Android Malware Detection

Contents

[Introduction to Data Science Assignment 1](#_Toc80544440)

[Business Understanding 3](#_Toc80544441)

[Ideas 3](#_Toc80544442)

[Selected Idea 3](#_Toc80544443)

[Data Acquisition 3](#_Toc80544444)

[Understanding 3](#_Toc80544445)

[Wrangling 1 7](#_Toc80544446)

[Wrangling 2 7](#_Toc80544447)

[Feature Selection 7](#_Toc80544448)

[Results of Feature Engineering Technique 1 8](#_Toc80544449)

[Results of Feature Engineering Technique 2 9](#_Toc80544450)

[Modelling 10](#_Toc80544451)

[Results of ML Technique 1 11](#_Toc80544452)

[Results of ML Technique 2 12](#_Toc80544453)

[Comparison 13](#_Toc80544454)

[Conclusion 13](#_Toc80544455)

[Recommendation/Suggestion 14](#_Toc80544456)

## Business Understanding

With the popularity of Android devices, the number of applications made for the android operating system is also increasing day by day. But the biggest challenge in this scenario is to identify if an application is an authentic application or a malware. This project tries to identify an application as malware/not based on the permissions required by the application.

## Ideas

We have listed out some of the ideas to solve the problem.

1. Find the number of rows and columns in the given dataset to see what we are working with.
2. Find the number of discrete and continuous variables in the dataset.
3. If there are missing values in the continuous variables, check the number of missing values & try substituting them with mean, median and mode to see which performs the best.
4. If there are missing values in the discrete variables, impute them with either mode or create a new category for missing values if they are more in number.
5. For finding correlation between the different columns, plot them to see how the distribution looks like. Remove the columns which have high correlation.
6. Perform Principal Component Analysis (PCA) to find the variance between the 2 components in type as well as visualise the distribution between them.
7. Find the important features by doing univariate selection, principal component analysis or by passing them through ExtraTreesClassifier which will give you the important features by mapping each independent variable with the dependent variable.

## Selected Idea

1. When we did initial exploratory data analysis, we found that the dataset was about permissions given for a particular application. Permissions are wither given or not which means the data is only available as 0s and 1s.
2. Hence the first step taken was to see which independent features dominated most of the malicious and benign types and plot them to see how the distribution was for the top 10 categories.
3. Next step was to see the distribution between the 0s and 1s in the dependent feature column to see whether the dataset it balanced or imbalanced.
4. Perform PCA to see whether the points w.r.t the dependent variable type is spread and clearly distinguishable or closely clustered.
5. Once done, the next step is to find the correlation between the different columns and remove the highly correlated ones.
6. Then we have implemented a couple of steps to find the most importance independent features before training the model with different algorithms.

## Data Acquisition

1. The data is given as a part of one file “Dataset.csv”.
2. Since it is a dataset where permissions are either given or not given for certain applications, the data consists of independent columns which are mainly Nominal Binary in type.
3. The size of the dataset is 271 KB.
4. It consists of 398 rows and 331 columns.
5. The dataset is balanced since the dependent variable column “type” has a perfectly equal distribution of 0s and 1s with each having 199 rows respectively.

## Understanding

1. There are 331 columns where 330 columns are independent variables, and 1 column is the dependent variable.
2. There are no missing values in any of the columns. All the columns have details of whether a permission was given for an application or not.



1. We have visualised in a couple of charts where most permissions were given & type was Malicious and where most permissions were given & type was Benign to find out the insights about a difference between the permissions used by the malware and the benign applications.
2. Most permissions given where the type is Malicious

A picture containing text, indoor, screenshot

Description automatically generated

Chart, bar chart

Description automatically generated

1. Most permissions given where the type is Benign

Text

Description automatically generated

Chart, bar chart

Description automatically generated

1. We have also plotted the dependent variable to make sure the distribution between 0s and 1s is balanced.

Chart, bar chart

Description automatically generated

1. We have a heatmap of correlation between different columns in the dataset. Even though the number of columns is huge, this helps in identifying whether there are more columns with correlation between them or not.

Table

Description automatically generated

1. We have also performed Principal Component Analysis (PCA) by fitting the data into 2 components. Once done, we can see how much amount of information can be provided by the 2 principal components after the data has been projected to a lower dimensional space. When we plot the 2 components along with the different labels in dependent variable, can observe the 2 classes benign and malicious when projected to a 2-D space are separable to a certain extent.
2. Another observation would be compared benign class, malicious class is more separated and toward the right and top of the spread.

Chart, scatter chart

Description automatically generated

## Wrangling 1

1. Since there are no missing values, no imputation technique is required to handle it.
2. To remove the columns which are correlated, in this approach we have set the threshold as 0.4. This means any combination of columns which have correlation greater than or equal to 0.4, one of them will be removed.
3. To find the important features, we are going with an approach of Univariate selection.
4. Here we will find out the important features will be found out by mapping each independent variable will be mapped the dependent column to find the importance of the column. The function used to find the importance will be chi squared tests.
5. The features having higher scores will be more related to the dependent features and hence chosen as features for the model.

## Wrangling 2

1. Since there are no missing values, no imputation technique is required to handle it.
2. To remove the columns which are correlated, in this approach we have set the threshold as 0.6. This means any combination of columns which have correlation greater than or equal to 0.6, one of them will be removed.
3. To find the important features, we are going with an approach of ExtraTreesClassifier method of selection.
4. Here ExtraTreesClassifier method will help to give the importance of each independent feature with a dependent feature. Feature importance will give you a score for each feature of your data, the higher the score more important or relevant to the feature towards your output variable.

## Feature Selection

The steps mentioned in Wrangling 1 will be considered as steps for Feature Engineering Technique 1 and the steps mentioned in Wrangling 2 will be considered as steps for Feature Engineering Technique 2.

## Results of Feature Engineering Technique 1

1. As a first step in this approach, we are removing all the independent variables which have a correlation greater than 0.4.
2. There are 57 independent variables removed as a part of this step. Below is the list of columns removed as a part of this step.

Text, letter

Description automatically generated

1. The next step is to find the important find the important features which add value to finding the dependent variable.
2. To do so, we are going with sklearn feature selection. We are using the chi square in SelectKBest features to get the importance scores for all the remaining independent variables.
3. Once we have the scores, we select all the independent variables which have an importance score of greater than 0. Once we do so, the shape of the data variable is 398 rows x 88 columns.
4. The top 10 important features and their scores are as shown below.

Text

Description automatically generated

Chart, bar chart

Description automatically generated

## Results of Feature Engineering Technique 2

1. As a first step in this approach, we are removing all the independent variables which have a correlation greater than 0.4.
2. There are 41 independent variables removed as a part of this step. Below is the list of columns removed as a part of this step.

Text, letter

Description automatically generated

1. The next step is to find the important find the important features which add value to finding the dependent variable.
2. To do so, we are going with ExtraTreesClassifier approach. This class implements a meta estimator that fits several randomized decision trees on various sub-samples of the dataset and uses averaging to improve the predictive accuracy and control over-fitting. Once we have the scores, we select all the independent variables which have an importance score of greater than 0. Once we do so, the shape of the data variable is 398 rows x 88 columns.
3. The top 10 important features and their scores are as shown below.

A picture containing text, indoor

Description automatically generated

Chart, bar chart

Description automatically generated

## Modelling

1. In the initial phase to select the right algorithms for the given dataset, we have chosen a set of 6 algorithms to run the data through. The top 2 algorithms with the best accuracy will be chosen to implement the feature engineering techniques mentioned above.
2. The set of algorithms chosen initially are:
   1. Logistic Regression
   2. Multinomial Naive Bayes
   3. Decision Trees Classifier
   4. Random Forest Classifier
   5. XGBoost
   6. SVC
3. Below are the metrics for each of the algorithms:

Text

Description automatically generated Text

Description automatically generated

1. Based on the accuracy scores seen above, we can see that Logistic Regression, Random Forest Classifier & XGBoost perform the best with the given data.
2. For our purposes, we have chosen Random Forest Classifier and XGBoost as ML1 & ML2.

## Results of ML Technique 1

#### ML1 + FE1

1. We have chosen Random Forest as our ML1. So, when we implement feature engineering technique 1 with it, we get the metrics as mentioned below.
2. Accuracy: 0.925
3. Train Classification Report

Table

Description automatically generated

1. Test Classification Report

Table

Description automatically generated

1. Train Confusion Matrix

A picture containing text

Description automatically generated

1. Test Confusion Matrix

A picture containing chart

Description automatically generated

#### ML1 + FE2

1. We have chosen Random Forest as our ML1. So, when we implement feature engineering technique 2 with it, we get the metrics as mentioned below.
2. Accuracy: 0.925
3. Train Classification Report

Table

Description automatically generated

1. Test Classification Report

Table

Description automatically generated

1. Train Confusion Matrix

A picture containing text

Description automatically generated

1. Test Confusion Matrix

A picture containing chart

Description automatically generated

## Results of ML Technique 2

#### ML2 + FE1

1. We have chosen XGBoost as our ML2. So, when we implement feature engineering technique 1 with it, we get the metrics as mentioned below.
2. Accuracy: 0.8875
3. Train Classification Report

Table

Description automatically generated

1. Test Classification Report

Table

Description automatically generated

1. Train Confusion Matrix

A picture containing text

Description automatically generated

1. Test Confusion Matrix

A picture containing text

Description automatically generated

#### ML2 + FE2

1. We have chosen XGBoost as our ML2. So, when we implement feature engineering technique 2 with it, we get the metrics as mentioned below.
2. Accuracy: 0.9
3. Train Classification Report

Table

Description automatically generated with medium confidence

1. Test Classification Report

Table

Description automatically generated

1. Train Confusion Matrix

A picture containing text

Description automatically generated

1. Test Confusion Matrix

A picture containing graphical user interface

Description automatically generated

## Comparison

|  |  |  |
| --- | --- | --- |
| Machine Learning Technique | Feature Engineering Technique | Accuracy |
| Random Forest (ML1) | Correlation (0.4) + SelectKBest (FE1) | 0.925 |
| XGBoost (ML2) | Correlation (0.4) + SelectKBest (FE1) | 0.8875 |
| Random Forest (ML1) | Correlation (0.6) + ExtraTreesClassifier (FE2) | 0.925 |
| XGBoost (ML2) | Correlation (0.6) + ExtraTreesClassifier (FE2) | 0.9 |

## Conclusion

1. Based on the different algorithms used to initially select the top 2 and then pairing it with the different feature engineering approaches, we can see that Random Forest classifier fits the best for the given data.
2. Random Forest Classifier performs best with both feature engineering techniques. It seems to do well on the validation data where 20% of the total data not seen during the training phase is used check performance when compared to other models.
3. Hence, it really does answer the problem of identifying the application as malware or not based on the permissions required by the application with accuracy being greater than 90%.
4. With more amount of data in rows, we could probably even increase the accuracy getting it closer to 100%.
5. The model with existing metrics can surely be used in real time scenario.

## Recommendation/Suggestion

1. The model can be put into practice in real time. However, our recommendation is that there would be a need for feedback loop, where any prediction below a certain threshold must be raised for human validation.
2. The validated/corrected data should be used to train the model further on the predictions it missed or had a low confidence for thus completing the feedback loop.