Exercise project 3 – Feature engineering

Use the same datasets as in the previous exercise projects, if possible. In case your dataset doesn't have enough things for optimization, you can use another smaller dataset to demonstrate missing optimization steps for more points):

- ***Regression dataset***
- ***Classification dataset***

It's also okay to use whatever information you find during this exercise project in any other on-going machine learning course from the same instructor, like deep learning!

It's advisable to make multiple Jupyter notebooks for each dataset, since the notebooks get easily very cluttered.

Step 1:

Use the information found in Exercise project 1, and decide proper optimization attempts for each dataset you have.

Try each optimization to at least one of the dataset if possible. If it is not viable to try a certain approach to either dataset (because there's nothing to fix), you can search for a more simple example dataset to demonstrate how it works for a given dataset.

Try to test at least one tool of each group:

- **Optimal variable selection**
  - **Use any number of common variable importance selection tools to determine if you can remove some variable(s) from your dataset**
  - **Note: sometimes there's nothing to remove, so don't remove variable(s) if none of the tools suggest that**
  - **Remember also removing redundant variables or variables with multicollinearity if needed!**
    - **VIF-test, correlation/phik-matrix, SHAP/LIME, similarity in distributions, Mutual Information…**
  - **Common tools (use at least 2 of them and cross validate their results)**
    - **SelectKBest, Fisher score**
    - **RFE**
    - **Mutual Information**
    - **Decision Trees**
    - **SHAP, LIME**
    - **Correlation matrix, phik-matrix**
    - **etc. (you're free to use other tools as well if you please)**

- **Variable combinations, feature generation and interaction features (try at least one category of operations):**
  - **Interaction features:**
    - **Multiplicative interactions**
    - **Additive interactions**
    - **Categorical interactions**
    - **Ratio features**
  - **Deep Feature Synthesis (DFS)**
    - **Determine potential variables for combinations**
    - **Create new variables based on existing data**
    - **Evaluate the most important variables with optimal variable selection tools (previous page)**
  - **Cluster profiling**
    - **In addition to cluster values:**
      - **Calculate average etc. of some variable based on each cluster**

- **Redundancy management and dimensionality reduction:**
  - **PCA – Principal Component Analysis**
  - **Evaluate also the variable importances for each component and lost accuracy of dimensionality reduction**

- **High cardinality management (optional)**
  - **Agglomerative clustering (don't use with extremely large datasets, or you might crash your computer)**
  - **Target encoding**
  - **Frequency encoding**
  - **Feature hashing**
  - **Grouping rare categories together**
  - **Derive new variables from high cardinality variable(s) to reduce cardinality**
  - **Remember CategoryEncoders –module!**

See the examples from lectures (Git) on how to use each one of the tools and methods. Other tools not listed above are also welcome, but make sure they are actually useful in feature engineering.

Step 2:

After choosing the methods from previous step and applying them to your datasets, re-analyze quickly again the major problems in your datasets (based on analysis done on Exercise project 1) and evaluate were you able to fix the found problems in this exercise project.

If some problem seems to be difficult to fix with these tools, discuss potential reasons and ideas on why that might happen with your datasets.

Step 3:

Finally evaluate the new optimized dataset as a whole. How close did you get to "optimal" regarding feature engineering, like variable redundancy, optimal variables etc. ?

Which methods were most effective in your dataset?

Do the datasets still represent the real life phenomenon well? (compare averages, deviation, variance, correlations etc. to the original dataset and modified dataset, how different they are now?

Use df.describe() and df.var() to get mean, median, deviation and variance. Also consider inspecting correlations and phik-matrix.

Ideally the previously mentioned metrics are not too far off from the original dataset.

Remember: also in feature engineering, it's possible to have a situation where the dataset has transformed into an unrealistic representation of the original dataset. In other words, it's perfect in theory, but not in practice. This is why we need balance when we do optimization, and sometimes we need to be reasonable with the optimizations when aiming for most optimal solutions.