DS-SF-27 Final Project

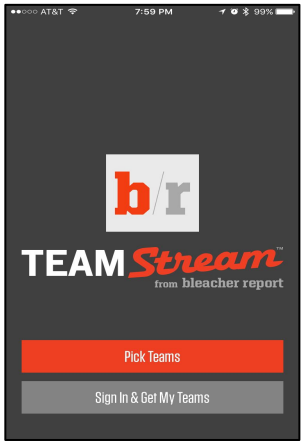
Andrew Burke

Project Problem & Hypothesis

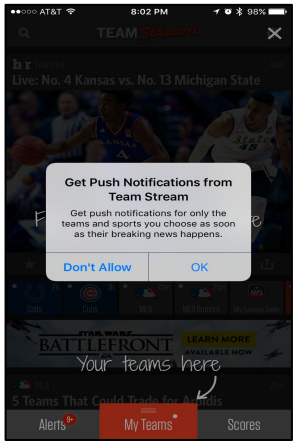
- Problem
  - I want to predict when a user will be retained after using our mobile app for the first time. If we are able to reliably predict this, we can then design features to improve retention.
- Hypothesis
  - As users add more streams/view more articles/enable push notifications in their first session, the higher the probability they will be retained.

- Machine Learning Model
  - This is a classification problem, and the outcome of the machine learning model will be the probability that a user will be retained after their first session.

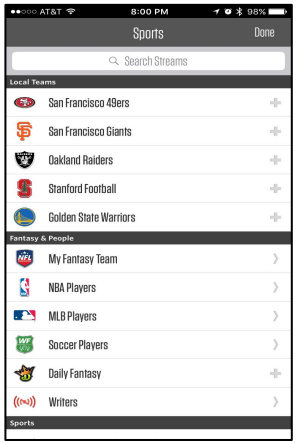
New User Flow



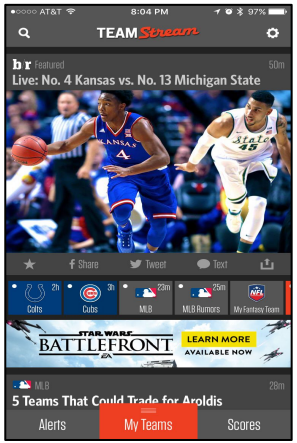
Launch App



Enable Notifications



Add Streams



View Articles

The Data

```
df = pd.read_csv(os.path.join('..', 'Datasets', 'BR_data.csv'))
df = df.set_index('user_identities.identity')
df
```

	articles_viewed	streams_added	sessions	greater_than_5_sessions	push_enabled
user_identities.identity					
4.32E+17	1	1	1	0	0
2.39E+18	1	5	1	0	0
5.27E+18	2	5	1	0	0
0000000000000a419698266553970235	3	3	1	0	0
000000000000a6409315927365159953	1	8	1	0	0
...	...	...	...	...	...
ffeeaa76195841efab257d819dac7bbb	2	27	12	1	0
fff0a4ec1e7548b2a8b3245defbeabc1	15	51	9	1	0
fff16bbe71f74b06a2e0b58186be2db9	1	5	2	0	0
fff8033f52ac4d67ae61f2e70559e681	2	3	3	0	0
fffc6535ff9441ee81e321e95374ed77	2	28	6	1	1

25854 rows × 5 columns

EDA

```
df.describe()
```

	articles_viewed	streams_added	sessions	greater_than_5_sessions	push_enabled
count	25854.000000	25854.000000	25854.000000	25854.000000	25854.000000
mean	6.213623	65.120716	12.847026	0.637812	0.092597
std	10.057690	196.832907	15.016058	0.480642	0.289872
min	1.000000	1.000000	1.000000	0.000000	0.000000
25%	2.000000	8.000000	4.000000	0.000000	0.000000
50%	3.000000	21.000000	8.000000	1.000000	0.000000
75%	7.000000	58.000000	16.000000	1.000000	0.000000
max	481.000000	9953.000000	288.000000	1.000000	1.000000

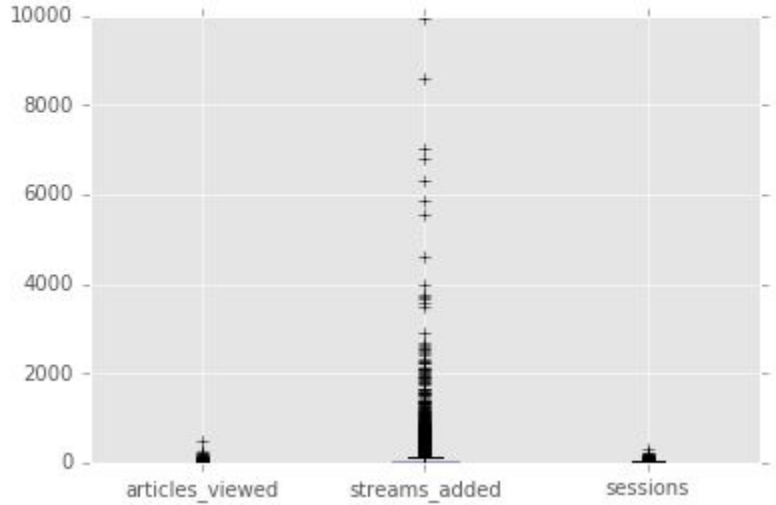
```
df.corr()
```

	articles_viewed	streams_added	sessions	greater_than_5_sessions	push_enabled
articles_viewed	1.000000	0.202627	0.582199	0.277830	-0.038056
streams_added	0.202627	1.000000	0.415350	0.188577	-0.024599
sessions	0.582199	0.415350	1.000000	0.492285	-0.018064
greater_than_5_sessions	0.277830	0.188577	0.492285	1.000000	-0.001644
push_enabled	-0.038056	-0.024599	-0.018064	-0.001644	1.000000

EDA

```
df[['articles_viewed', 'streams_added', 'sessions']].plot(kind = 'box')
```

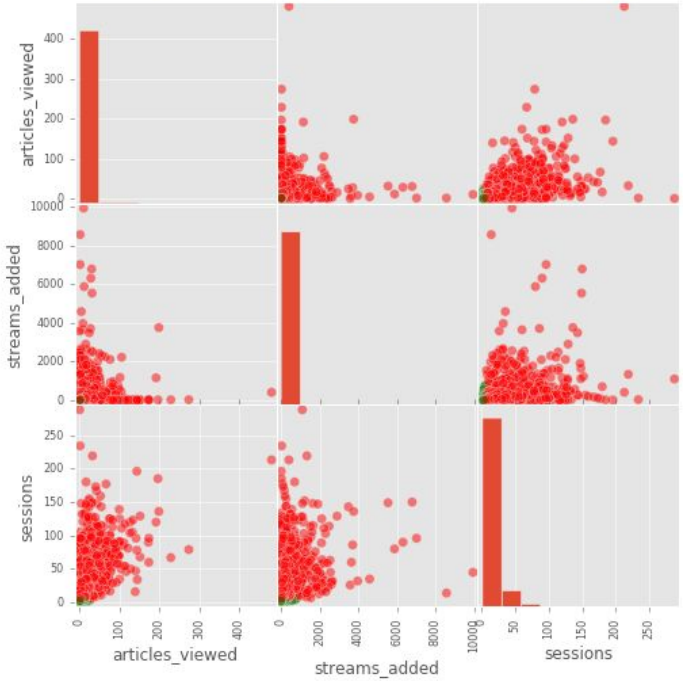
```
<matplotlib.axes._subplots.AxesSubplot at 0x11c470090>
```



EDA

```
pd.tools.plotting.scatter_matrix(df[ ['articles_viewed', 'streams_added', 'sessions'] ], s = 200,
figsize = (8, 8), c = color)
```

```
array([[<matplotlib.axes._subplots.AxesSubplot object at 0x1250d7d10>,
        <matplotlib.axes._subplots.AxesSubplot object at 0x1258a6dd0>,
        <matplotlib.axes._subplots.AxesSubplot object at 0x1258344d0>],
       [<matplotlib.axes._subplots.AxesSubplot object at 0x125602610>,
        <matplotlib.axes._subplots.AxesSubplot object at 0x12541f690>,
        <matplotlib.axes._subplots.AxesSubplot object at 0x125484450>],
       [<matplotlib.axes._subplots.AxesSubplot object at 0x123b2c4d0>,
        <matplotlib.axes._subplots.AxesSubplot object at 0x125978690>,
        <matplotlib.axes._subplots.AxesSubplot object at 0x125c5d490>]], dtype=object)
```



EDA

```
Q1_sa = df.streams_added.quantile(0.25)
Q3_sa = df.streams_added.quantile(0.75)

IQR_sa = Q3_sa - Q1_sa

IQR_sa
```

```
50.0
```

```
df.drop(df[df.streams_added > Q3_sa + 1.5 * IQR_sa].index, inplace = True)
df.shape[0]
```

```
23025
```

```
Q1_av = df.articles_viewed.quantile(0.25)
Q3_av = df.articles_viewed.quantile(0.75)

IQR_av = Q3_av - Q1_av

IQR_av
```

```
5.0
```

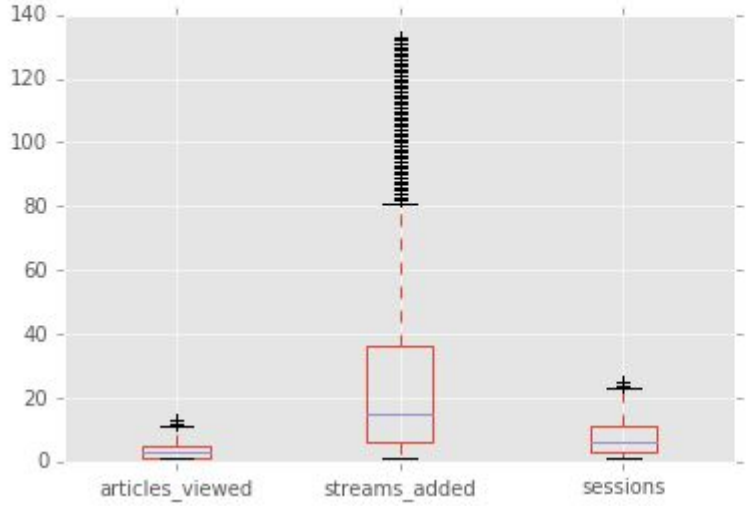
```
df.drop(df[df.articles_viewed > Q3_av + 1.5 * IQR_av].index, inplace = True)
df.shape[0]
```

```
21242
```

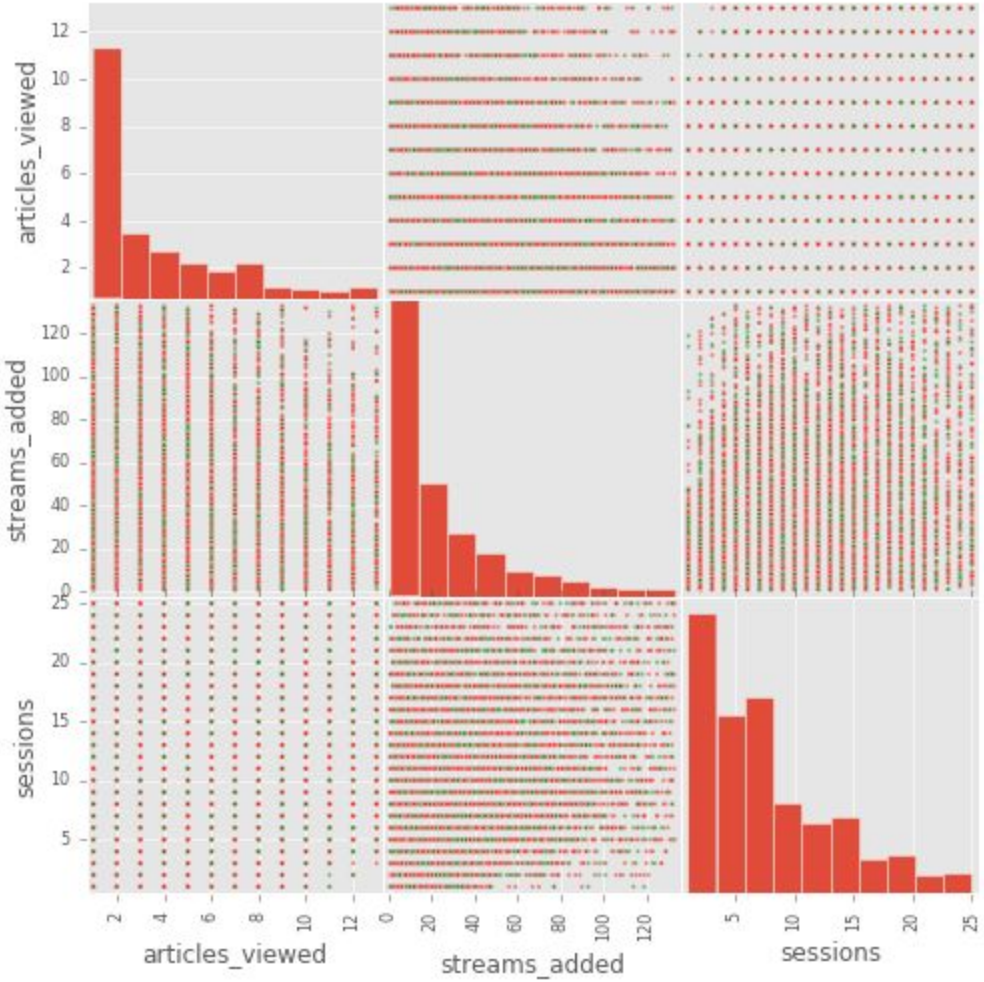
EDA

```
df[['articles_viewed', 'streams_added', 'sessions']].plot(kind = 'box')
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x11dea2650>
```

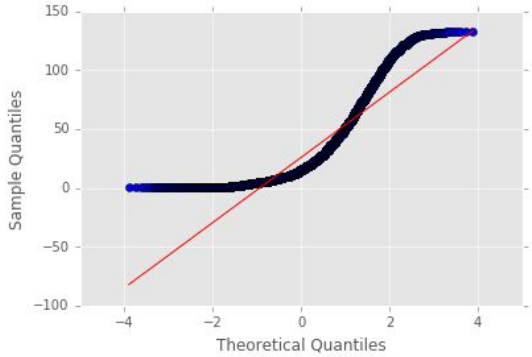


EDA

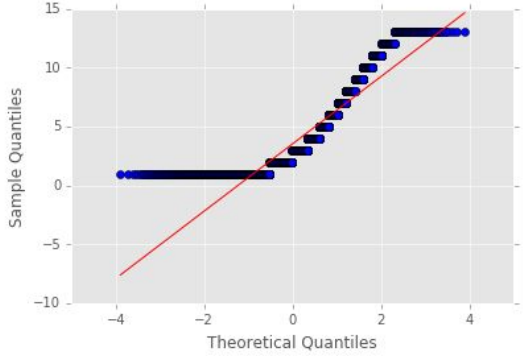

articles_viewed
streams_added
sessions
articles_viewed
streams_added
sessions

EDA

```
sm.qqplot(df.streams_added, line = 's')

pass
```




```
sm.qqplot(df.articles_viewed, line = 's')

pass
```

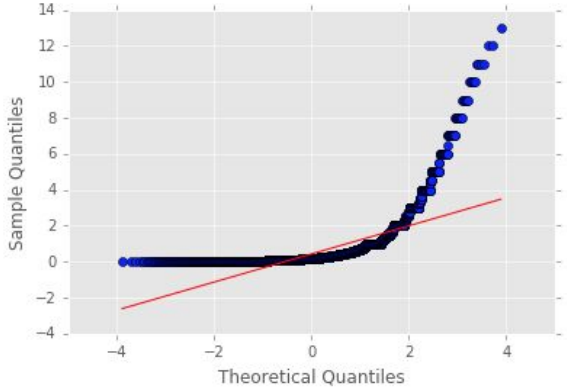


EDA

```
articles_per_stream = df.articles_viewed / df.streams_added
articles_per_stream
```

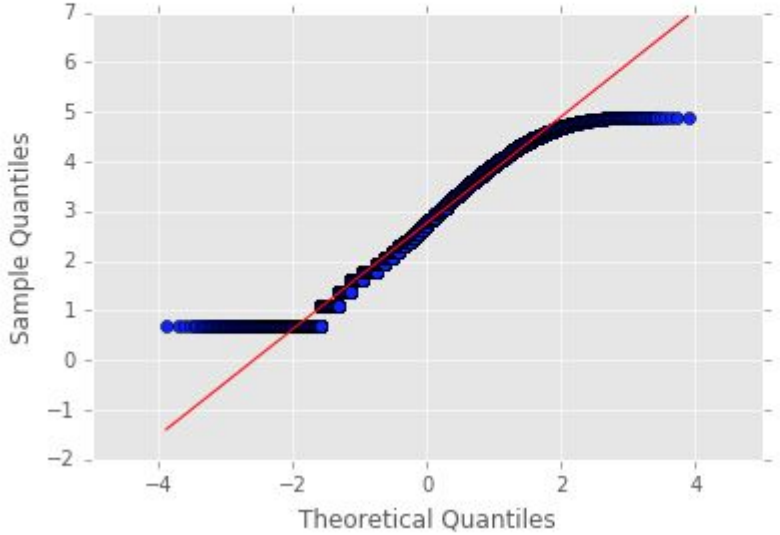
```
user_identities.identity
4.32E+17                                1.000000
2.39E+18                                0.200000
5.27E+18                                0.400000
0000000000000a419698266553970235        1.000000
000000000000a6409315927365159953        0.125000
                                           ...
ffebf65d7954494cabf9c7d932f5c916        0.500000
ffeeaa76195841efab257d819dac7bbb        0.074074
fff16bbe71f74b06a2e0b58186be2db9        0.200000
fff8033f52ac4d67ae61f2e70559e681        0.666667
fffc6535ff9441ee81e321e95374ed77        0.071429
dtype: float64
```

```
sm.qqplot(articles_per_stream, line = 's')

pass
```



EDA

```
sm.qqplot(df.streams_added.apply(lambda x: np.log(1 + x)), line = 's')
#sm.qqplot(df.streams_added, line = 's')


pass
```



Next Steps

- Random forest and feature importance
- Set up training and test sets
- Run cross validation on training set for logistic regression model
- Evaluate model
- Run final model on test set