Report The DNS Anomaly Detection

Team Members:

1- Youssef Saeed Thabet	2205064
2- Basel Yasser	2205015
3- Mohamed Ramadan	2205000

About The Data We got:

1- We get the data from the Kaggle website.

2- The data contains 15 Features or 15 columns and the 10000 rows.

3- The Data Features are:

- Query Volume: Number of DNS queries per time unit.
- Query Rate: Rate of DNS queries per time unit.
- Query Type: Type of DNS query (A, AAAA, MX, etc.).
- Query Domain: Domain name being queried.
- Response Time: Time taken to receive a response to a query.
- TTL Value: Time-to-live value in the DNS response.
- Source IP Address: IP address of the client making the query.
- Destination IP Address: IP address of the DNS server.
- Transaction ID: Unique identifier for each DNS transaction.

- Query Size: Size of the DNS query in bytes.

- Response Size: Size of the DNS response in bytes.

- Entropy of Transaction ID: A measure of randomness in the transaction ID.

- Frequency of Query Domain: How often the domain is queried.

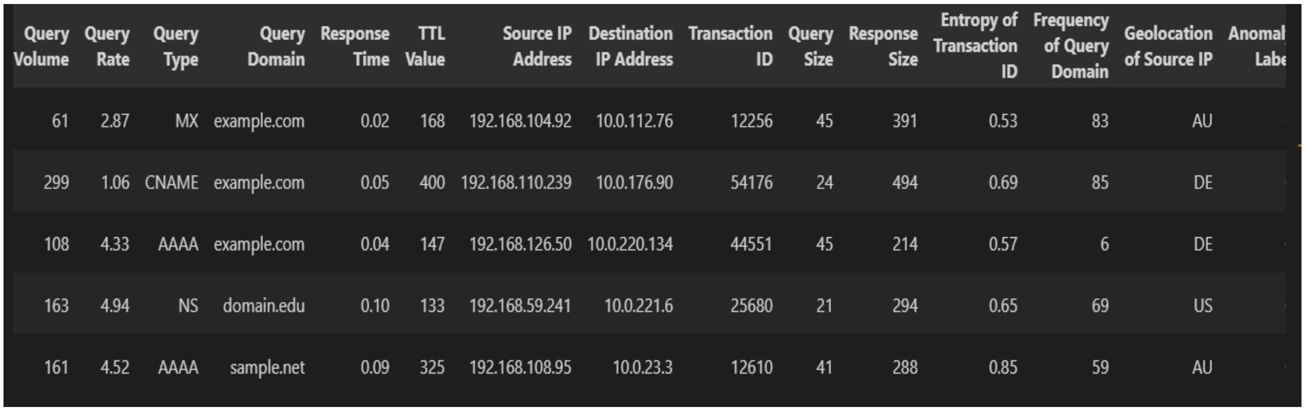
- Geolocation of Source IP: Geographic location of the client.

- Anomaly Label: A binary label indicating whether the record is anomalous (1) or normal (0)

1- First Importing the basic libraries for Preprocessing and Machine Learning model and printing the first 10 rows :

```
import pandas as pd
from sklearn.preprocessing import LabelEncoder
from sklearn.model_selection import train_test_split
from sklearn.ensemble import RandomForestClassifier,
GradientBoostingClassifier
from sklearn.linear_model import LogisticRegression
from sklearn.metrics import accuracy_score, classification_report,
confusion_matrix
from imblearn.over_sampling import SMOTE
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn.metrics import accuracy_score, classification_report,
confusion_matrix
import warnings
warnings.filterwarnings("ignore")
df = pd.read_csv('synthetic_dns_traffic_data.csv')
df.head()
```

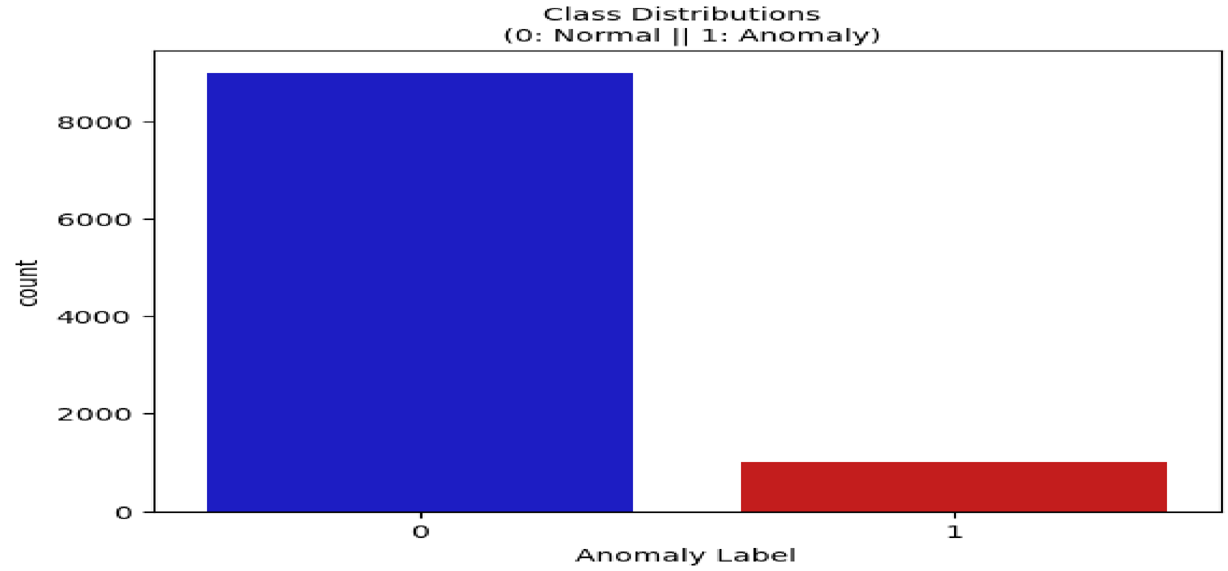
2- displaying the first 10 rows in the data like that :



Query Volume	Query Rate	Query Type	Query Domain	Response Time	TTL Value	Source IP Address	Destination IP Address	Transaction ID	Query Size	Response Size	Entropy of Transaction ID	Frequency of Query Domain	Geolocation of Source IP	Anomal Labe
61	2.87	MX	example.com	0.02	168	192.168.104.92	10.0.112.76	12256	45	391	0.53	83	AU	
299	1.06	CNAME	example.com	0.05	400	192.168.110.239	10.0.176.90	54176	24	494	0.69	85	DE	
108	4.33	AAAA	example.com	0.04	147	192.168.126.50	10.0.220.134	44551	45	214	0.57	6	DE	
163	4.94	NS	domain.edu	0.10	133	192.168.59.241	10.0.221.6	25680	21	294	0.65	69	US	
161	4.52	AAAA	sample.net	0.09	325	192.168.108.95	10.0.23.3	12610	41	288	0.85	59	AU	

3-After that we doing the class distribution of the data anomaly and not anomaly :

```
import seaborn as sns
import matplotlib.pyplot as plt
colors = ["#0101DF", "#DF0101"]
sns.countplot(x='Anomaly Label', data=df, palette=colors)
plt.title('Class Distributions \n (0: Normal || 1:
Anomaly)', fontsize=10)
```



4- After That we doing the data transformation for each feature:

```
df.dtypes
```

5- After That we encoding categorical features (objects):

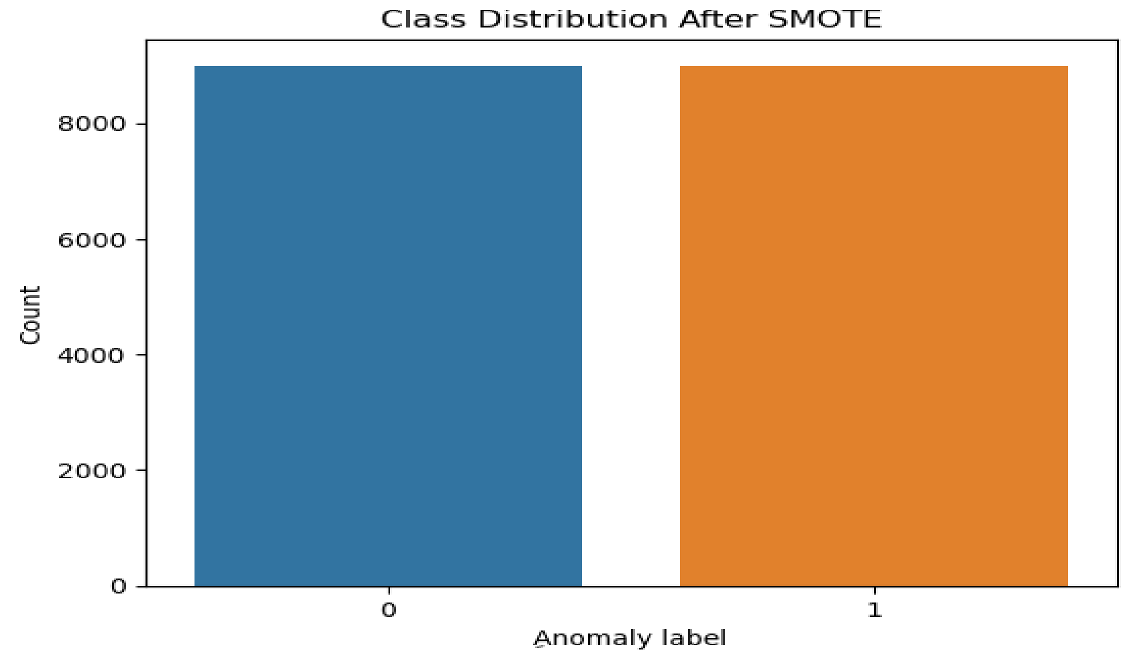
```
from sklearn.preprocessing import LabelEncoder
label_encoder = LabelEncoder()
df['Query Type'] = label_encoder.fit_transform(df['Query Type'])
df['Query Domain'] = label_encoder.fit_transform(df['Query Domain'])
df['Source IP Address'] = label_encoder.fit_transform(df['Source IP
Address'])
df['Destination IP Address'] =
label_encoder.fit_transform(df['Destination IP Address'])
df['Geolocation of Source IP'] =
label_encoder.fit_transform(df['Geolocation of Source IP'])
df.head()
```

6- After That we split the features and the target class (Anomaly label).

```
features = df.drop(columns=['Timestamp', 'Anomaly Label'])
target = df['Anomaly Label']
```

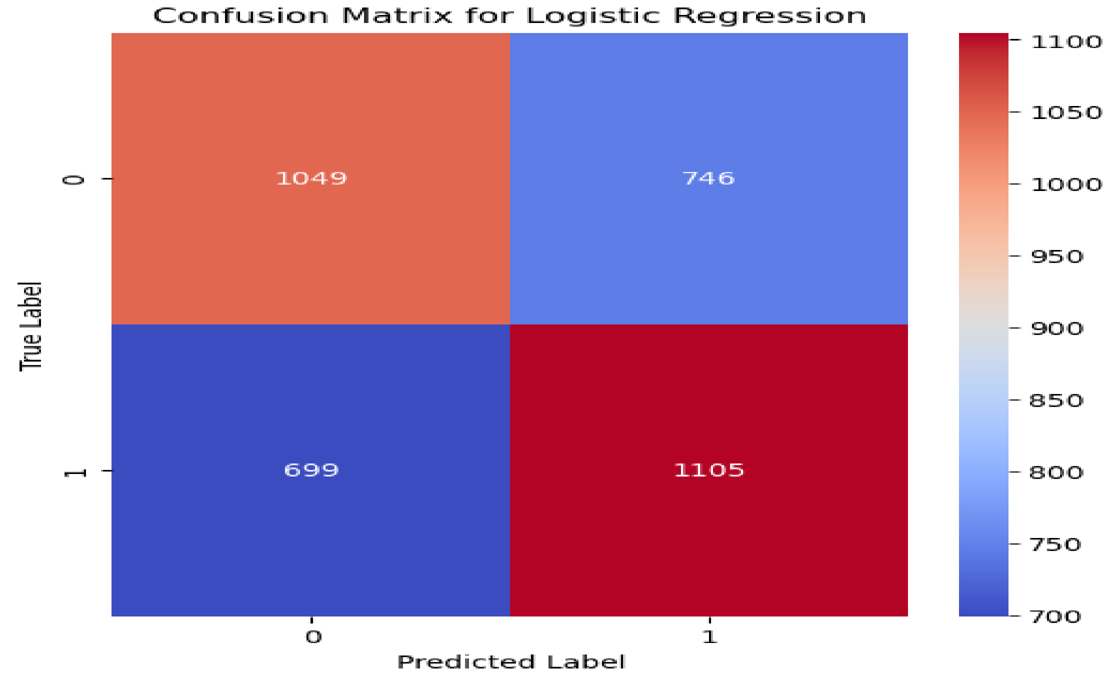
7- After that we doing Data Balancing to balance The data using SMOTE library:

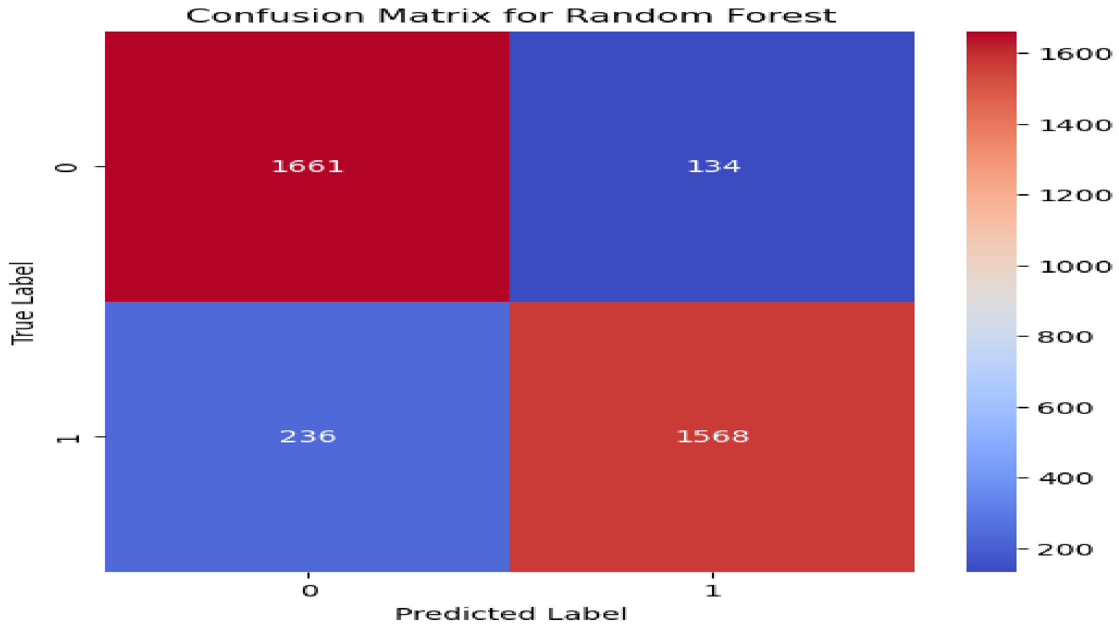
```
from imblearn.over_sampling import SMOTE
smote = SMOTE(random_state=42)
X_resampled, y_resampled = smote.fit_resample(features,
target)
```

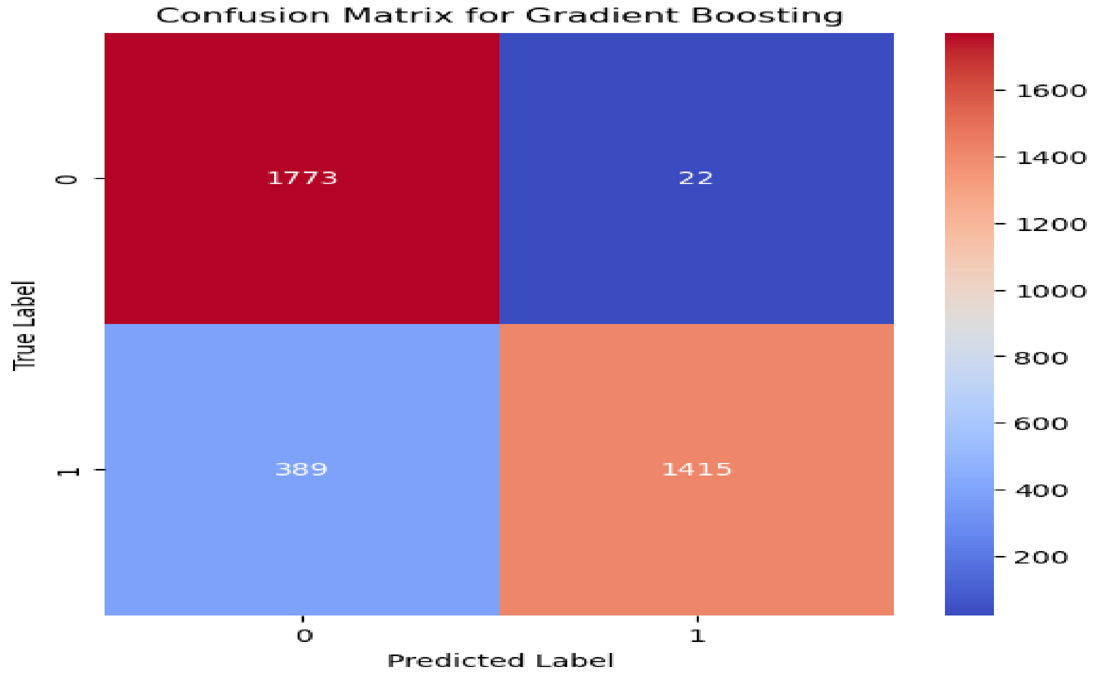


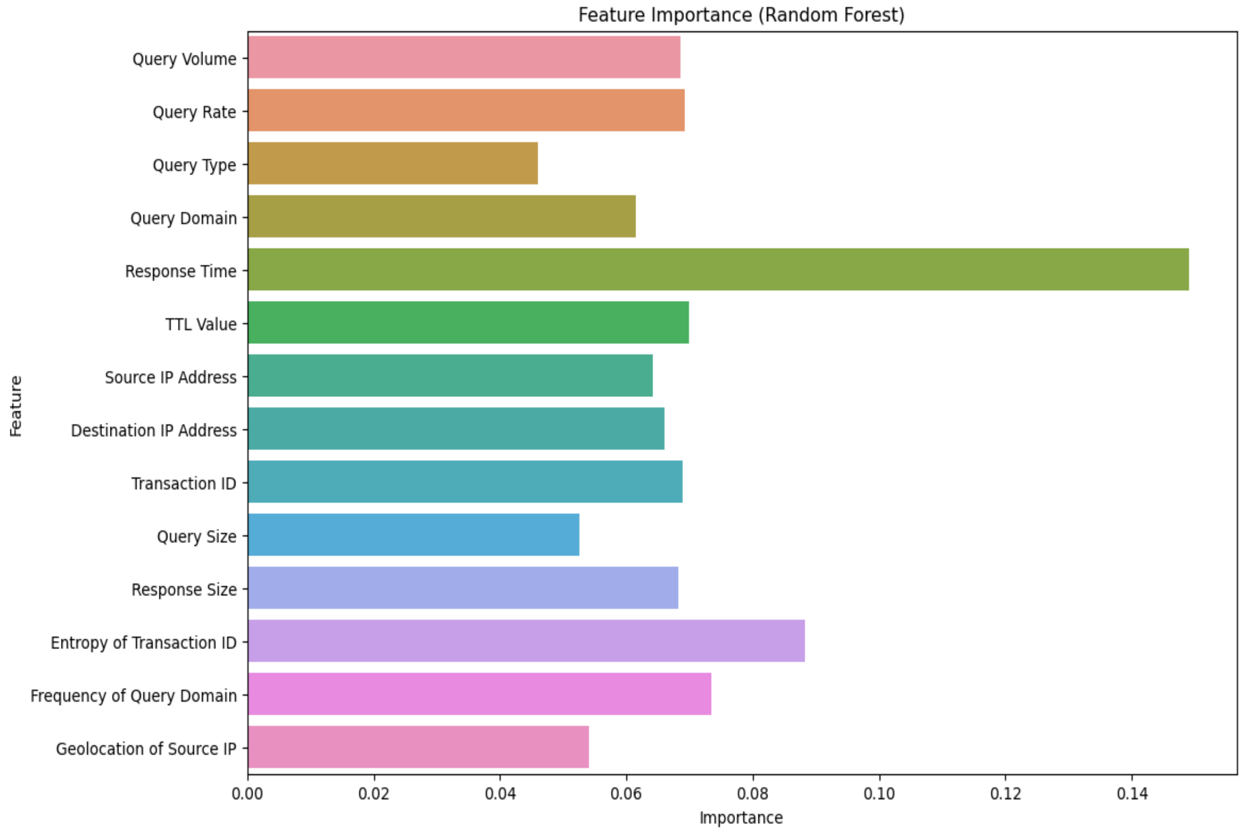
8- After that we doing Data splitting for testing 20% and Training for 80%:

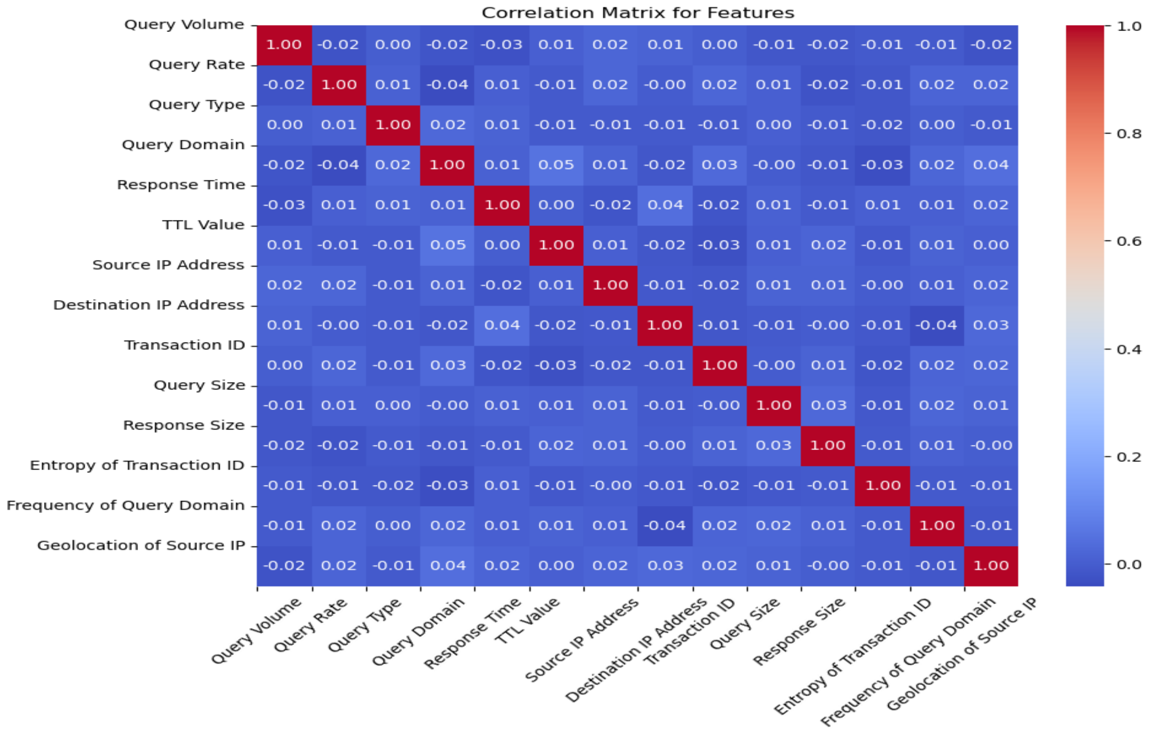
9- After that we tarin 3 models first, RandomForestClassifier Second, GradientBoostingClassifier Third, LogisticRegression.




Confusion Matrix for Random Forest
True Label
0
1
1661
134
236
1568
0
1
Predicted Label
1600
1400
1200
1000
800
600
400
200


Confusion Matrix for Gradient Boosting
True Label
0
1
1773
22
389
1415
0
1
Predicted Label
1600
1400
1200
1000
800
600
400
200





	Query Volume	Query Rate	Query Type	Query Domain	Response Time	TTL Value	Source IP Address	Destination IP Address	Transaction ID	Query Size	Response Size	Entropy of Transaction ID	Frequency of Query Domain	Geolocation of Source IP
Query Volume	1.00	-0.02	0.00	-0.02	-0.03	0.01	0.02	0.01	0.00	-0.01	-0.02	-0.01	-0.01	-0.02
Query Rate	-0.02	1.00	0.01	-0.04	0.01	-0.01	0.02	-0.00	0.02	0.01	-0.02	-0.01	0.02	0.02
Query Type	0.00	0.01	1.00	0.02	0.01	-0.01	-0.01	-0.01	-0.01	0.00	-0.01	-0.02	0.00	-0.01
Query Domain	-0.02	-0.04	0.02	1.00	0.01	0.05	0.01	-0.02	0.03	-0.00	-0.01	-0.03	0.02	0.04
Response Time	-0.03	0.01	0.01	0.01	1.00	0.00	-0.02	0.04	-0.02	0.01	-0.01	0.01	0.01	0.02
TTL Value	0.01	-0.01	-0.01	0.05	0.00	1.00	0.01	-0.02	-0.03	0.01	0.02	-0.01	0.01	0.00
Source IP Address	0.02	0.02	-0.01	0.01	-0.02	0.01	1.00	-0.01	-0.02	0.01	0.01	-0.00	0.01	0.02
Destination IP Address	0.01	-0.00	-0.01	-0.02	0.04	-0.02	-0.01	1.00	-0.01	-0.01	-0.00	-0.01	-0.04	0.03
Transaction ID	0.00	0.02	-0.01	0.03	-0.02	-0.03	-0.02	-0.01	1.00	-0.00	0.01	-0.02	0.02	0.02
Query Size	-0.01	0.01	0.00	-0.00	0.01	0.01	0.01	-0.01	-0.00	1.00	0.03	-0.01	0.02	0.01
Response Size	-0.02	-0.02	-0.01	-0.01	-0.01	0.02	0.01	-0.00	0.01	0.03	1.00	-0.01	0.01	-0.00
Entropy of Transaction ID	-0.01	-0.01	-0.02	-0.03	0.01	-0.01	-0.00	-0.01	-0.02	-0.01	-0.01	1.00	-0.01	-0.01
Frequency of Query Domain	-0.01	0.02	0.00	0.02	0.01	0.01	0.01	-0.04	0.02	0.02	0.01	-0.01	1.00	-0.01
Geolocation of Source IP	-0.02	0.02	-0.01	0.04	0.02	0.00	0.02	0.03	0.02	0.01	-0.00	-0.01	-0.01	1.00

In this project we concentrate on features all columns except the timestamp and the target (Anomaly label) and the target is the Anomaly label.

The best model that has the higher accuracy is the Random Forest algorithm with 90% accuracy after that the Gradient Boosting after that the Logistic Regression

Thank You !

DR : Mohamed Mostafa