Analytics Cup 2022/2023

Fraud Detection - Developing a Classification Model for Prediction of Unauthorized Reselling Activities

The Challenge

This year's Analytics Cup takes place in cooperation with *Siemens Advanta Consulting.* The business model of one of their clients, a large international company headquartered in Germany (in the following: "the Company"), is to produce parts and machinery that are used to enhance the industrial processes of their customers. While some of those customers have distribution agreements with the Company, the majority of customers agreed in contracts to integrate the products into their production processes. The Company wants to prevent that some of the latter customers still distribute the products and thus act as unauthorized resellers. These activities not only lead to economic losses but also poses legal threats and potential brand damages to the Company. Hence, it is necessary to reveal those activities as soon as possible, which is, however, not a trivial task. Currently, the detection process relies on fixed rules based on the results of simple statistical evaluation and regular customer visits.

Since these manual inspections are cost- and time-intensive, you and your team obtained the task to develop a predictive model that detects whether one of the customers acts as a reseller. To achieve this, you will utilize the learned concepts of the lecture and multiple real-world data sets provided and anonymized by the Company and *Siemens Advanta Consulting*. Your model will lay the foundation for developing decision-support tools for the Company, and thus, a central element in the development of this model is to ensure the explainability of the predictions generated by your model. Decision makers need to understand which factors affect your classification to develop mechanisms that prevent unauthorized reselling activities in the future.

You are provided with seven data sets in this challenge. The first data set is the *classification.csv* file, which contains information on whether a customer acts as a reseller or not. The sales activities of the customers are stored in the *customers.csv* file, and the information about the corresponding sales of the line items is in the *sales-orders.csv*. Note that some customers might share a sales order. In this case, you cannot join both tables on the *Sales_Order* but need also to match the *Item_Position*. In some cases, there does not exist a matching *Item_Position* in either of both tables. Set those entries to zero and re-match the tables. Each sales order consists of one or multiple line items, and the information about the entire sales order is stored in the *sales-orders-header.csv* table. Additionally, a sales order can consist of a service. In this scenario, there exists a matching partner of the *Material_Class* value in the *service-maps.csv* file. Otherwise, the sales order does not correspond to a service. Finally, the *business-units.csv* contains information about the business units dealing with a sales order.

YOUR TASK: Use these data sets to develop a model that can predict the outcome of column *Reseller* in table *classification.csv*, which indicates whether a customer acts as a reseller (Reseller=1) or utilizes the products in their internal processes (Reseller=0). For those transactions that have a *test_set_id* and for which you are not given the Reseller entry (the *"private test set"*), you must make predictions that will form your submission. See the *submission_template.csv* for details about the required format.

Furthermore, the Company expects you to not only develop a black-box model but that you can explain the predictions accordingly. Thus, you should also be prepared to give a short presentation where you explain which factors affect your prediction. We will randomly select top teams to present their approach internally. Hence, you will *not* upload the presentation to the platform.

Your project will be evaluated and graded based on the predictions and your ability to explain those accordingly.

Note that in some places, we (the Business Analytics team) have artificially degraded the data quality and made some other modifications to the data to tune the difficulty of the challenge. 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

classification.csv: The file describes which customer acts as a reseller.

Column	Description
Customer_ID	(Primary Key, String) A unique identifier for a customer.
Reseller	(Integer) Label indicating whether a customer acts as a reseller.
Test_set_id	(Integer) Id in the test data set.

customers.csv: The file provides the information about the customers and their sales orders.

Column	Description
Sales_Order	(Foreign Key, String) A unique identifier for a sales order.
Item_Position	(Foreign Key, String) A unique item position within a sales order.
Type	(String) The type of the sales order: SOP – Shipped to Party STP – Sold to Party
Customer_ID	(Foreign Key, String) A unique identifier for a customer.

sales_orders_header.csv: The file contains information about the sales orders.

Column	Description
Sales_Organization	(String) The identifier of the internal organization that manages the sales order.
Sales_Order	(Foreign Key, String) A unique identifier for a sales order.
Creation_Date	(Date) Date of creation.
Creator	(String) Id of the employer who created the line item.
Document_Type	(String) Type of sales order.
Release_Date	(Date) Date when the sales document was released from the company.
Delivery	(String) Status of the delivery.
Net_Value	(Double) Value of the entire sales order.

sales-order.csv: This file provides information about the line items within a sales orders.

Column	Description
Sales_Order	(Primary Key, String) A unique identifier for a sales order.
Item_Position	(Primary Key, String) A unique item position within a sales order.
Num_Items	(Integer) The number of sold units.
Material_Code	(String) A code identifying the material of the item.
Material_Class	(Foreign Key, Integer) An identifier for the material class of the item.
Cost_Center	(String) The cost center that manages the sales order.
Net_Value	(Double) The value of the line item.

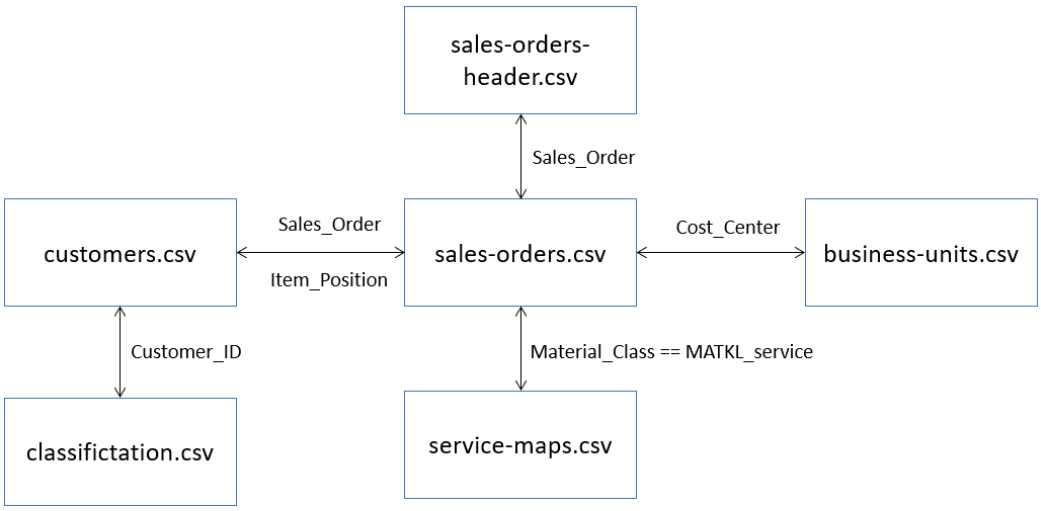
business_units.csv: This file provides the information about the business units dealing with the sales orders.

Column	Description
YHKOKRS	(Double) The identifier of the business unit.
Cost_Center	(Primary Key, String) The identifier of a cost center.
Business_Unit	(String) The name of the corresponding business unit.

service_map.csv: This file contains the unique identifiers of all services.

Column	Description
MATKL_service	(Double) A identifier for a service.

The following scheme provides an overview on how the tables relate to each other.



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** use Automatic Machine Learning packages. The goal of the challenge is that you can improve your Data Science skill, and not that the team with the highest computational power wins.
- 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.)

AC Office Hours

To support you in participating in the challenge, we will offer you 1-on-1 support sessions, where you can discuss several individual questions regarding the challenge with a TA or tutor. We will provide multiple slots, where each team can book **one** slot via Moodle. We plan to conduct those sessions after 2/3 of the processing period is completed. The actual dates will be announced via Moodle.

Peer Reviewing

Recent investigations of *peer reviewing* revealed that this concept enhances the learning effect of students. Thereby, participants do not only observe potential alternative approaches but also actively think about the improvement of those. To leverage these effects, we would like to provide you with an **optional** peer review loop. You will have the possibility to exchange your solutions with another randomly selected team and discuss your approaches after the challenge has ended. We will announce and describe the process via Moodle.