keyword: ‘vegan’

Gifford Tompkins
Project 3 | DSI 09
18 October 2019

Problem Statement

MeatFreeLiving.net is a health food and lifestyle retailer and online community. We sell meat-free products and host online forums for a largely Vegan and Vegetarian clientele.

Though we are strictly meat-free, we do not sell exclusively vegan products.

To avoid advertising our egg and dairy products to our vegan users, we want to develop a model using Natural Language Processing to identifying our vegan users through their posts and comments on our online forums.

Model Guidelines and Objectives

Our model will try to identify Vegans and will optimize for **specificity** (i.e. *reduce false negatives*). We will aim for high specificity for two reasons:

1. A large number of our vegan users are animal rights activists. Some might have strong feelings about animal products and potentially leave our site/company if they are advertised egg or dairy products.
2. To avoid wasting advertising budget on users who are not interested in egg or dairy products. Vegetarians are likely still interested in vegan products, though not vice-a-versa. Therefore, ***false positives*** will not be as much of a concern for our model.

We will use ~10,000 posts from the two subreddits r/Vegan and r/Vegetarian to develop several models and then identify the method that has the highest accuracy and specificity.

Data Collection and Baseline Score

49.74%

50.26%

baseline

- Collected data by running ten, 500-document, batch requests from each subreddit with a 5 second delay through the PushiftIO API. Adjusting the 'before' parameter after each pull to avoid pulling duplicates.

- Looked at **average word count** and **character count** for posts but did not see any outstanding correlation, so decided to leave it out of my analysis.

- Ran ***'title'*** and ***'selftext'*** through **BeautifulSoup** and **WordNetLemmatizer** to clean the text and combined them into a single ***'text'*** column for analysis.

- Duplicates posts were removed before establishing our baseline score.

Data Exploration and Signalling Words

- 'vegan', 'non vegan', 'vegan year', 'new vegan'
- 'animal', 'cow', 'dog', 'pig', 'pet', 'farm', 'people', 'human', 'world'
- 'plant based', 'plant', 'vegan food'
- 'dairy', 'milk', 'product', 'leather', 'animal product', 'kill', 'industry', 'shoe', 'company', 'climate', 'climate change', 'environment'
- 'article', 'video', 'news', 'argument', 'say', 'study', 'watch', 'http', 'www', 'http www', ’com’, 'org',

- ‘vegetarian', 'vegetarian vegan', 'vegetarianism',
- 'meat', 'eating meat', 'fish', 'eat meat'
- 'recipe', 'veggie', 'meal', 'eat', 'eating', 'burger', 'tofu', 'make', 'bean', 'cook', 'time', 'chicken', 'protein', 'food', 'diet', 'dinner', 'cooking', 'dish', 'salad', 'veg', 'rice', 'mushroom', 'cheese', 'egg', 'sauce', 'easy', 'tomato', 'feel', 'vegetable', 'potato', 'idea'
- ‘Week’, ‘day’, ‘year’

Model Selection: The Machine

To select the best model, we created a function that passed different combinations of vectorizers and classifiers into a GridSearchCV function to find the best hyperparameters.

After building this function, our process will be as follows:

1. Select a set of hyperparameters for our ***vectorizer*** to fit our models.*
2. Fit and score the classifiers through the function mentioned above.
3. Store and evaluate their metrics into a dataframe.
    a. Metrics include:
        i. Best cross-validation score
        ii. Training Score
        iii. Test Score
        iv. Sensitivity
        v. Specificity
        vi. Run-time**
4. Finally, create an ensemble method using the best combination of vectorizer, classifier and hyperparameters from the previous 3 steps and evaluate that performance against our individual models.

Vectorizers

*evaluated with a DecisionTreeClassifier**

CountVectorizer()
TfidVectorizer()

```
vectorizer_params = {'vec__max_df': [0.80, 0.90, 1.0],
          'vec__max_features': [1000, 3000],
          'vec__min_df': [3, 5],
          'vec__ngram_range': [(1, 1),(1,2)],
          'vec__stop_words': [None,'english']]}

count_vectorizer_params = {'max_df': 0.8,
          'max_features': 3000,
          'vec__min_df': 5,
          'vec__ngram_range': (1, 2),
          'vec__stop_words': ‘english’}

tfid_vectorizer_params = {'max_df': 0.8,
          'max_features': 3000,
          'vec__min_df': 3,
          'vec__ngram_range': (1, 2),
          'vec__stop_words': ‘english’}
```

* *These hyperparameters gave the best training score for a DecisionTreeClassiffier with default values.*

Classifiers

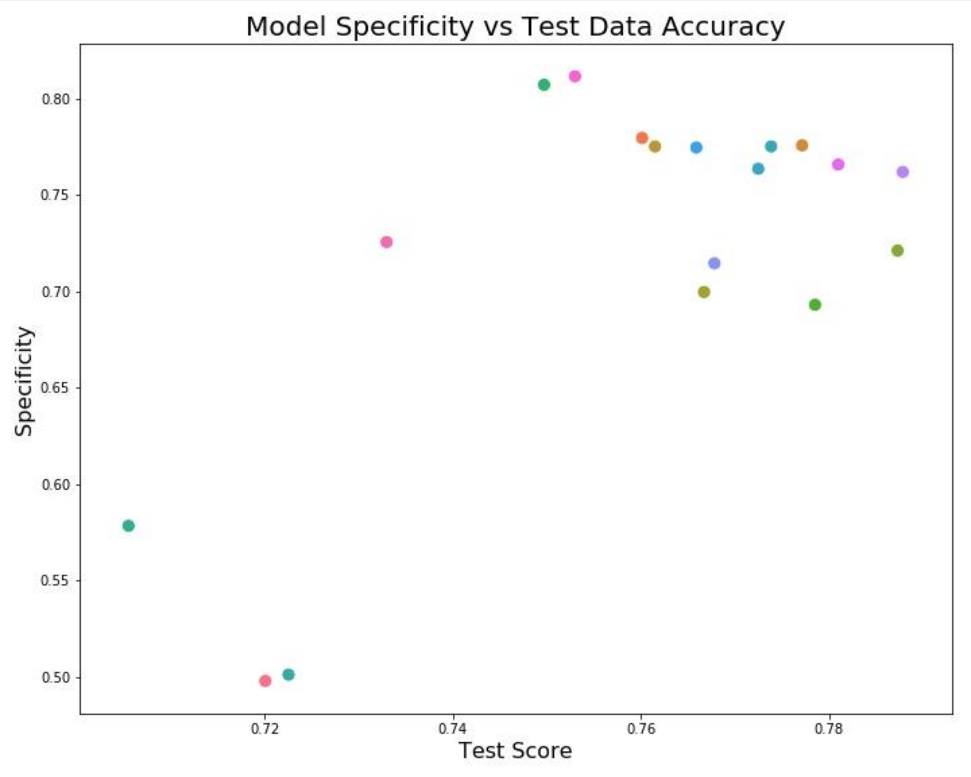
DecisionTreeClassifier()
BaggingClassifier()
RandomForestClassifier()
ExtraTreesClassifier()
AdaBoostClassifier()
LogisticRegression()
RidgeClassifier()
MultinomialNB()
GaussianNB()

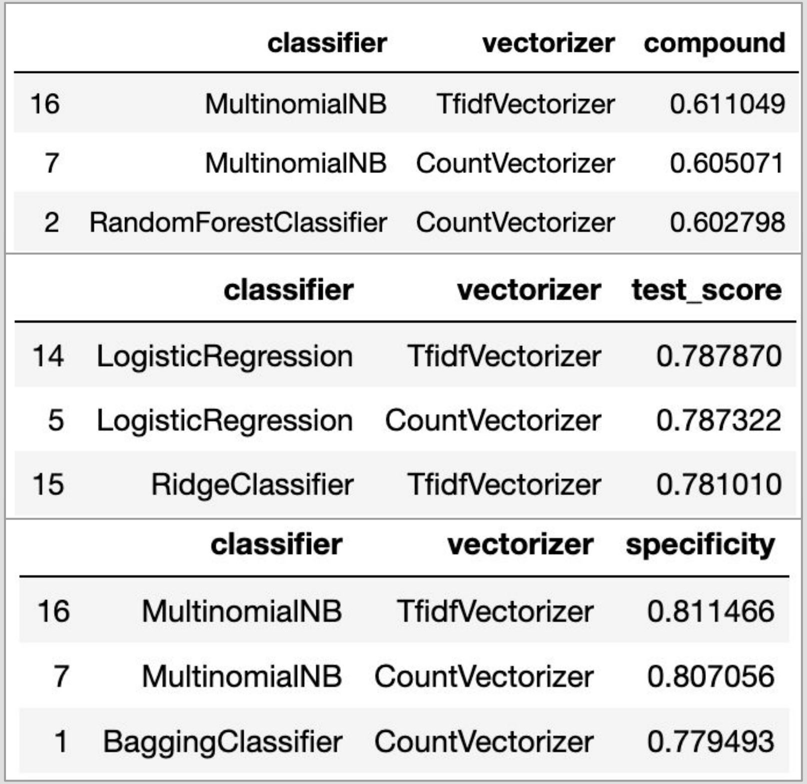
3.6 hrs
1536 + 1 models

2.5 hours to loop through **288 different vectorizers** to find best TfidVectorizer and CountVectorizer hyper parameters.

1.1 hours to loop through **768 Classifier** models for each Vectorizer, totalling **1536 fitted models**

Top Performing Models



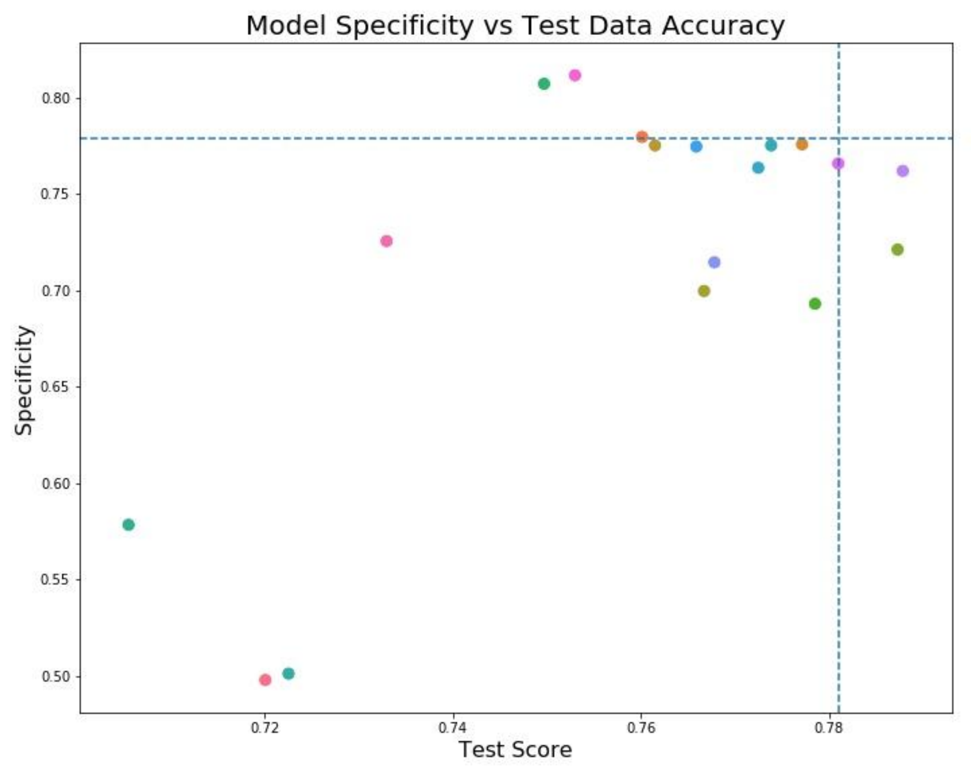


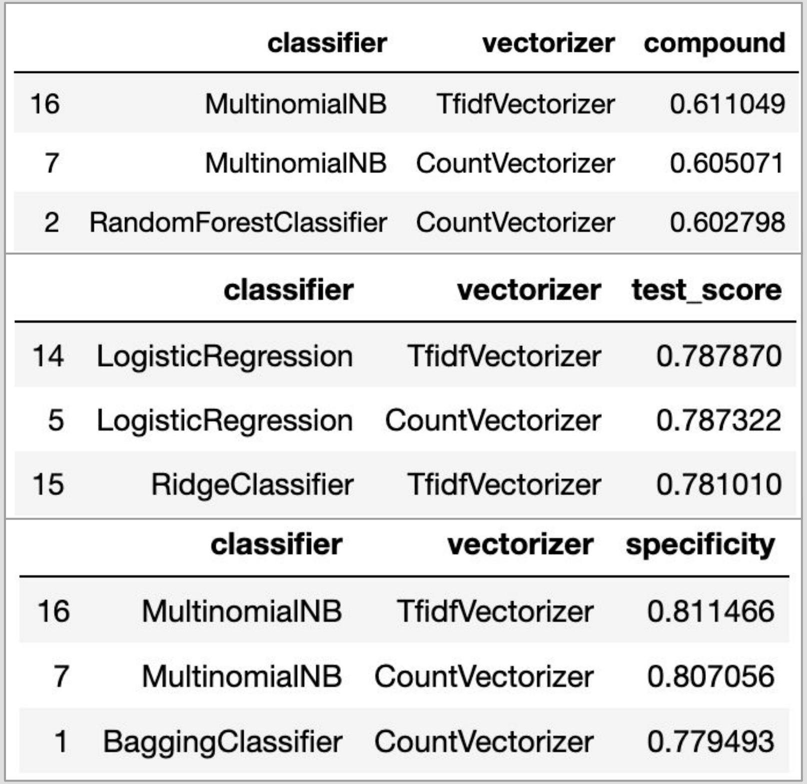
	classifier	vectorizer	compound
16	MultinomialNB	TfidfVectorizer	0.611049
7	MultinomialNB	CountVectorizer	0.605071
2	RandomForestClassifier	CountVectorizer	0.602798

	classifier	vectorizer	test_score
14	LogisticRegression	TfidfVectorizer	0.787870
5	LogisticRegression	CountVectorizer	0.787322
15	RidgeClassifier	TfidfVectorizer	0.781010

	classifier	vectorizer	specificity
16	MultinomialNB	TfidfVectorizer	0.811466
7	MultinomialNB	CountVectorizer	0.807056
1	BaggingClassifier	CountVectorizer	0.779493

Voting Classifier: Selecting the Top Performers



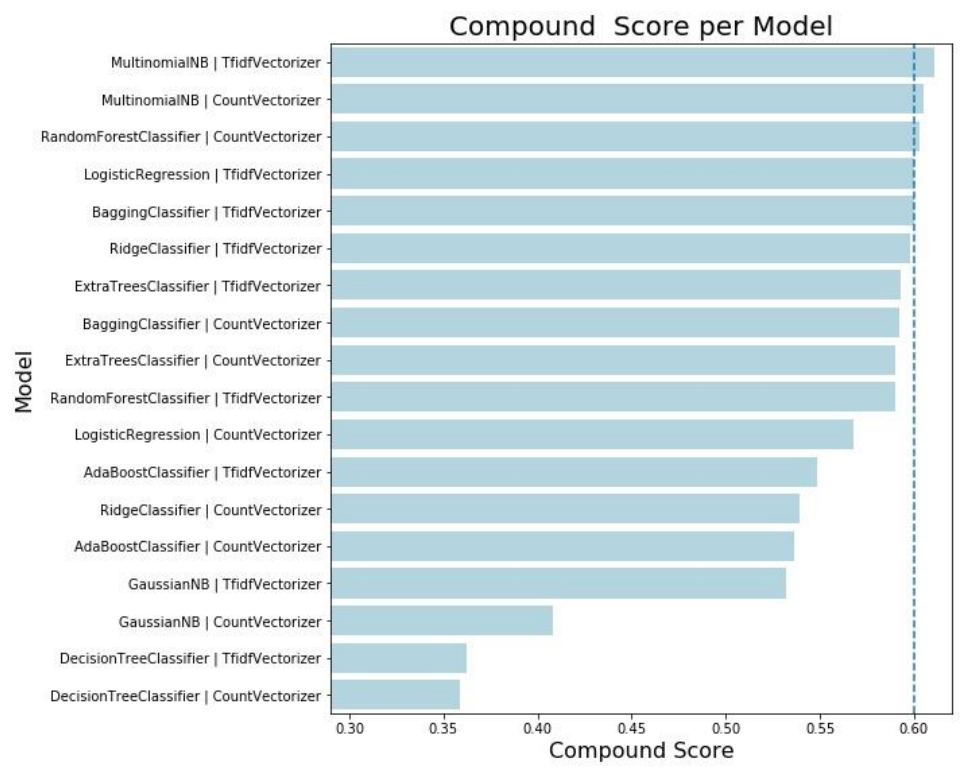


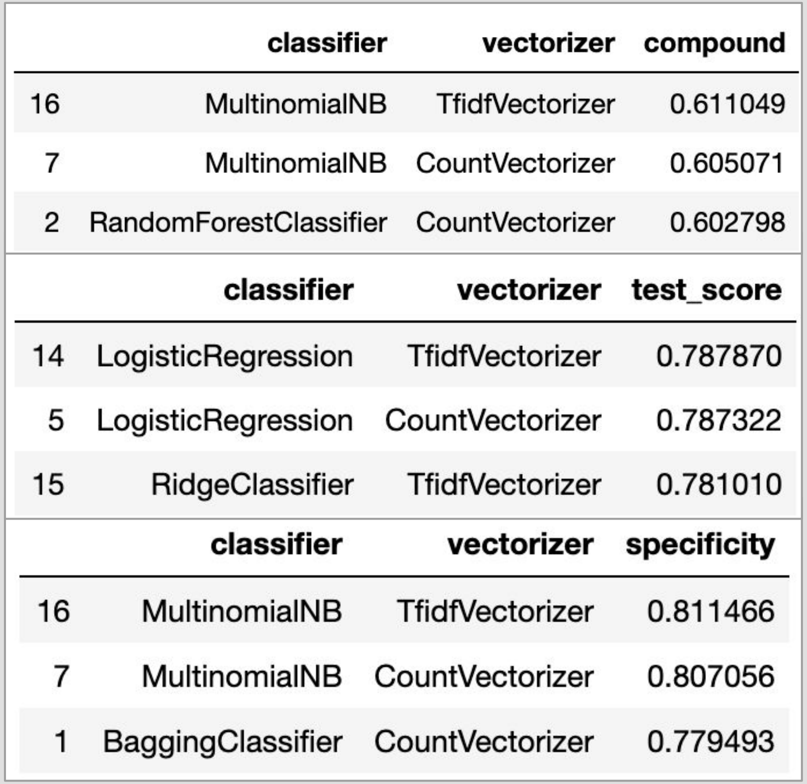
	classifier	vectorizer	compound
16	MultinomialNB	TfidfVectorizer	0.611049
7	MultinomialNB	CountVectorizer	0.605071
2	RandomForestClassifier	CountVectorizer	0.602798

	classifier	vectorizer	test_score
14	LogisticRegression	TfidfVectorizer	0.787870
5	LogisticRegression	CountVectorizer	0.787322
15	RidgeClassifier	TfidfVectorizer	0.781010

	classifier	vectorizer	specificity
16	MultinomialNB	TfidfVectorizer	0.811466
7	MultinomialNB	CountVectorizer	0.807056
1	BaggingClassifier	CountVectorizer	0.779493

Voting Classifier: Selecting the Top Performers





	classifier	vectorizer	compound
16	MultinomialNB	TfidfVectorizer	0.611049
7	MultinomialNB	CountVectorizer	0.605071
2	RandomForestClassifier	CountVectorizer	0.602798

	classifier	vectorizer	test_score
14	LogisticRegression	TfidfVectorizer	0.787870
5	LogisticRegression	CountVectorizer	0.787322
15	RidgeClassifier	TfidfVectorizer	0.781010

	classifier	vectorizer	specificity
16	MultinomialNB	TfidfVectorizer	0.811466
7	MultinomialNB	CountVectorizer	0.807056
1	BaggingClassifier	CountVectorizer	0.779493

Model Evaluation and Comparison

Voting Classifier

- Training Score: 83.3%
- Test Score: 78.3%
- Specificity: 78.0%
- Compound: 0.611
- Improvement: 55.19% from baseline

MultinomialNB with TFIDF

- Training Score: 78.9%
- Test Score: 75.3%
- Specificity: 81.1%
- Compound: 0.611
- Improvement: 49.82% from baseline

MultinomialNB and TFID:

Specificity: 81.1%

Accuracy: 75.3%

Compound Score: 0.611

Accuracy Improvement: 49.82% from baseline of 50.26%

Further Steps and Recommendations

Though our model does not have the highest accuracy, it does have the **highest specificity** out of our 1564 models.

Still, at 81.1% this means ~2 out of 10 vegan users are still being labeled as vegetarian.

Recommendation:

- We continue tweaking hyperparameters, including increasing the threshold for a user to be classified as a Vegan. This would result in a higher ***specificity score***.
- However, this would ***decrease accuracy*** and result in more Vegetarians being classified as Vegan.

Further Steps:

- We would need to do a cost-benefit analysis on the capital loss from reducing our egg-and-dairy advertisement to misidentified Vegetarians against the profit loss of potentially losing a Vegan customer from out site completely.

thank you.