Data Science I

Overall Points: / 100

Exercise 7: Classification & Clustering

<u>Submission Deadline: January 26 2026, 07:00 UTC</u>
University of Oldenburg
Winter 2025/2026
Instructors: Jannik Schröder, Wolfram "Wolle" Wingerath

Submitted by: <your names here>

Part 1: Logistic Regression & Gradient Descent

/ 25

1.) Suppose we are training a model using stochastic gradient descent. How do we know if we are converging to a solution?

Solution:

< your solution here >

2.) Do gradient descent methods always converge to the same point? Please explain your reasoning.

Solution:

< your solution here >

3.) Consider the following labeled data points:

In [25…

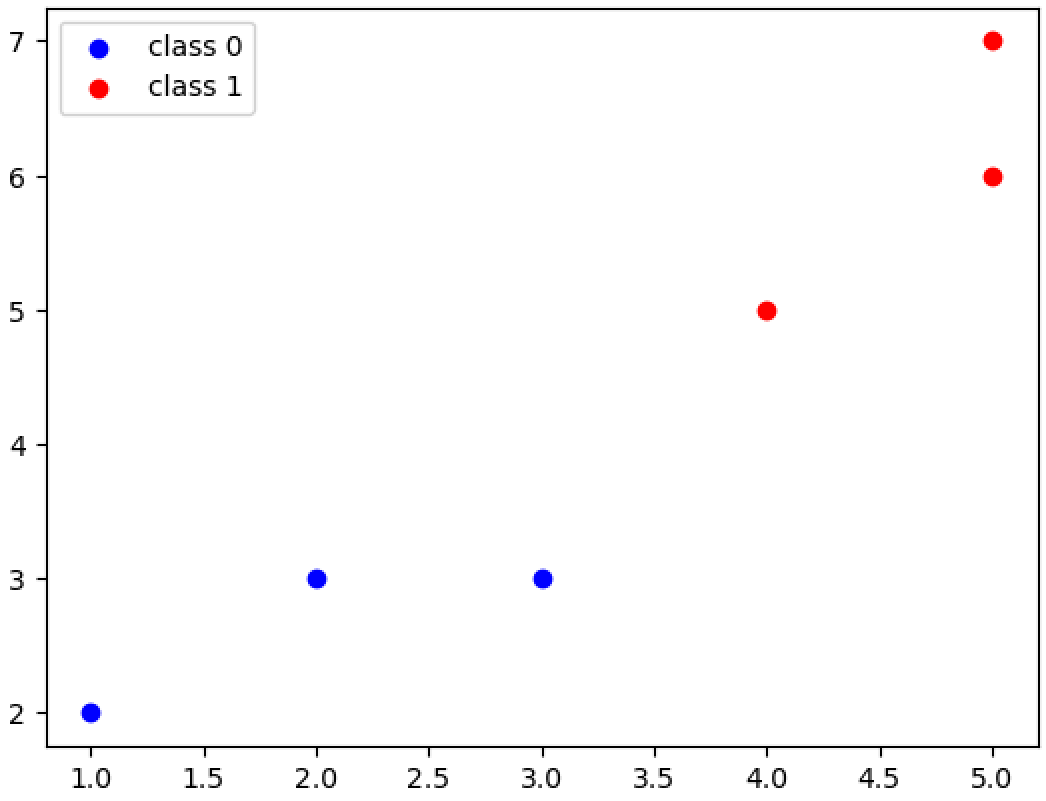
```python
import matplotlib.pyplot as plt

# The data points in 2D feature space
data_points = [(1, 2), (2, 3), (3, 3), (4, 5), (5, 6), (5, 7)]

# The labels corresponding to the data points
```

```
labels = [0, 0, 0, 1, 1, 1]
class_0 = [point for point, label in zip(data_points, labels) if label == (
class_1 = [point for point, label in zip(data_points, labels) if label == 1

# Plot the data points
plt.scatter(*zip(*class_0), color='blue', label='class 0')
plt.scatter(*zip(*class_1), color='red', label='class 1')
plt.legend(loc='upper left')
plt.show()
```



a) Please train a logistic regression model using `sklearn.linear_model` and draw the decision boundary.

Solution:

< your solution here >

b) Please explain how you got from the model parameters to the equation for the separating line.

Solution:

< your solution here >

Part 2: Support Vector Machines & Classification

/ 25

4.) Consider the following set of data points in 2D space where the labels are stored in the variable y and the feature vectors are stored in the variable X:

In [25…

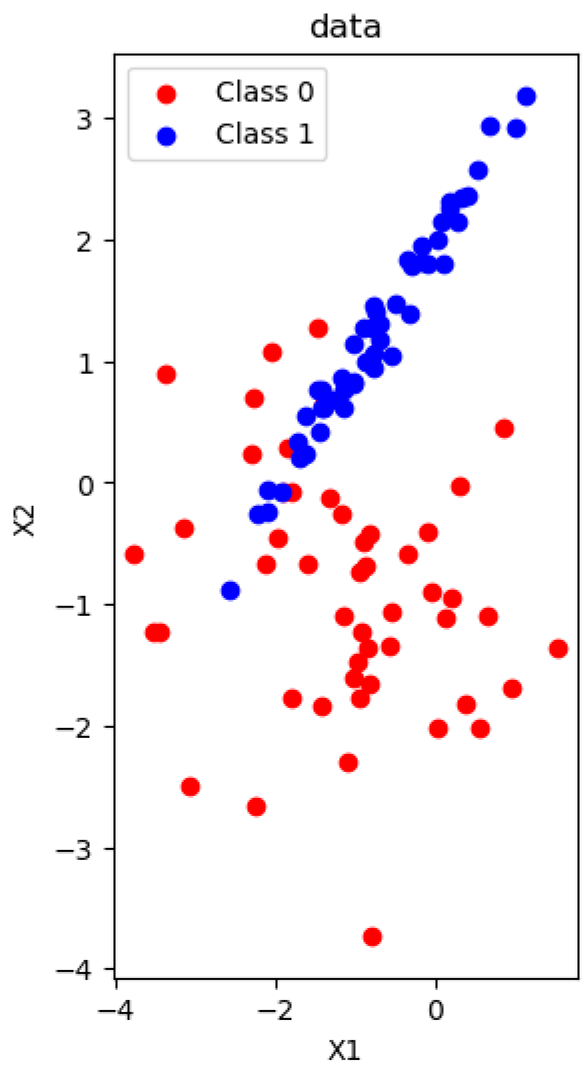
```
import numpy as np
from sklearn import datasets

X, y = datasets.make_classification(n_samples=100, n_features=2, n_informat


# Separate data points by class for the sake of plotting
X0 = [X[i] for i in range(len(X)) if y[i]==0]
X1 = [X[i] for i in range(len(X)) if y[i]==1]

# Create scatter plot
plt.figure(figsize=(3,6))
plt.scatter([x[0] for x in X0], [x[1] for x in X0], color='red', label='Cla
plt.scatter([x[0] for x in X1], [x[1] for x in X1], color='blue', label='C

# Set the labels and title
plt.xlabel('X1')
plt.ylabel('X2')
plt.title('data')
plt.legend()
plt.show()
```



a) Please split the dataset into a training set and a test set. You can use `sklearn.model_selection.train_test_split` for this purpose.

Solution:

< your solution here >

b) For at least two different kernel functions (e.g. 'linear', 'rbf', 'poly', or 'sigmoid'), do the following:

1. **Please train a Support Vector Machine classifier on the training set. Use the SVC class from `sklearn.svm` for the task.**
2. **Please evaluate the classifier on the test set using accuracy as the metric. You can use the `sklearn.metrics.accuracy_score` to achieve this.**

3. **Now please visualize the decision boundary of both classifiers. You can create a meshgrid of points in the feature space, predict the label of each point with the trained classifier, and then use a contour plot to visualize the decision boundary.**

Solution:

< your solution here >

Part 3: Distance Metrics & Nearest-Neighbor Methods

/ 25

5.) Is the edit distance on text strings a metric? Please provide an answer and explain your reasoning.

Solution:

< your solution here >

6.) Construct a two-class point set on n ≥ 10 points in two dimensions, where every point would be misclassified according to its nearest neighbor (kNN classification with k = 1).

Solution:

< your solution here >

7.) How does classification performance change when you classify by the 3 nearest neighbors (kNN classification with k = 3)?

Solution:

< your solution here >

Part 4: Networks & Clustering

/ 25

8.) For each of the following graph-theoretic properties, please provide a use case or an example of a real-world network that satisfies / does not satisfy the property.

a) Directed vs. undirected.

Solution:

< your solution here >

b) Weighted vs. unweighted.

Solution:

< your solution here >

c) Simple vs. non-simple.

Solution:

< your solution here >

d) Sparse vs. dense.

Solution:

< your solution here >

e) Embedded vs. topological.

Solution:

< your solution here >

f) Labeled vs. unlabeled.

Solution:

< your solution here >

9.) Again, consider the following set of data points in 2D space:

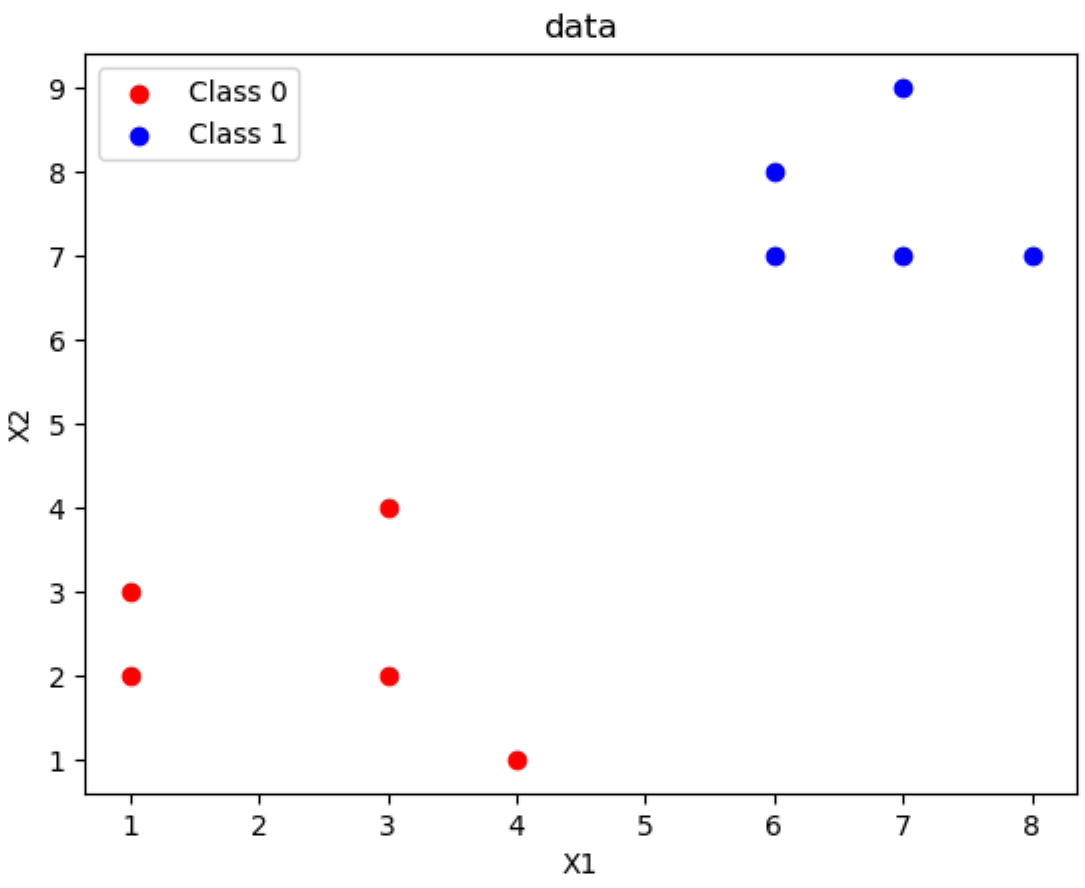
In [25…

```
X_knn = np.array([[1, 2], [3, 4], [1, 3], [4, 1], [3, 2], [6, 8], [8, 7],
y_knn = np.array([0, 0, 0, 0, 0, 1, 1, 1, 1, 1])

# Separate data points by class for the sake of plotting
X_knn_0 = [X_knn[i] for i in range(len(X_knn)) if y_knn[i]==0]
X_knn_1 = [X_knn[i] for i in range(len(X_knn)) if y_knn[i]==1]

# Create scatter plot
plt.figure()
plt.scatter([x[0] for x in X_knn_0], [x[1] for x in X_knn_0], color='red',
plt.scatter([x[0] for x in X_knn_1], [x[1] for x in X_knn_1], color='blue'

# Set the labels and title
plt.xlabel('X1')
plt.ylabel('X2')
plt.title('data')
plt.legend()
plt.show()
```



Please perform a KNN classification of the new point p = [4, 4] for k=3 using `sklearn.neighbors.KNeighborsClassifier`. Use the Euclidean distance as a metric. (Note: Please plot the new point p in the same plot with the other points from the dataset, but color it differently from the points in the dataset.)

Solution:

< your solution here >

Finally: Submission

Save your notebook and submit it (as both **notebook and PDF file**). And please don't forget to ...

- ... choose a **file name** according to convention (see Exercise Sheet 1, but please **add your group name as a suffix** like `_group01`) and to
- ... include the **execution output** in your submission!