GENERAL ASSEMBLY

COMMENT TOXICITY CLASSIFICATION

By Nikit Sharma

Staging the Question

Problem…

“Discussing things you care about can be difficult. The threat of abuse and harassment online means that many people stop expressing themselves and give up on seeking different opinions. Platforms struggle to effectively facilitate conversations, leading many communities to limit or completely shut down user comments.” – Kaggle.com

Hypothesis…

Multi-label classification: Using Wikipedia comments dataset (from https://www.kaggle.com/c/jigsaw-toxic-comment-classification-challenge) , labeled by human raters for **toxic*** behavior, predict the probability of each type of **toxicity*** for each type of comment.

* comments that are rude, disrespectful or otherwise likely to make someone leave a discussion)

Previous Research

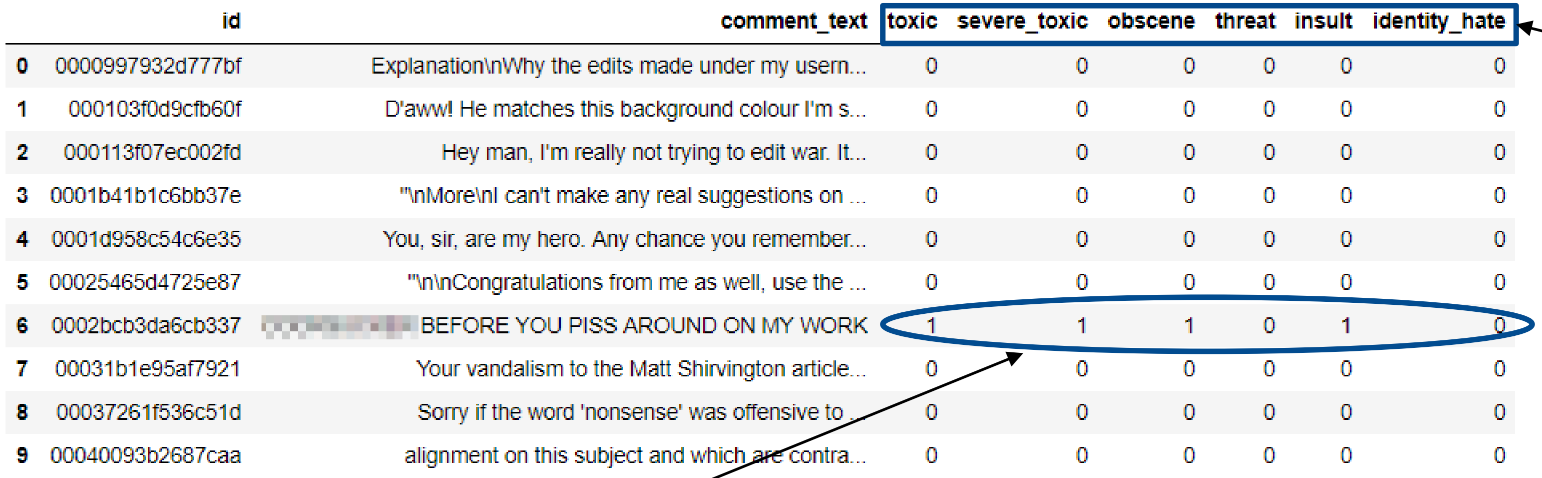
- Studies ongoing at **“The Conversation AI team”**, a research initiative founded by **Jigsaw** and **Google** (both a part of **Alphabet**) to build tools to help improve online conversation study of negative online behaviors, like toxic comments.

- Tools and models publicly made available through **Perspective API**.

- Current models still make errors, and don’t allow users to select which types of toxicity they’re interested in finding (e.g. some platforms may be fine with profanity, but not with other types of toxic content)

Problem Context

- Multi-label classification problem not multi-class classification i.e. an observation could belong to more than one class at the same time.

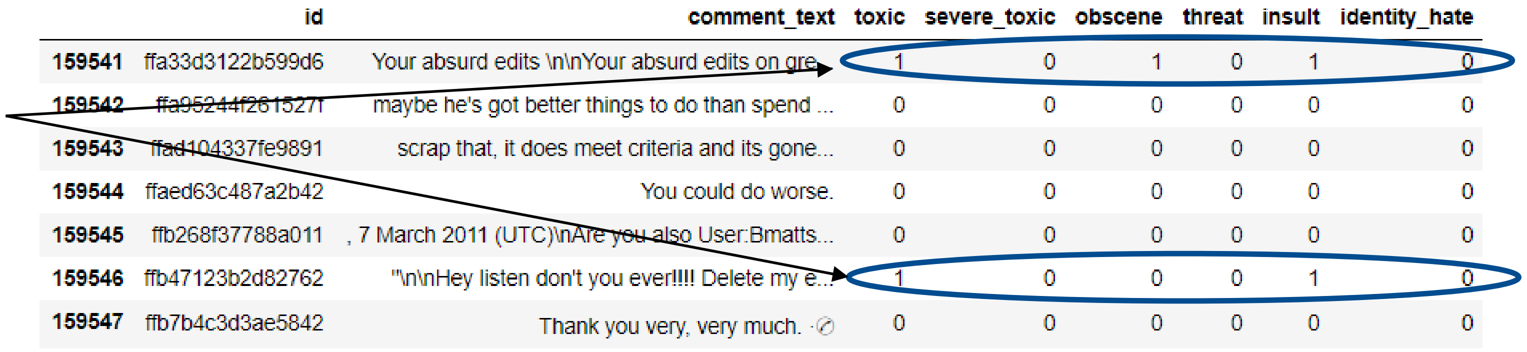


	id	comment_text	toxic	severe_toxic	obscene	threat	insult	identity_hate
0	0000997932d777bf	Explanation\nWhy the edits made under my usern...	0	0	0	0	0	0
1	000103f0d9cfb60f	D'aww! He matches this background colour I'm s...	0	0	0	0	0	0
2	000113f07ec002fd	Hey man, I'm really not trying to edit war. It...	0	0	0	0	0	0
3	0001b41b1c6bb37e	"\nMore\nI can't make any real suggestions on ...	0	0	0	0	0	0
4	0001d958c54c6e35	You, sir, are my hero. Any chance you remember...	0	0	0	0	0	0
5	00025465d4725e87	"\n\nCongratulations from me as well, use the ...	0	0	0	0	0	0
6	0002bcb3da6cb337	[redacted] BEFORE YOU PISS AROUND ON MY WORK	1	1	1	0	1	0
7	00031b1e95af7921	Your vandalism to the Matt Shirvington article...	0	0	0	0	0	0
8	00037261f536c51d	Sorry if the word 'nonsense' was offensive to ...	0	0	0	0	0	0
9	00040093b2687caa	alignment on this subject and which are contra...	0	0	0	0	0	0

A comment can belong to one (or more) of 6 toxicity classes -

- Toxic
- Severe Toxic
- Obscene
- Threat
- Insult
- Identity Hate

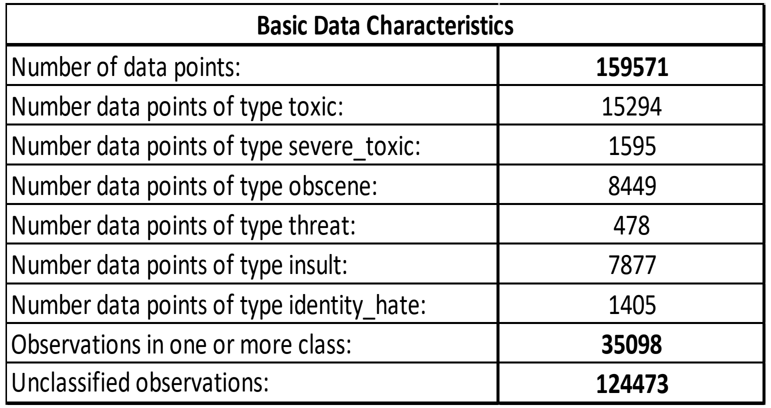
Instances of comments belonging to multiple classes at once.



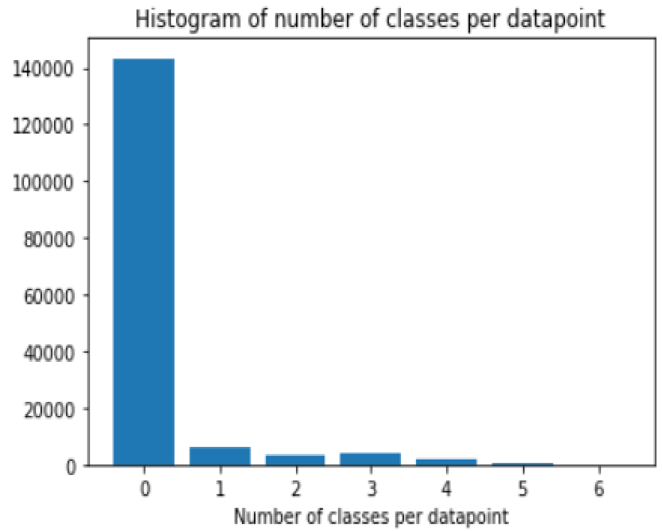
	id	comment_text	toxic	severe_toxic	obscene	threat	insult	identity_hate
159541	ffa33d3122b599d6	Your absurd edits \n\nYour absurd edits on gre...	1	0	1	0	1	0
159542	ffa95244f261527f	maybe he's got better things to do than spend ...	0	0	0	0	0	0
159543	ffad104337fe9891	scrap that, it does meet criteria and its gone...	0	0	0	0	0	0
159544	ffaed63c487a2b42	You could do worse.	0	0	0	0	0	0
159545	ffb268f37788a011	, 7 March 2011 (UTC)\nAre you also User:Bmatts...	0	0	0	0	0	0
159546	ffb47123b2d82762	"\n\nHey listen don't you ever!!!! Delete my e...	1	0	0	0	1	0
159547	ffb7b4c3d3ae5842	Thank you very, very much.	0	0	0	0	0	0

Exploratory Data Analysis

- ~160k observations in total
- ~125k with zero toxicity of any type
- ~35k classified in one or more toxicity categories

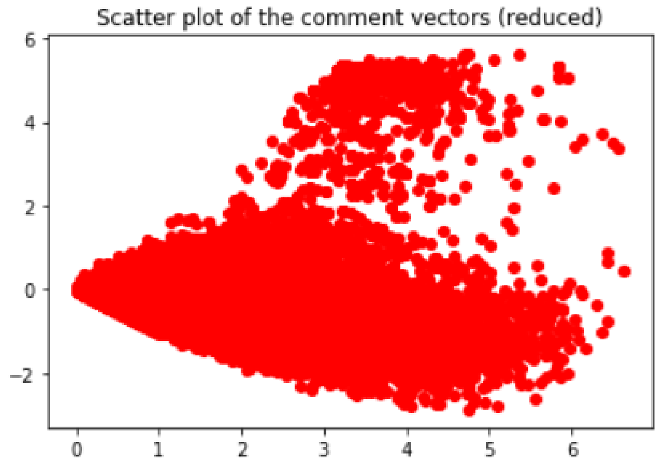


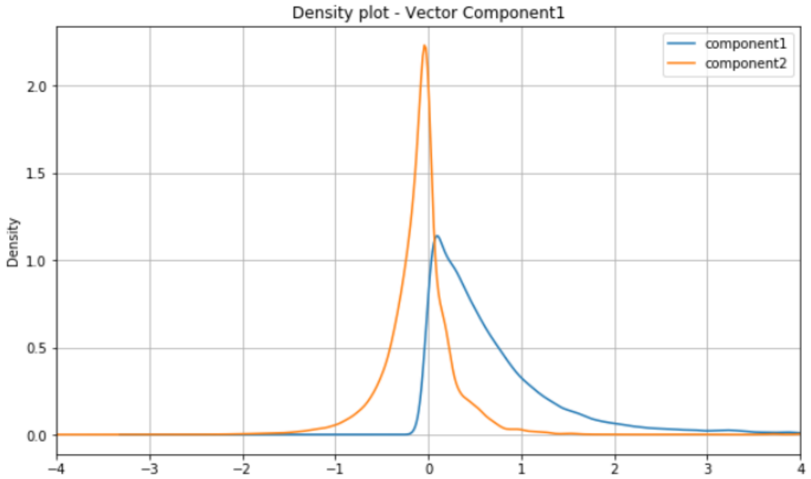
Basic Data Characteristics	
Number of data points:	**159571**
Number data points of type toxic:	15294
Number data points of type severe_toxic:	1595
Number data points of type obscene:	8449
Number data points of type threat:	478
Number data points of type insult:	7877
Number data points of type identity_hate:	1405
Observations in one or more class:	**35098**
Unclassified observations:	**124473**



- reduced *(2 components* only)* scatter plot post embedding *(count vectorization of full dataset)*
- skewed distribution suggests imbalanced classes

- density graph (on far right) shows the vector projections are a rough approximation of a normal distribution with some overlap between the 2 components

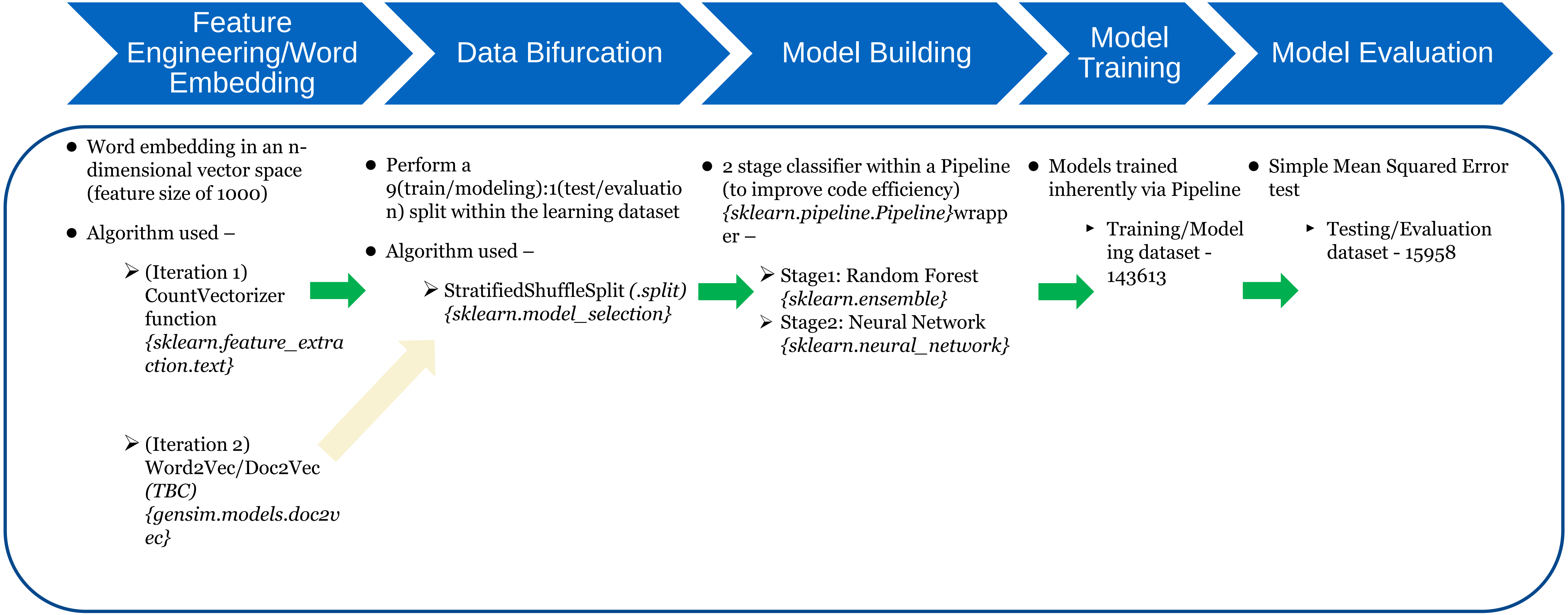




* each *component* is a linear combination of word probabilities from the comment/observation corpus with vector space probabilities assigned to each
* each *comment* is a linear combination of components

Approach

The objective of building the model was to optimize the predicted probability of each observation falling into each of the 6 classes



Feature Engineering/Word Embedding

- Word embedding in an n-dimensional vector space (feature size of 1000)
- Algorithm used –
  - (Iteration 1) CountVectorizer function *{sklearn.feature_extraction.text}*
  - (Iteration 2) Word2Vec/Doc2Vec *(TBC)* *{gensim.models.doc2vec}*

Data Bifurcation

- Perform a 9(train/modeling):1(test/evaluation) split within the learning dataset
- Algorithm used –
  - StratifiedShuffleSplit *(.split)* *{sklearn.model_selection}*

Model Building

- 2 stage classifier within a Pipeline (to improve code efficiency) *{sklearn.pipeline.Pipeline}*wrapper –
  - Stage1: Random Forest *{sklearn.ensemble}*
  - Stage2: Neural Network *{sklearn.neural_network}*

Model Training

- Models trained inherently via Pipeline
  - Training/Modeling dataset - 143613

Model Evaluation

- Simple Mean Squared Error test
  - Testing/Evaluation dataset - 15958

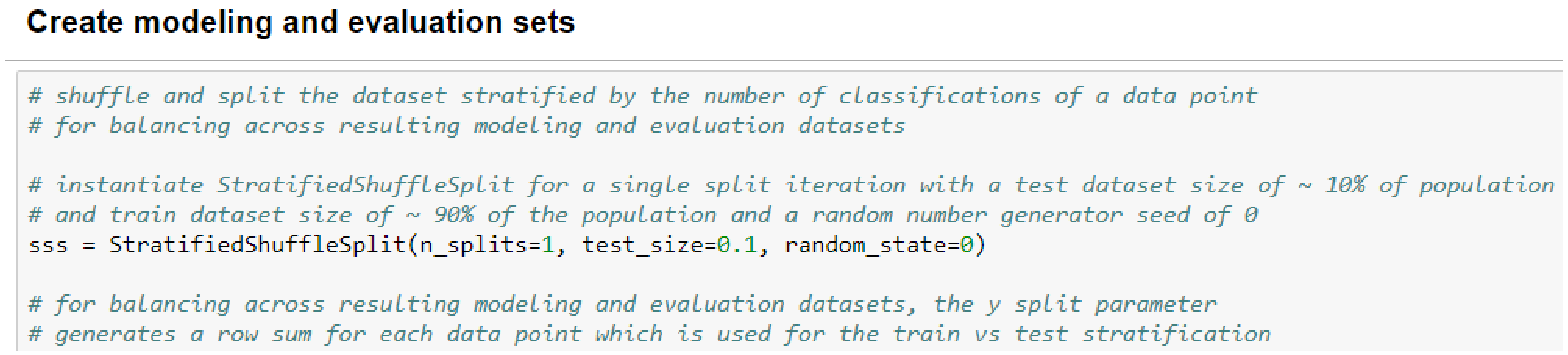
Split Data Characteristics and Prediction Metrics

Iteration 1 (CountVectorizer)

Train Data – Mean Squared Error: 0.050813
Test Data – Mean Squared Error: 0.157229

Iteration 2 (TfIDf/Doc2Vec)

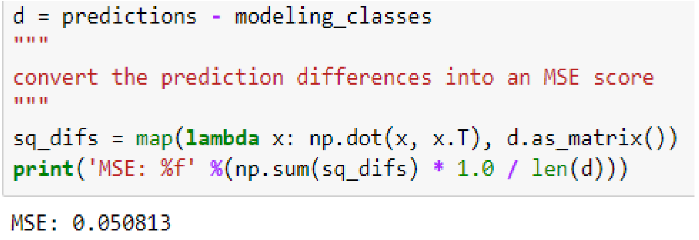
Train Data – Mean Squared Error: 0.034155
Test Data – Mean Squared Error: 0.128779



Create modeling and evaluation sets

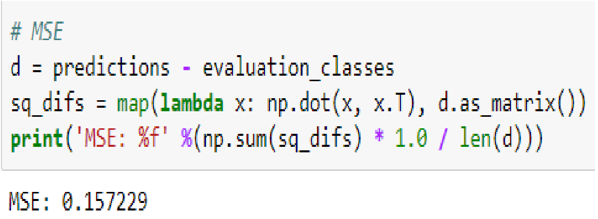
```
# shuffle and split the dataset stratified by the number of classifications of a data point
# for balancing across resulting modeling and evaluation datasets

# instantiate StratifiedShuffleSplit for a single split iteration with a test dataset size of ~ 10% of population
# and train dataset size of ~ 90% of the population and a random number generator seed of 0
sss = StratifiedShuffleSplit(n_splits=1, test_size=0.1, random_state=0)

# for balancing across resulting modeling and evaluation datasets, the y split parameter
# generates a row sum for each data point which is used for the train vs test stratification
```

Iteration 1 (CountVectorizer)



```
d = predictions - modeling_classes
"""
convert the prediction differences into an MSE score
"""
sq_difs = map(lambda x: np.dot(x, x.T), d.as_matrix())
print('MSE: %f' %(np.sum(sq_difs) * 1.0 / len(d)))
```

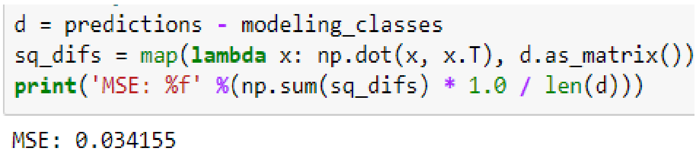
```
MSE: 0.050813
```



```
# MSE
d = predictions - evaluation_classes
sq_difs = map(lambda x: np.dot(x, x.T), d.as_matrix())
print('MSE: %f' %(np.sum(sq_difs) * 1.0 / len(d)))
```

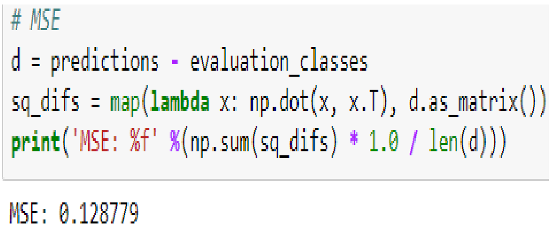
```
MSE: 0.157229
```

Iteration 2 (TfIDf/Doc2Vec)



```
d = predictions - modeling_classes
sq_difs = map(lambda x: np.dot(x, x.T), d.as_matrix())
print('MSE: %f' %(np.sum(sq_difs) * 1.0 / len(d)))
```

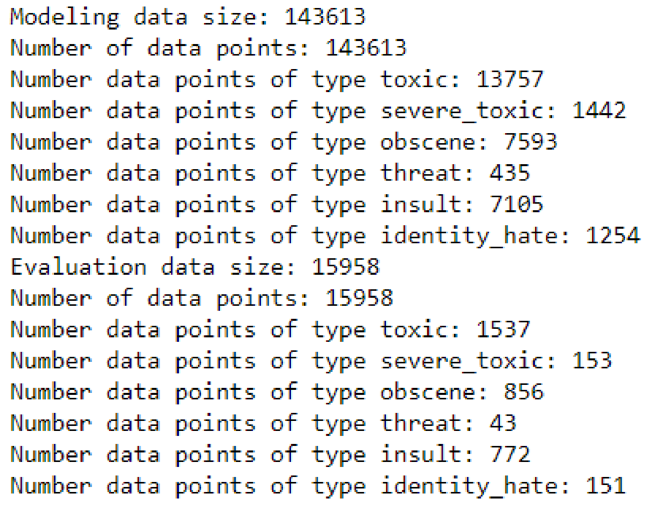
```
MSE: 0.034155
```



```
# MSE
d = predictions - evaluation_classes
sq_difs = map(lambda x: np.dot(x, x.T), d.as_matrix())
print('MSE: %f' %(np.sum(sq_difs) * 1.0 / len(d)))
```

```
MSE: 0.128779
```

Split Data Characteristics (Train:Test - 9:1)



```
Modeling data size: 143613
Number of data points: 143613
Number data points of type toxic: 13757
Number data points of type severe_toxic: 1442
Number data points of type obscene: 7593
Number data points of type threat: 435
Number data points of type insult: 7105
Number data points of type identity_hate: 1254
Evaluation data size: 15958
Number of data points: 15958
Number data points of type toxic: 1537
Number data points of type severe_toxic: 153
Number data points of type obscene: 856
Number data points of type threat: 43
Number data points of type insult: 772
Number data points of type identity_hate: 151
```

Next Steps

- Model tuning
  - trialling stage 1 and stage 2 classifiers other than Random Forests and Neural Networks
  - Parametric tuning of input parameters (# embedding feature size, # estimators in random forest, others)
  - evaluating model with metrics other than MSE (f-score)
  - dimensionality reduction using PCA or TruncatedSVD
- Exploring extant research to further improve model
- Unintended bias analysis and elimination (https://github.com/conversationai/unintended-ml-bias-analysis)
- Visualising the results

References

- http://scikit-learn.org/stable/modules/multiclass.html
- http://scikit-learn.org/stable/modules/classes.html#sklearn-metrics-metrics
- http://scikit-learn.org/stable/modules/generated/sklearn.neural_network.MLPClassifier.html
- http://scikit-learn.org/stable/modules/generated/sklearn.model_selection.StratifiedShuffleSplit.html
- http://scikit-learn.org/stable/modules/generated/sklearn.multioutput.MultiOutputClassifier.html
- http://scikit-learn.org/stable/modules/generated/sklearn.decomposition.TruncatedSVD.html
- https://stackoverflow.com/questions/28160335/plot-a-document-tfidf-2d-graph
- https://stackoverflow.com/questions/33091376/python-what-is-exactly-sklearn-pipeline-pipeline
- https://stackoverflow.com/questions/21208420/why-does-x-dotx-t-require-so-much-memory-in-numpy
- https://docs.python.org/2/tutorial/classes.html
- https://www.tutorialspoint.com/python/python_classes_objects.htm
- http://www.datascienceassn.org/sites/default/files/users/user1/lsa_presentation_final.pdf
- https://www.quora.com/What-exactly-does-the-fit_transform-function-do-to-your-data-explanatory-variable

Appendix

Pipeline

- Automates several steps of the learning process
- 2 primary inputs – Transformer (find set of features, generate new features, select only some good features) and Estimator (performs fit and predict)
- Trains the transformer and then applies the classifier to make predictions

TruncatedSVD

- Dimensionality reduction technique
- Works on term count/tf-idf matrices as returned by vectorizers

MultiOutputClassifier

- Multi target classifier.
- This strategy consists of fitting one classifier per target.
- This is a simple strategy for extending classifiers that do not natively support multi-target classification (such as a basic random forest classifier).

MLPClassifier

- Multi-layer perceptron classifier.
- This model optimizes the log-loss function using LBFGS or stochastic gradient descent. *(LBFGS chosen for this model as it is better suited for smaller datasets)*