Projekat 3 – Big data

Nikola Đorđević 1567

Kreiranje modela (app_3a.py)

Dodavanje klasnog atributa "visit_in_worktime"

```
# Kreiranje kolone koja sadrži sat cekiranja
df_hour = df.withColumn('visit_hour', hour('check_in_time'))

# Kreiranje kolone 'visit_in_worktime' koja će imati vrednost 'True' ako je cekiranje u radno vreme, a 'False' ako nije
df_worktime = df_hour.withColumn('visit_in_worktime', when((col('visit_hour') >= 9) & (col('visit_hour') < 17), True).otherwise(False))
df_worktime = df_worktime.drop('visit_hour', 'time_spent', 'location_id')
```

- Dodaje se klasni atribut "visit_in_worktime" koji pokazauje da li je čekiranje za vreme radnog vremena ili ne
- Atrbiut uzima vrednost True ukoliko je vreme čekiranja između 9č i 17č, u suprotnom je njegova vrednost False

Priprema i obučavanje modela

```
# Podela na training i test set
train, test = df_worktime.randomSplit([0.7, 0.3], seed=42)

# Konvertovanje kolone 'visit_in_worktime' u 0 ili 1
train = train.withColumn('visit_in_worktime', when(col('visit_in_worktime') == True, 1).otherwise(0))
test = test.withColumn('visit_in_worktime', when(col('visit_in_worktime') == True, 1).otherwise(0))

# Kreiranje vektora feature-a
feature_cols = ["user", "check_in_time", "latitude", "longitude"]
assembler = VectorAssembler(inputCols=feature_cols, outputCol="features")

# Transformacija skupa za treniranje i skup za testiranje
train = assembler.transform(train).select("features", "visit_in_worktime")
test = assembler.transform(test).select("features", "visit_in_worktime")

# Kreiranje i treniranje Decision Tree klasifikatora
dt = DecisionTreeClassifier(featuresCol="features", labelCol="visit_in_worktime")
model = dt.fit(train)
model.save(sys.argv[2])
```

- Vrši se podela na test i trening skup podataka
- Biraju se kolone koje će se koristiti za obučavanje modela
- Vrši se obučavanje modela
- Model se čuva na HDFS-u

Evaluacija modela

```
# Klasifikacija test set-a
predictions = model.transform(test)

# Evaluacija klasifikatora

# Tacnost
evaluator = MulticlassClassificationEvaluator(predictionCol="prediction", labelCol="visit_in_worktime", metricName="accuracy")
accuracy = evaluator.evaluate(predictions)
print("Tacnost modela: ", accuracy)

# F1 score-a
f1_evaluator = MulticlassClassificationEvaluator(predictionCol="prediction", labelCol="visit_in_worktime", metricName="f1")
f1_score = f1_evaluator.evaluate(predictions)
print("F1 skor modela: ", f1_score)

# Recall
recall_evaluator = MulticlassClassificationEvaluator(predictionCol="prediction", labelCol="visit_in_worktime", metricName="weightedRecall")
recall = recall_evaluator.evaluate(predictions)
print("Odziv modela: ", recall)

# Precision
precision_evaluator = MulticlassClassificationEvaluator(predictionCol="prediction", labelCol="visit_in_worktime", metricName="weightedPrecision")
precision = precision_evaluator.evaluate(predictions)
print("Preciznost modela: ", precision)

print("Raspodela po klasama")
predictions.groupBy("visit_in_worktime").count().show()

# Matrica konfuzije
predictions = predictions.withColumn("visit_in_worktime", col("visit_in_worktime").cast("double"))
predictionAndLabels = predictions.select("prediction", "visit_in_worktime").rdd
metrics = MulticlassMetrics(predictionAndLabels)
print("Matrica konfuzije:")
print(metrics.confusionMatrix().toArray())
```

- Evaluacija modela se vrši nad test skupom podataka
- Kao mere za evaluaciju se koriste tačnost, f1 – mera, odziv, preciznost i matrica konfuzije

Evaluacija modela - rezultati

```
Tacnost modela:  0.9019337016574586
F1 skor modela:  0.9017181005939445
Odziv modela:  0.9019337016574586
Preciznost modela:  0.9067808826673667
Raspodela po klasama
+-----------------+-----+
|visit_in_worktime|count|
+-----------------+-----+
|                1| 1467|
|                0| 1429|
+-----------------+-----+

Matrica konfuzije:
[[1365.   64.]
 [ 220. 1247.]]
```

Skripta za pokretanje aplikacije

```
#!/bin/bash

spark/bin/spark-submit --master spark://spark-master:7077 app_3.py hdfs://namenode:9000/data_3.csv hdfs://namenode:9000/DecisionTreeModel_12
```

- Kao drugi argument navodimo putanju do skupa podataka koji će se koristiti za obučavanje i evaluaciju modela, dok je treći argument putanja do lokacije na HDFS-u gde želimo snimiti model
- Za obučavnje modela je iskorišćen data_3.csv skup podataka, koji sadrži 10000 slogova, i koji je kreiran na osnovu originalnog brightkite skupa podataka
- U data_3.csv se nalazi po 5000 instanci iz obe klase, da bi se uspostavila balansiranost

Web UI

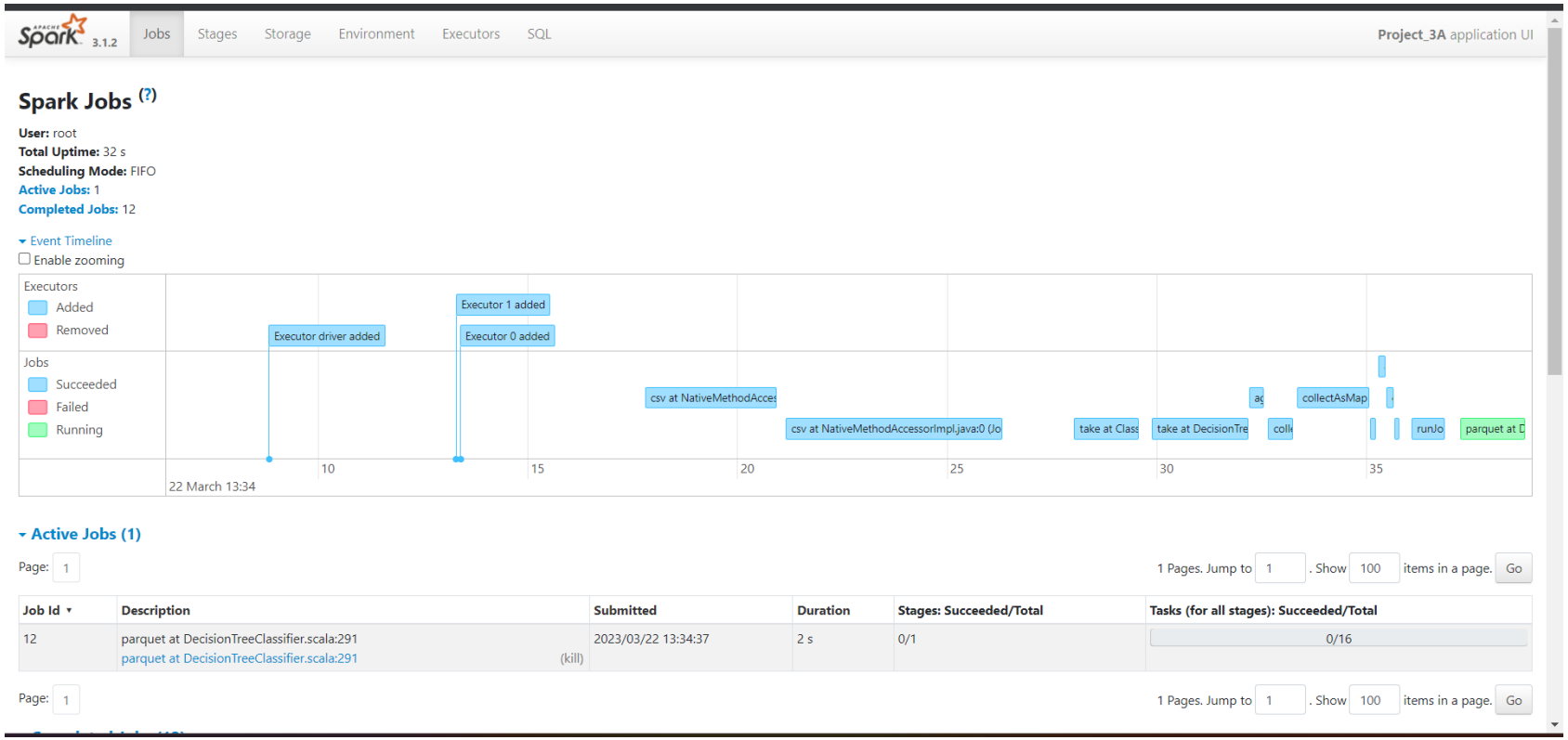
Application: Project_3A

ID: app-20230322133408-0003
Name: Project_3A
User: root
Cores: Unlimited (16 granted)
Executor Limit: Unlimited (2 granted)
Executor Memory: 1024.0 MiB
Executor Resources:
Submit Date: 2023/03/22 13:34:08
State: RUNNING
Application Detail UI

Executor Summary (2)

ExecutorID	Worker	Cores	Memory	Resources	State	Logs
1	worker-20230322125054-172.23.0.5-40587	8	1024		RUNNING	stdout stderr
0	worker-20230322125055-172.18.0.6-38403	8	1024		RUNNING	stdout stderr

Web UI


Spark 3.1.2
Jobs
Stages
Storage
Environment
Executors
SQL
Project_3A application UI
Spark Jobs (?)
User: root
Total Uptime: 32 s
Scheduling Mode: FIFO
Active Jobs: 1
Completed Jobs: 12
Event Timeline
Enable zooming
Executors
Added
Removed
Executor driver added
Executor 1 added
Executor 0 added
Jobs
Succeeded
Failed
Running
csv at NativeMethodAcces
csv at NativeMethodAccessorImpl.java:0 (Jo
take at Class
take at DecisionTre
collectAsMap
runJo
parquet at D
10
15
20
25
30
35
22 March 13:34
Active Jobs (1)
Page: 1
1 Pages. Jump to 1 . Show 100 items in a page. Go
Job Id
Description
Submitted
Duration
Stages: Succeeded/Total
Tasks (for all stages): Succeeded/Total
12
parquet at DecisionTreeClassifier.scala:291
parquet at DecisionTreeClassifier.scala:291 (kill)
2023/03/22 13:34:37
2 s
0/1
0/16
Page: 1
1 Pages. Jump to 1 . Show 100 items in a page. Go

Klasifikacija podataka sa Kafka topic-a (app_3b.py)

Čitanje i parsiranje podataka sa Kafka topic-a

```
df = spark \
  .readStream \
  .format("kafka") \
  .option("kafka.bootstrap.servers", "kafka:9092") \
  .option("subscribe", "topic2") \
  .option("startingOffsets", "earliest") \
  .option("maxOffsetsPerTrigger", 1) \
  .load()

parsed_df =  df.selectExpr("CAST(value AS STRING)")

split_col = split(parsed_df['value'], ',')
parsed_df = parsed_df.withColumn("user", split_col.getItem(0))
parsed_df = parsed_df.withColumn("check_in_time", split_col.getItem(1))
parsed_df = parsed_df.withColumn("latitude", split_col.getItem(2))
parsed_df = parsed_df.withColumn("longitude", split_col.getItem(3))
parsed_df = parsed_df.withColumn("location_id", split_col.getItem(4))
parsed_df = parsed_df.withColumn("time_spent", split_col.getItem(5))
parsed_df = parsed_df.withColumn("time_stamp", split_col.getItem(6))

parsed_df = parsed_df.where(parsed_df.user != 'user')
parsed_df = parsed_df.withColumn("check_in_time", to_timestamp(parsed_df["check_in_time"], "yyyy-MM-dd'T'HH:mm:ss.SSSXXX"))
parsed_df = parsed_df.withColumn('check_in_time', unix_timestamp('check_in_time'))

parsed_df = parsed_df.withColumn("user", parsed_df["user"].cast(IntegerType()))
parsed_df = parsed_df.withColumn("latitude", parsed_df["latitude"].cast(DoubleType()))
parsed_df = parsed_df.withColumn("longitude", parsed_df["longitude"].cast(DoubleType()))
```

Učitavanje modela i predikcije za podatke sa Kafka topic-a

```
# Ucitavanje modela
model_path = "hdfs://namenode:9000/DecisionTreeModel_5"
dt_model = DecisionTreeClassificationModel.load(model_path)

assembler = VectorAssembler(
    inputCols=["user", "check_in_time", "latitude", "longitude"],
    outputCol="features")

parsed_df = assembler.transform(parsed_df)

# Klasifikacija
predictions = dt_model.transform(parsed_df)
```

Upis rezultata u InfluxDB

```python
def write_to_influxdb(df, epoch_id):
    # Inicijalizacija InfluxDB klijenta
    token = "Xl99hGe7tyPW-wWrgpZRo8vOxA6bK2nR-X3MEoqkigZqnSG1vSqpKOoBmLZdWpWbYKKMKNEfqAAX4FMoKhd5ug=="
    org = "brightkite-org"
    bucket = "brightkite-bucket"
    client = InfluxDBClient(url="http://influxdb:8086", token=token)

    # Kreiranje instance WriteApi klase
    write_api = client.write_api(write_options=SYNCHRONOUS)

    for row in df.collect():
        point = Point("predictions_5b") \
            .field("user", row.user) \
            .field("latitude", row.latitude) \
            .field("longitude", row.longitude) \
            .field("visit_in_worktime", row.prediction)
        write_api.write(bucket=bucket, org=org, record=point)
```

Skripta za pokretanje

```
#!/bin/bash

spark/bin/spark-submit --master spark://spark-master:7077 --packages org.apache.spark:spark-sql-kafka-0-10_2.12:3.1.2 app_4.py hdfs://namenode:9000/DecisionTreeModel_5
```

- Kao drugi argument komande linije navodimo putanju do modela na HDFS-u

Web UI

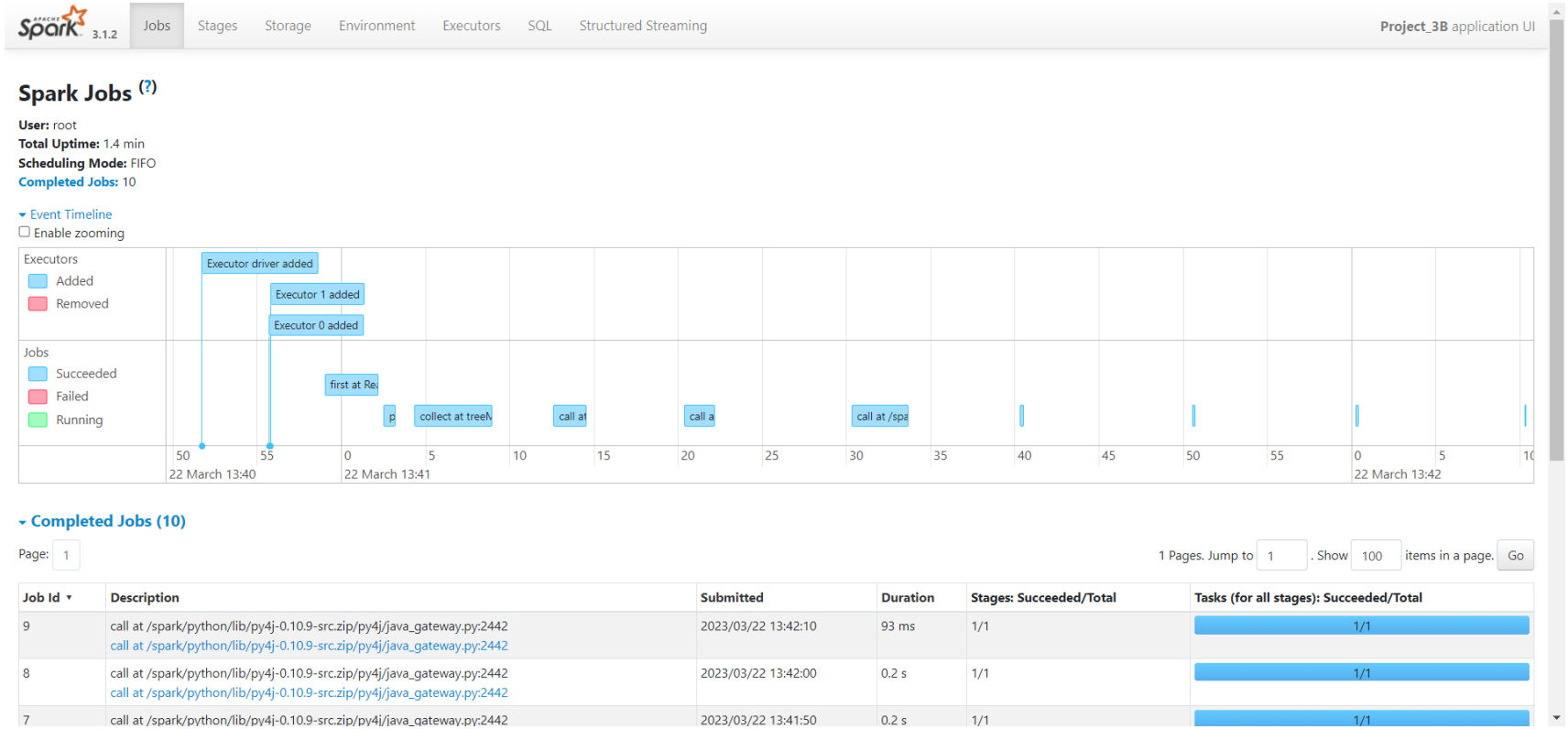
Application: Project_3B

ID: app-20230322134051-0005
Name: Project_3B
User: root
Cores: Unlimited (16 granted)
Executor Limit: Unlimited (2 granted)
Executor Memory: 1024.0 MiB
Executor Resources:
Submit Date: 2023/03/22 13:40:51
State: RUNNING
Application Detail UI

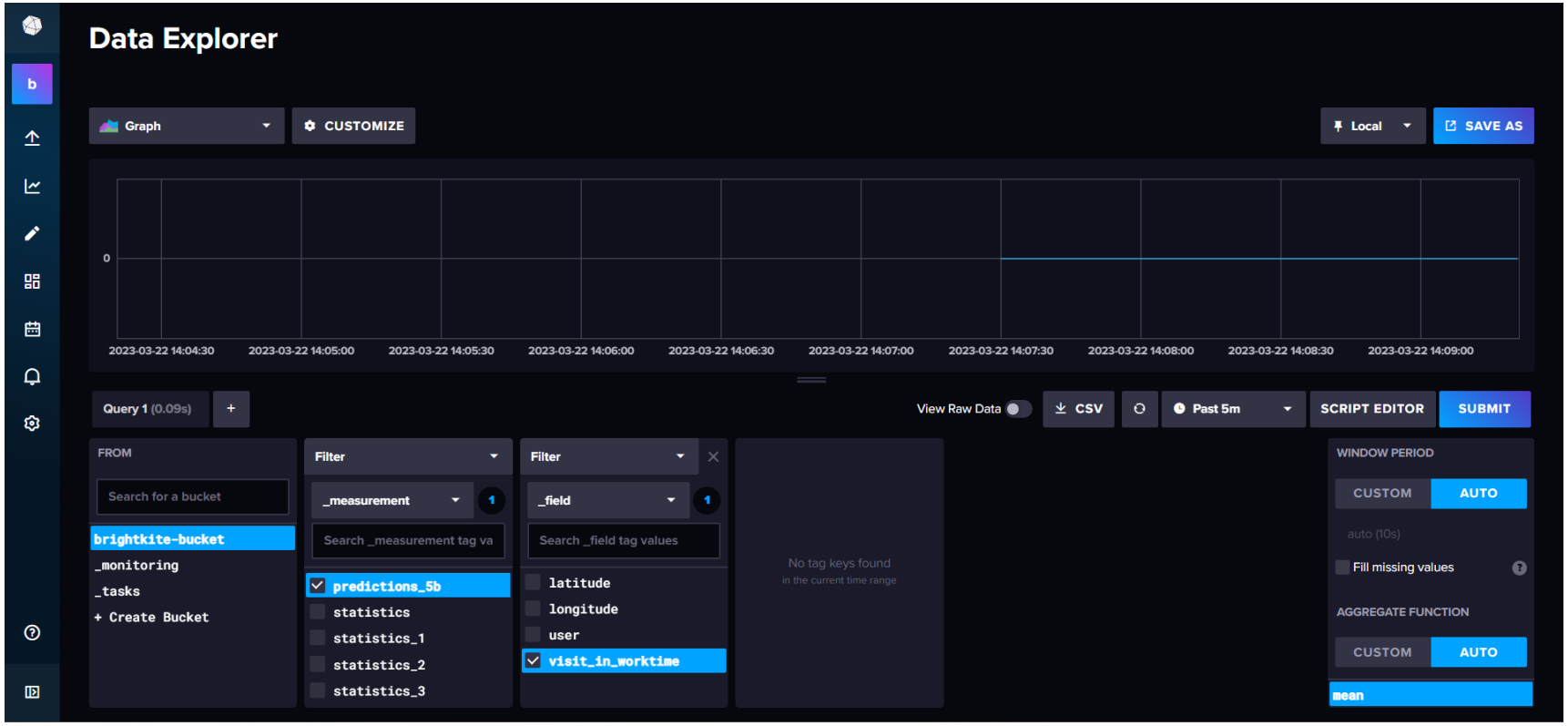
Executor Summary (2)

ExecutorID	Worker	Cores	Memory	Resources	State	Logs
1	worker-20230322125054-172.23.0.5-40587	8	1024		RUNNING	stdout stderr
0	worker-20230322125055-172.18.0.6-38403	8	1024		RUNNING	stdout stderr

Web UI

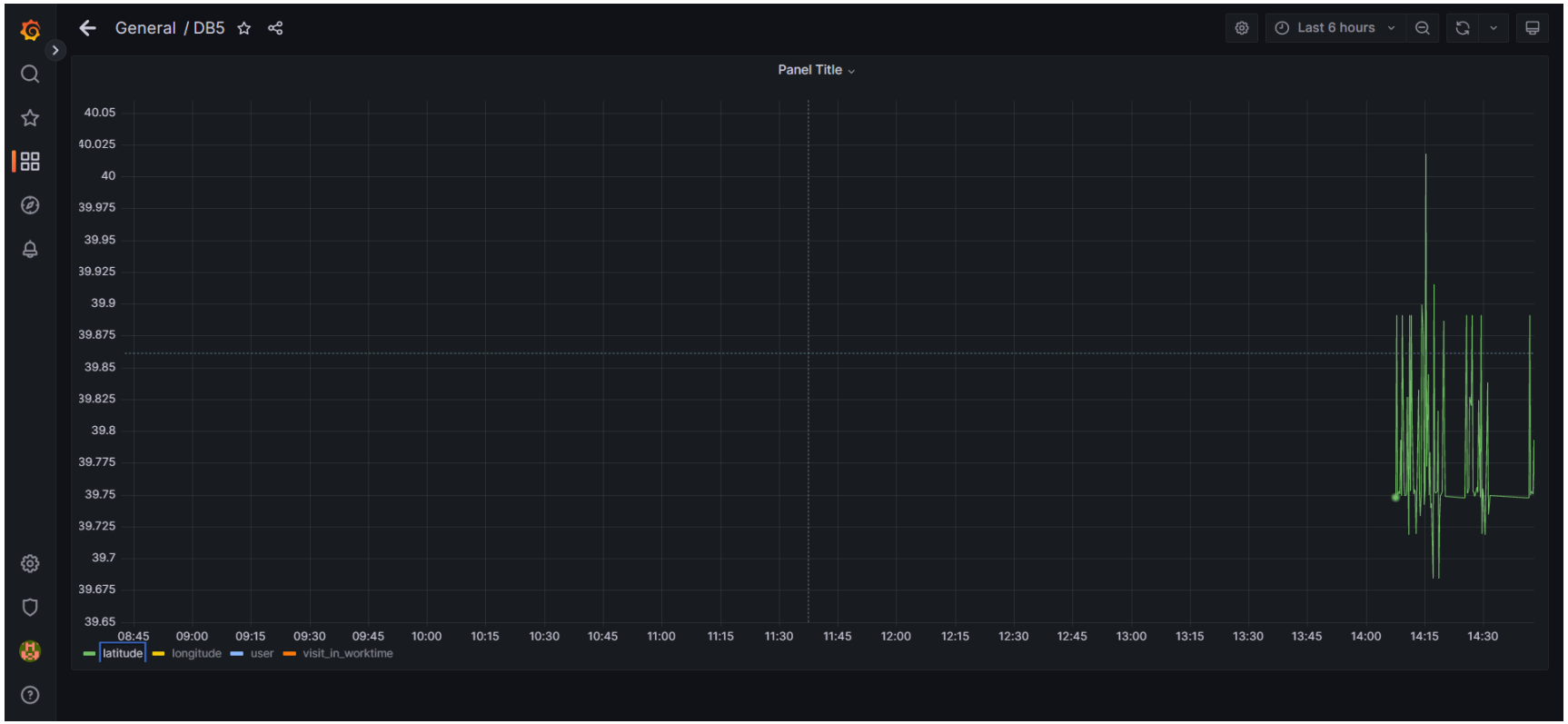

Spark 3.1.2
Jobs
Stages
Storage
Environment
Executors
SQL
Structured Streaming
Project_3B application UI
Spark Jobs (?)
User: root
Total Uptime: 1.4 min
Scheduling Mode: FIFO
Completed Jobs: 10
Event Timeline
Enable zooming
Executors
Added
Removed
Executor driver added
Executor 1 added
Executor 0 added
Jobs
Succeeded
Failed
Running
first at Re
collect at treeN
call at
call a
call at /spa
50
55
0
5
10
15
20
25
30
35
40
45
50
55
0
5
22 March 13:40
22 March 13:41
22 March 13:42
Completed Jobs (10)
Page: 1
1 Pages. Jump to 1 . Show 100 items in a page. Go
Job Id
Description
Submitted
Duration
Stages: Succeeded/Total
Tasks (for all stages): Succeeded/Total
9
call at /spark/python/lib/py4j-0.10.9-src.zip/py4j/java_gateway.py:2442
call at /spark/python/lib/py4j-0.10.9-src.zip/py4j/java_gateway.py:2442
2023/03/22 13:42:10
93 ms
1/1
1/1
8
call at /spark/python/lib/py4j-0.10.9-src.zip/py4j/java_gateway.py:2442
call at /spark/python/lib/py4j-0.10.9-src.zip/py4j/java_gateway.py:2442
2023/03/22 13:42:00
0.2 s
1/1
1/1
7
call at /spark/python/lib/py4j-0.10.9-src.zip/py4j/java_gateway.py:2442
2023/03/22 13:41:50
0.2 s
1/1
1/1

Rezultat u InfluxDB



Vizuelizacija rezultata u Grafani

Vizuelizacija rezultata u Grafani



Vizuelizacija rezultata u Grafani