Malware Classification

GA SG DSI 26
Capstone Project Presentation

Kwek Jun Hong

01

Introduction and Problem Statement

Introduction

Malware are intrusive software developed to steal data and damage or destroy computers and computer systems.

Malware includes viruses, worms, Trojan viruses, spyware, adware, and ransomware.

Introduction

Malware
Industry

Cost of Malware attacks estimated to reach USD 10.5 trillion by 2025 from USD 3 trillion in 2015.

Ransomware is the major malware threat to users and businesses.

Shift to remote working conditions have contributed to the increase in malware attacks.

Problem Statement

Role as a Data Analyst in a PC operating system company.

Develop a model that is able to:

- Predict if a machine has being infected by malware
- Identify which features are important to malware prediction
- Propose recommendations accordingly.

02

Import Data and Data Cleaning

Import Data – First Look

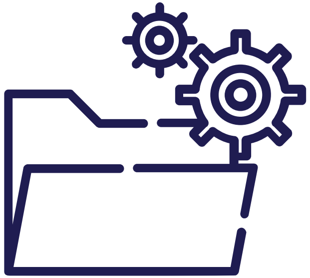
- Datasets obtained from Kaggle competition (Microsoft Malware Prediction)

- Train dataset: 8.9 million rows with 83 features
- Test dataset: 7.8 million rows with 82 features (without the dependent variable)

- Dependent variable is whether malware is detected on the machine. Malware is not classified into the different types of malware within this dependent variable.

Import Data – First Look

- Features in the datasets include the different parameters of the computer that are recorded.
    - Country, City, Language identifier
    - Organization identifier
    - Antivirus software identifier (enabled, installed, product state)
    - Hardware related identifier (processor, system storage size, type of system)
    - Software related identifier (Engine version, OS version, OS install type)
- Features comes in both numerical and object data types.
- Features are all nominal categorical features



Import Data – Dependent Variable

- **Class balance of the dependent variable (‘HasDectection’)**
  - Class 0, No Malware: 0.50
  - Class 1, Has Malware: 0.49

Data Cleaning – Duplicates & Null Values

Duplicates

- No duplicate values within the train dataset

Null Values

- Features with more than 80% null values shall be dropped
- Features shall also be dropped if deemed to be irrelevant
- Remaining features shall be imputed with the mode of the category

Data Cleaning – Duplicates & Null Values

High Cardinality

- Features that have a lot of unique values will suffer from the curse of dimensionality
- Group the values outside a threshold of 70% of the data as a separate group (‘Other’ or 0.0 depending on data type)

```
Name of feature:  census_mdc2formfactor
Number of unique values:  13
Notebook        0.641521
Desktop         0.218695
Convertible     0.045438
Detachable      0.033429
AllInOne        0.032739
PCOther         0.015687
LargeTablet     0.007524
SmallTablet     0.003519
SmallServer     0.000967
MediumServer    0.000379
Name: census_mdc2formfactor, dtype: float64
```

```
Name of feature:  census_mdc2formfactor
Number of unique values:  3
Notebook    0.641521
Desktop     0.218695
Other       0.139784
Name: census_mdc2formfactor, dtype: float64
```

Data Cleaning – Duplicates & Null Values

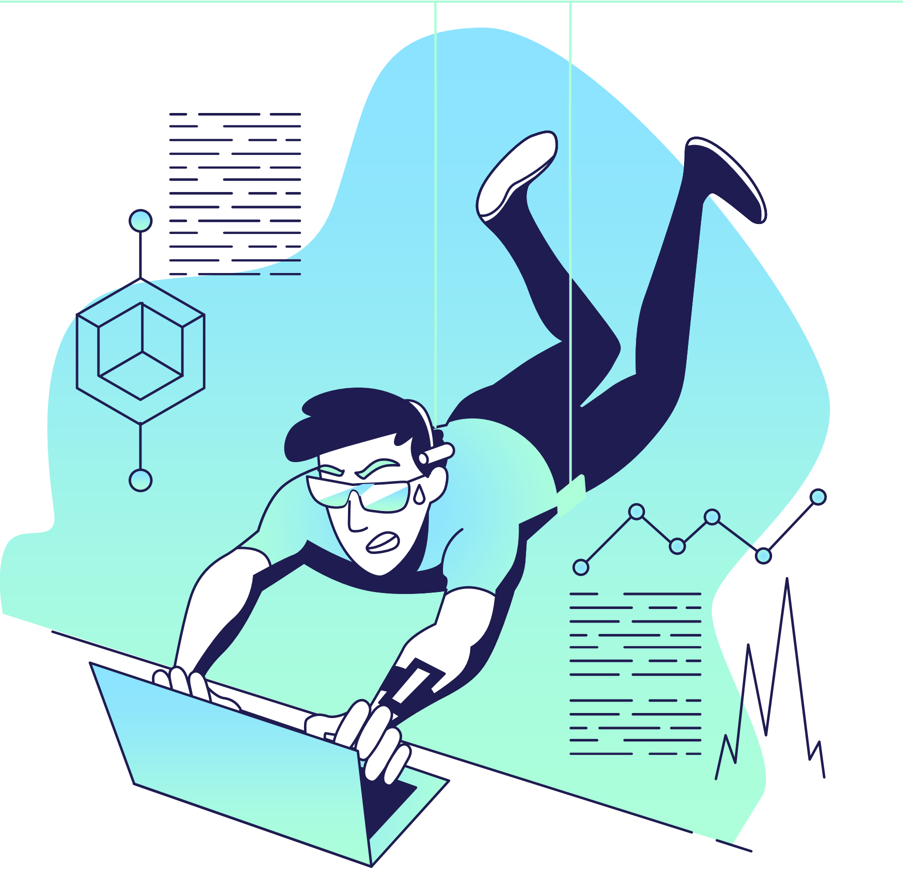
Single Values

- Features with more than 90% of the data in a single value shall be dropped

Number of features dropped: **50**

03

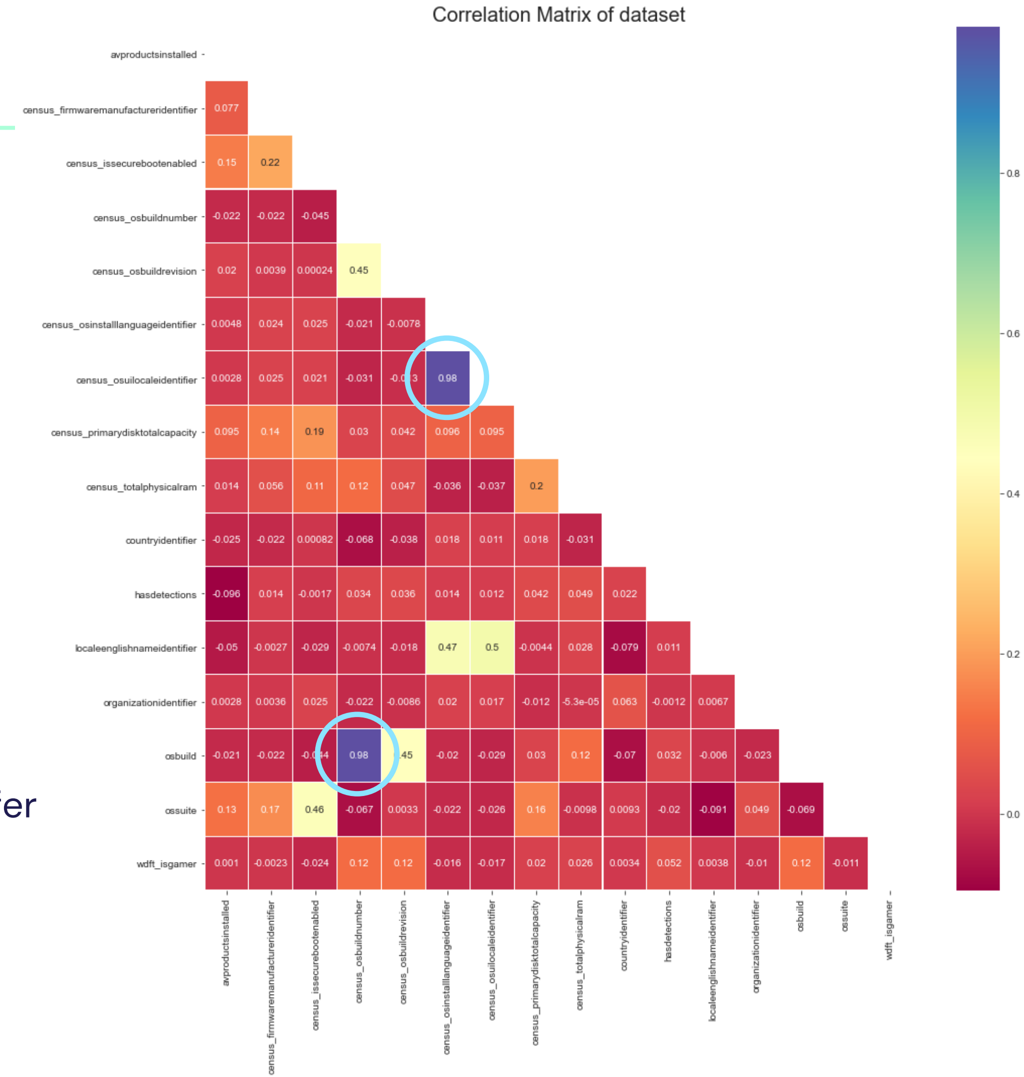
EDA



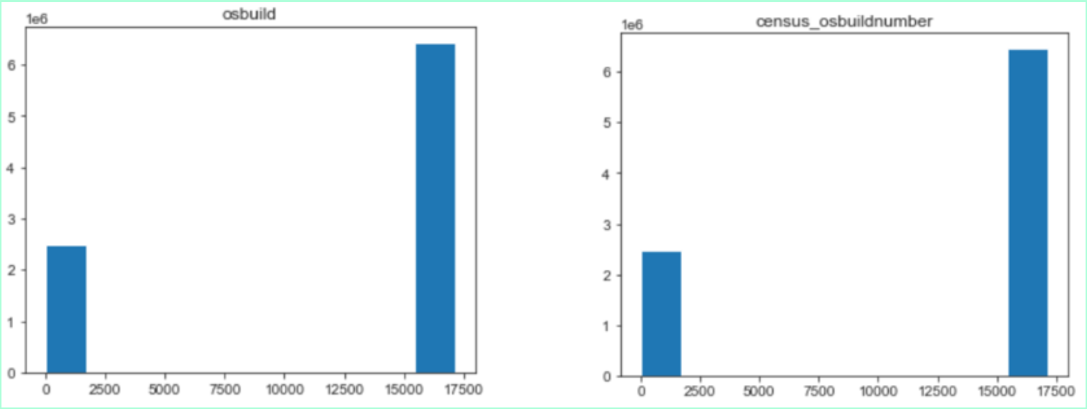
EDA – Correlation Matrix

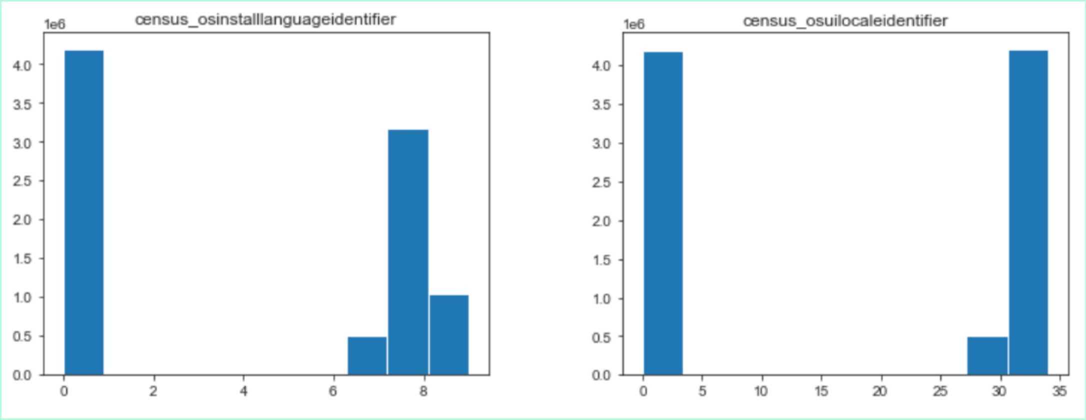
- Not useful for categorical features
- Used to only identify the collinear features in the dataset

- Collinear features include:
  - OsBuild and Census_OsBuildNumber
  - Census_OsUiLocaleIdentifer and Census_OsInstallLanguageIdentifer

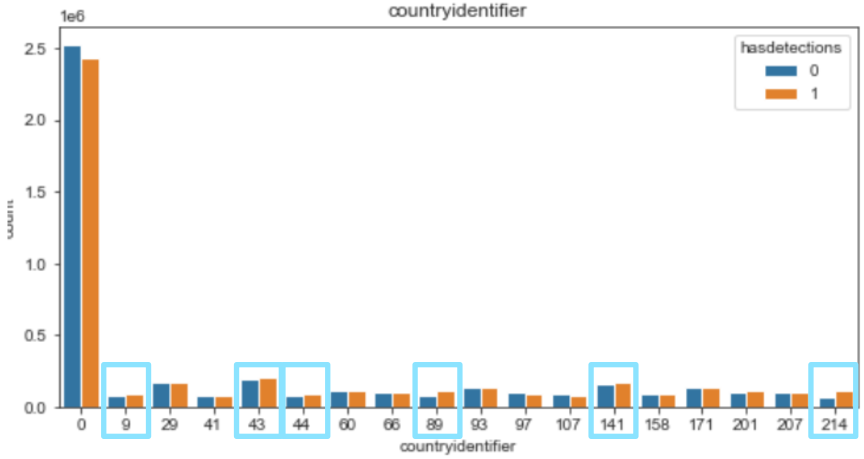


EDA – Collinear Features





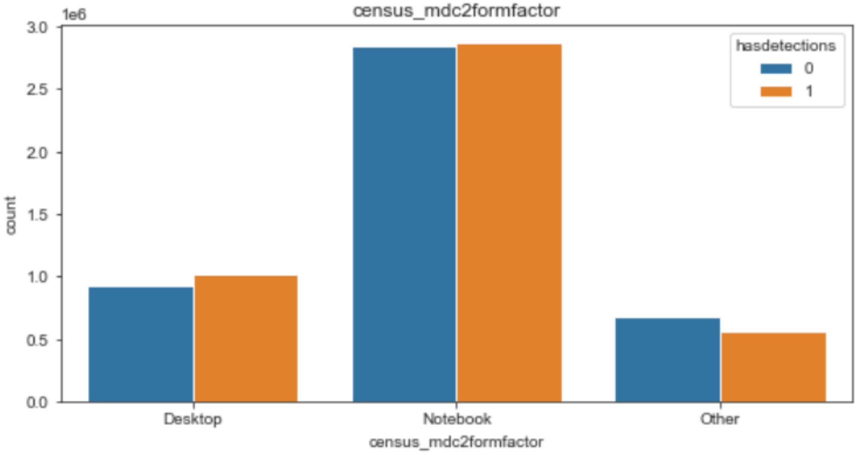
EDA



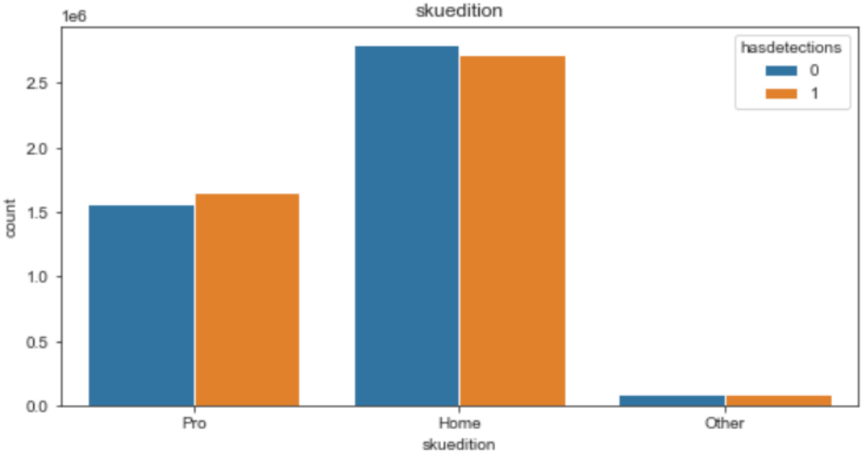
- 0 – All other industries
- 18 – Cosmetic
- 27 - Retail

- Countries are masked for confidentiality in dataset.
- Majority class is 0 which is all other countries.

EDA

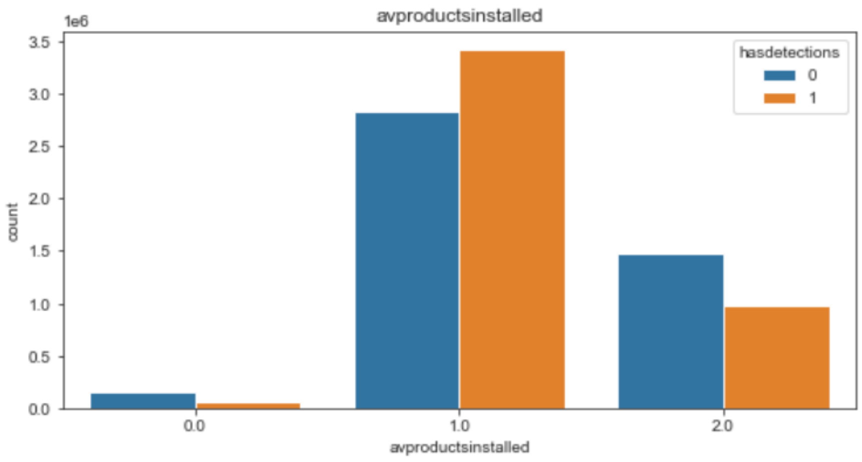


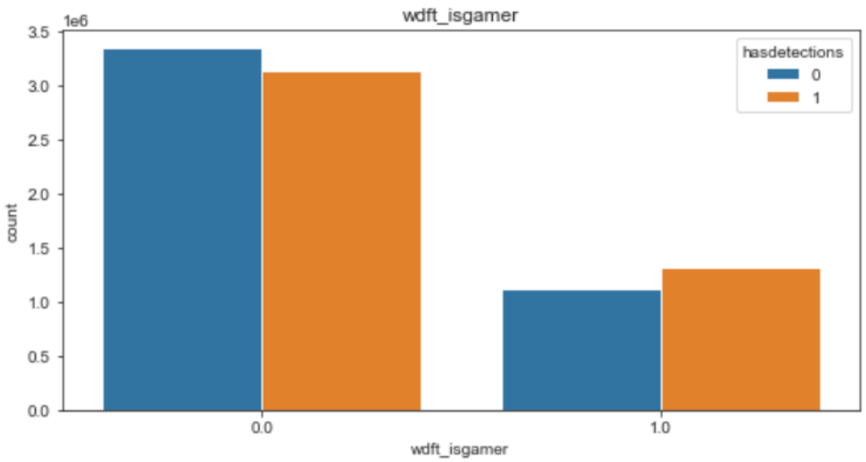
- Desktop users has more malware detection than Notebook and Other



- Pro edition has more detection compared to Home edition of the OS

EDA



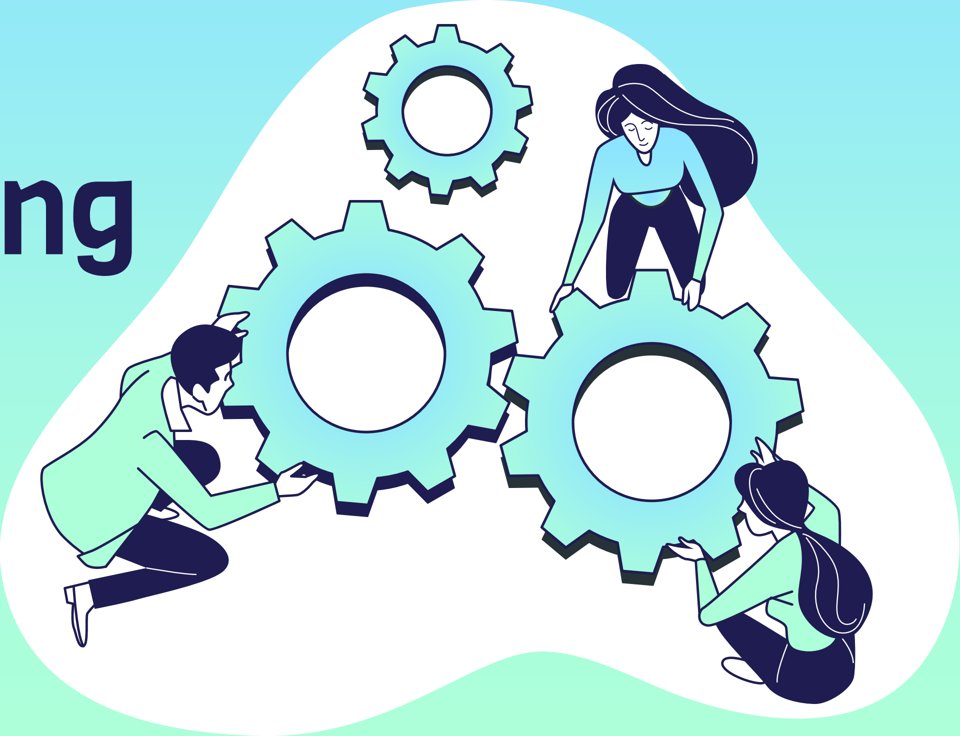


- Installing 1 antivirus gives a higher malware detection rate than 2 antivirus

- Being identified as a gamer (1.0) gives a higher malware detection rate than a non gamer.

04

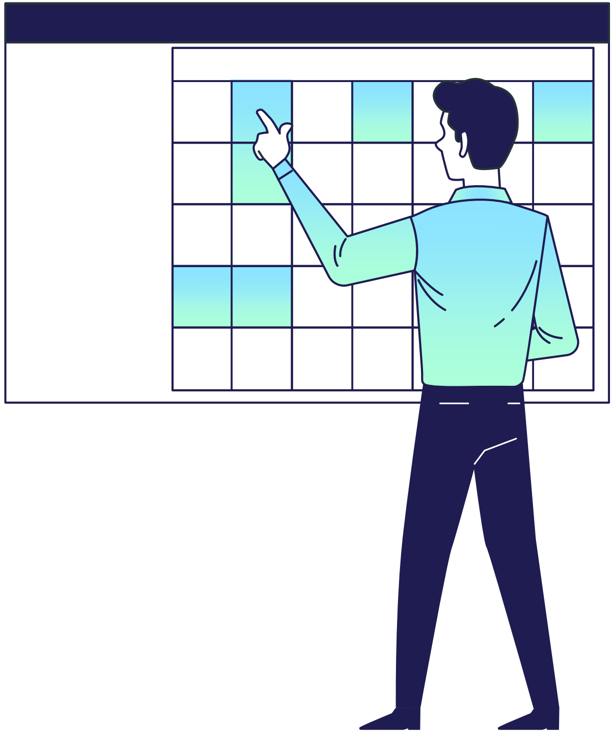
Model Data Pre-processing



Random Sampling of Data

- Take 100,000 rows from the train dataset (8.9 million rows) due to limitations in hardware and time constraint

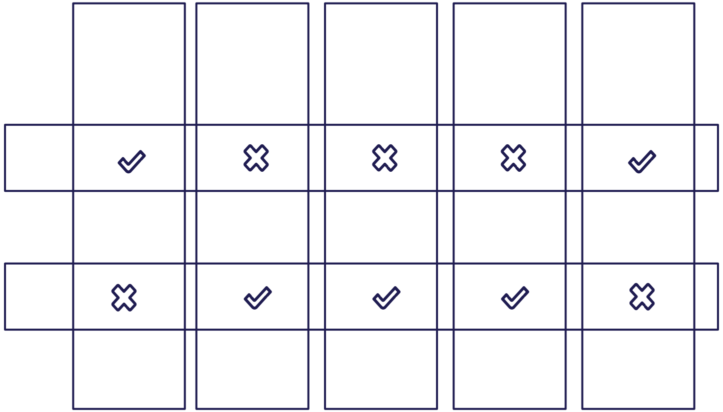
- Check that the balance of the dependent variable stays the same after the random sampling



Model Data Preparation and Pre-processing

- Create a train and test split from the train dataset based on the default split

- Features are categorical
  - Will not be Standard Scaled even if it is numerical and
  - Features will all be One Hot Encoded before modeling



05

Modelling

You could enter a subtitle
here if you need it

Models Used

Models	
Baseline Model	Balance of Dependent Variable
Model 1	Logistic Regression
Model 2	K-Neighbors Classifier
Model 3	Random Forest Classifier
Model 4	Light GBM
Model 5	Keras Sequential Neural Network

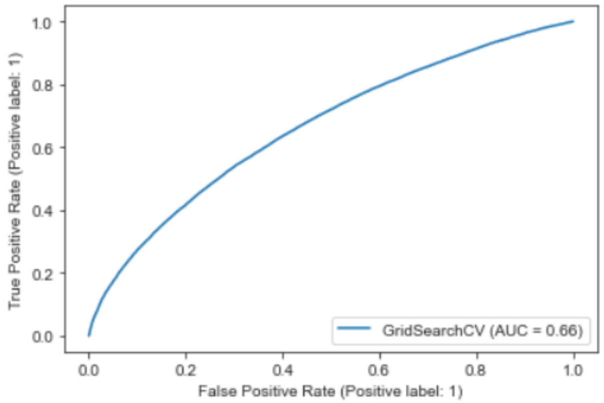
- Binary Classification problem with balanced data
- Baseline Model Score: 0.50
  - Class 1: Malware present in machine
  - Class 0: Malware not present in machine
- Classification Metrics:
  - ROC AUC
  - Recall *(True Positive/ Total Actual Positive)*

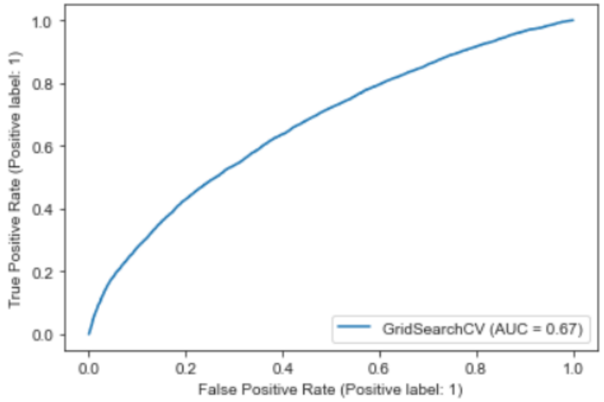
Model 1: Logistic Regression

- Train AUC: 0.665
- Test AUC: 0.668
- Recall: 0.593

Confusion Matrix

	Predicted Negative	Predicted Positive
Actual Negative	8075	4413
Actual Positive	5095	7417





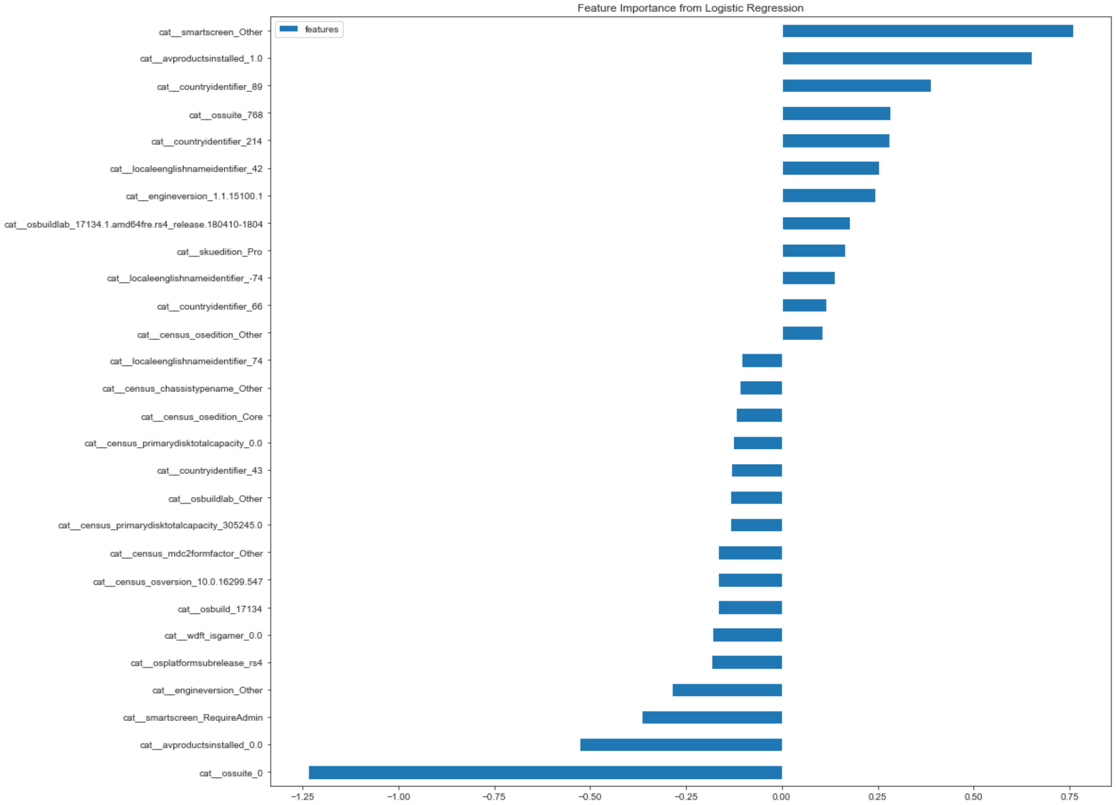
Model 1: Logistic Regression

<u>Class 1 (Malware Detected):</u>

- Smartscreen_other
- Avproductsinstalled_1.0
- Countryidentifier_89
- Countryidentifier_214
- Skuedition_Pro

<u>Class 0 (Malware not Detected):</u>

- Ossuite_0
- Avproductsinstalled_0
- Smartscreen_RequireAdmin
- Countryidentifier_43

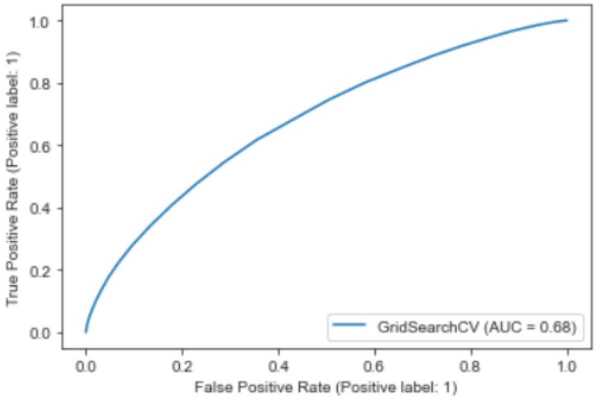


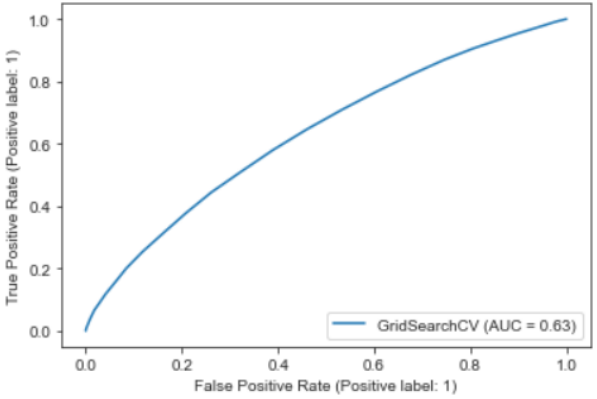
Model 2: K Neighbors Classifier

- Train AUC: 0.681
- Test AUC: 0.633
- Recall: 0.580

Confusion Matrix

	Predicted Negative	Predicted Positive
Actual Negative	7624	4864
Actual Positive	5253	7259



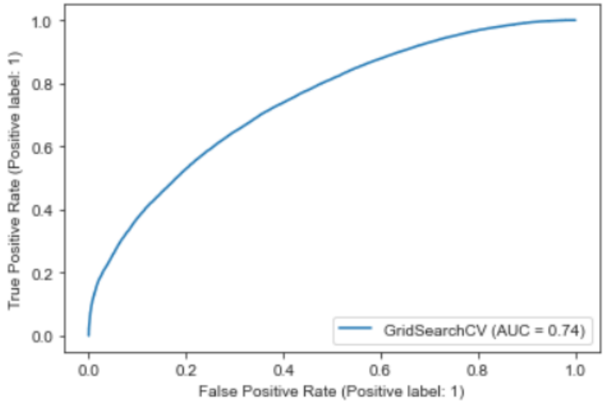


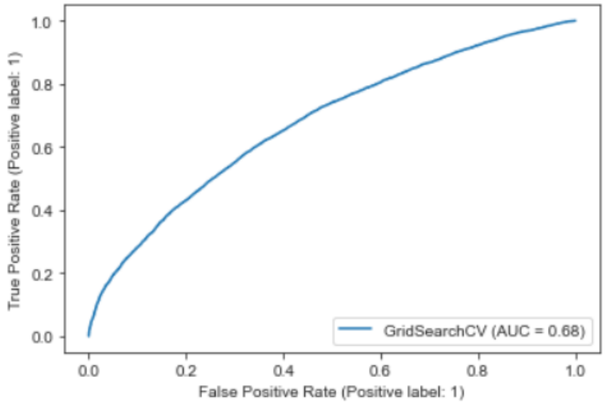
Model 3: Random Forest Classifier

- Train AUC: 0.745
- Test AUC: 0.678
- Recall: 0.581

Confusion Matrix

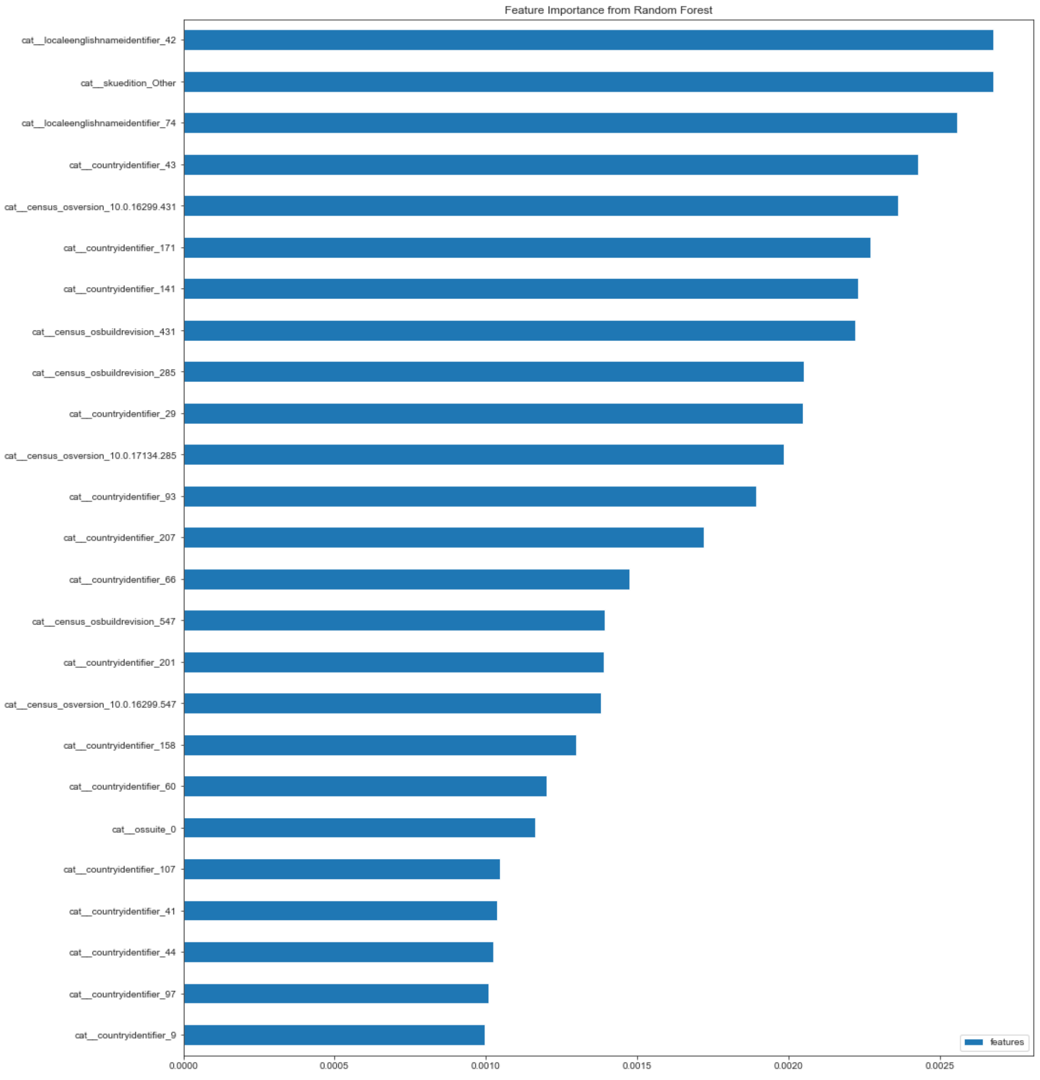
	Predicted Negative	Predicted Positive
Actual Negative	8439	4049
Actual Positive	5238	7274





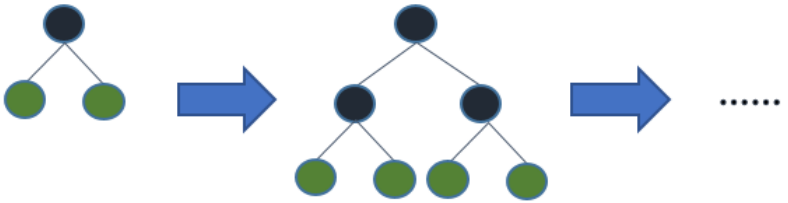
Model 3: Random Forest Classifier

Feature importance of Random Forest Classifier

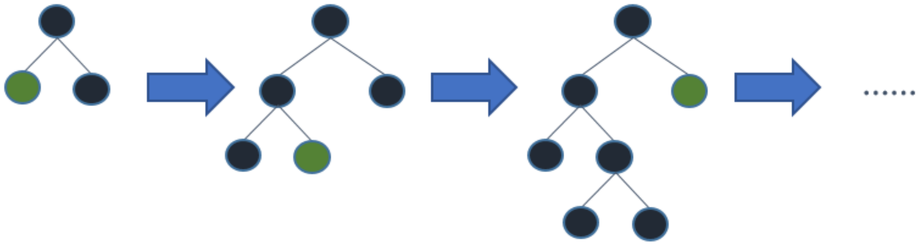


Model 4: Light GBM

- Introduced by Microsoft, Light Gradient Boosting or LightGBM is a highly efficient gradient boosting decision tree algorithm.
- The difference of Light GBM compared with other decision tree learning algorithm is that Light GBM will grow trees leaf-wise instead of level-wise.
- Light GBM will choose the leaf with the max delta loss to grow.
- Advantages:
  - Faster training speed and higher efficiency.
  - Lower memory usage.
  - Better accuracy.
  - Support of parallel, distributed, and GPU learning.
  - Capable of handling large-scale data.



Level-wise tree growth



Leaf-wise tree growth

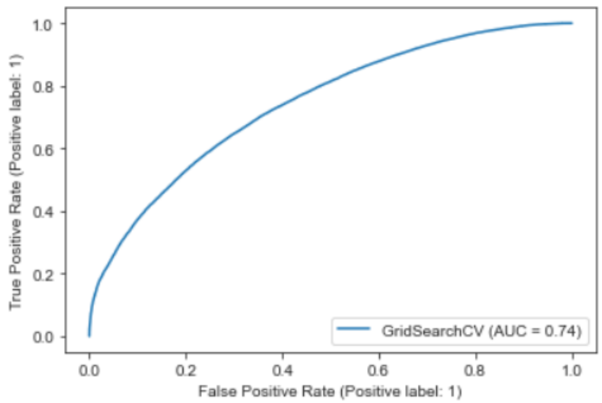
Light GBM Documentation

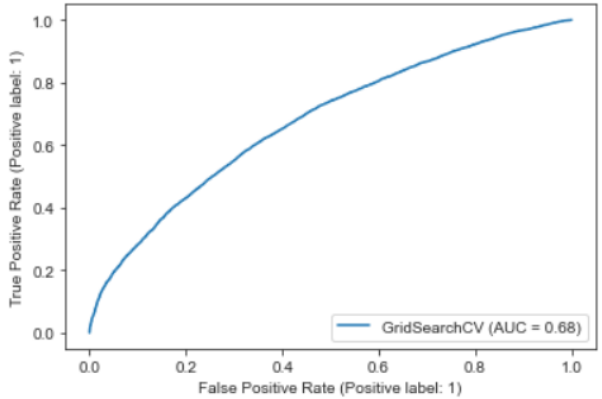
Model 4: Light GBM

- Train AUC: 0.706
- Test AUC: 0.682
- Recall: 0.599

Confusion Matrix

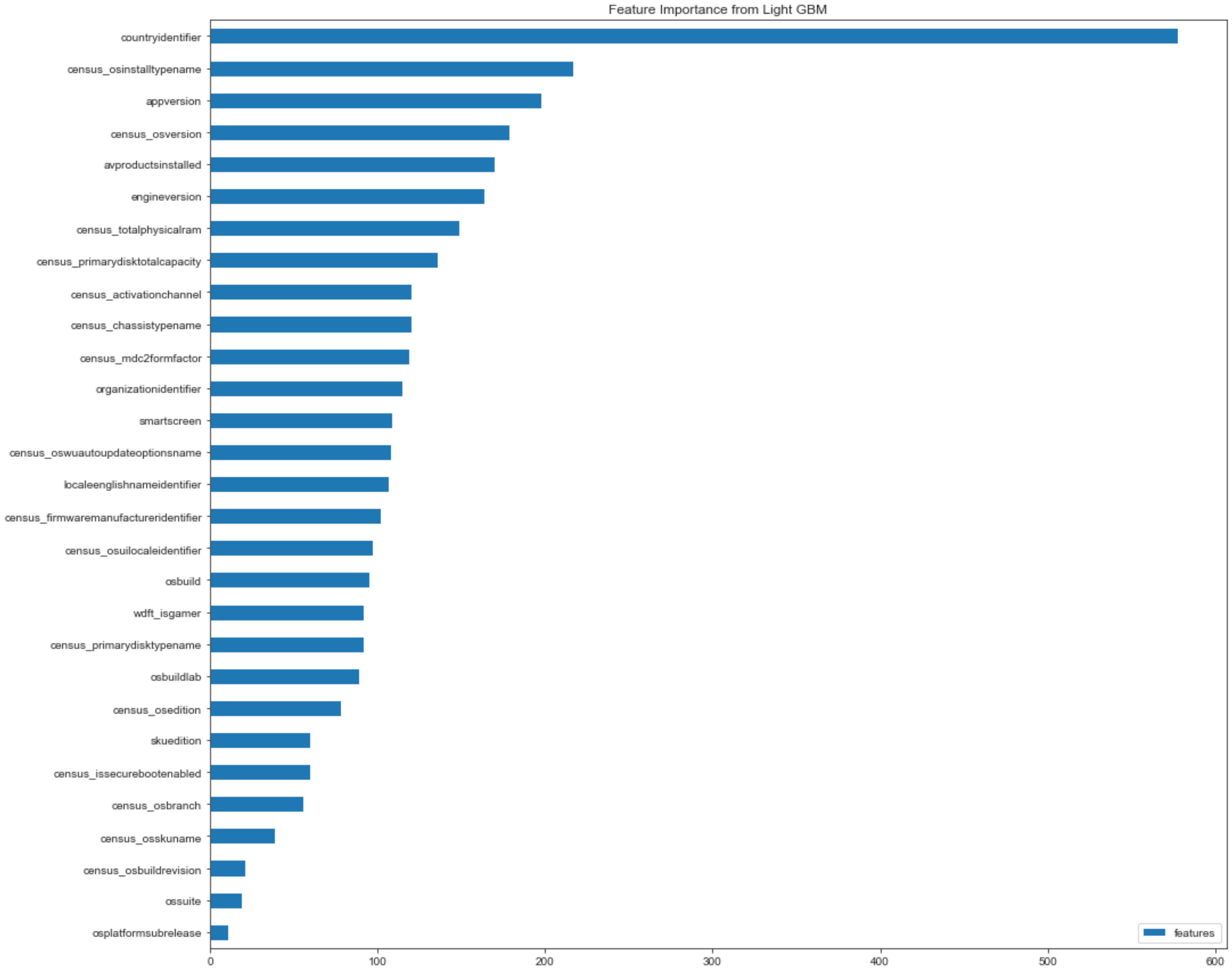
	Predicted Negative	Predicted Positive
Actual Negative	8280	4208
Actual Positive	5012	7500





Model 4: Light GBM

Feature importance of Light GBM

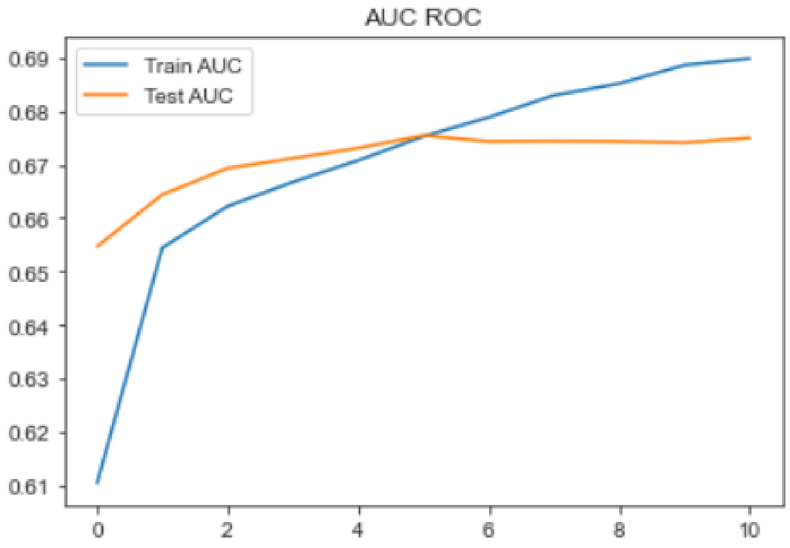


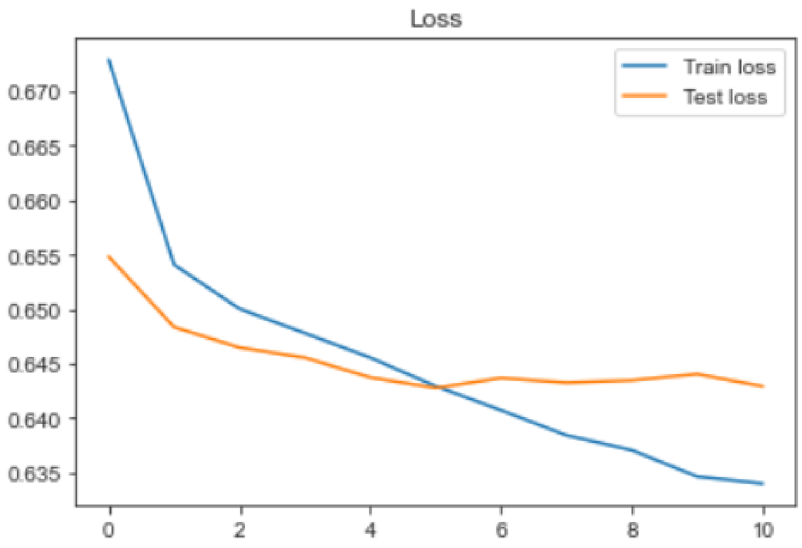
Model 5: Keras Sequential Neural Networks

- Train AUC: 0.689
- Test AUC: 0.675
- Recall: 0.605

```
Model: "sequential"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 dense (Dense)               (None, 82)                6806

 dense_1 (Dense)             (None, 40)                3320

 dropout (Dropout)           (None, 40)                0

 dense_2 (Dense)             (None, 20)                820

 dropout_1 (Dropout)         (None, 20)                0

 dense_3 (Dense)             (None, 1)                 21

=================================================================
Total params: 10,967
Trainable params: 10,967
Non-trainable params: 0
_________________________________________________________________
```

Model 5: Keras Sequential Neural Networks



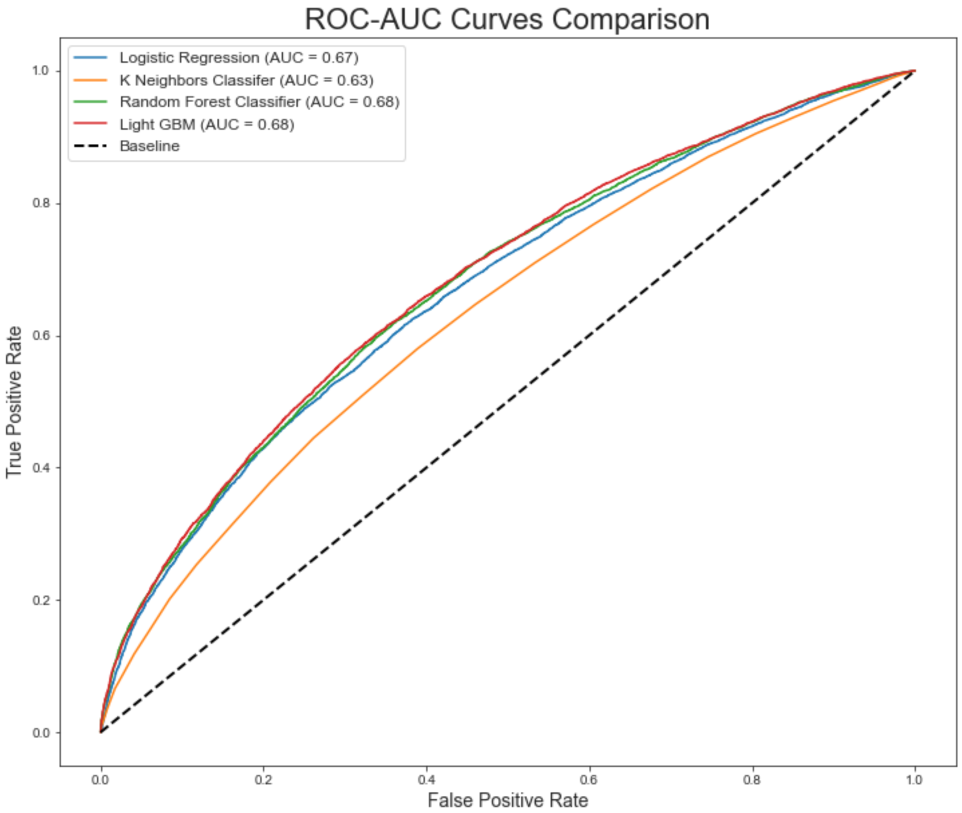


Model Evaluation

Models	Train AUC ROC	Test AUC ROC	Recall
Logistic Regression	0.665	0.668	0.593
K-Neighbors Classifier	0.681	0.633	0.580
Random Forest Classifier	0.745	0.678	0.581
Light GBM	0.706	0.682	0.599
Keras Sequential NN	0.689	0.675	0.605

- Baseline Model: 0.50
- Best Model would be the Light GBM based on the metrics of AUC ROC and Recall.
- Keras Sequential NN performed relatively well with a better Recall score and but a worst Test AUC score.

Model Evaluation



Kaggle Submission

Kaggle Leaderboard: Private Score – 0.676 Public Score – 0.714

Submission and Description	Private Score	Public Score	Use for Final Score
kaggle_df.csv a few seconds ago by jhkwek LGBM	0.54497	0.58448	☐
kaggle_df.csv 4 days ago by jhkwek RF	0.54014	0.57481	☐
kaggle_df.csv 5 days ago by jhkwek LR test	0.52149	0.56535	☐

06

Conclusion

Limitations, Future works
and Recommendations

Limitations

- The manner of imputation and dealing the null values and features with high cardinality could lead to information loss.
- Several of the features are masked at the source for confidentiality and that could make certain features hard to understand or interpret.
- The dataset seems to be obtained in 2018 and the values in the features is likely to be outdated as computer technology moves at a fast pace and the trained model is not likely to do well trying to predict malware in machines in 2022.
- The dataset only identifies if the machine is infected by malware but did not specify the type of malware that the machine is infected by. If more information on the type of malware that the machine.

Future Works

- Including extra time series information based on the time where the train and test dataset are scraped. The majority of train data are observations in August and September 2018 while test data is October and November 2018

- As the information in some of the features are masked for confidentiality like in features Country Identifier, Census_FirmwareManufacturerIdentifier and the codes used to represent the classes within the features are not interpretable. This would affect some of the methods of imputation or encoding that can be done for those features which can lead to some loss in information during modelling.

- The number of rows used for training is only 100,000 rows due to the limitations in time and hardware which could lead to some information loss when the entire train dataset is not used. This can lead to a slightly poorer performance of the models.

Recommendations

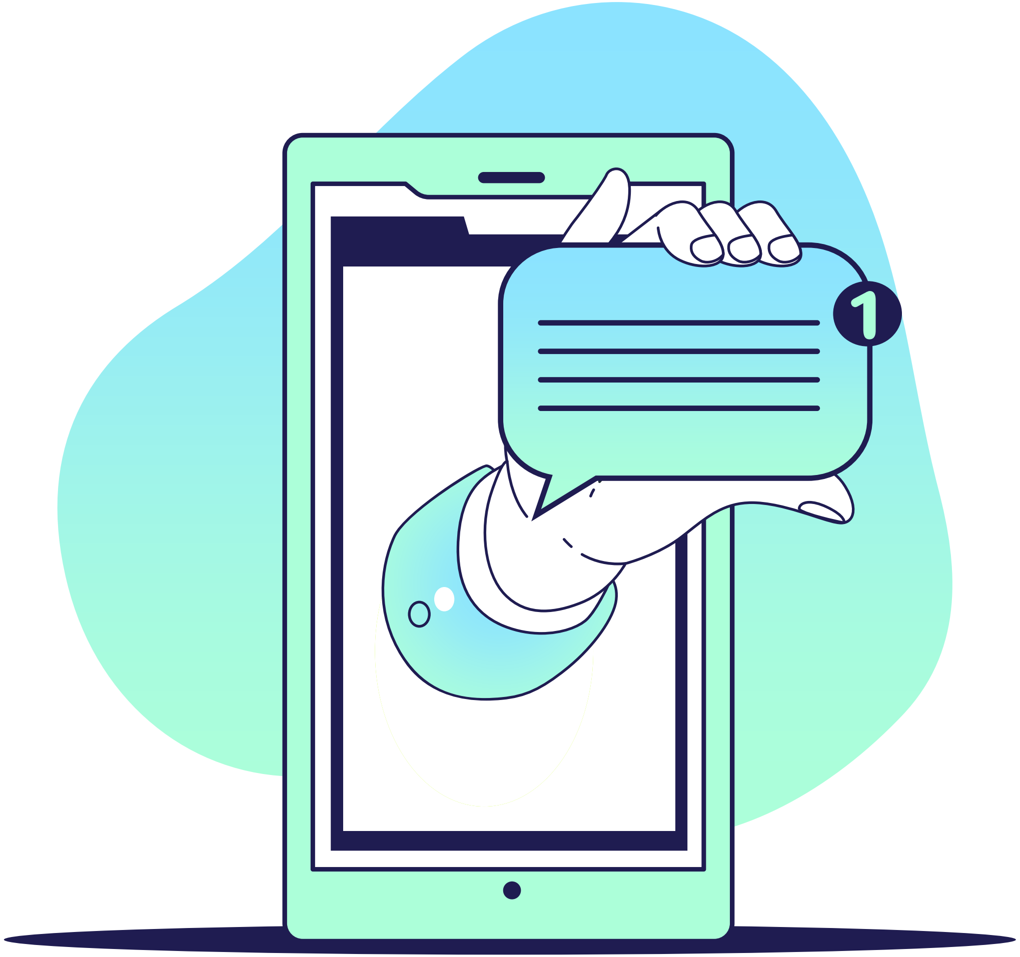
- Using the feature importance from the models tested, the company can focus additional efforts in devising stronger security protocols or solutions for industries, countries or certain OS build versions that are more vulnerable to malware attacks.
- The company can look to recommend users in regions and industries that could be more vulnerable to malware attacks to upgrade the operating system for more security features based on this model results.

The End

Thanks for your attention!

THANKS!

Do you have any questions?
youremail@freepik.com
+91 620 421 838
yourcompany.com



CREDITS: This presentation template was created by **Slidesgo**, including icons by **Flaticon**, infographics & images by **Freepik** and illustrations by Stories

Please keep this slide for attribution.