National College of Ireland

Configuration Manual

MSc Research Project
Data Analytics

Trinita John Peter
Student ID: X23266848

School of Computing
National College of Ireland

Supervisor: Vikas Tomer

National College of Ireland

MSc Project Submission Sheet

School of Computing

Student Name:	Trinita John Peter		
Student ID:	X23266848		
Programme:	Data Analytics	**Year:**	2024-2025
Module:	MSc (Research) Practicum		
Lecturer:	Vikas Tomer		
Submission Due Date:	11/08/2025		
Project Title:	Optimizing Actuarial Risk Stratification and Equitable Premium Allocation in Health Insurance companies via Supervised Ensemble Learning		
Word Count: 1560	**Page Count: 3**		

I hereby certify that the information contained in this (my submission) is information pertaining to research I conducted for this project. All information other than my own contribution will be fully referenced and listed in the relevant bibliography section at the rear of the project.
ALL internet material must be referenced in the bibliography section. Students are required to use the Referencing Standard specified in the report template. To use other author's written or electronic work is illegal (plagiarism) and may result in disciplinary action.

Signature: Trinita John Peter

Date: 10/08/2025

PLEASE READ THE FOLLOWING INSTRUCTIONS AND CHECKLIST

Attach a completed copy of this sheet to each project (including multiple copies)	□
Attach a Moodle submission receipt of the online project submission, to each project (including multiple copies).	□
You must ensure that you retain a HARD COPY of the project, both for your own reference and in case a project is lost or mislaid. It is not sufficient to keep a copy on computer.	□

Assignments that are submitted to the Programme Coordinator Office must be placed into the assignment box located outside the office.

Office Use Only	
Signature:	
Date:	
Penalty Applied (if applicable):	

Configuration Manual

Trinita John Peter
Student ID: x23266848

The purpose of this configuration manual for research is to facilitate the viewer with providing essential steps in running the solution.

- The coding environment is Jupyter notebook for ease of execution, visualizing data and documentation of the workflow. But it also supports other Python based editing environments.
- Creating environment variables is considered a good code practice to isolate current project's dependencies that might later overlap and cause version issues.
- The dataset can be downloaded to local from Kaggle under "enhanced_health_insurance_claims.csv".
- Required packages include pandas, numpy, sklearn, matplotlib, seaborn, imblearn, metrics, columntransformer, simple imputer, fastml.
- Concise summary of the attributes is obtained by using `fastml` library, time feature conversion is done using pandas datetime function and derived time features are added using dt.month, dt.year attributes.
- Initial data quality assessment is performed that includes checking for nulls, duplicates, data types, distributions on the health variables.
- EDA visualizations are plotted - histogram for distribution of claim amounts, boxplot on provider specialty, violin plot distribution for claim amount by gender, barplot for claim type, amount and status. Pearson correlation matrix and heatmap are computed.
- Feature engineering is performed for the following:
    - Age bins, income bins using pandas.qcut() to divide a continuous variable (PatientIncome) into quartile-based categories.
    - Flags claims that are greater than the 75th percentile
    - Create combination feature using concatenation (ClaimType + ClaimSubmissionMethod).
- One hot Encoding for numerical and categorical features are performed.
- SMOTE is used to train the model to overcome class imbalance, improve recall and precision for the minority class, prevent bias.
- The target of train and test data is set to "IsHighClaim" and the rest of the attributes are independent features.
- Column transformer typically processes encoding of numerical and categorical, with models "RandomForestClassifier", "GradientBoostingClassifier" and XGBoost. The number of boosting stages which are the decision trees is set to `n_estimators=100` with `random_state` set to 42 for reproducibility.
- Hyperparameter tuning method ` RandomizedSearchCV` is applied on data with parameters: ` n_iter=20` implying the model needs to itern for 20 combinations and ` cv=3` to obtain a 3-fold cross-validation that splits the dataset into 3 equal parts and final score is the average of the 3.
- Threshold tuning with custom value 0.3 is set, to flag high claim amount especially for cost sensitive applications.
- Logarithmic transformation is applied on skewed data for the XGBoosting Classifier Model.
- Classification report and ROC AU curve imply this model outperforms and can be highly reliable for risk assessment, transparent policy pricing and building trust between policyholders and insurers.

Detailed explanation of code

1. Primarily, all libraries and modules essential to solve the research question are imported and installed accordingly.
2. The data in imported into Jupyter notebook from local file location and a copy of it is made, on which all steps are performed.
3. The time column is converted into datetime64 nanosecond for uniformity, from which the month and year attributes are extracted to facilitate feature engineering.
4. EDA is carried out in order to understand the distribution of data across various variables.
    a. Histogram for distribution of claim amount
    b. Boxplot to detect outlier
    c. Violin plot to see the density of data values
    d. Barplot with hue for distribution of claim amount by its type and status as opposed to amount
5. To compute the correlation between numerical variables, pearson matrix is calculated, that revealed no such relationship exists within the variables.
6. Time based features are extracted in order to provide the model with as many information as possible to facilitate accurate predictions.
7. Ages : 0, 18, 35, 60, 100 are labelled as the following categories : 'Child', 'Youth', 'Adult', 'Senior' to ease binary classification. Similarly, the income is binned into the following category: 'Low', 'Medium', 'High', 'Very High'.
8. Threshold tuning is introduced in the next step to flag high claims in such a way that the model captures high outlier. This is performed by a simple calculation that creates a new column if the data value is greater than 75 %.
9. The Claim interaction feature is created by concatenating claim type and claim submission method.
10. Post feature engineering techniques, redundant or non-relevant columns are dropped.
11. Essential steps are performed in the next cells for model tuning, such as defining feature and target variables, in this case is integer values of “IsHighClaim” column.
12. Numerical and categorical features are identified as part of the preprocessing steps. Multiple parameters such as encoded features, numerical attributes with categorical values are passed into a Column transformer for parallel processing.
13. In the next step, the models that are considered are initialised, which is random forest classifier and gradient boosting. Dataset is split into train and test data with a fixed random state of 42 for reproducibility, which are then passed into a pipeline.
14. Model is then fitted on the test and train data. Predictions are carried out on test data, which is compared against the values of y test.
15. A classification report is provided along with confusion matrix, that reveals that the model is not at its best performance. Hence further enhancements are made.
16. Randomized search CV is implemented for both the above initialised models that takes into account the best possible set of combinations to act as predicting factors.
17. This hyperparameter tuning technique has also not proved to be quite effective, in which case, the count of both the classes are checked.
18. Due to this class imbalance, SMOTE is introduced before modelling the data. While the accuracy and precision metrics have improved, state of the art research implies there is more room for improvement by using other enhancements.
19. A custom threshold of 0.3 is set to optimize the results on cost sensitive applications.
20. Following are the conclusions that could be drawn from the results:
    a. Class 1 Recall: 37% — model misses 63% of high-risk cases

b. Class 1 Precision: 23% — most predicted positives are wrong. This model does not meet the goal because:
    - ➔ It fails to reliably identify high-risk patients (Class 1).
    - ➔ It misclassifies a large number of cases, leading to poor resource allocation.
    - ➔ It offers low predictive confidence, which undermines fair policy pricing.
c. Hence this model is not effective for the research which does not help Identify high-risk individuals for targeted resource planning and performance.

21. XGBoost, an ensemble method, handles missing values and class imbalances effectively. This statement is backed up by numerous literature research papers survey carried out initially.
22. Applying logarithmic transformations on the target data can produce accurate predictions on classification tasks. By making the largely spread data, skewed, the model learns easily the hidden pattern in the values.
23. A pipeline with XGBoost is built and class imbalance is manually handled by the formula: scale = negative / positive, where negative = sum(y_train == 0) class 0- not high claim and positive = sum(y_train == 1) class 1- Ishighclaim.
24. Following are the conclusions that are drawn from the classification report, ROC AUC curve and confusion matrix:
    a. Well Predictive Performance- correctly identifies almost every high-risk and low-risk patient, which is crucial for insurance risk modeling.
    b. Accurate Risk Anticipation:
        - ➔ With only 1 misclassified case out of 900, the model demonstrates exceptional ability to anticipate risk.
        - ➔ This supports resourceful asset allocation by flagging high-risk individuals with confidence.
    c. Fair and Transparent Policy Pricing:
        - ➔ High precision and recall mean fairer rates can be offered to policyholders based on accurate risk profiles.
    d. Reliable model evaluation:
        - ➔ used confusion matrix, classification report, and ROC AUC—a comprehensive evaluation strategy.
        - ➔ These metrics confirm that model is not just accurate, but also balanced and trustworthy.