CAPSTONE PROPOSAL

Projects domain background:

The objective of this Capstone project is to apply data science techniques in the area of natural language processing and obtain the original text after a specific transformation is applied.

Problem statement:

According to a research at Cambridge University, it doesn't matter in what order the letters in a word are, the only important thing is that the first and last letter be at the right place
This is true for generally all humans that can read, but it is true for computers too?

Datasets and inputs:

I will use initially use the Blogger Corpus (http://u.cs.biu.ac.il/~koppel/BlogCorpus.htm), which should be enough for the task, but if it wasn't the case, the Wikipedia Links data (https://code.google.com/archive/p/wiki-links/downloads) would be used too.
On the contrary, if the former already consists in too much data, only a subset of it will be used.

Solution statement:

I will implement and deploy a Neural Network model and create a web to show it’s performance to human entered text.

Benchmark model:

I haven’t found any model to compare to.

Set of evaluation metrics:

I will measure the performance by the percentage of words converted back correctly on each blog post.

Outline of the project design:

This project will be done in a SageMaker notebook mainly.

As the preprocessing steps, the blogs will be separated into subwords with the punctuation marks as delimitators (including ' and - inside what would be considered a complete word) and each letter of the subwords will be transformed into numerical value (creating its subsequent vocabulary), resulting in an array of integers. This arrays will have to be padded to enter the same length input according to one of the largest words on the English dictionary: Supercalifragilisticexpialidocious, considering no medical specific word larger than it will be inputted (as a non-medical student will not be able to order them correctly too probably).

The array will be transformed as explained above and will be the input to the model later on. But before use it as input, the dataset will be split into training and test sets to be able to check the performance of the model on unseen data.

The model will consist on a Neural Network, more specifically probably an RNN, which seems well fitted for the task, although I've not decided completely yet as depending on the results obtained, other NN may be tested.

Finally, the model will be deployed and a simple web will be created to showcase the performance of the model using human inputted text.

The above-mentioned design it's the first design I've done, it could add some more functionalities but the basic structure would be the same.