Analytics Cup 2021/2022

How good is the Offer? - Developing a Classification Model for Predicting the Success of Sales Offers for Smart Infrastructure

The Challenge

This year's Analytics Cup takes place in cooperation with *Siemens Advanta Consulting*. One of their clients, a large international company headquartered in Germany (in the following: "*The Company"*), sells various Smart Infrastructure products and services to business partners ("*customers*"). These products range from fire safety products to surveillance cameras to network-enabled air conditioners. In the sales process, the Company will approach a potential customer with a specific *offer* for a (packet of) product(s) and related services (e.g. installation, maintenance for a fixed period of time). When creating these individual offers, determining the right price is a challenging and crucial task for the Company: a reasonable choice will generate a good profit margin for the Company and, simultaneously, satisfy the needs of the customer, thus achieving a good chance of being accepted. If the price is set too high, however, the customers will likely reject the offer. Besides the price, many other factors may influence the decision of whether an offer is accepted or not: type of industry, length of a business relationship, willingness to pay in the past, etc.

In this challenge, you and your team have the task to develop a predictive model that classifies offers into successful and unsuccessful ones using multiple real-world data sets provided by the Company and *Siemens Advanta Consulting*. You will find offers for existing customers at facilities where the Company has already deployed other Smart Infrastructure products, but also offers to entirely new facilities without a previously existing business relationship. The products comprise the categories *fire*, *security*, and *comfort*. Your model will lay the foundation for developing decision-support tools for the Company's sales departments.

You are provided with three data sets in this challenge. The first data set is the "customers.csv" file and contains 8452 records about the Company's (potential) customers located in France and Switzerland. The column *CUSTOMER* consists of IDs, which uniquely identify customers within a country. However, two different customers may have the same ID if they belong to different countries.

The "transactions.csv" contains the 26151 offers that the Company has made to its (potential) customers. There is a distinction between *main offers* and *sub offers*: A main offer is uniquely identified by its primary key in the column *MO_ID.* Each main offer can have zero, one, or multiple associated *sub offers*, which are uniquely identified by their key in column *SO_ID.* You will need to make predictions on the *sub* offer level of granularity. If a *main offer* has no *sub offers* attached, you should treat the main offer itself as if it were a sub offer.

The "geo.csv" file contains further information about the Company's sales offices. You may assume that all the offers made by a particular sales office will be to a customer within the same country as the sales office itself.

YOUR TASK: Use these data sets to develop a model that can predict the outcome of the column *OFFER_STATUS*, which indicates whether a customer accepts an offer (your model predicts *OFFER_STATUS*=1) or rejects it (your model predicts *OFFER_STATUS*=0).

For those transactions that have a TEST_SET_ID and for which you are not given the OFFER_STATUS (the "*private test set*"), you must make predictions that will form your submission. See the submission_template.csv for details about the required format. You project will be evaluated and graded based on these predictions.

Note that in some places, we (the Business Analytics team) have artificially degraded the data quality and made some other modifications to the data in order to tune difficulty of the Analytics Cup. Any data quality issues you notice should therefore not be attributed to Siemens Advanta Consulting or the Company.

Evaluation

Your predictions will be evaluated based on the performance measure of *balanced accuracy* – the arithmetic mean of Sensitivity and Specificity – that your submission achieves on the private test set.

		Truth (an offer was accepted)	
Your prediction		YES	NO
	YES	True Positive	False Positive
	NO	False Negative	True Negative
		Sensitivity = True Positive Rate = $\frac{TP}{TP+FN}$	Specificity = True Negative Rate = $\frac{TN}{FP+TN}$
		Balanced Accuracy = BAC = $\frac{Sensitivity+Specificity}{2}$	

The Data

customers.csv

Column	Description
CUSTOMER	(integer) A unique identifier for the customer for each country
REV_CURRENT_YEAR REV_CURRENT_YEAR.1 REV_CURRENT_YEAR.2	(numeric) The revenue of the corresponding organization for the current year, and for previous two years (given in CURRENCY)
CREATION_YEAR	(Date) The year of the first customer contact
OWNERSHIP	(String) The form of organization of the customer
COUNTRY	(String) The country a customer operates in
CURRENCY	(String) The name of the currency of REV_CURRENT_YEAR.1

transactions.csv

Column	Description
MO_ID SO_ID	(String, primary keys) *main offer* id and *sub offer* ids. Each main offer may have none, one, or multiple suboffers.
CUSTOMER	(Int) A unique identifier for the customer for each country. Foreign key for customers.csv.
END_CUSTOMER	(String) A unique identifier for the end customer that the product(s) will be delivered to (This may be different from the "CUSTOMER" who is the Company's counterpart when making the offer.)
OFFER_PRICE	(Double) The final price of the offer in EUR
SERVICE_LIST_PRICE	(Double) The list price for the offered service in EUR
MATERIAL_COST	(Double) The costs to the Company of the materials of an offer in EUR
SERVICE_COST	(Double) The costs of the service of an offer in EUR
PRICE_LIST	(String) The reference price list for the transaction
ISIC	(Int) International Standard Industrial Classification for the identification of the branch of industry a customer is working in.
MO_CREATED_DATE SO_CREATED_DATE	(String) The creation date of the corresponding MO_ID and SO_ID

TECH	(String) Technology group of the offer
	BP = Building Products, C = Comfort
	E = Entertainment, F = Fire, FP = Fire Protection, S = Security

OFFER_TYPE	(String) The type of the offer. *(no further information about the meanings of types can be provided.)*
BUSINESS_TYPE	(String) The type of business relationship at the time of the offer.
COSTS_PRODUCT_A COSTS_PRODUCT_B COSTS_PRODUCT_C COSTS_PRODUCT_D COSTS_PRODUCT_E	(Double) A few specific products (A-E) of Company are known to have extremely high or low margins. When these products are part of the offer, the offer related cost information of these products is given in these columns.
SALES_LOCATION	(String) The location where this offer was made
OFFER_STATUS	(String) Indicates whether a customer accepted an offer or rejected it.
TEST_SET_ID	(int) An integer ID if this offer is part of the test set for which you need to submit predictions. NA otherwise.

geo.csv

Column	Description
COUNTRY	(String) The Country in which the sales office is located
SALES_OFFICE	(String) The name of the Sales Office
SALES_BRANCH	(String) The branch which the sales office belongs to. (each branch may have multiple offices.)
SALES_LOCATION	(String) The location of the sales office

You are not allowed to share the data with any person outside of this challenge, and you are only allowed to use it for this challenge. Furthermore, you need to delete all data files once the cup is finished.

Submission Rules

Submissions

A valid submission contains of a csv-file containing predictions and a script that generates these predictions from the data that you have been given. Your submitted script **must be self-contained** and reproducible, more on that below. Your prediction file will be graded automatically and judged based on the performance measure of **balanced accuracy** it achieves on the test set. Your team can make **up to 10** valid submissions. Note that invalid submissions do not count to this limit. Only the **best valid submission** from your team will be evaluated for grading.

We have provided you with a sample submission file (with entirely random predictions) which you can use to check whether the format of your generated submission is correct.

Make sure that all submissions adhere to the following **naming scheme**. This ensures that you keep track of your files as a team, which is especially important if we invite you to a clarification meeting.

Prediction-File:	*predictions_group_name_number.csv*
Script:	*script_group_name_number.R*

Prohibitions

The following things are strictly prohibited and will result in disqualification:

- You may **NOT** hard-code predictions for any instances in the test set. All predictions must be based on your model output.
  This applies both to individual predictions (i.e. **forbidden**: prediction[test_set_id==201] <- 1) as well as to fixed rules (**forbidden**: prediction[CUSTOMER==5] <- 0).
  Note: Hard-coding **features** to be used in the model is generally allowed.
- You may **NOT** work together with other teams. If we find that you copied work or cooperated, both teams will be disqualified.

If you are unsure about whether something is allowed or not, please reach out to us or ask in the moodle forum! In cases of ambiguities, we reserve final judgment on whether a given submission violates the rules above!

Reproducibility

All submissions must be **reproducible**, i.e. the submitted R script must reproduce the same prediction file, even when run on a different machine at a different time. To ensure this, your scripts should (at least) follow the following guidelines:

- Import all packages that you use **at the very top** of the file. If you implicitly use a backend package via tidymodels/parsnips (`set_engine`) (or via an mlr-learner, etc), please explicitly import the library anyway, or, at a minimum, add a comment to the top of your file.
- At the top of your script, right after the imports, set `**set.seed(2022)**` to seed R’s random number generator (rerunning the script will then give you the same results in random operations). Some machine learning packages (such as h2o) manage their own random

number generator that's not managed by R. If you use such packages, set the seed in the same manner.

- Do NOT change the file names of the training and test data sets. Your script should `read` the files (and write submissions) from/to **its own directory**. (i.e. `read_csv('customers.csv')`, rather than `read_csv('C:/Users/name/my_files/more_directories/I_renamed_the_customer_file.csv)'
- Do NOT modify the content of the data files provided. All data preparation should happen WITHIN the provided script.
  You may want to save intermediate results that took a long time to generate (data, models, etc.) to disk and read them again. That is fine for prototyping, but not for the final script you submit.

The following last point will not be handled as strictly but you should nevertheless adhere to it:

- Your submitted script should be a (reasonably) minimal implementation to generate your model. We don't expect you to spend any time on optimizing this, but please use good judgment to avoid unnecessary computation in evaluation.
  Example 1: *To find your perfect model, you performed a hyper-parameter search that took 3 days to run. Your submitted script should then only train your final model using the (hard-coded) final hyperparameters that you found. Don't include the search in your file. In such a case, add a short comment about how you arrived at the hyperparameters (or comment out the code for the search)*
  Example 2: *You trained 20 models and decided on your favorite one to create a submission at the end. Your submitted script should only trigger training of your favorite model, not all 20 models. (Delete or comment out the code for the other models in your submission.)*
- Although good solutions should be possible in <<10 min runtime on modest hardware (e.g. 5-year old laptops), some groups might have models that take longer to train. If your script takes a very long time to run, please include a comment at the top of your script that includes approximate runtime and info about your computer. (e.g. `#1.5 hours on dual-core laptop with 4GB RAM`).

If your submitted solution is not reproducible, we reserve the right to disqualify your team from the Analytics Cup.

Frequently Asked Questions

Additional External Data

You might want to consider including external data in your analysis that is not part of the data set but might be valuable for predicting. This is generally allowed, but you must ensure that your submission stays reproducible. Thus, load the data directly via the URL in your R script or include it with the *dput* command. Do not store any additional data states on your disk.

Languages other than R

Some students have asked whether they may use other languages than R (such as python) for the Analytics Cup. This is permitted in general, but we cannot provide you with any support and you must adhere to the same standards of reproducibility found above.
If you want to use python, you must submit a single, self-contained python script. For grading of python scripts, all packages you use must be installable via conda or pip.
If you want to use any language other than R or python, please contact us beforehand.

Jupyter and Rmd-Notebooks

Some students prefer writing code in Rmd or Jupyter notebooks rather than flat .R scripts.
Submissions uploaded as notebooks are generally accepted, but must adhere to the same Reproducibility requirements outlined above. Especially, the cells in your notebook must run **in order**, **from top to bottom**, and should not contain additional exploratory analysis steps, in particular none that take a long time to run. For submissions uploaded as .ipynb files, additionally you **must** strip all cell output from the file before submitting.

Cloud Services, Google Colab, etc.
The challenge is designed to run fairly comfortably on modest hardware (e.g. 5 year old laptops with ~4GB RAM and dual-core processors). If you want to use cloud platforms to build your models, you may do so, but you will nevertheless have to submit self-contained scripts that can run on our local machine. (You may **not** submit a link to a repository, etc., instead.)