

# ROOT2AI Assignment

- Mir Muhammad Mujtaba

# Problem Description

We are given a dataset which has 2 columns namely 'Text' column and 'Target columns'. The objective of this task to train a machine learning algorithm that can predict the class label given the text. The 'Target' column has 11 different labels.




# Approach

There was only one input variable ('Text' column) that was given. Firstly, I performed the generic text processing. Secondly, I used both the classical machine learning algorithms i.e., Decision Trees and Logistic Regression as well as Deep Learning model i.e., LSTM for the task.

For the classical approach, the vector representation for sentences was derived as follows: For example, the sentence is 'i love machines'. I took the Glove vector representation of individual words in the sentence i.e., v1 for 'i', v2 for 'love' and v3 for 'machines', summed them and took the average.

For LSTM, I simply fed the Glove vector representation for each word sequentially and on top of that built a multi layered perceptron.



# 1. Decision Tree Algorithm

Decision Tree gave the poorest result among all the three models. In terms of function mapping complexity, Decision Trees lie between Logistic Regression and Neural networks. This means that in Decision Trees the decision boundary that is learned finally is parallel to either of the axis in the input space. So the poor performance can mean that the data points can be separated using a simpler function which is why the second model that I used was Logistic Regression. Accuracy on test data was 0.44.



# Limitation

The input variables used are the average of the Glove Vector representation. One disadvantage of this approach is that the sentences which have an inherent sequential order lose this sequential information as a result. As a result, this has an effect on the final model. Also, since Decision Tree can learn non-linear function which in this problem seems not to give beneficial results.



## 2. Logistic Regression

Logistic Regression learns a hyperplane that separates the data points belonging to different classes. Due to the reasons mentioned before, I trained a Logistic Regression and the performance actually improved as I thought it would. Accuracy on test data was 0.59.



# Limitation

Just like Decision Trees, here too the loss of sequential information due to the use of average of Glove vectors as a representation for sentences is a factor that does have an impact on the final results.



### 3. LSTM based Neural Network

Even though Logistic Regression gave better results, I wanted the results to improve. Since neural networks are universal function approximators, therefore I used them as well. As expected the results improved significantly. Best accuracy score on test data was 0.66





# Limitation

In case of LSTM, the number of parameters exceed 3 Million which makes the model too complex for this problem. The very fact that the model starts to overfit just after the second epoch proves it. Also LSTMs can be slower to train since the number of parameters are very high.

