CSE2510: MACHINE LEARNING
JORDI SMIT AND GOSIA MIGUT

Practical Assignment

October 1, 2020

1 INTRODUCTION

During this course, you have encountered many machine learning algorithms and concepts. Now it is time to put your knowledge to the test. In this assignment, you will explore the effectiveness of five machine learning algorithms in different scenarios. During this assignment, we expect you to examine the different algorithms critically and properly analyse their effectiveness.

1.1 ASSIGNMENT

In this assignment, you and your teammate will explore different datasets. Each dataset has its own shortcomings and problems that you will have to overcome to find the best possible classifier. During this assignment, you will explore the effectiveness of the following algorithms on these datasets:

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

For each dataset, we expect you to follow the following structure:

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 decide how hard the problem is and what the right evaluation metric is for this problem.

2. **Data preparation**: In this step, we expect you to transform the data into the expected format for the algorithm. This step might include some data cleaning, data encoding, data transformations, etc.
   Note: We don't expect you to do any fancy stuff like outlier handling etc. but it is essential that the data is in the right format and has the right distribution and/or scale.

3. **Experiments**: In this step, we expect you to do two things. First, you should fit and fine-tune the algorithms. This step might include hyper-parameter selection, grid search, cross-validation, model evaluation, etc. Secondly, you should evaluate and compare the different models. This step might consist of selecting the right evaluation metrics, visualization, etc.

During this assignment you are allowed to use the Python library Scikit-learn. The beauty of this library is that all its machine learning algorithms follow the same API. You only need to know what the fit, predict and test methods do and you can use every algorithm in the library. To help you get started with this assignment, we have provided you with a template Jupyter Notebook for each dataset. This template takes care of the data loading part and gives you some hints. We have also uploaded an optional Scikit-learn primer notebook to BrightSpace that explains Scikit-learn's most important methods.
Note: We highly encourage that you use Scikit-learn. If you decided to implement you own version of the algorithms you won't get any additional points.

1.2 Suggest reading material

During this assignment, you might encounter some definitions that might be new to you. Or you might remember them from the lectures, but they have become a bit fuzzy. Either way, if you encounter some of these terms, we expect you to research them yourself as part of this assignment. For this, you can always use the the provided reading material. However, sometimes it might be beneficial to read about the topic from another perspective. Luckily for you, the online machine learning community is also very rich, especially if you use the search query: "sklearn + CONCEPT" or "machine learning blog + CONCEPT" you will find dozens of explanations. To help you get started with some of the definitions, we have provided you with the following links with useful reading materials:

- Baseline algorithm
- Preprocessing continuous features
- Preprocessing categorical features
- Preprocessing missing values
- Classification metrics
- Hyper-parameters tuning

- Cross-validation
- Scikit-learn Glossary

1.3 Assessment

This assignment constitutes 35% of the final grade for the Machine Learning course. In this assignment, you will be graded based on both your report and the predictive power of your best algorithms. Your final grade for this assignment consists of two parts: the report counts for 85% of the grade, and the remaining 15% depends on the predictive power of your algorithms. So in this assignment, you will mainly be graded based on your analysis. So always clearly explain what you did and why you did it in a short and concise way. Also, always strictly follow the word limits specified in each question. Points may be deducted if you go over this limit. It might also happen that the reviewer stops reading when they reach this limit. The same also holds for the plot limits. The predictions of your best classifiers will also be graded based on how it compares to our worst classifier and our best classifier under the same constraints and dataset. So for example, if our classifiers achieve an accuracy of 0.69 and 0.95 respectively and your classifier achieves 0.90 you will get $(0.9-0.69)/(0.95-0.69) = 0.81$ of these points.
Note: We might use a different evaluation metric than shown in the example. The metric we use depends on the dataset. It is up to you to explore the dataset and select the right metric.
Note: While the final prediction power of your algorithm does contribute towards your grade, your analysis will weight significantly higher (see the rubric).

1.4 Deliverables

Your main deliverable for this assignment is your final report. This report should contain the answers to all the questions stated in section 2. We also want to see the code you implemented for your experiments and the predictions of your classifier per dataset. These files should be delivered in a compressed zip file with the following format:

- **Group_XX.zip**: Compressed file for submission. Should contain the following:
  - *<group>_report.pdf*: The report as **PDF** file.
  - *<group>_problem_census.ipynb*: A runnable Jupyter notebook with the source code of your experiments for the census dataset problem.
  - *<group>_classes_problem_census.txt*: The predicted classes of the unknown samples for the census dataset problem.
  - *<group>_problem_mnist.ipynb*: A runnable Jupyter notebook with the source code of your experiments for the MNIST dataset problem.
  - *<group>_classes_problem_mnist.txt*: The predicted classes of the unknown samples for the MNIST dataset problem.

The deadline for the assignment is **October 28th at 23:59**.

2 QUESTIONS

The following sections will help you to walk through the machine learning experiment process. Each section has its own questions. All these questions 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 be using the following default hyper-parameters:

- GaussianNB
  - use only sklearn's default values.
- DecisionTreeClassifier
  - $max\_depth = None$
  - $min\_samples\_leaf = 2$
  - $random\_state = 42$
- KNeighborsClassifier
  - $n\_neighbors = 3$
  - $weights = "distance"$
- SVC
  - $C = 10$
  - $kernel = "poly"$
  - $random\_state = 42$
- LogisticRegression
  - $C = 10$
  - $penalty = "none"$
  - $random\_state = 42$

Note: Leave all other hyper-parameters to the default values.
Note: Take special note of the random_state variables. If you don't set this, you won't get deterministic results.

1. (4 points) For each of the five algorithms list key strength and key weakness. Use no more than 250 words in total (+- 50 per algorithm).
2. (3 points) Carefully read the Scikit-learn hyper-parameter documentation for each of the five algorithms. Based on this documentation explain how the previously mentioned hyper-parameters effect the algorithms and their performance. Express yourself clearly and provide your reasoning. Use no more than 300 words in total (+- 75 per algorithm).
   Note: You don't have to write anything about the Naive Bayes since it has not hyper-parameters of interest.

2.2 US CENSUS

In this part of the assignment, you will explore the dataset from the United States 1994 Census. This data is stored in a tabular format and might contain some missing values. It is your job to create a binary classifier that can predict whether a person makes over $50,000 a year using this dataset. The dataset 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 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 below significantly simpler.
Note: Don't start by creating stuff that is too fancy. Start simple and make sure you have at least something, and at least some answers to all the questions. You can always come back and improve later when you have additional information.

2.2.1 DATA EXPLORATION

1. (1 point) Explore the features and target variables of the dataset. 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. (1 point) Algorithmic bias can be a real problem in Machine Learning. So based on this, should we use the *Race* and the *Sex* features in our machine learning algorithm? Clearly explain what you believe, also provide us with arguments why. Note this question will be graded based only on your argumentation. Use no more than 75 words.

2.2.2 DATA PREPARATIONS

1. (2 points) This dataset hasn't been cleaned, yet. Do this by finding all the missing values and handling them. How did you handle these missing values? Clearly explain which values were missing and how you handled them. Use no more than 100 words.
2. (2 points) All Scikit-learn's implementations of these algorithms expect numerical features. Check for all features if they are in numerical format. If not, transform these features into numerical ones. Clearly explain which features you transformed, how you transformed them and why these transformations. Use no more than 75 words. (You might want to read the preprocessing documentation of Scikit-learn for handy tips.)
3. (Bonus 2 point) Have you done any other data preprocessing steps? If you did, explain what you did and why you did it. Use no more than 100 words.

2.2.3 EXPERIMENTS

1. (1 point) Now set up your experiment. Clearly explain how you divided the data and how you ensured that your measurements are valid. Use no more than 100 words.
2. (2 points) Fit the five algorithms using the default hyper-parameters from section 2.1. Create a useful plot that shows the performances of the algorithms. Clearly explain what this plot tells us about the performances of the algorithms. Also, clearly explain why you think some algorithms perform better than others. Use no more than 150 words and two plots (but 1 is sufficient).
3. (2 points) Now perform hyper-parameter tuning on the key hyper-parameters you have previously identified. Clearly explain what you did to be systematic, what you did to get fair results, what trade-off accuracy vs resources trade-off, etc. Use no more than 200 words.
   Note: First focus on tuning the default hyper-parameters, this should be sufficient. Only look at others if time permits it.
4. (2 points) Compare the performance of the algorithms with and without hyper-parameter tuning. How did the tuning affect your result? Clearly explain the results and the differences. Use no more than 100 words and two plots (but 1 is sufficient).

5. Select your best algorithm for this dataset and use it to make your predictions for the unknown samples. Please note in you algorithm which algorithm you chose.

2.3 MNIST

The MNIST dataset is a large database of handwritten digits. Each row in the dataset is a 28×28 grey scale image. Each feature $x_{i,j}$ represents the pixel value in the ith row and jth column. We have also provided you with a downsampled versions of this dataset. In this down-sampled version, all the images are 8×8 grey scale images. Except for the downsampling, both datasets are exactly the same and have the exact same order. You see this by plotting the ith image of both datasets.
Note: Don't start by creating stuff that is too fancy. Start simple and make sure you have at least something, and at least some answers to all the questions. You can always come back and improve later when you have additional information.

2.3.1 Data Exploration

1. (1 point) 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.

2.3.2 Data Preparations

1. (3 points) Examine the features of both the datasets and decide if you need to do any data cleaning or preprocessing. If not, clearly explain why not. If yes, clearly explain why and what you did. Use no more than 100 words. (You might want to read the additional reading materials).

2.3.3 Experiments

1. (1 point) Now set up your experiment. Clearly explain how you divided the data and how you ensured a valid measurement. Use no more than 100 words.
2. (2 points) Fit the five algorithms using Scikit-learn's default hyper-parameters. Create a useful plot that shows the performances of the algorithms. Clearly explain what these plots tell us about the performances of the algorithms. Also, clearly explain why you think some algorithms perform better than others and why some of them preform better on one dataset than the other. Use no more than 200 words and two plots (but 1 is sufficient).
3. (2 points) Now perform hyper-parameter tuning on the key hyper-parameters you have previously identified. Clearly explain what you did and how you did this. Use no more than 200 words.

4. (2 points) Compare the performance of the algorithms with and without hyper-parameter tuning. Also, make a comparison with your original baseline. How did the tuning affect your result? Clearly explain the results and the differences. Use no more than 200 words and two plots (but 1 is sufficient).

5. (1 points) Compare the performance of the algorithms with the 8x8 and 28x28 features. What effect do the additional features have? Also, state what you think causes this effect. Use no more than 75 words.

6. Select your best algorithm for this dataset and use it to make your predictions for the unknown samples. Feel free to use either the 8x8 or 28x28 features. Please note in your report which algorithm and feature set you chose.

2.4 CONCLUSION

1. (3 points) Which conclusions can we draw about the five algorithms examined during this assignment? For each algorithm briefly discuss the key thing you noticed about it during this assignment. Use no more than 250 words in total (+- 50 words per algorithm).