Algorithme de détection de tentative d'hameçonnage par analyse d'URL.

Préparation de l'environnement de travail.

In [ ]:

```
#Importation des librairies nécessaires à la création d'un modèle.
import pandas as pd
import numpy as np
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
from scipy.stats import norm
from scipy import stats
from scipy.stats import yeojohnson
from sklearn.model_selection import train_test_split
from sklearn.ensemble import RandomForestClassifier
from sklearn.metrics import classification_report
from sklearn.metrics import accuracy_score
import optuna
import joblib
import warnings
#On vient aussi filtrer les avertissements émis par la librairie 'seaborn_oldcor
warnings.filterwarnings('ignore', category=FutureWarning, module='seaborn._oldcc
#Lecture des données.
df = pd.read_csv('C:/Users/bouch/OneDrive/Documents/M2MI/MASTER/Etude de cas/arc
```

Préparation des données.

In [ ]:

```
#Exploration de nos données.
print(f"{df.dtypes}\n")
#Taille de nos données.
print(f"Dimensions: {df.shape[0]} x {df.shape[1]}\n")
#Classification du types de données.
datatype_counts = df.dtypes.value_counts()
for dtype, count in datatype_counts.items():
    print(f"{dtype}: {count} columns")

#Abandon de la colonnes 'id'.
df = df.drop("id", axis=1)
#Vérification de la distribution de nos données.
df.describe()
```

```
id                         int64
NumDots                    int64
SubdomainLevel             int64
PathLevel                  int64
UrlLength                  int64
NumDash                    int64
NumDashInHostname          int64
AtSymbol                   int64
TildeSymbol                int64
NumUnderscore              int64
NumPercent                 int64
NumQueryComponents         int64
NumAmpersand               int64
NumHash                    int64
NumNumericChars            int64
NoHttps                    int64
RandomString               int64
IpAddress                  int64
DomainInSubdomains         int64
DomainInPaths              int64
HttpsInHostname            int64
HostnameLength             int64
PathLength                 int64
QueryLength                int64
DoubleSlashInPath          int64
NumSensitiveWords          int64
EmbeddedBrandName          int64
CLASS_LABEL                int64
dtype: object

Dimensions: 10000 x 28

int64: 28 columns
```

Out[ ]:

	NumDots	SubdomainLevel	PathLevel	UrlLength	NumDash	NumD
count	10000.000000	10000.000000	10000.000000	10000.000000	10000.000000	
mean	2.445100	0.586800	3.300300	70.264100	1.818000	
std	1.346836	0.751214	1.863241	33.369877	3.106258	
min	1.000000	0.000000	0.000000	12.000000	0.000000	
25%	2.000000	0.000000	2.000000	48.000000	0.000000	
50%	2.000000	1.000000	3.000000	62.000000	0.000000	
75%	3.000000	1.000000	4.000000	84.000000	2.000000	
max	21.000000	14.000000	18.000000	253.000000	55.000000	

8 rows × 27 columns

In [ ]:

```
#Vérification de la présence de valeurs manquantes.
null = df.isnull().sum()
for i in range(len(df.columns)):
    print(f"{df.columns[i]}: {null[i]} ({(null[i]/len(df))*100}%)")
total_cellules = np.prod(df.shape)
```

```
total_absent = null.sum()
print(f"\nPourcentage total de valeures manquantes: {total_absent} ({(total_abse
```

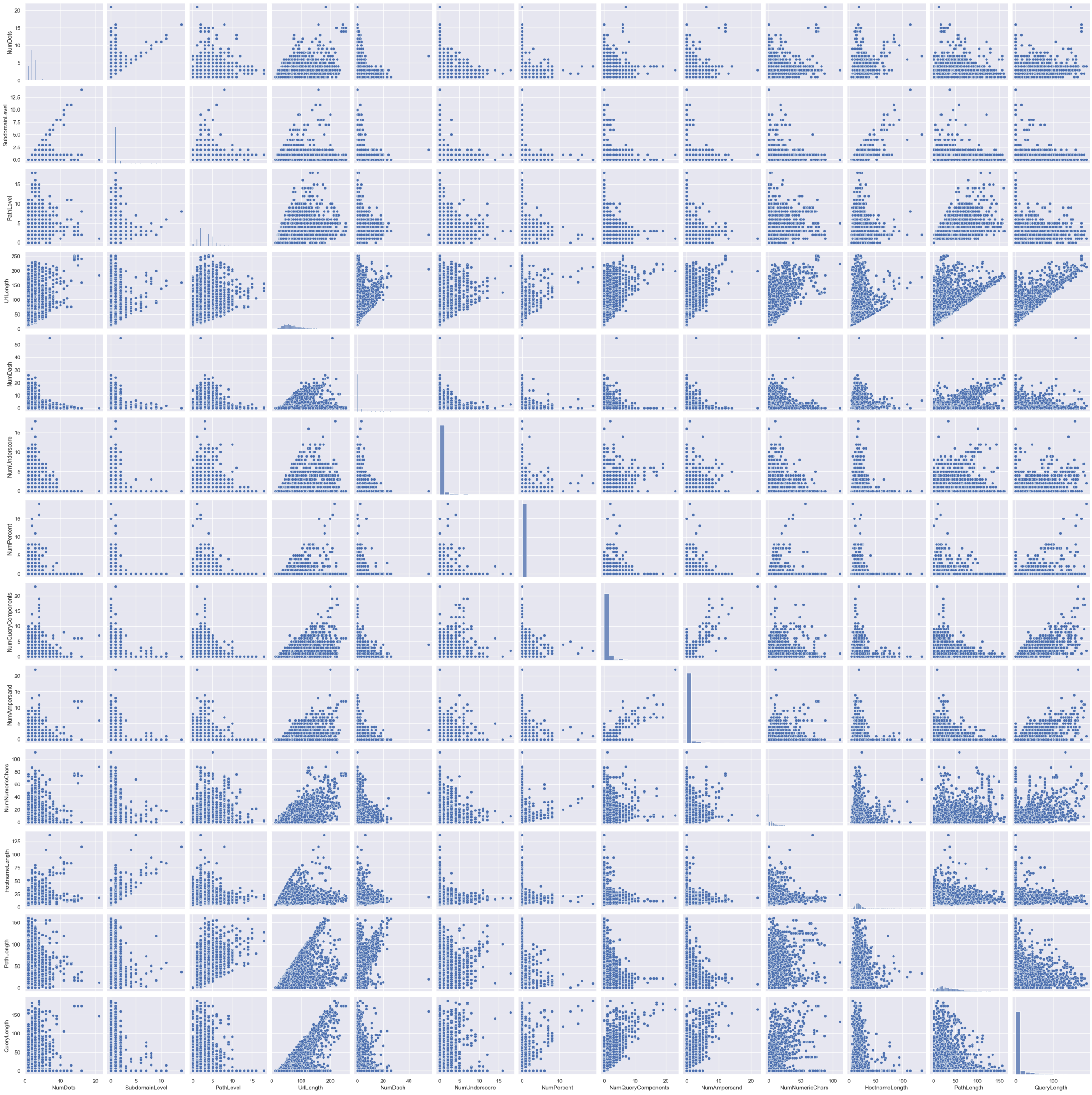
```
NumDots: 0 (0.0%)
SubdomainLevel: 0 (0.0%)
PathLevel: 0 (0.0%)
UrlLength: 0 (0.0%)
NumDash: 0 (0.0%)
NumDashInHostname: 0 (0.0%)
AtSymbol: 0 (0.0%)
TildeSymbol: 0 (0.0%)
NumUnderscore: 0 (0.0%)
NumPercent: 0 (0.0%)
NumQueryComponents: 0 (0.0%)
NumAmpersand: 0 (0.0%)
NumHash: 0 (0.0%)
NumNumericChars: 0 (0.0%)
NoHttps: 0 (0.0%)
RandomString: 0 (0.0%)
IpAddress: 0 (0.0%)
DomainInSubdomains: 0 (0.0%)
DomainInPaths: 0 (0.0%)
HttpsInHostname: 0 (0.0%)
HostnameLength: 0 (0.0%)
PathLength: 0 (0.0%)
QueryLength: 0 (0.0%)
DoubleSlashInPath: 0 (0.0%)
NumSensitiveWords: 0 (0.0%)
EmbeddedBrandName: 0 (0.0%)
CLASS_LABEL: 0 (0.0%)

Pourcentage total de valeures manquantes: 0 (0.0%)
```

```
C:\Users\bouch\AppData\Local\Temp\ipykernel_6240\567799338.py:4: FutureWarning: S
eries.__getitem__ treating keys as positions is deprecated. In a future version,
integer keys will always be treated as labels (consistent with DataFrame behavio
r). To access a value by position, use `ser.iloc[pos]`
  print(f"{df.columns[i]}: {null[i]} ({(null[i]/len(df))*100}%)")
```

Exploration des données.

Variables continues et relations entre variables.

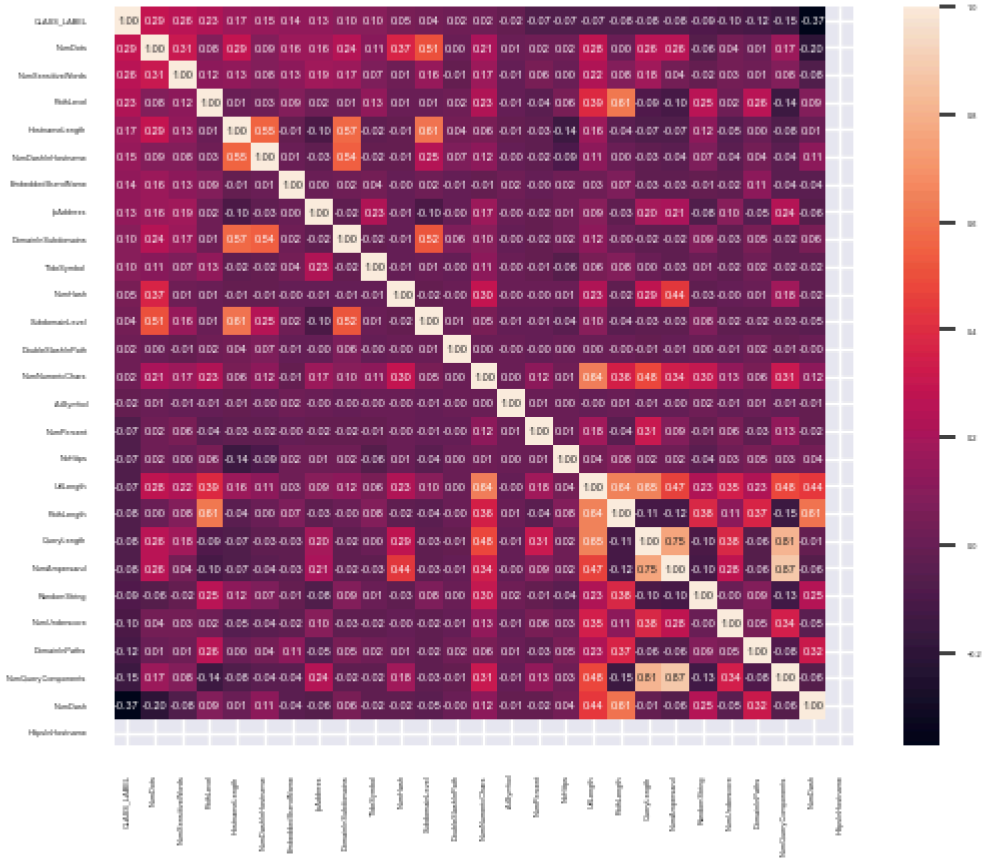
```
In [ ]: #Visualisation de nos variables continues.
        def is_continuous(series):
            return series.nunique() > 10

        colonne_continue = [col for col in df.columns if is_continuous(df[col])]
        sns.pairplot(df[colonne_continue], height = 2.5)
        plt.show()
```



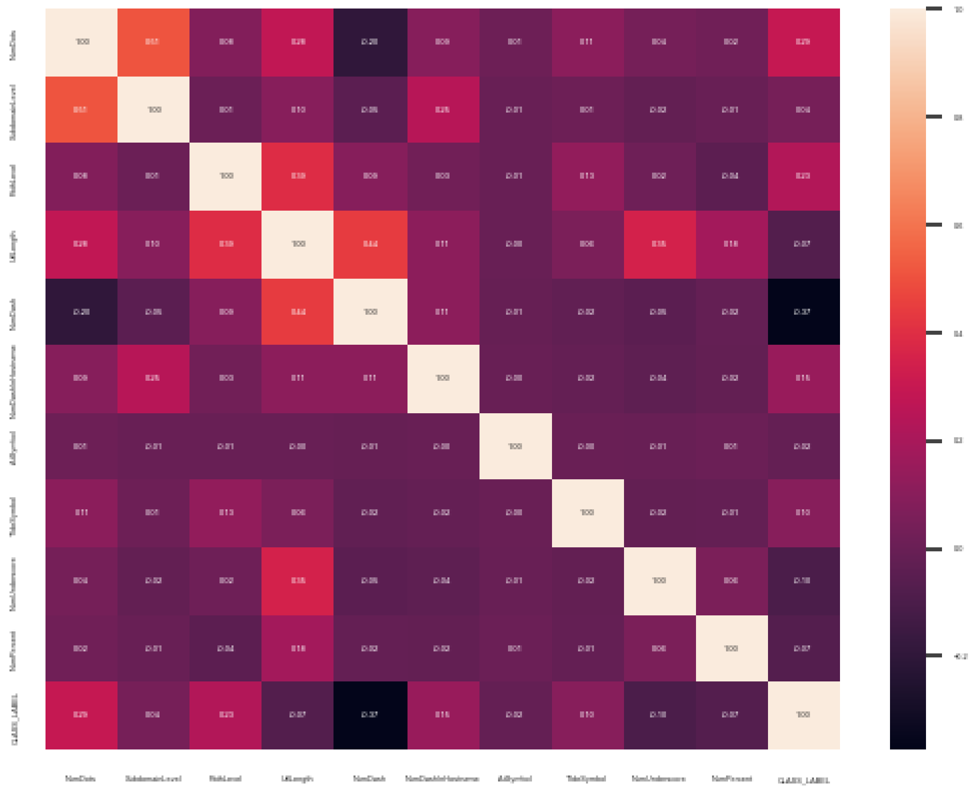
Etude des corrélations.

```
In [ ]: #Affichage des corrélations.
        corr = df.corr()
        cols = corr.nlargest(50, 'CLASS_LABEL')['CLASS_LABEL'].index
        cm = np.corrcoef(df[cols].values.T)
        sns.set_theme(font_scale=0.25)
        hm = sns.heatmap(cm, cbar=True, annot=True, square=True, fmt='.2f', annot_kws={'
        plt.show()

        #Affichage ajusté au nombre de variables.
        def hm(df, idx_s, idx_e):
            y = df['CLASS_LABEL']
            temp = df.iloc[:, idx_s:idx_e]
            temp['CLASS_LABEL'] = y
            sns.heatmap(temp.corr(), annot=True, fmt='.2f')
            plt.show()

        hm(df, 0, 10)
        hm(df, 10, 20)
        hm(df, 20, 30)
```

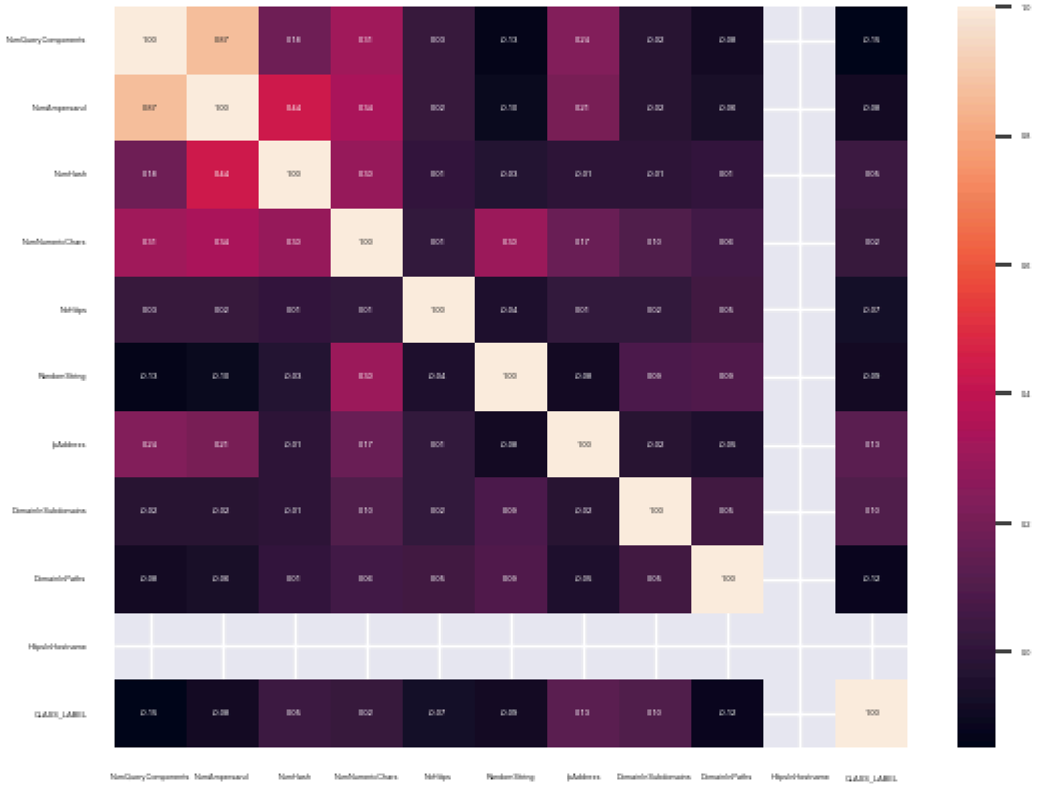
```
c:\Users\bouch\AppData\Local\Programs\Python\Python312\Lib\site-packages\numpy\li
b\function_base.py:2897: RuntimeWarning: invalid value encountered in divide
  c /= stddev[:, None]
c:\Users\bouch\AppData\Local\Programs\Python\Python312\Lib\site-packages\numpy\li
b\function_base.py:2898: RuntimeWarning: invalid value encountered in divide
  c /= stddev[None, :]
```



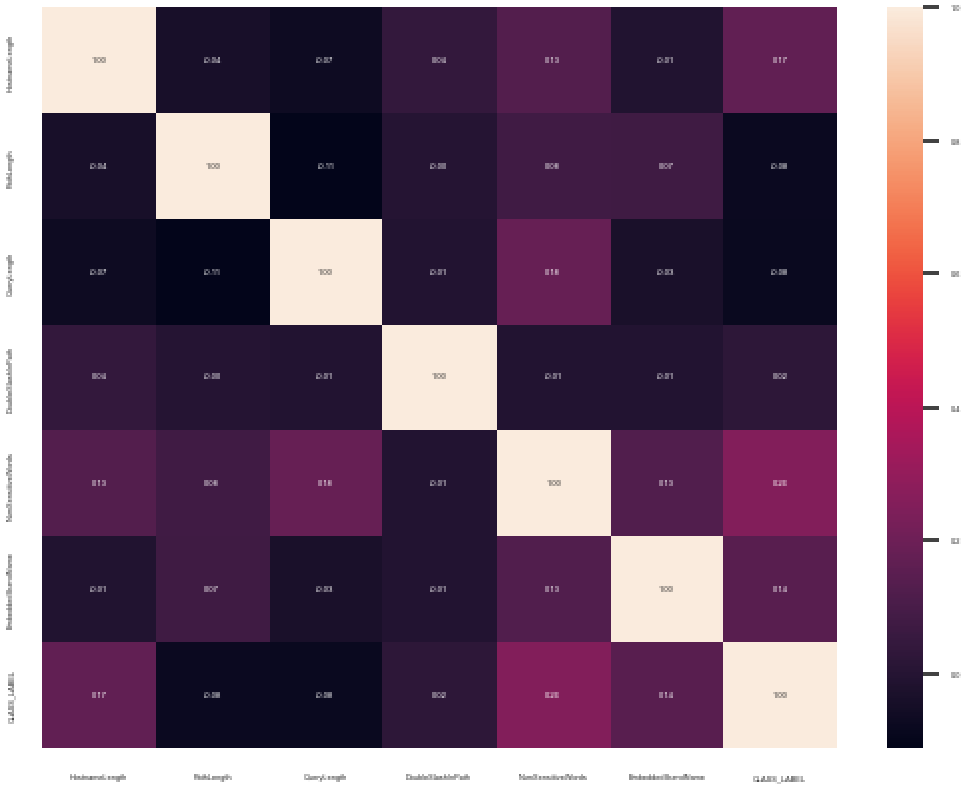
```
C:\Users\bouch\AppData\Local\Temp\ipykernel_6240\2592846879.py:13: SettingWithCop
yWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stabl
e/user_guide/indexing.html#returning-a-view-versus-a-copy
  temp['CLASS_LABEL'] = y
```



```
C:\Users\bouch\AppData\Local\Temp\ipykernel_6240\2592846879.py:13: SettingWithCop
yWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stabl
e/user_guide/indexing.html#returning-a-view-versus-a-copy
  temp['CLASS_LABEL'] = y
```

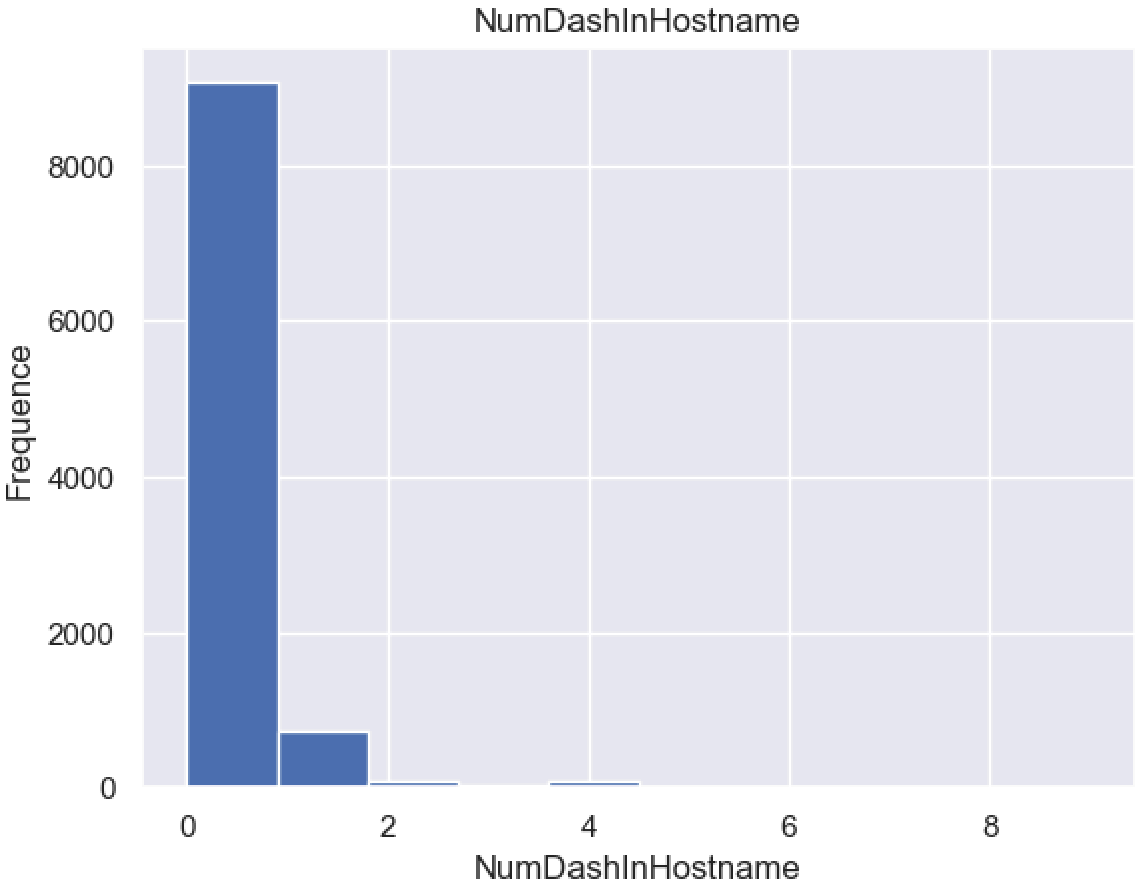


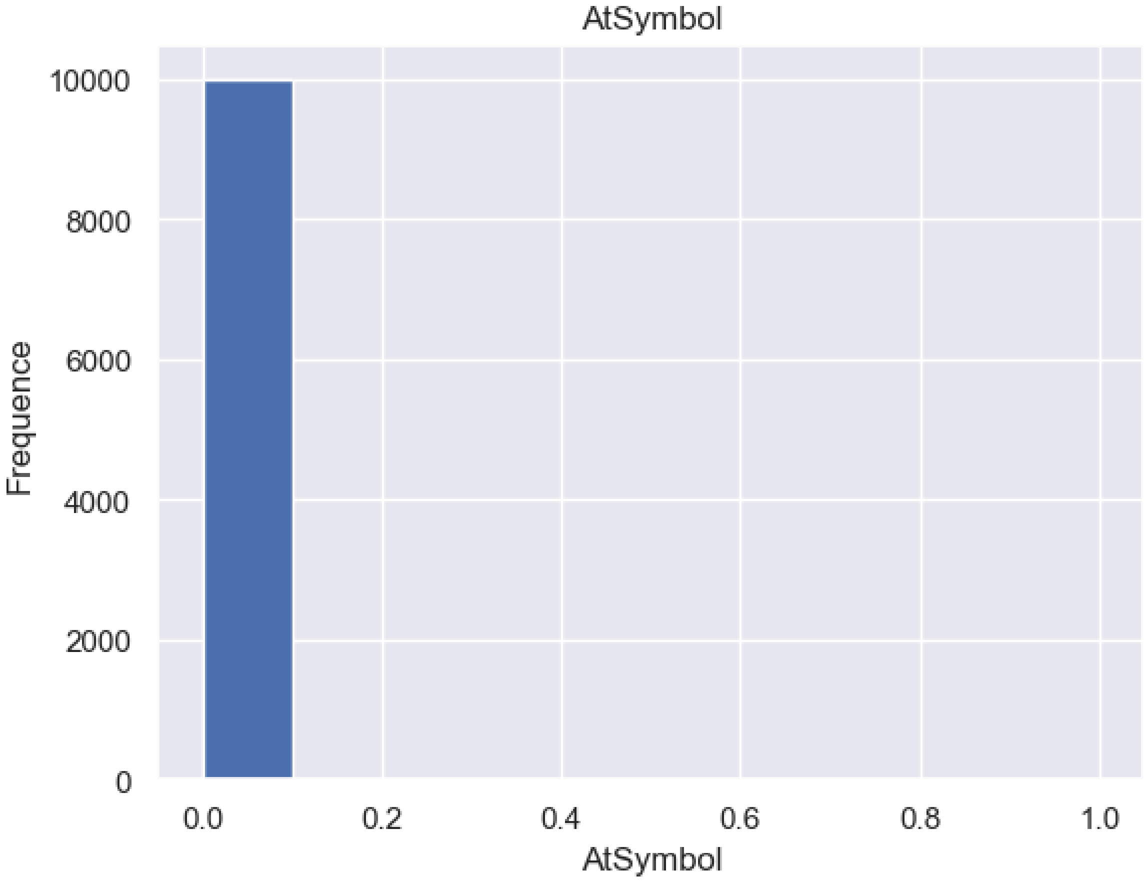
```
C:\Users\bouch\AppData\Local\Temp\ipykernel_6240\2592846879.py:13: SettingWithCop
yWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stabl
e/user_guide/indexing.html#returning-a-view-versus-a-copy
  temp['CLASS_LABEL'] = y
```

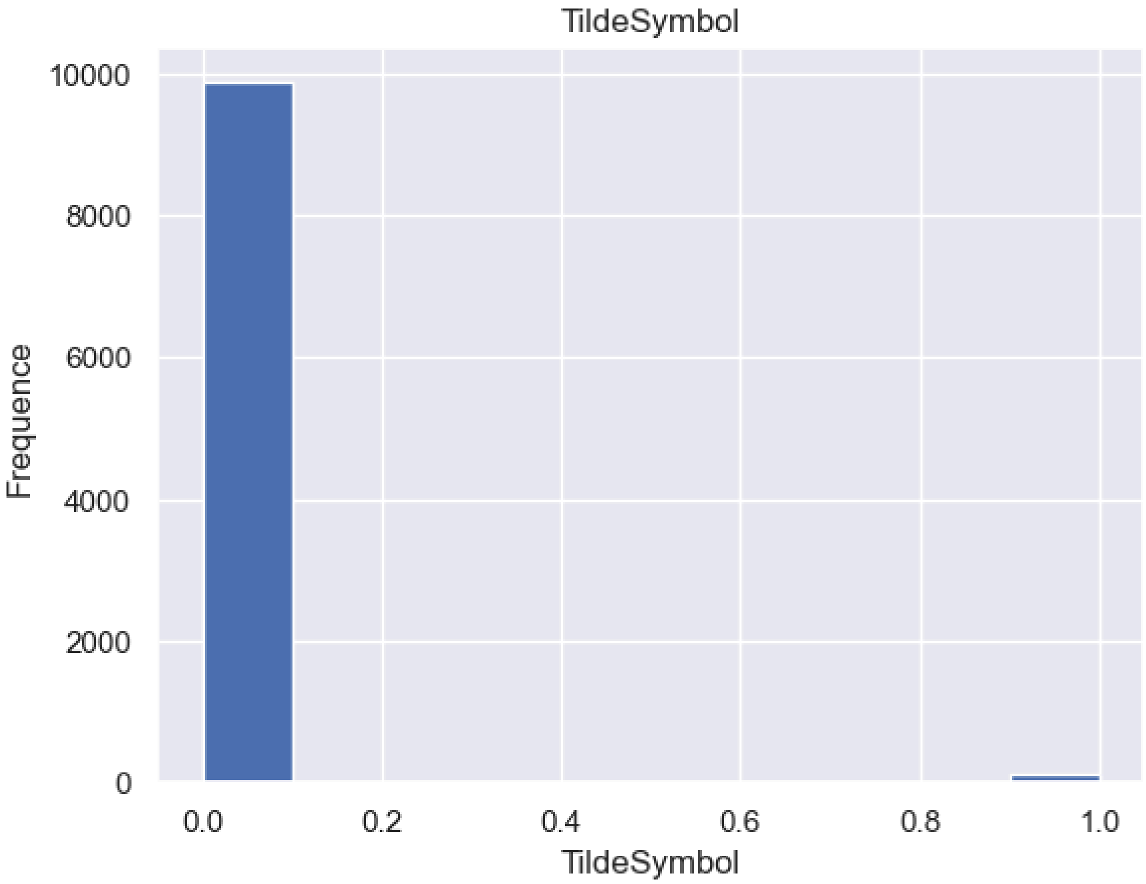


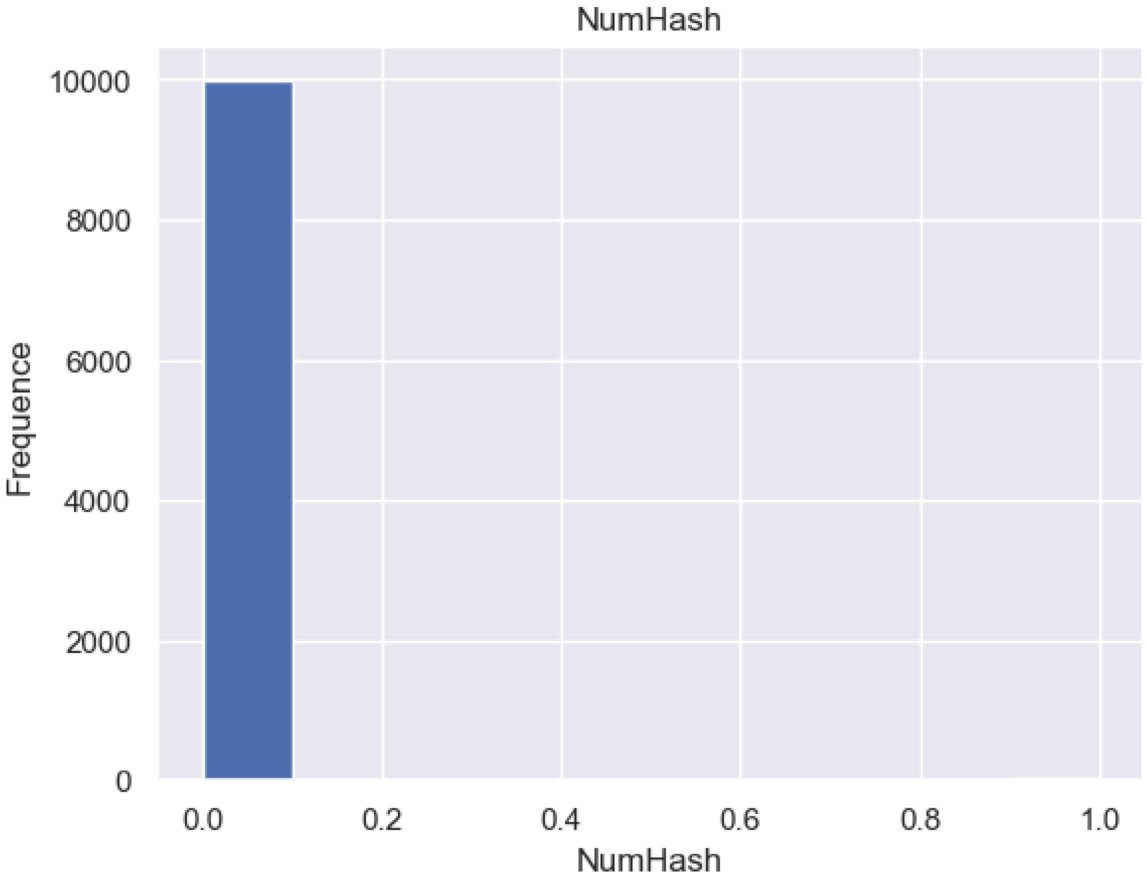
In [ ]:

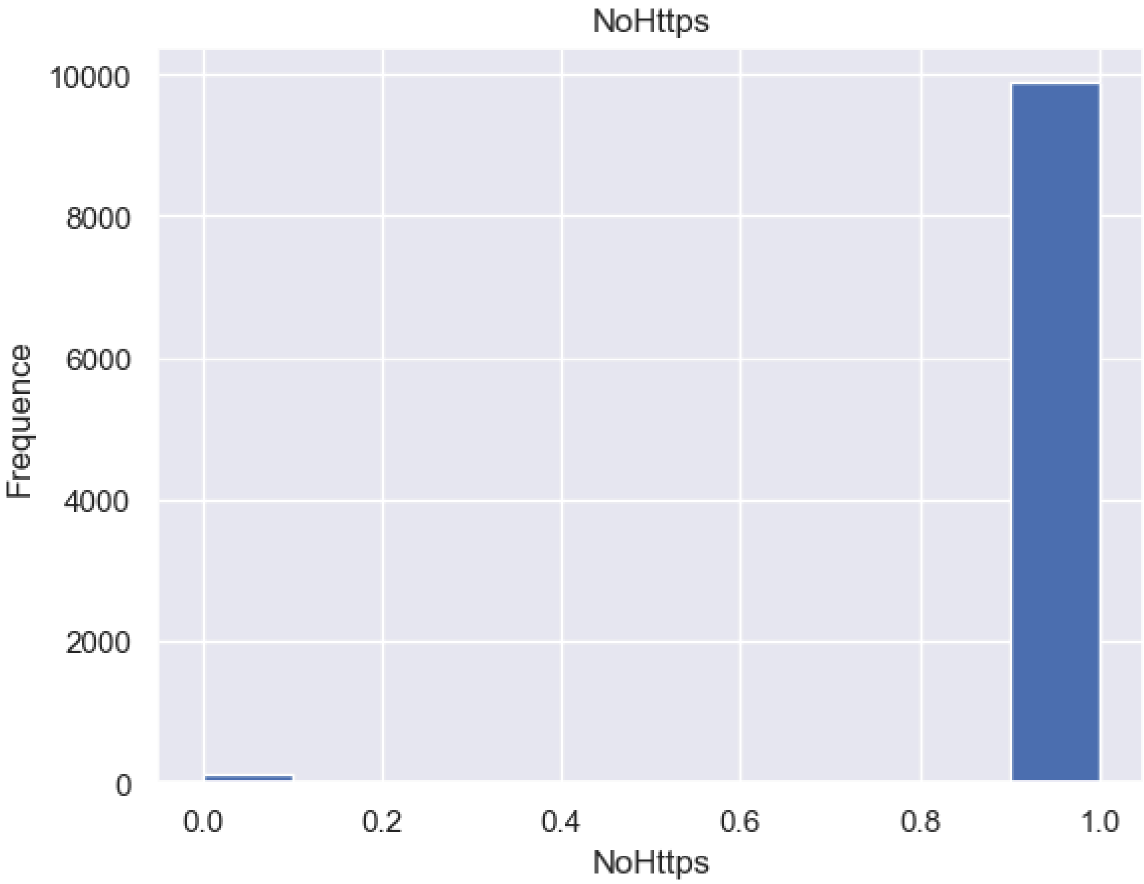
```python
#Vérification de l'étendue de nos données.
colonne_ordinale = [col for col in df.columns if col not in colonne_continue]
sns.set_theme(font_scale=1)
for col in colonne_ordinale:
    plt.hist(df[col], bins=10)
    plt.xlabel(col)
    plt.ylabel('Frequence')
    plt.title(f'{col}')
    plt.show()
    def normal(mean, std, color="black"):
        x = np.linspace(mean-4*std, mean+4*std, 200)
        p = stats.norm.pdf(x, mean, std)
        z = plt.plot(x, p, color, linewidth=2)

for nom_col in colonne_continue:
    fig1, ax1 = plt.subplots()
    sns.histplot(x=df[nom_col], stat="density", ax=ax1)
    normal(df[nom_col].mean(), df[nom_col].std())

    fig2, ax2 = plt.subplots()
    stats.probplot(df[nom_col], plot=ax2)

    plt.show()
```

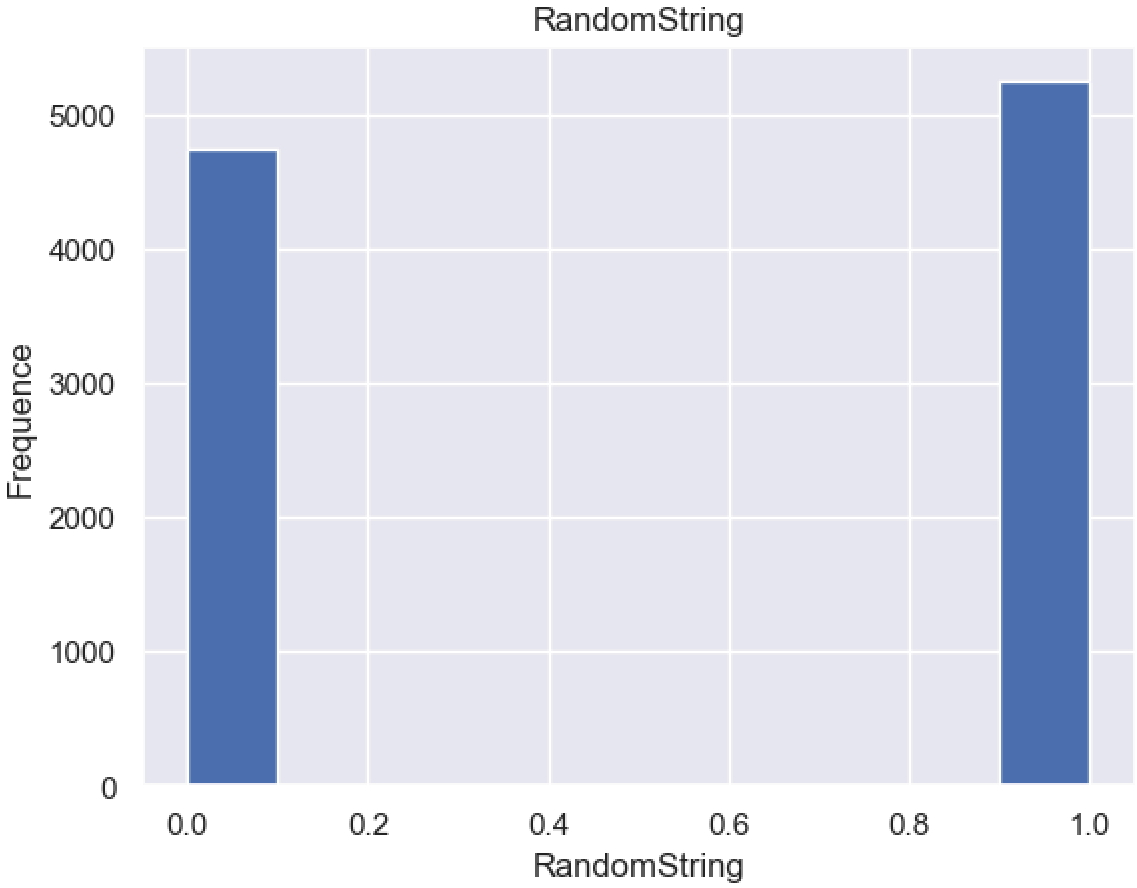

NumDashInHostname
Frequence
0
2000
4000
6000
8000
0
2
4
6
8
NumDashInHostname

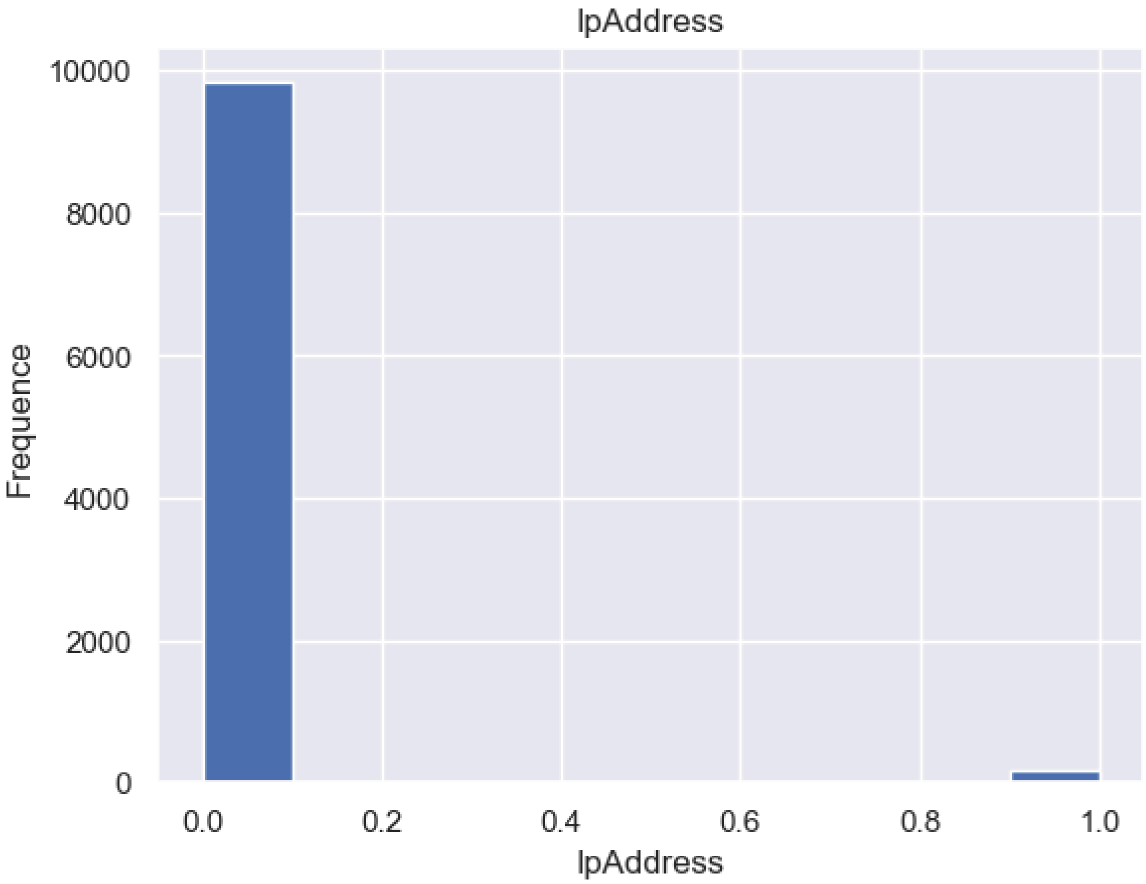

AtSymbol
Frequence
0
2000
4000
6000
8000
10000
0.0
0.2
0.4
0.6
0.8
1.0
AtSymbol

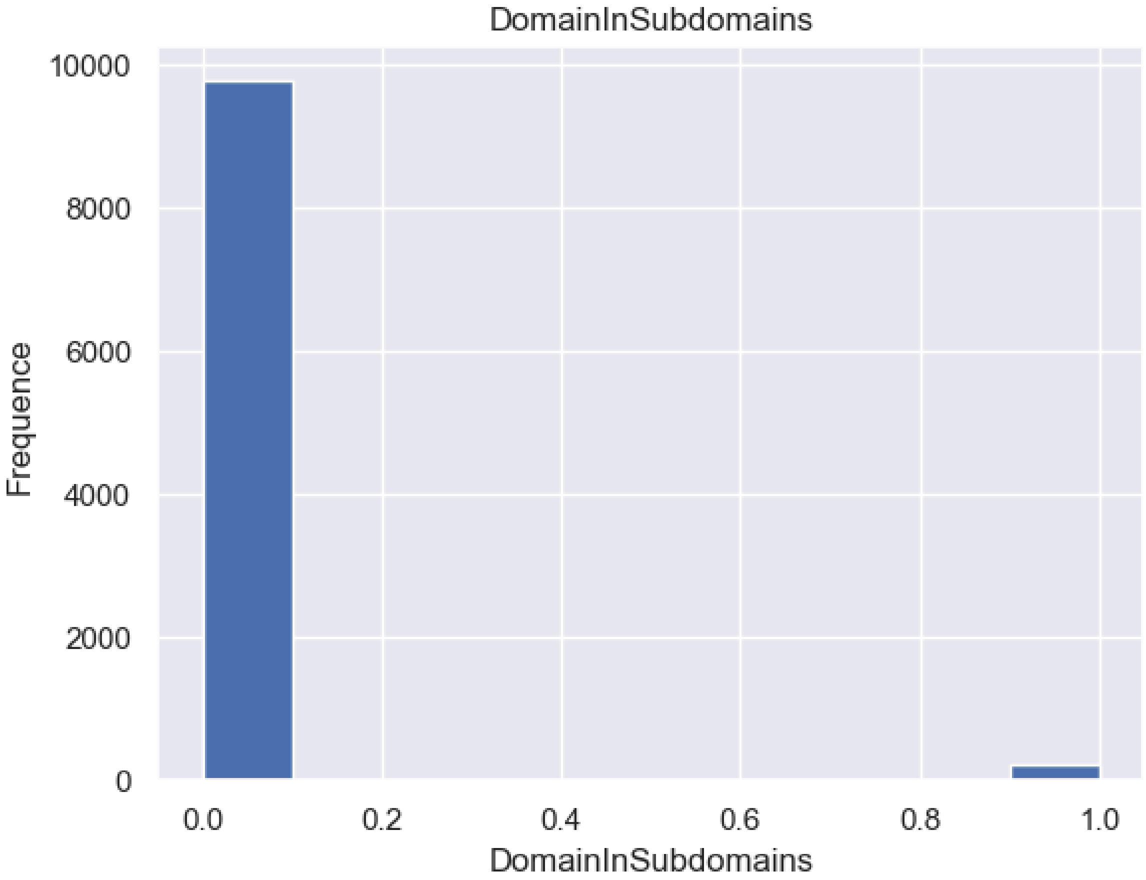

TildeSymbol
10000
8000
6000
4000
2000
0
Frequence
0.0
0.2
0.4
0.6
0.8
1.0
TildeSymbol

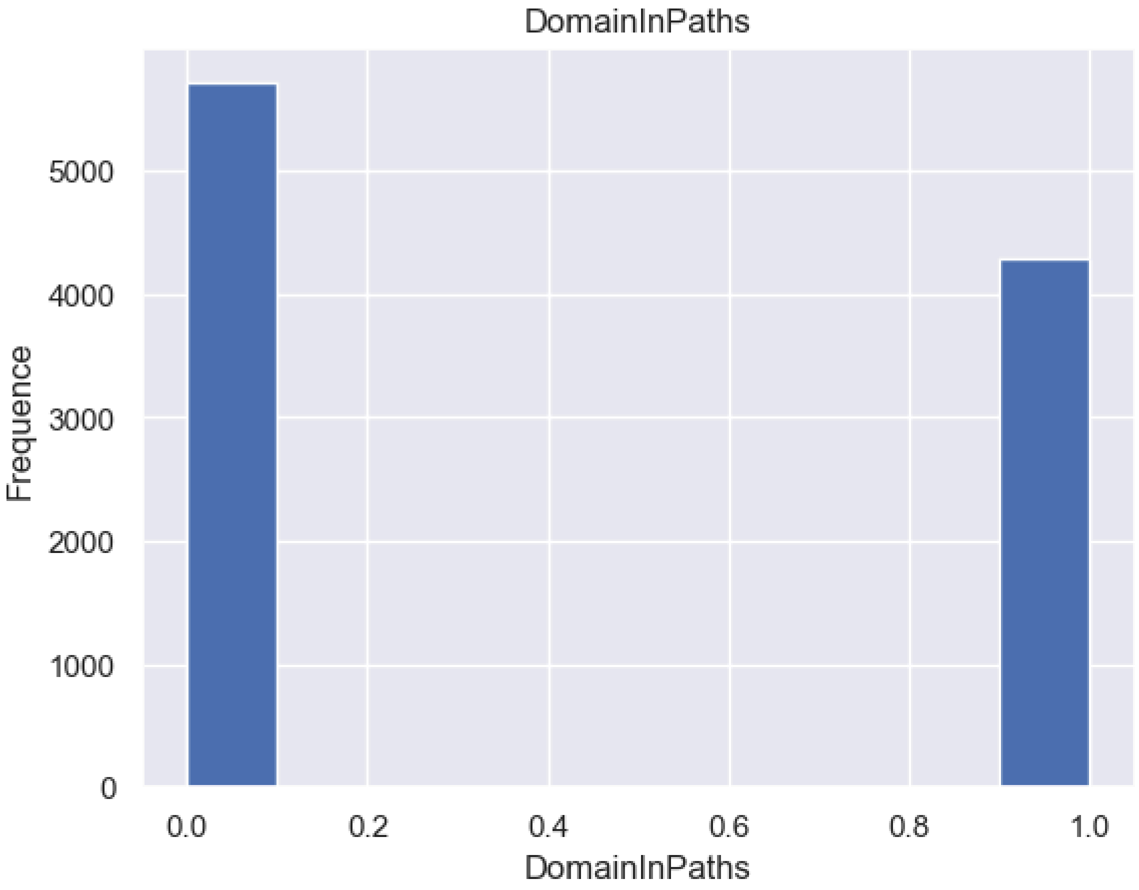

NumHash
10000
8000
6000
4000
2000
0
Frequence
0.0
0.2
0.4
0.6
0.8
1.0
NumHash

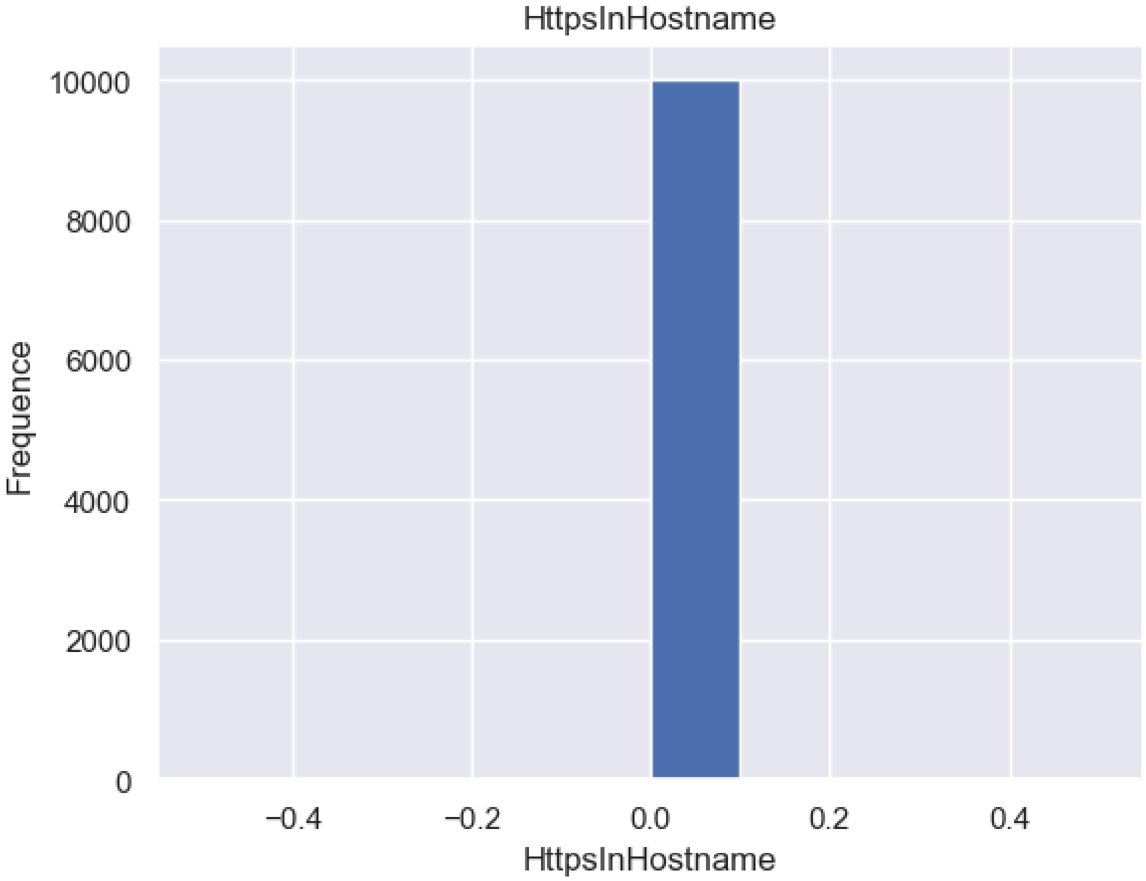

NoHttps
Frequence
0
2000
4000
6000
8000
10000
0.0
0.2
0.4
0.6
0.8
1.0
NoHttps

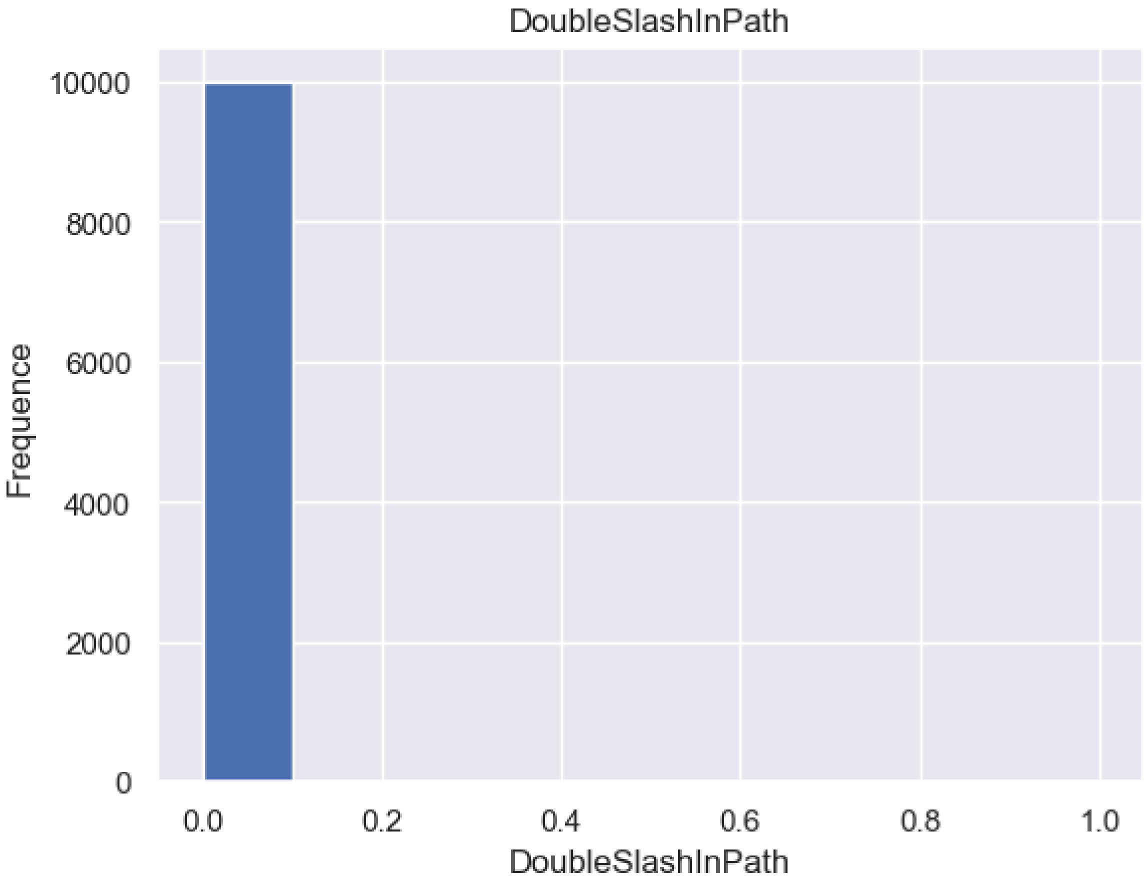

RandomString
Frequence
0
1000
2000
3000
4000
5000
0.0
0.2
0.4
0.6
0.8
1.0
RandomString

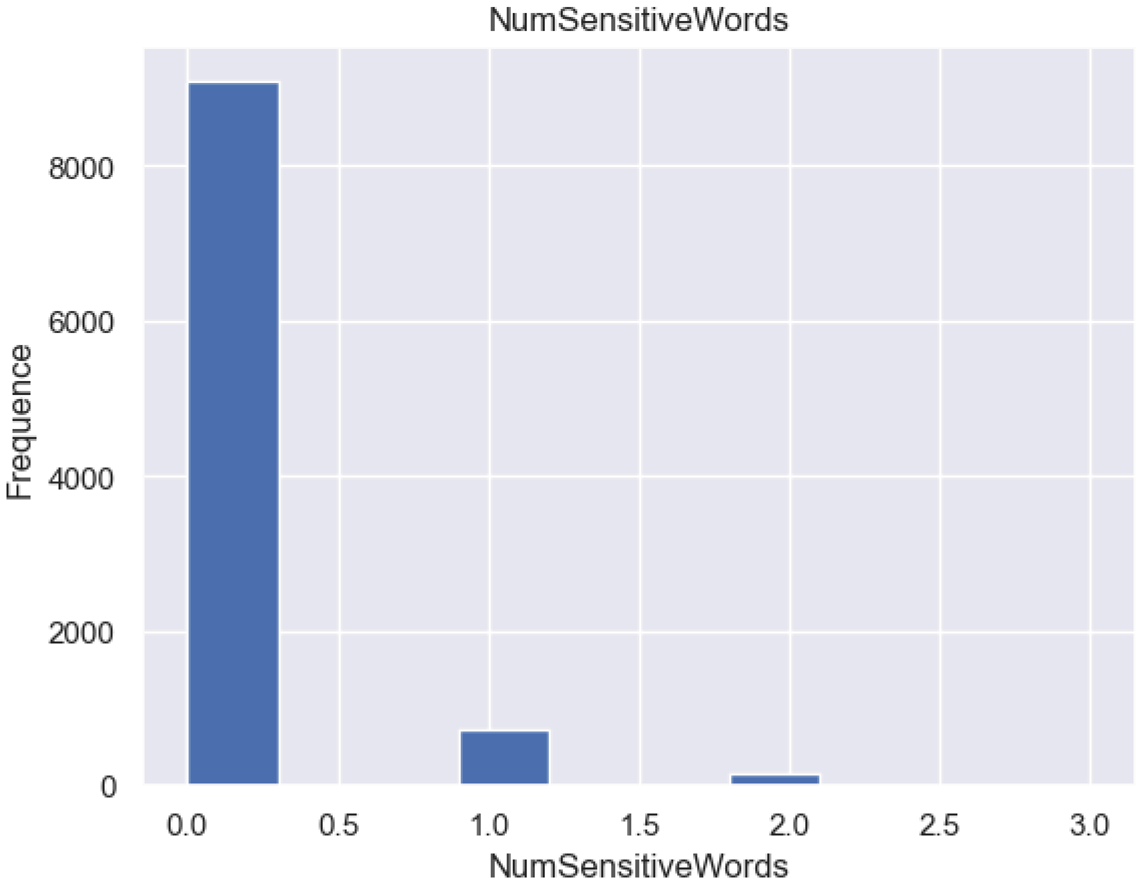

IpAddress
Frequence
10000
8000
6000
4000
2000
0
0.0
0.2
0.4
0.6
0.8
1.0
IpAddress

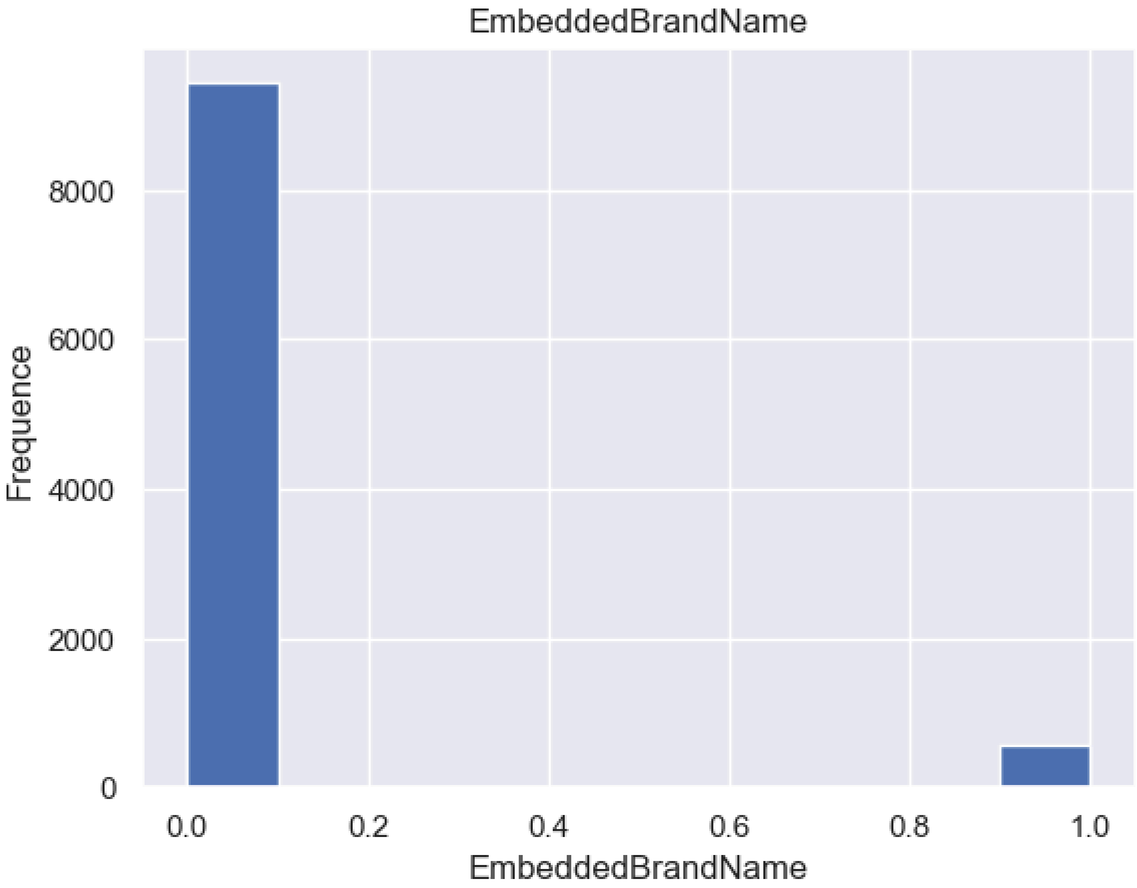

DomainInSubdomains
Frequence
10000
8000
6000
4000
2000
0
0.0
0.2
0.4
0.6
0.8
1.0
DomainInSubdomains


DomainInPaths
Frequence
0
1000
2000
3000
4000
5000
0.0
0.2
0.4
0.6
0.8
1.0
DomainInPaths

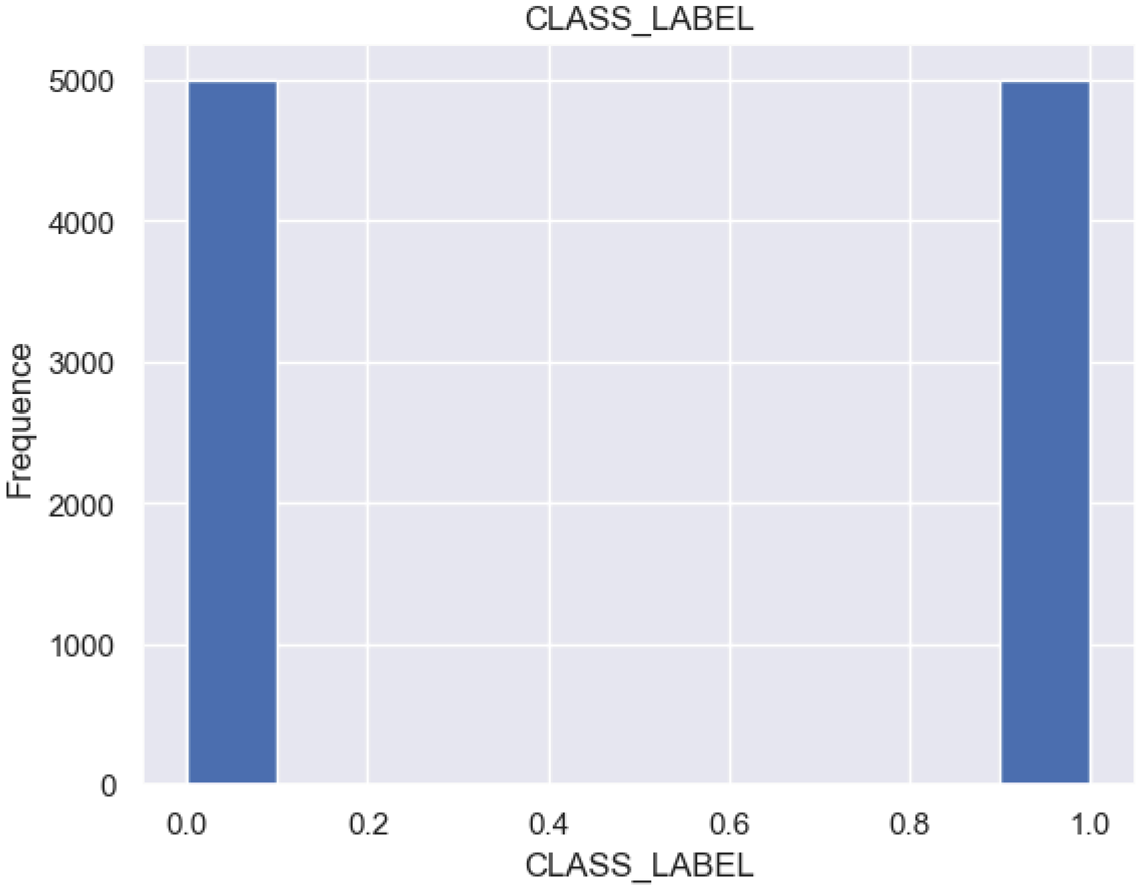

HttpsInHostname
Frequence
0
2000
4000
6000
8000
10000
−0.4
−0.2
0.0
0.2
0.4
HttpsInHostname


DoubleSlashInPath
Frequence
0
2000
4000
6000
8000
10000
0.0
0.2
0.4
0.6
0.8
1.0
DoubleSlashInPath

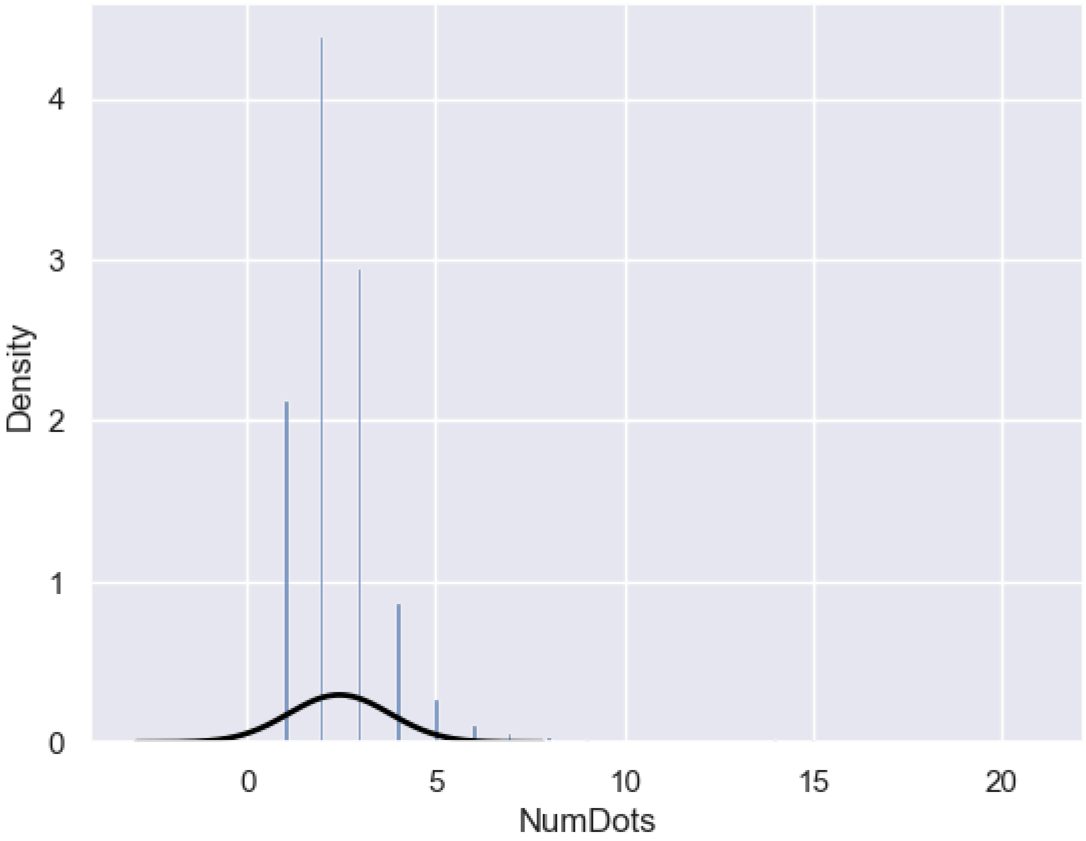

NumSensitiveWords
Frequence
0
2000
4000
6000
8000
0.0
0.5
1.0
1.5
2.0
2.5
3.0
NumSensitiveWords

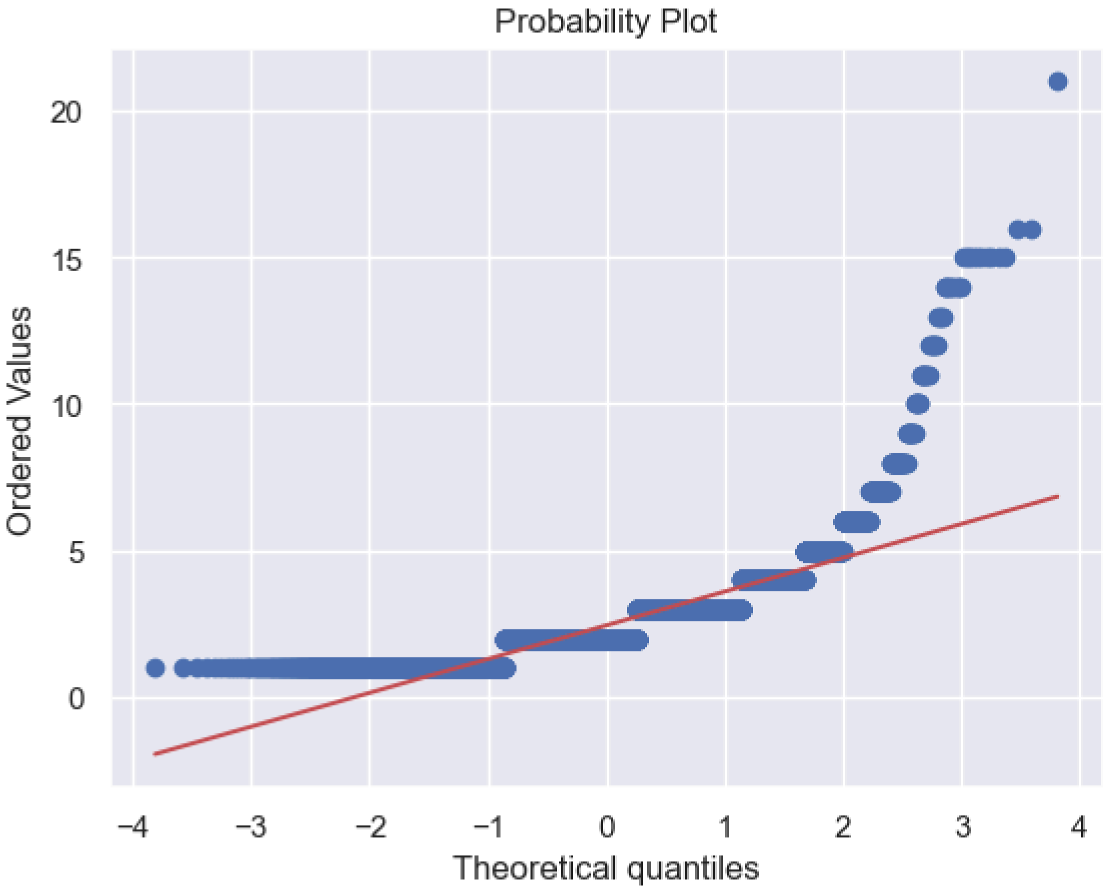


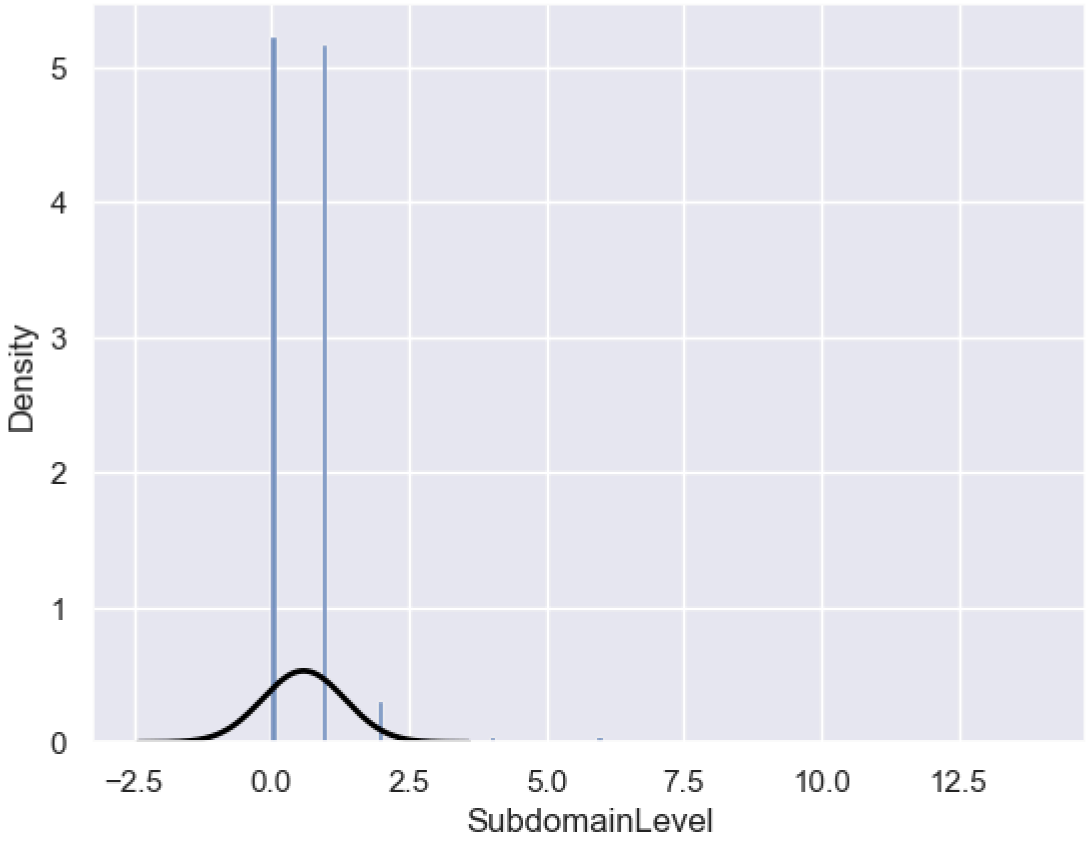
EmbeddedBrandName

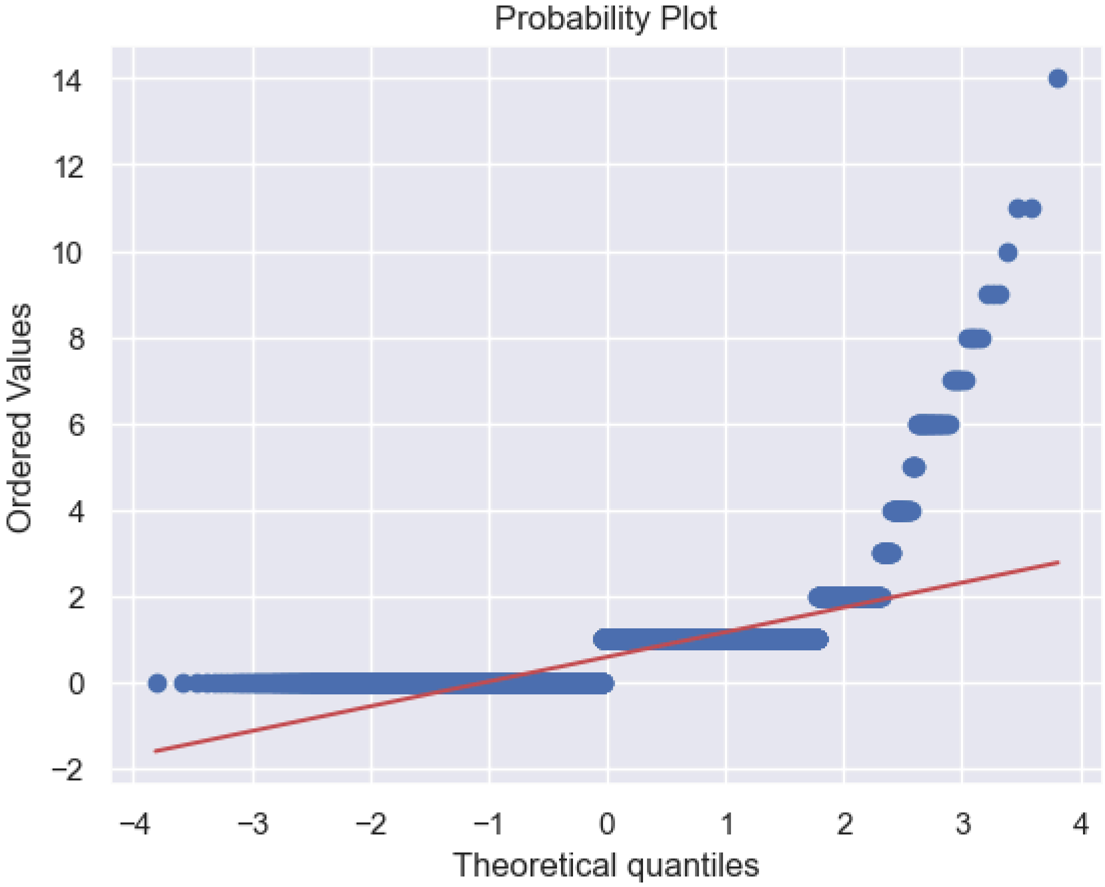


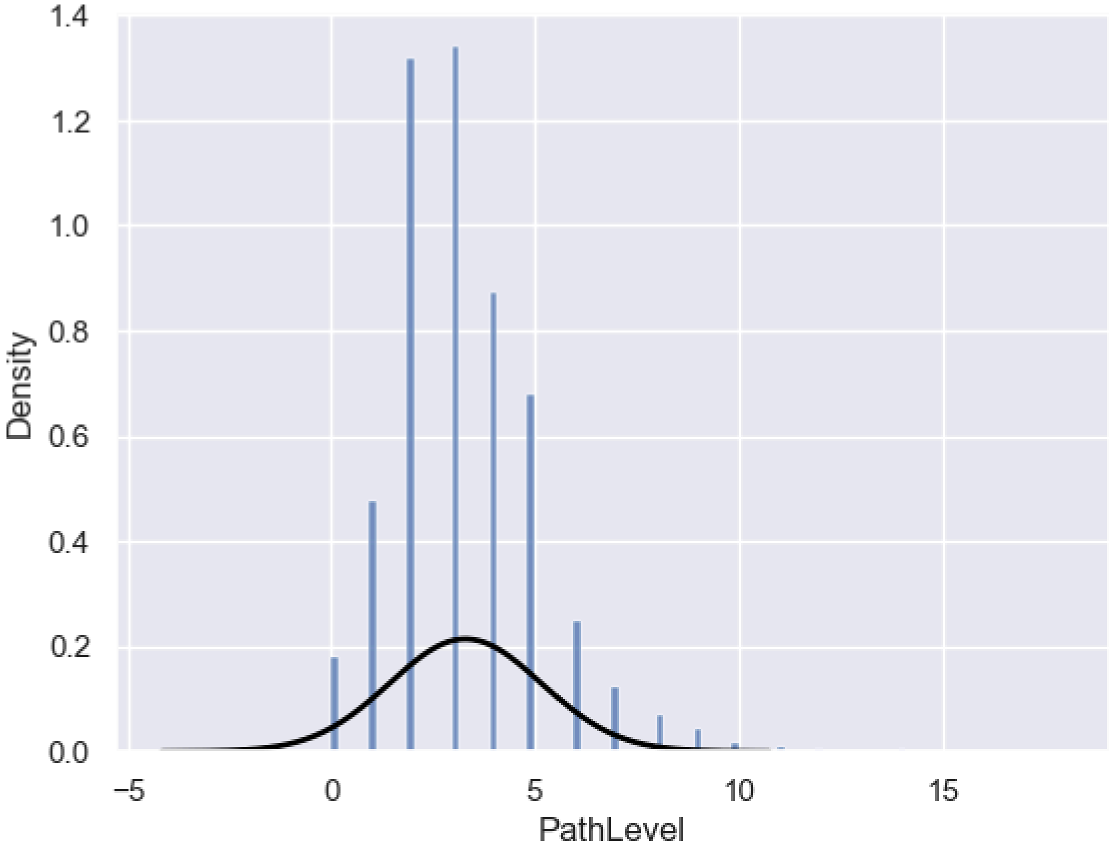
CLASS_LABEL

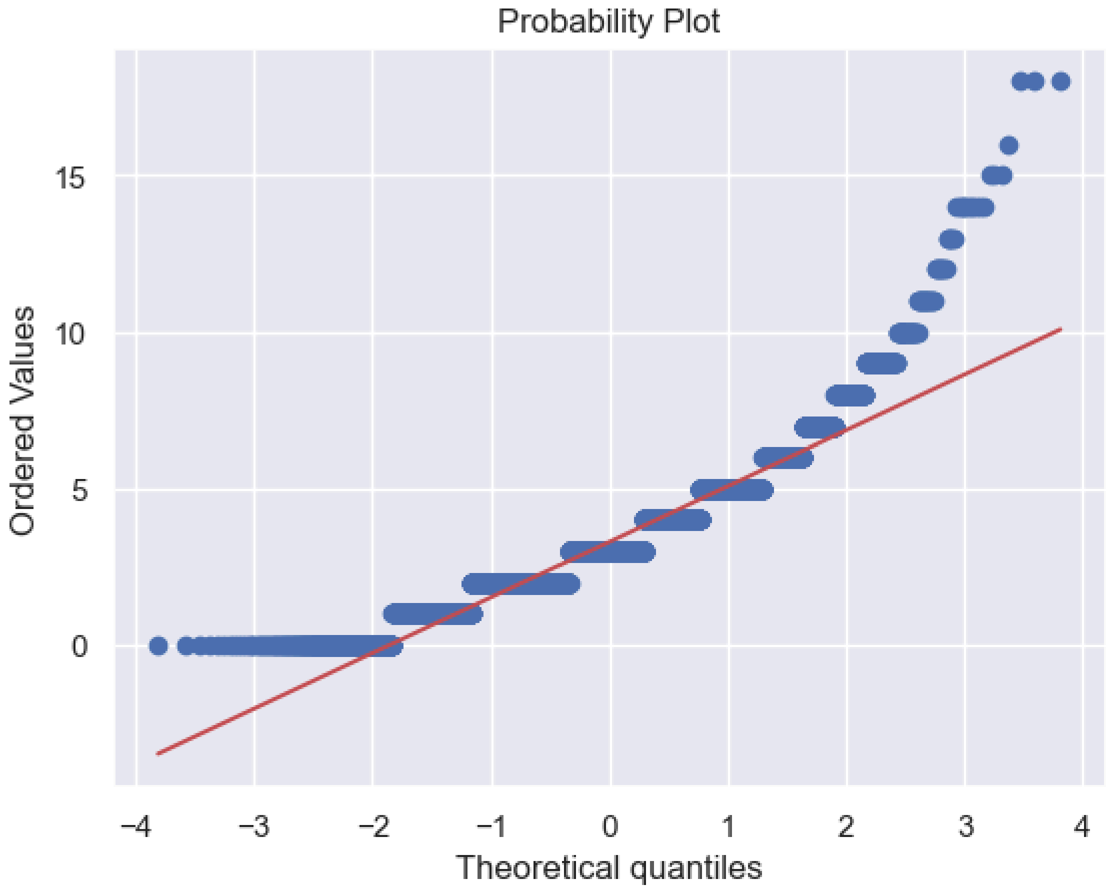

Density
0
1
2
3
4
0
5
10
15
20
NumDots

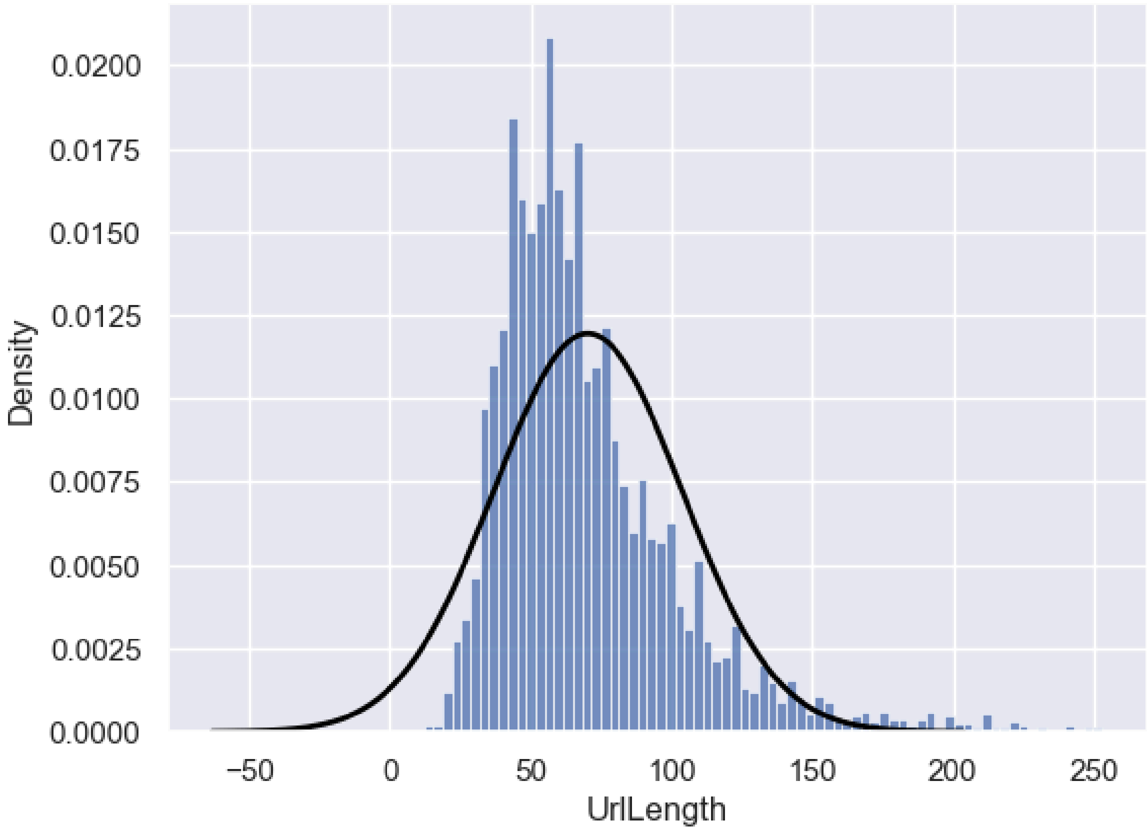

Probability Plot
Ordered Values
0
5
10
15
20
−4
−3
−2
−1
0
1
2
3
4
Theoretical quantiles

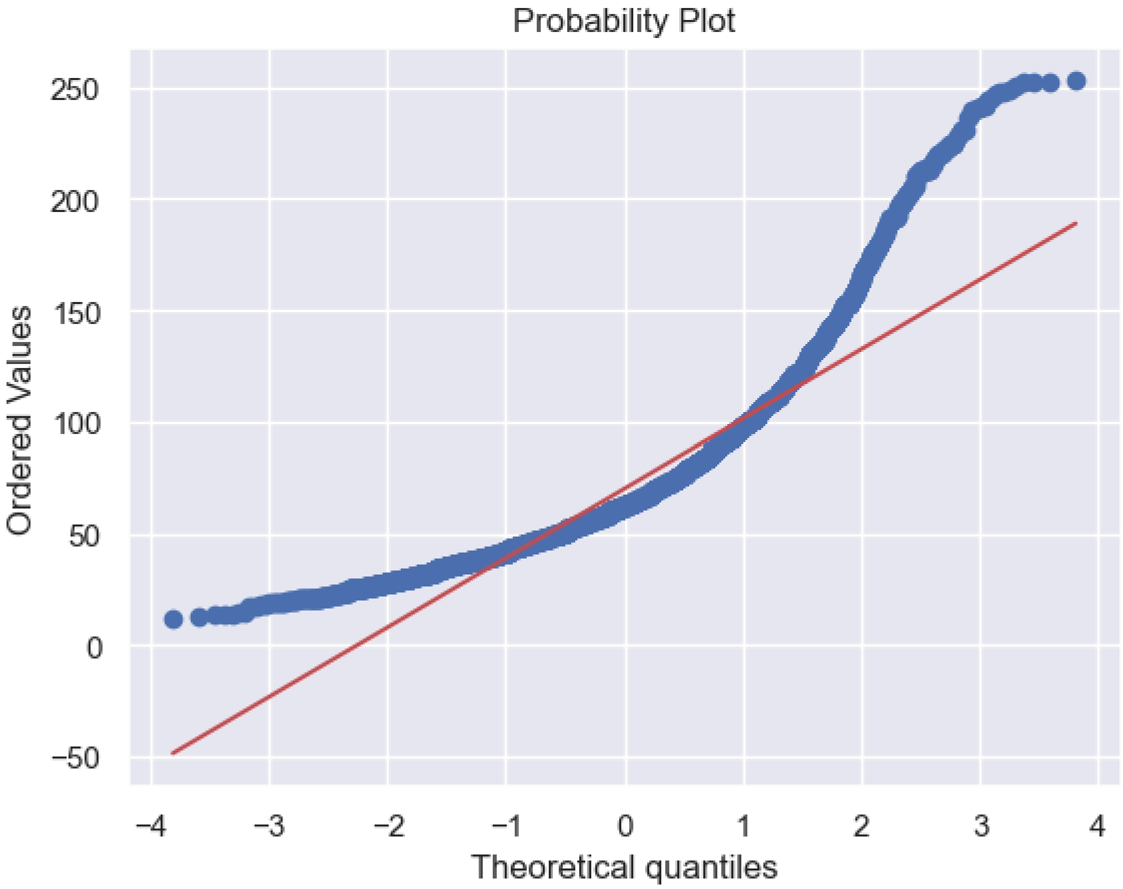

Density
5
4
3
2
1
0
−2.5
0.0
2.5
5.0
7.5
10.0
12.5
SubdomainLevel

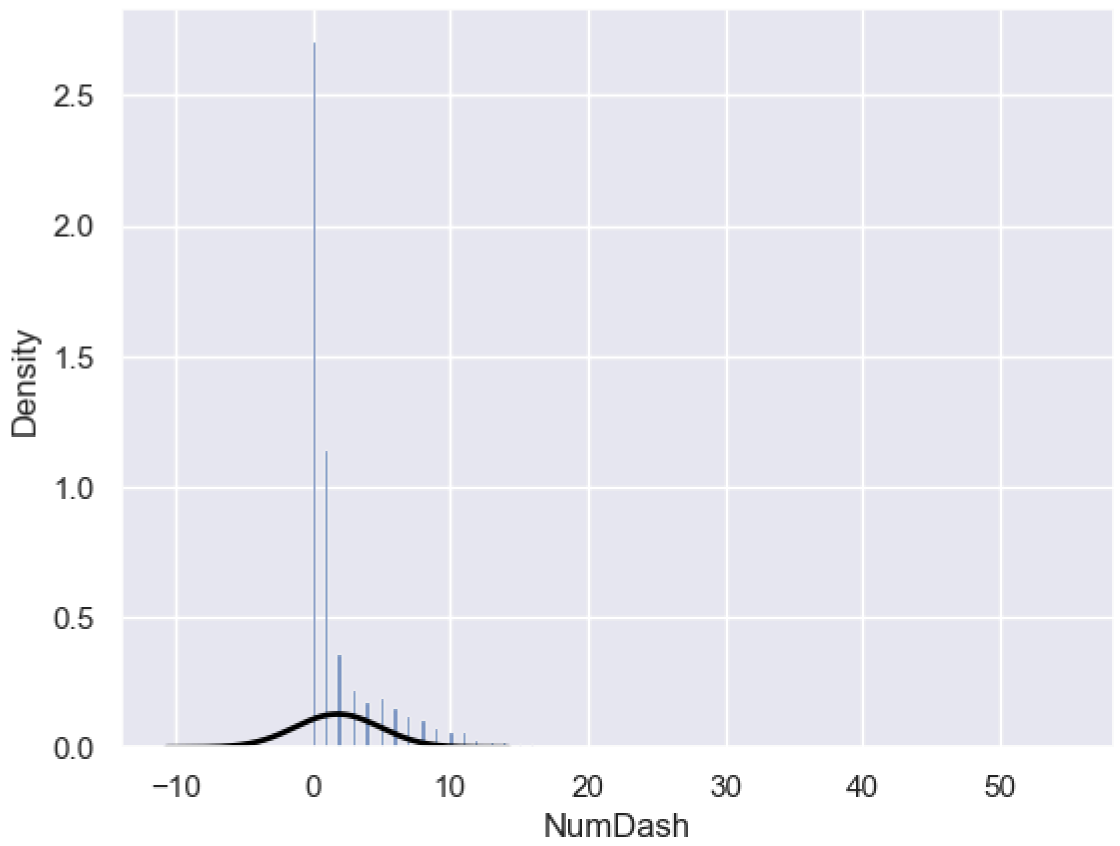

Probability Plot
Ordered Values
14
12
10
8
6
4
2
0
−2
−4
−3
−2
−1
0
1
2
3
4
Theoretical quantiles

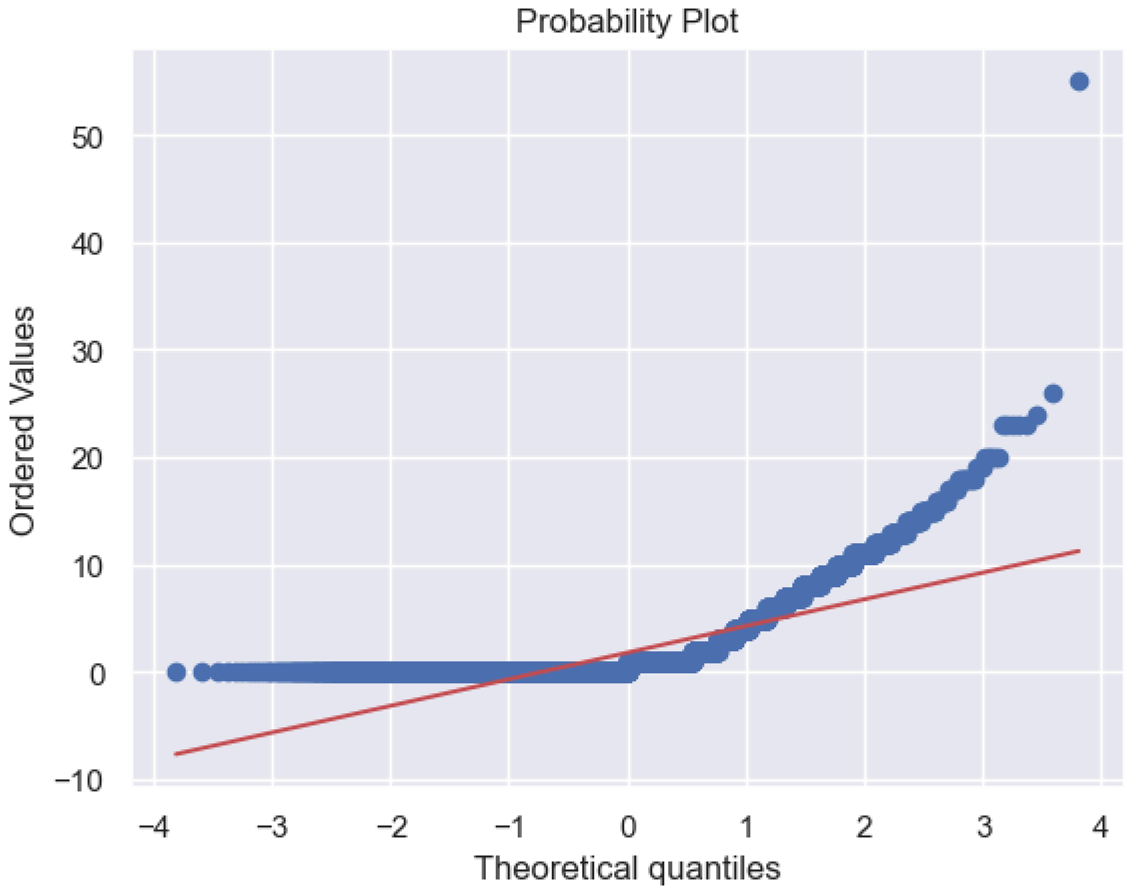

Density
1.4
1.2
1.0
0.8
0.6
0.4
0.2
0.0
−5
0
5
10
15
PathLevel

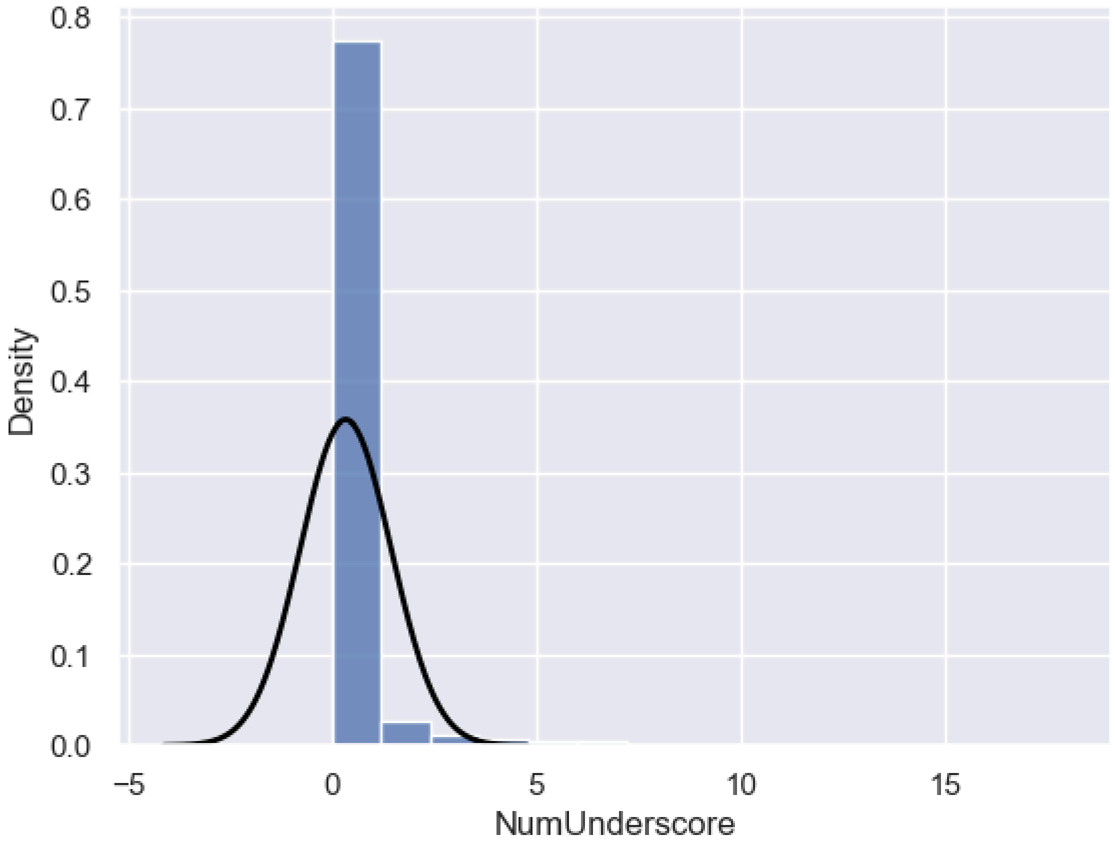

Probability Plot
Ordered Values
15
10
5
0
−4
−3
−2
−1
0
1
2
3
4
Theoretical quantiles

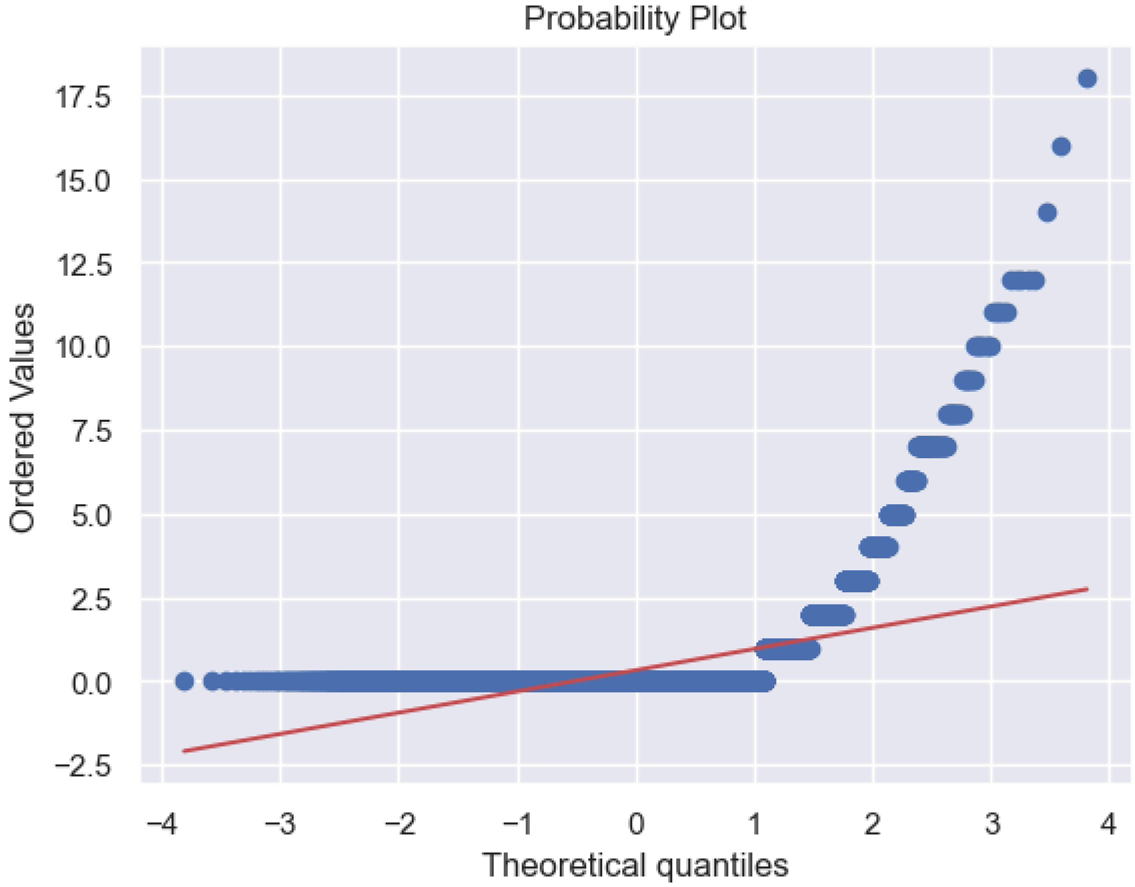

0.0200
0.0175
0.0150
0.0125
0.0100
0.0075
0.0050
0.0025
0.0000
Density
−50
0
50
100
150
200
250
UrlLength

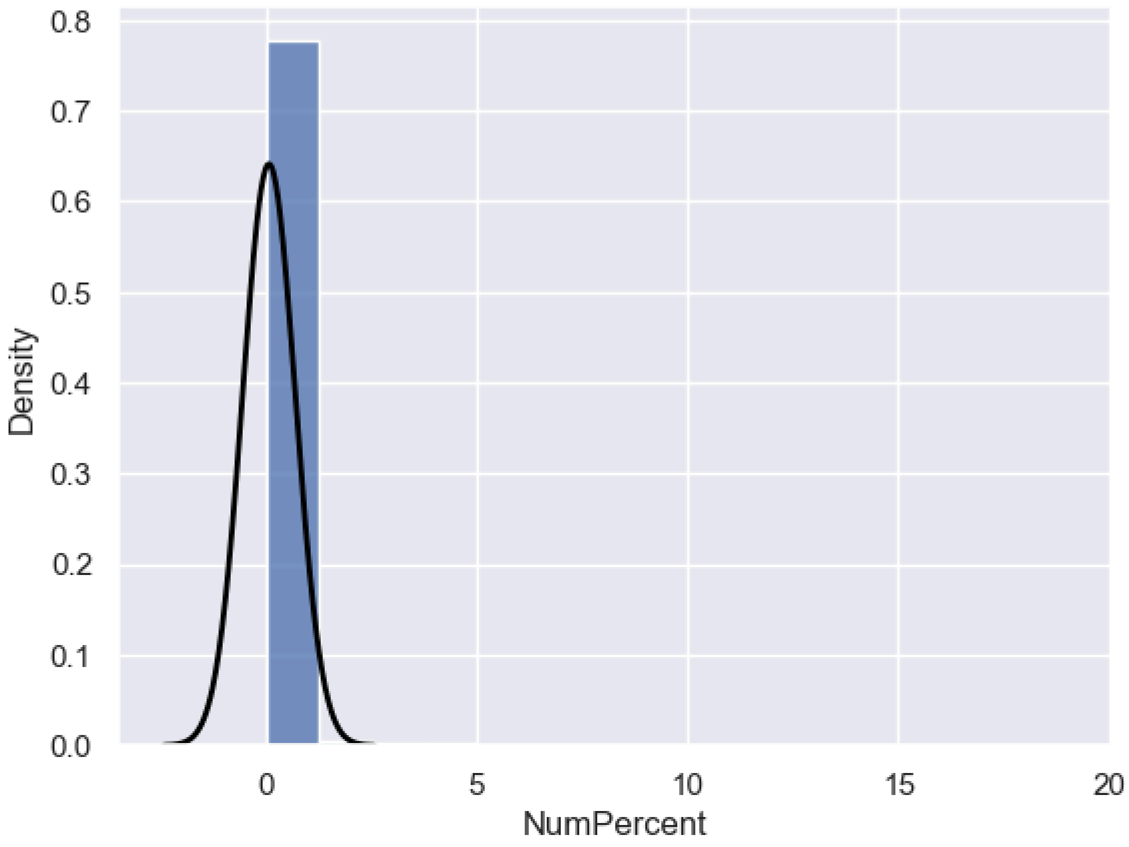

Probability Plot
250
200
150
100
50
0
−50
Ordered Values
−4
−3
−2
−1
0
1
2
3
4
Theoretical quantiles

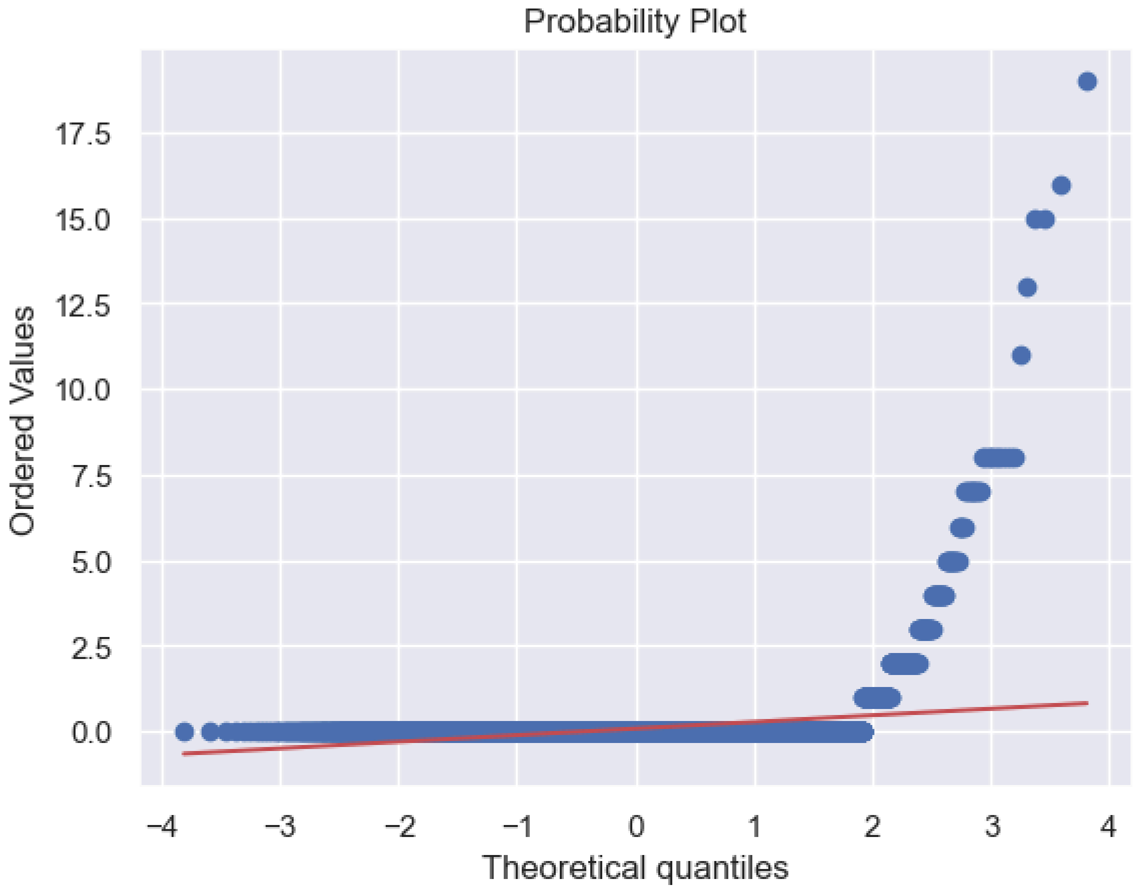

Density
2.5
2.0
1.5
1.0
0.5
0.0
−10
0
10
20
30
40
50
NumDash

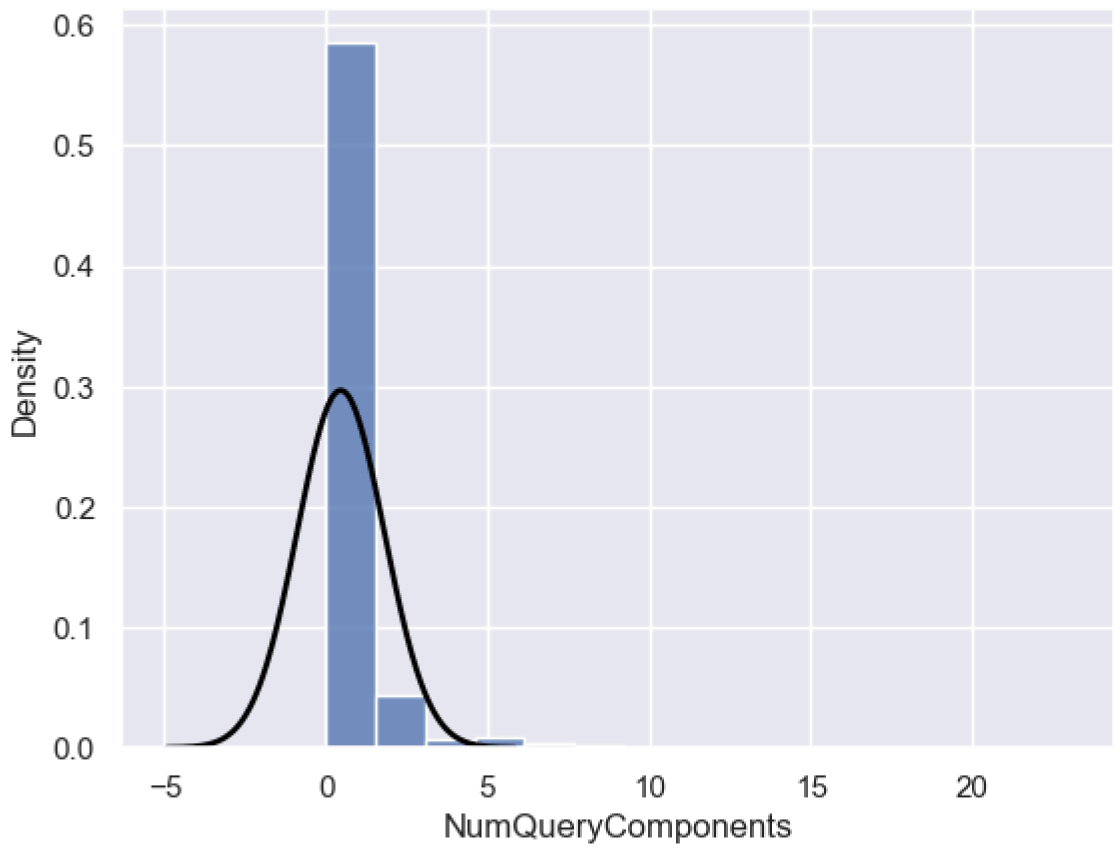

Probability Plot
Ordered Values
50
40
30
20
10
0
−10
−4
−3
−2
−1
0
1
2
3
4
Theoretical quantiles

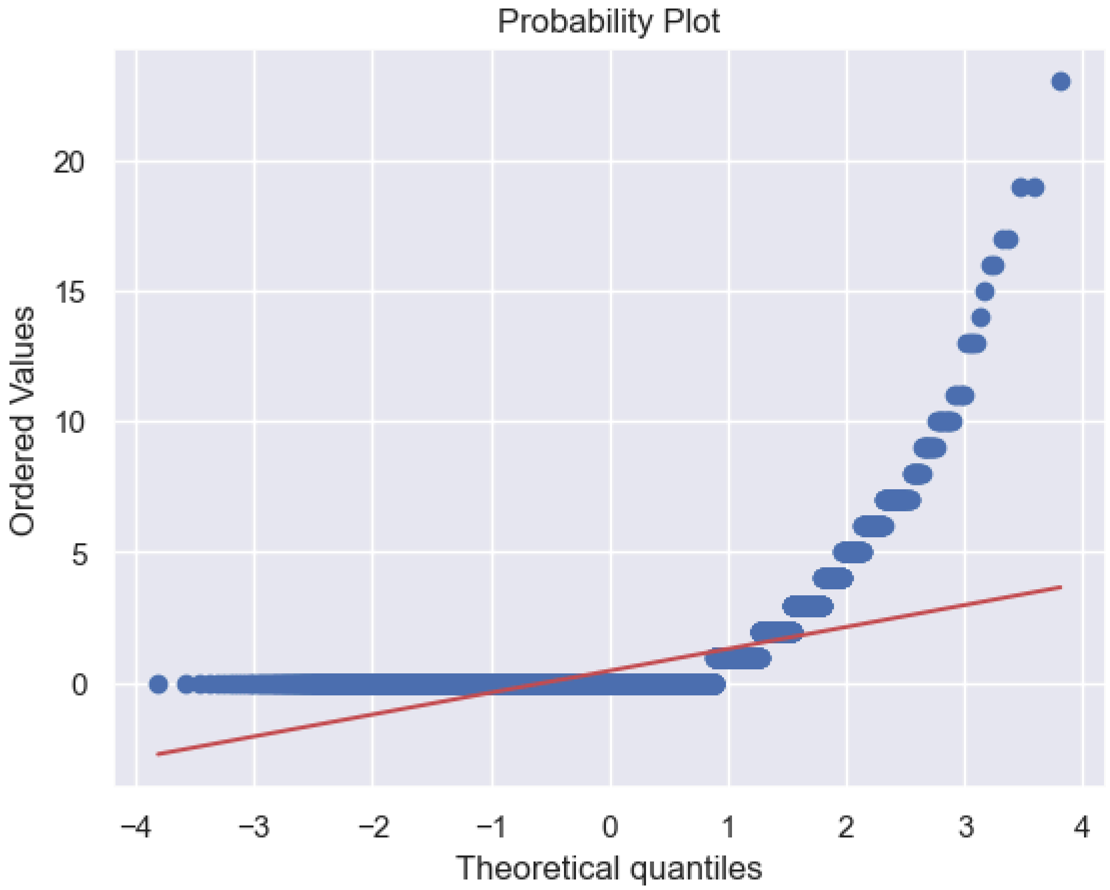

0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0.0
Density
−5
0
5
10
15
NumUnderscore

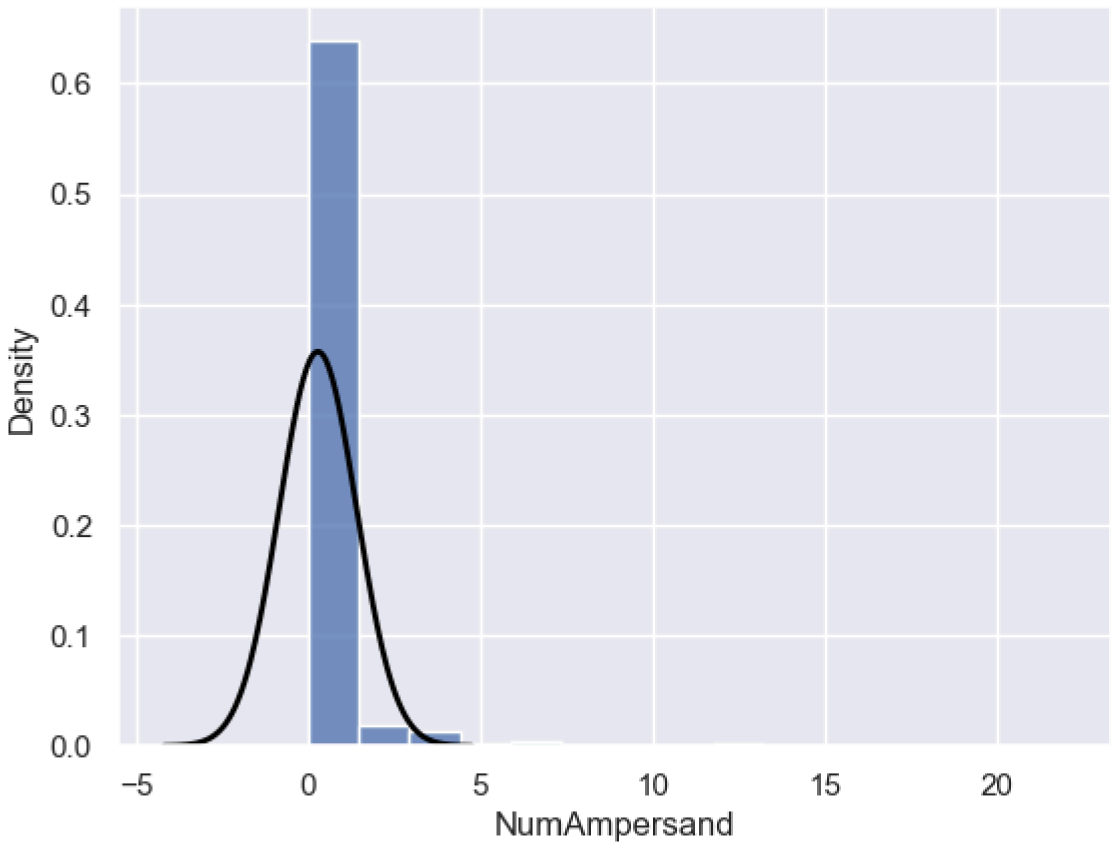

Probability Plot
17.5
15.0
12.5
10.0
7.5
5.0
2.5
0.0
−2.5
Ordered Values
−4
−3
−2
−1
0
1
2
3
4
Theoretical quantiles

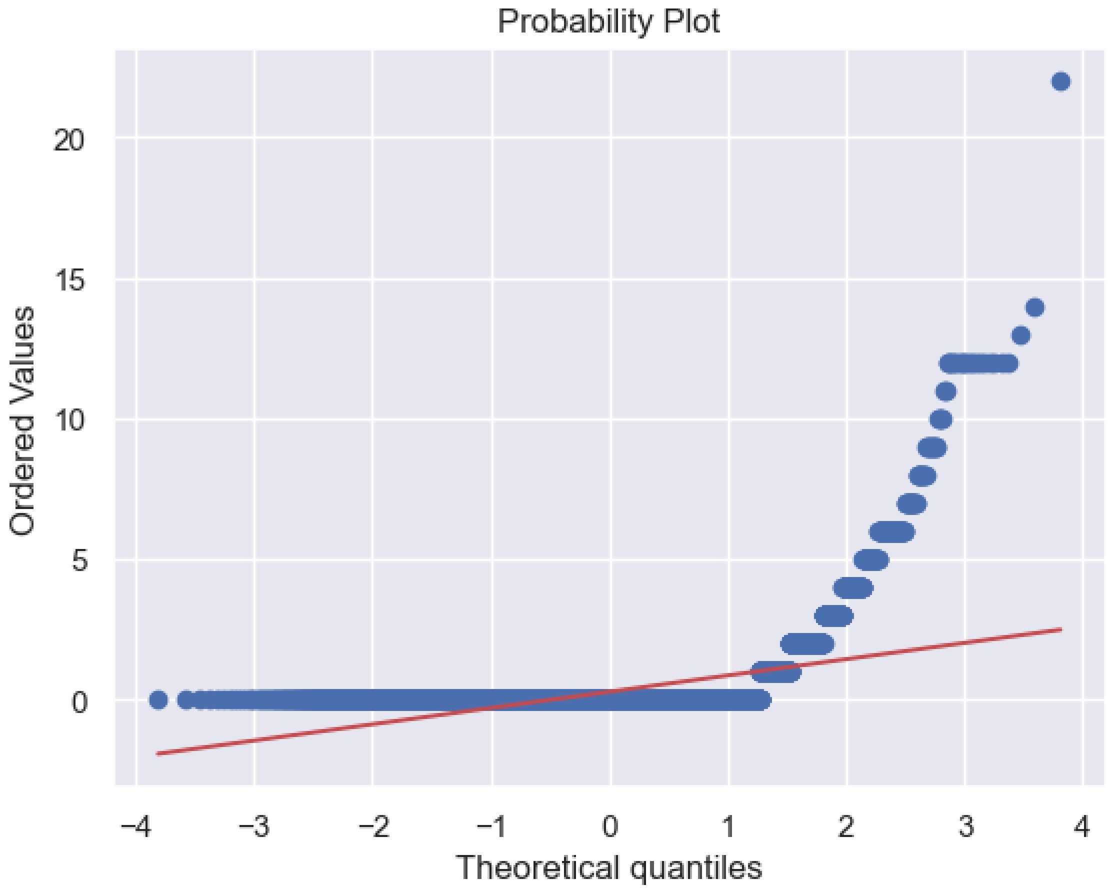

Density
0.0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0
5
10
15
20
NumPercent

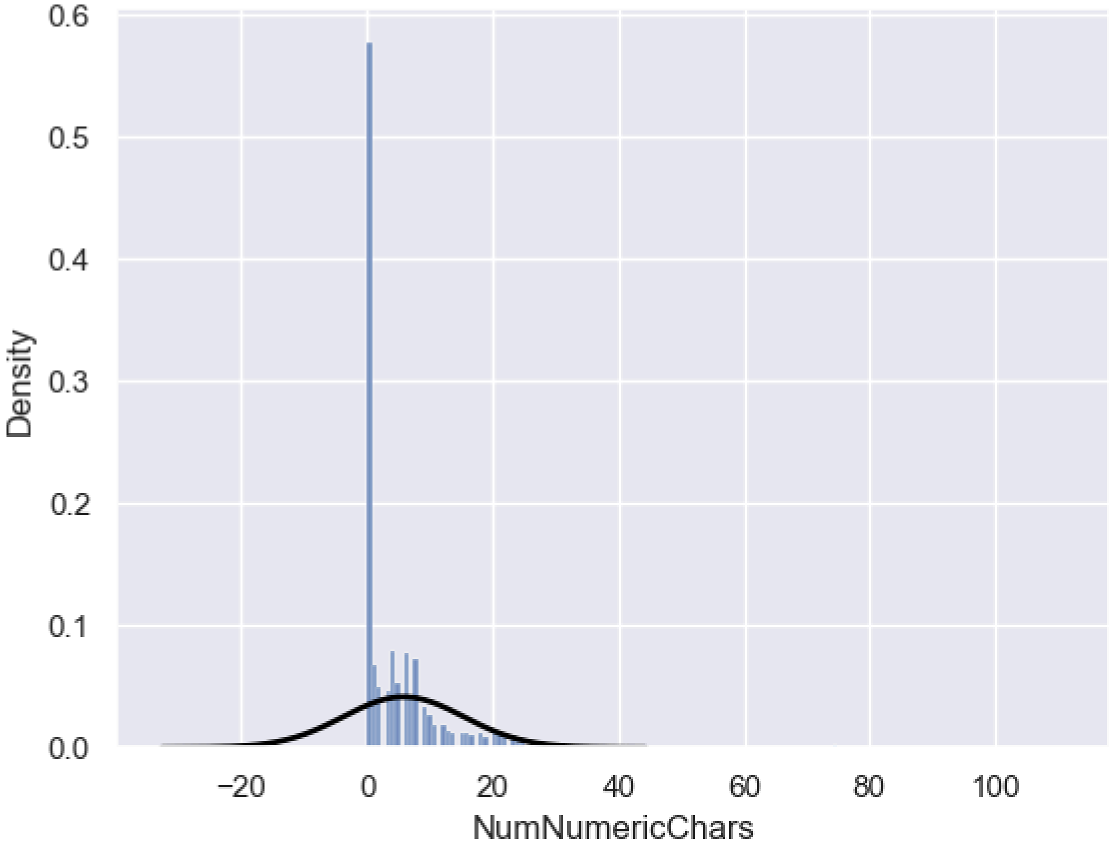

Probability Plot
Ordered Values
0.0
2.5
5.0
7.5
10.0
12.5
15.0
17.5
−4
−3
−2
−1
0
1
2
3
4
Theoretical quantiles

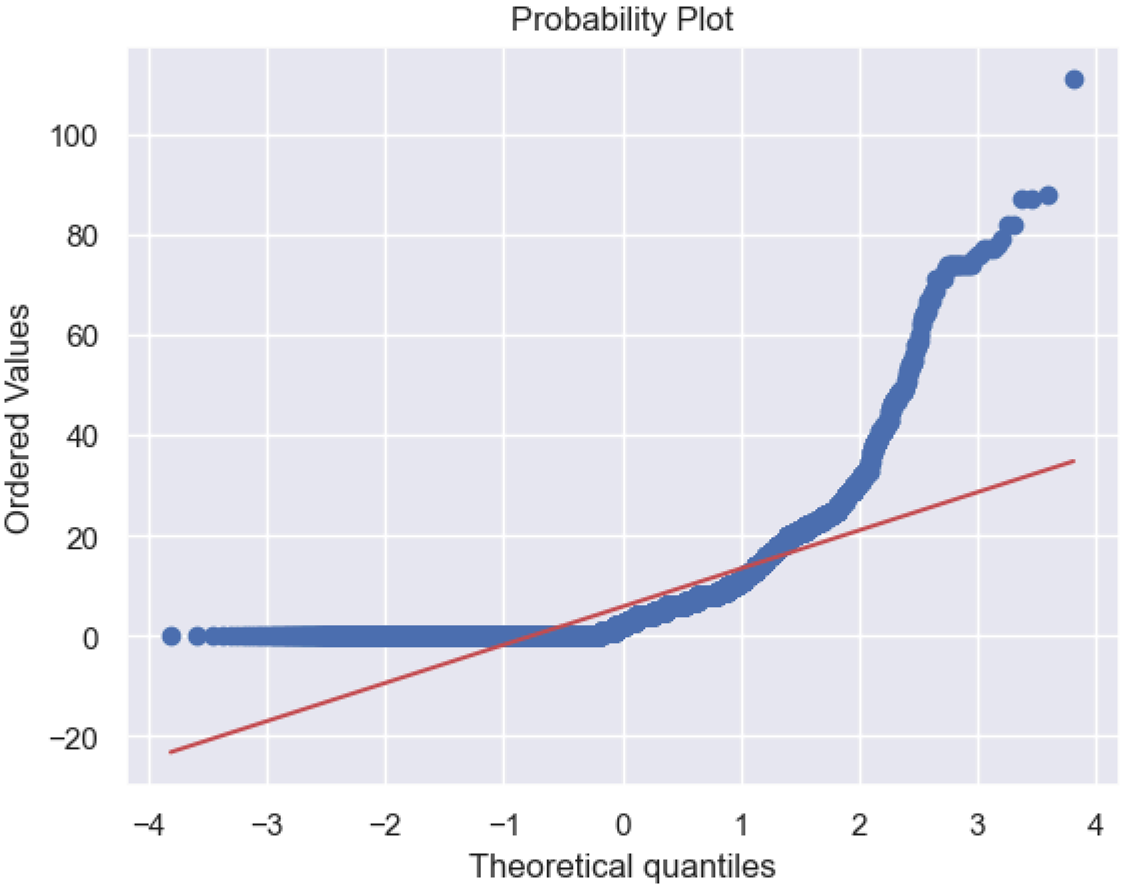

0.6
0.5
0.4
0.3
0.2
0.1
0.0
Density
−5
0
5
10
15
20
NumQueryComponents

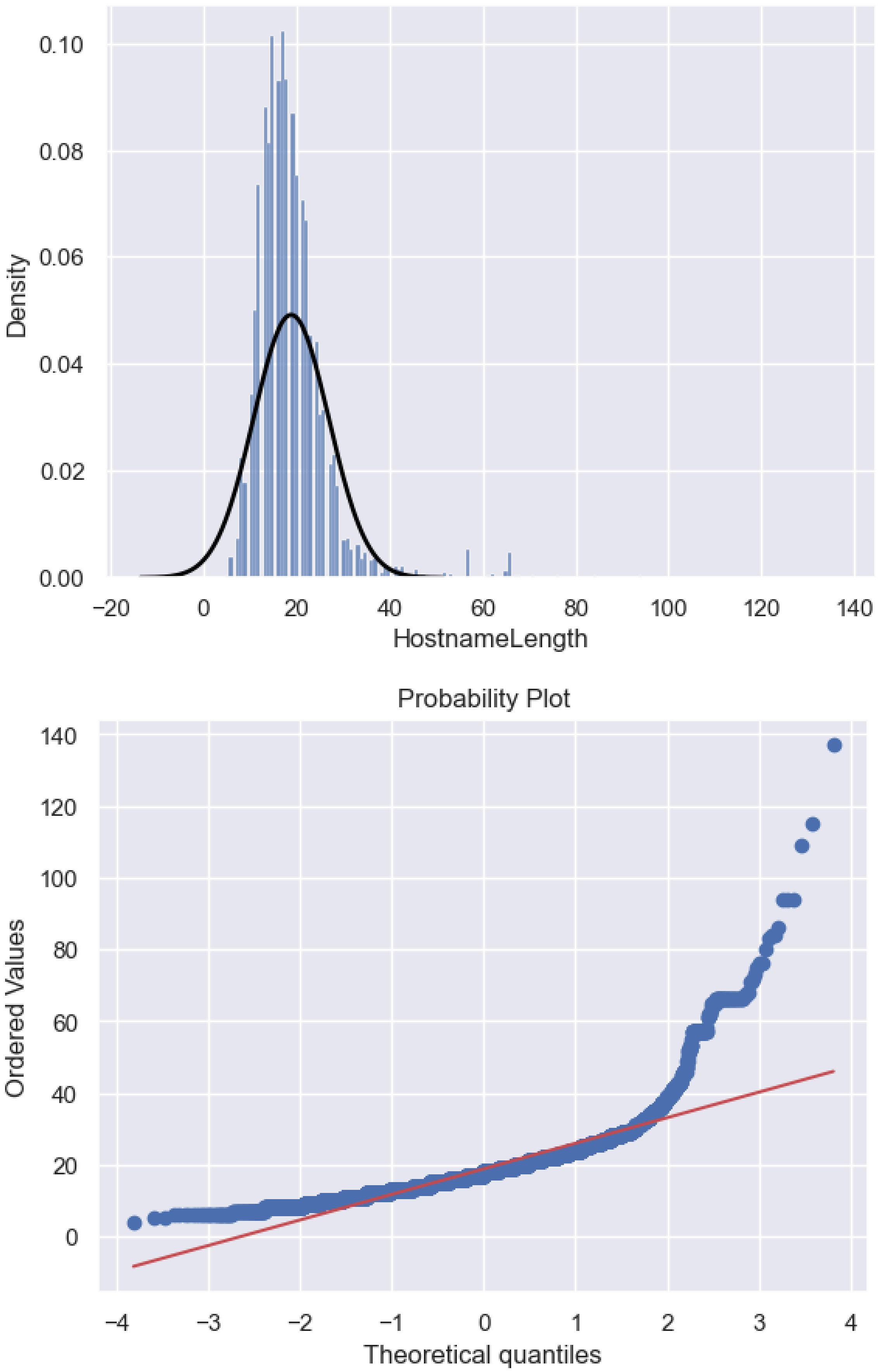

Probability Plot
20
15
10
5
0
Ordered Values
−4
−3
−2
−1
0
1
2
3
4
Theoretical quantiles

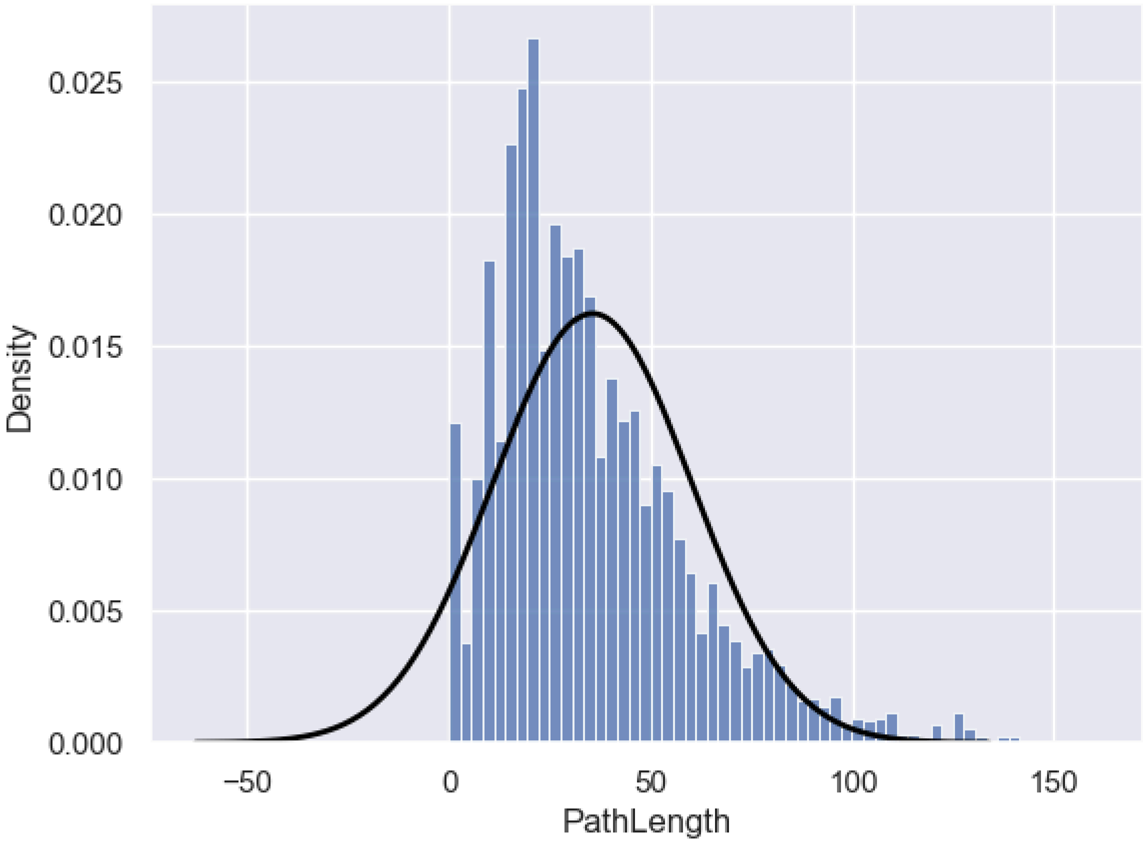

Density
0.0
0.1
0.2
0.3
0.4
0.5
0.6
−5
0
5
10
15
20
NumAmpersand

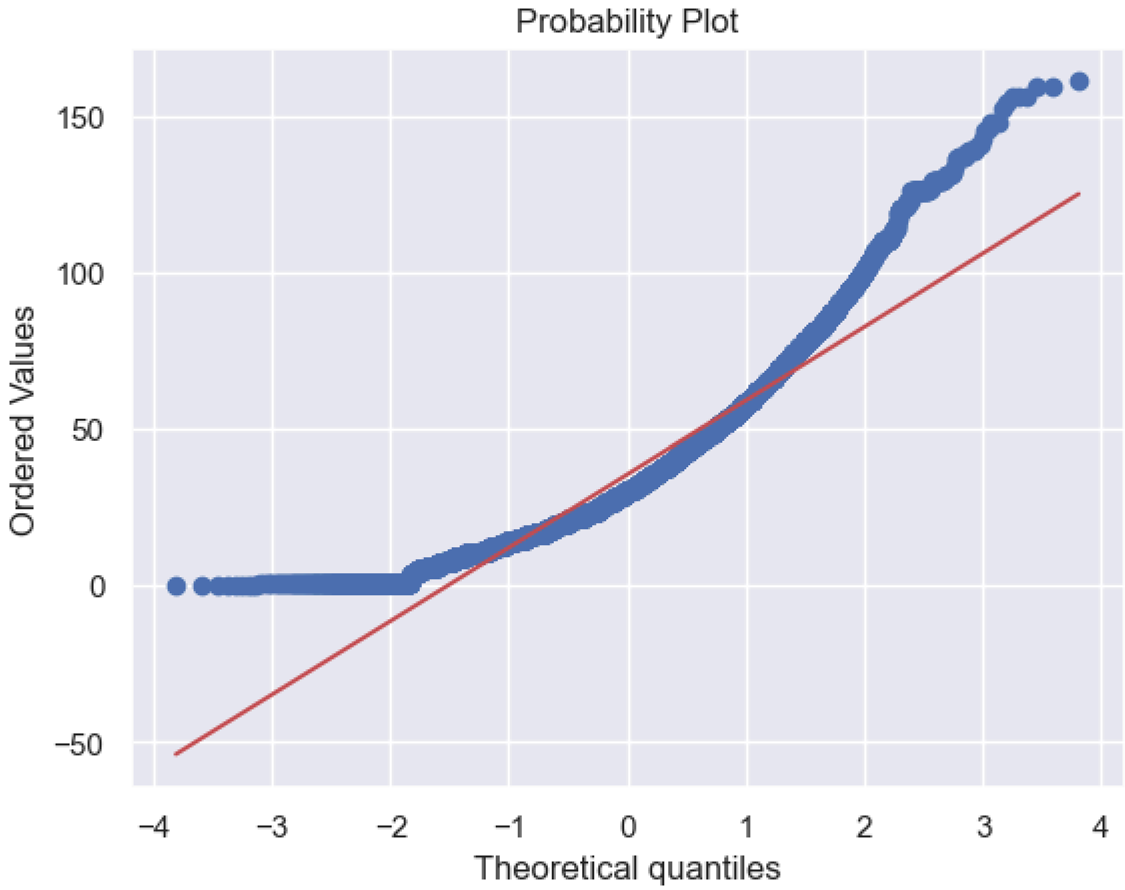

Probability Plot
Ordered Values
0
5
10
15
20
−4
−3
−2
−1
0
1
2
3
4
Theoretical quantiles

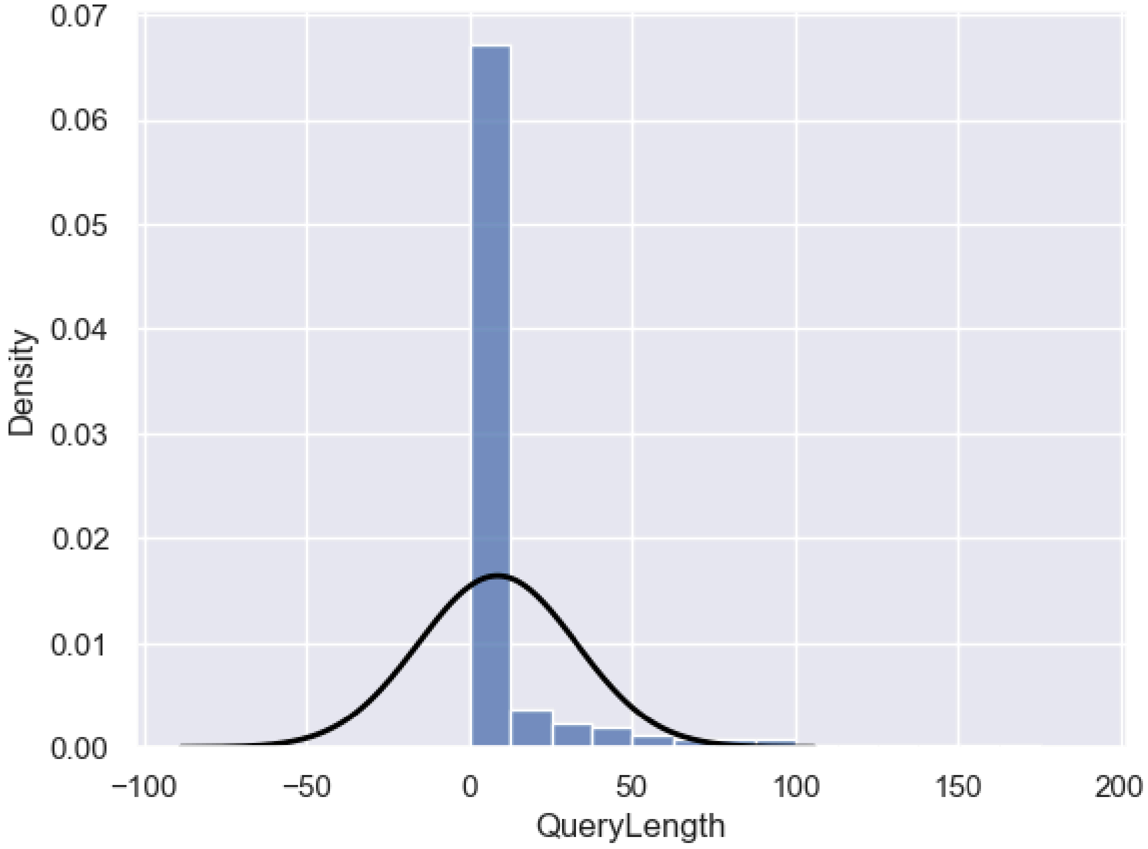

0.6
0.5
0.4
0.3
0.2
0.1
0.0
Density
−20
0
20
40
60
80
100
NumNumericChars

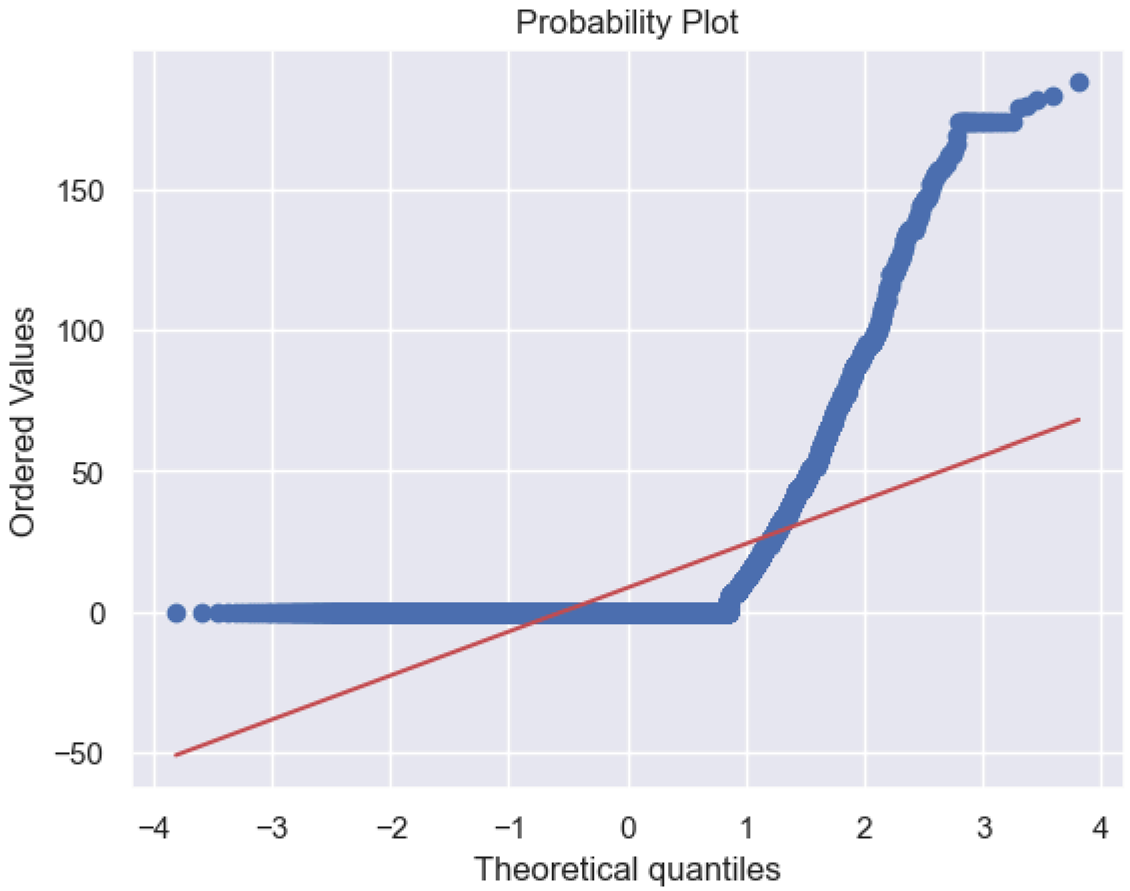

Probability Plot
100
80
60
40
20
0
−20
Ordered Values
−4
−3
−2
−1
0
1
2
3
4
Theoretical quantiles


0.10
0.08
0.06
0.04
0.02
0.00
Density
−20
0
20
40
60
80
100
120
140
HostnameLength
Probability Plot
140
120
100
80
60
40
20
0
Ordered Values
−4
−3
−2
−1
0
1
2
3
4
Theoretical quantiles


0.025
0.020
0.015
0.010
0.005
0.000
Density
−50
0
50
100
150
PathLength


Probability Plot
150
100
50
0
−50
Ordered Values
−4
−3
−2
−1
0
1
2
3
4
Theoretical quantiles



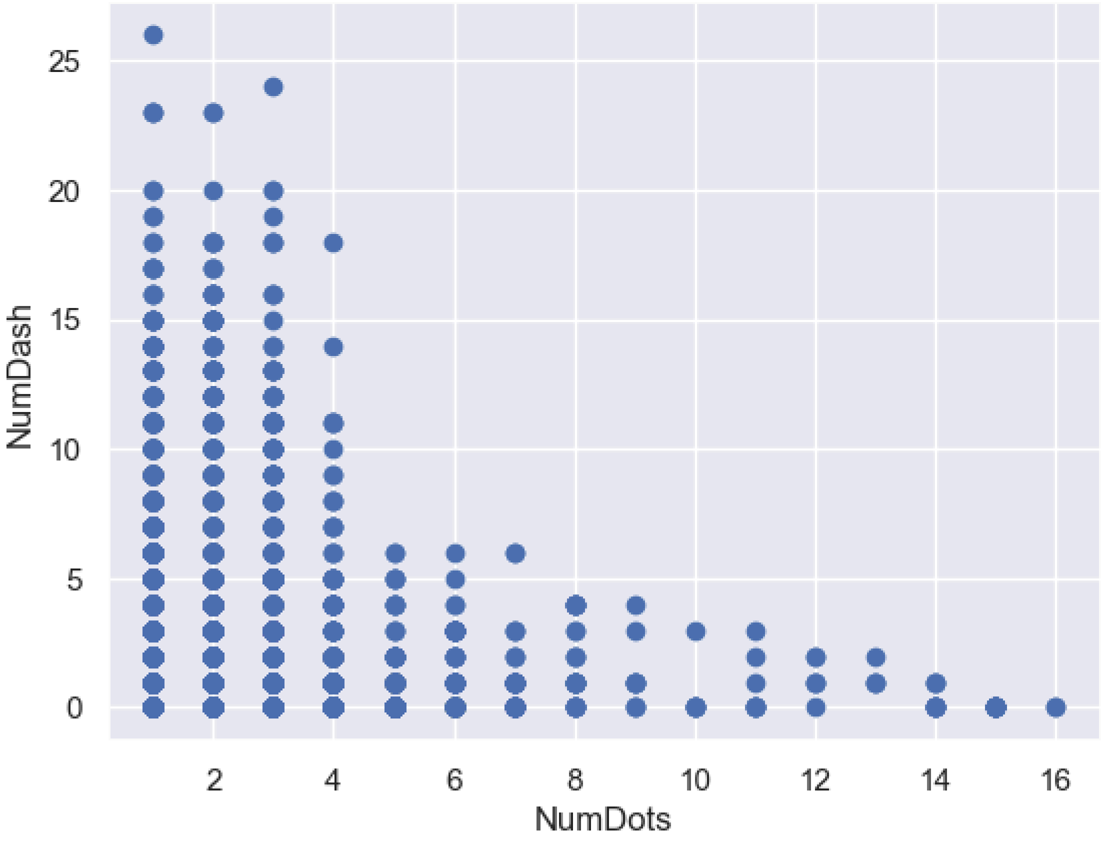

Probability Plot

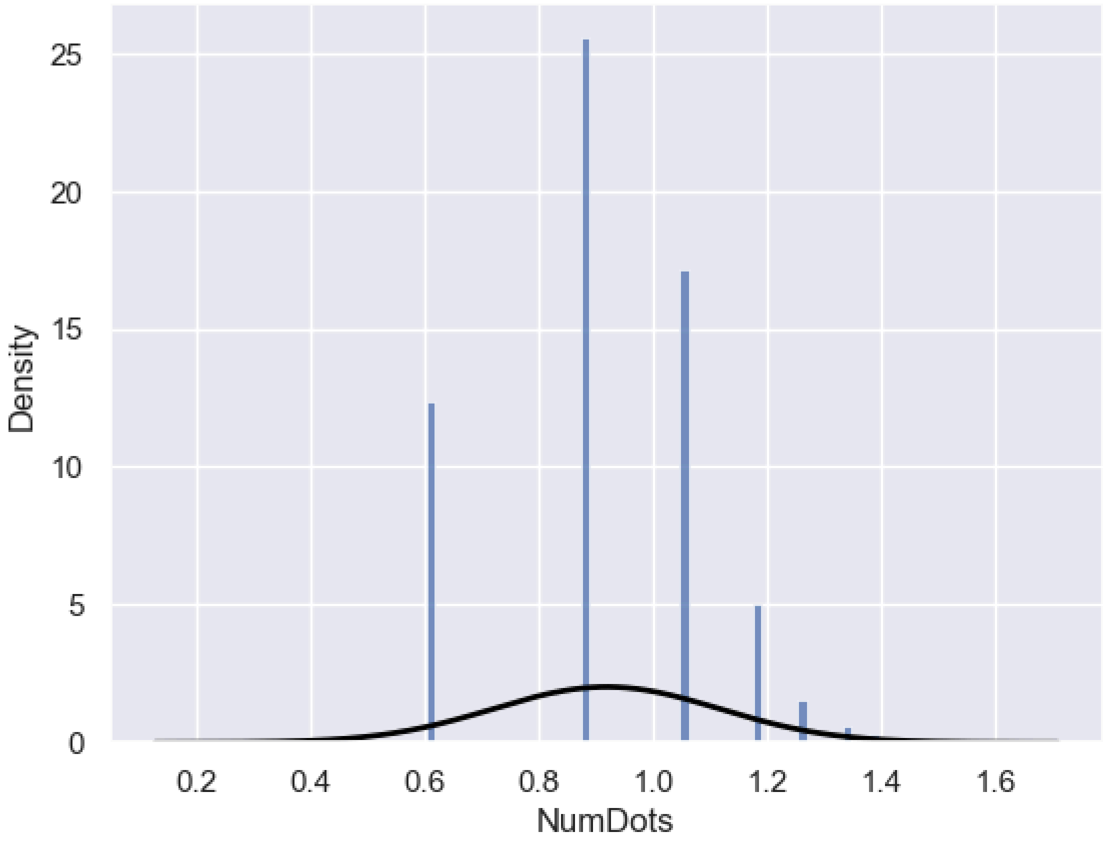
Nettoyage de nos données

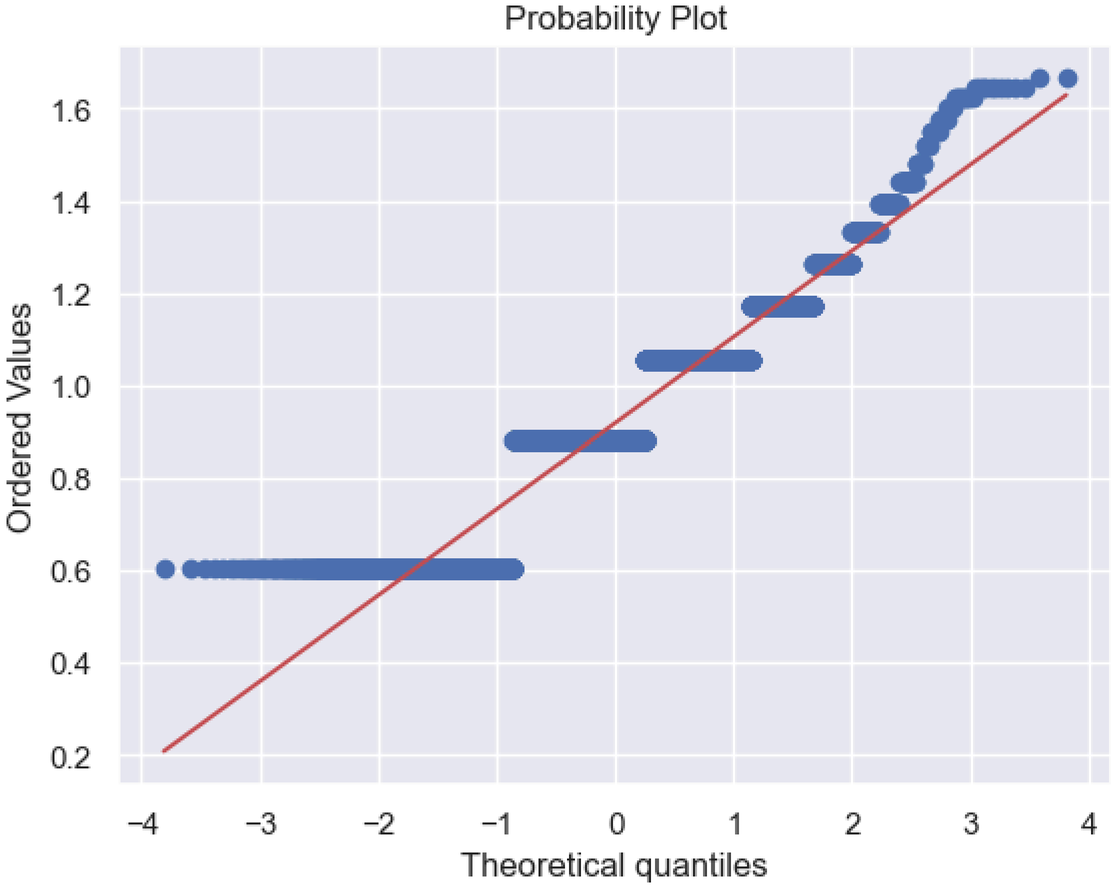
In [ ]:

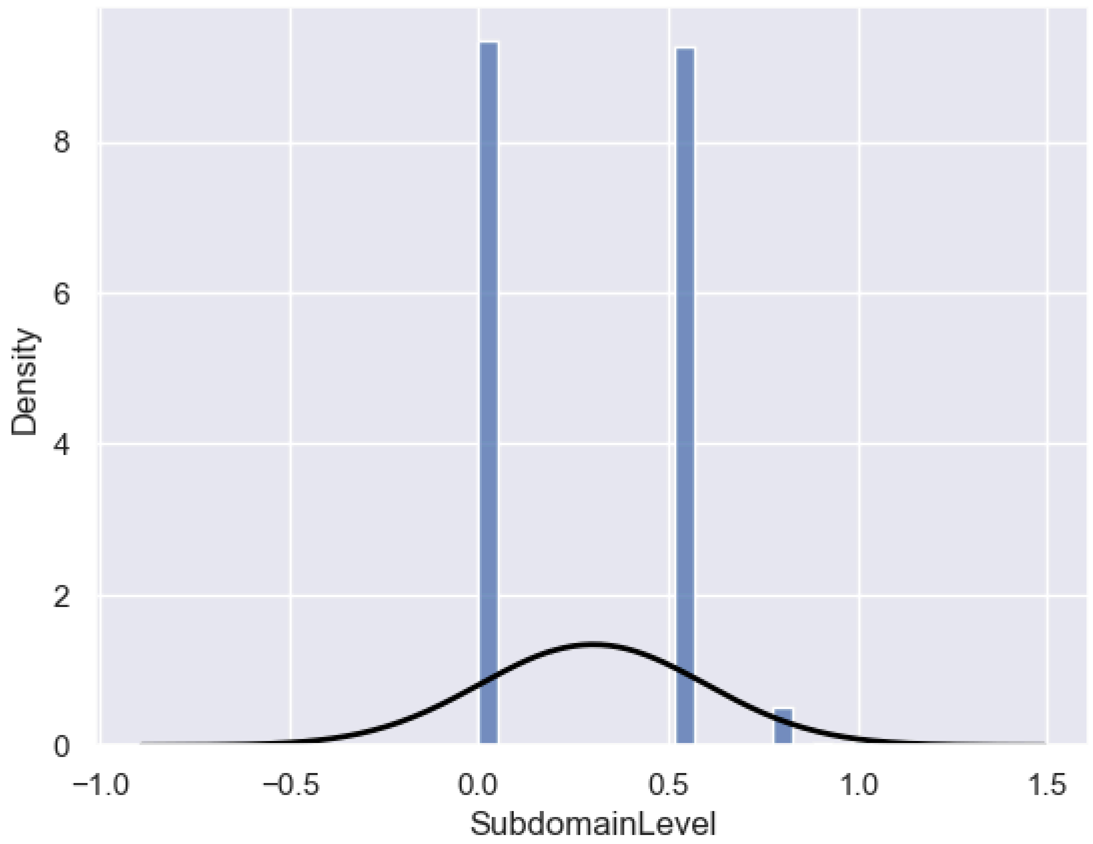
```
#Correction de nos valeurs aberrantes.
df = df[df['NumDots'] < 20]
df = df[df['NumDash'] < 40]
```

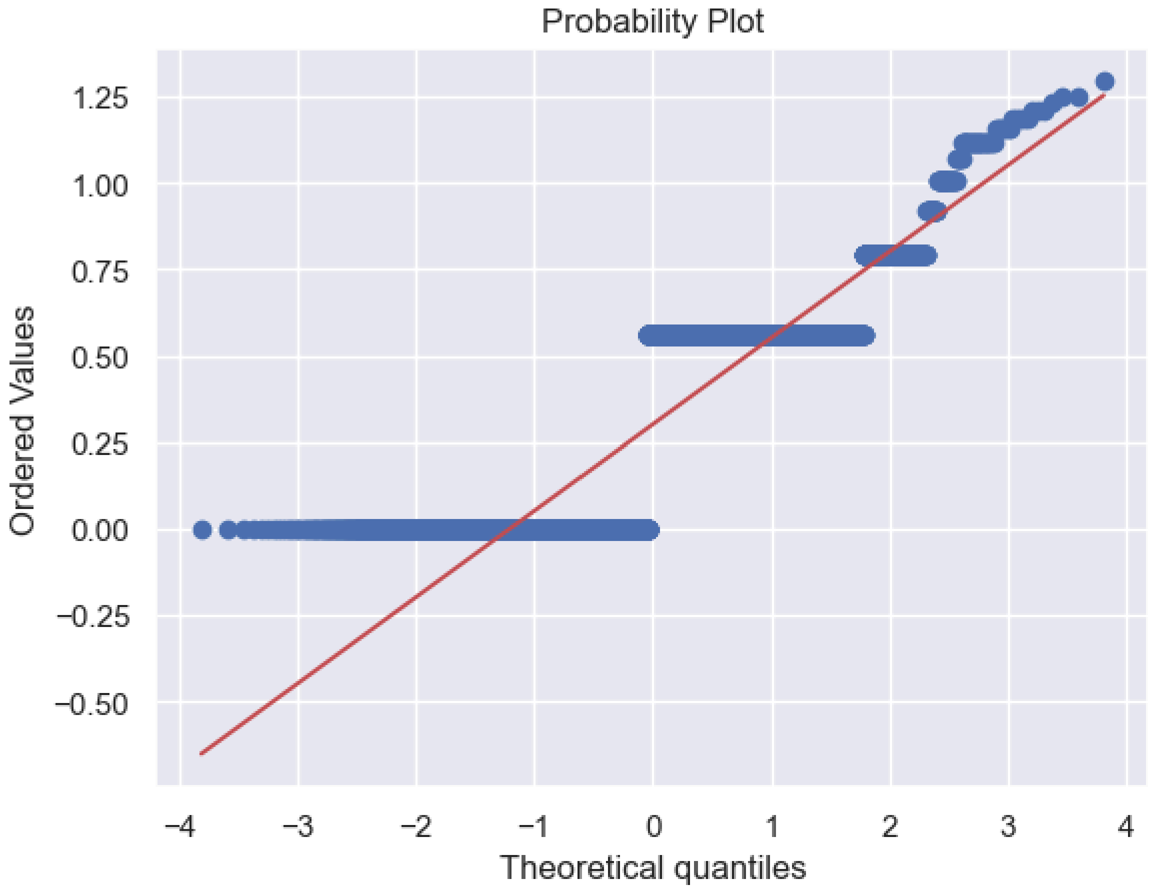
```
plt.scatter(x=df['NumDots'], y=df['NumDash'])
plt.xlabel('NumDots')
plt.ylabel('NumDash')
plt.show()

for col in colonne_continue:
    df[col], _ = yeojohnson(df[col])

for nom_col in colonne_continue:
    fig1, ax1 = plt.subplots()
    sns.histplot(x=df[nom_col], stat="density", ax=ax1)
    normal(df[nom_col].mean(), df[nom_col].std())

    fig2, ax2 = plt.subplots()
    stats.probplot(df[nom_col], plot=ax2)

    plt.show()
```

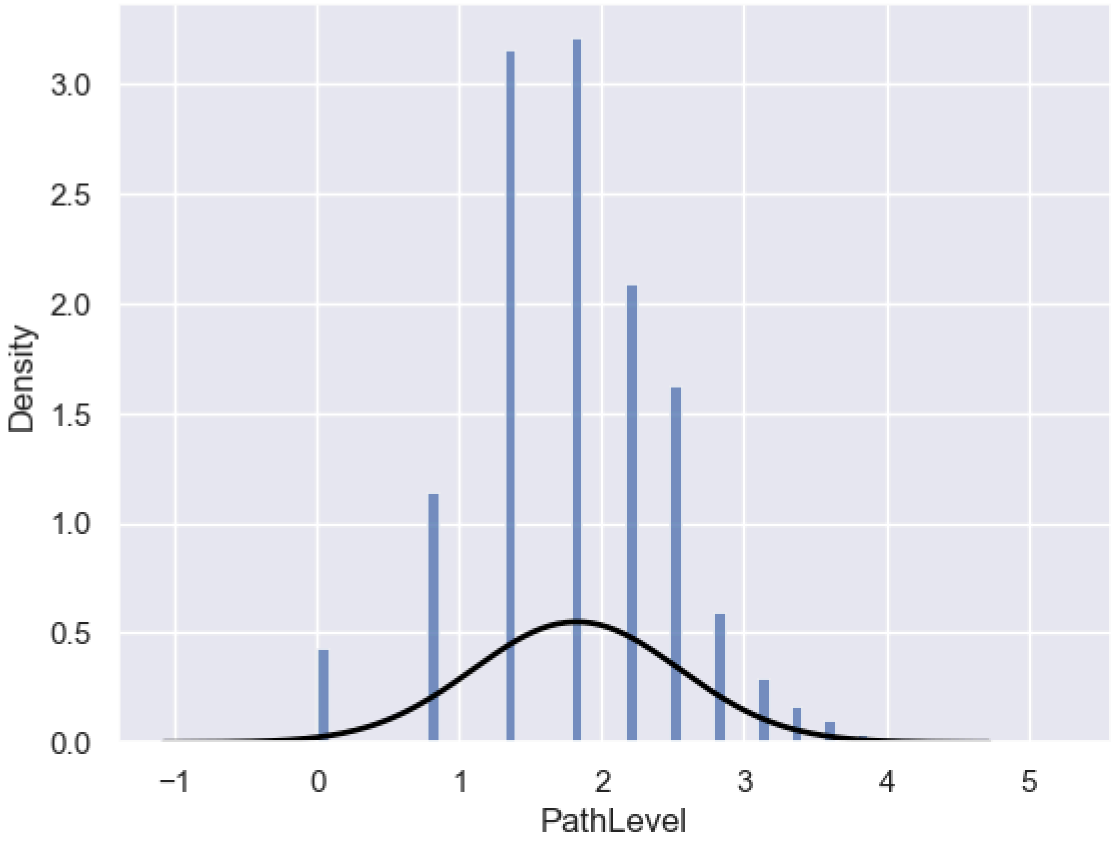


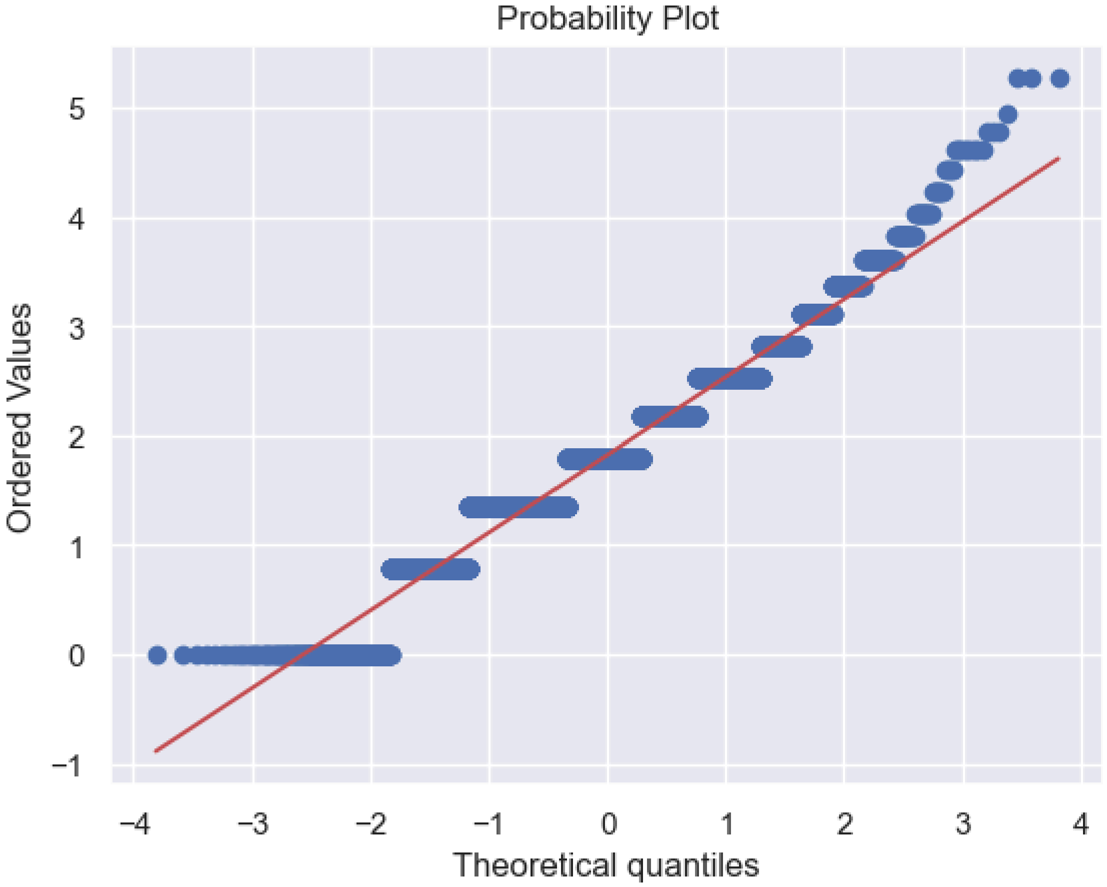

25
20
15
10
5
0
Density
0.2
0.4
0.6
0.8
1.0
1.2
1.4
1.6
NumDots

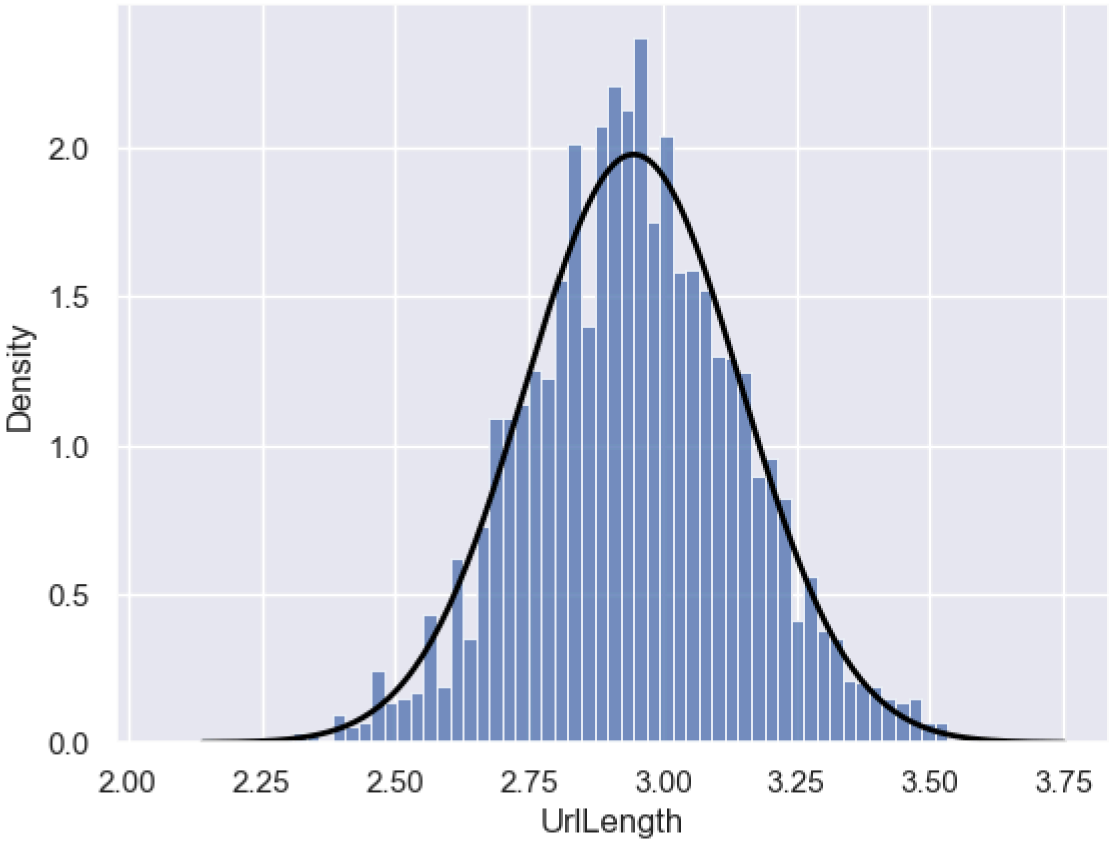

Probability Plot
1.6
1.4
1.2
1.0
0.8
0.6
0.4
0.2
Ordered Values
−4
−3
−2
−1
0
1
2
3
4
Theoretical quantiles

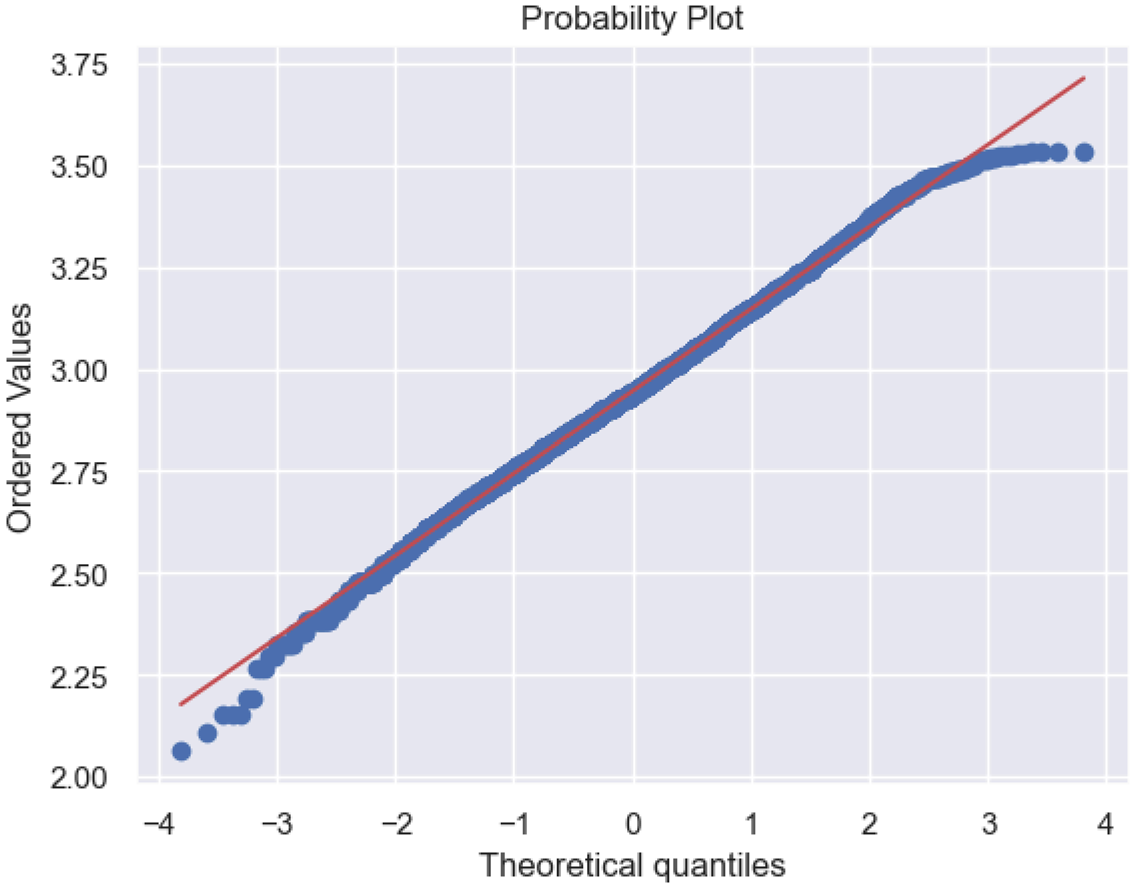

Density
8
6
4
2
0
−1.0
−0.5
0.0
0.5
1.0
1.5
SubdomainLevel

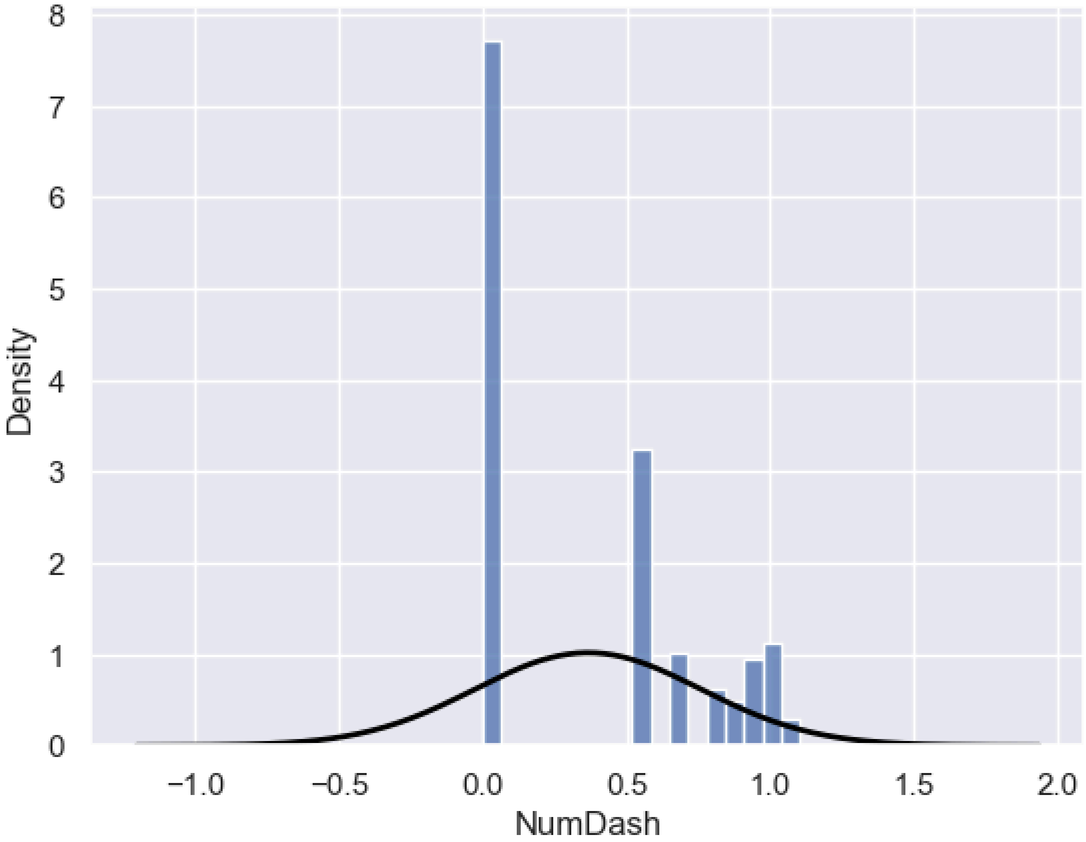

Probability Plot
Ordered Values
1.25
1.00
0.75
0.50
0.25
0.00
−0.25
−0.50
−4
−3
−2
−1
0
1
2
3
4
Theoretical quantiles

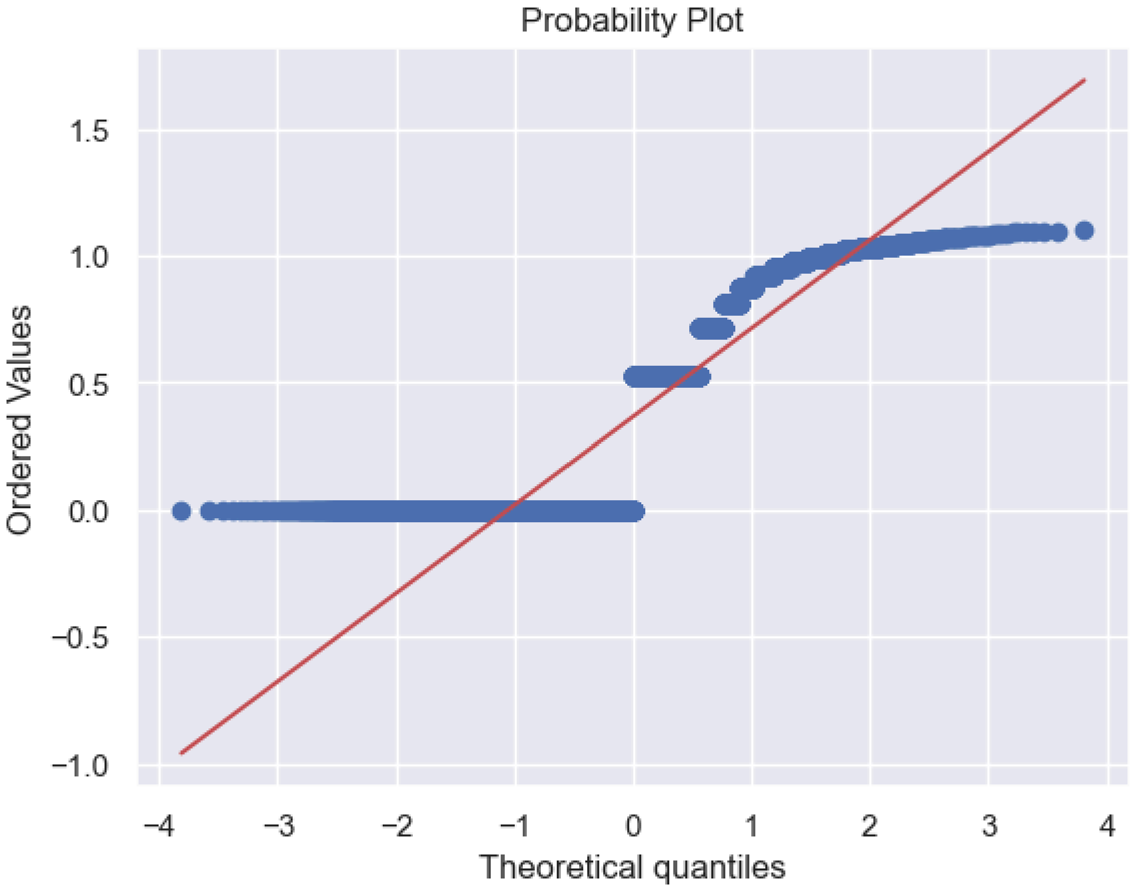

Density
0.0
0.5
1.0
1.5
2.0
2.5
3.0
−1
0
1
2
3
4
5
PathLevel

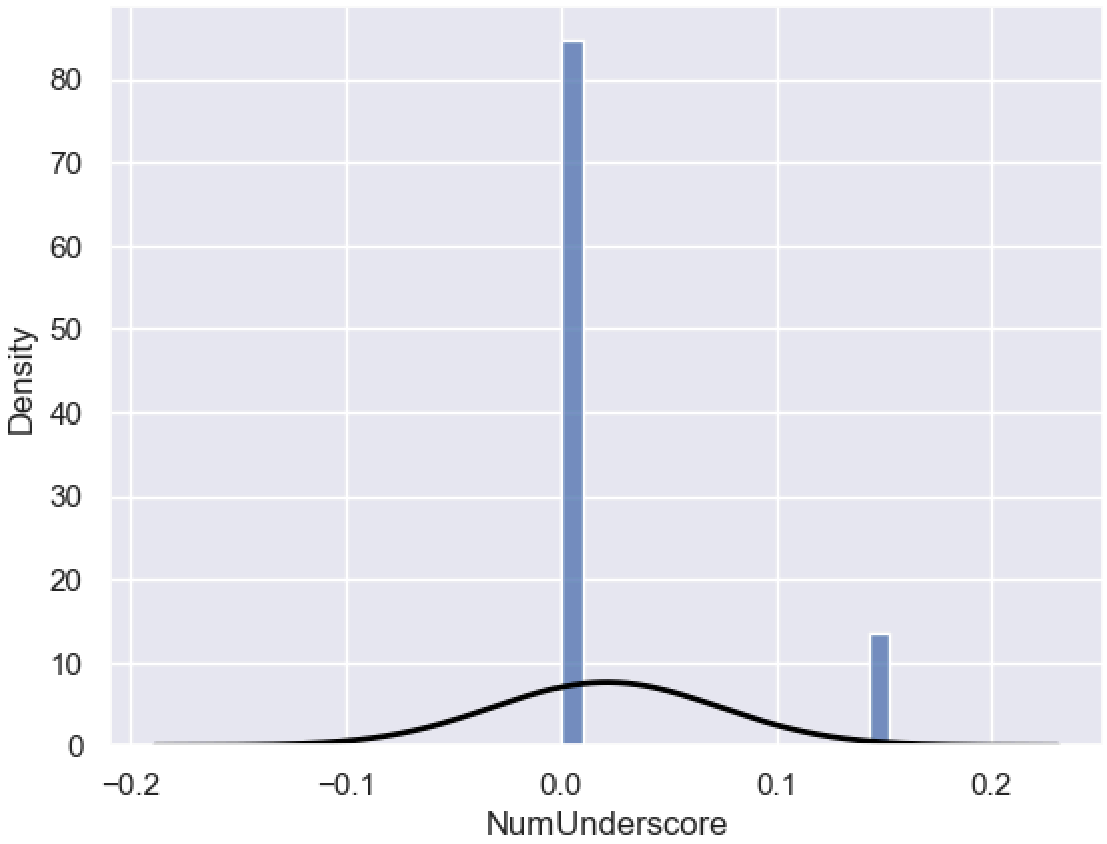

Probability Plot
Ordered Values
−1
0
1
2
3
4
5
−4
−3
−2
−1
0
1
2
3
4
Theoretical quantiles

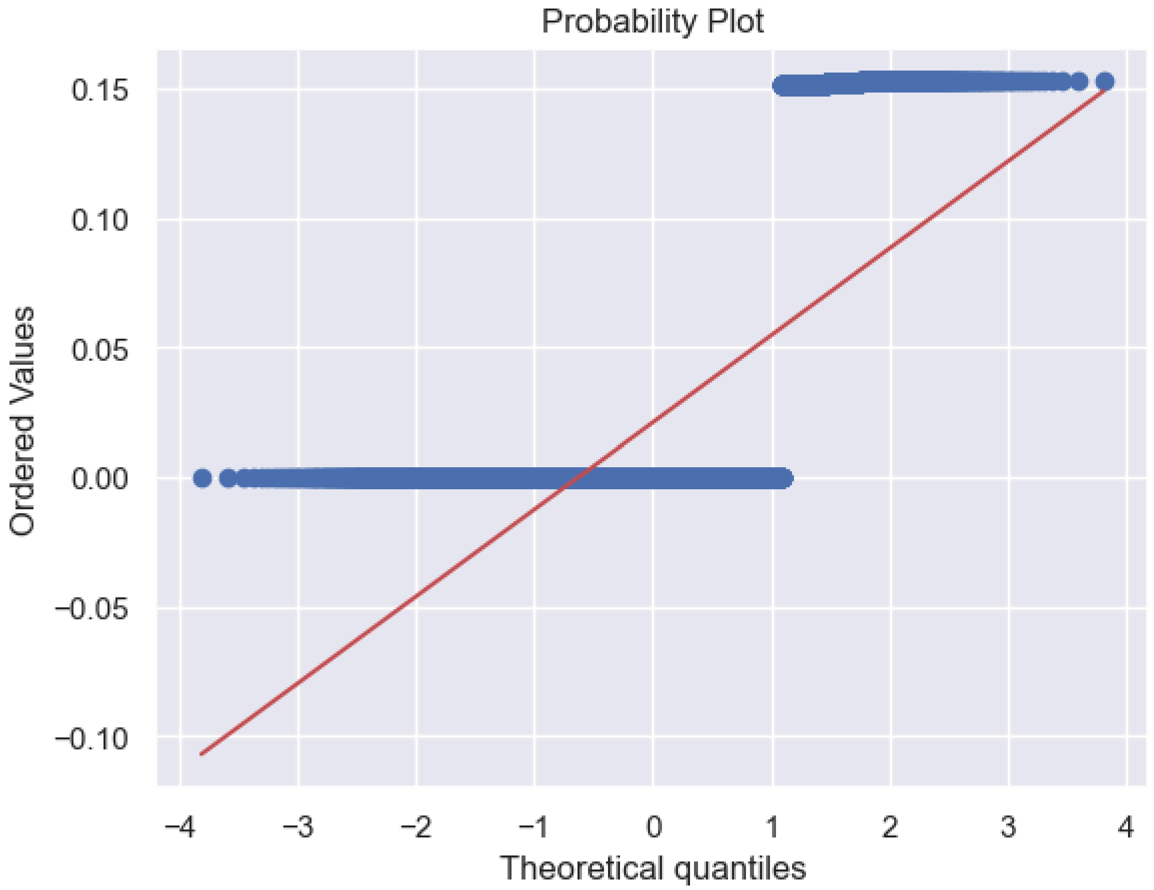

Density
2.0
1.5
1.0
0.5
0.0
2.00
2.25
2.50
2.75
3.00
3.25
3.50
3.75
UrlLength

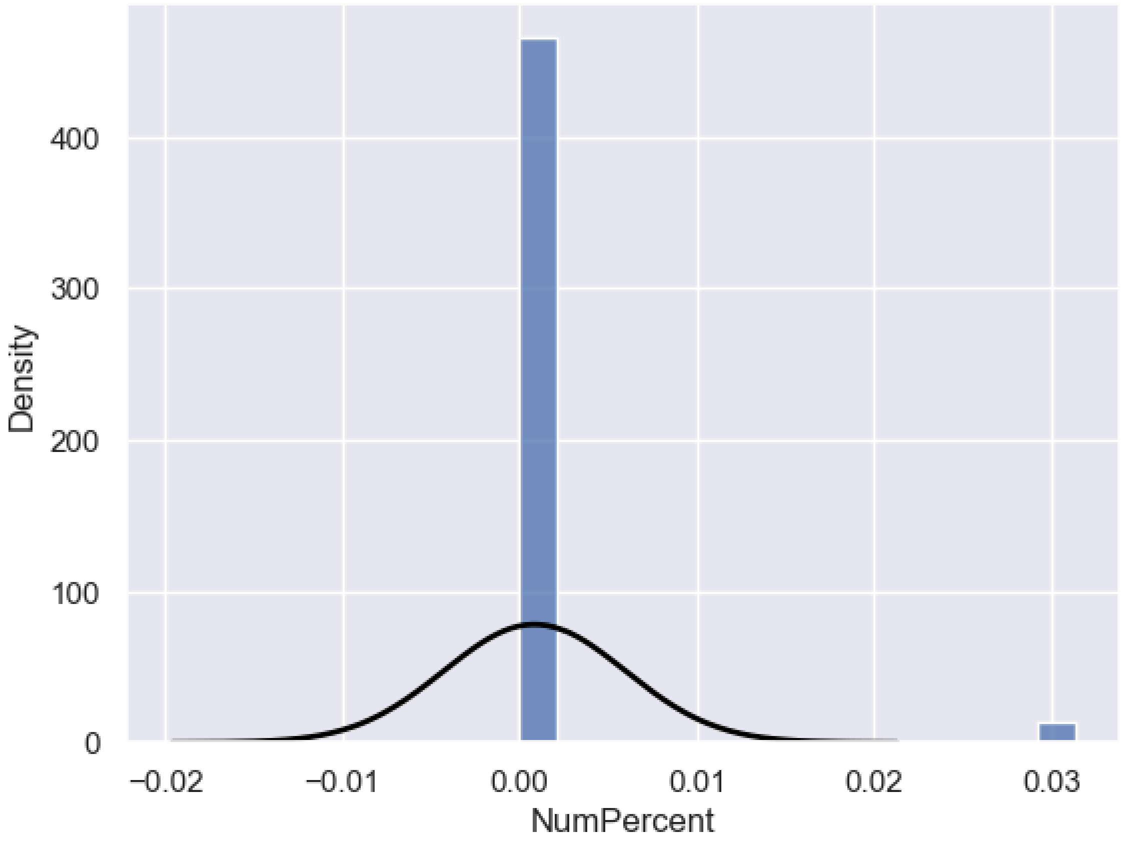

Probability Plot
Ordered Values
3.75
3.50
3.25
3.00
2.75
2.50
2.25
2.00
−4
−3
−2
−1
0
1
2
3
4
Theoretical quantiles

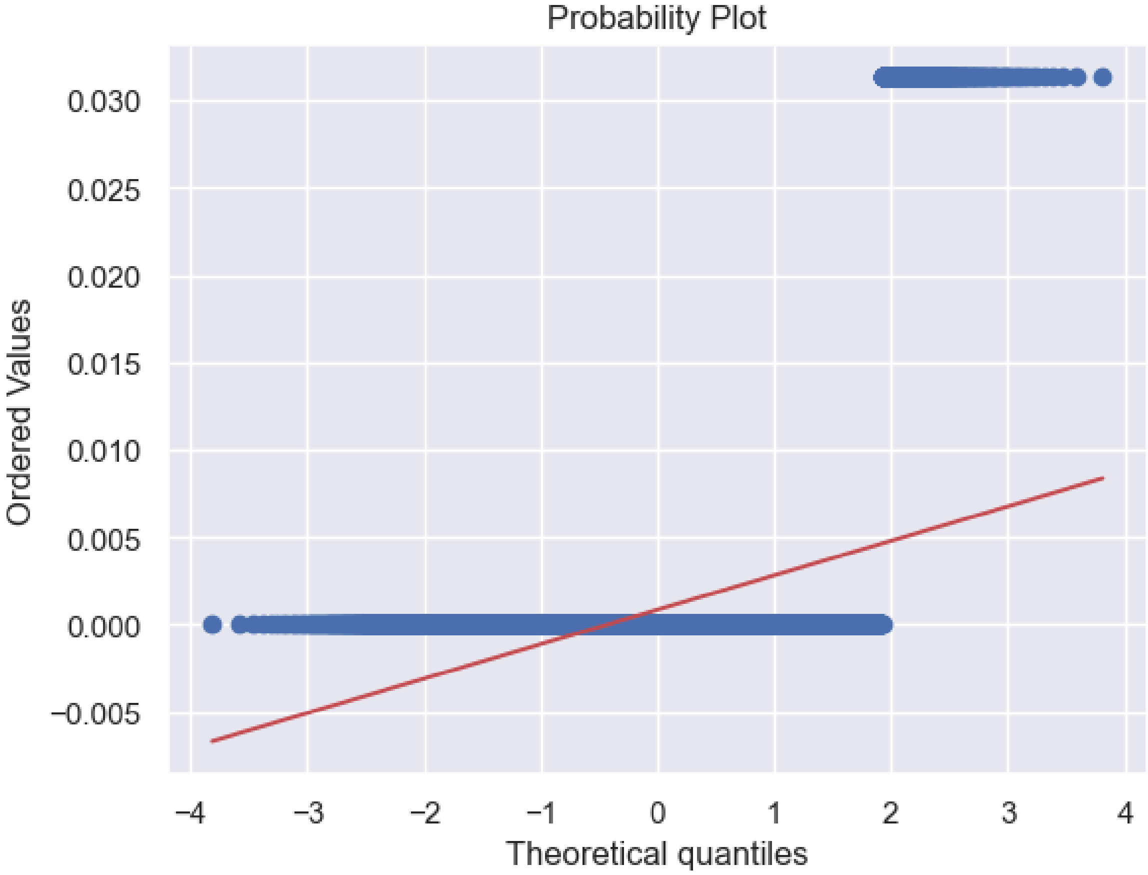

Density
8
7
6
5
4
3
2
1
0
−1.0
−0.5
0.0
0.5
1.0
1.5
2.0
NumDash

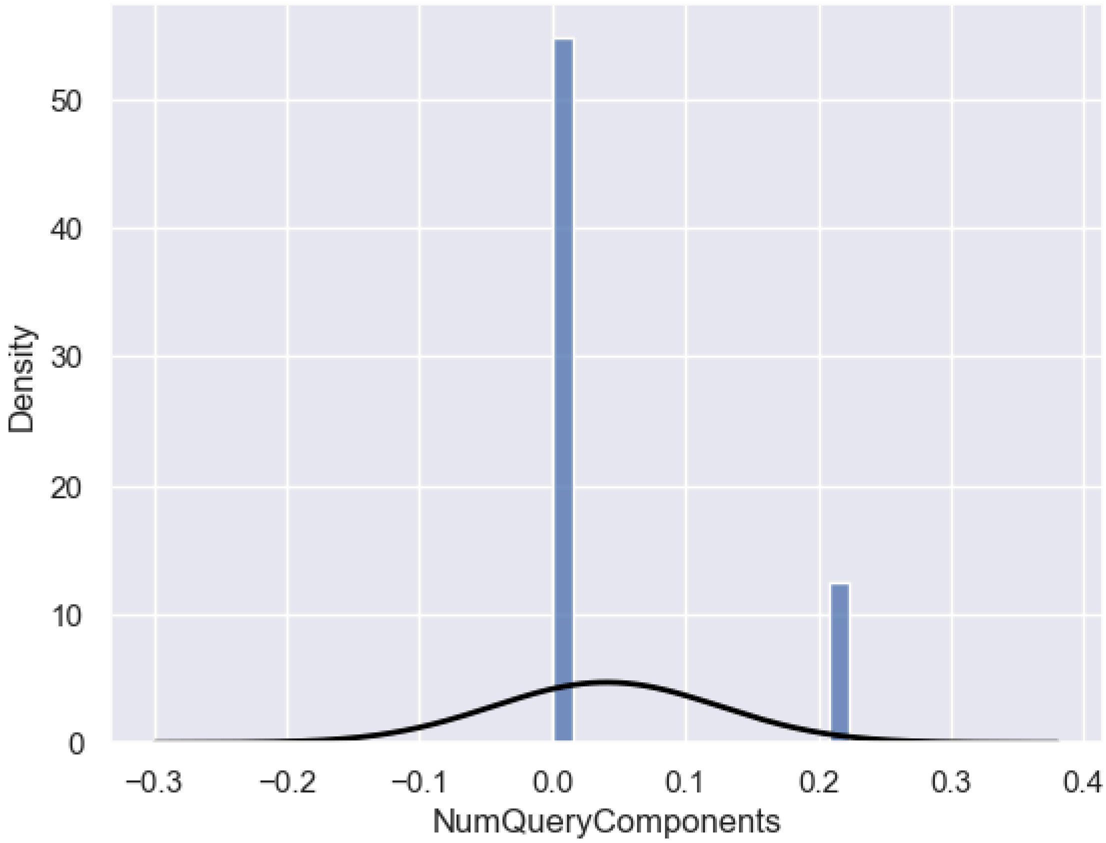

Probability Plot
Ordered Values
1.5
1.0
0.5
0.0
−0.5
−1.0
−4
−3
−2
−1
0
1
2
3
4
Theoretical quantiles

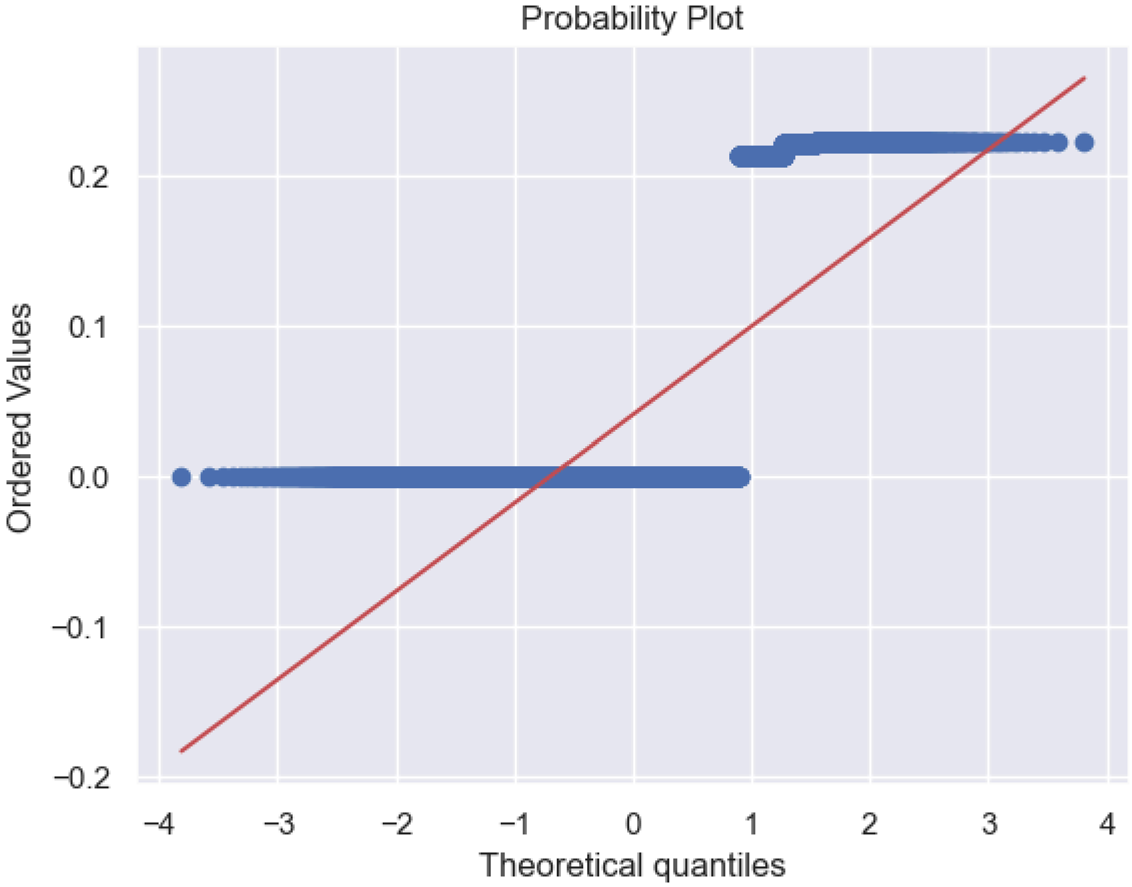

Density
0
10
20
30
40
50
60
70
80
−0.2
−0.1
0.0
0.1
0.2
NumUnderscore

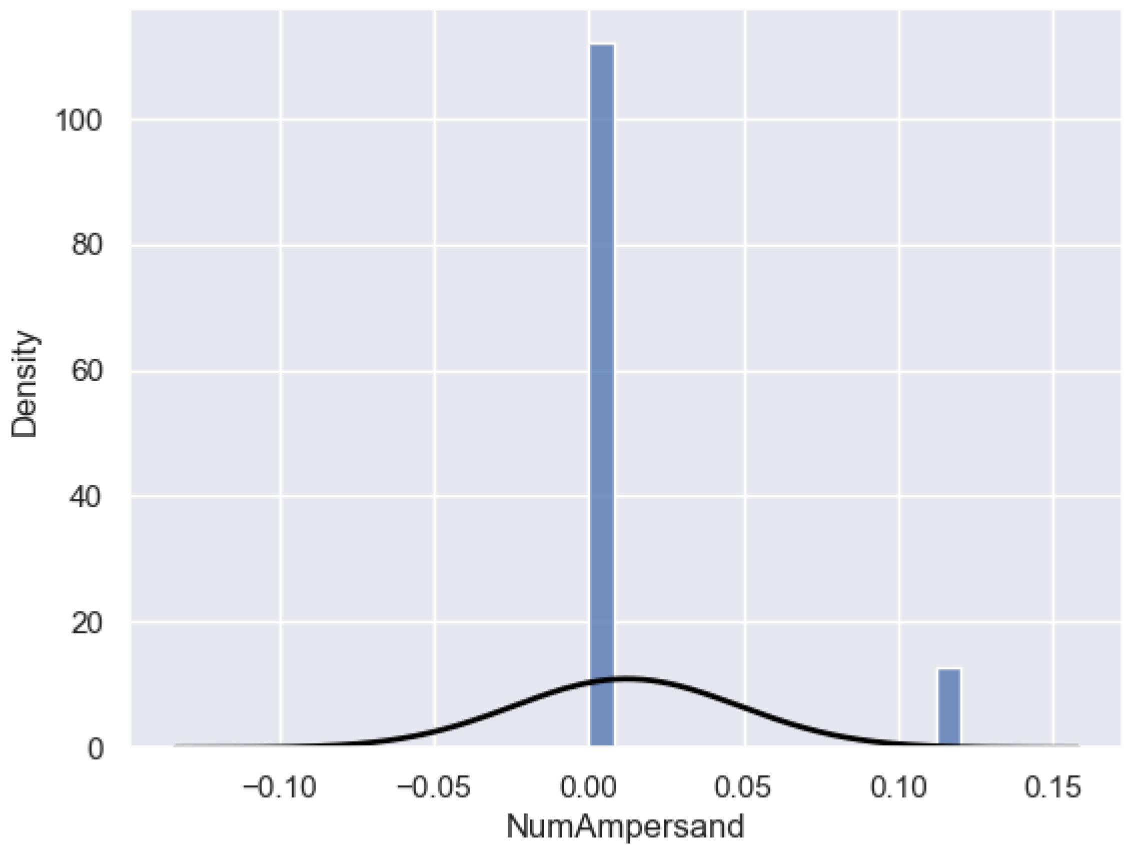

Probability Plot
Ordered Values
0.15
0.10
0.05
0.00
−0.05
−0.10
−4
−3
−2
−1
0
1
2
3
4
Theoretical quantiles

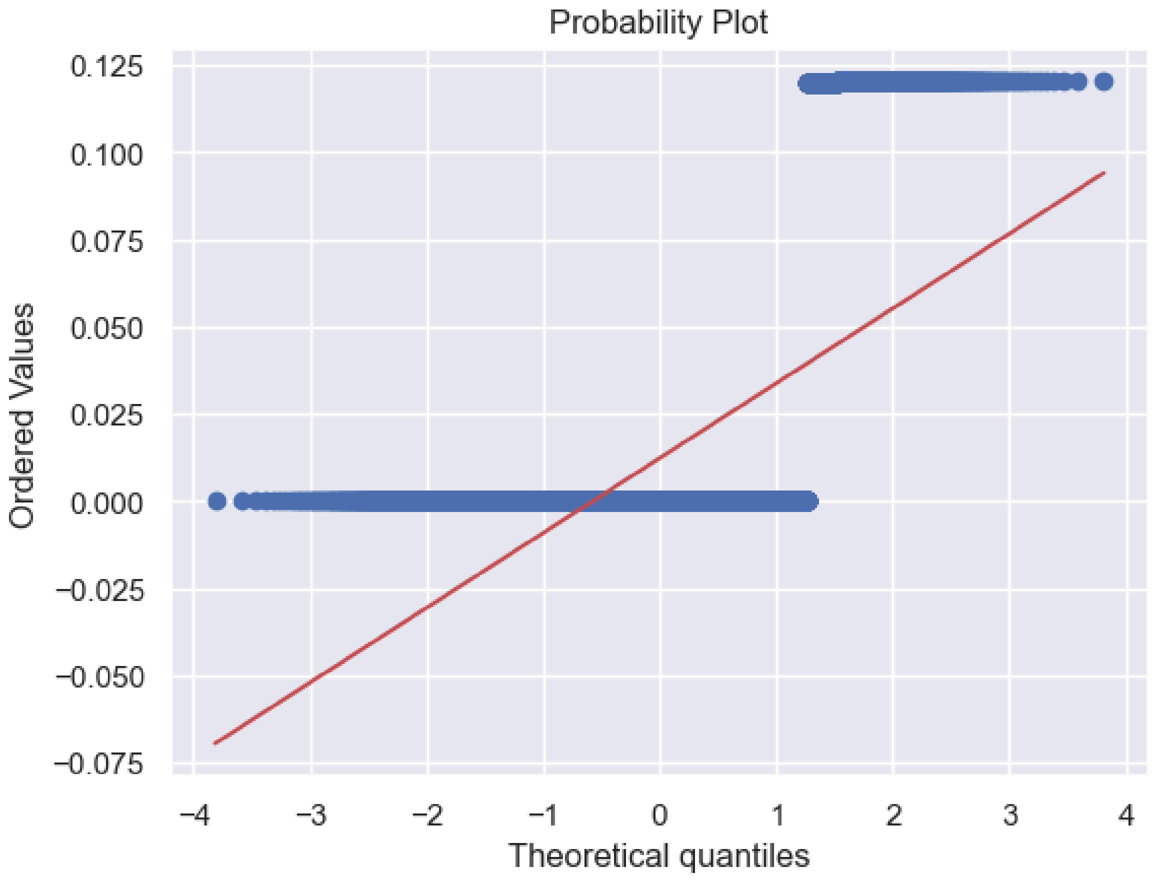

Density
400
300
200
100
0
−0.02
−0.01
0.00
0.01
0.02
0.03
NumPercent

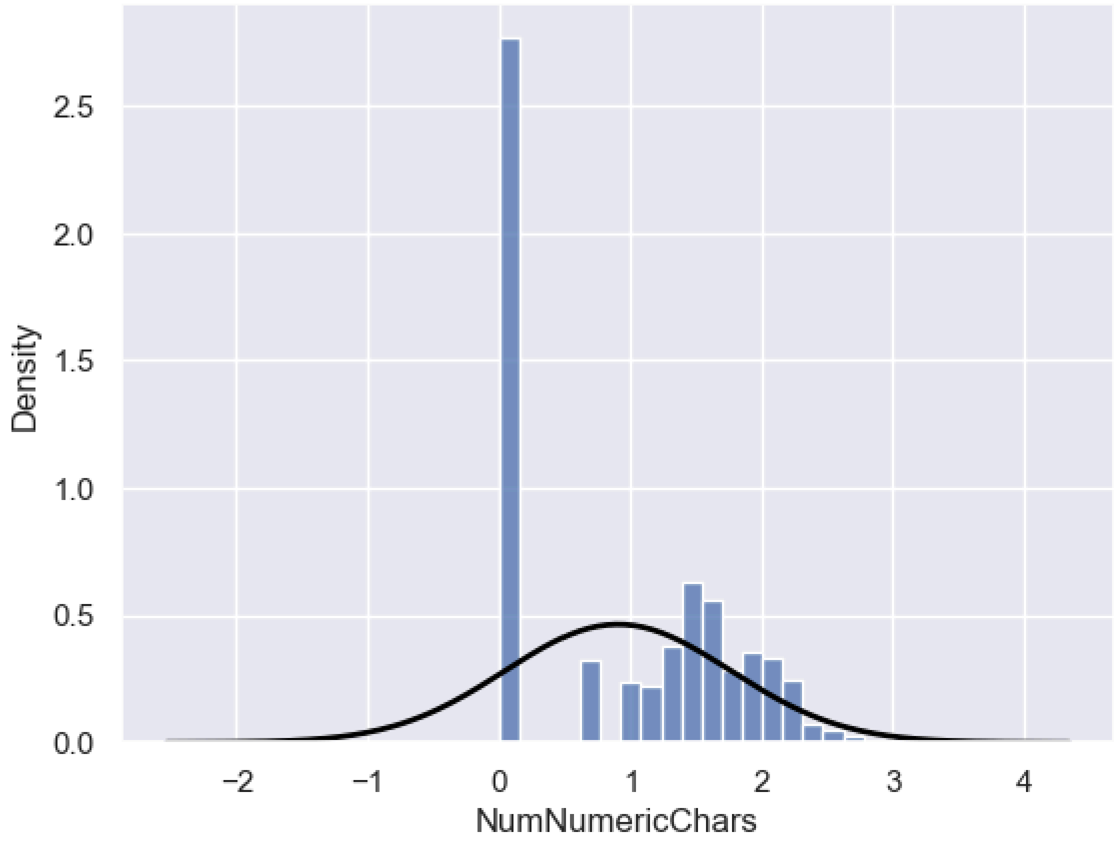

Probability Plot
Ordered Values
0.030
0.025
0.020
0.015
0.010
0.005
0.000
−0.005
−4
−3
−2
−1
0
1
2
3
4
Theoretical quantiles

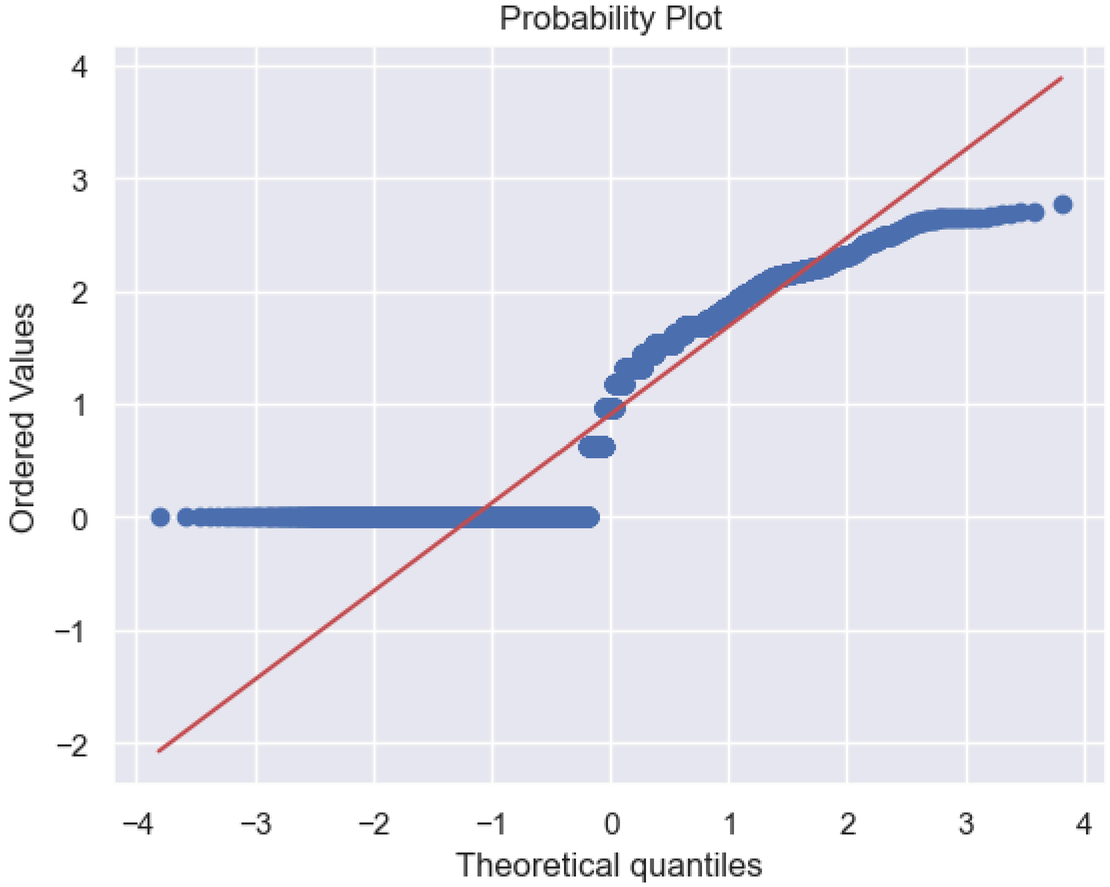

Density
50
40
30
20
10
0
−0.3
−0.2
−0.1
0.0
0.1
0.2
0.3
0.4
NumQueryComponents

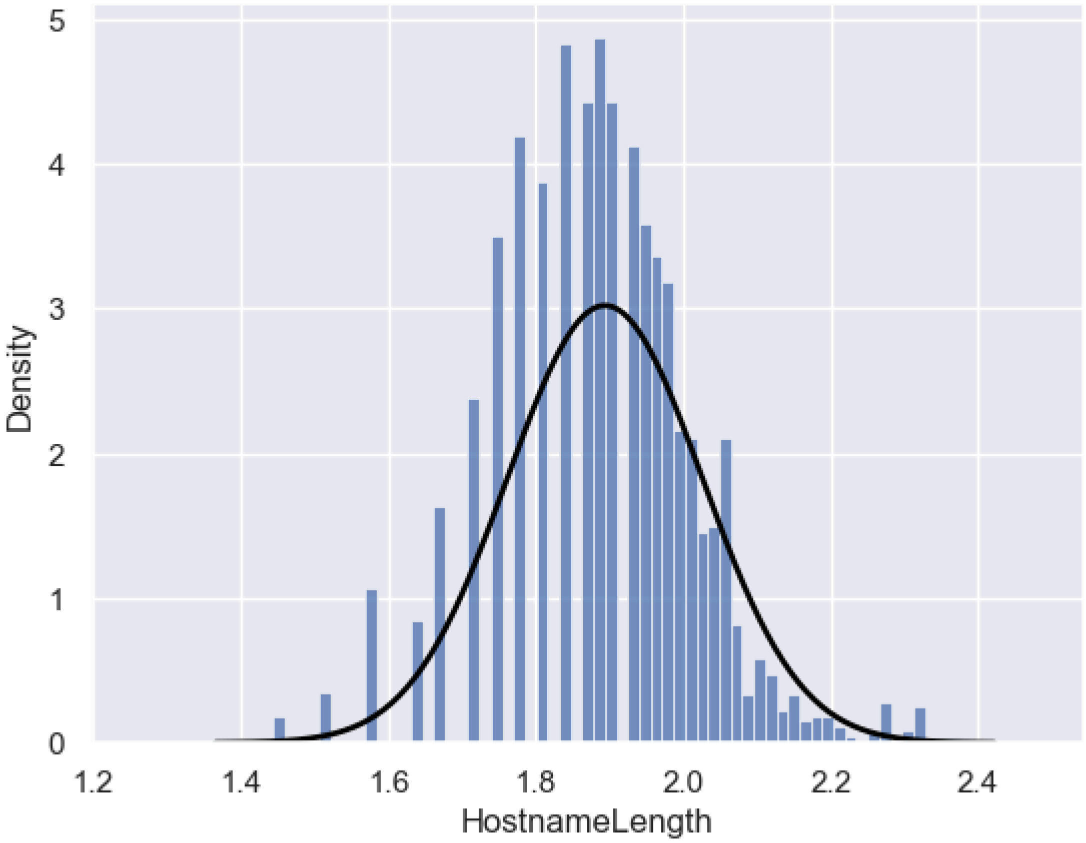

Probability Plot
Ordered Values
0.2
0.1
0.0
−0.1
−0.2
−4
−3
−2
−1
0
1
2
3
4
Theoretical quantiles

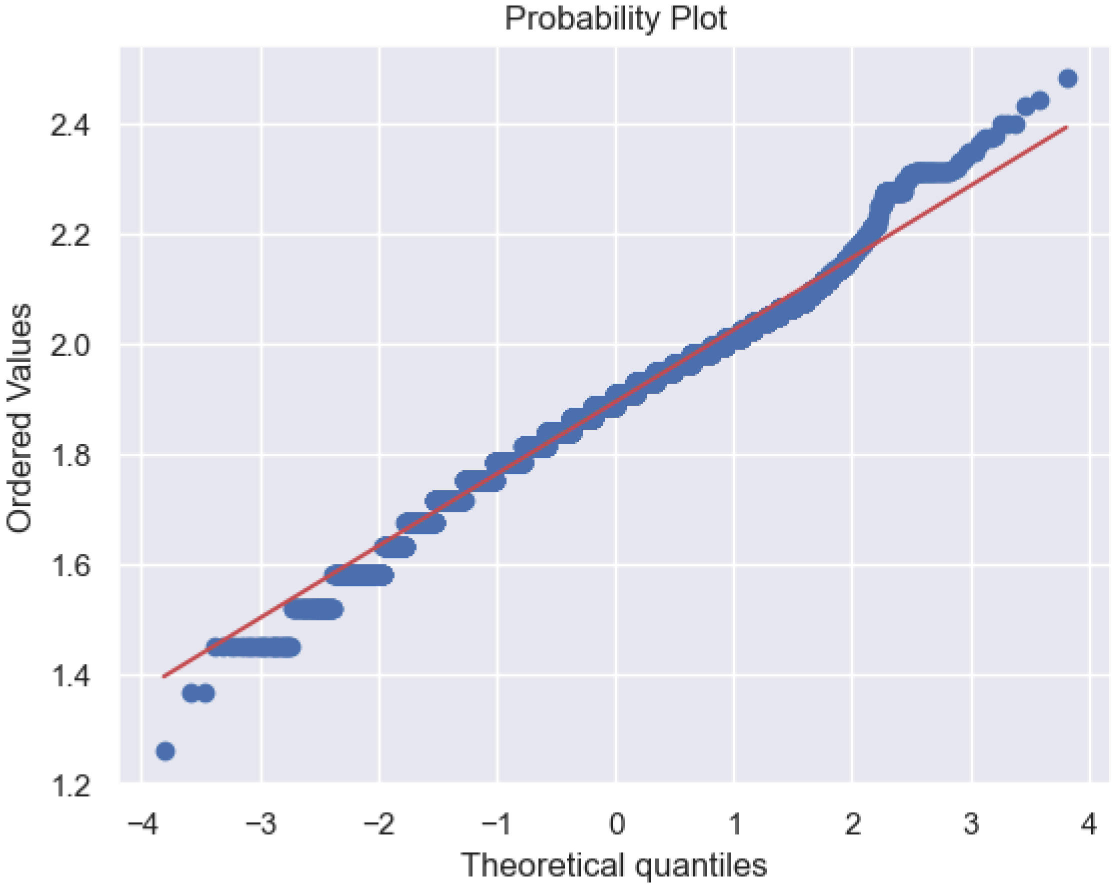

Density
100
80
60
40
20
0
−0.10
−0.05
0.00
0.05
0.10
0.15
NumAmpersand

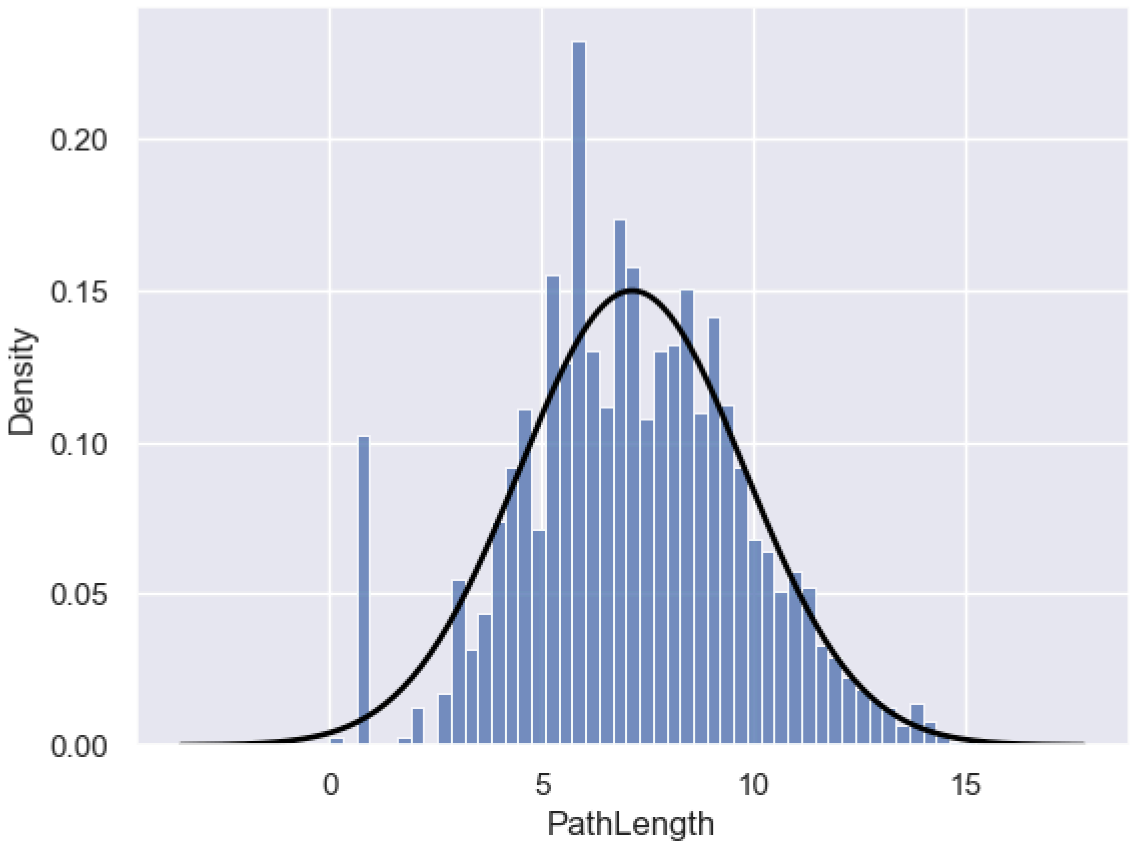

Probability Plot
Ordered Values
0.125
0.100
0.075
0.050
0.025
0.000
−0.025
−0.050
−0.075
−4
−3
−2
−1
0
1
2
3
4
Theoretical quantiles

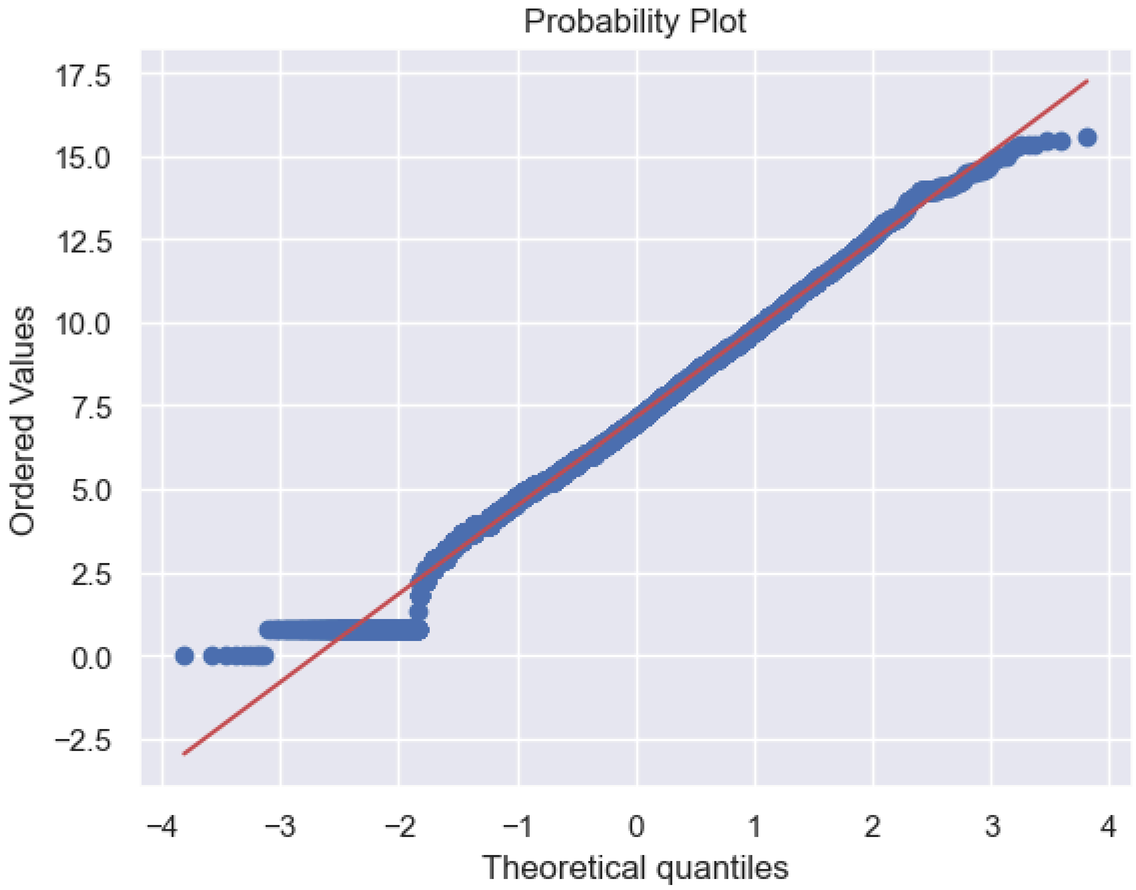

Density
2.5
2.0
1.5
1.0
0.5
0.0
−2
−1
0
1
2
3
4
NumNumericChars

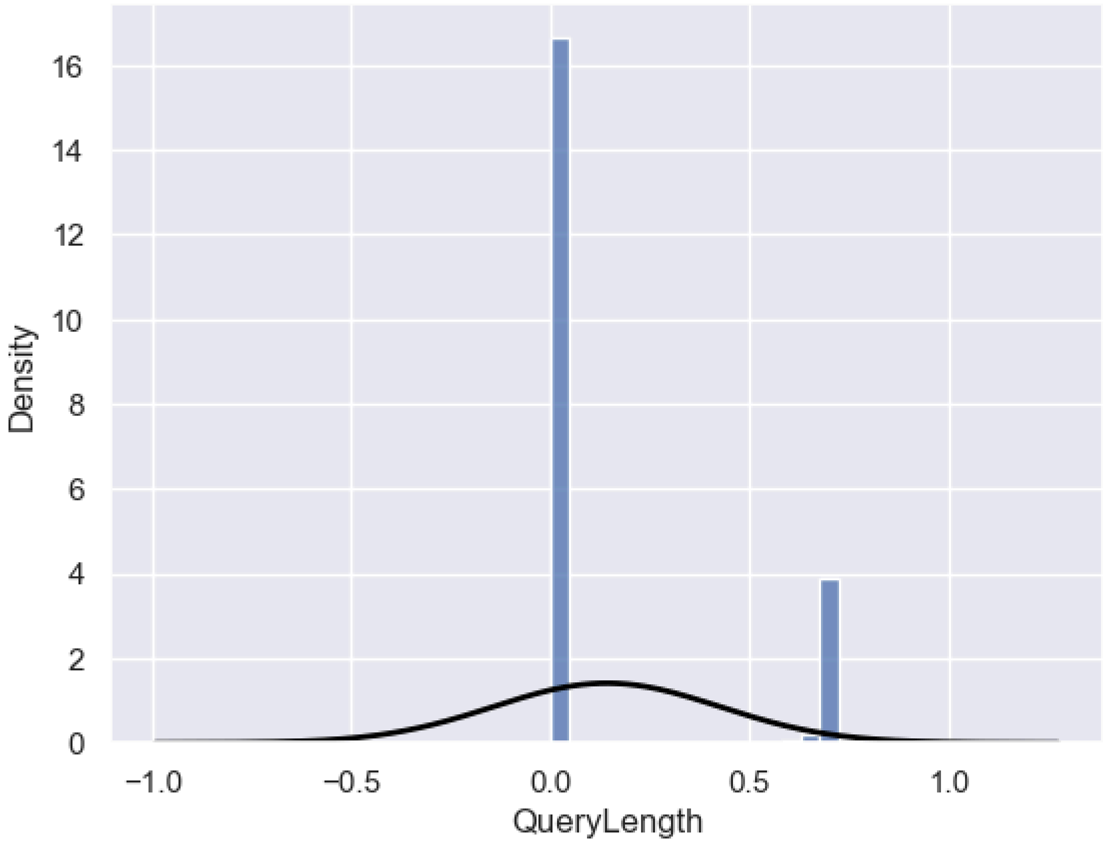

Probability Plot
Ordered Values
4
3
2
1
0
−1
−2
−4
−3
−2
−1
0
1
2
3
4
Theoretical quantiles

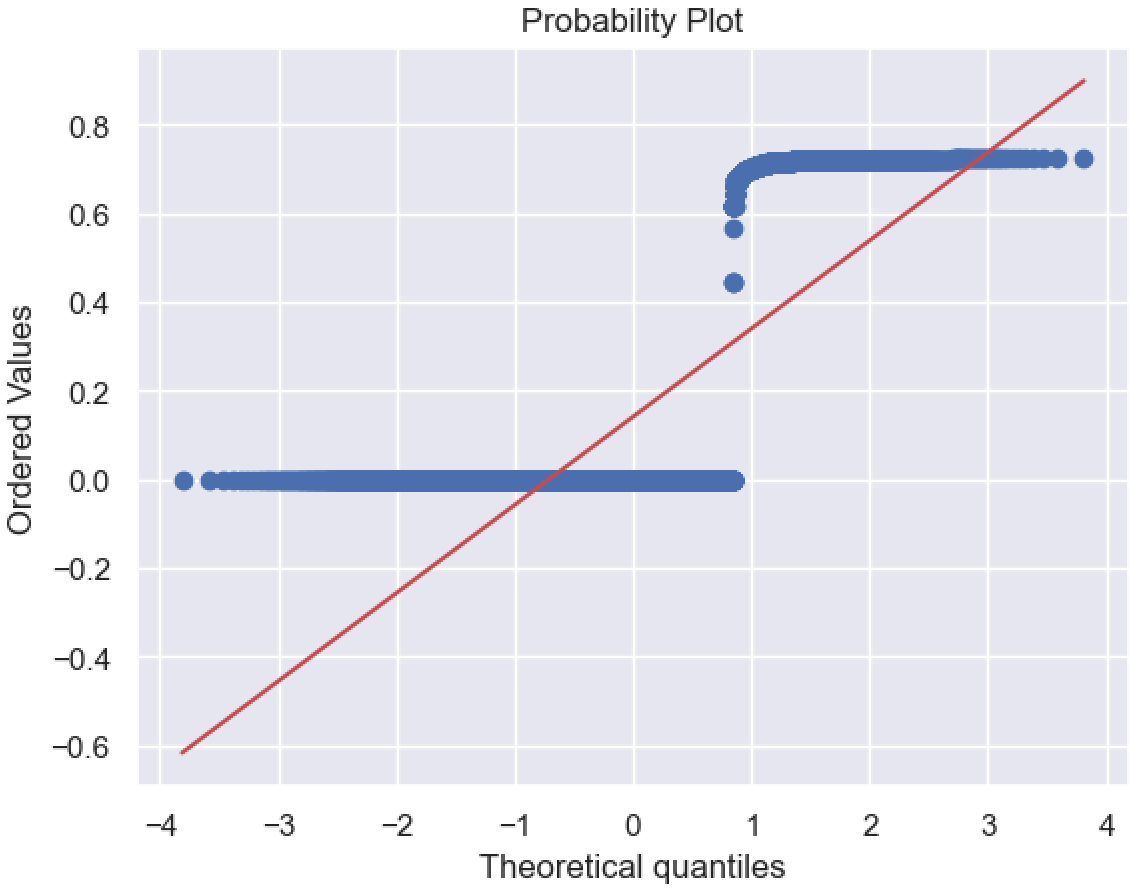

Density
0
1
2
3
4
5
1.2
1.4
1.6
1.8
2.0
2.2
2.4
HostnameLength


Probability Plot
Ordered Values
1.2
1.4
1.6
1.8
2.0
2.2
2.4
−4
−3
−2
−1
0
1
2
3
4
Theoretical quantiles


0.20
0.15
0.10
0.05
0.00
Density
0
5
10
15
PathLength


Probability Plot
17.5
15.0
12.5
10.0
7.5
5.0
2.5
0.0
−2.5
Ordered Values
−4
−3
−2
−1
0
1
2
3
4
Theoretical quantiles





Création du modèle statistique.

récuperation de la meilleur configuration.

In [ ]:

```
#Extraction de notre colonne"Class_label".
col = df.columns.to_list()
#Supression de celle-ci.( du data-set)
col.remove('CLASS_LABEL')

#Définition d'une nouvelle base de données.
X = df[col]
y = df["CLASS_LABEL"]

#On sépare notre échantillon.
X_entrainement, X_test, Y_entrainement, y_test = train_test_split(X, y, test_siz

#On définit les paramètres de notre forêt.
def objectif(essaie):
    n_estim = essaie.suggest_int('n_estimations', 10, 300)
    prodondeur_maximum = essaie.suggest_int('prodondeur_maximum', 2, 32, log=Tru
    echantillons_min_div = essaie.suggest_int('echantillons_min_div', 2, 20)
    echantillons_min_noeud = essaie.suggest_int('echantillons_min_noeud', 1, 20)

    clf = RandomForestClassifier(
        n_estimators=n_estim,
        max_depth=prodondeur_maximum,
        min_samples_split=echantillons_min_div,
        min_samples_leaf=echantillons_min_noeud,
        random_state=42,
        n_jobs=-1
    )

    clf.fit(X_entrainement, Y_entrainement)

    y_pred = clf.predict(X_test)

    exactitude = accuracy_score(y_test, y_pred)

    return exactitude
#On crée une étude pour maximiser la valeur de l'exactitude en lançant notre fon
etude = optuna.create_study(direction='maximize')
etude.optimize(objectif, n_trials=100)
#On extrait alors le meilleur essai nous permettant d'automatiser la recherche d
meilleur_essaie = etude.best_trial
resultat = meilleur_essaie.params
```

```
[I 2024-07-01 15:09:52,556] A new study created in memory with name: no-name-1560
2e86-1869-496b-aed3-40224c0872f5
```

```
[I 2024-07-01 15:09:52,925] Trial 0 finished with value: 0.816 and parameters:
{'n_estimations': 162, 'prodondeur_maximum': 2, 'echantillons_min_div': 7, 'echan
tillons_min_noeud': 16}. Best is trial 0 with value: 0.816.
[I 2024-07-01 15:09:53,406] Trial 1 finished with value: 0.8315 and parameters:
{'n_estimations': 231, 'prodondeur_maximum': 3, 'echantillons_min_div': 20, 'echa
ntillons_min_noeud': 16}. Best is trial 1 with value: 0.8315.
[I 2024-07-01 15:09:53,610] Trial 2 finished with value: 0.8895 and parameters:
{'n_estimations': 59, 'prodondeur_maximum': 13, 'echantillons_min_div': 19, 'echa
ntillons_min_noeud': 13}. Best is trial 2 with value: 0.8895.
[I 2024-07-01 15:09:53,855] Trial 3 finished with value: 0.896 and parameters:
{'n_estimations': 89, 'prodondeur_maximum': 24, 'echantillons_min_div': 18, 'echa
ntillons_min_noeud': 20}. Best is trial 3 with value: 0.896.
[I 2024-07-01 15:09:54,707] Trial 4 finished with value: 0.878 and parameters:
{'n_estimations': 209, 'prodondeur_maximum': 8, 'echantillons_min_div': 10, 'echa
ntillons_min_noeud': 14}. Best is trial 3 with value: 0.896.
[I 2024-07-01 15:09:55,559] Trial 5 finished with value: 0.8375 and parameters:
{'n_estimations': 201, 'prodondeur_maximum': 3, 'echantillons_min_div': 15, 'echa
ntillons_min_noeud': 4}. Best is trial 3 with value: 0.896.
[I 2024-07-01 15:09:56,492] Trial 6 finished with value: 0.89 and parameters: {'n
_estimations': 228, 'prodondeur_maximum': 25, 'echantillons_min_div': 15, 'echant
illons_min_noeud': 18}. Best is trial 3 with value: 0.896.
[I 2024-07-01 15:09:57,187] Trial 7 finished with value: 0.891 and parameters:
{'n_estimations': 289, 'prodondeur_maximum': 10, 'echantillons_min_div': 9, 'echa
ntillons_min_noeud': 14}. Best is trial 3 with value: 0.896.
[I 2024-07-01 15:09:57,796] Trial 8 finished with value: 0.853 and parameters:
{'n_estimations': 228, 'prodondeur_maximum': 4, 'echantillons_min_div': 12, 'echa
ntillons_min_noeud': 13}. Best is trial 3 with value: 0.896.
[I 2024-07-01 15:09:58,060] Trial 9 finished with value: 0.89 and parameters: {'n
_estimations': 107, 'prodondeur_maximum': 12, 'echantillons_min_div': 6, 'echanti
llons_min_noeud': 20}. Best is trial 3 with value: 0.896.
[I 2024-07-01 15:09:58,133] Trial 10 finished with value: 0.8965 and parameters:
{'n_estimations': 10, 'prodondeur_maximum': 32, 'echantillons_min_div': 16, 'echa
ntillons_min_noeud': 7}. Best is trial 10 with value: 0.8965.
[I 2024-07-01 15:09:58,220] Trial 11 finished with value: 0.8905 and parameters:
{'n_estimations': 10, 'prodondeur_maximum': 32, 'echantillons_min_div': 2, 'echan
tillons_min_noeud': 7}. Best is trial 10 with value: 0.8965.
[I 2024-07-01 15:09:58,329] Trial 12 finished with value: 0.8965 and parameters:
{'n_estimations': 18, 'prodondeur_maximum': 20, 'echantillons_min_div': 16, 'echa
ntillons_min_noeud': 8}. Best is trial 10 with value: 0.8965.
[I 2024-07-01 15:09:58,468] Trial 13 finished with value: 0.9005 and parameters:
{'n_estimations': 27, 'prodondeur_maximum': 18, 'echantillons_min_div': 16, 'echa
ntillons_min_noeud': 8}. Best is trial 13 with value: 0.9005.
[I 2024-07-01 15:09:58,626] Trial 14 finished with value: 0.903 and parameters:
{'n_estimations': 48, 'prodondeur_maximum': 17, 'echantillons_min_div': 13, 'echa
ntillons_min_noeud': 2}. Best is trial 14 with value: 0.903.
[I 2024-07-01 15:09:58,915] Trial 15 finished with value: 0.905 and parameters:
{'n_estimations': 70, 'prodondeur_maximum': 16, 'echantillons_min_div': 13, 'echa
ntillons_min_noeud': 1}. Best is trial 15 with value: 0.905.
[I 2024-07-01 15:09:59,484] Trial 16 finished with value: 0.8605 and parameters:
{'n_estimations': 118, 'prodondeur_maximum': 6, 'echantillons_min_div': 12, 'echa
ntillons_min_noeud': 1}. Best is trial 15 with value: 0.905.
[I 2024-07-01 15:09:59,729] Trial 17 finished with value: 0.904 and parameters:
{'n_estimations': 65, 'prodondeur_maximum': 14, 'echantillons_min_div': 13, 'echa
ntillons_min_noeud': 1}. Best is trial 15 with value: 0.905.
[I 2024-07-01 15:10:00,103] Trial 18 finished with value: 0.879 and parameters:
{'n_estimations': 142, 'prodondeur_maximum': 7, 'echantillons_min_div': 7, 'echan
tillons_min_noeud': 4}. Best is trial 15 with value: 0.905.
[I 2024-07-01 15:10:00,339] Trial 19 finished with value: 0.8965 and parameters:
{'n_estimations': 74, 'prodondeur_maximum': 14, 'echantillons_min_div': 13, 'echa
ntillons_min_noeud': 3}. Best is trial 15 with value: 0.905.
```

```
[I 2024-07-01 15:10:00,719] Trial 20 finished with value: 0.8575 and parameters:
{'n_estimations': 147, 'prodondeur_maximum': 5, 'echantillons_min_div': 9, 'echan
tillons_min_noeud': 5}. Best is trial 15 with value: 0.905.
[I 2024-07-01 15:10:01,000] Trial 21 finished with value: 0.909 and parameters:
{'n_estimations': 50, 'prodondeur_maximum': 17, 'echantillons_min_div': 13, 'echa
ntillons_min_noeud': 1}. Best is trial 21 with value: 0.909.
[I 2024-07-01 15:10:01,179] Trial 22 finished with value: 0.8975 and parameters:
{'n_estimations': 51, 'prodondeur_maximum': 10, 'echantillons_min_div': 14, 'echa
ntillons_min_noeud': 1}. Best is trial 21 with value: 0.909.
[I 2024-07-01 15:10:01,633] Trial 23 finished with value: 0.9015 and parameters:
{'n_estimations': 93, 'prodondeur_maximum': 15, 'echantillons_min_div': 11, 'echa
ntillons_min_noeud': 5}. Best is trial 21 with value: 0.909.
[I 2024-07-01 15:10:01,895] Trial 24 finished with value: 0.8925 and parameters:
{'n_estimations': 67, 'prodondeur_maximum': 10, 'echantillons_min_div': 18, 'echa
ntillons_min_noeud': 1}. Best is trial 21 with value: 0.909.
[I 2024-07-01 15:10:02,256] Trial 25 finished with value: 0.897 and parameters:
{'n_estimations': 125, 'prodondeur_maximum': 22, 'echantillons_min_div': 10, 'ech
antillons_min_noeud': 10}. Best is trial 21 with value: 0.909.
[I 2024-07-01 15:10:02,449] Trial 26 finished with value: 0.9015 and parameters:
{'n_estimations': 45, 'prodondeur_maximum': 12, 'echantillons_min_div': 14, 'echa
ntillons_min_noeud': 3}. Best is trial 21 with value: 0.909.
[I 2024-07-01 15:10:02,731] Trial 27 finished with value: 0.902 and parameters:
{'n_estimations': 90, 'prodondeur_maximum': 17, 'echantillons_min_div': 12, 'echa
ntillons_min_noeud': 6}. Best is trial 21 with value: 0.909.
[I 2024-07-01 15:10:02,848] Trial 28 finished with value: 0.8795 and parameters:
{'n_estimations': 35, 'prodondeur_maximum': 8, 'echantillons_min_div': 17, 'echan
tillons_min_noeud': 3}. Best is trial 21 with value: 0.909.
[I 2024-07-01 15:10:03,323] Trial 29 finished with value: 0.898 and parameters:
{'n_estimations': 179, 'prodondeur_maximum': 23, 'echantillons_min_div': 7, 'echa
ntillons_min_noeud': 10}. Best is trial 21 with value: 0.909.
[I 2024-07-01 15:10:03,794] Trial 30 finished with value: 0.9135 and parameters:
{'n_estimations': 169, 'prodondeur_maximum': 27, 'echantillons_min_div': 5, 'echa
ntillons_min_noeud': 2}. Best is trial 30 with value: 0.9135.
[I 2024-07-01 15:10:04,382] Trial 31 finished with value: 0.912 and parameters:
{'n_estimations': 167, 'prodondeur_maximum': 27, 'echantillons_min_div': 3, 'echa
ntillons_min_noeud': 2}. Best is trial 30 with value: 0.9135.
[I 2024-07-01 15:10:05,116] Trial 32 finished with value: 0.9105 and parameters:
{'n_estimations': 172, 'prodondeur_maximum': 28, 'echantillons_min_div': 2, 'echa
ntillons_min_noeud': 2}. Best is trial 30 with value: 0.9135.
[I 2024-07-01 15:10:05,834] Trial 33 finished with value: 0.91 and parameters:
{'n_estimations': 172, 'prodondeur_maximum': 31, 'echantillons_min_div': 2, 'echa
ntillons_min_noeud': 3}. Best is trial 30 with value: 0.9135.
[I 2024-07-01 15:10:06,371] Trial 34 finished with value: 0.9065 and parameters:
{'n_estimations': 174, 'prodondeur_maximum': 29, 'echantillons_min_div': 2, 'echa
ntillons_min_noeud': 4}. Best is trial 30 with value: 0.9135.
[I 2024-07-01 15:10:06,844] Trial 35 finished with value: 0.9095 and parameters:
{'n_estimations': 165, 'prodondeur_maximum': 27, 'echantillons_min_div': 4, 'echa
ntillons_min_noeud': 3}. Best is trial 30 with value: 0.9135.
[I 2024-07-01 15:10:07,334] Trial 36 finished with value: 0.8185 and parameters:
{'n_estimations': 191, 'prodondeur_maximum': 2, 'echantillons_min_div': 4, 'echan
tillons_min_noeud': 5}. Best is trial 30 with value: 0.9135.
[I 2024-07-01 15:10:08,243] Trial 37 finished with value: 0.9115 and parameters:
{'n_estimations': 264, 'prodondeur_maximum': 21, 'echantillons_min_div': 4, 'echa
ntillons_min_noeud': 2}. Best is trial 30 with value: 0.9135.
[I 2024-07-01 15:10:09,384] Trial 38 finished with value: 0.9115 and parameters:
{'n_estimations': 266, 'prodondeur_maximum': 21, 'echantillons_min_div': 4, 'echa
ntillons_min_noeud': 2}. Best is trial 30 with value: 0.9135.
[I 2024-07-01 15:10:10,246] Trial 39 finished with value: 0.902 and parameters:
{'n_estimations': 286, 'prodondeur_maximum': 21, 'echantillons_min_div': 4, 'echa
ntillons_min_noeud': 6}. Best is trial 30 with value: 0.9135.
```

```
[I 2024-07-01 15:10:11,034] Trial 40 finished with value: 0.9085 and parameters:
{'n_estimations': 262, 'prodondeur_maximum': 24, 'echantillons_min_div': 5, 'echa
ntillons_min_noeud': 2}. Best is trial 30 with value: 0.9135.
[I 2024-07-01 15:10:11,791] Trial 41 finished with value: 0.9125 and parameters:
{'n_estimations': 248, 'prodondeur_maximum': 25, 'echantillons_min_div': 3, 'echa
ntillons_min_noeud': 2}. Best is trial 30 with value: 0.9135.
[I 2024-07-01 15:10:12,547] Trial 42 finished with value: 0.906 and parameters:
{'n_estimations': 255, 'prodondeur_maximum': 20, 'echantillons_min_div': 3, 'echa
ntillons_min_noeud': 4}. Best is trial 30 with value: 0.9135.
[I 2024-07-01 15:10:13,398] Trial 43 finished with value: 0.9135 and parameters:
{'n_estimations': 252, 'prodondeur_maximum': 27, 'echantillons_min_div': 6, 'echa
ntillons_min_noeud': 2}. Best is trial 30 with value: 0.9135.
[I 2024-07-01 15:10:14,019] Trial 44 finished with value: 0.894 and parameters:
{'n_estimations': 239, 'prodondeur_maximum': 25, 'echantillons_min_div': 6, 'echa
ntillons_min_noeud': 17}. Best is trial 30 with value: 0.9135.
[I 2024-07-01 15:10:14,634] Trial 45 finished with value: 0.8945 and parameters:
{'n_estimations': 212, 'prodondeur_maximum': 28, 'echantillons_min_div': 6, 'echa
ntillons_min_noeud': 12}. Best is trial 30 with value: 0.9135.
[I 2024-07-01 15:10:15,771] Trial 46 finished with value: 0.9075 and parameters:
{'n_estimations': 297, 'prodondeur_maximum': 19, 'echantillons_min_div': 3, 'echa
ntillons_min_noeud': 4}. Best is trial 30 with value: 0.9135.
[I 2024-07-01 15:10:16,604] Trial 47 finished with value: 0.9125 and parameters:
{'n_estimations': 213, 'prodondeur_maximum': 26, 'echantillons_min_div': 5, 'echa
ntillons_min_noeud': 2}. Best is trial 30 with value: 0.9135.
[I 2024-07-01 15:10:17,340] Trial 48 finished with value: 0.8995 and parameters:
{'n_estimations': 214, 'prodondeur_maximum': 26, 'echantillons_min_div': 8, 'echa
ntillons_min_noeud': 6}. Best is trial 30 with value: 0.9135.
[I 2024-07-01 15:10:17,976] Trial 49 finished with value: 0.899 and parameters:
{'n_estimations': 193, 'prodondeur_maximum': 32, 'echantillons_min_div': 6, 'echa
ntillons_min_noeud': 8}. Best is trial 30 with value: 0.9135.
[I 2024-07-01 15:10:18,831] Trial 50 finished with value: 0.9105 and parameters:
{'n_estimations': 239, 'prodondeur_maximum': 24, 'echantillons_min_div': 5, 'echa
ntillons_min_noeud': 2}. Best is trial 30 with value: 0.9135.
[I 2024-07-01 15:10:19,867] Trial 51 finished with value: 0.914 and parameters:
{'n_estimations': 276, 'prodondeur_maximum': 27, 'echantillons_min_div': 5, 'echa
ntillons_min_noeud': 2}. Best is trial 51 with value: 0.914.
[I 2024-07-01 15:10:20,741] Trial 52 finished with value: 0.907 and parameters:
{'n_estimations': 277, 'prodondeur_maximum': 27, 'echantillons_min_div': 5, 'echa
ntillons_min_noeud': 3}. Best is trial 51 with value: 0.914.
[I 2024-07-01 15:10:21,539] Trial 53 finished with value: 0.9085 and parameters:
{'n_estimations': 249, 'prodondeur_maximum': 19, 'echantillons_min_div': 3, 'echa
ntillons_min_noeud': 4}. Best is trial 51 with value: 0.914.
[I 2024-07-01 15:10:22,075] Trial 54 finished with value: 0.834 and parameters:
{'n_estimations': 220, 'prodondeur_maximum': 3, 'echantillons_min_div': 8, 'echan
tillons_min_noeud': 15}. Best is trial 51 with value: 0.914.
[I 2024-07-01 15:10:22,763] Trial 55 finished with value: 0.912 and parameters:
{'n_estimations': 202, 'prodondeur_maximum': 32, 'echantillons_min_div': 5, 'echa
ntillons_min_noeud': 2}. Best is trial 51 with value: 0.914.
[I 2024-07-01 15:10:23,312] Trial 56 finished with value: 0.9155 and parameters:
{'n_estimations': 154, 'prodondeur_maximum': 23, 'echantillons_min_div': 3, 'echa
ntillons_min_noeud': 1}. Best is trial 56 with value: 0.9155.
[I 2024-07-01 15:10:23,754] Trial 57 finished with value: 0.885 and parameters:
{'n_estimations': 149, 'prodondeur_maximum': 12, 'echantillons_min_div': 8, 'echa
ntillons_min_noeud': 19}. Best is trial 56 with value: 0.9155.
[I 2024-07-01 15:10:24,265] Trial 58 finished with value: 0.911 and parameters:
{'n_estimations': 136, 'prodondeur_maximum': 23, 'echantillons_min_div': 7, 'echa
ntillons_min_noeud': 1}. Best is trial 56 with value: 0.9155.
[I 2024-07-01 15:10:25,179] Trial 59 finished with value: 0.8985 and parameters:
{'n_estimations': 279, 'prodondeur_maximum': 18, 'echantillons_min_div': 20, 'ech
antillons_min_noeud': 5}. Best is trial 56 with value: 0.9155.
```

```
[I 2024-07-01 15:10:26,003] Trial 60 finished with value: 0.9085 and parameters:
{'n_estimations': 228, 'prodondeur_maximum': 16, 'echantillons_min_div': 6, 'echa
ntillons_min_noeud': 1}. Best is trial 56 with value: 0.9155.
[I 2024-07-01 15:10:26,554] Trial 61 finished with value: 0.913 and parameters:
{'n_estimations': 190, 'prodondeur_maximum': 29, 'echantillons_min_div': 3, 'echa
ntillons_min_noeud': 2}. Best is trial 56 with value: 0.9155.
[I 2024-07-01 15:10:27,016] Trial 62 finished with value: 0.9095 and parameters:
{'n_estimations': 157, 'prodondeur_maximum': 29, 'echantillons_min_div': 3, 'echa
ntillons_min_noeud': 3}. Best is trial 56 with value: 0.9155.
[I 2024-07-01 15:10:27,660] Trial 63 finished with value: 0.915 and parameters:
{'n_estimations': 187, 'prodondeur_maximum': 25, 'echantillons_min_div': 5, 'echa
ntillons_min_noeud': 1}. Best is trial 56 with value: 0.9155.
[I 2024-07-01 15:10:28,118] Trial 64 finished with value: 0.854 and parameters:
{'n_estimations': 190, 'prodondeur_maximum': 4, 'echantillons_min_div': 2, 'echan
tillons_min_noeud': 1}. Best is trial 56 with value: 0.9155.
[I 2024-07-01 15:10:28,501] Trial 65 finished with value: 0.9085 and parameters:
{'n_estimations': 134, 'prodondeur_maximum': 25, 'echantillons_min_div': 3, 'echa
ntillons_min_noeud': 3}. Best is trial 56 with value: 0.9155.
[I 2024-07-01 15:10:29,140] Trial 66 finished with value: 0.9125 and parameters:
{'n_estimations': 180, 'prodondeur_maximum': 30, 'echantillons_min_div': 4, 'echa
ntillons_min_noeud': 1}. Best is trial 56 with value: 0.9155.
[I 2024-07-01 15:10:29,864] Trial 67 finished with value: 0.907 and parameters:
{'n_estimations': 244, 'prodondeur_maximum': 22, 'echantillons_min_div': 5, 'echa
ntillons_min_noeud': 4}. Best is trial 56 with value: 0.9155.
[I 2024-07-01 15:10:30,308] Trial 68 finished with value: 0.9055 and parameters:
{'n_estimations': 116, 'prodondeur_maximum': 14, 'echantillons_min_div': 7, 'echa
ntillons_min_noeud': 2}. Best is trial 56 with value: 0.9155.
[I 2024-07-01 15:10:30,937] Trial 69 finished with value: 0.9155 and parameters:
{'n_estimations': 156, 'prodondeur_maximum': 23, 'echantillons_min_div': 3, 'echa
ntillons_min_noeud': 1}. Best is trial 56 with value: 0.9155.
[I 2024-07-01 15:10:31,503] Trial 70 finished with value: 0.9105 and parameters:
{'n_estimations': 154, 'prodondeur_maximum': 22, 'echantillons_min_div': 4, 'echa
ntillons_min_noeud': 1}. Best is trial 56 with value: 0.9155.
[I 2024-07-01 15:10:32,284] Trial 71 finished with value: 0.9105 and parameters:
{'n_estimations': 163, 'prodondeur_maximum': 24, 'echantillons_min_div': 3, 'echa
ntillons_min_noeud': 1}. Best is trial 56 with value: 0.9155.
[I 2024-07-01 15:10:32,852] Trial 72 finished with value: 0.909 and parameters:
{'n_estimations': 202, 'prodondeur_maximum': 19, 'echantillons_min_div': 2, 'echa
ntillons_min_noeud': 3}. Best is trial 56 with value: 0.9155.
[I 2024-07-01 15:10:33,469] Trial 73 finished with value: 0.9135 and parameters:
{'n_estimations': 182, 'prodondeur_maximum': 29, 'echantillons_min_div': 2, 'echa
ntillons_min_noeud': 2}. Best is trial 56 with value: 0.9155.
[I 2024-07-01 15:10:34,091] Trial 74 finished with value: 0.9135 and parameters:
{'n_estimations': 181, 'prodondeur_maximum': 29, 'echantillons_min_div': 2, 'echa
ntillons_min_noeud': 1}. Best is trial 56 with value: 0.9155.
[I 2024-07-01 15:10:34,652] Trial 75 finished with value: 0.913 and parameters:
{'n_estimations': 180, 'prodondeur_maximum': 31, 'echantillons_min_div': 2, 'echa
ntillons_min_noeud': 1}. Best is trial 56 with value: 0.9155.
[I 2024-07-01 15:10:34,969] Trial 76 finished with value: 0.815 and parameters:
{'n_estimations': 146, 'prodondeur_maximum': 2, 'echantillons_min_div': 2, 'echan
tillons_min_noeud': 3}. Best is trial 56 with value: 0.9155.
[I 2024-07-01 15:10:35,449] Trial 77 finished with value: 0.9115 and parameters:
{'n_estimations': 155, 'prodondeur_maximum': 28, 'echantillons_min_div': 4, 'echa
ntillons_min_noeud': 1}. Best is trial 56 with value: 0.9155.
[I 2024-07-01 15:10:35,953] Trial 78 finished with value: 0.9 and parameters: {'n
_estimations': 127, 'prodondeur_maximum': 23, 'echantillons_min_div': 5, 'echanti
llons_min_noeud': 9}. Best is trial 56 with value: 0.9155.
[I 2024-07-01 15:10:36,374] Trial 79 finished with value: 0.9045 and parameters:
{'n_estimations': 104, 'prodondeur_maximum': 17, 'echantillons_min_div': 6, 'echa
ntillons_min_noeud': 4}. Best is trial 56 with value: 0.9155.
```

```
[I 2024-07-01 15:10:36,829] Trial 80 finished with value: 0.91 and parameters:
{'n_estimations': 137, 'prodondeur_maximum': 21, 'echantillons_min_div': 4, 'echa
ntillons_min_noeud': 2}. Best is trial 56 with value: 0.9155.
[I 2024-07-01 15:10:37,553] Trial 81 finished with value: 0.9115 and parameters:
{'n_estimations': 189, 'prodondeur_maximum': 30, 'echantillons_min_div': 3, 'echa
ntillons_min_noeud': 2}. Best is trial 56 with value: 0.9155.
[I 2024-07-01 15:10:38,124] Trial 82 finished with value: 0.9145 and parameters:
{'n_estimations': 185, 'prodondeur_maximum': 29, 'echantillons_min_div': 2, 'echa
ntillons_min_noeud': 1}. Best is trial 56 with value: 0.9155.
[I 2024-07-01 15:10:38,726] Trial 83 finished with value: 0.91 and parameters:
{'n_estimations': 182, 'prodondeur_maximum': 26, 'echantillons_min_div': 2, 'echa
ntillons_min_noeud': 1}. Best is trial 56 with value: 0.9155.
[I 2024-07-01 15:10:39,433] Trial 84 finished with value: 0.895 and parameters:
{'n_estimations': 170, 'prodondeur_maximum': 32, 'echantillons_min_div': 2, 'echa
ntillons_min_noeud': 12}. Best is trial 56 with value: 0.9155.
[I 2024-07-01 15:10:39,924] Trial 85 finished with value: 0.91 and parameters:
{'n_estimations': 160, 'prodondeur_maximum': 27, 'echantillons_min_div': 3, 'echa
ntillons_min_noeud': 3}. Best is trial 56 with value: 0.9155.
[I 2024-07-01 15:10:40,611] Trial 86 finished with value: 0.9145 and parameters:
{'n_estimations': 200, 'prodondeur_maximum': 20, 'echantillons_min_div': 4, 'echa
ntillons_min_noeud': 1}. Best is trial 56 with value: 0.9155.
[I 2024-07-01 15:10:41,226] Trial 87 finished with value: 0.9095 and parameters:
{'n_estimations': 201, 'prodondeur_maximum': 20, 'echantillons_min_div': 4, 'echa
ntillons_min_noeud': 2}. Best is trial 56 with value: 0.9155.
[I 2024-07-01 15:10:41,919] Trial 88 finished with value: 0.9145 and parameters:
{'n_estimations': 221, 'prodondeur_maximum': 24, 'echantillons_min_div': 6, 'echa
ntillons_min_noeud': 1}. Best is trial 56 with value: 0.9155.
[I 2024-07-01 15:10:42,913] Trial 89 finished with value: 0.911 and parameters:
{'n_estimations': 224, 'prodondeur_maximum': 18, 'echantillons_min_div': 6, 'echa
ntillons_min_noeud': 1}. Best is trial 56 with value: 0.9155.
[I 2024-07-01 15:10:43,570] Trial 90 finished with value: 0.8985 and parameters:
{'n_estimations': 235, 'prodondeur_maximum': 11, 'echantillons_min_div': 5, 'echa
ntillons_min_noeud': 3}. Best is trial 56 with value: 0.9155.
[I 2024-07-01 15:10:44,240] Trial 91 finished with value: 0.9085 and parameters:
{'n_estimations': 196, 'prodondeur_maximum': 23, 'echantillons_min_div': 7, 'echa
ntillons_min_noeud': 2}. Best is trial 56 with value: 0.9155.
[I 2024-07-01 15:10:44,907] Trial 92 finished with value: 0.915 and parameters:
{'n_estimations': 205, 'prodondeur_maximum': 25, 'echantillons_min_div': 4, 'echa
ntillons_min_noeud': 1}. Best is trial 56 with value: 0.9155.
[I 2024-07-01 15:10:45,567] Trial 93 finished with value: 0.915 and parameters:
{'n_estimations': 217, 'prodondeur_maximum': 25, 'echantillons_min_div': 5, 'echa
ntillons_min_noeud': 1}. Best is trial 56 with value: 0.9155.
[I 2024-07-01 15:10:46,236] Trial 94 finished with value: 0.915 and parameters:
{'n_estimations': 208, 'prodondeur_maximum': 25, 'echantillons_min_div': 5, 'echa
ntillons_min_noeud': 1}. Best is trial 56 with value: 0.9155.
[I 2024-07-01 15:10:46,800] Trial 95 finished with value: 0.8845 and parameters:
{'n_estimations': 218, 'prodondeur_maximum': 8, 'echantillons_min_div': 4, 'echan
tillons_min_noeud': 1}. Best is trial 56 with value: 0.9155.
[I 2024-07-01 15:10:47,438] Trial 96 finished with value: 0.9145 and parameters:
{'n_estimations': 207, 'prodondeur_maximum': 20, 'echantillons_min_div': 4, 'echa
ntillons_min_noeud': 1}. Best is trial 56 with value: 0.9155.
[I 2024-07-01 15:10:48,154] Trial 97 finished with value: 0.9085 and parameters:
{'n_estimations': 208, 'prodondeur_maximum': 16, 'echantillons_min_div': 4, 'echa
ntillons_min_noeud': 1}. Best is trial 56 with value: 0.9155.
[I 2024-07-01 15:10:48,921] Trial 98 finished with value: 0.912 and parameters:
{'n_estimations': 210, 'prodondeur_maximum': 20, 'echantillons_min_div': 5, 'echa
ntillons_min_noeud': 1}. Best is trial 56 with value: 0.9155.
[I 2024-07-01 15:10:49,626] Trial 99 finished with value: 0.912 and parameters:
{'n_estimations': 222, 'prodondeur_maximum': 25, 'echantillons_min_div': 4, 'echa
ntillons_min_noeud': 1}. Best is trial 56 with value: 0.9155.
```

Entraînement du modèle.

```
In [ ]: model = RandomForestClassifier(n_estimators=resultat['n_estimations'], max_depth
        model.fit(X_entrainement, Y_entrainement)
```

```
Out[ ]: ▾                    RandomForestClassifier

        RandomForestClassifier(max_depth=23, min_samples_split=3, n_estimators=
        154,
                               n_jobs=-1, random_state=42)
```

Évaluation de la performance du modèle.

```
In [ ]: y_prediction = model.predict(X_test)
        print(classification_report(y_test, y_prediction))
```

```
              precision    recall  f1-score   support

           0       0.93      0.90      0.91      1000
           1       0.90      0.93      0.92      1000

    accuracy                           0.92      2000
   macro avg       0.92      0.92      0.92      2000
weighted avg       0.92      0.92      0.92      2000
```