MACHINE LEARNING PROJECT

Master's in Data Science and Advanced Analytics

NOVA Information Management School

Universidade Nova de Lisboa

WCB Compensation Model Report

Group 44

Brandon Seichal, 20240592

Bruna Simões, 20240491

Guilherme Cordeiro, 20240527

Maria Cardoso, 20240493

Rafael Silva, 20240511

Fall Semester 2024-2025

ABSTRACT

The present project's purpose is to develop machine learning models to enhance the efficiency, fairness, and accuracy of the New York Workers' Compensation Board (WCB)'s evaluation process for workplace injury claims. Workers' compensation systems are essential for providing employees with financial and medical support, yet their decision-making processes are often labor-intensive and prone to inconsistencies. The proposed models aim to streamline claim evaluations by providing a data-driven framework for decision-making, ultimately improving outcomes for both employees and the WCB.

We worked with a dataset of nearly 600,000 claims with 33 features related to demographics, injuries, and timelines. Data preprocessing focused on addressing issues like missing values, inconsistent formats, and outliers. New features were created to enrich the dataset such as time intervals and missing value indicators. Encoding techniques and scaling transformations were applied on the features to ensure compatibility with used algorithms. Claim Injury Type (claim classification) was defined as the primary target variable, but the likelihood of claim resolution without WCB intervention was also briefly examined as a secondary target.

Feature selection employed statistical and embedded methods, with majority voting to retain robust predictors while minimizing redundancy. Modeling leveraged a range of algorithms, including decision trees, logistic regressions, and ensemble methods, with hyperparameter tuning and resampling strategies like Tomek Links to address class imbalance. The models were evaluated using cross-validation and compared by their macro F1 Score. Additionally, custom ensemble strategies, Weighted One-vs-Rest and Stacking, were developed.

Results highlight the effectiveness of ensemble methods in multiclass prediction, though challenges remain in accurately classifying minority classes due to severe class imbalance. Future research could focus on advanced resampling methods, additional data preprocessing techniques, and increased computational resources to optimize model performance. Despite current limitations, the developed models demonstrate potential for supporting WCB decision-making, reducing reliance on subjective judgment, and improving fairness in workers' compensation systems.

TABLE OF CONTENTS

1. Introduction

As consultants, we were tasked with analyzing employee data collected with the objective of developing a model that can automate the decision-making process for workplace injury claims for the New York Workers' Compensation Board (WCB).

Workers' compensation systems are essential in the protection of employees, since they provide financial support and benefits in the event of workplace injuries (Cloeren 2016). Despite this, the process of evaluation and approval of claims can be very complex, meticulous and time-consuming. To address these challenges, machine learning techniques can offer a strong foundation for building a data-driven policy-making system, delivering greater simplicity, and consistency in decision-making, since they present a superior predictive performance when compared to traditional statistical models, and can capture non-linear relationships and handle diverse types of data (Fernandes 2022).

Recent academic research has shown the transformative potential of machine learning when it comes to enhancing the decision-making process within workers' compensation systems. For instance, (Meyers 2018) utilized a machine learning algorithm to code over 1.2 million workers' compensation claims, effectively identifying high-risk industries and injury causes, which targets correct interventions. (Zanke 2022) has also explored the application of artificial intelligence and machine learning in workers' compensation risk management, highlighting their efficacy in predicting workplace injuries, facilitating return-to-work programs, and detecting fraud, which collectively contribute to improved safety outcomes and cost savings.

This project demonstrates comprehensive data preprocessing, data transformation, and the application of various algorithms and models with the purpose of answering the following question: How can machine learning models be effectively utilized to predict workplace injury claim outcomes, ensuring fairness, accuracy, and efficiency in the decision-making process for the New York Workers' Compensation Board? We aim to develop models that can learn accurately from the historical data we have access to, to make a precise prediction, and to start building a solution that enhances fairness and efficiency in workers' compensation systems and makes them less reliable on human judgement, inherently biased.

2. Data Exploration and Preprocessing

2.1 Exploratory Data Analysis

For this research, we have available two data sets: one for training our models and another for testing. Our training dataset consists of nearly 600,000 claims with 33 features that can be divided into three categories: demographic, injury-related, and time-based [Table 4 – Feature dimensions mapping].In the first phase of our project, we explored the data to check for potential issues and incoherence like undesirable characters, missing values and outliers, while also reviewing data types and overall consistency.

2.1.1 Data Coherence

First, we reviewed data types to ensure they were correctly formatted for meaningful analysis. Numeric codes, such as those for injury claims and zip codes, were converted to objects to avoid them being treated as numerical features. Date-related features were stored as strings, posing potential issues, so we converted them to date formats. Any undesirable characters, such as "UK" in *Medical Fee Region* (5.83%), "UNKNOWN" in *Carrier Type* (0.30%), and "U" and "X" in *Gender* (0.82% and 0.01%, respectively), were treated as missing values.

During preliminary data exploration, we identified several data integrity issues. Notably, 19,445 observations were missing all relevant data except for the *Assembly Date* and *Claim Injury Type*. As these records provided no meaningful information, we removed them from the dataset. Additionally, *OIICS Nature of Injury*

Description was entirely empty, and the *WCB Decision* field contained only a single unique value, “Not work related”. Although the last one could potentially serve as a target variable, it fell outside the scope of our analysis. As a result, we excluded these features.

Moreover, we observed instances where the *Accident Date* was after other dates, like *Assembly Date* and *C-2 Date*, which is inconsistent since the date of the accident should always come first. These observations were also removed from the dataset.

We noticed that a small number of injuries labeled as “COVID-19” in *WCIO Nature of Injury Description* were not flagged in *COVID-19 Indicator* (<0.05%). To keep the data consistent, we updated these cases by setting *COVID-19 Indicator* to “Y”. Additionally, some Zip Code values were inconsistent, including entries with fewer than five characters or containing strings (3.25%). Zip Codes with fewer than three characters and string values were marked as missing, while four-digit Zip Codes were adjusted by multiplying them by ten to meet the standard format. In addition, we identified cases where a single code was linked to multiple descriptions across all features where both were available. To maintain consistency, we prioritized the descriptions as the most reliable source of information.

Finally, *Claim Identifier* was set as the index, serving as a unique identifier for each claim.

2.1.2 Setting Target Variables

The goal of this project is to predict WCB's deliberations on benefit awards (***Claim Injury Type***) and the likelihood of resolving claims without the Board’s direct intervention (***Agreement reached***). Our analysis revealed **highly imbalanced data** on the classification of *Claim Injury type*, with the majority being “2. NON-COMP” (representing 50.83% of the data set) and the least frequent classes “7. PTD” (0.01%) and “8. DEATH” (0.08%), and on the classification of *Agreement Reached*, where 0 (no agreement without the Board’s direct intervention) accounted for 95.33% of the cases.

2.1.3 Numerical Variables

Our preliminary analysis of numerical features used basic statistics—mean, median, mode, maximum, and minimum – and the examination of their distributions [Figure 1]. This helped identify the outliers and anomalies, guiding our approach for imputing missing values and treating outliers. We were able to identify these anomalies:

- *Average Weekly Wage* showed that more than half of the observations (58.54%) were recorded as 0. Since these workers were in industries not typically volunteer-related and represented nearly half of the dataset, it is unlikely all were unpaid. Given this unreliability, we treated zero values as missing. Additionally, this variable is highly skewed, with a standard deviation of $6,100 and numerous outliers, which will need to be addressed.
- *Age at Injury* and *Birth Year* revealed ages ranging from 0 to 117 which are not logically valid for this analysis. The minimum legal working age in the U.S. is generally 16 years old (Fair Labor Standards Act (FLSA) – Child Labor Provisions n.d.). Additionally, it is highly uncommon for individuals over the age of 80 to be working, as the full retirement age for Social Security benefits ranges between 66 and 67 years for most workers in the U.S. (Retirement Benefits 2024).
- *IME-4 Count* showed some outliers and 77% missing values. Once this variable represents the number of medical forms received, missing data likely means no forms were submitted.

2.1.4 Categorical Variables

Focusing on categorical features, histograms [Figure 2 and Figure 3] show that the most frequently reported WCB district office is New York, as expected. Additionally, 20% of claims come from workers in the healthcare and social assistance sector.

We also found a strong connection between *WCIO Cause of Injury Description* and *WCIO Part of Body Description*. Many claims list causes such as lifting, contact with fellow workers, or strain/injury caused by physical force, commonly linked to affected areas like the lower back, knees, and shoulders as the most frequently. These observations align with *Claim Injury Type* distribution, where “Non-Comp”, “Temporary”, and “Med Only” appear as the most frequent categories.

In examining *Carrier Type*, originally defined on a scale from 1 to 5, it was observed that category 5 comprised three distinct values with a low number of observations. To ensure robustness, these sparse subcategories were consolidated into one. Furthermore, both *Carrier Name* and *Carrier Type* refer to the primary insurance provider but differ in granularity. *Carrier Name* provides more specific information, with 2,046 unique values, while *Carrier Type* groups divide into 5 categories.

2.1.5 Date Variables

The remaining set of variables were analyzed with different visualizations to better understand and identify any relationships they may have [Figure 4].We observed that the integrity of the data within *Accident Date* and *C-2 Date* was less likely to be compromised due to the lack of data that was missing, both 0.64% and 2.54% respectively, versus *C-3 Date* and *First Hearing Date* which had 67.38% and 73.73% of their total data missing.

For cases where claims had no designated date for First Hearing, the assumption was made that the claim never reached a stage where a hearing would be necessary. This assumption aligns with how the data is represented, as the dataset includes only resolved claims. Therefore, even if a *First Hearing Date* is missing, the cases were still finalized, indicating that they did not reach this procedural step because it was unnecessary.

Additionally, nearly half of the claims are initiated by the employer submitting a claim form (*C-2 Date*), compared to the employee submitting a claim form (*C-3 Date*). This pattern indicates that the claim assembly process is typically triggered by either party, and the key information lies in identifying the date of the initial report submission.

2.1.6 Multivariate Relationships

By examining relationships between features [Figure 5 and Figure 6], we uncovered the following insights:

- Districts like NYC report higher injury claim frequencies in counties such as Queens and Kings. Including both *District Name* and *County of Injury* could improve model performance, as regional patterns appear strongly tied to claim rates [Figure 7].
- *Medical Fee Region* IV, the most frequent region, correlates with specific counties [Figure 8]. However, Claim Injury Type shows limited variation across regions, possibly indicating low predictive value [Figure 9].
- *Birth Year* and *Age at Injury*, *Carrier Name* and *Carrier Type*, and all location features, showed strong correlations, which were expected. *Industry Code Description* also correlated with *Carrier Name* and *Carrier Type*, suggesting a link between industry and its associated insurance providers.

Although these correlations were observed, we retained all variables for preprocessing and will defer decisions on feature removal until the selection stage.

To further assess how features relate to Claim Injury Type, we analyzed their influence:

- Features like *Attorney/Representative, Carrier Type, COVID-19 Indicator, District Name, Gender, Industry Code, WCIO Codes, Average Weekly Wage,* and *IME-4 Count* showed distinct distributions across **Claim Injury Type**, supporting their predictive potential [Figure 10, Figure 11 and Figure 12].

- Missing or zero values in features like *First Hearing Date*, *C-2 Date* and *Average Weekly Wage*, affects target distributions. To address this, we created flags for these features to their impact is incorporated into the model [Figure 13 and Figure 14].

2.2 Splitting Data

To ensure reliable model evaluation, we split the dataset into training and validation sets using the hold out method with a 70/30 split, ensuring stratification to maintain the distribution of the target variable. From this point onward, we disregard the target variables – *Claim Injury Type* and *Agreement Reached* – in our analysis. Also, it is worth noting that, although we are predicting two target variables, when performing data split, the choice of target was not relevant. However, the approach after preprocessing will change according to each target variable.

2.3 Preprocessing

2.3.1 Outlier Removal

As identified during the EDA, boxplots revealed the presence of outliers that needed to be addressed. To manage these, we used a combined approach of the Interquartile Range (IQR) method and Winsorization at the 99.5th percentile. Specifically, claims exceeding the IQR upper bound were removed only if the upper bound was greater than the 99.5th percentile. For values beyond the 99.5th percentile that didn't meet this criterion, we applied Winsorization to limit the influence of extreme outliers while maintaining the integrity of the dataset. This approach led to the removal of 0.4% of the observations from the dataset [Figure 15].

2.3.2 Imputation of Missing Values

To address missing values, we focused on maintaining the integrity of the dataset while minimizing information loss and preserving its statistical characteristics [Table 5].

During the analysis, we found that 1% of rows had missing values across all features related to codes and descriptions, which are crucial for predicting claims compensation, as they detail the claims and injuries. Since *Industry Code Description*, which these features depend on, was also missing, imputing them would lead to cascading errors and reduce data reliability. To prevent this, we removed these rows. Most missing value were imputed using the most frequent category within specific groups based on correlated features. However, there were some exceptions:

- *Accident Date, C-2 Date,* and *C-3 Date* were imputed sequentially using observed patterns. However, *First Hearing Date* was not imputed, as the missing values indicate no hearing occurred.
- *Age at Injury* and *Birth Year* were imputed by calculating differences relative to Accident Date.
- For *Average Weekly Wage*, missing values were imputed using a Random Forest based on correlated features. This ensured the distribution of Average Weekly Wage remained consistent after imputation [Figure 16].
- Features like *Alternative Dispute Resolution* (<0.01% missing) and *Carrier Type* (0.32% missing) had their missing values imputed using the overall mode.

Further details about the strategy applied to each individual feature can be found in [Table 6].

2.3.3 Feature Engineering

Through the implementation of different strategies to create new features, we were able to gain additional insights about our dataset.

We used 01/01/2020 as a reference date to mark the start of the assembly period. Using this date, we calculated the difference between it and the other dates, resulting in three new features: *Assembly Date Referenced, Accident Date Referenced*, and *First Report Referenced*.

To explore the potential impact of who submits the first report, we introduced a feature called *First Report Submitter***.** We also created *Days Until First Report* which measures the time from *Accident Date* to the first claim receipt (*C-2 Date* or *C-3 Date*) and *Days Until Assembly which* represents the time from Accident Date and First Assembly Date. Although these features are highly correlated, further analysis will help determine which one should be removed during feature selection.

In addition, to assess whether seasonality plays a role in the dataset, we added *Assembly Quarter*, dividing assembly dates into quarterly periods. Finally, we created *Log Average Weekly Wage* to smooth extreme values, reduce the high variance of the original variable, and stabilize its distribution.

2.3.4 Encoding

To prepare our categorical variables for modelling purposes, we needed to further treat and transform the data. For this reason, features with only two unique values were converted to binary format (0 and 1) and for the remaining ones, **Frequency Encoding** was applied to replace categories with their occurrence counts, capturing information about the distribution while maintaining numerical consistency for further steps.

2.3.5 Scaling

To prepare our numerical variables, we applied scaling to standardize feature distributions. This step ensured that all features were comparable and prevented variables with larger ranges from disproportionately influencing the model. After evaluating several scaling methods, we chose **Power Transform** as it reduced skewness and normalizes the distribution of variables, improving the overall distribution of features. Further details on the methods considered and their evaluation can be found in [Table 7 and Figure 17].

3. Feature Selection

The process to assess and rank features depends on the data types we identified earlier during data preprocessing and was performed separately for each target variable, following the same strategy for both.

Our feature selection process prioritized identifying impactful features while addressing redundancy. By categorizing features into types like numerical and binary, we applied statistical methods: **ANOVA** (only for numerical) and **Chi-squared** tests (only for binary), and **Embedded Methods**, such as **Elastic Net**, **LASSO**, and **Random Forest**. The threshold for each method in this process is dynamically determined based on the relative importance of each variable. It is calculated as a fraction of the maximum feature importance, specifically by dividing the maximum importance by the total number of features. This adaptive threshold ensures that only variables with a proportionally significant contribution to the model are retained.

To ensure robustness, we used a majority voting strategy, retaining features selected by at least two out of the four applied methods for each feature. Following this, we analyzed correlations among the selected features using **Spearman's correlation** and removed highly correlated ones to reduce redundancy.

For predicting ***Claim Injury Type***, the most influential feature identified across all methods was *IME-4 Count*. Additional features of considerable importance included *Average Weekly Wage*, *Gender*, the absence of hearing (*First Hearing Date Missing*), and various date-based measures. In contrast, for predicting ***Agreement Reached***, the most decisive features were related to the insurance provider, including *ce_Carrier Name* and *ce_Carrier Type*. Temporal variables - such as *Accident Date Referenced*, *Assembly Date Referenced*, and *First Report Referenced* - also contributed notably to these models. Although descriptions of the nature and cause of injury proved to be relevant for predicting *Claim Injury Type*, they held comparatively less importance in predicting *Agreement Reached*. For more details, consider [Table 7, Table 8 and Table 9].

4. Modeling Claim Injury Type

Due to the highly imbalanced dataset - a challenge identified early in our analysis - we decided to implement the models using the original sampling and an under-sampling approach with **Tomek Links [1]**. We opted for an under-sampling technique since oversampling a dataset this size would increase the computational challenges and inefficiencies we were already encountering.

To assess our models, we used **Stratified Cross-Validation with 4 and 5 folds**, ensuring a balanced class distribution in each fold. To minimize computational overhead, we performed hyperparameter tuning using **Random Search**. To offset the constraints when compared to grid search, we first performed a preliminary parameter study on all models using a hold-out validation approach, which is faster and allowed us to refine the parameter ranges for Random Search. The models tested included Decision Tree, Logistic Regression, Neural Networks and Ensemble Methods – **Hierarchical Gradient Boosting**, **XGBoost [2]**, **LightGBM [3]** and **Random Forest**.

Through this preliminary analysis, using comparative boxplots, we gained insights into the parameter grids, which informed our tuning strategy. For example, when optimizing the *max_depth* parameter for XGBoost [Figure 18], we evaluated values from 3 to 9. The results indicated that increasing *max_depth* led to significant overfitting, as evidenced by the performance metrics. Consequently, we adjusted the parameter grids for all models based on these observations as a hyperparameter optimization. The models that achieved the highest validation F1 Scores using the hold-out method were further evaluated with a more robust strategy: cross-validation [Figure 19]. To evaluate the models, we focused on three key metrics: precision, recall, and F1 Score. The F1 Score balances precision, which measures the proportion of correctly predicted positive cases, and recall, which assesses the model's ability to identify all actual positive cases. This makes it a particularly effective performance metric for imbalanced datasets like ours.

Additionally, even though we thoroughly analyze the features and their behavior, as detailed in the previous section, we employ feature selection within the cross validation. By performing feature selection within each fold, the features chosen are specific to the training data of that fold, ensuring that the process remains unbiased and avoids data leakage. This approach allows the model to adapt to varying feature subsets across folds, ultimately mitigating the risk of overfitting and providing a more accurate assessment of model performance (Trevor Hastie 2017). The results obtained are illustrated in the following table:

Models	**Train F1 Score** (1)	**Val F1 Score** (1)	**Train F1 Score** (2)	**Val F1 Score** (2)
Random Forest	0.381	0.366	0.415	0.366
LightGBM	0.487	0.372	0.471	0.370
Decision Tree	0.401	0.379	0.442	0.368
XGBoost	0.415	0.362	0.454	0.348
Logistic Regression	0.373	0.388	0.397	0.388
HGBoost	0.429	0.388	0.460	0.378

Table 1 - Model Results for Claim Injury Type with Cross Validation. (1) No Resampling; (2) Tomek Links Under sampling

For further details about the optimal hyperparameters obtained using this procedure, consider [Table 10].

Among the evaluated models without any resampling techniques, **Hierarchical Gradient Boosting (HGB)** and **Logistic Regression** consistently achieved the highest F1 Scores, both showing a validation score of 0.388.

This performance underscores the ability of these models in predicting Claim Injury Type. Specifically, HGB, an ensemble method, utilizes boosting techniques to enhance predictive accuracy, whereas Logistic Regression is valued for its simplicity and effectiveness in linear classification tasks. We deployed both models and ultimately preferred HGB, as it achieved a higher Kaggle score of 0.377.

The confusion matrices [Figure 20] reveal that both models excel at predicting the dominant classes ("2. NON-COMP" and "4. TEMPORARY"). However, HGB performed slightly better on minority classes, at predicting "5. PPD SCH LOSS" (Permanent Partial Disability - Scheduled Loss), "7. PTD" (Permanent Total Disability) and "8. DEATH". Despite these strengths, all models faced challenges in accurately predicting the category "6. PPD NSL" (Permanent Partial Disability - Non-Scheduled Loss). This difficulty likely stems from the class imbalance in the dataset, where this category is underrepresented compared to others, leading to limited training examples for the models to learn from. We can also observe from the confusion matrices that the classes are often misclassified as neighboring classes. This suggests that the models struggle to differentiate between similar or closely related classes, probably because these classes share characteristics or have similar distributions.

The results also reveal discrepancies between training and validation score across models, indicating the presence of overfitting. The gaps range from 1.5% in Random Forest to 11.5% in LightGBM, with an average **overfitting rate of approximately 4%**. LightGBM exhibits the most significant overfitting, as evidenced by its high training score of 0.487 compared to validation score of 0.372. This tells us that while LightGBM is highly effective at modeling the training data, its performance on unseen data is weakened. Simpler models such as **Decision Tree** and **Random Forest** demonstrate smaller gaps between training and validation scores, with overfitting rates of 2.3% and 1.6% respectively, indicating a good trade-off between complexity and generalization.

Additionally, the results also reveal the impact of Tomek Links on model performance. While under sampling can improve training performance across all models, it does not consistently lead to improved validation performance and may, in fact, intensify overfitting in almost all models. Logistic Regression stands out as the only model that kept its validation despite increased training performance. Conversely, ensemble methods like Random Forest and HGBoost showed increased training scores without significant validation gains, suggesting that under sampling may not be beneficial and may lead these models to fit more closely to the reduced training data, thereby increasing the risk of overfitting.

5. Modelling Agreement Reached

In this section, we present and analyze the results obtained from predicting Agreement Reached. Given the binary nature of the target variable – where 1 means that an agreement was reach and 0 indicated the absence of an agreement without the involvement of WCB - and the inherent computational constraints, we strategically opted to utilize the Hold-Out Method for model evaluation instead of the more computationally intensive cross-validation approach. For optimization we used random search with 50 iterations. The results for each model are illustrated in the table below.

Models	Train F1 Score (1)	Val F1 Score (1)	Train F1 Score (2)	Val F1 Score (2)
Logistic Regression	0.514	0.492	0.505	0.489
Random Forest	0.567	0.573	0.567	0.575
XGBoost	0.547	0.523	0.576	0.536
Perceptron	0.596	0.599	0.565	0.598
Multi-Layer Perceptron	0.5803	0.590	0.598	0.569

Table 2 - Model Results for Agreement Reached using Hold-Out Validation. (1) No Resampling; (2) Tomek Links Under sampling

For further details about the hyperparameters obtained using this procedure, consider [Table 11].

Among the evaluated models, we noticed that the Neural Networks (both **Perceptron** and **Multi-Layer Perceptron**) achieved the highest F1 Score during validation (0.599 and 0.590, respectively, without resampling), highlighting the performance of neural network in handling binary classification tasks

By removing overlapping observations using Tomek Links, we noticed a small change in the results. For the Perceptron, the resampling improved the model's ability to detect the minority class, thereby increasing recall. However, this improvement in recall was accompanied by a slight decrease in precision, as the number of false positives increased. This trade-off highlights the balance between sensitivity and precision when applying under sampling techniques [Figure 21].

6. Customized Ensemble Strategies

The primary objective of this section is to develop a strategy to address the significant class imbalance in the dataset, which has led to poor performance in predicting underrepresented classes.

Given the multiclass nature of the main target variable – Claim Injury Type – and the inherent computational constraints, we used the **Hold-Out Method** to evaluate the performance of the models throughout this process.

Our chosen approach involves the customization of ensemble models, using individual binary classifiers, each trained to predict one of the eight classes of the target variable. Subsequently, we experimented with various methods to combine the predictions of these classifiers into a cohesive final output.

This layered approach allows for a more granular investigation of model performance across each class, enabling the customization of tailored strategies to determine whether they can yield better results than the previously obtained ones and better capture underrepresented classes.

6.1 One-Vs-Rest Strategy

To determine the base estimators, we employed the **One-Vs-Rest (OvR)** multiclass strategy, which transforms a multiclass classification problem into multiple binary classification tasks. Specifically, an estimator is trained to distinguish each class from all other classes. During prediction, each classifier provides a score (e.g. probabilities in the case of a logistic regression) during prediction, and the final prediction for an observation is determined by selecting the class with the highest score.

Implementation was done using 6 different base estimators using hyperparameters that avoid overfitting and that are computationally cheap, to ensure total model runtime is reasonable. Among these, **Logistic Regression Classifiers** provided the best results [Table 12], so they will be our reference models moving forward. These initial results serve as a benchmark to gauge the progress and efficacy of more advanced modeling techniques.

6.2 Weighted One-Vs-Rest

Building upon the OvR strategy, the Weighted One-Vs-Rest approach introduces specific weights to each class's predicted probabilities, thereby allowing certain classes to be prioritized during prediction.

To identify the optimal combination of weights that should be attributed to each class, we compared model results where all weights, except for one, remained constant. The results of these experiments, as illustrated on [Figure 22], provided strong insights as to how weights should be assigned. An important caveat to note is that the best combination of weights may not necessarily be the combination of the best weight for each individual class. There is also the possibility that choosing the optimal weights based on this split could lead to a model that does not generalize well.

To mitigate these issues, we defined a small interval for each class weight rather than assigning fixed values. These intervals served as parameter ranges for Random Search and were incorporated into a **Stratified Cross Validation** with 4 folds to find the weight combination that would perform best on average across all folds.

The best combination of weights produced an average validation F1 Score, as detailed in [Table 13].

The results showed an improved performance for underrepresented classes (classes 3, 6, and 8) at the expense of a slight decline in performance for majority classes (classes 2 and 4). Specifically, while recall for minority classes increased, precision slightly decreased due to an uptick in false positives. Despite these trade-offs, the overall macro F1 Score shows improvement. However, class 7 remained too underrepresented and would need too high of a weight to be captured. This would lead to a small improvement in the class at the expense of a large decrease in performance for the remaining classes, so it is preferrable not to assign any weight to this class altogether.

We had concerns that this general application of a weighting scheme might have been comparatively more prone to overfit to the specific characteristics of a split, however, concerns regarding the generalizability were addressed by observing consistent validation F1 Scores across all folds, ranging from 0.413 to 0.424, suggesting that the model maintains robust and is not overfitting to specific data splits.

6.3 Stacking Ensemble

Our final approach involves the use of stacking to create a meta-model where the first-level classifiers are the logistic regression models trained for each class. The predicted probabilities from these base classifiers create a new dataset and serve as input features for the meta-model, which synthesizes these predictions to generate the final classification output.

A critical consideration in stacking is the risk of overfitting arising from training base estimators and generating new datasets on the same training data. To mitigate this, we split the train set into 4 folds, and for each iteration, the base estimators were trained on 3 folds and made predictions on the remaining fold. This leads to a set of **Out-of-Fold** predictions, which were concatenated to form the first-level transformation of the train set. Subsequently, base estimators are then trained on the entire training set and used to predict on the validation set, thereby forming its first-level transformation.

This procedure enabled us to obtain a transformed training and validation set, which then were used to train different models, compare their results, and choose the best performing one to finally obtain a meta-model. Using a methodology like the one in section 4, we used Random Search to find the best parameters for each algorithm, that yielded the results shown in the table below.

They were also compared with the results from the models obtained from the same algorithms and combination of parameters, except fitted on the original dataset. Despite the different datasets they are trained on, both these sets of models were trained on the same row-wise train and validation split of dataset, to ensure a valid comparison.

Model	**Train Macro F1**	**Validation Macro F1**	**Train Macro F1** (no stacking)	**Validation Macro F1** (no stacking)
RandomForest	0.415	0.390	0.398	0.353
XGBoost	0.422	0.387	0.472	0.340
LightGBM	0.598	0.382	0.634	0.345
LogisticRegression	0.376	0.379	0.374	0.381
DecisionTree	0.403	0.311	0.452	0.306

Table 3– Hold-out results using best found parameters. First rows consider a first-level transformation of the dataset.

The best performing meta-model was a Random Forest classifier. This meta-model surpassed our initial benchmark validation F1 Score of 0.357 but exhibited slightly lower performance when compared to the Weighted One-vs-Rest strategy. It is also important to note that this approach led to the models yielding better validation scores, but also considerably reduced overfitting on training data.

A notable limitation of this stacking approach is the lack of diversity among base models, as all base classifiers are Logistic Regression models. While this uniformity simplifies the transformed dataset and is more intuitively understandable, it reduces the ensemble's diversity. However, it is important to keep in mind that a fundamental principle of stacking is the diversity within the ensemble, as distinct models are prone to making different types of errors. This diversity allows the models to complement one another, thereby enhancing the overall performance of the meta-model. (Henriques 2023).

7. Conclusion

The primary goal of this project was to develop models capable of automating the decision-making process for workers' compensation claims, focusing on predicting two specific outcomes: the classification of injury claims and whether a claim could be resolved without the direct intervention of the WCB.

Our findings partially align with our initial expectations. While we anticipated that ensemble methods would perform well on complex and imbalanced datasets, the extent of class imbalance presented a greater challenge than expected, limiting model performance for minority classes. Despite these challenges, our customized ensemble strategies—including Weighted One-vs-Rest and Stacking—achieved significant improvements in predictive performance. The Histogram Gradient Boost and the Logistic Regression showed to be the most effective for predicting compensation categories, while neural networks performed best for predicting agreements reached.

Our results also revealed limitations. The models effectively predicted whether a claim is compensated or not, but they struggle to accurately classify specific compensation categories, particularly for underrepresented classes. At this stage, the models show promise in supporting preliminary claim evaluations, particularly for identifying claims that are likely to proceed to compensation.

One of the major challenges we faced was limited computational power. Certain models, such as the Support Vector Classifier (SVC), and experimental approaches - such as testing cross-validation with various scalers and encoders or performing exhaustive hyper-parameter tuning using grid search - were not feasible to execute efficiently on our available hardware.

Future research could focus on trying different scaling techniques and encoding methods for categorical variables, adding new features to improve predictions, and using more powerful computational resources to optimize model parameters and test more complex models. While we tested SMOTE to address class imbalance, it resulted in a dataset that was too large to process efficiently. On the other hand, under sampling helped manage dataset size but increased overfitting by removing valuable information from majority classes. Finding a better balance between these approaches, or exploring new resampling methods, could improve both the efficiency and accuracy of the models.

All Jupyter Notebooks and Python code used to develop this project are freely available at the following link: https://github.com/margaridabcardoso/ML_PROJECT_44.

Annex A: Visual Representations

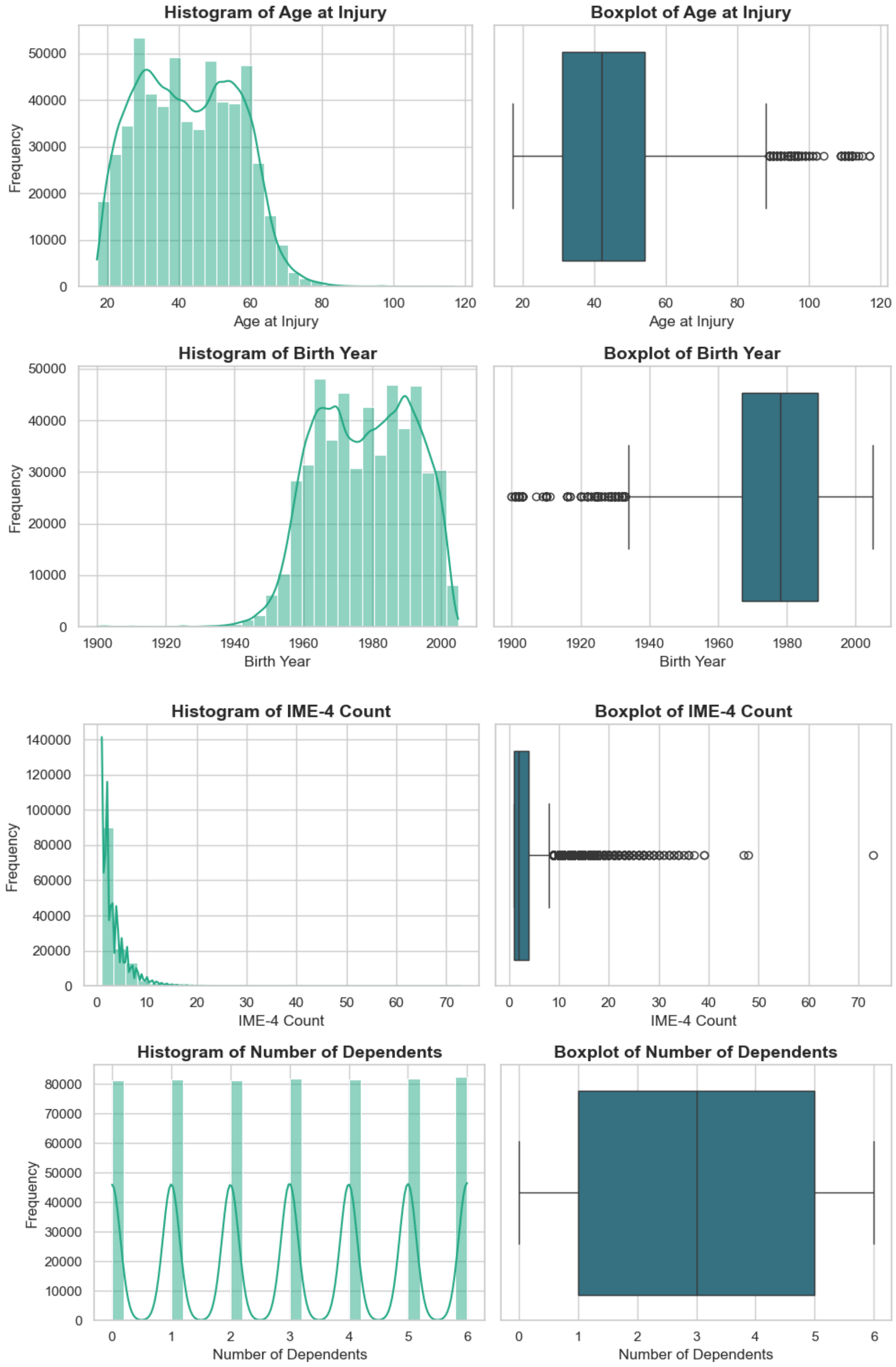


Figure 1 - Histograms and boxplots of the numeric features before outliers' removal

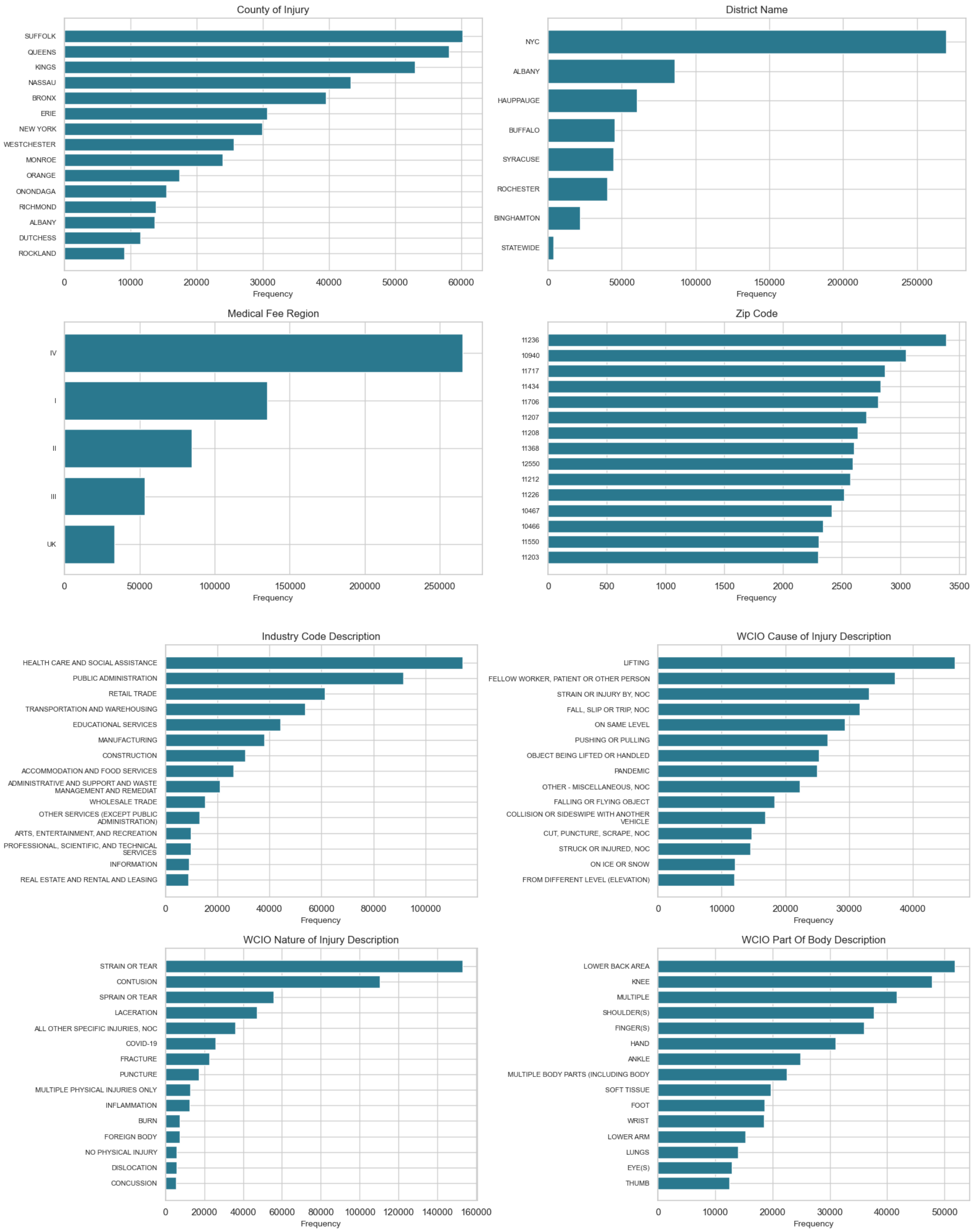


Figure 2 - Histograms of the 15 most common values for each category (1)

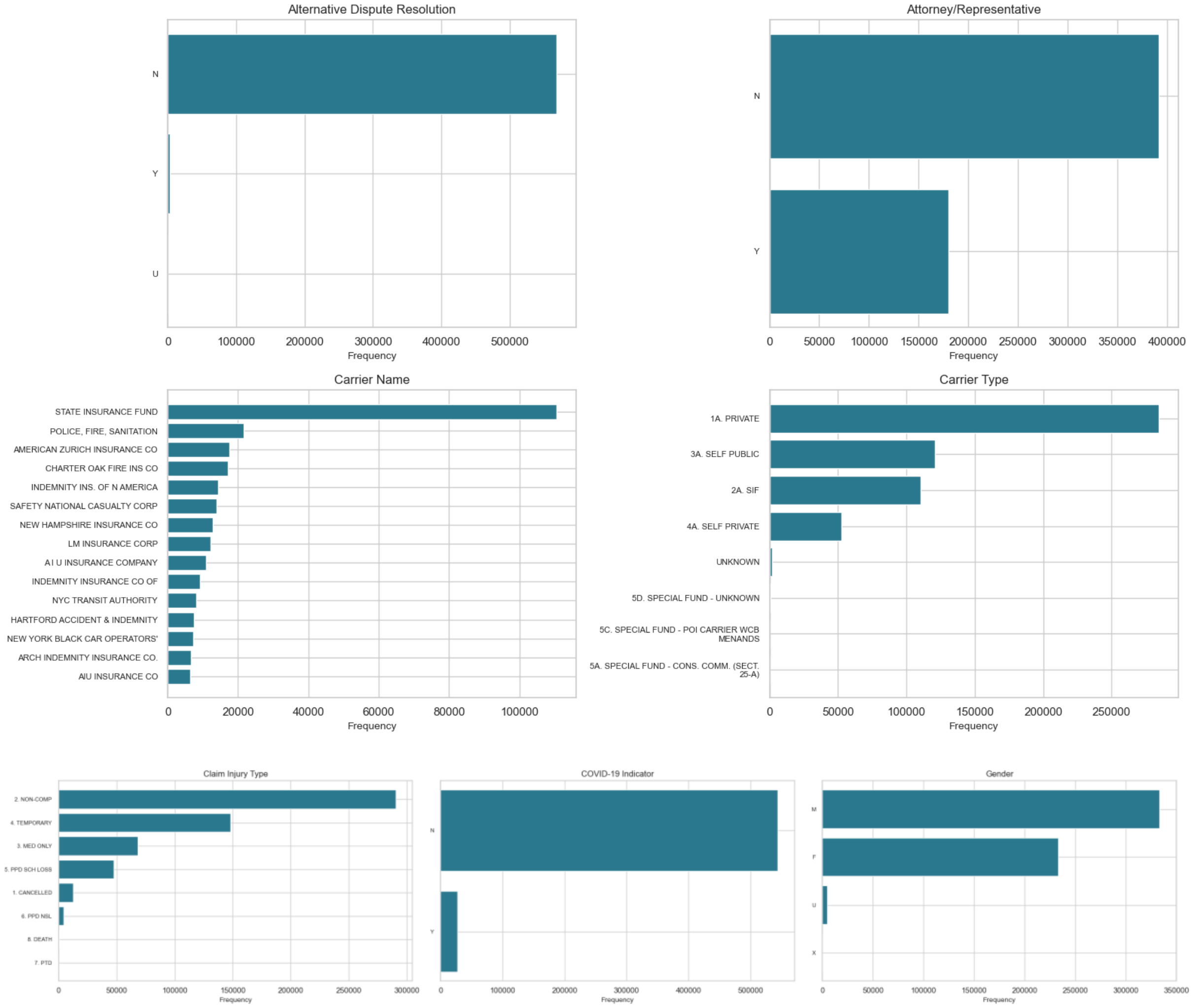


Figure 3 - Histograms of the 15 most common values for each category (2)

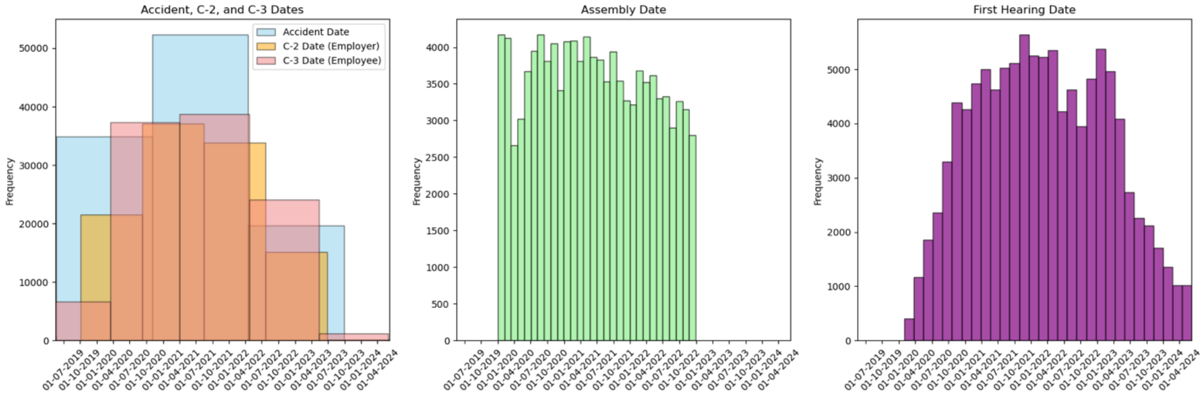


Figure 4 – Histograms of time-based features

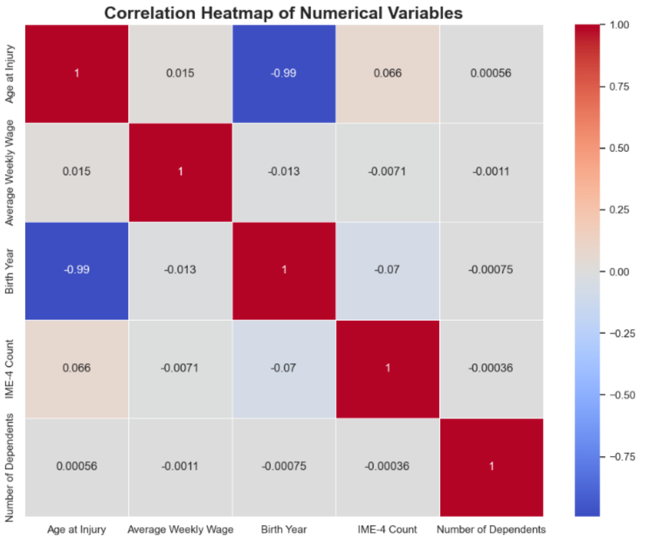


Figure 5 - Correlation Heatmap of Numerical Features

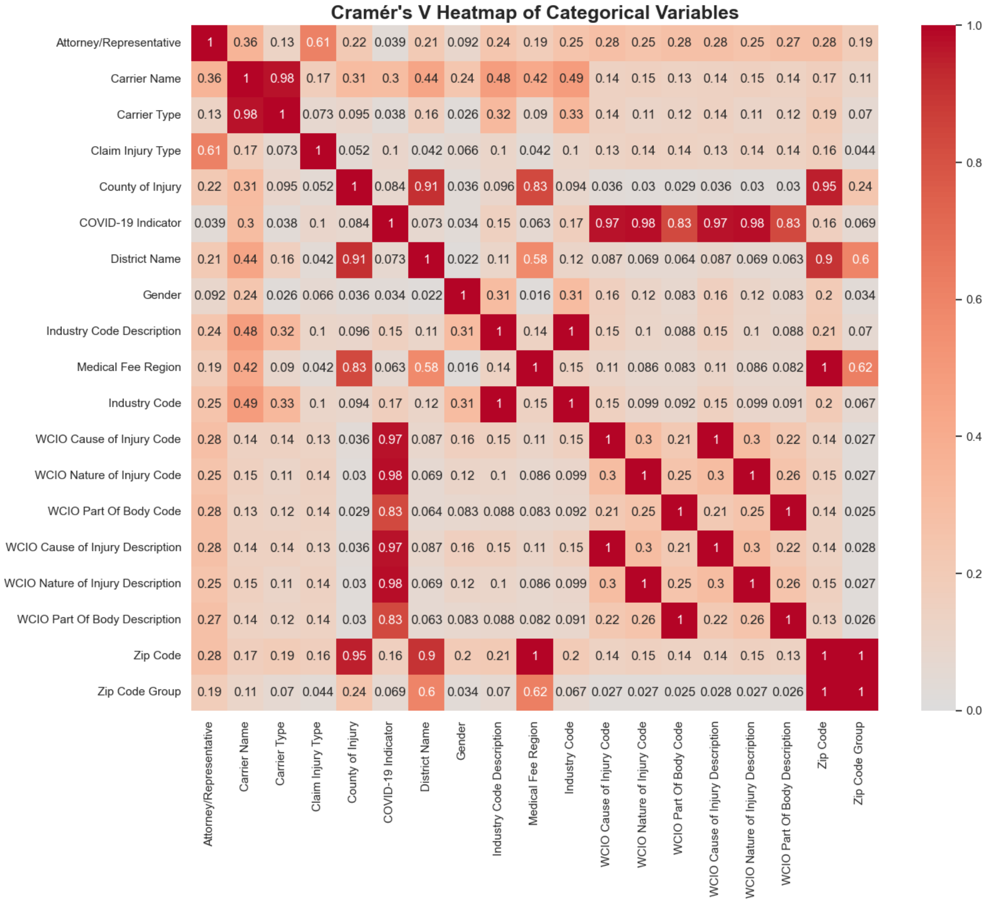


Figure 6- Crámer's V Heatmap of Categorical Features

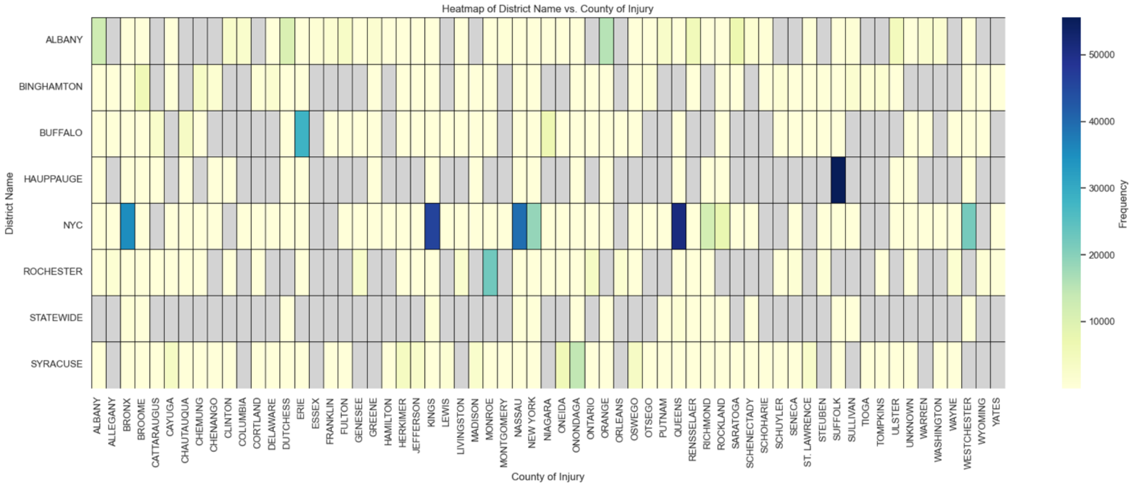


Figure 7 - Heatmap of the relationship between claims in each county and claims in each district

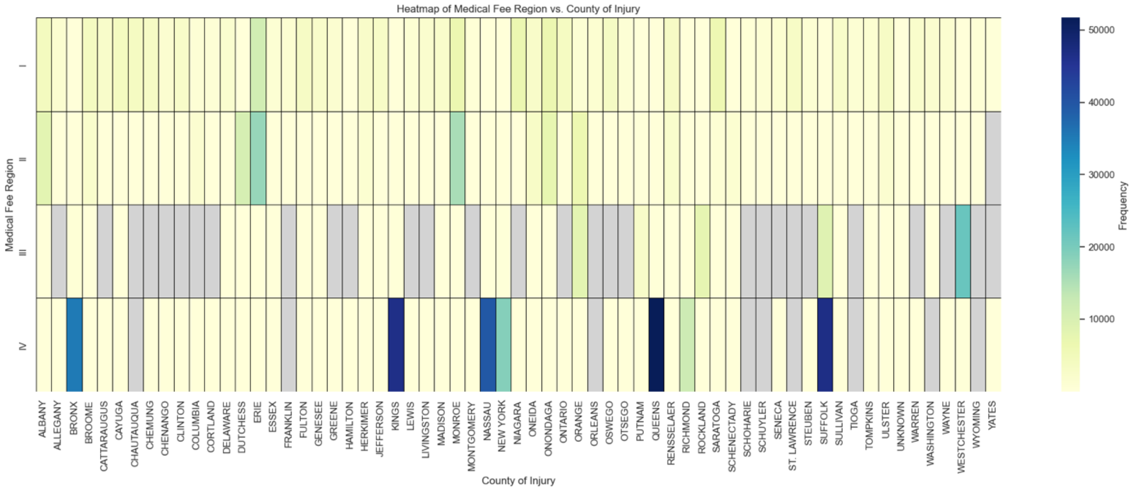


Figure 8 - Heatmap of the relationship between claims by medical fee region and claims by county of injury

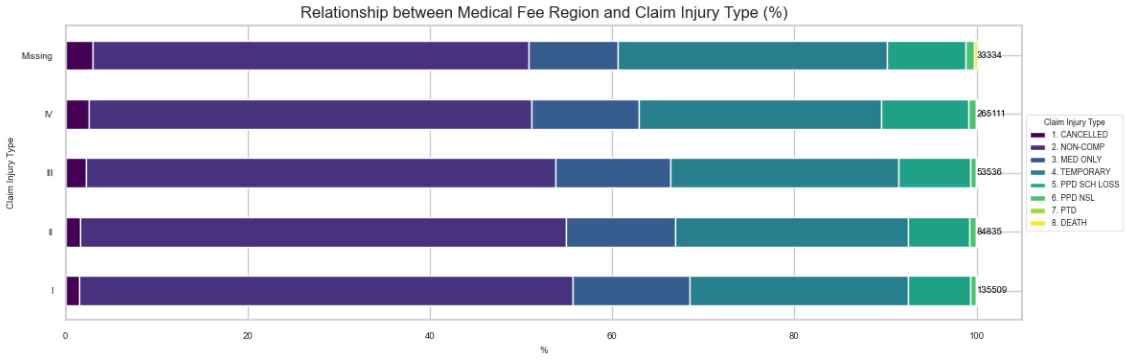
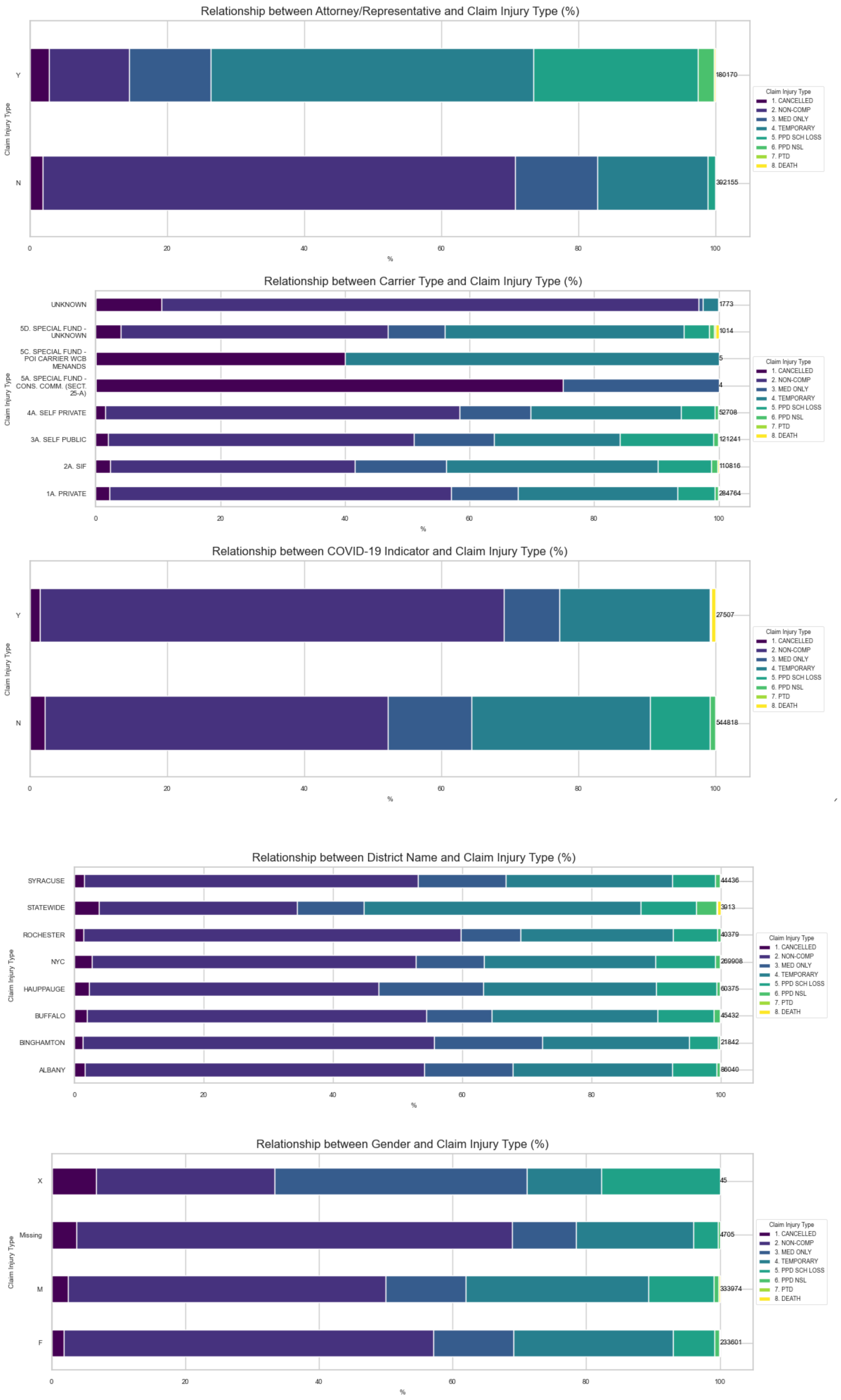


Figure 9 - Proportional representation of the target by Medical Fee Region


Relationship between Attorney/Representative and Claim Injury Type (%)

Relationship between Carrier Type and Claim Injury Type (%)

Relationship between COVID-19 Indicator and Claim Injury Type (%)

Relationship between District Name and Claim Injury Type (%)

Relationship between Gender and Claim Injury Type (%)

Figure 10 - Proportional representation of the target in categorical features (1)

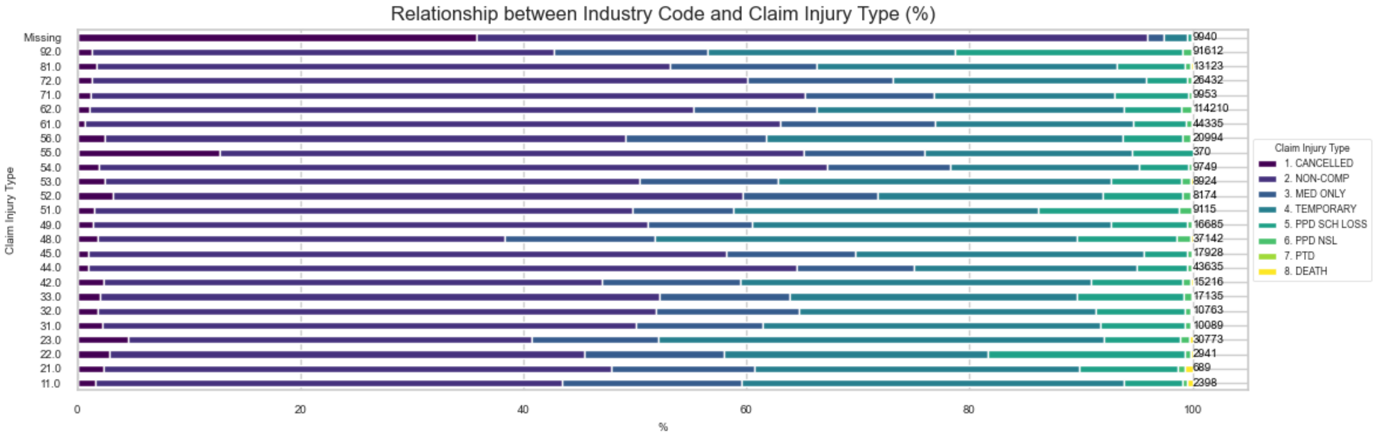


Figure 11 - Proportional representation of the target in categorical features (2)

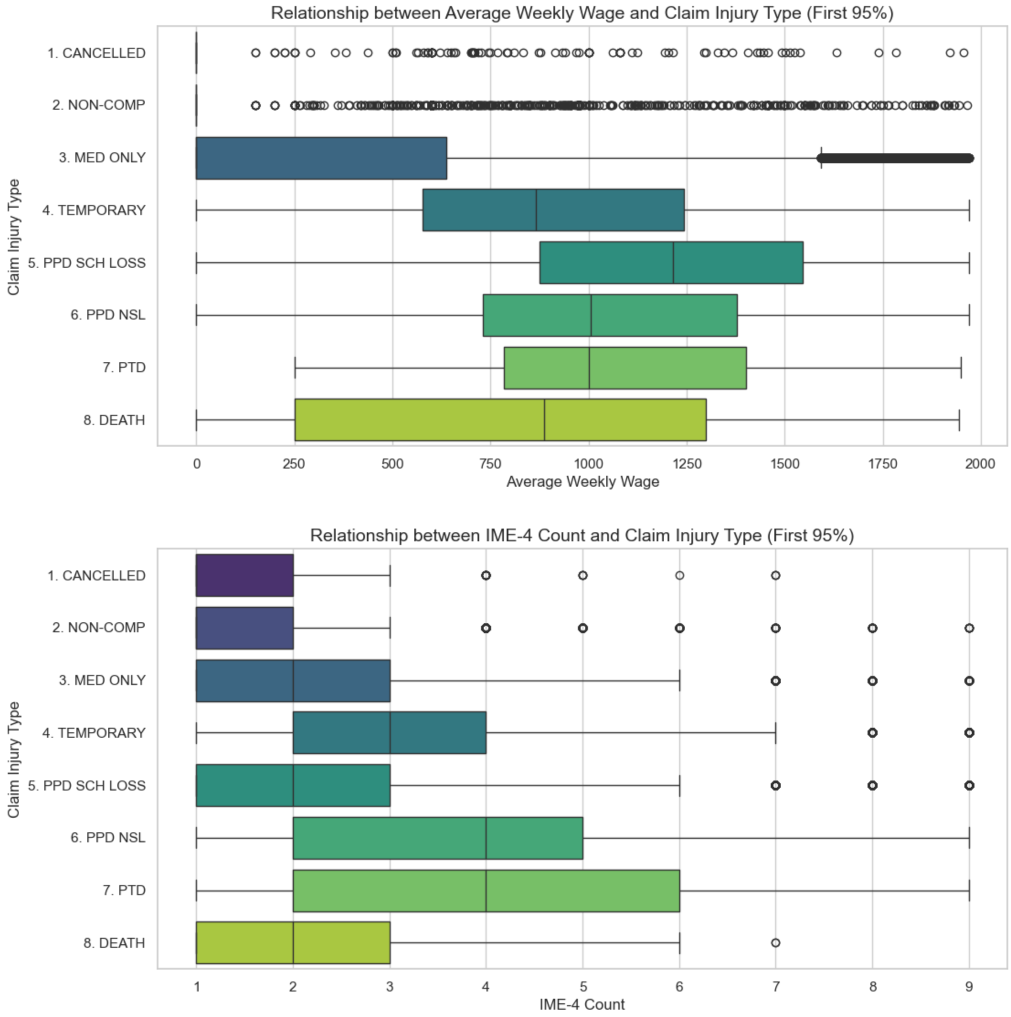


Figure 12 - Distribution of numerical features according to the target

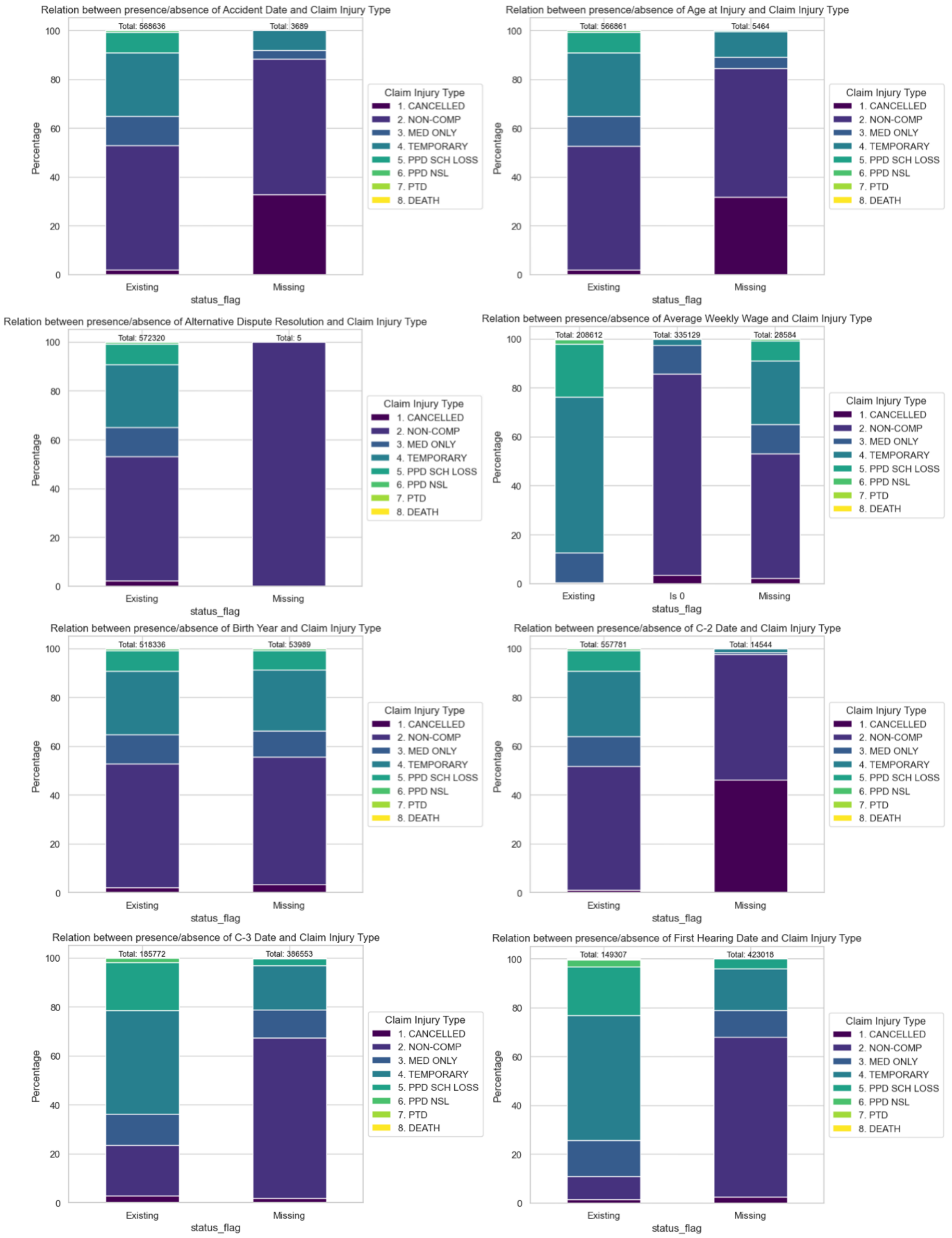


Figure 13 - Proportional representation of the target according to the presence/absence of features (1)

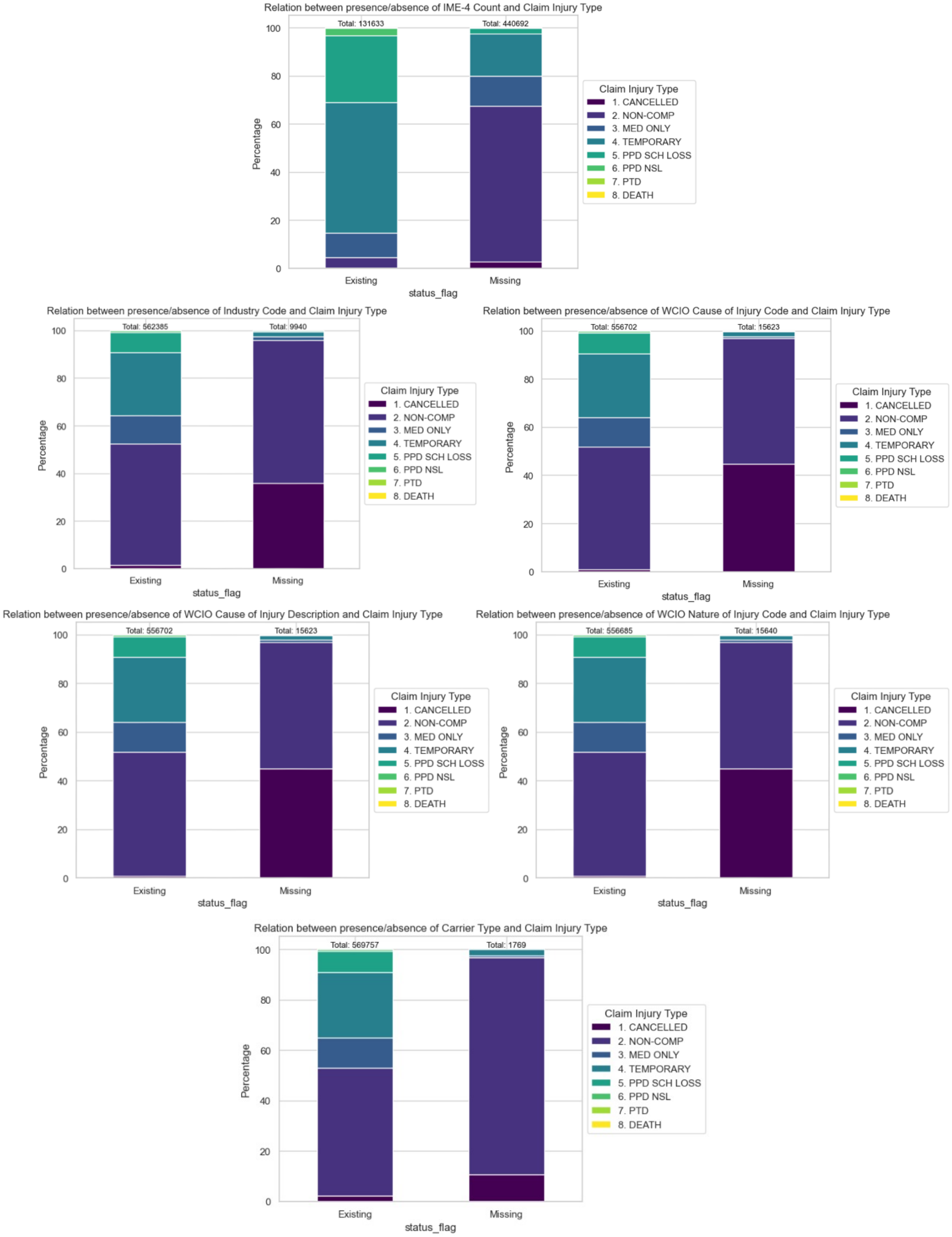


Figure 14 - Proportional representation of the target according to the presence/absence of features (2)

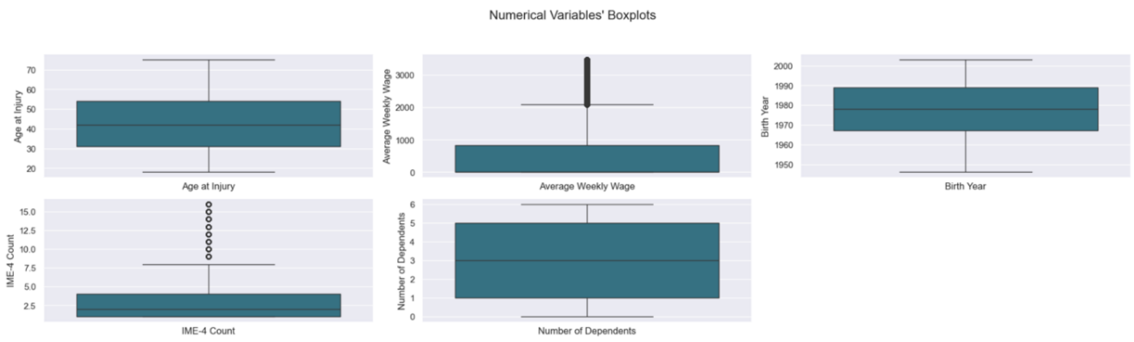


Figure 15 - Boxplots of the numeric features after outliers' removal

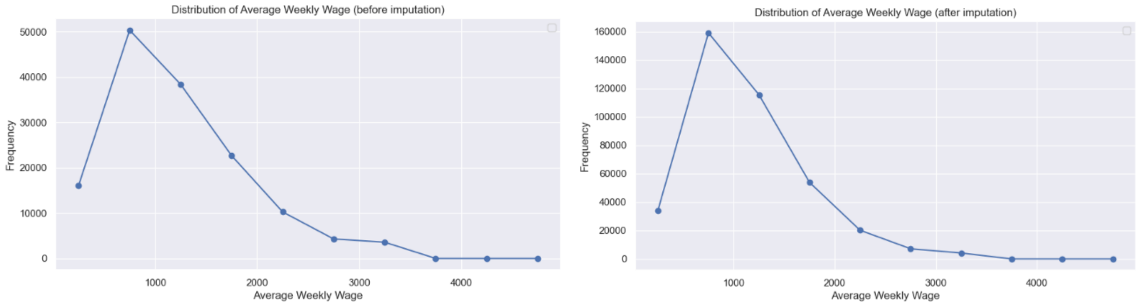


Figure 16 – Distribution of Average Weekly Wage before and after imputation

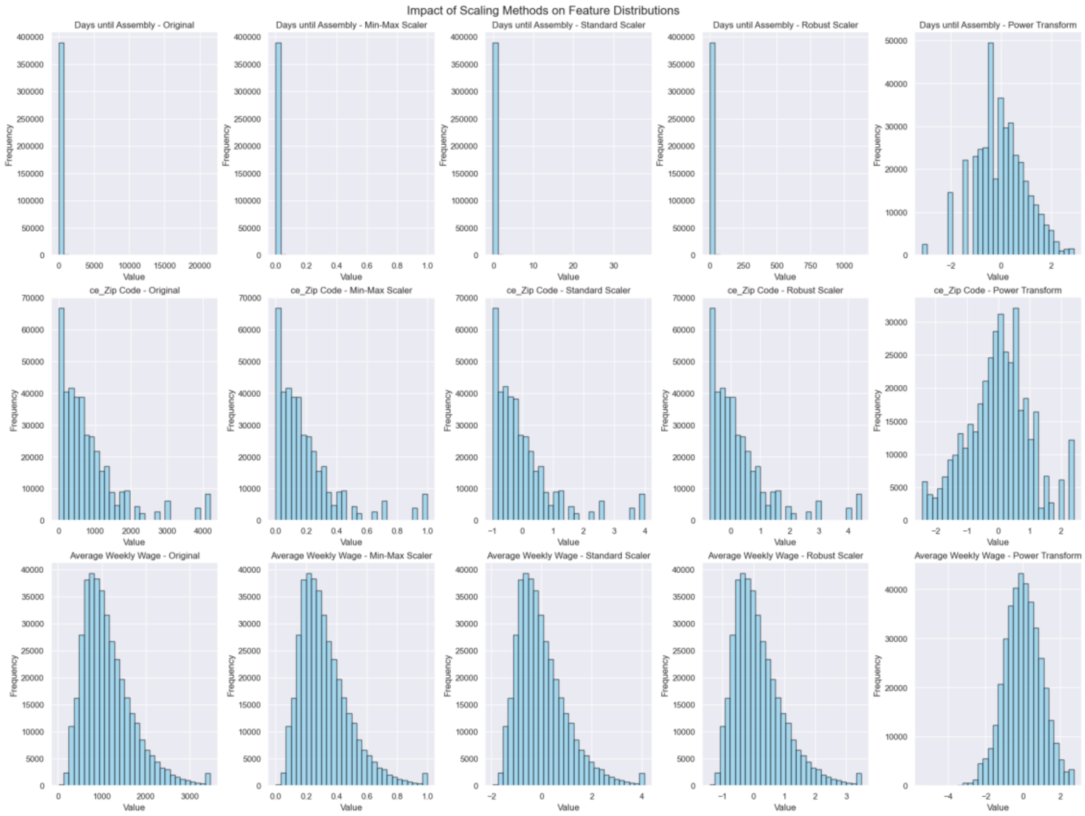


Figure 17 – Visualization of the impact of Scaling Methods on feature distributions

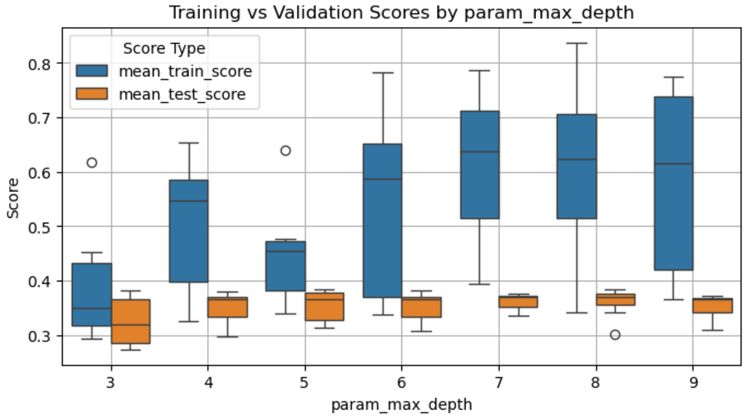


Figure 18 – Impact of max_depth parameter in XGBoost model

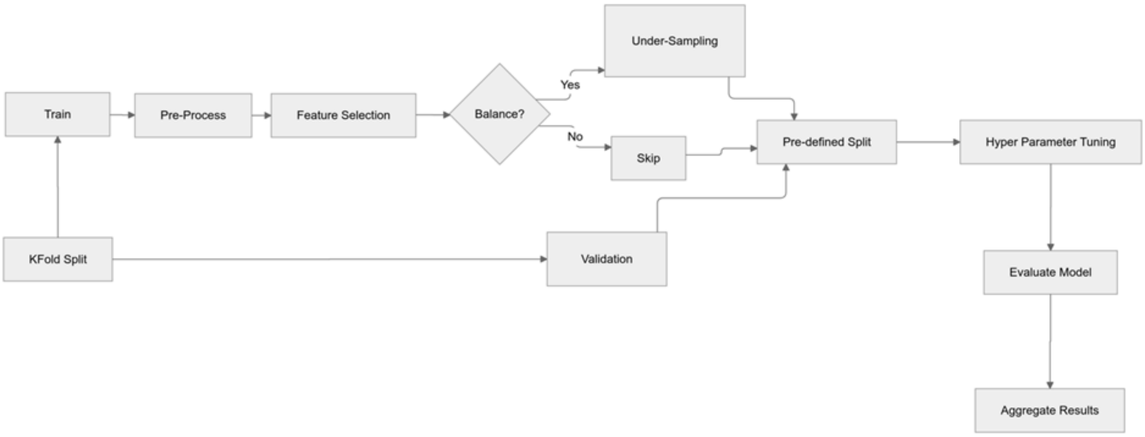


Figure 19 – Flowchart of the pipeline for cross validation

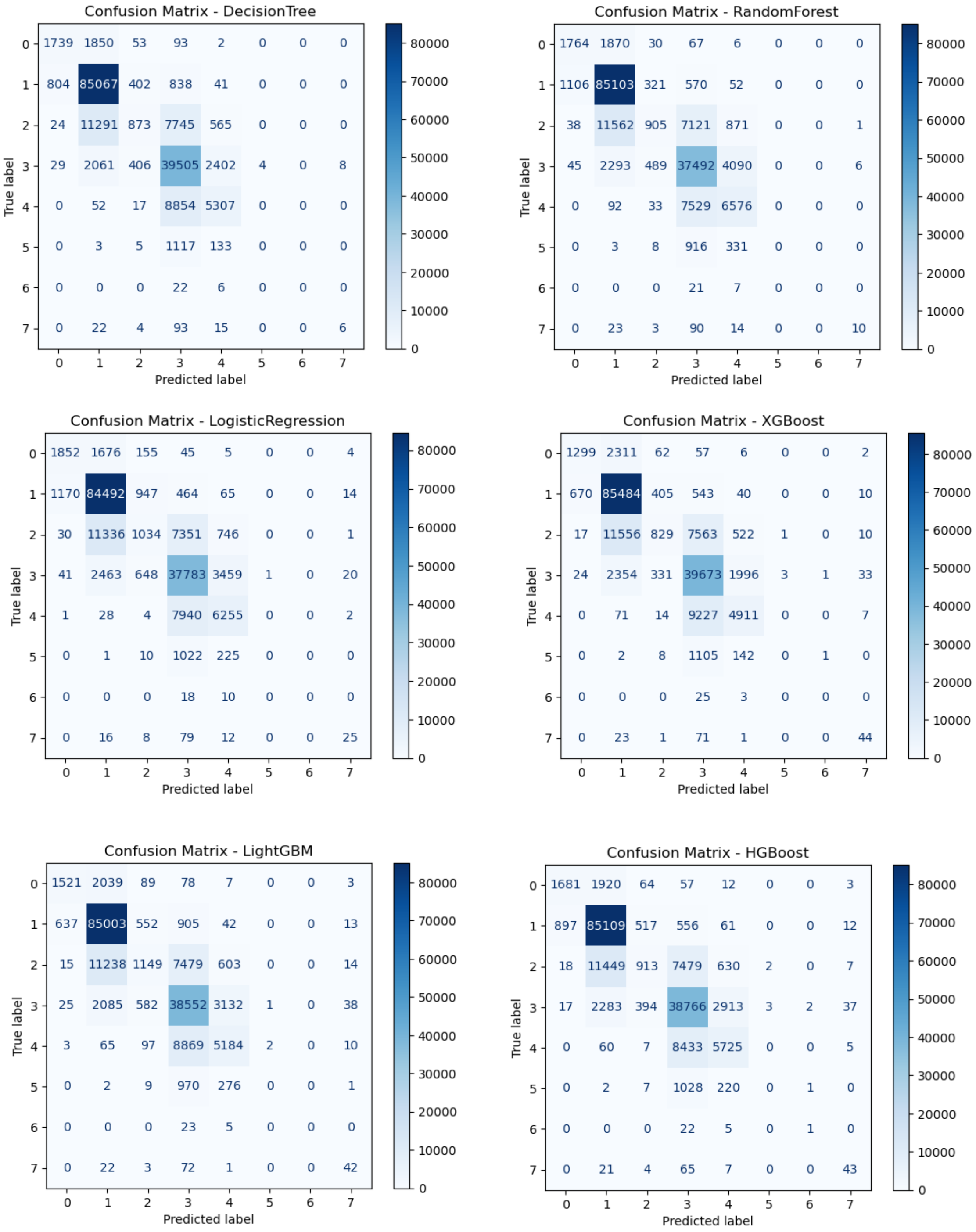


Figure 20– Confusion matrices illustrating the models' performance to predict Claim Injury Type

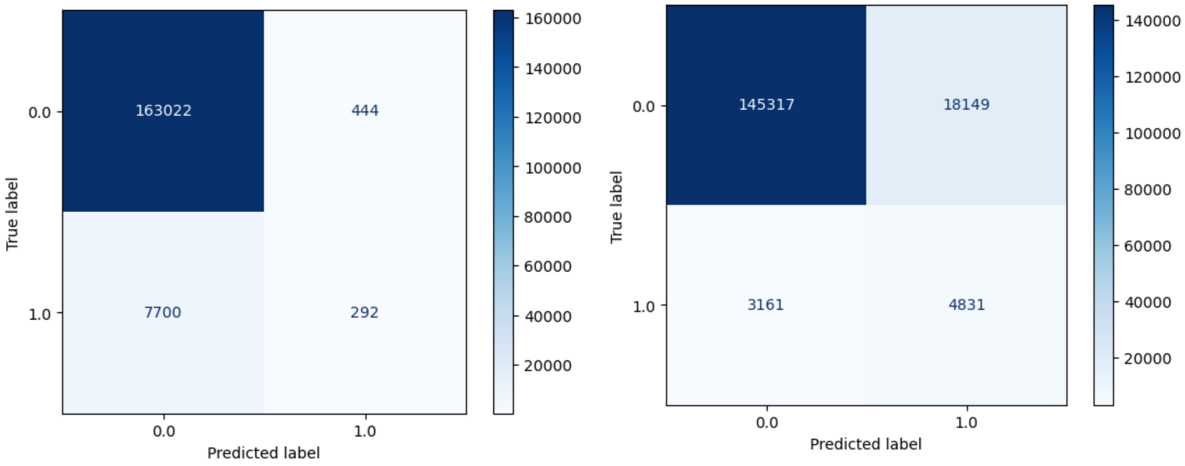


Figure 21 - Confusion Matrices - Perceptron Model predicting Agreement Reached (1) without under sampling techniques and (2) using Tomek Links

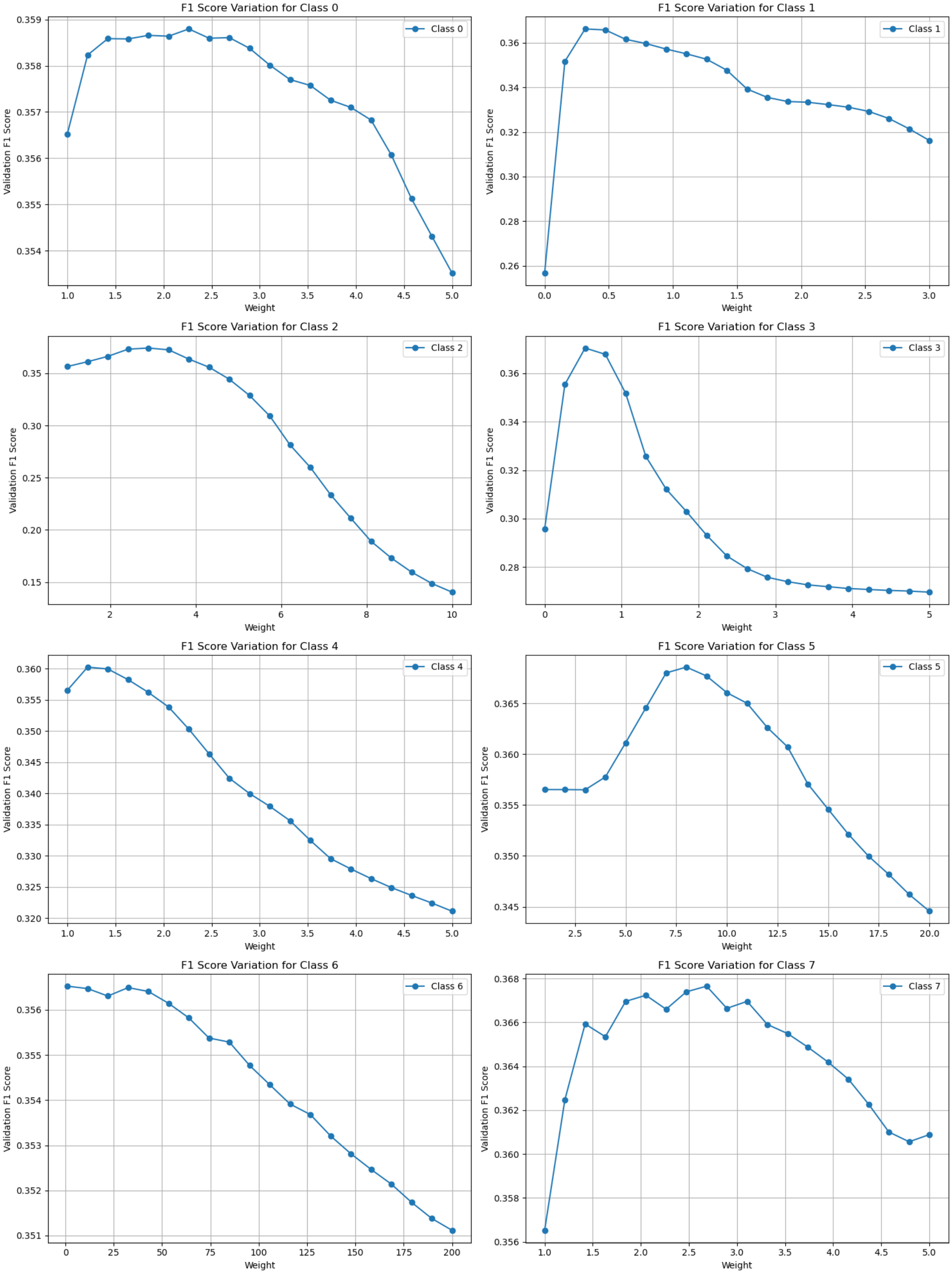


Figure 22 – Weighted OneVsRest Classifier validation results, by separately varying the weights for each class.

ANNEX B: TABLES

DIMENSIONS	FEATURES
DEMOGRAPHIC	Average Weekly Wage Birth Year County of Injury District Name Gender Industry Code Industry Code Description Medical Fee Region Zip Code
TIME-BASED	Accident Date Assembly Date C-2 Date C-3 Date First Hearing Date
INJURY-RELATED	Age at Injury Attorney/ Representative Carrier Name Carrier Type County of Injury COVID-19 Indicator IME-4 Count WCIO Cause of Injury Code WCIO Cause of Injury Description WCIO Nature of Injury Code WCIO Nature of Injury Description WCIO Part Of Body Code WCIO Part Of Body Description

Table 4 – Feature dimensions mapping

FEATURES	% MISSING
IME-4 Count	76.97%
First Hearing Date	73.89%
C-3 Date	67.53%
Medical Fee Region	5.77%
Zip Code	5.04%
Average Weekly Wage	5.03%
WCIO Part Of Body Description	2.98%
WCIO Nature of Injury Description	2.73%
WCIO Cause of Injury Description	2.73%
C-2 Date	2.54%
Industry Code Description	1.73%
Age at Injury	0.95%
Gender	0.83%
Accident Date	0.65%
Carrier Type	0.32%
Alternative Dispute Resolution	(<0.01) %

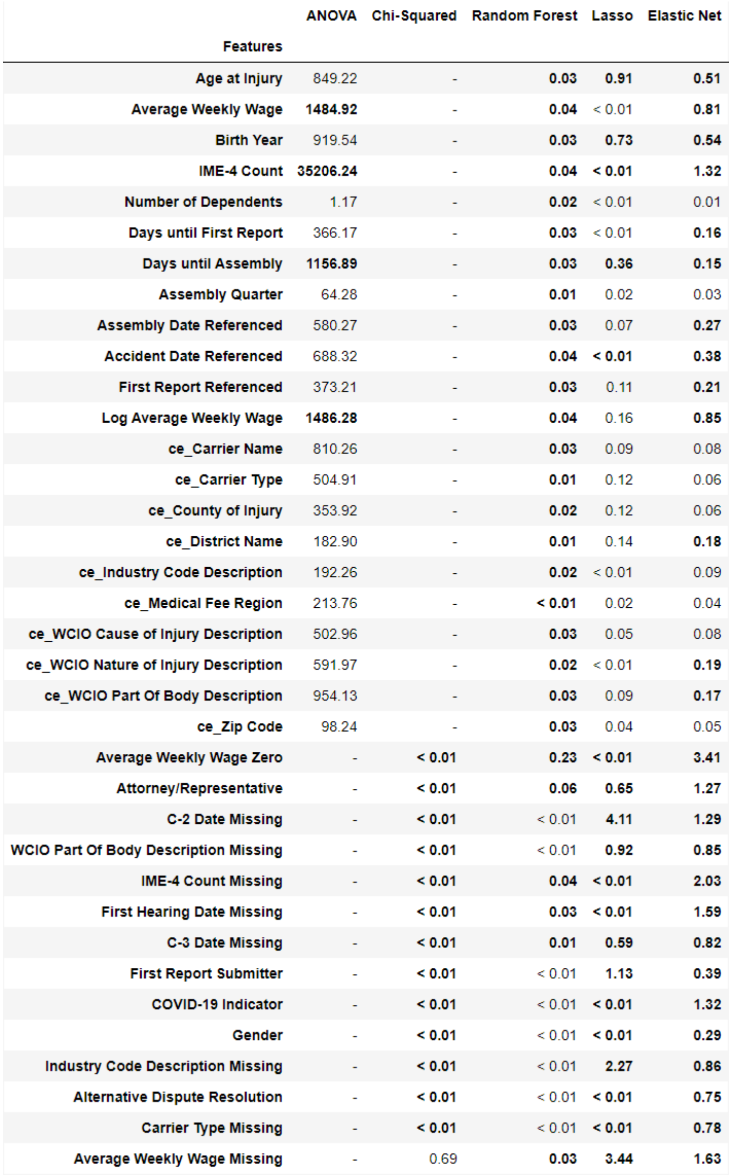
Table 5 – Proportion of missing values in each feature

Features	Strategy
Age at Injury	Impute missing values in a by calculating the year difference between the Accident Date and Birth Year. The remaining missing values are imputed the median age at injury. The same rationale is applied to Birth Year.
Accident Date	Imputation follows these steps: 1. Determine the earliest date among C-2, C-3, and Assembly for claims with missing Accident Date. 2. Impute Accident Date using the median of the appropriate series based on the earliest date.
Alternative Dispute Resolution	Imputation of missing values with the mode.
Average Weekly Wage	Imputation is done using a Random Forest regression model that uses as predictors Age at Injury, Industry Code, Medical Fee Region and Gender.
C-2 Date and C-3 Date	Imputation of missing dates is based on the median difference between a reference date - Accident Date and the target date to impute – C-2 and C-3 Date.
Carrier Type	Imputation of missing values with the mode.
First Hearing Date	We do not impute it since the information we pretend to take from this feature is if this date is missing or not, so we replace this feature with the one previously created.
IME-4 Count	Imputation of missing values with zero values. The reason behind this decision is that the lack of medical reports means they were not needed.
WCIO Cause of Injury Description	Imputed with the mode of the WCIO Cause of Injury Description for each Industry Cod Description e. If this methodology fails, imputation is made with the overall mode. Although WCIO Cause of Injury Description doesn't appear to be related to the Industry Code Description, expert judgement tells us otherwise.
WCIO Nature of Injury Description	Imputed with the mode of the WCIO Nature of Injury Description for each Industry Code Description. If this methodology fails, imputation is made with the overall mode. Although WCIO Nature of Injury Description doesn't appear to be related to the Industry Code, expert judgement tells us otherwise.
WCIO Part Of Body Description	Imputed with the mode of the WCIO Part Of Body Description for each Industry Code Description. If this methodology fails, imputation is made with the overall mode. Although WCIO Part of Body Description doesn't appear to be related to the Industry Code Description, expert judgement tells us otherwise.
Gender	Imputed with the mode of the groups specified by Industry Code Description, Average Weekly Wage and County of Injury.
Industry Code	Imputed with the mode of the groups specified by Age at Injury, Average Weekly Wage and County of Injury.
Medical Fee Region	Imputed with the mode of the Medical Fee Region of each County of Injury.
Zip Code	Imputed with the mode of the groups specified by County of Injury and Medical Fee Region.

Table 6 - Strategy for Missing Values Imputation

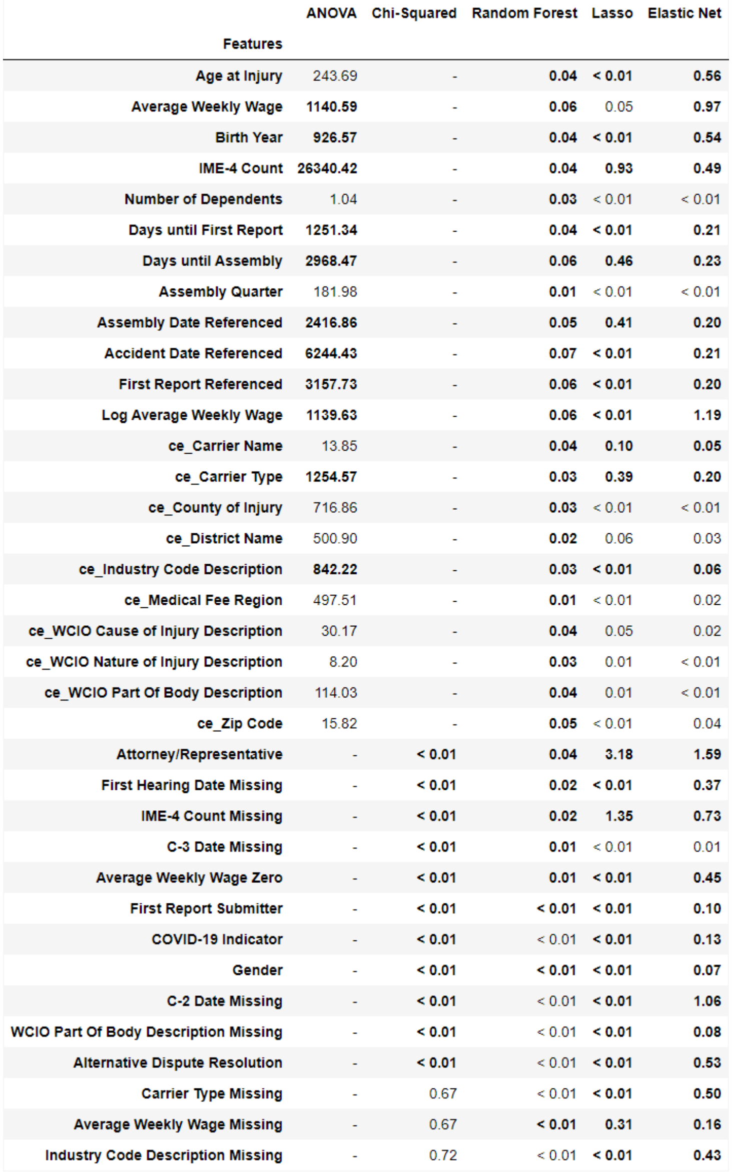
Scaler	Description	Observations
Min-Max Scaler	Rescales features to a specified range, typically [0, 1].	Effective in normalizing the features but retained skewness in data distribution (for example in IME-4 Count).
Standard Scaler	Standardize features by removing the mean and scaling to unit variance.	This scaler assumes data follows a Gaussian distribution. It was less effective for features with outliers.
Robust Scaler	Centers data by subtracting the median and scales according to the IQR, reducing the influence of outliers.	This approach was used when we hadn't refined the outlier treatment, as it reduces the impact of outliers. However, after refining our outlier treatment, we concluded that other scalers (e.g., Power Transform) were more suitable for achieving normality.
Power Transform	Applies a power transformation to make data more Gaussian-like, stabilizing variance and minimizing skewness.	Power Transformation encompasses a family of parametric transformations, including the **Yeo-Johnson** transformations, which are designed to stabilize variance and make the data closer to a normal distribution. This scaler effectively reduced skewness and improved data normality, making it the final selected scaling method.

Table 7 - Evaluation of Scaling Methods and Reasons for Exclusion



Features	ANOVA	Chi-Squared	Random Forest	Lasso	Elastic Net
Age at Injury	849.22	-	**0.03**	**0.91**	**0.51**
Average Weekly Wage	**1484.92**	-	**0.04**	< 0.01	**0.81**
Birth Year	919.54	-	**0.03**	**0.73**	**0.54**
IME-4 Count	**35206.24**	-	**0.04**	**< 0.01**	**1.32**
Number of Dependents	1.17	-	**0.02**	< 0.01	0.01
Days until First Report	366.17	-	**0.03**	< 0.01	**0.16**
Days until Assembly	**1156.89**	-	**0.03**	**0.36**	**0.15**
Assembly Quarter	64.28	-	**0.01**	0.02	0.03
Assembly Date Referenced	580.27	-	**0.03**	0.07	**0.27**
Accident Date Referenced	688.32	-	**0.04**	**< 0.01**	**0.38**
First Report Referenced	373.21	-	**0.03**	0.11	**0.21**
Log Average Weekly Wage	**1486.28**	-	**0.04**	0.16	**0.85**
ce_Carrier Name	810.26	-	**0.03**	0.09	0.08
ce_Carrier Type	504.91	-	**0.01**	0.12	0.06
ce_County of Injury	353.92	-	**0.02**	0.12	0.06
ce_District Name	182.90	-	**0.01**	0.14	**0.18**
ce_Industry Code Description	192.26	-	**0.02**	< 0.01	0.09
ce_Medical Fee Region	213.76	-	**< 0.01**	0.02	0.04
ce_WCIO Cause of Injury Description	502.96	-	**0.03**	0.05	0.08
ce_WCIO Nature of Injury Description	591.97	-	**0.02**	< 0.01	**0.19**
ce_WCIO Part Of Body Description	954.13	-	**0.03**	0.09	**0.17**
ce_Zip Code	98.24	-	**0.03**	0.04	0.05
Average Weekly Wage Zero	-	**< 0.01**	**0.23**	**< 0.01**	**3.41**
Attorney/Representative	-	**< 0.01**	**0.06**	**0.65**	**1.27**
C-2 Date Missing	-	**< 0.01**	< 0.01	**4.11**	**1.29**
WCIO Part Of Body Description Missing	-	**< 0.01**	< 0.01	**0.92**	**0.85**
IME-4 Count Missing	-	**< 0.01**	**0.04**	**< 0.01**	**2.03**
First Hearing Date Missing	-	**< 0.01**	**0.03**	**< 0.01**	**1.59**
C-3 Date Missing	-	**< 0.01**	**0.01**	**0.59**	**0.82**
First Report Submitter	-	**< 0.01**	< 0.01	**1.13**	**0.39**
COVID-19 Indicator	-	**< 0.01**	< 0.01	**< 0.01**	**1.32**
Gender	-	**< 0.01**	< 0.01	**< 0.01**	**0.29**
Industry Code Description Missing	-	**< 0.01**	< 0.01	**2.27**	**0.86**
Alternative Dispute Resolution	-	**< 0.01**	< 0.01	**< 0.01**	**0.75**
Carrier Type Missing	-	**< 0.01**	< 0.01	**< 0.01**	**0.78**
Average Weekly Wage Missing	-	0.69	**0.03**	**3.44**	**1.63**

Table 8 - Feature Selection Results for Claim Injury Type



Features	ANOVA	Chi-Squared	Random Forest	Lasso	Elastic Net
Age at Injury	243.69	-	**0.04**	**< 0.01**	**0.56**
Average Weekly Wage	**1140.59**	-	**0.06**	0.05	**0.97**
Birth Year	**926.57**	-	**0.04**	**< 0.01**	**0.54**
IME-4 Count	**26340.42**	-	**0.04**	**0.93**	**0.49**
Number of Dependents	1.04	-	**0.03**	< 0.01	< 0.01
Days until First Report	**1251.34**	-	**0.04**	**< 0.01**	**0.21**
Days until Assembly	**2968.47**	-	**0.06**	**0.46**	**0.23**
Assembly Quarter	181.98	-	**0.01**	< 0.01	< 0.01
Assembly Date Referenced	**2416.86**	-	**0.05**	**0.41**	**0.20**
Accident Date Referenced	**6244.43**	-	**0.07**	**< 0.01**	**0.21**
First Report Referenced	**3157.73**	-	**0.06**	**< 0.01**	**0.20**
Log Average Weekly Wage	**1139.63**	-	**0.06**	**< 0.01**	**1.19**
ce_Carrier Name	13.85	-	**0.04**	**0.10**	**0.05**
ce_Carrier Type	**1254.57**	-	**0.03**	**0.39**	**0.20**
ce_County of Injury	716.86	-	**0.03**	< 0.01	< 0.01
ce_District Name	500.90	-	**0.02**	0.06	0.03
ce_Industry Code Description	**842.22**	-	**0.03**	**< 0.01**	**0.06**
ce_Medical Fee Region	497.51	-	**0.01**	< 0.01	0.02
ce_WCIO Cause of Injury Description	30.17	-	**0.04**	0.05	0.02
ce_WCIO Nature of Injury Description	8.20	-	**0.03**	0.01	< 0.01
ce_WCIO Part Of Body Description	114.03	-	**0.04**	0.01	< 0.01
ce_Zip Code	15.82	-	**0.05**	< 0.01	0.04
Attorney/Representative	-	**< 0.01**	**0.04**	**3.18**	**1.59**
First Hearing Date Missing	-	**< 0.01**	**0.02**	**< 0.01**	**0.37**
IME-4 Count Missing	-	**< 0.01**	**0.02**	**1.35**	**0.73**
C-3 Date Missing	-	**< 0.01**	**0.01**	< 0.01	0.01
Average Weekly Wage Zero	-	**< 0.01**	**0.01**	**< 0.01**	**0.45**
First Report Submitter	-	**< 0.01**	**< 0.01**	**< 0.01**	**0.10**
COVID-19 Indicator	-	**< 0.01**	< 0.01	**< 0.01**	**0.13**
Gender	-	**< 0.01**	**< 0.01**	**< 0.01**	**0.07**
C-2 Date Missing	-	**< 0.01**	< 0.01	**< 0.01**	**1.06**
WCIO Part Of Body Description Missing	-	**< 0.01**	< 0.01	**< 0.01**	**0.08**
Alternative Dispute Resolution	-	**< 0.01**	< 0.01	**< 0.01**	**0.53**
Carrier Type Missing	-	0.67	< 0.01	**< 0.01**	**0.50**
Average Weekly Wage Missing	-	0.67	**< 0.01**	**0.31**	**0.16**
Industry Code Description Missing	-	0.72	< 0.01	**< 0.01**	**0.43**

Table 9 – Feature Selection Results for Agreement Reached

Models	Best Parameters (1)	Best Parameters (2)
Decision Tree	class_weight=None, criterion='gini', max_depth=10, max_features=None, min_samples_leaf=1, min_samples_split=42, splitter='random'	class_weight=None, criterion='entropy', max_depth=25, max_features=None, min_samples_leaf=9, min_samples_split=42, splitter='random'
Logistic Regression	C=2782.56	C=4.6416
Random Forest	bootstrap=False, class_weight=None, max_depth=10, max_features=None, min_samples_leaf=6, min_samples_split=6, n_estimators=100	
XGBoost	colsample_bytree=0.8, gamma=0.45, learning_rate=0.1, max_depth=3, min_child_weight=3, n_estimators=250, reg_alpha=0.2, reg_lambda=1.04, scale_pos_weight=27, subsample=0.4	
LightGBM	colsample_bytree=1.0, learning_rate=0.2336, max_depth=9, min_child_samples=15, n_estimators=400, num_leaves=80, reg_alpha=27.83, reg_lambda=2.15, scale_pos_weight=4.0, subsample=0.6	
HGBoost	early_stopping=False, l2_regularization=0.0, learning_rate=0.05, max_depth=3, max_iter=200, max_leaf_nodes=31, min_samples_leaf=100, scoring='accuracy'	

Table 10 - Optimal Hyperparameters for predicting Claim Injury Type using Cross-Validation; (1) No Resampling; (2) Tomek Links Under sampling

Models	Best Parameters (1)	Best Parameters (2)
Logistic Regression	solver: lbfgs, penalty: l2, max_iter: 500, C: 100	{solver: liblinear, penalty: l1, max_iter: 100, C: 1}
Random Forest	{n_estimators: 10, min_samples_split: 2, min_samples_leaf: 1, max_depth: 3, bootstrap: True} {n_estimators: 10, min_samples_split: 2, min_samples_leaf: 1, max_depth: 3, bootstrap: True}	
XGBoost	{subsample: 0.6, reg_lambda: 1.5, reg_alpha: 0.1, n_estimators: 20, min_child_weight: 1, max_depth: 4, learning_rate: 0.2, gamma: 0.2, colsample_bytree: 1.0}	
Perceptron	{penalty: l1, max_iter: 2000, fit_intercept: True, eta0: 1, alpha: 0.0001}	{penalty: l2, max_iter: 2000, fit_intercept: False, eta0: 0.01, alpha: 0.001}
Multilayer Perceptron	{solver: adam, max_iter: 200, learning_rate: constant, hidden_layer_sizes: (50, 50, 50), alpha: 0.0001, activation: logistic}	{solver: adam, max_iter: 200, learning_rate: constant, hidden_layer_sizes: (100,), alpha: 0.0001, activation: relu}

Table 11 – Optimal Hyperparameters for predicting Agreement Reached using Cross-Validation; (1) No Resampling; (2) Tomek Links Under sampling

Base Estimators **(OneVsRestClassifier)**	**Train F-1** **Score**	**Val F1** **Score**
Logistic Regression	0.332	0.357
XGBoost (n_estimators=10)	0.310	0.306
RidgeClassifier	0.288	0.304
DecisionTree (max_depth=3)	0.251	0.253
LightGBM (n_estimators=10, max_depth=3)	0.254	0.252
Perceptron	0.262	0.238

Table 12 – Model Results for OneVsRestClassifier for each Base Estimator

Approach		**1. CANCELLED**	**2. NON-COMP**	**3. MED ONLY**	**4. TEMP**	**5. PPD SCH LOSS**	**6. PPD NSL**	**7. PTD**	**8. DEATH**	**MACRO F1 SCORE**
OVR LR	**score**	0.52	0.90	0.02	0.76	0.48	0.00	0.00	0.18	0.36
WEIGHTED OVR LR	**weights**	2.0	0.8	3.0	0.95	1.25	7.5	0	1.5	-
	score	0.54	0.89	0.23	0.71	0.51	0.12	0.00	0.26	0.41

Table 13 - Model results in validation set comparing One Vs Rest Strategy and Weighted One vs Rest Strategy using Hold-Out Method

ANNEX C: DEFINITIONS

1. Tomek Links

Tomek Links is a method used to reduce data by identifying pairs of samples (x_i, x_j) from different classes that are close to each other in feature space. Two samples form a Tomek Link if the following conditions are satisfied:

i. x_i and $x_j$belong to different classes;

ii. The distance (typically Euclidean distance) between x_i and x_j is minimal, meaning that there is no other sample x_k for which $d(x_i, x_k) < d(x_i, x_j)$ or $d(x_j, x_k) < d(x_i, x_j)$.

By removing one or both samples (usually the majority class sample, since we are trying to improve the representation of the minority classes), the dataset becomes less noisy, which improves model performance and interpretability. (Elhassan and Adam, E. A. 2016)

2. XGBoost (Extreme Gradient Boosting)

XGBoost is an ensemble technique based on Gradient Boosting, introduced by Chen and Guestrin (Chen 2016), that builds models sequentially, where each model attempts to correct the errors of the previous one, typically using decision tree as the base models.

The training process minimizes a regularized objective function, defined as:

$$\mathcal{L}(f) = \sum_{i=1}^{n} l(y_i, \widehat{y_i}) + \sum_{k=1}^{K} \Omega(f_k) \qquad (1)$$

with $l(y_i, \widehat{y_i})$ as the loss function that measures the difference between the observed value y_i and the predicted $\widehat{y_i}$. As a result, XGBoost is trained to predict the residuals of the combined ensemble.

$\Omega(f_k)$ is the regularization term for the complexity of the tree k, defined as:

$$\Omega(f_k) = \gamma T + \frac{1}{2}\lambda \sum_{j=1}^{T} w_j^2 \qquad (2)$$

where T is the number of leaves and w_j are the leaf weights. Parameters γ and λ control the number of leaves and leaf weights, respectively. For this reason, it prevents overfitting because incorporates regularization terms that control the penalty for tree complexity.

3. LightGBM (Light Gradient Boosting Machine)

LightGBM, developed by Microsoft (Ke et al., 2017), is a scalable gradient boosting optimized for speed and performance by minimizing the regularized objective function (1). However, the particularity is that this model prioritizes informative samples by focusing on data points with large gradient magnitudes (Gradient Based One Side Sampling). The gradient g_i is defined as:

$$g_i = \frac{\delta l(y_i, \widehat{y_i})}{\delta \widehat{y_i}} \tag{3}$$

This prioritization works by retaining a proportion a of data points with the largest gradient magnitudes (more informative) and a randomly sampled subset of the remaining points with the proportion b. To maintain the balance of the loss function, the gradients of the randomly sampled points are scaled up by a factor. This approach ensures that the most informative samples are prioritized during training without significant loss of information.

Additionally, LightGBM reduces dimensionality by grouping mutually exclusive features (Exclusive Feature Bundling) by finding a partition $P = \{P_1, P_2, \dots, P_k\}$ such that the sum of the overlap between P_i and P_j, with $i \neq j$, is approximately 0.

The model employs a growth strategy that allows, at each iteration, the selection of the leaf with the highest loss reduction for splitting. (Guolin Ke 2017)

References

Chen, T., & Guestrin, C. "XGBoost: A Scalable Tree Boosting System." *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining.* ACM, 2016. 785–794.

Cloeren, M., & al. "Defining documentation requirements for coding quality care in workers' compensation." *Journal of Occupational and Environmental Medicine*, 2016: 1270-1275.

Elhassan, T. A. M.,, and Adam, E. A. "Classification of imbalance data using Tomek Link (T-Link) combined with random under-sampling (RUS) as a data reduction method." *Journal of Informatics and Data Mining.*, 2016.

Fair Labor Standards Act (FLSA) – Child Labor Provisions. n.d. https://www.dol.gov/general/topic/hiring/workersunder18.

Fernandes, F. T. *Machine learning em saúde e segurança do trabalhador: perspectivas, desafios e aplicações.* University of São Paulo Digital Library of Theses and Dissertations., 2022.

Guolin Ke, Qi Meng, Thomas Finley, Taifeng Wang, Wei Chen, Weidong Ma, Qiwei Ye, Tie-Yan Liu. "LightGBM: A Highly Efficient Gradient Boosting Decision Tree." *Proceedings of the 31st Conference on Neural Information Processing Systems (NIPS 2017).* Curran Associates, Inc., 2017. 3149–3157.

Henriques, Roberto. *Ensemble Learning for Machine Learning: Combining Multiple Models for Improved Prediction.* Nova IMS, 2023.

Meyers, A. "Applying Machine Learning to Workers' Compensation Data to Identify Industry-Specific Ergonomic and Safety Prevention Priorities Ohio, 2001 to 2011." *Journal of Occupational and Environmental Medicine.*, no. 1 (2018): 55-73.

Retirement Benefits. Social Security Administration, 2024.

Trevor Hastie, Robert Tibshirani, Jerome Friedman. *The Elements of Statistical Learning: Data Mining, Inference, and Prediction.* Springer, 2017.

Zanke, P. "Exploring the Role of AI and ML in Workers' Compensation Risk Management." *Human-Computer Interaction Perspectives* 2, no. 1 (2022): 24-44.