CS 5433: Bigdata Management
Programming Assignment 3
Task1 – Data Correction

Group 4

Description of Dataset:

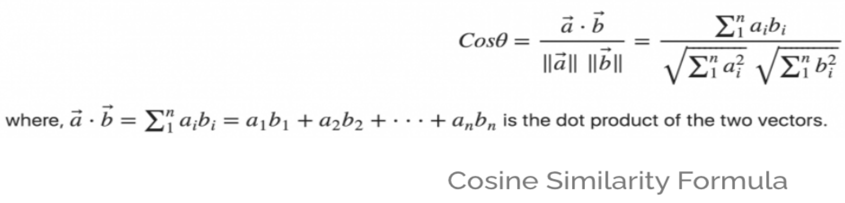
Our dataset("/user/kaggle/kaggle_data/IndianUniversityRankingFrom2017to2021.csv") contains the data related to the rankings of Indian universities from the years 2017 to 2021 where the data consists of 10 columns and 500 records without any null values. The columns of the dataset are,

- Institute ID which is a "String" column
- Name – Name of the university/institute of type "String"
- City – Name of the city where university is located, which is of type "String"
- State – Name of the State where university is located, which is of type "String"
- PR Score – PR Score of the university which is of type "Double"
- PR Rank – PR Rank of the university which is of type "Integer"
- PR Score – PR Score of the university which is of type "Double"
- Score – Score of the university which is of type "Double"
- Year –Year (contains values 2017,2018,2019,2020 & 2021) is of type "Integer"
- Rank – Rank of the university which is of type "Integer"

➔ As we don't contain any null values in our dataset, we have inserted some null values in the column "PR Rank".

Cosine Similarity:

Cosine Similarity is described as a type of similarity measure which is used to measure how similar the data frames are which is irrespective of their size. In the terms of mathematics, it describes the cosine of angle between the formed vectors which are projected in a multi-dimensional space. This similarity measure is very advantageous because if the angle between them is smaller then there is higher similarity. The formula for cosine similarity is described below:



$$Cos\theta = \frac{\vec{a} \cdot \vec{b}}{\|\vec{a}\| \ \|\vec{b}\|} = \frac{\sum_1^n a_i b_i}{\sqrt{\sum_1^n a_i^2} \sqrt{\sum_1^n b_i^2}}$$

where, $\vec{a} \cdot \vec{b} = \sum_1^n a_i b_i = a_1 b_1 + a_2 b_2 + \cdots + a_n b_n$ is the dot product of the two vectors.

Cosine Similarity Formula

Fig 1,1: Cosine similarity formula

Approach:

There are no null values in the given dataset “IndianUniversityRankingFrom2017to2021.csv”, So as per the instructions we copied this .csv file from given path to local and then we created “113” null values in “PR Rank” column.

Original path:

/user/kaggle/kaggle_data/ IndianUniversityRankingFrom2017to2021.csv

After creating null values, the modified .CSV file is uploaded to below path

Modified file path:

/user/sdarapu/Group4_DataSet/IndianUniversityRankingFrom2017to2021.csv

The file which is modified by placing null values is taken as input file for the data correction task.

At first we created “Assign3_Group4_Task1.py” in the Hadoop cluster. Please refer to “Group_4_Program_1.pdf” for full code and author comments.

Code Explanation:

a) Imported the required libraries

- from pyspark.sql import SparkSession → This library is imported to create a sparksession. This sparksession can be used to create a dataframe.
- from pyspark.ml.feature import StringIndexer,VectorAssembler → StringIndexer library is imported for converting the String columns into Double type and VectorAssembler is imported for merging multiple columns into a vector column.
- from pyspark.ml import Pipeline – Pipeline is imported to run the stages in sequence.
- from pyspark.sql.types import * - It is imported for using the pyspark sql datatypes.
- from sklearn.metrics.pairwise import cosine_similarity – It is imported to calculate cosine similarity between two samples.
- import pandas as pd – Import python pandas for further actions.
- import numpy as np – Import numPy and give an alias name.

b) Created Spark Session

spark = SparkSession.builder.appName("Assign3_Group4_Task1").getOrCreate() → Here, we provided the name to our application by setting a string “Assign3_Group4_Task1” to.appName() as a parameter. Next, used .getOrCreate() to create and instantiate SparkSession into our object “spark”.

c) Reading csv file into PySpark DataFrame

data=spark.read.csv("hdfs://hadoop-nn001.cs.okstate.edu:9000/user/sdarapu/Group4_DataSet/IndianUniversitiesRankingFrom2017to2021.csv ", header = True, inferSchema = True) → By using spark.read.csv() method, we first passed the given csv file location and we used "inferSchema" attribute and set its value as True which will automatically take schema from the given file into Pyspark Dataframe.

d) Check if there are any null values in DataFrame

```
data.select([count(when(col(c).isNull(), c)).alias(c) for c in data.columns]).show()
```

By using above statement we checked if there are any null values. If present it will display the null values existed in spark dataframe.

e) Function which changes the column labels to indices labels by using String Indexer method

```
def getIndex(ar_1, ar_2):
  return StringIndexer(inputCol=ar_1, outputCol=ar_2)
```

It will take arguments where one is the input column in the csv file and another is the output column.

f) Columns which are passed to the above function, and they are converted to string indexers

```
InsID_indexer=getIndex("Institute ID", "INSTITUTE ID" )
data=InsID_indexer.fit(data).transform(data)

Name_indexer= getIndex("Name", "NAME" )
data=Name_indexer.fit(data).transform(data)

State_indexer= getIndex("State", "STATE" )
data=State_indexer.fit(data).transform(data)

City_indexer= getIndex("City", "CITY" )
data=City_indexer.fit(data).transform(data)
```

All String Indexers are passed to fit and transform method and stored in dataframe.

g) Convert spark dataframe to pandas dataframe for further operations

```
df=data.toPandas()
```

h) Replace out of range values in Score Column with -1 so that they can be treated as null values

```
df['Score'] = np.where((df['Score']>=65)|(df['Score']<=35),-1,df['Score'])
```

Used numpy to replace the Score values which are less than 35 and greater than 65.

i) Replace all NaN values with -1 for PR Rank column

```
df['PR Rank'] = df['PR Rank'].replace(np.nan,-1)
```

Used replace method with -1 so that they can be identifiable.

j) Check if there are any null values in data frame after replacing with -1

```
df2=df[df.isna().any(axis=1)]
```

isna() method will check if there are any NaN values in dataframe and they are stored in df2.

k) Extract all the rows with -1 and store in another dataframe

```
df_null = df[df.values == -1]
```

Check if all values in a row contains -1, If it is there then extract that row and place it in df_null data frame.

l) Drop the rows from main dataframe(df) which are in df_null

```
df.drop(df_null.index, axis = 0, inplace = True)
```

Drop the rows from main data frame which are placed in df_null.

m) Calculating cosine similarity for the dataframes df_null,df

```
cosinesimilarity = cosine_similarity(df_null,df)
```

calculate cosine similarity for both data frames and store the result in the variable.

n) Identifying the indexes which are having high similarity, and these are used to fetch from main data frame

```
rows_list = []
c = 0
## method checks the similarity
def checkSim(c_s, index):
  ## returns the maximum similarity values by iterating through indexes
  return np.where(c_s[index]==np.max(c_s[index]))

#iterating all the values in the cosine similarity values
while c < len(cosinesimilarity):
  #pass the values to above function and store the value in variable
  max_val = checkSim(cosinesimilarity, c)
  #check the value is empty or not
  if max_val[0][0] is not empty :
    #append those values into one list
    rows_list.append(max_val[0][0])
  #increment the values to go to next value
  c = c+1
#print the values in list
print(rows_list)
```

This method checks for the maximum similar values which are obtained after performing cosine similarity method and appends to a list named rows_list. This list contains the indexes which are to be placed in df_null.

o) Place all the indexes of df_null in a list for which we have to replace values.

```
rplcr_list = df_null.index
list_replacer = rplcr_list.tolist()
print(list_replacer)
```

All the index in df_null data frame are passed to list. This list should be iterated in next steps to fill those values which are having -1.

p) Loop used to replace the values in df_null and appending them to main dataframe(df) after replacement.

```
i = 0
j=0
indexx = 0
#loop which citerates the values in list_replacer
while indexx < len(list_replacer):
  #getting the indexes of rows containing null values along with columns
  col_tp = df_null[df_null.index == list_replacer[indexx]]
  #Identifying and storimg column names to be replaced
  colname = (col_tp.columns[(col_tp == -1).iloc[0]]).tolist()[0]
  #fetching the index value from clean dataset and stores in f
  f=rows_list[i]
  #fetching the value at particular row and obtained column and store in h
  h=df.at[f,colname]
  #place the value in dataframe contain null and column
  df_null.at[list_replacer[indexx],colname]=h
  #fetch the inserted row from df_null and store it in another dataframe
  d2=df_null.iloc[j]
  #append all the newly inserted rows to original dataframe
  df=df.append(d2,ignore_index= True)
  #incrementing the values to get correct indexes
  indexx = indexx +1
  i=i+1
  j=j+1
```

This loop is used to replace values in df_null based on the similarity indices picked from rows_list and after they are appended to main dataframe(df). After all this dataframe shouldn’t have any null values.

q) Convert pandas dataframe to spark dataframe

```
sparkdf=spark.createDataFrame(df)
```

r) Check spark dataframe contain any null values

```
sparkdf.select([count(when(col(c).isNull(),c)).alias(c)
for c in sparkdf.columns]).show()
```

s) Save the output to .CSV file

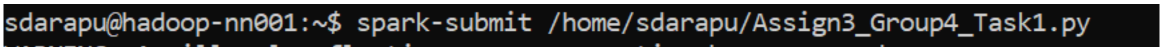
- sparkdf.coalesce(1).write.option("header","true").csv("hdfs:////user/sdarapu/Assign3_Group4_Task1_Output")
- sparkdf.coalesce(1).write.option("header","true").csv("hdfs:////user/sdarapu/Assign3_Group4_Task1_Output_inpfor_Task2-4")

Note: Here we are saving the output into two folders:

- user/sdarapu/Assign3_Group4_Task1_Output → This can be used to check the resultant data
- /user/sdarapu/Assign3_Group4_Task1_Output_inpfor_Task2-4 → The resultant data file stored in this folder is used as input for the remaining tasks 2-4. "You may get this folder already exists error while running this task, please ignore it". As we already storing the data in the above specified folder ("Group4_task1_Output")

Steps to Execute the Code:

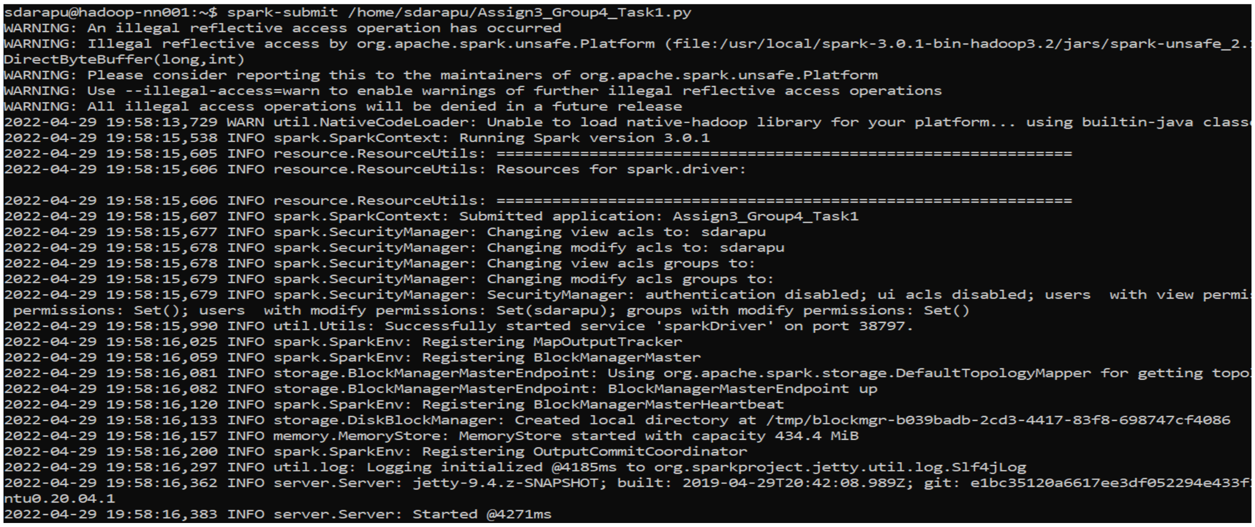
i. To run the code, we have executed below command as shown below



```
sdarapu@hadoop-nn001:~$ spark-submit /home/sdarapu/Assign3_Group4_Task1.py
```

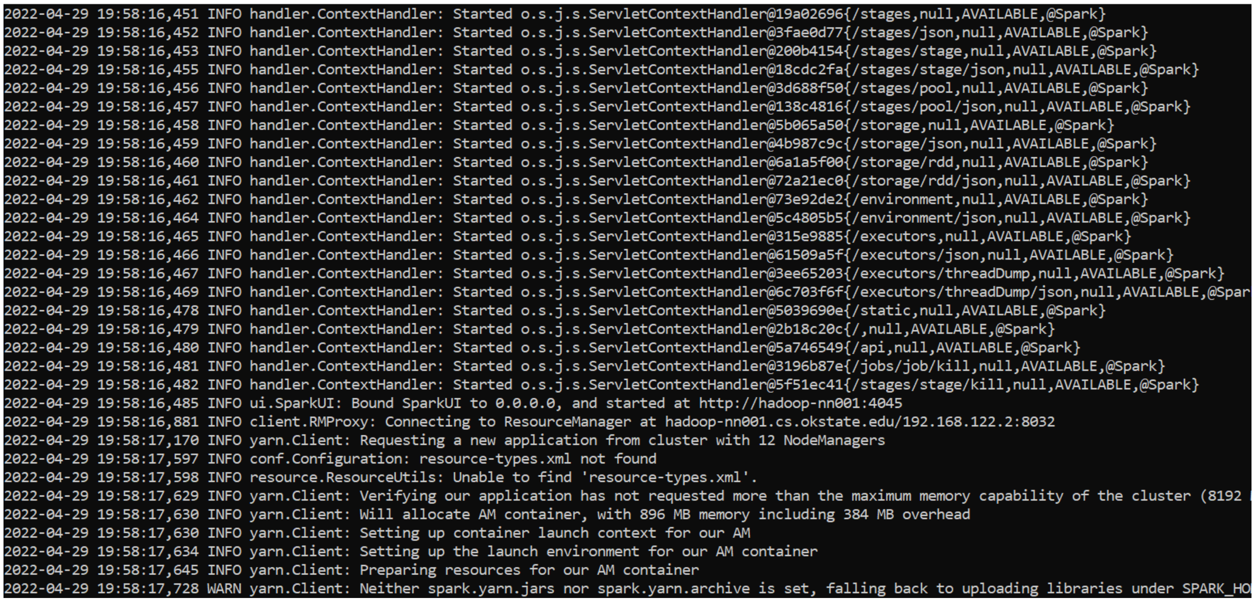
Fig 1,2: Command to execute

ii. Above command execute as follows



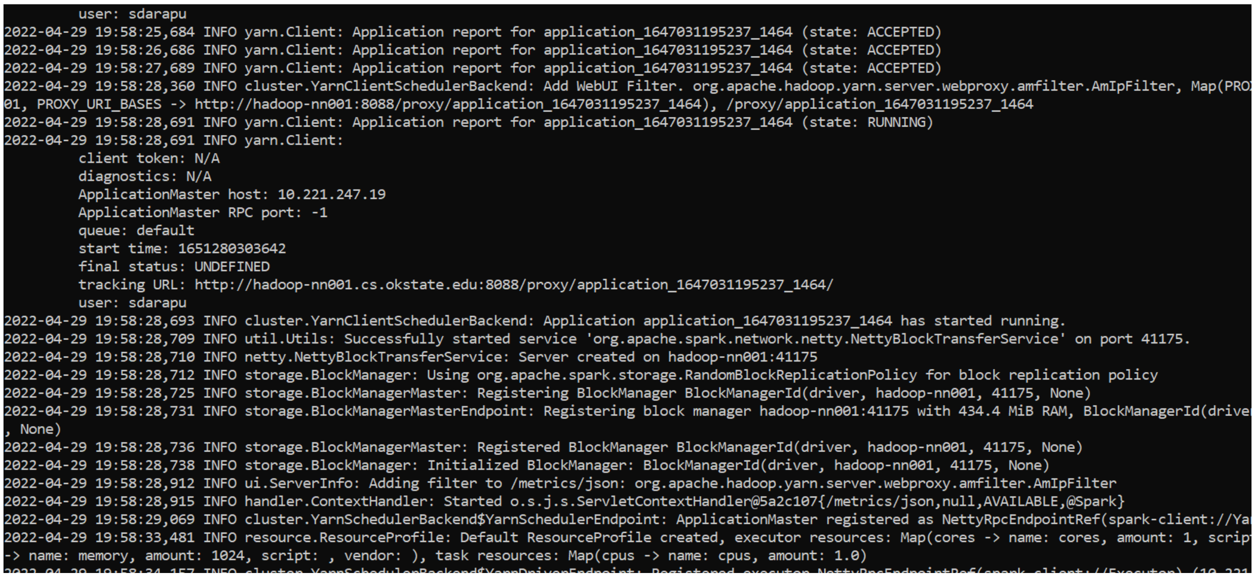
```
sdarapu@hadoop-nn001:~$ spark-submit /home/sdarapu/Assign3_Group4_Task1.py
WARNING: An illegal reflective access operation has occurred
WARNING: Illegal reflective access by org.apache.spark.unsafe.Platform (file:/usr/local/spark-3.0.1-bin-hadoop3.2/jars/spark-unsafe_2.
DirectByteBuffer(long,int)
WARNING: Please consider reporting this to the maintainers of org.apache.spark.unsafe.Platform
WARNING: Use --illegal-access=warn to enable warnings of further illegal reflective access operations
WARNING: All illegal access operations will be denied in a future release
2022-04-29 19:58:13,729 WARN util.NativeCodeLoader: Unable to load native-hadoop library for your platform... using builtin-java classe
2022-04-29 19:58:15,538 INFO spark.SparkContext: Running Spark version 3.0.1
2022-04-29 19:58:15,605 INFO resource.ResourceUtils: ==============================================================
2022-04-29 19:58:15,606 INFO resource.ResourceUtils: Resources for spark.driver:

2022-04-29 19:58:15,606 INFO resource.ResourceUtils: ==============================================================
2022-04-29 19:58:15,607 INFO spark.SparkContext: Submitted application: Assign3_Group4_Task1
2022-04-29 19:58:15,677 INFO spark.SecurityManager: Changing view acls to: sdarapu
2022-04-29 19:58:15,678 INFO spark.SecurityManager: Changing modify acls to: sdarapu
2022-04-29 19:58:15,678 INFO spark.SecurityManager: Changing view acls groups to:
2022-04-29 19:58:15,679 INFO spark.SecurityManager: Changing modify acls groups to:
2022-04-29 19:58:15,679 INFO spark.SecurityManager: SecurityManager: authentication disabled; ui acls disabled; users  with view permi
 permissions: Set(); users  with modify permissions: Set(sdarapu); groups with modify permissions: Set()
2022-04-29 19:58:15,990 INFO util.Utils: Successfully started service 'sparkDriver' on port 38797.
2022-04-29 19:58:16,025 INFO spark.SparkEnv: Registering MapOutputTracker
2022-04-29 19:58:16,059 INFO spark.SparkEnv: Registering BlockManagerMaster
2022-04-29 19:58:16,081 INFO storage.BlockManagerMasterEndpoint: Using org.apache.spark.storage.DefaultTopologyMapper for getting topo
2022-04-29 19:58:16,082 INFO storage.BlockManagerMasterEndpoint: BlockManagerMasterEndpoint up
2022-04-29 19:58:16,120 INFO spark.SparkEnv: Registering BlockManagerMasterHeartbeat
2022-04-29 19:58:16,133 INFO storage.DiskBlockManager: Created local directory at /tmp/blockmgr-b039badb-2cd3-4417-83f8-698747cf4086
2022-04-29 19:58:16,157 INFO memory.MemoryStore: MemoryStore started with capacity 434.4 MiB
2022-04-29 19:58:16,200 INFO spark.SparkEnv: Registering OutputCommitCoordinator
2022-04-29 19:58:16,297 INFO util.log: Logging initialized @4185ms to org.sparkproject.jetty.util.log.Slf4jLog
2022-04-29 19:58:16,362 INFO server.Server: jetty-9.4.z-SNAPSHOT; built: 2019-04-29T20:42:08.989Z; git: e1bc35120a6617ee3df052294e433f
ntu0.20.04.1
2022-04-29 19:58:16,383 INFO server.Server: Started @4271ms
```

Fig 1,3: Execution process



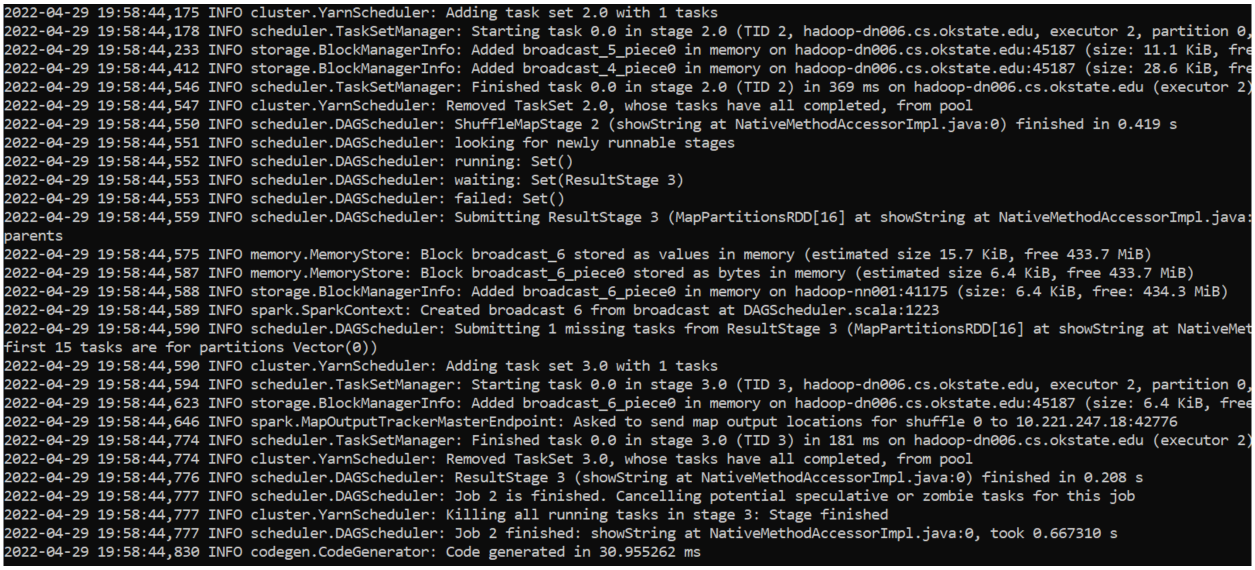
```
2022-04-29 19:58:16,451 INFO handler.ContextHandler: Started o.s.j.s.ServletContextHandler@19a02696{/stages,null,AVAILABLE,@Spark}
2022-04-29 19:58:16,452 INFO handler.ContextHandler: Started o.s.j.s.ServletContextHandler@3fae0d77{/stages/json,null,AVAILABLE,@Spark}
2022-04-29 19:58:16,453 INFO handler.ContextHandler: Started o.s.j.s.ServletContextHandler@200b4154{/stages/stage,null,AVAILABLE,@Spark}
2022-04-29 19:58:16,455 INFO handler.ContextHandler: Started o.s.j.s.ServletContextHandler@18cdc2fa{/stages/stage/json,null,AVAILABLE,@Spark}
2022-04-29 19:58:16,456 INFO handler.ContextHandler: Started o.s.j.s.ServletContextHandler@3d688f50{/stages/pool,null,AVAILABLE,@Spark}
2022-04-29 19:58:16,457 INFO handler.ContextHandler: Started o.s.j.s.ServletContextHandler@138c4816{/stages/pool/json,null,AVAILABLE,@Spark}
2022-04-29 19:58:16,458 INFO handler.ContextHandler: Started o.s.j.s.ServletContextHandler@5b065a50{/storage,null,AVAILABLE,@Spark}
2022-04-29 19:58:16,459 INFO handler.ContextHandler: Started o.s.j.s.ServletContextHandler@4b987c9c{/storage/json,null,AVAILABLE,@Spark}
2022-04-29 19:58:16,460 INFO handler.ContextHandler: Started o.s.j.s.ServletContextHandler@6a1a5f00{/storage/rdd,null,AVAILABLE,@Spark}
2022-04-29 19:58:16,461 INFO handler.ContextHandler: Started o.s.j.s.ServletContextHandler@72a21ec0{/storage/rdd/json,null,AVAILABLE,@Spark}
2022-04-29 19:58:16,462 INFO handler.ContextHandler: Started o.s.j.s.ServletContextHandler@73e92de2{/environment,null,AVAILABLE,@Spark}
2022-04-29 19:58:16,464 INFO handler.ContextHandler: Started o.s.j.s.ServletContextHandler@5c4805b5{/environment/json,null,AVAILABLE,@Spark}
2022-04-29 19:58:16,465 INFO handler.ContextHandler: Started o.s.j.s.ServletContextHandler@315e9885{/executors,null,AVAILABLE,@Spark}
2022-04-29 19:58:16,466 INFO handler.ContextHandler: Started o.s.j.s.ServletContextHandler@61509a5f{/executors/json,null,AVAILABLE,@Spark}
2022-04-29 19:58:16,467 INFO handler.ContextHandler: Started o.s.j.s.ServletContextHandler@3ee65203{/executors/threadDump,null,AVAILABLE,@Spark}
2022-04-29 19:58:16,469 INFO handler.ContextHandler: Started o.s.j.s.ServletContextHandler@6c703f6f{/executors/threadDump/json,null,AVAILABLE,@Spar
2022-04-29 19:58:16,478 INFO handler.ContextHandler: Started o.s.j.s.ServletContextHandler@5039690e{/static,null,AVAILABLE,@Spark}
2022-04-29 19:58:16,479 INFO handler.ContextHandler: Started o.s.j.s.ServletContextHandler@2b18c20c{/,null,AVAILABLE,@Spark}
2022-04-29 19:58:16,480 INFO handler.ContextHandler: Started o.s.j.s.ServletContextHandler@5a746549{/api,null,AVAILABLE,@Spark}
2022-04-29 19:58:16,481 INFO handler.ContextHandler: Started o.s.j.s.ServletContextHandler@3196b87e{/jobs/job/kill,null,AVAILABLE,@Spark}
2022-04-29 19:58:16,482 INFO handler.ContextHandler: Started o.s.j.s.ServletContextHandler@5f51ec41{/stages/stage/kill,null,AVAILABLE,@Spark}
2022-04-29 19:58:16,485 INFO ui.SparkUI: Bound SparkUI to 0.0.0.0, and started at http://hadoop-nn001:4045
2022-04-29 19:58:16,881 INFO client.RMProxy: Connecting to ResourceManager at hadoop-nn001.cs.okstate.edu/192.168.122.2:8032
2022-04-29 19:58:17,170 INFO yarn.Client: Requesting a new application from cluster with 12 NodeManagers
2022-04-29 19:58:17,597 INFO conf.Configuration: resource-types.xml not found
2022-04-29 19:58:17,598 INFO resource.ResourceUtils: Unable to find 'resource-types.xml'.
2022-04-29 19:58:17,629 INFO yarn.Client: Verifying our application has not requested more than the maximum memory capability of the cluster (8192
2022-04-29 19:58:17,630 INFO yarn.Client: Will allocate AM container, with 896 MB memory including 384 MB overhead
2022-04-29 19:58:17,630 INFO yarn.Client: Setting up container launch context for our AM
2022-04-29 19:58:17,634 INFO yarn.Client: Setting up the launch environment for our AM container
2022-04-29 19:58:17,645 INFO yarn.Client: Preparing resources for our AM container
2022-04-29 19:58:17,728 WARN yarn.Client: Neither spark.yarn.jars nor spark.yarn.archive is set, falling back to uploading libraries under SPARK_HO
```

Fig 1,4: Execution Process



```
        user: sdarapu
2022-04-29 19:58:25,684 INFO yarn.Client: Application report for application_1647031195237_1464 (state: ACCEPTED)
2022-04-29 19:58:26,686 INFO yarn.Client: Application report for application_1647031195237_1464 (state: ACCEPTED)
2022-04-29 19:58:27,689 INFO yarn.Client: Application report for application_1647031195237_1464 (state: ACCEPTED)
2022-04-29 19:58:28,360 INFO cluster.YarnClientSchedulerBackend: Add WebUI Filter. org.apache.hadoop.yarn.server.webproxy.amfilter.AmIpFilter, Map(PRO
01, PROXY_URI_BASES -> http://hadoop-nn001:8088/proxy/application_1647031195237_1464), /proxy/application_1647031195237_1464
2022-04-29 19:58:28,691 INFO yarn.Client: Application report for application_1647031195237_1464 (state: RUNNING)
2022-04-29 19:58:28,691 INFO yarn.Client:
        client token: N/A
        diagnostics: N/A
        ApplicationMaster host: 10.221.247.19
        ApplicationMaster RPC port: -1
        queue: default
        start time: 1651280303642
        final status: UNDEFINED
        tracking URL: http://hadoop-nn001.cs.okstate.edu:8088/proxy/application_1647031195237_1464/
        user: sdarapu
2022-04-29 19:58:28,693 INFO cluster.YarnClientSchedulerBackend: Application application_1647031195237_1464 has started running.
2022-04-29 19:58:28,709 INFO util.Utils: Successfully started service 'org.apache.spark.network.netty.NettyBlockTransferService' on port 41175.
2022-04-29 19:58:28,710 INFO netty.NettyBlockTransferService: Server created on hadoop-nn001:41175
2022-04-29 19:58:28,712 INFO storage.BlockManager: Using org.apache.spark.storage.RandomBlockReplicationPolicy for block replication policy
2022-04-29 19:58:28,725 INFO storage.BlockManagerMaster: Registering BlockManager BlockManagerId(driver, hadoop-nn001, 41175, None)
2022-04-29 19:58:28,731 INFO storage.BlockManagerMasterEndpoint: Registering block manager hadoop-nn001:41175 with 434.4 MiB RAM, BlockManagerId(drive
, None)
2022-04-29 19:58:28,736 INFO storage.BlockManagerMaster: Registered BlockManager BlockManagerId(driver, hadoop-nn001, 41175, None)
2022-04-29 19:58:28,738 INFO storage.BlockManager: Initialized BlockManager: BlockManagerId(driver, hadoop-nn001, 41175, None)
2022-04-29 19:58:28,912 INFO ui.ServerInfo: Adding filter to /metrics/json: org.apache.hadoop.yarn.server.webproxy.amfilter.AmIpFilter
2022-04-29 19:58:28,915 INFO handler.ContextHandler: Started o.s.j.s.ServletContextHandler@5a2c107{/metrics/json,null,AVAILABLE,@Spark}
2022-04-29 19:58:29,069 INFO cluster.YarnSchedulerBackend$YarnSchedulerEndpoint: ApplicationMaster registered as NettyRpcEndpointRef(spark-client://Ya
2022-04-29 19:58:33,481 INFO resource.ResourceProfile: Default ResourceProfile created, executor resources: Map(cores -> name: cores, amount: 1, scrip
-> name: memory, amount: 1024, script: , vendor: ), task resources: Map(cpus -> name: cpus, amount: 1.0)
```

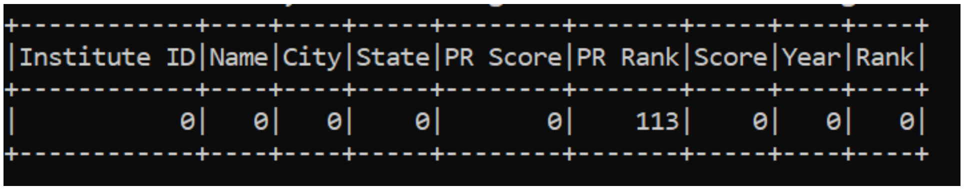
Fig 1,5: Execution Process



```
2022-04-29 19:58:44,175 INFO cluster.YarnScheduler: Adding task set 2.0 with 1 tasks
2022-04-29 19:58:44,178 INFO scheduler.TaskSetManager: Starting task 0.0 in stage 2.0 (TID 2, hadoop-dn006.cs.okstate.edu, executor 2, partition 0,
2022-04-29 19:58:44,233 INFO storage.BlockManagerInfo: Added broadcast_5_piece0 in memory on hadoop-dn006.cs.okstate.edu:45187 (size: 11.1 KiB, fre
2022-04-29 19:58:44,412 INFO storage.BlockManagerInfo: Added broadcast_4_piece0 in memory on hadoop-dn006.cs.okstate.edu:45187 (size: 28.6 KiB, fre
2022-04-29 19:58:44,546 INFO scheduler.TaskSetManager: Finished task 0.0 in stage 2.0 (TID 2) in 369 ms on hadoop-dn006.cs.okstate.edu (executor 2)
2022-04-29 19:58:44,547 INFO cluster.YarnScheduler: Removed TaskSet 2.0, whose tasks have all completed, from pool
2022-04-29 19:58:44,550 INFO scheduler.DAGScheduler: ShuffleMapStage 2 (showString at NativeMethodAccessorImpl.java:0) finished in 0.419 s
2022-04-29 19:58:44,551 INFO scheduler.DAGScheduler: looking for newly runnable stages
2022-04-29 19:58:44,552 INFO scheduler.DAGScheduler: running: Set()
2022-04-29 19:58:44,553 INFO scheduler.DAGScheduler: waiting: Set(ResultStage 3)
2022-04-29 19:58:44,553 INFO scheduler.DAGScheduler: failed: Set()
2022-04-29 19:58:44,559 INFO scheduler.DAGScheduler: Submitting ResultStage 3 (MapPartitionsRDD[16] at showString at NativeMethodAccessorImpl.java:
parents
2022-04-29 19:58:44,575 INFO memory.MemoryStore: Block broadcast_6 stored as values in memory (estimated size 15.7 KiB, free 433.7 MiB)
2022-04-29 19:58:44,587 INFO memory.MemoryStore: Block broadcast_6_piece0 stored as bytes in memory (estimated size 6.4 KiB, free 433.7 MiB)
2022-04-29 19:58:44,588 INFO storage.BlockManagerInfo: Added broadcast_6_piece0 in memory on hadoop-nn001:41175 (size: 6.4 KiB, free: 434.3 MiB)
2022-04-29 19:58:44,589 INFO spark.SparkContext: Created broadcast 6 from broadcast at DAGScheduler.scala:1223
2022-04-29 19:58:44,590 INFO scheduler.DAGScheduler: Submitting 1 missing tasks from ResultStage 3 (MapPartitionsRDD[16] at showString at NativeMet
first 15 tasks are for partitions Vector(0))
2022-04-29 19:58:44,590 INFO cluster.YarnScheduler: Adding task set 3.0 with 1 tasks
2022-04-29 19:58:44,594 INFO scheduler.TaskSetManager: Starting task 0.0 in stage 3.0 (TID 3, hadoop-dn006.cs.okstate.edu, executor 2, partition 0,
2022-04-29 19:58:44,623 INFO storage.BlockManagerInfo: Added broadcast_6_piece0 in memory on hadoop-dn006.cs.okstate.edu:45187 (size: 6.4 KiB, free
2022-04-29 19:58:44,646 INFO spark.MapOutputTrackerMasterEndpoint: Asked to send map output locations for shuffle 0 to 10.221.247.18:42776
2022-04-29 19:58:44,774 INFO scheduler.TaskSetManager: Finished task 0.0 in stage 3.0 (TID 3) in 181 ms on hadoop-dn006.cs.okstate.edu (executor 2)
2022-04-29 19:58:44,774 INFO cluster.YarnScheduler: Removed TaskSet 3.0, whose tasks have all completed, from pool
2022-04-29 19:58:44,776 INFO scheduler.DAGScheduler: ResultStage 3 (showString at NativeMethodAccessorImpl.java:0) finished in 0.208 s
2022-04-29 19:58:44,777 INFO scheduler.DAGScheduler: Job 2 is finished. Cancelling potential speculative or zombie tasks for this job
2022-04-29 19:58:44,777 INFO cluster.YarnScheduler: Killing all running tasks in stage 3: Stage finished
2022-04-29 19:58:44,777 INFO scheduler.DAGScheduler: Job 2 finished: showString at NativeMethodAccessorImpl.java:0, took 0.667310 s
2022-04-29 19:58:44,830 INFO codegen.CodeGenerator: Code generated in 30.955262 ms
```

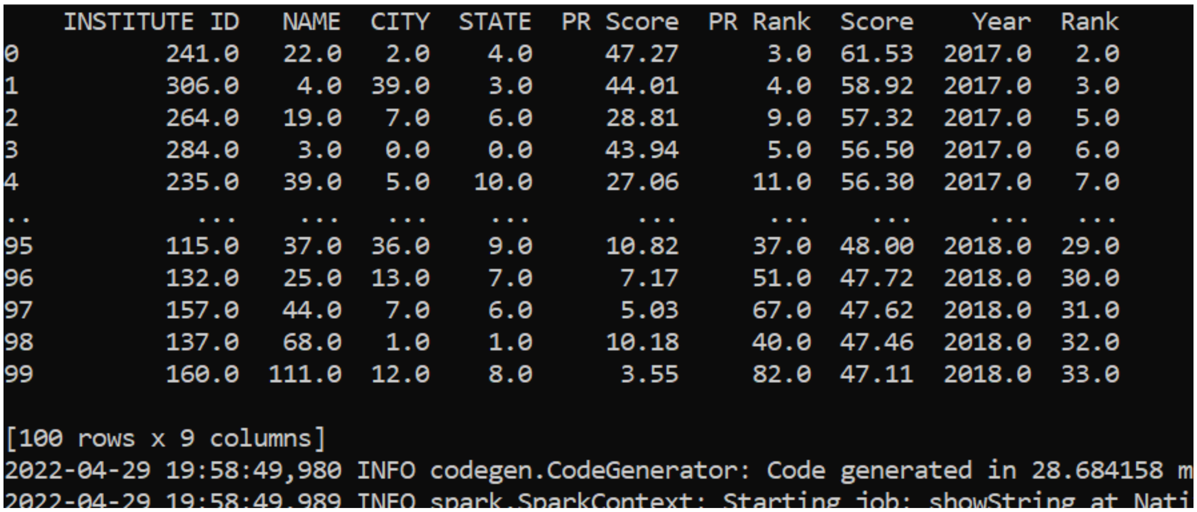
Fig 1,6: Execution Process

<u>Output displayed on console:</u>



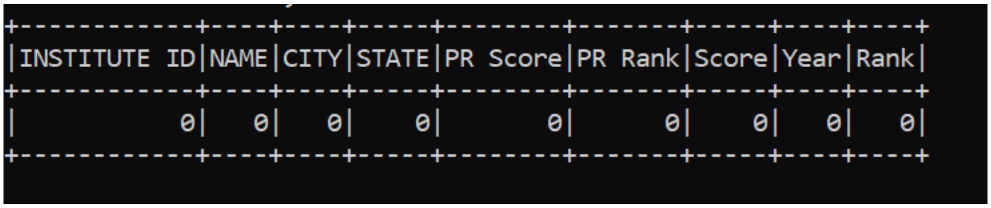
```
+------------+----+----+-----+--------+-------+-----+----+----+
|Institute ID|Name|City|State|PR Score|PR Rank|Score|Year|Rank|
+------------+----+----+-----+--------+-------+-----+----+----+
|           0|   0|   0|    0|       0|    113|    0|   0|   0|
+------------+----+----+-----+--------+-------+-----+----+----+
```

Fig: 1,7: PR Rank contains null values in 113 rows



```
    INSTITUTE ID   NAME  CITY  STATE  PR Score  PR Rank  Score    Year  Rank
0          241.0   22.0   2.0    4.0     47.27      3.0  61.53  2017.0   2.0
1          306.0    4.0  39.0    3.0     44.01      4.0  58.92  2017.0   3.0
2          264.0   19.0   7.0    6.0     28.81      9.0  57.32  2017.0   5.0
3          284.0    3.0   0.0    0.0     43.94      5.0  56.50  2017.0   6.0
4          235.0   39.0   5.0   10.0     27.06     11.0  56.30  2017.0   7.0
..           ...    ...   ...    ...       ...      ...    ...     ...   ...
95         115.0   37.0  36.0    9.0     10.82     37.0  48.00  2018.0  29.0
96         132.0   25.0  13.0    7.0      7.17     51.0  47.72  2018.0  30.0
97         157.0   44.0   7.0    6.0      5.03     67.0  47.62  2018.0  31.0
98         137.0   68.0   1.0    1.0     10.18     40.0  47.46  2018.0  32.0
99         160.0  111.0  12.0    8.0      3.55     82.0  47.11  2018.0  33.0

[100 rows x 9 columns]
2022-04-29 19:58:49,980 INFO codegen.CodeGenerator: Code generated in 28.684158 m
2022-04-29 19:58:49,989 INFO spark.SparkContext: Starting job: showString at Nati
```

Fig 1,8: Displays dataframe without any null values and out of range values of first 100 rows

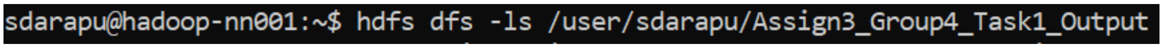


```
+------------+----+----+-----+--------+-------+-----+----+----+
|INSTITUTE ID|NAME|CITY|STATE|PR Score|PR Rank|Score|Year|Rank|
+------------+----+----+-----+--------+-------+-----+----+----+
|           0|   0|   0|    0|       0|      0|    0|   0|   0|
+------------+----+----+-----+--------+-------+-----+----+----+
```

Fig 1,9: Checks and confirms that all rows in PR Rank and Score doesn't contain any null values

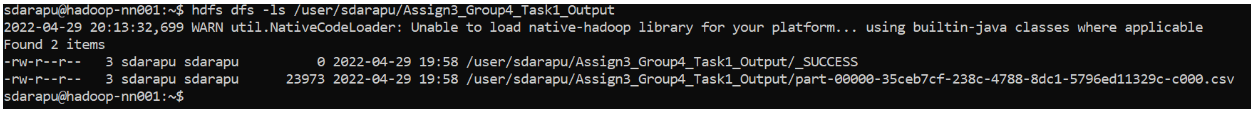
<u>Output stored in the specified folder under HDFS:</u>

i. To see the result stored in the specified folder ("Assign3_Group4_Task1_Output"), execute the below command as shown after that we can see "csv" file in which the data is saved.



```
sdarapu@hadoop-nn001:~$ hdfs dfs -ls /user/sdarapu/Assign3_Group4_Task1_Output
```

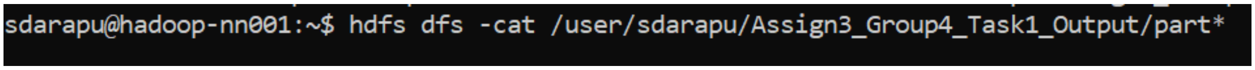
Fig 1,10: View list of contents



```
sdarapu@hadoop-nn001:~$ hdfs dfs -ls /user/sdarapu/Assign3_Group4_Task1_Output
2022-04-29 20:13:32,699 WARN util.NativeCodeLoader: Unable to load native-hadoop library for your platform... using builtin-java classes where applicable
Found 2 items
-rw-r--r--   3 sdarapu sdarapu          0 2022-04-29 19:58 /user/sdarapu/Assign3_Group4_Task1_Output/_SUCCESS
-rw-r--r--   3 sdarapu sdarapu      23973 2022-04-29 19:58 /user/sdarapu/Assign3_Group4_Task1_Output/part-00000-35ceb7cf-238c-4788-8dc1-5796ed11329c-c000.csv
sdarapu@hadoop-nn001:~$
```

Fig 1,11: List of files in the respective folder

ii. To view the data in the folder "Assign3_Group4_Task1_Output" at once, execute the below command.

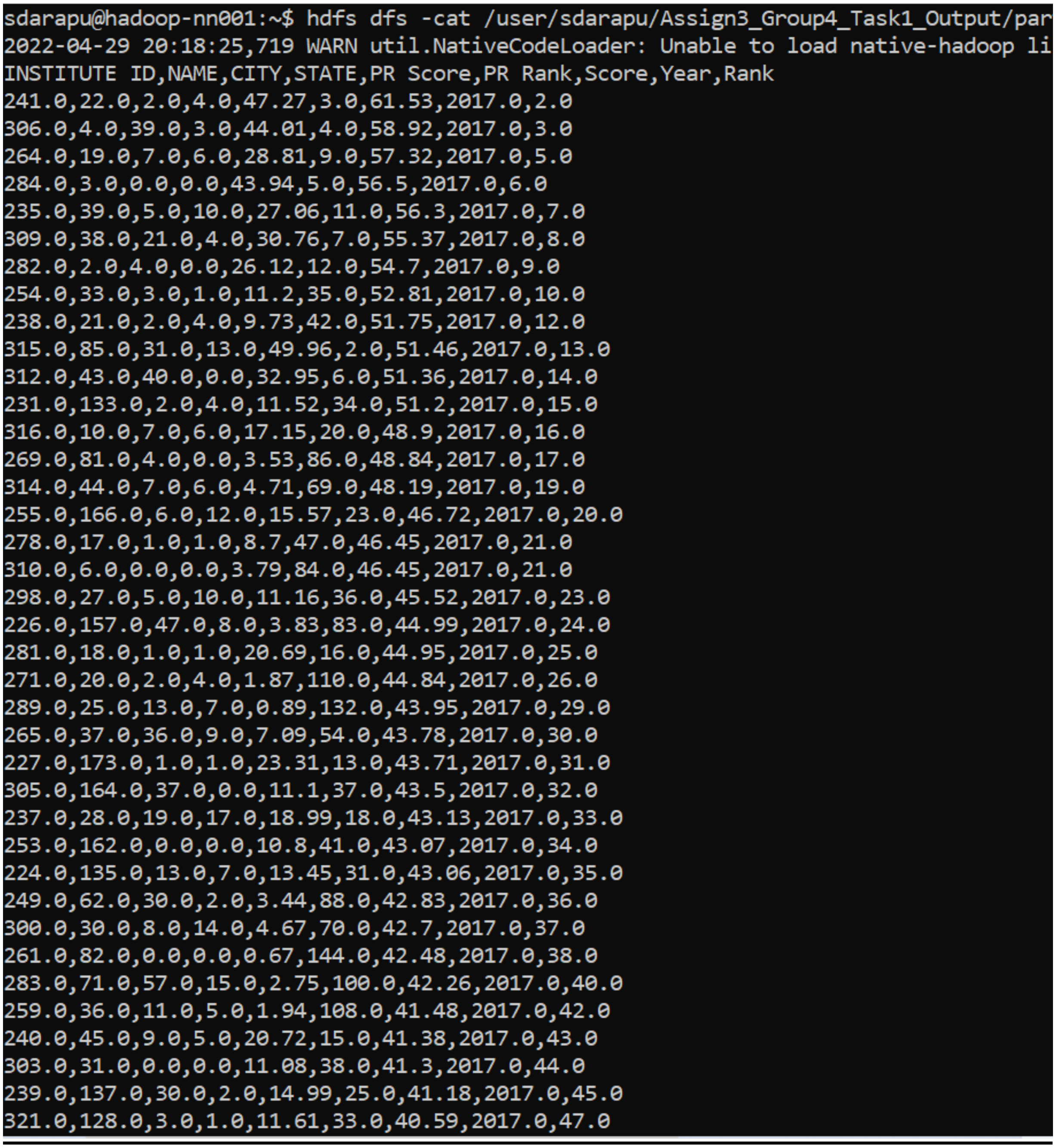


```
sdarapu@hadoop-nn001:~$ hdfs dfs -cat /user/sdarapu/Assign3_Group4_Task1_Output/part*
```

Fig 1,12: To view the data use the above command

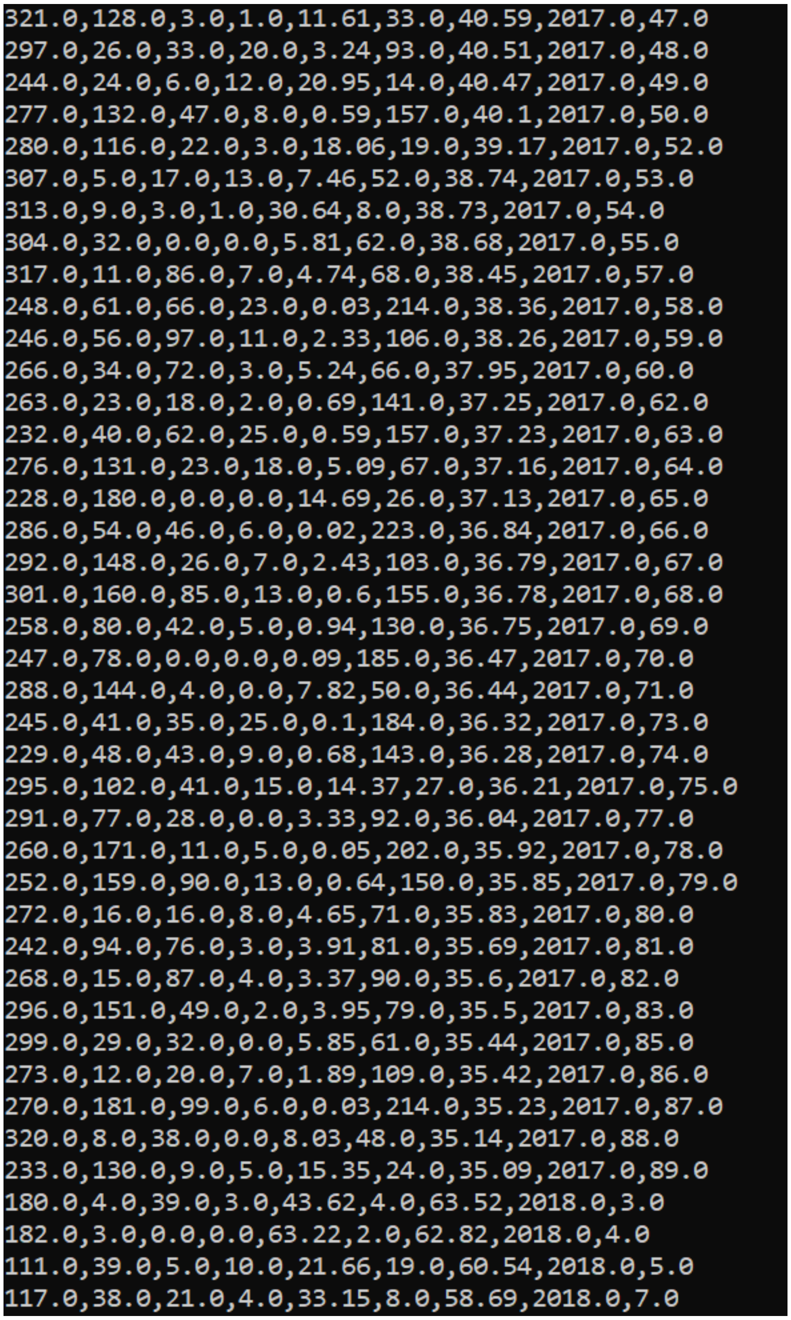
Note: Above commands can be used to view the data in the folder "Assign3_Group4_Task1_Output_inpfor_Task2-4"

Output display:



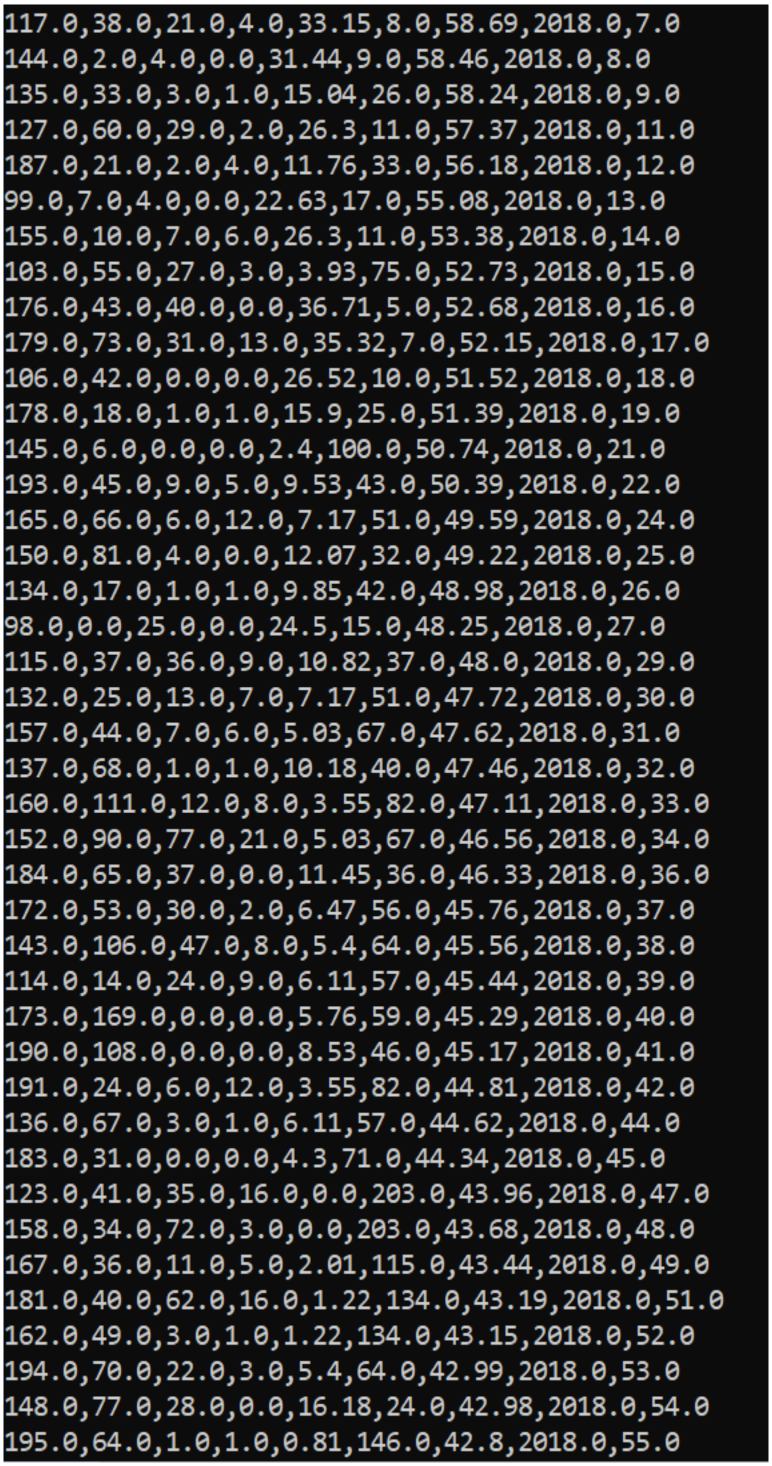
```
sdarapu@hadoop-nn001:~$ hdfs dfs -cat /user/sdarapu/Assign3_Group4_Task1_Output/par
2022-04-29 20:18:25,719 WARN util.NativeCodeLoader: Unable to load native-hadoop li
INSTITUTE ID,NAME,CITY,STATE,PR Score,PR Rank,Score,Year,Rank
241.0,22.0,2.0,4.0,47.27,3.0,61.53,2017.0,2.0
306.0,4.0,39.0,3.0,44.01,4.0,58.92,2017.0,3.0
264.0,19.0,7.0,6.0,28.81,9.0,57.32,2017.0,5.0
284.0,3.0,0.0,0.0,43.94,5.0,56.5,2017.0,6.0
235.0,39.0,5.0,10.0,27.06,11.0,56.3,2017.0,7.0
309.0,38.0,21.0,4.0,30.76,7.0,55.37,2017.0,8.0
282.0,2.0,4.0,0.0,26.12,12.0,54.7,2017.0,9.0
254.0,33.0,3.0,1.0,11.2,35.0,52.81,2017.0,10.0
238.0,21.0,2.0,4.0,9.73,42.0,51.75,2017.0,12.0
315.0,85.0,31.0,13.0,49.96,2.0,51.46,2017.0,13.0
312.0,43.0,40.0,0.0,32.95,6.0,51.36,2017.0,14.0
231.0,133.0,2.0,4.0,11.52,34.0,51.2,2017.0,15.0
316.0,10.0,7.0,6.0,17.15,20.0,48.9,2017.0,16.0
269.0,81.0,4.0,0.0,3.53,86.0,48.84,2017.0,17.0
314.0,44.0,7.0,6.0,4.71,69.0,48.19,2017.0,19.0
255.0,166.0,6.0,12.0,15.57,23.0,46.72,2017.0,20.0
278.0,17.0,1.0,1.0,8.7,47.0,46.45,2017.0,21.0
310.0,6.0,0.0,0.0,3.79,84.0,46.45,2017.0,21.0
298.0,27.0,5.0,10.0,11.16,36.0,45.52,2017.0,23.0
226.0,157.0,47.0,8.0,3.83,83.0,44.99,2017.0,24.0
281.0,18.0,1.0,1.0,20.69,16.0,44.95,2017.0,25.0
271.0,20.0,2.0,4.0,1.87,110.0,44.84,2017.0,26.0
289.0,25.0,13.0,7.0,0.89,132.0,43.95,2017.0,29.0
265.0,37.0,36.0,9.0,7.09,54.0,43.78,2017.0,30.0
227.0,173.0,1.0,1.0,23.31,13.0,43.71,2017.0,31.0
305.0,164.0,37.0,0.0,11.1,37.0,43.5,2017.0,32.0
237.0,28.0,19.0,17.0,18.99,18.0,43.13,2017.0,33.0
253.0,162.0,0.0,0.0,10.8,41.0,43.07,2017.0,34.0
224.0,135.0,13.0,7.0,13.45,31.0,43.06,2017.0,35.0
249.0,62.0,30.0,2.0,3.44,88.0,42.83,2017.0,36.0
300.0,30.0,8.0,14.0,4.67,70.0,42.7,2017.0,37.0
261.0,82.0,0.0,0.0,0.67,144.0,42.48,2017.0,38.0
283.0,71.0,57.0,15.0,2.75,100.0,42.26,2017.0,40.0
259.0,36.0,11.0,5.0,1.94,108.0,41.48,2017.0,42.0
240.0,45.0,9.0,5.0,20.72,15.0,41.38,2017.0,43.0
303.0,31.0,0.0,0.0,11.08,38.0,41.3,2017.0,44.0
239.0,137.0,30.0,2.0,14.99,25.0,41.18,2017.0,45.0
321.0,128.0,3.0,1.0,11.61,33.0,40.59,2017.0,47.0
```

Fig 1,13: Output



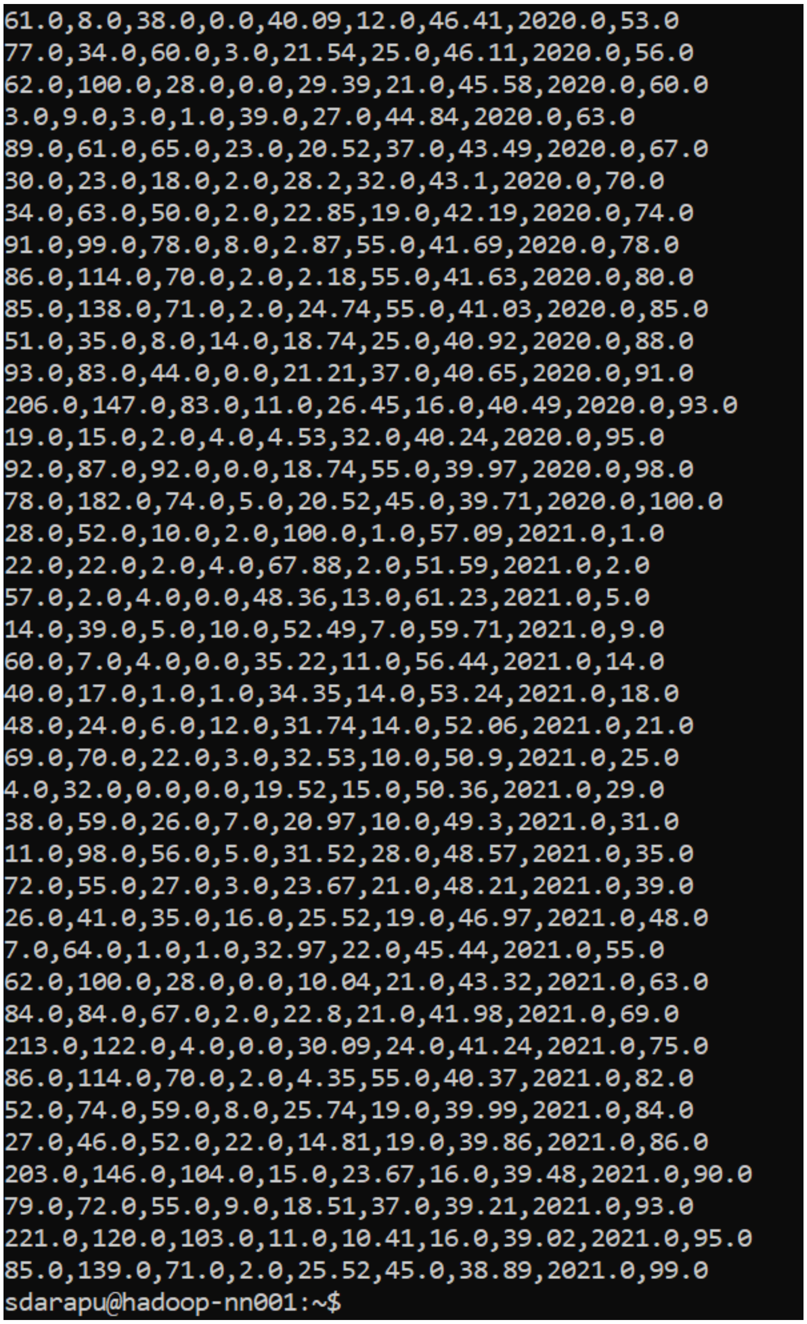
```
321.0,128.0,3.0,1.0,11.61,33.0,40.59,2017.0,47.0
297.0,26.0,33.0,20.0,3.24,93.0,40.51,2017.0,48.0
244.0,24.0,6.0,12.0,20.95,14.0,40.47,2017.0,49.0
277.0,132.0,47.0,8.0,0.59,157.0,40.1,2017.0,50.0
280.0,116.0,22.0,3.0,18.06,19.0,39.17,2017.0,52.0
307.0,5.0,17.0,13.0,7.46,52.0,38.74,2017.0,53.0
313.0,9.0,3.0,1.0,30.64,8.0,38.73,2017.0,54.0
304.0,32.0,0.0,0.0,5.81,62.0,38.68,2017.0,55.0
317.0,11.0,86.0,7.0,4.74,68.0,38.45,2017.0,57.0
248.0,61.0,66.0,23.0,0.03,214.0,38.36,2017.0,58.0
246.0,56.0,97.0,11.0,2.33,106.0,38.26,2017.0,59.0
266.0,34.0,72.0,3.0,5.24,66.0,37.95,2017.0,60.0
263.0,23.0,18.0,2.0,0.69,141.0,37.25,2017.0,62.0
232.0,40.0,62.0,25.0,0.59,157.0,37.23,2017.0,63.0
276.0,131.0,23.0,18.0,5.09,67.0,37.16,2017.0,64.0
228.0,180.0,0.0,0.0,14.69,26.0,37.13,2017.0,65.0
286.0,54.0,46.0,6.0,0.02,223.0,36.84,2017.0,66.0
292.0,148.0,26.0,7.0,2.43,103.0,36.79,2017.0,67.0
301.0,160.0,85.0,13.0,0.6,155.0,36.78,2017.0,68.0
258.0,80.0,42.0,5.0,0.94,130.0,36.75,2017.0,69.0
247.0,78.0,0.0,0.0,0.09,185.0,36.47,2017.0,70.0
288.0,144.0,4.0,0.0,7.82,50.0,36.44,2017.0,71.0
245.0,41.0,35.0,25.0,0.1,184.0,36.32,2017.0,73.0
229.0,48.0,43.0,9.0,0.68,143.0,36.28,2017.0,74.0
295.0,102.0,41.0,15.0,14.37,27.0,36.21,2017.0,75.0
291.0,77.0,28.0,0.0,3.33,92.0,36.04,2017.0,77.0
260.0,171.0,11.0,5.0,0.05,202.0,35.92,2017.0,78.0
252.0,159.0,90.0,13.0,0.64,150.0,35.85,2017.0,79.0
272.0,16.0,16.0,8.0,4.65,71.0,35.83,2017.0,80.0
242.0,94.0,76.0,3.0,3.91,81.0,35.69,2017.0,81.0
268.0,15.0,87.0,4.0,3.37,90.0,35.6,2017.0,82.0
296.0,151.0,49.0,2.0,3.95,79.0,35.5,2017.0,83.0
299.0,29.0,32.0,0.0,5.85,61.0,35.44,2017.0,85.0
273.0,12.0,20.0,7.0,1.89,109.0,35.42,2017.0,86.0
270.0,181.0,99.0,6.0,0.03,214.0,35.23,2017.0,87.0
320.0,8.0,38.0,0.0,8.03,48.0,35.14,2017.0,88.0
233.0,130.0,9.0,5.0,15.35,24.0,35.09,2017.0,89.0
180.0,4.0,39.0,3.0,43.62,4.0,63.52,2018.0,3.0
182.0,3.0,0.0,0.0,63.22,2.0,62.82,2018.0,4.0
111.0,39.0,5.0,10.0,21.66,19.0,60.54,2018.0,5.0
117.0,38.0,21.0,4.0,33.15,8.0,58.69,2018.0,7.0
```

Fig 1,14:0utput



```
117.0,38.0,21.0,4.0,33.15,8.0,58.69,2018.0,7.0
144.0,2.0,4.0,0.0,31.44,9.0,58.46,2018.0,8.0
135.0,33.0,3.0,1.0,15.04,26.0,58.24,2018.0,9.0
127.0,60.0,29.0,2.0,26.3,11.0,57.37,2018.0,11.0
187.0,21.0,2.0,4.0,11.76,33.0,56.18,2018.0,12.0
99.0,7.0,4.0,0.0,22.63,17.0,55.08,2018.0,13.0
155.0,10.0,7.0,6.0,26.3,11.0,53.38,2018.0,14.0
103.0,55.0,27.0,3.0,3.93,75.0,52.73,2018.0,15.0
176.0,43.0,40.0,0.0,36.71,5.0,52.68,2018.0,16.0
179.0,73.0,31.0,13.0,35.32,7.0,52.15,2018.0,17.0
106.0,42.0,0.0,0.0,26.52,10.0,51.52,2018.0,18.0
178.0,18.0,1.0,1.0,15.9,25.0,51.39,2018.0,19.0
145.0,6.0,0.0,0.0,2.4,100.0,50.74,2018.0,21.0
193.0,45.0,9.0,5.0,9.53,43.0,50.39,2018.0,22.0
165.0,66.0,6.0,12.0,7.17,51.0,49.59,2018.0,24.0
150.0,81.0,4.0,0.0,12.07,32.0,49.22,2018.0,25.0
134.0,17.0,1.0,1.0,9.85,42.0,48.98,2018.0,26.0
98.0,0.0,25.0,0.0,24.5,15.0,48.25,2018.0,27.0
115.0,37.0,36.0,9.0,10.82,37.0,48.0,2018.0,29.0
132.0,25.0,13.0,7.0,7.17,51.0,47.72,2018.0,30.0
157.0,44.0,7.0,6.0,5.03,67.0,47.62,2018.0,31.0
137.0,68.0,1.0,1.0,10.18,40.0,47.46,2018.0,32.0
160.0,111.0,12.0,8.0,3.55,82.0,47.11,2018.0,33.0
152.0,90.0,77.0,21.0,5.03,67.0,46.56,2018.0,34.0
184.0,65.0,37.0,0.0,11.45,36.0,46.33,2018.0,36.0
172.0,53.0,30.0,2.0,6.47,56.0,45.76,2018.0,37.0
143.0,106.0,47.0,8.0,5.4,64.0,45.56,2018.0,38.0
114.0,14.0,24.0,9.0,6.11,57.0,45.44,2018.0,39.0
173.0,169.0,0.0,0.0,5.76,59.0,45.29,2018.0,40.0
190.0,108.0,0.0,0.0,8.53,46.0,45.17,2018.0,41.0
191.0,24.0,6.0,12.0,3.55,82.0,44.81,2018.0,42.0
136.0,67.0,3.0,1.0,6.11,57.0,44.62,2018.0,44.0
183.0,31.0,0.0,0.0,4.3,71.0,44.34,2018.0,45.0
123.0,41.0,35.0,16.0,0.0,203.0,43.96,2018.0,47.0
158.0,34.0,72.0,3.0,0.0,203.0,43.68,2018.0,48.0
167.0,36.0,11.0,5.0,2.01,115.0,43.44,2018.0,49.0
181.0,40.0,62.0,16.0,1.22,134.0,43.19,2018.0,51.0
162.0,49.0,3.0,1.0,1.22,134.0,43.15,2018.0,52.0
194.0,70.0,22.0,3.0,5.4,64.0,42.99,2018.0,53.0
148.0,77.0,28.0,0.0,16.18,24.0,42.98,2018.0,54.0
195.0,64.0,1.0,1.0,0.81,146.0,42.8,2018.0,55.0
```

Fig 1,15:Output



```
61.0,8.0,38.0,0.0,40.09,12.0,46.41,2020.0,53.0
77.0,34.0,60.0,3.0,21.54,25.0,46.11,2020.0,56.0
62.0,100.0,28.0,0.0,29.39,21.0,45.58,2020.0,60.0
3.0,9.0,3.0,1.0,39.0,27.0,44.84,2020.0,63.0
89.0,61.0,65.0,23.0,20.52,37.0,43.49,2020.0,67.0
30.0,23.0,18.0,2.0,28.2,32.0,43.1,2020.0,70.0
34.0,63.0,50.0,2.0,22.85,19.0,42.19,2020.0,74.0
91.0,99.0,78.0,8.0,2.87,55.0,41.69,2020.0,78.0
86.0,114.0,70.0,2.0,2.18,55.0,41.63,2020.0,80.0
85.0,138.0,71.0,2.0,24.74,55.0,41.03,2020.0,85.0
51.0,35.0,8.0,14.0,18.74,25.0,40.92,2020.0,88.0
93.0,83.0,44.0,0.0,21.21,37.0,40.65,2020.0,91.0
206.0,147.0,83.0,11.0,26.45,16.0,40.49,2020.0,93.0
19.0,15.0,2.0,4.0,4.53,32.0,40.24,2020.0,95.0
92.0,87.0,92.0,0.0,18.74,55.0,39.97,2020.0,98.0
78.0,182.0,74.0,5.0,20.52,45.0,39.71,2020.0,100.0
28.0,52.0,10.0,2.0,100.0,1.0,57.09,2021.0,1.0
22.0,22.0,2.0,4.0,67.88,2.0,51.59,2021.0,2.0
57.0,2.0,4.0,0.0,48.36,13.0,61.23,2021.0,5.0
14.0,39.0,5.0,10.0,52.49,7.0,59.71,2021.0,9.0
60.0,7.0,4.0,0.0,35.22,11.0,56.44,2021.0,14.0
40.0,17.0,1.0,1.0,34.35,14.0,53.24,2021.0,18.0
48.0,24.0,6.0,12.0,31.74,14.0,52.06,2021.0,21.0
69.0,70.0,22.0,3.0,32.53,10.0,50.9,2021.0,25.0
4.0,32.0,0.0,0.0,19.52,15.0,50.36,2021.0,29.0
38.0,59.0,26.0,7.0,20.97,10.0,49.3,2021.0,31.0
11.0,98.0,56.0,5.0,31.52,28.0,48.57,2021.0,35.0
72.0,55.0,27.0,3.0,23.67,21.0,48.21,2021.0,39.0
26.0,41.0,35.0,16.0,25.52,19.0,46.97,2021.0,48.0
7.0,64.0,1.0,1.0,32.97,22.0,45.44,2021.0,55.0
62.0,100.0,28.0,0.0,10.04,21.0,43.32,2021.0,63.0
84.0,84.0,67.0,2.0,22.8,21.0,41.98,2021.0,69.0
213.0,122.0,4.0,0.0,30.09,24.0,41.24,2021.0,75.0
86.0,114.0,70.0,2.0,4.35,55.0,40.37,2021.0,82.0
52.0,74.0,59.0,8.0,25.74,19.0,39.99,2021.0,84.0
27.0,46.0,52.0,22.0,14.81,19.0,39.86,2021.0,86.0
203.0,146.0,104.0,15.0,23.67,16.0,39.48,2021.0,90.0
79.0,72.0,55.0,9.0,18.51,37.0,39.21,2021.0,93.0
221.0,120.0,103.0,11.0,10.41,16.0,39.02,2021.0,95.0
85.0,139.0,71.0,2.0,25.52,45.0,38.89,2021.0,99.0
sdarapu@hadoop-nn001:~$
```

Fig 1,16: Output

<u>Discussion of Result:</u>

In this Data Correction Task, At First, we have manually created 113 Null values in PR Rank column and replaced with -1 for out-of-range values in Score column as well as for null values in PR Rank column. Later we found cosine similarity between the clean dataframe and dataframe contains -1 in the rows. By using this cosine similarity, we filled all those Null and out of range values with similar values which is obtained from the similarity method. All the output rows without containing any null values and out of range values are shown in above screenshots.