# Prediction Of Machine Being Hit By Malware

A Report submitted to the Rajiv Gandhi University of Knowledge Technologies in partial fulfilment of the degree of

**Bachelor of Technology**

**in**

## Computer Science and Engineering

By

**Bajidbee Shaik (S160383)**

**Harika Kalivarapu (S160189)**

**Sowjanya Nizampatnam (S160505)**

**Rishitha Puli (S160419)**



SRIKAKULAM - 532402

Andhra Pradesh,India



**CERTIFICATE**

This is to certify that the report entitled “**Prediction Of Machine Being Hit By Malware**” submitted by **Bajidbee Shaik**,bearing ID. No. S160383 ,**Harika Kalivarapu** bearing ID. No. S160189,**Sowjanya Nizampatnam**,bearing ID. No.S160505,**Rishitha Puli**,bearing ID.No. S160419 in partial fulfilment of the requirements for the award of Bachelor of Technology in Computer Science is a bonafide work carried out by them under my supervision and guidance. The report has not been submitted previously in part or in full to this or any other University or Institution for the award of any degree or diploma.

Sesha Kumar Nalluri,

Project internal guide, Department of

Computer Science and Engineering, RGUKT,SRIKAKULAM.

M.ROOPA,

Head of the Department,

Department of Computer

Science and Engineering

RGUKT,SRIKAKULAM.

### DECLARATION

We **Bajidbee Shaik,Harika Kalivarapu,Sowjanya Nizampatnam and Rishitha Puli** hereby declare that this report entitled “**Prediction of Machine Being Hit by Malware**” submitted by us under the guidance and supervision of **Sesha Kumar Nalluri** is a bonafide work. We also declare that it has not been submitted previously in part or in full to this University or other University or Institution for the award of any degree or diploma.

Date : 02/05/2022

Place : Nuzvid

(Bajidbee Shaik)

ID No. S160383

(Harika Kalivarapu)

ID No. S160189

(Sowjanya Nizampatnam)

S160505

(Rishitha Puli) S160419

# Acknowledgement

We would like to express my sincere gratitude to **Sesha Kumar Nalluri**, our project Guide, for valuable suggestions and keen interest throughout the progress of our course of research.

We are grateful to **M.Roopa** ,HOD CSE, for providing excellent computing facilities and a congenial atmosphere for progressing with our project.

At the outset, We would like to thank **Rajiv Gandhi University of Knowledge and Technologies, Srikakulam** for providing all the necessary resources for the successful completion of our course work. At last, but not the least, we thank our classmates and other students for their physical and moral support.

With Sincere Regards,

## Bajidbee Shaik,

**Harika Kalivarapu, Sowjanya Nizampatnam, Rishitha Puli.**

# Abstract

In this modern technology , the danger of malicious attacks has increased a lot through the internet . These malicious attacks take place via malware for financial damage, accessing private networks, stealing sensitive data, taking over computer systems to make use of its resources, etc.[23].

So this makes the prevention of malicious attacks an important task to go against Cyber crimes . So , our project aims to predict the machines that are being affected by malware in the future . Predicting the malware will help us to increase security to our devices . We are applying machine learning algorithms to predict the malware infection rates of computers based on the user habits and working style with the systems [23].

We are using Gradient Boosting framework as a reference, namely Light GBM, to build a model that predicts whether a system will soon be hit with malware . We have taken some parameters based on the user habits and system features and system condition . These parameters are given as input to our model . We have also assigned some weights to each parameter to predict the probability .

**Keywords :**

Malicious attacks , internet , prevention , Cyber crimes , prediction , security , Machine learning , Gradient Boosting framework , parameters .

# Contents

**Acknowledgments iii**

**Abstract iv**

1. **Introduction**
   1. Introduction 1
   2. Applications
   3. Problem Statement 10
   4. Organisation of Report 10
2. **Literature Survey 11**
   1. Malware Detection 11
      1. What is Malware 11
      2. Types Of Malware 11
      3. How A Malware Can Be Detected 12
   2. Microsoft Malware Prediction 14
   3. Machine Learning 14
3. **Light Gradient Boosting Algorithm 16**
   1. Introduction 16
   2. Dataset Collection 18
   3. Installation of Anaconda 19
   4. Weight allocation to parameters. 22
4. **Implementation and results** 24
   1. Procedure 24
   2. Implementation 25
      1. Open jupyter notebook through CMD 25
      2. Import Packages 25
      3. Training the model 26
      4. Testing the model 31
   3. Results 42
5. **Conclusion and future work** 43
6. **Bibliography** 44

# List of Figures

**Fig[1] : Malware Types 12**

**Fig[2] : Malware Detection Flow Chart 13**

**Fig[3] : Life-Wise Tree Growth Architecture 18**

**Fig[4] : Anaconda installation Step 1 20**

**Fig[5] : Anaconda installation Step 2 20**

**Fig[6] : Anaconda installation Step 3 21**

**Fig[7] : Reading train csv file 26**

**Fig[8] : No.of columns 26**

**Fig[9] : Count of each parameter 27**

**Fig[10] : Parameter division 27**

**Fig[11] : applyMap function 28**

**Fig[12] : fillNA function (1) 28**

**Fig[13] : fillNA function (2) 29**

**Fig[14] : Numerical columns method (1) 29**

**Fig[15] : Categorical columns method (1) 30**

**Fig[16] : Categorical columns method (2) 30**

**Fig[17] : Reading Test csv file 31**

**Fig[18] : fillNA function (3) 31**

**Fig[19] : fillNA function (4) 32**

**Fig[20] : Numerical columns method (2) 32**

**Fig[21] : Categorical columns method (3) 33**

**Fig[22] : Test clean csv file 33**

**Fig[23] : Reading test clean file 34**

**Fig[24] : Feature division 34**

**Fig[25] : High feature weights 35**

**Fig[26] : Medium and low feature weights 35**

**Fig[27] : High feature weight allocation 36**

**Fig[28] : High feature weight allocation (1) 37**

**Fig[29] : Medium feature weight allocation 38**

**Fig[30] : Medium feature weight allocation (1) 39**

**Fig[31] : Low feature weight allocation 40**

**Fig[32] : Probability calculation 41**

**Fig[33] : Result (1) 42**

**Fig[34] : Result (2) 42**

# Chapter 1 Introduction

## 1.1 Introduction

Malware prediction is one of the important steps in the security of computer systems. Along with advancement of technology anti-malware software industries receive a massive number of malware pirated files to be examined[2]. However, currently used signature-based methods are unable to provide accurate prediction of zero day attacks. The dark world hackers are using them to lure into systems through the points mentioned in the vulnerability databases. Hence, it is highly necessary to predict the malware at an early stage to avoid further loss[2]. That’s why machine learning based malware prediction arises. The objective of the project work predicts a computer that is being hit by the malware in the future based on the user actions and system condition . That helps billions of machines from damage before it happens.

Malware, or malicious software, is software created to infect a machine without the users knowledge or consent. It is actually a generic definition for all sorts of threats that can affect a computer. The objectives of a malware could include accessing private networks, stealing sensitive data, taking over computer systems to make use of its resources, or disrupting computing or operations.

The objective of our project is to create the model that provides the highest accuracy in predicting if a machine has a chance to be infected by malware. It is done by taking into account the different parameters on the machines, especially ones related to the machines malware protection status, and user habits.

## 1.2 Applications

* Data security on the computer .
* Awareness to the user .

## 1.3 Problem Statement

As we know , there are various families of malware and each malware has its own functionalities and will be injected into a system in different ways . So Predicting a machine which may have a chance of malware attack is needed , to reduce the infection rate and damage to data on the system . We took this problem and we studied various papers to find a solution for it . In the end we decided to build a model that is trained to provide the predicted probability value of a machine which will soon be infected by various families of malware based on the different properties of that machine and user habits .

## 1.4 Organization of Report

The rest of this thesis is organized as follows: Chapter 2 gives a literature survey about Malware and machine learning. Chapter 3 is about the Light Gradient Boosting method. Chapter 4 deals about procedure,implementation and results. The conclusion and the future scope of the work is given in Chapter 5.

**Chapter 2**

# Literature Survey

## 2.1 Malware Detection

### 2.1.1 What is Malware

Malware can be defined as a malicious software which is created to enter into someone's systems and to gain access to the data and to modify the data in the system . The malware can be injected into a system in many ways . The attacker tries to identify the loopholes and if he finds any , a malware file will be injected through a text message or E-mail or any fake link etc.. Malware is created to harm the devices and user data . There are vast numbers of malware and each has its own goal and will attack differently for different purposes . No matter what the type of malware and what its purpose , the attackers will just aim to steal the user data and to get some ransome . Each malware has its own unique identity and that identity is now known by some anti-malware software . So if we have anti-malware software in our system then it will show us what files are attacked by malware and also show the malware files present in our system by checking their unique identity .

Users need to be careful whenever downloading any file online . There may be a chance that the file is malicious . So if that file is downloaded then it will be easy for the attackers to inject the malware into our systems and will steal our data .

### 2.1.2 Types Of Malware



**Fig[1] : Malware types**

**1.Adware**

Adware, or advertising supported software, is software that displays unwanted advertisements on your computer.

**2.Rootkit**

A rootkit is software used by cyber criminals to gain control over a target computer or network . Rootkits can enter computers when users open spam emails and inadvertently download malicious software .

**3.Ransomware**

Ransomware, as the name indicates, is a type of malware that comes with a ransom. It locks and encrypts a victim’s device or data and demands a ransom to restore access .

**4.Spyware**

Cyber criminals use spyware to monitor the activities of the user . By logging the keystrokes a user inputs throughout the day, the malware can provide access to user names, passwords and personal data . **5.Trojan-horse**

A Trojan horse virus is a type of malware that downloads onto a computer disguised as a legitimate program .

**6.Phishing**

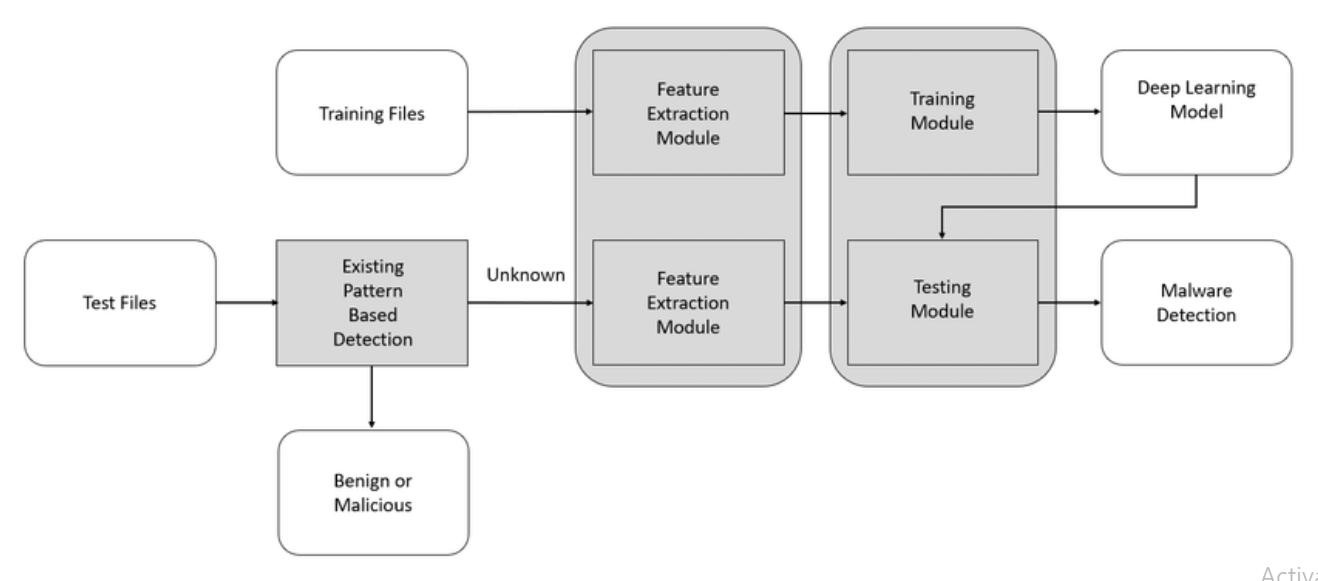
Phishing attacks are the practice of sending fraudulent communications that appear to come from a reputable source . It is usually done through email .

**7.Worms**

A computer worm is a type of malware that spreads copies of itself from computer to computer . **8.Viruses**

A computer program that can copy itself and infect a computer without permission or knowledge of the user . It is a type of malicious code or program written to alter the way a computer operates and is designed to spread from one computer to another .

### 2.1.3 How a malware is detected



**Fig[2] : Malware Detection Flow Chart**

A malware can be detected through the unique signature of the malware . Each Malware has its own unique signature . Malware detection can only be done by using the Malware signatures . Anti-malware software working principle is signature based only . The files in our system can checked through anti-malware software , this software will detect which file in our system is malicious file by checking the signatures of the previously detected malware and the files in our system .

In 2001 , detection of unknown malware based on static features is introduced for feature Extraction . In 2007 , to examine opcodes frequency distribution in malicious and non Malicious files , opcode as a malware detector is introduced[26] .

Decision trees are used in Bayesian Network and that gave 95.8 % accuracy .In the year 2008

Filters approach is used for feature selection in Artificial Neural Networks (ANN) .Some of the Algorithms such as Decision trees, Naive Bayes,Support Vector Machine classifiers gave nearly

94.9 % accuracy .

Recently some authors worked on a malware dataset released for kaggle dataset . In the year 2016, one of the author took Microsoft malware dataset and used hex dump-based features ( n-gram , Metadata , entropy , image representation and string length ) as well as features extracted from disassembled file ( Metadata , Symbol frequency , opcodes , etc. ) and XGBoost classification algorithm. They reported 99.8 % detection accuracy[26].

## 2.2 Microsoft Malware Prediction

Microsoft malware prediction is the problem statement which is raised in the competition that Is conducted by Kaggle . Kaggle took this problem statement because malware attacks are rising day by day . In order to get security and integrity of data of users and organizations , Microsoft tried to make a prediction of malware . According to survey , Microsoft Windows os is ruling in the market , so now the Microsoft is facing the biggest challenge to provide security to their systems . So they took a challenge to provide accurate and efficient results whether the Machine will be infected with malware or not to keep the system data secured and remain the Systems non-vulnerable from malware attacks .

As a part of their overall strategy, Microsoft challenged the data science community to develop the machine learning systems which can determine or predict if the machine will be soon hit by malware .

## 2.3 Machine Learning

Machine Learning (ML) can be defined as that it is the field of study that makes computers be able to learn without being explicitly programmed . It is the sub field of Artificial Intelligence (AI) . It was first coined by Arthur Samuel in 1959 .

Machine Learning is the process of training an ML model in which that model can perform operations on newly given input data . When compared to other fields of study , if any new input is added , the programmer or developer has to create software to handle the new case . In Machine Learning , it is different and is easy .

The Machine Learning algorithms are used to create an ML model based on the user input . These ML models can be trained and are tested to generate an output that gives a particular decision or prediction . Different machine learning methods have been proposed to address the problem of predicting malware attacks .Light Gradient Boosted Machine (LGBM) is the most popular classification technique currently used in detecting and predicting the malware .

**Chapter 3**

# Light Gradient Boosting Algorithm

# 3.1 Introduction

Light GBM is a gradient boosting framework that uses tree based learning algorithms.

It is used to increase efficiency of the model and reduce the memory usage.

**Advantages of Light GBM:**

1)Faster training:

It uses histogram based algorithms which fastens the training procedure.

2)Lower memory usage:

It replaces continuous values which result in lower memory usage.

3)Better Accuracy than any other boosting algorithm:

It gives better accuracy than other algorithms like Random forest and Conventional Neural Networks.

4)Compatibility:

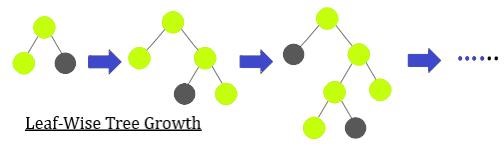
It is capable of handling larger datasets.

5)Parallel learning supported:

In light gbm, it supports Level-wise tree growth.

**Architecture :**

LightGBM splits the tree leaf-wise as opposed to other boosting algorithms that grow tree levelwise. It chooses the leaf with maximum delta loss to grow. Since the leaf is fixed, the leaf-wise algorithm has lower loss compared to the level-wise algorithm. Leaf-wise tree growth might increase the complexity of the model and may lead to overfitting in small datasets. Below is a diagrammatic representation of Leaf-Wise Tree Growth[25]:



**Fig[3] : Life-Wise Tree Growth Architecture**

**Light GBM Classifier:**

It is a fast, distributed, high-performance gradient boosting framework based on decision trees , used for classification,regression and many other learning tasks.

In our project, we’ve used Light GBM model for reference purpose.

**3.2 Dataset Collection:**

**System parameters:**

1. **Up to date :** Check if the system is up to date or not.
2. **Online/offline** : User working offline or online .
3. **Files are missed or Not :** Missing files without an intervention of the user**.**
4. **OS Genuine or Not :**Is the user using genuine OS or pirated OS.
5. **Firewall :** System firewall is active or not while using the system.
6. **Working Hours :** How many hours a user is using the system.
7. **Working Time :** At which time the user is using the system mostly like morning, evening , Afternoon.

**8. Anti Virus :** Antivirus is enabled or disabled.

**9. Default Browser :** Which browser you are using as default browser.

**10. Accept cookies :** Do you accept cookies while using any website?

**11. Allow Unnecessary Permissions :**Do you Allow permissions when installing the third party applications.

**12. New tools are appeared :** Without any intervention of the user new tools are appeared on toolbar and system or not.

**13. USB Connected :** While using the system is it connected to USB or not.

**14. Unrecognized icons :** Do you see icons on screen that you don’t know.

**15. Unknown Logins :** Anyone logged into your system without your knowledge.

**16. Unknown Programs are running :**Multiple programs are running in background while downloading or when you start the system .

**17. Os version :** Which version of OS is the user using.

**18. Performance good or bad :** whether system condition is good or bad while using .

**19. Spam mails :** Did you get spam mails through mail?

**20. Public or Private :** Whether connected to a public or private network while using the system.

**21. Advertisement :**Did you click any advertisements while using the system.

**22. Unknown Apps :** Unknown apps are installed or not without user knowledge.

**23. Unknown Files :**Any unknown files are downloaded without user knowledge.

**24. Default Search Engine changed or Not :**Default search engine is changed or not Without user knowledge.

**25. Unknown Activities like password changes :** Did Your password changed without Your knowledge.

**26. Unusual Network :** Whether an app running in the background is downloading data or not.

**27. Browser redirecting :** Does your browser redirect to other pages.

**28. Internet Traffic :** Did you face internet traffic (server busy) anytime.

**29. Control panel :** Did you face any problems while accessing the control panel.

**30. Unusual emails :** Changing the name of the sender that does not match with the email address.

**31. Intrusive pop-ups :** Do you get pop-ups that stop you from using the site?

**32. Changes in Home page :**Do you see any changes on your home page without your knowledge.

**33. Open source :** Whether you are downloading files from open source or not.

**34. Out dated apps using or Not :** whether you are using updated apps or not.

**35. Frequently Crashes :** Does your system crash frequently?

**36. Unknown Url :** Whether user clicks unknown urls or not.

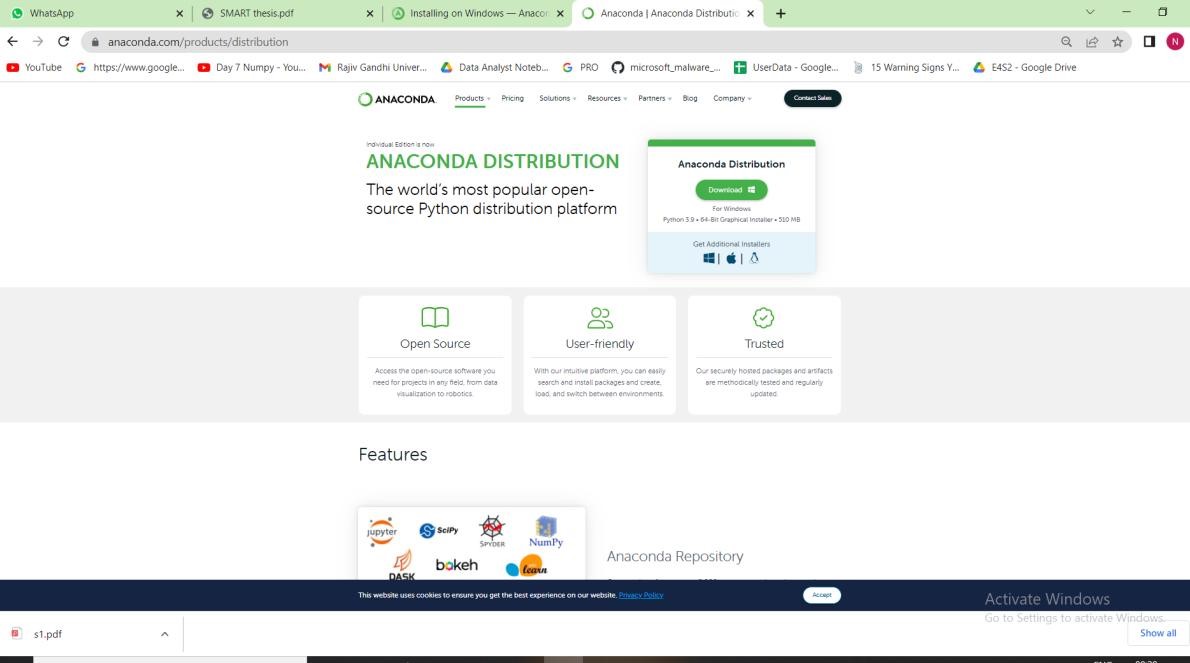
**37. Machine ID :** It is a unique identifier to identify each machine.

**38. Default search engine :** By default which search engine you are using.

## 3.3 Installing Anaconda

Installing Jupyter Notebook Using Anaconda. Anaconda is an open-source software that contains Jupyter, spyder, etc that are used for large data processing, data analytics, heavy scientific computing. Anaconda works for R and python programming languages. Jupyter Notebook is an open-source web application that allows you to create and share documents that contain live code, equations, visualizations, and narrative text. Uses include data cleaning and transformation, numerical simulation, statistical modeling, data visualization, machine learning, and much more. Jupyter has support for over 40 different programming languages and Python is one of them. Python is a requirement (Python 3.3 or greater, or Python 2.7) for installing the Jupyter Notebook itself. In order to install Jupyter using Anaconda, Please follow the following instruction[24]:

### 1 .Install Anaconda



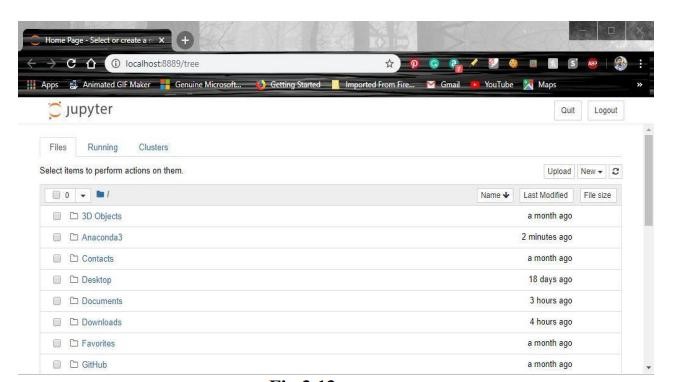
**Fig[4] : Step 1**

**2. Download .exe File.**



**Fig[5] : Step 2**

### 3. After installation click on Launch Jupyter Notebook



**Fig[6] : Step 3**

## 3.4 Weight allocation to parameters

We have taken the parameters to do a probability function on those parameters . So we have divided the parameters into 3 sections based on their weightage to consider . The parameters are divided as High weightage parameters , Medium weightage parameters and low weightage parameters . The weights are as follows :

**High Weightage Parameters : 0.6**

|  |  |  |
| --- | --- | --- |
| 1)online/offline | - | 0.055 |
| 2)Up to date | - | 0.035 |
| 3)Firewall | - | 0.04 |
| 4)Default Browser | - | 0.03 |
| 5)Spam mails | - | 0.055 |
| 6)public or private | - | 0.06 |
| 7)Os version | - | 0.03 |
| 8)Advertisement | - | 0.04 |
| 9)Unknown urls | - | 0.04 |
| 10)Open source | - | 0.035 |
| 11)Working Time | - | 0.03 |
| 12)Frequently Crashes | - | 0.03 |
| 13)default search engine | - | 0.025 |
| 14)USB Connected | - | 0.03 |
| 15)Allow Unnecessary Permissions | - | 0.04 |
| 16)Intrusive pop-ups  **Medium Weightage parameters : 0.3** | - | 0.025 |
| 1)Unknown programs are running | - | 0.01 |
| 2)OS Genuine or not | - | 0.025 |
| 3)Files are missed or not | - | 0.02 |

|  |  |  |
| --- | --- | --- |
| 4)New tools are appeared | - | 0.025 |
| 5)Unknown Files | - | 0.02 |
| 6)Unknown Apps | - | 0.025 |
| 7)Unknown Logins | - | 0.025 |
| 8)Anti Virus | - | 0.03 |
| 9)Accept cookies | - | 0.04 |
| 10)Browser redirecting | - | 0.02 |
| 11)Unusual emails | - | 0.02 |
| 12)Changes in Home page | - | 0.02 |
| 13)Default Search Engine changed or not  **Low Weightage parameters : 0.1** | - | 0.02 |
| 1)Unrecognized Icons | - | 0.015 |
| 2)Performance good or bad | - | 0.025 |
| 3)Internet Traffic | - | 0.015 |
| 4)Control panel | - | 0.02 |
| 5)Out dated apps using or not | - | 0.01 |
| 6)Working Hours | - | 0.01 |
| 7)Unknown Activities like password change - | | 0.01 |
| 8)Unusual Network - | | 0.005 |

**Chapter 4**

# Implementation and Results

## 4.1 Procedure

To implement Malware prediction it involves two steps. First we have to train the model and test the model with data. For training the model we need to train the model by dataset we have manually collected from users about system features. We need to import all the Necessary packages such as numpy,dataframe. We have to pass all the parameters to model through the train.csv file.

The train.csv will be given as input to the model. Next high\_features, medium\_features and low\_features are divided and stored as lists. Assigned weights for features based on their importance. Implemented a function calculate\_prob for calculating probability of machines being hit by malware.

## 4.2 Implementation

**4.2.1 Open jupyter notebook through cmd**

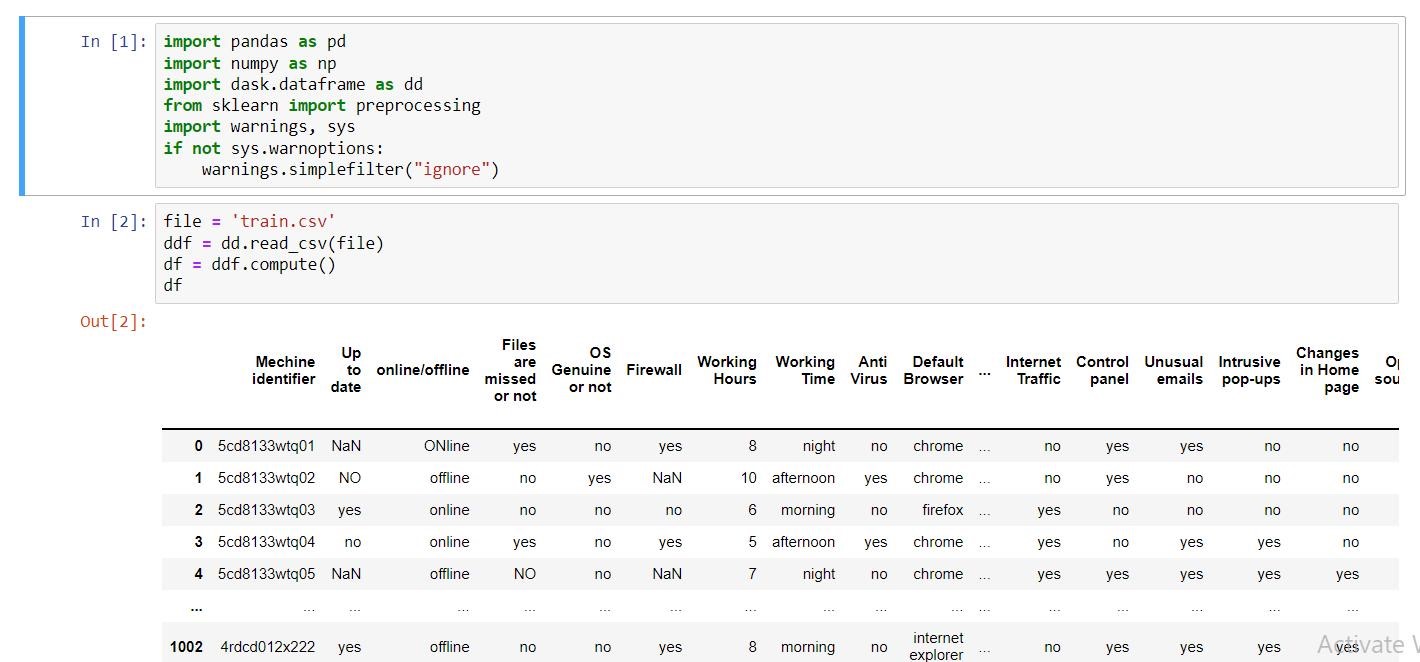
Python -m notebook

### 4.2.2 Import packages

import pandas as pd import numpy as np import dask.dataframe as dd from sklearn import preprocessing import warnings, sys if not sys.warnoptions:

warnings.simplefilter("ignore")

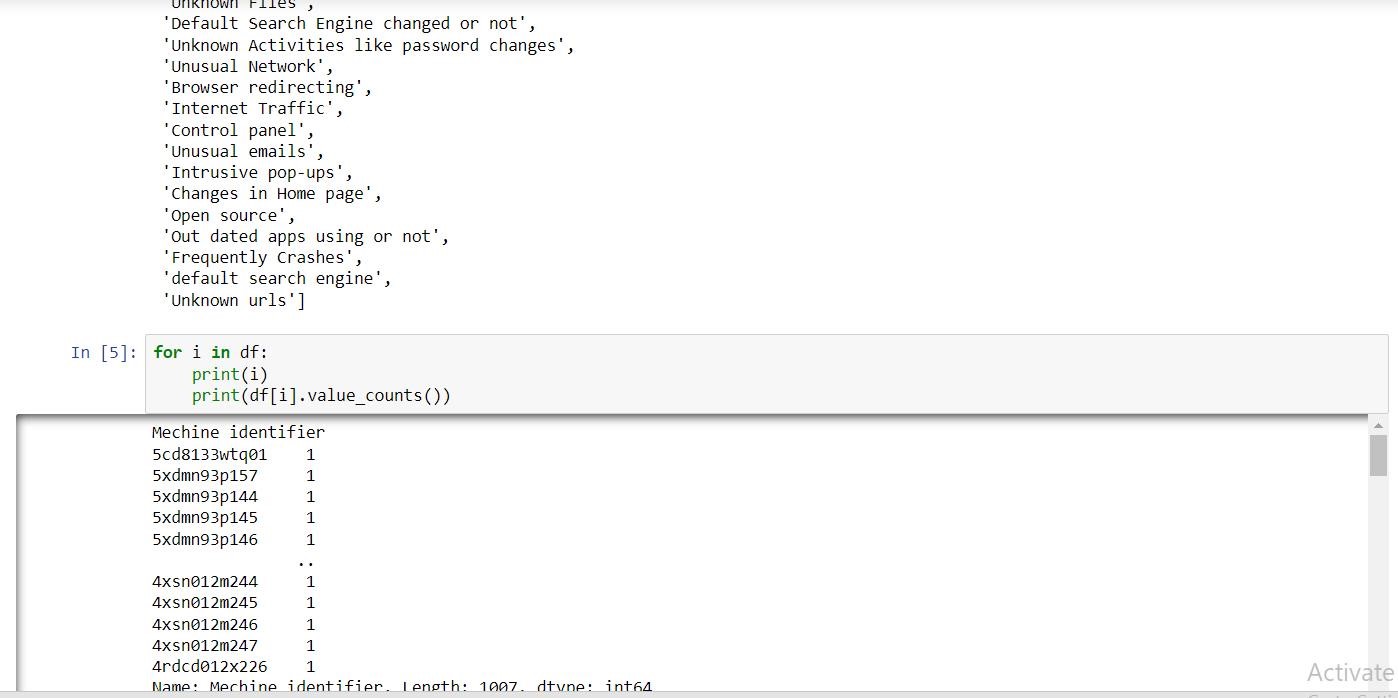
### 4.2.3 Training the model



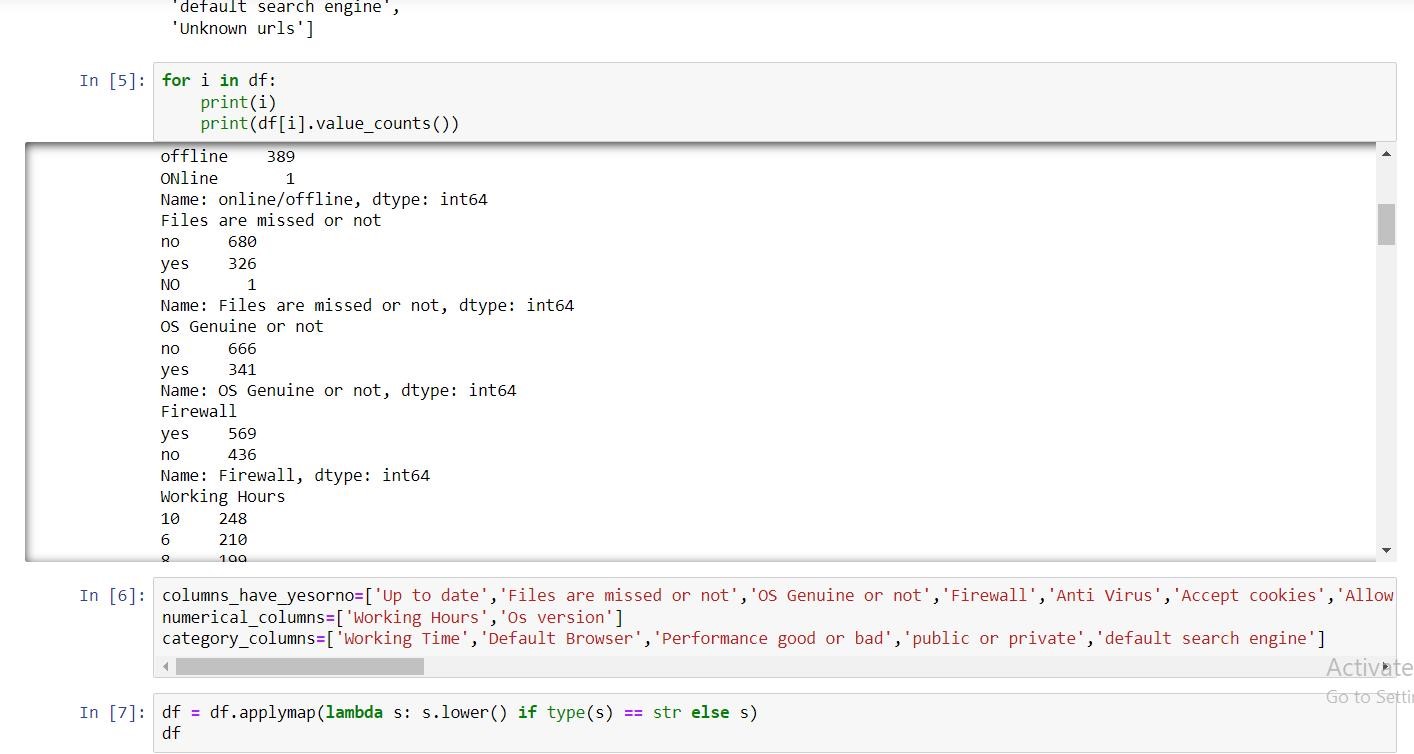
**Fig[7] : Reading Train csv file**



**Fig[8] : No.of columns**



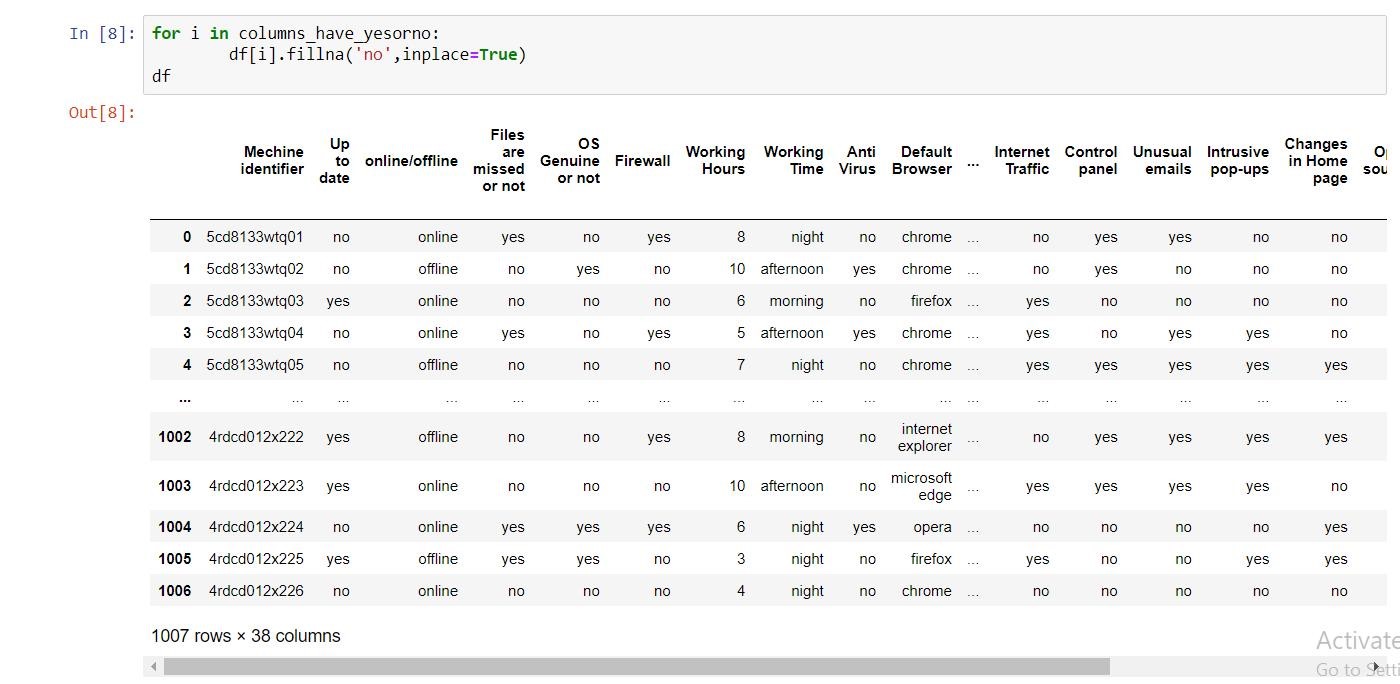
**Fig[9] : Count of each parameter**



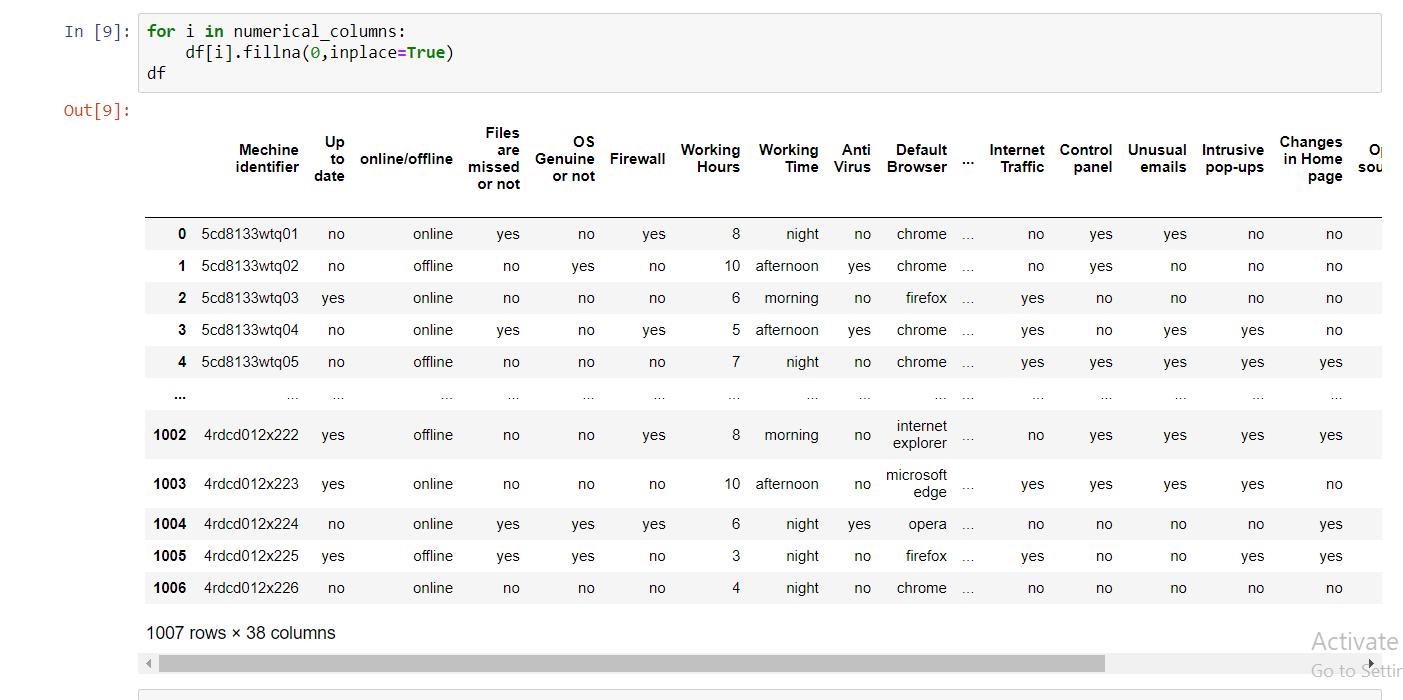
**Fig[10] : Parameter division**



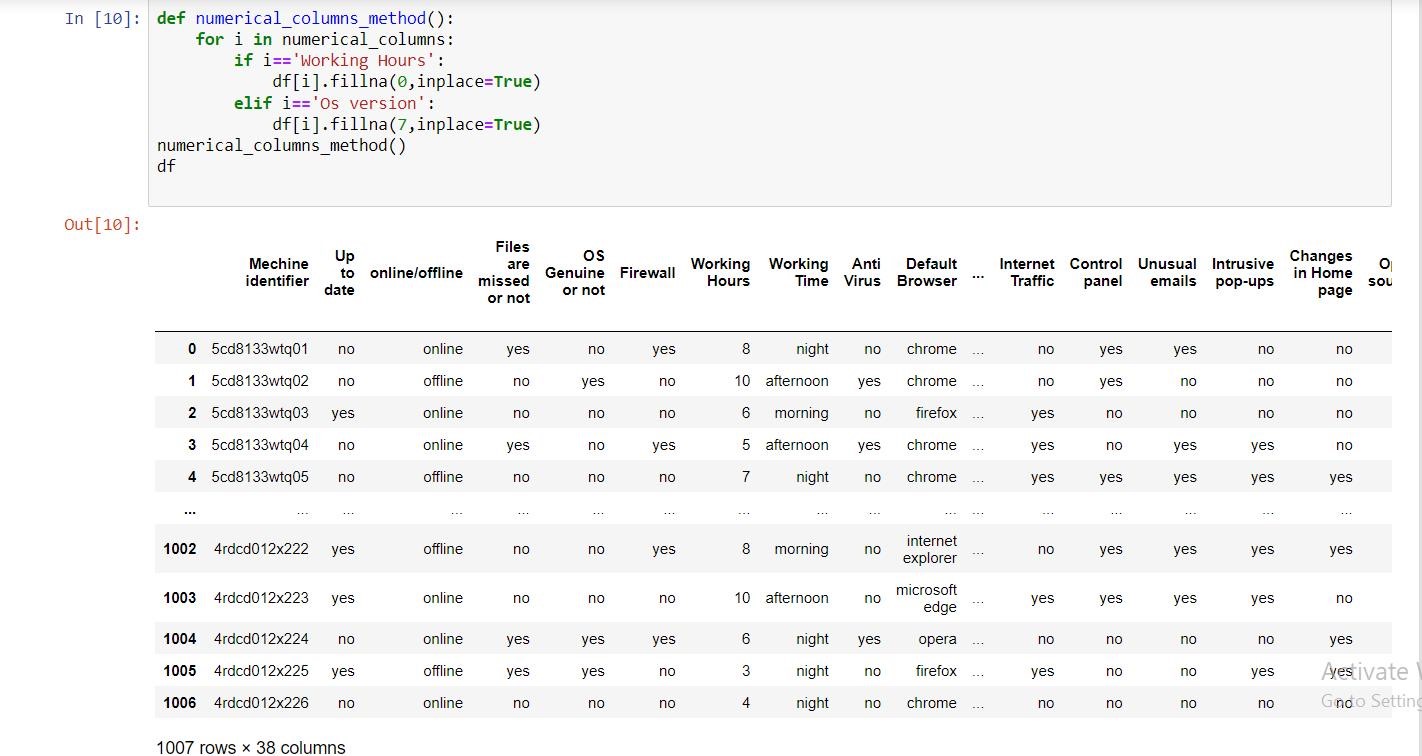
**Fig[11] : applyMap function**



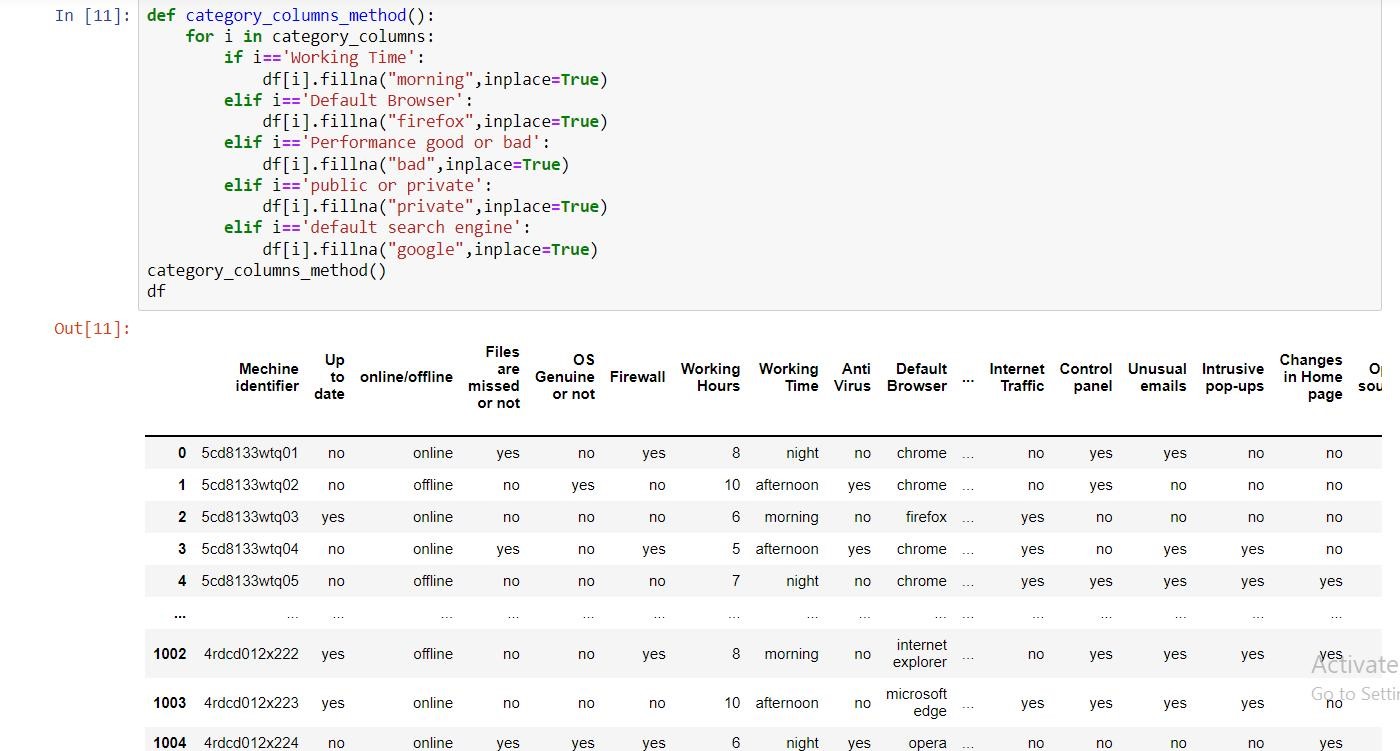
**Fig[12] : fillNA function (1)**



**Fig[13] : fillNA function (2)**



**Fig[14] : Numerical columns method (1)**



**Fig[15] : Categorical columns method(1)**

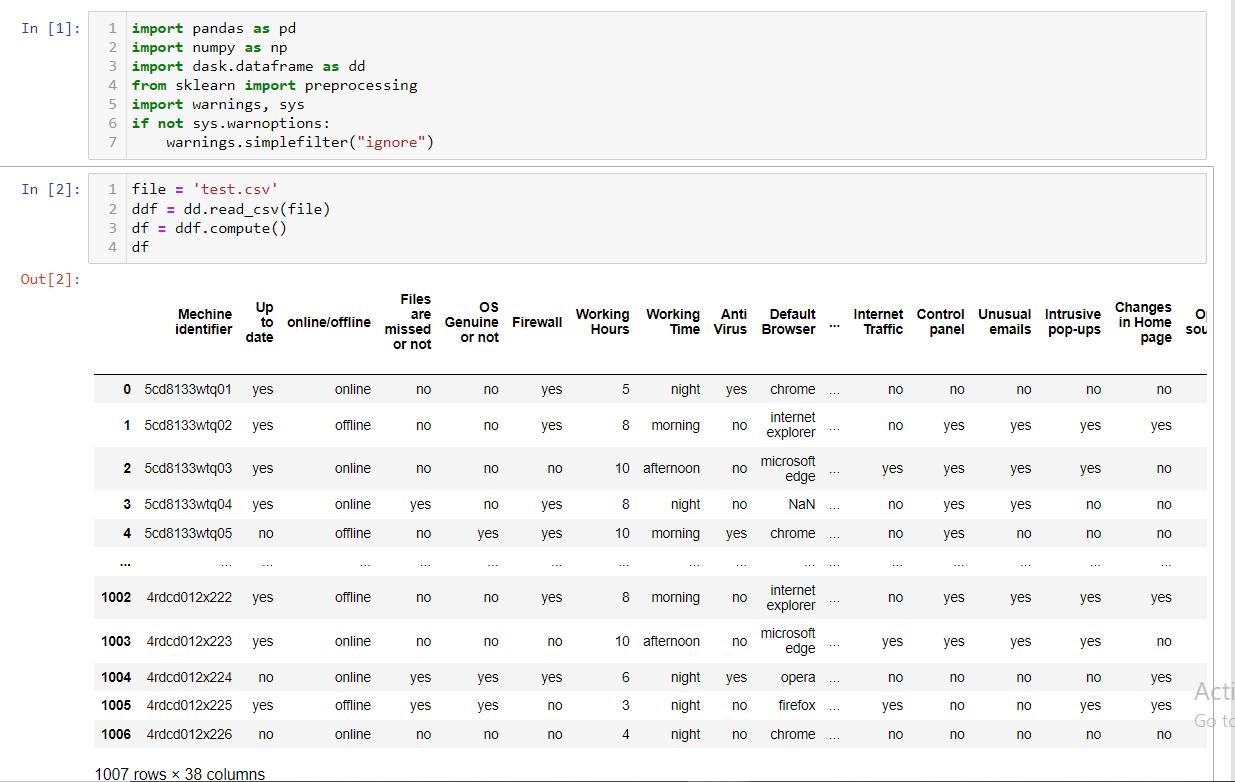


**Fig[16] : Categorical columns method(2)**

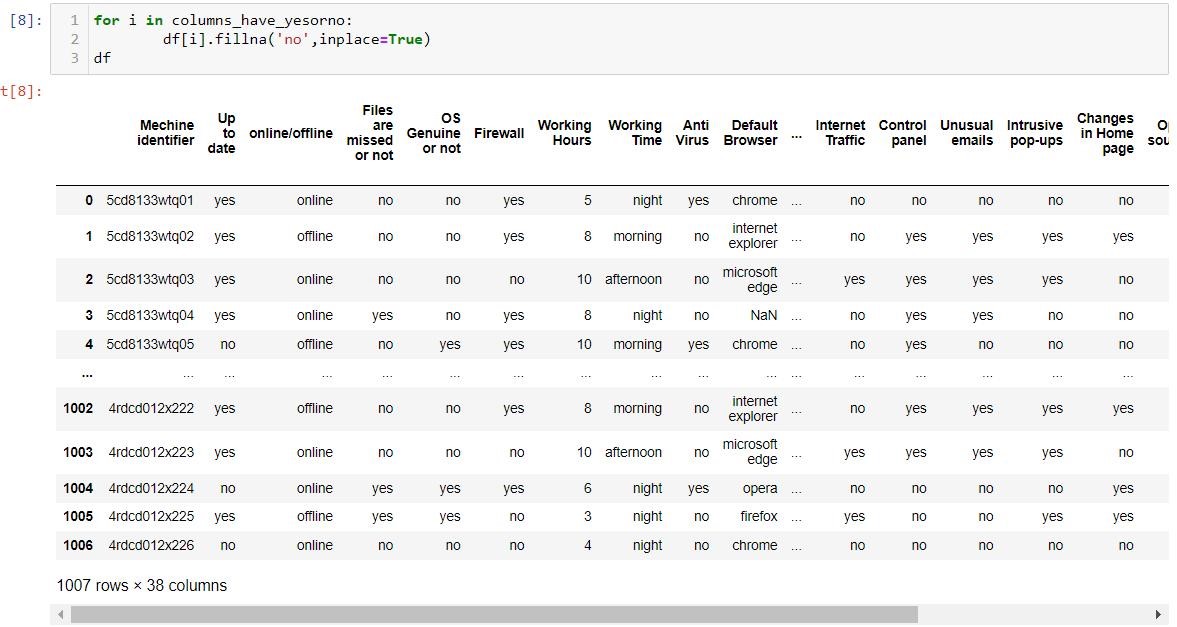
#Cleaned train data set will be formed.

Train\_clean.csv.

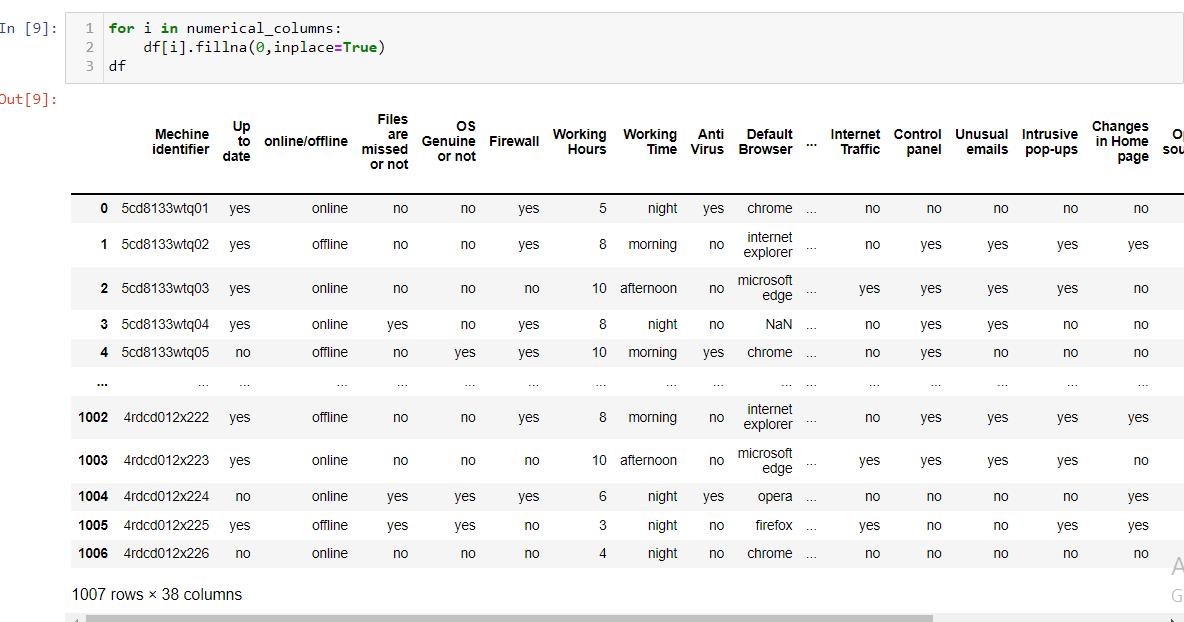
### 4.2.4 Testing model



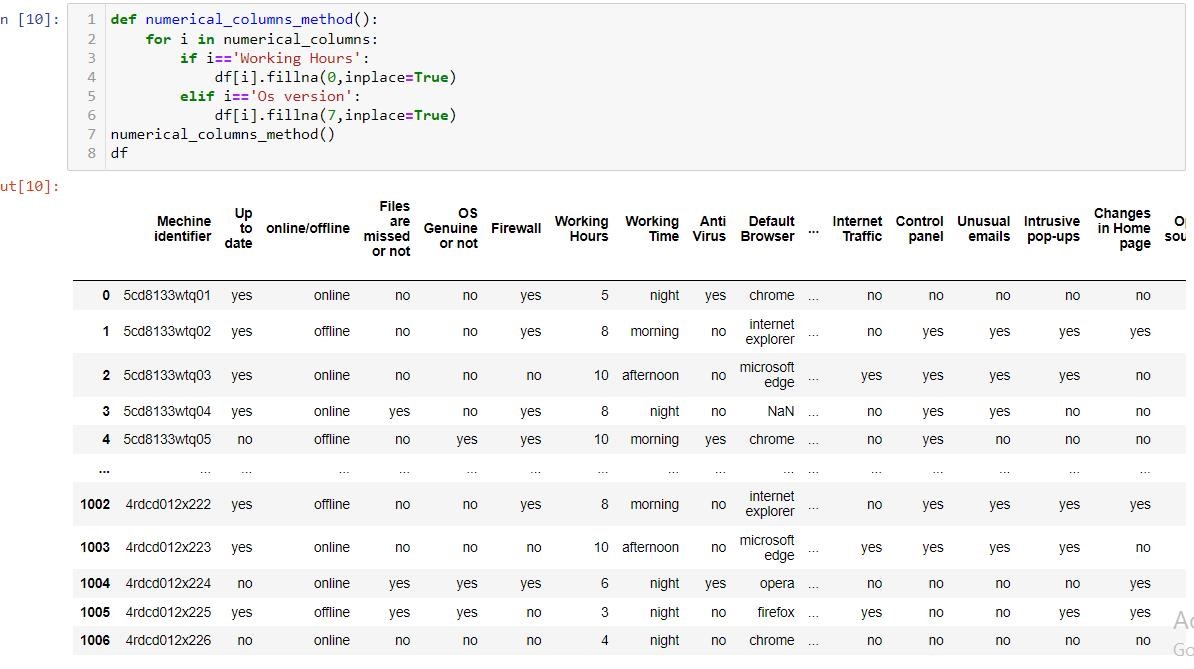
**Fig[17] : Reading Test csv file**



**Fig[18] : fillNA function (3)**



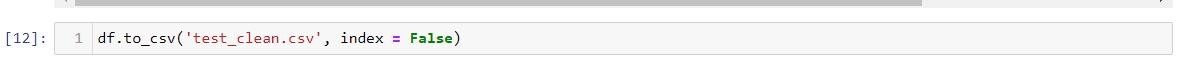
**Fig[19] : fillNA function (4)**



**Fig[20] : Numerical columns method (2)**



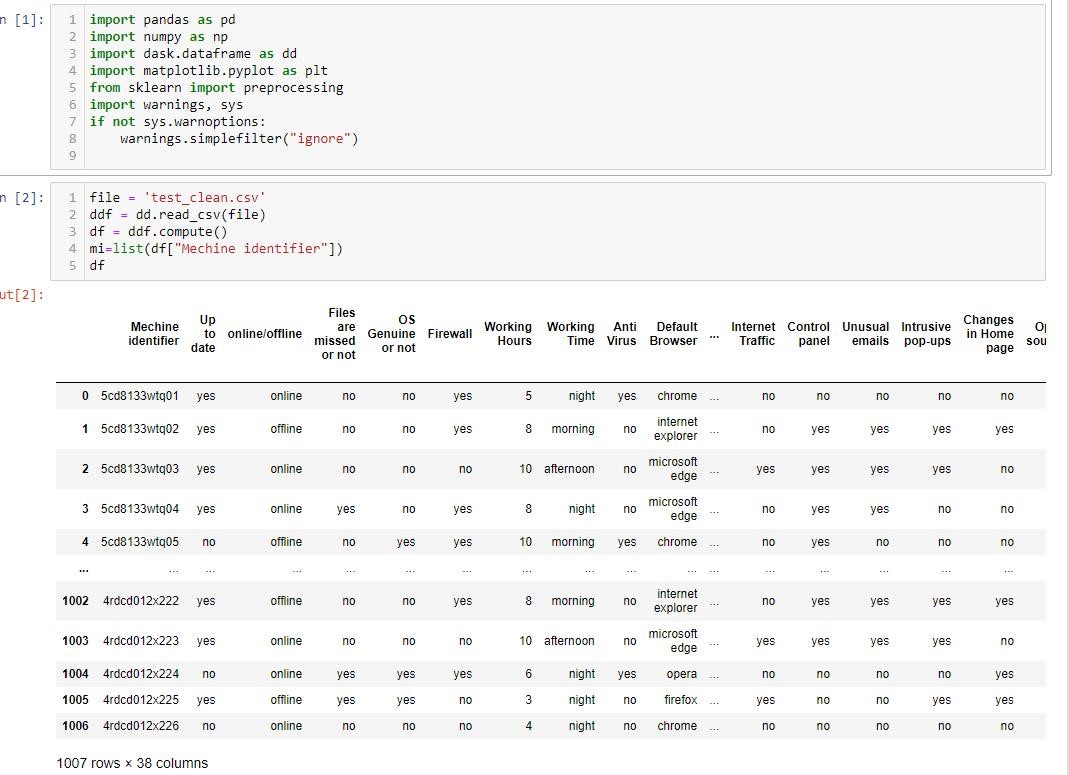
**Fig[21] : Categorical columns method (3)**



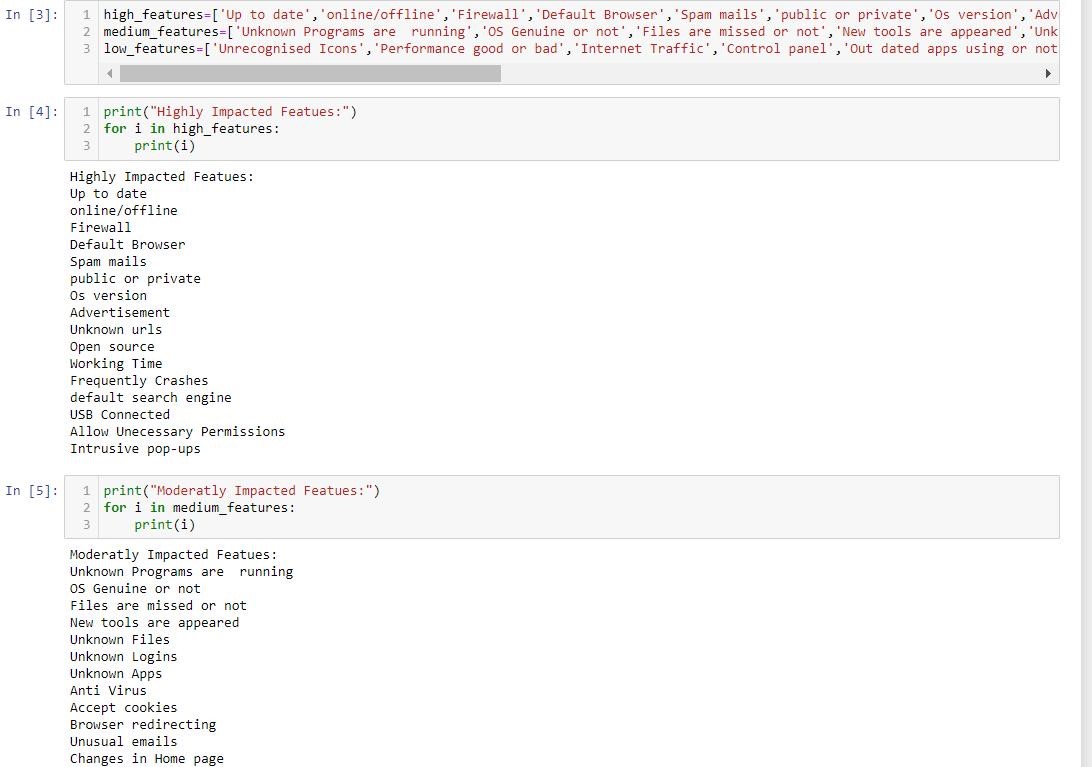
**Fig[22] : Test clean csv file**

# Cleaned train data set will be formed.

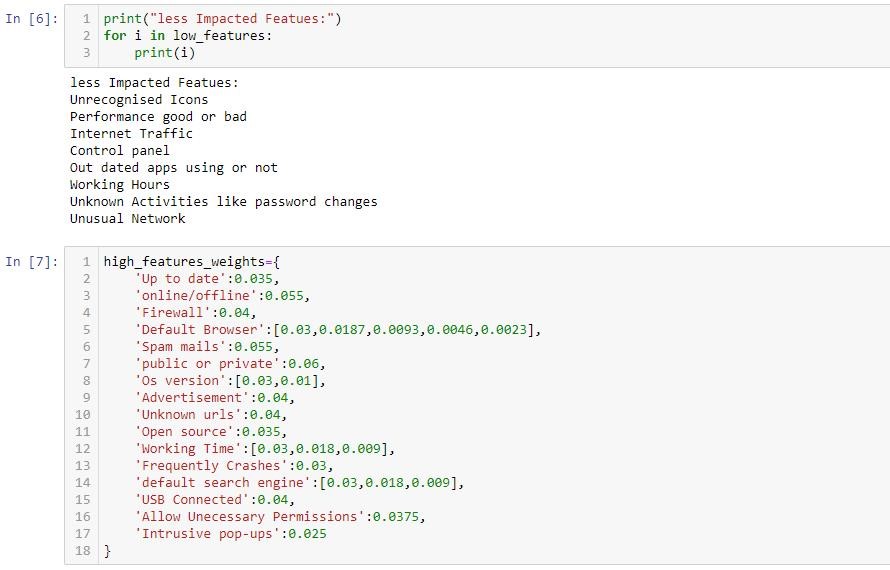
Test\_clean.csv. **final\_test\_file:**



**Fig[23] : Reading Test clean file**



**Fig[24] : Features division**



**Fig[25] : High features weights**



**Fig[26] : Medium and low features weights**



**Fig[27] : High feature weight allocation**



**Fig[28] : High feature weight allocation (1)**



**Fig[29] : Medium feature weight allocation**



**Fig[30] : Medium feature weight allocation (1)**

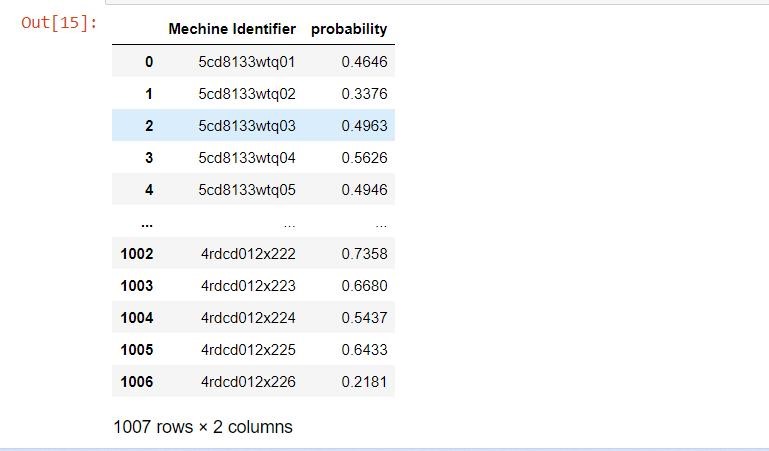


**Fig[31] : Low feature weight allocation**

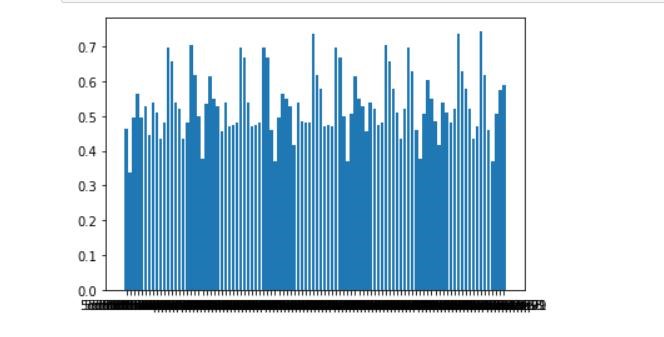


**Fig[32] : Probability calculation**

### 4.3 Result



**Fig[33] : Result (1)**



**Fig[34] : Result (2)**

**Chapter 5**

# Conclusion and Future Work

In this project, we have built a Machine learning model which can show the probability value of each machine that it may hit by malware . Based on the user habits and system conditions , we have calculated the probability of each machine . At the end it will show the predicted probability value of the machine . These values may change in the future according to user actions or based on the technology growth . The machine which has the low probability of malware attack now , may have high probability in the future and vice versa . So, our model needs continuous data to get the inputs on a daily basis . In the future , based on the continuous data we get from the users, we will build software which can be used by the users in their systems . That software will collect the data of user actions and system conditions , so our model which is implemented in that software will show the predicted probability and then the user can reduce his actions which lead to malware attacks .

# Bibliography

1. <https://www.techtarget.com/searchsecurity/definition/malware> .
2. <https://www.researchgate.net/publication/344678463_Malware_Prediction_System> .
3. <https://www.scitepress.org/Papers/2021/102649/102649.pdf> .
4. <https://www.geeksforgeeks.org/malware-and-its-types/amp/>
5. <https://www.vpnmentor.com/blog/which-browser-is-the-most-secure/>
6. <https://www.varonis.com/blog/cybersecurity-statistics>
7. [https://tech.hindustantimes.com/amp/laptops-pc/news/microsoft-removes-the-last-windows-10](https://tech.hindustantimes.com/amp/laptops-pc/news/microsoft-removes-the-last-windows-10-upgrade-blocks-why-users-need-to-update-71620552694376.html)upgrade-blocks-why-users-need-to-update-71620552694376.html
8. <https://www.codespeedy.com/predict_proba-for-classification-problem-in-python/>
9. <https://scikit-learn.org/stable/modules/generated/sklearn.linear_model.LogisticRegression.html>
10. <https://infosecawareness.in/>
11. <https://www.cert-in.org.in/>
12. <https://github.com/NicolasHug/Surprise>
13. <https://www.kaspersky.com/resource-center/threats/malware-system-vulnerability>
14. <https://top5-antivirussoftware.com/articles/10-warning-signs-of-malware-infection/>
15. <https://heimdalsecurity.com/blog/warning-signs-operating-system-infected-malware/>
16. <https://clutch.co/it-services/resources/5-signs-of-malware-infection-and-how-to-get-rid-of-it#five>
17. <https://clario.co/blog/computer-virus-symptoms-effects/>
18. <https://consoltech.com/blog/what-happens-if-your-computer-is-infected-by-malware/>
19. [https://us.norton.com/internetsecurity-malware-malware-101-how-do-i-get-malware-complex](https://us.norton.com/internetsecurity-malware-malware-101-how-do-i-get-malware-complex-attacks.html)attacks.html
20. <https://www.mcafee.com/en-us/antivirus/malware.html>
21. <https://www.rapid7.com/fundamentals/malware-attacks/>
22. <https://arstechnica.com/information-technology/2020/12/ongoing-malware-attacks-are-hitting-users-of4-major-browsers/>
23. <https://ieeexplore.ieee.org/document/9230624>
24. <https://www.geeksforgeeks.org/how-to-install-jupyter-notebook-in-windows/>
25. <https://ashokkumar2216.medium.com/lightgbm-light-gradient-boosting-machine-eb83e1a2ca0a>
26. <https://arxiv.org/ftp/arxiv/papers/1903/1903.02966.pdf>

46