# NETWORK INTRUSION DETECTION USING

# MACHINE LEARNING ALGORITHMS

# A PROJECT REPORT

*for*

# INFORMATION SECURITY ANALYSIS AND AUDIT

# (CSE3501)

*in*

# B. Tech (Information Technology)

*by*

# JAHNAVI MISHRA

# (18BIT0243)

# Fall semester, 2020

*Under the Guidance of*

# Prof. SUMAIYA THASEEN I

Associate Professor Grade 1, SITE



# School of Information Technology and Engineering

* **Design:**



* **Description:**
* **About the Dataset:**

In **CICDarknet2020** dataset, a two-layered approach is used to generate benign and darknet traffic at the first layer. The darknet traffic constitutes Audio-Stream, Browsing, Chat, Email, P2P, Transfer, Video-Stream and VOIP which is generated at the second layer. To generate the representative dataset, previously generated datasets, namely, [ISCXTor2016](https://www.unb.ca/cic/datasets/tor.html) and [ISCXVPN2016](https://www.unb.ca/cic/datasets/vpn.html), have been amalgamated and respective VPN and Tor traffic are combined in corresponding Darknet categories.

**No. of rows:** 1.4 Lacs approx..

**No. of columns:** 85

* First we will be importing all the python libraries and sklearn modules.
* **Importing the dataset:**

The dataset DARKNET 2020 can be downloaded from [***https://www.unb.ca/cic/datasets/darknet2020.html***](https://www.unb.ca/cic/datasets/darknet2020.html)

* **Data Cleaning Phase:**

In this phase:

* We will be handling the missing values and replacing them with the mean value of the respective column.
* Also the null and NaN values will also be replaced by the mean of the column.
* The columns also contain infinite values. To deal with such values, first we will find the row index of such values, then convert them into NaN and delete the entire row.
* **Data Pre-Processing Phase:**

**For binary classification:**

* The dataset is minimized by only including rows that have either benign or malicious classes i.e. Tor and Non-tor classes.
* This dataset is then balanced, that is, the number of rows of each class is made approximately equal by either over sampling or under sampling.
* Target variable is then separated from the dataframe and a new dataframe is created for the target variable. This target variable is then encode using Label Encoder to convert into integer from object.
* The dataset is then normalized using min-max scalar.
* The number of columns is large and we will not be needing all of them for classification. So we will be performing feature extraction using ranking technique to extract only the relevant columns and proceed further with only those columns.
* There are few columns like timestamp and flow-id that can be removed directly. Also columns like Src IP and Dst IP can be modified to be used further instead of deletion.

**For multi-class classification:**

The complete dataset is used in this case. Rest of the procedure remains same as the binary classification.

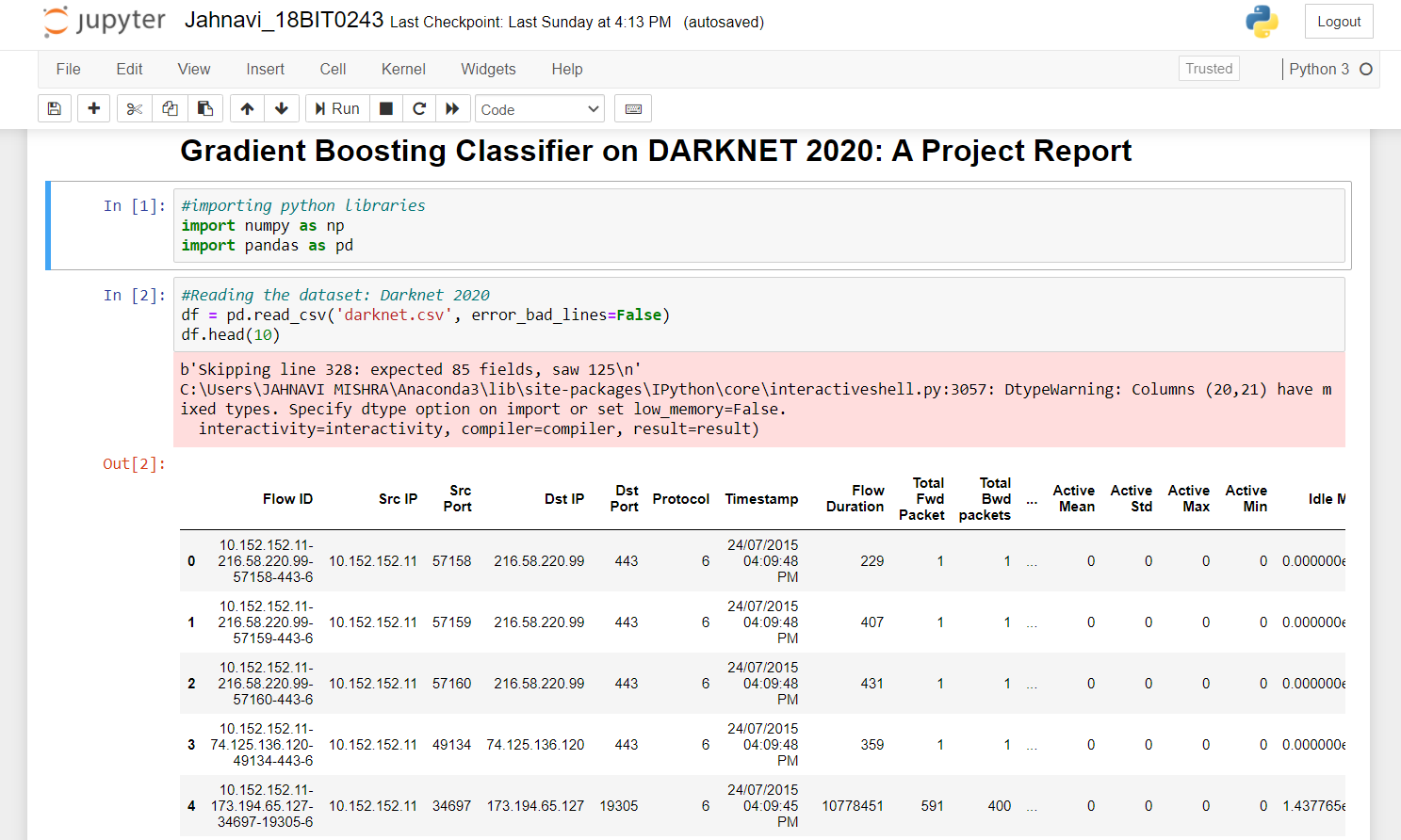
* **Model Training and Testing Phase:**

In this phase:

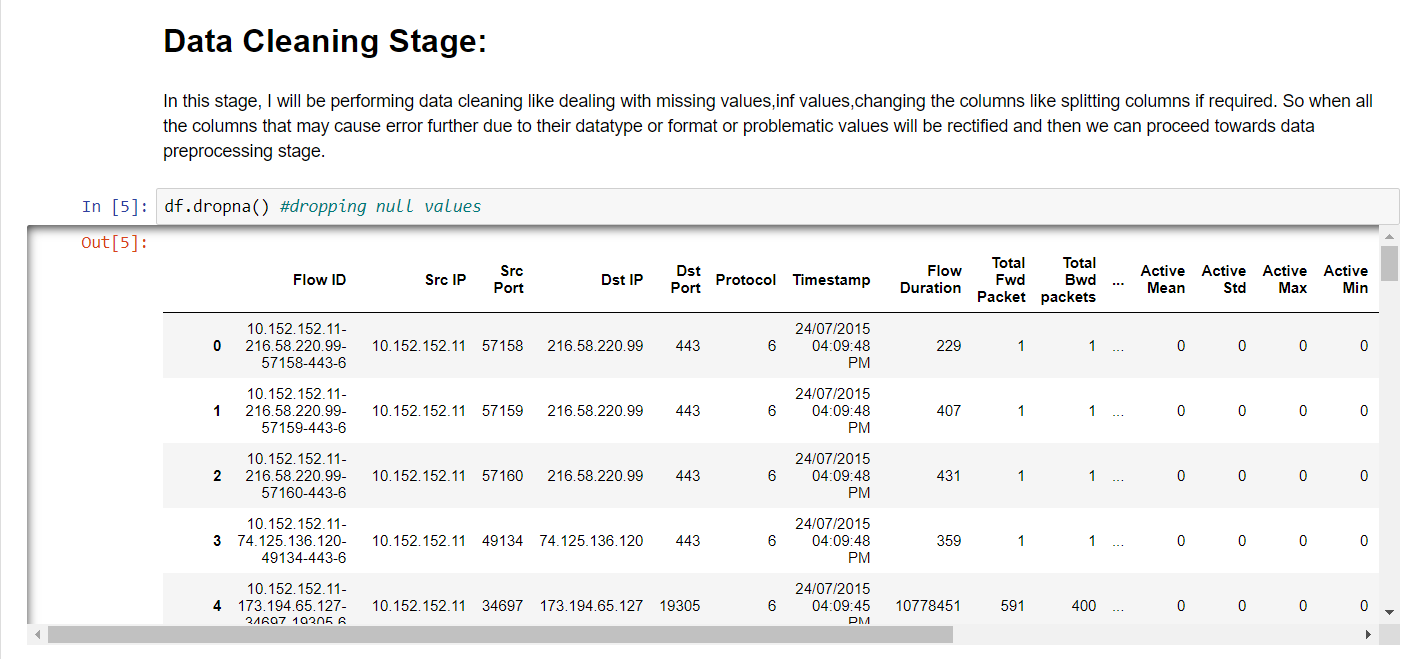
* First the dataset is split into training and testing dataset.
* Then we will be defining two classifiers: **Gradient Boosting Machine** and **AdaBoost**.
* Initially we will be passing random parameters and perform a baseline prediction. Classification Report is generated and accuracy is obtained. Confusion matrix is also obtained.
* Then we will be performing 10-Fold Cross Validation. After this we will be performing Hyperparameter tuning on both the models and tune three parameters each.

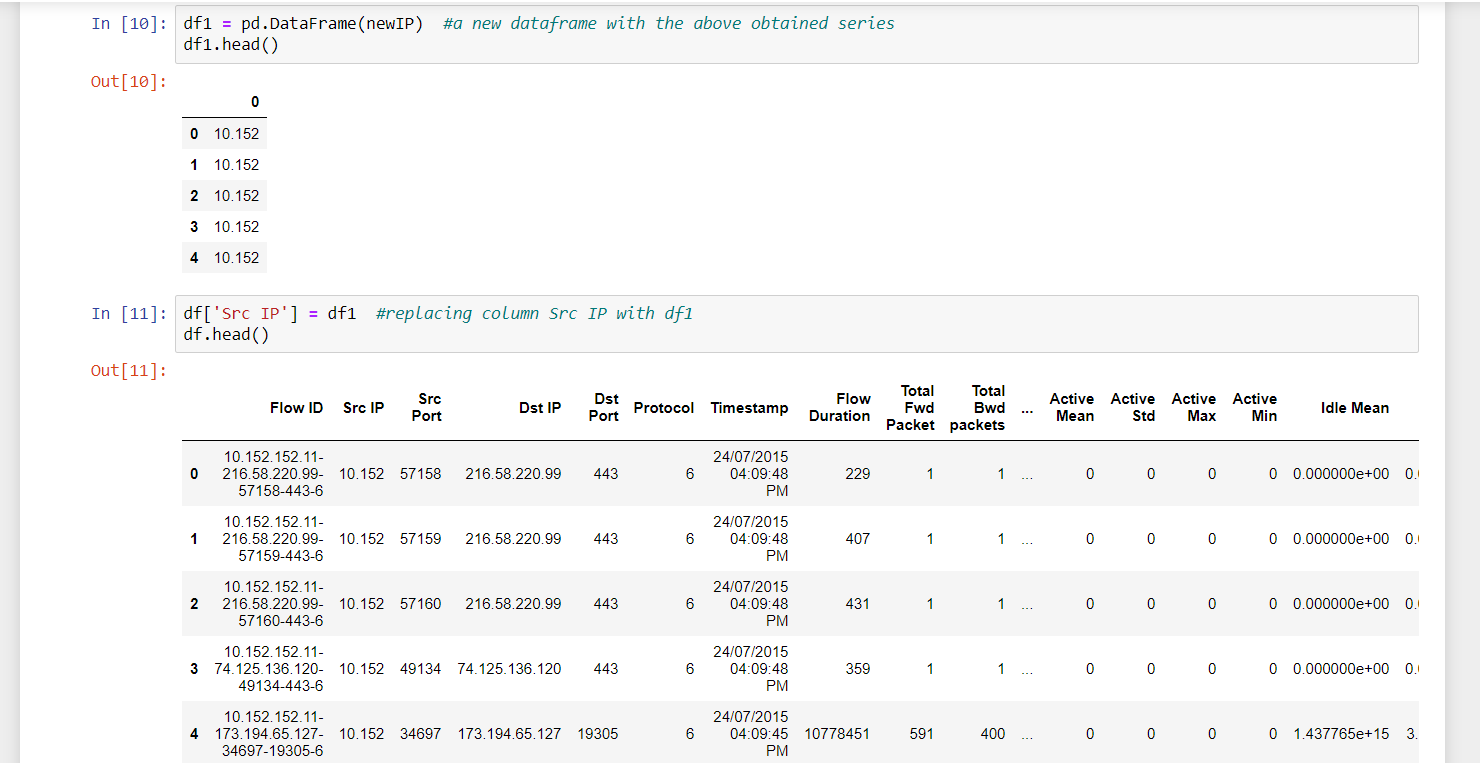
**The three best parameters will be obtained.**

* We will again perform prediction on the test dataset using a new model and passing the three best parameters. Classification report is generated again. Accuracy is obtained.
* The evaluation metrics of this new tuned model is then compared with the baseline model.
* Graphs like roc\_auc curves for binary classification and heatmaps are plotted wherever required.
* Also comparison is done between both the classifiers to get the more efficient one.
* **Code And Output:**
* **Importing Dataset:**

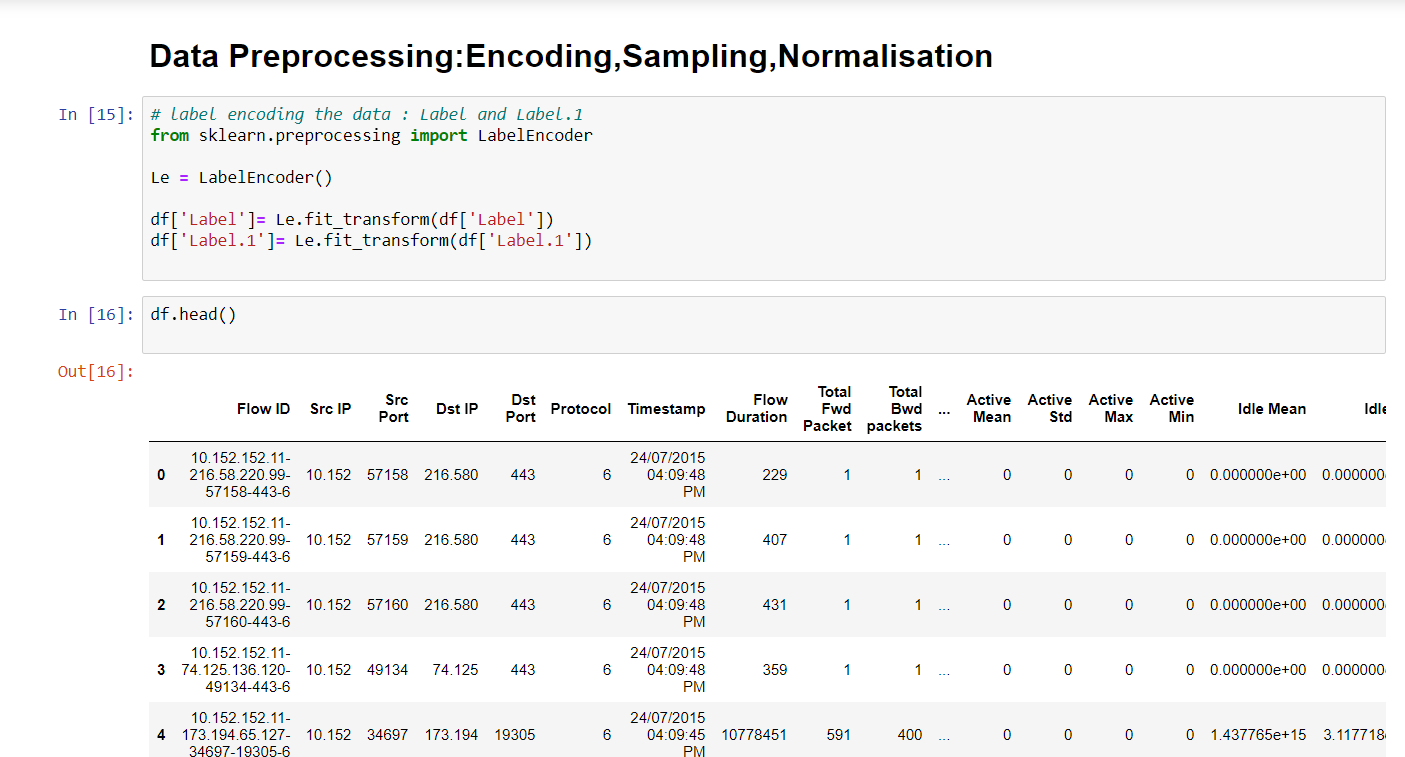


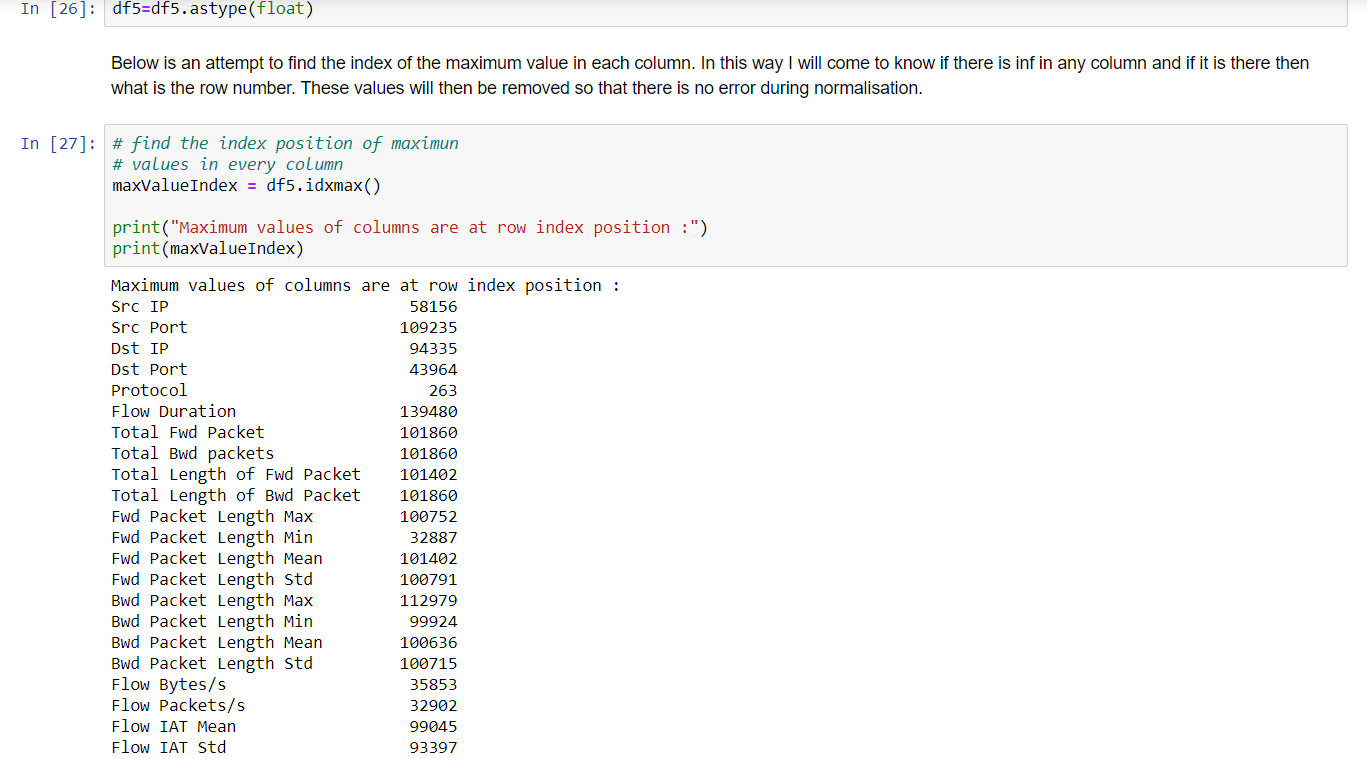
* **Data Cleaning:**

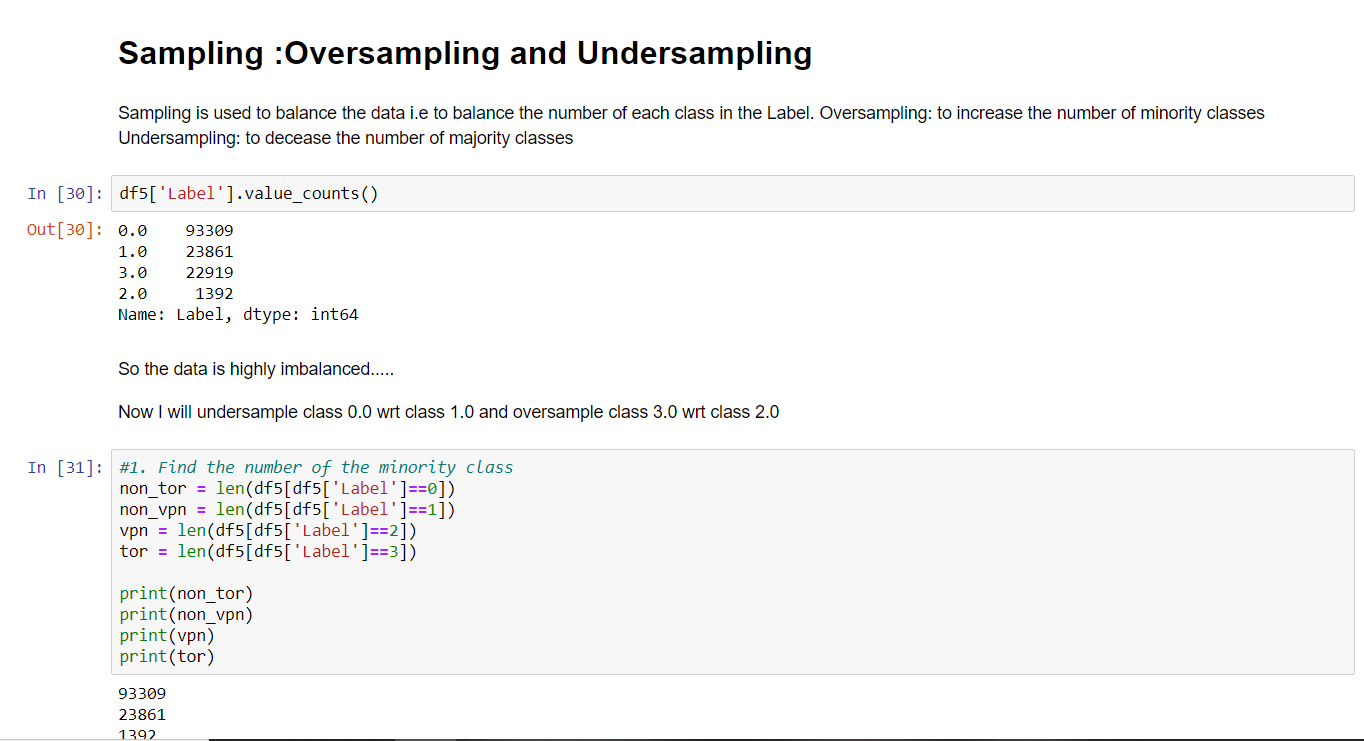




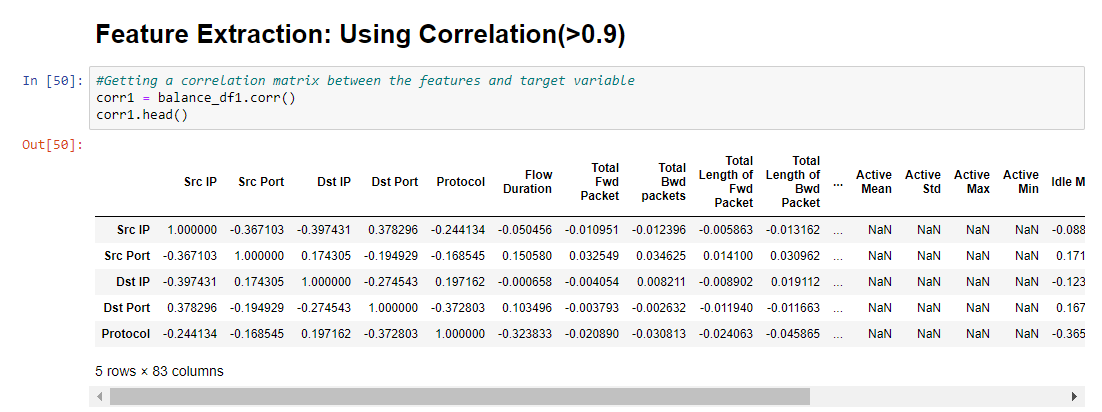
* **Data Pre-Processing:**

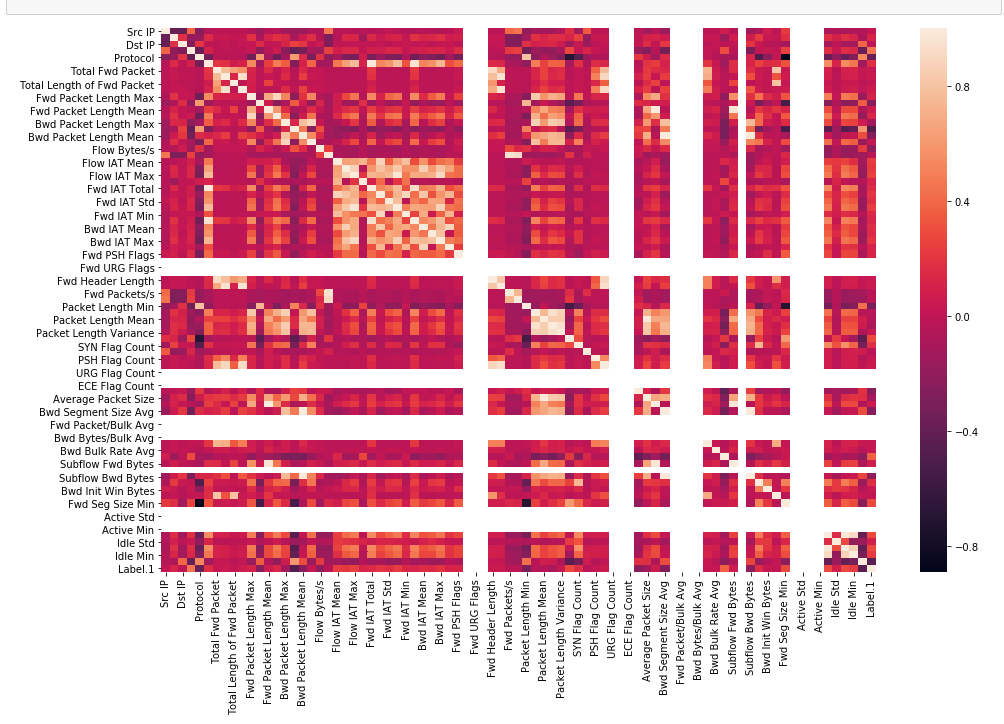






* **Feature Extraction:**





* **Model Training and Testing Phase:**

