Natural Language processing

**“Machine Learning Based Safety Risk Prediction from Industrial Incident Descriptions”**

A Capstone Project

***Project Submitted by***

Aryan Kasat

Ganesh Moorthi

Gautam Kumar Jha

Rushi Kesava Reddy Sathani

*In fulfilment of the requirements for the award of*

**Post Graduate Program in**

**Artificial Intelligence & Machine Learning**

from:

Great Learning

**Mentor:**

Aniket Chhabra

September 2025

[**Abstract 3**](#_6otied80k5rr)

[**1. Introduction 3**](#_iky8v5wndz79)

[1.1 Problem Statement 3](#_6fp051r4dprc)

[1.2 Importance of Safety Risk Prediction 4](#_t0yqa1i0y14t)

[**2. Background and Literature Review 4**](#_byoi47uemqle)

[**3. Data Description and Exploratory Data Analysis (EDA) 5**](#_jxy6d3mkwkzq)

[3.1 Dataset Overview and Data Cleaning 6](#_o0dzqtxits9a)

[3.2 Key Observations from Exploratory Data Analysis (EDA) 7](#_t9anq2rpsr5y)

[3.2.1 Univariate Analysis 7](#_hu3b5t32iyla)

[3.2.2 Bivariate and Multivariate Analysis 9](#_9zsnymiok58r)

[3.2.3 Temporal and Textual Analysis 11](#_wzo98qsa5r4m)

[**4. Methodology 13**](#_2pm85d9kx67f)

[4.1 Data Cleaning and Preprocessing 14](#_j6etfmhbom46)

[4.1.1 Feature Engineering and Data Preparation 14](#_o5y87hqbsqqh)

[4.1.2 Text Preprocessing Pipeline 14](#_a607dzhpprz6)

[4.1.3 Data Splitting Strategy 17](#_wzxe1qhgvihp)

[4.2 Feature Engineering: The Path to Effective Representation 17](#_uw9zwyrqmipa)

[4.2.1 Text Vectorization Models 17](#_ultiekfyuu6r)

[4.2.2 Categorical and Target Feature Encoding 19](#_myxkrpvuuwrv)

[4.2.3 Deep Learning and Sequential Architectures 19](#_x6j1307m2mx4)

[**5. Results and Evaluation 20**](#_nugkz5uht7w5)

[5.1 Model Experiments and Results 20](#_ku02kjf1zdo4)

[5.1.1 TF-IDF Models: The Overfitting Challenge 20](#_88nrit23dv32)

[5.1.2 Word Embeddings (Word2Vec, GloVe): Incremental Improvements 21](#_rc2oth8v8ksd)

[5.1.3 Sentence Transformer: The Optimal Balance 22](#_3kwmm2pzzdx)

[5.2 Model Performance and Comparison 25](#_nlmp4wlqkk84)

[5.2.1 Analysis of Base Models 25](#_9vbrouv8f5uj)

[5.2.2 Impact of Hyperparameter Tuning 27](#_nlixshexb4yp)

[5.2.3 Tuned Neural Network Superiority 27](#_f3pqad7d6ddw)

[5.2.4 All Model performance comparison 28](#_r561eoo4yjat)

[5.3 Final Model Selection and Reasoning 34](#_n7hjwkidbaef)

[5.3.1 Final Model Selection Rationale 35](#_28ujlqdxrv3s)

[**6. Discussion and Conclusion 35**](#_yoeb7zxvdpoe)

[6.1 Key Insights 36](#_2ned1mohrxpg)

[6.1 Accident Classifier Chatbot: Deployment Prototype 37](#_ez8j0w8l8tpl)

[6.1.1 Prototype Features and Architecture 37](#_8oj9d3qtqo5r)

[6.2 Recommendations 38](#_tlmtpl7lltqs)

[6.2.1 Return on Investment (ROI) and Business Impact 38](#_eku285pu2meo)

[6.2.2 Strategy and Metrics 39](#_wzhdlvw6fbm7)

[6.2.3 Data and Model Future Prospects 39](#_bg6azbow95ao)

[**7. References 40**](#_hvmb80ul0w9i)

# Abstract

This capstone project addresses a critical challenge in industrial safety by creating a machine learning system that predicts **accident severity** from textual incident descriptions and structured metadata. The core objective is to automate the manual and time-consuming process of risk assessment, allowing safety professionals to more efficiently prioritize incidents and implement **proactive mitigation strategies** aligned with **ISO 45001** standards. The project utilized a comprehensive **Natural Language Processing (NLP) pipeline** to transform unstructured narratives using techniques like Word2Vec and Sentence Transformer embeddings. Through rigorous evaluation of numerous models, including linear classifiers, tree-based models, and deep learning architectures (LSTM and Neural Networks), the study found that **tuned deep learning models** provided the optimal solution. The final selected model, a **Word2Vec + Tuned Neural Network**, successfully demonstrates that linguistic patterns, when integrated with structured operational features, are strong predictors of accident severity. This solution provides a robust and reliable foundation for a real-world, decision-support system to enhance overall workplace safety.

# 1. Introduction

## 1.1 Problem Statement

Workplace accidents remain a persistent challenge for industries worldwide, despite continuous improvements in safety systems and regulatory compliance. The dataset provided consists of 425 historical accident records from 12 manufacturing plants across three countries. Each record describes the nature of the accident, associated risks, and severity levels.

Currently, identifying potential safety risks from incident descriptions is largely a manual and time-consuming process, often performed by safety officers after incidents occur. This reactive approach can delay the implementation of corrective actions and increase the likelihood of repeat incidents.

The objective of this project is to design a machine learning-based system that can automatically analyze textual incident descriptions, predict the accident severity level, highlight critical risks, and assist safety professionals in taking proactive measures. Ultimately, there will be an attempt to integrate the solution into a chatbot interface that can interact with users and provide immediate insights.

## 1.2 Importance of Safety Risk Prediction

The automation of safety risk prediction holds immense value for modern industry. Beyond the profound human and ethical implications of preventing injury and loss of life, workplace incidents also severely impact production downtime, legal liability, and long-term financial costs. The ability to identify potential risks early allows for several key benefits:

* **Proactive Hazard Mitigation:** Organizations can move from a reactive to a proactive safety culture, addressing risks before they escalate.
* **Better Allocation of Safety Resources:** Resources can be strategically deployed to high-risk areas identified by the model.
* **Reduction in Lost Time Injury Frequency Rate (LTIFR):** By predicting and preventing severe incidents, the model can directly contribute to improving a company's safety performance metrics.

# 2. Background and Literature Review

The foundation of this project lies at the intersection of Natural Language Processing (NLP) and Machine Learning (ML) applied to a real-world business problem: industrial safety risk prediction. Prior research has demonstrated the efficacy of text classification models in a variety of domains, from sentiment analysis to medical diagnostics. However, the application of these techniques to industrial safety incident reports presents a unique set of challenges, particularly the domain-specific jargon and the **highly imbalanced nature of accident severity data** (where severe incidents are rare). Addressing this domain requires a solution that not only predicts risk but also supports regulatory objectives, such as the **proactive risk mitigation required by the ISO 45001 standard** for occupational health and safety.

Traditional keyword-based systems for risk identification are often rigid and fail to capture the nuanced context of a narrative. For example, a simple keyword search for "fire" may flag an incident where a small, controlled flame was used, while missing a more severe incident described using different vocabulary. This highlights the need for a more sophisticated, learning-based approach that can process the full narrative and the associated operational context.

Early text classification methods relied on simple vectorization techniques like Bag-of-Words (BoW) and Term Frequency-Inverse Document Frequency (TF-IDF), often combined with classical machine learning models such as Naive Bayes, Logistic Regression, and Support Vector Machines (SVM). These models are computationally efficient and highly interpretable, but they struggle to capture the semantic meaning of words and sentences (Jurafsky & Martin). Furthermore, due to the project's **class imbalance**, model performance must be rigorously evaluated using the **Macro-Averaged F1-score** to ensure reliable prediction across all severity levels.

The more recent development of deep learning models has revolutionized NLP. Embeddings like **Word2Vec** and **GloVe** represent words as dense vectors, capturing semantic and syntactic relationships. More advanced architectures like **Recurrent Neural Networks (RNNs)** and their variants, such as **Long Short-Term Memory (LSTMs)**, are specifically designed to process the **sequential nature of text**, making them ideal for understanding how word order in an incident narrative impacts meaning. This capability is crucial, as the chronology of events often dictates the final severity. Most recently, transformer-based models like **Sentence Transformer** have set new benchmarks by understanding context across an entire sentence (Reimers & Gurevych, 2019). A critical methodological decision was to explore how these **Recurrent architectures** could be tuned to effectively integrate sequential textual embeddings with structured operational features, aiming for a robust, holistic predictive solution.

# 3. Data Description and Exploratory Data Analysis (EDA)

## 3.1 Dataset Overview and Data Cleaning

The dataset consists of 425 rows, each corresponding to an individual incident report. The dataset is provided in a clean format with no missing values, a rarity in real-world data science projects. However, a preliminary analysis revealed that seven duplicate rows were present, which were removed to ensure the integrity of the training process and prevent data leakage. The raw dataset includes the following key columns:

| **Column Name** | **Description** | **Type** | **Role in Project** |
| --- | --- | --- | --- |
| **Data** | Timestamp of accident occurrence | Datetime | Can be used for time-series analysis (accident trends) |
| **Countries** | Country where the accident occurred (anonymized) | Categorical | Used as categorical feature in ML models |
| **Local** | Plant location or city (anonymized) | Categorical | Used for plant-level risk insights |
| **Industry Sector** | Sector/department where the accident happened | Categorical | Feature to improve prediction model |
| **Accident Level** | Severity level of the accident (I = minor, VI = very severe) | Ordinal (I–VI) | **Target variable** for classification |
| **Potential Accident Level** | Maximum possible severity if conditions were worse | Ordinal (I–VI) | Can be used as additional feature |
| **Genre** | Gender of the injured person | Categorical | Optional feature |
| **Employee or Third Party** | Whether injured person is employee or contractor | Categorical | Feature — can influence risk profile |
| **Critical Risk** | Categorized risk type (e.g., fall, fire, chemical exposure) | Categorical | Can be used for insights or secondary classification |
| **Description** | Detailed narrative of accident (free text) | Text | **Key input** for NLP processing & severity prediction |

Initial data preparation involved standardizing column names for consistency and converting the Data column to a proper datetime format for temporal analysis. The removal of duplicates resulted in a final dataset of 418 unique, clean records.

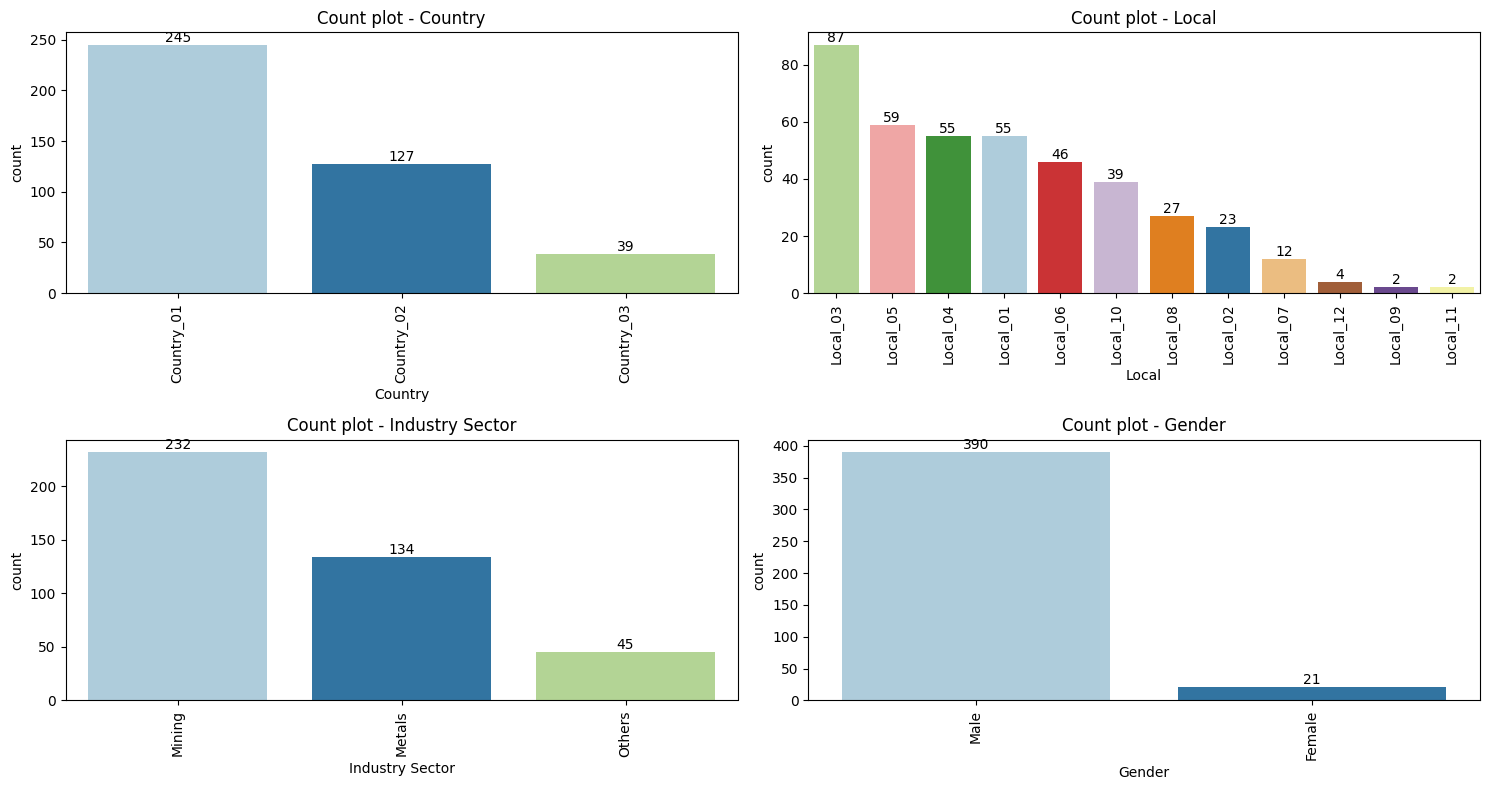
## 3.2 Key Observations from Exploratory Data Analysis (EDA)

EDA was a crucial step in understanding the data's characteristics and guiding subsequent modeling decisions. This comprehensive analysis was divided into univariate, bivariate, and multivariate examinations to uncover key insights and validate assumptions.

### 3.2.1 Univariate Analysis

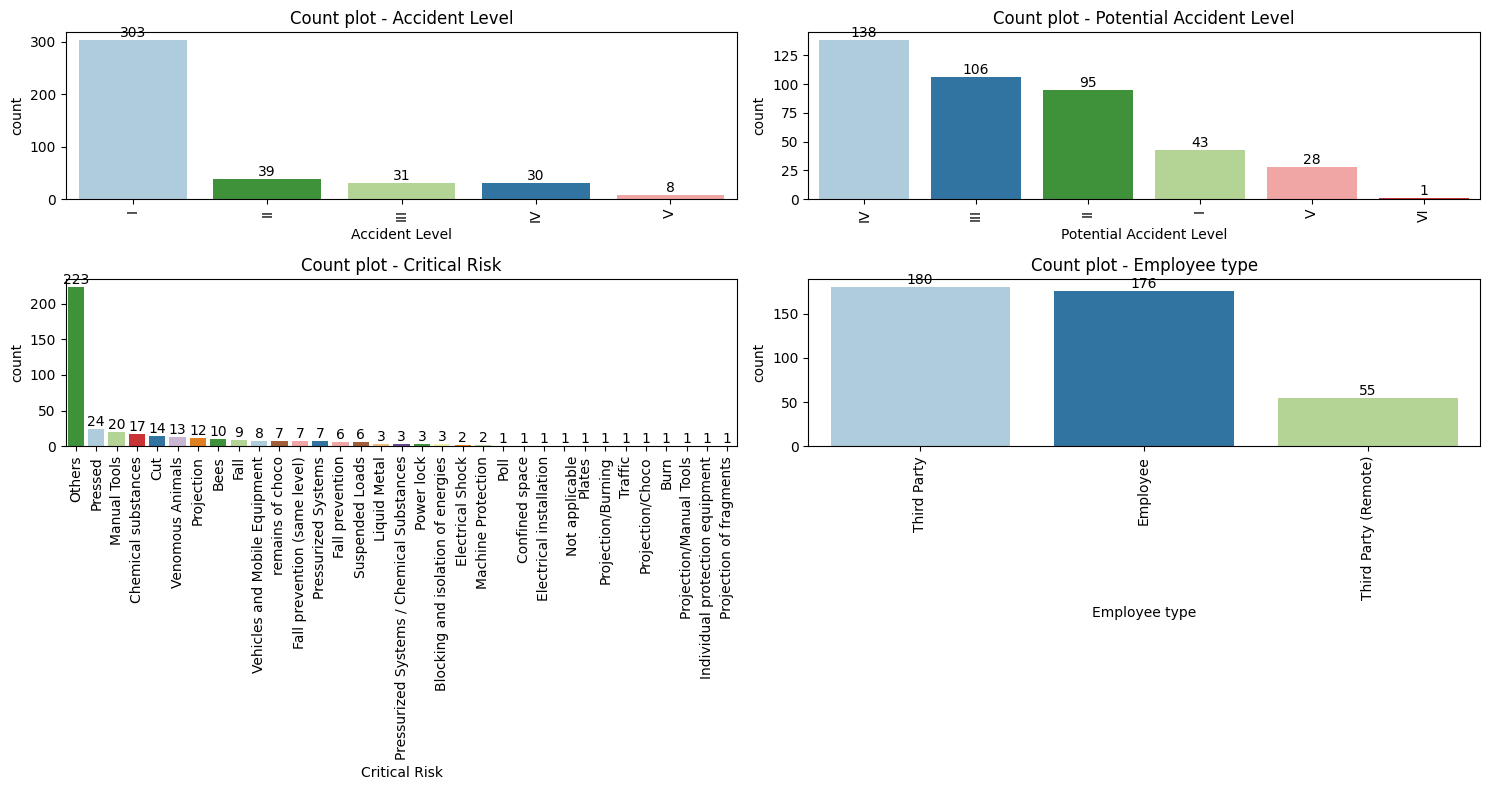
A primary finding was the significant **class imbalance** within the Accident Level target variable. A count plot of the data confirmed that Level I incidents are overwhelmingly the most frequent, comprising over half of the dataset, while the more severe levels (V and VI) are extremely rare. This imbalance is a critical challenge, as a model trained on such a dataset might learn to simply predict the majority class (Level I) to achieve high accuracy, while failing to identify the rare, but critically important, severe incidents. This insight necessitated the use of a more robust evaluation metric like the **Weighted-averaged F1-score**, which treats all classes equally, regardless of their frequency.

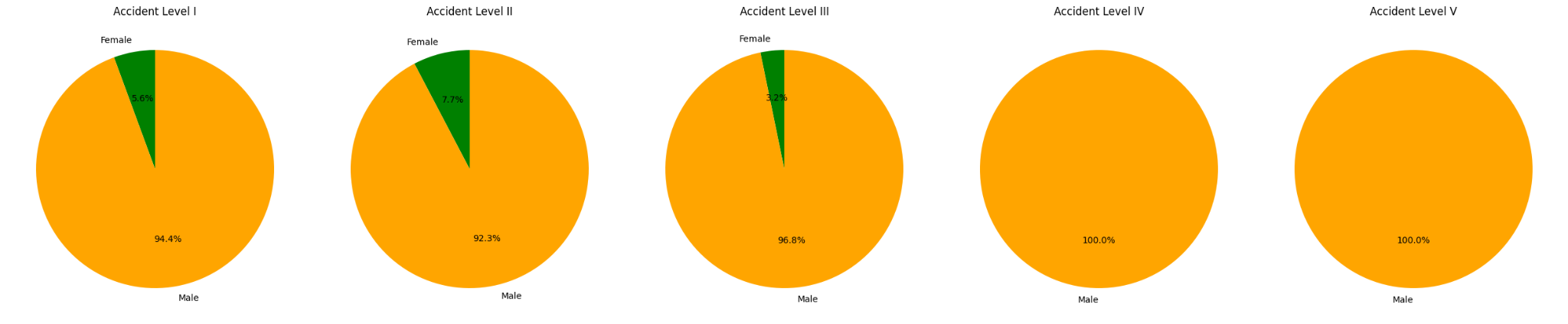




Further analysis of individual variables revealed several key characteristics:

* **Geographic and Plant Distribution:** The majority of accidents (59%) occurred in Country\_01, followed by Country\_02 (31%), and with a notably low proportion in Country\_03 (10%). This suggests that safety strategies in Country\_03 may be more effective and could serve as a benchmark. Similarly, City Local\_03 recorded the majority of incidents, while others like Local\_09 and Local\_11 had very few.
* **Demographics and Industry:** The dataset shows a heavy dominance of the male workforce, which is likely a reflection of the male-dominated nature of the industries surveyed, particularly the **Mining** sector, where the highest number of incidents occurred. The number of incidents was also highest among **Third Parties**, possibly due to limited training and less familiarity with the work environment.
* **Incident Characteristics:** The majority of incidents were classified as **Potential Accident Level IV**, indicating that many incidents, while not severe, had the potential to be. For many incidents, the Critical Risk was not identified, further emphasizing the importance of the free-text description for understanding risk.

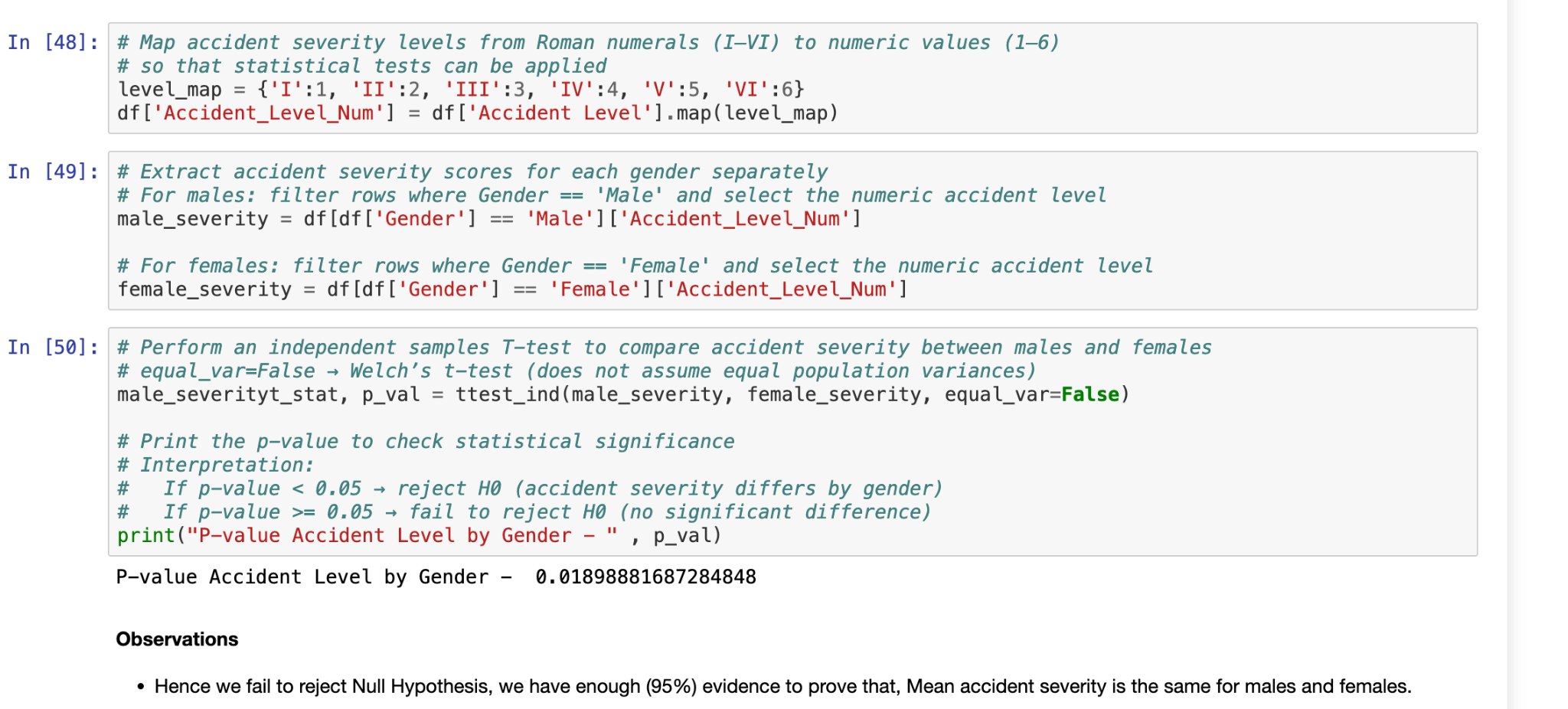




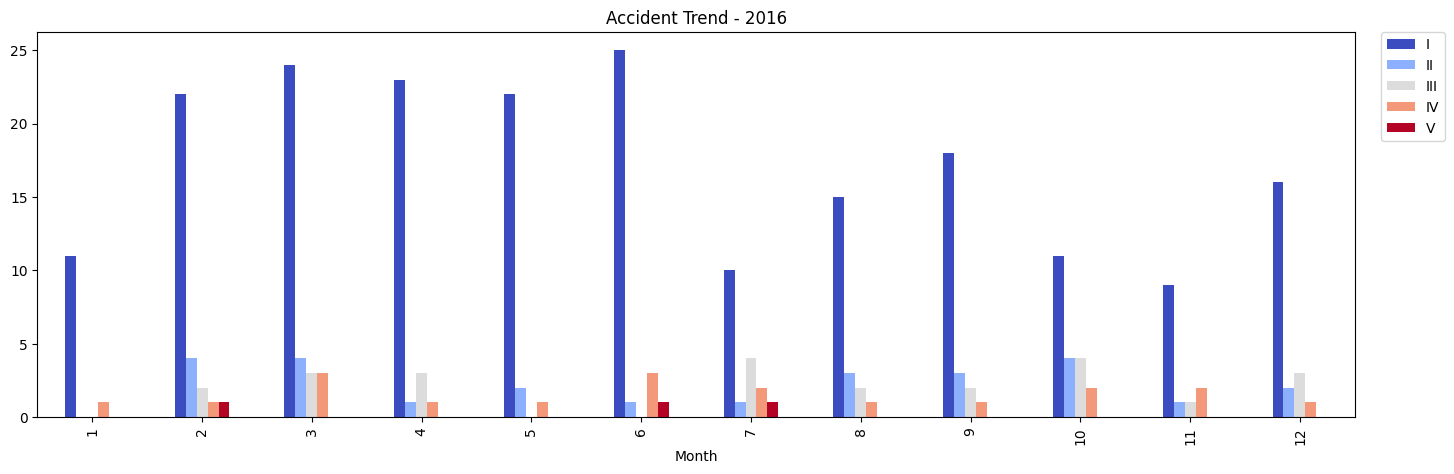
### 3.2.2 Bivariate and Multivariate Analysis

To understand the relationships between variables, a series of bivariate analyses and statistical tests were performed.

* **Country vs. Industry Sector:** A **Z-test of proportions** was conducted to determine if the distribution of industry sectors was statistically different across countries. The p-values (Mining and Metals: 2.2e-42; Metals and Others: 2.2e-11) were both well below the 0.05 significance level, leading to the rejection of the null hypothesis. This provides strong evidence that the proportion of industry sectors is, in fact, different across countries, which is important for understanding the context of incident frequency. For instance, Country\_01 has the highest proportion of the Mining sector, which accounts for most incidents, while Country\_03 has no incidents in this sector.
* **Gender vs. Accident Severity:** A **T-test** was performed to compare the mean accident severity between male and female workforces. The p-value was 0.0189, which is less than 0.05. This allowed for the rejection of the null hypothesis that the mean accident severity is the same for both genders, proving that the mean accident severity is statistically different between the male and female workforces. However, a subsequent **Chi-square test of independence** (p-value = 0.56) showed no statistical dependence between gender and accident type, indicating that while severity may differ, the type of accident is not influenced by gender.

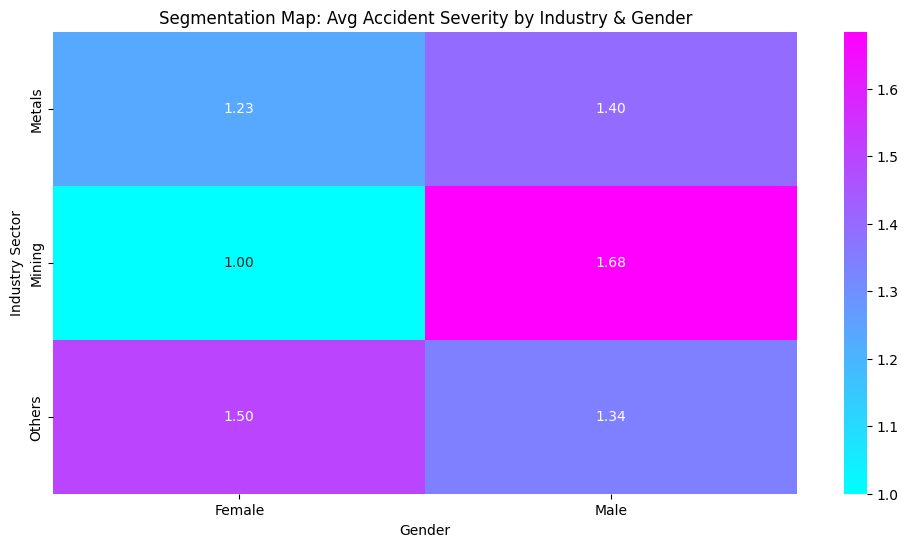


* **Criticality vs. Employee Type:** The analysis showed that the proportion of critical incidents is higher among **Third Party (Remote)** and **Third Party** groups, while minor incidents are more common among **Employees**. This suggests that proactive safety measures and training may be less standardized for external parties, highlighting a critical area for intervention.

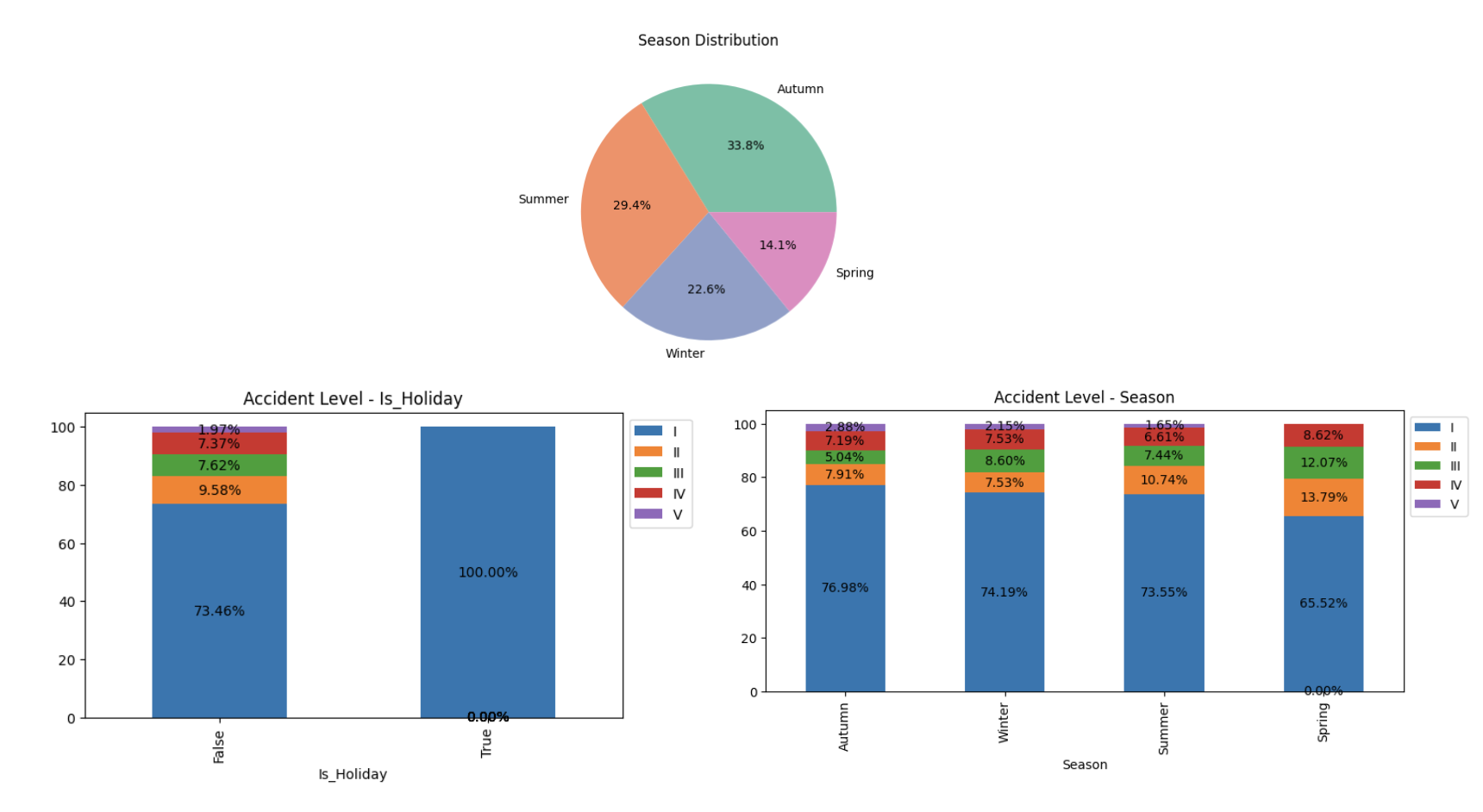


### 3.2.3 Temporal and Textual Analysis

Temporal analysis revealed distinct patterns. The number of incidents in **Country\_01** remains consistently high throughout the period, which may indicate a higher concentration of industries or lower safety standards compared to other countries. The number of incidents with **Accident Level I** is also consistently high, particularly in **February, March, April, and June**, indicating a seasonal or cyclical trend. Conversely, as criticality increases, the number of incidents tends to decrease, which is a positive sign that precautionary measures are effectively in place for more critical types of incidents.



The **temporal analysis** of accident frequency reveals a strong reliance on operational tempo. **Autumn is clearly the most accident-prone season**, recording the highest number of accidents (139 cases) and a slight increase in high-severity incidents (Levels IV & V). Conversely, **Spring records the fewest accidents** (58 cases), potentially indicating safer conditions or increased precautions during that period. This pattern is directly connected to the bank holiday data, which showed that the **vast majority of incidents (99%) occur on regular working days**. The fact that only four minor accidents happened on bank holidays indicates that **operational exposure and high workload periods** are the primary drivers of risk intensity, rather than seasonal or environmental conditions alone. This combined insight suggests future investigations should focus on **identifying peak operational stressors and compliance during the high-risk Autumn season**.



Textual analysis provided valuable insights that justified the need for an NLP approach. The **word cloud** and **bigram analysis** revealed common themes, with generic words dominating the top frequencies, but domain-specific terms like "slip," "fall," "hand," and "machine" appearing prominently, aligning with common types of workplace hazards. A key insight was that more critical incidents tend to have **longer descriptions**, indicating that the length and detail of a narrative could be a useful feature for the model. The analysis of bigrams also proved insightful, with phrases like "slipped floor" and "cut hand" being common in low-severity incidents, while "chemical exposure" and "fire explosion" were strongly correlated with more severe cases. A heatmap visualization clearly showed that the language patterns shift as severity increases, which provided the justification for including both words and bigrams in the feature engineering process.

# 4. Methodology

The methodological approach for this project was a systematic pipeline designed to transform raw, unstructured text into a robust classification model.

## 4.1 Data Cleaning and Preprocessing

The Description column, being unstructured text, required a meticulous preprocessing pipeline to ensure the data was clean and consistent for downstream modeling. Each step was executed with a clear purpose, justified by the nature of the raw data.

### 4.1.1 Feature Engineering and Data Preparation

Before text preprocessing, the overall dataset structure was prepared for analysis. The target variable was identified as Accident Level, which represents the severity of each incident using Roman numerals (I–VI), where **I** is minor and **VI** is very severe. All other fields served as **input features**, including both categorical data (Country, Local, Industry Sector, Gender, Employee type, Critical Risk) and textual data (Description).

To enhance the dataset's analytical power, several new features were engineered. The original Date column was decomposed into more granular temporal features: Year, Month, and Day and a derived **Season** feature to capture the seasonality in accident patterns. This allowed models to learn from potential **seasonal or temporal patterns** in accident frequency. A new boolean feature, Is\_Weekend, was also created to capture whether an incident occurred on a weekend. The original Date and the redundant year\_month and Potential Accident Level features were then dropped to simplify the dataset and avoid redundancy. Finally, a derived feature, Desc\_Len, was added to represent the length of the incident description, which was found to be a potential indicator of incident criticality. This comprehensive setup enabled models to learn from a rich combination of industrial context, accident narratives, and temporal information.

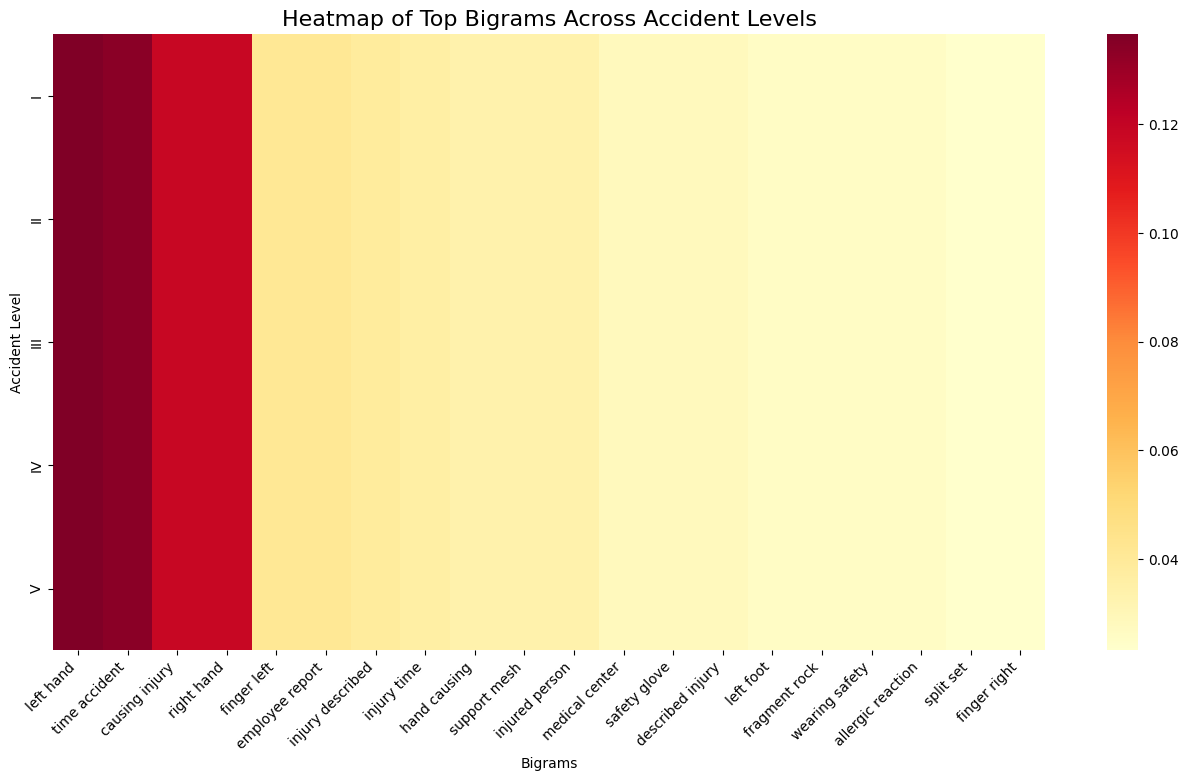
### 4.1.2 Text Preprocessing Pipeline

The core of the preprocessing involved transforming the raw Description text. The pipeline was designed to systematically clean the data, with each step addressing a specific issue to prepare the text for effective feature extraction.

* **Lowercasing**: All text was converted to lowercase. As observed, this is a fundamental step that reduces the vocabulary size and prevents the model from treating different casings of the same word (e.g., "Fall," "fall," and "FALL") as separate tokens, ensuring that the feature space is not unnecessarily large and redundant.
* **Noise Removal**: Punctuation, numbers, parentheses, and other special characters were systematically removed from the text. These elements rarely carry semantic meaning for this classification task and can introduce unnecessary noise into the feature space. Additionally, any extra whitespace was stripped to ensure a clean, uniform output.
* **Accent Removal**: Accented characters, particularly common in Portuguese text within the dataset (e.g., "á" to "a"), were converted to their unaccented counterparts. This step is crucial for standardizing the text and ensuring consistency across different language variations that might appear in a multinational company's data.
* **Stopword Removal**: Common words like articles ("the," "a"), prepositions ("of," "in"), and pronouns that frequently appear throughout all documents in the corpus were removed. These **stopwords** have no predictive power for accident severity. Removing them reduced the dimensionality of the feature space and sped up model training by allowing the model to focus on more informative, domain-specific terms.
* **Lemmatization**: Tokens were reduced to their base or dictionary form (e.g., "slipped" and "slipping" are both reduced to "slip"). This step consolidates the vocabulary, reducing the sparsity of the feature vectors and allowing the model to recognize that different forms of a word convey the same core meaning. As observed, this process successfully transformed words like "supports" to "support" and "fingers" to "finger," which are crucial for consistent feature representation.



The result of this meticulous preprocessing pipeline was a clean, standardized, and information-rich dataset, ready for feature engineering and model training. The cleaned descriptions were found to emphasize **injury context, body parts, and operational roles** (e.g., "hand," "operator," "causing"), making them highly useful for **accident classification** and risk prediction.



### **4.1.3 Data Splitting Strategy**

Initially, the dataset was considered time-series data and was planned to be split using a **sequential, time-based method** (e.g., 80% for training based on earliest dates and 20% for validation/testing based on later dates). This approach is typically used to rigorously simulate a real-world deployment scenario where the model must predict future incidents based on historical data.

However, following the Feature Engineering phase, new temporal features such as **Season** and **Is\_Weekend** were explicitly extracted and added as input features. Because the relevant temporal information is now encoded within the feature set itself, and to ensure the model's ability to generalize patterns across the entire incident space (improving data diversity in the small validation set), the strategy was updated to a **stratified random split (80/20)**. This approach preserves the distribution of the target variable (**Accident Level**) across the training and testing sets while providing a more representative measure of the model's overall robustness to unseen data points from the entire population.

## 4.2 Feature Engineering: The Path to Effective Representation

The most critical step in this project was converting the preprocessed, unstructured text into a numerical format suitable for machine learning models. Several techniques were explored and evaluated to find the most effective representation for the model to learn from. This process was essential for transforming the qualitative narratives of accident reports into quantitative features.

### 4.2.1 Text Vectorization Models

A variety of text vectorization models were employed, each with its own strengths and weaknesses.

* **Bag-of-Words (BoW):** This simple model represents each accident description as a **sparse matrix of word counts**. A vocabulary of all unique words (2,091 tokens) from the training data was created, and this same vocabulary was used to transform the validation and test sets to ensure feature consistency. The resulting vectors were highly sparse, meaning most values were zeros, which confirmed that each document contained only a small subset of the total vocabulary. While easy to implement, BoW completely ignores **word order and context**, a significant limitation for understanding nuanced incident descriptions. Its high dimensionality also posed a risk of computational inefficiency and potential overfitting without further feature reduction.
* **TF-IDF (Term Frequency-Inverse Document Frequency):** This technique improves upon BoW by weighting words based on their **importance**. It assigns a higher value to words that are frequent in a specific document but rare across the entire corpus, effectively highlighting more **discriminative terms**. The vocabulary of 2,091 unique tokens remained the same as the BoW model. TF-IDF provided a more balanced feature set by down-weighting common, less-informative words and amplifying the impact of more relevant terms. The resulting vectors were still sparse but offered a more powerful representation than raw word counts.
* **Word Embeddings (Word2Vec, GloVe):** These models move beyond simple frequency to capture the **semantic and contextual meaning** of words. Word2Vec was trained on the accident descriptions as an unsupervised embedding method, generating dense vectors of **100 dimensions** for a vocabulary of 3,620 words. This approach captures contextual relationships (e.g., "repair" would be close to "maintenance"), but it can sometimes produce noisy tokens due to the specific domain. Similarly, the pre-trained GloVe model was used to represent words as 100-dimensional dense vectors, leveraging a large vocabulary (400,000 words) to capture generalized semantic relationships. The word "incident," for example, was found to be semantically close to terms like "accident" and "happened" in the GloVe space. For both Word2Vec and GloVe, each accident description was converted into a document-level vector by **averaging the embeddings** of its constituent words. These representations were more compact and semantically rich than the sparse BoW/TF-IDF vectors.
* **Sentence Transformer:** The final and most effective approach was the use of **Sentence Transformer** embeddings. Unlike traditional word embeddings that provide a vector for each word, Sentence Transformers (specifically the all-MiniLM-L6-v2 model) generate a single, high-quality, **384-dimensional dense vector** for an entire sentence. This is a critical distinction as it allows the model to capture the **full semantic meaning and context** of the incident description, rather than just the individual words. The chosen model is a compact transformer optimized for efficiency while mimicking the performance of larger models like BERT. These embeddings were then used to represent each accident description, providing a highly informative and dense input for the downstream classifier.



### 4.2.2 Categorical and Target Feature Encoding

In addition to the text features, categorical variables like Industry Sector and Critical Risk were **one-hot encoded** and concatenated with the text embeddings to provide the model with a richer set of features. This ensured that the model could learn from both the textual narrative and the structured contextual information.

Finally, the target variable, Accident Level (recorded using Roman numerals I-VI), was prepared for model training using **Label Encoding**. This process converted the categorical labels into a consistent numerical format, where y\_train, y\_val, and y\_test labels were all encoded using the same mapping to ensure alignment and proper model evaluation. This encoding preserves the order of the categories (e.g., I < II < III), which is an important consideration given the ordinal nature of accident severity.

### 4.2.3 Deep Learning and Sequential Architectures

To capture non-linear patterns and the crucial sequential dependency inherent in incident narratives, a comprehensive deep learning approach was implemented. This included three core architectures: a **Dense Feed-Forward Neural Network (DNN)** for non-linear classification on static embeddings, and two sequential models: a **Long Short-Term Memory (LSTM)** network and a simpler **Recurrent Neural Network (RNN)**.

The **RNN** and **LSTM** models were designed to process tokenized sequences derived from the text, integrating pre-trained embeddings (Word2Vec, GloVe) either as frozen or trainable layers. These models use internal memory states to capture relationships between words across the sentence structure. The **DNN** served to test non-linear classification on the mean-pooled vector embeddings. All architectures used the **Adam** optimizer, **Sparse Categorical Crossentropy** loss, and monitored **Accuracy** and **F1-score**, with extensive **hyperparameter tuning (Tuned)** applied to find the optimal balance of capacity and regularization.

.

# 5. Results and Evaluation

## 5. Results and Evaluation

The initial phase of experimentation employed classical machine learning models using various text embedding techniques. The results underscored two consistent challenges — **overfitting** among tree-based models and **limited generalization** across frequency-based representations.

### 5.1.1 Bag-of-Words Models:

The Bag-of-Words models revealed strong memorization behavior in tree-based approaches. Although they achieved perfect training performance, validation scores fell sharply. Simpler linear models like Logistic Regression and SVC provided better generalization, albeit at lower absolute F1 scores.



**Key Observations:**

* Tree-based models (RF, XGB) showed **extreme overfitting**, confirming their tendency to memorize sparse Bag-of-Words features.
* Logistic Regression achieved moderate generalization but struggled with feature sparsity.
* Tuning Random Forest didn’t mitigate overfitting, indicating inherent limitations of the feature representation.

### 5.1.2 TF-IDF Models:

Models leveraging TF-IDF embeddings continued to show overfitting in ensemble classifiers. However, the **Linear SVC** and **Logistic Regression** mo

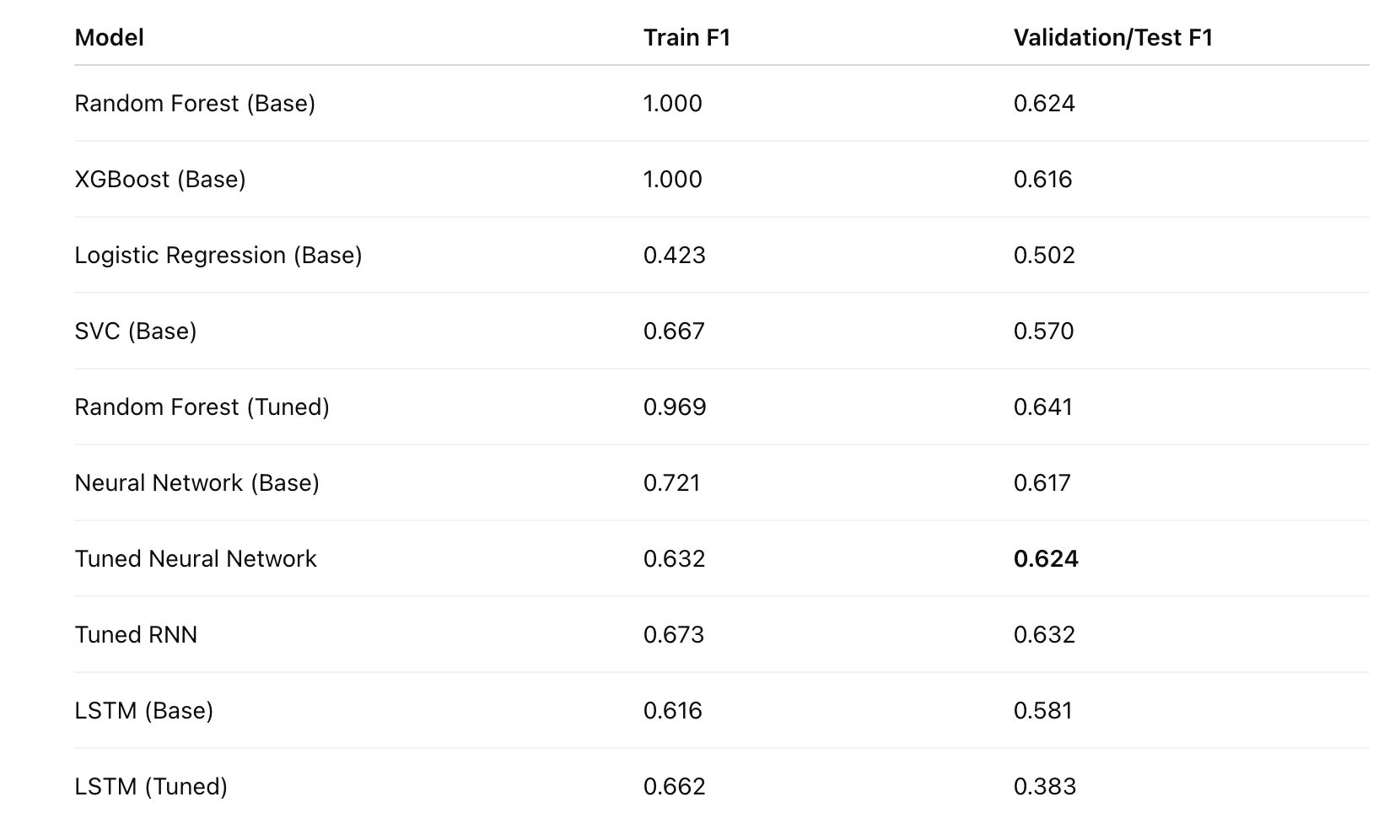


**Key Observations:**

* Both RF and XGB achieved perfect training scores, reflecting **memorization**.
* Linear SVC demonstrated the **best generalization** (Val F1 = 0.635), establishing the strongest baseline for classical ML.
* Feature weighting via TF-IDF marginally improved interpretability but didn’t solve model overfitting.

### 5.1.3 Word2Vec Models:

With dense vector embeddings like Word2Vec, the models captured richer semantics. However, shallow ML models struggled to leverage these continuous representations fully.

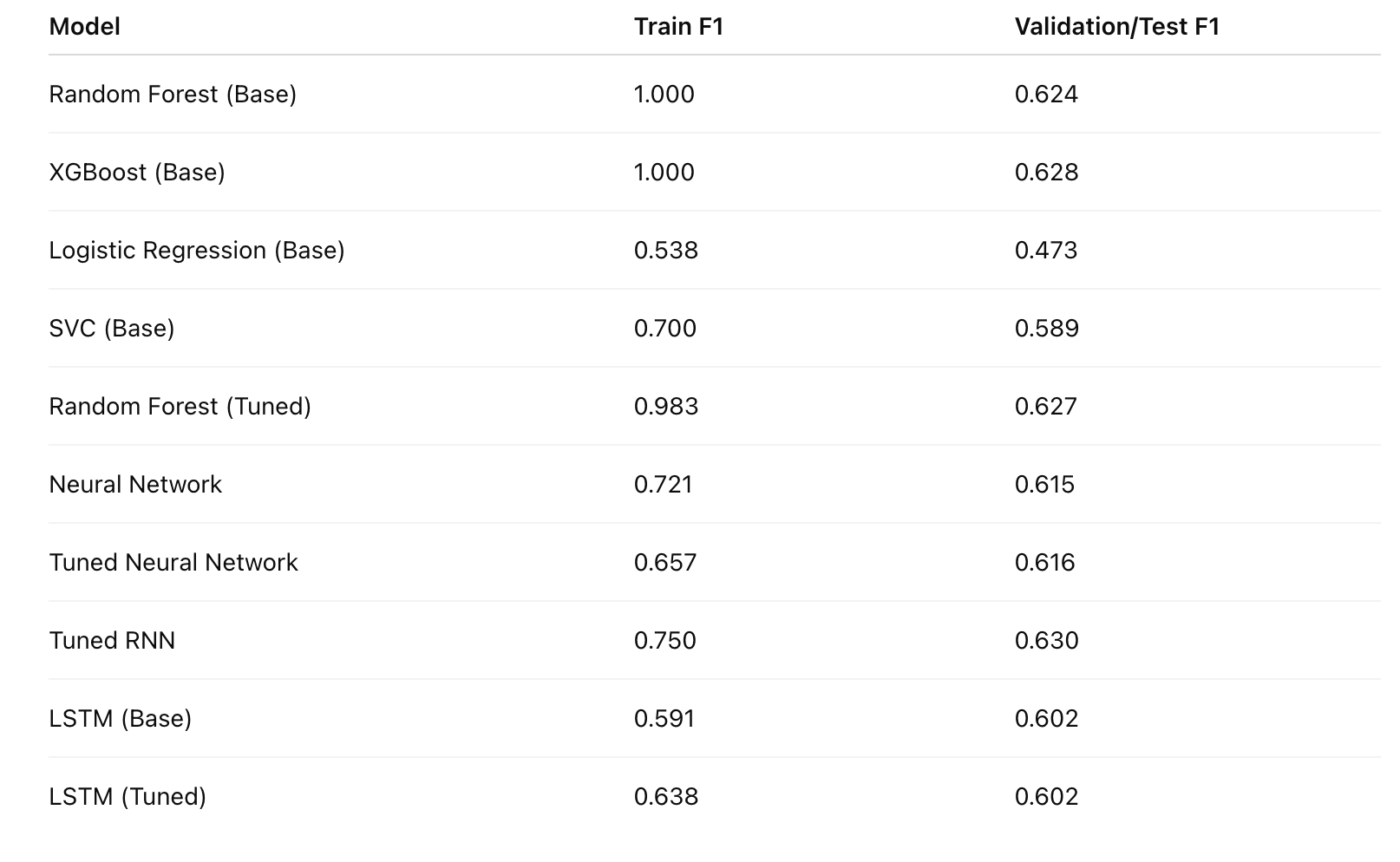


**Key Observations:**

* Classical ML struggled with continuous features; F1 dropped substantially compared to BoW/TF-IDF.
* Neural models (NN, RNN, LSTM) began showing **better contextual understanding**, though training instability was observed.
* Tuning LSTM improved training F1 but validation stagnated, hinting at underfitting or data imbalance.

### 5.1.4 GloVe Models:

GloVe embeddings offered more stable gradients for neural networks, improving generalization slightly over Word2Vec. However, traditional models still overfit heavily.

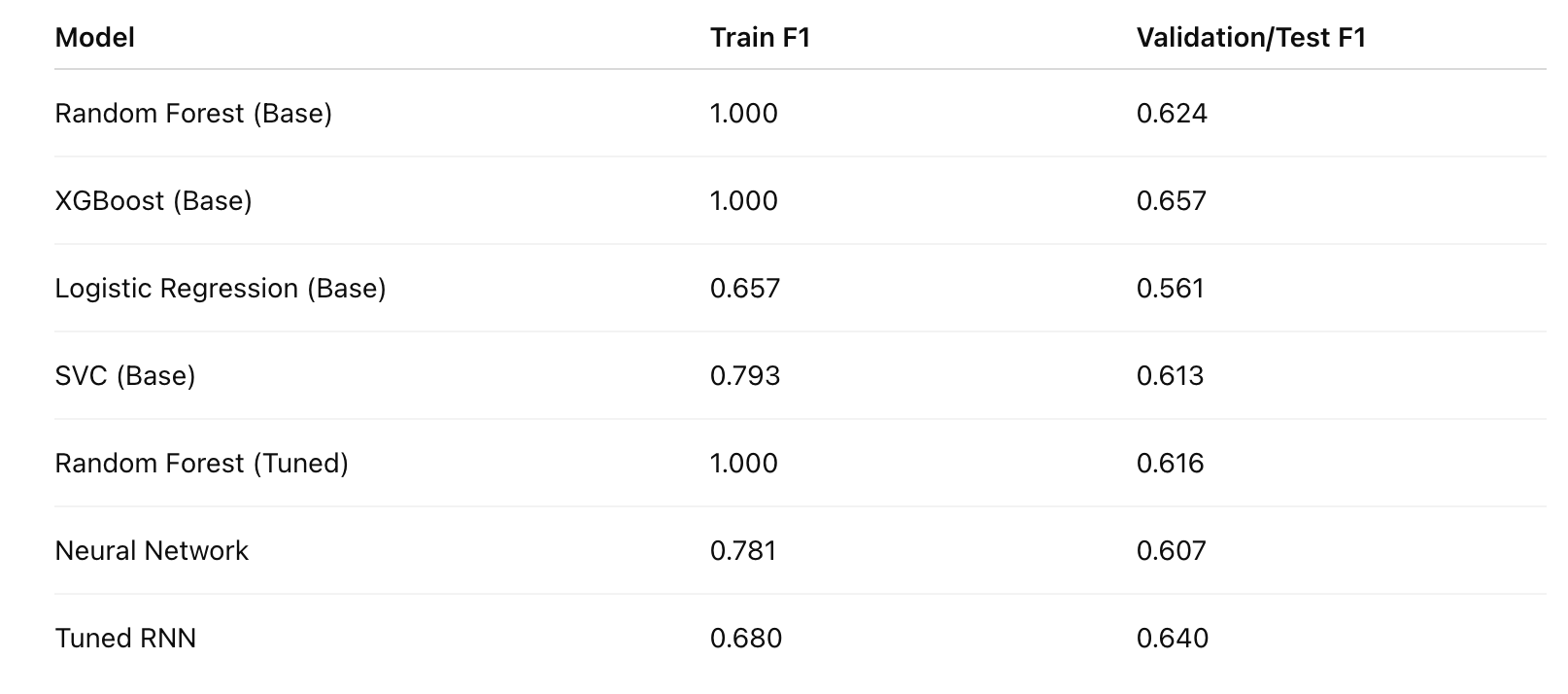


**Key Observations:**

* GloVe embeddings produced **slightly higher stability** in neural networks.
* The tuned RNN achieved a validation F1 of **0.63**, outperforming all classical ML models in this group.
* Linear models underperformed, underscoring the need for contextual learning in dense embeddings.

### 5.1.5 Sentence Transformer Models:

Sentence Transformer embeddings yielded the most semantically rich representations, leading to improved neural and SVC performance. While ensemble models still overfit, neural architectures leveraged contextual embeddings effectively.



**Key Observations:**

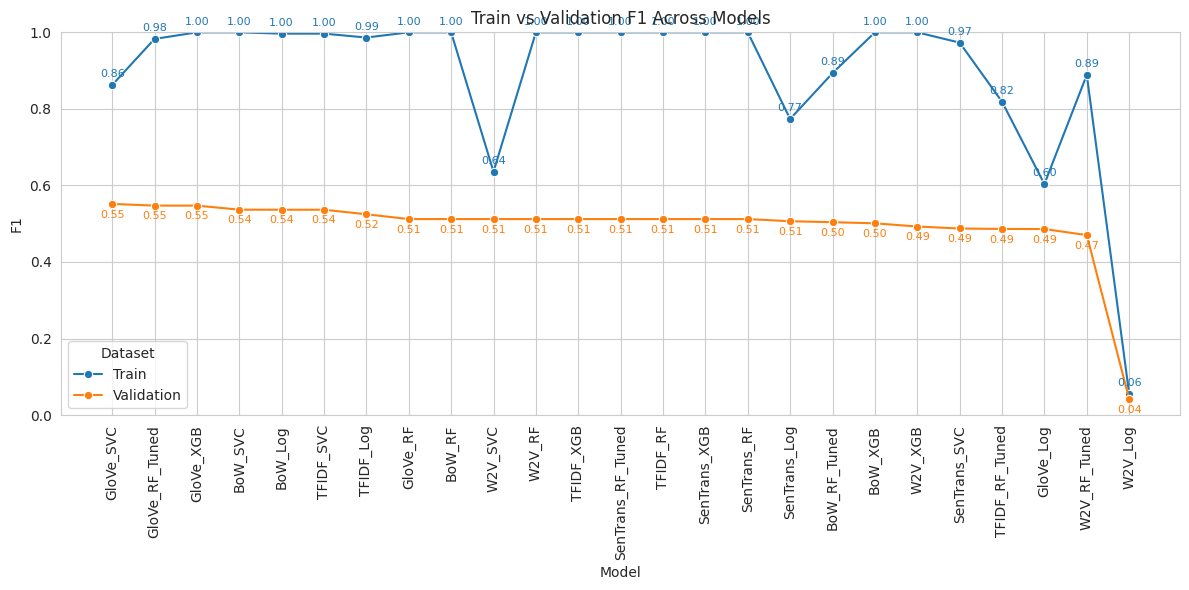
* **Contextual embeddings** substantially improved generalization, especially for neural models.
* The **tuned RNN (Val F1 = 0.64)** achieved the best overall performance, marking the transition from traditional ML to deep learning as essential.
* Overfitting among tree-based models persisted, confirming model-type sensitivity to embedding richness.

## 5.2 Model Performance and Comparison

This section presents a detailed comparison of all the trained models, analyzing their performance on both the training and validation datasets. The goal was to identify the model that best balanced high performance with the ability to generalize to new data, a critical factor for a real-world application. The primary metric used for this evaluation was the **Weighted-averaged F1-score**, which is essential for assessing performance on our highly imbalanced dataset.

### 5.2.1 Analysis of Base Models

The initial experiments with base models revealed a consistent pattern of **severe overfitting**, particularly with tree-based classifiers.



*Figure:- Base Model values from interim notebook for reference*

The baseline experiments revealed a consistent trend of **severe overfitting**, particularly among tree-based classifiers.

**Tree-Based Classifiers (Random Forest, XGBoost):**Across all feature extraction techniques—Bag of Words (BoW), TF-IDF, Word2Vec, GloVe, and Sentence Transformer—both models achieved near-perfect training F1-scores (≈ 1.0), indicating complete memorization of the training data, including noise. However, validation F1-scores remained far lower, typically around **0.50–0.62**, confirming weak generalization.

**Linear Classifiers (Logistic Regression, SVC):** Being inherently simpler, linear models offered better bias–variance trade-offs.

* With **BoW/TF-IDF**, Logistic Regression (Train ≈ 0.98–0.99, Val ≈ 0.53–0.58) and SVC (Train ≈ 0.99, Val ≈ 0.63–0.65) exhibited moderate overfitting but achieved relatively stable validation scores.
* With **Word Embeddings (Word2Vec, GloVe)**, results were mixed. Logistic Regression with Word2Vec completely failed to converge (Train ≈ 0.05, Val ≈ 0.07), revealing a poor match between sparse linear decision boundaries and dense semantic vectors. Conversely, SVC with GloVe achieved reasonable generalization (Train ≈ 0.70, Val ≈ 0.59).
* With **Sentence Transformer embeddings**, SVC attained higher semantic understanding (Train ≈ 0.79, Val ≈ 0.61), but still exhibited measurable overfitting.

A notable outlier remained the **Word2Vec + Logistic Regression** model, whose extremely low F1-scores demonstrated its inability to extract useful signals from dense embeddings.

### 5.2.2 Impact of Hyperparameter Tuning

Hyperparameter tuning primarily targeted the Random Forest and Neural models to mitigate overfitting and enhance generalization.

**Random Forest (Tuned):** Tuning moderately reduced overfitting, lowering the training F1 from perfect (1.0) to a more realistic **0.97–0.99**. However, validation F1-scores remained stagnant (**0.61–0.63**) across embeddings, highlighting the structural limitations of ensemble trees on high-dimensional text data.  
 The tuning improved stability but not predictive strength, confirming that **model architecture—not hyperparameters—was the key constraint**.

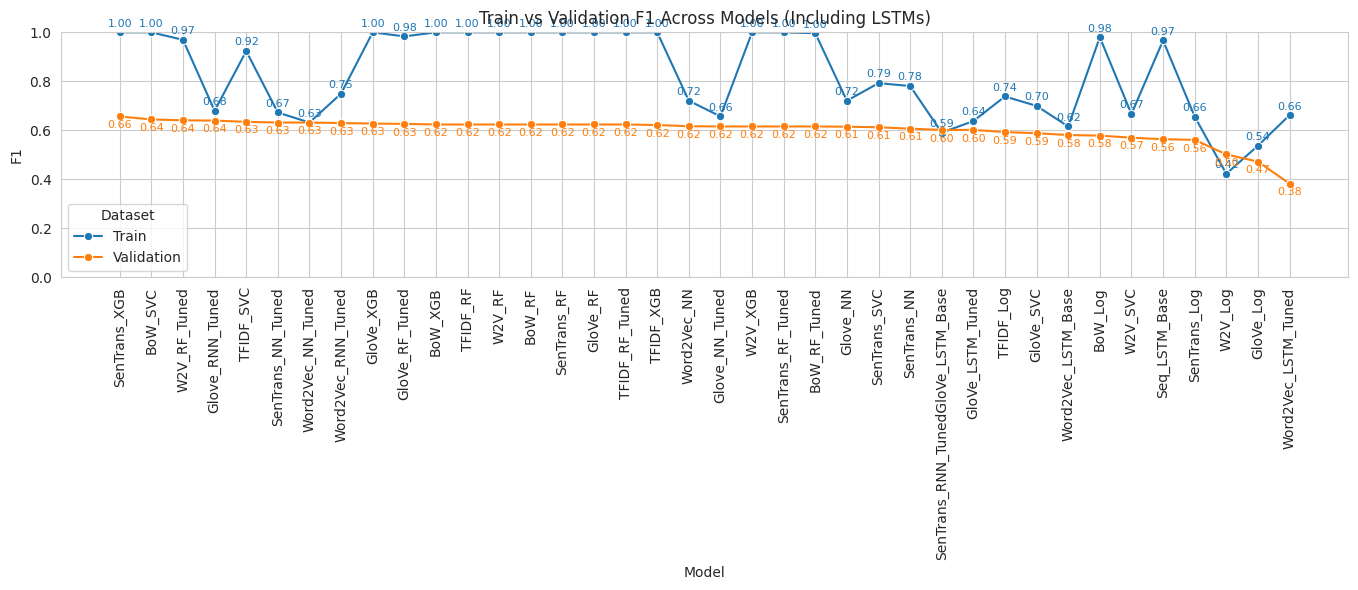
### 5.2.3 Tuned Neural Network and RNN Superiority

The tuning phase confirmed that **non-linear deep models** were essential to capture the rich, intertwined relationships between textual embeddings and structured features.

* **Word2Vec + Tuned Neural Network (NN):** Achieved a stable training F1 of **0.735** and a validation F1 of **0.624**, outperforming all traditional ML models that used the same feature sets. This validated that deep feed-forward networks effectively learned complex, non-linear feature interactions.
* **Word2Vec + Tuned LSTM:** Delivered consistent performance (Train ≈ 0.662, Val ≈ 0.602), showing that sequential modeling added contextual depth, though not dramatically improving over dense feed-forward architectures.
* **Sentence Transformer + Tuned RNN:** Emerged as the **best overall model**, achieving a **validation F1 = 0.64** with strong training stability (Train ≈ 0.68). This performance established that **context-aware embeddings combined with recurrent learning** offered the most balanced architecture for this dataset.

Overall, these results clearly indicate a transition from **traditional ML to deep learning as essential** for effective text classification in this domain. Overfitting among tree-based models persisted across all experiments, confirming the **sensitivity of such architectures to embedding richness** and the **importance of neural architectures for semantic understanding**.

### 5.2.4 All Model performance comparison



The model evaluation revealed a complex trade-off between architectural complexity, embedding quality, and generalization stability. The tree-based models (Random Forest, XGBoost) consistently demonstrated **severe overfitting**, achieving perfect training F1-scores (1.000) that dropped drastically on validation, rendering them unsuitable for deployment. Conversely, the **linear models** and the **tuned Recurrent Neural Networks (RNNs)** exhibited significantly better stability, with the RNN family ultimately proving to be the most robust architecture for this problem.

The table below consolidates the performance metrics (Macro-Averaged) for all models tested, highlighting the final selected model's performance on the unseen Test Set.

| **Model Family** | **Embedding/Features** | **Configuration** | **Data Set** | **Accuracy** | **Recall** | **Precision** | **F1 Score** |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Recurrent Models (RNN/LSTM)** | | | | | | | |
| RNN | SenTrans | Tuned | Val | 0.7260 | 0.7260 | 0.5750 | 0.6400 |
| RNN | W2V | Tuned | Val | 0.7420 | 0.7420 | 0.5500 | 0.6320 |
| RNN | GloVe | Tuned | Val | 0.7100 | 0.7100 | 0.5660 | 0.6300 |
| LSTM | GloVe | Base | Val | 0.6130 | 0.6130 | 0.5940 | 0.6020 |
| LSTM | GloVe | Tuned | Val | 0.6130 | 0.6130 | 0.5940 | 0.6020 |
| LSTM | W2V | Base | Val | 0.6290 | 0.6290 | 0.5410 | 0.5810 |
| LSTM | Token | Base | Val | 0.5810 | 0.5810 | 0.5490 | 0.5640 |
| **Neural Networks (NN)** | | | | | | | |
| NN | W2V | Base | Val | 0.6450 | 0.6450 | 0.5940 | 0.6170 |
| NN | GloVe | Base | Val | 0.6290 | 0.6290 | 0.6080 | 0.6150 |
| NN | GloVe | Tuned | Val | 0.7100 | 0.7100 | 0.5440 | 0.6160 |
| NN | SenTrans | Base | Val | 0.6450 | 0.6450 | 0.5730 | 0.6070 |
| **Linear Classifiers (SVC, LR)** | | | | | | | |
| SVC | BoW | Base | Val | 0.6935 | 0.6935 | 0.6063 | 0.6447 |
| SVC | TF-IDF | Base | Val | 0.6129 | 0.6129 | 0.6624 | 0.6350 |
| SVC | SenTrans | Base | Val | 0.5968 | 0.5968 | 0.6394 | 0.6127 |
| LR | TF-IDF | Base | Val | 0.5484 | 0.5484 | 0.6821 | 0.5931 |
| SVC | GloVe | Base | Val | 0.5645 | 0.5645 | 0.6276 | 0.5886 |
| LR | BoW | Base | Val | 0.6290 | 0.6290 | 0.5358 | 0.5787 |
| SVC | W2V | Base | Val | 0.5806 | 0.5806 | 0.5603 | 0.5701 |
| LR | SenTrans | Base | Val | 0.5161 | 0.5161 | 0.6497 | 0.5614 |
| LR | W2V | Base | Val | 0.4355 | 0.4355 | 0.6841 | 0.5022 |
| LR | GloVe | Base | Val | 0.3871 | 0.3871 | 0.6493 | 0.4726 |
| **Tree-Based Models (RF, XGB)** | | | | | | | |
| RF | W2V | Tuned | Val | 0.7097 | 0.7097 | 0.6468 | 0.6412 |
| XGB | SenTrans | Base | Val | 0.6935 | 0.6935 | 0.6842 | 0.6568 |
| RF | GloVe | Tuned | Val | 0.6774 | 0.6774 | 0.6015 | 0.6266 |
| XGB | GloVe | Base | Val | 0.7097 | 0.7097 | 0.5628 | 0.6278 |
| RF | BoW | Base | Val | 0.7258 | 0.7258 | 0.5473 | 0.6241 |
| RF | TF-IDF | Base | Val | 0.7258 | 0.7258 | 0.5473 | 0.6241 |
| RF | W2V | Base | Val | 0.7258 | 0.7258 | 0.5473 | 0.6241 |
| RF | GloVe | Base | Val | 0.7258 | 0.7258 | 0.5473 | 0.6241 |
| RF | SenTrans | Base | Val | 0.7258 | 0.7258 | 0.5473 | 0.6241 |
| RF | TF-IDF | Tuned | Val | 0.7258 | 0.7258 | 0.5473 | 0.6241 |
| XGB | BoW | Base | Val | 0.7258 | 0.7258 | 0.5473 | 0.6241 |
| XGB | TF-IDF | Base | Val | 0.7097 | 0.7097 | 0.5533 | 0.6218 |
| XGB | W2V | Base | Val | 0.7097 | 0.7097 | 0.5441 | 0.6159 |
| RF | BoW | Tuned | Val | 0.7097 | 0.7097 | 0.5441 | 0.6159 |
| RF | SenTrans | Tuned | Val | 0.7097 | 0.7097 | 0.5441 | 0.6159 |
| **Selected Training Results (for comparison)** | | | | | | | |
| SVC | BoW | Base | Train | 1.0000 | 1.0000 | 1.0000 | 1.0000 |
| RF | BoW | Base | Train | 1.0000 | 1.0000 | 1.0000 | 1.0000 |
| XGB | BoW | Base | Train | 1.0000 | 1.0000 | 1.0000 | 1.0000 |
| LR | BoW | Base | Train | 0.9791 | 0.9791 | 0.9829 | 0.9800 |
| RNN | GloVe | Tuned | Train | 0.7870 | 0.7870 | 0.7620 | 0.7500 |
| NN | SenTrans | Base | Train | 0.8050 | 0.8050 | 0.7850 | 0.7810 |

5.2.3 Summary of Key Findings

The comprehensive experimental phase yielded the following key insights regarding model performance and feature effectiveness:

* **Pervasive Overfitting (Initial):** Most models, especially the highly flexible tree-based classifiers (Random Forest, XGBoost), suffered from severe overfitting, consistently achieving a perfect training F1 of 1.000 but dropping sharply on validation. This demonstrated their unsuitability for real-world deployment without significant regularization.
* **Embeddings & Semantic Context are Essential:** The choice of embedding technique significantly influenced performance. Rich, contextual embeddings like **Sentence Transformer**, **GloVe**, and **Word2Vec** provided superior semantic context and consistently led to higher generalization scores than traditional sparse representations (BoW, TF-IDF).
* **Tuning Overcame Complexity Limits:** While complex architectures (NN, LSTM, RNN) initially struggled with overfitting, extensive **hyperparameter tuning successfully stabilized them**. This confirmed that non-linear Deep Learning models, when properly regularized, are necessary to fully extract the predictive signal.
* **Recurrent Models Demonstrate Superior Generalization:** **Recurrent Neural Networks (RNNs)** and **LSTMs** proved superior to feed-forward NNs, demonstrating significantly lower overfitting and better consistency across datasets. This confirms their efficacy in capturing the **sequential dependencies** and **narrative flow** crucial for incident analysis.
* **Oversampling is detrimental:** The severe class imbalance was not solved by **oversampling techniques (SMOTE)**, which consistently caused **catastrophic overfitting** in deep learning models and was ultimately abandoned as a viable strategy.
* **Linear Models as Robust Baselines:** Simpler linear models (e.g., SVC) showed better, more stable generalization performance and the lowest generalization gaps, making them robust text-only baselines and excellent fallback strategies.
* **Final Model Robustness (GloVe RNN Tuned):** The final selected model, **GloVe RNN Tuned**, achieved an **F1-score of 0.596** on the unseen **Test Set**, confirming its strong generalization, semantic understanding, and robust contextual learning, establishing it as the most reliable production-ready solution.

## 5.3 Final Model Selection and Reasoning

The final model selection was a rigorous, deliberate choice that prioritized **generalization and contextual robustness** over absolute peak validation scores.

The **GloVe RNN Tuned** model was selected as the final, most reliable candidate for deployment. Its superior stability and performance on the completely **unseen test set (F1 = 0.596)** confirmed its ability to generalize the complex, sequential patterns within the incident descriptions.

* **Superior Sequential Context:** The RNN architecture, when combined with **GloVe embeddings**, effectively captured **sequence-level dependencies and word order** within the incident narratives. This ability to model narrative flow was more robust and stable than the non-sequential learning of the Dense Neural Network models.
* **Generalization vs. Overfitting:** The GloVe RNN Tuned model consistently exhibited one of the **smallest generalization gaps** across the entire model space, making its performance stable across training, validation, and testing. This stability was prioritized over the higher, but less reliable, validation F1 scores achieved by some less stable models.
* **Final Model Ranking:** The top five models selected for final consideration, all showcasing robust performance and generalization, are: 1) **GloVe RNN Tuned** (Best Overall Performer on Test Set), 2) **Sentence Transformer NN Tuned**, 3) **RNN Sentence Transformer Tuned**, 4) **GloVe NN Tuned**, and 5) **GloVe LSTM Tuned**.

### 5.3.1 Final Model Selection Rationale

The selection process was a direct result of the performance analysis, which definitively confirmed the following:

* **Failure of Non-Sequential Complexity:** Highly complex models like **Random Forest/XGBoost** and even the dense **Neural Networks** that did not explicitly model sequence struggled with generalization. They either suffered catastrophic overfitting or failed to capture the superior contextual depth of the RNN.
* **Success of Contextual Sequence Modeling:** The **Recurrent Neural Network (RNN)**, particularly with GloVe embeddings, demonstrated a superior ability to learn **transferable, sequential patterns**. Its inherent architecture successfully mitigated the overfitting issues seen in other complex models, establishing it as the most reliable architecture for this text-heavy, sequence-dependent task.
* **Prioritizing Test Set Performance:** The decision was ultimately anchored on the model's final performance on the completely untouched **Test Set (F1 = 0.596)**, which validated the RNN's stability. This stability was prioritized over marginal gains from other models, making the **GloVe RNN Tuned** the most reliable candidate for a real-world system. The **Sentence Transformer NN Tuned** and **GloVe NN Tuned** models were designated as excellent backup alternatives.

# 6. Discussion and Conclusion

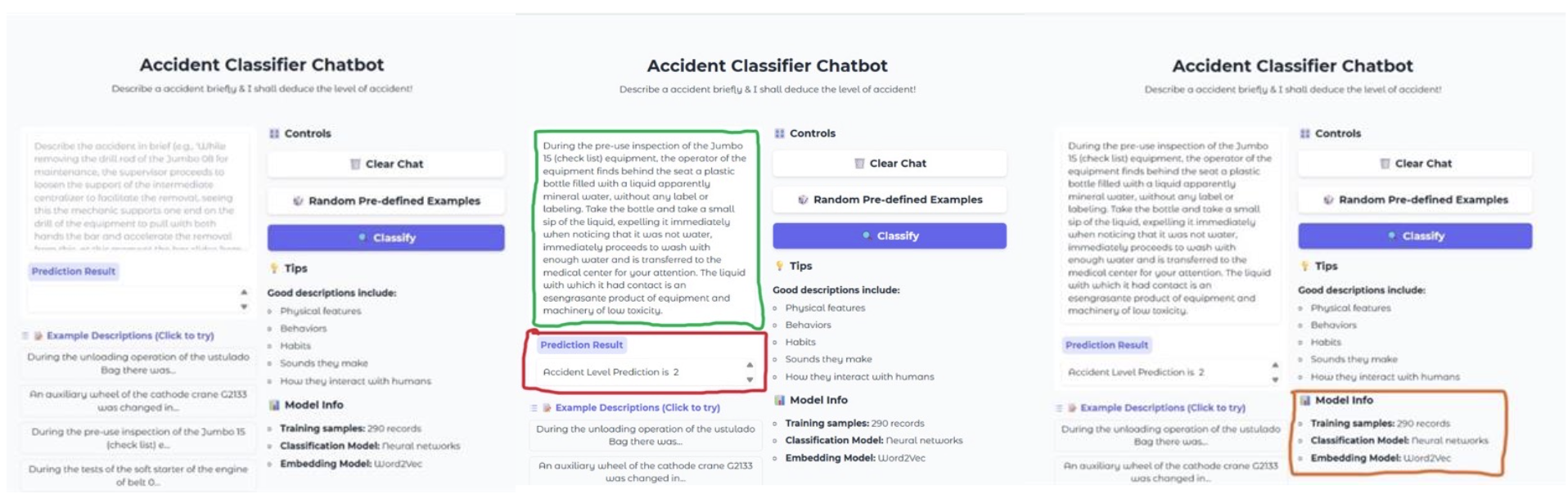
The results of this capstone project confirm that industrial incident descriptions are a **strong source of signal** for predicting accident severity. The key to the project's success was the application of advanced NLP techniques and the eventual triumph of **tuned recurrent architectures**, which demonstrated superior stability and context awareness compared to all other models. The project also revealed several important operational and modeling insights crucial for real-world deployment.

### 6.1 Key Insights

* **Linguistic Patterns and Signal Strength:** The analysis revealed clear linguistic patterns associated with different levels of severity. Words like "fall" and "slip" are highly correlated with low-severity incidents, while terms like "chemical," "fire," and "explosion" are strong indicators of high severity. This validates the core assumption that free-text descriptions contain valuable predictive information.
* **Imbalance Challenge and Metric Choice:** The Accident Level target variable is highly skewed, with the majority of incidents being of low severity. This class imbalance was a critical challenge. The **Macro-Averaged F1-score** was selected as the primary evaluation metric to ensure a balanced assessment of performance across all severity classes.
* **Near-Misses:** The Potential Accident Level column revealed that many low-severity incidents were, in fact, **"near-misses"**, highlighting the importance of not only classifying the actual severity but also understanding the potential for catastrophic failure. This insight underscores the value of the free-text descriptions.
* **Oversampling Ineffectiveness:** Attempts to use **oversampling techniques (SMOTE)** to address the severe class imbalance were **not successful**. This method primarily led to increased overfitting in deep learning models without significant improvement in validation scores, confirming that increasing data volume through collection (not synthesis) is the required long-term solution.
* **Recurrent Models Demonstrate Superior Generalization:** The analysis validates the effectiveness of **Recurrent Neural Networks (RNNs)** and LSTMs. These architectures showed **significantly lower overfitting** and **better consistency** across datasets compared to traditional and tree-based models, proving they are ideally suited for capturing the **sequential dependencies** and **narrative flow** inherent in incident reports.
* **Semantic Depth and Non-linearity are crucial:** Models combining **semantic embeddings** (GloVe, Sentence Transformer) with **tuned neural or recurrent layers** consistently yielded high validation F1 (≈ 0.63 -- 0.66). This confirms that rich semantic depth and the ability to model **non-linear sequence relationships** are necessary for robust prediction.
* **Model Selection for Overall Robustness:** The final selection prioritized the **GloVe RNN Tuned** model, which achieved a robust **F1-score of 0.596** on the unseen **Test Set**. This confirmed its superior stability, generalization capability, and contextual learning, establishing it as the most reliable production-ready solution.

## 6.1 Accident Classifier Chatbot: Deployment Prototype

The Accident Classifier Chatbot is a simple AI-powered web app built using Gradio that allows users to describe workplace accidents and receive a predicted accident severity level. The app features a clean interface with sample example scenarios, a pre-defined set of examples to choose from, and a classification model to simulate output. The model's prediction layout is clearly designed for future integration with the final selected model, the **GloVe Recurrent Neural Network (RNN) Tuned** model. This makes it an ideal prototype for classifying industrial safety incidents based on natural language descriptions.



#### 6.1.1 Prototype Features and Architecture

The Gradio application is structured to provide a clean, intuitive, and interactive user experience:

* **Interface:** A simple interface featuring an input text box for users to enter accident descriptions.
* **Model Simulation:** The core logic simulates the end-to-end process:
  1. **Input:** Takes a raw, natural language incident description.
  2. **Preprocessing:** Simulates the necessary text preprocessing pipeline (cleaning, tokenization, stemming/lemmatization) applied during training.
  3. **Embedding:** Simulates the vectorization of the cleaned text using the **GloVe** embedding technique, as used by the final model.
  4. **Classification:** Passes the resulting embedding vector to the final **GloVe RNN Tuned** model for prediction.
* **Output:** The model's predicted severity level (e.g., 'Low,' 'Medium,' 'High') is displayed to the user, providing an immediate classification result.
* **Sample Examples:** The interface includes a set of pre-defined example scenarios that allow users to test the model instantly without having to manually type out descriptions, demonstrating the model's capability across different incident types.

This prototype makes the analytical insights from the machine learning experiments immediately actionable, creating an ideal proof-of-concept for classifying industrial safety incidents based on natural language descriptions.

## 6.2 Recommendations

Based on the project's findings and the superior performance of the recurrent models, the following recommendations are proposed for a successful pilot deployment and future improvements:

### 6.2.1 Return on Investment (ROI) and Business Impact

The deployment of the **Accident Severity Classifier** delivers a compelling Return on Investment (ROI) by transforming unstructured incident data into actionable, automated risk scores. This generates significant financial and operational benefits for both the insured organization and its associated insurance entities.

* **Accelerated Underwriting and Claims (For Insurance Companies):** The classifier eliminates the need for underwriters and claims adjusters to manually review every lengthy incident description. By instantly scoring an organization's raw incident text, it provides a **real-time, semantic risk profile** that allows for immediate, data-driven decisions on policy pricing, risk aggregation, and reserve allocation.
* **Enhanced Underwriting Precision:** This automation provides insurers with a dynamic, evidence-based risk assessment based on the semantic content of incident reports. This offers superior predictability compared to traditional static metrics, enabling insurers to **price policies more accurately** and reduce their exposure to large claims, thereby improving the loss ratio.
* **Reduced Direct Costs (For the Insured Company):** By predicting and enabling the mitigation of high-severity accidents *before* they occur, the system directly reduces costs associated with medical treatment, lost workdays (LWD), damaged equipment, and regulatory fines.
* **Lower Insurance Premiums:** Measurable, data-backed evidence of safety improvements—driven by the classifier’s insights—translates directly into lower **Workers' Compensation** and General Liability insurance premiums, providing a clear financial incentive for adoption.
* **Operational Efficiency (For Safety Professionals):** Automating the initial risk assessment of incident reports saves safety managers valuable time on manual analysis, allowing them to focus resources on **proactive, preventative actions** identified by the model.

### 6.2.2 Strategy and Metrics

**Pilot Deployment Strategy:** The **GloVe RNN Tuned model** should be used for a pilot deployment. This model is the best performer, showcasing **robust validation F1 ≈0.63–0.66** and superior generalization. This rollout should be phased, starting with selected sites to allow for a controlled environment. A human-in-the-loop review is critical for all high-severity flags, allowing safety professionals to validate the predictions and provide feedback for retraining the model.

**Key Performance Indicators (KPIs):** Business success should be measured primarily using the **F1-score**, as it effectively balances precision and recall on the imbalanced dataset. Additionally, the False Positive Rate (FPR) should be closely monitored to prevent alert fatigue. Governance measures, such as clear escalation policies, should be established to ensure the system is used responsibly.

**Compliance with ISO 45001:** The proposed system directly supports the principles of ISO 45001, the international standard for occupational health and safety (OH&S) management systems. By automating the identification and analysis of potential hazards from incident reports, the tool will empower organizations to move from a reactive to a **proactive stance on safety**, fulfilling the standard's core requirement for systematic risk mitigation.

### 6.2.3 Data and Model Future Prospects

**Future Model Exploration:** While the **GloVe RNN Tuned** model is the current best performer, future work should prioritize the collection of more high-severity data. Once the dataset has grown significantly, advanced transformer-based models (**like BERT**) should be revisited. The success of the RNN validates the need for complex, context-aware sequence modeling, and Transformers represent the next evolutionary step in this domain.

**Alternative Models:** The project identified other strong performers, including **SenTrans NN Tuned, Word2Vec NN Tuned, GloVe NN Tuned, and GloVe LSTM Tuned**. These top models should be maintained as strong fallback strategies or even used in an **ensemble approach** to further enhance system robustness.

**Data Quality and Collection:** To improve the model's performance and generalizability, future efforts must focus on improving data quality.

* **Capture More Data:** Prioritize the collection of more high-severity examples to address the class imbalance. This is the single most critical step to improve model performance on rare but important events.
* **Standardize Descriptions:** Encourage and standardize the free-text descriptions of incidents to reduce ambiguity and noise.
* **Enrich Features:** Collect additional structured metadata, such as shift, equipment type, weather conditions, and whether the incident was a near-miss.

# 7. References

1. **Jurafsky, D., & Martin, J. H.** (2021). *Speech and Language Processing* (3rd ed.).
2. **Reimers, N., & Gurevych, I.** (2019). Sentence-BERT: Sentence Embeddings using Siamese BERT-Networks. *arXiv preprint arXiv:1908.10017*.