**Task 1 : Audio cat vs dog classification**

**Libraries used :** pandas, numpy, librosa, tensorflow, matplotlib, seaborn, sklearn

**Brief Code explanation:** The ipython notebook for this task contains well commented code. Hence for a clear understanding of the code, an ipython notebook should be referred to. Here's the brief summary of the approach.

* First we download the data into the working directory and then do basic directory looping using the OS module to read train and test data for each class , i,.e, cats and dogs. We only read the name of the file and corresponding wav file
* We then create a sample dataframe from the above data
* Next, we use mfccs algorithm to extract the features from the wav file. Librosa provides method to do this
* We extract features for both train and test data and use that as our training data and test data
* We build model using tensorflow and pass the training data to it
* Once we are done training, we create loss and accuracy curve and plot confusion matrix

**Results:**

| **Name** | **Model** | **architecture** | **Training score** | **Val score** |
| --- | --- | --- | --- | --- |
| Model | MLP | (100,150,1501) | 98 | 88 |
| Model2 | MLP | (25,25,25) | 100 | **9**1 |

**Task 2: Sentiment Analysis**

**Libraries used :**

Pandas, numpy, nltk, Tensorflow, matplotlib, string, sklearn, gensim.

**Brief Code explanation** : The ipython notebook for this task contains well commented code. Hence for a clear understanding of the code, an ipython notebook should be referred to. Here's the brief summary of the approach.

* The data was downloaded from the given site and loaded into pandas dataframe. The loaded data ( review\_text) was cleaned using nltk modules.
* Cleaning included :
* removing punctuations
* removing stop words
* lemmatization.
* Then the clean data was split into train test sets using skleanr’s train\_test\_split module. Then both the train and test set were transformed into vectors using the following algorithms
* Count vectorizer from sklearn
* TF-idf Vectorizer from sklearn
* Wrd2vec

* Next for each of the transformed final vectors, The following models were used with the vectors
* MLP classifier
* GRU from keras
* LSTM from keras
* For each of the above models, a confusion matrix, classification report was plotted.

**Results and inference:**

| Model | Embeddings | Architecture | Train Score | Test score |
| --- | --- | --- | --- | --- |
| MLP | Countvectors | (8,8,8) | 85 | 72 |
| MLP | Tf-id | (10,5) | 83 | 74.08 |
| GRU | Default embeddings | 32 GRU units, embedding dim = 100, dense = 5 | 91 | 67 |
| GRU | Word2vec | Same “ “ | 69 | 69 |
| LSTM | word2vec | Same with LSTM units = 32 | 69 | 69 |

The confusion matrix for the first two MLP models gives decent results but the same for the last three models is problematic because the last three models predict 4 ( meaning 5) as ratings always.This may be as a result of class imbalance in the dataset. To solve this either we need to create synthesized data to balance the class or get more data for other classes.