Big Data & Business Analytics

Raul Cardenas | Luisa Toro | Thomas Werner | Aman Kumar

PROBLEM STATEMENT

Define scope of the project

PROBLEM

What makes a tweet viral? Is it possible to classify a tweet as viral or not based on its metadata?

BACKGROUND

Twitter wants to rely less on ads and more on subscrition services

RELEVANCE

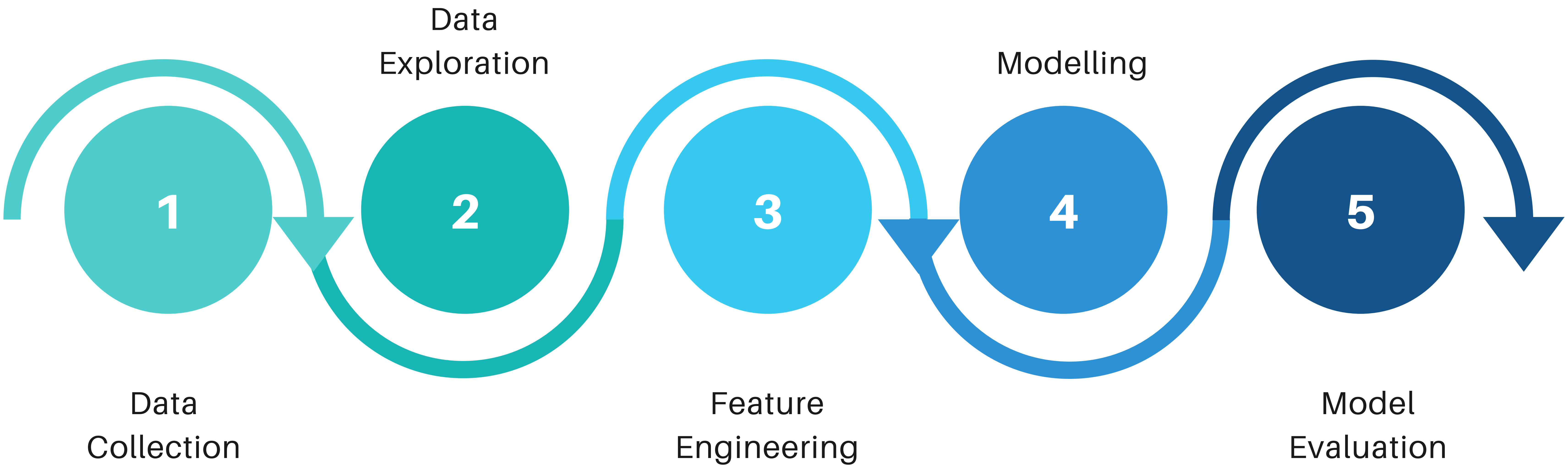
Current value proposition of Twitter Blue is not enticing enough

OBJECTIVE

Leverage AI to develop a comprehensive suite of creator tools to improve the value proposition of Twitter Blue

PROCESS

5-Step ML Process



I+II. DATA COLLECTION & EXPLORATION

Explore and extract relevant features

CONTENT

11,00 tweets extracted using Twitter's API

SPAN

2006 to 2018

RELEVANT FEATURES

- *retweet count* (int64)
- *favorite count* (int64)
- *created at* (datetime64[ns, UTC])
- *text* (object)
- *user* (object)
- *metadata* (object)

III. FEATURE ENGINEERING

Data preparation for modelling

1

TARGET COLUMN

Create a target column for the model to predict

2

ADD NEW FEATURES

Transform tweet metadata to generate desired variables for predictive modeling

3

ASSESS DATA IMBALANCE

Resample data to avoid biasing the model towards the dominant class

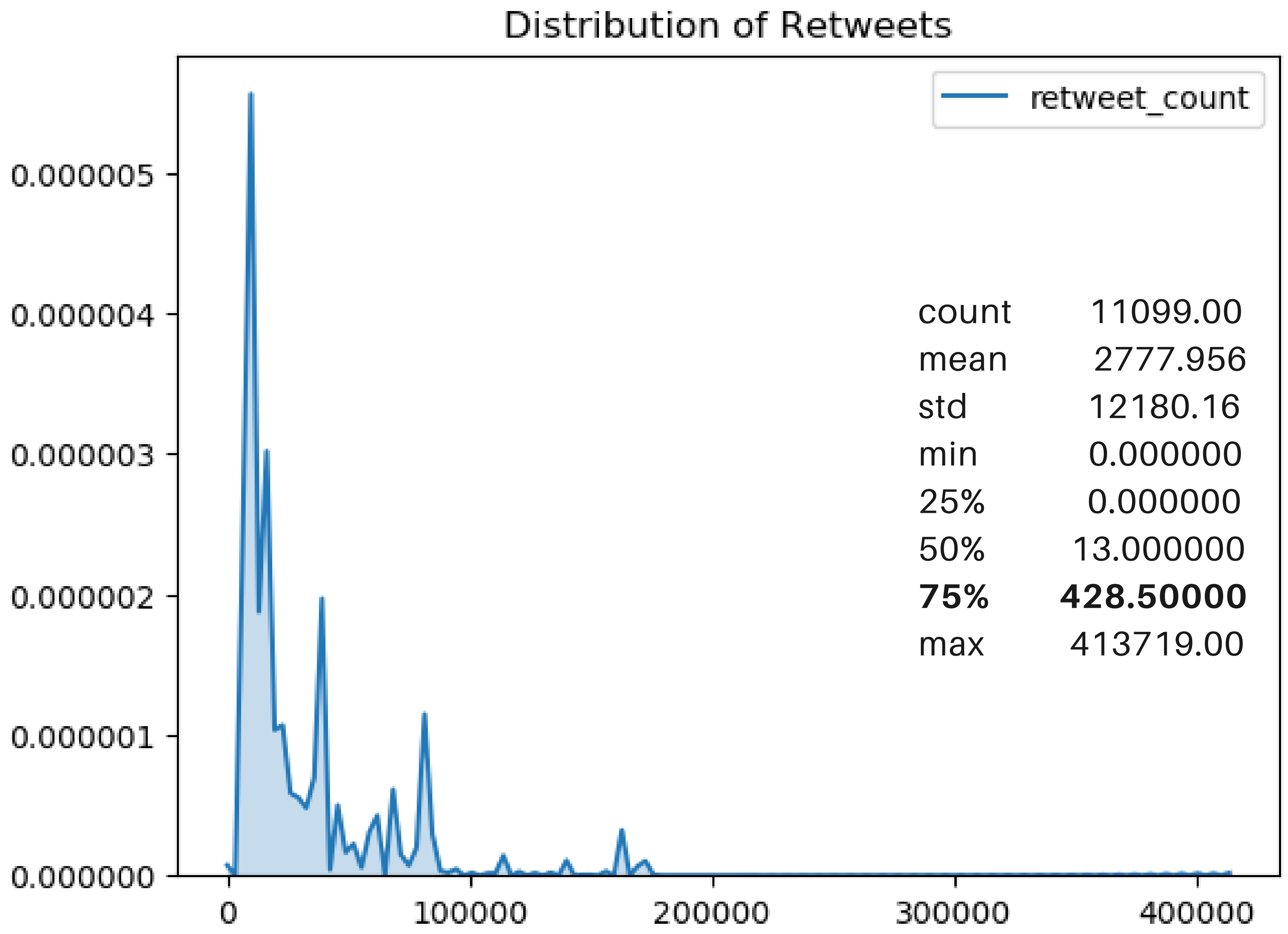
4

ENCODING + SCALING

Encode categorical features and normalize data

TARGET COLUMN

What makes a tweet viral?

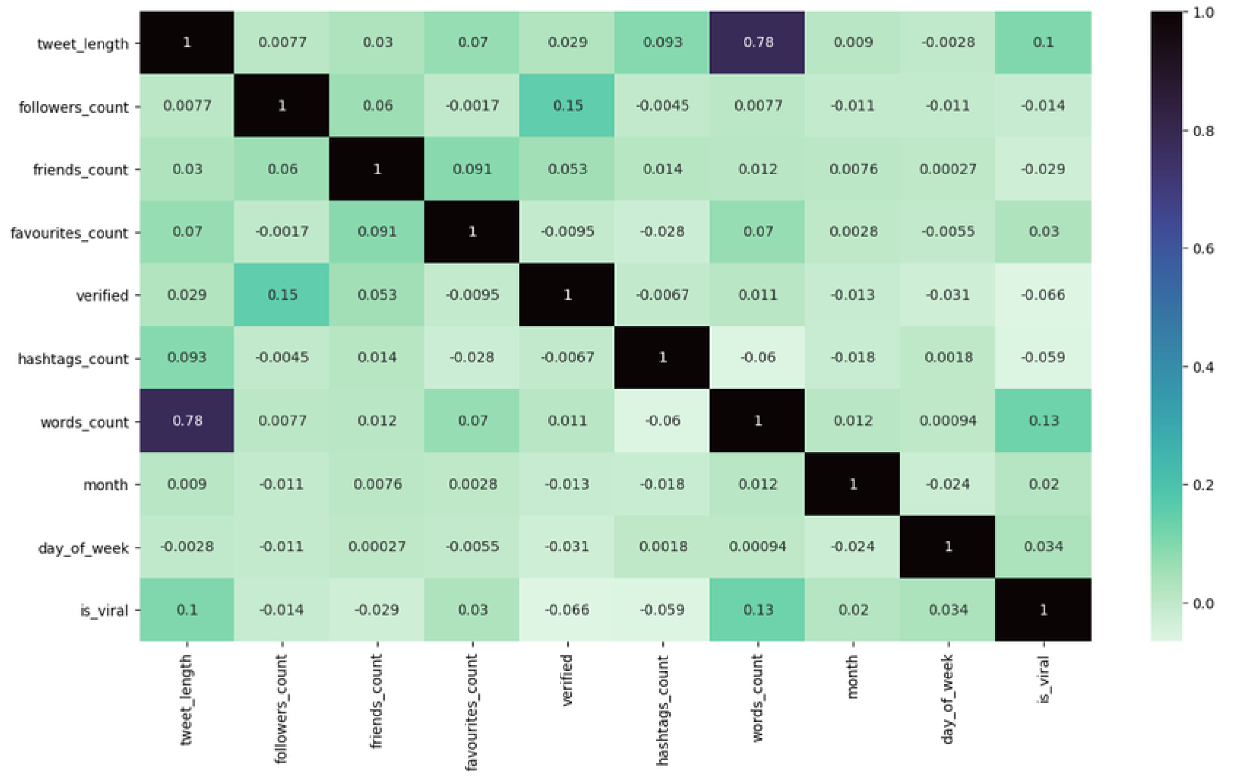


RETWEET COUNT	IS VIRAL
138	No
429	Yes
10,000	Yes
...	...

FEATURE ENGINEERING

List of features used for modelling

SAMPLE	COLUMNS
138	**tweet_length:** number of characters in tweet
300	**followers_count:** number of followers
145	**friends_count:** number of friends
5	**favorites_count:** total number of favorited
True	**verified:** account verification status
en	**language:** language used in tweet
2	**hashtags_count:** number of hashtags used in tweet
75	**words_count:** number of words in tweet
9	**month:** month in number format
5	**day_of_week:** day of the week in number format
positive	**sentiment:** tweet sentiment analysis using Dilbert NLP model



IV+V. MODELLING & EVALUATION

1

TRAIN/TEST SPLIT

80/20 train-test split (oversampling was performed after splitting the data)

2

MODELLING

Random Forest classifier, K-Neighbors classifier & Logistic Regression

3

EVALUATION

Confusion matrix + Performance metrics

4

CONCLUSION

Model selection, outcomes, and recommendations

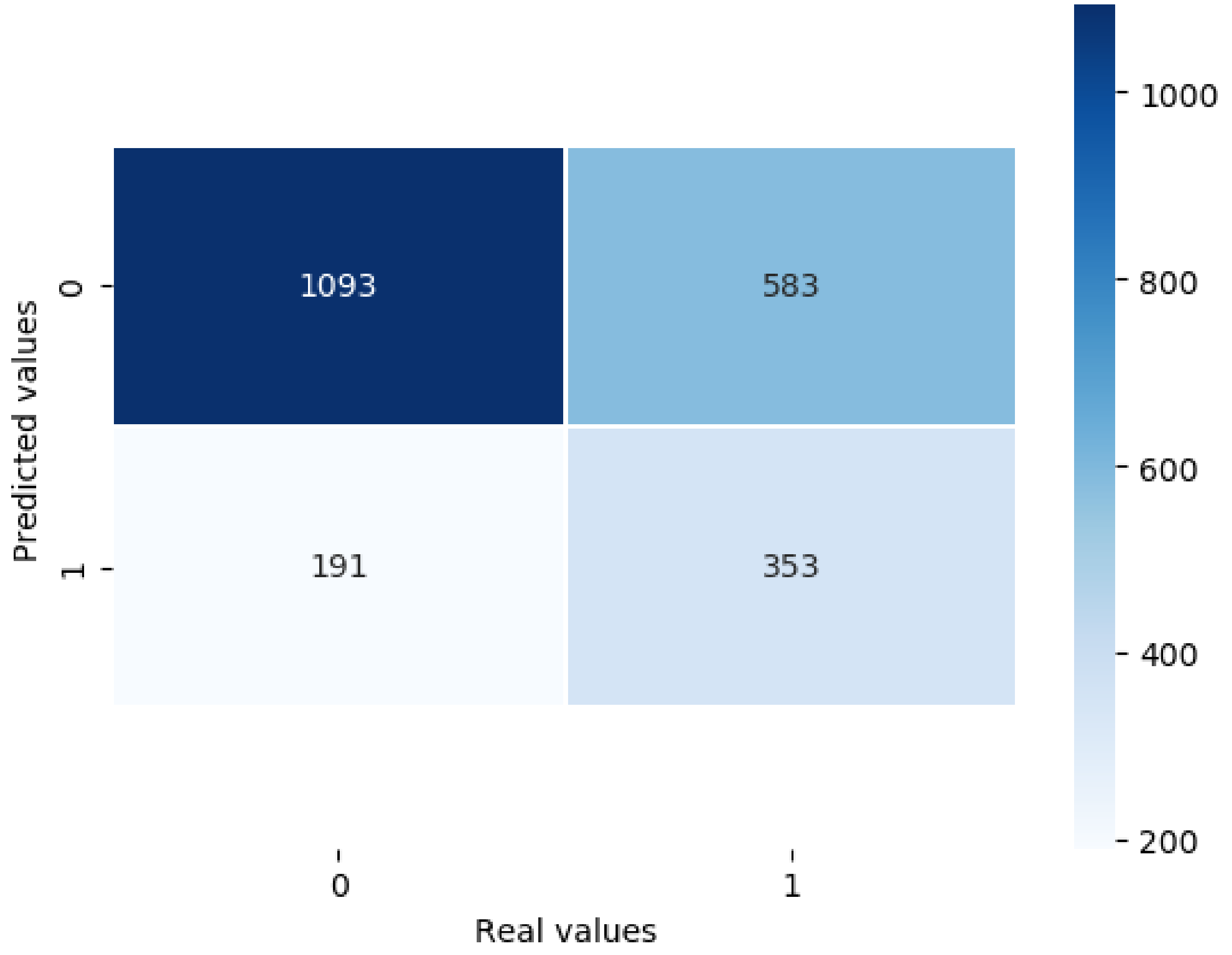
RANDOM FOREST CLASSIFIER

Confusion matrix + Performance metrics

Train: 0.8596
Test: 0.6540

	PRECISION	RECALL	F1-SCORE	SUPPORT
0	0.85	0.66	0.74	1676
1	0.38	0.65	**0.48**	544
Accuracy			0.65	2220
Macro Avg.	0.62	0.65	0.61	2220
Micro Avg.	0.74	0.65	0.68	2220

0 = not viral, 1 = viral - oversampled results

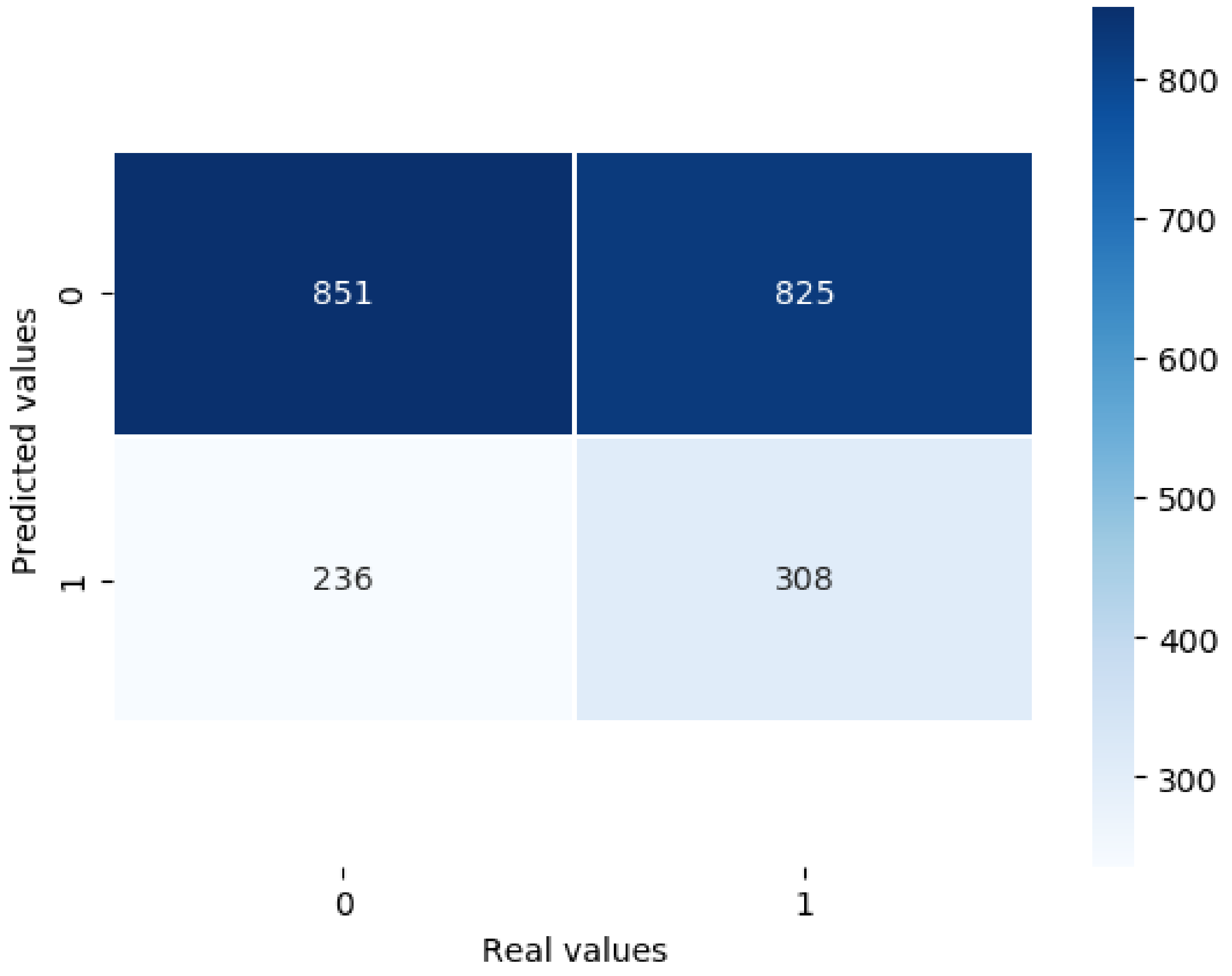


K-NEIGHBORS CLASSIFIER

Confusion matrix + Performance metrics

Train: 0.5868
Test: 0.5220

	PRECISION	RECALL	F1-SCORE	SUPPORT
0	0.78	0.51	0.62	1676
1	0.27	0.57	**0.37**	544
Accuracy			0.52	2220
Macro Avg.	0.53	0.54	0.49	2220
Micro Avg.	0.66	0.52	0.56	2220



0 = not viral, 1 = viral - oversampled results

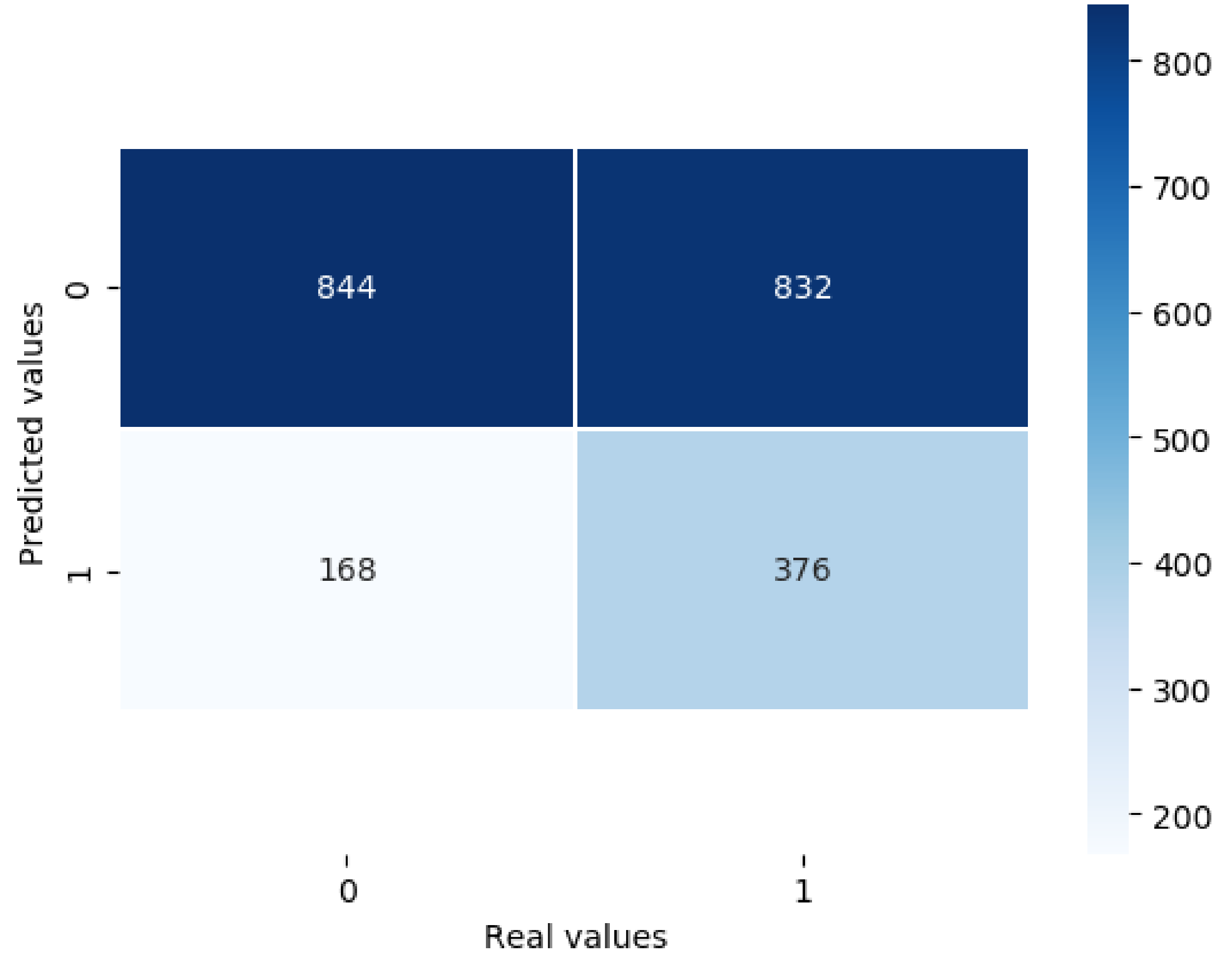
LOGISTIC REGRESSION

Confusion matrix + Performance metrics

Train: 0.5941
Test: 0.5495

	PRECISION	RECALL	F1-SCORE	SUPPORT
0	0.83	0.50	0.63	1676
1	0.31	0.69	**0.43**	544
Accuracy			0.55	2220
Macro Avg.	0.57	0.60	0.53	2220
Micro Avg.	0.71	0.55	0.58	2220

0 = not viral, 1 = viral - oversampled results



Conclusions

Model selection, value add, recommendations

MODEL SELECTION

Random Forest

1. Best f1-score for class 1
2. Ovrftg. can be addressed
3. Less false positives
4. Highest accuracy
5. Highest recall for class 0
6. Feature importance

VALUE ADD

Relevance

1. Feature importance
2. Understand user behavior
3. Improve engagement

NEXT STEPS

Recommendations

1. Improve target definition
2. Ensemble methods
3. Collect more data
4. Add new features (geo)
5. Resampling methods
6. Improve NLP analysis