ML Assignment 1

The GitHub is created.

Task 1

For executing the task, I am using scikit-learn in python which has PCA; t-SNE and Random Forrest. First, the visualization, here just for the parametric data, it is first normalized. This data is taken to execute the PCA and to fit the t-SNE. Figure 1 visualizes the results of PCA and t-SNE when all the present numeric parameters including the casual, registered and count are used.

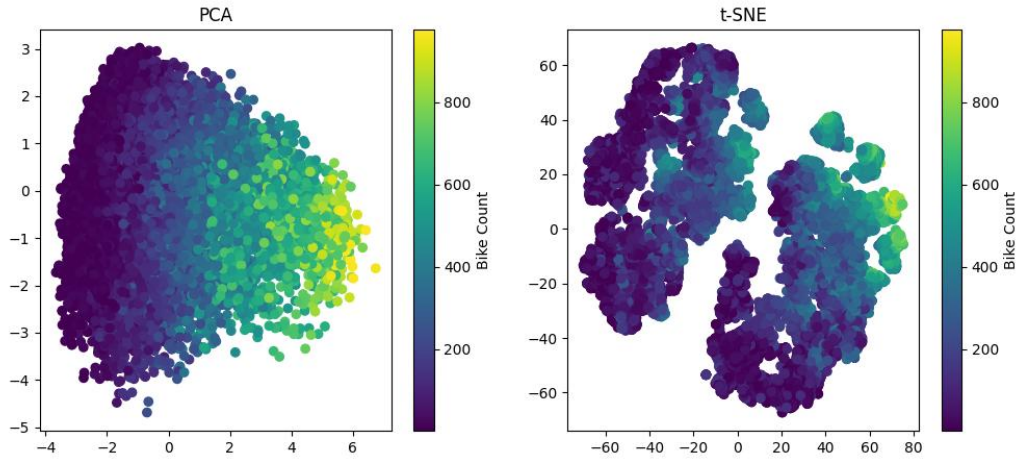


Figure 1: Visualization of the full data including all the numeric parameters

The next visualization is done excluding the casual, registered and count, only focusing on the features (Figure 2). As this is the data to be predicted. In the following just the prediction of the count is investigated further. In the PCA features are represented as PC1 is very correlated with the temperature(temp) and the felt one. PC1 and PC2 are slightly correlated with months. PC2 is correlated with the humidity. In a second step the same is done for the data but all-time data is excluded. In Figure 3 the location of the parameters represents the influence of different features in the data reduction. Overall, one can see that neighbor properties which are kept in the t-SNE. For the t-SNE the data is split according to the year and

Table 1: Loading of the PC1 and PC2, meaning the feature representation

	PC1	PC2
year	0.052	-0.088
month	0.240	0.342
hour	0.177	-0.444
weekday	-0.003	0.044
month	0.240	0.342

temp	0.671	0.008
feel temp	0.671	0.032
humidity	-0.076	0.625
windspeed	-0.040	-0.533

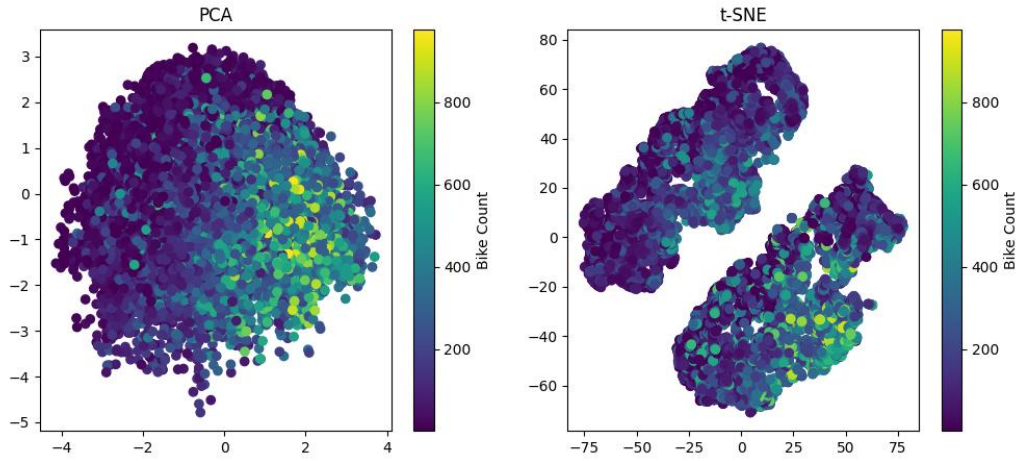


Figure 2: Visualization of the data using PCA and t-SNE

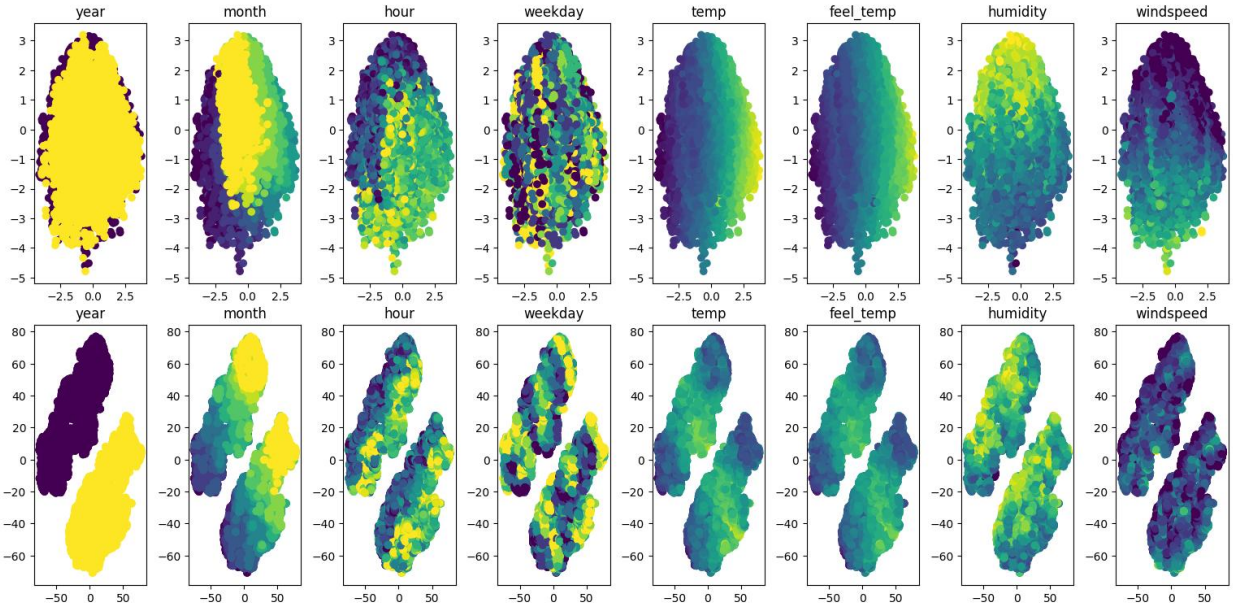


Figure 3: Where each feature is present in the dimensional reduction. The upper row is the result for PCA and the lower row for t-SNE. The more yellow a dot the higher the value of the feature is.

a trend for the month and hour is visible. The weekday seem seasonal. The temperature seems centered. For PCA the temperature and the humidity are the highest in respective direction which is according to the PC Loading for PC1 and PC2 respectively. Now for all

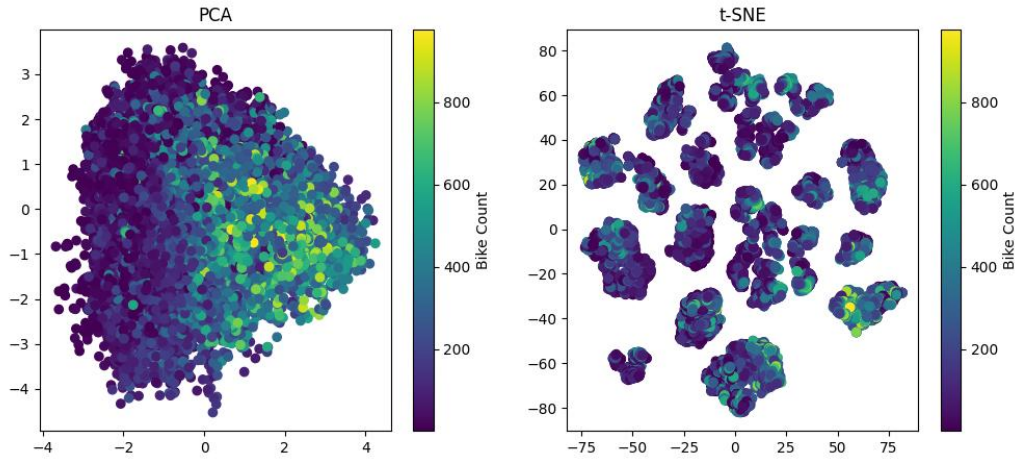
the data and all parametric features, a Random Forrest model is trained and evaluated. For this, the pretreated data is used with the in scikit-learn method train test split to split the data into testing and training data. As comparison metric R^2 and MSE are used. The best result is achieved taking the full data into account, the second best using the with t-SNE dimensionally reduced data, but the R^2 value is very much lower for t-SNE compared to the Original data but the model trained with PCA processed data performance the worst.

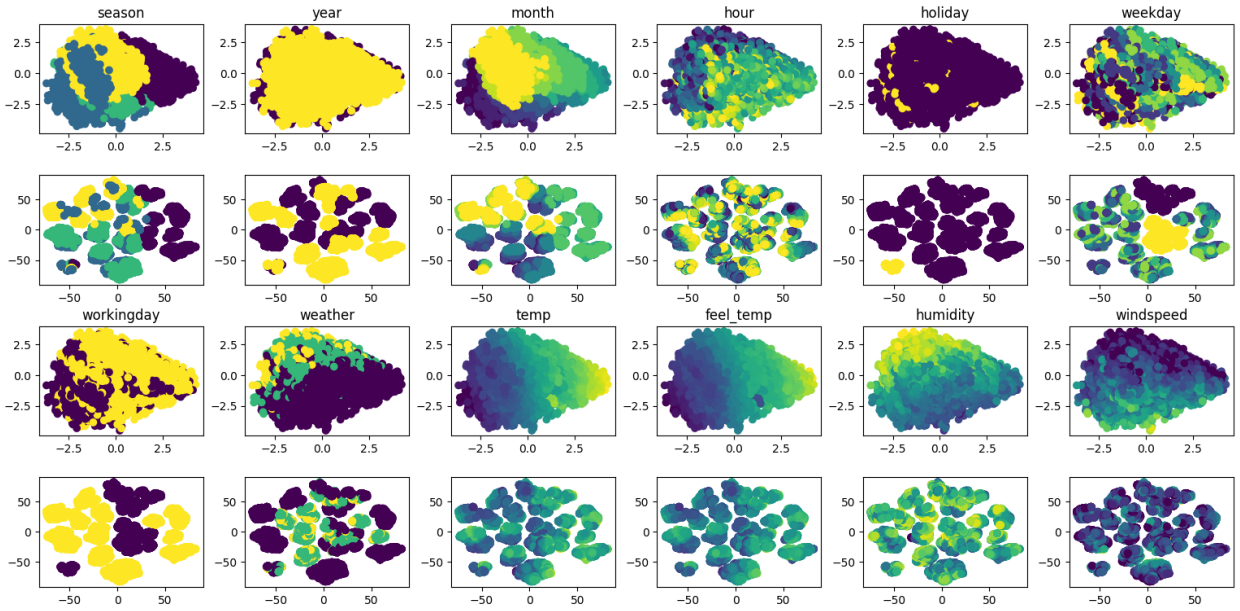
Table 2: Results of predicting the count using the parametric values and Random Forest with 50 estimators and different pretreated data.

	Original data	PCA	PCA	t-SNE	t-SNE
number of features	all excluding counts	2	3	2	3
MSE	2550	29356	16044	13970	12911
R^2	0.92	0.24	0.49	0.56	0.59

The pipeline is reaped in state of removing the non-parametric values, these values are converted to a numeric value, if the number of unique instances does not exceed 10. The count numbers are excluded in the data.

Visualization





With all the categorial features included the importance of local information is kept, for example, for the holiday and weekday. Compared with the former data the data seems to be more separated for the DR with t-SNE. The Random Forrest results did not get improved.

Task 2

Using SOM one can reduce the dimensionality of the data, where for a sample x_i the best weight w_j is iteratively found and being updated. The best weight is found by using the Euclidean distance between x_i and w_j. The model preserves local information and clusters it.

The weights are randomly initialized.

1000 iterations

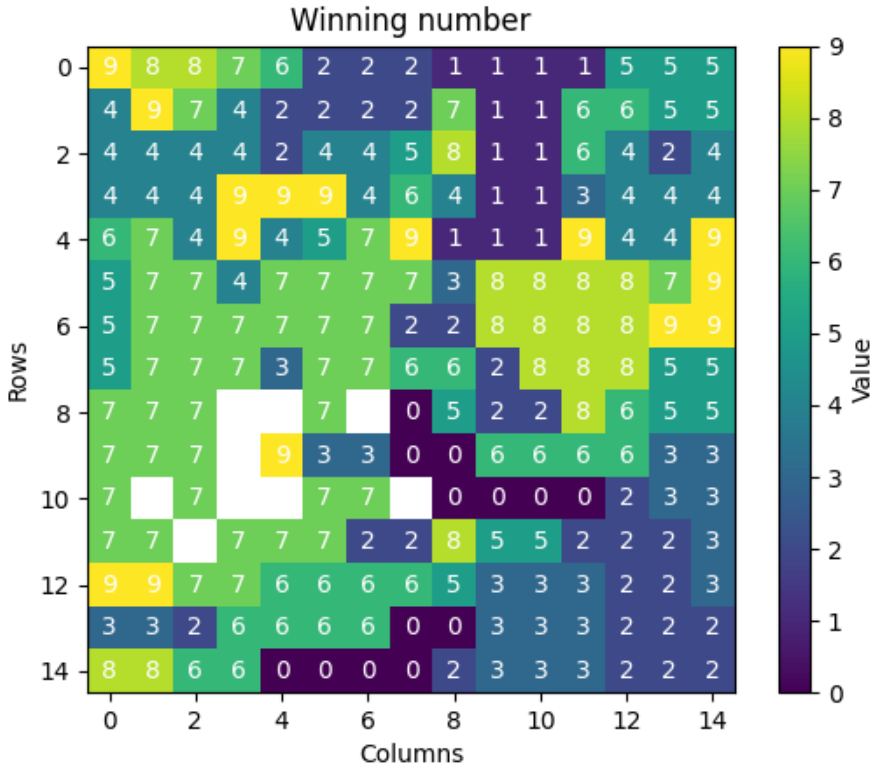


Figure 4: Winner of each Pixel after 1000 iterations

10000 iterations

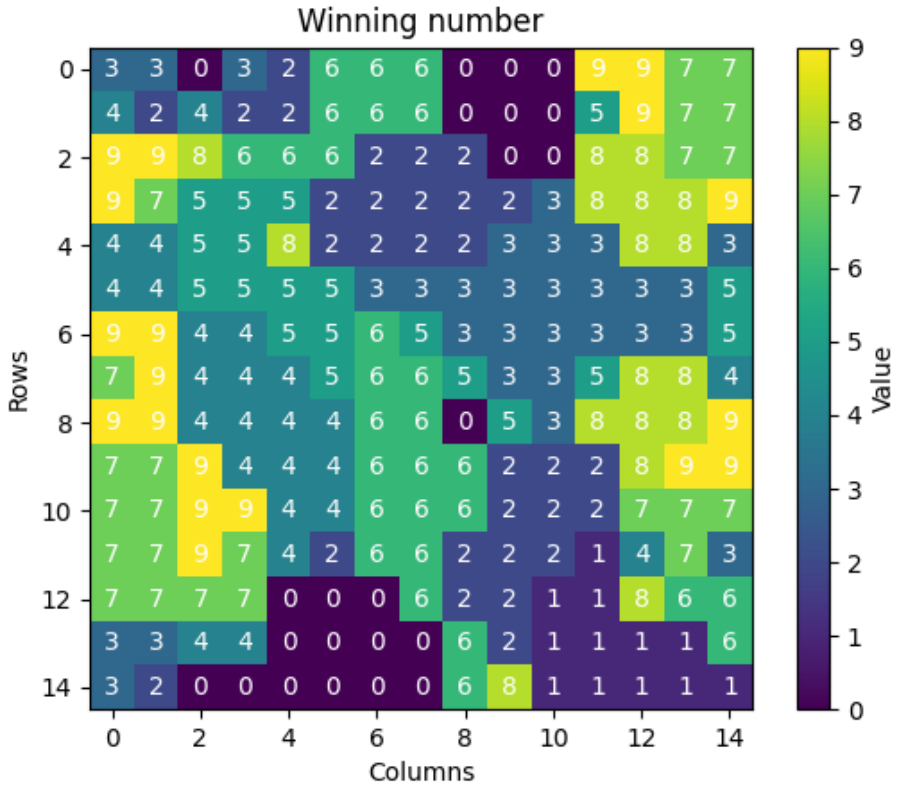


Figure 5: Winner of each Pixel after 10000 iterations

50000 iterations

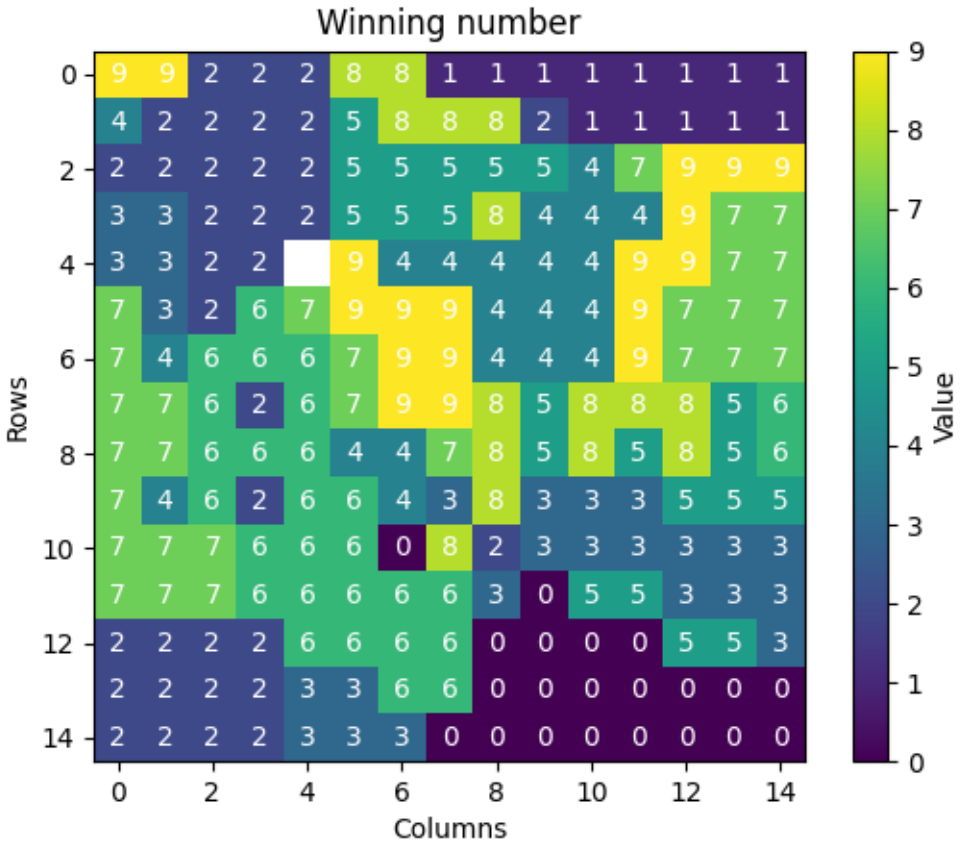


Figure 6: Winner of each Pixel after 50000 iterations

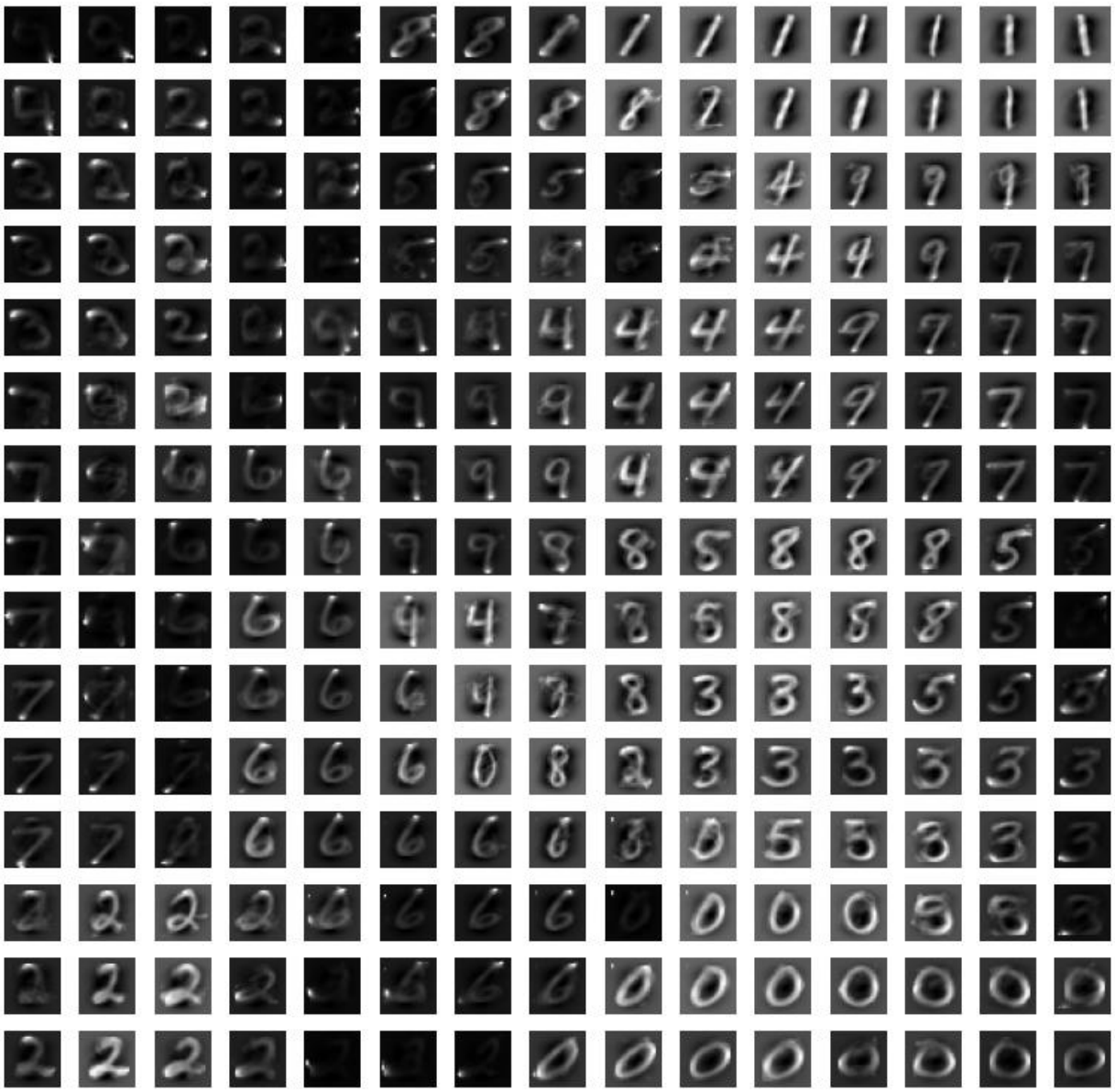


Figure 7: Weights for the SOM for each pixel after 50000 iterations

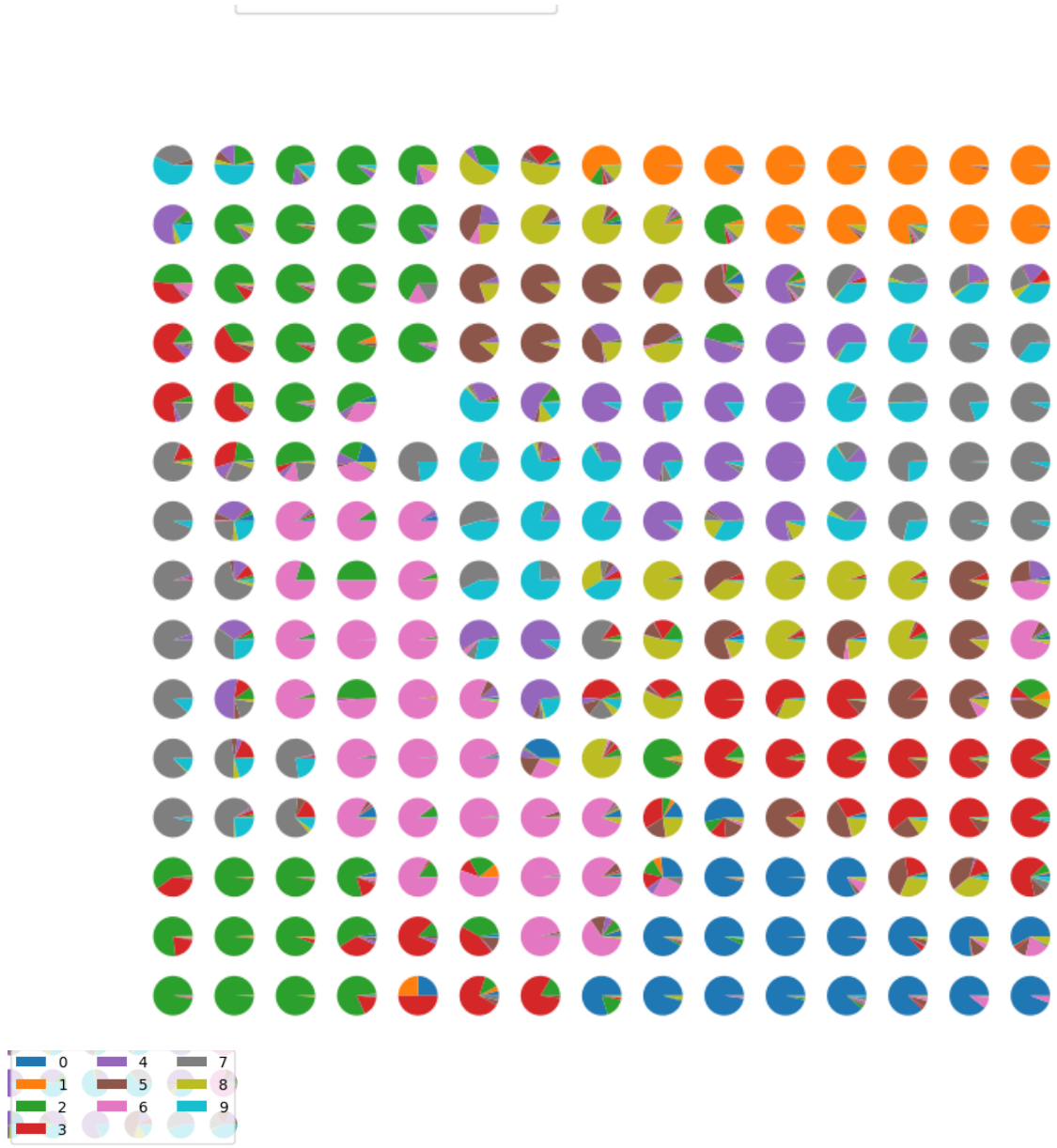


Figure 8: Diagram with Pie chart for every pixel for the relative amount of the presence of the number in the pixel

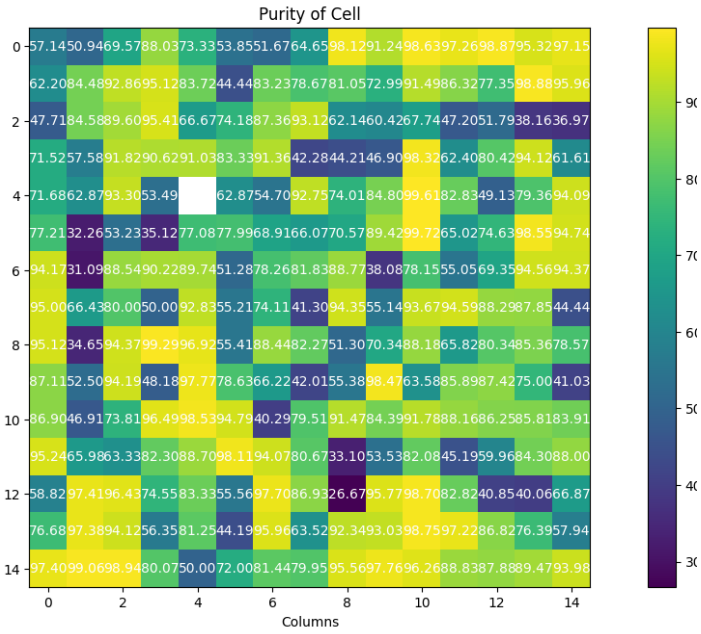


Figure 9 Percentage of purity of the neuron/pixel wr.t to the most present value. Purity count(most present value)/count (total in pixel).

The clustering reveals local similarities between different numbers, for example the pixels of 4 and 9 are both very similar, which can be seen in Figure 7 and 8. This might be a problem for a classification ML approach, and the training or cost function needs to be adjusted to focus on these difficult classes. Further the distance between 1 and 7 (similar numbers) and 0 is bigger than between 1 and 7

Sources

https://scikit-learn.org/stable/modules/generated/sklearn.manifold.TSNE.html 1/11/25

https://www.datacamp.com/tutorial/introduction-t-sne 1/11/25

https://github.com/JustGlowing/minisom/tree/master 1/11/25

https://medium.com/@soumallya160/a-beginners-guide-to-self-organizing-map-som-deep-learning-e0e30a7534e3 1/11/25