MSBA 6330 Harvesting Big Data: Project #2

12/15/2015

Analysis of US College Scorecard Data

Analytics with big data

Shuo, Rohan, Wenqiuli, Kaushik

I) INTRODUCTION

The U.S. Department of Education has released College Scorecard data, which provides information about colleges' over years (year 1996 to 2013). The data provides detailed information about various features of undergraduate schools in US such as average SAT score, admission rate, student employment on campus, percentage of students graduating within 4/6 years, percentage of students by race, fellowships, grants, tuition fees, earnings, ability to repay loan, eligibility to receive federal financial aid, etc. Considering the size of dataset, its high dimensional features and a goal to derive insights, use of big data technologies in this scenario made a perfect sense.

The insights derived from this data could help measure the performance of colleges. These insights can be helpful in scenarios as mentioned below:

- Students or parents who need information about colleges during application. The performance of college over years can help them predict answers to several questions such as - How good is the college. Can I afford it over 4 years of graduation? What are my chances of getting fellowship/grants? Can I get into Stanford with my SAT score?
- Banks which offer loans to students might need college information. The insight about performance of college can help Banks predict the credibility of loan provided to students in particular college (probability that they will repay or will default).
- Government (education department) may consider investment in some underperforming colleges

This report starts with a brief data description, followed by methodology, analyses and conclusions.

II) BACKGROUND

The dataset we analyzed was obtained from U.S. Department of Education's College Scorecard Data website and contains information about undergraduate colleges from 1996 to 2013. The data is very rich in terms of documentation and effectively provides

explanation of weights and measures used in the features. The motivation behind choosing this data set was its high dimensional feature space. It consists of 1954 features (columns) and 124700 rows, making the total data of size 1.5GB and a perfect candidate to apply Big Data Technologies.

To explore various Big Data techniques, we started with measuring the performance of Hive, Impala and Spark. Since, Spark outperformed, we decided to use it and built our transformations and queries in Scala programming language. In parallel, we used Mahout for cluster analysis to find any interesting patterns or insights in the data. To get a better performance (in terms of speeds and high computing resources) we used Amazon Web Services – EMR with one master node and four core nodes.

III) METHODOLOGY

1. **Performance of Hive/Impala/Spark**
    a. Performance (in terms of speed of execution) of these big data technologies was measured with respect to below questions. These questions form aggregate queries and hence can act as a benchmark for measuring the performance (aggregate queries use most of the system resources as they involve selection, projection, join and group-by operators)
        i. Extract average SAT score of students (**SAT_AVG)**.
        ii. Extract admission rate for universities (**ADM_RATE**).
        iii. Extract In-state tuition and fees (**TUITIONFEE_IN**)
        iv. Extract Average Faculty Salary(**AVGFACSAL**)
        v. Extract Repayment Rate(**RPY_3YR_RT)**
    b. Create and execute scripts on Hive/Impala/Spark.
    c. Measure the time. To remain consistent with hardware for measuring performance, we used **Amazon EMR** with a master node and four core nodes.
2. **Data Preprocessing**
    a. Combine all .csv files into one file - *Scorecard.csv*. Add a column ***Year*** to the file which can be used in later data processing steps (filter conditions). The data in this field would correspond to the year appended in merged file names. We created a python script to merge these files.

b. Replace all NULL by empty strings.
c. Clean the data by removing noise such as *'Privacy Suppressed'* entries for particular fields. '*Privacy Suppressed'* is present in fields where data is not provided by corresponding colleges for privacy reasons.
d. Generate CREATE TABLE commands for Hive, Impala and Spark SQL. *As the dimensionality of data is very large, creating manual CREATE TABLE statements isn't feasible. We wrote a* ***python script*** *to generate CREATE TABLE Scripts for hive, Impala and Spark*

3 Spark Implementation

For Spark, we used Scala programming language and spark-csv package. Spark shell can be started as:

```
spark-shell --packages com.databricks:spark-csv_2.10:1.1.0
```

a. Create Spark SQL context. This is created on top of Spark Context (org.apache.spark.SparkContext), which is required by spark-csv package and Spark SQL query execution.

```
val sqlContext = new org.apache.spark.sql.SQLContext(sc)
```

b. Similarly, create HiveQL context. This is created on top of Spark Context (org.apache.spark.SparkContext)
c. Create table scorecard.
d. Load data into RDDs. Perform operations like transform, filter and group by on this data. This involves operations like casting of columns from STRING to FLOAT or STRING to INT, etc.
e. Create .scala scripts for extracting data for analysis.
    i. Scripts for Yearwise results
    ii. Scripts for Statewise results
    iii. Scripts for Trend Analysis.
f. Execute the scripts. After execution of scripts, data is directly written into local folder in csv format with use of the **spark-csv** package. Data would be generated in similar fashion as other MapReduce jobs and stored in part-00* files.

g. Once data is extracted successfully, combine data from all folders into one .csv file for each folder (We used a bash script that would merge all part-* files from respective folders into one file (for each folder).
h. Use the data extracted for Exploration and Insights

4 Mahout Cluster Analysis

We used Impala to create the data for cluster analysis. *impalaExtractionForMahout.sql* provides more details about the whole dataset, with respect to how we selected only 2013 data (the most recent year), and how we chose the following attributes. We were interested in these variables as they do not have too many missing values, which is the primary requirement for cluster analysis (Mahout, K-Means Clustering using Mahout, 2014). Attributes used for descriptive analyses are:

Attributes	**Definition**
SAT_AVG	Average SAT Score
ADM_RATE	Admission Rate of University
TUITIONFEE_IN	In-state tuition Fees
AVGFACSAL	Average Faculty Salary
C150_4_White	Completion rate for white students
C150_4_Black	Completion rate for black students
C150_4_Hisp	Completion rate for Hispanic students

a. We output this dataset into a local folder and set the field delimiter as space, as required by Mahout clustering analysis.

```
impala-shell -i localhost -f mpalaExtractionForMahout.sql -o mahoutClusterInput.txt --output_file_field_delim=' '
```

b. We created a folder in hadoop to put the dataset into HDFS
c. We converted this file into a sequential file which is required by Mahout clustering analysis.

```
mahout org.apache.mahout.clustering.conversion.InputDriver
```

d. We run clustering analysis by Mahout using this file (with all default settings and

k=3), and output the result into a local folder. You can find detailed commands in our uploaded code folder.

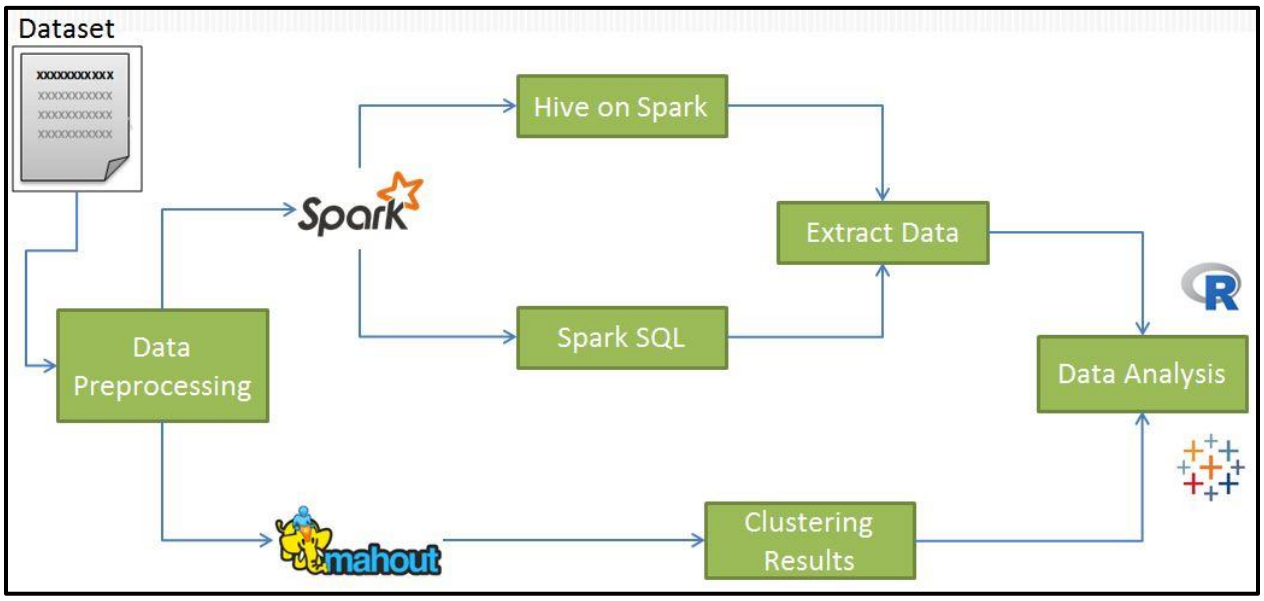

Figure 1 Workflow

IV) FINDINGS

I. Hive vs Impala vs Spark Speed comparison

Attributes	Spark	Hive	Impala
SAT_AVG	4.3	27.45	5.2
ADM_RATE	5.68	27.98	6.45
TUITIONFEE_IN	6.24	28.5	7.87
AVGFACSAL	7.11	29.3	8.02
RPY_3YR_RT	6.52	28.74	7.9
Total running time(seconds)	29.85	141.97	35.44

Table 1: Speed Comparison across different tools

This result confirms that Impala is faster than Hive, as Hive relies on MapReduce jobs while Impala does not. Spark dominates Hive and Impala in terms of speed, since it stores in

memory. Another important finding we can notice is that speed of Impala is competitive to Spark SQL (not overall Spark). This is because it has daemons running on the nodes (in Hadoop) which cache some of the data that is in HDFS, so that these daemons can return data quickly without having to go through a whole Map/Reduce job. Though competitive, Impala lacks several features that Spark has (such as Hive on Spark and other Hive capabilities - SerDe), and it is suitable for environments which deal with just query processing and extraction.

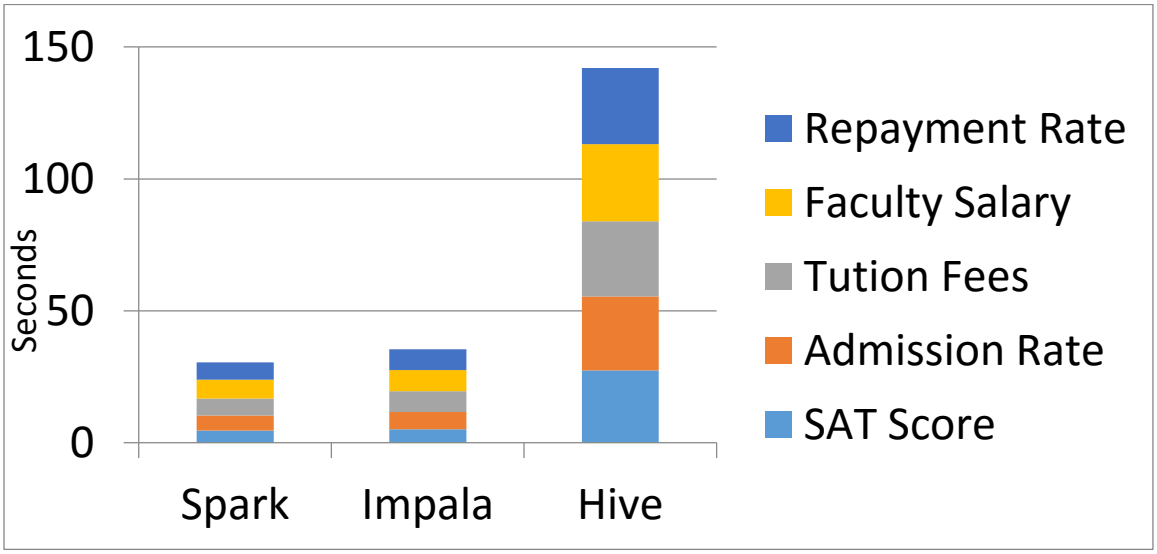

Figure 2 Performance of Hive vs Impala vs Spark

II. Analyses

All the data required for our analyses was queried using Spark. Data Analysis and Visualizations have been done using R, Excel and Watson Analytics.

Choosing a school becomes challenging as the trend in the number of schools has been increasing over the past 18 years. Fig.3 shows the count of schools between 1996 and 2013.

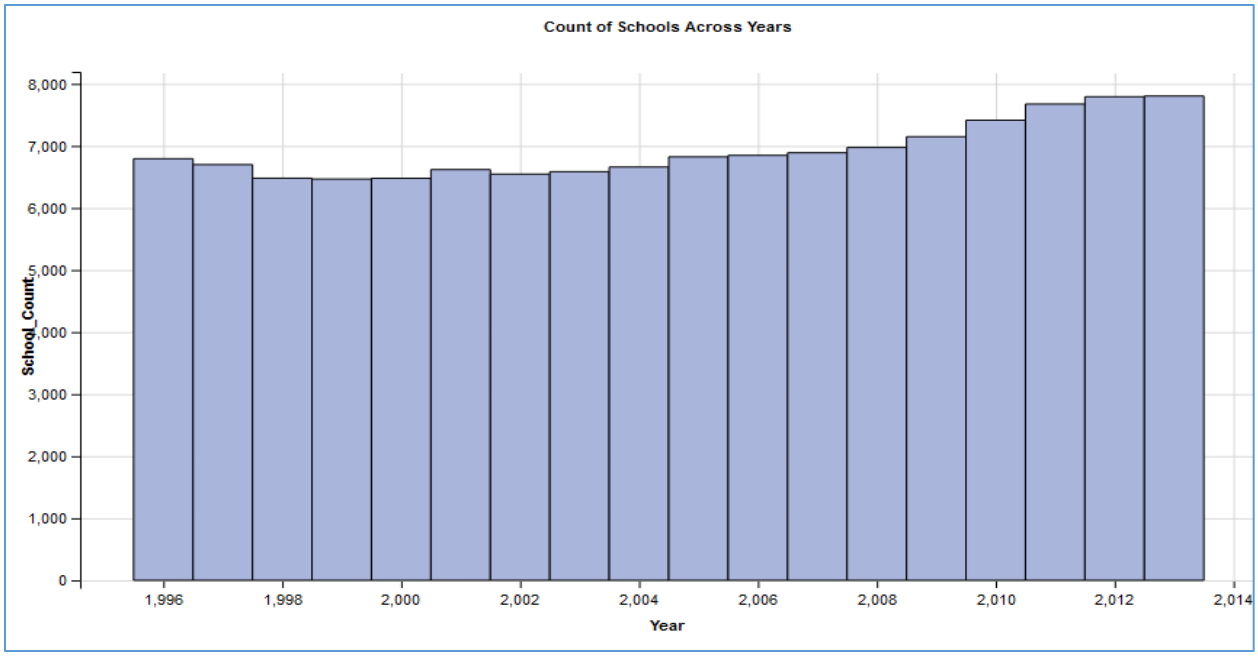

Figure 3: Trend in the number of schools over years

1. How do earnings vary with respect to colleges' control: Colleges are categorized into three types based on the controlling authority: 1) Public 2) Private for Profit 3) Private Non Profit. We wanted to identify if the earnings vary with respect to control of the college.

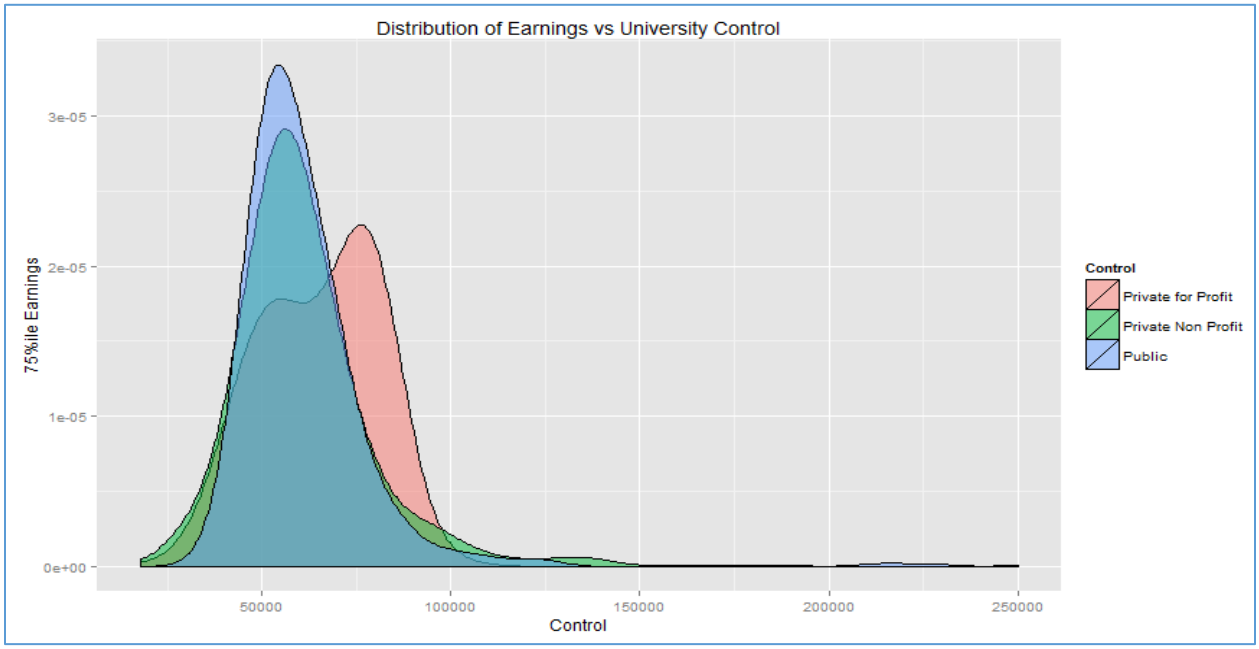

Figure 4: Distribution of earnings vs college control

As there are many NULL values for earnings in the year 2012 and 2013, we considered 2011 earnings data for our analysis. Here is a density plot (Fig.4) of 75 percentile earnings 10 years after graduation (and not enrolled in any other institution). One observation is that the median for Public and Private Non Profit schools appear very similar. However, the distribution of Private for profit schools is somewhat to bimodal.

2. What is the performance across States?

To understand which states have better repayment rates (proportion of students who have not defaulted on loan payments) we plotted the following heat map (Fig.5) It shows (based on color intensity) Vermont (VT), Nebraska (NE) and North Dakota (ND) have the highest repayment rates across all states. However, these states have relatively less number of colleges compared to other states.

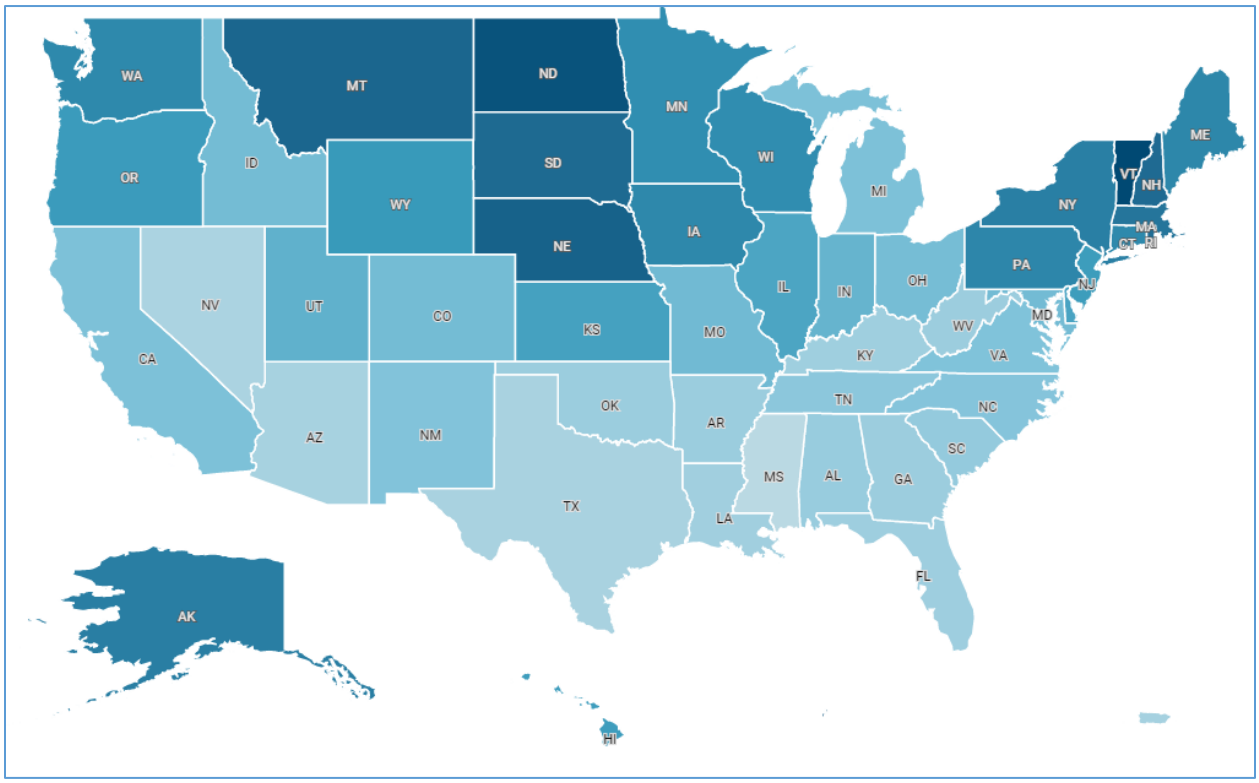

Figure 5: Heat Map with repayment rates across states

3. Trend Analysis:

a) SAT Score Trend: We looked at how average SAT scores changed for the top 5 colleges (based on average SAT score) in 2013. From the figure (Fig.6), the average SAT scores are

slightly increasing. Moreover, there is an abrupt increase in the University of Chicago's SAT scores between 2001 and 2013.

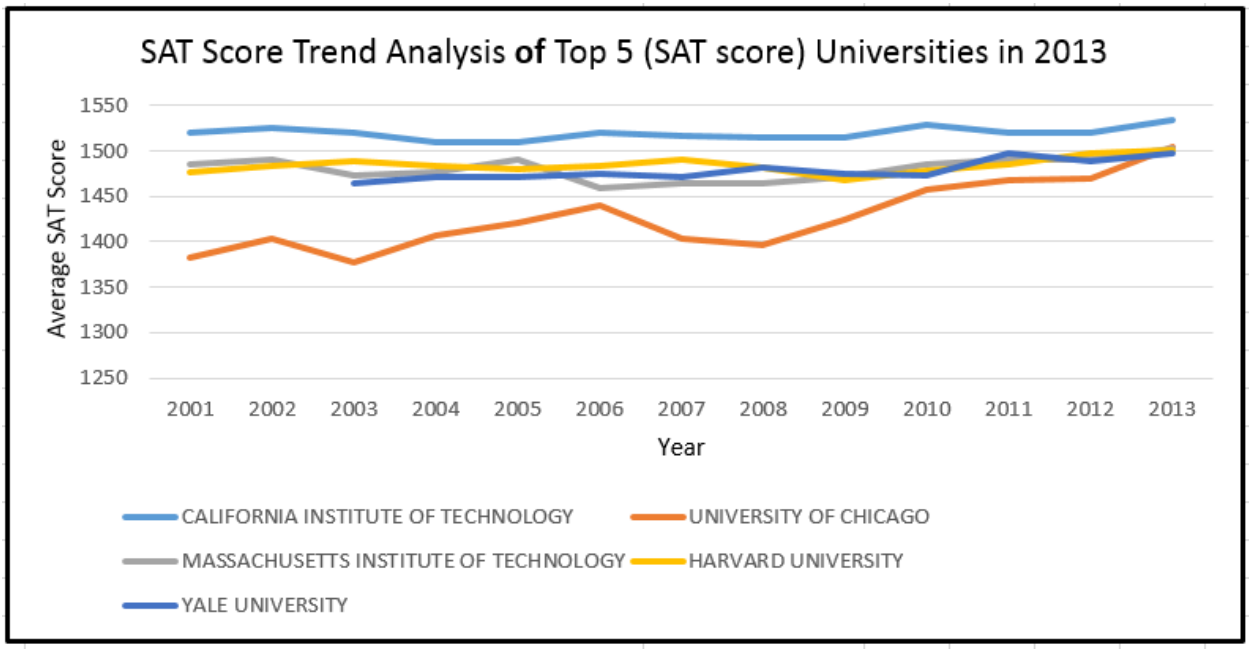

Figure 6: SAT score trend for the top 5 colleges

b) Trend based on Control: From the figure (Figure 7: Count of colleges by control), the number of Private for Profit colleges increased with a high frequency. There was a 43% increase between 1996 and 2013 in the number of Private for Profit colleges, while the number of Public and Private Nonprofit colleges remained stable over years.

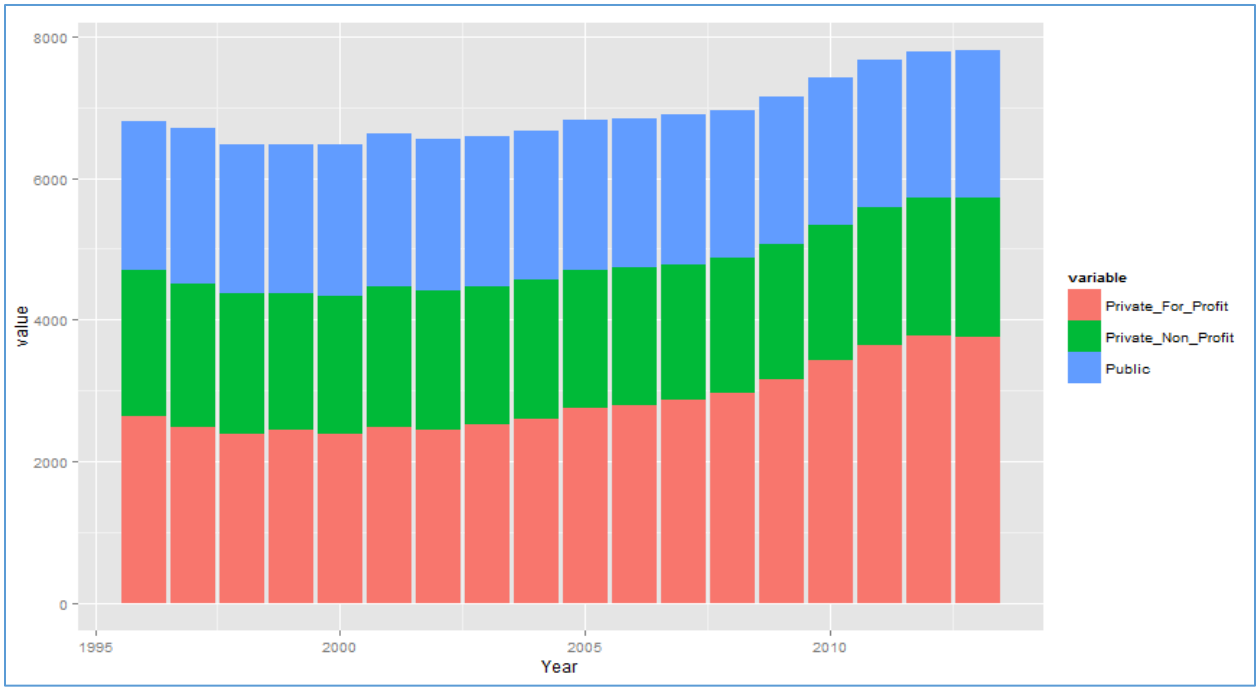

Figure 7: Count of colleges by control

4. Cluster analysis using Mahout:

Below are the centroids of each cluster

n	SAT_AVG	ADM_RATE	TUITION FEE_IN	AVGFACSAL	RPY_3YR_RT	C150_4_White	C150_4_Black	C150_4_Hisp
260	1191.38	0.566	38983.65	9350.8	0.905	0.741	0.621	0.678
454	1023.74	0.668	25749.37	6382.38	0.816	0.547	0.372	0.442
566	1031.36	0.67	9095.09	7672.39	0.782	0.508	0.373	0.432

Table 2: Interpreting different clusters

Cluster 1 comprises of "best schools", as they have highest average SAT score for students admitted and highest completion rate for all White, Black, and Hispanic students. Also, these schools have the lowest admission rate, highest tuition fees (might be private schools), and highest average faculty salary, which may explain a bit as to why these are best schools. The first cluster also has the smallest size (260), which indicates that best schools are rare.

V) SUMMARY AND CONCLUSIONS:

In this project we analyzed US colleges' performance by processing data using Spark, Impala, and Hive. Given the high dimension of the original dataset, we created python

scripts to merge files and generate tables in Spark, Impala, and Hive. We also compared the speed of Spark, Impala, and Hive. To remain consistent with hardware for measuring performance, we used Amazon EMR with a master node and four core nodes. We aggregated data and extracted different attributes (using Spark) for every year to perform certain college level, state level and trend analyses. In parallel, we used Mahout for cluster analysis, to find patterns in colleges' performance. Here are our primary conclusions:

- There is an increasing trend in the number of colleges (Figure 3)
- States Vermont (VT), Nebraska (NE) and North Dakota (ND) have the highest repayment rates (Figure 5).
- The number of Private for Profit colleges has increased by 43% (b/w 1996 and 2013) while the number of Public and Private Nonprofit colleges remained stable over years (Figure 7)
- Best colleges are very few (colleges in cluster number 1), and they have low admission rate, high tuition fees, and high average faculty salary.

VI) LESSONS LEARNT AND FUTURE DIRECTIONS:

We tried different big data methodologies (Hive, Impala, Spark, AWS, and Mahout) and became well versed with these technologies towards the end of our project. However, we faced a couple of challenges while working on the project. One interesting fact we noticed was that Impala has competitive performance with respect to Spark SQL (not overall Spark). This is because Impala has daemons running on all the nodes which cache some of the data that is in HDFS and return it quickly whenever a table is queried. Though competitive, Impala lacks several features that Spark has (such as Hive on Spark and other Hive capabilities - SerDe), and it is suitable for environments which deal with just query processing and extraction.

Also, it is better to combine other techniques such as Python and R to support our work in Big Data, since they have their own advantages. An example would be – the machine learning and data analysis part in R is very much mature as compared to the upcoming Big Data Technologies. Thus, we can utilize its potential by combining on the data extracted from Big Data Technologies. Though RHadoop and RSpark have been launched, they haven't yet met the capabilities of what R can do. Future work might include applying various classification/regression models on this rich dataset to analyze patterns and predict

values such as which college is better in terms of ROI, with a given SAT Score, tuition budget, race and other user background, what is the chance of admission into a college and getting a Pell Grant/fellowships, etc.

VI) BIBLIOGRAPHY:

Mahout. (2014). *K-Means Clustering using Mahout*. Retrieved from Apache Mahout Documentation: https://mahout.apache.org/users/clustering/k-means-clustering.html