Email click through rate - Jackson Yang

Table of Content:

Problem Statement

The marketing department wants to know the important factors impacting the email click-through-rate(CR) and how to improve the CR. They want to increase the percentage of *customers who click on the link inside the email*.

Variables description

- **email_id** : the Id of the email that was sent. It is unique by email
- **email_text**: two different versions of the email have been sent: one has "long text" (i.e. has 4 paragraphs)and one has "short text" Oust two paragraphs)
- **email_version**: some emails were "personalized" (i.e. they had the name of the user receiving the email in the incipit, such as "Hi John,"), while some emails were "generic" (the incipit was just "Hi,")
- **hour**: the local time on which the email was sent
- **weekday**: the weekday on which the email was sent
- **user_country**: the country where the user receiving the email is based. It comes from the user ip address when they created the account
- **user_past_purchases**: how many items in the past were bought by the user receiving the email
- **clicked** - Whether the user has clicked on the link inside the email. This is our label and, most importantly, the goal of the project is to increase this

1. Load data and libraries

In [307…

```python
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns

np.warnings.filterwarnings('ignore')
```

In [308…

```python
df = pd.read_csv('course1_data_emails.csv', index_col= "email_id")
```

2. Exploratory data analysis

2.1 General overview of the data

In [309…

```python
df.head()
```

Out[309]:

email_id	email_text	email_version	hour	weekday	user_country	user_past_purchases	clicked
8	short_email	generic	9	Thursday	US	3	0
33	long_email	personalized	6	Monday	US	0	0
46	short_email	generic	14	Tuesday	US	3	0
49	long_email	personalized	11	Thursday	US	10	0
65	short_email	generic	8	Wednesday	UK	3	0

In [310…

```python
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 99950 entries, 8 to 999998
Data columns (total 7 columns):
 #   Column               Non-Null Count  Dtype
---  ------               --------------  -----
 0   email_text           99950 non-null  object
 1   email_version        99950 non-null  object
 2   hour                 99950 non-null  int64
 3   weekday              99950 non-null  object
 4   user_country         99950 non-null  object
 5   user_past_purchases  99950 non-null  int64
 6   clicked              99950 non-null  int64
dtypes: int64(3), object(4)
memory usage: 6.1+ MB
```

Inference

- no null value
- data type are all ok

In [311…

```python
df.describe().T
```

Out[311]:

	count	mean	std	min	25%	50%	75%	max
hour	99950.0	9.059100	4.439618	1.0	6.0	9.0	12.0	24.0
user_past_purchases	99950.0	3.878559	3.196324	0.0	1.0	3.0	6.0	22.0
clicked	99950.0	0.020700	0.142380	0.0	0.0	0.0	0.0	1.0

Inference

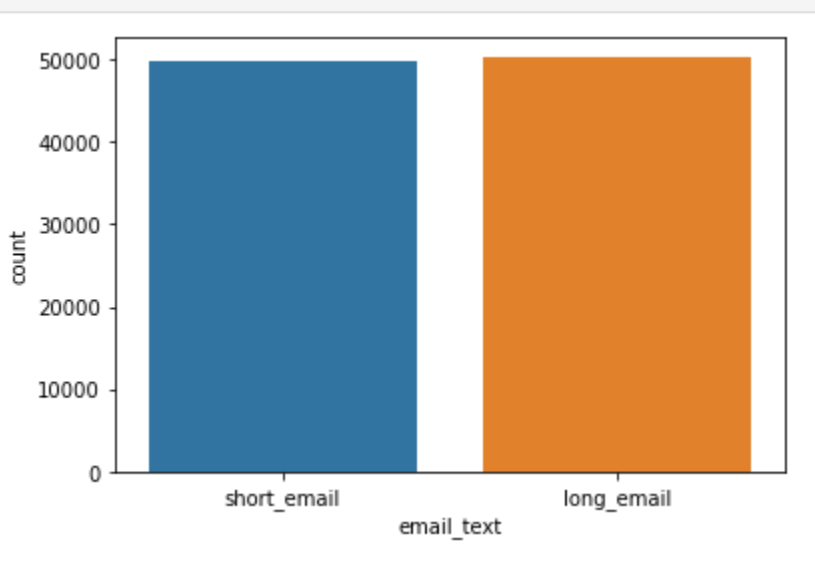
- We can see that CR is quite low with only 2.07%
- On average, the receivers of the emails consumed 3.87 items

2.2 Data Visualization

2.2.1 email_text

In [312…

```python
sns.countplot(data=df, x='email_text');
```



In [313…

```python
df['email_text'].value_counts()
```

Out[313]:

```
long_email     50248
short_email    49702
Name: email_text, dtype: int64
```

Inference

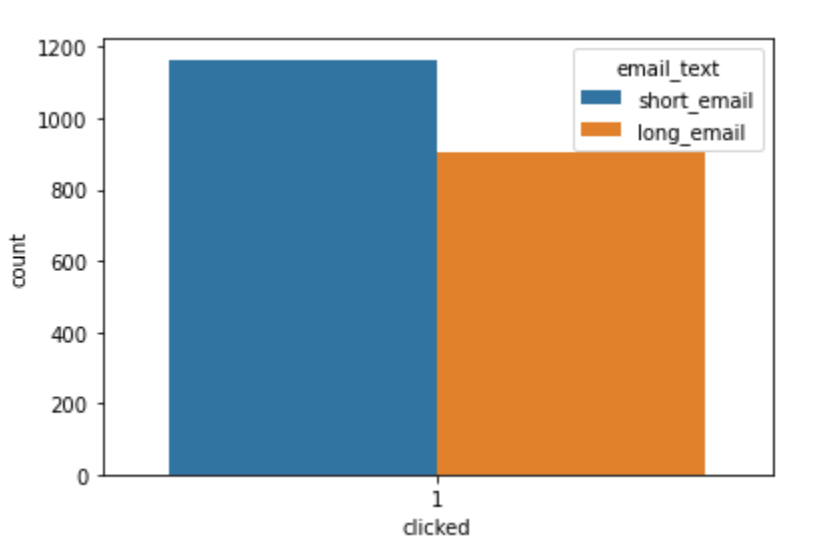
- we can see that long email is slightly more than short email
- we can probably guess that people who received the short-email might be more likely to open the email compared to the long email (let's find out if this is true)

In [314…

```python
sns.countplot(data = df[df.clicked == 1], x='clicked',hue='email_text')
```

Out[314]:

```
<AxesSubplot:xlabel='clicked', ylabel='count'>
```



Inference

- Our intuition is correct. There is around 2000 difference
- Short_email is more effective

2.2.2 email_version

In [315…

```python
df['email_version'].value_counts()
```

Out[315]:

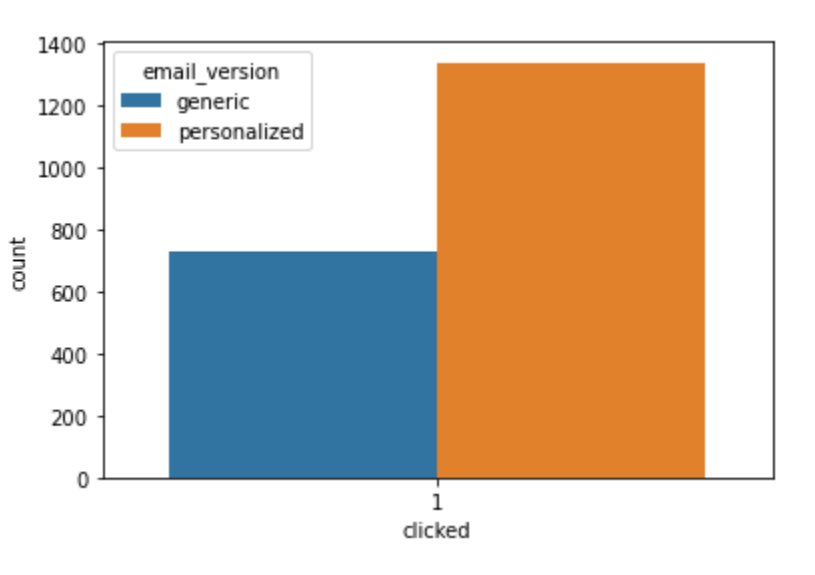
```
generic         50178
personalized    49772
Name: email_version, dtype: int64
```

In [316…

```python
sns.countplot(data = df[df.clicked == 1], x='clicked',hue='email_version')
```

Out[316]:

```
<AxesSubplot:xlabel='clicked', ylabel='count'>
```



Inference

- we can see that personalized (hi John...) is much more effective.
- This makes sense because people tend to open the emails/read the emails that have the personalized headings.

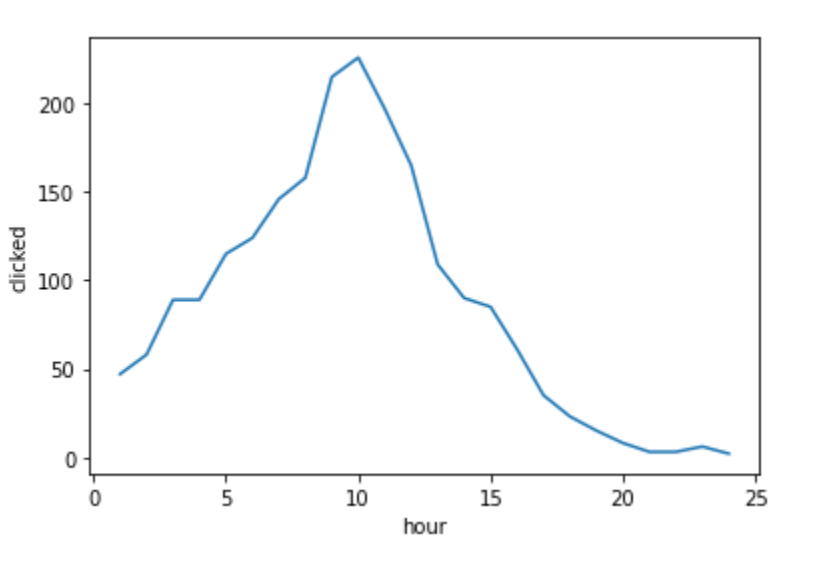
2.2.3 Hour/Weekday

In [317…

```python
hour_sort = df.groupby('hour').sum('clicked').sort_values('clicked',ascending = False)
sns.lineplot(data=hour_sort, x='hour', y='clicked');
```

Out[317]:

```
<AxesSubplot:xlabel='hour', ylabel='clicked'>
```



Inference

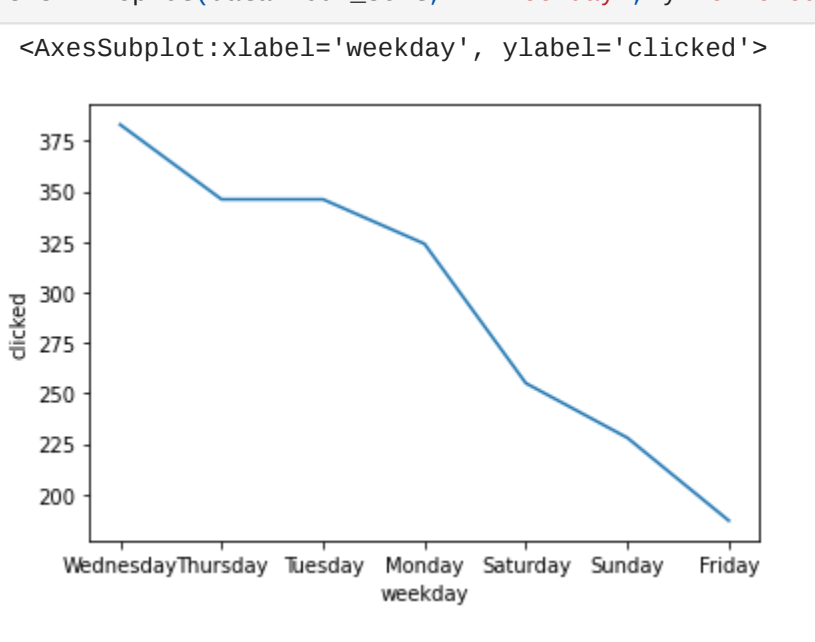
- We can observe that around 10 a.m., people are more inclined to click on the link.
- After 10 a.m., the click-through rate decreases.
- To increase the click-through rate, the company can aim to send out the majority of its emails around 10 a.m.

In [319…

```python
hour_sort = df.groupby('weekday').sum('clicked').sort_values('clicked',ascending = False)
sns.lineplot(data=hour_sort, x='weekday', y='clicked');
```

Out[319]:

```
<AxesSubplot:xlabel='weekday', ylabel='clicked'>
```



Inference

- Company should try to launch the marketing campaign on Wednesday/Thursday/Tuesday (around Wednesday)
- And we should avoid Friday because people almost never click the link in the emails on Friday

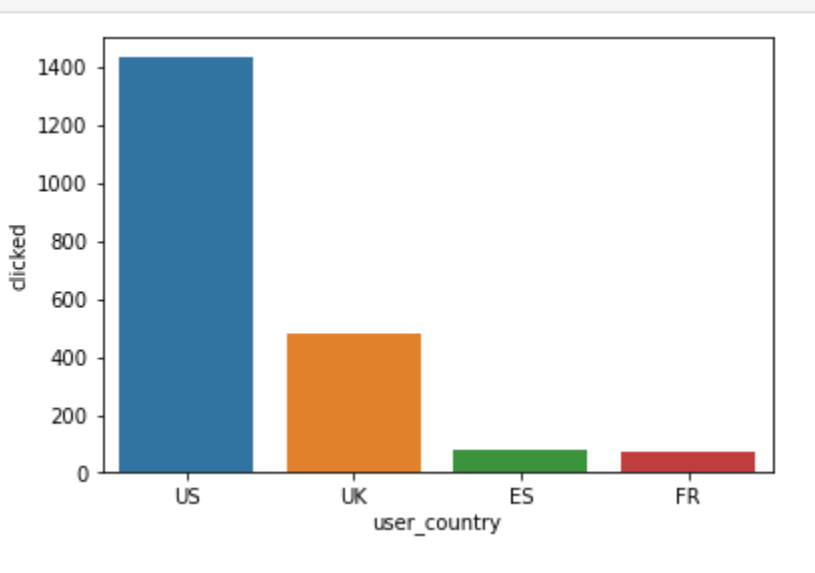
2.2.4 user countries

In [289…

```python
user_sort = df.groupby('user_country').sum('clicked').sort_values(
    'clicked', ascending=False)

user_sort = user_sort.reset_index()#reset index

sns.barplot(data=user_sort, x='user_country', y='clicked');
```



Inference

- US has the highest clicked number
- FR&ES has the lowest (for the marketing department, we should either find a way to increase the CR or abandon them)

3. Model Building

In [290…

```python
import statsmodels.api as sm
```

3.1 Converting time into categories

In [291…

```python
df.hour.value_counts().sort_values(ascending= False)

# 0 am  - 6 am : night
# 6 am - 12 pm : morning
# 12 pm - 18 pm : afternoon
# 18 pm - 24 pm : evening

df['time_category'] = ''

df.loc[df['hour'].between(0,6), 'time_category'] = 'night'
df.loc[df['hour'].between(6,12), 'time_category'] = 'morning'
df.loc[df['hour'].between(12,18), 'time_category'] = 'afternoon'
df.loc[df['hour'].between(18,24), 'time_category'] = 'evening'

df.drop('hour', axis = 1, inplace=True)
```

3.2 Getting the reference level before converting into dummies

Before converting the categorical variables into dummies, we need to know which ones are the reference levels for them

In [292…

```python
df_category = df.select_dtypes(['object']).astype('category')
```

In [293…

```python
df_category
```

Out[293]:

email_id	email_text	email_version	weekday	user_country	time_category
8	short_email	generic	Thursday	US	morning
33	long_email	personalized	Monday	US	morning
46	short_email	generic	Tuesday	US	afternoon
49	long_email	personalized	Thursday	US	morning
65	short_email	generic	Wednesday	UK	morning
...	...	...	...	...	...
999969	short_email	generic	Thursday	US	evening
999972	long_email	personalized	Tuesday	US	morning
999976	long_email	personalized	Wednesday	UK	night
999980	long_email	generic	Thursday	FR	morning
999998	long_email	generic	Friday	FR	afternoon

99950 rows × 5 columns

In [294…

```python
print('These are the reference levels')
print('---------------------------')
df_category.apply(lambda x: x.cat.categories[0])
```

```
These are the reference levels
---------------------------
```

Out[294]:

```
email_text        long_email
email_version        generic
weekday               Friday
user_country              ES
time_category      afternoon
dtype: object
```

3.3 Dummies etc.,

In [295…

```python
df = pd.get_dummies(df, drop_first=True)
```

In [296…

```python
#add intercept for multiple logistc regression
df['intercept'] = 1

#Getting data and target variable
X = df.drop('clicked',axis = 1) #data
y = df['clicked'] #data
```

3.4 Build the logistic regression model

The reason we choose logistic model because the output is binary(clicked(yes) OR didn't click (no))

In [297…

```python
#first model

logit_1 = sm.Logit(y,X).fit()
```

```
Optimization terminated successfully.
         Current function value: 0.092690
         Iterations 9
```

In [298…

```python
print(logit_1.summary())
```

```
                           Logit Regression Results
==============================================================================
Dep. Variable:                clicked   No. Observations:                99950
Model:                          Logit   Df Residuals:                    99934
Method:                           MLE   Df Model:                           15
Date:                Wed, 07 Sep 2022   Pseudo R-squ.:                 0.08002
Time:                        06:17:24   Log-Likelihood:                -9264.4
converged:                       True   LL-Null:                       -10070.
Covariance Type:            nonrobust   LLR p-value:                     0.000
==============================================================================================
                                 coef    std err          z      P>|z|      [0.025      0.975]
----------------------------------------------------------------------------------------------
user_past_purchases            0.1879      0.006     32.809      0.000       0.177       0.199
email_text_short_email         0.2776      0.045      6.127      0.000       0.189       0.366
email_version_personalized     0.6382      0.047     13.601      0.000       0.546       0.730
weekday_Monday                 0.5432      0.093      5.814      0.000       0.360       0.726
weekday_Saturday               0.2844      0.098      2.908      0.004       0.093       0.476
weekday_Sunday                 0.1836      0.100      1.834      0.067      -0.013       0.380
weekday_Thursday               0.6228      0.092      6.744      0.000       0.442       0.804
weekday_Tuesday                0.6167      0.092      6.675      0.000       0.436       0.798
weekday_Wednesday              0.7594      0.091      8.358      0.000       0.581       0.937
user_country_FR               -0.0790      0.163     -0.486      0.627      -0.398       0.240
user_country_UK                1.1565      0.122      9.474      0.000       0.917       1.396
user_country_US                1.1430      0.116      9.855      0.000       0.916       1.370
time_category_evening         -0.3506      0.139     -2.530      0.011      -0.622      -0.079
time_category_morning          0.0058      0.054      0.108      0.914      -0.100       0.112
time_category_night           -0.2564      0.067     -3.808      0.000      -0.388      -0.124
intercept                     -6.6845      0.148    -45.029      0.000      -6.975      -6.394
==============================================================================================
```

Inference

- Weekday Sunday is not significant (p value greater than 0.05), which is consistent with what we discovered in the EDA section.
- User_country_FR is not significant (p value greater than 0.05), which is consistent with what we discovered in the EDA section.
- Time morning(6-12) has a very high P value. We should drop this variable

In [301…

```python
#second model

#reset X (getting rid of not significant variables)
X_2 = df.drop(['clicked','weekday_Sunday','user_country_FR','time_category_morning'],axis = 1) #data

logit_2 = sm.Logit(y,X_2).fit()
```

```
Optimization terminated successfully.
         Current function value: 0.092709
         Iterations 9
```

In [302…

```python
print(logit_2.summary())
```

```
                           Logit Regression Results
==============================================================================
Dep. Variable:                clicked   No. Observations:                99950
Model:                          Logit   Df Residuals:                    99937
Method:                           MLE   Df Model:                           12
Date:                Wed, 07 Sep 2022   Pseudo R-squ.:                 0.07984
Time:                        06:20:05   Log-Likelihood:                -9266.2
converged:                       True   LL-Null:                       -10070.
Covariance Type:            nonrobust   LLR p-value:                     0.000
==============================================================================================
                                 coef    std err          z      P>|z|      [0.025      0.975]
----------------------------------------------------------------------------------------------
user_past_purchases            0.1880      0.006     32.808      0.000       0.177       0.199
email_text_short_email         0.2771      0.045      6.116      0.000       0.188       0.366
email_version_personalized     0.6388      0.047     13.616      0.000       0.547       0.731
weekday_Monday                 0.4466      0.076      5.906      0.000       0.298       0.595
weekday_Saturday               0.1877      0.081      2.319      0.020       0.029       0.346
weekday_Thursday               0.5264      0.074      7.084      0.000       0.381       0.672
weekday_Tuesday                0.5199      0.074      6.994      0.000       0.374       0.666
weekday_Wednesday              0.6626      0.072      9.149      0.000       0.521       0.805
user_country_UK                1.1952      0.094     12.751      0.000       1.012       1.379
user_country_US                1.1821      0.086     13.802      0.000       1.014       1.350
time_category_evening         -0.3526      0.134     -2.633      0.008      -0.615      -0.090
time_category_night           -0.2610      0.057     -4.572      0.000      -0.373      -0.149
intercept                     -6.6232      0.108    -61.196      0.000      -6.835      -6.411
==============================================================================================
```

Inference

- all variables are significant

4. Main takeaways

4.1 user_country

- The reference level for country is ES(Spain)
- We can see that UK/US countries perform significantly better than non-English speaking countries
  - The reason could either be that our translation is bad OR they are not our target market

We should either hire a better team to translate the materials OR try to target different market

4.2 weekday

- The reference level for weekday is Friday
- We can see that Monday to Saturday perform consistently well than Friday except Sunday which we've already removed

We should avoid sending emails from Friday to Sunday and try to send the emails in the middle of the week as Wednesday has the highest coefficient which indicates that they can generate more clicks(the response of this model).

4.3 email_length

- The reference level is long email
- shorter emails can create more clicks than the long emails

This might indicate that we should try to make our email content concise and avoid any verbose sentences or long content

4.4 personalized

- The reference level is generic email
- Personalized email generates much more clicks than the generic emails.

Therefore, consider how much more clicks that can be generated with personalized emails, we should try to make the emails more personalized and avoid any generic content in the following email campaign

4.5 time_category

- The reference level is afternoon (12 - 18)
- We can see that evening(18-24) night(0-6) perform poorly compared to the afternoon time categories
- Morning drop because it's not significant

Therefore, the marketing team should aim to send out emails in the afternoon(12 am to 6 pm)