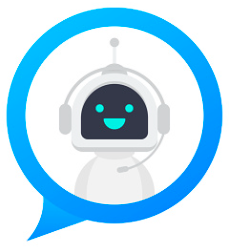


**Interim Project Report**

**Group 12 – NLP 2**



**NLP Based Chatbot**

**\_\_**

**PGP AIML Online Feb21 B**

This interim report summarizes the 1st 3 deliverables mentioned in the project document.

1. Summary of problem statement, data and findings. Every good abstract describes briefly what was intended at the outset, and summarizes findings and implications.

**Problem description:**

Domain: Industrial safety. NLP based Chatbot.

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

**Abstract:**

The need for manual labor in industries has decreased dramatically as technology has advanced. However, there is still a long way to go before industries are completely automated. Accidents and injuries are common when human activity is involved, despite all of the safety procedures and precautions put in place. Such injuries have the potential to be lethal. Depending on the industry, industrial accidents can take a variety of forms. A single spark, for example, in a firecracker factory might cause the entire business to burn down, resulting in the loss of lives and property. Workplace injuries are a major source of concern for both employees and managers. Industrial events must be classified into numerous categories in order to identify whether the occurrence was caused by an accident, negligence, or incompetence. Recurrences are avoided, the frequency and severity of occurrences are reduced, and the repercussions are minimized. To do so, we use exploratory data analysis on a dataset from one of Brazil's largest businesses to determine the leading causes of industrial accidents, the nature of the incidents, the types of employees wounded, and so on. To highlight this risk in automated way we plan to create a Chatbot application that uses natural language processing to classify the accident into different significant hazards based on the accident description.

**Significance of solution:**

The solution solved this problem by using a ML/DL based Chatbot utility which can help the professionals to highlight the safety risk and determine the accident level & potential level involved in any accident as per the incident description.

**Data description:**

This database records accidents from 12 different plants in 03 different countries.

Below are column description:

* + **Data:** Timestamp or time/date information
  + **Countries:** In which country the accident occurred.
  + **Local:** The city where the manufacturing plant is located.
  + **Industry sector:** In which sector the plant belongs to.
  + **Accident level:** From I to VI, it registers how severe was the accident (I means not severe but VI means very severe).
  + **Potential Accident Level:** Depending on the Accident Level, the database also registers how severe the accident could have been (due to other factors involved in the accident).
  + **Gender:** Whether the person is Male or Female.
  + **Employee or Third Party:** Whether an injured person is an employee or a third party.
  + **Critical Risk:** Short description of the risk involved in an accident.
  + **Description:** Detailed description of how the accident occurred.

**Observation:-**

There are 425 rows & 11 columns in the Data frame

# Column Non-Null Count Dtype

--- ------ -------------- -----

0 Unnamed: 0 425 non-null int64

1 Data 425 non-null object

2 Countries 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 Genre 425 non-null object

8 Employee or Third Party 425 non-null object

9 Critical Risk 425 non-null object

10 Description 425 non-null object

Index(['Unnamed: 0', 'Data', 'Countries', 'Local', 'Industry Sector',

'Accident Level', 'Potential Accident Level', 'Genre',

'Employee or Third Party', 'Critical Risk', 'Description'],

dtype='object')

From the above output, we see that except first column all other columns data type is object.

Categorical columns - 'Countries', 'Local', 'Industry Sector', 'Accident Level', 'Potential Accident Level', 'Genre', 'Employee or Third Party', 'Critical Risk', 'Description'

*# Check unique values of all columns except 'Description' column*

------------------------------------------------------------

Unique values of "Date" column

------------------------------------------------------------

['2016-01-01 00:00:00' '2016-01-02 00:00:00' '2016-01-06 00:00:00'

'2016-01-08 00:00:00' '2016-01-10 00:00:00' '2016-01-12 00:00:00'

'2016-01-16 00:00:00' '2016-01-17 00:00:00' '2016-01-19 00:00:00'

'2016-01-26 00:00:00' '2016-01-28 00:00:00' '2016-01-30 00:00:00'

'2016-02-01 00:00:00' '2016-02-02 00:00:00' '2016-02-04 00:00:00'

'2016-02-06 00:00:00' '2016-02-07 00:00:00' '2016-02-08 00:00:00'

'2016-02-21 00:00:00' '2016-02-25 00:00:00' '2016-02-09 00:00:00'

'2016-02-10 00:00:00' '2016-02-15 00:00:00' '2016-02-14 00:00:00'

'2016-02-13 00:00:00' '2016-02-16 00:00:00' '2016-02-17 00:00:00'

'2016-02-19 00:00:00' '2016-02-20 00:00:00' '2016-02-18 00:00:00'

'2016-02-22 00:00:00' '2016-02-24 00:00:00' '2016-02-29 00:00:00'

'2016-02-26 00:00:00' '2016-02-27 00:00:00' '2016-03-02 00:00:00'

'2016-03-03 00:00:00' '2016-03-04 00:00:00' '2016-03-05 00:00:00'

'2016-03-06 00:00:00' '2016-03-09 00:00:00' '2016-03-11 00:00:00'

'2016-03-13 00:00:00' '2016-03-12 00:00:00' '2016-03-14 00:00:00'

'2016-03-16 00:00:00' '2016-03-10 00:00:00' '2016-03-17 00:00:00'

'2016-03-18 00:00:00' '2016-03-19 00:00:00' '2016-03-22 00:00:00'

'2016-03-25 00:00:00' '2016-03-30 00:00:00' '2016-03-31 00:00:00'

'2016-04-01 00:00:00' '2016-04-03 00:00:00' '2016-04-02 00:00:00'

'2016-03-24 00:00:00' '2016-04-04 00:00:00' '2016-04-05 00:00:00'

'2016-04-07 00:00:00' '2016-04-08 00:00:00' '2016-04-11 00:00:00'

'2016-04-14 00:00:00' '2016-04-16 00:00:00' '2016-04-15 00:00:00'

'2016-04-17 00:00:00' '2016-04-18 00:00:00' '2016-04-21 00:00:00'

'2016-04-22 00:00:00' '2016-04-23 00:00:00' '2016-04-26 00:00:00'

'2016-04-28 00:00:00' '2016-04-29 00:00:00' '2016-04-30 00:00:00'

'2016-05-01 00:00:00' '2016-05-02 00:00:00' '2016-05-04 00:00:00'

'2016-05-03 00:00:00' '2016-05-05 00:00:00' '2016-05-11 00:00:00'

'2016-05-12 00:00:00' '2016-05-14 00:00:00' '2016-05-17 00:00:00'

'2016-05-19 00:00:00' '2016-05-18 00:00:00' '2016-05-22 00:00:00'

'2016-05-20 00:00:00' '2016-05-24 00:00:00' '2016-05-25 00:00:00'

'2016-05-27 00:00:00' '2016-05-26 00:00:00' '2016-06-01 00:00:00'

'2016-06-02 00:00:00' '2016-06-03 00:00:00' '2016-06-04 00:00:00'

'2016-06-05 00:00:00' '2016-06-08 00:00:00' '2016-06-07 00:00:00'

'2016-06-10 00:00:00' '2016-06-13 00:00:00' '2016-06-16 00:00:00'

'2016-06-18 00:00:00' '2016-06-17 00:00:00' '2016-06-19 00:00:00'

'2016-06-21 00:00:00' '2016-06-22 00:00:00' '2016-06-23 00:00:00'

'2016-06-24 00:00:00' '2016-06-29 00:00:00' '2016-07-02 00:00:00'

'2016-07-04 00:00:00' '2016-07-08 00:00:00' '2016-07-07 00:00:00'

'2016-07-09 00:00:00' '2016-07-10 00:00:00' '2016-07-11 00:00:00'

'2016-07-14 00:00:00' '2016-07-15 00:00:00' '2016-07-16 00:00:00'

'2016-07-18 00:00:00' '2016-07-20 00:00:00' '2016-07-21 00:00:00'

'2016-07-23 00:00:00' '2016-07-27 00:00:00' '2016-07-29 00:00:00'

'2016-07-30 00:00:00' '2016-08-02 00:00:00' '2016-08-01 00:00:00'

'2016-08-04 00:00:00' '2016-08-11 00:00:00' '2016-08-12 00:00:00'

'2016-08-14 00:00:00' '2016-08-15 00:00:00' '2016-08-18 00:00:00'

'2016-08-19 00:00:00' '2016-08-22 00:00:00' '2016-08-24 00:00:00'

'2016-08-25 00:00:00' '2016-08-29 00:00:00' '2016-08-27 00:00:00'

'2016-08-30 00:00:00' '2016-09-01 00:00:00' '2016-09-02 00:00:00'

'2016-09-04 00:00:00' '2016-09-03 00:00:00' '2016-09-06 00:00:00'

'2016-09-05 00:00:00' '2016-09-13 00:00:00' '2016-09-12 00:00:00'

'2016-09-15 00:00:00' '2016-09-17 00:00:00' '2016-09-16 00:00:00'

'2016-09-20 00:00:00' '2016-09-21 00:00:00' '2016-09-22 00:00:00'

'2016-09-27 00:00:00' '2016-09-29 00:00:00' '2016-09-30 00:00:00'

'2016-10-01 00:00:00' '2016-10-03 00:00:00' '2016-10-04 00:00:00'

'2016-10-08 00:00:00' '2016-10-10 00:00:00' '2016-10-11 00:00:00'

'2016-10-13 00:00:00' '2016-10-18 00:00:00' '2016-10-20 00:00:00'

'2016-10-23 00:00:00' '2016-10-24 00:00:00' '2016-10-26 00:00:00'

'2016-10-27 00:00:00' '2016-10-29 00:00:00' '2016-11-04 00:00:00'

'2016-11-08 00:00:00' '2016-11-11 00:00:00' '2016-11-13 00:00:00'

'2016-11-19 00:00:00' '2016-11-21 00:00:00' '2016-11-23 00:00:00'

'2016-11-25 00:00:00' '2016-11-28 00:00:00' '2016-11-29 00:00:00'

'2016-11-30 00:00:00' '2016-12-01 00:00:00' '2016-12-08 00:00:00'

'2016-12-09 00:00:00' '2016-12-10 00:00:00' '2016-12-12 00:00:00'

'2016-12-13 00:00:00' '2016-12-15 00:00:00' '2016-12-16 00:00:00'

'2016-12-19 00:00:00' '2016-12-23 00:00:00' '2016-12-22 00:00:00'

'2016-12-26 00:00:00' '2016-12-28 00:00:00' '2016-12-30 00:00:00'

'2016-12-31 00:00:00' '2017-01-02 00:00:00' '2017-01-05 00:00:00'

'2017-01-06 00:00:00' '2017-01-07 00:00:00' '2017-01-08 00:00:00'

'2017-01-09 00:00:00' '2017-01-10 00:00:00' '2017-01-12 00:00:00'

'2017-01-14 00:00:00' '2017-01-17 00:00:00' '2017-01-20 00:00:00'

'2017-01-21 00:00:00' '2017-01-23 00:00:00' '2017-01-24 00:00:00'

'2017-01-25 00:00:00' '2017-01-27 00:00:00' '2017-01-29 00:00:00'

'2017-01-28 00:00:00' '2017-01-31 00:00:00' '2017-02-01 00:00:00'

'2017-02-04 00:00:00' '2017-02-05 00:00:00' '2017-02-07 00:00:00'

'2017-02-08 00:00:00' '2017-02-09 00:00:00' '2017-02-13 00:00:00'

'2017-02-14 00:00:00' '2017-02-15 00:00:00' '2017-02-16 00:00:00'

'2017-02-17 00:00:00' '2017-02-23 00:00:00' '2017-02-25 00:00:00'

'2017-02-26 00:00:00' '2017-02-27 00:00:00' '2017-03-01 00:00:00'

'2017-03-02 00:00:00' '2017-03-04 00:00:00' '2017-03-06 00:00:00'

'2017-03-08 00:00:00' '2017-03-09 00:00:00' '2017-03-10 00:00:00'

'2017-03-15 00:00:00' '2017-03-18 00:00:00' '2017-03-22 00:00:00'

'2017-03-25 00:00:00' '2017-03-31 00:00:00' '2017-04-04 00:00:00'

'2017-04-05 00:00:00' '2017-04-07 00:00:00' '2017-04-06 00:00:00'

'2017-04-10 00:00:00' '2017-04-08 00:00:00' '2017-04-11 00:00:00'

'2017-04-13 00:00:00' '2017-04-12 00:00:00' '2017-04-23 00:00:00'

'2017-04-19 00:00:00' '2017-04-25 00:00:00' '2017-04-24 00:00:00'

'2017-04-28 00:00:00' '2017-04-29 00:00:00' '2017-04-30 00:00:00'

'2017-05-05 00:00:00' '2017-05-06 00:00:00' '2017-05-10 00:00:00'

'2017-05-16 00:00:00' '2017-05-17 00:00:00' '2017-05-18 00:00:00'

'2017-05-19 00:00:00' '2017-05-23 00:00:00' '2017-05-30 00:00:00'

'2017-06-04 00:00:00' '2017-06-09 00:00:00' '2017-06-11 00:00:00'

'2017-06-14 00:00:00' '2017-06-15 00:00:00' '2017-06-17 00:00:00'

'2017-06-18 00:00:00' '2017-06-24 00:00:00' '2017-06-20 00:00:00'

'2017-06-23 00:00:00' '2017-06-19 00:00:00' '2017-06-22 00:00:00'

'2017-06-29 00:00:00' '2017-07-04 00:00:00' '2017-07-05 00:00:00'

'2017-07-06 00:00:00' '2017-07-09 00:00:00']

------------------------------------------------------------

Unique values of "Country" column

------------------------------------------------------------

['Country\_01' 'Country\_02' 'Country\_03']

------------------------------------------------------------

Unique values of "Local" column

------------------------------------------------------------

['Local\_01' 'Local\_02' 'Local\_03' 'Local\_04' 'Local\_05' 'Local\_06'

'Local\_07' 'Local\_08' 'Local\_10' 'Local\_09' 'Local\_11' 'Local\_12']

------------------------------------------------------------

Unique values of "Industry Sector" column

------------------------------------------------------------

['Mining' 'Metals' 'Others']

------------------------------------------------------------

Unique values of "Accident Level" column

------------------------------------------------------------

['I' 'IV' 'III' 'II' 'V']

------------------------------------------------------------

Unique values of "Potential Accident Level" column

------------------------------------------------------------

['IV' 'III' 'I' 'II' 'V' 'VI']

------------------------------------------------------------

Unique values of "Gender" column

------------------------------------------------------------

['Male' 'Female']

------------------------------------------------------------

Unique values of "Employee type" column

------------------------------------------------------------

['Third Party' 'Employee' 'Third Party (Remote)']

------------------------------------------------------------

Unique values of "Critical Risk" column

------------------------------------------------------------

['Pressed' 'Pressurized Systems' 'Manual Tools' 'Others'

'Fall prevention (same level)' 'Chemical substances' 'Liquid Metal'

'Electrical installation' 'Confined space'

'Pressurized Systems / Chemical Substances'

'Blocking and isolation of energies' 'Suspended Loads' 'Poll' 'Cut'

'Fall' 'Bees' 'Fall prevention' '\nNot applicable' 'Traffic' 'Projection'

'Venomous Animals' 'Plates' 'Projection/Burning' 'remains of choco'

'Vehicles and Mobile Equipment' 'Projection/Choco' 'Machine Protection'

'Power lock' 'Burn' 'Projection/Manual Tools'

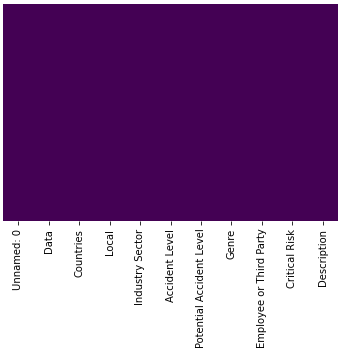
'Individual protection equipment' 'Electrical Shock'

'Projection of fragments']

* We observed that there are records of accidents from 1st Jan 2016 to 9th July 2017 in every month. So there are no outliers in the 'Date' column.
* There are only three country types so there are no outliers in 'Country' column.
* There are 12 Local cities where manufacturing plant is located and it's types are in sequence so there are no outliers in 'Local' column.
* There are only three Industry Sector types which are in sequence so there are no outliers in 'Industry Sector' column.
* There are only five Accident Level types which are in sequence so there are no outliers in 'Accident Level' column.
* There are only six Potential Accident Level types which are in sequence so there are no outliers in 'Potential Accident Level' column.
* There are only two Gender types in the provided data so there are no outliers in 'Gender' column.
* There are only three Employee types in the provided data so there are no outliers in 'Gender' column.

1. Summary of the Approach to EDA and Pre-processing. Include any insightful visualization you have teased out of the data. If you’ve identified particularly meaningful features, interactions or summary data, share them and explain what you noticed. Visual displays are powerful when used well, so think carefully about what information the display conveys.

**Null values check:**

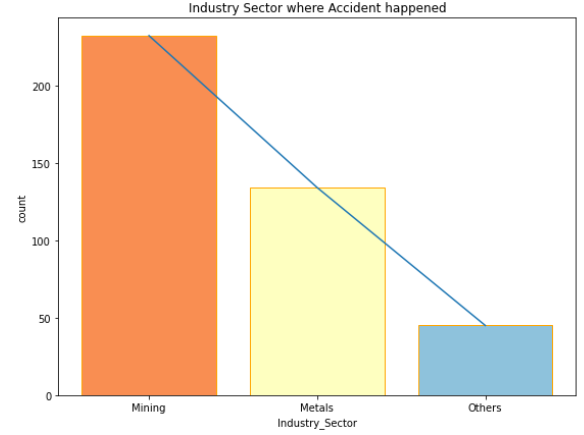


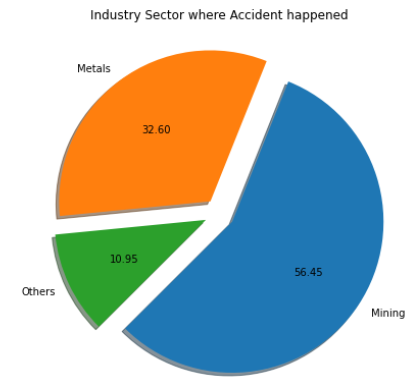
**Observation: -**

* From HeatMap we conclude that there are no null or Nan values provided in data.

**Univariate Analysis:**

Industry Sector column:

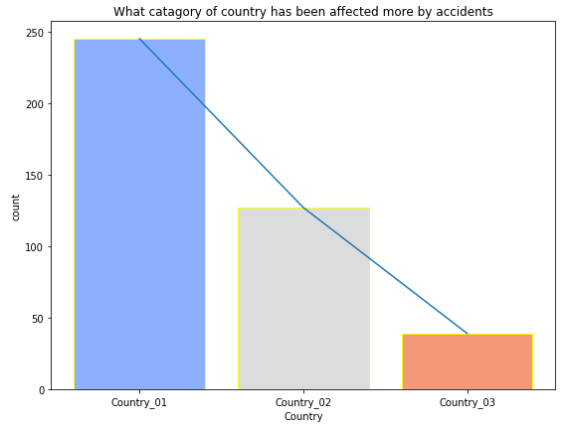


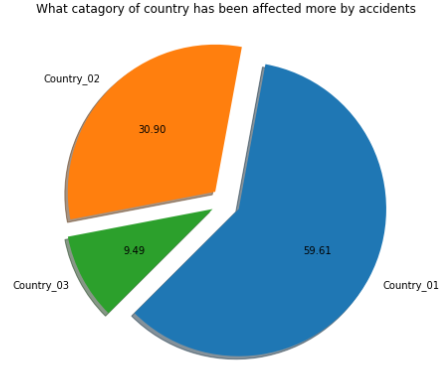


**Observation:-**

* It clearly indicates that mining industries witness more accidents.
* We can say that the number of accidents in Mining Industry is considerably more than that in the Metal Industry, therefore mining job is more risky than the latter.

Country column:

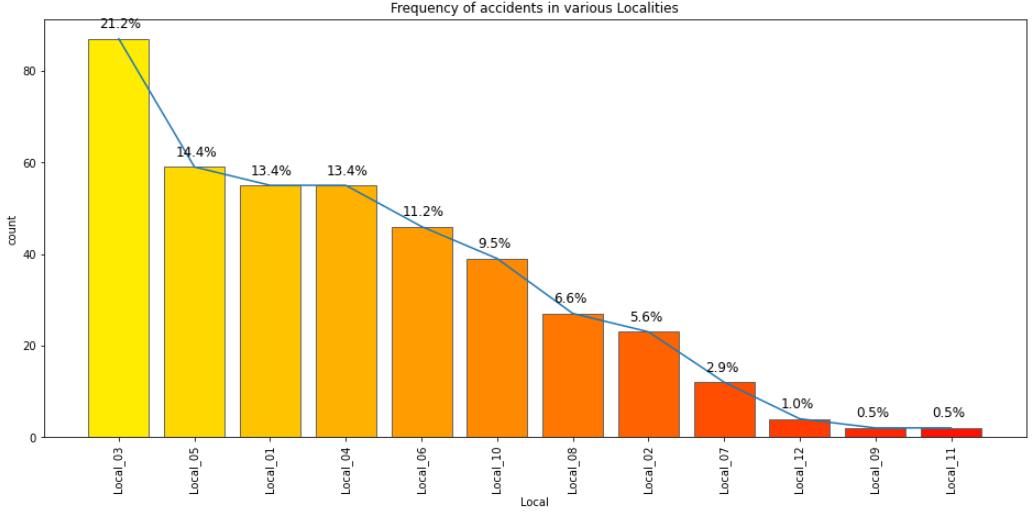




**Observation:-**

* Country\_01 is highly affected with 59.61%.
* Country\_02 is 2nd highest affected with 30.90%.
* And Country\_03 is least affected with 9.49%.

Local column:

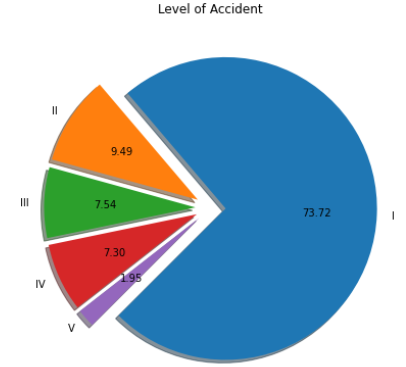


**Observation:-**

* Local\_03 has recorded maximum number of accidents which is approx. 21% of all the plants in the country.
* This is followed by Local-05, Local-01 and so on.

Accident Level column:

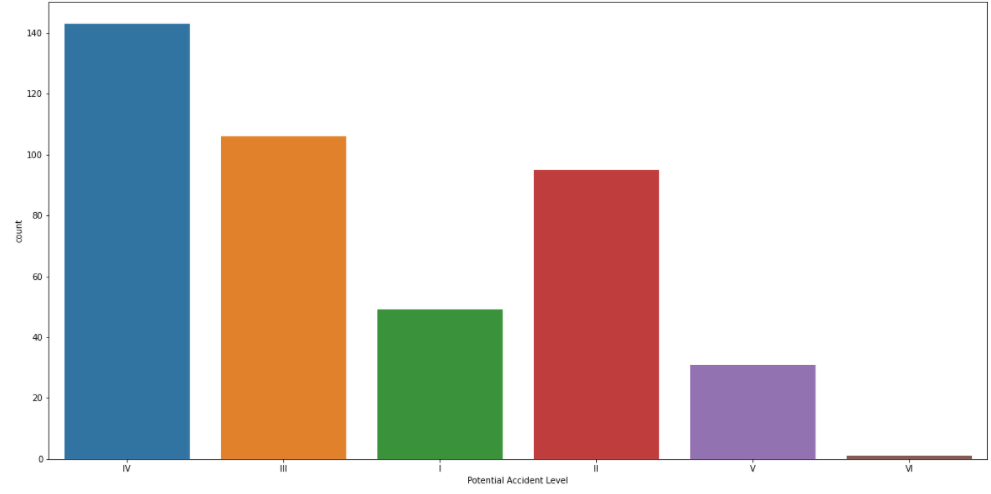




**Observation:-**

* Level I signifies not severe and Level V signifies very severe.
* Accidents with the level I are most common. These are due to small misses, like people forgot their PPE, or they dropped a tool, etc.

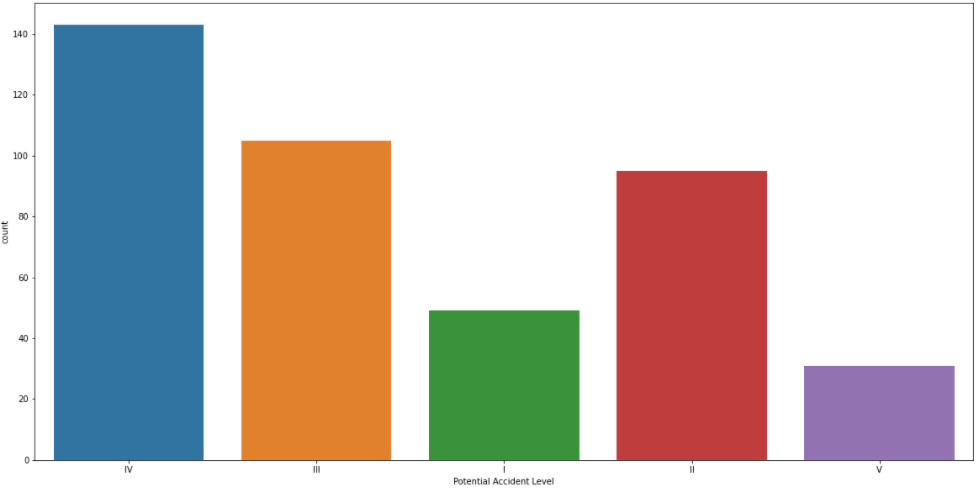
Potential Accident Level column:

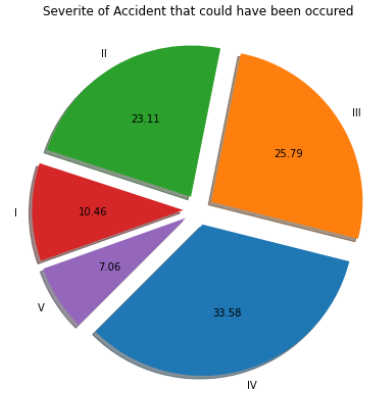


**Observation:-**

* The dataset provided here is imbalanced.
* Potential Accident Level IV has highest number of records i.e. 143.
* Potential Accident Level III has second highest number of records i.e. 106.
* Potential Accident Level II has third highest number of records i.e. 95.
* Potential Accident Level VI has lowest number of records i.e. 1.

Potential Accident Level column after deleting level VI:

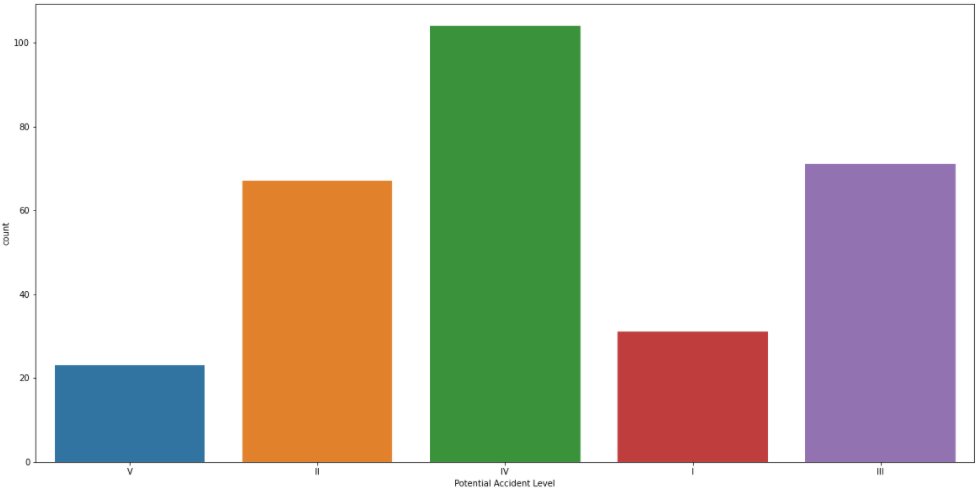




**Observation:-**

* Level VI has only 1 record in dataset hence deleted.
* Number of records for rest of the levels remains same.

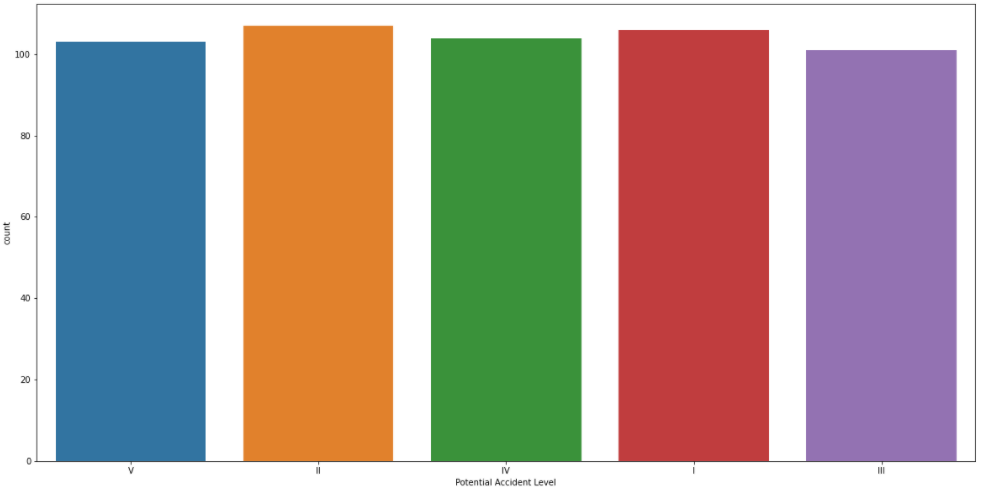
Potential Accident Level column after train-test split:



**Observation:-**

* Above bar plot is of train data after 70%-30% train-test split

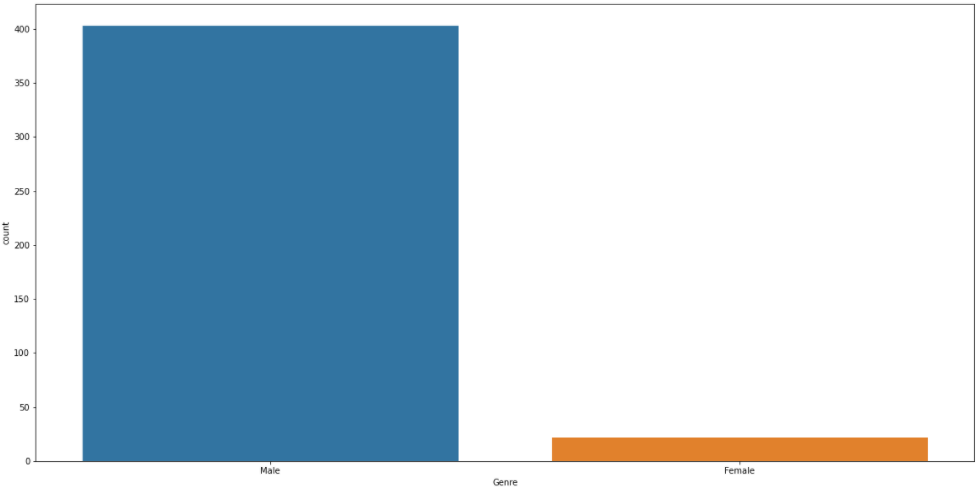
Potential Accident Level column after data augmentation:

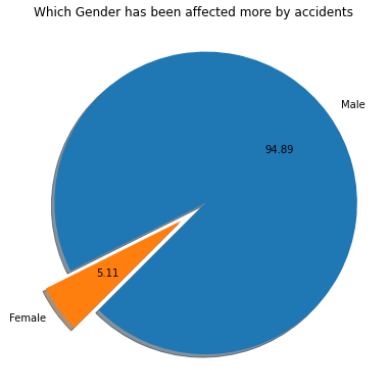


**Observation:-**

* Before data augmentation the data was highly imbalance.
* Now the data is somewhat balanced.

Genre column:

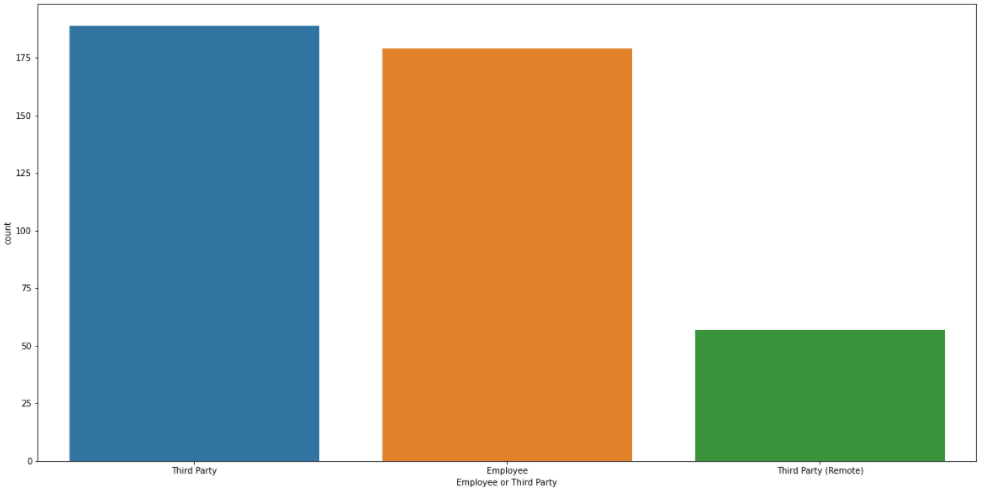


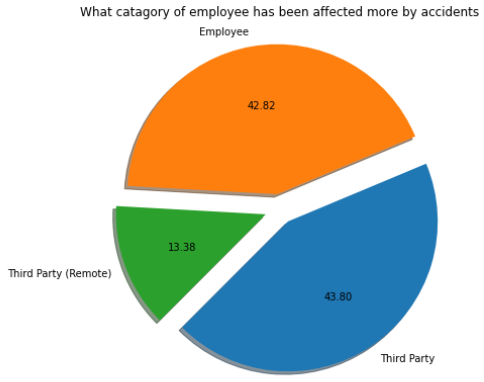


**Observation:-**

* Under Genre there are 2 categories - Male & Female.
* In dataset, Male population is higher than Females.
* Male population is 403 which is 94.82% in that category.
* Female population is 22 which is 5.18% in that category.

Employee or Third Party column:

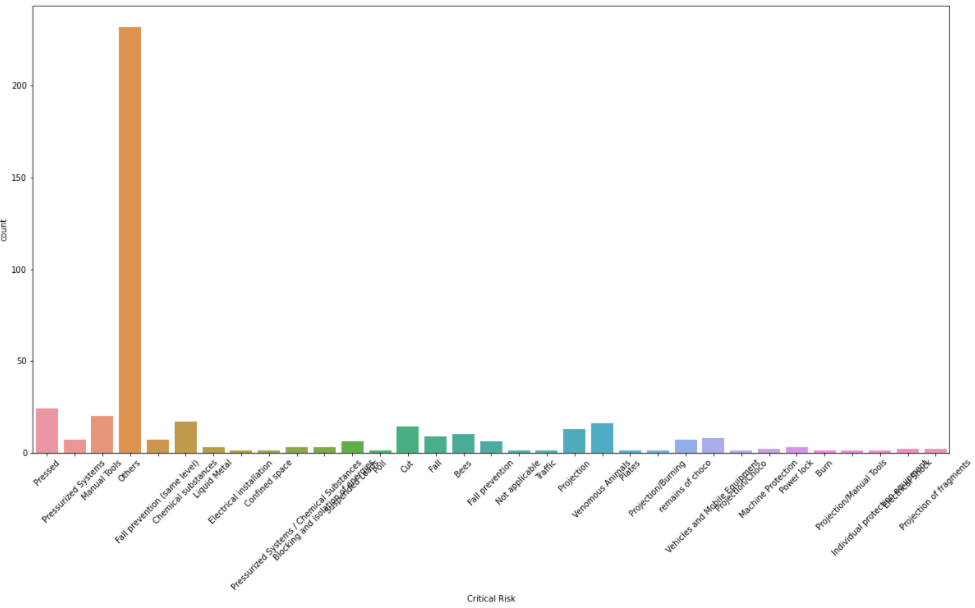




**Observation:-**

* Under Employee or Third Party column there are 3 categories - Third Party, Employee & Third Party (Remote).
* In dataset, Third Party category is highest - 189 which is 44.47% in that category.
* Employee category is second highest - 179 which is 42.12% in that category.
* Third Party (Remote) category is third highest - 57 which is 13.41% in that category.

Critical Risk column:



**Observation:-**

* Under Critical Risk column there are 33 categories -
  + Pressed.
  + Manual Tools
  + Chemical substances
  + Venomous Animals
  + Cut
  + Projection
  + Bees
  + Fall
  + Vehicles and Mobile Equipment
  + Fall prevention (same level)
  + remains of choco
  + Pressurized Systems
  + Fall prevention
  + Suspended Loads
  + Blocking and isolation of energies
  + Pressurized Systems / Chemical Substances
  + Power lock
  + Liquid Metal, Projection of fragments
  + Machine Protection
  + Electrical Shock
  + Individual protection equipment
  + Projection/Manual Tools
  + Burn
  + Poll
  + Projection/Choco
  + Projection/Burning
  + Plates
  + Confined space
  + Traffic
  + Not applicable
  + Electrical installation
  + Others
* In dataset, “Others” category is highest - 232 which is 54.59% in that category.
* Pressed is second highest - 24 which is 5.65% in that category.
* Manual Tools category is third highest - 20 which is 4.71% in that category.

Statistical evaluation on Description column:

Count 425.000000

Mean 368.280000

STD 178.944426

Min 94.000000

25% 227.000000

50% 335.000000

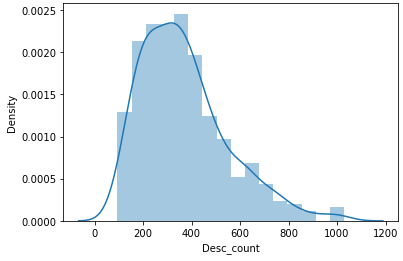
75% 457.000000

Max 1029.000000

**Observation:-**

* In dataset we have Description column, a new column called Desc\_count has been added based on Description column.
* Desc\_count contains total character count under Description for each record.
* We have used describe function on this Desc\_count column which shows various statistics.
  + There total 425 records.
  + The mean of Desc\_count column is 368.28
  + Standard deviation is 178.94 for Desc\_count column.
  + The minimum description length is 94 characters.
  + The maximum description length is 1029 characters.

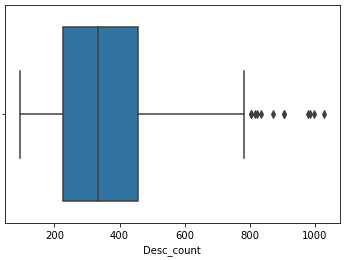
Desc\_count column:



**Observation:-**

* Above is distribution plot or Gaussian curve for Desc\_count.
* The curve is not exactly bell shaped & is skewed.
* The curve is more skewed on right side whereas on left side its steep.

Desc\_count column:

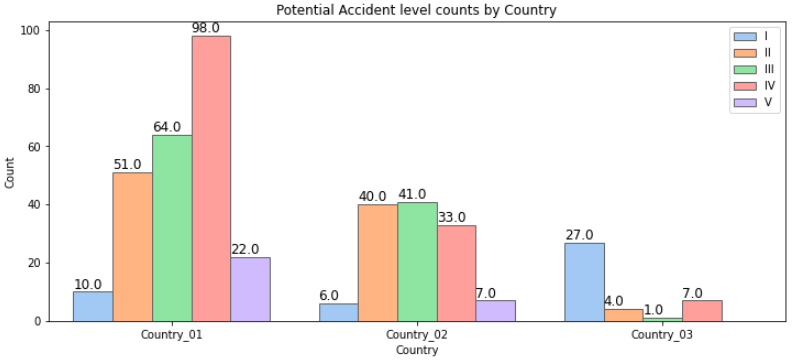


**Observation:-**

* Desc\_count column has outliers near max whisker.
* There are more number of data between 75% & Max whisker as compared to 25% & min whisker.
* Also there are more number of data between 50% & 75% points.

**Bivariate Analysis:**

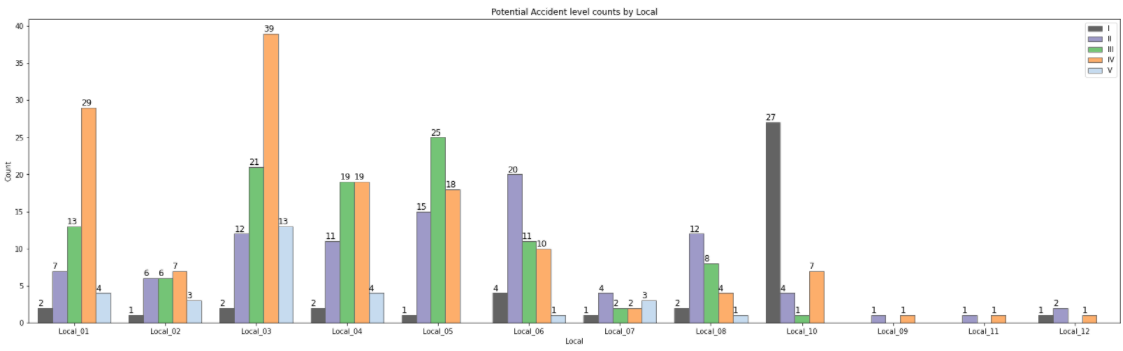
Potential Accident Level column versus Country column:



**Observation:-**

* Country\_01 has more number of severe accidents especially Level IV
* Country\_02 has moderate accidents across all the levels
* Country\_03 'level I' accidents counts is more compared to country\_01 and country\_02 but less severe accidents.

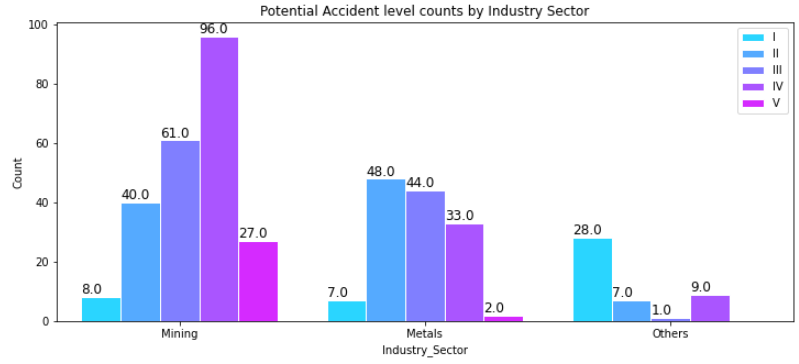
Potential Accident Level column versus Local column:



**Observation:-**

* Local\_03 (which also belongs to Country\_01) is where most of the accidents happen.

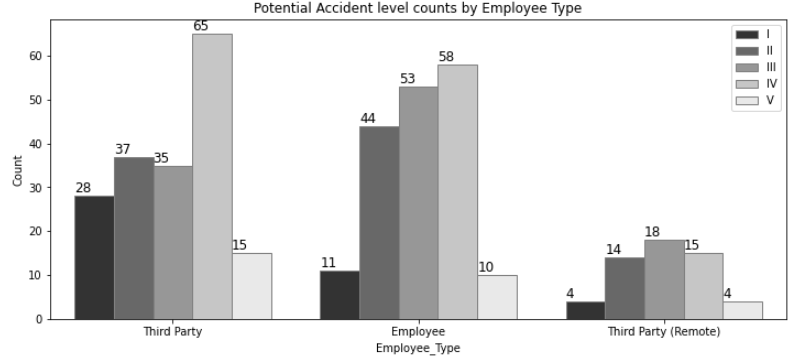
Potential Accident Level column versus Industry Sector column:



**Observation:-**

* Out of all industries, Mining Industry has seen some accidents whose level is the most severe and the corresponding potential Accident level is also highest.
* This is followed by Metal industry and other.
* Severity levels of the incidents are more in Mining sector (rate of level 4 is slightly higher than the level 2 & 3)

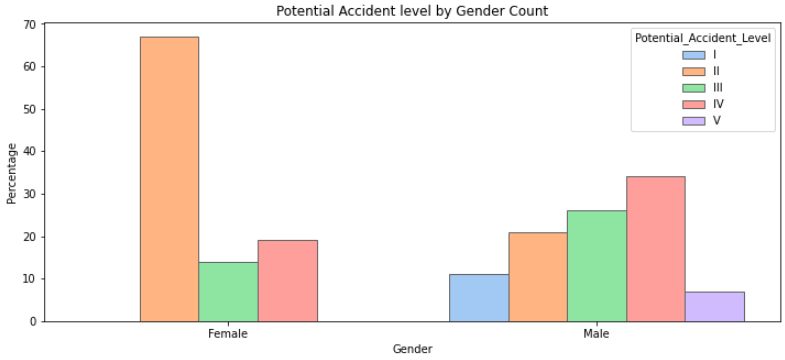
Potential Accident Level column versus Employee Type column:



**Observation:-**

* Third Party Employees are more involved in Accidents
* We can observe that apart from Accident Level I the people are also facing severe accidents (Accident Level IV) in the industry.

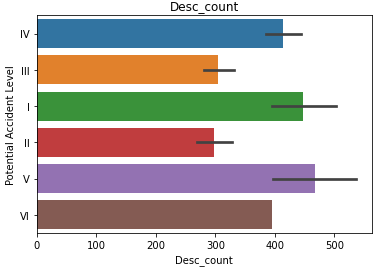
Potential Accident Level column versus Gender column:



**Observation:-**

* Males are more involved in Severe Accidents whereas Females are suffering with less Severe ones (Level II specifically)

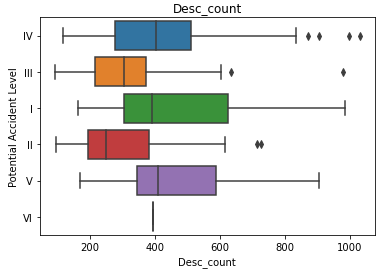
Potential Accident Level column versus Desc\_count column:



**Observation:-**

* Above is a bar plot between Potential Accident Level column & Desc\_count column.
* Potential Accident Level V has highest number of Desc\_count value.
* Potential Accident Level I has 2nd highest number of Desc\_count value.
* Potential Accident Level IV has 3rd highest number of Desc\_count value.

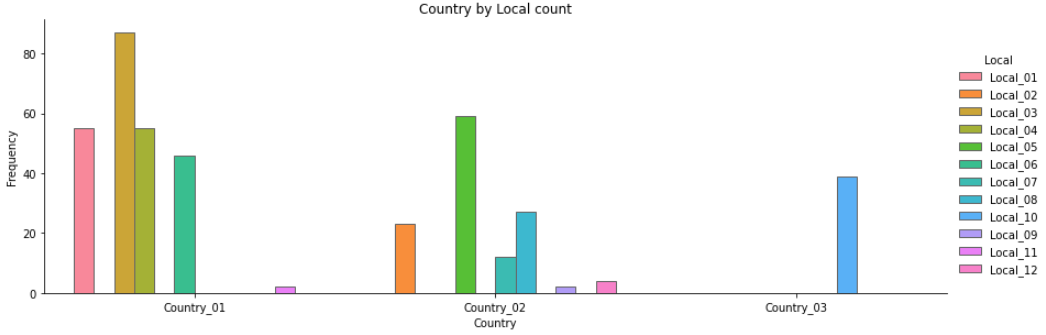
Potential Accident Level column versus Desc\_count column:



**Observation:-**

* Above is a box plot between Potential Accident Level column & Desc\_count column.
* Potential Accident Level IV has highest number - 4 of outliers near max whisker.
* Potential Accident Level III & II has 2 outliers near max whisker.
* None of the box plot has outliers near min whisker.
* All Potential Accident Levels has more number of data between 75% & Max whisker as compared to 25% & Min whisker.
* For Potential Accident Level IV & III there are more number of data between 50% & 75% points.
* For Potential Accident Level I, II & V there are more number of data between 25% & 50% points.
* For Potential Accident Level VI there is only 1 record hence box plot is not visible.

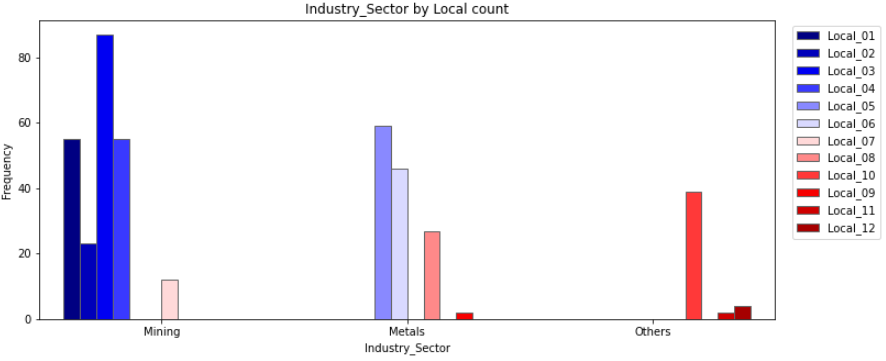
Accident count by Country & Local:



**Observation:-**

* Maximum number of accidents are contributed by Country\_01 and the plants/locals belongs to Country\_01 as we saw in the previous plots where Local\_01 and Local\_05 (belongs to Country\_01) have the highest number of incidents.

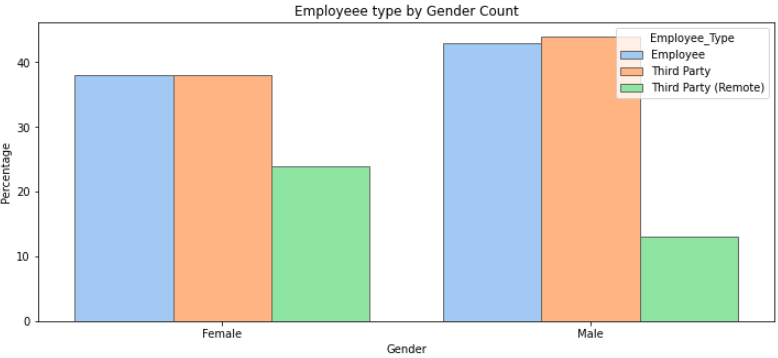
Accident count by Industry Sector & Local:



**Observation:-**

* It can be observed that Industry Sector depends on the Local area too.
* Local Area 1, 2, 3, 4 and 7 belong to Mining Sector.
* Local Area 5, 6, 8 and 9 belong to Metal Sector.
* Local Area 10, 11 and 12 belong to Other Sectors.

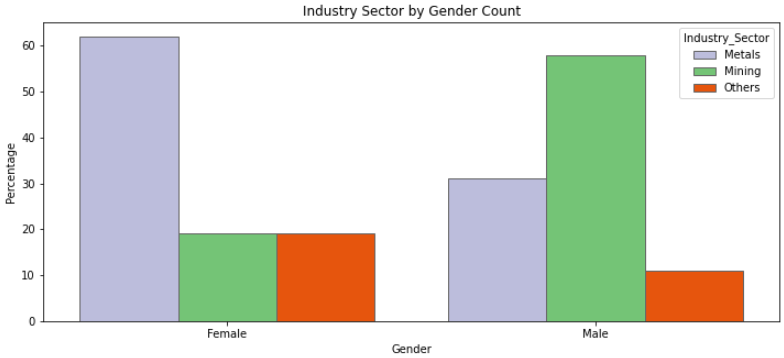
Employee type by Gender count:



**Observation:-**

* Division of employee type in Male and Female is almost the same.
* As seen in the above plots third party employees are slightly higher than the employee count in both of the sectors (Male & Female).
* Proportion of Female third party remote employee is moderately higher than that of the Males.

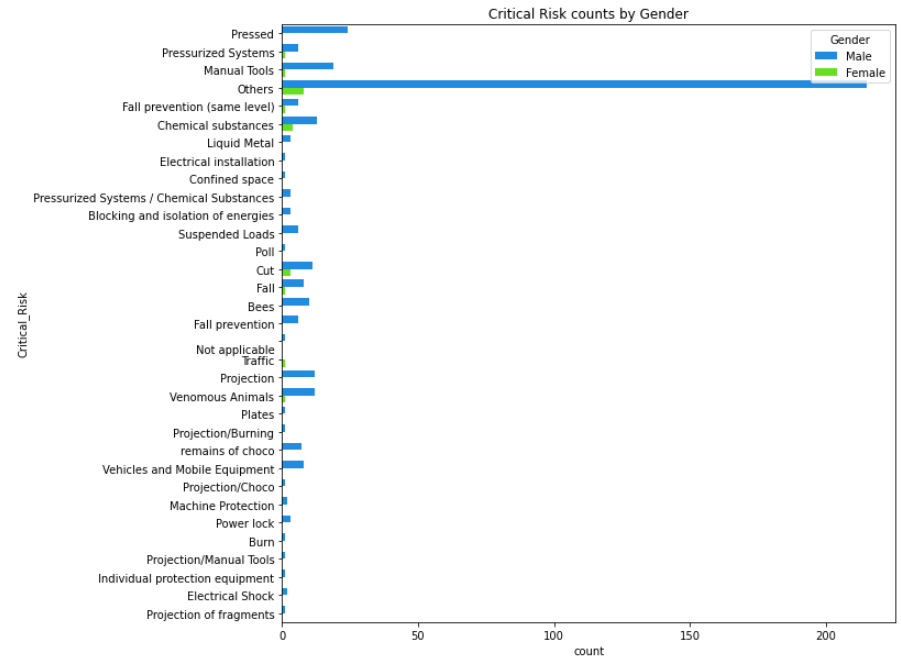
Industry Sector by Gender count:



**Observation:-**

* There is a major difference in mining and metals industries within Males and Females.
* We can observe the distinct safety levels by the industry sectors towards Male and Female.

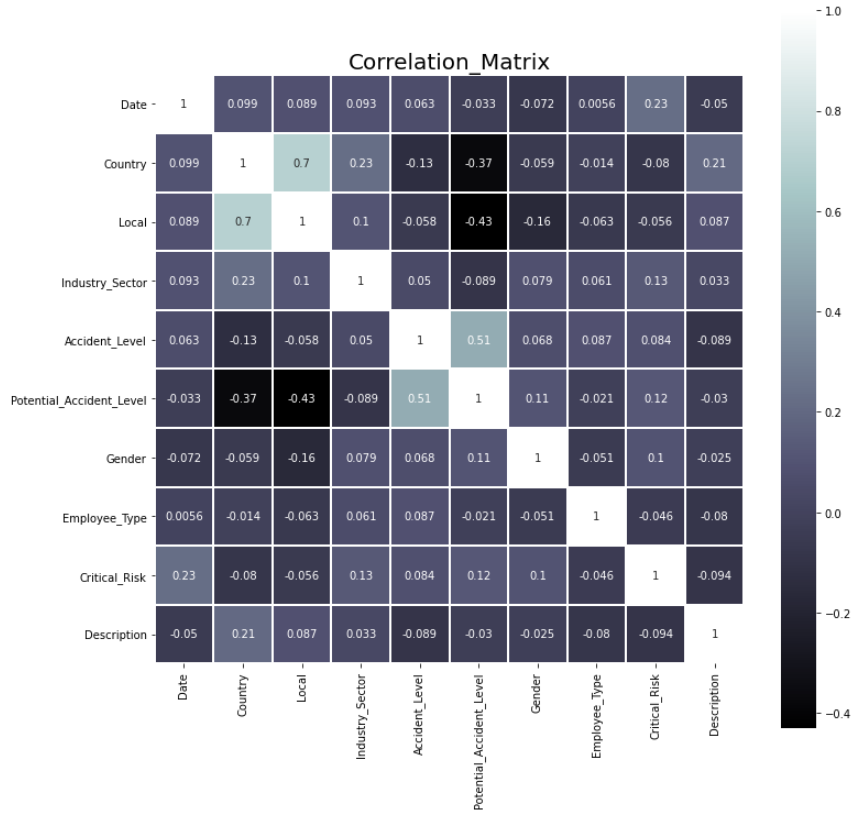
Critical Risk counts by Gender:



**Observation:-**

* There are some amount of females who involved in Critical Risks such as Chemical Substances, Cut, Fall other than Others category.
* As dataset has more number of male data points, the plot bars tends to increase on the same class.

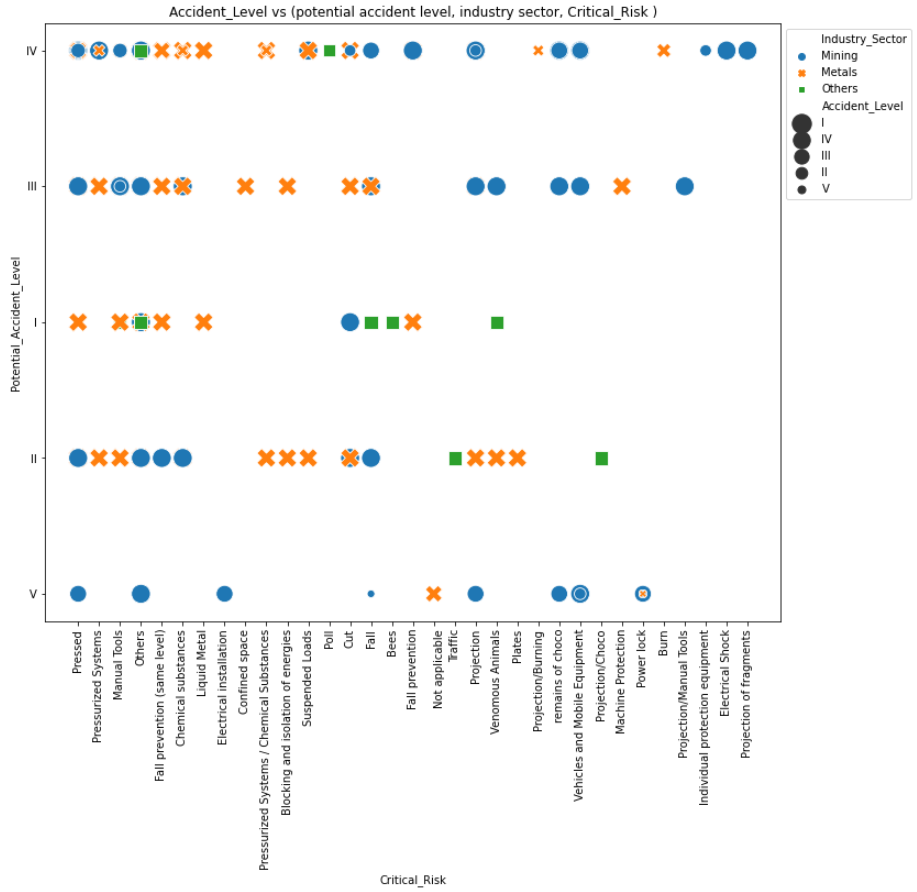
Correlation Plot:



**Observation:-**

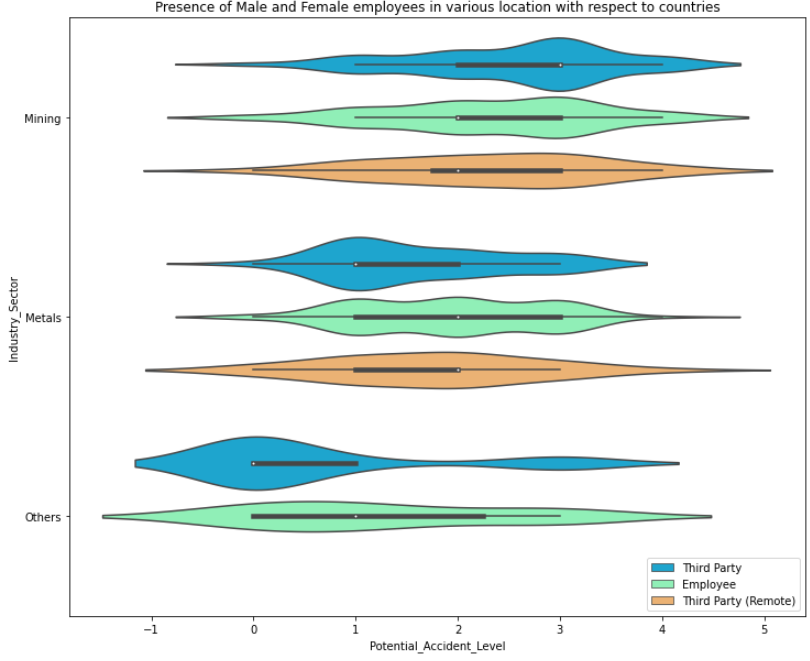
* Accident\_Levels are moderately correlated with the Potential\_Accident\_Levels which indicates minor/major incidents have the potential to cause similar/severe accident levels in an industry.

**Multivariate Analysis:**



**Observation:-**

* Out of all industries, Mining Industry has had most severe accidents and their corresponding potential Accident level is also high.
* Most of the accidents have been reported from mining industry and almost all of them are of moderate level, followed by Metal industry and Others.
* The Potential accident level and Accident level are equivalent to each other. Some of the critical risks are less severe and have reported maximum number of accidents.

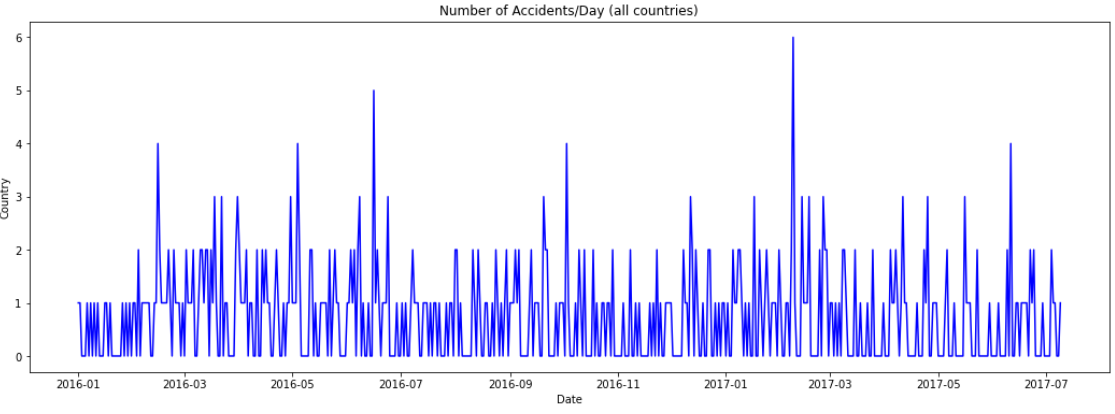


**Observation:-**

* For Metals industry, the distribution of potential accident level for third party employee is right skewed, that means most of the accidents have moderate risk.
* For Direct employee, there is an equal distribution between low and moderate risk. Same for Third party remote employees, but it is long tailed.
* For other industries, most of the third party employees are affected by very low risk which is a good sign.
* Similarly, for Direct employees the potential accident level is distributed well and the mode of the distribution is located at a low risk level (with a very extended inter quartile range).
* No data is available for Third party remote employees who belong to industries.

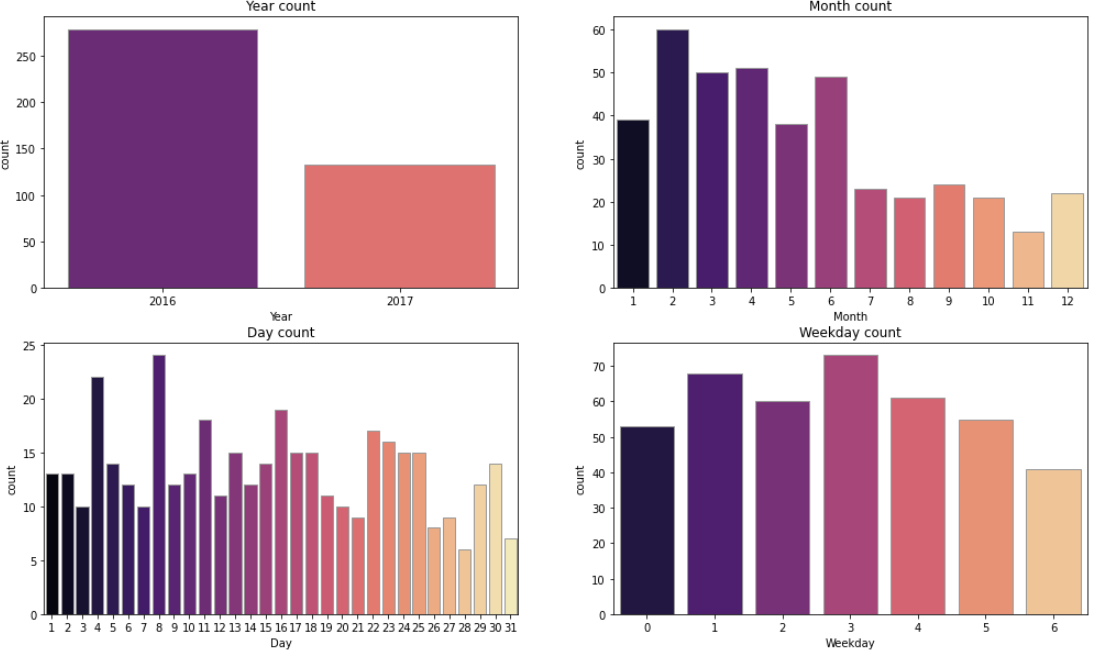
**Time Series Analysis**

Accident Level counts by Year



**Observation:-**

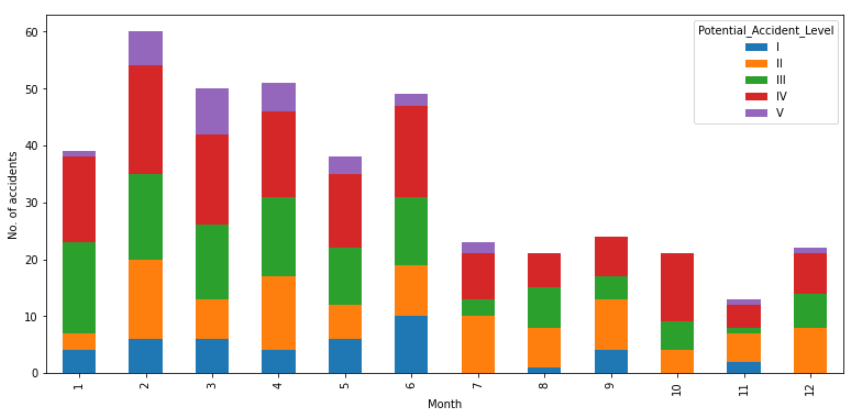
* It can be noticed that there are multiple peaks of accidents every year. There is a high peak in February of 2017



**Observation:-**

* It can be observed that more number of accidents occurred in 2016 compared to 2017, in year 2016 we have all 12 months of data whereas year 2017 has only 7 months of data.
* It seems that the number of accidents decreased in latter of the year / month.
* The number of accidents increased during the middle of the week and declined since the middle of the week.

Potential Accident Level counts by Month:



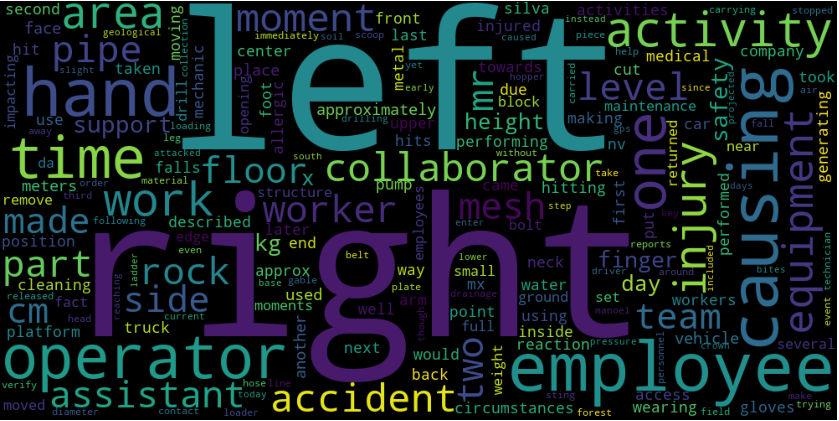
**Observation:-**

* Accident level have the tendency that non-severe levels (I, II) decreased throughout the year but severe (III, IV) levels did not change much.
* Initial months see more accidents which seem to decrease by year end
* High level accidents (V) occur in the initial months

**Overall observation:-**

* Columns – Unnamed, Date, Countries & Local don’t play any crucial role in prediction of Potential Accident Level.
* After performing dimensional reduction technique – Principal Component Analysis (PCA) it has been seen that columns – Industry Sector, Genre, Employee or Third Party & Critical Risk also don’t play any crucial role in prediction of Potential Accident Level.
* Mentioned columns would be dropped before performing further analysis.

**Word Cloud:**



**Observation:-**

* The word cloud displayed above is good, but some of the words are larger than the others.
* The size of the word in the word cloud is proportional to the frequency of the word inside the corpus.
* From the above image, we can see that the stop words are not displayed.
* Also, observe that words like employee, right, accident, left, causing, time, operator, etc. are the most prominent words in the word cloud.

### Text Pre-processing:

Text pre-processing is a method to clean the text data and make it ready for the model. Text data contains noise in various forms like emotions, punctuations, text in a different case. When we talk about Human Language, there are different ways to say the same thing. This is the main problem we have to deal with because machines will not understand words, they need numbers so we need to convert text to numbers in an efficient manner. Below are various text pre-processing techniques that are used:

Convert to Lower case: If the text is in the same case, it is easy for a machine to interpret the words because the lower case and upper case are treated differently by the machine. Example, words like Ball and ball are treated differently by machine. So, we need to make the text in the same case and the most preferred case is a lower case to avoid such problems.

Let’s take the first description record in the dataset as an example through the preprocessing steps:

Graphical user interface, text, application

Description automatically generated

Removal of Punctuation: One of the other text processing techniques is removing punctuations. There are total 32 main punctuations that need to be taken care of. We can directly use the string module with a regular expression to replace any punctuation in text with an empty string. 32 punctuations which string module provide us is listed below.

A picture containing application

Description automatically generated

Removing unwanted spaces: Most of the time text data contain extra spaces or while performing the above pre-processing techniques more than one space is left between the texts so we need to control this problem. Regular expression library performs well to solve this problem.

Graphical user interface, application, Word

Description automatically generated

Removal of Stopwords: Stopwords are the most commonly occurring words in a text which do not provide any valuable information. Stopwords like they, there, this, where, etc. are some of the stopwords. NLTK library is used to remove stopwords and include approximately 180 stopwords which it removes. If we want to add any new word to a set of words then it is easy using the add method. Below is code snippet for removal of stopwords.

Application, Word

Description automatically generated with medium confidence

Stemming: Stemming is a process to reduce the word to its root stem for example run, running, runs, runed derived from the same word as run. Basically stemming do is remove the prefix or suffix from word like ing, s, es, etc. NLTK library is used to stem the words. The stemming technique is not used for production purposes because it is not so efficient technique and most of the time it stems the unwanted words. So, to solve the problem another technique came into the market as Lemmatization. There are various types of stemming algorithms like porter stemmer, snowball stemmer. Porter stemmer is widely used present in the NLTK library.

Lemmatization: Similar to stemming, used to stem the words into root word but differs in working. Actually, Lemmatization is a systematic way to reduce the words into their lemma by matching them with a language dictionary.

**Handling Imbalanced Dataset:**

When classes are imbalanced, standard classifiers are usually biased towards the majority class. In this case, a shift is necessary from the general paradigm that optimizes the overall classification accuracy to one that emphasizes the trade-off between precision and recall.

Dealing with imbalanced data in classification: Data augmentation is commonly used in computer vision. In computer vision, we can almost certainly flip, rotate, or mirror an image without risk of changing the original label. However, in natural language processing (NLP), the story is totally different. Changing one word has the potential to change the meaning of the entire sentence. So we can’t come up with easy rules for data augmentation. The simplest way to fix imbalanced dataset is simply balancing them by oversampling instances of the minority class or undersampling instances of the majority class. Using advanced techniques like SMOTE (Synthetic Minority Over-sampling Technique) will help us create new synthetic instances from minority class.

* Undersampling: An effort to eliminate data point from the majority class randomly until the classes are balanced. There is a likelihood of information loss which might lead to poor model training.
* Oversampling: This is the process to replicate minority class instances randomly. This approach can overfit and lead to inaccurate predictions on test data.
* SMOTE on the embeddings: SMOTE is an oversampling technique where the synthetic samples are generated for the minority class. This algorithm helps to overcome the overfitting problem posed by random oversampling. It focuses on the feature space to generate new instances with the help of interpolation between the positive instances that lie together. SMOTE is works on numeric data. But in our case, we must deal with text data. So, to apply SMOTE on the text data, we need to first convert the text data into numeric vectors and then apply SMOTE on the numeric data that we get as result of the vectorization process. Also, we need to make it a point to apply SMOTE only on the train split of the data to avoid unwanted overfitting and Data leakage. So, SMOTE should be applied after the train-test split of the data.
* Data Augmentation: Imbalanced text data can be dealt with the help Data Augmentation. It describes applying transformations to our original labelled examples to construct new data for the training set. This simply means we want to generate more data and more examples from our current dataset. There are different methods of text augmentation:
* Synonym replacement
* Lexical based replacement
* Back translation
* Generative models

NLPAug is a python library that can be used for textual augmentation. NLPAug provides three different types of augmentation:

1. Character level augmentation
2. Word level augmentation
3. Flow/ Sentence level augmentation
4. Deciding Models and Model Building. Based on the nature of the problem, decide what algorithms will be suitable and why? Experiment with different algorithms and get the performance of each algorithm.

Based on the nature of the problem which was mentioned under the questionnaire 1, this problem is classification based problem. So we have to choose such algorithms which can help us to solve classification problems. This problem would be solved by implementing Machine Learning (ML) models & Deep Learning (DL) models. Following algorithms will be implemented under each model.

**Machine Learning (ML) models:**

Logistic Regression algorithm:

* Logistic regression is Machine Learning algorithms which comes under the Supervised Learning technique. It is used for predicting the categorical dependent variable using a given set of independent variables.
* Logistic regression predicts the output of a categorical dependent variable. Therefore the outcome must be a categorical or discrete value. It can be either Yes or No, 0 or 1, true or False, etc. but instead of giving the exact value as 0 and 1, it gives the probabilistic values which lie between 0 and 1.
* Logistic Regression is much similar to the Linear Regression except that how they are used. Linear Regression is used for solving Regression problems, whereas Logistic regression is used for solving the classification problems.
* In Logistic regression, instead of fitting a regression line, we fit an "S" shaped logistic function, which predicts two maximum values (0 or 1).
* The curve from the logistic function indicates the likelihood of something such as whether the cells are cancerous or not, a mouse is obese or not based on its weight, etc.
* Logistic Regression is a significant machine learning algorithm because it has the ability to provide probabilities and classify new data using continuous and discrete datasets.
* Logistic Regression can be used to classify the observations using different types of data and can easily determine the most effective variables used for the classification.

Naive Bayes Classifier algorithm:

* Naive Bayes algorithm is a supervised learning algorithm which is based on Bayes theorem and used for solving classification problems.
* It is mainly used in text classification that includes a high-dimensional training dataset.
* Naive Bayes Classifier is one of the simple and most effective Classification algorithms which helps in building the fast machine learning models that can make quick predictions.
* It is a probabilistic classifier which means it predict on the basis of the probability of an object.
* Some popular examples of Naive Bayes Algorithm are spam filtration, Sentimental analysis, and classifying articles.

Random Forest algorithm:

* Random Forest is machine learning algorithm that comes under supervised learning technique. It can be used for both Classification and Regression problems in ML. It is based on the concept of ensemble learning, which is a process of combining multiple classifiers to solve a complex problem and to improve the performance of the model.
* Random Forest is a classifier that contains a number of decision trees on various subsets of the given dataset and takes the average to improve the predictive accuracy of that dataset.
* Instead of relying on one decision tree, the random forest takes the prediction from each tree and based on the majority votes of predictions, and it predicts the final output.
* The greater number of trees in the forest leads to higher accuracy and prevents the problem of overfitting.

Support Vector Machine (SVM) algorithm:

* Support Vector Machine or SVM is Supervised Learning algorithms, which is used for Classification as well as Regression problems. It is primarily used for Classification problems in Machine Learning.
* The goal of the SVM algorithm is to create the best line or decision boundary that can segregate n-dimensional space into classes so that we can easily put the new data point in the correct category in the future. This best decision boundary is called a hyperplane.
* SVM chooses the extreme points/vectors that help in creating the hyperplane. These extreme cases are called as support vectors, and hence algorithm is termed as Support Vector Machine.

K-Nearest Neighbor (KNN) algorithm:

* K-Nearest Neighbor is Machine Learning algorithms based on Supervised Learning technique.
* K-NN algorithm assumes the similarity between the new case/data and available cases and put the new case into the category that is most similar to the available categories.
* K-NN algorithm stores all the available data and classifies a new data point based on the similarity. This means when new data appears then it can be easily classified into a well suite category by using K-NN algorithm.
* K-NN algorithm can be used for Regression as well as for Classification but it is mostly used for the Classification problems.
* K-NN is a non-parametric algorithm, which means it does not make any assumption on underlying data.
* It is also called a lazy learner algorithm because it does not learn from the training set immediately instead it stores the dataset and at the time of classification, it performs an action on the dataset.
* KNN algorithm at the training phase just stores the dataset and when it gets new data, then it classifies that data into a category that is much similar to the new data.

**Deep Learning (DL) models:**

Long Short-Term Memory (LSTM) algorithm:

* Long short-term memory (LSTM) is an artificial recurrent neural network (RNN) architecture used in the field of deep learning.
* Unlike standard feed forward neural networks, LSTM has feedback connections. It can process not only single data points (such as images), but also entire sequences of data (such as speech or video).
* A common LSTM unit is composed of a cell, an input gate, an output gate and a forget gate. The cell remembers values over arbitrary time intervals and the three gates regulate the flow of information into and out of the cell.
* LSTM networks are well-suited to classifying, processing and making predictions based on time series data, since there can be lags of unknown duration between important events in a time series.
* LSTMs were developed to deal with the vanishing gradient problem that can be encountered when training traditional RNNs.
* Relative insensitivity to gap length is an advantage of LSTM over RNNs.

Bidirectional Long Short-Term Memory (Bi-LSTM) algorithm:

* Bidirectional long-short term (Bi-LSTM) memory is the process of making any neural network have the sequence information in both directions backwards (future to past) or forward (past to future).
* In bidirectional, our input flows in two directions, making a Bi-LSTM different from the regular LSTM.
* With the regular LSTM, we can make input flow in one direction, either backwards or forward. However, in bi-directional, we can make the input flow in both directions to preserve the future and the past information.

Gated Recurrent Units (GRU) algorithm:

* To solve the Vanishing-Exploding gradients problem often encountered during the operation of a basic Recurrent Neural Network, many variations were developed. One of the most famous variations is the Long Short Term Memory Network (LSTM). One of the lesser-known but equally effective variations is the Gated Recurrent Unit Network (GRU).
* GRU consists of only three gates and does not maintain an Internal Cell State. The information which is stored in the Internal Cell State in an LSTM recurrent unit is incorporated into the hidden state of the Gated Recurrent Unit. This collective information is passed onto the next Gated Recurrent Unit. The different gates of a GRU are as described below:-
* Update Gate: It determines how much of the past knowledge needs to be passed along into the future. It is analogous to the Output Gate in an LSTM recurrent unit.
* Reset Gate: It determines how much of the past knowledge to forget. It is analogous to the combination of the Input Gate and the Forget Gate in an LSTM recurrent unit.
* Current Memory Gate: It is often overlooked during a typical discussion on Gated Recurrent Unit Network. It is incorporated into the Reset Gate just like the Input Modulation Gate is a sub-part of the Input Gate and is used to introduce some non-linearity into the input and to also make the input Zero-mean. Another reason to make it a sub-part of the Reset gate is to reduce the effect that previous information has on the current information that is being passed into the future.

Bidirectional Encoder Representations from Transformers (BERT) algorithm:

* BERT stands for Bidirectional Encoder Representations from Transformers. It is designed to pre-train deep bidirectional representations from unlabeled text by jointly conditioning on both left and right context. As a result, the pre-trained BERT model can be fine-tuned with just one additional output layer to create state-of-the-art models for a wide range of NLP tasks.
* BERT is based on the Transformer architecture.
* BERT is pre-trained on a large corpus of unlabeled text including the entire Wikipedia (that’s 2,500 million words!) and Book Corpus (800 million words). This pre-training step is half the magic behind BERT’s success. Because as we train a model on a large text corpus, our model starts to pick up the deeper and intimate understandings of how the language works.
* BERT is a “deeply bidirectional” model. Bidirectional means that BERT learns information from both the left and the right side of a token’s context during the training phase.

1. How to improve your model performance? What are the approaches you can take to improve your model? Can you do some feature selection, data manipulation and model improvements?

Following measures would be taken into consideration after model creation. Based on the outcome of measures the model performance would be considered.

Training versus Validation Accuracy graph:

This graph is helpful to study the accuracy of Training data versus Validation data. In this graph Accuracy is on y-axis & number of epochs on x-axis. Such graph helps us to know at which epoch we should stop the training. Usually Training accuracy should be better than Validation accuracy. But when Validation accuracy is better than Training accuracy then it means that the data augmentation we are applying to the training data is making the task significantly harder for the network. (i.e. we are applying the augmentation only to your training data, not your validation data).

There are either of the two ways to resolve this:

1. If we decrease the range of the augmentation transformations then we should be able to see the training and validation loss get closer together.
2. If we apply the exact same augmentation transformations to the validation data as we did for training data, then we should be able to see the validation accuracy drop below the training accuracy.

Training versus Validation Loss graph:

Loss function quantifies how “good” or “bad” a given predictor is at classifying the input data points in a dataset. The smaller the loss, the better a job the classifier is at modeling the relationship between the input data and the output targets.

In this graph Loss is on y-axis & number of epochs on x-axis. Such graph helps us to know when the model is going to be over fitted or not. When Training loss is constantly decreasing but the validation loss isn't then it means that model is completely overfitting. This means that the current model is complex enough to 'memorize' the patterns in the training data. In such situations, we need to regularize your model. One can try all or either of the combinations mentioned below:

1. Reduce the layers of the neural network.
2. Reduce the number of neurons in each layer of the network to reduce the number of parameters.
3. Add dropout and tune its rate.
4. Use L2 normalization on the parameter weights and tune the lambda value.
5. If possible add more data for training.
6. Adding more data may not always be economically viable option so 'Data Augmentation' can play a big role here. Data Augmentation depends on the type of input to the neural nets.

Confusion Matrix:

A confusion matrix is a table that is often used to describe the performance of a classification model (or "classifier") on a set of test data for which the true values are known.

Let's consider an example confusion matrix here.

|  |  |  |
| --- | --- | --- |
| n=165 | Predicted: NO | Predicted: YES |
| Actual: NO | 50 | 10 |
| Actual: YES | 5 | 100 |

* There are two possible predicted classes: "yes" and "no". If we were predicting the presence of a disease, for example, "yes" would mean they have the disease, and "no" would mean they don't have the disease.
* The classifier made a total of 165 predictions (e.g., 165 patients were being tested for the presence of that disease).
* Out of those 165 cases, the classifier predicted "yes" 110 times, and "no" 55 times.
* In reality, 105 patients in the sample have the disease, and 60 patients do not.

The basic terms being used are:

* True Positives (TP): These are cases in which we predicted yes (they have the disease), and they do have the disease.
* True Negatives (TN): We predicted no, and they don't have the disease.
* False Positives (FP): We predicted yes, but they don't actually have the disease. (Also known as a "Type I error.")
* False Negatives (FN): We predicted no, but they actually do have the disease. (Also known as a "Type II error.")

|  |  |  |  |
| --- | --- | --- | --- |
| n=165 | Predicted: NO | Predicted: YES |  |
| Actual: NO | TN = 50 | FP = 10 | 60 |
| Actual: YES | FN = 5 | TP = 100 | 105 |
|  | 55 | 110 |  |

Below is a list of rates that are often computed from a confusion matrix:

* Accuracy: Overall, how often is the classifier correct?
  + (TP+TN)/total = (100+50)/165 = 0.91
* Misclassification Rate: Overall, how often is it wrong?
  + (FP+FN)/total = (10+5)/165 = 0.09
  + equivalent to 1 minus Accuracy
  + also known as "Error Rate"
* True Positive Rate: When it's actually yes, how often does it predict yes?
  + TP/actual yes = 100/105 = 0.95
  + also known as "Sensitivity" or "Recall"
* False Positive Rate: When it's actually no, how often does it predict yes?
  + FP/actual no = 10/60 = 0.17
* True Negative Rate: When it's actually no, how often does it predict no?
  + TN/actual no = 50/60 = 0.83
  + equivalent to 1 minus False Positive Rate
  + also known as "Specificity"
* Precision: When it predicts yes, how often is it correct?
  + TP/predicted yes = 100/110 = 0.91
* Prevalence: How often does the yes condition actually occur in our sample?
  + actual yes/total = 105/165 = 0.64

Classification Report:

The classification report displays the precision, recall, F1, and support scores for the model.

There are four ways to check if the predictions are right or wrong:

* TN / True Negative: the case was negative and predicted negative
* TP / True Positive: the case was positive and predicted positive
* FN / False Negative: the case was positive but predicted negative
* FP / False Positive: the case was negative but predicted positive

*Precision*: What percent of your predictions were correct?

* Precision is the ability of a classifier not to label an instance positive that is actually negative. For each class, it is defined as the ratio of true positives to the sum of a true positive and false positive.
* Precision: Accuracy of positive predictions.
* Precision = TP/(TP + FP)

*Recall*: What percent of the positive cases did you catch?

* Recall is the ability of a classifier to find all positive instances. For each class it is defined as the ratio of true positives to the sum of true positives and false negatives.
* Recall: Fraction of positives that were correctly identified.
* Recall = TP/(TP+FN)

*F1 score*: What percent of positive predictions were correct?

* The F1 score is a weighted harmonic mean of precision and recall such that the best score is 1.0 and the worst is 0.0. F1 scores are lower than accuracy measures as they embed precision and recall into their computation. As a rule of thumb, the weighted average of F1 should be used to compare classifier models, not global accuracy.
* F1 Score = 2\*(Recall \* Precision) / (Recall + Precision)

*Support*:

* Support is the number of actual occurrences of the class in the specified dataset. Imbalanced support in the training data may indicate structural weaknesses in the reported scores of the classifier and could indicate the need for stratified sampling or rebalancing. Support doesn’t change between models but instead diagnoses the evaluation process.

Feature Selection:

For few columns like Unnamed, Date, Countries & Local it was very straight forward to remove those columns. After performing Principal Component Analysis (PCA) it was clear that we need to remove Industry Sector, Genre, Employee or Third Party & Critical Risk columns from the dataset.

Data manipulation:

The original dataset was imbalanced. Potential Accident Level VI had only 1 record hence we deleted that record. For rest of the dataset we balanced the data by performing data augmentation using “NLPAug” package. By doing this we were able to create different variation of description in order to balance dataset.