Classification

Machine Learning
uczeniemaszynowe@cs.uni.wroc.pl

04.11.2025, 10:00
Lab

Task 1. [1 point]

Repeat task 3 from list 2 for the kNN model. Use the dataset `advertising.csv`. Build kNN models using both the original data (with unscaled variables) and normalized data. Compare these two models, record your observations, and write conclusions.

Task 2. [2 points]

- Build a logistic regression model using the dataset `bank.csv`[1]. Compare the results for the thresholds `[0.1, 0.25, 0.5, 0.75, 0.9]`.
- Then, compute TPR and FPR for thresholds `[0, 0.01, 0.02, ..., 0.98, 0.99, 1]` and plot the ROC curve.
- Choose the optimal threshold assuming that the cost of a Type I error (False Positive) is 10 and the cost of a Type II error (False Negative) is 3.
- Select the two best predictors and build a model using only these two variables. For the chosen threshold, mark the regions classified into each of the two classes. What characteristic pattern do you observe? How does the boundary between the classes relate to the model's parameter vector?

Task 3. [2 points]

In this task, we will examine how similarities between products are encoded in product embeddings obtained from a recommender system[2].

The file `embeddings.pkl` contains a matrix of product embeddings, where each row of the matrix corresponds to one product. These embeddings were obtained from a recommender system. The file `item_list.txt` contains the original product identifiers. The file `meta_Books.json.gz`[3] contains product metadata — our embeddings concern products from the Books category. The product identifier is stored in the `asin` field.

[1] https://www.kaggle.com/datasets/janiobachmann/bank-marketing-dataset/data
[2] https://github.com/kuandeng/LightGCN
[3] more information and further instructions: https://cseweb.ucsd.edu/ jmcauley/datasets/amazon/links.html

Split the data into training and test sets; if necessary, you may also create an additional validation set. Use a kNN model to classify product categories and subcategories, which can be found in the metadata. Check which distance metric is most appropriate for this task.
Hints:

- Pay attention to the different input data formats. We have a file stored in binary format using the `pickle` library, a text file, and a compressed file in which each line is a JSON object.
- Information about available metrics in the scikit-learn package can be found here: https://docs.scipy.org/doc/scipy/reference/spatial.distance.html.
- Check whether a product belongs to one or multiple classes and consider how such a situation should be handled. You may find it convenient to propose your own model implementation.

Task 4. [2 points]

Build a linear regression model using the `adult.csv` dataset[4] to predict whether an individual earns more than $50k. Use the `statsmodels` library[5]. Perform variable selection using the following procedure:

- Build a baseline model.
- Check the results of adding each unused variable.
- Identify the variable that gives the greatest improvement in model quality.
- Using a selected information criterion (AIC, BIC), determine whether it makes sense to include this variable in the model. If so, include the variable in the model and identify the next best predictor. Otherwise, do not include the selected variable and terminate the procedure.

Evaluate the final model.
Hints:

- You can download the data directly from Kaggle:

```
import kagglehub

# Download latest version
path = kagglehub.dataset_download("uciml/adult-census-income")

print("Path to dataset files:", path)
```

- Useful methods and attributes: `model.summary()`, `model.aic`, `model.bic`.
- To fit the intercept, use `statsmodels.tools.add_constant`.

[4] https://www.kaggle.com/datasets/uciml/adult-census-income
[5] https://www.statsmodels.org/stable/generated/statsmodels.discrete.discrete_model.Logit.html

Task 5. [3 points]

Familiarize yourself with the `Titanic Dataset.csv`[6]. Fill in the missing values in two different ways, following the instructions below. Then compare the linear regression models predicting the variable `survived`.

- Check what percentage of missing values occurs in the subsequent columns.
- Check whether there is a relationship between categorical variables and missing values in the numerical columns.
- Baseline solution:
  - remove columns with a high share of missing values,
  - impute missing numerical values with the mean of the feature,
  - for categorical variables, use the most frequent value.
- Advanced solution:
  - replace the `Cabin` variable with a categorical feature indicating whether the value was missing,
  - extract the titles ("Mr", "Mrs", ...) from the passenger names, use them to impute the ages of the passengers by filling in the median age for individuals with a given title,
  - others as above.

Discussion for the lab

- What other procedures for variable selection for the model are possible?
- Analyze the rationale of the proposed methods for imputing missing values for the subsequent columns.

[6] https://www.kaggle.com/code/sakshisatre/titanic-s-missing-data-visualizing-null-values/input