Student VISA Prediction Based on Student’s Profile

Logo

Description automatically generated

**NAME OF THE PARTICIPANTS**

|  |  |
| --- | --- |
| Suhas Siddarajgari Tellatakula | 11626111 |
| Sai Tejesh Gonemadatala | 11608671 |
| Sai Praneeth Reddy Avula | 11582402 |
| Sai Rohith Varma Kantem | 11606743 |

**UNDER THE GUIDANCE**

**Dr. Zeenat Tariq**

PROF. OF THE DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

University Of North Texas, Denton

**INDEX**

|  |  |
| --- | --- |
| TOPICS | PAGE NO |
| ABSTRACT | 3 |
| MOTIVATION | 3 |
| SIGNIFICANCE | 3 |
| OBJECTIVE | 3 |
| FEATURES | 3 |
| WORKFLOW | 4 |
| IMPLEMENTATION | 5 |
| DATA PRE-PROCESSING | 28 |
| ANALYSIS AND PRELIMINARY RESULTS | 38 |
| References / Bibliography | 40 |
| Project Management: | 40 |

**ABSTRACT**

Every year thousands of students appear for visa interviews to study in universities abroad. And only a handful of them clears the visa interview. The student would never know the reason for rejection. Often the visa officer evaluates the student’s profile and issues a visa. If there are any drawbacks in the student profile then there is a high chance that the student might face rejection. The application we developed will evaluate students’ profiles based on the criteria like student scores, work experience, Family background, etc., and outputs the chance of getting a visa approved in percentage. The application also suggests where the student is lacking. Thus, helping in building the student profile to increase the chance of getting approved.

**MOTIVATION**

The number of students planning to pursue higher studies abroad every year is increasing rapidly. But unfortunately, only a few of them manage to get their visas approved by their desired country’s embassy. This is because of a lack of knowledge of the criteria the embassy looks up to. So, we are trying to build a solution that takes different information about the student like their test scores, financial ability, and other relevant factors, and predicts the chances of his/her visa approval.

**SIGNIFICANCE**

This tool can be extremely useful when one can check their chances of visa approval based on their profiles. They can always plan and build their profiles by the factors which will help them get their Visa approved based on the results shown in this tool. This helps a candidate decide when to plan and apply for a visa without wasting their time and money.

**OBJECTIVE**

This tool is aimed to do the following things:

* To predict a candidate’s chances of visa approval in terms of percentage.
* To display a message saying “Unable *to predict the chances because of invalid input value”*, if any candidate has an error value in any column.
* To suggest to the candidate the aspects he/she must improve to increase his chances of visa approval.
* To analyze the dataset and generate different statistics using HDFS and Hive.

**FEATURES**

The main features of this tool are:

* The tool can predict a student’s visa approval chances based on his/her profile based on Machine

Learning model.

* To perform data analysis on the student data using the Hadoop ecosystem components like HDFS

, Hue, Hive and using Solr-Lucene, Cassandra too.

* The tool should be able to suggest to a student where he/she should improve to increase his/her chances of visa approval.
* The tool should throw an error message if a candidate has incomplete information.

**Related Work (Background):**

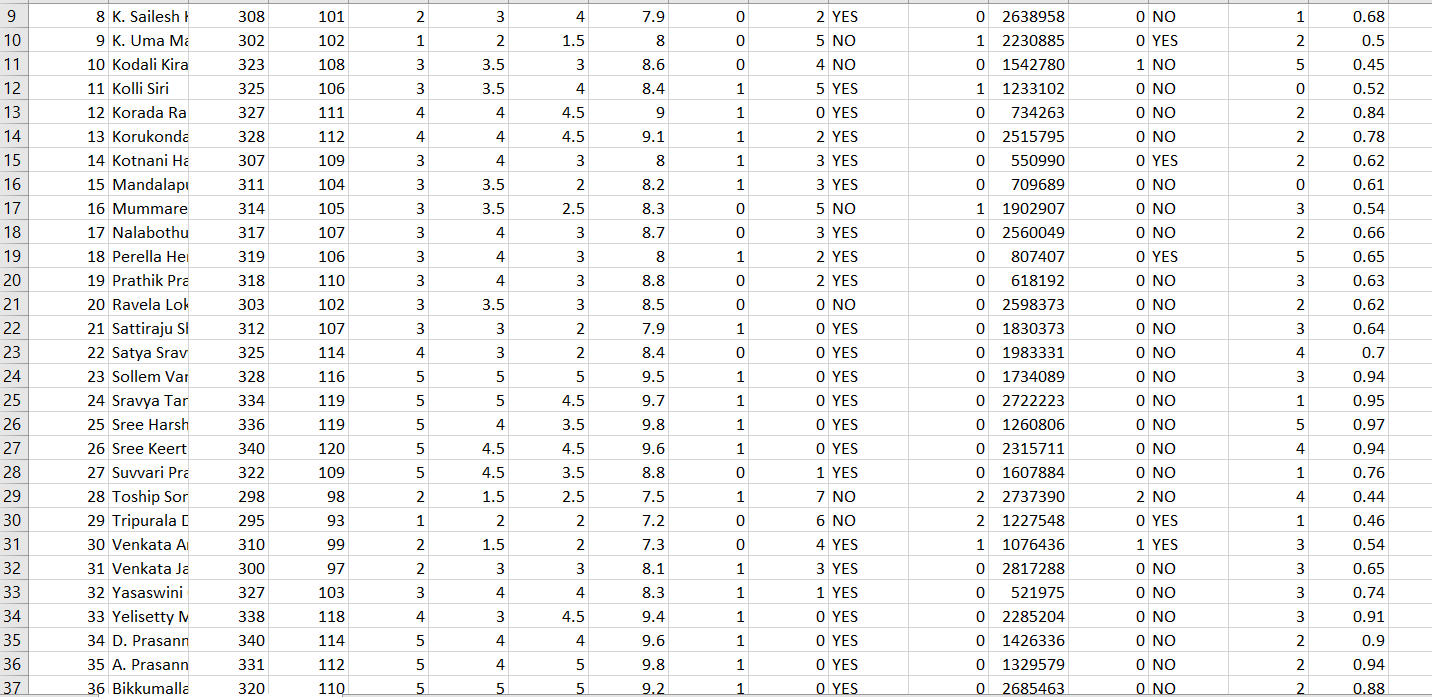
Many students approach consultancies for profile evaluation and spend a lot of money to identify drawbacks in their profiles. So, to address this problem we started researching about different possible ways to find a solution. Using Artificial Intelligence models and relevant datasets we can predict the chance of getting approved. But there isn’t any suitable dataset that is readily available online. So we started looking for the dataset that is closest to our desired dataset and we found a dataset on Kaggle that matches our requirement. The name of the data set is “Graduate Admission Prediction”. We took this dataset deleted irrelevant columns and added more columns that are required.

**References:** <https://www.kaggle.com/code/aryantiwari123/graduate-admission-prediction/data>

**Dataset:**

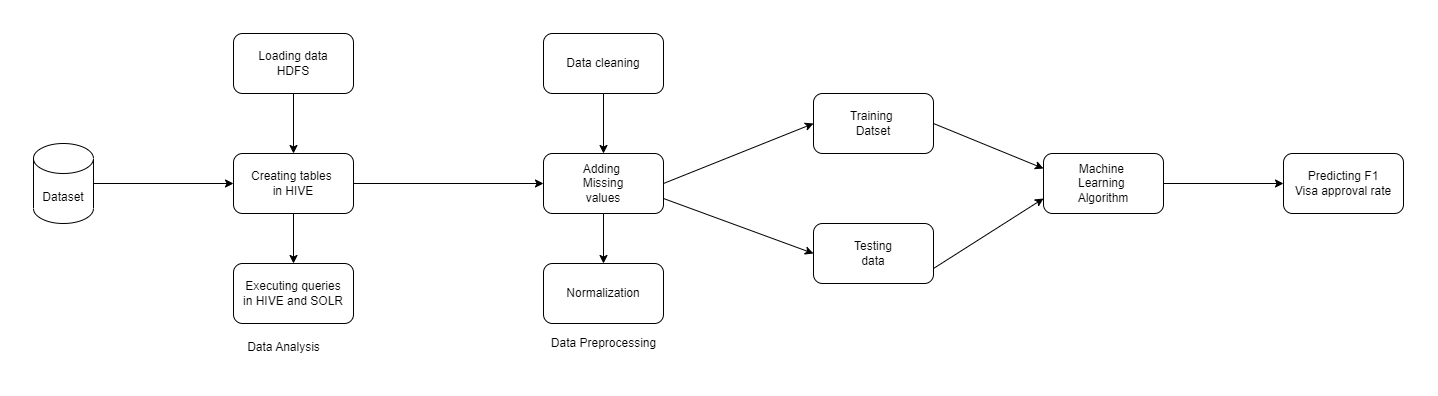
The dataset that we are working on has 17 columns of which 16 columns are regular columns that are used for prediction and one main column called Chance of visa approval is used to display the result. The dataset includes important columns like Gre and tofel scores, university ratings, CGPA, work experience, US relatives, no of times rejected earlier, and no of backlogs.

Our dataset has 400 records of which 300 records are used for training purposes and the rest 100 are used for testing purposes.



**Features And Implementation:**

**Architecture/Model:**



**Evaluation models:**

Firstly, we load the dataset into Hadoop Distributed File System (HDFS). From the data, we create tables in the hive by mentioning appropriate data types and Primary keys. Once the table is created, we can use queries to manipulate the data in these tables and can display results. For executing queries, we have used a Hadoop component called HIVE and SOLR which is a java-based search platform. With this, the data analysis part is completed and later we clean the data and pre-process the data to train ML models.

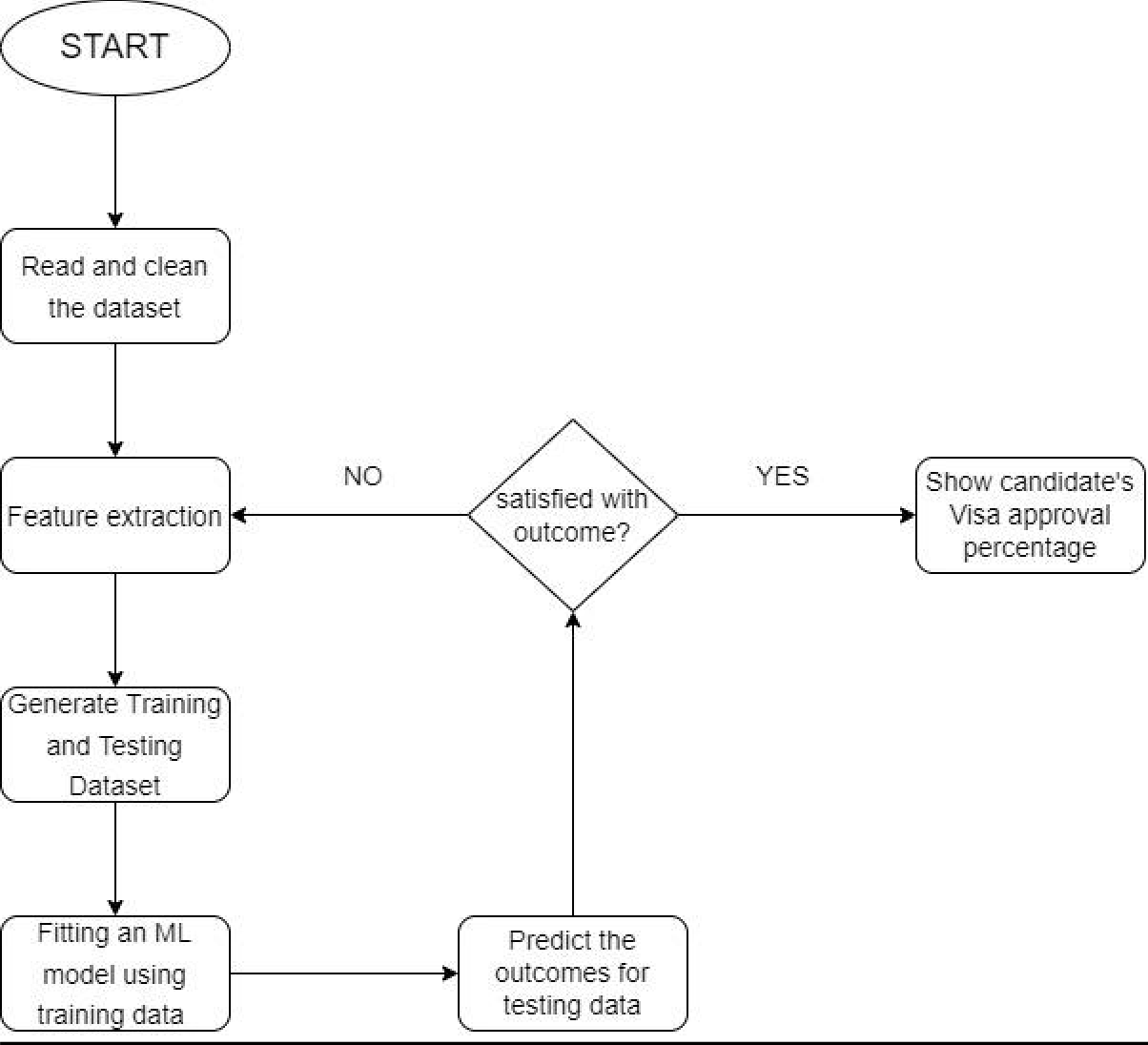
**Steps:**

1. The first step is to fetch an appropriate dataset with a decent number of records that matches the requirement.
2. After fetching the dataset, the next step is to analyze the dataset. In this project, we have used Hadoop ecosystem components for data analysis. In data analysis, we load data into HDFS, create tables in the hive, and use HIVE and SOLR for executing queries
3. Next step after data analysis is data preprocessing. In this step, we clean the data I.e., we eliminate null values and remove any unwanted data. We add any missing value to the columns in the tables.
4. After the above step the dataset is ready, now using the dataset we train ML models by using 75% of available data, and the remaining 25% is used for testing purposes.
5. Furthermore, once the ML models are trained, we use different Machine learning algorithms like K- Nearest neighbor and Random Forest and select the best algorithm which gives the highest accuracy.
6. The last step is to display the chance of F1 visa approval to the student based on the student’s profile and we’ll suggest in which areas the student can improve his profile to increase the chances of getting approved.

**Detail design of Features:**

* As mentioned earlier, to predict the F1 visa approval rate we take the student’s profile like student CGPA, GRE and TOFEL scores, work experience, no of backlogs, and no of times previously refused as inputs. We are using HiveQL language to write queries and manipulate data in datasets.
* To visualize the results after analyzing the data we are using excel to generate meaningful pie charts and bar graphs. In later increments, we will use data visualization python libraries like matplotlib to represent results.
* Using our custom dataset, we are training and testing Machine Learning models. These ML models analyze student’s profiles and identify the areas where a student is lacking. For instance, If the student has low GRE scores the application suggests reappearing for exams to get better scores.

**Workflow:**

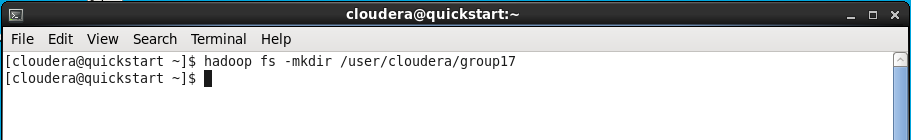


**Workflow description:**

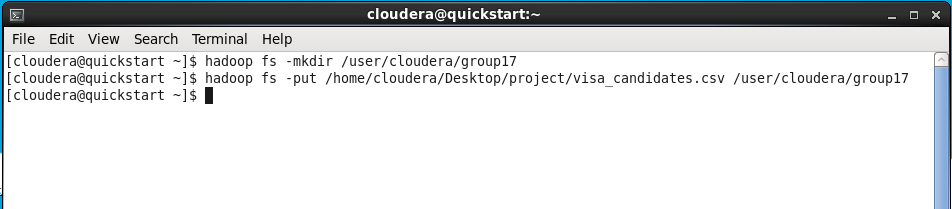
We start by collecting data, and after collecting data we read that read and clean using python during the cleaning phase we look for null values and eliminate those values from the data set. Once the data is cleaned and ready, we move to the feature extraction stage. In this phase, we transform some raw data into numerical values. The data thus available is split into two parts, One part is used for training purposes and another part is used for testing the ML models. After that using machine learning algorithms, we predict the outcomes of testing data. If the results are satisfactory then we proceed to the next step which is displaying the chances of getting a visa approved. If the results are not up to the mark, then we move back to the feature extraction step and reiterate the whole process until a satisfactory outcome is produced.

**Implementation:**

We Installed Virtual Machine on our host system, opened Cloudera in Virtual System, and worked on the Cloudera platform. Open a terminal in Cloudera.



Now we have created a new directory named “group17” by using the “mkdir” function.

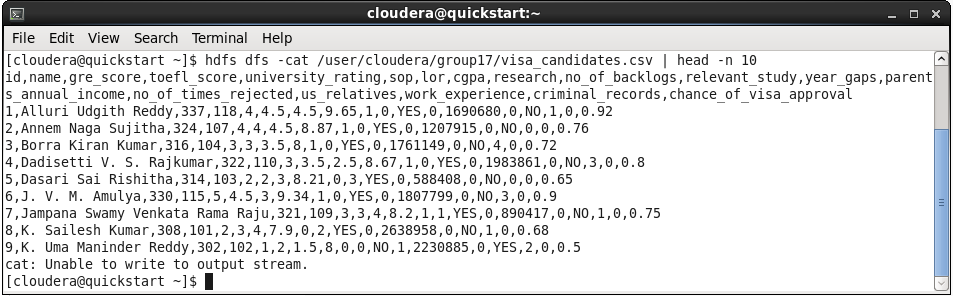


We have our source file “visa\_candidates.csv” on the Desktop. Now we are moving our source file from the desktop to the directory “group17” which we have created before.

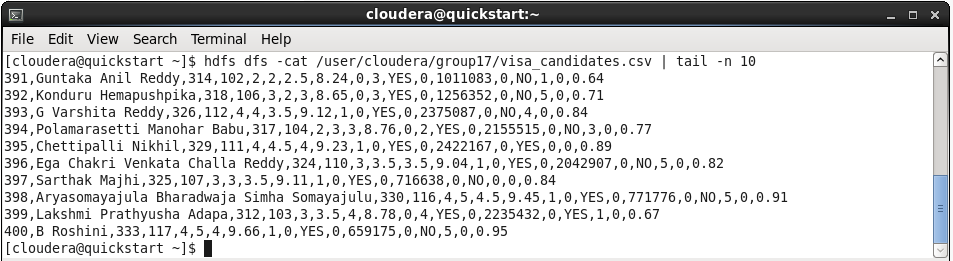
Graphical user interface, text, email

Description automatically generated

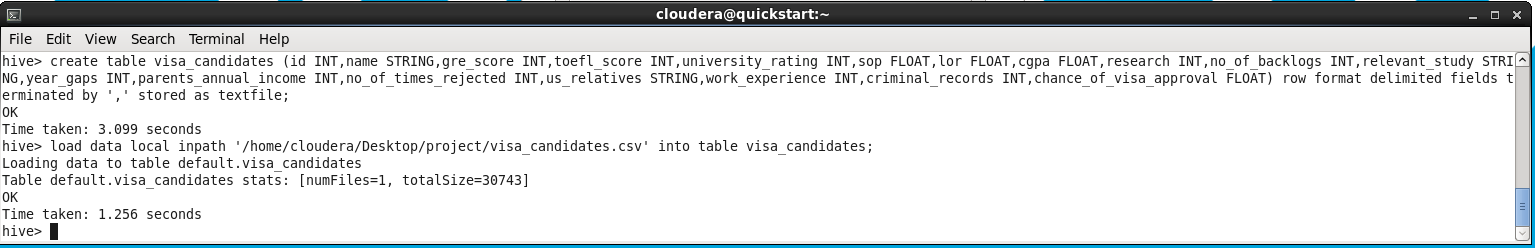
Visualizing the data in “Hue” to check whether the Dataset loaded into HDFS successfully or not.



We have used the “cat” command and “head” command to display the first ten lines of the Data from our source file “visa\_candidates.csv”

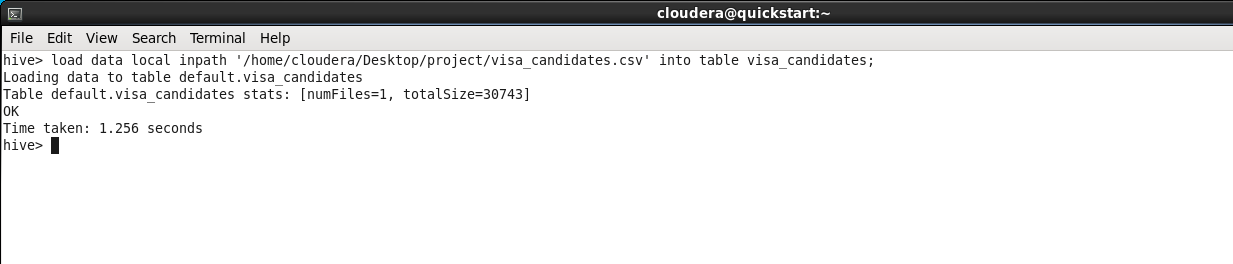


Now we are visualizing the last ten lines from our source file by using the above “cat” and “tail” commands.



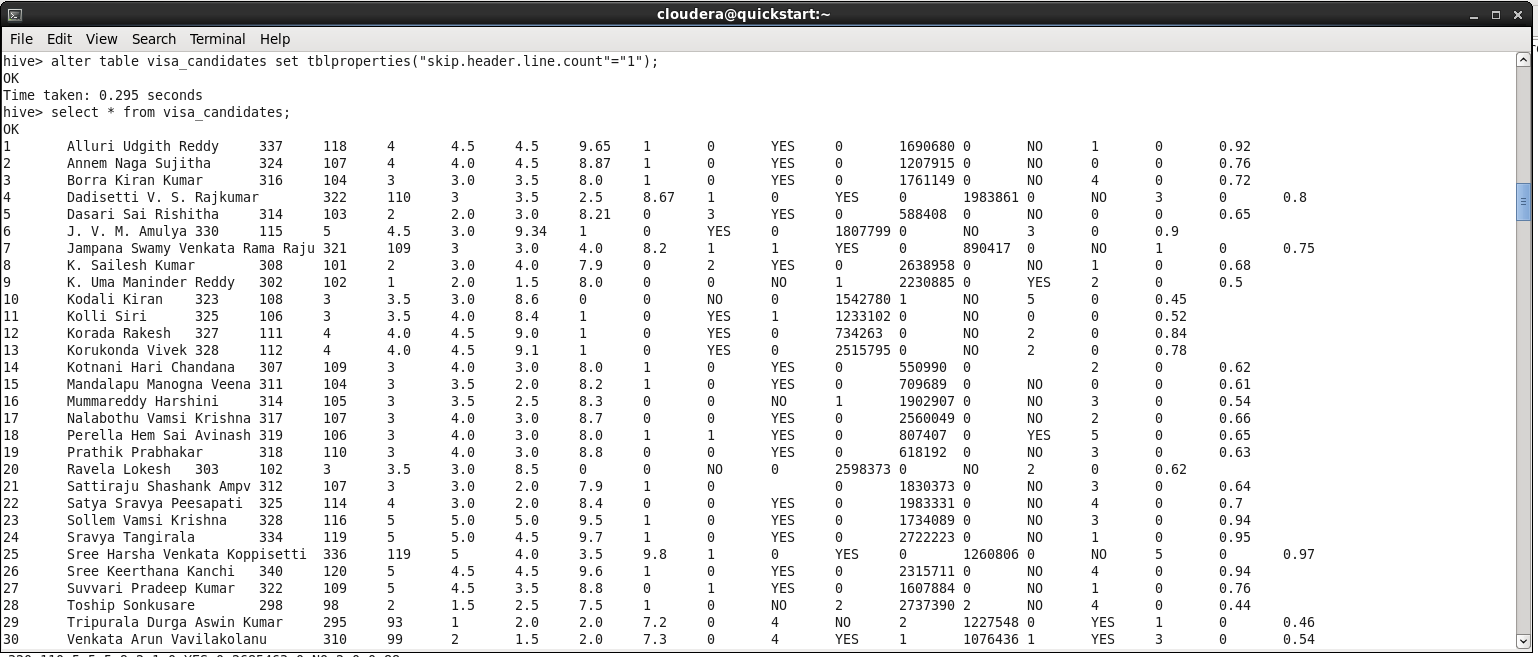
We Create the tables in HIVE and load the entire dataset.

To load the Data into the table which we have created. HIVE provides us with the facility to load datasets from files that hold on HDFS.



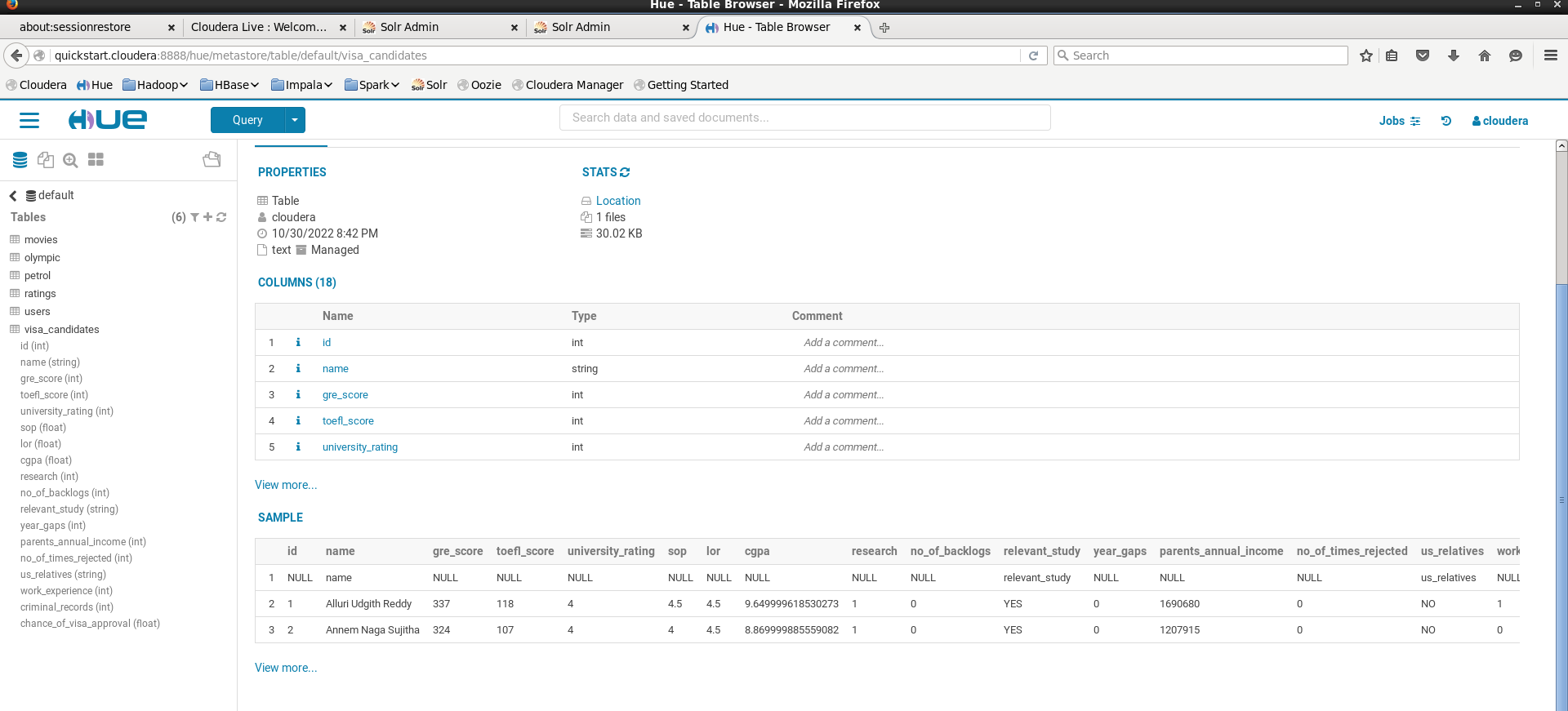
The Command used to load the data is:” LOAD DATALOCAL INPATH WRITE INTO TABLE”;

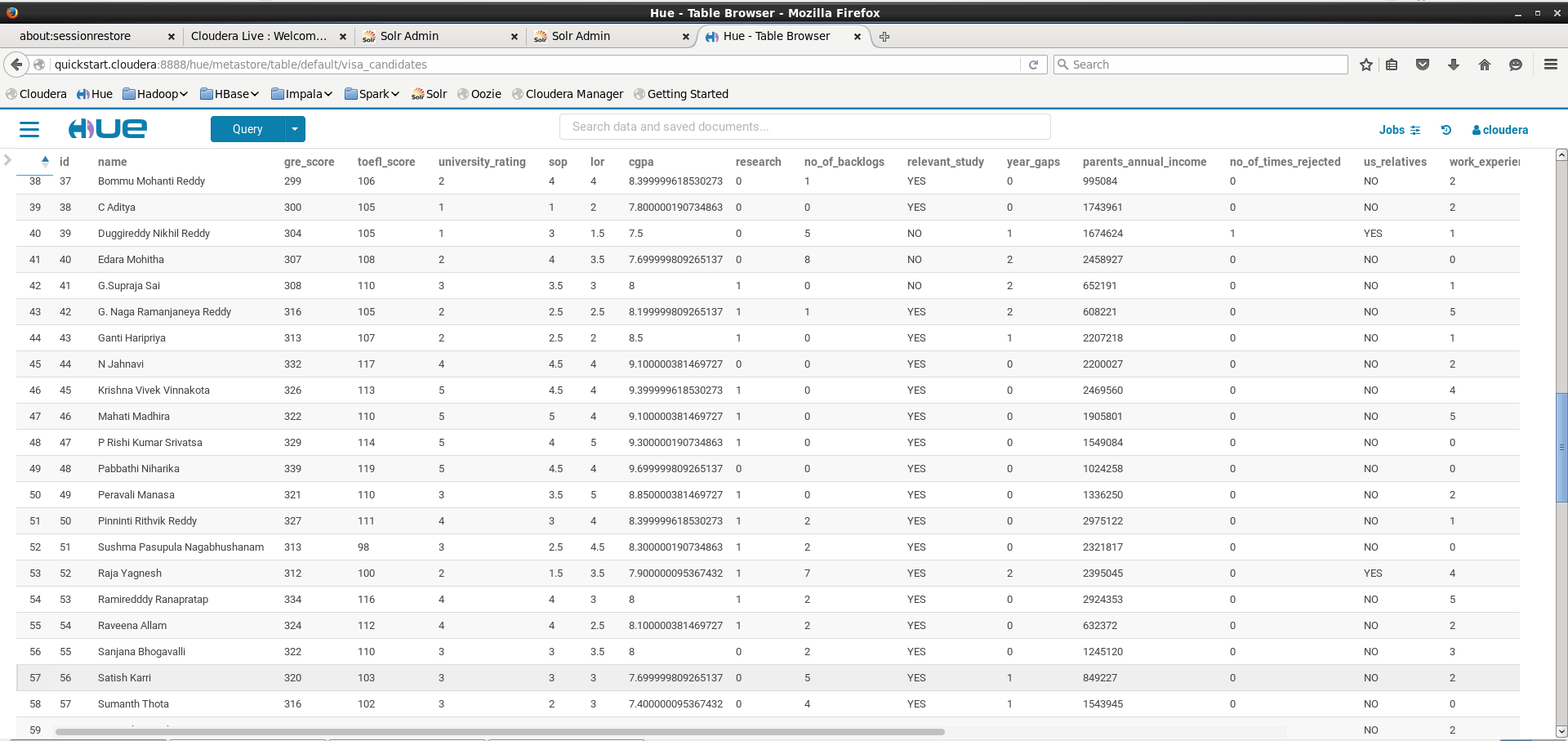
Display data in Command Prompt in HIVE



Using “select” command we displayed the data that is loaded into the table by this we can confirm that the data is successfully loaded into the table

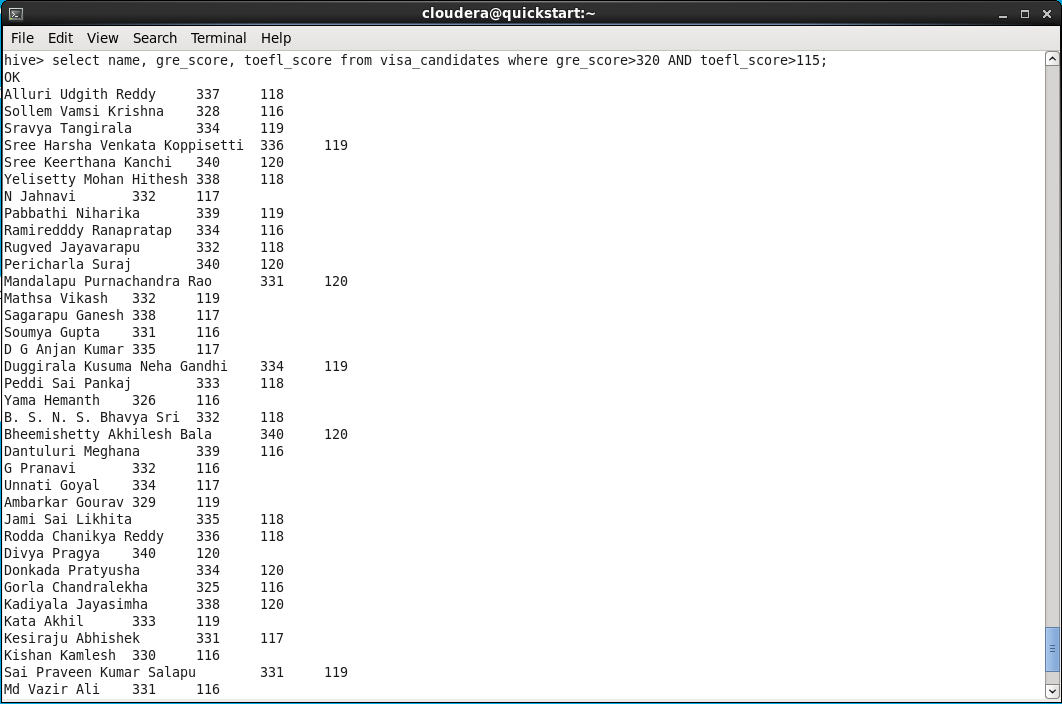
Visualize un Hue





Displaying the loaded dataset in HUE browser by which we can check the data that is loaded into the table we created is

**Finding Names and scores whose gre>320 and toefl>115**

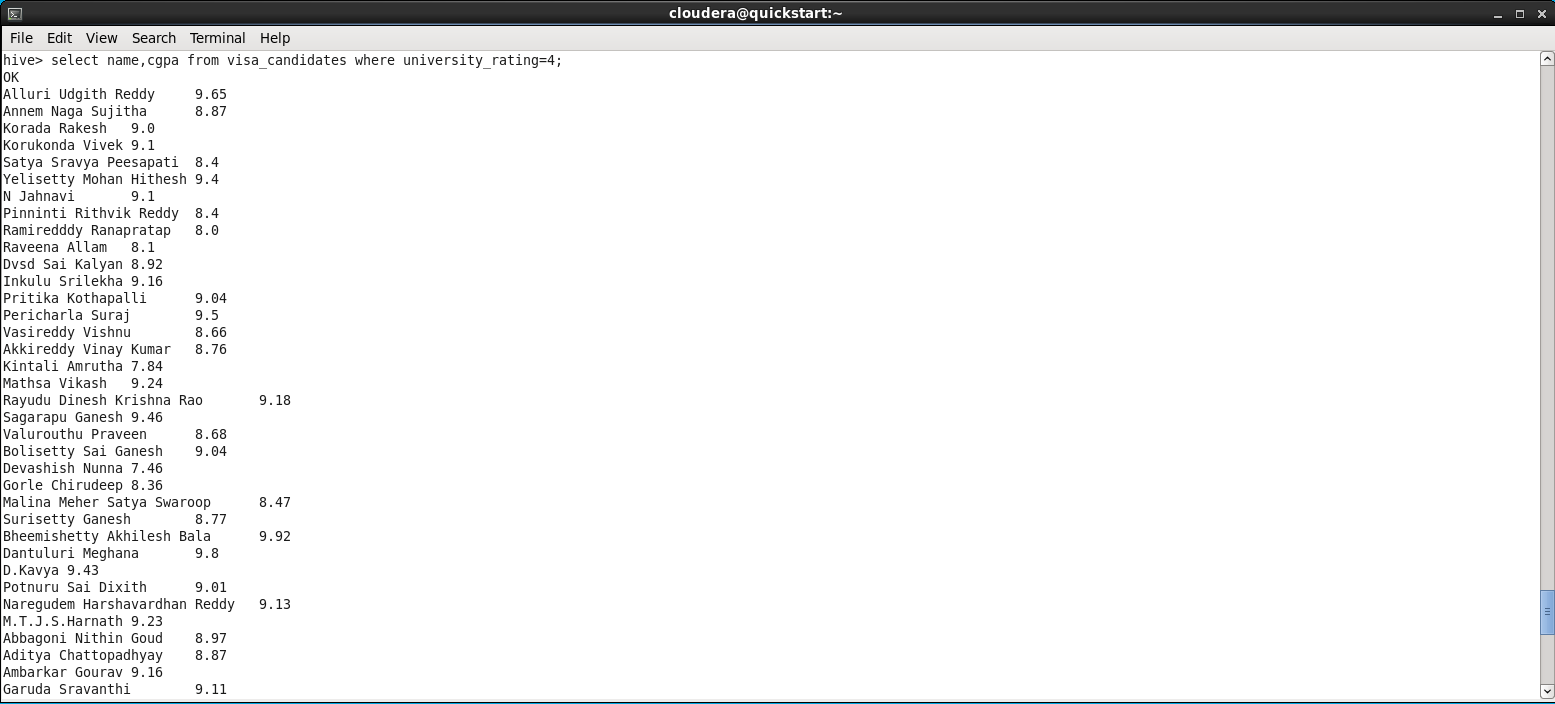


We are listing out the people who secured “gre” score above 320 and “toefl” score above 115 by using the above queries.

The main purpose of executing this query to list the people with high GRE and TOEFL scores so they have the most

visa acceptable.

**Finding names and cgpa for top universities**

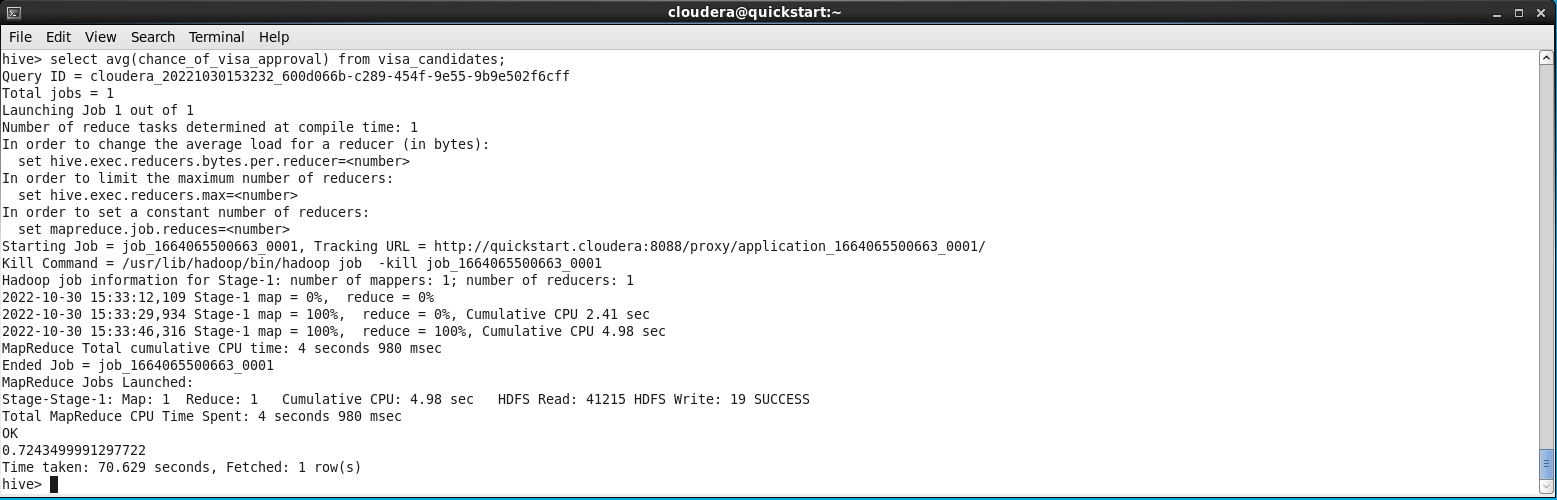


We are finding the people who are form top rated universities by using the above queries. In querie we analyze the dataset and provided the cgpa and name of students from top rated

universities.

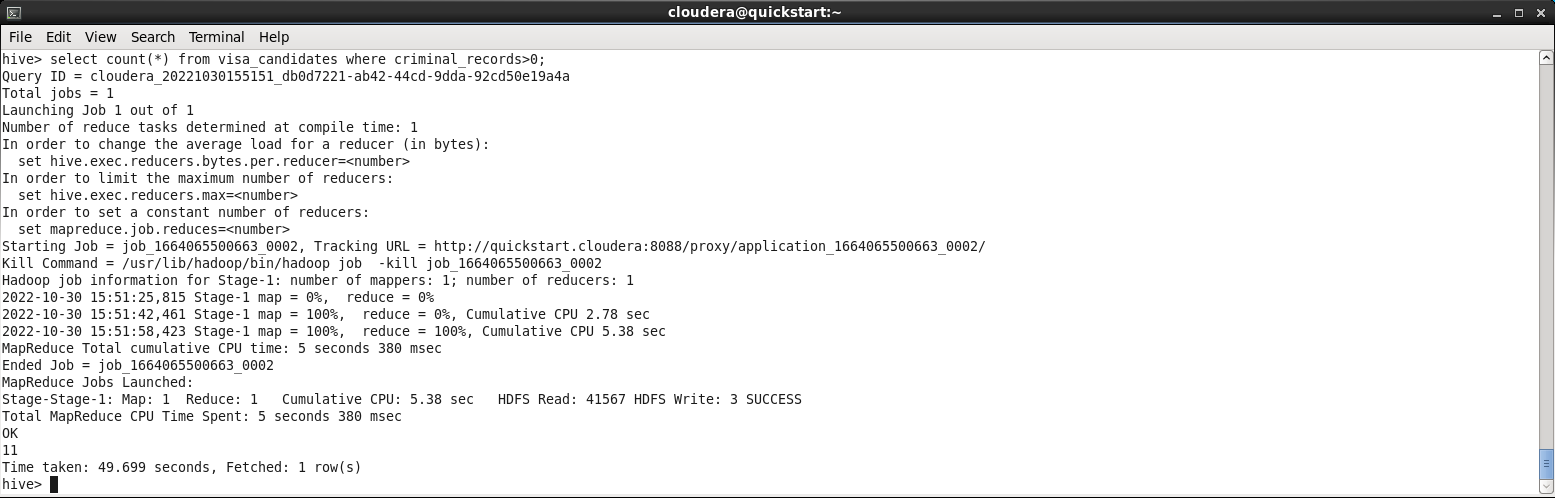
CGPA of the students also comes into picture when considering for the visa approval so we have taken this query for checking the candidates.

**Avg of visa approval**



We are finding the average visa approval rate by using above querie. This quire calls the mapper, reducer because the task executed through MapReduce process.

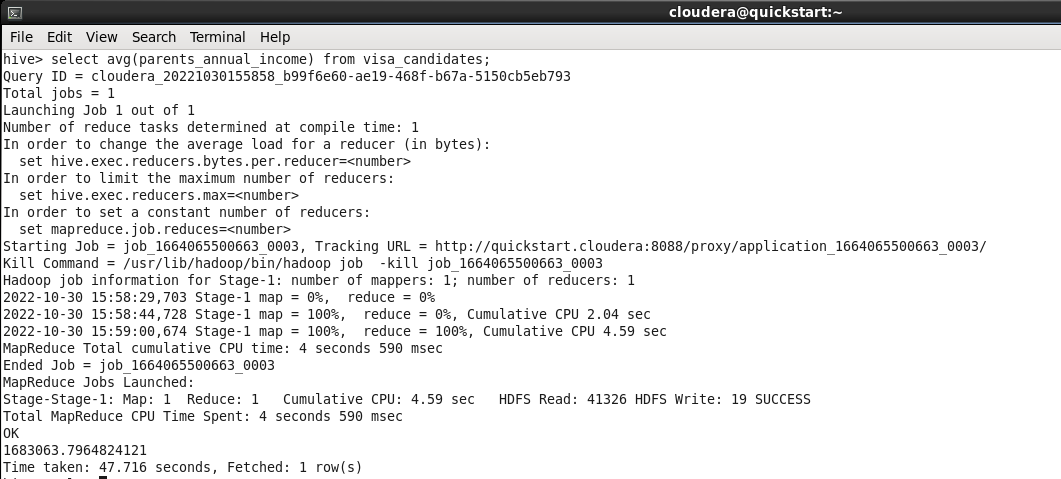
**Finding people with more criminal records**



In order for easy verification and approval of visa for any person their criminal record is checked all the time. Now here we are finding the number of people who have criminal record on their profile. For this task the above querie calls the MapReduce process.

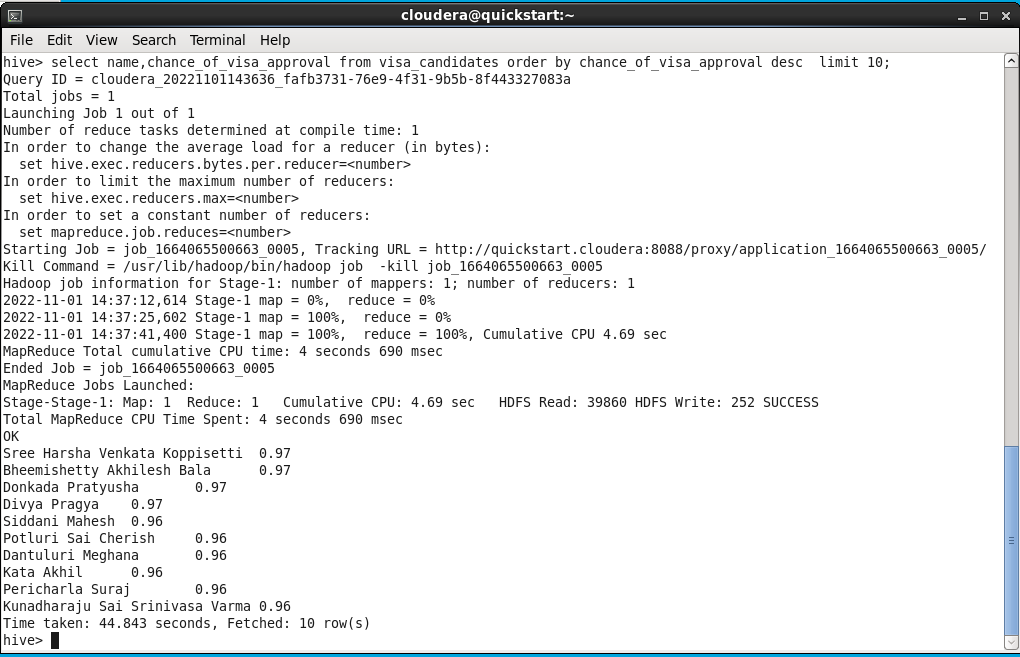
People with more criminal records have the least visa approval rate so, A condition is checked that people with more criminal records the least chance of visa approval.

**Avg of parent’s income**



For some particular visa’s like f1and j1, we need check the income in their bank account on in their parent’s bank account as most common of all these are “F1” visas which student visa so some has to sponsor them for their education. So, we are Finding the average income of parents of people who applied for visa. the above querie uses the MapReduce process to analyze the large data and provide accurate output.

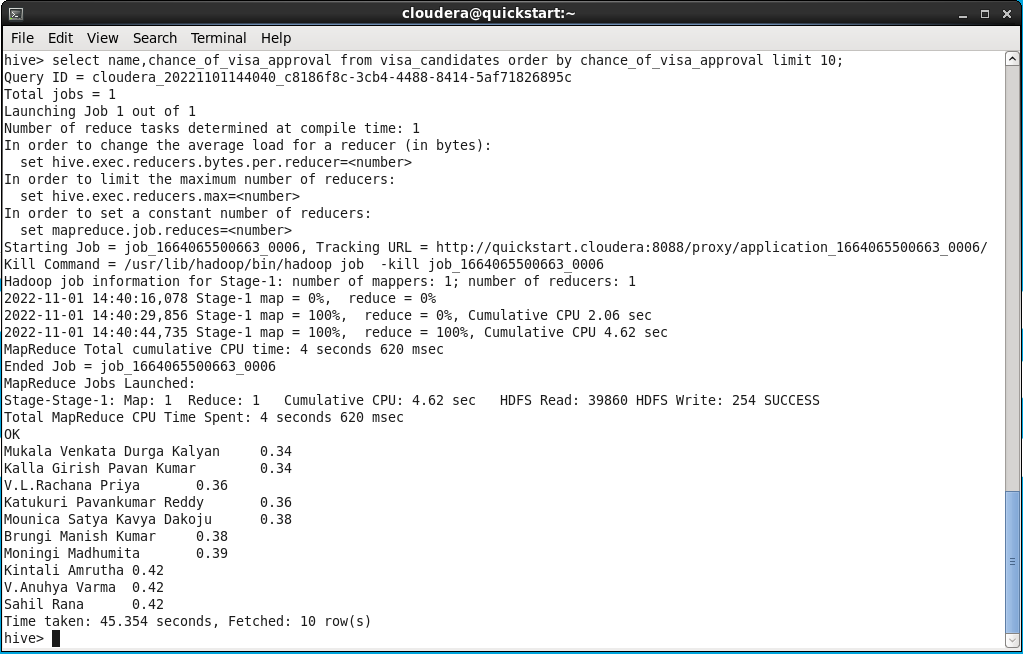
**Finding the top ten people visa approval rate**



Based on the data we find the top people with visa approval rate for this task we used above querie which calls the MapReduce which retries the accurate data

To display the top 10 students with high visa approval rate so that they can get an idea about who are the highest visa approved students.

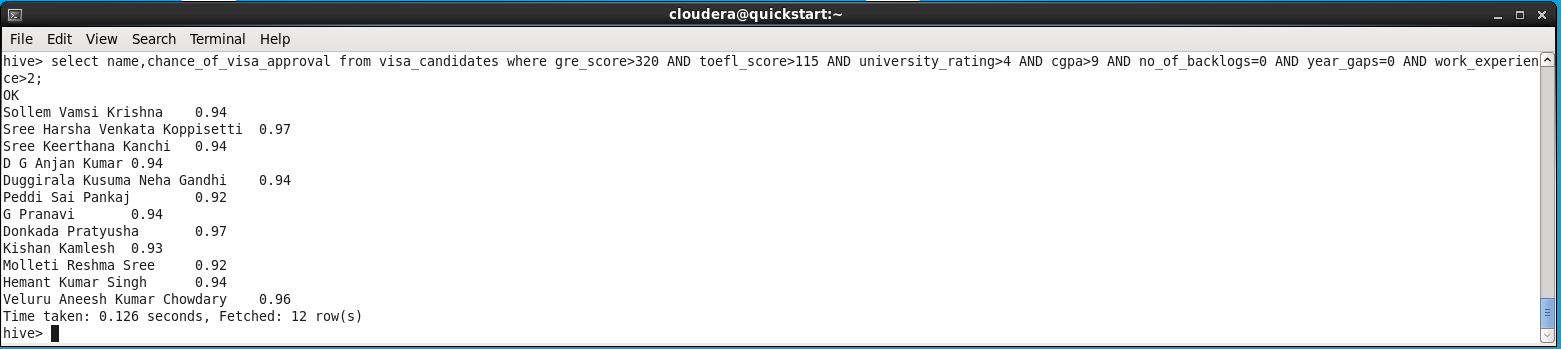
**Finding the last ten people visa approval rate**



Based on the data we find the leaste people with visa approval rate for this task we used above querie which calls the MapReduce which retries the accurate data

To display the least 10 students with high visa approval rate so that they can get an idea about who are the least visa approved students.

Final query



Here we are displaying the data of people whose profile is perfectly suited for visa approval based on their GRE, TOEFL scores along with the rating of their university and CGPS secured.

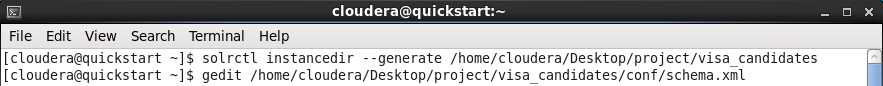
SOLR

Now we are performing several tasks on the visa\_candidates dataset using the SOLR tool



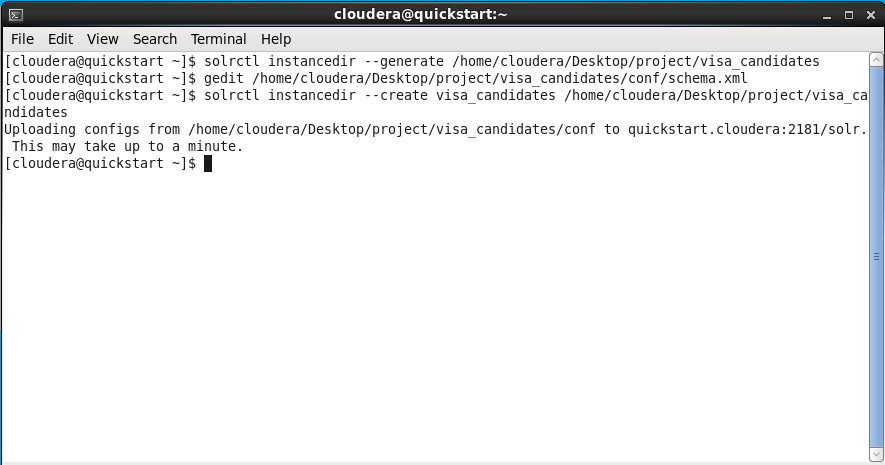
The above command is used for creating a new instance for the “visa\_candidates” and you can see a folder is created on the name “visa\_candidates” in which all the schema and other configurations are stored.

Changing Schema

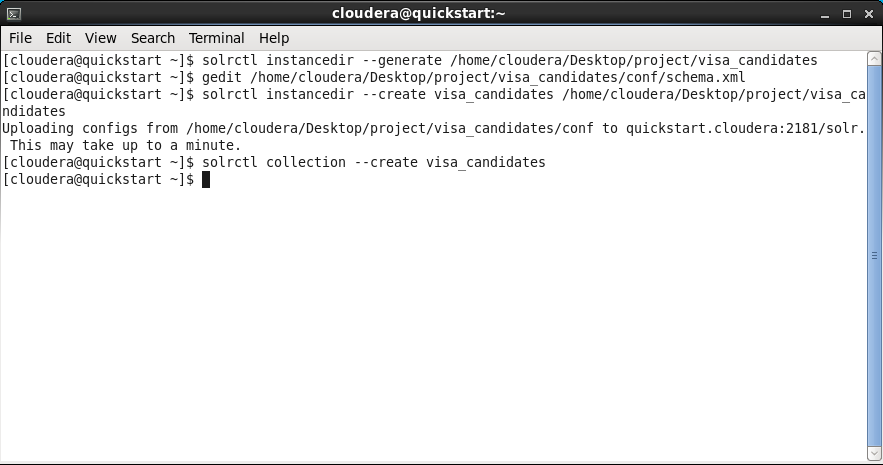


We need to edit the schema because all are not present in the old schema to add new features we are changing the schema using gedit.

We will change the schema in the schema.xml

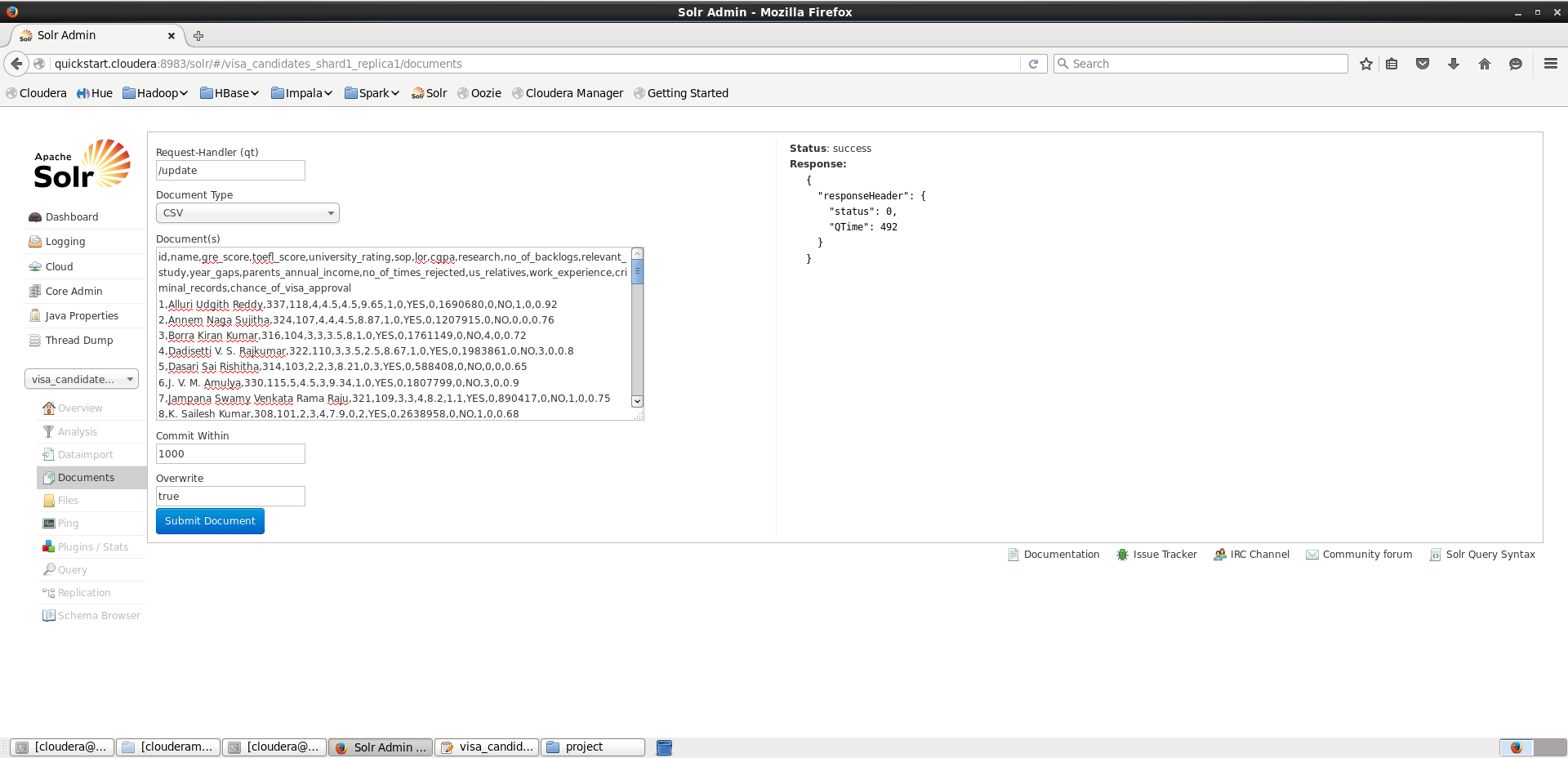


Here in the --create command the contents of the instance directory are uploaded to the zookeeper.



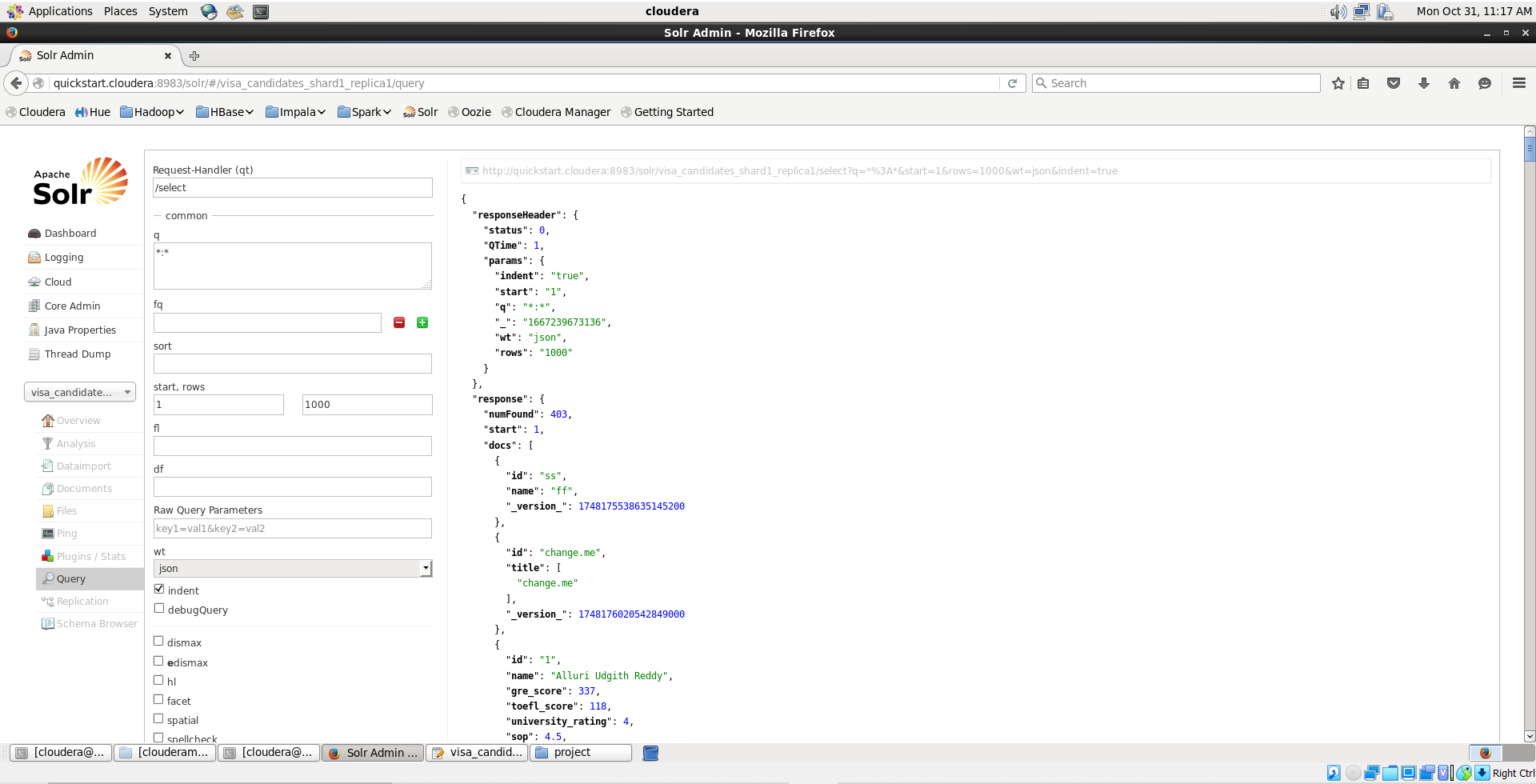
Here the “--create visa\_candidates” command a new collection is created in the solr which is named visa\_candidates

Loading Data into SOLR



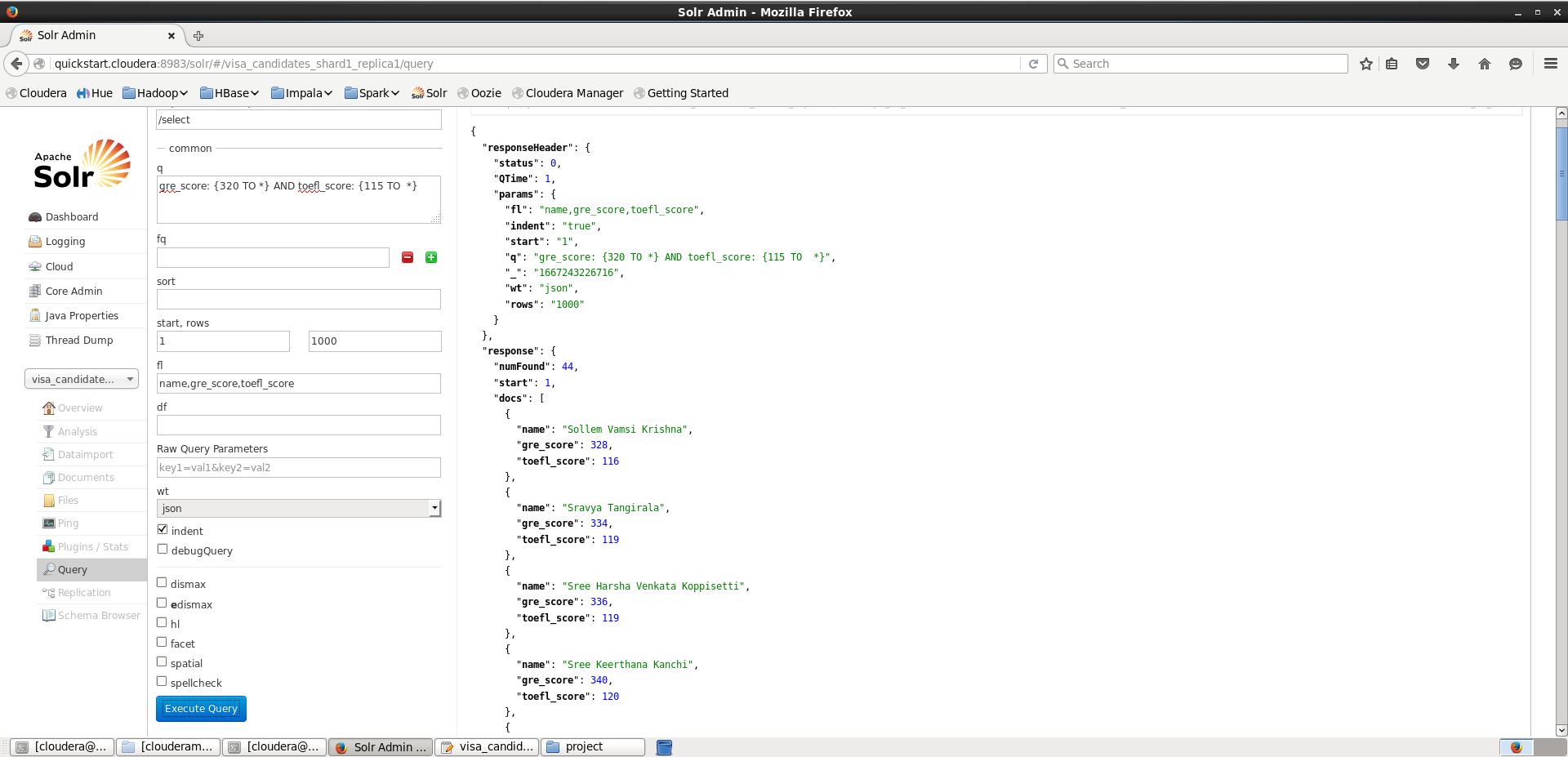
For uploading the data into solr we follow these steps, Open solr and go to Core-Admin there you can find the visa\_candidates. Now select the visa\_candidates from the selector and make sure the document type is ‘CSV’ then upload all the data from the visa\_candidates dataset into the document section of solr and then click submit Document button for submitting.

Displaying the data



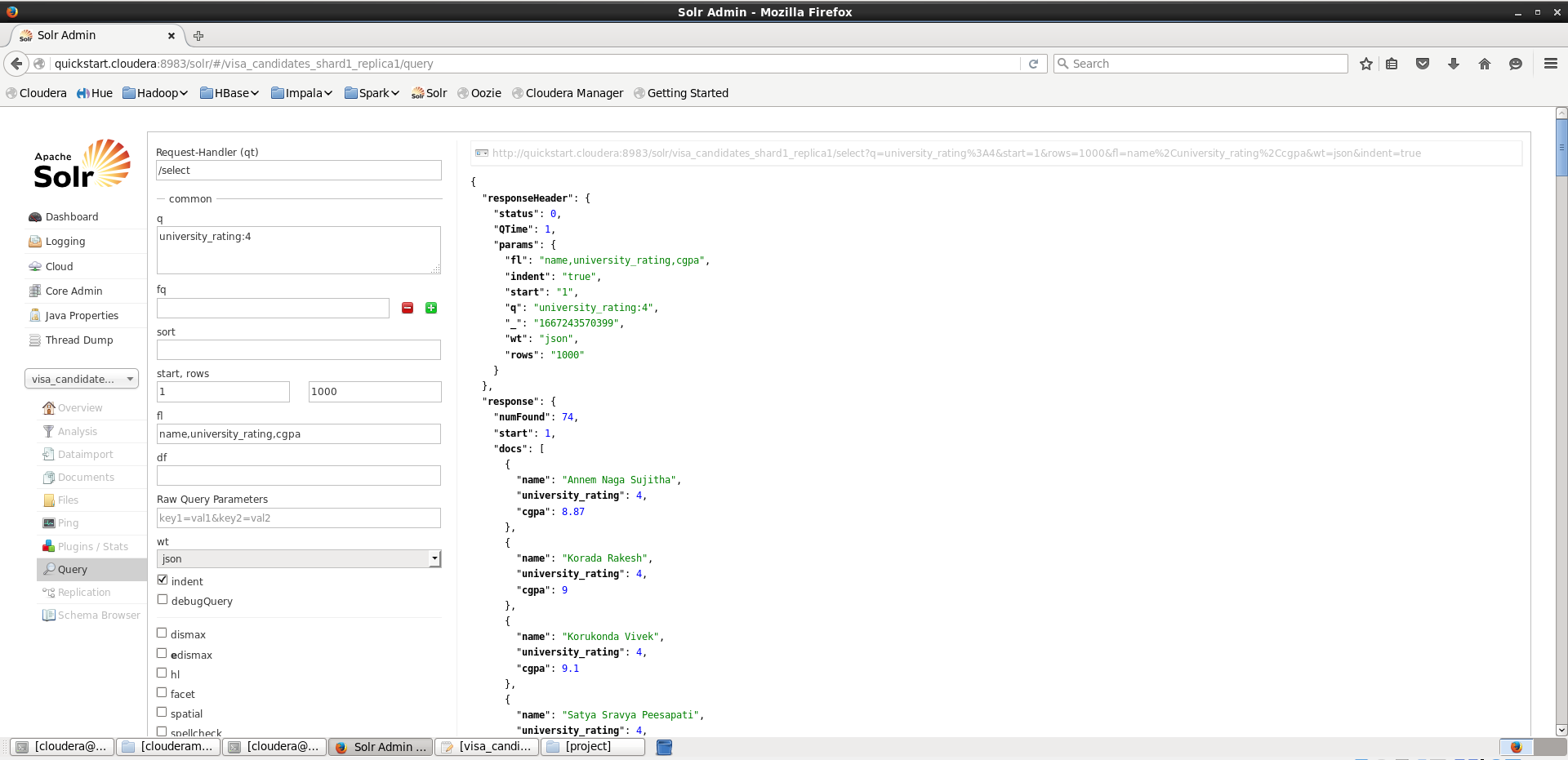
Here we are listing all the data of visa-applied candidates, we used the above queries “\*;\*” we should add the id, name, gre\_score, .. etc. then the data will be retrieved.The abovee im is the output.

Finding ‘gre’ and ‘toefl’ scores



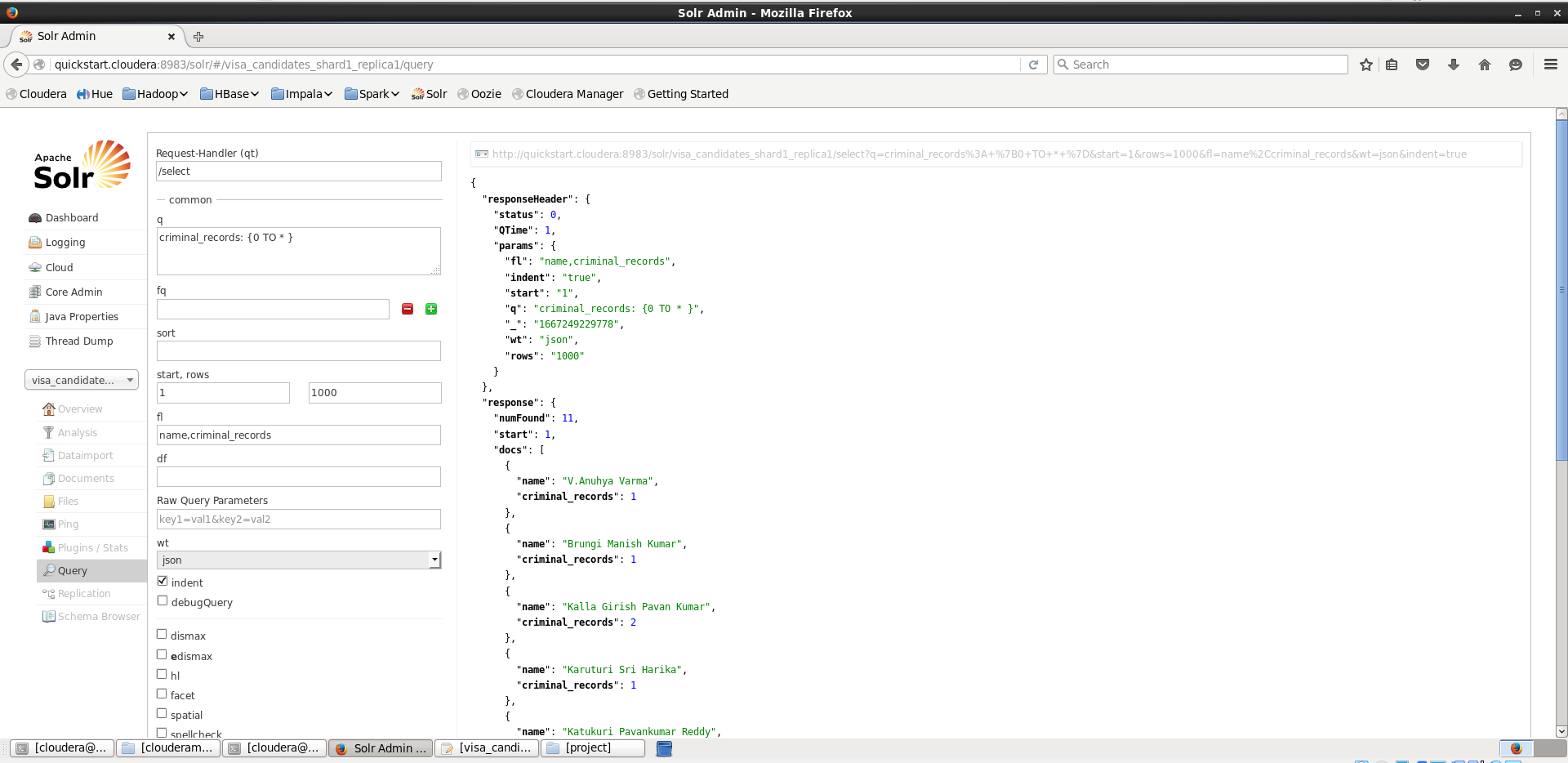
Here we are displaying the score and names of people who secured marks above 320 in ‘GRE’ and above 115 in ‘TOEFL’. To display this we used gre\_scores and toefl\_scores and combined both relations with the ‘AND’ Operator.

Finding University Rating



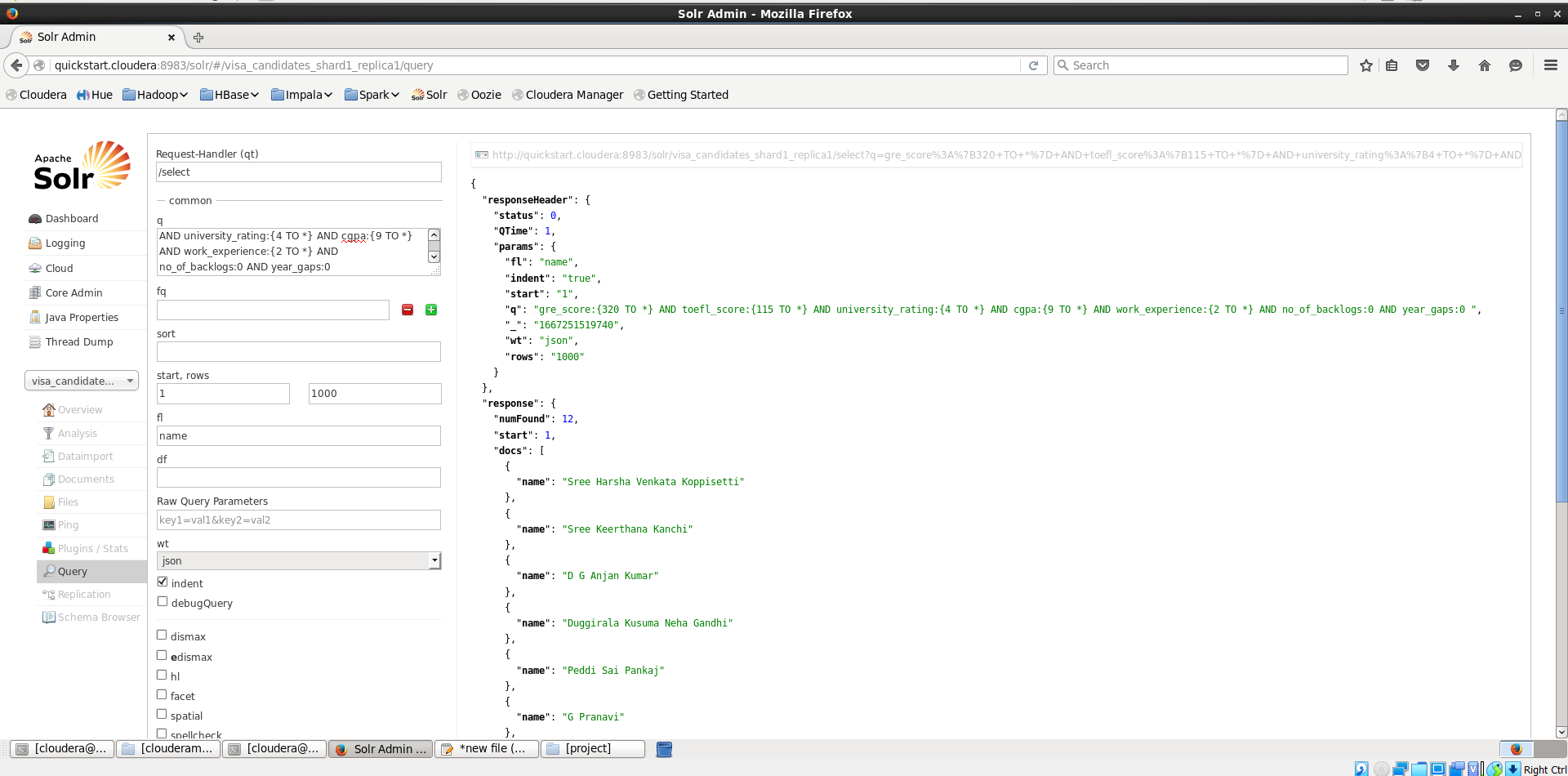
To display the university rating we should use the “university\_rating:4” query so the universities whose rating is 4 will be retrieved above image is the output.

Total Criminal Records



To display people who have criminal records we must use the “criminal\_records:{0 to \*}” query , in which the values inside flower brackets are from min number to maximum number , here ‘\*’ means topmost number in the dataset and this will provide the output with the people name and their criminal records count .

Final Query



Here we are finding the names of people with best possible profile based on the uploaded data by the above command, from which we are sorting out the people whohaves university ratingof 4 and above ,CGPA of 9 and above, work\_experience of 2 plus years with zero backlogs and no year\_gap.

**Cassandra**

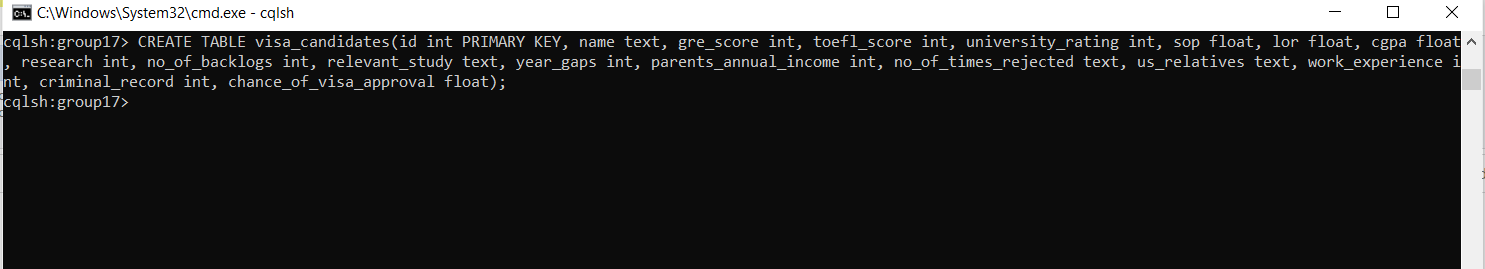
Opening of Cassandra

Text

Description automatically generated

We created a keyspace with the name group17 and the replication is done shown as above

Creating of table



Now we created a table with name visa\_candidates and described its attributes are shown as above.

Inserting values into the table

Text

Description automatically generated

Now after creating the table we have loaded the data into the table using the local CSV file which is present in our system, using the appropriate command.

Checking whether the data is loaded or not

A computer screen capture

Description automatically generated with low confidence

Using select \* we can see all the data in the table so that we can check whether all data is imported or not into the

table.

Finding the highest approval chances based on some conditions

Text

Description automatically generated

For this, we have taken some considerations such as gre\_score, toefl\_score, chance\_of\_visa\_approval, cgpa

and their parents\_annual\_income.

Null value insertion

Shape

Description automatically generated

Here by checking the value at id =75 we found that the cgpa value is null.

Text

Description automatically generated

* Using the update and set command we have inserted the cgpa value for the id=75.
* Also, check whether the value is reflected or not by using the select command

**Data Pre-Processing:**

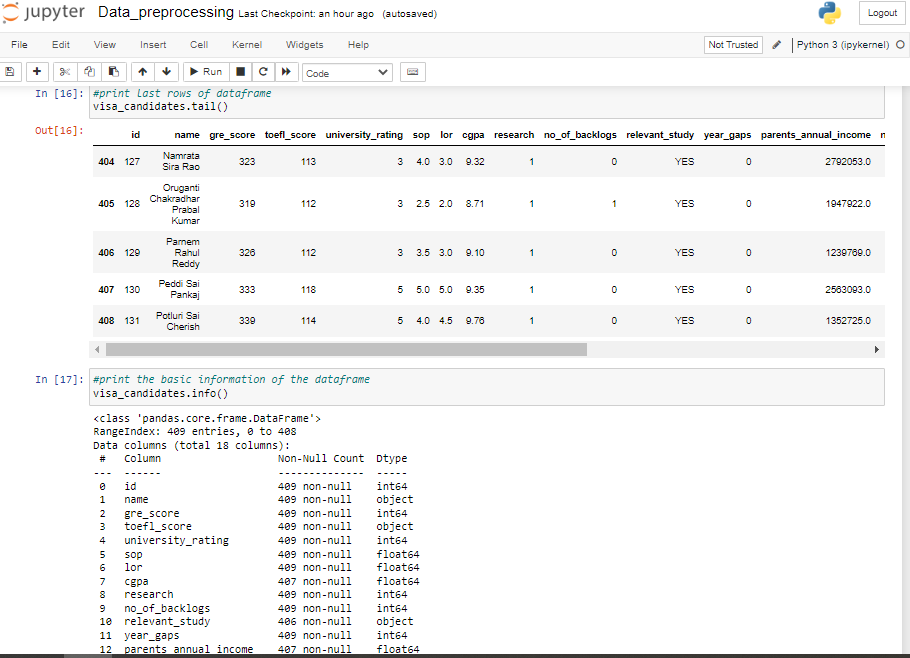
Text

Description automatically generated with medium confidence

Here we are trying to use libraries like pandas, NumPy, and matplot lib by importing them first. Then we

create a data frame by using the read\_csv(file path) method. Next, we check the data frame by viewing the head and tail of the data frame.

Dataframe.info() gives details of the dataset like the columns and number of entries and their datatypes as shown above.



Graphical user interface, application

Description automatically generated

Dataframe. describe() provides the basic statistics of the dataset like count, range, maximum, minimum, average/mean, median, and standard deviation as shown.

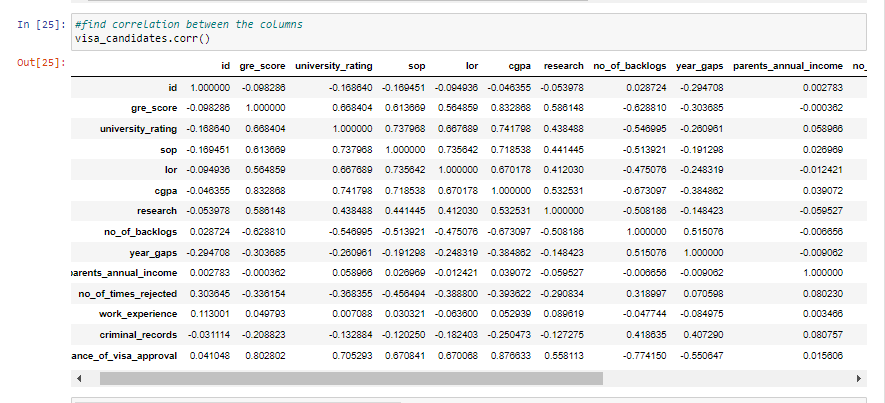
Next, we check if there are null values in our data using. ISNA().sum().

Graphical user interface, text, application

Description automatically generated

Here we try to drop these null values using drop() and check for null values again using ISNA().sum().Graphical user interface, application

Description automatically generated

Next, we check for duplicate rows and then remove these duplicates using drop\_duplicates() and check again for any duplicates.

Here we try to find out the correlation between the columns in the dataset, more specifically we are trying to get the relation between each column and visa\_chances prediction.

The next few figures are visual representations of these relations between a few important columns and visa\_approval\_chances like gre\_score, cgpa,no\_of\_backlogs, etc using a plot. scatter(x,y).Chart, scatter chart

Description automatically generated

* The students test scores and chance of visa approval are directly proportional to each other. If the students have good test scores then the probability of getting visa is increased greatly.

\*

Chart, scatter chart

Description automatically generated

* The higher CGPA will greatly improve the chance of visa approval.

Chart

Description automatically generated

* Similar to that of test scores and cgpa. The relation between university\_rating and chance of visa approval is directly proportional.

Chart, scatter chart

Description automatically generated

* The no of backlogs is inversely proportional to the chance of visa approval. If the no of backlogs are high then the visa approval chances are greatly reduced.

Chart

Description automatically generated

A picture containing table

Description automatically generated

As a final step in data preprocessing, we convert any string-indicated values to integers like yes/no values here and are mapped to 1/0 values for further processing.

**Analysis and Preliminary Results:**

* The above plot graph depicts the correlation between CGPA and the Chance of visa approval
* The parent’s annual income contributes only a partton the criteria but high annual income can bolster the students profile and can help in improving approval chances
* The chance of visa approval and criminal records are inversely proportional to each other. Any prior criminal records in the student’s profile will tremendously affect the visa approval rate.

**References / Bibliography:**

1. <https://www.kaggle.com/code/aryantiwari123/graduate-admission-prediction>
2. <https://www.kaggle.com/datasets/saddamazyazy/go-to-college-dataset>
3. <https://www.jetir.org/papers/JETIR2204278.pdf>
4. <https://www.kaggle.com/code/akhilkasare/h-1b-visa-prediction-using-machine-learning>
5. <https://www.kaggle.com/code/campusx/gre-admission-prediction/data>
6. <https://ieeexplore.ieee.org/document/8933628>

**Project Management:**

* **Work Completed:**

Description –

In this project increment 1 , we aimed at finishing the data analysis part of the project which we finished . We used HDFS to store our dataset in Hadoop by making a directory and putting our dataset into it. Then we used hive for easy query execution instead of implementing direct map-reduce code. We created hive tables and ran different queries analyzing the data. Then we implemented the same queries in Solr-Lucene and Cassandra too.

Responsibility –

Suhas Siddarajgari Tellatakula - All Data analysis features and report

Sai Tejesh Gonemadatala - All Data analysis features and report

Sai Rohith Varma Kantem - All Data analysis features and report

Sai Praneeth Reddy Avula - All Data analysis features and report

Contributions –

Suhas Siddarajgari Tellatakula – 25% in implementing Data analysis features and report

Sai Tejesh Gonemadatala - 25% in implementing Data analysis features and report

Sai Rohith Varma Kantem - 25% in implementing Data analysis features and report

Sai Praneeth Reddy Avula - 25% in implementing Data analysis features and report

* **Work to be completed:**

**Description** –

In the next increment , We are supposed to use different Machine Learning algorithms and the cleaned data set and analyse the results from algorithms . Once we compare the results , we pick the best algorithm giving us best accuracy.

**Responsibility** –

Suhas Siddarajgari Tellatakula - All Machine learning features and report

Sai Tejesh Gonemadatala - All Machine learning features and report

Sai Rohith Varma Kantem - All Machine learning features and report

Sai Praneeth Reddy Avula - All Machine learning features and report

**Expected Contributions** –

Suhas Siddarajgari Tellatakula – 25% in implementing Machine learning features and report

Sai Tejesh Gonemadatala - 25% in implementing Machine learning features and report

Sai Rohith Varma Kantem - 25% in implementing Machine learning features and report

Sai Praneeth Reddy Avula - 25% in implementing Machine learning features and report