

# Revolutionizing Risk Assessment: A Novel Machine Learning Approach for Motor Insurance Liability Prediction

Yogesh Selvaraj  
Trivandrum, India  
Yogeshsp12@gmail.com

**Abstract - Assessing Insurance Liability accurately is of paramount importance for insurance companies to maintain financial stability and reduce the losses. Manual liability prediction in the insurance industry faces challenges due to subjective judgments, inconsistent data interpretation, and time-intensive processes. Human biases and lack of real-time insights hinder accurate risk assessment, leading to delayed claims processing, and potential inaccuracies in determining liability. Traditional methods for determining liability often involve complex rule-based systems that struggle to adapt to the dynamic nature of insurance data. In this context, machine learning techniques offer a promising avenue to enhance accuracy and efficiency in predicting insurance liability thereby reducing the losses. This Research proposes an innovative solution that integrates text mining techniques with a machine learning classification model to enhance liability prediction accuracy while providing transparent model explanations. Also the Research presents a comprehensive study and implementation of a machine learning classification model designed to predict the liability of an insurance company related to a claim. The model leverages a diverse imbalanced dataset containing historical claims information, policy details, claimant profiles, accident characteristics and other relevant variables. By employing advanced data preprocessing techniques, feature engineering, text-mining and state-of-the-art classification algorithms, the model learns complex patterns and relationships within the data. The solution contains multiple steps in building a ML based solution to help the business to accurately identify the Liability which involves 1)Data Compilation and Preprocessing 2) Text Mining and Feature Extraction 3) Handling Real world Imbalanced Data 4)Model Development 5) Training, Evaluation & Hyper-Parameter Optimization 6) Performance Metrics and Explainability Evaluation 7) Model Explainability**

**Keywords—** Machine Learning, Text Mining, Imbalanced Data, LightGBM.

## I. INTRODUCTION

Managing time expectations is the key driver of satisfaction—meaning, a prompt claim settlement is still the best advertisable punch line for insurance firms. While a new era of claims settlement benchmarks are being set with AI, the industry is shifting their attitude towards embracing the real potential of intelligent technologies that can shave-off valuable time and money from the firm's bottom-line.” The insurance claims process involves a series of procedures from the time the insurer is alerted to when a settlement is made in which First notice of loss(FNOL) starts the wheel of the claims cycle and is when the policyholder or any other person notifies the insurer of an unfortunate event. In the case of Motor insurance, a driver or Third Party informs the insurance company of a crash that occurred involving a vehicle.



**Fig. 1.** General end to end process flow of a Motor claim

Figure 1 shows the process flow of a Motor claim at which the Claims was notified to the insurance firm which is called as the First notice of loss stage and the claims handler whose role is to determine the Liability and the amount of settlement. The claims handler determines the nature and severity of the damage to the policyholder's car. His/her assessment relies on various information regarding the Incident by the reporter and other details regarding the incident.

The idea of this Research is to utilize the historical data of the claims notified and develop a liability prediction model which would enable the handlers to swiftly decide on Claims liability at FNOL Stage by deploying a Machine Learning Model.

## II. RESEARCH DESIGN & METHODOLOGY

### A. Objective of the Research

The objective of this Research is to utilize the historical data of the claims notified and develop a liability prediction model which would enable the handlers to swiftly decide on Claims liability at FNOL Stage. The idea is to develop a multi classifier model which predicts the Liability of the claim by using a trained model with historical data obtained at the FNOL stage and to predict the liability using Machine Learning Technique during the Final Stage of settlement.

The model is designed to predict in which Liability Group the claim will fall into as mentioned below,

|              |              |            |
|--------------|--------------|------------|
| FULLY LIABLE | SPLIT LIABLE | NOT LIABLE |
|--------------|--------------|------------|

### B. Data compiling and Pre-Processing

Claims notified for 3 years are collected and have extracted certain information of the claims at FNOL Stage which are relevant for the objective and have mapped certain variables in order to reduce the complexity of the data. The model is trained only using the claims which were settled to predict the Liability at the time of settlement.

Input variables, considered for building the model:

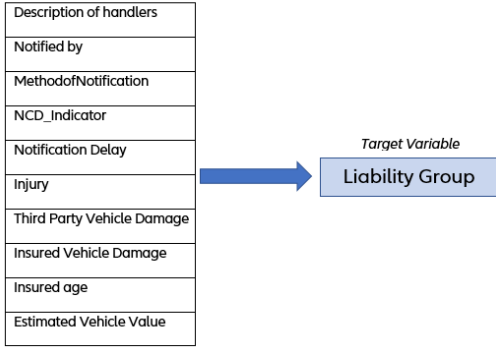


Figure 2 represents the Input variables and the Target variable at which the model is predicting.

### C. Text Mining and Feature Extraction

Text Mined the Description entered by handlers which is the free Text entered by the handlers about the event based on the information they receive as they are some vital information which can be fed into the model to identify the Liability.

The following Text Mining operations have been done on the claim handlers description to convert the free text into variables which the machine learning model could understand, 1. Tokenization 2. Lowercasing 3. Stemming and Lemmatization 4. Spell Checking and Correction 5. Removed stop words 6. Special character removal 7. Fixed Typos 8. Removed noises using Parts of speech Tagging 9. Tried Count Vectorizer & TFIDF models.

Count Vectorizer captures the presence of words but doesn't consider the significance of individual words within a document. This led us to try out some other enhanced bag of words models. Hence we tried other enhanced bag of words models.

The second model we evaluated is the term frequency-inverse document frequency (TF-IDF) bag of words model. In this model, documents are still represented as vectors, but instead of binary values (0s and 1s), each word in the document is assigned a score. These scores are computed by multiplying the term frequency (TF) and inverse document frequency (IDF) values for each word. Therefore, the score of any word in any document can be expressed using the following equation:

$$TFIDF(word, doc) = TF(word, doc) * IDF(word)$$

Hence, within this approach, two matrices must be computed: one encompassing the inverse document frequency for each word across the entire corpus of documents, and another encompassing the term frequency for each word within each individual document. The formulas for calculating both matrices are as follows:

$$TF(word, doc) = \frac{\text{Frequency of word} \in \text{the doc}}{\text{No. of words} \in \text{the doc}}$$

$$IDF(word) = \log_e \left( 1 + \frac{\text{No. of docs}}{\text{No. of docs with word}} \right)$$

The TF-IDF model performs effectively by assigning significance to less common words, as opposed to treating all words equally, as seen in the binary bag-of-words model. Description example : "It is alleged that our insured collide with the rear of Third Party's vehicle"

| Doc/<br>Words | allege<br>Insure | Insure<br>collide | rear | thirdparty |
|---------------|------------------|-------------------|------|------------|
| D1            | 0.1              | 0.1               | 0.17 | 0.27       |
| D2            | 0                | 0.09              | 0    | 0.01       |
| D3            | 0.18             | 0                 | 0.09 | 0          |
| D4            | 0                | 0                 | 0    | 0          |

Table 2. TFIDF Vector for the description

### D. Handling Real world Imbalanced Data

In the world of data analysis and machine learning, the quality of the dataset can make or break the success of your models. One common challenge that data scientists often face is data imbalance, where one class significantly outnumbers the others. This disparity can lead to biased model outcomes and reduced predictive power. In this article, we will discuss how we encountered a data imbalance issue and successfully addressed it using the Synthetic Minority Over-sampling Technique (SMOTE).

The Research involved building a Multi-classification model to predict the Liability group encountered a severe imbalance in Liability Groups.

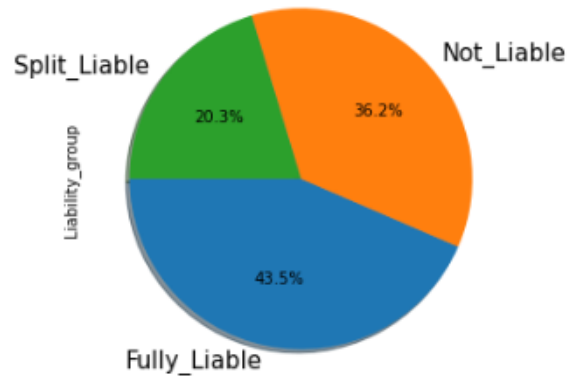


Figure 3 represents the percentage of each Liability group in the data set.

The majority class (Fully Liable) dominated the dataset, accounting for around 44% of the samples, while the minority class (Split Liable) represented only 20%. This skewed distribution could severely impact our model's ability to detect Liability Groups accurately.

The challenges faced due to the imbalance is as mentioned 1. Biased Model 2. Low Recall 3. Misleading Accuracy

In order to extract the best possible output from the Machine Learning algorithm the data needs to be balanced and have tried balancing techniques such as over sampling and under

sampling and combined both over sampling and under sampling together on the classes and have finalized over sampling 20% of the minority class using the SMOTE Method which brought a better accuracy while training the model.

**SMOTE** (Synthetic Minority Over-sampling Technique) is an oversampling technique where the synthetic samples are generated for the minority class. This algorithm helps to overcome the overfitting problem posed by random oversampling. It focuses on the feature space to generate new instances with the help of interpolation between the positive instances that lie together.

It's crucial to apply SMOTE only to the training dataset, ensuring that the validation and test datasets remain unbiased and representative of real-world scenarios.

### E. Model Development

Choosing the right algorithm for a multi-classification model is a critical decision that can have a profound impact on the success of the machine learning model and the overall Research. Selecting the right algorithm for a multi-classification Research is a pivotal decision that should be based on a deep understanding of the problem, the data, and the Research's requirements. A well-informed choice can lead to a more accurate, efficient, and interpretable model, contributing to the success of the Research and its ability to provide valuable insights and predictions.

The final prediction model is determined after several permutation and exploring different techniques,

- Compared 15 classifier model in Pycaret package
- Finalized Ridge Classifier, Gradient Boosting , LightGBM Models
- Built Ensembled models
- Tried One vs Others approach has been tried
- Tried grouping multiple groups into one class and rest as others
- Built 3 different models getting trained on each level of training data in order not to lose any data and combined the results of each model to get a better accuracy.

Also we have tried PyCaret which particularly helped in choosing the right algorithm which better suited for the condition of data.

*PyCaret simplifies the process of algorithm selection by automating the training, evaluation, and comparison of various classification algorithms. It empowers data scientists to quickly identify the best-performing models for their specific datasets, making it a valuable tool for efficient and effective model development.*

The algorithm that gives the best Accuracy score along with the best ROC & AUC score and brought better recall value on all the three classes is the Light Gradient Boosting Classifier(LGBM).

*Light Gradient boosting is a machine learning technique for regression and classification problems, which produces a prediction model in the form of an ensemble of weak prediction models, typically decision trees. It builds the model in a stage-wise fashion, and it generalizes them by allowing optimization of an arbitrary differentiable loss function. LGBM relies on the intuition that the best possible next model, when combined with previous models, minimizes the overall prediction error. Also the LGBM algorithm has a provision of providing class weights which can be better suited for an imbalanced dataset.*

### F. Training , Evaluation & Hyperparameter Optimization

Multiple Machine Learning model have been built and the best one chosen based on the different model validation matrices and after testing the model with real world unseen data.

```
Classification report:
              precision    recall  f1-score   support

Fully_Liable      0.96      0.89      0.92       467
  Not_Liable      0.76      0.81      0.79       388
 Split_Liable      0.70      0.71      0.71       228

 accuracy              0.83       1083
 macro avg           0.81      0.81      0.81       1083
 weighted avg        0.83      0.83      0.83       1083
```

The LGBM built has been validated by using different model validation matrices such as , K-Fold cross validation has been used to understand whether the model is getting overfitted or underfitted.

```
[0.81717452 0.81440443 0.81440443 0.84487535 0.83379501 0.81994446
 0.80886427 0.7700831 0.8033241 0.79501385]
Accuracy: 81.22%
```

Model is able to generate an Accuracy of 82%.

90% of data is used for training each model.

10% of data is used to test the prediction power of models.

Accuracy = Ability of model to predict the outcome correctly.

Recall – Ability of the model to correctly identify the True Positives.

The Model is getting a better ROC AUC scores

```
One-vs-One ROC AUC scores:
0.906417 (macro),
0.922873 (weighted by prevalence)
One-vs-Rest ROC AUC scores:
0.927940 (macro),
0.962679 (weighted by prevalence)
```

Tested the model with various un-seen data and figured out the model is performing consistent on the new test data.

In this Research we have fine tuned the model parameters using Bayesian Optimization which brought better results along with using the class weight parameters manually in

the Light GBM model. Also tried optimizing the parameters using Randomized Search, However the parameters getting better results were obtained by doing the Bayesian Optimization.

Best classification report:

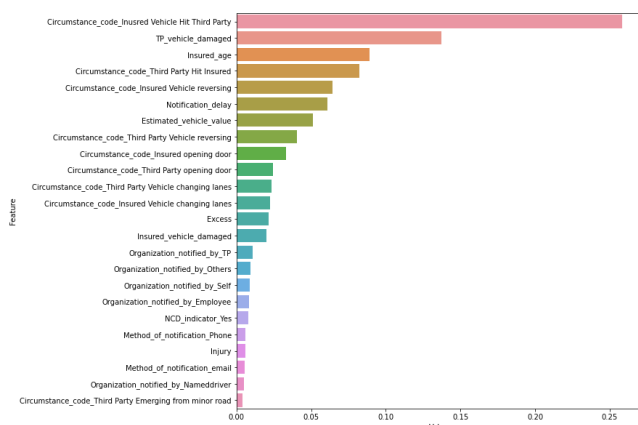
|              | precision | recall | f1-score | support |
|--------------|-----------|--------|----------|---------|
| Fully_Liable | 1.00      | 0.84   | 0.91     | 467     |
| Not_Liable   | 0.79      | 0.80   | 0.79     | 388     |
| Split_Liable | 0.62      | 0.80   | 0.70     | 228     |
| accuracy     |           |        | 0.82     | 1083    |
| macro avg    | 0.80      | 0.81   | 0.80     | 1083    |
| weighted avg | 0.84      | 0.82   | 0.83     | 1083    |

In machine learning, hyperparameter optimization or tuning is the problem of choosing a set of optimal hyperparameters for a learning algorithm. A hyperparameter is a parameter whose value is used to control the learning process. Bayesian Optimization builds a probability model of the objective function and uses it to select hyperparameter to evaluate in the true objective function.

### G. Model Explainability and Interpretation

The finalised model is a Light gradient boosting classifier, which is not natively explainable due to its complexity, so instead, we rely on model Explainability frameworks to provide an explanation for a single prediction, showing the level of influence each individual output feature had on the model output. The model prediction is evaluated using an explainer, specifically SHAP, and was able to reproduce and explain a specific prediction, using the exact version of the model that was used for the original prediction.

The figure below illustrates the feature importance of the trained multi classifier model based on importance type as gain.



*SHAP* which stands for *Shapley Additive explanations* is a solution concept used in *Game Theory* that involves fairly distributing both gains and costs to several actors working in coalition. *SHAP* values interpret the impact of having a certain value for a given feature in comparison to the prediction we'd make if that feature took some baseline value. The summary plot combines feature importance with

*feature effects*. Each point on the summary plot is a *Shapley value* for a feature and an instance. The position on the y-axis is determined by the feature and on the x-axis by the *Shapley value*.

Now lets see how the Features in our dataset contribute towards the predicting liability group.



### III. CONCLUSION

In this capstone Research we provide an effective method to build a multi-class classification model. It is very important to understand the business context and figure out the best metrics that should be Optimized to understand the model performance. Along with this, understanding the data and choosing the best class balancing method will further increase the model's performance.

One of the major challenges in building the model is predicting the Split Liable class. As we observe the data, the Split Liable class has an essence of both the 'Fully Liable' class as well as the 'Not Liable' class respectively. However, using the right class balancing technique along with the parameter 'class weight' in an optimal way helped us to improve the model accuracy as well as other key metrics. imbalance issue and successfully addressed it using the Synthetic Minority Over-sampling Technique (SMOTE).

### IV. REFERENCES

1. Manisha Pravin Mali, Dr. Mohammad Atique, "Applications of Text Classification using Text Mining", pp. 1-4, 2014.
2. Ahsan, Md Manjurul, M. A. Parvez Mahmud, Pritom Kumar Saha, Kishor Datta Gupta, and Zahed Siddique. 2021. Effect of Data Scaling Methods on Machine Learning Algorithms and Model Performance. *Technologies* 9: 52.
3. H. He, and E. Garcia, "Learning from imbalanced data", *IEEE Transactions on Knowledge & Data Engineering*, vol. 9, pp. 1263-1284. 2008.
4. F. Provost, "Machine learning from imbalanced data sets 101", In *Proceedings of the AAAI'2000 workshop on imbalanced data sets*, vol. 68, pp. 1-3, 2000.





.