**Relax Project Report**

# **Problem Identification**

What are the features that impact a customer for signing in more than 2 times a week on different days?

# **Data Exploration**

|  |
| --- |
| <class 'pandas.core.frame.DataFrame'>  Int64Index: 207917 entries, 0 to 207916  Data columns (total 13 columns):  # Column Non-Null Count Dtype  --- ------ -------------- -----  0 time\_stamp 207917 non-null object  1 user\_id 207917 non-null int64  2 visited 207917 non-null int64  3 object\_id 207917 non-null int64  4 creation\_time 207917 non-null object  5 name 207917 non-null object  6 email 207917 non-null object  7 creation\_source 207917 non-null object  8 last\_session\_creation\_time 207917 non-null float64  9 opted\_in\_to\_mailing\_list 207917 non-null int64  10 enabled\_for\_marketing\_drip 207917 non-null int64  11 org\_id 207917 non-null int64  12 invited\_by\_user\_id 116887 non-null float64  dtypes: float64(2), int64(6), object(5) |
| Figure 1: Histogram of Emotion Column |

Most of the data is int or object data a histogram didn’t show anything interesting.

# **Data Wrangling**

Here are the data wrangling tasks:

1. Merge the **df\_engage** , **user\_id** and **object\_id** into **df\_join**

2. Convert **time\_stamp** and **creation\_time** into datetime dtypes

3. Drop **invited\_by\_user\_id**

4. Create dummy columns for **creation\_source** and drop a column

5. Group **df\_join** by **time\_stamp** with a frequency of 1 week with the beginning of the week on a Monday

6. Create a dataframe called **df\_group** that is grouped by **user\_id** and **time\_stamp**

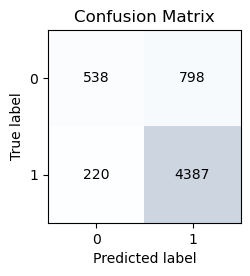
with frequency of every week, with the week starting on Monday

7. Count the logins of **df\_group** using .size() method for the groupby object

8. Reset index and turn **df\_count** into a DataFrame using the **user\_id** column as the index to join on using right join(preserve the original dataframe and feature rows

# **Modeling**

Since the data was unbalanced after merging the dataframes, I choose to do oversampling and under sampling, and resampled X and y before splitting the data into training and testing datasets. I used MCC as the metric since it was a balanced metric, along with a confusion metric. I used a default random forest model, got a MCC score of .44, which isn’t good enough but is starting to show progress from .10 when I started. The confusion matrix also looks better.



Chart

Description automatically generated

# **Conclusion & Recommendations**

Seems like Random Forest was starting to work well, looks like the **last\_session\_creation\_time** was the most important(but not really useful because it's the last time they logged in) in the model what organization they belong to as well as invited by another users, opting into mailing list and getting marketing emails.

For future projects I would recommend trying out more ensemble methods, as well as tuning them, collecting data just for the week as well as the weekends, and doing some A/B testing to see what changes work or not.