AIML

CAPSTONE PROJECT - FINAL REPORT

NLP CHATBOT For Industry Safety

Team Mentor

Rohit Gupta

Team Members

Sarita Ghadge
Rohan Bhattacharya
Meetu Chandra
Aditya Sharma
Nishant Kumar Thakur

Problem Statement

Major Industries in Brazil are facing multiple workplace hazards leading to multiple accidents, some even leading to death. The aim of this project is to build an AI powered Chatbot for employees and stakeholders to understand the various reason for the accidents. This can help the employees take the necessary precautions and for the stakeholders to identify the root cause for the accidents and put a fix in place.

The Chatbot will be powered by an AI model that has been trained to classify the accident, given a summary of the various scenarios when the accident happened in past.

Approach

An NLP based AI model will be trained on industry data from **3 countries** and **12 plants.** The data is from industries operating majority in Metal and mining. Machine learning models using different strategies will be developed and compared to identify the best performing model.

Overview of Final Process

To accomplish this project, we followed a systematic methodology consisting of data exploration, preprocessing, and model development

1. **Data Exploration:** We began with exploratory data analysis (EDA) to gain insights into the dataset's structure, size, and quality. Some column names needed renaming for better representation.
2. **Data Preprocessing:** The pre-processing step includes creating a data preprocessing/cleaning pipeline.
3. **Data Augmentation:** Since the dataset was limited, we used the Language Model (LLM) to augment it by adding synthetic data using the Gemini Frontier Model API. We created a data-preprocessing/cleaning pipeline for this synthetic data, ensuring that it was consistent with the existing data.
4. **Model Selection and Combination:** We used multiple models like traditional models like Random Forest and improved them. We then started with powerful neural network models like ANN, LSTM, etc., which were improved upon as well. Finally, we used state-of-the-art LLM models for more accurate predictions
5. **Result in metrics analysis:** Models were evaluated based on multiple metrics. New architectures / models were tested based on them and eventually, a consistently best-performing model was selected

Data Overview

We analysed data from one of the largest industries in Brazil, which consists of 425 accident records from January 2016 to September 2017 across 3 countries and 12 locations. The dataset includes key information provided in the table below

S. No.	Column Name	Description about the column
1	Unnamed	Index Column **Data type**: Integer
2	Data	Date of accident **Data type**: datetime **Range**: January 2016 to July 2017
3	Countries	which country the accident occurred (anonymised) **Data Type:** Object **Unique values**: Country_01 Country_02 Country_03
4	Local	The city where the manufacturing plant is located (anonymised) **Data Type**: Object **Unique values**: Local_01 to Local_12
5	Industry sector	Which sector the plant belongs to?

		Data Type: Object **Unique values**: Mining, Metas, Others
6	Accident level	From I to VI, it registers how severe was the accident (I means not severe but VI means very severe) **Data Type**: Object **Unique values**: I,II, III, IV, V
7	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) **Data Type**: Object **Unique values**: I,II, III, IV, V, VI
8	Gender	If the person is male of female **Data Type**: Object **Unique values**: Male, Female
9	Employee or Third Party	if the injured person is an employee or a third party **Data Type**: Object **Unique values**: Third Party Employee Third Party (Remote)
10	Critical Risk	Some description of the risk involved in the accident **Data Type**: Object **Unique values**: There are number of unique values like – could be a short description for the accident description.
11	Description	Detailed description of how the accident happened. **Data Type**: Object (textual data)

Table: Data columns and their description

The below steps were performed:

Exploratory Data Analysis (EDA)

Initial Analysis:

There were 425 records with 10 columns. None of the columns had missing values. All the columns except for Date and description were categorical.

Renaming columns and removal from initial analysis

Few of the column names had to be renamed to as per the data they were representing: -

- "Data" was renamed to Date
- "Genre" was renamed to Gender.
- "Unnamed" which was an index columns was dropped.
- "Countries" was renamed to "Country"
- "Local" was renamed to "Location"
- "Industry Sector" was renamed to "Industry_Sector"
- "Accident Level" was renamed to "Accident_Level"
- "Potential Accident Level" was renamed to "Potential_Accident_Level"
- "Employee or Third Party" was renamed to "Employee_Type"
- "Critical Risk" was renamed to "Critical_Risk"

Check for Missing values

There were no missing values in the dataset

Check for Duplicates

- There were 7 records that were duplicates, and the duplicate records were dropped. That left us with 418 records.
- Out of the remaining 418 records, 7 accident descriptions are repeated in the data.
- These are accidents that happened at the same time, where a group was involved, and there are different records for each person.
- As this corresponds to different records they were not removed

Pre-Processing "Data" Column

The Date column was split into Year, Month, Date, and Day of the week to check if there is any pattern.

Unique values of the categorical columns

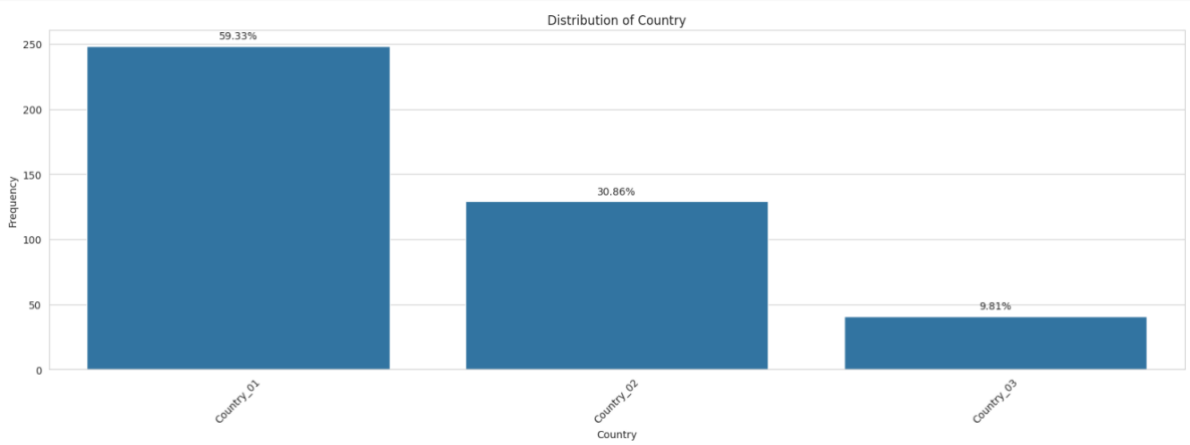
Unique values for the categorical columns were identified (the values are available in the above table, *Table: Data columns and their description).*

Univariate Analysis

Count plots were used to understand the spread of data across each feature.

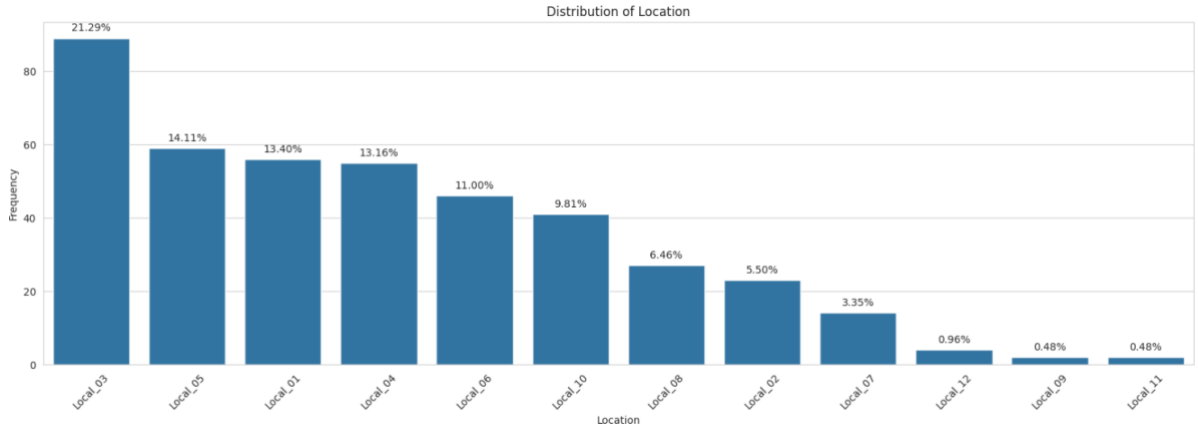
Country

There is data for 3 countries, 60% of the data is for Country_01, 31% for countrty_02 and only 10% for country_03. It is possible country_01 is more prone to accidents.



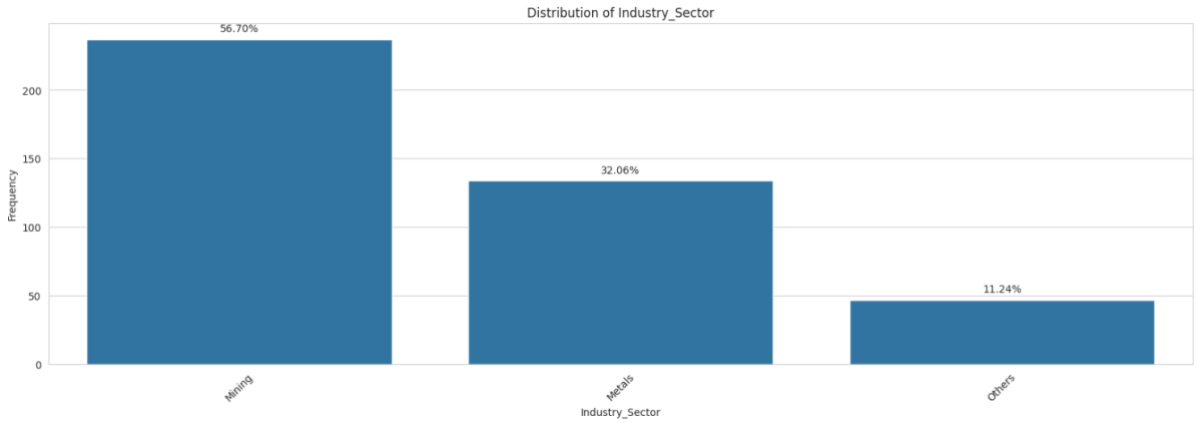
Location

There is data for 12 locations across 3 countries. Local_03 has seen the maximum number of accidents, which is around 20% of all the accident cases recorded.



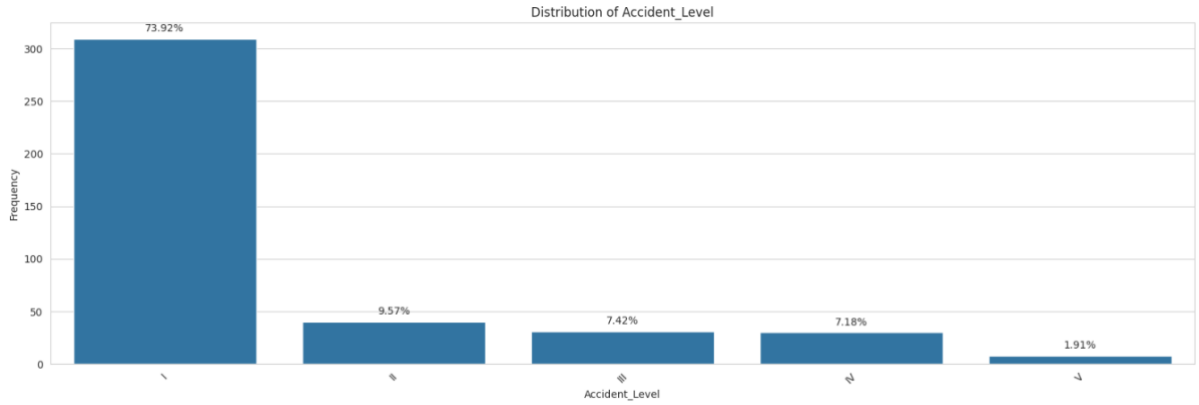
Industry Sector

Mining sector has the most accident cases than any other sector. Thus, we can say that jobs in the mining industry sector are riskier than metal or any other sector.



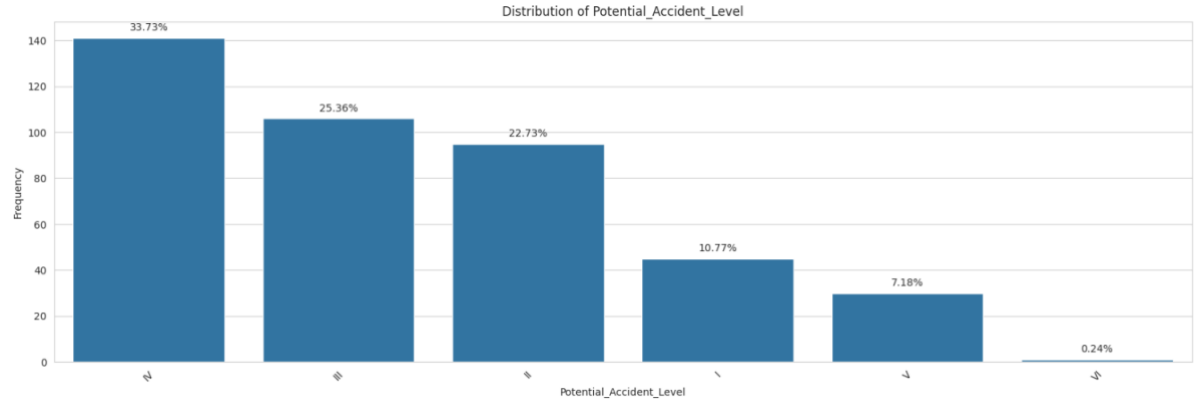
Accident Levels

Accident levels are mostly of Level 1 with 74% of the data, followed by 9.57% of level 2 and ~7% for levels 3 and 4 and ~2% for level 5. There have been no accidents of level 6 which is the highest level.



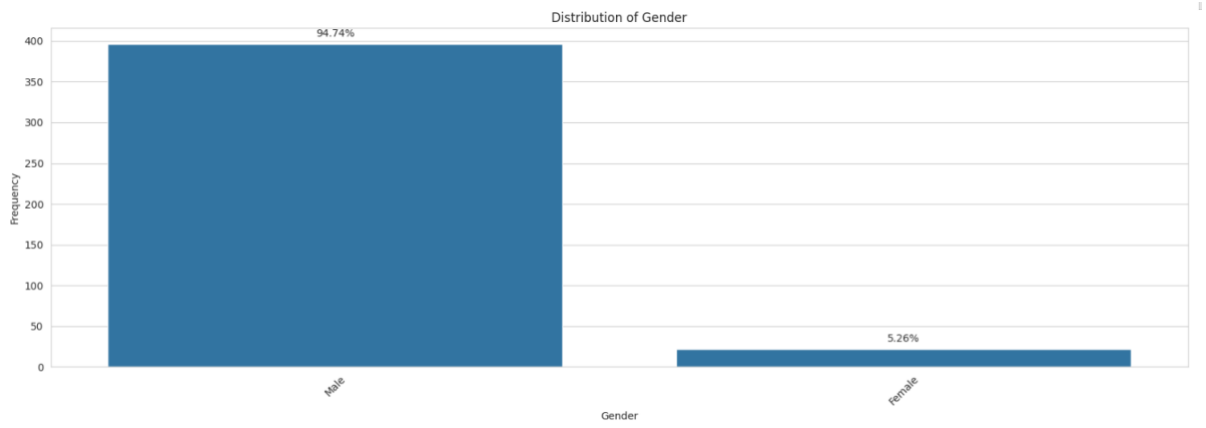
Potential Accident Level

Potential accident level indicates how severe the accident would have been due to other factors involved in the accidents. As per the graph, level IV has the highest count, which corresponds to moderate severity of accidents, followed by 25.3% of level 3. Also 0.24% chances of the most severe level 6.



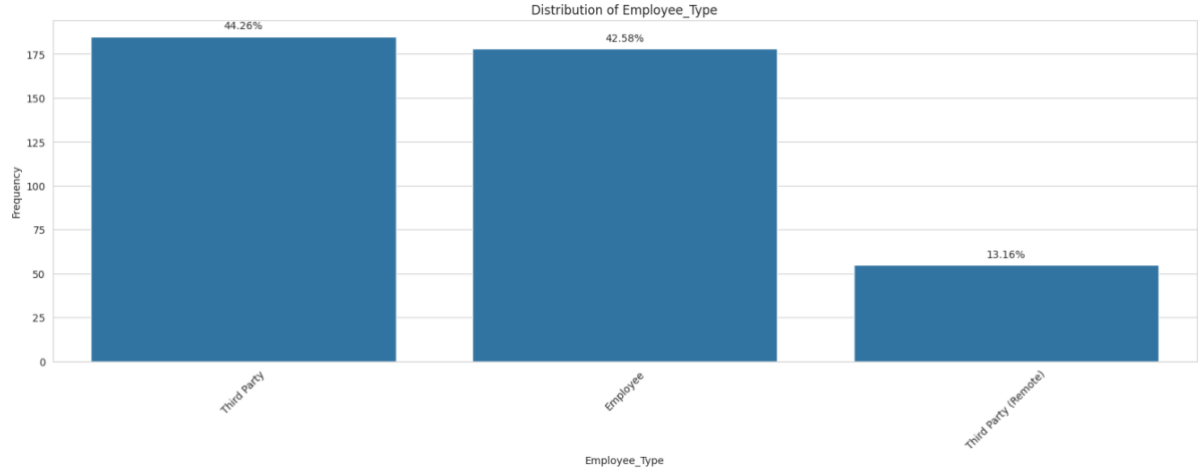
Gender

Dataset is more biased towards male employees; this is possible because Mining and Metal industries are more male dominant.



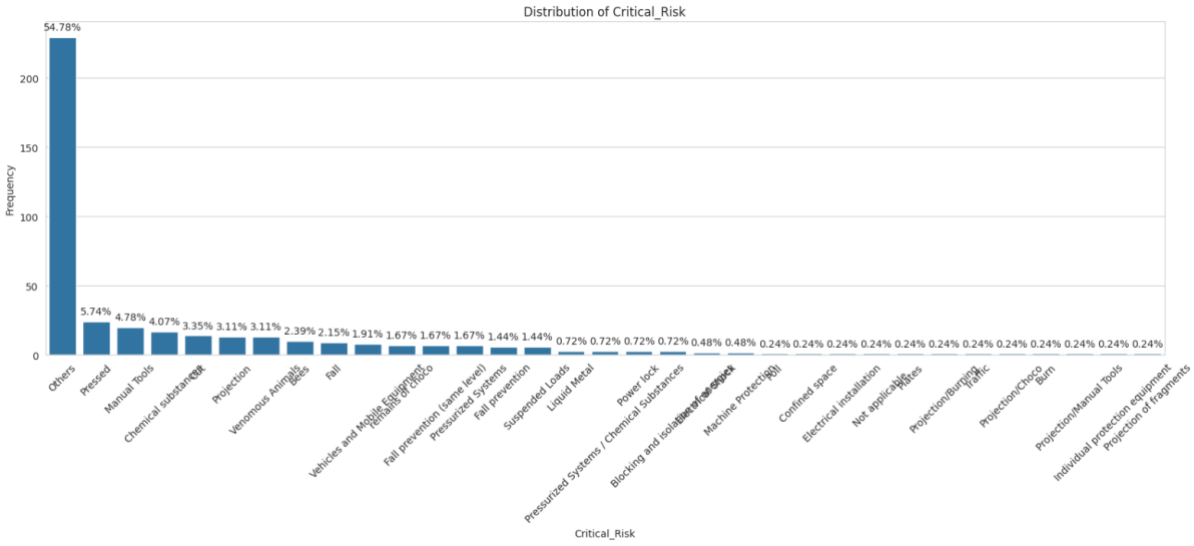
Employee Type

Total number of internal employees and Third-Party employees is more or less the same. But, we can also see that Third party remote employees are comparatively less in number.



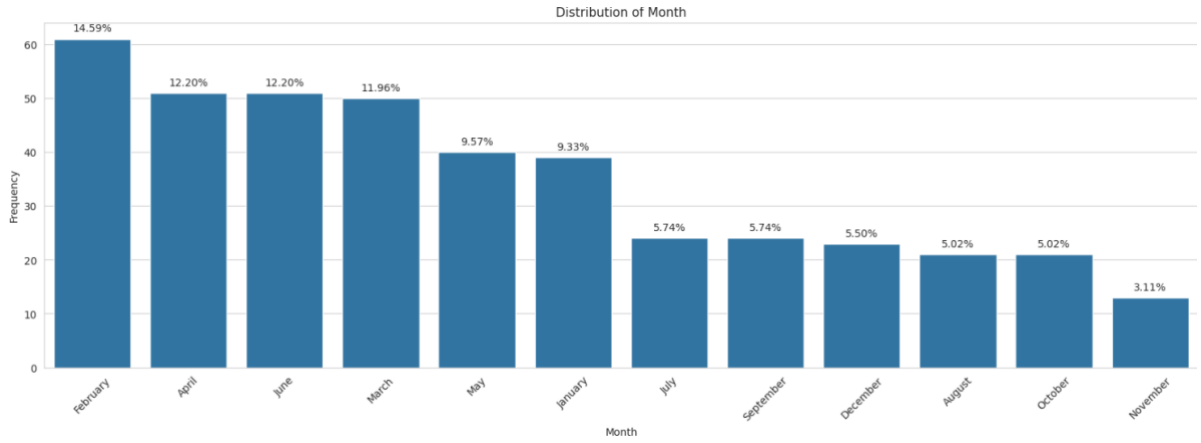
Critical Risks

Most of the Critical Risks are classified as 'Others'. It holds around 55% of the total Critical Risks. It is followed by Pressed, Manual tools, Chemical substances, etc.

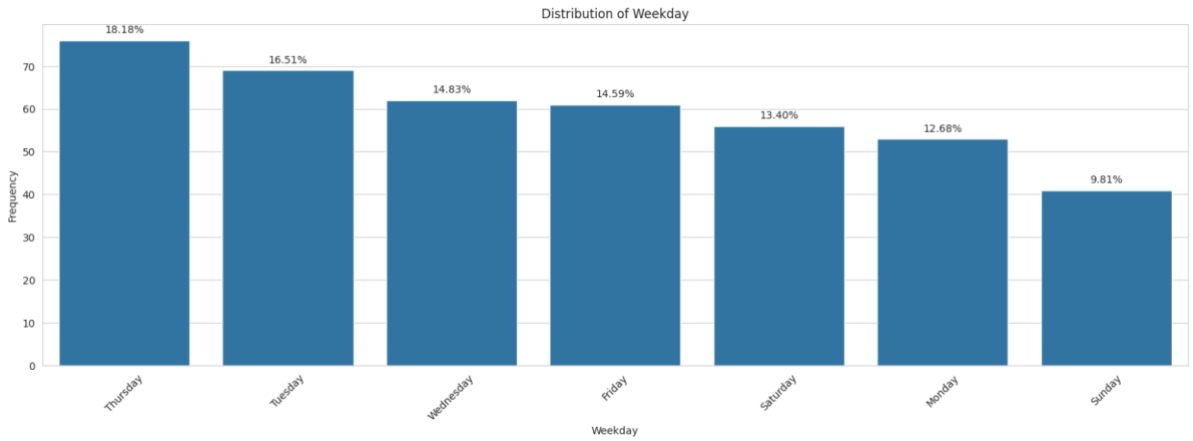


Date

- Data was available of 2 years 2016 and 2017.
- Year and Day column did not seem to any relevance for the analysis and were dropped.
- Accidents were more frequent in the first half of the year with maximum accidents in February at 14.59% followed April, June and March. November recorded the least number of potential accidents.



- Thursdays are more prone to accidents followed by Tuesday. Sunday has the least

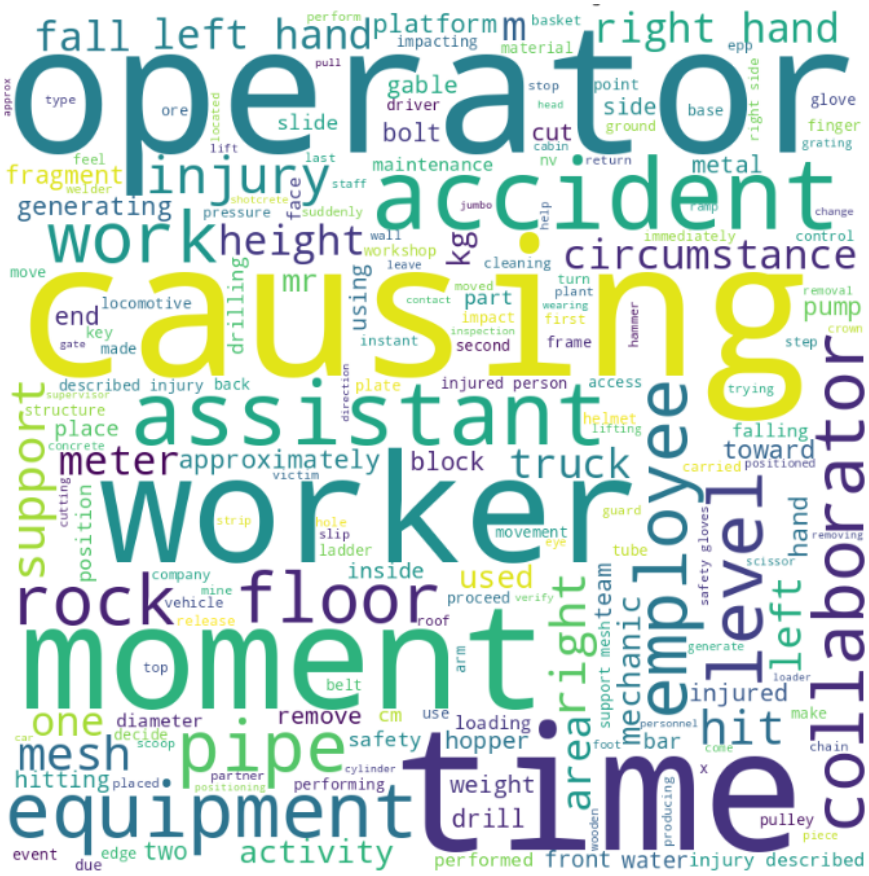


Description

Word Cloud was plotted to see which words were most frequent for the various industries and accident levels.

For Industry

Mining



Metals

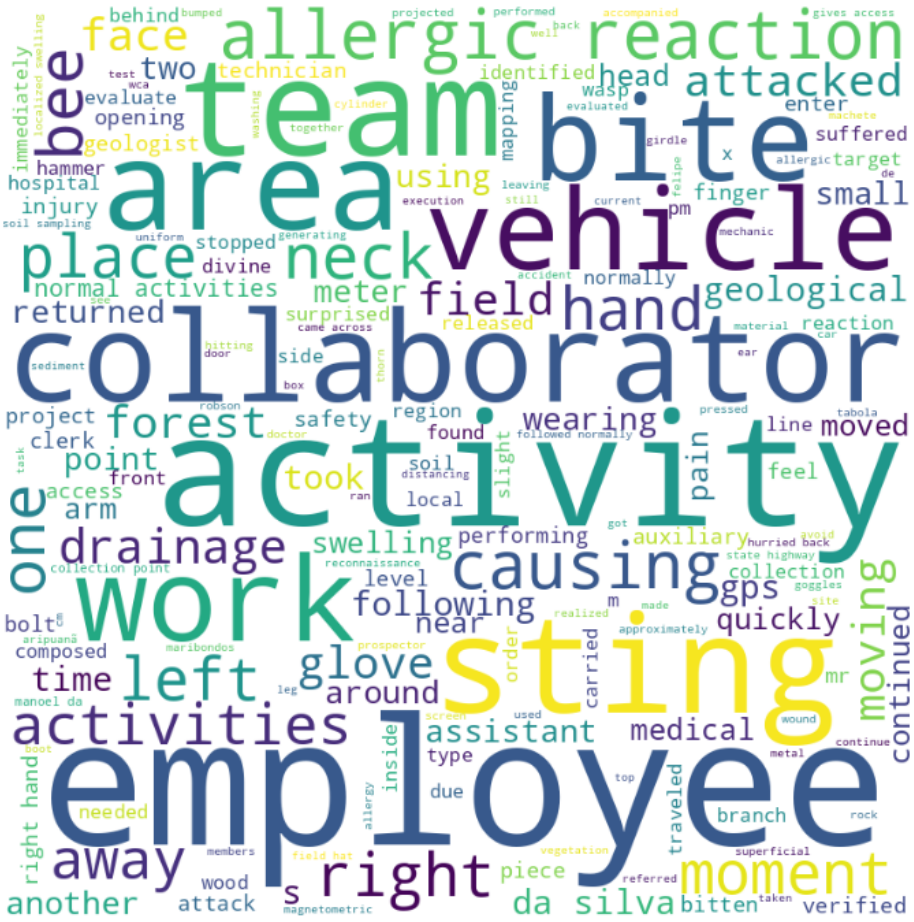
Others

Observations

Industry	10 most Frequent Keywords	Observations / Comments
Mining	causing, operator, time, worker, moment, accident, assistant, equipment, level, work	most of the accidents seem to be related to equipment and the impacted people were the equipment's operators.
Metals	employee, causing, operator, hit, activity, left hand, right, finger, left, area	most of the accidents seems to have been caused by some injury to the left hand or right finger where the operator was hit by something.
Others	employee, activity, team, causing, sting, bite, area, hand, left, right Looks like	most of the accidents seem to be due to bee bites on left or right area of the body.

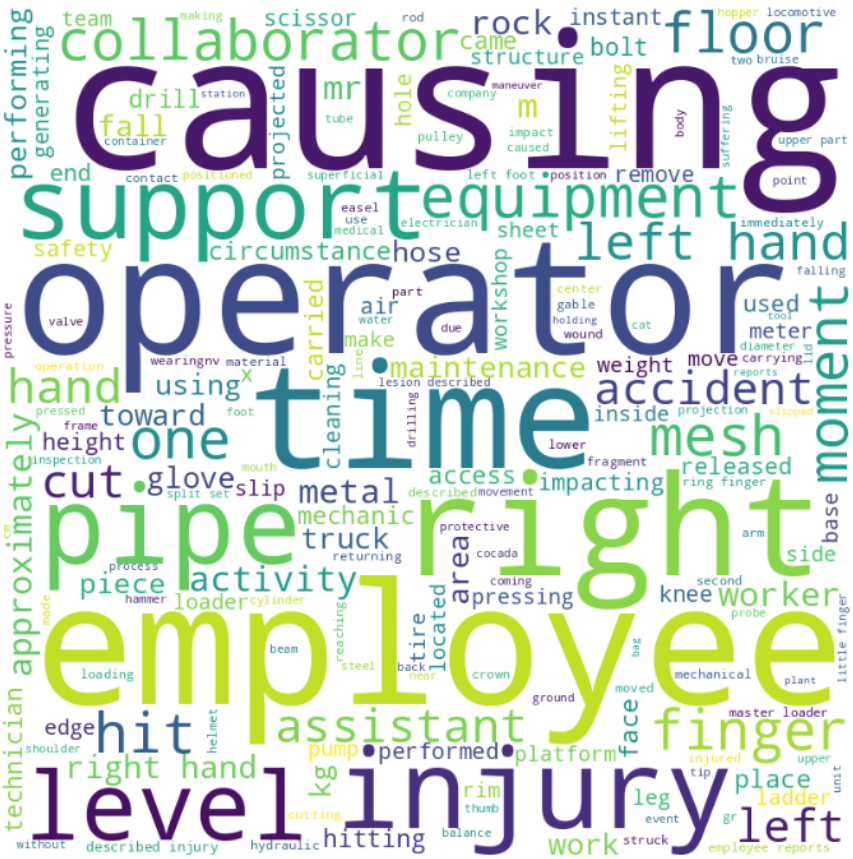
Word Cloud For Potential Accident Levels

1. Level 1



2. Level 2

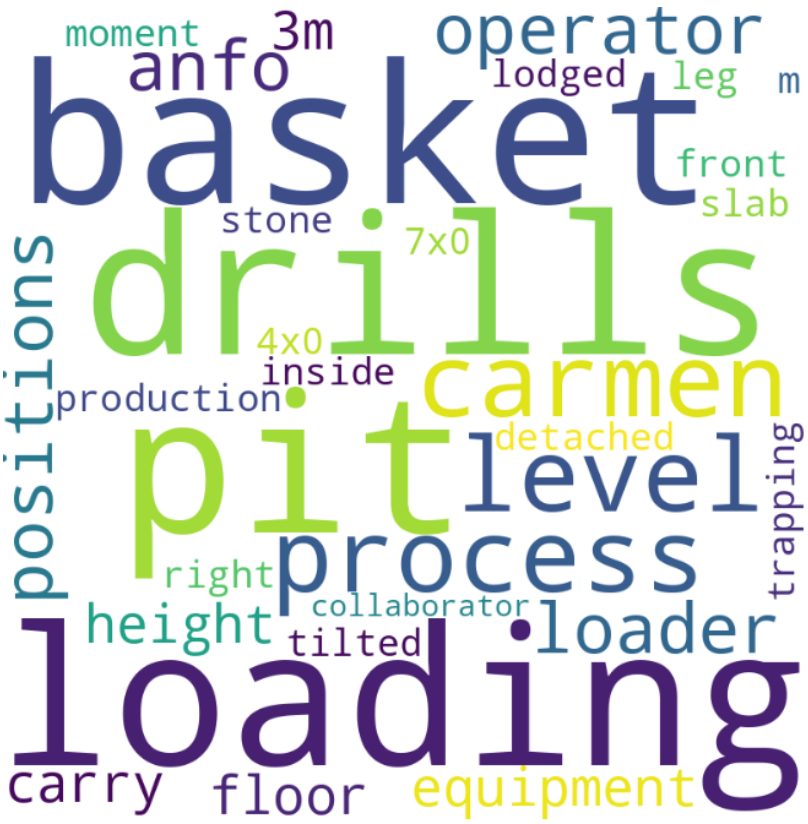
3. Level 3



4. Level 4

5. Level 5

6. Level 6



Observations

Potential Accident Level	10 most Frequent Keywords	Comments
Level 6	loading, drills, pit, basket, process, carmen, level, operator, positions, anfo	Major severity accidents seem to be forecasted at the drilling sites that could involve miners going down the cave in a basket or carmen using the vehicles on the site.
Level 5	operator, left, right, worker, side, equipment, m, work, part, causing	Severity 5 accidents are more prone with related to equipments were a part of the human body is injured.
Level 4	causing, operator, time, moment, employee, assistant, worker, accident, work, fall	Level 4 are more related to accidents that could happened due to falls.
Level 3	causing, employee, operator, time, right, injury, pipe, support, level, equipment	Level 3 could be injuries caused due to incorrect levels and burns due to pipe emissions
Level 2	causing, employee, right, worker, activity, mesh, left, hit, cut, time	Levels 2 could be minor cuts during working
Level 1	employee, activity, team, sting, collaborator, area, work, bite, vehicle, allergic reaction	Level 1 seem to be more related to bee stings and allergic reactions to some fumes/chemicals used in the industry.

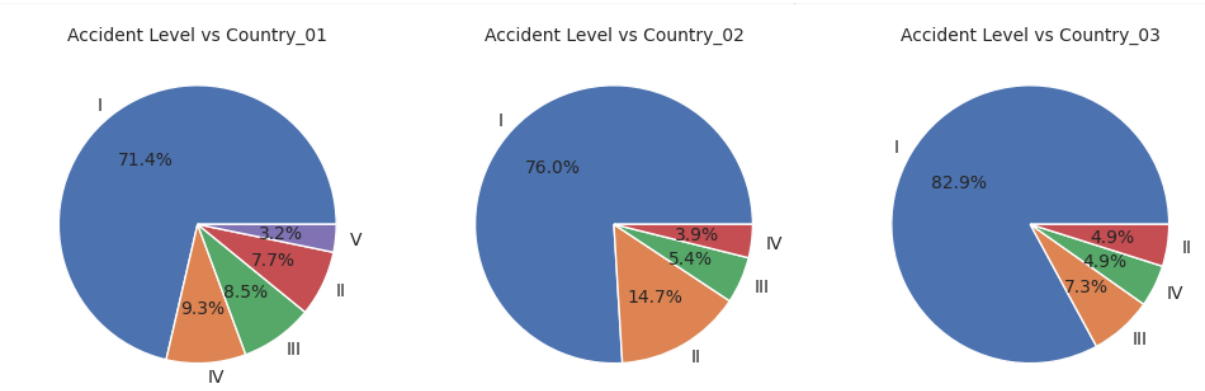
Bivariate Analysis

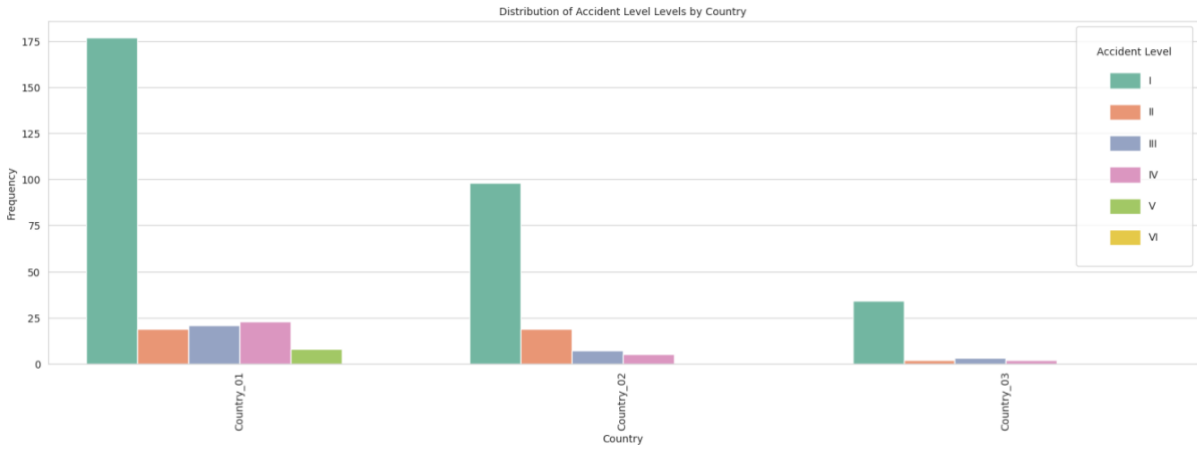
Bivariate analysis was then performed to see the patterns and spread of each feature with the accident levels and Potential accident levels. Below are the results and the observations:

With Accident Levels

Accident Level/Country

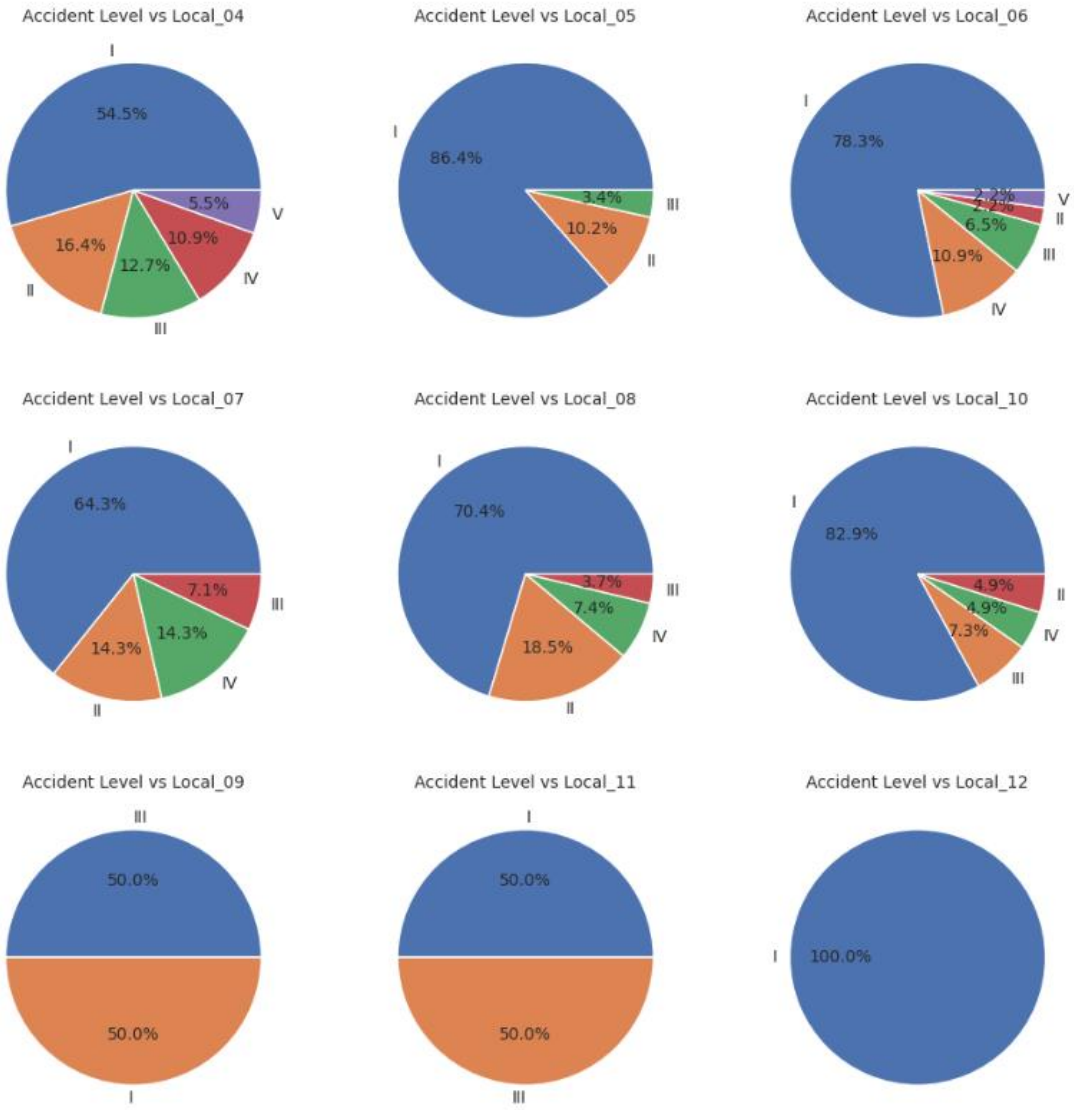
- More than 70% of accidents in all three countries are Level 1 accidents.
- The severity of accidents is highest in count_01. All the Level 5 accidents occurred only in country_01. It would be interesting to investigate further to find out why this is. Why are the most severe accidents specific to country_01?

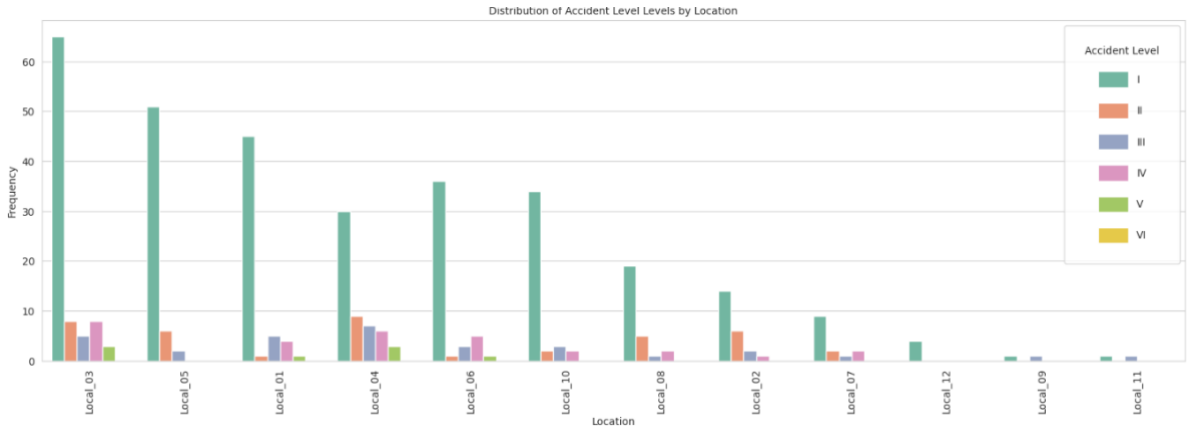




Accident Level/Location

1. Majority of accidents in all localities are level 1 accidents. (In fact, local_12 observed only level 1 accidents.)
2. Local_09 and local_11 observed only level 1 and 3 accidents, that too in equal proportion.
3. We can observe that the proportion of Level 4 and level 5 accidents is the highest in local_04.

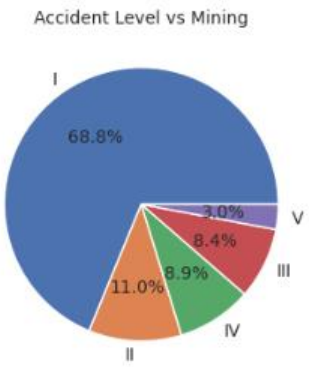


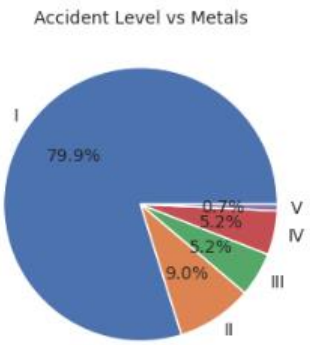


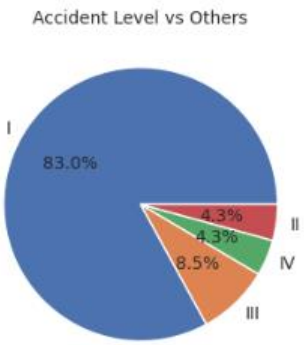
Accident Level/Industry Sector

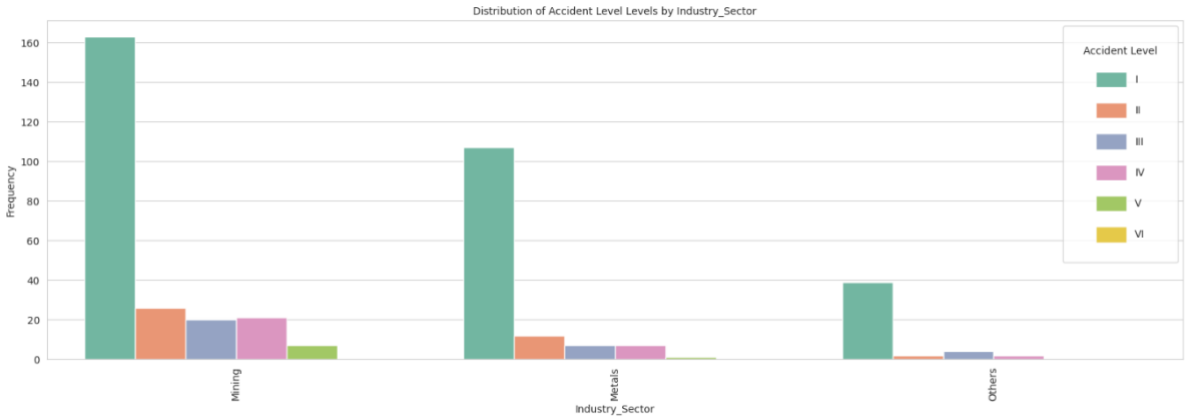
1. Mining sector has highest proportion of level 4 and level 5 accidents among all the sectors.

2. Proportion of level 1 accidents is highest in Others Sector.



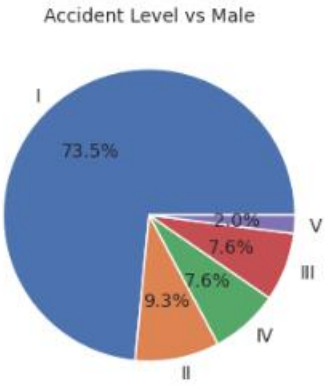


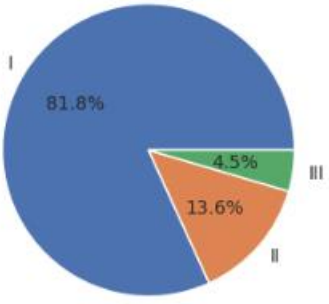


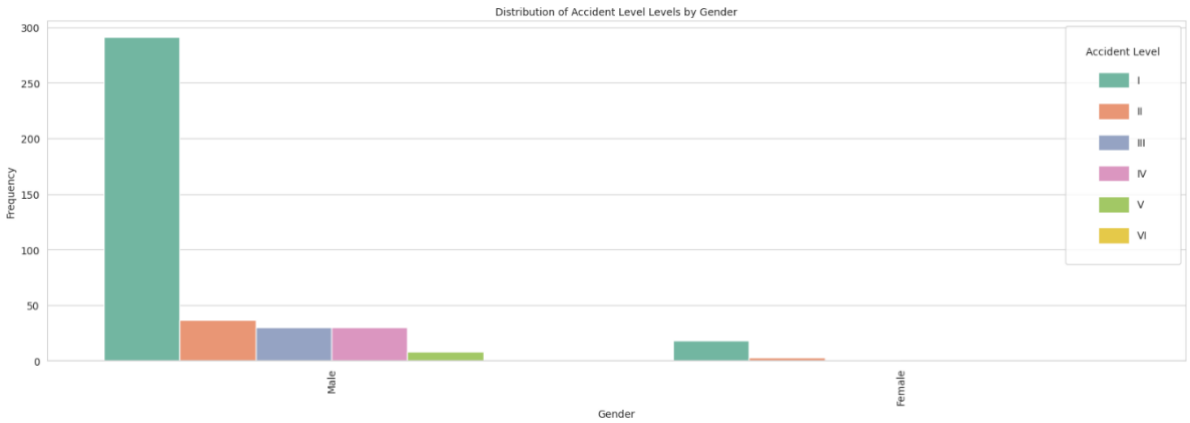


Accident Level/Gender

1. Men have faced more severe accidents than women.
2. Around 10% of overall accidents faced by men are level 4 and level 5 accidents, whereas women did not face any level 4 or 5 accident.
3. More than 80% of accidents faced by women are level 1 accidents.

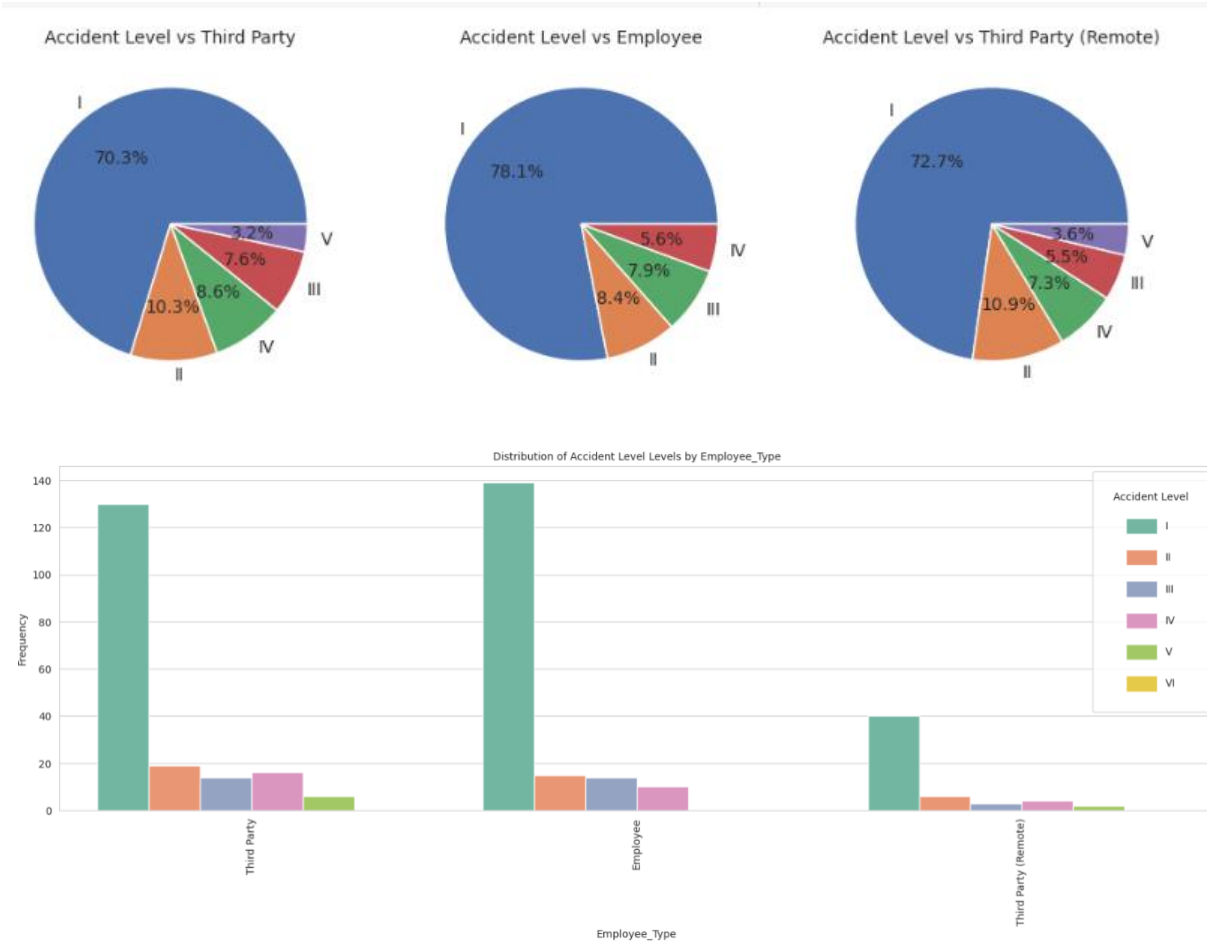






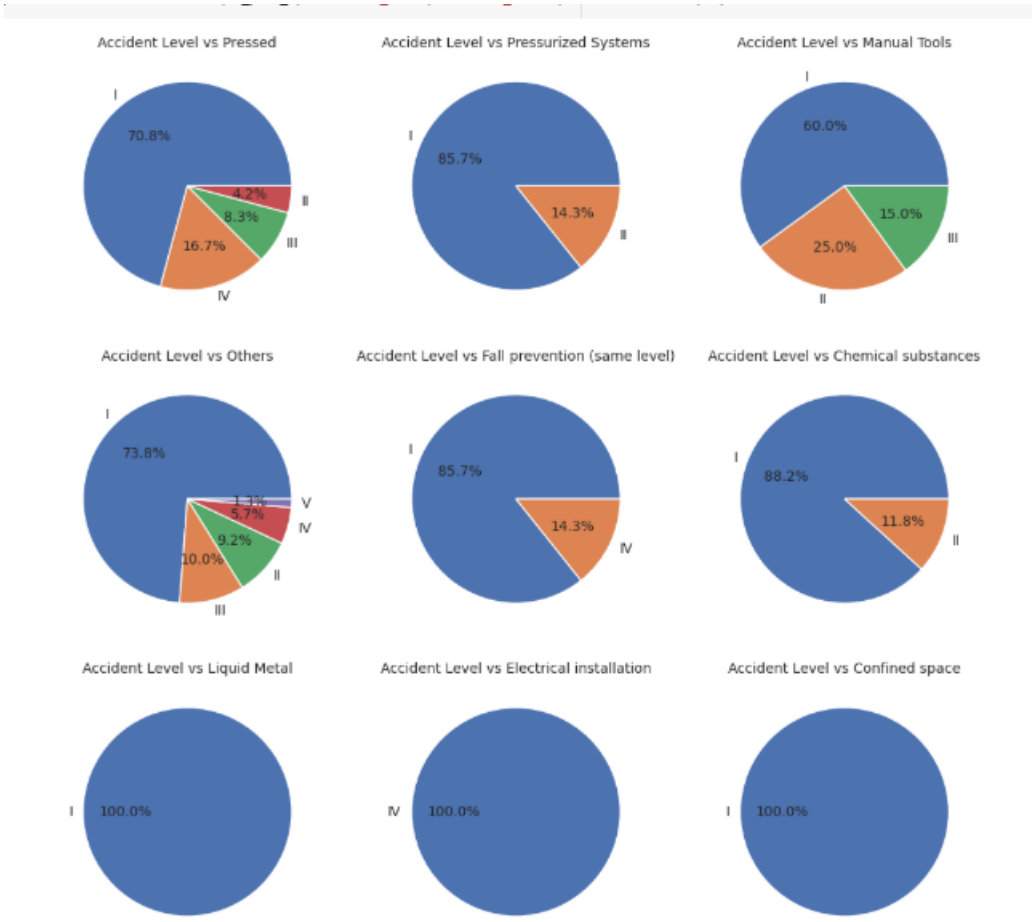
Accident Level/Employee Type

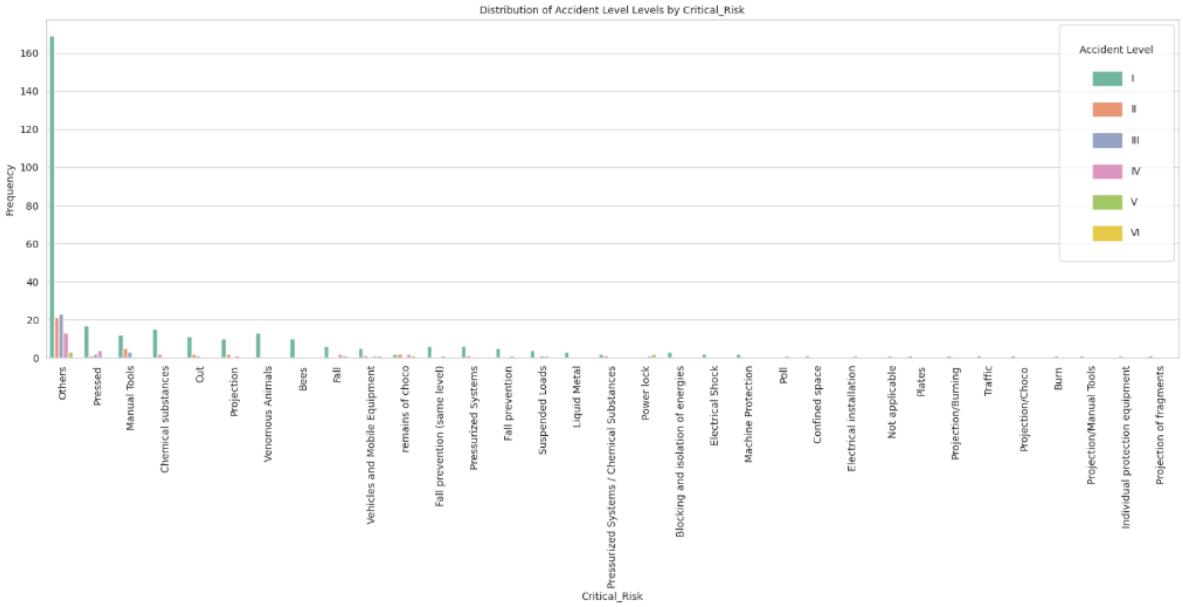
1. Level 1 accidents for all employee types is over 70%.
2. Internal employees did not face any level 5 accident, whereas proportion of level 5 accidents for both Third Party and Third Party remote is roughly the same.
3. This tells us that it is slightly more risky to be a Third party or remote employee than to be an internal employee.



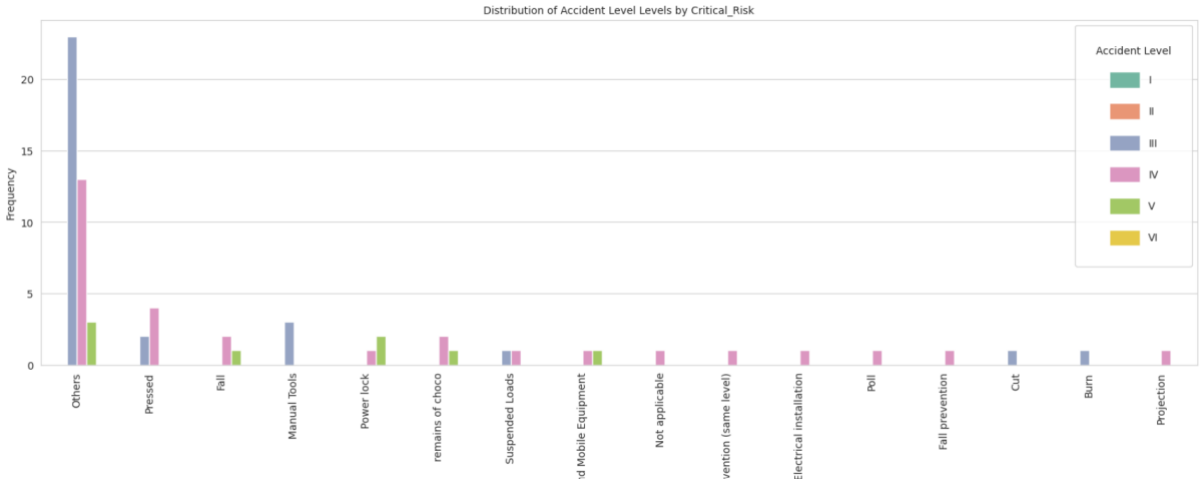
Accident Level/Critical Category

As there are several categories, displaying the pie-chart for a few of them below.





The most sever accidents for Levels 3,4 and 5 are for 16 critical Risk of Others, Pressed, falls, manual tools, power rack, remains of choco and others

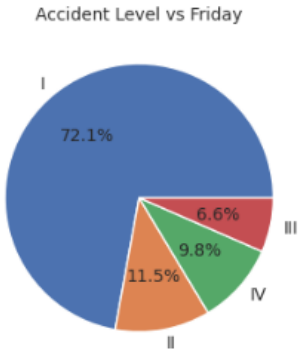


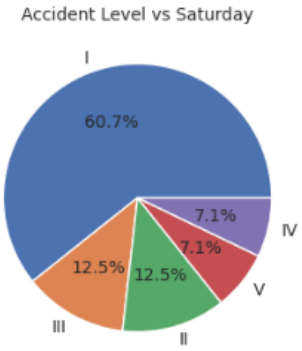
Accident Level/Month

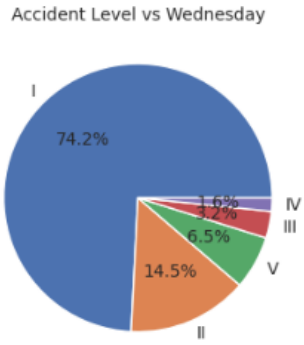
There have been severe accidents throughout the year. Severity 5 accidents all happened in May, June and July with levels 3 and 4 spread across the years. Every month there have been at-least 25-30% accidents with October being the most impacted month having almost 20% of accidents of level 3 and 4.

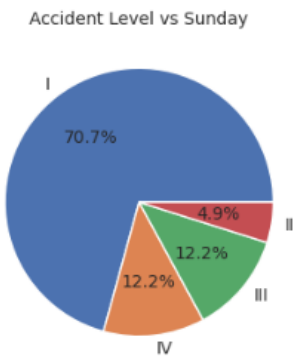
Accident Level/Weekday

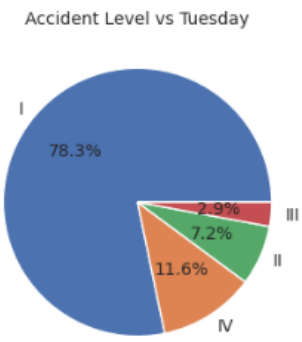
The ratio of severe accidents to the accidents that occur on a particular weekday is highest for Saturdays and Sundays. It is possible security measures are not being adhered to strictly on these days.

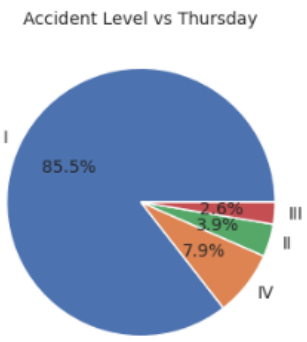

Accident Level vs Friday
I
72.1%
6.6%
III
9.8%
11.5%
IV
II

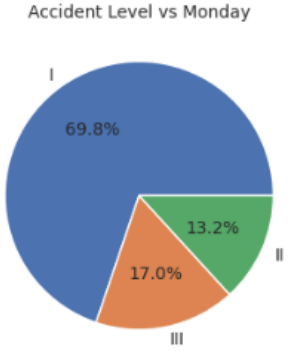

Accident Level vs Saturday
I
60.7%
7.1%
IV
7.1%
12.5%
12.5%
V
III
II

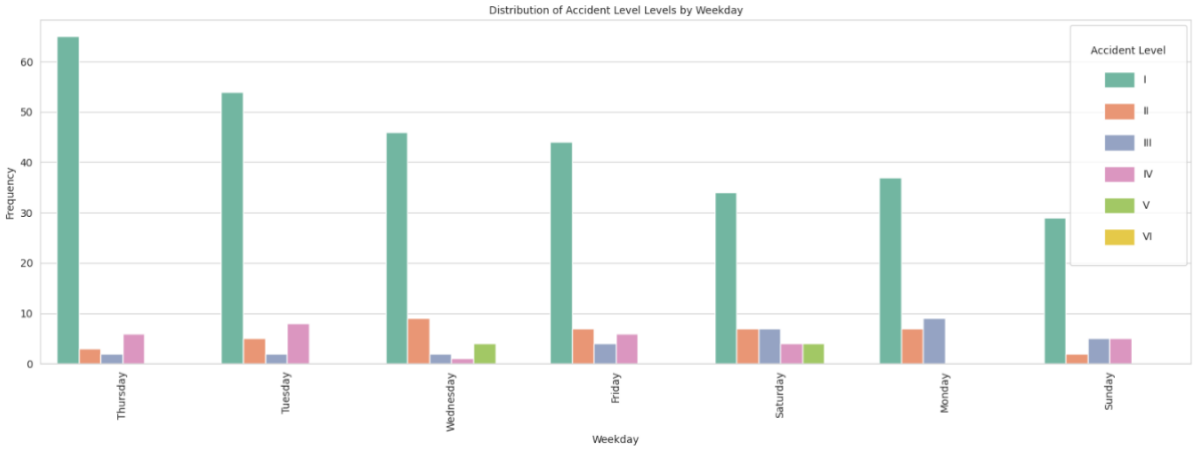

Accident Level vs Wednesday
I
74.2%
1.6%
IV
3.2%
III
6.5%
14.5%
V
II


Accident Level vs Sunday
I
70.7%
4.9%
II
12.2%
12.2%
III
IV


Accident Level vs Tuesday
I
78.3%
2.9%
III
7.2%
II
11.6%
IV


Accident Level vs Thursday
I
85.5%
2.6%
III
3.9%
II
7.9%
IV


Accident Level vs Monday
I
69.8%
13.2%
II
17.0%
III


Distribution of Accident Level Levels by Weekday
Accident Level
I
II
III
IV
V
VI
Frequency
0
10
20
30
40
50
60
Thursday
Tuesday
Wednesday
Friday
Saturday
Monday
Sunday
Weekday

With Potential Accident Levels

Accident Level/Country

Decide on the target column – analyse correlation between Accident Level/Potential accident level

The correlation between accident level and potential accident levels was analyzed to decide which column should be our target column.

Several techniques were used:-

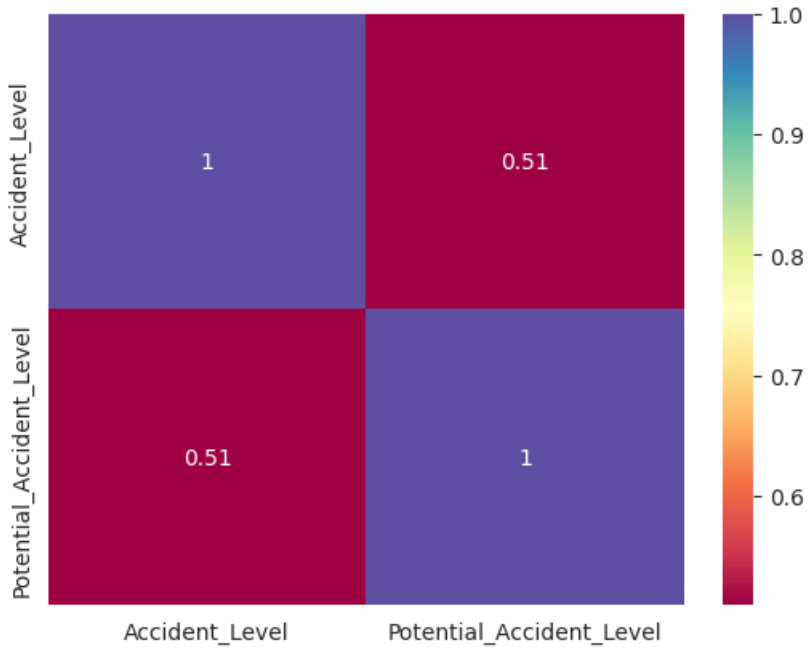
Pearson and Spearman coefficient

Between Accident Level and Potential Accident Level:

The Pearson correlation coefficient is 0.51

The Spearman correlation coefficient is 0.50

Heat Map



Cross tab

Though the potential levels are higher the actual accident levels are less severe.

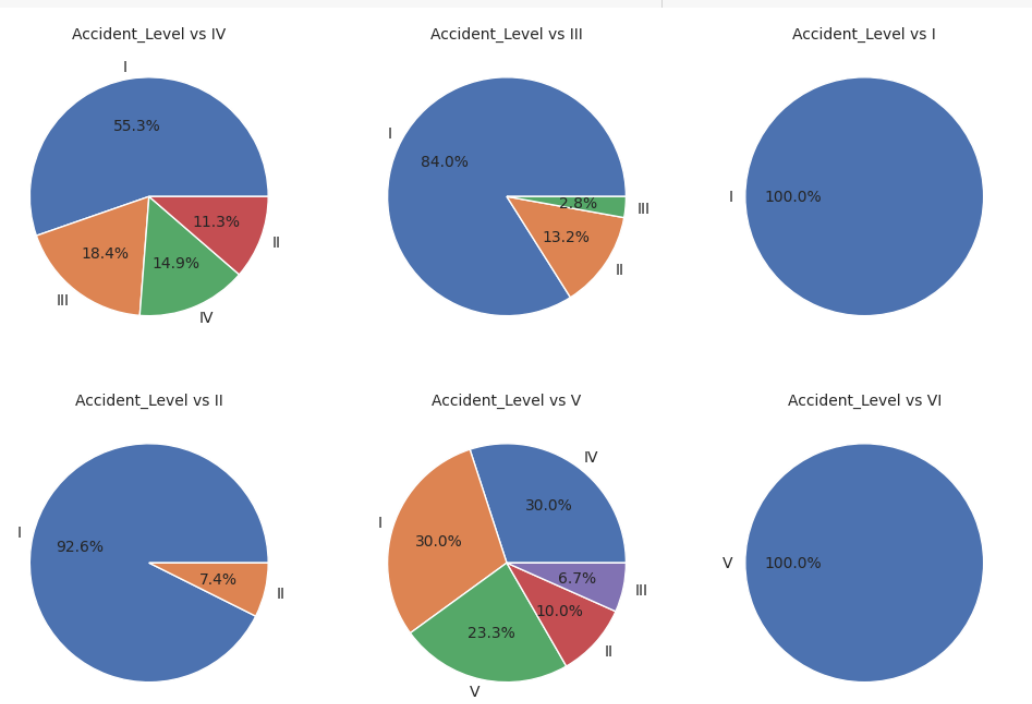
Potential_Accident_Level	1	2	3	4	5	6
Accident_Level						
1	45	88	89	78	9	0
2	0	7	14	16	3	0
3	0	0	3	26	2	0
4	0	0	0	21	9	0
5	0	0	0	0	7	1

Below graph show the percentage times when accident level was similar to what was the potential accident level for severity levels 3,4 and 5

Level V: Only 23.3% of the times accident level was of severity IV when the potential was also IV

Level IV: Only 14.9% of the times accident level was of severity IV when the potential was also IV

Level III: Only 2.8% of the times accident level was of severity IV when the potential was also IV



Observation and Inference:

- Both correlation coefficients (0.51 for Pearson and 0.50 for Spearman) suggest a ***moderate positive relationship*** between Accident_Level and Potential_Accident_Level. This indicates that higher levels of accidents are likely associated with higher potential levels of accidents.
- The similarity in values (0.51 for Pearson and 0.50 for Spearman) suggests that the relationship is reasonably ***consistent across both linear and monotonic assessments***. This consistency strengthens the reliability of the observed association.

Given the requirement for this project, we will take a pessimistic approach and warn the user of a higher potential accident that may be encountered. We have, therefore, considered the Potential Accident Level as the target variable.

Model Building

Data Preprocessing

The below data pre-processing and clean-up steps were performed:-

- As the number of severity VI potential accidents data is very small, the rows were merged as severity VI.
- Also, the Potential Accident Level has an inherent order representing increasing severity. Maintaining the numerical relationship is important for analysis, hence the use of **Ordinal Encoding.**
- Country and critical risk (critical risk is just a classification of the description column.)

For the Basic Machine Learning Model

For the training a basic machine learning model the below data preprocessing steps were also performed:-

- The categorical columns were encoded using several techniques based on the nature of the data in that feature
    - **Location**: Frequency encoding

      Locations are anonymized but numbered sequentially from Local_01 to Local_12

      Some locations have significantly higher accident rates than others and the distribution is highly skewed.

    - **Industry sector** : One Hot encoding as there are only 3 categories and have no inherent ordering..
        - **Employee Type**: Employee Type has only 3 categories and have no inherent ordering. Hence One hot encoding was used.
    - **Gender**: Binary Encoding as it has only 2 values.
    - **Month and Weekday**: For Month and Weekday we would proceed with **Cyclical Encoding** as they represent time data. December is as close to January as November, and Sunday is as close to Monday as Saturday. Cyclical encoding would preserve this cyclical nature of data

Data Preprocessing on the Description column (for NLP-based models)

Several text preprocessing were applied on the description column

- NER (Named Entity Recognition) processing

    - The **description** contains names of persons and places. A pre-trained model from Hugging Face was used to extract and replace the named entities from the text.
- Lowercase conversion
- Timestamp conversion
    - We observe quite a few mentions of time in various formats. e.g. "9:45 am", "14:16", "04:50 p.m.", etc. Let us replace all these with period of the day - morning, afternoon, evening or night.
- Remove Numbers and Special characters
- Removing unwanted spaces
- Create a local synonym library (specific for the Mining and Metal industry)
- Add POS Tagging
- Stop word removal
- Lemmatization
- Spell Checker and Correction
- Combining features as sentences

Training and Test Dataset

Independent and dependent variables have been created with below dimensions:

- Train set: 267 rows and 12 columns
- Validation set: 67 rows and 12 columns
- Test set: 84 rows and 12 columns
-

Data Up-sampling for minority classes

For the Basic Machine Learning Model

SMOTE was used to up-sample the records for all the severity levels to handle the class imbalance. After this, the train set has 445 rows (and 12 columns).

For the NLP models

The below techniques were used and the comparison was performed using data from both these techniques

Synonym replacement technique

Criteria for Model Evaluation

Several machine learning models were developed with **a primary focus on optimizing Recall scores, particularly for higher accident severity levels (3, 4, 5, and 6).**

Since the goal is to predict whether an accident will occur and assess its severity, the emphasis was placed on achieving accurate predictions for severe accidents. This prioritization reflects the **critical importance of identifying high-severity incidents, even if it means tolerating lower precision scores for lower-severity predictions**.

The rationale for this approach is that failing to predict a severe accident or predicting a lower severity, has far graver consequences than overestimating the likelihood or severity of an incident. By focusing on higher Recall scores, the models aim to minimize the risk of undetected severe accidents while providing actionable insights for safety professionals.

Models Considered

The below models have been considered:

Traditional Classification ML Models (without NLP)

Random Forests – This model works well for structured data representations like TF-IDF or word embeddings.

We used this model with various inputs to get the best recall accuracy. All the models were designed and trained in a Random Forest Classifier model.

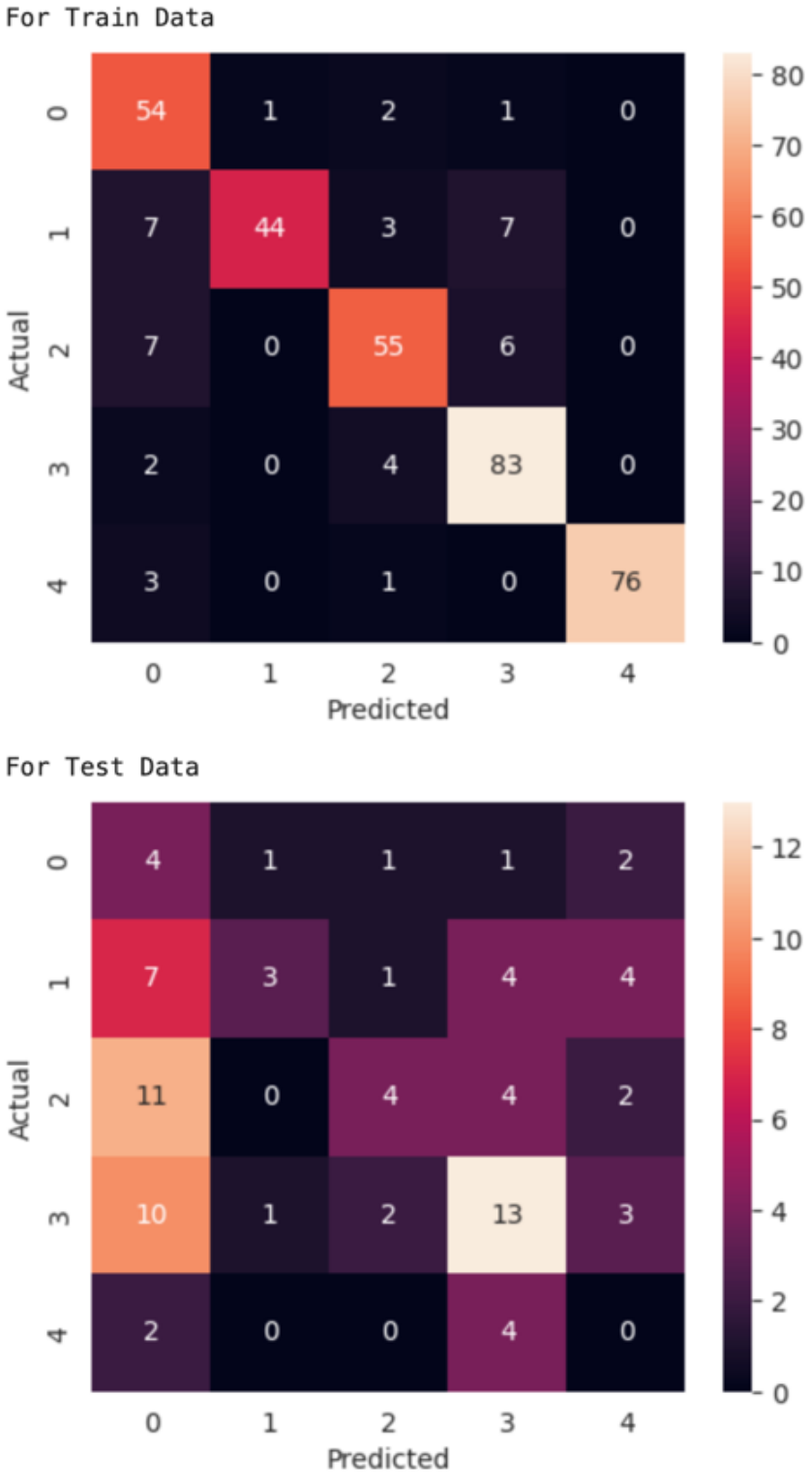
1. **Model 1 (Base Model (without Description))** - A base model was built using a Random Forest Classifier with all the relevant features from the received dataset, except for the description
2. **Model 2 [Base Model with best Features (extracted from Model 1)]:** The second version was created using only the most important features returned from the base random forest model. We have limited it to 10 top-most important features
3. **Model 3 [Base Model with Hyper Tuning]:** Then the base model was hyper-tuned using RandomSearchCV and GridSearchCV.

Traditional Classification ML Models (with NLP)

We then used several techniques to create embedding and then built a random forest. The below embedding techniques were used:-

a. Word2Vec (vector size - 300)
b. Glove (vector size – 100)
c. Sentence Transformer (using 'sentence-transformers/all-MiniLM-L6-v2' model from hugging face, which has a vector size of 384)

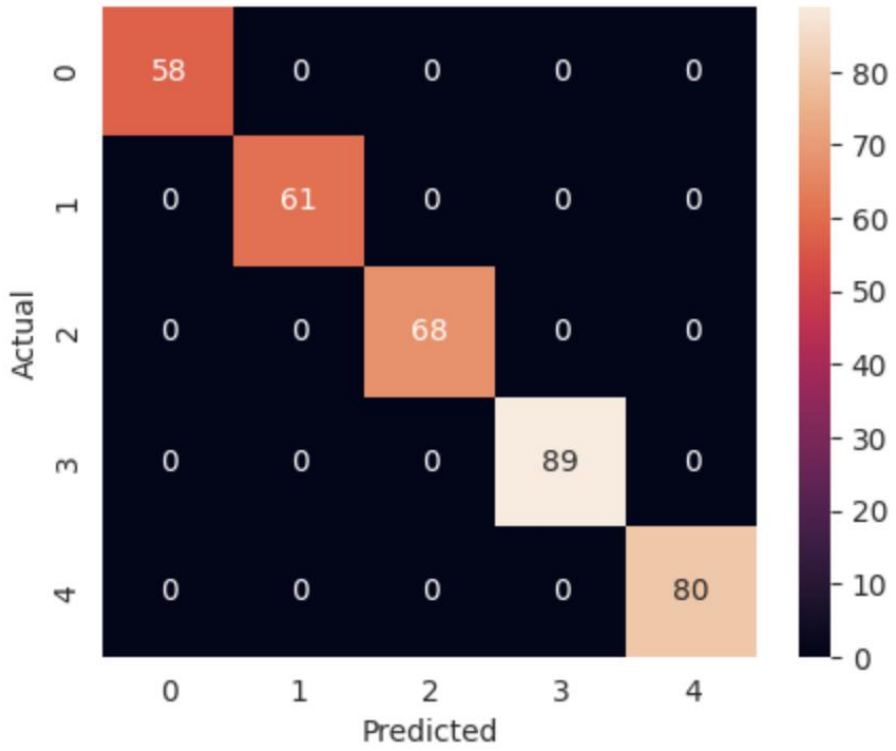
4. **Model 4 [With Word2Vec embeddings]:** a base Random Forest classifier model was built using the word2Vec embeddings



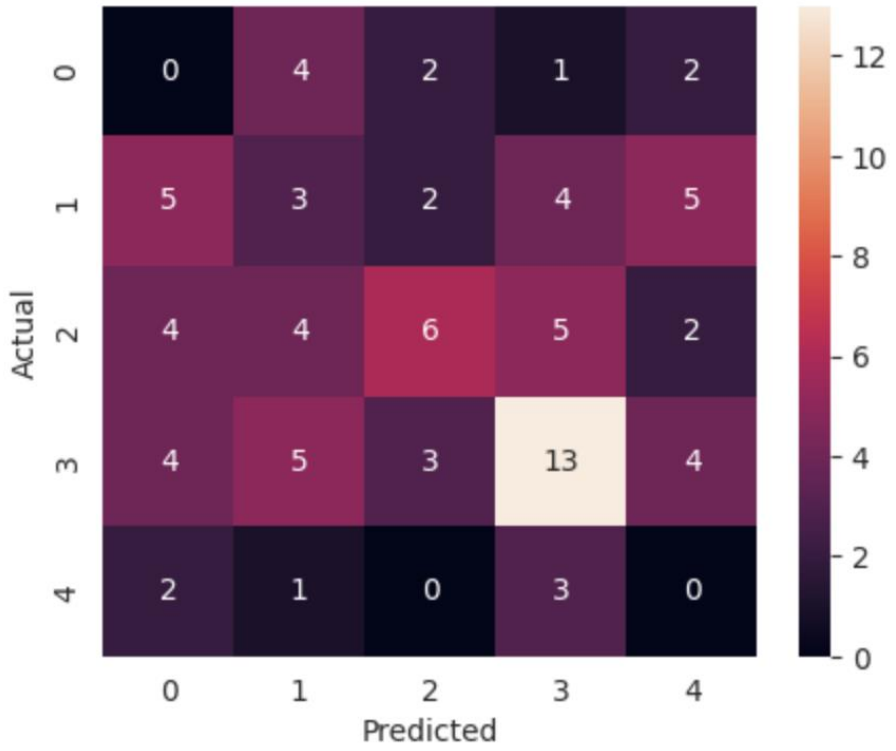
Confusion Matrix for Word2Vec Base Model

5. **Model 5 [Word2Vec and class balancing]:** Random Forest classifier model with word2Vec em beddings and class balancing

For Train Data

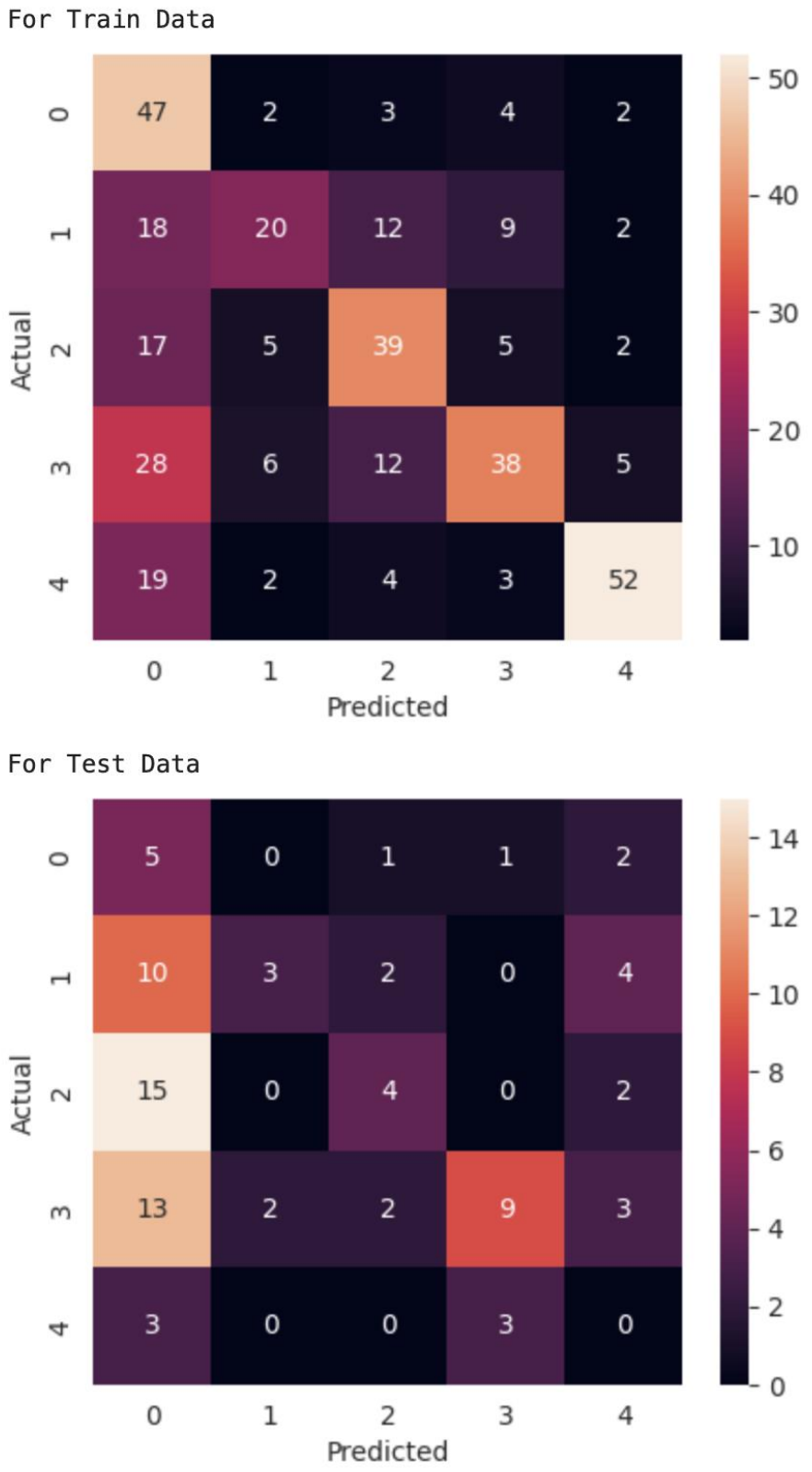


For Test Data



Confusion Matrix for Word2Vec Model with class balancing

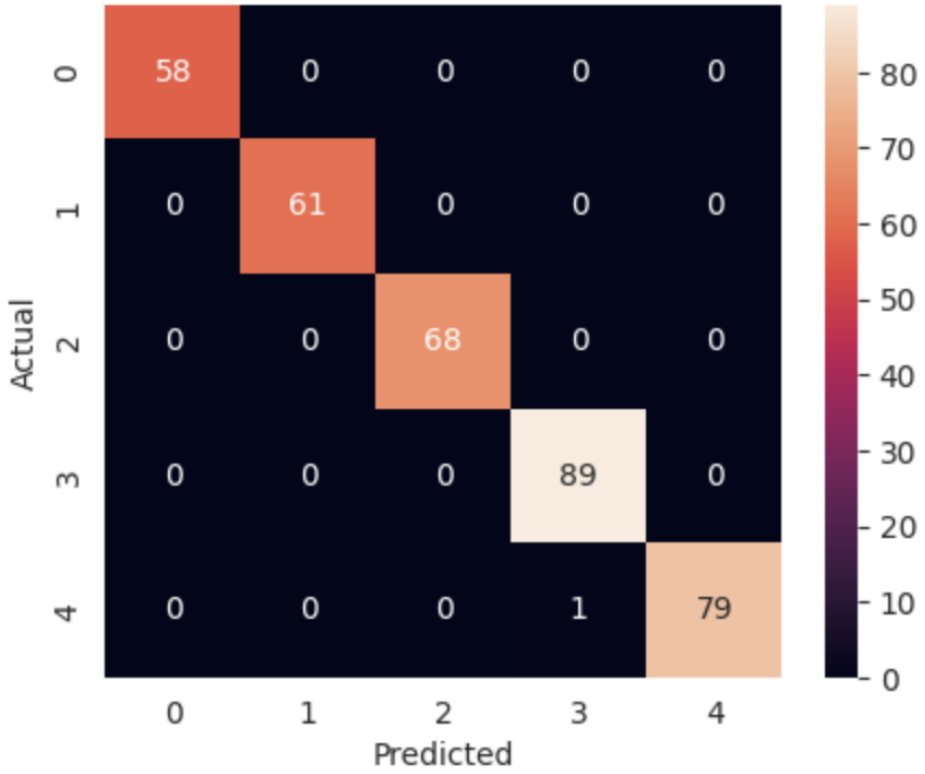
6. **Model 6 [Tuned Model with Word2Vec and class balancing]:** Hyper-tuned word2Vec Model with class balancing.



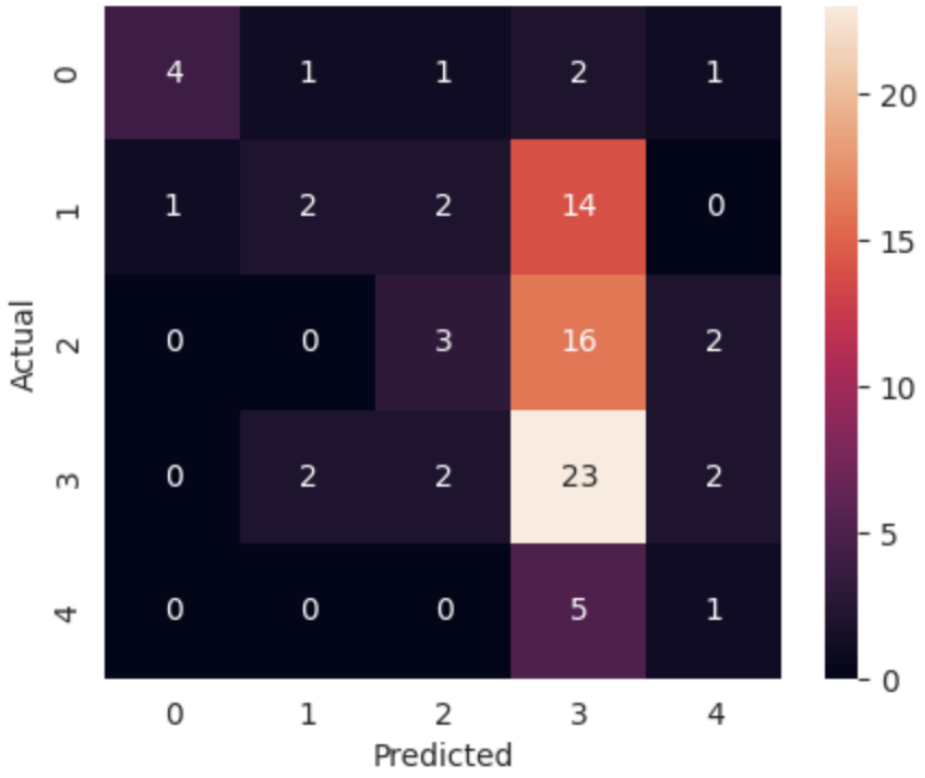
Confusion Matrix for Tuned Word2Vec Model with class balancing

7. **Model 7 [With GloVe embeddings]:** a base Random Forest classifier model was built using the **GloVe** embeddings

For Train Data



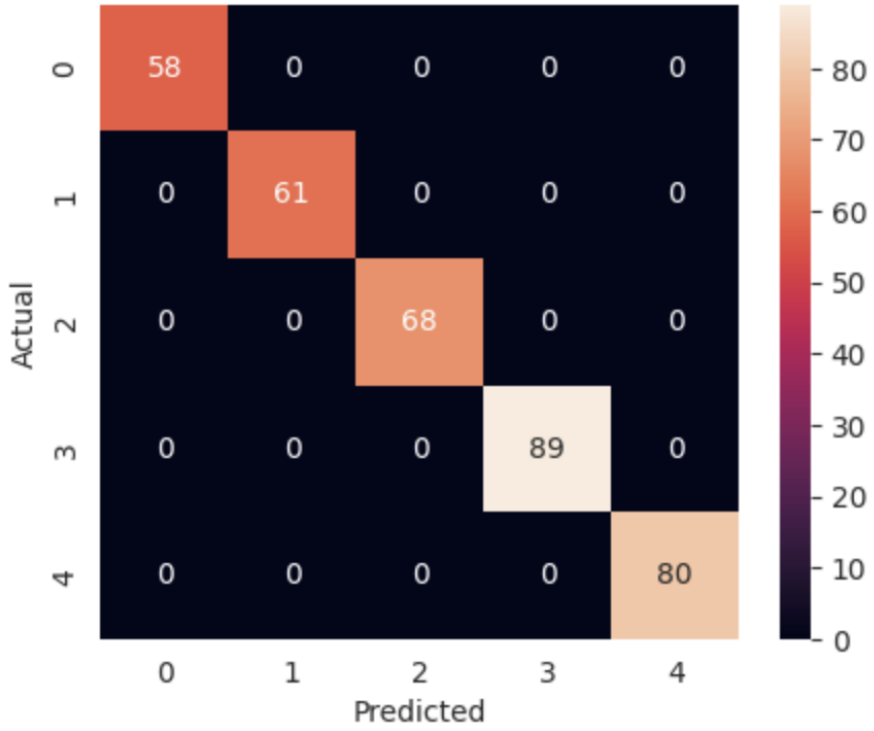
For Test Data



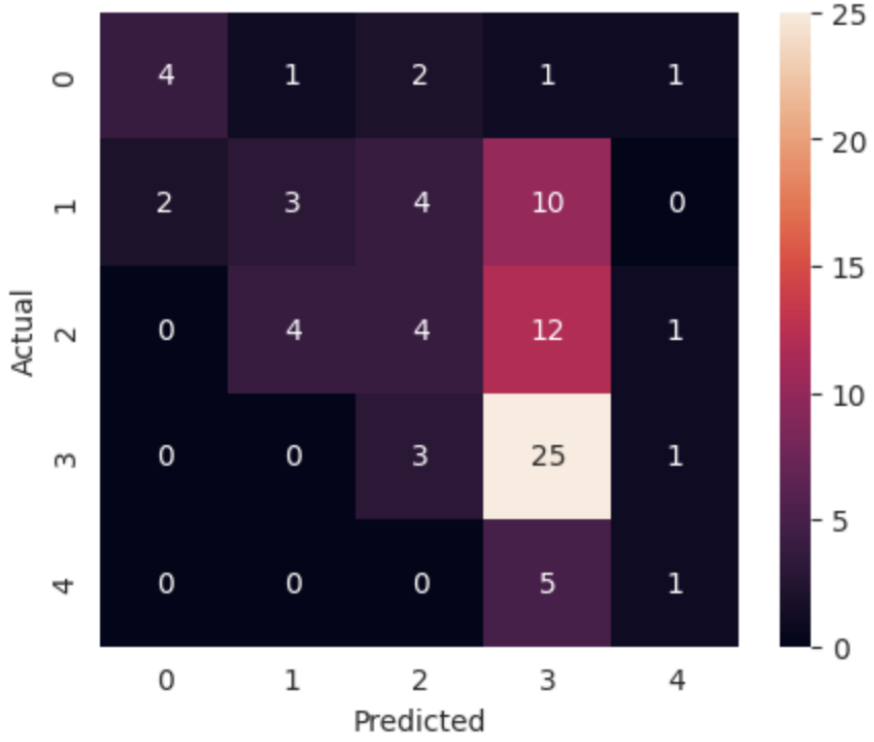
Confusion Matrix for Glove Base Model

8. **Model 8 [GloVe and class balancing]:** Random Forest classifier model with **GloVe** embeddings and class balancing

For Train Data



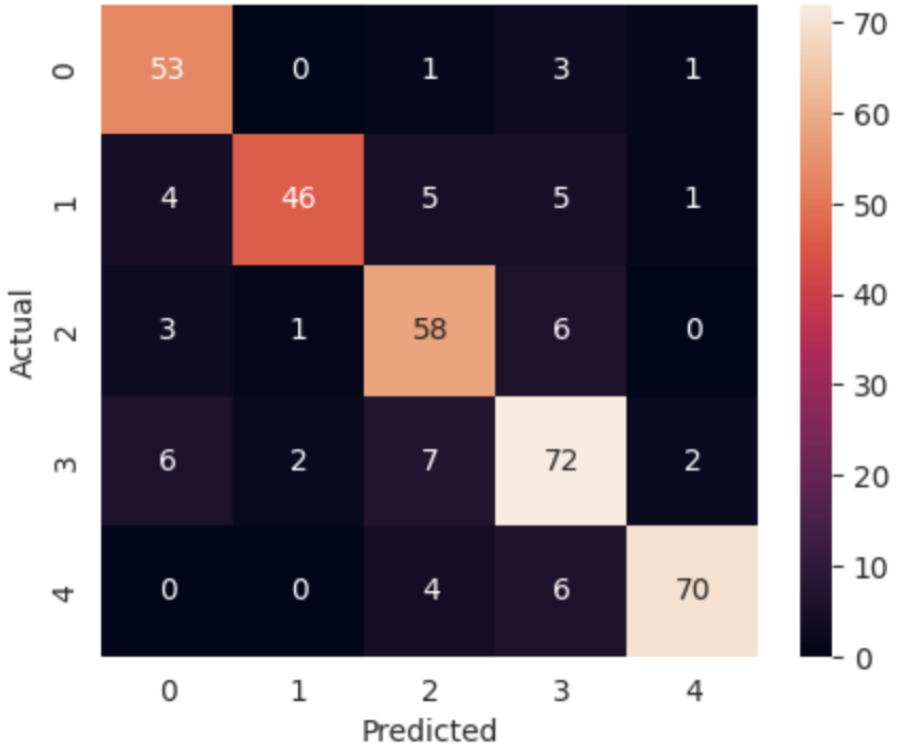
For Test Data



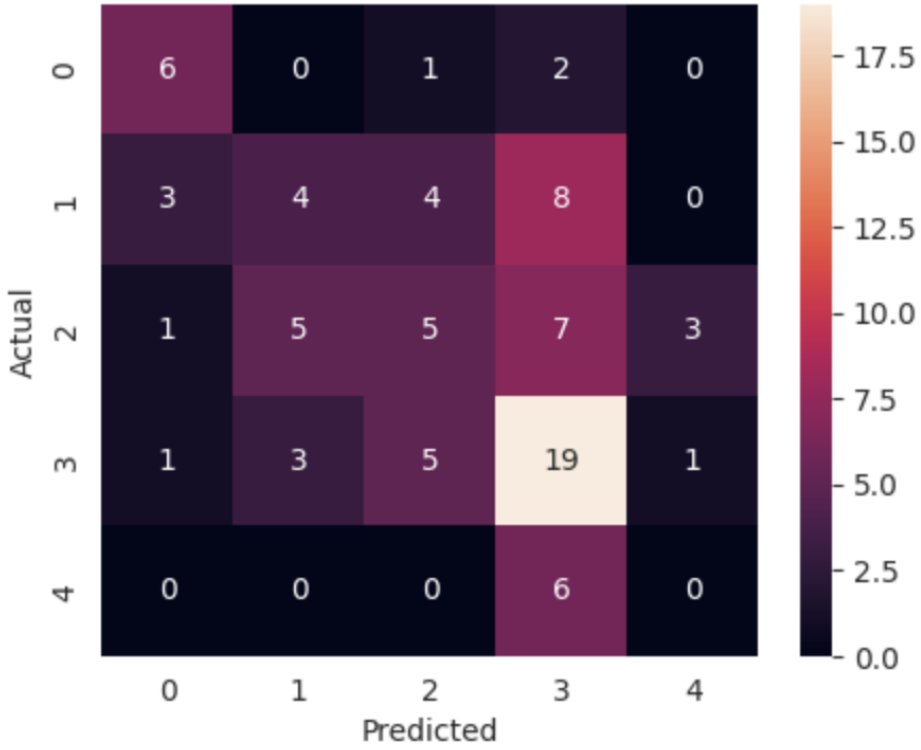
Confusion Matrix for Glove Model with class balancing

9. **Model 9 [Tuned Model with Word2Vec and class balancing]:** Hyper-tuned **GloVe** Model with class balancing

For Train Data



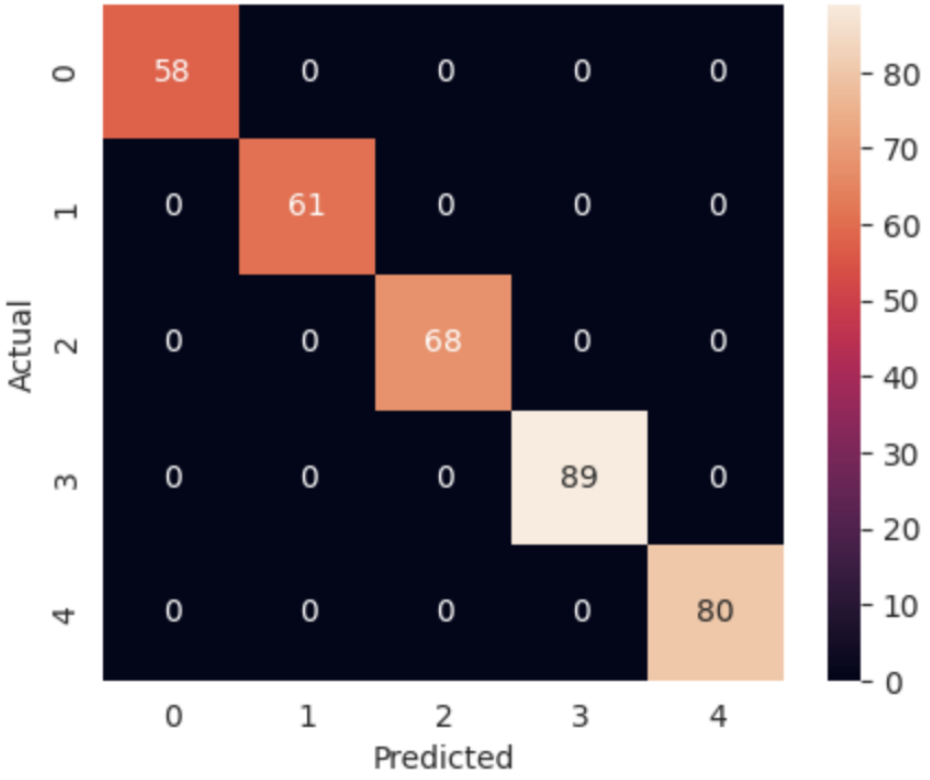
For Test Data



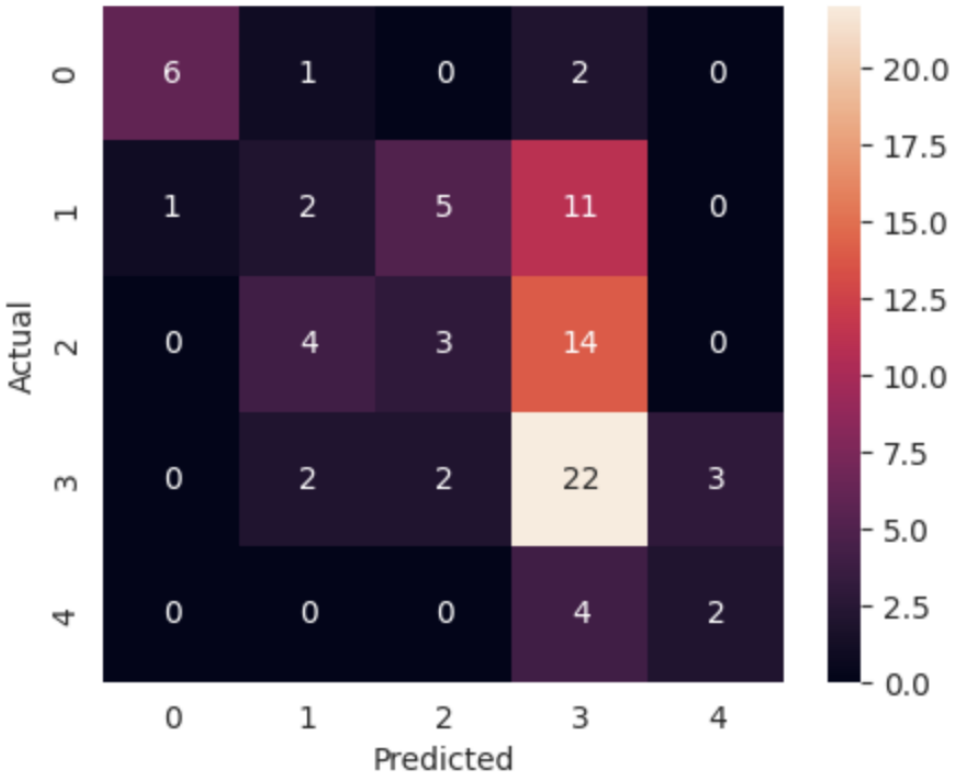
Confusion Matrix for Tuned Glove Model with class balancing

10. **Model 10 [With transformer embeddings]:** a base Random Forest classifier model was built using the **transformer** embeddings

For Train Data



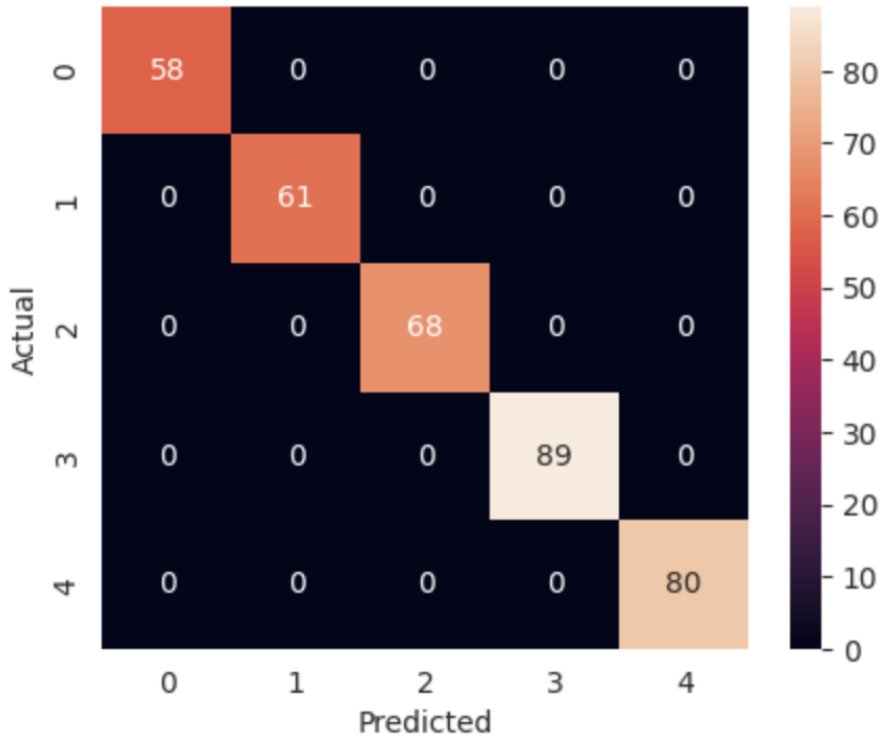
For Test Data



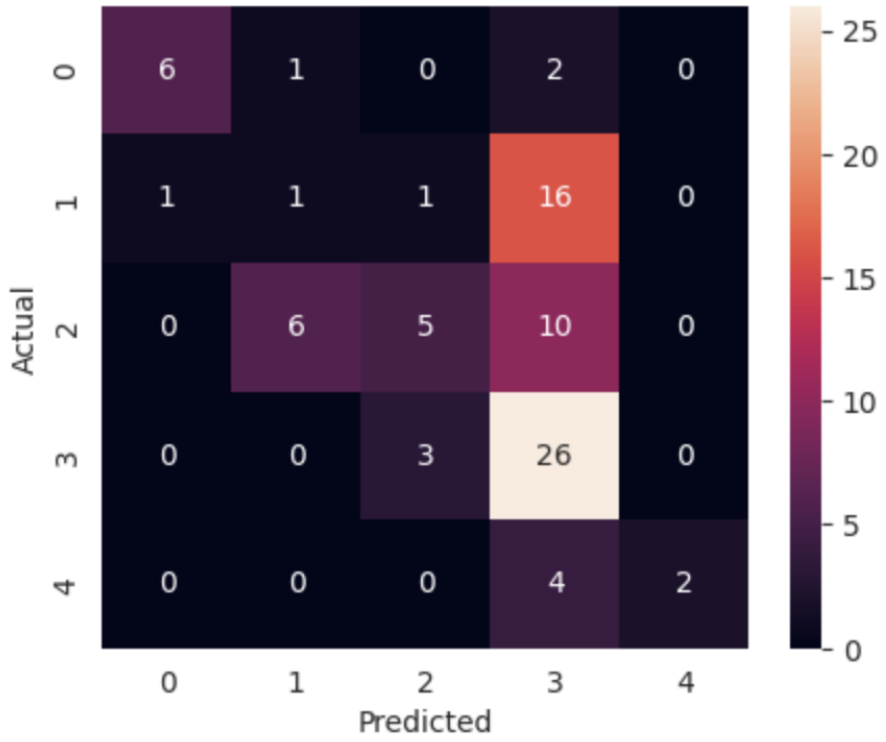
Confusion Matrix for Base Model with Transformer embeddings

11. **Model 11 [transformer and class balancing]:** Random Forest classifier model with **transformer** embeddings and class balancing

For Train Data



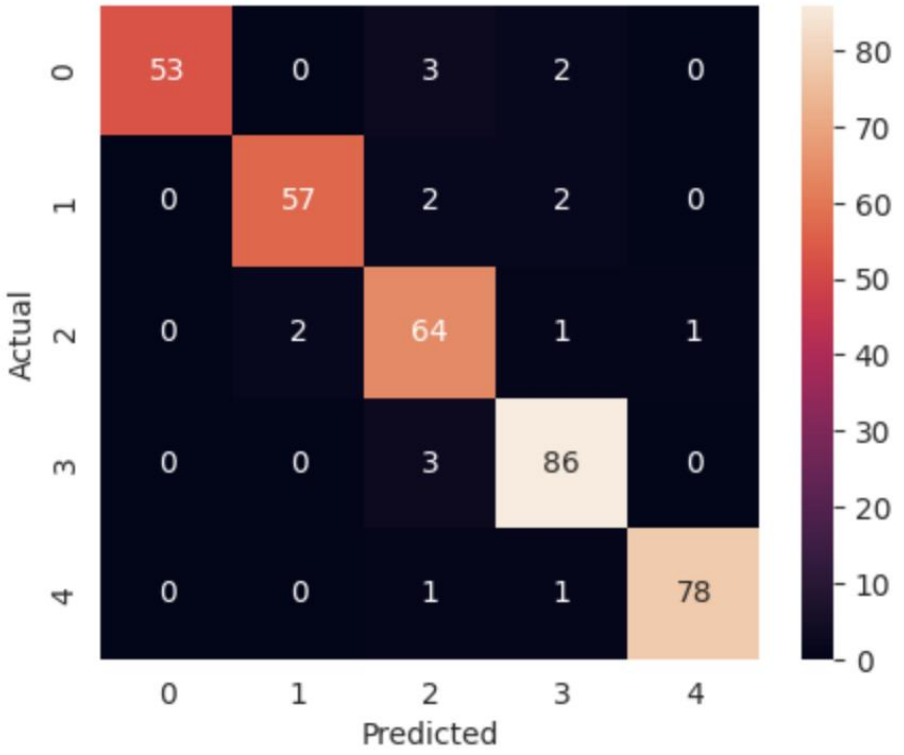
For Test Data



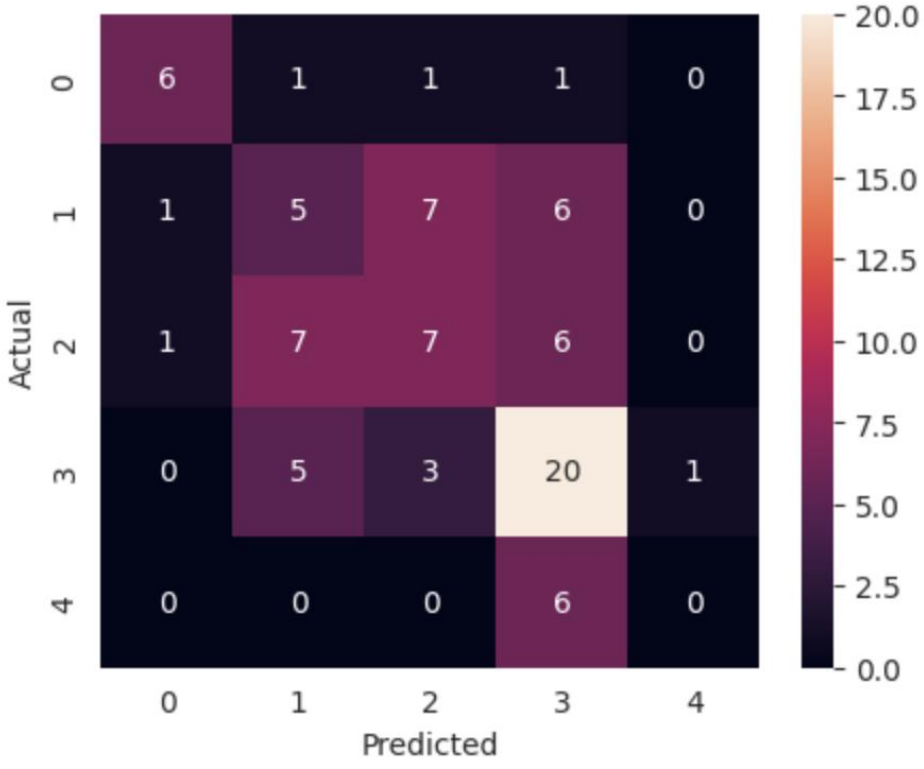
Confusion Matrix for Base Model with Transformer embeddings and class balancing

12. **Model 12 [Tuned Model with transformer and class balancing]:** Hyper-tuned **transformer** Model with class balancing

For Train Data



For Test Data



Confusion Matrix for Tuned Model with Transformer embeddings and class balancing

Model Performances

Comparing recall scores and accuracies:

We compared the models using their recall scores for each accident level and weighted F1 scores as shown in the following table [Recalls for severity level 5 (level 6 was added to level 5), etc.]

Model	Recall 5	Recall 4	Recall 3	Recall 2	Recall 1	Weighted F1-Score
Base Model (without Description)	0	0.2	0	0	0.8	0.36
Base Model with best features (extracted from Model 1)	0	0.17	0.17	0.12	0.82	0.63
Base Model with Hyper Tuning	0	0.2	0	0.17	0.78	0.6
Base Model with Word2Vec	0	0.45	0.19	0.16	0.44	0.28
Base Model with Word2Vec and class balancing	0	0.45	0.29	0.16	0	0.26
Tuned Model with Word2Vec and class balancing	0	0.31	0.19	0.16	0.56	0.25
Base Model with GloVe	0.17	0.79	0.14	0.11	0.44	0.39
Base Model with GloVe and class balancing	0.17	0.86	0.19	0.16	0.44	0.44
Tuned Model with GloVe and class balancing	0	0.66	0.24	0.21	0.67	0.4
Base Model with Transformer	0.33	0.76	0.14	0.11	0.67	0.41
Base Model with Transformer and class balancing	0.33	0.9	0.24	0.05	0.67	0.47
Tuned Model with Transformer and class balancing	0	0.69	0.33	0.26	0.67	0.45

Best Performing Traditional ML Classification Model

Based on the criteria of recall scores, and the criteria of minimizing the underprediction of severity, we can select the best performing traditional ML classifier among all the classifiers built as the Random Forest class balanced model that has been training on Transformer embeddings as it's able to correctly predict more of the sever potential accidents than any of the other models.

Use advanced Neural Network Classification Models

In Milestone-2, we have considered more advanced models, which are anticipated to perform better than the traditional ML models.

Some of these models can be:

- Artificial Neural Networks (ANN)
- Recurrent Neural Network (RNN)
- Long-Short Term Memory (LSTM)
- Large Language Model (LLM)

Use LLM for data augmentation

Since the data is limited, we have used LLMs to augment the existing cleaned dataset. We will use the Gemini Frontier Model (driven by API) to add more synthetic data. The below approach is followed:-

1. Created data-preprocessing / cleaning pipeline which includes removing extra spaces, removing special characters and numbers, converting numeric time to time periods like morning, etc. This pipeline will be used on all synthetic data created using Gemini.
2. Join a few descriptions randomly for a particular combination of features (potential accident level, industry, weekday, month, etc.) to give as few shot prompts to the model.
3. Use the most frequent words from the Word Cloud that were plotted again the potential accident levels and also industry-wise in the prompt

We were able to generate below additional synthetic dataset:

Target Class	Rows added
1	67
2	36
3	28
4	0
5	75

Embeddings for Advanced Neural Network Classification Models

We have used the same sentence transformer 'sentence-transformers/all-MiniLM-L6-v2' model from the hugging face for all advanced neural network classification models, which has a vector size of 384

Artificial Neural Networks (ANN) Classifiers

Base ANN Model Classifier

We have used a neural network model with 3 layers: an input layer with 128 neurons, a hidden layer with 64 neurons, and an output layer with 5 neurons.

The model uses ReLU activation, dropout regularization, and L2 regularization to prevent overfitting.

The model is compiled with the Adam optimizer and categorical cross-entropy loss and uses early stopping with a patience of 10 epochs.

Layer (type)	Output Shape	Param #
dense_35 (Dense)	(None, 128)	49,280
dropout_22 (Dropout)	(None, 128)	0
dense_36 (Dense)	(None, 64)	8,256
dense_37 (Dense)	(None, 5)	325

Total params: 57,861 (226.02 KB)
Trainable params: 57,861 (226.02 KB)
Non-trainable params: 0 (0.00 B)

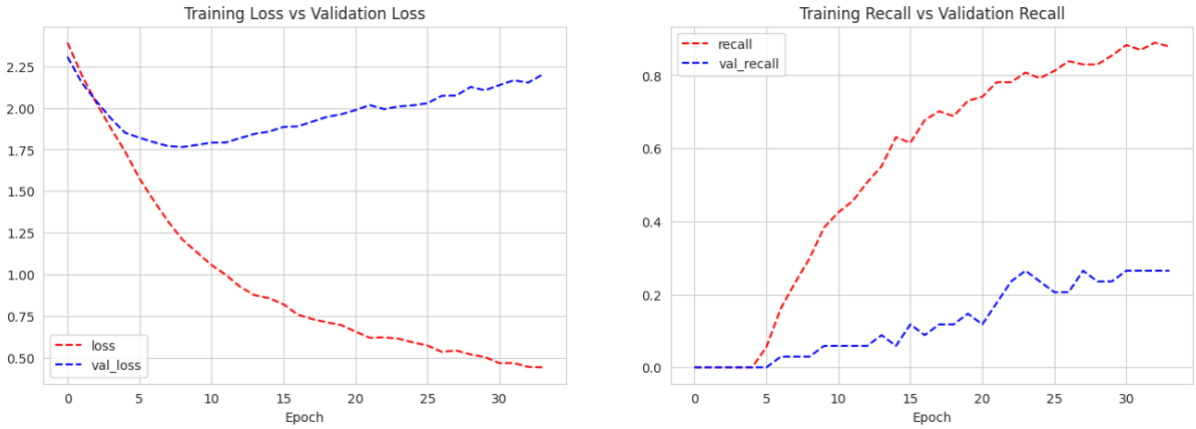


Figure: Training vs Validation Plots for Base ANN Model

As suggested by the graphs above, the training loss and validation loss are both decreasing steadily, indicating that the model is learning effectively. Training recall and validation recall are both increasing, suggesting that the model's ability to identify positive instances is improving.

However, there is a slight gap between training and validation recall, which might suggest some overfitting. Techniques to reduce overfitting, such as early stopping or regularization, could benefit the model.

Overall, the model appears to be performing well, but further analysis and potential adjustments to prevent overfitting could lead to even better results.

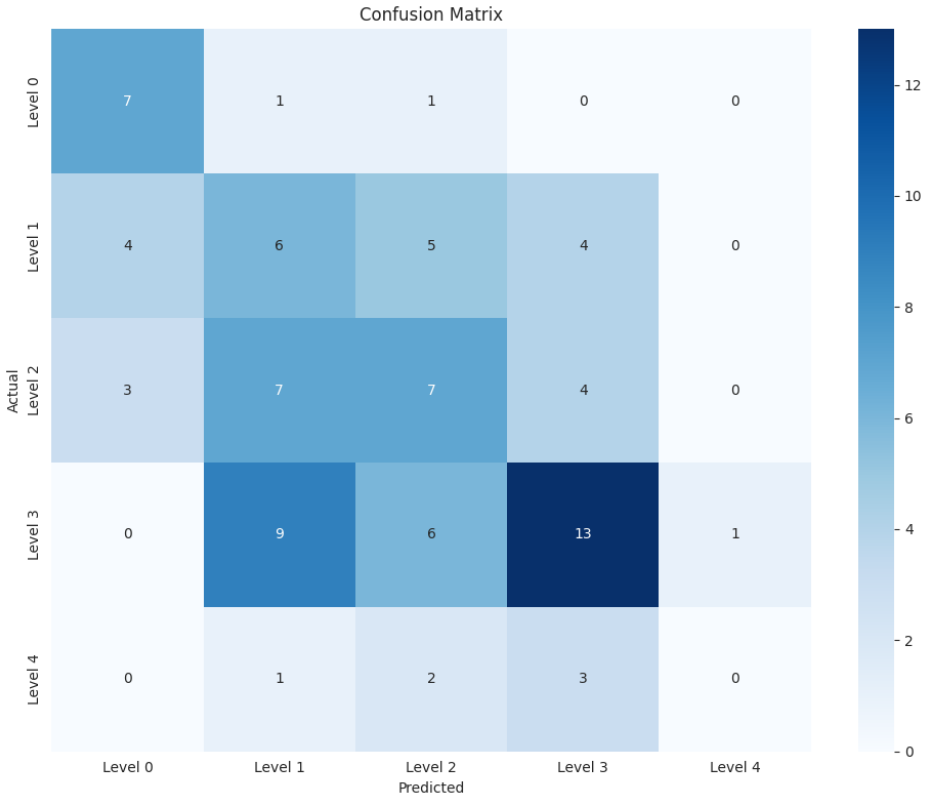


Figure: Confusion Matrix for Base ANN Model

Updated ANN Model Classifier

We have created an updated neural network model with 3 layers: 2 hidden layers with 16 and 8 neurons, and an output layer with 5 neurons.

The model uses batch normalization, ReLU activation, and dropout regularization to prevent overfitting.

The model is compiled with the Adam optimizer (learning rate 10^{-3}) and categorical cross-entropy loss, and tracks accuracy and recall metrics.

Layer (type)	Output Shape	Param #
dense_29 (Dense)	(None, 16)	6,160
batch_normalization_16 (BatchNormalization)	(None, 16)	64
activation_4 (Activation)	(None, 16)	0
dropout_19 (Dropout)	(None, 16)	0
dense_30 (Dense)	(None, 8)	136
batch_normalization_17 (BatchNormalization)	(None, 8)	32
dropout_20 (Dropout)	(None, 8)	0
dense_31 (Dense)	(None, 5)	45

Total params: 6,437 (25.14 KB)
Trainable params: 6,389 (24.96 KB)
Non-trainable params: 48 (192.00 B)

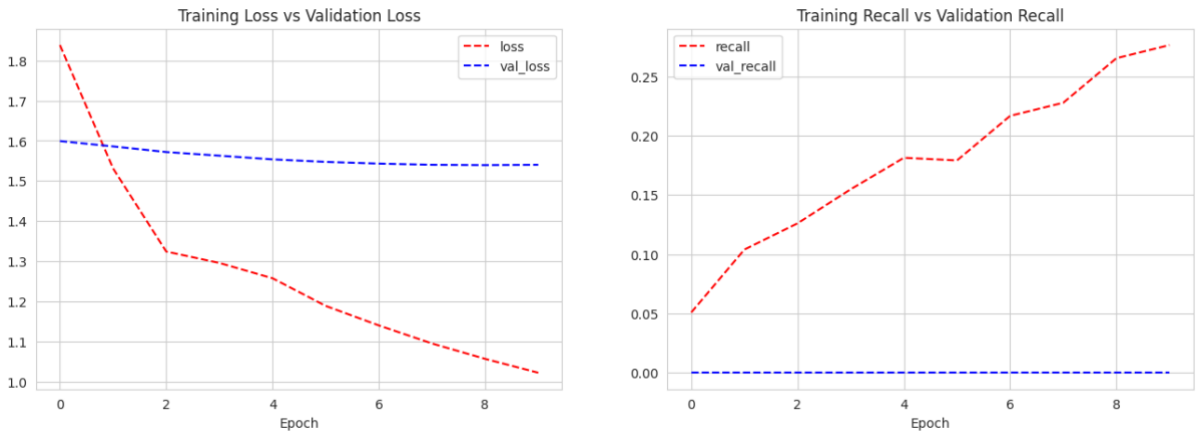


Figure: Training vs Validation Plots for Updated ANN Model

Both training loss and validation loss are decreasing steadily, indicating that the model is learning effectively. However, there is a significant gap between training and validation loss, which suggests that the model is overfitting.

Training recall is increasing, but validation recall remains very low and flat, further confirming the overfitting issue. The model is likely memorizing the training data but failing to generalize to new, unseen data.

To address this, techniques like early stopping, regularization, or increasing the size of the training dataset could be implemented to improve the model's performance on unseen data.

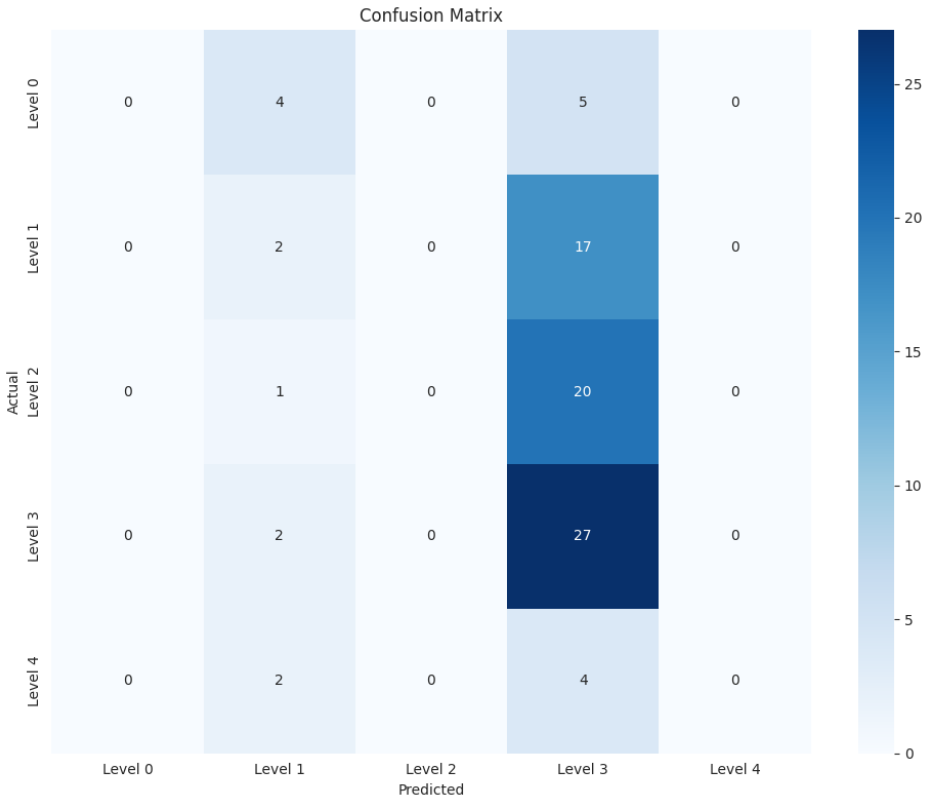


Figure: Confusion Matrix for Updated ANN Model

Long Short Term Memory Recurrent Neural Network Model (LSTM RNN) Classifiers

Base LSTM Model Classifier

We have defined a Long Short-Term Memory (LSTM) neural network model with 2 LSTM layers and 2 dense layers. The model uses dropout regularization to prevent overfitting.

The model is compiled with the Adam optimizer and categorical cross-entropy loss, and tracks accuracy and recall metrics.

Layer (type)	Output Shape	Param #
lstm (LSTM)	(None, 128)	262,656
dropout (Dropout)	(None, 128)	0
dense (Dense)	(None, 64)	8,256
dropout_1 (Dropout)	(None, 64)	0
dense_1 (Dense)	(None, 5)	325

Total params: 271,237 (1.03 MB)
Trainable params: 271,237 (1.03 MB)
Non-trainable params: 0 (0.00 B)

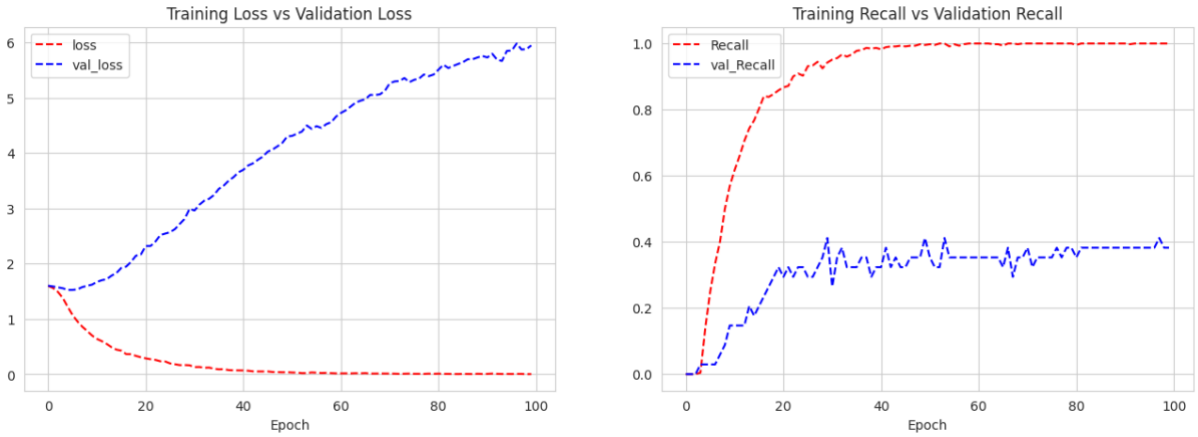


Figure: Training vs Validation Plots for Base LSTM Model

Training loss has decreased significantly, while validation loss has stabilized after a slight initial decrease. This suggests the model is learning but may be overfitting to the training data.

Training recall has increased steadily and reach almost 1, while validation recall has plateaued after an initial rise to 0.39. This further indicates overfitting, as the model is performing well on training data but not generalizing to new data.

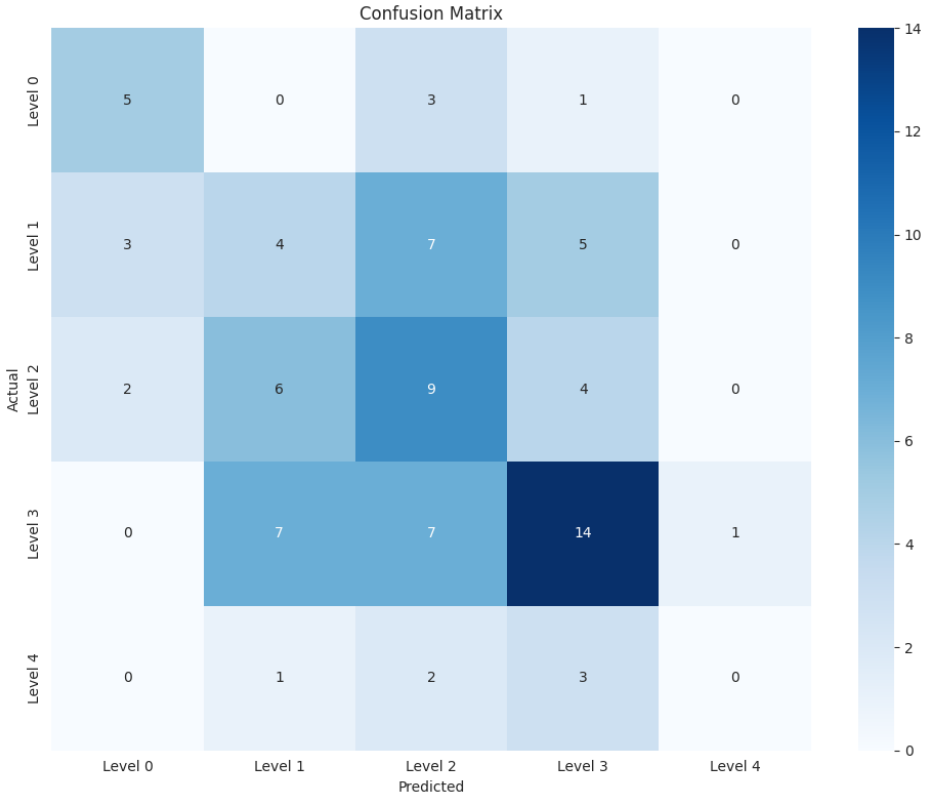


Figure: Confusion Matrix for Base LSTM Model

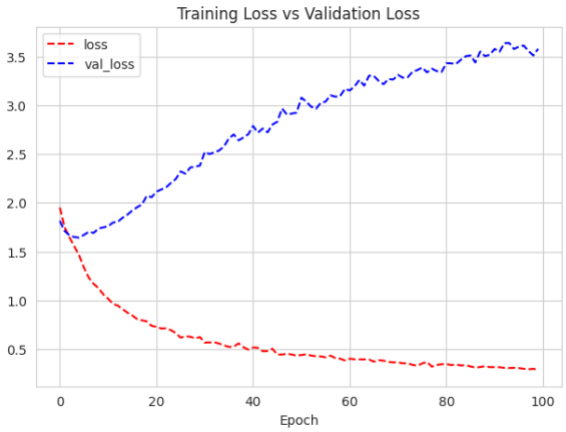
We will try to simplify the last dense layer now and see if the results get any better

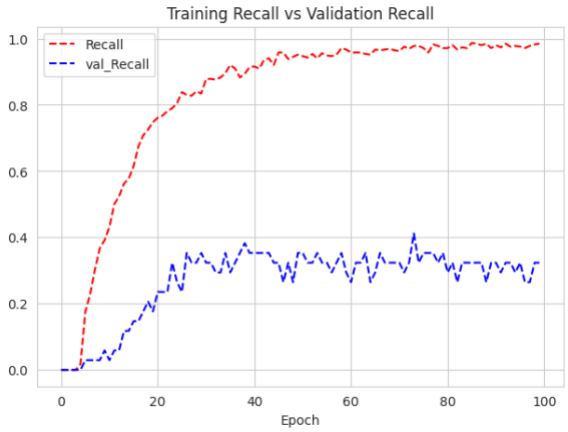
Updated LSTM Model Classifier – Variant A

Here, we have reduced the number of neurons in the first dense layer to 32, aiming to simplify the model and enhance its ability to generalize.

Layer (type)	Output Shape	Param #
lstm_1 (LSTM)	(None, 128)	262,656
dropout_2 (Dropout)	(None, 128)	0
dense_2 (Dense)	(None, 32)	4,128
dropout_3 (Dropout)	(None, 32)	0
dense_3 (Dense)	(None, 5)	165

Total params: 266,949 (1.02 MB)
Trainable params: 266,949 (1.02 MB)
Non-trainable params: 0 (0.00 B)





Observation

- Still very high Training Recall and very low Validation Recall.
- Validation loss decreased to some extent but still high (~3.5)
- The model is still overfitting

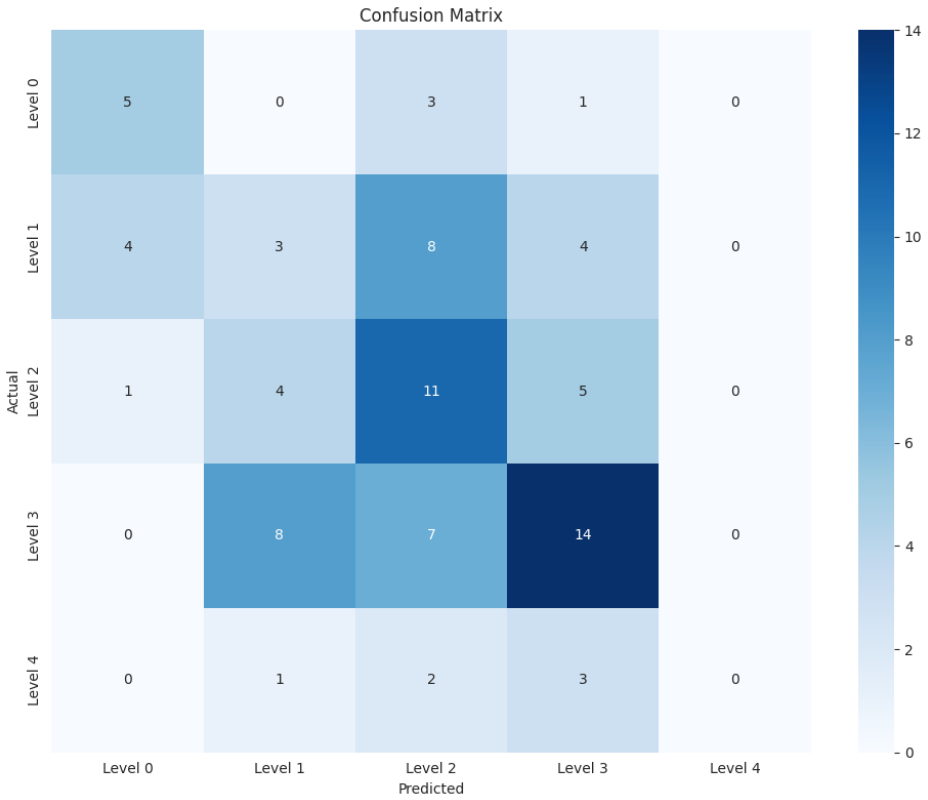


Figure: Confusion Matrix for Updated LSTM Model – Variant A

Updated LSTM Model Classifier – Variant B

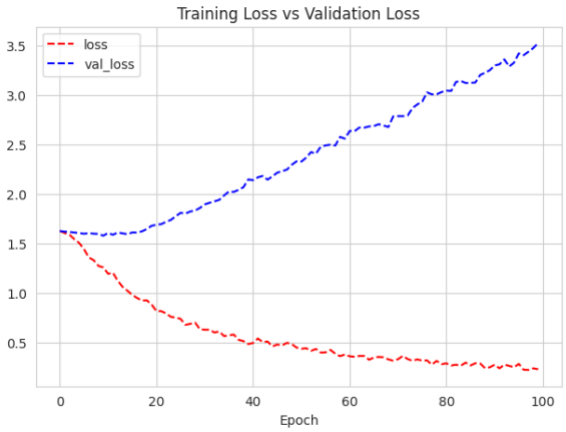
Several key adjustments were made to refine the model further. The number of LSTM units was decreased from 128 to 32, which may enhance computational efficiency and mitigate overfitting.

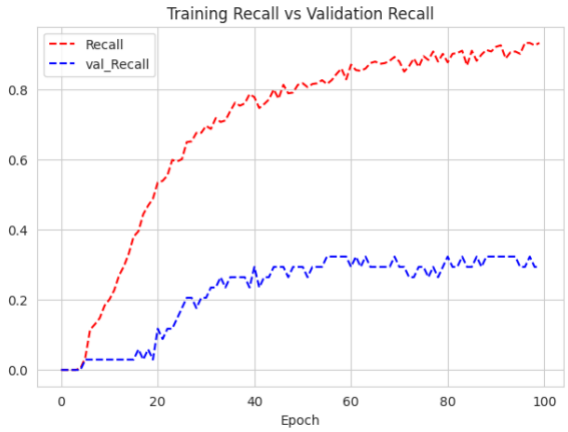
Additionally, the strength of L2 regularization was reduced, allowing the model to have larger weights and potentially capture more complex patterns.

Furthermore, the Adam optimizer was configured with an explicit learning rate of 0.001, providing more control over the training process.

Layer (type)	Output Shape	Param #
lstm_5 (LSTM)	(None, 32)	53,376
dropout_10 (Dropout)	(None, 32)	0
dense_10 (Dense)	(None, 16)	528
dropout_11 (Dropout)	(None, 16)	0
dense_11 (Dense)	(None, 5)	85

Total params: 53,989 (210.89 KB)
Trainable params: 53,989 (210.89 KB)
Non-trainable params: 0 (0.00 B)





Observation:

- Loss decreasing, overfitting likely.
- Recall increasing, validation plateaus.
- Model overfits, needs further regularization.

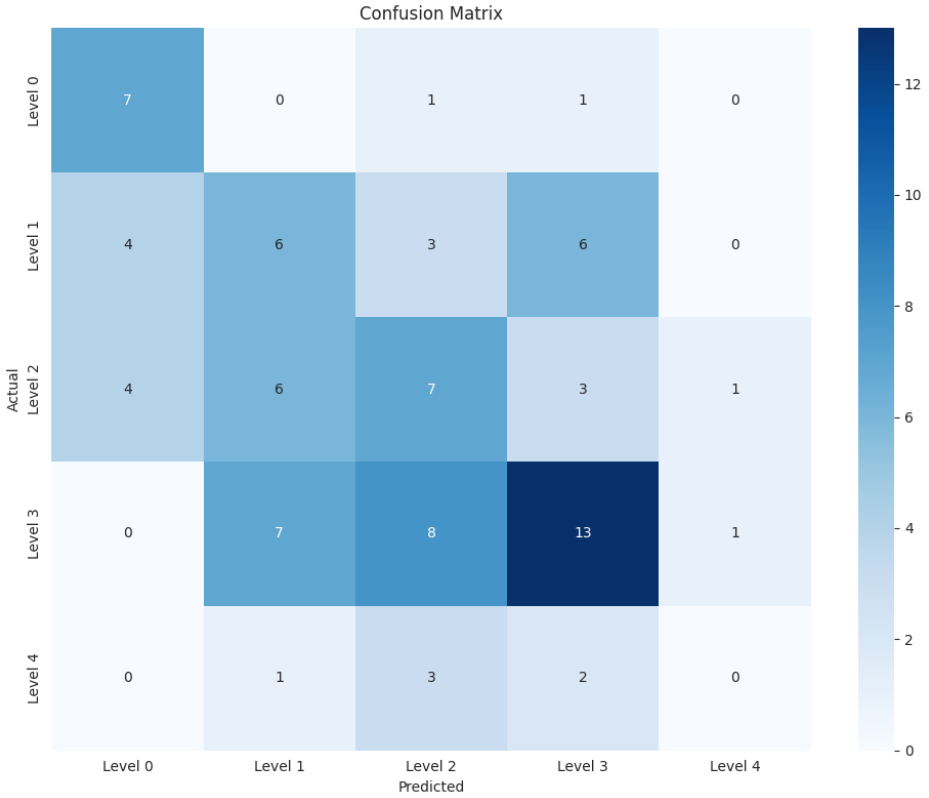


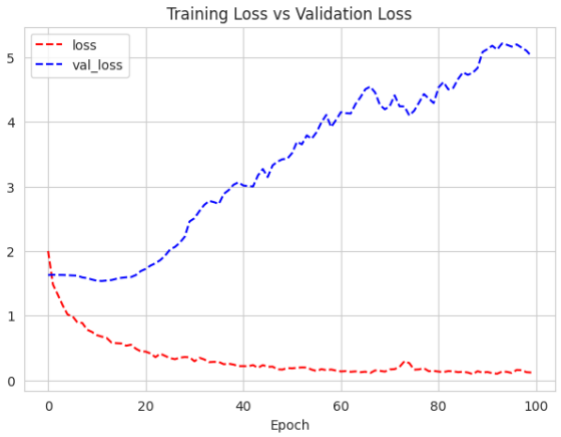
Figure: Confusion Matrix for Updated LSTM Model – Variant B

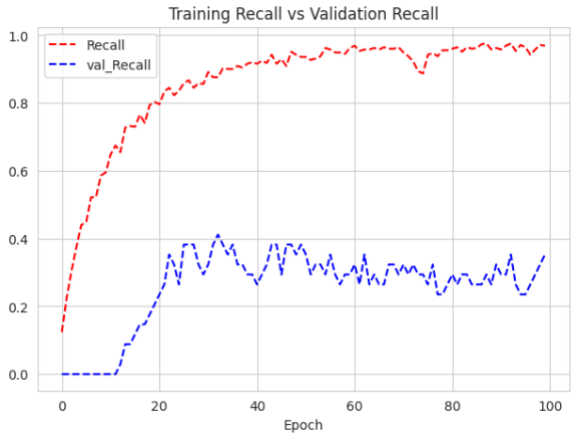
Updated LSTM Model Classifier – Variant C

We have added additional dropout layers to further prevent overfitting

Layer (type)	Output Shape	Param #
lstm_6 (LSTM)	(None, 32)	53,376
batch_normalization (BatchNormalization)	(None, 32)	128
dropout_12 (Dropout)	(None, 32)	0
dense_12 (Dense)	(None, 16)	528
batch_normalization_1 (BatchNormalization)	(None, 16)	64
dropout_13 (Dropout)	(None, 16)	0
dense_13 (Dense)	(None, 5)	85

Total params: 54,181 (211.64 KB)
Trainable params: 54,085 (211.27 KB)
Non-trainable params: 96 (384.00 B)





Observations:

- Training loss decreases steadily, but validation loss plateaus, indicating potential overfitting.
- Training recall increases, while validation recall plateaus or fluctuates, suggesting the model is learning on training data but not generalizing well to unseen data.

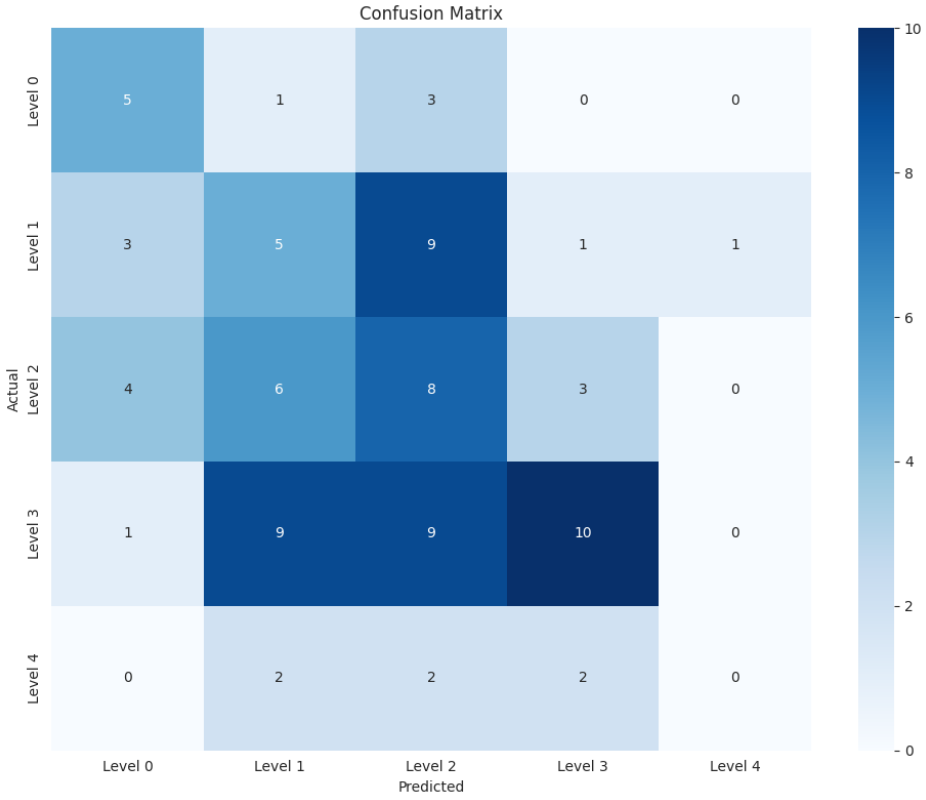


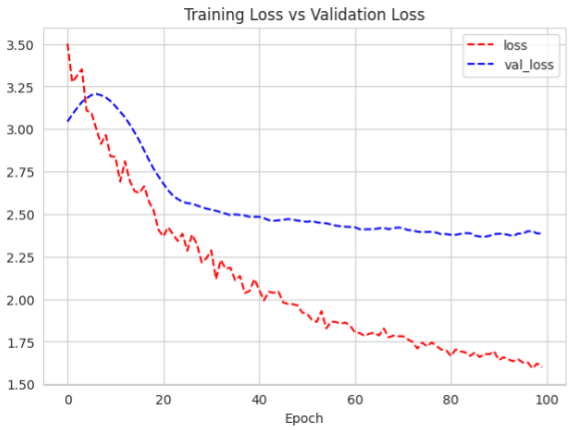
Figure: Confusion Matrix for Updated LSTM Model – Variant C

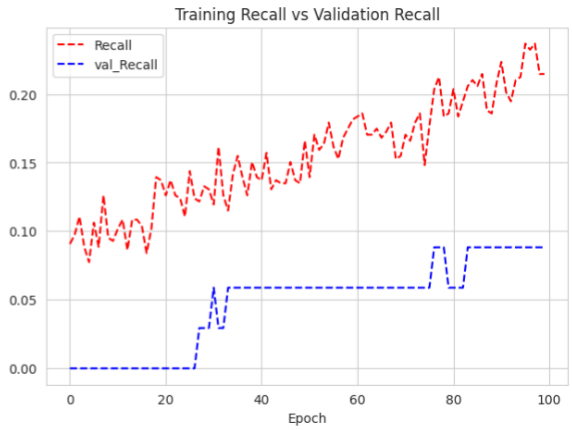
Updated LSTM Model Classifier – Variant D

Here, LSTM and dense layers are further simplified, with additional layers for batch normalization

Layer (type)	Output Shape	Param #
lstm_34 (LSTM)	(None, 1, 16)	25,664
batch_normalization_46 (BatchNormalization)	(None, 1, 16)	64
lstm_35 (LSTM)	(None, 8)	800
batch_normalization_47 (BatchNormalization)	(None, 8)	32
dropout_42 (Dropout)	(None, 8)	0
dense_42 (Dense)	(None, 8)	72
batch_normalization_48 (BatchNormalization)	(None, 8)	32
dropout_43 (Dropout)	(None, 8)	0
dense_43 (Dense)	(None, 5)	45

Total params: 26,709 (104.33 KB)
Trainable params: 26,645 (104.08 KB)
Non-trainable params: 64 (256.00 B)





Observations:

- The model is learning effectively, with both training and validation loss decreasing steadily.
- Training and validation recall are showing overall improvement, indicating the model's ability to identify positive instances.
- The plot suggests better generalization and reduced overfitting, although some instability is evident, requiring further tuning.

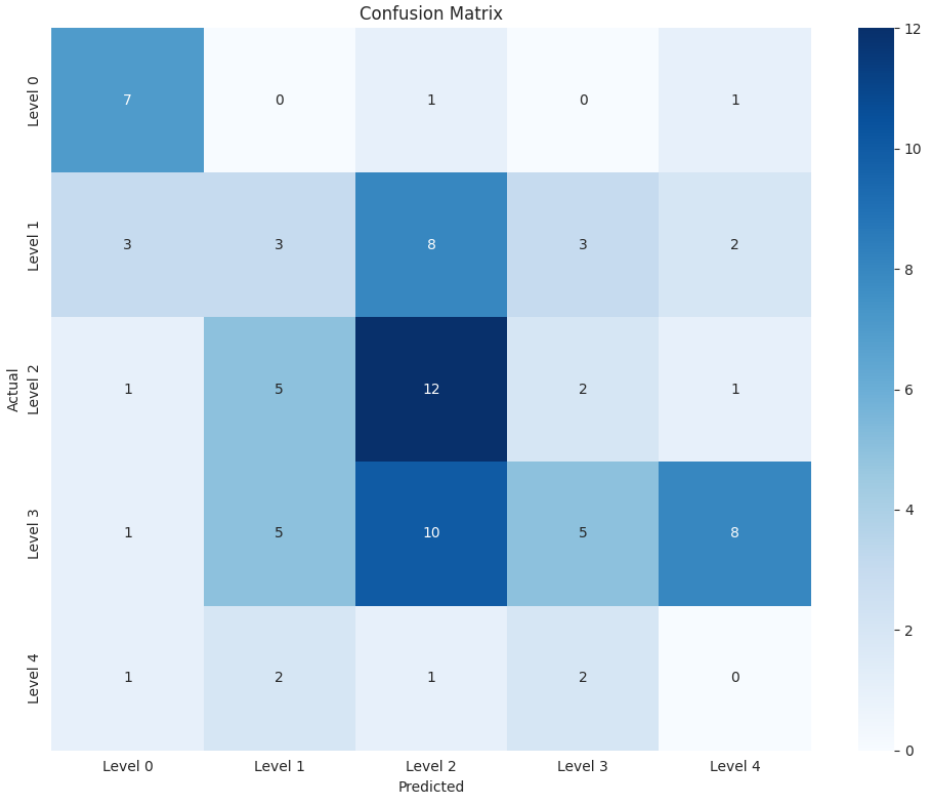
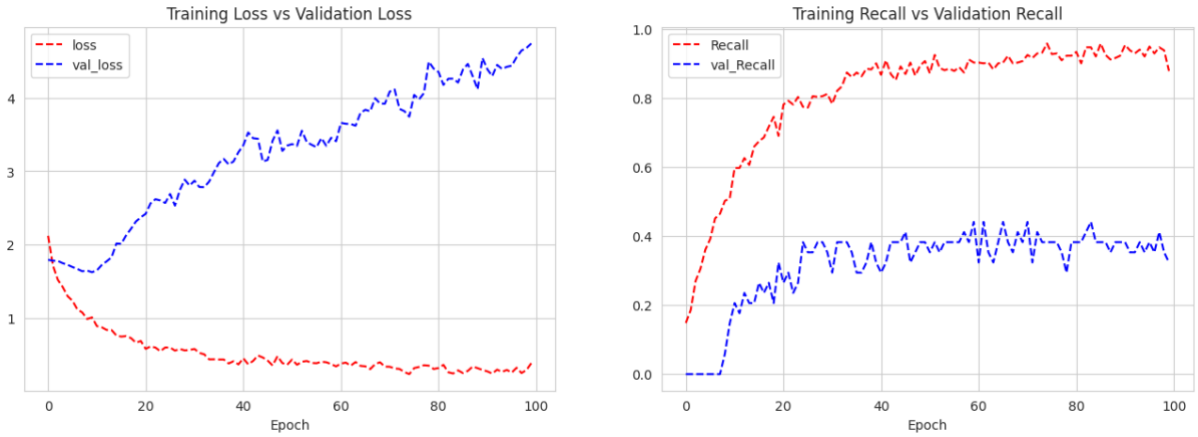


Figure: Confusion Matrix for Updated LSTM Model – Variant D

Updated LSTM Model Classifier – Variant E

Layer (type)	Output Shape	Param #
lstm_36 (LSTM)	(None, 1, 32)	53,376
batch_normalization_49 (BatchNormalization)	(None, 1, 32)	128
lstm_37 (LSTM)	(None, 16)	3,136
batch_normalization_50 (BatchNormalization)	(None, 16)	64
dropout_44 (Dropout)	(None, 16)	0
dense_44 (Dense)	(None, 16)	272
batch_normalization_51 (BatchNormalization)	(None, 16)	64
dropout_45 (Dropout)	(None, 16)	0
dense_45 (Dense)	(None, 5)	85

Total params: 57,125 (223.14 KB)
Trainable params: 56,997 (222.64 KB)
Non-trainable params: 128 (512.00 B)



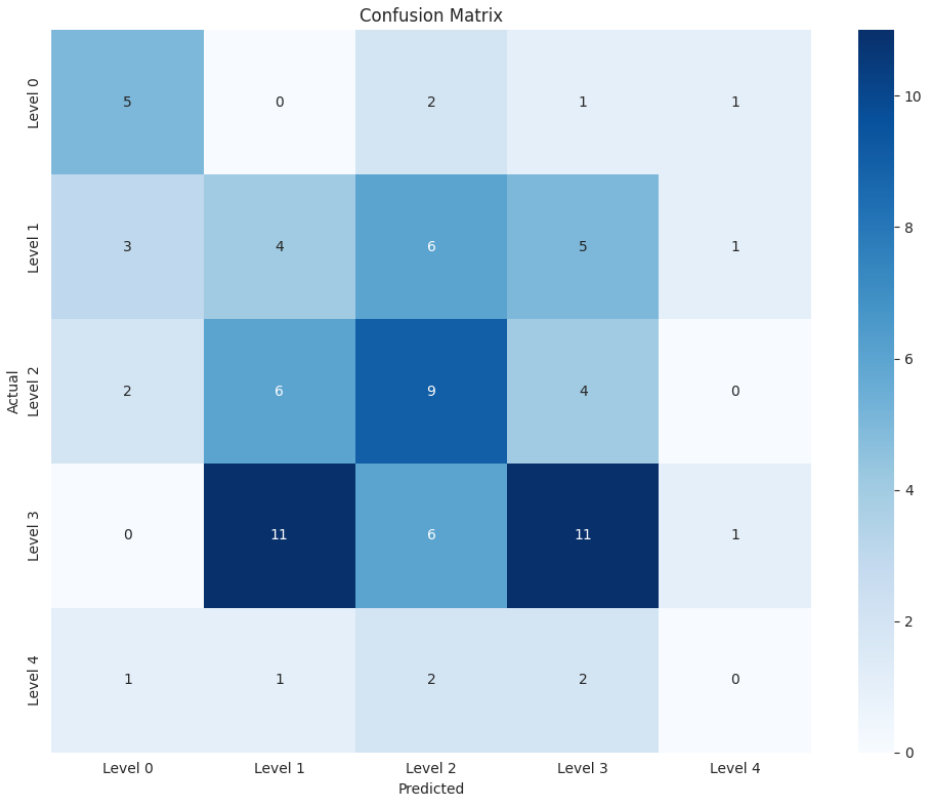


Figure: Confusion Matrix for Updated LSTM Model – Variant E

LLM Models

Data Preparation

- We have removed Named-entity details for consumption by LLM models
- Converted ordinal values to text, say 'Local_01' to 'one'
- Replacing specific values in 'Employee_Type' and 'Location' with more generic terms
- Using a combined description column

Using RAG (Retrieval Augmented Generation) for the ‘description’ column

we have used RAG for providing the contextual information as a few shot prompts. As the description column has very industry-specific data, we used RAG to provide more domain-specific knowledge and historical data to the model for accurate predictions.

- We have used Local Chroma DB for storing embeddings
- The Embedding Model is *models/embedding-001 models/embedding-001 (vector size=128)*, which is compatible with Gemini Models
- We have used the Gemini model for RAG
- The document contains ‘Description’, and metadata columns include ‘Potential_Accident_Level` and `Industry_Sector`

Also as the prediction needed some reasoning, we used ReAct Prompting which helped the model to correctly predict several severe incidents correctly

Retrieving similar incidents and their metadata from Vector Storage

We are able to provide an incident description, which is then searched within ChromaDB, which gives a list of matching incidents, with a high similarity index

Using ReAct Prompting with RAG

We have used ReAct as well (Reasoning and Acting), a framework that synergizes reasoning and acting capabilities within LLMs.

ReACT with RAG workflow that was used can be described as below:

- **Input**: Description text.
- **ReAct Reasoning**: The model retrieved similar incidents using RAG and reasons through the retrieved data.
- **Classification**: The reasoning leads to a well-informed classification. (a loop to get more examples from RAG was executed as part of the ReAct prompting by LangChain till the confidence level of the model was high)

This combination leverages RAG's structured specific knowledge and ReAct's logical reasoning power, making it a superior approach for this complex text classification task.

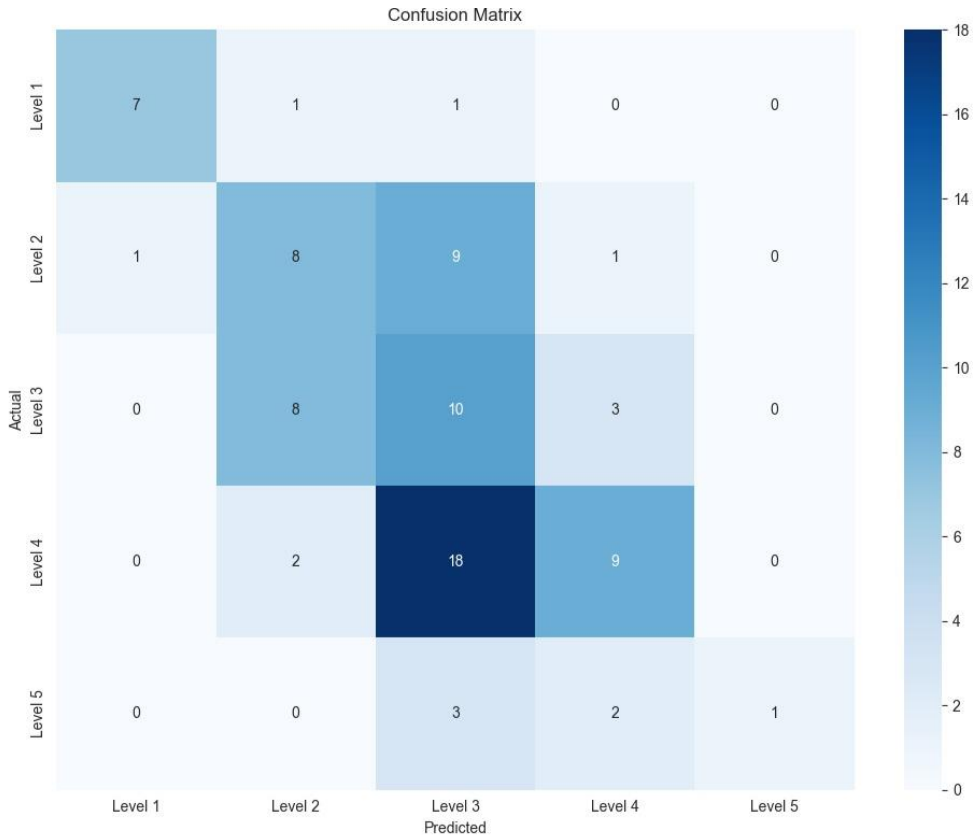


Figure: Confusion Matrix for LLM Gemini Pro Model (with RAG)

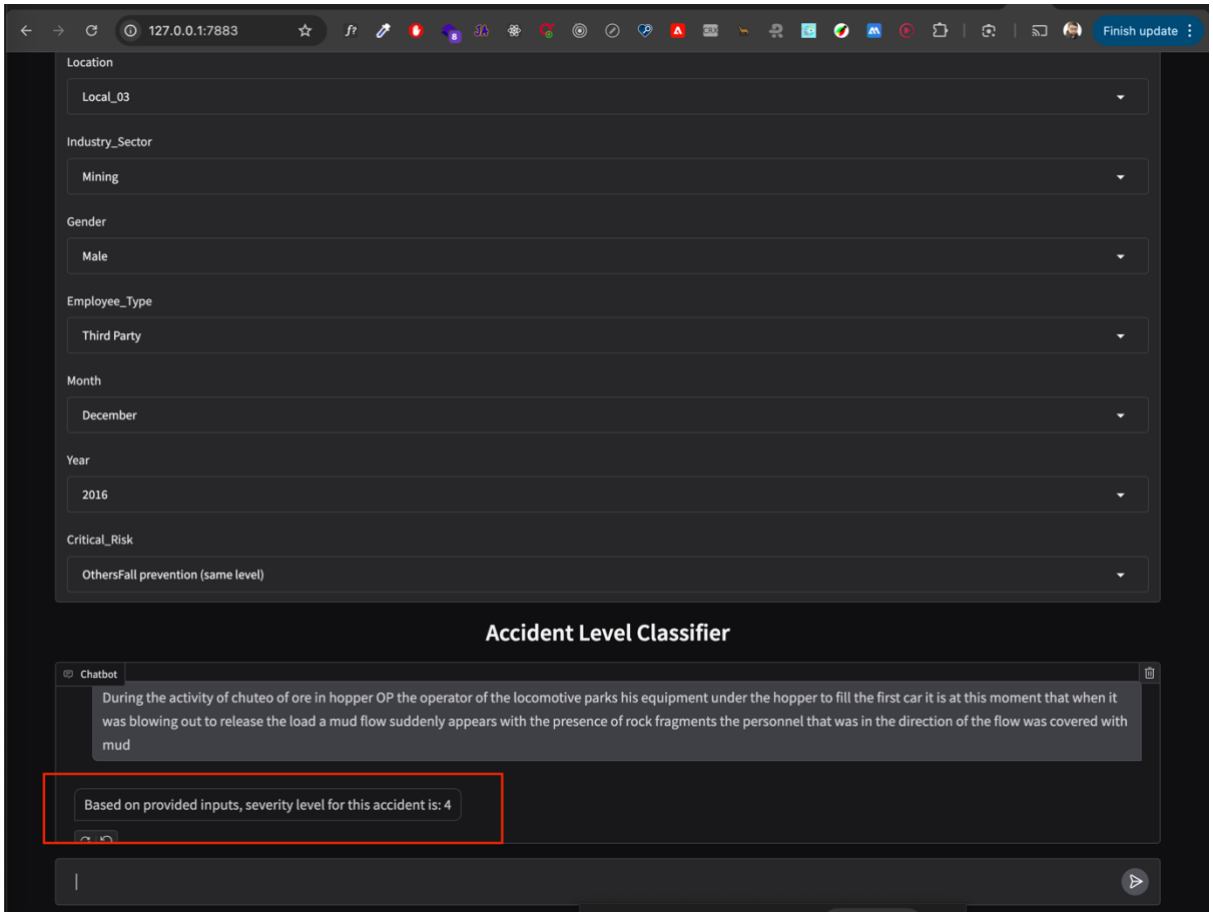
Advanced NN Models Performance Comparison

Model	Accuracy	Recall 5	Recall 4	Recall 3	Recall 2	Recall 1
Base ANN Model	0.39	0.0	0.44	0.33	0.25	0.77
Updated ANN Model	0.34	0.0	0.93	0.0	0.1	0.0
Base LSTM Model	0.38	0.0	0.51	0.32	0.22	0.55
Updated LSTML Model – Variant A	0.39	0.0	0.5	0.52	0.15	0.55
Updated LSTML Model – Variant B	0.39	0.0	0.45	0.33	0.31	0.77
Updated LSTML Model – Variant C	0.33	0.0	0.34	0.38	0.26	0.55
Updated LSTML Model – Variant D	0.32	0.0	0.17	0.57	0.15	0.77
Updated LSTML Model – Variant E	0.34	0.0	0.38	0.43	0.21	0.55
LLM_Gemini_Pro (with RAG)	0.41	0.16	0.31	0.47	0.42	0.82

Looking at all the models, it seems that **LLM Gemini Pro (with RAG)** is giving the best results (both accuracy and recall)

ChatBot UI

We were able to use Gradio to create a Chatbot UI, which takes non-descriptive inputs like Country, Location etc. as dropdowns; and descriptions as input text message



As next steps, we will try to host the UI on a publicly accessible URL

Implications

While the selected model has shown some improvement, it still requires further fine-tuning to achieve optimal performance.

Currently, the best accuracy achieved is approximately 41%, indicating significant room for enhancement.

Moreover, the model struggles to accurately predict the most severe class, which is critical in real-world applications and emphasizes the need for continued refinement.

Limitations

Several constraints were encountered while developing the model:

- **Data limitations**: The training dataset was limited in size, which can impact the model's ability to generalize.
- **Synthetic data limitations**: Although synthetic data was used to supplement the limited real-world data, it cannot fully replicate the complexity of real-world scenarios.

- **Class imbalance**: The dataset suffered from class imbalance, with the most severe class being underrepresented.
- **Limited correlations**: Bivariate analysis revealed limited correlations between columns, making it challenging to identify meaningful relationships.
- **Domain-specific language models**: The absence of industry-specific large language models (LLMs) for domains like metal and mining hindered the model's performance. General-purpose LLMs failed to deliver satisfactory results.
- **Feature gaps**: The dataset lacked relevant features that could have improved the model's accuracy.
- **Description bias**: The missing presence of strongly descriptive keywords in the descriptions may have diluted the severity, affecting the model's predictions.

Closing Reflections

- **Handling challenges with real-world data**: Real-world data poses a unique challenge, as it tends to be more disparate and heterogeneous compared to synthetic data. In contrast, synthetic data is often more uniform and easier to work with, as it is generated through controlled processes.

- **Teamwork**: This project highlighted the importance of teamwork, time management, communication, and adaptability. Future projects can benefit from these lessons, emphasizing the value of collaboration and continuous learning.

- **Early Feedback:** Reach out to mentors more frequently and get early feedback

Conclusion and Further Consideration for Improvement

The best model we do have is the **Gemini LLM model supported by RAG**, which has the best Accuracy and Recall metric. More training data would have helped us develop more robust traditional Machine Learning classifiers.

Among all the embeddings, transformer-based embeddings give the best performance.

However, the abysmal performance of these models is sufficient to convince us that none of the models built on this small data set can be deployed for predictions in a real-time production environment.

To further enhance the model's performance and overcome current limitations, the following areas can be explored:

- **Model experimentation and optimization**: Utilize GridSearchCV for systematic experimentation and hyperparameter tuning to identify the most suitable models for the niche industry.
- **Infrastructure upgrades:** Access to more powerful GPU machines would facilitate faster experimentation and training of complex models.
- **Domain-specific resources**: Leverage industry-specific dictionaries and terminology to improve the model's understanding of domain-specific concepts.
- **Advanced NLP techniques**: Investigate cutting-edge NLP methods, such as Agentic Models, to potentially achieve better results.
- **Data refinement**: Fine-tune the model on a more accurate and relevant dataset to improve its performance.
- **Prompting methods**: Experiment with different prompting techniques, like Chain-of-Thought (COT), to enhance the model's output.

- **Alternative LLM models**: Explore other domain-specific LLM models to identify the most suitable one for the industry.

Additionally, exploring **End-to-end deployment** options to streamline the model's integration into real-world applications.