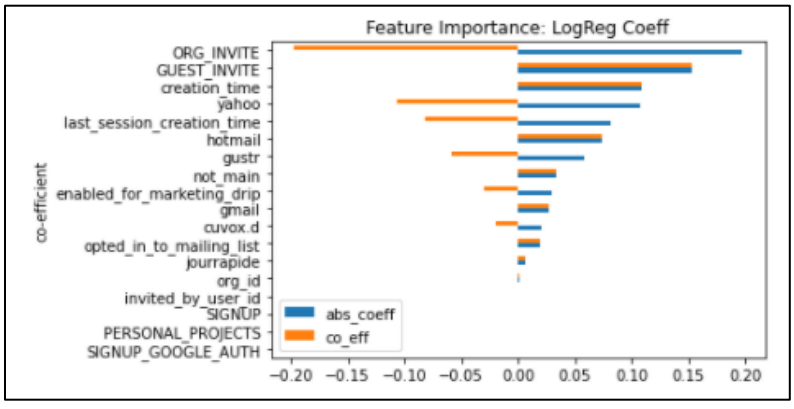
Relax Data Science Challenge Writeup

1. Inspect and understand the user data to choose and wrangle Features Set: We start by noticing that we can extract potentially useful modelling data from *email*, *creation_time* and *last_session_creation_time (last_session)* by retrieving the email domain and discretizing periodic features of the datetime data. Without further information on product, we pragmatically bin *creation_time* to calendar quarters and *last_session* to 6-hour periods starting from midnight. It's somewhat tenuous to include *last_session* as we do not know if this is representative of general login behavior of the user, but we will keep it to see if important. If it is, we can investigate that question further otherwise we just drop it. We also keep *invited_by_user_id* as this could be very interesting information for the business if it has predictive power. It would allow us to incentivize these "start-recruiters" and explore of they have a profile that we could use to proactively find new recruiters. Name is dropped for features set. We also clean up the new email domain category by setting any domain under 10% to "not_main" after noting that the top 6 domains account for 90% of emails. We also use pandas *get_dummies* to form one-hot-vectors for the categorical columns.
We notice that *last_session* and *invited_by_user_id* have null values representing 27% and 46% respectively. We decide not to use imputation as this could badly skew the data for discretized features. Moreover, the aim is identifying important variables and not necessarily best predictive capacity at this stage, so less data is more palatable and we drop rows with any nan. Ideally, these variables turn out not to be important and we just drop them for the final predictive model anyway.

2. Wrangle engagement data to form binary target variable: We use various pandas manipulations, *aggfuncs* and the inbuilt *rolling()* function to calculate a column of one or zeros for 'adopted' for each user. We note that we only have an adoption rate of 18% meaning we may want to consider imbalanced data when focusing on the predictive model. We do an inner join on the *user_id* (*object_id*) column with the features set and then separate the features and target dataframes / arrays for input into the modelling functions.

3. Features Importance: We will use logistic regression with balanced class_weights to get the feature importance on the full set of features with truncated data (no nans) as the baseline. We also do a check (not presented against random forest features importance to make sure we are not missing out on anything material). This gives the following results:



Top Features	
1.	Email Domain
2.	Creation Source
3.	Creation Time
4.	Enabled for marketing
5.	Mailing opt in

Confusion Matrix

Predicted / Actual	not adopted	adopted
not adopted	429	333
adopted	99	95

Support: 4,776

4. Next steps: The next steps would be to limit the model to the features above (null features not here so can enlarge sample substantially to 12,000 support and there is clearly room for modelling improvement!) and run for various combinations of features choosing the 'most important features' in reference to the model that gives the best metric on the test set. This process will tease out feature correlation better and the most important features will also depend on the chosen metric and the implied trade off between false positives and false negatives.