Digit recognition

Final Presentation

Presented by Daria Morkis, Alex Kohlmann and Karolina Walach

Supervisor: Dr. Leonid Kostrykin, PD Dr. Karl Rohr

Tutor: Hannah Winter

What does our code include?

1. PCA
2. Self-implemented KNN
3. `KDTree` and `KNeighborsClassifier`
4. confusion matrix and classification report
5. SVM
6. CNN

Used Libraries

TensorFlow

scikit learn

NumPy

pandas

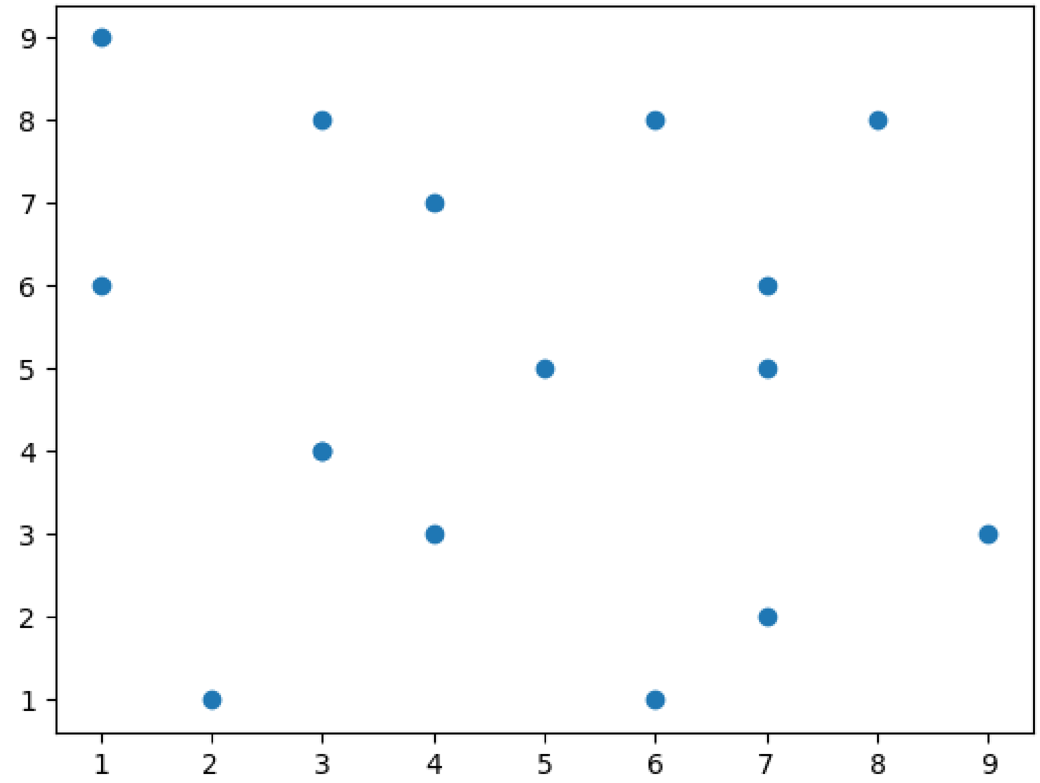
SciPy

matplotlib

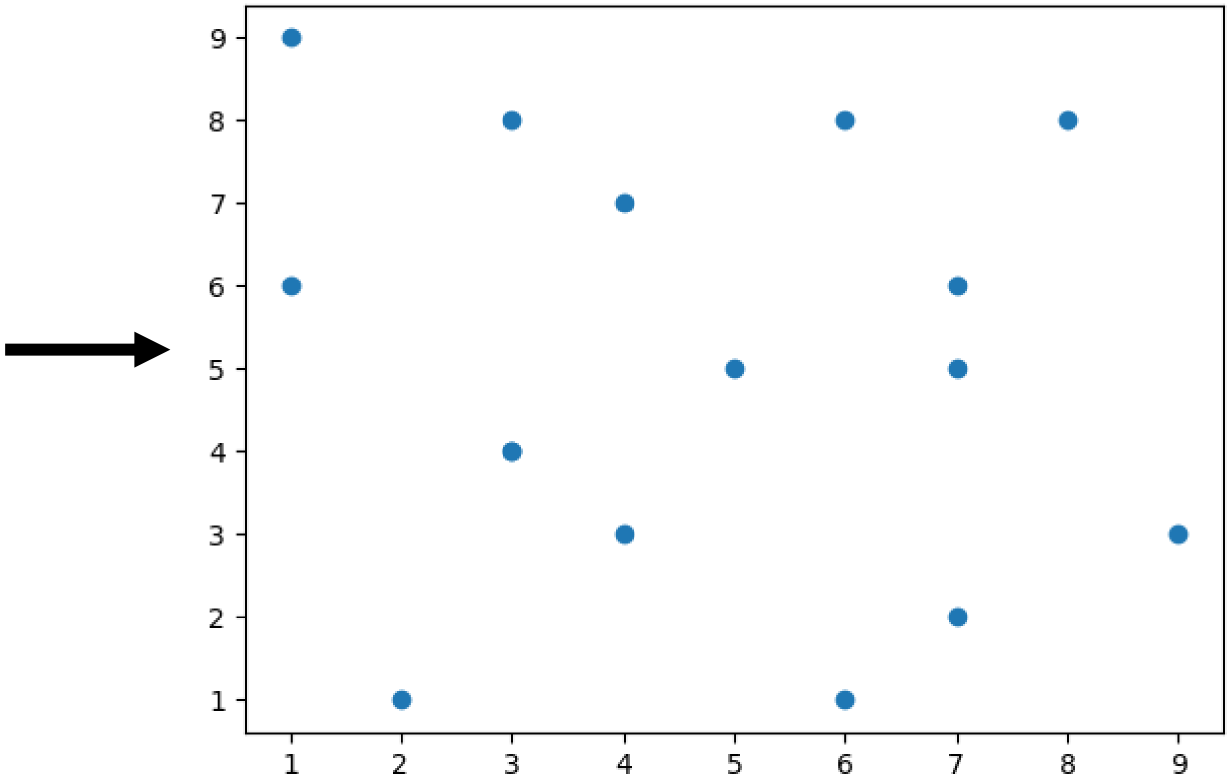
KD Tree

Fast way to calculate accuracy

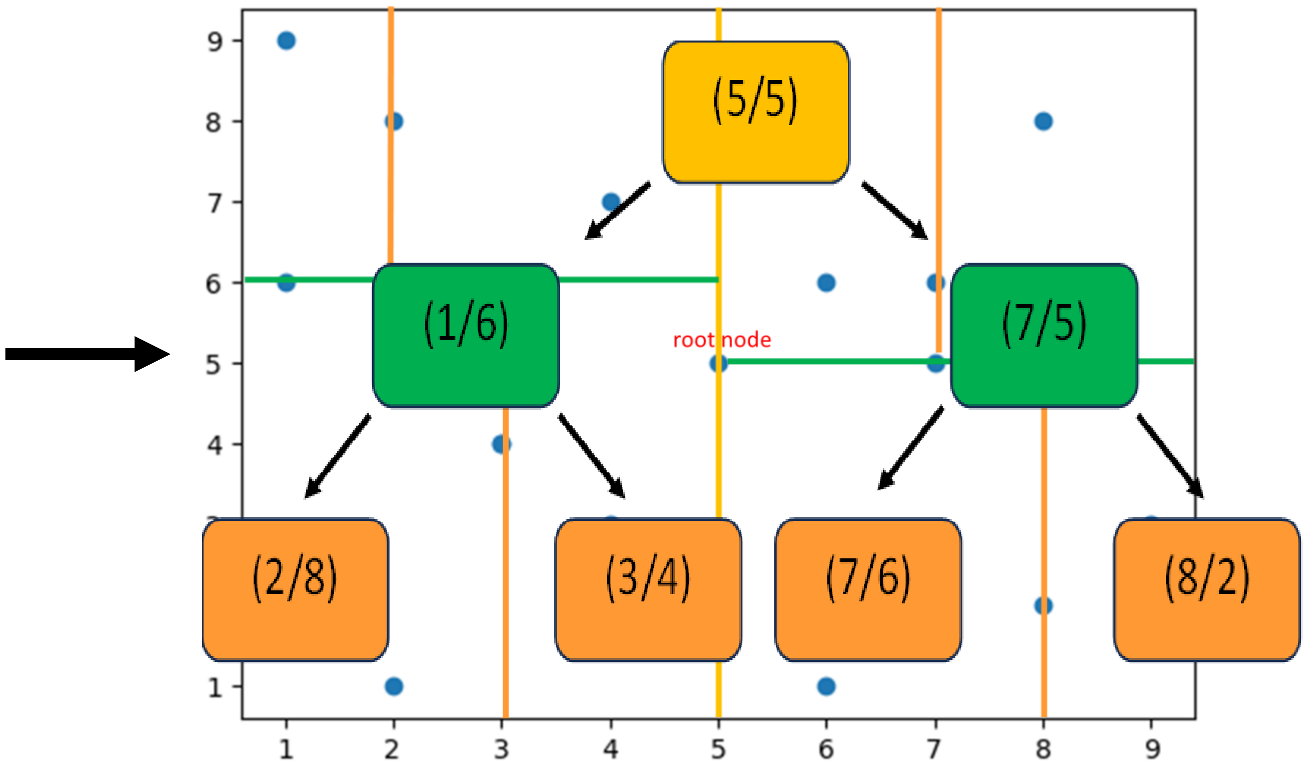
→narrows down the area where the nearest neighbor is searched



KD Tree



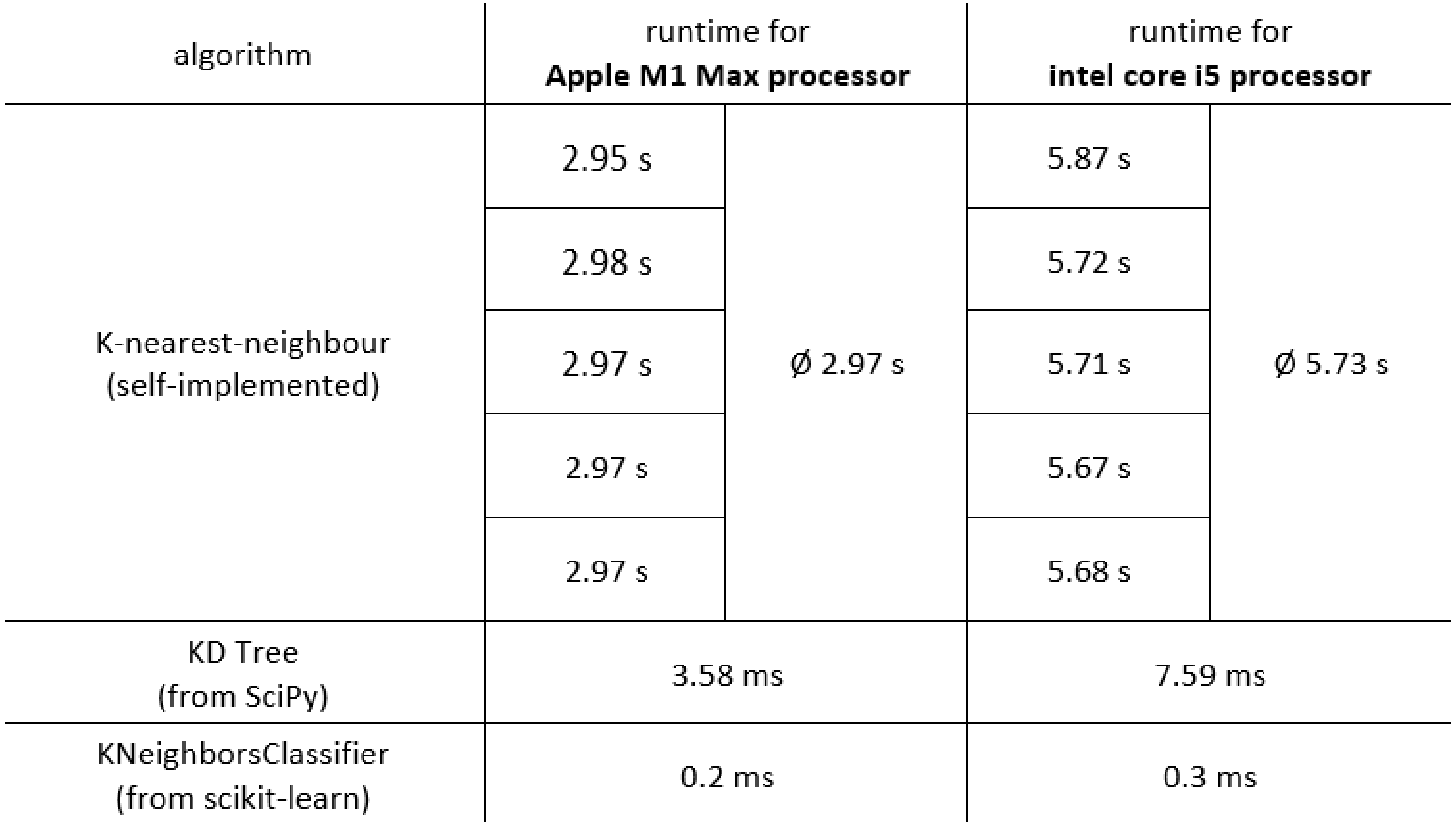
KD Tree



KNeighborsClassifier

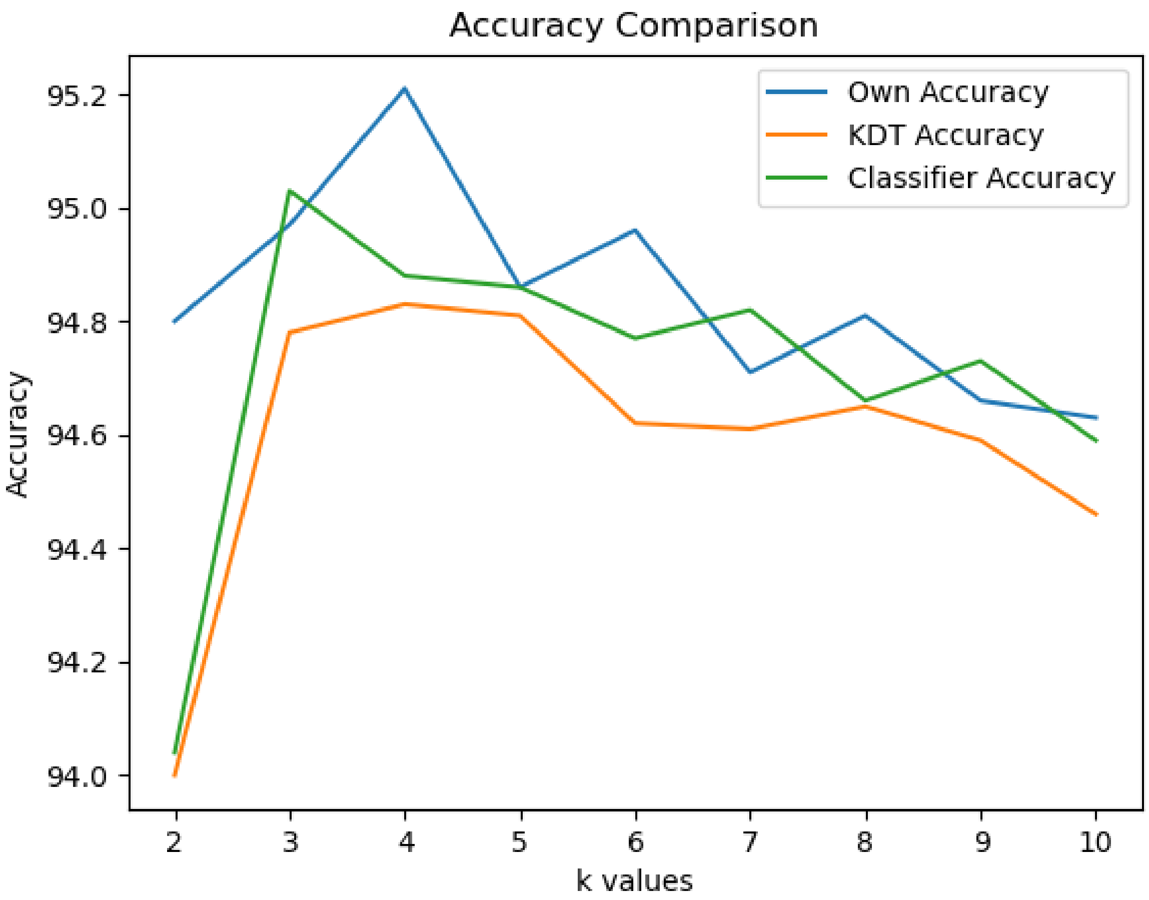
```
sklearn.neighbors.KNeighborsClassifier(n_neighbors=5,
weights='uniform', algorithm='auto', leaf_size=30, p=2, metric='minkowski'
, metric_params=None, n_jobs=None)
```

Runtime of algorithms



algorithm	runtime for **Apple M1 Max processor**		runtime for **intel core i5 processor**	
K-nearest-neighbour (self-implemented)	2.95 s	Ø 2.97 s	5.87 s	Ø 5.73 s
	2.98 s		5.72 s	
	2.97 s		5.71 s	
	2.97 s		5.67 s	
	2.97 s		5.68 s	
KD Tree (from SciPy)	3.58 ms		7.59 ms	
KNeighborsClassifier (from scikit-learn)	0.2 ms		0.3 ms	

KNN vs. KD-Tree vs. KNeighborsClassifier



Best k-value at pc=330

KNN:	k=4 with 95.21%
KD-Tree:	k=4 with 94.83%
Classifier:	k=3 with 95.03%

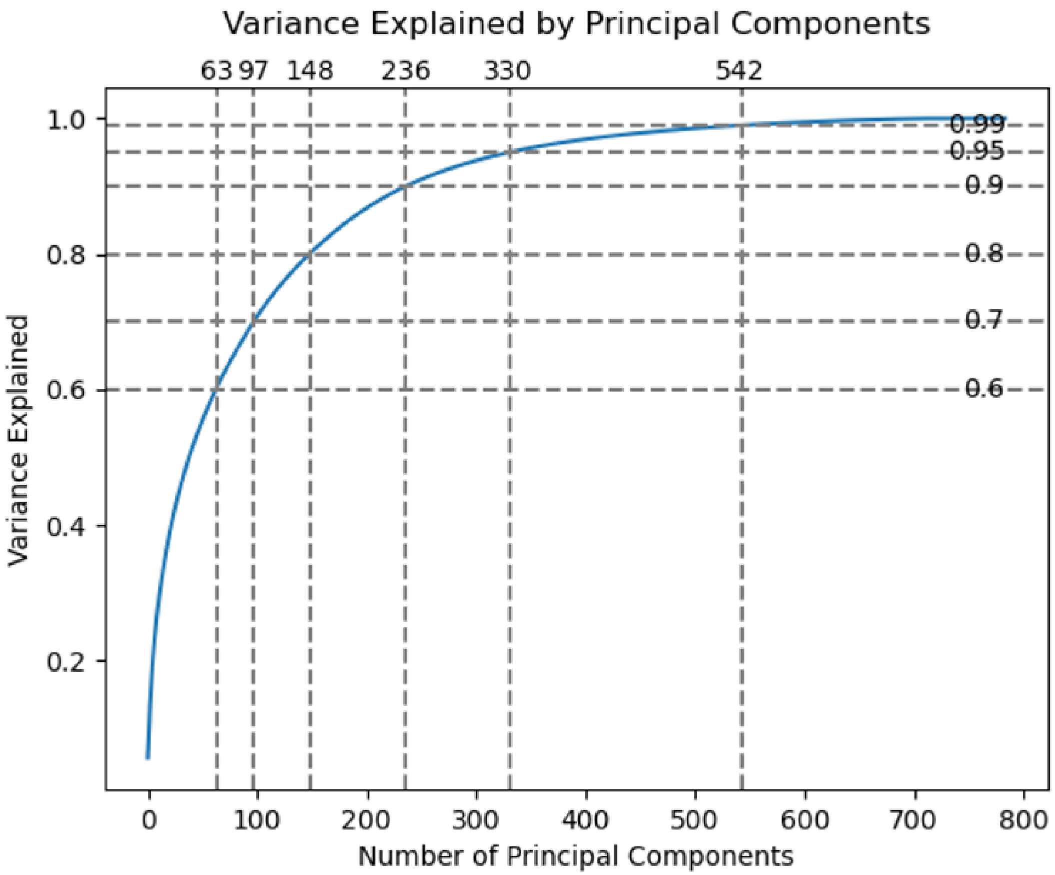
Why is k relatively small?
→ Differences of the first euclidean distances are small in ascending order

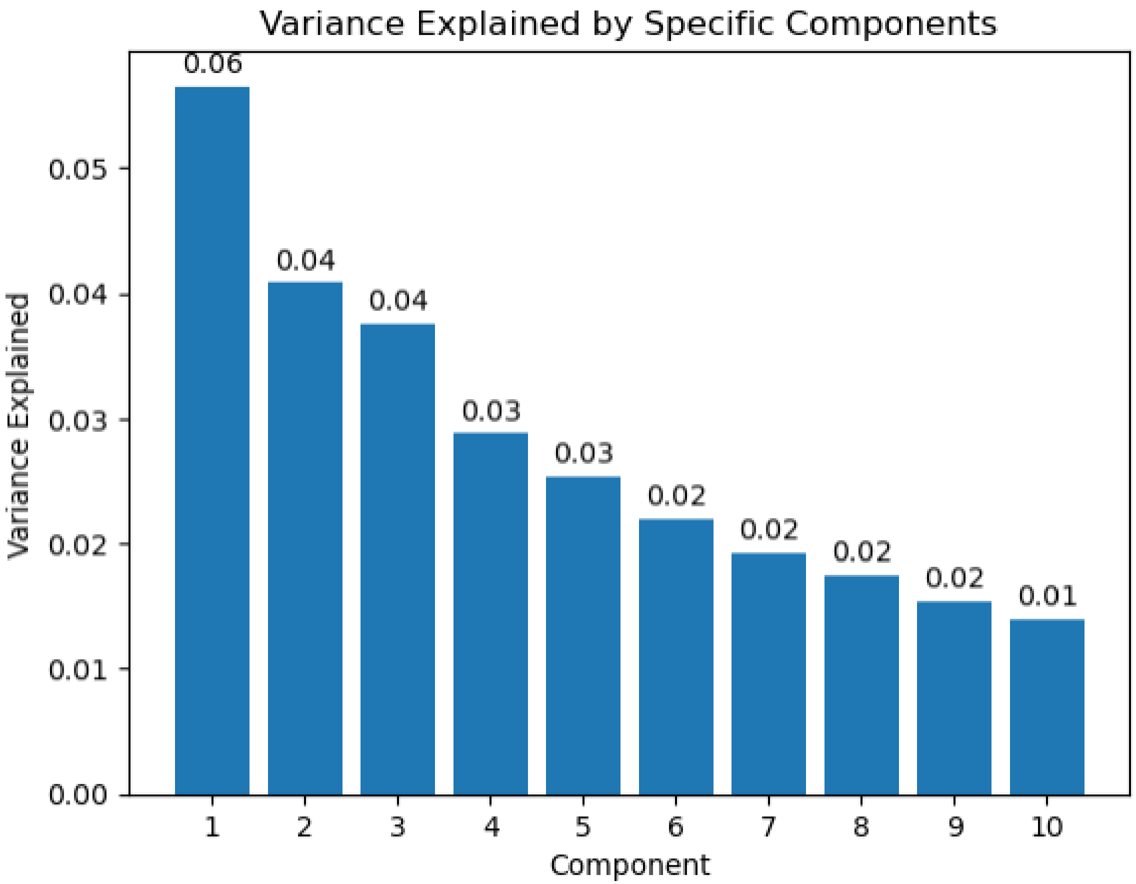
Why are the accuracies different?

Possible explanations:

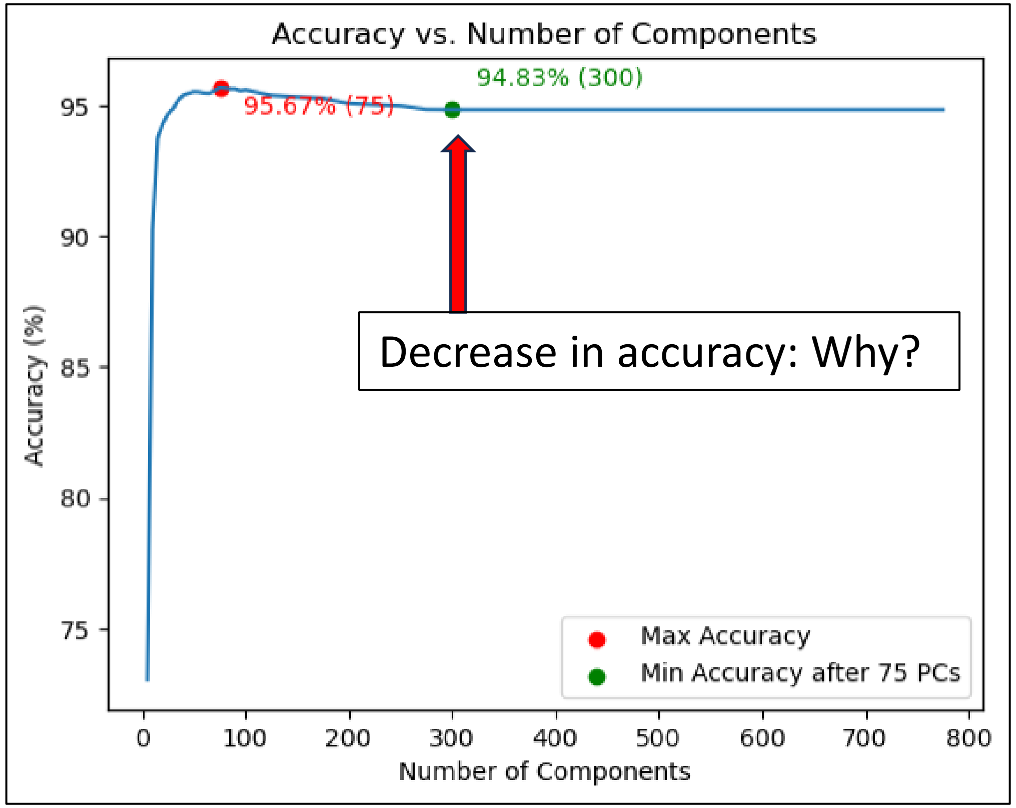
1. Different rounding of euclidean distances
2. Different selection of nearest neighbor
3. Different selection of most common label (when k even)

Principal components

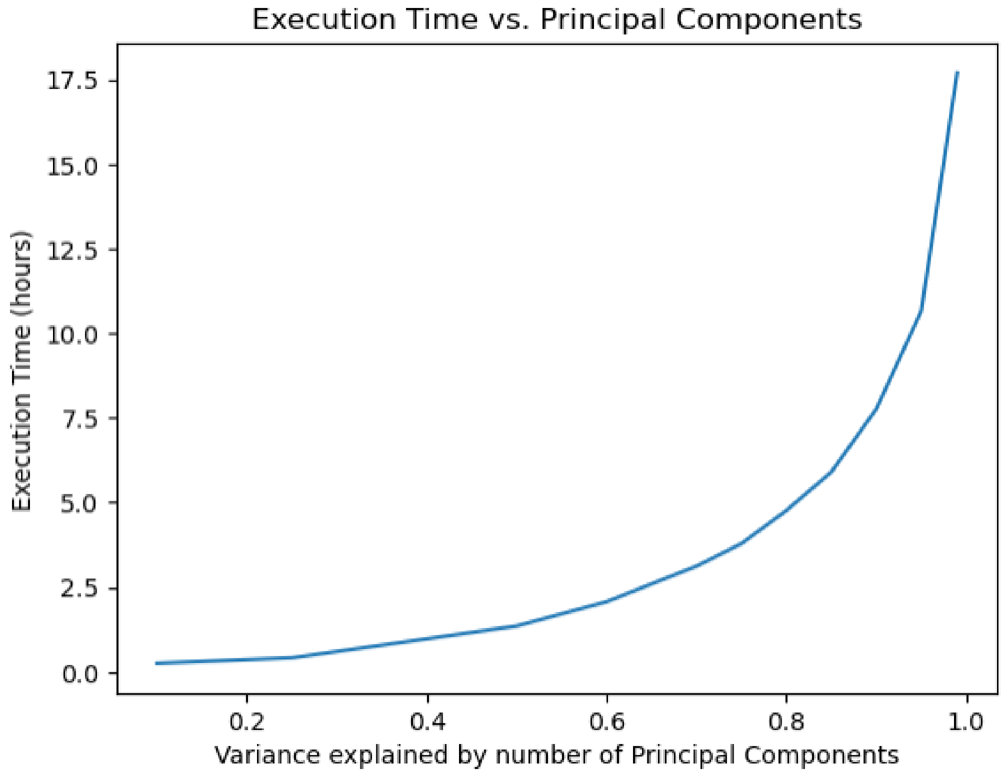




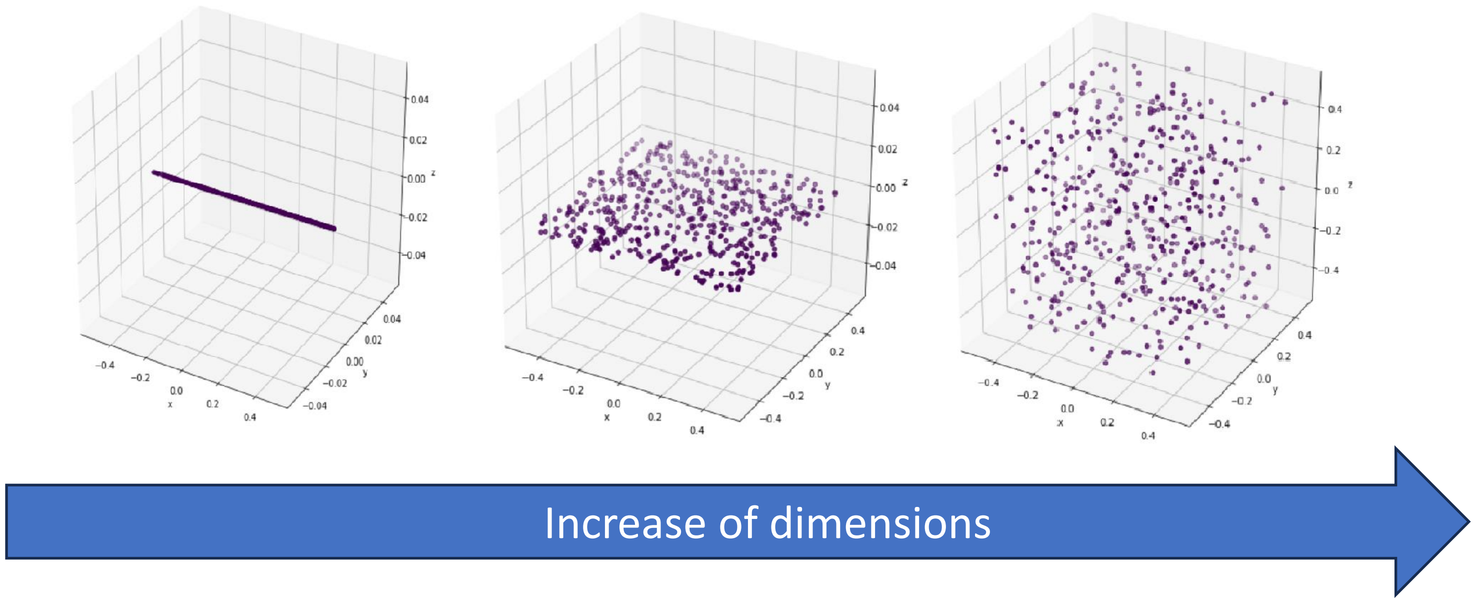
Execution time and accuracies for different numbers of PCs



Decrease in accuracy: Why?



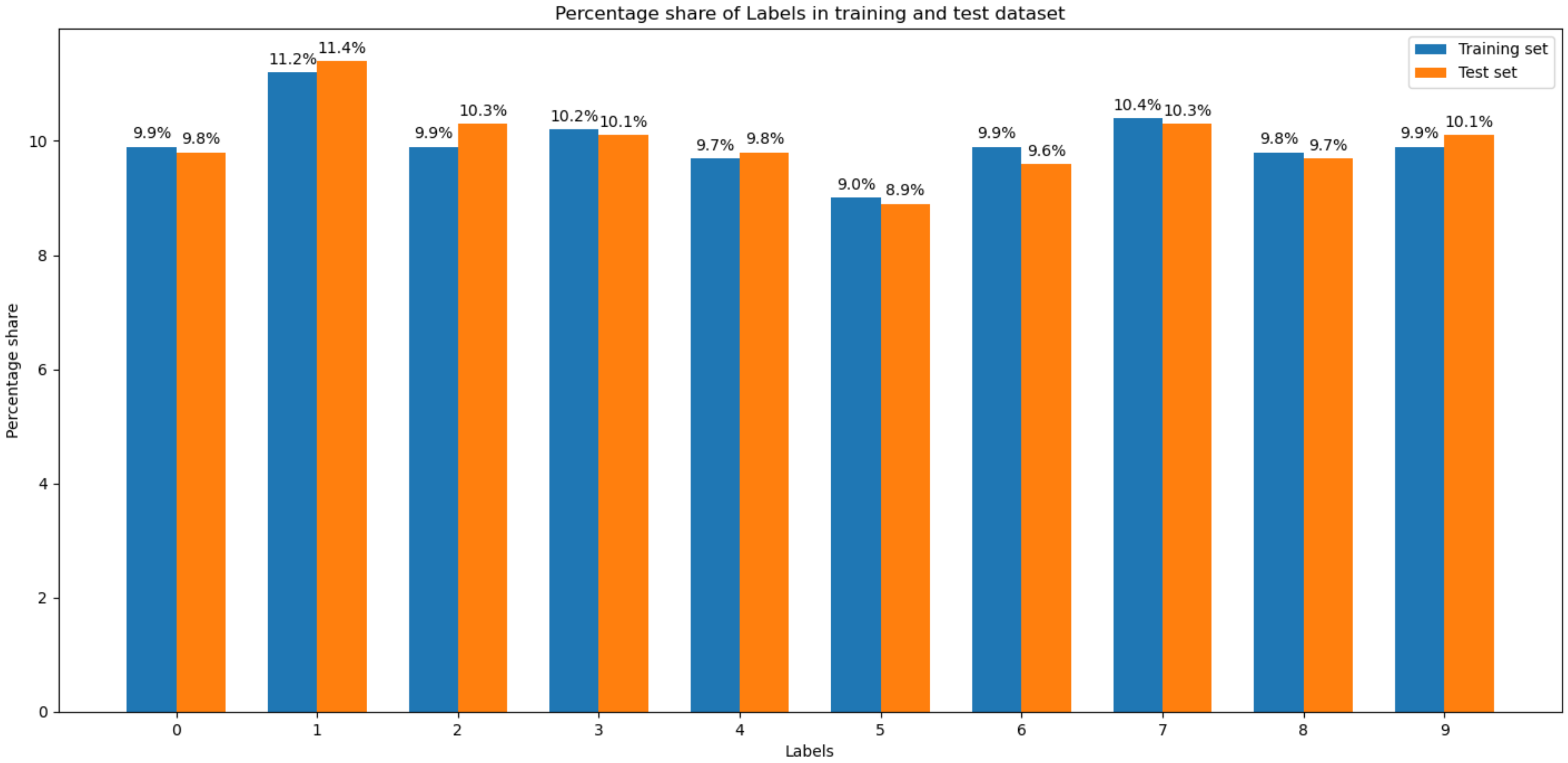
Curse of dimensionality



Increase of dimensions

- More PCs -> more information
- More possibilities to differ from one another
- increase of distance between datapoints
- Finding k-nearest neighbors becomes more difficult

Error Analysis – balanced or imbalanced?



Error Analysis – classification report

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN}$$

$$Precision = \frac{TP}{TP + FP}$$

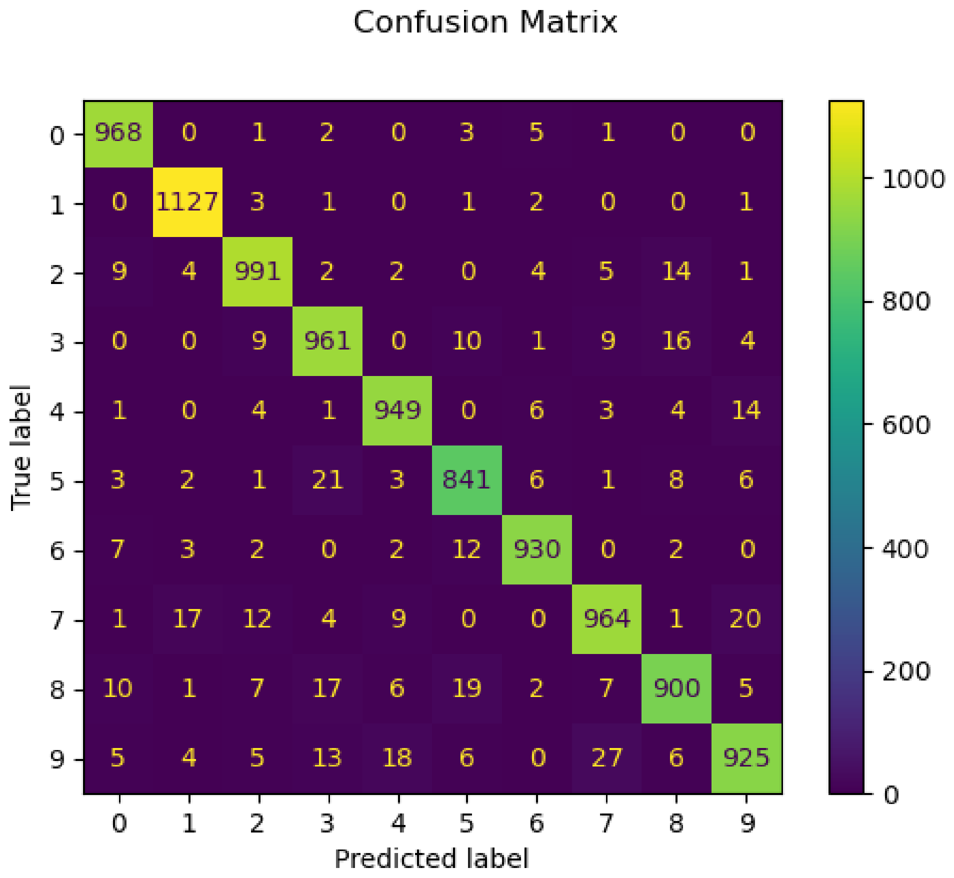
$$Recall = \frac{TP}{TP + FN}$$

$$F1 = 2 \times \frac{Precision \times Recall}{Precision + Recall}$$

Error Analysis – classification report

digit	precision	recall	f1-score	support
0	0.96	0.99	0.98	980
1	0.97	0.99	0.99	1135
2	0.96	0.96	0.96	1032
3	0.94	0.96	0.95	1010
4	0.96	0.96	0.96	982
5	0.94	0.95	0.95	892
6	0.97	0.97	0.97	958
7	0.95	0.94	0.95	1028
8	0.95	0.94	0.94	974
9	0.95	0.93	0.93	1009
accuracy			0.96	10000
macro avg	0.96	0.96	0.96	10000
weighted avg	0.96	0.96	0.96	10000

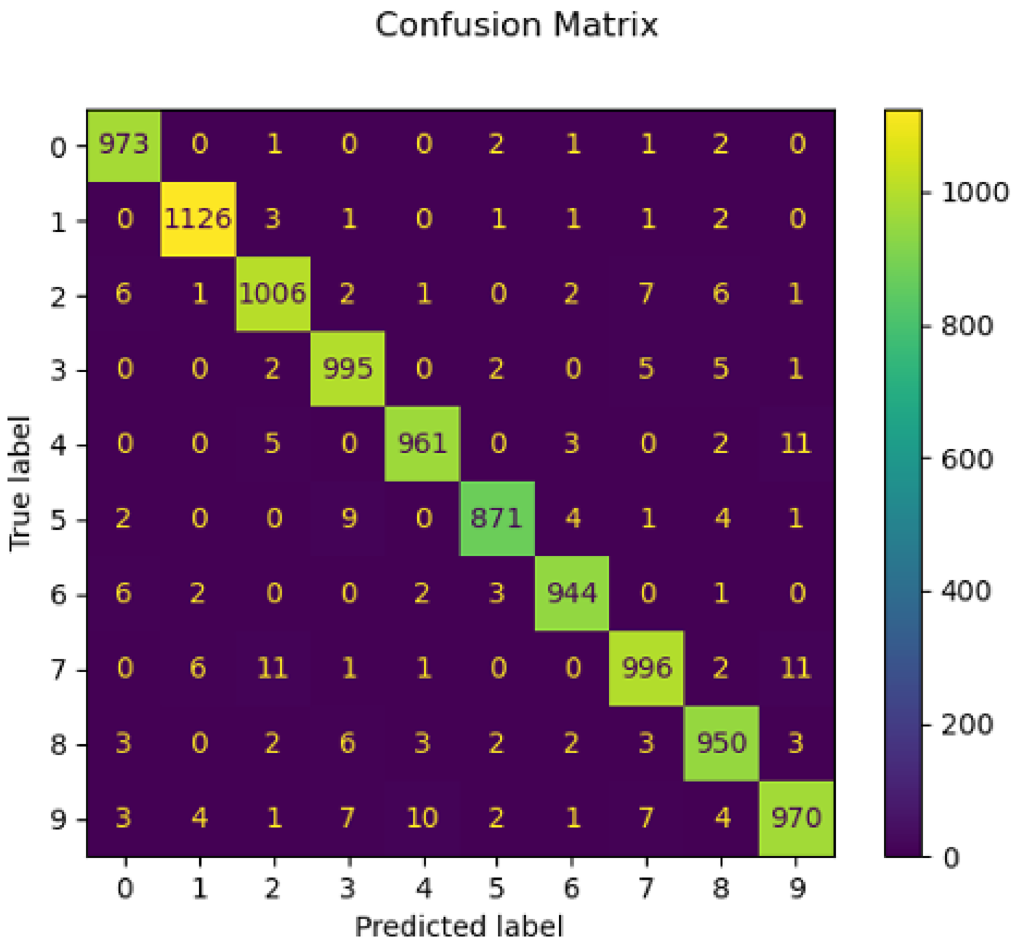
Error Analysis – confusion matrix



True label \ Predicted label	0	1	2	3	4	5	6	7	8	9
0	968	0	1	2	0	3	5	1	0	0
1	0	1127	3	1	0	1	2	0	0	1
2	9	4	991	2	2	0	4	5	14	1
3	0	0	9	961	0	10	1	9	16	4
4	1	0	4	1	949	0	6	3	4	14
5	3	2	1	21	3	841	6	1	8	6
6	7	3	2	0	2	12	930	0	2	0
7	1	17	12	4	9	0	0	964	1	20
8	10	1	7	17	6	19	2	7	900	5
9	5	4	5	13	18	6	0	27	6	925

For k = 4 and variance = 0.64
Accuracy = 95.89%

Improvements – SVM



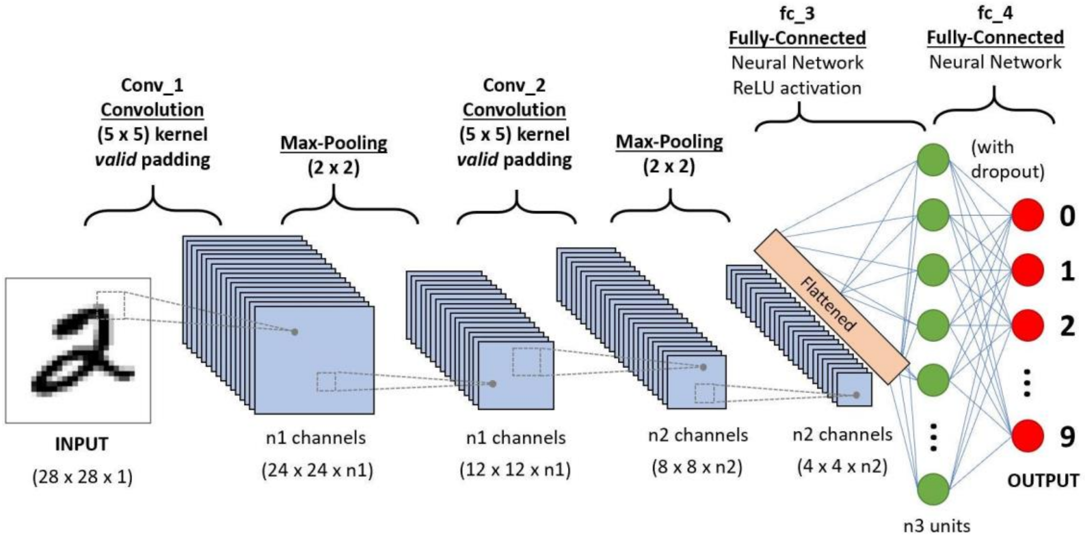
Confusion Matrix

True label \ Predicted label	0	1	2	3	4	5	6	7	8	9
0	973	0	1	0	0	2	1	1	2	0
1	0	1126	3	1	0	1	1	1	2	0
2	6	1	1006	2	1	0	2	7	6	1
3	0	0	2	995	0	2	0	5	5	1
4	0	0	5	0	961	0	3	0	2	11
5	2	0	0	9	0	871	4	1	4	1
6	6	2	0	0	2	3	944	0	1	0
7	0	6	11	1	1	0	0	996	2	11
8	3	0	2	6	3	2	2	3	950	3
9	3	4	1	7	10	2	1	7	4	970

Accuracy = 97.92%

Improvements – CNN

- State-of-the-art method
- Accuracies of up to 99.80% possible
- Our CNN: 99.08%
- Reason: hierarchical feature extraction and end-to-end optimization



References

Grant, P. (2019). k-Nearest Neighbors and the Curse of Dimensionality. https://towardsdatascience.com/k-nearest-neighbors-and-the-curse-of-dimensionality-e39d10a6105d. accessed on: 11.07.2023

Haran, B. (2022). K-d Trees - Computerphile. https://www.youtube.com/watch?v=BK5x7IUTIyU. accessed on: 11.07.2023

Hucker, M. (2020). Tree algorithms explained: Ball Tree Algorithm vs. KD Tree vs. Brute Force.

Meigarom (2017). Dimensionality Reduction — Does PCA really improve classification outcome? https://towardsdatascience.com/dimensionality-reduction-does-pca-really-improve-classification-outcome-6e9ba21f0a32. accessed on: 12.07.2023

Géron, A. (2022). Hands-on machine learning with Scikit-Learn, Keras, and TensorFlow. O'Reilly Media, Inc.

Kanstrén, T. (2020). A Look at Precision, Recall, and F1-Score. https://towardsdatascience.com/a-look-at-precision-recall-and-f1-score-36b5fd0dd3ec. accessed on: 11.07.2023

Kasperek, D., Podpora, M., and Kawala-Sterniuk, A. (2022). Comparison of the Usability of Apple M1 Processors for Various Machine Learning Tasks. Sensors 22, 8005.

Klein, B. NumPy Tutorial. https://www.python-kurs.eu/numpy.php. accessed on: 07.07.2023

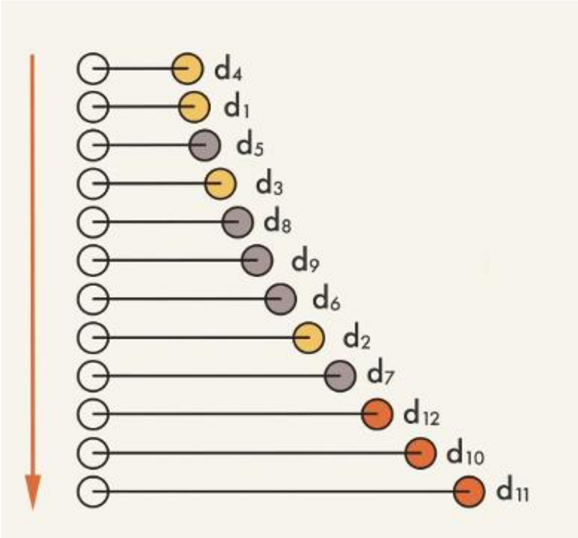
LeCun, Y., Bengio, Y., and Hinton, G. (2015). Deep learning. Nature 521, 436-444. 10.1038/nature14539.

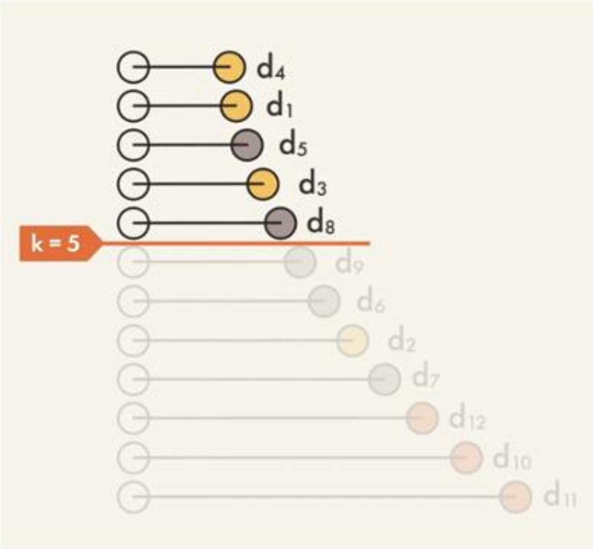
scikit-learn. sklearn.neighbors.KNeighborsClassifier. https://scikit-learn.org/stable/modules/generated/sklearn.neighbors.KNeighborsClassifier.html. accessed on: 13.07.2023

Additional Slides

K-Nearest Neighbors

1. Calculating euclidean distance between test data point and train data points
2. Sort distances in ascending order
3. Select top k-rows
4. Majority vote
5. Calculate accuracy





What could we have improved?

Our Project:

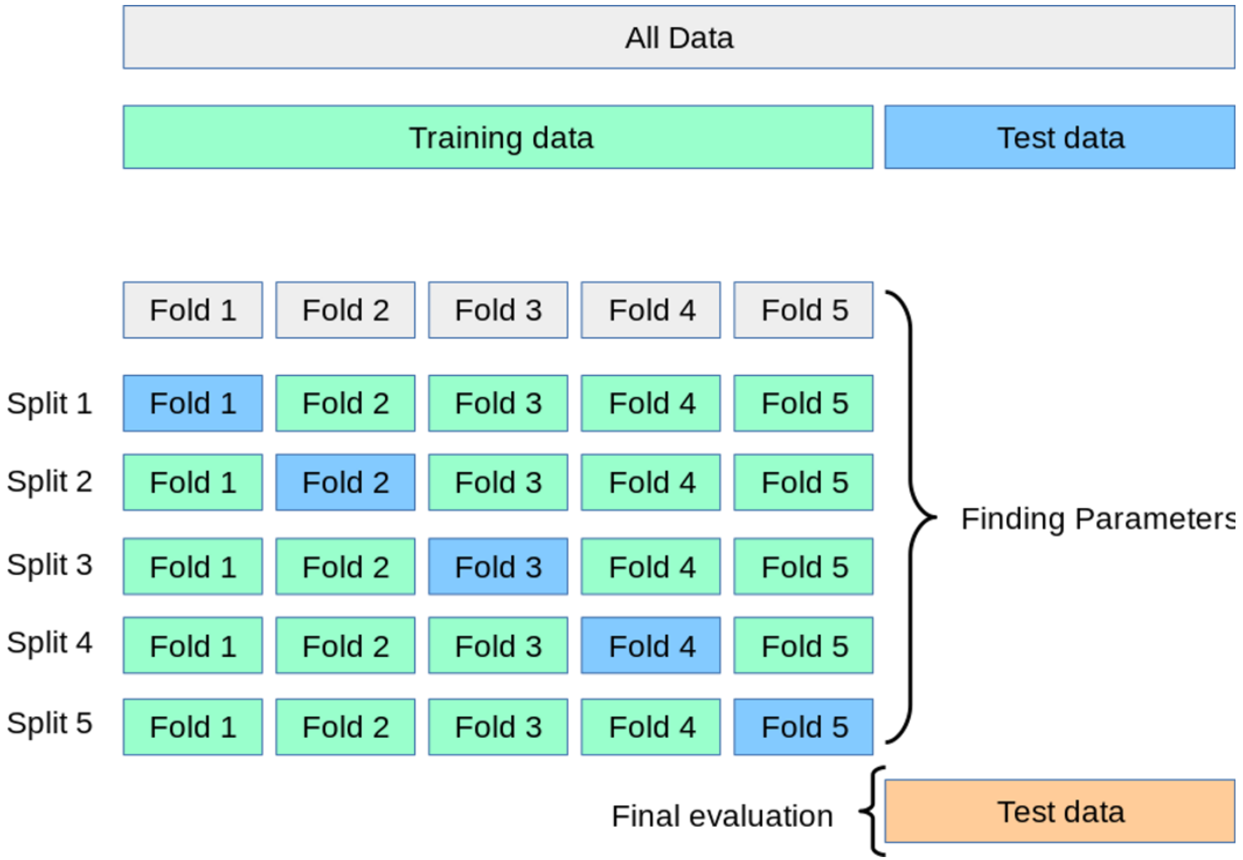
- 1 train data set
- 1 test data set

Improvement:

➢ Split of data sets

- 1 train data set
- 1 validation data set
- 1 test data set

K-fold cross validation



All Data

Training data | Test data

Fold 1 | Fold 2 | Fold 3 | Fold 4 | Fold 5

Split 1: 🟦 Fold 1 | Fold 2 | Fold 3 | Fold 4 | Fold 5

Split 2: Fold 1 | 🟦 Fold 2 | Fold 3 | Fold 4 | Fold 5

Split 3: Fold 1 | Fold 2 | 🟦 Fold 3 | Fold 4 | Fold 5

Split 4: Fold 1 | Fold 2 | Fold 3 | 🟦 Fold 4 | Fold 5

Split 5: Fold 1 | Fold 2 | Fold 3 | Fold 4 | 🟦 Fold 5

Finding Parameters

Final evaluation: Test data