NATURAL LANGUAGE PROCESSING CHATBOT INTERFACE

CAPSTONE PROJECT GROUP 4 - INTERIM REPORT

2021-2022

SUBMITTED BY

Krishnakant Reddy B

Mahadeva Reddy

Pradeebha Benildus

Shyam Kumar

Vemula Ganesh

Vishal Verma

Project Mentor: Shyam Muralidharan

Great Lakes Institute of Management

CONTENTS

Abstract

Chapters

Abstract

A chatbot is a software or computer program that simulates human conversation or "chatter" through text or voice interactions.

Users in both business-to-consumer (B2C) and business-to-business (B2B) environments increasingly use chatbot virtual assistants to handle simple tasks. Adding chatbot assistants reduces overhead costs, uses support staff time better, and enables organizations to provide customer service during hours when live agents aren't available.

Types of Chatbot-

There are many types of chatbots available, a few of them can be majorly classified as follows:

Text-based chatbot: In a text-based chatbot, a bot answers the user’s questions via a text interface.

Voice-based chatbot: In a voice or speech-based chatbot, a bot answers the user’s questions via a human voice interface.

There are mainly two approaches used to design the chatbots, described as follows:

In a ***Rule-based*** approach, a bot answers questions based on some rules on which it is trained. The rules defined can be very simple to very complex. The bots can handle simple queries but fail to manage complex ones.

Self-learning bots are the ones that use some Machine Learning-based approaches and are definitely more efficient than rule-based bots. These bots can be further classified into two types: Retrieval Based or Generative

There are many types of chatbots available depending on the complexity, a few of them can be majorly classified as follows:

Traditional chatbots: Traditional chatbots are driven by system and automation, mainly through scripts with minimal functionality and the ability to maintain only system context.

Current chatbot: Current chatbots are driven by back and forth communication between the system and humans. They have the ability to maintain both system and task contexts.

Future chatbot: Future chatbots can communicate at multiple levels with automation at the system level. They have the ability to maintain the system, task, and people contexts. There is a possibility of the introduction of master bots and eventually a bot OS.

1. INTRODUCTION

This capstone project is based on designing an ML/DL-based chatbot utility that can help the professionals to highlight the safety risk as per the incident description.

1.1 Problem Statement

Domain: Industry Safety. NLP-based Chatbot.

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 an environment.

Data Description:

This database is basically records of accidents from 12 different plants in 03 different countries where 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 (anonymized)
- ***Local***: the city where the manufacturing plant is located (anonymized)
- **Industry sector*:*** which sector the plant belongs to
- ***Accident level***: from I to VI, it registers how severe was the accident (I mean not severe 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 or female
- ***Employee or Third Party***: if the injured person is an employee or a third party
- ***Critical Risk***: some description of the risk involved in the accident
- ***Description***: Detailed description of how the accident happened.

1.2 Objective

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

1.3 Data sources

Link to download the dataset:

https://drive.google.com/file/d/1_GmrRP1S2OIa02KlfOBNkYa8uxazGbfE/view?usp=sharing

Original dataset link:

https://www.kaggle.com/ihmstefanini/industrial-safety-and-health-analytics-database

2. DATA PRE-PROCESSING AND EXPLORATORY DATA ANALYSIS

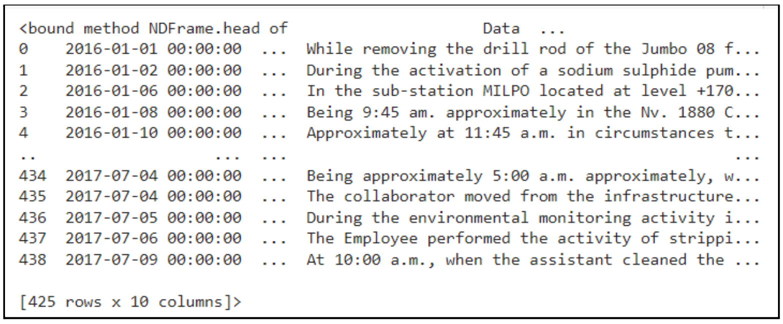
2.1 Data Pre-Processing

Data preprocessing is a data mining technique that is used to transform the raw data in a useful and efficient format.

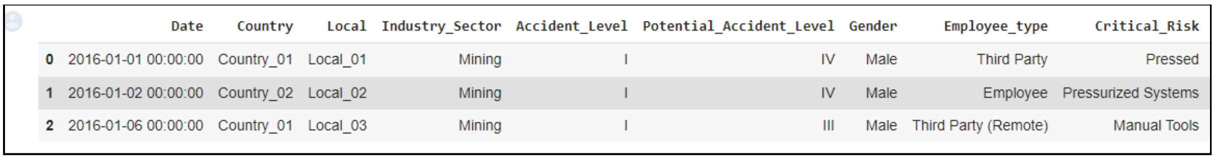
2.1.1. Importing data and Re-naming columns:

We Imported the Dataset from the given CSV file and the size of the dataset is 425 * 10. The Column names of imported data are not in the prescribed format for python libraries. So renaming it with the proper names.

Input data -"DataSet-industrial_safety_and_health_database_with_accidents_description.csv"



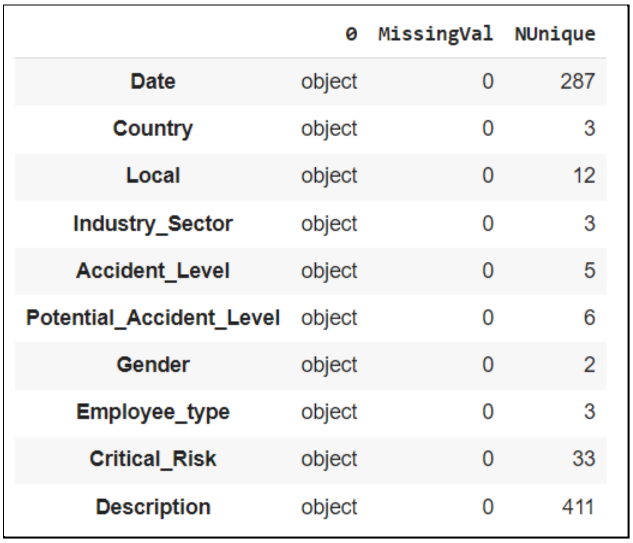
```
<bound method NDFrame.head of                          Data  ...
0    2016-01-01 00:00:00  ...  While removing the drill rod of the Jumbo 08 f...
1    2016-01-02 00:00:00  ...  During the activation of a sodium sulphide pum...
2    2016-01-06 00:00:00  ...  In the sub-station MILPO located at level +170...
3    2016-01-08 00:00:00  ...  Being 9:45 am. approximately in the Nv. 1880 C...
4    2016-01-10 00:00:00  ...  Approximately at 11:45 a.m. in circumstances t...
..                   ...  ...                                               ...
434  2017-07-04 00:00:00  ...  Being approximately 5:00 a.m. approximately, w...
435  2017-07-04 00:00:00  ...  The collaborator moved from the infrastructure...
436  2017-07-05 00:00:00  ...  During the environmental monitoring activity i...
437  2017-07-06 00:00:00  ...  The Employee performed the activity of strippi...
438  2017-07-09 00:00:00  ...  At 10:00 a.m., when the assistant cleaned the ...

[425 rows x 10 columns]>
```



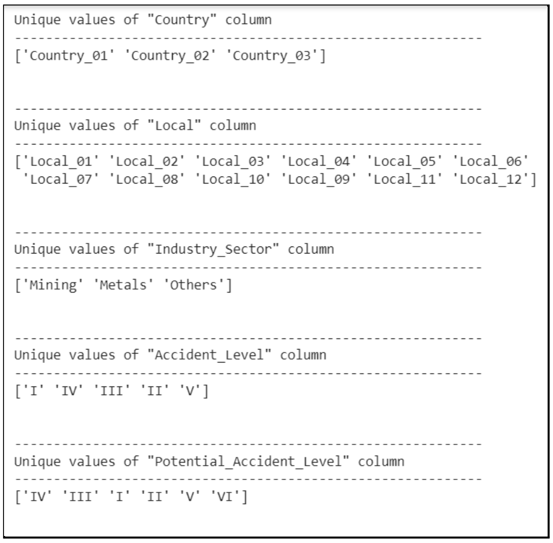
	Date	Country	Local	Industry_Sector	Accident_Level	Potential_Accident_Level	Gender	Employee_type	Critical_Risk
0	2016-01-01 00:00:00	Country_01	Local_01	Mining	I	IV	Male	Third Party	Pressed
1	2016-01-02 00:00:00	Country_02	Local_02	Mining	I	IV	Male	Employee	Pressurized Systems
2	2016-01-06 00:00:00	Country_01	Local_03	Mining	I	III	Male	Third Party (Remote)	Manual Tools

2.1.2. Checking Missing Values and outliers:

By Checking the missing values we found there are no null values and outliers in the dataset. We have only 3 countries, 12 Locals, 3 Industry sectors, 5 accident levels, 6 potential accident levels, 3 Employee types, and 33 Critical Risks including Not applicable.



	0	MissingVal	NUnique
Date	object	0	287
Country	object	0	3
Local	object	0	12
Industry_Sector	object	0	3
Accident_Level	object	0	5
Potential_Accident_Level	object	0	6
Gender	object	0	2
Employee_type	object	0	3
Critical_Risk	object	0	33
Description	object	0	411



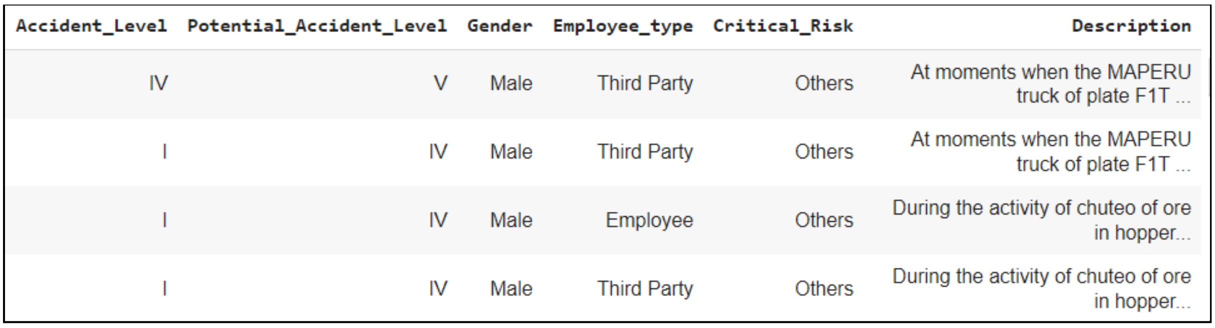
```
Unique values of "Country" column
--------------------------------------------------------------
['Country_01' 'Country_02' 'Country_03']

--------------------------------------------------------------
Unique values of "Local" column
--------------------------------------------------------------
['Local_01' 'Local_02' 'Local_03' 'Local_04' 'Local_05' 'Local_06'
 'Local_07' 'Local_08' 'Local_10' 'Local_09' 'Local_11' 'Local_12']

--------------------------------------------------------------
Unique values of "Industry_Sector" column
--------------------------------------------------------------
['Mining' 'Metals' 'Others']

--------------------------------------------------------------
Unique values of "Accident_Level" column
--------------------------------------------------------------
['I' 'IV' 'III' 'II' 'V']

--------------------------------------------------------------
Unique values of "Potential_Accident_Level" column
--------------------------------------------------------------
['IV' 'III' 'I' 'II' 'V' 'VI']
```

In Accident Level and Potential Accident Level, Five types of accident levels (1 to 5) are present. But, Six types of Potential Accident Levels (1 to 6) are there and there is only one value registered under 'Potential Accident level 6'. So Replace it with level 5.

2.1.3. Removing Duplicates:

By Checking the Duplicates, we have 7 duplicate values on the entire dataset, let us drop them.

By Checking the 'Description' Column alone as a subset we can see 14 Duplicates since the unique value in the Description is 411 and the total value is 418 after removing duplicates. So dropping the second instance(7 rows).



Accident_Level	Potential_Accident_Level	Gender	Employee_type	Critical_Risk	Description
IV	V	Male	Third Party	Others	At moments when the MAPERU truck of plate F1T ...
I	IV	Male	Third Party	Others	At moments when the MAPERU truck of plate F1T ...
I	IV	Male	Employee	Others	During the activity of chuteo of ore in hopper...
I	IV	Male	Third Party	Others	During the activity of chuteo of ore in hopper...

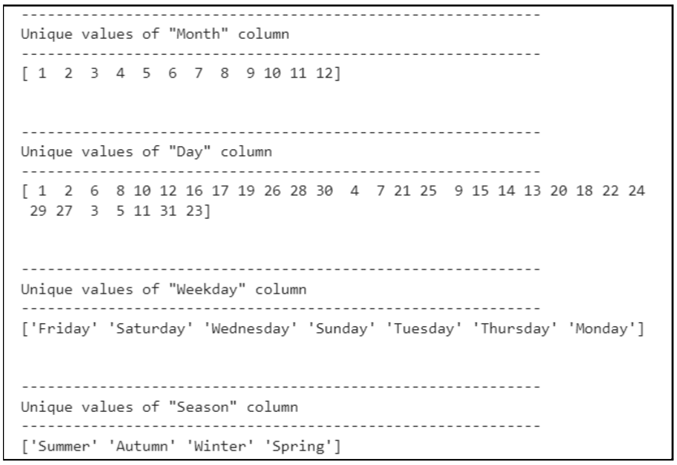
After removing the duplicates the size of the dataset is 411*10.

2.1.4. PreProcess - Time Series Data:

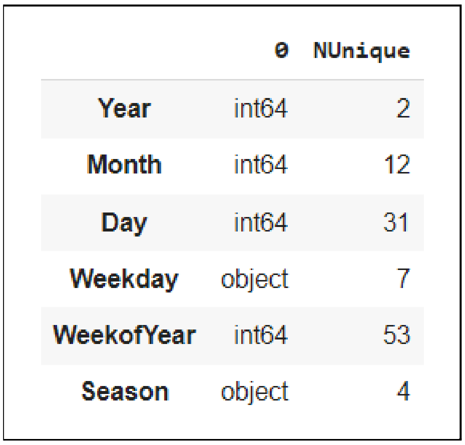
The Entire data was captured between January 2016 - September 2017. We split the Date columns into a date, month, year, day, weekday, and week of the year. The Countries where the dataset was collected are anonymized but they are all located in South America - Brazil. Brazil has four climatological seasons as below.

- ➢ Spring: September to November
- ➢ Summer: December to February
- ➢ Autumn: March to May
- ➢ Winter: June to August

Based on the month variable we created a new feature called a season. Then we checked unique values of all the time variables, based on the result we can conclude the accidents happen all days and months of the year.



```
------------------------------------------------------------
Unique values of "Month" column
------------------------------------------------------------
[ 1  2  3  4  5  6  7  8  9 10 11 12]


------------------------------------------------------------
Unique values of "Day" column
------------------------------------------------------------
[ 1  2  6  8 10 12 16 17 19 26 28 30  4  7 21 25  9 15 14 13 20 18 22 24
 29 27  3  5 11 31 23]


------------------------------------------------------------
Unique values of "Weekday" column
------------------------------------------------------------
['Friday' 'Saturday' 'Wednesday' 'Sunday' 'Tuesday' 'Thursday' 'Monday']


------------------------------------------------------------
Unique values of "Season" column
------------------------------------------------------------
['Summer' 'Autumn' 'Winter' 'Spring']
```



	0	NUnique
Year	int64	2
Month	int64	12
Day	int64	31
Weekday	object	7
WeekofYear	int64	53
Season	object	4

2.1.5.Text PreProcessing:

Text preprocessing is a method to clean the text data and make it ready to feed data to the model. Text data contains noise in various forms like emotions, punctuation, text in different cases.

For Text Preprocessing we created a separate python file called NLP_text_preprocess.py file and a Class called PreProcessing in it. We have several steps that need to be done for text Preprocessing. These steps are passed as arguments that can be controlled with the help of the config.py file which has all the step names as Boolean.

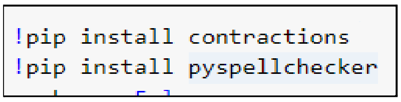
```
from NLP_text_preprocess import PreProcessing
```

We need to call our python file like above in our main file and the NLP_Text_preprocess file and config file need to be maintained in the same folder. In order to use this NLP_Text_preprocess we need to import two libraries beforehand, we are importing it.

The Steps Followed in Text Preprocessing in same order:

- ➢ Convert into Lower case
- ➢ Removing the URL
- ➢ Removing the Special characters
- ➢ Remove Time Formats
- ➢ Expand Contradictions
- ➢ Removing Punctuation
- ➢ Removing Whitespaces
- ➢ Check the spelling mistakes
- ➢ Removing the Stop words
- ➢ Convert the words into lemma form

Libraries to install beforehand



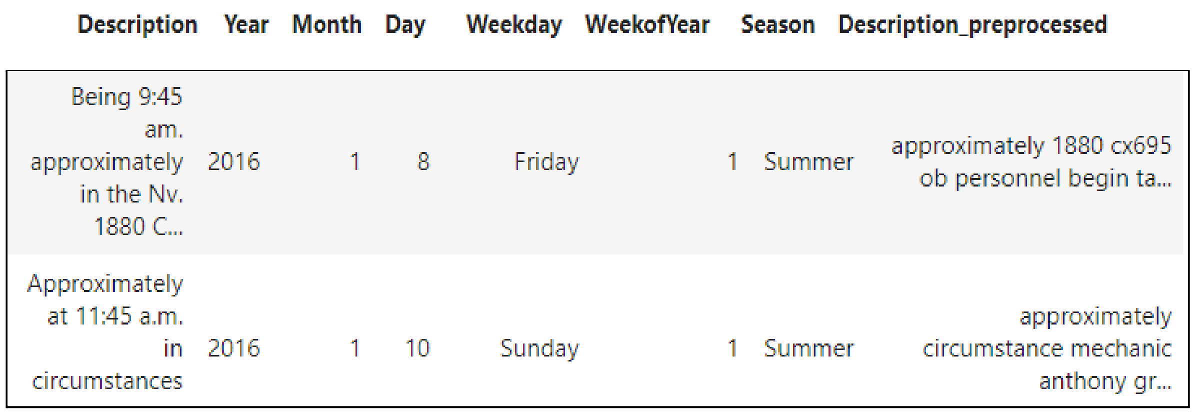
```
!pip install contractions
!pip install pyspellchecker
```

Special Characters - 'å¼«¥ªº©ð±§µæ¹¢³¿®ä£' these characters are removed in our data.
Time Formats - 10:20.AM These time formats are not needed for our analysis, so we removed them.

Expand Contradictions - Contractions are words or combinations of words that are shortened by dropping letters and replacing them with an apostrophe.
Example:- Was not will be converted into wasn't. We are expanding the Contradictions.
Spelling Mistakes - We are using the Spell Checker library for commonly misspelled words.

Stop Words- Stop words are a set of commonly used words in a language. But, in our dataset, some stop words might play a vital role so we are not removing those words.
not_stop_words = ["not","nor", "after","before","above", "below","between"]

Lemmatization - Lemmatization usually refers to doing things properly with the use of a vocabulary and morphological analysis of words, normally aiming to remove inflectional endings only and to return the base or dictionary form of a word, which is known as the lemma.
We use the WordNetLemmatizer library to convert the word into lemma form.
For Performing some operations we need to convert the sentences into tokens. After doing all the preprocessing steps we convert the tokens back into sentences and we are created a new column 'Description_Preprocessed' to have processed data.

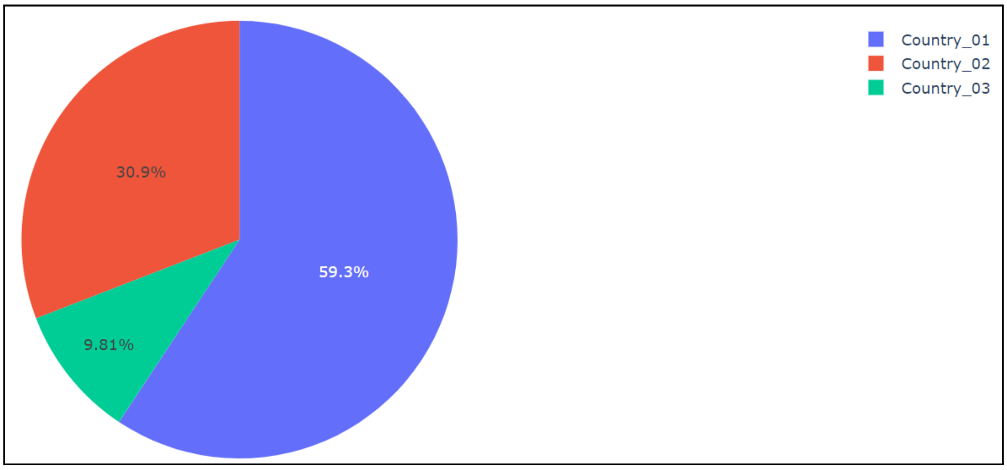


Description	Year	Month	Day	Weekday	WeekofYear	Season	Description_preprocessed
Being 9:45 am. approximately in the Nv. 1880 C...	2016	1	8	Friday	1	Summer	approximately 1880 cx695 ob personnel begin ta...
Approximately at 11:45 a.m. in circumstances	2016	1	10	Sunday	1	Summer	approximately circumstance mechanic anthony gr...

2.2. Exploratory Data Analysis

Exploratory Data Analysis refers to the critical process of performing initial investigations on data so as to discover patterns,to spot anomalies,to test hypotheses and to check assumptions with the help of summary statistics and graphical representations.

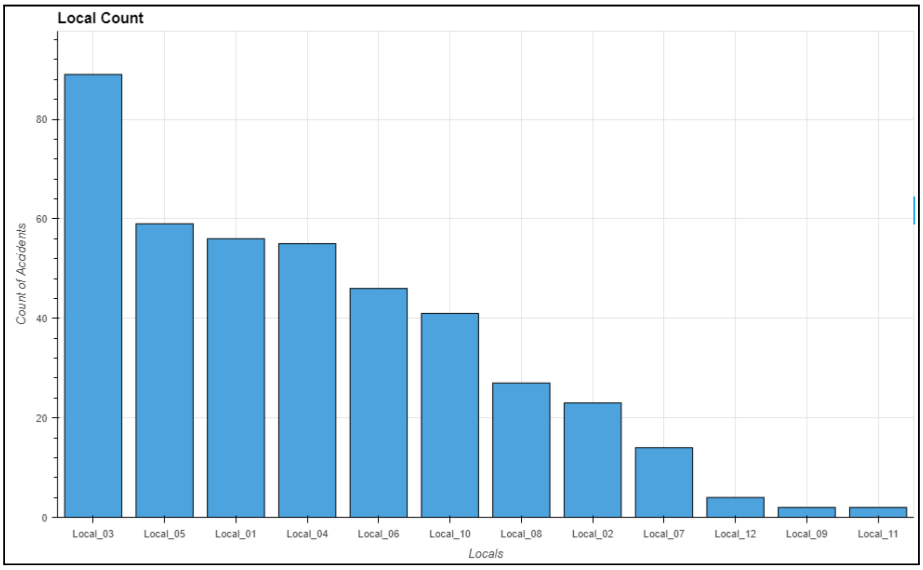
2.2.1. Distribution of Accidents Based on Country Level



In the given dataset, we have three countries, Country_01, Country_02, and Country_03. The proportion of accidents based on Country level are as follows:

1. **Country_01**: Accidents- 248 i.e. **59.0%**
2. **Country_02**: Accidents- 129 i.e. **31.0%**
3. **Country_03**: Accidents- 41 i.e. **10.0%**

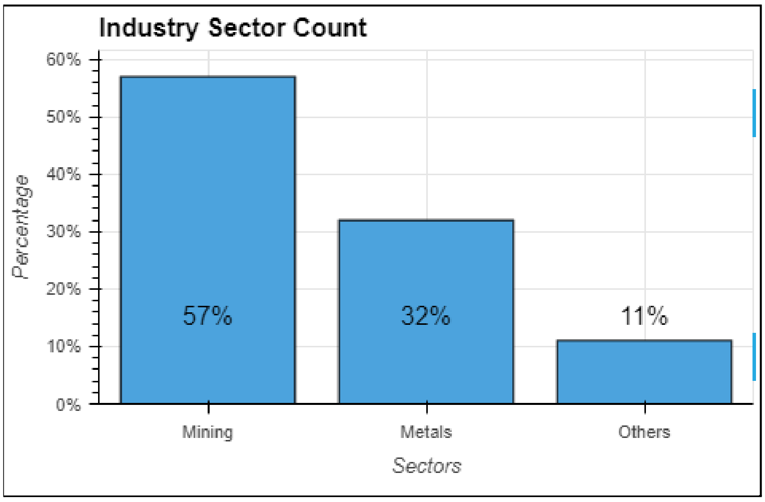
2.2.2. Distribution of Manufacturing Plants Based on Local



In the given dataset, we have a total of twelve locals where the manufacturing plants are located. The distribution of the manufacturing plants based on Local is shown in the above plot and the observations are as follows:

1. **Maximum** number of manufacturing plants are located in **Local_03**
2. **Lowest** number of manufacturing plants are located in **Local_09**

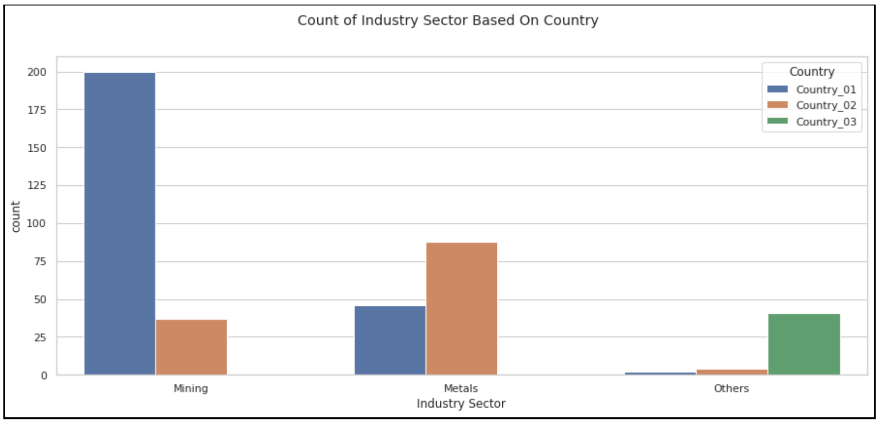
2.2.3. Distribution of Industry Based on Sector



The manufacturing plants are distributed into three types of Industry Sector, Mining, Metals, and Others. The distribution of these industries based on the sector is shown in the above plot and the observations are as follows:

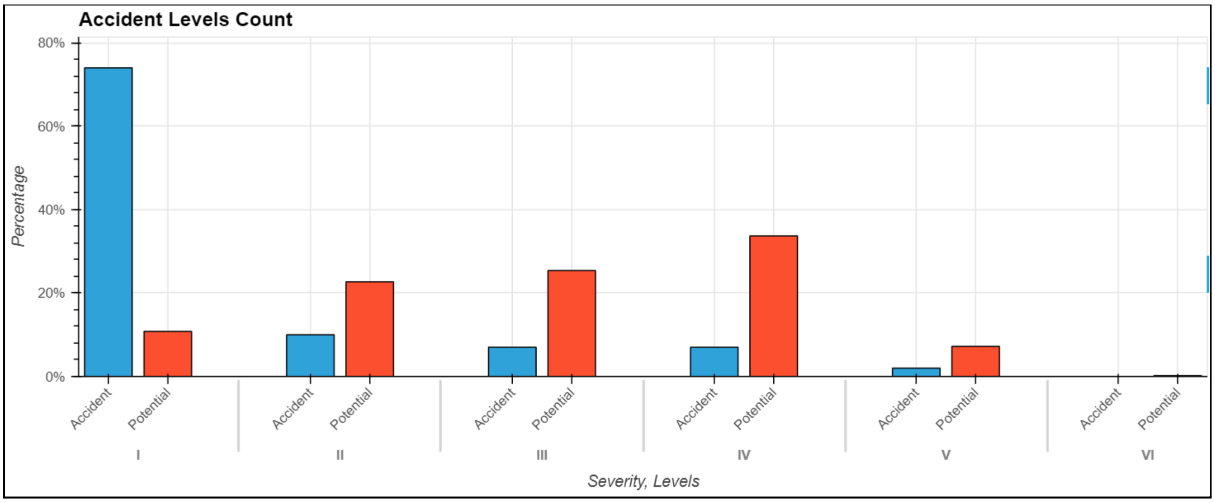
1. **Mining Sector**: Industry Count- 237 i.e. 57.0%
2. **Metals Sector**: Industry Count- 134 i.e. 32.0%
3. **Others Sector**: Industry Count- 47 i.e. 11.0%

2.2.4. Distribution of Industries Based on Country and Sector



The above distribution shows that Country_01 and Country_02 have only Mining and Metals sectors whereas Country_03 has manufacturing plants from all the three sectors.

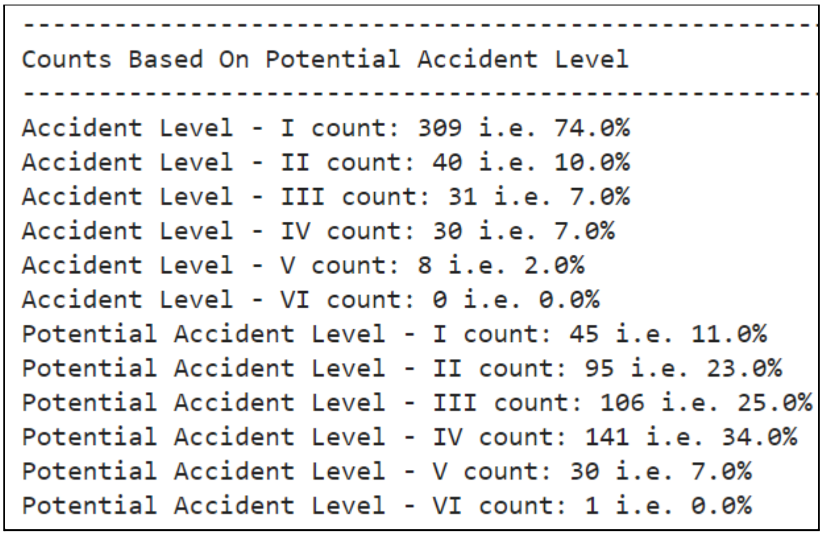
2.2.5. Distribution of the Accident Level & Potential Accident Level



In the given dataset, the level of accidents is distributed into five groups and for each accident level, we have potential accident levels distributed into six groups.

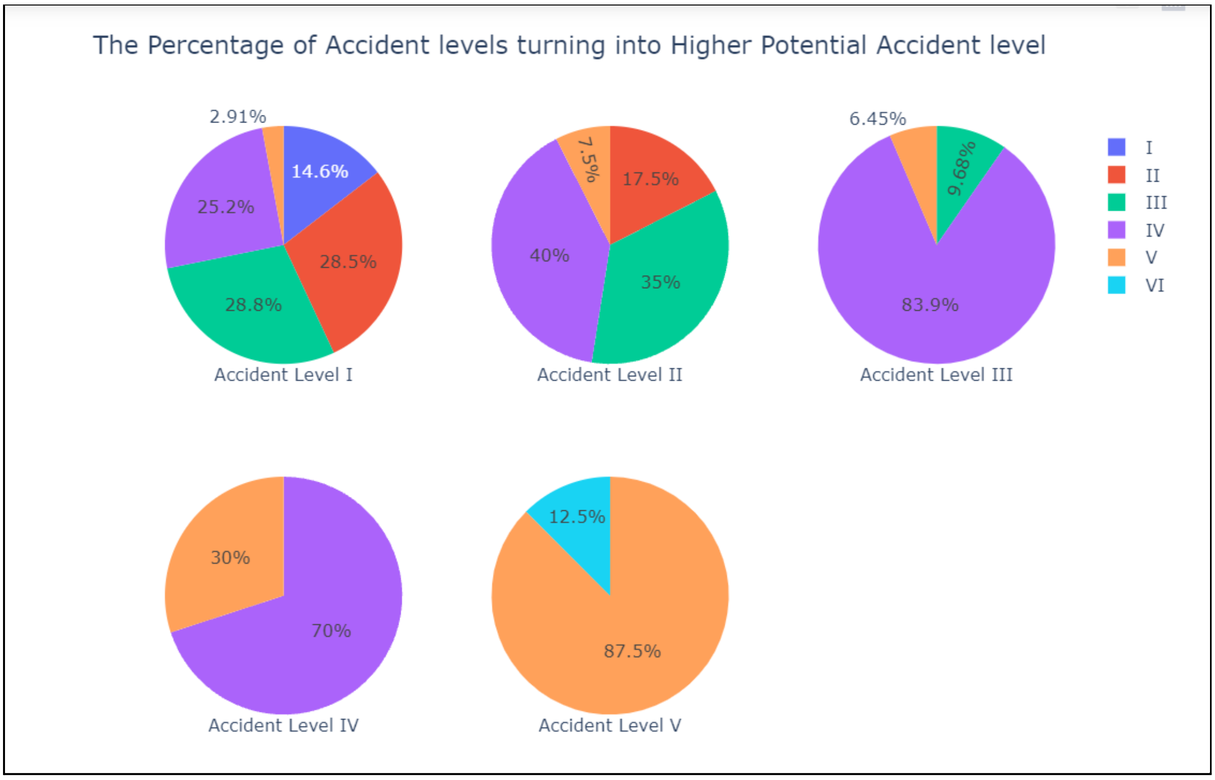
The observations from the above plot are as follows:

1. The number of accidents decreases as the Severity Level of Accident increases
2. The number of accidents increases as the Potential Accident Level increases



```
-----------------------------------------------------
Counts Based On Potential Accident Level
-----------------------------------------------------
Accident Level - I count: 309 i.e. 74.0%
Accident Level - II count: 40 i.e. 10.0%
Accident Level - III count: 31 i.e. 7.0%
Accident Level - IV count: 30 i.e. 7.0%
Accident Level - V count: 8 i.e. 2.0%
Accident Level - VI count: 0 i.e. 0.0%
Potential Accident Level - I count: 45 i.e. 11.0%
Potential Accident Level - II count: 95 i.e. 23.0%
Potential Accident Level - III count: 106 i.e. 25.0%
Potential Accident Level - IV count: 141 i.e. 34.0%
Potential Accident Level - V count: 30 i.e. 7.0%
Potential Accident Level - VI count: 1 i.e. 0.0%
```

2.2.6. The possibilities of lower-level accidents turning into higher-level accidents



These charts show the probabilities of any accident level turning into higher potential level accidents:

1. When the accident level is identified as level 1, there are higher chances (100-14.6 = 85.4%) of it turning into a potential accident level greater than 1, which should be taken care

2. For the accident level 2 also, it has higher chances (100-17.5=82.5%) of turning into major accidents

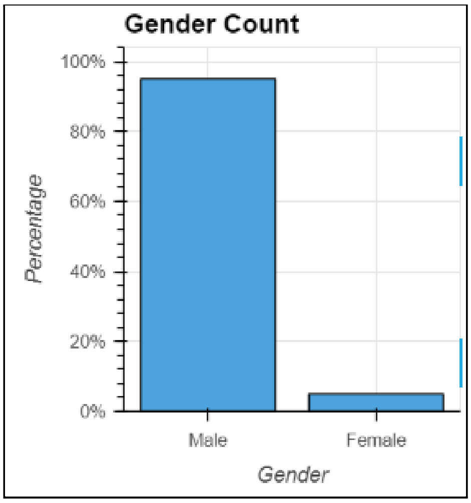
3. When the accident level is identified as 3, it is more likely to turn into higher Potential levels (100-9.68)=90.32

4. For the accident level 4, there is less chance (30%, not actually very less) to turn into higher potential level accidents

5. For the accident level 5, there is less chance (12.5%) to turn into higher potential level accidents

It can be concluded that safety measures have to be taken care of to make the lower level (I, II, III) accidents not turn into major accidents as there are high chances of 85.4%, 82.5%, and 90.32%

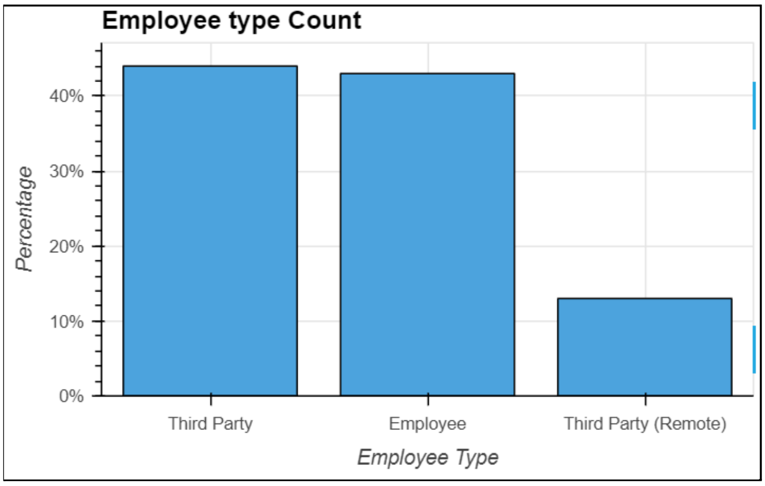
2.2.7. Distribution of Accidents Based on Gender



From the above plot, we have observed that the proportion of Female workers is very less as compared to the Male workers.

1. **Male** Count- 396 i.e. **95.0%**
2. **Female** Count- 22 i.e. **5.0%**

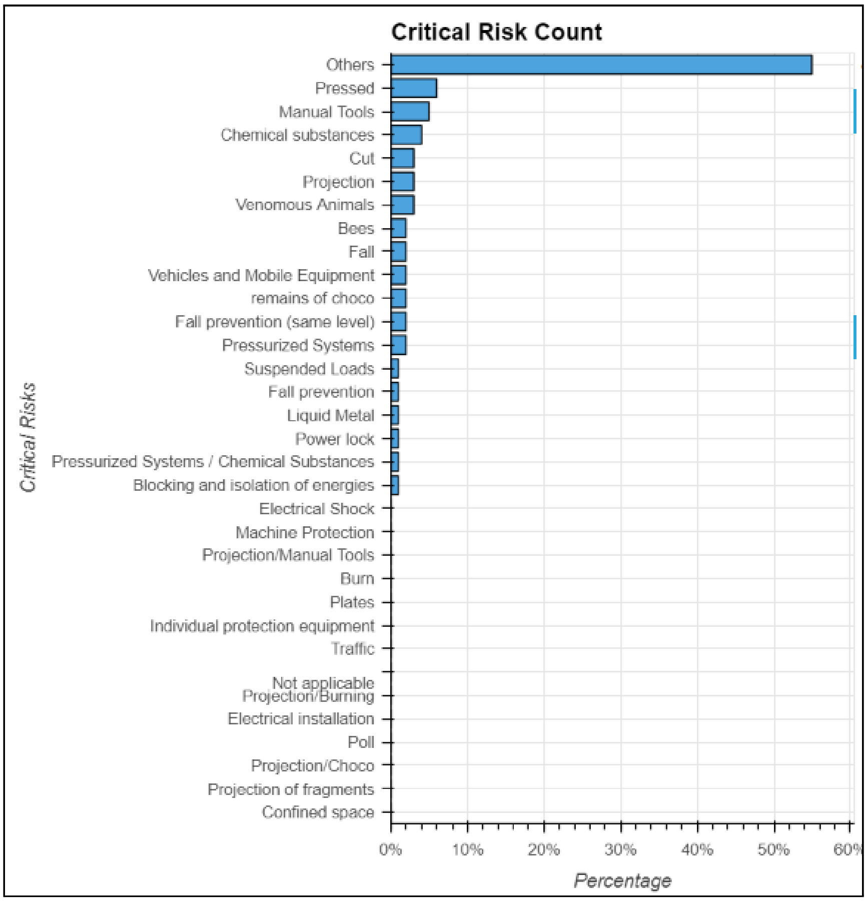
2.2.8. Distribution of Employee Based on Employee Type



The maximum number of employees is from the Third-party and the minimum is from the Third Party (Remote).

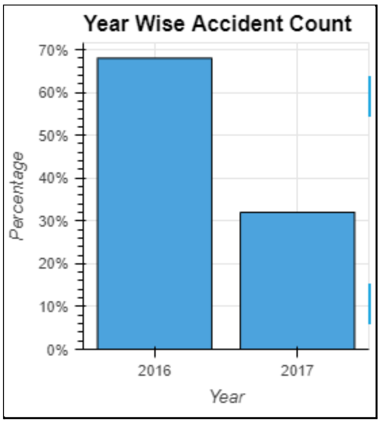
1. **44.0%** out of the total employees are from **Third Party** type
2. **43.0%** out of the total employees are from **Employee** type
3. **13.0%** out of the total employees are from **Third Party (Remote)** type

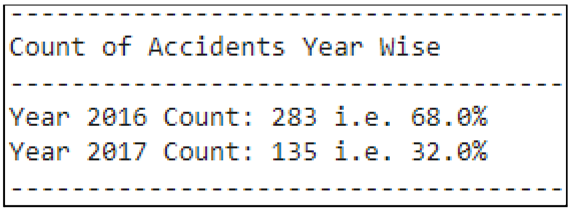
2.2.9. Distribution of Accidents Based on Critical Risk



The maximum number of accidents i.e., 55.0% of the total is falling under the Critical Risk 'Others'. Apart from 'Others', the second and third most Critical Risk counts are from 'Pressed' and 'Manual Tools'.

2.2.10. Accidents Counts Based on Year

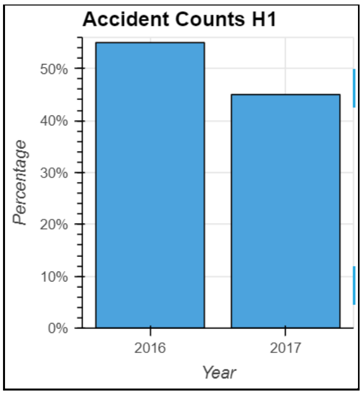


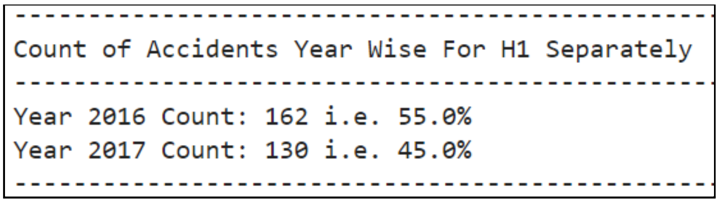


```
--------------------------------------
Count of Accidents Year Wise
--------------------------------------
Year 2016 Count: 283 i.e. 68.0%
Year 2017 Count: 135 i.e. 32.0%
--------------------------------------
```

The maximum number of accidents as per the given dataset is recorded in the year 2016 as compared to 2017. Although, this has to be noted that in the given dataset, data is not available from August-2017.

Comparing data for H1 only for 2016 and 2017.

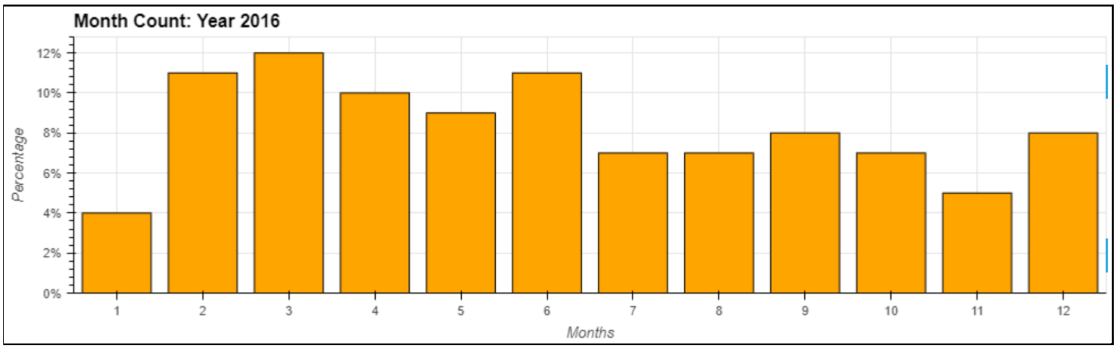




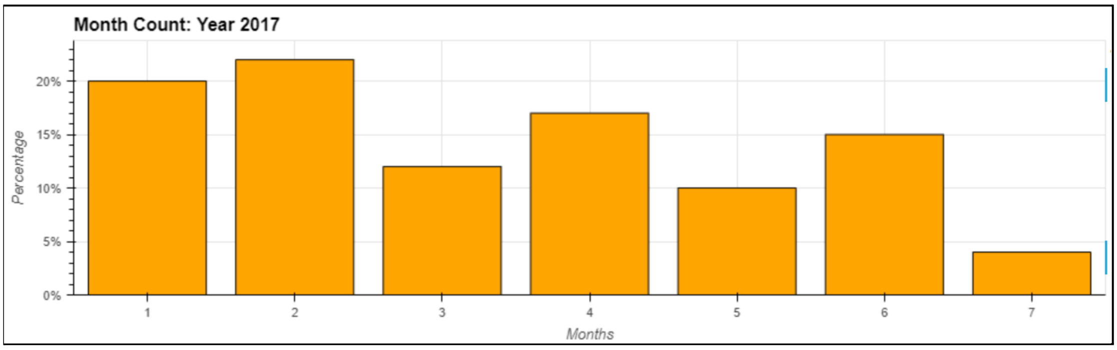
```
-------------------------------------------------
Count of Accidents Year Wise For H1 Separately
-------------------------------------------------
Year 2016 Count: 162 i.e. 55.0%
Year 2017 Count: 130 i.e. 45.0%
-------------------------------------------------
```

On comparing the accidents count for the first six months for both 2016 and 2017, we have observed that the maximum number of accidents recorded is from the year 2016 as compared to 2017. The counts have decreased from last year by **10%**.

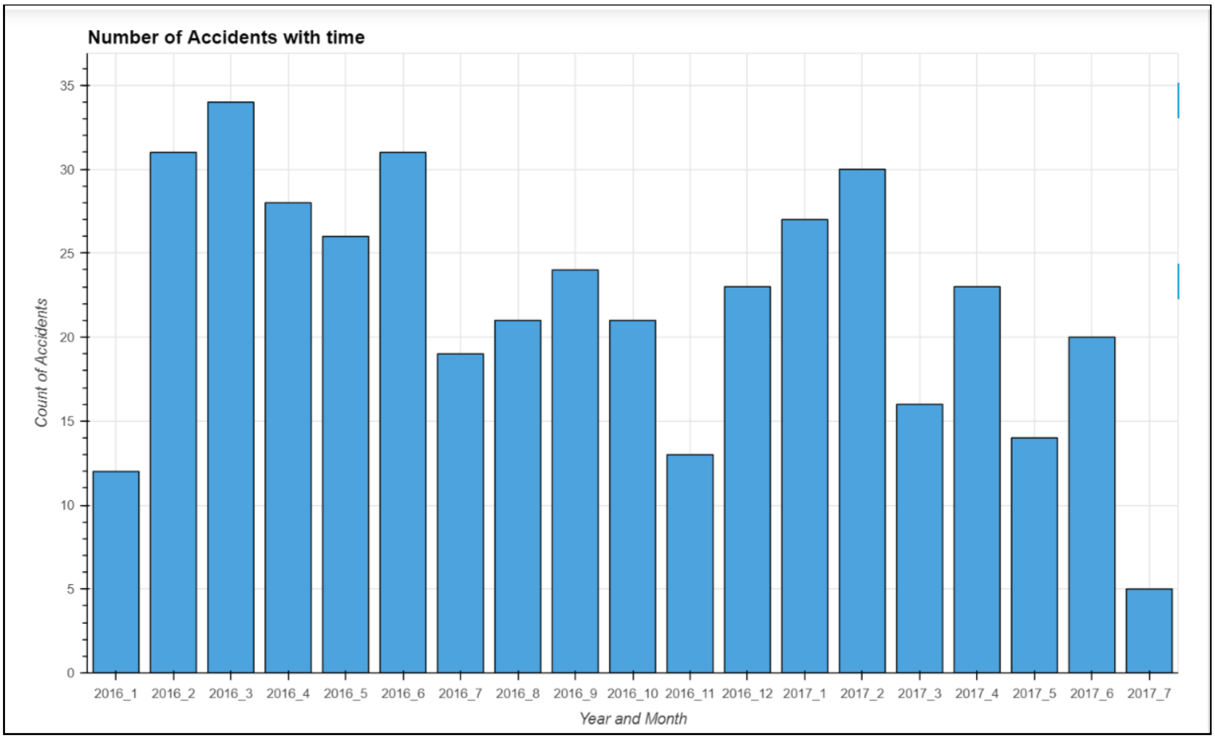
2.2.11.Distribution of Accidents Based on Month For 2016 and 2017 Separately



In the year 2016, the maximum number of accidents recorded in the first six months as compared to the last six months. Also, the maximum accidents recorded in the month of March and minimum in January.

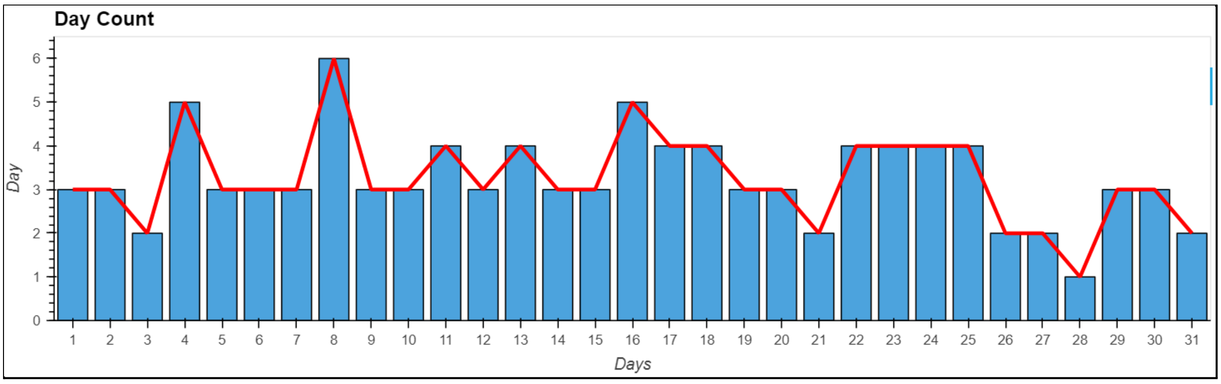


For the year 2017, if we compare data for the first six months, we have observed that the maximum number of accidents recorded in January and minimum in June.



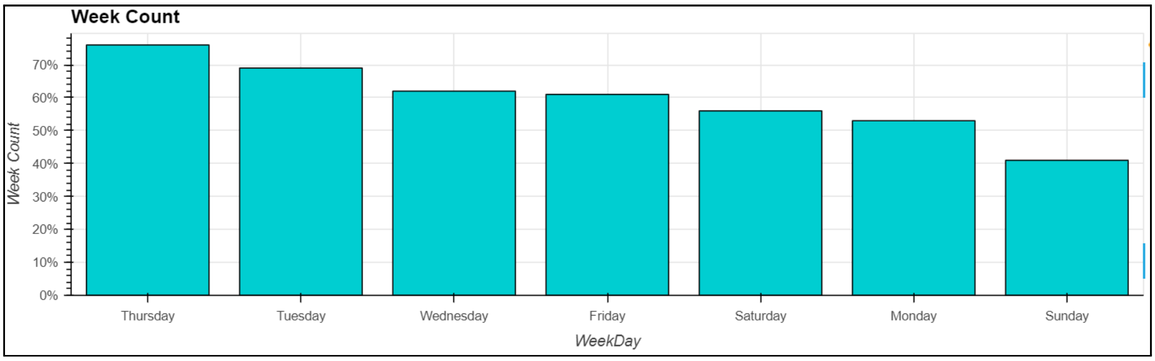
- In the first half of both years, the number of accidents is more.
- But as the number of accidents is reducing with time (2016 to 2017), the 2017 first half has fewer accidents compared to 2016 first half
- The number of accidents in the years 2016 and 2017, is fluctuating. But overall it is decreasing towards the end of the year. It can be concluded that precautions are taken to reduce accidents.

2.2.12. Count of Accidents Based on Days



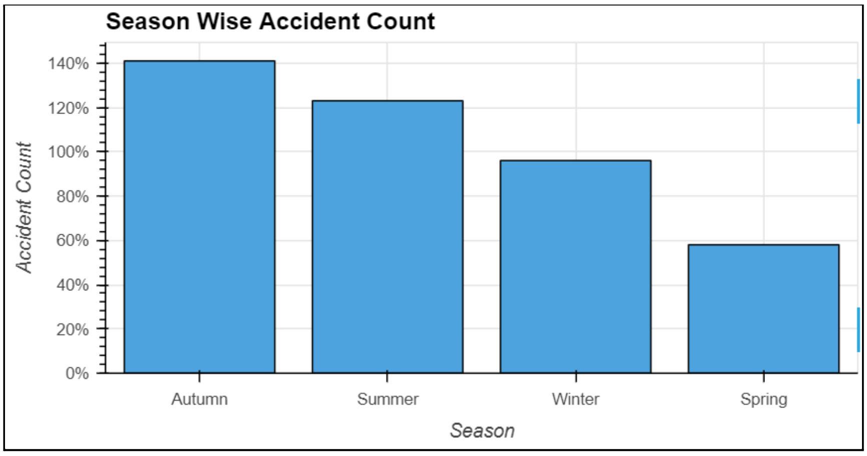
The maximum number of accidents was recorded on the 4th, 8th, and 16th day of each month.

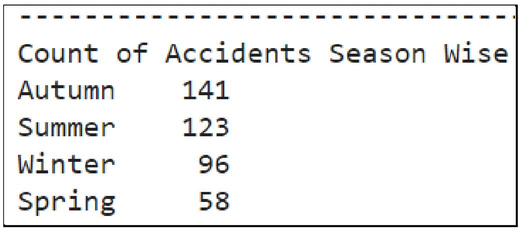
2.2.13. Count of Accidents Based on WeekDay



The maximum number of accidents increased during the middle of the week and declined after the middle of the week. Also, we have observed that accidents occurred on weekends also.

2.2.14. Count of Accidents Based on Season

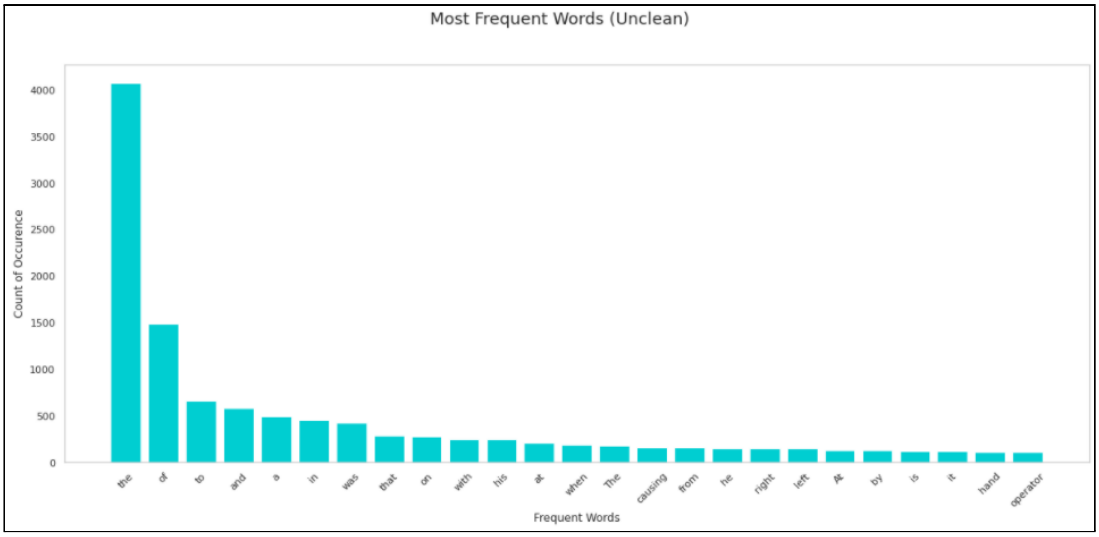




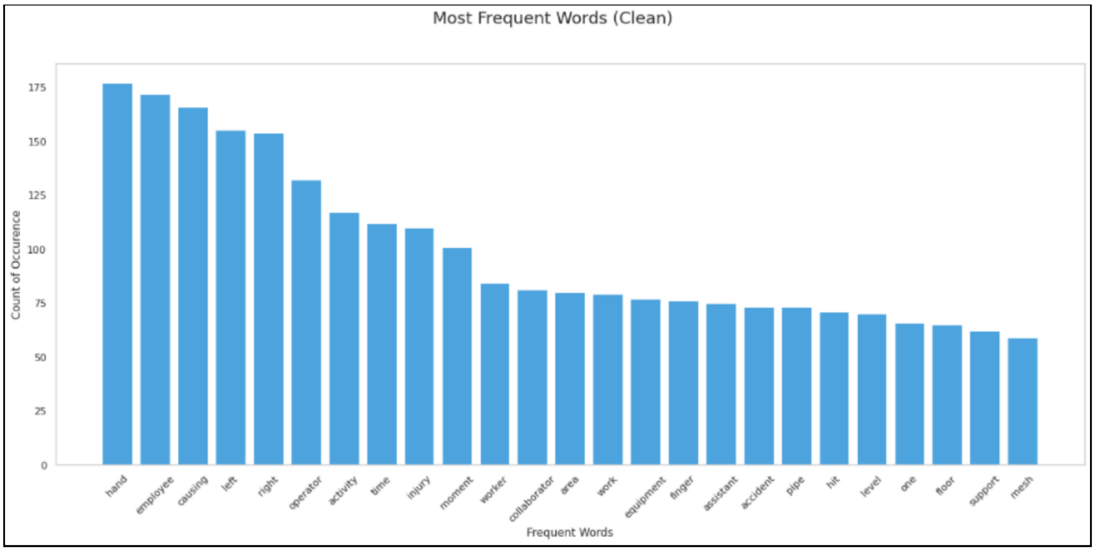
```
------------------------------
Count of Accidents Season Wise
Autumn    141
Summer    123
Winter     96
Spring     58
```

The maximum number of accidents reported in the Autumn and Summer seasons and the minimum in the Spring season. The occurrence of accidents is related to the climate (especially temperature).

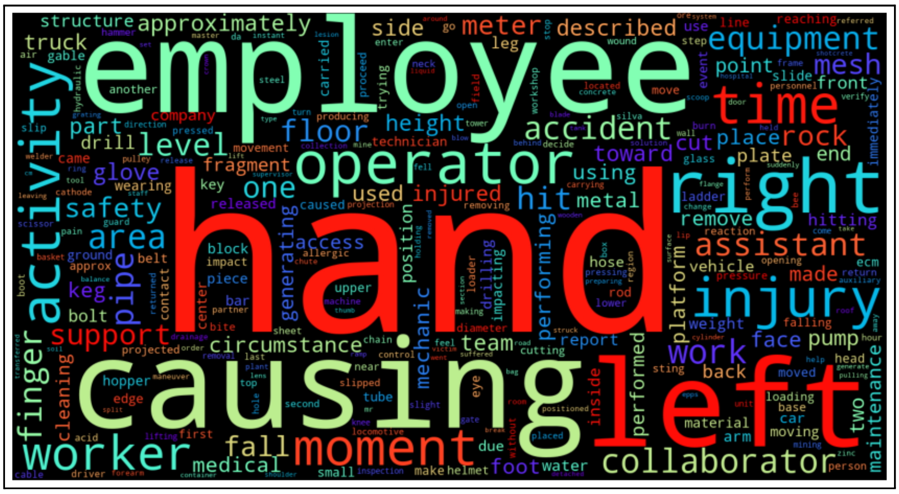
2.2.15. Most Frequent Words in the Description (Unclean and Cleaned Data).



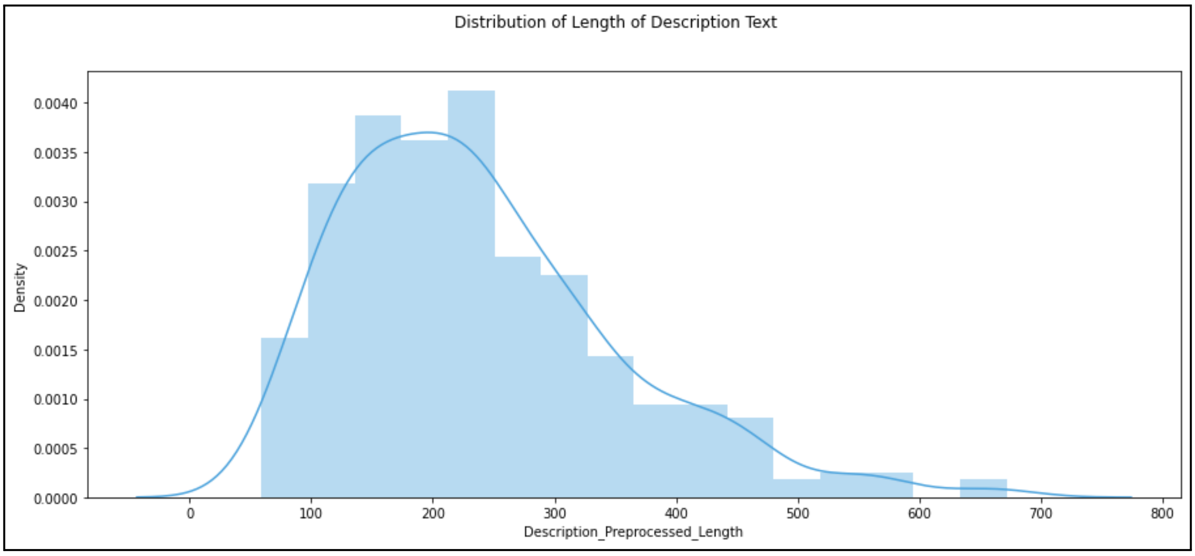
In the unclean data, the top three most frequent words are ‘the’, ’of’, and ’to’.



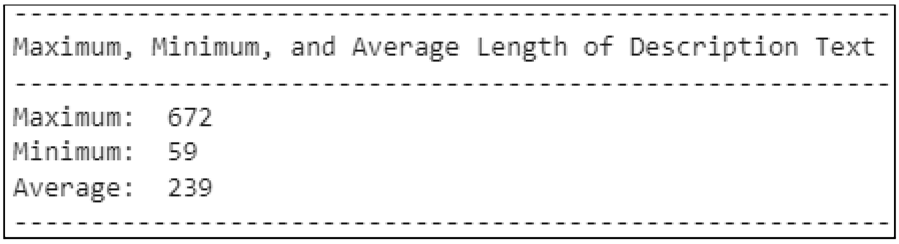
In the processed data, the top three most frequent words are ‘hand’, ’employee’, and ’causing’. The above three words can also be visualized using the Word Cloud.



2.2.16. Distribution of Length of Description Text.

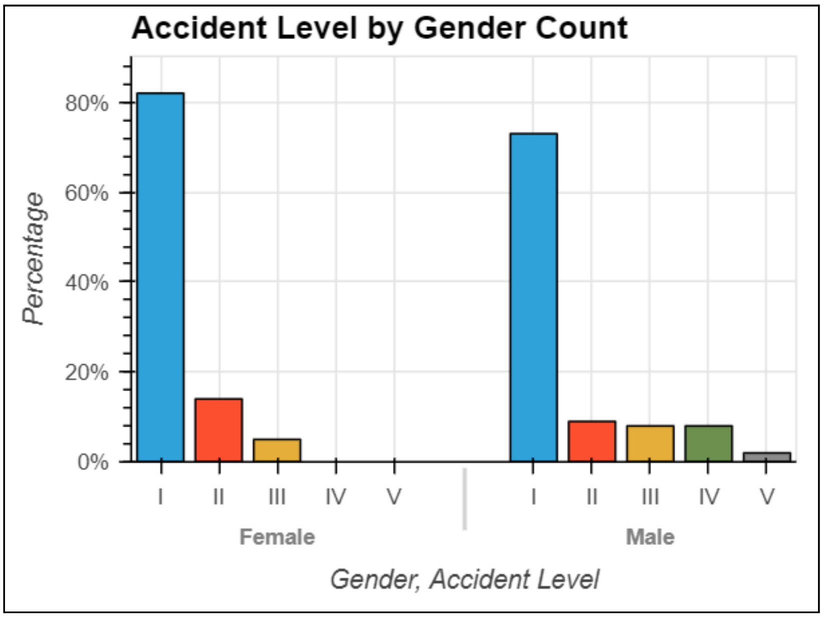


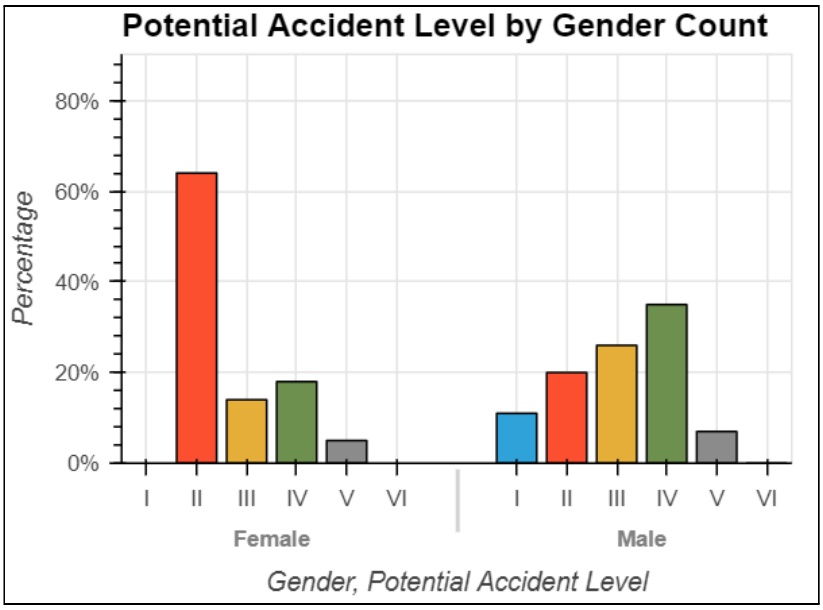
The distribution of the length of words is shown in the above distribution plot. We have observed that the distribution is positively skewed and most of the text is of the length of 200 to 300.



```
-----------------------------------------------------------
Maximum, Minimum, and Average Length of Description Text
-----------------------------------------------------------
Maximum:  672
Minimum:  59
Average:  239
-----------------------------------------------------------
```

2.2.17. Accident Level and Potential Accident Level by Gender Count





1. From the above plot, the proportion of accident levels in each gender is not equal and males have higher accident levels than females

2. Potential accident level IV is higher in Male as compared to Female

3. Potential accident level II is higher in Females as compared to Male

2.2.18. Hypothesis Testings

1. Check the proportion of metal, mining, and others sector in Country_01 and whether the difference is statistically significant?

We have observed that the Metals and the Mining industry are not available in Country_03. Also, the distribution of each industry country-wise differs significantly.

State the Ho and Ha:

Ho = There is no difference in the proportion of the industry sector.

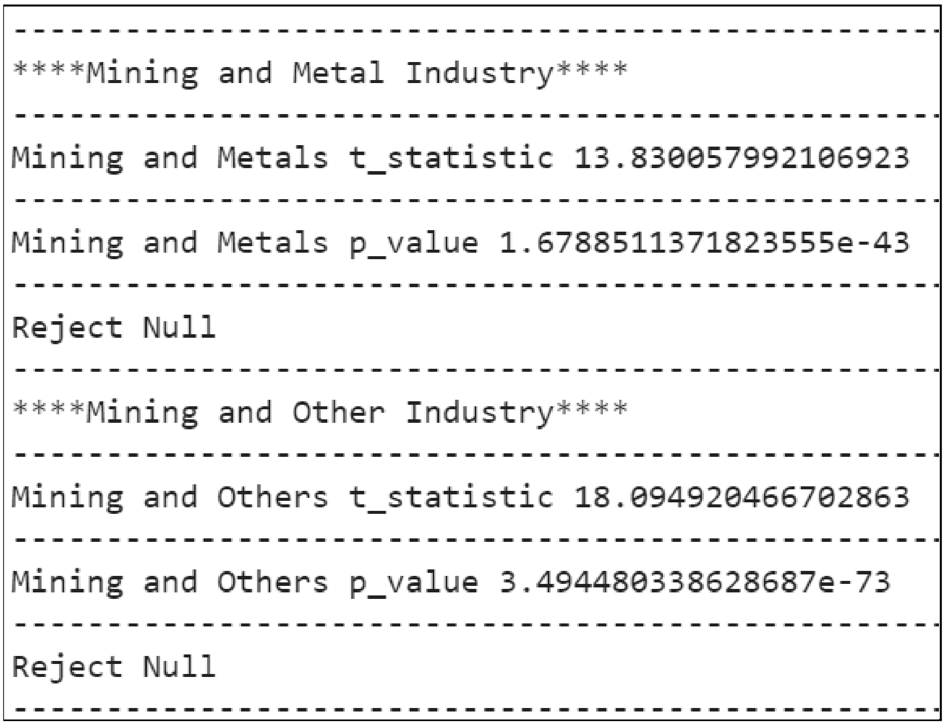
Ha = There is a difference in the proportion.

Decide the significance level: **alpha = 0.05**

Identify the test-statistic: **Z-test** of proportions

Proportions of mining, metals, others in country_01 = 81.0%, 19.0%, 1.0% respectively.

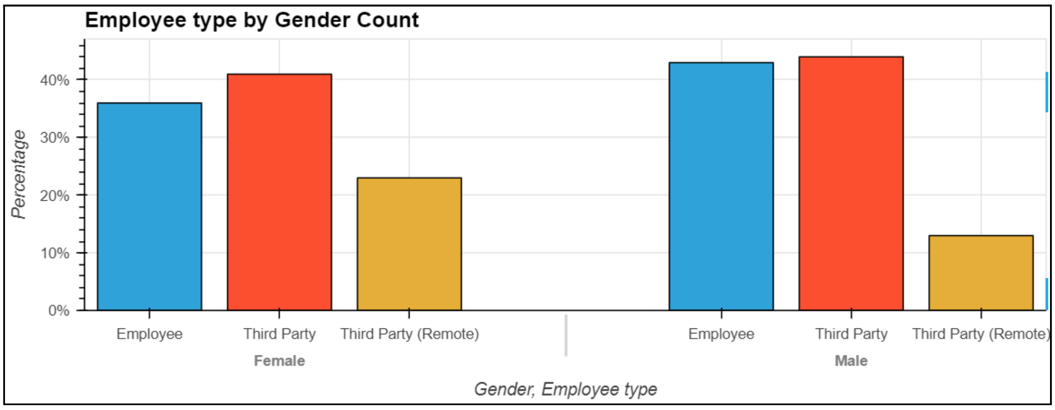
Calculate the p_value using test-statistic.



```
--------------------------------------------------
****Mining and Metal Industry****
--------------------------------------------------
Mining and Metals t_statistic 13.830057992106923
--------------------------------------------------
Mining and Metals p_value 1.6788511371823555e-43
--------------------------------------------------
Reject Null
--------------------------------------------------
****Mining and Other Industry****
--------------------------------------------------
Mining and Others t_statistic 18.094920466702863
--------------------------------------------------
Mining and Others p_value 3.494480338628687e-73
--------------------------------------------------
Reject Null
--------------------------------------------------
```

From the above, we have enough (95% confident) evidence to prove that the proportion of industry sectors is ***different*** in country_01.

2. Employee type by Gender - Is the distribution of employee type differ significantly gender-wise?



a. Proportion of third-party employees in each gender is equal

b. The proportion of the third party(remote) employees in each gender are not equal

c. The proportion of own employees in each gender is not equal

Let's check if that difference is statistically significant?

State the Ho and Ha:

Ho = The proportions of own employees in each gender are equal.

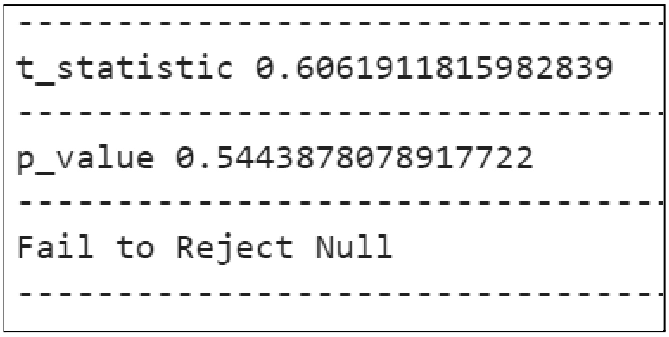
Ha = The proportions of own employees in each gender are not equal.

Decide the significance level: **alpha = 0.05**

Identify the test-statistic: **Z-test** of proportions

Proportion of own employee types in male, female = 43.0%, 36.0% respectively.

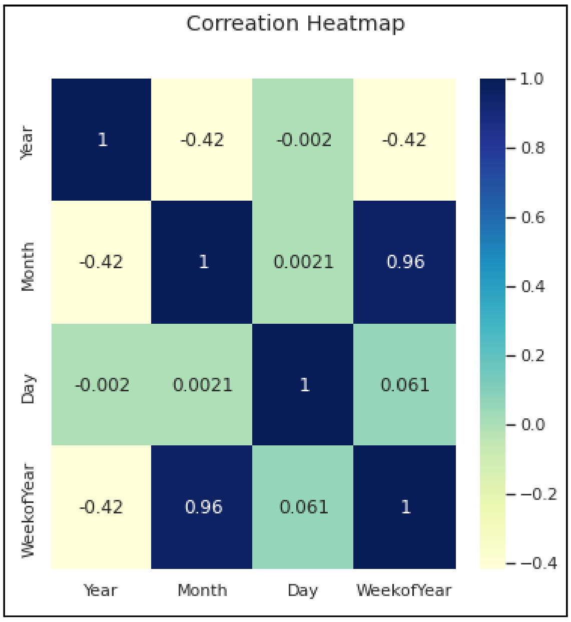
Calculate the p_value using test-statistic.



```
--------------------------------
t_statistic 0.6061911815982839
--------------------------------
p_value 0.5443878078917722
--------------------------------
Fail to Reject Null
--------------------------------
```

From the above, we fail to reject Null Hypothesis, we have enough (95% confident) evidence to prove that the proportion of own employees in each gender is ***equal***.

2.2.19. Correlation Between the Features



The above heatmap shows that WeekofYear feature is having very high positive correlation with Month feature.

2.2.19. Summary of Exploratory Data Analysis

1. **Local**: Highest manufacturing plants are located in Local_03 city and lowest in Local_09 city.
2. **Country**: Percentage(%) of accidents occurred in respective countries: 59% in Country_01, 31% in Country_02 and 10% in Country_03.
3. **Industry Sector**: Percentage(%) of manufacturing plants belongs to respective sectors: 57% to Mining sector, 32% to Metals sector and 11% to Others sector.

4. **Country and Industry Sector**: Metals and Mining industry sector plants are not available in Country_03. Distribution of industry sectors differ significantly in each country.
5. **Accident Levels**: The number of accidents decreases as the Accident Level increases and increases as the Potential Accident Level increases.
6. **Gender**: There are more men working in this industry as compared to women.
7. **Employee type**: 44% Third party employees, 43% own employees and 13% Third party(Remote) employees working in this industry.
8. **Gender and Employee type**: Proportion of third party employees in each gender is equal, third party(remote) employees in each gender are not equal and own employees in each gender are not equal.
9. **Gender and Accident Levels**: Males have a higher accident level than females. There are many low risks at general accident level, but many high risks at potential accident level.
10. **Correlation**: WeekOfYear feature is showing very high positive correlation with Month Feature.

3. MODEL BUILDING

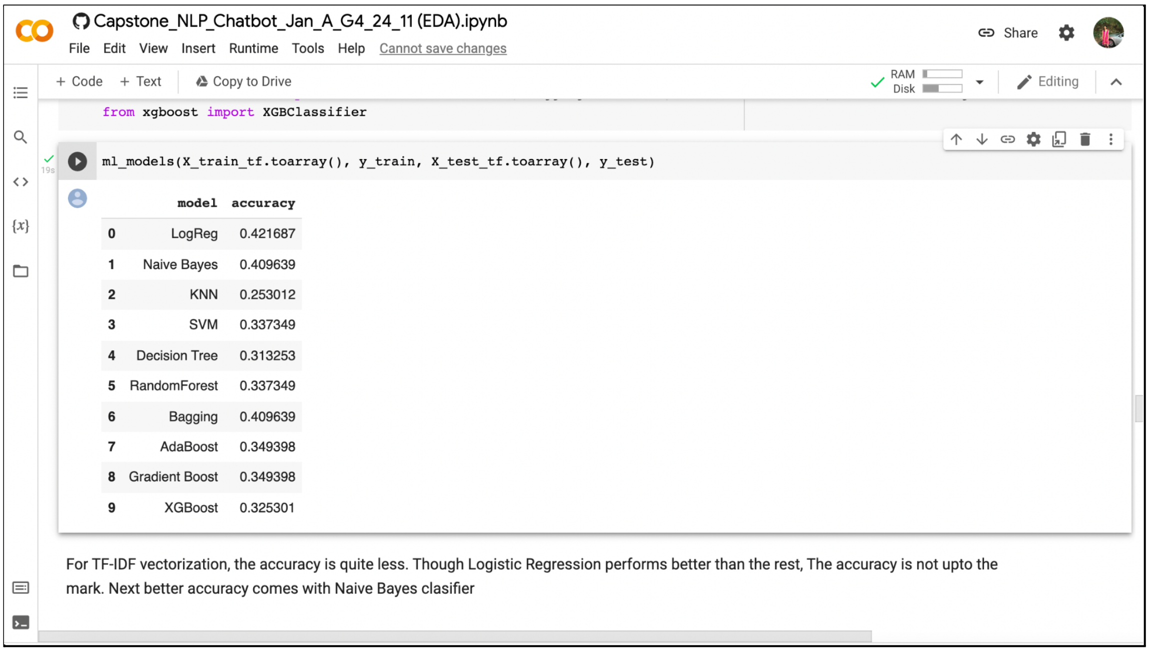
Currently, we are using three methods to build the models.

1. ML models
2. Artificial Neural Networks
3. LSTM

3.1. Using Machine Learning Models

Using ML models: Have built the basic ML models with Processed description Vs Potential Accident Levels and observed the below:

Using TF-IDF:


Capstone_NLP_Chatbot_Jan_A_G4_24_11 (EDA).ipynb
File Edit View Insert Runtime Tools Help Cannot save changes
Share
+ Code + Text
Copy to Drive
RAM
Disk
Editing
19s

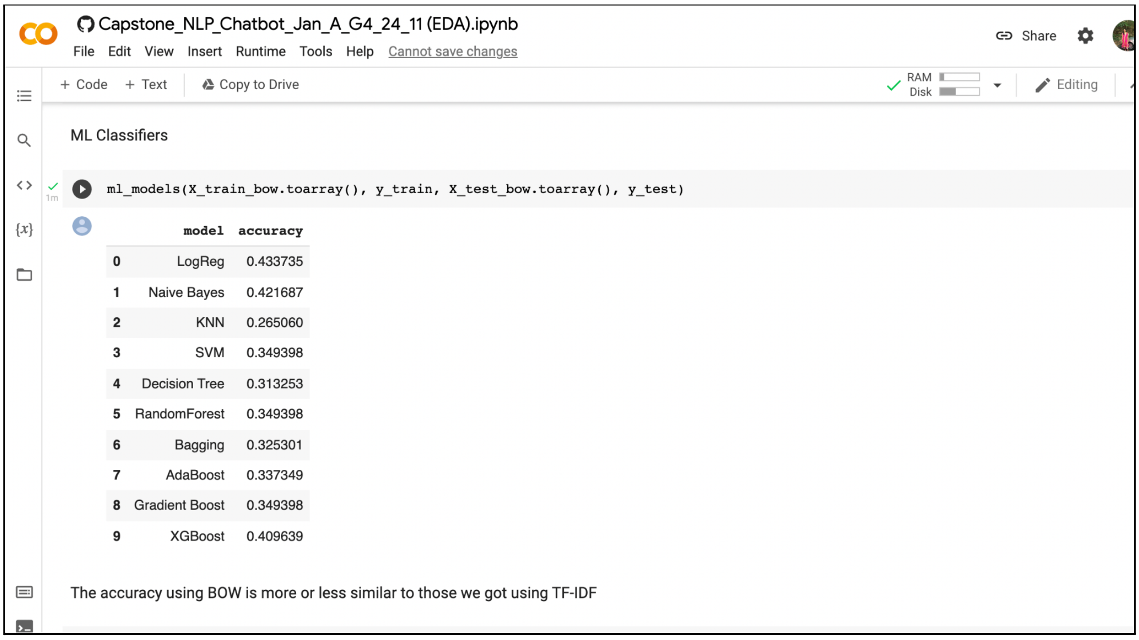
```
from xgboost import XGBClassifier
```

```
ml_models(X_train_tf.toarray(), y_train, X_test_tf.toarray(), y_test)
```

	model	accuracy
0	LogReg	0.421687
1	Naive Bayes	0.409639
2	KNN	0.253012
3	SVM	0.337349
4	Decision Tree	0.313253
5	RandomForest	0.337349
6	Bagging	0.409639
7	AdaBoost	0.349398
8	Gradient Boost	0.349398
9	XGBoost	0.325301

For TF-IDF vectorization, the accuracy is quite less. Though Logistic Regression performs better than the rest, The accuracy is not upto the mark. Next better accuracy comes with Naive Bayes clasifier

Using Bag of Words:


Capstone_NLP_Chatbot_Jan_A_G4_24_11 (EDA).ipynb
File Edit View Insert Runtime Tools Help Cannot save changes
Share
+ Code + Text
Copy to Drive
RAM
Disk
Editing

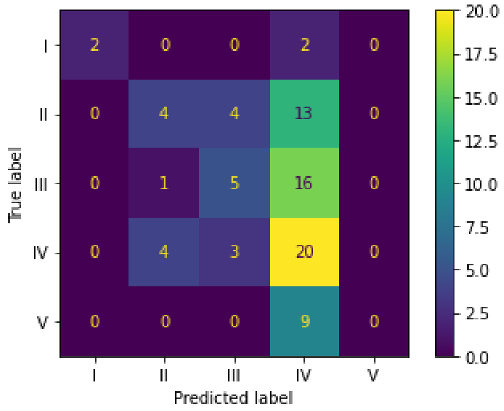
ML Classifiers

```
ml_models(X_train_bow.toarray(), y_train, X_test_bow.toarray(), y_test)
```

	model	accuracy
0	LogReg	0.433735
1	Naive Bayes	0.421687
2	KNN	0.265060
3	SVM	0.349398
4	Decision Tree	0.313253
5	RandomForest	0.349398
6	Bagging	0.325301
7	AdaBoost	0.337349
8	Gradient Boost	0.349398
9	XGBoost	0.409639

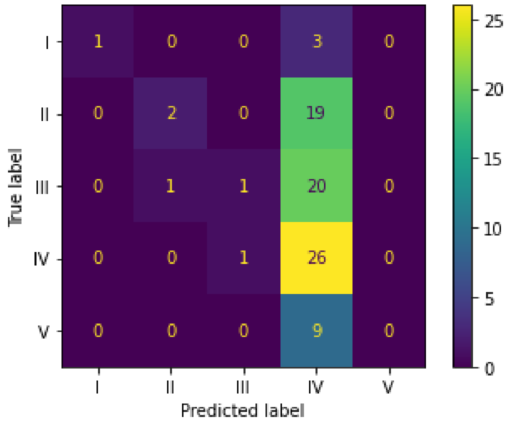
The accuracy using BOW is more or less similar to those we got using TF-IDF

```
LogReg
              precision    recall  f1-score   support

           I       1.00      0.50      0.67         4
          II       0.44      0.19      0.27        21
         III       0.42      0.23      0.29        22
          IV       0.33      0.74      0.46        27
           V       0.00      0.00      0.00         9

    accuracy                           0.37        83
   macro avg       0.44      0.33      0.34        83
weighted avg       0.38      0.37      0.33        83
```

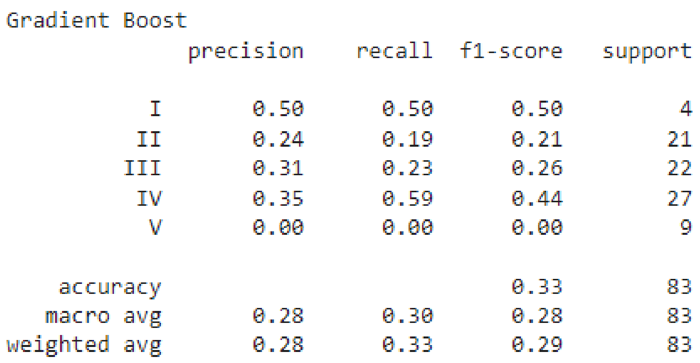


Logistic Regression

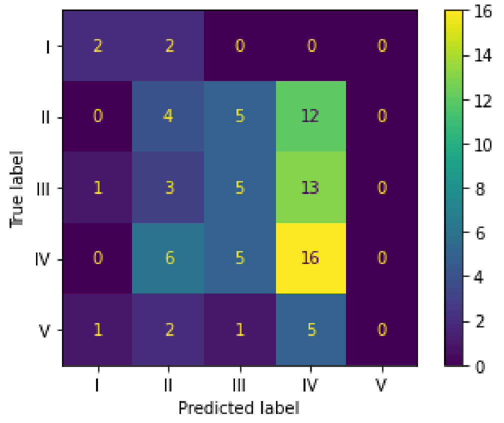
```
SVM
              precision    recall  f1-score   support

           I       1.00      0.25      0.40         4
          II       0.67      0.10      0.17        21
         III       0.50      0.05      0.08        22
          IV       0.34      0.96      0.50        27
           V       0.00      0.00      0.00         9

    accuracy                           0.36        83
   macro avg       0.50      0.27      0.23        83
weighted avg       0.46      0.36      0.25        83
```



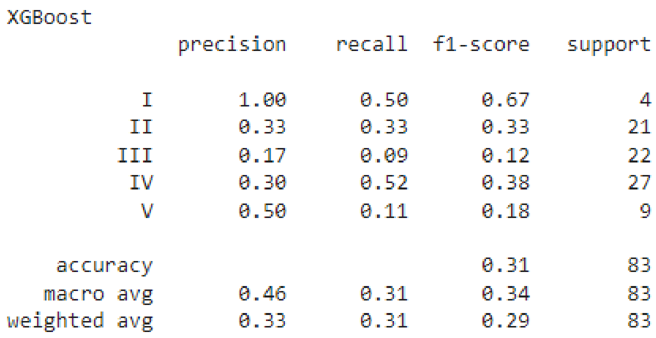
Support Vector Machine



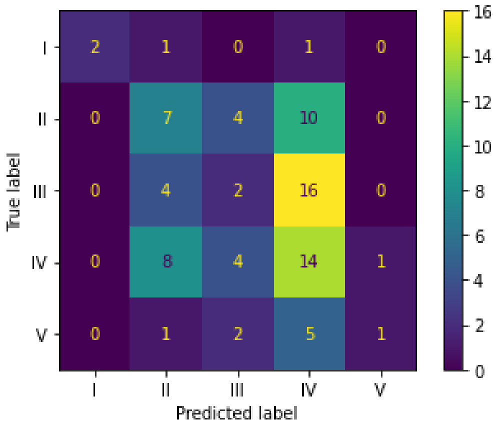
```
Gradient Boost
              precision    recall  f1-score   support

           I       0.50      0.50      0.50         4
          II       0.24      0.19      0.21        21
         III       0.31      0.23      0.26        22
          IV       0.35      0.59      0.44        27
           V       0.00      0.00      0.00         9

    accuracy                           0.33        83
   macro avg       0.28      0.30      0.28        83
weighted avg       0.28      0.33      0.29        83
```



Gradient Boosting



```
XGBoost
              precision    recall  f1-score   support

           I       1.00      0.50      0.67         4
          II       0.33      0.33      0.33        21
         III       0.17      0.09      0.12        22
          IV       0.30      0.52      0.38        27
           V       0.50      0.11      0.18         9

    accuracy                           0.31        83
   macro avg       0.46      0.31      0.34        83
weighted avg       0.33      0.31      0.29        83
```

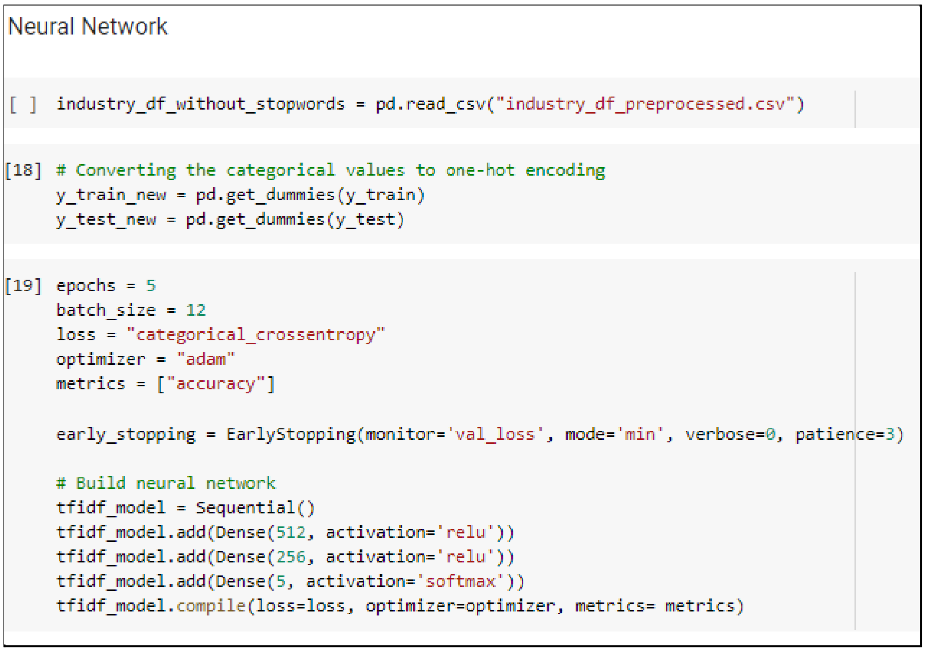


XG Boost

3.2. Using Artificial Neural Networks:

Have built the basic ANN models with Processed description Vs Potential Accident Levels and observed the below:

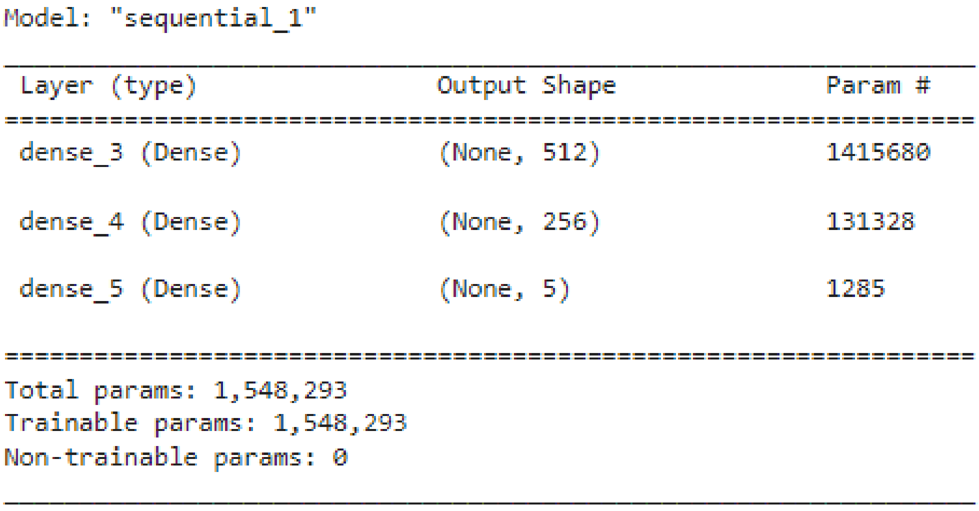
3.2.1. ANN Layer and Summary



Neural Network

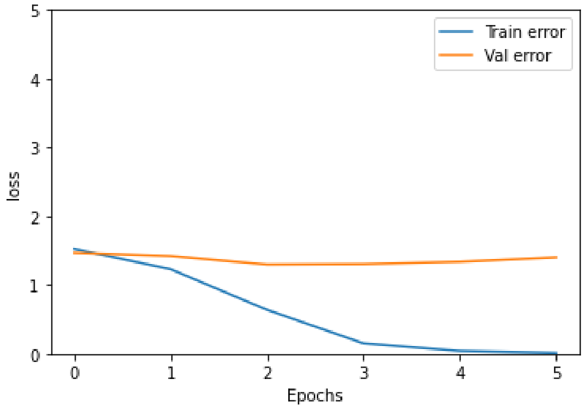
```
[ ]  industry_df_without_stopwords = pd.read_csv("industry_df_preprocessed.csv")

[18] # Converting the categorical values to one-hot encoding
     y_train_new = pd.get_dummies(y_train)
     y_test_new = pd.get_dummies(y_test)

[19] epochs = 5
     batch_size = 12
     loss = "categorical_crossentropy"
     optimizer = "adam"
     metrics = ["accuracy"]

     early_stopping = EarlyStopping(monitor='val_loss', mode='min', verbose=0, patience=3)

     # Build neural network
     tfidf_model = Sequential()
     tfidf_model.add(Dense(512, activation='relu'))
     tfidf_model.add(Dense(256, activation='relu'))
     tfidf_model.add(Dense(5, activation='softmax'))
     tfidf_model.compile(loss=loss, optimizer=optimizer, metrics= metrics)
```

ANN Layers

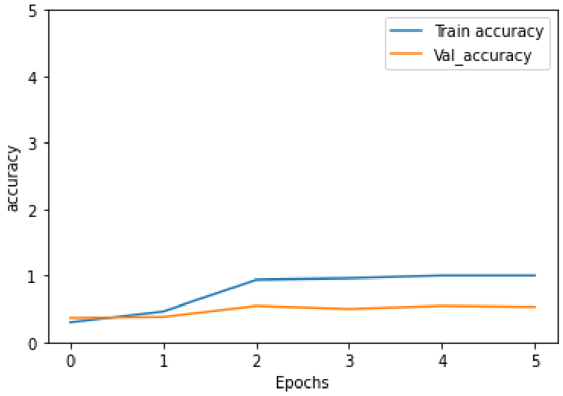


```
Model: "sequential_1"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 dense_3 (Dense)             (None, 512)               1415680

 dense_4 (Dense)             (None, 256)               131328

 dense_5 (Dense)             (None, 5)                 1285

=================================================================
Total params: 1,548,293
Trainable params: 1,548,293
Non-trainable params: 0
_________________________________________________________________
```

Model Layout Summary

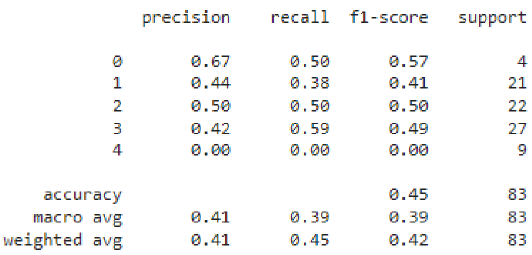


Training loss vs Validation loss



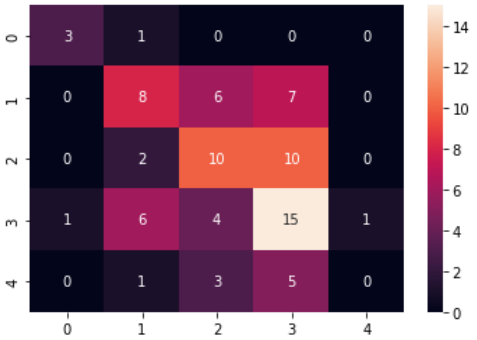
Train accuracy vs Validation accuracy

3.2.2. Confusion Matrix & Classification Report



```
              precision    recall  f1-score   support

           0       0.67      0.50      0.57         4
           1       0.44      0.38      0.41        21
           2       0.50      0.50      0.50        22
           3       0.42      0.59      0.49        27
           4       0.00      0.00      0.00         9

    accuracy                           0.45        83
   macro avg       0.41      0.39      0.39        83
weighted avg       0.41      0.45      0.42        83
```

Classification report



Confusion Matrix

3.3. LSTM:

Long Short-Term Memory (LSTM) networks are a type of recurrent neural network capable of learning order dependence in sequence prediction problems. This is a behavior required in complex problem domains like machine translation, speech recognition, and more. LSTMs are a complex area of deep learning.

In our model we took the processed Descriptions(text data) as our features and the Potential Accident Level as our Targets.

3.3.1. Targets [Y] - We label encode the targets to convert the Roman letters into numbers ranging from 0 to 4, then we one-hot encode it. The targets will be converted into size of 411*5.

3.3.2. Features[X] - For LSTM we don't remove the stopwords while preprocessing and we used Tokeniser from keras library to convert the words into numbers with maximum words of 3000.

Vocabulary size - 2841

Maximum number of words in the description - 183

The Maximum length of our features are kept as 185 and we pad all our data in our description. With this the data size will be 411*185.

3.3.3. Glove Embeddings- For Embedding our features we use Glove Embeddings 6B to 200D. We created a Dictionary for our Glove embeddings.Then we created a Embedding matrix with our Vocabulary as our row size and 200D as our columns. The Embedding Matrix size will be 2841*200 which will act as the embedding weights for our LSTM model.

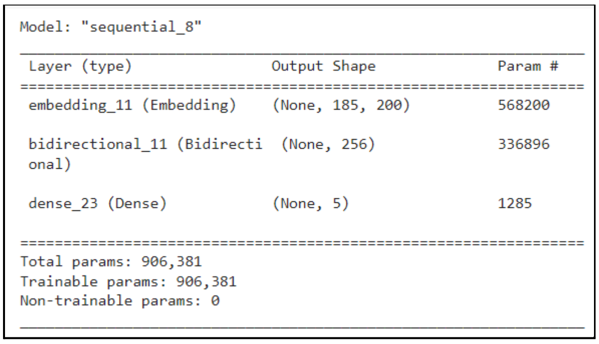
3.3.4. Models: Then we created two models,

Sequential Model- In this model we used Embedding layer with Bidirectional LSTM then finally dense layer with softmax activation function. The Final Layer is used as Sigmoid since we have multiple targets as our output.

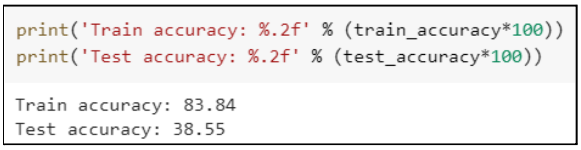
With batch size 8 and running for 10 epochs we got,

Optimizer - Adam

Loss - Categorical Cross Entropy



```
Model: "sequential_8"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 embedding_11 (Embedding)    (None, 185, 200)          568200

 bidirectional_11 (Bidirecti  (None, 256)              336896
 onal)

 dense_23 (Dense)            (None, 5)                 1285

=================================================================
Total params: 906,381
Trainable params: 906,381
Non-trainable params: 0
_________________________________________________________________
```



```
print('Train accuracy: %.2f' % (train_accuracy*100))
print('Test accuracy: %.2f' % (test_accuracy*100))

Train accuracy: 83.84
Test accuracy: 38.55
```

We can clearly see that the base Model over fits the data.

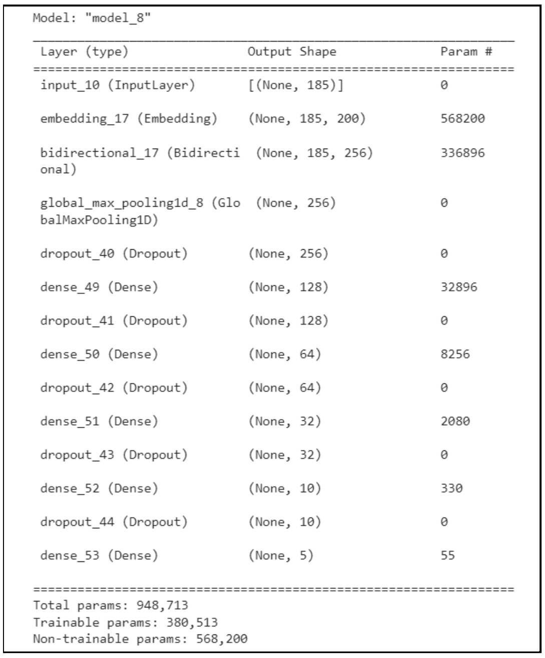
LSTM model with Dense Layers- In this model we used Embedding layer with Bidirectional LSTM Layer along with 5 dense layers and Drop out layers to gradually reduce the networks from 256 to 5. First four layers use the 'Relu' activation function and the last layer uses the 'Softmax' activation function.

With batch size 8 and running for 30 epochs we got,

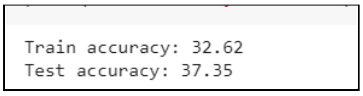
Optimizer - Adam/SGD

Loss - Categorical Cross Entropy

Learning rate - 0.001/0.0001



```
Model: "model_8"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 input_10 (InputLayer)       [(None, 185)]             0

 embedding_17 (Embedding)    (None, 185, 200)          568200

 bidirectional_17 (Bidirecti  (None, 185, 256)         336896
 onal)

 global_max_pooling1d_8 (Glo  (None, 256)              0
 balMaxPooling1D)

 dropout_40 (Dropout)        (None, 256)               0

 dense_49 (Dense)            (None, 128)               32896

 dropout_41 (Dropout)        (None, 128)               0

 dense_50 (Dense)            (None, 64)                8256

 dropout_42 (Dropout)        (None, 64)                0

 dense_51 (Dense)            (None, 32)                2080

 dropout_43 (Dropout)        (None, 32)                0

 dense_52 (Dense)            (None, 10)                330

 dropout_44 (Dropout)        (None, 10)                0

 dense_53 (Dense)            (None, 5)                 55

=================================================================
Total params: 948,713
Trainable params: 380,513
Non-trainable params: 568,200
```



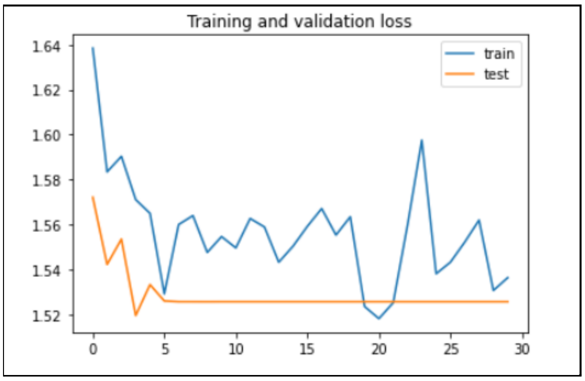
```
Train accuracy: 32.62
Test accuracy: 37.35
```

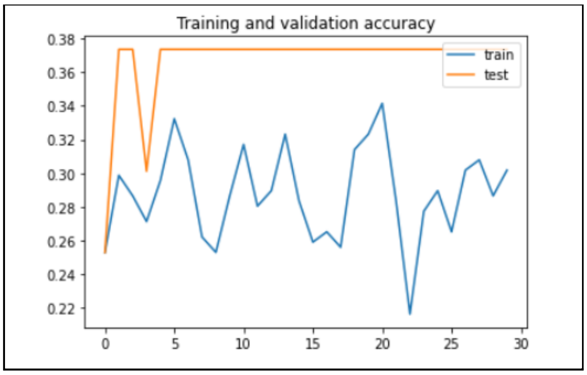
This Model doesn't overfit, but gives very less test and training accuray.

This LSTM Models have been run on various combinations such as,

Models	Accuracy
With Stop words	32 - 36 %
Without Stop Words	33 - 37%
SGD Optimizer	28 - 34%
Adam Optimizer	33 - 37%
Learning Rate 0.001	34 - 37%
Learning Rate 0.0001	32 - 36%

With all these combinations we got the accuracy ranges from 32 - 37%. Clearly LSTM doesn't work well with this data. Since the number of data points are very less we got very low accuracy, we need more data to improve the model performance.



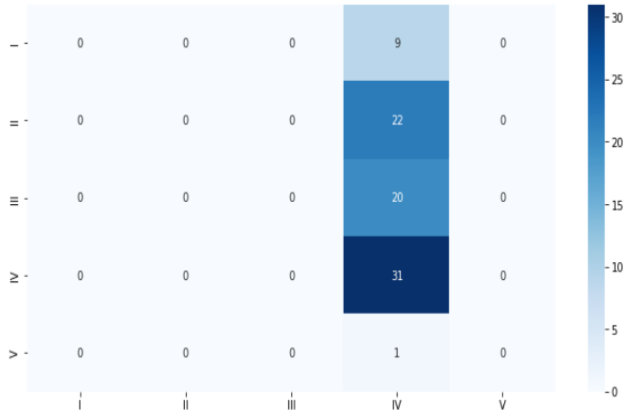


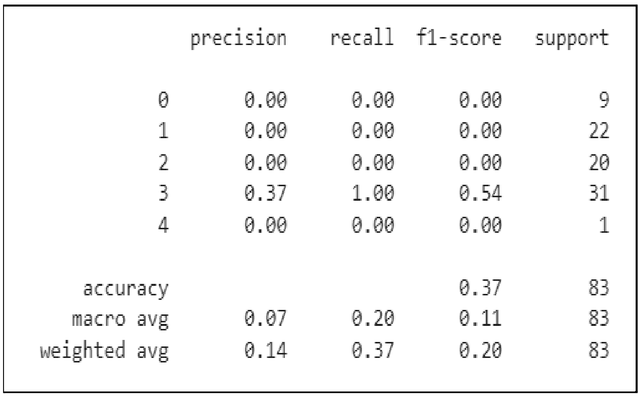
Loss Graph

Accuracy Graph

3.3.5. Confusion Matrix & Classification Report:

This model only predicts potential accident level 4 for all our test data and we got a very less weighted average F1 score. Need further analysis or resample the data to get better accuracy.





```
              precision    recall  f1-score   support

           0       0.00      0.00      0.00         9
           1       0.00      0.00      0.00        22
           2       0.00      0.00      0.00        20
           3       0.37      1.00      0.54        31
           4       0.00      0.00      0.00         1

    accuracy                           0.37        83
   macro avg       0.07      0.20      0.11        83
weighted avg       0.14      0.37      0.20        83
```

Confusion Matrix

Classification Report

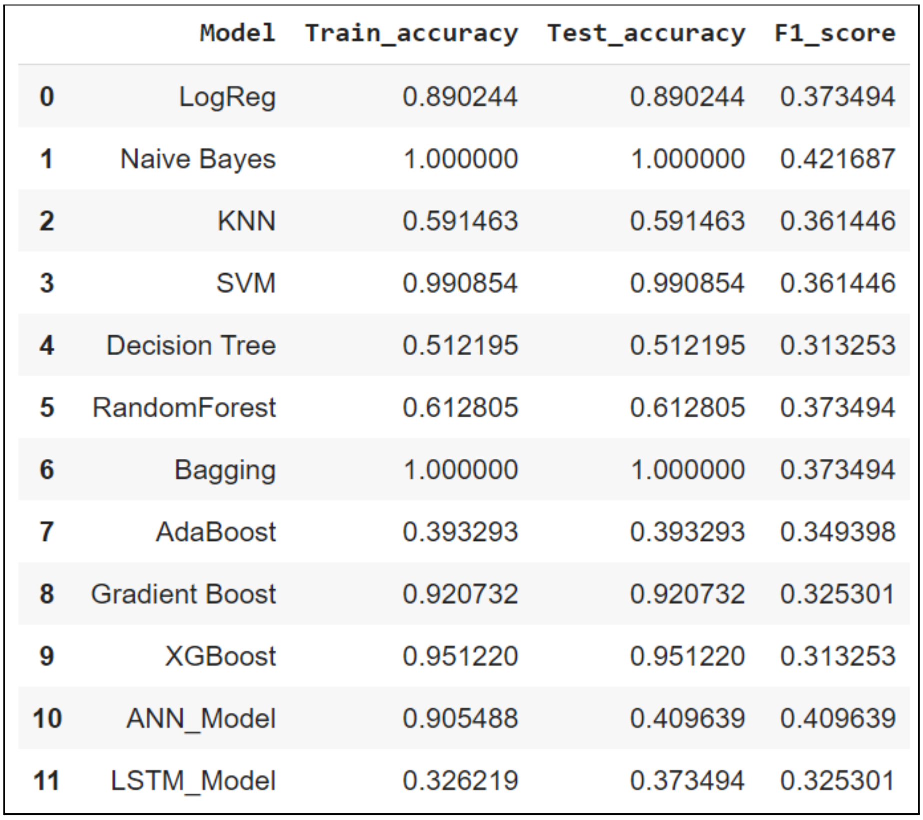
4. MODEL PERFORMANCE AND SUMMARY

4.1 Model Performance

Most model-performance measures are based on the comparison of the model's predictions with the (known) values of the dependent variable in a dataset. We have built 10 different Machine learning models, an Artificial Neural Network model and a LSTM model for analysis. The Naive Bayes and Bagging model clearly overfits the data. We have good train and test accuracy for logistic regression but low F1 score so we can neglect it. The F1 scores are very less for all the models.

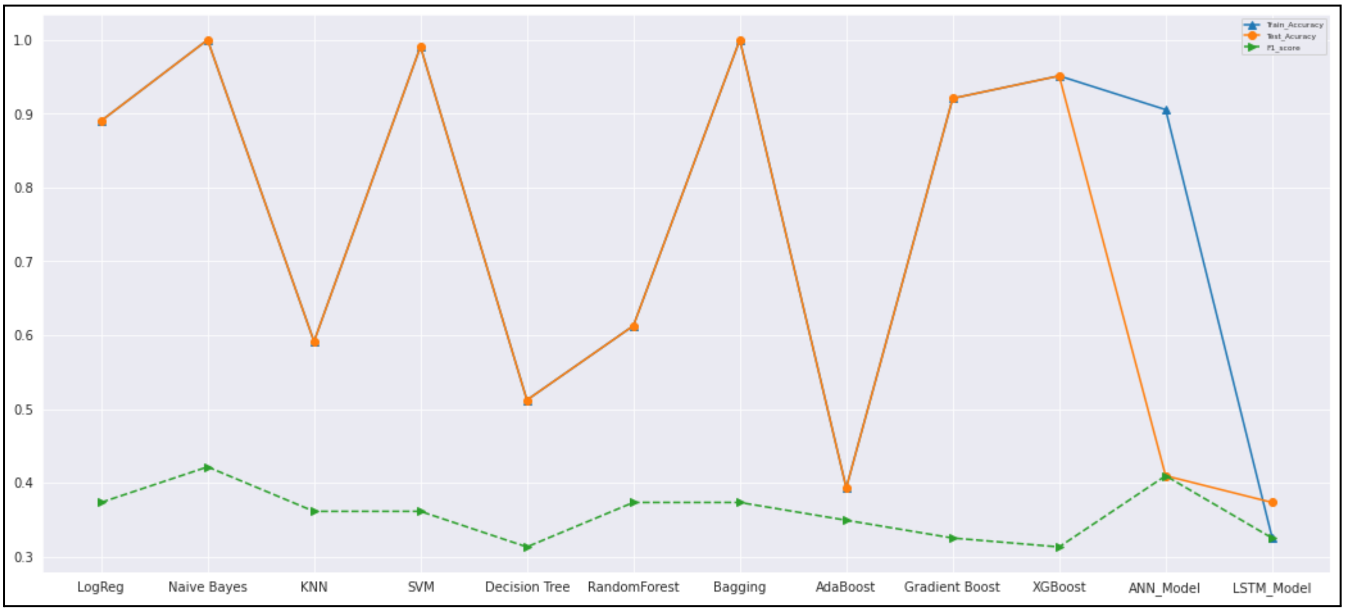
The model with Higher F1 Score is ANN model, but still the model overfits with higher train accuracy and low test accuracy. We need to build much more complex models to have better accuracy and predictions.

4.2 Model Summary



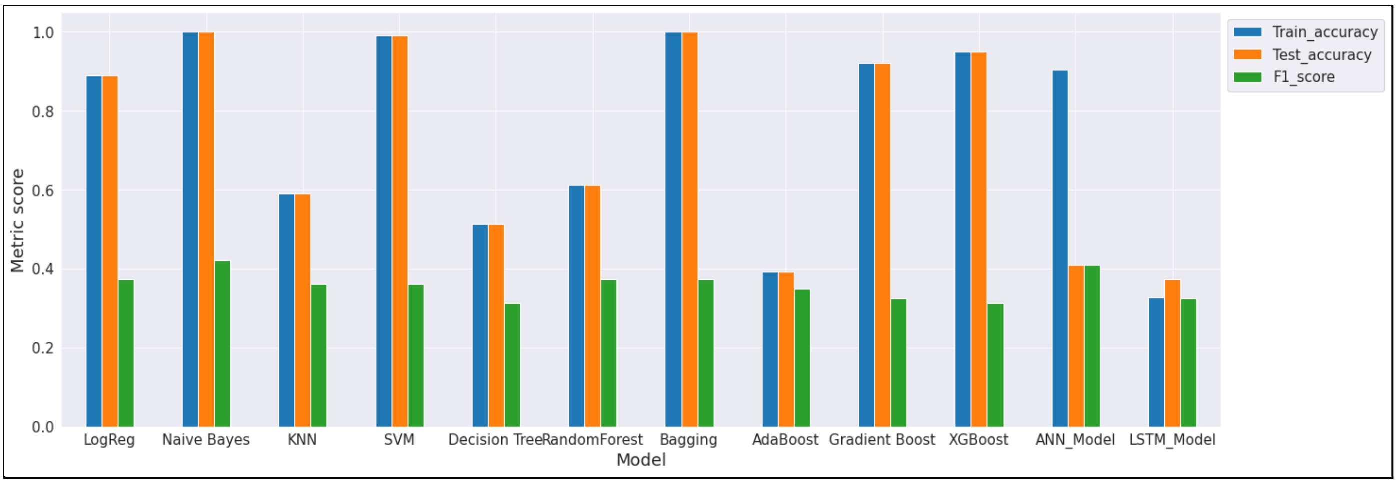
	Model	Train_accuracy	Test_accuracy	F1_score
0	LogReg	0.890244	0.890244	0.373494
1	Naive Bayes	1.000000	1.000000	0.421687
2	KNN	0.591463	0.591463	0.361446
3	SVM	0.990854	0.990854	0.361446
4	Decision Tree	0.512195	0.512195	0.313253
5	RandomForest	0.612805	0.612805	0.373494
6	Bagging	1.000000	1.000000	0.373494
7	AdaBoost	0.393293	0.393293	0.349398
8	Gradient Boost	0.920732	0.920732	0.325301
9	XGBoost	0.951220	0.951220	0.313253
10	ANN_Model	0.905488	0.409639	0.409639
11	LSTM_Model	0.326219	0.373494	0.325301

Model Performance - Train Accuracy, Test Accuracy and F1 Score



Model Performance graphs

From the Graphs we can see that most of the ML models overfit, the ANN and LSTM have decent test accuracy but again low F1 Score.



Metric Score for all the models

4.3. Future Models to Implement:

These are the upcoming models to be implemented and check its corresponding performance.

- Hyper Parameter tuning for the models
- CNN +LSTM Multi Input Models
- LSTM/GRU + LSTM/GRU Multi Input Models
- BERT
- Simple Transformers
- Experiment with Text Data Augmentation

5. GUI BUILDING:

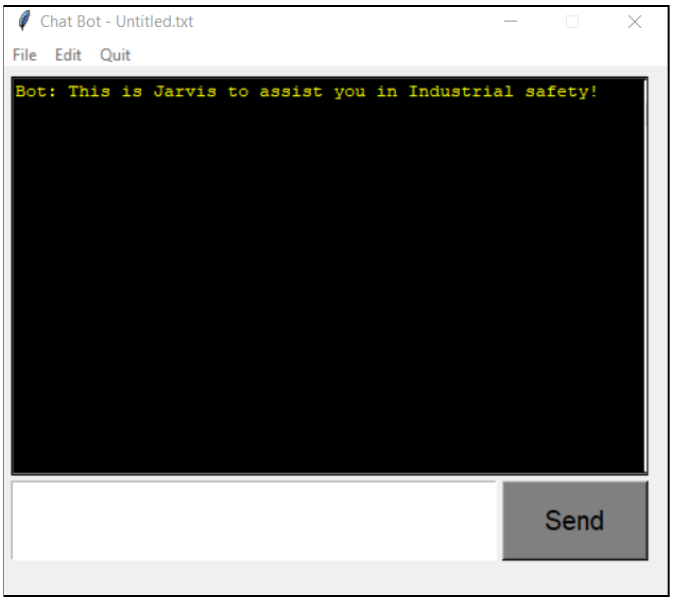
GUI means Graphical User Interface. It is the common user Interface that includes Graphical representation like buttons and icons, and communication can be performed by interacting with these icons rather than the usual text-based or command-based communication.

5.1 Chatterbot:

ChatterBot is a machine-learning based conversational dialog engine built in Python which makes it possible to generate responses based on collections of known conversations. The language independent design of ChatterBot allows it to be trained to speak any language.

Our Project Objective is to design a Clickable UI based chatbot interface which accepts text as input and replies back with relevant answers.AI chatbot with GUI using Python Tkinter. This chatbot uses NLP(Natural language Processing) and takes an Article as input and responds to user commands based on that Article.

When receiving the greeting message from the User our industry safety Chatbot greets the User. After that while entering the description it will process the text with our finalised model and predict the Potential accident level and return it back to the user.



Bot: This is Jarvis to assist you in Industrial safety!

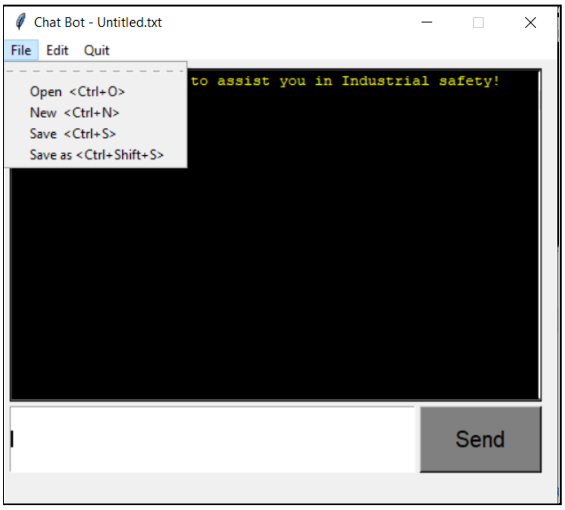
Chatbot Layout

The chatbot as the layout of,

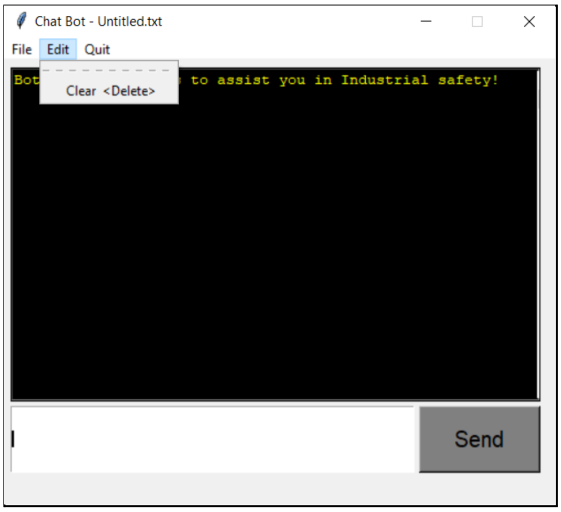
- The black window is the text window - where all our conversations are displayed
- The White Box is the Chat Window - where the User enters the text
- Send Button - To send the chat to the bot.
- Menu bar - contains File, Edit, Quit menus.

The Menu bars have some events like:

- New - Open a new Window to start the conversation
- Open - To open already saved conversation
- Save as - Create a new .txt file and save our conversation in our local drive
- Save - Save the current Conversation
- Clear - To clear the conversation
-

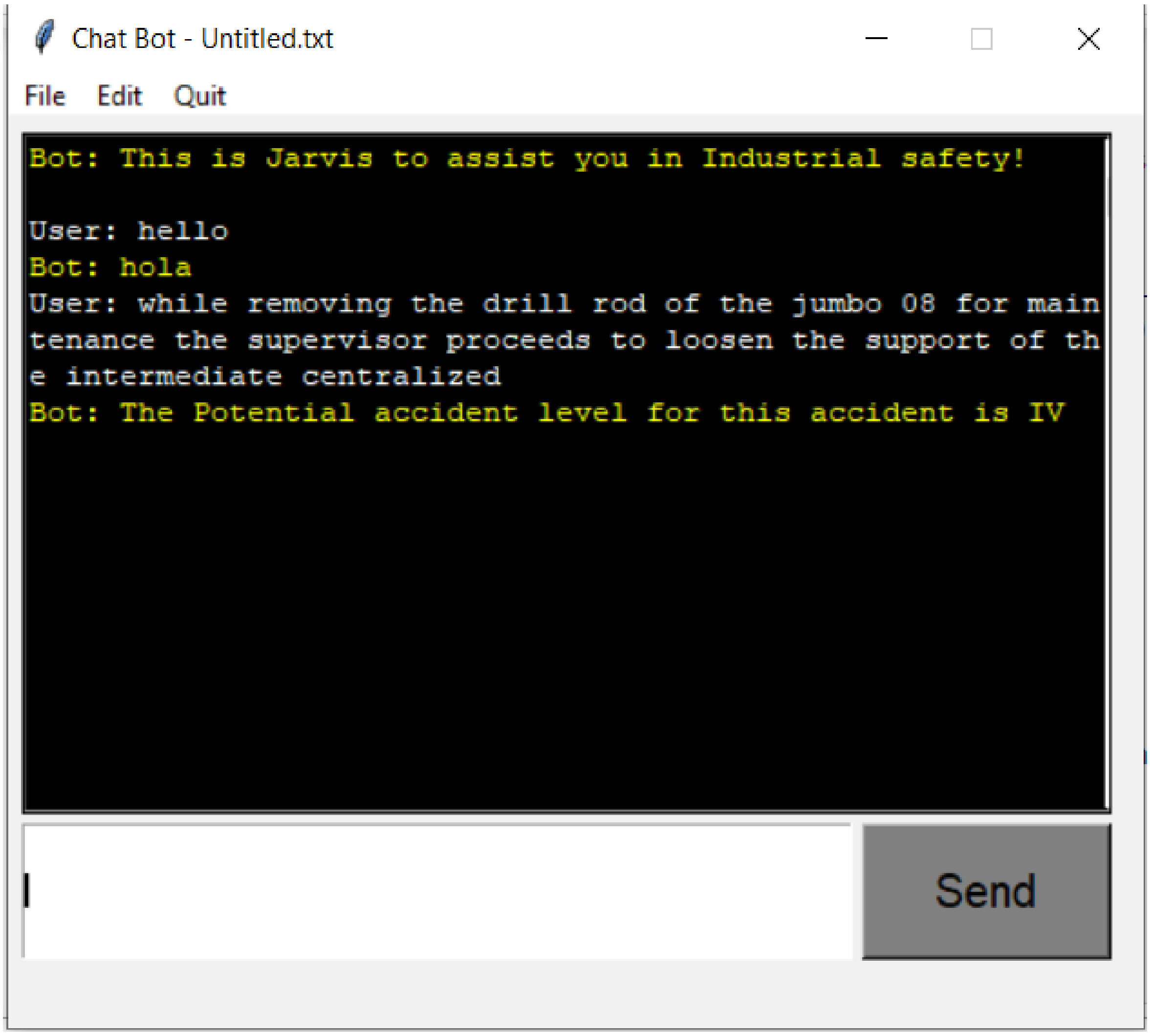


to assist you in Industrial safety!



Bot to assist you in Industrial safety!

Menus on chatbot



```
Bot: This is Jarvis to assist you in Industrial safety!

User: hello
Bot: hola
User: while removing the drill rod of the jumbo 08 for main
tenance the supervisor proceeds to loosen the support of th
e intermediate centralized
Bot: The Potential accident level for this accident is IV
```

Chatbot with sample data and prediction

6. REFERENCE

1. https://searchcustomerexperience.techtarget.com/definition/chatbot
2. https://www.mygreatlearning.com/blog/basics-of-building-an-artificial-intelligence-chatbot/#types of chatbot
3. https://towardsdatascience.com/exploratory-data-analysis-8fc1cb20fd15