Exercise 1 – Classification

For the Digits dataset, I applied a number of classification methods, namely: K-Nearest Neighbor(K-NN), Support Vector Machines (SVM), Multi-Layer Perceptron (MLP). For each classifier I performed a 10-fold cross-validation and implemented hyper optimization with grid-search for tuning the hyperparameters of each classifier.

For the K-NN, I examined both hamming and euclidean distance metrics, run tests between 3, 5, 10, 15, 20 and 30 nearest neighbors. The best distance metric was euclidean, the best number of neighbors was 3 and got an accuracy of 0.983259.

For the SVM classifier I examined the linear and Rbf kernels, the values of 1, 10, 100 and 1000 for penalty parameters, the values 0.001 and 0.0001 for the kernel coefficient. The resulted accuracy was 0.991368, the best kernel was RBF, 0.001 coefficient and 10 as a penalty parameter.

For the MLP classifier I examined the number of hidden layers, in a range of 100, 150, 170 and 200. I tried values 0.001, 0.0001 and 0.00001 for the L2 regularization parameter. Initialization values for the Learning rate 0.001, 0.01. I tested different activation functions, namely relu, tanh, sigmoid and I chose to use Adam algorithm for optimization.
The resulted accuracy was 0.981951, the best number of hidden layers were 150, the best value for the hyperparameter of the L2 regularizer was 0.001 and the tanh seemingly gave the best performance on this dataset.

Classifier	K-NN	SVM	MLP
Accuracy	0.983259	0.991368	0.989951

Confusion matrices only for the test set after the cross-validation and the hyperparameter tuning.

```
[376   0   0   0   0   0   0   0   0   0]
[  0 389   0   0   0   0   0   0   0   0]
[  0   0 380   0   0   0   0   0   0   0]
[  0   0   0 389   0   0   0   0   0   0]
[  0   0   0   0 387   0   0   0   0   0]
[  0   0   0   0   0 376   0   0   0   0]
[  0   0   0   0   0   0 377   0   0   0]
[  0   0   0   0   0   0   0 387   0   0]
[  0   0   0   0   0   0   0   0 380   0]
[  0   0   0   0   0   0   0   0   0 382]
```

Confusion matrix for SVM-MLP

```
[374   0   0   0   1   0   1   0   0   0]
[  0 387   1   0   0   0   0   1   0   0]
[  0   0 380   0   0   0   0   0   0   0]
[  0   1   0 386   0   1   0   0   1   0]
[  0   0   0   0 385   0   1   1   0   0]
[  0   1   0   0   0 373   0   0   0   2]
[  0   2   0   0   0   0 375   0   0   0]
[  0   0   0   1   1   0   0 385   0   0]
[  0   6   0   0   0   0   0   0 373   1]
[  0   0   0   1   1   0   0   2   1 377]
```

Confusion matrix for K-NN

Considering the results and the fact that we frequently use the K-NN as a baseline method, and it achieves such a high accuracy, we can assume that the handwritten digits dataset is an easy classification problem.

Exercise 2 – Regression

At the Abalone dataset, I applied regression methods, namely Linear regression, Polynomial regression, the Lasso method, Multi-Layer Perceptron and Gaussian processes. For each regression method I performed a 10-fold cross-validation over the training data, using the Mean Square Error as a loss function and implemented hyper optimization with grid-search for tuning the hyperparameters of each classifier.

For the Linear regression which is a hyperparameter-free method, the model achieves a mean square error of 5.2175582866.

For the polynomial regression I tested the model only for the degrees 1, 2, 3, 4, 5 because higher degrees were computationally expensive and could not be performed. For the polynomial degree 1 the model achieves a mse of 5.2175582866, and for polynomials of higher degree the mse was 7.12282798036, 39.6915230822, 79502.8790361, 2519701985.67 respectively. I suppose that the steep increase of the mse is a result of the increasing complexity of the high degrees which may cause overfitting of the model. This could be solved with a regularizer.

Lasso regression is similar to linear regression adding an L1 regularization parameter. I examined the penalty parameter for which, I tried the values 0.00001, 0.0001, 0.001, 0.01, which resulted 5.10838825, 5.10552885, 5.09146143 and 5.29577172 respectively. 0.001 achieves the best mse. Lasso regression accomplishes a better performance than Linear regression.

For the MLPs I chose between relu, sigmoid and tanh activation functions, I tested the 0.001, 0.0001 and 0.00001 for the L2 regularization parameter. I choose initialization of learning rate between 0.001 and 0.001. The results were 4.59 mean square error, where the best activation function was relu, best number of hidden layers was 150 and the regularization parameter 0.001.

For the Gaussian processes, I used a Linear kernel where the best parameter was 1e-9 with an mean square error of 5.212635

Regression Method	Linear	Polynomial	Lasso	MLPs	Gaussian processes
Mean square error	5.2175582866	5.2175582866 (1^{st} degree)	5.10838825	4.59	5.212635

From the mse results we can clearly see that the MLP had the best performance on the dataset. Although, even after computing the mean absolute error (which was approximately 1.49 rings) for getting a more intuitive result of the error, we can say that we are far from having a good model that can automatically estimate the rings(age) of the snails.