**DATA ANALYSIS ASSIGNMENT**

**INTRODUCTION**

This assignment is all about development of applications to analyze and create insights for a publicly available dataset. The entire work can be divided into the following sub-tasks :

* Data cleaning
* Data ingestion
* Query
* Apply ML algorithms to classify the data
* Visualization

The dataset used as input is all about classification of two different income groups of adults in USA. It is obtained from kaggle.com. (<https://www.kaggle.com/wenruliu/adult-income-dataset/version/2>). Data is cleaned using Numpy and Pandas python based libraries. Data ingestion, query and visualization are bounded in a single Spring MVC application. Here, the cleaned dataset is stored in MongoDB. And a backend rest api that is built using Spring Boot facilitates the transaction to the database from the UI.

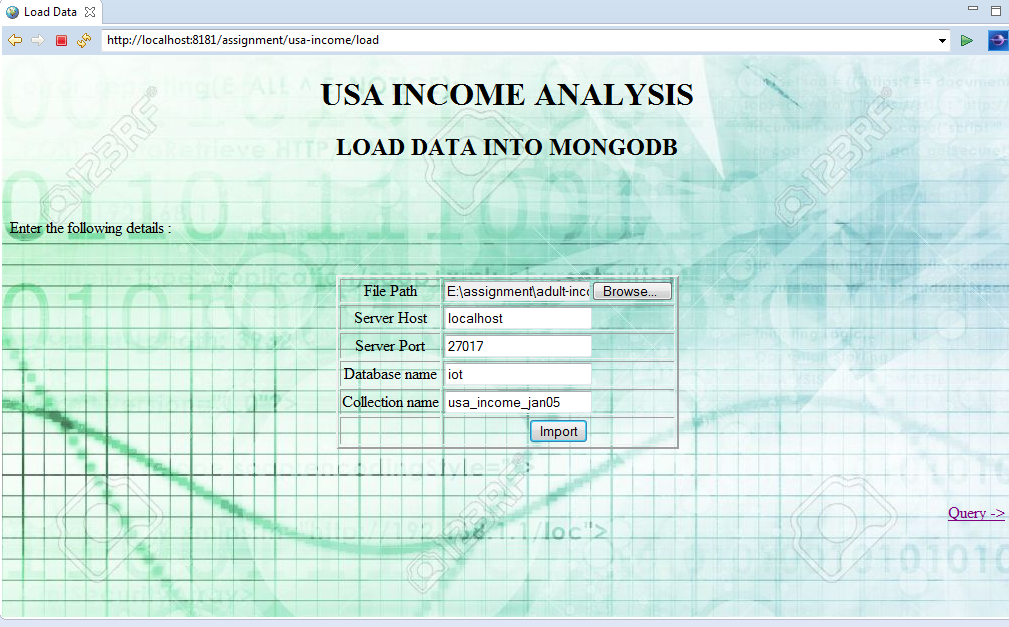
**DATA CLEANING**

Before storing the data in a database, it should be made or transformed in such a way that it can be subjected to any ML algorithms in future. For this purpose, all fields which does not have any value are removed and also all string values are created into corresponding binary classified numerical values. For example, the field “workclass” has different types of values like, private, Local-gov, Fed-gov, etc.. For simplicity it can be classified into only two types of values – gov and private. If the person belongs to a government organization, then that field will be given as 1 and non-government persons will be given 0. The same kind of imputing of values is done in all the other fields also. Also, few fields like education, occupation, relationship will not contribute any effect in generating the result in the classification problem, so they are also dropped from the dataframe. Therefore, the cleaned data will be as follows:

|  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| row | age | fnlwgt | educational-num | capital-gain | capital-loss | hours-per-week | workclass | marital-status | race | Male | native-country | >50K |
| 0 | 25 | 226802 | 7 | 0 | 0 | 40 | 0 | 0 | 0 | 1 | 1 | 0 |
| 1 | 38 | 89814 | 9 | 0 | 0 | 50 | 0 | 1 | 1 | 1 | 1 | 0 |
| 2 | 28 | 336951 | 12 | 0 | 0 | 40 | 1 | 1 | 1 | 1 | 1 | 1 |
| 3 | 44 | 160323 | 10 | 7688 | 0 | 40 | 0 | 1 | 0 | 1 | 1 | 1 |
| 5 | 34 | 198693 | 6 | 0 | 0 | 30 | 0 | 0 | 1 | 1 | 1 | 0 |

**DATA INGESTION**

Data ingestion is done using mongo-java-driver, which is a library that is used to perform MongoDB database transactions using java. From the UI, the user can choose the file containing the cleaned data and if we click on the import button, the file present in our local file system will be converted into a MongoDB document and inserted into a MongoDB collection.



Following are the logs displayed in the console.

Jan 05, 2019 10:12:24 PM com.mongodb.diagnostics.logging.JULLogger log

INFO: Cluster created with settings {hosts=[localhost:27017], mode=SINGLE, requiredClusterType=UNKNOWN, serverSelectionTimeout='30000 ms', maxWaitQueueSize=500}

Jan 05, 2019 10:12:24 PM com.mongodb.diagnostics.logging.JULLogger log

INFO: Opened connection [connectionId{localValue:1, serverValue:103}] to localhost:27017

Jan 05, 2019 10:12:24 PM com.mongodb.diagnostics.logging.JULLogger log

INFO: Monitor thread successfully connected to server with description ServerDescription{address=localhost:27017, type=STANDALONE, state=CONNECTED, ok=true, version=ServerVersion{versionList=[3, 0, 4]}, minWireVersion=0, maxWireVersion=3, maxDocumentSize=16777216, logicalSessionTimeoutMinutes=null, roundTripTimeNanos=9753873}

Jan 05, 2019 10:12:24 PM com.mongodb.diagnostics.logging.JULLogger log

INFO: Cluster description not yet available. Waiting for 30000 ms before timing out

Jan 05, 2019 10:12:24 PM com.mongodb.diagnostics.logging.JULLogger log

INFO: Opened connection [connectionId{localValue:2, serverValue:104}] to localhost:27017

45222 records are inserted into the collection usa\_income\_jan05.

Mongodb commands to check the documents inserted :

> use iot

switched to db iot

> show collections

device

emp

employee

employees

emptph

pets

system.indexes

ticket

timestamp

usa\_income

usa\_income\_jan05

user\_collection

user\_jan01

> db.usa\_income\_jan05.find()

{ "\_id" : ObjectId("5c30de708f375710c0846854"), "capital-loss" : 0, "race" : 0,

"native-country" : 1, "hours-per-week" : 40, "Male" : 1, "educational-num" : 7,

"capital-gain" : 0, ">50K" : 0, "fnlwgt" : 226802, "workclass" : 0, "row" : 0,

marital-status" : 0, "age" : 25 }

{ "\_id" : ObjectId("5c30de718f375710c0846855"), "capital-loss" : 0, "race" : 1,

"native-country" : 1, "hours-per-week" : 50, "Male" : 1, "educational-num" : 9,

"capital-gain" : 0, ">50K" : 0, "fnlwgt" : 89814, "workclass" : 0, "row" : 1, "

arital-status" : 1, "age" : 38 }

{ "\_id" : ObjectId("5c30de718f375710c0846856"), "capital-loss" : 0, "race" : 1,

"native-country" : 1, "hours-per-week" : 40, "Male" : 1, "educational-num" : 12

"capital-gain" : 0, ">50K" : 1, "fnlwgt" : 336951, "workclass" : 1, "row" : 2,

"marital-status" : 1, "age" : 28 }

{ "\_id" : ObjectId("5c30de718f375710c0846857"), "capital-loss" : 0, "race" : 0,

"native-country" : 1, "hours-per-week" : 40, "Male" : 1, "educational-num" : 10

"capital-gain" : 7688, ">50K" : 1, "fnlwgt" : 160323, "workclass" : 0, "row" :

3, "marital-status" : 1, "age" : 44 }

{ "\_id" : ObjectId("5c30de718f375710c0846858"), "capital-loss" : 0, "race" : 1,

"native-country" : 1, "hours-per-week" : 30, "Male" : 1, "educational-num" : 6,

"capital-gain" : 0, ">50K" : 0, "fnlwgt" : 198693, "workclass" : 0, "row" : 5,

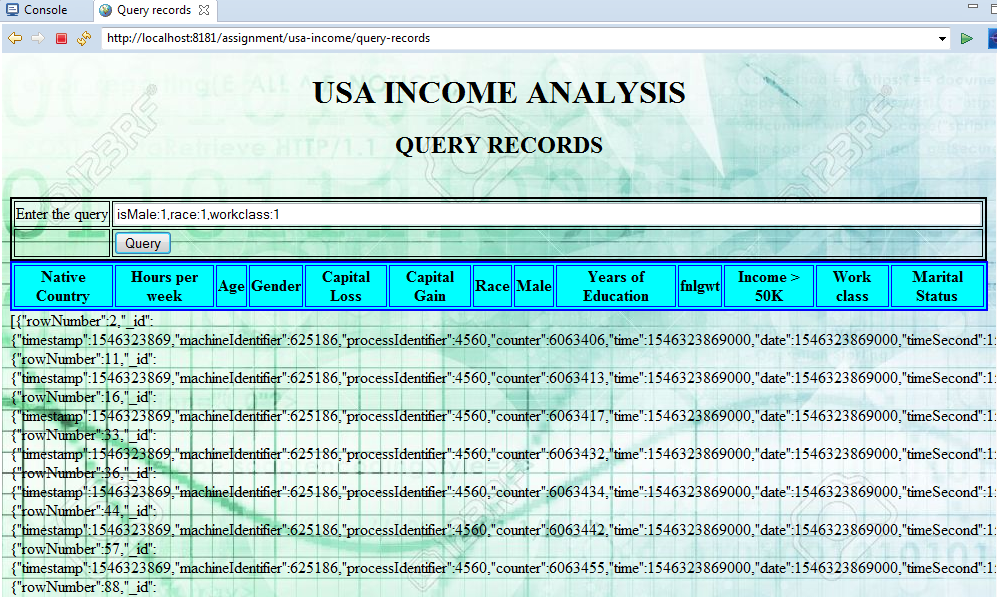
marital-status" : 0, "age" : 34 }

> db.usa\_income\_jan05.count()

45222

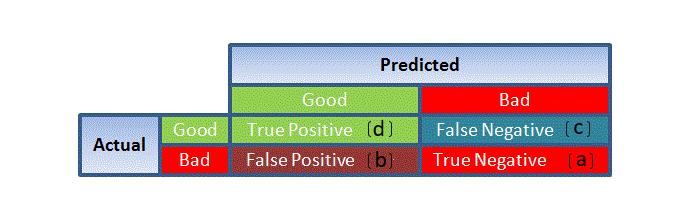
**QUERY (Under Construction)**

When we click on the Query -> link, it will take us to another page which allows the user to enter the criteria on which the records are displayed. As of now, this page is under construction. It just displays the records as json but not in the form of table.



**APPLYING ML ALGORITHMS TO CLASSIFY THE DATA**

This dataset can be used to solve logistic regression problem. That is, the available dataset can be broadly classified into persons having income>50 and <=50K. For this, sklearn python based library is used and finally the confusion matrix for the dataset is generated. Confusion matrix is nothing but a tabular representation of Actual vs Predicted values. This helps us to find the accuracy of the model



The confusion matrix for 30% of the test dataset is as follows :

array([[9876, 363],

[2438, 890]], dtype=int64)

**VISUALIZATION:**

Unser construction

**CONCLUSION**

Thus a dataset is analyzed, cleaned and transformed and ML insights are created using Spring framework and other Python based ML libraries.