Machine Learning Project

Master in Data Science and Advanced Analytics

**NOVA Information Management School**

Universidade Nova de Lisboa

# WCB Decision - Machine Learning Project
# Exploratory Data Analysis

## Group 08

Afonso Gamito, 20240752

João Rodrigues, 20241037

Rute D'Alva Teixeira, 20240667

Samuel Mendes, 20240751

Tomás Oliveira, 20211576

Fall/Spring Semester 2024-2025

# Abstract

The New York Workers' Compensation Board (WCB) is the entity responsible for assembling and deciding on claims whenever there is a workplace injury.

Given our target variable's multi-class distribution, we trained a model to predict if a worker's injury claim would be of type 1, 2, 3, 4, 5, 6, 7 or 8 – with 1 being non compensable, the least severe and 8 Death, the most severe.

WCB'S dataset was submitted to a scrutinized cleaning, treating all missing values, all detected inconsistencies and all outliers considered errors.

Categorical data was encoded with frequency/count encoder, and target variable was label encoded, removing the string part of each unique category. All numerical data was scaled with a normalization method including binary variables, treated as numerical in our approach.

Our Feature Selection process combines filter, wrapper and embedded methods, with a final approach including Variance Statistical Test (26 selected features), Kendall's Correlation between input variables and against target (31 and 12 features selected, respectively), ANOVA (15 features selected), LASSO (28 features selected), Random Forest Classifier (17 features selected) and Decision Trees (18 features selected).

To elect our final predictor, we tested our validation dataset with a total of 10 distinct models, assessing the best performance considering F1-Score's macro average comparison metric, referenced as f1 from here on, and also attending to the test dataset's performance considering kaggle's competition score, referenced as kg from here on. From this assessment, the results yielded for each model are:

LightGBM (f1 0.4599, kg 0.41025) our best performed model, followed by XGBoost (f1 0.4556, kg 0.39258), next, Random Forest (f1 0.4095, kg 0.32603), Decision Trees (f1 0.3484, kg0.24308), Neural Network (f10.3208, kg 0.24528) with parameters suggested by professor Ricardo in lab classes, Logistic Regression (f1 0.2750, kg 0.19487) and Naive Bayes (f1 0.2069, kg 0.16755). All models are tested with aimple intuitive parameter settings.

For an optimized implementation, we introduced a Grid Search (GS) approach, yielding the following results: GS Random Forest (f1 0.4325, kg 0.34096) model used for our interface simulation, GS LightGBM (f1 0.4524, kg 0. 41025), GS XGBoost (f1 0.4592, kg 0.40752).

On a final note, our models performed under expected, even though we followed a solid and thorough data preprocessing, a methodical and transparent feature selection workflow and a consistent deployment of the assessed models.

## Contents

## List of Tables

## List of Figures

# I. Introduction

This project is based on a dataset gathered by The New York Workers' Compensation Board (WCB), an entity responsible for assembling and deciding on claims whenever there is a workplace injury.

We were asked to develop a machine learning model that accurately predicts the claim injury type, to assist in decision-making for incoming claims. This report documents our workflow, from data pre-processing to a final prediction.

The Data Exploration and Pre-Processing section introduces the dataset, highlighting incoherencies, missing values and outliers' treatment, described over the following two pages. Next, we explain the process of Feature Engineering decisions (first two paragraphs) and the rationale that led to decision-making of Encoding (six paragraphs) and Scaling (subsequent section).

In Section III, we present our implementation of resampling techniques (first three paragraphs) and explore in depth our Feature Selection procedure, justifying each method's application, considering their underlying assumptions and overall data structure. Finally, we discuss Model Assessment and our ultimate predictive model.

Additionally, our open-ended section refers to a user-friendly interface developed to predict a claim injury type, based on each user-specified input. This tool can be integrated into WCB's official webpage as a simulation for workers filing claims, or as a resource to streamline WCB's internal operations.

# II. Data Exploration and Preprocessing

## Incoherencies, Unusual and Missing Values

The project's dataset consists of detailed information regarding the claims data received by WCB administers, from the start of 2020 till the end of 2022.

This multivariate dataset includes very important features such as as Claim Dates (e.g., *Accident Date*), Worker Demographics (e.g., *Gender*), Claim and Case Information (e.g., *Claim Identifier*), Location and Region (e.g., *Medical Fee Region*), Incident and Injury Details (e.g., *COVID-19 Indicator*), Industry Classification (e.g., *Industry Code*), Injury Descriptions and Codes (e.g., *WCIO Cause of Injury Code*) and Claim Outcomes (e.g., *Claim Injury Type*),. Accident Date, Assembly Date, Age at Injury, Gender and some others. Check section A (Table 1) in the annexes for more details regarding the initial variables. However, our dataset has important modifications: first, it consists of training and test sets with 593471 entries and 33 variables (including the target) and 387975 entries and 30 variables (which make sense because here we don´t have the 3 target variables: Claim Injury Type, Agreement Reached and WCB Decision), respectively.

At the beginning of our analysis, we uploaded our dataset to the notebook to perform the subsequent data preprocessing on the training and test sets simultaneously. Therefore, we randomly split our training set in a stratified way, which retained 70% of the data while the rest became our validation set (30%). Such stratification enabled the project dataset class distribution to be maintained during the split.

Before applying any preprocessing to the variables, it is imperative to begin by exploring the characteristics of our data, as this allows us to deeply understand the topic and how each variable should be treated, what inconsistencies to address, how missing values and outliers should be managed, among many other factors.

It is worth emphasizing that all these concerns have been detailed with explanations of how we solved them in section B. We began by making a superficial exploration to gain insights about the datatypes, missing values, and possible incoherencies in our data to advance to a visual analysis of our variables to identify patterns, distributions, and frequencies. Furthermore, we checked for duplicate values in our data set, finding none likely due to them being removed doing missing values treatment.

We immediately noticed some variables were with wrong data types, so we changed a few (section B, table 2), something we knew we had to solve before starting modeling.

Then, we noticed some missing values (section B, table 3) when we were analyzing the number of observations in each variable, a concern that should also be addressed before the modeling phase. Given this, we decided to drop two variables (*C-3 Date* and *First Hearing Date*). Although, at this step, we know that *Claim Indentifier* would make perfect sense as index variable because it has all its observations as unique values, we decided not to include it as index because it would present a problem when we would be applying our model.

Regarding incoherences, we discovered several discrepancies in the dataset during the data exploration phase that, if ignored, may have seriously compromised the validity of our research. Date variables, for example, included erroneous values such Birth Year = 0 and incidents that happened after the assembly date. Furthermore, some people had an Average Weekly Wage of $0, which is unlikely for employed people, while others were listed as being under the legal working age. We produced a thorough table outlining each of these discrepancies and the strategies used to address or resolve each problem in order to guarantee data quality and accuracy. To avoid distorted data and erroneous conclusions in the subsequent study, it was imperative to address these differences.

## Data Preprocessing: Data Exploration and Outliers

We proceeded with data exploration in the training set by counting the initial variables' values frequency and plotting histograms and boxplots. These techniques allowed us to capture and understand the dataset's inherent characteristics, including high distributed values and outliers.

We treated outliers for this variables: IME-4 Count (based on our research were we couldn´t determine the exact number of IME-4 forms that can be attached to a claim, so we consider 72, this dataset maximum number, to be an unrealistic value that may negatively affect our model, for that reason, we decided to adjust these values by establishing a limit of 8, grounded in reasonable logic and in the visualization in Figure 1, where 8 seems to be the last value not considered an outlier and since in this case we only observe outliers in the upper limit, we think the best approach is to replace all values higher than 8 with the median, which is 2); Age ate Injury (we saw that ages between 0-15 represent 1% of the data and are highly unlikely for employment or work-related injuries and ages 75 and more represent 0,5% and, while plausible for injuries, are unlikely for employment, so for that reason, we excluded the extreme 1% on each end to define reasonable minimum and maximum values); Birth Year (we replace 0 values with the subtraction between 'Accident Date' year and 'Age at Injury); Average Weekly Wage (after reviewing the handout description, we note that this dataset includes claims from volunteer workers in specified areas, therefore, in some cases, an Average Weekly Wage of 0 is an acceptable value, so combining this insight with domain knowledge, we conclude that the industries where the specified volunteer work occurs are identified by Codes 56, 92, and 62, with that said, we assume that the Average Weekly Wage values for these industries are accurate and valid, even if 0 and we treat all the others by imputing the previously calculated Average Weekly Wage by Industry Code). To understand better our analyses, check the figures 1-4 in section D.

## Data Preprocessing: Feature Engineering, Data Encoding and Scaling

In order to do feature engineering, we eliminated variables that had too many missing values, as well as were not reliable or were irrelevant. Depending on the nature of the variable, various aspects were also changed, such as dividing variables which represent a date into components like month and year or transforming them into binary indicators. We also created several new variables, as we believed there was additional valuable information to be extracted from the existing ones, which could be very useful in further analysis. The table 6 (present in the section B) provides a thorough description of the new variables and detailed transformations of the provided ones (either if it´s a new created variable or a transformed/modified variable from the existing ones).

Then, we decided to drop some variables: *WCB Decision* and *Agreement Reached* because these features give us information that are unknown at the start of a claim, according to our Handout Descriptive and for that reason, it is useless to train our model with information that we shouldn't have available for future predictions; *Industry Code Description, WCIO Cause of Injury Description, WCIO Nature of Injury Description* and *WCIO Part Of Body Description* because we already have the corresponding code for each of this features, which makes keeping both unnecessary, we let us choosing to drop Description and keep Code, for the fact that it is easier to work with numbers; *Number of Dependents* because the variable is unreliable (is uniformly distributed, all values have identical frequencies and, in some cases, the number of dependents doesn't align with the workers' age logically).

Our dataset is composed, in great part, by categorical features. This aspect alone adds a lot of complexity to the work in our hands, since the handling of categorical data requires a very sensitive approach. We consider the encoding process one of the factors that most contributed to the performance of our model, since most machine-learning algorithms require the conversion of textual data into numeric format, to improve responsiveness. With that being said, we considered the general interpretability assumptions our model does to orient our decision-making in encoding.

We carefully studied and considered the use of Dummy variables/One Hot encoding, Label/Ordinal Encoding, Target Encoding/Weighted mean target Encoding and Count Encoding.

In parallel with understanding how our model interprets our categorical data, to evaluate the best encoding technique, we must understand its characteristics:

- **Ordinality**, our categorical data doesn't have an inherent order. For that reason, categorical data represented as a number still needs to be properly encoded, since most models interpret there is ordinality in numbers. This factor alone excludes label/ordinal encoding as a viable option;

- **Unique values count,** a high count of unique categories immediately excludes the use of One-Hot Encoding for such features, since it would be converting every unique category (minus one) into a new feature. This decision avoids introducing multicollinearity and high computational costs, due to a larger dataset;
- **Finite and/or fixed number of categories seen on training data,** it's important to assess if unique categories have a limited and predetermined number, to ensure that in future predictions the model is properly trained to handle all possible values of categorical data. We verified that majority of our data has a fixed number, seen on training data. Dummy variables would make sense in these cases otherwise, we wouldn't be training our model with enough information to make this encoding meaningful.

These factors, although significant, must be assessed in combination for a complete evaluation.

After this analysis, we favoured the use of Count Encoding for all categorical data, since it works in data with both high and low unique value counts; it introduces ordinal meaning to the number representation – higher numbers mean more frequent values; has a simple and intuitive coding application and is a technique that can be applied in all categorical features at once. Upon this decision, we validated its assumptions: after this encoding, categorical data becomes numerical and must be treated as such; and unseen data on future datasets will be represented as 0.

Data scaling is applied to numerical data, to ensure that the features with different distributions and scales are transformed into a comparable range. This is an important preparation before feature selection, enabling the use of thresholds to select features. On the other hand, it is performed after encoding, to include the numerically encoded data- this happens for conformity of the process, although we recognize that encoded data cannot be meaningfully scaled. Finally, it is a necessary step to improve model performance and convergence.

For our dataset, we considered the scaling methods of normalization, standardization and Robust scaler's Interquartile Range method.

After analyzing each method's assumptions, we chose to apply the normalization method, since it doesn't assume our data is normally distributed, and being that outlier's treatment has been ensured before, in the pre-processing stage.

Regarding the scaler´s range, we started with the default one. After scaling, we validated the results in each feature, focusing on how close each maximum value is to 1. From this analysis, we found that most of our features besides binary ones ae represented very close to 1, with up to 7 decimal places precision. After reflecting on these findings, we considered that our data could benefit from a wider scale range, for a more expressive distribution of between each distinct value.

However, after testing ranges from 0 to 5, 10, 100 and 1000, keeping all other factors fixed, there was no improvement in the model's F1 macro average score, in fact, the score was penalized by 0.01 equally in all cases. For this reason, we kept the scaler's default range of 0 to 1.

## III. Multiclass Classification

### Hybrid-sampling and Feature Selection

To address the problem of class imbalanced that we are facing, we must adopt an over-sampling and under-sampling technique before proceeding to feature selection, that is, an hybrid-sampling. In one hand, we will use an over-sampling technique which improves minority class detection and enable us to keep all observations while addressing imbalance, on the other hand, we will use an under-sampling technique which provide us faster training, reduce noise and focuses on core information. Thus, we chose the Synthetic Minority Over-sampling Technique (SMOTE) to apply the over-sampling technique for reasons described above.

The implementation of this techniques combined balances dataset efficiently, combined strengths of both methods and mitigates overfitting risk, given us a much more balanced dataset, converting our data from: (2: 203754, 4: 103955, 3: 48234, 5: 33796, 1: 8734, 6: 2948, 8: 329, 7: 68) to (2: 100000, 4: 50000, 3: 48234, 5: 33796, 1: 20000, 6: 20000, 7: 10000 e 8: 10000), reducing our overall number of data in approximately 27%. This implementation is performed after encoding and before scaling, since this implementation requires all data to be numerical. For this reason, binary variables are treated as numerical.

**For feature selection**, we will deep-dive into the decision process that led to each feature selection method included, and how such implementation took place. To begin, we should first distinguish the methods used into four groups: Filter Methods, Embedded Methods, Wrapper Methods and Standalone methods.

**Regarding the Filter Methods**, our initial approach involves using statistical tests on our data to rank features by relevancy, according to each test's criteria. To ensure the appropriateness of these tests for our dataset, we refer to a diagram from Machine Learning's class materials, which provides the theoretical foundation for this process (check figure 5, section C).

As we can see, through that figure, it is crucial to understand our dataset characteristics, in this case, the nature of our input and output variables, in order to assess which methods better align with this fact. From observing the diagram, we can also point to an important tradeoff: if our output variable is categorical, it introduces Chi-Squared and Mutual Information methods for categorical data, and if it is numerical, it introduces Pearson and Spearman's Correlation indices for numerical data. This insight is particularly valuable for our dataset, because our output variable can be interpreted as either numerical or categorical. This fact is due to our target variable's ordinality: categories show an ascending level of severity in injury. The original representation of the target's categories is a mix of strings and numbers. However, with the appropriate encoding, they can easily be converted into numerical, without compromising the model's interpretability. To properly justify the decision made about the target, we must first introduce a new factor into the discussion: the underlying assumptions of each Filter method.

Now, we will test each filter method based on its underlying assumptions. For that, we have a deeper understanding of which statistical test can be performed based on categorical or numerical data, but its application still depends heavily on each test's assumptions and whether our data complies or not.

We will explore this in detail:

Variance: This filter method tests how much our data points deviate from the mean, with no known assumptions that could discard this method. There were no univariate features, so now features were discarded in place, and because our data was properly scaled, we applied a threshold to discard low variance features. **We selected 26 features from Variance's Statistical test**.

Pearson's Correlation: This correlation is immediately excluded from our range of options, because it assumes that our data is normally distributed, which is not the case.

Spearman and Kendall's Tau Correlation: Neither of these methods require Gaussian-like data distribution, making them a viable option. However, they do require that the data is either continuous or ordinal.

For that reason, we must validate that all our numerical features – including numerically encoded – comply with this assumption. It isn't the case for binary variables, so to perform this analysis, we must use a subset of data. Ultimately, Spearman's correlation could have been used, but it gave the same information as Kendall's correlation. For that reason, we chose to keep only Kendall's to limit **our final selection, aiming for a maximum of 17 features**.

This correlation can be assessed:

- Between input variables, to identify pairs who are highly correlated, since high correlation indicates redundancy, meaning that both features provide similar information to the model. In such cases, one of the correlated features can be discarded. This helps to reduce multicollinearity and improve the model's interpretability and efficiency. **We selected 31 features from Kendall's correlation**.
- Between input and the output variable, to identify which features have stronger predictive power. Low correlation may indicate that a feature is potentially unimportant for our model- in such cases, the feature can be discarded. This helps to improve the model's interpretability and efficiency, by reducing the noise. However, we don't want to remove solely based on this correlation, since we could be overlooking nonlinear relationships. **We selected 12 features from 'Target Kendall's' correlation.**

ANOVA: This method is used, since it is applicable for both categorical and numerical input variables, and because the target variable's nature also complies with this method. This method doesn't assume data is normally distributed, making it a sound method for feature selection on this dataset.

The optimal number of k was adjusted through trial and error, based on how many features we wanted to include in our model. **ANOVA selected 15 features.**

Chi Square: This method tests independence of categorical data against the target variable, with no known assumptions discarding this approach. The code for this method is obtained from lab 3 materials about Feature Selection. Although in our initial approach this method was tested, in our final approach we excluded this method from our Feature Selection process.

Mutual Information: More complex approach on dependence between variables, commonly explored in Pearson's correlation. It tells us on average, how much knowing one variable reduces the uncertainty (entropy) of another variable. This method handles better discrete variables, and it is possible to test mutual information between input variables and between input variables and target. During our research, no further assumptions on this method were found. Although in our initial approach this method was tested, in our final approach we excluded this method from our Feature Selection process.

Our initial approach was to outline the workflow during Feature Selection, we decided to interpret both the target variable and our binary features as categorical, favoring the tradeoff of Chi-Squared and Mutual Information over Pearson and Spearman methods. This decision was very intuitive, since Pearson's assumptions rule out this method, and Spearman would not be applicable on binary features, since they are not ordinal or continuous. Following this approach, we would be able to apply Chi-Squared and Mutual Information on our categorical data – binary outcome features, and numerical data would be exclusively continuous or ordinal, making all other filter methods viable.

However, after deploying our model and facing its performance score, our final approach was to identify the need to introduce resampling techniques, thus reviewing our entire Feature Selection process. For that reason, as explained before, binary features are treated as numerical - there is no categorical data. This decision removes Chi Squared and Mutual Information from our process.

Thresholds: Our decision for the threshold used was a textbook approach, considering 0.01 for Lasso and Variance. However, to test variance, we considered this threshold to not be penalizing enough, which led to us raising it to 0.05. For Kendall's redundancy tests, a textbook approach of 0.8 was applied and for relevancy tests, 0.15.

**Regarding the Wrapped Methods**, we will discuss methods that involve training a model and selecting features based on how the model performs – these methods consider the interaction between features and the chosen algorithm. The wrapper methods we selected are applicable on both numerical and categorical data. For that reason, in this section we use the joined dataset, X_train_processed.

RFE: We apply a Recursive Feature Elimination, that will recursively fit a model and eliminate, one by one, the least important features based on their performance score, until it reaches the optimal number defined, or until the model stops improving. A drawback of this method is that it could introduce overfitting.

RFE - Logistic Regression: For this method, we start by using Logistic Regression as a base model estimator, since it performs well for multi-classification tasks. To assess the optimal number of features, our *n_features_to_select* parameter, we run a code tested in class materials, from a range of 1 to 18. We started with 10 and raised the limit to 18 from trial and error, after noticing our optimal number was always the maximum. From the output of this code, we set an optimal number of **16 variables**.

**RFE initial and final approach -** Initially, we selected 16 features from RFE's method on each predictor, Logistic Regression and SVM classifier. However, after deployment of model we analysed the classification report with and without this method and observed the following:

- Excluding RFE's methods from selection: We accurately predict values of all classes, with an F1 macro average score of 0.46.
- Including RFE's methods from selection: We don't make prediction on class 7, a minority class, and the F1 macro average score lowers to 0.41.

RFE - Support Vector Machines: Still in RFE, we use as a different base estimator Support Vector

Machines, since it also performs well for multi-classification tasks. However, because this vector-based model considers all records of each datapoint and all its restraints, it is highly costly in computation. This limited our analysis, and an optimal number of features wasn't assessed. For that reason, we reference the same number as LR's optimal.

**Regarding the Embedded Methods,** this set of methods performs feature selection as part of the training process of the model and learns which features best contribute to the accuracy of the model while it is being created.

Lasso Regression: Lasso, or L1 Regularization, applies to a regression model and is a suitable feature selection method for our dataset, as it helps in both regularization and feature selection. Since our input variables are numerical, Lasso can perform effectively, allowing for automatic feature selection while improving model performance, by reducing overfitting. We selected 28 features from Lasso's cost function.

Random Forest Classifier: Method chosen, since we will also deploy this model, and it measures the level of impurity (gini and entropy criteria) that each feature is responsible for reducing to the model. We selected 17 features from Random Forest Classifier.

Regarding the Standalone Methods, we include Decision Trees, which is a supervised learning algorithm, just like Random Forest, selects features by measuring the level of impurity that each one reduced in the model. We selected 18 features from Decision Trees.

Lastly, in total, we tested 7 different methods in our entire dataset, **with a final selection of 16 features**. We integrated all feature selection methods tested for each feature into a voting system, short-listing our final selection with only the features that were selected on majority of methods. This majority is calculated by ruling that the proportion of '*keep*' must be higher than the proportion of '*discard*', by at least 2 methods. For example, *IME-4 Count* is selected by 4 methods and discarded by 3, so it won't be included in our final selection. To visualize our voting system in detail, please check table 7, section D.

## Model Assessment

To determine which model performed the best for our assignment, we implemented and assessed several of them. Random Forest, Logistic Regression, Neural Networks, Naive Bayes, DT, LightGBM, and XGBoost are among the models that have been put to the test (each explained in Table 8). Additionally, we used a Voting Classifier that integrated XGBoost, Random Forest, and Logistic Regression in effort to enhance model performance. The Voting Classifier achieved a strong score, demonstrating that combining these models provided a notable performance boost.

For evaluation purposes, we prioritized metrics such as F1 Score, Recall, and Precision, as accuracy would not provide a meaningful evaluation for this particular problem, given the class imbalances. These metrics offered a better view of model performance, especially in identifying how well each model handles different classes.

As we can observe in Table 10 (Section D), Random Forest, XGBoost, and LightGBM were the models that performed the best out of all the ones we evaluated. In order to improve these three algorithms' performance even further, we focused on optimizing them. We used a combination of Grid Search and Cross Validation to accomplish this, and for this procedure, we intentionally employed 10% of the dataset, ensuring the class distribution was maintained. This approach helped us extract the optimal hyperparameters efficiently while maintaining computational feasibility.

For the final model training, we applied the ideal parameters to the complete dataset and chose the best configuration. The focus on Random Forest, XGBoost, and LightGBM, along with the optimization process, significantly improved their scores in terms of precision, recall, and F1 score, confirming them as the top-performing models for this task.

## IV. Open-Ended Section

The first part of our application focuses on a prediction model that allows users to input data about a workplace injury and receive a prediction based on those inputs. This model is particularly useful for organizations such as the Workers' Compensation Board (WCB) to predict the most common injury cases across the general population, providing valuable insights into prevalent trends and risk factors. Additionally, companies and industries can analyze injury cases within their specific context to make informed decisions that reduce the number of injuries, minimize the severity of incidents and implement proactive safety measures. Acting as a decision-support tool, the predictive model helps stakeholders identify potential risks and take preventive actions to improve workplace safety.

The second part of our project is an interactive map that visualizes regional data related to workplace injuries. Divided into four regions, the map displays key information, including demographic details about the affected individuals and insights into the industries most frequently involved in injuries. This geographic perspective can influence decision-making at various levels, such as empowering local authorities like municipal councils and mayors to implement targeted measures and policies for the most affected areas. By presenting data visually, the map enhances understanding, helps identify regions requiring immediate attention and supports better resource allocation to improve workplace safety. Is important ro refer that the model use to create this interface was the Random forest because, even though is wasn't our best model in terms of performance, it was the the best performer model from the model that we learn in class. For that reason we decided to apply Random Forest instead of XGBoost.

## V. Conclusion

In order to provide an automated and trustworthy decision-support tool, the goal of this project was to create a model that would forecast the New York Workers' Compensation Board's (WCB) ruling on workplace injury claims. We used an extensive procedure that included data preprocessing, feature engineering, and a careful evaluation of multiple categorization methods.

We tackled important issues during data preprocessing, including imputing missing data without distorting the dataset's integrity and respecting Data Leakage, fixing incoherent items, where he had to give a special effort in order to ensure that every incoherence was founded and properly dealt with, and reducing class imbalance using hybrid sampling strategies to guarantee that minority classes were adequately represented, which was fundamental to the better performance of the model. Choosing the most pertinent variables for feature selection was difficult, and we had to balance techniques which were not easy to set and define the corresponding thresholds to make sure the finished model caught significant correlations without adding redundancy or overfitting. Important choices were also impacted by computational limitations, such as restricting the amount of cross-validation used during model testing and giving priority to randomized over grid search for hyperparameter tweaking.

Feature selection was another critical step, requiring the application of multiple methods, including ANOVA and Kendall's correlation, to ensure the most valuable variables were retained and variables such as Industry Group were discarded due to the low correlation with the target. This process involved fine-tuning thresholds and making decisions that balanced model complexity with interpretability.

Further in the model implementation, a few algorithms were tested, whereas XGBoost, LightGBM and Random Forest had the best F1 Score, Recall and Precision – metrics chosen to evaluate the model. The other algorithms applied did not perform as well as we thought, potentially due to suboptimal hyperparameter selection or intrinsic characteristics of the dataset, leading us to focus our efforts on refining and optimizing the best-performing models. After applying a Grid Search, the results dind´t increase, so, for that reason, our best model was the LightGBM without Grid Search.

Our open-ended section showed how our model may be used in real-world scenarios, like forecasting injury trends and displaying regional data on an interactive map, an innovative way to compare data visually, providing insightful information about trends in workplace injuries and exposing regional differences. In addition to improving local decision-making, the next step could be implementing the model to the entire country, to be able to find the disparities among the different regions.

In summary, our project effectively produced reliable and understandable models that struck a balance between practical applicability and predictive performance. We made sure that every choice, from feature selection to model deployment, was well-founded and repeated by following open procedures and strict evaluation standards.

## Bibliography

Domain Knowledge and context research:
1. https://www.cordovanolaw.com/practice-areas/workers-compensation/workers-compensation-process/
2. https://www.nycbar.org/get-legal-help/article/workers-comp/workers-compensation-process%E2%80%AF/
3. https://www.wcb.ny.gov/content/main/TheBoard/glossary.jsp

Medical Fee Region and Zip Code relation:
https://www.qrconcepts.com/MediaWiki/images/e/e8/2018_Official_New_York_State_Workers'_Compensation_Medical_Fee_Schedule.pdf

Industry Code for Average Weekly Wage outlier treatment:
https://www.naics.com/six-digit-naics/?v=2017&code=92

Forms details:
https://www.wcb.ny.gov/content/main/forms/Forms_INSURER.jsp

Understanding feature selection methods:

Correlation:
https://medium.com/@aastha.code/interpreting-correlation-matrix-in-data-science-1318c5ac5c09

Mutual Information
https://medium.com/swlh/a-deep-conceptual-guide-to-mutual-information-a5021031fad0

Understanding encoding techniques:
https://medium.com/aiskunks/categorical-data-encoding-techniques-d6296697a40f

# Annexes

## Section A: Variables' Metadata

**Table 1: List of Variables' Metadata**

| Information | Variable | Description | Type |
|---|---|---|---|
| **Claim Dates** | *Accident Date* | Injury date of the claim | Object |
| | *Assembly Date* | The date the claim was first assembled | Object |
| | *C-2 Date* | Date of receipt of the Employer's Report of Work-Related Injury/Illness or equivalent (formerly Form C-2) | Object |
| | *C-3 Date* | Date Form C-3 (Employee Claim Form) was received | Object |
| | *First Hearing Date* | Date the first hearing was held on a claim at a WCB hearing location. A blank date means the claim has not yet had a hearing held | Object |
| **Worker Demographics** | *Age at Injury* | Age of injured worker when the injury occurred | Float64 |
| | *Birth Year* | The reported year of birth of the injured worker | Float64 |
| | *Gender* | The reported gender of the injured worker | Object |
| | *Zip Code* | The reported ZIP code of the injured worker's home address | Object |
| | *Number of Dependents* | Number of dependents | Float64 |
| **Claim and Case Information** | *Alternative Dispute Resolution* | Adjudication processes external to the Board | Object |
| | *Attorney/Representative* | Is the claim being represented by an Attorney? | Object |
| | *Claim Identifier* | Unique identifier for each claim, assigned by WCB | Int64 |
| | *Carrier Name* | Name of primary insurance provider responsible for providing workers' compensation coverage to the injured worker's employer | Object |
| | *Carrier Type* | Type of primary insurance provider responsible for providing workers' compensation coverage | Object |
| | *Average* | The wage used to calculate workers' | Float64 |

| | | | |
|---|---|---|---|
| | *Weekly Wage* | compensation, disability, or an Paid Leave wage replacement benefits | |
| **Location and Region** | *County of Injury* | Name of the New York County where the injury occurred | Object |
| | *District Name* | Name of the WCB district office that oversees claims for that region or area of the state | Object |
| | *Medical Fee Region* | Approximate region where the injured worker would receive medical service | Object |
| **Incident and Injury Details** | *COVID-19 Indicator* | Indication that the claim may be associated with COVID-19 | Object |
| | *IME-4 Count* | Number of IME-4 forms received per claim. The IME-4 form is the “Independent Examiner's Report of Independent Medical Examination” form | Float64 |
| **Industry Classification** | *Industry Code* | NAICS code and descriptions are available at: https://www.naics.com/search-naics-codes-by-industry/ | Float64 |
| | *Industry Code Description* | 2-digit NAICS industry code description used to classify businesses according to their economic activity | Object |
| **Injury Descriptions and Codes** | *OIICS Nature of Injury Description* | The OIICS nature of injury codes & descriptions are available at https://www.bls.gov/iif/oiics_manual_2007.pdf | Float64 |
| | *WCIO Cause of Injury Code* | The WCIO cause of injury codes & descriptions are at https://www.wcio.org/Active%20PNC/WCIO_Cause_Table.pdf | Float64 |
| | *WCIO Cause of Injury Description* | See description of field above | Object |
| | *WCIO Nature of Injury Code* | The WCIO nature of injury are available at https://www.wcio.org/Active%20PNC/WCIO_Nature_Table.pdf | Float64 |
| | *WCIO Nature of Injury Description* | See description of field above | Object |
| | *WCIO Part Of Body Code* | The WCIO part of body codes & descriptions are available at https://www.wcio.org/Active%20PNC/WCIO_Part_Table.pdf | Float64 |
| | *WCIO Part Of* | See description of field above | Object |

| | | | |
|---|---|---|---|
| | *Body Description* | | |
| **Claim Outcomes** | *Agreement Reached* | Binary variable: Yes if there is an agreement without the involvement of the WCB -> unknown at the start of a claim | Float64 |
| | *WCB Decision* | Multiclass variable: Decision of the WCB relative to the claim: "Accident" means that claim refers to workplace accident, "Occupational Disease" means illness from the workplace. -> requires WCB deliberation so it is unknown at start of claim | Object |
| | *Claim Injury Type* | Main target variable: Deliberation of the WCB relative to benefits awarded to the claim. Numbering indicates severity | Object |

Source: Authors.

## Section B: Data Exploration

**Table 2: Changing Data Types**

| **Change Data Types** | **Variables** |
|---|---|
| Float to integer | *Age at Injury, Birth Year, IME-4 Count* and *Number of Dependents* |
| Float to object | *OIICS Nature of Injury Description, Agreement Reached, Claim Identifier, Industry Code, WCIO Cause of Injury Code, WCIO Nature of Injury Code* and *WCIO Part Of Body Code* |
| Object to dates | *C-2 Date, C-3 Date, First Hearing Date, Accident Date,* and *Assembly Dates* |

Source: Authors.

**Table 3: Unusual and Missing Values**

| Variable | Problem | Solution |
|---|---|---|
| *Accident Date* | 3,9% missing values | Dropping rows with all missing values. Imputing median date difference between Assembly and Accident Date. |
| *C-2 Date* | 5,8% missing values | Fill missing values with the correspondent Assembly Date |
| *C-3 Date* | 68,5% missing values | Dropping the variable |
| *First Hearing Date* | 74,6% missing values | Dropping the variable |
| *Age at Injury* | 3,3% missing values | Dropping rows with all missing values |
| *Birth Year* | 5,1% missing values | Fill missing values with the difference between Accident Date and Age at Injury |
| *Gender* | 3,3% unusual ("U") / missing values | Transforming the "U" into missing values. Dropping rows with all missing values. Imputing missing values with mode |
| *Zip Code* | 8,2% missing values | Imputing missing values with the mode of Zip Code per Medical Fee Region |
| *Number of Dependents* | 3,3% missing values | Dropping rows with all missing values |
| *Alternative Dispute Resolution* | 3,3% missing values | Dropping rows with all missing values. Imputing missing values with mode |
| *Attorney/Representative* | 3,3% missing values | Dropping rows with all missing values |
| *Carrier Name* | 3,3% missing values | Dropping rows with all missing values |
| *Carrier Type* | 3,3% unusual ("U") / missing values | Transforming the "U" into missing values. Dropping rows with all missing values. Imputing missing values with mode |
| *Average Weekly Wage* | 5% missing values | Fill missing values with |

| | | |
|---|---|---|
| | | conditional median grouped by Industry Code, excluding zeros |
| *County of Injury* | 3,3% unusual ("U") / missing values | Transforming the "U" into missing values. Dropping rows with all missing values. |
| *District Name* | 3,3% missing values | Dropping all missing values |
| *Medical Fee Region* | 3,3% unusual ("U") / missing values | Transforming the "U" into missing values. Dropping rows with all missing values |
| *COVID-19 Indicator* | 3,3% missing values | Dropping rows with all missing values |
| *IME-4 Count* | 76,9% missing values | Doesn´t consist in a problem because not all claims must have this form, so 0 is a possible value |
| *Industry Code* | 5% missing values | Imputing missing values with mode |
| *Industry Code Description* | 5% missing values | Imputing missing values with mode |
| *OIICS Nature of Injury Description* | 100% missing values | Dropping the variable |
| *WCIO Cause of Injury Code* | 6% missing values | Imputing missing values with conditional mode grouped by Industry Code |
| *WCIO Cause of Injury Description* | 6% missing values | Imputing missing values with conditional mode grouped by Industry Code |
| *WCIO Nature of Injury Code* | 6% missing values | Imputing missing values with mode |
| *WCIO Nature of Injury Description* | 6% missing values | Imputing missing values with mode |
| *WCIO Part Of Body Code* | 6,2% missing values | Imputing missing values with conditional mode grouped by Industry Code |
| *WCIO Part Of Body Description* | 6% missing values | Imputing missing values with conditional mode grouped by Industry Code |
| *Agreement Reached* | 3,3% missing values | Dropping rows with all missing values |
| *WCB Decision* | 3,3% missing values | Dropping rows with all missing values |

| | | |
|---|---|---|
| *Claim Injury Type* | 3,3% missing values | Dropping rows with all missing values |

Source: Authors.

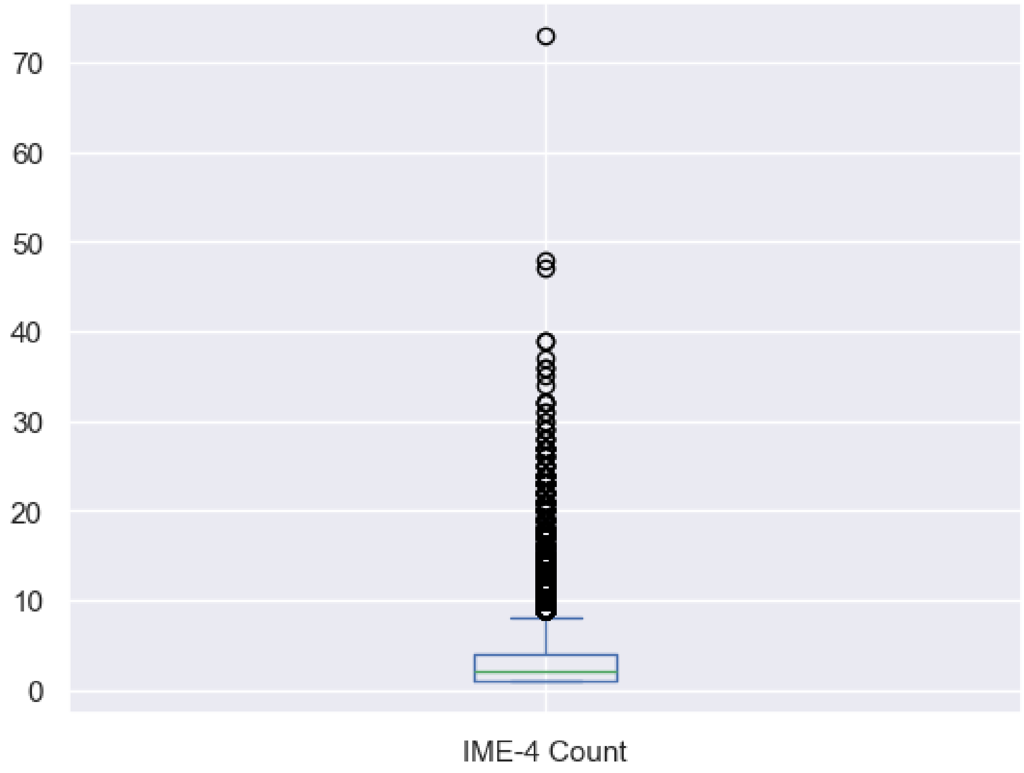
**Table 4: Incoherencies Treatment**

| Variable | Problem | Solution |
|---|---|---|
| *WCIO Part Of Body Code* | One negative value of '-9' | Transforming negative value in positive since there was no corresponding code with the number “9”. |
| *WCIO Part Of Body Description* | Difference number of unique values comparing the corresponding Code variable | Assumption that the code appearing first is the correct one, we replace the duplicated descriptions for the first occurrence of its corresponding code |
| *Industry Code Description* | Difference number of unique values comparing the corresponding Code variable | Assumption that the code appearing first is the correct one, we replace the duplicated code for the first occurrence of its corresponding descriptions |
| *WCIO Cause of Injury Description* | Difference number of unique values comparing the corresponding Code variable | Assumption that the code appearing first is the correct one, we replace the duplicated code for the first occurrence of its corresponding descriptions |
| *Accident Date* | Some accident dates occur after the assembly date or the C-2 date (cannot occur because to be properly documented was to have the prerequisites for the accident) | Switching values, in order to Accident Date occurs before its assembly |

| | | |
|---|---|---|
| *Birth Year* | Rows with values "0" | Replacing the value based on the difference between the Accident Date and Age at Injury |
| *Zip Code* | Presence of letters in this variable could indicate data entry errors | Transforming these rows into missing values |
| *Average Weekly Wage* | Zero as values could misleading to understand if it's either a volunteers salary or a missing or incorrect data | Zero values of industries that cannot have volunteers were treated as missing values and replaced by the median per industry |
| *Number of Dependents* | Unreliable data (distribution shows unusual similar frequencies, such as individuals under 20 having multiple dependents) | Dropping the variable, since a lot of variables did not make sense |

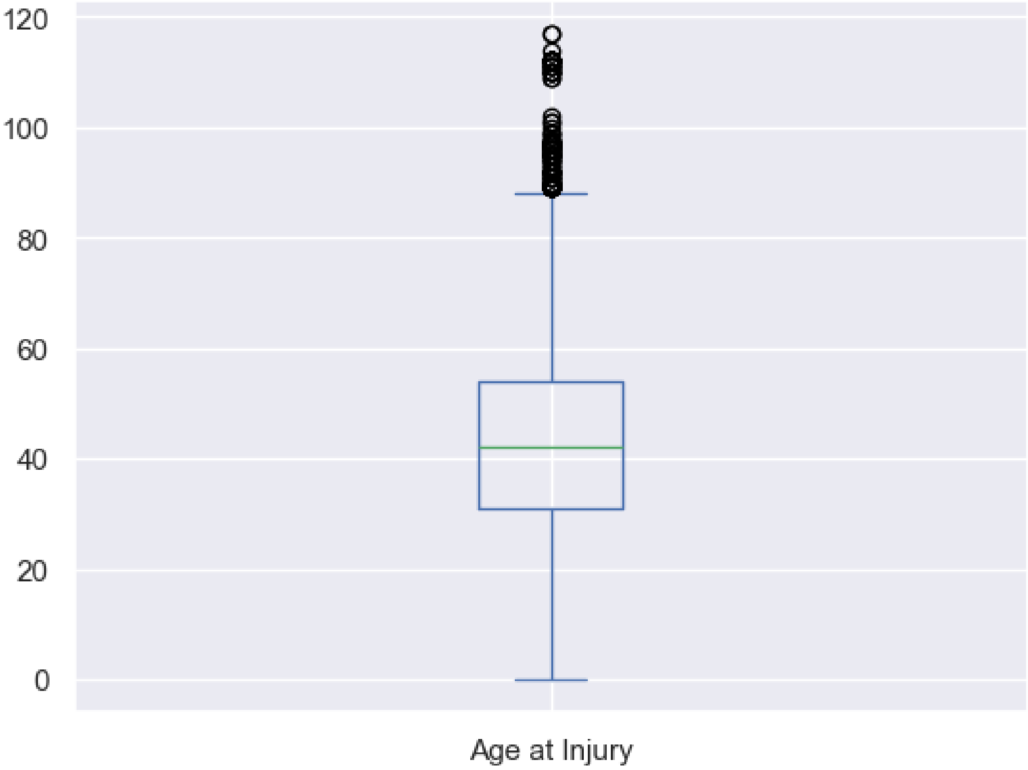
Source: Authors.

Figure 1: IME-4 Count Box plot



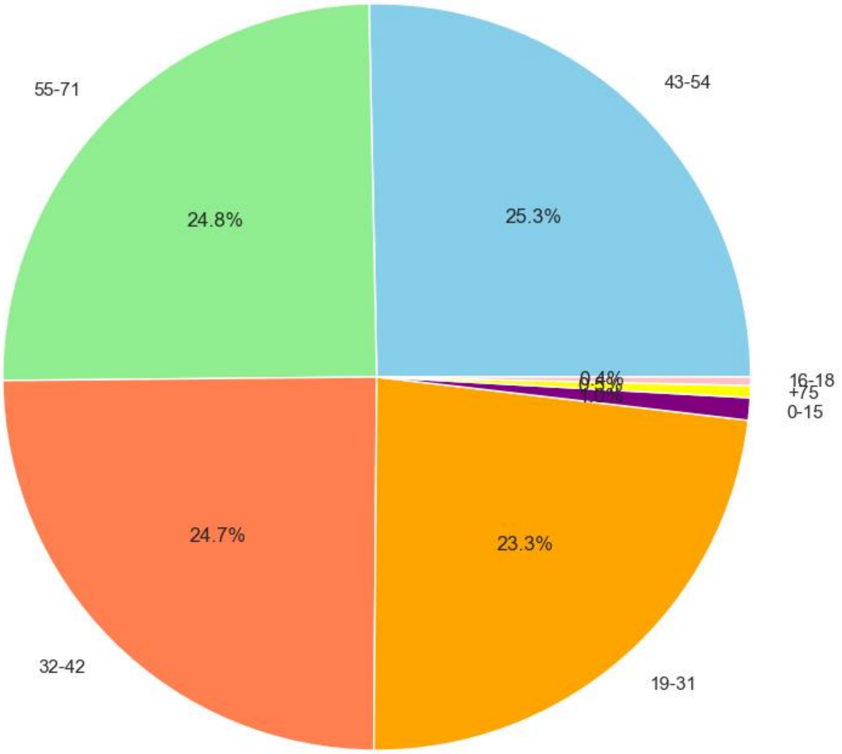
Source: Authors.

Figure 2: Age at Injury Box plot



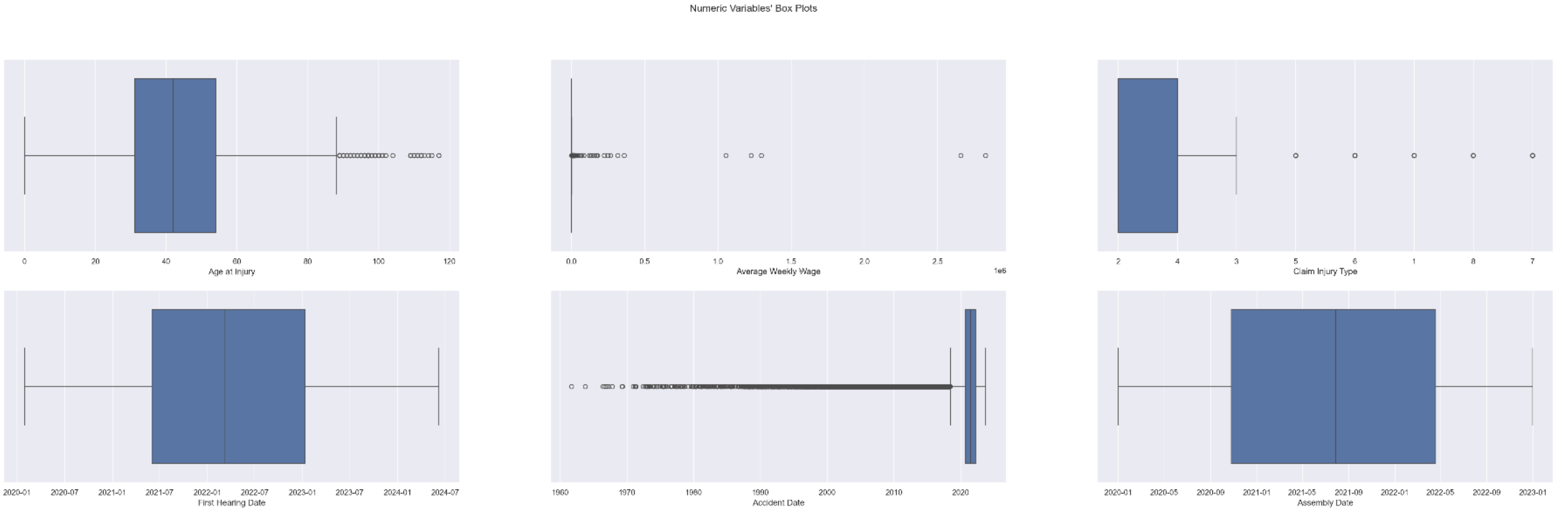
Source: Authors.

Figure 3: Age at Injury Pie chart



Source: Authors

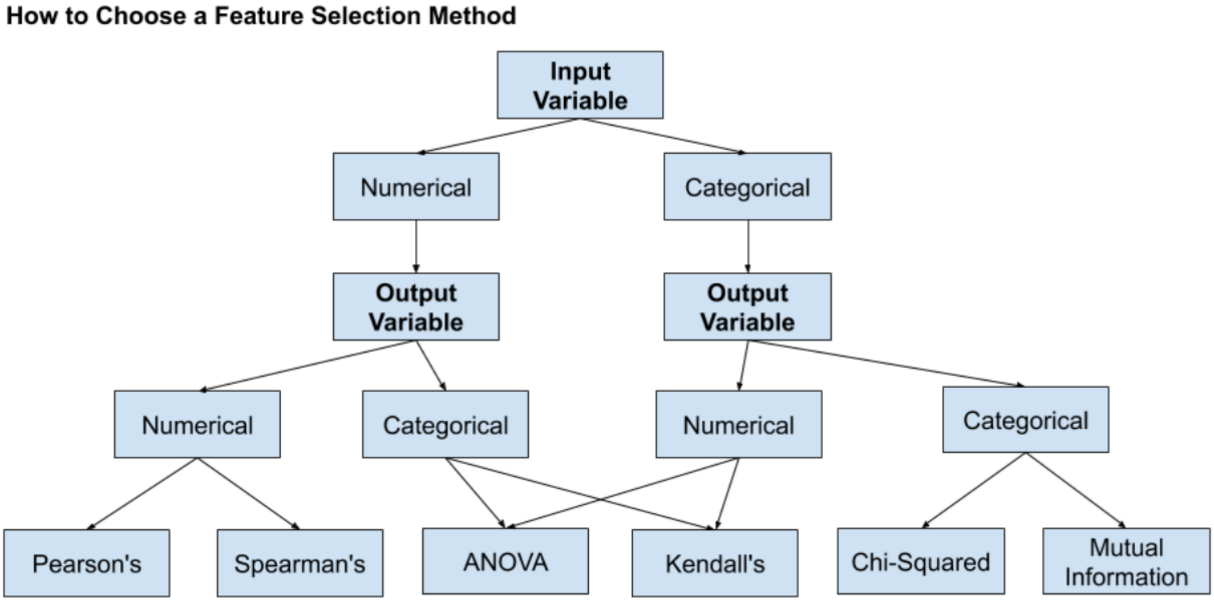
Figure 4: Numeric Variables' Box Plots



Source: Authors

## Section C: Variables for Feature Selection

Figure 5: Feature Selection Method



Source: Diagram from slide 7 of lecture 3, about Statistical Tests

**Table 5: Initial Variables for Feature Selection**

| Variable | Context | Decision |
|---|---|---|
| *Accident Date* | Date variable | Extract the month and year to new different variables |
| *Assembly Date* | Date variable | Extract the month and year to new different variables |
| *C-2 Date* | Date variable | Extract the month and year to new different variables |
| *C-3 Date* | Date variable | Drop because of the extreme percentage of missing values |
| *First Hearing Date* | Date variable | Drop because of the extreme percentage of missing values |
| *Age at Injury* | Numerical variable | Kept as is |
| *Birth Year* | Numerical variable | Kept as is |
| *Gender* | Categorical with three different values | Transformed into a new binary variable |
| *Zip Code* | Categorical Variable | Kept but transformed into a numerical variable |
| *Number of Dependents* | Unreliable variable because of is high | Dropped |
| *Alternative Dispute Resolution* | Categorical with two different values | Transformed into a new binary variable |
| *Attorney/Representative* | Categorical with two different values | Transformed into a new binary variable |
| *Claim Identifier* | Categorical variable | Kept as is |
| *Carrier Name* | Categorical variable | Kept as is |
| *Carrier Type* | Categorical variable | Kept as is |
| *Average Weekly Wage* | Numerical variable | Kept as is |
| *County of Injury* | Categorical variable | Kept as is |
| *District Name* | Categorical variable | Kept as is |
| *Medical Fee Region* | Categorical Variable | Kept but transformed into a numerical variable |
| *COVID-19 Indicator* | Categorical with two different values | Transformed into a new binary variable |
| *IME-4 Count* | Numerical variable | Kept as is |
| *Industry Code* | Categorical variable | Kept as is |
| *Industry Code Description* | Categorical variable | Kept as is |
| *OIICS Nature of Injury Description* | Variable composed of 100% of missing values | Dropped |

| | | |
|---|---|---|
| *WCIO Cause of Injury Code* | Categorical variable | Kept as is |
| *WCIO Cause of Injury Description* | Categorical variable | Kept as is |
| *WCIO Nature of Injury Code* | Categorical variable | Kept as is |
| *WCIO Nature of Injury Description* | Categorical variable | Kept as is |
| *WCIO Part Of Body Code* | Categorical variable | Kept as is |
| *WCIO Part Of Body Description* | Categorical variable | Kept as is |
| *Agreement Reached* | Categorical with two different values | Transformed into a new binary variable |
| *WCB Decision* | Categorical variable with no variance | Dropped |

Source: Authors.

**Table 6: Newly Created Variables for Feature Selection**

| **Variable** | **Engineering Action** | **Description** |
|---|---|---|
| *Days from Accident to C-2* | Created | Represents the number of days that it took to receive the employer report since the accident happened |
| *Days from Accident to Assembly* | Created | Represents the number of days that it took to first assemble the claim since the accident happened |
| *Log_Average_Weekly_Wage* | Created | Represents the logarithmic function of Average Weekly Wage in order to minimize the effect of extreme values |
| *Age_Group* | Created | The Age at Injury were split into age groups, to try to find patterns |
| *Claim Antiguity* | Created | Represents the order of accidents antiguity, where the first one to happen is represented with 1 and the most recent one with n |
| *C-2 under Deadline* | Created | Check if C-2 was completed under deadline, which is 10 days |
| *Forms Delivered Count* | Created | Counts how many forms were delivered, including C-2, C-3 and IME |
| *Valid Full Claim* | Created | Check if every forms was completed |

| | | |
|---|---|---|
| | | and if every forms respected its deadline |
| *cause_of_injury_groups* | Created | Categorize similar cause of injuries into broader categories, making it easier to analyze and identify patterns |
| *body_part_groups* | Created | Categorize close parts of the body into broader categories, making it easier to analyze and identify patterns |
| *Nature of Injury Group* | Created | Categorize similar nature of injuries into broader categories, making it easier to analyze and identify patterns |
| *industry_groups* | Created | Categorize similar industry into broader categories, making it easier to analyze and identify patterns |
| *Accident Year* | Transformed | Represents the year of the accident, which we believe will be very useful in our analysis |
| *Accident Month* | Transformed | Similar to Accident Year, represents the month of the accident. Here we expect to find some differences related to the time of e year |
| *Assembly Year* | Transformed | Represents the year of the assembly, which is probably the same as Accident Year |
| *Assembly Month* | Transformed | Represents the month of the assembly, useful to calculate the distance from the accident to the assembly |
| *C-2 Date Year* | Transformed | Represents the year of C-2 |
| *C-2 Date Month* | Transformed | Represents the month of C-2 |
| *Alternative Dispute Resolution* | Transformed | This variable had two possible values – yes and no; was transformed into a binary |
| *Attorney/Representative* | Transformed | This variable had two possible values – yes and no; was transformed into a binary |
| *COVID-19 Indicator* | Transformed | This variable had two possible values – yes and no; was transformed into a binary |
| *WCB Decision* | Transformed | Despite this variable having just one |

| | | |
|---|---|---|
| | | value, we transformed it into a binary |
| *Gender* | Transformed | This variable had three possible values – M (male), F (female) and X (non-binary); as the most common value is male, we transformed Gender in a binary where 1 represents Male and 0 represents Not Male |
| *Medical Fee Region* | Transformed | This variable was encoded, because the initial part of each value used to be a roman numeral, so we transformed it into the correspondent number |

Source: Authors.

## Section D: Feature Selection

**Table 7: Feature Selection**

| Feature | Variance | Kendall | Target Kendall | ANOVA | LASSO | DT | Random Forest | Result |
|---|---|---|---|---|---|---|---|---|
| *Age at Injury* | Keep | Keep | Keep | Keep | Discard | Keep | Keep | **Selected** |
| *Alternative Dispute Resolution* | Discard | Keep | Discard | Discard | Discard | Discard | Discard | **Not Selected** |
| *Attorney/ Representative* | Keep | Keep | Keep | Keep | Keep | Keep | Keep | **Selected** |
| *Average Weekly Wage* | Discard | Keep | Keep | Discard | Discard | Keep | Keep | **Not Selected** |
| *Birth Year* | Discard | Discard | Keep | Keep | Keep | Keep | Keep | **Selected** |
| *COVID-19 Indicator* | Discard | Keep | Discard | Discard | Keep | Discard | Discard | **Not Selected** |
| *Gender* | Keep | Keep | Discard | Discard | Keep | Discard | Discard | **Not Selected** |
| *IME-4 Count* | Discard | Keep | Keep | Keep | Discard | Discard | Keep | **Not Selected** |

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| *First Hearing Date* | Keep | Keep | Keep | Keep | Keep | Discard | Keep | **Selected** |
| *C-3 Delivery* | Keep | Keep | Keep | Keep | Keep | Discard | Discard | **Selected** |
| *Days from Accident to C-2* | Discard | Keep | Discard | Discard | Discard | Keep | Keep | **Not Selected** |
| *Days from Accident to Assembly* | Discard | Discard | Discard | Discard | Keep | Keep | Keep | **Not Selected** |
| *Log_Average_Weekly_Wage* | Discard | Discard | Keep | Keep | Keep | Keep | Keep | **Selected** |
| *Claim Antiguity* | Keep | Discard | Discard | Discard | Keep | Keep | Keep | **Not Selected** |
| *C-2 under Deadline* | Keep | Keep | Discard | Keep | Keep | Discard | Discard | **Not Selected** |
| *Forms Delivered Count* | Discard | Discard | Keep | Keep | Keep | Keep | Keep | **Selected** |
| *Valid Full Claim* | Keep | Keep | Keep | Keep | Keep | Discard | Discard | **Selected** |
| *Accident Year* | Discard | Discard | Discard | Discard | Discard | Discard | Discard | **Not Selected** |
| *Accident Month* | Keep | Keep | Discard | Discard | Keep | Discard | Discard | **Not Selected** |
| *Assembly Year* | Keep | Keep | Keep | Keep | Keep | Discard | Discard | **Selected** |
| *Assembly Month* | Keep | Discard | Discard | Discard | Keep | Discard | Discard | **Not Selected** |
| *C-2 Year* | Discard | Keep | Discard | Discard | Discard | Discard | Discard | **Not Selected** |
| *C-2 Month* | Keep | Keep | Discard | Discard | Keep | Discard | Discard | **Not Selected** |
| *Carrier Name* | Keep | Keep | Discard | Discard | Keep | Keep | Keep | **Selected** |
| *Carrier Type* | Keep | Keep | Discard | Discard | Keep | Discard | Discard | **Not Selected** |

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| *County of Injury* | Keep | Keep | Discard | Discard | Keep | Keep | Keep | **Selected** |
| *District Name* | Keep | Keep | Discard | Discard | Keep | Discard | Discard | **Not Selected** |
| *Industry Code* | Keep | Keep | Discard | Discard | Keep | Keep | Keep | |
| *Medical Fee Region* | Keep | Keep | Discard | Discard | Discard | Discard | Discard | **Not Selected** |
| *WCIO Cause of Injury Code* | Keep | Keep | Discard | Discard | Keep | Keep | Keep | **Selected** |
| *WCIO Nature of Injury Code* | Keep | Keep | Discard | Keep | Keep | Keep | Keep | **Selected** |
| *WCIO Part Of Body Code* | Keep | Keep | Discard | Keep | Keep | Keep | Keep | **Selected** |
| *Zip Code* | Discard | Keep | Discard | Discard | Keep | Keep | Keep | **Not Selected** |
| *Age_Group* | Keep | Keep | Discard | Discard | Discard | Discard | Discard | **Not Selected** |
| *Injury Group* | Keep | Keep | Discard | Discard | Keep | Discard | Discard | **Not Selected** |
| *Body Part Group* | Keep | Keep | Discard | Keep | Keep | Keep | Keep | **Selected** |
| *Nature of Injury Group* | Keep | Keep | Discard | Keep | Discard | Discard | Discard | **Not Selected** |
| *Industry Group* | Keep | Keep | Discard | Discard | Keep | Keep | Discard | **Not Selected** |

Source: Authors.

**Table 8: Models Explanation**

| Models | Explanation |
|---|---|
| Random Forest | To enhance classification performance, this ensemble learning technique combines several decision trees. With the help of feature selection and random sampling, it reduces overfitting and produces reliable predictions when working with big datasets |
| Logistic Regression | A straightforward and understandable linear model that works well for binary classification applications. It struggled with the dataset's class imbalance and lacked the complexity required to get top results, despite its adequate performance |
| Neural Networks | To represent non-linear correlations in data, this model makes use of interconnected layers of nodes. Despite its adaptability, it underperformed in our investigation because of insufficient fine-tuning and required a large amount of processing resources |
| Naive Bayes | A probabilistic model that assumes feature independence and is based on Bayes' theorem. Although computationally efficient, it fared worse than more sophisticated approaches and had trouble with feature correlations |
| LightGBM | A gradient-boosting system made to be quick and effective, particularly when working with big datasets. LightGBM produced outstanding outcomes, showcasing its ability to manage intricate patterns and unbalanced data |
| XGBoost | Another potent gradient-boosting technique that is renowned for its precision and effectiveness. With its outstanding performance, XGBoost demonstrated its capacity to manage imbalanced classes and a variety of feature interactions |
| Decision Tree | In order to generate predictions, the Decision Tree model divides data into subsets according to feature values, creating a structure resembling a tree. It offers insights on feature relevance and performs well with non-linear data. It is interpretable and intuitive, although it is prone to overfitting, which can be lessened by methods like tree depth limitation or pruning. |

Source: Authors.

## Section E: Metrics Calculation for each model

**Table 9: Metrics Calculation for each model (validation set results sorted by f1 score)**

| Model | Precision | Recall | F1 Score | Kaggle Score |
|---|---|---|---|---|
| **LightGBM** | 0,4770 | 0,4627 | 0,4599 | 0,4103 |
| **XGBoost** | 0,4479 | 0,4835 | 0,4556 | 0,3926 |
| **Random Forest** | 0,3804 | 0,5205 | 0,4095 | 0,3260 |
| **Decision Tree** | 0,3349 | 0,3986 | 0,3484 | 0,2431 |
| **Neural Networks** | 0,3197 | 0,5034 | 0,3208 | 0,2453 |
| **Logistic Regression** | 0,2877 | 0,5224 | 0,2750 | 0,1949 |
| **Nayve Bayes** | 0,2617 | 0,3976 | 0,2069 | 0,1676 |

Source: Authors.

**Table 10: Metrics Calculation for each model with optimizations (validation set results sorted by f1 score)**

| Model | Precision | Recall | F1 Score | Kaggle Score |
|---|---|---|---|---|
| **Grid Search XGBoost** | 0,4577 | 0,4766 | 0,4592 | 0,4075 |
| **Grid Search LightGBM** | 0,4521 | 0,4727 | 0,4524 | 0,4103 |
| **Grid Search Random Forest** | 0,4133 | 0,4823 | 0,4325 | 0,341 |

Source: Authors.

**NOTE:** Where is our Git Hub repository: https://github.com/joaopr03/ml