ISSN 2616-5775 Vol. 6, Issue 8: 68-74, DOI: 10.25236/AJCIS.2023.060808

Insurance Fraud Detection Based on XGBoost

Haoran Zheng[1,a,#], Fan Peng[2,b,#], Yawen Tian[3,c,#], Zizhou Zhang[4,d,#], Wenting Zhang[5,e,#]

[1]Software Engineering, Shandong University of Technology, Zibo, Shandong, China
[2]Internet of Things Project, Hebei University of Technology, Tianjin, China
[3]McMaster University, Toronto, Ontario, Canada
[4]Nttingham University, Ningbo, Zhejiang, China
[5]Qihua Academy Nanchang, Nanchang, Jaingxi, China
[a]2490465137@qq.com, [b]1649824692@qq.com, [c]1255495322@qq.com, [d]ssyzz30@nottingham.edu.cn, [e]77998282qq.com
([#]Co-first author)These authors contributed equally to this work.

Abstract: *This research conducted a comprehensive study on predicting customer car insurance claims using Gradient Boosting Decision Tree (GBDT) and XGBoost models. The process included data exploration, feature engineering, model evaluation, and parameter tuning. The dataset was explored based on variable types and missing values, and further processed through mean encoding and outlier removal. Date features were also manipulated to create more meaningful features. Two models, GBDT and XGBoost, were trained and evaluated based on their AUC (Area Under the Curve) values. Both models demonstrated good predictive power, with GBDT slightly outperforming XGBoost. The results of this study provide valuable insights for predicting insurance claims, offering significant implications for further research and practical applications.*

Keywords: *GBDT, XGBoost, Machine Learning, Car Insurance Fraud Detection*

1. Introduction

A major use in the insurance profession is now identifying insurance fraud. Insurance, as an integral part of the financial system, is essential to the protection of livelihoods and the advancement of society. However, insurance fraud has been on the rise, costing insurance providers and the entire financial system an enormous amount of money. Such conduct gravely undermines the stability and operation of insurance businesses and erodes the public's trust in the insurance system. The purposeful giving of misleading information or careful planning of accidents and losses on the part of the insured in order to receive false protection is referred to as insurance fraud. This unethical behavior not only jeopardizes the interests of insurance providers, but it also places an additional burden on loyal customers, raising the general price of insurance.

The risk control methods used by insurance firms for traditional insurance fraud prevention is quite easy to understand. Insurance firms essentially get anti-fraud tips through sample cases and operator experience. However, insurance companies frequently lack the capacity to thoroughly analyze and mine basic underwriting and claims data in order to rapidly recognize hidden risk indicators, which makes it difficult to identify fraud signals in time while managing cases. An additional challenge faced by insurance firms in addressing fraud is the absence of a data exchange mechanism. Machine learning, deep learning, and other algorithms are being used and developed quickly due to the big data technology's increasing maturity and the rapid increase in computing power, technological innovations like image recognition.

Currently, machine learning technology is crucial to the insurance industry's efforts to combat fraud, particularly when it comes to accurately identifying and preventing fraud. Machine learning technology is being employed mostly in anti-fraud applications by developing pertinent models, diligently establishing the features of fraud cases, and employing algorithms to quantitatively estimate the level of fraud risk in claims examples. This approach can dramatically reduce labor expenses for insurance firms while increasing the reliability and efficiency of fraud risk identification.

ISSN 2616-5775 Vol. 6, Issue 8: 68-74, DOI: 10.25236/AJCIS.2023.060808

2. Related Work

With the rise of machine learning algorithms, researchers have started to apply techniques such as Random Forest, Support Vector Machine, and others to the field of insurance fraud detection. These methods can automatically learn patterns from data and build predictive models to assist in fraud detection. In recent years, ensemble learning techniques like XGBoost and Gradient Boosting Trees have been widely used for fraud detection. Hancock, J.T., and Khoshgoftaar, T.M.[1] proposed using the Catboost algorithm to achieve higher average AUC values, resulting in better performance compared to other algorithms in insurance fraud detection tasks. Sri Ghattamaneni, Ricardo Portilla, Nikhil Gupta[2] mentioned improving fraud detection model accuracy by combining multiple base models to create an anti-fraud framework.

Furthermore, the rapid development of deep learning techniques has sparked a research trend in the insurance fraud detection domain. Sumaya Sanober et al.[3] proposed using a novel deep learning framework implemented in Spark for financial fraud detection, leading to higher precision and accuracy. Additionally, using deep autoencoders has been shown to improve accuracy and enhance fraud detection effectiveness.

As discussed by Wang Weiwei[4], utilizing data and model training can significantly improve the accuracy of system analysis and reduce operational maintenance costs. This paper introduces a novel feature selection method based on stacking and compares the proposed architecture with various algorithm models through analytical methods. The innovation and differences of this method mainly lie in the aspects of feature engineering and multi-model ensemble. First, this study addresses the characteristics of insurance fraud features by employing encoding techniques like MeanEncoder to process categorical variables, fully utilizing their classification information. Second, the study incorporates multiple machine learning models such as GBDT and XGBoost, enhancing fraud detection robustness and reliability by combining predictions from different models.

Moreover, this research extensively validates and applies the proposed approach on multiple real insurance datasets, covering various insurance types and fraud scenarios. As Wang Chen et al.[5] mentioned, combining fraud risk warning intelligent outreach methods achieves end-to-end communication-side prevention and control. The validation on real data adds practical significance and operability to the results of this study. Furthermore, the application in actual insurance business further demonstrates the feasibility and effectiveness of this research.

In conclusion, by optimizing feature engineering and introducing a multi-model ensemble strategy, this research improves the accuracy and efficiency of fraud detection. Moreover, the validation in practical applications provides valuable references for risk management and business decision-making in the insurance industry.

3. Model method

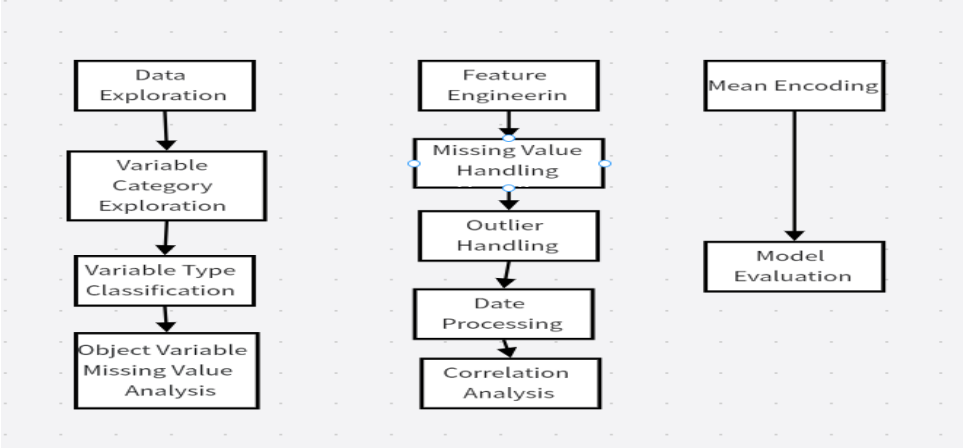

Figure 1: Model method

ISSN 2616-5775 Vol. 6, Issue 8: 68-74, DOI: 10.25236/AJCIS.2023.060808

3.1. Data exploration

The content of data exploration can help us better understand the feature distribution, missing values, and correlations among variables in the dataset, providing a foundation for subsequent feature engineering and modeling (Figure 1).

Variable type exploration: By classifying the data according to variable types into numerical and object variables, we can explore variable types. Numerical variables include continuous variables, discrete variables, and single-value variables, while object variables are non-numerical variables.

In this study, we first used the 'data.dtypes' method to perform data type analysis, categorizing variables into two data types: 'int64' and 'object'. Numerical features were stored in the 'numerical_feature' variable, while object-type features were stored in the 'object_feature' variable.

Analysis of Continuous, Discrete, and Single-value Variables: Based on the different characteristics of the numerical variables in the data, we can further categorize them into continuous variables, discrete variables, and single-value variables. Continuous variables refer to variables that can take any numerical value, discrete variables are those that can only take a finite or countable number of values, and single-value variables are variables that take only one value in a given dataset.

In this study, we draw frequency charts to examine the types and distribution of discrete variables, in order to better understand the characteristics of the data. Through these frequency charts, we can intuitively understand the value range, frequency of occurrence, and potential imbalances of each discrete variable.

Missing Value Analysis for Object Variables: We determine whether each column of object variables has missing values, and filter out the rows with missing values from the original data.

3.2. Feature engineering

Missing Value Processing: Firstly, by identifying the feature list with missing values, we can determine which features have missing values.

Then, for these features with missing values, we fill in the missing values using the mode. With the fillna function and mode() method, we can replace missing values with the mode of the corresponding feature.

Outlier Processing: Outliers refer to values in the dataset that are significantly different from other observations. For abnormal data, we can exclude them from the model training to achieve better results.

By using the 3σ rule (3-Sigma rule), we can determine the upper and lower limits based on the data's mean and standard deviation, and values beyond this range are considered outliers.

Date Processing: For features containing dates, such as 'policy_bind_date' and 'incident_date', we first use the pd.to_datetime function to convert them into date format.

Then, by calculating the time difference between the two dates, we create a new feature 'delta_time', representing the number of days between the accident date and the policy signing date.

We also extract the month from 'incident_date' and convert it to a string format as a new feature, 'picked_month'.

Correlation Analysis: By calculating the correlation matrix among numerical variables and visualizing the correlation with a heatmap, we can identify variables with correlation higher than 0.6.

In this case, we find a high correlation between 'age' and 'customer_months', and remove them from the numerical features.

Moreover, we also remove the target variable 'fraud' from the correlation analysis.

Mean Encoding: For variables with more than 10 discrete values, we use MeanEncoder to encode the discrete values into continuous values for subsequent analysis.

First, we identify the feature list to be mean-encoded and remove these features from the object features.

Next, we use the MeanEncoder class to mean-encode these features, and add the encoded features to the dataset.

ISSN 2616-5775 Vol. 6, Issue 8: 68-74, DOI: 10.25236/AJCIS.2023.060808

Finally, we use LabelEncoder to label-encode the remaining object features, converting them into numerical features.

3.3. Model Evaluation

In this study, we employed the Gradient Boosting Decision Tree (GBDT) and XGBoost models, and evaluated them to determine their performance in prediction tasks. First, we split the dataset into training and testing sets, with the training set accounting for 70% of the total samples. Then, we initialized the GBDT and XGBoost models using pre-set parameters. Next, we trained these two models on the training set, fitting them using the fit function.

XGBoost is a machine learning algorithm that improves predictive performance by combining multiple decision trees. It utilizes gradient boosting techniques to iteratively train decision trees and progressively refine the predictions. XGBoost also introduces regularization and automatic feature importance learning to enhance the model's generalization capacity and reduce overfitting. With parallel computation, XGBoost performs excellently when handling large-scale data. In general, XGBoost is widely applied to various machine learning tasks due to its high performance, efficiency, and accuracy.

Subsequently, we used the trained models to make predictions on the testing set and extracted the positive class probabilities from the prediction probabilities. To assess model performance, we computed the AUC (Area Under the Curve) metric.

Finally, we compiled the evaluation results of the two models into a table, which includes the model names and their corresponding AUC values. By comparing the AUC values, we were able to understand the relative performance of the GBDT and XGBoost models in prediction tasks.

In our study, two main model methods were adopted, which are the Gradient Boosting Decision Tree (GBDT) and XGBoost. These methods can be understood as a process of executing a series of iterative steps.

For GBDT, we first initialize a model that minimizes the loss, represented as F0(x). Then, a series of loop iterations are carried out. In each iteration, we compute the gradient for each data point and fit a new decision tree. Afterwards, we update the model through the learning rate and the decision tree. The new model formula is as follows:

$$Fm(x) = Fm\text{-}1(x) + Learning\_rate * Tree \quad (1)$$

Finally, by continually iterating in this manner, we obtain the final model.

Here is the pseudo-code description for the Gradient Boosting Decision Tree (GBDT)(Table 1).

Table 1: GBDT

Method Name:GBDT
Input: Data: A dataset of customer car insurance claims, including features x, such as insurance number, annual premium, age, etc.
Output: The probability of predicting as the positive class.

begin
Initialize the model.F0(x) = argmin(Σl(yi, c)),Is the minimal loss across all samples.
for i=1 to Maximum_rounds do
for each (x, y) in Data do
Compute the gradient g_i for each sample. end
Generate a new decision tree such that Σ[g_i*score] is minimized, where score is the score of the leaf node.
Add the new decision tree to the model,Fm(x) = Fm-1(x) + Learning_rate * Tree
end
Return the final model.F(x)
End.

For XGBoost, the method is similar to GBDT, but it introduces an additional regularization step, as well as computation of the second-order gradient. On each data point, we compute gradients g_i and h_i, and then fit a new decision tree taking into account the regularization factor. The model is updated in the same way as GBDT, and the new model formula is:

$$Fm(x) = Fm\text{-}1(x) + Learning\_rate * Tree \quad (2)$$

ISSN 2616-5775 Vol. 6, Issue 8: 68-74, DOI: 10.25236/AJCIS.2023.060808

Here is the pseudo-code description for the Gradient Boosting Decision Tree (XGBoost) (like Table 2).

Table 2: XGBoost

Method Name:XGBoost
Input: Data: A dataset of customer car insurance claims, including features x, such as insurance number, annual premium, age, etc.
Output: The probability of predicting as the positive class.

begin
Initialize the model.F0(x) = argmin(Σl(yi, c)),Is the minimal loss across all samples.
for i=1 to Maximum_rounds do
for each (x, y) in Data do
Compute the gradient g_i for each sample g_i and h_i
End
function XGBoost(Data, Maximum_rounds, Learning_rate)
Generate a new decision tree such that Σ[g_i*score + 0.5 * h_i * score^2 + lambda * ||score||^2] is minimized,where score is the score of the leaf node.
Add the new decision tree to the model.,Fm(x) = Fm-1(x) + Learning_rate * Tree
end
Return the final model F(x)
End

The results of this study indicate that both the GBDT and XGBoost models demonstrated good predictive power on the test set. These results provide a strong basis for the selection of the optimal model and have significant guiding significance for further research and applications.

4. Test

4.1. Dataset

	policy_id	age	customer_months	policy_bind_date	policy_state	policy_csl	policy_deductable	policy_annual_premium	umbrella_limit	insured_zip
0	122576	37	189	2013-08-21	C	500/1000	1000	1465.71	5000000	455456
1	937713	44	234	1998-01-04	B	250/500	500	821.24	0	591805
2	680237	33	23	1996-02-06	B	500/1000	1000	1844.00	0	442490
3	513080	42	210	2008-11-14	A	500/1000	500	1867.29	0	439408
4	192875	29	81	2002-01-08	A	100/300	1000	816.25	0	640575
...	...	...	...	...	...	...	...	...	...	...
695	1008425	37	196	1997-06-29	C	250/500	500	1301.20	0	474615
696	770702	43	229	2001-05-29	A	250/500	500	1434.94	8000000	444476
697	755099	35	209	2003-01-11	C	100/300	500	1639.46	0	639608
698	693804	44	275	2003-07-22	B	500/1000	2000	1042.29	0	432061
699	598086	47	263	1996-08-15	C	500/1000	500	1282.56	0	433809

Figure 2: Dataset

4.2. Hardware Environment

Table 3: Hardware Configuration

Name	Configuration
Cpu	5900hx
Gpu	3080laptop
RAM	32G 3000hz

4.3. Evaluation Standard

In this case, the performance of the model is evaluated using the confusion matrix threshold（Table 3 and Figure 2).

4.4. Model Training

The dataset is divided using the train_test_split function. X_train and y_train are the features and

ISSN 2616-5775 Vol. 6, Issue 8: 68-74, DOI: 10.25236/AJCIS.2023.060808

labels of the training set, while X_test and y_test are the features and labels of the test set. According to the given parameter train_size=0.7, the training set accounts for 70% of the total dataset and the test set accounts for 30%.

X_train,X_test,y_train,y_test = train_test_split(mean_X_train,y_label,train_size=0.7)

4.5. Parameter Tuning

In this test, the parameters of the GBDT and XGBoost models were adjusted and the impact of changes in model parameters on model performance was recorded (Figure 3).

For the GBDT model, we adjusted the learning_rate to 0.1, n_estimators (number of weak learners) to 30, max_depth (maximum depth of the tree) to 3, and min_samples_split (the minimum number of samples required for further splitting of internal nodes) to 300.

For the XGBoost model, we adjusted the learning_rate to 0.01, reg_alpha (weight of the L1 regularization term) to 0, max_depth (maximum depth of the tree) to 3, gamma (minimum loss reduction required to make a further partition on a leaf node) to 0, and min_child_weight (minimum sum of instance weight needed in a child) to 1(Table 4).

Table 4: XGBoost model

Model	Parameter	Value
GBDT	learning_rate	0.1
	n_estimators	30
	max_depth	3
	min_samples_split	300
XGBoost	learning_rate	0.01
	reg_alpha	0
	max_depth	3
	gamma	0
	min_child_weight	1

4.6. Feature Importance Ranking

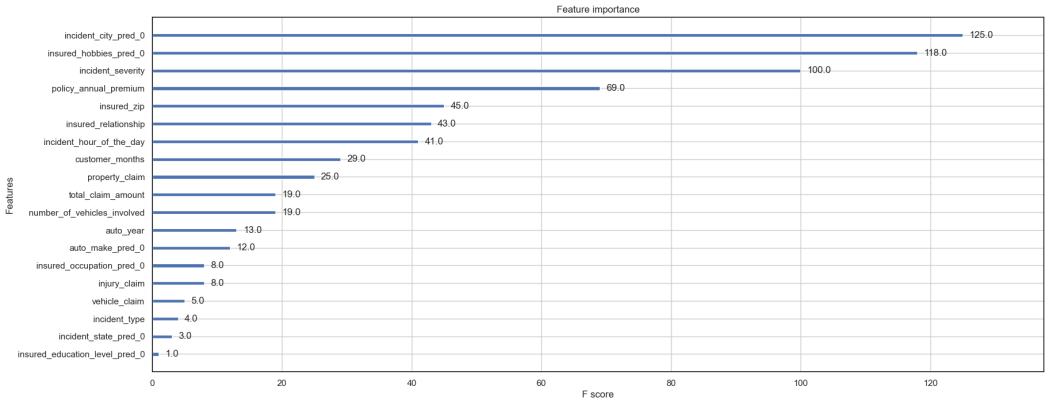

Figure 3: Feature Importance Ranking.

4.7. Model Evaluation

The prediction accuracy of the GBDT model is slightly higher than that of the XGBoost model (Table 5)

Table 5: Model Evaluation.

	GBDT	XGBoost
AUC	0.884230	0.884230

ISSN 2616-5775 Vol. 6, Issue 8: 68-74, DOI: 10.25236/AJCIS.2023.060808

5. Conclusion

This paper establishes XGBoost and Gradient Boosting Decision Trees (GBDT) models for data classification. Each category is analyzed, and the data is processed to handle missing values and outliers. Finally, the models are trained and evaluated. The novelty of this paper lies in using two gradient boosting ensemble learning algorithms, XGBoost and GBDT, for the classification task. Different parameters are adjusted to control the complexity and fitting ability of GBDT and XGBoost, such as learning rate, tree depth, regularization coefficient, etc. These parameters can impact the model's generalization ability and training speed, and their selection and optimization depend on the characteristics of the dataset. Model optimization can be further explored by trying different types of machine learning models, such as decision trees, random forests, support vector machines, gradient boosting trees, neural networks, etc. Different models may exhibit varying adaptability to the data's features and patterns, and experimenting with various models can help identify the most suitable one. Fine-tuning model parameters can be achieved by adjusting the chosen model's settings to optimize its performance. Techniques like cross-validation and grid search can be utilized to find the best combination of parameters.

References

[1] Hancock, J. T., & Khoshgoftaar, T. M. (2021). Gradient boosted decision tree algorithms for medicare fraud detection. SN Computer Science, 2(4), 268.

[2] Sri, G., & Ricardo, P. (2021). Combining Rules-Based and Machine Learning Models to Combat Financial Fraud .The Databricks Blog.

[3] Sanober, S., Alam, I., Pande, S., Arslan, F., Rane, K. P., Singh, B. K., ... & Shabaz, M. (2021). An enhanced secure deep learning algorithm for fraud detection in wireless communication. Wireless Communications and Mobile Computing, 2021, 1-14.

[4] Wang, X., Yi, Z., & Wu, H. (2018, August). Research and Improvement of Internet Financial Anti-Fraud Rules Based on Information Gain and Support. In Journal of Physics: Conference Series (Vol. 1069, No. 1, p. 012104). IOP Publishing.

[5] Wang, C., Luo, Q., Pan, L., Yuan, T. S., & Liu, Y. Z. (2022). Research and Application of Real-time Intelligent Anti-fraud System Based on Trusted AI and Spatio-temporal Big Data. Telecommunications Engineering Technology and Standardization. 35(12), 34-39.