

-



**Data Science Eng With Python**

**FDA Adverse Event Reporting System**

**(FAERS)**

**Final Project Report**

**Team 1**

Priyal Chaudhari

Ranadeep Guha

Prashant Vikram Singh

SUMMARY

The report summarizes the analysis performed on FDA Adverse Event Reporting System, provided byFDA: <https://www.fda.gov/Drugs/GuidanceComplianceRegulatoryInformation/Surveillance/AdverseDrugEffects/ucm082193.htm>

We then build multi-class classification machine learning models using the datasets to classify the outcome of the drug prescribed by the health professional and reaction associated with the drug. The problem presented is divided into 2 sections:

**Section 1: Data wrangling**

* Data Download and pre-processing
* Exploratory Data analysis in Python

**Section 2: Building and evaluating model**

* **Classification** using Multi-Class Logistic Regression, Multi-Class Random Forest, Multi-Class Neural Network, One vs All Multi-Class SVN Algorithms

1 PART 1: DATA INGESTION AND WRANGLING

1.1 THE DATA

***FAERS (FDA Adverse Event Reporting System) Dataset***

The FDA Adverse Event Reporting System (FAERS) is a database that contains information on adverse event and medication error reports submitted to FDA.The FDA produces [FDA Adverse Event Reporting System (FAERS)](http://www.fda.gov/Drugs/GuidanceComplianceRegulatoryInformation/Surveillance/AdverseDrugEffects/ucm082193.htm) quarterly data. The FAERS include the following:

* demographic and administrative information and the initial report image ID number (if available);
* drug information from the case reports;
* reaction information from the reports;
* [patient outcome information from the reports;](http://www.nber.org/data/fda-adverse-event-reporting-system-faers-data.html#outc)
* [information on the source of the reports;](http://www.nber.org/data/fda-adverse-event-reporting-system-faers-data.html#outc)
* [a "README" file containing a description of the files.](http://www.nber.org/data/fda-adverse-event-reporting-system-faers-data.html#outc)

**Data Download and pre-processing:**

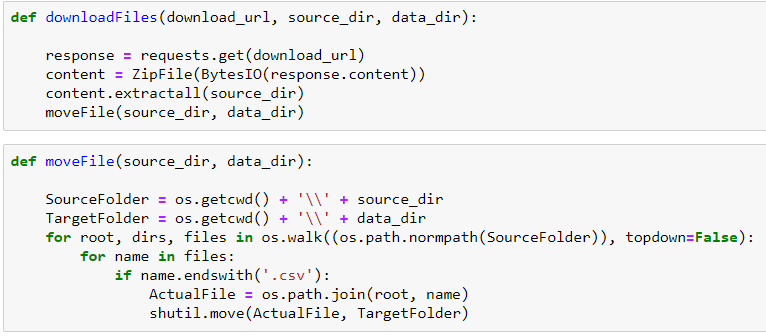
The very first challenge was to programmatically download the data from FDA website (<http://www.fda.gov>), download and preprocess the “FAERS” data file starting from Q1 2014 till Q1 2018.

To download the file programmatically, without any manual intervention we have used the python requests library to navigate to the website and created source directory (FAERSsrc), where all the related files will be downloaded and destination directory where all the files required for our analysis will be saved.

We will be using **Beautiful Soup** package, a powerful python package for data scrapping from the FDA website and download all the quarterly files for the “FAERS” files for our analysis purpose.



After getting the files from the FDA website, we will download all the quarter file in unzipped formed in our source folder and then we will move all the required file into the destination folder.



The files which are using for our analysis are:

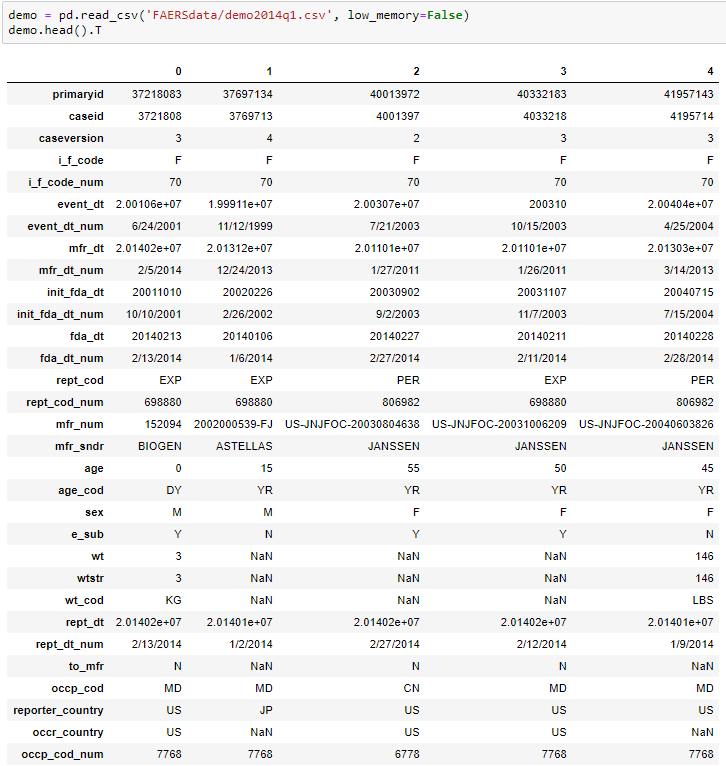
* demographic and administrative information and the initial report image ID number (if available);
* drug information from the case reports;
* reaction information from the reports;
* [patient outcome information from the reports;](http://www.nber.org/data/fda-adverse-event-reporting-system-faers-data.html#outc)

1.2 Data Preprocessing and Cleaning

Since we are working on quarterly data set, we need to merge all the data from all the quarter to create a single data set for our model training and testing. To achieve this goal, we will join our data present in four different files into one frame based on the primary reported Id and case Id.

We start with the demographic file and join the data of this file with the drugs reported based on the keys, which will be joined with the reaction dataset and the outcome of a drug and reaction.

**DEMO.CSV**

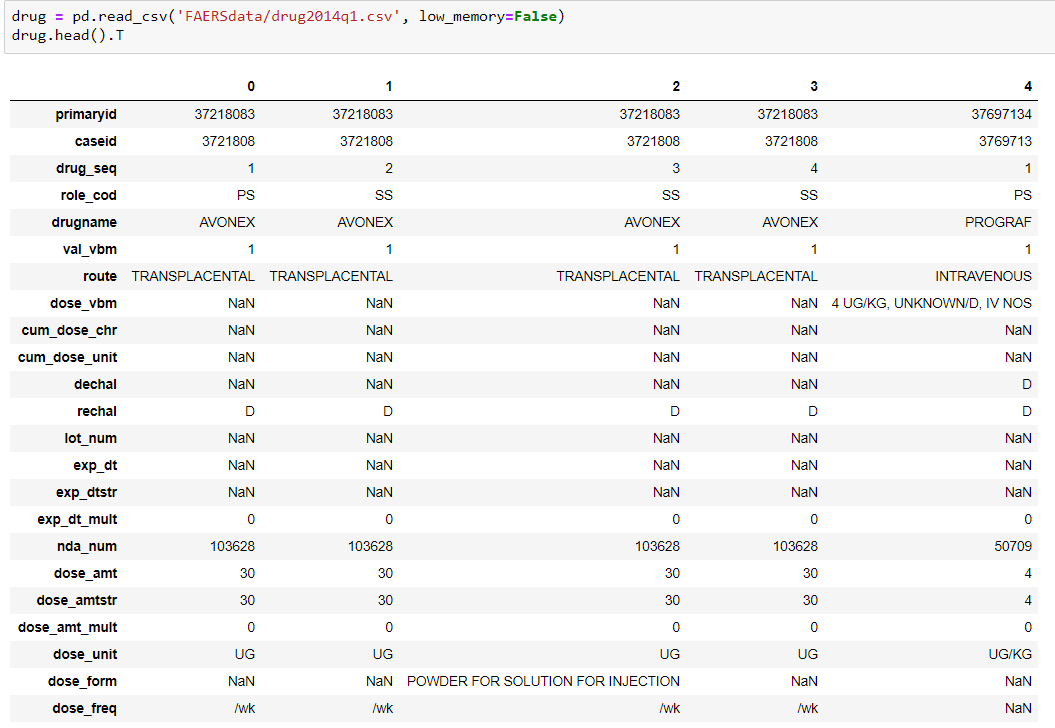


We have many Null values and spaces (an invalid values) which we need to handle before using these files to create a final dataset. We must make sure that our data is in proper format with same datatype. Since we have large amount of data to be processed, we choose to keep only those rows where we have all the details related to Drug and their adverse react.

**DRUG.CSV**

Drug files consist of all the drug which are reported to FEARS database with the dosage type, amount. Dosage form and other details related to the drug.

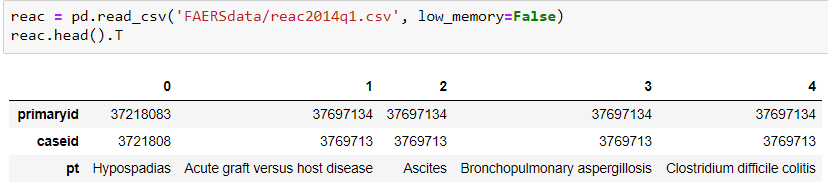
NOTE: Since we have millions of drugs reported, we have taken only those drugs which are reported as primary suspect for causing the reaction.



As we have many empty column values in our drug file, we need to clean those to ensure that we don’t have any NAN/NA value in our data. Also, we need to take care of the data type of column. These columns will be required for creating the final dataset.

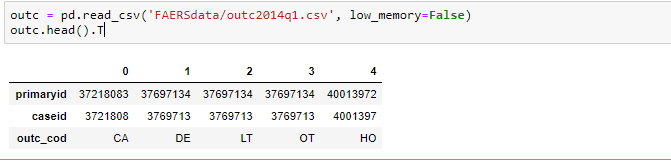
**REACTION.CSV**

Reaction files consist of all the reaction reported based on the number of drug associated with the case Id. With one drug, we have many associated reactions which are reported to FAERS database. We don’t have any Null/NAN values in the reaction files.



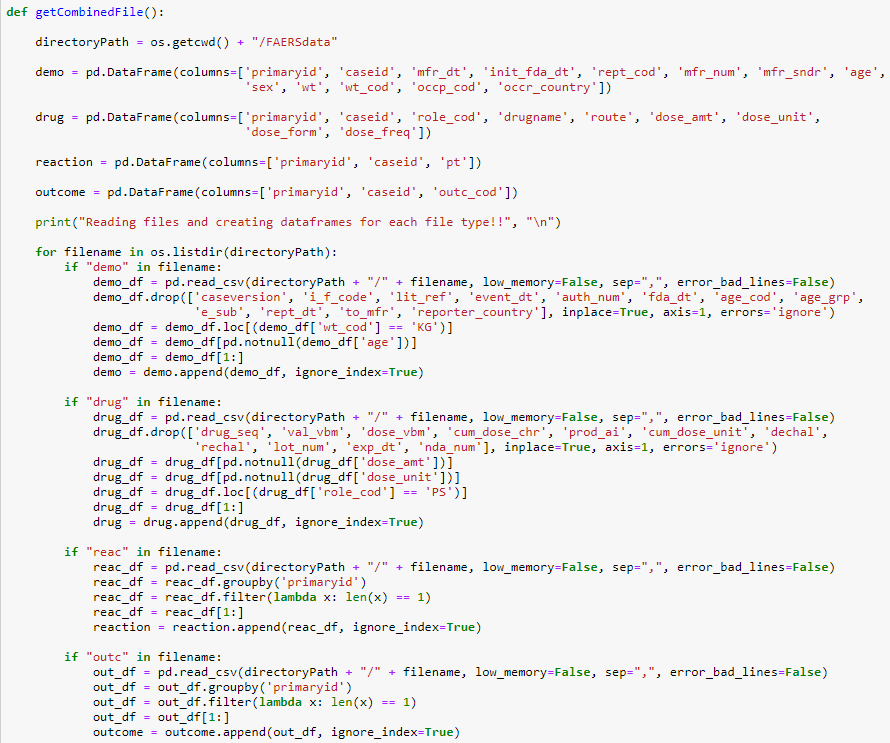
**OUTCOME.CSV**

Outcome files consist of the outcome reported based on the number of drug and reaction associated with a case Id. We don’t have any Null/NAN values in the outcome files.



We have used the standard Reaction Name as proposed by **MedDRA or Medical Dictionary for Regulatory Activities** which is a clinically validated international medical terminology dictionary used by regulatory authorities in the pharmaceutical industry.

For Drug Name standardization, we have used **RxNorm** provides normalized names for clinical drugs and links its names to many of the drug vocabularies commonly used in pharmacy management

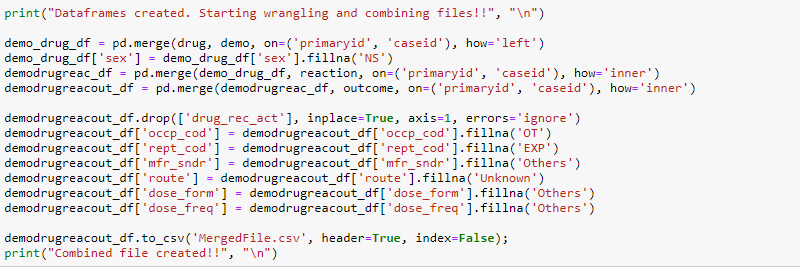


1.3 Creating Cleaned FAERS Dataset CSV file (Output)

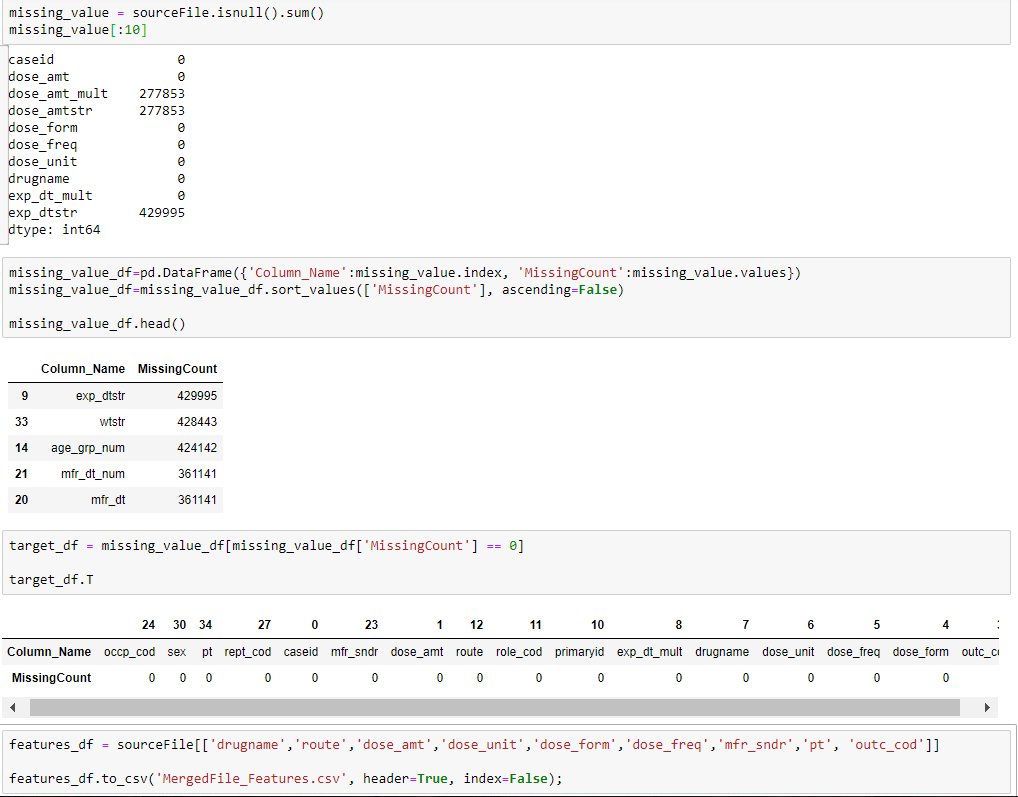
The result will be a joined dataset consisting of all the details of Drug, Reaction and their outcome

in a csv formed.

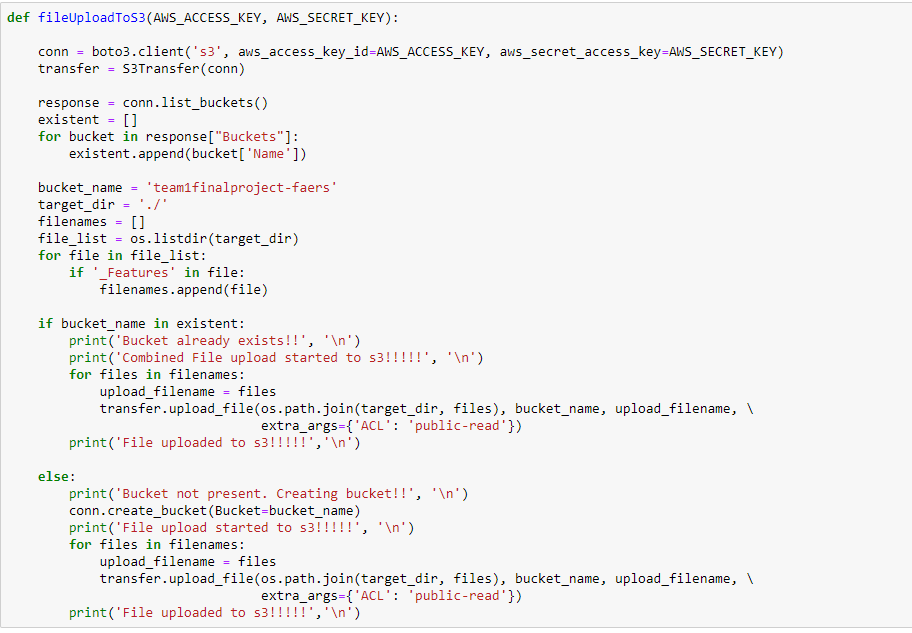
Once, we are done with the cleaning of all the four files, we will create a summarized data set using the python joins file based on primary report id and case id. For this step, we created a function which will join all the four-data frame associated with the final data set.



Then we computed the count of null values in a column and dropped all those columns where NULL is present even after wrangling and wrote the final output file.



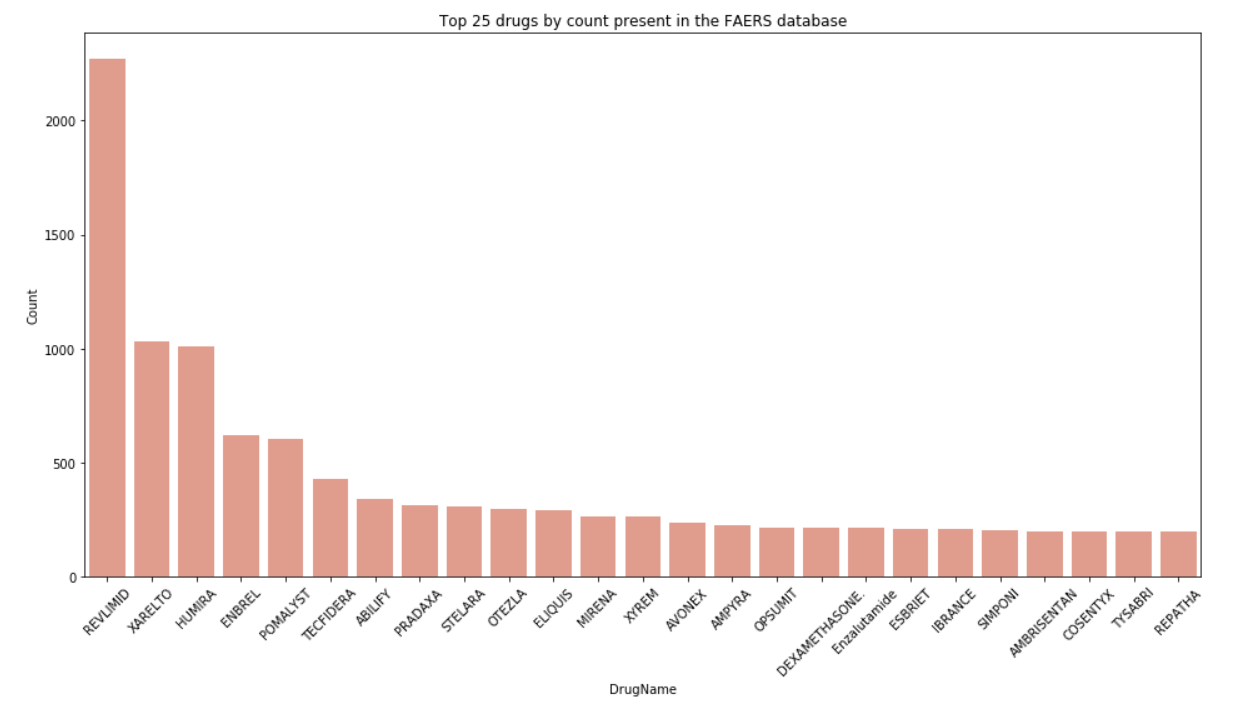
Then we uploaded the features file to Amazon S3 which can then be consumed on Azure ML and Amazon ML platform for Machine learning programming.



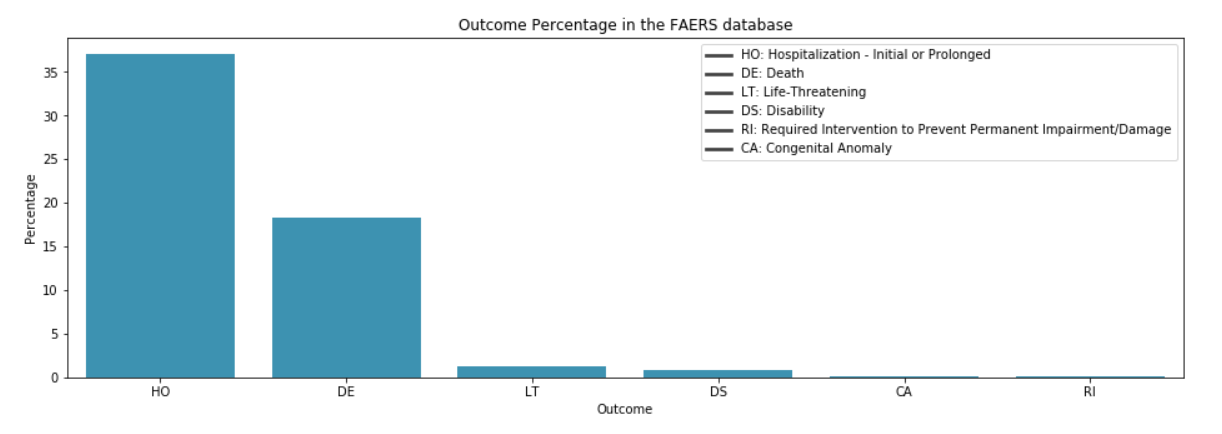
PART 2: Exploratory Data Analysis

2.1 Analysis – Python (MatplotLib, Seaborn)

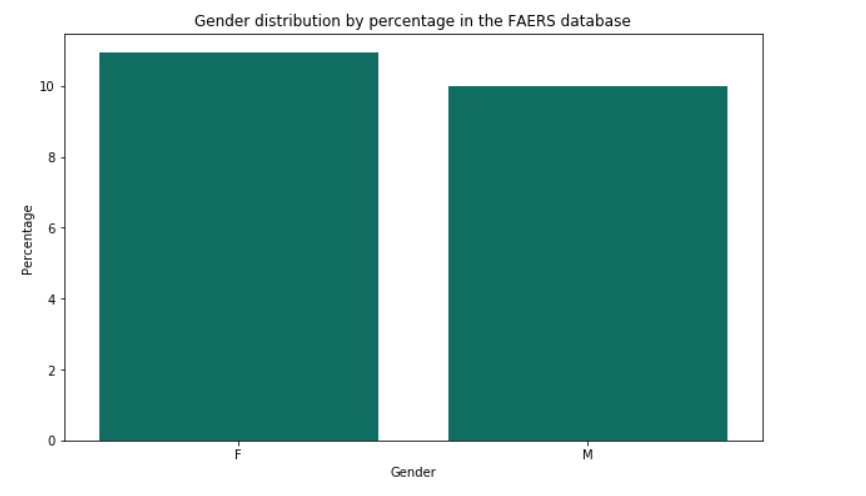
* **Drug Counts**: contains the top 25 drugs by count present in the FAERs database.



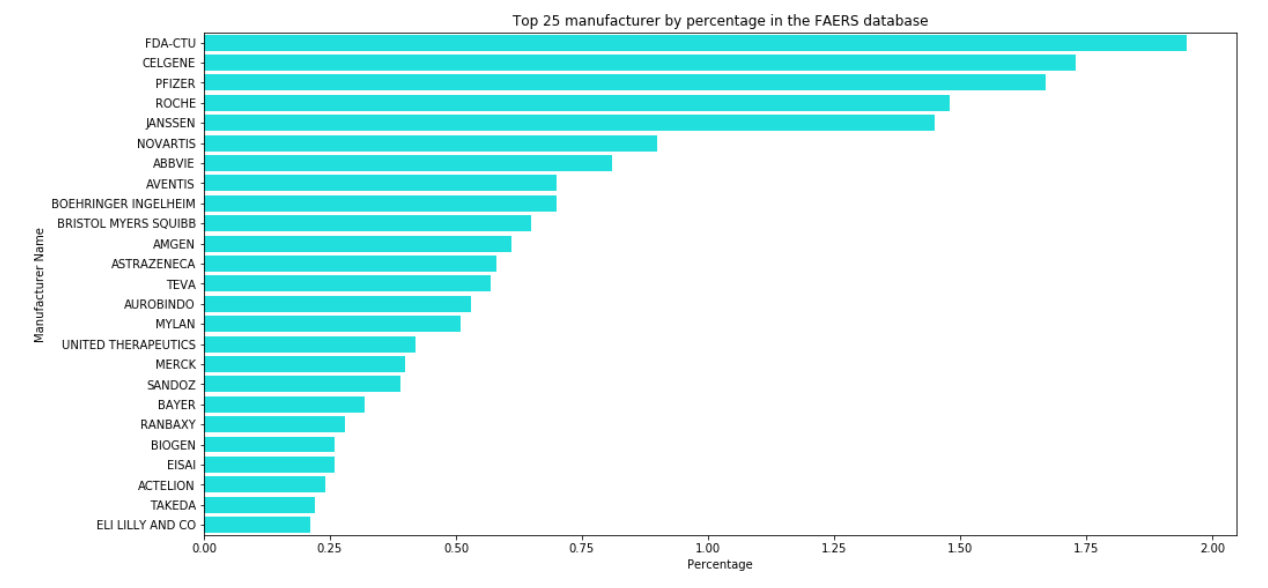
* **Outcome distribution**: provides you the percentage of the outcome distribution based on existing data.



* **Gender**: gives you the % amount about the gender category. Majority of the values in this field were NULL so, we have eliminated those while creating this graph



* **Manufacturer distribution**: Top 10 Manufactures which have been reported and % amount by cases received. Majority of the values in this field were NULL so, we have eliminated those while creating this graph.



**PART 3:** **MULTI - CLASS CLASSIFICATION**

**What Is Multiclass Classification?**

Each training point belongs to one of N different classes. The goal is to construct a function which, given a new data point, will correctly predict the class to which the new point belongs.

In our cleaned FAERS dataset, we have a column named **outc\_cod**, which signifies the primary outcome for e.g. (‘OT’,’DE’) drug intake and reaction occurs. In our project, we are using multi-class classification to predict the probability of the outcome based on features selected from feature engineering.

CODE MEANING\_TEXT

---- ------------

DE Death

LT Life-Threatening

HO Hospitalization - Initial or Prolonged

DS Disability

CA Congenital Anomaly

RI Required Intervention to Prevent

Permanent Impairment/Damage

OT Other

3.1 GENERIC APPROACH: CLASSIFCATION

As we have already cleaned the FAERS data file, we will be using the same file for our model training and testing.

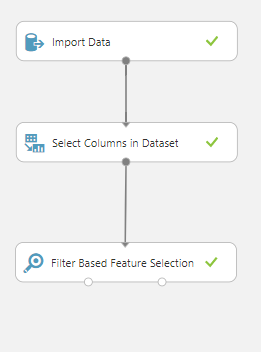
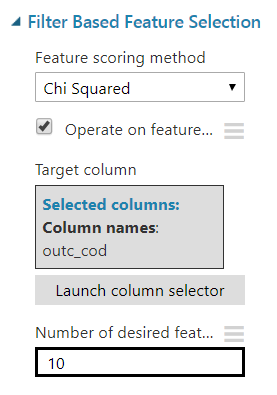
We will use three approaches to find out the outcome of our classification model:

* Using Azure ML Studio (For selecting the best model for our problem)
* Using Amazon ML web service (uses **Stochastic gradient descent**)
* Python Sklearn Library to test Neural Network

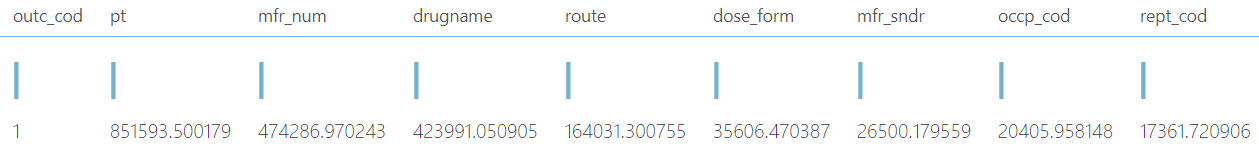
3.1.1 FEATURE SELECTION

Before proceeding with our models, we have done feature selection using **Chi Squared method**.

The best features that add to the predictive power of the model will be retained and irrelevant features removed from the model.

We have the selected variables as per the **Chi squared Method** and used those for further analysis while making sure that all the datasets contain same number of columns. Performing all the feature selection methods we shortlisted below features to best predict our model.



Now, we will discuss the algorithm used in the below section for all the Machine Learning algorithm used for classifications

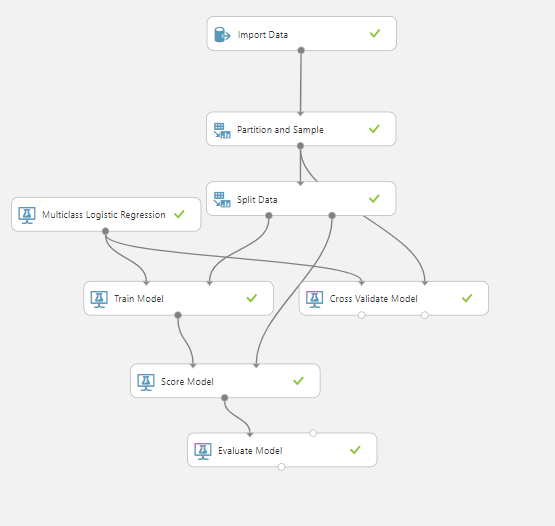
3.1.2 DIFFERENT MACHINE LEARNING ALGORITHMS AND OUTPUT

We have used the following Algorithm for our Multiclass classification problem namely:

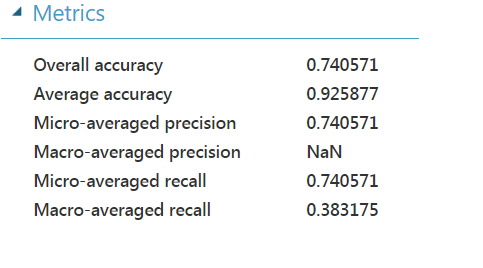
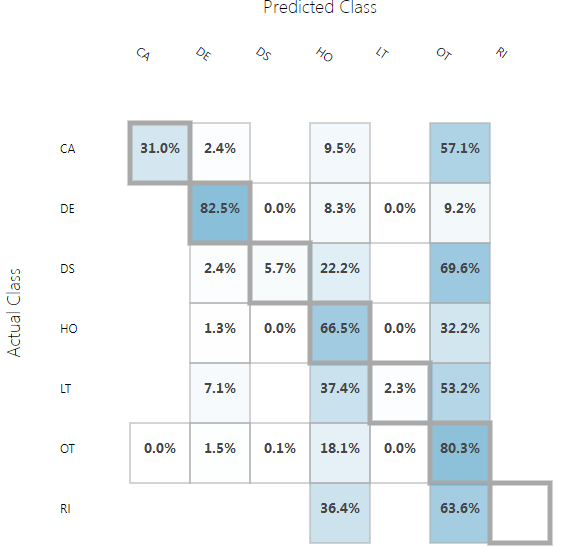
* **Multiclass Logistic Regression**
* **Multi-class Decision Forest**
* **Multiclass Neural Network**

**Multiclass Logistic Regression**

Multiclass logistic regression is used when the [dependent variable](https://en.wikipedia.org/wiki/Dependent_variable) in question is [nominal](https://en.wikipedia.org/wiki/Level_of_measurement#Nominal_measurement) (equivalently categorical, meaning that it falls into any one of a set of categories which cannot be ordered in any meaningful way) and for which there are more than two categories.

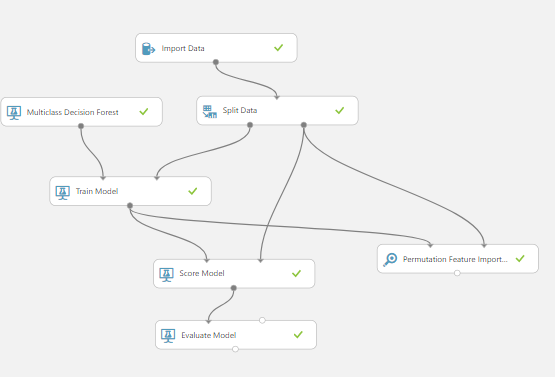


**Result evaluation**

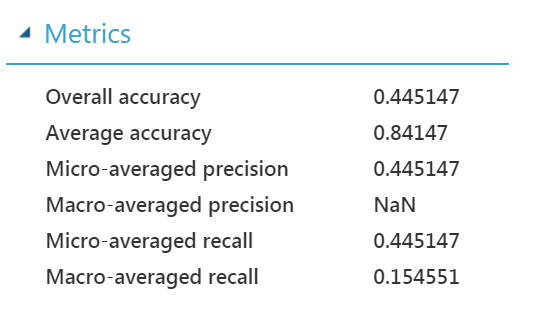
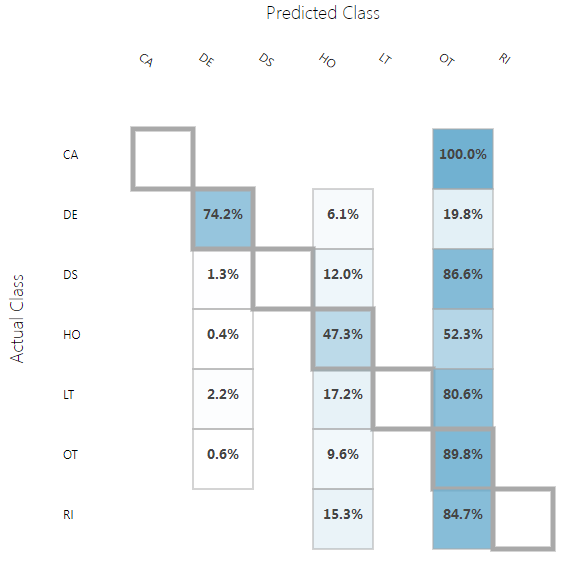
 

**Multi-class Decision Forest**

We can use the **Multiclass Decision Forest** module to create a machine learning model based on the *decision forest* algorithm. A decision forest is an ensemble model that very rapidly builds a series of decision trees, learning from tagged data.



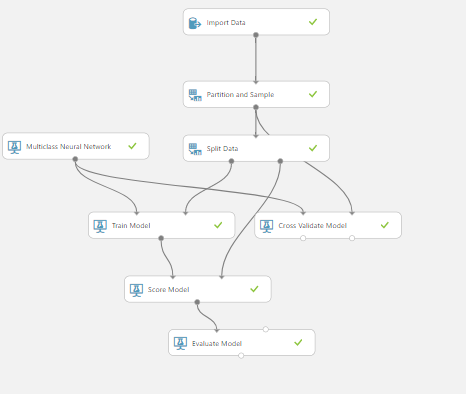
**Result evaluation**

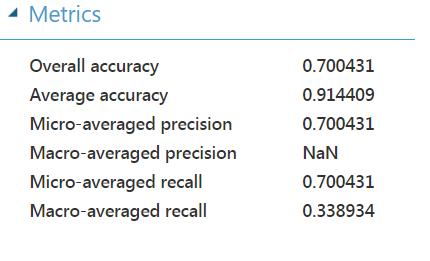
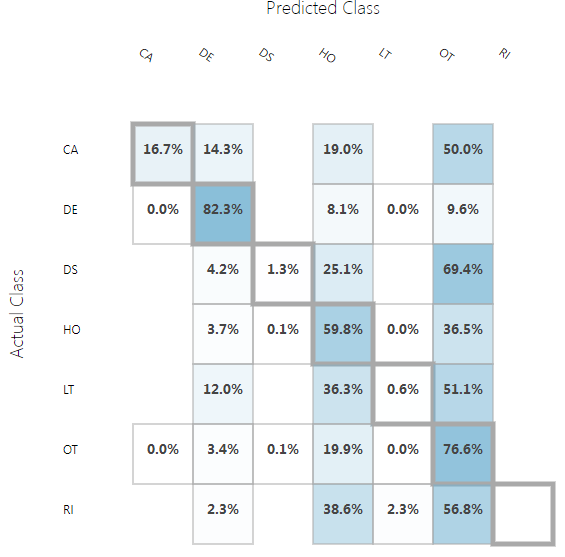
**Multiclass Neural Network**

We can use the **Multiclass Neural Network** module to create a neural network model that can be used to predict a target that has multiple values. For example, neural networks are frequently used in complex computer vision tasks, such as digit or letter recognition, document classification, and pattern recognition.

Classification using neural networks is a supervised learning method, and therefore requires a *tagged dataset* that includes a label column.



**Result evaluation**

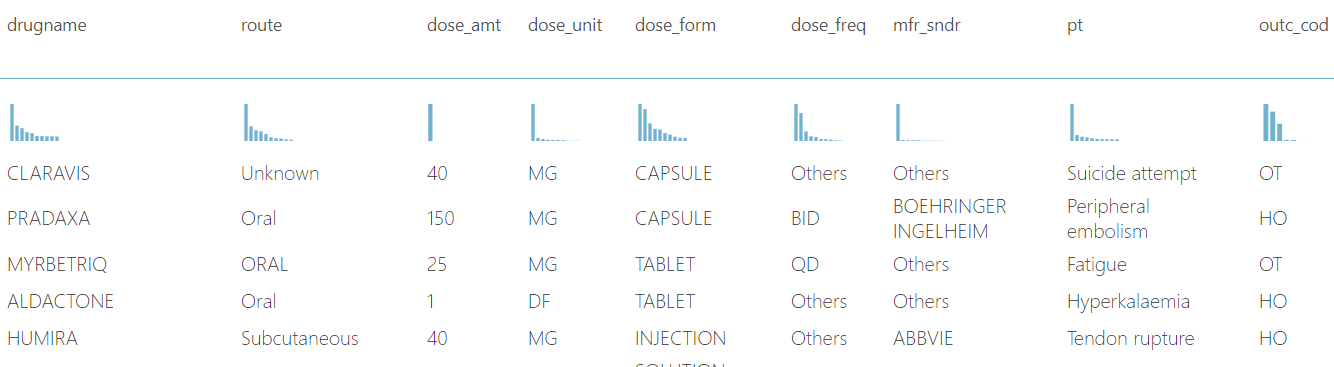
**3.1.3** **SELECTING BEST MODEL FOR OUR PROBLEM STATEMENT**

After we run all the models, we select **Multiclass** **Neural Network** as our best trained Model to classify the outcome of the drug and Reaction with other factor affecting like Dosage Amount, frequency etc.

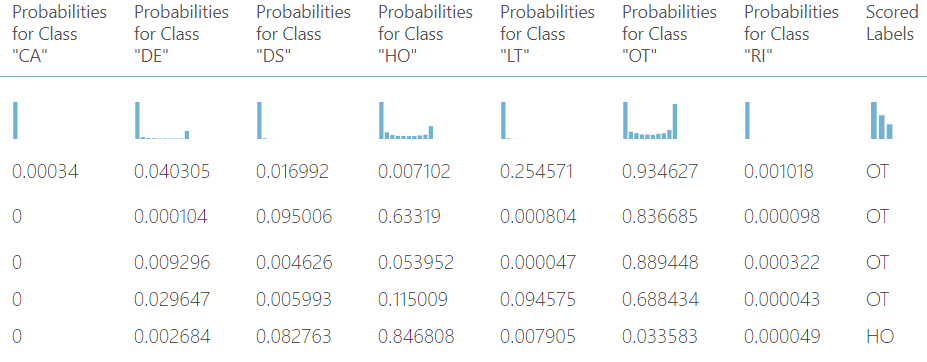
We have chosen this model based on the following parameters:

* Accuracy
* Recall

**Scored Labels:** When we visualize the Score Model component we get scored labels

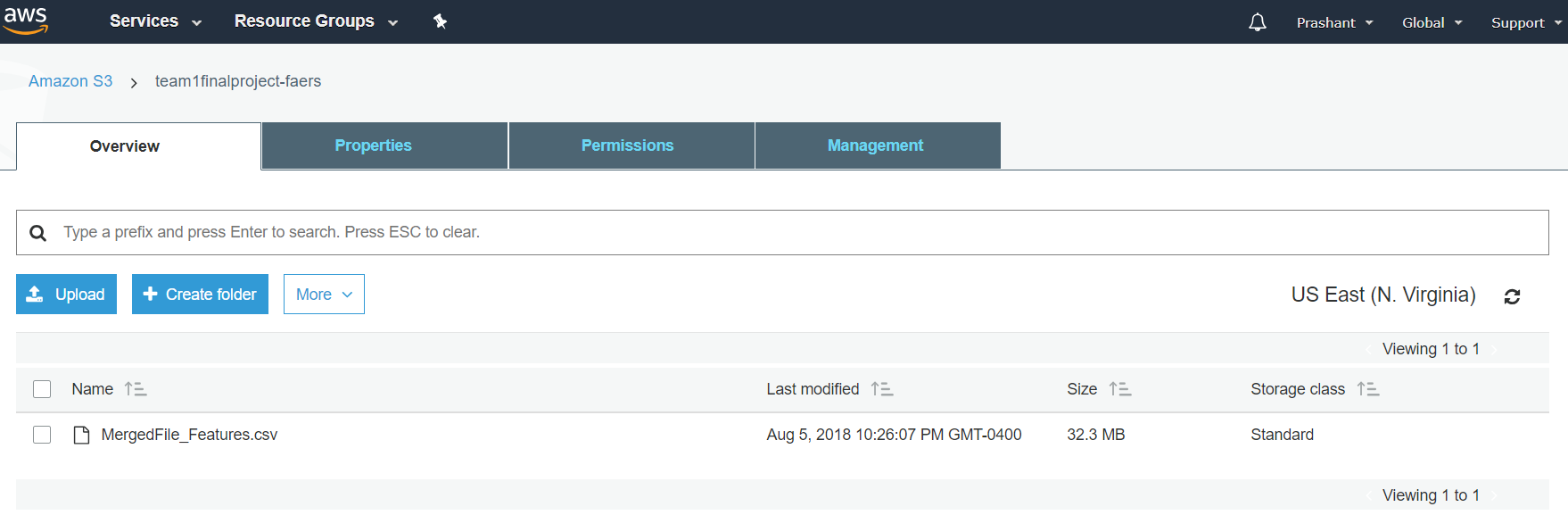


**Scored Probabilities:** When we visualize the Score Model component we get scored probabilities

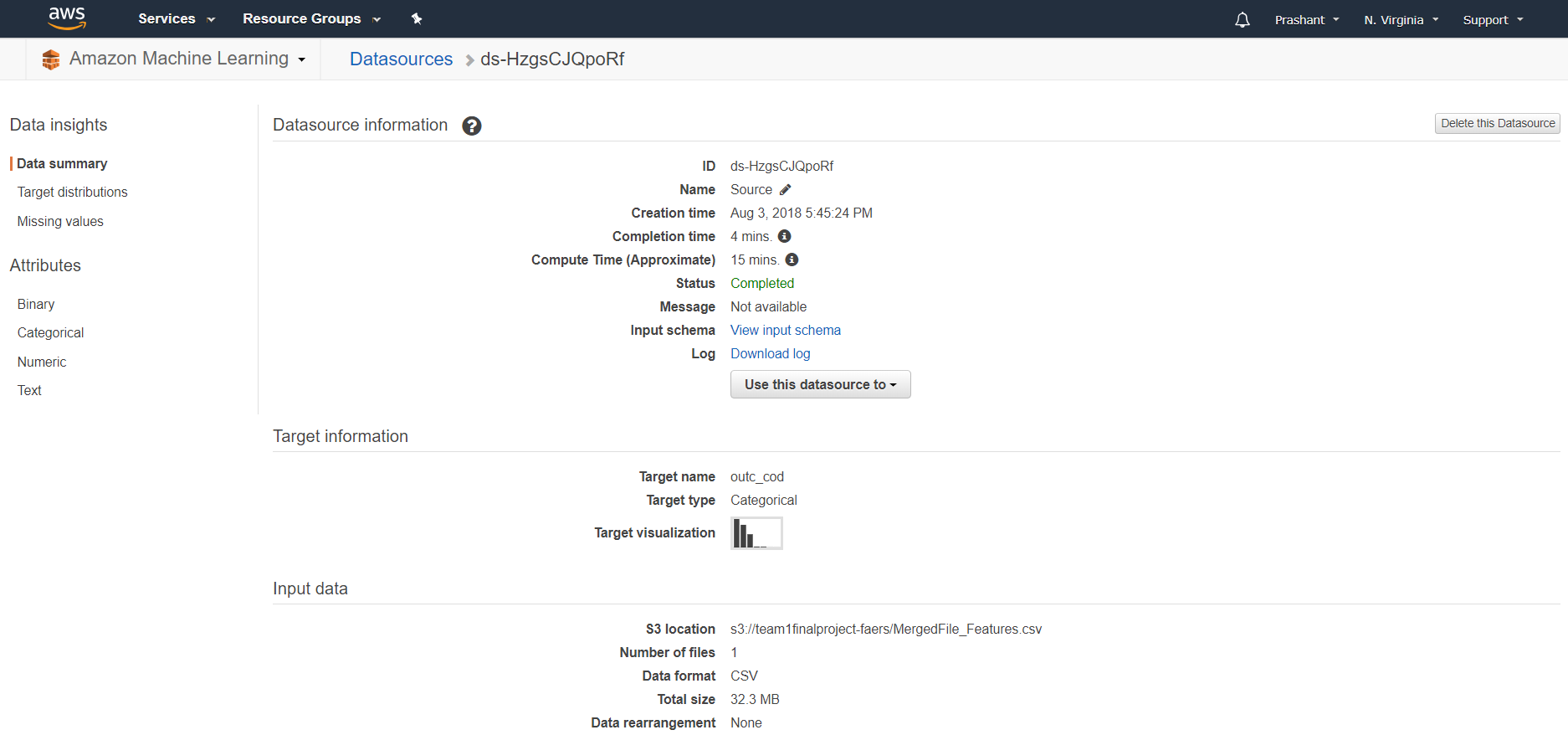


**3.1.4 AMAZON MACHINE LEARNING WEBSERVICE**

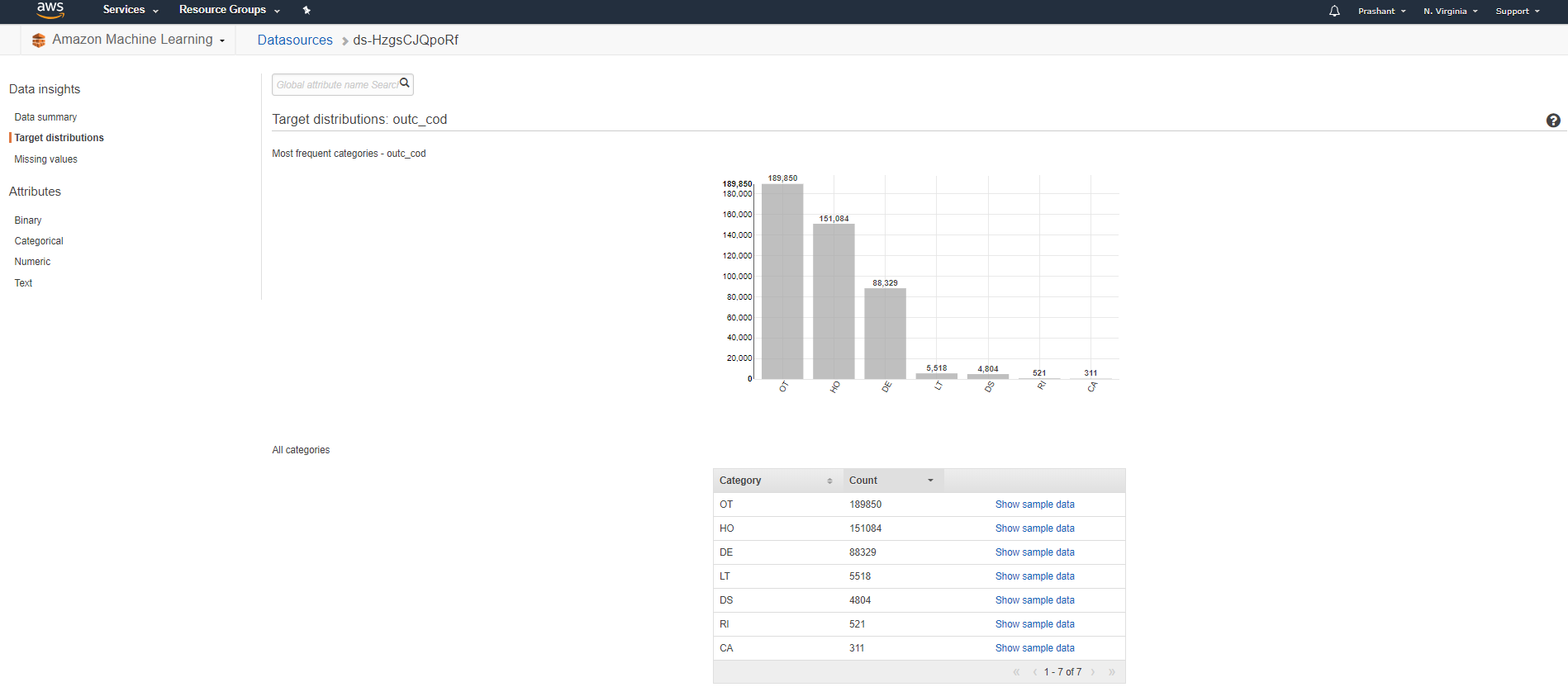
1. Taken the cleaned csv file(MergedFile\_Features.csv) from Amazon S3, which was uploaded by our PART-1 web scrapping and cleaning python script.



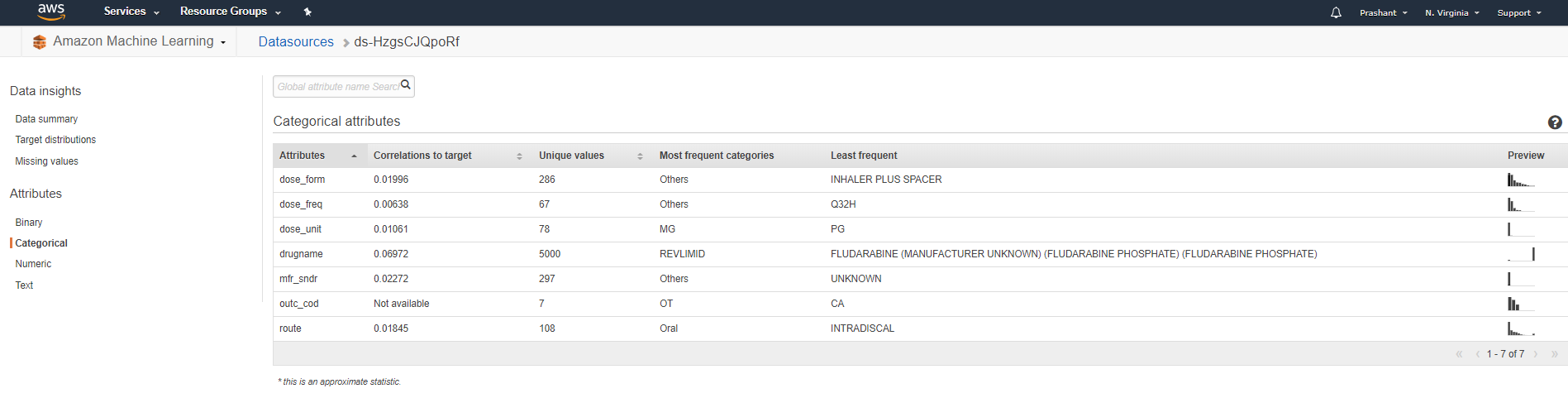
1. Created a data source and evaluated the correlated variables with respect to outcome.



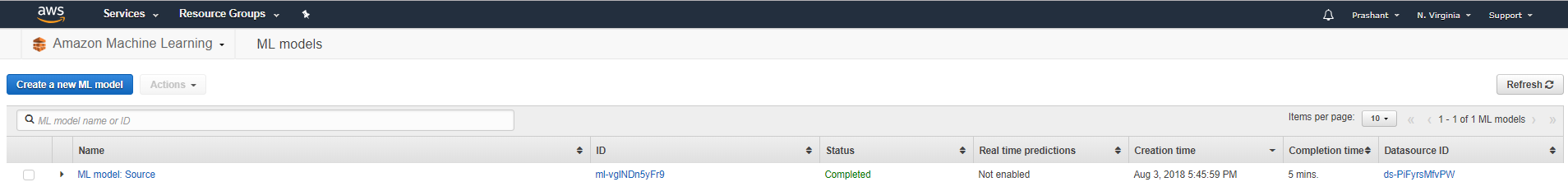
1. Outcome variable distribution



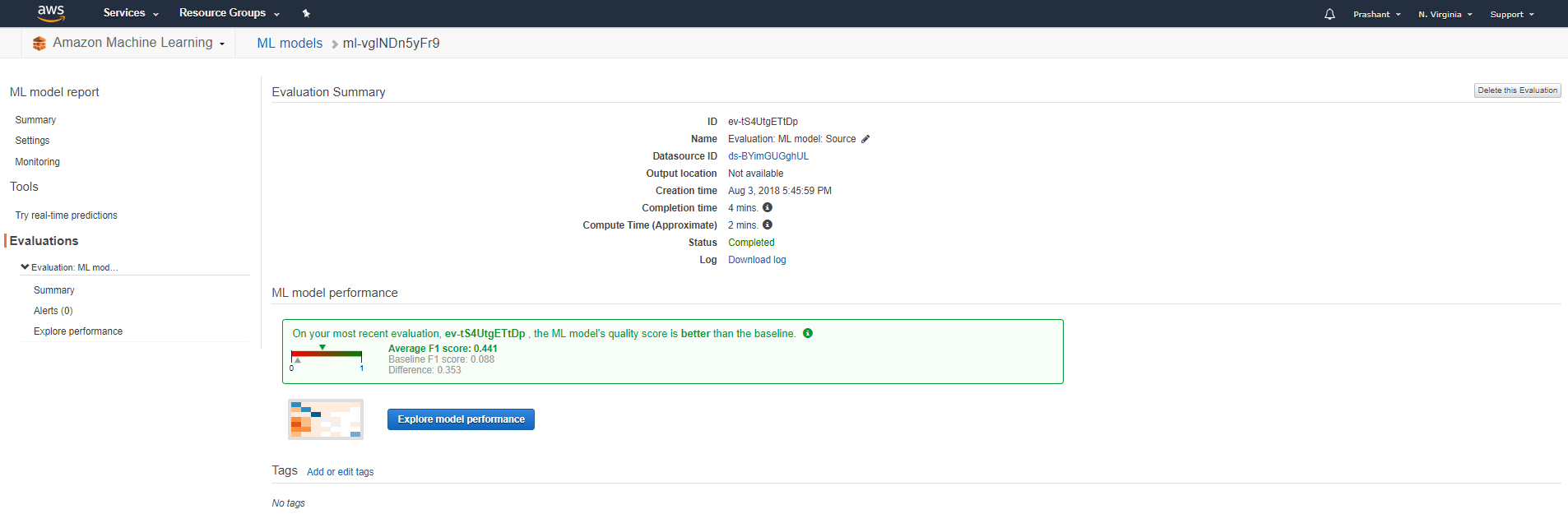
1. Correlation to target



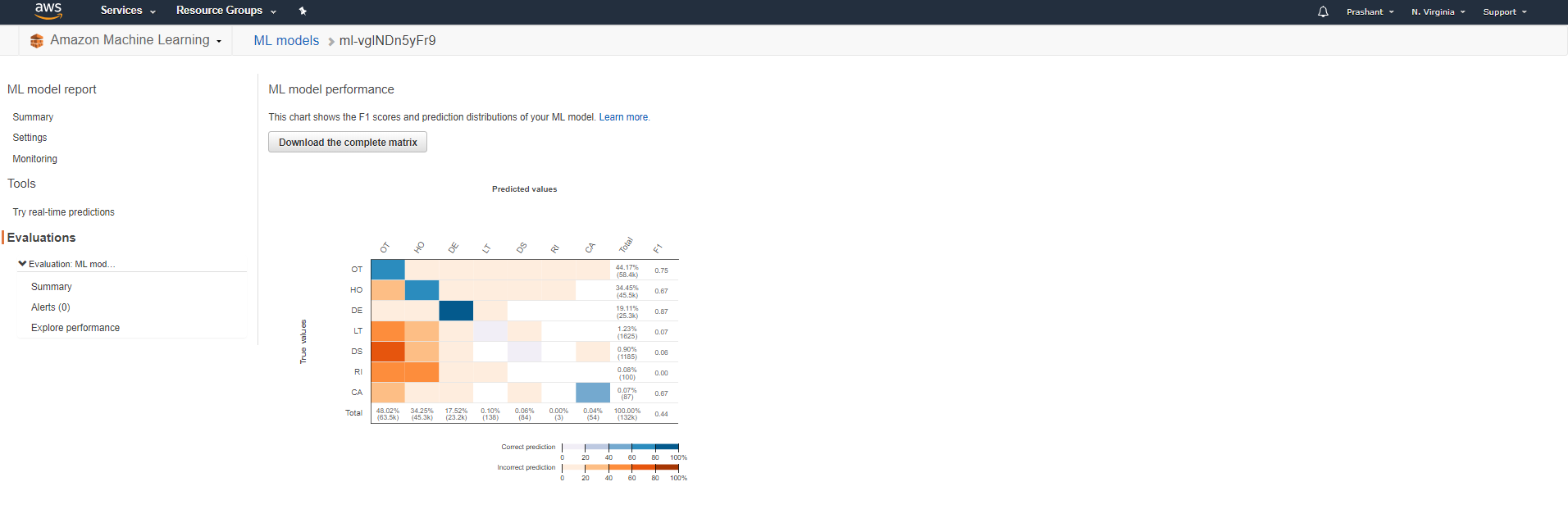
1. Trained a model with **L2-regularization technique** and inbuilt **SGD** learning algorithm.



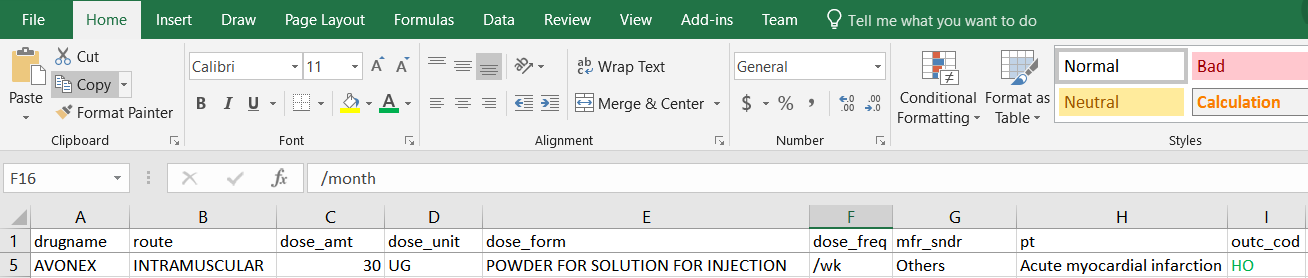
1. Evaluate the model performance.



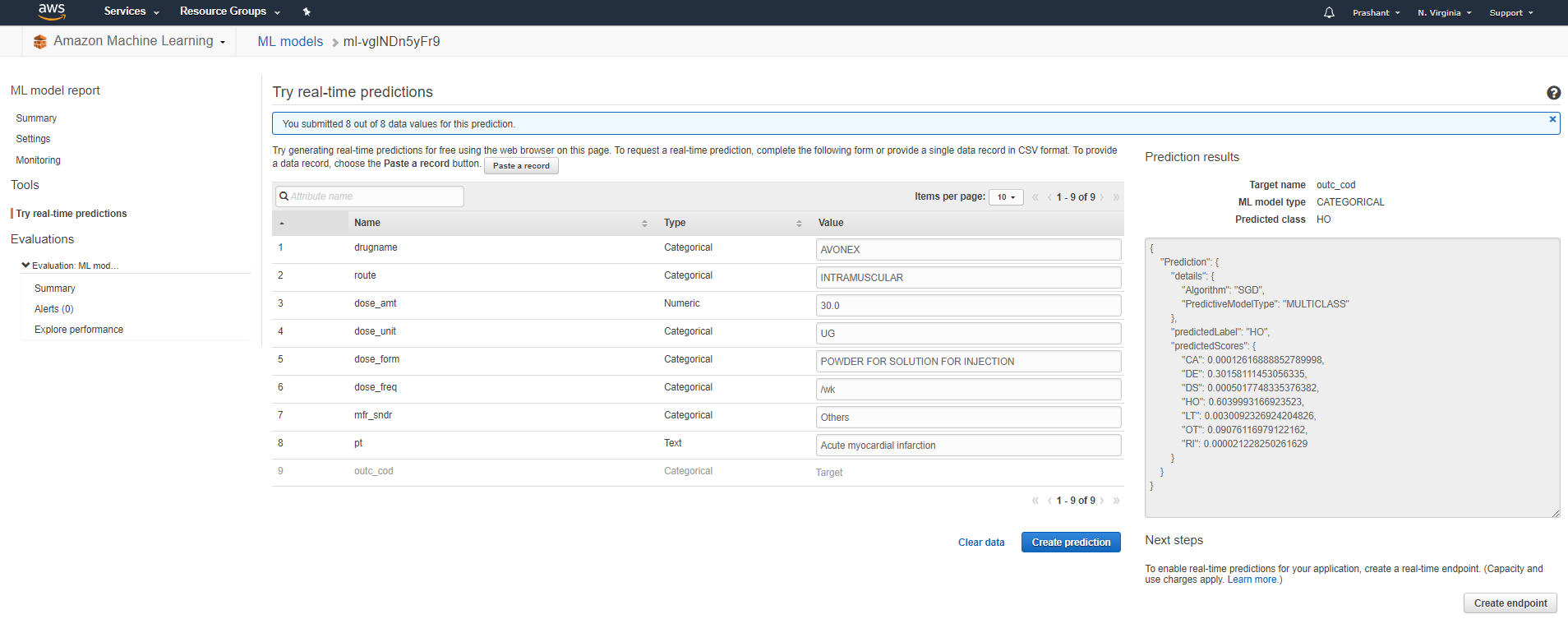
1. Confusion Matrix:



1. Tested our model with one of the existing rows inside input with **OUTCOME – HO (Hospitalization- highlighted)**

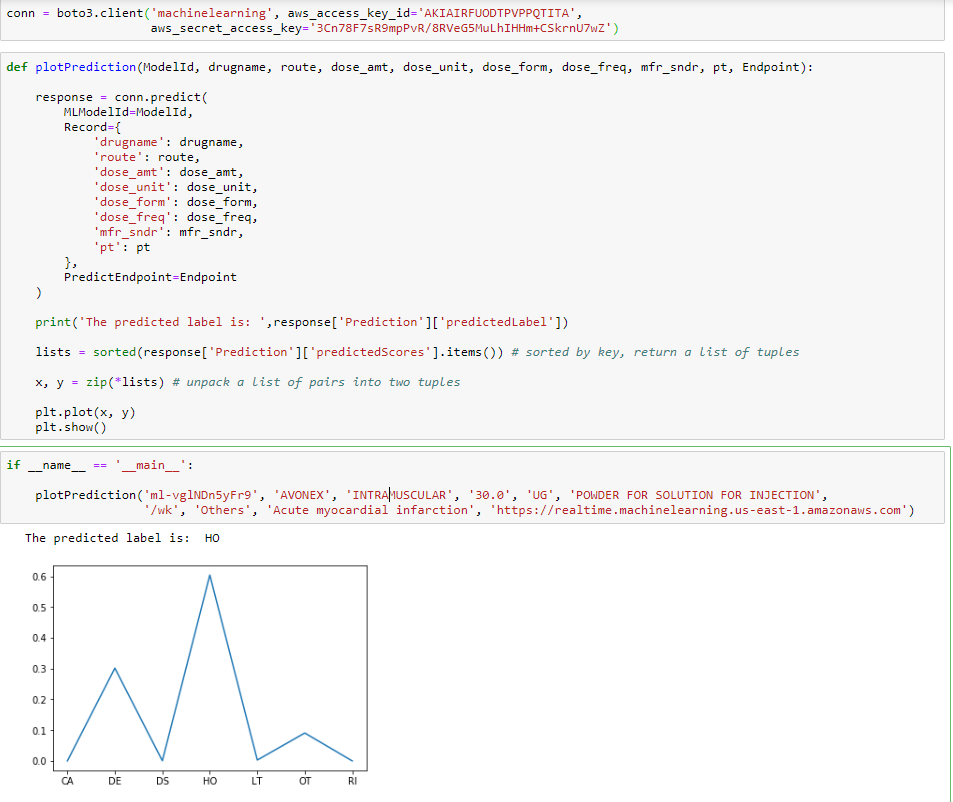


Output from Amazon ML – HO with 60% Scored probabilities



1. Enabled a real-end point URL (API) for the real time predictions.

URL: <https://realtime.machinelearning.us-east-1.amazonaws.com>

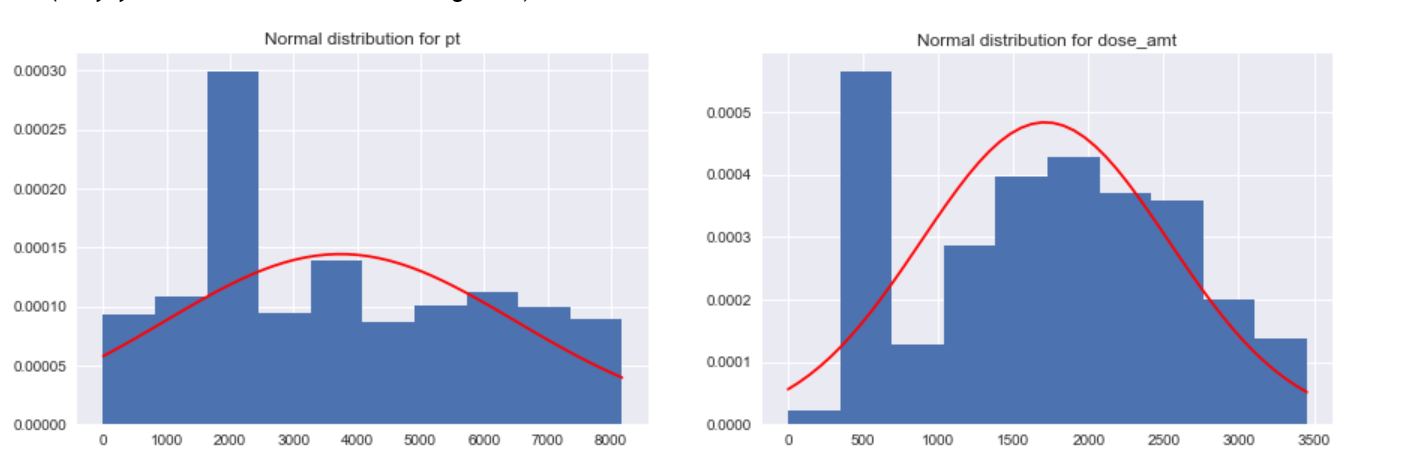


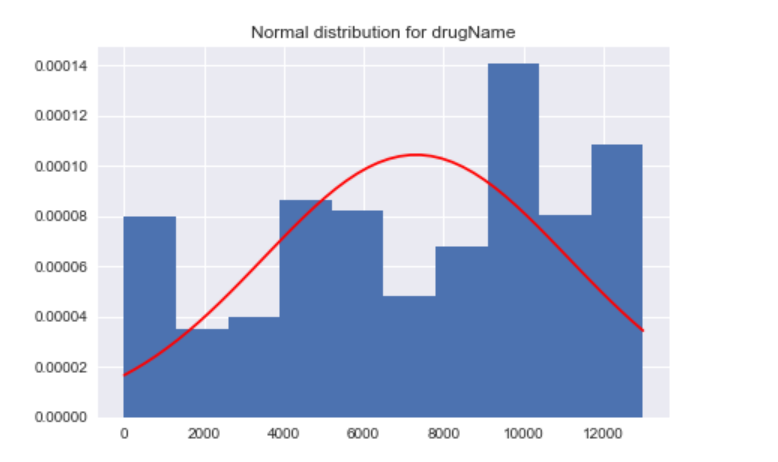
**FEATURE EXPLORATION USING MAXIMUM LIKELYHOOD AND CORRELATION PLOT**

* We used maximum likelihood method to explore the columns and data and to see which columns are follow similar distribution
* Most of columns do not follow similar distribution but these 3 columns seem to follow normal distribution graph.

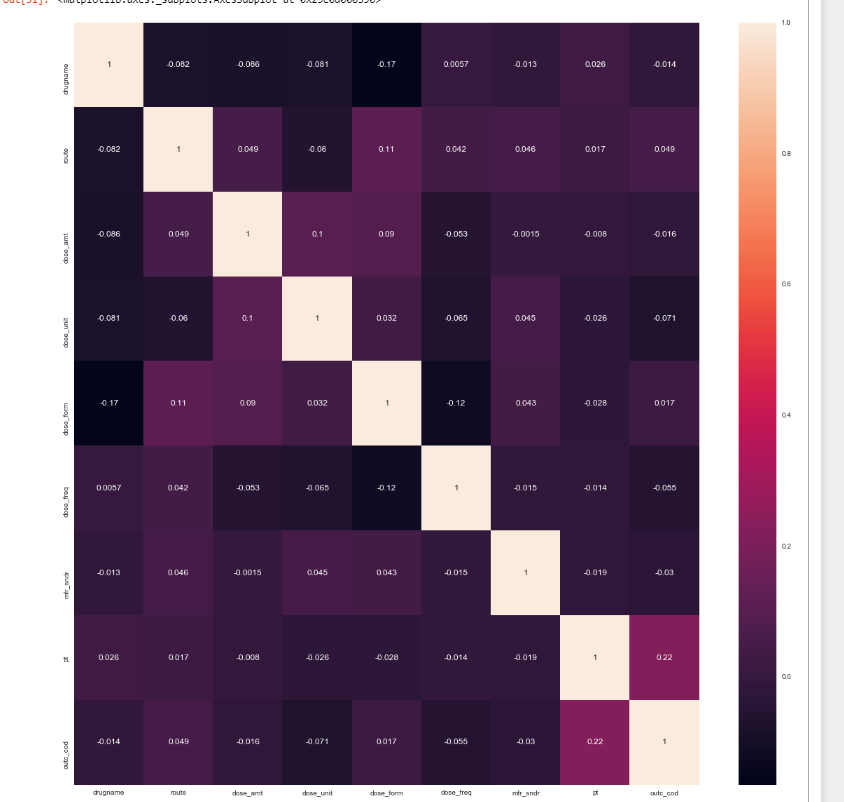
**MLE**

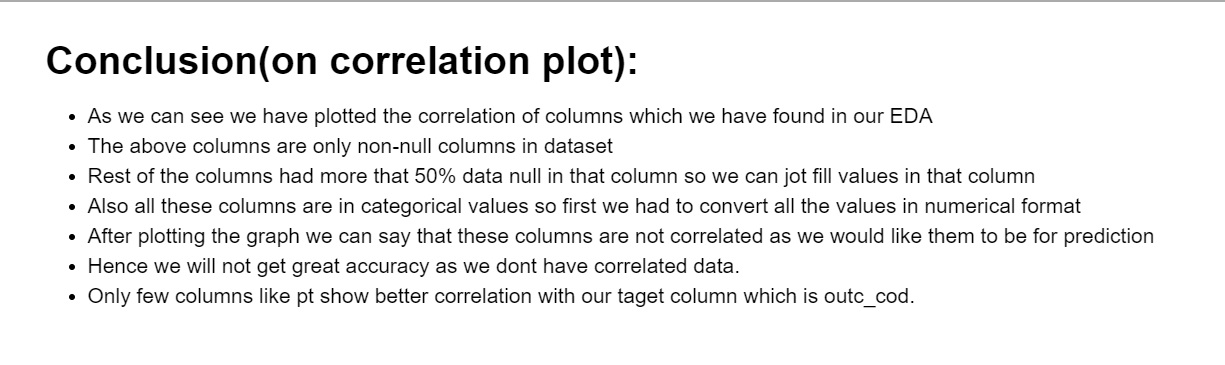






* After this we tried to explore these features using the correlation plot which will give us more idea that which features are more correlated.



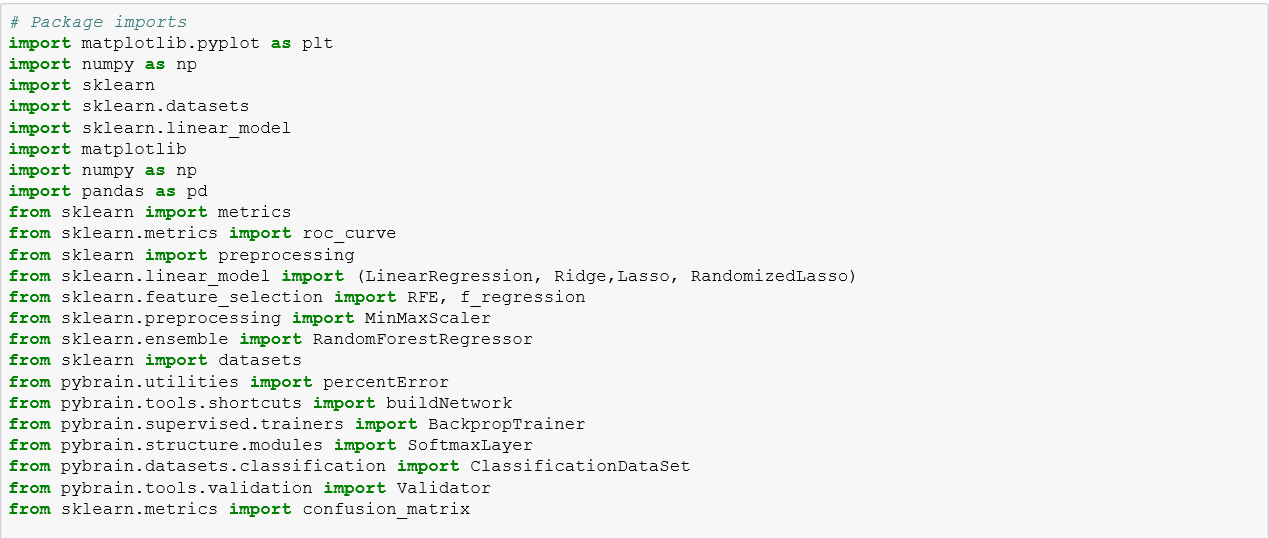


**SKLEARN – IMPEMENTING CLASSIFICATION IN NOTEBOOK**

**SKLEARN – IMPEMENTING CLASSIFICATION IN NOTEBOOK**

We Implement Decision tree classifier, KNeighborsClassifier, Gaussian Naïve Bayes, Multiclass Logestic

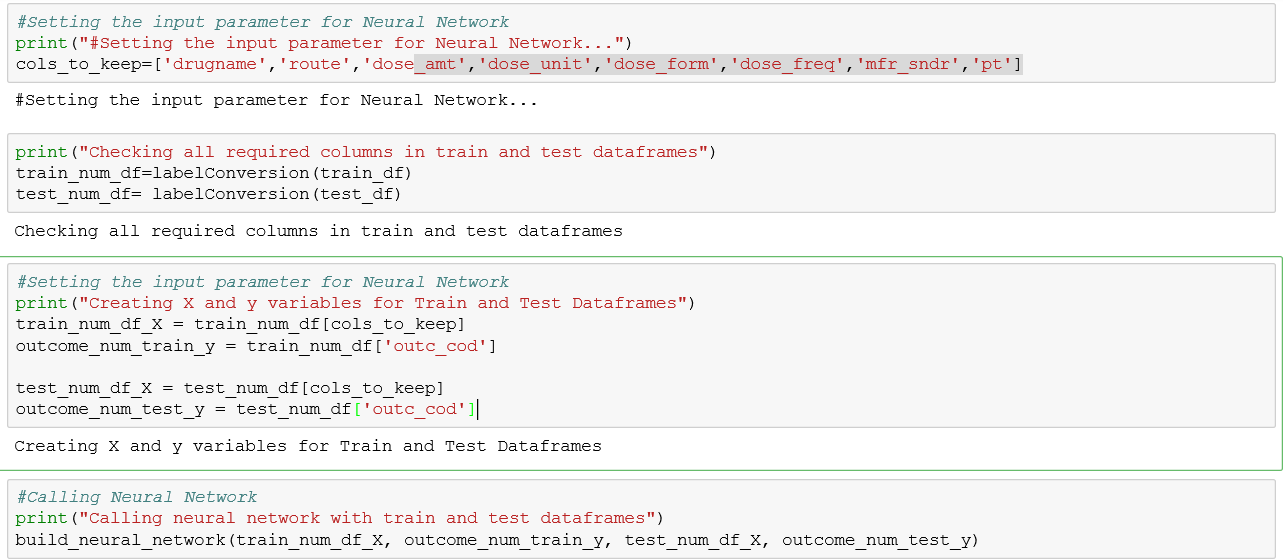
Regression on Jupyter Notebook to the test the accuracy of our model and see the total error. To implement it, we install pybrain and add the following dependency:



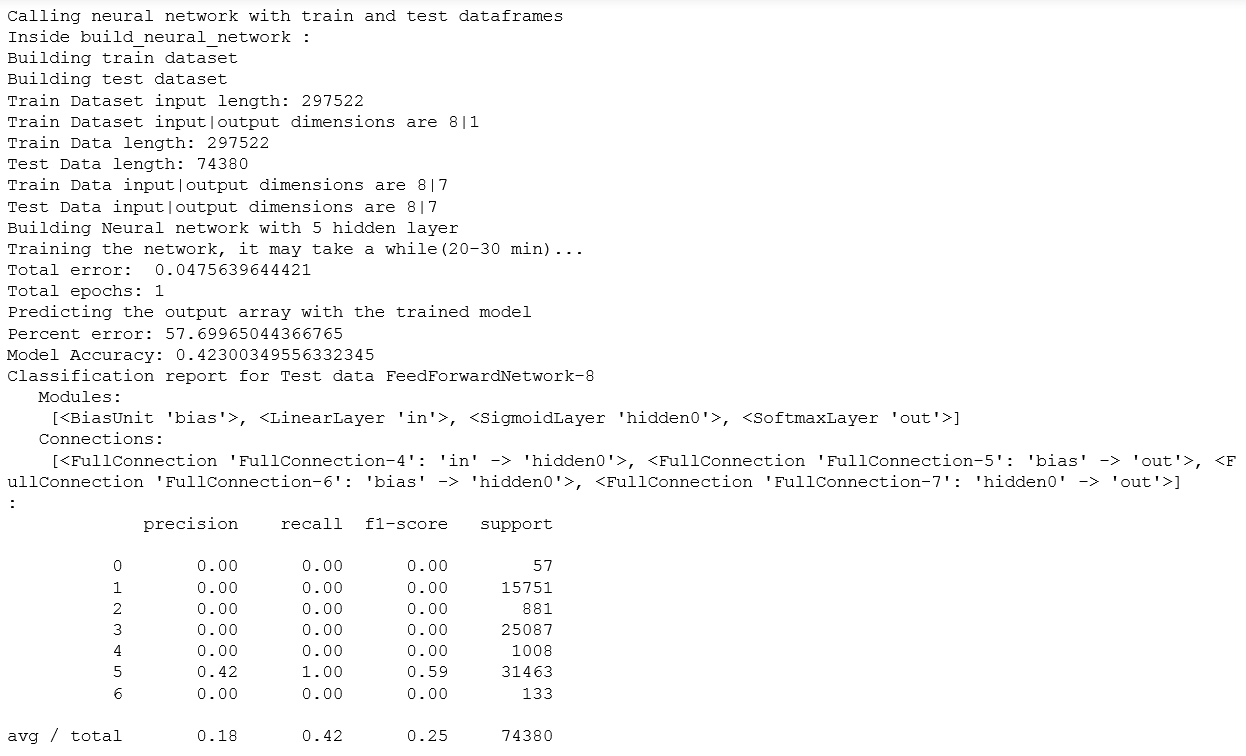
We will then select the best 10 features to train the model and verify result on the test data set to classify our outcome in one of the probable class.



Now we will apply Neural Network Algorithm:

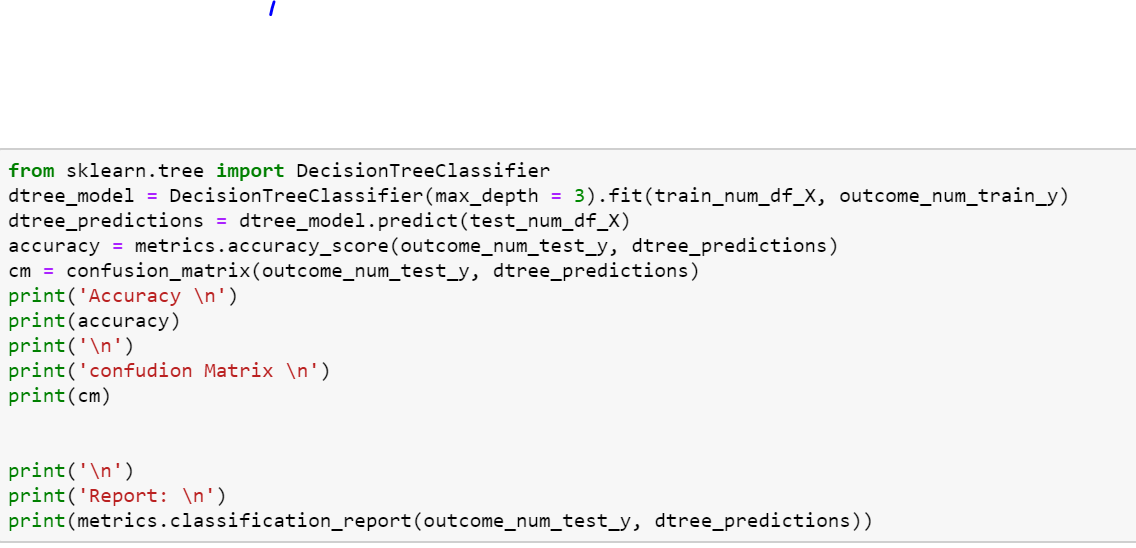


**RESULT:**

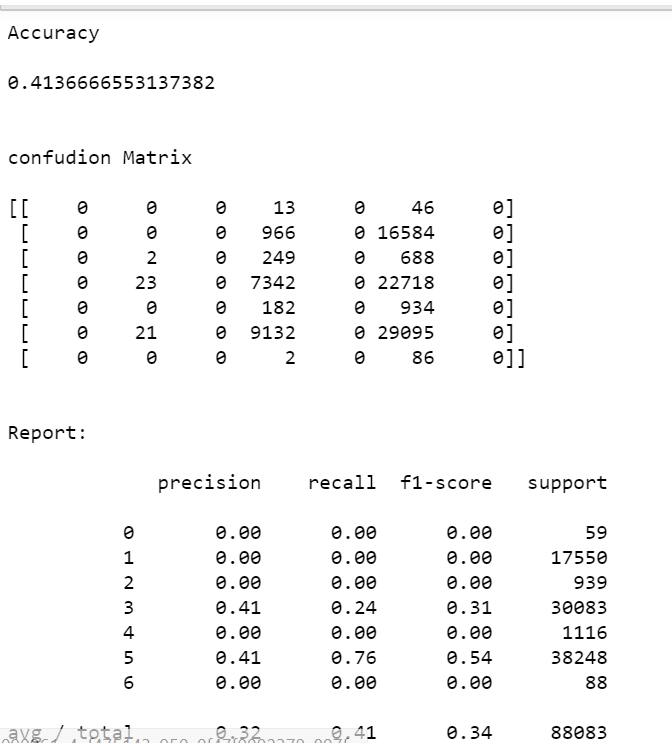


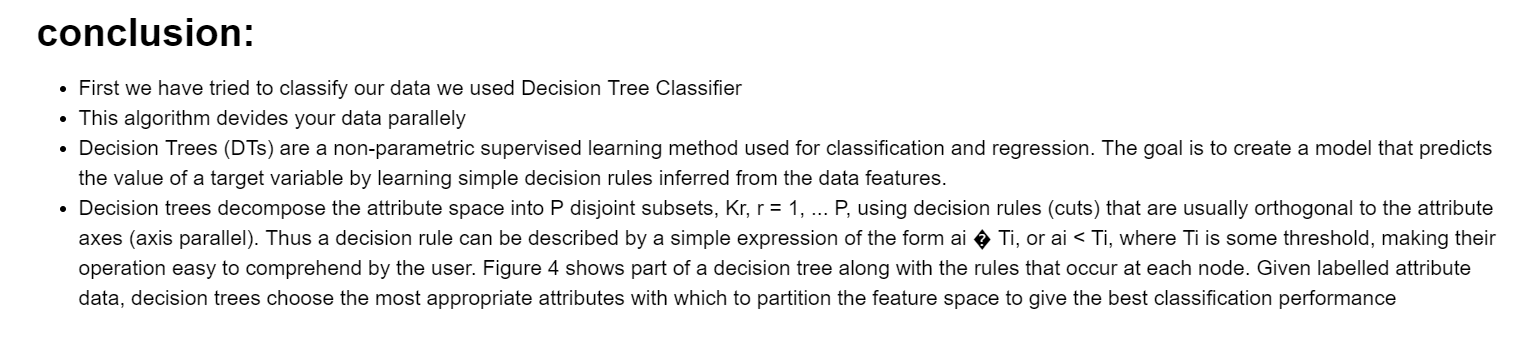
**Decision Tree Classifier:**

* We have tried to implement different classifier algorithms to test the accuracy and difference in speed and results

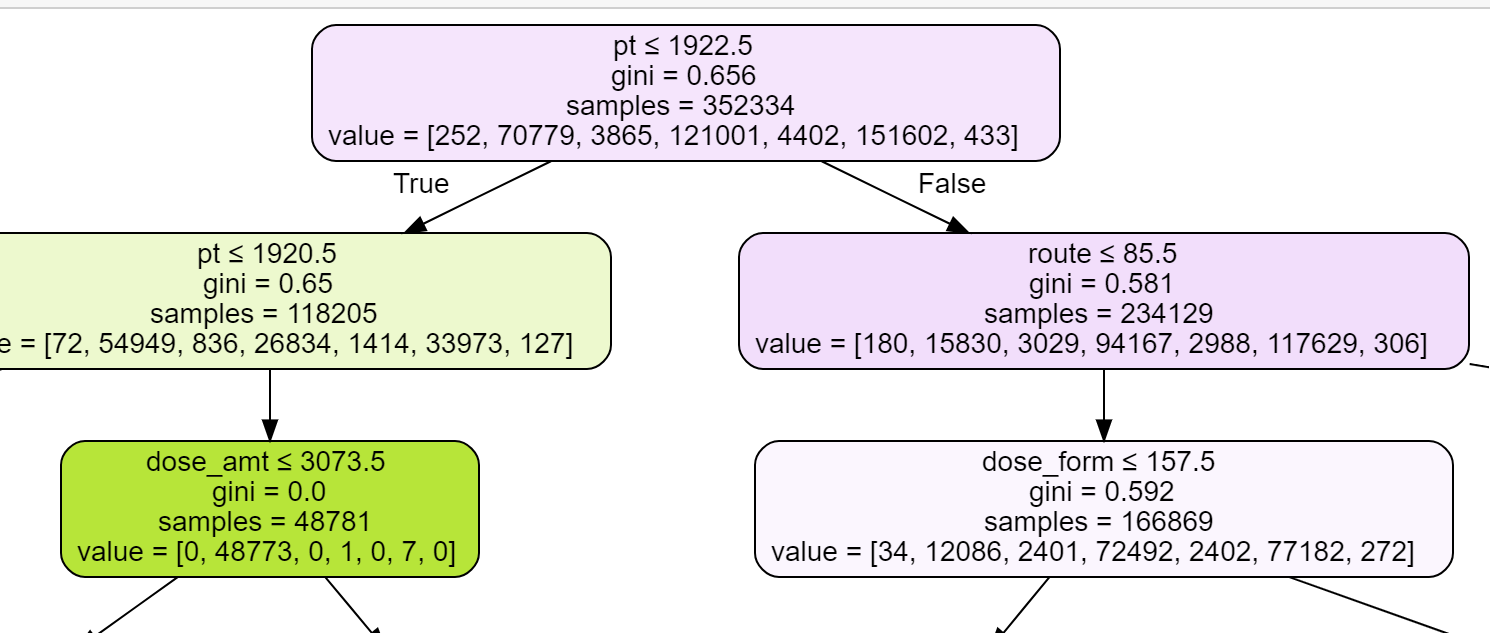


**Results:**

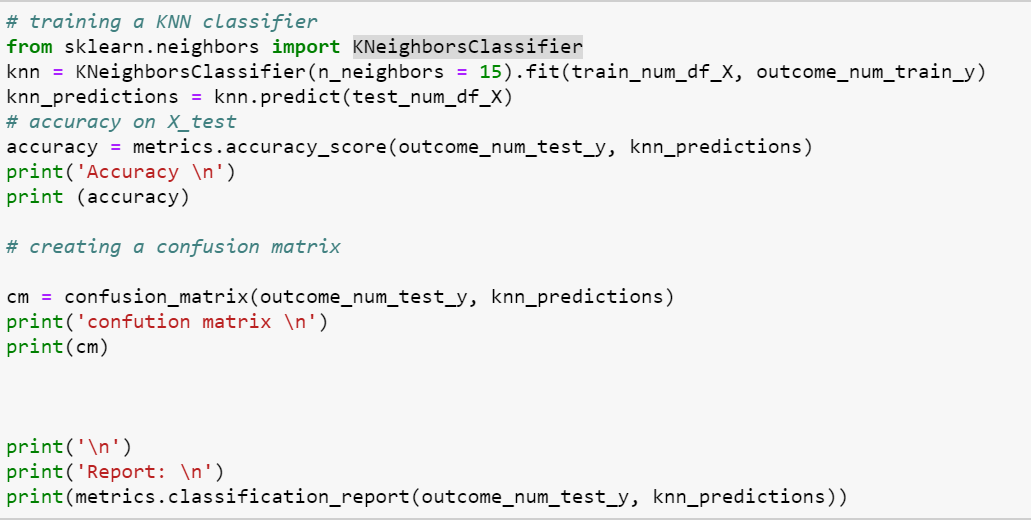




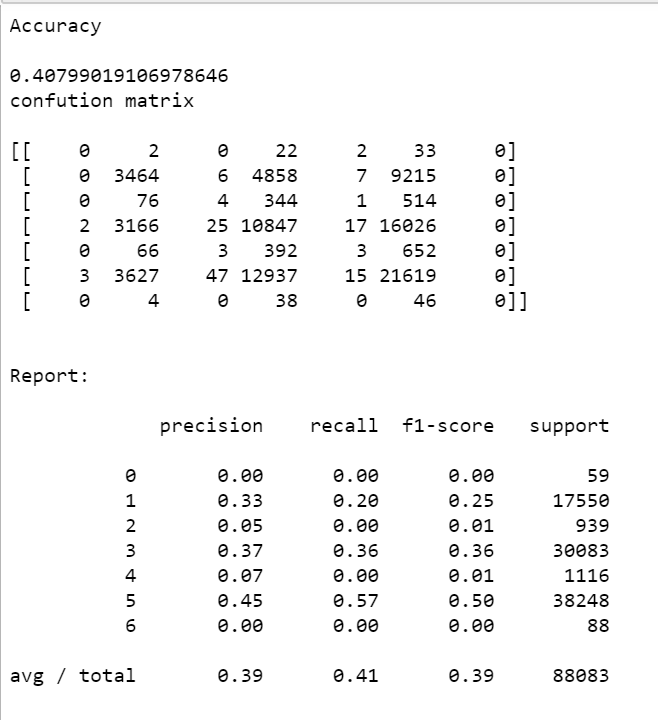
**Tree:**

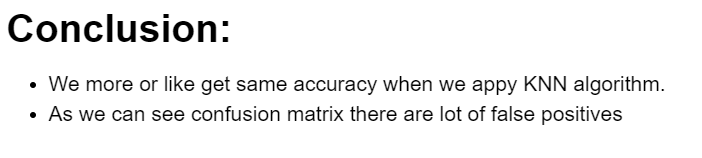


**K- Nearest Neighbors Classifier:**

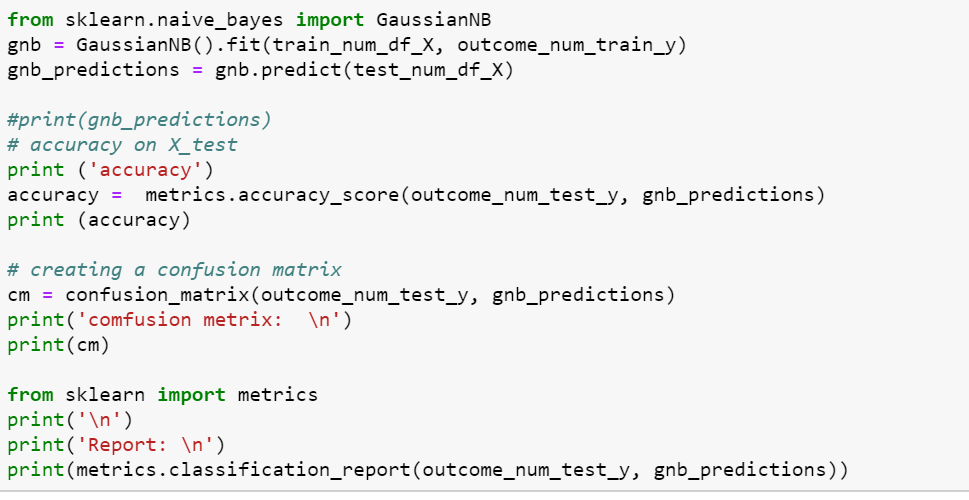


**Results:**

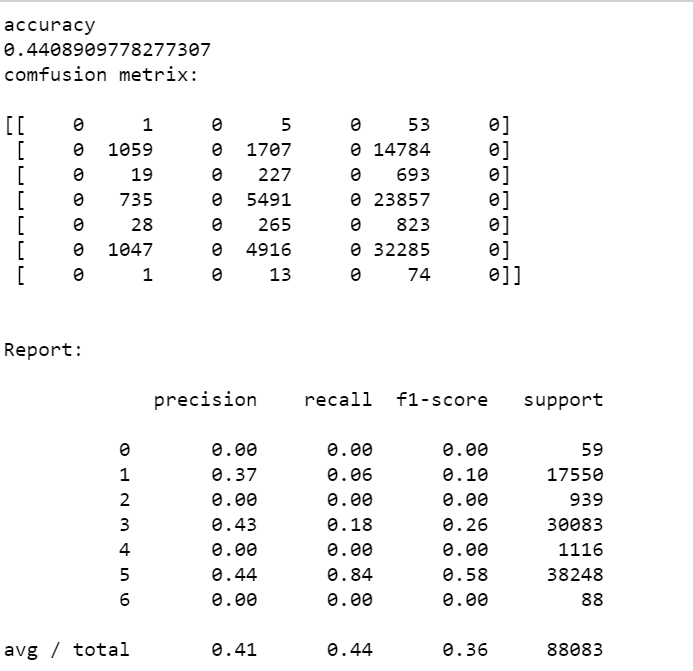


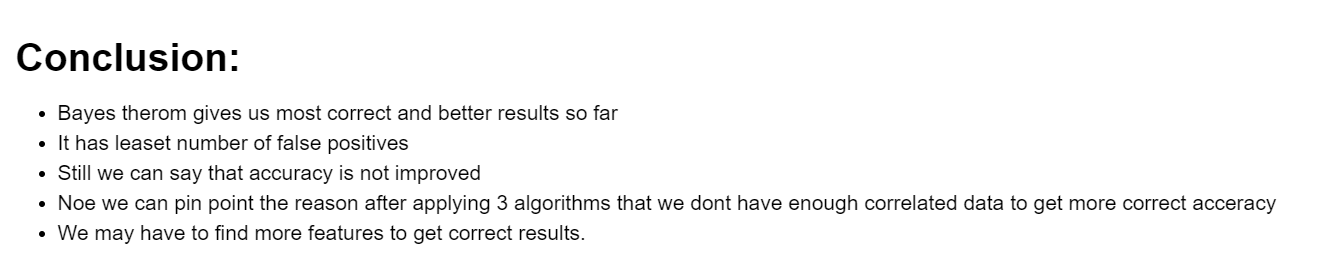


**Gaussian Naïve Bayes :**

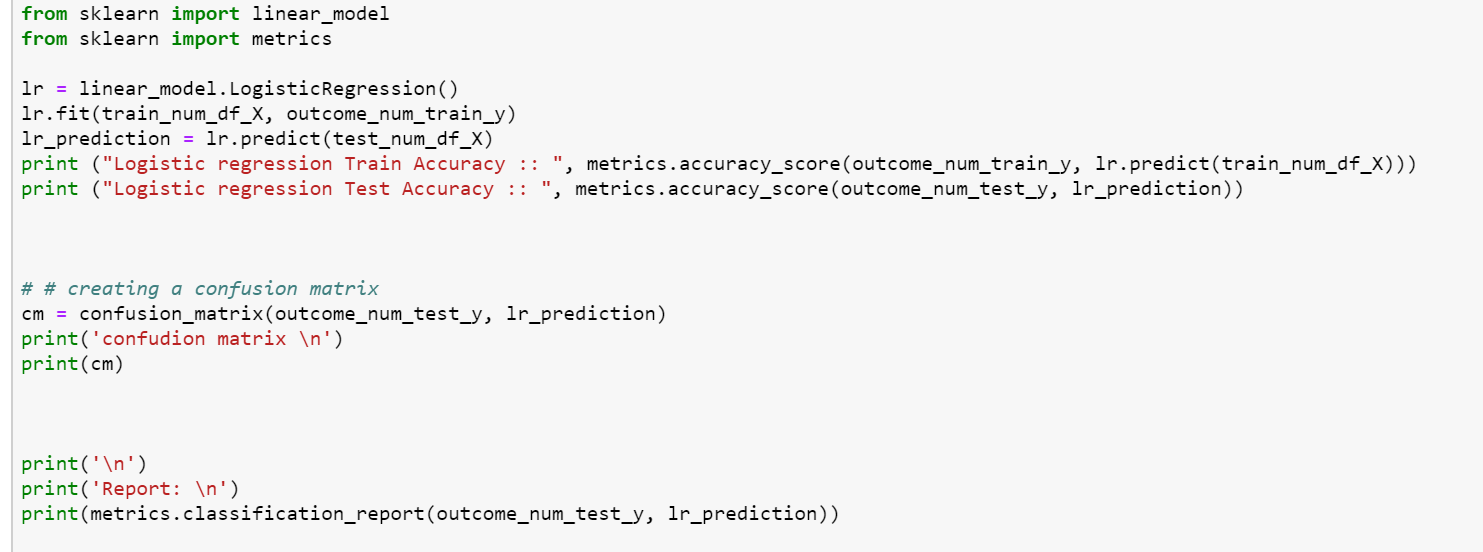


**Results:**

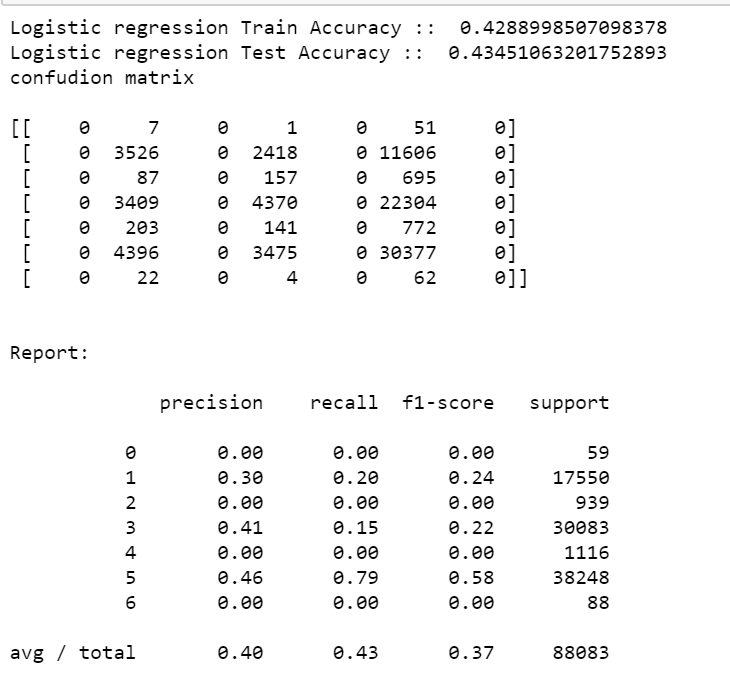




**Logistic Regression Classifier:**



**Results:**



**ROC CURVES FOR EACH CLASS:**

