

Deep NLP Classification

Daniel Alvarado

What is the situation?

- Chatbot prompt is 'Describe a time when you have acted as a resource for someone else'

- 80 responses

- If a response is 'not flagged', the user can continue talking to the bot. If it is 'flagged', the user is referred to help

- 125 resumes were queried from Indeed.com with keyword 'data scientist', location 'Vermont'

- If a resume is 'not flagged', the applicant can submit a modified resume at a later date. If it is 'flagged', the applicant is invited to interview.

Who cares?

- For the chatbot responses; mental health professionals, user of the chatbot, family of the user
- For the resumes; employer, potential employee

The Data

- Downloaded from <https://www.kaggle.com/samdeeplearning/deepnlp>
- This data is used to identify the variables behind flagged therapy chatbot responses and flagged resume submissions
- Flagged chatbot responses result in referring the user to professional mental health help and flagged resumes are invited to interview.

EDA

- Word clouds were created for both datasets
- Top 50 most used words
- Most common 5-word phrases
- Least common words
- Average word length and average sentence length analysis
- I will only upload a few images for the sake of presentation length

Word Clouds



The most popular words in chatbot responses are 'friend', 'people', and 'helped'.

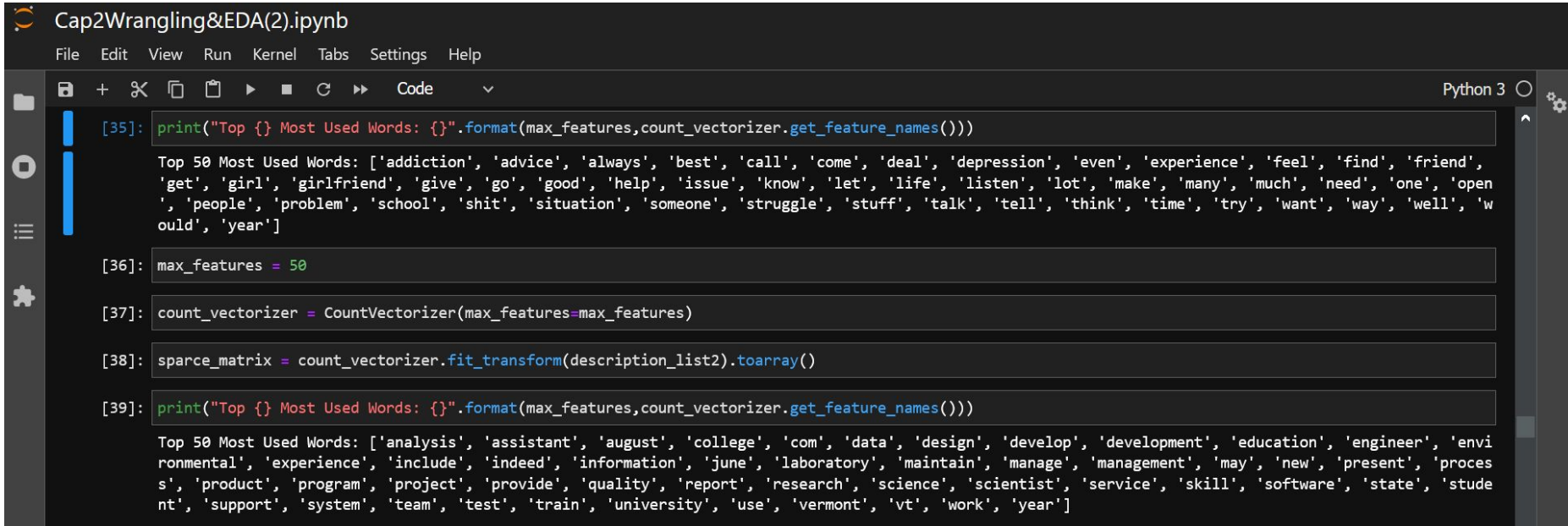
```
In [10]: # Bag of resume words

wordcloud(data2['resume text'])
```



The most popular words in the data science resumes are 'research', 'data', and 'development'.

Top 50 Most Used Words



```
Cap2Wrangling&EDA(2).ipynb
File Edit View Run Kernel Tabs Settings Help

[35]: print("Top {} Most Used Words: {}".format(max_features, count_vectorizer.get_feature_names()))

Top 50 Most Used Words: ['addiction', 'advice', 'always', 'best', 'call', 'come', 'deal', 'depression', 'even', 'experience', 'feel', 'find', 'friend',
'get', 'girl', 'girlfriend', 'give', 'go', 'good', 'help', 'issue', 'know', 'let', 'life', 'listen', 'lot', 'make', 'many', 'much', 'need', 'one', 'open',
', 'people', 'problem', 'school', 'shit', 'situation', 'someone', 'struggle', 'stuff', 'talk', 'tell', 'think', 'time', 'try', 'want', 'way', 'well', 'w
ould', 'year']

[36]: max_features = 50

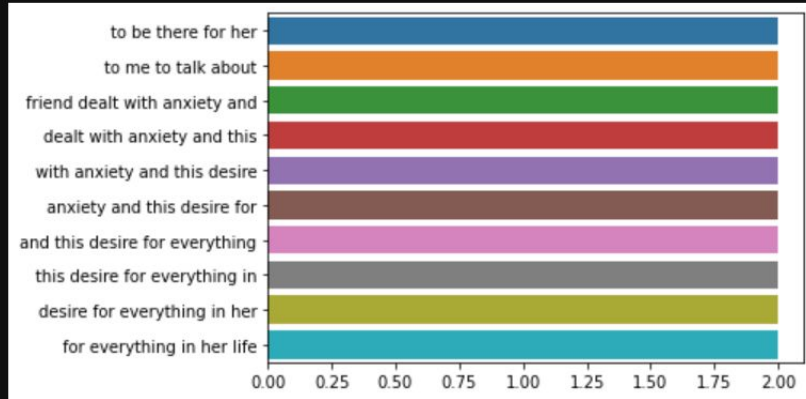
[37]: count_vectorizer = CountVectorizer(max_features=max_features)

[38]: sparse_matrix = count_vectorizer.fit_transform(description_list2).toarray()

[39]: print("Top {} Most Used Words: {}".format(max_features, count_vectorizer.get_feature_names()))

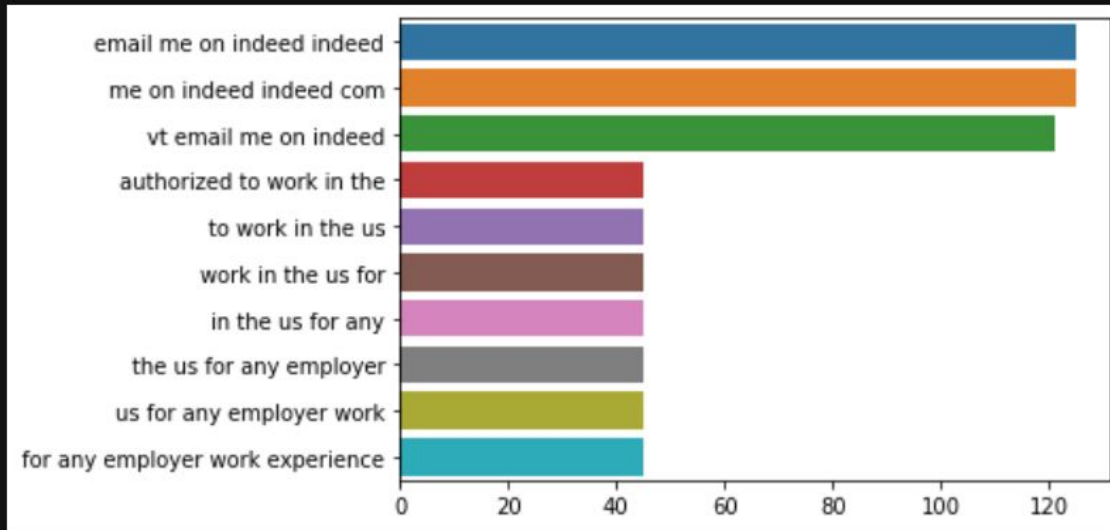
Top 50 Most Used Words: ['analysis', 'assistant', 'august', 'college', 'com', 'data', 'design', 'develop', 'development', 'education', 'engineer', 'envi
ronmental', 'experience', 'include', 'indeed', 'information', 'june', 'laboratory', 'maintain', 'manage', 'management', 'may', 'new', 'present', 'proces
s', 'product', 'program', 'project', 'provide', 'quality', 'report', 'research', 'science', 'scientist', 'service', 'skill', 'software', 'state', 'stude
nt', 'support', 'system', 'team', 'test', 'train', 'university', 'use', 'vermont', 'vt', 'work', 'year']
```

Most Common 5-word Phrases in Chatbot Responses



The pentagrams for the chatbot responses paints a clear picture into how people responded to the prompt. Being there for someone and dealing with anxiety are the main themes.

Most Common 5-word Phrases in Resumes



The resume pentagrams are identical. 5 word phrases in resumes pertain to working and invitations to email via indeed.com

Modeling and Methods

- Supervised learning problem
- Binary classification: 0 = not flagged, 1 = flagged
- Class label had to be converted to binary values

Models

- Accuracy and precision were the metrics to measure for both the classification models
- Multinomial and Gaussian Naive Bayes
- Random Forest
- Support Vector Classifier
- Tensorflow deep learning model

Modeling Steps

- Minimal to light preprocessing required, data was well written and abundant
- Used sklearn's CountVectorizer for preprocessing
- Train/Test split of 75/25

Results

Multinomial Naive Bayes

```
In [8]: x = data.response_text
y = data['class']
x_train,x_test,y_train,y_test = train_test_split(x,y, test_size = 0.25, random_state=42)
x_train_dtm = vect.fit_transform(x_train)
x_test_dtm = vect.transform(x_test)
NB.fit(x_train_dtm,y_train)
y_predict = NB.predict(x_test_dtm)
metrics.accuracy_score(y_test,y_predict)
```

Out[8]: 0.65

```
In [9]: metrics.precision_score(y_test,y_predict)
```

Out[9]: 0.3333333333333333

```
In [10]: metrics.f1_score(y_test,y_predict)
```

Out[10]: 0.46153846153846156

Random Forest

```
In [11]: rf = RandomForestClassifier(max_depth=10,max_features=10)
rf.fit(x_train_dtm,y_train)
rf_predict = rf.predict(x_test_dtm)
metrics.accuracy_score(y_test,rf_predict)
```

Out[11]: 0.75

Results

Multinomial Naive Bayes for Resumes

```
In [12]: x = data2.resume_text
y = data2['class']
x_train,x_test,y_train,y_test = train_test_split(x,y, test_size = 0.25, random_state=42)
x_train_dtm = vect.fit_transform(x_train)
x_test_dtm = vect.transform(x_test)
NB.fit(x_train_dtm,y_train)
y_predict = NB.predict(x_test_dtm)
metrics.accuracy_score(y_test,y_predict)
```

```
Out[12]: 0.71875
```

```
In [13]: metrics.precision_score(y_test,y_predict)
```

```
Out[13]: 0.75
```

```
In [14]: metrics.f1_score(y_test,y_predict)
```

```
Out[14]: 0.39999999999999997
```

Random Forest for Resumes

```
In [15]: rf = RandomForestClassifier(max_depth=10,max_features=10)
rf.fit(x_train_dtm,y_train)
rf_predict = rf.predict(x_test_dtm)
metrics.accuracy_score(y_test,rf_predict)
```

```
Out[15]: 0.65625
```

The precision and f1_scores for the random forest models was 0, so these metrics will be ignored.

Results

- The random forest classifier gave the highest accuracy score for the chatbot responses at 0.75, 0.10 more than the multinomial naive bayes classifier.
- For the resumes, the multinomial naive bayes classifier gave a slightly higher accuracy score than the random forest. 0.72 for multinomial naive bayes and 0.65 for the random forest classifier.

Results

- The f1 score and precision score of the chatbot responses are 0.46 and 0.33, respectively for multinomialNB. For the resumes, the multinomialNB f1 score and precision scores are 0.4 and 0.75, respectively.
- Precision can be thought of as the fraction of positive predictions that actually belong to the positive class. The multinomialNB classifier on the resumes yield a high precision score, close to its accuracy score.
- This model is what we're looking for in a resume classifier, a predictor that has a low false positive rate!

Deep learning using TensorFlow and Keras

```
Epoch 3/15
6/6 [=====] - 0s 3ms/step - loss: 0.6762 - accuracy: 0.7212
Epoch 4/15
6/6 [=====] - 0s 3ms/step - loss: 0.6699 - accuracy: 0.6988
Epoch 5/15
6/6 [=====] - 0s 3ms/step - loss: 0.6608 - accuracy: 0.7013
Epoch 6/15
6/6 [=====] - 0s 3ms/step - loss: 0.6481 - accuracy: 0.7280
Epoch 7/15
6/6 [=====] - 0s 3ms/step - loss: 0.6436 - accuracy: 0.7100
Epoch 8/15
6/6 [=====] - 0s 3ms/step - loss: 0.6355 - accuracy: 0.7079
Epoch 9/15
6/6 [=====] - 0s 3ms/step - loss: 0.6329 - accuracy: 0.6981
Epoch 10/15
6/6 [=====] - 0s 3ms/step - loss: 0.6190 - accuracy: 0.7152
Epoch 11/15
6/6 [=====] - 0s 3ms/step - loss: 0.6060 - accuracy: 0.7294
Epoch 12/15
6/6 [=====] - 0s 3ms/step - loss: 0.6048 - accuracy: 0.7148
Epoch 13/15
6/6 [=====] - 0s 3ms/step - loss: 0.6136 - accuracy: 0.6831
Epoch 14/15
6/6 [=====] - 0s 3ms/step - loss: 0.5826 - accuracy: 0.7370
Epoch 15/15
6/6 [=====] - 0s 3ms/step - loss: 0.5828 - accuracy: 0.7171
```

-An epoch is a full pass through the entire training data set.

-The neural network saw each unique sample 15 times in this case.

-The in-sample predictions all had an accuracy around 0.70 +/- 0.03

--The loss function is a type of error function, used to update the weights between the inputs and the 'neurons' to reduce loss on each successive evaluation

-The loss function used was 'binary_crossentropy', used for binary classification

Deep learning using TensorFlow and Keras

It's worth mentioning that neural networks are great at fitting. Did this model overfit? Ideally, the model would generalize and learn patterns and attributes of the original text to determine a flag or not.

```
val_loss, val_acc = model.evaluate(testing_padded, testing_labels_final)
print(val_loss, val_acc)
```

```
2/2 [=====] - 0s 6ms/step - loss: 0.5856 - accuracy: 0.7317
0.585637629032135 0.7317073345184326
```

-The out-of-sample validation has a slightly higher loss than the last epoch the model ran through. The accuracy is also slightly higher.

-A high delta in either category implies that there is overfitting within the model. I chose a train/test split of 80/20 because it gave me both in-sample and out-of-sample scores that were similar.

Summary/Conclusion

- The accuracy score for both the GaussianNB and Random Forest classifiers on the chatbot responses are 0.75.
- The GaussianNB classifier performs better on the chatbot response data than the MultinomialNB classifier.
- For the resumes, the MultinomialNB f1 score, precision scores, and accuracy scores are 0.4, 0.75, and 0.72 respectively. This would be the best model to use on resume scores.
- The out-of-sample deep learning validation yielded an accuracy of 0.73 on a combined (resume + responses) data set.
- The validation of the loss function and accuracy score on the TF-Keras model was consistent with the other 15 epochs.
- The model did not overfit on the training data and successfully generalized and learned the patterns of flagged inputs.

Further Research

Comparing different chatbot prompt responses would give more insight into what responses are flagged or not.

-Granted, these prompts would have to be similar, in the same vein as the original. An example prompt could be 'Explain a situation where you provided comfort to someone.'

-For the resumes, including other jobs in the tech industry would yield a more comprehensive assessment of what employers look for in a resume when hiring.

-It would also be interesting to analyze data science resumes over a broader region. This may yield insights into what companies across this hypothetical region require in a data scientist.