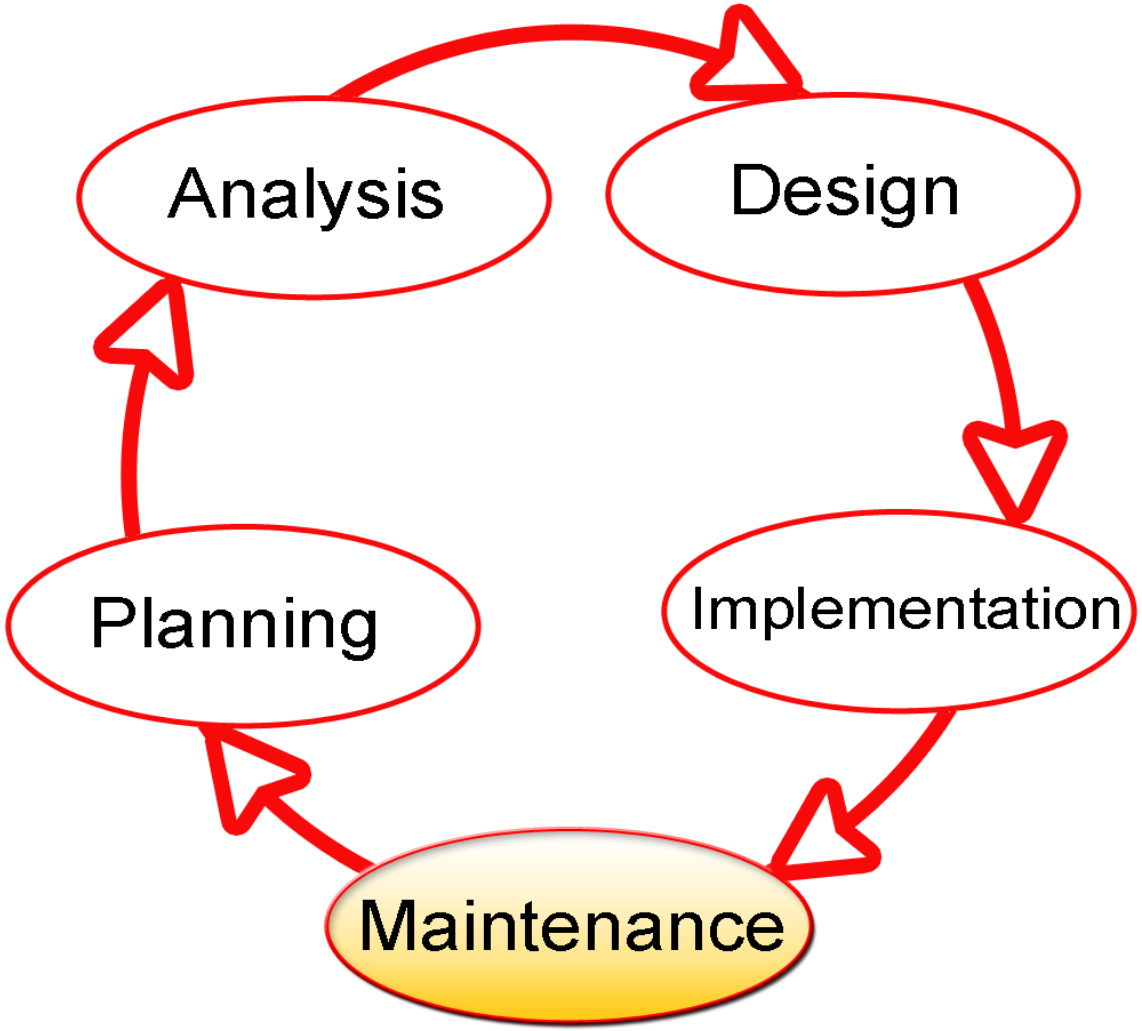
iSegment

Document Segmentor using Machine Learning

The life cycle used during the project:



Planning:

- The planning started with dealing with the every day problem of messed up folders and investing important time of the day to segment them.
- The project planning started with curating the proposal document and then collecting a dataset to look for prototype solution.

Analysis:

- Many datasets were downloaded and looked upon until the BBC dataset was selected for final implementation.
- The BBC dataset is structured and will help getting some real life scenario results as we look into the news articles in it.
- The 400+ documents in each category further helps to drill down the problem more accurately.

Design:

- If it would have been unstructured data then lot of pre-processing would have been needed for same but since I am sticking to structured data as scope for this project, I have not implemented any pre-processing pipeline.
- Though I worked with SpaCy and Gensim for various pre-processing steps as was introduced during the Nanodegree.

Final Pipeline:

Input

Mention target input folder that will be used for training.

Feature

Use TF-IDF vectorizer and chi2 to get features, labels and n-grams from the training dataset.

Model

Use classifier for the training purpose.

Target

Mention the location of target folder.

Predict

Predict the labels for the dataset in the test data.

Segregate

Segregate all the documents from input folder to the categorical folders in the target folder.

Train Dataset:

- Train – The train dataset involves 5 categories of documents:
  1) Business
  2) Entertainment
  3) Politics
  4) Sport
  5) Tech
- Each category contains 10 documents from BBC dataset.
- Each document contains more than hundred words including stopwords.

Test Dataset:

- Test – The test dataset involves mixed bag of all the categories of data into one folder.
- To ease the accuracy calculation procedure, I have renamed the files to their respective categories, though it does not in any sense helps the model in classifying the document.
- The output folders will contain all the test dataset documents only and not training data.

Inputs:

- Input_path – Define path to the training folder.
- Output path is always the folder with name: sorted_(some_random_number) which will contain all the segregated folders and in them we will have documents.

Models Comparison:

- I have applied three models for classification:

Model	Accuracy
Random Forest Classifier	73.9%
Multinomial Naïve Bayes	95.6%
Logistic Regression	86.9%

- Multinomial Naïve Bayes with Random state 7 has exceeded other two models.
- According to me, this happened because the dataset was structures, I can further make the model automatically segregating by choosing the best accuracy model on it's own.

Conclusion:

- Multinomial Naïve Bayes model is working pretty good and showing State of the art results with only 3 mis-classified documents.
- It takes around 7 seconds to run the whole model to segregate around 46 documents.
- I have used k-shot learning to train the model and the training dataset contains only 10 documents of each type. This provides user with ease in-case not sufficient of supervised, labelled data is present.