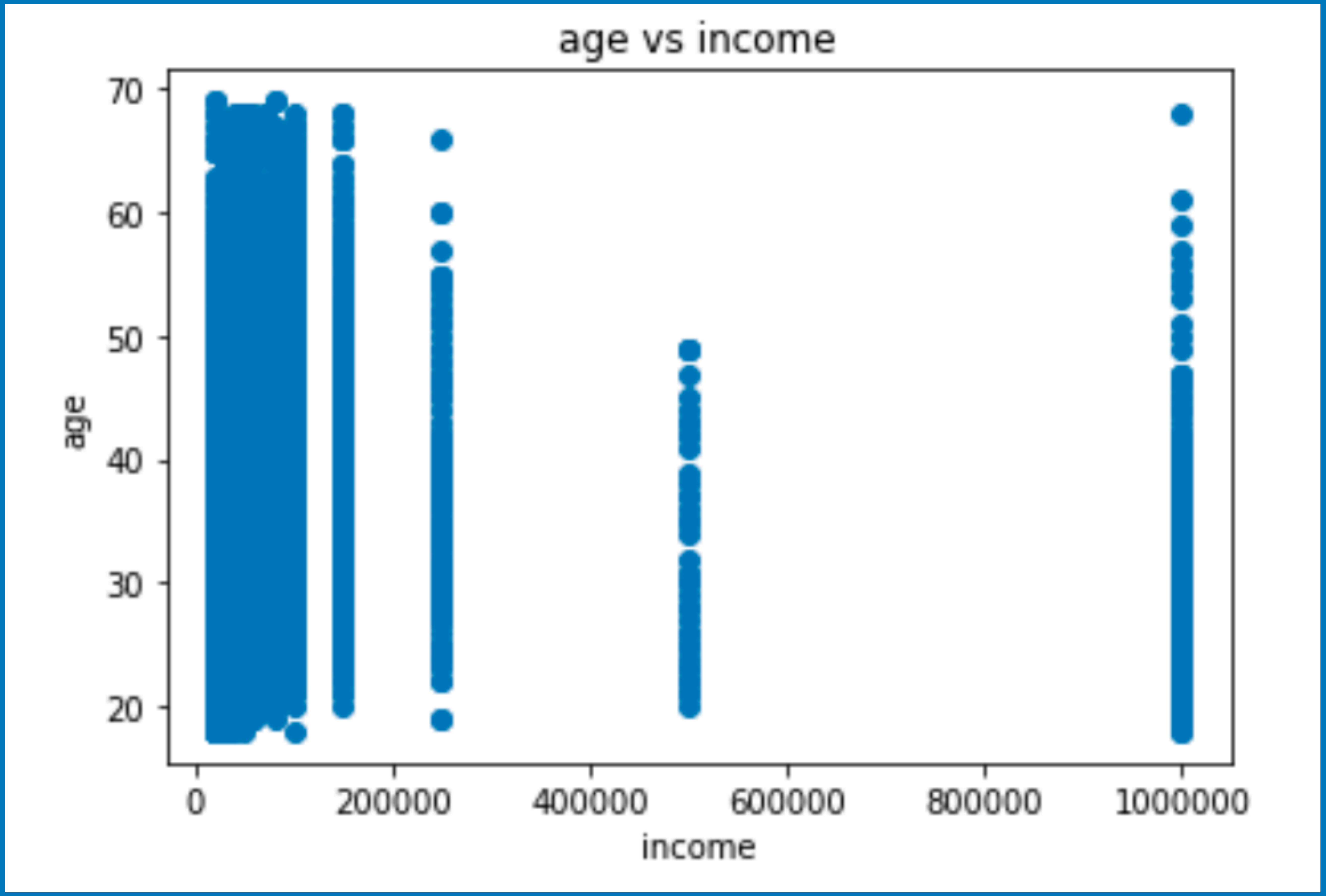
OKCupid Data

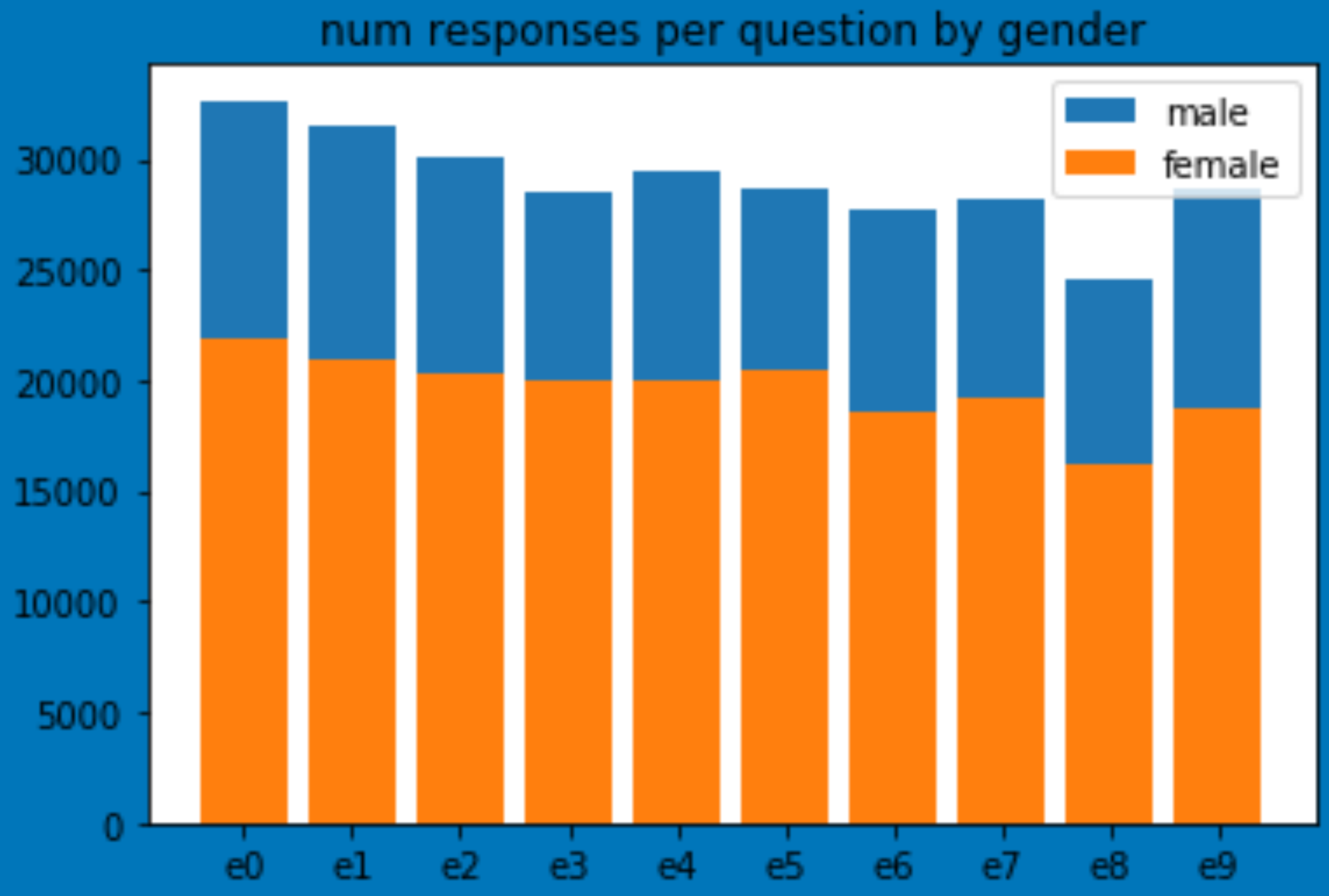
Machine Learning Fundamentals
Richard Lewis
11/10/2018

Exploration of the Dataset



I thought that I would explore the relationship between age and income but after plotting them, it doesn't look like there's a clear relationship between the two… plus, the fact that it's a self-reported category makes me think that some respondents might not be telling the truth.

Exploration of the Dataset



This chart show the number of responses per essay broken down by gender. Women and men answered all essays at about the same rate. Question 8 was not especially popular with either gender.

Exploration of the Dataset

Of the 9 essays, essay3: The first thing people usually notice about me, is the only one that asks the respondent to describe what they think that *other* people think about them. I think that this will be the question that provides the highest indication of gender.

Question to Answer

Can essay3 (the first thing people usually notice about me) be used to predict the gender of the respondent?

Of the 9 essays, essay3 is the only one that asks the respondent to describe what they think that *other* people think about them. I think that this will be the question that provides the highest indication of gender.

Classification Approaches: NB

My first approach is to use a Naives Bayes classifier on essay3. I took the following steps to prepare the data:

- Separate the essays into male and female responses to make sure that there are close to the same number of essays in each group. There are 28,467 male essays and 20,003 female essays
- Drop empty rows because empty rows tell us nothing in this case
- Replace “\n” and “<br />” with spaces and put essays back into one group
- Split the data into training and test groups (with train and test labels)
- Create vectors and the classifier
- Get accuracy, precision, and recall metrics

Classification Approaches: NB

Confusion matrix:

```
[[4017 1702]
 [1641 2334]]
```

n=9694	Predicted: 0	Predicted: 1
Actual: 0	4017	1702
Actual: 1	1641	2334

Accuracy: 0.655148

Precision: 0.578295

Recall: 0.587170

These numbers are all pretty low. The accuracy is barely higher than a coin toss.

Augmenting the Dataset

Creating columns for the top 100 words (part 1)

```python
male_word_freq = get_word_frequency(male_essay3_text)
female_word_freq = get_word_frequency(female_essay3_text)

# get top 100 words for both sets of essays
top_male_words = male_word_freq[0:100]['word'].tolist()
top_female_words = female_word_freq[0:100]['word'].tolist()

# pluck out the 100 unique ones between the lists
top_words = []
[top_words.append(x) for x in (top_male_words + top_female_words) if x not in top_words]
top_words = top_words[0:100]
```

First, I created a function called get_word_frequency which, given a list of strings returns a data frame containing usage frequencies for each word in the list of strings sorted in descending frequency order. I call this function on the essays by male and female respondents. Then, I take the 100 most frequently used words from each set of essays. Last, I combine the lists and remove duplicates.

Augmenting the Dataset

Creating columns for the top 100 words (part 2)

```
# create a new column in the essay3_word_counts dataframe
for word in top_words:
    essay3_word_counts[word+'_count'] = essay3_word_counts.essay3.apply(
        lambda x: x.split().count(word))
```

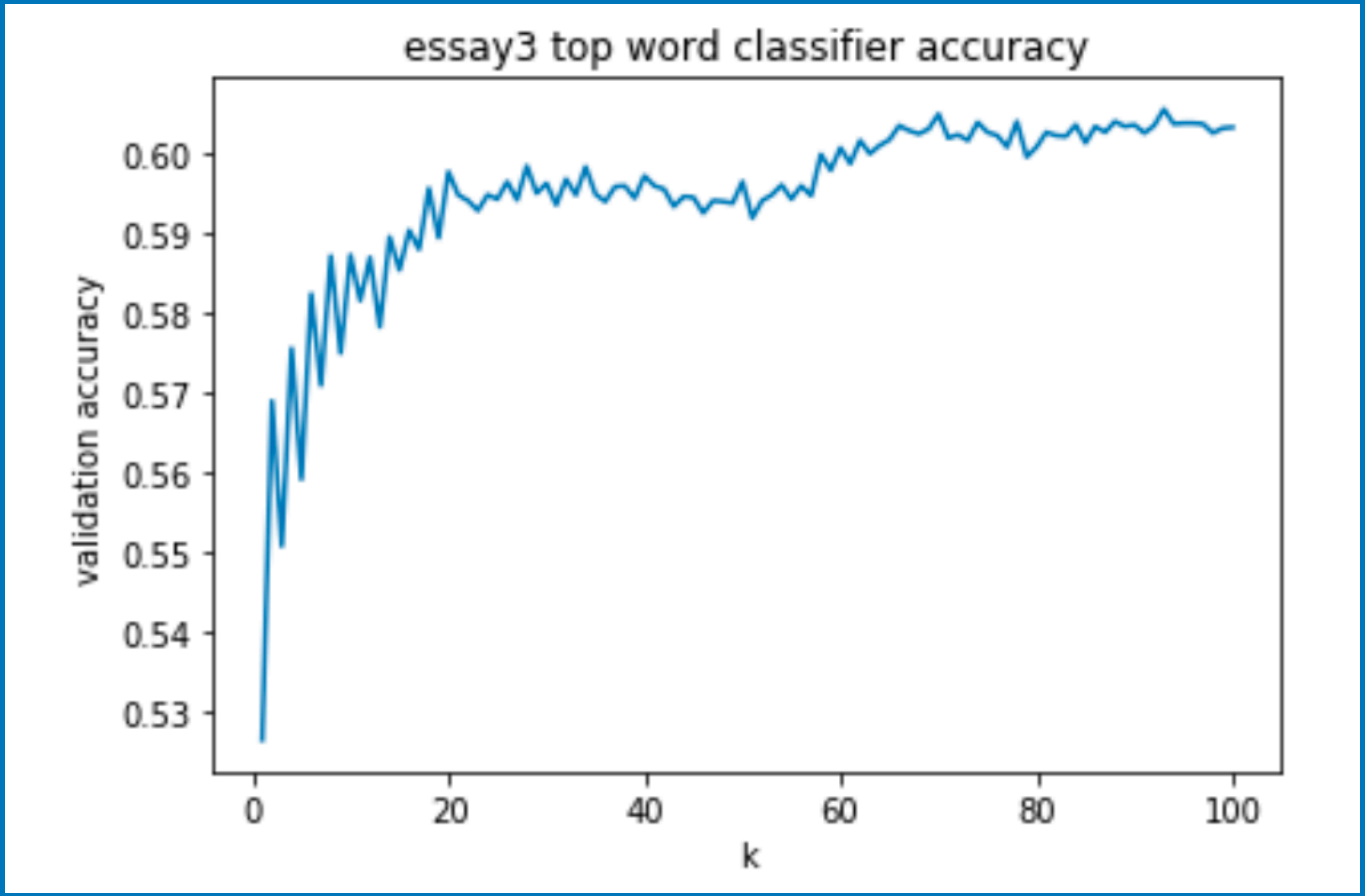
Now that I've got the 100 most frequently used words across both male and female essays, I need to create columns for each. Using a for loop, I create a column for each word in a new data frame (essay3_word_counts). Each row in each word's column contains the number of times that that word appears in each essay in my dataset.

Classification Approaches: KNN

Since the accuracy, precision and recall were so low on the Naive Bayes approach, I tried a KNeighborsClassifier:

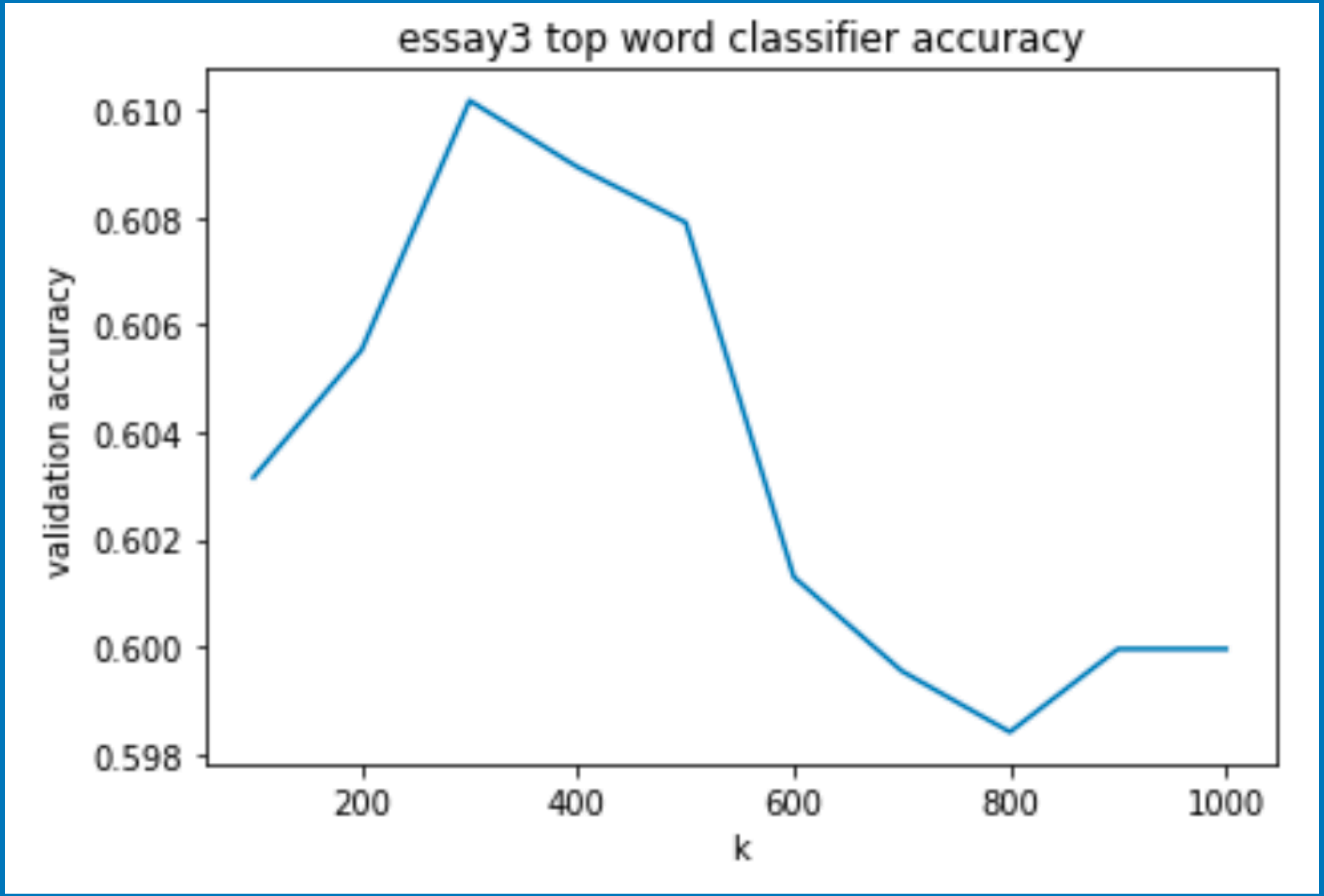
- Get the 100 most frequently used words in both the male and female essays
- Get the 100 unique words in these two groups
- Create a column for each unique word and a list of labels for each essay
- Convert dataframe and labels to lists and use KNeighborsClassifier
- Get accuracy, precision, and recall metrics

Classification Approaches: KNN



I tried values of K from 1 to 100 and got the highest accuracy (0.603) at around 100.

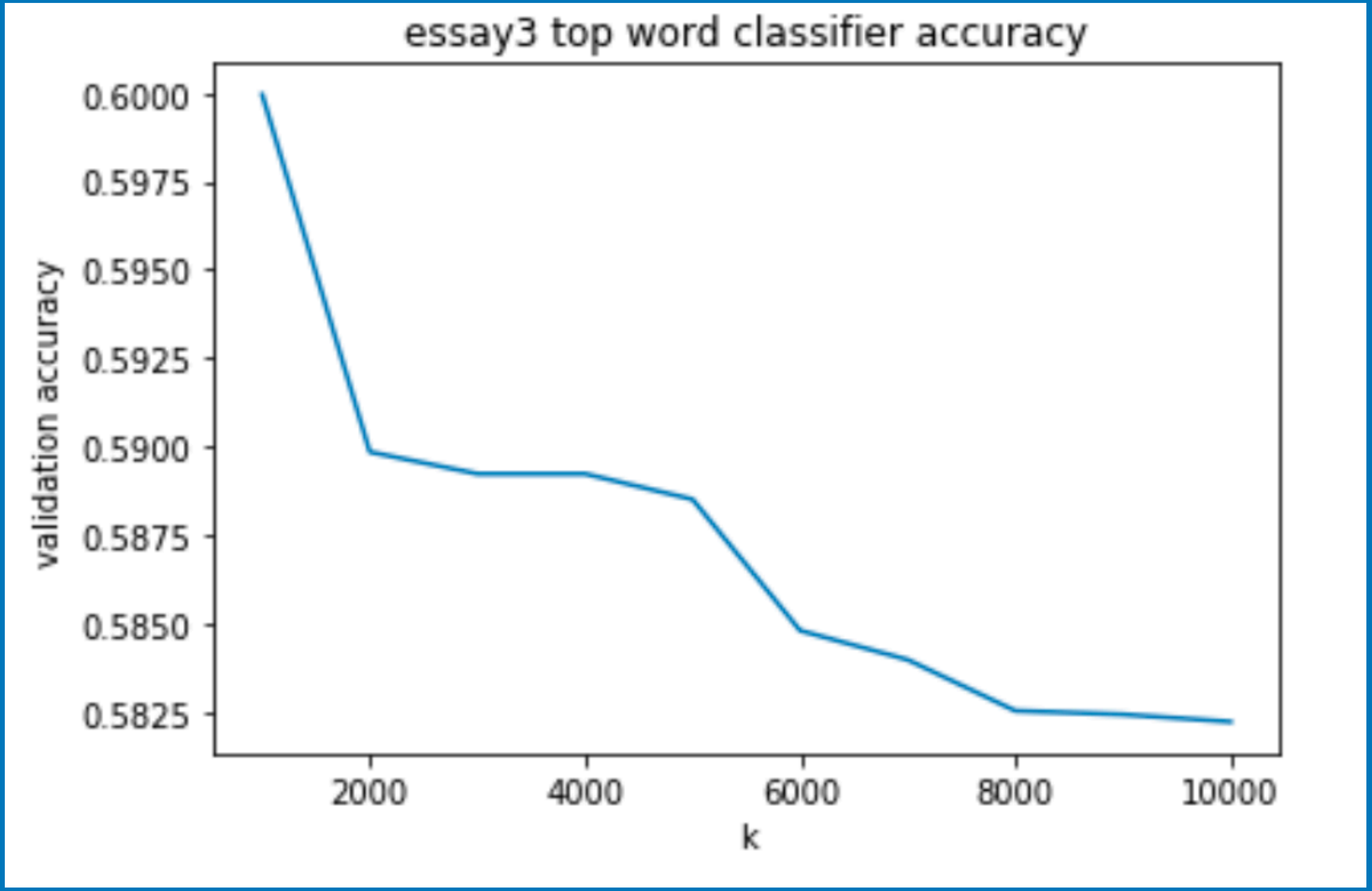
Classification Approaches: KNN



Because accuracy was highest near to the highest K value that I initially checked, I wanted to see if accuracy would continue to increase at even higher K values.

I checked accuracy for 100 K values from 100 to 1,000. The highest accuracy (0.610) was found when K was 300.

Classification Approaches: KNN



Just for kicks, I tried every thousand K values from 1,000 to 10,000. Accuracy steadily decreased.

Classification Approaches: KNN

Confusion matrix:

```
[[5127 516]
 [3263 788]]
```

Accuracy: 0.610171

Precision: 0.61

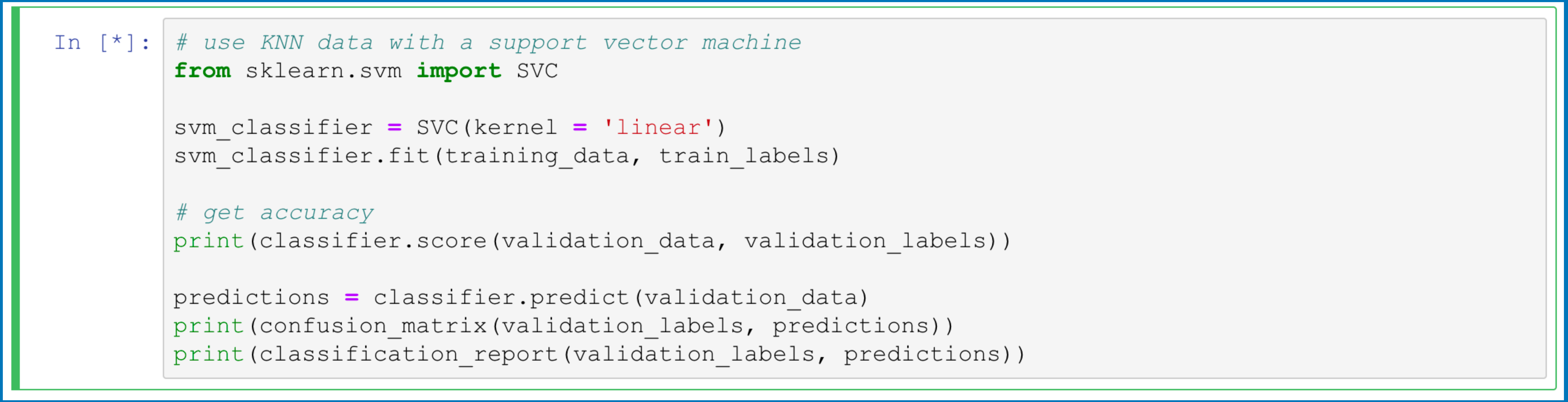
Recall: 0.61

n=9699	Predicted: 0	Predicted: 1
Actual: 0	5127	516
Actual: 1	3268	788

These numbers are all pretty low. The accuracy is barely higher than a coin toss.

From a speed perspective, the algorithm ran fairly quickly.

Regression Approaches: SVM



```
In [*]: # use KNN data with a support vector machine
        from sklearn.svm import SVC

        svm_classifier = SVC(kernel = 'linear')
        svm_classifier.fit(training_data, train_labels)

        # get accuracy
        print(classifier.score(validation_data, validation_labels))

        predictions = classifier.predict(validation_data)
        print(confusion_matrix(validation_labels, predictions))
        print(classification_report(validation_labels, predictions))
```

I attempted to use scikitlearn's SVM but my Jupyter notebook hung for about 20 minutes, after which I killed the kernel and moved on.

Regression Approaches: KNN

I used the same dataset from the KNN Classifier and got an accuracy score of -0.019285010143822667. I was unable to produce a confusion matrix.

Conclusions/Next steps

I thought that there would be strong correlation between language usage and gender. It turns out that with these algorithms and this dataset, I was unable to find such a correlation.

Further exploration on this question might involve removing stop words from my list of 100 most used words. Additionally, it might make sense to try to find a correlation between words used and education or income level