

# Learning and Exploring

## WEEK-1

- 1.) In the first week started exploring python, django, mongodb and unit testing in python.
- 2.) Python is completed and took the idea and workflow of django and mongodb.
  - Model view template in django,
  - How it stored data, JSON/BSON, insert update delete.
- 3.) Explored the product and project Incare (looked at the staging)
- 4.) Looked at the code base and repositories what is helper, where is all the business logic written

## WEEK-2

- 1.) Project was given so started exploring things on that.
- 2.) Started learning and researching about NLP and its libraries and other different things.
  - tokenizing, POS tagging, parsing, regular expressions, lemmatization, stemming
  - Dialogflow, tensorflow, bert, spacy
- 3.) Read some research papers to get the idea of the algorithms which involves various algos like use of TF-IDF weighting, use of annie and rapier.
- 4.) Explored concepts of machine learning to get the idea of it.
  - Regression
  - Clustering

## WEEK-3

- 1.) Came up with my own 3 approaches.
  - a.) 1st approach: Count the number of occurrences of stored triggering words and select the particular sentence with count > threshold. After that, map those sentences according to given functions.
    - Use of nltk library to sentence tokenize and store bag of words.
    - Challenges faced were mainly in mapping the sentence to a particular functions
    - The problem with the approach is that it is very static and the model will not learn itself in time it will just run the same things on different datasets.
  - b.) 2nd approach: using nltk libraries and machine learning algorithms to solve the problem.
    - a. Lexical features
      - Tokenizer
      - Spell correction
    - b. Syntactic Features
      - POS - Temporal expression tagging (sequence, duration and range) & POS tagging
      - Syntactic pattern - Look for past tense tag (eg., VBD, VBN, etc.,) and ignore them as they are past events
    - c. Semantic Features

- Synonymy – from NLTK WordNet – to retrieve words that are synonymous to required events (Marriage, Birthday, Meeting, Anniversary, Seminar)
- Named Entity recognition to find location of the event
- For the time of the event, regular expressions and DateTime is used.

Since, it will be a fully fledged model so there it will require large files.

c.) 3rd approach: using a 3rd party API which is dialog flow. Training the model and then importing it on our python modules.

- Problem with this approach is training of the model is very static because it is generally used as a chat bot so training it for every message.

NOTE: working simultaneously in 2nd and 3rd approach because 1st one is a naive solution and wanted some improved strategy.