Health insurance lead prediction with machine learning algorithms

Anja Hrvatič,
Faculty of Computer and Information Science, University of Ljubljana

Abstract—It is important for insurance companies to be data-driven. Big amounts of data on their customers, policies, and claims is generated and collected as a result. It can be used for marketing purposes. This work focuses on predicting customer's response when they are offered a health insurance policy on the insurer's website. With data on customers and policy, they were recommended, prediction models were built. The article describes the preprocessing of data and how to obtain optimal results with the used algorithms. Linear regression, decision tree, k-nearest neighbors, random forest, and stacking model were used to predict the response.

Index Terms—Machine learning algorithms, recommendation system, health insurance.

1 Introduction

INSURANCE companies are aware of the importance of collecting and analyzing data on their activities. They can then use it to describe and predict various trends relevant to their business. This article explains ways to use data on potential customers to predict their response when they were presented with the application form for health insurance on the insurance company's website.

The data contained basic information about customers, their current policy, the policy they were recommended, and their response. As this is a binary classification problem, chosen machine learning algorithms were logistic regression, k-nearest neighbors, decision tree, random forest, and naive Bayes. In the end stacking of all 5 algorithms was performed to see if the stacked model performs better than each model individually.

In the beginning, we touch on data analysis and its preprocessing for machine learning. Due to the imbalance of the classes of the target variable, a way of overcoming this problem is described. Then all the used algorithms are described. To evaluate the models, cross-validation and classification accuracy were used and are discussed in the literature review section. Thereafter methodology part follows, where the used methods are described and explained. In the results section, the performance of individual prediction models is presented and explained. Models are then compared with each other to obtain the best three models according to different assessment criteria.

2 Literature review

To predict the customer's response machine learning algorithms were implemented. To choose the appropriate algorithm, it is essential to understand how they work. Since the problem was a case of a binary classification problem we chose logistic regression, k-nearest neighbor, decision tree, random forest, and naive Bayes classifiers as potential models. In the end stacking of those algorithms was performed as it can give us a model with higher classification accuracy. Below are brief descriptions of these algorithms.

2.1 Logistic regression

Logistic regression is a method that predicts the odds ratio of the outcome depending on the feature variables (covariates). It is a case

- *E-mail: ah40785@student.uni-lj.si*

of generalized linear models as it assumes a linear relationship between the logit function of the outcome and the predictor values. By comparing the probability with a pre-defined threshold, the object is assigned to a class accordingly (figure 1). For logistic regression, the target variable must be categorical and feature attributes need to be discrete [1].

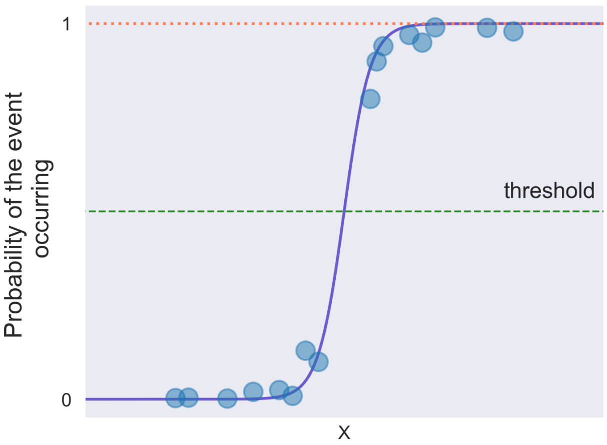

Figure 1. **Logistic regression.** The sigmoid function generates a probability output. According to the threshold, outputs are classified.

2.2 K-nearest neighbours

The k-nearest neighbors algorithm uses the concept of distance for classifying data objects. It calculates the distances between the tested, and the training data samples to identify their nearest neighbors. The tested sample gets assigned to the class of its nearest neighbor. The parameter k decides how many nearest neighbors influence the classification. Many distance functions can be used to evaluate the distance [2].

2.3 Decision tree

The decision tree consists of internal nodes, edges, and terminal nodes (leaves). Internal nodes correspond to attributes, edges correspond to subsets of attribute values and leaves correspond to class labels. Each path from the root to the leaf represents a decision rule. The method is based on selecting the best attribute on every level of the tree. Information about the class frequencies can be found in a leaf. They can be used to evaluate the class probability distribution in the leaf. The total number of learning samples in it denotes the reliability of the estimation [3].

2.4 Naive Bayes

Bayes' Theorem enables the evaluation of the probability of a hypothesis given our prior knowledge. It is given by equation 1:

$$P(h|d) = \frac{P(d|h) \times P(h)}{P(d)} \tag{1}$$

Where P(h\d) is the probability of hypothesis h given d, also called posterior probability, and P(d\h) is the probability of d given that hypothesis was true. P(h) is the probability of hypothesis h and is called the prior probability of h. P(d) is the probability of event d happening. The goal is to calculate the posterior probability for different hypotheses. The hypothesis with the highest probability is called the maximum posterior hypothesis.

Naive Bayes classification simplifies the problem by assuming conditional independence of attributes. This is a strong assumption that is very unlikely in real data, but the model can perform very well nonetheless [4].

2.5 Random forest

Random forest is a type of ensemble machine-learning algorithm. Random forests are improvements over bagged decision trees. Even with bagging, decision trees can have a lot of structural similarities which results in high correlations between their predictions. Random forest algorithm changes the way the sub-trees are learned so the predictions of different trees are correlated to a lesser extent. The random forest does not have all the attributes available to look through to find the best one. It is limited to a random sample of features. The number of them has to be specified as a parameter [4].

2.6 Stacking

Stacking uses a variety of models to create a new model. It selects different trained models to design a general model called stacked generalization. Stacking learns how to best combine predictions of the contributing models. The architecture usually involves two or more base models and the meta-model that combines their predictions. The meta-model is therefore trained on the predictions made by base models. In the case of classification, the inputs of the meta-model are class labels. Stacking is appropriate when predictions made by base models have low correlation [5].

3 Methodology

First, the data was analyzed to find interesting features and anomalies. We checked if the target classes were balanced in if there were any missing values. The imbalance was dealt with random oversampling and missing values were replaced with the mode of the attribute. To see if there are any obvious connections visualizations were created.

The models were built using the python scikit-learn module. 10-fold cross-validation was used to evaluate the performance of obtained models. It split the data into ten parts and repeated the algorithm ten times. For comparison, the models were also trained and assessed on split data. The training set accounted for 70 % and testing for 30 % of the data. Before running the machine learning algorithms we determined the best parameters. For every model, the confusion matrix was checked to calculate the classification accuracy. The standard deviation of classification accuracy was evaluated as well. Both of them served as criteria for their performance.

4 Data overview

The dataset had 14 variables. The connection between response City_Code, Health_Indicator, and Holding_Policy_Duration was checked. No obvious correlations between them were found. Figure 3 shows the correlation between response and Accomodation_Type, Reco_Insurance_Type, and Is_Spouse. For the attributes Holding_Policy_Duration and Holding_Policy_Type there were 40 % of missing values. For the attribute Health_Indicator, there were 23 % missing values. All of them were replaced by the mode of each attribute so that they could still be used for machine learning. Figure 2 shows an imbalance in the response classes which is our target variable. This was dealt with random oversampling as having imbalanced responses could result in a bias toward the negative response.

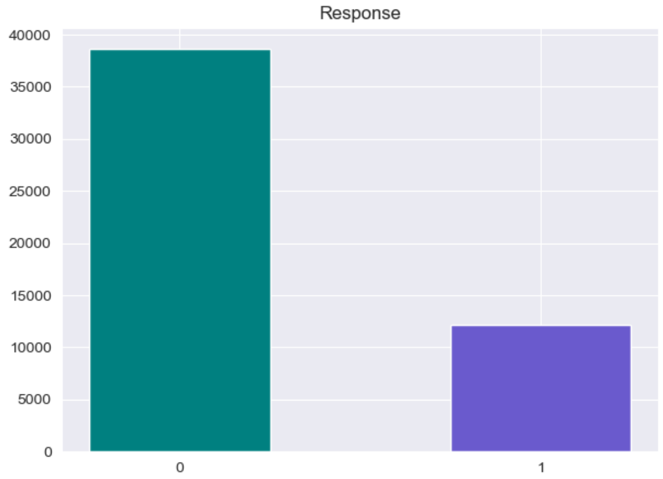

Figure 2. **Imbalance of response classes.** Response 1 means that person applied for the policy. The figure shows an imbalance in the target variable. There is way fewer positive responses.

5 Results

Many machine learning algorithms have parameters that can be tuned to obtain better results. Before running the algorithms optimal parameters were identified. Then all algorithms were run. To compare their performance classification accuracy and standard deviation were evaluated for each model. The results are summarized in the table 1, followed by a more detailed description of the optimal parameters and results for each model that was evaluated with cross-validation. Comparison is also shown in figure 4

Table 1 and figure 4 show that the best model is the model obtained by stacking all individual

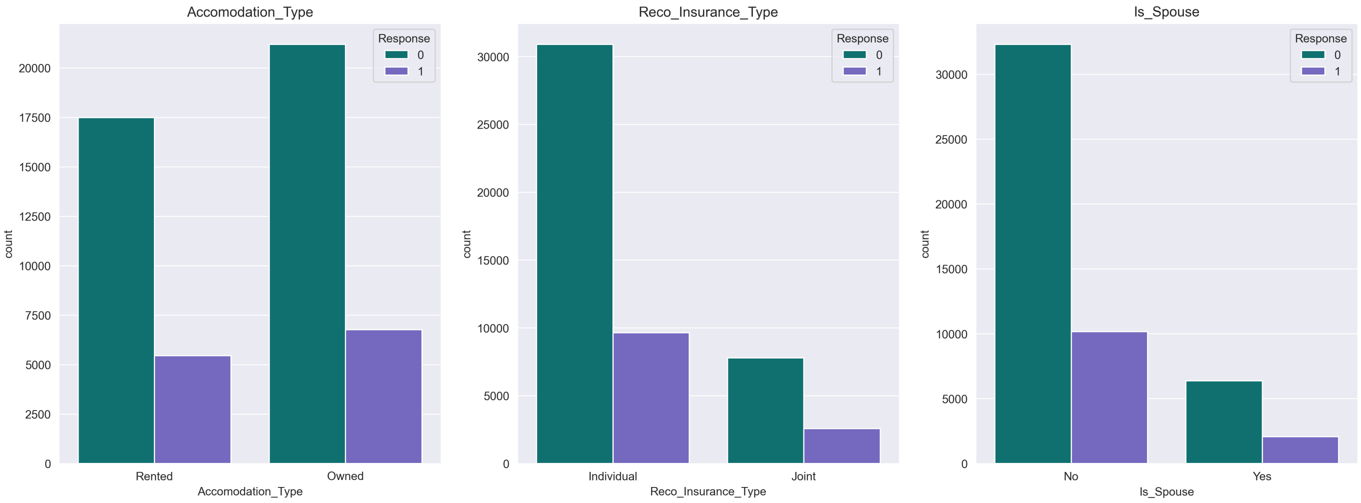

Figure 3. **Correlation between response and different attributes.**

Table 1
Table of classification accuracy (CA) and standard deviation (STD) for each model.

Algorithm	CA	STD
Logistic regression	0.544	0.006
KNN	0.851	0.004
Decision tree	0.868	0.005
Naive Bayes	0.550	0.005
Random forest	0.940	0.003
Stacking	0.966	0.002

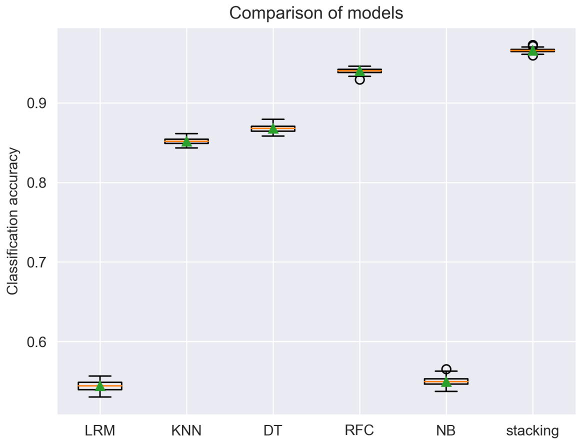

Figure 4. **Comparison of models.** Figure shows the distribution of classification accuracies for each model.

models. Besides stacking, random forest performed the best. Models K-nearest neighbors and decision tree had similar success. Logistic regression and naive Bayes classifiers failed to predict the client's response. From figure 4 it can also be noticed that the variance values are small, especially for the better-performing models. Low variance demonstrates the stability of cross-validation for those models.

5.1 Logistic regression

There were no specially set parameters for logistic regression. The model was able to predict positive responses but failed at predicting negative ones. The classification accuracy for the model was 0.544 with a standard deviation of 0.006. For comparison confusion matrix on split, data was plotted to show the success of predicting each class. This is shown in figure 5. The model that was trained and tested on split data had a classification accuracy of 0.545.

5.2 K-nearest neighbours

Parameter k stands for the number of neighbors that influence classification. In this case, it was determined that the best k value is 1. The distance metric was set to Minkowski which results in the standard Euclidean distance as p was set to 2. For comparison confusion matrix on split data was plotted to show the success

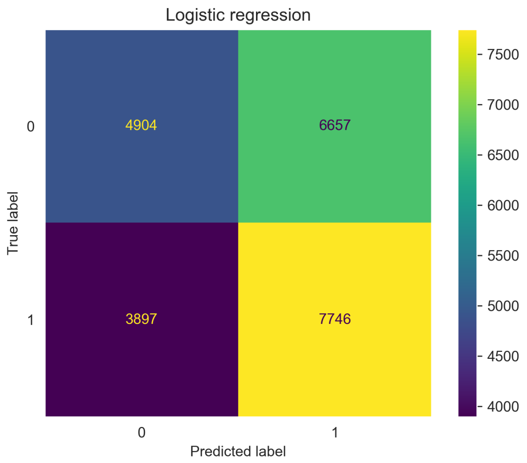

Figure 5. **Logistic regression classifier confusion matrix.** The model is good at predicting positive responses but fails at predicting negative ones.

of predicting each class (figure 6). It predicted a positive response to a very good extent. The positive class is still predicted better than the negative one but the prediction of both labels improved compared to logistic regression. The classification accuracy when cross-validation was used, was 0.851 with a standard deviation of 0.004. From split data the model's classification accuracy was 0.814.

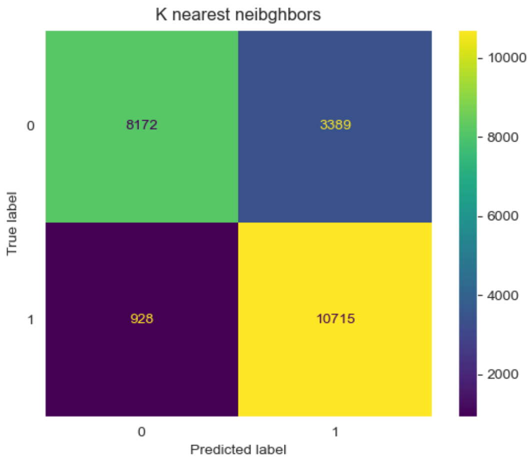

Figure 6. **K-nearest neighbors confusion matrix.** The model is good at predicting positive responses but is weaker at predicting negative ones.

5.3 Decision tree

Gini was chosen as a function that measures the quality of each split. The nodes were expanded until all leaves were pure or until they contained less than 2 samples as at least 2 samples are required to split an internal node. For comparison confusion matrix on split data was plotted to show the success of predicting each class (figure 7). The model was biased towards the positive class. The classification accuracy when cross-validation was used, was 0.868 with a standard deviation of 0.005. The accuracy for the model with the same parameters obtained with split data was 0.83.

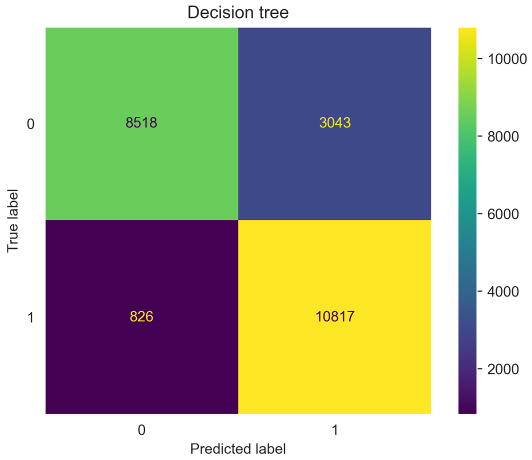

Figure 7. **Decision tree confusion matrix.** The model is good at predicting positive responses but is weaker at predicting negative ones.

5.4 Naive Bayes

For the Naive Bayes model, prior probabilities were adjusted according to the data. Parameter var_smoothing was set to 2.310e-09. This parameter represents the portion of the largest variance of all features that are added to variances for calculation stability. The classification accuracy of the model evaluated with cross-validation was 0.550 with a standard deviation of 0.005. The accuracy for the model with the same parameters obtained with split data was 0.552. The confusion matrix for the latter approach is shown in figure 8. The model was able to predict the positive response but did not succeed in predicting the negative class.

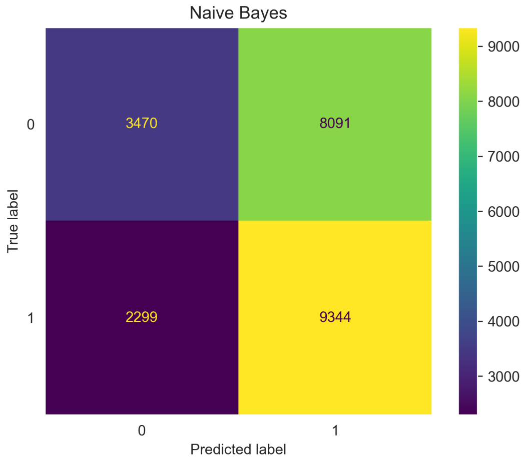

Figure 8. **Naive Bayes classifier confusion matrix.** The model is good at predicting positive responses but fails at predicting negative ones.

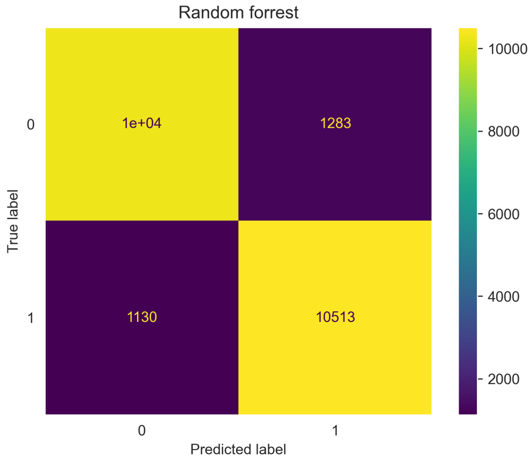

Figure 9. **Random forest classifier confusion matrix.** The model is good at predicting both responses.

5.5 Random forest

The random forest consisted of 100 trees. The function that measured the quality of each split was gini. Nodes were expanded until all leaves were pure or they contained less than two samples as 2 was the minimum number of samples required to split an internal node. The model evaluated with cross-validation had a classification accuracy of 0.940 with a standard deviation of 0.003. The accuracy of the model obtained with split data was 0.895. Its confusion matrix is shown in figure 9.

5.6 Stacking ensemble method

Outputs of described estimators were inputs of a final estimator. The final estimator was the logistic regression model and it was evaluated using cross-validation. The classification accuracy was 0.966 with a standard deviation of 0.002. Stacking was the best model obtained in this research.

6 Conclusion

Different approaches to solving the problems were explained. The obtained models were compared according to their classification accuracy and standard deviation. First, models were built on data using 10-fold cross-validation technique. The results were compared to those obtained by training and evaluating the model on split data. 70 % of the data was used for training and 30 % was used for testing.

Cross-validation gave better results than splitting. The three best models were:

1) Stacking model (CA=0.966, STD=0.002).
2) Random forest (CA=0.940, STD=0.003).
3) Decision tree (CA=0.868, STD=0.005).

Predictions with stacking are good and could be used for recommending health insurance policies to customers. It is expected that the results of the random forest model will be similar to or better than the results of the decision tree since the random forest is an ensemble of decision trees. Models evaluated on split data were often biased towards positive response, especially logistic regression and naive Bayes classifier. This can be the result of data preprocessing, as filling in missing values with the mode can generate a bias. It could also be the result of random oversampling that was used for balancing the response classes.

In general, the obtained results were good but predictive models could be furtherly improved. For further research different approaches for dealing with missing data and target variable classes imbalance could be tested. Regarding stacking, hyperparameters of the final estimator (logistic regression) could be finetuned. Different final estimators might give better results so it would be worth testing them.

References

[1] E. Y. Boateng and D. A. Abaye, “A review of the logistic regression model with emphasis on medical research,” *Journal of Data Analysis and Information Processing*, vol. 07, pp. 190–207, 2019.

[2] N. Ali, D. Neagu, and P. Trundle, “Evaluation of k-nearest neighbour classifier performance for heterogeneous data sets,” *SN Applied Sciences*, vol. 1, 12 2019.

[3] I. Kononenko and M. Kukar, *Machine Learning and Data Mining*. Elsevier Science, 2007. [Online]. Available: https://books.google.si/books?id=G_rgpjsnsG8C

[4] J. Brownlee, “Master machine learning algorithms discover how they work and implement them from scratch i master machine learning algorithms ,” 2016. [Online]. Available: http://MachineLearningMastery.com

[5] I. Czarnowski and P. Jedrzejowicz, “Stacking-based integrated machine learning with data reduction,” vol. 72. Springer Science and Business Media Deutschland GmbH, 2018, pp. 92–103.