B.M.S College of Engineering

P.O. Box No.: 1908 Bull Temple Road,

Bangalore-560 019

DEPARTMENT OF INFORMATION SCIENCE & ENGINEERING

Course – Big Data Analytics

Course Code – 20IS8OEBDA

Final report on Project work

World Happiness Report analysis using apache spark

Submitted to – Shubha Rao V

Submitted by -

Rajesh BT - 1BM18IS079

Prahlad Nayak - 1BM18IS066

B.M.S College of Engineering

P.O. Box No.: 1908 Bull Temple Road,

Bangalore-560 019

DEPARTMENT OF INFORMATION SCIENCE & ENGINEERING

CERTIFICATE

Certified that the Project has been successfully presented at **B.M.S College Of Engineering** by **Rajesh BT**, **Prahlad Nayak** bearing USN: **1BM18IS079, and 1BM18IS066** in partial fulfillment of the requirements for the IV Semester degree in **Bachelor of Engineering in Information Science & Engineering** of **Visvesvaraya Technological University, Belgaum** as a part of the course **Big Data Analytics *(course code)*** during academic year 2020-2021.

Faculty Name – Shubha Rao V

Designation – Associate Professor

Department of ISE, BMSCE

TABLE OF CONTENTS

Abstract:

The project deals with analysis of the World Happiness Report using Spark. PySpark is an interface for Apache Spark in Python. It not only allows you to write Spark applications using Python APIs, but also provides the PySpark shell for interactively analyzing your data in a distributed environment. PySpark supports most of Spark's features such as Spark SQL, DataFrame, Streaming, MLlib (Machine Learning) and Spark Core. We used functions of PySpark like SparkContext, CreateDataFrame, Aggregate, Sum, Collect, Count, CountDistinct, etc.

Problem Statement:

Given the World Happiness Report of all the countries of the last 15 years, we try to give the countries a ranking based on their performance in the World Happiness Report over the past 15 years, so that we can predict the best country to live based on all the parameters. We also try to rank the countries based on many other criterias and try to infer some key points from the dataset and conclude a few.

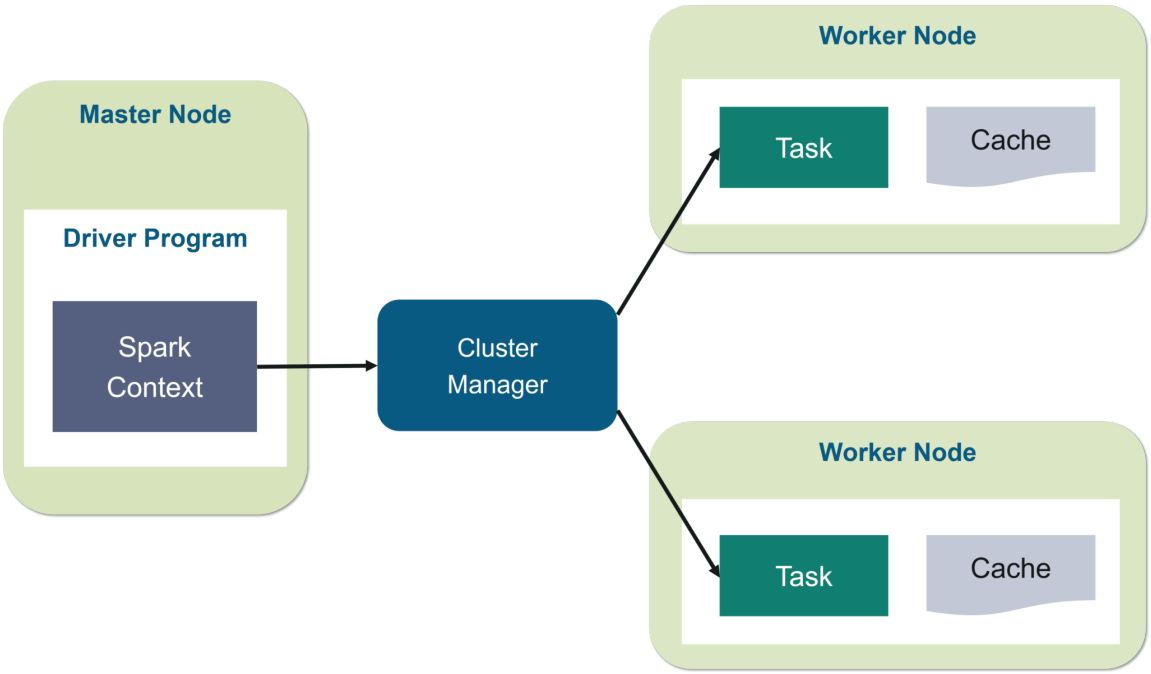
Introduction:

The World Happiness Report is a landmark survey of the state of global happiness . The report continues to gain global recognition as governments, organizations and civil society increasingly use happiness indicators to inform their policy-making decisions. Leading experts across fields – economics, psychology, survey analysis, national statistics, health, public policy and more – describe how measurements of well-being can be used effectively to assess the progress of nations. The reports review the state of happiness in the world today and show how the new science of happiness explains personal and national variations in happiness. How can we measure something like Happiness quantitatively, especially at a country level? So, a report is generated by the United Nations Sustainable Development Solutions Network, known as the Word Happiness Report. The report is based on 6 criterias, i.e., GDP per Capita, Healthy Life Expectancy, Social Support, Freedom to make life choices, Generosity, Perception of Corruption. A final score ranging from 0 to 8 is awarded to each country.

Overview of the project:

The Apache Spark is a lightning-fast cluster computing designed for fast computation. It was built on top of Hadoop MapReduce and it extends the MapReduce model to efficiently use more types of computations which includes Interactive Queries and Stream Processing.

High Level Design:



Tools Used:

- Google Colab
- Pyspark
- Numpy
- Pandas

Implementation/code:

```
import numpy as np # linear algebra
import pandas as pd # data processing, CSV file I/O

import os
for dirname, _, filenames in os.walk('/kaggle/input'):
    for filename in filenames:
        print(os.path.join(dirname, filename))

!pip install pyspark

!ls

!pwd


# **Loading the dataset**

pdf = pd.read_csv('world-happiness-report.csv')
pdf.head(2)

from pyspark import SparkContext
sc = SparkContext(appName='happiness-report')
sc

from pyspark.sql import SQLContext

sqlContext = SQLContext(sc)
sqlContext
```

```
import pyspark.sql.functions as F

#convert pandas dataframe into pyspark
sdf = sqlContext.createDataFrame(pdf)
sdf.show(5)


# **Filter**
latest_year = sdf.agg(F.max('year').alias('max_year'))
latest_year.show()

type(latest_year)

latest_year = latest_year.collect()
latest_year

latest_year = latest_year[0]['max_year']
latest_year

sdf_filtered = sdf.where("year = '{}'".format(latest_year))

sdf_filtered.show(5)

sdf_filtered.count()
```

```
# **Groupby and Aggregation**

overall_stats = sdf.agg(
    F.count("*").alias("number_of_records"), # to count the number of records in the dataset
    F.countDistinct("Country name").alias("number_of_countries"), # to get the distinct count of counties in column 'county'
)

overall_stats.show(1, False)

overall_stats_filtered = sdf_filtered.agg(
    F.count("*").alias("number_of_records"), # to count the number of records in the dataset
    F.countDistinct("Country name").alias("number_of_countries"), # to get the distinct count of counties in column 'county'
)

overall_stats.show(1, False)

sdf_filtered.orderBy("Country Name").show(10, False)

country_summary = sdf.groupBy(
    "Country name"
).agg(
    F.avg("Life Ladder").alias("life_ladder_avg"),
    F.avg("Log GDP per capita").alias("gdp_avg"),
    F.avg("Social Support").alias("social_support_avg"),
    F.count("*").alias("number_of_records"),
```

```
    F.countDistinct("Country name").alias("number_of_countries")
)

country_summary.orderBy("Country Name").show(20, False)

from pyspark.sql.functions import col

avgColumns = [col('gdp_avg'), col('social_support_avg')]

averageFunc = sum(x for x in avgColumns)/len(avgColumns)

country_overall = country_summary.withColumn('Result(Avg)', averageFunc)
country_summary.withColumn('Result(Avg)',averageFunc).orderBy("Country
Name").show(truncate=False)


# **Parition By & Window Function**

from pyspark.sql.window import Window

country_overall.show(2, False)


country_summary_ranked = country_overall.withColumn(
    "life_rank", # column name
    F.rank().over(Window.orderBy("life_ladder_avg"))
)

country_summary_ranked.show(5, False)

country_summary_ranked = country_summary_ranked.withColumn(
```

```
    "life_rank_desc",
    F.rank().over(Window.orderBy(F.desc("life_ladder_avg")))
)

country_summary_ranked.orderBy("life_ladder_avg", ascending=False).show(5, False)

country_overall = country_overall.na.drop(subset=["Result(Avg)"])

country_summary_ranked = country_overall.withColumn(
    "Overall Rank",
    F.rank().over(Window.orderBy(F.desc("Result(Avg)")))
)

country_summary_ranked.orderBy("Result(Avg)", ascending=False).show(15, False)
```

Results/ Snapshots:

Country name	life_ladder_avg	gdp_avg	social_support_avg	number_of_records	number_of_countries
Afghanistan	3.594666666666667	7.650833333333334	0.5084166666666666	12	1
Albania	5.019384615384615	9.384384615384615	0.7162307692307692	13	1
Algeria	5.389875	9.328875000000002	NaN	8	1
Angola	4.420249999999999	8.990000000000002	0.73825	4	1
Argentina	6.310133333333335	10.033800000000001	0.9044000000000001	15	1
Armenia	4.513571428571429	9.270357142857142	0.7185714285714286	14	1
Australia	7.282071428571429	10.755571428571429	0.9473571428571427	14	1
Austria	7.242230769230769	10.886846153846154	0.9295384615384618	13	1
Azerbaijan	4.941000000000001	9.519571428571428	0.7705714285714287	14	1
Bahrain	6.0017272727272735	10.730818181818181	NaN	11	1
Bangladesh	4.7544666666666675	8.1286	0.6070666666666668	15	1
Belarus	5.571071428571428	9.760071428571425	0.9067142857142857	14	1
Belgium	6.9815	10.798714285714286	0.9201428571428573	14	1
Belize	6.2035	8.8875	0.8145	2	1
Benin	4.047916666666667	7.985916666666667	0.47716666666666674	12	1
Bhutan	5.196666666666666	9.169666666666666	0.8490000000000001	3	1
Bolivia	5.733399999999999	8.8976	0.8063333333333332	15	1
Bosnia and Herzegovina	5.19076923076923	9.430769230769231	0.783076923076923	13	1
Botswana	3.9963333333333337	9.67875	0.8261666666666665	12	1
Brazil	6.620866666666667	9.583933333333333	0.8944	15	1

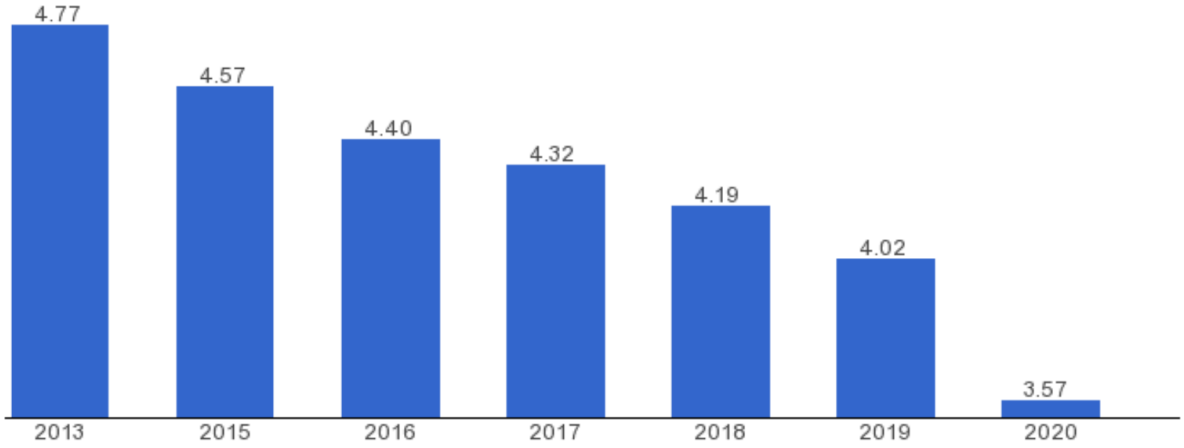
only showing top 20 rows

Country name	life_ladder_avg	gdp_avg	social_support_avg	number_of_records	number_of_countries	Result(Avg)	life_rank	life_rank
Denmark	7.6804	10.879199999999999	0.9571333333333335	15	1	5.918166666666666	166	1
Finland	7.597153846153845	10.749230769230775	0.949	13	1	5.849461538461537	165	2
Switzerland	7.5483	11.0954	0.9436	10	1	6.0195	164	3
Norway	7.512400000000001	11.0391	0.9504999999999999	10	1	5.9948	163	4
Netherlands	7.466285714285713	10.886785714285717	0.9335000000000001	14	1	5.9101428571428585	162	5

only showing top 5 rows

Country name	life_ladder_avg	gdp_avg	social_support_avg	number_of_records	number_of_countries	Result(Avg)	Overall Rank
Luxembourg	7.0471818181818175	11.607090909090909	0.9204545454545454	11	1	6.263772727272727	1
Singapore	6.504230769230769	11.328999999999999	0.8807692307692309	13	1	6.104884615384615	2
Switzerland	7.5483	11.0954	0.9436	10	1	6.0195	3
Ireland	7.067714285714286	11.064857142857145	0.9600000000000001	14	1	6.012428571428573	4
Norway	7.512400000000001	11.0391	0.9504999999999999	10	1	5.9948	5
Iceland	7.4465	10.860625	0.9775	8	1	5.919062500000001	6
Denmark	7.6804	10.879199999999999	0.9571333333333335	15	1	5.918166666666666	7
Netherlands	7.466285714285713	10.886785714285717	0.9335000000000001	14	1	5.9101428571428585	8
Austria	7.242230769230769	10.886846153846154	0.9295384615384618	13	1	5.908192307692308	9
Sweden	7.369466666666667	10.8196	0.9279333333333333	15	1	5.873766666666667	10
Germany	6.843133333333332	10.811000000000002	0.9242	15	1	5.867600000000001	11
Belgium	6.9815	10.798714285714286	0.9201428571428573	14	1	5.859428571428571	12
Australia	7.282071428571429	10.755571428571429	0.9473571428571427	14	1	5.851464285714286	13
Finland	7.597153846153845	10.749230769230775	0.949	13	1	5.849461538461537	14
United Kingdom	6.917599999999999	10.688466666666669	0.9453333333333334	15	1	5.816900000000001	15

only showing top 15 rows



References:

- https://spark.apache.org/docs/latest/api/python/
- https://spark.apache.org/
- https://www.kaggle.com/ajaypalsinghlo/world-happiness-report-2021?select=world-happiness-report.csv
- https://worldhappiness.report/ed/2021/
- https://databricks.com/glossary/pyspark
- https://realpython.com/pyspark-intro/