Date	25/10/2023
Project name	BIG DATA ANALYSIS WITH IBM CLOUD DATABASE

Phase 4

BIG DATA ANALYSIS WITH IBM CLOUD DATABASES

1. **IBM Cloud Account Setup:**
   - If you don't have one, create an IBM Cloud account.
   - Subscribe to the necessary services like IBM Cloudant, IBM Cloud Object Storage, IBM Cloud Data Engine, and IBM Watson Studio.

2. **Data Ingestion:**
   - Ingest data from various sources (e.g., sensors, social media, customer databases).
   - Implement data pipelines and connectors to extract data into a format suitable for analysis.
   - Consider using IBM Cloud Data Integration for data ingestion and transformation tasks.

3. **Data Storage:**
   - Set up IBM Cloud Object Storage for storing ingested data.
   - Organize data into buckets and define access controls.
   - Create a data storage strategy that considers data retention and archiving requirements.

4. **Data Processing:**
   - Utilize IBM Cloud Data Engine (perhaps on Apache Spark) to process and analyze the stored data.
   - Develop data processing pipelines and scripts for extracting valuable insights.
   - Optimize the data processing workflows for efficiency and scalability.

5. **Machine Learning:**
   - Use IBM Watson Studio to build, train, and deploy machine learning models.
   - Select and implement machine learning algorithms for predictive analytics and pattern recognition. • Monitor model performance and retrain models as needed.

6. **Data Visualization:**
   - Integrate data visualization tools like Tableau or Power BI.
   - Create interactive and informative dashboards to visualize insights.
   - Ensure that visualization tools can connect to data stored in IBM Cloud databases.

7. **Infrastructure Setup:**
   - Configure and optimize IBM Cloud Foundry environments for hosting applications, if needed.
   - Ensure that the infrastructure can handle the demands of data processing and visualization.

8. **Data Governance and Compliance:**
   - Implement data governance policies and access controls to ensure data security and compliance.
   - Document data lineage and data usage for auditing and compliance purposes.

9. Scalability and Performance:

- Monitor the performance of your Cloud Foundry applications and data processing pipelines.
- Implement auto-scaling and load balancing as needed to handle fluctuations in data volume.

DATASET **https://www.dofactory.com/sql/download-sample-database**

SOURCE CODE

```python
import nltk
from nltk.sentiment import SentimentIntensityAnalyzer
from cloudant.client import Cloudant
import matplotlib.pyplot as plt

# Initialize NLTK sentiment analysis
nltk.download('vader_lexicon')
sia = SentimentIntensityAnalyzer()

# Initialize Cloudant client
cloudant_credentials = {
    'username': 'apikey-v2-2lnp8ems0zddh4alc6h2ryg5fsktjwd4634s3t4n3pf0',
    'password': '5c3eb941b4ecf0e7e16f2689eba75648',
    'host': '97192523-7f6a-4f64-a77d-737285ff455f-
bluemix.cloudantnosqldb.appdomain.cloud',
    'port': '443',
    'url': 'https://' + '97192523-7f6a-4f64-a77d-737285ff455f-
bluemix.cloudantnosqldb.appdomain.cloud'
}
cloudant_client = Cloudant(cloudant_credentials['username'],
cloudant_credentials['password'],
                        url=cloudant_credentials['url'], connect=True)

# Function to retrieve customer reviews from Cloudant
def get_customer_reviews():
    database = cloudant_client['your_customer_reviews_database_name']
    return [doc for doc in database]

# Function to perform sentiment analysis on customer reviews
def analyze_sentiment(reviews):
    sentiment_scores = []
    for review in reviews:
        sentiment = sia.polarity_scores(review['text'])
```

```python
            sentiment_scores.append((review['date'], sentiment['compound']))
    return sentiment_scores

# Function to create and display the sentiment analysis line chart
def create_sentiment_line_chart(dates, scores):
    plt.figure(figsize=(10, 6))
    plt.plot(dates, scores, marker='o')
    plt.title('Customer Review Sentiment Analysis Over Time')
    plt.xlabel('Date')
    plt.ylabel('Sentiment Score')
    plt.xticks(rotation=45)
    plt.grid(True)
    plt.tight_layout()
    plt.show()

# Example usage
if __name__ == '__main__':
    # Get customer reviews from Cloudant (simulated data)
    customer_reviews = [
        {'date': '2021-01-01', 'text': "Great product and excellent
service."},
        {'date': '2021-02-01', 'text': "Disappointed with the product
quality."},
        {'date': '2021-03-01', 'text': "Outstanding customer support!"},
        {'date': '2021-04-01', 'text': "The product met my expectations."},
        {'date': '2021-05-01', 'text': "Very poor service experience."},
    ]

    # Analyze sentiment of customer reviews
    sentiment_scores = analyze_sentiment(customer_reviews)

    # Separate the dates and sentiment scores for the line chart
    dates, scores = zip(*sentiment_scores)

    # Create and display the sentiment analysis line chart
    create_sentiment_line_chart(dates, scores)
```

Conclusion

IBM Cloud databases offer a variety of services that can be used to build a complete big data analysis solution. By combining these services with other tools, such as data visualization tools and machine learning tools, businesses can gain valuable insights from their data.