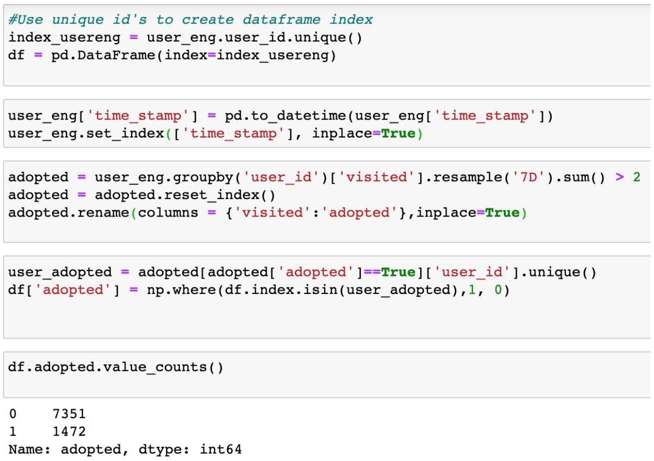
Relax Data Science Challenge

Brandon Arcilla

Some of the important features that can predict future adoption of this product is login count, average days between each login, what quarter of the year do the logins occur, weekday or weekend login occurrence, and when the user created their account.

Before getting the important features, I first had to find out which users adopted the product.



```
#Use unique id's to create dataframe index
index_usereng = user_eng.user_id.unique()
df = pd.DataFrame(index=index_usereng)
```

```
user_eng['time_stamp'] = pd.to_datetime(user_eng['time_stamp'])
user_eng.set_index(['time_stamp'], inplace=True)
```

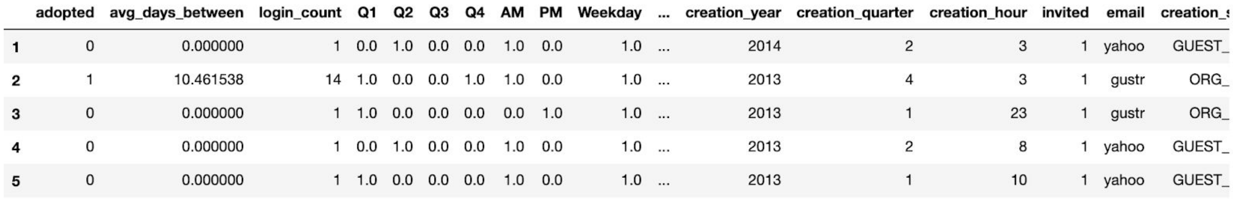
```
adopted = user_eng.groupby('user_id')['visited'].resample('7D').sum() > 2
adopted = adopted.reset_index()
adopted.rename(columns = {'visited':'adopted'},inplace=True)
```

```
user_adopted = adopted[adopted['adopted']==True]['user_id'].unique()
df['adopted'] = np.where(df.index.isin(user_adopted),1, 0)
```

```
df.adopted.value_counts()
```

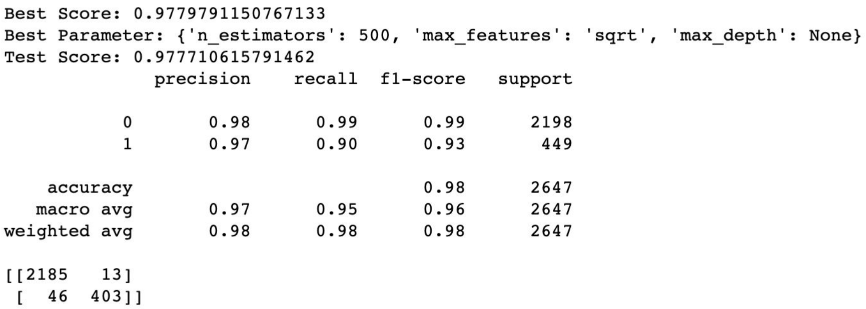
```
0    7351
1    1472
Name: adopted, dtype: int64
```

Many different features were created using the user engagement table and the user information table, such as login count, average days between each login, what quarter did the login occur, did a login occur during the weekend or weekday, when was the account created, was the user invited, what email domain is used.



	adopted	avg_days_between	login_count	Q1	Q2	Q3	Q4	AM	PM	Weekday	...	creation_year	creation_quarter	creation_hour	invited	email	creation_s
1	0	0.000000	1	0.0	1.0	0.0	0.0	1.0	0.0	1.0	...	2014	2	3	1	yahoo	GUEST_
2	1	10.461538	14	1.0	0.0	0.0	1.0	1.0	0.0	1.0	...	2013	4	3	1	gustr	ORG_
3	0	0.000000	1	1.0	0.0	0.0	0.0	0.0	1.0	1.0	...	2013	1	23	1	gustr	ORG_
4	0	0.000000	1	0.0	1.0	0.0	0.0	1.0	0.0	1.0	...	2013	2	8	1	yahoo	GUEST_
5	0	0.000000	1	1.0	0.0	0.0	0.0	1.0	0.0	1.0	...	2013	1	10	1	yahoo	GUEST_

These features were fed into a Random Forest Classifier to predict whether the user has adopted or not with a test accuracy of 98%.

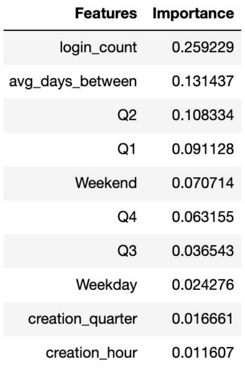


```
Best Score: 0.9779791150767133
Best Parameter: {'n_estimators': 500, 'max_features': 'sqrt', 'max_depth': None}
Test Score: 0.977710615791462
              precision    recall  f1-score   support

           0       0.98      0.99      0.99      2198
           1       0.97      0.90      0.93       449

    accuracy                           0.98      2647
   macro avg       0.97      0.95      0.96      2647
weighted avg       0.98      0.98      0.98      2647

[[2185   13]
 [  46  403]]
```



Features	Importance
login_count	0.259229
avg_days_between	0.131437
Q2	0.108334
Q1	0.091128
Weekend	0.070714
Q4	0.063155
Q3	0.036543
Weekday	0.024276
creation_quarter	0.016661
creation_hour	0.011607