

Problem Statement: Products Top/Flop Prediction

Task Summary

A fashion e-commerce company is planning its collections for the upcoming year. Therefore the company put together many potential products as candidates and now would like to estimate which products would be successful (top) or not (flop). To do so, you are provided with data on the past years' top and flop products. This will allow you to create a small machine-learning application (without a UI).

Data Overview (shared in separate files)

You have two data sets:

- Historic data: Products of the past two years and their attributes (including a label that categories the item stop or flop); file: historic.csv (8000 products)
- Prediction data: Potential products of the upcoming year and their attributes (but no label about the success); file: prediction_input.csv (2000 product candidates)

Columns:

- item_no: Internal identifier for a past product or a product candidate for the future.
- category: Category of the product.
- main_promotion: Main promotion that would be/was used to promote the product.
- color: The main color of the product.
- stars: Stars of reviews from a comparable product of a competitor (from 0= very negative reviews to 5 = very positive reviews).
- success_indicator: Indicator whether a product was successful (top) or not (flop) in the past. Only given for the historic data.

Technical Requirements

- EDA:
 - Perform EDA and submit in a notebook named "eda.ipynb".
 - For each analysis/chart code block add a comments markdown block.
- Modeling:
 - Prepare the training pipeline and submit it in a script named "model_<model_name>.py". Where <model_name> is the model algorithm you are using. Create at least two different models and corresponding model files, one of them must be an artificial neural network (ann) model.

- Object-oriented, class-based approach to be followed. Model classes must consist of at least the following functions: load (to load data), preprocess (preprocessing steps), train (training steps), test (test steps, that also generate evaluation summary), and predict (for inference).
- Model Selection:
 - Prepare the model selection pipeline and submit it in a script named “model_selction.ipynb”
 - Run different models prepared in the “Modeling” task and showcase evaluation metrics. You can perform hyperparameter tuning if required.
 - Finally, choose one model and provide a summary of “Why the model was chosen?” in the notebook at the end.
- The notebooks shared should be pre-runned and with the output cells
- Use appropriate coding standards and provide in-line comments where required.
- Use Python 3.7 +
- The final submission should be a zip file with all the required files described above named “<submission_date>_<full_name>.zip”

Notes

- You will agree to only submit your own work! You are free to do research online or in the relevant literature.
- The given data sets are stochastically generated and are not real data of our company. Nevertheless, it is possible to make proper predictions based on the attributes.
- Please submit by 6PM after obtaining the challenge.