Malware Detection
Team AAP

Aarthi Mundla(IMT2018046) Aarthi.Sree@iiitb.org
Pavan Sudeesh(IMT2018517) Pavan.Sudeesh@iiitb.org
Abhigna Banda(IMT2018002) abhigna.banda@iiitb.org

1. INTRODUCTION

Malware, short for malicious software, is a blanket term for viruses, worms, trojans and other harmful computer programs hackers use to wreak destruction and gain access to sensitive information.

The effects of malware are slowing our computer, leaking sensitive information. It is important to identify the malware and remove it at an early stage. Malware detection is the process of scanning the computer and files to detect malware. Some standard malware detections are:

1.1. Signature-Based Detection

Each of the malware has a unique code. When the system receives a file, malware scanner sends this code to cloud-storage db and checks if it matches in the list of virus codes. If it matches, it is a malware.

1.2. Heuristic Analysis:

It works on rules. Each file has some restrictions(like direct access to hard drive not allowed). If any file has crossed these limits, it comes under a suspicious file.

1.3. Sandbox:

It is a cell within the computer in which the file is analysed and released only if the file is legit.
When the threat is identified, it is deleted from the computer.

2. DATASET GIVEN TO US

The dataset given to us is Malware Detection Santander Teams Kaggle competition dataset. It has 83 columns and more than 5,00,000 rows. Our column of interest is the 'HasDetections' column. It predicts whether a system with certain properties can detect a malare. Our goal is to predict the probability predicted for the HasDetections column. This column 'HasDetections ' is not perfectly balanced and has unequal ones and zeros. (Where one means the system has a malware and zero means the system does not have a malware.)

It is a classification problem. Most ml classification algorithms are sensitive to unbalance in the predictor classes. An unbalanced dataset will bias the prediction model towards the more common class so it is important to balance the data.

3. DATA PREPROCESSING

The cause of missing values can be data corruption or failure to record data. The handling of missing data is very important during the preprocessing of the dataset as many machine learning algorithms do not support missing values.

3.1. Removal of unnecessary columns

There were some columns in which null values are like 3,00,000 which means more than fifty per cent of data were null so we removed those columns. A model trained with the removal of all missing values creates a robust model.

3.2. Filling of Null values

The columns have 2 types of data
- Categorical type
- Numerical type
Categorical type : For categorical type data, we replaced them with a new category termed as "UNKNOWN"
Numerical type : Numerical type is either int or float. We have replaced it with the mode of their respective column. This method can prevent the loss of data also.

3.3. Encoding

The performance of a machine learning model not only depends on the model and the hyperparameters but also on how we process and feed different types of variables to the model. Since most machine learning models only accept numeric variables, preprocessing the categorical variables becomes a necessary step. We need to convert these categorical variables to numbers such that the model is able to understand and extract valuable information. The data we are working on do not have any inherent order. Hence we used one-hot encoding. In one hot encoding, for each level of a categorical feature, we create a new

variable. Each category is mapped with a binary variable containing either 0 or 1. Here, 0 represents the absence, and 1 represents the presence of that category.

We did one-hot-encoding for all categorical data type columns by using one-hot encoder(a predefined function).

4. METRICS

There are various metrics which we can use to evaluate the performance of ML algorithms, classification algorithms. The performance of ML algorithms measured and compared will be dependent entirely on the metric you choose.
Some of the performance metrics that can be used to evaluate predictions for classification problems are :

Confusion Matrix : It gives us the "True Positives (TP)", "True Negatives (TN)", "False Positives (FP)", "False Negatives (FN)" values with the help of which we can calculate Accuracy, Precision, Recall,Specificity.

F1 : This score will give us the harmonic mean of precision and recall.
F1 = 2*(Precession*Recall)/ (Precession+ Recall)

AUC ROC: AUC (Area Under Curve)-ROC (Receiver Operating Characteristic) is a performance metric, based on varying threshold values, for classification problems. AUC-ROC metric will tell us about the capability of the model in distinguishing the classes. Higher the AUC, better the model.
As the metric used for scoring is AUROC instead of submitting all our predictions we splitted our train data into 2 parts using train-test-split and trained our model on 80 percent of train data and tested it on the remaining 20 percent data.

5. MODELLING

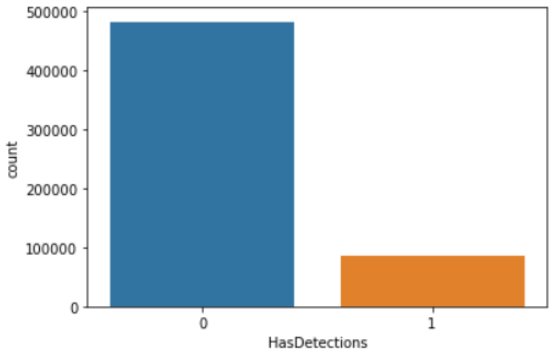
The process of modeling means training a machine learning algorithm to predict the labels from the features, tuning it for the business need, and validating it on holdout data. The output from modeling is a trained model that can be used for inference, making predictions on new data points.
As our question is a classification problem so we used classification models for solving that. Models we used were
1.)Logistic regression
2.)Naive Bayes
3.)Random forest
4.)Xgboost
5.)Light gbm

Evaluation of different types of models

	accuracy	**recall**	**precision**	**f1-score**
Logistic regression	0.8499991192996671	5.871301080319399e-05	0.5	0.0001174122343548197 8
Naive Bayes	0.8080513624434 15	0.2118365429779239	0.3011937557392103	0.2487332391161972 7
Xgboost	0.8580575273457454	0.1110263034288 3983	0.6595744680851063	0.1900598020001 0053
Light gbm	0.8580575273457454	0.1110263034288 3983	0.6595744680851063	0.1900598020001 0053

6. BALANCING DATA

Our data have unequal ones and zeros in the has-detections column and our graph looks like ..

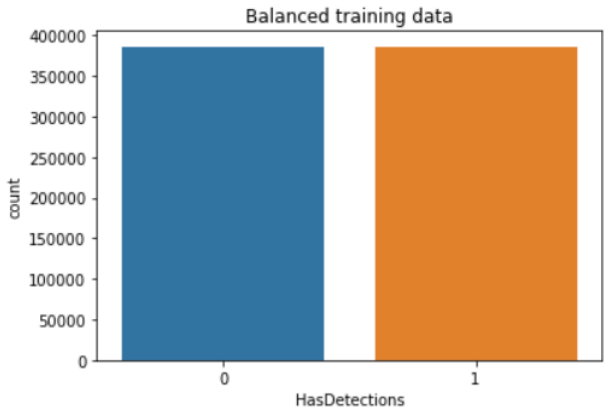


It has approximately 15 percent 1's and 85 percent 0's

Machine Learning algorithms tend to produce unsatisfactory classifiers when faced with imbalanced datasets. Hence we need to balance data. We can balance the data either by increasing the frequency of the minority class(over sampling) or decreasing the frequency of the majority class(under sampling). We have chosen to increase the frequency of the minority class. This method leads to no information loss.

As the number of 0's are less than the number of 1's so we decided to oversample the data in order to make the count of 0's and 1's equal. There are many techniques to over sample but we used 'Smote' (Synthetic Minority Over-sampling Technique) to over sample the data. It is a technique in which synthetic examples of minority class are produced with the help of k nearest neighbours (k-NN) algorithm.

After balancing the data, the graph looks like



MODELLING AFTER BALANCING DATA

Model	Leaderboard Score
Logistic regression	0.502
Naive bayes	0.602
Light gbm	0.706

We got a better leaderboard score with light gbm.
So we have started varying hyperparameters in light gbm. We tried with many variations and the respective Auroc scores are mentioned below for the light gbm model.
- With no parameters- 0.7086
- With Num_leaves, max_depth changing

Num_leaves	max_depth	Auroc score
30	100	0.7082
60	100	0.7123
60	150	0.7123
80	100	0.7140
100	100	0.7141

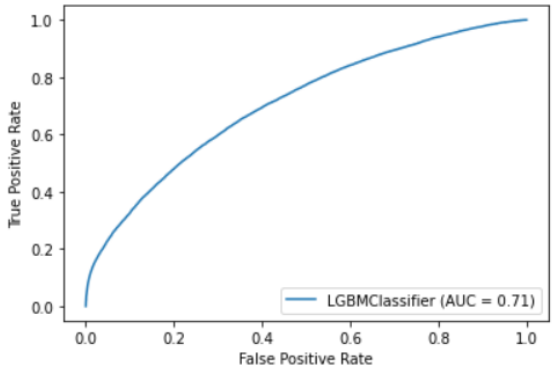
->With num_leaves =100, max_depth = 100 being fixed and varying learning rate

Learning rate	Auroc score
0.1	0.7141
0.2	0.710
0.05	0.707
0.5	0.695
0.09	0.7145
0.08	0.7141
0.095	0.7146
0.096	0.7143

->With num_leaves =100, max_depth = 100, learning_rate = 0.095 fixed and n_estimators varying

N_estimaters	Auroc score
100	0.7146
200	0.713
300	0.712
1000	0.7049
110	0.71453
90	0.7143
50	0.706
105	0.71455

We got our maximum auroc score when max_depth = 100 and num_leaves =100.
We have submitted them and got the best leader board score as 0.71138.



This shows our model is correct upto what extent(LGBM).

7. CHALLENGES

We also tried with models like random forest after balancing the data but it was taking a lot of time to give results hence we could not work further on it.

Things that we could have done to improve accuracy.
We could have combined various models. We could have also tried Stacking and Blending. We have randomly given values to hyper-parameters instead we could have used Grid-search techniques.

8. CONCLUSION

In this assignment we learnt about effects of imbalance data, how do we balance them. We have also learnt about model evaluation metrics like f1 score and auc-roc scores. We learnt about tuning hyperparameters and Grid Search technique useful for finding appropriate hyper-parameters.

9. FUTURE SCOPE

Various permutations and combinations can be applied on the feature extraction mechanisms, hence comparing the accuracy of malware detection with previous results. With the invasion of unaccustomed and contradistinctive cyber-attacks on a daily basis, the field of malware analysis has a lot of scope for improvement.

10. ACKNOWLEDGEMENTS

Would like to thank Professor G. Srinivas Raghavan and our Machine Learning Teaching Assistant Tejas Kotha for giving us guidance and support throughout the assignment. We would also like to thank other TA's who helped us in understanding the concepts. It was a great learning experience for us.

11. REFERENCES

1.)https://www.sciencedirect.com/topics/
computer-science/malware-detection
2.)https://enterprise.comodo.com/
what-is-malware-detection.php
3.)https://scikit-learn.org/stable/modules/generated/sklearn.
preprocessing.OneHotEncoder.html
4.)https://lightgbm.readthedocs.io/en/latest/pythonapi/
lightgbm.LGBMClassifier.html
5.)https://scikit-learn.org/stable/modules/generated/sklearn.
preprocessing.OneHotEncoder.html
6.)https://www.analyticsvidhya.com/blog/2020/08/
types-of-categorical-data-encoding/
7.)https://towardsdatascience.com/
7-ways-to-handle-missing-values-in-machine-learning-1a6326adf79e
8.)https://towardsdatascience.com/
handling-imbalanced-datasets-in-machine-learning-7a0e84220f28
9.)https://machinelearningmastery.com/
tactics-to-combat-imbalanced-classes-in-your-machine-learning-dataset/