Section 1: Setting up drive, path, packages and loading the data

```
## To load up drive

%cd drive/MyDrive/CSE_519_assignment/
```

```
/content/drive/MyDrive/CSE_519_assignment
```

```
## Import statements

import pandas as pd
import seaborn as sns
import os
from sklearn import metrics
from sklearn.linear_model import LogisticRegression
from sklearn.model_selection import train_test_split
from sklearn.ensemble import RandomForestClassifier

root_dir = os.getcwd()
pd.set_option('display.max_columns', None)
pd.set_option('display.max_rows', None)
```

```
## Added use_cols and dtypes to use while loading data

use_cols = ["MachineIdentifier", "SmartScreen", "AVProductsInstalled", "AppVersion'
           "EngineVersion", "AVProductStatesIdentifier", "Census_OSVersion", "Censu
           "RtpStateBitfield", "Census_ProcessorModelIdentifier", "Census_PrimaryDi
            "Census_InternalPrimaryDiagonalDisplaySizeInInches", "Wdft_RegionIdenti
           "AvSigVersion", "IeVerIdentifier", "IsProtected", "Census_InternalPrimar
            "Census_OSWUAutoUpdateOptionsName", "Census_OSEdition", "Census_Genuine
           "Census_OEMNameIdentifier", "Census_MDC2FormFactor", "Census_FirmwareMar
            "Census_OSBuildNumber", "Census_IsPenCapable", "Census_IsTouchEnabled",
            "Census_SystemVolumeTotalCapacity", "Census_PrimaryDiskTotalCapacity",
           ]
dtypes = {
        'MachineIdentifier':                                    'category',
        'ProductName':                                          'category',
        'EngineVersion':                                        'category',
        'AppVersion':                                           'category',
        'AvSigVersion':                                         'category',
        'IsBeta':                                               'int8',
        'RtpStateBitfield':                                     'float16',
        'IsSxsPassiveMode':                                     'int8',
        'DefaultBrowsersIdentifier':                            'float16',
        'AVProductStatesIdentifier':                            'float32',
        'AVProductsInstalled':                                  'float16',
```

✓ 0s completed at 1:32 AM

```
    'Platform':                                             'category',
    'Processor':                                            'category',
    'OsVer':                                                'category',
    'OsBuild':                                              'int16',
    'OsSuite':                                              'int16',
    'OsPlatformSubRelease':                                 'category',
    'OsBuildLab':                                           'category',
    'SkuEdition':                                           'category',
    'IsProtected':                                          'float16',
    'AutoSampleOptIn':                                      'int8',
    'PuaMode':                                              'category',
    'SMode':                                                'float16',
    'IeVerIdentifier':                                      'float16',
    'SmartScreen':                                          'category',
    'Firewall':                                             'float16',
    'UacLuaenable':                                         'float32',
    'Census_MDC2FormFactor':                                'category',
    'Census_DeviceFamily':                                  'category',
    'Census_OEMNameIdentifier':                             'float16',
    'Census_OEMModelIdentifier':                            'float32',
    'Census_ProcessorCoreCount':                            'float16',
    'Census_ProcessorManufacturerIdentifier':               'float16',
    'Census_ProcessorModelIdentifier':                      'float16',
    'Census_ProcessorClass':                                'category',
    'Census_PrimaryDiskTotalCapacity':                      'float32',
    'Census_PrimaryDiskTypeName':                           'category',
    'Census_SystemVolumeTotalCapacity':                     'float32',
    'Census_HasOpticalDiskDrive':                           'int8',
    'Census_TotalPhysicalRAM':                              'float32',
    'Census_ChassisTypeName':                               'category',
    'Census_InternalPrimaryDiagonalDisplaySizeInInches':    'float16',
    'Census_InternalPrimaryDisplayResolutionHorizontal':    'float16',
    'Census_InternalPrimaryDisplayResolutionVertical':      'float16',
    'Census_PowerPlatformRoleName':                         'category',
    'Census_InternalBatteryType':                           'category',
    'Census_InternalBatteryNumberOfCharges':                'float32',
    'Census_OSVersion':                                     'category',
    'Census_OSArchitecture':                                'category',
    'Census_OSBranch':                                      'category',
    'Census_OSBuildNumber':                                 'int16',
    'Census_OSBuildRevision':                               'int32',
    'Census_OSEdition':                                     'category',
    'Census_OSSkuName':                                     'category',
    'Census_OSInstallTypeName':                             'category'
```

```
        'Census_IsSecureBootEnabled':                           'int8',
        'Census_IsWIMBootEnabled':                              'float16',
        'Census_IsVirtualDevice':                               'float16',
        'Census_IsTouchEnabled':                                'int8',
        'Census_IsPenCapable':                                  'int8',
        'Census_IsAlwaysOnAlwaysConnectedCapable':              'float16',
        'Wdft_IsGamer':                                         'float16',
        'Wdft_RegionIdentifier':                                'float16'
        }
```

```
## Loading primary data

df = pd.read_csv(root_dir + "/train.csv", usecols=use_cols, dtype=dtypes)
```

```
##·Print·data·to·check·values

df.head()
```

	MachineIdentifier	EngineVersion	AppVersion	AvSigVers
0	0000028988387b115f69f31a3bf04f09	1.1.15100.1	4.18.1807.18075	1.273.17
1	000007535c3f730efa9ea0b7ef1bd645	1.1.14600.4	4.13.17134.1	1.263.
2	000007905a28d863f6d0d597892cd692	1.1.15100.1	4.18.1807.18075	1.273.13
3	00000b11598a75ea8ba1beea8459149f	1.1.15100.1	4.18.1807.18075	1.273.15
4	000014a5f00daa18e76b81417eeb99fc	1.1.15100.1	4.18.1807.18075	1.273.13

```
## Print data to check shape


df.shape
```

```
    (8921483, 39)
```

75%	7.0	5.344700e+04	2.0
max	35.0	7.050700e+04	7.0

Section 2: Measure of Power (Q2a & 2b)

```
## So, PrimaryDisktype can denote a slower hdd or a faster ssd, so checking which c

df["Census_PrimaryDiskTypeName"].unique()
```

```
     ['HDD', 'SSD', 'UNKNOWN', 'Unspecified', NaN]
     Categories (4, object): ['HDD', 'SSD', 'UNKNOWN', 'Unspecified']
```

```
## Giving 0.5 score if the type is HDD, 1 if its SSD and 0 if it's unkown.

def change_Census_PrimaryDiskTypeName(name):
    if name == "HDD":
        return 0.5
    elif name == "SSD":
        return 1
    else:
        return 0

df["Census_PrimaryDiskTypeName"] = df["Census_PrimaryDiskTypeName"].apply(change_Ce
```

```
df[['Census_SystemVolumeTotalCapacity', 'Census_TotalPhysicalRAM','Census_InternalP
```

```
        result[feature_name] = (df[feature_name] - min_value) / (max_value - min_valu
    return result
df = normalize_feature(df, column_names_to_normalize)
```

```
## Calculating the power by random assignment of weights to above features.

df["power"] = (0.2*df["Census_SystemVolumeTotalCapacity"] + 0.4*df["Census_TotalPhy
```

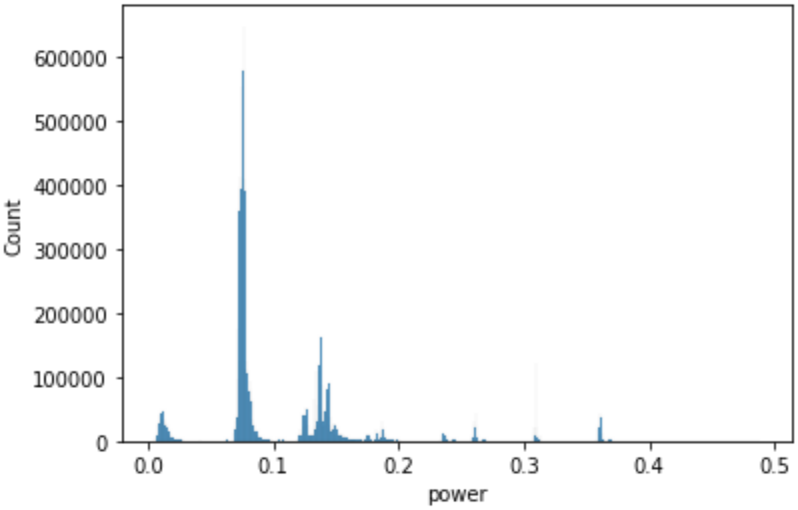
```
df["power"].head()
```

```
    0    0.077567
    1    0.074396
    2    0.141104
    3    0.014694
    4    0.074712
    Name: power, dtype: float64
```

```
## Plotting power and the count of each level.

sns.histplot(df, x='power')
```

```
    <matplotlib.axes._subplots.AxesSubplot at 0x7fb068ab0cd0>
```

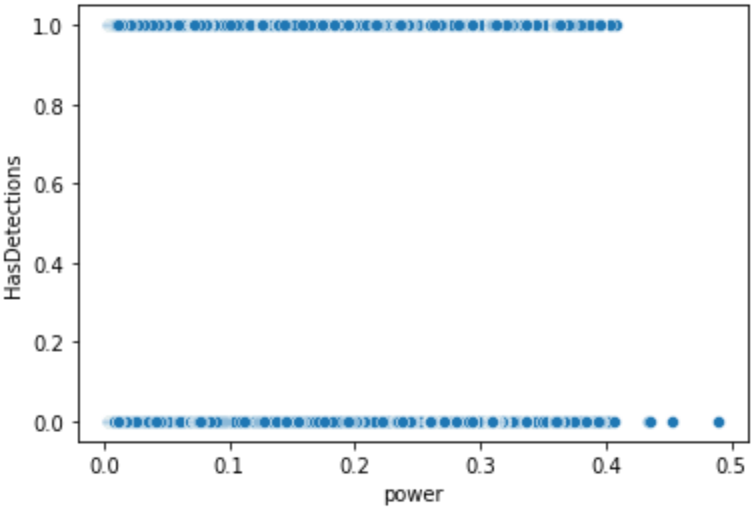


max 4.889679e-01

```
## Power vs malware detection plot -

sns.scatterplot(data=df, y="HasDetections", x="power")
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x7fb0446e1190>
```



```
Census_OSBuildNumber  HasDetections
7600                  0                1.000000
7601                  1                0.571429
                      0                0.428571
9200                  0                0.500000
                      1                0.500000
Name: HasDetections, dtype: float64
```

```
os_revision_number_group.head()
```

```
Census_OSBuildRevision  HasDetections
0                       1                0.514819
                        0                0.485181
```

Section 4

```
## Checking whether the number of AVDs installed
```