Begonnen am	Montag, 18. Dezember 2023, 13:42
Status	Beendet
Beendet am	Donnerstag, 21. Dezember 2023, 14:53
Verbrauchte Zeit	3 Tage 1 Stunde
Bewertung	**28,50** von 30,00 (**95**%)

Frage 1

Richtig

Erreichte Punkte 1,00 von 1,00

Which statements about unsupervised learning are true?

Wählen Sie eine oder mehrere Antworten:

- [x] It does not need labels for training ✓
- [x] Its goal is to find patterns in the data. ✓
- [ ] Its goal is to learn a mapping from input data to output data.
- [ ] It needs labels for training

Your answer is correct.

Die richtigen Antworten sind: Its goal is to find patterns in the data., It does not need labels for training

Frage 2

Richtig

Erreichte Punkte 1,00 von 1,00

Which of the following methods solve a **supervised** learning problem?

Wählen Sie eine oder mehrere Antworten:

- [x] Ordinary Least Squares Regression ✓
- [ ] Non-Negative Matrix Factorization
- [x] (Kernel) Ridge Regression ✓
- [ ] K-Means Clustering
- [x] Perceptron ✓
- [x] Nearest Centroid Classifier ✓
- [ ] Principal Component Analysis
- [x] Linear Discriminant Analysis ✓

Your answer is correct.

Die richtigen Antworten sind: Linear Discriminant Analysis, Nearest Centroid Classifier, Perceptron, Ordinary Least Squares Regression, (Kernel) Ridge Regression

Frage 3

Richtig

Erreichte Punkte 1,00 von 1,00

Cross-Validation can be used to ...

Wählen Sie eine oder mehrere Antworten:

- [ ] ... detect outliers in the data
- [ ] ... train a model that will be deployed
- [x] ... estimate the generalization error ✓
- [x] ... find optimal parameter values ✓

Your answer is correct.

Die richtigen Antworten sind: ... estimate the generalization error, ... find optimal parameter values

Frage 4

Richtig

Erreichte Punkte 1,00 von 1,00

Which statements are true? Nested Cross-Validation ...

Wählen Sie eine oder mehrere Antworten:

- [x] a. ... performs multiple cross-validations. ✓
- [ ] b. ... can be **only** used to estimate the generalization error, but is more accurate than usual cross validation.
- [x] c. ... can be used for parameter tuning while it also estimates the generalization error. ✗
- [ ] d. ... is the same as cross-validation.

Your answer is correct.

Die richtige Antwort ist: ... performs multiple cross-validations.

Frage 5

Richtig

Erreichte Punkte 1,00 von 1,00

Which statement about **P**rincipal **C**omponent Analysis is **false**?

Wählen Sie eine Antwort:

- ○ PCA finds the direction that maximizes the variance of the projected data
- ◉ The first k **PC**s are the eigenvectors corresponding to the ~~smallest~~ biggest k eigenvalues ✓
- ○ The data mapped onto the first k principal components is uncorrelated.

Your answer is correct.

Die richtige Antwort ist: The first k **PC**s are the eigenvectors corresponding to the smallest k eigenvalues

Frage 6

Richtig

Erreichte Punkte 1,00 von 1,00

Tick the correct statements regarding Principal Component Analysis (PCA). 对 outlier 敏感.

Wählen Sie eine oder mehrere Antworten:

- [ ] The first principal component corresponds to the direction of ~~least~~ biggest variance
- [x] 不同 Distinct principal components must be orthogonal ✓
- [ ] Negative eigenvalues indicate negative correlation eigenvalue不可为负
- [x] The covariance matrix is symmetric and positive semi-definite ✓

Your answer is correct.

Die richtigen Antworten sind: Distinct principal components must be orthogonal, The covariance matrix is symmetric and positive semi-definite

Frage 7

Richtig

Erreichte Punkte 1,00 von 1,00

Linear Kernel PCA is cheaper to compute than standard PCA if, ...

Wählen Sie eine Antwort:

- ( ) ... the data in non-negative
- ( ) ... the number of samples is larger than the number of features
- ( ) ... the noise in the data follows a Gaussian distribution
- (•) ... the number of features is larger than the number of samples ✓

Your answer is correct.

Die richtige Antwort ist: ... the number of features is larger than the number of samples

- If there are more dimensions than samples ($n \ll d$)
  Compute PCA on linear kernel matrix $X^\top X \in \mathbb{R}^{n \times n}$
- If there are more samples than dimensions ($d \ll n$)
  Compute PCA on covariance matrix $XX^\top \in \mathbb{R}^{d \times d}$

Frage 8

Richtig

Erreichte Punkte 1,00 von 1,00

Nonnegative Matrix Factorization ...

PCA / NMF 都用于 reduce dimensionality

Wählen Sie eine oder mehrere Antworten:

- [x] ... is an iterative method ✓
- [x] ... learns a lower dimensional embedding ✓
- [x] ... yields interpretable results for non-negative valued data ✓
- [ ] ... maps the input to a higher dimensional feature space

Your answer is correct.

Die richtigen Antworten sind: ... learns a lower dimensional embedding, ... is an iterative method, ... yields interpretable results for non-negative valued data

Frage **9**

Richtig

Erreichte Punkte 8,00 von 8,00

Insert your implementation of the function pca() from the notebook. Make sure to use inputs and returns of the function just as described in the notebook.

Antwort: (Abzugssystem: 0 %)

Antwort zurücksetzen

```python
def pca(X, ncomp=10):
    ''' Principal Component Analysis
    INPUT:  X       - dxn array of n data points with d features
            ncomp   - number of principal components to estimate
    OUTPUT: W       - d x ncomp array of directions of maximal variance,
                     sorted by their eigenvalues
            H       - ncomp x n array of projected data '''

    ncomp = min(np.hstack((X.shape, ncomp)))
    #  ... your code here ...

    # center the data
    x_means = np.mean(X,axis = 1)
    X = X - x_means.reshape(-1,1)
    #print(x_means,np.hstack((X.shape, ncomp)),X_centered)
    # compute linear kernel
    K = X.T @ X
    # compute eigenvectors and sort them according to their eigenvalues
    eigenvalues, eigenvectors = np.linalg.eig(K)
    #print(eigenvalues, eigenvectors)

    idx = np.argsort(-eigenvalues)
    eigenvalues = eigenvalues[idx]
    eigenvectors = eigenvectors[:,idx]
    #print(idx,eigenvalues.shape, eigenvectors.shape,X.shape)

    # select k largest eigenvalues
    W = X @ eigenvectors[:,:ncomp]
    H = W.T @ X
    # compute W and H
    return W, H
```

	Test	Erwartet	Erhalten	

✓	`import numpy as np`
	`# create random rotated data`
	`np.random.seed(1)`
	`X = np.array([[1,2],[0,2]]) @ np.random.randn(2,300)`
	`# apply liner kernel pca`
	`W, H = pca(X,2)`
	`# normalize eigenvectors (non normalized because of kernel context)`
	`W = W / np.diag(W.T @ W)[None, :]`
	`# project back to original space`
	`R = W @ H`
	`# compare to X normalized`
	`Xn = X - X.mean(axis=1, keepdims=True)`
	`print(np.allclose(R, Xn))`
		True	True	✓

	Test	Erwartet	Erhalten	

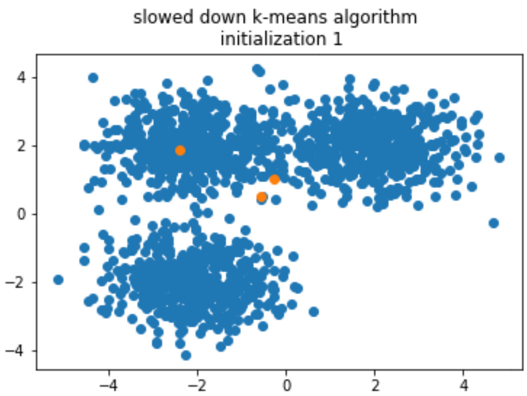
✓	`import numpy as np`
	`# create random rotated data`
	`np.random.seed(1)`
	`X = np.random.rand(20,20) @ np.random.rand(20,300)`
	`# apply liner kernel pca`
	`W, H = pca(X,20)`
	`# normalize eigenvectors (non normalized because of kernel context)`
	`W = W / np.diag(W.T @ W)[None, :]`
	`# project back to original space`
	`R = W @ H`
	`# compare to X normalized`
	`Xn = X - X.mean(axis=1, keepdims=True)`
	`print(np.allclose(R, Xn))`
		True	True	✓

Alle Tests bestanden! ✓

Richtig

Bewertung für diese Einreichung: 8,00/8,00.

Information



Frage 10

Richtig

Erreichte Punkte 1,00 von 1,00

Initialize the centroids to the mean of X and separate them by adding standard normal noise to it (as described in Task 2. A in the notebook).

Antwort: (Abzugssystem: 0 %)

Antwort zurücksetzen

```
import numpy as np

np.random.seed(1)

d, n, k = 2, 500, 3
X = np.random.randn(d, n)

#centroids = # compute the matrix of centroids (d x k)

idx = np.random.choice(n,k)

centroids = X[:, idx]
```

	Test	Eingabe	
✓	round(centroids[0][0],4)	-0.0998	✓
✓	round(centroids[1][2],4)	-0.8472	✓

Alle Tests bestanden! ✓

Richtig

Bewertung für diese Einreichung: 1,00/1,00.

Frage 11

Richtig

Erreichte Punkte 3,00 von 3,00

For step 1 of the k means algorithm, we need the distance between each data point x_i and each centroid μ_j.

Complete the function distmat that calculates a matrix $Dist \in \mathbb{R}^{n \times m}$.

Antwort: (Abzugssystem: 0 %)

Antwort zurücksetzen

```
import numpy as np

def distmat(X, Y):
    """ Distance Matrix
    INPUT:      X           - dxn array of N data points with d features
                Y           - dxm array of M data points with d features
    OUTPUT:     distmat     - nxm array s.t. D[i, j] = || x_i - y_j ||^2
    Hint: np.tile might be helpful
    """

    d, n = np.shape(X)
    d_y, m = np.shape(Y)
    assert d == d_y

    # calculate the distance matrix
    # ... your code here ...
    #dist = np.zeros(1)
    dist = np.zeros((n,m))
    for i in range(n):
        for j in range(m):
            dist[i,j] = np.sum((X[:,i] - Y[:,j])**2)



    return dist
```

	Test	Erwartet	Erhalten	
✓	np.random.seed(3) X = np.random.randn(10,20) Y = np.random.randn(10,5) print(distmat(X, Y))	[[19.05459673 49.80046971 12.04469063 31.60051139 26.15388087] [17.30121641 11.45715976 9.32182249 8.89565008 13.71033844] [14.04142679 24.89194967 13.21157321 13.38536058 21.66263494] [23.0439199 50.90861883 23.9635248 38.63976426 47.52835569] [32.45822809 37.88427093 23.25191876 27.58444617 18.49459745] [18.49029931 29.77753166 16.00435534 14.02314878 16.66977269] [17.46542632 32.31299977 18.29267833 21.82045356 21.66789513] [16.73717553 16.37335167 6.15982101 16.85732793 11.36520831] [20.20753085 54.48197801 17.69450879 45.04736398 34.75789074] [17.4530542 25.01821764 12.61102795 19.86847094 24.19008106] [20.65668009 18.41018333 13.27256928 16.58148216 18.00918868] [21.26637574 25.16938127 30.8026436 20.97310663 20.87062217] [22.45077568 25.42815504 13.74633768 16.13634877 9.95600904] [20.4624784 29.09407173 16.56423721 9.77490254 24.10762538] [17.45192654 15.31948003 9.9598384 9.32034937 13.25615985] [28.09105835 52.24253558 23.01621664 30.92847166 29.64954333] [14.04362076 37.16868139 10.91208124 11.04538385 26.05270115] [11.12093489 40.33424146 9.11996721 27.99672981 17.36761056] [19.26314221 20.66795688 10.25572774 18.91013105 16.56425754] [27.7111169 50.0081247 25.77038572 27.67096015 42.84422069]]	[[19.05459673 49.80046971 12.04469063 31.60051139 26.15388087] [17.30121641 11.45715976 9.32182249 8.89565008 13.71033844] [14.04142679 24.89194967 13.21157321 13.38536058 21.66263494] [23.0439199 50.90861883 23.9635248 38.63976426 47.52835569] [32.45822809 37.88427093 23.25191876 27.58444617 18.49459745] [18.49029931 29.77753166 16.00435534 14.02314878 16.66977269] [17.46542632 32.31299977 18.29267833 21.82045356 21.66789513] [16.73717553 16.37335167 6.15982101 16.85732793 11.36520831] [20.20753085 54.48197801 17.69450879 45.04736398 34.75789074] [17.4530542 25.01821764 12.61102795 19.86847094 24.19008106] [20.65668009 18.41018333 13.27256928 16.58148216 18.00918868] [21.26637574 25.16938127 30.8026436 20.97310663 20.87062217] [22.45077568 25.42815504 13.74633768 16.13634877 9.95600904] [20.4624784 29.09407173 16.56423721 9.77490254 24.10762538] [17.45192654 15.31948003 9.9598384 9.32034937 13.25615985] [28.09105835 52.24253558 23.01621664 30.92847166 29.64954333] [14.04362076 37.16868139 10.91208124 11.04538385 26.05270115] [11.12093489 40.33424146 9.11996721 27.99672981 17.36761056] [19.26314221 20.66795688 10.25572774 18.91013105 16.56425754] [27.7111169 50.0081247 25.77038572 27.67096015 42.84422069]]	✓

Alle Tests bestanden! ✓

Richtig

Bewertung für diese Einreichung: 3,00/3,00.

Frage 12

Richtig

Erreichte Punkte 4,00 von 4,00

Assign each data point to its closest centroid as indicated by the distances measured in the matrix "dist".

Antwort: (Abzugssystem: 0 %)

Antwort zurücksetzen

```
import numpy as np

def get_closest(dist):
    """ Distance Matrix
    INPUT:      dist       - nxm array of distance between data point n and m
    OUTPUT:     closest    - nxk array that indicates for each of the N data points
                             in X the closest centroid in current iteration.
                             Each row in closest only holds one non-zero entry.
                             closest[i, j] == 1 <=>
                             centroids[:, j] is closest to data point X[:, i]
    """

    n, k = dist.shape
    closest = np.zeros((n, k), dtype='bool')

    # compute the matrix indicating the closest centroid
    # ... your code here ...

    #closest = np.zeros(1)
    #closest = np.zeros(1)
    max_by_row = np.min(dist,axis = 1)[:,None]
    closest = max_by_row == dist
    return closest
```

<table>
<tr><th></th><th>Test</th><th>Erwartet</th><th>Erhalten</th><th></th></tr>
<tr><td>✓</td><td><pre>np.random.seed(3)
dist = np.abs(np.random.randn(30,3))
print(get_closest(dist))</pre></td><td><pre>[[False False True]
 [False True False]
 [False False True]
 [True False False]
 [False False True]
 [True False False]
 [True False False]
 [True False False]
 [False False True]
 [True False False]
 [True False False]
 [True False False]
 [True False False]
 [True False False]
 [True False False]
 [False False True]
 [False True False]
 [True False False]
 [False False True]
 [False True False]
 [False False True]
 [False False True]
 [True False False]
 [False False True]
 [True False False]
 [False False True]
 [False False True]
 [True False False]
 [False False True]
 [True False False]]</pre></td><td><pre>[[False False True]
 [False True False]
 [False False True]
 [True False False]
 [False False True]
 [True False False]
 [True False False]
 [True False False]
 [False False True]
 [True False False]
 [True False False]
 [True False False]
 [True False False]
 [True False False]
 [True False False]
 [False False True]
 [False True False]
 [True False False]
 [False False True]
 [False True False]
 [False False True]
 [False False True]
 [True False False]
 [False False True]
 [True False False]
 [False False True]
 [False False True]
 [True False False]
 [False False True]
 [True False False]]</pre></td><td>✓</td></tr>
</table>

Alle Tests bestanden! ✓

Richtig

Bewertung für diese Einreichung: 4,00/4,00.

Frage 13
Richtig
Erreichte Punkte 4,00 von 4,00

Update each cluster center to the mean of the members in that cluster

Antwort: (Abzugssystem: 0 %)

Antwort zurücksetzen

```
import numpy as np

def update_centroids(X, closest):
    """
    INPUT:  X           - dxn array of N data points with D features
            closest     - nxk array that indicates for each of the N data points
                          in X the closest centroid after convergence.
                          Each row in closest only holds one non-zero entry.
                          closest[i, j] == 1 <=>
                          centroids[:, j] is closest to data point X[:, i]
    OUTPUT: centroids   - dxk array of k centroids with d features
    """

    # ... your code here ...

    #centroids = np.zeros(1)
    d,n = X.shape
    n1,k = closest.shape
    centroids = np.zeros((d,k))

    for i in range(k):
        k_cluster = X[:,closest[:,i]]
        centroids[:,i] = np.mean(k_cluster,axis = 1)


    return centroids
```

	Test	Erwartet	Erhalten	

✓	`np.random.seed(3)`
	`exp = np.array([[-0.02054159, -0.00991774, -0.01217853], [0.03886024, -0.06677293, -0.04202068]])`
	
	`X = np.random.randn(2,1500)`
	`closest = np.zeros([1500,3],'bool')`
	`idx = np.random.randint(3,size=[1500])`
	`closest[np.arange(idx.size),idx] = True`
	
	`r = update_centroids(X, closest)`
	`print(np.isclose(exp,r))`
		`[[ True  True True]`
		` [ True  True True]]`
			`[[ True  True True]`
			` [ True  True True]]`
				✓

Alle Tests bestanden! ✓

Richtig

Bewertung für diese Einreichung: 4,00/4,00.

Information

Suppose we only have two data points, $\begin{bmatrix} 0 \\ 1 \end{bmatrix}$ and $\begin{bmatrix} 0 \\ 2 \end{bmatrix}$, $X = \begin{bmatrix} 0 & 0 \\ 1 & 2 \end{bmatrix}$. $\begin{bmatrix} 0.5 \\ 1 \end{bmatrix}$

In the following, you will have to answer the following questions:

- What would be the principal directions $W = [\mathbf{w}_1, \mathbf{w}_2]$?
- What will be the variance of the projected data onto each of the principal components $\mathrm{Var}(\mathbf{w}_1^\top X)$, $\mathrm{Var}(\mathbf{w}_2^\top X)$?
- What is H?

Frage 14

Falsch

Erreichte Punkte 0,00 von 0,50

What are the principal components/eigenvectors?

$\mathbf{w}_1 = ?$ and $\mathbf{w}_2 = ?$

$X = \begin{bmatrix} 0 & 0 \\ 1 & 2 \end{bmatrix}$ $\bar{X} = \begin{bmatrix} 0 \\ 3/2 \end{bmatrix}$

$\hat{\Sigma} = \frac{1}{2} \begin{bmatrix} 0 & 0 \\ -\frac{1}{2} & \frac{1}{2} \end{bmatrix} \begin{bmatrix} 0 & -\frac{1}{2} \\ 0 & \frac{1}{2} \end{bmatrix}$

$= \frac{1}{2} \cdot \begin{bmatrix} 0 & 0 \\ 0 & \frac{1}{2} \end{bmatrix} = \begin{bmatrix} 0 & 0 \\ 0 & \frac{1}{4} \end{bmatrix}$

$\lambda = \frac{1}{4}, 0$

$\det(\hat{\Sigma} - \lambda I) = \det \begin{bmatrix} -\lambda & 0 \\ 0 & \frac{1}{4} - \lambda \end{bmatrix} = -\frac{1}{4}\lambda + \lambda^2 = 0$

$\lambda = \frac{1}{4}, 0$

$\hat{\Sigma} v = \lambda v$

$\begin{bmatrix} 0 & 0 \\ 0 & \frac{1}{4} \end{bmatrix} \begin{bmatrix} v_1 \\ v_2 \end{bmatrix} = 0 = \begin{bmatrix} 0 \\ \frac{1}{4} v_2 \end{bmatrix}$

$v_1 = 1$ $v_2 = 0$

$v_2 = \begin{bmatrix} 1 \\ 0 \end{bmatrix}$

$\begin{bmatrix} 0 & 0 \\ 0 & \frac{1}{4} \end{bmatrix} \begin{bmatrix} v_1 \\ v_2 \end{bmatrix} = \frac{1}{4} \begin{bmatrix} v_1 \\ v_2 \end{bmatrix} = \begin{bmatrix} 0 \\ \frac{1}{4} v_2 \end{bmatrix}$

$v_1 = \begin{bmatrix} 0 \\ 1 \end{bmatrix}$

Wählen Sie eine oder mehrere Antworten:

- [ ] $\begin{bmatrix} 3 \\ 2 \end{bmatrix}$ $\begin{bmatrix} 0 & \frac{1}{4} \end{bmatrix} \begin{bmatrix} 0 & 0 \\ 1 & 2 \end{bmatrix}$
- [x] $\begin{bmatrix} 1 \\ 1 \end{bmatrix}$ ✗
- [ ] $\begin{bmatrix} 1 \\ 0 \end{bmatrix}$
- [ ] $\begin{bmatrix} -2 \\ 0 \end{bmatrix}$
- [ ] $\begin{bmatrix} 1 \\ 2 \end{bmatrix}$
- [ ] $\begin{bmatrix} 0 \\ 0 \end{bmatrix}$
- [x] $\begin{bmatrix} 0 \\ 1 \end{bmatrix}$ ✓

Your answer is incorrect.

Die richtigen Antworten sind: $\begin{bmatrix} 1 \\ 0 \end{bmatrix}$

, $\begin{bmatrix} 0 \\ 1 \end{bmatrix}$

Frage 15

Falsch

Erreichte Punkte 0,00 von 0,50

What is the variance of the projection to the first eigenvector?

Wählen Sie eine Antwort:

- $\mathrm{Var}(\mathbf{w}_1^\top X) = 0.5$ ✗
- $\mathrm{Var}(\mathbf{w}_1^\top X) = 0.25$
- $\mathrm{Var}(\mathbf{w}_1^\top X) = 5$

Your answer is incorrect.

Die richtige Antwort ist: $\mathrm{Var}(\mathbf{w}_1^\top X) = 0.25$

Frage 16

Falsch

Erreichte Punkte 0,00 von 0,50

The variance of the data projected to the second eigenvector is

Wählen Sie eine Antwort:

- $\mathrm{Var}(\mathbf{w}_2^\top X) = 0$
- $\mathrm{Var}(\mathbf{w}_2^\top X) = 0.5$ ✗
- $\mathrm{Var}(\mathbf{w}_2^\top X) = -1$

Your answer is incorrect.

Die richtige Antwort ist:
$\mathrm{Var}(\mathbf{w}_2^\top X) = 0$

Frage 17

Richtig

Erreichte Punkte 0,50 von 0,50

What is the data projected to the first **PC**?

Wählen Sie eine Antwort:

$$\begin{bmatrix} 0 & 1 \end{bmatrix} \begin{bmatrix} 0 & 0 \\ 1 & 2 \end{bmatrix} = \begin{bmatrix} 1 & 2 \end{bmatrix}$$

- $\begin{bmatrix} 2 & 1 \end{bmatrix}$
- $\begin{bmatrix} 0 & 2 \end{bmatrix}$
- $\begin{bmatrix} 1 & 0 \end{bmatrix}$
- $\begin{bmatrix} 1 & 2 \end{bmatrix}$ ✓

Your answer is correct.

Die richtige Antwort ist: $\begin{bmatrix} 1 & 2 \end{bmatrix}$