

Capstone Project Apr'21

Batch:AIML.O.May'20A

NLP-2 Semi Ruled ChatBot

Mentor: Srihari Nalabolu

Group6B: Amaresh-Aprajita-Sampan-Shivang

Contents

0.1Introduction	2
0.1.1 Overview	2
0.1.2 Problem Statement	3
0.2Data Analysis	4
0.2.1 Data Pre-processing:	4
0.2.2 EDA	5
0.3Data Modelling	19
0.3.1 ML Models and Neural Network:	20
0.3.2 Bidirectional LSTM Model:	22
0.4Building Chatbot	23













0.1 Introduction

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

0.1.1 Overview

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

Rule Based vs AI Bots

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

0.1.2 Problem Statement

DOMAIN:

Industrial safety. NLP based Chat-bot.

CONTEXT:

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

DATA DESCRIPTION:

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

COLUMNS DESCRIPTION:

Data: timestamp or time/date information

Countries: which country the accident occurred (anonymous)

Local: the city where the manufacturing plant is located (anonymous)

Industry sector: which sector the plant belongs to

Accident level: from I to VI, it registers how severe was the accident (I means 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.

Link to download the dataset:

https://drive.google.com/file/d/1_GmrRP1S20Ia02K1f0BNkYa8uxazGbfE/view?usp=sharing

PROJECT OBJECTIVE:

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

0.2 Data Analysis

0.2.1 Data Pre-processing:

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

Glimpse of raw data:

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

Insights gained on data and data pre-processing:

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

[2] It is worth noting that components of this dataset were anonymized to hide the location and name of each facility.

[3] No missing values in the dataset. Also, removing unnecessary column named "Unnamed" as we do not know any related metadata and adds no value to the analysis.

[4] Five Point Summary analysis:

Country_01 is the country where most of the accidents happen (more than 50%).

Local_03 (which also belongs to Country_01) is where most of the accidents happen.

Mining is also the most significant contributor to accidents.

Male (95%) and Third Party (43%) also counts for kind of people that suffers more accident.

[5] Countries where the dataset was collected is anonymized but they are all located in South America. So in this analysis, let's assume the dataset was collected in Brazil. Brazil has

four climatological seasons as below.

Spring : September to November

Summer : December to February

Autumn : March to May

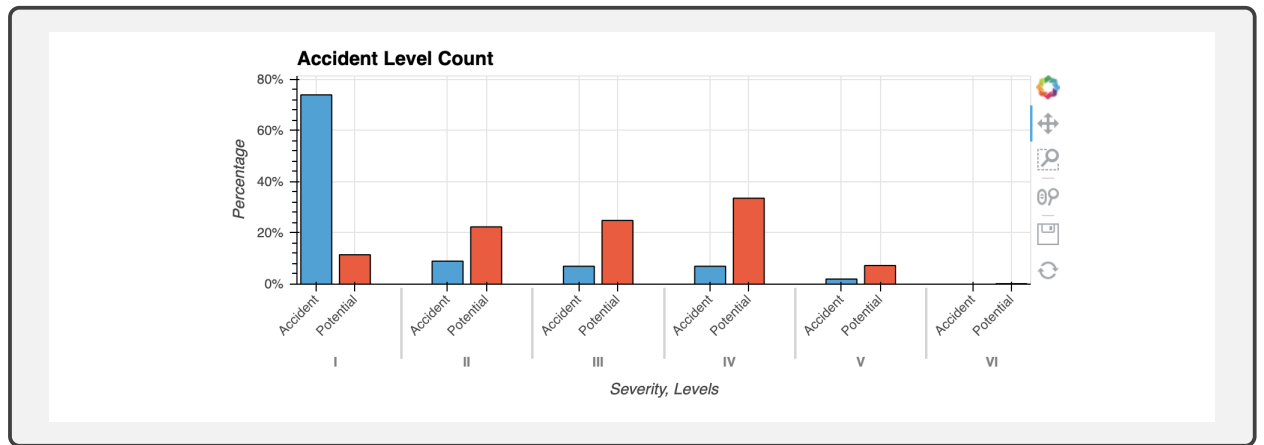
Winter : June to August

We created seasonal variable based on month variable.

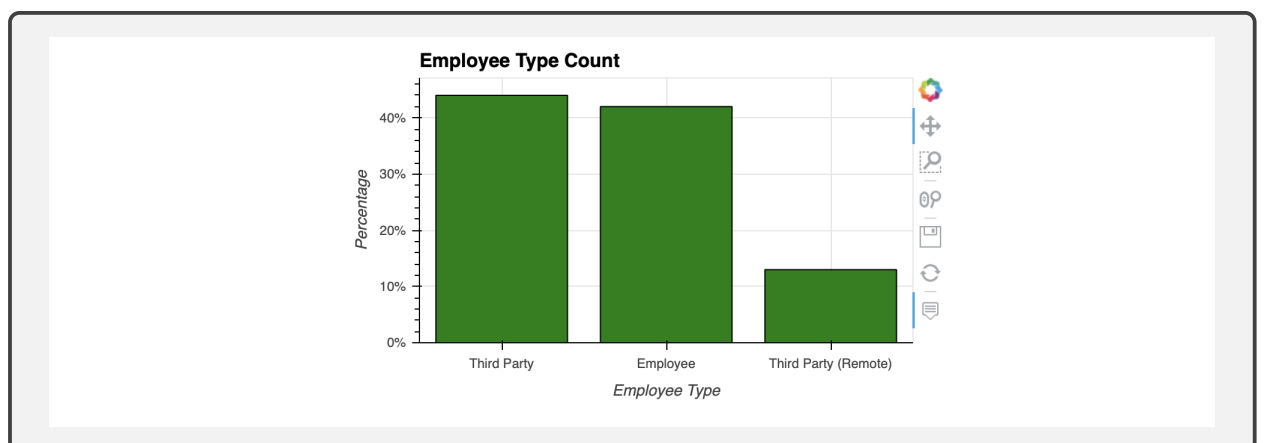
0.2.2 EDA

Insights gained from exploratory data analysis:

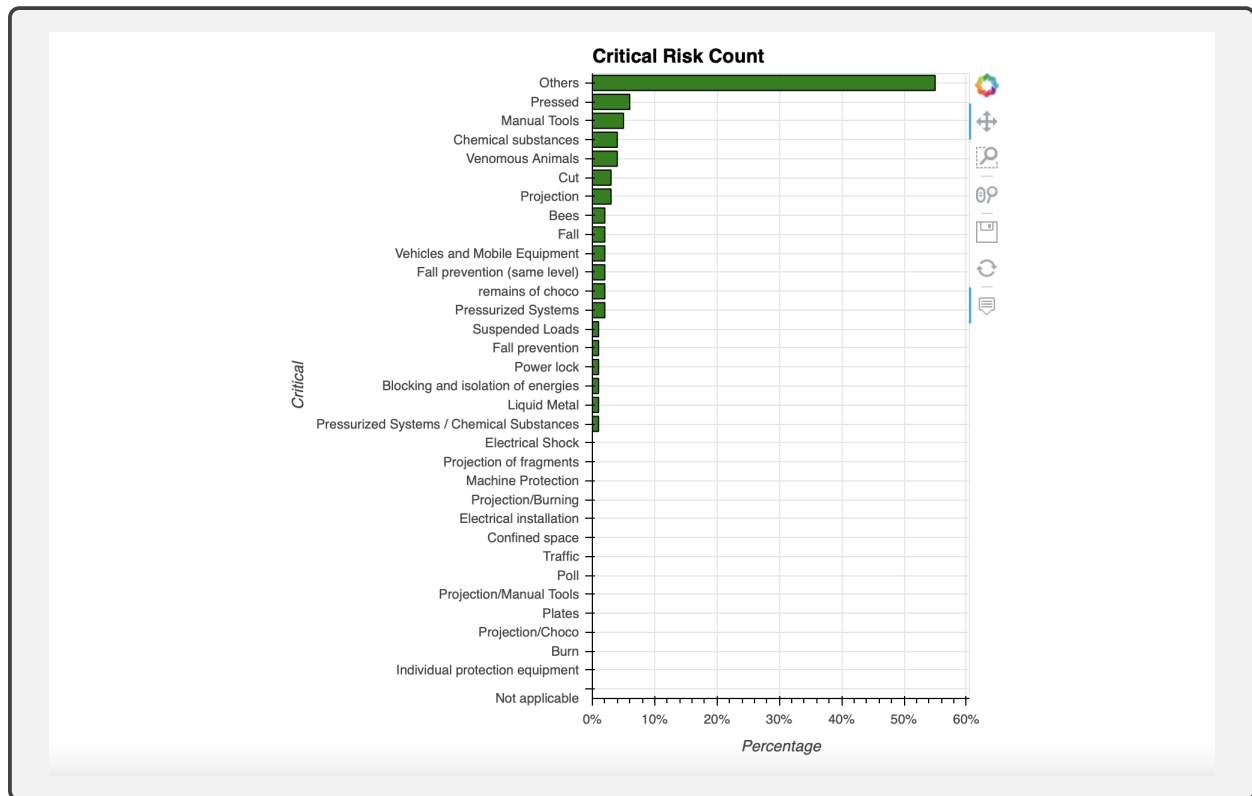
[1]The number of accidents decreases as the Accident Level increases and vice versa with Potential Accident Level.



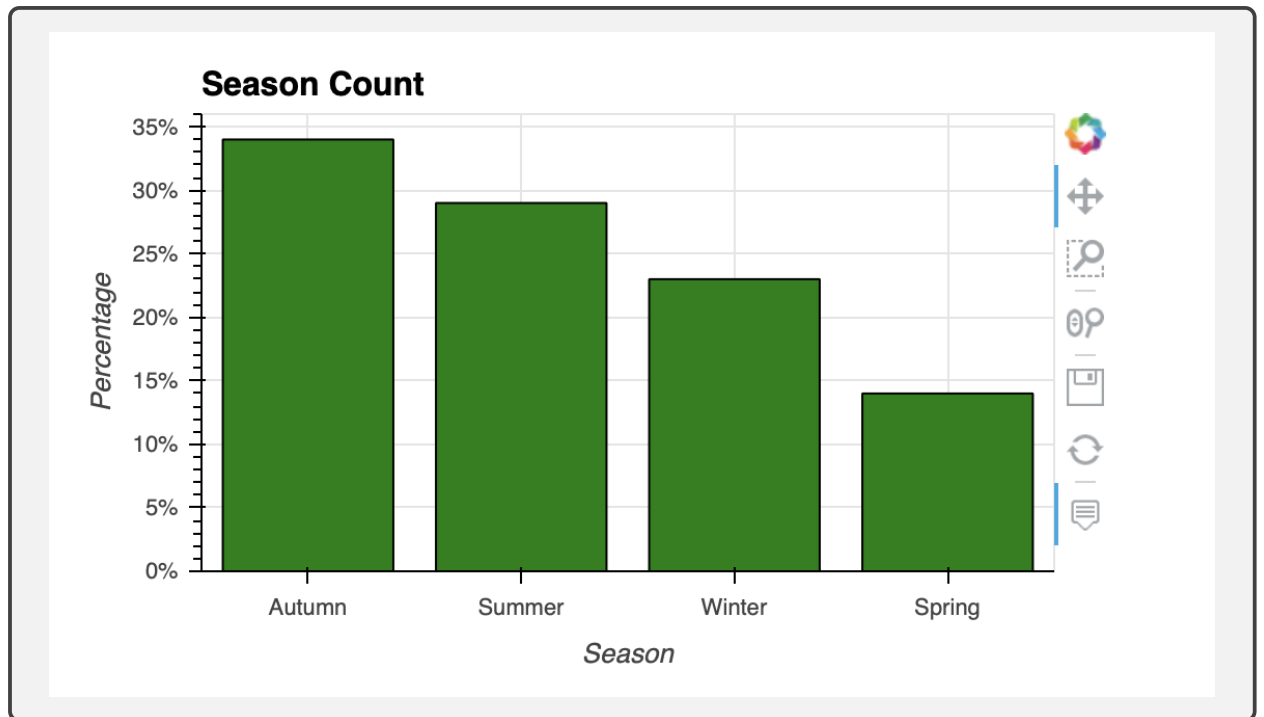
[2]The large number of Third Party employee type indicates the difference of employment system in gender or industry sector.



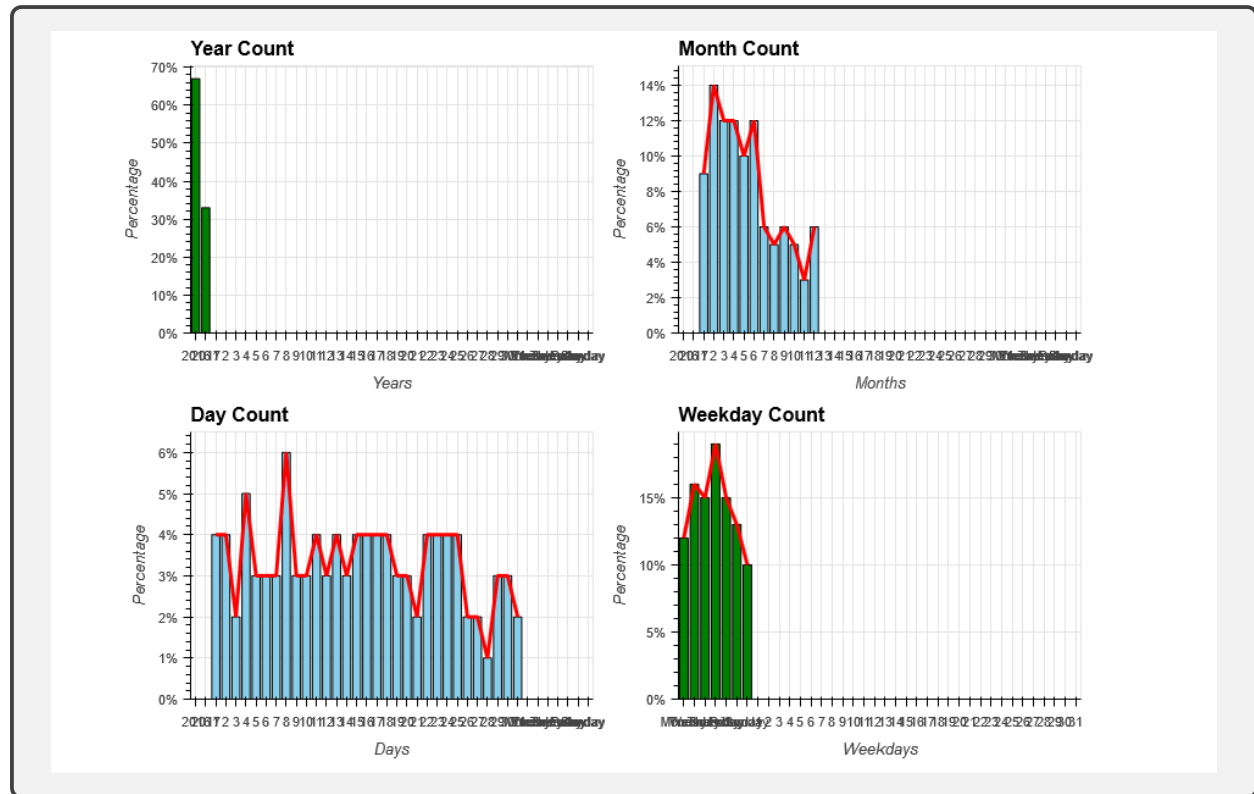
[3]More than half the incidents fall under the category 'Others'. A possible explanation might be that the entries for this database were entered through a form or program that had a limited number of valid inputs, so 'Others' was the most valid bin for the incident. It could also be that these entries were changed to 'Others' as part of the anonymization process. Or it also could be that the details were lost in translation.



[4]The number of accidents increased in Summer and Autumn. It is thought that the occurrence of accidents is related to the climate (especially temperature).



[5]It seems that the number of accidents decreased in later of the year/month.The number of accidents increased during the middle of the week and declined since the middle of the week.



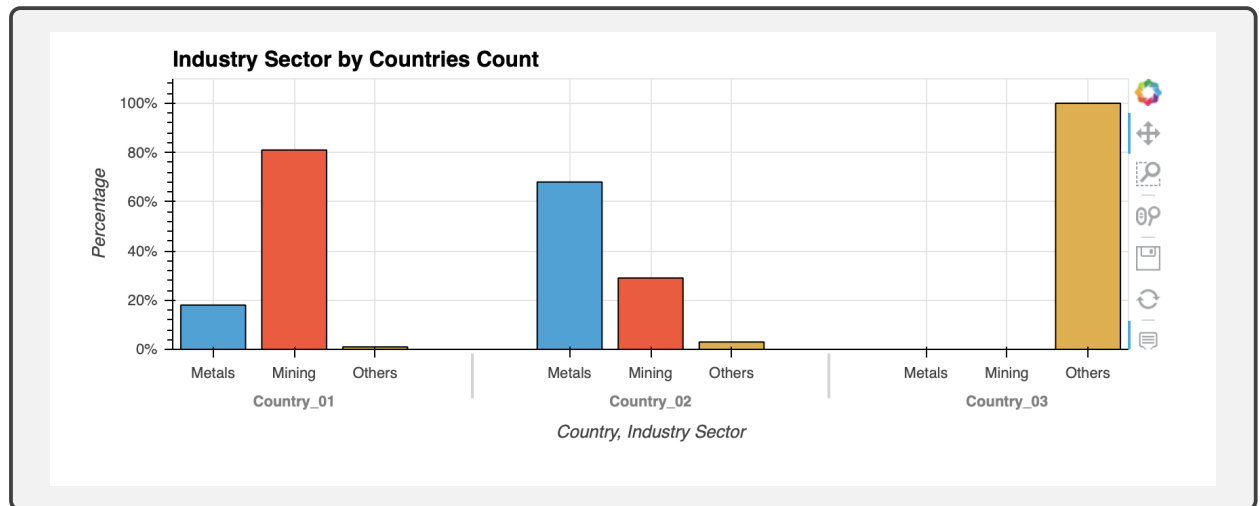
[6]Industry Sector by Countries Analysis:

We can see that there are major industries by countries.

Country_01 : Mining

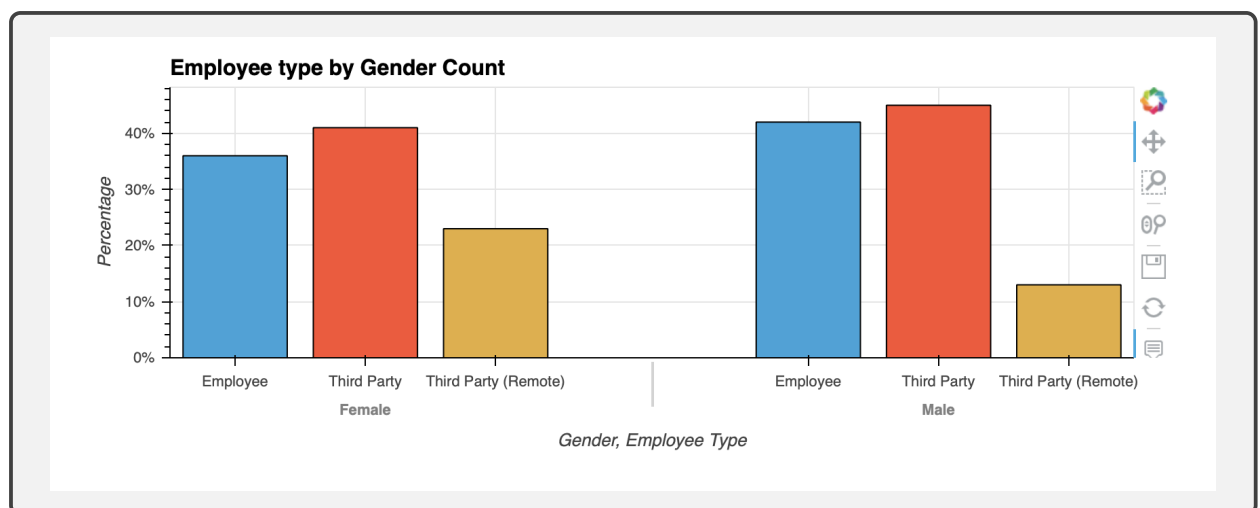
Country_02 : Metals

Country_03 : Other



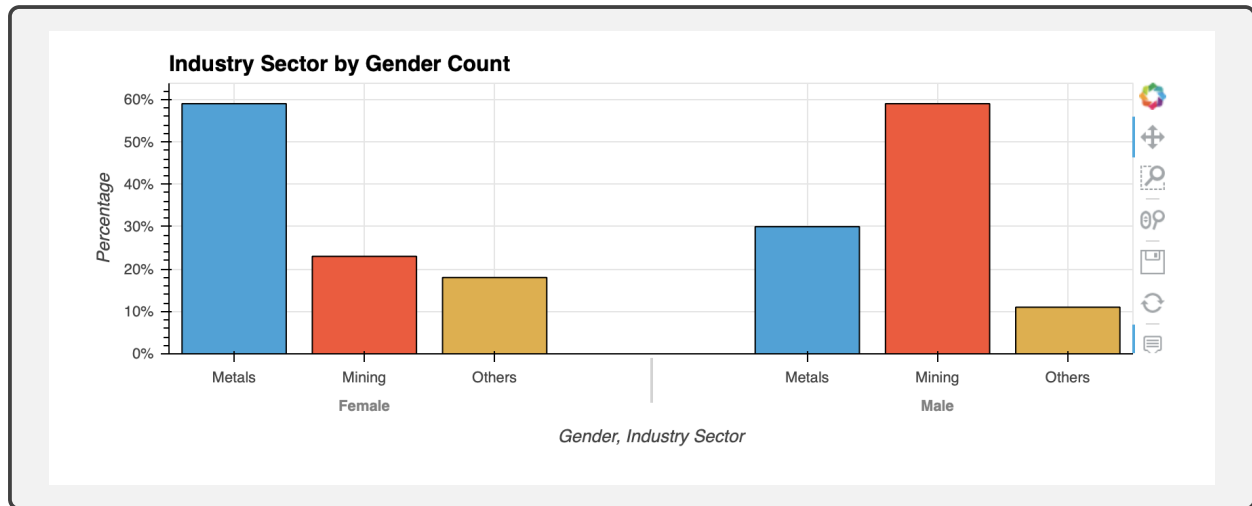
[7]Employee type by Gender Analysis:

Ratio of employee types by gender is not different in each gender. The proportion of female with Third Party(Remote) is slightly higher than that of males. It is thought that this is because males have more on-site work and females often do work far away from relatively safe sites



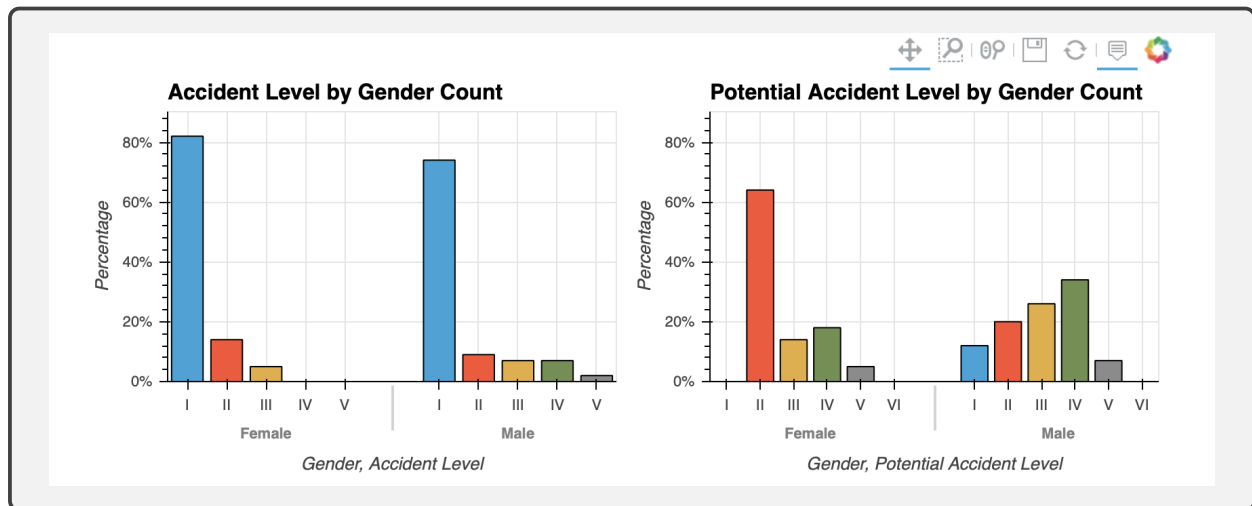
[8] Industry Sector by Gender Analysis:

There are differences mainly in metals and mining between males and females. Same as employee type above, it is thought that this is due to different safety level by industry sector.



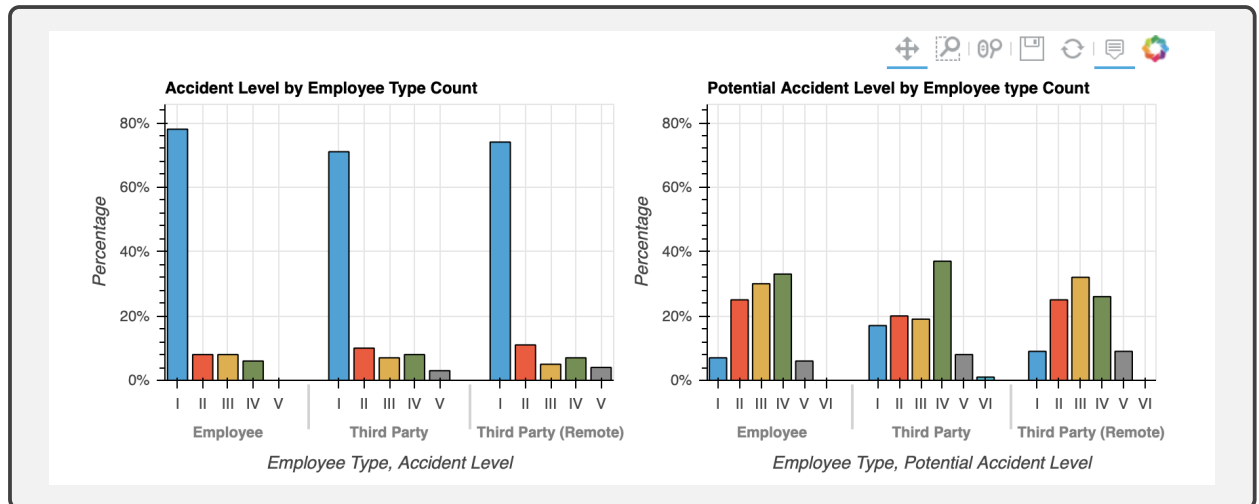
[9] Accident Levels by Gender Analysis:

In terms of accident levels, there are many mild risks at general accident level, but many serious risks at potential accident level. It can be said that many potential accidents are overlooked and potentially high-risk accidents are possible. In terms of gender, the general trend is the same, but males have a higher accident levels than females.



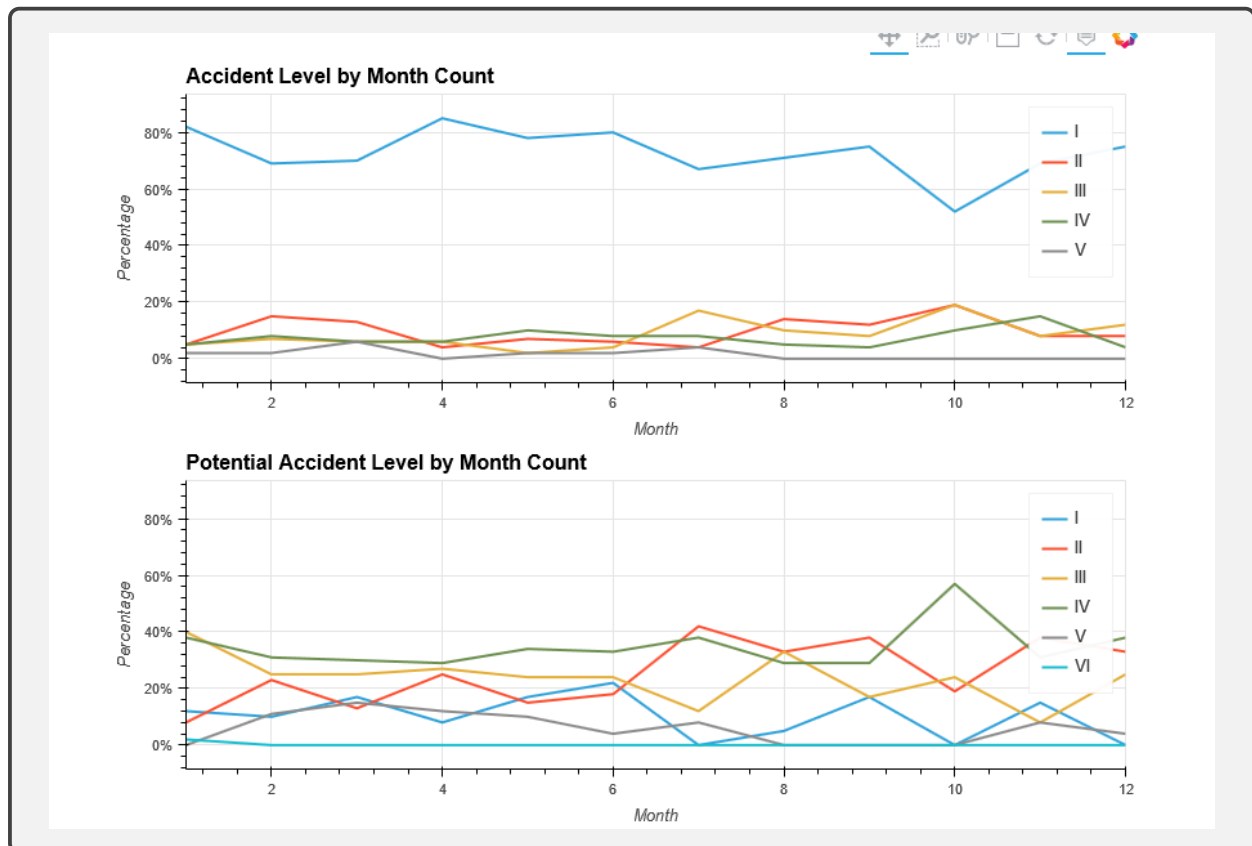
[10] Accident Levels by Employee type Analysis:

For both accident levels, the incidence of Employee is higher at low accident levels, but the incidence of Third parties seems to be slightly higher at high accident levels.



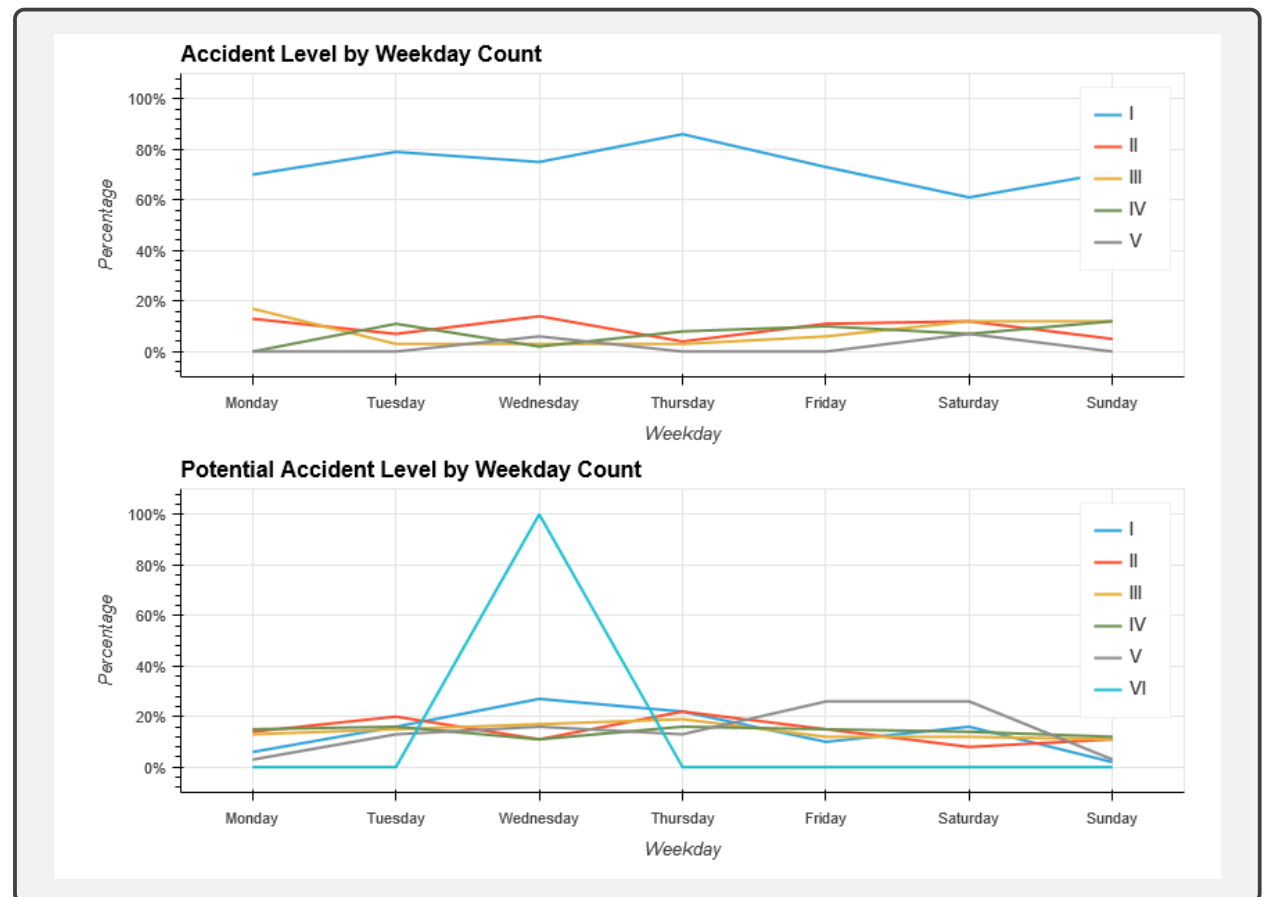
[11]Accident Levels by Month Analysis:

Both of the two accident level have the tendency that non-severe levels decreased throughout the year, but severe levels did not changed much, and some of these levels increased slightly in the second half of the year. The fact above seems to be related to the skill level of the employees, and while their experiences can reduce minor mistakes, sometimes they can make serious mistakes accidentally.



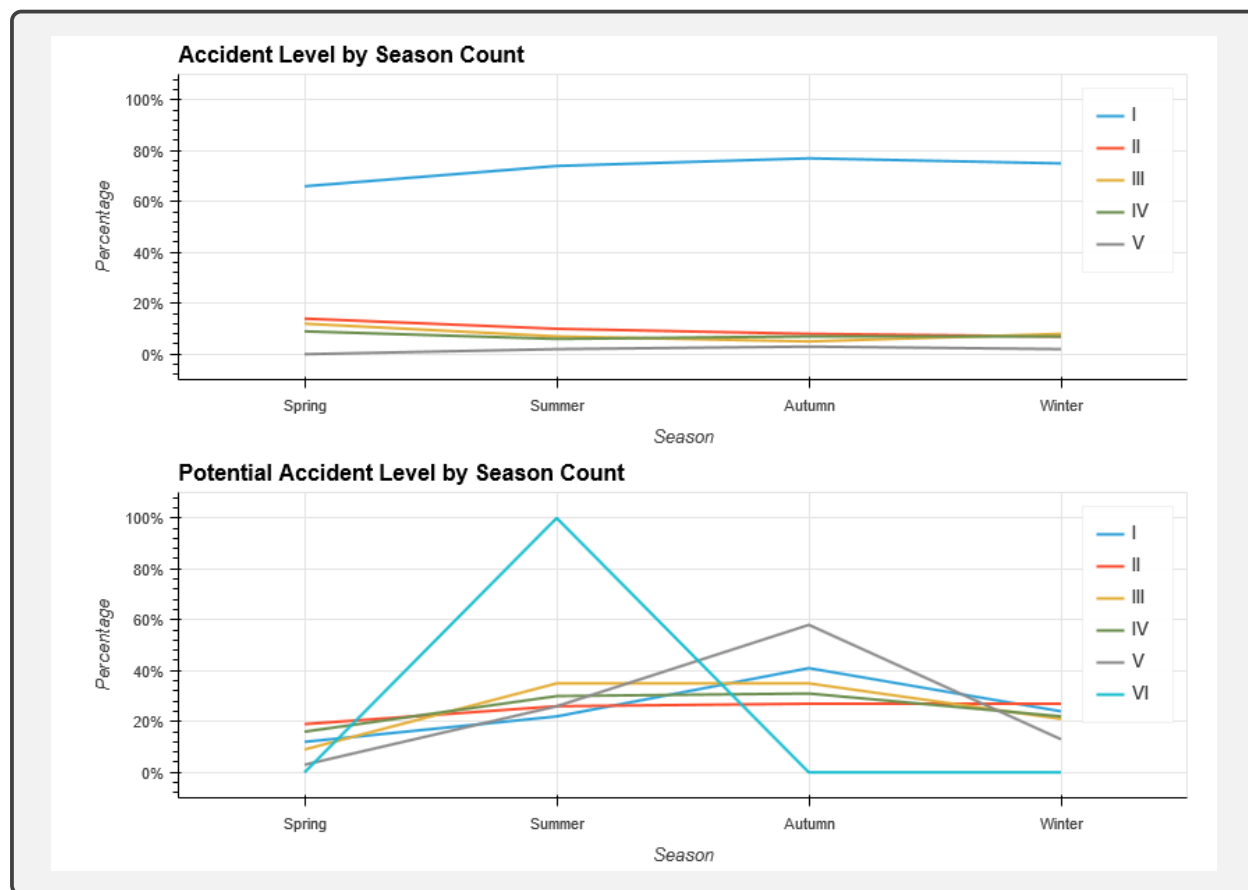
[12]Accident Levels by Weekday Analysis:

Both of the two accident level is thought that non-severe levels decreased in the first and the last of the week, but severe levels did not changed much. It can be said that employees' experiences against work can reduce minor mistakes.



[13] Accident Levels by Season Analysis:

As same as accident levels by month, both of the two accident level have the tendency that non-severe levels decreased throughout the year, but severe levels did not changed much, and some of these levels increased slightly in the second half of the year.



[14] Wordcloud Analysis with respect to accident level severity:

There are many hand-related and movement-related words occurring frequently in the description. Hand-related : left, right, hand, finger and glove Movement-related : fall, hit, carry, lift and slip



Fig1 : Wordcloud generated based on Accident Level 1

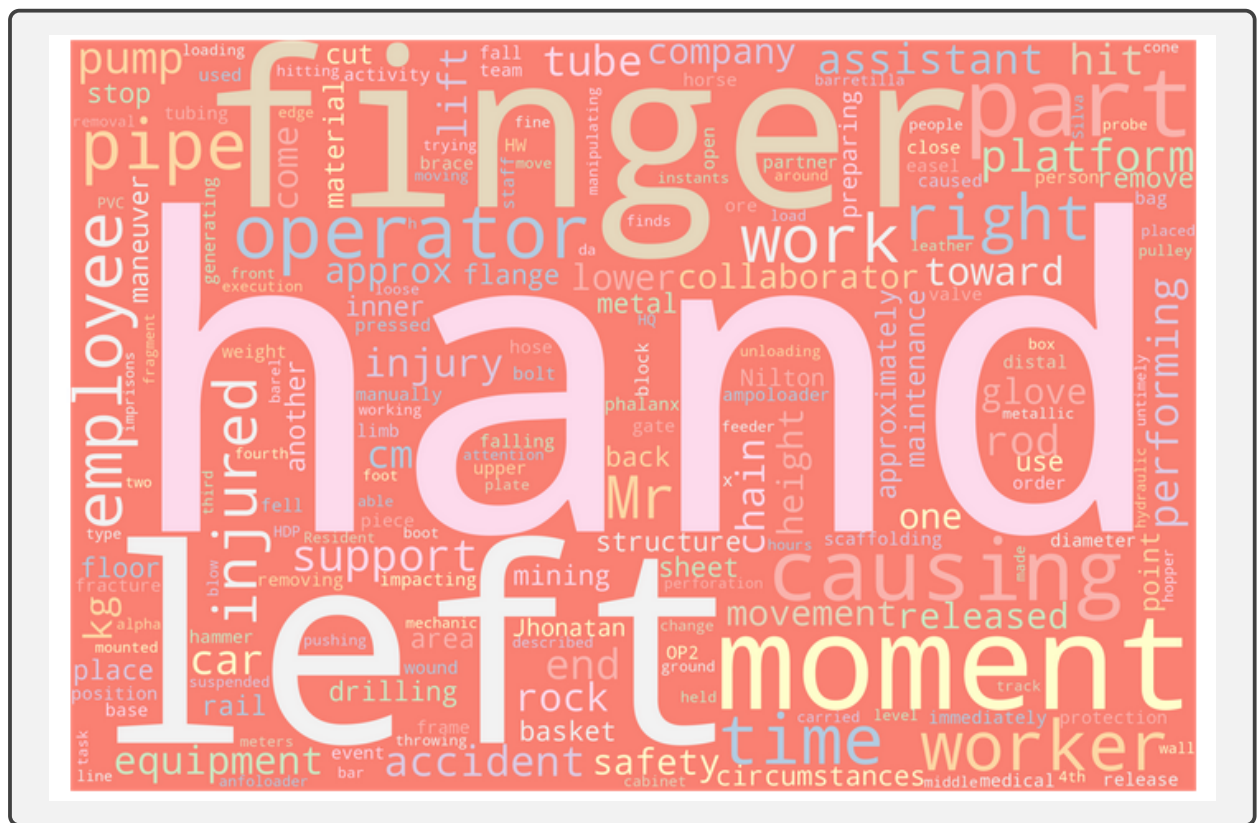


Fig3 : Wordcloud generated based on Accident Level 3



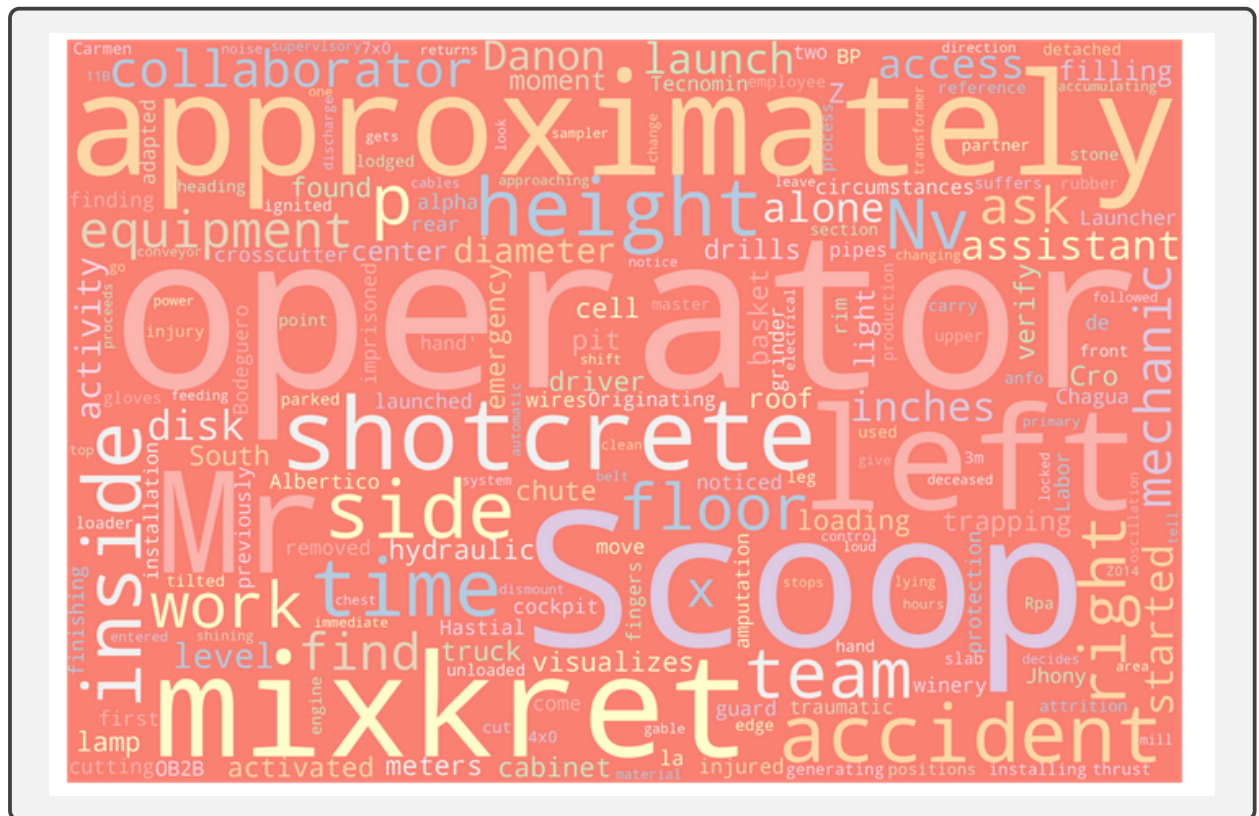


Fig5 : Wordcloud generated based on Accident Level 5

Inference:

Through a combination of basic NLP and rooting around in the data a reasonable case that Non-Safety Conscious Hand Operations are main cause of incidents. By being mindful of moving parts, using appropriate tool for job and refraining from excess force can reduce amount of incidents that occurs.

0.3 Data Modelling

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

0.3.1 ML Models and Neural Network:

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

Methodology-

Pre-processing-

[1.] Date column being splitted by Date, Month and Year

[2.] Description column had to undergo few data treatment like pre-processing (lowercase, stopwords, stemming, tensors) and word embedding using glove.

[3.] Created dummies for each column except Description and imputed values for each unique level of Potential Accident Level.

Opting for Potential Accidents Level as our target variable as Accident Level didn't have sufficient data points for each severity level which was evident by most of the data falling under severity 1 contributing to biasness. Concluding this basis on the lowest accuracy, precision and recall when model was applied on raw data as well as over sampled data considering accident level as target variable. Compare to Accident Level, Potential Accident Level has somewhat balanced values except for the fact Level 6 has only 1 record. Hence, dropped Accident Level.

Modeling- We applied different ML models like RandomForestClassifier, DecisionTreeClassifier, BernoulliNB, BaggingClassifier on raw data and found that Random Forest give better testing accuracy than compared to other models. In order to improve our model validation, we tried replicating the data for and applied upsampling using SMOTE as there was only one value for severity level 6 of Potential Accident Level which when splitted into train and test would get included in either of the two so to avoid this bias, the records were balanced using SMOTE and then models were applied and we observe that BernoulliNB performs far better than any other models applied so far.

Note: Although SMOTE has proved to be an effective tool for handling the class imbalance problem, it may overgeneralize the minority class as it does not take care of the distribution of majority class neighbors, especially when the minority class is very sparse with respect to the majority class. Hence in this case study we have first replicated the data for each class then proceeded with SMOTE.

Applied Dense NN with different activation functions and parameter tuning, it yields good accuracy with train data but worst validations with test data due to overfitting which might be caused due to complexity of the model but even if dropout layer were included, model didn't perform better.

	Model	accuracy
0	RandomForestClassifier	0.400000
1	RandomForestClassifier GloVe	0.445312
2	DecisionTreeClassifier	0.421875
3	BernoulliNB	0.414062
4	BaggingClassifier	0.406250
5	RandomForestClassifier SMOTE	0.356322
6	DecisionTreeClassifier SMOTE	0.344828
7	SGDClassifier SMOTE	0.333333
8	BernoulliNB SMOTE	0.471264
9	BaggingClassifier SMOTE	0.356322
10	Neural Network	0.297043
11	Neural Network tune	0.285455

We clearly see that BernoulliNB using SMOTE outperforms other models and also gives better results in terms of accuracy, precision, recall and F1 measure.

0.3.2 Bidirectional LSTM Model:

Bidirectional recurrent neural networks(RNN) are really just putting two independent RNNs together. This structure allows the networks to have both backward and forward information about the sequence at every time step.

Methodology-

Post preprocessing, all the columns were clubbed together with Description.

Getting all the columns in description

```
In [27]: df1['Description'][0]
Out[27]: 'Country_01 Local_01 Mining Male Third Party Pressed 2016 1 January removing drill rod jumbo maintenance supervisor proceed
s loosen support intermediate centralizer facilitate removal seeing mechanic supports one end drill equipment pull hands ba
r accelerate removal moment bar slides point support tightens fingers mechanic drilling bar beam jumbo'
```

Indexing each word for tensors value. When processing sequence data, it is very common for individual samples to have different lengths, hence used padding. Padding comes from the need to encode sequence data into contiguous batches: in order to make all sequences in a batch fit a given standard length, it is necessary to pad or truncate some sequences. Since the input data for a deep learning model must be a single tensor, samples that are shorter than the longest item need to be padded with some placeholder value. Post applying SMOTE, model doesn't perform better and hence we stick towards original data test validation.

	precision	recall	f1-score	support
0	0.91	0.67	0.77	15
1	0.40	0.46	0.43	26
2	0.38	0.37	0.38	27
3	0.50	0.62	0.55	29
4	0.33	0.10	0.15	10
5	0.00	0.00	0.00	0
accuracy			0.48	107
macro avg	0.42	0.37	0.38	107
weighted avg	0.49	0.48	0.47	107

Since Bidirectional LSTM performs slightly better than other models, we would be considering this model to proceed ahead with our Chatbot.

0.4 Building Chatbot

We will be exploring Tkinter – python GUI programming tool to build our chatbot for this project. We will explore how we can deploy a model and check real-time predictions using Tkinter. We will first build a classification model that will classify on Potential Accident severity level. Then we will make a GUI using Tkinter and will check predictions on new data points.

Methodology-

[1]Creating GUI

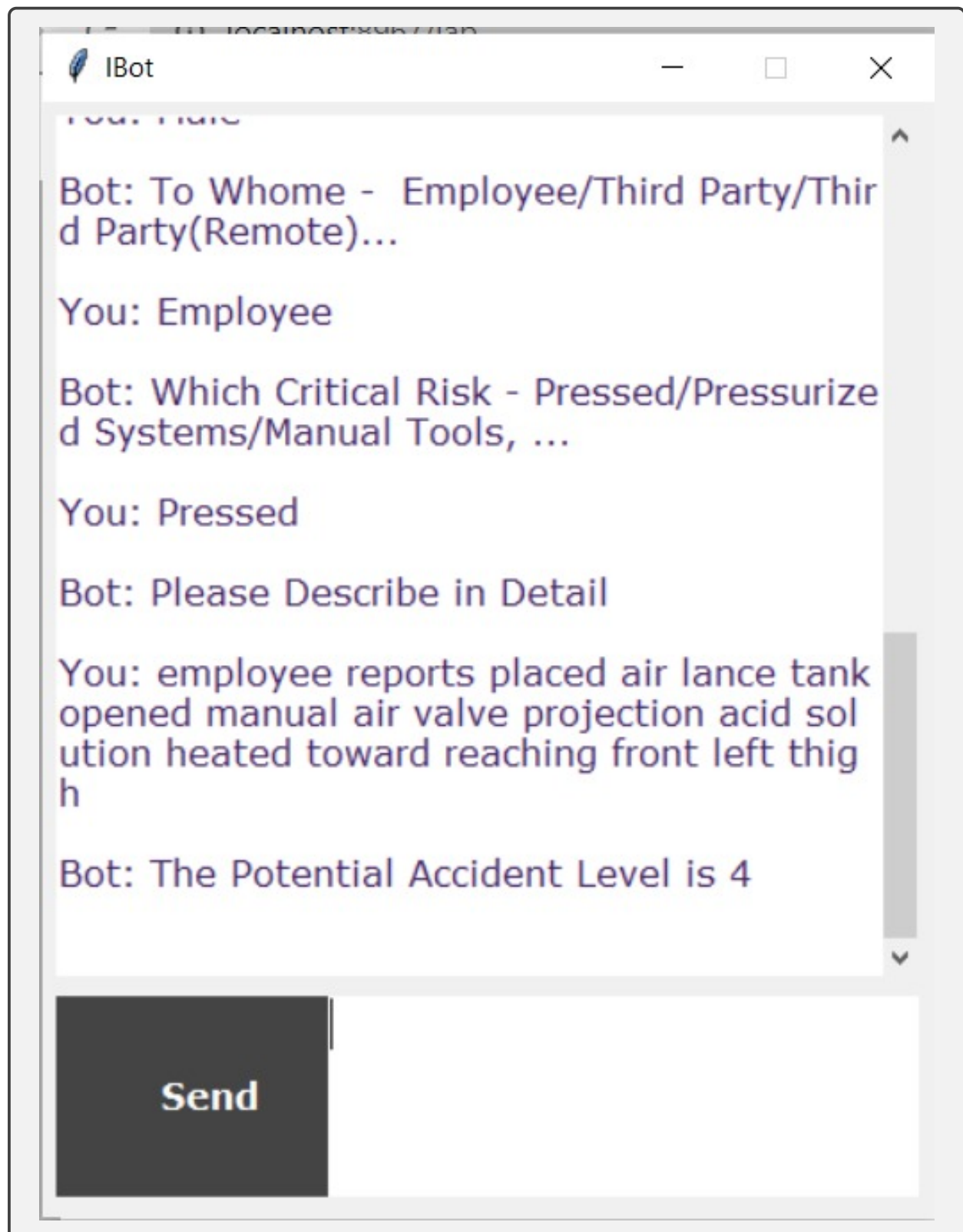
Tkinter is a library written in Python that is widely used to create GUI applications. It is very easy to build GUI using Tkinter and the process is even faster. Tkinter has several widgets that can be used while developing GUI. These include buttons, radio buttons, checkboxes, etc. We will see how we can make a GUI Tkinter after we build the model.

[2]Model Building

As discussed earlier, we are proceeding ahead with Bidirectional LSTM model as the accuracy is higher than compared to other models. Once we are done with the model validation, we pickle this model that would be used to compute predictions for new data points.

[3]Computing Predictions Using the Tkinter GUI in Real-Time

To predict the class, we will need to provide input in the same way as we did while training. So we will create some functions that will perform text preprocessing and then predict the class. After predicting the class, we will get a random response from the list of intents. Now we will develop a graphical user interface. We will take the input message from the user and then use the helper functions we have created to get the response from the bot and display it on the GUI.



Execution :-

For the Execution part we have created separate files. Those files are:-

Source_File

Main_File

Utils

Saved Model

Source_File:-

This file is an independent file. Not linked to any other file. This file imports data, cleans the data and applies the Bi-Directional LSTM model on the data. We have not used SMOTE or any other Data Balance techniques. We have taken Potential_Accident_Level as our Target Variable. We have saved our model from this file. The accuracy of the model turns out to be 48

Main_File :-

Main file is predominantly a TKinter GUI File which creates the Chatbot GUI.

Also, we load the model, the model which we saved from Source_file. We import a function from util.py to create the inputs taken from USER to tensor.

We have created the intents for greeting and have also put the conditions to verify the user input.

Utils (Utils.py):-

This is the python file, where we have imported data, created vocabulary list and created a function to convert the given input description to tensor.

The “tweet_to_tensor” function is the function which we will be calling in our Main_File.

Saved Model (Model.h5):-

This is the model which we have saved from our Source_File.

Inference:

In this project, we understood about chatbots and implemented a deep learning version of a chatbot in Python. Probably we are suffering from Data Insufficiency, hence even upsampling of data is not helping much on modelling accuracy.

However, we have successfully built a semi-ruled chatbot which helps predicting Potential Accident Level in accordance with the user’s input.

Future Work:

We can enhance this project by building AI chatbot. (0.1)

References

- 1) <https://www.chatbot.com/chatbot-guide/>
- 2) <https://www.forbes.com/sites/cognitiveworld/2020/02/23/choosing-between-rule-based-bots-and-ai-bots/?sh=35b006f0353d>
- 3) <https://analyticsindiamag.com/complete-tutorial-on-tkinter-to-deploy-machine-learning-applications/>
- 4) <https://chatbotslife.com/introducing-conversational-chat-bots-using-rule-based-applications/>
- 5) <https://www.python-engineer.com/posts/chatbot-gui-tkinter/>