Fraud Detection in Health Insurance Claims – A Machine Learning (ML) Approach

June 25, 2021

Claims Fraud Detection using XGBoost

Pritha Datta
Assistant Vice President
ManipalCigna Health Insurance Company

Institute of Actuaries of India

Fraud Detection in Health Insurance Claims: Bridging the Gap

Institute of Actuaries of India

Situation – Current State

- There are a **variety of fraud patterns**:
  - Fraud by **healthcare providers**
  - Fraud by **Insurance subscribers**
  - **Conspiracy frauds** or nexus of providers, customers and distribution channels
- Rule-based and manual fraud detection approach results is a lot of **false investigations**

Complications – The Gap / Trigger

- Incidence of frauds is significantly less than the total number of claims – **class imbalance**
- Ever evolving nature of fraudulent claims
- Costs of the two types of classification errors (FP* and FN*) are not the same

Questions – before we start

- Is there adequate data, i.e. **data depth**?
- Is the data clean and usable, i.e. **data quality**?
- Data system sophistication and preparedness

Desired Future State

- We are able to detect 100% of the fraudulent claims
- We are able to **minimize the incorrect fraud classifications** – i.e. minimize both FP* and FN*

***Note : FP – False Positive | FN – False Negative**

Key Machine Learning Concepts

Machine Learning vs. Rule-Based Systems in Fraud Detection

Institute of Actuaries of India

Rule-based fraud detection	ML-based fraud detection
Catching obvious fraudulent scenarios	Finding hidden and implicit correlations in data
Requires much manual work to enumerate all possible detection rules	Automatic detection of possible fraud scenarios
Multiple verification steps that harm user experience	The reduced number of verification measures
Long-term processing	Real-time processing

Figure 1 : Comparison of Rule-based and ML-based fraud detection

There are **two types of ML approaches** that are commonly used – both independently or combined:

- **Supervised ML** : training an algorithm on labeled historical data i.e. where you have an input (X) and output (Y) variable. Goal is to learn the mapping function from X to Y i.e. Y = f(X), and use the same to predict the output variables of a new input dataset
  - Supervised learning problems can be further grouped into **regression** and **classification problems**

- **Unsupervised ML** : processing unlabeled data i.e. where you only have input data (X) and no corresponding output variables. Goal for unsupervised learning is to model the underlying structure or distribution in the data in order to learn more about the data
  - Unsupervised learning problems can be further grouped into **clustering** and **association problems**

Supervised Learning : Classification Algorithm

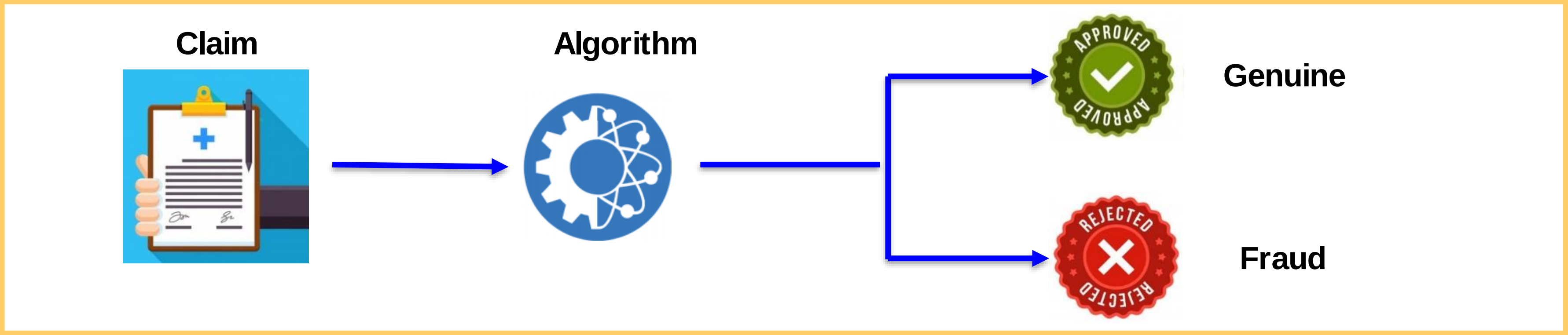


Figure 2 : Diagrammatic representation of a binary classification algorithm

Classification predictive modeling is the task of approximating a mapping function from input variables to **discrete** output variables – Male or Female, True or False, Fraud or Genuine, etc.

Types of Classification:

- **Binary Classification**: Classification task with two possible outcomes
- **Multi-class classification**: Classification with more than two classes
- **Multi-label classification**: Classification task where each sample is mapped to a set of target labels

Types of Classification Algorithm:

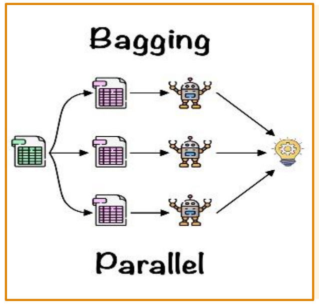
- Logistic Regression
- Naïve Bayes classifier
- Support Vector machines
- K-nearest Neighbour
- Decision Tree

Figure 2*

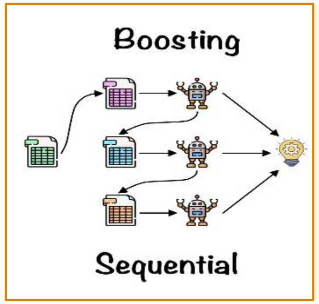
Ensemble Learning : Aggregating Weak Learners

Ensemble learning is a machine learning method where multiple models (often called “weak learners”) are trained to solve the same problem and combined to get better results.

The main hypothesis is that when weak models are correctly combined we can obtain more accurate and/or robust models.
Three major kinds of meta-algorithms that aims at combining weak learners:



Bagging
Considers **homogeneous** weak learners, learns them **independently** in parallel and combines them following a deterministic averaging process



Boosting
Considers **homogeneous** weak learners, learns them **sequentially** and combines them following a deterministic strategy

Stacking
Considers **heterogeneous** weak learners, learns them in parallel and combines them by training a meta-model to output a prediction based on the different weak models predictions

Tree-Based Models : Decision Tree and Ensemble Trees

Tree-based models use a series of if-then rules to generate predictions from one or more **decision trees**.

Advantages:
- Straightforward interpretation
- Good at handling complex, non-linear relationships

Disadvantages:
- Predictions tend to be weak, as singular decision tree models are prone to overfitting
- Unstable, as a slight change in the input dataset can greatly impact the final results

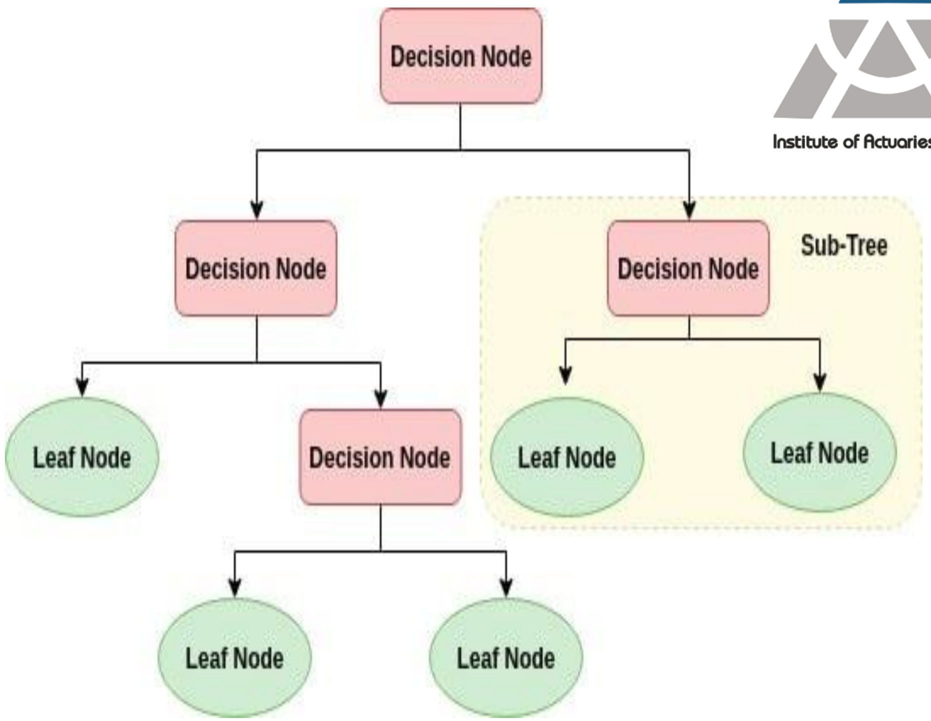


Figure 3 : Visualizing a Decision Tree

Ensemble algorithms that utilize decision trees as weak learners have multiple advantages:

- Easy to understand and visualize
- Can handle mixed data types
- Account for multi-collinearity
- Better at handling outliers and noise
- Non-parametric, no specific distribution
- Can handle unbalanced and large data
- Do not tend to overfit
- Computationally inexpensive

Case Study : Claims Fraud Detection using XGBoost

Advanced Fraud Detection : How to Build a Robust System?

Institute of Actuaries of India

Labeling Data

It is hard to manually classify new and sophisticated fraud attempts by their implicit similarities. It is thus essential to **apply unsupervised learning models to segment data items into clusters to unearth hidden patterns** such as a nexus between hospital and agents, certain fraud prone locations or just cleaning data and identifying outliers.

Techniques : K-means clustering, Association Mining, Text Mining

Training Supervised Model

Once the data is labeled, it captures not only the proven past fraud/non-fraud items, but also suspicious patterns and nexuses. The next step is to **use the labeled dataset to train supervised models** that will be used to detect fraudulent transactions in the future.

Techniques : Logistic Regression, Decision Tree, Random Forest, XGBoost – to name a few.

Ensembling : To make predictions more accurate it is advisable to build multiple models using the same method or combine entirely different methods. It **leverages the strengths of multiple different methods** and provides the most precise output.

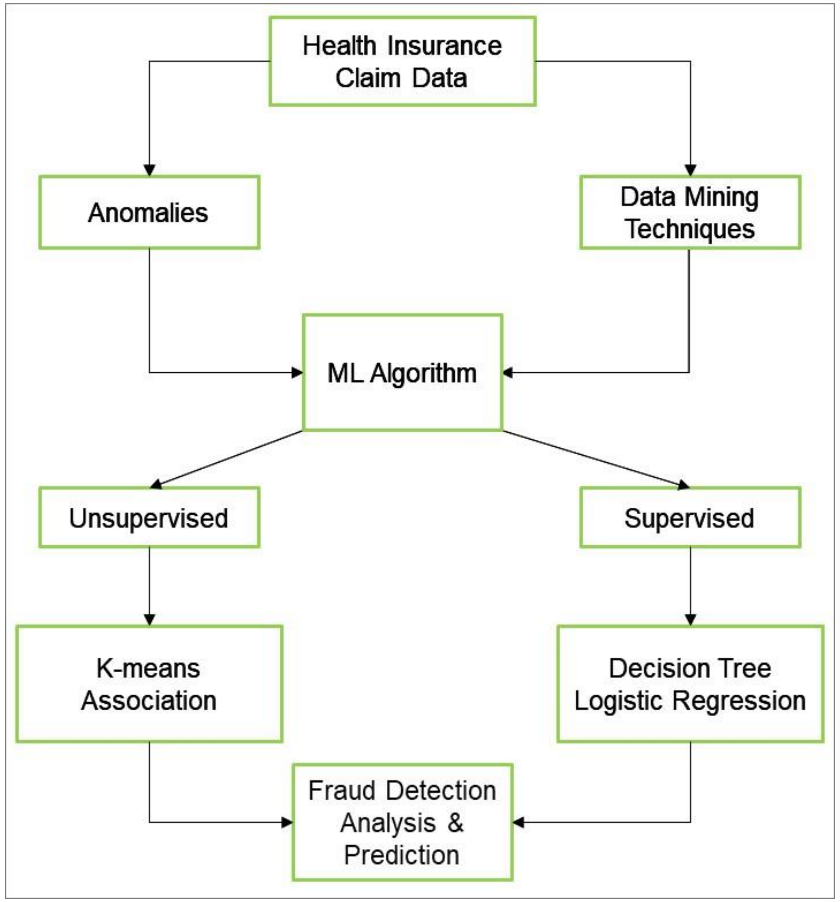


Figure 4 : Diagrammatic representation of an advanced fraud detection process

Model Building and Comparison

Step 1 : Data Preparation

Types of data :
Claims Data | Policy Information | Customer Demographics | Provider Information | Distribution Channel Information

Claim No	Fraud	Disease Group	Member_Gender	Age_Band	…
claim_34669	0	RESPIRATORY	Female	0 - 17	...
claim_5894	0	INFECTIOUS	Female	26-35	...
claim_23443	1	RESPIRATORY	Female	0 - 17	...
claim_68392	0	INFECTIOUS	Male	26-35	...

Figure 5 : Excerpt from the data matrix for XGBoost

Data Cleaning & Standardization : includes outlier treatments, missing value treatments and approaches like text mining

Exploratory Data Analysis : to identify existing data patterns and anomalies

Feature Engineering : process of transforming raw data into features that better represent the underlying problem to the predictive models, resulting in improved model performance on unseen data

Step 2 : Model Development

- Divide dataset into **training data (70%)** and **test data (30%)** in a statistically random manner
- Based on the initial model performance, different features are engineered and re-tested
- In order to improve model performance, the parameters that affect the performance are tweaked and re-tested
- Identify the “best” algorithm using model diagnostics – **XGBoost** in this case
- **Use XGBoost algorithm to create a model to predict fraudulent claims**

Interpreting the Model : Output & Threshold Selection

Model Output

- The model provides a measure of the certainty or uncertainty of a prediction – **propensity score**

- This score is converted into a class label, governed by a parameter known as the **decision threshold** - 0.5 is the default for normalized predicted probabilities

- Along with propensity scores, the model provides a relative **importance matrix** – containing the most relevant drivers for our model

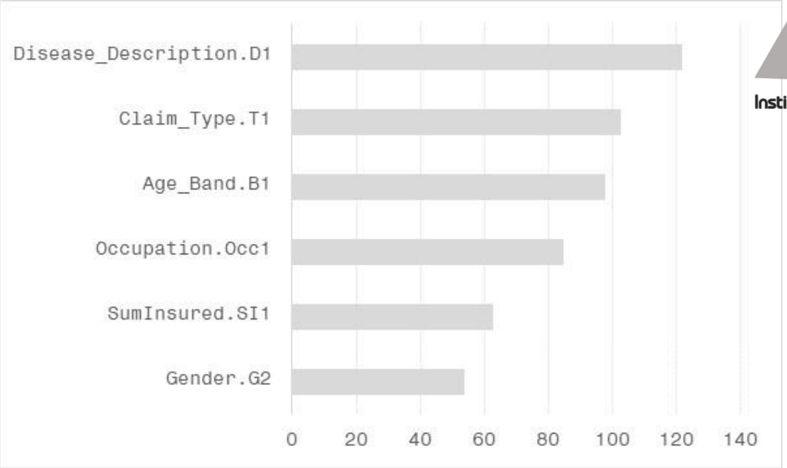


Figure 6 : XGBoost Feature Importance Bar Chart

Threshold Selection:

For a binary classification problem with class labels 0 and 1:

- Prediction < 0.5 = Class 0
- Prediction >= 0.5 = Class 1

Default threshold may not represent an optimal interpretation, due to:

- The class imbalance in data
- The cost of one type of misclassification is more than another type of misclassification

Interpreting the Model : Performance Criterion

Model Performance Criterion:

- **ROC** is a method of visualizing classification quality, which shows the dependency between TPR* and FPR* at different thresholds

- For each threshold we obtain a (TPR, FPR) pair, which corresponds to one point on the ROC curve

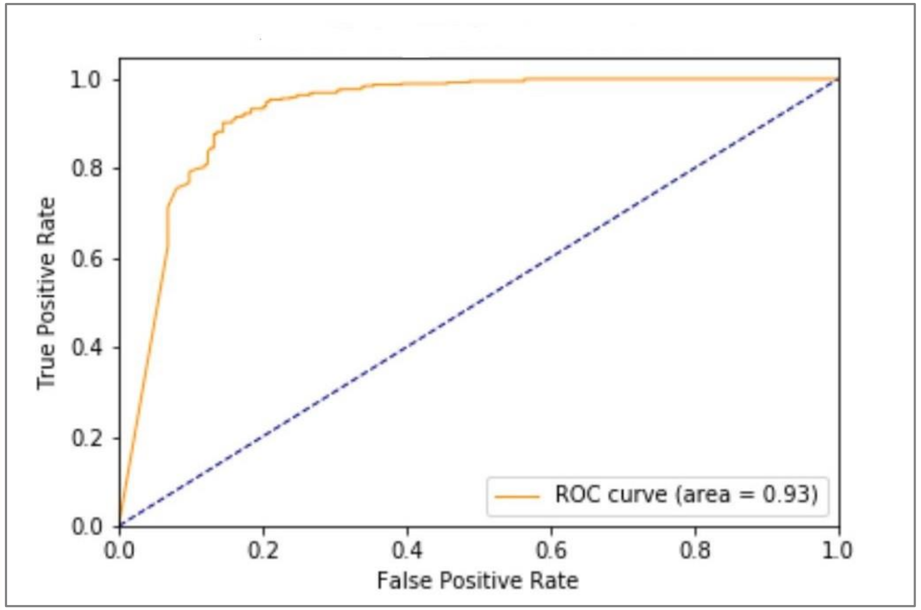


Figure 7 : TPR vs FPR represented as ROC to determine AUC

Predicted		Actual	
		Normal	Fraud
	Normal	True Negatives (TN)	False Negatives (FN)
	Fraud	False Positives (FP)	True Positives (TP)

Figure 8 : Confusion Matrix

For each classification with one value of the threshold we also have the corresponding **Confusion Matrix**

- **AUC** : The perfect model leads to AUC = 1 (100% TPR and 0% FPR)

- **Gini Coefficient** : GC = 2 *AUC – 1 (the classifier's advantage over a purely random one)
  GC = 1 denotes a perfect classifier

***Note : TPR – True Positive Rate = TP / (TP + FN) | FPR – False Positive Rate = FP / (FP + TN)**

Interpreting the Model : Optimal Threshold Selection

Institute of Actuaries of India

Youden's J Statistic:

- J = Sensitivity* + Specificity* – 1
- J = TPR + (1 – FPR) – 1 = TPR – FPR

- We can then choose the threshold with the largest J statistic value

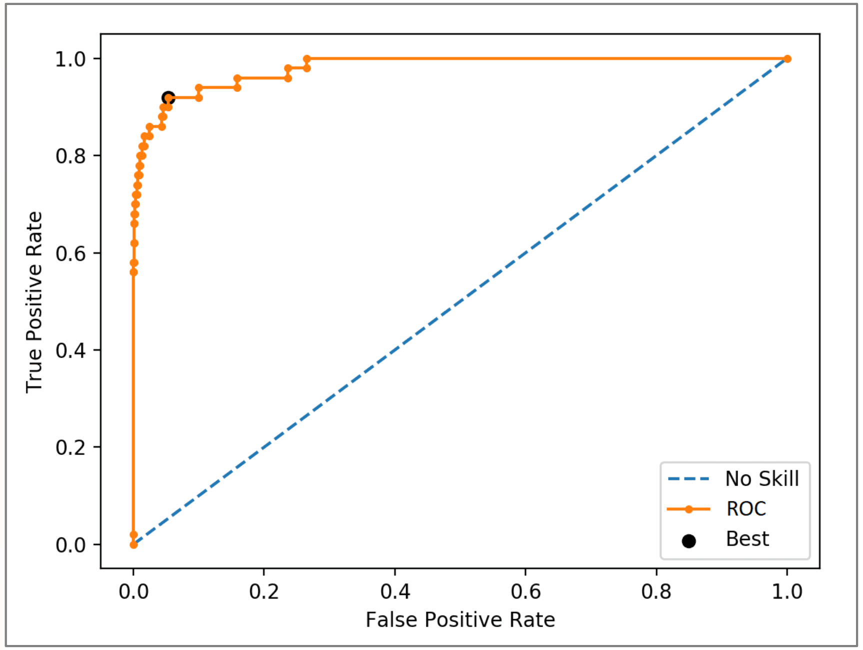


Figure 9 : ROC with optimal threshold

Points to note:

- **Optimal threshold** does not necessarily optimize the accuracy
- Accuracy is highly affected by class imbalance
- The use of a single index is therefore not generally recommended

- **From a practical usage perspective, the threshold can be chosen based on a cost-benefit calculation**
- **The benefit is the "saved" claim cost and the cost is the expenses incurred for investigation**

***Note : Sensitivity = TPR | Specificity = 1- FPR**

Model Selection : Why was XGBoost Chosen?

Institute of Actuaries of India

During model development phase multiple algorithms are tested. For our case study, the following were tested:

- **Logistic regression – with ROSE and SMOTE (sampling techniques)**
  - Logistic regression does not support imbalanced classification directly. It requires heavy over/under sampling for model convergence
  - Accuracy of the model at a defined threshold was lesser the accuracy of the tree-based models

- **Tree-based Model: Random Forest and XGBoost**
  - While both are ensemble decision trees, the two main differences are:
    - **How trees are built: Random Forest** works on the principle of **bagging** while **XGBoost** works on **boosting** - with each "new" model correcting the errors of the previous one
    - **Combining results**: **Random Forest** combines results at the end of the process (by averaging or "majority rules") while **XGBoost** combines results along the way

  - **Random Forest** and **XGBoost** each excel in different areas
    - Random forests perform well for multi-class object detection
    - XGBoost performs well when you have unbalanced data
    - For our case study the Random Forest Model was rejected due to overfitting

- **Final algorithm chosen was XGBoost – highest accuracy without overfitting**

Implementation : Dynamic, Real-time Fraud Detection

- Once deployed, the model should be refreshed at a regular interval to incorporate the new fraud patterns
- A robust feedback loop is extremely important for the success of any ML model

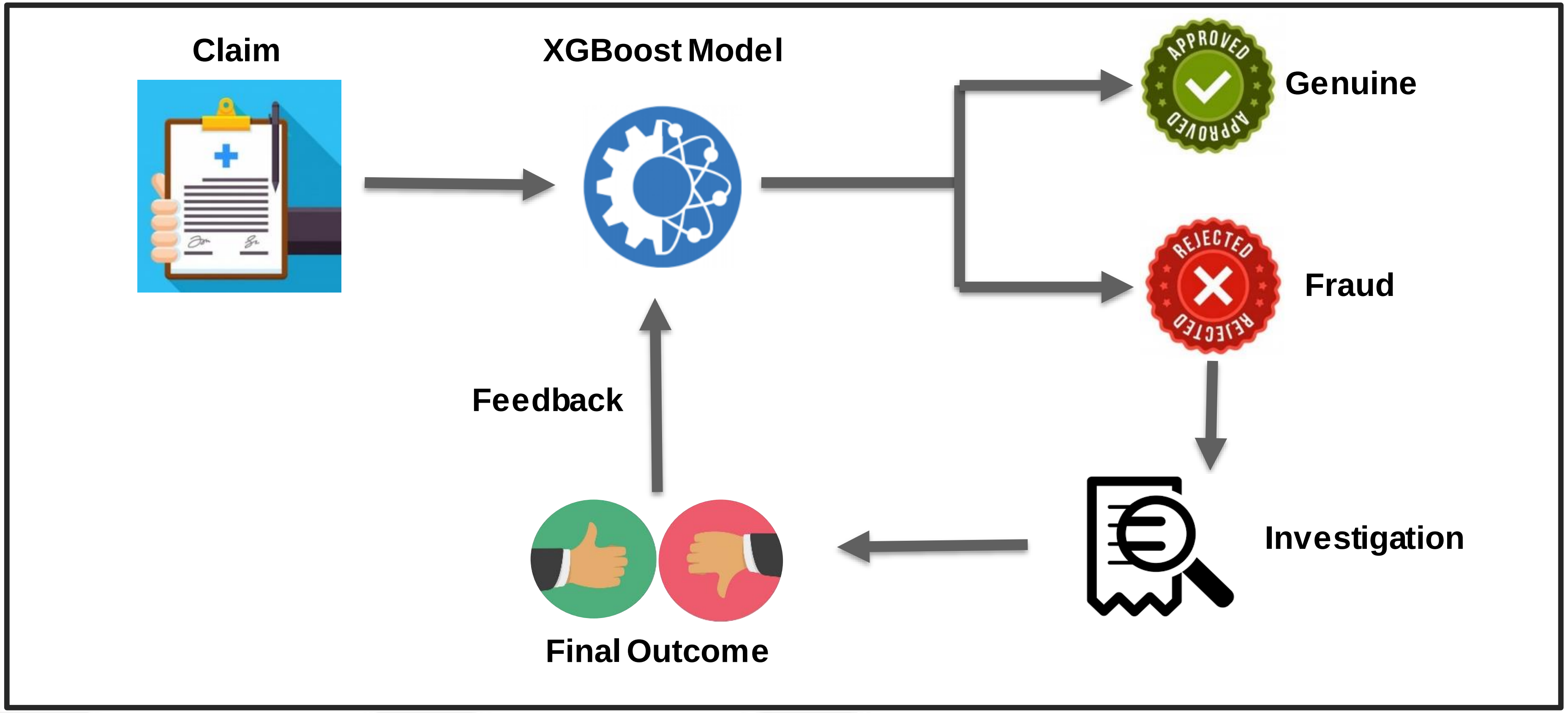


Figure 10 : Practical Implementation Approach

Starting up with XGBoost
There is a comprehensive guide on the XGBoost documentation website.
It covers installation details, tutorials across different operating platforms and languages

Key Takeaways

Institute of Actuaries of India

ML models are **trained to identify already established fraud patterns**

- There will be a bias towards existing fraud patterns

- Needs to be revisited at regular intervals, during the initial phase, to evaluate and tune the model

Success of the model depends on the variety of data available **(data depth),** the usability of the available data and a robust feedback loop

- **Predictive quality depends more on data than on algorithm**
    - There is no single BEST algorithm
    - Performance varies on the type of data one is working with

- **Outperformance by Ensemble Classifiers** : aggregation of weak classifiers can out-perform predictions from a single strong performer

References

- **XGBoost Documentation** - https://xgboost.readthedocs.io/en/latest/index.html
- **Decision Tree Classification in Python** - https://www.datacamp.com/community/tutorials/decision-tree-classification-python
- **Feature Importance and Feature Selection With XGBoost in Python** - https://machinelearningmastery.com/feature-importance-and-feature-selection-with-xgboost-in-python/
- **A Gentle Introduction to Threshold-Moving for Imbalanced Classification** - https://machinelearningmastery.com/threshold-moving-for-imbalanced-classification/
- **Youden's J statistic** - https://en.wikipedia.org/wiki/Youden%27s_J_statistic
- **Fraud Detection: How Machine Learning Systems Help Reveal Scams in Fintech, Healthcare, and eCommerce, by altexsoft**
- *Figure 8 - Comparative Analysis of Machine Learning Techniques for Detecting Insurance Claims Fraud* - *https://www.wipro.com/analytics/comparative-analysis-of-machine-learning-techniques-for-detectin/*