**Group\_9\_Task 5**:

Identifying the principal components for the input data set

**Approach:**

Here we have used Data Frame, PCA, and standard Scalar among spark’s ML APIs for principal component analysis.

For this, first, we need to read the input data file into the data frame. Since the input to the PCA is a variable in a vector format, we need to convert the data frame into the vector using vectorAssembler and the output is a principal component score. For better results for principal component analysis used Standardization. There are two stages, one is the PCA model, and the other is actual calculation. Both the Eigenvector and contribution rate can be obtained from the PCA model. Here we have calculated up to the third principal component.

**Dataset:** We have used the input dataset as ‘marketing\_campaign.csv’ which we have written from the input datapath ‘/user/kaggle/kaggle\_data/marketing\_campaign.csv’

**Input Columns:** Standard variable

**Output Column:** Main component score, Contribution rate, and Eigenvector of PCA model

* We have imported all the required libraries like vectorAssembler, and PCA then read the input data set from CSV file to date frame.
* Then converted the strings in the data frame column into integers using StringIndexer and One hot encoder.
* Since PCA will take the input variable in vector format and output the main component score, we have used the vectorAssembler to convert the data frame into a vector format.
* To get a better result for principal component analysis, standardization is done using a standard scalar. It creates the model from the input variable which is from the vectorAssembler with the fit, the input data again with transform and outputs the standard variable.
* Once we have the standardized variable from the standard scalar, then call the principal component analysis (PCA) API with input as the standard variable, and the PCA will output the principal component score. In PCA, k will tell us up to what principal component the model is calculated. Here, we have calculated up to the third principal component i.e., k=3. PCA model is created by fitting the standardized vector.
* The model’s pc values give us the Eigenvector and explainedVariance gives the contributions rate. By selecting the output of the PCA model we will get the main component score

**Instructions to compile and run the program:**

**Imports Used:**

import sys

from pyspark import SparkContext, SparkConf

from pyspark.sql.types import IntegerType

from pyspark.ml.feature import StandardScaler

from pyspark.ml.feature import PCA

from pyspark.sql.session import SparkSession

from pyspark.ml.feature import StringIndexer

from pyspark.ml.feature import OneHotEncoder

from pyspark.ml.linalg import Vectors

from pyspark.ml.feature import VectorAssembler

As mentioned, we have used the input dataset as ‘marketing\_campaign.csv’ which we have written from the input datapath ‘/user/kaggle/kaggle\_data/marketing\_campaign.csv’

Output: Output is displayed on the console.

1. Open csx and login into the cluster.

2. Upload the Group\_9\_Task\_5.py file into session storage (local or csh)

3. Command: spark-submit Group\_9\_Task\_5.py

**Discussion of Results:**

PCA API will only take the vector format, for we have converted the input data frame into the vector using vector assembler and if some strings are present, we have converted it using StringIndexer and oneHotEncoder. Generally, the principal component analysis gives better results after standardizations. For that, a standard scalar is used and fed that standard variate to the PCA. Eigenvector, contribution rate, and main component score are obtained with the PCA model.

Reference:

<https://linuxtut.com/en/7714a1cc9be56588b72a/>