Doc2vec, also called a paragraph vector, is a NLP way to to represent documents by encoding them as numeric feature vectors. To build a document vector using a distributed bag of words approach, a document vector is trained by being forced to predict random words from the paragraph as its output.

To build this model, we built dataframes of the source data by combining the document feature files and abstract text files into a testing and training dataframe was determined by the data providers at Kaggle. The test of the abstracts was processing into Spacy document objects and then the stop words were removed using the base Spacy english language model. The source data was then converted into tagged document objects where the classification of the mutation type was attached to the associated text data. Two types of document were created for each document, one used the text from the paper abstract and one used the concatenation of the gene and the variation type. The tagged documents were trained into a document vector using the Doc2Vec model in Gensim and then classified using a linear regression model. This training was done at three different alpha values (0.002, 0.001, 0.003) with 0.002 being the recommended default value and for multiple training epochs (1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 30, 40, 60, 80, 100).

The results of classification with this model are shown below:

For the lower values of alpha, 0.001 and 0.002, the model trained on the abstract text (AT) outperformed the model trained on just the gene and variant (GV), but only by about 10% at best with the GV trained model consistently having about 70% accuracy vs about 80% accuracy on the F1 score. For both models the GV accuracy was consistent for the number of training epochs with a slight drift downward as the number increased. The AT model however, had its accuracy drop by about half around 20 epochs for the alpha of 0.002 and at round 10 epochs for the 0.001 alpha. The model trained at an alpha of 0.003 had a slightly lower overall average and its AT model dropped of at around 8 training epochs. From this, for use of this model, our tests suggest that training on the abstract text data is best with a small alpha value and a small number of training epochs will produce results as good as a greater number of training epochs without risk of a drop off. If you are pressed for time and compute resources, train the model on just the gene involved and the variation type will produce results of almost the same accuracy with no risk of significant drop off using significantly fewer computational resources.

