Intro to Machine Learnings Application

Fall
2020

Microsoft Malware Prediction

FINAL PROJECT REPORT DRAFT

JAGRATI SHARMA

Table of Contents

1. Executive Summary

This Kaggle competition is to predict the probability of a malware occurrence on the machine. Microsoft hosted this competition, Windows Defender ATP Research, Northeastern University College of Computer and Information Science, and Georgia Tech Institute for Information Security & Privacy. The Windows machine's data is provided in the test and train dataset. That is, each row in the dataset represents a unique Windows machine.

The training data is of size 4.08 GB, and the testing data is of size 3.54 GB. The unique identifier in the datasets is the 'MachineIdentifier' column. The column to be predicted for the test dataset is the 'HasDetected' column, which tells if the machine has detected malware or not.

The dataset consists of 82 columns and one target column. Initially, the train data consist of 89,21,483 rows, and test data consist of 7853253 rows. For now, I considered using 4,00,000 rows from the beginning for both the train and test dataset for this project.

There were columns with the object data type, which I transformed into a categorical variable or float type appropriately. The dataset consists of columns that have more than 95% of data empty along with more than 50% of columns that have no data empty. I found that column 'PuaMode,' which represents 'Pua Enabled mode from the service' is 99.97% empty, and a column named 'Census_ProcessorClass' representing, 'A classification of processors into high/medium/low. Initially used for Pricing Level SKU' is 99.56% empty. The data description mentions that this column is no longer maintained and updated.

I also calculated the percentage of value in the most significant category to find whether the column has different values to determine the target column. I found that columns like isBeta, AutSampleOptIn, PauMode, Census_IsPortableOperatingSystem, Census_DeviceFamily, Census_ProcessorClass, UacLuaenable, and Census_IsVirtualDevice have more than 99% values in the same categories. That implied that they do not contribute much to the prediction of the target variable.

The target column has around 50% of the data that denoted the malware has detected in the machine, and the other 50% says that it is not detected. There is significantly less or no correlation among the variables.

2. Benchmarking of Other Solutions

Notebook Name	Feature Approach	Model Approach	Pvt/Public Score
Detecting Malwares with LGBM (https://www.kaggle.com/fabiendaniel/detecting-malwares-with-lgbm/log)	This notebook uses a utility function that helps managing memory. They first make a census of the variables, by type, and define the set. Depending on the cardinality of each variable, notebook used one-hot-encoding, frequency and label encoding.	This notebook uses sparse matrix and Light GBM model using only a subset of the training data in 5 iterations, in order to fit in memory. The prediction is performed in chunks of size 100,000.	**Private Score** 0.61934 **Public Score** 0.67746
Beginning Challenge (https://www.kaggle.com/nikkisharma536/beginning-challenge)	This notebook first iterates through all the columns of the data-frame and modify the data type to reduce memory usage. This notebook also makes use of label encoding and uses search grid for optimal parameters before moving to apply the model to get the best score and prediction.	This model also uses Light GBM along with GridSearchCV model for the prediction. The fitting is 5 folds for each of 1 candidate, totaling to 5 fits.	**Private Score** 0.54554 **Public Score** 0.59653
Random Forest Feature Importances (https://www.kaggle.com/harmeggels/random-forest-feature-importances)	This notebook uses the utility function that helps managing memory. The notebook uses backward approach for selecting the variables.	The notebook uses random forest classifier for modelling. The validation set split is based on the index. The notebook uses FastAI deep learning library to reduce and allow faster repetition cycles. It also immediately creates a reset function to removing the columns with have less feature importance.	**Private Score** 0.58521 **Public Score** 0.65542

a. About feature approach

For the first notebook, they first make a census of the variables by type, i.e., numeric or numerical or categorical or binary, and define the set. The notebook limits the size of the training set to 4,000,000 rows. They are using encoding schemes like one-hot encoding and label encoding for the categorical variables. It uses the Light GBM to identify the feature importance. Depending upon the variables, they used the built-in LGBM treatment of categorical. That resulted in an efficient way for feature selection.

The second notebook uses the same approach as the first one for modifying the variables' data, and for reducing the memory, they also use the encoding scheme for the categorical variables. But they are also trying to replace the null value with mean values. They try to use the search grid for choosing the optimal parameter.

The third notebook uses the same method as the other two for the data-modifying and the reduction of the memory. The training data replaced the category variables with the category codes and replaced the nan values in the numerical columns with the median. This notebook uses a backward approach for selecting the best variables to be used in the modeling. It checks for the overfitting of the model and the model's performance, followed by looking for the feature importance of the model to discard the variables that are not necessary. Then the notebook applies the models again and follows the steps to check the feature importance till the time it is optimal.

b. For modeling approaches

The first notebook uses sparse matrix along Light GBM model using only a subset of the training data to fit in memory. For training the model, the notebook performs the training in 5 iterations. It is trained using Light GBM. It is a framework that uses a tree-based learning algorithm. The notebook performs training until validation scores don't improve for 200 rounds. Using such a robust algorithm for modeling gives this notebook an advantage over other of the score.

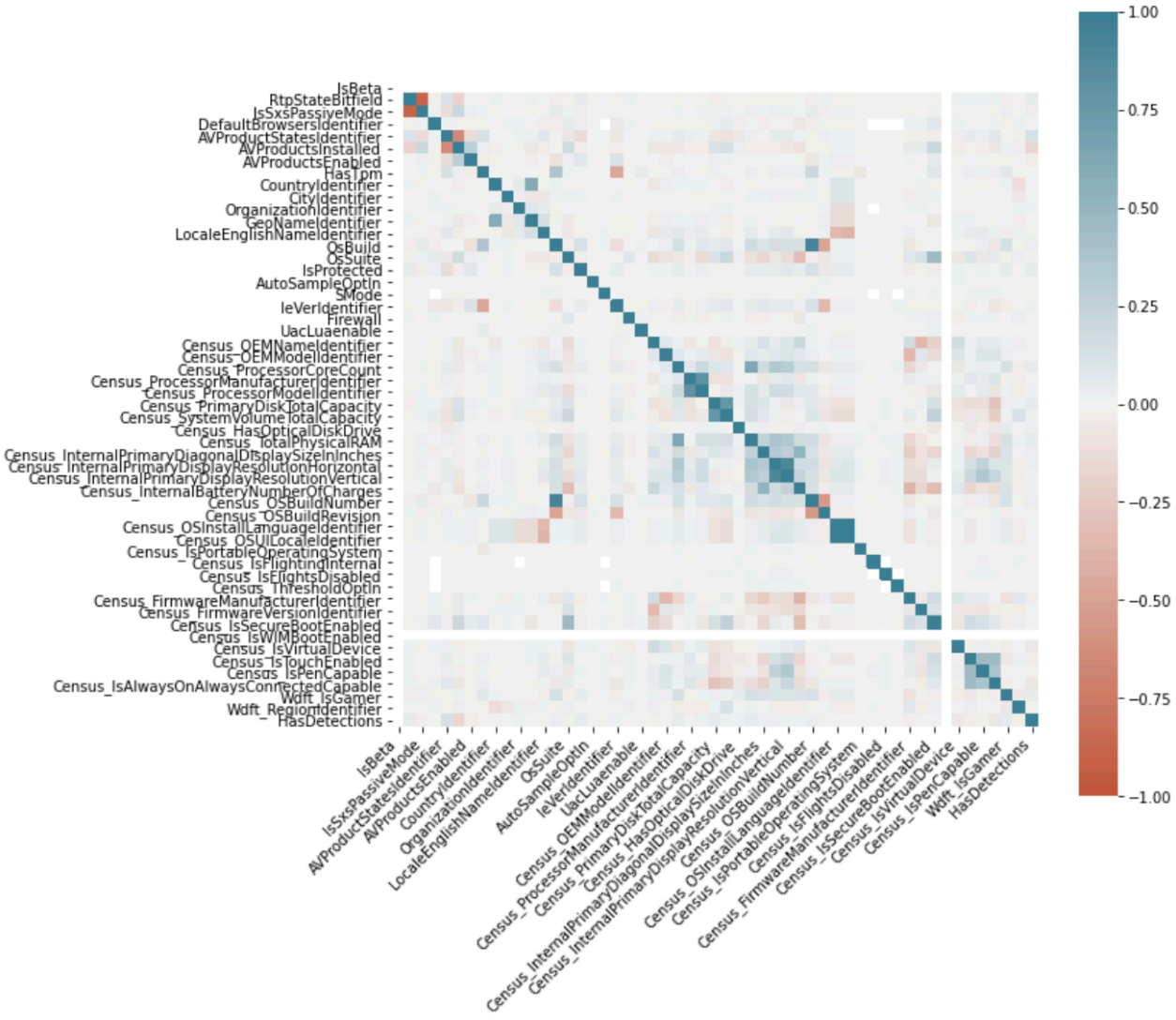
The second notebook also uses Light GBM with the GridSearchCV model for the prediction. The fitting is five folds for each of 1 candidate, totaling five fits. This model exhaustively generates candidates from a grid of parameter values specified with the parameter. Using such a powerful model and the grid searching in this big dataset may result in poor performance compared to the other notebooks.

The third notebook uses a random forest classifier for modeling. The notebook uses FastAI deep learning library to reduce and allow faster repetition cycles. As this is a big dataset, using an in-depth learning approach gives the notebook advantage over the second notebook.

3. Data description and Initial Processing

The dataset uses only 2,00,000 rows from the beginning of the original dataset for both training and testing as compared to the original 89,21,483 and 78,53,253 rows respectively. The datatypes are also modified for a better approach to the dataset understanding and visualization. The object variables are loaded as categories, the binary values are switched to int8, for the binary values which consist of the missing values are switched to float16 datatype, for better memory usage the 64bits encoding are switched to 32- or 16-bit encoding.

The correlation plot below is on raw/unprocessed data. The data looks highly unrelated or with minimum relation with each other.



3.1. Columns with biased values

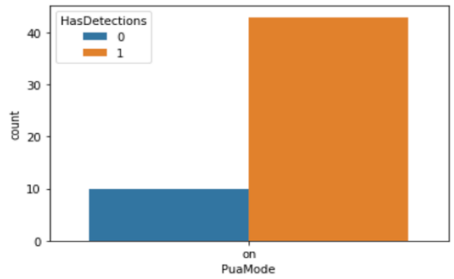
As a next step in the data preprocessing, percentage of values in the biggest category are identified. As this will help in identifying the columns which consist of values belonging to only a single category which will not have any significant impact on the target values. The dataset contains columns which have more than 90% of values belonging to the same category.

	Feature	Unique_values	Percentage of values in the biggest category	type
5	IsBeta	1	100.0000	int8
27	AutoSampleOptIn	2	99.9965	int8
28	PuaMode	1	99.9735	category
65	Census_IsPortableOperatingSystem	2	99.9315	int8
35	Census_DeviceFamily	2	99.8410	category
41	Census_ProcessorClass	3	99.5635	category
33	UacLuaenable	5	99.2915	float32
76	Census_IsVirtualDevice	2	99.1250	float16
1	ProductName	2	98.9195	category
12	HasTpm	2	98.7680	int8
7	IsSxsPassiveMode	2	98.2910	int8
69	Census_IsFlightsDisabled	2	98.2285	float16
11	AVProductsEnabled	5	97.0015	float16
6	RtpStateBitfield	6	96.9735	float16
32	Firewall	2	96.7640	float16
20	OsVer	11	96.7630	category
18	Platform	4	96.6095	category
78	Census_IsPenCapable	2	96.2150	int8
8	DefaultBrowsersIdentifier	363	95.1485	float16
26	IsProtected	2	94.1495	float16
29	SMode	2	94.0360	float16
70	Census_FlightRing	7	93.6615	category
79	Census_IsAlwaysOnAlwaysConnectedCapable	2	93.5110	float16
45	Census_HasOpticalDiskDrive	2	92.3525	int8
55	Census_OSArchitecture	3	90.8530	category
19	Processor	3	90.8465	category

3.1.1. PuaMode

For the column PuaMode, percentage value in the biggest category is 99.97%. The column also has 99.97% of missing values with only one unique value that is ‘on’. Hence, this column is not contributing in any way to the target variable.

```
on     53
Name: PuaMode, dtype: int64
```

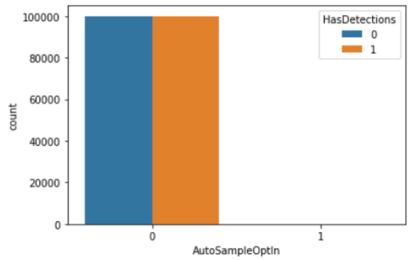



```
199947
```

3.1.2. AutoSampleOptIn

For the column AutoSampleOptIn, percentage value in the biggest category is 99.99%. The column does not have any missing values. They have two unique value that is 0 and 1 which have a huge difference in the count. Hence, this column is not contributing in any way to the target variable.

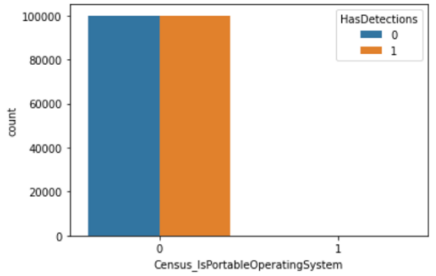
```
0    199993
1         7
Name: AutoSampleOptIn, dtype: int64
```



3.1.3. isPortableOperatingSystem

For the column isPortableOperatingSystem, percentage value in the biggest category is 99.93%. There are two unique values, 0 and 1. The count of 0 values is significantly more than count of 1. Hence this column is biased for one value and have less/no impact on the target variable unless the targeted of accuracy is 99%.

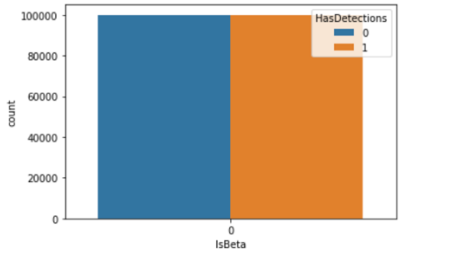
```
0    199863
1       137
Name: Census_IsPortableOperatingSystem, dtype: int64
<matplotlib.axes._subplots.AxesSubplot at 0x7f4c22d922e8>
```



3.1.4. isBeta

For the column isBeta, percentage value in the biggest category is 100%. As this column had only one value without any missing value for the chosen dataset. This column is not useful in predicting the target variable.

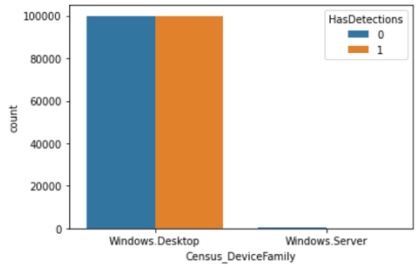
```
0    200000
Name: IsBeta, dtype: int64
<matplotlib.axes._subplots.AxesSubplot at 0x7f4c119901d0>
```



3.1.5. Census_DeviceFamily

For the column Census_DeviceFamily, percentage value in the biggest category is 99.84%. There are two unique values, 0 and 1. The count of 0 values is significantly more than count of 1. Hence this column is biased for one value and have less/no impact on the target variable unless the targeted of accuracy is 99%.

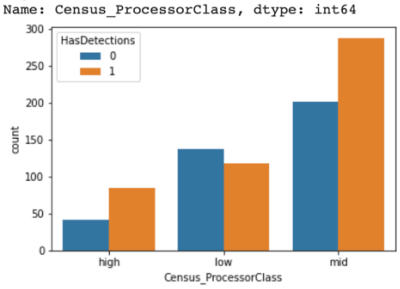
```
Windows.Desktop    199682
Windows.Server        318
Name: Census_DeviceFamily, dtype: int64
```



3.1.6. Census_ProcessorClass

For the column Census_ProcessorClass, percentage value in the biggest category is 99.56%. There are three unique values, mid, low, and high. The count of the missing values is also significantly more than the present values. Hence this column may have less/no impact on the target variable unless the targeted of accuracy is 99%.

```
mid     490
low     256
high    127
Name: Census_ProcessorClass, dtype: int64
```

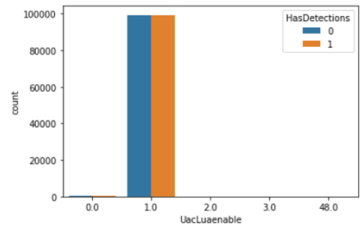



```
199127
```

3.1.7. UacLuaenable

This attribute reports whether or not the "administrator in Admin Approval Mode" user type is disabled or enabled in UAC. For this, percentage value in the biggest category is 99.29%. The count of 1.0 value is significantly more than count of others. Hence this column is biased for one value and have less/no impact on the target variable unless the targeted of accuracy is 99%.

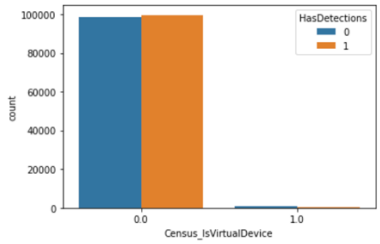
```
1.0     198583
0.0       1158
48.0         9
3.0          1
2.0          1
Name: UacLuaenable, dtype: int64
```



3.1.8. Census_IsVirtualDevice

For the column Census_IsVirtualDevice, percentage value in the biggest category is 99.12%. There are two unique values, 0 and 1. The count of 0 values is significantly more than count of 1. Hence this column is biased for one value and have less/no impact on the target variable unless the targeted of accuracy is 99%.

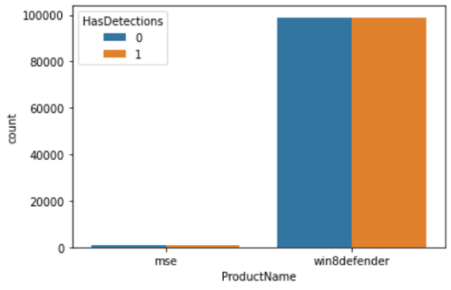
```
0.0     198250
1.0       1402
Name: Census_IsVirtualDevice, dtype: int64
```



3.1.9. ProductName

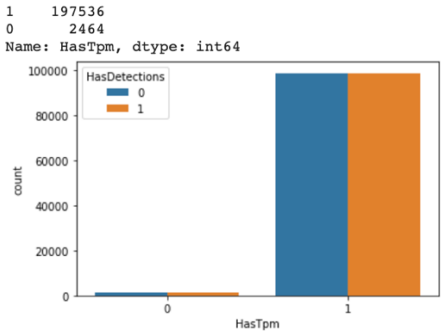
For the column ProductName, percentage value in the biggest category is 98.91%. There are two unique values, win8defender and mse. The count of one value is significantly more than count of other. This column has no missing values. Hence, this column is biased for one value and have less/no impact on the target variable unless the targeted of accuracy is 98%.

```
win8defender    197839
mse               2161
Name: ProductName, dtype: int64
```



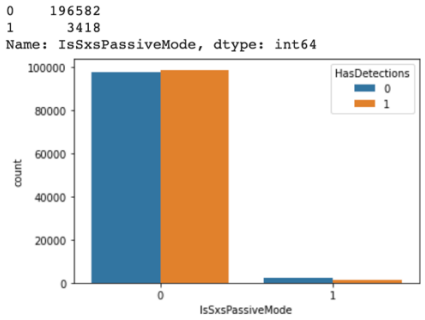
3.1.10. HasTpm

For the column HasTpm, percentage value in the biggest category is 98.76%. There are two unique values, 0 and 1. The count of 1 value is significantly more than count of 0. Hence this column is biased for one value and have less/no impact on the target variable unless the targeted of accuracy is 98%.



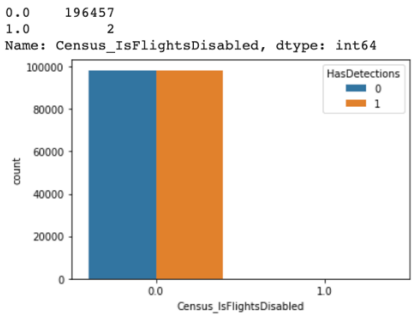
3.1.11. IsSxsPassiveMode

For the column IsSxsPassiveMode, percentage value in the biggest category is 98.29%. There are two unique values, 0 and 1. The count of 0 values is significantly more than count of 1. Hence this column is biased for one value and have less/no impact on the target variable unless the targeted of accuracy is 98%.



3.1.12. Census_IsFlightsDisabled

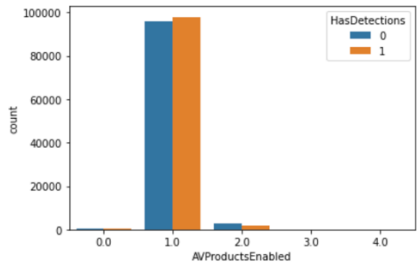
For the column Census_IsFlightsDisabled, percentage value in the biggest category is 98.22%. There are two unique values, 0 and 1. The count of 0 values is significantly more than count of 1. Hence this column is biased for one value and have less/no impact on the target variable unless the targeted of accuracy is 98%.



3.1.13. AVProductsEnabled

For the column AVProductsEnabled, percentage value in the biggest category is 97%. There are five unique values, 0.0, 1.0, 2.0, 3.0, and 4.0. The count of one value is significantly more than count of others. Hence this column is biased for one value and have less/no impact on the target variable unless the targeted of accuracy is 97%.

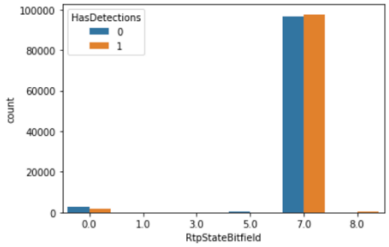
```
1.0    194003
2.0      4514
0.0       556
3.0       116
4.0         7
Name: AVProductsEnabled, dtype: int64
```



3.1.14. RtpStateBitfield

For the column RtpStateBitfield, percentage value in the biggest category is 96.97%. There are six unique values. The count of one value is significantly more than count of others. Hence this column is biased for one value and have less/no impact on the target variable unless the targeted of accuracy is 96%.

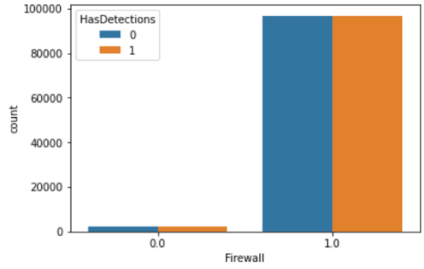
```
7.0    193947
0.0      4198
8.0       537
5.0       450
3.0        62
1.0        42
Name: RtpStateBitfield, dtype: int64
```



3.1.15. Firewall

For the column Firewall, percentage value in the biggest category is 96.76%. There are two unique values, 0 and 1. The count of 1 value is significantly more than count of 0. Hence this column is biased for one value and have less/no impact on the target variable unless the targeted of accuracy is 96%.

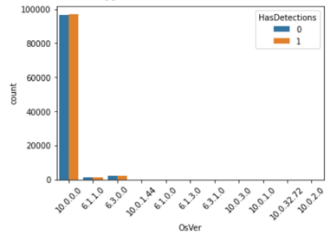
```
1.0    193528
0.0      4394
Name: Firewall, dtype: int64
```



3.1.16. OsVer

For the column OsVer, percentage value in the biggest category is 96.76%. The count of one value is significantly more than count of others. Hence this column is biased for one value and have less/no impact on the target variable unless the targeted of accuracy is 96%.

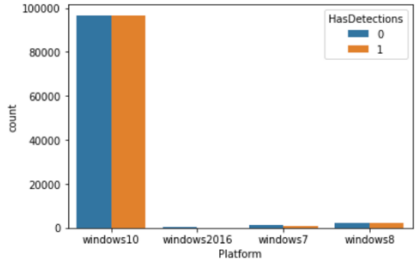
```
10.0.0.0      193526
6.3.0.0         4324
6.1.1.0         2123
6.1.0.0           13
10.0.1.0           4
10.0.3.0           4
6.3.1.0            2
10.0.2.0           1
10.0.32.72         1
6.1.3.0            1
10.0.1.44          1
Name: OsVer, dtype: int64
```



3.1.17. Platform

For the column Platform, percentage value in the biggest category is 96.60%. There are four unique versions of windows. The count of one value is significantly more than count of others. Hence this column is biased for one value and have less/no impact on the target variable unless the targeted of accuracy is 96%.

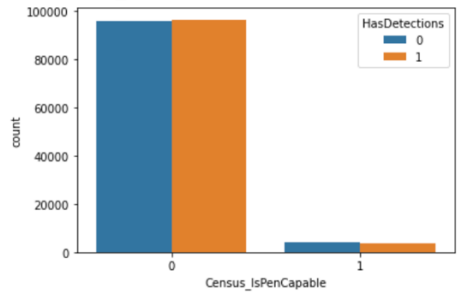
```
windows10      193219
windows8         4326
windows7         2137
windows2016       318
Name: Platform, dtype: int64
```



3.1.18. Census_IsPenCapable

For the column Census_IsPenCapable, percentage value in the biggest category is 96.21%. There are two unique values, 0 and 1. The count of 0 values is significantly more than count of 1. Hence this column is biased for one value and have less/no impact on the target variable unless the targeted of accuracy is 96%.

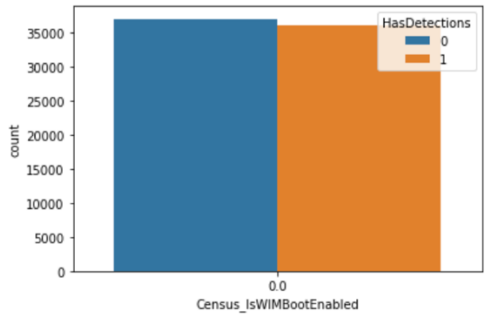
```
0    192430
1      7570
Name: Census_IsPenCapable, dtype: int64
```



3.1.19. Census_IsWIMBootEnabled

The columns with more than 96% of percentage are removed from the dataset. As they were imbalances and biased to one value and not others. Apart from these columns, the dataset consists of column Census_IsWIMBootEnabled, it has only one unique value. This means that the column plays no contribution in predicting the target value. We remove this column as well from the train and test dataset.

```
0.0    73161
Name: Census_IsWIMBootEnabled, dtype: int64
```



3.2. Missing Values

The dataset also consists of missing data. The following table shows percentage of the missing values in the training dataset after removing the columns with imbalances in the values. There are 4 columns which contains more than 50% of the missing values.

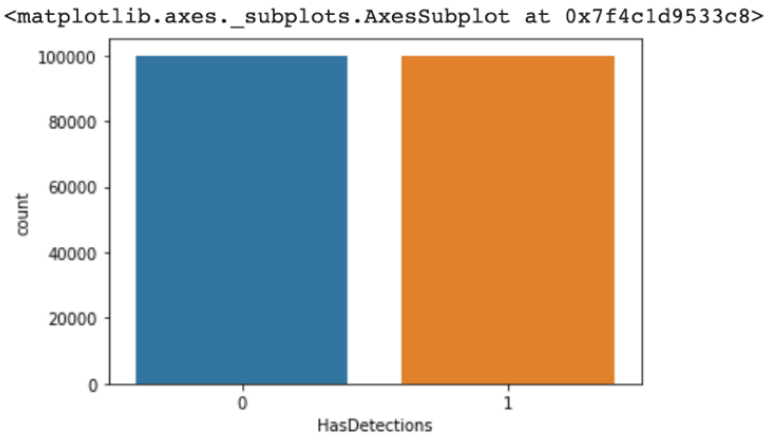
	Feature	Unique_values	Percentage of missing values	type
4	DefaultBrowsersIdentifier	363	95.1485	float16
53	Census_IsFlightingInternal	2	83.0340	float16
38	Census_InternalBatteryType	23	71.0440	category
55	Census_ThresholdOptIn	2	63.5040	float16
21	SmartScreen	11	35.5615	category
9	OrganizationIdentifier	40	30.7685	float16
19	SMode	2	5.9315	float16
8	CityIdentifier	23378	3.6075	float32
62	Wdft_RegionIdentifier	15	3.3685	float16
61	Wdft_IsGamer	2	3.3685	float16
39	Census_InternalBatteryNumberOfCharges	2780	3.0325	float32
56	Census_FirmwareManufacturerIdentifier	226	2.0505	float16
57	Census_FirmwareVersionIdentifier	16797	1.7930	float32
24	Census_OEMModelIdentifier	24416	1.1225	float32
23	Census_OEMNameIdentifier	973	1.0410	float16
32	Census_TotalPhysicalRAM	283	0.9110	float32
60	Census_IsAlwaysOnAlwaysConnectedCapable	2	0.8105	float16
20	IeVerIdentifier	167	0.6810	float16
48	Census_OSInstallLanguageIdentifier	39	0.6655	float16
30	Census_SystemVolumeTotalCapacity	76808	0.5865	float32
28	Census_PrimaryDiskTotalCapacity	711	0.5865	float32
34	Census_InternalPrimaryDiagonalDisplaySizeInInches	425	0.5555	float16
36	Census_InternalPrimaryDisplayResolutionVertical	310	0.5545	float16
35	Census_InternalPrimaryDisplayResolutionHorizontal	261	0.5545	float16
27	Census_ProcessorModelIdentifier	1562	0.4615	float16

For missing values, irrespective of the datatype the columns, they are replaced by the mode. This gives the column value which is already know to the dataset.

3.3. Columns related to the target column

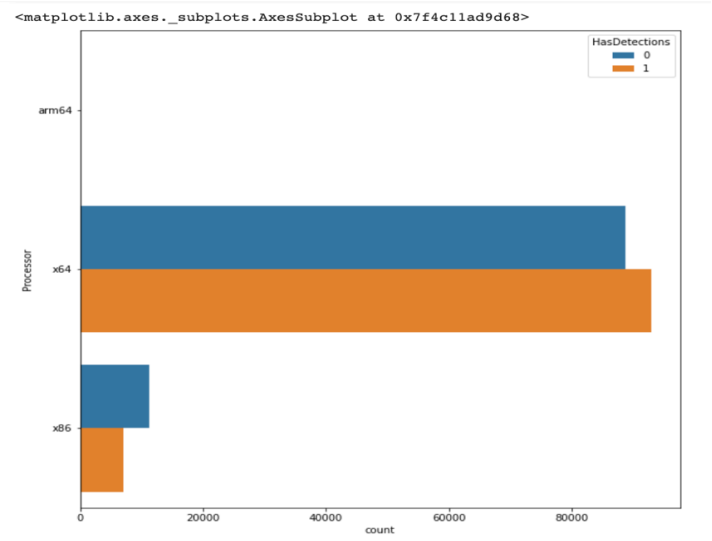
3.3.1. HasDetections

The target column has almost 50% of detection rate. The count of rows that says yes for detection is 1,00,060 and the count of rows that says there was no detection is 99,940.



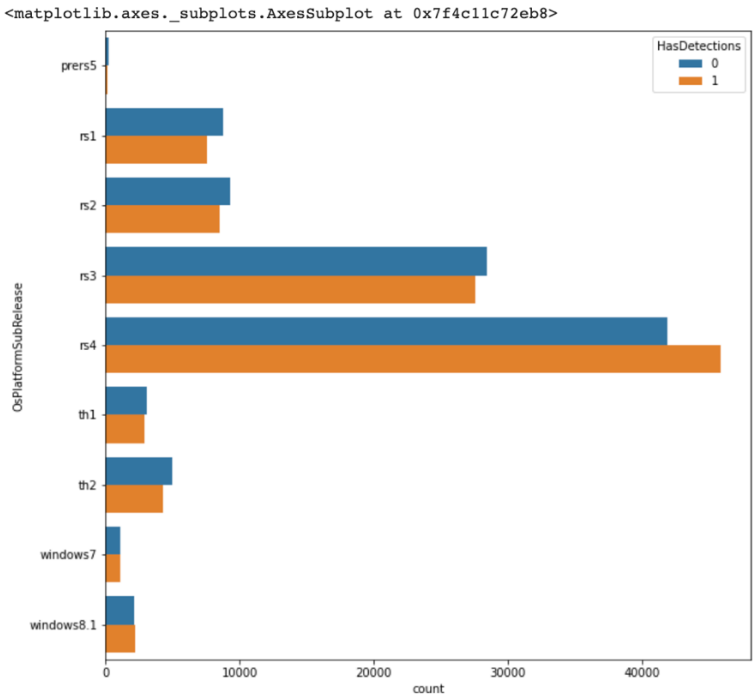
3.3.2. Processor

This column is the process architecture of the installed operating system. The processor has 3 unique values as arm64, x64, and x86. For x64 there were more detection and for x86 there were more no detections. The x64 is the biggest category in this column.



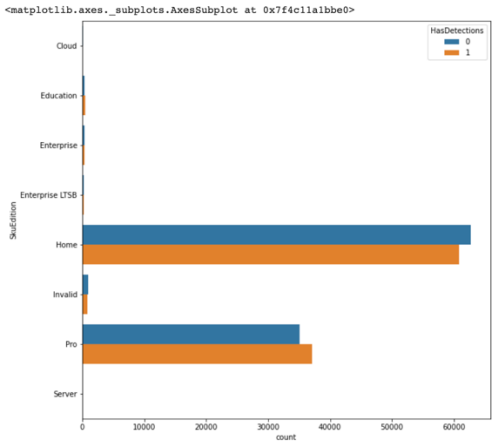
3.3.3. OsPlatformSubRelease

For OS platform sub release, there are 9 unique categories. The highest category is of rs4 and the lowest amount of data is of prers5 category.



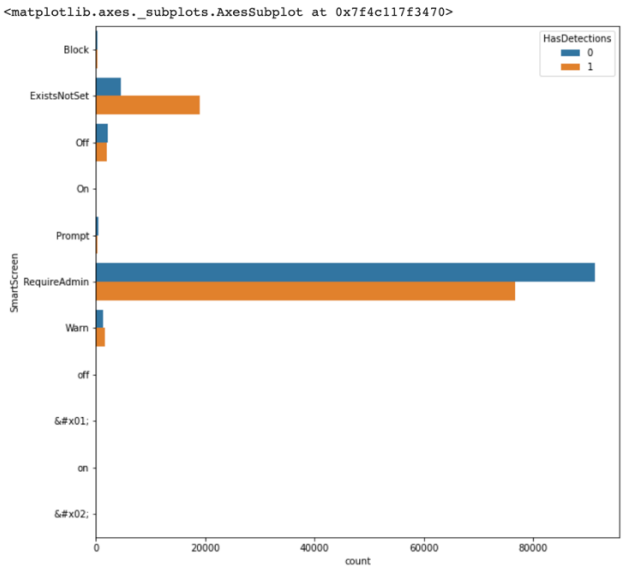
3.3.4. SkuEdition

The goal of this feature is to use the Product Type defined in the MSDN to map to a 'SKU-Edition' name that is useful in population reporting. The home and pro category have the highest detection rate as compared to others.



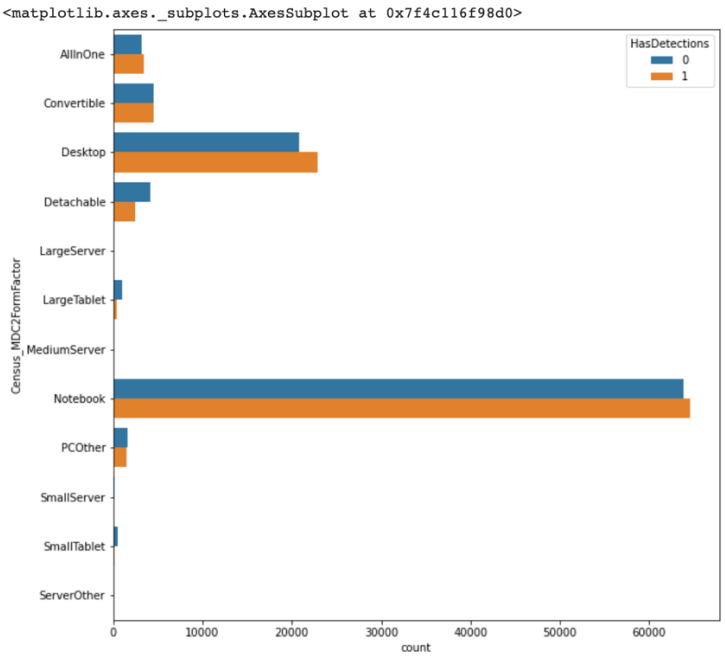
3.3.5. SmartScreen

This is the SmartScreen enabled string value from registry. There were more detection in the ExistsNotSet as compared to no detection. For the category RequireAdmin there were more values as compares to the other categories of the same column.



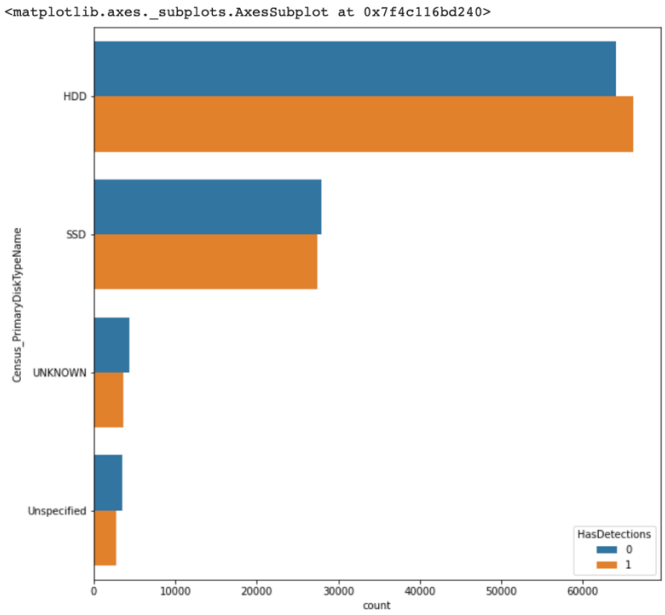
3.3.6. Census_MDC2FormFactor

This column is a grouping based on a combination of Device Census level hardware characteristics. The logic used to define Form Factor is rooted in business and industry standards and aligns with how people think about their device. The notebook category is most popular and has the equal number of detections and no detections.



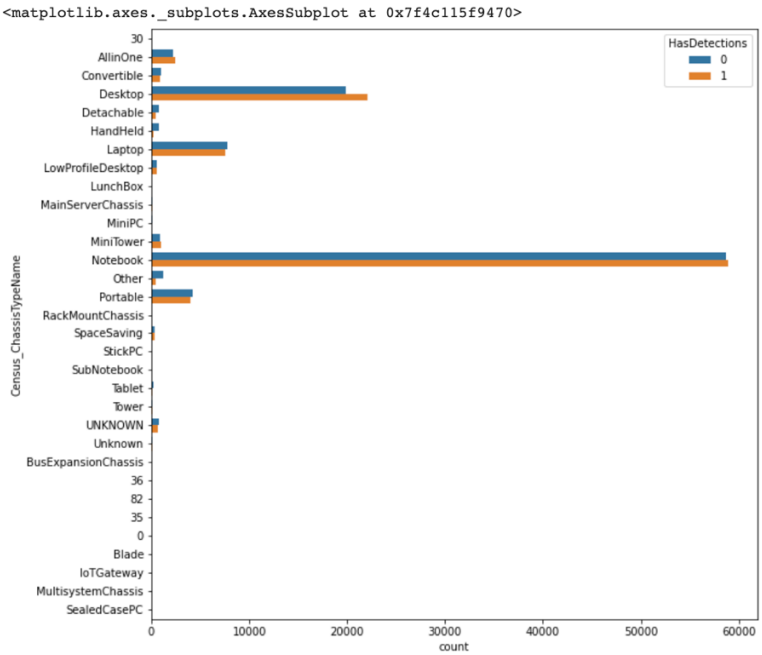
3.3.7. Census_PrimaryDiskTypeName

This column has major two categories - HDD or SSD. The one which has not knows are named as unknown and the one which has no data are specified as unspecified. The HDD is the most popular for the disk type.



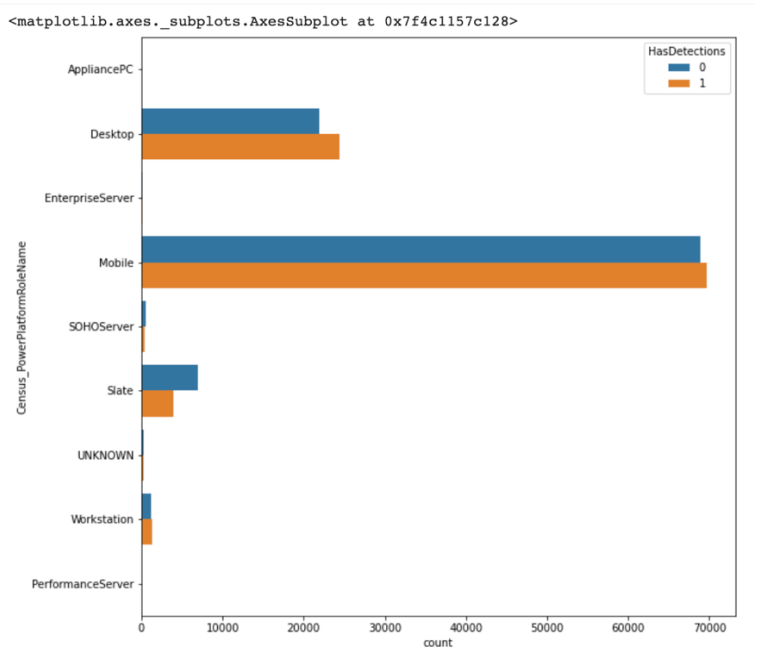
3.3.8. Census_ChassisTypeName

Retrieves a numeric representation of what type of chassis the machine has. A value of 0 means xx. The notebook and desktop are the most popular categories.



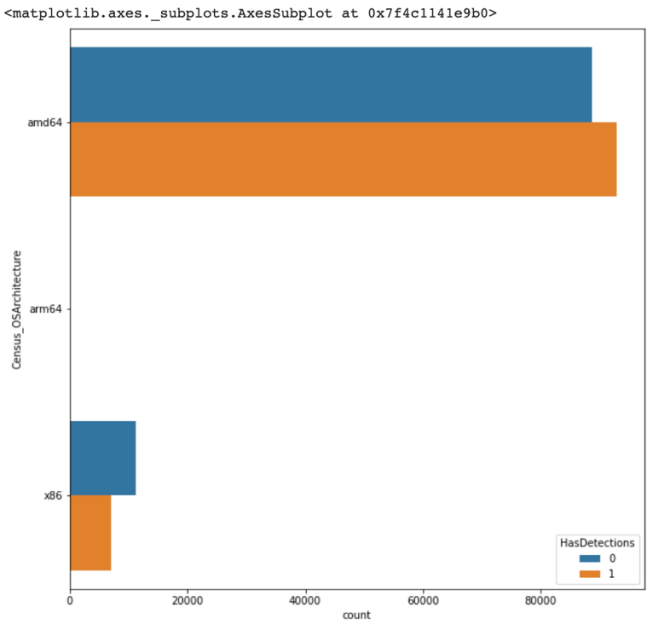
3.3.9. Census_PowerPlatformRoleName

Indicates the OEM preferred power management profile. This value helps identify the basic form factor of the device. It looks like the mobile are the most popular followed by the desktop and state category.



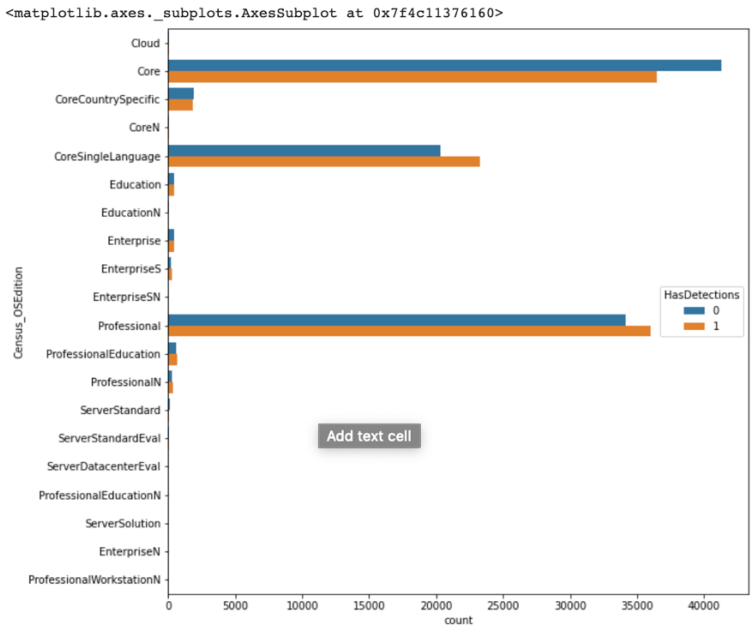
3.3.10. Census_OSArchitecture

This column tells us about the architecture on which the OS is based. This column is derived from OSVersionFull. For this one, amd64 is the most famous category followed by x86.



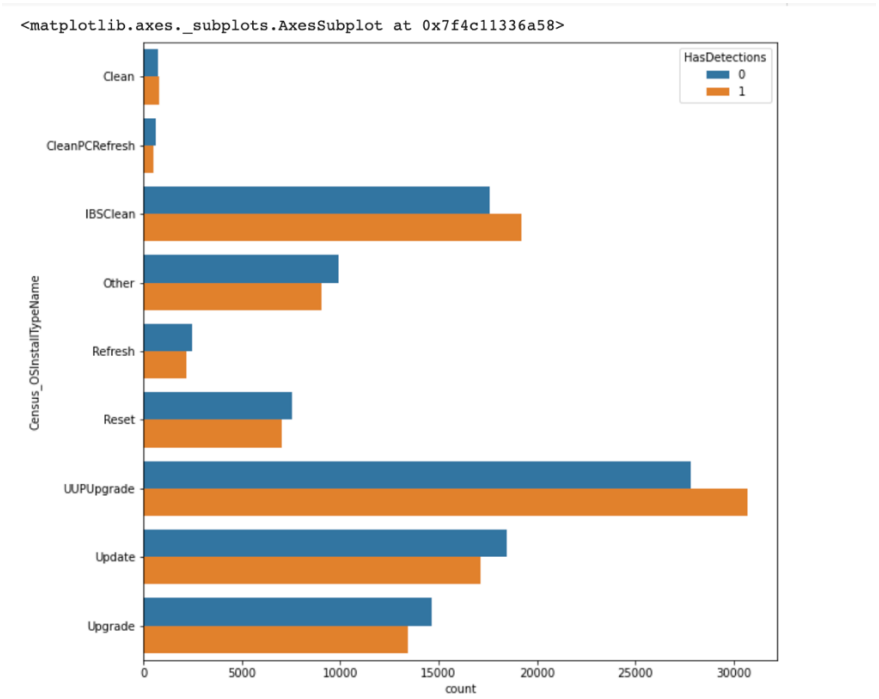
3.3.11. Census_OSEdition

This column specifies the edition of the current OS. For this one, Core, Professional is the most famous category followed by CoreSingleLanguage.



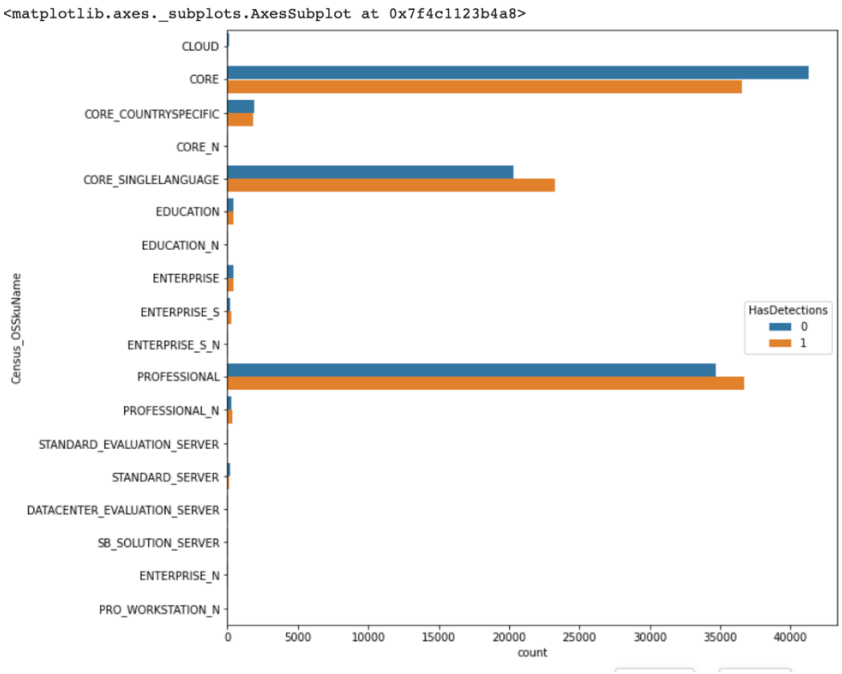
3.3.12. Census_OSInstallTypeName

This column is description of what install was used on the machine. For this one, all the categories are famous expect Clean and CleanPCRefresh.



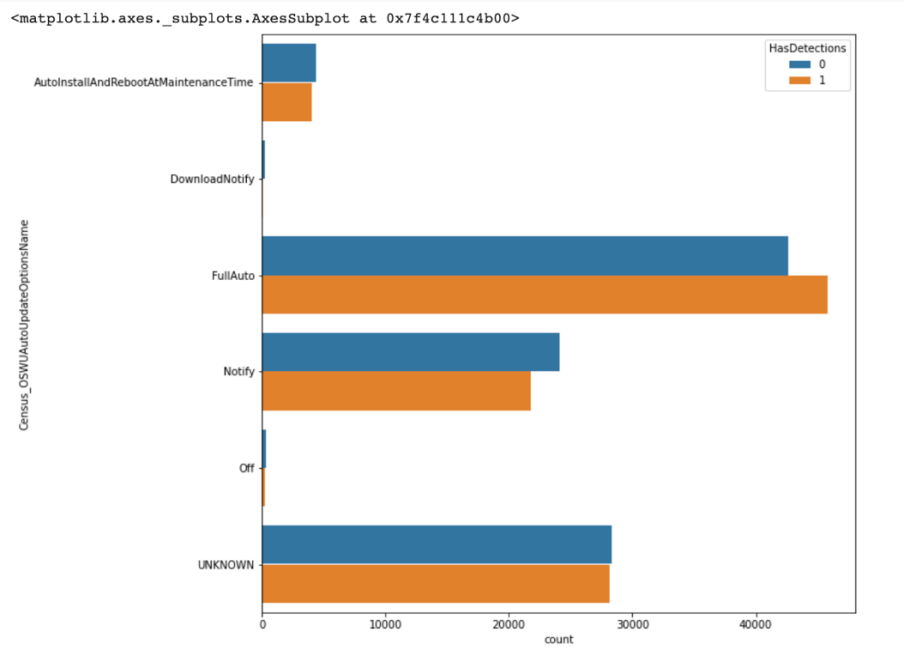
3.3.13. Census_OSSkuName

This column tells us the OS edition friendly name. The most common categories are Core, CoreSingleLanguage, and Professional.



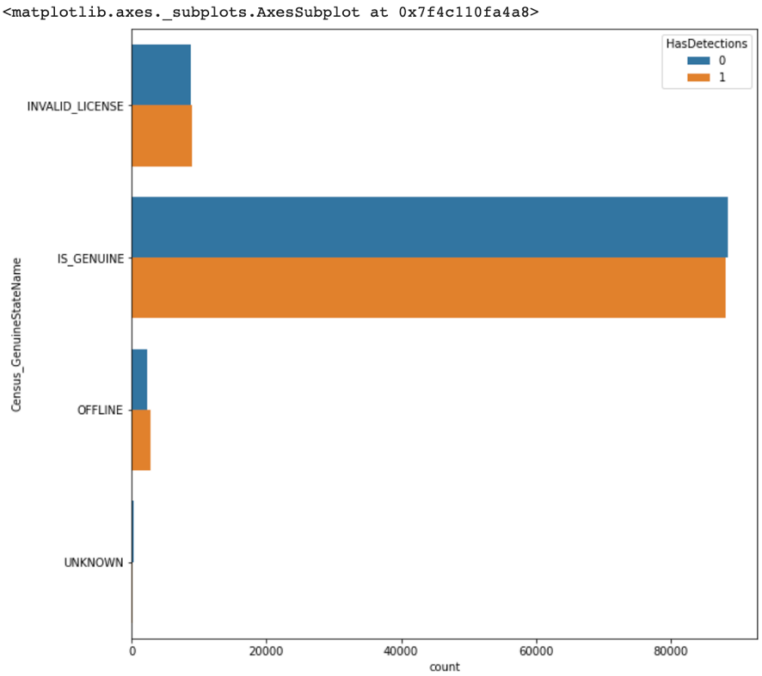
3.3.14. Census_OSWUAutoUpdateOptionsName

This column tells us about the WindowsUpdate auto-update settings on the machine. Most popular category is FullAuto, followed by unkown, and notify.



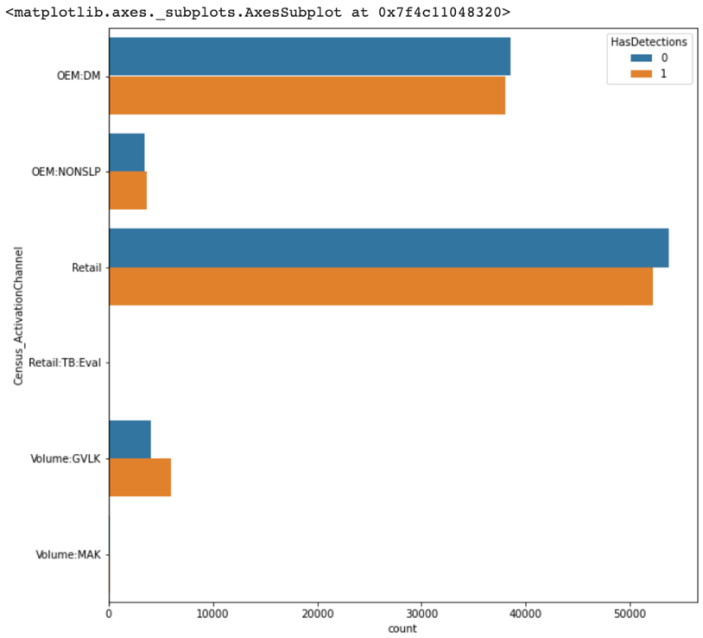
3.3.15. Census_GenuineStateName

This column gives us the friendly name of OSGenuineStateID. The most common category is the genuine category.



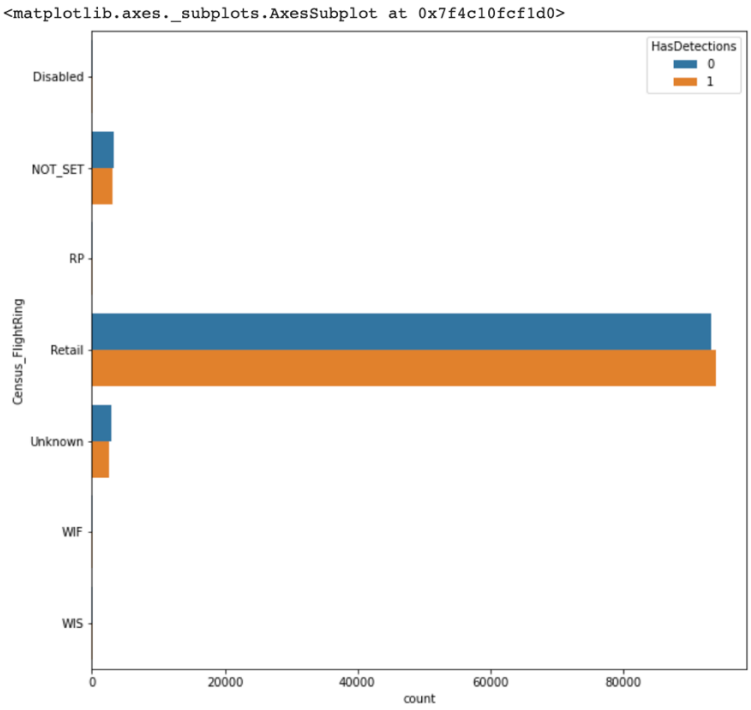
3.3.16. Census_ActivationChannel

This column is the retail license key or Volume license key for a machine. The most popular category is the Retail followed by OEM-DM category.



3.3.17. Census_FlightRing

This column contains the information about the ring that the device user would like to receive flights for. This might be different from the ring of the OS which is currently installed if the user changes the ring after getting a flight from a different ring.



REFERENCES

- https://www.kaggle.com/artgor/is-this-malware-eda-fe-and-lgb-updated#Data-exploration
- https://www.kaggle.com/youhanlee/my-eda-i-want-to-see-all
- https://www.kaggle.com/harmeggels/random-forest-feature-importances/comments
- https://www.kaggle.com/theoviel/load-the-totality-of-the-data
- https://www.kaggle.com/c/microsoft-malware-prediction/data
- https://www.kaggle.com/bogorodvo/lightgbm-baseline-model-using-sparse-matrix
- https://www.kaggle.com/nikkisharma536/beginning-challenge
- https://towardsdatascience.com/better-heatmaps-and-correlation-matrix-plots-in-python-41445d0f2bec
- https://www.kaggle.com/general/74235