Cloud computing and bigdata

Sample final exam questions

Question 1: Multiple choice questions:

1. Which of the following is an example of Quasi-structured data?

A. A relational database table

B. A JSON log file from a web server

C. An Excel sheet with strict column definitions

D. A PNG image file

2. Which sentence is true about containerizations?

A. Containers require a full guest operating system per application.

B. Containerization offers less portability than virtual machines.

C. Containers share the host OS kernel, making them lightweight and efficient.

D. Containerization is only possible on Linux-based systems.

---

3. ....is an example of SaaS cloud model

A. Dropbox

B. Microsoft OneDrive

C. Google docs

D. All the mentioned answers

---

4. ....is true about the Hypervisor

A. It runs only on cloud-native applications.

B. It allows multiple virtual machines to run on a single physical machine.

C. It is only used in desktop operating systems.

D. Hypervisors are obsolete due to containerization.

Solutions:

1:B

2:C

3:D

4:B

Question 2: Code examples

Given the below Colab data analysis example, then answer the following questions:

```
# Import required libraries:
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
from sklearn import linear_model
# Read the CSV file :
data = pd.read_csv("https://s3-api.us-geo.objectstorage.softlayer.net/cf-courses-data/CognitiveClass/ML0101ENv3/labs/FuelConsumptionCo2.csv")
data.head()
# Let's select some features to explore more :
data = data[["ENGINESIZE","CO2EMISSIONS"]]
# ENGINESIZE vs CO2EMISSIONS:
plt.scatter(data["ENGINESIZE"] , data["CO2EMISSIONS"] , color="blue")
plt.xlabel("ENGINESIZE")
plt.ylabel("CO2EMISSIONS")
plt.show()
# Generating training and testing data from our data:
# We are using 80% data for training.
train = data[:(int((len(data)*0.8)))]
test = data[(int((len(data)*0.8))):]
# Modeling:
# Using sklearn package to model data :
regr = linear_model.LinearRegression()
train_x = np.array(train[["ENGINESIZE"]])
train_y = np.array(train[["CO2EMISSIONS"]])
regr.fit(train_x,train_y)
# The coefficients:
print ("coefficients : ",regr.coef_) #Slope
print ("Intercept : ",regr.intercept_) #Intercept
# Plotting the regression line:
plt.scatter(train["ENGINESIZE"], train["CO2EMISSIONS"], color='blue')
plt.plot(train_x, regr.coef_*train_x + regr.intercept_, '-r')
plt.xlabel("Engine size")
plt.ylabel("Emission")
# Predicting values:
# Function for predicting future values :
def get_regression_predictions(input_features,intercept,slope):
 predicted_values = input_features*slope + intercept
 return predicted_values
# Predicting emission for future car:
my_engine_size = 3.5

```

```
43   estimatd_emission =
44   get_regression_predictions(my_engine_size,regr.intercept_[0],regr.coef_[0][0])
45   print ("Estimated Emission :",estimatd_emission)
46   # Checking various accuracy:
47   from sklearn.metrics import r2_score
48   test_x = np.array(test[['ENGINESIZE']])
49   test_y = np.array(test[['CO2EMISSIONS']])
50   test_y_ = regr.predict(test_x)
51   print("Mean absolute error: %.2f" % np.mean(np.absolute(test_y_ - test_y)))
52   print("Mean sum of squares (MSE): %.2f" % np.mean((test_y_ - test_y) ** 2))
     print("R2-score: %.2f" % r2_score(test_y_ , test_y) )
```

A. What does the above code do?

Answer: it applies the linear regression model to model the relationship between the 'ENGINESIZE' and the 'CO2EMISSIONS'.

B. What is the line of code used to apply the linear regression model?

Answer : line 23

C. What is the value of the Engine used as an example to apply the prediction anslysis?

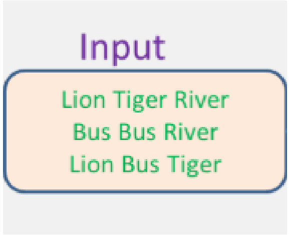
Answswer 3.5 (see line 41)

D. What are the three metrics used to evaluate the model accuracy?

Answer: Mean absolute error, Mean sum of squares (MSE and the R2-score. (See line 50-52)

Question 3: MapReduce example

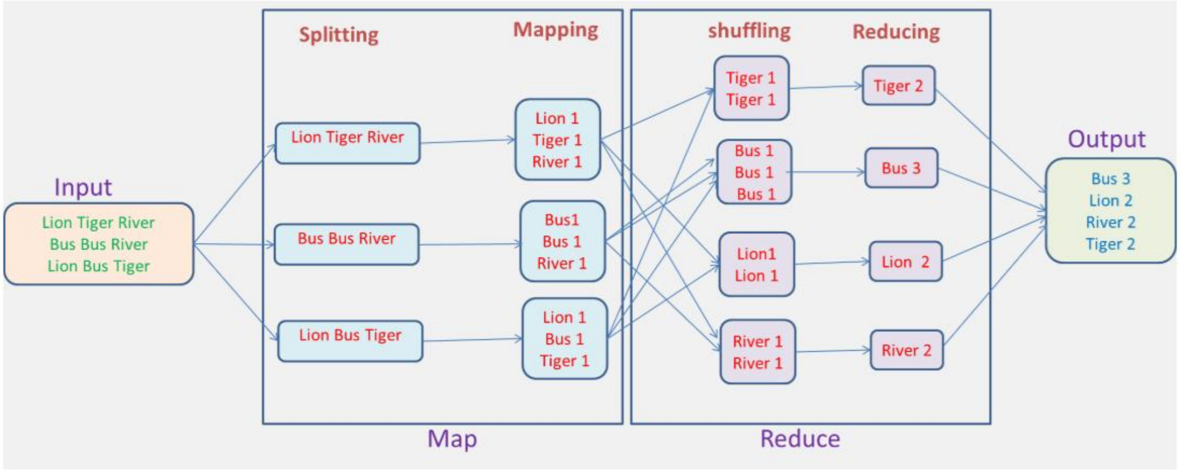
Show the MapReuce algorithm steps needed to find out the number of occurrence of each word in the below input:



Input

Lion Tiger River
Bus Bus River
Lion Bus Tiger

Solution


Input
Lion Tiger River
Bus Bus River
Lion Bus Tiger
Splitting
Mapping
Lion Tiger River
Bus Bus River
Lion Bus Tiger
Lion 1
Tiger 1
River 1
Bus1
Bus 1
River 1
Lion 1
Bus 1
Tiger 1
shuffling
Reducing
Tiger 1
Tiger 1
Bus 1
Bus 1
Bus 1
Lion1
Lion 1
River 1
River 1
Tiger 2
Bus 3
Lion 2
River 2
Output
Bus 3
Lion 2
River 2
Tiger 2
Map
Reduce