Machine Learning Project: Malware Detection using Machine Learning

Problem Statement:

In the current world scenario, the internet has become a huge part of everyone's lives, with millions and millions of data packets being downloaded and uploaded each day. With such a huge transfer rate, it becomes very easy for some to sneak in just one unit of data, capable of massive destruction of an application, a whole system, or the entire network. One may never find out if the file one is about to download contains malware or not. With one download containing so many files, it is nearly impossible to physically find out if a file contains a malware or not. Thankfully, this job has been made easier by Machine Learning algorithms.

Dataset:

The dataset which can be used for the model is the Brazillian Malware Dataset.
link: https://github.com/fabriciojoc/brazilian-malware-dataset

Solution:

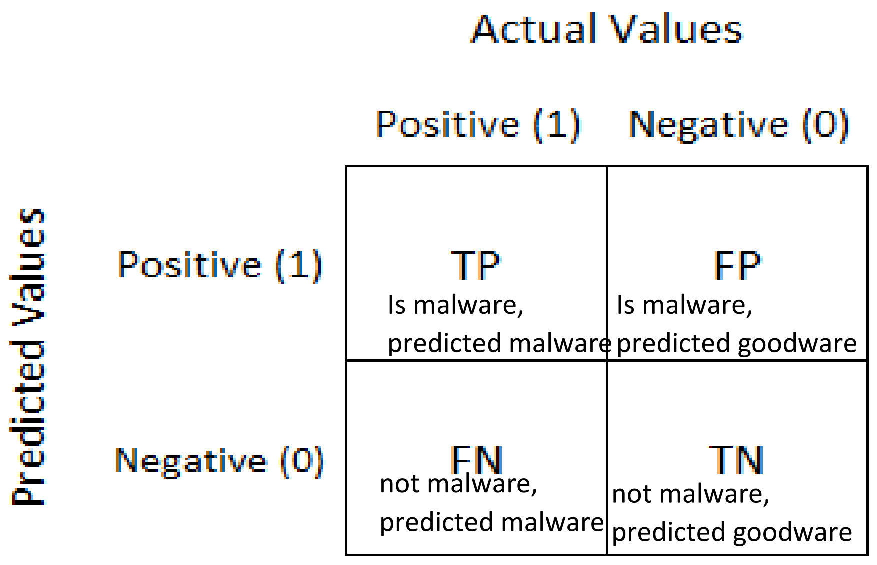
Every time the user downloads something, before running/opening/executing it, the user can pass it through a trained ML model, which will predict if the data contains malware or not. If it does, the user can delete the file right away, avoiding any damage to the system. If it does not have any malware, the user can safely open/run the file.

Historical model for comparison(benchmark):

It has been found through a paper that the Random Forest Classifier has been expected to be the best model, with around 97% accuracy. It was compared with models like the Decision Tree classifier, the Naïve Bayes Classifier and the AdaBoost Classifier. I would like to see how it compares with the XGBoost Classifier. Though it may require some more training time, it might prove to fit better to the dataset then the Random Forest Classifier.

Testing Metrics:

Although accuracy may be a good measure, we might want to observe the confusion matrix:



		Actual Values	
		Positive (1)	Negative (0)
Predicted Values	Positive (1)	TP Is malware, predicted malware	FP Is malware, predicted goodware
	Negative (0)	FN not malware, predicted malware	TN not malware, predicted goodware

In our case, accuracy will not be the best metric. Though we can afford having False Negatives (hey, the model's playing it safe), we cannot have any False Positives at all. Hence, a good testing metric would be the Recall value of the model, over its precision. We can check the f-beta score of the model, beta being set to a value close to 1. The testing metric would be the f-beta score.

Project Design:

Like every machine learning project, we start with:

1. Data Preprocessing:
   We already know where the data is. After downloading, an analysis is needed on the dataset, to find out columns having too many Nulls, columns having redundant values, columns not making any difference to the analysis, etc. Also, we need to perform separate preprocessing for the string type columns, (one hot encoding, etc). After normalising, we'll have a dataset ready to feed to our models, after being split into training, validation and testing data.
2. Model Training:
   Since we've already fixated on the model, we are going to implement the XGB Classifier and fit it to the training data. We will evaluate the model using the validation set.
3. Model Tuning:
   Obviously, we cannot expect the model to be the best right at the start. Some hyperparameter tuning will definitely be required (We may use the XGB's hyperparameter tuner to find the best possible estimator). We will train the new optimised model once again, and see how it works.
4. Model Prediction:
   Once the model is ready, it can make predictions on any data provided to it.
5. Model Deployment:
   If working well, the model could be deployed on an AWS API Gateway API, fed with a Lambda function. We can use a localhost webpage to make calls to the API, leading to the model predicting whether the input file has malware or not.

Future Idea:

We could create something like a Google Chrome extension, where the file's download link could be pasted, and a local server downloads the file itself, and checks it through the model. The extension's colour could show the level of danger in the model.