Machine Learning
Programming Assignment I

Ujjwal Sharma and dr. Stevan Rudinac

The following assignments will test your understanding of topics covered in the first two weeks of the course. These assignments **will count towards your grade** and should be submitted through Canvas by **14.11.2019 at 12:59 (CET)**. You can choose to work individually or in pairs. You can get at most 10 points for these assignments, which is 10% of your final grade.

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. 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.
- 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.

Submission

You can submit your solutions within a Jupyter Notebook (*.ipynb). To test the code we will use Anaconda Python 3.6. Please state the names and student ids of the authors (at most two) at the top of the submitted file.

1 Data

In the zip file for this assignment, you will find two accompanying files:

- A data file named `data.csv`. This is the dataset for Assignment 1.
- A companion document `feature_desc.pdf` with a brief explanation of the features present in the data.

The supplied data consists of 11 *features* and a single real-valued label. Please refer to the accompanying `feature_desc.pdf` file for a detailed explanation of the data. The last column of the data (`satisfaction`) is the label and should be extracted before training. You will find the `pandas` library extremely helpful for working with this data. For each experiment, you are also required to split the data into train, validation (if needed) and test splits. Choose an appropriate split ratio accordingly.

✉ u.sharma@uva.nl, s.rudinac@uva.nl

2 Regression

In this week, you've been introduced to the *Regression* task which models relationships between a real-valued scalar *dependent* variable and multiple *independent* variables. In this task, we will use regression models to regress a consumer satisfaction score (`satisfaction`) for a paper supplier given the supply/demand dynamics with three customers. Each of your models will use 11 predictive features or independent variables to regress the real-valued satisfaction score in the final column of the accompanying data file.

2.1 Regression : Tasks

For this assignment, you will implement the following regressors:

1. An ordinary least-squares linear regression model. Available from `sklearn.linear_model`
2. A ridge regression model that adds L2 regularization to the ordinary least-squares model. Available from `sklearn.linear_model`
3. A lasso regression model that adds L1 regularization to the ordinary least squares model. Available from `sklearn.linear_model`

For the above models, you will perform the following tasks:

1. Fit the ordinary least-squares model to the data. Once completed, report the *mean squared error* and the R^2 coefficient of determination.
2. Fit the ridge and lasso regression models to the data. To find an optimal value for the α hyperparameter, you can use the scikit-learn grid search functionality in `sklearn.model_selection.GridSearchCV`. You will need to compute the optimal α for both models. Report the best α value from your search.

Tip: For grid search over parameters, it may be a good idea to consult the `sklearn` documentation to check the default value for that parameter and devise a suitable search strategy.

3 Classification

Before starting with the classification task, you will need to convert the real-valued `satisfaction` variable into discrete class labels. We can use the condition below to create these labels:

$$\text{satisfaction label} = \begin{cases} 1, & \text{if satisfaction} \geq 0 \\ 0, & \text{otherwise} \end{cases} \tag{1}$$

3.1 Classification : Tasks

In this section, we will use the 11 features to predict if there is general satisfaction (or dissatisfaction) with the paper producer. For this task, you will implement the following classifiers:

1. A k-nearest neighbor (k-NN) model. The model is available from `sklearn.neighbors`
2. A linear SVM model with default parameters. Available from `sklearn.svm`

For the above models, you will perform the following tasks:

1. Fit the k-NN model to the data. The value of k should be obtained from a grid-search. Once completed, report the *accuracy score* averaged over test data. Additionally, report the best model from your search and its parameters.
2. Fit the SVM model to the data and report the *accuracy score* averaged over test data.

Hint: For the SVM task, use the `sklearn.svm.LinearSVC` estimator instead of `sklearn.svm.SVC`. The `LinearSVC` estimator is written on a newer framework and is significantly faster.

Grading

Experiment	Points
OLS Regression	1
Lasso Regression	1
Ridge Regression	1
k-NN Classification	1
SVM Classification	1
Grid Search and Cross Validation	2
Report and Code Quality	3