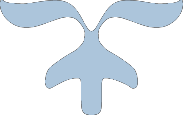


CSC 570 BIG DATA

ANALYTICS

A Project on Recommender System & Rank



May 7, 2017

RAJESH JAISWAL AND RAJAT HATWAR



Contents

INTRODUCTION: .................................................................................................................. 2

PROBLEM DEFINITION AND PROJECT GOALS:...................................................................

DATA ANALYSIS: ....................................................................................................................

EXPERIMENTAL RESULTS**:** ...................................................................................................

CONCLUSION: .................................................................................................................... 12

REFERENCES: ..................................................................................................................... 13

i



**Acknowledgement**

Working over this project on “Recommender System and Rank" was a source of immense knowledge to us. We would like to express our sincere gratitude to Dr. Elham Khorasani (Buxton) for always support and guidance throughout the course work. We really appreciate your support and are thankful for your

cooperation.



1. Introduction:

The College Scorecard project is designed to increase transparency, putting the power in the hands of students and families to compare how well individual postsecondary institutions are preparing their students to be successful. This project provides data to help students and families compare college costs and outcomes as they weigh the tradeoffs of different colleges, accounting for their own needs and educational goals.

These data are provided through federal reporting from institutions, data on federal financial aid, and tax information. These data provide insights into the performance of institutions that receive federal financial aid dollars, and the outcomes of the students of those institutions. A complete set of these data for all active and data are available on Scorecard data website, a subset of these data elements are displayed on the consumer-facing College Scorecard website.

* ALTERNATING LEAST SQUARES

ALS recommender is a matrix factorization algorithm that uses Alternating Least Squares with Weighted-Lamda-Regularization (ALS-WR). It factors the user to item matrix A into the user-to-feature matrix U and the item-to-feature matrix M: It runs the ALS algorithm in a parallel fashion.  The ALS algorithm should uncover the latent factors that explain the observed user to item ratings and tries to find optimal factor weights to minimize the least squares between predicted and actual ratings.

* RANK FUNCTION

RANK calculates the rank of a value in a group of values. The return type is NUMBER. Rows with equal values for the ranking criteria receive the same rank. Oracle Database then adds the number of tied rows to the tied rank to calculate the next rank. Therefore, the ranks may not be consecutive numbers. This function is useful for Top-N and Bottom-N reporting.

* As an aggregate function, RANK calculates the rank of a hypothetical row identified by the arguments of the function with respect to a given sort specification. The arguments of the function must all evaluate to constant expressions within each aggregate group, because they identify a single row within each group. The constant argument expressions and the expressions in the ORDER BY clause of the aggregate match by position. Therefore, the number of arguments must be the same and their types must be compatible.
* As an analytic function, RANK computes the rank of each row returned from a query with respect to the other rows returned by the query, based on the values of the value\_exprs in the order\_by\_clause.

2. PROBLEM DEFINITION AND PROJECT GOAL

* PROBLEM DEFINITION

The College Scorecard project is designed to increase transparency, putting the power in the hands of students and families to compare how well individual postsecondary institutions are preparing their students to be successful. This project provides data to help students and families compare college costs and outcomes as they weigh the tradeoffs of different colleges, accounting for their own needs and educational goals using students SAT and ACT score values.

* PROJECT GOAL

Provide the best suited college names and it’s all details for a student depending on his ACT and SAT score and region on USA.

* DATA SET DETAILS

Source: <https://collegescorecard.ed.gov/data/documentation/>

Size: 1.5 + Gigabyte

Variables: 1743

Duration: 10 years’ data

Format: Datasets available in comma separated format

This dataset holds the details about the college names, college degree availability and required scores which help new student to find college according to his/her ACT and SAT scores. Also provide cost of attendance and salary after once education complete

3. DATA ANALYSIS

* DATA LOADING
* Data Load
  + Load input file in RDD (Resilient distributed dataset) with Hadoop & Spark configuration delimit values by: “,”
  + Split each and every input value and fetch required values in new line.
  + Filter out new line data with regular expression Generate combination of feature columns and target column
* Methods used in procedure
  + “Map” function to transform data.
  + “Split” function to split strings.
  + “flatMap” to generate combination of required data.
  + “fileText” to read data from file to load in RDDs

**Procedure:**

* We loaded the data in the RDD from input folder.
* After loading of data, we separated each data field by “,” and combination of those value make one a new line of our data RDD
* We then selected only the required data field from the all data field and created new RDD
* The Next step we created data frame from given RDD to work better way or to implement SQL queries on it

**Snapshots:**

* DATA CLEANING
* NULL Data Handling
  + Data have total 1743 variable values out of which we fetch only 68.
  + Out of those 68 we are using 10 + variables.
  + These 10 variable fields have many null data observations.
* Methods used in procedure
  + “Map” function to transform data.
  + “Split” function to split strings.
  + “FIlter” to generate combination of required data.

**Procedure**:

* Applied data cleaning on your required data set
* As data have multiple rows with “NULL” values which will hamper the accuracy of our algorithm.
* We removed data which have “NULL” values in each required variables.
* Then converted this data to Rating data format to apply data model on it make it ready.
* We applied filter on whole line of RDD using “NOT and Contains”

**Snapshots:**

* ALGORITHM TRAINING
* Data Preparation
  + Load cleaned data into the new RDD(Resilient distributed dataset) without having any noise data or dirty data.
  + Then splitted data into the training and test data set
  + Training dataset holds 80% data and test data hold 20 % of total data .
  + Trained model on train dataset.
* Accuracy model
  + To calculate the accuracy of the given data model we used below parameter
  + RMSE
  + Precision
  + Recall

**Procedure**:

* Trained data using ALS data predication model having all features column and target column into it with 80 % data.
* Using this data set predicted values of target column for the test dataset of size 20 % of total data.
* Then Calculated below parameter using actual test dataset and predicted dataset to check accuracy of our model
* RMSE
* Precision
* Recall
* RANK IMPLEMENTATION
* Data Creation
  + Created data frame having columns region, states and admission rate.
  + Created new data frame having columns states, college\_name and admission\_rate
  + Implemented rank function on above to data frame to suggest best fit college
  + Joined above data frame to use in final selection.
* Method Implemented
  + “Map” function to transform data.
  + “filter” function to generate graph.
  + “Spark SQL” and “Joins”
  + “Rank” function

**Procedure**:

* By using data frame functionality created two tables. Table “RSTable” with region, state and admission\_rate
* Table “SCTable” with columns state, college name and admission\_rate
* Both table was sorted using rank on admission\_rate value
* Then used select queries using region and state as a primary column of two tables combinations.
* While combining above table used state as primary columns and selected only those data where state values are equal
* Implemented these SQL queries using spark.sql

SNAp Shots

* FINAL RANK IMPLEMENTATION
* Converted predicted RDD in to data frame then in table “ PDTable” and by using select query combining this table with the resultant table of previous two tables.
* Considered region as a primary key for selection and condition.
* Then returned college names which have highest admission rate and highest rank
* Also provide other required data for particular AT and SAT score entries value

4. EXPERIMENTAL RESULTS:

**Steps performed to attended final goal:**

* **In first step we cleaned data from whole chunk of data using filter method.**
* **Then implemented an algorithm to predict values of the region column which is our target parameter to predict.**
* **Then created a data frame of predicted results and compared it with the actual datasets.**
* **Then to check its accuracy, implemented below parameters as given.**

**RMSE**

**Precision**

**Recall**

**Also create two tables in previous Rank function and combine the as per the need and selected only thos colleges for a particular marks having highest admission rate to get admission.**

**Sourcecode SnapShot:**

5. CONLUSION:

In this program got opportunity to learn how Data Algorithms work to predict or recommend in the real world. We also used the SQL rank function Thus, we Got over all idea of rank function, data frame, data cleaning and implementation of ml -libraries of websites in a search engine.

6. REFERENCES:

* + [**https://collegescorecard.ed.gov/data/documentation/**](https://collegescorecard.ed.gov/data/documentation/)
  + [**https://spark.apache.org/docs/2.1.0/ml-classification-regression.html#input-columns-1**](https://spark.apache.org/docs/2.1.0/ml-classification-regression.html#input-columns-1)
  + <http://spark.apache.org/docs/latest/api/scala/index.html#package>
  + <https://cwiki.apache.org/confluence/display/Hive/LanguageManual>
  + [http://spark.apache.org/docs](http://spark.apache.org/docs/latest/api/scala/index.html#package)