MDSAA

Master's Degree Program in

Data Science and Advanced Analytics

Machine Learning

Predicting Claim Injury Type

Diogo Melo, ID: 20240698

Francisca Salgado, ID: 20241040

Jéssica Vicente, ID: 20230744

João Diogo Trindade, ID: 20240590

Group 11

NOVA Information Management School

Instituto Superior de Estatística e Gestão de Informação

Universidade Nova de Lisboa

19 December, 2024

INDEX

Content

1. ABSTRACT 2

2. INTRODUCTION 3

3. METHODOLOGY 4

3.1. DATA UNDERSTANDING & EXPLORATION 4

3.2. DATA PREPROCESSING 5

3.2.1. MISSING VALUES | TRANSFORMATIONS 6

3.2.2. OUTLIERS 7

3.3. ENCODERS 7

3.3.2. SCALING DATA 8

4. MULTICLASS MODEL ASSESSMENT AND EVALUATION 9

5. OPEN-ENDED SECTION 11

6. CONCLUSION 12

7. REFERENCES 13

8. ANNEXES 14

1. ABSTRACT

This report addresses inefficiencies in the New York Workers' Compensation Board (WCB) system for processing compensation claims, which currently relies on manual, labor-intensive methods. Our project aims to develop predictive models to streamline decision-making, reduce delays, and enhance the accuracy of claim resolutions. Using data from 2020 to 2022, the primary objectives were twofold: (1) to predict whether an agreement could be reached without WCB involvement ("Agreement Reached") and (2) to classify the type of injury associated with a claim through a multiclass classification model.

The methodology involved comprehensive data preprocessing to address missing values and outliers, alongside feature engineering and selection to create and refine predictive variables. Variables deemed irrelevant were dropped, while new features were crafted to improve model performance. For the binary classification task, metrics such as F1 Binary were prioritized to evaluate performance. For the multiclass classification model, metrics including F1 score, precision, recall, and accuracy were used to identify the optimal approach. Hyperparameter tuning was employed to further enhance model efficiency and performance.

The findings underscore the critical role of effective feature selection and model optimization in improving prediction accuracy. The study highlights the potential of machine learning to alleviate the workload of human assessors, ensuring a more efficient and reliable claims management system. By automating key processes, this project provides actionable insights into modernizing the WCB's operations, significantly improving both system efficiency and claims resolution reliability.

2. INTRODUCTION

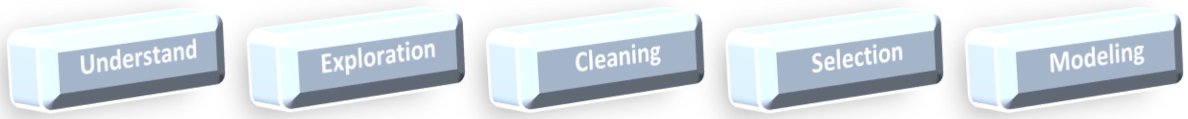
The New York Workers' Compensation Board (WCB) faces significant challenges in processing workers' compensation claims, particularly due to the manual and labor-intensive review process. This inefficiency often leads to delays in claim resolution, negatively impacting the lives of claimants and placing additional pressure on the system.

The main aim of this project is to develop a robust classification model that can predict the type of injury a claim will be, using data from 2020 to 2022. To improve this process, machine learning offers a promising solution by automating decision making, which could lead to faster and more accurate claim resolutions. Hyperparameter tuning and feature selection will be critical steps to optimize the model's performance and ensure its accuracy and reliability. In addition, the project will explore the importance of different features in the prediction process and evaluate alternative predictor variables, such as the "Agreement Reached". By automating the process, this approach will significantly reduce the workload of human adjudicators and minimize delays in the WCB's claims processing system.

Challenges like those faced by the WCB have been addressed in other sectors, notably healthcare and insurance. In healthcare, predictive models have been used to predict hospital readmissions and improve patient outcomes, demonstrating the ability of machine learning to streamline processes and uncover critical factors. Similarly, the insurance industry has successfully used classification models to optimize claims processing and detect fraud, demonstrating the transformative potential of data-driven decision-making in complex systems.

3. METHODOLOGY

The following workflow outlines the methodology employed for data processing, ensuring a systematic and efficient transformation of raw data into actionable insights. By integrating historical data with newly created features, this study aims to enhance the model's accuracy and predictive power by retaining only the most informative features.



3.1. DATA UNDERSTANDING & EXPLORATION

The dataset is divided into two parts: a training set containing claims data from the beginning of 2020 to the end of 2022, and a test set containing claims compiled from January 2023 onwards. The training had 593,471 observations and 33 columns, with each record describing a claim through various attributes. The test dataset retains the same descriptive attributes but excludes any information that depends on decisions made after the claims were filed. This distinction supports the primary objective of building and validating machine learning models on the training data to predict the type of claim injury. This target variable includes several categories representing the outcome of the claim, including NON-COMP, TEMPORARY, MED ONLY, PPD SCH LOSS, CANCELLED, PPD NSL, DEATH, and PTD.
Further in the Report, in some cases, we will divide those in 4 groups based on injury severity:
No compensation (NON-COMP & CANCELLED), **low severity** (MED ONLY & TEMPORARY), **medium severity** (PPD SCH LOSS & PPD NSL) and **high severity** (PTD & DEATH).

Further ahead on **Open Mind Section**, we also describe the identical process we implemented to predict the Agreement Reached Feature, which was then used to predict the multiclass classification.

In this phase of **data understanding and exploration**, we focused on getting to **know our data** using different **python methods**, looking for **inconsistencies** and **plotting the train data** for a clear view.
We started by checking for columns with **100% of missing values** as it wouldn´t contribute any information to the analysis, and so, “OIICS Nature of Injury Description” was instantly drop.
In addition, the column “WCB Decision” was excluded due to not being relevant for the purpose of the analysis.

Regarding **Claim Identifiers**, the train dataset contained both 7-digit and 9-digit claim identifier values. However, all claims with 9-digit identifiers had missing data, s**o only the claims with 7-digit identifiers were retained to maintain data integrity**. In terms of data type transformations, all date type columns were updated to **date/time format** to ensure proper handling for time-based analysis and feature engineering.

Building on this foundation, we **examined key characteristics to gain additional insight**. This was made through **EDA** as we did go **through every feature** and tried to **look for inconsistencies** and both **information gain towards both of target variables** (Agreement Reached and Claim Injury Type).

Two of the main inconsistencies we found:

- In many cases **all forms date and assembly date** were **before** the **actual Accident date**, leading to some **problems defining the best solution for time** between date features.
- Regarding **Age at Injury**, the minimum age was **zero**. Nonetheless, there were cases where **Birth Year** was totally different, and we couldn´t retrieve **correct age** due to missing value in Accident date column.

Important insights obtained via EDA:

1. **High severity** injuries are more likely to happen in old seniors (above 75+) see annexes (Fig 1);
2. When dealing with **medium/high severity injuries**, it´s **most likely** that a person will have at **least one IME4** (except for Death, where 80% of times, there is no IME-4) see annexes (Fig 2);
3. As **severity increases**, is **more likely** a person **had a hearing held.** see annexes (Fig 3);
4. The fact a person **had an Alternative dispute resolution**, makes it **99.3% likely to belong to class "NON-COMP"**. The remaining 0.7% are "CANCELLED". see annexes (Fig 4);
5. Having an attorney is also very important specially in **medium severity injuries, where every time each one happen, it´s at least 89% likely to have an attorney.** see annexes (Fig 5);
6. Regarding top 20 Insurance companies, we can observe notice some behaviors. For instance, in claim "**PPD SCH LOSS**" 45.8% are from "**POLICE, FIRE, SANITATION**". see annexes (Fig 6);
7. Regarding the **general body parts**, notice how **Trunk**, **Head** and **Multiple Body** parts are the ones with the **higher** proportion in **high severity injuries.** see annexes (Fig 7);
8. The most **frequently** grouped **cause of injuries** is Contact with Objects and Equipment, Overexertion and Bodily Reaction and Falls or Slips. see annexes (Fig 8);
9. Regarding the **Nature of Injury**, we can see that **Tumors and Cancers** are the group where more 10.5% of observations are **high severity injuries.** see annexes (Fig 9).
10. People **associated with Covid 19** are also the ones with hig**her proportion in high severity** injuries, specific on **Death**, where from all the people who **died**, around **36**% was associated with **Covid-19**. see annexes (Fig 10)

3.2. DATA PREPROCESSING

To facilitate model selection and prevent data leakage, the dataset was split into two subsets: a training set (70% of the data) and a validation set (30% of the data). This process ensured that the model could **learn patterns from the training set**, while the **validation** set served as a **reference** to **evaluate performance** on **unseen data**. After the initial organization of the test dataset, additional transformations were carried out to prepare the variables appropriately for the modelling process.

The OIICS Nature of Injury Description column was removed from the **test dataset because it was irrelevant (contained 100% missing values)**, helping to reduce dimensionality. Date-related columns were converted to the datetime format to ensure correct interpretation of the variables. In case of invalid values, they were automatically replaced with NaT (Not a Time). We also fixed data types of some columns that were supposed to be Integers.

3.2.1. MISSING VALUES | TRANSFORMATIONS

The missing values were mostly represented by the NaN value, which allowed us to detect their presence using the “isnull” function. Since some machine learning algorithms do not support missing values, we addressed them appropriately.

Using the *Birth Year* variable, we resolved some cases of **zero values in *Age at Injury***. When missing values were present, they were imputed using the median. For valid *Birth Year* entries, the age at the time of the accident was calculated as the **difference between *Accident Date* and *Birth Year***. After processing the *Age at Injury* variable, the *Birth Year* column was removed because the age had already been computed.

Since **date-related** variables showed **high levels of missing values**, particularly ***First Hearing Date*** and ***C-3 Date***, we created **binary** variables indicating whether the date was **present** (1) or **absent** (0).

Missing values *from “Accident date” column* was handled using the mode, and we found it beneficial to extract temporal components such as *day, day of the week, month and year”.*

To effectively manage some of the variables, we transformed some into **binary variables**:

- For ***Attorney/Representative, COVID-19 Indicator,*** and ***Alternative Dispute Resolution,*** a value of 1 indicates "Yes" and 0 indicates "No".
- For ***Number of Dependents***, 1 represented the presence of dependents, while 0 indicated no dependents.
- For ***Gender***, **missing values** were filled with the **mode**, and values were encoded as **1** for **Male** and **0** for **Female**.

For **WCIO variables** such as *WCIO Part of Body Code, WCIO Nature of Injury Code, WCIO Cause of Injury Code*, and *Industry Code*, missing values were replaced with the label "Not applied". These variables were categorized into smaller groups to reduce cardinality. Further ahead, initial WCIO descriptions were dropped. Only initial codes stored as objects and created general groups were kept in dataset.

Regarding Zip Code, we found a table **mapping all NYS Zip Codes to each county**, so we **merged** it into our dataset. This was made to return the counties of the zip codes which helped us to **reduce cardinality** in initial “Zip Code”. After the match, **missing values** were filled with the **most frequent county**.

For the ***Carrier Type*** variable, the group decided to **merge all Special Fund Carrier Types** into a single category labeled "**5. SPECIAL FUND**" because their individual frequencies were quite low. Also, a new binary variable was created from the *Carrier Type*, where 1 indicates "UNKNOWN" and 0 all the rest. This distinction was made for the model recognize that this person didn´t have/provide insurance information.

For the ***Carrier Name*** variable due to **high cardinality**, the top 20 most frequent carrier names in the training data were retained as categories, while less frequent names were grouped under "Other". However, we didn´t drop initial column and take them both into feature selection.

Since the ***IME-4 Count*** variable contained 77.6% missing values, **these were replaced with 0**. A new **binary variable** was created based on *IME-4 Count*, with 1 indicating a count greater than zero and 0 otherwise. This transformation highlighted whether records associated with "**IME-4" existed or not**.

For the **Average Weekly Wage** variable, missing values were **imputed** with the **median** of the **training** set by **Industry**.

3.2.2. OUTLIERS

In this project, the **IQR method** was used to handle outliers. Numerical features were first visualized using box plots, which revealed significant inconsistencies. We are talking about features like "**IME4-Count**", "**Average Weekly Wage**", "**Age at Injury**", "**Assembly delay**", "**c2 to accident delay**".

However, for the model to choose what feature was the best, we also **created additional features** by **categorizing** the initial one. This is also a **good practice to deal with outliers**, although being subject to **some loss of information**. Nevertheless, we **kept both features** (initial and categorized one) to see how the model would react.

3.3. ENCODERS

3.3.1. Encodings for Categorical Variables

Initially, **One-Hot Encoding** was applied to transform categorical variables into numerical representations dealing very well with models such as Neural Networks. However, due to the curse of dimensionality we decided not to go through.

Frequency Encoding was ultimately chosen why?

1. **Frequency of each category** can be a **valuable information** that a model can learn from.
2. **Handles high cardinality** which was present in some of our columns.
3. Keep the **same number of features**.

Disadvantages:

1. Labels that appear the **same number of times or labels that have low number of frequencies** could lead the model to **loss information** and **risk of overfitting**.
2. Doesn´t **handle unseen categories** too well.

How we tried to overcome this with the parameters:

1. Use of "**handle_unknown**" set to "value" to handle unseen categories with a default value (typically 0). This was particularly useful for Assembly and Accident Date due to in test set not being present the year 2024.
2. Use of "**min_group_size**" set to 50 to select only categories that appear at least 50 times. This helped to control overfitting and complexity for the model.

3.3.2. SCALING DATA

Why bother to scale the data?

Because without scaling, features with larger ranges can dominate the learning process, leading to biased models. Scaling ensures that each feature contributes proportionately to the model.

StandardScaler was used to normalize numerical variables, where it transforms the distribution of each feature to have a mean of zero and a standard deviation of one.

We also considered using **MinMax** scaler, however, **StandardScaler** is relatively robust to the presence of outliers compared to **MinMax** scaling, as it relies on the mean and standard deviation rather than the range of the data.

3.3.3. FEATURE SELECTION

In this topic, a combination of feature selection methods, including Permutation Importance with Logistic Regression and RidgeCV, filter and embedded methods was employed. Each method aimed to identify the most relevant variables while discarding those with minimal contribution to the model.

Table 1- Pros and Cons of methods in Feature Selection

Method	Pros	Cons
Variance	Low variance features might add noise to the model.	Low variance features can be important if a minority category is important for a class.
Correlation Matrix	If features are highly correlated, might introduce redundancy.	Despite high correlation, variables can be important for the model.
Mutual Information Gain	Provide Importance of each feature regarding the target.	Bias Towards Features with More Categories.
Permutation Importance (Logistic Regression)	Can capture non-linear relationships between features and the target variable.	Can be sensitive to noise in the data.
Permutation Importance (RidgeCV)	Adds a penalty which helps in handling multicollinearity.	Importance scores can vary by specific train-test split, which can lead to instability in the results.
LassoCV	Performs both variable selection and regularization.	Can struggle with multicollinearity.

We also determine thresholds for some of those methods by observing which features didn´t really have any importance in that method. For instance, in correlation set above 90%, in permutation importance defined 0.005 or 0.0001. On the other hand, LassoCV picked some to discard.

For our Multiclass Model, we tested two different set of features:

1. **Drop** the **discard features** by at **least 3 of** the **5 methods** (except for Variance). Also, **variables** with **same information** but **aggregate** in **categories** or **similar** were also drop (the one with poorest performance) - **kept 29 features**
2. **Drop only** the **discard features by at least 3 of** the **5 methods** (except for Variance) – keep 39 features).

Decisions / Results: Drop only ['County of Injury', 'Number of Dependents', '9_11_period', 'Carrier_binary', 'Accident_Day', 'Age_Category']. see annexes (Fig 11)

Despite this 1st way sound **practical** to **reduce redundance**, the model XGBoost performed well (0.45 F1 Score Macro) and around 0.43 in Kaggle.

However, **maintaining** some of the features that give the same information seemed to add a **little extra perspective** and more data for the model to focus, leading to a **slightly better result in both validation F1 Macro (0.466)** and **Loss (0.500),** also it gave us the **best performance on Kaggle (0.447).**

4. MULTICLASS MODEL ASSESSMENT AND EVALUATION

METRICS:

- **Macro F1** is a performance metric used to evaluate classification models, **especially on imbalanced datasets**. It is the **average** of the F1-scores calculated for **each class** in the dataset, treating all **classes equally**, regardless of their size.
- **Precision** measures to determine how many of the predicted positive results are correct.
- **Recall** can be of **all the actual positives**, **how many did I identify**?
- **Log Loss** is a performance metric for classification models that measures how well the predicted probabilities match the actual class labels. A log loss **closer to zero** means that the model **predicts probabilities close** to the **true labels**.

Taking into **consideration these metrics**, the ones who constantly show good metrics were **MLP**, **XGB** and **HISTGB**. **So, for HISTGB and MLP**, we will see further ahead how the most important parameters would affect the performance of the model.

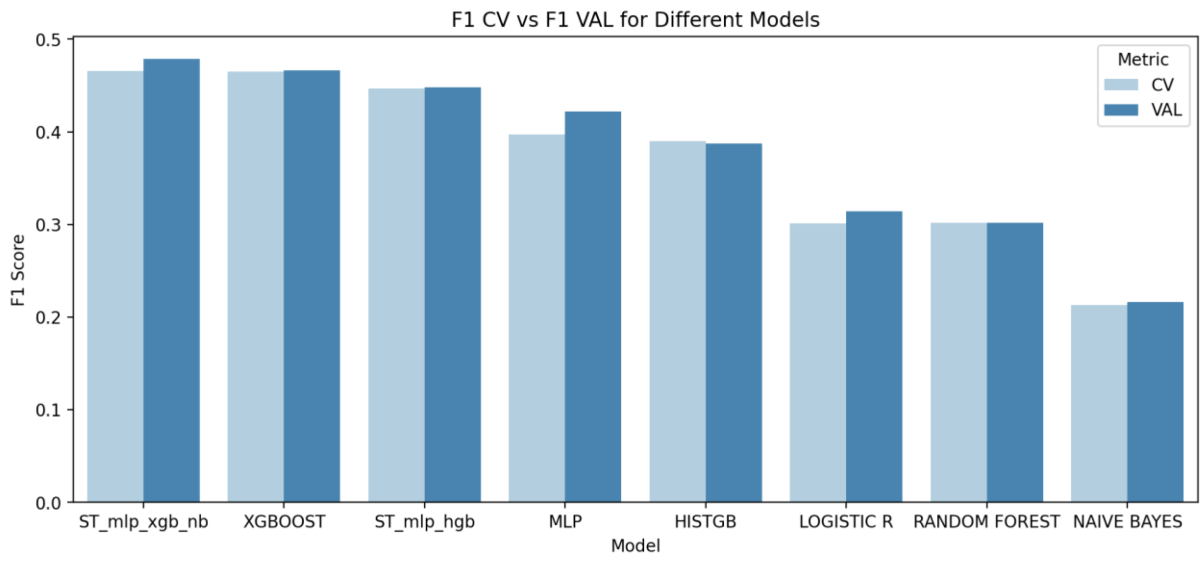


Figure 12- Models Performance in Cross Validation vs Validation using Macro F1

XGBoost works by iteratively **fitting decision trees** on the **residuals of previous trees**, **continuously improving its predictions**. This iterative process allows it to be **fine-tuned for optimal performance**.

So, to tune the parameters, we did a **Grid Search** for some of the most **important parameters** that could **help** to **control overfitting** and **address class imbalanced**, such as, (n_estimators, max_depth, scale_pos_weight, max_delta_step, reg_lambda, learning_rate and reg_alpha).

We obtained a good **precision** (0.573) and a not so good **recall** (0.447), where we see that model **struggles** the most on **predicting** "**Permanent Total Disability**". see annexes (Fig 13)

In the case of **MLP,** which is a type of neural network that consists of **multiple layers of neurons**, including an input layer, **one or more hidden layers**, and an **output layer**, is also particularly well-suited for **complex problems**. This Neural Network model performed great although when we're using one hot encoding, the model seems to perfume a little better.

Regarding parameters, we used **fewer neurons** than the **number of features** because it acts as a form of **regularization**, **reducing** the **risk** of **overfitting**, a **low learning rate** to assure model weights are updated slowly, "Adam" **optimizer** and "ReLU" for **activation function**.

For **Logistic Regression**, we adjusted the class weights to give more importance to the minority classes. **Multinomial** and **OneVsRest** were considered.

For **Random Forest** and **HistGradientBoost**, similar parameters of XGBoost were applied, such as max depth, min samples split, max features used at each split, class weight, max leaf nodes and others. This helped the models to not overfit. See example of in annexes (Fig 14)

Regarding our **multiclass classification**, we also **explore** different **stacking** between models. Our goal was to **maximize** both **precision** and **recall**. This was made after testing individual models. We also wanted to **stack different types of models**, (Stack RFC with HISTGB/XGBoost **was not an option**).

Two different stack models were created:

1. Stack of **MLP Classifier** (High precision) with **HistGradientBoost** (high recall).
2. Stack of **2 strong models (XGB and MLP)** with a considered low performance model individually in this scenario, **Naïve Bayes**, however, it seemed that the model was doing **well regarding recall**, so it was **worth** the shot.

Results:

- The stack of **MLP** with **HGB increased** the Macro F1 Score to **0.448** on validation set. The models alone had previously **0.422 and 0.388**, respectively.
- The stack of **XGBoost & MLP** with **Naive Bayes** gave us an **improvement** of XGBoost alone, **from 0.465 to 0.479** on validation set. see annexes (Fig 15)

5. OPEN-ENDED SECTION

Due to the initial insights obtained through visualizations between **Agreement Reached** and **Claim Injury Type** (main target), we decided to do a **binary classification** task aimed to predict the **likelihood** of achieving an **agreement** without the intervention of the WCB as we believed this feature would **help** to **predict Claim Injury Type.**

To prepare the data for model training, the **same preprocessing steps applied to Multiclass problem**, were implemented (except for target variable, of course). We didn´t use Claim Injury Type to predict these.

In this case, and despite using different methods of feature selection, we opted to use **purely** the **RFECV Logistic Regression**, since were going to use that **model** for **evaluation** of binary classification task.

Regarding some parameters, "**Class weight**" was balanced to address the **class imbalance**. We try to use **over/under sampling** techniques such as SMOTE, Random Oversampling and others, however, to **maintain consistency** in our data and don´t get artificial or repeated samples, in the end we didn´t use it. To ensure a fair representation of the target classes across training and validation datasets, Stratified K-Fold Cross-Validation with five splits was applied.

The model's performance was evaluated using several metrics, with the **F1 Score (Binary)** being the primary evaluation criterion, given the **imbalanced** nature of the classification task.

Precision and **Recall** were also assessed to measure the accuracy of positive predictions and the coverage of true positives. To enhance **performance** on the **minority class (1),** a **threshold** was set to **maximize recall** for that class while maintaining precision. Higher recall is **crucial** when it is important to **identify as many positive instances as possible**, even if it means including some false positives. See annexes (Fig 16)

Finally, a **confusion matrix** was created, and we observe that the model doesn´t perform well on predicting minority class (1), since it fails 6280 and only get right 3095. See annexes (Fig 17)

6. CONCLUSION

Now that we have reached this point, we can confidently say that our goals were achieved, regardless of the F1 Macro Score. We thoroughly understood and prepared our data and problem, comprehending the features and striving to maximize their potential. Our main challenge was predicting some of the minority classes. The models we created struggled to differentiate between them. For instance, our model couldn't accurately predict occurrences of “Permanent Total Disability”, possibly due to its similarity to the "Death" category, leading to misclassification.

To improve future models, techniques such as oversampling with strategies to avoid overfitting could be beneficial. This would ensure that minority classes are better represented and more accurately predicted.

As mentioned in the Introduction, many industries such as healthcare, insurance, credit fraud detection, and criminal behavior analysis have successfully utilized classification models to optimize the processing of customer data. This demonstrates the transformative potential of data-driven decision-making in complex systems. In this work, despite not using all the necessary tools or having access to all available information to achieve an excellent score, we validate our point that it can be done. While it is more challenging to classify certain individual classes, we can consider them as aggregate classes. For example, attributes may indicate no compensation, or low, medium, or high severity.

Regarding Categorical features and their high cardinality, we think that embedding vectors could improve our encoders.

Future work could also incorporate doing all in a pipeline to prevent data leakage and to get the best results.

7. REFERENCES

Assembled workers' compensation claims, beginning 2011. (n.d.). *New York State Open Data*. https://data.ny.gov/Government-Finance/Assembled-Workers-Compensation-Claims-Beginning-20/jshw-gkgu/about_data

Employee disability benefits. (n.d.). *New York State Workers' Compensation Board*. https://www.wcb.ny.gov/content/main/DisabilityBenefits/employee-disability-benefits.jsp

Ensemble methods. (n.d.). *scikit-learn*. https://scikit-learn.org/stable/modules/ensemble.html

XGBoost Parameters: XGBoost Parameters — xgboost 2.1.3 documentation & XGBoost Part 2 (of 4): Classification

How to Configure XGBoost for Imbalanced Classification - MachineLearningMastery.com

Encoders: A Benchmark and Taxonomy of Categorical Encoders | by Vadim Arzamasov | Towards Data Science

Handling Categorical Data in Python Tutorial. (n.d.). Www.datacamp.com.

https://www.datacamp.com/tutorial/categorical-data

Metrics: sklearn.metrics — scikit-learn 1.6.0 documentation

Feature Importance: 4.2. Permutation feature importance — scikit-learn 1.6.0 documentation

Multiclass: sklearn.multiclass — scikit-learn 1.6.0 documentation

Groups: Find your classification unit, industry, or rate - WorkSafeBC

Grid Search: GridSearchCV — scikit-learn 1.6.0 documentation

We also used ChatGPT and Copilot in Microsoft EDGE to check for inconsistencies when we were creating function, creating subplots, making sure the titles were ok.

Practical Classes Notebooks NOVA IMS 2024/2025: by Ricardo Santos & Leon Debatin

8. ANNEXES

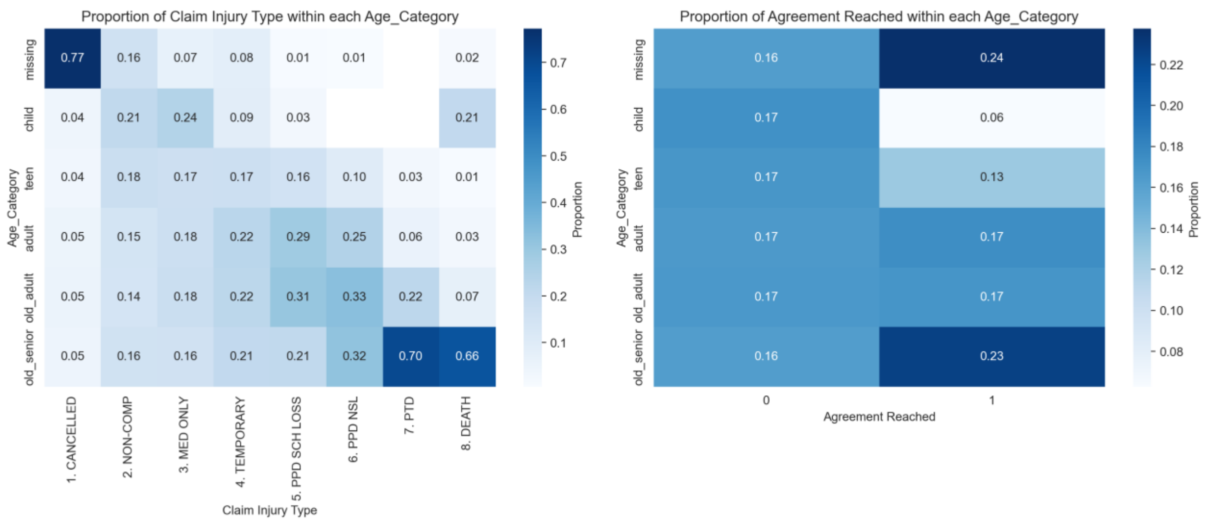


Fig 1 – Proportion of target variables within each Age Category

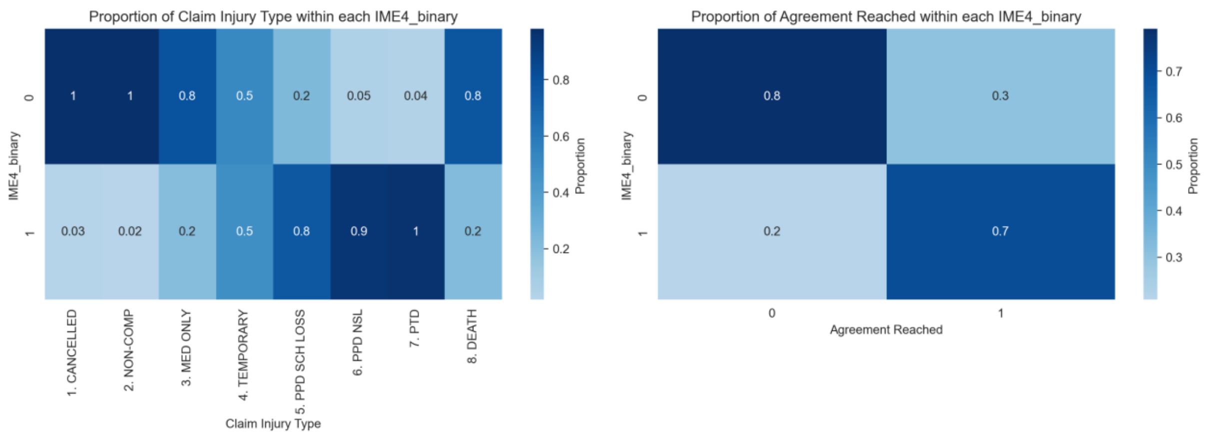


Fig 2 – Proportion of target variables within each IME4 Binary

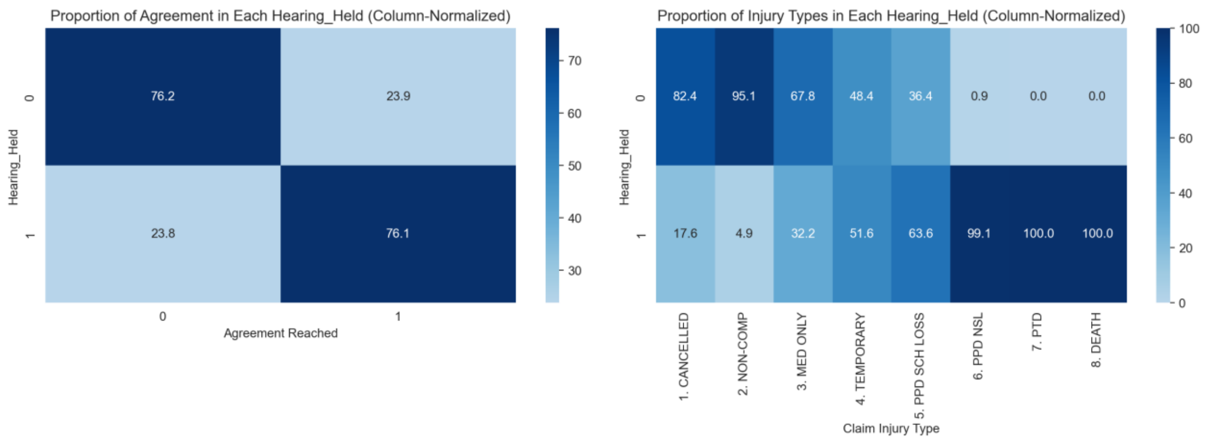


Fig 3 – Proportion of target variables in each Hearing Held

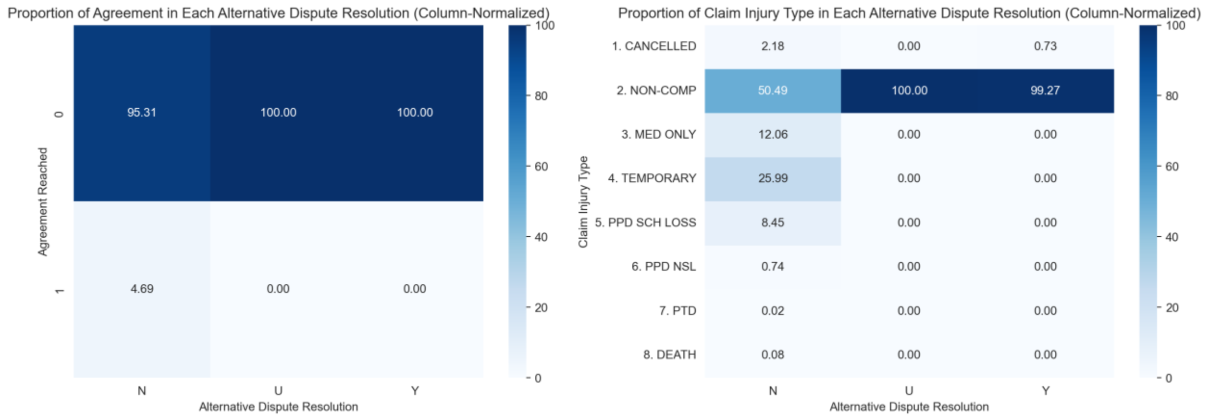


Fig 4 – Proportion of target variables in each Alternative dispute resolution

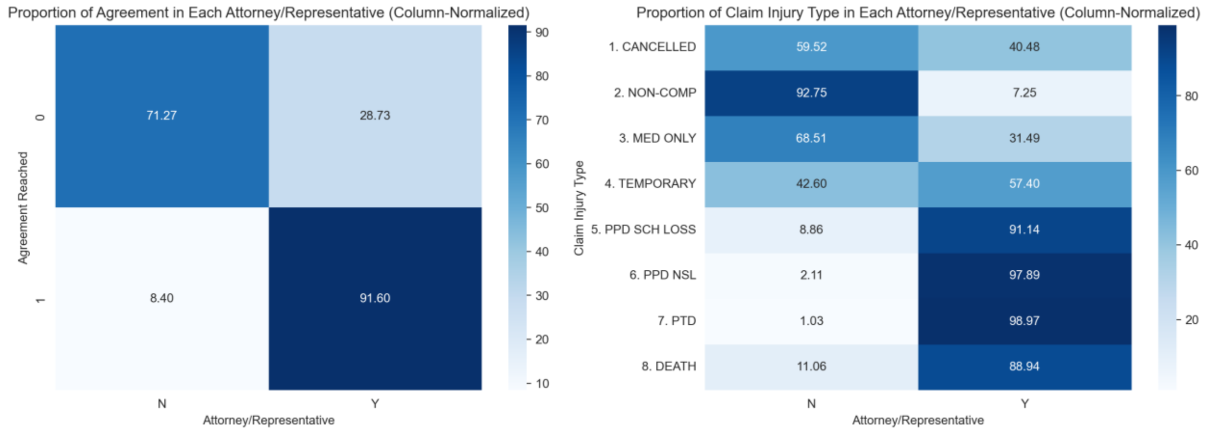


Fig 5 – Proportion of target variables in having or not Attorney

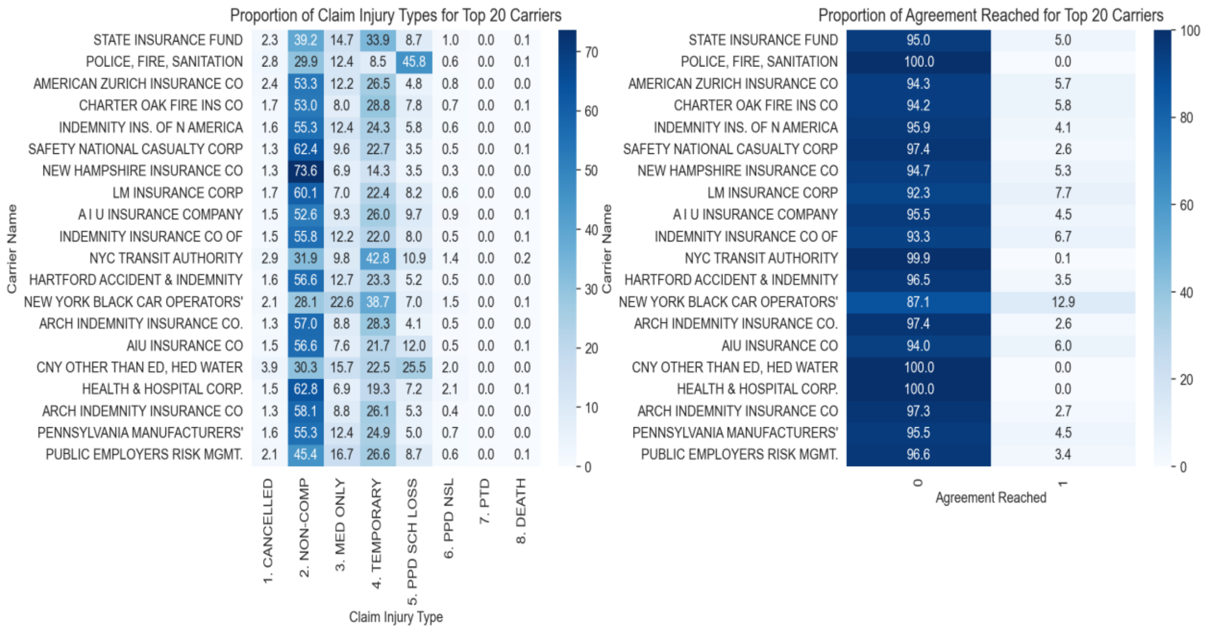


Fig 6 – Proportion of target variables in top 20 carriers

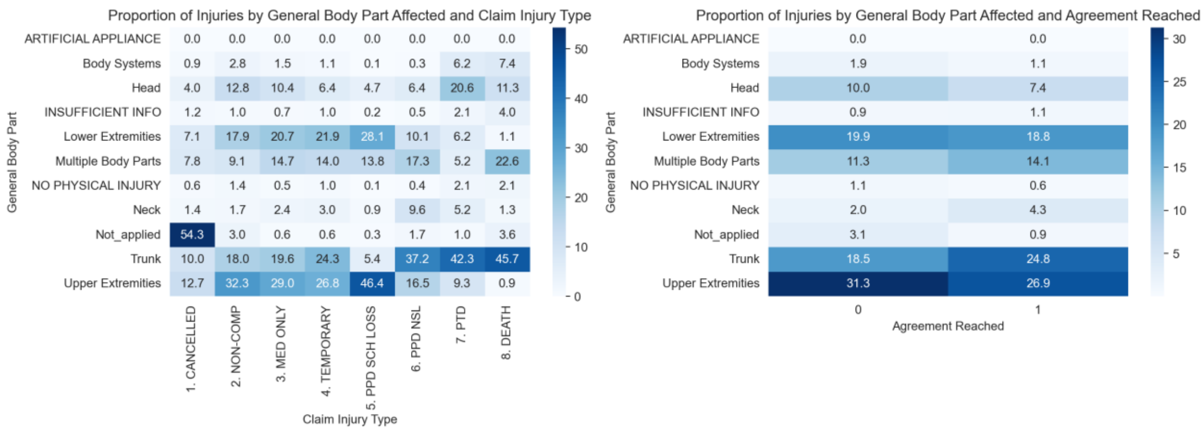


Fig 7 – Proportion of target variables by general body part

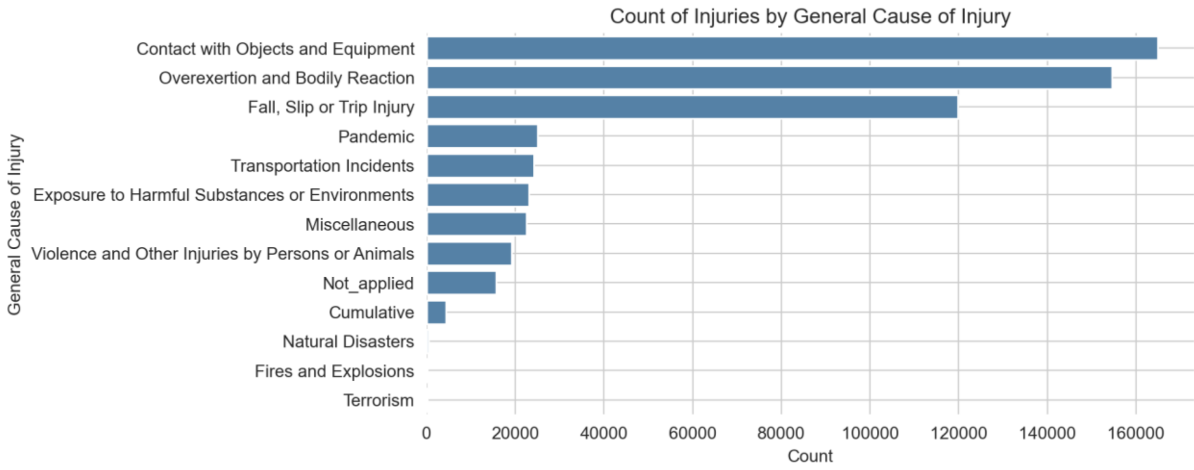


Fig 8 – Frequency of grouped cause of injuries

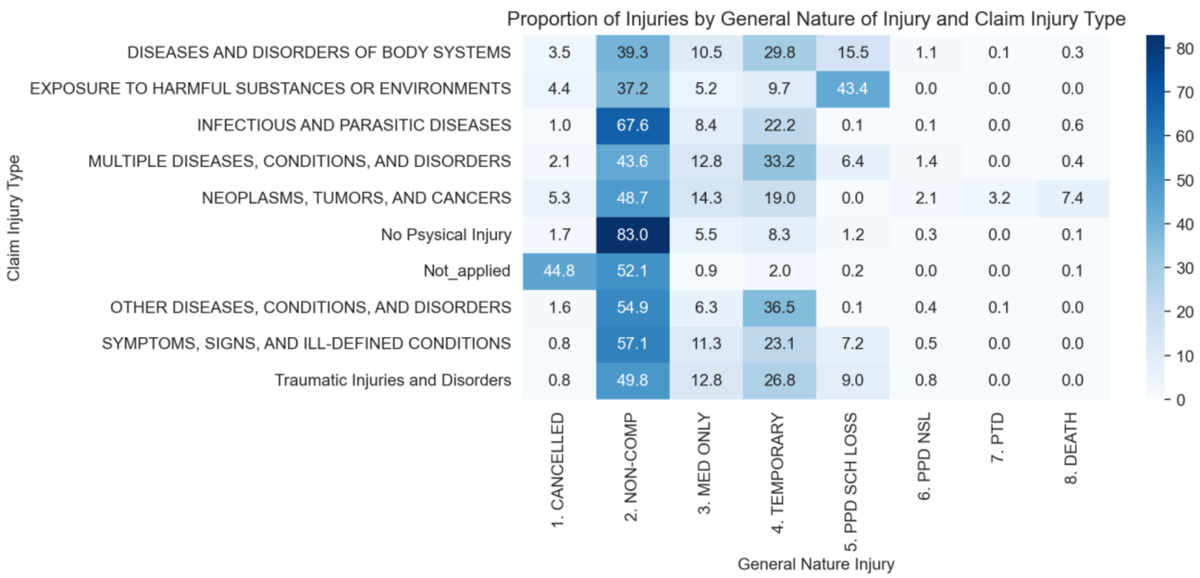


Fig 9 – Proportion of each grouped nature by Claim Type.

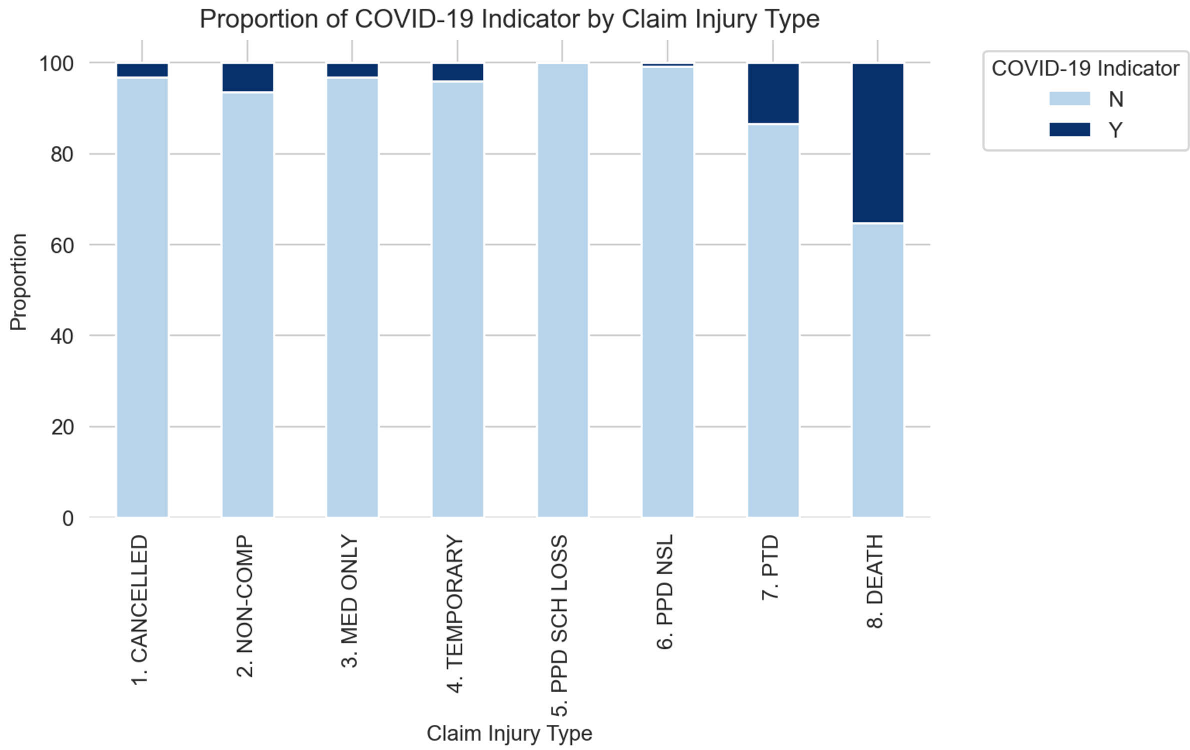


Fig 10 – Proportion of Covid 19 by Claim Type.

	Feature	Lasso	Ridge	Mutual_Info	Highly_Corr	Importance_Logistic
0	County of Injury	Discard	Discard	Keep	Discard	Keep
1	Number of Dependents	Discard	Discard	Discard	Keep	Keep
2	9_11_period	Discard	Keep	Discard	Keep	Discard
3	dependents_binary	Discard	Discard	Keep	Keep	Keep
4	Carrier_binary	Discard	Keep	Discard	Keep	Discard
5	Accident_Day	Keep	Discard	Discard	Keep	Discard
6	Accident_DOW	Keep	Discard	Keep	Keep	Discard
7	Age at Injury	Keep	Discard	Keep	Discard	Keep
8	Carrier Name	Keep	Discard	Keep	Keep	Keep
9	WCIO Part Of Body Code	Keep	Discard	Keep	Keep	Keep
10	general_industry	Keep	Discard	Keep	Keep	Discard
11	Age_Category	Keep	Discard	Keep	Discard	Discard
12	Gender	Keep	Discard	Keep	Keep	Discard
13	Industry Code	Keep	Discard	Keep	Keep	Discard
14	Carrier Type	Keep	Discard	Keep	Keep	Keep
15	Alternative Dispute Resolution	Keep	Keep	Discard	Keep	Keep
16	Accident_Month	Keep	Keep	Discard	Keep	Keep
17	Accident_Year	Keep	Keep	Keep	Discard	Discard
18	c2_to_accident_delay	Keep	Keep	Keep	Discard	Keep
19	Assembly_year	Keep	Keep	Keep	Discard	Discard
20	C3_Received	Keep	Keep	Keep	Discard	Keep
21	c3_to_accident_delay_category	Keep	Keep	Keep	Discard	Keep
22	IME-4 Count	Keep	Keep	Keep	Discard	Keep
23	IME4_binary	Keep	Keep	Keep	Discard	Keep
24	assembly_delay	Keep	Keep	Keep	Discard	Keep
25	Home_county	Keep	Keep	Keep	Discard	Discard
26	c2_to_accident_delay_category	Keep	Keep	Keep	Discard	Keep
27	District Name	Keep	Keep	Keep	Keep	Discard
28	Accident_date_present	Keep	Keep	Keep	Keep	Discard

Fig 11 – Feature Selection for Multiclass.

```
Macro F1 Score: 0.4660
Log Loss: 0.5003

Classification Report:
              precision    recall  f1-score   support

         0.0      0.746     0.528     0.619      3743
         1.0      0.863     0.980     0.918     87324
         2.0      0.584     0.102     0.174     20672
         3.0      0.767     0.903     0.829     44552
         4.0      0.699     0.686     0.692     14484
         5.0      0.246     0.013     0.024      1263
         6.0      0.000     0.000     0.000        29
         7.0      0.680     0.362     0.472       141

    accuracy                          0.812    172208
   macro avg      0.573     0.447     0.466    172208
weighted avg      0.784     0.812     0.773    172208
```

Fig 12 – Classification Report for Multiclass.

```
#hgb_low_depth = HistGradientBoostingClassifier(max_depth = 5).fit(X_train_selected, y_train)
#hgb_medium_dept = HistGradientBoostingClassifier(max_depth = 8).fit(X_train_selected, y_train)

#df = pd.DataFrame(columns = ['Time','Train','Test'], index = ['low','medium'])
#show_results_rfc(df, hgb_low_depth, hgb_medium_dept)
```

	Time	Train	Test
low	36.009+/-13.69	0.421+/-0.02	0.403+/-0.01
medium	28.196+/-10.83	0.435+/-0.02	0.415+/-0.01

Fig 13 – HISTGB max depth choice.

```
Macro F1 Score: 0.4786
Log Loss: 0.5272

Classification Report:
              precision    recall  f1-score   support

         0.0      0.763     0.508     0.610      3743
         1.0      0.866     0.976     0.918     87324
         2.0      0.502     0.139     0.217     20672
         3.0      0.771     0.893     0.827     44552
         4.0      0.709     0.672     0.690     14484
         5.0      0.297     0.050     0.085      1263
         6.0      0.000     0.000     0.000        29
         7.0      0.718     0.362     0.481       141

    accuracy                          0.811    172208
   macro avg      0.578     0.450     0.479    172208
weighted avg      0.778     0.811     0.778    172208
```

Figure 15 – Classification Report of stack model (XGB, MLP & Bayes)

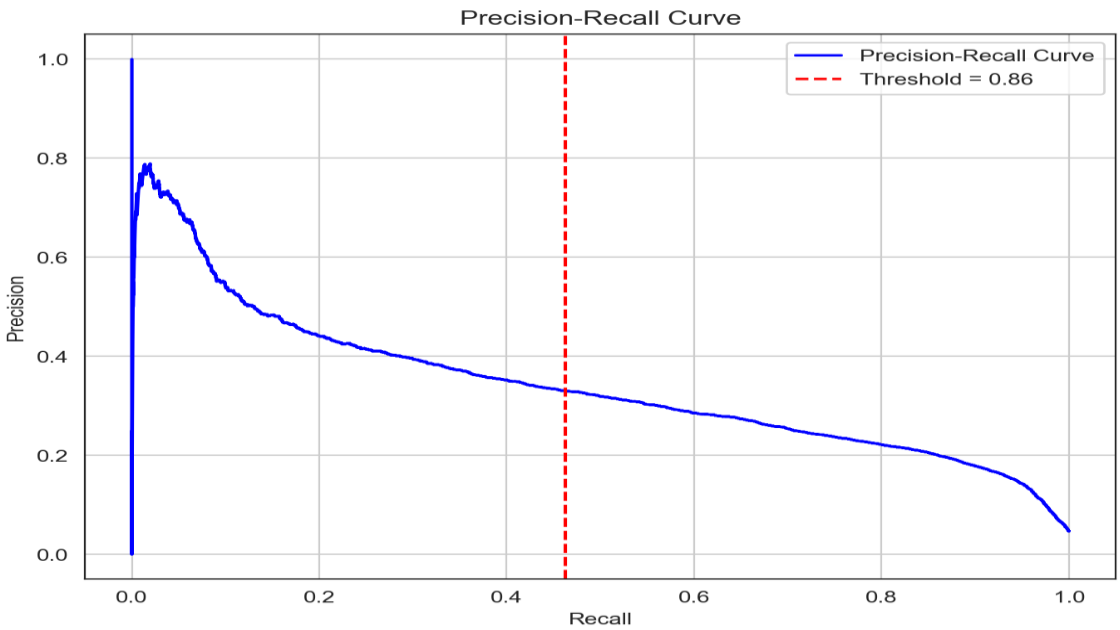


Figure 16 - Precision – Recall Curve for Binary Classification

	Predicted Class 0	**Predicted Class 1**
Actual Class 0	130,530	6,280
Actual Class 1	3,602	3,095

Table 2 - Confusion Matrix for Binary Classification