Auto Insurance Claims Prediction Model Based on Bayesian Optimization and Light GBM

Zhen Li[1,a], Anjun Song[1,b]

[1]Information engineering college, Shanghai Maritime University, Shanghai 201306, China.

[a]Li_zhen0125@163.com, [b]ajsong@shmtu.edu.cn

Abstract

In recent years, the generalized linear model has been widely used in auto insurance pricing. With the development of artificial intelligence, some studies have begun to use machine learning models to predict whether auto insurance claims will be settled, but only default parameters are used in the establishment of machine learning models. In the data preprocessing stage, this paper uses Boruta algorithm to filter out 22 features from 36 features, put 22 features into the Light GBM model, classify and determine whether claims will occur, and use Bayesian to optimize parameters. Finally, Five indicators including ROC curve, accuracy rate, precision rate, recall rate, and F1 value are introduced to evaluate model performance. The results show that compared with other algorithms, Light GBM based on Bayesian optimization has better evaluation indicators than other models in terms of auto insurance claims prediction.

Keywords

Auto Insurance Claims; Rate Determination; Light GBM; Bayesian Optimization; Boruta.

1. Introduction

With the increase in the number of cars in my country, purchasing car insurance is a necessary condition for cars on the road. Therefore, the insurance company is also very important in determining the rate of auto insurance. By establishing a suitable model to predict claims, and thereby adjusting the appropriate insurance premium rate, the insurance industry's profit in the auto insurance business market can be improved.

At present, the early research on the determination of auto insurance rates is mainly based on the generalized linear model [1], and the model has been deeply studied and improved. Many insurance companies are still using these methods. Nelder and McCullag applied the generalized linear model (GLM) to the field of actuarial insurance in 1989, and assumed that the amount of claims obeys the Gamma distribution, the number of claims obeys the Negative Binomia distribution or the Possion distribution, and the Tweedie distribution of claims intensity. In 2014, Kelvin combined the Generalized Linear Additive Model (GAMLSS) with Bayesian methods to model insurance claim data. Meng Shengwang and Yang Liang [2] proposed a random effect zero inflation loss number regression model. By adding a random effect zero inflation loss number regression model on the basis of the Poisson distribution, and introducing a quadratic smoothing term, the results show that the model is The combined effect is better.

With the re-emergence of artificial intelligence (AI), in 2012 GuelMan tried to use machine learning algorithm gradient boosting numbers to model claims frequency and intensity separately, and used undersampling and cross-checking to process data for the problem of data imbalance. The result It shows that the gradient boosting tree is better than the traditional generalized linear model in terms of the frequency and intensity of claims.

Subsequently, in 2015, Lee et al. proposed the Delta Boosting method. Based on model analysis, it was shown that this method has higher accuracy than generalized linear models and gradient boosting trees in practical applications. Subsequently, Lee and Antinio applied a variety of algorithms such as neural networks, decision trees, generalized linear models, and generalized additive models to claim probability scenarios. The results showed that neural networks have the highest prediction accuracy,

but there are also overfitting problems. Meng Shengwang and Huang Yifan proposed that the XGBoost model has a good predictive ability for the probability of claim settlement, and the importance of XGBoost features can provide a reference for the determination of the rate.

However, in the above-mentioned research, feature selection is not performed in the preprocessing stage, and the hyperparameters of the machine learning model used in the modeling are all default values. Since the machine learning model relies too much on parameters, the default parameters will cause greater errors in the model. This paper proposes a Light GBM algorithm based on Boruta's feature selection for auto insurance claims data and Bayesian optimization. The RPubs open source data set was used to train the model, and after verification through the test set, it was found that the use of Bayesian-optimized Light GBM to build a car insurance claims prediction model was superior to other algorithms in terms of performance and generalization.

2. Methods and Models

In this paper, Boruta algorithm is used for feature selection, and Bayesian optimization-based Light GBM is used to establish a prediction model for auto insurance claims. Use RPubs open source data set to construct training set and test set to carry out model training. The car owner feature class, vehicle feature class, and policy feature class are used as the input of the model to determine whether a claim has been settled as the output feature, and finally the optimal model parameters are obtained through Bayesian optimization.

2.1 Boruta feature selection method.

Boruta algorithm is a wrapper based on random forest classification algorithm, which is implemented in R package randomForest (Liaw and Wiener 2002) [7]. The random forest classification algorithm is a relatively fast classification algorithm, which can usually be realized without adjusting the parameters, and gives a numerical estimate of the importance of features. It is an ensemble method for classification through voting of multiple unbiased weak classifiers-decision trees. These trees are independently developed on different wrapper samples in the training set. The importance measurement of attributes is the loss of classification accuracy due to the random arrangement of attribute values among objects. Calculate separately all the trees in the forest that are classified with a given attribute. Then calculate the average and standard deviation of the loss of precision. In addition, the Z-score calculation method of dividing the average loss by the standard deviation can be used as the importance. Unfortunately, the Z score is not directly related to the statistical significance of the feature importance returned by the random forest algorithm, because its distribution is not N(0,1). However, at Boruta, we use the Z score as an important metric because it takes into account the fluctuations in the average accuracy loss between trees in the forest.

Since the Z score cannot be used directly to measure importance, some external reference is needed to determine whether the importance of any given attribute is significant, that is, whether it can be distinguished from the importance of random fluctuations. Therefore, Boruta algorithm designs an information system with random attributes. For each attribute, we create a corresponding "shadow" attribute whose value is obtained by moving the value of the original attribute between objects. Then, we use all the attributes of this extended system to perform classification and calculate the importance of all attributes.

The importance of shadow attributes can be non-zero due to random fluctuations. Therefore, the imported set of shadow attributes is used as a reference for deciding which attributes are really important.

Due to the randomness of the random forest classifier, the importance metric itself is also different. In addition, it is very sensitive to the existence of unimportant attributes (including shadow attributes) in the information system. It also depends on the specific implementation of shadow attributes. Therefore, the process of regenerating shadow attributes needs to be repeated to obtain statistically valid results. In short, Boruta is based on the same idea of forming a random forest classifier, that is, adding randomness to the system and collecting results from a random sample set can reduce the

misleading influence of random fluctuations and correlations. Therefore, this additional randomness will provide us with a clearer view of which attributes are really important. The Boruta algorithm steps are as follows:

1. Assuming that the sample data X is m rows and n columns, there are m samples and n features, where m>1, n>1;

2. First copy the original feature sample X to obtain the copied feature sample X1;

3. Extract the copied feature sample X1 according to P(0<=P<1) to extract (mp)*n sets of samples. If m*p is not an integer, it can be rounded up and recorded as [m *p], when p= 1:00 is the original algorithm. Each column of the n columns of data is shuffled randomly, and then put back into the original feature sample X1 to obtain the current feature sample X1, which is still the m*n group of data, but is different from the original Compared with the algorithm, mixed and disturbed [m*p]*n sets of data;

4. Perform row transformation on the feature sample X1, randomly shuffle the row order, and get the shadow feature sample D;

5. Combine the original sample X with the characteristic sample D to obtain the final mixed sample;

6. Run the random forest regression model on the mixed sample, and calculate the mean reduction precision MeanImp of each variable no longer in the model;

7. Define the largest MeanImp in the shadow feature as MaxImp.

8. Based on the MeanImp of the original input feature, mark the feature variables larger than MaxImp as "important" features, and the others as "tentative";

9. Delete all shadow features;

10. Repeat steps 2-10 until all the feature importance marks are completed;

2.2 Light GBM.

Light GBM is an efficient distributed gradient boosting decision tree algorithm proposed by Microsoft in 2017 [3], which can be used for classification and regression tasks. Light GBM uses the histogram algorithm to find the best segmentation point of the data. Compared with the pre-sort traversal algorithm of the XGBoost algorithm, the histogram algorithm occupies less memory and effectively reduces the complexity of data segmentation.

To solve the time-consuming problem of training large samples of high-dimensional data, Light GBM adopts (GOSS) gradient-based unilateral sampling scheme, which can maintain accuracy while reducing data training. GOSS keeps all samples with larger gradients, and uses random sampling in samples with smaller gradients. When calculating the information gain, GOSS introduces a constant multiplier to the data with small gradients to offset the influence on the data distribution.

From the perspective of feature reduction, Light GBM uses (EBF) to bundle mutually exclusive features to improve computational efficiency. First, EBF sorts the features according to the number of non-zero values, then calculates the conflict ratio between different features, and finally traverses each feature and tries to merge the features to minimize the conflict ratio.

2.3 Bayesian Optimization.

Bayesian optimization for machine learning parameter tuning was proposed by J. Snoek (2012). The main idea is that, given the optimized objective function (generalized function, you only need to specify the input and output, without knowing the internal structure and mathematical properties.), by continuously adding sample points to update the posterior distribution of the objective function until the posterior distribution basically fits the true distribution. Simply put, it takes into account the information of the last parameter, so as to better adjust the current parameter.

Bayesian optimization has two core processes, Prior Function (PF) and Acquisition Function (AC). Acquisition function can also be called Utility Function (Utility Funtcion), but it is generally called acquisition function. PF mainly uses Gaussian process regression (it can also be other PF functions,

but Gaussian process regression is used more); AC mainly includes EI, PI, and UCB. At the same time, the balance of exploration and exploitation is also done through AC.

2.4 Evaluation Index.

In order to measure the model accuracy of the auto insurance claims model, it is evaluated by the accuracy rate, precision rate AUC value, recall rate, and F1 value indicators of the confusion matrix:

Table 1. Confusion matrix

Predict\Actual	P	N
P	TP	FN
N	FP	TN

TN: True Negative, negative samples are predicted as negative samples.

FN: False Negative, negative samples are predicted as positive samples.

FP: False Positive, positive samples are predicted to be negative samples.

TP: True Negative, positive samples are predicted to be positive samples.

AUC: The area under the ROC curve, which is between 0 and 1. The larger the AUC value, the higher the accuracy of the model.

$$\mathbf{accuracy} = \frac{\boldsymbol{TP + TN}}{\boldsymbol{TP + TN + FP + FN}}$$

$$\boldsymbol{precision} = \frac{\boldsymbol{TP}}{\boldsymbol{TP + FP}}$$

$$\boldsymbol{Recall} = \frac{\boldsymbol{TN}}{\boldsymbol{TN + FP}}$$

$$\boldsymbol{F1} = \frac{\boldsymbol{2 * Precision * Recall}}{\boldsymbol{Precision + Recall}}$$

Finally, for the evaluation of the model, this article also uses the ROC curve to visualize the effect of the two-class model. There are two parameters in the ROC curve, namely TPR and FPR. TPR is the ratio of the number of negative samples predicted to be negative samples to the actual total negative samples. The larger the value, the better. FPR is the ratio of the number of samples predicted to be negative samples but actually positive samples to the total number of actual positive samples. The smaller the value, the better.

$$\boldsymbol{TPR} = \frac{\boldsymbol{TP}}{\boldsymbol{TP + FN}}, \boldsymbol{FPR} = \frac{\boldsymbol{FP}}{\boldsymbol{FP + TN}};$$

3. Experiments and results

The auto insurance claims data comes from the RPubs public data set. The data contains a total of 27 features, which are divided into 4 types of owner attributes, vehicle attributes, policy attributes, and target attributes, with CLAIM_FLAG as the dependent variable. In this article, the operating platform is R7-4800HCPU, 16GB 3200MHz memory, the programming language is Python and common libraries (Pandas, Numpy, Sklearn, etc.). SVM, random forest, logistic regression are used for comparison to verify the superiority of the method.

3.1 Evaluation Index.

First, the missing values of the data are counted. In this article, Python is mainly used to count and fill the missing values of the data. According to insurance claims prediction, the missing values of the original data are mainly deleted and filled in two operations, and the missing values of the data are counted. As shown in Table 2, according to the data available in the table, all missing values are less than 30%, so the data is filled in. The missing value statistics are shown in Table 2, where the AGE feature and the Birth feature are duplicated, that is, the Birth calculation is used to fill in the

missing AGE attributes. After the calculation, it is found that the age is less than 18 years old, that is, a minor is unable to obtain a driver's license, and 7 sample data are judged as abnormal The data is deleted, and then the Birth feature is deleted. For the remaining five columns of features, the median is filled, and OCCUPATION is a type variable, usually filled with the mode, but for the accuracy of the data, the missing OCCUPATION feature is deleted, after deletion After keeping 9630 samples and removing the target features, each sample contains 24 features.

Table 2. Missing value statistics table

Columns	Type	Missing Num	Missing Radio
AGE	Int64	7	0.007%
YOJ	Int64	548	5.3%
INCOME	Int64	570	5.5%
HOME_VAL	Int64	575	5.6%
CAR_AGE	Int64	639	6.2%
OCCUPATION	Object	665	6.5%

There are a total of ten types of features in this data set, as shown in Table 3, which are coded step by step according to the meaning of the features. According to the attribute value, the values of PARENT1, MSTATUS, RED_CAR, REVOKED, URBANICITY, and GENDER can be feature dualized. Since the education level in the EDUCATION feature is from low to high, the Education feature can be converted to a corresponding increasing value. Type, and finally carry out one-hot encoding of CAR_USE, CAR_TYPE, and OCCUPATION, and delete this feature after one-hot encoding.

Table 3. Category feature description table

Columns	Type
PARENT1	[NO, YES]
MSTATUS	[NO, YES]
GENDER	[M, F]
EDUCATION	[PhD, High school, Bachelors, Master, <High School]
OCCUPATION	[NO, YES]
CAR_USE	[Private, Commercial]
CAR_TYPE	[Minivan, Van, SUV, Sports Car, Pickup, Panel Truck]
RED_CAR	[NO, YES]
REVOKED	[NO, YES]
URBANICITY	[Urban, Rural]

After preprocessing, the data contains a total of 36 features. In order to improve the learning rate and generalization ability of the model, the Boruta algorithm is used to perform feature selection on this data set.

3.2 Feature Selection.

The Boruta package in python is used for feature selection. After 44 rounds of iterations, 22 features are finally determined, 13 features are rejected, and 1 feature is uncertain. The iteration is shown in Fig 1. Finally, the ranking_ method in Boruta is called to view the importance of the features, the selected feature value is 1, the tentative feature is 2, and the rejected feature value is greater than 2, and the result is shown in Fig 2.

3.3 Auto Insurance Claim Prediction Model.

The modeling process of this article is as follows: Based on the Boruta feature selection result, the data set is divided into training set and test machine, and then Ligth GBM, random forest, SVM, logistic regression are selected for modeling, and the Light GBM effect can be obtained from Fig 3. Best, as shown in Table 4, Light GBM performs well for all indicators.

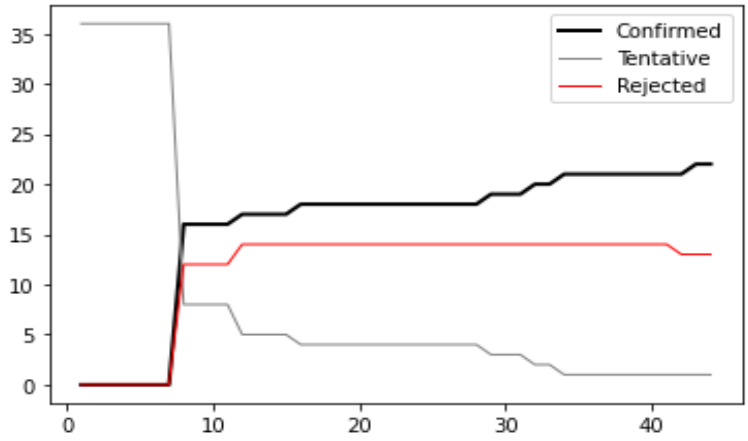

Fig. 1 Boruta Iterative

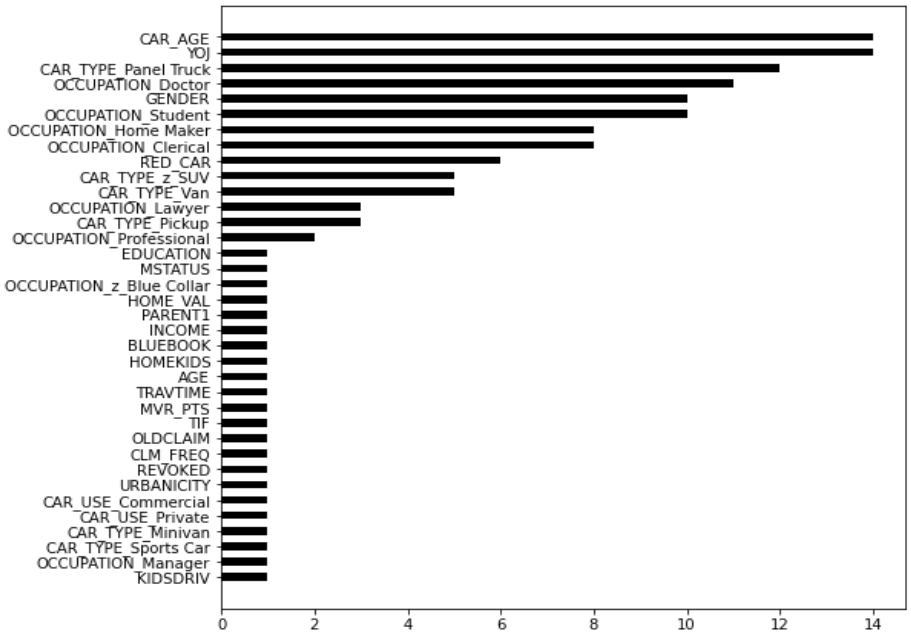

Fig. 2 Boruta-rank

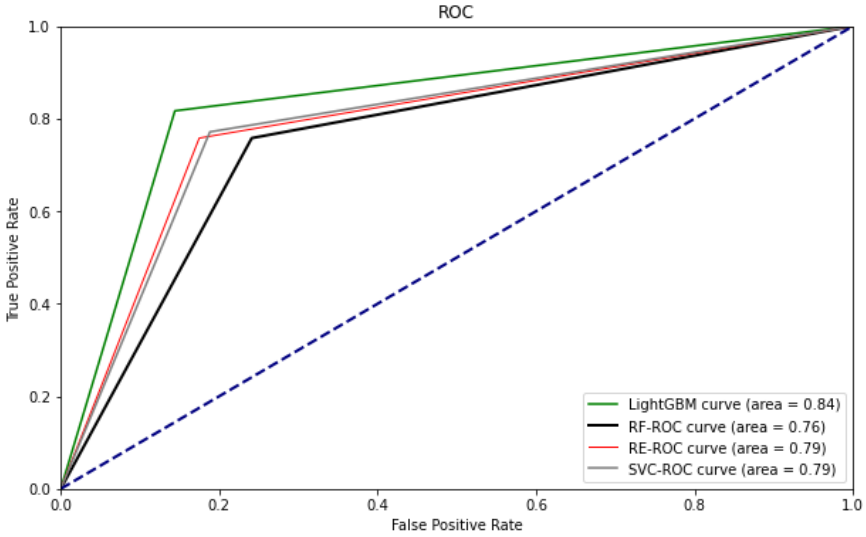

Fig. 3 ROC Curve Number of parameter optimization iterations

Table 4. Evaluation index of each model

Model	Accuracy	Precision	Recall	F1	AUC
Light GBM	0.836	0.837	0.836	0.836	0.836
RF	0.758	0.758	0.758	0.758	0.758
RE	0.791	0.792	0.792	0.791	0.792
SVM	0.791	0.792	0.792	0.791	0.792

On the basis of the default parameters, Bayesian optimizes the parameters of n estimators, Learning rate, num leaves, and max depth. The parameter ranges are shown in Table 4. After 30 iterations, the parameters are optimal at 28 iterations. Solution, the iteration diagram is shown in Figure 4. The AUC of the training set is close to 0.92, the Bayesian parameters are input into the model, the model is trained through the training set, and then the various indicators of the model are evaluated through the test set, and the model evaluation structure is shown in Table 5.

Table 5. Light GBM Parameter Range

Parameters	Value	BY
n_estimators	[100 :800]	546
learning_rate	[0.01 :1]	0.01
num_leaves	[20 :200]	112
max_depth	[2 :10]	10

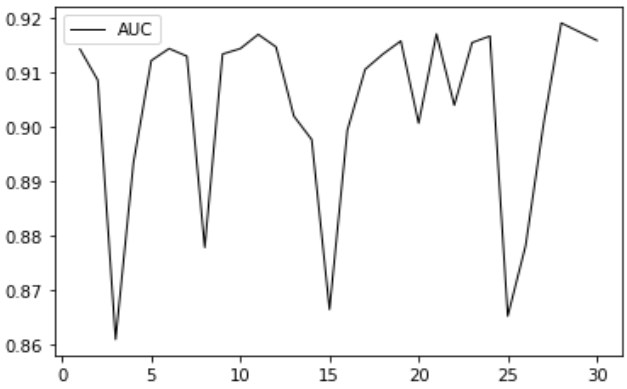

Fig. 4 Number of parameter optimization iterations

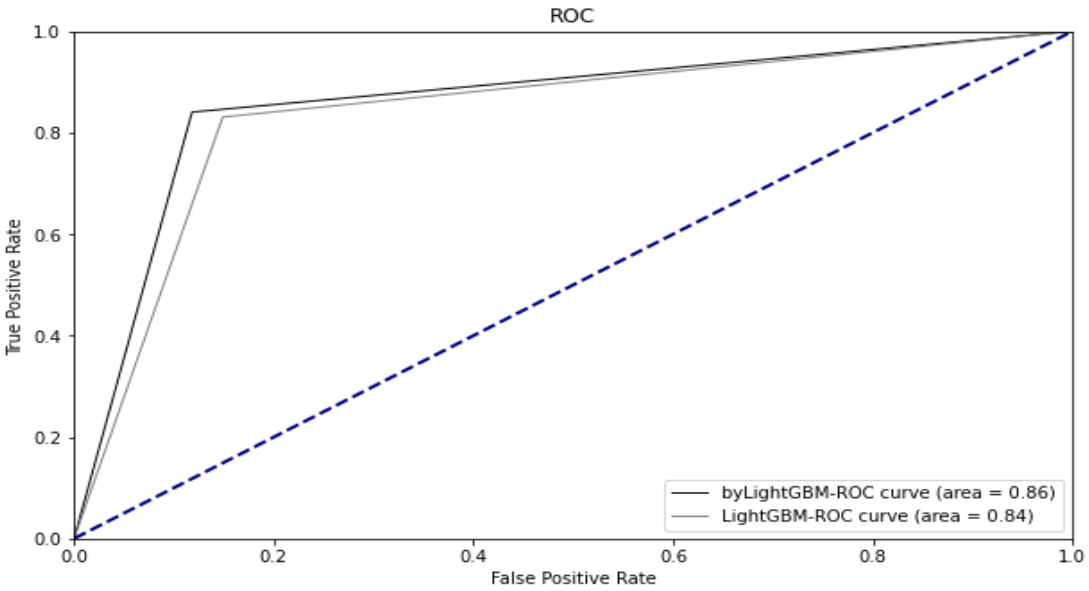

Fig. 5 Light GBM optimized based on Bayesian

Table 6. Light GBM optimized based on Bayesian

Model	Accuracy	Precision	Recall	F1	AUC
Light GBM	0.836	0.837	0.836	0.836	0.836
By- Light GBM	0.860	0.860	0.860	0.860	0.860

It can be seen from Figure 5 that the Light GBM model optimized by Bayesian parameters has improved under the five evaluation indicators, and is significantly better than random forest, SVM, and logistic regression. In summary, the Light GBM auto insurance claims prediction model based on Bayesian parameter optimization in this paper has good accuracy and generalization ability. In addition, Light GBM can output the feature contribution of each feature to the prediction result. Fig 6 shows the contribution of each feature output by the model. It can be seen that the six features of BLUEBOOK, INCOME, AGE, TRAVTIME, HOME_VAL, and OLDCLAIM have a significant impact on the results of claims.

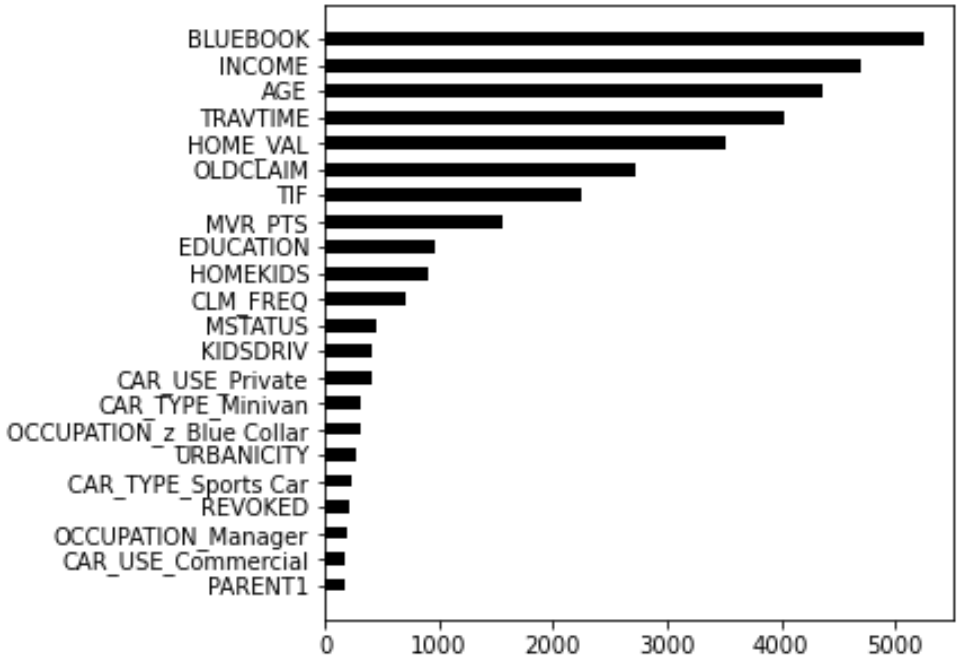

Fig. 6 Feature Importance

4. Conclusion

This paper proposes a car insurance claims prediction model based on Bayesian optimization of Light GBM, which can well realize the prediction of car insurance claims, and uses Boruta algorithm for feature selection in the feature extraction part. 22 features are selected from 36 features Modeling, while ensuring accuracy, simplifies the learning rate and improves the generalization ability of the model. In addition, Light GBM also provides a feature contribution method, which can effectively determine the main factors affecting auto insurance claims, and provide a certain reference basis for the determination of auto insurance rates.

References

[1] P. McCullag and J. A. Nelder, “Generalized Linear Models,” Chapman and Hall, London, 1983.

[2] W.S. Meng, L. Yang, et al. Random Effect Zero Inflated Claim Number Regression Model[J]. Statistical Research, 2015,32(11):97-102.

[3] W.S.Meng, et al.Neural network model and prediction of auto insurance claim frequency[J]. Statistical Research, 2012, 29(03):22-26.

[4] W.S.Meng, H.T.Wang et al. Individual Claim Reserve Evaluation Model Based on Machine Learning Algorithm[J]. Insurance Research, 2019(09):88-101.

[5] Y.Yang. Research on the Forecast of the Claim Amount in Auto Insurance[D]. Guizhou University of Finance and Economics, 2020.

[6] Y.Z.Zeng, A.B.Wu et al. Prediction of Auto Insurance Claim Frequency Based on Machine Learning[J]. Statistics and Information Forum, 2019, 34(05):69-78.

[7] Miron B. Kursa, Witold R. Rudnicki. Feature Selection with the Boruta Package[J]. Journal of Statistical Software, 2010,36(11).

[8] Abdulatif Aoihan Alresheedi,Mohammed Abdullah Al Hagery. Forecasting the Global Horizontal Irradiance based on Boruta Algorithm and Artificial Neural Networks using a Lower Cost[J]. International Journal of Advanced Computer Science and Applications (IJACSA), 2020, 11(9).

[9] Jerome H Friedman. Greedy function approximation: a gradient boosting machine. Annals of statistics, pages 1189–1232, 2001.

[10] Charles Dubout and François Fleuret. Boosting with maximum adaptive sampling. In Advances in Neural Information Processing Systems, pages 1332–1340, 2011.

[11] Tianqi Chen and Carlos Guestrin. Xgboost: A scalable tree boosting system. In Proceedings of the 22Nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pages 785–794. ACM, 2016.

[12] Greg Ridgeway. Generalized boosted models: A guide to the gbm package. Update, 1(1):2007, 2007.

[13] Qi M. LightGBM: A Highly Efficient Gradient Boosting Decision Tree. 2017.

[14] Blanchard Antoine,Sapsis Themistoklis. Bayesian optimization with output-weighted optimal sampling[J]. Journal of Computational Physics,2021,425.

[15] Srinivas, N., et. al. Gaussian process optimization in the bandit setting: No regret and experimental design. ICML 2010.

[16] Jones, D., et. al., Efficient global optimization of expensive black-box functions. J. Global Optimization, 1998.

[17] Jungtaek Kim, Michael McCourt, Tackgeun You, Saehoon Kim, Seungjin Choi. Bayesian optimization with approximate set kernels[J]. Machine Learning, 2021(prepublish).