COMPARING LOGISTIC REGRESSION, DECISION TREE, AND k-NN TO CLASSIFY CHURNING CUSTOMERS IN A BANK

Introduction

Against the context of a bank, we are hooked on the idea of finding out whether a specific customer would leave the credit card services that they are using from the bank or not. Having an idea of that would do bank officers a good turn since they would be able to pay better care to these customers.

The report incorporates five parts: Introduction, Problem formulation, Methods, Results, and Conclusion. In the introduction, we have offered you a brief introduction of the motivation for this ML project from the banking industry perspective. The second part would place emphasis on the first component of ML which is data. Then comes the third one which is the Methods section where models and loss function are discussed. In the Results, we picked the model and assessed the performance on the test set. Finally, the Conclusion briefly gave a short discussion of future work for improvements.

Problem Formulation

From the dataset, each credit card service user is considered to be a data point. There are 19 features. The property of interest-the label- in this case is “if the customer leaves the current credit card services?”. It is a binary variable taking on either ‘Existing’ or ‘Attired’ value. (More details in the table below)

Features	Quantity of interest(label)
Age	If this customer leaves the current credit card services?
Gender	
The total number of people dependent on the financial support of the credit card user	
Education level	
Marital status	
Income	
Card category	
Time of relationship	
Total number of products held by the customers	
Total number of months being inactive in the last 12 months	
The number of contacts in the last 12 months	
Credit limit	
The total amount that credit card users don't pay on time	
The total amount left to use in the credit card	
Change in transaction amount (Q4 over Q1)	
Change in transaction count(Q4 over Q1)	
The total transaction amount(12 months)	
The total transaction counts(12 months)	
Average card utilization ratio	

Methods

Dataset description: I have 10127 data points in total (no missing values). I found this nice dataset from the site:

https://www.kaggle.com/sakshigoyal7/credit-card-customers.

Hypothesis space: From the outset, I could tell that my label y is not continuous but rather categorical. Then, it is safe to say that this is the classification problem rather than regression. There are several hypothesis spaces for classifying data points that we have had a chance to discuss in chapter 3 of the course book. In this ML Project, I grabbed three of them which are firstly the logistic regression. It learns a predictor from the linear hypothesis space (Jung, 2021). Secondly, the decision tree which is considered to be piecewise-constant over the regions of the feature space X. For each leaf node of a decision tree, there would be a corresponding region **(**ibid**)**. And the last one is the k-nn, which uses metric hypothesis space. The idea of k-nn classifier is to label a new data point on the basis of the *k* nearest labeled data points **(**ibid**)**. Three models used in this project come from different types (linear, piecewise-constant, and metric) of hypothesis space.

Loss function: In an effort to measure the quality of my classifier map, I called on the zero one loss. A straightforward reason for picking this type of loss function could be that the accuracy score and the zero-one loss are rather related to each other. To be specific, while the former

approximates the correct classification probability P(y = y^), the latter is 1 minus that probability. Python offers us such a convenience solution for the accuracy score by providing the built-in method called score() computing exactly this accuracy. Based on it, I would only then have to do a slight computation to come up with the zero one loss.

Splitting data I firstly split the original data into two sets following the proportions: 0.8 and 0.2. There is no universal rule for a specific number. However, the point is to maintain a large number of data for training the predictor map. That is why the former would be used for training and validating the predictors while the latter part would be left untouched and would only be used for testing the predictors.

Given that I do not have any idea at hand which model could perform the best, I should do some comparison in order to have a better understanding in selecting the most suitable model. In the 0.8 part, I further did a k fold with k =5. This would split the data into 5 blocks. For each iteration, 4 blocks would be used to train the map and the left one would be regarded as the validation set. For each model, I came up with 5 classifiers maps with the corresponding values for the training errors and for the validation errors. I then computed their mean values.

I followed the same procedure of computing training errors and validation errors for all models that are referred to in my project.

Results

The results are as follows:

	Logistic regression	Decision tree	k-nn
Mean training error	0.16	0.0	0.11
Mean validation error	0.16	0.13	0.14

On the basis of their performance on the validation sets, I would make my choice for the best one. I could spot straight away that the decision tree yields the smallest mean validation error (0.13< 0.14<0.16). So, the decision tree would be the selected model for this ML project.

One more thing to notice is that in terms of decision tree, the mean training error (0.0) is much smaller than the mean validation error (0.13). This would be discussed in the conclusions.

At this point I had 5 decision tree maps (thanks to k-fold split with k=5). Instead of assessing the performance of each of these maps on the test set and then computing the mean test error, a more typical strategy is to use the whole 0.8 part to train only one best decision tree map. The training error of this map is also 0. Then, I assessed its performance on the test set. The test error was computed to be approximately 0.13.

Conclusions

As noted above, the mean training error of the decision tree map is no way near the mean validation error. This indicates overfitting. In order to go about this, future work could be collecting more data for training. In addition to that, some heuristic approaches could also be helpful to handle overfitting decision trees such as pre pruning and post pruning (Bramer, 2007).

REFERENCES

Bramer, M. (2007). Avoiding overfitting of decision trees. *Principles of data mining*, 119-134.

Jung, A. (2021). Machine Learning. The Basics. mlbook.cs.aalto.fi

https://www.kaggle.com/sakshigoyal7/credit-card-customers.