# Capstone Project – 3

## Android Authenticity Prediction


**Team**

Arunav Goswami

Nayanjyoti Sharma

Mohammed Saad Pasha

# Problem Statement

- This dataset consists of apps needed permissions during installation and run-time. We collect apps from three different sources google play, third-party apps and malware dataset. This file contains more than 30,000 Android apps. features extracted at the time of installation and execution. One file contains the name of the features and others contain .apk file corresponding to it extracted permissions with respective package. Apps are collected from Google's play store, hiapk, app china, Android, mumayi , gfan slideme, and pandaapp. These .apk files are collected from the last three years continuously and contain 81 distinct malware families. But, Here you are only supposed to predict whether the app is benign(0) or malware(1).

# Content

- Data Pipeline
- Data Description
- Exploratory Data Analysis
- Feature Selection
- Machine Learning Algorithms
- Model Validation and Selection
- Evaluation Matrix of all the Models
- Model Explainability – LIME, ELI5
- Challenges
- Conclusion

# Data Pipeline

- **Data Processing :** Checking for Missing values and Duplicate values.
- **EDA & Feature Engineering: -** Analyzing each feature individually, creation of new features according to our need, dropping of features by checking correlation and VIF, handling of outliers, standardization and normalization of features.
- **Model Creation and Validation :** Fitting of Machine Learning models into training and testing dataset, evaluation of performance metrics and Hyperparameter Tuning.
- **Model Explainability – LIME, ELI5**

# Data Summary

**Dependent variable :**

- Class :- Whether the app is Benign(0) or Malware(1) :-

**Independent variables :**

- App :- Name of the App
- Package :- OBB/Data package installed in root folder
- Category :- App Category (eg. Entertainment, Adventure, puzzle, Action, Antivirus, etc.)
- Description :- App Description
- Rating :- Rating out of 5
- Number of ratings :- No. of Ratings given by users
- Price :- Price of the App
- Related apps :- Apps related to installed App
- Dangerous (D) permissions count :- No. of Dangerous Permissions allowed by user
- Safe (S) permissions count :- No. of Safe Permissions allowed by user
- Default : Access DRM content. (S) :- 0 : No , 1 : Yes
- Phone calls : modify phone state (S) :- 0 : No , 1 : Yes

# Data Summary

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 29999 entries, 0 to 29998
Data columns (total 10 columns):
 #   Column                       Non-Null Count  Dtype  
---  ------                       --------------  -----  
 0   App                          29998 non-null  object 
 1   Package                      29999 non-null  object 
 2   Category                     29999 non-null  object 
 3   Description                  29996 non-null  object 
 4   Rating                       29999 non-null  float64
 5   Number of ratings            29999 non-null  int64  
 6   Price                        29999 non-null  float64
 7   Related apps                 29244 non-null  object 
 8   Dangerous permissions count  29795 non-null  float64
 9   Safe permissions count       29999 non-null  int64  
dtypes: float64(3), int64(2), object(5)
memory usage: 2.3+ MB
```

**Showing summary for only 10 columns**

# Data Summary

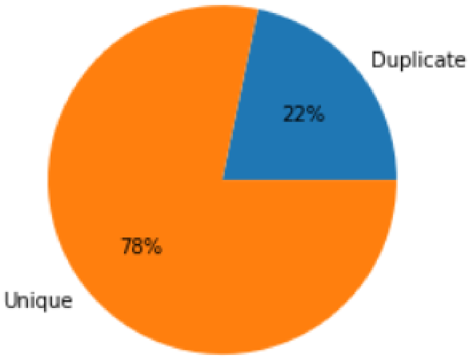
Numerical Features

| | count | mean | std | min | 25% | 50% | 75% | max |
|---|---|---|---|---|---|---|---|---|
| **Rating** | 29999.0 | 3.537215 | 1.424685 | 0.0 | 3.3 | 4.0 | 4.4 | 5.00 |
| **Number of ratings** | 29999.0 | 6852.608454 | 45868.991636 | 0.0 | 4.0 | 46.0 | 716.0 | 1908590.00 |
| **Price** | 29999.0 | 0.625707 | 3.222620 | 0.0 | 0.0 | 0.0 | 0.0 | 158.07 |
| **Dangerous permissions count** | 29795.0 | 3.111160 | 3.052602 | 0.0 | 1.0 | 2.0 | 4.0 | 30.00 |
| **Safe permissions count** | 29999.0 | 1.353978 | 1.523491 | 0.0 | 0.0 | 1.0 | 2.0 | 16.00 |

Categorical Features

| | count | unique | top | freq |
|---|---|---|---|---|
| **App** | 29998 | 22823 | Tic Tac Toe | 47 |
| **Package** | 29999 | 23485 | com.shazam.android | 10 |
| **Category** | 29999 | 30 | Entertainment | 2827 |
| **Description** | 29996 | 23552 | Phrasebook and Translator contains all the ess... | 40 |
| **Related apps** | 29244 | 23868 | {com.openkava.spinpic} | 38 |

# Data Pre-Processing- Duplicate



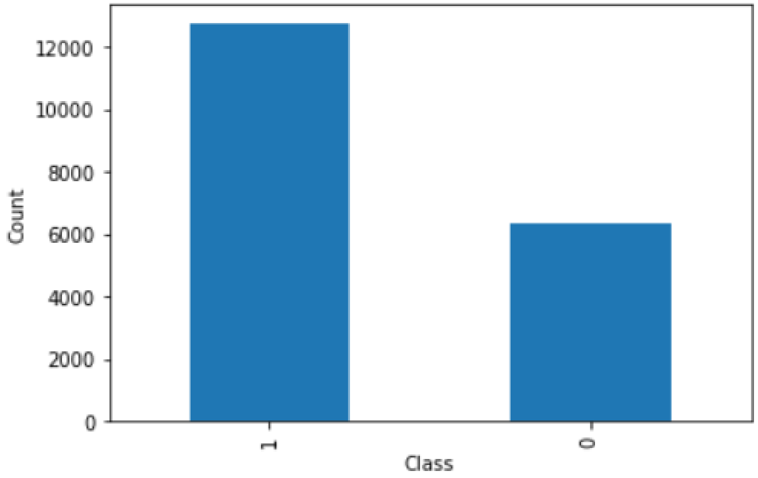

- Checking Duplicate values for the feature ‘Package’
- We drop the rows with duplicate values

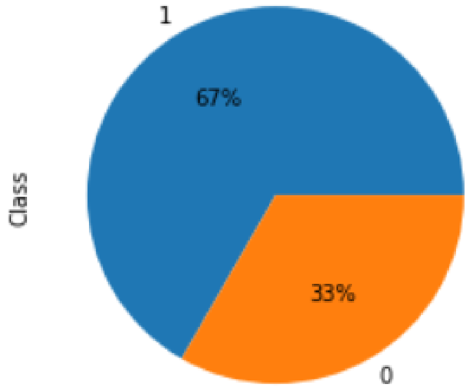
# Data Pre-Processing- Missing

| | total_missing_values | missing_percentage |
|---|---|---|
| **App** | 1.0 | 0.01 |
| **Description** | 3.0 | 0.02 |
| **Related apps** | 610.0 | 3.17 |
| **Dangerous permissions count** | 169.0 | 0.88 |

- We drop "App“, "Description" and "Related apps" from the dataset.
- We drop the rows with **missing values** of dangerous permission count.

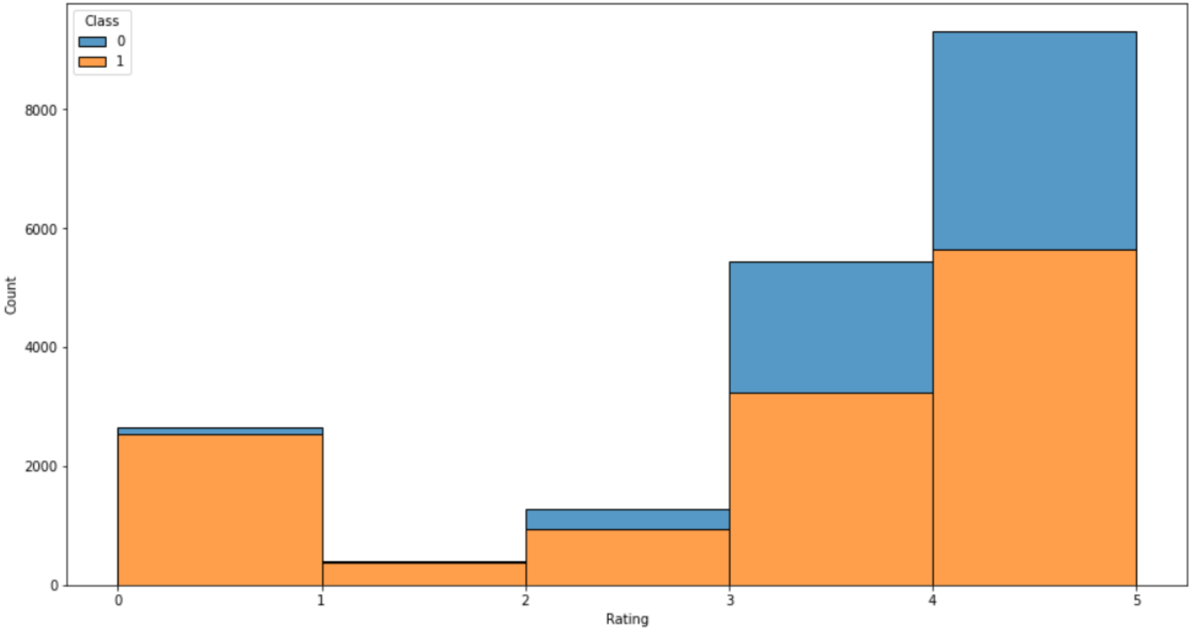
# EDA – Dependent Feature



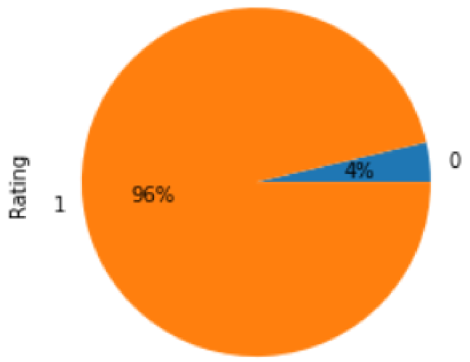




67% apps are malware and rest 33% are Benign in total.

# EDA – Rating



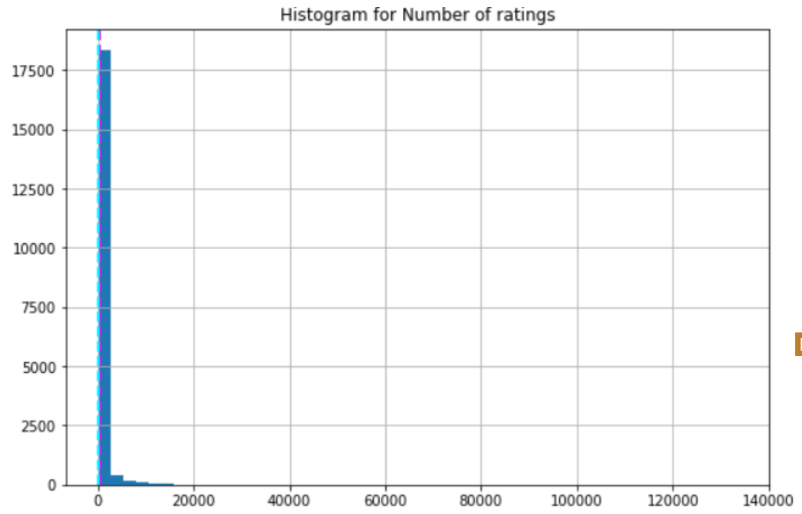

Only for Zero Rating



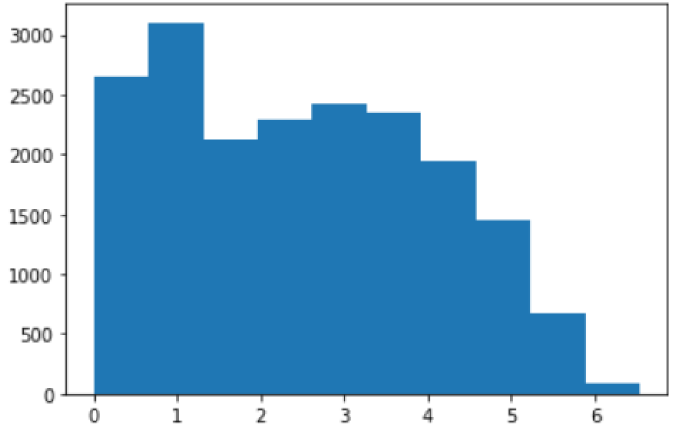

Between Rating 0 to 3, most of the apps have malware. From 3 to 5, there are more benign apps as compared to ratings between 0-3.

96% apps are malware if it has 0 rating

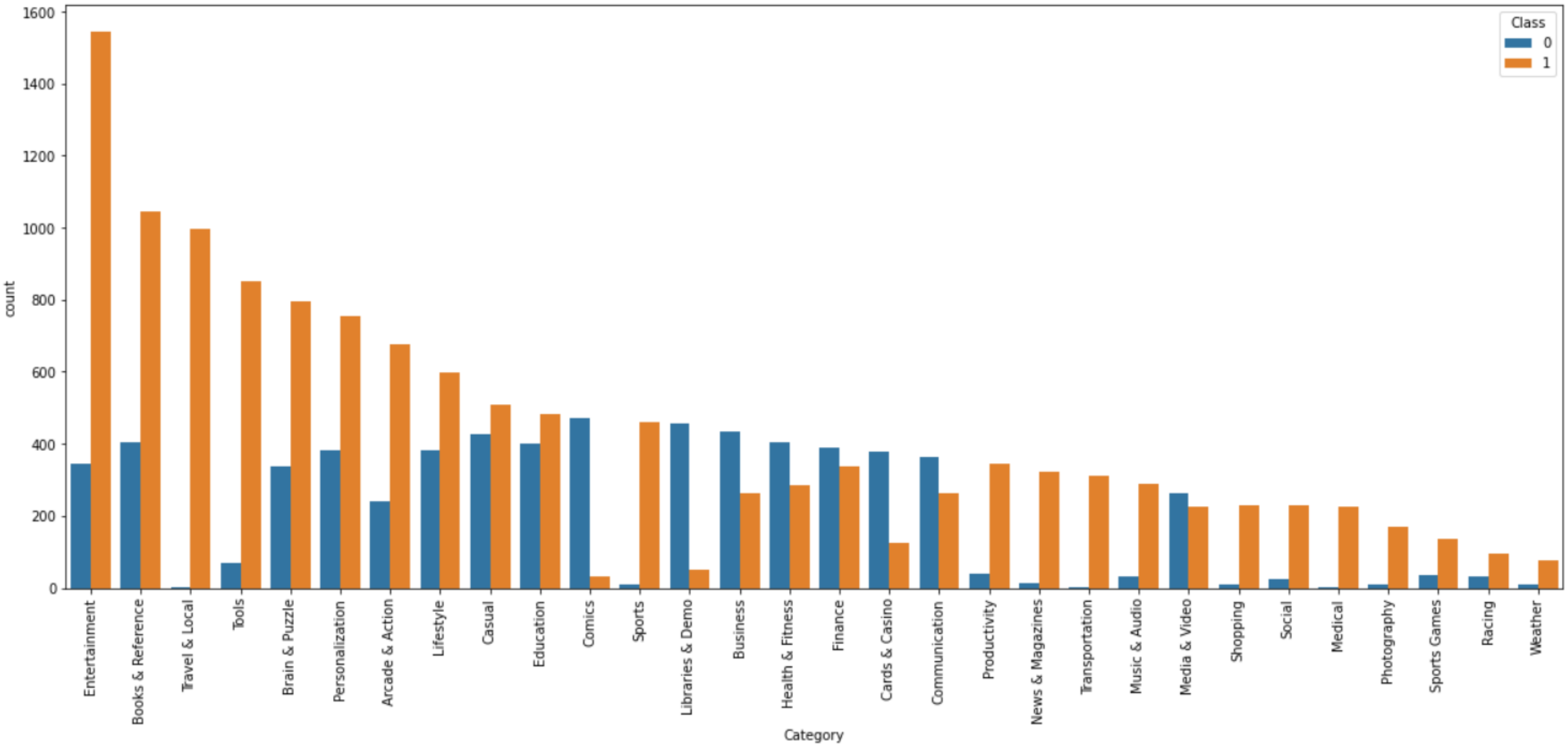
# EDA – Number of ratings



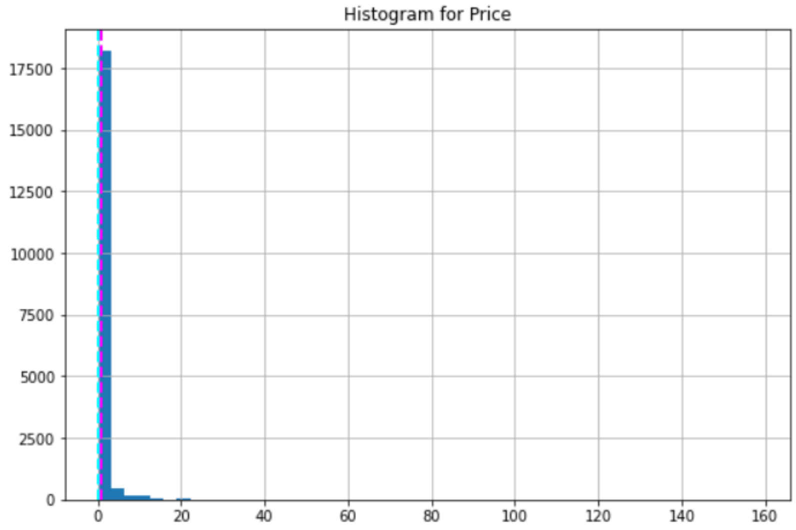

Using **Boxcox**

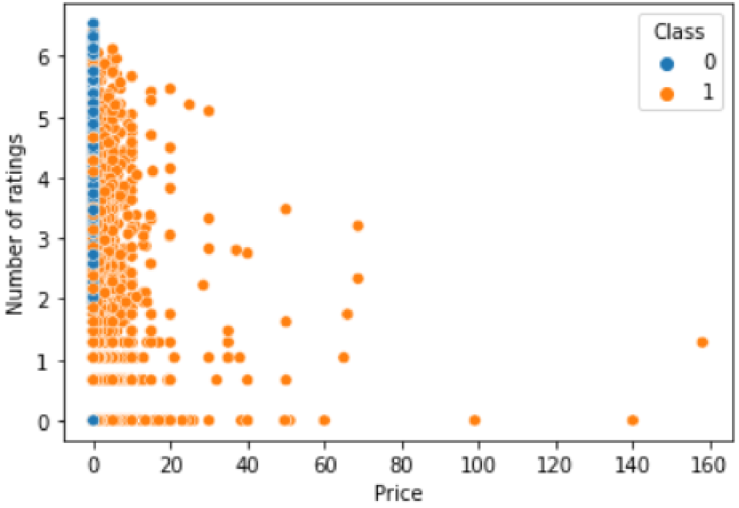


# EDA – Category



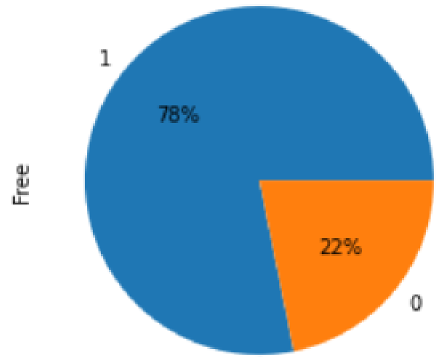
# EDA – Price

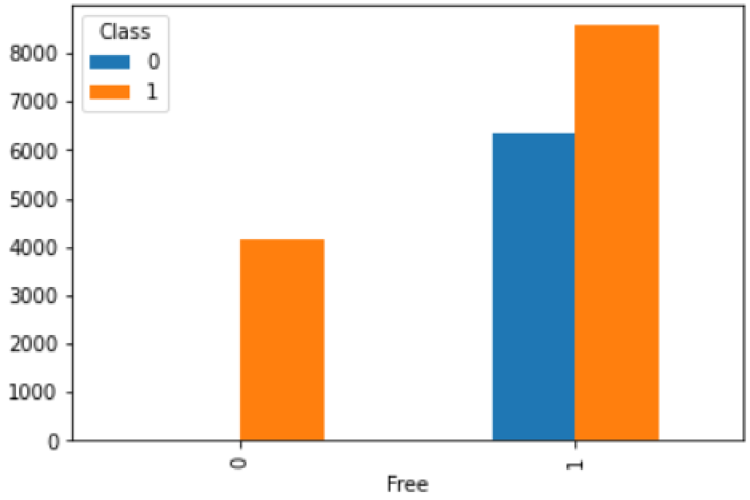



- For apps priced between 0 to 20 has got most number of ratings by customers.

# EDA – Free

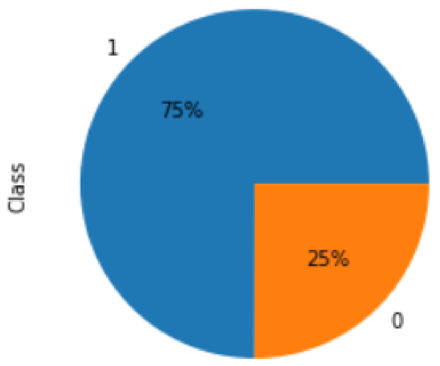


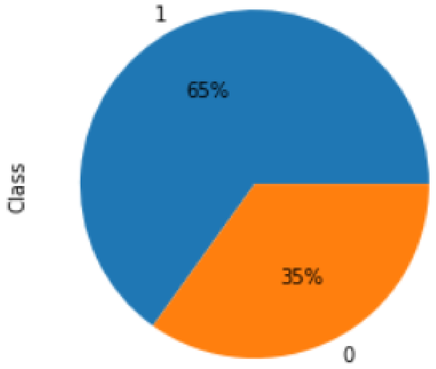

**A new feature is created –'Free' from 'Price'**

78% apps are free
All paid apps are malware
In Free version, number of malware apps is higher than benign.

# EDA- Dangerous permissions count

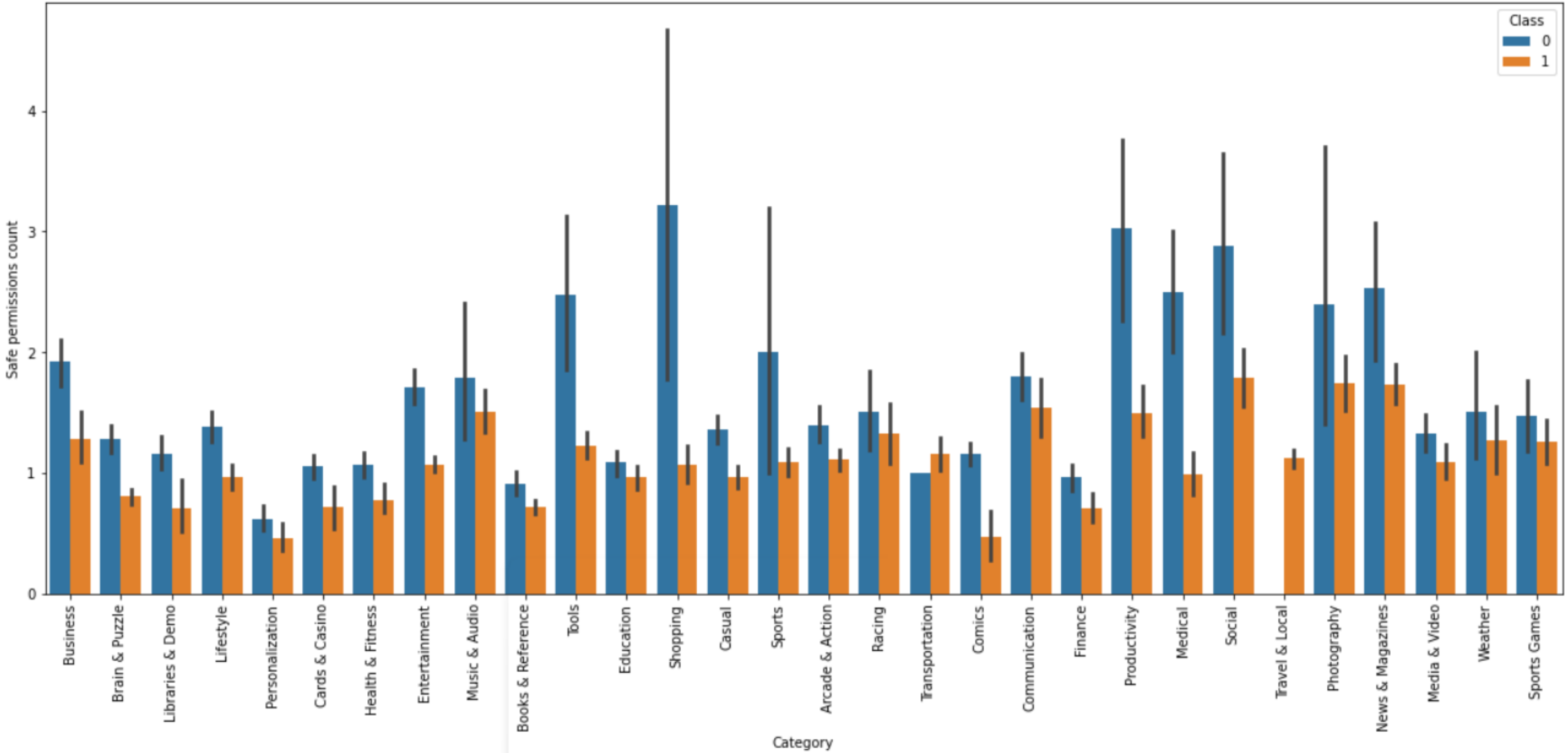



App without Dangerous permissions(= 0)




App with Dangerous permissions(> 0)

# EDA- Safe permissions count



# Feature Selection- All permission columns - all columns containing android permissions

| | permission | frequency |
|---|---|---|
| **0** | Default : Access DRM content. (S) | 4 |
| **1** | Default : Access Email provider data (S) | 10 |
| **2** | Default : Access all system downloads (S) | 0 |
| **3** | Default : Access download manager. (S) | 8 |
| **4** | Default : Advanced download manager functions.... | 1 |
| **...** | ... | ... |
| **168** | Your personal information : retrieve system in... | 5 |
| **169** | Your personal information : set alarm in alarm... | 7 |
| **170** | Your personal information : write Browser's hi... | 235 |
| **171** | Your personal information : write contact data... | 593 |
| **172** | Your personal information : write to user defi... | 15 |

173 rows × 2 columns

Frequency<201

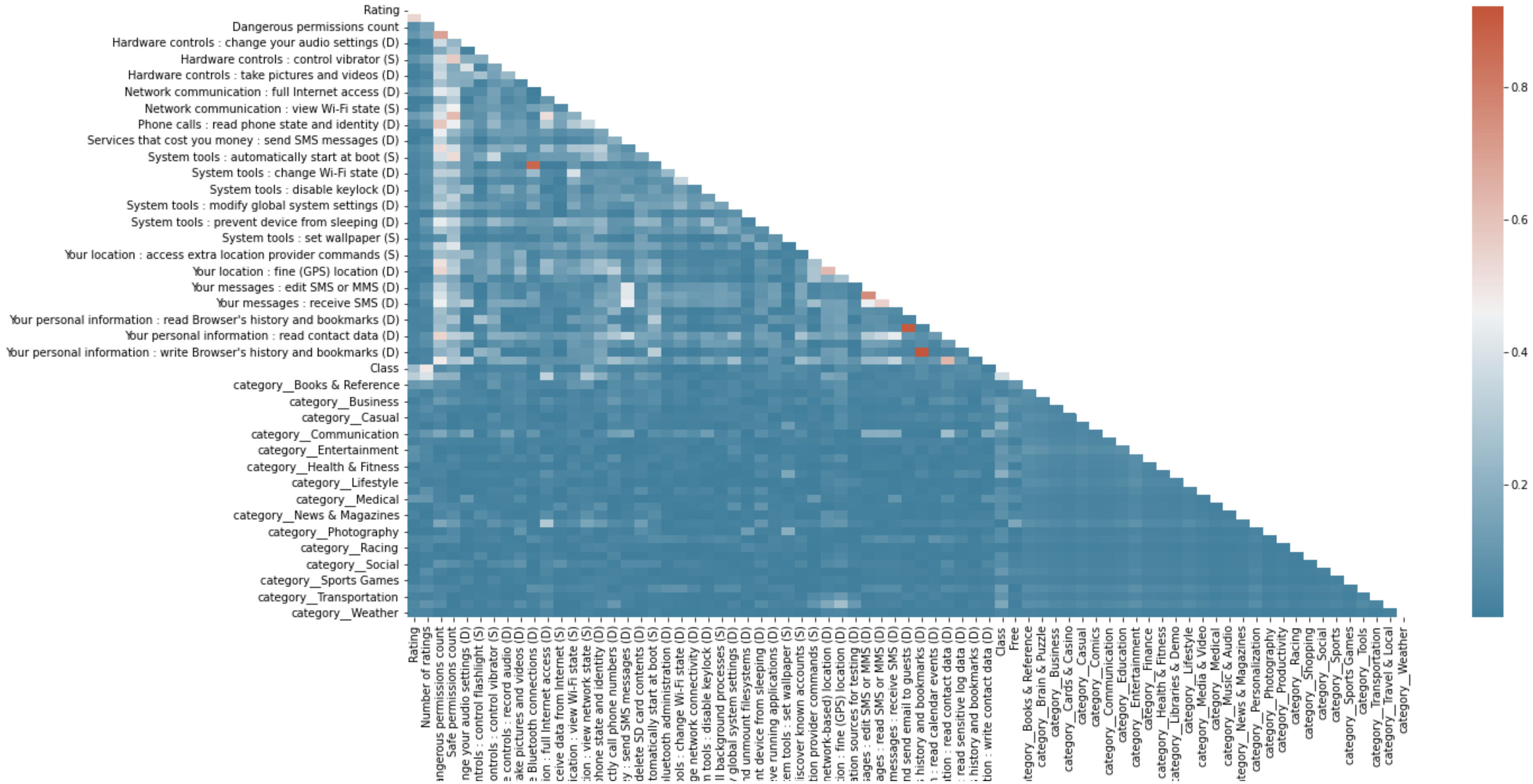
| | permission | frequency |
|---|---|---|
| **0** | Default : Access DRM content. (S) | 4 |
| **1** | Default : Access Email provider data (S) | 10 |
| **2** | Default : Access all system downloads (S) | 0 |
| **3** | Default : Access download manager. (S) | 8 |
| **4** | Default : Advanced download manager functions.... | 1 |
| **...** | ... | ... |
| **162** | Your personal information : choose widgets (S) | 26 |
| **167** | Your personal information : read user defined ... | 14 |
| **168** | Your personal information : retrieve system in... | 5 |
| **169** | Your personal information : set alarm in alarm... | 7 |
| **172** | Your personal information : write to user defi... | 15 |

133 rows × 2 columns

Took all 173 columns that are related to different android permissions and calculated the frequency of each.

So we found total of 133 permission features that are rarely used. We drop these permission columns from our dataset.

# Feature Selection- Correlation




Correlated Features above 90% are dropped: i) 'Your personal information : read calendar events (D)‘ and ii) ‘Your personal information : write Browser's history and bookmarks (D)’

# Feature Selection- VIF

```
dropped_variables = calculate_vif(df, thresh = 10)
```

```
dropping 'Dangerous permissions count' at index: 2
dropping 'Safe permissions count' at index: 2
Remaining variables:
Index(['Rating', 'Number of ratings',
       'Hardware controls : change your audio settings (D)',
       'Hardware controls : control flashlight (S)',
       'Hardware controls : control vibrator (S)',
       'Hardware controls : record audio (D)',
       'Hardware controls : take pictures and videos (D)',
       'Network communication : create Bluetooth connections (D)',
       'Network communication : full Internet access (D)',
       'Network communication : receive data from Internet (S)',
       'Network communication : view Wi-Fi state (S)',
       'Network communication : view network state (S)',
       'Phone calls : read phone state and identity (D)',
       'Services that cost you money : directly call phone numbers (D)',
       'Services that cost you money : send SMS messages (D)',
       'Storage : modify/delete USB storage contents modify/delete SD card contents (D)',
       'System tools : automatically start at boot (S)',
       'System tools : bluetooth administration (D)',
       'System tools : change Wi-Fi state (D)',
       'System tools : change network connectivity (D)',
       'System tools : disable keylock (D)',
       'System tools : kill background processes (S)',
       'System tools : modify global system settings (D)',
       'System tools : mount and unmount filesystems (D)',
       'System tools : prevent device from sleeping (D)',
       'System tools : retrieve running applications (D)',
       'System tools : set wallpaper (S)',
       'Your accounts : discover known accounts (S)',
       'Your location : access extra location provider commands (S)',
       'Your location : coarse (network-based) location (D)',
       'Your location : fine (GPS) location (D)',
```

Calculated VIF and dropped features with threshold >10

Dropped columns are:
i) 'Dangerous permissions count' and ii) 'Safe permissions count'

# Machine Learning Algorithms

- Logistic Regression
- Decision tree
- Random Forest
- Gradient Boost
- KNN
- Naive Bayes
- XGBoost

# Model's Evaluation Matrices

| | | AUC | Accuracy | Precision | Recall | F1 Score | Confusion Metrix |
|---|---|---|---|---|---|---|---|
| Train | Logistic Regression | 0.863173 | 0.877206 | 0.908824 | 0.905958 | 0.907389 | [[3686, 807], [835, 8044]] |
| | Decision Tree | 0.708777 | 0.761816 | 0.791581 | 0.870481 | 0.829158 | [[2458, 2035], [1150, 7729]] |
| | Random Forest | 0.997768 | 0.99813 | 0.998312 | 0.998874 | 0.998593 | [[4478, 15], [10, 8869]] |
| | Gradient Boosting | 0.857012 | 0.873916 | 0.902248 | 0.908548 | 0.905387 | [[3619, 874], [812, 8067]] |
| | KNN | 0.896516 | 0.903455 | 0.935691 | 0.917671 | 0.926594 | [[3933, 560], [731, 8148]] |
| | Naive Bayes | 0.783651 | 0.728238 | 0.96244 | 0.614709 | 0.750241 | [[4280, 213], [3421, 5458]] |
| | XGBoost | 0.855491 | 0.86883 | 0.905336 | 0.896159 | 0.900724 | [[3661, 832], [922, 7957]] |
| Test | Logistic Regression | 0.863992 | 0.878053 | 0.91368 | 0.904454 | 0.909044 | [[1540, 330], [369, 3493]] |
| | Decision Tree | 0.700614 | 0.754536 | 0.795427 | 0.855774 | 0.824498 | [[1020, 850], [557, 3305]] |
| | Random Forest | 0.864946 | 0.875436 | 0.91795 | 0.895132 | 0.906397 | [[1561, 309], [405, 3457]] |
| | Gradient Boosting | 0.848652 | 0.864445 | 0.90369 | 0.894096 | 0.898868 | [[1502, 368], [409, 3453]] |
| | KNN | 0.84411 | 0.855722 | 0.905423 | 0.877525 | 0.891256 | [[1516, 354], [473, 3389]] |
| | Naive Bayes | 0.775085 | 0.717551 | 0.95497 | 0.609529 | 0.744113 | [[1759, 111], [1508, 2354]] |
| | XGBoost | 0.846328 | 0.858339 | 0.906233 | 0.880891 | 0.893382 | [[1518, 352], [460, 3402]] |

**Observation: Logistic Regression has the highest F1 Score for testing dataset. So, we will select this model and find the best hyper parameters for it.**

# Hyperparameter Tuning

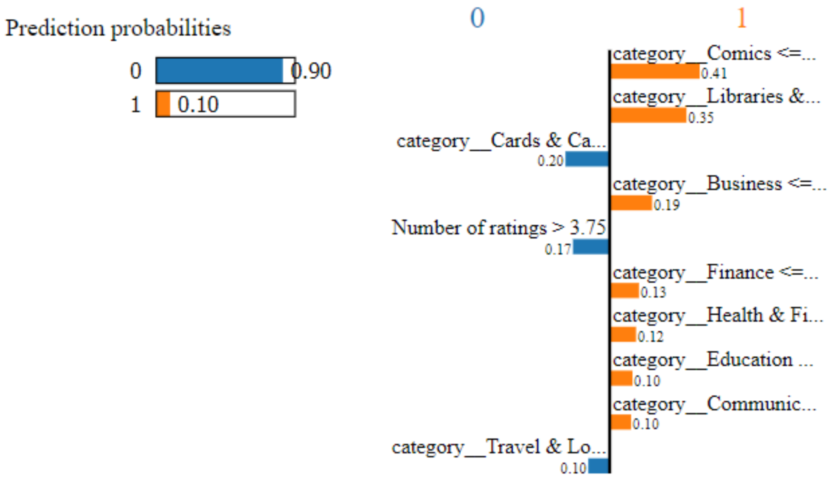
Logistic Regression → Randomized Search CV → Logistic Regression (Tuned)

```
For Train Data:
AUC  :  0.8631725736123369
Accuracy  :  0.8772061023033204
Precision  :  0.9088238617105412
Recall  :  0.9059578781394301
F1 Score  :  0.9073886068809927
Confusion Metrix  :  [[3686  807]
 [ 835 8044]]
----------------------------------------
For Test Data:
AUC  :  0.8639915313613794
Accuracy  :  0.8780530355896721
Precision  :  0.9136803557415643
Recall  :  0.9044536509580529
F1 Score  :  0.9090435914118414
Confusion Metrix  :  [[1540  330]
 [ 369 3493]]
```

C=10000.0,
max_iter=8000,
penalty='none',
solver='saga'

```
For Train Data:
AUC  :  0.8640051936479535
Accuracy  :  0.8776548010768771
Precision  :  0.9097182939246521
Recall  :  0.9056200022525059
F1 Score  :  0.9076645219550739
Confusion Metrix  :  [[3695  798]
 [ 838 8041]]
----------------------------------------
For Test Data:
AUC  :  0.8652074096433923
Accuracy  :  0.8785764131193301
Precision  :  0.9150498164656529
Recall  :  0.9036768513723459
F1 Score  :  0.9093277748827514
Confusion Metrix  :  [[1546  324]
 [ 372 3490]]
```

# Model Explainability - LIME




| Feature | Value |
|---|---|
| category__Comics | 0.00 |
| category__Libraries & Demo | 1.00 |
| category__Business | 0.00 |
| category__Cards & Casino | 0.00 |
| Number of ratings | 5.22 |
| category__Health & Fitness | 0.00 |
| category__Education | 0.00 |
| category__Finance | 0.00 |
| category__Lifestyle | 0.00 |
| category__Travel & Local | 0.00 |

# Model Explainability – ELI5

| Contribution? | Feature | Value |
|---:|---|---:|
| +10.546 | Free | 1.000 |
| +5.906 | Number of ratings | 5.011 |
| +1.624 | category__Casual | 1.000 |
| +0.683 | Network communication : view network state (S) | 1.000 |
| +0.513 | Rating | 4.200 |
| -0.029 | Network communication : full Internet access (D) | 1.000 |
| -0.242 | Phone calls : read phone state and identity (D) | 1.000 |
| -16.042 | <BIAS> | 1.000 |

# Conclusion

i. 22 % rows consists of duplicate values.
ii. Given dataset is slightly imbalanced because 67% apps are malware and rest 33% are Benign.
iii. Between Rating 0 to 3, most of the apps have malware. From 3 to 5, there are more benign apps as compared to ratings between 0-3.
iv. For the categories, ‘Travel & Local’, ‘Tools’, ‘Sports’ etc., almost all apps are malware. For the categories, ‘Comics’, ‘Libraries & Demo’ etc, almost all apps are benign.
v. All paid apps are malware and number of malware apps is higher than benign in the free apps. But it does not makes sense for all paid apps are to be malware. It may be due to misclassification of apps.
vi. We use F1 score since our dataset is slightly imbalanced and there is a serious downside to predicting false negatives. Among all models, Logistic Regression has the best F1 Score of almost 91% for both train and test dataset.

# Challenges

- The biggest challenge we had to overcome was that the  number of features in our dataset was above 180.
- We had multiple classification models which gave slightly lower than our best model score.
- Feature Selection was a very big challenge.
- Computation Time is also one of the major challenge.

# Thank You