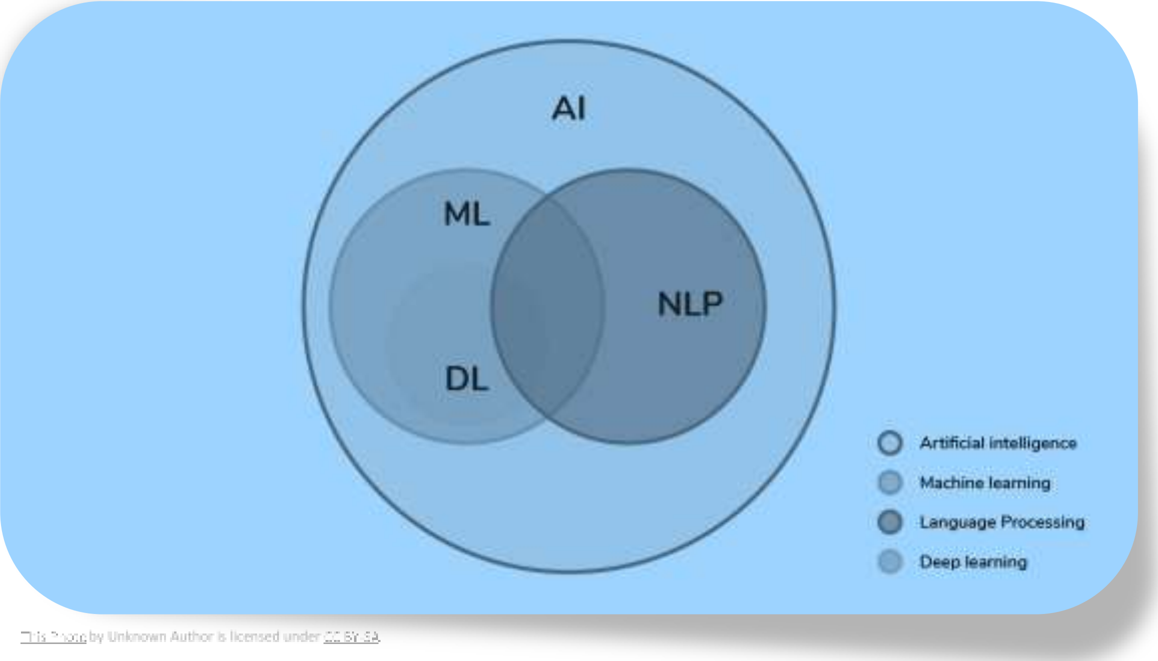
AI ML Capstone Project – NLP 2 Final Report

NLP based Chatbot project for safety department of Industrial companies



This Photo by Unknown Author is licensed under CC BY-SA

Project Final Report prepared by

Amol Bulbule
Dr. Kaushik Sarkar
Manoj Vardhan
Suresh Rajagopalan
Venkatesh Chandiran

Presented to Great learning Mentor **Mr. Srikanth Girijala**

Date: 12-Sep-2021

Disclaimer: This project is a collective effort by our team and has been done for capstone evaluation by great learning only. This is for strict internal use and confidentiality to be maintained. This is not to be shared with other group or used for external purposes. The images in front and last page are from online sources. We do not own the copyrights for them. The images captured as part of our logical flow/code architecture etc. are to be used only for internal evaluation purposes of Great learning.

Table of Contents

1. Scope of the work / Objective of this project

1.1 A high-level Abstract Summary of project

The objective of this project is to design a ML/DL based chatbot utility for the Industrial safety domain/department of any industrial companies/organizations. There is an urgent need for industries/companies around the globe to understand why employees still suffer some injuries/accidents in plants. Sometimes, they even die in such an environment. The industrial safety department is an important function of any industries/companies with plants where the work environment can be dangerous, unsafe and prone to accidents such as fire, hazardous chemicals etc. They define policies and procedures to provide a safe and hazardous-free work environment for their employees and third-party contractors. This chatbot utility will be designed and developed with the intent of helping the professionals / employees/ third party-contractors to highlight the safety risk as per the incident description.

Chatbots gained popularity when NLP algorithms and models have improved over the past decade. One of the many applications of chatbot is to automate some of the functions of their call centre or L1 helpdesk teams. They are aimed at bringing in efficiencies and productivity improvements of a L1 helpdesk team where repetitive queries can be handled as part of first level touch point. Companies across the globe implement chatbot programs as part of their digital automation initiatives to have first level of resolution to their employees/contractors based on some of the common repeatedly occurring incidents/problems. The majority of the cause of raising incidents or requests is more to do with not knowing where the information is present. For such cases, Chatbot utilities are designed to give directions to the possible resolutions. For complex problems where human intervention will be required, they can be designed with an in-built call to action (CTA) function to direct the employees towards resolution.

In this problem statement case, we will design a chatbot utility to help the professionals highlight the safety risk based on some of the incident descriptions that the users provide. Based on the safety risk, they can potentially take the next course of action.

1.2 Brief summarisation of intent, findings and implications

- Our first step was data pre-processing and in finding the right target variables.
    - We found some interesting conclusions through our univariate and bivariate and statistical tests.
    - This data was collected over a period of 1.5 years from 2016 to Jun 2017.
        - Winter seasons typically has more accidents.
        - Many accidents are reported in Mining industry followed by Metal.
        - Basis the accident data reported, Country-01 typically has more mining plants, country-02 has more metal manufacturing plants whereas country -03 has other industrial accidents reported.
        - Locations or plants 01, 03, 04 and 05 have recorded more accidents than other locations.
        - Female gender related incidents are seen more in metals industry.
    - We conducted series of hypothesis tests using Chi-square to conclude which parameters are statistically significant for deriving the accident and

    potential accident levels. Apart from Date and Employee type, our observation is all of them are significant and contributing to the outcome of the accident levels.
  - The severity levels of Accident and Potential Accident levels for the same incident are having varied numbers. Typically, this can mean a false positive or negative cases, i.e., the cases that are genuinely less severe are reported as serious and vice versa. Hence, text analysis using various models are conducted. This prompted us to use text-based NLP models for predicting accident levels and use other categorical columns to predict potential accident levels.
- Text processing analysis:
  - One observation was if the length of description field is more than 100 words, then potential accident levels are evenly distributed whereas the accident levels with severity -1 is way higher.
  - N-gram analysis showed the word “finger left hand” is often repeated in many incidents.
  - We used glove embedding, we formed the top 100 glove embedded vectors / words that contribute to the accident levels.
  - We used the text processing models to predict the accident levels
- We did the feature engineering using XG Boost and SMOTE for sampling.
- Model evaluations – after running through multiple models, we concluded the following
  - ML supervised learning – Random forest classifier
    - We used the various categorical columns apart from the text description to predict the potential accident level target variable. This variable tells what could have been the possible impact taking various factors into consideration
    - Our findings are ML Random forest classifier gave around 66% on test data
  - LSTM model
    - We used this model on text processing (description of incident field) to predict the accident level.
    - The outcome of this prediction will be to tell the end users whether they have been subjected to a low grade or a high-grade severities/casualty.
    - The model gave around 94% accuracy.
    - The performance is also far better with less than 10 seconds to predict the variable.
  - We continuously improved the model to remove false positive cases.
  - We had used other models like ANN/CNN but LSTM was found to be better
- Our GUI for chatbot utility is a flask-based container on which pytorch code was deployed which takes in the values entered from a HTML file and feeds into the model for prediction. The resultant is also a result.html one.
  - The GUI function used to capture the description of the incident. Using this, predict the accident level using LSTM
  - Capture the other parameters like country, industry, plant locations etc. to predict the potential accident level using ML models.
  - We also built our chatbot with a call to action based on pre-defined rules.

In Summary:

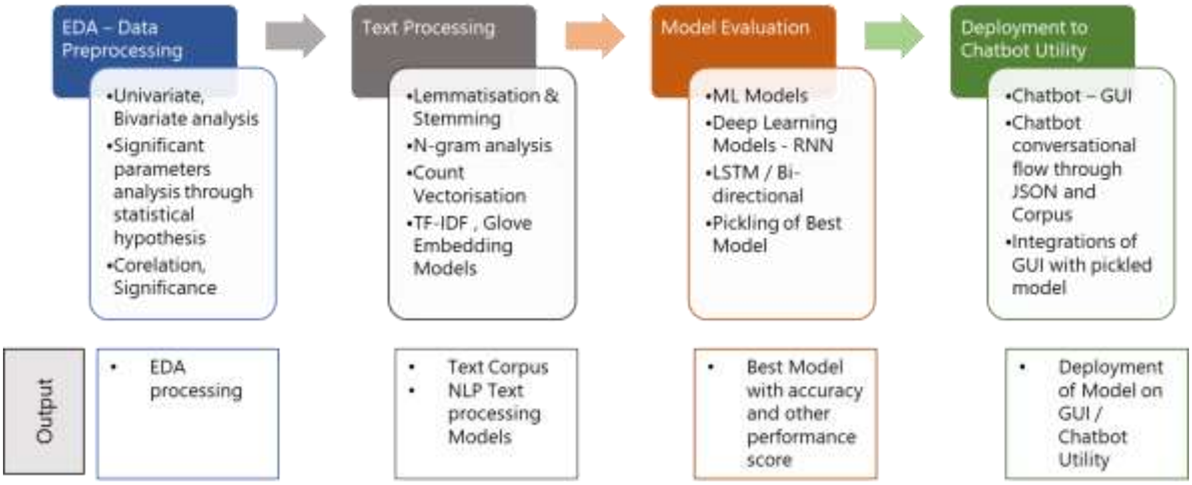
We had tried and implemented our learnings across this one year in our project. We had tried a simple and effectively usable deployment method which performed better in real time.

In the subsequent sections of this document, we will describe the following:

- overview of our final process
    - Our methodology approaches towards data processing and model algorithms
- Walk through of our solution
    - Describing each stage into building unto the final solution
- Model evaluation
    - Objective, prominent parameters and evaluation of various successful models
- Comparison to benchmark
- Visualisations
- Implications
- Limitations
- Closing Reflections

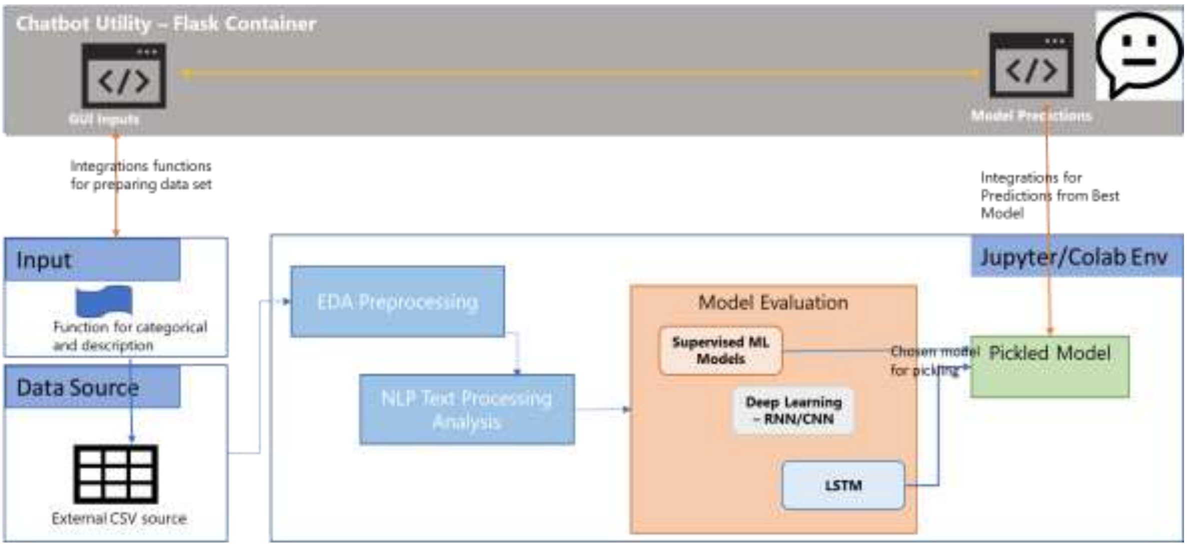
2. Overview of the final Process

The high-level approach of our methodology is as below

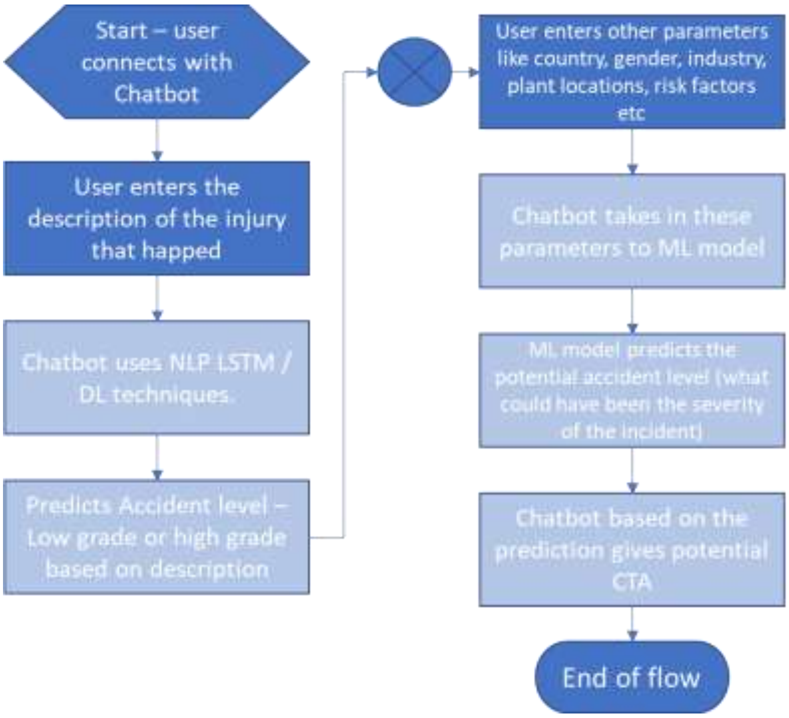

EDA – Data Preprocessing
•Univariate, Bivariate analysis
•Significant parameters analysis through statistical hypothesis
•Corelation, Significance
Text Processing
•Lemmatisation & Stemming
•N-gram analysis
•Count Vectorisation
•TF-IDF , Glove Embedding Models
Model Evaluation
•ML Models
•Deep Learning Models - RNN
•LSTM / Bi-directional
•Pickling of Best Model
Deployment to Chatbot Utility
•Chatbot – GUI
•Chatbot conversational flow through JSON and Corpus
•Integrations of GUI with pickled model
Output
• EDA processing
• Text Corpus
• NLP Text processing Models
• Best Model with accuracy and other performance score
• Deployment of Model on GUI / Chatbot Utility

At a very high level, we will be approaching the problem doing the data preprocessing of various columns presented to us. We will incorporate the text processing analysis on the description field using the NLP model techniques learnt. We will use multi-pronged approach to the model evaluation to have the best model predict the accident level and potential accident level values. Finally, this will all be deployed in Chatbot GUI utility through which our end business user will interact and automatically get the best desired call to action through our model prediction.

The high-level representative or logical architecture is as shown below:


Chatbot Utility – Flask Container
GUI Inputs
Model Predictions
Integrations functions for preparing data set
Integrations for Predictions from Best Model
Input
Function for categorical and description
Data Source
External CSV source
Jupyter/Colab Env
EDA Preprocessing
NLP Text Processing Analysis
Model Evaluation
Supervised ML Models
Deep Learning – RNN/CNN
LSTM
Chosen model for pickling
Pickled Model

We will be using the below logical flow chart from an end user perspective. This particular perspective was incorporated post the feedbacks from multiple mentoring sessions. Overall idea is to use the description field to predict the accident levels and using other categorical variable to predict the potential accident levels. This feedback was useful as we try to parameterise (or argumentized) both the text and categorical variable and send to the various ML models, the results were not satisfactory. There are so many features available for prediction that skews the overall accuracy. This approach of splitting NLP and ML model techniques allow us to predict the accident level and potential accident level effectively. The processing is also faster. That is required in chatbot as we cannot allow the end user prediction to be taking more time. The output of the model prediction is split into two stages and predicting accident level first and potential accident level in the subsequent stage, instead of having to predict only one variable through a single model.


Start – user connects with Chatbot
User enters the description of the injury that happed
Chatbot uses NLP LSTM / DL techniques.
Predicts Accident level – Low grade or high grade based on description
User enters other parameters like country, gender, industry, plant locations, risk factors etc
Chatbot takes in these parameters to ML model
ML model predicts the potential accident level (what could have been the severity of the incident)
Chatbot based on the prediction gives potential CTA
End of flow

2.1 Salient Features of the data

The data set comes from one of the biggest industries in Brazil and in the world. This dataset basically records accidents/incidents from 12 different plants taken in three different countries over a period of 18 months between 2016 and 2017. Every data record is an accident or an incident occurred and reported and extracted in the form of a CSV file to us.

The various columns that are captured as part of the data set is as below:

- Data: timestamp or time/date information
  - This is a date field with a timestamp. Most of the timestamps recorded are from the database and the time of the incident is not recorded as the exact timestamp but as 00:00:00.
- Countries: which country the accident occurred (anonymised)
  - Since three countries' data are recorded, country_01 02 and 03 are the three values. It's more of a categorical data. The data is presented for 3 different countries.

- Local: the city where the manufacturing plant is located (anonymised)
    - 12 locations or plants from which accident data is captured. Again, a categorical column
- Industry sector: which sector the plant belongs to
    - Industry sectors are dangerous and accident-prone ones like metals (metal processing or manufacturing ones) and mining where operations happen in dangerous and chemical hazardous mines. There is third value as others whichever is not fallen into metals and mining
- Accident level: from I to VI, it registers how severe was the accident (I means not severe but VI means very severe)
    - This is the original recorded accident level. One being "not severe" to 6 being the "highest severity" one.
- 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)
    - What could have been the severity of the accident level. Again, one being "not severe" to 6 being the "highest severity" one.
- Gender
    - if the person is male of female
- Employee or Third Party
    - if the injured person is an employee or a third party
- Critical Risk
    - some categorisation of the risk factors involved based on the information of the accident
- Description
    - Detailed description of how the accident happened. This will be used for NLP pre-processing along with other parameters which predicts the
    - possible accident level and accident levels.

The potential target variable can be Accident level or Potential Accident level. Using our EDA and pre-processing analysis, we will ascertain which parameters will help in predicting these closely. Since this column will be having multiple values, we will be employing multi classification ML models, later use deep learning models like RNN and finally employ LSTM models. We will ascertain one final model which gives us accurate predictions.

Both accident and potential accident levels are important in its own way. So, we will use multiple models to derive them individually. First, we will use NLP LSTM text-based description analysis/prediction of accident levels, followed by running ML model to predict the potential accident level

There are around 425 records and 11 columns in the original data set. There are no null values in the data set. One variable "Unnamed" is a redundant variable.

All the variables are of object or categorical type.

Of them the "Data" variable is a datetime variable, yet put in place in the dataframe as an object variable. Description is a free flow text whereas the remaining attributes like country, local, critical risk, Gender, Employee type and Accident level (potential included) are all categorical in nature.

Note: **Data** variable seems to be an object, but in real sense it is a Date column

There are errors in variable names: "Data" and "Genre", which needs correction. Genre is Gender column. Data is Date column. The column of Description contains textual data that needs to be analysed using NLP.

Dropping & renaming Data Variables:

From the dataset, we understand

'Unnamed' column is dropped as this doesn't contribute to the analysis

Renaming variables: we renamed the following variables to have a meaningful column in the data frame

- Data → Date
- Countries → Country
- Genre → Gender
- Employee or Third PArty → Employee Type

Duplicate Removal:

From the dataset of 425 rows and 11 columns, after carrying out the drop duplicate, we infer the clean dataset of 418 rows and 10 columns.

Check Missing Values

The dataset doesn't contain any missing Values

2.2 Pre-Processing Steps:

We will do the following as part of Data Pre-processing steps

- Pre-Processing steps:
  - Getting to know the data set better. Assessing the data using various EDA techniques, to know about dataframes, the null values if any, removing duplicates have been carried out.
- Univariate analysis
- Bivariate analysis
- Statistical tests
- Concluding on EDA part

a) Secondary Variable creation:

Since the Date column has a timestamp which is only updated as 00:00:00, we will create secondary variables to identify any specific time period that has an impact on the accidents that occurred. So, in order to check that we will create the secondary variables and split a date into day, month, week and year. Accidents may increase or decrease throughout the year or month, so we need to add datetime features such as year, month and day.

Further, countries where the dataset was collected are all located in South America. So, in this analysis, let's assume the dataset was collected in Brazil.

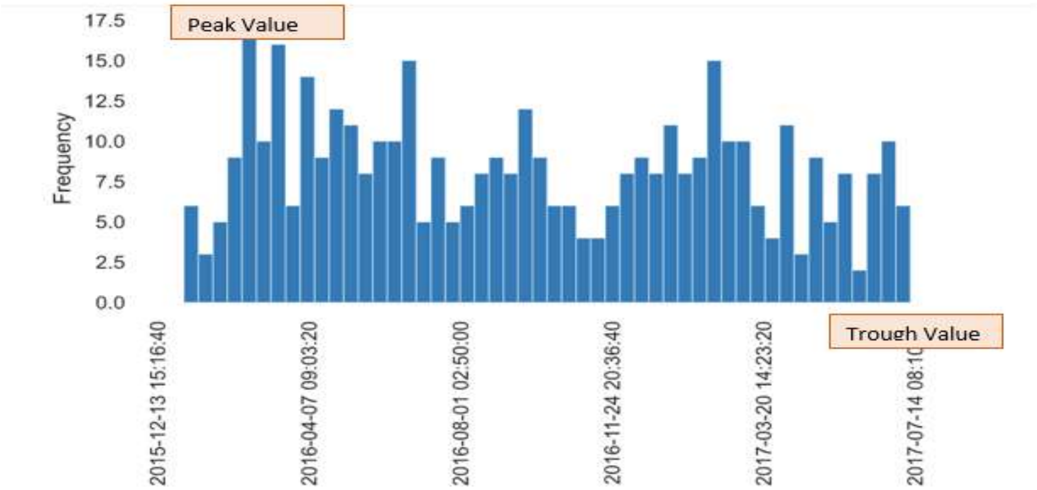
Here Brazil has four climatological seasons as below.

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

We can create seasonal variables based on month variables. After the creation of the secondary variables, the new data frame has this set of columns:

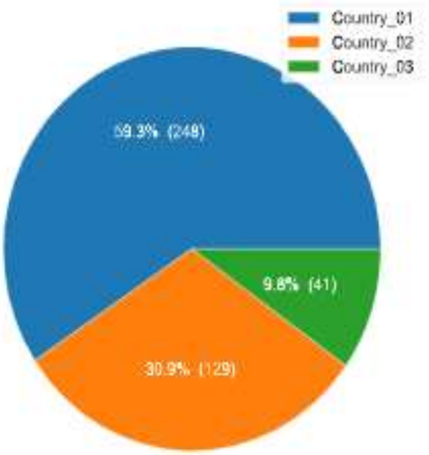
b) Basic Variable analysis

The below is a histogram of peak frequency of accidents date wise. We can see peaks on some days and troughs on some days. It is evident that they must have taken some actions which might have led to this. Having said that, let us analyse date wise here.



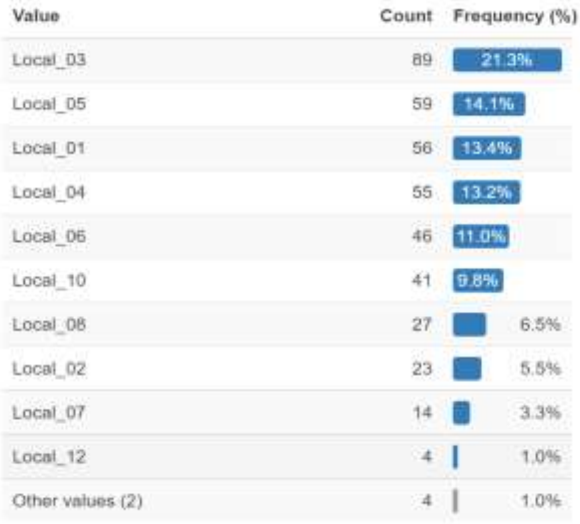
We cannot ascertain at this point if any actions were taken to lead that led to increased or decreased accident levels on that day. However, looking at other patterns, it is clear that date cannot be a factor in predicting the variable. But it is important for recording purpose.

Country wise data profiling:



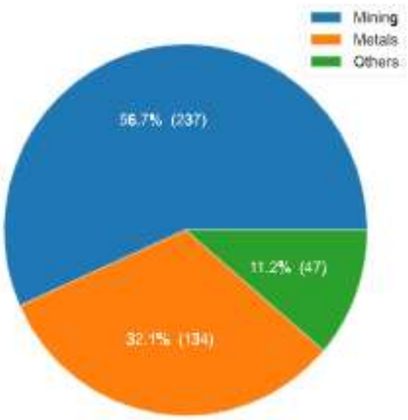
Country_01 has the maximum occurrence of accidents followed by Country_02 and Country_03.

Local variable: The following is the distribution of the local variable across 12 locations.



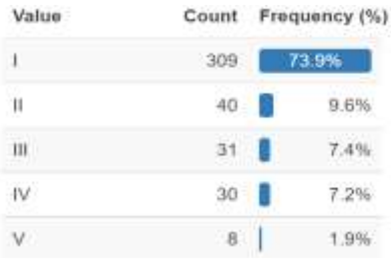
Value	Count	Frequency (%)
Local_03	89	21.3%
Local_05	59	14.1%
Local_01	56	13.4%
Local_04	55	13.2%
Local_06	46	11.0%
Local_10	41	9.8%
Local_08	27	6.5%
Local_02	23	5.5%
Local_07	14	3.3%
Local_12	4	1.0%
Other values (2)	4	1.0%

Industry sector wise:



It is clear that the maximum incidents occurred in the Mining industry followed by Metals and others. This clearly indicates Mining has more dangerous zones and accident-prone work environments. Metal Industry also has a significant number of 32% which indicates industrial plants will have exposure to unsafe practices.

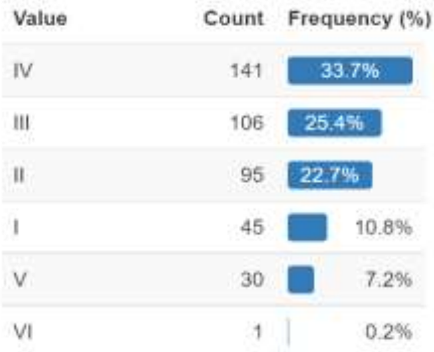
Reported Accident Levels (recorded accident levels):



Value	Count	Frequency (%)
I	309	73.9%
II	40	9.6%
III	31	7.4%
IV	30	7.2%
V	8	1.9%

Accident level I being the least severe and V being the most severe ones, it is clear from this distribution that Severity - I have been recorded the most. The less severe and least severe categories contribute to 80% of the accidents that happen. There seems to be a data imbalance here. But are they really the lowest severity ones? Let us analyse potential accident levels.

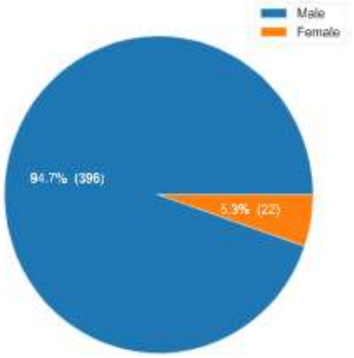
Potential Accident Levels:



Value	Count	Frequency (%)
IV	141	33.7%
III	106	25.4%
II	95	22.7%
I	45	10.8%
V	30	7.2%
VI	1	0.2%

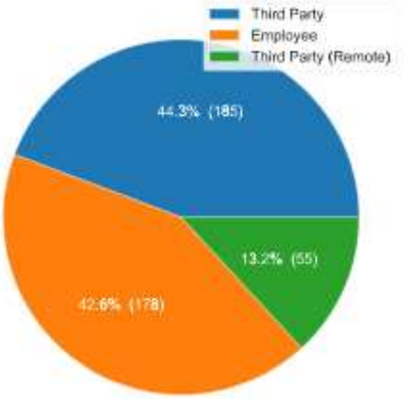
Based on the analysis, it was recorded later that what could have been the accident levels later. This shows a very different picture. If we count the high (III), very high (IV) and the most severe levels (V), it adds up to more than 65% of the cases. This indicates to us that this may be a better target variable than the accident levels as it was more skewed towards the lower end.

Gender:



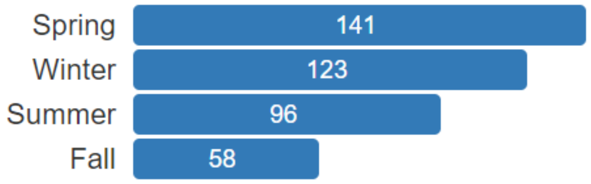
Gender clearly shows data imbalance. We will do statistical significance later to check if it really aids in our prediction.

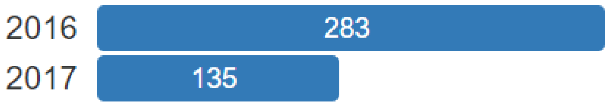
Employee Type:

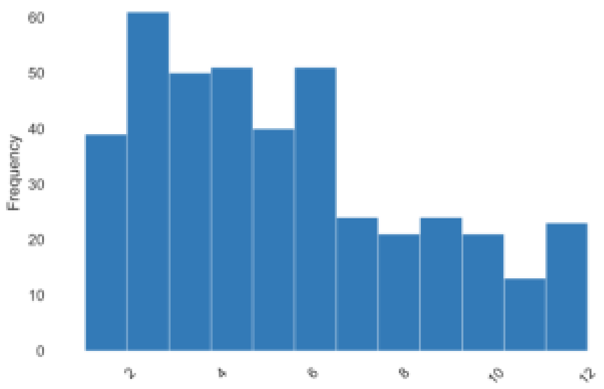


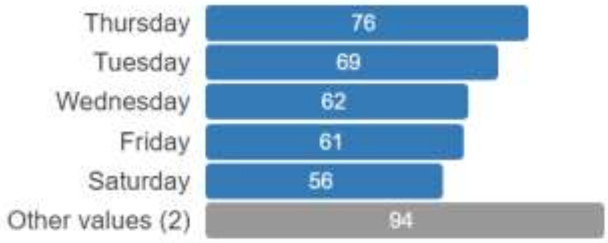
Third parties and employees have had equal frequency of accidents. But one important observation is that remote locations contributed to ~13% of incidents. This clearly indicates that some mining sites have accident prone work areas.

Date - Day, week, month, year and seasons:



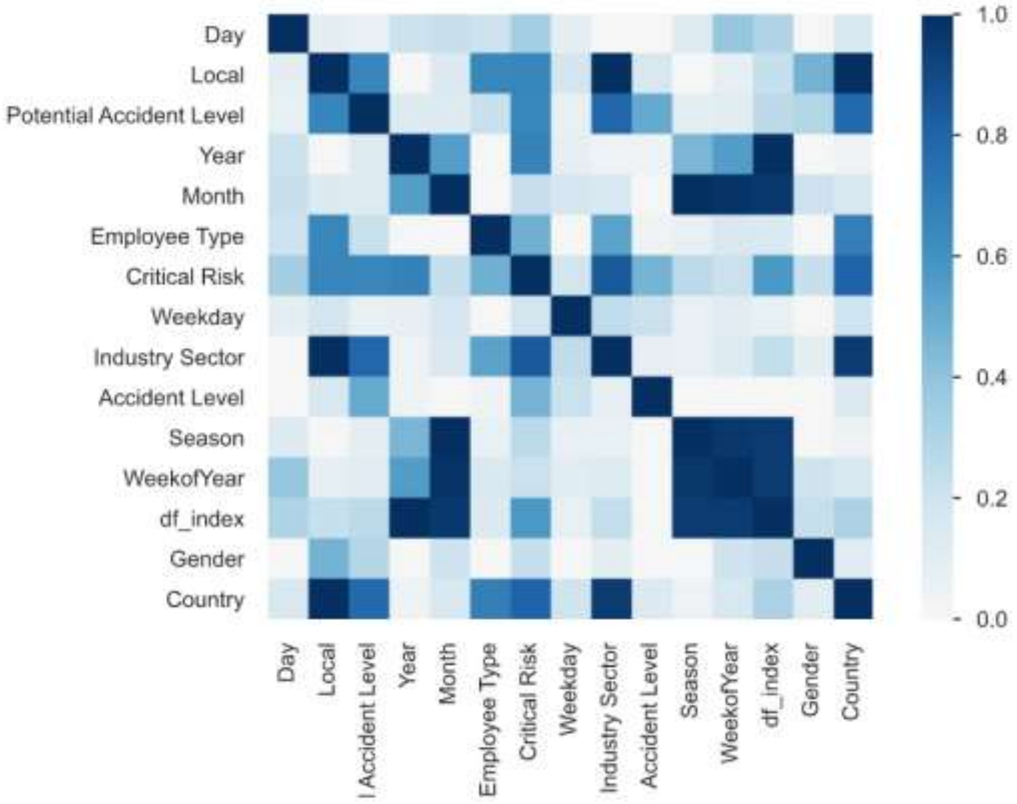






Between 2016 and 2017, the recorded data is for the full year of 2016 whereas 6 months for 2017. So, most of the individual analysis will always have this imbalance. If any seasonality affects the accidents, we can check in statistical tests.

We did correlations for various date columns and checked. It was a weak correlation between all variables.

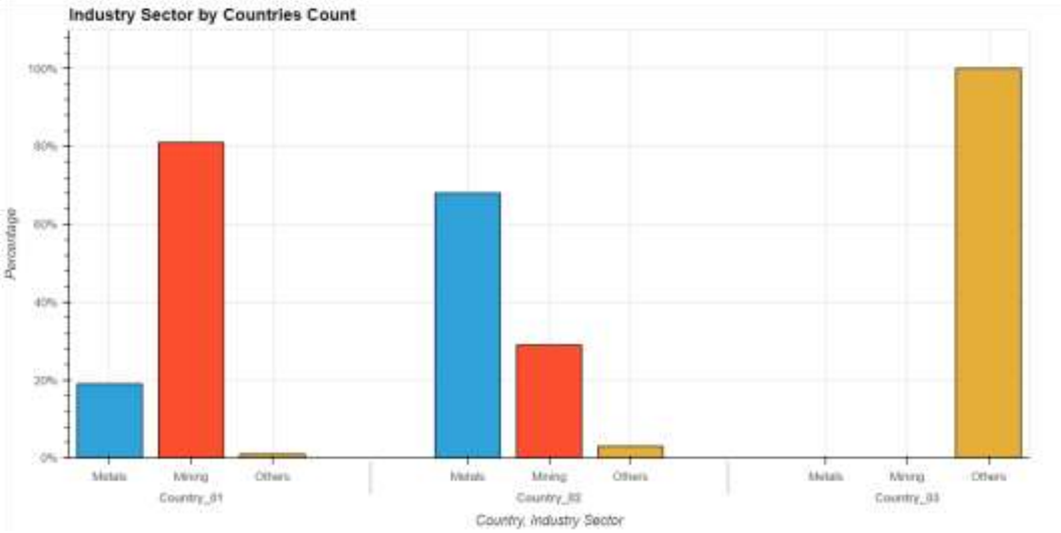


2.3 Visualisation Insights (Bivariate analysis) and Statistical Analysis:

2.3.1 Checking other bivariate relationships

- Country_01 : Mining
- Country_02 : Metals
- Country_03 : Others

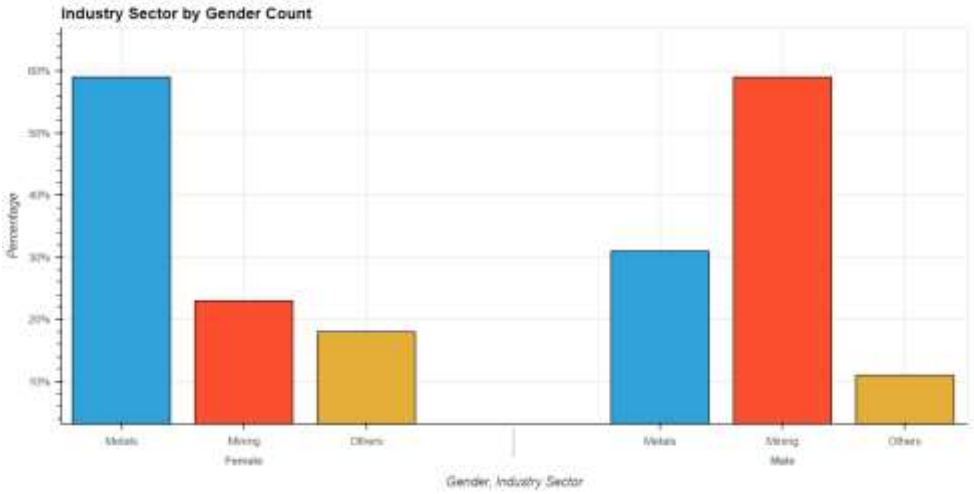
The below is the distribution of industry segments for each of the country data. Mining is highly present in Country 01 for which accident levels are recorded.



Let’s try slicing the data Gender-wise with respect to the Industry Sector.

We observe:

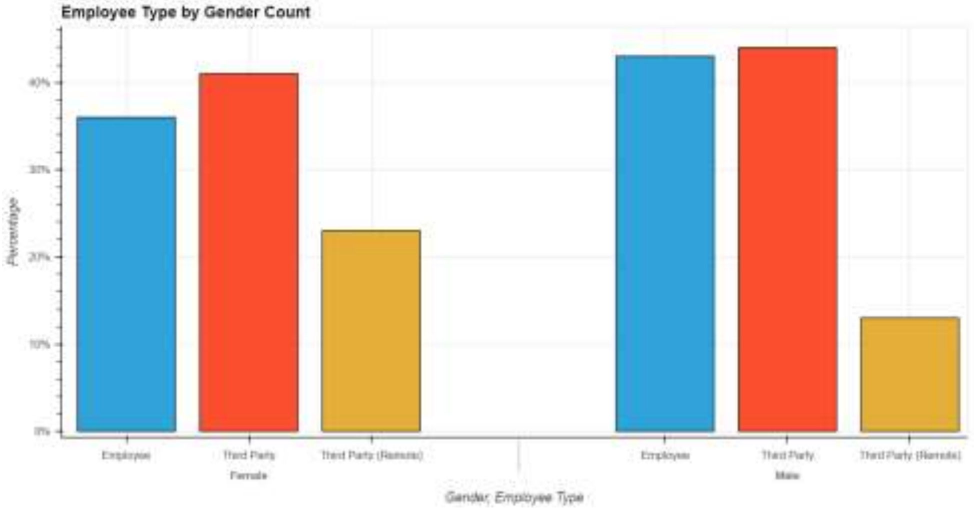
- Mining activity is majorly carried out by Male and Metals by females. In other words, Mining environments are largely male dominated because of their remote work areas. The Metal industry is plant or factory set up; hence the presence of females is fairly higher there. It is just a possible conjecture but no significant evidences seen or observed through data.



Let's further slice Employee Type with respect to Gender:

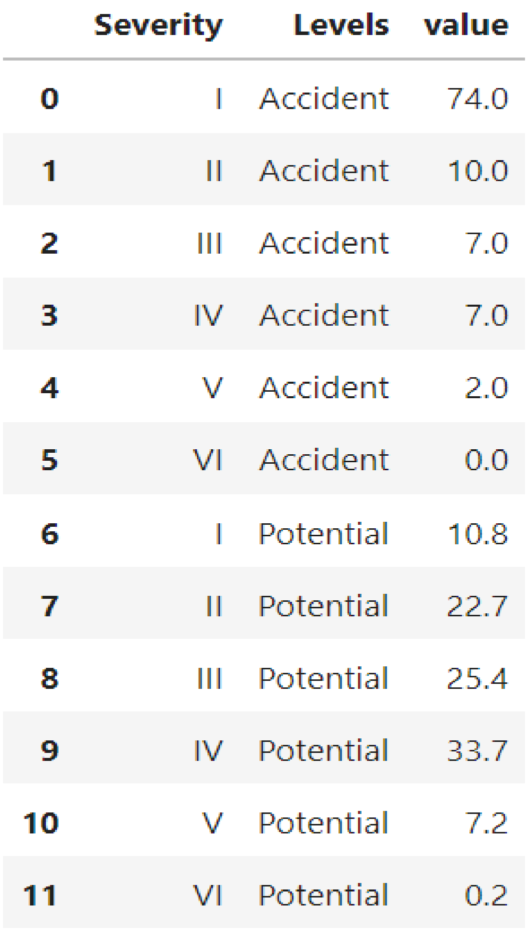
We observe:

- There is no significant difference in ratio of employee types to gender
- The proportion of females with Third Party (Remote) is slightly higher than that of males.
- It could be a conjecture that companies tend to employ female contractors in difficult or dangerous work environment than using their own female employees

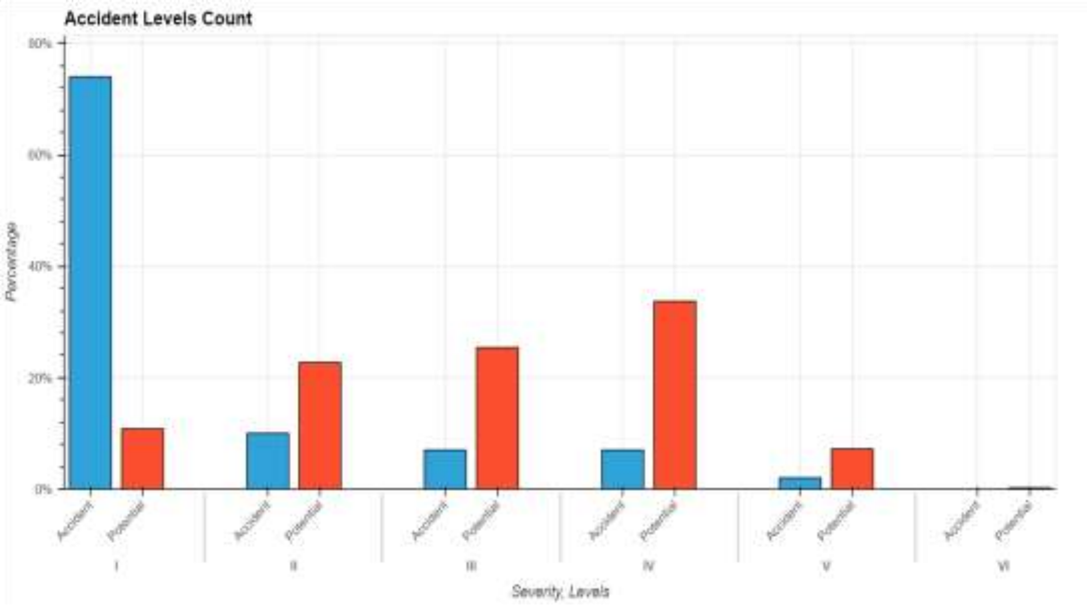


2.3.2 Exploring association with Potential Accident Level and Accident Level

In the last section, we saw some contrasting finds on accident level and potential ones.



	Severity	Levels	value
0	I	Accident	74.0
1	II	Accident	10.0
2	III	Accident	7.0
3	IV	Accident	7.0
4	V	Accident	2.0
5	VI	Accident	0.0
6	I	Potential	10.8
7	II	Potential	22.7
8	III	Potential	25.4
9	IV	Potential	33.7
10	V	Potential	7.2
11	VI	Potential	0.2



Assessing the trend between Potential accident level and Accident level, we can infer

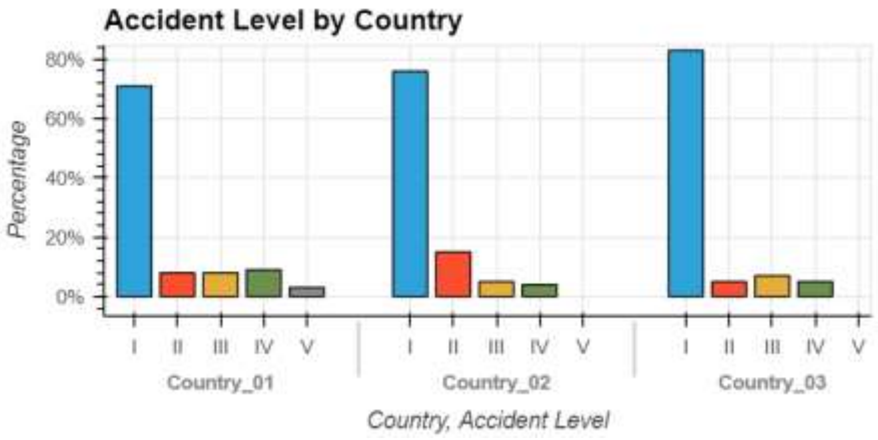
- The accident levels being recorded as “Low” may in turn be severe ones. That is an important observation for us.

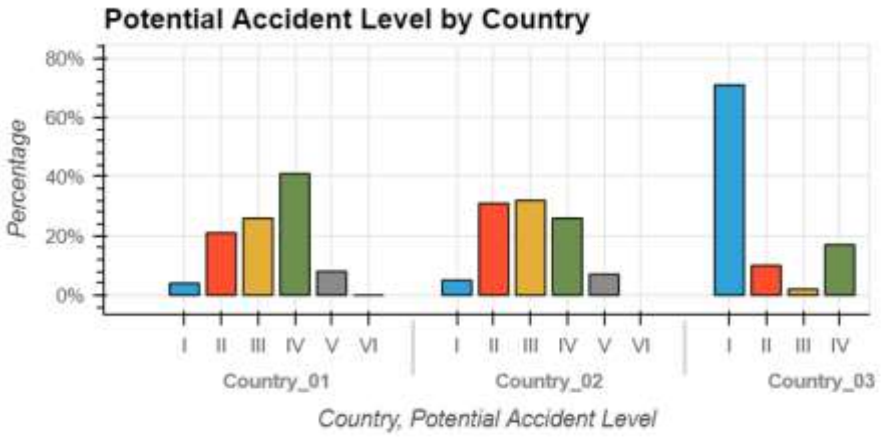
- This may be due to “Potential Accident Level” being overlooked and potentially high-risk accidents are possible.

  The other way of looking at this data point is: The accident level is recorded as per the incident reported. Whereas Potential accident level is what might have been the real severity. The below data point shows that for most of the low severe accident levels recorded, the potential accident levels might have been higher. This is true for all accident levels. That means, potential accident level is a much better variable to be predicted. This can be our target variable.

  We will choose both *Potential Accident Level and Accident levels* as our *Target variable. We will run multiple models to predict them separately. Using LSTM model, we will predict the accident level and using ML model, we will predict potential accident level with other parameters. We are looking at a Multi classification problem here, since Potential Accident level variable has more than 2 values.*

Exploring the association of all the variables with regards to Accident level and Potential Accident Levels

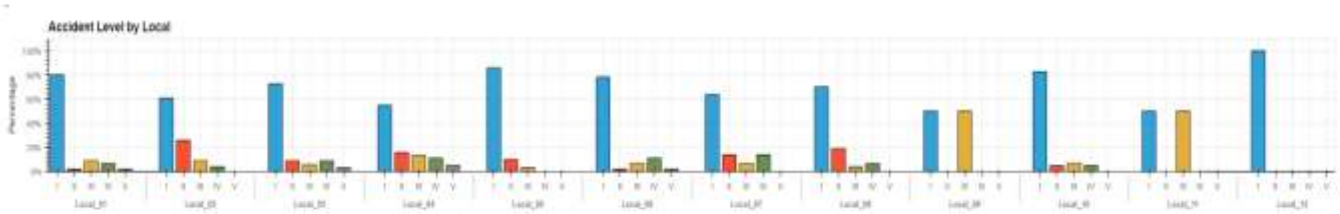


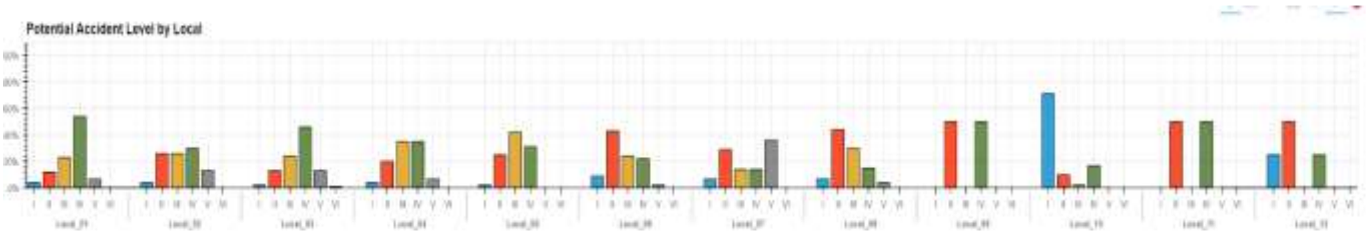


Inferences:

- All countries have significantly higher Severity - I (Lowest severe) counts, whereas if you see potential accident levels, it is leaning more towards Severity Level III and IV.
- This again validates our observation that the potential accident level is a much more valid target.

Locals and Accident levels

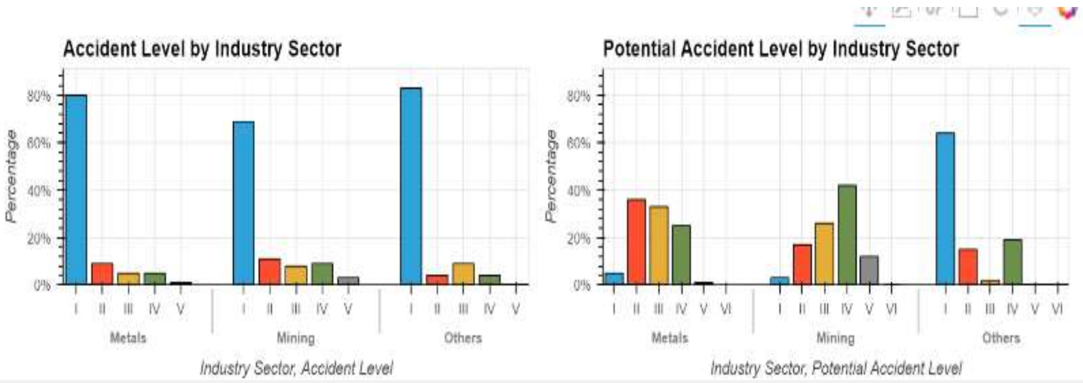




Inferences:

- Similar inferences as above can be observed and derived.
- The local_09 and local_11 has an equal number of recorded III incidents. This is also changed in potential accident levels to IV.
- Local_10 seems to be incurring the maximum least severe incidents. Are they imparting better practices or are their training programs better or are their work conditions safe? We do not have that much data to validate or observe such outlier behaviour.

Industry Sector

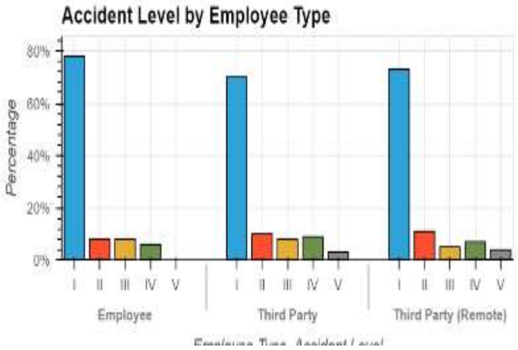


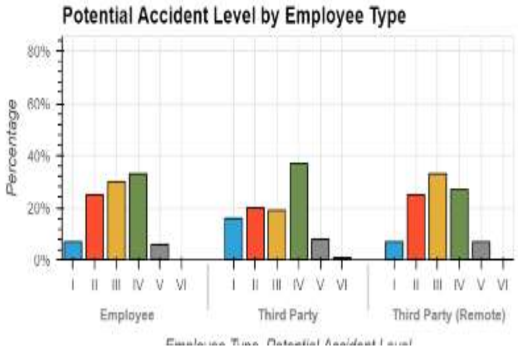
Inferences:

- The potential Accident levels III and IV are higher in the Mining area whereas in Metals, Levels II and III are higher.

- Other Industries do not encounter many higher severe accidents than mining and metals. Mostly (60%) it is all Least severity ones.

Employee Type:


Accident Level by Employee Type


Potential Accident Level by Employee Type

Inferences:

- Potential Accident levels are almost equally distributed for severity level II, III and IV for both third party contractors and employees. The remote third party also encounters more higher severity incidents.
- One thing can be inferred is employee type is important to be recorded but incidents occur across various types. There aren't specific observations which lead us to believe employees have a lower rate of severe incidents than third parties or vice versa.
- This also may indicate a possible conjecture that the training program imparted to any employee is almost the same. But focus should be given to improving the training program for any employee on how to work safely in such a dangerous environment.

Critical risk factors

Inferences:

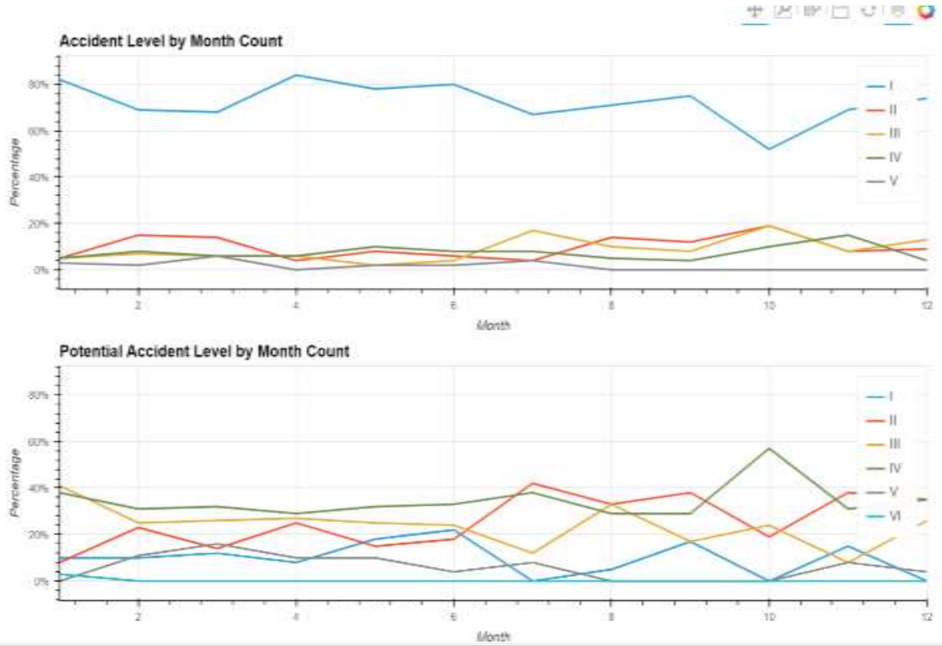
- Many not applicable categories have had potentially severe accidents than what was recorded. This may be derived from text processing which we will analyse in the below sections.
- Bees attack typically suggest it's a lower grade severity
- Burns category - can be potentially higher severe ones than what was recorded. The severity of burn degree may be derived from text analysis.
- Some Chemical Substances can lead to higher severe accidents in potential accident levels.
- Electrical shock - higher severe incidents. The recorded severity levels are lowest. It certainly requires an update to SOP or better training programs.
- Confined Spaces, Cut or a Fall category is generally recorded as low severe apart from few serious cases whereas in reality, this can be a potentially higher severe one

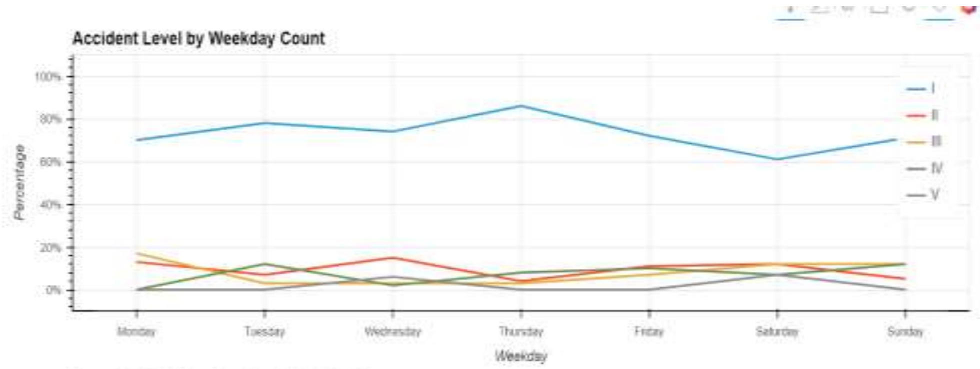
- The Liquid Metal category has potentially higher severe category ones than the recorded ones.
- *The Poll category has been recorded and observed in the right way. Is it that the systems or SOPs are better maintained for this type of incident?*
- *Powerlock category is typically observed as potentially the highest severe* incident type. Even some of the recorded observations validate that. As in the above point, *are the systems or SOPs better maintained for this type of incident?*
- Electrical installation - again a higher severe category was observed than what was recorded. Even in recorded cases, it was all severity - 4 ones. Maybe a little better training or updating SOP may rectify this error. It is a possible conjecture.
- Venomous animals can have slightly higher severe categories. This need to be observed from text processing
- Suspended loads also are treated as higher severe incidents than what was recorded. Again, few changes to SOP and text analysis can lead us to right accident level determination.

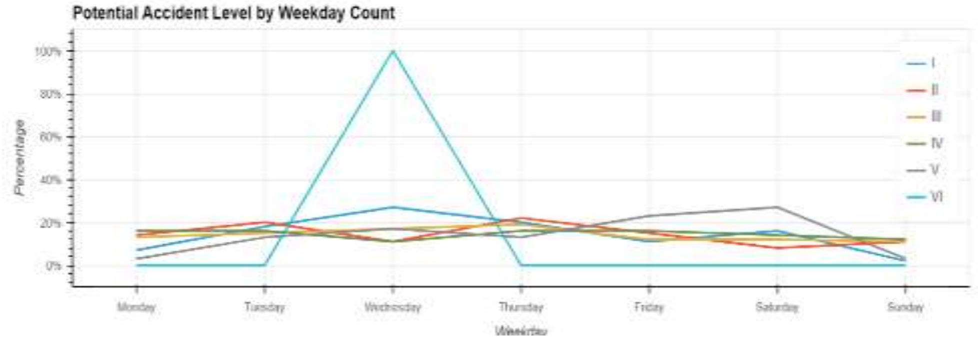
This critical risk factor parameter is really important. It tells some important observations with regards to either limitations in training program/ SOPs or the analysis at first level recording information was poor.

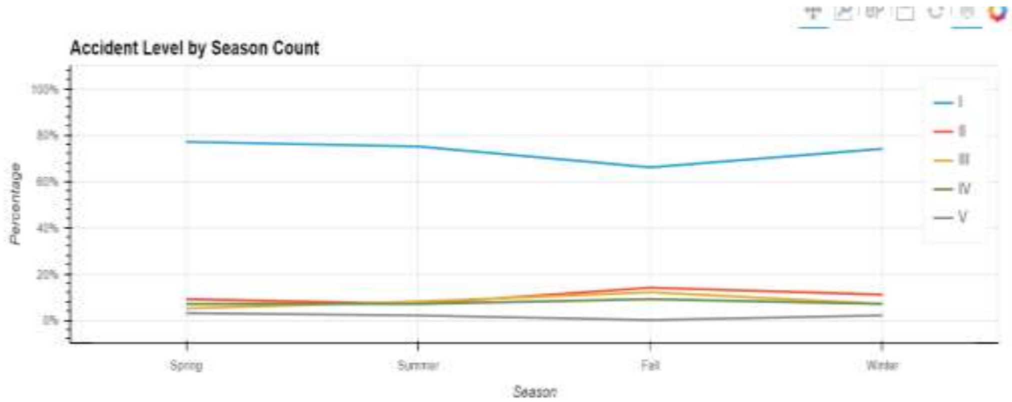
This can be a potential case to be automated using text analysis and running through our models to have better prediction of accident levels. Once the accident levels are identified closer to the right level, the treatment towards it can be addressed. Lot of data analysis to such levels can potentially save the people working in such an environment.

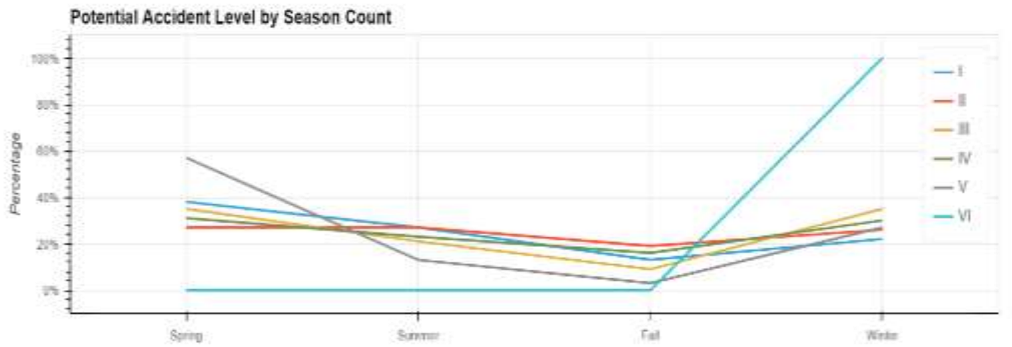
<u>Date columns</u>











Inferences:

- There are no observable patterns with regards to month or week day or seasons. Only potential accident level VI shows some higher variations in the winter season. But only one record is there for potential accident level VI, so we need not worry too much on these parameters.
- As earlier said, the date column is important for recording purposes but may not have a significant impact on accident levels or incidents occurring. We are not seeing a pattern here to have deeper analysis. Nor if any higher or

lower incident pattern seen can be validated or analysed through other data points.

- One study that can emerge from this analysis of date columns: We can go back and see the logs in specific locals or countries and record if they did something different that particular day, week or month. But that is not the scope of this document or project.

Conclusion of Bivariate analysis:

- Association of various parameters like Date, Country, Locations, Employee type, Gender, Critical Risk factor on both accident and potential accident levels throw some important observations which we have highlighted in this section.
- Are these mere parameters or are they significantly influence or impact accident levels will be addressed through statistical tests in the next section?

2.4 Statistical Tests

Since all are categorical variables, we will use the chi-square test for finding the statistical significance of each of them on the potential Accident levels. We will try to formulate null and alternate hypothesis and run statistical tests using chi-square and determine whether to reject or accept null hypotheses.

Potential Accident Level & Countries:

- Ho = The proportions of **Potential Accident Level** is not different in different **countries**
- Ha = The proportions of **Potential Accident Level** is different in different **countries**
- Decide the significance level: alpha = 0.05

We observe:

It is observed that the p-value is 4.1017395785840915e-34

At 0.05 significance, as p-value<level of Significance, we reject the null hypothesis.

i.e., The proportions of Potential Accident Level is different in different countries.

we also have observed that Higher potential accident level is occurring more in Country_01. Statistically significant difference between countries.

Our Conclusion: This shows that countries are an important parameter for our potential accident level prediction. This will be a part of our analysis.

Potential Accident Level & City:

- Ho = The proportions of **Potential Accident Level** is not different in different **City**

- Ha = The proportions of **Potential Accident Level** is different in different **City**
- Decide the significance level: alpha = 0.05

We observe:

It is observed that the p-value is 3.728976101772464e-28

At 0.05 significance, as p-value<level of Significance, we reject the null hypothesis.

i.e., The proportions of Potential Accident Level is different in different City

Potential accident level (PAL) is different from location to location.

- PAL I is unusually higher in L10
- PAL II is high L3, L4, L5, L6, L8
- PAL III, IV are high in L1, L3, L4, L5
- PAL V is unusually high in L7

Our Conclusion: City or locations is an important feature that will be required for analysis.

Industry Sector & Potential Accident Level:

- Ho = The proportions of **Potential Accident Level** is not different in different **Industry Sector**
- Ha = The proportions of **Potential Accident Level** is different in different **Industry Sector**
- Decide the significance level: alpha = 0.05

We observe:

It is observed that the p-value is 1.992985715278311e-36

At 0.05 significance, as p-value<level of Significance, we reject the null hypothesis.

i.e., The proportions of Potential Accident Level is different in different Industry Sector

Higher potential accident levels are observed in mining, intermediate in metals sector compared to other sectors. Statistically significant difference between industry sector

Conclusion: Industry sector is an important feature to be considered for analysis.

Gender & Potential Accident Level:

- Ho = The proportions of **Potential Accident Level** is not different in different **Gender**
- Ha = The proportions of **Potential Accident Level** is different in different **Gender**
- Decide the significance level: alpha = 0.05

It is observed that the p-value is 0.0003681748355215793

At 0.05 significance, as p-value<level of Significance, we reject the null hypothesis.

i.e., The proportions of **Potential Accident Level** is different in different **Gender**

Conclusion: Gender will still be an important parameter despite the data imbalance for each type.

Critical Risk & Potential Accident Level:

- Ho = The proportions of **Potential Accident Level** is not different in different **Critical Risk**
- Ha = The proportions of **Potential Accident Level** is different in different **Critical Risk**
- Decide the significance level: alpha = 0.05

We observe:

It is observed that the p-value is 6.463580982130032e-22

At 0.05 significance, as p-value<level of Significance, we reject the null hypothesis.

i.e., The proportions of **Potential Accident Level** is different in different **Critical Risk area**

Conclusion: Critical Risk area is also important parameter for analysis

Weekday & Potential Accident Level:

- Ho = The proportions of **Potential Accident Level** is not different in different **Weekday**
- Ha = The proportions of **Potential Accident Level** is different in different **Weekday**
- Decide the significance level: alpha = 0.05

We observe:

It is observed that the p-value is 0.30204663583713837

At 0.05 significance, as p-value>level of Significance, we accept the null hypothesis.

i.e., The proportions of **Potential Accident Level** is not different in different **Weekday**

All Potential Accident levels are typically low on Sundays, yet the difference is not statistically significant

Conclusion: While week day or date is not important and since dates recorded for 2016 for a full year whereas only 6 months data recorded for 2017 . Week day or Date may not be that significant parameter for analysis. Date is needed for recording the accident information. It does not have much significance beyond that. Nor any pattern seen towards specific days or time or seasonal variation is affecting the accident.

Month & Potential Accident Level:

- Ho = The proportions of **Potential Accident Level** is not different in different **Month**
- Ha = The proportions of **Potential Accident Level** is different in different **Month**
- Decide the significance level: alpha = 0.05

We observe:

It is observed that the p-value is 0.30204663583713837

At 0.05 significance, as p-value>level of Significance, we accept the null hypothesis.

i.e. The proportions of **Potential Accident Level** is not different in different **Month**

Like above point, not specific analysis or statistical significance seen for Month column. But it is required to be captured as part of date column.

2.5 Conclusion of EDA:

- EDA analysis has thrown some important decisions on our parameters along with potential accident levels. This will help us in building our chatbot utility to see which parameter will be significant and should be asked.
- We concluded that Accident level Potential accident level will be our target variable for different models.
- Like we said above, “Date” does not add much value but can be used for only recording purposes.
- Gender can be important parameter although it has data imbalances.
- Higher potential accident level is occurring more in Country_01. Statistically significant difference between countries.
- Statistically significant difference between various locations or locals.
- Higher potential accident level is more in mining, intermediate in metals sector compared to other sectors. Statistically significant difference between industry sectors.
- All potential accident levels were more in males. Statistically significant difference between gender.
- Having observed many of these and while these parameters are important, these alone cannot predict the potential accident level. They are still categorical and there is no sure way of using such categorical variables to predict the accident levels.
- Hence, we require text processing or analysis on the description to enhance our prediction of potential accident levels. This will be taken in the next section.

- **Why and how EDA is important for our Chatbot utility: Chatbot will be the final output that would be visible to end users. All these statistical analysis shows an important aspect for our Chatbot utility. This will aid in designing our chatbot questions or GUI questions to take input on all these parameters that can potentially impact or predict the potential accident levels.**

2.6 Text Data Pre-processing: Description Variable or Field

This section will detail the entire text preprocessing of the “Description” field or variable. There are a lot of insights that can be derived from text analysis. Our conjecture is reading through detailed descriptions of the accidents might have enabled the data collection team to add a potential accident level column manually later. Hence, it is all the more important and essential to analyse the “description” column to infer more about the accident level and potential accident levels.

NLP preprocessing steps are taken sequentially before applying the model on the data. The sequential processes are:

- Remove stop words
- Tokenize the words
    - Convert all words to lowercase ones, avoid any capital cases
    - Converting apostrophe to the standard lexicons
    - Removing punctuations

- Lemmatization & Stemming
    - stemming technique only looks at the form of the word whereas lemmatization technique looks at the meaning of the word. It means after applying lemmatization, we will always get a valid word
- Removing non-alphabetical characters like '(', '.' or '!'
    - We have also observed many punctuations, alphanumeric characters and special characters. These add no significance to prediction. Hence, removing them will help

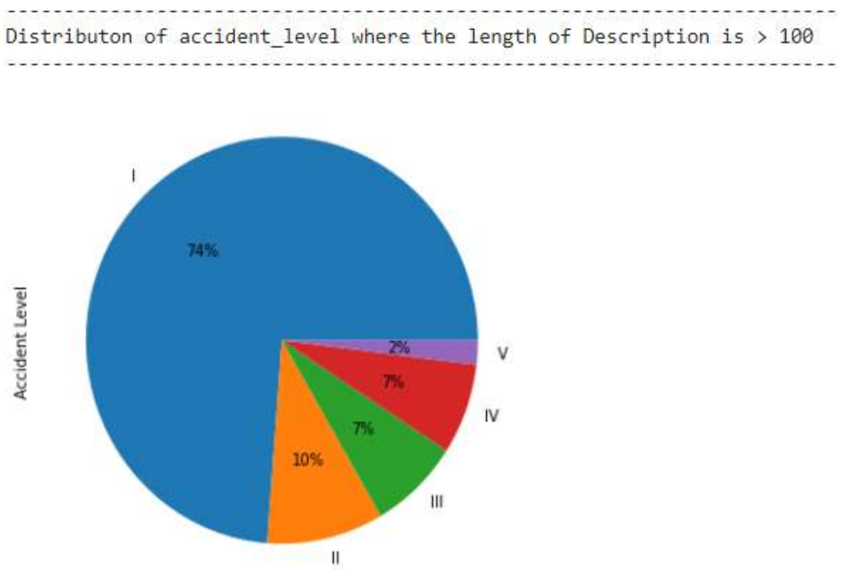
The first step in any text processing is removing of stop words. These stop words are commonly found in any vocabulary and do not amount to any prediction of target variable. Our target variable is accident level in text analysis.

We removed all these stop words from the description column.

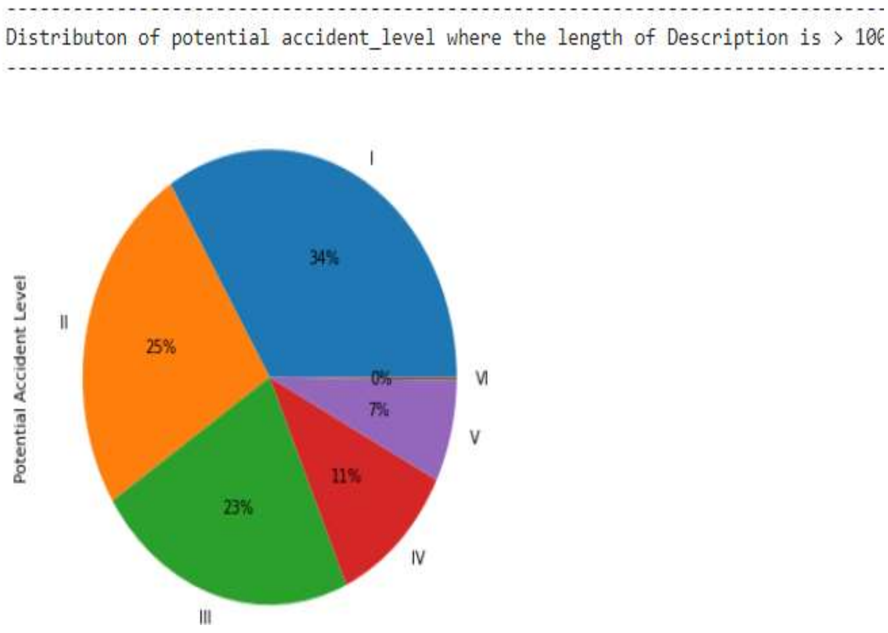
{"didn't", 'more', 'she', 'here', 'if', 'he', 'we', "aren't", 'a', "hasn't", "she's", "they'd", "you're", 'its', 'further', "he'd", 'which', 'kg', 'an', 'had', "don't", 'ever', 'too', 'after', 'until', 'herself', 'why', 'yours', 'this', 'of', "he'll", "she'll", "we've", "it's", 'has', 'and', 'does', 'his', 'been', 'me', "won't", 'since', 'am', 'during', "wouldn't", "i'm", 'with', 'himself', 'i', 'but', 'r', 'few', 'just', 'out', 'the', 'by', "that's", 'should', 'in', 'to', 'at', "i'll", 'down', 'once', 'my', 'else', "we're", 'your', 'up', 'did', 'k', 'yourselves', 'same', 'each', 'most', 'nor', 'only', "they'll", "i've", "they've", 'ourselves', 'both', "haven't", 'there', "you'll", 'be', 'like', 'ours', 'under', "how's", 'having', 'is', 'over', "here's", "shan't", 'these', "i'd", 'where', 'him', 'were', "shouldn't", 'into', 'being', "mustn't", 'below', 'as', 'they', 'was', 'all', 'otherwise', 'while', "where's", 'because', "let's", 'those', 'do', "he's", 'off', 'own', 'not', 'whom', 'also', 'however', 'who', 'pm', 'so', "when's", 'yourself', "couldn't", 'get', 'cannot', 'when', 'again', 'http', 'for', 'from', 'how', "we'd", 'then', 'doing', 'about', 'cm', "wasn't", 'than', 'myself', 'other', "what's", "who's", 'above', 'it', 'can', 'itself', "why's", 'on', 'com', "isn't", 'very', 'ought', 'through', 'some', 'would', "she'd", "hadn't", 'shall', 'no', "there's", 'against', "can't", 'their', 'you', 'our', 'such', 'are', 'them', 'themselves', "you've", 'could', "they're", 'before', 'any', 'her', 'www', "weren't", "doesn't", "we'll", 'that', 'what', "you'd", 'between', 'or', 'have', 'hers', 'theirs'}

2.6.1 Observations basis the analysis of the error description and distribution of accident level

1. Let us analyse a random sample of 5 accident descriptions where the length of the accident description is greater than 100.



```
------------------------------------------------------------------------
Distributon of accident_level where the length of Description is > 100
------------------------------------------------------------------------
```



```
---------------------------------------------------------------------------
Distributon of potential accident_level where the length of Description is > 10
---------------------------------------------------------------------------
```

Inferences:

- If you see some description texts, they are referring to some machine models with numerical digits. In singularity, the numerical or model number may not be useful for prediction. However, the entire machine model along with Machine name may be useful for prediction. We will use n-gram and other models below in coming sections.

- A random analysis on length of description column over 100 words, shows some 74% cases are recorded as accident level - I.
- The same is reduced for potential accident level reduced to 34%.
- Although this has no direct correlation, what one can see is whether a deeper description over many words will yield the results differently. It is not seen or observed. We will head to other analysis

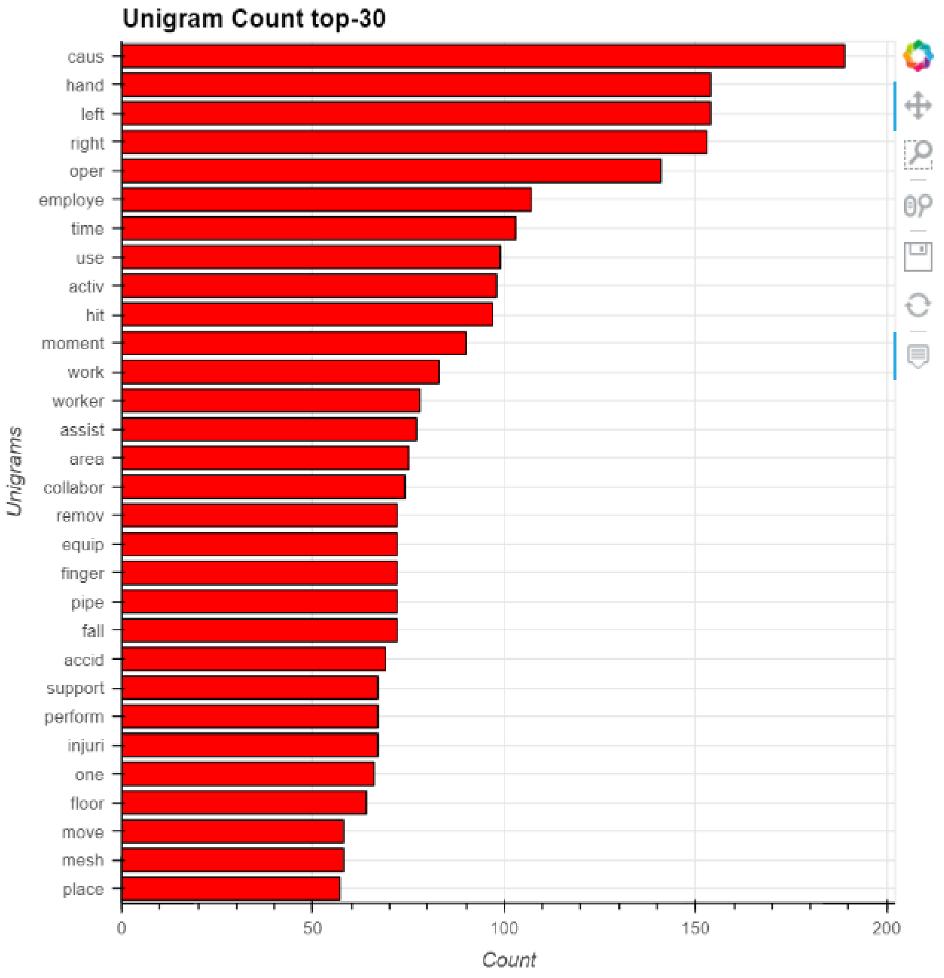
2.6.2 Text Pre-processing and Exploration

N-Gram Analysis:

Description of accidents is important to understand the cause of accidents, so we need to discover characteristic words or phrases indicating the situation when accidents occurred.

N-grams is a very important tool in NLP exploration for finding sets of common co-occurring words, where *n* refers to the number of consecutives occurring words. For instance, *'left', 'hand', 'finger'* would be a 3-word n-gram or trigram.

Using the Unigram:

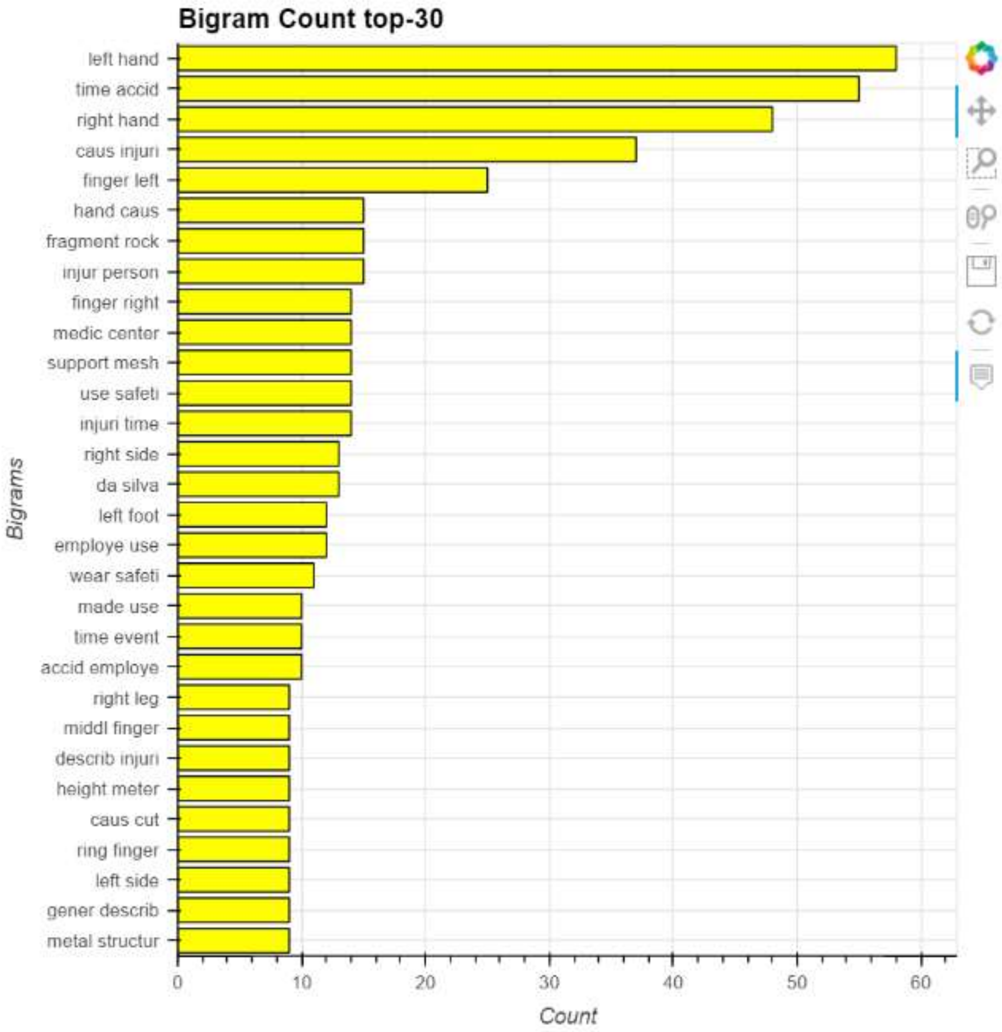


Inferences:

- We observe that top 5 words are <u>‘cause’, ‘hand’, ‘left’, ‘right’, ‘oper’.</u>

  - A hand in itself may not be a great indicator or predictor. So, we may require a bigram model to check if the right hand or left hand has any significance.
- There are several words related to hands. For example, left, hand, right and finger. How these all help us in predicting. So, should we drop this? Such an analysis will be done using n-gram models.
- Moreover, there are several words related to movement of something. For example, hit, remove, fall and move.
- Unigram model has so many top words but they are all very generic and in singularity, may not be of much use. Let us use bigram model and see if any better words can be used for our analysis.

Using the Bigram:

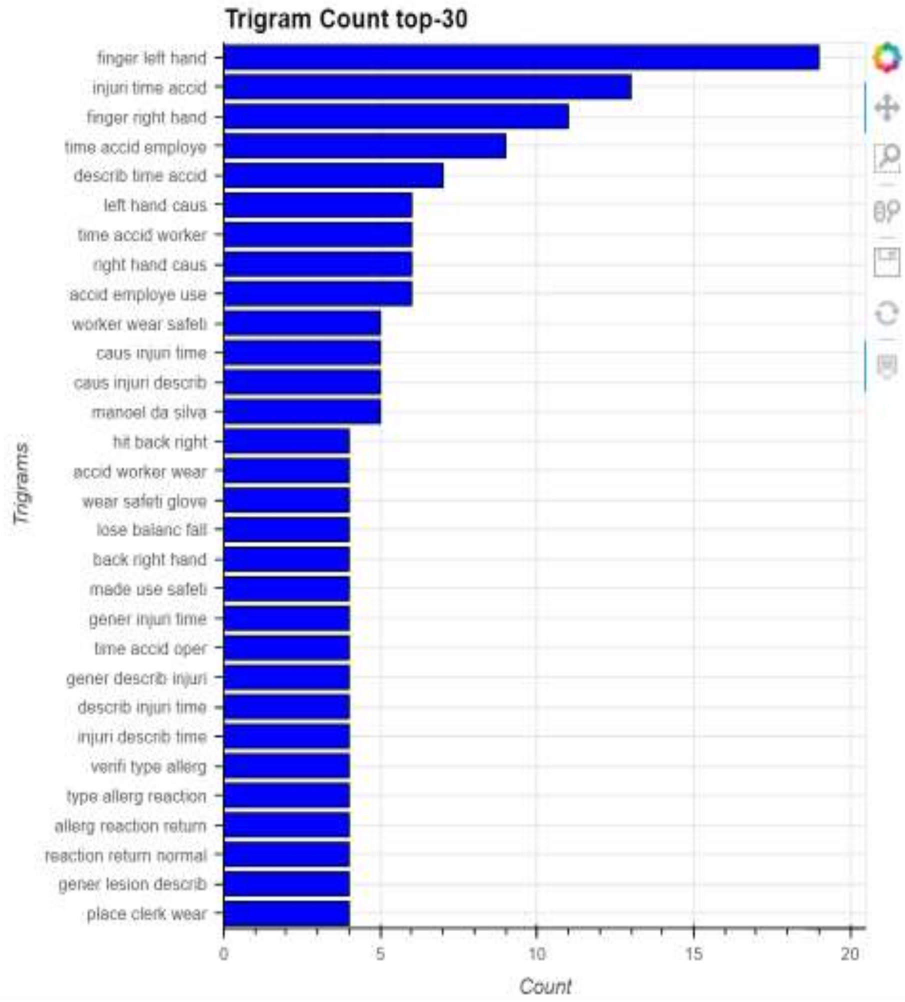


Inferences:

- There are so many phrases which are related to hands. For example: left hand, right hand, finger left, finger right, middle finger and ring finger.
- There are also some phrases which are related to other body parts. For example: left foot and right leg, right side

- How is this relevant for prediction apart from being a body part. In itself, the body parts only depict the area where injury occurred but it tells nothing about severity levels. Having said, these n-gram models help us in visualising how many times a particular body part is affected across. Here across 400 odd accidents reported, the left hand is the major one that got injured the most with 70 times. That is useful for more numerical analysis but not significant in predicting the severity levels.

Using the Trigram:



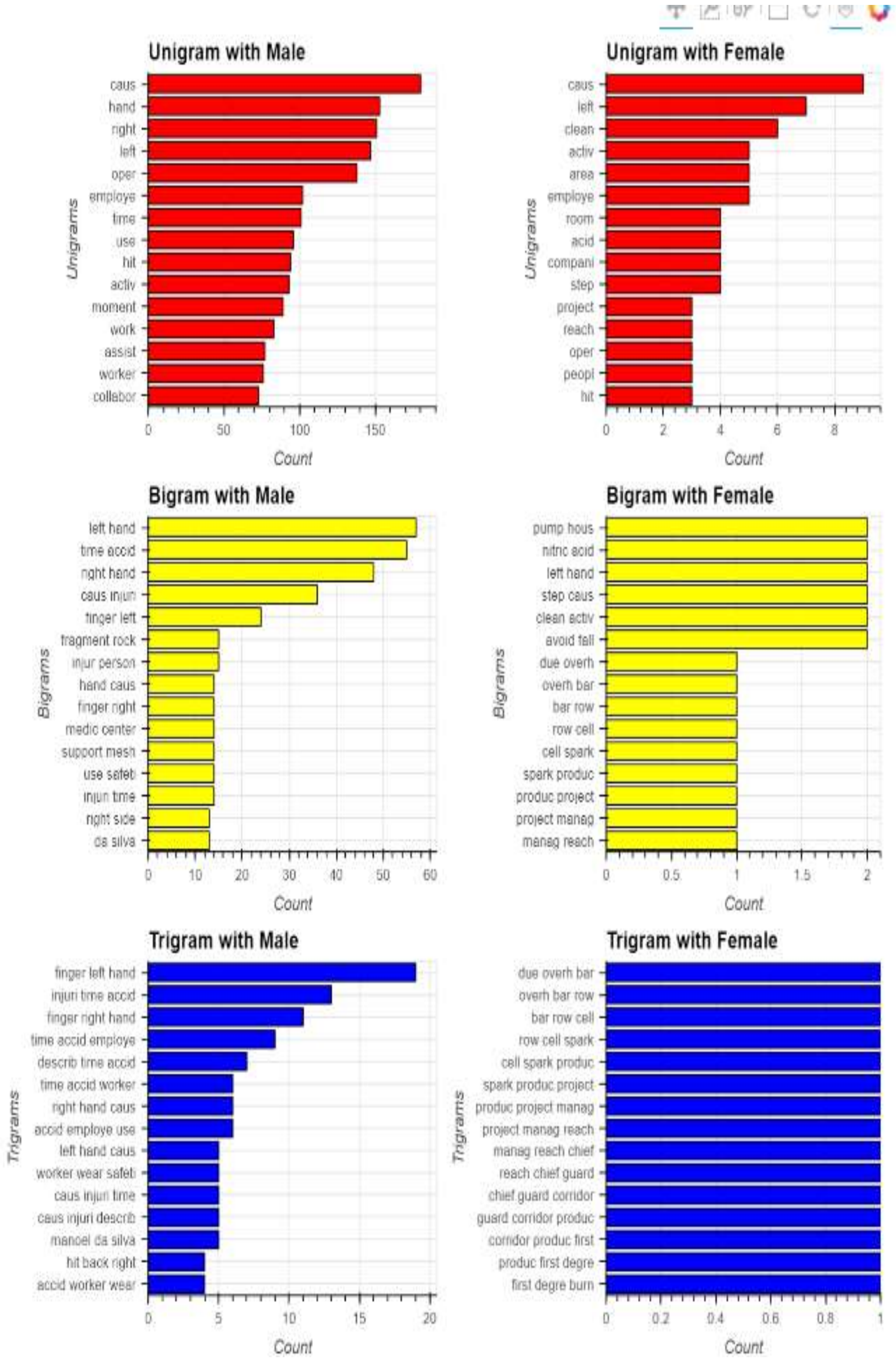
Inferences:

- Like what we found in Unigram and Bigram, there are also many phrases which are related to hands or other body parts and the confidence level increases on the body parts in which the injury is reported more. For example: - finger left hand, finger right-hand, left-hand cause, right hand cause, one hand glove
- According to Ngram analysis, we can say that operations related to hands are where the number of injuries or accidents have occurred more.

- So, it is important for us to see this n-gram analysis along with other variables. We will analyse for any patterns that can help us in going towards the accident level severity prediction.

Bivariate n-gram analysis:

Now we will try to do the n-gram analysis along with other variables like Gender, industry, accident levels etc.

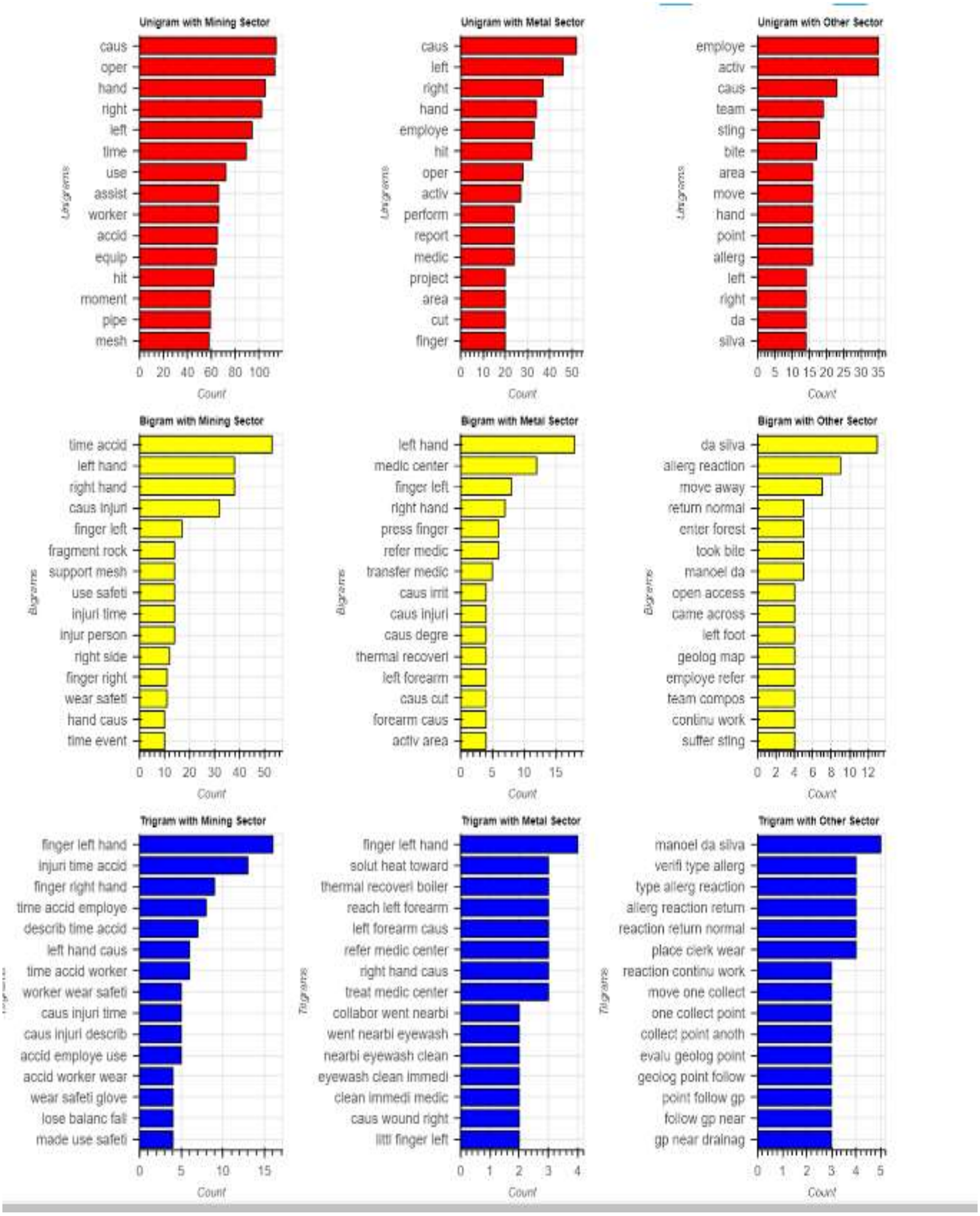


Inferences:

- It is evident from the data that the number of injured people is male and the typical injuries are related to left-hand, left-hand finger, right hand finger. There is no pattern apart from the body parts are time

- We observe that very few females are getting injured and in the similar categories like the left-hand finger. The female tri-gram also shows its more to do with spark or injury caused during production related operations.
- We observe that the injury nature within the female category is distributed evenly than in males.

N-gram analysis against the Industrial Sector as variable:

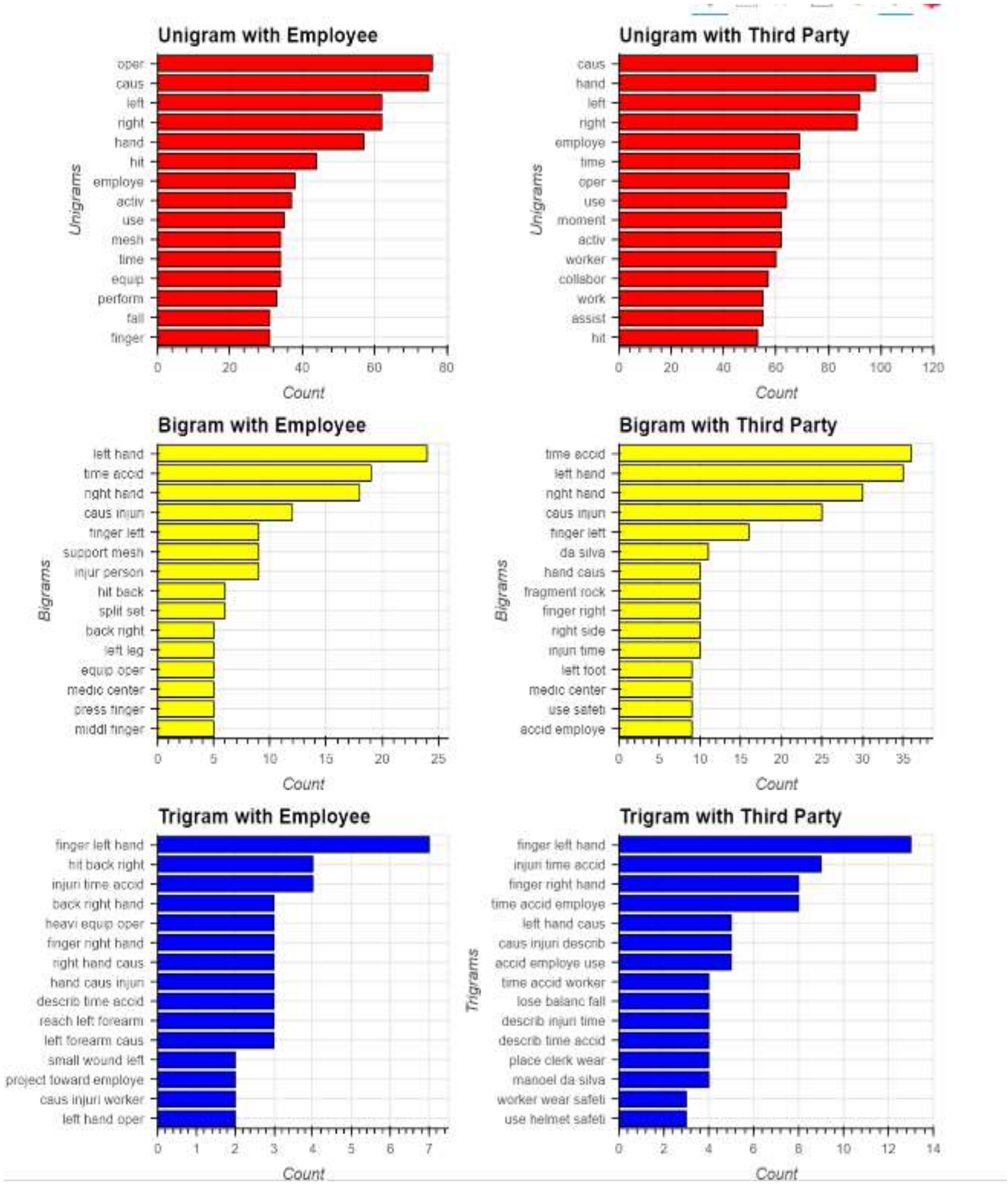


Inferences:

- Mining sector has more accidents and injuries related to left hand finger, right hand finger, workers not wearing safety uniform, workers not wearing glove
- Metal sector has less accidents and the injuries and accidents are related to left hand, solution heat, thermal recovery boiler, related to eye -wash cleaning

- Other sector injuries are very less and largely related to allergic reactions, entering the forest and took bite etc., as per the n-gram analysis.
- Mining sector uses Pipe and equipment, Metal sector shows thermal recovery boilers, eye related injuries/accidents and other sectors the accidents are observed due to employees entering forest, bite related and sting related and also related to open access/ area.

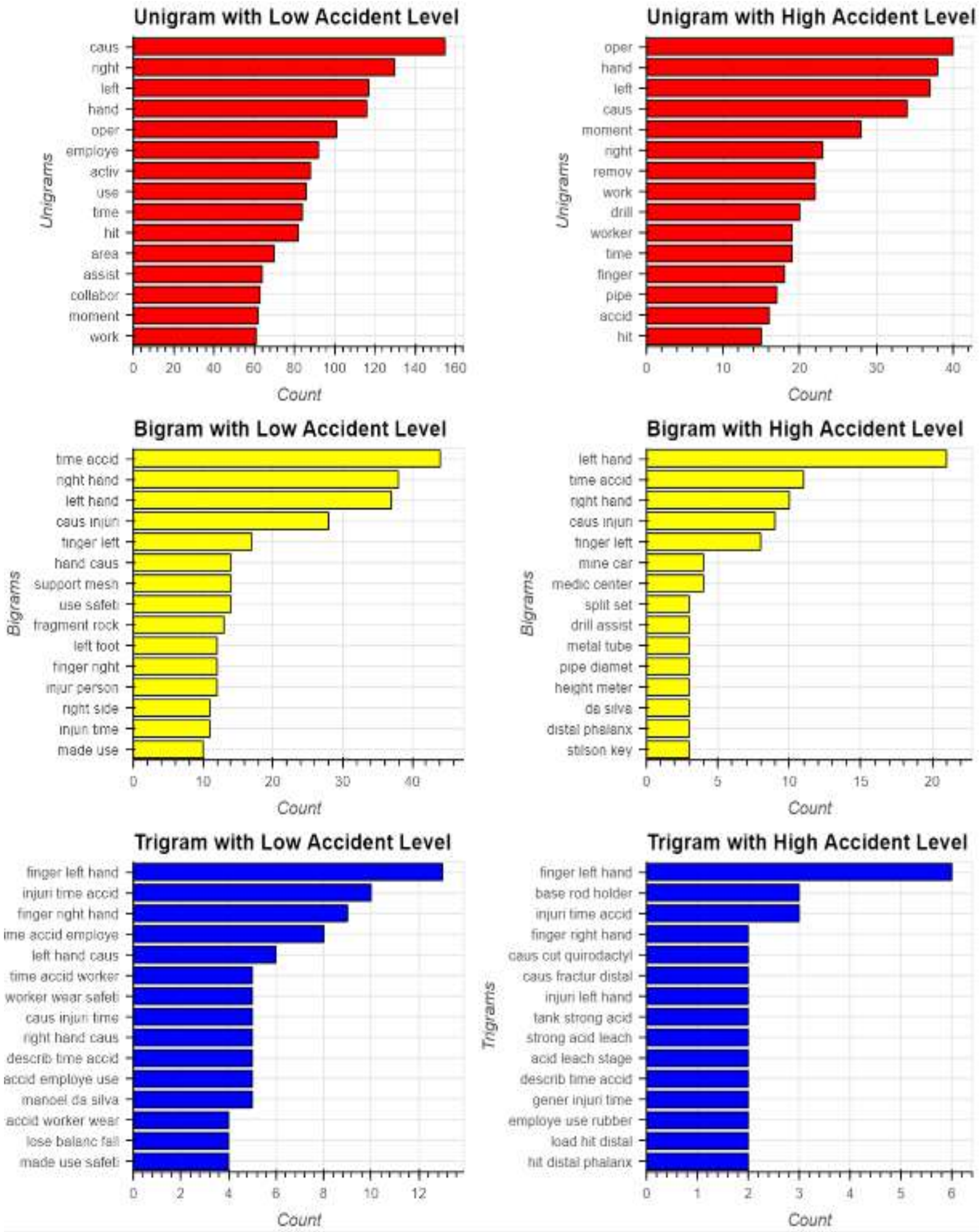
N-gram analysis with the employee type (Employee or Third Party)



Inferences:

- The observations indicate that there are more accidents in Third Party but the injury nature is similar to Employee.
- Both the types of employees face similar problems of left hand and right-hand finger injuries.
- The n-gram analysis indicates that Third Party involved in Other sectors and more injury prone due to insect bite or sting.

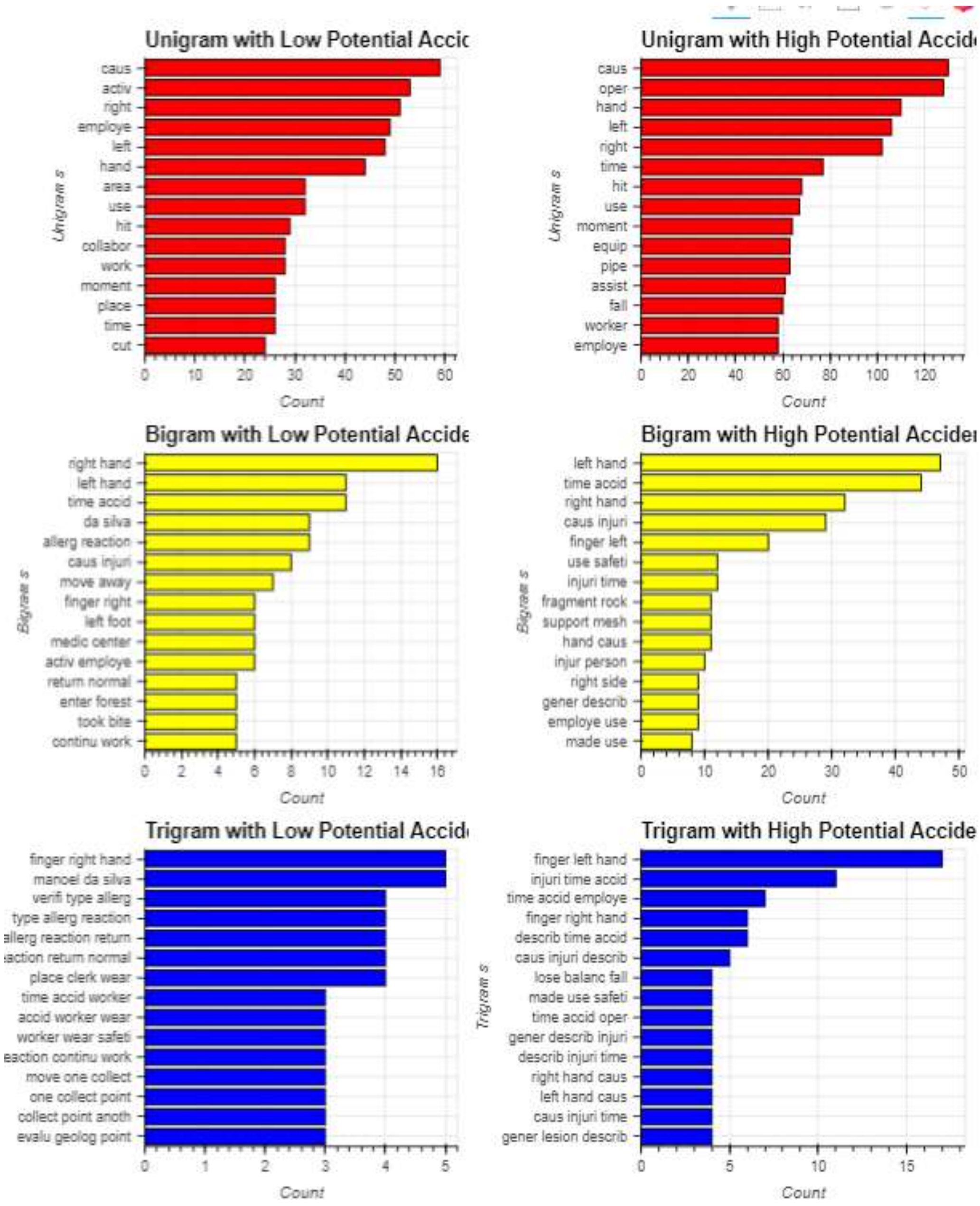
N-gram analysis with accident level as variable



Inferences:

- Let us classify the accident level as low (accident level I, II) & high (accident level II, IV, V)
- Count of Low accident levels is higher than high accident level and also it is observed that both accidents are related to left, right hand finger related injuries as the major contributors.
- High accident levels are due to Metal sector and the words such as metal rod, acid etc., are indicative of such injuries
- Here one thing to be observed is that finger left hand is present for high accident levels and also for low accident levels. That means, we need more words or more elements along with severity of accidents like a burn or a minor injury to predict the levels.

N-gram analysis with potential accident level as variable



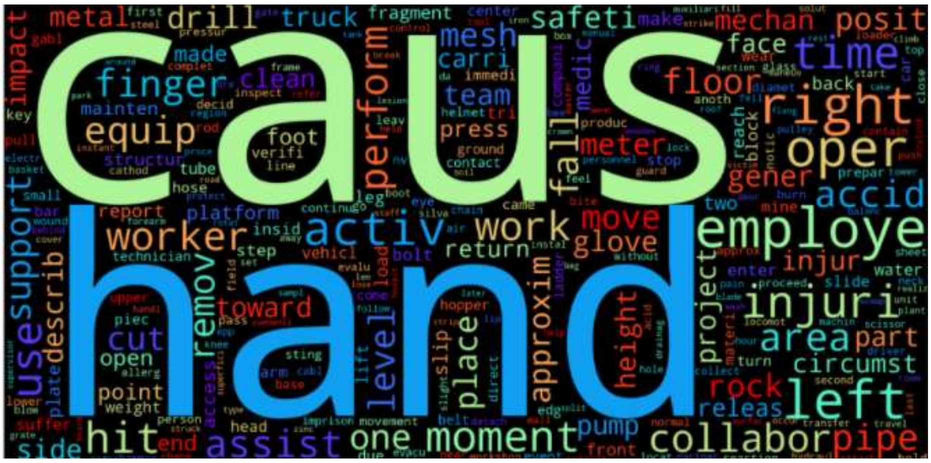
77

Inferences:

- Basis the n-gram analysis, it is observed that low potential level is due to allergic reaction, injuries to left hand finger, right hand finger
- Same observation as accident levels.

2.6.3 Observations on Word Cloud

There are many body-related, employee related, movement-related, equipment-related and accident-related words.



Inferences:

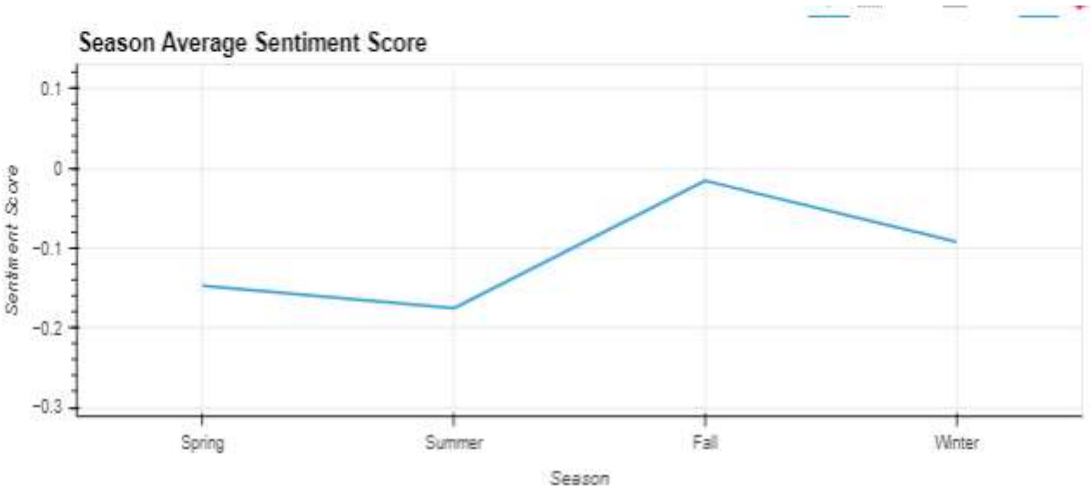
- Body-related: left, right, hand, finger, face, foot and glove
- Employee-related: employee, operator, collaborator, assistant, worker and mechanic
- Movement-related: fall, hit, lift and slip
- Equipment-related: equipment, pump, meter, drill, truck and tube
- Accident-related: accident, activity, safety, injury, causing
- Word-cloud also suggest the same set of words that have been occurring the maximum number of times.

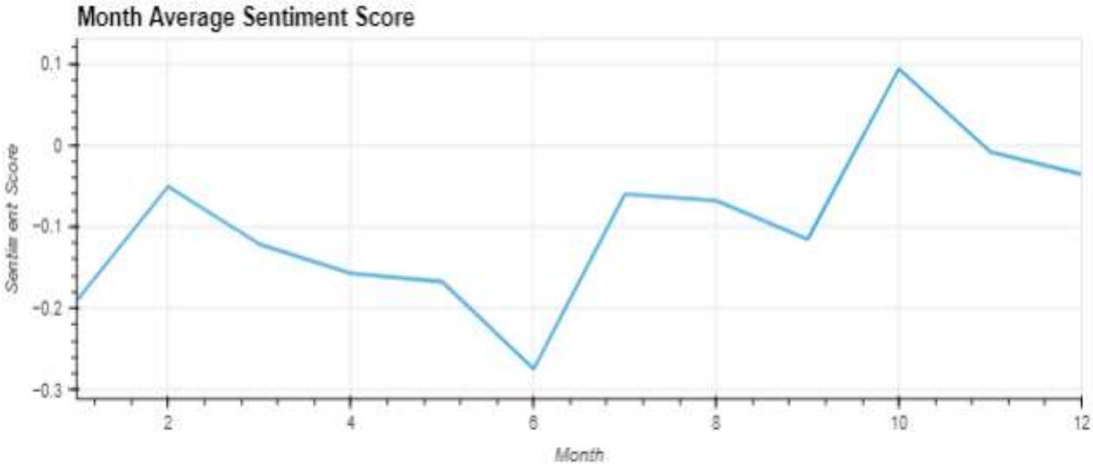
2.6.4 NLP Pre-processing Summary:

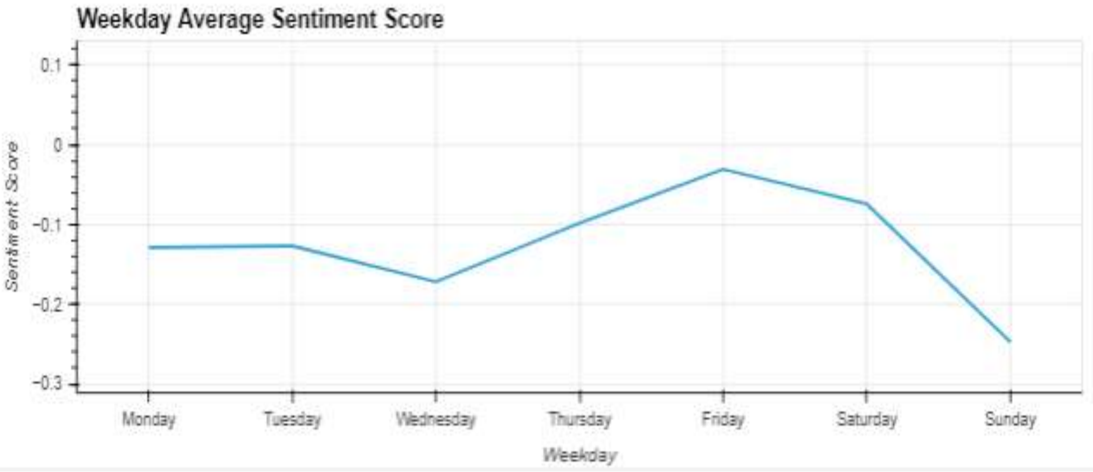
- 74% of data where accident description > 100 is captured in low accident level.
- 34% of data where accident description > 100 is captured in high medium potential accident level.
- 25% of data where accident description > 100 is captured in medium potential accident level.
- 23% of data where accident description > 100 is captured in low potential accident level.
- Few of the NLP pre-processing steps taken before applying model on the data
  - Converting to lower case, avoid any capital cases
  - Converting apostrophe to the standard lexicons
  - Removing punctuations
  - Lemmatization
  - Removing stop words
- After pre-processing steps:
  - Minimum line length: 64
  - Maximum line length: 672
  - Minimum number of words: 10
  - Maximum number of words: 98

2.6.5 Sentiment Analysis:

Based on the input variable date of the accident, a sentiment analysis was performed to determine the correlation between the accidents with day of the week, season of the year, month of the year and the accident. We built the sentiment analysis through *SentimentIntensityAnalyser* function and calculated the sentiment score through *polarity scores.*







Inferences:

- ***Sentiment basis the month of the year:***

The sentiment score is very low in June and is observed that the number of accidents is very high during that month and October the sentiment is very high and positive indicating a smaller number of accidents.

- ***Sentiments basis the season of the year***:

  Based on the low sentiment score, it is believed that the number of accidents is very high in summer and estimated to be low in fall season.

- ***Sentiments basis the day of the week:***

Based on the average sentiment score for each day of the week, it is evident that Friday being less accident prone and Sunday indicating more employee risk due to the number of accidents being high.

2.6.6 Feature Engineering - Embedding Models:

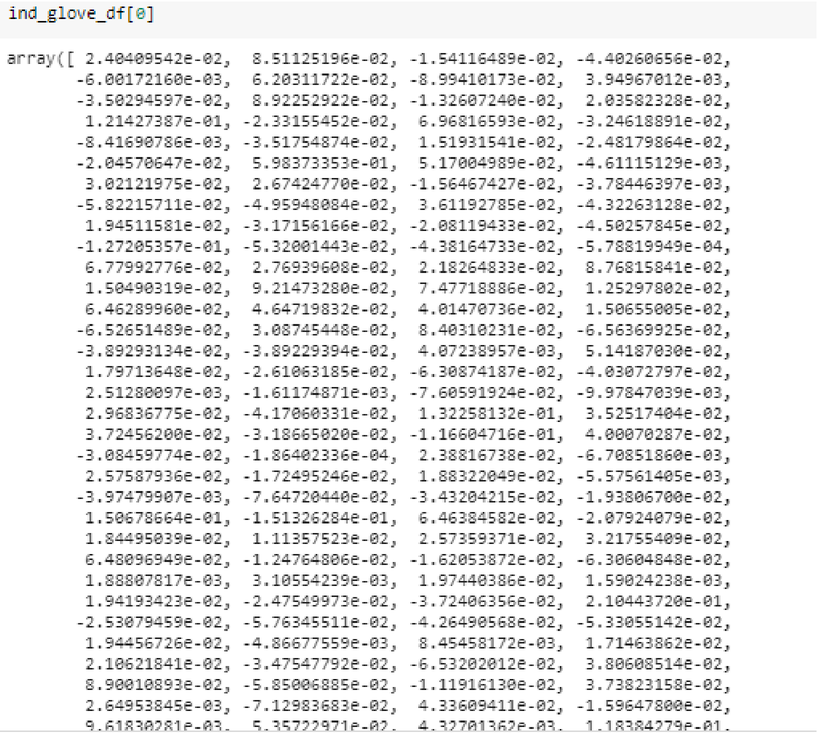
Since we had done sentiment analysis and n-gram analysis, we will use TF-IDF vectorisation and other embedding models like Glove and word2vec to analyse further on which words or vector of words help in predicting the right or closest potential accidental level. We take all the top n-gram words for our analysis. Why we need to use the word embedding models is that they help us really better in understanding the language semantics and in turn summarizing.

Word embeddings are a family of natural language processing techniques aiming at mapping semantic meaning into a geometric space. This is done by associating a numeric vector to every word in a dictionary, such that the distance (e.g., L2 distance or more commonly cosine distance) between any two vectors would capture part of the semantic relationship between the two associated words. The geometric space formed by these vectors is called an *embedding space*. Word embeddings are computed by applying dimensionality reduction techniques to datasets of co-occurrence statistics between words in a corpus of text. This can be done via neural networks (the “word2vec” technique), or via matrix factorization.

The two of the most common word embeddings are: **Word2Vec** and **GloVe,** and both of them are equally popular. But GloVe(“Global Vectors for Word Representation”) as the name suggests is better for preserving the **global contexts** as it creates a global co-occurrence matrix by estimating the probability a given word will co-occur with other words.

- Use TF IDF transformation model on the data frame
- We remove hyphenated words in critical risk factors
- We create the dummies for each of the columns and create one-hot encoding as shown below. The list of columns is large, so we will use word2vec at a later stage.

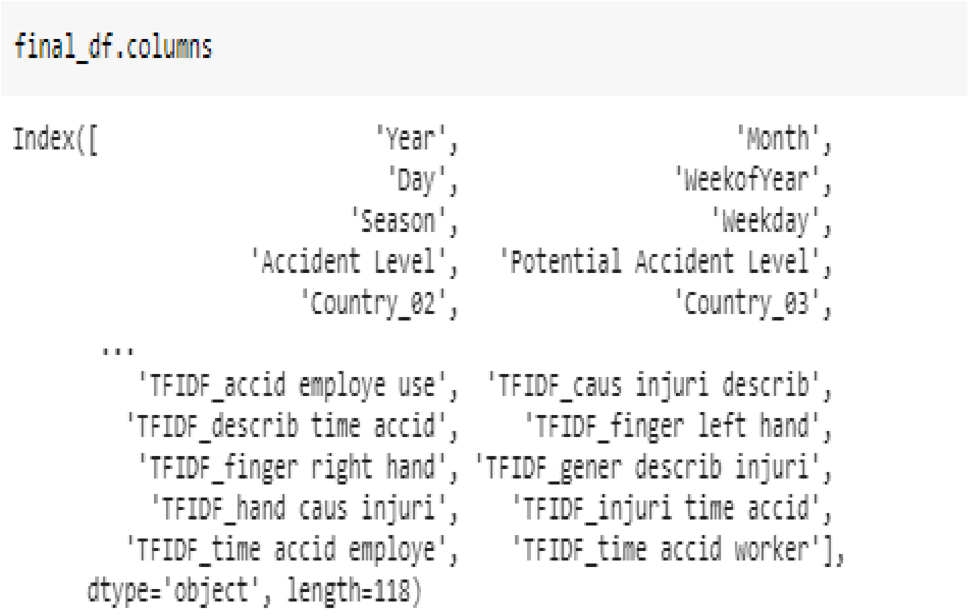
- We then use glove embeddings to get the number of words. We found 40K word vectors for the description vocabulary set.
- For each of these sentences, we create normalised vector.
- Normalised array for a glove embedded dataframe is shown as below:



```
ind_glove_df[0]
```

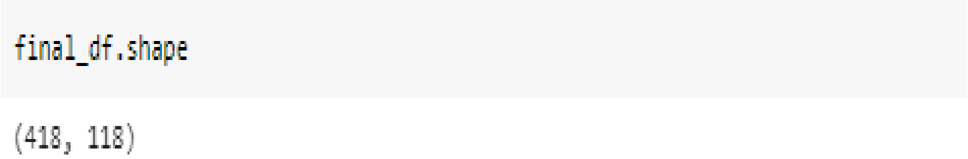
```
array([ 2.40409542e-02,  8.51125196e-02, -1.54116489e-02, -4.40260656e-02,
       -6.00172160e-03,  6.20311722e-02, -8.99410173e-02,  3.94967012e-03,
       -3.50294597e-02,  8.92252922e-02, -1.32607240e-02,  2.03582328e-02,
        1.21427387e-01, -2.33155452e-02,  6.96816593e-02, -3.24618891e-02,
       -8.41690786e-03, -3.51754874e-02,  1.51931541e-02, -2.48179864e-02,
       -2.04570647e-02,  5.98373353e-01,  5.17004989e-02, -4.61115129e-03,
        3.02121975e-02,  2.67424770e-02, -1.56467427e-02, -3.78446397e-03,
       -5.82215711e-02, -4.95948084e-02,  3.61192785e-02, -4.32263128e-02,
        1.94511581e-02, -3.17156166e-02, -2.08119433e-02, -4.50257845e-02,
       -1.27205357e-01, -5.32001443e-02, -4.38164733e-02, -5.78819949e-04,
        6.77992776e-02,  2.76939608e-02,  2.18264833e-02,  8.76815841e-02,
        1.50490319e-02,  9.21473280e-02,  7.47718886e-02,  1.25297802e-02,
        6.46289960e-02,  4.64719832e-02,  4.01470736e-02,  1.50655005e-02,
       -6.52651489e-02,  3.08745448e-02,  8.40310231e-02, -6.56369925e-02,
       -3.89293134e-02, -3.89229394e-02,  4.07238957e-03,  5.14187030e-02,
        1.79713648e-02, -2.61063185e-02, -6.30874187e-02, -4.03072797e-02,
        2.51280097e-03, -1.61174871e-03, -7.60591924e-02, -9.97847039e-03,
        2.96836775e-02, -4.17060331e-02,  1.32258132e-01,  3.52517404e-02,
        3.72456200e-02, -3.18665020e-02, -1.16604716e-01,  4.00070287e-02,
       -3.08459774e-02, -1.86402336e-04,  2.38816738e-02, -6.70851860e-03,
        2.57587936e-02, -1.72495246e-02,  1.88322049e-02, -5.57561405e-03,
       -3.97479907e-03, -7.64720440e-02, -3.43204215e-02, -1.93806700e-02,
        1.50678664e-01, -1.51326284e-01,  6.46384582e-02, -2.07924079e-02,
        1.84495039e-02,  1.11357523e-02,  2.57359371e-02,  3.21755409e-02,
        6.48096949e-02, -1.24764806e-02, -1.62053872e-02, -6.30604848e-02,
        1.88807817e-03,  3.10554239e-03,  1.97440386e-02,  1.59024238e-03,
        1.94193423e-02, -2.47549973e-02, -3.72406356e-02,  2.10443720e-01,
       -2.53079459e-02, -5.76345511e-02, -4.26490568e-02, -5.33055142e-02,
        1.94456726e-02, -4.86677559e-03,  8.45458172e-03,  1.71463862e-02,
        2.10621841e-02, -3.47547792e-02, -6.53202012e-02,  3.80608514e-02,
        8.90010893e-02, -5.85006885e-02, -1.11916130e-02,  3.73823158e-02,
        2.64953845e-03, -7.12983683e-02,  4.33609411e-02, -1.59647800e-02,
        9.61830281e-03,  5.35722971e-02,  4.32701362e-03,  1.18384279e-01,
```

- Before we head to final model building, this is our embedded model shape



```
final_df.columns
```

```
Index([                   'Year',                   'Month',
                           'Day',              'WeekofYear',
                        'Season',                 'Weekday',
                'Accident Level', 'Potential Accident Level',
                    'Country_02',              'Country_03',
       ...
          'TFIDF_accid employe use',  'TFIDF_caus injuri describ',
         'TFIDF_describ time accid',     'TFIDF_finger left hand',
          'TFIDF_finger right hand', 'TFIDF_gener describ injuri',
           'TFIDF_hand caus injuri',    'TFIDF_injuri time accid',
         'TFIDF_time accid employe',    'TFIDF_time accid worker'],
      dtype='object', length=118)
```



```
final_df.shape
```

```
(418, 118)
```

2.7 Conclusion of Text preprocessing or analysis on description field:

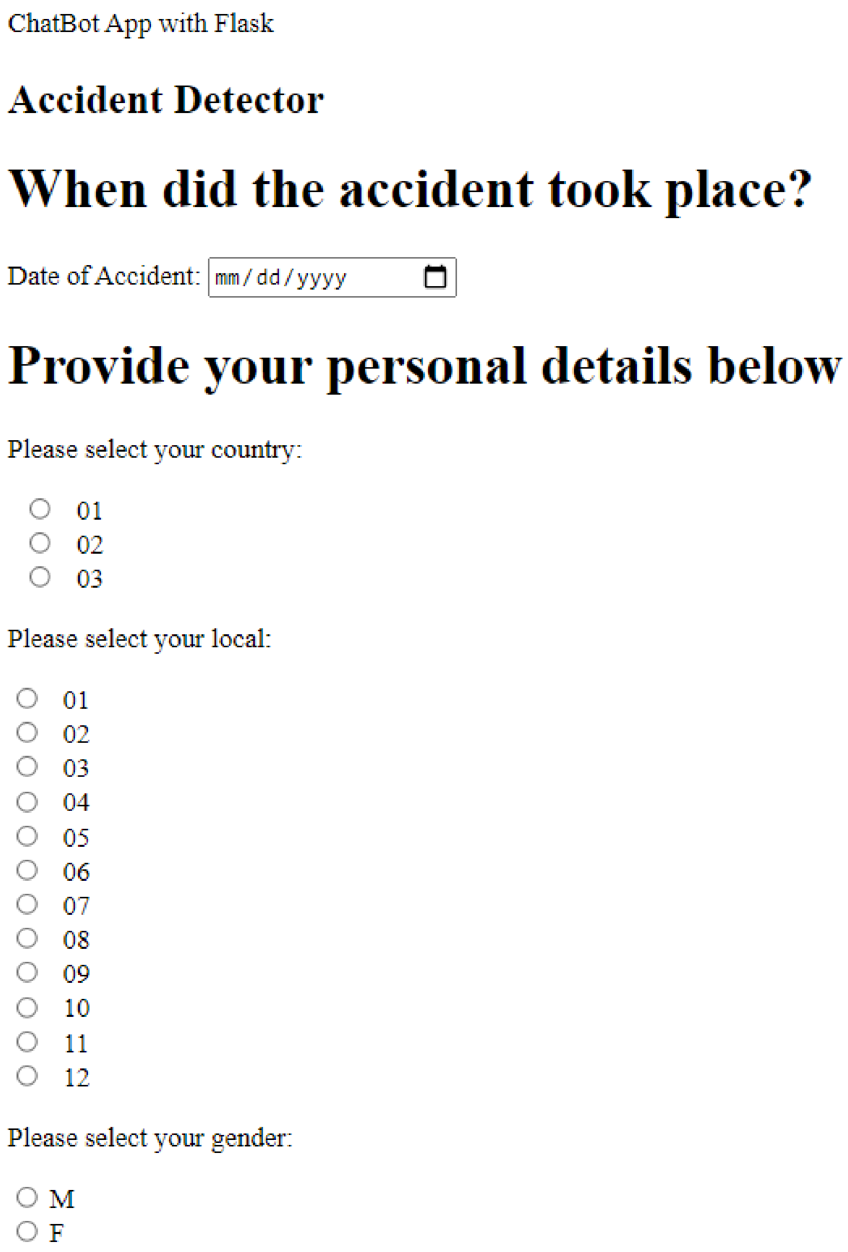
- We had done categorical column analysis on EDA section, whereas in Text processing, we took the description column and tried to do NLP model analysis.
- We used the n-gram analysis for finding the multi words that go together for our analysis
- We also did lot of feature engineering and we used glove embedding and TF IDF and came out with the finalised list of one hot encoded parameter which will be used for the model building which will be done in next section.
- **How this will be used for Chatbot utility:** Our chatbot utility will be required to take some parameters as input like Country, Locals, Gender, Industry, Employee type and critical risk factor etc. We also would want them to describe their injuries. In this field, text processing will have to be done. Hence, this is how it will be used in final GUI or NLP Chatbot utility program.

2.8 GUI Component

We used Flask component to deploy PyTorch models in production. Using Flask in this way is by far the easiest way to start serving your PyTorch models.

We take the input in the form of .html file where user inputs the relevant fields.

As earlier mentioned, we used Potential Accident level as target for our ML Random Classifier model whereas the accident level is the target for the NLP based LSTM classifier using the text description.



ChatBot App with Flask

Accident Detector

When did the accident took place?

Date of Accident: mm/dd/yyyy

Provide your personal details below

Please select your country:

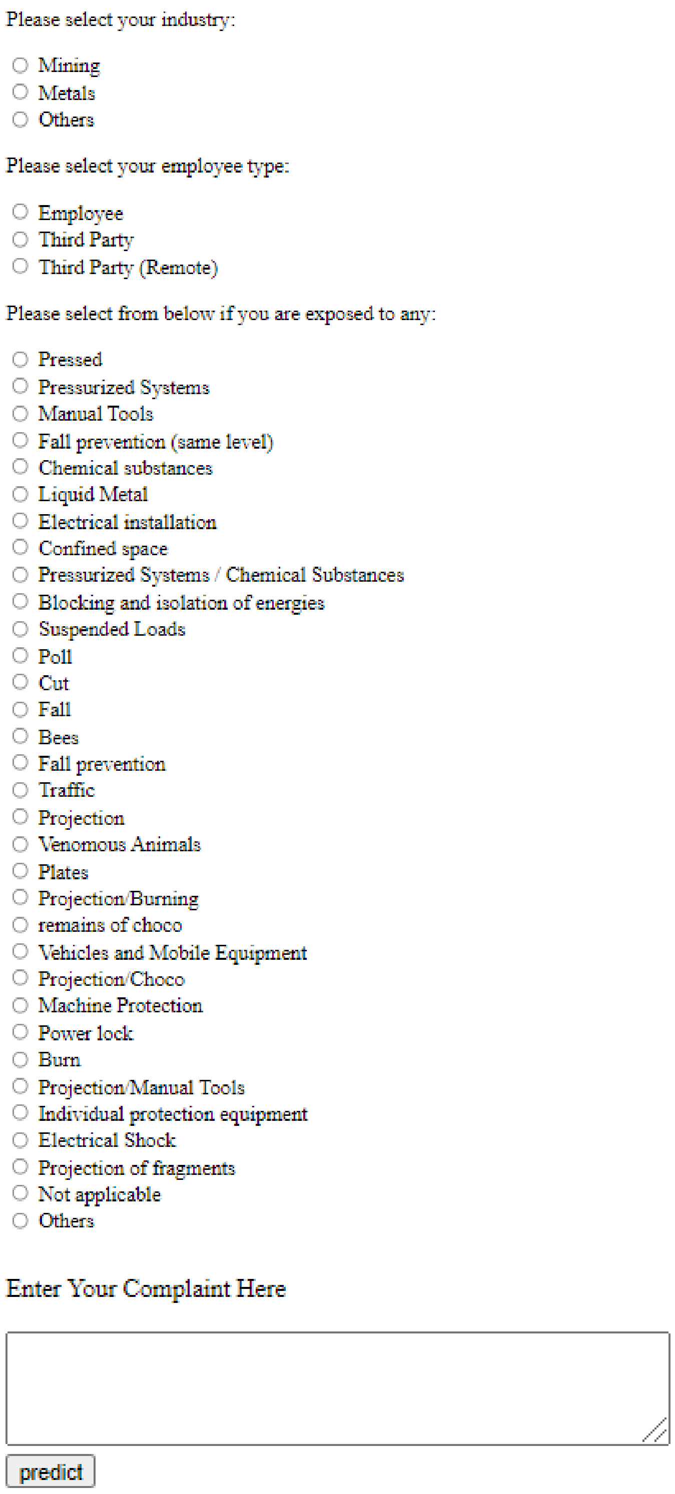
○ 01
○ 02
○ 03

Please select your local:

○ 01
○ 02
○ 03
○ 04
○ 05
○ 06
○ 07
○ 08
○ 09
○ 10
○ 11
○ 12

Please select your gender:

○ M
○ F



Please select your industry:

○ Mining
○ Metals
○ Others

Please select your employee type:

○ Employee
○ Third Party
○ Third Party (Remote)

Please select from below if you are exposed to any:

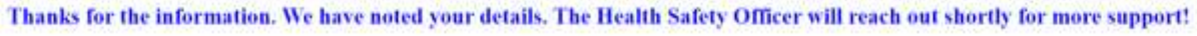
○ Pressed
○ Pressurized Systems
○ Manual Tools
○ Fall prevention (same level)
○ Chemical substances
○ Liquid Metal
○ Electrical installation
○ Confined space
○ Pressurized Systems / Chemical Substances
○ Blocking and isolation of energies
○ Suspended Loads
○ Poll
○ Cut
○ Fall
○ Bees
○ Fall prevention
○ Traffic
○ Projection
○ Venomous Animals
○ Plates
○ Projection/Burning
○ remains of choco
○ Vehicles and Mobile Equipment
○ Projection/Choco
○ Machine Protection
○ Power lock
○ Burn
○ Projection/Manual Tools
○ Individual protection equipment
○ Electrical Shock
○ Projection of fragments
○ Not applicable
○ Others

Enter Your Complaint Here

After the user had given the inputs, we take this predict function on a Post method and the resultant output is captured in results HTML with multiple condition.

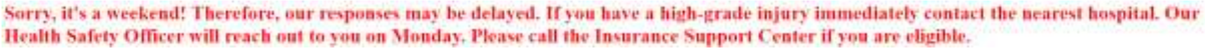
DATE:

Here basis the date field entered, We check if its weekday and weekend. The message for weekday is as follows:



Thanks for the information. We have noted your details. The Health Safety Officer will reach out shortly for more support!

If its weekend, we send out this information alerting the employee to connect to nearest hospital.



Sorry, it's a weekend! Therefore, our responses may be delayed. If you have a high-grade injury immediately contact the nearest hospital. Our Health Safety Officer will reach out to you on Monday. Please call the Insurance Support Center if you are eligible.

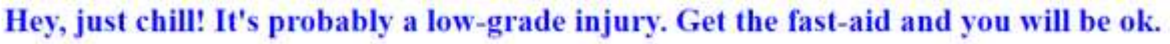
ACCIDENT LEVEL PREDICTION:

Next as per the inputs received on description of the accident column, we run through the biLSTM saved model with weights and biases. This will predict accident level.

The output of this are either high grade and low-grade severity level. Then we give out message as in following based on the severity levels.

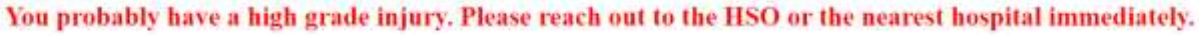
If its high grade one, we give out the information in such a way the employee can take appropriate action.

For lower grade ones, the following is the output.



Hey, just chill! It's probably a low-grade injury. Get the fast-aid and you will be ok.

For higher – grade ones, the following is the output.

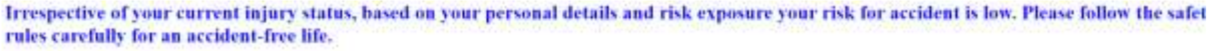


You probably have a high grade injury. Please reach out to the HSO or the nearest hospital immediately.

POTENTIAL ACCIDENT LEVEL PREDICTION:

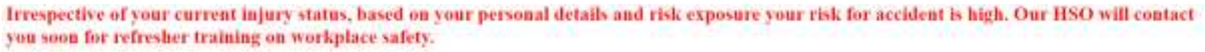
We send the inputs of the categorical columns to a saved ML (Random forest classifier). Here we predict the potential accident level. The accident level predicted above is from description given. However, potential accident level can be influenced by other factors. Esp. industry, country or location prone to accidents and critical risk factors. These can probably define what could have been the level with other inputs. So, we give slightly more pronounced output message.

If potential accident level is low grade, then we send this message out.



Irrespective of your current injury status, based on your personal details and risk exposure your risk for accident is low. Please follow the safet rules carefully for an accident-free life.

If its high-grade ones , then this is the message.



Irrespective of your current injury status, based on your personal details and risk exposure your risk for accident is high. Our HSO will contact you soon for refresher training on workplace safety.

The colour coding is also added with **red** and **blue** to give sense of urgency to the employees. Since the CTA is more rule based, this can be altered easily depending on the situation. Code will be easy to manage and maintain. Performance also is better and faster using this mechanism.

3. Detailed Walk-through of Solution

a. Import Libraries
b. Initial analysis of overall data set
c. EDA : Detailed analysis discussed in above section 2
d. Text processing analysis -detailed analysis discussed in above section 2
e. Model building –
    i. LSTM Models
        1. We had used LSTM model for sending text description
        2. After curating the text by removing duplicate, stop words and indexed or tokenised the words
        3. We had taken maximum length of description field to be 100; the maximum words that our dictionary will be containing is 5000 from the description fields after removing stop words. We also felt the maximum parameter that can be sent to model is 50. This is derived from n-gram analysis that at least 15 tri-gram words help in some sort of prediction or analysis. Our padding is kept at 50.
        4. LSTM model – we had used SoftMax function with binary classification of low grade and high-grade accident levels. We had not used for multiple classification as we felt that the sampling for each of the levels of accident level is very less. Accuracy also will fall. We also felt that there would be lot of false positives coming out of that model. Hence, low grade are accident levels 1 and 2 whereas high grade accident levels are 3, 4 and 5.
        5. Our accuracy is more than 70%.
        6. We could predict and give comfort to the end user telling them what grade of accident level you are in.
    ii. ANN/CNN Models
        1. We had applied the text description field and passed through the ANN and CNN models for predicting the accident levels.
        2. We built a fully connected layer and predicted the results.
        3. The accuracy is similar to LSTM for CNN model whereas other model had lesser accuracy.
        4. We will continue to use LSTM for NLP text description analysis
    iii. ML Supervised learning
        1. Since most other fields are categorical apart from description, we had used ML supervised learning models and predicted the potential accident level. What could have been or might have been your level of accident taking into account your industry, gender, employee type , country and plant location you operate in.
        2. We had used lazy classifiers but in our final model, we will go with random forest classifier. The other models like SVC is also good but random forest classifier accuracy is better than the rest
        3. We will use this for predicting the potential accident levels
f. GUI and model deployment
    i. The input HTML captures the required input.
    ii. The first step would be to get the date and description of the injury.

iii. Using this, we predict the accident level, either a low grade one (severity level 1 and 2) or a higher grade one (Severity level 3,4 and 5) using LSTM model.
iv. Post that, we will capture more details about country, location of plant, gender, employee type, day of the incident, industry and critical risk factors governing the accident. These parameters will go into our supervised model (random forest) to calculate the potential accident level.
v. Based on this potential accident level, we came out with call to action based on rules defined.
vi. We also captured the date to indicate what to do if it's a weekend or a weekday.

4. Model Evaluation

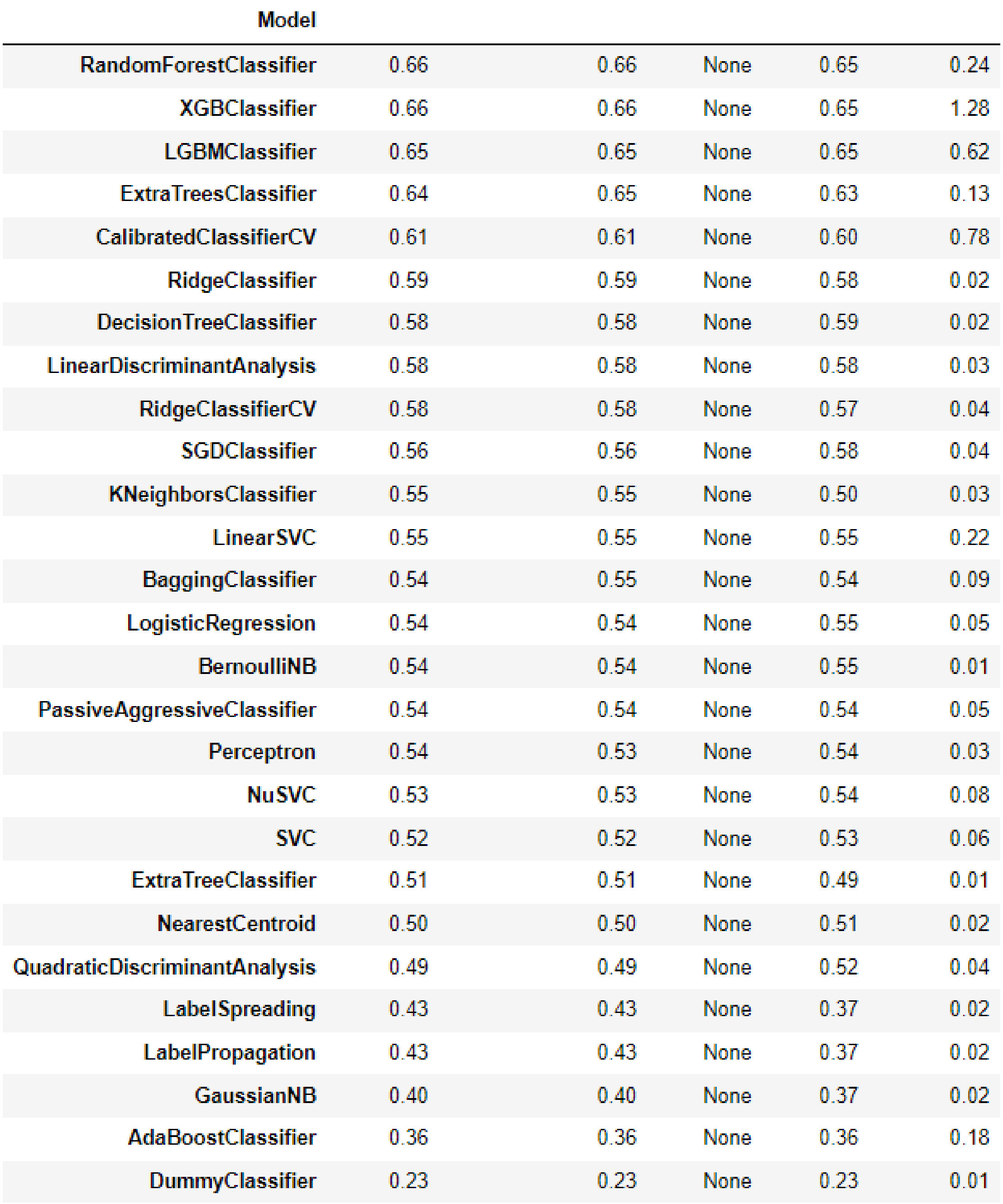
We were initially thinking to build the model, sending all text + categorical columns input to the models. However, our observation based on the model building / execution is the accuracy was far lower as there are far too many features as part of X array. More features got added from the text description field. Even if we restricted the overall text description to say under 50, it would not have yielded the right prediction of the accident level or potential accident level.

Another factor was the model was taking more time to run. Processing time is important esp. in Chatbot utility since the end user would be waiting for the responses thrown from the model.

We also got the feedback from mentor on why not split the text model and other model. Through this, we can predict accident level using text-based NLP or Deep learning techniques and we can predict potential accident levels using the other categorical columns captured in the data set.

The possible target columns are accident level and potential accident levels for each of the models.

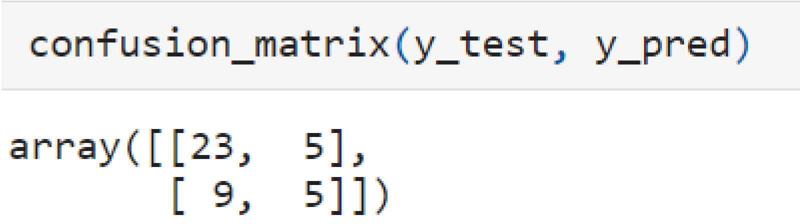
The ML model classifier – we used lazy classifier and found out that Random forest classifier has better accuracy than other.



Model					
RandomForestClassifier	0.66	0.66	None	0.65	0.24
XGBClassifier	0.66	0.66	None	0.65	1.28
LGBMClassifier	0.65	0.65	None	0.65	0.62
ExtraTreesClassifier	0.64	0.65	None	0.63	0.13
CalibratedClassifierCV	0.61	0.61	None	0.60	0.78
RidgeClassifier	0.59	0.59	None	0.58	0.02
DecisionTreeClassifier	0.58	0.58	None	0.59	0.02
LinearDiscriminantAnalysis	0.58	0.58	None	0.58	0.03
RidgeClassifierCV	0.58	0.58	None	0.57	0.04
SGDClassifier	0.56	0.56	None	0.58	0.04
KNeighborsClassifier	0.55	0.55	None	0.50	0.03
LinearSVC	0.55	0.55	None	0.55	0.22
BaggingClassifier	0.54	0.55	None	0.54	0.09
LogisticRegression	0.54	0.54	None	0.55	0.05
BernoulliNB	0.54	0.54	None	0.55	0.01
PassiveAggressiveClassifier	0.54	0.54	None	0.54	0.05
Perceptron	0.54	0.53	None	0.54	0.03
NuSVC	0.53	0.53	None	0.54	0.08
SVC	0.52	0.52	None	0.53	0.06
ExtraTreeClassifier	0.51	0.51	None	0.49	0.01
NearestCentroid	0.50	0.50	None	0.51	0.02
QuadraticDiscriminantAnalysis	0.49	0.49	None	0.52	0.04
LabelSpreading	0.43	0.43	None	0.37	0.02
LabelPropagation	0.43	0.43	None	0.37	0.02
GaussianNB	0.40	0.40	None	0.37	0.02
AdaBoostClassifier	0.36	0.36	None	0.36	0.18
DummyClassifier	0.23	0.23	None	0.23	0.01

We incorporated the ANN/CNN and LSTM models for the text description fields. We ran this model to predict the first target column called ACCIDENT LEVEL.

Then we collected the inputs on other parameters like Gender, Employee type, Industry, country and location of plant, Critical risk factors to derive the POTENTIAL ACCIDENT LEVEL. This tells what could have been or might have been the accident levels basis the other parameters that influence this target variable. We had used ML Random forest classifier for the same. This model had the best accuracy with 66%. Confusion Matrix was as below:



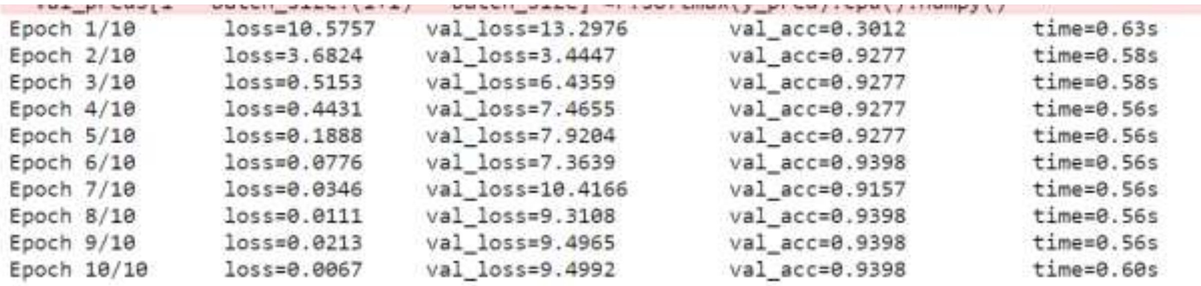
```
confusion_matrix(y_test, y_pred)
```

```
array([[23,  5],
       [ 9,  5]])
```

We applied the gridsearch CV for hyper parameter tuning and got the best performing model at 84% accuracy.

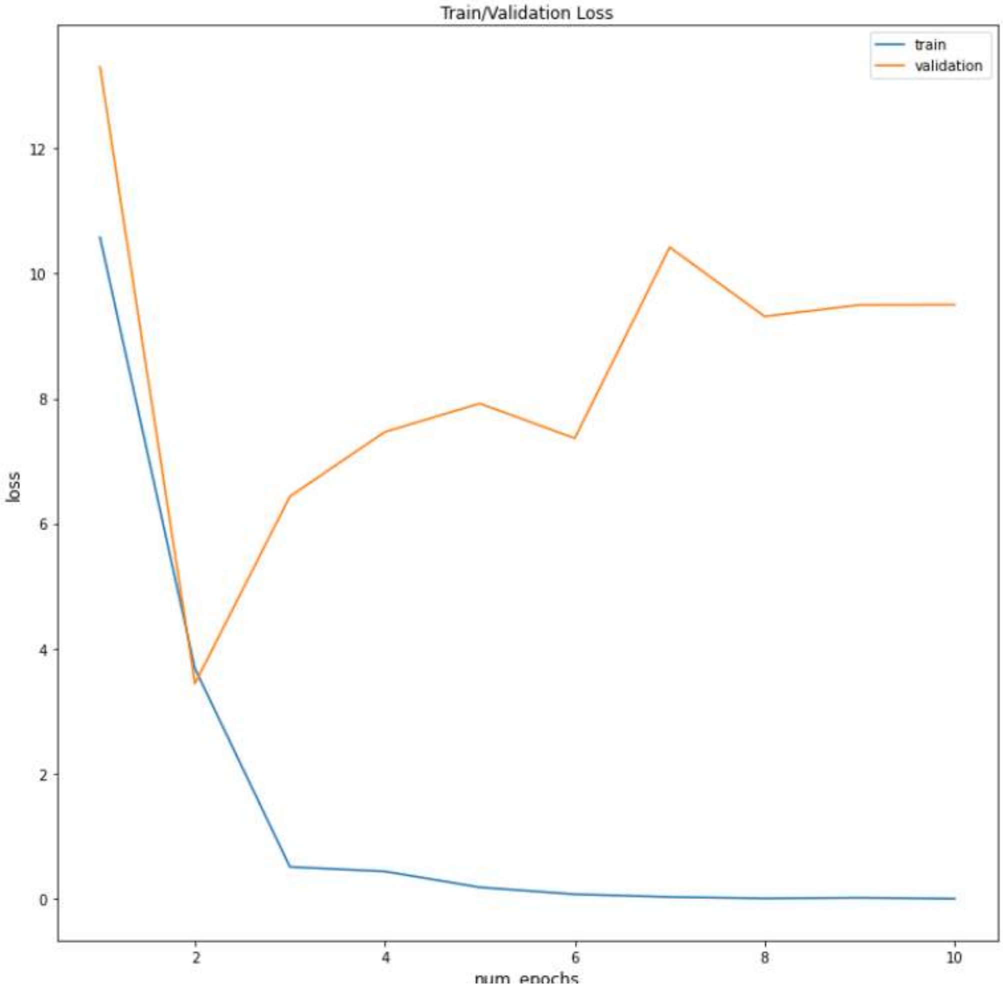
For Deep learning model, we used biLSTM model for text description field to predict accident level using ADAM optimiser and cross entropy loss for sum with learning rate of 0.001

For 10 epochs, the below is the outcome.

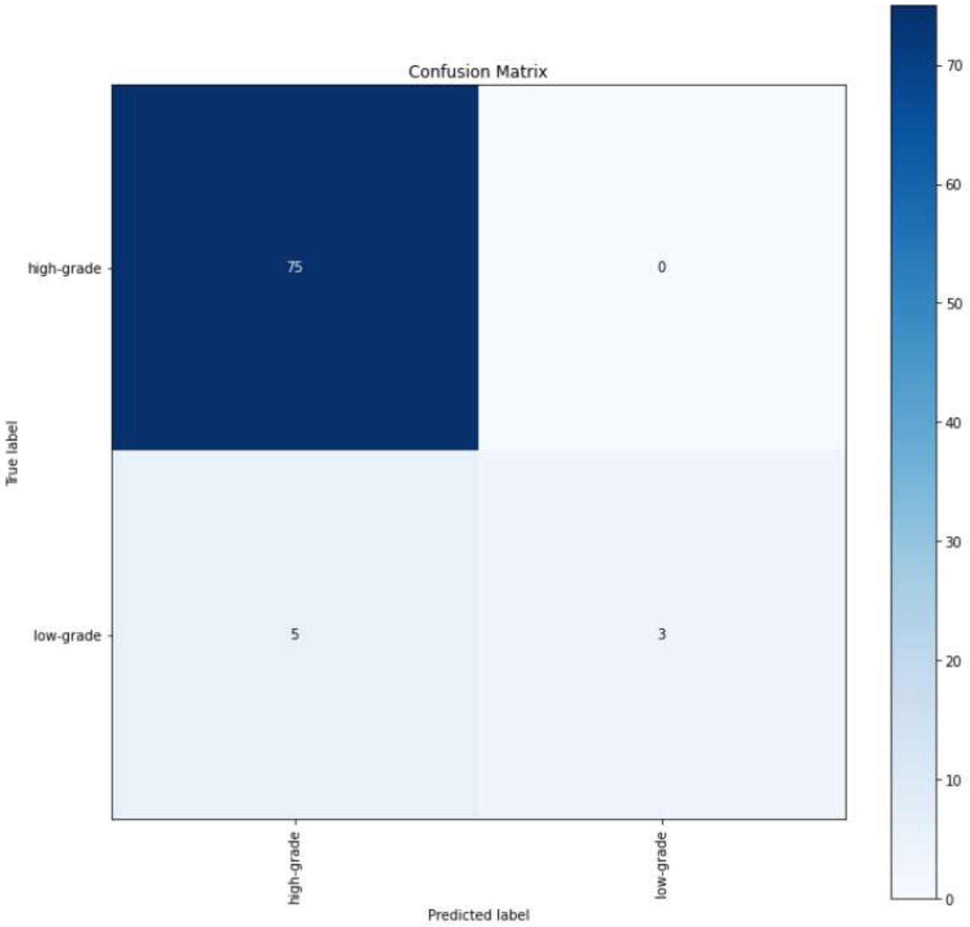


```
Epoch 1/10      loss=10.5757   val_loss=13.2976        val_acc=0.3012          time=0.63s
Epoch 2/10      loss=3.6824    val_loss=3.4447         val_acc=0.9277          time=0.58s
Epoch 3/10      loss=0.5153    val_loss=6.4359         val_acc=0.9277          time=0.58s
Epoch 4/10      loss=0.4431    val_loss=7.4655         val_acc=0.9277          time=0.56s
Epoch 5/10      loss=0.1888    val_loss=7.9204         val_acc=0.9277          time=0.56s
Epoch 6/10      loss=0.0776    val_loss=7.3639         val_acc=0.9398          time=0.56s
Epoch 7/10      loss=0.0346    val_loss=10.4166        val_acc=0.9157          time=0.56s
Epoch 8/10      loss=0.0111    val_loss=9.3108         val_acc=0.9398          time=0.56s
Epoch 9/10      loss=0.0213    val_loss=9.4965         val_acc=0.9398          time=0.56s
Epoch 10/10     loss=0.0067    val_loss=9.4992         val_acc=0.9398          time=0.60s
```

As you can see the accuracy in the second epoch only went to 92% and the final accuracy is 94%. The most important parameter is it took only 2 mins to run 2 epochs. The validation loss after sometime do not yield much results, so we can run for max 2-3 epochs for better results and faster performance.



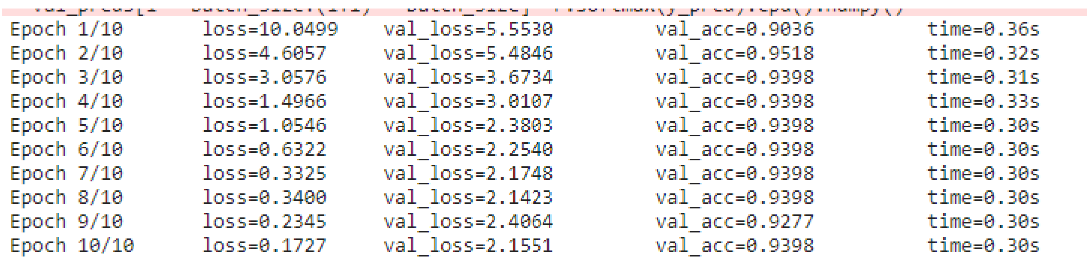
The confusion matrix is as below:



As you can see, the false positive cases are so low. The model almost able to predict the right severity level of accident level right.

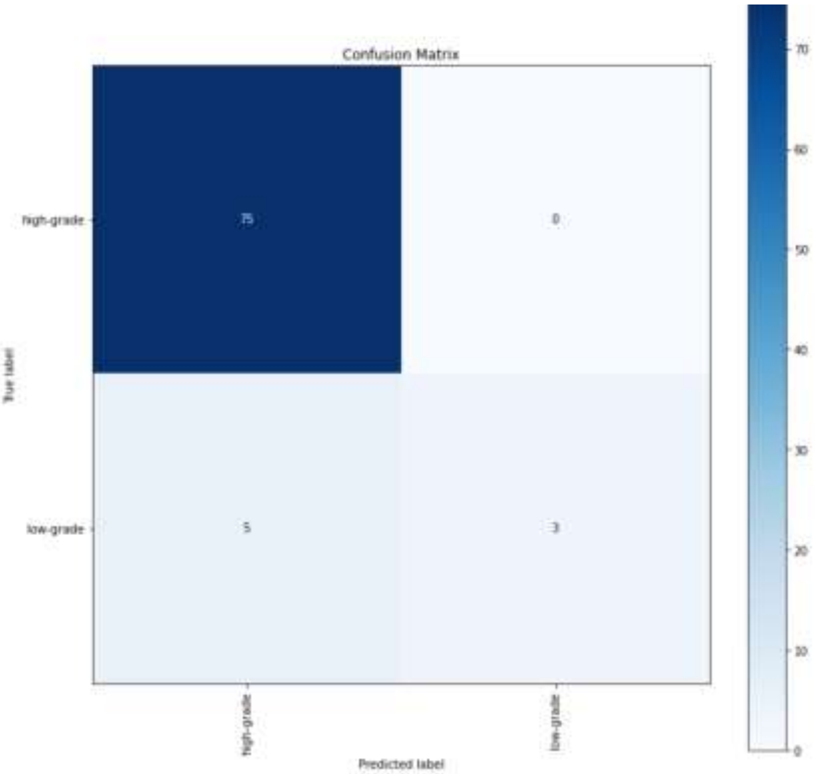
We tried biLSTM model with reduced learning rate of 0.01, there is a slight reduction in accuracy but the time taken is almost the same for all epochs.

When we tried the same with CNN model, we got a better accuracy in epoch 1 only.



```
Epoch 1/10     loss=10.0499   val_loss=5.5530     val_acc=0.9036     time=0.36s
Epoch 2/10     loss=4.6057    val_loss=5.4846     val_acc=0.9518     time=0.32s
Epoch 3/10     loss=3.0576    val_loss=3.6734     val_acc=0.9398     time=0.31s
Epoch 4/10     loss=1.4966    val_loss=3.0107     val_acc=0.9398     time=0.33s
Epoch 5/10     loss=1.0546    val_loss=2.3803     val_acc=0.9398     time=0.30s
Epoch 6/10     loss=0.6322    val_loss=2.2540     val_acc=0.9398     time=0.30s
Epoch 7/10     loss=0.3325    val_loss=2.1748     val_acc=0.9398     time=0.30s
Epoch 8/10     loss=0.3400    val_loss=2.1423     val_acc=0.9398     time=0.30s
Epoch 9/10     loss=0.2345    val_loss=2.4064     val_acc=0.9277     time=0.30s
Epoch 10/10    loss=0.1727    val_loss=2.1551     val_acc=0.9398     time=0.30s
```

Confusion matrix is also the same.



As earlier pointed, we avoided the multiple classification false positive error by aggregating severity level 1 and 2 into low grade and other three severity into high grade. This way, the classification accuracy improved a lot.

In Conclusion:

We have finalised the **ML model Random forest classifier** for predicting the potential accident level using the categorical columns. We have finalised the **LSTM model for NLP text** analysis of description to predict the accident levels.

5. Comparison to Benchmark

Our data set had only 418 records. There was a class imbalance for individual severity levels for both potential accident level and accident level data. If we had used this as it is, it would have resulted in lot of false positives for each of the severity levels. So, to avoid that problem, we had used binary classification based on text LSTM model to predict if an accident level is a low-grade (severity level 1 and 2) or high-grade (severity level 3, 4 and 5). This resulted in good accuracy levels. This also processed the model faster.

For ML model too, since there are not many columns (since we reduced the text descriptors vector in x-array), the prediction of potential accident level was much better at 84% for random forest classifier.

This accuracy level is comparable to benchmark one. Had there been more data set, more diversified sampling of the various severity levels, then the possibility of better accuracy can be achieved.

6. Visualisations

We had included few visualisations added as part of EDA and Text processing in section 2 and also section 4 on Model evaluation.

7. Implications

Observations:

1. We have seven duplicate values in this dataset and dropped those duplicate values.
2. We have no outliers in this dataset.
3. We have no missing values in this dataset.
4. Extracted the day, month and year from Date column and created new features such as weekday, weekofyear and seasons.
5. Target variable – 'Accident Level' distribution is not equal (I: 309, II: 40, III: 31, IV: 30, V: 8).
6. Class imbalance issue is handled using below methods and found out that, for this particular dataset, with original data we have achieved the better results.
    1. Resampling techniques: Oversampling minority class
    2. SMOTE: Generate synthetic samples
7. **RandomForestClassifier model with an accuracy of 66% is our best model.**
8. **Finally bidirectional LSTM model can be considered to productionalized the model to predict the accident level.**

Our chatbot utility can be used by end business user who are out there in the field to get a first sense of whether their injury is low or high grade. What they should do immediately to get the next course of action based on the grade of injury.

As we have seen our accident level prediction is at 94% and potential accident level accuracy prediction is also at 66%, we can tell with confidence that 70% of the description provided by the end user, we can predict 70% accurately from the data dictionary built by us.

We can also give them call to action which is now sort of rule based. We can turn this into a recommendation engine based on industry, country and plant locations etc. and give more closer and better CTA to end user.

As we stated in the start of the document, that organizations safety department look to use chatbot as part of their digital transformation in using data to provide first hand good insight of such accident levels so that the employees' safety can be quickly handled. Data and insights provided will help big time in taking quicker actions. Our solution can be comprehensively used by such department in ensuring the first level severity level and possible call to action which can help them further.

This also have the ability to add more use cases around safety whenever data is gathered. We can also use this first level chatbot info to add to our data set and keep improving the models. We can potentially add more conversational use cases and make it more interactive one than the one we designed. Given the time, this is a workable solution that can be used by domain users effectively.

Recommendations:

1. In this project, we analysed based on the data set that the main causes of accidents are mistakes in hand-operation and time-related factor.
2. To reduce the occurrences of accidents, more stringent safety standards in hand-operation will be needed in period when many accidents occur.
3. NLP analysis on 'Description' helped in identifying the accident levels effectively.
4. Few more information such as machining data (ex. CNC, Current, Voltage) in plants, weather information, employee's personal data (ex. age, experience in the industry sector, work performance), can help in possibly estimating the cause of accidents accurately.
5. With a greater number of observations than current data set, we can evaluate the performance of those models and get the better results.
6. There are quite a lot of Critical risk descriptions, but with the help of SME we can decide whether this column has outliers or not and also SME can help us in understanding the data better.

8. Limitations

The limitations currently as we see it are:

- We have a smaller number of observations to analyse the cause of accidents correctly and rather we should collect a greater number of observations to get better results.
- Less number of features available in dataset.
- Lack of access to quality data. Model require more data sampling for all the severity levels. That could have helped in identifying exactly the severity level than currently telling them low or high-grade ones. A greater data observation can help. Over sampling methods like SMOTE etc. cannot help in solving this completely.
- We could have made more conversational chatbot. Getting the parameters through conversation would have enhanced the solution.
- Given the time, we could not work on better UI experience.
- We also can add continuous learning back to keep improving the model. But again, after certain accuracy level, the model will not learn but only apes the data set.

Where does our model fall short in the real world?

- Once we deploy the finalised model in Production, we feel we might get less f1-score as compared to productionalized model results.
- Since we are predicting the accident level, we need to be 100% sure or at least close to 100% so that we can prevent the lot of accidents in industry. What is the “allowable limit” have to be worked with industry expert in the organisation.

What can you do to enhance the solution? -- Need to work on limitations.

9. Closing Reflections

What did we learn from the process?

- How to work on Data Science project to end-to-end.
- How to handle class imbalance data set.
- How to build different ANN and CNN model architectures for handling multi-class classification problems.
- How to build different NLP architectures for handling text data.
- It is an important industrial use case problem that requires solution. Many organizations can have data, but to derive insights using the NLP techniques and also the other ML models will go a long way in solving such problems.
- The learning also provided us how to deploy and how to utilise them in real world with real use cases.

What will we do differently next time

- We spent lot of time in addressing the target variable. Once that clarity came in, things became clearer. We will try to understand the core business problem right and then use learning to apply it better as per the problem.
- We will explore more feature engineering and feature selection techniques
- We will build the real chatbot using Streamlit or some other applications

THANK
YOU!