MICROSOFT MALWARE DETECTION

Minakshi Mohanty
June 2020

Introduction: Malware is a collective name for any kind of malicious software, designed to infiltrate and attack systems, servers or gain unauthorized access to networks. Some common types of malwares include:

1. **Virus:** Viruses are designed to damage its target computer. They can cause data corruption, reformatting of hard disk, or complete shutdown of system. A virus can replicate itself and spread to other computers through email attachments and files downloaded off internet. They can attach to programs and get executed when the user runs the infected program.
2. **Worm:** Worms are one of the most common forms of malware. They are standalone programs that replicates itself to infect other systems, without requiring the need to be executed by someone. They are often used to execute a payload. A payload is a piece of code that can delete files on a host system, encrypt data for a ransomware attack, steal information, and create botnets.
3. **Trojan Horse:** Trojan horse enters the host system disguised as a normal harmless file or a program that can trick users into downloading and installing it. Once installed, it gives access to cyber criminals to steal data, install more malware, modify files, monitor user activity and conduct denial of service (DoS).
4. **Spyware:** Spywares are designed to track user's browsing and internet activity. They get installed on host's computer, without their knowledge, either by bundling with legitimate software or trojans. They can monitory user activity, collect keystrokes, store login, account information and financial data.
5. **Ransomware:** This is a type of malware that holds user's data captive and demands payments to release data back. They can restrict user access either by encrypting files on hard drive or locking down the system and displaying messages that demand ransom.

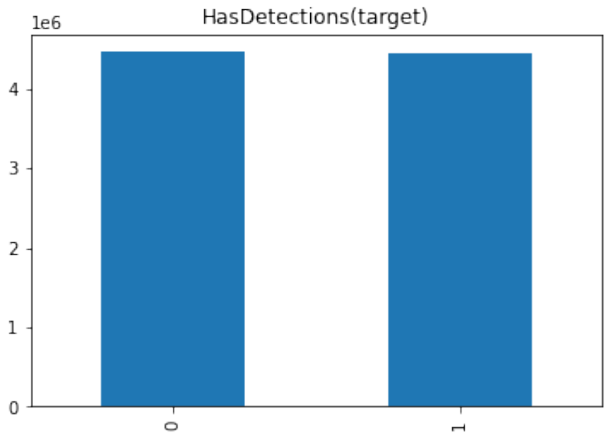
Objective: The goal is to predict the presence of a malware in a Window's system.

Dataset: Microsoft Corporation, Windows Defender ATP Research, Northeastern University College of Computer and Information Science, and Georgia Tech Institute for Information Security & Privacy provides the dataset, through their competition hosted on Kaggle. The link to the competition is: (https://www.kaggle.com/c/microsoft-malware-prediction/data).

Data Wrangling:

This dataset has been put together by combining heartbeat and threat reports collected by Microsoft's endpoint protection solutions, Windows Defender. The data contains properties of the machine and malware infections. There are 82 features in this dataset. The target variable is "HasDetections". It indicates that if malware was detected on the machine. Each row in the dataset corresponds to a machine that can be uniquely identified by its "MachineIdentifier". In this section, we explore through every feature and prepare the dataset for modeling. The description of every column is written below.

The dataset contains 8921483 entries. The target variable, "HasDetections" has 2 values. 0 corresponds to no malware detection and 1 corresponds to malware detection. There are 4462591entries in class 0 and 4458892 entries in class 1. The dataset is balanced.



Histogram of "HasDetection"

Since the dataset has 82 features, it is crucial to reduce the some of the variables. The table below shows the top ten variables with missing data.

	Total	Percent
PuaMode	8919174	99.974119
Census_ProcessorClass	8884852	99.589407
DefaultBrowsersIdentifier	8488045	95.141637
Census_IsFlightingInternal	7408759	83.044030
Census_InternalBatteryType	6338429	71.046809
Census_ThresholdOptIn	5667325	63.524472
Census_IsWIMBootEnabled	5659703	63.439038
SmartScreen	3177011	35.610795
OrganizationIdentifier	2751518	30.841487
SMode	537759	6.027686

Variables with percentage of missing data

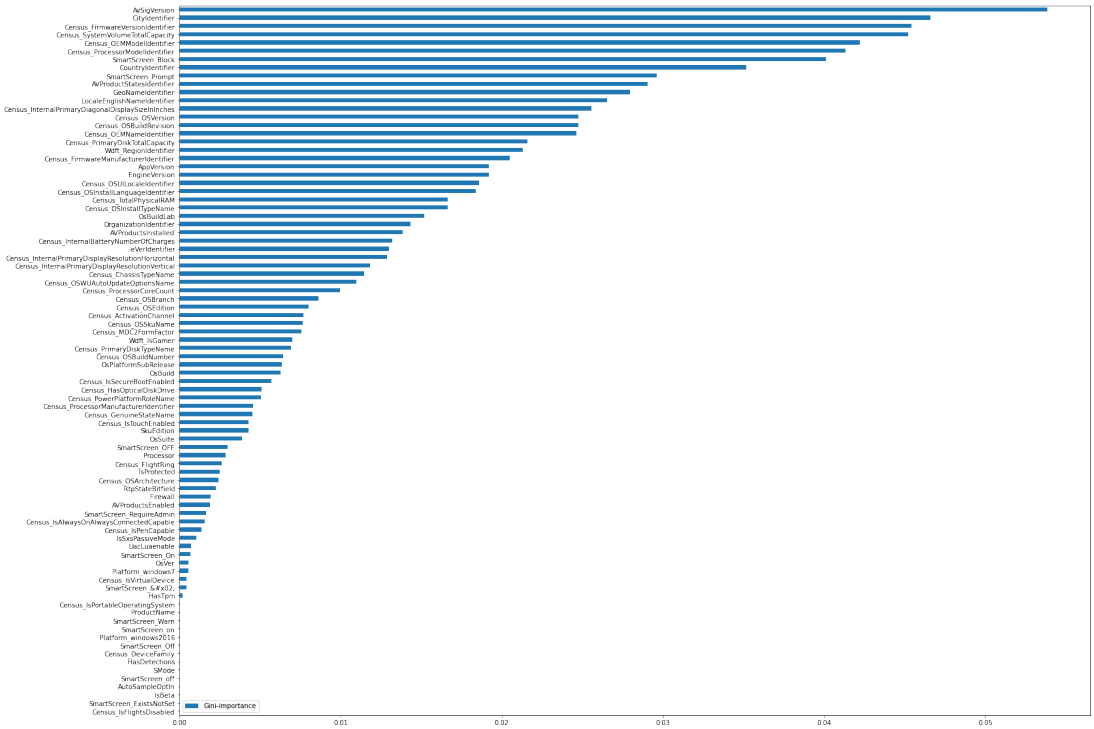
PuaMode, Census_ProcessorClass, DefaultBrowsersIdentifier, Census_IsFlightingInternal and Census_InternalBatteryType have over 60% missing data. These variables can be removed from the analysis.

After a careful analysis of each and every variable, it can be seen that all the variables are of category datatype. Most of the variables have very high cardinality. Below, is an image showing the number of categories in some variables.

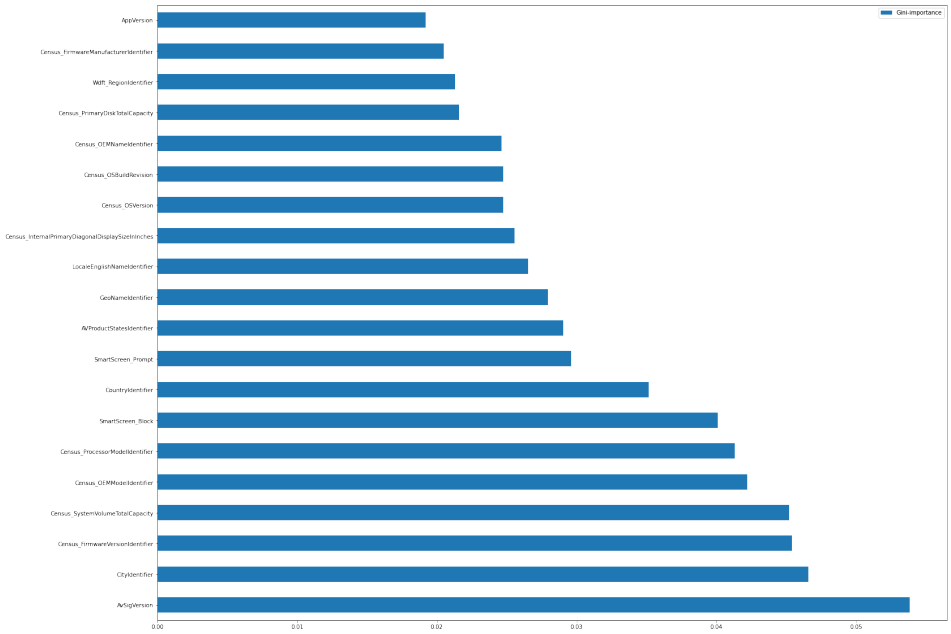
```
ProductName : 6 categories
EngineVersion : 70 categories
AppVersion : 110 categories
AvSigVersion : 8531 categories
Platform : 4 categories
Processor : 3 categories
OsVer : 58 categories
OsPlatformSubRelease : 9 categories
OsBuildLab : 664 categories
SkuEdition : 8 categories
SmartScreen : 22 categories
Census_MDC2FormFactor : 13 categories
Census_DeviceFamily : 3 categories
Census_PrimaryDiskTypeName : 5 categories
Census_ChassisTypeName : 53 categories
Census_PowerPlatformRoleName : 11 categories
Census_OSVersion : 469 categories
Census_OSArchitecture : 3 categories
Census_OSBranch : 32 categories
Census_OSEdition : 33 categories
Census_OSSkuName : 30 categories
Census_OSInstallTypeName : 9 categories
Census_OSWUAutoUpdateOptionsName : 6 categories
Census_GenuineStateName : 5 categories
Census_ActivationChannel : 6 categories
Census_FlightRing : 10 categories
```

For variables with more that 2 unique categories, we use frequency encoding. For others, one-hot encoding is done. The prepared data is saved to a csv file for easier access. Due to hardware limitations, we use on a part of the entire dataset for our modelling.

Feature Selection: To select the best features, we run the data through a Random Forest Classifier and plot the feature importance.



Feature Importance Plot



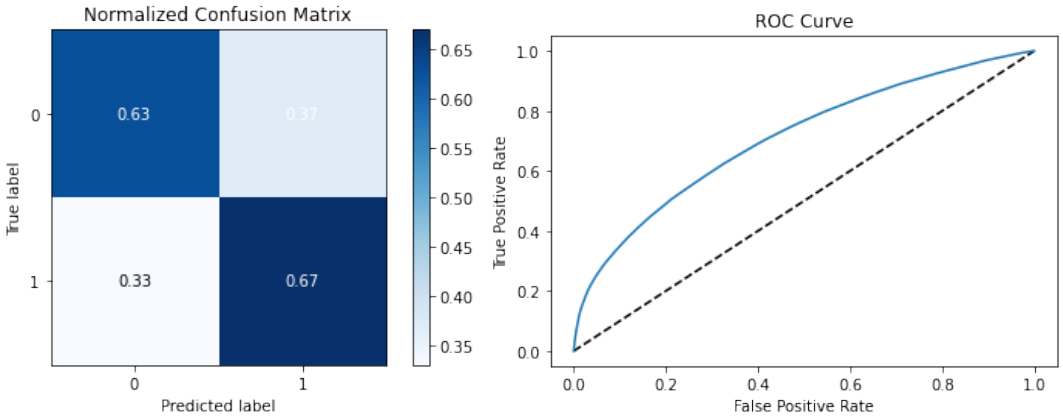
Top 20 features

	Gini-importance
AvSigVersion	0.053834
CityIdentifier	0.046586
Census_FirmwareVersionIdentifier	0.045409
Census_SystemVolumeTotalCapacity	0.045213
Census_OEMModelIdentifier	0.042214
Census_ProcessorModelIdentifier	0.041319
SmartScreen_Block	0.040108
CountryIdentifier	0.035146
SmartScreen_Prompt	0.029595
AVProductStatesIdentifier	0.029032
GeoNameIdentifier	0.027955
LocaleEnglishNameIdentifier	0.026538
Census_InternalPrimaryDiagonalDisplaySizeInInches	0.025540
Census_OSVersion	0.024762
Census_OSBuildRevision	0.024733
Census_OEMNameIdentifier	0.024640
Census_PrimaryDiskTotalCapacity	0.021590
Wdft_RegionIdentifier	0.021293
Census_FirmwareManufacturerIdentifier	0.020493
AppVersion	0.019204

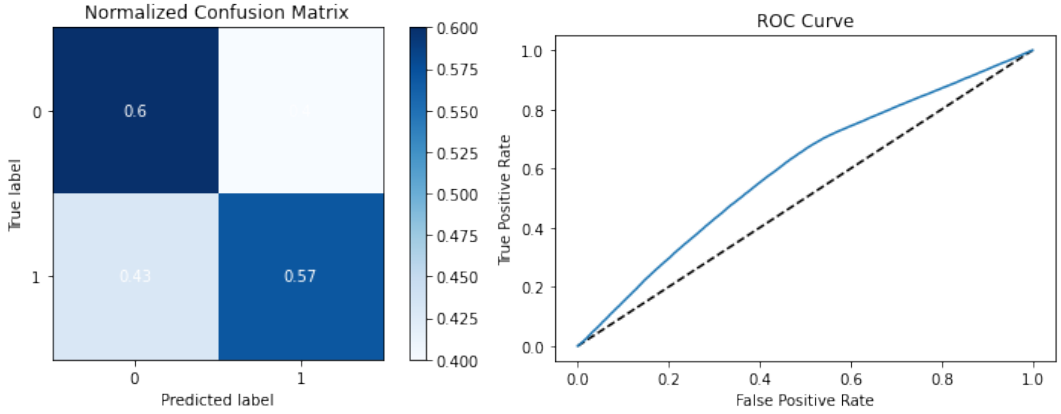
Machine learning Modeling and Optimization:

Using the top 20 features, we run the data through four different classifiers and compare their performance.

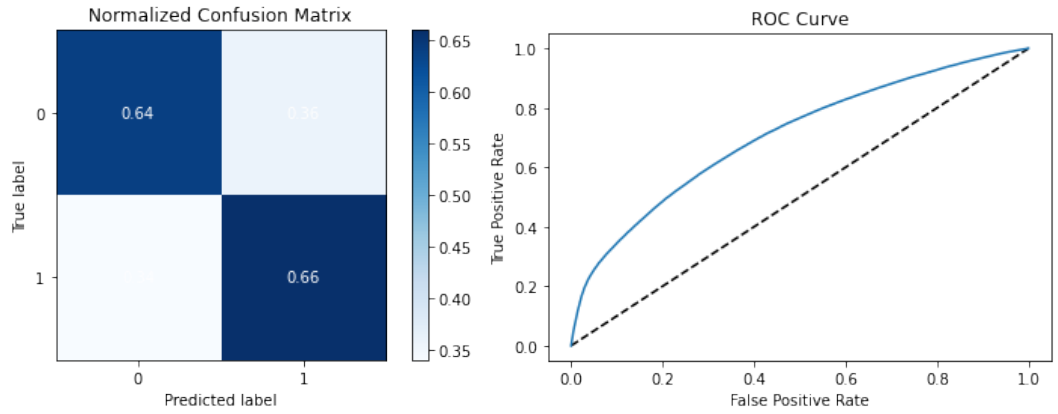
1) **Random Forest Classifier:** A random forest estimator fits a number of decision tree classifiers on different sub-samples of the dataset. It uses averaging to improve predictive accuracy and control overfitting[(2)].



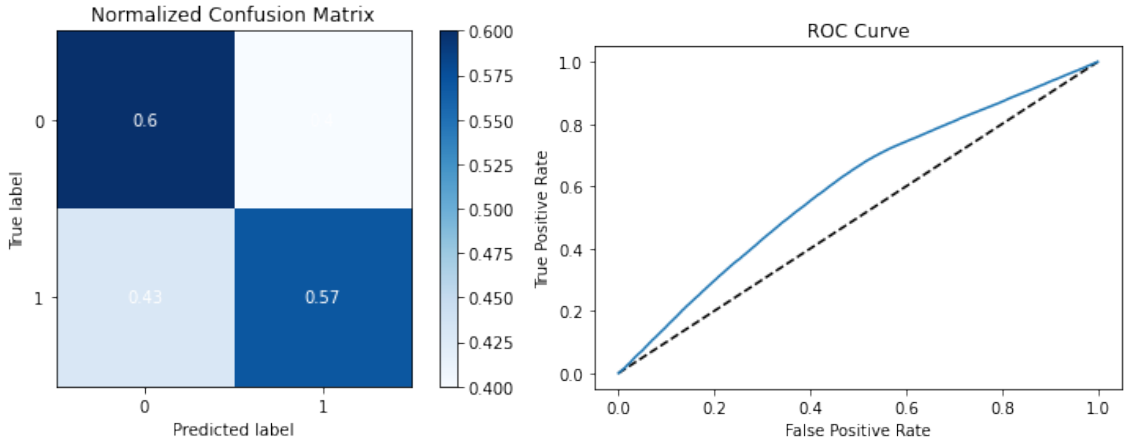
2) **Logistic Regression:**



3) **AdaBoost Classifier with Decision Trees as base estimator**



4) **AdaBoost Classifier with Logistic Regression as base estimator**



5)

Model Evaluation:

In this section, we evaluate and compare the performance of the model on the test dataset. The models were built on training dataset.

Model	**Accuracy**	**AUC**	**Precision (weighted avg)**	**Recall (weighted avg)**
Random Forest Classifier	0.65	0.70	0.65	0.65
Logistic Regression	0.58	0.59	0.61	0.58
AdaBoost Classifier (Decision Tree)	0.65	0.70	0.65	0.65
AdaBoost Classifier (Logistic Regression)	0.58	0.59	0.61	0.58

Random forest classifier and AdaBoost classifier with Decision Tree as base estimator have the same performance. Logistic regression and AdaBoost classifier with Logistic Regression as base estimator have same performance, but lower than that of random forest and AdaBoost.

Conclusion: The models showed average performance. For future work, a combination of more feature engineering methods can be used to select the best features. Other boosting models such as XGBoost, Gradient boost classifiers can also be tested. Due to limitations in hardware, only a part of data was used for modeling. This can be addressed in future revisions.

References:

1. Coverage Image courtesy: https://www.hellotech.com/blog/how-to-remove-malware-from-windows-10
2. Scikit-learn Random Forest Classifier : https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.RandomForestClassifier.html

Appendix A:

Unavailable or self-documenting column names are marked with an "NA".

Variable Name	Description
MachineIdentifier	Individual machine ID
ProductName	Defender state information e.g. win8defender
EngineVersion	Defender state information e.g. 1.1.12603.0
AppVersion	Defender state information e.g. 4.9.10586.0
AvSigVersion	Defender state information e.g. 1.217.1014.0
IsBeta	Defender state information e.g. false
RtpStateBitfield	NA
IsSxsPassiveMode	NA
DefaultBrowsersIdentifier	ID for the machine's default browser
AVProductStatesIdentifier	ID for the specific configuration of a user's antivirus software
AVProductsInstalled	NA
AVProductsEnabled	NA
HasTpm	True if machine has tpm
CountryIdentifier	ID for the country the machine is located in
CityIdentifier	ID for the city the machine is located in
OrganizationIdentifier	ID for the organization the machine belongs in, organization ID is mapped to both specific companies and broad industries
GeoNameIdentifier	ID for the geographic region a machine is located in
LocaleEnglishNameIdentifier	English name of Locale ID of the current user
Platform	Calculates platform name (of OS related properties and processor property)
Processor	This is the process architecture of the installed operating system
OsVer	Version of the current operating system
OsBuild	Build of the current operating system
OsSuite	Product suite mask for the current operating system.
OsPlatformSubRelease	Returns the OS Platform sub-release (Windows Vista, Windows 7, Windows 8, TH1, TH2)
OsBuildLab	Build lab that generated the current OS. Example: 9600.17630.amd64fre.winblue_r7.150109-2022
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 valid Product Type are defined in %sdxroot%\data\windowseditions.xml. This API has been used since Vista and Server 2008, so there are many Product Types that do not apply to Windows 10. The 'SKU- Edition' is a string

	value that is in one of three classes of results. The design must hand each class.
IsProtected	This is a calculated field derived from the Spynet Report's AV Products field. Returns: a. TRUE if there is at least one active and up to date antivirus product running on this machine. b. FALSE if there is no active AV product on this machine, or if the AV is active, but is not receiving the latest updates. c. null if there are no Anti Virus Products in the report. Returns: Whether a machine is protected.
AutoSampleOptIn	This is the SubmitSamplesConsent value passed in from the service, available on CAMP 9+
PuaMode	Pua Enabled mode from the service
SMode	This field is set to true when the device is known to be in 'S Mode', as in, Windows 10 S mode, where only Microsoft Store apps can be installed
IeVerIdentifier	NA
SmartScreen	This is the SmartScreen enabled string value from registry. This is obtained by checking in order, HKLM\SOFTWARE\Policies\Microsoft\Windows\System\SmartScreenEnabled and HKLM\SOFTWARE\Microsoft\Windows\CurrentVersion\Explorer\SmartScreenEnabled. If the value exists but is blank, the value "ExistsNotSet" is sent in telemetry.
Firewall	This attribute is true (1) for Windows 8.1 and above if windows firewall is enabled, as reported by the service.
UacLuaenable	This attribute reports whether or not the "administrator in Admin Approval Mode" user type is disabled or enabled in UAC. The value reported is obtained by reading the regkey HKLM\SOFTWARE\Microsoft\Windows\CurrentVersion\Policies\System\EnableLUA.
Census_MDC2FormFactor	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. (Examples: Smartphone, Small Tablet, All in One, Convertible...)
Census_DeviceFamily	AKA DeviceClass. Indicates the type of device that an edition of the OS is intended for. Example values: Windows.Desktop, Windows.Mobile, and iOS.Phone
Census_OEMNameIdentifier	NA
Census_OEMModelIdentifier	NA
Census_ProcessorCoreCount	Number of logical cores in the processor
Census_ProcessorManufacturerIdentifier	NA
Census_ProcessorModelIdentifier	NA
Census_ProcessorClass	A classification of processors into high/medium/low. Initially used for Pricing Level SKU. No longer maintained and updated
Census_PrimaryDiskTotalCapacity	Amount of disk space on primary disk of the machine in MB

Census_PrimaryDiskTypeName	Friendly name of Primary Disk Type HDD or SSD
Census_SystemVolumeTotalCapacity	The size of the partition that the System volume is installed on in MB
Census_HasOpticalDiskDrive	True indicates that the machine has an optical disk drive (CD/DVD)
Census_TotalPhysicalRAM	Retrieves the physical RAM in MB
Census_ChassisTypeName	Retrieves a numeric representation of what type of chassis the machine has. A value of 0 means xx
Census_InternalPrimaryDiagonalDisplaySizeInInches	Retrieves the physical diagonal length in inches of the primary display
Census_InternalPrimaryDisplayResolutionHorizontal	Retrieves the number of pixels in the horizontal direction of the internal display.
Census_InternalPrimaryDisplayResolutionVertical	Retrieves the number of pixels in the vertical direction of the internal display
Census_PowerPlatformRoleName	Indicates the OEM preferred power management profile. This value helps identify the basic form factor of the device
Census_InternalBatteryType	NA
Census_InternalBatteryNumberOfCharges	NA
Census_OSVersion	Numeric OS version Example-10.0.10130.0
Census_OSArchitecture	Architecture on which the OS is based. Derived from OSVersionFull. Example- amd64
Census_OSBranch	Branch of the OS extracted from the OsVersionFull. Example- OsBranch = fbl_partner_eeap where OsVersion = 6.4.9813.0.amd64fre.fbl_partner_eeap.140810-5-5
Census_OSBuildNumber	OS Build number extracted from the OsVersionFull. Example OsBuildNumber = 10512 or 10240
Census_OSBuildRevision	OS Build revision extracted from the OsVersionFull. Example OsBuildRevision = 1000 or 16458
Census_OSEdition	Edition of the current OS. Sourced from HKLM\Software\Microsoft\Windows NT\CurrentVersion@EditionID in registry. Example: Enterprise
Census_OSSkuName	OS edition friendly name (currently Windows only)
Census_OSInstallTypeName	Friendly description of what install was used on the machine i.e. clean
Census_OSInstallLanguageIdentifier	NA
Census_OSUILocaleIdentifier	NA
Census_OSWUAutoUpdateOptionsName	Friendly name of the WindowsUpdate auto-update settings on the machine.
Census_IsPortableOperatingSystem	Indicates whether OS is booted up and running via Windows to Go on a USB stick.
Census_GenuineStateName	Friendly name of OSGenuineStateID. 0 = Genuine
Census_ActivationChannel	Retail license key or Volume license key for a machine.

Census_IsFlightingInternal	NA
Census_IsFlightsDisabled	Indicates if the machine is participating in flighting.
Census_FlightRing	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.
Census_ThresholdOptIn	NA
Census_FirmwareManufacturerIdentifier	NA
Census_FirmwareVersionIdentifier	NA
Census_IsSecureBootEnabled	Indicates if Secure Boot mode is enabled.
Census_IsWIMBootEnabled	NA
Census_IsVirtualDevice	Identifies a Virtual Machine (machine learning model)
Census_IsTouchEnabled	Is this a touch device ?
Census_IsPenCapable	Is the device capable of pen input ?
Census_IsAlwaysOnAlwaysConnectedCapable	Retreives information about whether the battery enables the device to be AlwaysOnAlwaysConnected .
Wdft_IsGamer	Indicates whether the device is a gamer device or not based on its hardware combination.
Wdft_RegionIdentifier	NA