Assignment 1

Data Mining and Machine Learning (2IIG0)

Ghada Sokar

This is your first assignment of the Data Mining and Machine learning course. The assignment is composed of seven questions, six of them are multiple-choice. The last question is open-ended. Multiple-choice questions have four possible choices; among these, only one answer is correct. The number of points for each question is written at the beginning of it. The maximum achievable score is 100 points. In all the questions except the open-ended one, you have to implement the algorithm required in the question yourself from scratch using Python, analyze the results you get, and carefully choose one of the choices. Only in the last question, we are going to use scikit-learn.

Submit the answers for the MC questions in the **HW1** quiz in Canvas. You have to submit this quiz individually. That is, every group member has to fill out his/her own quiz, even when you are encouraged to work on the exercises together. You can submit the quiz multiple times, the latest one will be graded. Please fill this quiz out carefully.

The open question can be submitted on a group level by means of the **Open Question HW1** assignment in Canvas. That is, one of your group members should upload:

- A PDF document containing the answer to the open question. Include a paragraph about the workload distribution as well. We assume that you can decide yourself how to organize the work in your group. But if someone didn't contribute at all, or only marginally, then you should note this in the PDF.
- A Python file/ notebook containing the implementations you used to answer the implementation-based questions. This notebook is your insurance, it shows what you did and which results you got. If you want to argue for a higher grade/get partial points then you can back it up with the results stated in the uploaded notebook. However, if we see that you couldn't provide a working implementation, but just guessed the correct answer in the MC format, then we will also subtract the corresponding points from your grade.

1. (10 points) Given matrix A $= \begin{pmatrix} 0.27 & -0.4 & 0.32 \\ 0.31 & 0.37 & -0.61 \\ 0.15 & 0.05 & -0.39 \end{pmatrix}$ such that $A \in \mathbb{R}^{n \times d}$ represents n observations of d features.

   Which of the following statements is **correct**?

   A. The vector of average feature values is $\mu = [0.063, 0.023, -0.063]$

   B. The rows (observations) are orthogonal. 正交

   C. The rows (observations) are orthonormal. 标准正交

   D. None of the above.

2. (10 points) Consider the function

$$f(\mathbf{x}) = x_1^4 + 4x_1 + 2x_2 + \frac{1}{2}x_2^4$$

   We want to find the minimum of f using the gradient decent optimization algorithm. Therefore:

   - Implement a function that takes a point $\mathbf{x} = (x_1, x_2)$ and returns the the partial derivatives of f at this point.
   - Perform gradient descent update using the function implemented in the previous step.
   - Choose a stopping criteria to stop the iterations of gradient update.

   Which of the following statements is **correct**?

   A. Gradient descent converges to the minimum in less than 10 iterations using stepsize $\eta = 1$ and a starting point $\mathbf{x}_0 = (1, 1)$.

   B. Gradient descent converges to the minimum in less than 20 iterations using stepsize $\eta = 0.1$ and a starting point $\mathbf{x}_0 = (2, 1)$.

   C. With a starting point $\mathbf{x}_0 = (5, 5)$ and stepsize $\eta = 1$, the algorithm converges faster than starting from point $\mathbf{x}_0 = (1, 1)$ and stepsize $\eta = 0.01$.

   D. The stepsize $\eta = 0.1$ is too large for the model to converge.

3. (10 points) In the Assignment .zip file, you can find a small dataset divided into two *.csv* files: *data/diabetes_train_data.csv* and *data/diabetes_validate_data.csv*. This dataset is originally from the National Institute of Diabetes and Digestive and Kidney Diseases. The objective is to predict based on diagnostic measurements whether a patient has diabetes. The first two columns represents the glucose concentration and the blood pressure of a patient respectively. The last column represents whether a patient has diabetes ($Outcome = 1$) or not ($Outcome = 0$). Suppose that we wish to predict whether a new patient will has diabetes or not. We will take this decision using k-nearest neighbor model with Euclidean distance metric. To this end:

   - Import the data in Python.
   - Plot the data to explore it.
   - Train a k-nearest-neighbors classifier using $k = 1$ on the training data. Calculate the classification accuracy on the validation data.
   - Repeat the previous step using $k = 2$, $k = 3$ and $k = 4$.

   Which of the following statements is **correct**?

   A. All the patients in the validation set that do not have diabetes are correctly classified by the KNN model with $k = 3$.

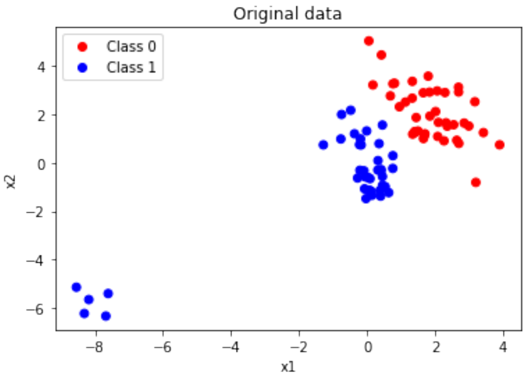
   B. Using $k = 3$ achieves higher validation accuracy than $k = 2$.

   C. Using $k = 4$ achieves higher validation accuracy than $k = 2$.

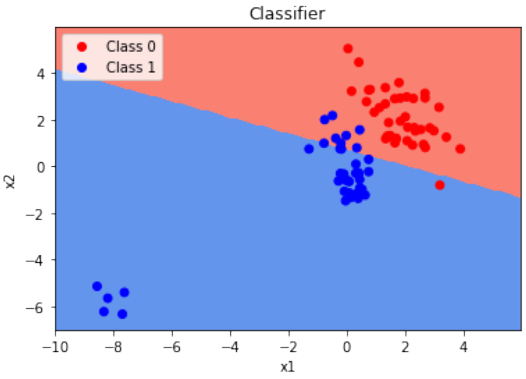
   D. The classifier achieves 70.07% on the validation set using $k = 1$.

4. (10 points) Consider the two-classes dataset shown in the left Figure (a). The data has two features (x_1, x_2). Suppose that we train a certain linear classifier and get the following the decision boundary: $y = -0.078x_1 - 0.227x_2 + 0.165$

The decision boundary and predictions of the training data are also shown in the right Figure (b) as background colors.



(a)



(b)

We would like to evaluate the performance of that classifier on the following validation data. Therefore:

- Calculate the confusion matrix for the given classifier on the following validation data.

ID	x_1	x_2	y
1	2	1.5	0
2	-2	0	1
3	3	2.5	0
4	3	3.5	0
5	-2	3	1
6	2	3	0
7	-2	1	1
8	-1.5	1	1

From the figures above and your evaluation, which of the following statements is **correct**?

A. The training data is not linearly separable.

B. This classifier results from training linear SVM on the training data.

C. 75% of the samples of class 0 in the validation data is correctly classified.

D. 25% of the samples of class 1 in the validation data is misclassified as class 0.

In the following two exercises, you are going two work with *Heart Disease* dataset. You can find this dataset in the Assignment .zip file splitted into two $.csvfiles$: $data/heart\_train\_data.csv$ and $data/heart\_validate\_data.csv$. This dataset is taken from the UCI Machine Learning repository. However, this is a modified version in which a subset of the original features are only considered. We consider three features. The description of the features and the target is as follows:

1. cp (Chest Pain Type): [0: Typical Angina, 1: Atypical Angina, 2: Non-Anginal Pain, 3: Asymptomatic]
2. exang (Exercise Induced Angina): [1 = yes, 0 = no]
3. thal (Thallium heart scan): [1 = normal, 2 = fixed defect, 3 = reversible defect]
4. target: [0 = disease, 1 = no disease]

5. (20 points) We would like to build a Naive Bayes classifier to predict whether a new patient has a heart disease or not. To this end:

- Import the data in Python and explore it a bit.

- Estimate the class probability $p(y)$.
- Estimate the conditional probability $p(x_d|y)$ which is the probability of feature x_d given class y.
- Implement a function that takes an input and return the corresponding prediction based on the estimations calculated in the previous two steps.

Based on your implementation, which of the following statements is **correct**?

A. A patient that has asymptomatic chest pain and exercise induced angina but normal thallium heart scan will be diagnosed as not having a heart disease by the Naive Bayes classifier.

B. 48% of the people in the training data has atypical angina.

C. The probability of having a normal thallium heart scan given that the patient has heart disease equals to 0.075.

D. The Naive Bayes reaches 80% accuracy on the validation set.

6. (20 points) Now using the same dataset, we would like to diagnose the patients using decision tree classifier. Therefore:

- Use the CART algorithm explained in the lecture to implement the decision tree.
- For the cost: use entropy as impurity measure.
- Stopping criterion: stop at tree depth 3 (8 leaves).
- During the CART algorithm, consider all One-vs-All decisions when learning decision nodes.

1. cp (Chest Pain Type): [0: Typical Angina, 1: Atypical Angina, 2: Non-Anginal Pain, 3: Asymptomatic]

2. exang (Exercise Induced Angina): [1 = yes, 0 = no]

4. target: [0 = disease, 1 = no disease]

3. thal (Thallium heart scan): [1 = normal, 2 = fixed defect, 3 = reversible defect]

Based on your implementation, which of the following statements is **correct**?

A. The decision tree has higher accuracy on the validation set than the Naive Bayes classifier.

B. The decision of the root of the tree is based on whether thal is a reversible defect or not (thal == 3).

C. A patient that has typical angina, no exercise induced angina, and a reversible defect thallium heart scan will be diagnosed as having a heart disease by the decision tree classifier.

cp = 0, exang = 0, thal = 3

D. Only one leaf node of the tree have impurity equals zero.

7. (20 points) **Open Question**. In this exercise, we will explore SVM kernels on a toy dataset using scikit-learn. In Assignment .zip file, you can find *data/toy_dataset.csv* and Python notebook (SVM.ipynb) with a startup code. We would like to study the behavior of each kernel and check underfitting and overfitting. Therefore, import the data in Python and explore it. Then:

(A) (10 points) Compare different kernels

(i) Fit SVM with linear, polynomial and RBF kernels with default parameter values. Use SVC function in the svm module of scikit-learn.

(ii) Plot the decision boundary for each kernel with the helper function given in the notebook.

(iii) Interpret the plots and compare the behavior of the three kernels.

(B) (10 points) Now, we will optimize the two hyperparameters $'C'$ and gamma $'\gamma'$ of the $'rbf'$ kernel using with GridSearchCV from the model_selection module of scikit-learn. Check the documentation of GridSearchCV.

(i) Create a grid with the following values:
$'gamma'$: [1e-4, 1e-3, 1e-2, 1e-1, 1, 2],
$'C'$: [1e-2, 1e-1, 1, 2, 5, 10]

(ii) Use GridSearchCV with SVC(kernel='rbf') as classifier, and 3-fold-cross-validation (cv) [1].

[1] In k-fold-cross validation, the data is splitted into k parts (folds). We perform the training procedure k times. Each time, we take one fold as the validation set and the remaining $k-1$ folds as the training set. We fit the model on the training set and compute the validation score on the chosen fold (part). After computing k validation scores, we take the average.

(iii) Plot a heatmap of the results (average validation scores) using the provided helper function.

(iv) Interpret the heatmap. Analyze the effect of different values of hyperparameters. Does any combination of C and γ leads to underfitting or overfitting?

(v) Report the accuracy of the best model you get. State the hyperparameters used.