STUDENT 1: 11355 | ASAD TARIQ SHEIKH

STUDENT 2: 10718 | MUZAMMIL AHMED

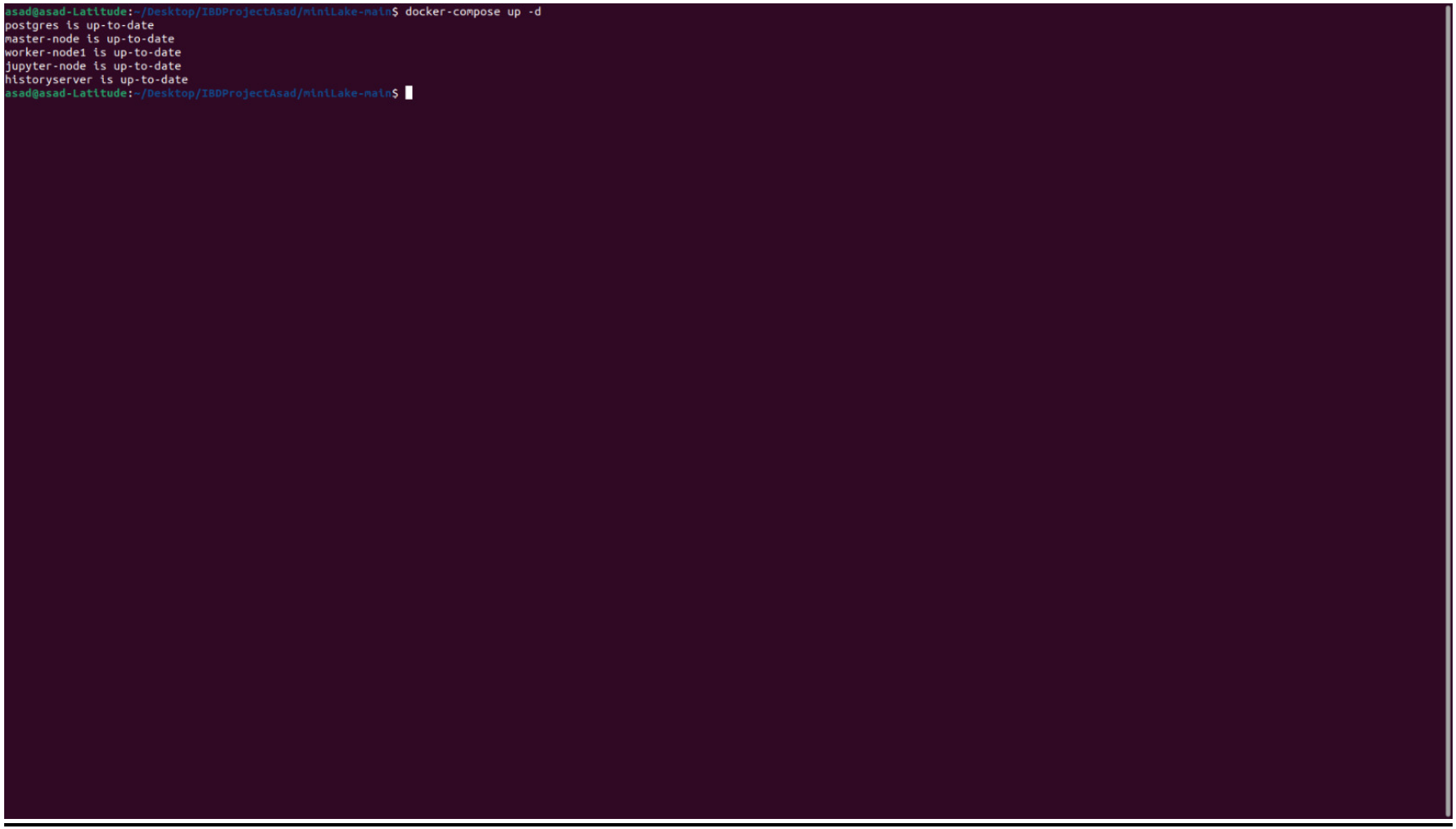
CLASS ID: 114195

COURSE: INTRODUCTION TO BIG DATA

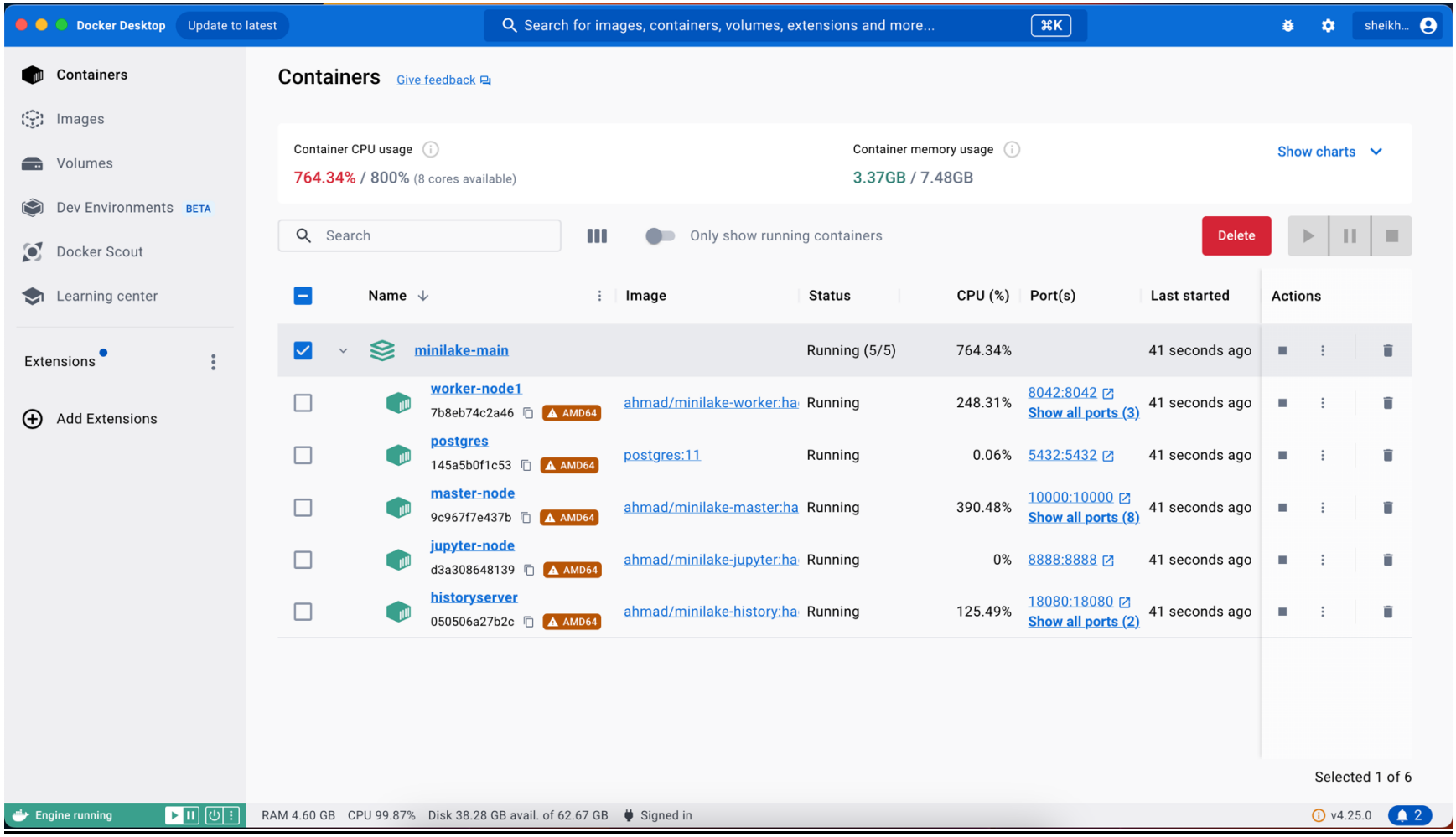
INSTRUCTOR: MUHAMMAD AHMAD

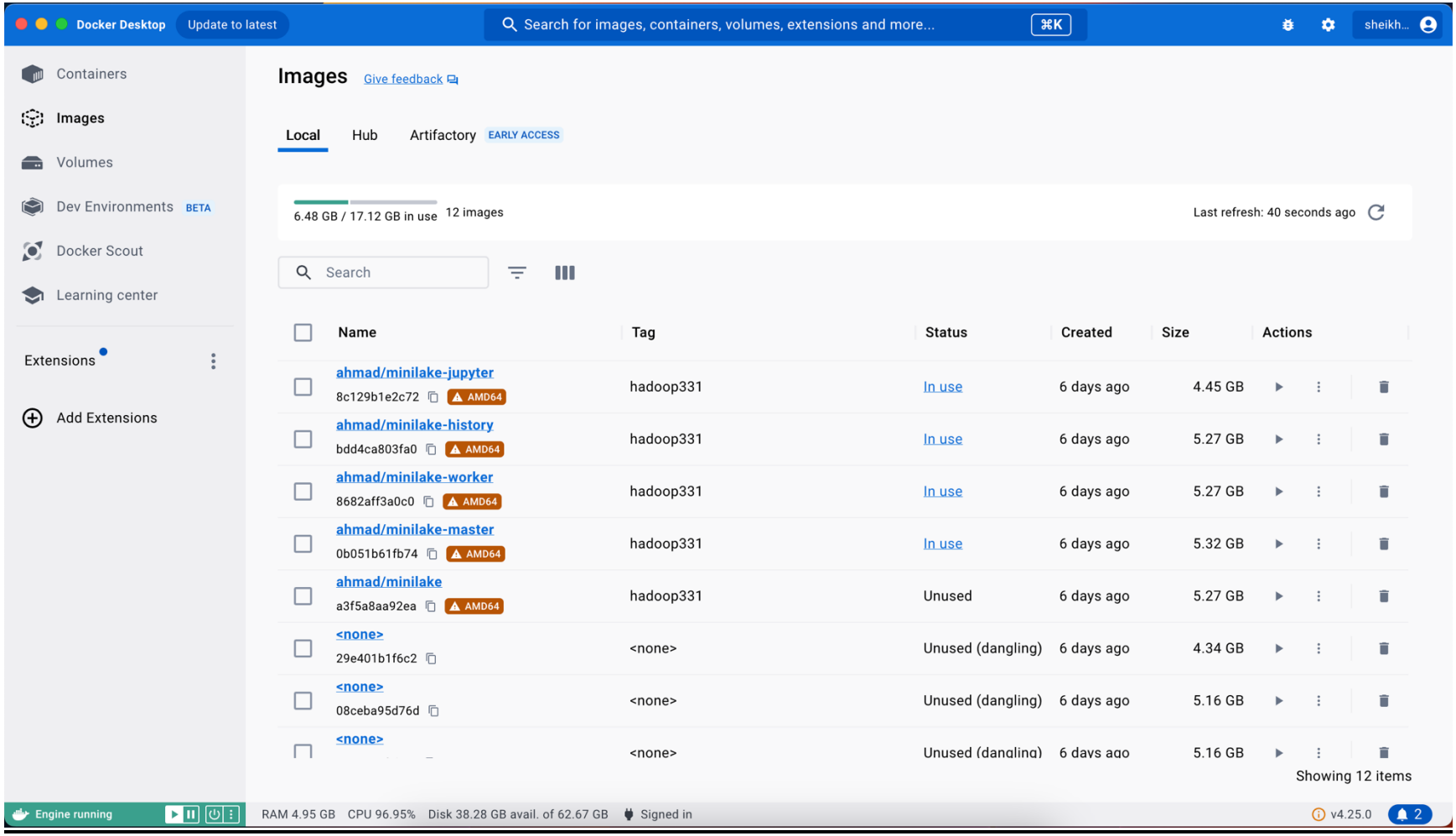
CRYPTOCURRENCY PROJECT

STEP 1: TERMINAL (docker-compose)



STEP 2: DOCKER (All the images running)




Docker Desktop
Update to latest
Search for images, containers, volumes, extensions and more...
⌘K
sheikh...
Containers
Images
Volumes
Dev Environments BETA
Docker Scout
Learning center
Extensions
Add Extensions
Images
Give feedback
Local
Hub
Artifactory EARLY ACCESS
6.48 GB / 17.12 GB in use
12 images
Last refresh: 40 seconds ago
Search
Name
Tag
Status
Created
Size
Actions
ahmad/minilake-jupyter
8c129b1e2c72
AMD64
hadoop331
In use
6 days ago
4.45 GB
ahmad/minilake-history
bdd4ca803fa0
AMD64
hadoop331
In use
6 days ago
5.27 GB
ahmad/minilake-worker
8682aff3a0c0
AMD64
hadoop331
In use
6 days ago
5.27 GB
ahmad/minilake-master
0b051b61fb74
AMD64
hadoop331
In use
6 days ago
5.32 GB
ahmad/minilake
a3f5a8aa92ea
AMD64
hadoop331
Unused
6 days ago
5.27 GB
<none>
29e401b1f6c2
<none>
Unused (dangling)
6 days ago
4.34 GB
<none>
08ceba95d76d
<none>
Unused (dangling)
6 days ago
5.16 GB
<none>
<none>
Unused (dangling)
6 days ago
5.16 GB
Showing 12 items
Engine running
RAM 4.95 GB
CPU 96.95%
Disk 38.28 GB avail. of 62.67 GB
Signed in
v4.25.0
2

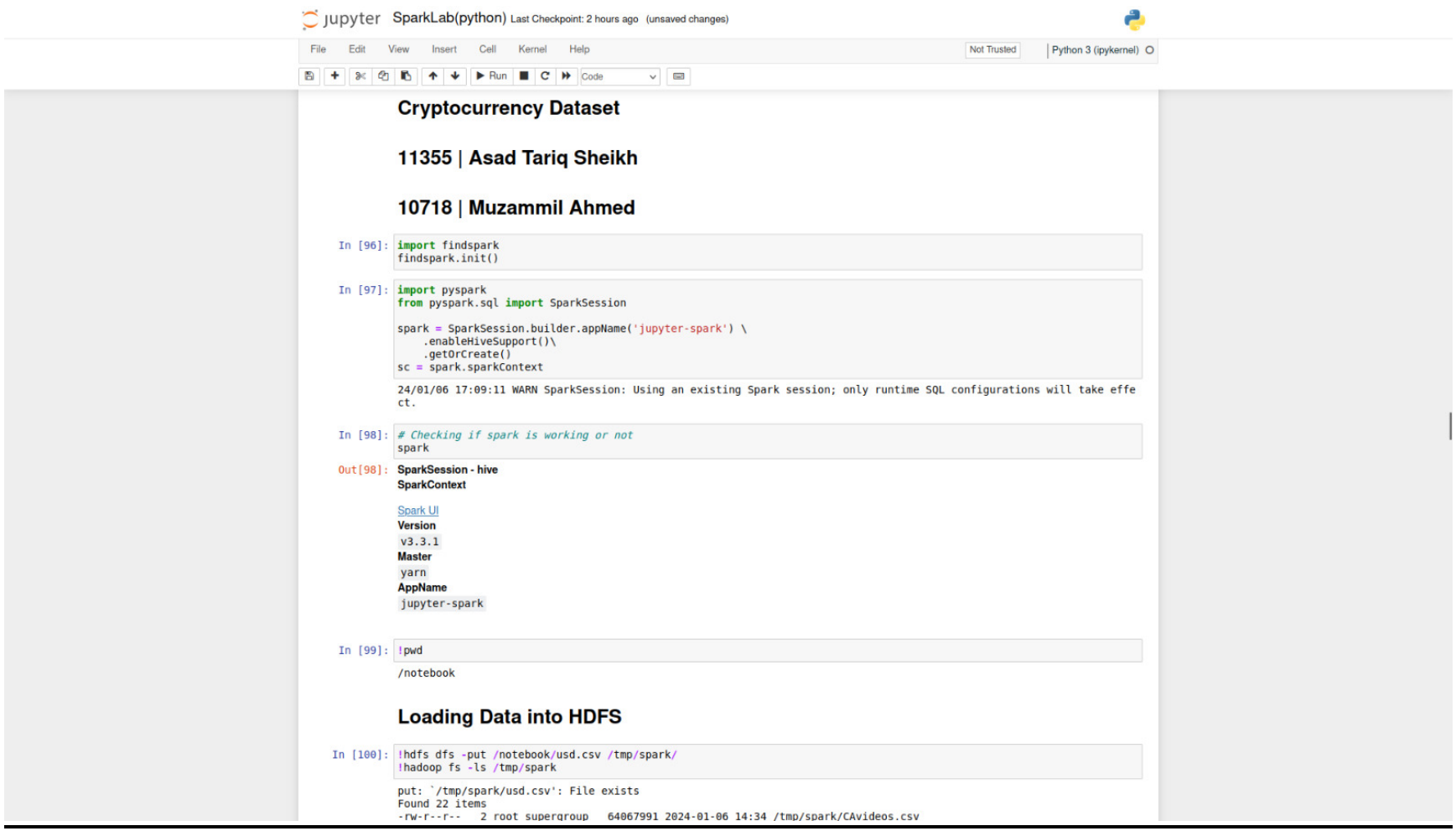

jupyter
Quit
Files
Running
Clusters
Select items to perform actions on them.
Upload
New
0
/
Name
Last Modified
File size
pyspark.ipynb
7 days ago
7.34 kB
SparkLab(python).ipynb
a day ago
46.2 kB
Untitled.ipynb
7 days ago
12.4 kB
bitcoin.csv
seconds ago
206 kB
usd.csv
seconds ago
206 kB

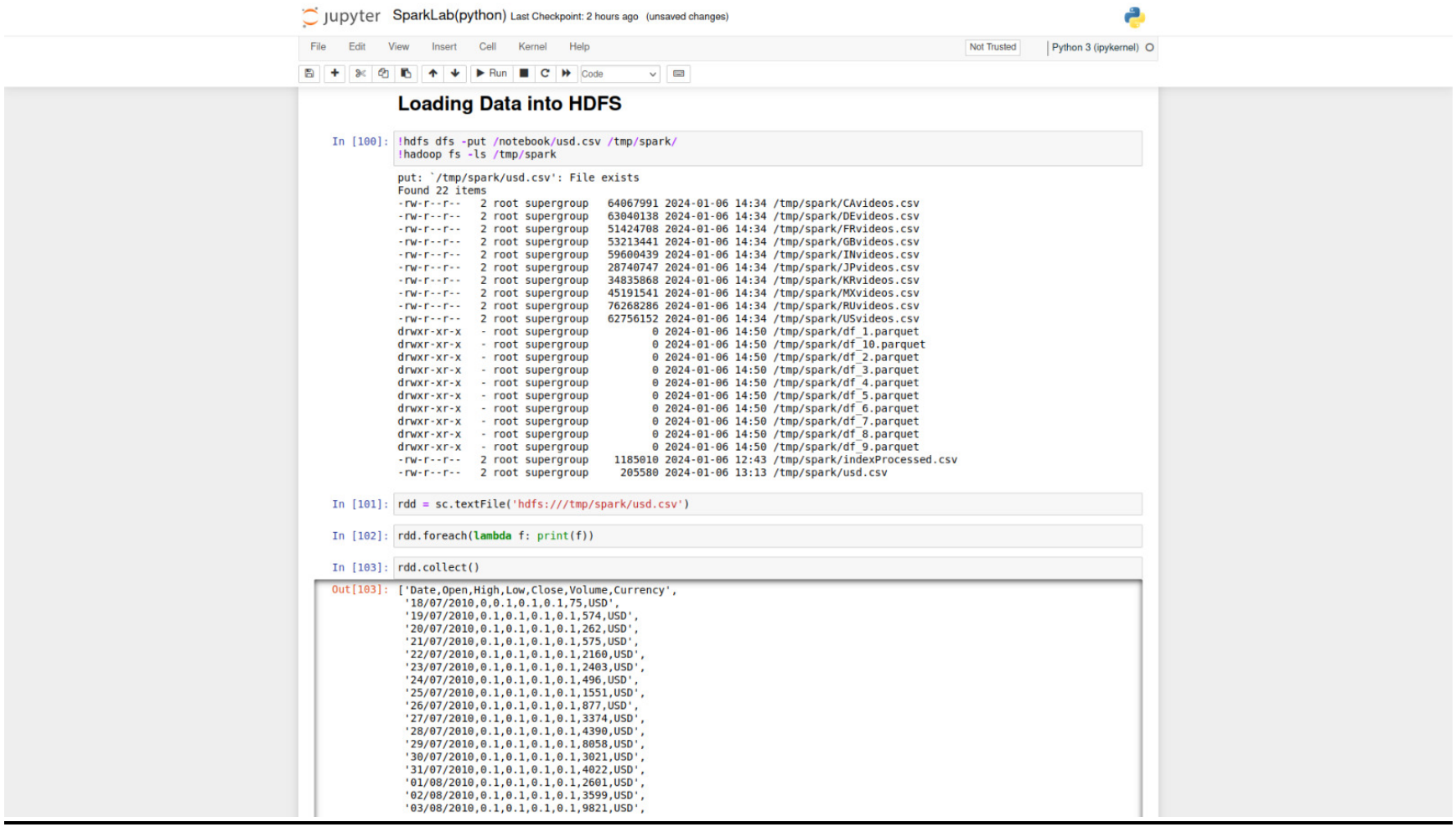
STEP 3: HIVE: CREATE TABLE `CRYPTO`

jupyter

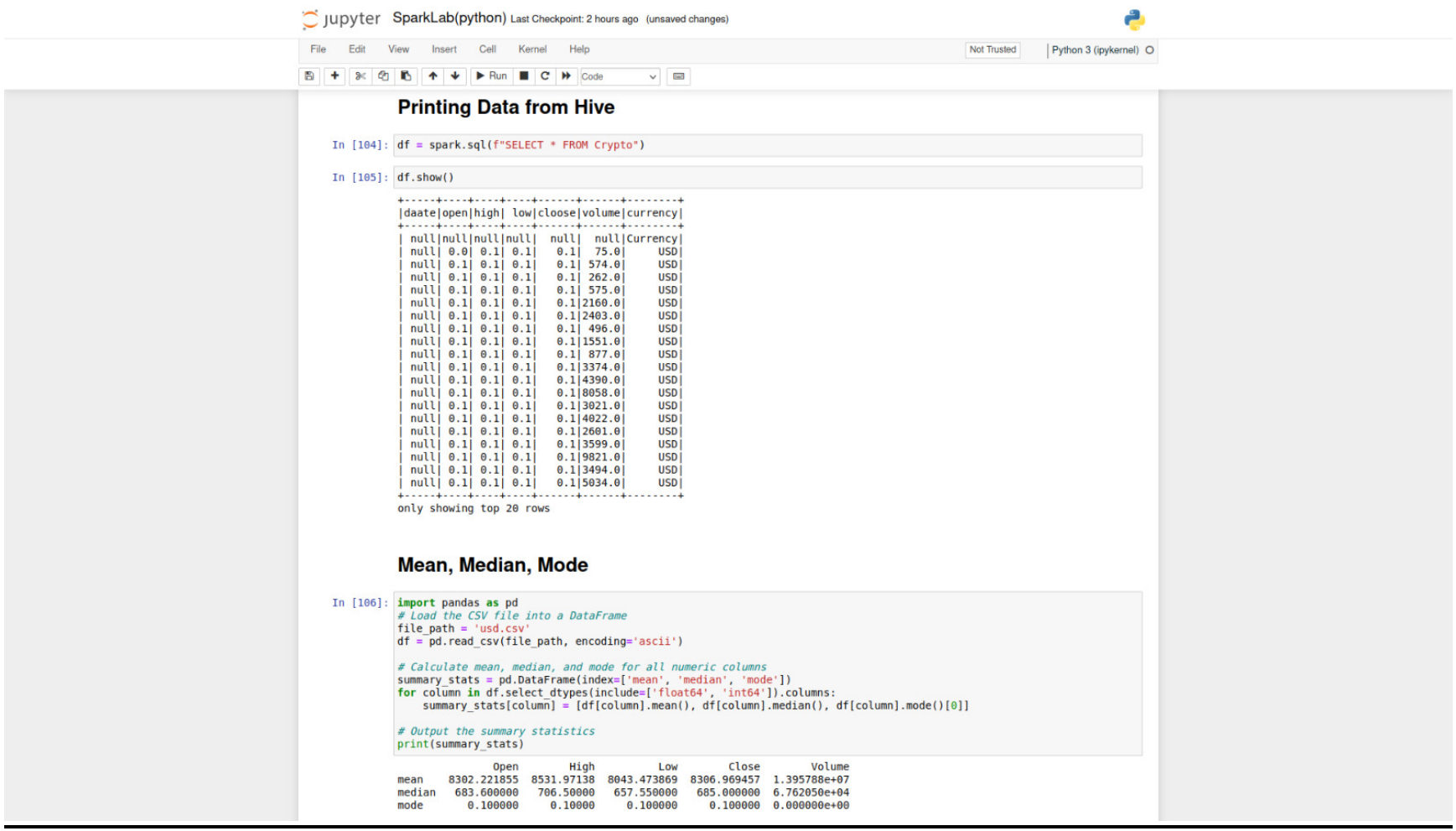
```
(base) root@root:~# hive
SLF4J: Class path contains multiple SLF4J bindings.
SLF4J: Found binding in [jar:file:/opt/hive/lib/log4j-slf4j-impl-2.17.1.jar!/org/slf4j/impl/StaticLoggerBinder.class]
SLF4J: Found binding in [jar:file:/opt/hadoop/share/hadoop/common/lib/slf4j-reload4j-1.7.36.jar!/org/slf4j/impl/StaticLogge
rBinder.class]
SLF4J: See http://www.slf4j.org/codes.html#multiple_bindings for an explanation.
SLF4J: Actual binding is of type [org.apache.logging.slf4j.Log4jLoggerFactory]
Hive Session ID = 5ff86310-9fbf-440b-8dbf-333dd07b65ea

Logging initialized using configuration in jar:file:/opt/hive/lib/hive-common-3.1.3.jar!/hive-log4j2.properties Async: true
Hive-on-MR is deprecated in Hive 2 and may not be available in the future versions. Consider using a different execution en
gine (i.e. spark, tez) or using Hive 1.X releases.Hive Session ID = 71e20623-35f6-46e9-a5b9-8ae0b52a9208

hive> CREATE TABLE IF NOT EXISTS Crypto(
    > daate date,
    > open double,
    > high double,
    > low double,
    > cloose double,
    > volume double,
    > currency string)
    > COMMENT 'CryptoData'
    > ROW FORMAT DELIMITED
    > FIELDS TERMINATED BY ',';
OK
Time taken: 0.895 seconds
hive> LOAD DATA INPATH '/tmp/spark/usd.csv' INTO TABLE Crypto;
Loading data to table default.crypto
OK
Time taken: 0.926 seconds
hive>
```

STEP 4: PYSPARK: LOAD DATA INTO HDFS

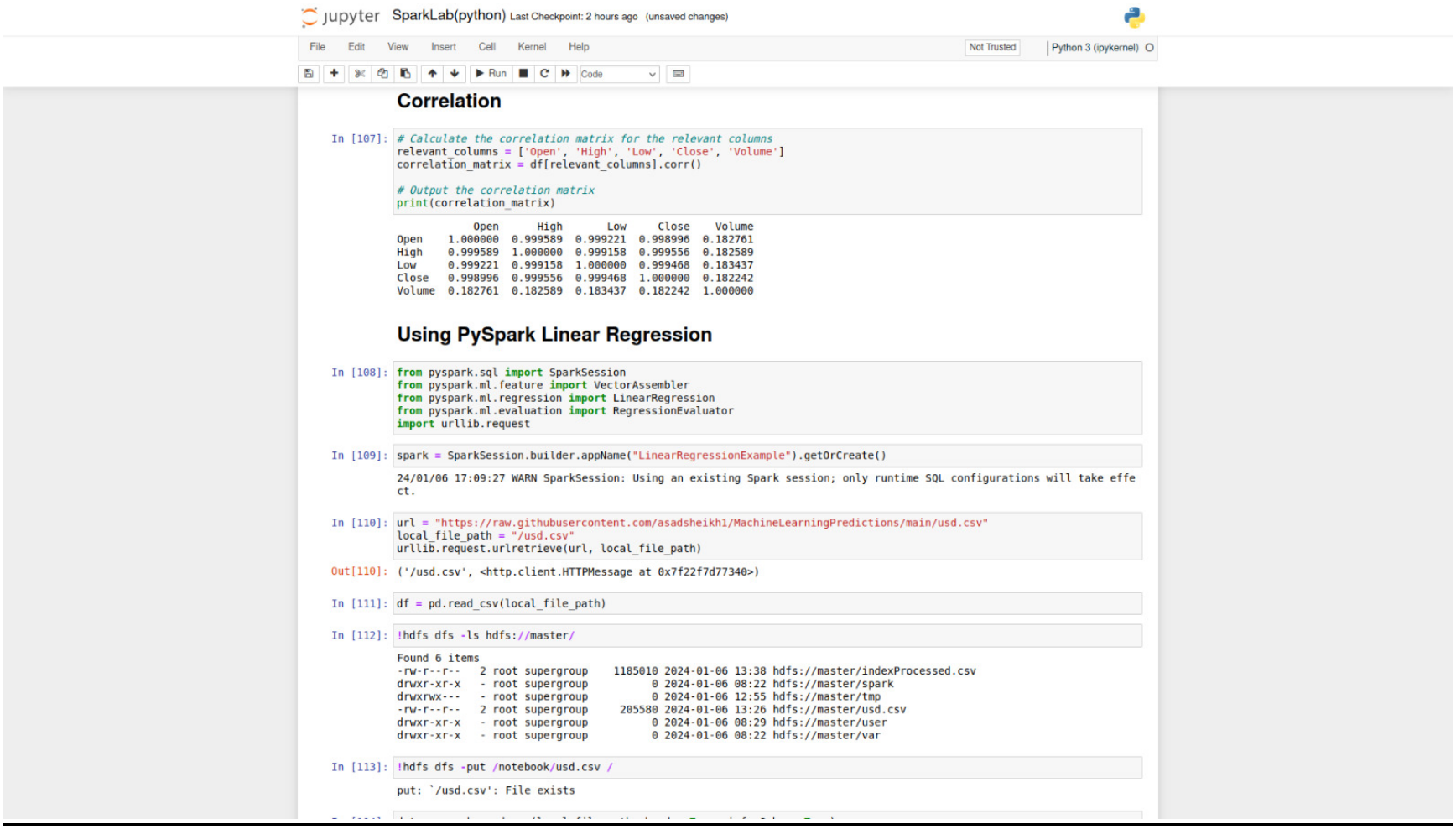



jupyter SparkLab(python) Last Checkpoint: 2 hours ago (unsaved changes)
File Edit View Insert Cell Kernel Help
Not Trusted
Python 3 (ipykernel)
Run
Code
Loading Data into HDFS
In [100]:
!hdfs dfs -put /notebook/usd.csv /tmp/spark/
!hadoop fs -ls /tmp/spark
put: `/tmp/spark/usd.csv': File exists
Found 22 items
-rw-r--r-- 2 root supergroup 64067991 2024-01-06 14:34 /tmp/spark/CAvideos.csv
-rw-r--r-- 2 root supergroup 63040138 2024-01-06 14:34 /tmp/spark/DEvideos.csv
-rw-r--r-- 2 root supergroup 51424708 2024-01-06 14:34 /tmp/spark/FRvideos.csv
-rw-r--r-- 2 root supergroup 53213441 2024-01-06 14:34 /tmp/spark/GBvideos.csv
-rw-r--r-- 2 root supergroup 59600439 2024-01-06 14:34 /tmp/spark/INvideos.csv
-rw-r--r-- 2 root supergroup 28740747 2024-01-06 14:34 /tmp/spark/JPvideos.csv
-rw-r--r-- 2 root supergroup 34835868 2024-01-06 14:34 /tmp/spark/KRvideos.csv
-rw-r--r-- 2 root supergroup 45191541 2024-01-06 14:34 /tmp/spark/MXvideos.csv
-rw-r--r-- 2 root supergroup 76268286 2024-01-06 14:34 /tmp/spark/RUvideos.csv
-rw-r--r-- 2 root supergroup 62756152 2024-01-06 14:34 /tmp/spark/USvideos.csv
drwxr-xr-x - root supergroup 0 2024-01-06 14:50 /tmp/spark/df_1.parquet
drwxr-xr-x - root supergroup 0 2024-01-06 14:50 /tmp/spark/df_10.parquet
drwxr-xr-x - root supergroup 0 2024-01-06 14:50 /tmp/spark/df_2.parquet
drwxr-xr-x - root supergroup 0 2024-01-06 14:50 /tmp/spark/df_3.parquet
drwxr-xr-x - root supergroup 0 2024-01-06 14:50 /tmp/spark/df_4.parquet
drwxr-xr-x - root supergroup 0 2024-01-06 14:50 /tmp/spark/df_5.parquet
drwxr-xr-x - root supergroup 0 2024-01-06 14:50 /tmp/spark/df_6.parquet
drwxr-xr-x - root supergroup 0 2024-01-06 14:50 /tmp/spark/df_7.parquet
drwxr-xr-x - root supergroup 0 2024-01-06 14:50 /tmp/spark/df_8.parquet
drwxr-xr-x - root supergroup 0 2024-01-06 14:50 /tmp/spark/df_9.parquet
-rw-r--r-- 2 root supergroup 1185010 2024-01-06 12:43 /tmp/spark/indexProcessed.csv
-rw-r--r-- 2 root supergroup 205580 2024-01-06 13:13 /tmp/spark/usd.csv
In [101]:
rdd = sc.textFile('hdfs:///tmp/spark/usd.csv')
In [102]:
rdd.foreach(lambda f: print(f))
In [103]:
rdd.collect()
Out[103]:
['Date,Open,High,Low,Close,Volume,Currency',
'18/07/2010,0,0.1,0.1,0.1,75,USD',
'19/07/2010,0.1,0.1,0.1,0.1,574,USD',
'20/07/2010,0.1,0.1,0.1,0.1,262,USD',
'21/07/2010,0.1,0.1,0.1,0.1,575,USD',
'22/07/2010,0.1,0.1,0.1,0.1,2160,USD',
'23/07/2010,0.1,0.1,0.1,0.1,2403,USD',
'24/07/2010,0.1,0.1,0.1,0.1,496,USD',
'25/07/2010,0.1,0.1,0.1,0.1,1551,USD',
'26/07/2010,0.1,0.1,0.1,0.1,877,USD',
'27/07/2010,0.1,0.1,0.1,0.1,3374,USD',
'28/07/2010,0.1,0.1,0.1,0.1,4390,USD',
'29/07/2010,0.1,0.1,0.1,0.1,8058,USD',
'30/07/2010,0.1,0.1,0.1,0.1,3021,USD',
'31/07/2010,0.1,0.1,0.1,0.1,4022,USD',
'01/08/2010,0.1,0.1,0.1,0.1,2601,USD',
'02/08/2010,0.1,0.1,0.1,0.1,3599,USD',
'03/08/2010,0.1,0.1,0.1,0.1,9821,USD',

STEP 5: PRINT DATA FROM HIVE & FIND MEAN, MEDIAN, MODE, & CORRELATION



STEP 6: MACHINE LEARNING MODEL & PREDICTION



jupyter SparkLab(python) Last Checkpoint: 2 hours ago (autosaved)

File Edit View Insert Cell Kernel Help Not Trusted Python 3 (ipykernel)

Run Code

```
drwxr-xr-x   - root supergroup          0 2024-01-06 08:22 hdfs://master/var
```

In [113]:
```
!hdfs dfs -put /notebook/usd.csv /
```

```
put: `/usd.csv': File exists
```

In [114]:
```
data = spark.read.csv(local_file_path, header=True, inferSchema=True)
```

In [115]:
```
data
```

Out[115]:
```
DataFrame[Date: string, Open: double, High: double, Low: double, Close: double, Volume: bigint, Currency: string]
```

In [116]:
```
data.printSchema()
data.show()
```

```
root
 |-- Date: string (nullable = true)
 |-- Open: double (nullable = true)
 |-- High: double (nullable = true)
 |-- Low: double (nullable = true)
 |-- Close: double (nullable = true)
 |-- Volume: long (nullable = true)
 |-- Currency: string (nullable = true)

+----------+----+----+---+-----+------+--------+
|      Date|Open|High|Low|Close|Volume|Currency|
+----------+----+----+---+-----+------+--------+
|18/07/2010| 0.0| 0.1|0.1|  0.1|    75|     USD|
|19/07/2010| 0.1| 0.1|0.1|  0.1|   574|     USD|
|20/07/2010| 0.1| 0.1|0.1|  0.1|   262|     USD|
|21/07/2010| 0.1| 0.1|0.1|  0.1|   575|     USD|
|22/07/2010| 0.1| 0.1|0.1|  0.1|  2160|     USD|
|23/07/2010| 0.1| 0.1|0.1|  0.1|  2403|     USD|
|24/07/2010| 0.1| 0.1|0.1|  0.1|   496|     USD|
|25/07/2010| 0.1| 0.1|0.1|  0.1|  1551|     USD|
|26/07/2010| 0.1| 0.1|0.1|  0.1|   877|     USD|
|27/07/2010| 0.1| 0.1|0.1|  0.1|  3374|     USD|
|28/07/2010| 0.1| 0.1|0.1|  0.1|  4390|     USD|
|29/07/2010| 0.1| 0.1|0.1|  0.1|  8058|     USD|
|30/07/2010| 0.1| 0.1|0.1|  0.1|  3021|     USD|
|31/07/2010| 0.1| 0.1|0.1|  0.1|  4022|     USD|
|01/08/2010| 0.1| 0.1|0.1|  0.1|  2601|     USD|
|02/08/2010| 0.1| 0.1|0.1|  0.1|  3599|     USD|
|03/08/2010| 0.1| 0.1|0.1|  0.1|  9821|     USD|
|04/08/2010| 0.1| 0.1|0.1|  0.1|  3494|     USD|
|05/08/2010| 0.1| 0.1|0.1|  0.1|  5034|     USD|
|06/08/2010| 0.1| 0.1|0.1|  0.1|  1395|     USD|
+----------+----+----+---+-----+------+--------+
only showing top 20 rows
```

In [117]:
```
assembler = VectorAssembler(inputCols=['Open', 'High', 'Low', 'Volume'], outputCol='features')
data = assembler.transform(data)
```

jupyter SparkLab(python) Last Checkpoint: 2 hours ago (autosaved)

File Edit View Insert Cell Kernel Help Not Trusted Python 3 (ipykernel)

Run Code

In [117]:
```
assembler = VectorAssembler(inputCols=['Open', 'High', 'Low', 'Volume'], outputCol='features')
data = assembler.transform(data)
```

In [118]:
```
data = data.withColumnRenamed('Close', 'label')
```

In [119]:
```
train_data, test_data = data.randomSplit([0.8, 0.2], seed=1234)
```

In [120]:
```
lr = LinearRegression(featuresCol='features', labelCol='label')
```

In [121]:
```
lr_model = lr.fit(train_data)
```

```
24/01/06 17:09:35 WARN Instrumentation: [d6ac34b7] regParam is zero, which might cause numerical instability and ove
rfitting.
```

In [122]:
```
predictions = lr_model.transform(test_data)
```

In [123]:
```
evaluator = RegressionEvaluator(labelCol='label', predictionCol='prediction', metricName='rmse')
rmse = evaluator.evaluate(predictions)
evaluator = RegressionEvaluator(labelCol='label', predictionCol='prediction', metricName='rmse')
rmse = evaluator.evaluate(predictions)
```

In [124]:
```
print("Mean Absolute Error: ", metrics.mean_absolute_error(Y_test, y_pred))
print("Mean Squared Error: ", metrics.mean_squared_error(Y_test, y_pred))
print("Root Mean Squared Error: ", np.sqrt(metrics.mean_squared_error(Y_test, y_pred)))
print("Coefficient of Determination (R2 Score): ", metrics.r2_score(Y_test, y_pred))
```

```
Mean Absolute Error:  0.0008327265398649373
Mean Squared Error:  2.8538136120404585e-06
Root Mean Squared Error:  0.001689323418425394
Coefficient of Determination (R2 Score):  0.9999378230963039
```

Using Linear Regression

In [125]:
```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import MinMaxScaler
from sklearn.linear_model import LinearRegression
from sklearn import metrics
import warnings
from sklearn import metrics
warnings.filterwarnings("ignore")
```

In [126]:
```
data = pd.read_csv("https://raw.githubusercontent.com/asadsheikh1/MachineLearningPredictions/main/usd.csv")
```

In [52]:
```
data
```

jupyter SparkLab(python) Last Checkpoint: 2 hours ago (autosaved)

File Edit View Insert Cell Kernel Help Not Trusted Python 3 (ipykernel)

Run Code

```
warnings.filterwarnings("ignore")
```

In [126]:
```
data = pd.read_csv("https://raw.githubusercontent.com/asadsheikh1/MachineLearningPredictions/main/usd.csv")
```

In [52]:
```
data
```

Out[52]:

	Date	Open	High	Low	Close	Volume	Currency
0	18/07/2010	0.0	0.1	0.1	0.1	75	USD
1	19/07/2010	0.1	0.1	0.1	0.1	574	USD
2	20/07/2010	0.1	0.1	0.1	0.1	262	USD
3	21/07/2010	0.1	0.1	0.1	0.1	575	USD
4	22/07/2010	0.1	0.1	0.1	0.1	2160	USD
...	...	...	...	...	...	...	...
4415	19/08/2022	23201.6	23202.3	20807.8	20831.3	339472	USD
4416	20/08/2022	20830.7	21357.4	20784.8	21138.9	206943	USD
4417	21/08/2022	21138.9	21692.4	21077.4	21517.2	177522	USD
4418	22/08/2022	21516.8	21517.4	20912.1	21416.3	251833	USD
4419	23/08/2022	21416.5	21458.2	21271.2	21309.0	251695	USD

4420 rows × 7 columns

In [53]:
```
data.isnull().sum()
```

Out[53]:
```
Date        0
Open        0
High        0
Low         0
Close       0
Volume      0
Currency    0
dtype: int64
```

In [54]:
```
data.describe()
```

Out[54]:

	Open	High	Low	Close	Volume
count	4420.000000	4420.000000	4420.000000	4420.000000	4.420000e+03
mean	8302.221855	8531.971380	8043.473869	8306.969457	1.395788e+07
std	14598.398937	14992.605971	14146.499234	14599.046649	1.645973e+08
min	0.000000	0.100000	0.000000	0.100000	0.000000e+00
25%	122.500000	129.975000	118.300000	122.800000	3.013600e+04
50%	683.600000	706.500000	657.550000	685.000000	6.762050e+04
75%	8877.250000	9122.700000	8666.900000	8886.075000	1.769402e+05
max	67528.700000	68990.600000	66334.900000	67527.900000	4.468697e+09

jupyter SparkLab(python) Last Checkpoint: 2 hours ago (autosaved)

File Edit View Insert Cell Kernel Help Not Trusted Python 3 (ipykernel)

Run Code

In [54]:
```
data.describe()
```

Out[54]:

	Open	High	Low	Close	Volume
count	4420.000000	4420.000000	4420.000000	4420.000000	4.420000e+03
mean	8302.221855	8531.971380	8043.473869	8306.969457	1.395788e+07
std	14598.398937	14992.605971	14146.499234	14599.046649	1.645973e+08
min	0.000000	0.100000	0.000000	0.100000	0.000000e+00
25%	122.500000	129.975000	118.300000	122.800000	3.013600e+04
50%	683.600000	706.500000	657.550000	685.000000	6.762050e+04
75%	8877.250000	9122.700000	8666.900000	8886.075000	1.769402e+05
max	67528.700000	68990.600000	66334.900000	67527.900000	4.468697e+09

In [55]:
```
data.nunique()
```

Out[55]:
```
Date        4420
Open        3462
High        3453
Low         3463
Close       3469
Volume      4369
Currency       1
dtype: int64
```

In [56]:
```
data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 4420 entries, 0 to 4419
Data columns (total 7 columns):
 #   Column    Non-Null Count  Dtype
---  ------    --------------  -----
 0   Date      4420 non-null   object
 1   Open      4420 non-null   float64
 2   High      4420 non-null   float64
 3   Low       4420 non-null   float64
 4   Close     4420 non-null   float64
 5   Volume    4420 non-null   int64
 6   Currency  4420 non-null   object
dtypes: float64(4), int64(1), object(2)
memory usage: 241.8+ KB
```

In [57]:
```
data['Date'] = pd.to_datetime(data['Date'])
data['Date'] = (data['Date'] - data['Date'].min()).dt.days
```

In [58]:
```
data.plot(x='Date', y='Open', kind='scatter')
plt.title('Date vs Open')
plt.xlabel('Date')
plt.ylabel('Open')
plt.show()
```

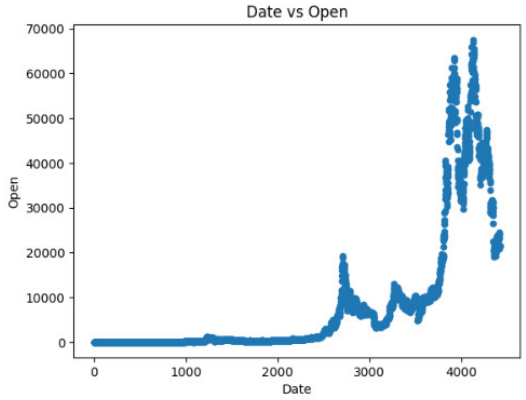
Date vs Open

jupyter SparkLab(python) Last Checkpoint: 2 hours ago (autosaved)

File Edit View Insert Cell Kernel Help Not Trusted Python 3 (ipykernel)

Run Code

In [58]:
```
data.plot(x='Date', y='Open', kind='scatter')
plt.title('Date vs Open')
plt.xlabel('Date')
plt.ylabel('Open')
plt.show()
```



In [59]:
```
data.describe()
```

Out[59]:

	Date	Open	High	Low	Close	Volume
count	4420.000000	4420.000000	4420.000000	4420.000000	4420.000000	4.420000e+03
mean	2209.500000	8302.221855	8531.971380	8043.473869	8306.969457	1.395788e+07
std	1276.088424	14598.398937	14992.605971	14146.499234	14599.046649	1.645973e+08
min	0.000000	0.000000	0.100000	0.000000	0.100000	0.000000e+00
25%	1104.750000	122.500000	129.975000	118.300000	122.800000	3.013600e+04
50%	2209.500000	683.600000	706.500000	657.550000	685.000000	6.762050e+04
75%	3314.250000	8877.250000	9122.700000	8666.900000	8886.075000	1.769402e+05
max	4419.000000	67528.700000	68990.600000	66334.900000	67527.900000	4.468697e+09

In [60]:
```
Columns = ['Date', 'Open', 'High', 'Low', 'Close', 'Volume']
```

In [61]:
```
min_max_scaler = MinMaxScaler()
data[Columns] = min_max_scaler.fit_transform(data[Columns])
data.head()
```

jupyter SparkLab(python) Last Checkpoint: 2 hours ago (autosaved)

File Edit View Insert Cell Kernel Help Not Trusted Python 3 (ipykernel)

Run Code

In [60]:
```
Columns = ['Date', 'Open', 'High', 'Low', 'Close', 'Volume']
```

In [61]:
```
min_max_scaler = MinMaxScaler()
data[Columns] = min_max_scaler.fit_transform(data[Columns])
data.head()
```

Out[61]:

	Date	Open	High	Low	Close	Volume	Currency
0	0.000000	0.000000	0.0	0.000002	0.0	1.678342e-08	USD
1	0.000226	0.000001	0.0	0.000002	0.0	1.284491e-07	USD
2	0.000453	0.000001	0.0	0.000002	0.0	5.863007e-08	USD
3	0.000679	0.000001	0.0	0.000002	0.0	1.286729e-07	USD
4	0.000905	0.000001	0.0	0.000002	0.0	4.833624e-07	USD

In [62]:
```
data['Currency'] = (data['Currency'] == 'USD').astype(int)
```

In [63]:
```
X = data.drop(['Open'] , axis = 1)
Y = data['Open']
```

In [64]:
```
X_train , X_test , Y_train , Y_test = train_test_split(X , Y, test_size = 0.2 , random_state = 30)
```

Linear Regression

In [65]:
```
model = LinearRegression()
model.fit(X_train, Y_train)
```

Out[65]:
```
▾ LinearRegression
LinearRegression()
```

In [66]:
```
X.size
```

Out[66]: 26520

In [67]:
```
Y.size
```

Out[67]: 4420

In [68]:
```
y_pred = model.predict(X_test)

plt.scatter(Y_test, y_pred)
plt.xlabel("Actual Values (Y_test)")
plt.ylabel("Predicted Values")
plt.title("Linear Regression: Actual vs Predicted")
plt.show()
```

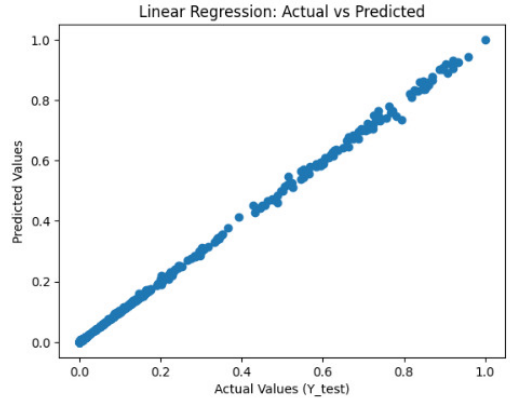
Linear Regression: Actual vs Predicted

jupyter SparkLab(python) Last Checkpoint: 2 hours ago (autosaved)

File Edit View Insert Cell Kernel Help Not Trusted Python 3 (ipykernel)

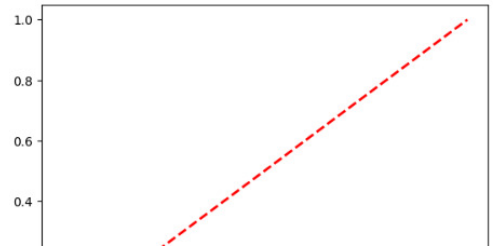
Run Code

In [68]:
```
y_pred = model.predict(X_test)

plt.scatter(Y_test, y_pred)
plt.xlabel("Actual Values (Y_test)")
plt.ylabel("Predicted Values")
plt.title("Linear Regression: Actual vs Predicted")
plt.show()
```



In [69]:
```
plt.plot([min(Y_test), max(Y_test)], [min(Y_test), max(Y_test)], linestyle='--', color='red', linewidth=2)
```

Out[69]:
```
[<matplotlib.lines.Line2D at 0x7f5598525f10>]
```



jupyter SparkLab(python) Last Checkpoint: 4 hours ago (autosaved)

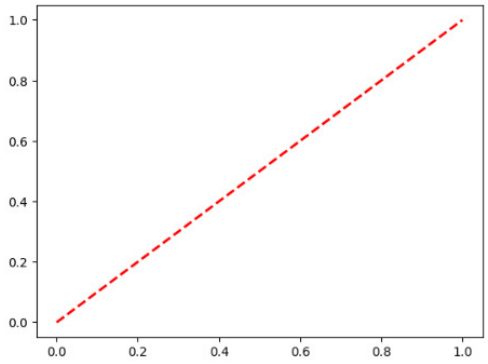
File Edit View Insert Cell Kernel Help Not Trusted Python 3 (ipykernel)

Run Code

Actual Values (Y_test)

In [146]:
```
plt.plot([min(Y_test), max(Y_test)], [min(Y_test), max(Y_test)], linestyle='--', color='red', linewidth=2)
```

Out[146]:
```
[<matplotlib.lines.Line2D at 0x7f2307345f70>]
```



In [147]:
```
y_pred = model.predict(X_test)
```

In [148]:
```
predicted_vs_target = pd.DataFrame({'Predicted': y_pred, 'Target': Y_test})
predicted_vs_target
```

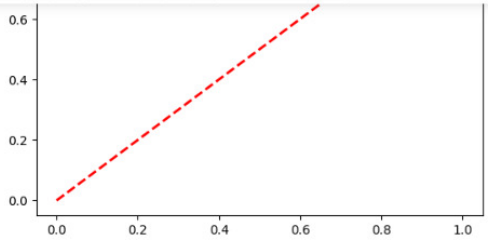
Out[148]:

	Predicted	Target
4319	0.444401	0.445420
4042	0.684378	0.675168
3300	0.143587	0.141870
1852	0.003933	0.003964
3324	0.151067	0.153852
...	...	...
664	0.000064	0.000074
29	-0.000019	0.000001
1452	0.009044	0.009112
3050	0.063872	0.064578
2717	0.205293	0.204209

jupyter SparkLab(python) Last Checkpoint: 4 hours ago (autosaved)

File Edit View Insert Cell Kernel Help Not Trusted Python 3 (ipykernel)

Run Code



In [147]:
```
y_pred = model.predict(X_test)
```

In [148]:
```
predicted_vs_target = pd.DataFrame({'Predicted': y_pred, 'Target': Y_test})
predicted_vs_target
```

Out[148]:

	Predicted	Target
4319	0.444401	0.445420
4042	0.684378	0.675168
3300	0.143587	0.141870
1852	0.003933	0.003964
3324	0.151067	0.153852
...	...	...
664	0.000064	0.000074
29	-0.000019	0.000001
1452	0.009044	0.009112
3050	0.063872	0.064578
2717	0.205293	0.204209

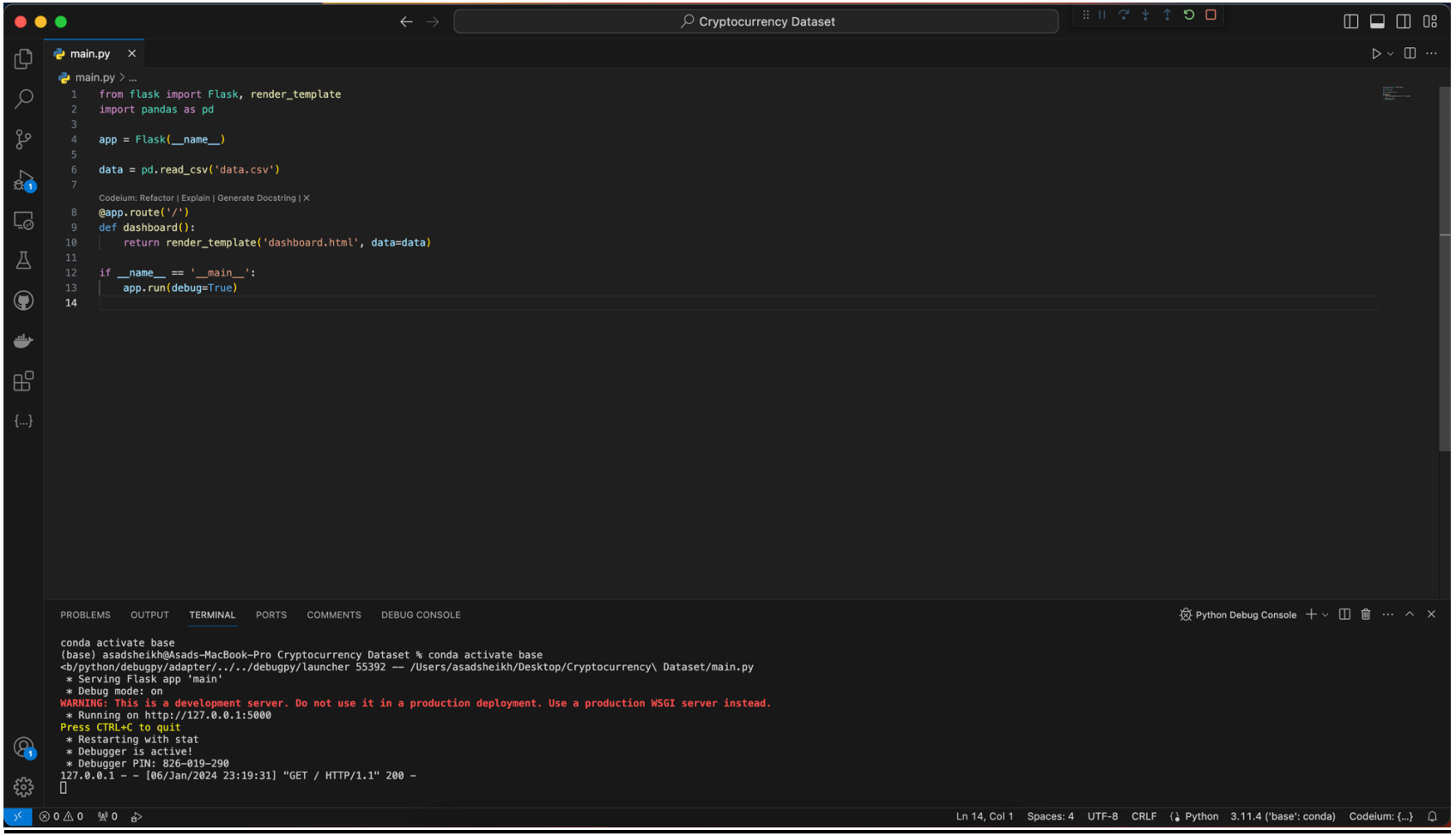
884 rows × 2 columns

In [149]:
```
print("Mean Absolute Error: ", metrics.mean_absolute_error(Y_test, y_pred))
print("Mean Squared Error: ", metrics.mean_squared_error(Y_test, y_pred))
print("Root Mean Squared Error: ", np.sqrt(metrics.mean_squared_error(Y_test, y_pred)))
print("Coefficient of Determination (R2 Score): ", metrics.r2_score(Y_test, y_pred))
```

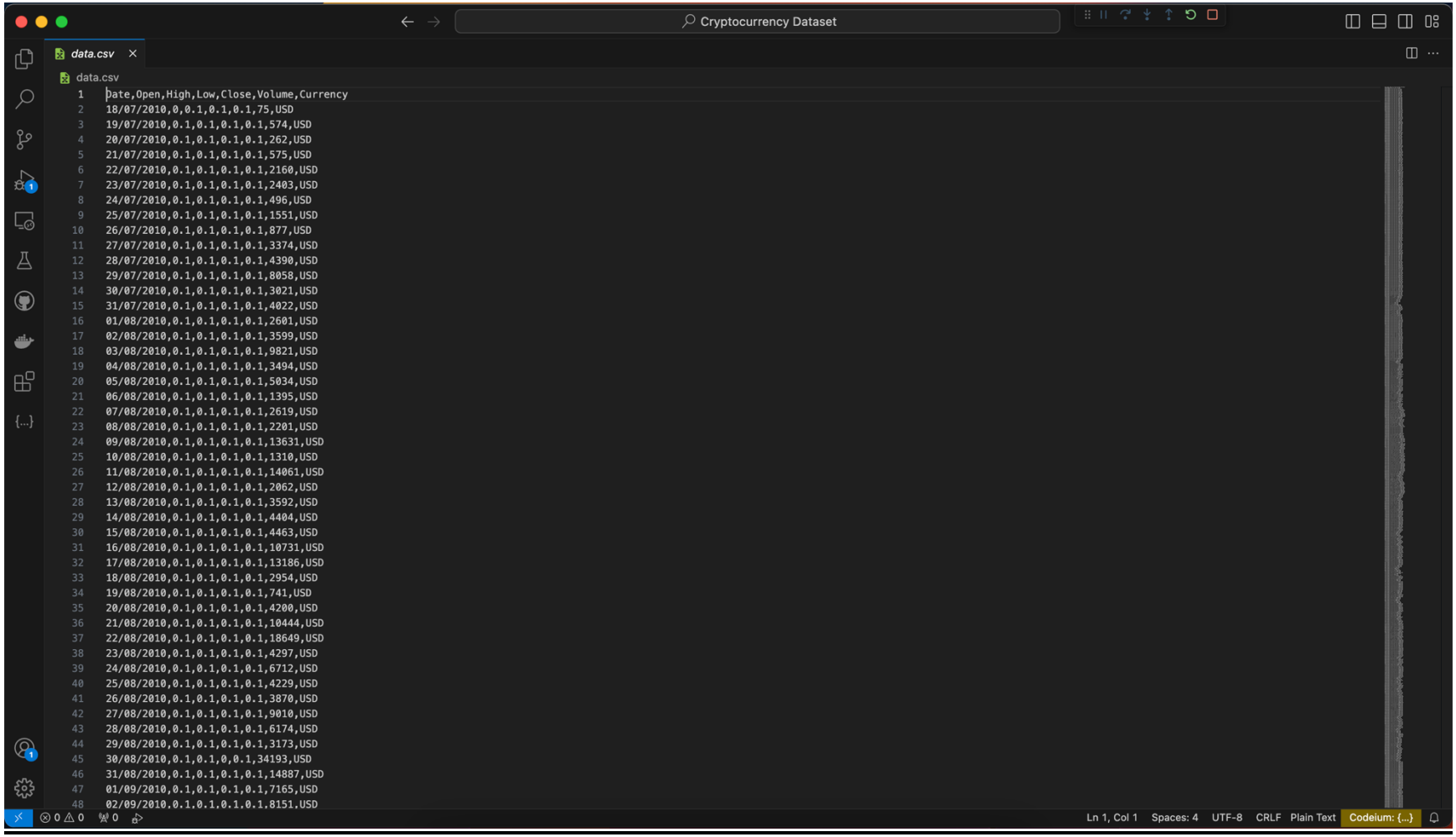
```
Mean Absolute Error:  0.0018525007111413413
Mean Squared Error:  2.366342063341151e-05
Root Mean Squared Error:  0.004864506206534381
Coefficient of Determination (R2 Score):  0.9995404653805043
```

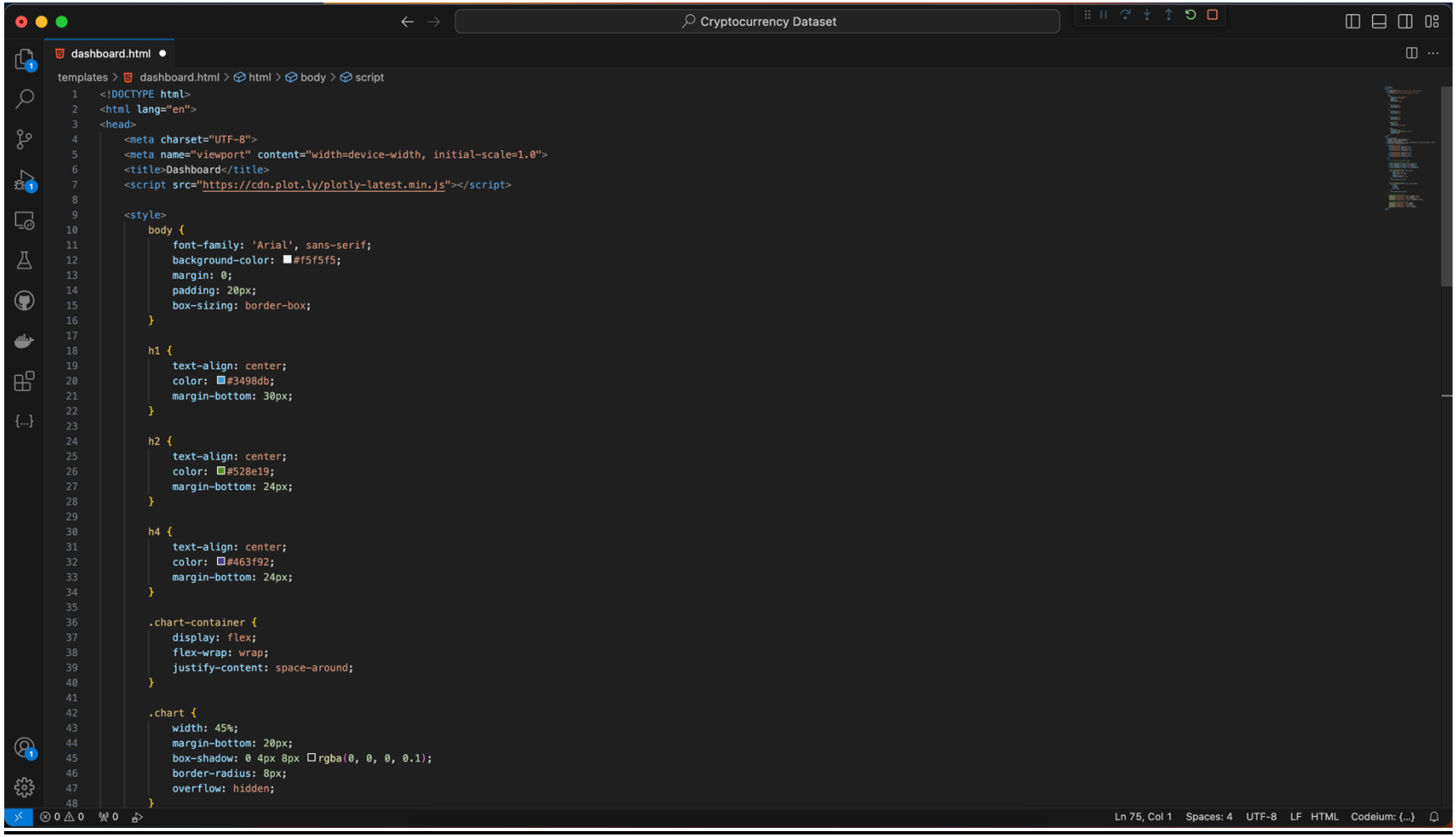
In [ ]:

STEP 7: DASHBOARDING THROUGH FLASK

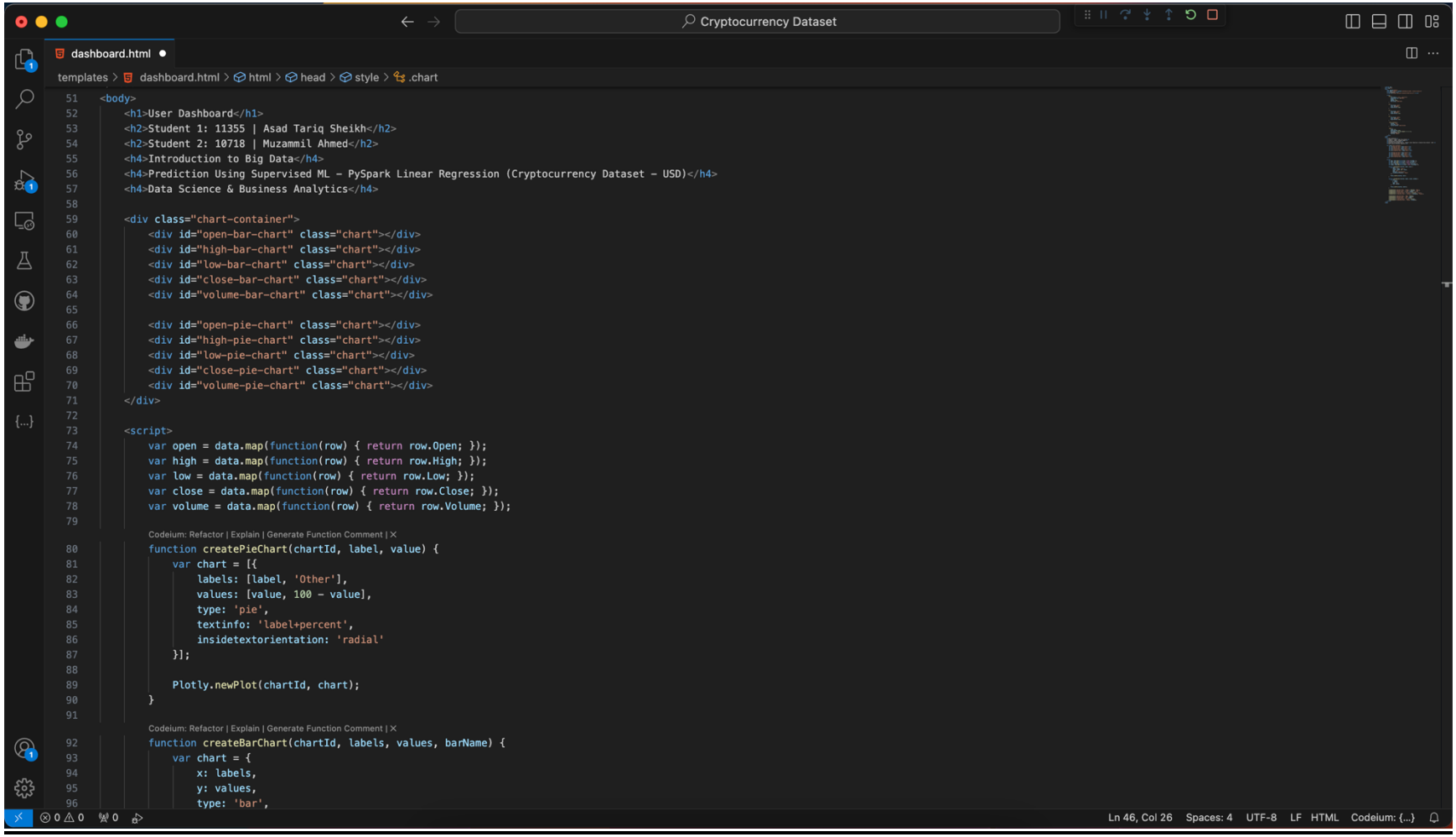


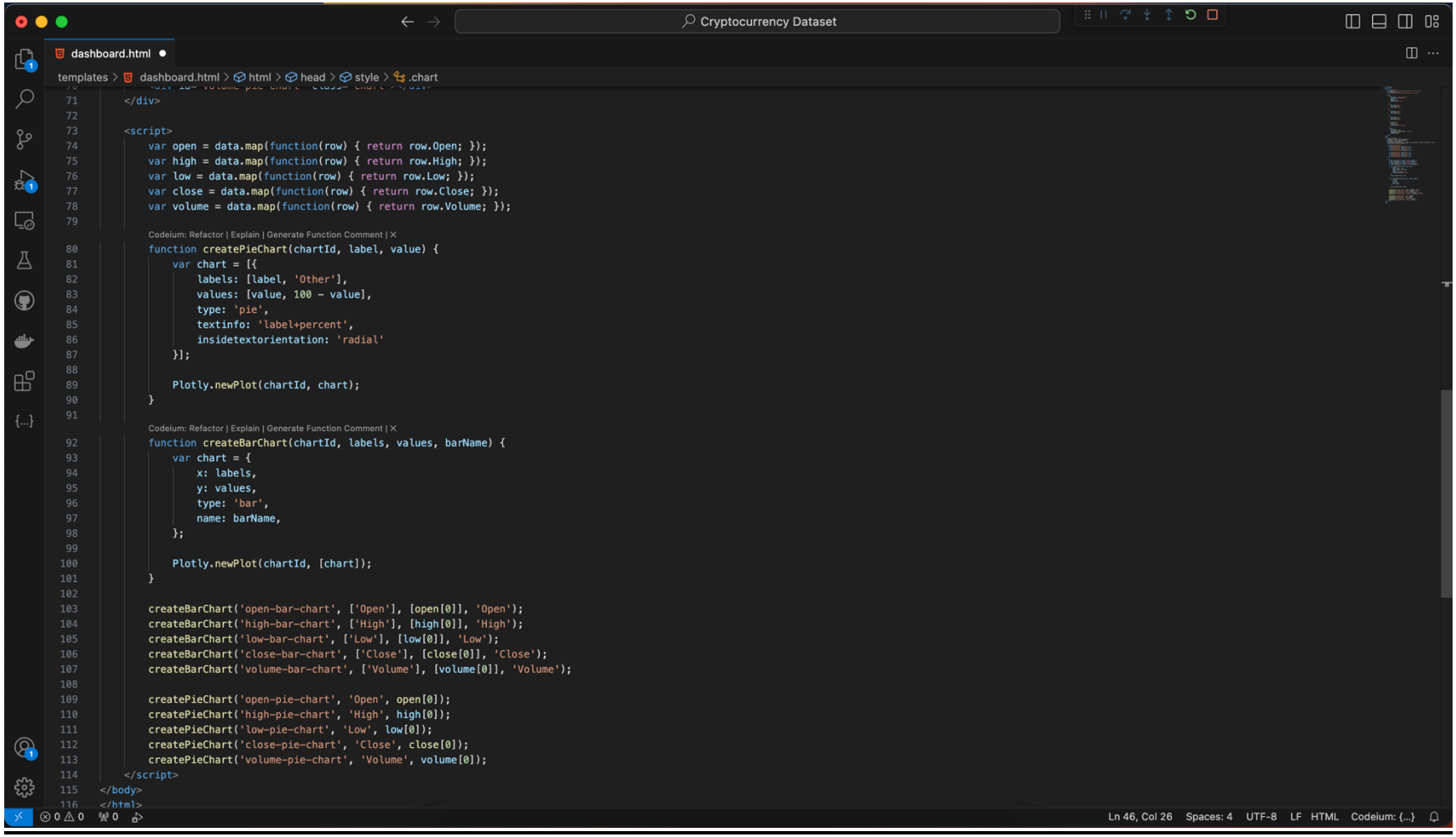
DATA.CSV FILE




Cryptocurrency Dataset
dashboard.html
templates > dashboard.html > html > body > script
<!DOCTYPE html>
<html lang="en">
<head>
<meta charset="UTF-8">
<meta name="viewport" content="width=device-width, initial-scale=1.0">
<title>Dashboard</title>
<script src="https://cdn.plot.ly/plotly-latest.min.js"></script>
<style>
body {
font-family: 'Arial', sans-serif;
background-color: #f5f5f5;
margin: 0;
padding: 20px;
box-sizing: border-box;
}
h1 {
text-align: center;
color: #3498db;
margin-bottom: 30px;
}
h2 {
text-align: center;
color: #528e19;
margin-bottom: 24px;
}
h4 {
text-align: center;
color: #463f92;
margin-bottom: 24px;
}
.chart-container {
display: flex;
flex-wrap: wrap;
justify-content: space-around;
}
.chart {
width: 45%;
margin-bottom: 20px;
box-shadow: 0 4px 8px rgba(0, 0, 0, 0.1);
border-radius: 8px;
overflow: hidden;
}
Ln 75, Col 1 Spaces: 4 UTF-8 LF HTML Codeium: {...}

DASHBOARD.HTML FILE




Cryptocurrency Dataset
dashboard.html
templates > dashboard.html > html > head > style > .chart

<script>
var open = data.map(function(row) { return row.Open; });
var high = data.map(function(row) { return row.High; });
var low = data.map(function(row) { return row.Low; });
var close = data.map(function(row) { return row.Close; });
var volume = data.map(function(row) { return row.Volume; });
Codeium: Refactor | Explain | Generate Function Comment | ×
function createPieChart(chartId, label, value) {
var chart = [{
labels: [label, 'Other'],
values: [value, 100 - value],
type: 'pie',
textinfo: 'label+percent',
insidetextorientation: 'radial'
}];
Plotly.newPlot(chartId, chart);
}
Codeium: Refactor | Explain | Generate Function Comment | ×
function createBarChart(chartId, labels, values, barName) {
var chart = {
x: labels,
y: values,
type: 'bar',
name: barName,
};
Plotly.newPlot(chartId, [chart]);
}
createBarChart('open-bar-chart', ['Open'], [open[0]], 'Open');
createBarChart('high-bar-chart', ['High'], [high[0]], 'High');
createBarChart('low-bar-chart', ['Low'], [low[0]], 'Low');
createBarChart('close-bar-chart', ['Close'], [close[0]], 'Close');
createBarChart('volume-bar-chart', ['Volume'], [volume[0]], 'Volume');
createPieChart('open-pie-chart', 'Open', open[0]);
createPieChart('high-pie-chart', 'High', high[0]);
createPieChart('low-pie-chart', 'Low', low[0]);
createPieChart('close-pie-chart', 'Close', close[0]);
createPieChart('volume-pie-chart', 'Volume', volume[0]);
</script>
</body>
</html>
Ln 46, Col 26 Spaces: 4 UTF-8 LF HTML Codeium: {...}

STEP 8: SHOWING DASHBOARD (BAR & PIE GRAPHS)

User Dashboard

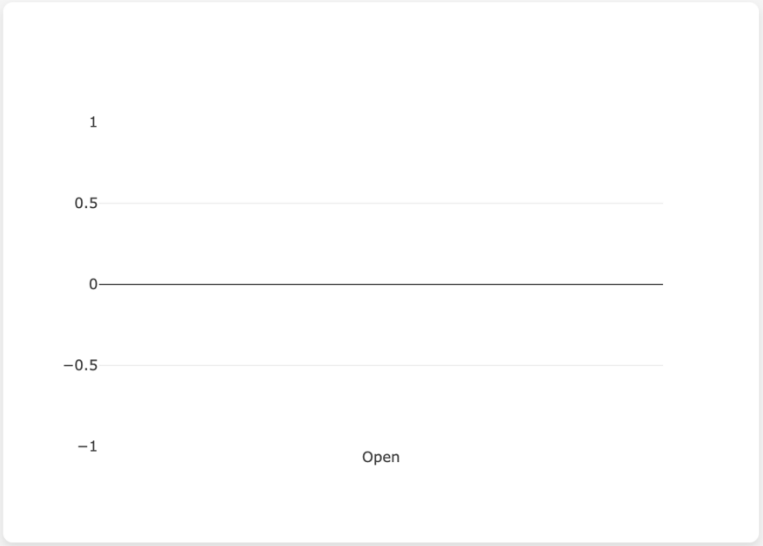
Student 1: 11355 | Asad Tariq Sheikh

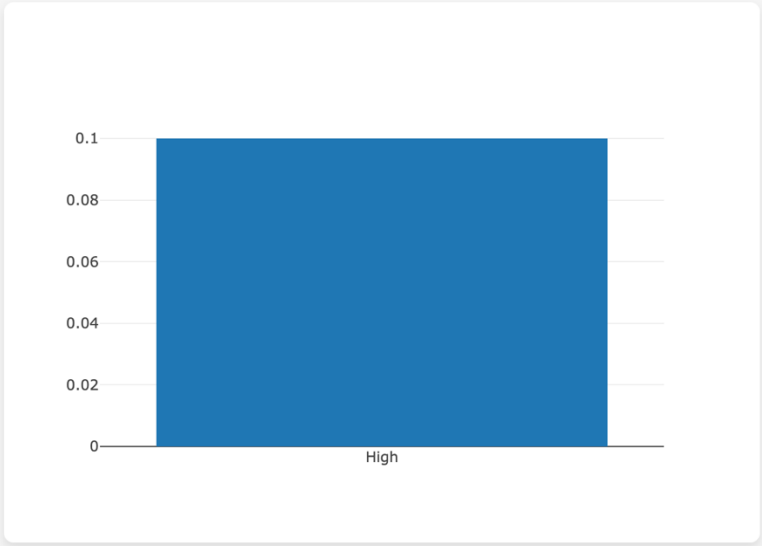
Student 2: 10718 | Muzammil Ahmed

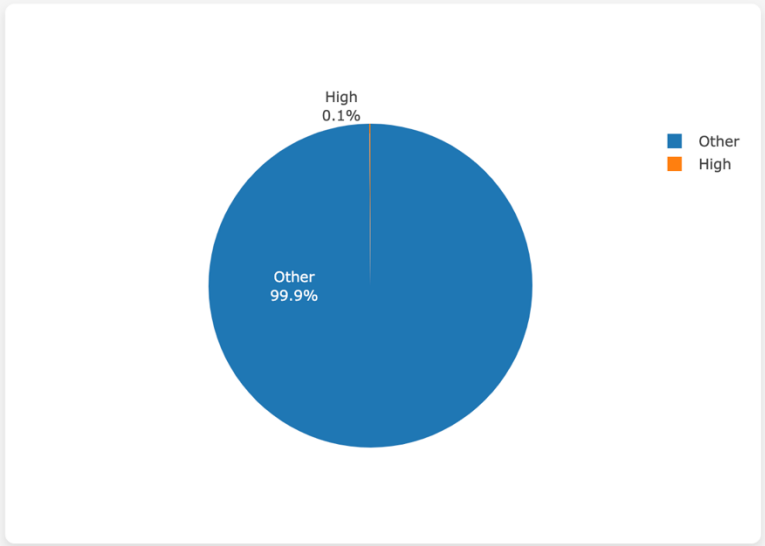
Introduction to Big Data

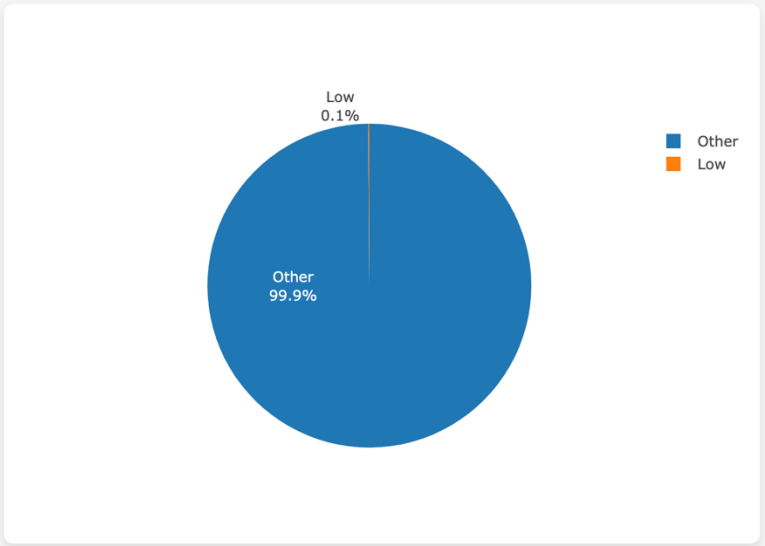
Prediction Using Supervised ML - PySpark Linear Regression (Cryptocurrency Dataset - USD)

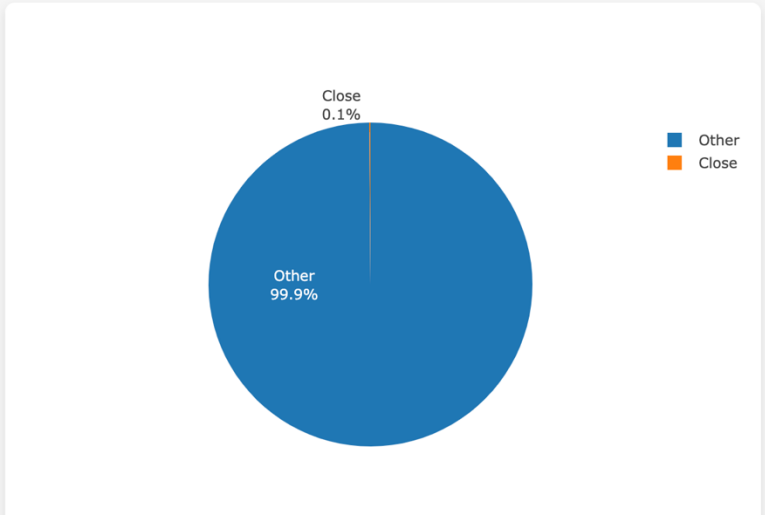
Data Science & Business Analytics

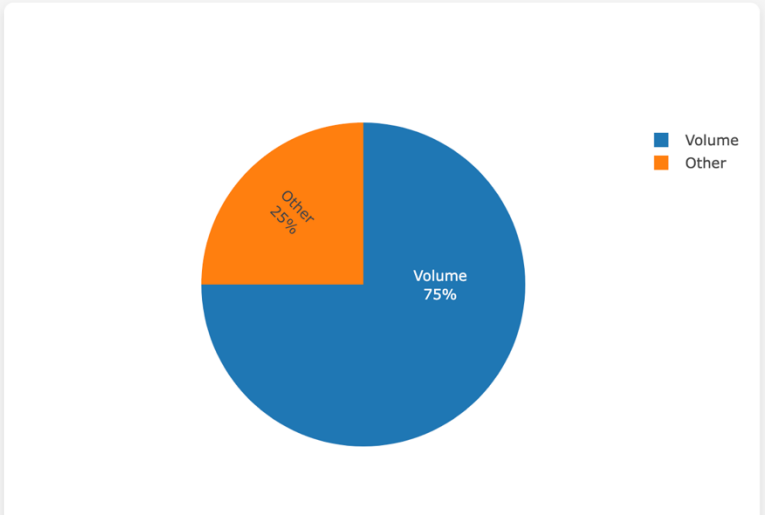





High
0.1%
Other
99.9%
Other
High


Low
0.1%
Other
99.9%
Other
Low


Close
0.1%
Other
99.9%
Other
Close


Other
25%
Volume
75%
Volume
Other

GITHUB LINK: DOCKER FILE USED

https://github.com/shaikhahmad/miniLake

YOUTUBE VIDEO LINK (IN CASE THE VIDEO FILE IS DAMAGED)

https://youtu.be/MYA7njFURCY

CONFIGURATION OF MACHINE USED FOR THE PROJECT

Core i7 6th Generation
Quad Core Processor
256Gb SSD
2GB AMD FirePro GPU
RAM: 16GB
Architecture: x86_64
CPU op-mode(s): 32-bit, 64-bit
Address sizes: 39 bits physical, 48 bits virtual
Byte Order: Little Endian
CPU(s): 8
On-line CPU(s) list: 0-7
Vendor ID: GenuineIntel
Model name: Intel(R) Core(TM) i7-6820HQ CPU @ 2.70GHz
CPU family: 6
Model: 94
Thread(s) per core: 2
Core(s) per socket: 4
Socket(s): 1
Stepping: 3
00:02.0 VGA compatible controller: Intel Corporation HD Graphics 530 (rev 06) (prog-if 00 [VGA controller])
asad-latitude
    description: Laptop
    product: Latitude E5570 (06DF)
    vendor: Dell Inc.
    serial: FKN8JC2
    width: 64 bits
    capabilities: smbios-3.0.0 dmi-3.0.0 smp vsyscall32
    configuration: boot=normal chassis=laptop family=Latitude sku=06DF uuid=4c4c4544-004b-4e10-8038-c6c04f4a4332
  *-core
       description: Motherboard
       product: 06YF8N
       vendor: Dell Inc.
       physical id: 0
       version: A00
       serial: /FKN8JC2/CN1296366U0028/

*-firmware
    description: BIOS
    vendor: Dell Inc.
    physical id: 0
    version: 1.34.3
    date: 11/20/2022
    size: 64KiB
    capacity: 16MiB