HW 4: a) Models fit to text data and b) Neural Networks

Start Assignment

Due Dec 10 by 11:59pm **Points** 10 **Submitting** a file upload
File Types html and ipynb

Homework Instructions:

Your final document should be an ipynb or an html file generated from a Jupyter notebook. (i.e.-save your notebook via file>download as>html). The file should be uploaded to this assignment on the course website.

In answering each of the following questions **please include a) the question as a markdown header in your Jupyter notebook, b) the raw code that you used to generate any results, tables, or figures, and c) the top ten or fewer rows of the dataframe (do not include more than ten rows for any table in your report).**

Include any plots or figures generated from your code as well.

Homework Questions:

Part 1: Build a classification model using text data

In part one of the homework, you will solve a text classification task.

You can download the following clickbait data-sets from the aimodelshare library:

```
import aimodelshare as ai
X_train, X_test, y_train_labels, y_test_labels, example_data, lstm_model, lstm_model2 = ai
.import_quickstart_data("clickbait")
```

The data consists of headlines that signify clickbait or not. Training and test data are stored in the following objects: X_train, X_test, y_train_labels, y_test_labels. The remaining objects can be ignored.

In a real application this might allow us to find out what is hard news information (or perhaps to

choose among headlines that are more likely to be clicked).

Use cross-validation to evaluate the results. Use a robust metric for classification (AUC or F1-Score for example), and inspect all models by visualizing the coefficients. (See helper function in our in class notebook for text models.)

To complete part one of the homework do the following:

Import the text data, vectorize the clickbait headline column into an X matrix. Then run logistic regression at least three times and select a single best model. Note that you should create three logistic regression models with different different tokenization approaches. You should not change your modeling approach, you should simply experiment with different tokenizers. Be sure to explain your choices and evaluate your models using cross validation and using test set data.

Part 2: Build a predictive neural network using Keras

To complete part two of the homework do the following:

Run a multilayer perceptron (feed forward neural network) with two hidden layers on the iris dataset using the keras Sequential interface.

Data can be imported via the following link:

http://vincentarelbundock.github.io/Rdatasets/csv/datasets/iris.csv

Include code for selecting the number of hidden units using GridSearchCV and evaluation on a test-set. Describe the differences in the predictive accuracy of models with different numbers of hidden units. Describe the predictive strength of your best model. Be sure to explain your choice and evaluate this model using the test set.