Foundations of Data Science - Final Project
Insurance Claims Over Cars

Team Members:
Hassan Teymoori - 1947458, Mohammadreza Shabani - 1966731
Mehrdad Hassanzadeh - 1961575, Sara Zeynalpour - 1997862
Mahsa Barghi Mehmandari - 1959831

Abstract

Data Science and Machine Learning approaches are helping us to make a prediction about something. Here with the power of Machine Learning algorithms we want to predict if a customer who is applying for insurance will claim his insurance from the insurance company or not, depending on his personal previous data. We trained two models Logistic regression and Naive Bayes models in order to classify our dataset which respectively gave us 85 and 81 percent in terms of accuracy.

Introduction

We all know that with the help of Data Science and Machine Learning approaches we are able to make a prediction about something. This prediction can be related to the climate changes, fluctuations in financial markets, possibility of having cancer of a patient or etc.

We are able to offer insurance companies a system that will help them to gain more profit by just giving insurance to the applicants who are less likely to claim their insurance in future. To do so, we use two main approaches Logistic Regression (LR) and Naive Bayes. We will see that in order to have a better prediction power we take into account the regularization of the approaches and also use Cross Validation to take the best model among the possible ones. At the end, we will show our results based on a dataset which was taken from Kaggle.

Related Works

We searched a lot, but we could not find any related work with this topic except these following articles which are about "Strategies for detecting fraudulent claims in the automobile insurance industry".

Comité Européen des Assurances (1996)
Coalition Against Insurance Fraud (2001)
Insurance Information Institute (2004)

Dataset and Benchmark

The dataset in this project has been taken from Kaggle (Link to Dataset). A company has shared its annual car insurance data that contains 10k samples and 19 columns (10000 x 19). The columns resemble practical world features. The 'OUTCOME' column (Target column) indicates 1 if a customer has claimed his/her loan; otherwise 0 (31.33 percent claimed and 68.67 percent not claimed).

The dataset contains 7 categorical and 11 numerical features (53 percent numerical, 47 percent categorical).There exist several categorical features that were encoded by "one-hot encoding"). There is no duplicate record in the dataset, while there are 982 missed values in 'CREDIT_SCORE' and 957 in 'ANNUAL_MILEAGE' columns that were handled by imputation and dropping. Since the percentage of missing values is not high and imputing the missing values did not improve the performance, therefore we decided to comment it out due to computation power, instead we used dropping as the solution.

By plotting the distribution of each column, it was found that some features have normal distribution while there are a couple of skewed features, either positively or negatively. The distribution of each feature with respect to the target value was examined as well. Furthermore, the correlation of each pair of features alongside their correlation with the outcome have been observed. "RACE" and "VEHICLE_TYPE" were eliminated from the dataset due to their low contribution to the target value.

With the help of boxplot, we checked the possible outliers in each feature. The Interquartile Range (IQR) method was implemented as a mathematical tool to identify outliers. Finally, the Winsorization transformation was chosen to handle the outliers.

After all of the mentioned modifications, we were left out with 5613 samples of class 0 and 2536 samples of class 1.

Exploring the internet, we could not find any related dataset to consider as a real dataset to analyse our trained model. As a result, to have a benchmark, we decided to split our dataset into two different test and training portions. This technique is also very effective in order to prevent information leakage.

Methods

Among several Machine Learning algorithms, we selected Logistic Regression and Naïve Bayes to tackle our case. A combination of 5-fold cross validation and hyperparameter tuning, applied on the training portion, was used to find the parameters of the best Logistic Regression model.

Furthermore, in order to be able to apply a combination of feature engineering techniques and learning process specifically on the training set, we used the notion of pipeline. After that, the same transformers were applied to the test set.

In Logistic regression, we adjusted the following parameters to different values: - C, Penalty: the regularization parameters.
- Max_iter: maximum number of iterations (We noticed that iterations higher than 100 did not affect the performance).
- Class weight: used since our target values were not completely balanced.

- fit-intercept: the “bias” (False: without bias, True: with bias)
- Solver: optimization function.

As in Naive Bayes we cannot define proper parameters which lead to completely different estimators, we didn’t apply hyperparameter tuning for this method.

Experimental Results

Using the classifiers mentioned in the method section, the two analytic methods were executed over the splitted Insurance Claims Dataset. The results of the two classifiers are recorded under the 80:20 test option. The evaluation metrics that we obtained from the Logistic regression and Naive Bayes classifiers are summarized in Table I.

Table I. Evaluation results

Methods	Accuracy	Precision	Recall	F1-Score	ROC-AUC
Logistic Regression	0.85	0.84	0.84	0.84	0.80
Naive Bayes	0.81	0.81	0.81	0.81	0.76

Parameters that we got for the best model are: C:6.3, Penalty:L1, Max_iter:10, Class weight: 0.72, 1.6, fit_intercept: True , Solver: Liblinear
The Logistic Regression algorithm outperforms the Naive Bayes in all metrics. More precisely, the Logistic Regression yields 85 percent accuracy under 5fold-CV, which is 4 percent higher than the accuracy obtained by Naive Bayes. By other test options, like 50:50, we obtained the same result as in Table I, which is a bit surprising since we expected that Naive Bayes would perform better than Logistic Regression on the smaller dataset.

The result of our experiment is not consistent with the literature. For instance, Ng and Jorden (2002) suggested a mathematical proof of error properties of both models, according to which, when the training size reaches infinity, the discriminative model including the Logistic Regression performs better than the generative model like Naive Bayes. Ng and Jorden found that the generative model reaches its asymptotic faster (O(log n)) than the discriminative model(O (n)), i.e., the generative model (Naive Bayes) reaches the asymptotic solution for fewer training sets than the discriminative model (Logistic Regression). However, contrary to Ng and Jorden, we could not encounter this behavior in our comparison even with the 90:10 percent of division.

Conclusion and Future Work

In our case Logistic Regression worked better than Naive Bayes. However, we assumed that by having small size datasets Naive Bayes should work better than Logistic Regression. We believe that our result can be explained by the nature of Big Data’s interdependence. More precisely, Naive Bayes assumes that the features are conditionally independent, meaning that all of the features are independent from one another. Real datasets are however never perfectly independent, rather they can be quite interdependent. In short, as we discussed in the class, Naive Bayes method has a higher bias but lower variance, compared to Logistic Regression. If the dataset follows the bias, then Naive Bayes will be a better classifier. Based on the correlation heatmap, as shown on the notebook, we assumed that there are some correlations between different features and this can explain why Naive Bayes underperformed in comparison to Logistic Regression.

As a future work, the performance of other methods such as Decision Trees, Random Forest, Support Vector Machine (SVM) and Multi Layer Perceptron (MLP) can be examined. Also using approaches like DownSampling, UpSampling and SMOTE can be used to make the dataset completely balanced with respect to the target value.

References

1. Ng, A. Y., Jordan, M. I. (2002). On discriminative vs. generative classifiers: A comparison of logistic regression and naive bayes. In Advances in neural information processing systems (pp. 841-848). [Link]
2. Compare the effect of different scalers on data with outliers, scikit-learn official webpage, [Link]
3. Implementation of the linear model for Logistic regression, scikit-learn official webpage, sklearn.linear_model.LogisticRegression [Link]
4. Cross-validation: evaluating estimator performance, scikit-learn official webpage, [Link]
5. An introduction on how to fine-tune Machine and Deep Learning models using techniques such as: Random Search and GridSearch, Hyperparameters Optimization [Link]
6. Cross-Validation and Hyperparameter Tuning: How to Optimize your Machine Learning Model [Link]
7. Every data scientist has a tab open to StackOverflow[Link]