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

STEPS

1. Read chunks of data to identify data types
2. Read full dataset
3. Remove features with more than 50% missing data
4. Remove features with little variation in values
5. Label categorical variables and replace missing value with most frequent value for each feature
6. For numerical values, fill missing values with median value for each feature
7. Divide the dataset into 80% train and 20% validation set
8. Train RandomForest
9. Train LGBM
10. Use more accurate model on test set

Training

RandomForest

Validation accuracy is 65%

Using Light Gradient Boost

Validation auc is 74%

Using LGB model on test data gives an accuracy of 62%

Feature Importance:

Plotting the feature importance shows that region identifier has highest contribution, census features have some contribution and os related features have very little or no contribution.