**FINAL PROJECT DESCRIPTION AND DELIVERABLES**

**Table of Contents**

[1. 3](#_Toc118892790)

[a. Project description 3](#_Toc118892791)

[b. Feature extraction 4](#_Toc118892792)

[c. Data visualization 6](#_Toc118892793)

[d. Data pre-processing and data splitting techniques 10](#_Toc118892794)

[e. Description of the model(s) used for analysis 11](#_Toc118892795)

[References 17](#_Toc118892796)

# 1.

## a. Project description

In recent times cyber attacks are happing almost on a daily basis and during this time securing the network has become a crucial part of the organization. Checking all the loopholes and vulnerabilities of a network manually is a time-consuming and tiresome process. That is when data analysis and machine learning come in very handy from time to time. In the case of making the overall network more secure, identifying all the vulnerabilities is a crucial part and implementation of machine learning can be done for making this overall process automated. There are various information such as vulnerabilities of each device, network protocol types, port numbers, and so on can be extracted by performing data analysis. Once the network engineer gets all the necessary information so then they can be able to fix those loopholes accordingly. While performing the data analysis, network engineers can be able to implement various tools and techniques like data mining, data visualization and so on. Data visualization can be very helpful because the network engineer can be able to understand the vulnerabilities much easier way without digging deep inside the data. In the data visualization process, all the vulnerabilities can be shown in the graphical representation and this process can also be very beneficial in the case of showing new insights and small details of the dataset as well.

In the above project, the implementation of three machine-learning algorithms has to be done in the case of identifying the vulnerability of the system with the help of using the “CVSS” values. The “CVSS” values are Common Vulnerability Scoring System which can help to determine whether the devices on the network are vulnerable or not based on a scoring system. The network engineers can be able to make the system more secure by lowering the score values. While performing the overall data analysis and implementation of the machine learning algorithms a given “Training Dataset” can be used. In that dataset, there are ten different columns that can be seen which are - ‘CVSS’, ‘Plugin ID’, ‘CVE’, ‘Protocol’, ‘Port’, ‘Name’, ‘Description’, ‘Solution’, and ‘See Also’. In the case of identifying the vulnerabilities, the ‘CVSS’ column can be used as a targeted column.

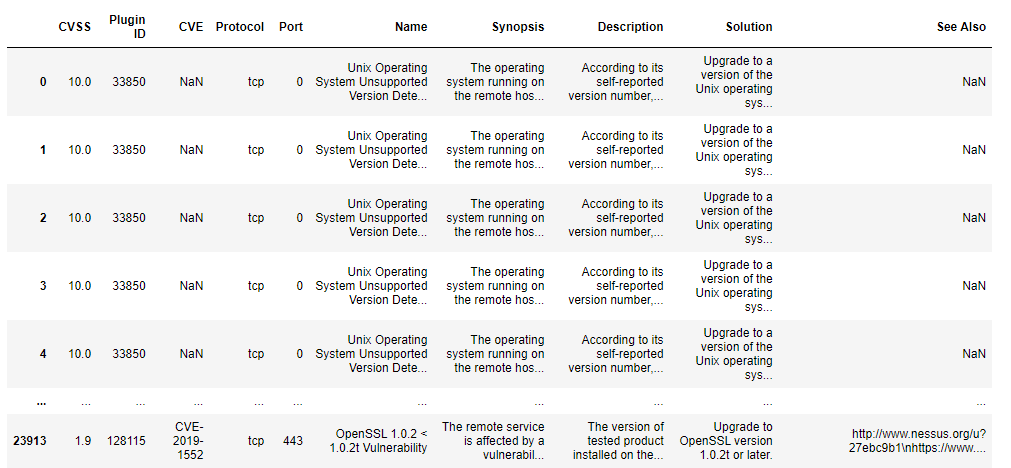
***Objectives***

In the above project, there is various work that can be done but the main objectives of the project are -

* To perform data analysis in the case of identifying the vulnerabilities in the system
* To implement a Common Vulnerability Scoring System for understanding the vulnerability level of each device
* To implement three machine learning algorithms that can able to extract the vulnerability of the devices in the network

## b. Feature extraction

In the case of performing the above project data plays a major role. There is a type of machine learning algorithm that can be implemented here and in that case of implementation of those machine learning algorithms, a dataset has to be used accordingly. In this case, a “Training Dataset” has already been given and there are ten different columns that can be seen in the dataset. There is a particular sort of AI calculation that might be used in this present circumstance, and in such an occurrence, a dataset must be utilized suitably. On this occasion, a "Preparing Dataset" has proactively been given, and the dataset has 10 unique segments.



**Figure 1: Original Dataset**

(Source: Acquired through python notebook)

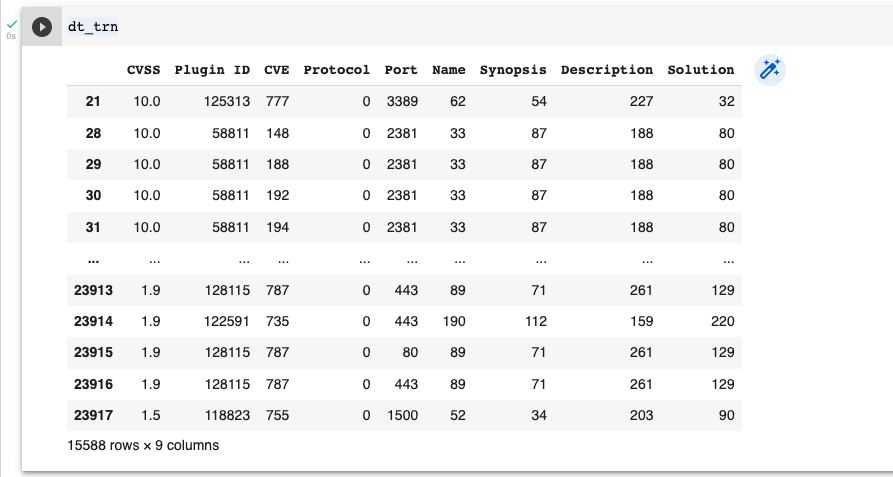
In the above figure the given “Training Dataset” can be seen which is imported into the “Python Environment”. Checking the overall dataset is an important concept because by checking the dataset one can be able to understand the overall dataset. There are some changes that have been made in the dataset before implementing the machine learning algorithms.

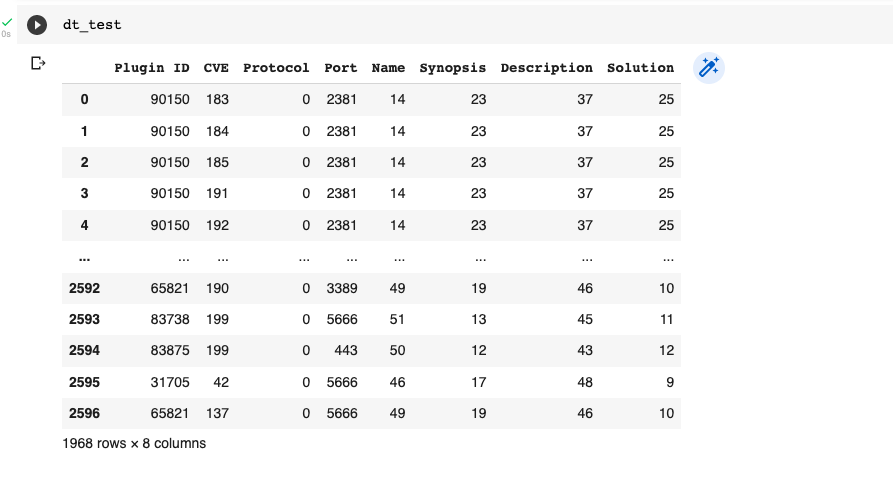


**Figure 2: Converting the dataset into categorical**

(Source: Acquired through python notebook)

In the above figure, the conversion has been made in the dataset. All the data that was in the text format has been converted into categorical values and by doing that all the machine learning algorithms can be implemented accordingly. The changes have been done for all the columns of the “Training Dataset”.





**Figure 3: Dataset after the conversion**

(Source: Acquired through python notebook)

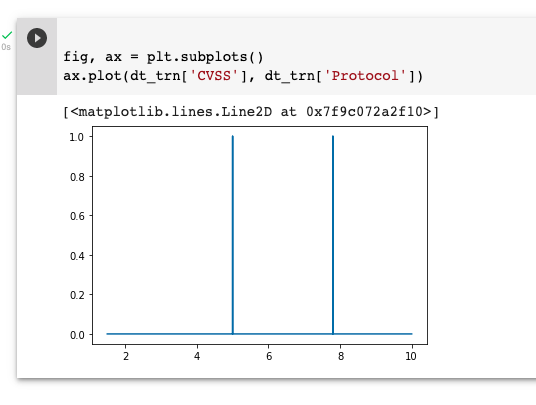
In the above figure, the dataset has been checked and there are some changes that can be seen in the dataset. In the above figure, all the text values have been converted into some numerical values and the changes can be seen for ten columns as well.

## c. Data visualization

Visualization of data is generally representing data or information in a graphical way. During the time of big data, the tools and techniques are important for analyzing a large amount of data. Data visualization is an important aspect of data analysis and it plays a major role in the project. While using the data visualization technique, data can be represented in a graphical chart form. This graphical representation can be very beneficial for the network engineer in the case of understanding the vulnerabilities of each device in a network (Styawan *et al*., 2019). In the case of performing data analysis, a “Training Dataset” has been used and the data visualization can be performed in the ‘Python’ environment. Using data visualization can be very beneficial because it can be able to show new information and insights about the dataset. The network engineer can be able to understand all the concepts of the data without digging deep inside the dataset.

It is such an important factor as it helps to interact, see, and understand the data in a better way. It is a practice that translates data to a visual context including graphs, for making it easy to understand as well as extract the relevant information. It helps in identifying the trends, patterns, and outliers in a massive data set.

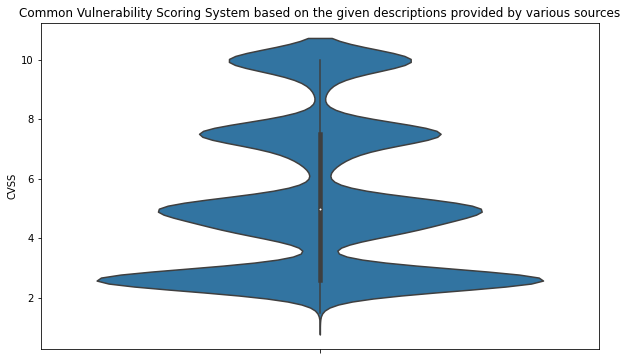
Performing a descriptive analysis can also be seen in the project and it is a very important aspect of the whole project (Grujić *et al*., 2019). The distribution of the data can be understood by performing descriptive analysis from time to time. In the case of handling the statistical data, descriptive data analysis plays a major role and in the case of descriptive analysis, there are two types of variables that can be used. One type of variable is quantitive and the other type of variable is categorical. In the case of quantitive variables creating histograms and box plots can be done whereas in the case of categorical variables bar charts and pie charts can be implemented.



**Figure 4: Line graph**

(Source: Acquired through python notebook)

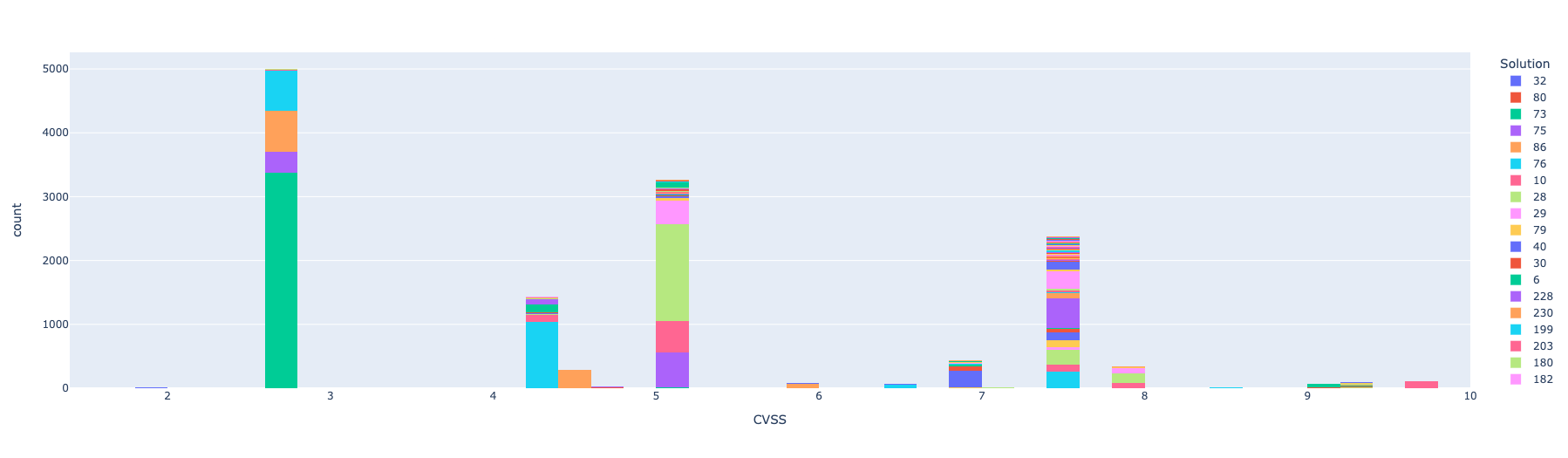
In the above figure, a line plot has been generated with the help of using the “Python” software. While generating the line graph there are two columns have been taken from the given dataset which are - ‘CVSS’ and ‘protocol’. There is two axes that can be seen in the above figure and both axis contains some numerical values.



**Figure 5: Violin plot**

(Source: Acquired through python notebook)

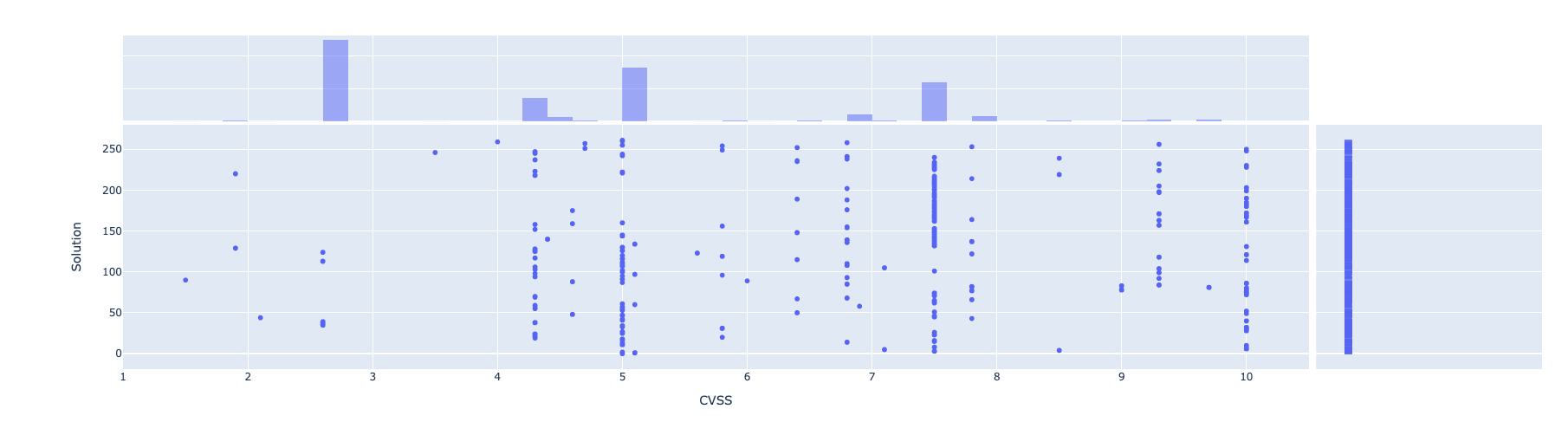
In the above figure, a violin plot has been generated in the “Python” environment. In the case of generating the violin plot, the ‘CVSS’ column has been taken and based on the column the graph has been generated.



**Figure 6: Interactive bar graph with plotly library**

(Source: Acquired through python notebook)

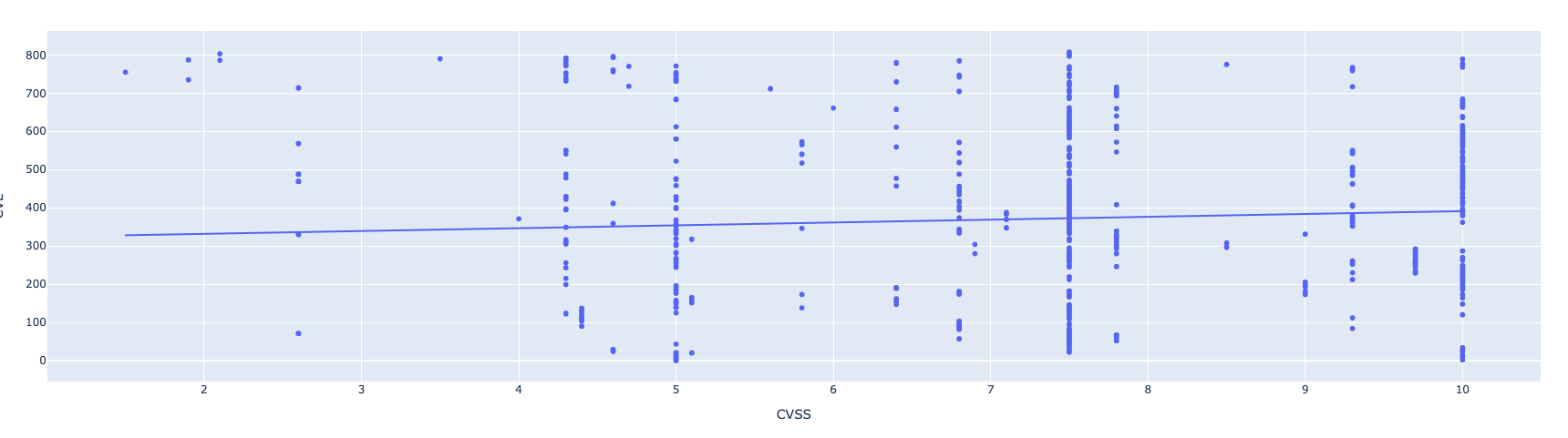
In the above figure, an interactive bar graph has been generated with the help of using the ‘CVSS’ and ‘Solution’ columns of the given dataset. In the case of generating the above graph, the “plotly” library has been used here.



**Figure 7: Interactive scatter-bar graph with plotly library**

(Source: Acquired through python notebook)

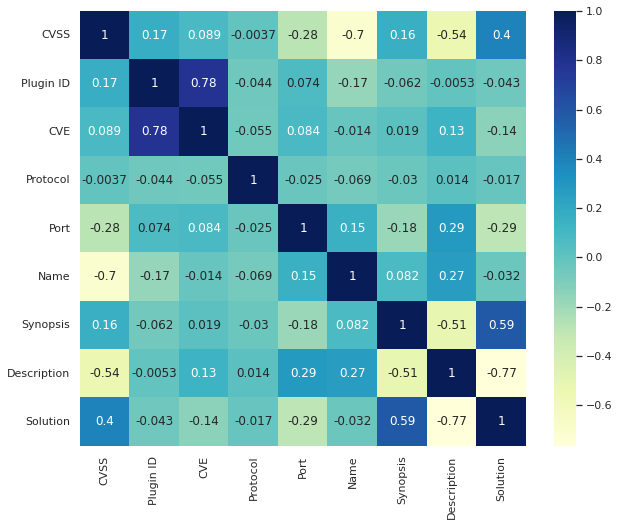
In the above figure, an interactive scatter-bar graph has been generated with the help of using “Python” software. There are two columns ‘CVSS’ and ‘Solution’ has been taken in the case of generating the graph. There is two axes that can be seen that contain some numerical values.



**Figure 8: Interactive scatter-line graph with plotly library**

(Source: Acquired through python notebook)

In the above figure, an interactive scatter-line graph has been generated with the help of using “Python” software. There are two columns ‘CVSS’ and ‘CVE’ has been taken in the case of generating the graph. In the case of generating the scatter-line graph, a “plotly” library has been used accordingly.



**Figure 9: Correlation heatmap**

(Source: Acquired through python notebook)

In the above figure, a correlation heatmap has been generated with the help of using the “Python” software. In the heatmap, various columns of the dataset can be seen on both axes.

## d. Data pre-processing and data splitting techniques

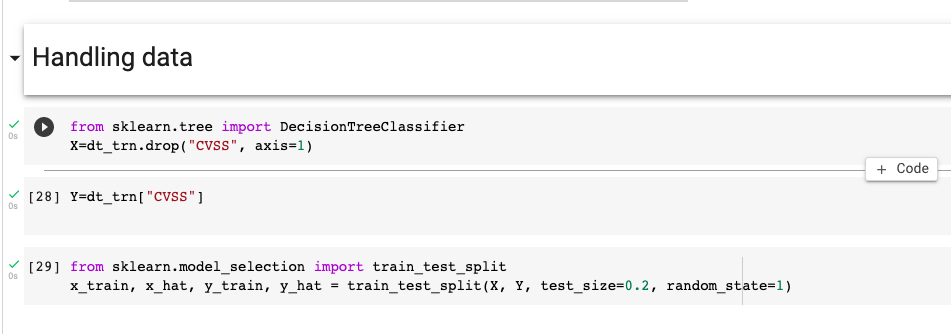
In the case of identifying the vulnerabilities of each device in a network, performing a data analysis can be very beneficial. There are some steps that can be taken by the network engineers before performing data analysis. Shaping the raw data is a crucial part of the project and that is why data preprocessing can be done. Data preprocessing has various segments and one of the main segments of data preprocessing is data cleaning. There are some unnecessary values like null values and duplicate values that can be seen in the dataset and these values can affect the overall decision-making process negatively. That is why it is crucial and much recommended to clean the dataset before performing data analysis (Reddy and Sajjan, 2021). The overall performance of the data analysis can be increased on a tremendous level once all the unnecessary data has been removed properly.

There are some other errors that can also be seen in the dataset. Some missing values and other errors can be seen and those errors can be fixed by the analyst. Once fixing all the errors can be done then the data transformation can be performed accordingly (Nusrat *et al*., 2019). These steps can be very helpful for implementing a suitable data mining process. In the case of data transformation scaling the data can be very useful because it can show the range (Perin *et al*., 2018). In this project, a “Training Dataset” can be seen where some values are in the string format and that is why changing those values into the categorical is important. The implementation of the machine learning algorithm can not be done if the data has not been converted into categorical values. A "Training Dataset" has been utilised to analyse data, and the "Python" environment may be used to visualise the data (Stančin, and Jović, 2019). Utilizing data visualisation may be quite advantageous since it can provide fresh details and insights about the dataset. Without delving too deeply into the dataset, the network engineer may be able to comprehend all of the ideas included therein.

Data mining can also be implemented in the case of extracting new pieces of information and insights into the dataset (Hammad *et al*., 2021). There are some data mining tools that can be implemented in the above project and by implementing the data mining techniques the network engineers can be able to identify all the vulnerabilities that can be seen in each device of a network (Narasamma and Sreedevi, 2021). These are the few techniques that can be used in the case of checking and identifying the vulnerabilities of each system in a network.

## e. Description of the model(s) used for analysis

In the case of making a model that can be able to check the vulnerabilities of each device in a network, there are three machine-learning models have been implemented here accordingly. Those three machine learning models are - ***“Ridge Regression”***, ***“Lasso Regression”***, and ***“Random Forest Regression”***. The network engineer can be able to make a model that can automatically identify all the vulnerabilities of each device with the help of using these three algorithms



**Figure 9: Setting data parameters for model training**

(Source: Acquired through python notebook)

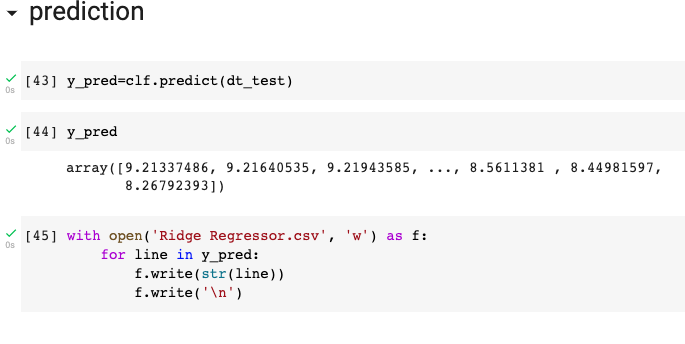
In the above figure, all the parameters have been set accordingly. The parameters have been set with the help of using the “Python” programming language. While training the model, the “CVSS” column of the given dataset has been used.



**Figure 10: Model training in “ridge regression”**

(Source: Acquired through python notebook)

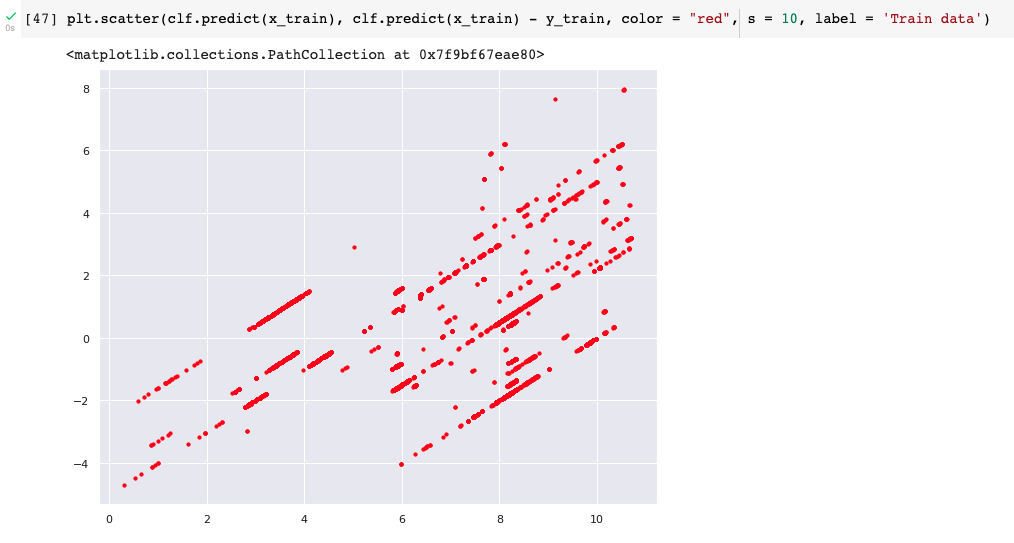
In the case of solving regression problems, ***“Ridge Regression”*** can be very beneficial from time to time. It can be implemented in those scenarios where independent variables can be seen in the dataset. It can be very beneficial in the case of working with a higher number of predictive variables. It can also be used where multicollinearity can be seen in the dataset as well. There are some errors that can be seen and it can be able to reduce those errors from time to time.



**Figure 11: Predicting outcomes using “ridge regression”**

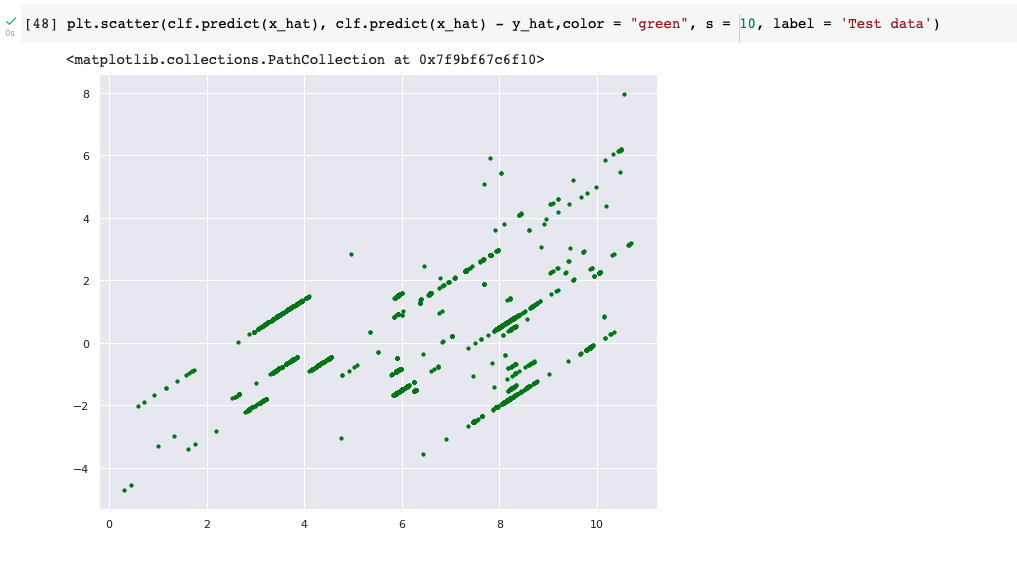
(Source: Acquired through python notebook)

In the above figure the accuracy score of the “Ridge Regression” can be seen and there are some outcomes that can be seen in the above figure.



**Figure 12: Scatter plot for the training data**

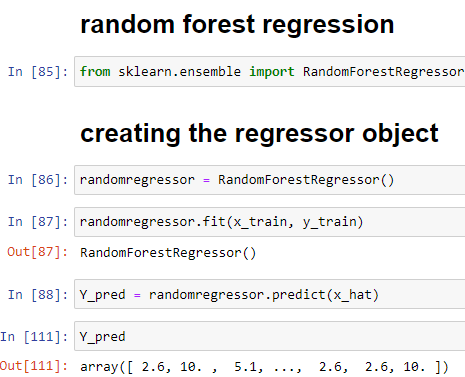
(Source: Acquired through python notebook)



**Figure 13: Scatter plot for the test data**

(Source: Acquired through python notebook)

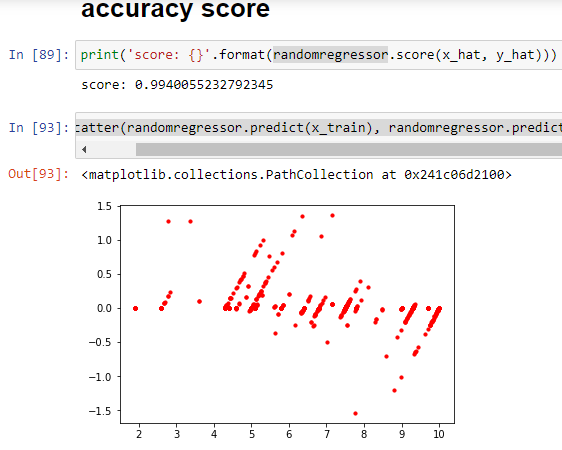
Another machine learning model that has been used here for making the model is the ***“Random Forest Regression”***. It can also be very useful in the case of solving regression problems. It uses different machine learning algorithms in the case of operating a single task. That is the reason why it can be able to speed up the overall process by a tremendous amount. Implementation of a ***“Random Forest Regression”*** has been done in the project because it is a simple machine-learning algorithm that can be able to produce highly accurate results from time to time.



**Figure 14: “Random forest regressor”**

(Source: Acquired through python notebook)

In the above figure a ***“Random Forest Regressor”*** has been generated here with the help of using “Python” programming language.

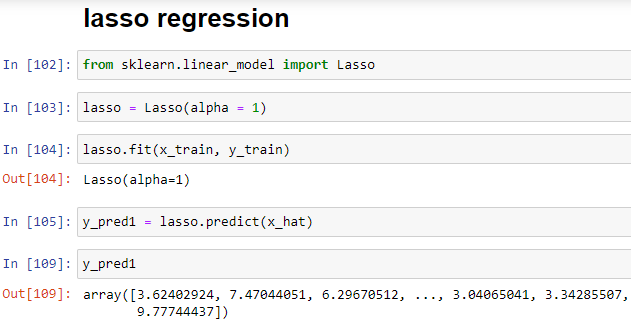


**Figure 15: Accuracy score of the “random forest regressor” model**

(Source: Acquired through python notebook)

In the above figure the accuracy score of the “Random Forest Regression” has been generated and the accuracy of this model is 0.99.

The accuracy will increase if the number of ***“decision trees”*** is increased. It can be able to prevent an overfitting problem as well. An advantage that can be seen while using this algorithm is in the training time it takes very less time and the accuracy will is very high. These are the reason why the implementation of the ***“Random Forest Regression”*** has been done in the above project.

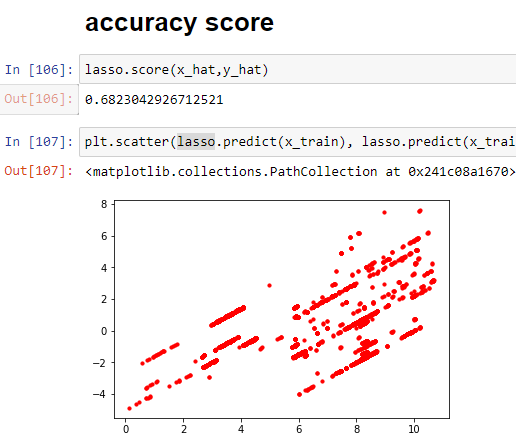


**Figure 16: Model training using the “lasso regression” algorithm**

(Source: Acquired through python notebook)

In the above figure, model training has been done for the ***“Lasso Regression”*** and the training has been made with the help of using the “Python” software.

Implementation of the ***“Lasso Regression”*** model can also be seen in the project. It is a regression that can be used for getting accurate results from time to time. While performing a regression, it uses regularization techniques. It also uses a method called shrinkage. The data can be shrunk to the central point and then it will show as mean. It is the most simple model that uses few parameters for performing a task. It can also be used when multicollinearity can be seen in the dataset. In the ***“Lasso Regression”***, the ‘L1 regularization technique’ can be seen. Overfitting of the data can be prevented with the help of using the ***“Lasso Regression”****.*

****

**Figure 17: Accuracy score of the model designed in lasso regression**

(Source: Acquired through python notebook)

In the above figure, the accuracy score of the ***“Lasso Regression”*** can be seen and the accuracy score of the model is 0.68.

# 

# 

# 

# References

Perin, C., Vuillemot, R., Stolper, C.D., Stasko, J.T., Wood, J. and Carpendale, S., 2018, June. State of the art of sports data visualization. In *Computer Graphics Forum* (Vol. 37, No. 3, pp. 663-686).

Hammad, G., Reyt, M., Beliy, N., Baillet, M., Deantoni, M., Lesoinne, A., Muto, V. and Schmidt, C., 2021. pyActigraphy: Open-source python package for actigraphy data visualization and analysis. *PLoS Computational Biology*, *17*(10), p.e1009514.

Grujić, N., Novović, O., Brdar, S., Crnojević, V. and Govedarica, M., 2019, March. Mobile Phone Data visualization using Python QGIS API. In *2019 18th International Symposium INFOTEH-JAHORINA (INFOTEH)* (pp. 1-6). IEEE.

Styawan, Y., Andika, P.P., Suhendi, C., Firdaus, R., Sudibyo, M.R., Erlangga, I.F. and Ry, R.V., 2019, August. The preliminary results of Lindu software: a free seismological data processing using python framework. In *IOP Conference Series: Earth and Environmental Science* (Vol. 311, No. 1, p. 012078). IOP Publishing.

Narasamma, V.L. and Sreedevi, M., 2021. Twitter based Data Analysis in Natural Language Processing using a Novel Catboost Recurrent Neural Framework. *International Journal of Advanced Computer Science and Applications*, *12*(5).

(Dejdar *et al*., 2019)

Reddy, M.V. and Sajjan, M., 2021. Phoneme and Phone Pre-processing Using CLIR Techniques. In *Data Science and Security* (pp. 299-304). Springer, Singapore.

Shumaly, S., Yazdinejad, M. and Guo, Y., 2021. Persian sentiment analysis of an online store independent of pre-processing using convolutional neural network with fastText embeddings. *PeerJ Computer Science*, *7*, p.e422.

Stančin, I. and Jović, A., 2019, May. An overview and comparison of free Python libraries for data mining and big data analysis. In *2019 42nd International Convention on Information and Communication Technology, Electronics and Microelectronics (MIPRO)* (pp. 977-982). IEEE.

Nusrat, S., Harbig, T. and Gehlenborg, N., 2019, June. Tasks, techniques, and tools for genomic data visualization. In *Computer Graphics Forum* (Vol. 38, No. 3, pp. 781-805).

Dejdar, P., Munster, P. and Horvath, T., 2019, July. High-speed data acquisition and signal processing using cost effective arm+ fpga processors. In *2019 42nd International Conference on Telecommunications and Signal Processing (TSP)* (pp. 593-596). IEEE.