International Institute of Information Technology Bangalore

FastHEAL Malware Detection ML project

Team Panacea

Kedar Deshpande IMT2020523
Ankrutee Arora IMT2020034

Includes all the slides since first milestone.

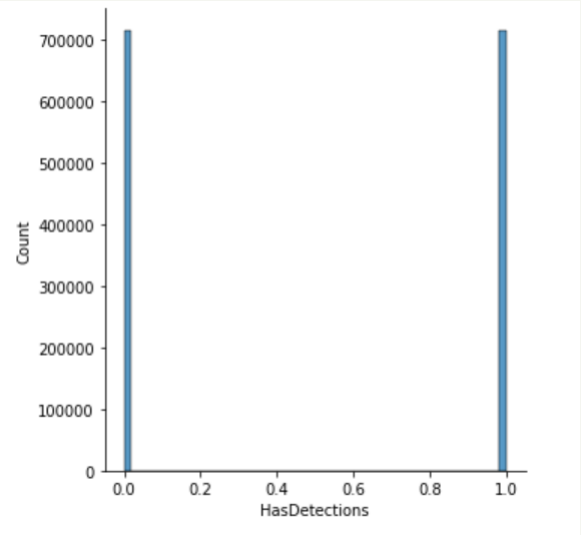
About the data

Number of rows: 1427437

Number of columns: 84

Balanced dataset

Null values present in both train and test datasets



Using the given data we need to predict the probability of a system having a malware presence.

Therefore, we need to fill the HasDetections column of test data.

Preprocessing and EDA

Preparing clean data to train models

Null and ? values

- There were many null values, but no ? in the dataset.
- These columns were dropped because of high number of null values

[DefaultBrowsersIdentifier, OrganizationIdentifier, PuaMode,SmartScreen, Census_ProcessorClass, Census_InternalBatteryType, Census_IsFlightingInternal, Census_ThresholdOptIn, Census_IsWIMBootEnabled]

- Rows were dropped with null values (<10% values from the column were null)
- Were planning to replace null with mean/mode/median for the coulmns with medium amount of null values, but no such cases.
- Checked for duplicate rows (as we removed some columns).
- Checked that test and train datasets have the same columns.

After this:
Rows=1149234
Columns=75

Modifications on test data

- It is to be noted that we cannot remove any row in test data.
- Hence for null values, instead of dropping the rows, we replaced them with mode.
- We thought that it is safer to assume missing information about the system to be the most common one found in the data.

Analysed each column

- We went through each of the remaining columns and analysed whether they should be label encoded, one hot encoded, kept as it is(numeric or binary), or dropped.

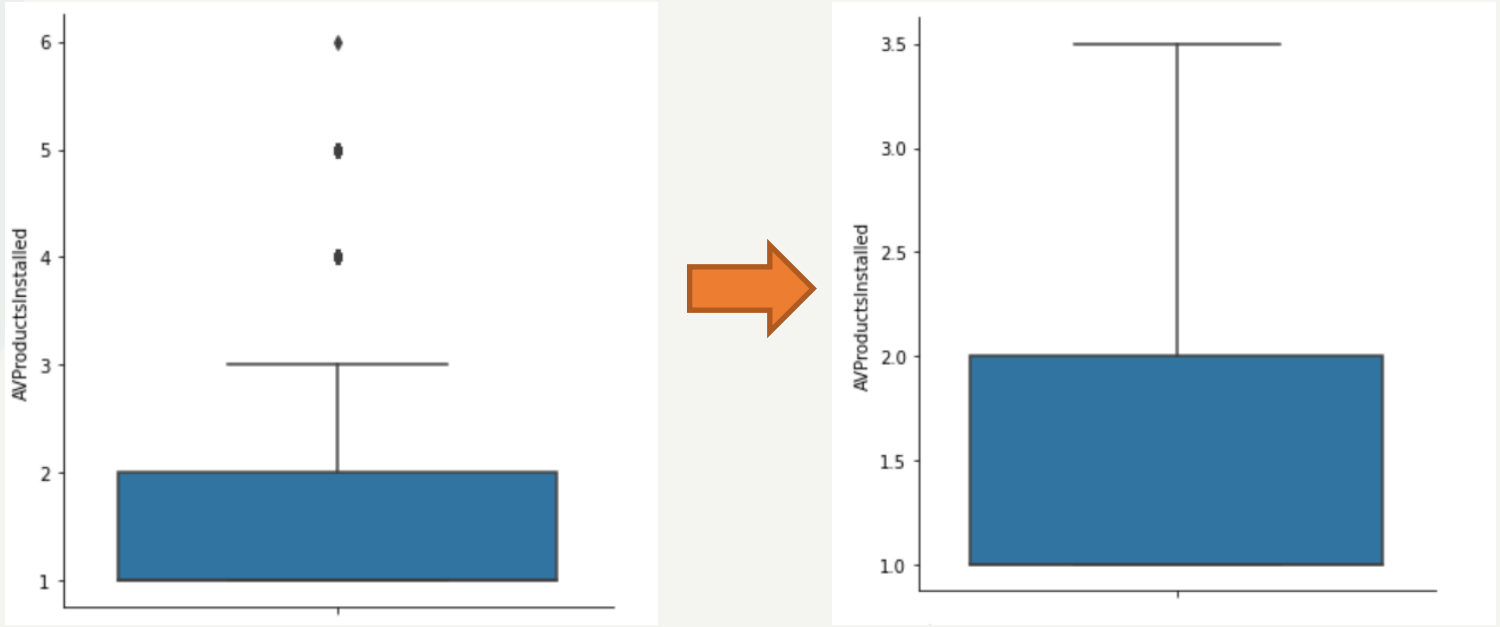
- **Label encoded columns**: EngineVersion, AppVersion, AvSigVersion, OsBuild, Census_OSVersion, Census_OSBuildNumber, Census_OSBuildRevision

- **One hot encoded columns**: Processor, OsSuite, OsPlatformSubRelease, SkuEdition, Census_MDC2FormFactor, Census_ProcessorManufacturerIdentifier, Census_PrimaryDiskTypeName, Census_PowerPlatformRoleName, Census_OSArchitecture, Census_OSBranch, Census_Edition, Census_OSInstallTypeName, Census_OSWUAutoUpdateOptionsName, Census_GenuineStateName, Census_ActivationChannel, Census_FlightRing

- These columns were kept as it is: AVProductsInstalled, AVProductsEnabled, Census_ProcessorCoreCount, Census_PrimaryDiskTotalCapacity, Census_SystemVolumeTotalCapacity, Census_TotalPhysicalRAM, Census_InternalPrimaryDiagonalDisplaySizeInInches, Census_InternalPrimaryDisplayResolutionHorizontal, Census_InternalPrimaryDisplayResolutionVertical, Census_InternalBatteryNumberOfCharges, Census_HasOpticalDiskDrive, Census_IsFlightsDisabled, Census_IsSecureBootEnabled, Census_IsVirtualDevice, Census_IsTouchEnabled, Census_IsAlwaysOnAlwaysConnectedCapable, Wdft_IsGamer
- All other columns were dropped because either they had >95% of same value or too many categories.
- ‘Unnamed: 0’ column is same as MachineIdentifier
- MachineIdentifier was not used for training.

Outlier Detection

- Found the outliers and brought them in bounds for all the numeric columns.
- Example:



Standardisation

- Standardised all the numeric columns
- Differences in the ranges would make a poor model.
- Standardisation makes sure that the columns are of similar ranges.

Correlation

- Created the correlation matrix for all columns.
- Observed that the correlations were very low with the output.
- These are some columns with highest correlation with output.

EngineVersion', 'AVProductsInstalled', 'Census_ProcessorCoreCount’, 'Census_PrimaryDiskTotalCapacity’, 'Census_TotalPhysicalRAM', 'Census_InternalPrimaryDiagonalDisplaySizeInInches’, 'Census_IsAlwaysOnAlwaysConnectedCapable’, 'Wdft_IsGamer', 'x0_x64', 'x0_x86', 'x0_Slate', 'x0_amd64', 'x0_x86'

- Finally

1149234 rows, 146 columns (MachineIdentifier and HasDetections) in training data.

Saved the train and test data as csv file for easier access to train models.

Model Training

Logistic Regression, Random Forest

Logistic Regression

- Trained the logistic regression model after train-test split.
- Test ROC AUC score = 0.599780674439361
- Public Scoreboard -> 0.58352

Random Forest

- n_estimators=200 (not hyperparameter-tuned)
- Trained the logistic regression model after train-test split.
- Overfitting observed- train-> 0.98, test-> 0.63
- Test ROC AUC score = 0.6033215587790373
- Public Scoreboard -> 0.64602

Milestone 2 starts here….

XG Boost

- Applied XGBoost Classifier to our dataset
- Tried many different n_estimators values like 100, 200, 300, 400, 500.
- Gave best public scoreboard results with n_estimators=100
- No overfitting. Train-> 0.661 Test-> 0.642
- Score = 0.65559

Change in pre-processing

- We changed our pre-processing a bit
- We did not one-hot encode any of the columns. Instead of that we label encoded them.
- Tree based models perform a bit badly when data is sparse (like in one hot encoded data).
- So now we tried some models on this dataset.

Random Forest

- Public Score = 0.64861
- Increased
- Overfitting

XG Boost

- Tried XGBoost for various n_estimators.
- Still best results were from n_estimators = 100

- Public Score = 0.66211
- No overfitting

LightGBM

- Tried LightGBM on our dataset with n_estimators=100, 200, 300, 400, 500
- Best parameter -> 300
- Public Score = 0.66598, Training score- 0.685 Test score-0.672
- No Overfitting
- **Best one till now**

Milestone 3 starts here..

Tuning the hyperparameters

- After this we tried different number of leaves and learning rates to try and improve our score.
- We tried increasing number of leaves to 50, 100, 150, 200, 250. We tried 0.01,0.001 learning rate. We also increased number of estimators.
- Here we got best results for n_estimators=1500, learning_rate=0.01, and no. of leaves=250.
- Public Score = 0.69561 Training Score-0.7215 Test Score-0.705

Changing Preprocessing

- We had removed some of the rows which had null values in the columns with <10% null values.
- Now instead of removing rows, we just filled the null values with mode of that column(same as we did for test dataset).
- We did see a small improvement in our scores. Played around a few hyperparameters.
- New best score- 0.6971 n_estimators=3000, learning_rate=0.02, leaves=250

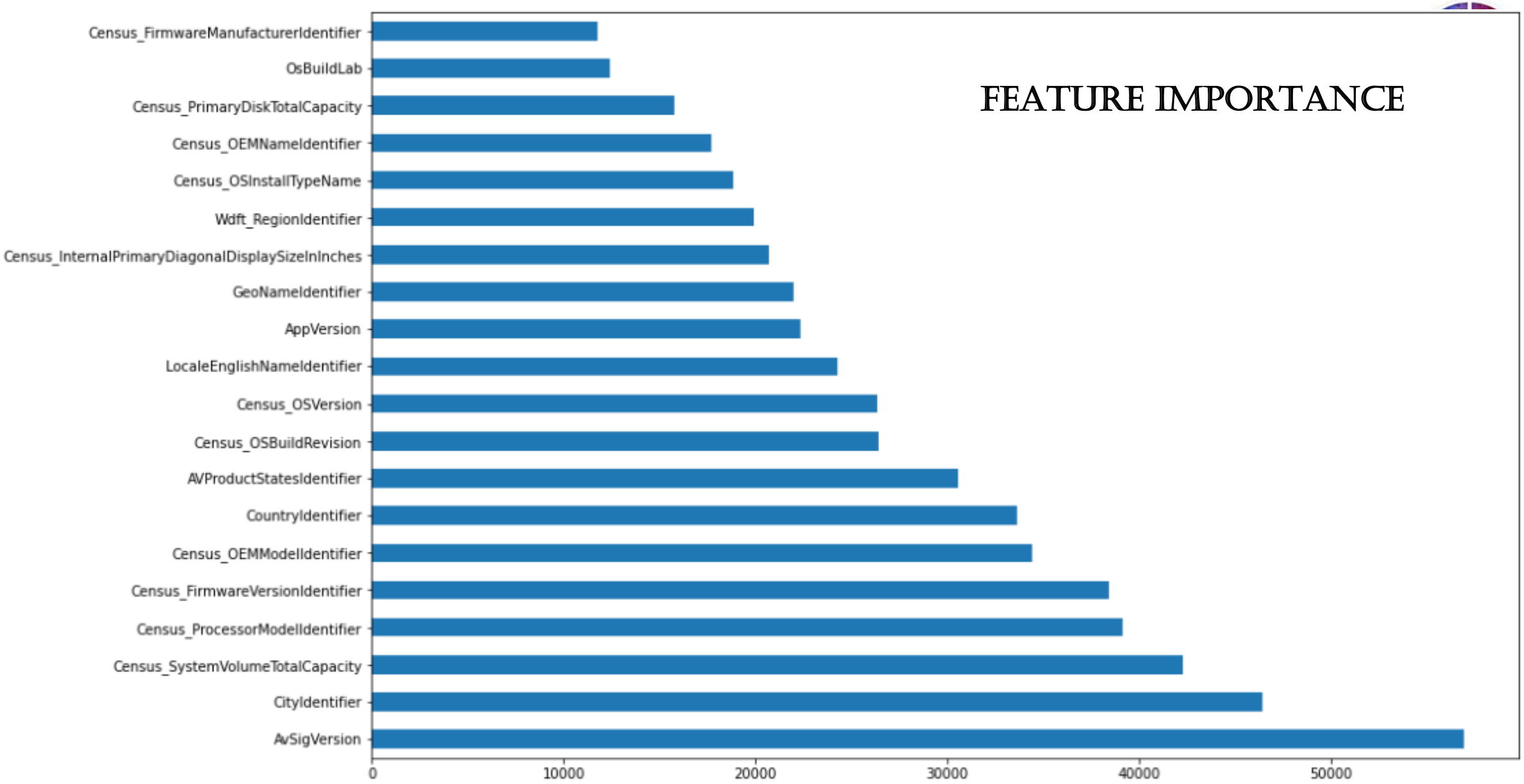
The Best Public Score

- Our best public score came when we included some more columns that we initially had dropped.
- Instead of dropping the columns with null values>30%, we now dropped above 90%. This step re-added 2 columns.
- Then we also included the identifier columns.
- Then added some columns with medium to high skew, removed only those with >98% skew.

What was the best score?

- These steps increased our score by a lot.
- Our final best score is : **0.72624**
- Training score-0.782 Testing score-0.7292
- n_estimators=3000, learning_rate=0.01, leaves=250

FEATURE IMPORTANCE



Thanks !

International Institute of Information Technology Bangalore

26/C, Electronics City, Hosur Road,

Bengaluru – 560 100, Karnataka, India

www.iiitb.ac.in

https://www.facebook.com/IIITBofficial/

https://www.linkedin.com/school/iiit-bangalore/

https://www.instagram.com/iiitb_official/

https://twitter.com/IIITB_official

https://www.youtube.com/user/iiitbmedia