**Week-2 Report:**

Problem Statement:

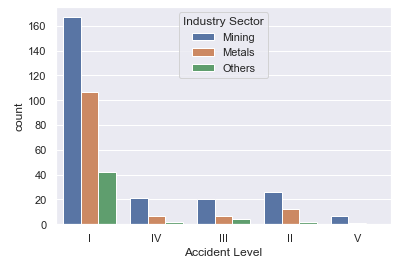
One of the biggest industries in world suffering with the employee’s accident and major injuries and sometimes employee’s also die in such environment. Our task is to create NLP based Chatbot so that professional get details of safety risk as per accident description.

Dataset:

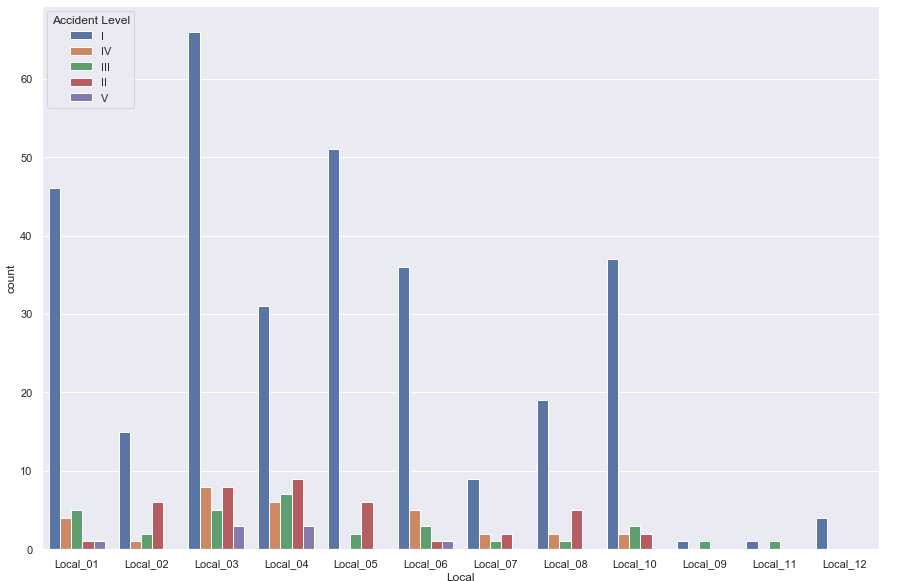
Dataset consist details of 12 different plants and have records as [“Date”, “country”, ”Industry sector”, ”Accident Level”, “Genre”, “critical Risk”, “Description”] with 425 rows and 11 column

Observation:

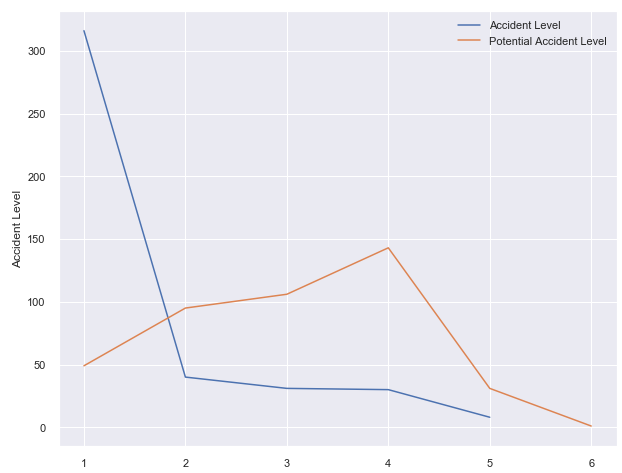
1. Dataset have 425 rows and 11 columns
2. Mining sector is impacted by all level of accident, most accident have recoded minimum severe level for mining sector.



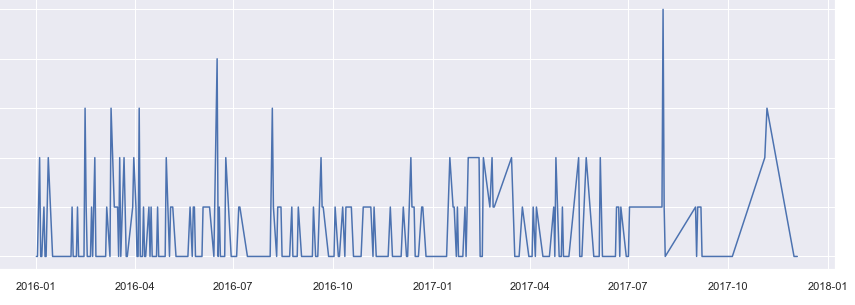
1. Men are involved in all level of accident where female are involved in 1st and 2nd level only which is minimal accident level.
2. Most of accident causes is not defined properly as it is selected as OTHER but few major accident causes are “Pressed”, “Manual Tools”, “Chemical substances”, “Cut”, “Bees”, “Venomous Animal”
3. City number 3 is involved in most of accident where city number 12 is involved in minimum accident.
4. City number 3 and 4 are involved in all type of accident level.



1. Most and all type of accident levels are recoded in mining sector followed by metal industry sector.
2. We can observe that most of accident have minimal accident level but high potential risk.



1. There are high number of accident have been recorded in 2017-07 and 2017-07



Data cleaning:

1. For better understanding change column name from data to date and Genre to Gender.
2. We do not have any null value.
3. Removed irrelevant data (Numbers and Punctuation) and converted to lower case and tokenize data.
4. Removed stop words.
5. We have performed stemming and lemmatization, stemming has changed most word incorrectly so we go with lemmatization where we got Better result.

Data Encoding (Initial):

1. Bag of words.
2. TF-IDF encoding.

Models:

1. Random Forest Classifier:

* Preparation:
  + Created new data frame with required columns only.
  + There are six different levels of Potential Accident Levels, replace each with numbers
  + Created TF-IDF for columns.
  + Splitting data in test and train with ratio of 30:70
* Model:
  + RandomForestClassifier(n\_estimators = 1000, n\_jobs=-1)
* Score:
  + 39.80
* Tuning:
  + Change the dataset with stop words and perform again.
  + Score:
    - 39.80
* Conclusion:
  + Random forest failed to perform, got very poor result for test classification.

1. FastText:

* Preparation:
  + Fastest expect “\_label\_” in column, added same in target column.
  + Saved data as text file
  + Splitting data into test and train in ratio of 20:80
* Model:
  + fasttext.train\_supervised('fastext.datas-fasttext-train.txt', wordNgrams = 2, epoch=300)
* Score:
  + - * (340, 0.9970588235294118, 0.9970588235294118)

* Performance Evaluation: Below is the Classification report of each category

CLASSIFICATIION METRICS

precision recall f1-score support

\_\_label\_\_IV 0.97 0.80 0.88 49

\_\_label\_\_III 0.78 0.87 0.83 95

\_\_label\_\_I 0.83 0.93 0.88 106

\_\_label\_\_II 0.91 0.90 0.90 143

\_\_label\_\_V 1.00 0.61 0.76 31

\_\_label\_\_VI 1.00 1.00 1.00 1

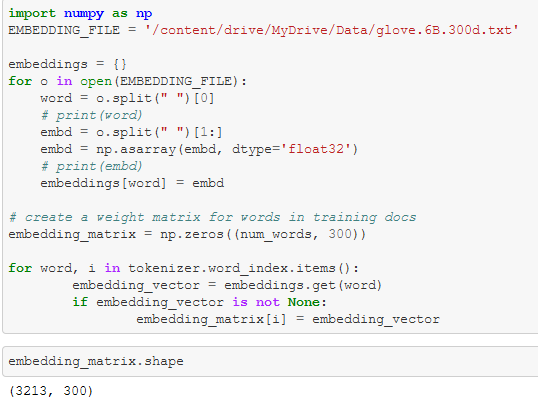
accuracy 0.87 425

macro avg 0.92 0.85 0.87 425

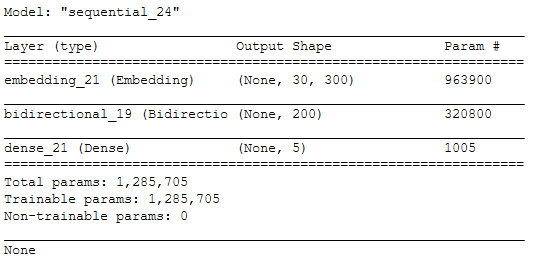
weighted avg 0.88 0.87 0.87 425

1. LSTM:

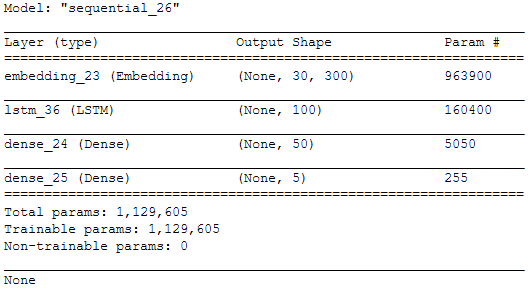
* Generation Embedding glove



* Preparation(Accidental level as target):
  + Import required modules and set vocab size to 5000
  + Set embedding vector feature to 100
* Model:
  + Model summary:
    - Model-1



* + - Model-2



* + Perform Label encoding on target columns.
  + Created dummy matrix of size same as target column label encoding to divide data into test and train properly.
* Score:
  + Train Model-1





* + Train Model-2



