Machine Learning

Programming Assignment III

Thanks to: Ujjwal Sharma, Shuai Wang, Hongyi Zhu, and Ilker Birbil

The following assignments will test your understanding of topics covered in the first five weeks of the course. This assignment will not count towards your grade but should be submitted through Canvas. You must submit this assignment in teams of 2 or 3.

- Alongside the code for your experiments, you are also required to present a report summarizing the observations and results of each of the experiments. For these reports, you can use text and graphs/plots (matplotlib). 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 openly available web solutions.
- Please ensure that all code blocks are functional before you finalize your submission.

Submission

You can submit your solutions within a Jupyter Notebook (*.ipynb). To test the code, we will use Anaconda Python (3.13). Please state the names and student ids of the authors at the top of the submitted file.

Part 1: Classification

Classification on Wine Dataset

In assignment one, you've used several regression models to predict the quality of wine. However, the label quality is not strictly continuous. Since quality is an integer in [0, 10], it can also be seen as a discrete, categorical label. Consequently, this problem can also be seen as a multi-class classification task. In this exercise, we ask you to predict wine quality in a multi-class classification setup using a LogisticRegression classifier. In less than 50 words, present the potential advantages and drawbacks of viewing this problem as a multi-class classification task.

Part 2: Gridsearch, Scalars, KNeighbors, LinearSVC, LogisticRegression

Implementation Details

In this assignment, you will work 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 and 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.

Data

With this assignment, you will receive two additional files:

- Data files titled train.csv, test.csv and test_label.csv.

The dataset relates to a Portuguese banking institution's direct marketing campaigns (phone calls). The data contains 17 features that encode various parameters such as age, job, marital status, and education. The classification goal is to predict if the client will subscribe to a term deposit The boolean label "y". You should train and valid on the train.csv, and test on test.csv and test label.csv.

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.

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 or 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.

Part 3: 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.

Part 4: Tree Models

The model structure introduced in Assignment II-A uses a Pipeline to wrap a *scaler* and a *classifier*. Since scaling is not needed for tree-based models, the resulting pipeline will end up containing only the classifier. Thus, for the tree-based models specified in this assignment, the usage of a pipeline is NOT required.

In this assignment, you are asked to use the DecisionTreeClassifier and RandomForestClassifier as the classifiers. With these models, you must perform the following experiments:

1. Initialize a DecisionTreeClassifier model and perform the following tasks:

(a) Fit the classifier on the data and report the model accuracy.

(b) Report all evaluation metrics listed in Section 3.

(c) Answer the question: Are these models resilient to overfitting when model hyperparameters have not been carefully selected? Supplement your explanation with suitable figures/tables if necessary. Please limit your explanation to a maximum of 50 words.

(d) Perform a grid search on the decision tree model parameters ['max_depth', 'max_features'] to evaluate the optimal values for these parameters. Report model hyperparameters for the best classifier model.

In less than 50 words, explain your observations and your assessment of the classifier. You can use a text box (i.e., Markdown Cell) in Jupyter to write down your analysis.

2. Initialize a RandomForestClassifier model and perform the following tasks:

(a) Fit the classifier on the data and report the model accuracy.

(b) Report all evaluation metrics listed in Section 3.

(c) Answer the following questions:

i. How are decision tree classifiers different from random forests on a structural level? (max. 50 words)

ii. Where would you choose decision trees over random forests and vice-versa? Demonstrate this using an appropriate example from your data. (max. 50 words)

iii. Is accuracy an appropriate evaluation metric for this classification task? if yes, in what kind of data may it not be a good metric? Justify your answer in less than 20 words.

(d) Perform a grid search on the random forest model parameters ['max_depth', 'max_features' 'n_estimators'] to evaluate the optimal values for these parameters. Re- port model parameters for your best classifier model.

In no more than 50 words, explain your observations and your assessment of the classifier. You can use a text box (i.e., Markdown Cell) in Jupyter to write down your analysis.

Part 5: Tree Model Evaluation

For all models in Assignments II-A and II-B, you must report:

3. Accuracy
4. Macro and Micro-Averaged Precision and Recall
5. F1 Score

Additionally, present your observations on what these scores mean for the models under consideration.

Grading

This assignment will not count towards your grade but should be submitted through Canvas.

References

[Cortez et al., 2009] Cortez, P., Cerdeira, A., Almeida, F., Matos, T., and Reis, J. (2009). Modeling wine preferences by data mining from physicochemical properties. *Decision support systems*, 47(4):547–553.