Analyse search terms on the e-commerce web server

In this assignment you will download the search term data set for the e-commerce web server and run analytic queries on it.

Install spark

In [66]:
```
!pip install pyspark
!pip install findspark
```

```
Requirement already satisfied: pyspark in /home/jupyterlab/conda/envs/python/lib/python3.7/site-packages (3.4.1)
Requirement already satisfied: py4j==0.10.9.7 in /home/jupyterlab/conda/envs/python/lib/python3.7/site-packages (from pyspark) (0.10.9.7)
Requirement already satisfied: findspark in /home/jupyterlab/conda/envs/python/lib/python3.7/site-packages (2.0.1)
```

import libraries

In [67]:
```
import findspark
findspark.init()

from pyspark import SparkContext, SparkConf
from pyspark.sql import SparkSession
```

Start session

In [68]:
```
# creating a SparkContext class
sc = SparkContext()

# creating SparkSession
Spark = SparkSession \
        .builder \
        .appName("Analyzing search terms on the e-commerce web server").getOrCreate()
```

```
23/08/10 18:40:59 WARN util.Utils: Service 'SparkUI' could not bind on port 4040. Attempting port 4041.
```

Importing SparkML libraries

In [69]:
```
from pyspark.ml.feature import VectorAssembler
from pyspark.ml.regression import LinearRegression
```

In [70]:
```
# Download The search term dataset from the below url
# https://cf-courses-data.s3.us.cloud-object-storage.appdomain.cloud/IBM-DB0321EN-SkillsNetwork/Bigdata%20and%20Spark/searchterms.csv
```

In [71]:
```
!wget https://cf-courses-data.s3.us.cloud-object-storage.appdomain.cloud/IBM-DB0321EN-SkillsNetwork/Bigdata%20and%20Spark/searchterms.csv
```

```
--2023-08-10 18:41:00--  https://cf-courses-data.s3.us.cloud-object-storage.appdomain.cloud/IBM-DB0321EN-SkillsNetwork/Bigdata%20and%20Spark/searchterms.csv
Resolving cf-courses-data.s3.us.cloud-object-storage.appdomain.cloud (cf-courses-data.s3.us.cloud-object-storage.appdomain.cloud)... 169.63.118.104
Connecting to cf-courses-data.s3.us.cloud-object-storage.appdomain.cloud (cf-courses-data.s3.us.cloud-object-storage.appdomain.cloud)|169.63.118.104|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 233457 (228K) [text/csv]
Saving to: ‘searchterms.csv.1’

searchterms.csv.1   100%[===================>] 227.99K  --.-KB/s    in 0.004s

2023-08-10 18:41:00 (58.0 MB/s) - ‘searchterms.csv.1’ saved [233457/233457]
```

In [72]:
```
Spark
```

Out[72]:

SparkSession - in-memory

SparkContext

Spark UI

Version	`v2.4.3`
Master	`local[*]`
AppName	`pyspark-shell`

In [73]:
```
# Load the csv into a spark dataframe
```

In [74]:
```
sdf = Spark.read.csv("searchterms.csv",header = True, inferSchema = True)
```

In [75]:
```
# Print the number of rows and columns
# Take a screenshot of the code and name it as shape.jpg)
```

In [76]:
```
num_rows = sdf.count()
num_cols = len(sdf.columns)

(num_rows,num_cols)
```

Out[76]: (10000, 4)

In [77]:
```
print(' Rows:%d,Columns:%d ' %(num_rows,num_cols))
```

```
Rows:10000,Columns:4
```

In [78]:
```
# Print the top 5 rows
# Take a screenshot of the code and name it as top5rows.jpg)
```

In [79]:
```
sdf.head(5)
```

Out[79]:
```
[Row(day=12, month=11, year=2021, searchterm='mobile 6 inch'),
 Row(day=12, month=11, year=2021, searchterm='mobile latest'),
 Row(day=12, month=11, year=2021, searchterm='tablet wifi'),
 Row(day=12, month=11, year=2021, searchterm='laptop 14 inch'),
 Row(day=12, month=11, year=2021, searchterm='mobile 5g')]
```

In [80]:
```
# Find out the datatype of the column searchterm?
# Take a screenshot of the code and name it as datatype.jpg)
```

In [81]:
```
sdf.dtypes[3]
```

Out[81]: ('searchterm', 'string')

In [82]:
```
sdf.dtypes[-1]
```

Out[82]: ('searchterm', 'string')

In [83]:
```
# How many times was the term `gaming laptop` searched?
# Take a screenshot of the code and name it as gaminglaptop.jpg)
```

In [84]:
```
sdf.createOrReplaceTempView('sdf')
```

In [85]:
```
Spark.sql("SELECT COUNT(*) AS search_times FROM sdf WHERE searchterm = 'gaming laptop' ").show()
```

```
+------------+
|search_times|
+------------+
|         499|
+------------+
```

In [86]:
```
# Print the top 5 most frequently used search terms?
# Take a screenshot of the code and name it as top5terms.jpg)
```

In [87]:
```
Spark.sql(" SELECT searchterm,COUNT(*) AS search_times FROM sdf GROUP BY searchterm ORDER BY search_times DESC LIMIT 5 ").show()
```

```
[Stage 8:=========================================>            (155 + 12) / 200]
+-------------+------------+
|   searchterm|search_times|
+-------------+------------+
|mobile 6 inch|        2312|
|    mobile 5g|        2301|
|mobile latest|        1327|
|       laptop|         935|
|  tablet wifi|         896|
+-------------+------------+
```

In [88]:
```
# The pretrained sales forecasting model is available at  the below url
# https://cf-courses-data.s3.us.cloud-object-storage.appdomain.cloud/IBM-DB0321EN-SkillsNetwork/Bigdata%20and%20Spark/model.tar.gz
```

In [89]:
```
!wget https://cf-courses-data.s3.us.cloud-object-storage.appdomain.cloud/IBM-DB0321EN-SkillsNetwork/Bigdata%20and%20Spark/model.tar.gz
```

```
--2023-08-10 18:41:05--  https://cf-courses-data.s3.us.cloud-object-storage.appdomain.cloud/IBM-DB0321EN-SkillsNetwork/Bigdata%20and%20Spark/model.tar.gz
Resolving cf-courses-data.s3.us.cloud-object-storage.appdomain.cloud (cf-courses-data.s3.us.cloud-object-storage.appdomain.cloud)... 169.63.118.104
Connecting to cf-courses-data.s3.us.cloud-object-storage.appdomain.cloud (cf-courses-data.s3.us.cloud-object-storage.appdomain.cloud)|169.63.118.104|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 1490 (1.5K) [application/x-tar]
Saving to: ‘model.tar.gz.1’

model.tar.gz.1      100%[===================>]   1.46K  --.-KB/s    in 0s

2023-08-10 18:41:05 (12.5 MB/s) - ‘model.tar.gz.1’ saved [1490/1490]
```

In [100…
```
!tar -xvf model.tar.gz.1 -C /resources/labs/DB0321EN/model
# !tar = to extract ter file
# -x: Indicates that you want to extract files from the archive.
# -v: Enables verbose mode, which shows the progress and details of the extraction process.
# -f: Specifies that the following argument is the name of the archive file to be operated on. In this case, it's model.tar.gz.
# -C: This specifies the directory where you want to extract the contents of the archive.
```

```
sales_prediction.model/
sales_prediction.model/metadata/
sales_prediction.model/metadata/part-00000
sales_prediction.model/metadata/.part-00000.crc
sales_prediction.model/metadata/_SUCCESS
sales_prediction.model/metadata/._SUCCESS.crc
sales_prediction.model/data/
sales_prediction.model/data/part-00000-1db9fe2f-4d93-4b1f-966b-3b09e72d664e-c000.snappy.parquet
sales_prediction.model/data/_SUCCESS
sales_prediction.model/data/.part-00000-1db9fe2f-4d93-4b1f-966b-3b09e72d664e-c000.snappy.parquet.crc
sales_prediction.model/data/._SUCCESS.crc
```

In [91]:
```
# Load the sales forecast model.
# Take a screenshot of the code and name it as loadmodel.jpg)
```

In [102…
```
from pyspark.ml.regression import LinearRegressionModel
```

In [103…
```
model = LinearRegressionModel.load('model/sales_prediction.model')
```

In [104…
```
model.featuresCol
```

Out[104]:
```
Param(parent='LinearRegression_6d5736f3dbe7', name='featuresCol', doc='features column name')
```

In [105…
```
model.params
```

Out[105]:
```
[Param(parent='LinearRegression_6d5736f3dbe7', name='aggregationDepth', doc='suggested depth for treeAggregate (>= 2)'),
 Param(parent='LinearRegression_6d5736f3dbe7', name='elasticNetParam', doc='the ElasticNet mixing parameter, in range [0, 1]. For alpha = 0, the penalty is an L2 penalty. For alpha = 1, it is an
L1 penalty'),
 Param(parent='LinearRegression_6d5736f3dbe7', name='epsilon', doc='The shape parameter to control the amount of robustness. Must be > 1.0.'),
 Param(parent='LinearRegression_6d5736f3dbe7', name='featuresCol', doc='features column name'),
 Param(parent='LinearRegression_6d5736f3dbe7', name='fitIntercept', doc='whether to fit an intercept term'),
 Param(parent='LinearRegression_6d5736f3dbe7', name='labelCol', doc='label column name'),
 Param(parent='LinearRegression_6d5736f3dbe7', name='loss', doc='The loss function to be optimized. Supported options: squaredError, huber. (Default squaredError)'),
 Param(parent='LinearRegression_6d5736f3dbe7', name='maxIter', doc='maximum number of iterations (>= 0)'),
 Param(parent='LinearRegression_6d5736f3dbe7', name='predictionCol', doc='prediction column name'),
 Param(parent='LinearRegression_6d5736f3dbe7', name='regParam', doc='regularization parameter (>= 0)'),
 Param(parent='LinearRegression_6d5736f3dbe7', name='solver', doc='The solver algorithm for optimization. Supported options: auto, normal, l-bfgs. (Default auto)'),
 Param(parent='LinearRegression_6d5736f3dbe7', name='standardization', doc='whether to standardize the training features before fitting the model'),
 Param(parent='LinearRegression_6d5736f3dbe7', name='tol', doc='the convergence tolerance for iterative algorithms (>= 0)'),
 Param(parent='LinearRegression_6d5736f3dbe7', name='weightCol', doc='weight column name. If this is not set or empty, we treat all instance weights as 1.0')]
```

In [92]:
```
# Using the sales forecast model, predict the sales for the year of 2023.
# Take a screenshot of the code and name it as forecast.jpg
```

In [110…
```
sdf1 = Spark.read.parquet('model/sales_prediction.model/data/part-00000-1db9fe2f-4d93-4b1f-966b-3b09e72d664e-c000.snappy.parquet')
```

In [111…
```
sdf1.printSchema
```

Out[111]:
```
<bound method DataFrame.printSchema of DataFrame[intercept: double, coefficients: vector, scale: double]>
```

In [113…
```
sdf1.head()
```

Out[113]:
```
Row(intercept=-13019.989140447298, coefficients=DenseVector([6.5226]), scale=1.0)
```

In [116…
```
# This function converts a scalar number into a dataframe that can be used by the model to predict.
def predict(year):
    assembler = VectorAssembler(inputCols=["year"],outputCol="features")
    data = [[year,0]]
    columns = ["year", "sales"]
    _ = Spark.createDataFrame(data, columns)
    __ = assembler.transform(_).select('features','sales')
    predictions = model.transform(__)
    predictions.select('prediction').show()
```

In [117…
```
predict(2023)
```

```
+------------------+
|        prediction|
+------------------+
|175.16564294006457|
+------------------+
```

```
23/08/10 19:30:29 WARN netlib.BLAS: Failed to load implementation from: com.github.fommil.netlib.NativeSystemBLAS
23/08/10 19:30:29 WARN netlib.BLAS: Failed to load implementation from: com.github.fommil.netlib.NativeRefBLAS
```