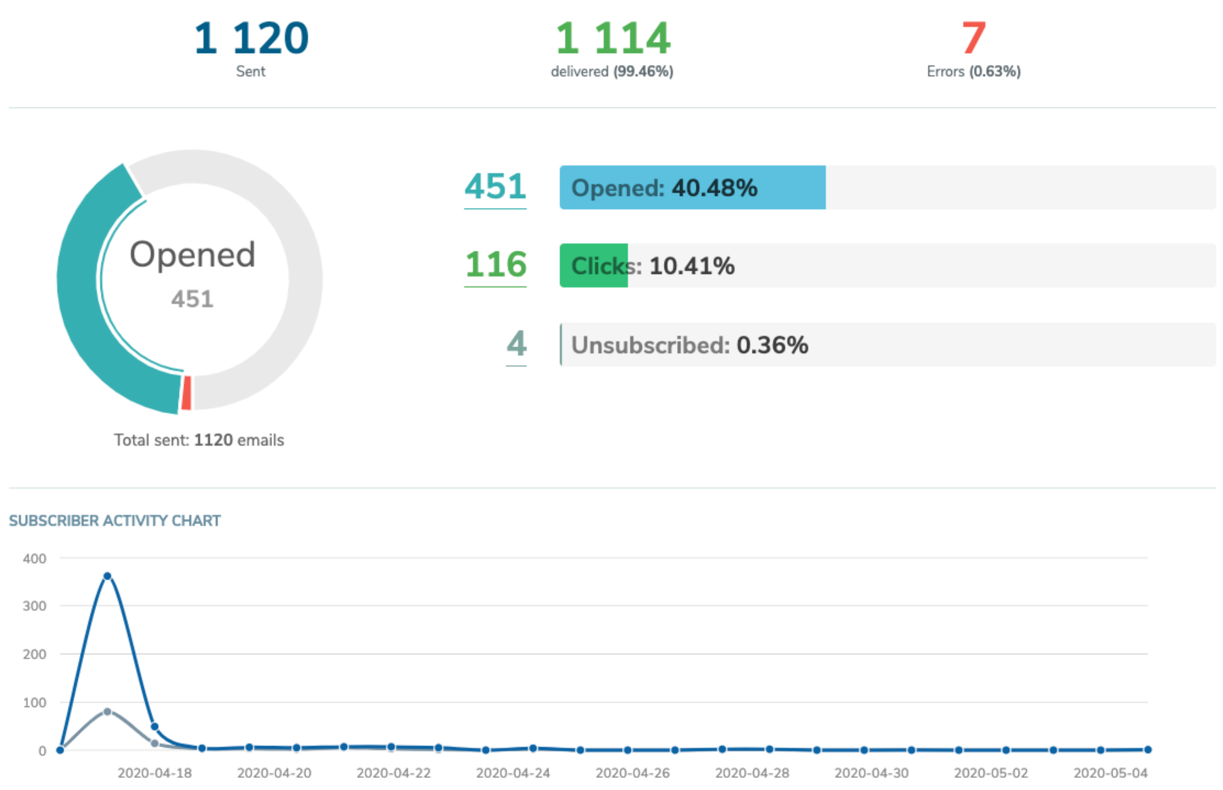
BARNALI PAUL

MAXIMIZING EMAILS CLICK TRHOUGH RATE HACKATHON

AUGUST 07, 2022



1 120
Sent

1 114
delivered (99.46%)

7
Errors (0.63%)

Opened
451

Total sent: 1120 emails

451 Opened: 40.48%

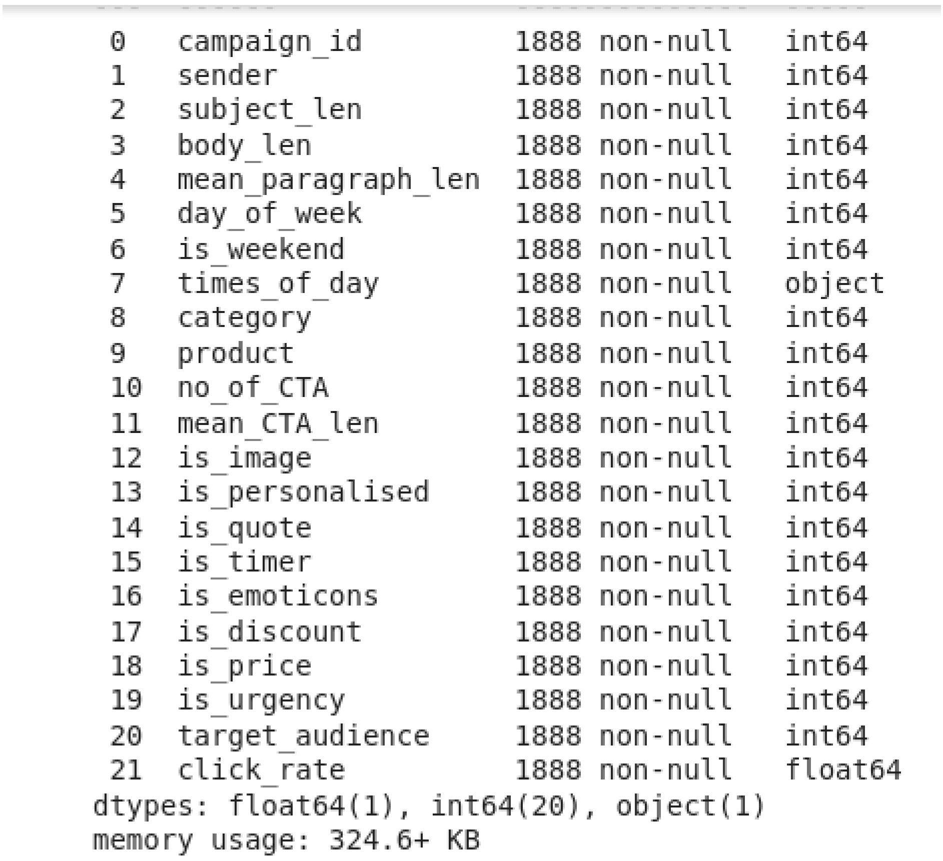
116 Clicks: 10.41%

4 Unsubscribed: 0.36%

SUBSCRIBER ACTIVITY CHART

Dataset info and description:

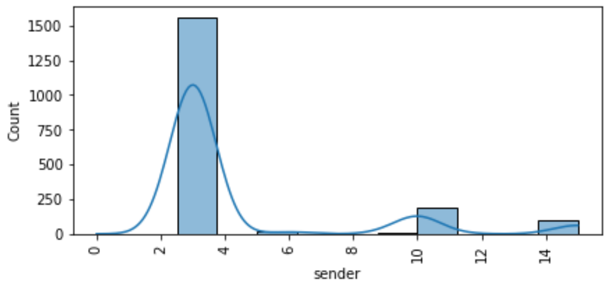
1888 rows and 22 columns:

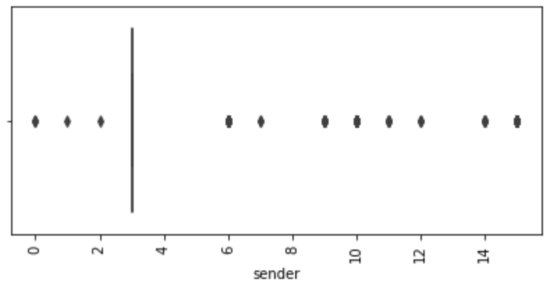


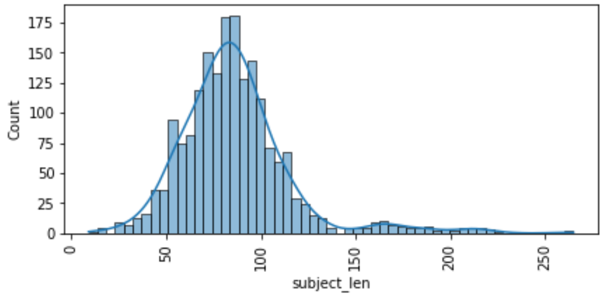
```
 0   campaign_id         1888 non-null   int64
 1   sender              1888 non-null   int64
 2   subject_len         1888 non-null   int64
 3   body_len            1888 non-null   int64
 4   mean_paragraph_len  1888 non-null   int64
 5   day_of_week         1888 non-null   int64
 6   is_weekend          1888 non-null   int64
 7   times_of_day        1888 non-null   object
 8   category            1888 non-null   int64
 9   product             1888 non-null   int64
 10  no_of_CTA           1888 non-null   int64
 11  mean_CTA_len        1888 non-null   int64
 12  is_image            1888 non-null   int64
 13  is_personalised     1888 non-null   int64
 14  is_quote            1888 non-null   int64
 15  is_timer            1888 non-null   int64
 16  is_emoticons        1888 non-null   int64
 17  is_discount         1888 non-null   int64
 18  is_price            1888 non-null   int64
 19  is_urgency          1888 non-null   int64
 20  target_audience     1888 non-null   int64
 21  click_rate          1888 non-null   float64
dtypes: float64(1), int64(20), object(1)
memory usage: 324.6+ KB
```

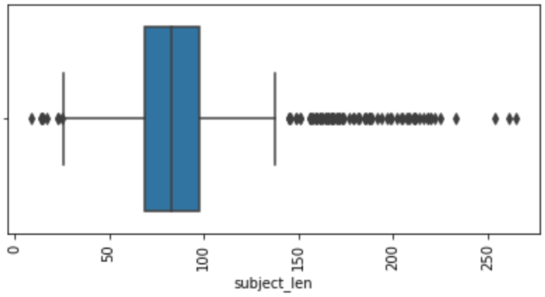
- No missing values
- Campaign_id holds no importance in the modeling, so this column will be dropped from the train dataset.
- One column “times_of_day” is of object dtype and hence will be encoded (manual).
- The column “is_price” is discrete and numerical, but should be of boolean type and thus will be converted to boolean type.

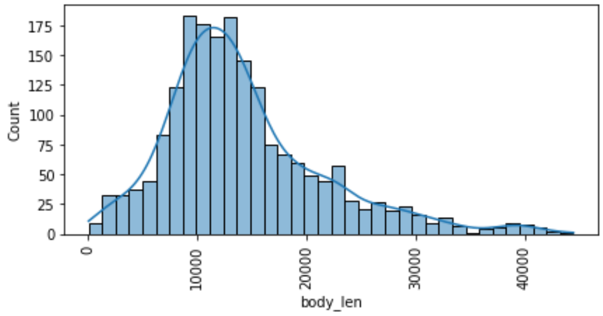
UNIVARIATE, BIVARIATE AND MULTIVARIATE ANALYSIS:

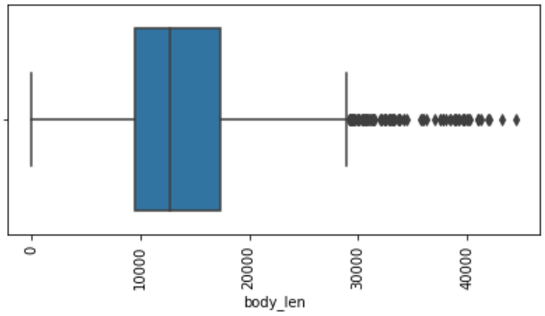


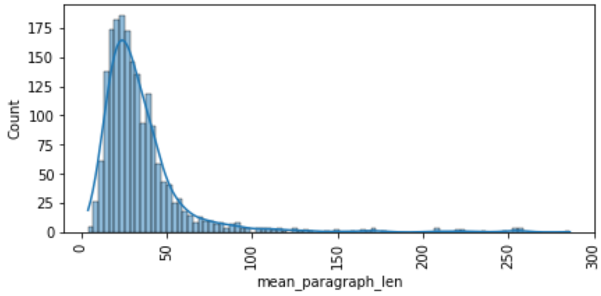


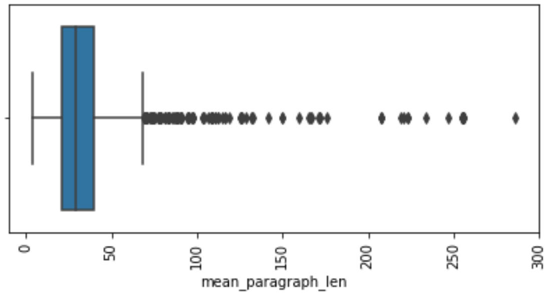


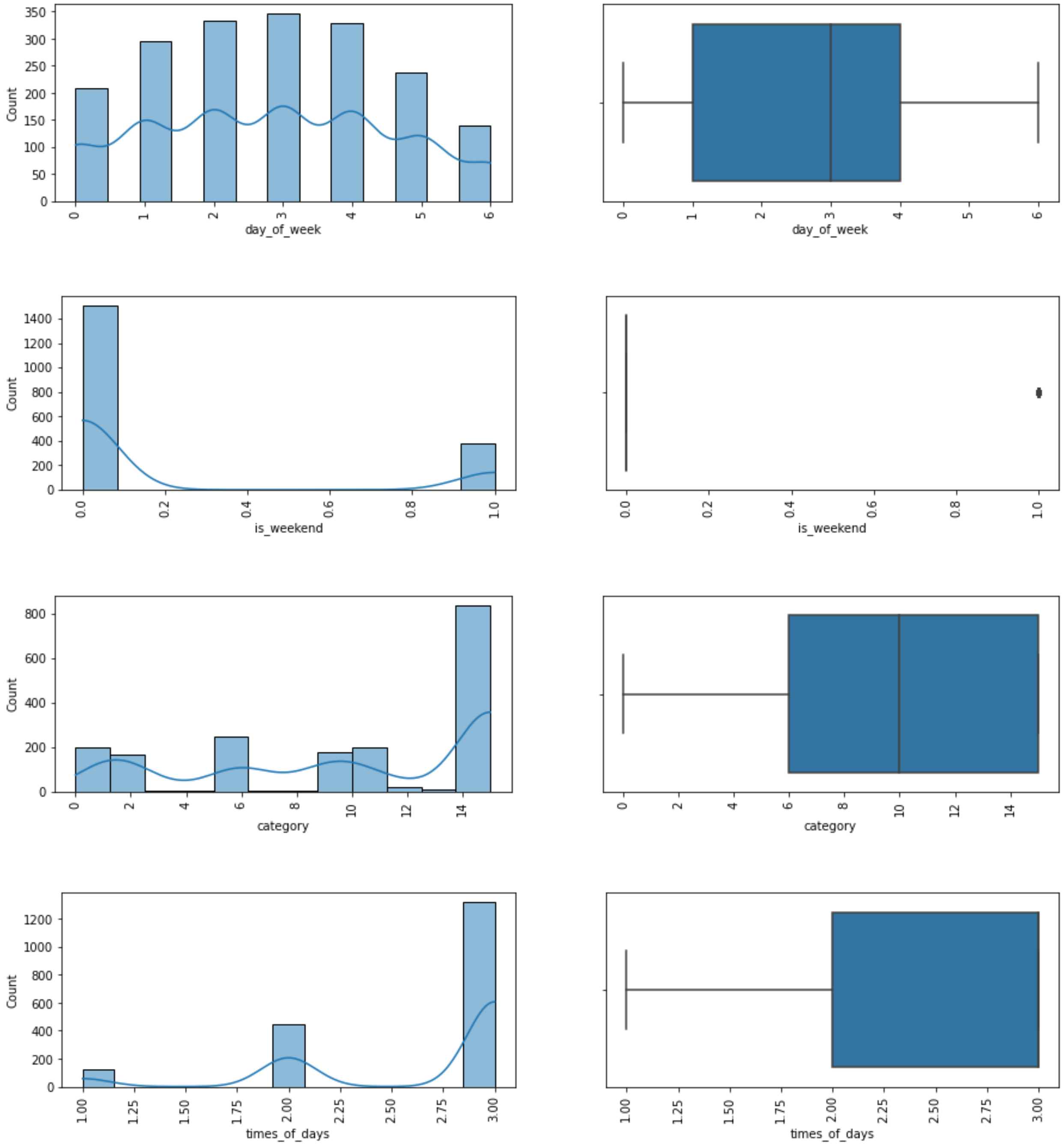


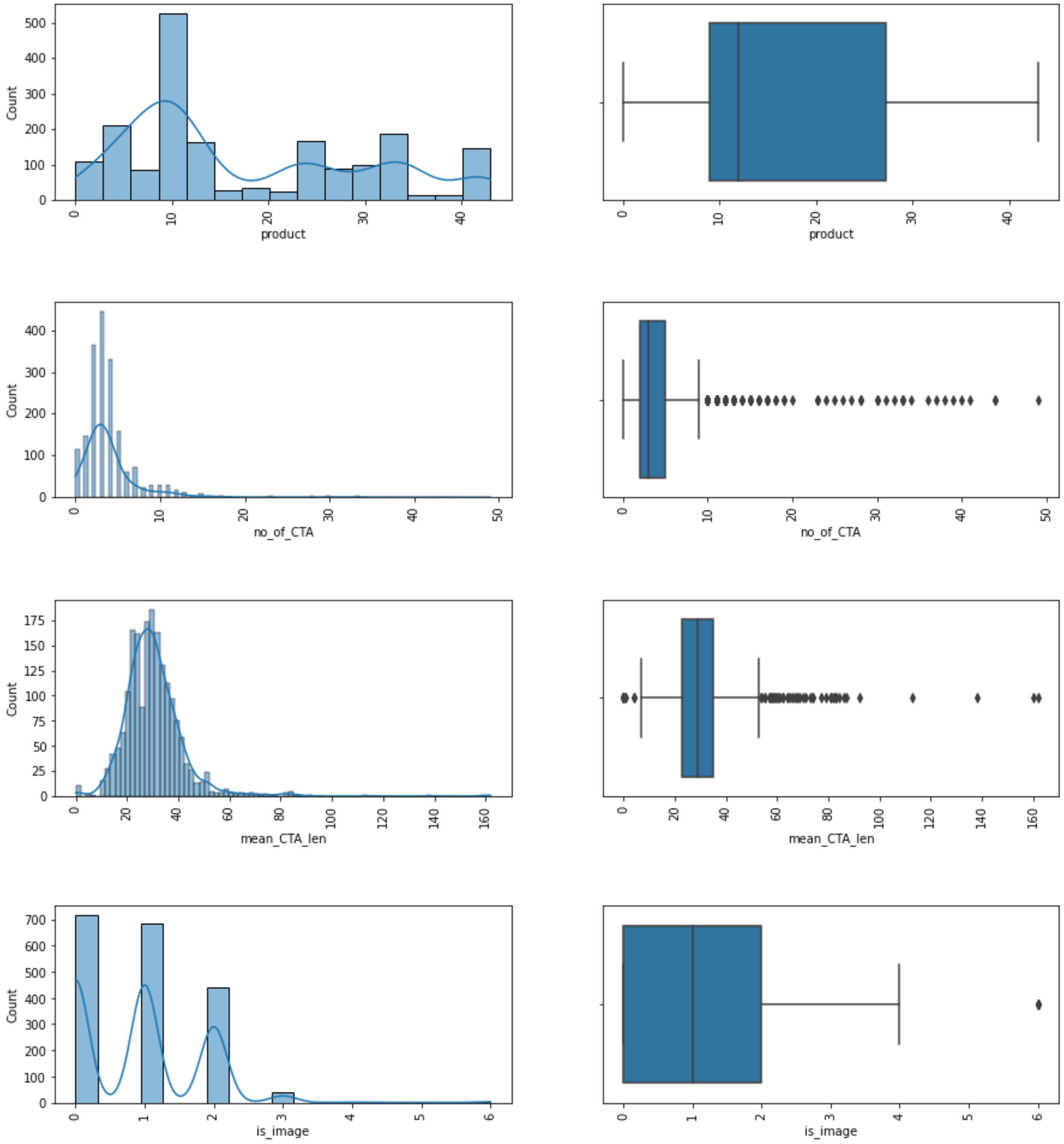




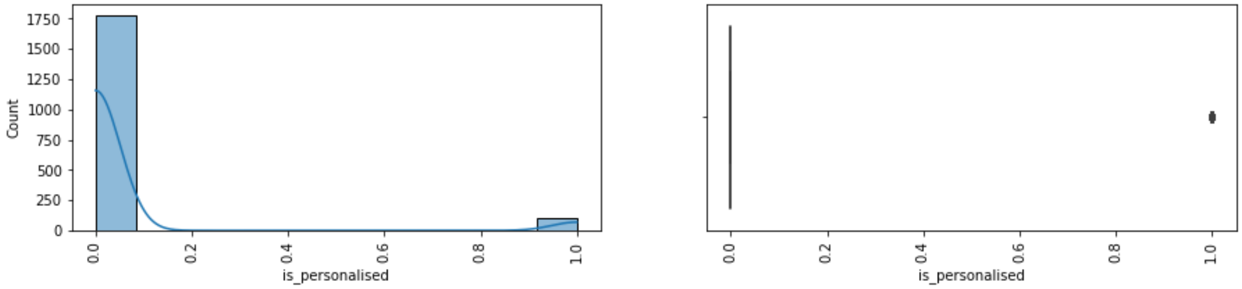


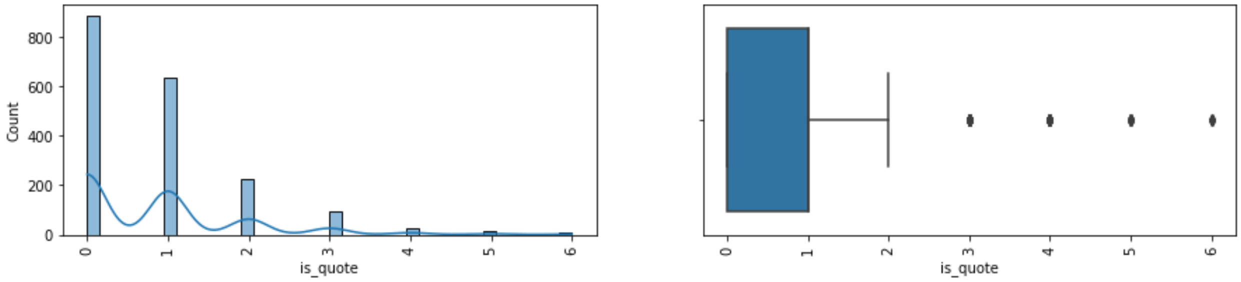


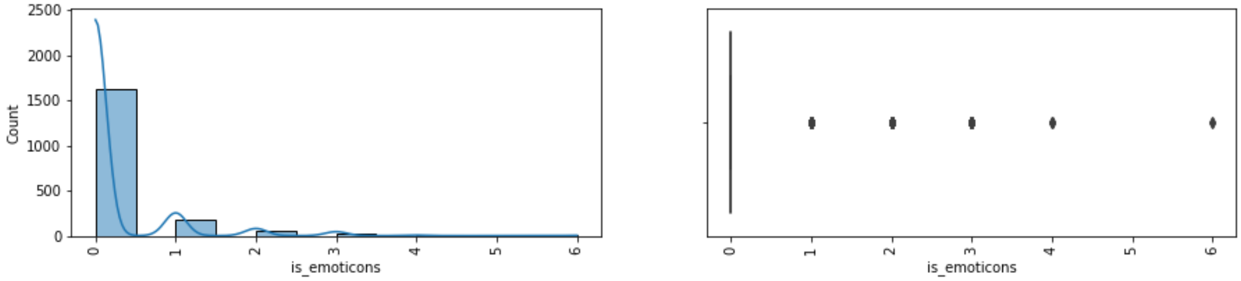

Count
0
50
100
150
200
250
300
350
0
1
2
3
4
5
6
day_of_week
0
1
2
3
4
5
6
day_of_week
Count
0
200
400
600
800
1000
1200
1400
0.0
0.2
0.4
0.6
0.8
1.0
is_weekend
0.0
0.2
0.4
0.6
0.8
1.0
is_weekend
Count
0
200
400
600
800
0
2
4
6
8
10
12
14
category
0
2
4
6
8
10
12
14
category
Count
0
200
400
600
800
1000
1200
1.00
1.25
1.50
1.75
2.00
2.25
2.50
2.75
3.00
times_of_days
1.00
1.25
1.50
1.75
2.00
2.25
2.50
2.75
3.00
times_of_days

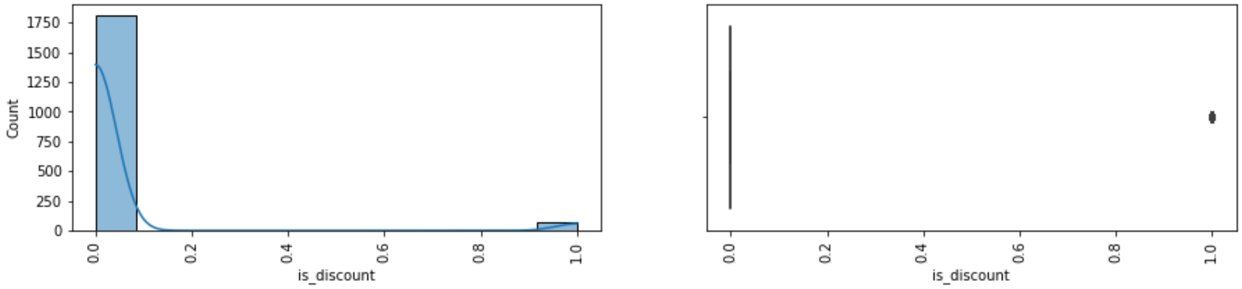


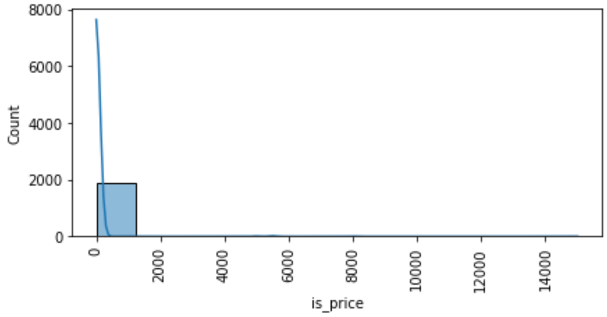
Count
0
100
200
300
400
500
0
10
20
30
40
product
0
10
20
30
40
product
Count
0
100
200
300
400
0
10
20
30
40
50
no_of_CTA
0
10
20
30
40
50
no_of_CTA
Count
0
25
50
75
100
125
150
175
0
20
40
60
80
100
120
140
160
mean_CTA_len
0
20
40
60
80
100
120
140
160
mean_CTA_len
Count
0
100
200
300
400
500
600
700
0
1
2
3
4
5
6
is_image
0
1
2
3
4
5
6
is_image

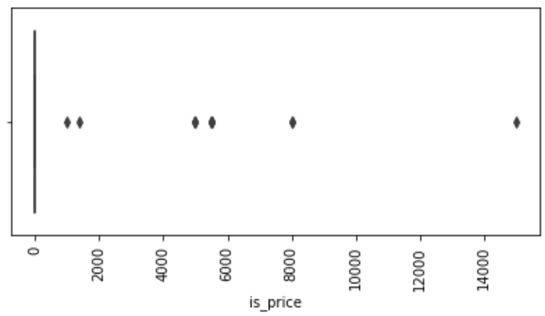

Count
1750
1500
1250
1000
750
500
250
0
0.0
0.2
0.4
0.6
0.8
1.0
is_personalised
0.0
0.2
0.4
0.6
0.8
1.0
is_personalised

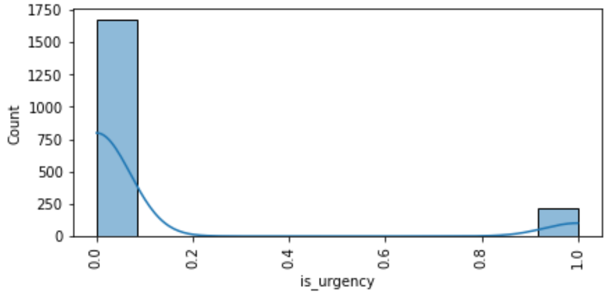

Count
800
600
400
200
0
0
1
2
3
4
5
6
is_quote
0
1
2
3
4
5
6
is_quote

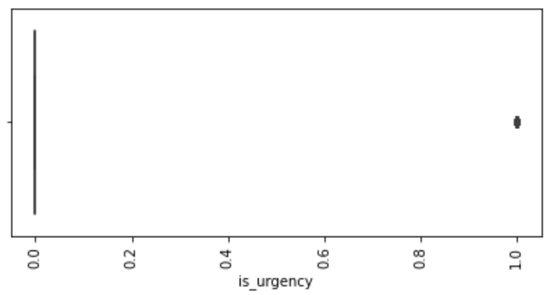

Count
2500
2000
1500
1000
500
0
0
1
2
3
4
5
6
is_emoticons
0
1
2
3
4
5
6
is_emoticons

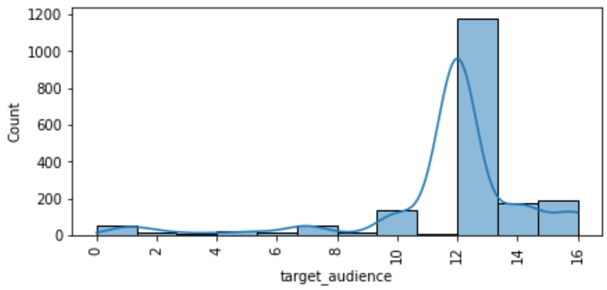

Count
1750
1500
1250
1000
750
500
250
0
0.0
0.2
0.4
0.6
0.8
1.0
is_discount
0.0
0.2
0.4
0.6
0.8
1.0
is_discount

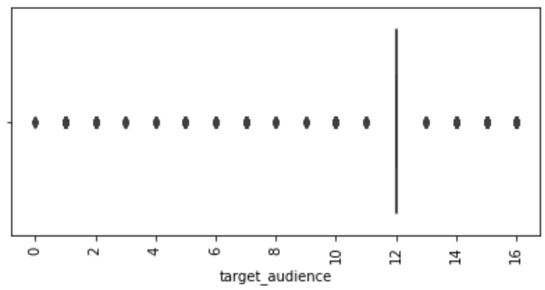

Count
8000
6000
4000
2000
0
0
2000
4000
6000
8000
10000
12000
14000
is_price

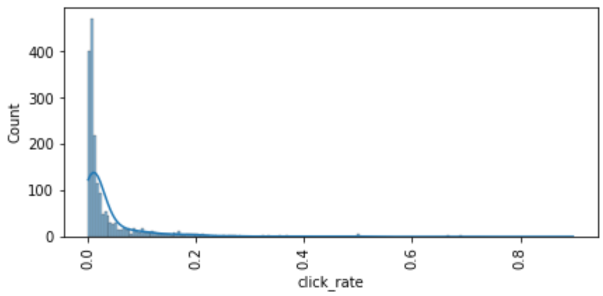

0
2000
4000
6000
8000
10000
12000
14000
is_price

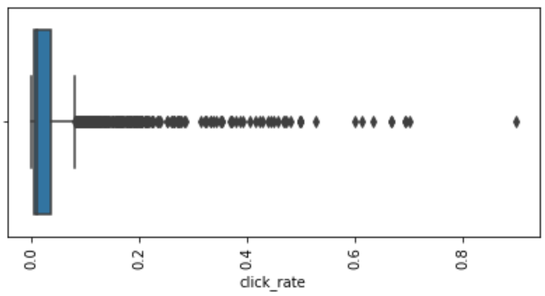

Count
1750
1500
1250
1000
750
500
250
0
0.0
0.2
0.4
0.6
0.8
1.0
is_urgency


0.0
0.2
0.4
0.6
0.8
1.0
is_urgency

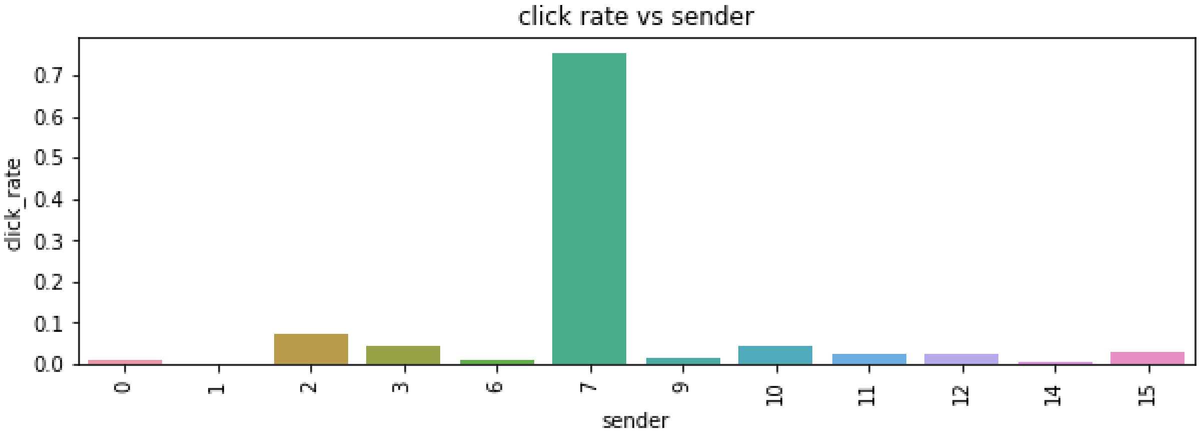

Count
1200
1000
800
600
400
200
0
0
2
4
6
8
10
12
14
16
target_audience

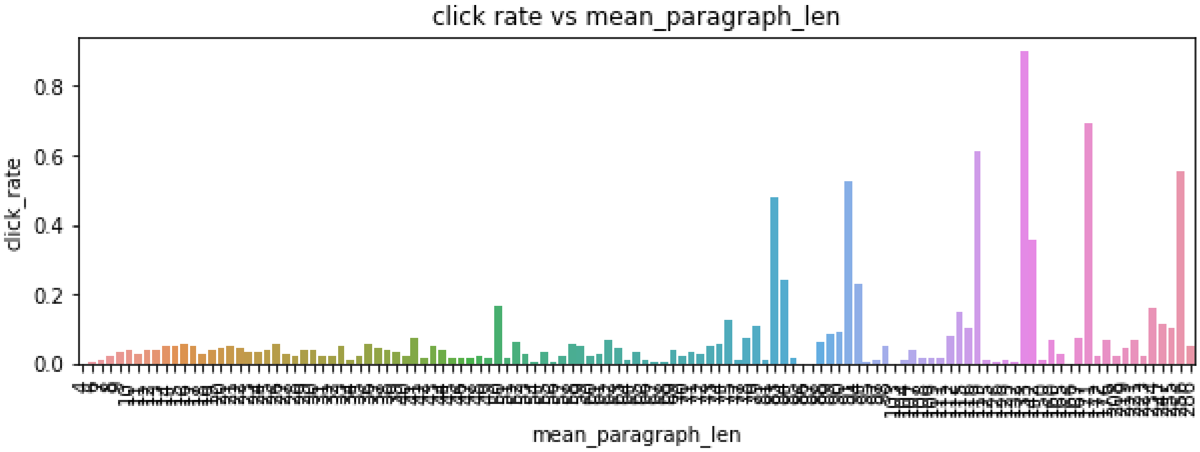

0
2
4
6
8
10
12
14
16
target_audience

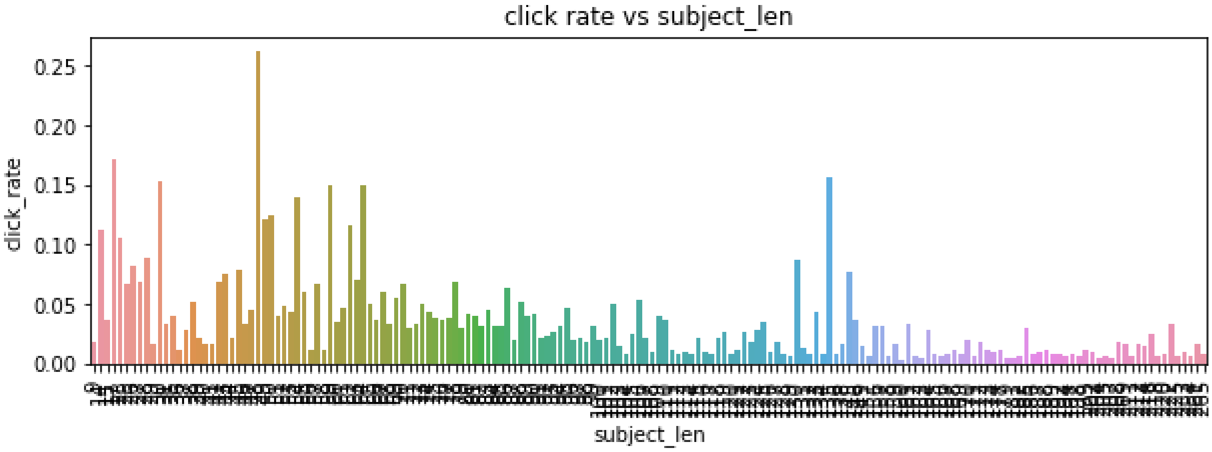

Count
400
300
200
100
0
0.0
0.2
0.4
0.6
0.8
click_rate

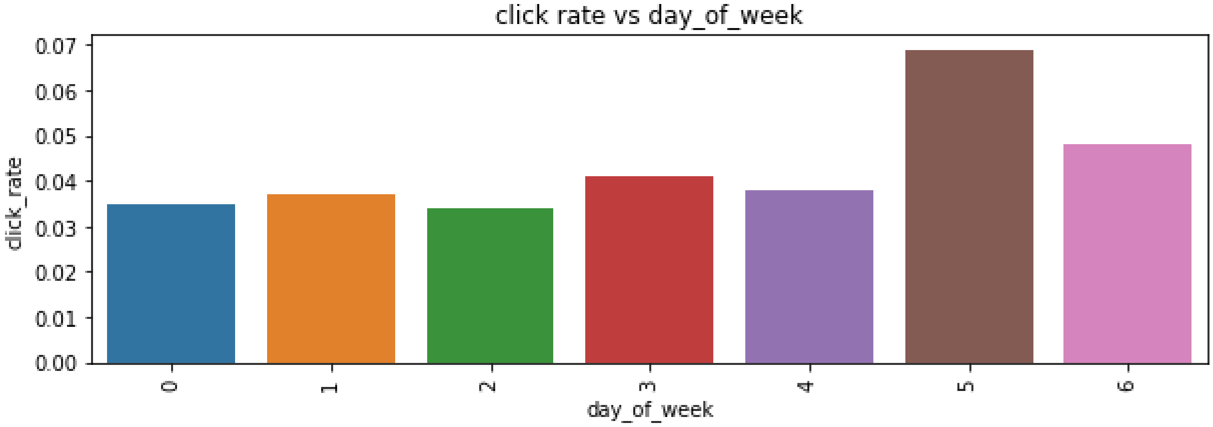

0.0
0.2
0.4
0.6
0.8
click_rate

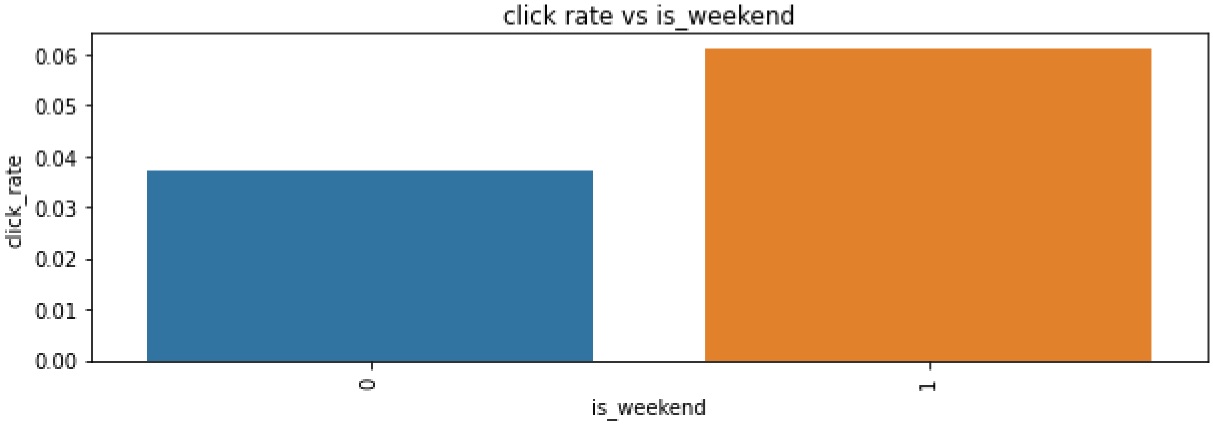
- Sender “3” is the most frequent sender among others.
- “Subject_len” follows a normal distribution with around 80-90 number of characters in a subject being the most frequent.
- “Body_len” has slightly a right-skewed distribution with 75 percentile of the distribution under 20000 characters.
- “Mean_paragraph_len” is highly right-skewed with 75 percentile of the distribution under 50 number of characters in a paragraph on an average.
- Fourth day of the week receives the highest number of emails.
- Emails is mostly sent on week days.
- Category “14” is the most frequent category emails are related to.
- Emails are mostly sent in the evening.
- 50 to 75 percentile of the distribution are mostly the type 10, 20 and 30 of products .
- Most of the emails have “five” Number of call to actions in an email.
- Average number of characters in a CTA is slightly right skewed with 75 percentile of the distribution under 40 characters.
- MOst of the emails have 0 to 2 images in them.
- Most of the emails are not personalized.
- Most of the emails do not have emoticons.
- Most of the emails do not have any price attached to them.
- Most of the emails do not have urgency in them.

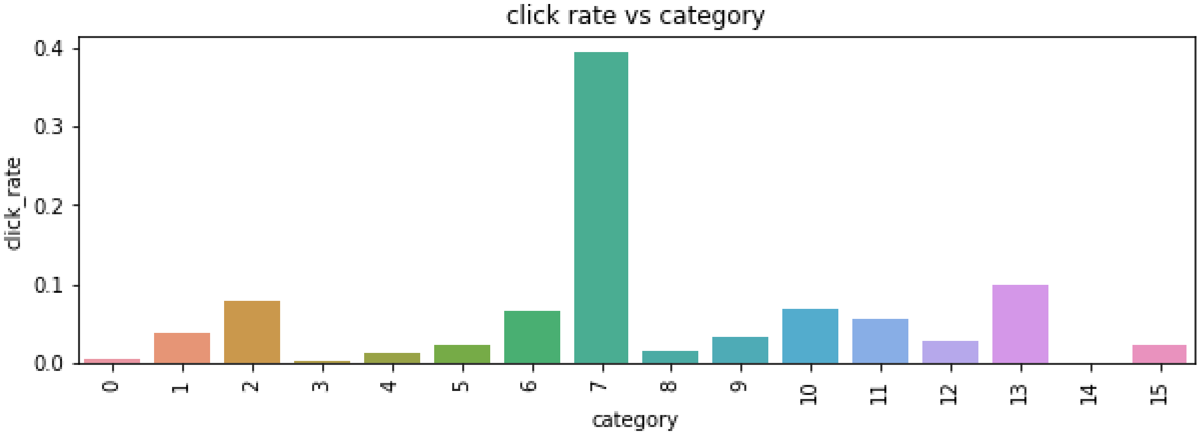

click rate vs sender
click_rate
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0.0
0
1
2
3
6
7
9
10
11
12
14
15
sender

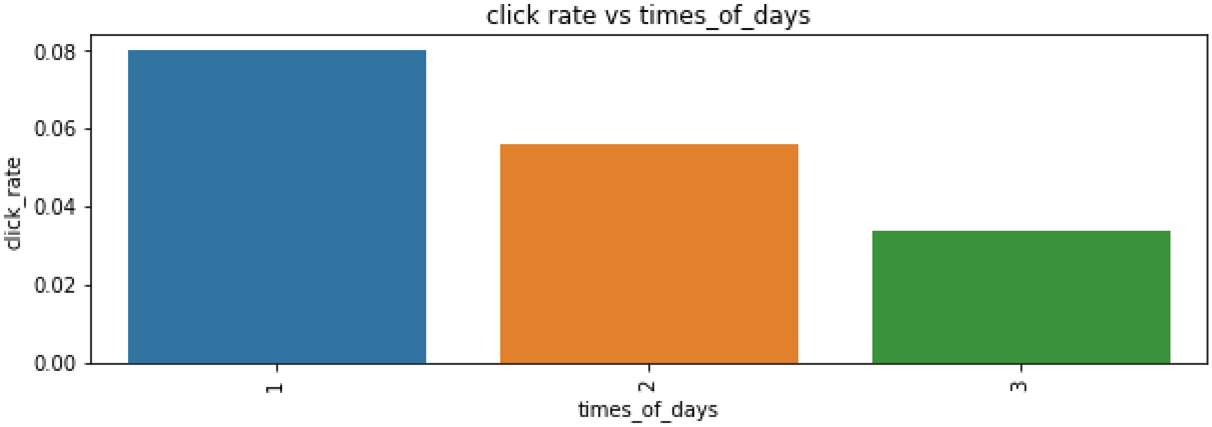

click rate vs mean_paragraph_len
click_rate
0.8
0.6
0.4
0.2
0.0
mean_paragraph_len

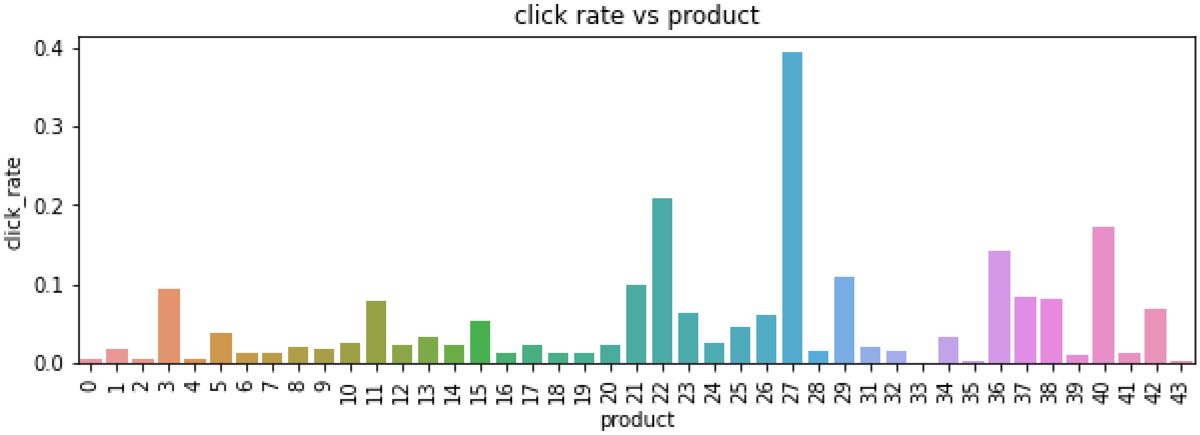

click rate vs subject_len
click_rate
0.25
0.20
0.15
0.10
0.05
0.00
subject_len

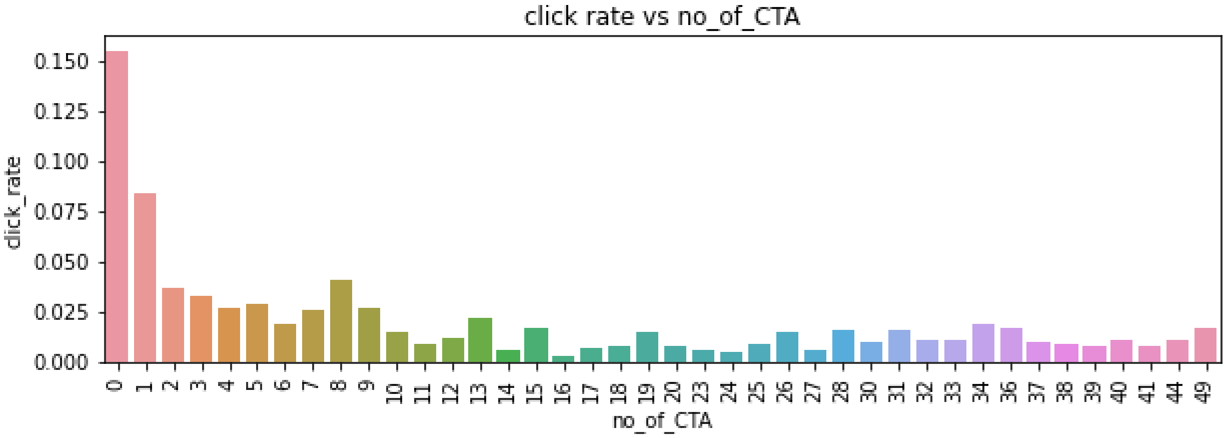

click rate vs day_of_week
click_rate
0.07
0.06
0.05
0.04
0.03
0.02
0.01
0.00
0
1
2
3
4
5
6
day_of_week

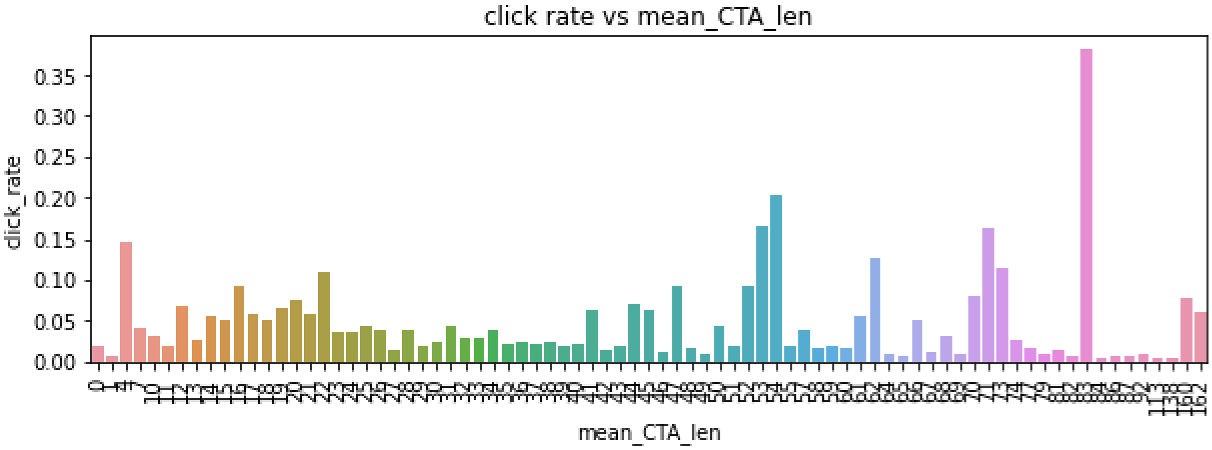

click rate vs is_weekend
click_rate
0.06
0.05
0.04
0.03
0.02
0.01
0.00
0
1
is_weekend

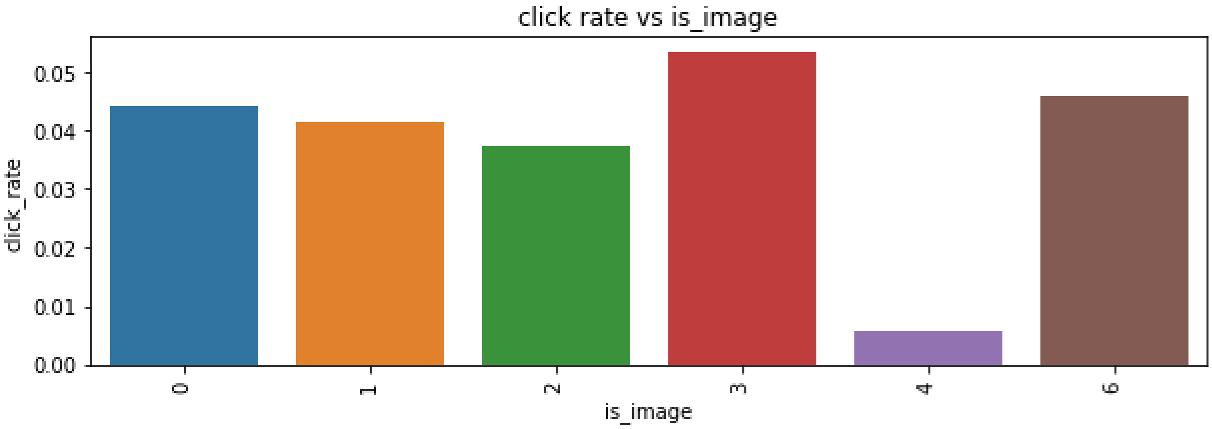

click rate vs category

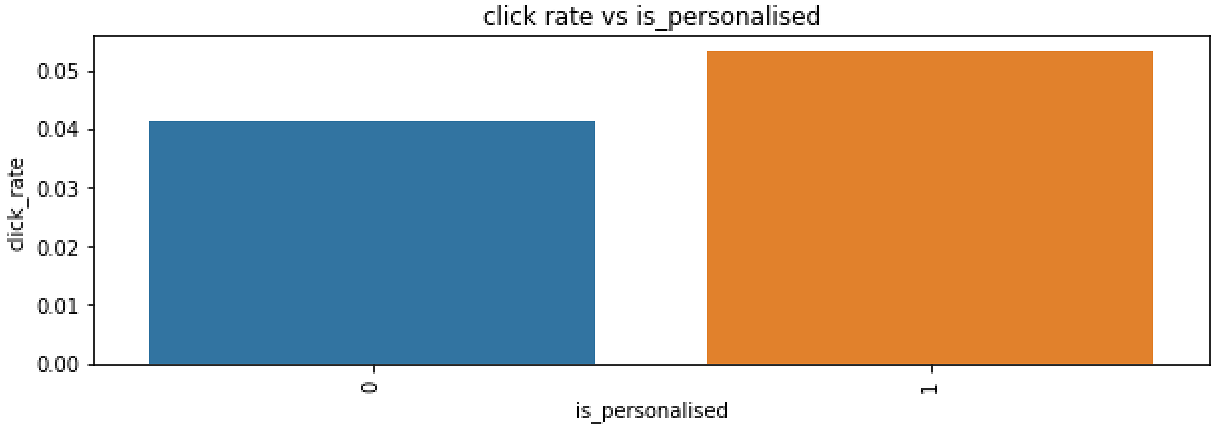

click rate vs times_of_days

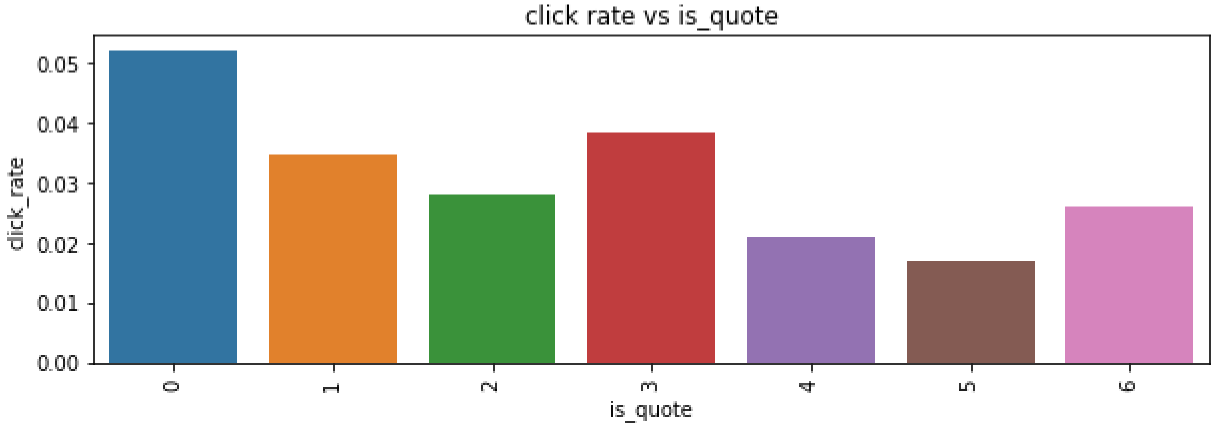

click rate vs product

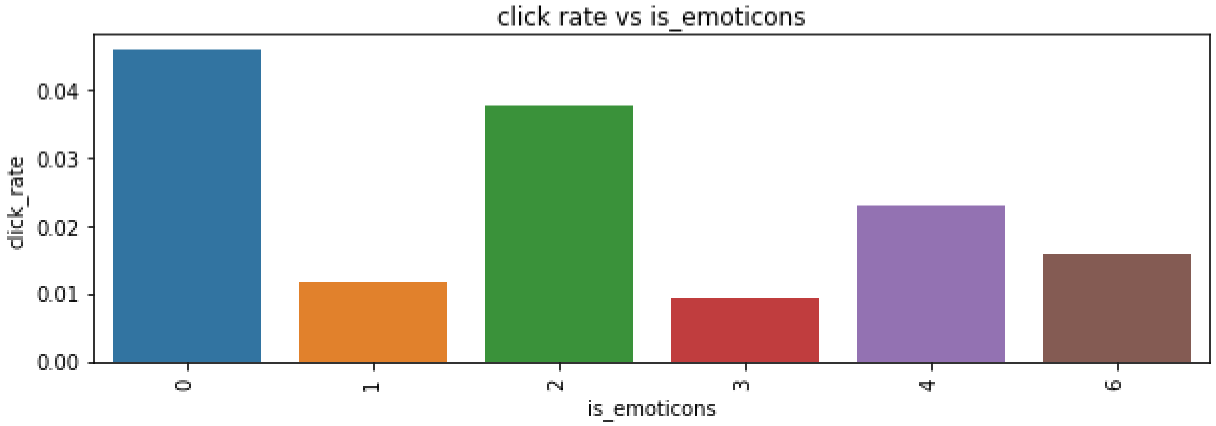

click rate vs no_of_CTA
click_rate
0.150
0.125
0.100
0.075
0.050
0.025
0.000
0
1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
23
24
25
26
27
28
30
31
32
33
34
36
37
38
39
40
41
44
49
no_of_CTA

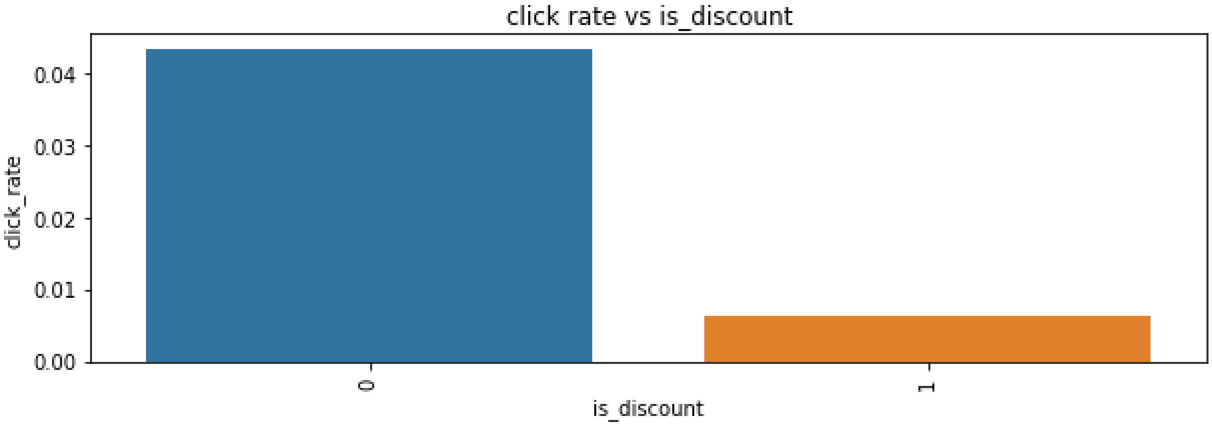

click rate vs mean_CTA_len
click_rate
0.35
0.30
0.25
0.20
0.15
0.10
0.05
0.00
0
1
4
7
113
138
160
162
mean_CTA_len

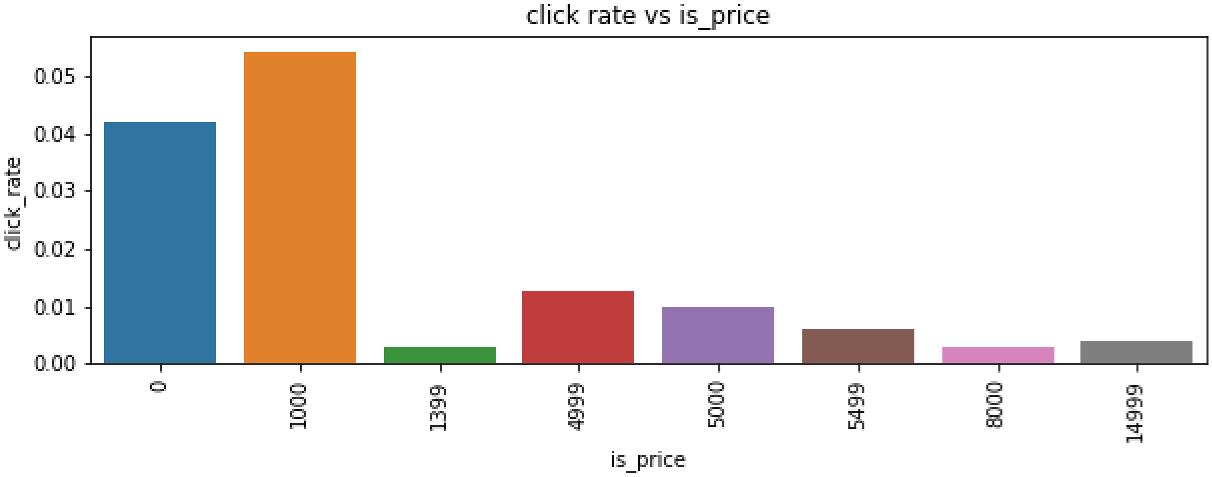

click rate vs is_image
click_rate
0.05
0.04
0.03
0.02
0.01
0.00
0
1
2
3
4
6
is_image

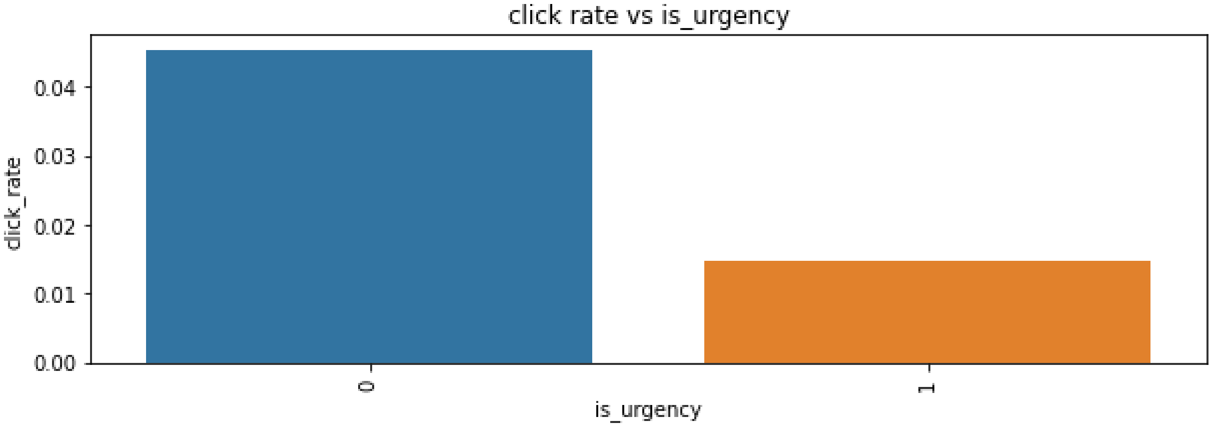

click rate vs is_personalised
click_rate
0.05
0.04
0.03
0.02
0.01
0.00
0
1
is_personalised

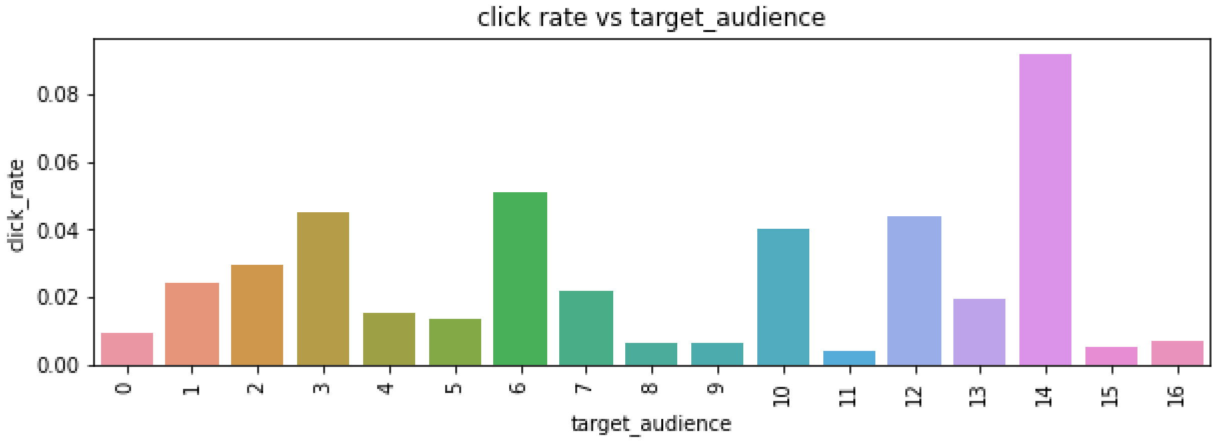

click rate vs is_quote
click_rate
0.05
0.04
0.03
0.02
0.01
0.00
0
1
2
3
4
5
6
is_quote


click rate vs is_emoticons
click_rate
0.04
0.03
0.02
0.01
0.00
0
1
2
3
4
6
is_emoticons

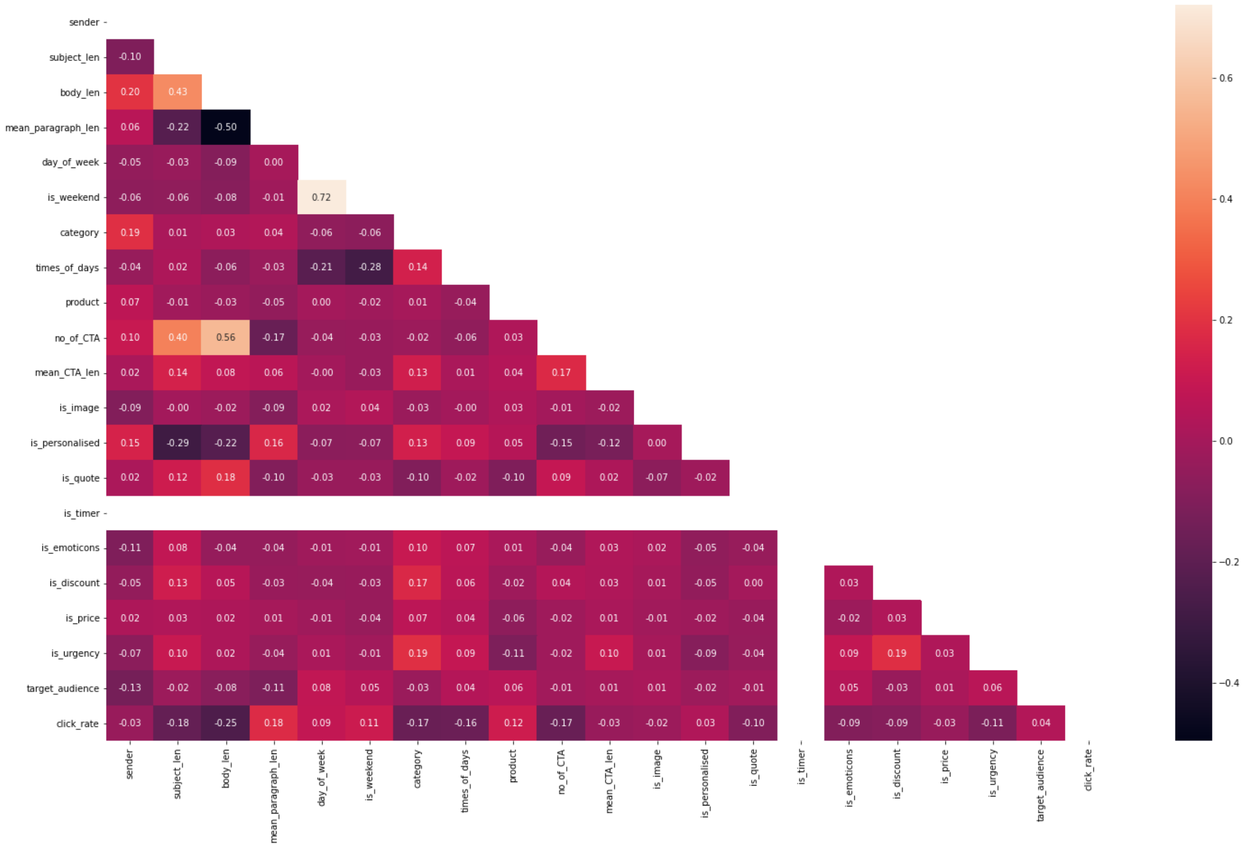

click rate vs is_discount
click_rate
0.04
0.03
0.02
0.01
0.00
0
1
is_discount


click rate vs is_price
click_rate
0.05
0.04
0.03
0.02
0.01
0.00
0
1000
1399
4999
5000
5499
8000
14999
is_price


click rate vs is_urgency
click_rate
0.04
0.03
0.02
0.01
0.00
0
1
is_urgency



- Sender “7” has the highest number of click-rate.
- Subjects having around 40 to 70 characters have the most click-rate.
- Paragraphs having characters in hundreds have higher probability of click-rate.
- Saturdays have the highest number of click-rate
- Emails sent on weekend have the higher probability of click-rate than the working days.
- Emails sent related category “7” have highest click-rate.
- Emails sent in the morning has the highest click-rate.
- Emails sent regarding product “27” has the highest clisk-rate.
- Emails having no CTAs have the highest click-rate which is “surprising”. Emails having 1 to 2 CTAs comes next in the probability of click-rate.
- CTAs containing characters around 80 have higher click-rate.
- Emails containing images attract people and thus have higher click-rate.
- Personalized emails attract more people and thus higher click-rate.
- Emails containig 0 quotes have the highest click-rate.
- Emails not relate to any “urgency” have highest click-rate.
- Emails having no or low price attached to them results in higher click-rate.



There is no significant Multicollinearity observed in the above heatmap.

MODELING BULIDING AND APPROACH:

For this dataset, I employed Ensemble models along with GridSearchCV for hyperparameters tuning.

Ensemble methods is a machine learning technique that combines several base models in order to produce one optimal predictive model. It is a solution to overcome the challenges that single base estimators face:

- High variance: One model can be very sensitive to the provided inputs to the learned features.
- Low accuracy: One model or one algorithm to fit the entire training data might not be good enough to meet expectations.
- Features noise and bias: The model relies heavily on one or a few features while making a prediction.

Ensemble Techniques

1. BAGGING: Bagging is based on a bootstrapping sampling technique. Bootstrapping creates multiple sets of the original training data with replacement. Replacement enables the duplication of sample instances in a set. Each subset has the same equal size and can be used to train models in parallel.
2. **Random Forest** Models. Random Forest Models can be thought of as **BAGG**ing, with a slight tweak. When deciding where to split and how to make decisions, BAGGed Decision Trees have the full disposal of features to choose from. Random Forest models decide where to split based on a random selection of features. Rather than splitting at similar features at each node throughout, Random Forest models implement a level of differentiation because each tree will split based on different features. This level of differentiation provides a greater ensemble to aggregate over producing a more accurate predictor.
3. **Gradient Boosting:** Gradient boosting algorithms are great techniques that have high predictive performance. Xgboost [2], LightGBM [3], and CatBoost are popular boosting algorithms that can be used for regression and classification problems.

TRAIN-VALIDATION DATASET SPLIT:

Taking out "click_rate" column from the train dataset and setting it as target variable.

Further splitting the trainX and trainY dataset to train and validation dataset into 90:10 ratio.

thus , X_train shape is: (1699, 20)

And, X_val shape is: (189, 20)

MODEL BULIDING:

1) RANDOM FOREST WITH GRIDSEARCHCV:

During the hyper parameters tuning with GridSearchCV, best_params_ comes out to be:

{'bootstrap': True,

'max_depth': 90,

'max_features': 0.3,

'min_samples_leaf': 3,

'min_samples_split': 8,

'n_estimators': 200}

RandomForest is buld upon theses params:

```
RFR = RandomForestRegressor(bootstrap = True, max_depth=90,
max_features=0.3, min_samples_leaf=3, min_samples_split=8,
n_estimators=200)
```

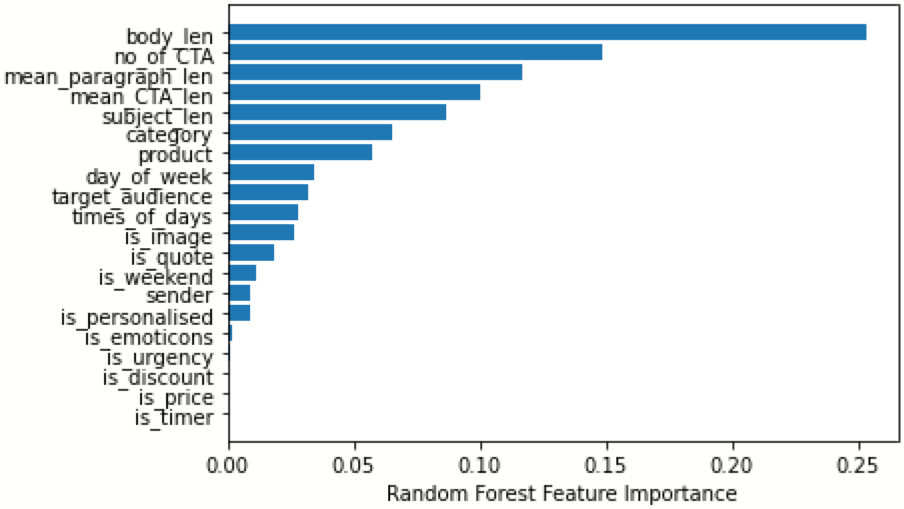
And the model is fitted on train dataset.

Performance of RFR model:

R-squared score on train dataset: 0.71

R-squared score on test dataset: 0.614

Feature importance comes out to be:



2) GradientBoosting with GridSearchCV:

During the hyper parameters tuning with GridSearchCV, best_params_ comes out to be:

{'max_depth': 80,

'max_features': 0.1,

'min_samples_leaf': 3,

'min_samples_split': 10,

'n_estimators': 200}

Gradient Boosting is build upon these params.

```
GBR = GradientBoostingRegressor(max_depth=80, max_features=0.1,
min_samples_leaf=3,min_samples_split=10, n_estimators=200)
```

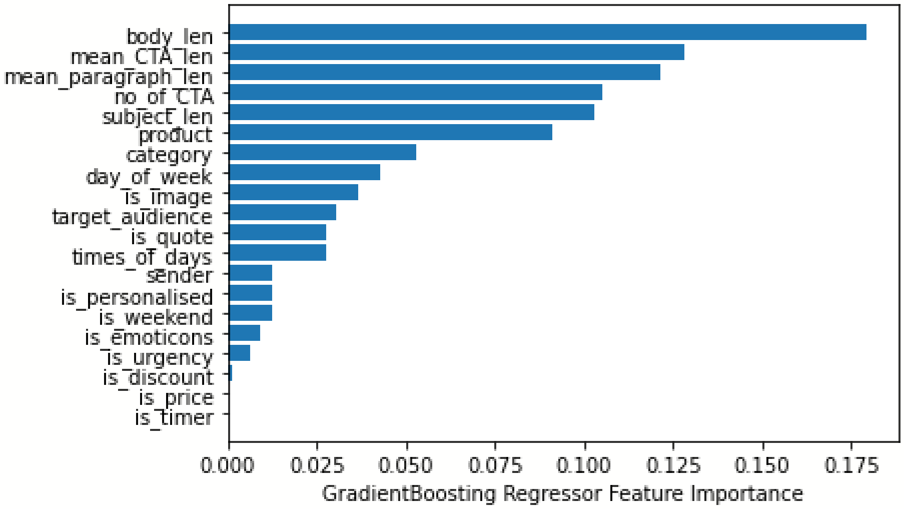
And then the model is fitted on the train dataset.

Performance of GBR model:

R-squared score on train dataset: 0.99

R-squared score on test dataset: 0.65

Feature importance comes out to be:



3) XGBoosting with GridSearch CV:

During the hyper parameters tuning with GridSearchCV, best_params_ comes out to be:

{'colsample_bytree': 0.7,

'learning_rate': 0.07,

'max_depth': 5,

'min_child_weight': 4,

'n_estimators': 500,

'nthread': 4,

'objective': 'reg:linear',

'silent': 1,

'subsample': 0.7}

XGBoost is build upon these params.

```
XGBR = XGBRegressor(colsample_bytree=0.7, learning_rate=0.07, max_depth=5,
            min_child_weight=4, n_estimators=500, nthread=4, silent=1,
            subsample=0.7)
```

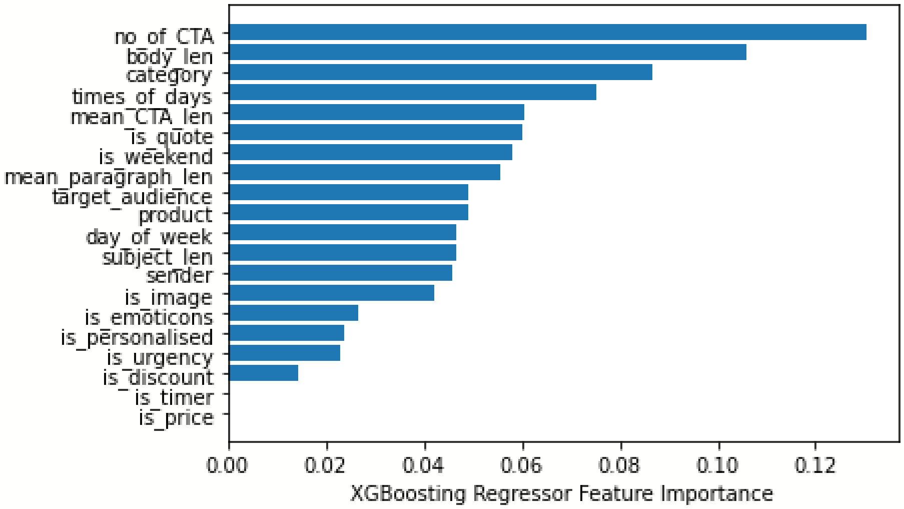
And then the model is fitted on the train dataset.

Performance of XGBR model:

R-squared score on train dataset: 0.96

R-squared score on test dataset: 0.67

Feature importance comes out to be:



Among the three models, XGBOOST with GRIDSEARCHCV is performing the best with 67% of R-squared score on the validation dataset.

Thus this particular model is used to predict on the test dataset as well.

BUSINESS INSIGHTS:

- **Number of CTAs in an email, length of the body, category of the emails, time of the day emails are sent, average number of characters in the CTAs, number of quotes in the emails, emails sent on weekends or not, number of characters present in paragraphs in the emails, who are the target audience, Products emails are related to, which day of the week emails are sent, number of characters present in the subject, senders of the emails and number of images present in the emails** are the major features that are influencing the Click-Rate.
- Looking at the BiVariate Analysis, following can be extracted out that can influence the email business:
    - Sender "7" has the highest number of click-rate.
    - Subjects having around 40 to 70 characters have the most click-rate.
    - Paragraphs having characters in hundreds have higher probability of click-rate.
    - Saturdays have the highest number of click-rate
    - Emails sent on weekend have the higher probability of click-rate than the working days.
    - Emails that are sent related to category "7" have highest click-rate.
    - Emails sent in the morning has the highest click-rate.
    - Emails sent regarding product "27" has the highest clisk-rate.
    - Emails having no CTAs have the highest click-rate which is "surprising". Emails having 1 to 2 CTAs comes next in the probability of click-rate.
    - CTAs containing characters around 80 have higher click-rate.
    - Emails containing images attract people and thus have higher click-rate.
    - Personalized emails attract more people and thus higher click-rate.
    - Emails containig 0 quotes have the highest click-rate.
    - Emails not related to any "urgency" have highest click-rate.

Thank you!