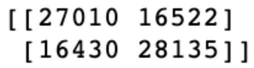
Quantitative Analysis of Results

For each of the three models, we looked at the R-squared value, confusion matrix, classification table, and receiver operating characteristics (ROC) curve.

1. Logistic Regression

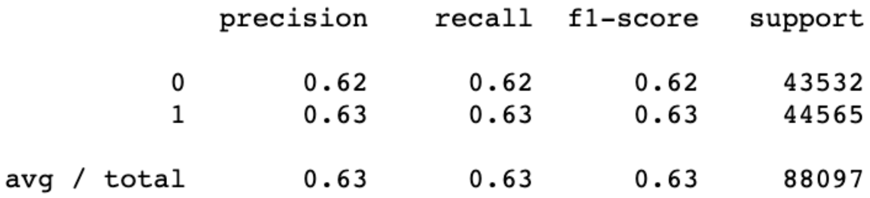
The logistic regression produced an R-squared value of 0.08. 8% of the data within the validation set fit within the logistic regression model, not very good. The confusion matrix in Figure 8-1 shows that the number of observations that were correctly and incorrectly predicted. Approximately, half of the cases were either predicted incorrectly or correctly.



```
[[27010 16522]
 [16430 28135]]
```

Figure 8-1. Confusion matrix for logistic regression.

The classification report below shows quite similar results to the classification table (Figure 8-2). We see that the precision in predicting class label 0 is 0.62, while the precision for 1 is 0.63. The overall weighted average is 0.63, furthering proving that the model is accurate about 63% of the time.

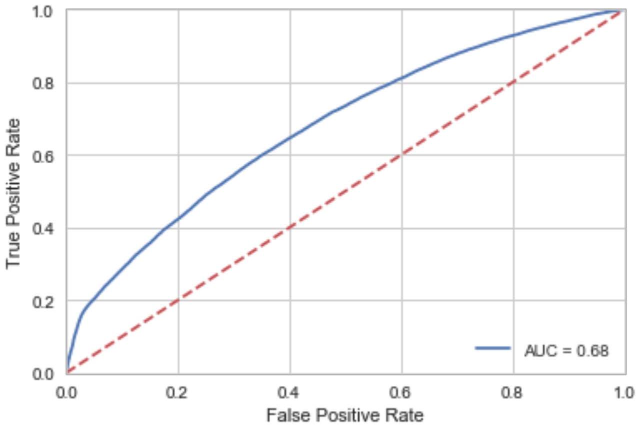


```
             precision    recall  f1-score   support

          0       0.62      0.62      0.62     43532
          1       0.63      0.63      0.63     44565

avg / total       0.63      0.63      0.63     88097
```

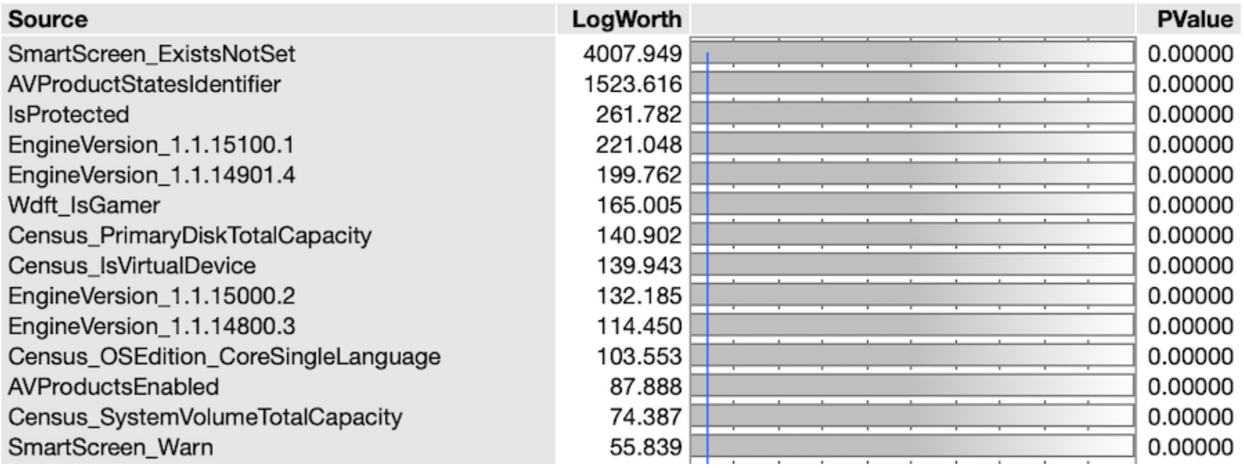
Figure 8-2. Classification table for logistic regression.

Figure 8-3 displays the ROC and shows an area under the curve of 0.68. Not as high as statisticians would like, but still feasible. Thus, we further examined other models to see if there is another way to find predict whether machine will be hit by malware or not.

Figure 8-3. ROC curve for logistic regression.



Further analyzing the model's infrastructure, we found that many of the variables were statistically significant to the dependent variable: Has Detections (Figure 8-4). We saw that the top three variables are"SmartScreen_ExistsNotSet",“AVProducctStatesIdentifie,”,and“IsProtected.” SmartScreen_ExistsNotSet means that the smart screen is not registered under the given criteria: Microsoft, Windows, System, etc. Further down the list, although not shown, we observed variables like Firewall and Is Protected, both of which are shown to be negatively related to the target variable 1, which is malware will hit the machine. This makes sense because if the firewall is one or a machine is protected, the response will be “1”. Since the relationship is negative, that means the dependent response will have a higher probability of being “0”, no malware detected.

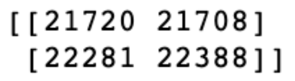


Source	LogWorth	PValue
SmartScreen_ExistsNotSet	4007.949	0.00000
AVProductStatesIdentifier	1523.616	0.00000
IsProtected	261.782	0.00000
EngineVersion_1.1.15100.1	221.048	0.00000
EngineVersion_1.1.14901.4	199.762	0.00000
Wdft_IsGamer	165.005	0.00000
Census_PrimaryDiskTotalCapacity	140.902	0.00000
Census_IsVirtualDevice	139.943	0.00000
EngineVersion_1.1.15000.2	132.185	0.00000
EngineVersion_1.1.14800.3	114.450	0.00000
Census_OSEdition_CoreSingleLanguage	103.553	0.00000
AVProductsEnabled	87.888	0.00000
Census_SystemVolumeTotalCapacity	74.387	0.00000
SmartScreen_Warn	55.839	0.00000

Figure 8-4: Screenshot of statistically significant models ranked by p-values within logistic regression.

2. Random Forests

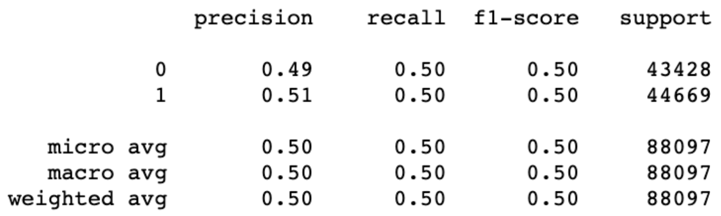
The confusion matrix shown in Figure 9-1, we saw that there were 22,281 observations that correctly predicted machines with malware and 21,708 observations that correctly predicted machines that were not yet infected with malware.



```
[[21720 21708]
 [22281 22388]]
```

Figure 9-1. Confusion matrix for random forests.

The classification table was also designed to view the precisioness of the model (Figure 9-2). Just as we saw in the confusion matrix, we see that the precision rate is approximate 0.50. Only about half of the cases can be correctly predicted with this particular model.



```
             precision    recall  f1-score   support

           0       0.49      0.50      0.50     43428
           1       0.51      0.50      0.50     44669

   micro avg       0.50      0.50      0.50     88097
   macro avg       0.50      0.50      0.50     88097
weighted avg       0.50      0.50      0.50     88097
```

Figure 9-2. Classification table for random forests.

Analyzing the ROC curve for this model, we see that the blue line curves above the diagonally dotted red line of no discrimination - hinting towards a good model (Figure 9-3). The ROC curve showed an area under the curve of 0.66. A little bit lower than the logistic regression, and not as good as the neural networks ROC results.

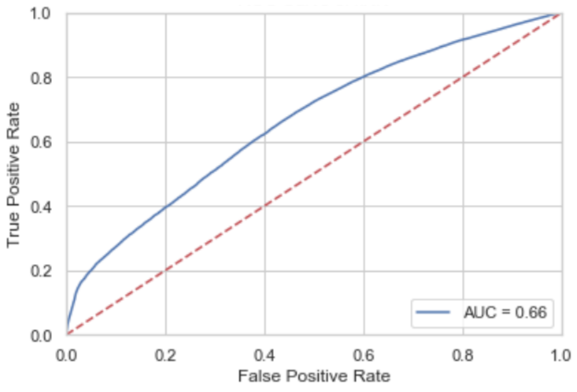
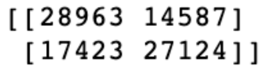


Figure 9-3. ROC curve for random forests model.

3. Neural Network

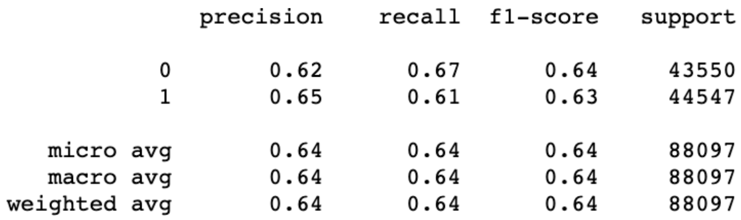
The confusion matrix for this model is shown in Figure 10-1. From the confusion matrix we learned that our model can correctly identify machines that was hit with malware 65% of the time. The model correctly identified 28,963 cases of infectious malware and 27,124 cases of uninfected machines.



```
[[28963 14587]
 [17423 27124]]
```

Figure 10-1. Confusion matrix of neural network.

Another cross-validation analysis we performed on the validation set is a classification report shown in Figure 10-2. The classification report displays the precision, recall, f1-score, and support in terms of the class labels 1 and 0. There is a 62% precision when prediction values 0 (machines without malware) and a 65% precision when prediction machines hit with malware. The micro and macro averages were taken into consideration and a weighted average of 0.64 is recorded.



```
              precision    recall  f1-score   support

           0       0.62      0.67      0.64     43550
           1       0.65      0.61      0.63     44547

   micro avg       0.64      0.64      0.64     88097
   macro avg       0.64      0.64      0.64     88097
weighted avg       0.64      0.64      0.64     88097
```

Figure 10-2. Classification report for logistic regression.

Lastly, a ROC curve is produced to analyze the false positive and true positive rates of predicting the binary result of our dependent variable as seen in Figure 10-3. The overall area under the curve for both of the class labels is 0.69, meaning that there is a 69% accuracy. With an area under the curve that high, we can consider this a relatively dependable model.

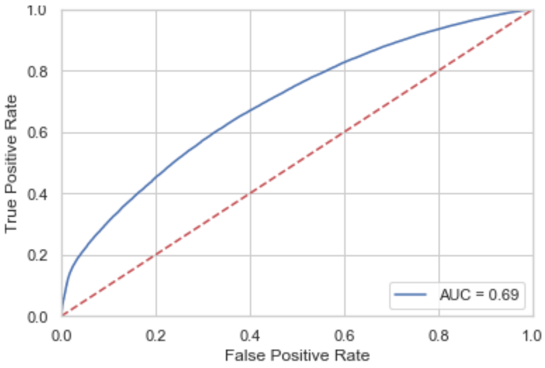


Figure 10-3. ROC curve for neural networks model.

Conclusion

Every organization that handles customer’s private credentials have to be more cautious about encountering malicious programs like malware. As technology grows and becomes more automated, hackers also grow too. Hackers are becoming more innovative when it comes to intercepting into other computer systems. Thus, being able to build a model that can predict whether a machine will be infected with malware or not based off of historical data can immensely help secure the safety of everyone's personal lives.

In this paper, we produced three different types of models that explored the dataset in explicitly different ways. We first performed the most generic model - logistic regression. The overall area under the curve for this model came out to be 68%, slightly better than half. The random forests model bases its model off of multiple decision trees and produced an AUC value of 66%. However, both the logistic regression and random forests model did not perform as well as the neural network model. The neural network model proved to have AUC value as 69%. Thus, we can conclude that out of all the models that we were able to create the neural network model could be used to help detect malicious malware given that information relating to the attributes used in the model can be provided.

Through these analysis, we learned that running any type of model with over 100 variables takes a lot of computing power. We also discovered that in order to comput up with any sort of visualizations it requires certain packages such as Keras. There was a lot of trial and error when it came to running our models because we saw that the models would originally work on a smaller subset of our data, but not the complete set, we faced this issue with our KNN and SVM model. Thus, throughout the month we researched why and minor appropriate adjustments to make our code run successfully. However, the R-squared values for our models were concerningly low.

Of the questions that we asked in the beginning of the paper, we were able to answer what variables are related to the dependent variable: Has Detections. We also learned which model would perform the best. As we mentioned before, the neural network model performed the best. We believe this is the case because a neural network is dedicated to larger datasets like ours and utilizes hidden layers to help categorize a machine into the right binary response. We also learned what components of a machine are more likely to help defend it from malware, some of these variables include: Protection, Firewalls, certains of engine versions, space capacity, RAM, and more.

Although a 69% accuracy rate is quite good, there must be other models that can be implemented to possibly provide a better statistical summary. We must also remember that our R-squared value was quite low, which means there must be other models that could reveal a more reliable score. Other models that we also looked into are K-Nearest Neighbors and Decision trees. The K-Nearest neighbors would classify the observations be looking at the similarities between the all the attributes of the dataset and the decision trees would be just produce a single tree rather than multiple. Other researchers have also looked into building more complicated models via machine learning and deep learning. Apart from running different models, we could possibly performed different methods to clean up the data. 83 variables that turned into 161 variables after creating dummy columns was a challenge to deal with; thus, we believe that there are alternative ways to have evaluated the variables in the original dataset.