Karla Dominguez Melo

Stanislav Mykhaylichenko

Xingyi Li

Japman Kaur Dhaliwal

CISC5950 Big Data Programming

05/08/2020

**Project 2 Part 4**

This project was developed and tested using a local cluster consisting of 3-nodes with Hadoop Distributed File System, running Ubuntu 16.04 LTS in GoogleCloud.

In this project, we are going to use a spark cluster for data processing jobs. The project consists of 4 parts.

For Part 4, we will analyze the Census Dataset from the UCI Machine Learning Repository. It provides the following data, 1. Training Set: adult.data 2. Test Set: adult.test 3. Data Description: adult.name.

The data contains anonymous information such as age, occupation, education, working class, etc. The goal is to train a binary classifier to predict the income which has two possible values > 50K and < 50K. There are 48842 instances and 14 attributes in the dataset. The data contains a good blend of categorical, numerical and missing values.

**Introduction**

In this part, we were asked to re-do Part 3, but instead of using the logistic regression algorithm to train the model and evaluate it to use both Random Forest Classifier and Decision Tree Classifier. All algorithms used come from the Apache Spark ML/MLlib.

**Data Selection**

Both the training and the test data didn’t contain headers, but to ease the readability of the code, we added the headers from the description to both sets.

**Analysis**

To complete this task, we began by starting a SparkSession. We used both the training and test set as input.

Before training the model, we needed to perform some pre-processing techniques. The data contained a number of categorical features, such as occupation and working class, that needed to be encoded to numbers for the algorithm to work. For this task we used StringIndexer, which converts strings into indices, combined the StringIndexer for each categorical column and used Pipeline to execute them all. Once this was done, we dropped the empty values and the columns we didn’t need, like the original categorical columns and left only the numerical features.

We used VectorAssembler to combine the features into a single feature vector, leaving out the class variable we’re trying to predict.

After both data sets were prepared, we used the Decision Tree algorithm and used the traindata for training the model. This model was used to predict the class for both train and test data. After the prediction was done, we were able to find both the train and test accuracy.

Finally, using the traindata and testdata without the Decision Tree prediction, we used the Random Forest algorithm and used the traindata for training the model. This model was used to predict the class for both train and test data. After the prediction was done, we were able to find both the train and test accuracy.

**Result**

The output for the program above was the following:

