SISTEMI ZA OBRADU I ANALIZU VELIKE KOLIČINE PODATAKA - Projekat 1

Filip Trajković 1574

KORIŠĆENI PROJEKTI

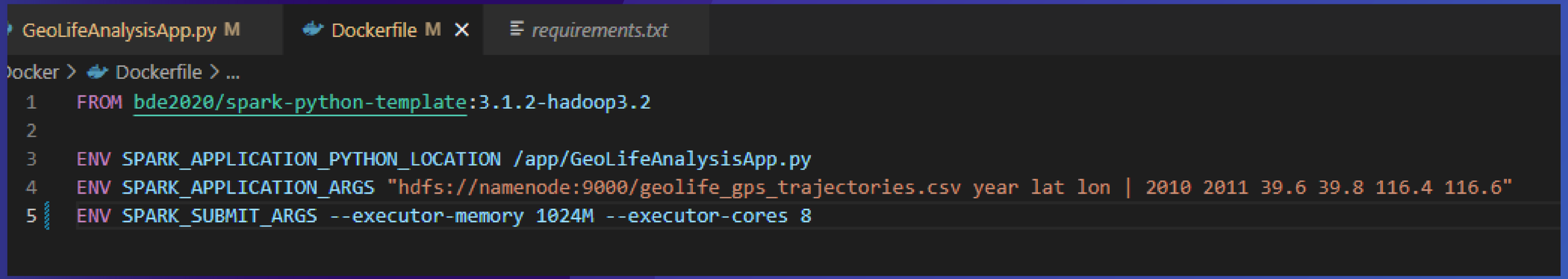
Apache Hadoop

Apache Spark

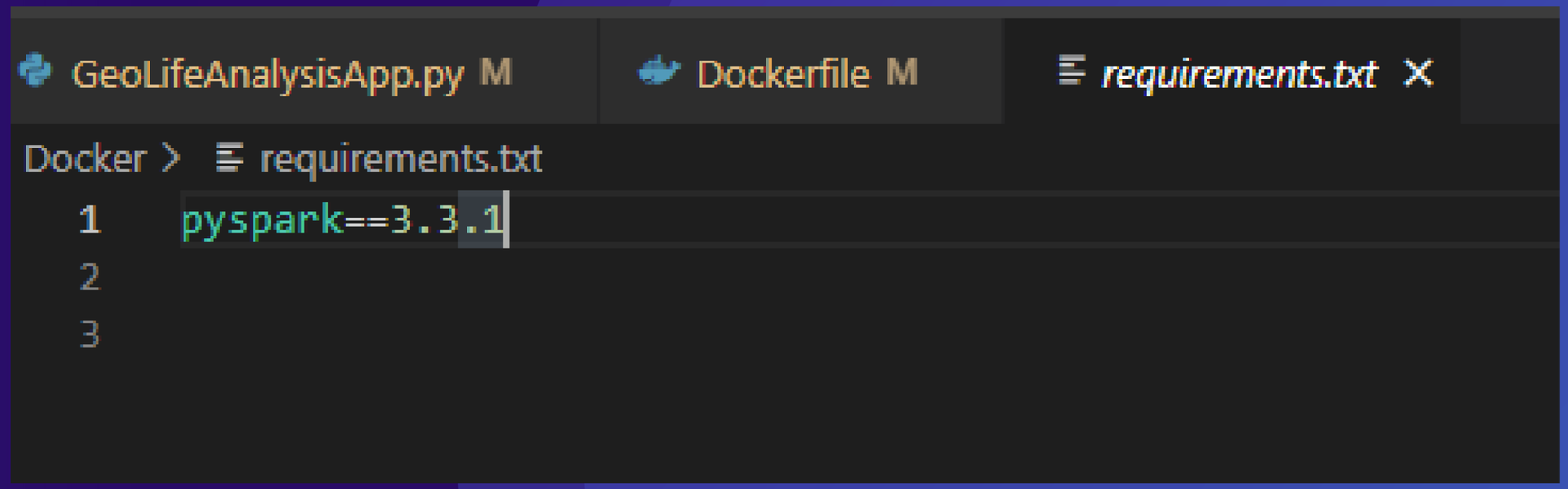
Spisak kontejnera

Name	Image
GeoLifeAnalysisApp ccbeb15dba4a	bde/spark-app
spark 5a4d90dfd384	bde2020/spark-base:3.1.2-hadoop3.2
verzija312	-
spark-worker-2 049df6ab4956	bde2020/spark-worker:3.1.2-hadoop3.2
spark-worker-1 a5226974beed	bde2020/spark-worker:3.1.2-hadoop3.2
resourcemanager c3db757afc50	bde2020/hadoop-resourcemanager:2.0.0-hadoop3.2.1-java8
historyserver 67ce0252265b	bde2020/hadoop-historyserver:2.0.0-hadoop3.2.1-java8
namenode 2e1963fc1b41	bde2020/hadoop-namenode:2.0.0-hadoop3.2.1-java8
nodemanager 76a098f9f2e9	bde2020/hadoop-nodemanager:2.0.0-hadoop3.2.1-java8
datanode 4fedf77ca4e4	bde2020/hadoop-datanode:2.0.0-hadoop3.2.1-java8
spark-master fb3845ff4535	bde2020/spark-master:3.1.2-hadoop3.2

Izgled Dockerfile-a



Izgled requirements.txt fajla →



Funkcije za filter podataka

```python
def filterDataFrameByLatitude(df, latMin, latMax):
    print("filterDataFrameByLatitude entered")
    return df.filter((df.lat >= latMin) & (df.lat <= latMax))

def filterDataFrameByLongitude(df, lonMin, lonMax):
    print("filterDataFrameByLongitude entered")
    return df.filter((df.lon >= lonMin) & (df.lon <= lonMax))

def filterDataFrameByAltitude(df, altMin, altMax):
    return df.filter((df.alt >= altMin) & (df.alt <= altMax))

def filterDataFrameByYear(df, yearMin, yearMax):
    print("filterDataFrameByYear entered")
    return df.filter((df.year >= yearMin) & (df.year <= yearMax))

def filterDataFrameByMonth(df, monthMin, monthMax):
    return df.filter((df.month >= monthMin) & (df.month <= monthMax))

def filterDataFrameByDay(df, dayMin, dayMax):
    return df.filter((df.day >= dayMin) & (df.day <= dayMax))

def filterDataFrameByHour(df, hourMin, hourMax):
    return df.filter((df.hour >= hourMin) & (df.hour <= hourMax))

def filterDataFrameByMinute(df, minuteMin, minuteMax):
    return df.filter((df.minute >= minuteMin) & (df.minute <= minuteMax))

def filterDataFrameBySecond(df, secondMin, secondMax):
    return df.filter((df.second >= secondMin) & (df.second <= secondMax))
```

Funkcija za učitavanje nepoznatog broja parametara

```
40  def readInputValues():
41      delimiterFound = False
42      features = {}
43      for arg in sys.argv[2:]:
44          if arg == "|":
45              delimiterFound = True
46              continue
47          if not delimiterFound:
48              if arg not in features.keys():
49                  features[arg] = []
50          else:
51              for key in features.keys():
52                  if (len(features[key]) != 2):
53                      features[key].append(float(arg))
54                      break
55      return features
56
```

Primena filter funkcija u zavisnosti od ulaznih podataka

```
57  def findDfBasedOnInputValues(inputDictionary, inputDf):
58          #tmpDf = inputDf.copy()
59          tmpDf = inputDf.alias('tmpDf')
60          tmpDf.show(20)
61          switcher = {
62                  "lat": filterDataFrameByLatitude,
63                  "lon": filterDataFrameByLongitude,
64                  "alt": filterDataFrameByAltitude,
65                  "year": filterDataFrameByYear,
66                  "month": filterDataFrameByMonth,
67                  "day": filterDataFrameByDay,
68                  "hour": filterDataFrameByHour,
69                  "minute": filterDataFrameByMinute,
70                  "second": filterDataFrameBySecond,
71              }
72          for key in inputDictionary.keys():
73              method = switcher.get(key, "nothing")
74              if (method != "nothing"):
75                  tmpDf = method(tmpDf, inputDictionary[key][0], inputDictionary[key][1])
76                  printLine()
77                  print("Filtered by key: " + key)
78                  tmpDf.show(20)
79          return tmpDf
```

Setup aplikacije

```
86  if __name__ == '__main__':
87      if len(sys.argv) < 2:
88          print("Usage: main.py <input folder> ")
89          exit(-1)
90
91      appName = "GeoLifeGpsAnalysis"
92
93      conf = SparkConf()
94      conf.setMaster("spark://spark-master:7077")
95      #conf.setMaster("local")
96      conf.set("spark.driver.memory","4g")
97
98      spark = SparkSession.builder.config(conf=conf).appName(appName).getOrCreate()
99      spark.sparkContext.setLogLevel("ERROR")
100
101     geoLifeSchema = StructType() \
102                         .add("time", "string")\
103                         .add("lat", "float")\
104                         .add("lon", "float")\
105                         .add("alt", "float")\
106                         .add("label", "string")\
107                         .add("user", "integer")
108
109     geoLifeDataFrame = spark.read.option("header", True).csv(sys.argv[1], schema=geoLifeSchema)
```

Pretprocesiranje podataka

```
114    splitDFTime = geoLifeDataFrame.withColumn("date",split(col("time")," ").getItem(0))\
115        .withColumn("exacttime",split(col("time")," ").getItem(1))\
116        .drop("time")
117
118    splitDFYear = splitDFTime.withColumn("year",split(col("date"),"-").getItem(0))\
119        .withColumn("month",split(col("date"),"-").getItem(1))\
120        .withColumn("day",split(col("date"),"-").getItem(2))\
121        .drop("date")
122
123
124    splitDFTime = splitDFYear.withColumn("hour",split(col("exacttime"),":").getItem(0))\
125        .withColumn("minute",split(col("exacttime"),":").getItem(1))\
126        .withColumn("second",split(col("exacttime"),":").getItem(2))\
127        .drop("exacttime")
128
129    df = splitDFTime.withColumn('hour', F.regexp_replace('hour', r'^[0]*', ''))
130    df = df.withColumn('minute', F.regexp_replace('minute', r'^[0]*', ''))
131    df = df.withColumn('second', F.regexp_replace('second', r'^[0]*', ''))
132    df = df.withColumn('day', F.regexp_replace('day', r'^[0]*', ''))
133    df = df.withColumn('month', F.regexp_replace('month', r'^[0]*', ''))
134
135    cols = ["year", "month", "day", "hour", "minute", "second"]
136    for col_name in cols:
137        df = df.withColumn(col_name, col(col_name).cast('float'))
138
```

Prikaz vozača i vozila na osnovu ulaznih podataka →

```
142
143     inputValuesDictionary = readInputValues()
144     print(inputValuesDictionary)
145
146     filteredDf = findDfBasedOnInputValues(inputValuesDictionary, df)
147     filteredDf.printSchema()
148     filteredDf.show(25)
149     filteredDf.describe().show()
150
151     df.select('user').distinct().sort('user').show(2000)
152
153     df.select('label').distinct().show(2000)
154
```

Izgled pretprocesiranih podataka

→

time	lat	lon	alt	label	user
2007-08-04 03:30:32	39.92171	116.47234	13.0	no_data	10
2007-08-04 03:30:33	39.921703	116.47234	13.0	no_data	10
2007-08-04 03:30:34	39.921696	116.47234	13.0	no_data	10
2007-08-04 03:30:35	39.921684	116.47234	13.0	no_data	10
2007-08-04 03:30:36	39.921673	116.47234	13.0	no_data	10
2007-08-04 03:30:38	39.92158	116.47231	13.0	no_data	10
2007-08-04 03:30:39	39.921574	116.47231	13.0	no_data	10
2007-08-04 03:30:40	39.92156	116.47229	13.0	no_data	10
2007-08-04 03:30:41	39.921566	116.47229	13.0	no_data	10
2007-08-04 03:30:42	39.92157	116.47229	13.0	no_data	10
2007-08-04 03:30:43	39.921577	116.4723	13.0	no_data	10
2007-08-04 03:30:44	39.92158	116.472305	13.0	no_data	10
2007-08-04 03:30:45	39.921562	116.472305	13.0	no_data	10
2007-08-04 03:30:46	39.921543	116.472305	13.0	no_data	10
2007-08-04 03:30:47	39.92153	116.47231	13.0	no_data	10
2007-08-04 03:30:49	39.921505	116.47232	13.0	no_data	10
2007-08-04 03:30:50	39.921494	116.47232	13.0	no_data	10
2007-08-04 03:30:52	39.921486	116.47232	13.0	no_data	10
2007-08-04 03:30:57	39.921486	116.47232	13.0	no_data	10
2007-08-04 03:30:58	39.921486	116.47232	13.0	no_data	10
2007-08-04 03:30:59	39.921482	116.47231	13.0	no_data	10
2007-08-04 03:31:01	39.921463	116.472305	13.0	no_data	10

Izgled postprocesiranih podataka

→

lat	lon	alt	label	user	year	month	day	hour	minute	second
39.79995	116.47387	0.0	subway	65	2011.0	8.0	30.0	11.0	28.0	41.0
39.79994	116.47388	0.0	subway	65	2011.0	8.0	30.0	11.0	28.0	41.0
39.799957	116.47386	0.0	subway	65	2011.0	8.0	30.0	11.0	28.0	41.0
39.79998	116.47384	0.0	subway	65	2011.0	8.0	30.0	11.0	28.0	41.0
39.79997	116.47385	0.0	subway	65	2011.0	8.0	30.0	11.0	28.0	41.0
39.799892	116.473915	0.0	subway	65	2011.0	8.0	30.0	11.0	28.0	42.0
39.799885	116.47392	0.0	subway	65	2011.0	8.0	30.0	11.0	28.0	42.0
39.799904	116.47391	0.0	subway	65	2011.0	8.0	30.0	11.0	28.0	42.0
39.799927	116.473885	0.0	subway	65	2011.0	8.0	30.0	11.0	28.0	42.0
39.799915	116.4739	0.0	subway	65	2011.0	8.0	30.0	11.0	28.0	42.0
39.79984	116.47395	0.0	subway	65	2011.0	8.0	30.0	11.0	28.0	43.0
39.799828	116.47396	0.0	subway	65	2011.0	8.0	30.0	11.0	28.0	43.0
39.79985	116.473946	0.0	subway	65	2011.0	8.0	30.0	11.0	28.0	43.0
39.799873	116.47393	0.0	subway	65	2011.0	8.0	30.0	11.0	28.0	43.0
39.799862	116.47394	0.0	subway	65	2011.0	8.0	30.0	11.0	28.0	43.0
39.79978	116.47399	0.0	subway	65	2011.0	8.0	30.0	11.0	28.0	44.0
39.799767	116.47401	0.0	subway	65	2011.0	8.0	30.0	11.0	28.0	44.0
39.799793	116.47398	0.0	subway	65	2011.0	8.0	30.0	11.0	28.0	44.0
39.799816	116.47397	0.0	subway	65	2011.0	8.0	30.0	11.0	28.0	44.0
39.799805	116.47398	0.0	subway	65	2011.0	8.0	30.0	11.0	28.0	44.0
39.799717	116.47404	0.0	subway	65	2011.0	8.0	30.0	11.0	28.0	45.0
39.799706	116.474045	0.0	subway	65	2011.0	8.0	30.0	11.0	28.0	45.0
39.799732	116.47403	0.0	subway	65	2011.0	8.0	30.0	11.0	28.0	45.0
39.799755	116.474014	0.0	subway	65	2011.0	8.0	30.0	11.0	28.0	45.0
39.799744	116.47402	0.0	subway	65	2011.0	8.0	30.0	11.0	28.0	45.0

Deskriptivna analiza podataka

```
+-------+------------------+------------------+-----------------+------+-----------------+-----------------+-
-+
|summary|               lat|               lon|              alt| label|             user|             year|
d|
+-------+------------------+------------------+-----------------+------+-----------------+-----------------+-
-+
|  count|              8946|              8946|             8946|  8946|             8946|             8946|
6|
|   mean| 39.774831096837644|116.45598258434886|55.146434160518666|  null|123.43729040912139|2010.5560026827632|
7|
| stddev|0.040842975761083845|0.02316312741696902|48.147154453540956|  null| 44.97862024869981|0.4968815728897937|2
5|
|    min|         39.600025|         116.40004|             -3.0|no_data|               65|           2010.0|
0|
|    max|              39.8|         116.52452|            186.0|  walk|              163|           2011.0|
0|
+-------+------------------+------------------+-----------------+------+-----------------+-----------------+-
-+
```

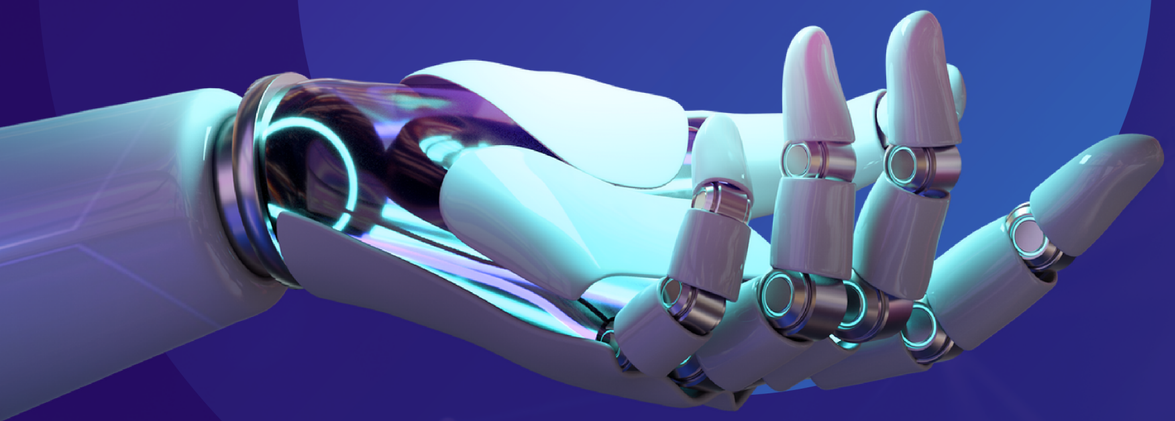
Rezultat

Completed Applications (2)

Application ID	Name	Cores	Memory per Executor
app-20230127005131-0001	GeoLifeGpsAnalysis	16	1024.0 MiB
app-20230127002326-0000	GeoLifeGpsAnalysis	16	1024.0 MiB

	Submitted Time	User	State	Duration
	2023/01/27 00:51:31	root	FINISHED	34 s
	2023/01/27 00:23:26	root	FINISHED	34 s

SCAN ME



HVALA NA PAŽNJI

ficax.al@elfak.rs