ASSIGNMENT 3 AI_CS

Hadeer Mohamed

UOTTAWA 300327273

Part I:

Exploratory Data Analysis (EDA):

- **Data Information:**

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 268074 entries, 0 to 268073
Data columns (total 16 columns):
 #   Column            Non-Null Count   Dtype
---  ------            --------------   -----
 0   timestamp         268074 non-null  object
 1   FQDN_count        268074 non-null  int64
 2   subdomain_length  268074 non-null  int64
 3   upper             268074 non-null  int64
 4   lower             268074 non-null  int64
 5   numeric           268074 non-null  int64
 6   entropy           268074 non-null  float64
 7   special           268074 non-null  int64
 8   labels            268074 non-null  int64
 9   labels_max        268074 non-null  int64
 10  labels_average    268074 non-null  float64
 11  longest_word      268066 non-null  object
 12  sld               268074 non-null  object
 13  len               268074 non-null  int64
 14  subdomain         268074 non-null  int64
 15  Target Attack     268074 non-null  int64
dtypes: float64(2), int64(11), object(3)
memory usage: 32.7+ MB
```

- **Data Description**

	FQDN_count	subdomain_length	upper	lower	numeric	entropy	special	labels	labels_max	labels_av
count	268074.000000	268074.000000	268074.000000	268074.000000	268074.000000	268074.000000	268074.000000	268074.000000	268074.000000	268074.00
mean	22.286596	6.059021	0.845420	10.410014	6.497586	2.485735	4.533577	4.788823	8.252233	4.80
std	6.001205	3.899505	4.941929	3.207725	4.499866	0.407709	2.187683	1.803256	4.415355	4.57
min	2.000000	0.000000	0.000000	0.000000	0.000000	0.219195	0.000000	1.000000	2.000000	2.00
25%	18.000000	3.000000	0.000000	10.000000	0.000000	2.054029	2.000000	3.000000	7.000000	3.16
50%	24.000000	7.000000	0.000000	10.000000	8.000000	2.570417	6.000000	6.000000	7.000000	3.66
75%	27.000000	10.000000	0.000000	10.000000	10.000000	2.767195	6.000000	6.000000	7.000000	4.00
max	36.000000	23.000000	32.000000	34.000000	12.000000	4.216847	7.000000	7.000000	32.000000	32.00

- **String Columns:**

```
S_dataset['longest_word'].value_counts()
```

```
2          109981
4           70188
N            4498
C            2969
9            1906
           ...
yaa             1
queue           1
kit             1
airdrop         1
mal             1
Name: longest_word, Length: 6224, dtype: int64
```

```
S_dataset['sld'].value_counts()
```

```
192                                 109517
224                                  70188
FHEPFCELEHFCEPFFFACACACACACACABN      4498
DESKTOP-3JF04TC                       1961
239                                   1906
                                      ...
freesgift                                1
secureserver                             1
airdropalert                             1
queue-it                                 1
lahemal                                  1
Name: sld, Length: 11112, dtype: int64
```

Data cleaning:

- **String columns is converted to integers**

```
S_dataset.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 268074 entries, 0 to 268073
Data columns (total 15 columns):
 #   Column            Non-Null Count   Dtype
---  ------            --------------   -----
 0   FQDN_count        268074 non-null  int64
 1   subdomain_length  268074 non-null  int64
 2   upper             268074 non-null  int64
 3   lower             268074 non-null  int64
 4   numeric           268074 non-null  int64
 5   entropy           268074 non-null  float64
 6   special           268074 non-null  int64
 7   labels            268074 non-null  int64
 8   labels_max        268074 non-null  int64
 9   labels_average    268074 non-null  float64
 10  longest_word      268074 non-null  int32
 11  sld               268074 non-null  int32
 12  len               268074 non-null  int64
 13  subdomain         268074 non-null  int64
 14  Target Attack     268074 non-null  int64
dtypes: float64(2), int32(2), int64(11)
memory usage: 28.6 MB
```

- **Check Skewness of the features**

```
S_dataset.skew()
```

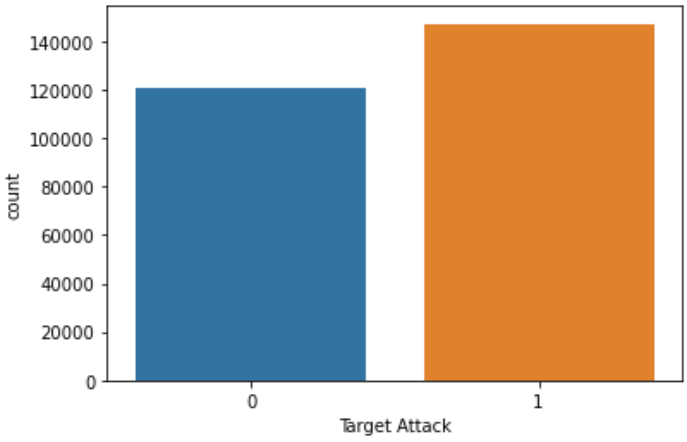
```
FQDN_count          -1.101731
subdomain_length    -0.590480
upper                5.988737
lower                0.343449
numeric             -0.594384
entropy             -0.140156
special             -0.902972
labels              -0.903680
labels_max           3.979910
labels_average       5.087081
longest_word         2.269378
sld                180.987411
len                  2.634801
subdomain           -1.176397
Target Attack       -0.197046
dtype: float64
```

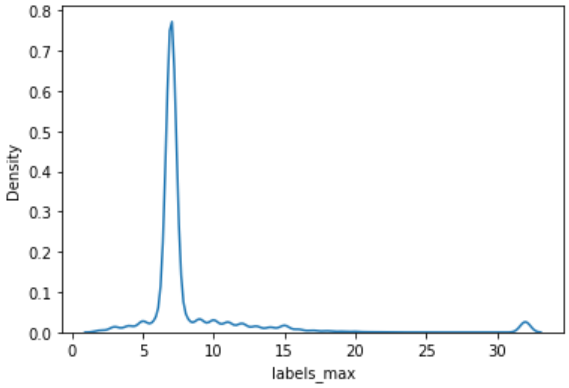
- **Check there is no null values**

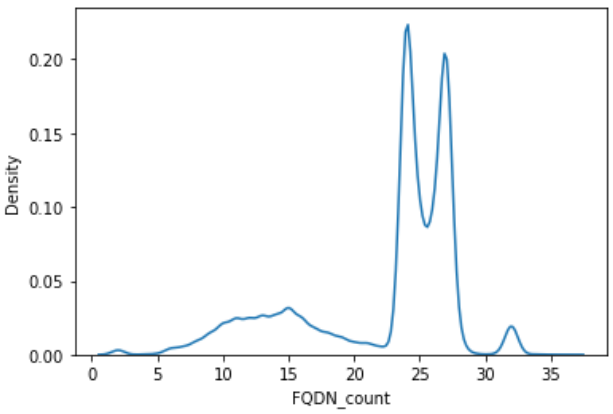
```
S_dataset.isnull().sum()
```

```
FQDN_count          0
subdomain_length    0
upper               0
lower               0
numeric             0
entropy             0
special             0
labels              0
labels_max          0
labels_average      0
longest_word        0
sld                 0
len                 0
subdomain           0
Target Attack       0
dtype: int64
```

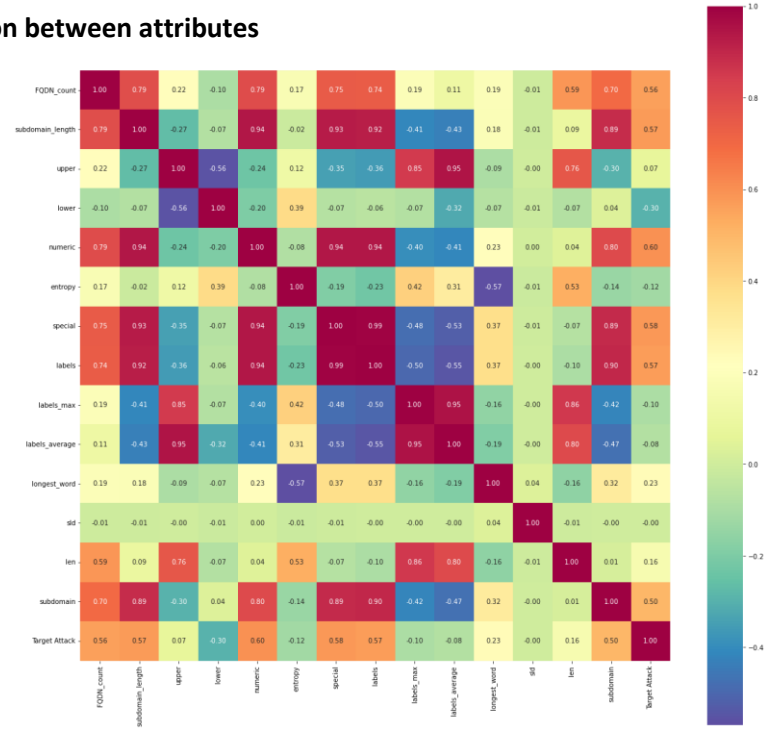
- **Count the Target Attacks**







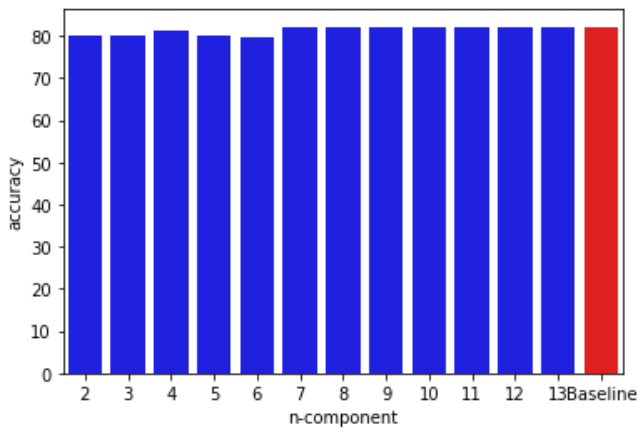
- **Correlation between attributes**



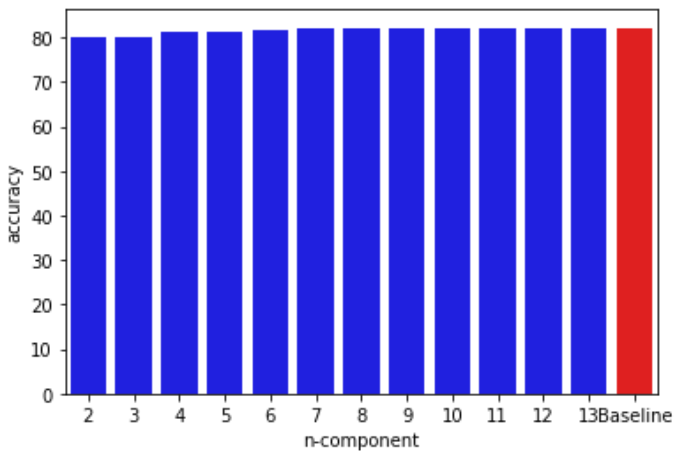
Feature Selection using filter selection:

- **Mutual_Info filter method:**
  - max mutal 82.20307370934049
  - Best value of n components: 13
  - ['FQDN_count', 'subdomain_length', 'lower', 'numeric', 'entropy', 'special', 'labels', 'labels_max', 'labels_average', 'longest_word', 'sld', 'len', 'subdomain']
- **Chi2 filter method:**
  - max chi2 82.20307370934049
  - Best value of n components: 8
  - **Best features**: ['FQDN_count', 'subdomain_length', 'upper', 'lower', 'numeric', 'special', 'labels', 'sld']

Chi2 filter method



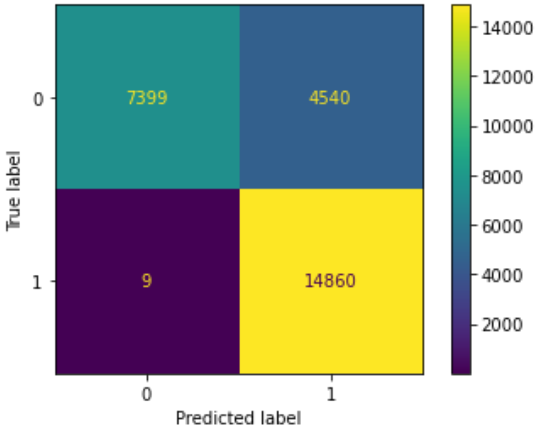
Mutual_Info filter method



Model Training and Evaluation:

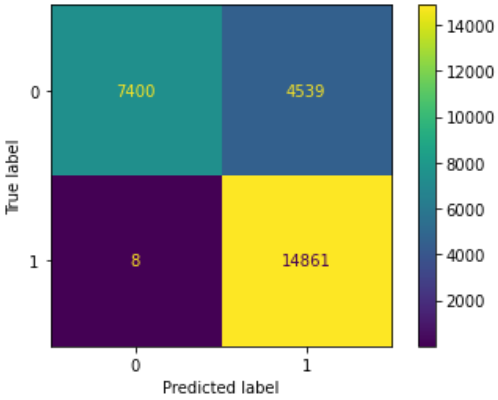
Decision Tree with Feature selection

```
Classification Report:

              precision    recall  f1-score   support

           0       1.00      0.62      0.76     11939
           1       0.77      1.00      0.87     14869

    accuracy                           0.83     26808
   macro avg       0.88      0.81      0.82     26808
weighted avg       0.87      0.83      0.82     26808


Confusion Matrix:

[[ 7399  4540]
 [    9 14860]]

Accuracy Score:

0.8303118472097881
```



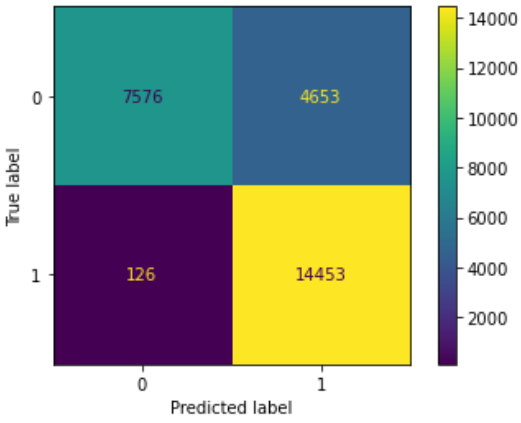
Decision Tree after hyperparameter tuning:

```
Classification Report:

              precision    recall  f1-score   support

           0       1.00      0.62      0.76     11939
           1       0.77      1.00      0.87     14869

    accuracy                           0.83     26808
   macro avg       0.88      0.81      0.82     26808
weighted avg       0.87      0.83      0.82     26808


Confusion Matrix:

[[ 7400  4539]
 [    8 14861]]

Accuracy Score:

0.8303864518054312
```



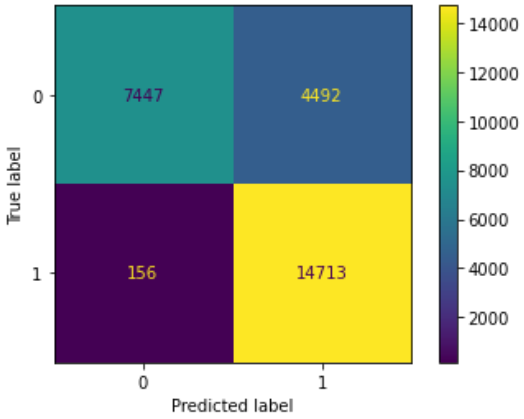
Logistic Regression:

```
Classification Report:

              precision    recall  f1-score   support

           0       0.98      0.62      0.76     12229
           1       0.76      0.99      0.86     14579

    accuracy                           0.82     26808
   macro avg       0.87      0.81      0.81     26808
weighted avg       0.86      0.82      0.81     26808


Confusion Matrix:

[[ 7576  4653]
 [  126 14453]]

Accuracy Score:

0.8217323187108326
```



Logistic regression with hyperparameter tuning:

```
Classification Report:

              precision    recall  f1-score   support

           0       0.98      0.62      0.76     11939
           1       0.77      0.99      0.86     14869

    accuracy                           0.83     26808
   macro avg       0.87      0.81      0.81     26808
weighted avg       0.86      0.83      0.82     26808


Confusion Matrix:

[[ 7447  4492]
 [  156 14713]]

Accuracy Score:

0.8266189197254551
```



The used Evaluation metric is **Accuracy** since the target data is approximately balanced

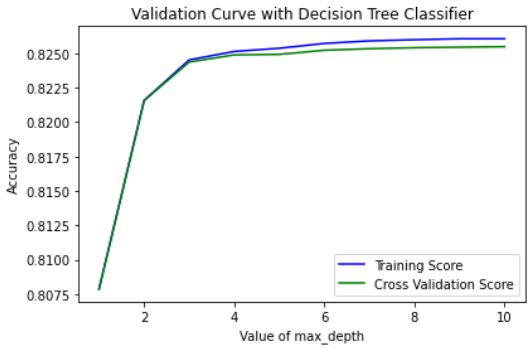
I used decision tree and logistic regression models it appears that decision tree (83.02) shows higher accuracy than logistic regression (82%) so I choose Decision tree as the champion model and perform hyperparameter tuning for DT in order to find the best hyperparameter and the result that the default DT's hyperparameter are the best.

Cross_Validatio for Decision tree model:

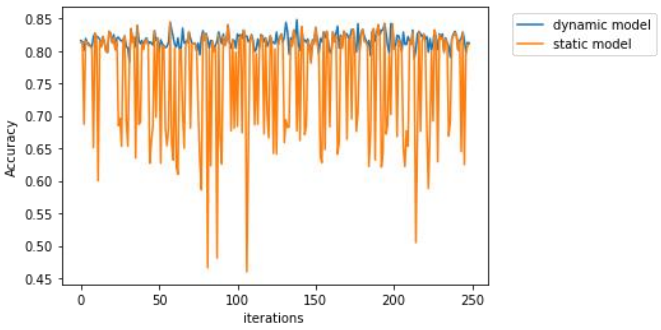
Accuracy: 0.83 (+/- 0.01)

K-fold cross validation score: fo each [0.81947035 0.82954122 0.83215218 0.82879523 0.81797837 0.82767624

0.82245431 0.8261194 0.83022388 0.82052239]



Part II (Dynamic Model):



Dynamic model (84.8%) got slightly higher accuracy than **static model (84.4%)**

```
Window 1
Dynamic Model accuracy without retrain = 81.3%
 The model will be trained on the new data
ACC of Dynamic Model after retrain = 81.6%
ACC of Static Model = 81.3%
========================================
Window 2
Dynamic Model accuracy without retrain = 81.2%
 The model will be trained on the new data
ACC of Dynamic Model after retrain = 81.39999999999999%
ACC of Static Model = 81.2%
========================================
Window 3
Dynamic Model accuracy without retrain = 68.7%
 The model will be trained on the new data
ACC of Dynamic Model after retrain = 80.10000000000001%
ACC of Static Model = 68.7%
========================================
```