Analytical Report

Yulin Zhou
i6336858

October 9, 2023

Abstract

For the convenience of reading and grading, the logic of this assignment is summarized below.

Section 1: Given the data set, I described it as the first step. Then, I found that the data was extremely unbalanced. To build a subset to validate the model, I split 10% of the original data(i.e., named Val.2 subset here) before class balancing.

Section 2: Before building the models, accuracy, precision, and recall were selected as the metrics to evaluate models. Then, there are five methods are applied in this document, including Decision tree, K-nearest neighbors, Logistic regression, Support vector machines, and Bayes model. Besides, their performances are also reported.

In Section 3: The univariate selection is implemented for feature selection. I select the top 10 ranked variables to keep, and drop others.

In Section 4: I compare the performance of the above five methods on the chosen metrics and analyze the effect of the feature selection. As we can see, the performance of most models is not reduced, and some of them are even slightly improved. Thus, feature selection works well in this case.

In Section 5: I briefly explain the characteristics of target customers based on the output of decision tree.

In Section 6: According to the requirement of this assignment, I apply the majority vote method as the primary principle, and the probability output of logistic regression as the secondary indicators to build the final prediction model.

As a result, there are 118 true target customers in the the predicted most promising 800 customers.

1 Data Exploration

1.1 Data Description

Given the labeled data set, firstly, I explore it. There are 85 independent variables and 1 dependent(i.e., predicted) variable. The label is a binary variable, with '1' presenting that the customer would buy the caravan policy, and '0' otherwise.

There are 5822 instances and only around 5.977% of them bought the insurance. Obviously, this is an unbalanced data set, and **class balancing** is needed to ensure a good performance for prediction.

As for the **missing data**, I found that there is no missing value. Therefore, I don't need to do anything for this part.

1.2 Data Pre-processing

1.2.1 Data Splitting

To train and evaluate the model, the data set is split here. Before class balancing, I split the 10% of the data as the Validation 2 data(i.e., Val.2) to see how well the trained models are on the unseen data set. Besides, for the methods of decision tree and KNN, Val.2 also could help me to decide the parameter. Therefore, a good parameter could be obtained instead of suffering from overfitting.

After the class balancing, for the decision tree and KNN, I randomly divided 70% of the balanced data as the training data, and 30% as validation 1 subset(i.e., Val.1). Considering some of the Val.1 is in the training data, thus the performance of the model on the Val.1 cannot be the only object for deciding the parameter.

To clarify, for the methods of logistics regression, SVM, and Bayes models, all of them use the whole balanced data to train the model, and Val.2 to evaluate the trained model. For the decision tree and KNN, both of them use training data to train the model and are evaluated by Val.1 and Val.2.

1.2.2 Class Balancing

There are only around 6% instances with the label '1'. Therefore, I did the class balancing by re-sampling.

After splitting Val.2, there are 4935 instances with the label '0' remaining, so I re-sample the remained data with the label '1' for 4935 times. Thus, I got the same number of instances with labels '1' and '0'. Then, I combine the remained data with the label '0' and the re-sampled data. Thus, the class-balanced data set with 9870 instances in total is obtained now.

By finishing this step, the trained model will have better performance in recall. Otherwise, even if the model predicts '0' for every instance, it would still have 95% accuracy, but really low recall.

2 Model Building and Training

In this section, I apply various methods to build the classification model to predict whether the customer would buy the insurance, including decision tree, logistic regression, support vector machines(i.e., SVM), and the Bayes model.

2.1 Choice of metrics

Before building the models, the metrics that serve for evaluating and selecting models are needed to decide.

Here I chose three metrics, namely accuracy, precision, and recall. Their calculation ways are shown below.

$$Accuracy = \frac{\text{True Positive} + \text{True Negative}}{\text{True Positive} + \text{False Negative}}$$

$$Precision = \frac{\text{True Positive}}{\text{True Positive} + \text{False Positive}}$$

$$Recall = \frac{\text{True Positive}}{\text{True Positive} + \text{False Negative}}$$

Accuracy could tell us the overall prediction accuracy. Since the customers who would buy the insurance are much more important, I introduce precision and recall to let me know the ability of the model to identify target customers. Specifically, precision could tell me how many percent of the predicted target customers are correct, and recall could show how many percent of the true target customers are identified by the models.

2.2 Decision Tree

In this part, I apply the decision tree method on the train set and try various depths to explore their performance on the validation set. The outcome is shown in the Fig.1.

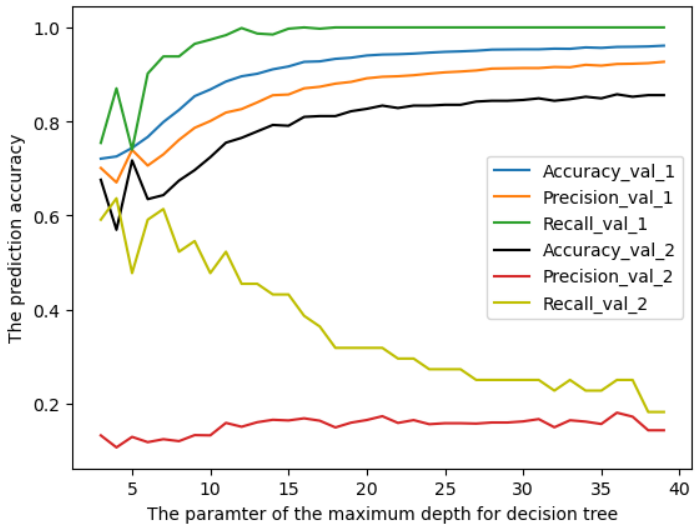

Figure 1: The performance of decision tree on different depths

As we can see, as for the performance of Val.1, all of the three metrics climb at first, and then keep almost the same. Recall always is in the lead, followed by accuracy, and finally Precision. The indexes are pretty good for Val.1, but quite not for Val.2. Although accuracy increases when the depth increases, the recall presents an opposite trend. Also, the precision is always really low(around 20%) on Val.2.

Based on Fig.1, it's reasonable to say that if we want to identify the almost true target, we have to tolerate an error rate of nearly 10%. In other words, about 10% of the predicted target customers would be wrong.

Besides, the complex decision tree is necessary, since the model with high recall is wanted.

2.3 K-Nearest Neighbors

There is an important parameter in the method of K-Nearest Neighbors, namely k, the chosen number of nearest neighbors. The outcome of the experiments on different k is shown in Fig.2.

It's interesting that all three metrics on Val.1 decrease when the value of k increases, but recall on Val.2 increases. Similar to the decision tree, the precision on the Val.2 is extremely low(around 7%).

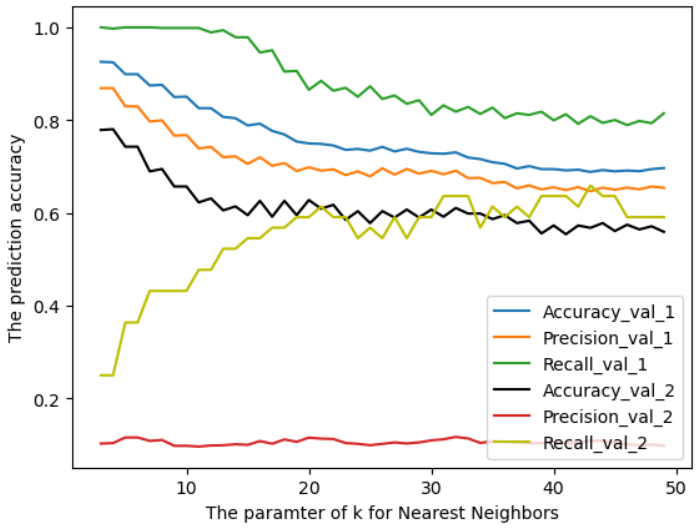

Figure 2: The performance of nearest neighbor on various k

2.4 Logistic Regression

As for the logistic regression method, its prediction performance on Val.2 subset is shown in Fig.3. In detail, the accuracy, precision, and recall are 0.67, 0.13, and 0.61 respectively.

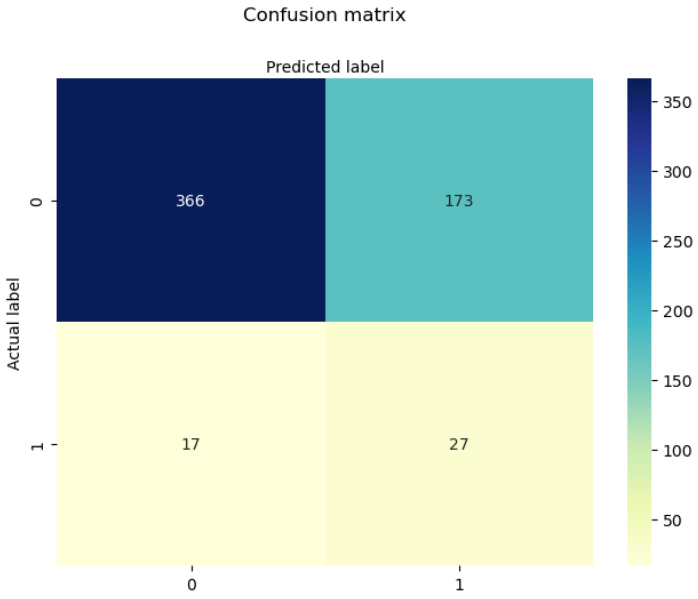

Figure 3: The performance of logistic regression on validation set

2.5 Support Vector Machine

In this part, I apply the SVM on the balanced data set and evaluate the model on Val.2 data set, which is as same as the previous part. The result shows that SVM could obtain 0.68, 0.13, and 0.56 on the accuracy, precision, and recall respectively.

2.6 Bayes Model

As for the performance of the Bayes Model, its accuracy, precision, and recall are 0.20, 0.08, and 1.0 respectively. Its recall is really high, but what to mention here is that the Bayes model predicts '1' for most of the instances in Val.2. Actually, the Bayes model doesn't work here.

3 Feature Selection

In the section 2, I build multiply models on the all available variables(i.e., 84). However, there is a high chance that some of them are not relevant to the predicted variables. Although the obtained models are good enough to apply to the unseen data, the models are difficult to explain and interpret. For example, the optimal decision tree in the section 2.2 is not human-friendly. Thus, feature selection is needed if an explanation is required.

3.1 Univariate Selection

Statistical tests can be used to select features that have the strongest relationship with the output variable. Here I use the 'SelectKBest' in the scikit-learn library to select useful features for prediction. Specifically, the chi-squared statistical test is used in this part.

Fig.4 below shows the score curve of the ranked variables. Obviously, there is a drop at the beginning and then followed by a gentle descent stage. After that, the scores of variables ranked at the bottom are almost equal and really low. The red line in Fig.4 splits the top 10 variables with the highest score. The selected variables are listed in the table. Then, I only use the top 10 variables to build new models for every classification model-building method mentioned above.

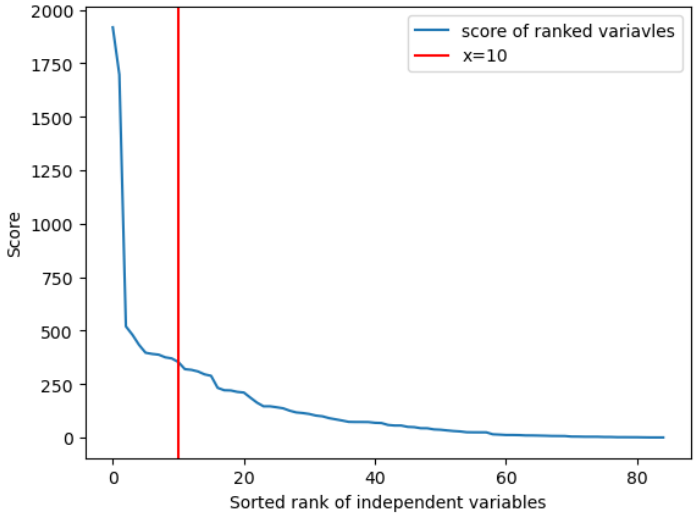

Figure 4: The curve of ranked score of independent variables

Table 1: The selected top 10 ranked variables

No.	Variable	Score	No.	Variable	Score
1	Contribution car policies	1918.2	2	Customer Subtype	1696.7
3	Rented house	519.4	4	Contribution fire policies	480.1
5	Income < 30.000	433.2	6	Lower level education	396.2
7	High level education	390.7	8	Number of car policies	387.2
9	Contribution boat policies	374.8	10	Home owners	369.6

4 Comparison of Classification Models and Influence of Feature Reduction

After feature selection, new models are required to be trained, and the processes are the same for section 2. The performance of the decision on different maximum depths and the KNN model on the various k are shown in Fig.5 and Fig.6. Besides, the detailed values are in the Fig.7.

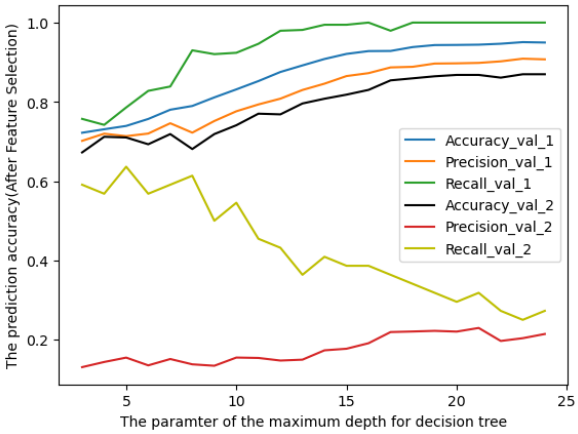

Figure 5: The performance of the decision on different maximum depths(After Feature Selection)

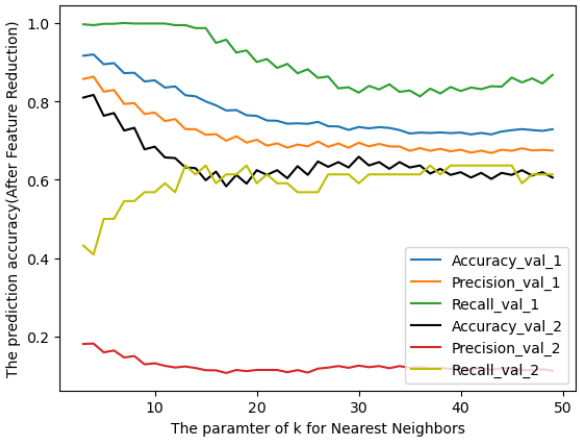

Figure 6: The performance of the decision on different k(After Feature Selection)

	Accuracy	Precision	Recall
dtc_allvar	0.569468	0.106464	0.636364
dtc_select	0.710120	0.154696	0.636364
knn_allvar	0.567753	0.109023	0.659091
knn_select	0.631218	0.123348	0.636364
logistics_allvar	0.674099	0.135000	0.613636
logistics_select	0.660377	0.133333	0.636364
svm_allvar	0.686106	0.132275	0.568182
svm_select	0.603774	0.118367	0.659091
bayes_allvar	0.204117	0.086614	1.000000
bayes_select	0.792453	0.153153	0.386364

Figure 7: The performance of all best models of every method

5 Characteristics of Target Customers

In order to describe the characteristics of the target customers, a simple model is needed. Therefore, I use the selected top 10 variables to train the model. As for the classification method, the decision tree is a good option when an explanation is needed. The outcome is shown in the Fig.8.

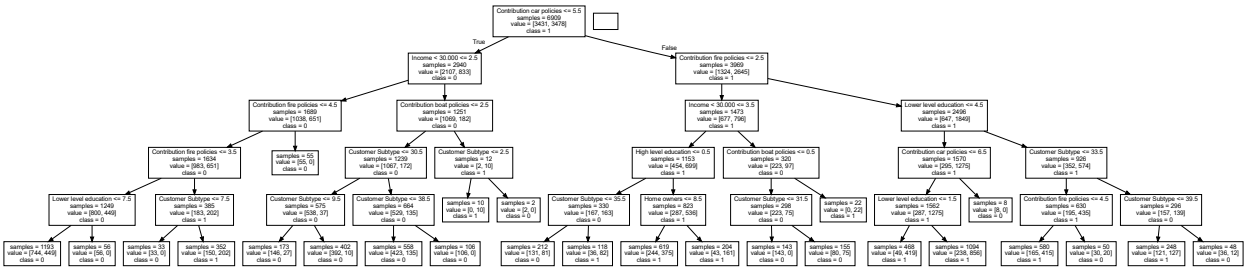

Figure 8: The decision tree(depth=5) trained from selected variables

As we can see, in general, the lower the level of education, the fewer contribution boat policies, and the less contribution fire policies, the less likely the family will buy the insurance. However, the affections among variables are very complex.

To have a better understanding, the shallow decision tree is shown in Fig.**??**.

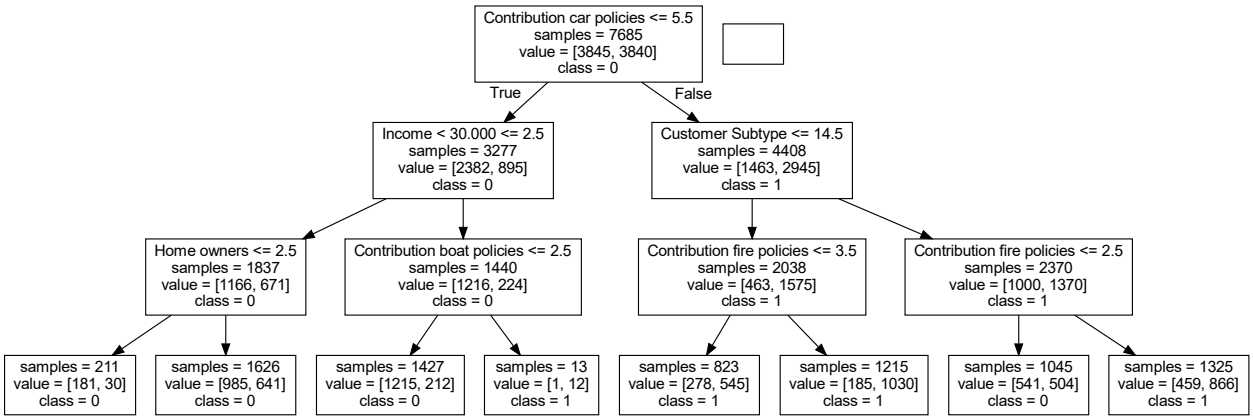

Figure 9: Shallow decision tree(depth=3) trained from selected variables

6 Customer Selection

Before customer selection, I build a majority vote model, which considers the number of votes first, and then the probability from the outcome of the logistics regression.

As a result, I get 118 of the predicted 800 promising customers.