# Solution Documentation

Team Name: **Naïve Bears**

Team Leader Name and email-id: **Rohit Upadhyay** ( [rupadhyay@deloitte.com](mailto:rupadhyay@deloitte.com) )

Additional team member names and email-ids: **Amit Agarwal** ( [amitagarwal7@deloitte.com](mailto:amitagarwal7@deloitte.com) )

Please return this form to [USIDLabscorner@deloitte.com](mailto:USIDLabscorner@deloitte.com)

## Background

Rohit Upadhyay, Senior Consultant with 6 years of experience in Data warehousing and Business Intelligence. He has completed the Machine Learning Guild Apprentice level and applied for Mastery level. He has built multiple POCs spanning across Regression, Classification and Image Classification using Deep learning. His team was runner up in the recent LSHC Hack-4-Cause challenge (Image Classification solution). He has worked with Amit in the past to build a Regression Model Solution, which was showcased to LSHC partners in the recently concluded LSHC week. He entered this competition to learn about implementation of deep learning in NLP.

Amit Agarwal, working as a Manager, has a total experience of over 11 year especially in Microsoft Technology. During his Project tenure he developed interest in the field of analytics and that led him to cross train himself in Machine Learning Skills. He along with his team even represented Bangalore location in the Mega Finale of Modelling competition organized as part of Analytics summit this year in Hyderabad. Having worked on multiple regression model in the past this competition provided an opportunity to foray into NLP area.

## Summary

We’ve built an RNN model with AWD-LSTM architecture, using FASTAI library. First, a language model was built using transfer learning, which was then utilized to build the classifier.

Optimization -

* The Language Model and Classifier was optimized by handling class imbalance of target labels
* Tuning the Batch size and dropout rate
* Finding optimum learning rate using callbacks
* Training the last few layers of the network individually rather than training all layers at once

We’ve used only the commentary text as the feature to train the model and utilized the total run in over as a rule to correct the predictions.

* Features used – Commentary text
* Software used – Jupyter Notebook
* Training time – 40 Minutes

# Appendix

## Code

The code should be automated such that if the input is the training data (or any external data allowed by the competition), we should be able to get the prediction data as the output.

Please attach all code required to produce the solution in an email to [USIDLabscorner@deloitte.com](mailto:USIDLabscorner@deloitte.com).

## Dependencies

* Programming language or tool - Python
* Training Environment – Kaggle Kernel for GPU computation
* Libraries – fastai, numpy, Pandas, matplotlib

## Instructions

Step by step instructions on how to create the predictions from the code. If any parameters are hard-coded rather than arrived at through a defined process, explain how the parameters were chosen.

* Variable ‘datapath’ stores location of training csv file
* Variable ‘datafile’ stores the training csv name,
* TrainingData.csv - contains only the **Target** and **Commentary** columns from original training csv
* TestData.csv – contains only the Commentary column from original test csv
* Submission.csv is the initial csv generated from classifier
* final.csv is the final output
* Model was trained on a GPU enabled Kaggle kernel, which takes 40 minutes for training