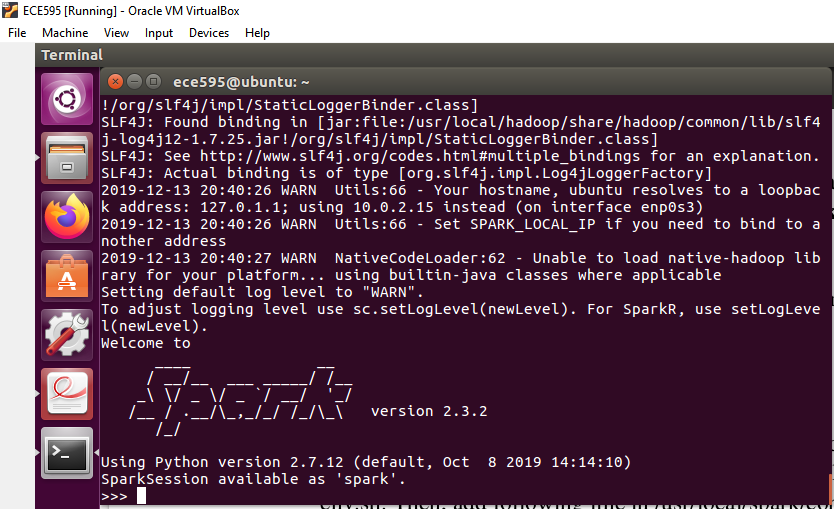
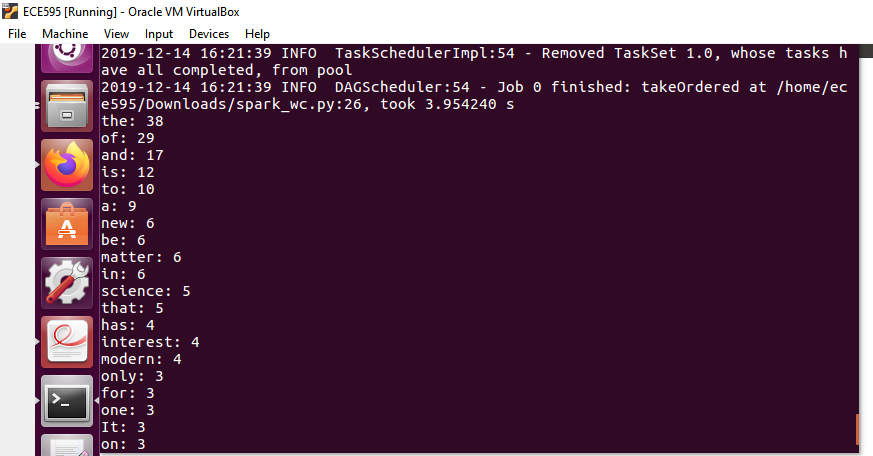
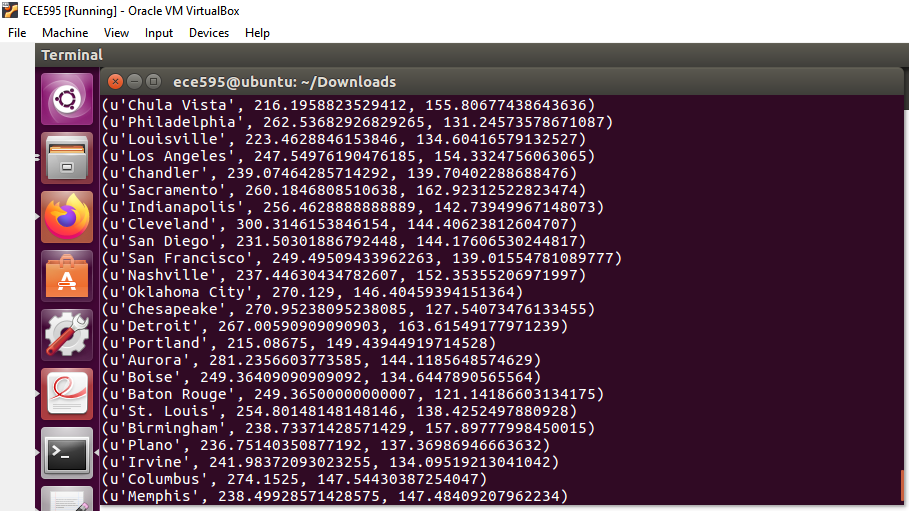
Electrical and Computer Engineering, Purdue University NorthwestBig Data (ECE 595), Fall 2019Hands-on Assignment 3, Due Date: 12/14/2019Objective: Spark basic programing, Spark SQL, and Spark MLLib  
Note: For grading, submit a single PDF file for the report and code files should be uploaded  
separately in the submission.  
Task 1 [4 points]: Spark Installation



**Task 2 [6 points]**: Write Spark application that reads uploaded **words.txt** file from HDFS cluster  
and finds top 10 most frequent words and their frequencies. In the text file, a few words may appear  
in different forms, e.g. The, the, you have to treat them same. In addition, some words may have  
double quote, single quote or other non-alphabet character in the prefix or suffix, your program  
should be able to remove them and then consider the remaining characters as word. Implement  
this program through RDD **transformation** and **action** operation. You may start with uploaded  
skeleton code **spark\_wc.py** for word count program. To run your spark application, execute  
**spark-submit <your Spark Python file name>.** Please use **firstname\_lastname\_task2.py**format for naming the program file.



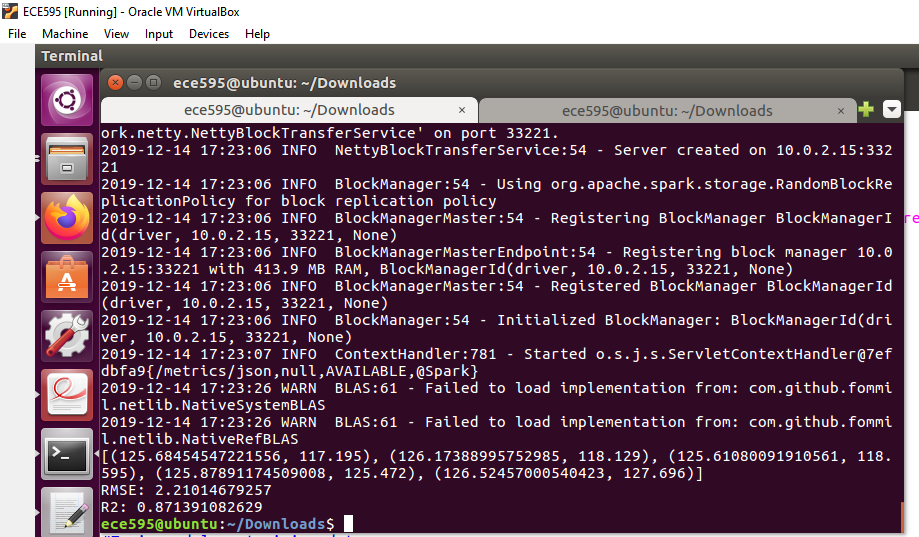
**Task 3 [7 points]**: Write Spark application that reads uploaded **sales.txt** file from HDFS cluster  
and finds average and standard deviation of stores’ sales in each city. Implement this program  
through Spark Dataframe and Spark SQL. You may start with the uploaded skeleton code  
**spark\_std.py.** To run your spark application, execute **spark-submit <your Spark Python file  
name>**. Please use **firstname\_lastname\_task3.py** format for naming the program file. You may  
refer dataframe slides and following link: <https://www.analyticsvidhya.com/blog/2016/10/sparkdataframe-and-operations/>



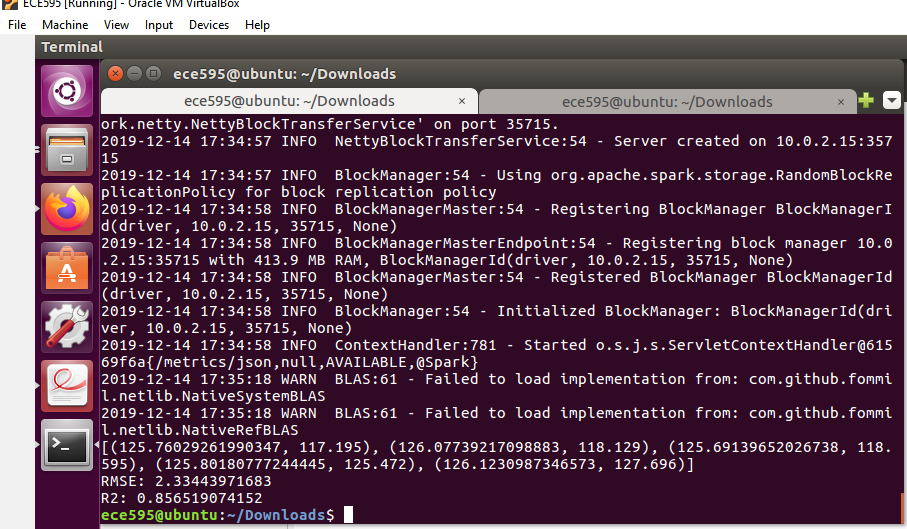
**Task 4 [5 points]**: Upload **airfoil.txt** file in HDFS cluster and run **spark\_ml\_linreg.py** Spark  
application. The application is Linear Regression implementation in Spark. The dataset used in  
this task is taken from here. You can visit the link to get the description of the dataset. Run the  
application by varying parameters like **maxIter**, **regParam**, and **elasticNetParam**. Mention your  
observation in the report. You may refer the following link:  
https://www.datacamp.com/community/tutorials/apache-spark-tutorial-machine-learning  
To run your spark application, execute **spark-submit <your Spark Python file name>.** Please  
use **firstname\_lastname\_task4.py** format for naming the program file.

Conclusion:

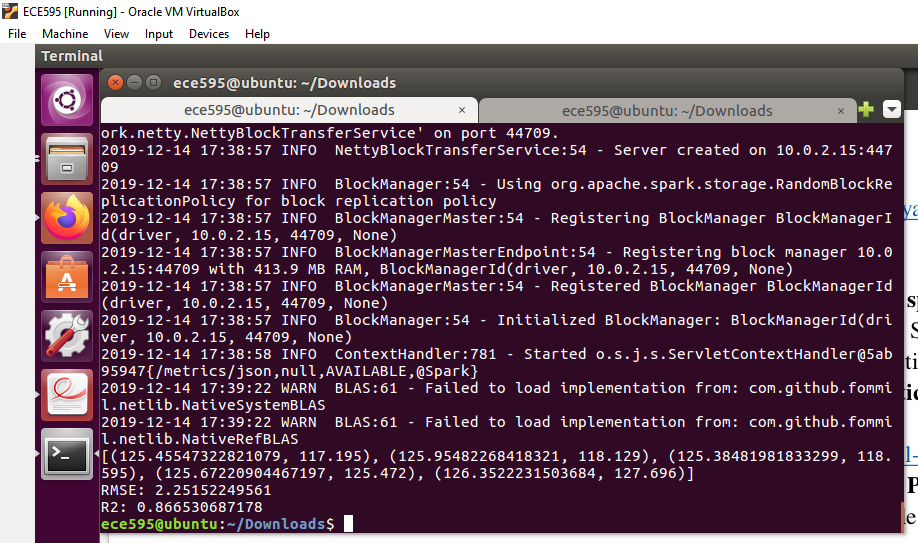
I have varied maximum Iteration from 10 to 20 first and here is the output.



Then I changed regularization parameter to 0.6 from 0.3 keeping maximum iteration to 20. Here is the output.



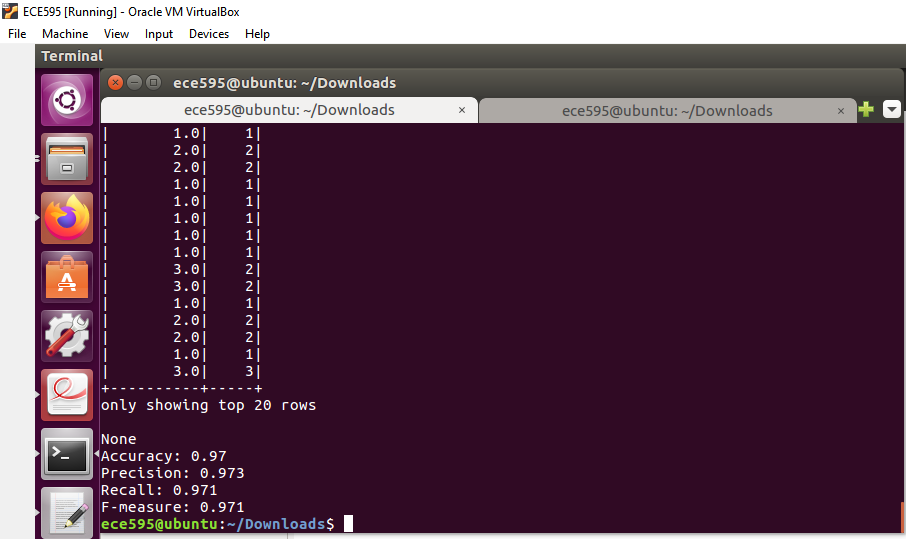
After that I changed elasticnet parameter to 0.16 from 0.8 and here is the output.



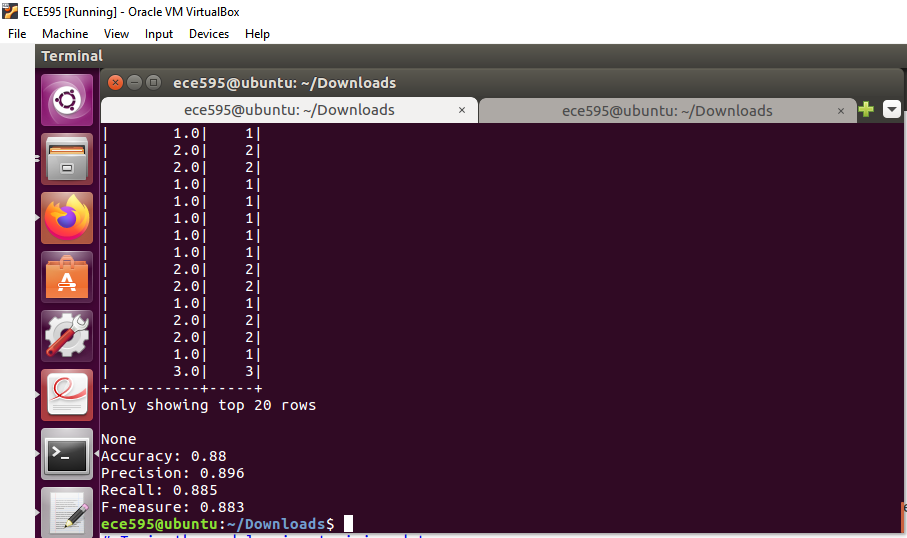
Therefore we can say that by increasing maximum iterations our R-squared value is improved but increasing the other 2 parameters our RMSE also increased and overall R2 value decreased.

**Task 5 [3 points]**: Upload **iris.txt** file in HDFS cluster and run **spark\_ml\_lrclf.py** Spark  
application. The application is Logistic Regression implementation in Spark. The dataset used in  
this task is taken from here. You can visit the link to get the description of the dataset. Run the  
application by varying parameters like **maxIter** and **regParam**. Mention your observation in the  
report. To run your spark application, execute **spark-submit <your Spark Python file name>.**Please use **firstname\_lastname\_task5.py** format for naming the program file.

First by changing maxiteration from 10 t0 20. Here is the output.

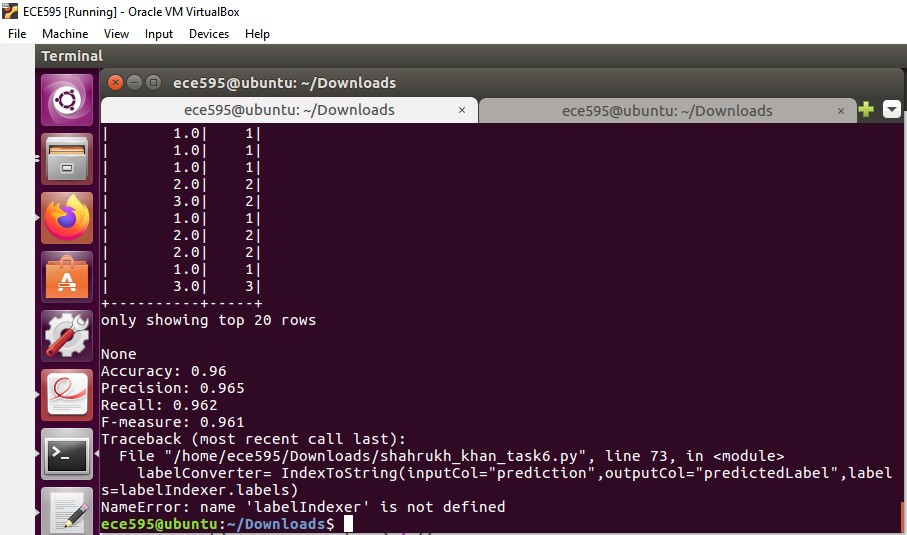


Then , changing regularization parameter from to 0.5 keeping max iteration to 20, here is the output.



We can say that by increasing max iterations and regularization parameter, our accuracy,precision, recall and f-measure all decreased.

**Task 6 [5 points]**: Use the uploaded **iris.txt** dataset in HDFS cluster and build a Random Forest  
classifier model. You can use **spark\_ml\_rfclf.py** for this task. Compare your results with Logistic  
regression implementation. Please use **firstname\_lastname\_task6.py** format for naming the  
program file.



In the conclusion we can say that there is no much change in the accuracy ,precision and recall values however, the values of these measures were higher in logistic regression.

**Task 7 [30 points]:** Implement K-Nearest Neighbors from scratch using the skeleton code given  
in **spark\_ml\_knnclf.py**. We have provided guidelines in the file to accomplish the task. The  
implementation is discussed using Spark dataframe and Spark SQL features and designed for  
**iris.txt** dataset only. Please use **firstname\_lastname\_task7.py** format for naming the program  
file.

