- The two dataframes, user table and usage summary table, need to be merged. In the user_table the column name ‘object_id was renamed user_id to make merging the two dataframes easier. There they were merged on user_id.

Code: **df = data2.merge(data1,on='user_id',how='outer')**

- Time_stamp column was renamed login_time for easier analysis
- Visited column had missing values after merging for customers who did not login in, here nans were filled with 0

Code: **df.visited = df.visited.fillna(0)**

- The response variable needs to be created which is “adopted user" who is a user who has logged into the product on three separate days in at least one sevenday period, to do this a few steps were taken
    - Create a grouped dataframe, grouped by user id so

group = df.groupby(['user_id'])['visited'].count().reset_index()

    - Then get users with visited value more than 1 because only 1 means they just created the account but did not login active_users = group[group.visited > 1]
    - A new column with 1 for active and 0 for not active is created by using a mask of the list of users

df_active = df[df.user_id.isin(active_users.user_id)]

    - Calculate 7 days from login time. If the next login time was in less than 7 days then true for active. Then the trues were summed per user and only users with 3 or more were kept

```
df_active['seven_days'] = df_active['login_time']+timedelta(7)
df_active['active'] = df_active.login_time <
                                        df_active.seven_days.shift(-1)
active_users =
df_active.groupby('user_id')['active'].sum().reset_index()
adopted_users = active_users[active_users.active>2]
```

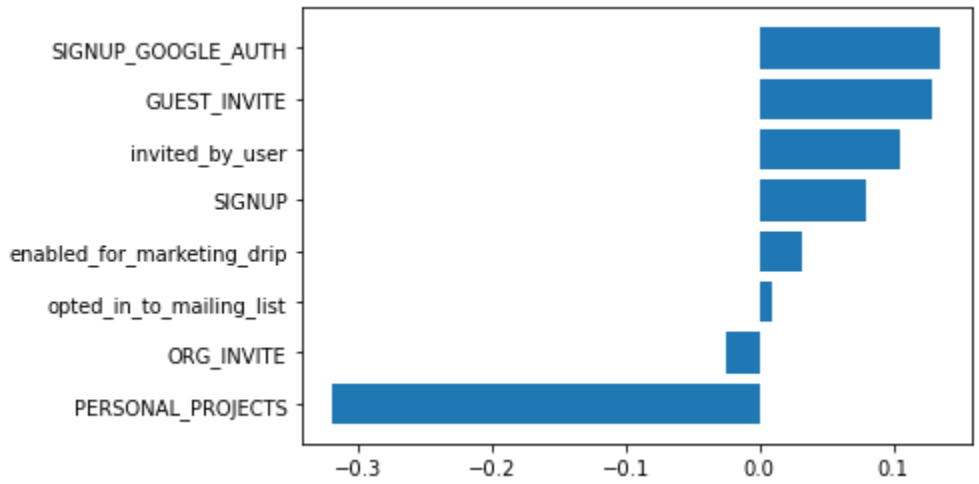
- A new feature was created Adopted_users that has 1 for adopted and 0 for not. Out of 12,000 users 2,106 are considered adopted, then this list was used to add a new feature to the original user data set with whether they were adopted or not.

```
adopted_user = np.where(df.user_id.isin(adopted_users.user_id),1,0)
df['adopted_user'] = adopted_user
```

- The data has a 17% adoption rate.

The user dataframe is modified:

- Features kept are: creation_source, opted_in_to_mailing_list, enabled_for_marketing_drip, org_id, invited_by_user_id, adopted_user
- invited_by _user_id has nans which I am assuming to be customers not invited, so it was replaced with 1s if invited and 0 if nan
- X and y were created, y is adopted
- Then creation_source was onehotencoded
- Train and test, then run logistic regression and run .coef_ for importance and this is the result I got



The most important feature is creation_source, followed by invited by user