Assignment
Business Analytics and Data Science
Winter 2020/2021

Scope and business setting

Customers send back a substantial part of the products that they purchase online. Return shipping is expensive for online platforms. For example, return orders are said to reach 50% for certain industries and products. Nevertheless, free return shipping has become a customer expectation and de-facto standard in the fierce online competition on clothing, but shops have indirect ways to influence customer purchase behavior. For purchases where return seems likely, a shop could, for example, restrict payment options or display additional marketing communication.

For this assignment, you are provided with real-world data by an online retailer[1]. Your task is to identify the items that are likely to be returned. When a customer is about to purchase an item, which is likely to be returned, the shop is planning to show a warning message. Pre-tests suggest that the warning message leads approx. 50% of customers to cancel their purchase. In case of a return, the shop calculates with shipping-related costs of 3 EUR plus 10% of the item value in loss of resale value. Your task is to build a targeting model to balance potential sales and return risk to optimize shop revenue. The data you receive and within the test set is artificially balanced (1:1 ratio between (non-)returns). Since the real business situation contains substantially more non-returns than returns, the misclassification costs include a correction factor of 5.

Model assessment

You are expected to provide a binary estimate (0/1) if the customer will return an item within the order. Let's introduce labels for the two classes of interest. We refer to the cases in which an item is returned as the positive class. Accordingly, online purchases in which all items are kept belong to the *negative* class. The performance of your prediction model will be evaluated in terms of expected costs. A model minimizing the expected costs will maximize the revenue of the online shop.

This assessment of prediction models in terms of their financial performance differs from the Kaggle challenge[2] in which we assess models in terms of the AUC. The reason for this difference is that Kaggle does support standard performance metrics like the AUC but does not allow for a custom performance metric such as costs or revenue.

Calculating the financial performance of a model involves handling asymmetric error costs. To see this, let us assume that, for a given case X_i, your model predicts an item to be returned: $\hat{y}_i = 1$. So the model predicts X_i to belong to the positive class. This implies that the shop will display a warning message to discourage the purchase. Two outcomes are possible:

> First, we have a true positive (TP) outcome if the warning discourages a customer from buying an item that she would have returned otherwise. This case does not entail a cost. On the other hand, if a customer receives a warning message and decides not to buy an item, which she would not have returned otherwise, then the shop loses revenue. This case describes a false positive (FP) case.

[1] https://www.kaggle.com/c/bads2021/data
[2] https://www.kaggle.com/c/bads2021/overview/evaluation

Two more outcomes are possible if your model predicts a negative outcome (i.e., that all items of an order are kept).

The true negative (TN) outcome occurs if the customer does keep all items. We again assume zero costs for this case. The fourth and last possible state is the false negative (FN) outcome. The model has predicted the customer to keep all items whereas she actually returns an item. In this case, that is a false negative error, costs for handling the return occur.

Equipped with this understanding and using the above information on costs and benefits, we can set up the following cost-matrix in which v denotes the value of the returned item and we assume asymmetric costs or error. Note that we discuss such asymmetric costs of error in the course Chapter on Imbalanced and Cost-Sensitive Learning[3].

		Actual class	
		Negative	Positive
Predicted class	Negative	0	$0.5 \cdot 5 \cdot (3 + 0.1 \cdot v)$
	Positive	$0.5 \cdot v$	0

Submission files and deliverables

Your submission to the BADS assignment will consist of two files:

1. Text file with your predictions in CSV format.
2. Jupyter notebook, which details your modeling approach.

Both files will be submitted via Moodle. We will update our Moodle page to provide a function for submitting your assignment solution. To make sure that we can process your submissions, please make sure that the naming of your files complies with the below specifications.

You can submit additional files together with the two main assignment files if you wish. For example, you could decide to move helper functions or class definitions into a separate *.py file and submit this file alongside the above main submission files. Using additional *.py files is possible but not required. However, you must submit any supplementary file that is needed to execute your notebook.

Predictions

For us to be able to evaluate your models, please submit your final predictions as a CSV file. Your file name has to display your student id. Assume your student id is 123456. In that case, your prediction should be named **123456.csv**.

The format of your CSV file should be the same as for the Kaggle challenge[4]. Specifically, we require the following format:

order_item_id , your_prediction

[3] See BADS slides chapter 11 on Imbalanced and Cost-Sensitive Learning.
[4] https://www.kaggle.com/c/bads2021/overview/evaluation

1, 1.0

2, 0.0

3, 1.0

...

You can easily produce this format using the ***to_csv()*** function of a Pandas DataFrame object. However, it is as easy to get the format wrong. The first row **must include the column headers** whereby the symbol to separate the columns is the comma. Row 2 and the following rows provide the predictions. The IDs that you include in the first column must match the order_item_id's of the unknown data. Also, you have to produce **one prediction for every order_item_id included in the unknown data**. The **column separator** is the **comma**. The **decimal separator** is the **dot**. We are processing your predictions using a Python script and will not incorporate complex routines for error correction. Your prediction file has to be correct. Otherwise, it will not be processed.

Finally, note **one important difference to the predictions that you submit to Kaggle**. On Kaggle, you submit predictions as a decimal number, which represents your model-estimated **return probability**. Formally, your Kaggle predictions are to represent an estimate $\hat{p}\ (\mathrm{y_i} = 1|\mathrm{X_i})$.

For the BADS assignment, on the other hand, we ask you to submit **discrete class predictions.** Formally, for the assignment, you are asked to submit predictions resulting from the following rule:

$$\hat{y}_i = \begin{cases} 1 & \hat{p}(y_i = 1|X_i) \geq \tau \\ 0 & \text{otherwise} \end{cases}$$

where τ represents the cut-off. We will check if your submitted predictions are either zero or one ($\hat{y}_i \in \{0,1\} \forall i$) using the NumPy function *unique()*. If this test fails, we will assume that you have submitted probabilistic predictions and apply a default cut-off of 0.5. This default cut-off may not be optimal. Consequently, it is in your interest to translate the probability predictions coming from your model into discrete class predictions yourself. This allows you to decide on a suitable cut-off and should improve the performance of your prediction model.

Notebook

The notebook that you submit is the **main source for grading**. We expect your notebook to document your modeling approach in a well-structured manner. To that end, we require your submission notebook to include the following Sections.

- Introduction
- Explanatory data analysis
- Data preparation
- Model tuning and selection
- Model evaluation
- Conclusions

You are encouraged to add subsections and/or add further main chapters where suitable. Also, it is up to you which of the above steps in the predictive modeling process (and thus Sections) you emphasize. However, every submission should touch on the above steps.

Beyond structuring, we recommend that your notebook shows a good balance of text, including tables and figures where suitable, codes, and results, and their discussion. For example, we expect you to briefly motivate relevant modeling decisions. This allows us to judge your competency as a data scientist. Likewise, any result, say a plot or table, deserves a discussion. It may be brief but without discussion, we have no way to follow your reasoning. For example, in the assignment you do not need to show us that you are familiar with standard functions like DataFrame.describe() or the like. We

expect you to know such basics. It is enough and appropriate to only produce those outputs that you are willing to discuss.

The above being said it is clear that codes alone are not enough. At the same time, the very point of a notebook is certainly to show your codes and we will consider them when grading your submission. Make sure to use comments extensively to ensure that we can follow your code easily. Likewise, if you use Python libraries that are not part of a standard Colab or Anaconda environment, please make sure that your notebook includes installation instructions. A typical example would be something like *!pip install name_of_the_package.*

Grading

To assess your submission, we will score every notebook according to a set of pre-defined criteria. The main categories of your scoring table and their weights are:

- Quality of the exposition (25%)
- Data preparation (25%)
- Modeling (25%)
- Code (15%)
- Predictive performance (10%)

We will define sub-categories for each of the above main categories. Scores ranging from zero to four will from will then be awarded for your performance in each sub-category. A score of zeros indicates that you did not address a topic at all while a score of 4 equates to very good performance. Adding up the scores over the sub-category provides an overall score for the corresponding main category. On this basis, we will calculate your total score as the weighted sum over the five scores for each of the main categories. Last, final grades will be defined based on the quantiles of the score distribution.

We will not reveal the sub-categories. Doing so would provide a concrete recipe for how to approach the modeling task. This is not intended. Rather, we want to see how you approach the task based on what you have learned in BADS. As to modeling and data preparation, the course materials should equip you with a comprehensive understanding of what can be done and will be awarded score points.

As to the quality of the exposition, we admit that academic writing is not a skill that we have practiced in BADS. However, it is a skill that every university student requires. We will consider aspects such as whether your exposition is readable and well-structured. Beyond such soft-skills the degree to which your exposition displays knowledge of data science, as well as the business problem that you aim to solve, is crucial. Also, we obviously expect your notebook to comply with academic standards. Credit any external source that you use. This includes but is not limited to the use of literature or blog posts. However, you do not have to cite any of the course materials. Likewise, it is clear that you will use the documentation of Python libraries and platforms like *StackOverflow*. You do not have to cite any of these.

Your score in the last category, predictive performance, will be determined based on your performance rank on the unknown data. More specifically, we will use the true labels of the cases in the unknown data and compute the expected costs of your predictions based on the above cost-matrix. Score points will then be awarded based on the quantiles of the cost-distribution. For example, we envision awarding a score of 4 for submissions in the top-5 or top-10 percent of the cost distribution. Hence, the principle is the same as in the Kaggle competition but the performance metric is different. Kaggle ranks predictions in terms of the AUC while the final assignment considers expected costs.

Finally, we will take the liberty to award various bonus points for any nice feature of your notebook that our grading approach, as just outlined, does not cover. Clever ideas, solutions that go beyond course content, and using other advanced data science methods will be recognized and credited. The same goes for the skillful use of scholarly literature.