Exercise Sheet 3 – Machine Learning

INF161 Autumn Semester 2020

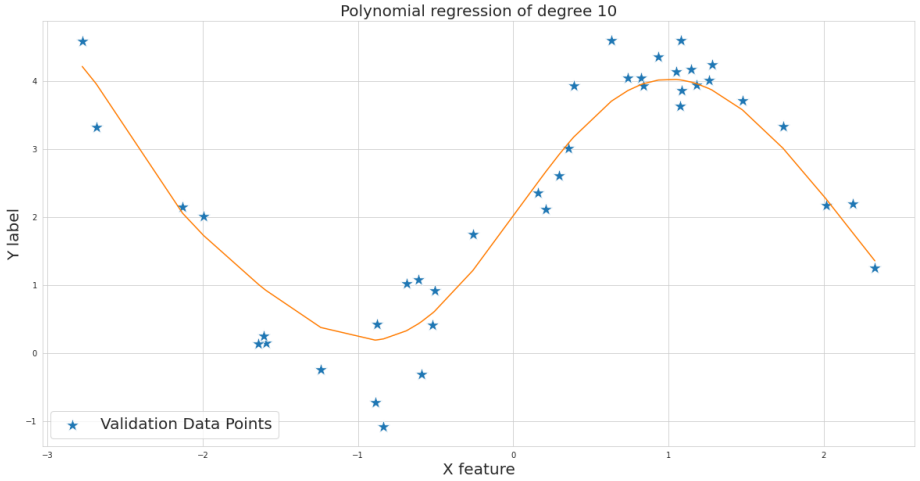
Exercise 1 *Regression*

- *Load the* ***"regression_nonlin.csv"*** *file into a dataframe.*
- *Divide the dataset into three sets of training, validation, and testing with corresponding proportions of* ***60%****,* ***20%****, and* ***20%****, respectively using the* ***train_test_split*** *function from the sklearn package. Set the initial random state to a specific initial seed (So that dataset is split into the same splits every time you run the notebook).*
- *Fit a linear regression model to the training data.*
- *Plot the training and validation datasets along with the regression line and compute the Mean Square Error on both datasets. Your figure should look like figure 1.*

Figure 1: Linear regression on the dataset

- *For each of the degrees [2, 5, 10, 20, 25], Fit a Polynomial Regression model of the chosen degree on the training dataset. Plot the results on the validation data along with the model prediction line like the example in figure 2 for each of the models.*

Figure 2: Example figure of Polinomial Regression on the dataset



- *Compute the Mean Square Error on training and validation datasets for each degree and compare it to simple linear model. Does any of your models overfit or underfit?*
- *How well does your best performing model on validation data, make predictions on the test dataset?*

Exercise 2 ***Classification***

- *Iris is a small dataset consisting of 150 vectors describing iris flowers, split into three different classes representing three species of the iris family. Each vector comes with a label (the name of the species) and a set of four features which are measurements of different parts of the flower. Those measurements tend to differ between the different species, thus it is possible for us to learn and evaluate a classifier from this dataset whose task will be to predict the species of an iris flower represented by aforementioned set of features. Load the Iris dataset directly from sklearn. You can alternatively download the dataset by clicking here.*

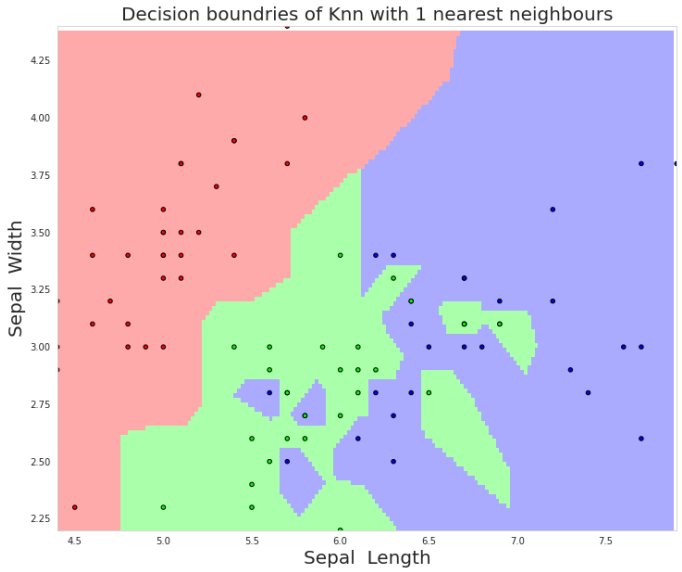
- *Store* ***the first two features (sepal length and sepal width)*** *in a matrix X. Also store the labels in a vector Y.*

- *Divide the dataset into three sets of training, validation, and testing with corresponding proportions of* ***60%****,* ***20%****, and* ***20%****, respectively using the* ***train_test_split*** *function from the sklearn package. Set the initial random state to a specific initial seed (So that dataset is split into the same splits every time you run the notebook).*

- *Perform a k-NN classification of your dataset for each k in 1, 5, 10, 20, 30. You can for instance use the* ***KNeighborsClassifier*** *class from sklearn.*

- *Plot the decision boundaries with the training points overlayed for every k (since there are three classes, you will need three different colors); the axis of the two selected features must be apparent in your decision boundaries plots. You can modify the code in this example. Your plots should look like the example below:*

Figure 3: Example figure of K-NN Classifier decision boundries on the Iris dataset



- *Plot the curves representing the training and validation accuracy as a function of different K.*
- *How well does your best performing model on validation data, make predictions on the test dataset?*