Reddit NLP Analysis

Building the Best Model

Scott Rosengrants
DEN-Flex-10

Objectives

Can a model be built to predict one subreddit post from another?

If so, how is it optimized? How do we know?

What is the optimal model?

Can a model be built to determine several subreddits from one another?

Subreddits

r/Cooking

r/Cooking is a place for the cooks of reddit and those who want to learn how to cook. Post anything related to cooking here, within reason.
1.6m Members

r/Keto

r/Keto is place to share thoughts, ideas, benefits, and experiences around eating within a Ketogenic lifestyle.
1.7m Members

r/EatCheapAndHealthy

Eating healthy on a cheap budget
1.7m Members

r/DIY

A place where people can come to learn and share their experiences of doing, building and fixing things on their own.
17.0m DIYers

r/DataScience

A place for data science practitioners and professionals to discuss and debate data science career questions.
184k Members

Can a model be built to predict one subreddit post from another?

What is needed?

- Reddit submission data
- Clean text data
- A transformer
- A predictor
- A score

How it was achieved

- Web Scraping (Pushshift Reddit API)
- Quick EDA (splitting data, feature and target extraction)
- TFIDF Vectorizer (frequency–inverse document frequency)
- Logistic Regression or Gaussian Naive Bayes
- Accuracy

How is the model optimized?

$$\text{Accuracy} = \frac{\text{Correct Predictions}}{\text{Total Observations}}$$

Parameters = Individual model settings for both the transformer and the model

TFIDF

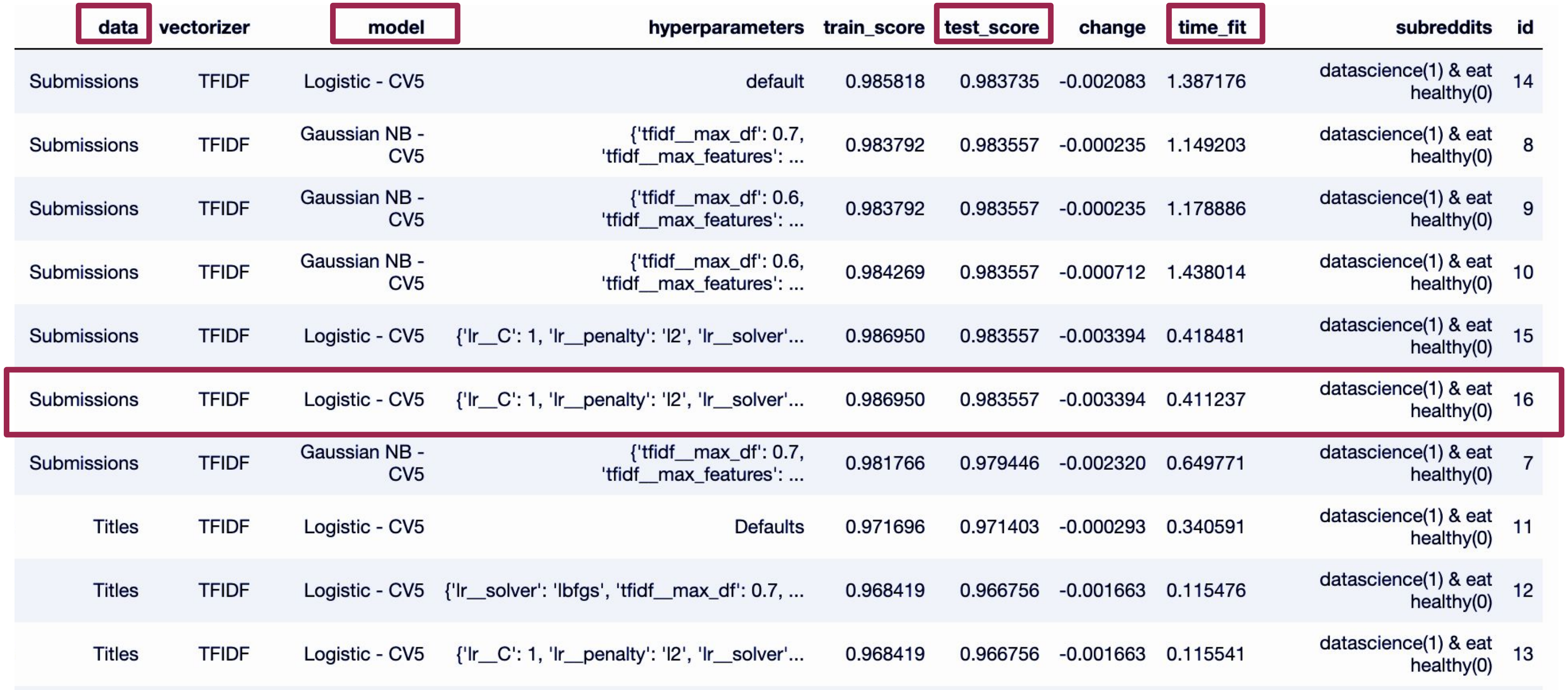
max_features
stop_words
ngram_range
max_df

Logistic Regression

C value
solver
penalty_type

Time = Time needed to fit

A quick look at the results



data	vectorizer	model	hyperparameters	train_score	test_score	change	time_fit	subreddits	id
Submissions	TFIDF	Logistic - CV5	default	0.985818	0.983735	-0.002083	1.387176	datascience(1) & eat healthy(0)	14
Submissions	TFIDF	Gaussian NB - CV5	{'tfidf__max_df': 0.7, 'tfidf__max_features': ...	0.983792	0.983557	-0.000235	1.149203	datascience(1) & eat healthy(0)	8
Submissions	TFIDF	Gaussian NB - CV5	{'tfidf__max_df': 0.6, 'tfidf__max_features': ...	0.983792	0.983557	-0.000235	1.178886	datascience(1) & eat healthy(0)	9
Submissions	TFIDF	Gaussian NB - CV5	{'tfidf__max_df': 0.6, 'tfidf__max_features': ...	0.984269	0.983557	-0.000712	1.438014	datascience(1) & eat healthy(0)	10
Submissions	TFIDF	Logistic - CV5	{'lr__C': 1, 'lr__penalty': 'l2', 'lr__solver'...	0.986950	0.983557	-0.003394	0.418481	datascience(1) & eat healthy(0)	15
Submissions	TFIDF	Logistic - CV5	{'lr__C': 1, 'lr__penalty': 'l2', 'lr__solver'...	0.986950	0.983557	-0.003394	0.411237	datascience(1) & eat healthy(0)	16
Submissions	TFIDF	Gaussian NB - CV5	{'tfidf__max_df': 0.7, 'tfidf__max_features': ...	0.981766	0.979446	-0.002320	0.649771	datascience(1) & eat healthy(0)	7
Titles	TFIDF	Logistic - CV5	Defaults	0.971696	0.971403	-0.000293	0.340591	datascience(1) & eat healthy(0)	11
Titles	TFIDF	Logistic - CV5	{'lr__solver': 'lbfgs', 'tfidf__max_df': 0.7, ...	0.968419	0.966756	-0.001663	0.115476	datascience(1) & eat healthy(0)	12
Titles	TFIDF	Logistic - CV5	{'lr__C': 1, 'lr__penalty': 'l2', 'lr__solver'...	0.968419	0.966756	-0.001663	0.115541	datascience(1) & eat healthy(0)	13

and so on ...

What is the optimal model?

After hundreds of iterations the best model was found to be a Logistic Regression model paired with a TFIDF Vectorizer.

Best Parameters:

TFIDF

max_features = 2000
stop_words = english
ngram_range = (1,1)
max_df = 0.7

Logistic Regression

C value = 1
solver = saga
penalty_type = L2

Test Scores: Train = 98.7% Test = 98.4%

Time: 0.42 Seconds

Let’s make it more challenging

Using the optimal Logistic Regression model and transformer let’s distinguish between two much more closely related subreddits.

r/Cooking
r/EatingCheapAndHealthly

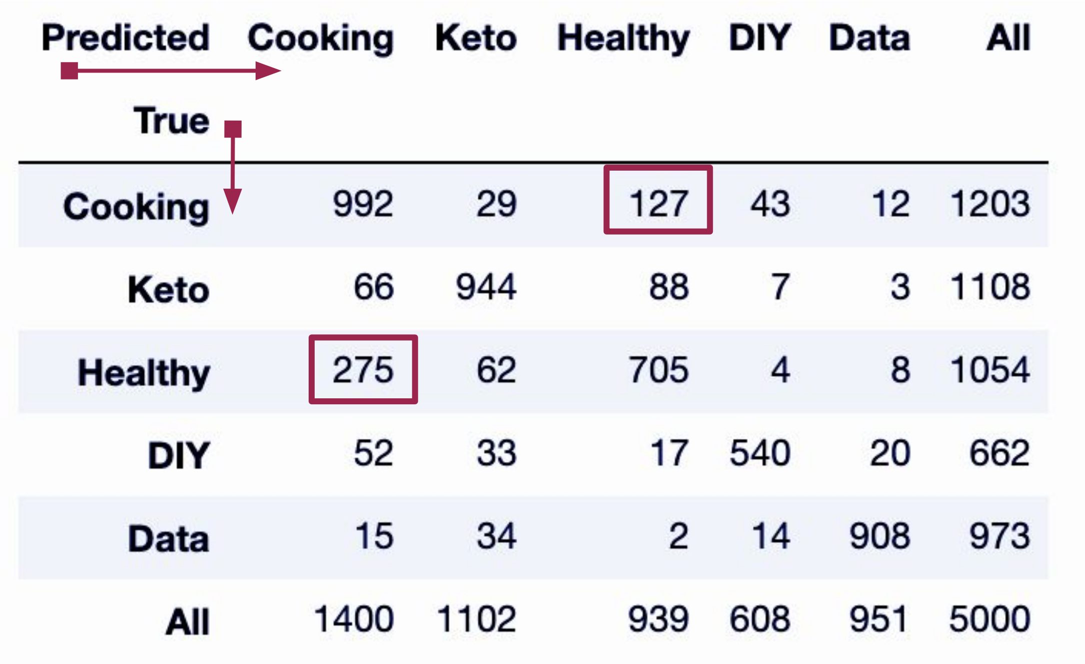
Test Scores: Train = 83.2% Test = 84.1%

Time: 0.38 Seconds

What this means: It is possible, but the model dips in performance by 14%

Can a model be built to determine several subreddits from one another?

C-Support Vector Classification with TFIDF



Predicted	Cooking	Keto	Healthy	DIY	Data	All
True						
Cooking	992	29	127	43	12	1203
Keto	66	944	88	7	3	1108
Healthy	275	62	705	4	8	1054
DIY	52	33	17	540	20	662
Data	15	34	2	14	908	973
All	1400	1102	939	608	951	5000

Next Steps

- Continue optimizing parameters for the closely related subreddits
- Tune the parameters of Classification Support Vector Machine model to improve performance
- Test the optimized models against the validation set of data
- Study the dictionaries of each optimized model to understand what words or phrases are most significant
- Automate this process for scale to be used with Reddit’s moderator bots

Questions?