INF264 - Homework 3

Natacha Galmiche

Instructions

- **Format: Jupyter notebook**
- **Deadline: Friday 16th September, 23:59**
- **Submission place:** `https://mitt.uib.no/courses/36686/assignments/60593`

1 Model selection for regression

In this exercise, we want to compare a model selection using a simple validation set with more refined model selection method such as the "KFold cross-validation" procedure.

Consider once again the Boston Housing dataset: `https://www.cs.toronto.edu/~delve/data/boston/bostonDetail.html`. Remember that in this dataset, each sample corresponds to a house, whose target price is to be inferred from 13 features contained in the dataset. You will implement a code that answers the following questions (you can get inspiration from the provided template code):

1. Load the Boston dataset: store the 13 features in a matrix X and the target price in a vector Y. **Hint:** You can use `load_boston()` from `sklearn.datasets`.
2. Investigate the Boston dataset. It is obviously not possible to visualize all the 14 dimensions at the same time, but it may be a good idea to represent the target price as a function of each of the 13 features individually. **Hint:** You can plot using the `scatter` function from `matplotlib.pyplot`.
3. Identify (at least) 2 features that are continuous and appear to be correlated to the target price. Extract these features to obtain a simpler feature matrix with fewer columns. **Note:** Ideally, the extracted features should be as (pairwise) decorrelated as possible.

We want to build a polynomial regression model that learns the target price from the extracted features. We are not sure about which polynomial order we should use to obtain the best model. Moreover, we were told that adding a regularization term can be useful to prevent overfitting, but once again we do not know how to choose this hyper-parameter. A solution is to perform model selection with k-fold cross-validation.

4. Implementation of a KFold cross-validation procedure in order to perform model selection of a Ridge model with respect to its hyper-parameters (polynomial order and regularization value), using the MSE metric. More specifically:
   - i Split the whole dataset into a train and a test set.
   - ii Split the train set into 5 (train,validation) k-folds (so $k = 5$ here). **Hint:** use `KFold` class from `sklearn.model_selection`.
   - iii Loop over each hyper-parameters instance
   - iv For current instance of hyper-parameters, loop over each (train, validation) fold
   - v In the inner loop, using the current set of hyper-parameters and the current (train,validation) fold, train a model on the train fold. **Hint:** You can first use `sklearn.preprocessing.PolynomialFeatures` to transform your data in a similar way as what we manually did last week in the exercise `basis_functions`, and then, you can use `sklearn.linear_model.Ridge` to build a linear regression model on this polynomial data.
   - vi Still in the inner loop, evaluate your current model using the MSE metric on the validation fold. **Hint:** you can use `mean_squared_error` from `sklearn.metrics`.
   - vii Now in the outer loop (after the inner loop finishes), compute the mean validation MSE over each (train,validation) fold.
   - viii Select the best model configuration based on the lowest mean validation MSE.

ix Train the selected model on the whole train set (train + validation) then evaluate the performance of your selected model using the test set.

Cross-validation can be done on the polynomial order ranging in $\{1, 2, 3\}$ and on the regularization value ranging in $\{0, 0.001, 0.01, 0.1\}$ for a total of 12 hyper-parameters combinations. Indicate which hyper-parameters combination obtained the best results with respect to the MSE metric during the KFold cross-validation procedure.

5. When fitting a Ridge model of 3rd degree, you should have encountered the warning "Singular matrix in solving dual problem. Using least-squares solution instead.". Can you explain why this warning occurred? How much do you need to increase the regularization hyper-parameter in order to get rid of this warning?

6. Perform a regular model selection using a simple validation set to select the best Ridge model with respect to the polynomial order and the regularization value. Comment on the differences with the KFold cross-validation procedure.

2 Model selection for classification

In this second exercise, we will first illustrate how misleading the accuracy metric can be when assessing a classifier on unbalanced data. Finally, we will cross-validate different classifiers on an unbalanced dataset with a relevant metric (you can get inspiration from the provided template code):

1. Creating your own unbalanced dataset

   (a) Create randomly generated binary datasets, with a 1st class ratio ranging in $\{0.6, 0.75, 0.9, 0.95, 0.98, 0.99\}$. **Hint:** You can use `sklearn.datasets.make_classification`

   (b) For each generated dataset, train and select a K-NN classifier using a regular model selection (no cross validation) and the accuracy as performance metric. **Hint:** Use `accuracy_score` from `sklearn.metrics`.

   (c) For each generated dataset, evaluate the selected model on the test dataset using successively the accuracy metric and the $F1$-score metric. In addition, for each generated dataset, plot the confusion matrix. **Hint:** Use `f1_score` and `confusion_matrix`, from `sklearn.metrics`.

   (d) Plot the accuracy and the f1 score with respect to the 1st class ratio in a single figure. Does the accuracy metric appear to assess the quality of your model in an appropriate way?

2. Using a predefined dataset

   (a) Load the custom randomly generated binary dataset contained in the file `custom_unbalanced_dataset.pickle`.

   (b) Visualize this dataset. How unbalanced is it?

   (c) Perform model selection on three different classification models: a K-NN classifier, a logistic regression classifier and a decision tree classifier. Use Kfold cross-validation with a number of fold sets $k > 5$. Indicate and justify which metric you decided to use to cross-validate the different models.

   (d) Train and evaluate the best model with the $F1$-score and the confusion matrix.