocessing:

Ensure your data is prepared for analysis, which includes data cleaning, feature engineering, and data transformation. This step is crucial for the success of advanced analytics.

3. Machine Learning Model Development:

If you're using machine learning, split your dataset into training and testing sets, train various models, and evaluate their performance through metrics such as accuracy, precision, recall, and F1-score.

4. Time Series Analysis:

If you're working with time series data, perform decomposition to separate the time series into trend, seasonality, and residual components. Then, use forecasting techniques like ARIMA or LSTM neural networks to make predictions.

5. Sentiment Analysis:

For sentiment analysis, tokenize and preprocess text data, then employ NLP libraries like NLTK or spaCy for feature extraction and sentiment classification.

6. Create Visualizations:

Use visualization libraries like Matplotlib, Plotly, or IBM Watson Studio for showcasing the analysis results. Here are some examples of the types of visualizations you can create:

Bar Charts and Pie Charts: For showing categorical data or proportions.

Line Charts: Ideal for visualizing time series data and trends over time.

Scatter Plots: To explore relationships between variables or clusters.

Heatmaps: Useful for displaying correlation matrices.

Box Plots: To visualize data distributions and identify outliers.

Word Clouds: For visualizing the most frequent words in text data (sentiment analysis).

Confusion Matrices and ROC Curves: If using machine learning, visualize model performance metrics.

7. Interactive Dashboards:

For a more dynamic and interactive representation of your analysis, consider building web-based dashboards using tools like Tableau, Power BI, or Dash by Plotly. Dashboards allow stakeholders to explore the data and insights in real-time.

8. Documentation and Communication:

Document your analysis process, including the steps taken, parameters used, and the rationale behind your decisions. Effective communication is crucial for sharing insights and findings with your team or stakeholders.

9. Iterate and Refine:

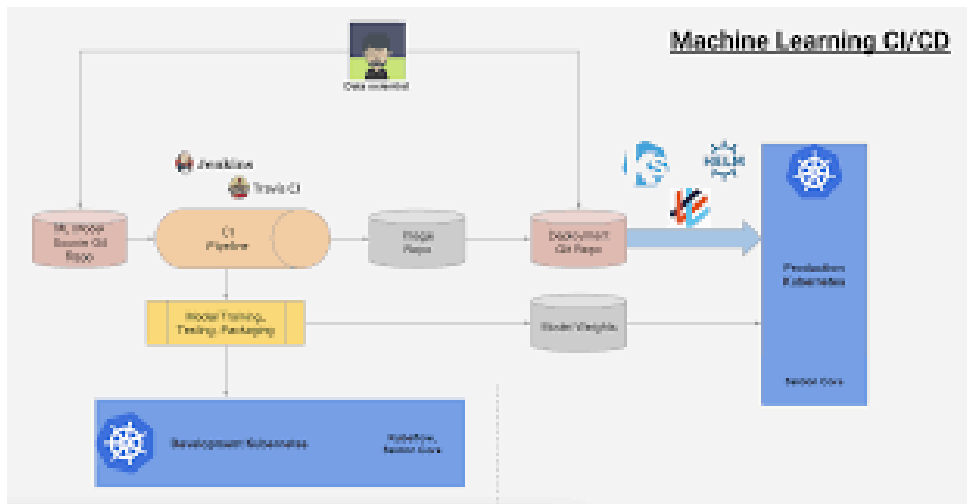
Big data analysis is often an iterative process. Analyze the results, gather feedback, and refine your analysis to gain deeper insights or improve model performance.

10. Automate and Operationalize:

If this analysis needs to be regularly updated or used in a production environment, consider automating the process and integrating it into your organization's systems.

Remember to tailor your analysis techniques and visualizations to your specific dataset and objectives, as different data types and business goals may require different approaches.

MACHINE LEARNING DEPLOYMENT MODEL:



USING PYTHON:

```
import matplotlib.pyplot as plt

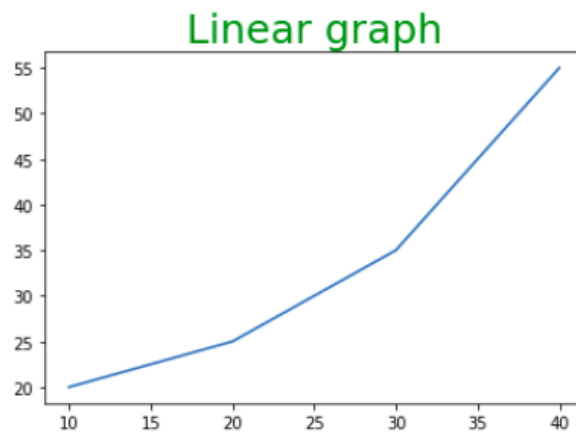
# initializing the data
x = [10, 20, 30, 40]
y = [20, 25, 35, 55]

# plotting the data
plt.plot(x, y)

# Adding title to the plot
plt.title("Linear graph", fontsize=25, color="green")

plt.show()
```

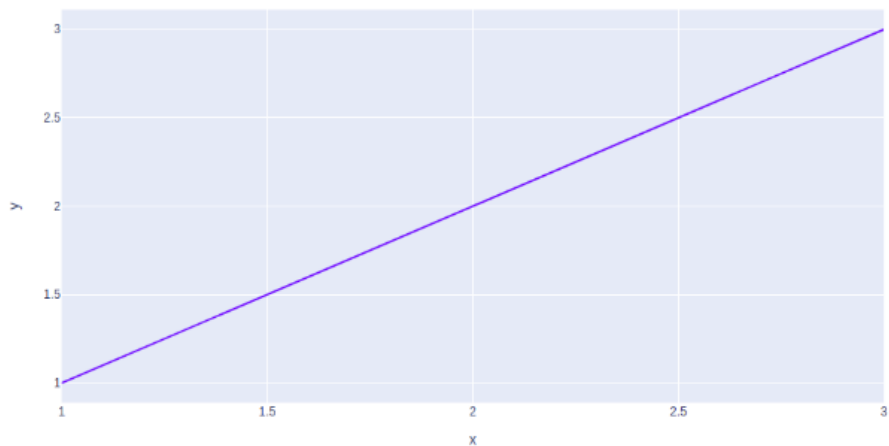
Output:



USING PLOTLY:

```
import plotly.express as px  
  
# Creating the Figure instance  
fig = px.line(x=[1,2, 3], y=[1, 2, 3])  
  
# printing the figure instance  
print(fig)
```

Output:



2. Column Chart :

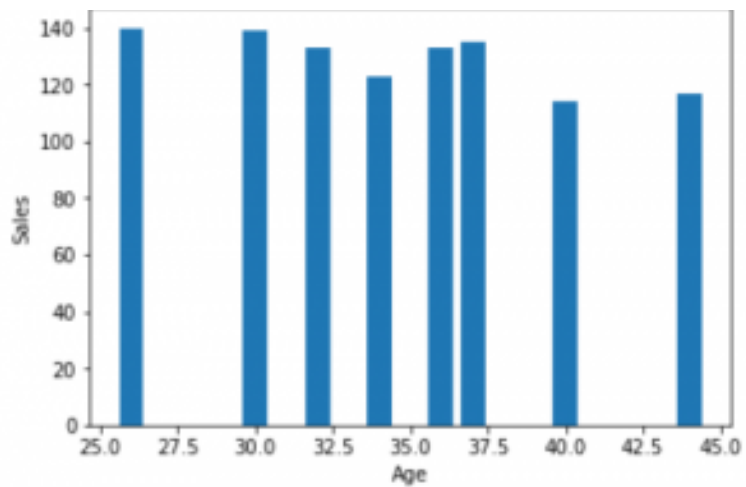
A column chart is used to show a comparison among different attributes, or it can show a comparison of items over time.

```
# Dataframe of previous code is used here

# Plot the bar chart for numeric values
# a comparison will be shown between
# all 3 age, income, sales
df.plot.bar()

# plot between 2 attributes
plt.bar(df['Age'], df['Sales'])
plt.xlabel("Age")
plt.ylabel("Sales")
plt.show()
```

OUTPUT:



README: Big Data Analysis Solution on IBM Cloud Databases

Table of Contents

- Introduction
- Prerequisites
- Deployment
- Data Analysis
- Website Navigation
- Updating Content
- Dependencies

Introduction

Provide a brief overview of the big data analysis solution, its purpose, and the technologies involved.

Prerequisites

List all the prerequisites and tools that users need to have before deploying and using your solution. Include:

- IBM Cloud account
- Access to IBM Cloud Databases
- Necessary data and datasets
- Web server environment, if applicable

Deployment

Explain step-by-step how to deploy the solution on IBM Cloud Databases. Include detailed instructions for setting up the database, configuring the server, and deploying your application. Mention any environment variables that need to be set.

Data Analysis

Describe how users can perform data analysis using your solution. Include example queries, data visualization, or any specific tools you've integrated for analysis.

Website Navigation

Provide a guide on how to navigate the website. Explain the menu structure, main features, and any interactive components. Use screenshots or diagrams if helpful.

Updating Content

Explain how users can update or modify the content on the website. Include instructions for adding new data, changing visuals, and updating text or multimedia elements.

Dependencies

the external dependencies, libraries, and APIs that your solution relies on. Provide installation instructions or links to relevant documentation for each dependency.

Certainly, prerequisites are the requirements and conditions that users must meet or have in place before they can successfully deploy and use your big data analysis solution on IBM Cloud Databases. These requirements are essential to ensure a smooth setup and operation of your solution. Here's a more detailed explanation of the typical prerequisites:

1. IBM Cloud Account: Users need an active IBM Cloud account. If they don't have one, they should sign up for an account at [IBM Cloud](<https://cloud.ibm.com/>).

2. Access to IBM Cloud Databases: Users should have access to IBM Cloud Databases, particularly the database service (e.g., Db2, PostgreSQL, or any other suitable option) that your solution relies on. Ensure that they have the necessary permissions and credentials to create and manage databases.

3. Necessary Data and Datasets: Depending on your specific big data analysis solution, users might need to have access to the relevant data or datasets that will be analyzed. Provide instructions on how to obtain or import these datasets into the database.

4. Web Server Environment: If your solution includes a web-based interface for data analysis, users might need to have a web server environment set up. This could include technologies like Node.js, Python with Flask or Django, or other relevant frameworks. Explain the requirements and how to set up this environment.

5. Internet Connection: A stable internet connection is necessary for accessing IBM Cloud services, downloading necessary dependencies, and for your web-based solution if applicable.

6. Operating System and Browser Compatibility: Mention any specific operating system or browser requirements that your solution might have. For example, if your web interface works best on Google Chrome or requires a particular operating system, let users know.

7. IBM Cloud CLI (Command Line Interface): If your deployment process involves using the IBM Cloud CLI, instruct users on how to install and configure it.

8. Database Client: Depending on the database used, users might need a compatible database client to interact with the database, execute queries, and manage data. Provide information on how to install and configure the necessary client tools.

9. Access Credentials and API Keys: Users should have access credentials, API keys, or other authentication methods required for accessing IBM Cloud services and the database. Instruct them on where to find or create these credentials and how to securely manage them.

10. Dependencies: List any software or libraries that your solution depends on, whether they are related to data analysis, web development, or other aspects of your project. Provide instructions for installing these dependencies.

By clearly outlining these prerequisites in your README, you help users prepare their environment and ensure that they have everything in place to successfully deploy and use your big data analysis solution.

Deployment refers to the process of taking your big data analysis solution from a development or testing environment and making it accessible and functional for users. In the context of deploying your solution on IBM Cloud Databases, here's a more detailed explanation of what deployment involves:

1. Internet Connection:

Create a Database**: Begin by setting up the database in IBM Cloud Databases. Depending on your solution, you might choose a relational database like Db2 or a NoSQL database like Cloudant. Create the necessary database instance.

2. Server Configuration:

Choose a Web Server Environment: If your solution includes a web-based interface for data analysis, select a suitable web server environment. Common choices include Node.js, Python with Flask or Django, or other frameworks.

Server Configuration: Configure the web server to run your application. This involves specifying the routes, handling HTTP requests, and connecting to the database.

3. Application Deployment:

Upload Code: Upload your solution's code to the server. This code should include the application logic, data analysis scripts, and any web interface files.

- **Install Dependencies**: Install any necessary dependencies and libraries. You might use a package manager like npm (for Node.js) or pip (for Python) to install required packages.

4. Database Connection:

Establish Database Connection: Configure your application to connect to the IBM Cloud Database. This typically involves providing the database's connection string or URL, along with the required credentials.

5. Environment Variables:

Set Environment Variables: Ensure that any sensitive information such as API keys or database credentials are stored as environment variables, not hard-coded in your application code. This enhances security and makes it easier to manage these settings.

6. Testing and Quality Assurance:

Test the Deployment: Before making your solution accessible to users, thoroughly test it to ensure it works as expected. Perform unit testing, integration testing, and test data analysis processes to catch and resolve any issues.

7. Scalability and Performance Optimization:

- Consider how your solution will handle increased user loads and data. Implement strategies for scalability and performance optimization, especially if your big data analysis solution is expected to handle large datasets.

8. Security and Access Control:

- Implement security measures to protect your database and application from unauthorized access and potential threats. Set up access control and user authentication as needed.

9. Documentation:

Create documentation that guides users on how to access and use your deployed solution. This should be part of your README file and should include clear instructions on how to navigate the website, perform data analysis, and update content.

10. Continuous Monitoring and Maintenance:

- Once your solution is deployed, it's important to continuously monitor its performance and ensure that it remains up-to-date. Regularly update dependencies, apply security patches, and respond to user feedback and issues.

11. Backup and Recovery:

- Implement backup and recovery procedures for your database to prevent data loss in case of unexpected failures or incidents.

Deployment is a crucial step in the lifecycle of your big data analysis solution, and it requires careful planning and execution to ensure that users can access and utilize your application effectively and securely.

I can provide you with a list of some common libraries and tools that are often used in big data analytics and their respective installation instructions or documentation links:

1. **Hadoop:**

- Documentation: [Hadoop Documentation](<https://hadoop.apache.org/docs/>)

2. **Spark:**

- Documentation: [Apache Spark Documentation](<https://spark.apache.org/docs/latest/>)

3. **Hive:**

- Documentation: [Hive Documentation](<https://cwiki.apache.org/confluence/display/Hive/Home>)

4. **HBase:**

- Documentation: [HBase Reference Guide](<https://hbase.apache.org/book.html>)

5. **Kafka:**

- Documentation: [Apache Kafka Documentation](<https://kafka.apache.org/documentation/>)

6. **Flink:**

- Documentation: [Apache Flink Documentation](<https://ci.apache.org/projects/flink/flink-docs-release-1.14/>)

7. **Cassandra:**

- Documentation: [Apache Cassandra Documentation](<https://cassandra.apache.org/doc/latest/>)

8. **Presto:**

- Documentation: [Presto Documentation](<https://prestodb.io/docs/current/>)

9. **Zeppelin:**

- Documentation: [Apache Zeppelin Documentation](<https://zeppelin.apache.org/docs/latest/>)

10. **Jupyter Notebook:**

- Documentation: [Jupyter Documentation](<https://jupyter.readthedocs.io/en/latest/>)

Conclusion:

The conclusion step of big data analytics with IBM cloud databases typically involves summarizing the key findings and insights obtained from the analysis. This step may include:

1. **Data Insights:** Present the most significant findings and patterns discovered during the analysis, including any trends, anomalies, or correlations in the data.
2. **Recommendations:** Provide actionable recommendations based on the insights to help inform decision-making and strategy. These recommendations can be aimed at improving processes, increasing efficiency, or making data-driven business decisions.
3. **Visualization:** Utilize data visualization tools to create compelling charts, graphs, and reports that communicate the results effectively to stakeholders.
4. **Documentation:** Ensure that all steps of the analysis process are documented comprehensively. This includes the data sources, tools used, methods applied, and any assumptions made.
5. **Future Steps:** Consider what further analysis or actions may be required to build upon the current findings and continually improve data-driven decision-making.
6. **Communicate Results:** Share the findings and recommendations with relevant stakeholders, such as management, team members, or clients, and engage in discussions to plan the next steps.