**ABSTRACT**

A Darknet is an overlay network within the Internet, and packets' traffic originating from it is usually termed as suspicious. In this project, common machine learning classification algorithms are used to identify Darknet traffic in IoT applications like smart home, inventory management, etc which have high chances of malicious activity and intrusion. The experiments are conducted on the new dataset CIC-Darknet2020 and the classifiers were trained to both binary and multiclass classification. In the first classification task, there were two classes: "Benign" and "Darknet", whereas in the second there were four classes: "Tor", "Non Tor", "VPN" and "Non VPN".

In this proposed work, ML algorithms like Decision Tree, Knn, Random forest classifier and ensemble learning models like GBM and LightGBM are used to classify the data into 8 different constituents of Darknet as mentioned above on the basis of parameters like source and destination IP, Flow ID, Forward and backward packet information etc. Then a novel ensemble model is proposed combining different algorithms using voting technique. Feature extraction is used to get the most useful and correlated parameters out of the 85 parameters. Baseline classification is done on each classifier and the proposed model followed by 10 fold cross validation and Hyperparameter tuning for getting the best parameters to enhance the performance. The proposed model is compared with all the classifiers and the existing state of the art models, in terms of evaluation measures like accuracy, precision, recall and F1 score.

**CONTENTS**

|  |  |
| --- | --- |
| **CONTENTS** | **Page** |
|  | **No.** |
|  |  |
| **Table of Contents** | 3 |
| [**List of Figures**](#_bookmark1)  **List of Tables/Charts** | [4](#_bookmark1)  4 |
|  |  |
| 1 [**INTRODUCTION**](#_bookmark2) | [5](#_bookmark2) |
| 2 **LITERATURE SURVEY** | 7 |
| 3 **HARDWARE AND SOFTWARE REQUIREMENTS** | 13 |
| 4 **DETAILED DESIGN** | **14** |
| 4.1 System Architecture | 14 |
| 4.2 UML Diagram | 15 |
| 4.3 Module Description | 16 |
| 4.4 Data Flow Diagram | 20 |
| 5 **IMPLEMENTATION** | **21** |
| 6 **PERFORMANCE METRICS**  7 **CONCLUSION AND FUTURE SCOPE**  8 **REFERENCES** | **38**  **44**  **45** |

# **List of Figures:**

|  |  |  |
| --- | --- | --- |
| **Fig No.** | **Title** | **Page No.** |
| 1 | Architecture Diagram | 17 |
| 2 | Use Case Diagram | 18 |
| 3 | Darknet Traffic Details | 19 |
| 4 | Dataset Description | 20 |
| 5 | Data Flow Diagram | 23 |
| 6 | Data before and after normalisation and scaling | 32 |
| 7 | Min-Max Scaler Formula | 32 |
| 8 | Hard Voting | 35 |
| 9 | Soft Voting | 35 |
| 10 | Comparison with existing State of Art Models | 43 |

**List Of Tables/Charts:**

|  |  |  |
| --- | --- | --- |
| **Table No./Graph No.** | **Title** | **Page No.** |
| 1 | Performance Comparison (Table) | 45 |
| 2 | Performance Comparison (Chart) | 45 |

|  |
| --- |
|  |
|  |
|  |

# **INTRODUCTION**

Darknet is a series of private websites that can only be accessed through encrypted channels. Unlike our regular online search services that can easily be tracked by governments, darknet can not be tracked easily by authorities.

Today, the darknet has become a virtual platform for the sale and purchase of many illegal drugs, such as drugs, pornography, and other illegal activities. On a dark web, people can search the Internet without leaving their online trace and without fear of being caught. However, if you are caught by the authorities for engaging in illegal activities in this secret forum, you could go to jail. With its quality of anonymous coverage, the stadium has become a safe haven for criminals operating in all forms of illegal activity. It has also become a source of concern for the government as the terrorists might use the platform.

Intrusion Detection System is a system used to monitor the network activities and alerts if any malicious or suspicious activity is detected to prevent further damage.

**The two types of widely used Intrusion Detection system are:** Anomaly based and Signature based.

**Signature-based:** Signature-based IDS detects potential threats by looking at specific patterns, such as byte sequencing in network traffic, or known malicious command sequences used by a malware. The wording comes from the antivirus application, which refers to these patterns obtained as signatures. Although IDS-based IDS can easily detect known attacks, it is not possible to detect new attacks, which is not your pattern available.

Drawbacks of Signature Based:

Inability to detect new or previously unknown attacks. If no signature exists to match an attack type, the new attack will go undetected. Therefore, keeping your signature database updated is important.

* Unable to perceive novel occurrences
* Suffer from false panics
* They have to be automated again for every new pattern to be perceived.

**Anomaly based:** New technologies designed to detect and familiarize yourself with anonymous attacks, mainly due to the explosion of malware. This acquisition method uses machine learning to create a defined model for reliable performance, and then compares new behaviours against this reliability model. Although this method allows for the detection of previously unknown attacks, it can have negative consequences: previously unknown legal activity can be accidentally classified as malicious.

**The transition from Signature-based to Anomaly based:**

Anomaly based NIDS can robotically infer attacks which are yet unknown, and therefore undetectable through signature based NIDS. An anomaly detection approach normally includes extraordinary steps: step one is known as schooling or training phase where in a normal visitors profile is generated; the second phase is referred to as anomaly detection, in which the learned profile is carried out to the modern site visitors to look for any deviations. A number of anomaly detection mechanisms has been proposed recently to hit upon such deviations, which can be labelled into statistical methods, statistics-mining methods and device mastering based strategies.

# **LITERATURE SURVEY**

## Tama, B. A., & Rhee, K. H. (2019) [1] proposed an improved detection performance of anomaly-based intrusion detection system (IDS) using gradient boosted machine (GBM). One of the main problems in the tree learning is to find the best split. To solve this, exact greedy algorithm is employed. Moreover, since GBM requires several tuned hyper-parameters, it is necessary not to use the same parameters on different datasets. Thus, a grid search is used to find the best parameters for each dataset. The difference of GBM over other classifiers, such as random forest, DNN, SVM, and CART is statistically assessed. The performance of GBM is compared with RF, DNN, SVM, CART. GBM is best performer in terms of specificity and AUC value.

## KHAFAJEH, H. (2020) [2] presented a comparative study of AdaBoost, GBM, Random Forest, Extra Trees, and Logistic Regression, KNN, Perceptron with LightGBM based on classification performance and computational cost. This model is selected due to, high-performance level, being accessible, various selections and fast implementation of hyper parameters. Recently, the anomaly-based NIDS utilizes machine learning methods has received much attention. It employs a one-sides-sampling (GOSS) approach to detect split value in data instances. Then LightGBM is compared with other classifiers and superiority of the model is justified. The accuracy, DR and AUC obtained by proposed algorithm is 0.983, 0.974, 0.981 respectively which clearly shows that it outperforms all other algorithms and techniques.

## Meftah, S., Rachidi, T., & Assem, N. (2019) [3] performed Logistic Regression, GBM and SVM as a part of a two step NID process. A major reason for using these techniques is their ability to improve their own detection performance by learning the patterns from network traffic on the fly, thus being able to detect unknown attacks.A hybrid (two-stage) classification-based NIDS is proposed that tests the efficiency, effectiveness and robustness of that dataset. In first stage classification, the comparison was made between LR, GBM, SVM. Results show that the use of SVM reports the highest accuracy (82.11%).The output of SVM is the fed to multi classifiers and a comparison is made. Specifically, the performance of Decision Trees (C5.0), Naïve Bayes and multinomial SVM is evaluated. Applying C5.0 gave the highest accuracy (74%) and F1 score (86%), and the hybrid classification improved the accuracy of results by up to 12%.

## Elish, M. (2019) [4] proposed Stochastic Gradient Boosting Trees (SGBT) as a novel computational intelligence technique. SGBT models have been selected as it overcomes low accuracy. The main characteristics of SGBT models include selecting variables automatically; no pre-processing needed; no outliers and over-training; missing values rectified automatically; and are reliable to inaccurate data; and speed efficient. SGBT and 16 other prediction models have been trained, optimized and cross validated using vulnerability data sets from multiple versions of two open-source Web applications written in PHP. The results show improved prediction accuracy, by 2.5 per cent and 7 per cent, by the SGBT models over the other 16 models in the two data sets.

## Tama, B. A., Nkenyereye, L., Islam, S. R., & Kwak, K. S. (2020) [5] used Stacked ensemble classifier whose base learners are random forest, gradient boosting machine, and XGBoost for anomaly detection in web traffic. But they have a poor performance due to poor architecture. So this technique is proposed. First the base learners are trained using training data, then GLM is trained to make a final prediction based on the prediction mixtures of the base learners. The aim of the stacked ensemble is to construct a diverse group of strong base learners. The performance metrics accuracy, F1, AUC and specificity are calculated and compared for GBM, RF, XGBOOST across all the four datasets. The proposed model is compared with GBM, Random Forest, and XGBOOST. The experimental results show that proposed model outperforms other algorithms in accuracy, F1, AUC and specificity.

## Acosta, M. R. C., Ahmed, S., Garcia, C. E., & Koo, I. (2020) [6] proposes a new and innovated approach based on extremely randomized trees algorithm for cyber attack detection in smart grid networks. In the recent times Machine Learning (ML) based detection tools are starting to gain a lot of traction, since they do not need any such complex models, they are generally fed the data of all the sensors of a grid in a steady, free and secure state, and the algorithm learns from it. In case any stealthy attack, the ML algorithm automatically detects a diversion from the ideal values and hence alerts the grid to reset. The performance was compared to MLP, RF, AdaBoost, KPCA Extra Trees, PCS-SVM,GA- SVM, PCA-iForest algorithms and it was found that it out performs all other algorithms by a margin of quite significance.

## Alswaina, F., & Elleithy, K. (2018) [7] proposed an ensemble classifier called extremely randomized trees that inculcates the optimized nature by reducing the permission set by 18%, by classifying permissions based on their importance and likelihood to cause damage in case of intrusion. A customized data set was used in the study, it contained the general information about the families list, SFCand (a candidate subset of selected features SF ), MBcand (a two-dimensional binary matrix result from applying SFCand ), MWcand (a two dimensional weighted matrix result from applying the weight of each features in SFCand ), and NoOfThreads. The algorithm was compared to SVM, Bagging, ID3 , Neural Networks, KN. The Random Forest based on randomized trees stood out to be the best in terms of accuracy and speed.

## Belouch, M., El Hadaj, S., & Idhammad, M. (2018) [8] proposed a comparative analysis of all the most famous intrusion detection methods that use machine learning algorithms such as SVMs, Naïve Bytes, Decision tree, and random forest. This paper gives us the insight on how tree-based algorithms have proven to be the best among all, and how randomized trees are superior. This paper uses apache spark to judge the algorithms on the basis of accuracy, sensitivity, specificity, training time, and as well as prediction time. This study proves that random forest is overall best algorithm for intrusion detection when it comes to machine learning algorithms, since it has the highest accuracy, sensitivity as well as specificity. With a record training time and prediction time.

## Malik, A.J., Khan, F.A.(2018) [9] proposed a technique which is a hybrid approach in which Particle swarm optimization is used for node pruning and the pruned DT is used for classification of the network intrusions. Single layer as well as the multi- objective algorithms are used to make sure comprehensive study. The data set used is a well-established one called KDD99Cup dataset. This dataset is an industry standard and very well respected by the cyber security community. The results of the paper proved to be highly fruitful, indicating a increase detection rate, accuracy and precision. Also, the false positive rate dropped in number. When compared to conventional DT techniques, a pruned DT showed better performance dude to various underlying reasons.

## Jie Gu , Lihong Wang, Huiwen Wang , Shanshan Wang (2019) [10] used SVM ensemble with feature augmentation for network intrusion detection. Thus the proposed model is called DT-EnSVM. It can achieve a good and robust performance, which has huge competitive advantages when compared to other existing methods in terms of accuracy, detection rate, false alarm rate and training speed. NSL-KDD dataset has been used. It contains TCP connection records that consist of 41 information features plus one labelling feature. The 41 informational features are used to describe the details of each TCP connection in the dataset; the labelling feature helps to specify each connection as either normal or abnormal. This model gave an accuracy of 99.41%.

## Soodeh Hosseini, Behnam Mohammad Hasani Zade (2020) [11] proposed a new hybrid intrusion detection method with two phases; a feature selection phase and an attack detection phase. In feature selection phases, MGA-SVM, which is a wrapping technique, is used. This method combines features of support vector machine and genetic algorithm with multi-parent-crossover and multi-parent-mutation. In attack detection phase, artificial neural network is used for attack detection. To improve its performance, a combination of a hybrid gravitational search and a particle swarm optimization is used to train the classifier. Thus the proposed model is called MGA-SVM-HGS-PSO-ANN. The proposed method attained a maximum detection accuracy of 99.3%, dimension reduction of NSL-KDD from 42 to 4 features, and needed only 3s as maximum training time.

## Gautam M. Borkar, Leena H. Patil, Dilip Dalgade, Ankush Hutke (2019) [12] used adaptive SVM classification for Intrusion Detection where an acknowledgement based method is utilized for reporting the malicious sensor nodes in Wireless Sensor Networks (WSN). This helps to detect attacks such as attacks such as DOS, probe, U2R, R2L incorporation with IDS. KDD cup 1999 dataset comprising of 41 features labelled as attack or normal is used. The proposed method attained a maximum detection accuracy of 98%.

## Weijian Fanga , Xiaoling Tan , Dominic Wilbur (2020) [13] combined the advantages of Elman neural network and robust SVM noise data elimination, to solve the safety risks of intrusion detection of information systems to ensure the safety of information systems. The data set was collected from a 9-week network data stream in a simulated military network. Result shows that when the false alarm rate is 0, the intrusion detection rate based on robust SVM neighbour classification is 87.3%; when the false alarm rate is 2.8%, the detection rate can reach as high as 100%.

## Muhammad Sarwar, Faisal Mehmood, Muhammad Abid, Abdul Qayyum Khan, Sufi Tabassum Gul, Adil Sarwar Khan (2019) [14] applied PCA, FDA, and SVM to detect and classify high impedance faults. Further, better results are obtained by using M- SVM classifier, fault classification rate of SVM is better than FDA. M-SVM algorithm can detect all types of HIF. Multiclass SVM gave the best result to detect and locate high impedance faults accurately. To obtain training data from the simulation, a fault model is simulated in Simulink and training data is obtained from voltage and current sensors installed in the network. The proposed method gave 100% dependability and 100% security.

# **HARDWARE AND SOFTWARE REQUIREMENTS**

## **Hardware:**

* Processor: 11th Gen Intel(R) Core(TM) i5-1135G7 @ 2.40GHz 2.42 GHz
* RAM: 8GB RAM
* HardDisk: Minimum 512 GB

## **Software:**

* Operating System: Windows 10
* IDE: Jupyter Notebook
* Programming Language: Python

# **DETAILED DESIGN**

* 1. **SYSTEM ARCHITECTURE**

**Data Preprocessing**

Removal or replacement of null and inconsistent values

**Feature Engineering**

Normalization

Label Encoding

Feature Selection

Sampling

Data Presentation

**Model Building Phase**

Train-Test Splitting

***Building Supervised ML Models:*** GBM, LGBM, Ensemble using Voting

***Learning Process:*** 10 cross validation and Hyperparameter Tuning using

Determining Best Model

**Model Evaluation and Performance Metrics**

Generating Classification Report, Confusion Matrix

Performance Comparison

**CIC Darknet 2020**

Fig 1: Architecture Diagram

* 1. **UML DIAGRAM (USE CASE)**

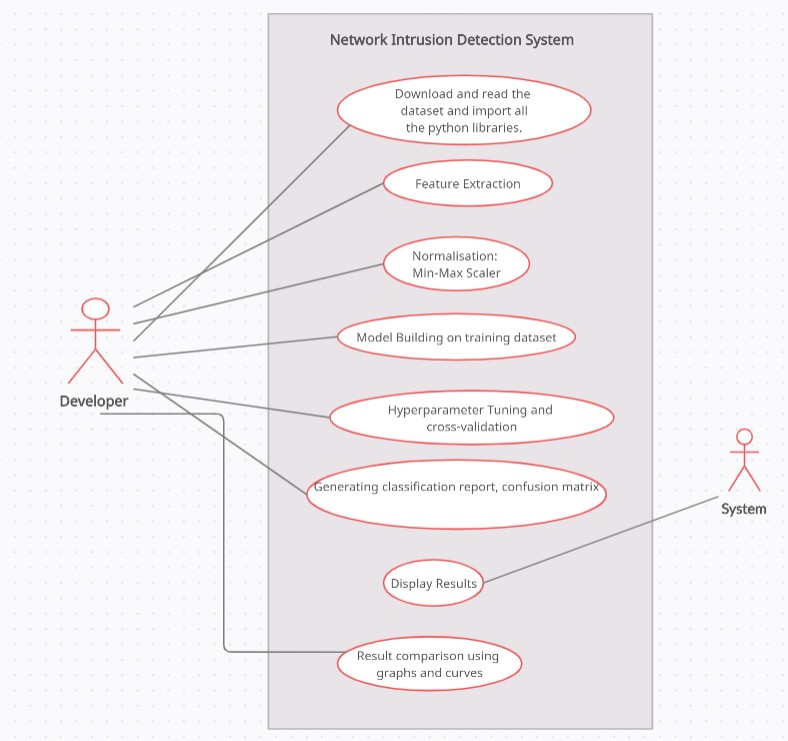
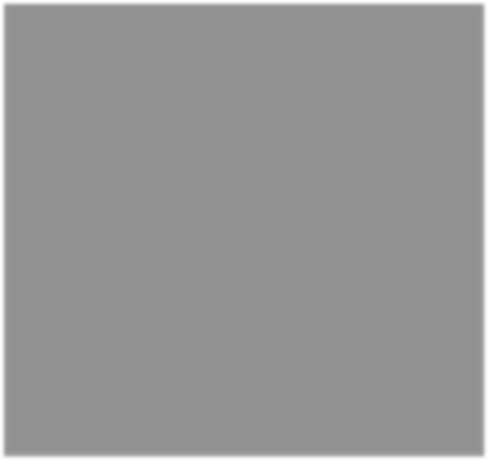


Fig.2: Use Case Diagram

* 1. **MODULE 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

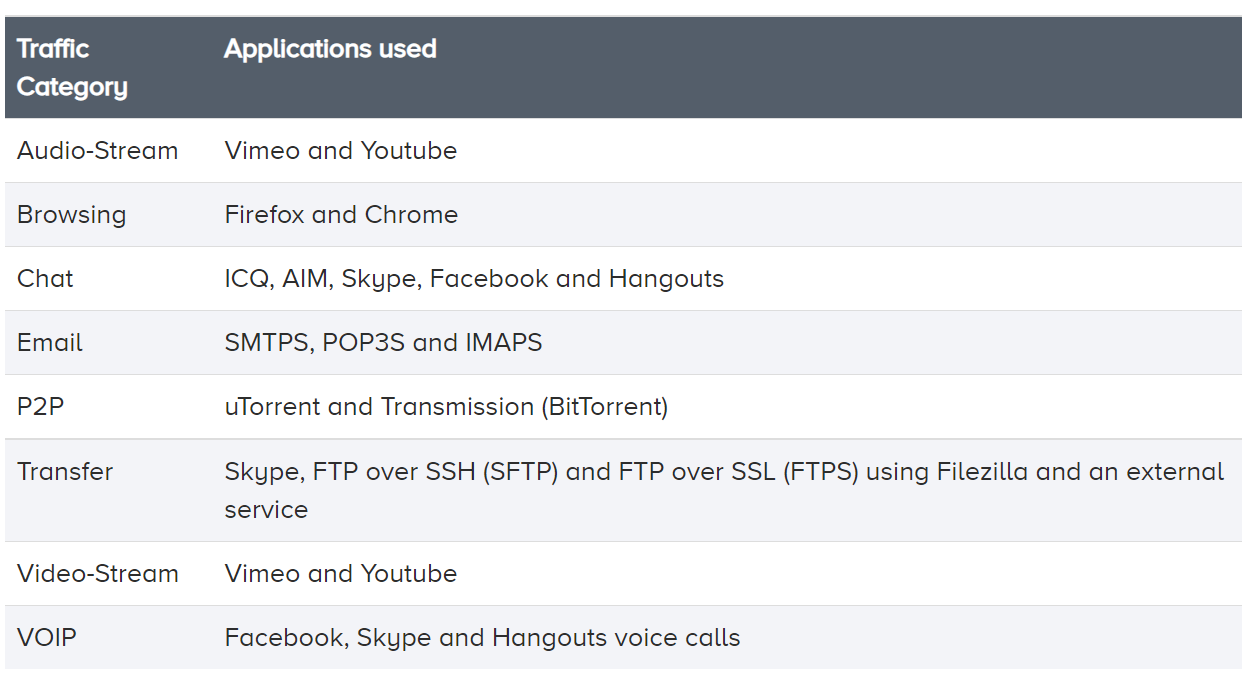
****

Fig 3: Darknet Traffic Details

## 

## 

## Fig 4: (a) presents the details of number of samples of benign and darknet traffic at first layer and (b) highlights the number of encrypted flows in our darknet traffic.

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 Preprocessing Phase:**

## For multiclass classification:

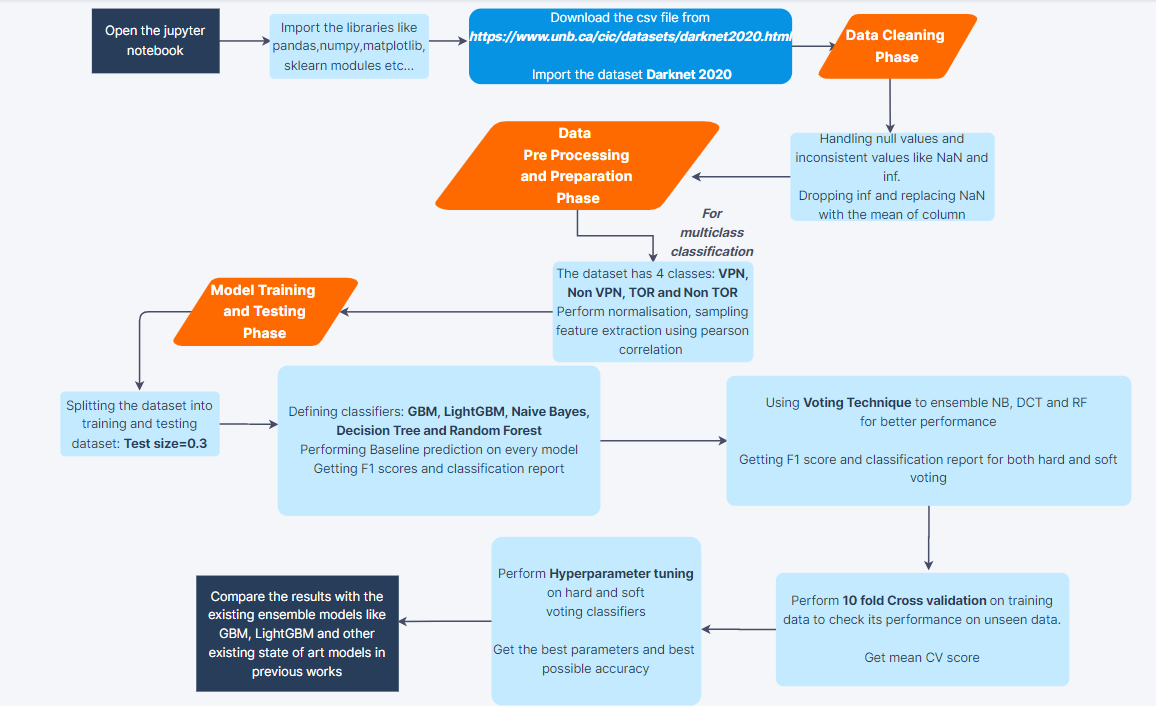
* Label encoding is done for all the classes to convert string to number which is needed for further computation.
* This dataset is then balanced, that is, the number of rows of each class is made approximately equal by either over sampling or under sampling.
* Target variable is then separated from 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.

## **Model Training and Testing Phase:**

In this phase:

* First the dataset is split into training and testing dataset.
* Various classifiers are defined: **GBM, LightGBM, Random Forest, Decision Tree Classifier and Knn**
* Initially all the models are defined with the respective hyper parameters. Classification Report is generated and accuracy is obtained. Confusion matrix is also obtained.
* Voting Technique is then used to combine and form an ensemble classifier using Random Forest, Naïve Bayes and Knn classifiers. Both hard voting and soft voting are used and the performance of this ensembled model proved to be better than the individual performances of all the three models as well as GBM and LightGBM.
* 10-Fold Cross Validation is done on both hard voting and soft voting classifier to determine the performance of the model on the unseen data.
* After this, hyper parameter tuning is done on these models on the training dataset, using GridSearch CV.

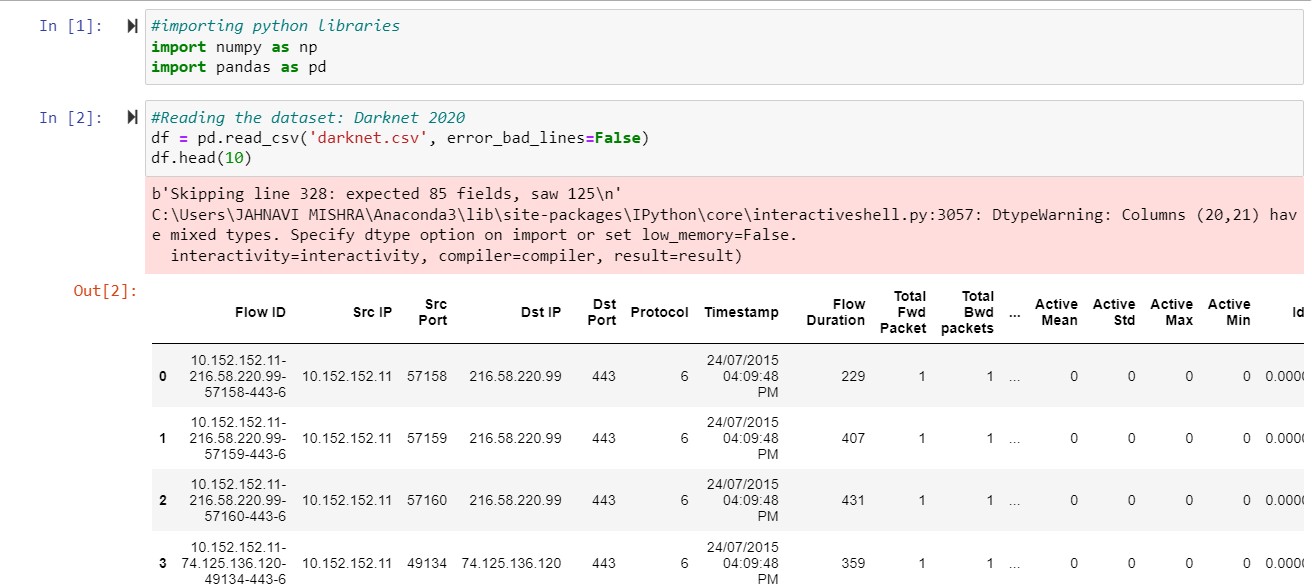
# **DATA FLOW DIAGRAM**



## Fig.5: Data Flow Diagram

1. **IMPLEMENTATION**

**5.1 Importing Dataset and Modules:**





Graphical user interface, text, table

Description automatically generated

* Correlation Matrix:

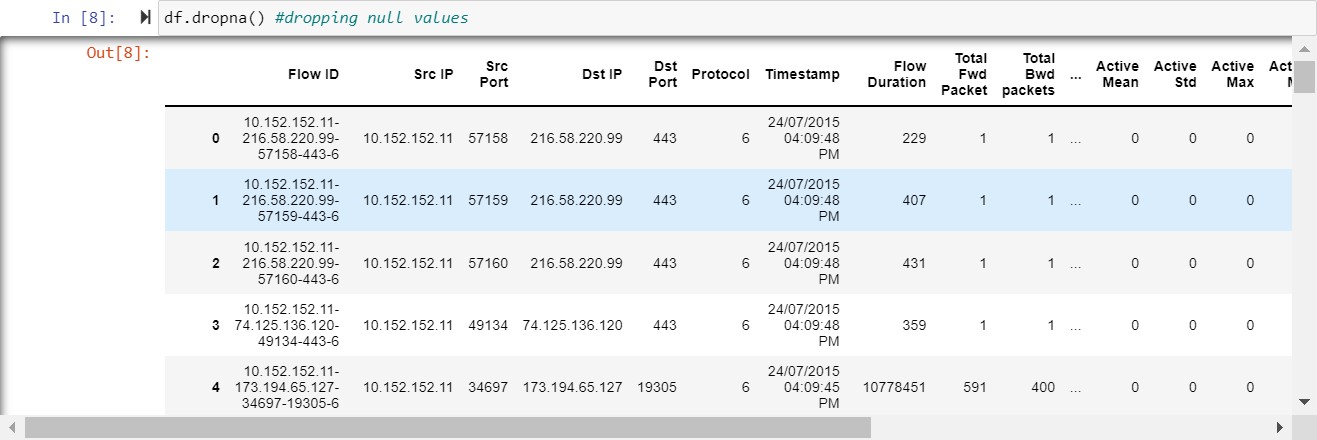
Table

Description automatically generated

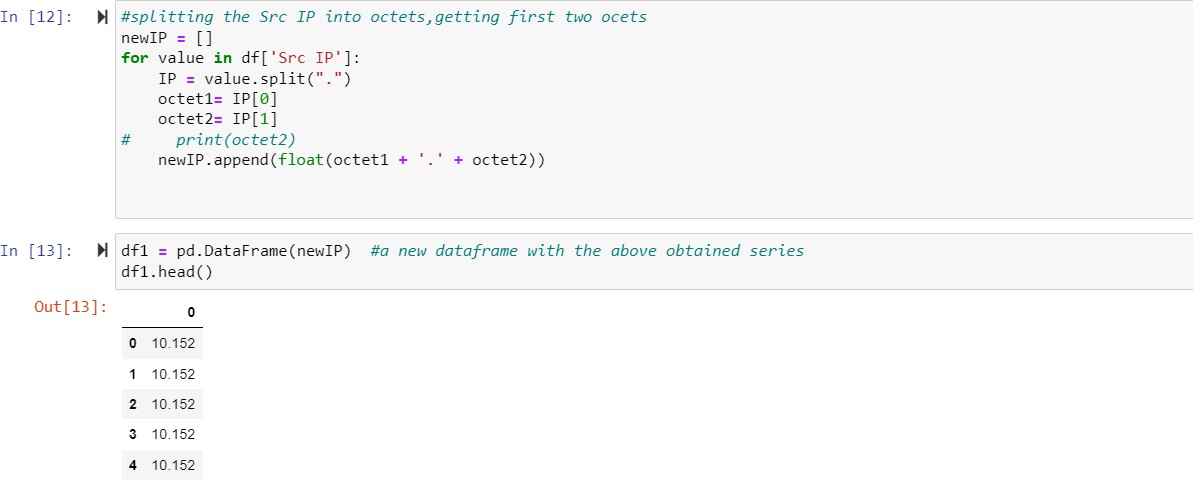
* 1. **Data Cleaning Phase:**

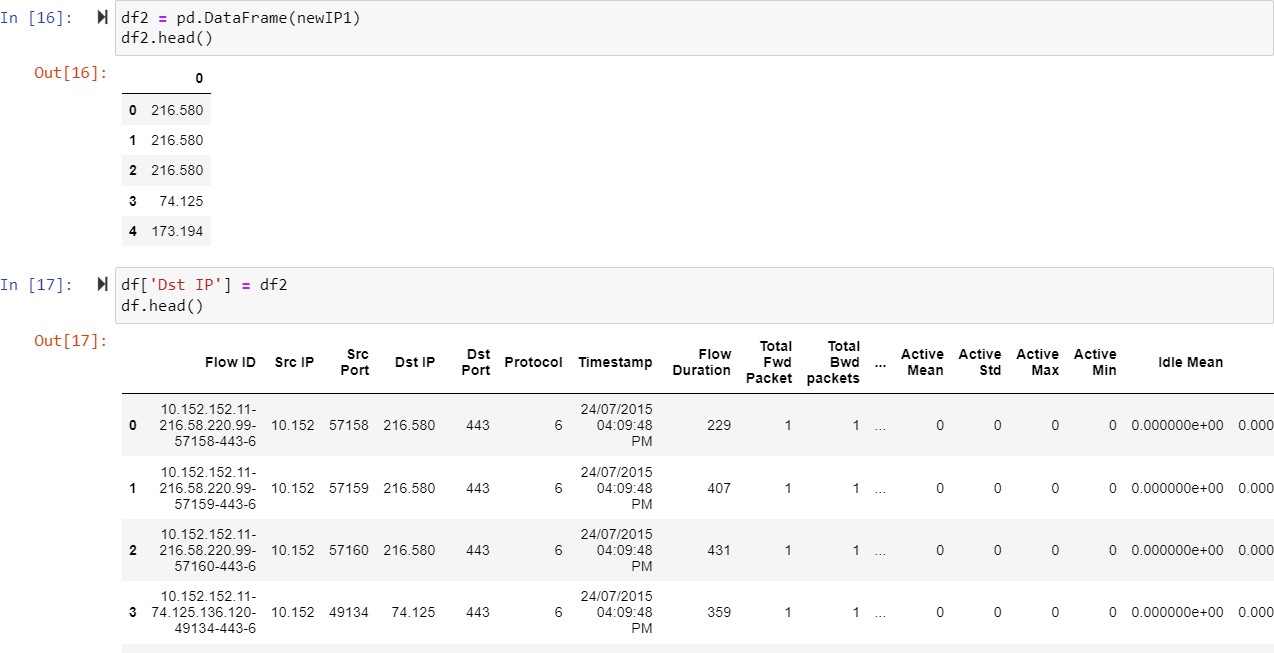
In this stage, I will be performing data cleaning like dealing with missing values, inf values, changing the columns like splitting columns if required. So when all the columns that may cause error further due to their datatype or format or problematic values will be rectified and then we can proceed towards data preprocessing stage.

* Dropping Null values:



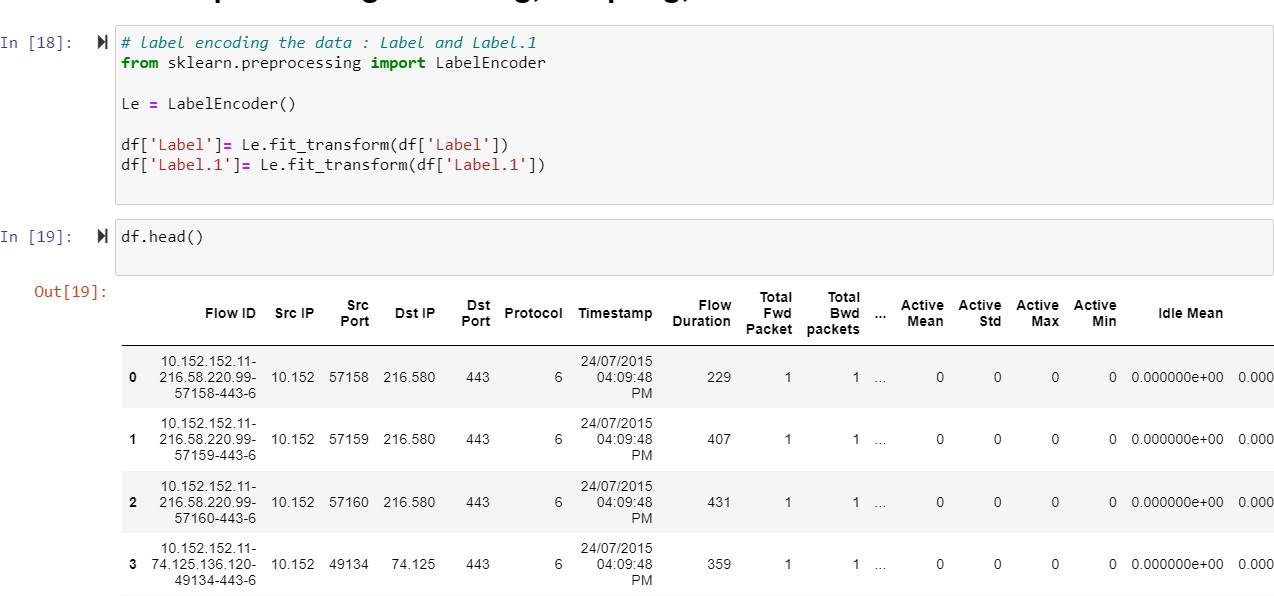
* Modifying Src and Dst IP columns:

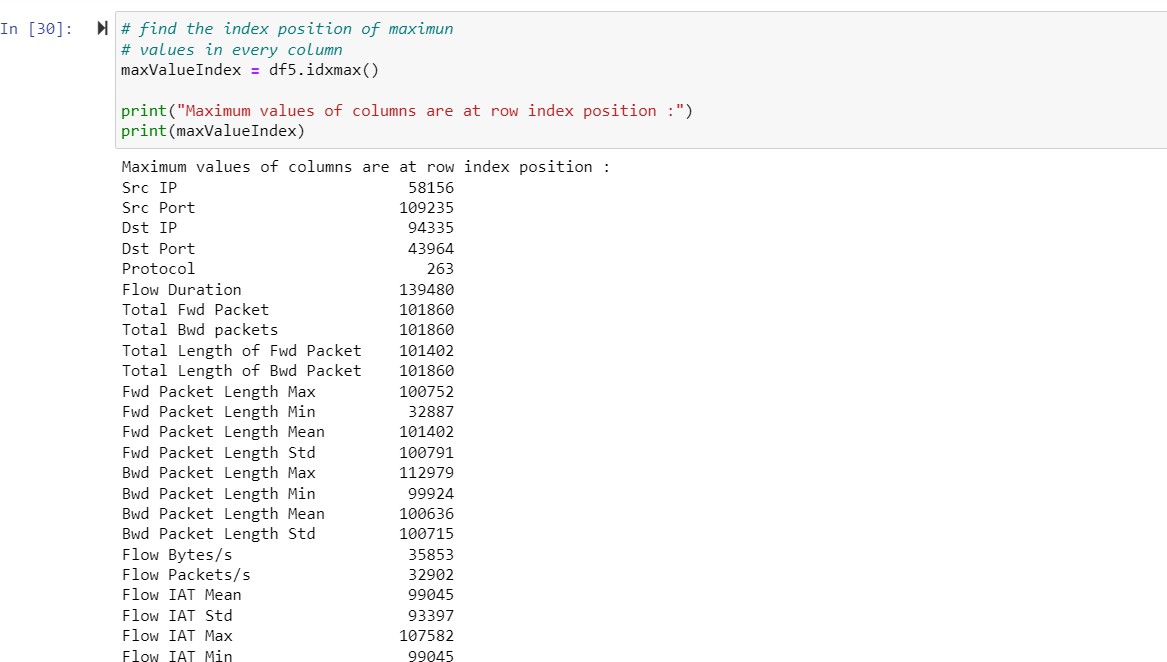




## **Data Pre Processing Stage:**

* + - **Label Encoding:**





Table

Description automatically generated

The next four operations, convert exponential values to float values for the ease of further computation.

Table

Description automatically generated

Graphical user interface, text, application, Word

Description automatically generated

Table

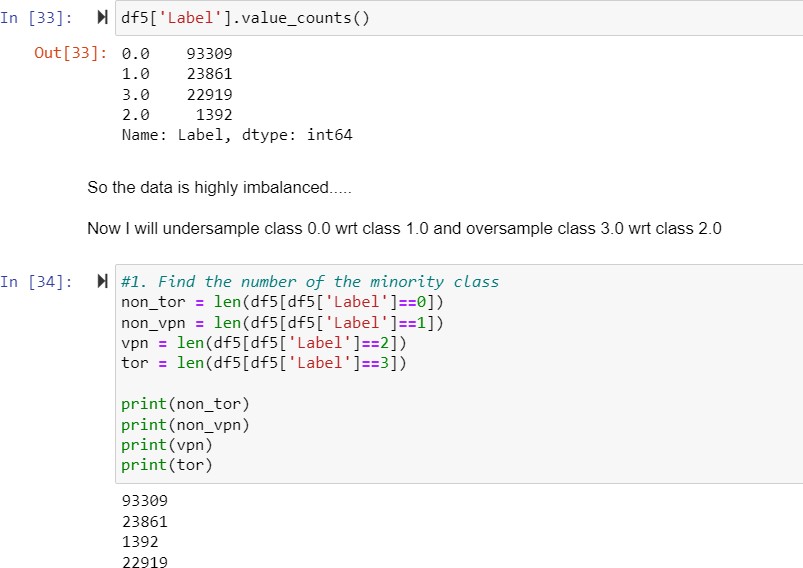
Description automatically generated

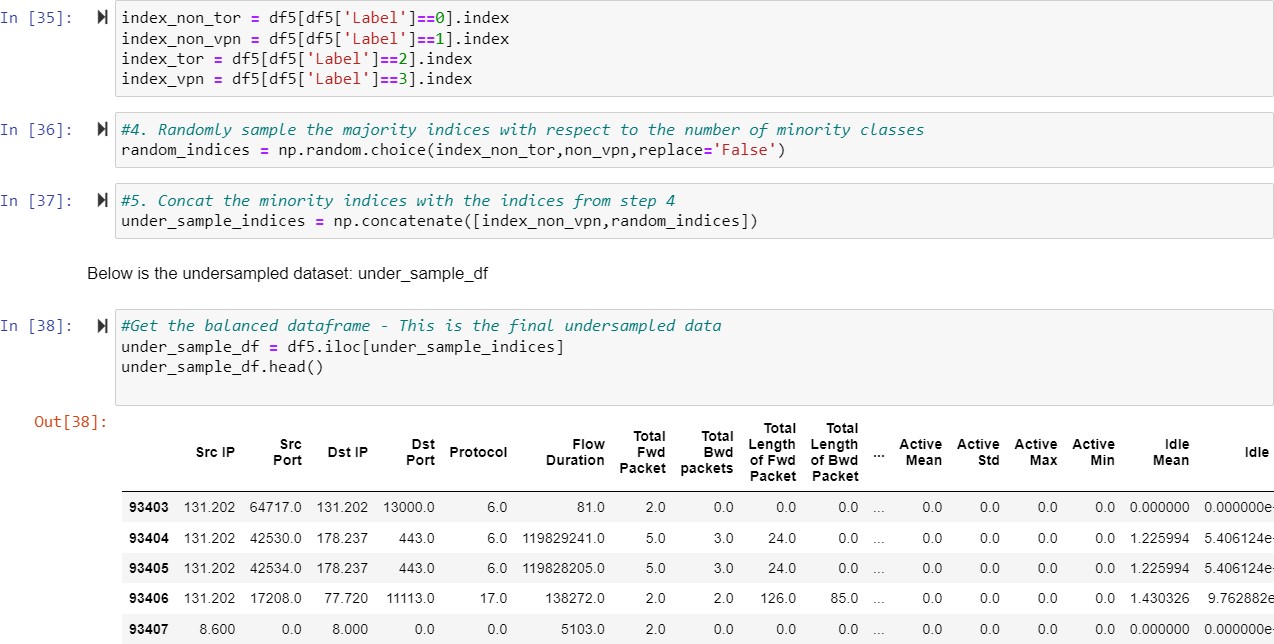
* + - **Sampling:**

Sampling is used to balance the data that is to balance the number of each class in the Label.

**Oversampling:** to increase the number of minority classes

**Undersampling:** to decrease the number of majority classes





# Graphical user interface, text, application Description automatically generated

# Graphical user interface, text, application, email Description automatically generated

# Final balanced Data frame:

# Graphical user interface, application Description automatically generated

## **Normalisation:**

## Normalization is a scaling technique, applied during data preparation phase to change the values of numeric columns in the dataset to use a common scale. It is not necessary for all datasets in a model. It is needed only when features of machine learning models have variable ranges.

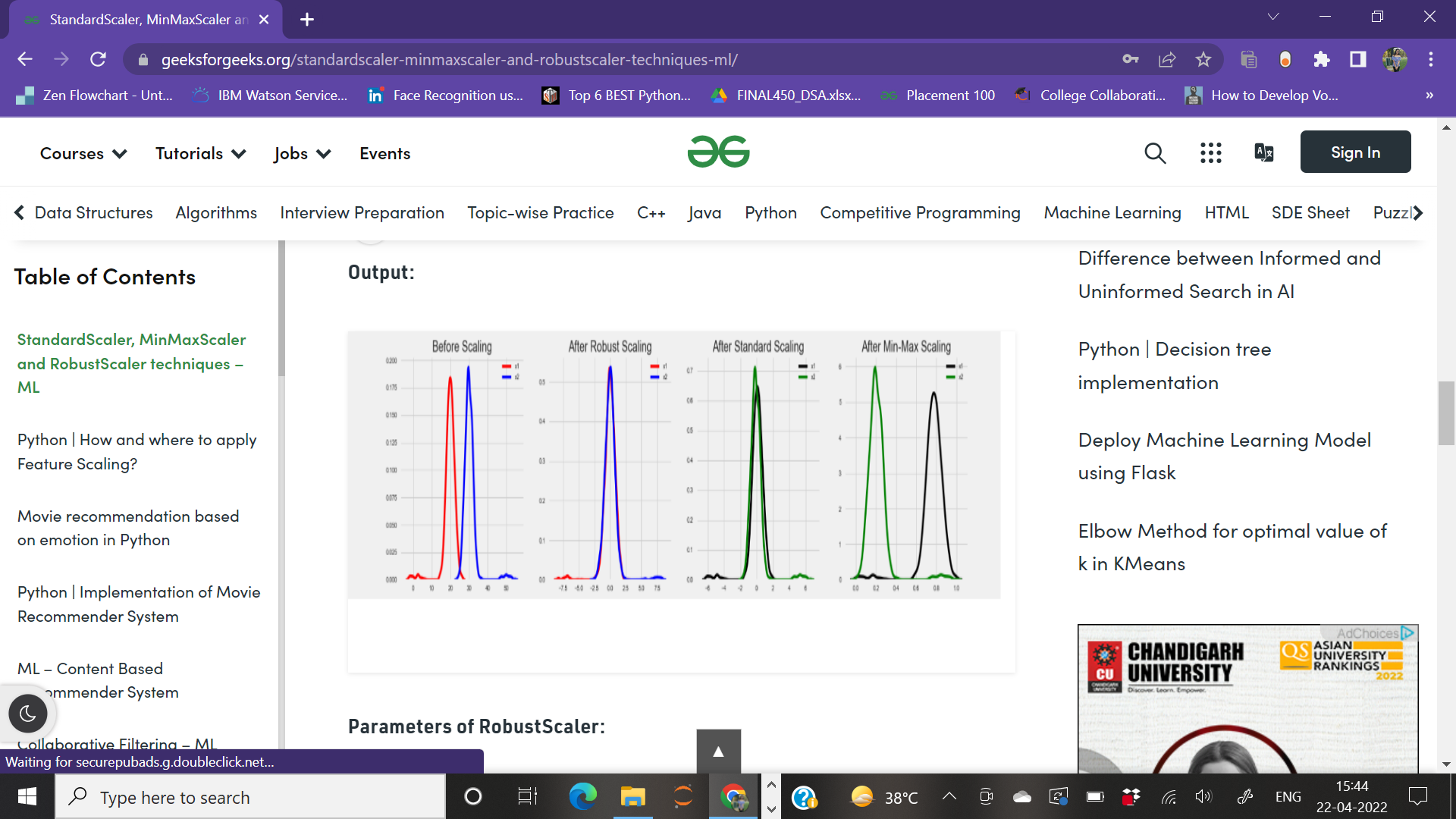


Fig 6: Data before and after scaling and normalisation

## All the values of the dataset are then normalised to a given range which makes the further computation easier. This is necessary for the consistency of the values of the dataset. I will be using MinMaxScaler for normalisation. So all the values will be converted into values between 0 to 1.

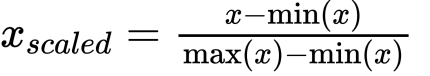
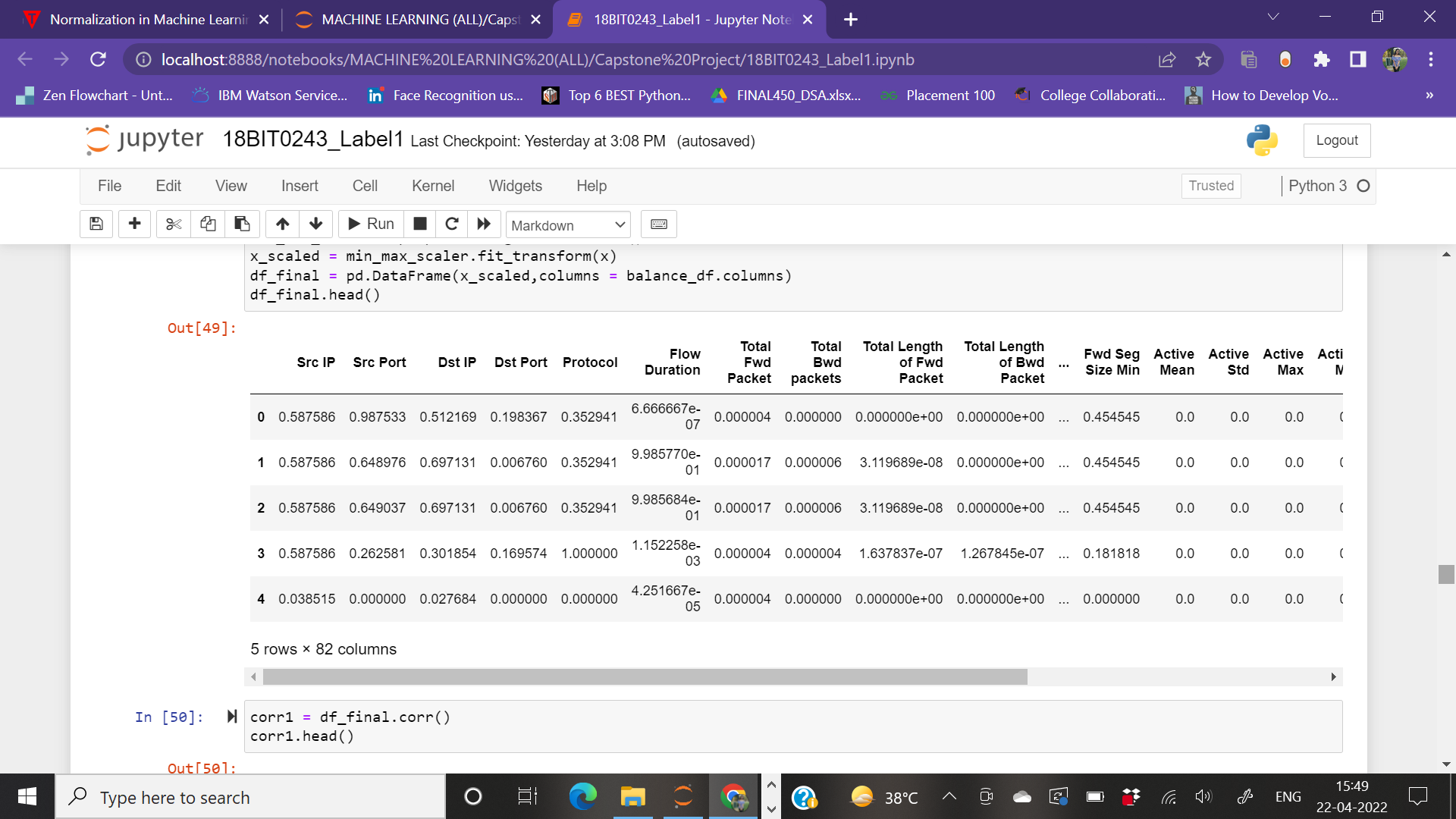


Fig 7: Min-Max Scaler Formula

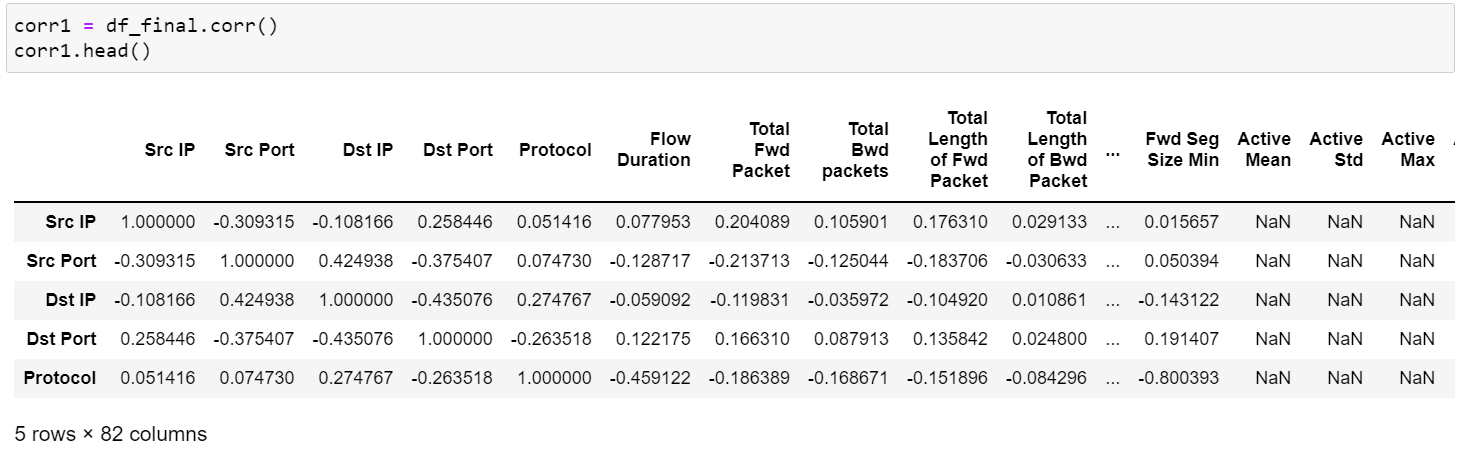


**Output:**



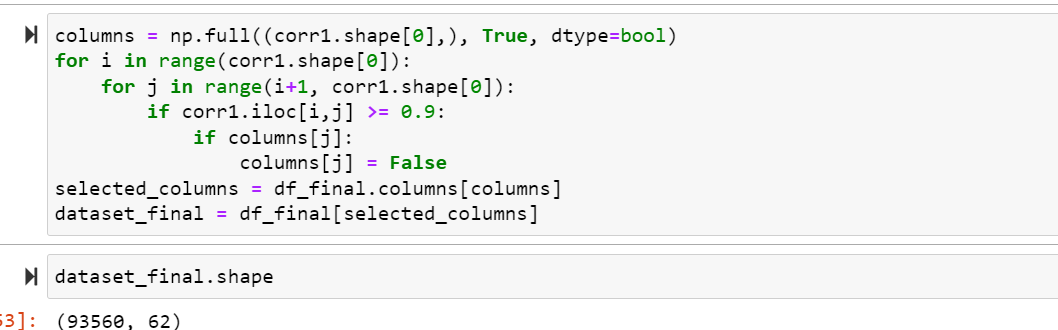
**Feature Selection**

* **Getting the correlation matrix**



As we can see for many pairs of columns, correlation does not even exist as an integer. Also for various columns, the correlation is as high as 1.000. This means that the columns have almost similar effect on the performance and can be treated as one. So we can keep one of them and remove the other to decrease the no. of unnecessary columns and improve the computation speed and performance.

Here pearson correlation coefficient is used and 0.9 has been considered as the threshold correlation.



Finally only 62 out of 85 columns are left.

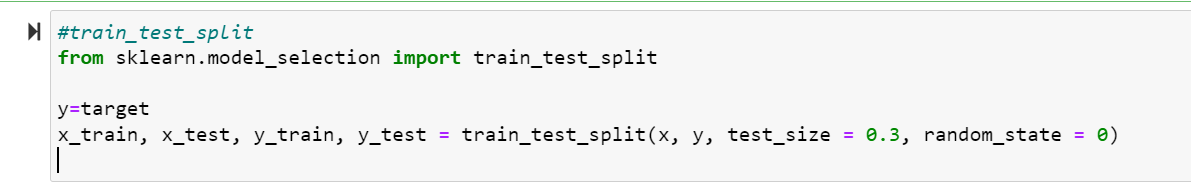
We can proceed further with this dataset.

* 1. **Model Building Phase:**

**This phase includes:**

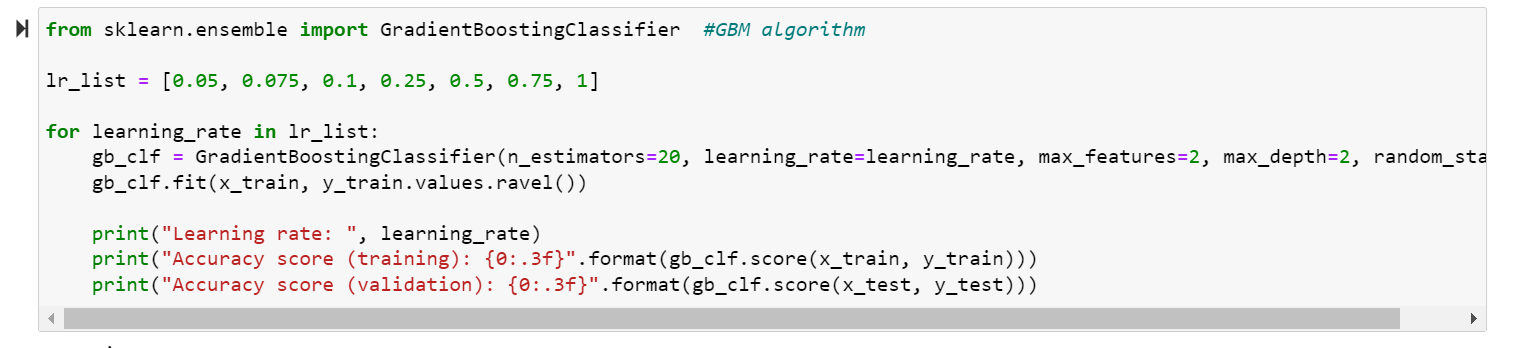
* Splitting into training and testing data
* Defining the model: GBM, LightGBM, Decision Tree, Random Forest, Knn
* Performing baseline tuning on test dataset for each model: Getting evaluation metrics
* Building an ensemble model combining Decision Tree , Random Forest and Knn using hard and soft voting techniques and performing baseline tuning: Getting evaluation metrics
* Performing 10 fold cross validation on the model with best performance in the baseline tuning
* Performing Hyperparameter Tuning using GridSearchCV: Getting the best parameters
* Comparing the models before and after tuning and determining the best possible score.

1. **Train Test Split**



1. **Model Building and Testing:**

* Building GBM Model and performing baseline tuning using different learning rates on test dataset



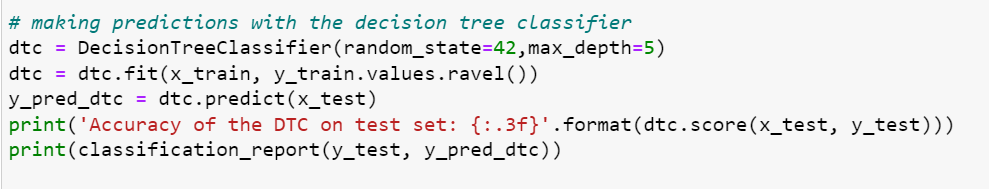
* Building LightGBM Model and performing baseline tuning using different learning rates on test dataset



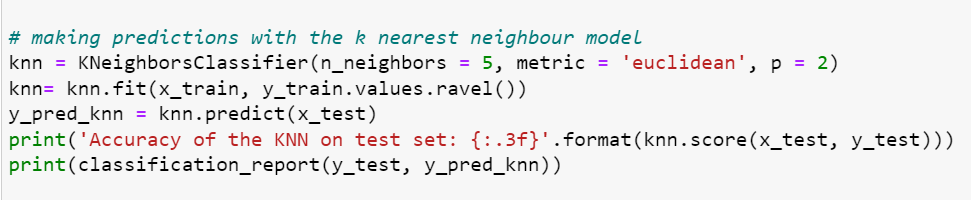
Graphical user interface, application

Description automatically generated

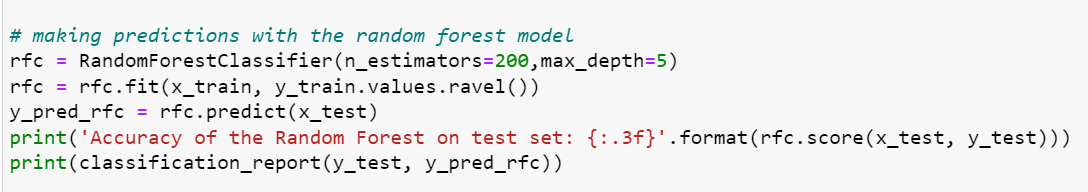
* Building Decision Tree Classifier and performing baseline tuning on test dataset



* Building Knn Classifier and performing baseline tuning on test dataset



* Building Random Forest Classifier and performing baseline tuning on test dataset



1. **Building Ensemble Model using Voting Technique:**

A voting ensemble is an ensemble machine learning model that unites the forecasts from multiple other models.

It is a technique that may be used to improve model performance, perfectly achieving better performance than any single model used in the ensemble.

A voting ensemble works by joining the forecasts from multiple models. It can be used for classification or regression. In the case of classification, the likelihoods for each label are tallied and the label with the majority vote is projected.

There are two tactics to the majority vote prediction for classification; they are hard voting and soft voting.

Hard voting involves tallying the predictions for each class label and projecting the class label with the most votes. Soft voting involves tallying the predicted likelihoods (or probability-like scores) for each class label and predicting the class label with the largest chance.

* **Hard Voting**. Predict the class with the largest sum of votes from models

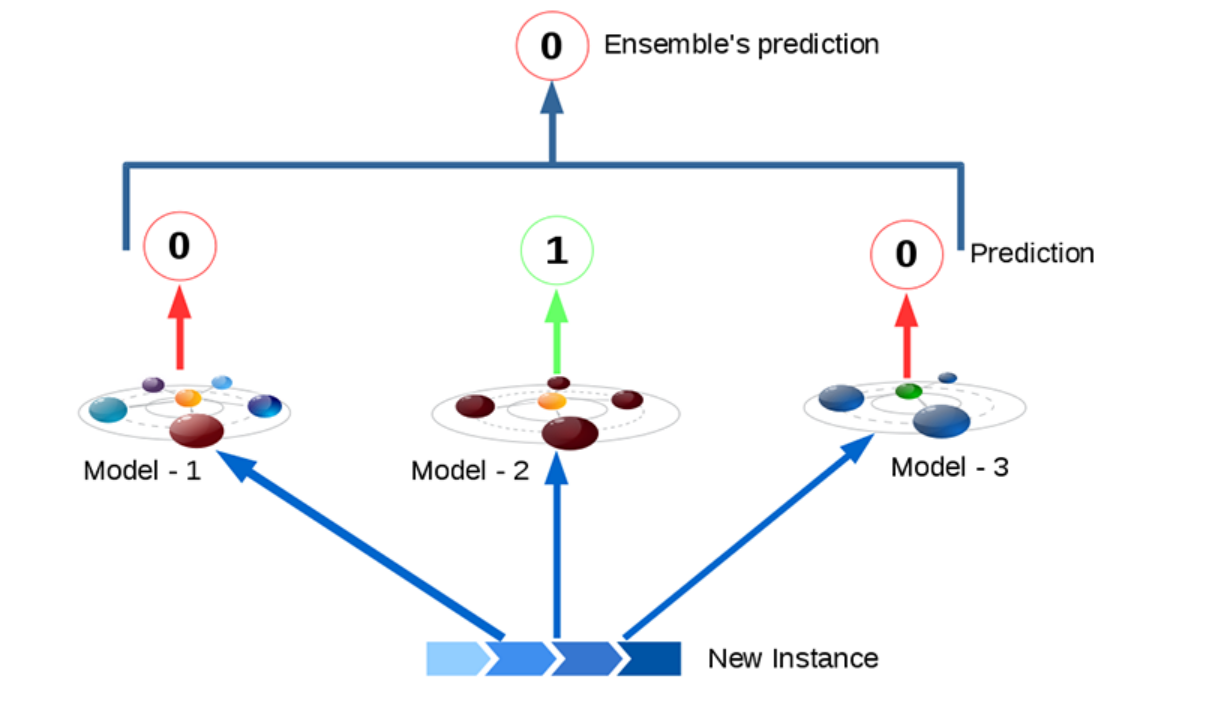


Fig 8: Hard Voting

* **Soft Voting**. Predict the class with the largest summed probability from models.

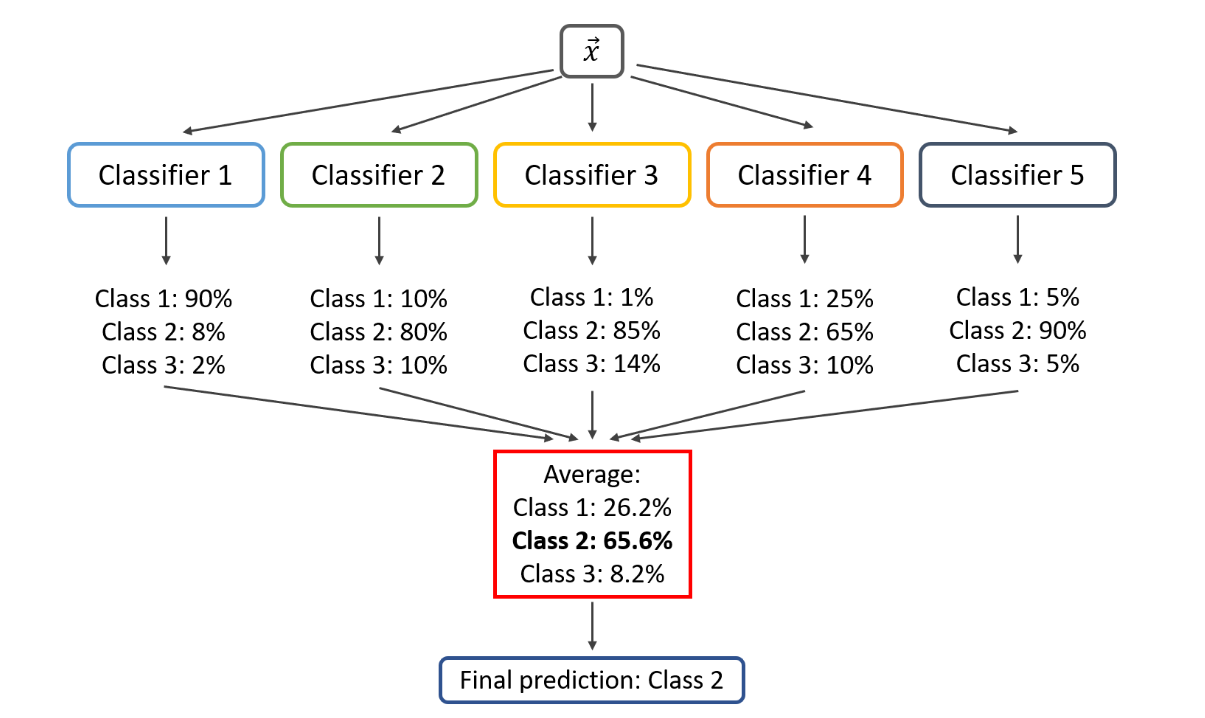
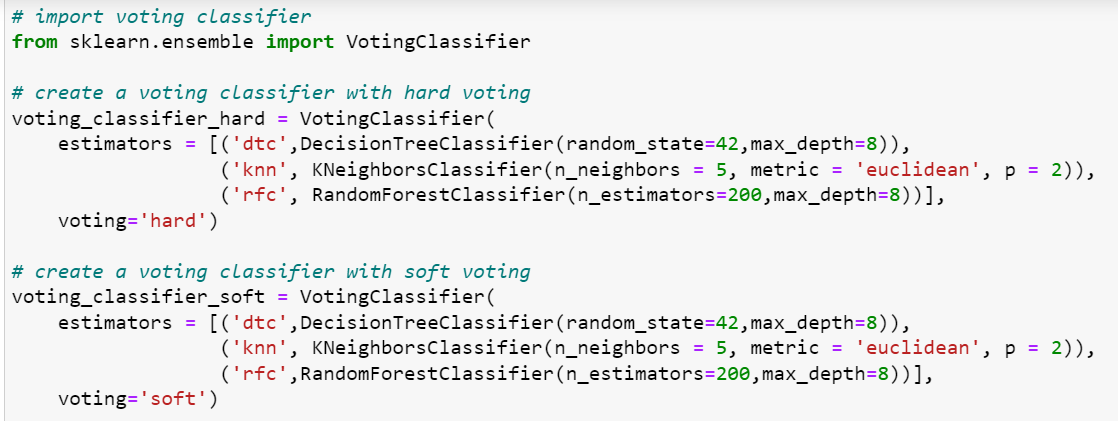
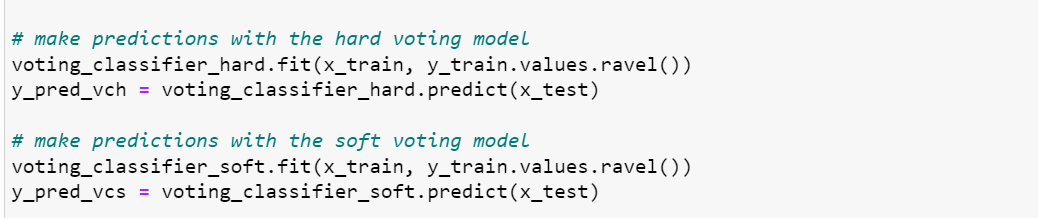
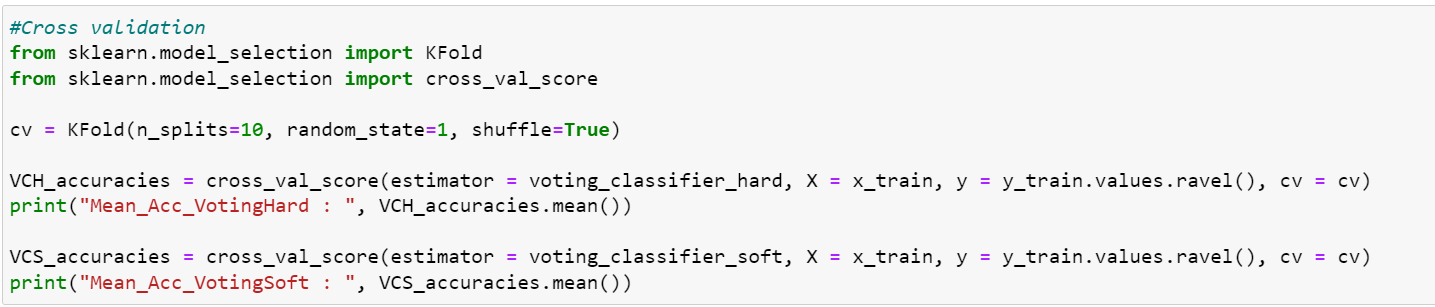


Fig 9: Soft Voting

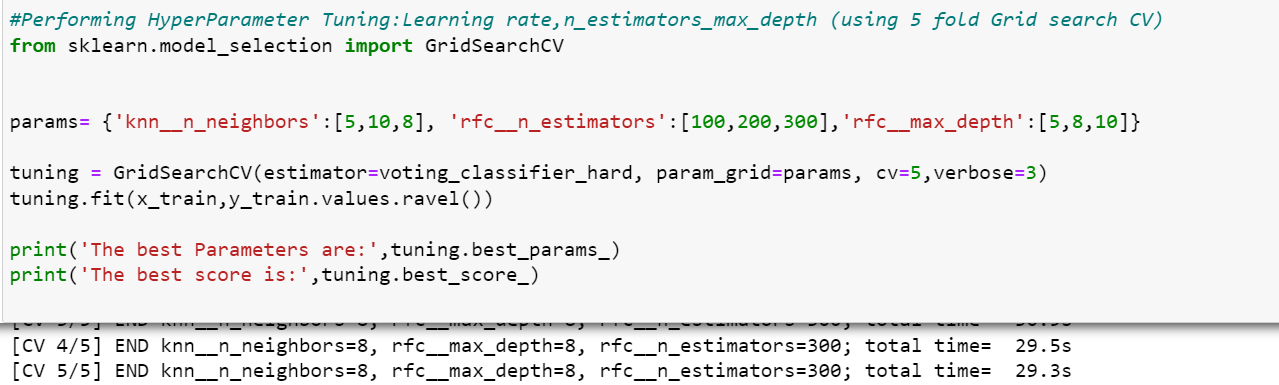


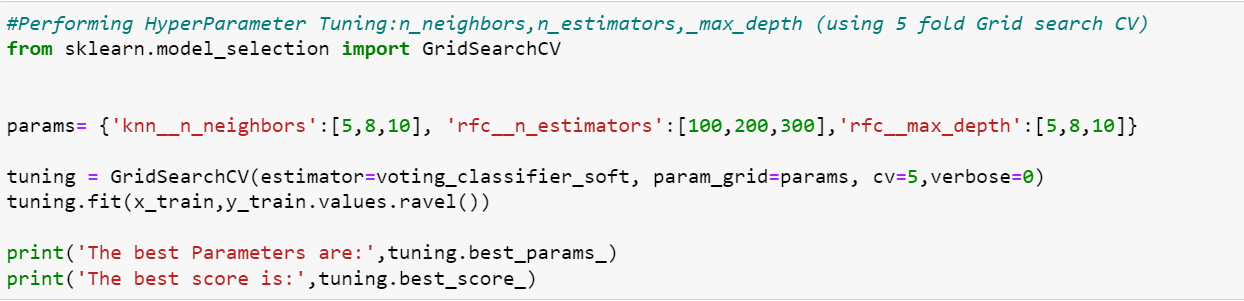


1. **Performing 10 Fold Cross-Validation on the hard and soft voting classifiers**



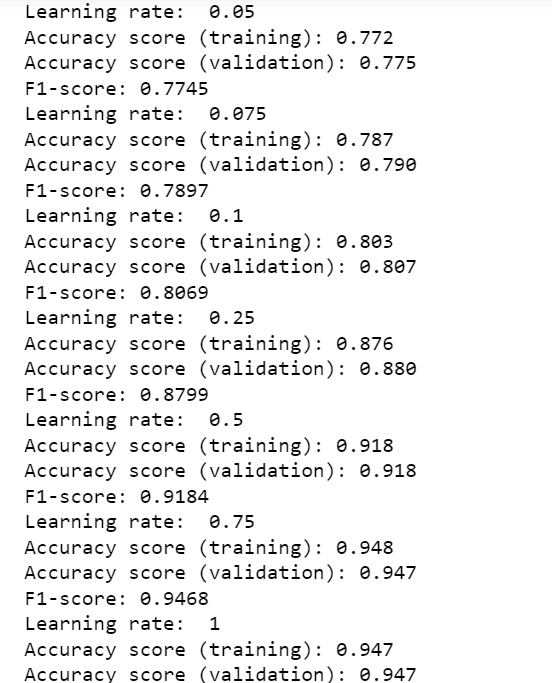
1. **Hyperparameter Tuning using GridSearchCV on the hard and soft voting classifiers.**





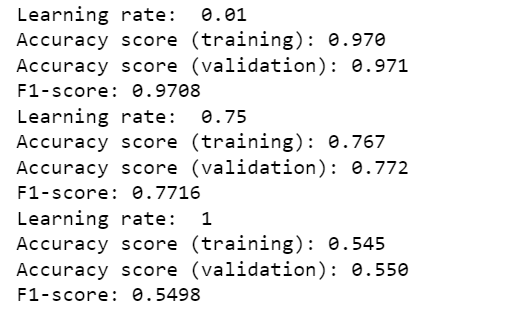
## **PERFORMANCE METRICS**

* **Evaluation Metrics of Gradient Boosting Classifier:**



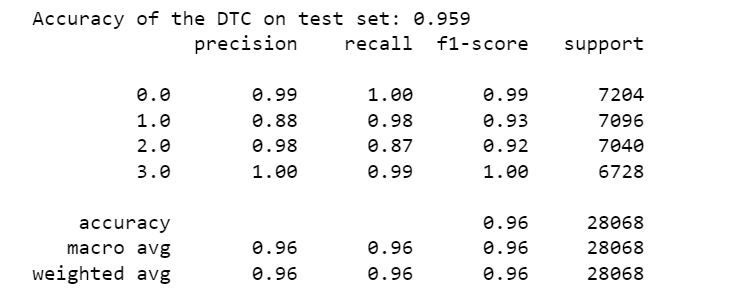
As we can see, GBM gives the best results when the learning rate is 0.75. Training accuracy is 94.8% and validation accuracy is 94.7%

* **Evaluation Metrics of LightGBM:**

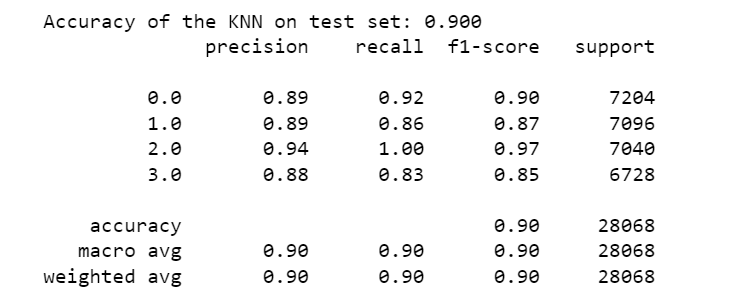


As we can see, LightGBM gives the best results when the learning rate is 0.01. Training accuracy is 97.0% and validation accuracy is 97.1%

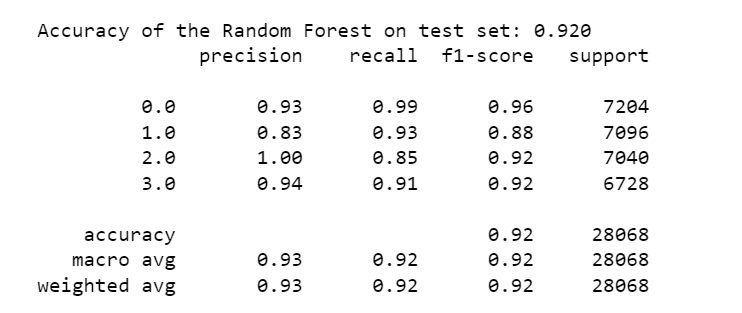
* **Evaluation Metrics of Decision Tree Classifier**



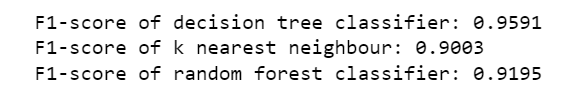
* **Evaluation Metrics of K-Nearest Neighbour Classifier**



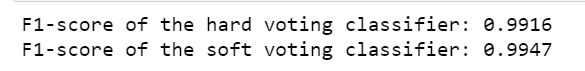
* **Evaluation Metrics of Random Forest Classifier**



* **F1 Scores:**



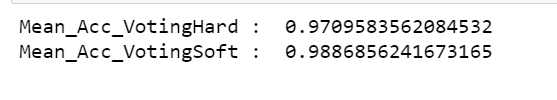
* **Evaluation metrics of Voting Classifier built by ensembling DTC, Random Forest and Knn:**



As we can see the performance of both and hard and soft voting classifier is better than the individual performance of the three models ensembled above as well as the pre existing ensemble models like GBM and LightGBM.

Hence the new ensemble model built is a successful one and has a great performance.

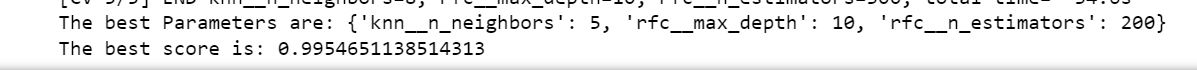
* **10 Fold Cross Validation**



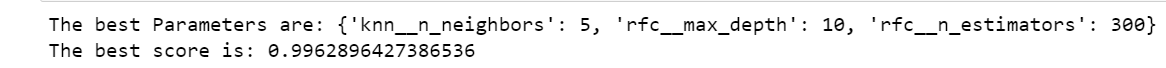
This shows the performance of the classifier on the unseen data.

* **Hyperparameter Tuning:**

For Hard Voting Classifier, the parameters tunedare'**knn\_\_n\_neighbors', 'rfc\_\_max\_depth', 'rfc\_\_n\_estimators'**



For Soft Voting Classifier, the parameters tuned are **'knn\_\_n\_neighbors', 'rfc\_\_max\_depth', 'rfc\_\_n\_estimators'**



**Final Comparison:**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Model | Gradient Boosting | LightGBM | Decision Tree Classifier | Random Forest Classifier | K-Nearest Neighbour | Voting Classifier  (Hard) | Voting Classifier  (Soft) |
| Accuracy | **94.7%** | **97.1%** | **95.9%** | **92%** | **90%** | **99.5%** | **99.6%** |
| F1 Score | **0.9468** | **0.9708** | **0.9591** | **0.9003** | **0.9195** | **0.9954** | **0.9962** |

Table/Chart 1: Performance Comparison

Table**/**Chart 2: Performance Comparison

**Existing Techniques:**

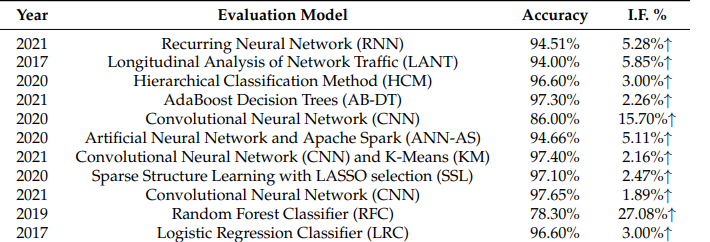


Fig 10: Comparison with Existing State of Art Models

Clearly the proposed ensemble voting classifier out performs existing state of art models with accuracy of 99.55% (Hard) and 99.6% (Soft).

**7. CONCLUSION AND FUTURE SCOPE**

An efficient Darknet traffic detection system (DTDS) has been proposed, modelled, implemented, assessed in this project report. The performance of the proposed algorithm is compared with five supervised machine-learning techniques including Gradient Boosting Classifier, Light Gradient Boosting Classifier, Random Forest Classifier, Decision Tree Classifier and K-Nearest Neighbour Classifier. The developed DTDS ML models were evaluated on CIC-Darknet-2020 dataset involving a large number of captured cyber-attacks and hidden services provided by the Darknet grouped into four classes (VPN, TOR, Non-VPN, Non-TOR). This work shows that the DTDS-based Ensemble model built using voting technique by ensembling classifiers like DTC, Random Forest and Knn is superior among these three models the other evaluated models, scoring 99.5% (Hard voting) and 99.6% (Soft Voting) in classification accuracy. When compared with other existing state-of-the-art models, the best results have improved the performance of existing DTDS models by a factor of (1.7~27%). Consequently, the proposed model can be efficiently deployed in firewalls and other intrusion detection devices to detect Tor and VPN activities in communication networks.

**8. REFERENCES**

[1] Tama, B. A., & Rhee, K. H. (2019). An in-depth experimental study of anomaly detection using gradient boosted machine. Neural Computing and Applications, 31(4), 955-965.

[2] KHAFAJEH, H. (2020). AN EFFICIENT INTRUSION DETECTION APPROACH USING LIGHT GRADIENT BOOSTING. Journal of Theoretical and Applied Information Technology, 98(05).

[3] Meftah, S., Rachidi, T., & Assem, N. (2019). Network based intrusion detection using the UNSW- NB15 dataset. International Journal of Computing and Digital Systems, 8(5), 478-487.

[4] Elish, M. (2019). Enhanced prediction of vulnerable Web components using Stochastic Gradient Boosting Trees. International Journal of Web Information Systems.

[5] Tama, B. A., Nkenyereye, L., Islam, S. R., & Kwak, K. S. (2020). An Enhanced Anomaly Detection in Web Traffic Using a Stack of Classifier Ensemble. IEEE Access, 8, 24120-24134.

[6] Acosta, M. R. C., Ahmed, S., Garcia, C. E., & Koo, I. (2020). Extremely randomized trees-based scheme for stealthy cyber-attack detection in smart grid networks. IEEE Access, 8, 19921-19933.

[7] Alswaina, F., & Elleithy, K. (2018). Android malware permission-based multi-class classification using extremely randomized trees. IEEE Access, 6, 76217-76227.

[8] Belouch, M., El Hadaj, S., & Idhammad, M. (2018). Performance evaluation of intrusion detection based on machine learning using Apache Spark. Procedia Computer Science, 127, 1-6.

[9] Malik, A.J., Khan, F.A. A hybrid technique using binary particle swarm optimization and decision tree pruning for network intrusion detection. Cluster Computing , 667–680 (2018). <https://doi.org/10.1007/s10586-017-0971-8>

[10] Jie Gu , Lihong Wang, Huiwen Wang , Shanshan Wang “A novel approach to intrusion detection using SVM ensemble with feature augmentation” (2019) DOI: 10.1016/j.cose.2019.05.022

[11] Soodeh Hosseini, Behnam Mohammad Hasani Zade “New hybrid method for attack detection using combination of evolutionary algorithms, SVM, and ANN” (2020) DOI: 10.1016/j.comnet.2020.107168

## [12] Gautam M. Borkar, Leena H. Patil, Dilip Dalgade, Ankush Hutke “A novel clustering approach and adaptive SVM classifier for intrusion detection in WSN: A data mining concept” (2019) DOI: 10.1016/j.suscom.2019.06.002

## [13] Weijian Fanga , Xiaoling Tan , Dominic Wilbur “Application of intrusion detection technology in network safety based on machine. learning” (2020) DOI: 10.1016/j.ssci.2020.104604

## [14] Muhammad Sarwar, Faisal Mehmood, Muhammad Abid, Abdul Qayyum Khan, Sufi Tabassum Gul, Adil Sarwar Khan “High impedance fault detection and isolation in power distribution networks using support vector machines” (2019)

[15] Abu Al-Haija, Q., Krichen, M., & Abu Elhaija, W. (2022). Machine-learning-based traffic detection system for IoT applications. *Electronics*, *11*(4), 556.