Capstone Project

Android App Authenticity Prediction
Supervised Classification Project

By - Dishant Toraskar

CONTENT

1. Problem Definition
2. Data Description
3. Exploratory Data Analysis
4. Feature Engineering
5. Evaluation Metrics
6. Model Building & Evaluation-
   i. Baseline Model
   ii. Random Forest Classifier
   iii. Gradient Boosting Classifier
   iv. XGBoost Classifier
7. AUC - ROC Curve
8. Model Explainability
9. Future Scope
10. Conclusion

Problem Definition

Android malware are malicious program that can damage the whole system, corrupt the data or even leak the sensitive information of the user. To prevent this we have to notify the user about the app before user downloads and use it.

To solve the above problem, we can build a machine learning model to classify an app as benign or maliciou/malware. This is a type of binary classification problem. With the help of many different classification algorithms we can solve this problem based on the features we have. Before training we must collect the data, get to know information and insights from data, and also do data preprocessing as and where needed.

Data Description

We are provided was a dataset consisting of 184 features and 30,000 rows.

There are 5 features of ‘object’ data type, 3 are of ‘float’ type and rest are ‘int’ type.

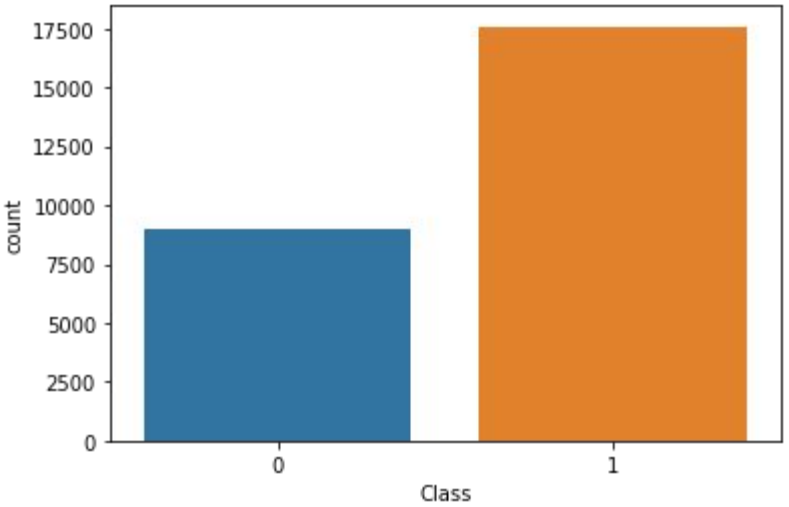
Target feature is the ‘Class’ column and it is a binary categorical feature.

Most of the numerical features are binary in nature consisting of 0 and 1 which are mostly related to permissions needed by the app while downloading.

Discrete numerical features include ‘Price’, ‘Rating’, ‘Number of ratings’, ‘Safe permissions count’ & ‘Dangerous permissions count’.

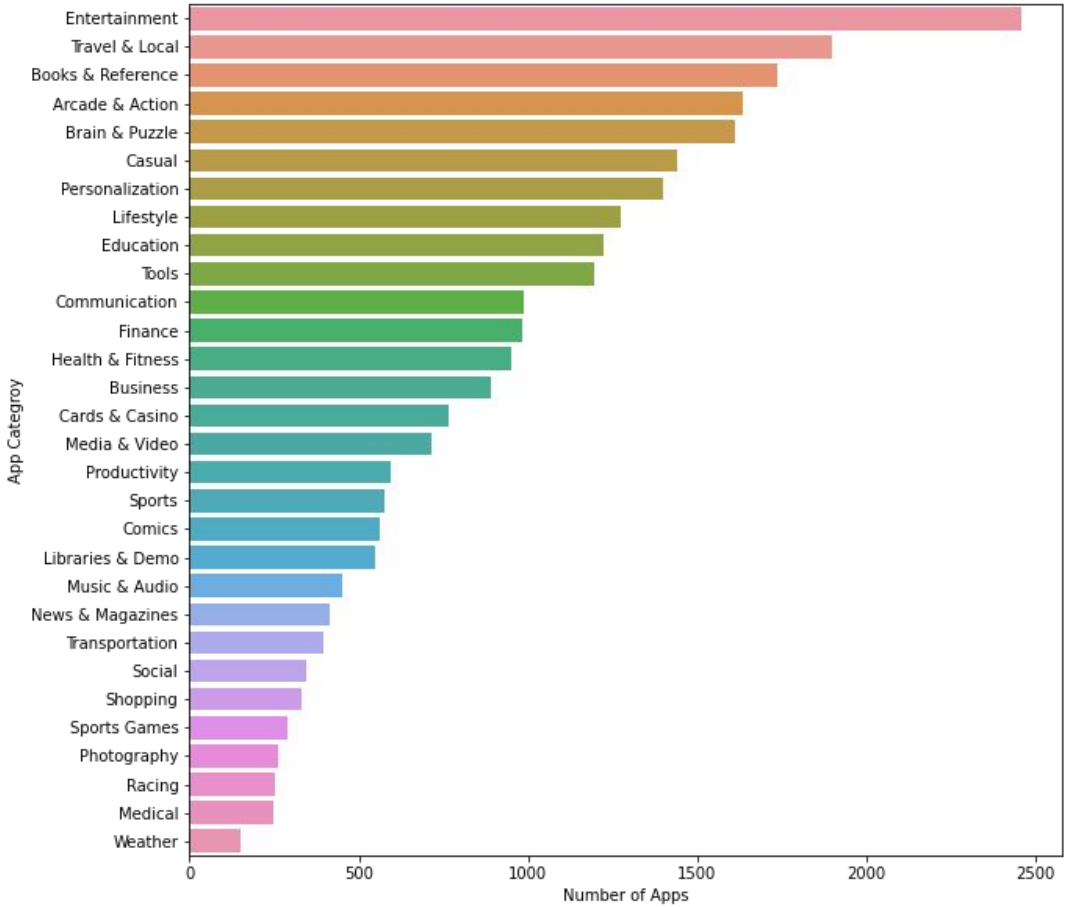
Exploratory Data Analysis

“Target Class”



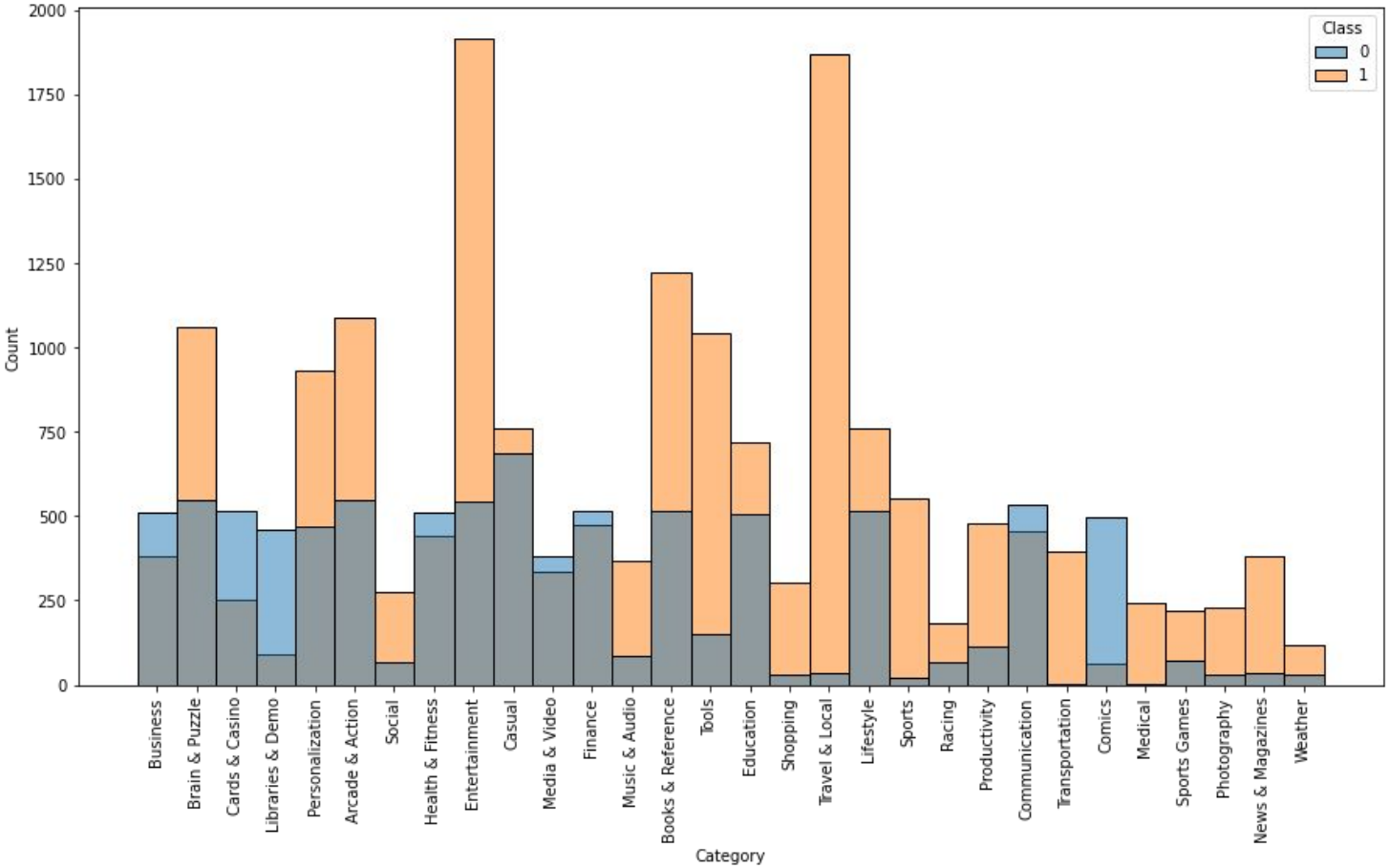
We can observe that most number of app belongs to malware(1) class, whilst approximately half of that belongs to benign(0) class.

“Category”

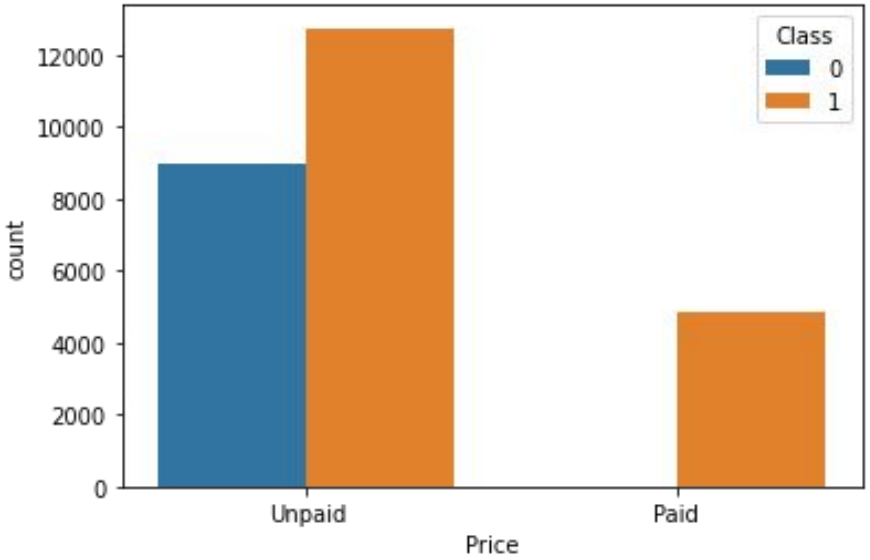


Most number of app belongs to Entertainment category followed by Travel, Books & Arcade. While least belongs to Weather category followed by Medical, Racing and Photography.

“Category w.r.t. Target Class”

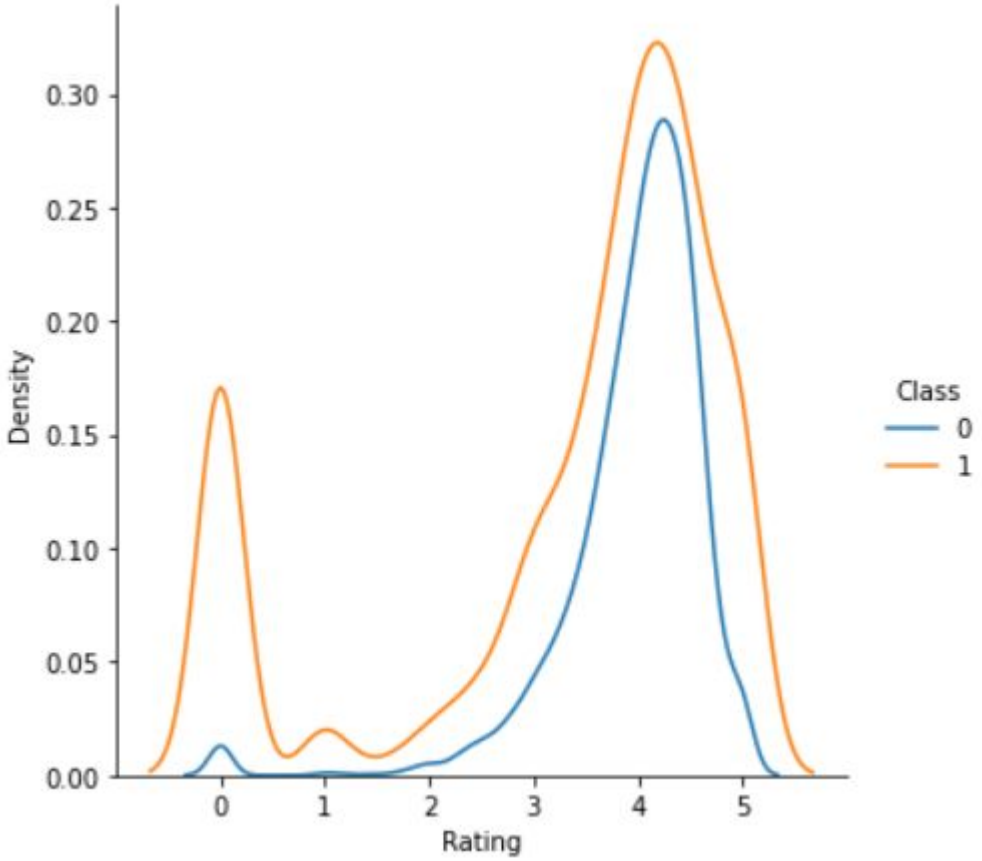


“Price (Paid/Unpaid) w.r.t. Target Class”



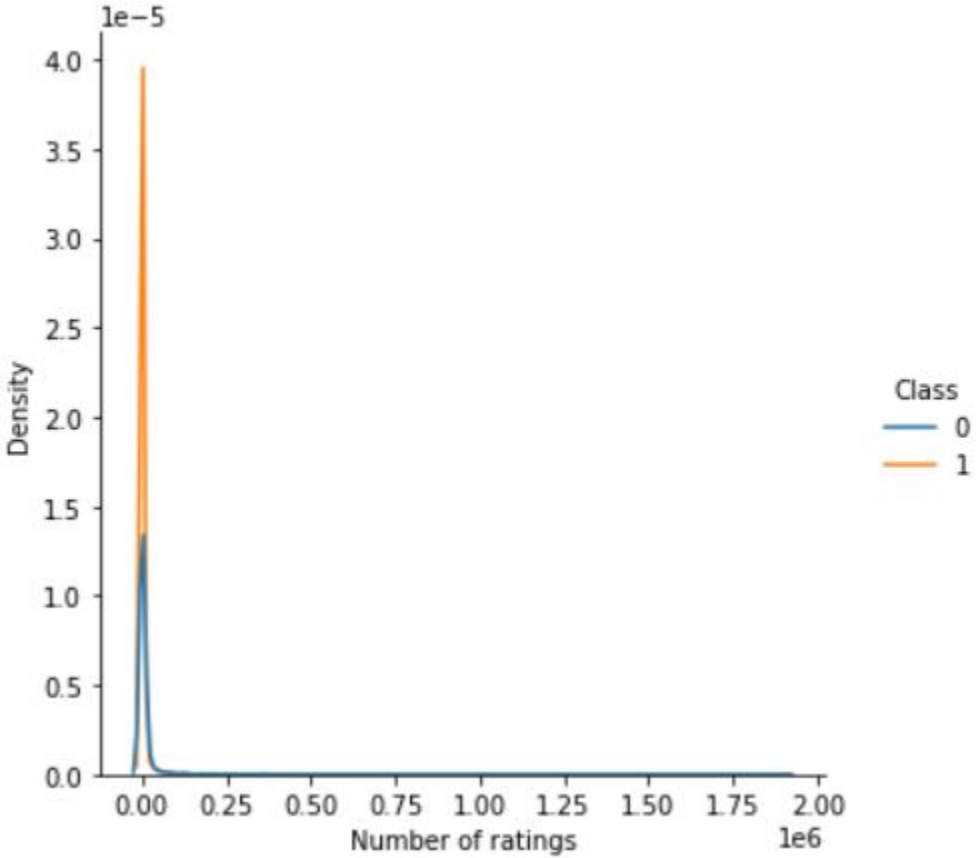
We can observe that, the apps which were paid are 100% malicious. There’s not even a single benign app which is paid.

“Ratings w.r.t. Target Class”



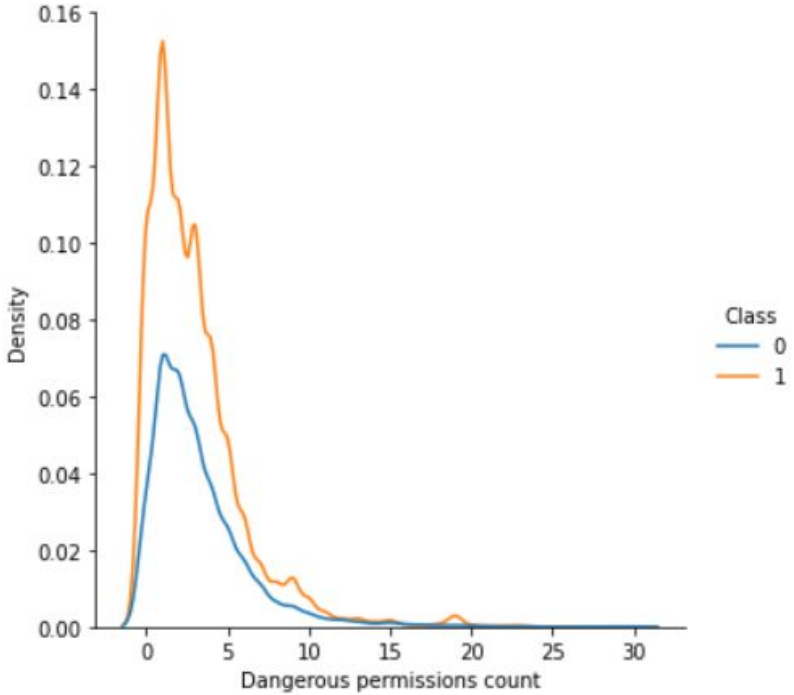
We can observe that when the rating was 0 or 1, then the proportion of apps belonging to the benign class were very less as compared to when the rating was 2 or greater than 2.

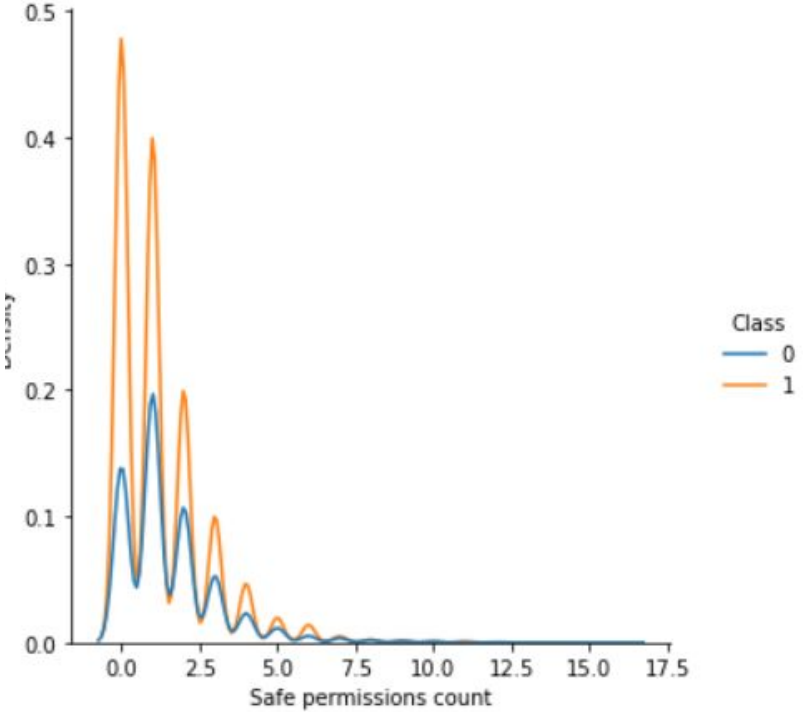
“Number of ratings w.r.t. Target Class”



We can observe that, there are many apps with ratings of 100,000 to 110,000. Thus, number proportion of apps belonging to maware class is also higher over there.

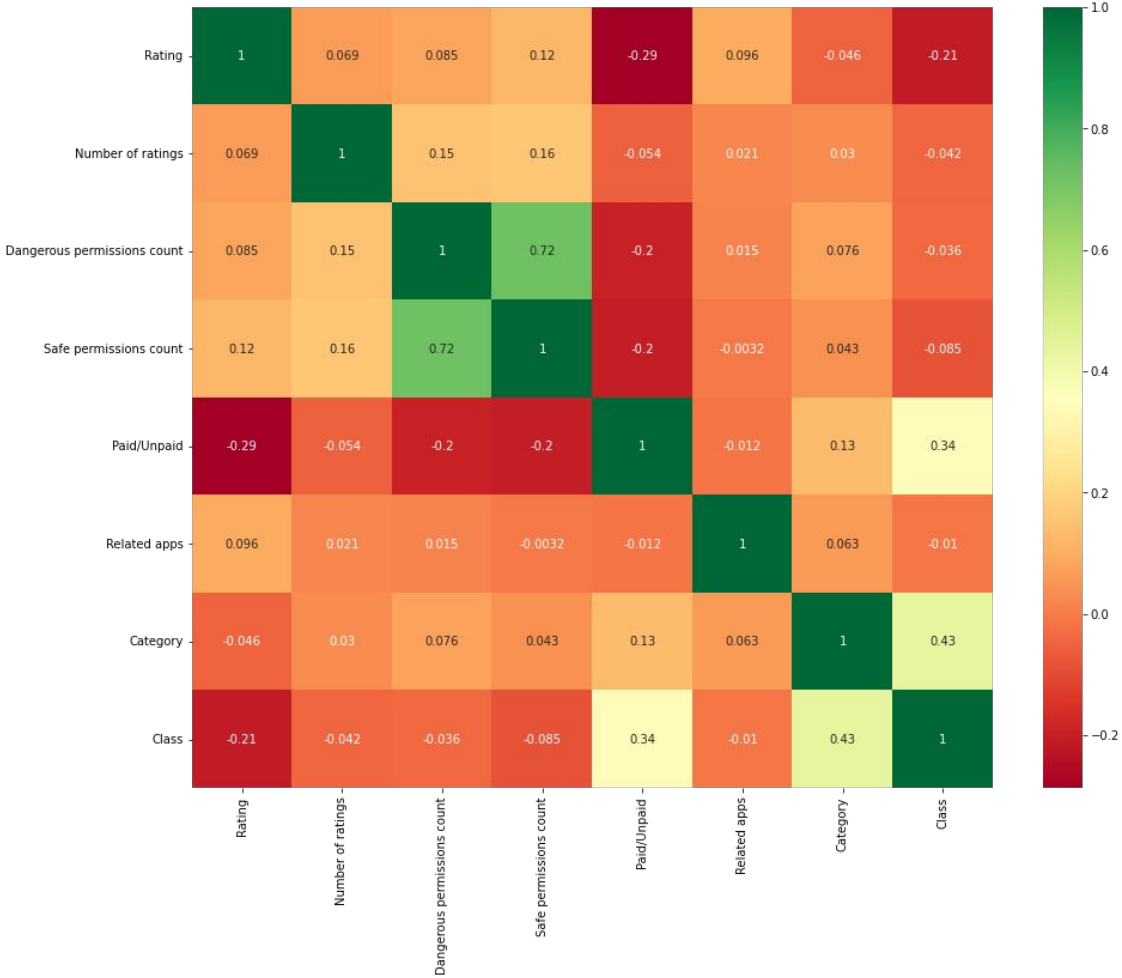
“Permissions count w.r.t. Target Class”





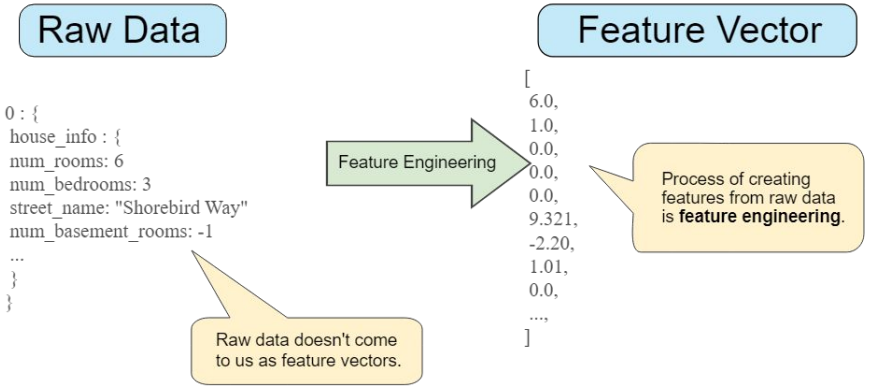
We can observe that ‘Dangerous permissions count’ & ‘Safe permissions count’ are correlated to each other.

“Data Correlation”



We can observe that there’s a strong positive correlation between Safe permissions count & Dangerous permissions count. There’s moderately strong relationship between Paid/Unpaid, Category & Class.

Feature Engineering



Feature engineering is the process of transforming raw data into useful data for machine learning process. It converts & extract the useful numerical information from the patterns of data to give it as a input to machine learning model. We have used feature engineering for converting categorical feature (“Category”) into numerical feature with the help of mean encoding.

Mean encoding represents a probability of target variable, condition on each unique value or the feature.

$$label_{mean} = \frac{\sum label\ with\ target\ 1}{total\ labels}$$

Other than mean encoding, I have manually converted “Price” column into “Paid/Unpaid” column with 0 as unpaid & 1 as paid.

Evaluation Metrics

Accuracy, Recall, Precision, F1-score and AUC - ROC are some of the evaluation metrics for classification algorithms.

Accuracy : Explains how often the classifier correctly predicts.

$$\textbf{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN}$$

Recall : Recall explains how many actual positive classes our model was able to predict out of total actual positive classes.

$$Recall = \frac{TruePositive}{TruePositive + FalseNegative}$$

Precision : Precision explains out of total positive predicted class, how many were actually positive.

$$Precision = \frac{TruePositive}{TruePositive + FalsePositive}$$

F1-score : It gives the combined idea about precision and recall. It is the harmonic mean of precision and recall.

$$F1 = 2.\frac{Precision \times Recall}{Precision + Recall}$$

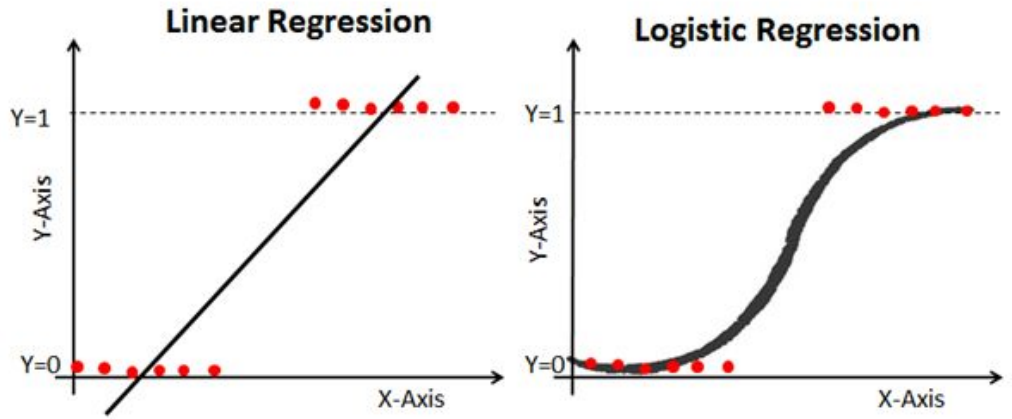
AUC - ROC : ROC is the probability curve and AUC represents, the degree or measure of separability. Higher the AUC, better the model is in classifying class 0 as 0 and class 1 as 1.

Model Building & Evaluation

Baseline Model

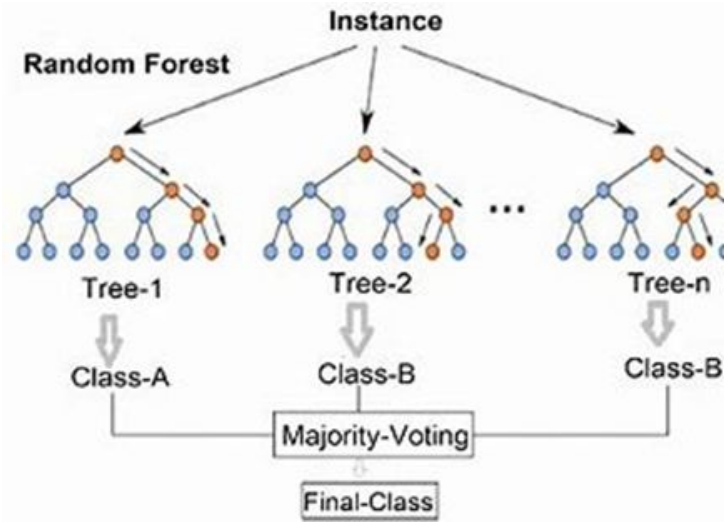
Baseline models are the most basic types of machine learning model. The model is trained without any feature engineering or even we can use random guessing for classification of class to build a baseline model. We have used logistic regression without any feature engineering or hyper parameter tuning to get a baseline model.

	Logistic Regression
test_accuracy	0.660963
recall_test	0.997158
precision_test	0.661826
f1_test	0.795601
auc_test	0.500247
cm_test	[[6, 1793], [10, 3509]]
train_accuracy	0.661698
precision_train	0.662294
recall_train	0.997229
f1_train	0.795962
auc_train	0.501325
cm_train	[[39, 7156], [39, 14034]]



Random Forest Classifier

Random forest classifier creates a set of decision trees from a random selected subset of training set and then it collects the votes from different decision trees to give us the final output.

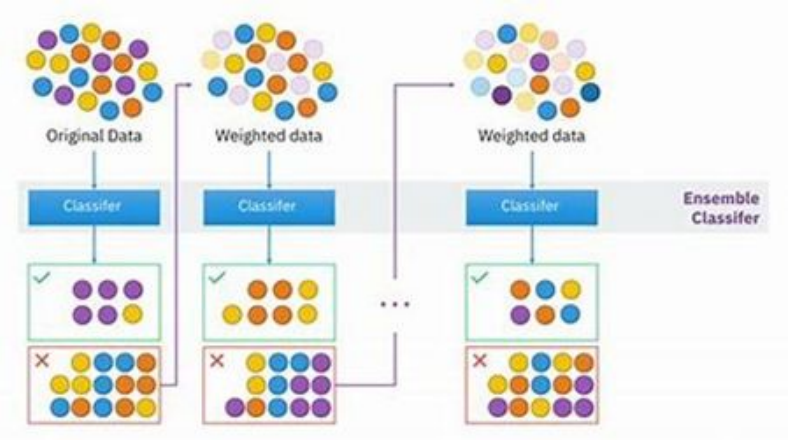


	precision	recall	f1-score	support
0	0.87	0.02	0.03	1799
1	0.66	1.00	0.80	3519
accuracy			0.67	5318
macro avg	0.77	0.51	0.41	5318
weighted avg	0.73	0.67	0.54	5318

We can observe that Random Forest Classifier is not performing good while predicting benign apps as it can only classify 3% of app that were benign.

Gradient Boosting Classifier

Gradient boosting classifier is a set of machine learning algorithms that include several weak learners and combine them into a strong big one with highly predictive output. In gradient boosting, the errors of previous model act as a weight for next model and so on.

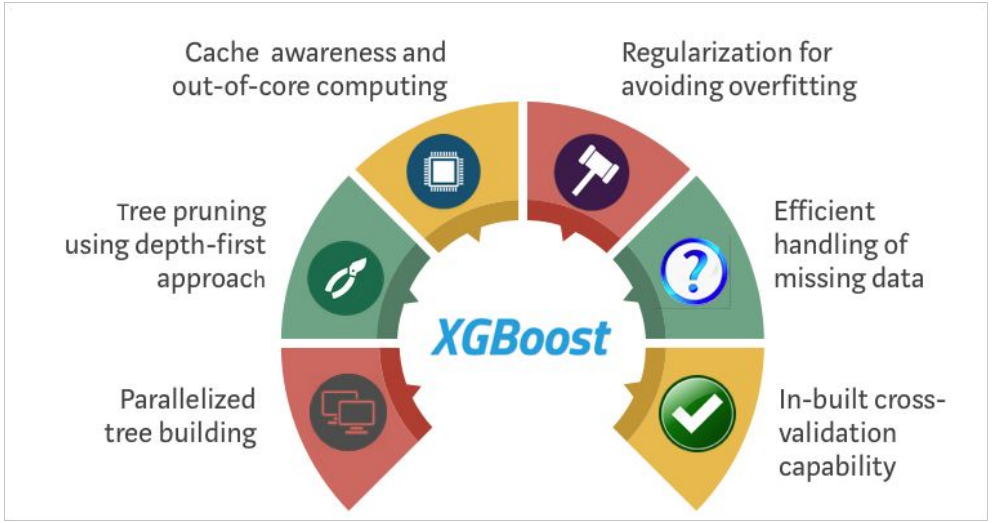


	precision	recall	f1-score	support
0	0.70	0.74	0.72	1799
1	0.86	0.84	0.85	3519
accuracy			0.81	5318
macro avg	0.78	0.79	0.79	5318
weighted avg	0.81	0.81	0.81	5318

We can observe that Gradient Boosting was able to classify 72% of the benign apps and 85% of the malware apps correctly with an accuracy of 81%.

XGB Classifier

XGBoost is an implementation of gradient boosting algorithm. It is a type of library that is basically designed to improve model performance and speed. It works similarly to gradient boosting classifier.

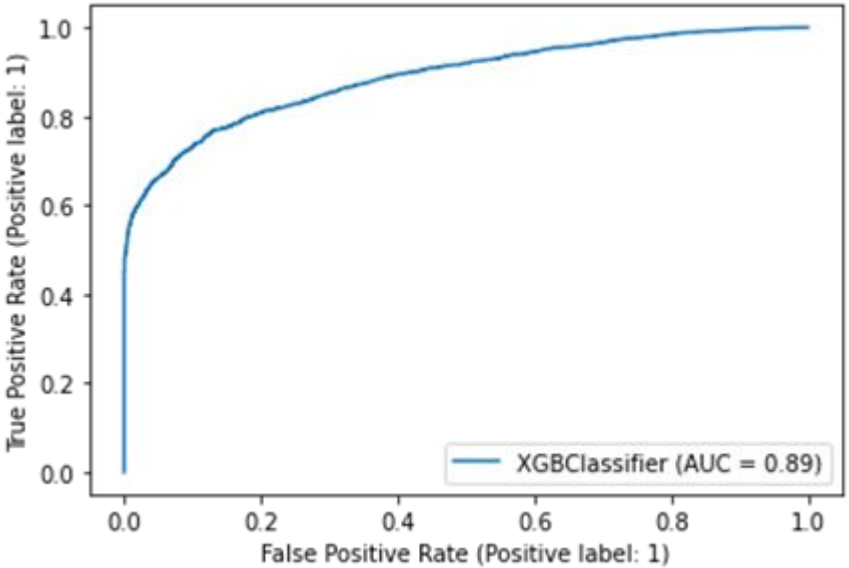


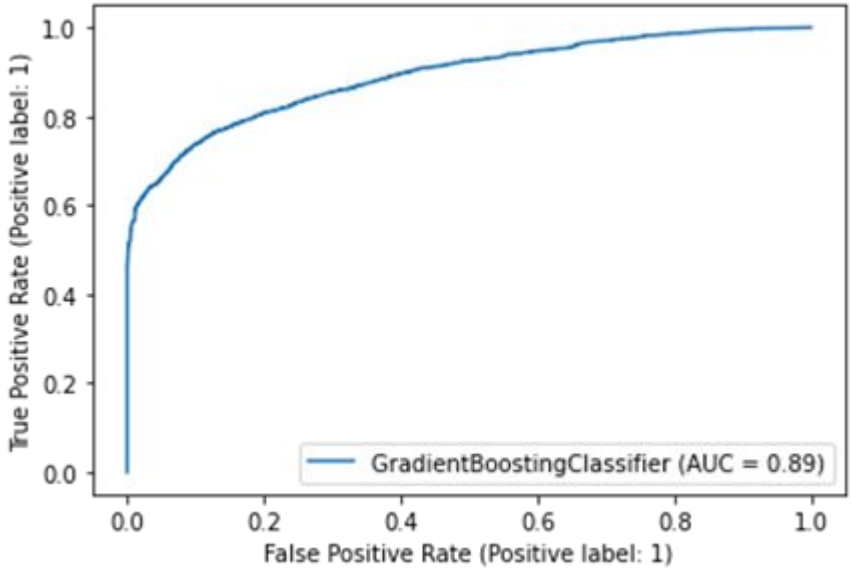
	precision	recall	f1-score	support
0	0.69	0.73	0.71	1799
1	0.86	0.84	0.85	3519
accuracy			0.80	5318
macro avg	0.78	0.78	0.78	5318
weighted avg	0.80	0.80	0.80	5318

We can observe that XGBoost is giving same performance as Gradient Boosting with 71% app correctly classified as benign and 85% for malware.

AUC - ROC Curve

AUC – ROC curve is one of the performance metrics for classification problems in machine learning. ROC is the probability curve and AUC represents the degree or measure of separability. Higher the AUC, better the model is in classifying class 0 as 0 and class 1 as 1.





The above 2 figures are AUC - ROC curve for Gradient Boosting & XGB Classifier. We can observe that both have AUC of 89%. We can choose any of the ML model based upon our business needs and based on computational complexity.

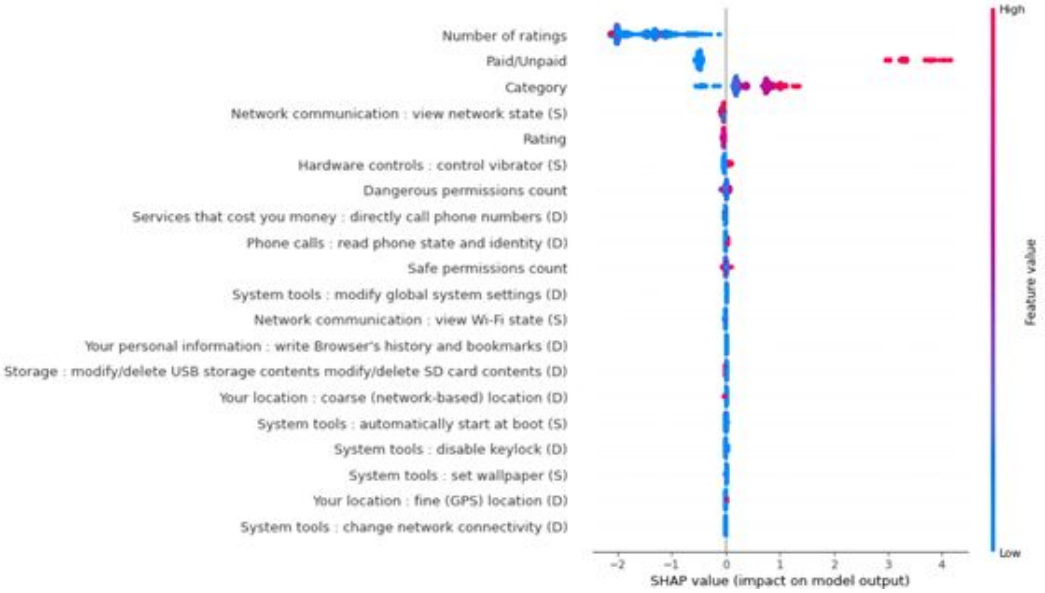
Model Explainability

Model explainability in machine learning means that you can explain what happens in your model from input to output.

Example –

We know that our model is classifying an app as benign or malware. This is the answer to what? But do we know why? Why our model classified some app as benign and some as malicious. Model explainability solves this problem of black box. It tells us the features that were responsible for classification of class and also by how much percent it was responsible.

For this model we have used SHAP (Shapley Additive exPlanations) for model explainability. It is used to calculate the impact of each part or features of the model to the final result.



Future Scope

1. We can use NLP techniques such as TF-IDF and Word Cloud on the text columns.
2. We can eliminate the outliers present in some of our numerical features or can handle them by value imputation.
3. Handle the missing values in a dataset more efficiently instead of just removing them.
4. Can resample our data preferably over-sampling.
5. Improve overall accuracy of the model with the help of NLP.
6. Deploy the model.
7. Research more about the domain to get more understanding about the domain knowledge.

Conclusion

1. Started with EDA.
2. Cleaned the dataset, did feature engineering and handled outliers for Price feature
3. Performed Mean Encoding for Category feature.
4. Build a baseline model using Logistic Regression.
5. Trained various others classification algorithm.
6. Found out that Gradient Boosting Classifier and Xtreme Gradient Boosting Classifier performed best.
7. Got an overall accuracy of 80%. With overall F1-score as 72% for app being benign and 85% for app being Malware.
8. Performed feature importance and found out that Number of ratings, price and category were the most important features in classifying the app as benign or malware.

THANK YOU