Implementation and evaluation of K-nearest neighbors (KNN) algorithm for handwritten digit recognition

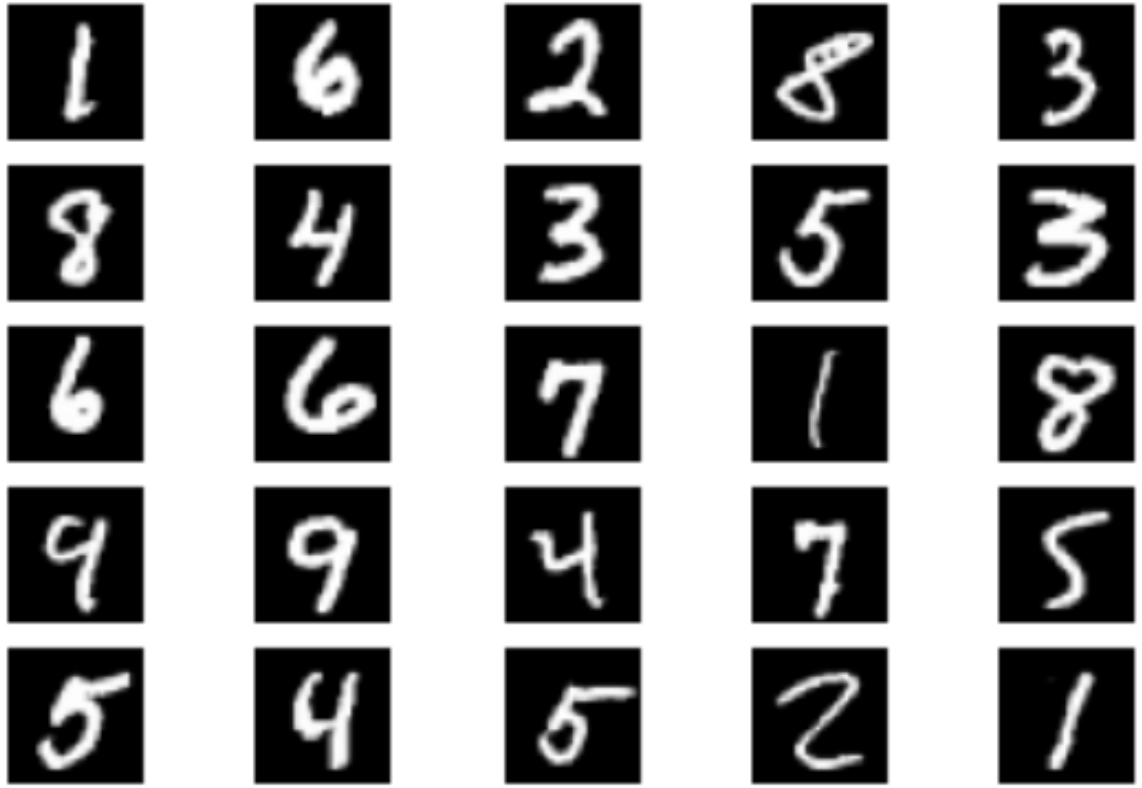
Final Presentation – Data Science 2021

Project 5 Group 2

Nina Gutzeit, Maximilian Hingerl, Emma Kray, Johannes Müller

Recap of 4 Milestones

1. Milestone: implementing data normalization
2. Milestone: implementing PCA
3. Milestone: implementing a classification algorithm
4. Milestone: testing the algorithm



Data Cleaning

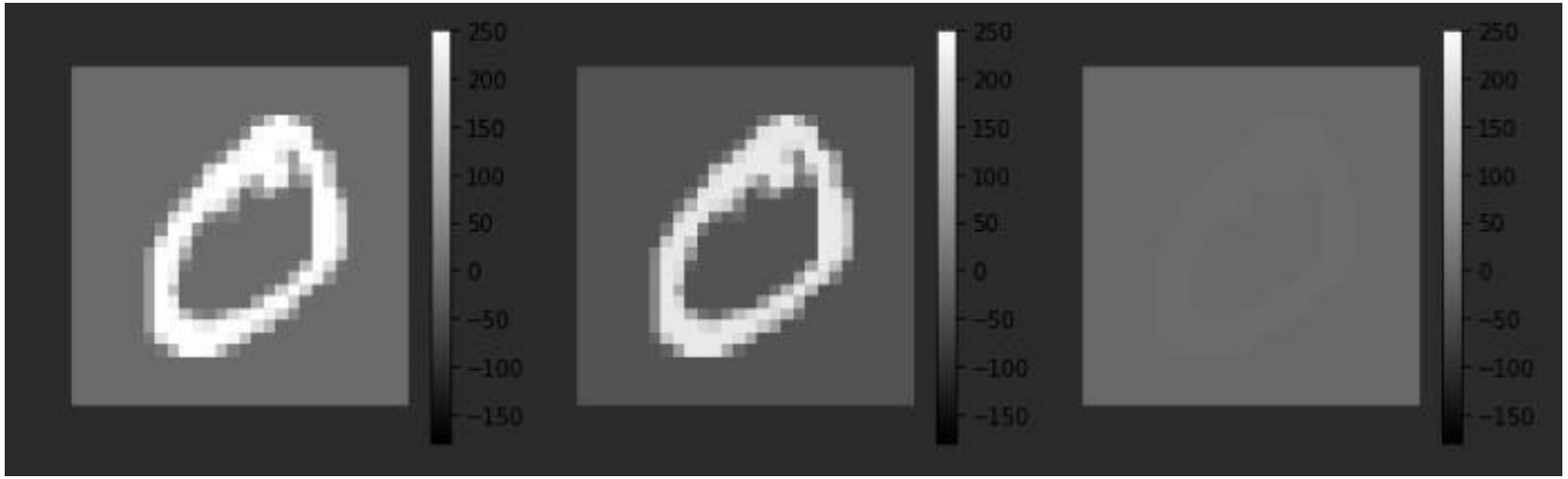
```
# Checking data for range
rm_range(train_values)
rm_range(train_labels, upper=10)
rm_range(test_values)
rm_range(test_labels, upper=10)
```

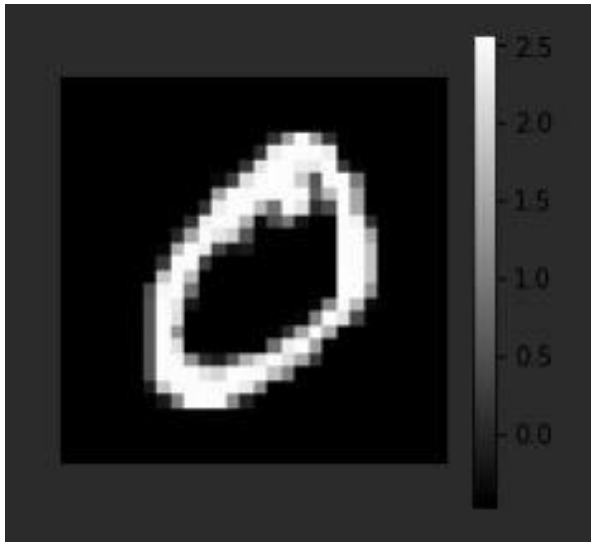
```
No values out of range.
No values out of range.
No values out of range.
No values out of range.
```

```
# Checking Training data for missing values:
any_na(train_values)
# Checking Test data for missing values:
any_na(test_values)
```

```
There are no missing values in this data.
There are no missing values in this data.
```

1.1 Standardization





Original image

Centered

$$(X_i - \overline{X})$$

Z- transformed

$$\frac{(X_i - \overline{X})}{\sigma_i}$$

Z-transformed, other color scale

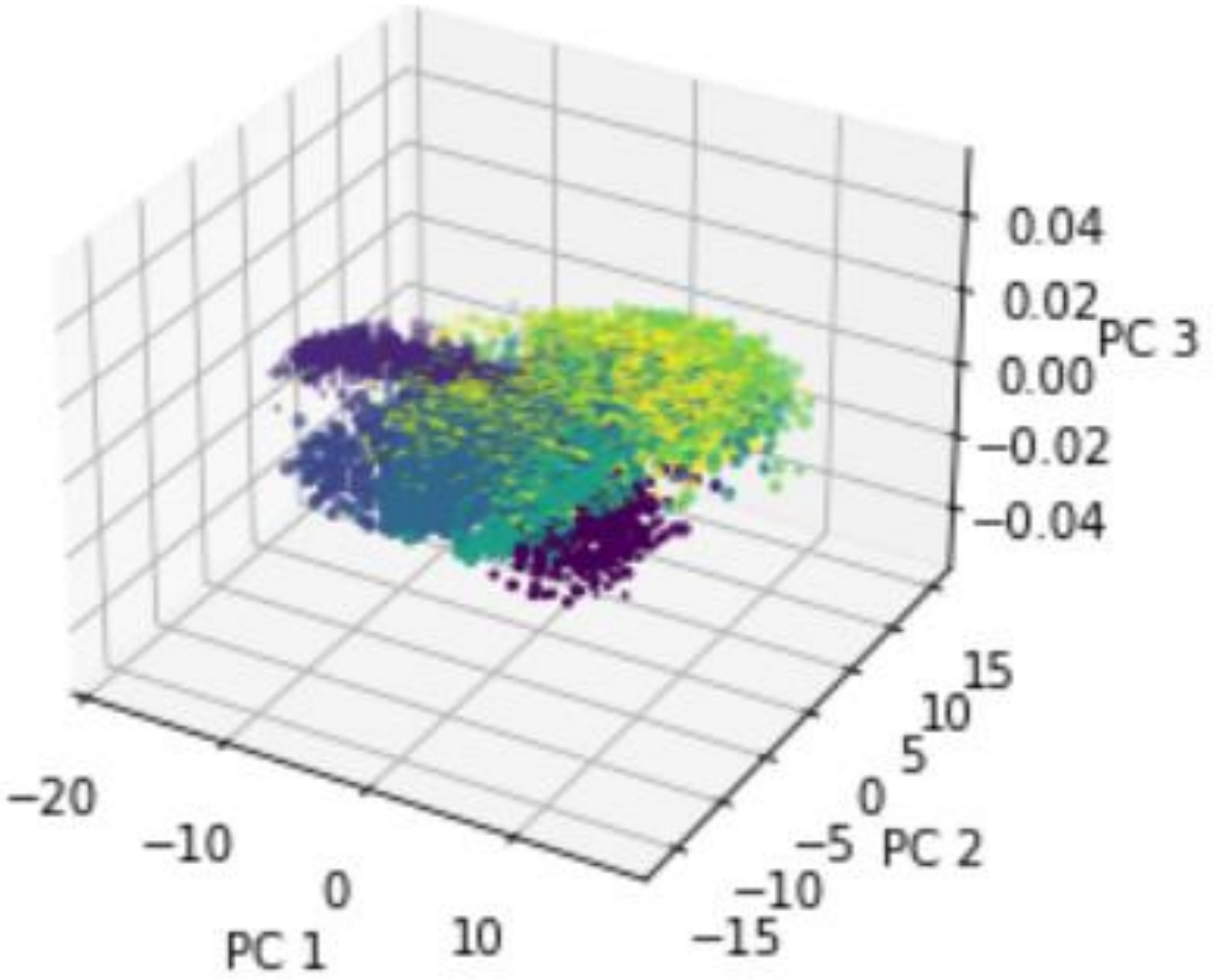
1.2 PCA Preparation

$$\operatorname{corr}(x,y) = \frac{1}{N-1} \cdot \sum_{i=1}^{N} \frac{\left(X_i - \overline{X}\right)}{\sigma_x} \frac{\left(Y_i - \overline{Y}\right)}{\sigma_y}$$

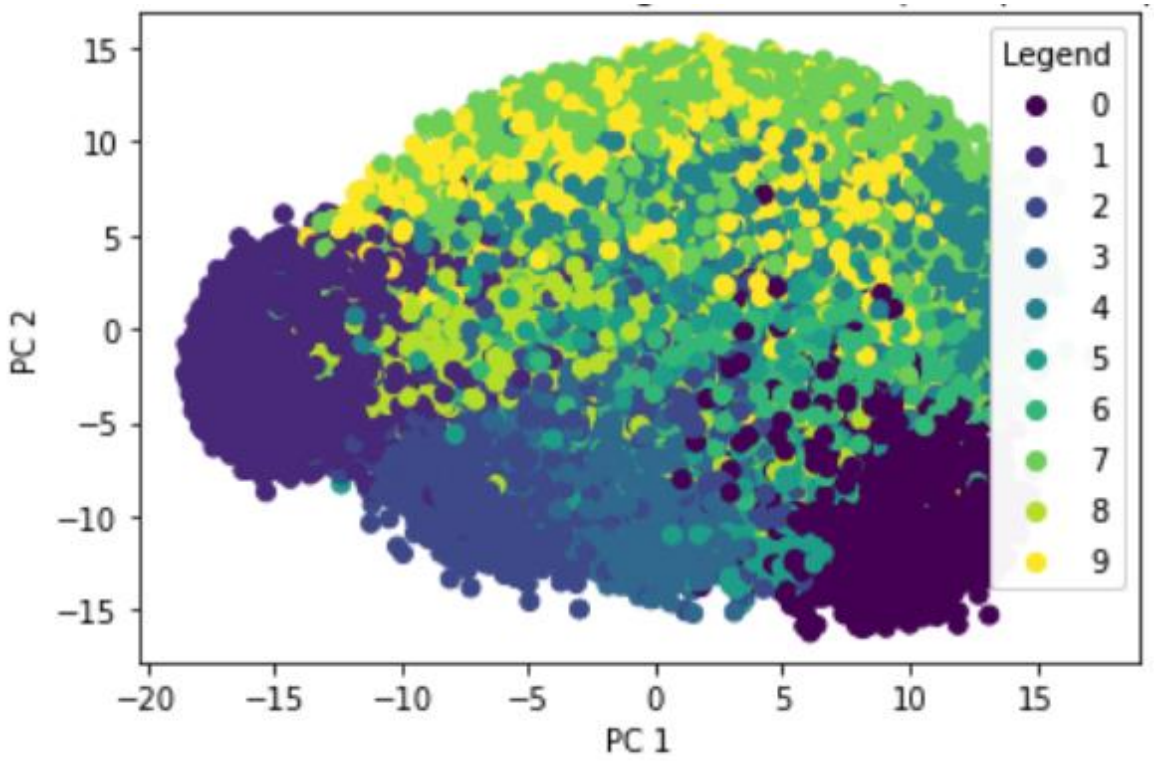
$$\operatorname{cov}(x,y) = \frac{1}{N-1} \cdot \sum_{i=1}^{N} \left(X_i - \overline{X}\right)\left(Y_i - \overline{Y}\right)$$

2. Principal Component Analysis

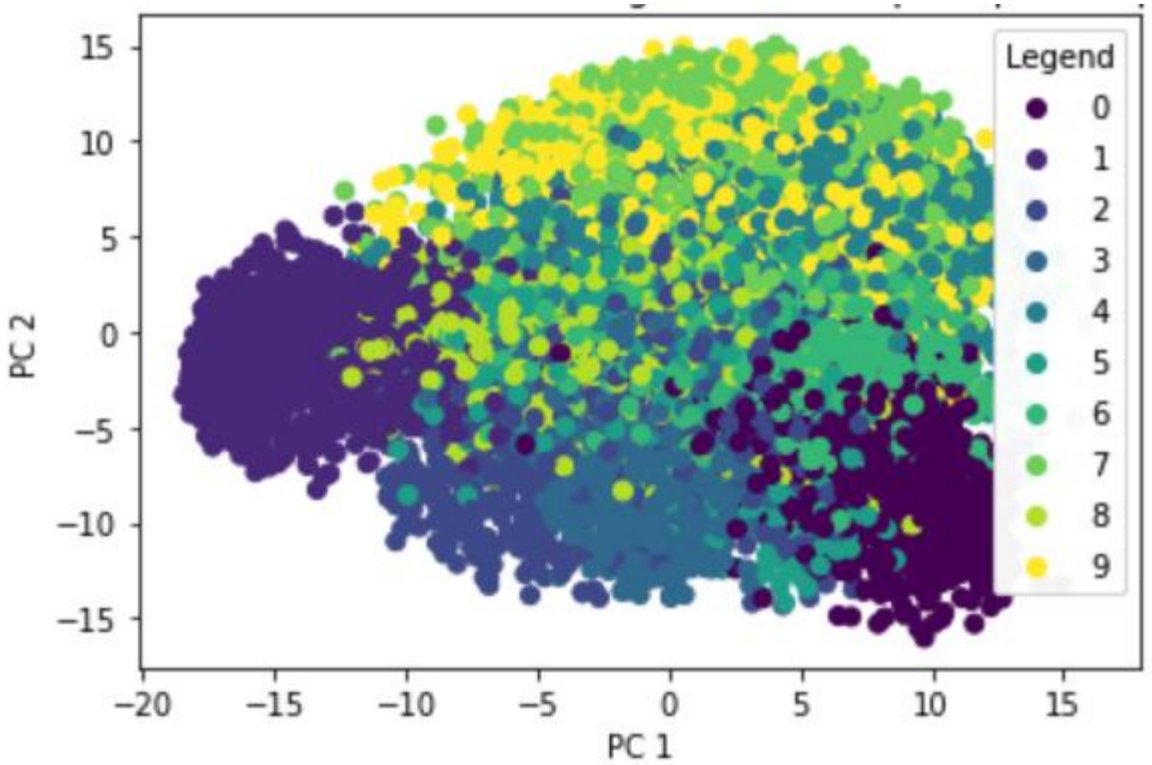
- 3D Plot
- Three best principal component
- Each Digit has a different color



2. Principal Component Analysis

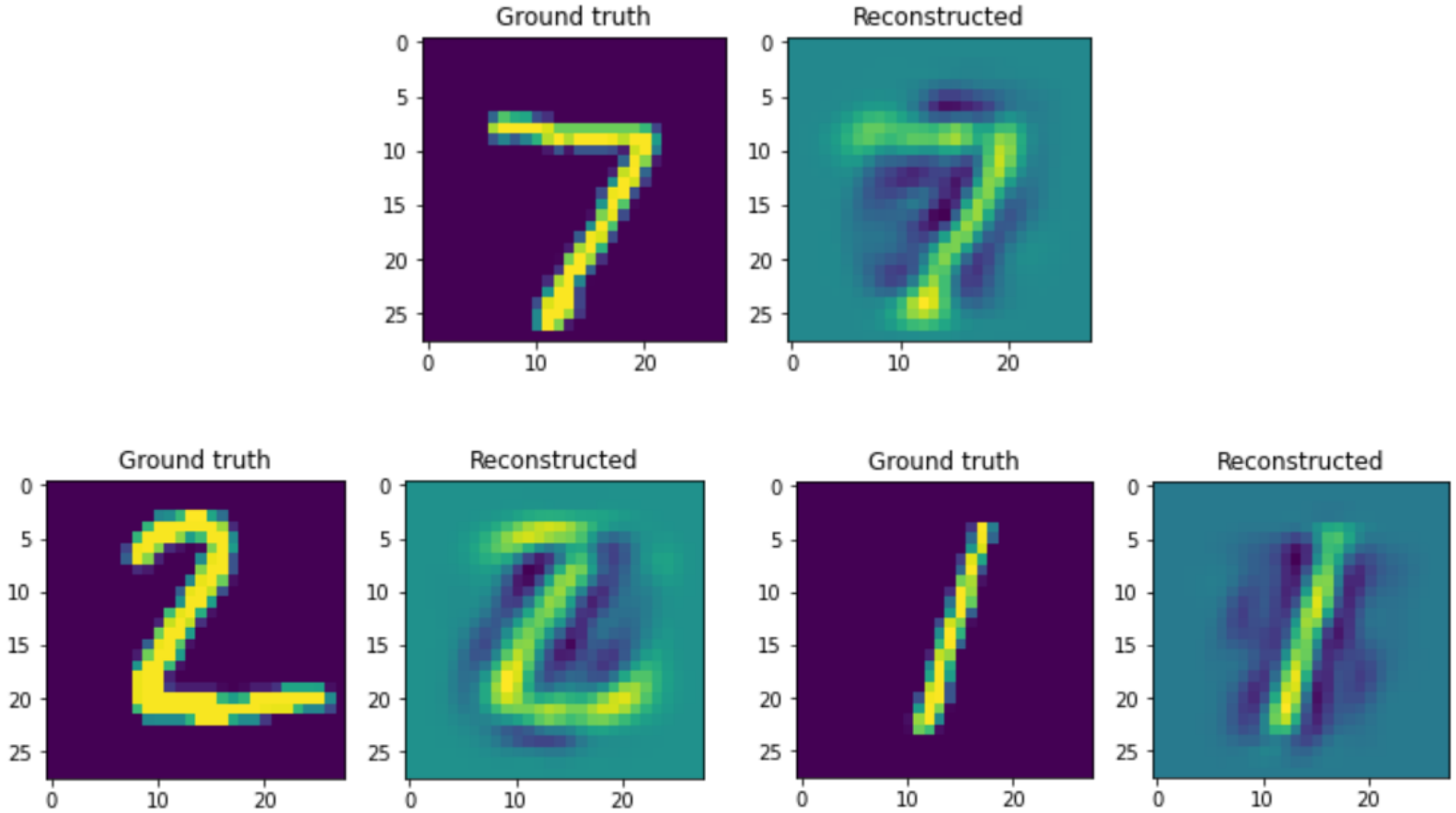


2D Visualization of training values

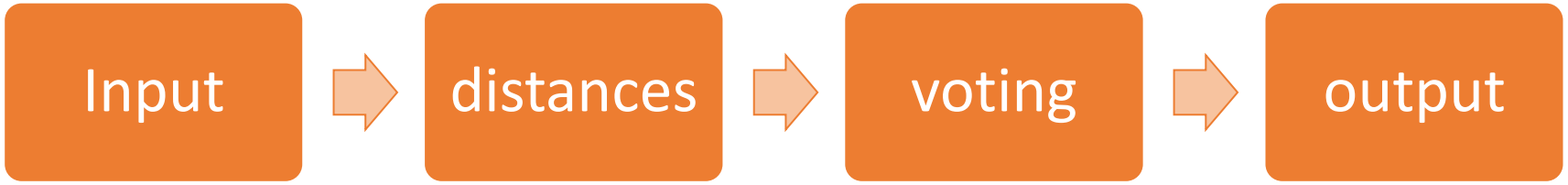


2D Visualization of test values

2.1 Comparison before and after PCA



3. KNN – The Algorithm



correctly classified vs wrongly classified numbers using the euclidean distance: 9803 vs 197

3.1 KNN – Distance Methods

$$d_{Euclidean}(x, y) = \sqrt{\sum_{i=1}^{n} (x_i - y_i)^2}$$

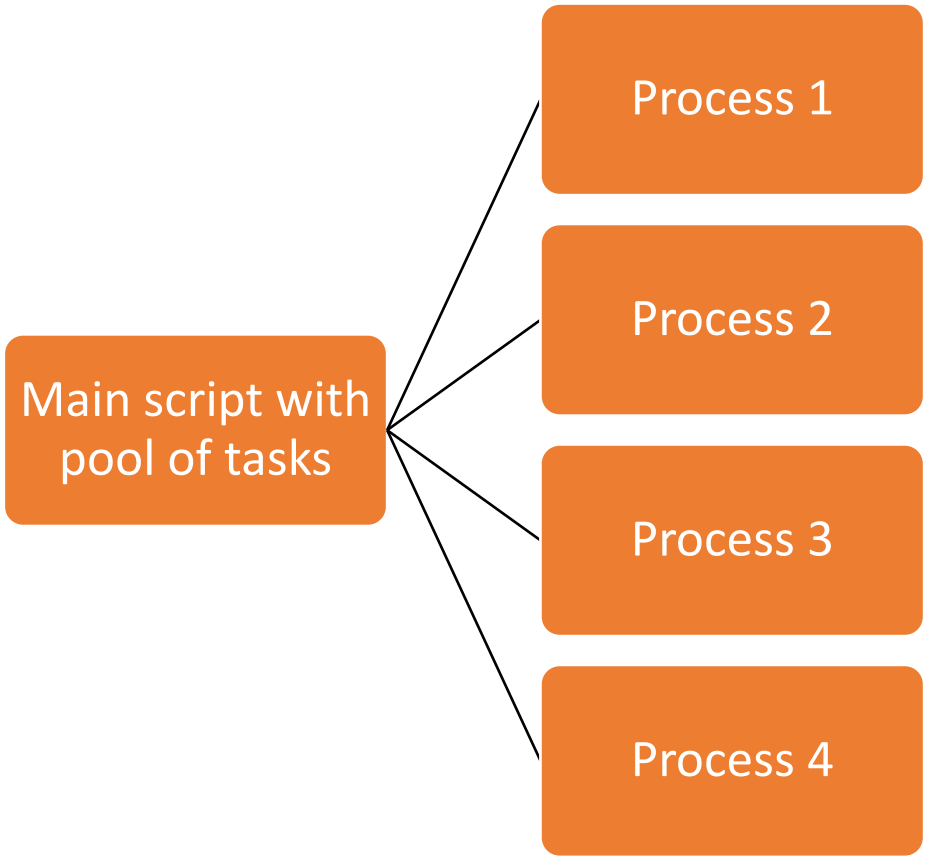
- x_i = training data points
- Y_i = test data points

$$d_{Manhattan}(x, y) = \sum_{i=1}^{n} |x_i - y_i|$$

```
correctly classified vs wrongly classified numbers using  the euclidean distance: 9803 vs 197
correctly classified vs wrongly classified numbers using the manhattan distance: 9786 vs 214
```

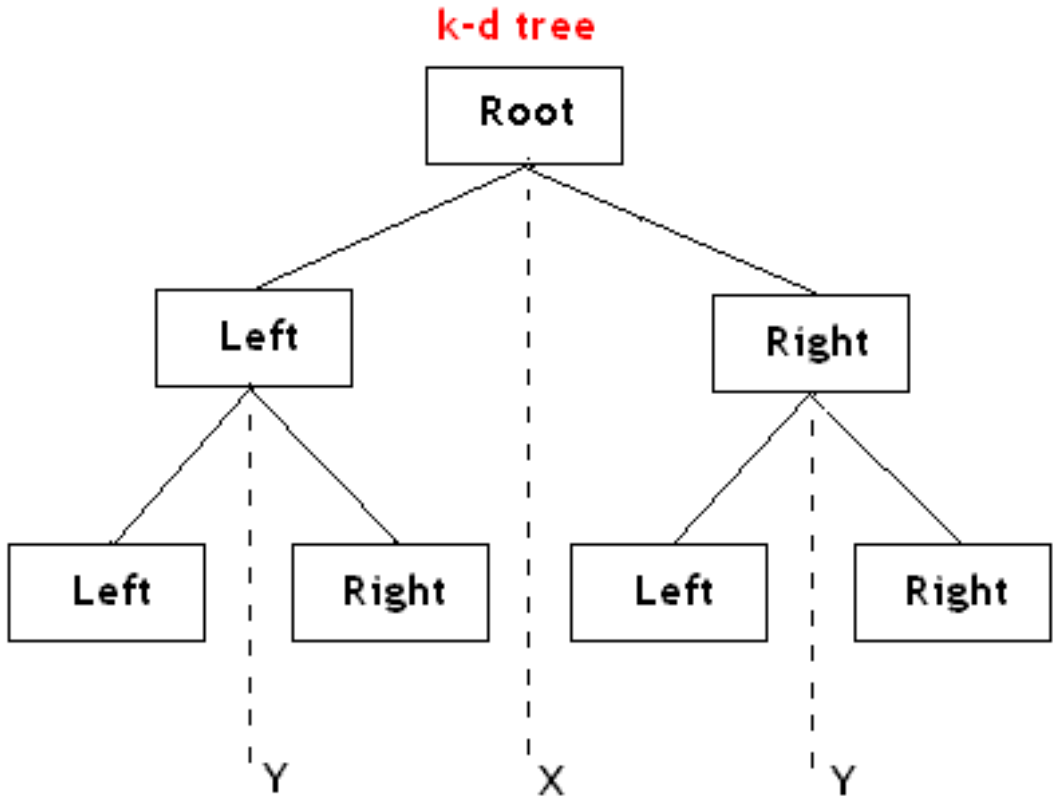
3.2 KNN – Multiprocessing

- Main script has pool of tasks
- Each process gets a chunk of tasks
  - Each process gets new chunk of tasks when finished
- Time improvement from about 2:40 mins to 1:10



3.3 KNN – kd-trees

- Space partitioning method
- Splitting training data along the median for each dimension
- KDTree function from scipy.spatial
- Reduced run time from 1:10 min to 0:10 min



https://www.codeproject.com/Articles/18113/KD-Tree-Searching-in-N-dimensions-Part-I

4. Testing and Optimizing

Fig. 8: 3D Visualization of KNN Accuracy

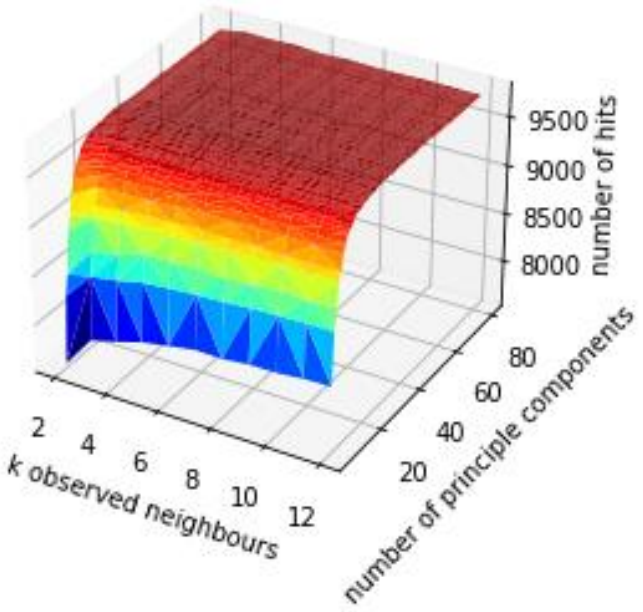


Fig. 7: KNN Accuracy Heatmap

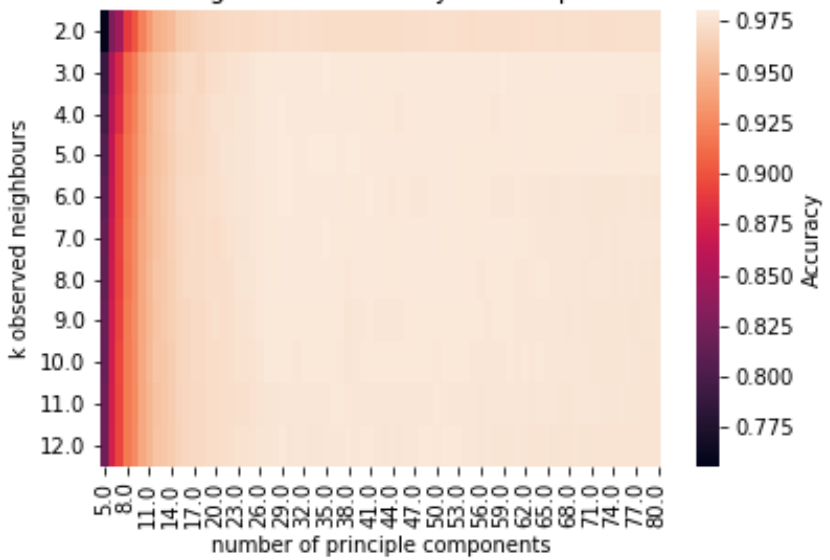
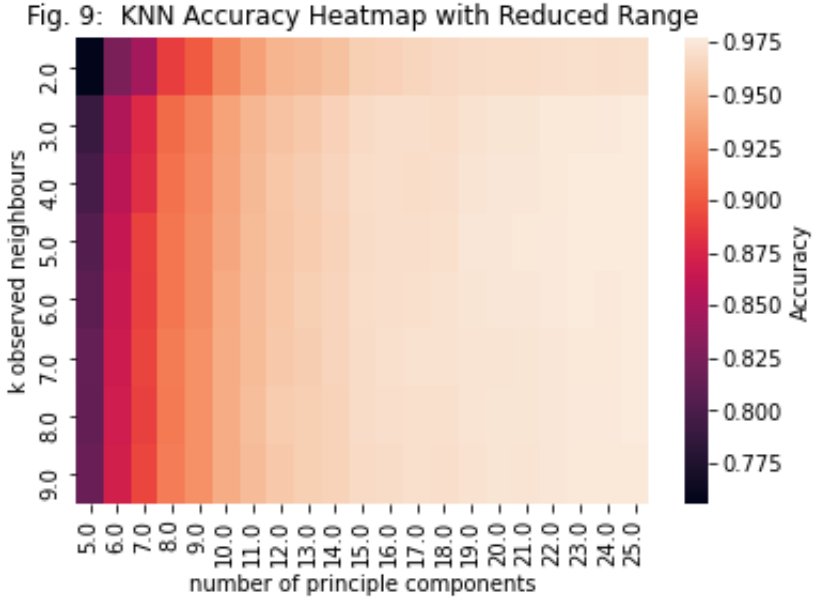


Fig. 9: KNN Accuracy Heatmap with Reduced Range

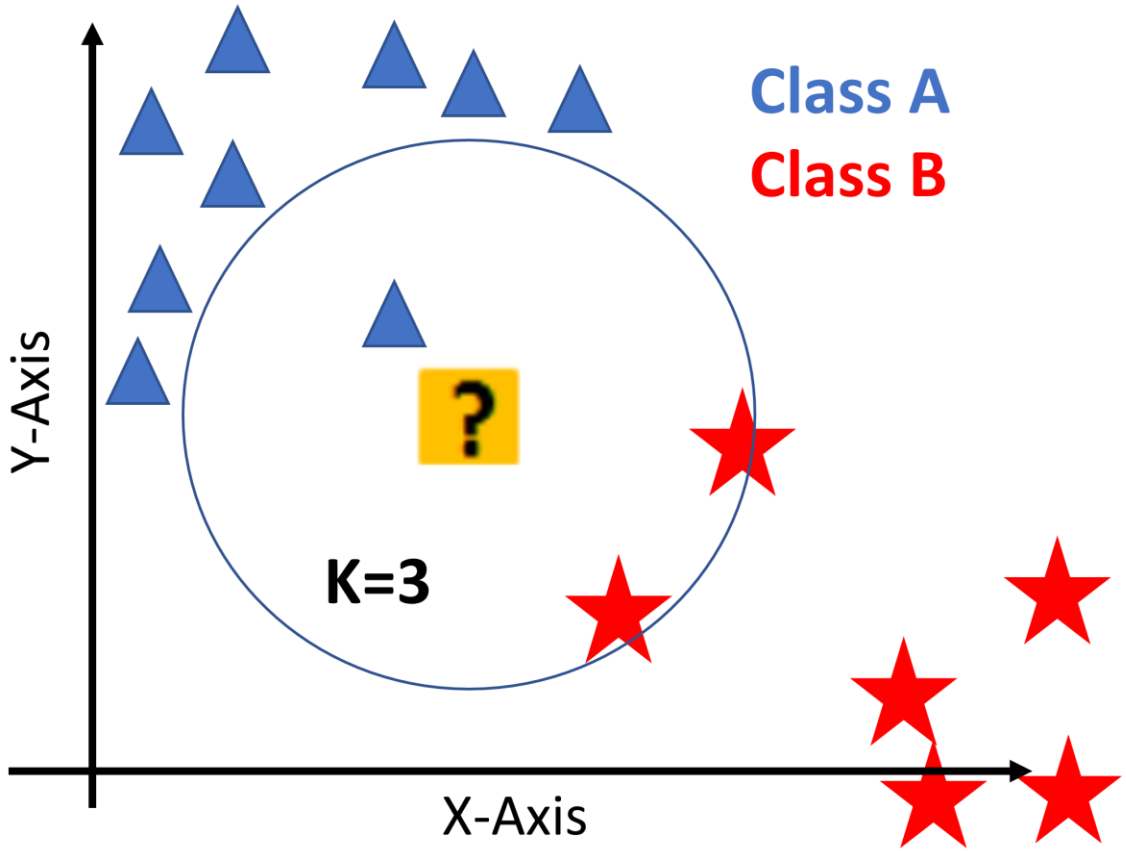


highest accuracy/number of correctly identified images and the corresponding n and k values: [[30   5 9803]]

Additional Implementations

Weighted KNN

- Extension of normal KNN
- Using distances of nearest neighbors

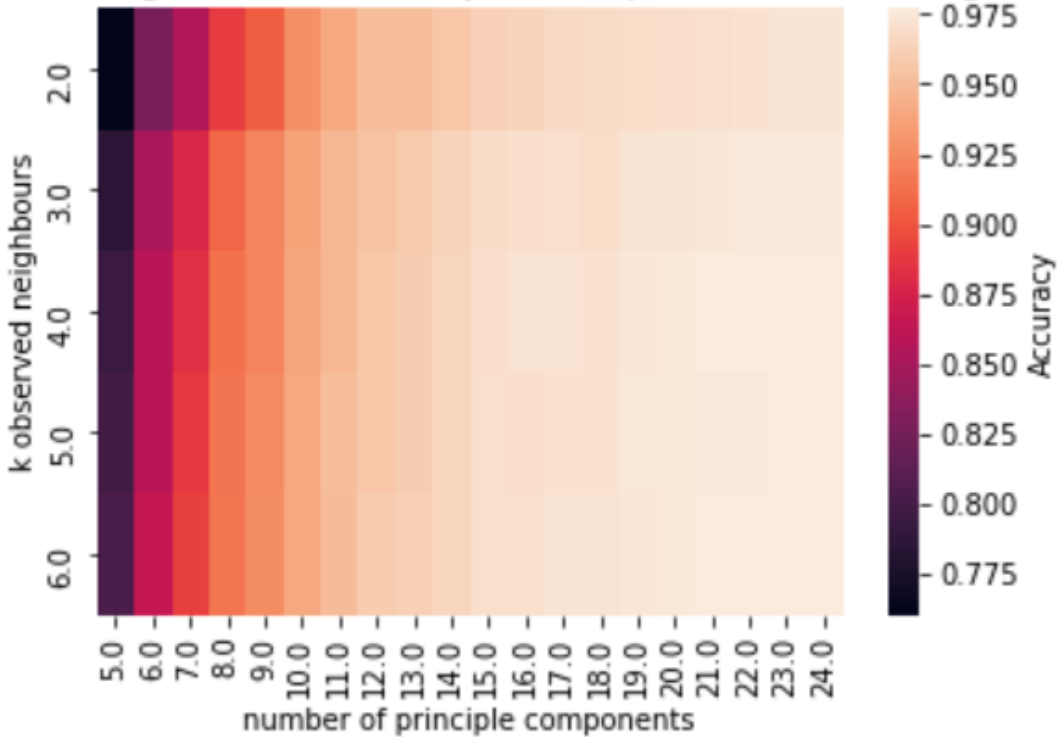


Performance of Weighted KNN

- k = 4
- num_components = 34

- 98.12% accuracy
- → Improved by 9 pictures

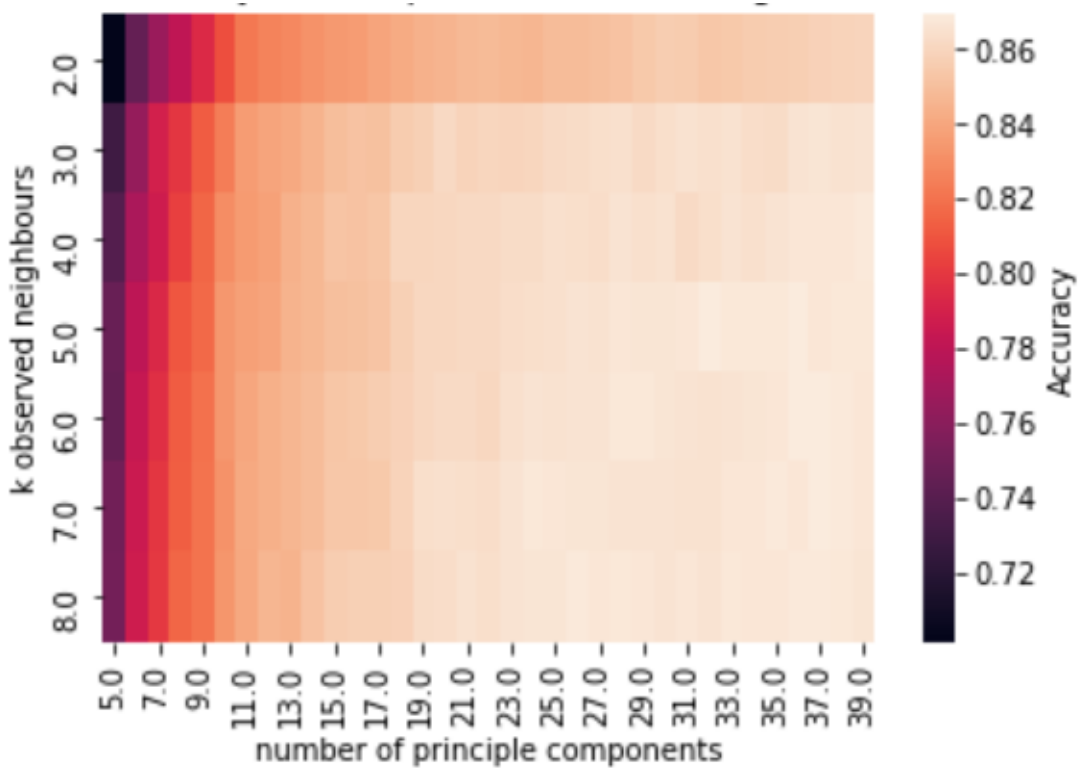
Fig. 10: Weighted KNN Accuracy Heatmap with Reduced Range

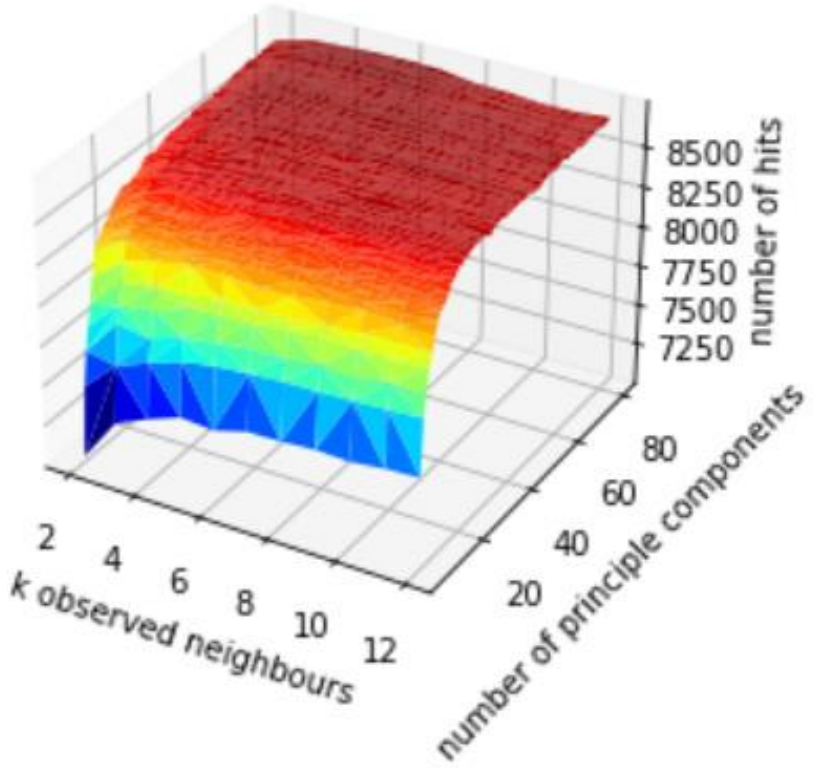


Fashion Mnist

- The data set:
  - Table
  - 28x28 pixels
  - Train set: 60 000 images
  - Test set: 10 000 images
- Applying algorithm

Fashion Mnist performance





- k = 6
- num_components = 75
- Accuracy 87%

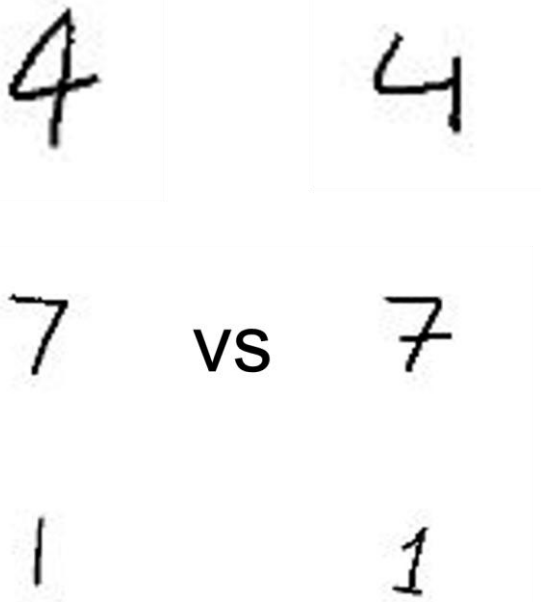
Phone Numbers

Please write your phone number in **bold** legible handwriting. For better results write your number in the centre of each box using a digital ballpoint pen. Write in the large without drawing beyond the border.

Phone Numbers: Limitations

- unique handwriting
- number variants: crossing 7s, capping 1s

→ Is 60000 images enough?



Thank you for your Attention!

Sources

- Beardmore, A. (2020, October 12). Uncovering the Environmental Impact of Cloud Computing. Earth.Org. https://earth.org/environmental-impact-of-cloud-computing/.
- Cook, G. (2012, April). How Clean is Your Cloud? greenpeace.org. https://www.greenpeace.org/static/planet4-international-stateless/2012/04/e7c8ff21-howcleanisyourcloud.pdf.s
- James, G., Witten, D., Hastie, T., & Tibshirani, R. (2021). K-nearest neighbors. In An introduction to statistical learning: with applications in R (p. 163). essay, Springer.
- Lucivero F. (2020). Big Data, Big Waste? A Reflection on the Environmental Sustainability of Big Data Initiatives. Science and engineering ethics, 26(2), 1009–1030. https://doi.org/10.1007/s11948-019-00171-7 (Lucivero, 2020)
- Sattiraju, N. (2020, April 2). Secret Cost of Google's Data Centers: Billions of Gallons of Water. Time. https://time.com/5814276/google-data-centers-water/. (Sattiraju, 2020)
- Walsh, B. (2014, April 2). New Greenpeace Report Shows the Environmental Impact of the Internet. Time. https://time.com/46777/your-data-is-dirty-the-carbon-price-of-cloud-computing/. (Walsh, 2014)