CAPSTONE PROJECT- INTERIM REPORT **Batch: AIML Feb'22A**

NLP-1 Industrial Safety Chatbot

Mentor: Abheesta Arnav

Group: Gaurav Kumar, Shreya Adarsh, Naga Hari Babu KV, Gayathri K, Kishorr SR, Sathya

Contents

1. Summary of problem statement, data and findings

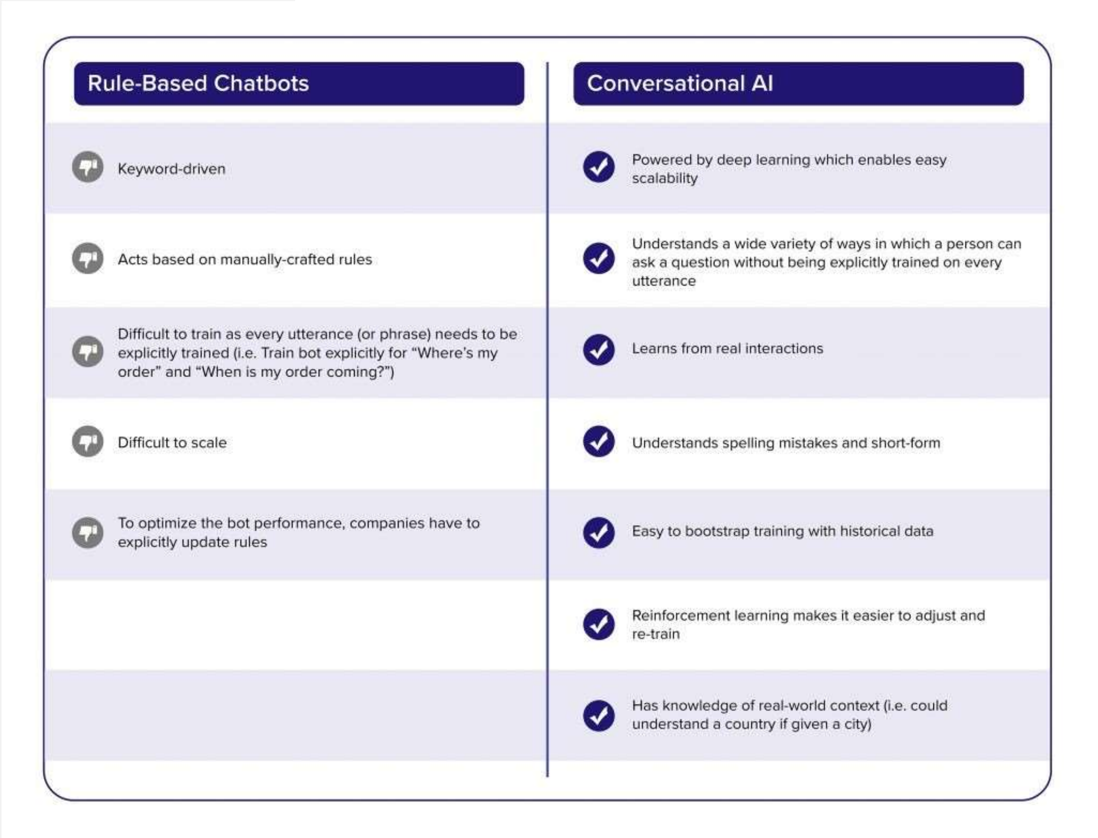
1.1 Introduction

This capstone project is based on building semi-rule chatbot which would help certain industries to understand why employees continue to suffer some injuries/accidents in plants and try to improvise on the same. This report gives us more insights based on all the given milestones.

1.2 Overview

Chatbots are software that use natural language processing (NLP) to engage in conversations with users. Rule-based chatbots provide answers based on a set of if/then rules that can varyin complexity. These rules are defined and implemented by a chatbot designer. At this point, it's worth adding that rule-based chatbots don't understand the context of the conversation. They provide matching answers only when a user uses a keyword or a command they were programmed to answer.

Rule Based vs AI Bots



Rule-Based Chatbots	Conversational AI
👎 Keyword-driven	✓ Powered by deep learning which enables easy scalability
👎 Acts based on manually-crafted rules	✓ Understands a wide variety of ways in which a person can ask a question without being explicitly trained on every utterance
👎 Difficult to train as every utterance (or phrase) needs to be explicitly trained (i.e. Train bot explicitly for "Where's my order" and "When is my order coming?")	✓ Learns from real interactions
👎 Difficult to scale	✓ Understands spelling mistakes and short-form
👎 To optimize the bot performance, companies have to explicitly update rules	✓ Easy to bootstrap training with historical data
	✓ Reinforcement learning makes it easier to adjust and re-train
	✓ Has knowledge of real-world context (i.e. could understand a country if given a city)

1.3 Problem Statement

DOMAIN:

Industrial safety. NLP based Chat-bot.

CONTEXT:

The database comes from one of the biggest industries in Brazil and in the world.It is an urgent need for industries/companies around the globe to understand why employees still suffer some injuries/accidents in plants. Sometimes they also die in such environment.

DATA DESCRIPTION:

This database is basically records of accidents from 12 different plants in 03 different countries which every line in the data is an occurrence of an accident.

COLUMNS DESCRIPTION:

Data: timestamp or time/date information

Countries: which country the accident occurred (anonymous)

Local: the city where the manufacturing plant is located (anonymous)Industry sector: which sector the plant belongs to

Accident level: from I to VI, it registers how severe was the accident (I means notsevere but VI means very severe)

Potential Accident Level: Depending on the Accident Level, the database also registers how severe the accident could have been (due to other factors involved in the accident) Gender: if the person is male of female Employee or Third Party: if the injured person is an employee or a third partyCritical Risk: some description of the risk involved in the accident Description: Detailed description of how the accident happened.

Link to download the dataset:
https://drive.google.com/file/d/1_GmrRP1S2OIa02KlfOBNkYa8uxazGbfE/view? usp=sharing

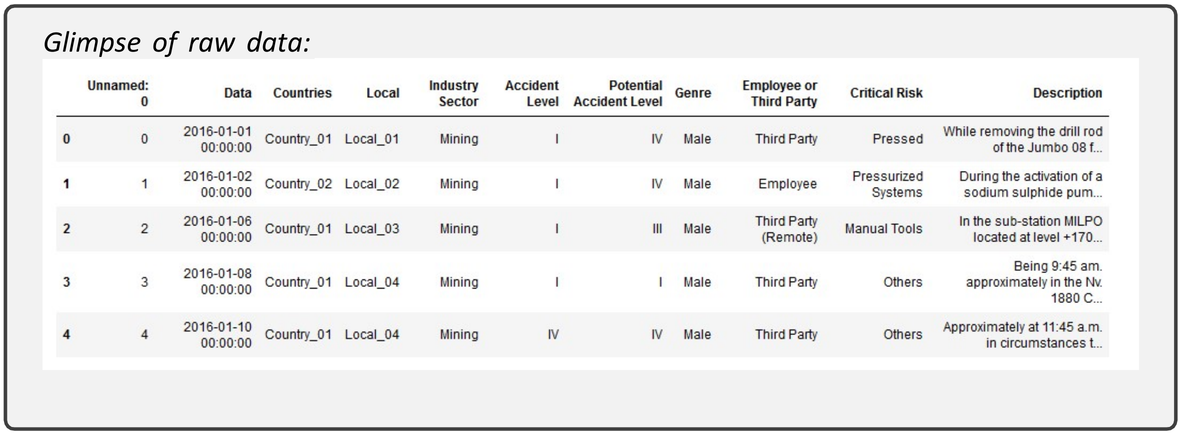
PROJECT OBJECTIVE:

Design a ML/DL based chatbot utility which can help the professionals to highlightthe safety risk as per the incident description.

1.4 Data Analysis

1.4.1 Data Pre-processing:

Post importing required libraries and data, we started checking on profile of the data and took actions accordingly. Pre-processing includes five-point summary, detecting null values, removalof unnecessary columns, data types, unique values for each column, changing few variables name, feature engineering, stopwords removal, stemming and tokenization.



Glimpse of raw data:

	Unnamed: 0	Data	Countries	Local	Industry Sector	Accident Level	Potential Accident Level	Genre	Employee or Third Party	Critical Risk	Description
0	0	2016-01-01 00:00:00	Country_01	Local_01	Mining	I	IV	Male	Third Party	Pressed	While removing the drill rod of the Jumbo 08 f...
1	1	2016-01-02 00:00:00	Country_02	Local_02	Mining	I	IV	Male	Employee	Pressurized Systems	During the activation of a sodium sulphide pum...
2	2	2016-01-06 00:00:00	Country_01	Local_03	Mining	I	III	Male	Third Party (Remote)	Manual Tools	In the sub-station MILPO located at level +170...
3	3	2016-01-08 00:00:00	Country_01	Local_04	Mining	I	I	Male	Third Party	Others	Being 9:45 am. approximately in the Nv. 1880 C...
4	4	2016-01-10 00:00:00	Country_01	Local_04	Mining	IV	IV	Male	Third Party	Others	Approximately at 11:45 a.m. in circumstances t...

Insights gained on data and data pre-processing:

[1]   Small data set(425x11) but with relevant information.

[2]   No missing values in the dataset.

[3]   Also, removing unnecessary column named ”Unnamed” as we do not know any related metadata and adds no value to the analysis.

[4]   Renaming few columns to their appropriate names for better data visualizing

[5]   There were ‘7’ duplicates found in the dataset and were dropped from the dataframe using appropriate function

[6]   Five Point Summary analysis:
Country 01 is the country where most of the accidents happen (more than 50%). Local 03 (which also belongs to Country 01) is where most of the accidents happen. Mining is also the most significant contributor to accidents.
Male (95%) and Third Party (43%) also counts for kind of people that suffers more accident.

[7]   Countries where the dataset was collected are anonymized, but they are all located in South America. So in this analysis, let’s assume the dataset was collected in Brazil.   Brazil has four climatological seasons as below.

Spring: September to November
Summer: December to February
Autumn: March to May
Winter: June to August
We created seasonal variable based on month variable.

2. Summary of the Approach to EDA and Pre-processing

Include any insightful visualization you have teased out of the data. If you've identified particularly meaningful features, interactions or summary data, share them and explain what you noticed. Visual displays are powerful when used well, so think carefully about what information the display conveys.

Once the libraries and data are imported, the next step is Exploratory Data Analysis (EDA). This step is used to analyze and investigate data sets and summarize their main characteristics, often employing data visualization methods. It is used for gaining a better understanding of data aspects like:

- main features of data
- variables and relationships that hold between them.
- identifying which variables are important for our problem.

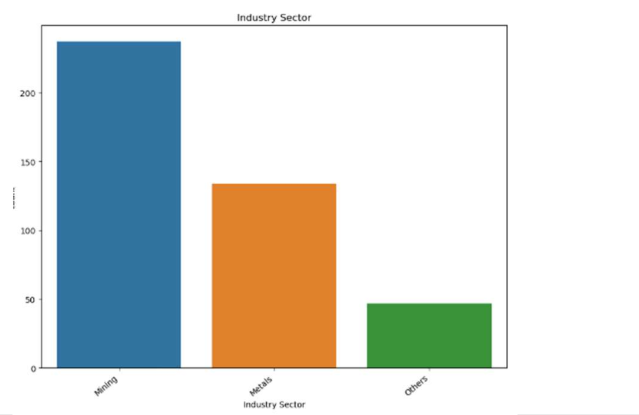
In this Project, we have formed the following Exploratory Data Analysis and Pre-Processing:

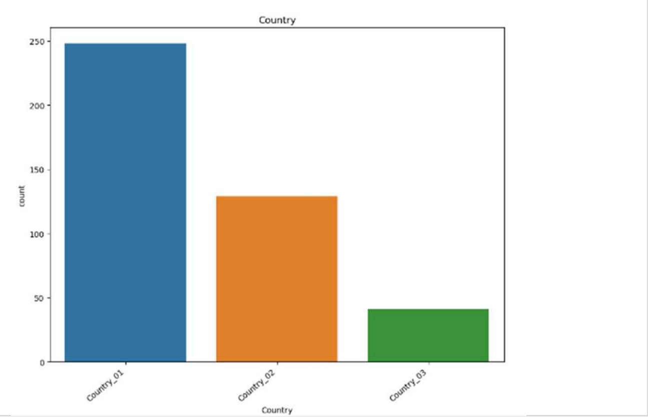
2.1 Data Cleansing

- Here we have dropped the 'unnamed' column and then renamed a few columns basis our requirements.
- Then we have added few more columns for better analyses of data such as:
  - Date
  - Year
  - Month
  - Day
  - Weekday
  - Week of the Year

Note: The above columns were extracted from 'Date' field given in the data.

- Removing Duplicates
  - We checked for duplicates and seven duplicate entries were found – which were removed.
- Adding 'Seasonality' to the data
  - Our hypothesis was that 'seasons' may have an impact on accidents especially in certain sectors. To prove or disprove our hypothesis, we decided to look through various websites that could help us map the months with corresponding seasons in Brazil.
  - Following seasons were added basis publicly known information:
    - Spring
    - Summer
    - Autumn
    - Winter
- Finding Null Values:
  - No NULL values were found in the dataset
  - No value found in the dataset is "not applicable (NA)"
- Checking for Unique Values:
  - We checked for unique values in each of the columns to establish typographical errors and understand the data further. For e.g.,
  - The dataset comprises of accident records from 3 different countries (Country-1, Country-2 & Country-3). All of these countries are from South America region.
  - The data is collected from three main sectors such as Mining, Metals and Others.
  - Mining makes up more than ~50% of the total records followed by Metals.
  - There are almost equal number of 'Employees' and 'Third-party' staff in the Accident records.



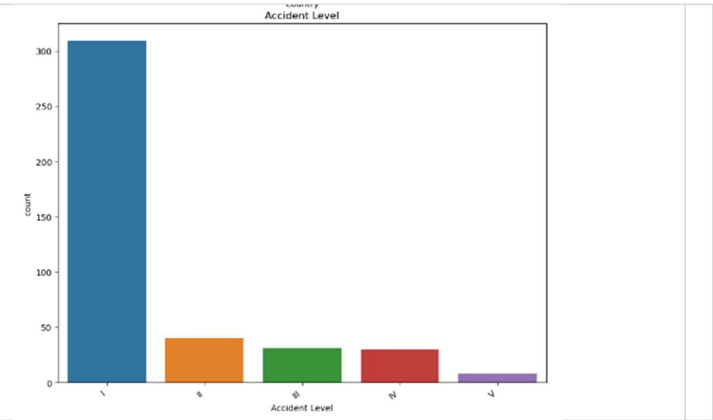


2.2 Key Insights of the Analysis performed.

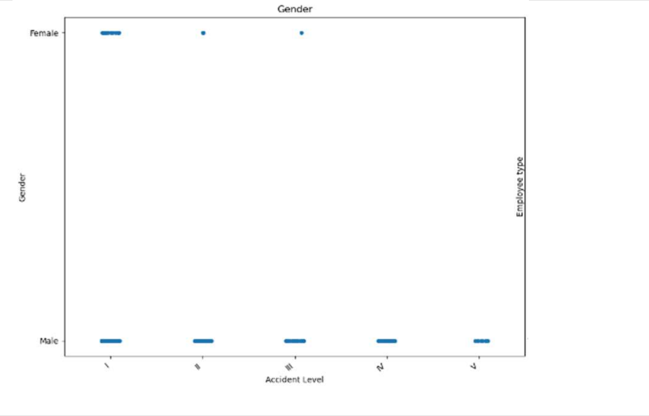
We did both Univariate and Bivariate analysis of data to analyze the distribution of the various variables present in the data and analysis of any concurrent relation between two variables or attributes.

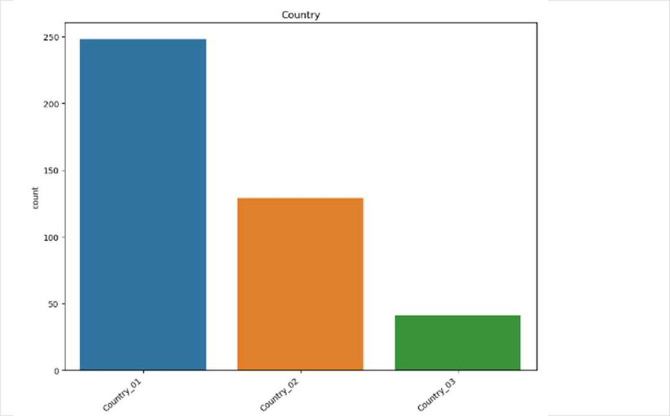
We would like to highlight some of the key insights from various analyses performed on the data.

- <u>Accident Level:</u>
  - Accident Level varies from Level-1 to Level-5 wherein Level-5 is the more serious type of Accident.
  - In the given dataset, the Accident Level is highly imbalanced. Around 75% of the total accidents are of Level-1.
  - Potential Accident Level: It varies from Level-1 to Level-6 in the increasing order of severity.



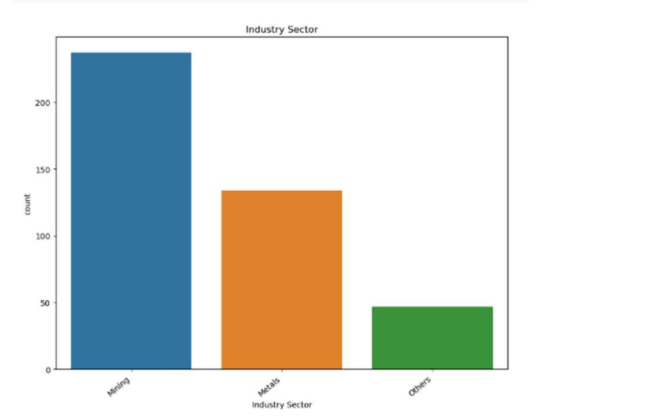
- <u>Accident level Analysis:</u>
  - Around 95% of the accidents have ‘Male’ workers. It also indicates that most of the risk prone areas have male workers.
  - Understandably, females are involved in the lesser severity accidents (Level 1 to 3). There is no record of Female involved in Level 4 or Level 5 severity Accidents. It seems there are more Males working in the high-risk areas.
  - The highest number of accidents are recorded in Country_01.
  - Most of the accidents (74%) are of low Severity (Severity-1). It seems minor accidents are large in numbers.

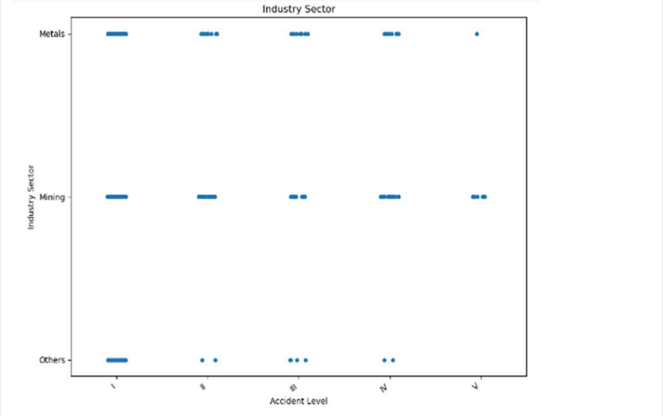




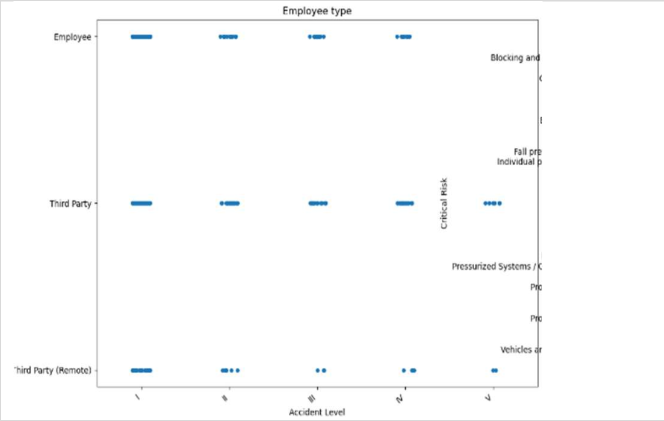
- Risk Areas:
  - The maximum number of risk areas is of category ‘Others’.
  - The other top three risks areas are ‘Pressed’, ‘Manual Tools’ and ‘Chemical substances’.

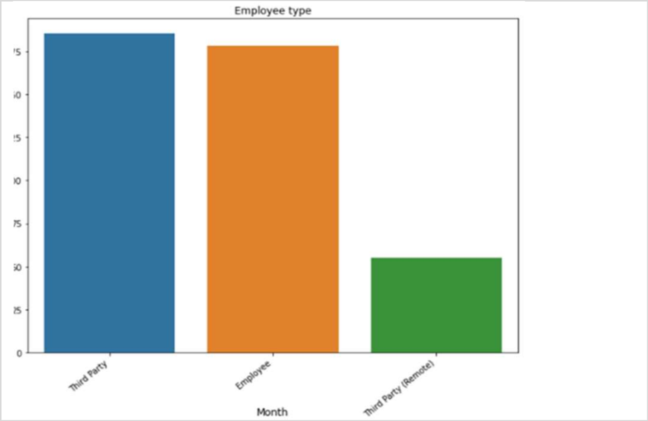
- Sector Analysis:
  - Of all the three sectors, Mining reported the greatest number of accidents, probably due to the nature of the work involved.
  - Mining has reported the greatest number of Level V accidents. Additionally, the occurrence of accidents is in months when it rains the most number of days in months. This is probably due to the fact that mining involves digging deep into the earth for precious metals and hence becomes risky in rainy days.



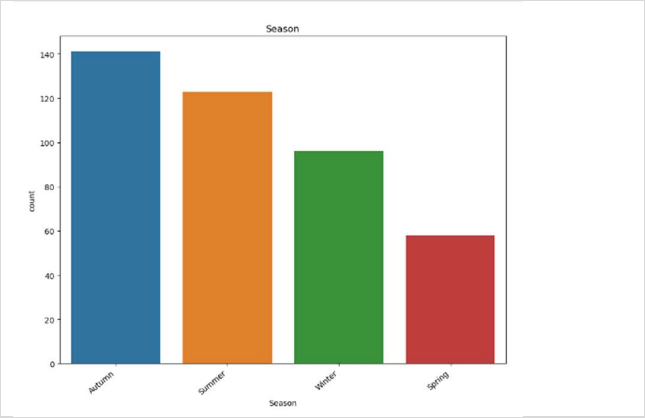


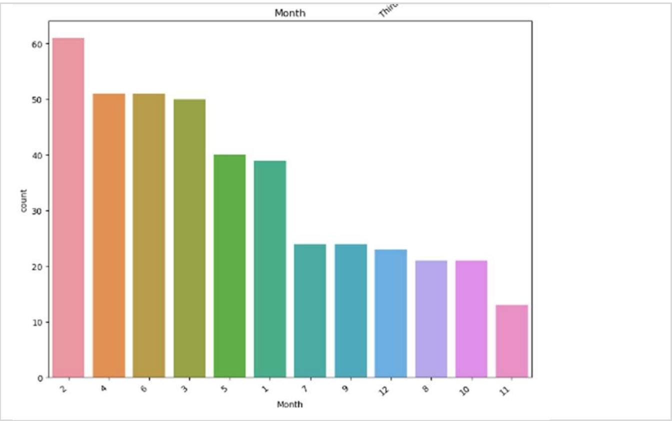
- Accident - Employee relation:
  - Most accidents that are reported have impacted third party staff. This may be due to inadequate training and risky safety measures for them.
  - Most number of Level V and Level IV happened with Third Party Staff. This may be due to inadequate training and safety gears provided to them.
  - Understandably, remote workers have a smaller number of accidents.



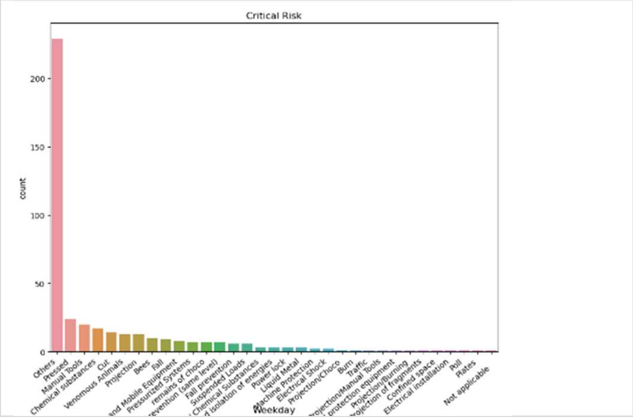


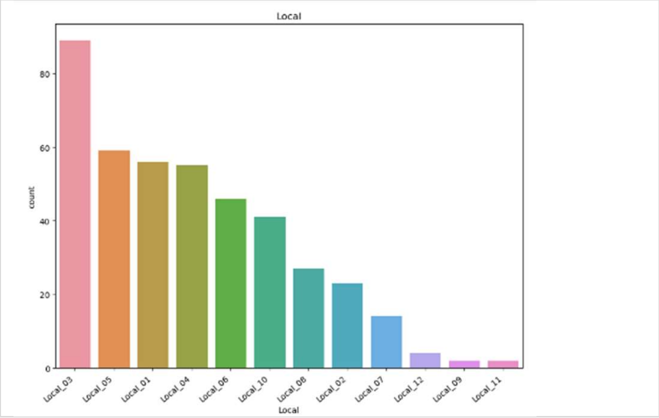
- Various Patterns in Accidents
  - The highest peak of accidents occurs in the month of February followed by April and June.
  - The months from January to March have more accidents and decreases over the year.
  - Brazil has most rainy days between December and March (across Summer to Autumn season) – a time when a greater number of accidents are reported.



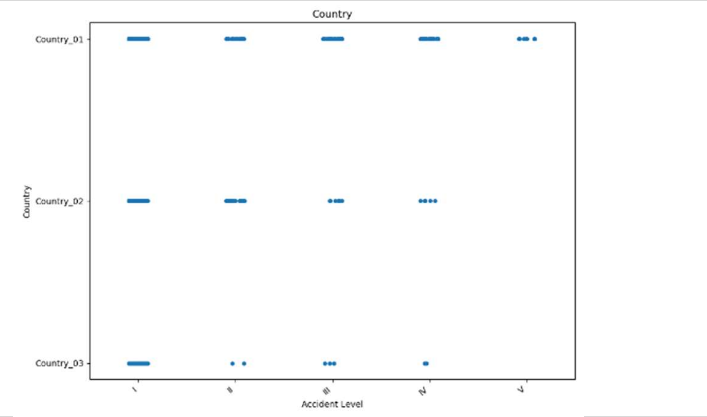


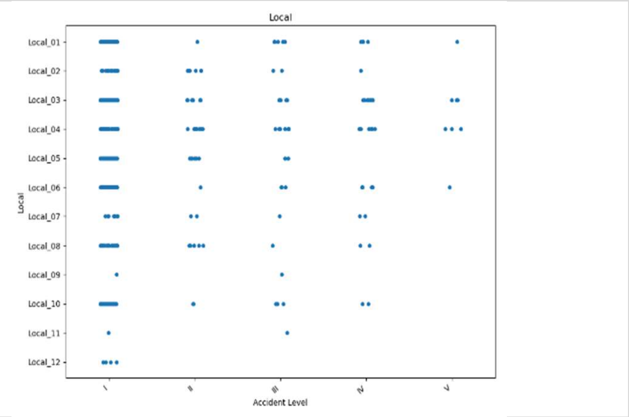
  - The majority of cities have reported accidents during the Autumn season.
  - ‘Critical Risk’ count is highest in ‘Others’ category followed by ‘pressed’ and ‘manual tools’.
  - Highest number of accidents are reported in Local_03 followed by Local_05, Local_01 and Local_04.





  - Country_01 is the only country that has reported ‘Accident Level V’. In addition, Country_01 has the reported maximum number of Accidents across all categories.
  - Local_04 has reported most number of accidents in ‘Accident Level V’.





3. Deciding Models and Model Building

3.1 Data Modelling

While building a predictive model we follow several different steps. We first do exploratory data analysis to understand the data well and do the required preprocessing. After the data gets ready, we do modelling and develop a predictive model. This model is then used to compute prediction on the testing data and the results are evaluated using different error metrics. In this project, for modelling part we have proceeded with different ML models, Neural Network and Bidirectional LSTM.

3.1.1 ML Models and Neural Network:

Applying different ML models and Neural Network will give us a comparative overview on the analysis. To start off we proceed ahead with further pre-processing and then built the respective model.

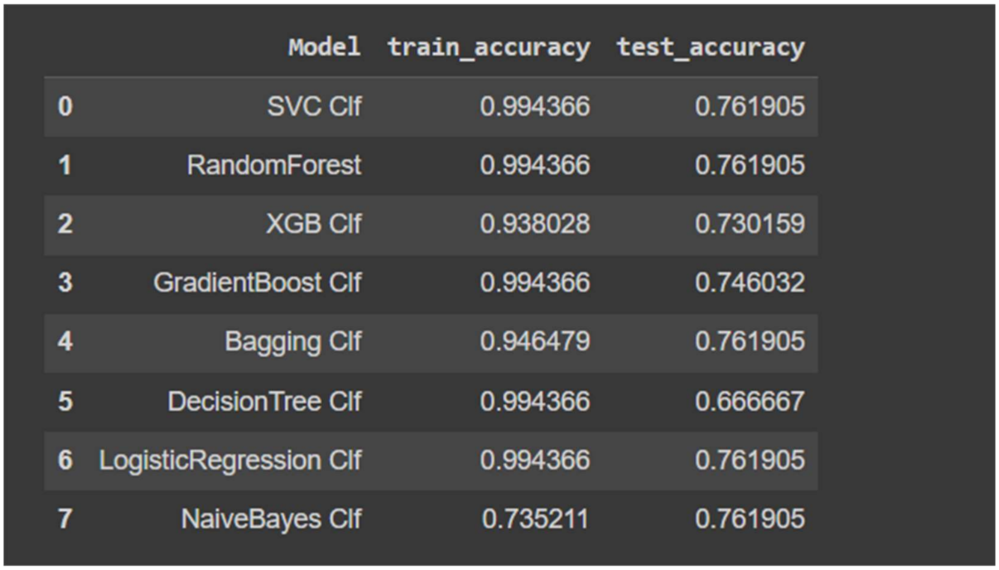
Methodology-

Pre-processing-

[1.] Description column had to undergo few data treatments like pre-processing (lowercase, stop words, s) and word embedding using glove.

3.1.1.1 Opting for Accidents Level as our target variable.

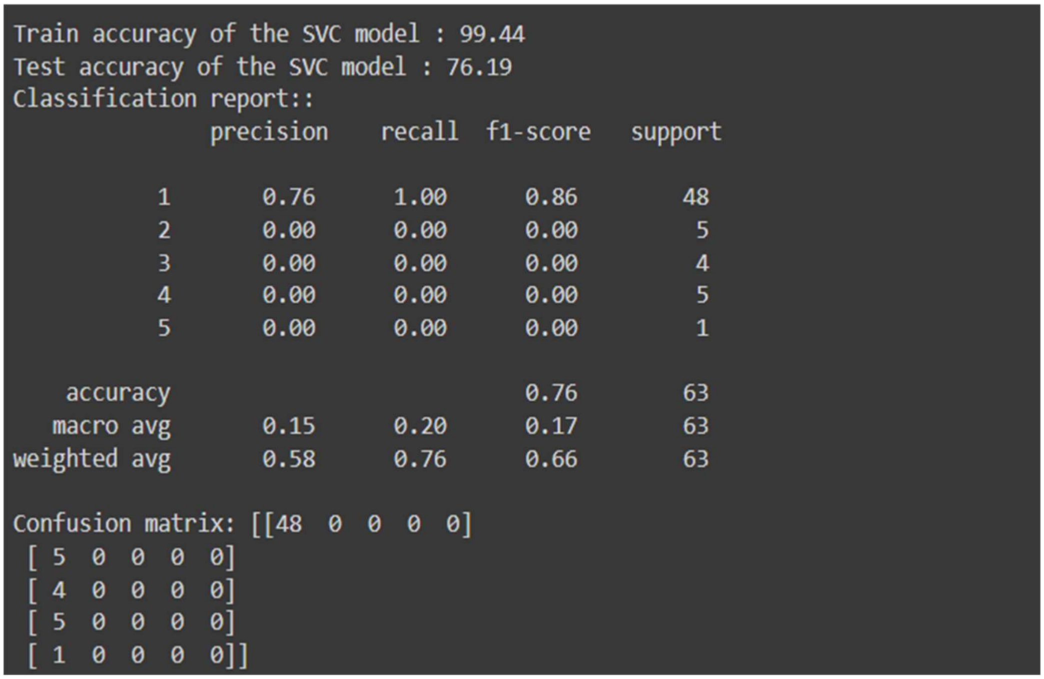
- **ML Models using Bag of Words using Count Vectorizer**
  Models used - Logistic Regression, DecisionTree Classifier, XGB Classifier, Bagging Classifier, GradientBoost Classifier, RandomForest Classifier, SVM Classifier, Naive Bayes Classifier and Bag of Words using Count Vectorizer



	Model	train_accuracy	test_accuracy
0	SVC Clf	0.994366	0.761905
1	RandomForest	0.994366	0.761905
2	XGB Clf	0.938028	0.730159
3	GradientBoost Clf	0.994366	0.746032
4	Bagging Clf	0.946479	0.761905
5	DecisionTree Clf	0.994366	0.666667
6	LogisticRegression Clf	0.994366	0.761905
7	NaiveBayes Clf	0.735211	0.761905

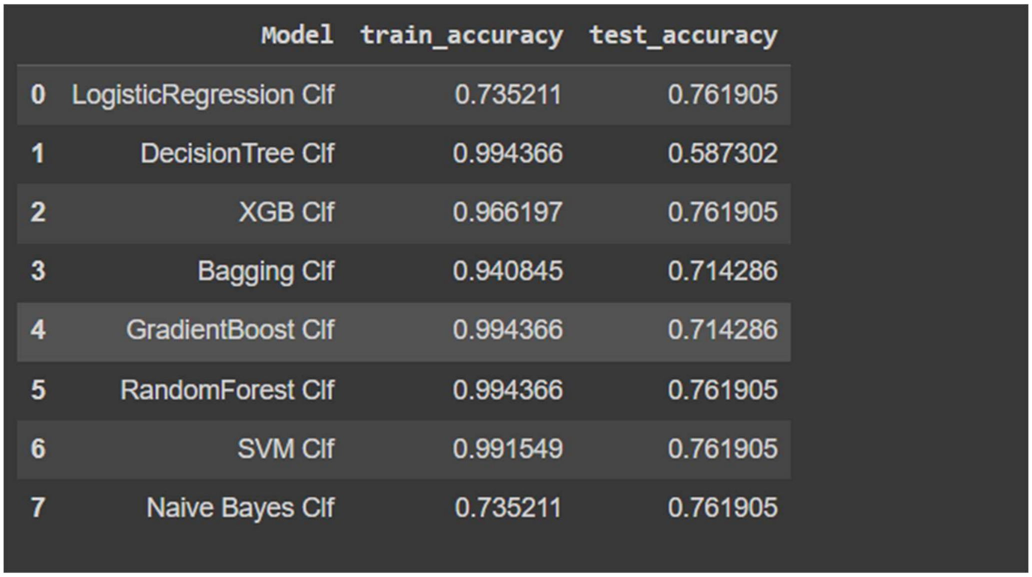
We clearly see that NaiveBayes classifier with best test score and DecisionTree classifier with least test score, where most of them are overfitting.

NaiveBayes Confusion matrix:



```
Train accuracy of the SVC model : 99.44
Test accuracy of the SVC model : 76.19
Classification report::
               precision    recall  f1-score   support

           1       0.76      1.00      0.86        48
           2       0.00      0.00      0.00         5
           3       0.00      0.00      0.00         4
           4       0.00      0.00      0.00         5
           5       0.00      0.00      0.00         1

    accuracy                           0.76        63
   macro avg       0.15      0.20      0.17        63
weighted avg       0.58      0.76      0.66        63

Confusion matrix: [[48  0  0  0  0]
 [ 5  0  0  0  0]
 [ 4  0  0  0  0]
 [ 5  0  0  0  0]
 [ 1  0  0  0  0]]
```

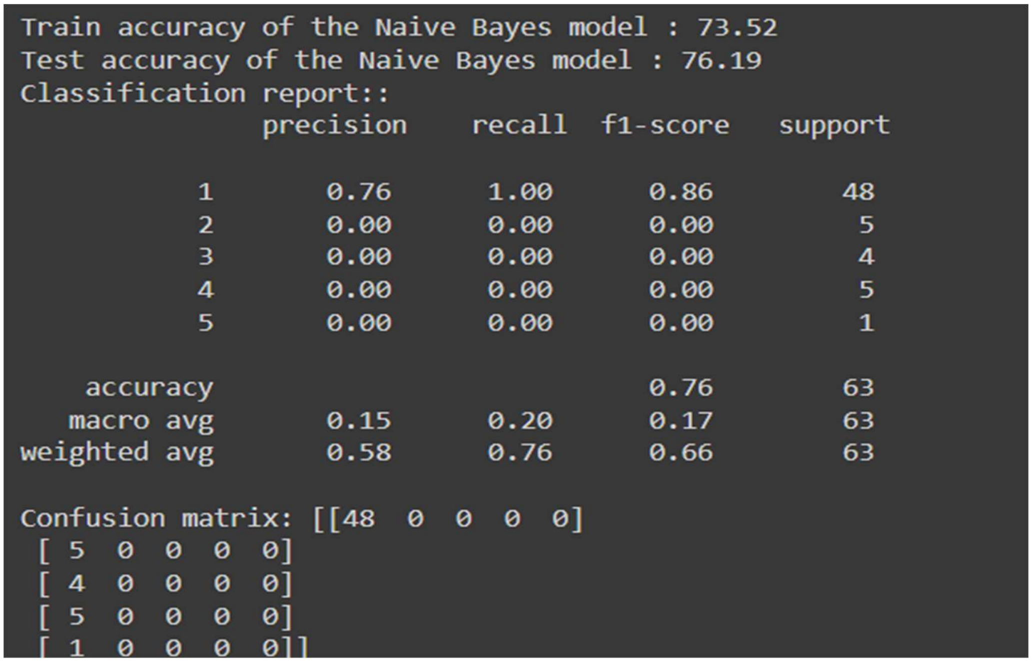
- **ML Models using tfidf Vectorizer.**
  Models used - Logistic Regression, DecisionTree Classifier, XGB Classifier, Bagging Classifier, GradientBoost Classifier, RandomForest Classifier, SVM Classifier, Naive Bayes Classifier and **tfidf Vectorizer**



	Model	train_accuracy	test_accuracy
0	LogisticRegression Clf	0.735211	0.761905
1	DecisionTree Clf	0.994366	0.587302
2	XGB Clf	0.966197	0.761905
3	Bagging Clf	0.940845	0.714286
4	GradientBoost Clf	0.994366	0.714286
5	RandomForest Clf	0.994366	0.761905
6	SVM Clf	0.991549	0.761905
7	Naive Bayes Clf	0.735211	0.761905

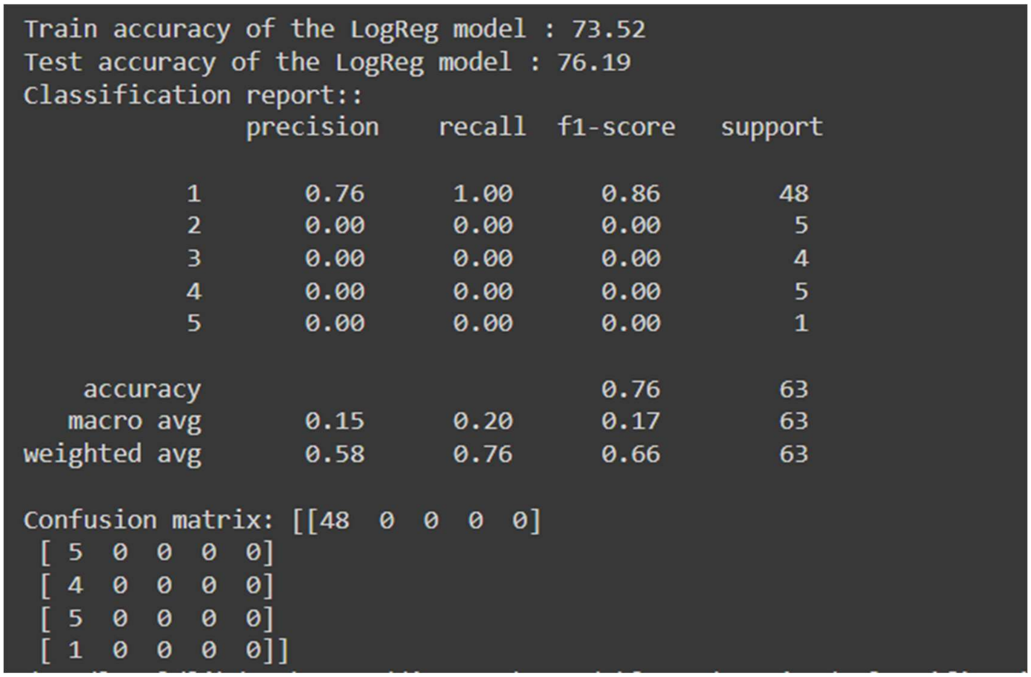
We clearly see that LogisticRegression classifier and NaiveBayes classifier with best test score and executing the same performance whereas DecisionTree classifier with least test score and where most of them are overfitting.

NaiveBayes Confusion matrix:



```
Train accuracy of the Naive Bayes model : 73.52
Test accuracy of the Naive Bayes model : 76.19
Classification report::
               precision    recall  f1-score   support

           1       0.76      1.00      0.86        48
           2       0.00      0.00      0.00         5
           3       0.00      0.00      0.00         4
           4       0.00      0.00      0.00         5
           5       0.00      0.00      0.00         1

    accuracy                           0.76        63
   macro avg       0.15      0.20      0.17        63
weighted avg       0.58      0.76      0.66        63

Confusion matrix: [[48  0  0  0  0]
 [ 5  0  0  0  0]
 [ 4  0  0  0  0]
 [ 5  0  0  0  0]
 [ 1  0  0  0  0]]
```

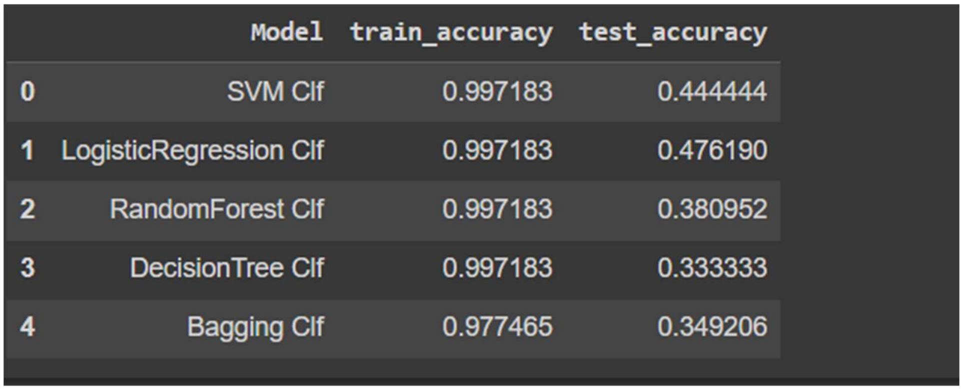
LogisticRegression Confusion matrix:



```
Train accuracy of the LogReg model : 73.52
Test accuracy of the LogReg model : 76.19
Classification report::
               precision    recall  f1-score   support

           1       0.76      1.00      0.86        48
           2       0.00      0.00      0.00         5
           3       0.00      0.00      0.00         4
           4       0.00      0.00      0.00         5
           5       0.00      0.00      0.00         1

    accuracy                           0.76        63
   macro avg       0.15      0.20      0.17        63
weighted avg       0.58      0.76      0.66        63

Confusion matrix: [[48  0  0  0  0]
 [ 5  0  0  0  0]
 [ 4  0  0  0  0]
 [ 5  0  0  0  0]
 [ 1  0  0  0  0]]
```

3.1.1.2 Opting for Potential Accident level as our target variable.

- Modeling- We applied different ML models like SVM Classifier, LogisticRegression Classifier, RandomForest Classifier, DecisionTree Classifier and Bagging Classifier



	Model	train_accuracy	test_accuracy
0	SVM Clf	0.997183	0.444444
1	LogisticRegression Clf	0.997183	0.476190
2	RandomForest Clf	0.997183	0.380952
3	DecisionTree Clf	0.997183	0.333333
4	Bagging Clf	0.977465	0.349206

4. How to improve your model performance

We have run the base model till now, but we will be tuning the models for better accuracy by adjusting the hyperparameters. Hyperparameters are the variables that are set before training a model and can have a significant impact on its performance. Here is a list of some of the hyperparameters we will be tuning for each model:

1. Linear Regression: Regularization parameter (e.g., L1, L2), normalization method

2. Logistic Regression: Regularization parameter (e.g., L1, L2), solver, regularization strength

3. Decision Trees: Maximum depth of tree, minimum number of samples required to split an internal node, minimum number of samples required to be at a leaf node

4. Random Forest: Number of trees, maximum depth of tree, minimum number of samples required to split an internal node, minimum number of samples required to be at a leaf node

5. Gradient Boosting: Number of trees, learning rate, maximum depth of tree

6. Support Vector Machines (SVM): Kernel type (linear, polynomial, radial basis function), regularization parameter, kernel coefficient

7. Naive Bayes: Prior probabilities, smoothing factor

The optimal values of these hyperparameters can be determined through methods such as grid search, random search, or Bayesian optimization. We will be using these methods to find optimal values of the hyperparameters.

We are seeing clear signs of overfitting in many cases, so we are going to deal with it too.

Overfitting occurs when a model is too complex and performs well on the training data but poorly on the validation or test data. Methods we plan to use to manage overfitting in our models:

1. Regularization: Adding a regularization term to the loss function that penalizes the magnitude of the model parameters. For example, L1 or L2 regularization in linear regression or logistic regression.
2. Early Stopping: Monitoring the performance of the model on a validation set during training and stopping the training process when the performance on the validation set starts to deteriorate.
3. Cross-Validation: Dividing the data into multiple folds, training the model on different subsets of the data and evaluating the performance on the remaining data. This gives a more robust estimate of the model’s performance and helps prevent overfitting.
4. Pruning: Removing some of the less important features from the model or reducing the depth of a decision tree or neural network.
5. Ensemble Methods: Combining the predictions of multiple models, such as random forests or gradient boosting, to produce a more robust prediction.

It is also important to keep in mind that increasing the size of the training data can often help reduce overfitting as well. We have limited data in our dataset so we will try to use SMOTE to manage this.

SMOTE (Synthetic Minority Over-sampling Technique) is a popular method for addressing class imbalance in a dataset. Class imbalance occurs when the number of instances in one class (the minority class) is significantly lower compared to the other class (the majority class) – **which is the case in our dataset too.**

This can lead to models that are biased towards the majority class and perform poorly on the minority class – **we are witnessing this with the models we have tried so far.**

SMOTE works by synthesizing new samples for the minority class based on existing samples. This is done by selecting two instances from the minority class, computing the difference between them, and using this difference to create a synthetic instance. This process is repeated until the minority class has a similar number of instances as the majority class.

By over-sampling the minority class, SMOTE aims to balance the distribution of the classes and prevent the model from being biased towards the majority class. It should be noted that while SMOTE can improve model performance on the minority class, it may also introduce overfitting if not used carefully. **So, we will be using cross-validation or other techniques to evaluate the performance of the model on unseen data.**

Seasonality in the accidents – with more accidents in certain industries during a particular time of the year – was found. In the next phase of this modelling exercise, we will further explore the implications of seasonality of accidents on the final model.

We expect that the model system should be able to use this hidden pattern in accident timing to be able to predict and respond to queries more accurately.

References

1. https://www.chatbot.com/chatbot-guide/
2. https://www.forbes.com/sites/cognitiveworld/2020/02/23/choosing-between-rule-based-bots-and-ai-bots/?sh=35b006f0353d
3. https://analyticsindiamag.com/complete-tutorial-on-tkinter-to-deploy-machine-learn
4. https://www.python-engineer.com/posts/chatbot-gui-tkinter/