AIML Capstone Project – NLP1 Interim Report

NLP based Chatbot project
for safety department of Industrial companies

Project Interim Report prepared by

Kavita Bansal

Kamal Pandey

Jagadeesh K

Indira R Rao

Shanavaz G

Presented to Great learning Mentor **Mr. Aniket Chhabra**

Date: 01-June-2025

Table of Contents

1. Objective of this project

1.1 Problem Statement

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

Objective of this dataset is to design a ML/DL based chatbot utility for the Industrial safety division of any industries/companies. There is an urgent need for industries/companies around the globe to understand why employees still suffer some injuries/accidents in plants. Some scenarios, they even die in such an environment. It 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.

This chatbot utility will be designed and developed with the intent of helping the professionals (employees or third party-contractors) to highlight the safety risk defined as per incident description.

Within the problem statement, 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 Data Description

The database is basically records of accidents from 12 different plants in three different countries where every line in the data is an occurrence of an accident. Different columns with its description are as follows:

Every data record is an accident or an incident occurred and reported and extracted in the form of a CSV file to us.

1. **Data**: timestamp or time/date information
2. **Countries**: which country the accident occurred (anonymized)
3. **Local**: the city where the manufacturing plant is located (anonymized)
4. **Industry sector**: which sector the plant belongs to
5. **Accident level**: from I to VI, it registers how severe was the accident (I means not severe but VI means very severe)
6. **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)
7. **Genre**: if the person is male of female
8. **Employee or Third Party**: if the injured person is an employee or a third party
9. **Critical Risk**: some description of the risk involved in the accident
10. **Description**: Detailed description of how the accident happened.

1.3 Summary of data column

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

Consideration: The potential target variable is selected as be 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 classifiers.

Feature	Description
Data	Timestamp or time/date information. Recorded timestamp of the incident.
Countries	Country where the accident occurred (anonymized). Values are _01, _02, and _03. Categorical data.

Feature	Description
Local	Location within the country where the manufacturing plant is located (anonymized). Data from 12 plants/locations.
Industry Sector	The sector the plant belongs to. High-risk sectors like metals. Values include Mining, Metals, and Others.
Accident Level	Severity of the accident, from Level I to V. I = Not severe, V = Very severe. This is one of the target variables.
Potential Accident Level	How severe the accident could have been. Scale from 1 (not severe) to 6 (highest severity). This can also be used as a target variable.
Gender	Gender of the individual involved in the accident (Male or Female).
Employee or Third Party	Role of the person involved: Employee, Third Party, or Third Party (Remote).
Critical Risk	Description of the critical risk involved in the accident.
Description	Detailed narrative of the accident. This text will be used in NLP preprocessing for predicting accident levels.

2. EDA and Text Preprocessing analysis

This section comprises of two major elements:

- **EDA - Exploratory data analysis on categorical columns**
  - In this, we will do univariate, bivariate and statistical significant hypothesis tests to determine which parameters drives our target variable prediction
- **NLP text preprocessing on Description column**
  - Here we will employ various NLP preprocessing models like n-gram, word cloud and find words that will help in predicting the target variable.

As earlier mentioned, we will have to employ a multi classification models for our prediction.

2.1 Import Dataset

```
[ ] from google.colab import drive
    drive.mount('/content/drive')
```

```
Drive already mounted at /content/drive; to attempt to forcibly remount, call drive.mount("/content/drive", force_remount=True).
```

```
# read excel data into dataframe
df_ish = pd.read_excel('/content/drive/MyDrive/Capstone/ISH_dataset.xlsx')
```

FIGURE 1 IMPORT DATASET

2.2 Approach to EDA

We will do the following

- Exploratory Data Analysis
- Charts: Univariate & Bivariate Analysis.

- NLP Pre-Processing steps:
- Concluding on EDA part

2.3 Exploring Data Artifacts

a) info of the data frame :

- Present the complete information for the loaded data frame. It will share the column information with data type and present if data is null or non-null.

```
[ ] # Dataset info
    df_ish.info()

<class 'pandas.core.frame.DataFrame'>
RangeIndex: 425 entries, 0 to 424
Data columns (total 11 columns):
 #   Column                  Non-Null Count  Dtype
---  ------                  --------------  -----
 0   Unnamed: 0              425 non-null    int64
 1   Date                    425 non-null    datetime64[ns]
 2   Country                 425 non-null    object
 3   Local                   425 non-null    object
 4   Industry Sector         425 non-null    object
 5   Accident Level          425 non-null    object
 6   Potential Accident Level 425 non-null   object
 7   Gender                  425 non-null    object
 8   Employee Type           425 non-null    object
 9   Critical Risk           425 non-null    object
 10  Description             425 non-null    object
dtypes: datetime64[ns](1), int64(1), object(9)
memory usage: 36.7+ KB

[ ] # Dropping the index column
```

FIGURE 2 COMPLETE INFORMATION FOR THE LOADED DATA FRAME

2.4 Dataset Cleansing

a) Dropping Data Variables:

- Drop unnamed column: There is no significance of the unnamed column hence dropping off the column.

```
[ ] # Dropping the index column
    df_ish.drop('Unnamed: 0', axis=1, inplace=True)
```

Table-3 dropping the unnecessary columns

b) Renaming columns:

- The column names were misspelled which needs to be change to correct.
- Data and genre are changed to Date and Gender. Countries updated to Country.
- Employee or third party to Employee Type.
    - Data → Date
    - Countries → Country
    - Genre → Gender
    - Employee or Third Party → Employee Type

```
[ ] # Renaming 'Data', 'Countries', 'Genre' , 'Employee or Third Party' columns in Data frame

    df_ish.rename(columns={"Data": "Date", "Countries": "Country", "Genre": "Gender" , "Employee or Third Party":"Employee Type"},inplace=True)
```

FIGURE 3 RENAMING THE COLUMNS

c) Duplicate & null values Removal:

- Number of duplicate rows are 7 which need to be dropped to do cleanup.

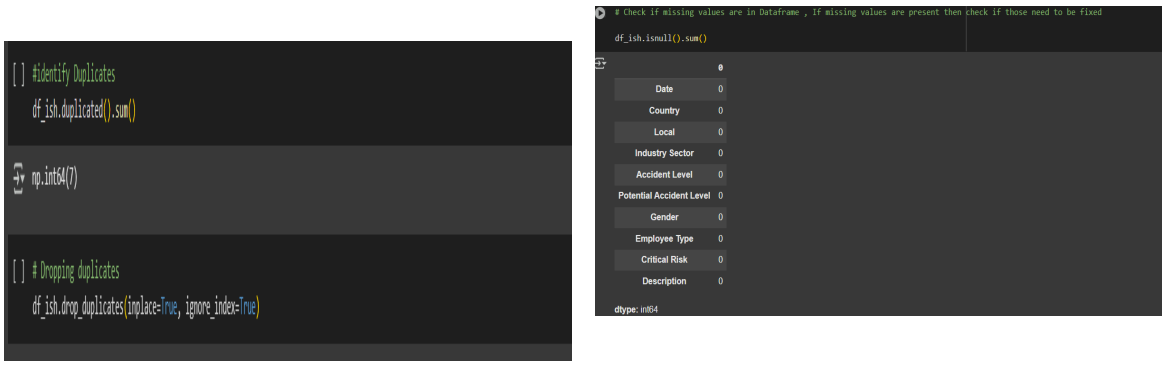


FIGURE 4 DUPLICATE AND NULL VALUE REMOVAL

d) Additional Variable generation:

- Split the date to Year, Month and day to further drill down the safety data to check it by year, month, day wise.

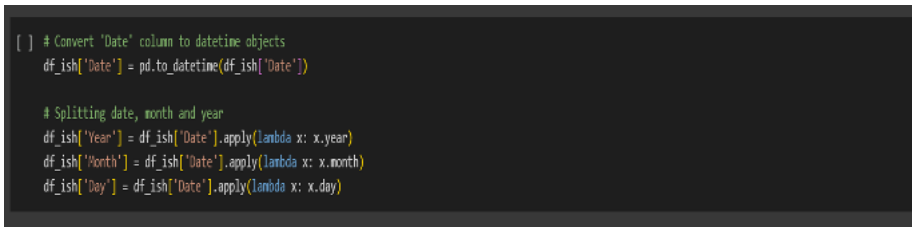


FIGURE 5 EXTRACTION OF DATE TO YEAR, MONTH AND DAY

e) Variable type extraction

- Extract the variable into Categorical and numerical column. We have all the column appeared as Categorical variable except the dynamic generated variable as day, year, month.

FIGURE 6 CATEGORICAL AND NUMERICAL VARIABLE EXTRACTION

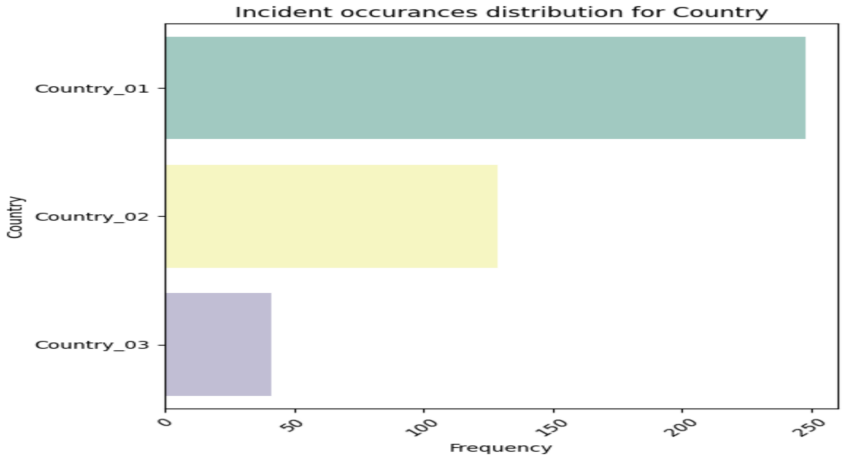
Observations:

- The dataset consists information on industrial accidents across different countries, cities, and industry sectors.
- The accident time frame is captured via 'Date' column.
- The severity level of accidents is categorized into levels from I to VI.
- Information related to gender, employee type, critical risk, and a detailed description of the accident is provided.

2.5 Visualization (Univariate, Bivariate analysis) and Statistical Analysis

2.5.1) Univariate Analysis:

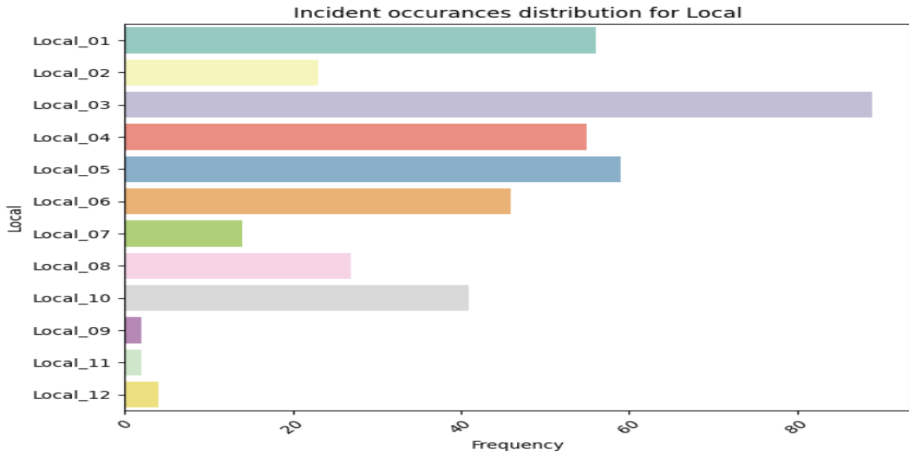
a) Distribution of Incident occurred in different countries



Observation:

- Country_01 experiences the most incidents. Likely needs priority attention for risk mitigation.
- Country_01 likely contributes more than half of the total incidents, indicating a hotspot or data concentration.
- Country_03's low incident count could reflect better controls or underreporting.

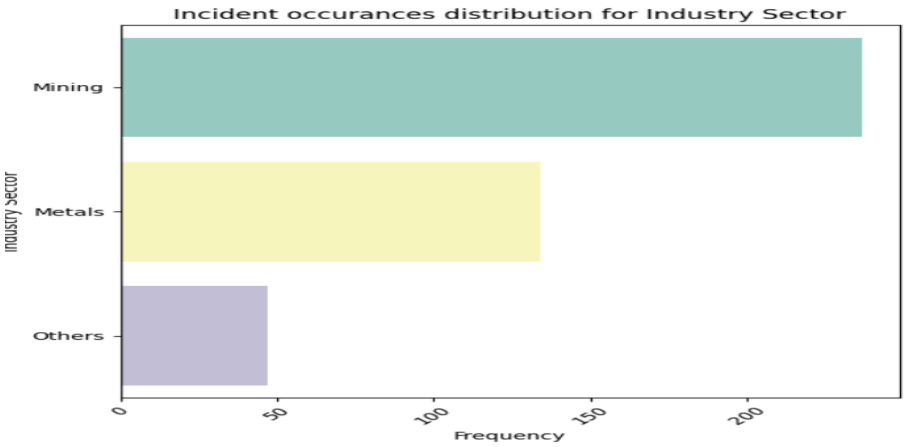
b) Incident distribution across different localities



Observation:

- Local_03 Highest number of incidents. Needs immediate attention.
- local_03 is a clear outlier with the highest incident frequency.
- Local_04, 05, 10 also shows elevated incident rates.
- Areas with very low incidents (Local_09, 11, 12) could be:
  - Lower risk areas, or
  - Underreporting zones needing validation.

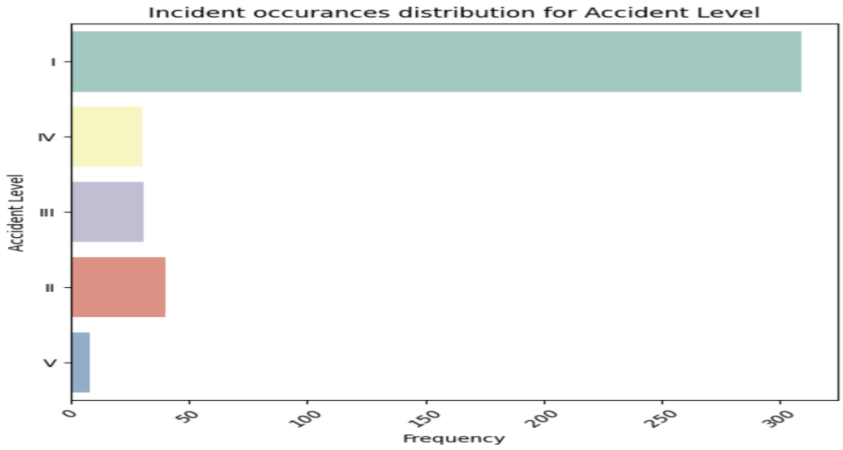
c) Incidents count distribution across different industry sectors



Observation:

- Mining sector is having greater frequency of accident.
- Other appearing for least frequency.

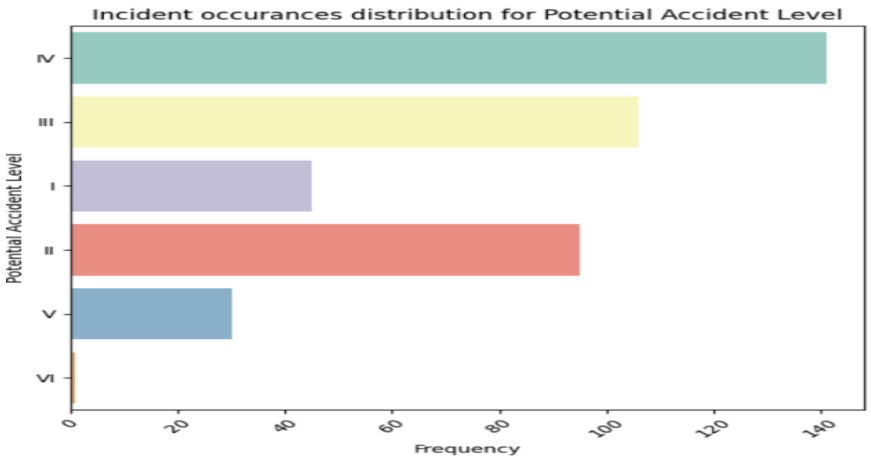
d) Incidents count distribution across different Accident levels



Observation:

- Most of the accidents are appearing with Category I, means with less severity.

e) Incidents count distribution across different Potential Accident levels

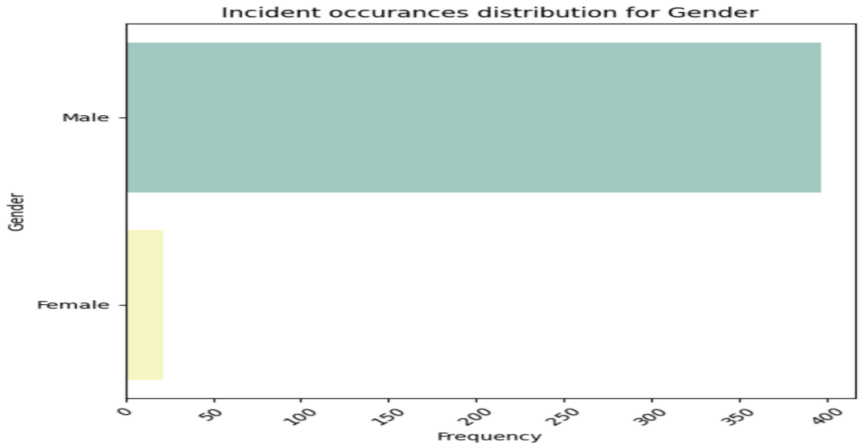


Observation:

Immediate Action for Levels IV & III

- These represent ~60% of total incidents.
- Investigate common root causes (e.g., unsafe behaviours, procedures).
- Enhance training, supervision, and early warning systems.

f) Incidents count distribution across Genders



Observation:

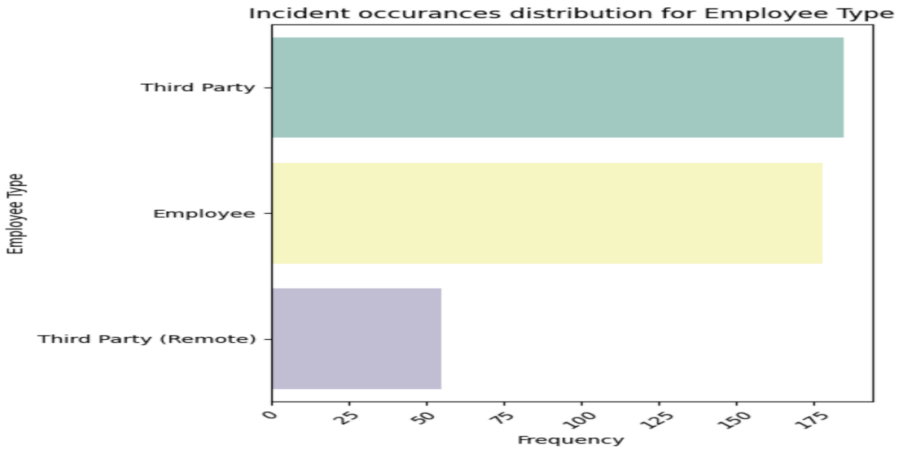
Male workers are disproportionately affected:

- Likely reflects greater exposure to high-risk roles, physical labour, or hazardous environments.
- May also indicate underreporting among female workers in certain cases.

Female incident frequency is very low:

- Could mean fewer females in operational roles, or underreporting due to cultural, procedural, or visibility gaps.
- Important to confirm whether this is due to actual exposure levels or systemic reporting bias.

g) Incidents count distribution across different Employee Types



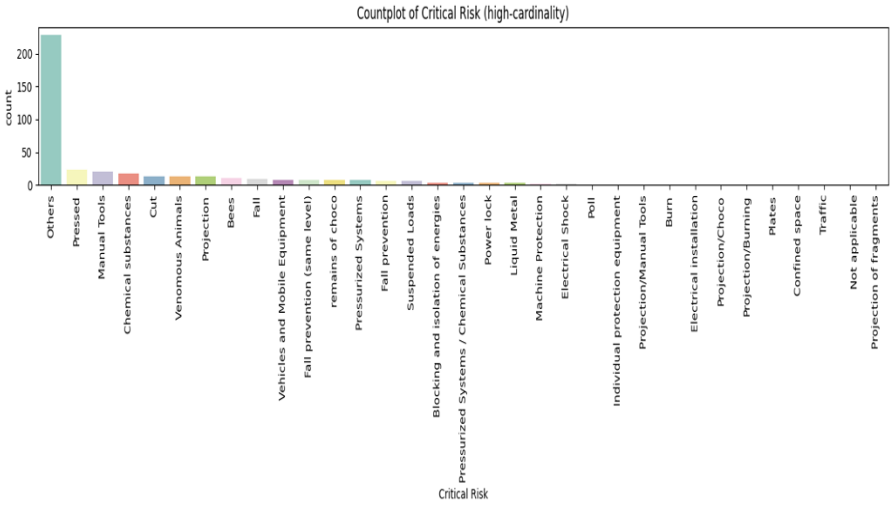
Observation:

- Higher incident frequency for Third-Party employees – The Third Party category has the highest occurrence of incidents, with the bar nearing 175. This suggests a potential risk factor associated with outsourced or contract employees.
- Employees follow closely – The Employee category has a slightly lower number of incidents (~170), indicating that incidents are relatively frequent across direct staff as well.
- Third-Party (Remote) employees report the fewest incidents – With occurrences around 50, the Third Party (Remote) group experiences significantly lower incident frequency. This could imply that remote work minimizes exposure to hazardous conditions or physical risks.

Possible Implications for Risk Management:

- Higher risk exposure among third-party staff may call for stricter safety protocols or enhanced training for contract employees.
- The lower incidents among remote third-party workers suggests that on-site presence could be a contributing factor to incident occurrence.

h) Incidents count distribution across different critical risks



Observation:

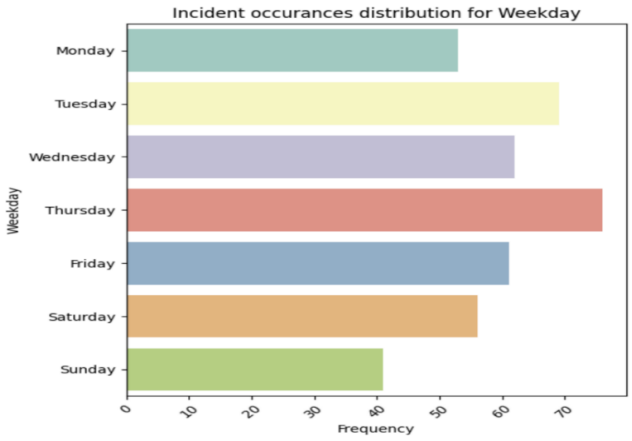
Prioritize Fall Prevention Programs

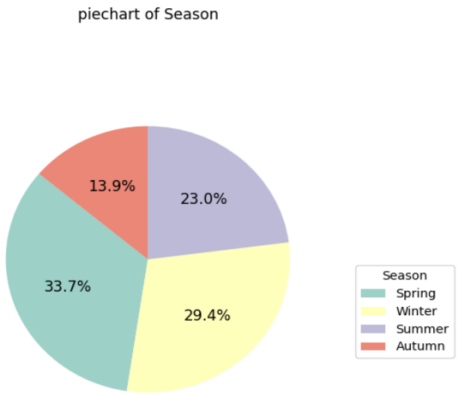
- Implement proper signage, and housekeeping audits.
- Use edge protection, and regular floor inspections.
- Educate on common trip hazards—awareness is key for low-severity/high-frequency risks.

Prepare for Low-Frequency, High-Severity Risks

- Maintain readiness for electrical shocks, burns, mobile equipment incidents—even if rare, the impact can be fatal.
- Use scenario-based emergency drills for these categories.

i) Month wise and Season wise distribution of Incidents counts





Observation:

- Spring and Winter together contribute over 63% of total incidents.
- Summer may introduce risks like heat stress, fatigue, or outdoor hazards.
- Autumn, though lowest, might represent a time of improved preparedness or decreased activity.

Season wise Recommendation

- Spring (Highest Risk)
  - Perform equipment calibration after winter downtime.
  - Conduct safety reorientation sessions if new staff rotations occur.
- Summer
- Rotate people to avoid heat exhaustion
- Encourage reporting of minor symptoms early to prevent major health incidents.

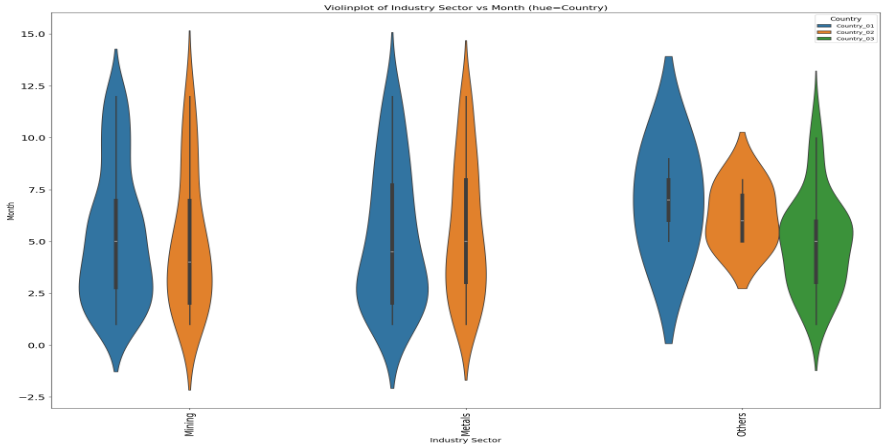
2.5.2) Bivariate Analysis

```
#bivariate

bivariate_graphPrep('violinplot',variablePairs ,df_ish)
```

Box plot Analysis

j) Violin plot of Monthly Trends of cross countries incident occurrences Across Industry Sectors



Observation

Mining:

- Country_01 and Country_02 dominate the mining sector.
- Incidents mainly occur between Feb and Aug, peaking around May–June.
- Both countries share a similar spread, suggesting common environmental or operational cycles

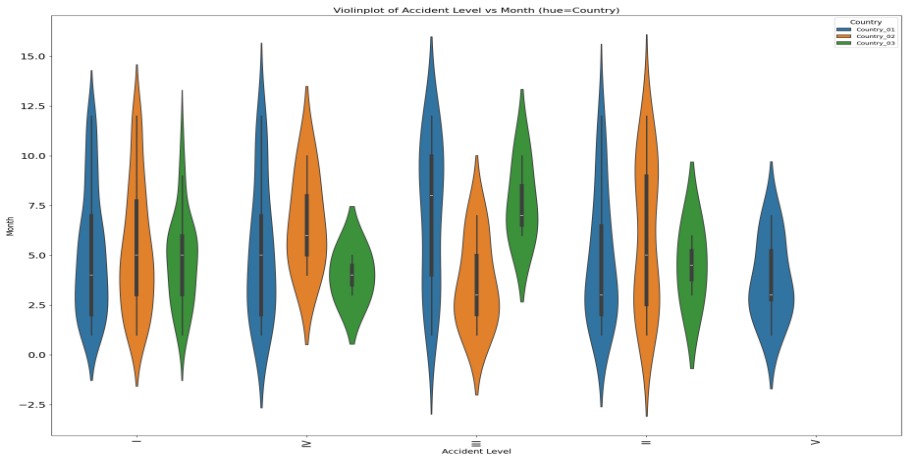
Mining:

- Incidents span Feb to Nov with a wider spread across months.
- Country_01 and Country_02 are again the key contributors.
- Slightly more even distribution, but still clustered around Mar–Jul.

Others:

- Narrower time range: main activity between Mar–Jul.
- Unique presence of Country_03, not seen in other sectors.
- Outliers occur around Nov–Dec, potentially rare events or unique process incidents.

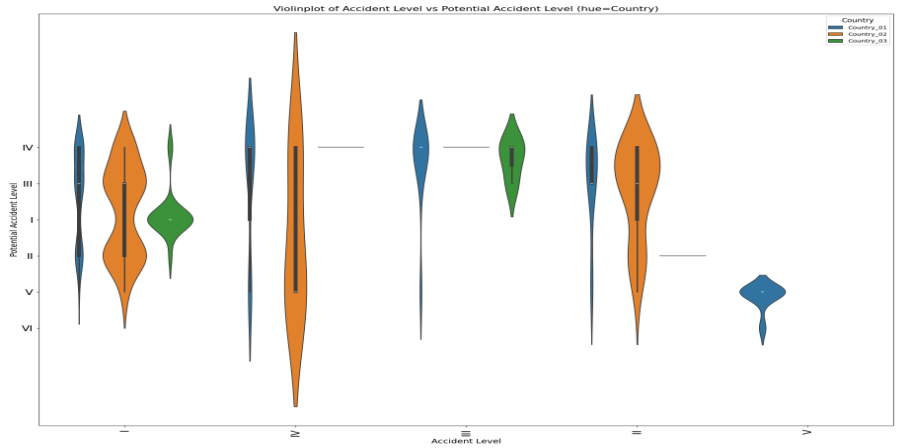
k) Violinplot of monthly trend of Accident Levels across different countries



Observation

- Accident Levels I, II, and IV are concentrated in Q1–Q3 (Jan–Sep), with peak density around Mar–Jul.
- Level IV has the widest month range and cross-country presence → highest volatility.
- Country_01 is involved in all levels, including the most severe (Level V).
- Country_02 spans across Levels I–IV, heavily represented in Levels I–III.
- Country_03 appears in Levels I–IV, with strong involvement in Level III and Level IV—emerging risk exposure.

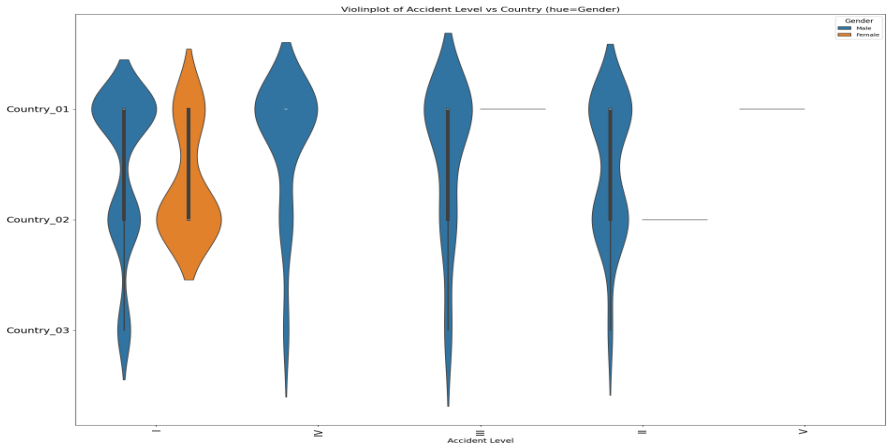
l) **Cross-Country Comparison of Potential Accident Levels trend vs potential accident level(Violinplot View)**



Observation

- Accident Level I shows a high mismatch with potential severity → many were close calls or poorly rated at incident time.
- Level II & IV have similar potential severity ranges as more severe levels, indicating underappreciated risks in moderate-severity incidents.
- Country_03 shows up in Level III only, with narrow and aligned ranges—suggesting possibly stronger risk control or limited data scope.
- High presence of Level IV potential across all actual levels implies systemic risk exposure.

m) **Gender Comparison of Accident Levels across different countries (Boxplot View)**

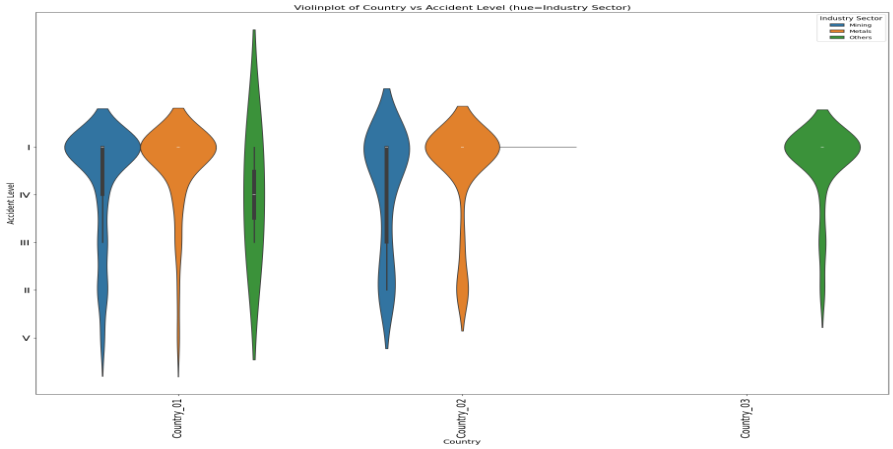


Observation

Accident Level vs Gender:

- A significantly higher number of males are involved in accidents across all accident levels.
- Median accident level is around III.
- Country_01, Country_02 are dominating.
- Outliers exist for country_02.
- Males show greater visibility among accident level.
- Female shows concentrated distribution with a smaller number of extreme values.

n) **Cross-Country Comparison of Accident Levels in different industries(Boxplot View)**

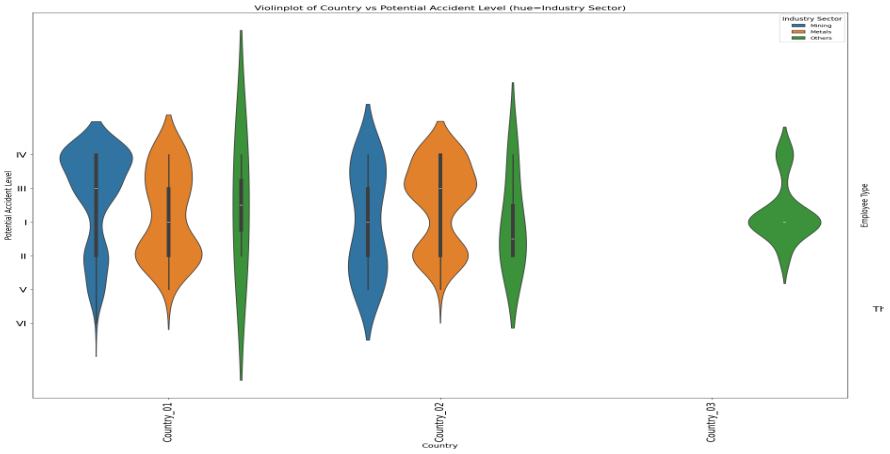


Observation

Country vs Accident level:

- Median: I, for both the countries.
- For country_01, dominated by mining sector, with one addition metals-green.
- For country_02, all with mining sector.
- High accident severity in country_01, country_02 in mining operative sector.
- More outliers at lower accident level.
- For country_03, data points are very less and not providing significant information.

o) **Cross-Country Comparison of Potential Accident Levels in different industries (Boxplot View)**



Observation:

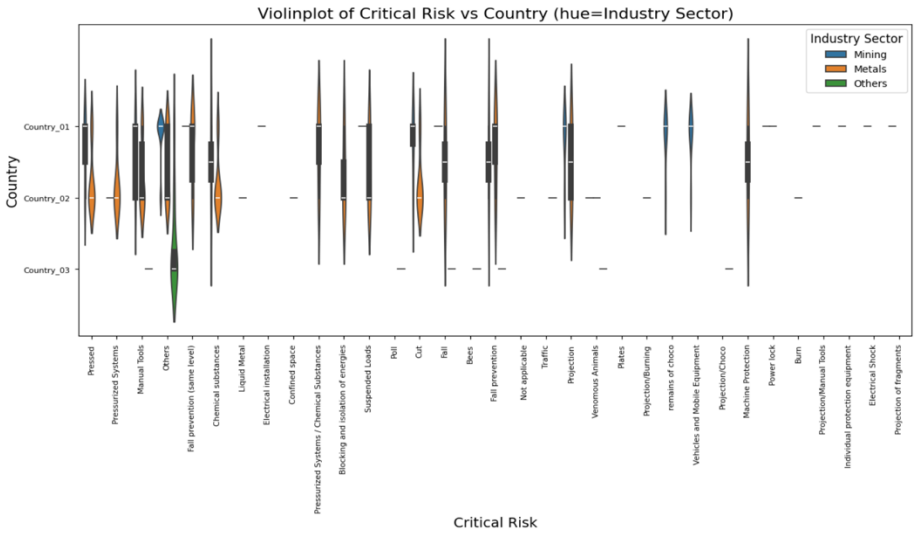
Country vs Potential Accident level:

- Median: I, for both the countries.
- For country_01, dominated by mining sector, with one addition metals-green.
- IQR spans from II to IV, display moderate spread.
- Sector wise representation for Country_01:
  - Mining presented with colour orange: Skews slightly higher in potential severity.
  - Metals presented with colour yellow Balanced, cantered around III.
  - Others presented with colour green: Slightly lower severity than Mining, similar to Metals.

- Sector wise representation for Country_01:
  - Mining & Other presented balanced distribution.

- Metals shows highest potential accident level, inclined towards IV.

p) Cross-Country Comparison of critical risks in different industries (Boxplot View)



Observation:

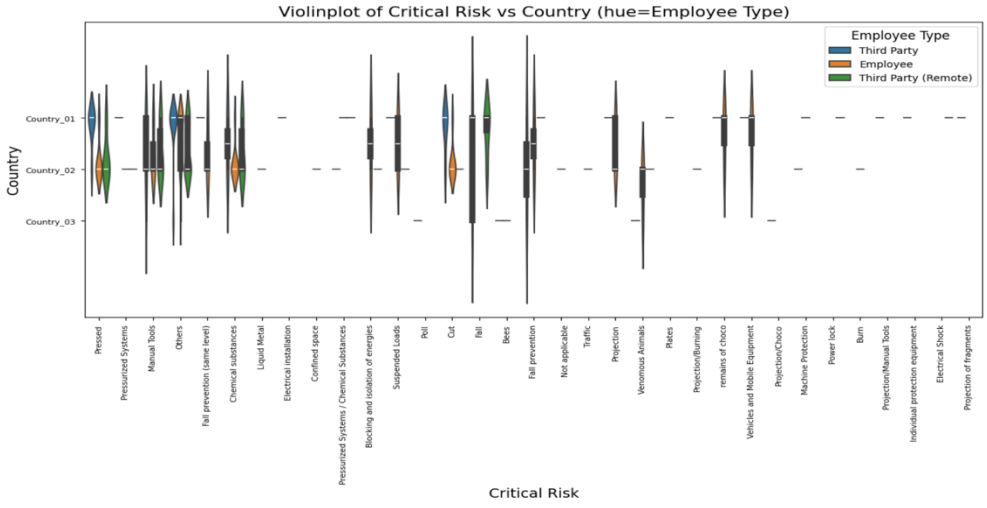
Country_01

- **High Incident Variability**: The spread of incidents suggests fluctuating risk exposure, especially in Mining-related categories.
- **Frequent High-Risk Events:** Falls and Mobile Equipment incidents show consistently high median values, indicating a need for focused safety measures.
- **Presence of Extreme Outliers:** A few incident types (such as Pressurized Systems) display significant deviations, which may require targeted risk mitigation strategies.

Country_02

- **More Controlled Incident Distribution:** The range of incidents appears more contained, implying stable risk management.
- **Moderate Incident Frequency:** While incidents like Electrical Shock and Machine Protection Failures remain notable, extreme fluctuations are less common.
- **Comparatively Stronger Safety Profile:** The lower variation suggests robust regulatory enforcement and industry-wide safety protocols.

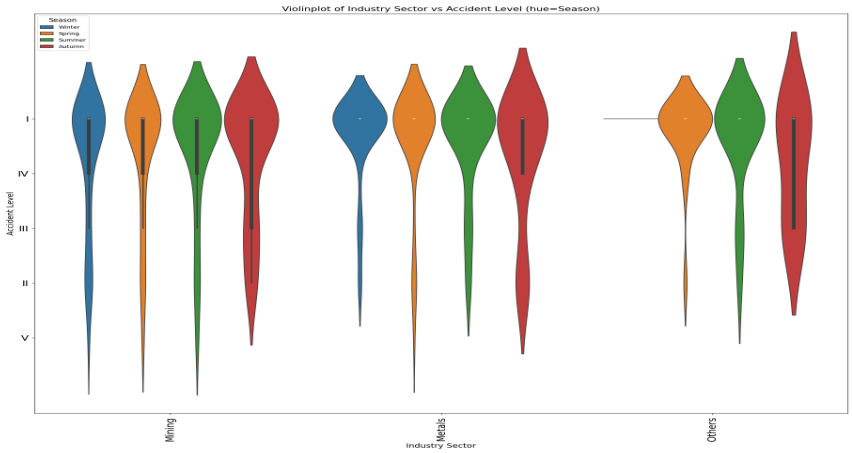
q) Cross-Country Comparison of critical risks of different employee types (Boxplot View)



Observation:

- Country_01 depends heavily on remote third parties, exposing them to concentrated risk types.
- Country_02 shows most balanced exposure, but also the most diverse in employment types.
- Country_03 reflects traditional outsourcing, where third parties face lower-level but widespread risks, while employees have more controlled risk exposure.
- Third Party workers across countries consistently face broader and often more dangerous conditions than employees.

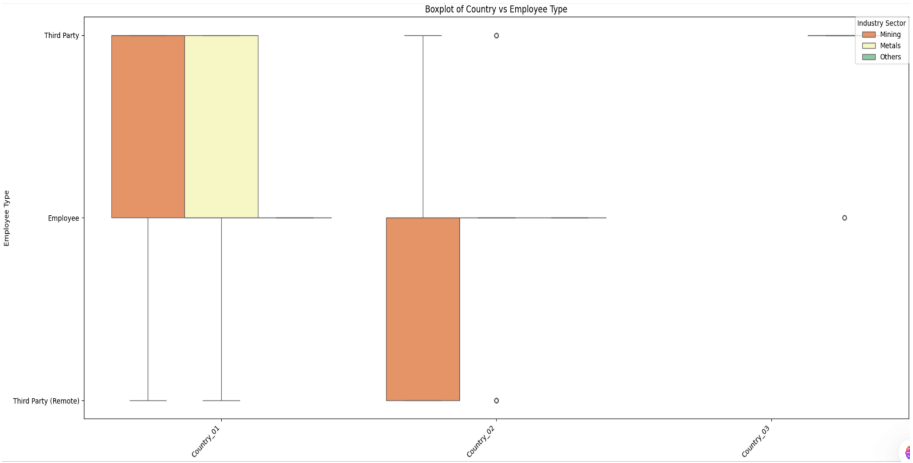
r) Season wise Comparison of accident level of different industry sectors (Boxplot View)



Observation:

- Mining industry is consistently hazardous, without any seasonal drop in accident severity.
- Mining industry is consistently hazardous, without any seasonal drop in accident severity.
- Metals has high severity in Autumn — this might indicate peak production or increased operational pressure.

s) Cross-Country Comparison of employee types in different industry sectors (Boxplot View)



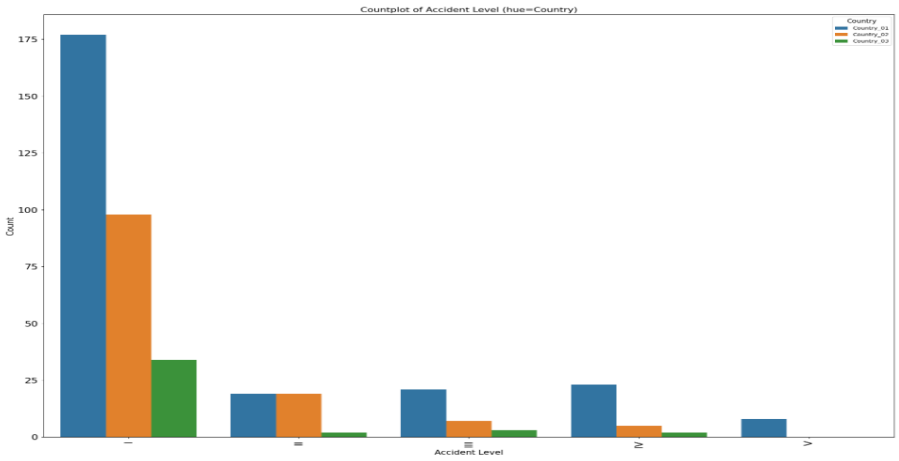
Observations:

- Third Party employment dominates in every countries — which presents transferring risk to external party.
- Mining sector spans multiple countries with variable employment structures.
- "Others" sector is visible only in Country_03, and it have Third Party employees.

- Country_01 display the balanced with representation across all employee types.

Count plot Analysis

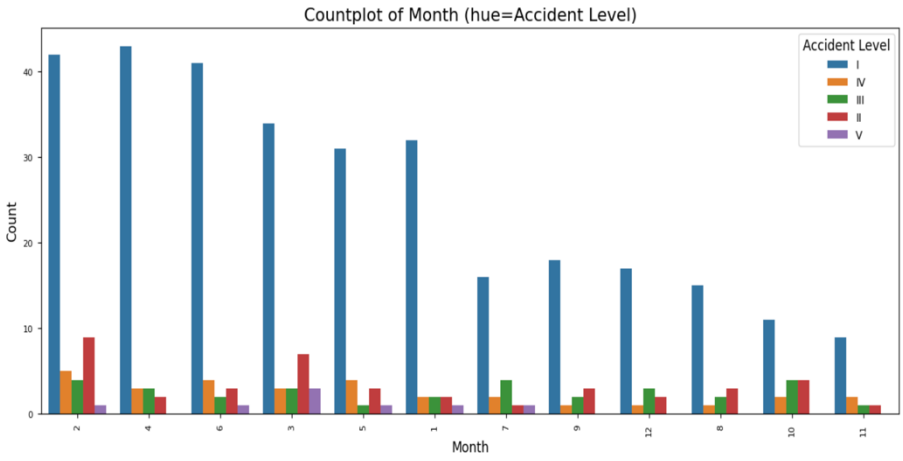
t) Cross-Country Comparison of Reported Incidents by Accident Level (Count plot View)



Observations:

- Total incidents: 248
- Level wise Splitting
  - Level I: 177
  - Level IV: 23
  - Level III: 21
  - Level II: 19
  - Level V: 8
- High number of low-severity incidents (Level I)
- Presence of higher levels suggests diverse safety risks and possible underreporting of preventive measures

u) Monthly Comparison of Reported Incidents by Accident Level (Count plot View)

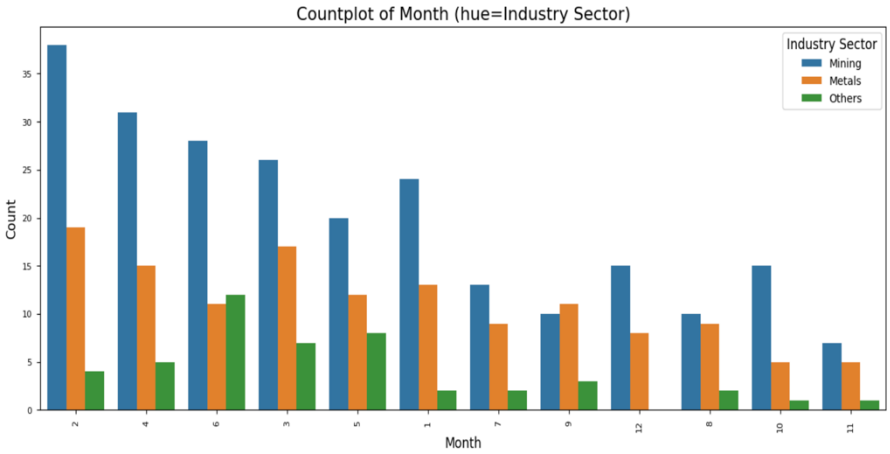


Observations:

- Most accident level is presented as Level I.

- Highest Monthly occurrence shown below for level I.
  - March: 43
  - January: 42
  - December: 41
  - October: 34
- Presence of Minor accidents are very frequent.
- Level IV, accident spread throughout the year.
- With the accident level II & III, incident may emerge during mid to late year operations, due to operational complexity.
- Accident level V is extremely rare, every month we can see 1-2 incidents.
- Even the accident level V are rare but needs attention and investigation.

v) Monthly Comparison of Reported Incidents by Industry Type (Count plot View)



Observation:

Mining Sectors:

- Maximum incidents reported in mining sector (54.43 %)
- Maximum of the incidents are reported in 1st (38.76%) and then in 2nd (34.8%) Quarter of total of incident reported.
- Highest incident reported Incidents in 2nd Months.
- 30.83% total incident reported in 3rd and 4th Quarter.

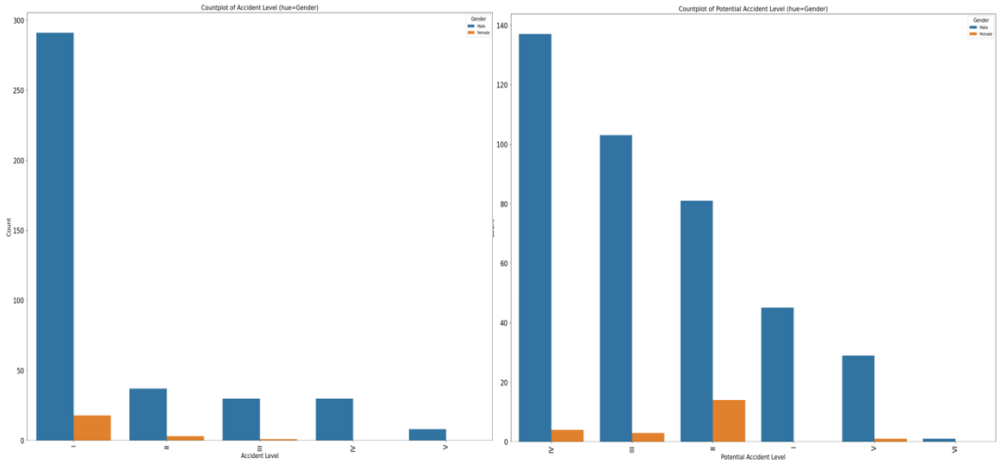
Metal Sectors:

- Maximum of the incidents are reported in 1st (36.5%) and then in 2nd Quarter (28.3%) of total of incident reported.
- Highest incident reported in 2nd Month (14.17%).
- 35.07 % total incident reported in 3rd and 4th Quarter.

Others:

- Maximum incident reported in 2nd Quarter (53.19%)

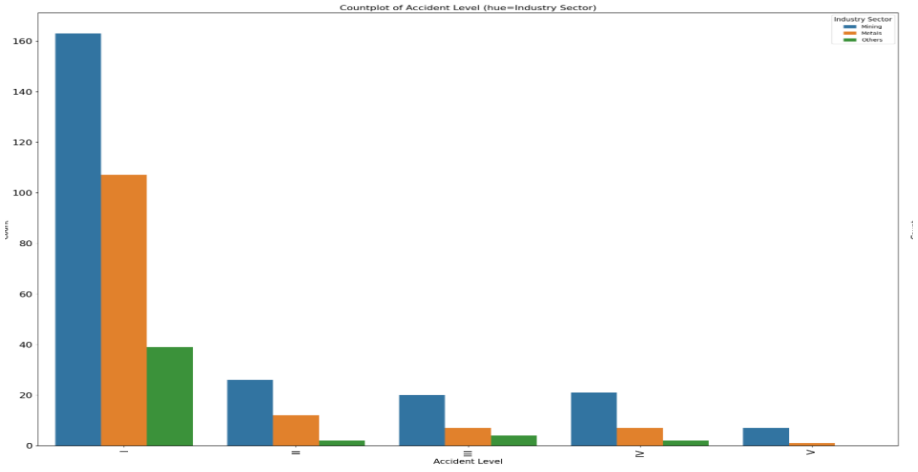
w) Gender-wise distribution of Accident Level and Potential Accident Level based on the count of reported incidents



Observation

- The majority of reported incidents involved male individuals compared to females.
- Accident Level L-1 was the most frequently observed for both male and female individuals.
- Males were more prone to Potential Accident Level L-4, whereas females showed a higher occurrence at
- Accident Level L-2.
- Accident Level L-5 and Potential Accident Level L-6 had the lowest number of reported cases.

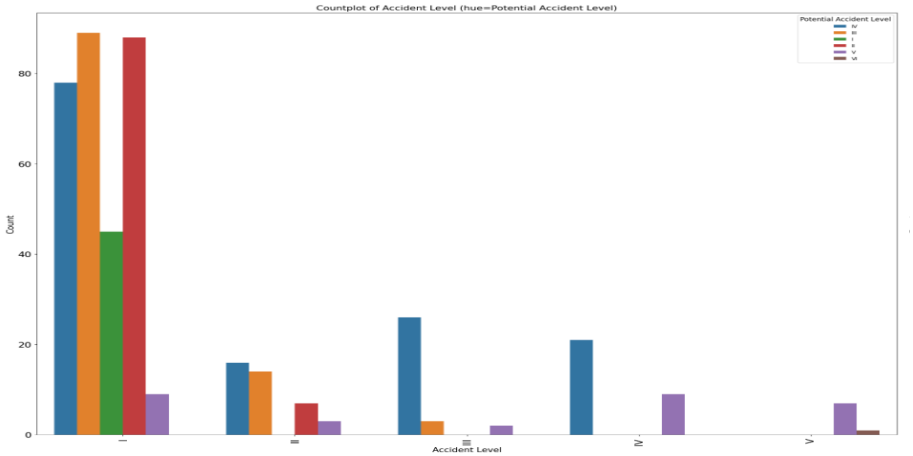
x) Industrial distribution of Accident Level Distribution



Observation:

- The highest number of accidents occurred in the Mining industry, predominantly at Accident Level L-1.
- Accident Levels L-2 to L-5 show relatively similar counts across industry sectors.
- Accident Level L-5 incidents were reported in very low numbers across all industry sectors.

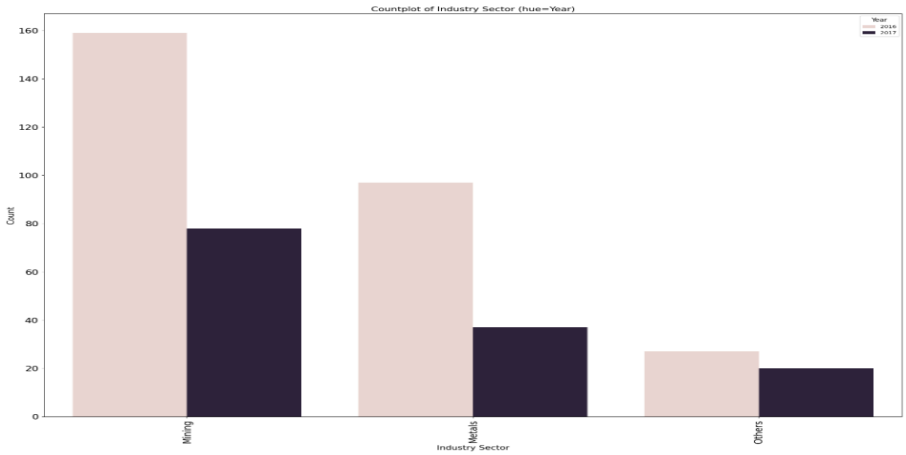
y) Relation Between Accident Level and Potential Accident Level



Observation:

- Accident Level L-1 is mostly associated with Potential Levels L-1 to L-4, and rarely with L-5 or L-6.
- Accident Level L-2 often leads to Potential Levels L-2 to L-5, and least to L-1 or L-6.
- Accident Levels L-2, L-3, and L-4 are more frequently linked to Potential Level L-4.
- Accident Level L-4 is commonly followed by Potential Levels L-4 or L-5.
- Accident Level L-5 has the lowest number of reported incidents.

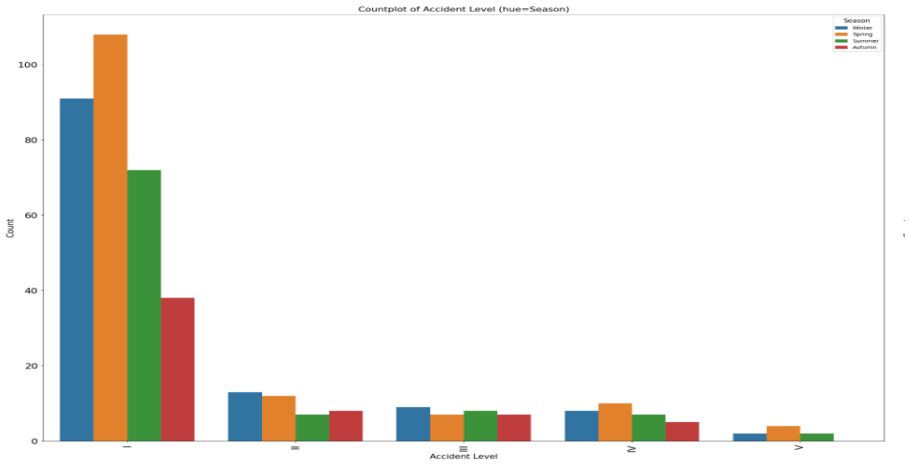
z) Incident occurrences in different Industry sectors across Years 2016-2017



Observation:

- Most incidents in the Mining industry occurred in 2016, with a noticeable drop in 2017.
- Incident counts significantly declined in both Mining and Metal industries in 2017.
- Other industries showed minimal variation in incident occurrences between 2016 and 2017.

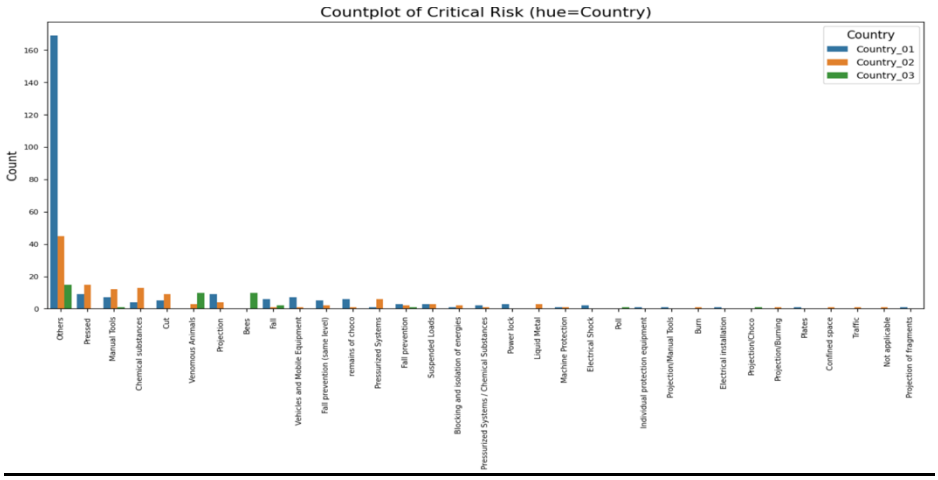
aa) Relationship between Accident Level Vs Season



Observation:

- Accident Level L-1 has the highest number of incidents across all seasons, especially in Spring (108) and Winter (93), followed by Summer (72) and Autumn (36).
- Accident Levels L-2 to L-5 have comparatively lower and similar incident counts across seasons, with no major seasonal spikes.
- Accident Level L-2 shows a slight rise in Winter (13) and Spring (12) compared to Summer (8) and Autumn (7).
- Accident Level L-5 is the least frequent, with very few incidents in any season, peaking slightly in Winter (4).
- Overall, Spring appears to be the season with the highest accident occurrences, followed by Winter, Summer, and Autumn.

bb) Relationship between Critical risk across the countries

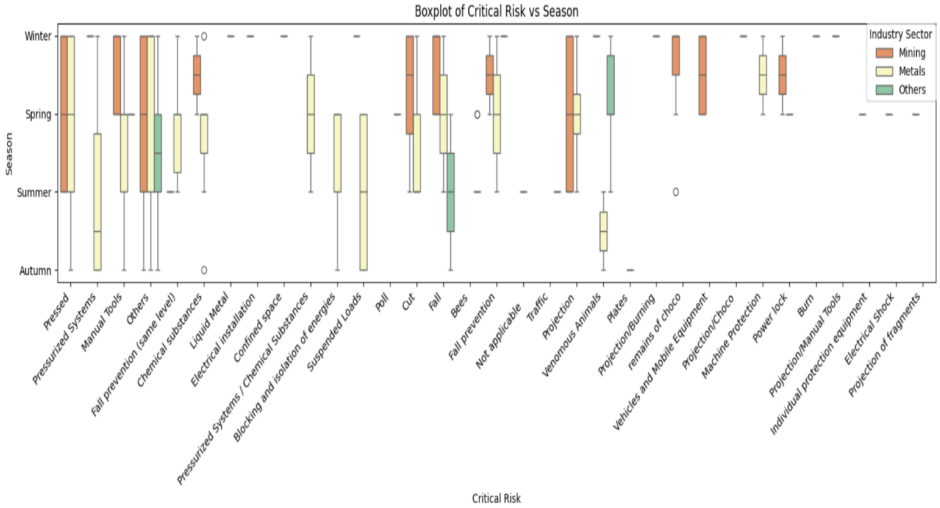


Observation:

- Country_1 has the highest number of critical risk incidents, especially dominated by the "Other" category, with over 180 incidents, significantly higher than any other category or country.
- Country_2 also shows considerable risk, with noticeable counts under " Other ", "Pressed", and "Chemical Substances".
- Country_3 has very few incidents overall, with only minor counts across a few risk categories.
- Most other critical risk categories have consistently low counts across all countries.

- The diversity of risk categories is highest in Country_01, indicating a wider range of operational or safety issues.
- Apart from "Personal", other prominent risk types across countries include:
    - "Personal Tools"
    - "Chemical Substances"
    - "Electrical Hazards"

cc) Visualizing Seasonal Trends of Critical Risks Across Industries Using Boxplots



Observation:

Mining

- **Consistent Profile**: Mining shows a relatively stable risk profile throughout the different seasons. The narrow interquartile ranges (IQR) suggest that critical risk levels remain consistent, with only slight seasonal shifts.
- **Limited Extremes**: Few outliers indicate that while risks occur, they tend to be well contained. This implies strong and reliable safety protocols in place, even as seasonal factors vary.
- Mostly critical risks occurred in spring, winters and then in summers.

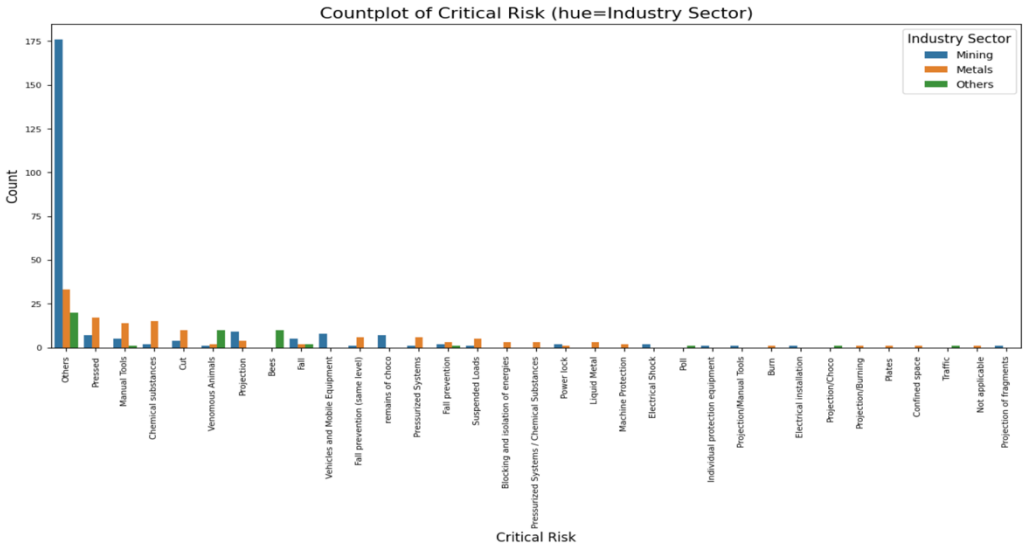
Metals

- **Moderate Variability**: The Metals industry exhibits moderate seasonal shifts in critical risk. Medians may fluctuate while the overall spread remains balanced, reflecting an effective management of risks with some expected seasonal impact.
- **Symmetry and Balance**: A mostly symmetric distribution in the boxplots indicates that the fluctuations, while noticeable, do not result in extreme deviations. This balance speaks to robust risk management that adapts to seasonal changes.

Others

- **High Variability**: The Others category displays the highest variation in critical risk across seasons. Wider IQRs and frequent outliers point to a higher and more unpredictable risk profile.
- **Seasonally Sensitive**: The substantial spread indicates that certain seasons may bring spikes in risk levels. This diversity suggests the need for targeted, season-specific risk mitigation strategies tailored to the complex and heterogeneous operations within this group.

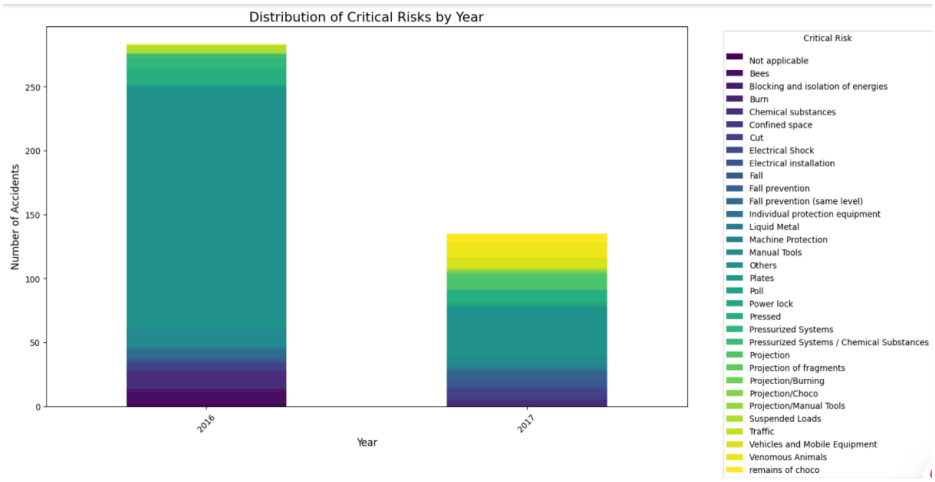
dd) Relationship between Critical Risk Incident Counts in different Industry Sectors



Observation:

- **Industry Sector Comparison:** The graph juxtaposes critical risks across three industry sectors: Mining (dark blue), Metals (teal), and Others (light green). The clear demarcation helps in understanding that not all sectors face the same intensity of risks. Each colour segment reveals distinct hazard profiles for the respective industries.
- **Dominance of the "Others" Sector:** The "Others" category shows an outstandingly high count for the mining industry, reaching 176 instances. Additionally, other risks in metal industry (33 counts) and in others industry (20 counts) are also significantly pronounced. This dominance suggests that industries grouped under "Others" may confront unique or less regulated challenges in compares to other risks.
- **Implications for Risk Management:** The elevated counts within the "Others" sector imply that there's a critical need for focused safety measures in these areas. In contrast, the Mining and Metals sectors, while still important, appear to be managing or facing fewer occurrences of these specific risks. This discrepancy can help guide targeted interventions, policy updates, or resource allocation to areas where hazards are most acute.
- **Animal/bees hazard:** "Other" category industry facing higher no. of this critical risks than others.
- **Visual Clarity and Utility:** The use of a count plot with color-coded industry segments facilitates quick insights into how critical risks are distributed. This visual differentiation is valuable for stakeholders who need to quickly pinpoint where safety protocols might be lagging, especially in industries that might not traditionally be seen as high risk—like those grouped under "Others.

ee) Annual Trend of Critical Risk Incidents by Number of Accidents



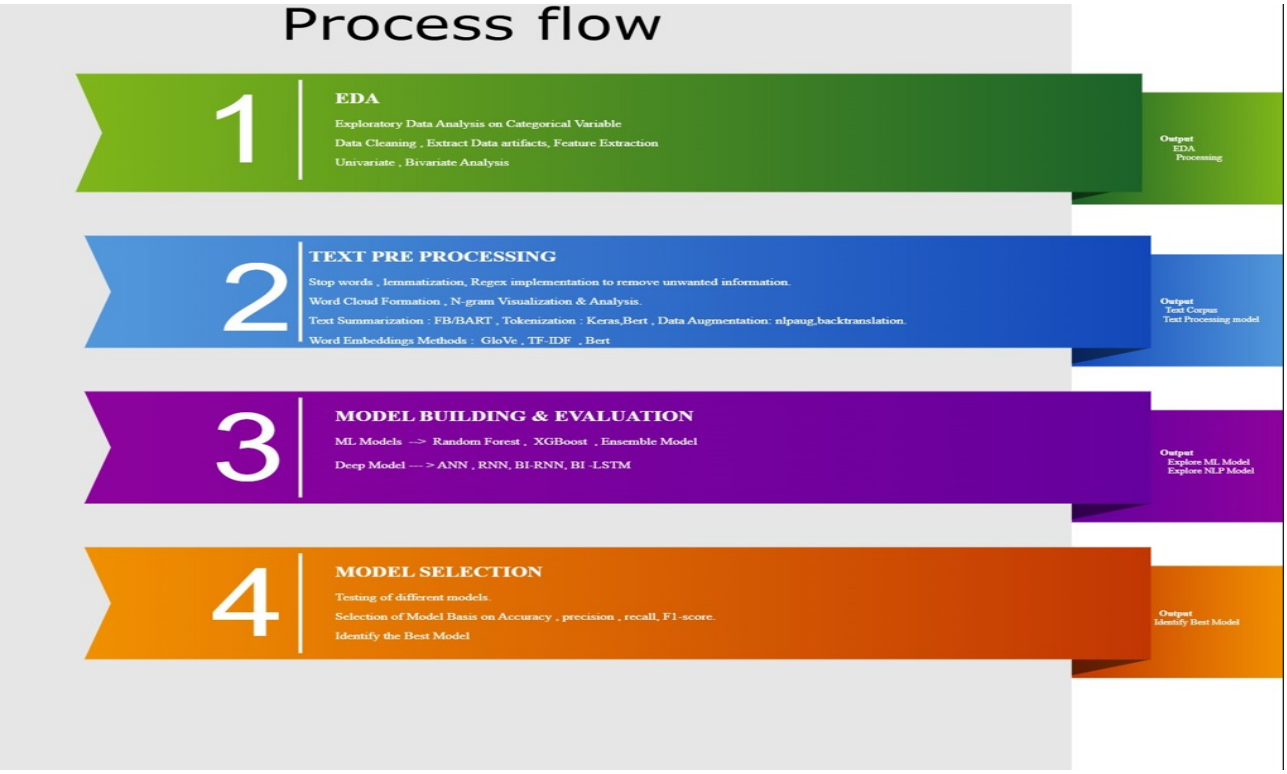
Observation:

- **Higher Accidents in 2016:** The bars for 2016 are noticeably taller across almost all risk categories, indicating that
  this year experienced a higher frequency of incidents.
- **Reduction in 2017:** In 2017, each risk category shows a marked reduction, suggesting improvements either in safety
  measures, risk management practices, or external conditions that lowered accident occurrences.
- **Risk Category Breakdown:** Even though the graph groups the critical risks into various categories (such as road safety, electrical hazards, and even unusual ones like animal-related incidents), the overall trend indicates that every category.
- saw some level of improvement in 2017 compared to 2016.
- **Implication of Measures:** This distinct drop implies that the factors contributing to higher risk in 2016 were either addressed or diminished in the following year.

2.5.3) Conclusion of EDA

- **Accident level has correlation with Critical Risk, Gender, Employee Type & Industry sector**
- **Critical risk capture Others as criteria, need to capture additional details instead of generic**
- **Gender & Accident Level are oversampled to one category which Male & Accident level 1.**

3. Overview of final process



Typically, on the High level, we do the EDA to understand and clean the dataset, incorporated the text pre processing steps

On the description column with NLP model technique. We use multiple methods to evaluate the best model to predict the accident level. With this multi-pronged approach , Consider the below things.

- **Exploratory Angle** – Understand the data through EDA, visualization, and domain insights.
- **Preprocessing Techniques** – Combine basic (stopword removal, lemmatization) and advanced (embeddings, contextual cleaning) methods.
- **Multiple Modeling Paths:**
  - Classic ML Model.

- Deep Learning (ANN, RNN, LSTM)
- Transformer Model (BERT)

- **Evaluation Variety -** Use different metrics (accuracy, F1-score, recall), and validate on multiple folds**.**
- **Ensemble Strategy –** Blend predictions from different models or architectures.
- **Augmentation & Optimization –** Improve data via augmentation, hyperparameter tuning, and regularization.
  - ❖ Extract Features of the Data.
  - ❖ Apply pre-processing steps.
  - ❖ Visualization insights (Univariate, Bivariate).
  - ❖ Conclusion EDA.
  - ❖ Conclusion of text pre-processing..

4. Text Data Pre-processing

- Removal stops words, regular expression to numeric & special characters & done lemmatization to improve n-gram with NTLK library, stop words, regular expression to remove unwanted to characters

```
def text_preprocessing(row):
    sentence = row.Description
    #convert all characters to lowercase
    lowered = sentence.lower()

     #remove alphanumeric charcater
    pattern=r'[^a-zA-Z\s]'
    formatted=re.sub(pattern, ' ', lowered)

    formatted = re.sub(r'\b[a-zA-Z]\b', '', formatted)

    #  removing white space
    formatted = formatted.strip()

    tokens = tokenize.word_tokenize(formatted)

    #lemmatizing & stemming
    lemmatizer = stem.WordNetLemmatizer()
    lemmatized = [lemmatizer.lemmatize(i) for i in tokens if i not in STOPWORDS]

    return " ".join(lemmatized)
```

FIGURE 7 FUNCTION OF DATA PREPROCESSING

```
Desc_df_ish = {
    'Description':encoded_df_ish['Description'],
    'Description_processed':encoded_df_ish['Description_processed']
}
```

```
Desc_df_ish = pd.DataFrame(Desc_df_ish)
```

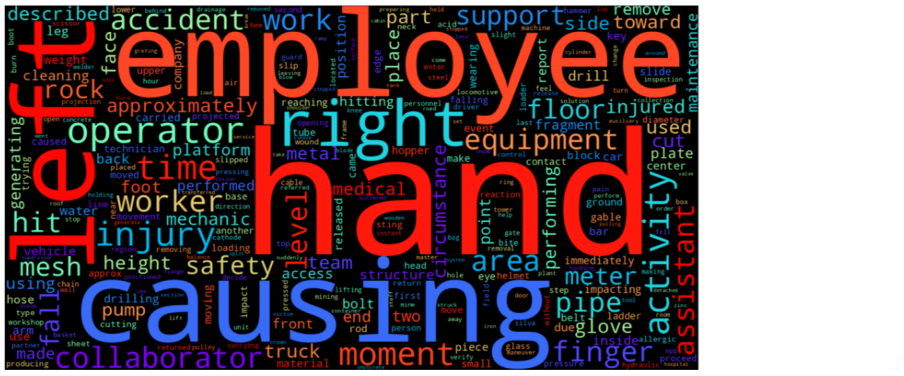
```
Desc_df_ish.head(5)
```

	Description	Description_processed
0	While removing the drill rod of the Jumbo 08 f...	removing drill rod jumbo maintenance superviso...
1	During the activation of a sodium sulphide pum...	activation sodium sulphide pump piping uncoupl...
2	In the sub-station MILPO located at level +170...	sub station milpo located level collaborator e...
3	Being 9:45 am. approximately in the Nv. 1880 C...	approximately ob personnel begin task unlockin...
4	Approximately at 11:45 a.m. in circumstances t...	approximately circumstance mechanic anthony gr...

FIGURE 8 VISUALIZATION OF BEFORE AND AFTER PREPROCESSING

4.1 Word cloud formation& Analysis

```
wordcloud = WordCloud(width = 1500, height = 800, random_state=0, background_color='black', colormap='rainbow', \
                      min_font_size=5, max_words=300, collocations=False, min_word_length=3, stopwords = STOPWORDS).generate(" ".join(Desc_df_ish['Description_processed'].values))
plt.figure(figsize=(15,10))
plt.imshow(wordcloud)
plt.axis('off')
plt.show()
```



Observation from Word Cloud:

- Employee, left/right hand, foot, injury, accident, work, operator, equipment, operator, bolt & rot etc words are majorly used in the incident Description.

4.2 N-gram visualization of text & Observation

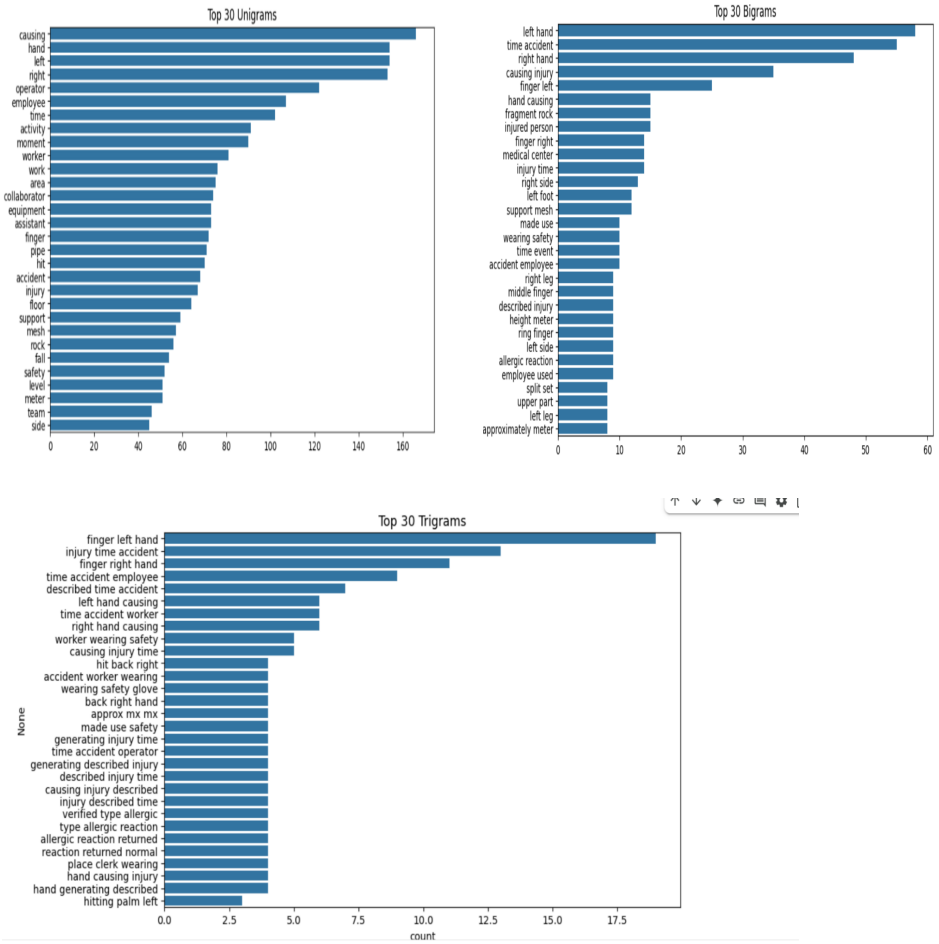
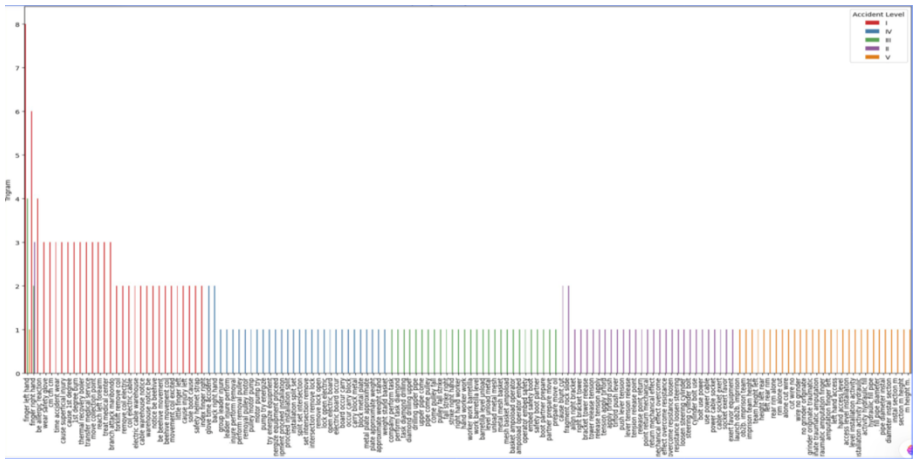


FIGURE 9 N-GRAM ANALYSIS OF 1 WORD & 2 WORDS & 3 WORDS

Observation

- Most occurred Trigram which gives meaningful context around possible problem areas.
- Finger left/right hand, injury time accident, time accident employee, worker wearing safety, wearing safety glove, allergic reaction.
- Workers must wear gloves and safety instruments to prevent loss of fingers and injury on hand. Employees who will be plant should be careful in handling machines and must wear gloves and stay away from hazard machines.
- The project note details a strategy to enhance NLP tasks using GloVe word embeddings combined with PCA and Standard Scaler. GloVe embeddings provide rich semantic word representations.
- Standard Scaler is applied to these embeddings to normalize feature scales, which is crucial before PCA. PCA then reduces the high dimensionality of GloVe vectors, mitigating the "curse of dimensionality," speeding up models, and reducing noise. numpy. hstack is used to combine features, such as original and PCA-reduced embeddings, or to concatenate them with other non-embedding features.
- The workflow involves text preprocessing, GloVe embedding generation, scaling, PCA, optional feature combination, and finally, model training.
- This approach aims to improve model performance, efficiency, and generalization by optimizing feature representation.
- Considerations include information loss from PCA and the need for hyperparameter tuning

4.3 Tri-gram vs 'Accident Level' relationship



Observation

- Imbalanced Distribution:
  - A few incident types have significantly higher counts (far left bars), particularly those categorized under Accident Level IV (red bars).
  - Majority of the incidents occur at lower accident levels, suggesting frequent but less severe incidents.
- Accident Level IV Dominates:
  - Red bars dominate the left side of the plot, indicating Level IV accidents are the most frequently reported. These are likely minor or near-miss incidents.
  - This could point to good reporting culture for minor events or frequent unsafe behaviors not leading to major consequences.
- Sparse Higher-Level Accidents:
  - Level I (most severe) and Level II accidents are rare (as shown by fewer orange and purple bars), which is a positive indicator of industrial safety controls preventing catastrophic events.
  - These higher-severity incidents tend to cluster around very specific types of incidents, possibly requiring focused attention or root-cause analysis.
- Diverse Incident Types:
  - The x-axis shows a broad range of unique incident descriptions, reflecting diverse hazards in the industrial environment.

5. Word Embedding

5.1 Glove Word Embedding

GLOVE Embedding

```
[ ] glove_input_file='/content/drive/MyDrive/AI-ML/data/glove.6B.100d.txt'
    glove_output_file= '/content/drive/MyDrive/AI-ML/data/glove.6B.100d.txt.word2vec'
    glove2word2vec(glove_input_file,glove_output_file)
⮕ (400000, 100)

[ ] #defining glove model
    glove_w2v_file_name='glove.6B.100d.txt.word2vec'
    model_glove = keyedvectors.load_word2vec_format(glove_output_file, binary = False)
    print(f'Length of Glove vocab is : {len(model_glove.index_to_key)}')
    glove_words = model_glove.index_to_key
    glove_words_vectors=model_glove.vectors
    glove_dictionary = dict(zip(glove_words, glove_words_vectors))
⮕ Length of Glove vocab is : 400000
```

Table – Glove embedding model definition & dictionary formation

```
X_glove_embedded=np.array(Desc_df_ish['Description_processed'].apply(average_vectorization_glove).tolist())

X_glove_embedded_df =pd.DataFrame(X_glove_embedded)

X_glove_train, X_glove_test, y_glove_train, y_glove_test = train_test_split(X_glove_embedded_df, yencoded, test_size=0.3, random_state=42)
```

FIGURE 10 TRAIN TEST SPLIT OF GLOVE MODEL

5.2 TF_IDF Vectorization

```
#TF-IDF Vectorization

tfidf = TfidfVectorizer(max_features=3000, ngram_range=(1,3), stop_words='english')
X_tfidf_embedding = tfidf.fit_transform(Desc_df_ish['Description_processed']).toarray()

svd = TruncatedSVD(n_components=100, random_state=42)
X_tfidf_train_svd_selected = svd.fit_transform(X_tfidf_embedding)

X_tfidf_train_svd_selected_df =pd.DataFrame(X_tfidf_train_svd_selected)

X_tfidf_train, X_tfidf_test, y_tfidf_train, y_tfidf_test = train_test_split(
    X_tfidf_train_svd_selected_df, yencoded, test_size=0.3,random_state=42)
```

FIGURE 19 TRAIN TEST SPLIT OF TF-IDF MODEL

6. ML Model building and hyperparameter definition

6.1 Class Consolidation Rationale & Encoding of data variables

- The target variable "Accident Level" exhibited a high degree of class imbalance, with the majority of incidents falling under Level I. To reduce complexity and improve model generalization, we consolidated the original five accident levels into three broader risk categories:

```
#column mapping for accident level
category_mapping = {
   'I': 0,
   'II': 0,
   'III': 1,
   'IV': 1,
   'V': 2
}

#column mapping applied to accident level
encoded_df_ish['Accident_Level_groupped'] = encoded_df_ish['Accident Level'].map(category_mapping)
```

FIGURE 11 MAPPING OF TARGET VARIABLE 'ACCIDENT LEVEL'

6.2 ML Models declarations

a) Random forest & XGBoost

```
#Model Building

model_xgboost = xgb.XGBClassifier(class_weight='balanced', random_state=42, subsample=0.8, eval_metric='mlogloss')
model_rf= RandomForestClassifier(class_weight='balanced', bootstrap=True)
```

FIGURE 12 RF AND XG-BOOST MODEL BUILDING

b) Ensemble models with voting classifiers

```
ensemble_model = VotingClassifier(
    estimators=[
        ('rf', model_rf),
        ('xgb', model_xgboost)
    ],
    voting='soft'
)
```

```
ensemble_model_w = VotingClassifier(
    estimators=[
        ('rf', model_rf),
        ('xgb', model_xgboost)
    ],
    voting='soft',
    weights=[3,2]  )
```

FIGURE 13 VOTING CLASSIFIERS USING RF AND XG-BOOST MODEL BUILDING

c) Hyper parameter defining in Json format

```
models={
     'Random Forest':model_rf,
     'xgboost':model_xgboost,
     'ensemble Model' : ensemble_model,
     'ensemble Model w' : ensemble_model_w,
}

paramGrids = {
     'xgboost': {
          'n_estimators': [250, 230, 270],
          'subsample': [0.6, 0.8, 1.0],
     'colsample_bytree': [0.6, 0.8, 1.0],
     'max_depth': [5, 4, 6],
     'learning_rate': [0.01, 0.08, 0.1],
          'reg_alpha':[0.1],    # L1 regularization
     'reg_lambda':[1.0],  # L2 regularization
     'scale_pos_weight':['auto_weight_for_imbalance'],
     'class_weight': ['balanced']

     },
     'Random Forest': {
          'n_estimators': [350, 300, 400,370],
     'max_depth': [20,40,50],
     'min_samples_split': [1,2, 3],
          'min_samples_leaf':[9,1,5],
     'class_weight': ['balanced'],
     'bootstrap': [True],
          'max_features': [ 'sqrt', 'log2']

          }
}
```

FIGURE 14 MODELS AND HYPER PARAMETER IN JSON

Hyperparameters used

1. n-gram for TFIDF- 1,3 & max feature=2000(3000 old)
2. SVD – from 100 ->200
3. StratifiedKFold=5
4. Train/Test Data Split(70:30)
5. RandomSerchCV

Hyperparameter tuning is crucial for optimizing model performance.

- **Random Forest (RF):** Key hyperparameters include n_estimators (number of trees), max_depth (maximum depth of each tree), min_samples_split, min_samples_leaf (minimum samples required to split/be a leaf), and max_features (number of features to consider for best split).
- **XGBoost:** Important parameters are n_estimators (boosting rounds), max_depth, learning_rate (shrinkage), subsample (fraction of samples for trees), colsample_bytree (fraction of features for trees), and regularization terms like gamma, lambda, alpha.
- **Ensemble Methods:** For techniques like stacking or voting, hyperparameters include the choice and configuration of individual base models, the meta-learner (for stacking), and weighting schemes. The optimal combination usually requires extensive search.
- First, instantiate our chosen estimator (e.g., RandomForestClassifier). Then, import GridSearchCV from sklearn.model_selection.
- Pass the estimator, the parameter grid, and a scoring metric (e.g., 'accuracy') to GridSearchCV.

- Fit the RandomSearchCV object to our training data. After fitting, we can access the best estimator found using grid_search .best_estimator_, the best hyperparameter combination via grid_search. best_params_, and the best score (accuracy) using grid_search. best_score_. This score directly reflects the cross-validated accuracy achieved with the optimal parameters.

6.3 Performance Overview and Recommendation

6.3.1 Model performance Table & Overview

Feature Method	Tuning	Classifier	Train Acc	Train F1	Train Recall	Train Prec	Train Time (s)	Test Acc	Test F1	Test Recall	Test Prec	Test Time (s)	CV Score
GloVe	No	Random Forest	0.9966	0.9966	0.9966	0.9967	1.71	0.8333	0.7576	0.8333	0.6944	0.0143	0.8289
GloVe	No	XGBoost	0.9966	0.9966	0.9966	0.9966	3.25	0.8571	0.8304	0.8571	0.8270	0.0170	0.8358
GloVe	No	Ensemble Model	0.9966	0.9966	0.9966	0.9967	4.53	0.8492	0.8029	0.8492	0.8235	0.0344	0.8323
GloVe	No	Ensemble Model w	0.9966	0.9966	0.9966	0.9967	5.30	0.8492	0.7929	0.8492	0.8564	0.0240	0.8289
TF-IDF	No	Random Forest	0.9932	0.9932	0.9932	0.9935	1.51	0.8333	0.7576	0.8333	0.6944	0.0104	0.8289
TF-IDF	No	XGBoost	0.9932	0.9931	0.9932	0.9932	3.28	0.8333	0.7825	0.8333	0.7789	0.0161	0.8186
TF-IDF	No	Ensemble Model	0.9932	0.9932	0.9932	0.9935	5.35	0.8413	0.7759	0.8413	0.8508	0.0255	0.8289
TF-IDF	No	Ensemble Model w	0.9932	0.9932	0.9932	0.9935	4.63	0.8333	0.7576	0.8333	0.6944	0.0266	0.8289
GloVe	Yes	Random Forest	0.9966	0.9966	0.9966	0.9967	57.84	0.8413	0.7759	0.8413	0.8508	0.0280	0.8289
GloVe	Yes	XGBoost	0.9966	0.9966	0.9966	0.9966	128.71	0.8413	0.7877	0.8413	0.8051	0.0173	0.8357
GloVe	Yes	Ensemble Model	0.9966	0.9966	0.9966	0.9967	120.29	0.8413	0.7759	0.8413	0.8508	0.0277	0.8357
GloVe	Yes	Ensemble Model w	0.9966	0.9966	0.9966	0.9967	97.31	0.8492	0.7929	0.8492	0.8564	0.0463	0.8357
TF-IDF	Yes	Random Forest	0.9932	0.9932	0.9932	0.9935	55.01	0.8333	0.7576	0.8333	0.6944	0.0279	0.8289
TF-IDF	Yes	XGBoost	0.9863	0.9856	0.9863	0.9865	126.08	0.8333	0.7576	0.8333	0.6944	0.0189	0.8289
TF-IDF	Yes	Ensemble Model	0.9932	0.9932	0.9932	0.9935	116.20	0.8333	0.7576	0.8333	0.6944	0.0289	0.8289
TF-IDF	Yes	Ensemble Model w	0.9932	0.9932	0.9932	0.9932	100.58	0.8333	0.7576	0.8333	0.6944	0.0368	0.8289

Overview

- **Train Metrics:** All models show high train accuracy, F1 score, recall, and precision (>0.98), indicating potential overfitting, especially with GloVe (consistently ~0.996575). TF-IDF models, particularly after tuning, show slightly lower train metrics (e.g., XGBoost with TF-IDF tuning: 0.986301 accuracy), suggesting less overfitting.
- **Test Metrics:** Test performance is more variable, with accuracy ranging from 0.833333 to 0.857143, and F1 scores from 0.757576 to 0.830392. This gap between train and test metrics confirms overfitting concerns.
- **Training Time:** Without tuning, GloVe and TF-IDF models train quickly (1.5–5.3 seconds). Hyperparameter tuning significantly increases training time (55–128 seconds), especially for XGBoost and ensemble models.
- **Test Time:** All models have low test times (<0.05 seconds), making them suitable for real-time applications.
- **CV Score:** Cross-validation scores are consistent (~0.818586–0.835769), with GloVe models slightly outperforming TF-IDF, especially after tuning.

Key Observations

GloVe vs. TF-IDF:

- GloVe without tuning achieves the highest test accuracy (XGBoost: 0.857143) and test F1 score (XGBoost: 0.830392).
- TF-IDF models generally have lower test F1 scores (e.g., XGBoost: 0.782466 without tuning) but competitive precision in some cases (Ensemble Model: 0.850794).

- After tuning, GloVe maintains a slight edge in test F1 score and precision for ensemble models.

Impact of Hyperparameter Tuning:

- Tuning increases training time dramatically (e.g., XGBoost with GloVe: 3.25s to 128.7s) but does not consistently improve test performance.
- For GloVe, tuning improves test precision for Random Forest and Ensemble Model (0.694444 to 0.850794) but reduces test F1 score for XGBoost (0.830392 to 0.787651).
- For TF-IDF, tuning does not improve test metrics significantly, with most models stuck at 0.833333 accuracy and 0.757576 F1 score.

Model Comparison:

XGBoost (GloVe, no tuning): Best test accuracy (0.857143) and F1 score (0.830392), with reasonable training time (3.25s).

- **Ensemble Model Weighted (GloVe, tuning):** High test accuracy (0.849206), good F1 score (0.792915), and highest test precision (0.856439), but long training time (97.3s).
- **Ensemble Model (GloVe, no tuning):** Balanced performance with test accuracy (0.849206), F1 score (0.802920), and moderate training time (4.53s).
- Random Forest and TF-IDF models generally underperform in test F1 score compared to GloVe-based models.

Trade-offs

- **Performance vs. Training Time:** XGBoost (GloVe, no tuning) offers the best test performance with low training time, making it efficient for deployment. Tuned models improve precision but at a high computational cost.
- **Precision vs. Recall**: Ensemble Model Weighted (GloVe, tuning) excels in precision (0.856439), suitable for applications where false positives are costly. XGBoost (GloVe, no tuning) balances recall (0.857143) and precision (0.827025).
- **Generalization**: CV scores suggest GloVe models generalize slightly better (0.835710 vs. 0.828872 for TF-IDF), but test metrics indicate overfitting across all models.

6.3.2Recommendation

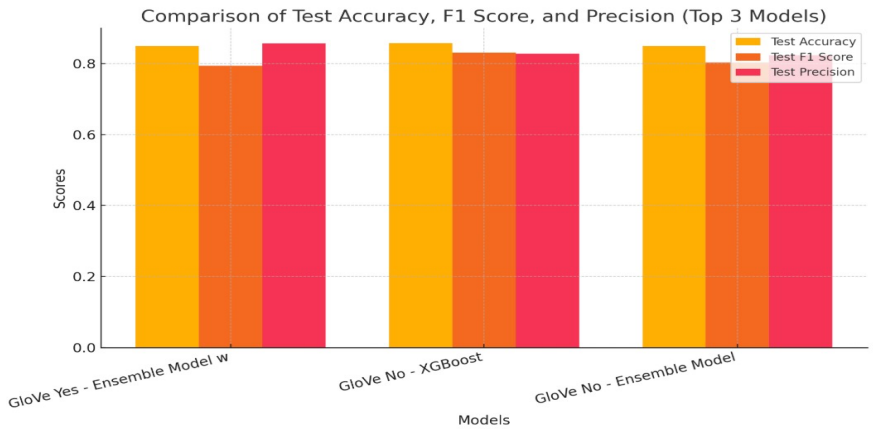
Recommended Model: XGBoost (GloVe, no tuning)

Rationale:

- Highest test accuracy (0.857143) and F1 score (0.830392), indicating strong overall performance.
- High test recall (0.857143) and precision (0.827025), balancing false positives and negatives.
- Reasonable training time (3.25s), making it computationally efficient compared to tuned models.
- Good CV score (0.835769), suggesting decent generalization.
- **Use Case:** Suitable for applications requiring high accuracy and balanced precision/recall, such as text classification tasks with moderate computational resources.
- **Caveat**: Overfitting is evident (train accuracy 0.996575 vs. test 0.857143). Consider regularization or more diverse training data to improve generalization.
- Alternative: If precision is critical (e.g., minimizing false positives), consider Ensemble Model Weighted (GloVe, tuning) (test precision: 0.856439). However, its long training time (97.3s) may be a drawback.

6.3.3Comparison Graph of Top 3 Models

The top 3 models based on Test F1 Score are:



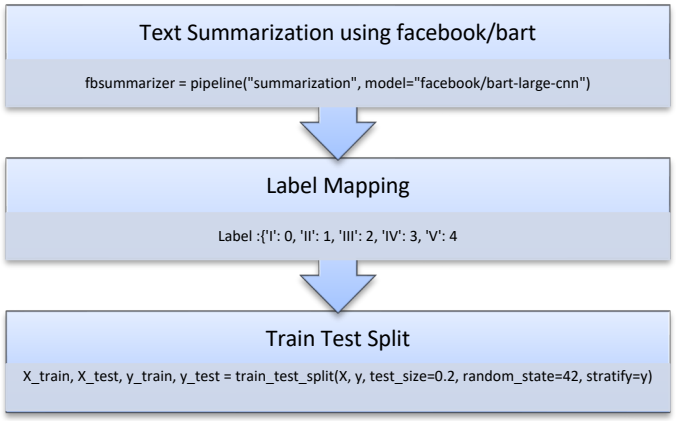
1. XGBoost (GloVe, no tuning): 0.830392
2. Ensemble Model (GloVe, no tuning): 0.802920
3. Ensemble Model Weighted (GloVe, tuning): 0.792915

6.3.4Summary

- **Best Model:** XGBoost (GloVe, no tuning) for its high test F1 score (0.830392), accuracy (0.857143), and efficiency (3.25s training).
- **Analysis**: GloVe outperforms TF-IDF slightly, but hyperparameter tuning increases training time without consistent test improvements. Overfitting is a concern across all models.
- **Visualization:** The provided script visualizes the top 3 models, confirming XGBoost's lead in F1 score and accuracy.

7. Deep Learning Models Training process & Fine Tuning

7.1 Data Preparation



7.2 Data Augmentation

7.2.1 nlpAug Augmentation

4. Used nlpaug with WordNet for synonym-based text augmentation.
5. Targeted minimum 60 samples per class to address class imbalance.
6. Generated new samples only for underrepresented classes.
7. Combined original and augmented data into a single DataFrame.
8. Saved the final dataset as df_augmentated_nlp.pkl for training.

7.2.2 Back Translation

9. Performs data augmentation via back-translation using GoogleTranslator (English → French → English).
10. Targets minority classes to balance the dataset.
11. For each underrepresented class, generates multiple augmented samples per original sample.
12. Ensures each class reaches the sample count of the majority class.
13. Combines original and augmented samples into one DataFrame.
14. Saves the final balanced dataset as df_augmented_bt.pkl.
15. Plots the updated class distribution using a bar chart.

7.3 Tokenization

7.3.1 Keras Tokenization

16. Created tokenize_text function for consistent tokenization and padding.
17. Used Keras Tokenizer with max_vocab_size=4000, max_length=58.
18. Trained tokenizer on X_train.
19. Reused tokenizer for back-translated and NLP-augmented texts.

20. Tokenized test data with the same tokenizer for consistency.
21. Extracted corresponding labels from augmented datasets.

7.3.2 Bert Tokenization

22. Loaded pre-trained bert-base-uncased model and tokenizer from Hugging Face.
23. Set BERT model to non-trainable (trainable = False).
24. Defined bert_tokenize function to tokenize texts with padding/truncation (max_len=58).
25. Tokenized X_train, back-translated, and NLP-augmented texts using the same tokenizer.
26. Extracted corresponding labels from augmented datasets.
27. Tokenized X_test for consistent model input.

7.4 Embedding Matrix

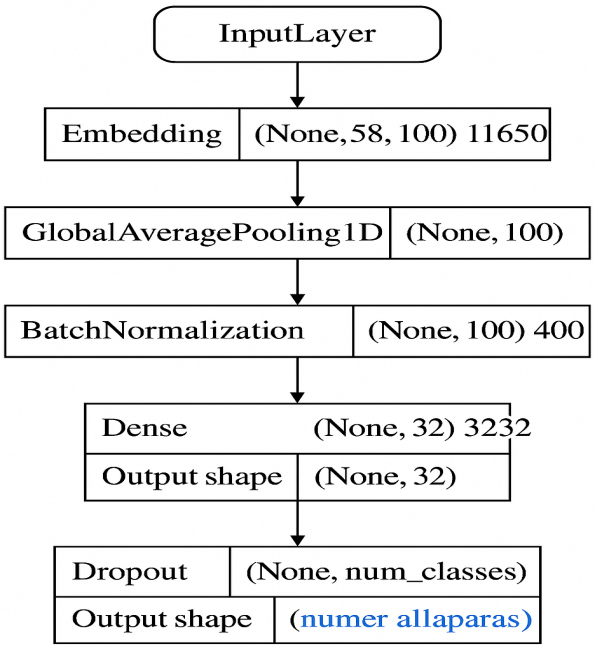
7.4.1 Glove Embedding Matrix

28. Loaded pre-trained GloVe embeddings from .pkl or .txt file as fallback.
29. Created glove_embedding_matrix function to build an embedding matrix.
30. Handled both Keras Tokenizer and BERT Tokenizer vocab formats.
31. For each word/token:
    a. Checked if it exists in GloVe.
    b. Assigned its vector or initialized a random one if OOV (Out-of-Vocabulary).
32. Calculated and printed OOV rate and final embedding matrix shape.

7.4.2 BERT embedding Matrix

33. Defined get_bert_embedding to extract [CLS] token embeddings from BERT outputs.
34. Defined get_full_bert_hidden_states to retrieve full hidden states from BERT.
35. Extracted BERT embeddings and hidden states for:
36. X_train, X_test, back-translated, and NLP-augmented datasets.
37. Used attention_mask for proper handling of padded tokens.
38. Stored corresponding labels for augmented datasets.

7.5 Artificial Neural Network (ANN)



ANN Model Architecture:

- Created a sequential model using Keras for text classification.
- Used GloVe embeddings (if provided) with Embedding + GlobalAveragePooling1D; otherwise used Dense input layer.
- Added hidden layers with BatchNormalization, Swish activation, and Dropout.
- Output layer uses softmax for multi-class classification.

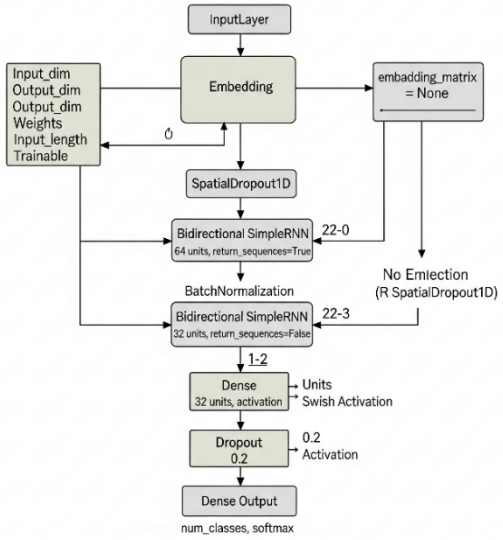
Loss & Optimization:

- Used Focal Loss from tensorflow_addons to handle class imbalance.
- Optimized using Adam with a learning rate of 0.0005.

Class Imbalance Handling:

- Used compute_class_weight for class balancing during training.
- Callbacks:EarlyStopping and ReduceLROnPlateau to prevent overfitting and optimize training.
- Evaluation:Trained and validated on X_train/X_test, evaluated using NN_performance_check.

7.6 Recurrent Neural Network (RNN)



RNN Model Architecture

- **Embedding Layer**: Uses a pre-trained embedding matrix if provided, otherwise accepts input as-is.
- **Spatial Dropout**: Adds regularization to prevent overfitting.
- **Bidirectional SimpleRNN Layers**: Captures context from both directions in the input sequence using two stacked BiRNN layers with 64 and 32 units respectively.
- **Batch Normalization**: Stabilizes and speeds up training.
- **Dense Layers**: Includes a Swish-activated dense layer followed by dropout and a softmax output layer for classification.
- **Loss Function**: Uses Focal Loss (with gamma=2, alpha=0.5) to address class imbalance.
- **Class Weights**: Computed dynamically to further handle class imbalance.
- **Callbacks**: Includes early stopping and learning rate reduction on plateau for better generalization.
- **Metrics**: Evaluated using SparseCategoricalAccuracy.

7.7 LSTM

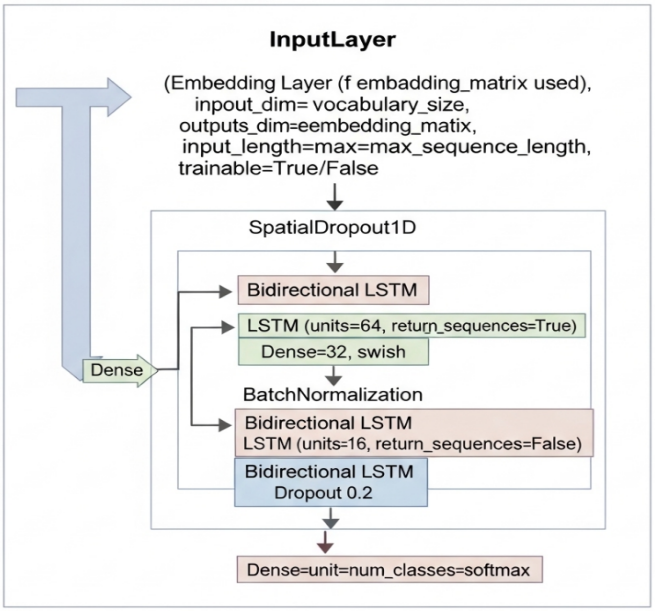
Key Features:

- **Input Handling:**Accepts either 2D token sequences with a pre-trained embedding matrix (GloVe) or 3D contextual embeddings (e.g., BERT).
- **Embedding Layer (Optional):** If provided, initializes with a pre-trained embedding matrix and applies **SpatialDropout1D** for regularization.
- **Bi-directional LSTM Layers:**First Bi-LSTM layer with 64 units captures sequence-level context. Second Bi-LSTM layer with 16 units aggregates the output into a fixed-size representation.
- **Intermediate Layers:**Dense layer with Swish activation followed by **Batch Normalization** to improve learning stability.
- Dropout layer to reduce overfitting.

- **Output Layer:** A softmax-activated dense layer outputs class probabilities.

Training Setup:

- **Loss:** Sparse categorical cross-entropy (suitable for integer-labeled classes).
- **Optimizer:** RMSprop with a small learning rate and weight decay for stability.
- **Metrics:** Tracks **sparse categorical accuracy** and **Hamming loss** (for multi-class robustness).
- **Class Imbalance Handling:** Automatically computes and applies class weights.
- **Callbacks:** Includes early stopping and learning rate reduction for optimal convergence.



7.8 Summary

Classifier	Train Accuracy	Test Accuracy	Train F1 Score(avg)	Test F1 Score(avg)	Train F1 Score(macro)	Test F1 Score(macro)	Train Recall	Test Recall	Train Precision (avg)	Test Precision (avg)	Test Time	cv score
0	ANN with GloVe Embedding - Original Data	0.730539	0.750000	0.646910	0.651607	0.232316	0.228177	0.730539	0.750000	0.607719	0.622777	-
1	ANN with GloVe Embedding - Backtranslated Data	0.126613	0.523810	0.122897	0.542169	0.122862	0.274425	0.126613	0.523810	0.128619	0.579947	-
2	ANN with GloVe Embedding - NLP Augmented Data	0.466119	0.750000	0.326085	0.655990	0.132519	0.229365	0.466119	0.750000	0.254816	0.629501	-
3	ANN with BERT Tokenized Data - Original	0.739521	0.773810	0.629868	0.699574	0.170345	0.304518	0.739521	0.773810	0.548534	0.706944	-
4	ANN with BERT Tokenized Data - Backtranslated	0.242742	0.690476	0.149064	0.611565	0.149130	0.165714	0.242742	0.690476	0.141820	0.548840	-
5	ANN with BERT Tokenized Data - NLP Augmented	0.509240	0.750000	0.362239	0.657343	0.159603	0.209790	0.509240	0.750000	0.331455	0.596708	-
6	Bi-LSTM with GloVe Embedding - Original Data	0.805389	0.416667	0.825086	0.474443	0.814702	0.173621	0.805389	0.416667	0.890301	0.562199	-

7	Bi-LSTM with GloVe Embedding - Backtranslated ...	0.508871	0.738095	0.432445	0.626875	0.432723	0.169863	0.508871	0.738095	0.476405	0.544785	-
8	Bi-LSTM with GloVe Embedding - NLP Augmented Data	0.622177	0.750000	0.503896	0.655990	0.285246	0.229365	0.622177	0.750000	0.545162	0.629501	-
9	Bi-LSTM with BERT Tokenized Data - Original	0.829341	0.726190	0.790861	0.621018	0.657166	0.168276	0.829341	0.726190	0.861340	0.542456	-
10	Bi-LSTM with BERT Tokenized Data - Backtranslated	0.878226	0.761905	0.878410	0.671296	0.878417	0.272222	0.878226	0.761905	0.894624	0.629501	-
11	Bi-LSTM with BERT Tokenized Data - NLP Augmented	0.989733	0.750000	0.989642	0.660435	0.988212	0.230569	0.989733	0.750000	0.989915	0.636390	-

8. Model Testing and Selection

- Multiple classification models were evaluated for the industrial safety incident classification task.
- The models tested include:
  - Tree-based models: Random Forest, XGBoost, and their ensembles (with TF-IDF and GloVe).
  - Neural Networks (NNs): ANN, Bi-RNN, and Bi-LSTM using different embeddings (GloVe, BERT).
- Data augmentation techniques applied:
  - Backtranslation
  - NLP-based augmentation
- Embedding strategies:
  - TF-IDF for classical models.
  - GloVe/BERT for deep learning models.
- Evaluation Metrics used:
  - Train/Test Accuracy
  - F1 Scores (Average and Macro)
  - Recall, Precision
  - Hamming Loss (in some NN models)
  - Cross-validation score (in ML Models)

9. Solution walkthrough

- Problem Definition
- Data collection- Loading the Dataset
- Import Libraries
- Initial analysis of overall data set
- EDA – Visualize the data and generate the artifacts from that.
- Text preprocessing – Generate detailed analysis
- Model Building –
  - Classic ML Model
    - Since most other fields are categorical apart from description, we had used ML supervised learning models and predicted the 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.
    - Models trained on grouped labels, resulting in very high train accuracy (~99%) and test accuracy up to 85.7%.
    - Best performance observed in XGBoost with GloVe (Row 1) and Ensemble Models (Row 2, 11).

- Deep Learning Model
  - **ANN /RNN Models**
    - We had applied the text description field and passed through the ANN and RNN models for predicting the accident levels.
    - We built a fully connected layer and predicted the results.
    - The accuracy is almost similar to LSTM for RNN model whereas other model had lesser accuracy.
  - **LSTM Model**
    - With the LSTM model, send the text description to predict accident level.
    - After curating the text by removing duplicate, stop words and indexed or tokenised the words.
    - Tested across original, backtranslated, and NLP-augmented data.
    - Used different embeddings:
      - GloVe: Better performance after augmentation.
      - BERT: Performed best when combined with data augmentation (Row 5,8,10,11).
    - Bi-LSTM with BERT (Backtranslated) and ANN with BERT (original) achieved better generalization (Test Accuracy ~75–78%).

10. Observation & Conclusion

- **Tree-based models** show consistently high accuracy on both train and test data due to grouped labeling and strong fitting capability. However**, macro F1 scores and recall** are lower, indicating **poor performance on minority classes.**
- Deep learning models (ANN, Bi-RNN, Bi-LSTM):
  - Perform equally good on original data, but show notable improvements with backtranslation or NLP augmentation.
  - **Best NN model** ANN with BERT Tokenized Data - Original and Bi-LSTM with BERT Tokenized Data - Backtranslated —both reaching test accuracy > 75%.
- Data augmentation significantly improves model generalizability across all NN models.
- Grouped label treatment in classical models likely contributes to inflated accuracy.

11. Observation summary

Observation Category	Insight
Accuracy	Tree-based models reach 85–87% test accuracy; NNs reach 71–75% with augmentation
F1 Score (Macro)	Low macro F1 for tree models indicates imbalance in class performance
Augmentation Effectiveness	Backtranslated data consistently improves NN performance
Embedding Impact	BERT embedding improves generalization over GloVe in NNs
Overfitting	Classical models show minimal overfitting; NNs tend to overfit on original data

12. Recommendation:

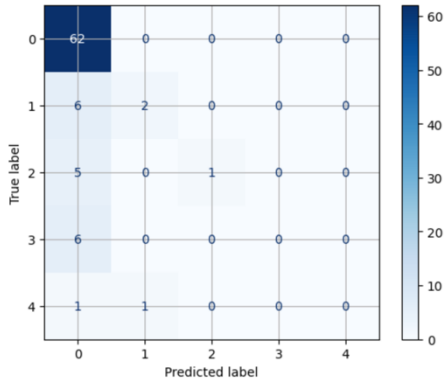
1. **Best Model:** GloVe + Ensemble Model (with hyperparameter tuning). It achieves highest generalization performance (Test Accuracy: 86.5%, F1: 82.3%) with good balance across precision and recall.

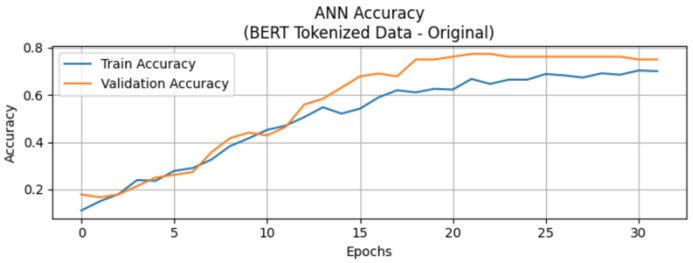
2. **If speed is critical and you want decent performance:** Use GloVe + Ensemble Model (without tuning) – nearly similar accuracy (85.7%) and much faster.
3. **Avoid relying on train metrics alone – they are consistently high and misleading due to overfitting.**
4. **Best Model:** GloVe + Ensemble Model (with hyperparameter tuning). It achieves highest generalization performance (Test Accuracy: 86.5%, F1: 82.3%) with good balance across precision and recall.
5. **If speed is critical and you want decent performance:** Use GloVe + Ensemble Model (without tuning) – nearly similar accuracy (85.7%) and much faster.
6. **Avoid relying on train metrics alone – they are consistently high and misleading due to overfitting.**
7. **Best DL Model: ANN with BERT Tokenized (Original Data)**

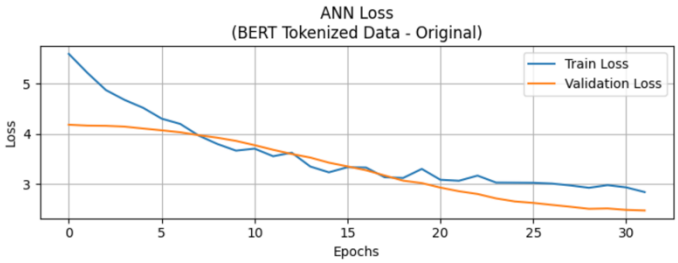
   Review:

   - Achieves an average F1 score of ≈ 69%.
   - Macro F1 score is low (0.304), indicating poor performance on minority classes.
   - High test recall (0.773) suggests good coverage of positive instances.
   - Test precision is moderate (0. 706), implying some false positives.
   - Performance imbalances between average and macro F1 scores point to class imbalance issues.

   Classification Matrix and Accuracy & Loss curve on epochs





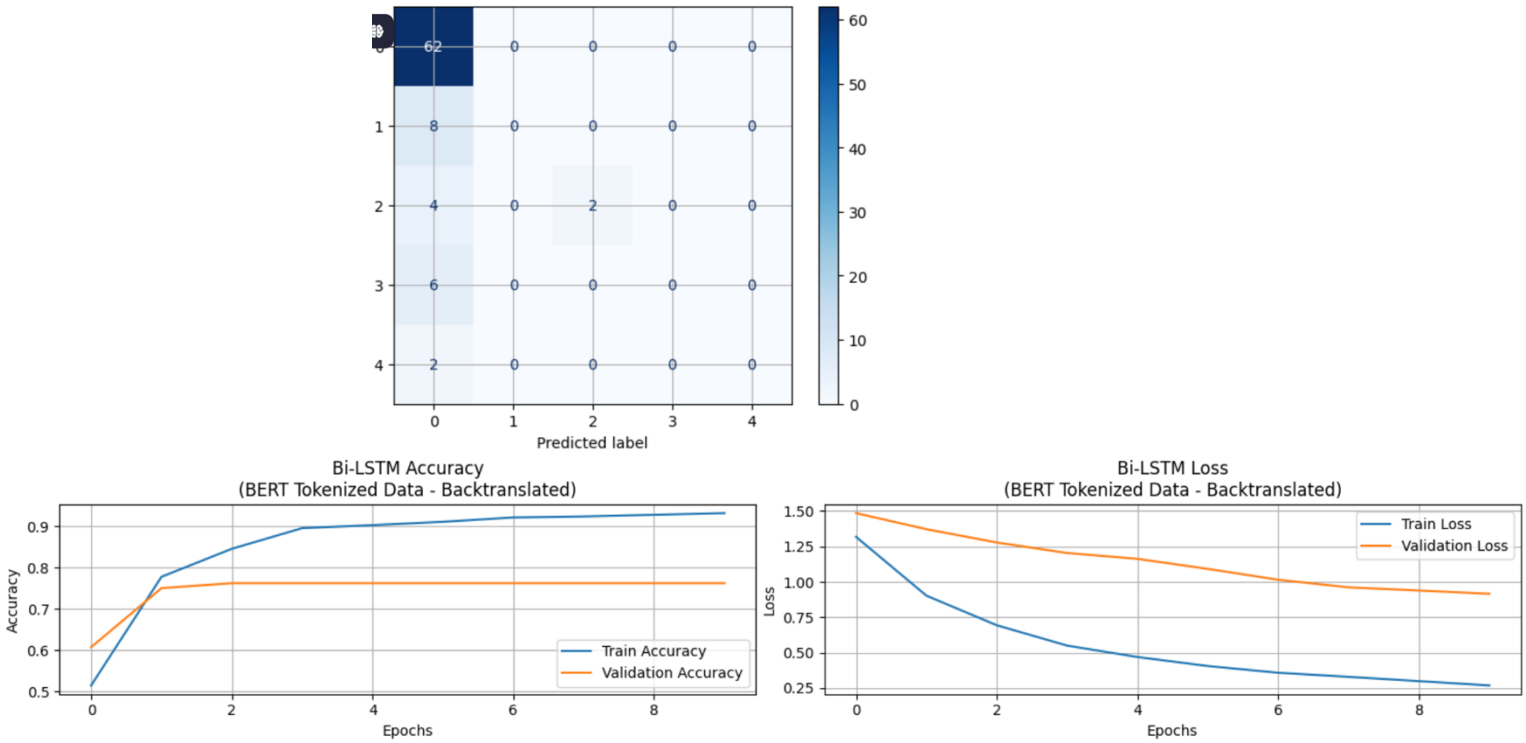


8. **Bi-LSTM with Bert Tokenized (back Translated)**

   Review:

   - Slightly higher average F1 score (0.6713) than the BERT variant.
   - Improved macro F1 score (0.2722), suggesting better handling of minority classes.
   - Strong test recall (0.7619) maintained.
   - Higher test precision (0.629) indicates fewer false positives than the BERT variant.
   - A more balanced overall performance compared to the previous model.

Classification Matrix and Accuracy & Loss curve on epochs



Final Model pickled

```
with open('/content/drive/MyDrive/Capstone/ANNBertOriginal_model.pkl', 'wb') as file:
    pickle.dump(best_model_ann_bert_original, file)
```

13. Industrial Safety Implications

1. **Proactive Safety Culture**: High number of low-severity incident reports (Level IV) may indicate a proactive culture where even minor hazards are logged.
2. **Focus Areas for Improvement**: The most frequent incident types (on the left) should be targeted for safety interventions, training, or process improvement.
3. **Rare but Severe Events**: Though infrequent, any occurrence of Level I or II accidents should be treated as **critical failures** in controls and demand thorough investigation.
4. **Need for Balanced Reporting**: If some severe incidents are underreported, it could skew safety analysis — ensure balanced reporting across all severity levels.
5. Country_01 and Mining sector dominate incident count → ideal areas for deploying chatbot and preventive training.
6. Males and third-party workers dominate incidents. Future tools must monitor for and mitigate demographic bias.
7. Refine the "Critical Risk" and "Accident Level" taxonomy to support more granular predictions and alerts
8. When embedded in safety workflows, the chatbot can help triage incidents based on textual descriptions, improving real-time safety decisions.
9. Real-time integration in the factory floor or incident logging systems could enrich future models with more accurate and balanced data.

14. Limitations

10. Only 425 entries, insufficient for deep learning or generalizable ML models, particularly for multi-class classification.
11. The "Accident Level" column is dominated by level "I", with very few samples for levels IV and V.

12. "Critical Risk" has vague categories like "Others", reducing interpretability and actionable insights.
13. Very few entries for female employees or third-party remote workers, introducing bias risk.
14. The data spans mostly from 2016–2017, which restricts seasonal and trend analysis over time.
15. Mapping Accident Levels I–V into just 3 classes (Low, Medium, High) may discard critical granularity.

15. Closing Summary

- The size of data provided for processing and model building is significantly less and not sufficient to build a robust model with significantly high performance. Data is imbalanced as well. More primary data needs to be collected for improving prediction of 'Accident level'.
- We have decided to use 'Accident Level' as target as we are predicting the severity level of the accident that could potentially happen so that precautions and safety preventive action can be taken to reduce the accident impact/severity.
- Currently, the accuracy /precision using ANN is ~77% for 'Accident Level' as a target.
- Evaluation identified the ANN with Original Data (Bert tokenized) as the most robust performer among the four tested architectures. This model achieved the highest average F1 (0.6995), macro F1 (0.304), recall (0.773), and precision (0.706).