Mini project: Leveraging Machine Learning for Revenue Optimization via Strategic Couponing

Introduction

In the realm of e-commerce, a common observation is that a significant proportion of customers engage in a single transaction and then cease further purchases. This phenomenon can be attributed to a multitude of factors. To mitigate this, e-commerce platforms employ a variety of strategies aimed at fostering customer loyalty. One such strategy involves the distribution of discount vouchers subsequent to the initial purchase, with the goal of incentivizing repeat transactions. However, indiscriminate distribution of these vouchers may not be an optimal strategy. This is because a segment of customers might have engaged in repeat purchases even in the absence of such incentives. Consequently, the redemption of these vouchers by such customers translates into a reduction in the retailer's profit. Empirical analyses conducted by the media retailer have demonstrated that for 10% of non-buyers, the voucher instigates a purchase with an average order value of €20. Thus, if a voucher is dispatched to a customer who would not have actually made another purchase, the revenue increases by an average of €1.5. On the other hand, sending a voucher to a customer who would have made a purchase anyway results in a revenue loss equivalent to the voucher value of €5. For customers who don't receive a voucher, there is no impact on revenues. Therefore, it is crucial to devise a more targeted approach for the distribution of these vouchers.

Task

The task at hand involves constructing a predictive model that leverages various features associated with a customer's initial order. The objective is to determine whether a €5.00 voucher should be issued to a specific customer. Detailed descriptions of these features can be found in the `data_dictionary.pdf` file.

The model should be designed to predict if a customer will place a subsequent order within a 90-day period following their initial purchase. This information is represented by the `target90` variable in the dataset.

The model's performance is evaluated based on the expected revenue across all customers in a given dataset. This is computed by considering the model's predictions in conjunction with the associated costs and revenues. It's crucial to note that the model's effectiveness is directly tied to its ability to maximize this expected revenue. Hence, the model should be optimized with this specific goal in mind.

The dataset is provided in the `train.csv` file.

Submission and evaluation

Your grade is composed as follows:

- **Jupyter notebook (50%):** Please create a Jupyter notebook that documents all steps you have done. Use markdown cells to guide the reader through your notebook and explain your code. Use comments where needed. Use plots to present your findings.

  Specifically, these are the steps we want to see:

  - You have familiarized yourself with the data, for example, through exploratory data analysis.

  - You have preprocessed the data in a meaningful manner, e.g. by creating new features, dealing with missing values, feature selection and feature encoding.
  - You have trained (at least) one model and tuned its hyperparameters.
  - You have evaluated the performance of your model.
  - You have tried to understand and interpret your model by applying the methods you learned in the lecture. Use global as well as local methods.

- **Model performance (20%):** We compare the expected revenue from your model with the reference revenue that would result if simply all customers were sent a voucher. To do so, we use a holdout test dataset. An example of such a dataset is given to you in the file `test_example.csv`. You can assume that the real test set has the same data structure.

  For this evaluation, please create a prediction function that can be easily applied to further customer instances, e.g. those from a holdout-test-set. This function should be embedded in a python file. Use the python script `predict.py` provided in ILIAS and make the necessary adjustments in the `predict()` function (positions where to insert code are marked). Please don't forget to include every step that is needed to get from the raw data to the final prediction, i.e. also include your data preprocessing.

  You can test your script using the `test_example.csv` by running the following command in a terminal: `python predict.py -i data/test_example.csv -o predictions.csv` (you may need to adjust the filepaths, check out `python predict.py --help` for an explanation of the arguments). You can expect your function to be properly coded if this runs without any errors and returns the expected results. You can check the results in the output file (in this case `predictions.csv`)

  IMPORTANT: Make sure to submit everything that is loaded from file within your function, e.g. fitted transformers or the fitted model. Otherwise, we are not able to run your script on our machines.

- **Presentation (30%):** Imagine you want to present your model to the marketing executive of the media retailer. The marketing executive doesn't have much time, so you must limit yourself to 10 minutes. In these 10 minutes, you should explain as clearly as possible why (or not) the model should be used by the company. The marketing executive is particularly interested in the variables according to which the model makes decisions and what's the model's benefit in monetary terms. This 10-minutes summary presentation will be followed by 10-minutes Q&A where we ask you questions about your submission.

In case of any questions, please don't hesitate to reach out via the ILIAS forum.

We highly encourage you to work in pairs, however, you can also work alone. Upload your solution as a zipped archive to ILIAS by 23:00, January 28, 2024. This should include the jupyter notebook (run all cells and don't clear the output), your presentation slides as PDF file and the prediction script (`predict.py`) as well as ALL files that are needed to run this script (see notes on *model performance* above). The presentations will take place on February 1, 2024 during the usual lecture time.

Make sure you comply with the Examination and Assessment Honor Code outlined below.

Examination and Assessment Honor Code

All members of the School of Business and Economics at the University of Tübingen (faculty, students, and alumni) share a commitment to honesty and integrity. In particular, all members follow the standards of scholarship and professionalism in all examinations and assessments.

By submitting this assignment, students agree to comply with this Examination and Assessment Honor Code.

Students who violate this Honor Code are in breach of this agreement and are subject to sanctions imposed by the School of Business and Economics, the University and its responsible bodies (e.g., the board of examiners ("Prüfungsausschuss")).

1. All members of the School of Business and Economics at the University of Tübingen (faculty, students, and alumni) have the obligation to report known violations to the responsible bodies (e.g., the board of examiners ("Prüfungsausschuss") or the Dean of Programs).

2. You must not represent another's work as your own.
3. You must not receive inadmissible assistance of any sort before, during, or after an examination or other forms of course work that is subject to assessment by faculty members.
4. You must not provide inadmissible assistance of any sort before, during, or after an examination or other forms of course work that is subject to assessment by faculty members
5. Violations of this Honor Code will be handled according to the rules and regulations laid out in the rules for this course.