Machine Learning
Programming Assignment II-a

Ujjwal Sharma, Athanasios Efthymiou and dr. Stevan Rudinac

The following assignments will test your understanding of topics covered in the first three weeks of the course. These assignments **will count towards your grade** and should be submitted through Canvas by **26.11.2020 at 08:59 (CET)**. You can choose to work individually or in pairs. You can get at most 5 points for these assignments, which is 5% of your final grade.

Submission

You can only submit a Jupyter Notebook (*.ipynb) for this homework. To test the code we will use Anaconda Python 3.7. Please state the names and student ID's of the authors (at most two) at the top of the submitted file. Please rename your notebook to add the name of the TA for the group you are assigned, e.g. assignment2a_{first_student_name}_{second_student_name}_{ujjwal_or_thanos}.ipynb. If you plan on working alone, please name the file as assignment2a_{student_name}_{ujjwal_or_thanos}.ipynb.

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.

1 Implementation Details

In this assignment, you will be looking at 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 assignments, 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.

✉ a.efthymiou@uva.nl, u.sharma@uva.nl, s.rudinac@uva.nl

2 Data

With this assignment, you will receive two additional files:

- A data file titled `data.csv`.
- An PDF document `desc.pdf` containing descriptions for individual features in the data.

This dataset contains user-interaction data from an e-commerce portal [Sakar et al., 2019]. The data contains 18 features that encode various parameters concerning user interaction like clicks, dwell time etc. and boolean label (`Revenue`) that reports if the user made a purchase or not. Please read `desc.pdf` for a complete description of the dataset.

3 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 meaningful 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. Since the column "Month" contains cyclical data, you can remove it from the data as this feature requires an encoding that has not been discussed in class.

4 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 classifier. In this homework, you are asked to use the `LinearSVC`, `LogisticRegression` and `KNeighborsClassifier` classifiers. For these classifiers, you must perform the following experiments:
   (a) For a `LinearSVC` 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.

   (b) 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.

(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.

5 Evaluation Metrics

For each of the pipelines, you must report classification metrics like accuracy, recall, F1 and micro/macro averaged precision statistics and present your observations on their implications for each of the models. These metrics will be discussed at the beginning of Week 5.

6 Grading

Component	Points
Scaling	1
Classifiers	1
Hyperparam Optimization	1
Experiments, Observations, Analysis and Code Quality	2

References

[Sakar et al., 2019] Sakar, C. O., Polat, S. O., Katircioglu, M., and Kastro, Y. (2019). Real-time prediction of online shoppers' purchasing intention using multilayer perceptron and lstm recurrent neural networks. *Neural Computing and Applications*, 31(10):6893–6908.