# (7082CEM)

# Coursework

Demonstration of a Big Data Program

MODULE LEADER: Dr. Marwan Fuad

Student Name: Manoj Sibi Mogan

SID: 10224061

**INDIVIDUAL’S ANNUAL INCOME ANALYSIS USING PYSPARK**

I can confirm that all work submitted is my own: Yes

Table of Contents

1. **Introduction**

In this paper I am going to explain and demonstrate how to setup and use PySpark realizing the entire project. The result of this project are also reported and discussed. Sophisticated platforms have started to develop and handle a massive amount of data. Spark is one of the famous distributed computation platform for big data analysis which comprises outstanding functionalities such as performing faster in large dataset, ease of use, fault tolerance and overcoming memory latency.

We start with introduction and detailed explanation of the stack of programs that are going to be used throughout this project and how to install them. Spark offers programming for different programming languages, Python has been considered for this study for its special qualities such as real time screen analytics, facilitates data visualisation and process faster on the framework. PySpark which is a combination of Spark and Python is implemented in this study for Spark data processing through PySpark which easily integrate and collaborate with RDD through navigating Py4J library. In this study Resilient Distributed Dataset (RDD) which is a primary data structure of Spark is used to split the nodes and quickly perform the calculation for the given dataset through its functions: transformation and actions. These are Python, Spark, and Tableau (Visual representation).

Data for this study is based on Individual’s Annual income which is covered worldwide. However due to time limitations part of the dataset is extracted from Kaggle dataset and used in this study for Big Data analysis and Data visualisation. The collected data contains 13 attributes which are best suited for this study. Education Qualification, Age, Occupation, Hours Per Week, Income are some of the attributes chosen for the study. After the selection of dataset PySpark was launched using number on installation steps. Next, pre-processing for dataset was completed through loading data (Loading through RDD and Dataframe). Following this, duplications were removed and columns were dropped including handing missing values. Then Dataframe operations were completed through different methods: Groupby, Distinct, Orderby, Built-in-functions, describe function and check value for specific column. This then led to exploratory analysis. The explanation for visualisations was shown in Tableau. More information about the study and results are discussed in details in the following sections.

1. **Implementation**
   1. **Background study**

**SPARK**

Spark is a distributed and highly scalable in-memory data analytics system, providing the ability to develop applications in Java, Scala, Python, as well as languages like R. It has one of the highest contribution/involvement rates among the Apache top level projects at this time. It intends to replace MapReduce model with a better solution. It emphasizes the computational efficiency of iterative and recursive algorithms and interactive queries of data mining. It claimed that it would be 10-20X faster than MapReduce for certain type of workload, such as performing iterative algorithm. Although it attempts to replace MapReduce, it did not abandon HDFS. It leverages Hadoop’s file storage system. Like many other Hadoop related projects, it is an open source project under Apache Software Foundation (ASF). Spark is a fast and general- purpose computation platform based on large clusters. In contrast to MapReduce that is basically designed for web crawler, indexing system and limited machine learning, Spark includes SQL, interactive query, data stream, graph, and machine learning analytic functions into its computation platform.

**SPARKCONTEXT**

SparkContext is the entry point to any spark functionality. When we run any Spark application, a driver program starts, which has the main function and your SparkContext gets initiated here. The driver program then runs the operations inside the executors on worker nodes. SparkContext uses Py4J to launch a JVM and creates a JavaSparkContext. By default, PySpark has SparkContext available as ‘sc’, so creating a new SparkContext won't work.

**RDD**

RDD stands for Resilient Distributed Dataset, these are the elements that run and operate on multiple nodes to do parallel processing on a cluster. RDDs are immutable elements, which means once you create an RDD you cannot change it. RDDs are fault tolerant as well, hence in case of any failure, they recover automatically. RDD operations are mainly separated into two divisions: Transformation and Actions. Transformations are the operations which are applied on a RDD to create a new RDD whereas Actions are the operations which instructs Spark to perform computation and send the result back to the driver.

**DATAFRAME**

Dataframe in PySpark is the distributed collection of structured or semi-structured data. This data in Dataframe is stored in rows under named columns which is similar to the relational database tables or excel sheets. It also shares some common attributes with RDD like Immutable in nature, follows lazy evaluations and is distributed in nature. It supports a wide range of formats like JSON, CSV, TXT and many more.

**PYSPARK**

PySpark is a great language for performing exploratory data analysis at scale, building machine learning pipelines, and creating ETLs for a data platform. Spark is a popular open source framework that ensures data processing with lightning speed and supports various languages like Scala, Python, Java, and R. It then boils down to your language preference and scope of work. The goal of this post is to show how to get up and running with PySpark and to perform common tasks.

The diagram below shows how both PySpark and Spark contexts are managed by Spark driver with the aid of local file system and through communicating with Spark worker through cluster manager.

Diagram

Description automatically generated

**2.2 Dataset**

An individual’s annual income results from various factors. Intuitively, it is influenced by the individual’s education level, age, gender, occupation, and etc. This dataset which is from US Census Reports (1994 – 1996). Therefore, a version of this dataset has been extracted from Kaggle dataset and used in this study for Big Data analysis and Data visualisation. This chosen dataset includes information on Education Qualification, Age, Occupation, Hours Per Week, Income etc. and demographics. There are 13 attributes and contains 32, 561 number of records in this study. The main aim of this study is to analyse income status of different employees through analysing the factors such income, educational qualification, working sector etc.

The table below shows the attributes from the dataset including its description and data types.

|  |  |  |
| --- | --- | --- |
| **Attribute** | **Description** | **Data Types** |
| Age | Age of the employee | integer |
| Work Sector | Employee working sector | string |
| UID | Employee Unique identification number | integer |
| Education Qualification | Employee educational qualification details | string |
| Education.Num | Employee educational qualification grade | Integer |
| Marital Status | Employee Marital details | string |
| Occupation | Employee work nature | string |
| Relationship | Employee’s relationship with their family | string |
| Capital.gain | Annual capital gain for the employee | integer |
| Capital.loss | Annual capital loss for the employee | integer |
| Hours.per.week | Hours worked by the employee in a week | integer |
| Native.Country | Country to which the employee belongs | string |
| Income | Annual income of the employee | integer |

**2.3 Installation Steps**

Machine used to perform this study has the hard disk drive with the capacity of 1TB, RAM with 16GB and the used operating system is macOS Catalina version 10.15.7. The steps installation steps followed are explained below:

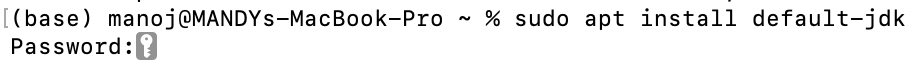
**Step 1:** Initially, it is important to check whether java is installed within the machine through looking at the current java version by entering the below mentioned command in Terminal. If the java is installed it will display the below message.

**Text

Description automatically generated**

And if the java is not installed below mentioned command needs to be used followed by the system password.

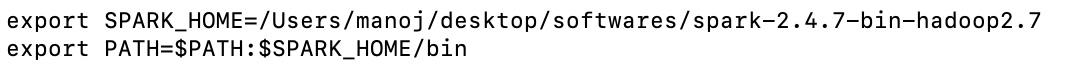
*sudo apt install default-jdk*



**Step 2**:- Download the latest version Spark from (<https://spark.apache.org/downloads.html>) and save the zip folder in the desktop or any folder. Enter the below command in the terminal to unzip the Spark file.

*tar -xzf spark-2.4.7-bin-hadoop2.7.tar*

**Step 3**:- under bash profile setting up the Spark Environment by using the command shown below.



**Step 4**:- To ensure whether the Spark is installed correctly with the below command.

*spark-shell*

Graphical user interface, text, application, letter, email

Description automatically generated

**Step 5**:- After the installation of Spark, need to install Python, Jupyter Notebook by downloading Anaconda (<https://www.anaconda.com/products/individual#macos>) and install it.

**Step 6**:- After the installation of Anaconda, it is important to check whether Python is installed in the machine. The following command will be used to check the Python version.

*python -version*



**Step 7:** After checking the Python version make sure whether Jupyter notebook is successfully installed the following command is used.

*jupyter notebook*

Graphical user interface, text, application

Description automatically generated

**2.4 Pre-processing the Dataset**

Dataset chosen for this study is not clean. In other words, it includes missing values, null values and duplications. Therefore, these inappropriate values are removed through pre-processing step.

**Reference Link:**

Spark – Mastering Apache Spark by Mike Frampton

Spark Content - <https://www.tutorialspoint.com/pyspark/pyspark_sparkcontext.htm>

RDD - <https://www.tutorialspoint.com/pyspark/pyspark_rdd.htm>

Data Frame - <https://www.edureka.co/blog/pyspark-programming/>

PySpark - <https://www.edureka.co/blog/pyspark-programming/>

PySpark Flow Diagram - <https://www.researchgate.net/figure/An-illustration-of-PySpark-data-flow-The-Python-environments-were-shaded-in-white-and_fig2_325055629>

Dataset - <https://www.kaggle.com/wenruliu/adult-income-dataset>

Spark Software: <https://spark.apache.org/downloads.html>

Anaconda: <https://www.anaconda.com/products/individual#macos>