

**School of Computer Science and Engineering**

# Real Time Malware Detection Using Machine Learning Algorithms

# Information Security Analysis and Audit

**G1**

18BCE0912 Malla Jyotsna

18BCE0686 S.A Hariprasad

Dr. SENDHIL KUMAR K.S

Associate Professor

School of Computer Science and Engineering

October 2020

**1.1AIM:**

Our project named “Real Time Malware Detection with the Use of Machine Learning Algorithms” is aimed at detecting the malware infected program executable files using machine learning algorithms. We are trying to compare the accuracy of 3 machine learning algorithms for detecting the malware files and then implementing the algorithm with the highest accuracy in developing an application which will take input in the form of files and give us the output of whether the files are malware infected or not.

**1.2 OBJECTIVE:**

* In this project we are trying to identify the malware infected program files.
* Usually malware detection is done through anti-virus software which compares the program to known malwares.
* But we are trying to detect malwares using Machine Learning Algorithms i.e by using the known features of malware and training a model to detect malwares.
* We are training the malware dataset using 3 ML algorithms:
* Classification Algorithm
* Clustering Algorithm
* XG Boost
* Depending on which model gives the highest accuracy in detecting malwares ; we would use that specific algorithm to make an application which can be used to detect malware -infected files.

**1.3 MOTIVATION**

* Earlier Malware detection was implemented using Apache Spark software.
* A malware dataset containing 100000 information and 34 features is taken and analyzed utilizing Spark.
* The toolkit incorporates execution for deep learning, factor machines, topic modelling, clustering, nearest neighbors.
* The equivalent malware dataset was additionally analysed utilizing Spark, an open source widely-used cluster-computing framework
* Spark is a quick and general engine for enormous information preparing, with inherent modules for streaming, SQL, AI and graph processing.
* It’s notable for its speed, usability, generality and the capacity to run virtually everywhere.
* The old model provides an accuracy of only 56-50% which is very less for a mall ware detection model.
* Moreover the old model works in Python 2.0 model which is going to stop functioning from this December 2020.
* The implementation of these algorithms was done using the programming language Python using scikit-learn. Scikit-learn is a free software which accounts for machine learning libraries for Python. It has different classification, regression and clustering algorithms and is designed to interact with digital libraries.

**2.1 INTRODUCTION**:

* Malware refers to malicious software which can damage or disable computers. Malware is created with the intent of stealing , encrypting or deleting sensitive information from our computer.It can also show down the performance speed of our computer ,browser speed or even create problems connecting to the internet.This is basically using the using and manipulation the systems without the owner/users permission. There are many different types of malwares such as computer viruses,computer worms,trojan horse,ransomware,rootkit and spyware.
* These malware needs to be detected so that the user gets an information that his system is getting corrupted and he can therefore take appropriate measures to prevent further attacks of malware and also he can take some steps to recover from the attacks.
* Machine learning has been recently introduced into the field of Malware Detection . Many algorithms have been used which results in differing accuracies in predicting weather the input files are malware or not.Many different algorithms like Apache Spark and TuriGraph Lab has been used to predict the malware infected files but the accuracy is less than 90% . Morever the algorithms have become non-existent in today’s world.
* In our project we are trying to train out dataset which consists of the features of both malware and non-malware infected files. We are training our dataset using 3 Machine-learning algorithms KNN Classification, Random Forest and XG Boost.
* The algorithm which gives the highest accuracy as output is used in developing an application which takes input as a set of infected and non-infected files and gives an output of the particular files which are containing malware.

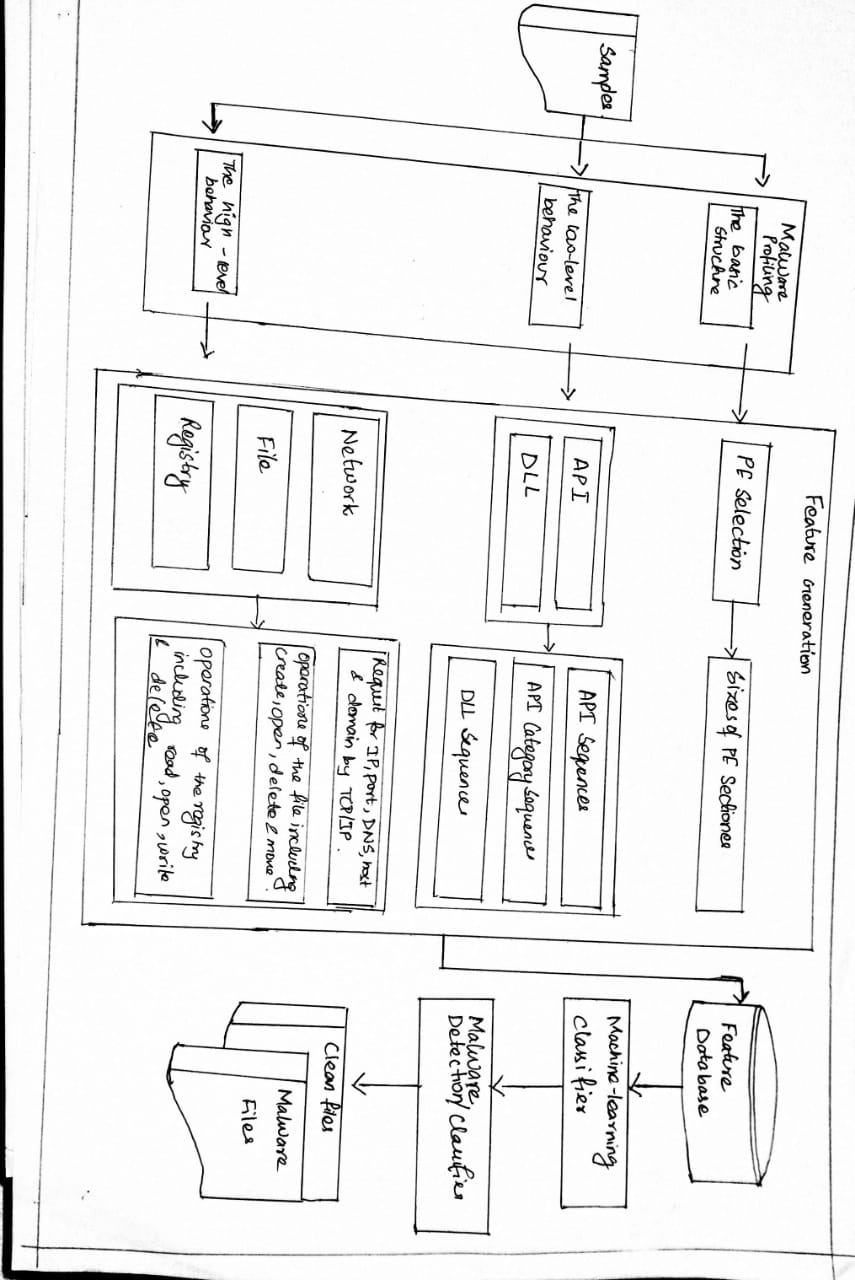
**3. INFORMATION SECURITY:**

* Information Security refers to a set of practises intended to keep the data secure from unauthorized access or interactions.
* Protects from unauthorised access, use, disclosure, disruption, modification, perusal, inspection, recording, or destruction.
* It is a part of Risk Management process.

The core function is to ensure the confidentiality, integrity and availability of data to the ‘right’ users within/outside of the organisation.

* Malware refers to malicious software which can damage or disable computers. Malware is created with the intent of stealing , encrypting or deleting sensitive information from our computer.It can also show down the performance speed of our computer ,browser speed or even create problems connecting to the internet.This is basically using the using and manipulation the systems without the owner/users permission.
* When owner of a system downloads a malware infected file , there is a huge risk that the confidential data contained may be breached or manipulated or even deleted . His activity on the system might also be monitored . This is a serious violation of the information security principles of ensuring that confidentiality and integrity of the data is maintained.
* Whenever we download some files we are clueless about whether it is malware infected or not . Anti-virus softwares mainly serve the purpose of detecting the malware infected downloads and alert us that the downloades files are corrupted.
* Our application detects the malware infected files using Machine Learning algorithms which is new method in the field of malware detection.
* The application gives us the freedom to check the files for malware whenever we want where-as antivirus softwares alert us only when some program files are newly downloaded.
* This also gives us the freedom to use it for a longer period without the fear of the application getting expired as in the case of the antivirus softwares.
* Thus our project helps to ensure that information security of an organization or an individual is maintained by detecting the malware infected files using the ML-algorithm which gives highest accuracy.

**3.1 ARCHITECTURAL DIAGRAM:**



**4. METHODOLOGY:**

**MODULES IMPLEMENTED:**

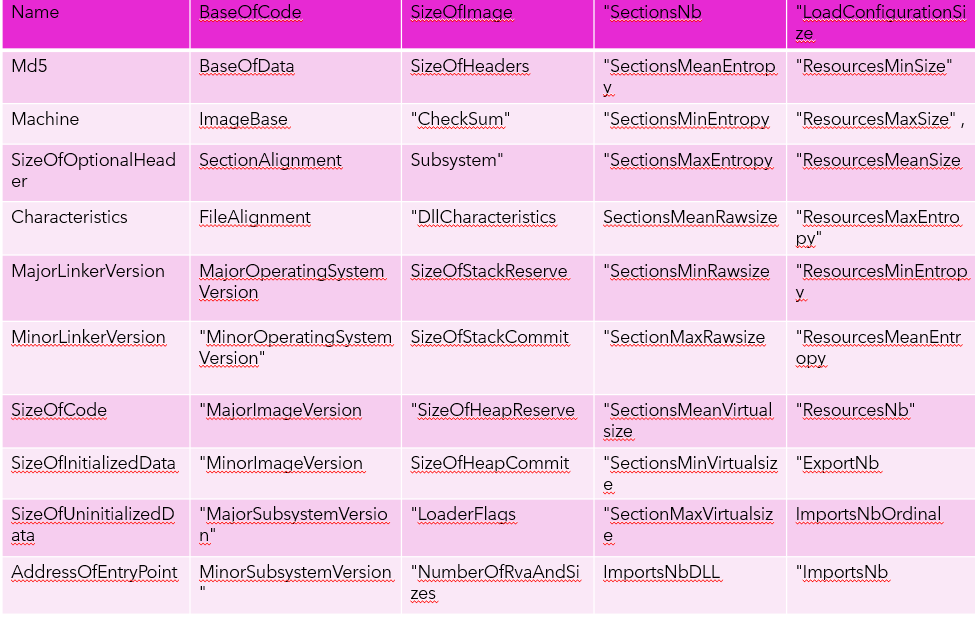
1.dataset preparation:

2. Training dataset using machine learning algorithms.

3. Deploying most accurate algorithm in the application.

**1.DATASET PREPARATION:**

* + In order to get started, we first need a set of data on which we can train our algorithms.
  + To create the data set we used file executable greenhouses infested with wax
  + We downloaded them from "https://virusshare.
  + The data set contains 10539 PE- files of which 6999 infested with malware and 3540 clean files.
  + There are 54 features in our dataset.



This is the list of major features and their importance.

1.MajorSubsystemVersion feature (0.155935) –

2. MajorOperatingSystemVersion feature (0.110355)

3. feature ImageBase (0.108349)

4. Feature Machine (0.068662)

5. feature DllCharacteristics (0.050951)

6. feature SectionsMaxEntropy (0.048521)

7. feature LoadConfigurationSize (0.038567)

8. ResourcesMaxEntropy feature (0.038000)

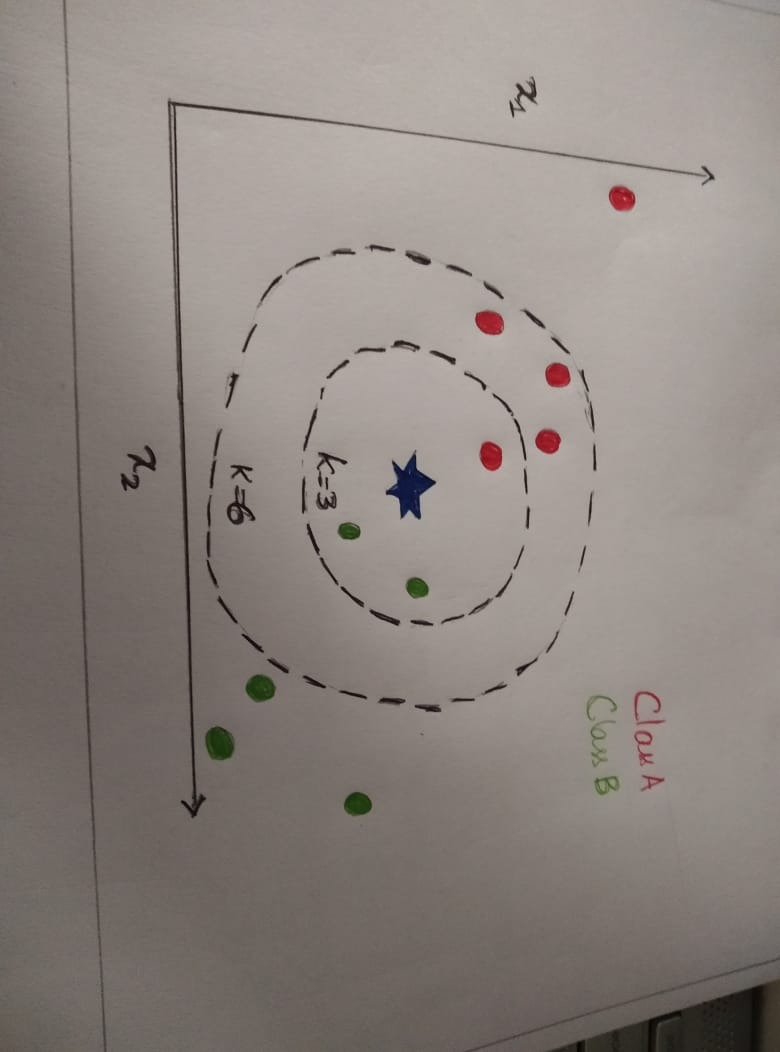
9. feature MajorLinkerVersion (0.032729)

10. feature ResourcesMinSize (0.022452)

**2. TRAINING DATASET USING MACHINE LEARNING ALGORITHMS:**

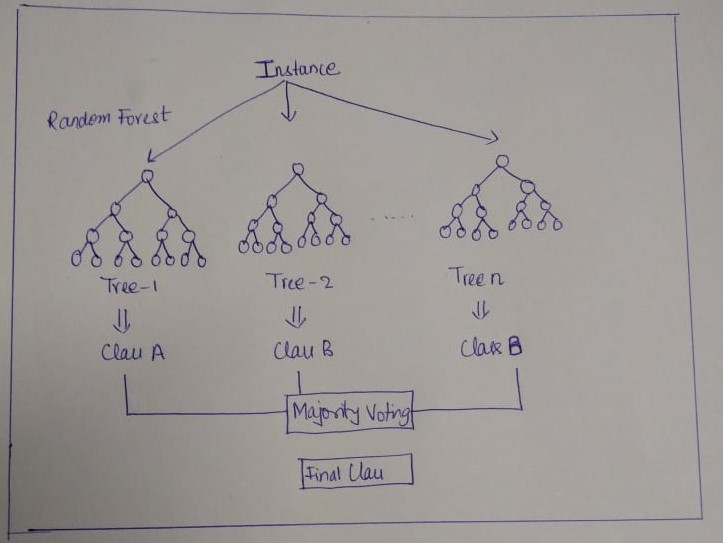
KNN-CLASSIFICATION ALGORITHM:

* K-NN K-Nearest Neighbour is one of the simplest Machine Learning algorithms based on Supervised Learning technique.
* K-NN algorithm assumes the similarity between the new case/data and available cases and put the new case into the category that is most similar to the available categories.
* K-NN algorithm stores all the available data and classifies a new data point based on the similarity.
* K-NN algorithm can be used for Regression as well as for Classification but mostly it is used for the Classification problems.



2. Classification Algorithm(Random Forest Algorithm) for Malware Detection:

* Takes a subset of data and clusters into subgroups
* Connecting datapoints to groups and subgroups we get a decision tree. The Algo then makes a group of decision tress called a forest.
* Remaining dataset used for predicting the tree which makes the best classification of datapoints is shown as output.
* Set of labels to determine the type of file where 1 represents malware and 0 represents clean files.

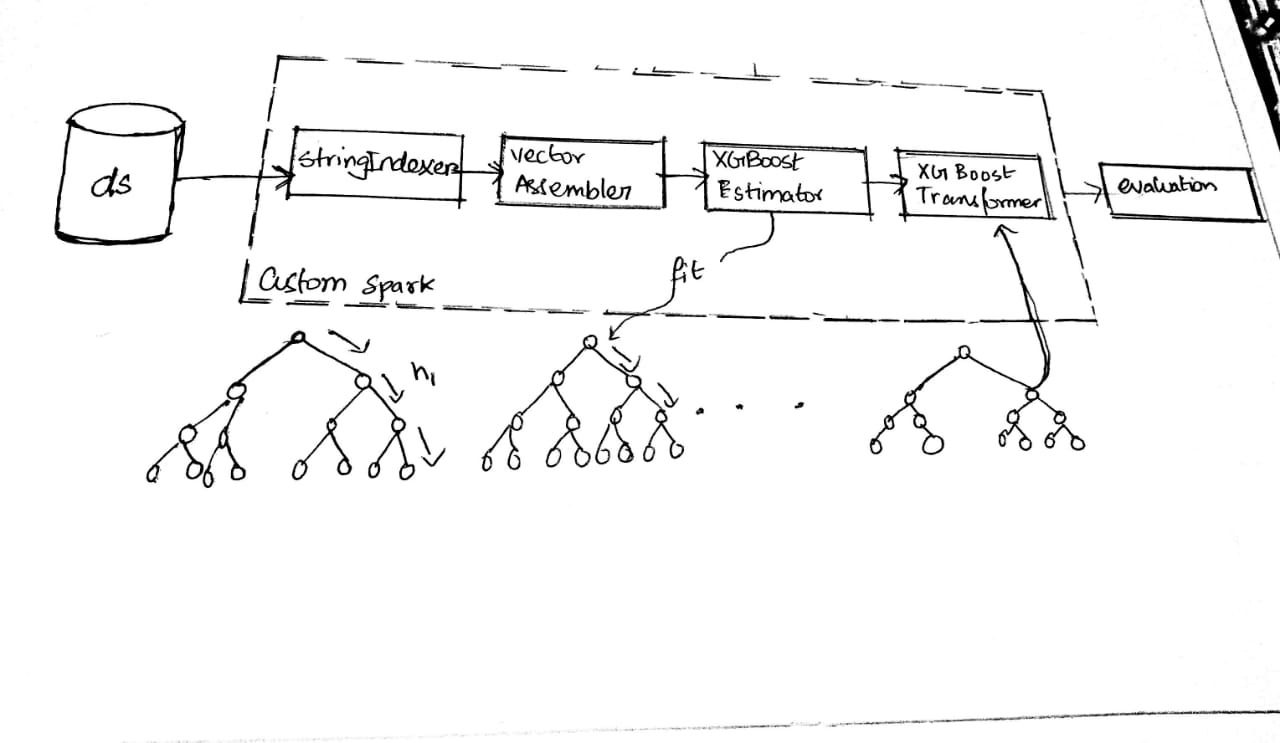


3. XG BOOST ALGORITHM:

* The main features that encourage the use of xg boost is
* Parallelization: XGBoost uses the method of sequential tree construction using parallel implementation.
* Tree Pruning**:** The stopping condition for tree splitting within GBM frame is greedy in nature and depends on the bad loss at the point of split
* Hardware Optimization: The algorithm is designed to achieve efficient use of hardware resources. This is achieved by cache awareness by allowing internal buffers in each thread to store gradient statistics.

The algorithm enhancements of XGBoost Algorithm:

* Regularization:
* It doesn't allow more complex models with both LASSO (L1) and Ridge (L2) rules to prevent overfitting.
* Sparsity Awareness:
* XGBoost gives best features for inputs by automatically learning best missing value depending on training loss and manages different types of sparsity patterns  in the data more efficiently.
* Weighted Quantile Sketch**:**
* XGBoost uses the shared weighted quantile sketch algorithm to directly find the optimal split points among weighted datasets.
* Cross-validation:
* The algorithm is given with built-in cross validation method at each iteration taking away the need to explicitly program this search and to give the exact number of boosting iterations required in a single run.

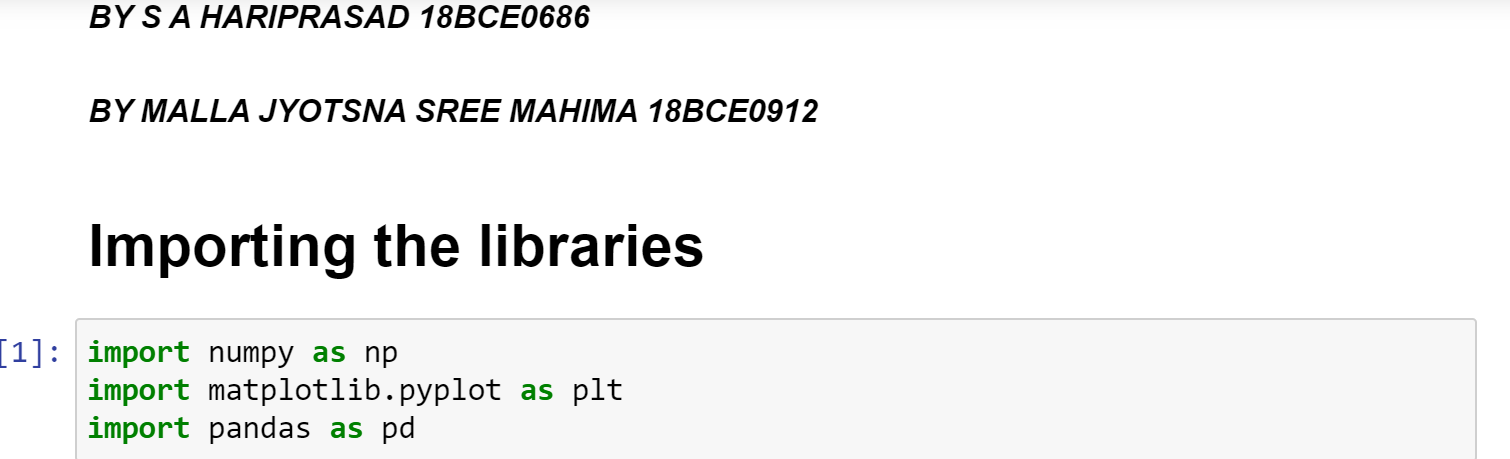


**3.DEPLOYMENT OF APPLICATION:**

After comparing the machine learning algorithms accuracies we choose the algorithm with the highest accuracy and use it for building the malware detection application.

**5.IMPLEMENTATION:**

**MALWARE DETECTION USING KNN CLASSIFICATION ALGORITHM:**



# 

# 

# 

# 

# 

# The confusion matrix is a specific table that allows the efficient visualization of an algorithm.

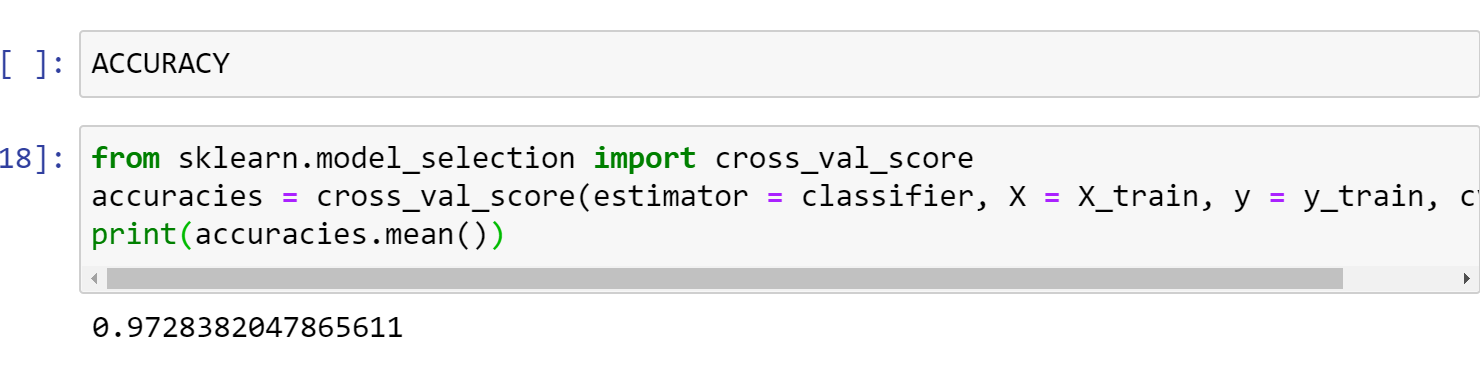
# • position 00 represents True Positives

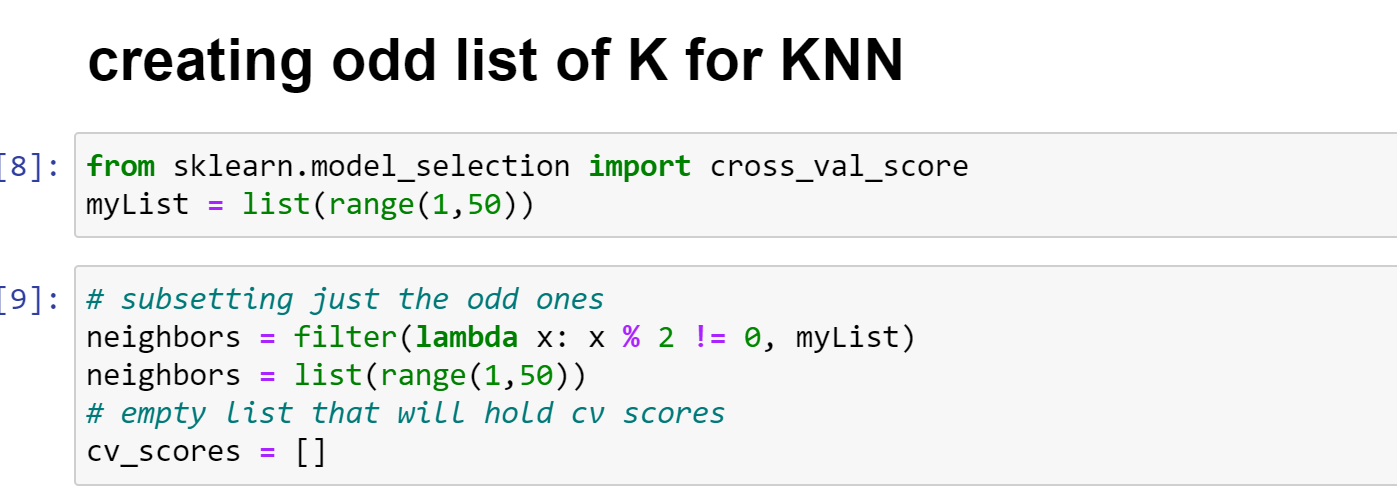
# • position 11 represents True Negatives

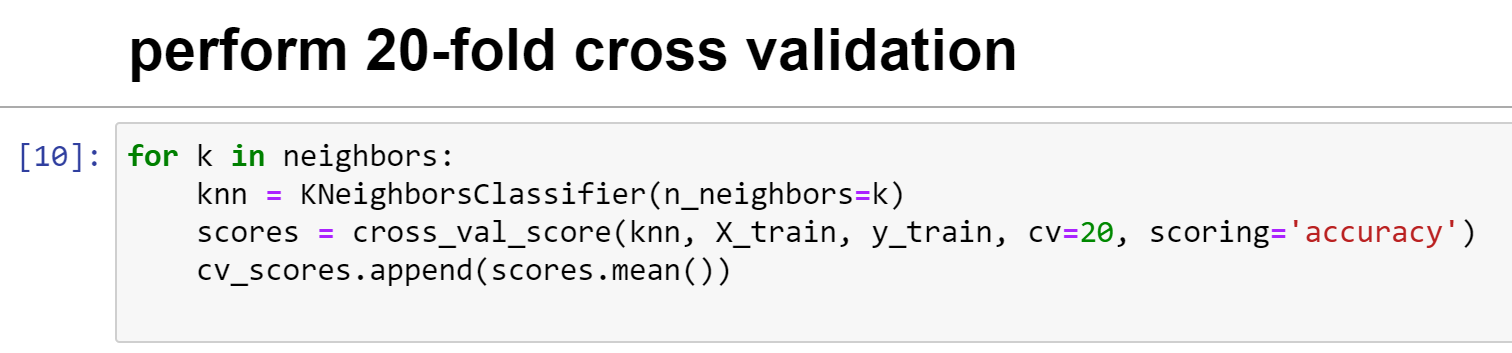
# • position 01 represents False Positives

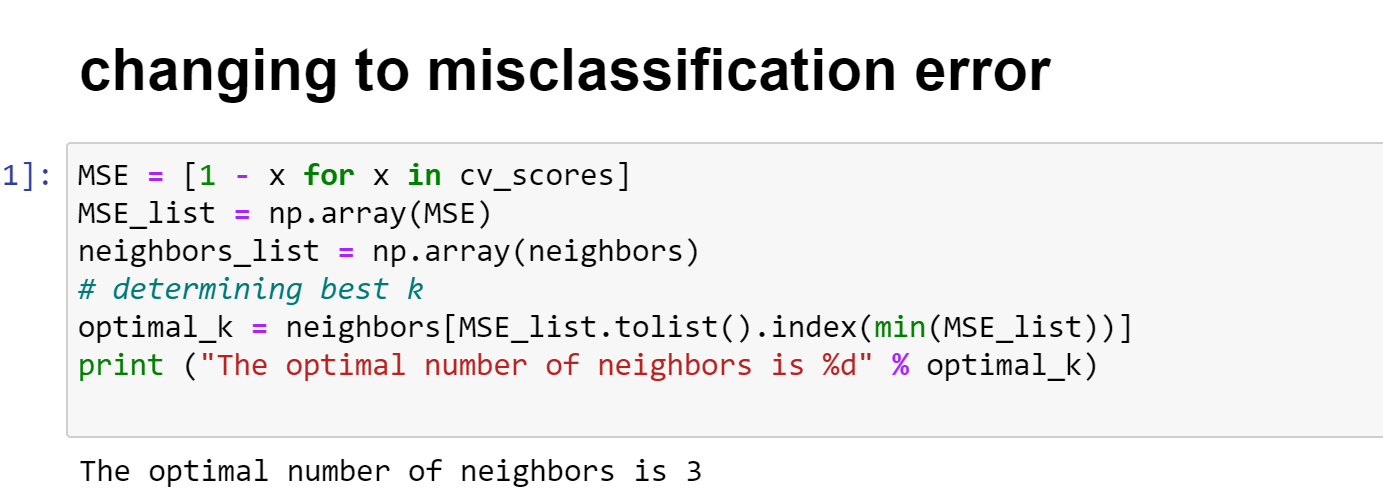
# • position 10 represents False Negatives

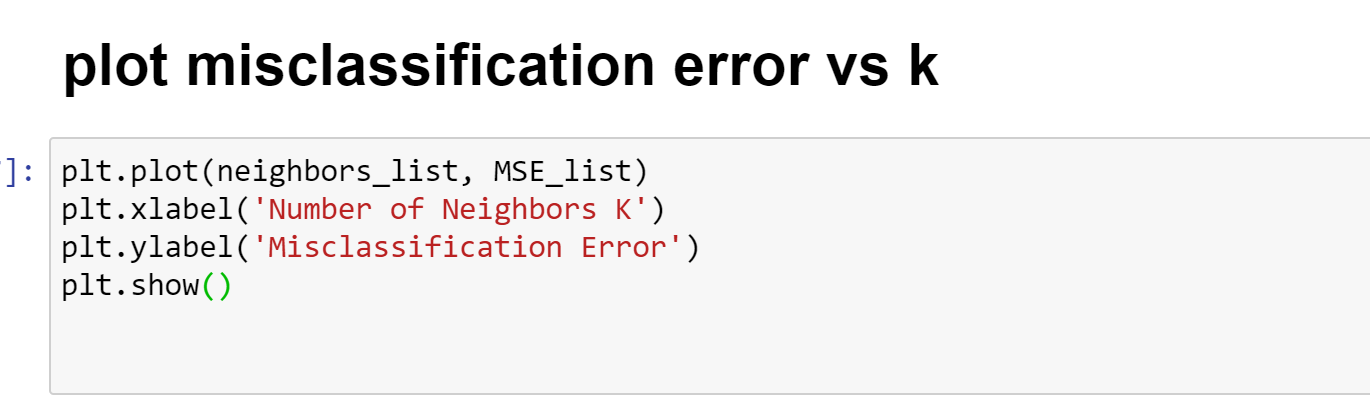
From what we can see from the confusion matrix our algorithm correctly predicted 2039 correctly and 69 are wrong predictions, which leads to an accuracy of 97.12%.

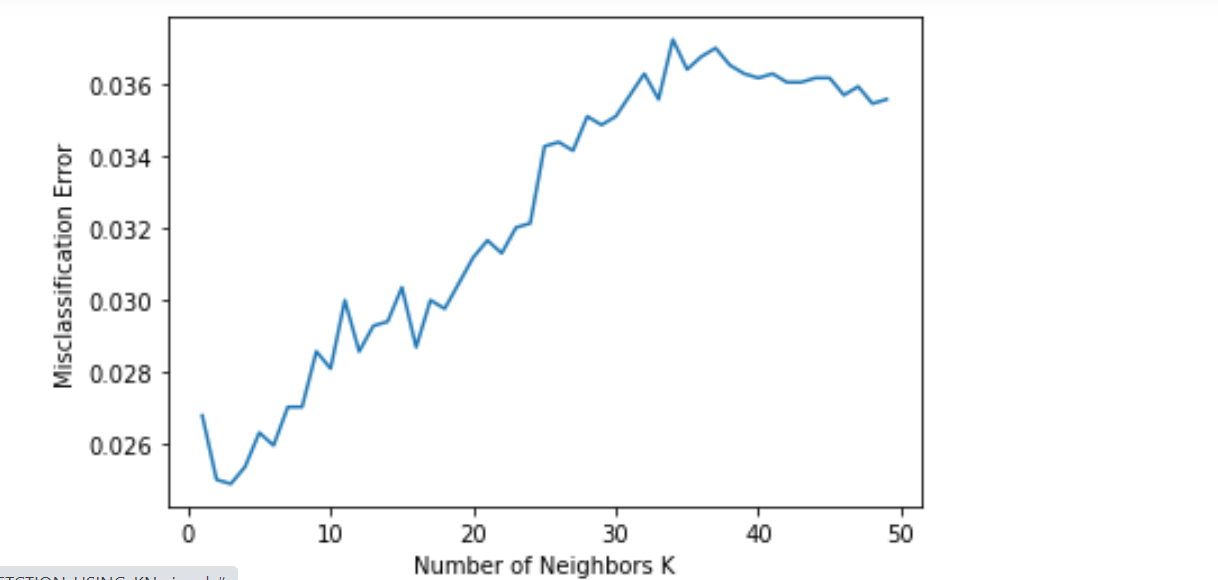




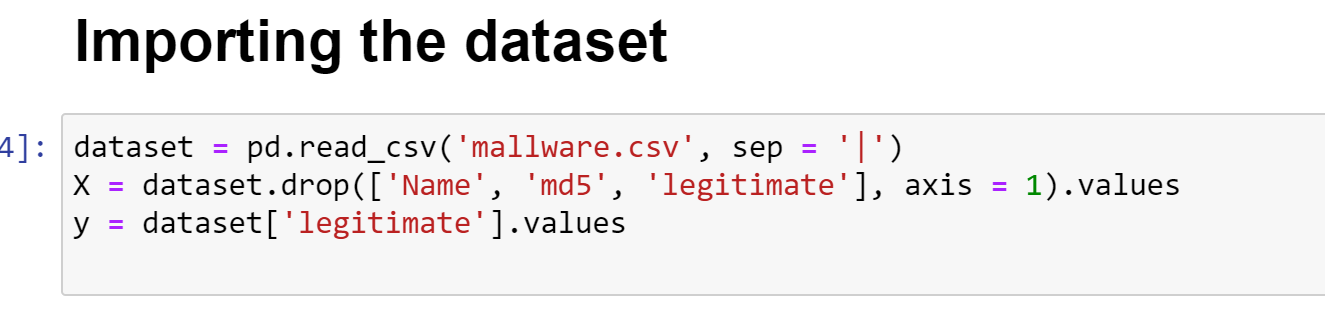


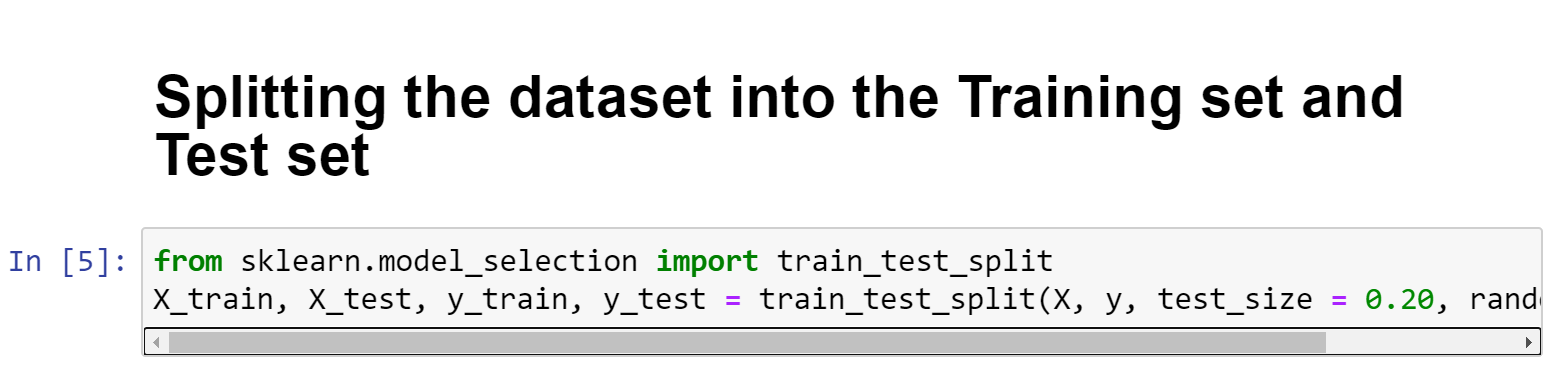


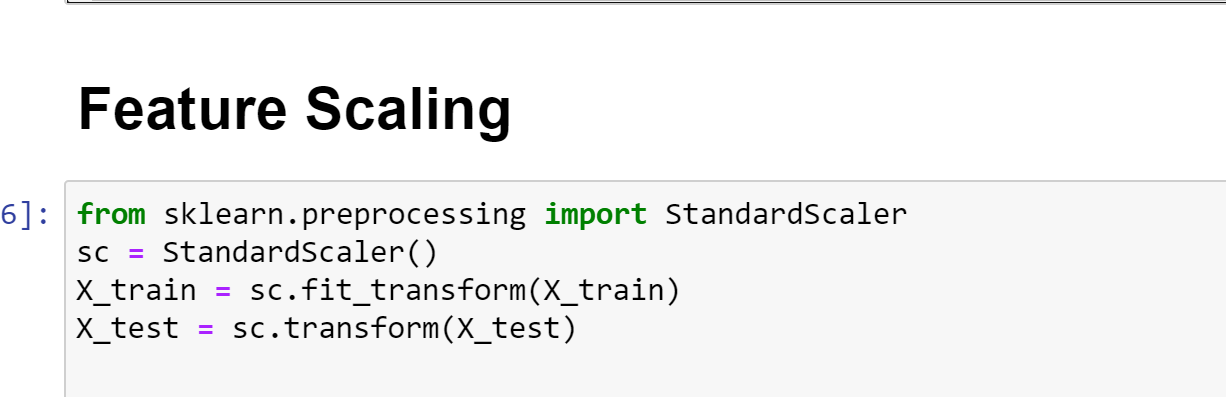


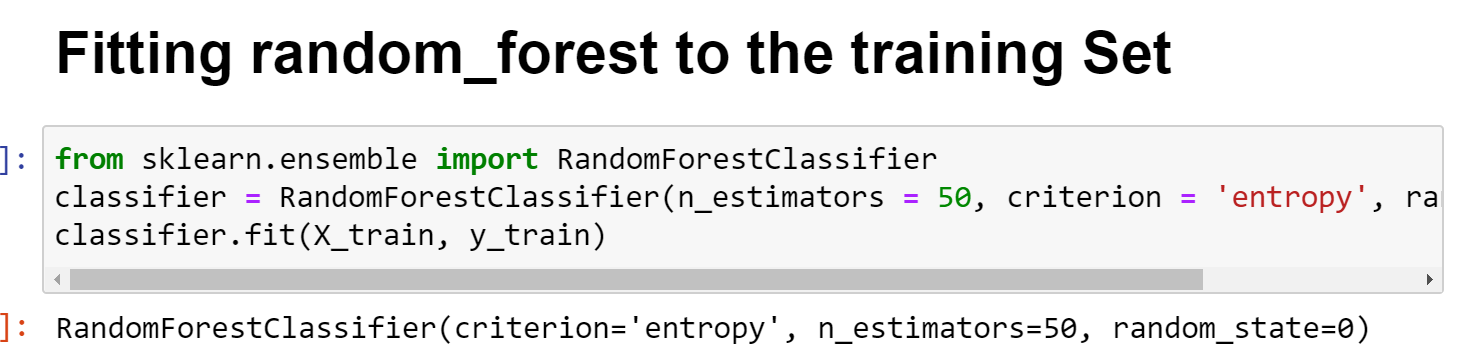


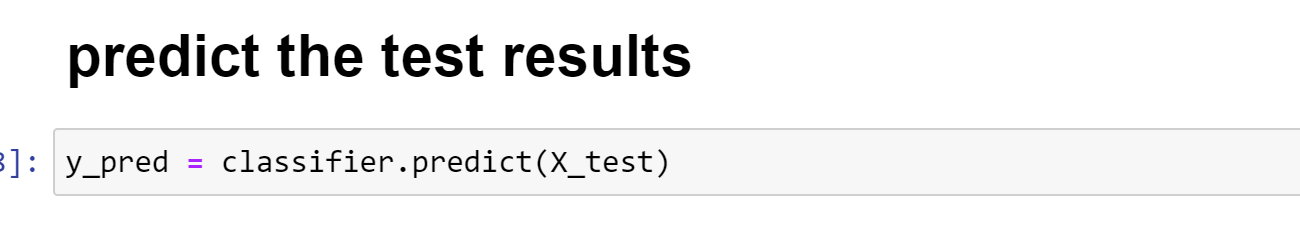
**2.RANDOM FOREST:**

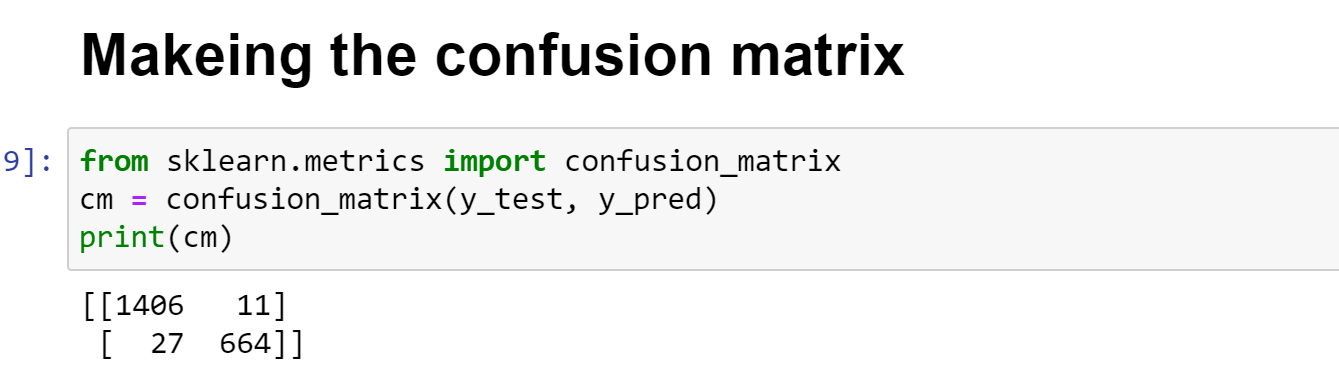


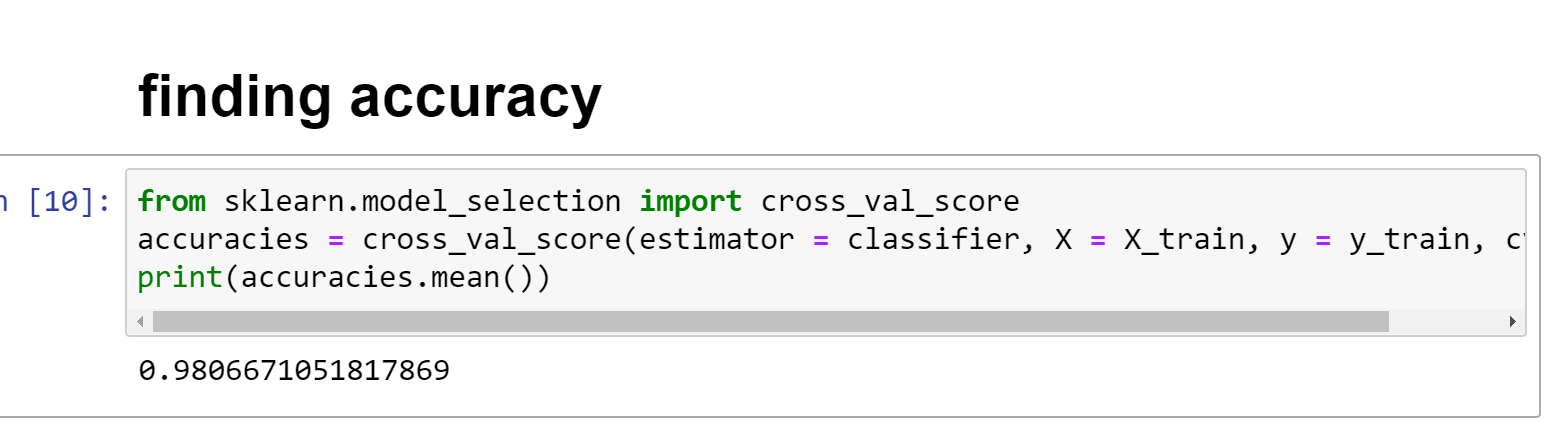




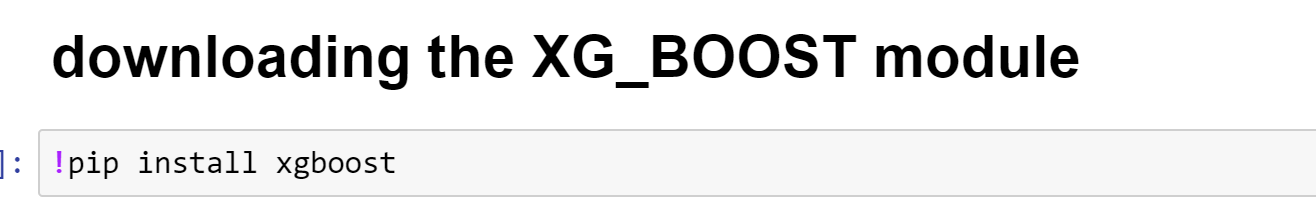






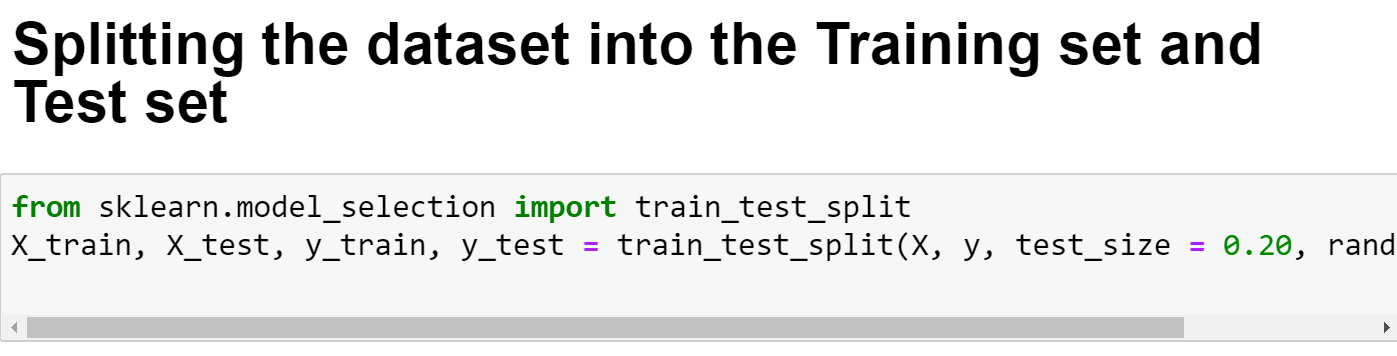


**3.XG BOOST ALGORITHM:**

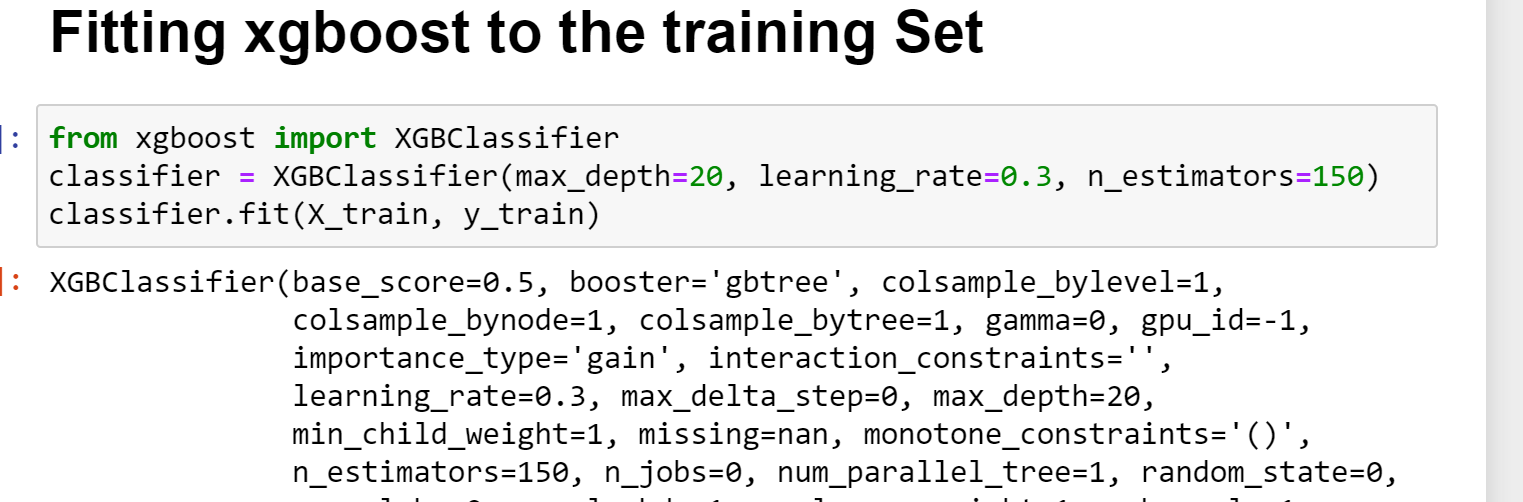


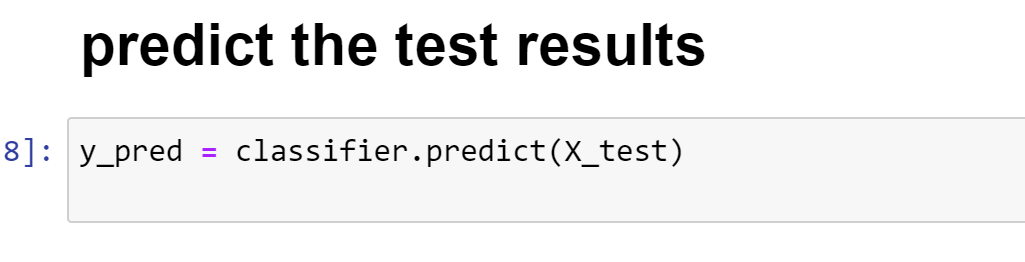


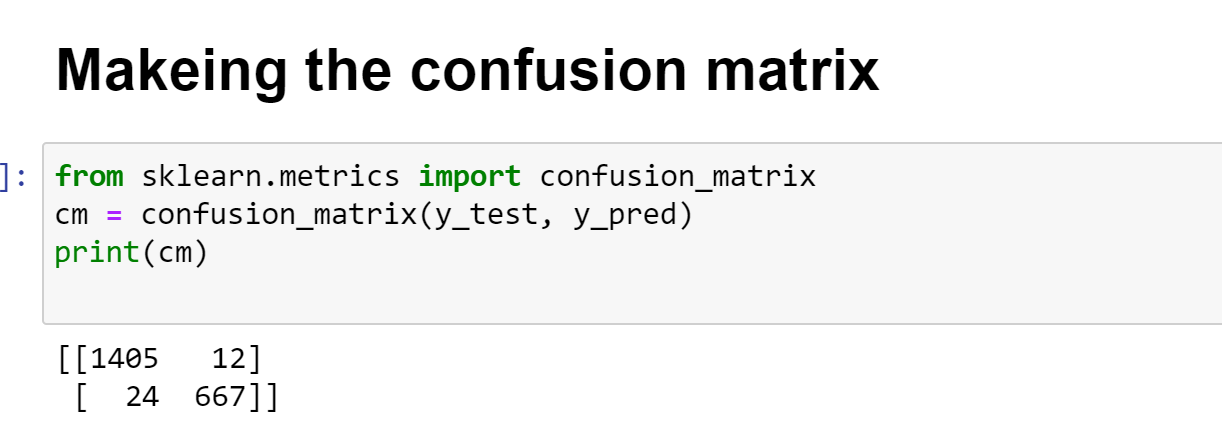


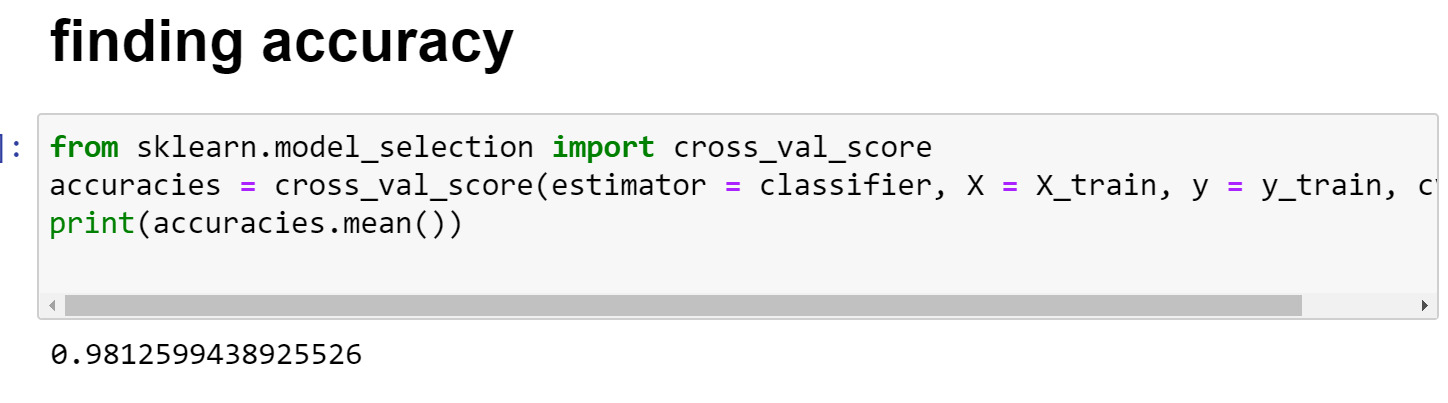




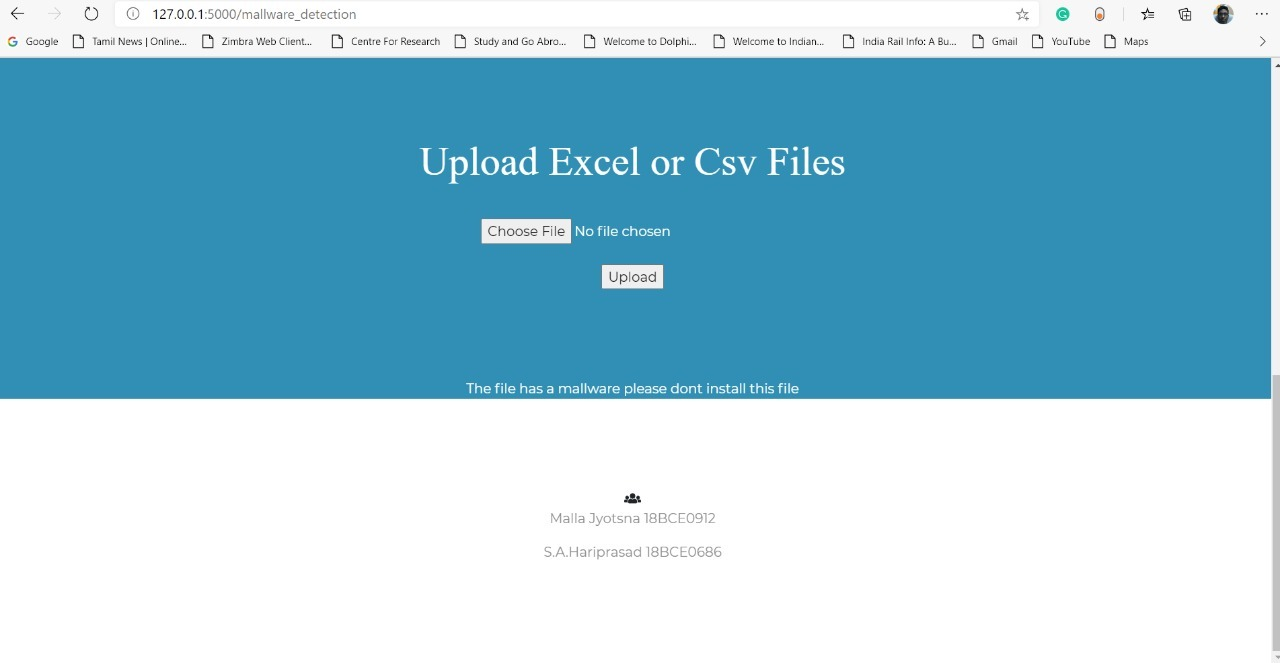


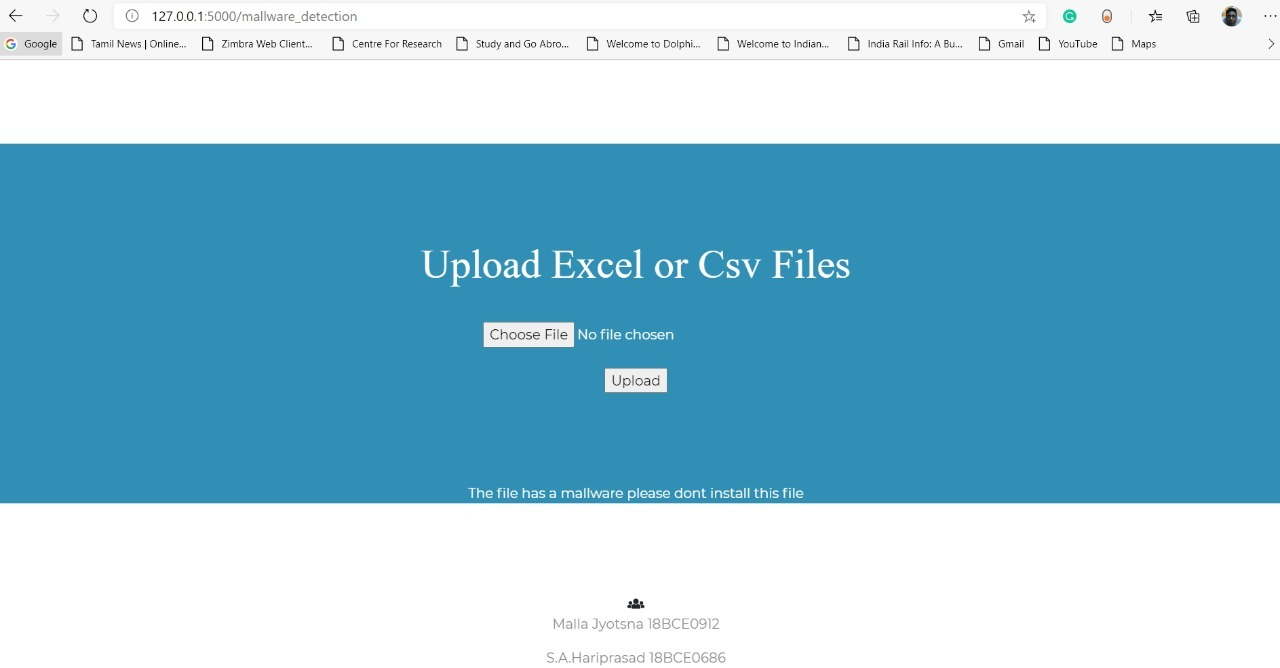






**APPLICATION BUILD FOR DETECTING MALWARE FILES:**





**6.CONCLUSION:**

We have seen that XG Boost Algorithm gives the highest accuracy in detecting the malware infected files . So an application is made which can be used by everyone to upload files and check if it contains malware files or not.So we have come up with a successful application which runs on the most accurate algorithm to detect malware infected files.

**7.REFERENCES:**

**References:**

Journals:

1. S. Gopalani and R. Arora, “Comparing apache spark and map reduce with performance analysis using k-means,” International journal of computer applications, vol. 113, no. 1, 2015.

2. I. Mavridis and H. Karatza, “Performance evaluation of cloud-based log file analysis with apache hadoop and apache spark,” Journal of Systems and Software, vol. 125, pp. 133–151, 2017.

3.Y. Low, J. Gonzalez, A. Kyrola, D. Bickson, and C. Guestrin, “Graphlab: A distributed framework for machine learning in the cloud,” arXiv preprint arXiv:1107.0922, 2011.

4.Y. Low, J. E. Gonzalez, A. Kyrola, D. Bickson, C. E. Guestrin, and J. Hellerstein, “Graphlab: A new framework for parallel machine learning,” arXiv preprint arXiv:1408.2041, 2014.

5.O. Segal, P. Colangelo, N. Nasiri, Z. Qian, and M. Margala, “Sparkcl: A unified programming framework for accelerators on heterogeneous clusters,” arXiv preprint arXiv:1505.01120, 2015.