Machine Learning
Programming Assignment II-a

Shuai Wang, Marleen de Jonge and dr. Stevan Rudinac

The following assignment will test your understanding of topics covered in the first three weeks of the course. This assignment **will count towards your grade** and should be submitted through Canvas by **28.11.2024 at 23:59 (CET)**. You must submit this assignment in groups (as registered on Canvas). You can get at most 5 points for this assignment, which is 5% of your final grade.

1 Instructions

- Alongside the code for your experiments, you are also required to present a report summarizing the observations and results of each of the experiments. You can use text and graphs/plots (`matplotlib`) for these reports. You should place these report blocks within the Jupyter Notebook in separate text cells. Plots can be appropriately placed near the text explanation. Your final submission should be a single Jupyter Notebook with code and report blocks.
- While it is perfectly acceptable to brainstorm and discuss solutions with other colleagues, please, do not copy code. We will check all submissions for code similarity with each other and with openly-available solutions on the web.
- Please, ensure that all code blocks are functional before you finalize your submission. Points will NOT be awarded for exercises where code blocks are non-functional.

2 Submission

You can submit your solutions within a Jupyter Notebook (*.ipynb). To test the code we will use Anaconda Python (3.10). Please, state the names and student ids of the authors at the top of the submitted file.

3 Implementation Details

In this assignment, you will be working on feature scaling and classification tasks. Since they have a fixed sequence of execution, it is required to use the sklearn `Pipeline` functionality to encapsulate your preprocessing transformations as well as classification models into a single estimator. In the following assignment, you should perform preprocessing, model fitting and prediction operations only with a `Pipeline` estimator.

Any grid search should also be performed on the `Pipeline`, not on standalone estimators or transforms.

4 Data

The popularity of online news can depend on multiple factors like the website where it was published, content, etc. Websites often track viewing statistics for individual articles to better understand the type of content readers are looking for. In this assignment, you are provided with data on 39k online news

✉ s.wang3@uva.nl, m.r.h.dejonge@uva.nl, s.rudinac@uva.nl

articles [Fernandes et al., 2015]. This dataset contains statistics on the word, topic and sentiment level for each article.

In the zip file OnlineNewsPopularity.zip, you will find two accompanying files:

- A data file named OnlineNewsPopularity.csv. This is the dataset that you have to use for Programming Assignment II.
- A companion document OnlineNewsPopularity.names with a brief explanation of the features present in the data. Please read this document carefully before starting this assignment.

The dataset contains 58 features that encode various attributes, inter alia, the number of words in the article, the total number of links within the article or the absolute polarity level of the article. The continuous variable ”shares” reports the total number of shares for a specific article. Please read the OnlineNewsPopularity.names document for a summary of the dataset.

5 Data Preprocessing

Similar to the previous assignment, `pandas` can help with loading and preprocessing the raw data. For the preprocessing stage of this assignment, you will need to perform the following tasks:

1. Load the data (CSV) file.
2. Inspect individual features to ensure they are in the right datatype. `Pandas` will try to intelligently infer the correct datatype, but you still need to inspect the results yourself.
3. Features that contain categorical data should be converted to a one-hot encoding. You will find `pd.get_dummies()` or `sklearn.preprocessing.OneHotEncoder` helpful for this task. Please, remember that these operations must be performed on the data before the train/test split.
4. For the classification task that you opt to solve for this assignment, you should convert the target variable ”shares” to a binary variable (0 or 1) using a specified predefined threshold set to 1,400, e.g. articles with less than 1,400 shares will be considered not popular, whereas articles with 1,400 or more shares will be considered popular.

6 Models

In this exercise, you will build pipelines with 2 components:

1. Feature Scaling: The range of raw values can vary widely in a dataset. To bring this variation within the same scale, feature scaling is helpful. For this task, you are asked to experiment with the `StandardScaler` and `MinMaxScaler` provided within sklearn to scale your data.
2. Classification: The second component of your pipeline is a linear classifier. In this homework, you are asked to use the `LogisticRegression`, `LinearSVC (SVM)` and `KNeighborsClassifier` classifiers. For these classifiers, you must perform the following experiments:

   (a) For a `LogisticRegression` classifier.

      i. Use `GridSearchCV` to find an optimal value for the “inverse of regularization strength” `C`.

      ii. Fit your classifier on scaled as well as unscaled versions of the data and report the estimator scores. Report your observations on the effects of scaling on model performance.

   In not more than 50 words, present your observations on the effects of `C` and feature scaling on model performance. This explanation should summarize your observations from the experiments above. You can use a text box (i.e. Markdown Cell) in Jupyter to write down your analysis. Feel free to experiment with other hyperparameters as well.

   (b) For a `LinearSVC (SVM)` classifier.

      i. Use `GridSearchCV` to find an optimal value for the regularization parameter `C`.

ii. Fit your classifier on scaled as well as unscaled versions of the data and report the estimator scores. Report your observations on the effects of scaling on model performance.

In not more than 50 words, present your observations on the effects of `C` and feature scaling on model performance. This explanation should summarize your observations from the experiments above. You can use a text box (i.e. Markdown Cell) in Jupyter to write down your analysis. Feel free to experiment with other hyperparameters as well.

(c) For a `KNeighborsClassifier` classifier.

i. Use `GridSearchCV` to find an optimal value for the "number of neighbors" `n_neighbors`.

ii. Fit your classifier on scaled as well as unscaled versions of the data and report the estimator scores. Report your observations on the effects of scaling on model performance.

In not more than 50 words, present your observations on the effect of `n_neighbors` and feature scaling on model performance. This explanation should summarize your observations from the experiments above. You can use a text box (i.e. Markdown Cell) in Jupyter to write down your analysis. Feel free to experiment with other hyperparameters as well.

7 Evaluation Metrics

For each of the pipelines, you must report the following classification metrics:

1. Accuracy
2. Macro and Micro-Averaged Precision and Recall
3. F1 Score

Additionally, present your observations on what these scores mean for the models under consideration. These metrics will be discussed at the beginning of Week 5.

8 Grading

Component	Points
Scaling	1
Classifiers	1
Hyperparameter Optimization	1
Experiments, Observations, Analysis & Code Quality	2

References

[Fernandes et al., 2015] Fernandes, K., Vinagre, P., and Cortez, P. (2015). A proactive intelligent decision support system for predicting the popularity of online news. In *Portuguese Conference on Artificial Intelligence*.