Predict the CLICK-THROUGH RATE OF ADS

Table of Contents:

1. Project Description
2. Dataset Description
3. Hypothesis
4. Workflow
5. Analysis
6. Results
7. Conclusion
8. Project Description
9. Dataset Description

The dataset was downloaded from <https://www.kddcup2012.org/c/kddcup2012-track2/data> which consists of following files with tab separated fields.

|  |  |  |
| --- | --- | --- |
| File | Size | Records |
| training.txt | 9.9GB | 149,639,105 |
| descriptionid\_tokensid.txt | 268MB | 3,171,830 |
| purchasedkeywordid\_tokensid.txt | 26MB | 1,249,785 |
| queryid\_tokensid.txt | 704MB | 26,243,606 |
| titleid\_tokensid.txt | 171MB | 4,051,441 |
| userid\_profile.txt | 283MB | 23,669,283 |

Description of **training.txt** dataset:

It consists of session log messages from a search engine named ‘soso.com’. Each session is a like an interaction between a user and search engine. Each session is a group of instances which is aggregated by certain fields like userId, AdId and query to reduce the size of dataset.

Each line of this file is a instance with the following fields:

* Click ­– With in the displayed results, the number of times a user clicked the Ad.
* Impression ­– It is the count of search sessions in which the Ad was displayed by the user using the query.
* DisplayURL ­– In addition to title and description of Ad, DisplayURL is also a property of Ad, which is hashed.
* AdID – It is a unique Id given for a particular Ad.
* AdvertiserID – It is a unique Id given for a particular advertiser
* Depth – Number of ads displayed in a session for a query by a user.
* Position – The position of Ad in the displayed list for a query.
* QueryID – It is an Id given for a query, more over it is a key of ‘queryid’ data file.
* KeywordID – It is a unique Id for each and every word in the query, it is a key of ‘purchasedkeyword’ datafile.
* TitleID – a unique Id given for a title of Ad, it is a key of ‘titleid’ data file.
* DescriptionID – a unique Id given for description of Ad, it is a key of ‘descriptionid’ data file
* UserID – It is a unique Id given for each and every user, it is a key of ‘userid’ data file.

Out of these 6 datasets, training.txt is the only dataset which contains the entire data and the remaining 5 datasets are like additional or supportive datasets which will help to build training data.

In each line of these additional data files, Id’s and Tokens fields are separated by TAB character and it maps Id to a list of tokens which are delimited by character ‘|’ and in this dataset, each of these tokens are represented by hash values for anonimity.

In “userid\_profile.txt” dataset, it is composed of gender and age fields which can be described as follows,

* Gender

1 --> Male

2 --> Female

0 --> Unknown

* Age

1 🡪 (0,12]

2 🡪 (12,18]

3 🡪 (18,24]

4 🡪 (24,30]

5 🡪 (30,40]

6 🡪 >40

Snapshot of Training Dataset:



1. Hypothesis

Algorithm for Data Analysis:

The data is grouped on all the sessions with same User ID, Query, Ad ID, depth and position. We want to predict the click through rate of ads for this. Regression Techniques can be used because the output will be a real number between 0 and 1. The output feature will be the click column divided by impression column.

To check the accuracy of our results we have used MSE (Mean Squared Error) parameter. In this the actual output and the expected output is subtracted and the square of it is taken. So MSE is nothing but the sum of all squared errors divided by the total number.

Dimensionality reduction technique is incorporated to reduce the number of features considering the current feature set is high. We have used PCA technique for dimensionality reduction.

The data was analysed using R ANOVA technique. It was concluded that the data is non-linear. Due to this we conclude that linear regression is not a good fit. Also, the number of features used are high the depth of the decision tree would be greater, so we assume that random forest might be a good fit. In order to conclude the same we analysed the following supervised learning techniques:

Linear Regression: In this approach we try to find the output variable relationship with predictor variables (independent variables). If the output is linearly dependent then we go for linear regression analysis.

Random Forrest: This is an ensemble learning used for non-linear multiple regression. In Random Forest several regression trees are used and the output will be the mean of the regression trees output. Bootstrap aggregation is used for sampling the data.

Decision Tree: It builds regression models in tree structure. The dataset is broken into smaller chunks of data while the decision tree is developed incrementally. The final result is a tree with decision nodes and leaf nodes.

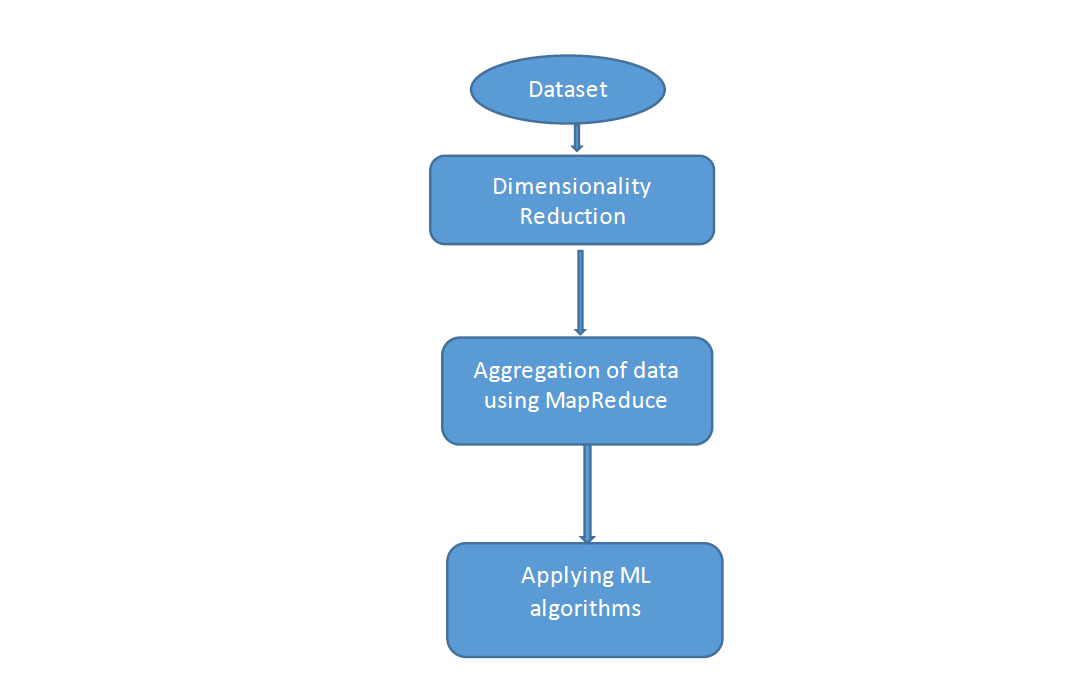
Boosting: In boosting a bunch of weak classifiers are used to make a strong classifier. It is mainly used to reduce variance and bias.

Big Data Strategy:

Spark is faster than traditional map reduce as it offers in memory computing and we can cache transformation for faster computation. Spark’s MLlib provides us the implementation of the algorithms discussed above.

We used the UTD cluster to execute the required functionality on the datasets.

Workflow:



Dimensionality Reduction:

Since the dataset has 10 features, if we do the analysis using all these features there might be a possibility of over fitting, also there are only few important features that contribute to the variability of the output. So we used PCA for dimensionality reduction and took 5 principle features which will explain the variability of the output.

Aggregation of data using Map Reduce:

The input data is already grouped under a setting so there is no need for data aggregation. We have formatted the data using map transformations. In formatting, first we have converted the tab delimited data into records with different column values and then we have divided the clicks column with impressions column which will be our dependent variable (output).

Applying ML Algorithms:

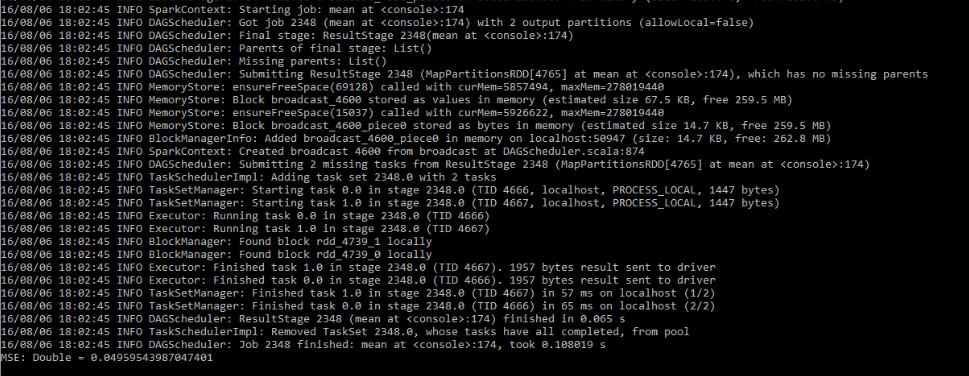
First we have split the data into 60% and 40% for training and testing respectfully. Then we have applied Spark MLlib for training and testing.

6. Analysis of Results:

Finally, we have built four regression models with and without dimensionality reduction which are Linear Regression, Decision Tree , Random Forest and Boosting. Here are the results we observed from each model,

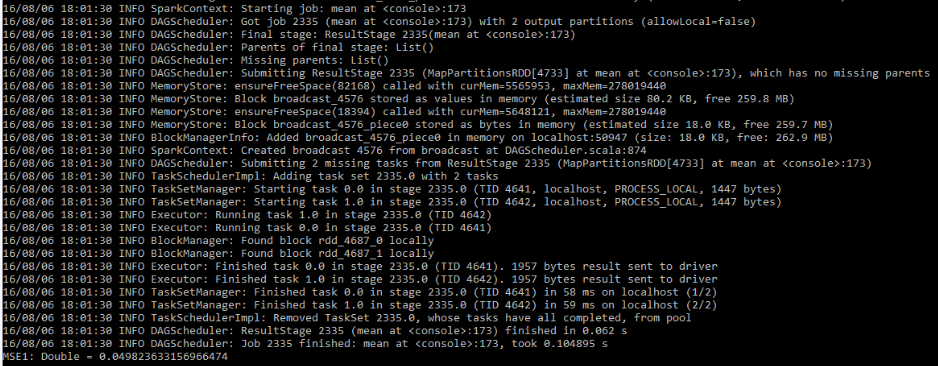
Linear Regression:

Without PCA:



Here, we got a Mean Squared Error of 0.049. It means that predicted value might vary about 0.22 from the real value. Even though, we didn’t know that distribution of output with respect to attributes is linear we have implemented this model to check if we get better results.

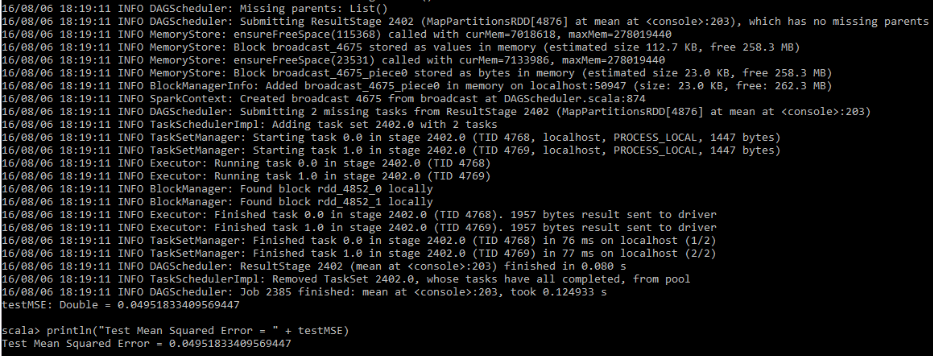
With PCA:



Here, we see that there is little increase in MSE after dimensionality reduction. We can say that only 5 attributes make most of the difference and there is no much overfitting with model without PCA.

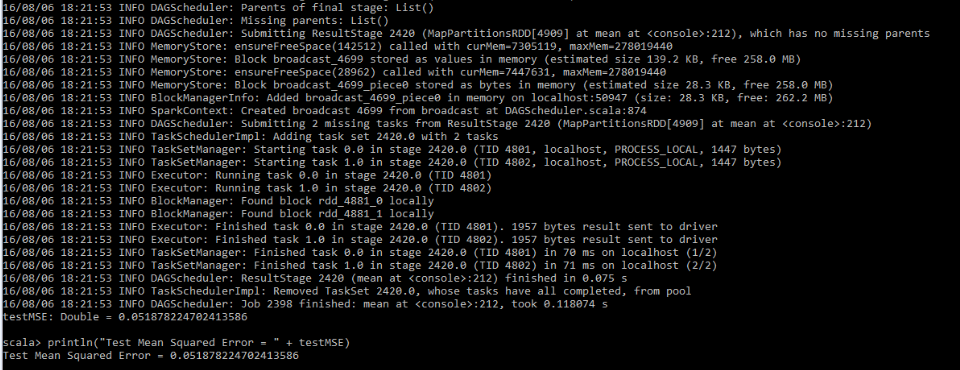
Decision Tree:

Without PCA:



We didn’t see much difference in MSE from Linear regression model and we considered all the features here to build the tree.

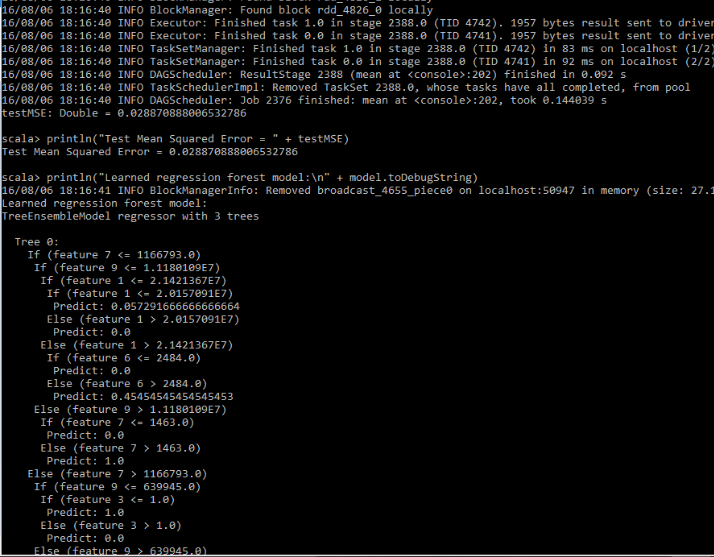
With PCA:



Here, we see that there is increase in MSE after dimensionality reduction.

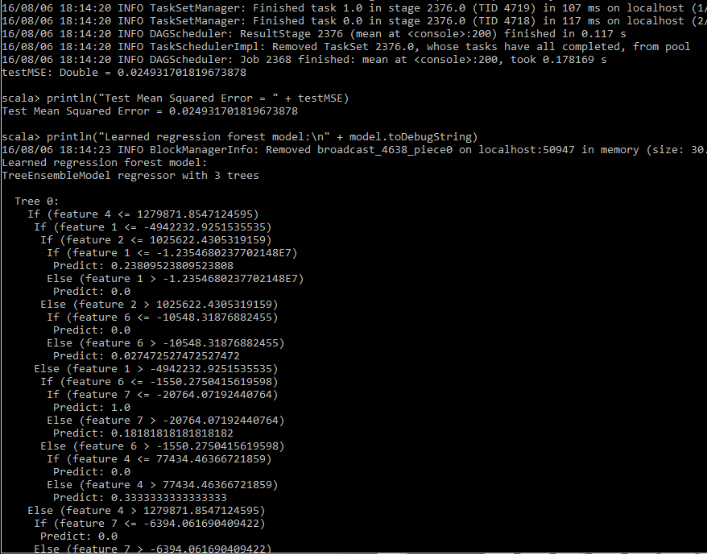
Random Forest:

Without PCA:



Here, we see that there is a great decrease in MSE with Random Forest. The great decrease in variance might be one of the reason and also as we got weaker results for Decision tree. There is a clear indication that combination of weak classifier gives strong classifier.

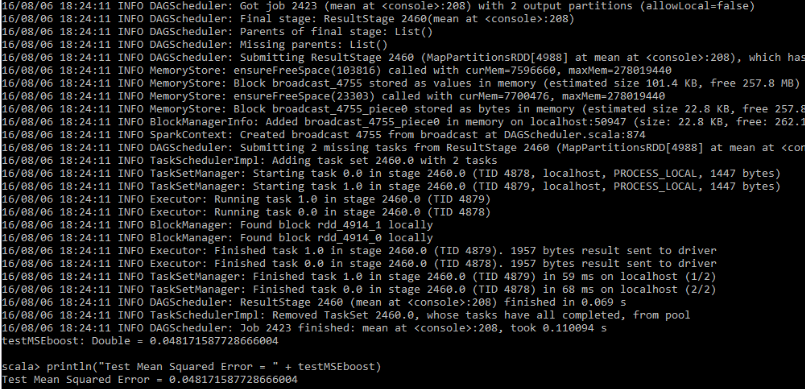
With PCA:



Here, we can see that there is not any much difference even after dimensionality reduction. We feel that there are only about 5 attributes which are making most of the difference in output variable.

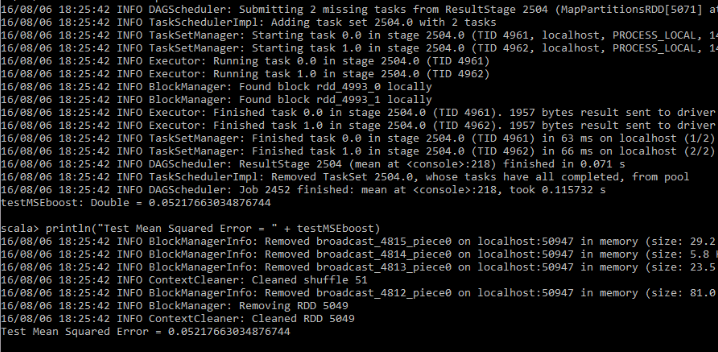
Boosting:

Without PCA:



Here, we can see that there is no any much improvement in MSE from linear regression.

With PCA:



Here, we see that there is little increase in MSE after dimensionality reduction.

|  |  |  |
| --- | --- | --- |
| Model | MSE(Without dim. reduction) | MSE (With dim. reduction) |
| Linear regression | 0.0495 | 0.0498 |
| Decision Tree | 0.0495 | 0.0518 |
| Random Forest | 0.0288 | 0.0249 |
| Boosting | 0.0481 | 0.0521 |

From the above all, we got better results with Random Forest with less MSE. The reason might be that the Random Forest is a combination of regression and there will be great reduction in variance. But, by Random Forest we lost some interpretability but got better results.