Microsoft malware Prediction

Nirmalendu Prakash

Business Problem

The malware industry continues to be a well-organized, well-funded market dedicated to evading traditional security measures. Once a computer is infected by malware, criminals can hurt consumers and enterprises in many ways.

Can we develop techniques to predict if a machine will soon be hit with malware?

Source- <https://www.kaggle.com/c/microsoft-malware-prediction/overview>

Data

Machine properties and machine infections was generated by combining heartbeat and threat reports collected by Microsoft's endpoint protection solution, Windows Defender.

Source-<https://www.kaggle.com/c/microsoft-malware-prediction/data>

3 files-train.csv, test.csv and sample\_submission.csv

Taking subset of data

The dataset for this problem is huge; reading random 50,000 rows from train.csv

Approach1

Feature Extraction

Only 8 out of 81 features are numerical.

1. Drop columns with little variation in values or dropping columns with many missing values. 49 features left after this.
2. Imputing data with mean for numerical features and median for categorical features

For the categorical features if the number of unique values in each is less than 20, do one-hot encoding otherwise drop the feature. The result is 171 features

1. For non binary columns, do standard normalization.
2. Since there are too many features for classification, do a PCA to reduce dimensions
3. Using a scree plot to choose number of dimensions, 10 dimensions chosen

Training

Divided dataset into train and test and used MLP, Logistic Regression, SVM, Random forest and an ensemble of MLP, Logistic Regression and SVM models.

Highest accuracy achieved is 58%

Approach2

Feature Extraction

1. Increased dataset to half a million
2. Drop columns with little variation in values or dropping columns with many missing values. 49 features left after this.
3. Imputing data with mean for numerical features and median for categorical features

For the categorical features if the number of unique values in each is less than 20, do one-hot encoding otherwise replace with frequency value. The result is 171 features

1. For non binary columns, do standard normalization.

Training

Using Light Gradient Boost

1. Discarding missing value features, using the imputed csv above. Gives an accuracy of 60%
2. Using all features (with missing and little variation columns), as it ignores missing values when computing the split and then allocates all data with missing values to whichever side of the split reduces the loss more.

Source- <https://mlexplained.com/2018/01/05/lightgbm-and-xgboost-explained/>

Accuracy is 61.6%

K Fold Cross validation and light GBM

Accuracy increases to 66.8% wit AUC 67.7%