Dynamic Deal Scoring Technical Documentation

Zagreb, 18th of May, 2021

1 Source code directory structure

There is main directory called `source_code` and seven subdirectories: `markdowns`, `r-util`, `model`, `dataset`, `rest_app`, `notebooks`, and `plots`:

1. `markdowns` - contains R markdown notebooks with exploratory data analysis
2. `r-util` - contains R script with helper functions
3. `model` - contains model related files in Python
4. `dataset` - contains data set with name `LUMEN_DS.csv`
5. `rest_app` - contains Flask app in Python
6. `plots` - contains plots in pdf format
7. `notebooks` - contains Jupyter Notebook file

The exact `source_code` directory structure is following:

```
source_code
├──markdowns
│  ├──Dynamic_Deal_Scoring_EDA.Rmd
│  ├──eda_part_1.Rmd
│  ├──eda_part_2.Rmd
│  ├──eda_part_3.Rmd
│  └──eda_part_4.Rmd
├──notebooks
│  └──Data_Preparation.ipynb
├──r-util
│  └──helper_functions.R
├──model
│  └──various Python files
├──dataset
│  └──LUMEN_DS.csv
├──rest_app
│  └──app
│     ├──various files
│     └──main.py
├──plots
│  └──plots from the exploratory data analysis
└──requirements.txt
```

IMPORTANT: data set `LUMEN_DS.csv` needs to be in directory `dataset`.

2 R

Markdowns `Dynamic_Deal_Scoring_EDA.Rmd`, `eda_part_1.Rmd`, `eda_part_2.Rmd`, `eda_part_3.Rmd`, and `eda_part_4.Rmd` are used for generating plots for documentation and for conducting exploratory data analysis. `helper_functions.R` is used as a help script for `eda_part_*.Rmd` files. R version required to run all R notebooks and scripts is 4.0.5

Best way to run R markdowns is by using RStudio[1]. It is advised to open a new project, load markdowns from the project and set working directory to `source_code/markdowns`. All the chunks of code can be reproduced by running them in the markdown using *CTRL + Shift + Enter* key combination on every chunk individually. It is important to note that the chunks need to be executed in the provided order. Another option is to run all the chunks at once from first to last from RStudio menu.

2.1 Libraries

Libraries used for `Dynamic_Deal_Scoring_EDA.Rmd`: *tidyverse, GGally, lubridate, dplyr, gridExtra, Cairo, grid, reshape2, RColorBrewer, kableExtra, scales, grid, schoolmath*, and *ggridges*.

Libraries used for eda_part_*.Rmd: *tidyverse, lubridate, tidytext, scales, chron, ggrepel* and *ggridges*.

To install a missing library use the following command inside RStudio console window:

```
install.packages("library_name")
```

3 Jupyter Notebook

Notebook *Data_Preparation.ipynb* is used for cleaning the dataset. Python version used in the implementation is 3.9.2. It is also required to have Jupyter[2] software installed and of course the specified Python version.

Notebook can be started from terminal. To start Jupyter Notebook from terminal, position yourself where the notebook is (`source_code/notebooks`) and type `jupyter notebook`. Localhost will be started in your default browser. Then go to browser and open *Data_Preparation.ipynb* file in browser. Now all the chunks of code's outputs can be reproduced by running the chunks of code in the notebook using *CTRL + Enter* key combination or in the menu bar *Cell –> Run All*. It is important to note that the chunks need to be executed in the provided order.

4 Python

To run Python related files it is mandatory to have Python 3 installed (preferably version 3.9.2[3]) and pip[4] package installer.

It is recommended to create virtual environment and run Python scripts inside virtualenv [5]:

1. Position yourself inside `source_code` directory using terminal (Linux) or cmd (Windows)
2. Run `sudo apt-get install python3.9-venv` (Linux) or `pip install virtualenv` (Windows) to install virtual environment
3. Create virtual environment: `python3.9 -m venv venv_name`
4. Activate virtual environment: `source venv_name/bin/activate` (Linux) or `venv_name\Scripts\activate` (Windows)
5. After activating environment run the following: `pip install -r requirements.txt` to install libraries

[1] https://www.rstudio.com/
[2] https://jupyter.org/
[3] https://www.python.org/downloads/
[4] https://pip.pypa.io/en/stable/installing/
[5] https://docs.python.org/3/library/venv.html

4.1 Libraries

All required libraries to run Python code are specified in `source_code/requirements.txt` file.

4.2 Running model

To obtain reported evaluation results of our baseline and CHAID models one needs to run two separate files:

1. `source_code/model/baseline_evaluation.py`
2. `source_code/model/chaid_evaluation.py`

Instructions are following:

1. Start virtual environment and libraries from `requirements.txt` file (already explained)
2. Position yourself inside `source_code/models`
3. For baseline run from command line: `python baseline_evaluation.py`
4. For CHAID model run from command line: `python chaid_evaluation.py`

To try the model and get generated tree plots, you can run two separate files for baseline and CHAID:

1. `source_code/model/baseline_test.py`
2. `source_code/model/chaid_test.py`

Instructions are following:

1. Start virtual environment and libraries from `requirements.txt` file (already explained)
2. Position yourself inside `source_code/models`
3. For baseline run from command line: `python baseline_test.py`
4. For CHAID model run from command line: `python chaid_test.py`

To generate decision tree plots, two packages are mandatory: `graphviz`[6] and `orca`[7].

4.3 Running Flask application

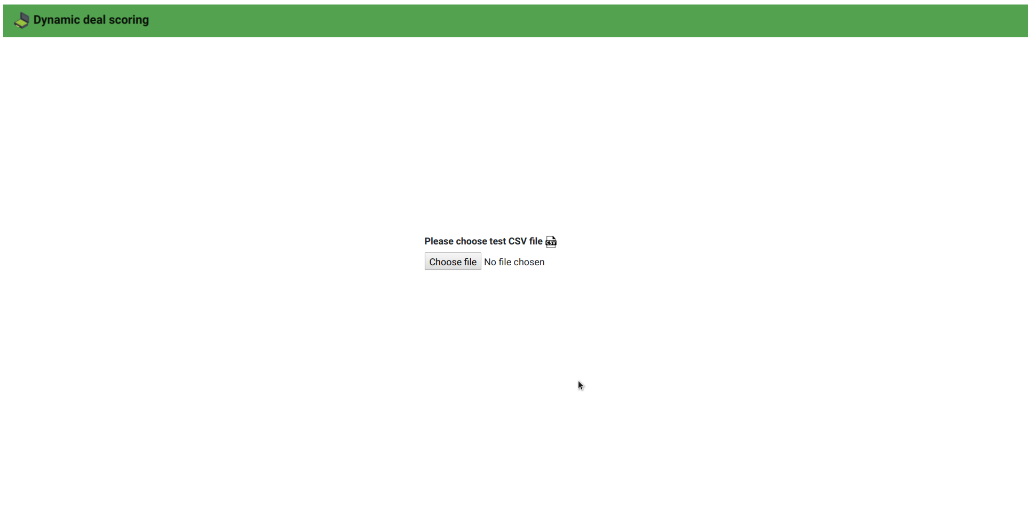
To show how our model predicts, we developed REST application in Flask. To run the Flask application, start virtual environment, install libraries from `requirements.txt` file and then do the following:

1. Position yourself into `source_code/rest_app/app` using command line
2. Run from command line: `python main.py`
3. Now go to web browser on `http://127.0.0.1:5000/` - you will see the following display from Figure 1
4. Next load .csv file by clicking on button *Choose file* and click |open| when you locate the data frame you want to load (Figure 2)
5. Figure 3 shows how the application looks after the successful load of the .csv file into application
6. Figure 4 shows how you can choose the row from the data frame to make a prediction
7. Figure 5 shows successful model prediction after clicking on the sound row of the loaded data frame, while Figure 6 shows prediction failure when clicked on a row for which prediction is not possible

IMPORTANT: It is not necessary to send POST request through the Flask application frontend. It is also possible to send POST request with payload to the endpoint `/scoring`.

[6] https://graphviz.org/download/

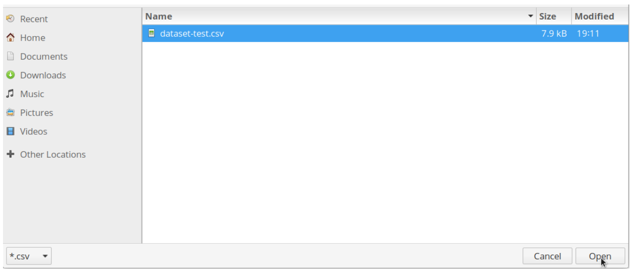
[7] https://github.com/plotly/orca#installation



Dynamic deal scoring

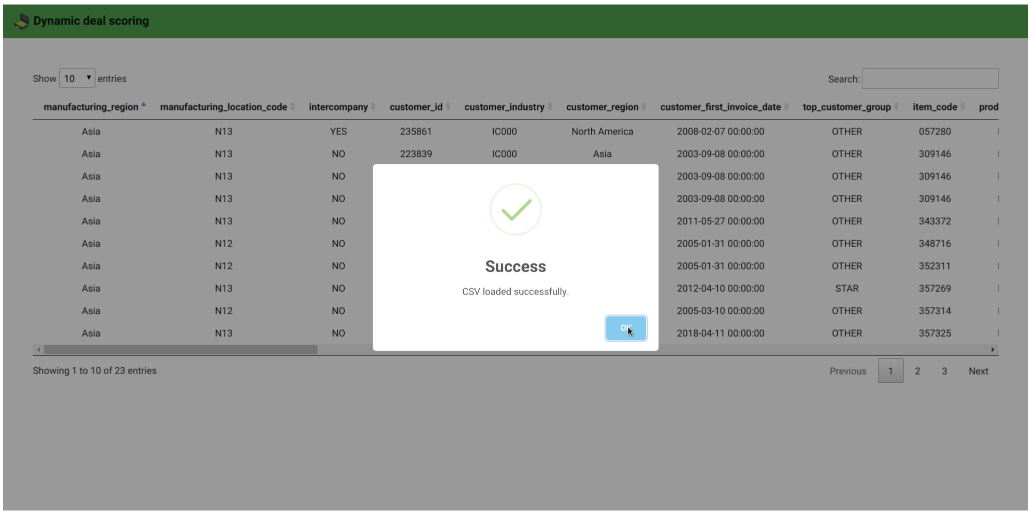
Please choose test CSV file

Figure 1: Flask application homepage.



Name	Size	Modified
dataset-test.csv	7.9 kB	19:11

Figure 2: Choosing .csv file on button click.



Dynamic deal scoring

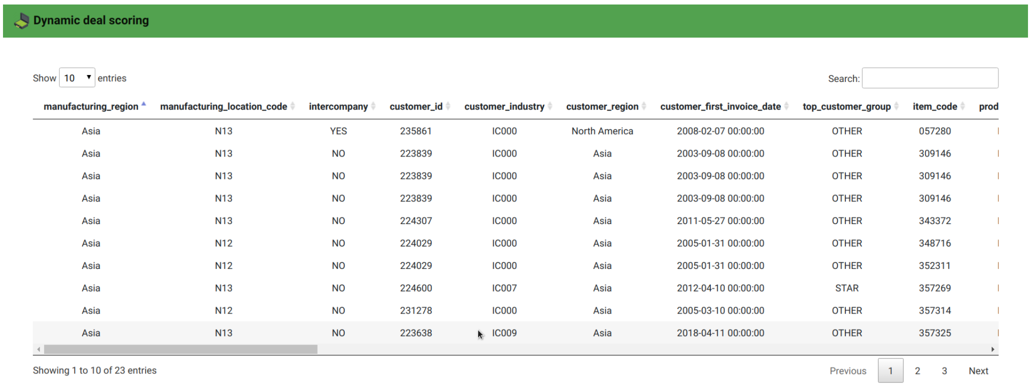
manufacturing_region	manufacturing_location_code	intercompany	customer_id	customer_industry	customer_region	customer_first_invoice_date	top_customer_group	item_code	prod
Asia	N13	YES	235861	IC000	North America	2008-02-07 00:00:00	OTHER	057280	
Asia	N13	NO	223839	IC000	Asia	2003-09-08 00:00:00	OTHER	309146	
Asia	N13	NO				2003-09-08 00:00:00	OTHER	309146	
Asia	N13	NO				2003-09-08 00:00:00	OTHER	309146	
Asia	N13	NO				2011-05-27 00:00:00	OTHER	343372	
Asia	N12	NO				2005-01-31 00:00:00	OTHER	348716	
Asia	N12	NO				2005-01-31 00:00:00	OTHER	352311	
Asia	N13	NO				2012-04-10 00:00:00	STAR	357269	
Asia	N12	NO				2005-03-10 00:00:00	OTHER	357314	
Asia	N13	NO				2018-04-11 00:00:00	OTHER	357325	

Success

CSV loaded successfully.

Showing 1 to 10 of 23 entries

Figure 3: Successfully loaded data frame display.



Dynamic deal scoring

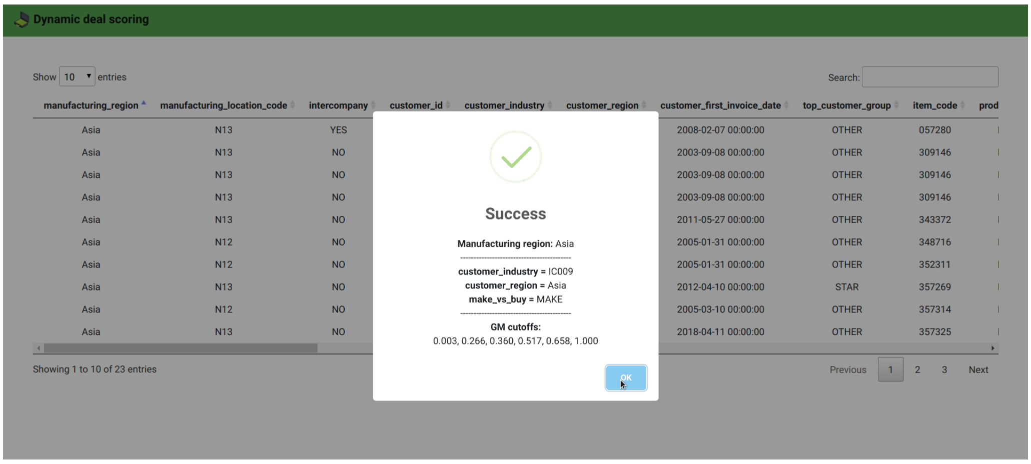
Show 10 entries

Search:

manufacturing_region	manufacturing_location_code	intercompany	customer_id	customer_industry	customer_region	customer_first_invoice_date	top_customer_group	item_code	prod
Asia	N13	YES	235861	IC000	North America	2008-02-07 00:00:00	OTHER	057280	
Asia	N13	NO	223839	IC000	Asia	2003-09-08 00:00:00	OTHER	309146	
Asia	N13	NO	223839	IC000	Asia	2003-09-08 00:00:00	OTHER	309146	
Asia	N13	NO	223839	IC000	Asia	2003-09-08 00:00:00	OTHER	309146	
Asia	N13	NO	224307	IC000	Asia	2011-05-27 00:00:00	OTHER	343372	
Asia	N12	NO	224029	IC000	Asia	2005-01-31 00:00:00	OTHER	348716	
Asia	N12	NO	224029	IC000	Asia	2005-01-31 00:00:00	OTHER	352311	
Asia	N13	NO	224600	IC007	Asia	2012-04-10 00:00:00	STAR	357269	
Asia	N12	NO	231278	IC000	Asia	2005-03-10 00:00:00	OTHER	357314	
Asia	N13	NO	223638	IC009	Asia	2018-04-11 00:00:00	OTHER	357325	

Showing 1 to 10 of 23 entries

Figure 4: By clicking on the row of the data frame, we can get prediction of the model for that row.



Dynamic deal scoring

Show 10 entries

Search:

manufacturing_region	manufacturing_location_code	intercompany	customer_id	customer_industry	customer_region	customer_first_invoice_date	top_customer_group	item_code	prod
Asia	N13	YES				2008-02-07 00:00:00	OTHER	057280	
Asia	N13	NO				2003-09-08 00:00:00	OTHER	309146	
Asia	N13	NO				2003-09-08 00:00:00	OTHER	309146	
Asia	N13	NO				2003-09-08 00:00:00	OTHER	309146	
Asia	N13	NO				2011-05-27 00:00:00	OTHER	343372	
Asia	N12	NO				2005-01-31 00:00:00	OTHER	348716	
Asia	N12	NO				2005-01-31 00:00:00	OTHER	352311	
Asia	N13	NO				2012-04-10 00:00:00	STAR	357269	
Asia	N12	NO				2005-03-10 00:00:00	OTHER	357314	
Asia	N13	NO				2018-04-11 00:00:00	OTHER	357325	

Showing 1 to 10 of 23 entries

Success

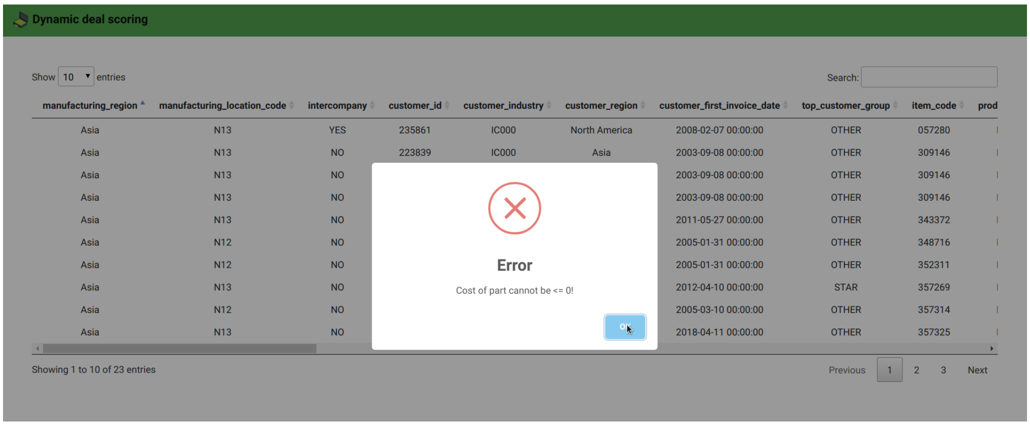
Manufacturing region: Asia

----

customer_industry = IC009
customer_region = Asia
make_vs_buy = MAKE

GM cutoffs:
0.003, 0.266, 0.360, 0.517, 0.658, 1.000

Figure 5: Successful prediction display.



Dynamic deal scoring

Show 10 entries

Search:

manufacturing_region	manufacturing_location_code	intercompany	customer_id	customer_industry	customer_region	customer_first_invoice_date	top_customer_group	item_code	prod
Asia	N13	YES	235861	IC000	North America	2008-02-07 00:00:00	OTHER	057280	
Asia	N13	NO	223839	IC000	Asia	2003-09-08 00:00:00	OTHER	309146	
Asia	N13	NO				2003-09-08 00:00:00	OTHER	309146	
Asia	N13	NO				2003-09-08 00:00:00	OTHER	309146	
Asia	N13	NO				2011-05-27 00:00:00	OTHER	343372	
Asia	N12	NO				2005-01-31 00:00:00	OTHER	348716	
Asia	N12	NO				2005-01-31 00:00:00	OTHER	352311	
Asia	N13	NO				2012-04-10 00:00:00	STAR	357269	
Asia	N12	NO				2005-03-10 00:00:00	OTHER	357314	
Asia	N13	NO				2018-04-11 00:00:00	OTHER	357325	

Showing 1 to 10 of 23 entries

Error

Cost of part cannot be <= 0!

Figure 6: Prediction error display.