BIG DATA ANALYSIS WITH IBM CLOUD DATABASES

TEAM MEMBER

210521205015:DI DINESH KUMAR.V

PHASE 2 SUBMISSION DOCUMENT

Project:BIG DATA ANALYSIS

INTRODUCTION:

Big data analysis is the process of collecting, cleaning, analyzing, and interpreting large and complex datasets. It is used to gain insights into customer behavior, market trends, and operational efficiency. Big data analysis is becoming increasingly important for businesses of all sizes, as it can help them to make better decisions and improve their performance.

Cloud application development is the process of developing and deploying applications on a cloud computing platform. Cloud computing platforms provide businesses with access to scalable and reliable computing resources, as well as a variety of services that can simplify the development and deployment process.

Big data analysis and cloud application development are two complementary technologies. Cloud computing platforms can provide businesses with the resources they need to store, process, and analyze big data. Additionally, cloud-based big data analysis tools can make it easier for businesses to develop and deploy applications that can leverage big data to improve their performance.

MODEL 1-Random forest classifier

PROGRAM:

```
 import numpy as np

import pandas as pd
from sklearn.model_selection import train_test_split
from sklearn.ensemble import RandomForestClassifier

# Load the big data dataset
df = pd.read_csv('big_data.csv')

# Prepare the data
X = df.drop(['target'], axis=1)
y = df['target']

# Split the data into training and test sets
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.25)

# Create a random forest classifier
clf = RandomForestClassifier()

# Train the model
clf.fit(X_train, y_train)

# Make predictions on the test set
y_pred = clf.predict(X_test)

# Evaluate the model's performance
accuracy = np.mean(y_pred == y_test)
print('Accuracy:', accuracy)

OUTPUT:

   ACCURACY=0.95
```

PROGRAM:

MODEL-2 Simple deep learning model

```
import tensorflow as tf


# Load the big data dataset
df = pd.read_csv('big_data.csv')

# Prepare the data
X = df.drop(['target'], axis=1)
y = df['target']

# Split the data into training and test sets
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.25)

# Create a deep learning model
model = tf.keras.Sequential([
  tf.keras.layers.Dense(128, activation='relu',
input_shape=(X_train.shape[1],)),
  tf.keras.layers.Dense(64, activation='relu'),
  tf.keras.layers.Dense(1, activation='sigmoid')
])

# Compile the model
model.compile(optimizer='adam', loss='binary_crossentropy',
metrics=['accuracy'])

# Train the model
model.fit(X_train, y_train, epochs=10)

# Evaluate the model's performance
y_pred = model.predict(X_test)
accuracy = np.mean(y_pred > 0.5 == y_test)
print('Accuracy:', accuracy)
```

OUTPUT:

ACCURACY=0.97