Personal vs promotional email

Students
Adham Khaled
Omar Ashraf
Aboubakr Elnaggar
Abdelrahman Atef

Introduction to the dataset

- We are required to take the given dataset and train the model to reach an end goal of classifying emails into personal or promotional (spam) when dealing with a load of mixed emails. In the end, the model classifies the email type into binary based classification where personal emails are labeled “1” and promotional emails are labeled "0”.

Motivation

Our end goal is to reach a binary based classification of 1's and 0's depending on the type of email.

Our Objectives

- Checking and handling class imbalances.,
- Detecting outliers.,
- Handing missing values.,
- Applying normalization, SMOTE, PCA, supervised (in our case Naive bayes) and/or unsupervised techniques,
- Compare between the two datasets in the end: raw vs. processed.

Dataset (Email types) : Data cleaning (train)

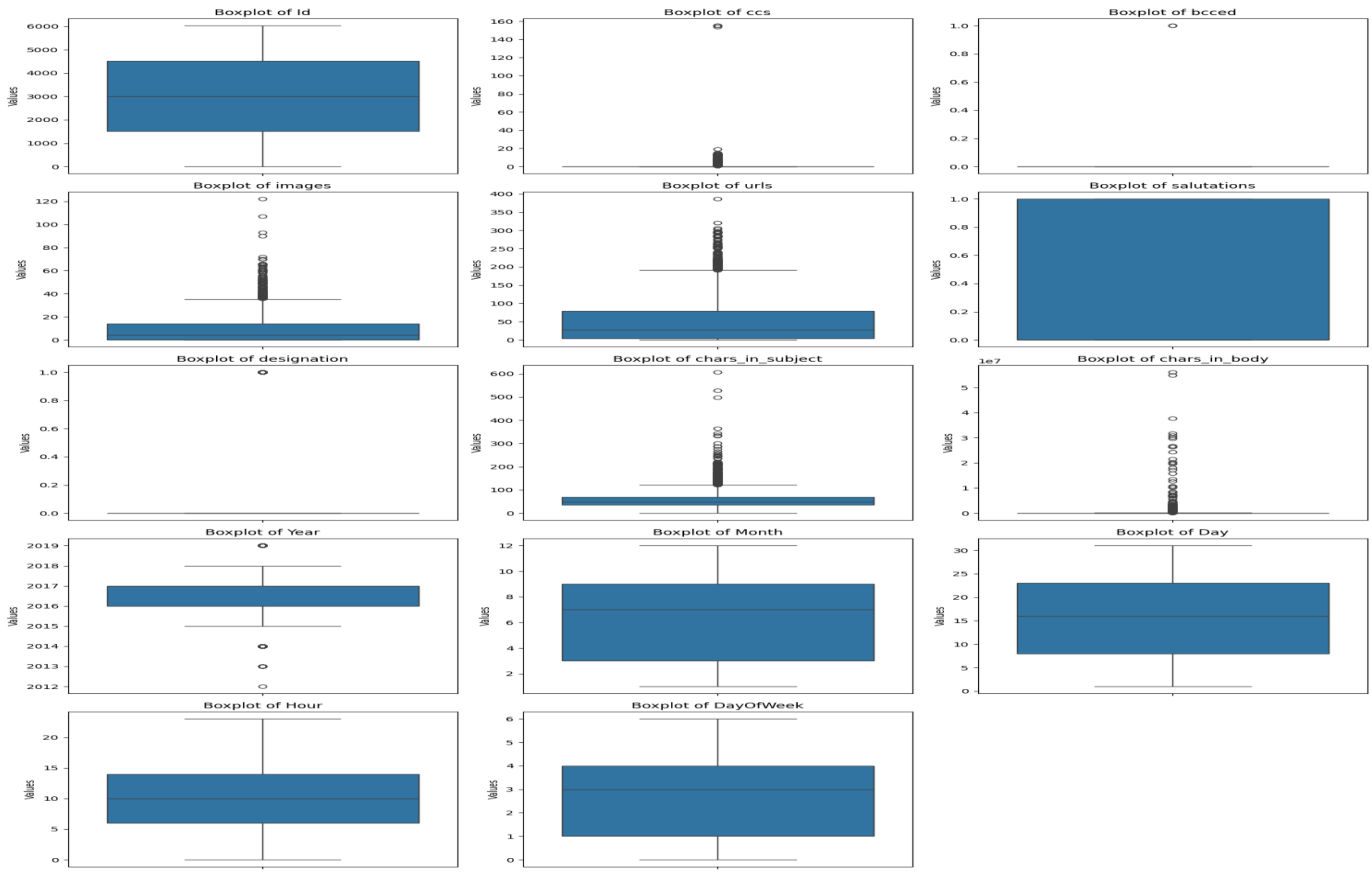
- First, the data consists of 14 columns
- (4 categorical columns)
- (10 numerical columns)
- We checked now in columns (org ,tld) they got missing values (114 ,114)
- so we handle categorical columns with mode and numerical columns with median.

Detecting outlier usinG IQR

As we can see , There is outliers need to be handled

```
Count of outliers in Id: 0
Count of outliers in ccs: 2907
Count of outliers in bcced: 30
Count of outliers in images: 578
Count of outliers in urls: 453
Count of outliers in salutations: 0
Count of outliers in designation: 1862
Count of outliers in chars_in_subject: 698
Count of outliers in chars_in_body: 827
Count of outliers in label: 0
Count of outliers in Year: 1202
Count of outliers in Month: 0
Count of outliers in Day: 0
Count of outliers in Hour: 0
Count of outliers in DayOfWeek: 0
```

Outliers

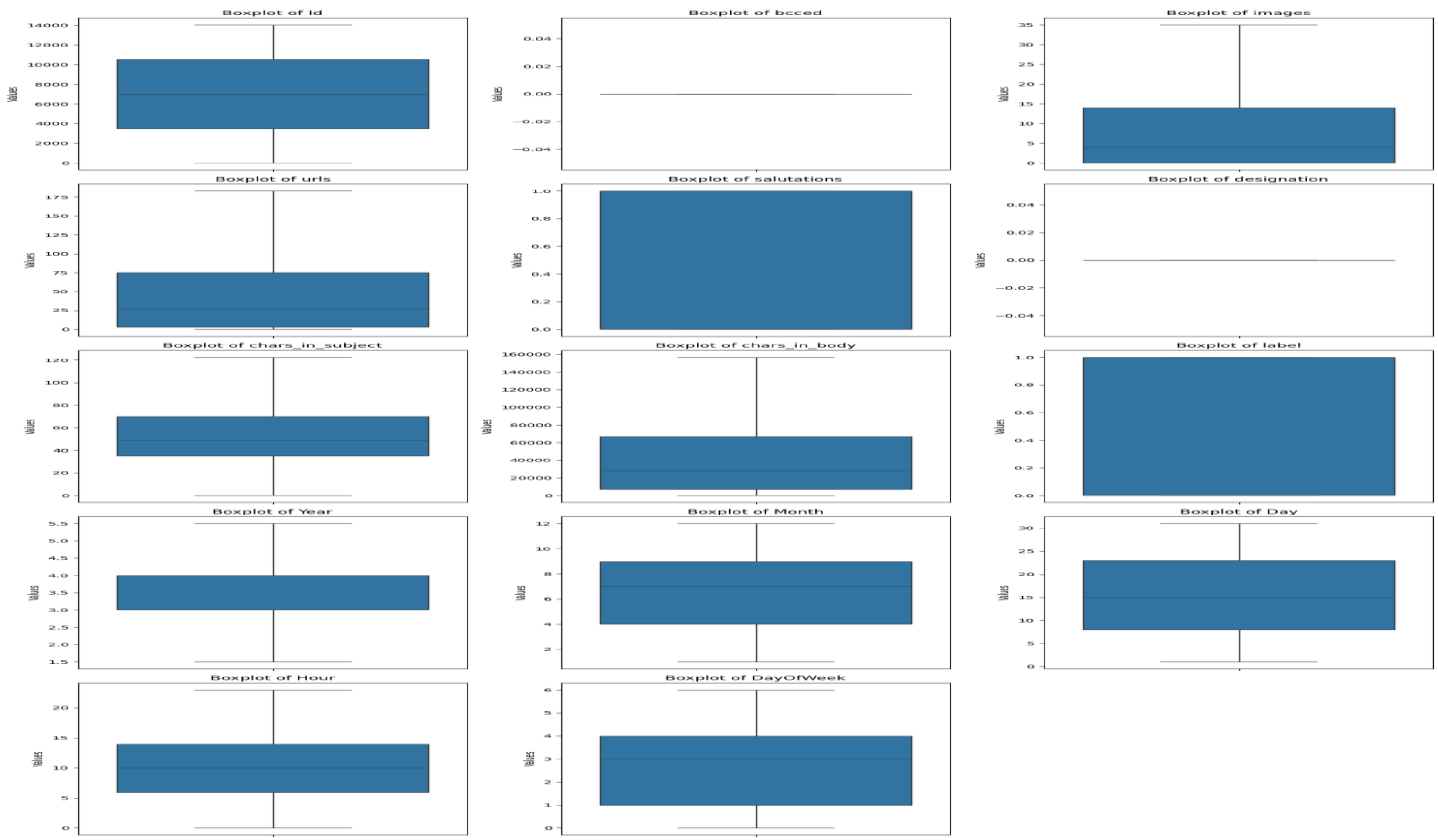

Boxplot of Id
Boxplot of ccs
Boxplot of bcced
Boxplot of images
Boxplot of urls
Boxplot of salutations
Boxplot of designation
Boxplot of chars_in_subject
Boxplot of chars_in_body
1e7
Boxplot of Year
Boxplot of Month
Boxplot of Day
Boxplot of Hour
Boxplot of DayOfWeek
Values

Handling outliers

Now as you can see we handled the outliers using capping.

```
Count of outliers in id: 0
Count of outliers in bcced: 0
Count of outliers in images: 0
Count of outliers in urls: 0
Count of outliers in salutations: 0
Count of outliers in designation: 0
Count of outliers in chars_in_subject: 0
Count of outliers in chars_in_body: 0
Count of outliers in label: 0
Count of outliers in Year: 0
Count of outliers in Month: 0
Count of outliers in Day: 0
Count of outliers in Hour: 0
Count of outliers in DayOfWeek: 0
```

Verify using graphs(outliers)



Normalization on train and test as well

we applied Yeo-Johnson transformation on numerical columns to normalize and reduce skewness and improve performance as well.

Test Set After Normalization:

	ccs	images	urls	chars_in_subject	chars_in_body
count	6029.0	6029.000000	6029.000000	6029.000000	6029.000000
mean	0.0	-0.001044	0.016529	0.000596	-0.004379
std	0.0	1.002202	1.002415	1.004758	1.004406
min	0.0	-1.191355	-1.508590	-4.510545	-2.159307
25%	0.0	-1.191355	-0.869068	-0.678523	-0.823975
50%	0.0	0.059020	0.102337	-0.072926	0.070131
75%	0.0	0.895282	0.891244	0.643635	0.802439
max	0.0	1.551845	1.653133	1.941880	1.739969

	Year	Month	Day	Hour	DayOfWeek
count	6.029000e+03	6029.000000	6029.000000	6029.000000	6029.000000
mean	1.796116e+00	-0.035223	0.031137	0.007097	-0.016520
std	1.974140e-13	1.001933	0.986558	1.008567	1.001178
min	1.796116e+00	-1.733809	-1.903015	-2.063418	-1.597310
25%	1.796116e+00	-1.044910	-0.784597	-0.642795	-0.831133
50%	1.796116e+00	0.179001	0.151785	0.078181	0.309604
75%	1.796116e+00	0.748038	0.853454	0.726228	0.784536
max	1.796116e+00	1.566710	1.576625	2.027307	1.626227

Mean and Std Comparison:

	Train Mean	Test Mean	Train Std	Test Std
ccs	2.549004e-01	0.000000	0.553829	0.000000e+00
images	3.940723e-17	-0.001044	1.000036	1.002202e+00
urls	3.536547e-17	0.016529	1.000036	1.002415e+00
chars_in_subject	-1.869317e-17	0.000596	1.000036	1.004758e+00
chars_in_body	2.172450e-17	-0.004379	1.000036	1.004406e+00
Year	-6.567872e-18	1.796116	1.000036	1.974140e-13
Month	1.212530e-17	-0.035223	1.000036	1.001933e+00
Day	-1.212530e-17	0.031137	1.000036	9.865583e-01
Hour	2.576627e-17	0.007097	1.000036	1.008567e+00
DayOfWeek	1.515663e-18	-0.016520	1.000036	1.001178e+00

Training Set After Normalization:

	ccs	images	urls	chars_in_subject
count	14064.000000	1.406400e+04	1.406400e+04	1.406400e+04
mean	0.254900	3.940723e-17	3.536547e-17	-1.869317e-17
std	0.553829	1.000036e+00	1.000036e+00	1.000036e+00
min	0.000000	-1.191355e+00	-1.508590e+00	-4.510545e+00
25%	0.000000	-1.191355e+00	-9.711283e-01	-6.785235e-01
50%	0.000000	5.902015e-02	1.023373e-01	-7.292614e-02
75%	0.000000	8.952824e-01	8.485319e-01	6.436346e-01
max	5.049856	1.551845e+00	1.653133e+00	1.941880e+00

	chars_in_body	Year	Month	Day	Hour
count	1.406400e+04	1.406400e+04	1.406400e+04	1.406400e+04	1.406400e+04
mean	2.172450e-17	-6.567872e-18	1.212530e-17	-1.212530e-17	2.576627e-17
std	1.000036e+00	1.000036e+00	1.000036e+00	1.000036e+00	1.000036e+00
min	-2.159307e+00	-1.927426e+00	-1.733809e+00	-1.903015e+00	-2.063418e+00
25%	-8.026675e-01	-5.195784e-01	-7.246180e-01	-7.845970e-01	-6.427954e-01
50%	5.322673e-02	4.104378e-01	1.790009e-01	4.434110e-02	7.818067e-02
75%	7.804405e-01	4.104378e-01	7.480377e-01	8.534535e-01	7.262284e-01
max	1.739969e+00	1.796116e+00	1.566710e+00	1.576625e+00	2.027307e+00

	DayOfWeek
count	1.406400e+04
mean	1.515663e-18
std	1.000036e+00
min	-1.597310e+00
25%	-8.311335e-01
50%	3.096041e-01
75%	7.845357e-01
max	1.626227e+00

plotting nomalization train (after)

plotting nomalization test(after)

Train and Test after normalization (states)

- values after normalization and after we do this we assign this to our dataframe train and test.

```
Train Norm Stats:
             images          urls  chars_in_subject  chars_in_body  \
count  1.406400e+04  1.406400e+04      1.406400e+04   1.406400e+04
mean   3.940723e-17  3.536547e-17     -1.869317e-17   2.172450e-17
std    1.000036e+00  1.000036e+00      1.000036e+00   1.000036e+00
min   -1.191355e+00 -1.508590e+00     -4.510545e+00  -2.159307e+00
25%   -1.191355e+00 -9.711283e-01     -6.785235e-01  -8.026675e-01
50%    5.902015e-02  1.023373e-01     -7.292614e-02   5.322673e-02
75%    8.952824e-01  8.485319e-01      6.436346e-01   7.804405e-01
max    1.551845e+00  1.653133e+00      1.941880e+00   1.739969e+00

               Year         Month           Day          Hour     DayOfWeek
count  1.406400e+04  1.406400e+04  1.406400e+04  1.406400e+04  1.406400e+04
mean  -6.567872e-18  1.212530e-17 -1.212530e-17  2.576627e-17  1.515663e-18
std    1.000036e+00  1.000036e+00  1.000036e+00  1.000036e+00  1.000036e+00
min   -1.927426e+00 -1.733809e+00 -1.903015e+00 -2.063418e+00 -1.597310e+00
25%   -5.195784e-01 -7.246180e-01 -7.845970e-01 -6.427954e-01 -8.311335e-01
50%    4.104378e-01  1.790009e-01  4.434110e-02  7.818067e-02  3.096041e-01
75%    4.104378e-01  7.480377e-01  8.534535e-01  7.262284e-01  7.845357e-01
max    1.796116e+00  1.566710e+00  1.576625e+00  2.027307e+00  1.626227e+00
Test Norm Stats:
            images         urls  chars_in_subject  chars_in_body  \
count  6029.000000  6029.000000       6029.000000    6029.000000
mean     -0.001044     0.016529          0.000596      -0.004379
std       1.002202     1.002415          1.004758       1.004406
min      -1.191355    -1.508590         -4.510545      -2.159307
25%      -1.191355    -0.869068         -0.678523      -0.823975
50%       0.059020     0.102337         -0.072926       0.070131
75%       0.895282     0.891244          0.643635       0.802439
max       1.551845     1.653133          1.941880       1.739969

               Year        Month          Day         Hour    DayOfWeek
count  6.029000e+03  6029.000000  6029.000000  6029.000000  6029.000000
mean   1.796116e+00    -0.035223     0.031137     0.007097    -0.016520
std    1.974140e-13     1.001933     0.986558     1.008567     1.001178
min    1.796116e+00    -1.733809    -1.903015    -2.063418    -1.597310
```

One hot encoding for train

- Now we used one hot encoding to change the categorical columns to binary columns (numerical) in order to let mL model process them.

tld_vnet.ibm.com	tld_wfp.org	tld_xoom.com	mail_type_Multipart/Mixed	mail_type_Text/Html
False	False	False	False	False
False	False	False	False	False
False	False	False	False	False
False	False	False	False	False
False	False	False	False	False
...	...	...	...	...
False	False	False	False	False
False	False	False	False	False
False	False	False	False	False
False	False	False	False	False
False	False	False	False	False

One hot encoding (test)

- Now we used one hot encoding to change the categorical columns to binary columns (numerical) in order to let mL model process them same as train.

tld_vnet.ibm.com	tld_wfp.org	tld_xoom.com	mail_type_Multipart/Mixed	mail_type_Text/Html
False	False	False	False	False
False	False	False	False	False
False	False	False	False	False
False	False	False	False	False
False	False	False	False	False
...	...	...	...	...
False	False	False	False	False
False	False	False	False	False
False	False	False	False	False
False	False	False	False	False
False	False	False	False	False

Data preprocessing

- Now we will compare strong correlation with our target column Label (Images,urls)

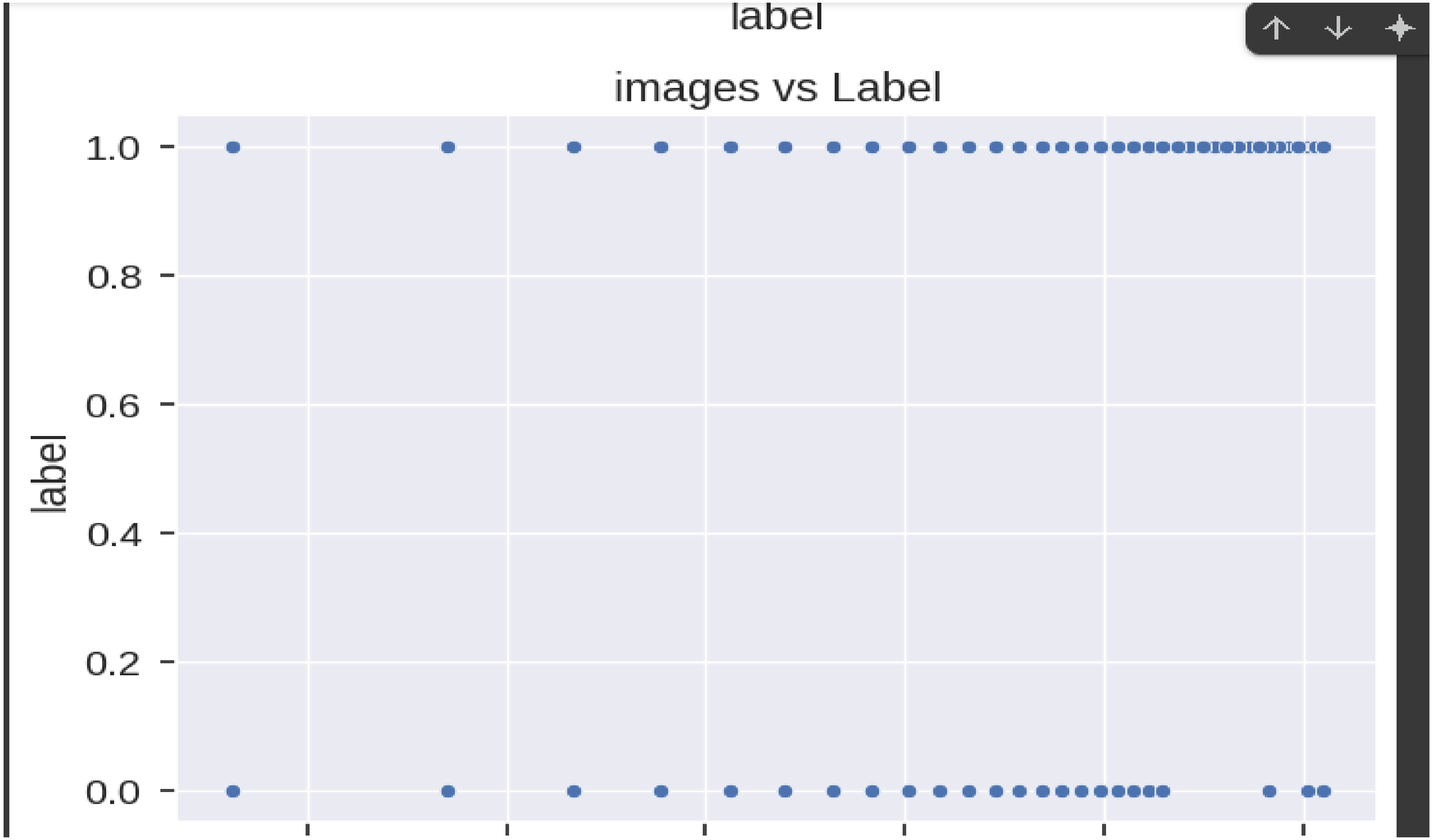
	label
label	1.000000
images	0.754686
urls	0.712750
tld_com	0.545389
chars_in_body	0.305947
chars_in_subject	0.238612
tld_in	0.212086
org_amazon	0.192694
org_linkedin	0.192023
org_twitter	0.186824
Year	0.185675

Data preprocessing

- Showing strong negative correlations.

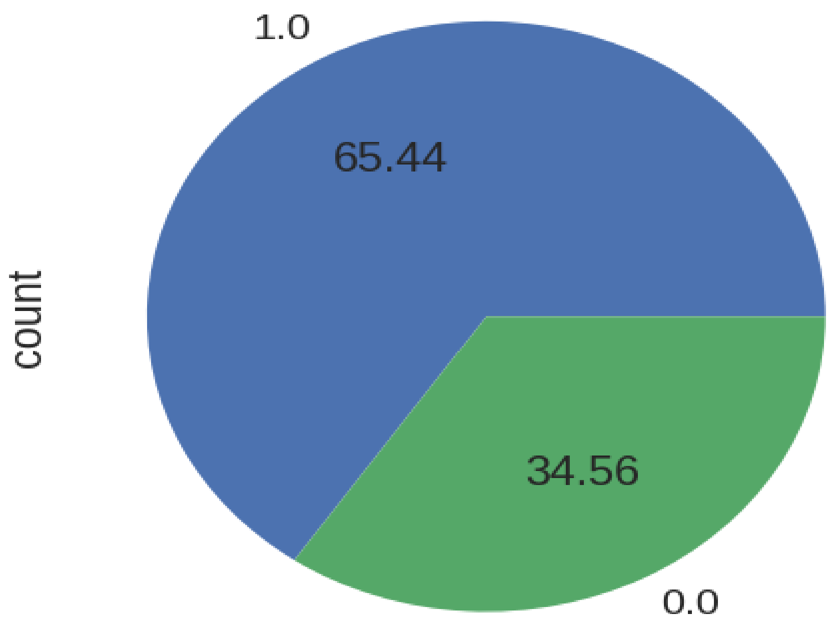
	label
org_iiitd	-0.926147
tld_ac.in	-0.913160
ccs	-0.629099
salutations	-0.418616
mail_type_multipart/related	-0.220166
org_centralesupelec	-0.175473
tld_fr	-0.169779
org_github	-0.145747
mail_type_text/plain	-0.136616
mail_type_multipart/mixed	-0.116402
Month	-0.055773

- Now we visulaize the strong correlation features with our target column (Label)



Handle class imbalancing

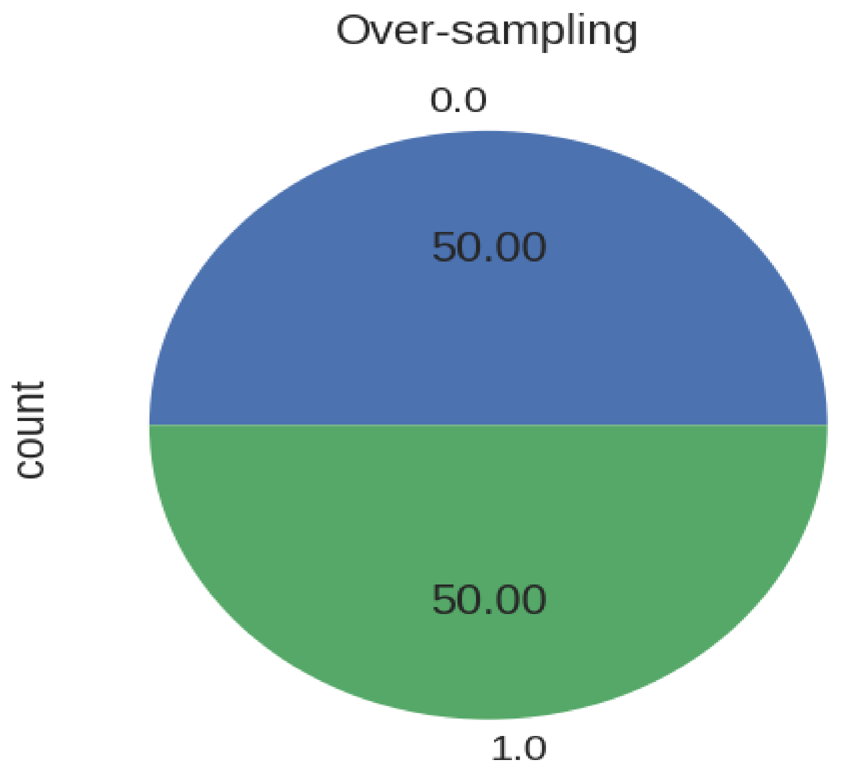
- Now as we see number of 1’s are higher than 0’s. Data is not balanced.



label	count
1.0	9204
0.0	4860

dtype: int64

- So we used SMOTE to handle class imbalance. We interpolated between existing ones and prevent overfitting.



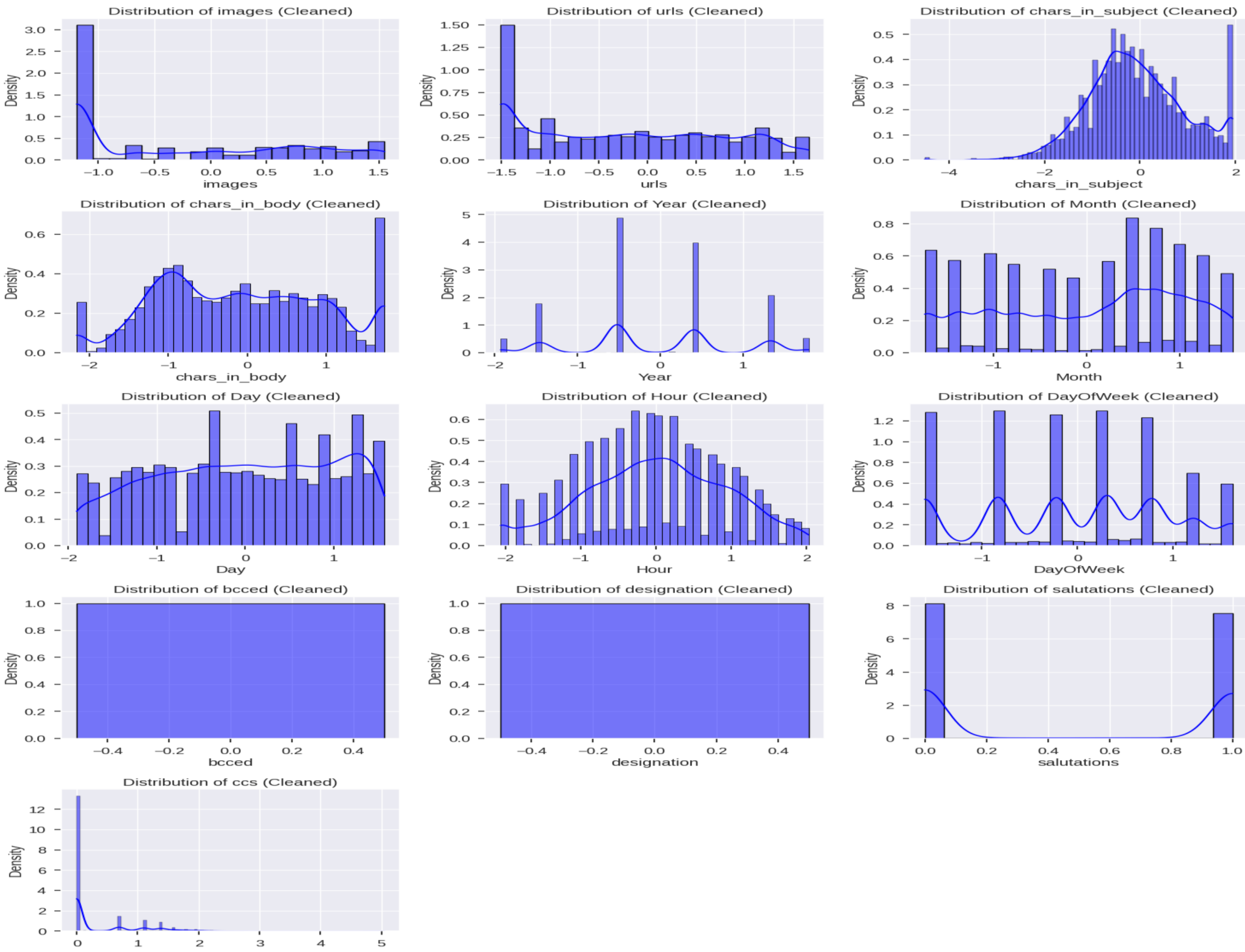
EDA

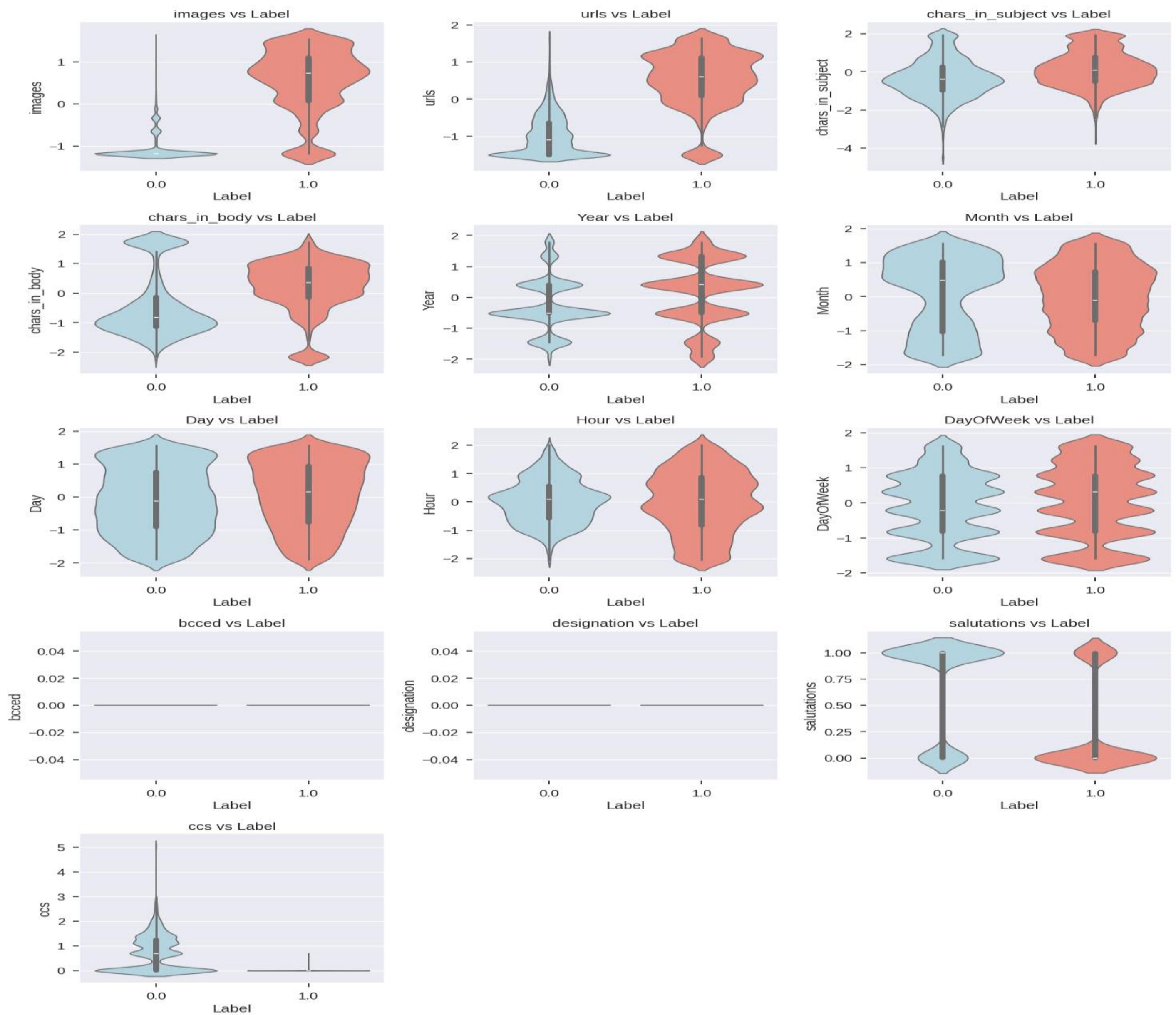
- EDA calculations.

```
Numerical Summary:
             images          urls  chars_in_subject  chars_in_body  \
count  18408.000000  18408.000000      18408.000000   18408.000000
mean      -0.248912     -0.233614         -0.074819      -0.100031
std        0.993751      1.002566          1.010118       1.033662
min       -1.191355     -1.508590         -4.510545      -2.159307
25%       -1.191355     -1.259350         -0.727704      -0.942047
50%       -0.649128     -0.290697         -0.151503      -0.150652
75%        0.726545      0.617305          0.582353       0.717951
max        1.551845      1.653133          1.941880       1.739969

               Year         Month           Day          Hour     DayOfWeek  \
count  18408.000000  18408.000000  18408.000000  18408.000000  18408.000000
mean      -0.057437      0.020802     -0.026834      0.004799     -0.024131
std        0.967919      1.016581      0.995938      0.943993      0.985443
min       -1.927426     -1.733809     -1.903015     -2.063418     -1.597310
25%       -0.519578     -0.869958     -0.808714     -0.642795     -0.831133
50%       -0.519578      0.179001      0.044341      0.078181      0.043594
75%        0.410438      0.809818      0.853454      0.726228      0.784536
max        1.796116      1.566710      1.576625      2.027307      1.626227

         bcced  designation   salutations           ccs         label
count  18408.0      18408.0  18408.000000  18408.000000  18408.000000
mean       0.0          0.0      0.480856      0.361067      0.500000
std        0.0          0.0      0.495867      0.620622      0.500014
min        0.0          0.0      0.000000      0.000000      0.000000
25%        0.0          0.0      0.000000      0.000000      0.000000
50%        0.0          0.0      0.000000      0.000000      0.500000
75%        0.0          0.0      1.000000      0.693147      1.000000
max        0.0          0.0      1.000000      5.049856      1.000000

Categorical Summary:
       org_126 org_3dr org_InsideApple org_Magento org_Mentor  \
count    18408   18408           18408       18408      18408
unique       2       2               2           2          2
top      False   False           False       False      False
freq     18404   18407           18406       18407      18405

       org_Neotechnology org_ZOOHIVERSE org_academia-mail org_agencenavigo  \
count              18408          18408             18408            18408
unique                 2              2                 2                2
top                False          False             False            False
freq               18405          18405             18373            18407
```

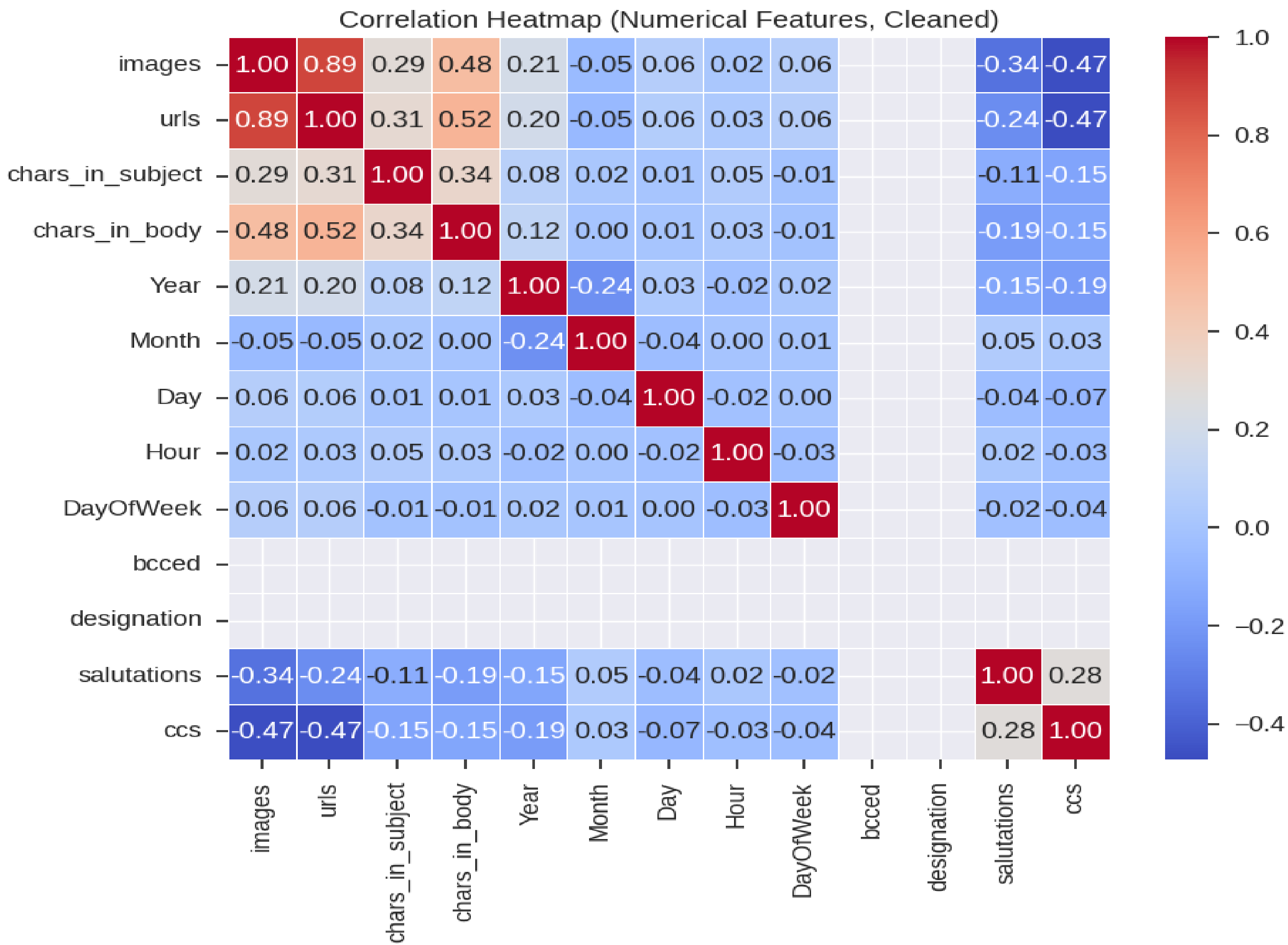
Histogram of EDA numerical

- We analyzed the distribution of (numerical features) using histogram and KDE plots to detect skewness.we visualized each feature against the target variable using violin plots to explore their separability.


Distribution of images (Cleaned)
images
Density
Distribution of urls (Cleaned)
urls
Density
Distribution of chars_in_subject (Cleaned)
chars_in_subject
Density
Distribution of chars_in_body (Cleaned)
chars_in_body
Density
Distribution of Year (Cleaned)
Year
Density
Distribution of Month (Cleaned)
Month
Density
Distribution of Day (Cleaned)
Day
Density
Distribution of Hour (Cleaned)
Hour
Density
Distribution of DayOfWeek (Cleaned)
DayOfWeek
Density
Distribution of bcced (Cleaned)
bcced
Density
Distribution of designation (Cleaned)
designation
Density
Distribution of salutations (Cleaned)
salutations
Density
Distribution of ccs (Cleaned)
ccs
Density


images vs Label
urls vs Label
chars_in_subject vs Label
chars_in_body vs Label
Year vs Label
Month vs Label
Day vs Label
Hour vs Label
DayOfWeek vs Label
bcced vs Label
designation vs Label
salutations vs Label
ccs vs Label
Label
0.0
1.0

Correlation heatmap



PCA (test and train)

- now we will use Aggregation merging 2 features by * (chars_in_body),(chars_in_subject).

```
Correlations:
 images                                0.802346
urls                                  0.749635
chars_in_subject                      0.245729
chars_in_body                         0.311861
Year                                  0.198740
                                        ...
mail_type_multipart/mixed            -0.118085
mail_type_multipart/related          -0.196301
mail_type_text/html                   0.195670
mail_type_text/plain                 -0.125544
total_chars                          -0.087020
Name: label, Length: 377, dtype: float64

Top 8 correlated columns:
 org_iiitd           -0.910899
tld_ac.in           -0.894680
images               0.802346
urls                 0.749635
tld_com              0.598497
ccs                 -0.579887
salutations         -0.445196
chars_in_body        0.311861
Name: label, dtype: float64
```

Model training

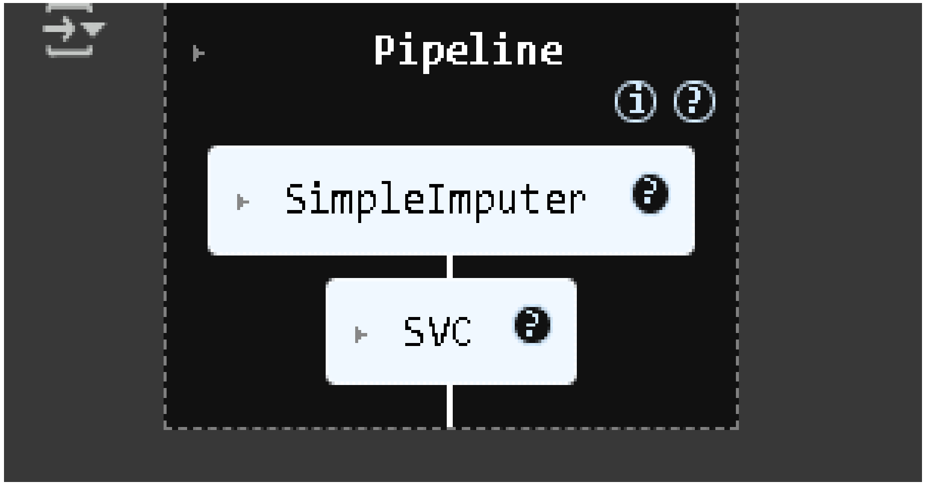
- We used SVC to train our model and if there is any nan values in the data we fill it with False to avoid errors in modeling.

- now we detected nan values in our x_train features.

```
print(X_train.isna().sum())
```

```
PC1                0
PC2                0
PC3                0
org_iiitd          0
tld_ac.in          0
images          3496
urls            3496
tld_com            0
ccs             3496
salutations     3496
```

- We will use simpleImputer to fill those nan values with mean as we used in our data.



Predicting F_score

- Now from x_test we want to predict the y_test prediction .so we will compare the x_test to y_test .

- Testing f1-Score: 0.9932267678136006
- Accuracy: 0.99

- Classification report:

Classification Report:

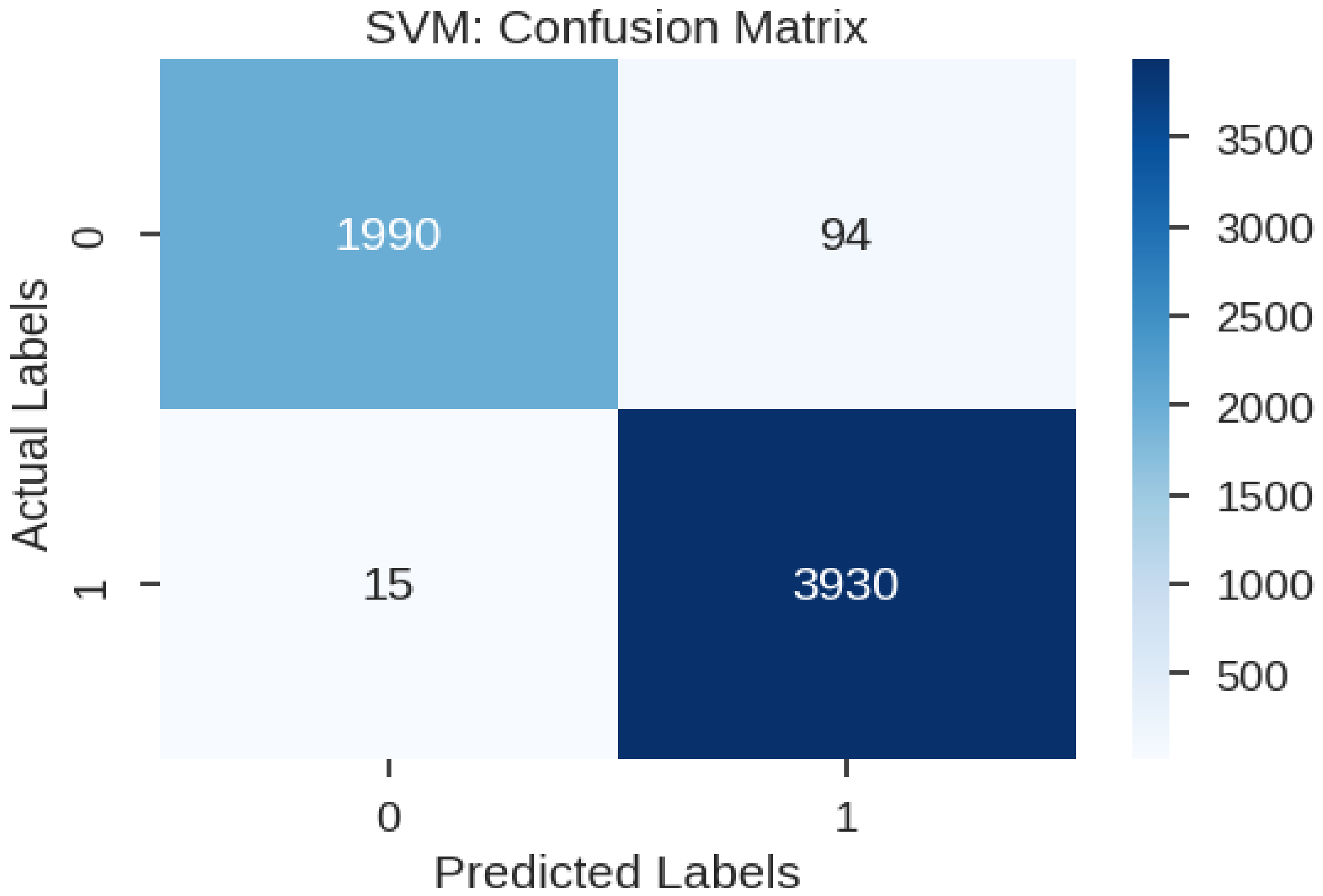
	precision	recall	f1-score	support
0.0	1.00	0.99	0.99	1841
1.0	0.99	1.00	0.99	1841
accuracy			0.99	3682
macro avg	0.99	0.99	0.99	3682
weighted avg	0.99	0.99	0.99	3682

Actual F_score leaderboard

- Now from x_test we want to predict the y_actual .so we will compare the x_test to y_actual .Testing f1-score Score: 0.9863219977412473

```
Accuracy: 98.19%
              precision    recall  f1-score   support

           0       0.99      0.95      0.97      2084
           1       0.98      1.00      0.99      3945

    accuracy                           0.98      6029
   macro avg       0.98      0.98      0.98      6029
weighted avg       0.98      0.98      0.98      6029
```

Now we will plot heatmap



Naive Bayes

- We used Naive Bayes but we got same result when we used SVC.

- Testing f1-score Score: 0.9863219977412473
- Accuracy: 98.19%
- Clustering here wont work because it should apply on numerical columns.