CSE2510: Machine Learning
Created by Jordi Smit and Gosia Migut
Revised by Aleksander Buszydlik and Tom Viering

Optional Assignment: Applications of ML

October 22, 2024

1 Introduction

In this course, you have encountered many machine-learning algorithms and concepts. Now it is time to put your knowledge to the test. This assignment asks you to explore the effectiveness of five machine-learning algorithms in different scenarios. We expect you to examine the algorithms critically and properly analyze their effectiveness.
Note: This assignment is fully optional and the material covered here is not part of the main learning goals of the course. However, this is a fun way to get some more experience with applying the concepts that you have learnt in the course. You can do the assignment on your own or with a teammate. In many questions, we have added a word limit to help you estimate how long your answer should be.

1.1 Assignment

In this assignment, you will explore two datasets: American Census and MNIST. Each of these has some problems that you will have to overcome to find the best possible classifier. You will work with the following algorithms:

- Gaussian Naive Bayes (week 2);
- K-Nearest Neighbors (week 3);
- Logistic Regression (week 4);
- Support Vector Machine (week 4);
- Decision Tree (week 6);

Please, follow the structure below:

1. **Data exploration**: In this step, we expect you to explore the properties of the dataset. This includes both the features and the target variables. At the end of this phase, you have to come up with the most suitable evaluation metric and the baseline that your classifier should beat for the chosen metric.
2. **Data preparation**: In this step, we expect you to transform the data into the expected format required by the algorithms. Note that the optimal format could differ per algorithm. Your actions may include data cleaning, encoding, transformations, etc.
   Note: We don't expect you to do any outlier handling (this can be very time-consuming).
3. **Experiments**: In this step, we expect you to do two things. First, you should fit and fine-tune the algorithms which may include hyper-parameter selection, grid search, etc. Second, you should compare the different models and come up with the best one for the task; for this best model you will have to supply its predictions on our test data.

While working on this assignment you are allowed to use the Python library Scikit-learn. All of its machine learning algorithms follow the same API: you only need to know what the `fit`, `predict`, and `test` methods do.

1.2 Suggested reading material

You might encounter some definitions that may be new to you, or you may have forgotten them. We expect you to research them yourself as part of this assignment. For this, you can always use the provided reading material, however, sometimes it might be beneficial to read about the topic from another perspective. Luckily, the online ML community is vibrant, you can find dozens of explanations for each concept. To help you get started:

- Baseline algorithm
- Preprocessing continuous features
- Preprocessing categorical features
- Preprocessing missing values
- Classification metrics
- Hyper-parameters tuning
- Cross-validation
- Pipelines (for pre-processing and training in one go)
- Scikit-learn Glossary

2 INTRODUCTORY QUESTIONS

The next sections will help you go through the machine learning experiment process. Each section has multiple questions; all of them must be answered in the report.

2.1 ALGORITHMS

Before we dive into the datasets, let's first explore the five different algorithms. For each algorithm, we will start with the following default hyper-parameters:

- `GaussianNB`
  - No hyper-parameters to be tuned
- `DecisionTreeClassifier`
  - `max_depth=None` (to be tuned)
  - `min_samples_leaf=2` (to be tuned)
  - `random_state=42`
- `KNeighborsClassifier`
  - `n_neighbors=3` (to be tuned)
  - `weights="distance"` (to be tuned)
- `SVC`
  - `C=10` (to be tuned)
  - `kernel="poly"` (to be tuned, consider: `"poly"`, `"linear"`, `"rbf"`)
  - `degree=3` (to be tuned)
  - `gamma="scale"` (to be tuned, consider: `"scale"`, `"auto"`, `"float"`)
  - `random_state=42`
- `SGDClassifier with log loss (Logistic Regression)`
  - `loss="log_loss"`
  - `alpha=10` (to be tuned)
  - `learning_rate="constant"`
  - `eta0=0.1` (to be tuned)
  - `penalty="none"` (to be tuned, consider `"l2"`, `"l1"`, `"none"`)
  - `random_state=42`

Note 1: We do not expect you to set other hyper-parameters (leave them at default values).
Note 2: Take special note of the `random_state` variables to ensure deterministic results.
Note 3: We ask you to use the `SGDClassifier` instead of the `LogisticRegression` (which also exists in Scikit) to have more control over convergence. You can always plot a training curve using the `partial_fit` method to decide if your classifier has converged.

1. For each of the five algorithms list one key strength and one key weakness. We do not count having more or less hyper-parameters as a valid strength or weakness. Use no more than 300 words in total (± 60 per algorithm).

2. Carefully read the Scikit-learn hyper-parameter documentation for each of the five algorithms. Based on this documentation explain how the hyper-parameters above affect the behavior of the classifier. Use no more than 400 words in total (± 100 per algorithm).
   Note 1: You do not have to write anything about the Naive Bayes classifier.
   Note 2: Also explain if some hyper-parameter A must be tuned based on the value of another hyper-parameter B.

3 US Census

In this part of the assignment, you will explore the dataset from the United States 1994 Census. It is your job to create a classifier that can predict whether a person makes over \$50,000 a year using this dataset. It has the following attributes:

- `age`;
- `education-num`: the number of years a person spent following any form of education;
- `hours-per-week`: how many hours per week a person works;
- `work-class`: the type of employment a person has;
- `education`: the highest level of completed education;
- `marital-status`;
- `occupation`: the sector that the person works in;
- `relationship`;
- `race`;
- `sex`;
- `native-country`;
- `salary`: the target variable;

Before you get started with this dataset please read all the questions below. They will give you some direction in your experiments. Once you have done this, don't start coding right away. First, explore the dataset a bit, this will make answering the questions significantly simpler.

Note: Don't start by creating fancy stuff. Start simple and make sure to have an answer to each question. You can improve your answers later when you have additional information.

3.1 Data exploration

1. Explore the features and target variables of the dataset and visualize them. What are the values? What is the right performance metric to use for this dataset? Clearly explain which performance metric you choose and why. Use no more than 125 words.
2. Come up with the simplest baseline we should aim to beat. What is the minimum performance that we should expect of our learners? Use no more than 30 words.
3. Algorithmic bias can be a real problem in machine learning. Should we use the `race` and `sex` features in our algorithm? Clearly explain what you believe and provide us with your argumentation. Use no more than 75 words.

3.2 Data preparation

1. Some features have missing values that should be handled. List all features with missing values and explain how you handled them. Use no more than 100 words.
2. Implementations of the algorithms in Scikit-learn expect numerical features. Check if all features are in a numerical format. If not, transform these features into numerical ones. List all features you transformed, explain how you transformed them, and why you chose these transformations. Use no more than 75 words.
3. For which algorithms is the scale of the features important? Explain why (not) for each of the five algorithms. Bring the features to the same scale if necessary and describe your approach. Use no more than 100 words.
4. Have you done any other data pre-processing steps? If you did, explain what you did and why you did it. We suggest you consider (1) applying PCA (where the number of components is a hyper-parameter) and (2) addressing the imbalanced nature of the data. Use no more than 100 words.

3.3 Experiments

1. Divide the data into different sets (e.g. using hold-out or cross-validation) to ensure valid performance evaluation and hyper-parameter tuning. Explain your decision and steps using no more than 100 words.
2. Fit the five algorithms using the default hyper-parameters from section 2.1. Create a useful plot or table that shows the performances of the algorithms. Clearly explain what it tells us about the performances of the algorithms. Use no more than 150 words and two visual aids (but 1 is sufficient).

3. Can you explain why the worst algorithm performs the worst and why the best algorithm performs the best? Investigate if this may be due to over- or underfitting. Give your reasoning using no more than 150 words.
4. Now we will perform hyper-parameter tuning. Clearly explain what you did to be systematic, what you did to get fair results, when did you decide to stop searching for better hyper-parameters, etc. Sometimes tuning can take a long time, and you will have to make some choices to ensure that the experiment doesn't take too long; explain these choices. Use no more than 200 words.
   Note: First focus on tuning the default hyper-parameters, this should be sufficient. Only look at other hyper-parameters if time permits it.
   Hint: Should the learning rate of logistic regression be re-tuned for different $alpha$?
5. Which algorithms improved when tuned and which did not? Illustrate your answer with a clear plot or table and uncertainty estimates. Use no more than 100 words.
6. Select your best algorithm for this dataset and use it to make your predictions for the unknown samples. Please note in your report which algorithm you chose.

4 MNIST

The MNIST dataset is a large database of handwritten digits. Each row in the dataset is a 28×28 greyscale image. A feature $x_{i,j}$ represents the pixel value in the ith row and jth column. We have also provided you with a downsampled version of this dataset. In this version, all digits are 8 × 8 greyscale images. Other than that, both datasets are the same.
Note: Again, don't start by creating fancy stuff. Start simple and make sure to have an answer to each question. You can improve your answers later.

4.1 Data Exploration

1. Explore the dataset by plotting the same image from both datasets side by side. How do these images compare? Which dataset do you expect to perform better? Clearly explain why you suspect that. Use no more than 75 words.
2. What is the right performance metric to use for this dataset? Also, come up with a baseline that we should aim to beat. Explain your reasoning. Use no more than 100 words.

4.2 Data Preparation

1. Examine the features of both datasets and decide if you need to do any data cleaning or pre-processing. If not, clearly explain why not. If yes, clearly explain why and what you did. Use no more than 100 words.

4.3 Experiments

1. Divide the data into different sets (e.g. using holdout or cross-validation), to ensure a valid performance evaluation and fair tuning of your hyper-parameters. Explain what you did using no more than 100 words.
2. Fit the five algorithms using the default hyper-parameters from section 2.1. Create a useful plot or table that shows the performances of the algorithms. Clearly explain what it tells us about the performances of the algorithms. Use no more than 150 words and two visual aids (but 1 is sufficient).
3. Can you explain why the worst algorithm performs the worst and why the best algorithm performs the best? Investigate if this may be due to over- or underfitting. Give your reasoning using no more than 150 words.
4. Now perform hyper-parameter tuning on the key hyper-parameters. Clearly explain what you did and how you did this. Use no more than 200 words.
5. Which algorithms improved when tuned and which did not? Illustrate your answer with a clear plot or table and uncertainty estimates. Use no more than 200 words.
6. Compare the performance of the algorithms with the 8×8 and 28×28 features. What effect do the additional features have? Does it agree with your expectations? If you observe any differences, state what you think causes them. Use no more than 100 words.
7. Select your best algorithm for this dataset and use it to make your predictions for the unknown samples. Feel free to use either the 8×8 or the 28×28 features. Please note in your report which algorithm and feature set you chose.

5 Conclusion

1. What conclusions can we draw about the five algorithms examined in this assignment? For each algorithm briefly discuss its key behavior (for example advantage or disadvantage) that you noticed while working on the two classification tasks. Use no more than 250 words in total (± 50 words per algorithm).