Machine learning: Assignment 1

Mathias Mellemstuen

26 September 2021

1 Content

The content of this report will describe the functions, libraries and procedures used to complete the two tasks in this assignment. Hence, this report will be divided into these sections respectively: Libraries, Functions, Procedures.

2 Libraries

The library that were used in this assignment are the following: Numpy, Pandas, Matplotlib, sklearn.

2.1 NumPy

The NumPy library is a library that contains mathematical functionality and is a good choice for doing data manipulation. The matrix and array functionality of NumPy is especially used in this assignment.

2.2 Pandas

Pandas library contains data analysis and manipulation tools, much like NumPy. Pandas also has a file reader / parser function. We are only using this function from the Pands library in this assignment.

2.3 Matplotlib

Matplotlib is a library that allows you to plot different graphs / plots on a canvas. Very powerful in terms of visualization.

2.4 Sklearn

Sklearn is a machine learning library for python. This library has functionality for almost all kinds of machine learning techniques. We are only using KNeighboursClassifier, train_test_split and LinearRegression from this library in this assignment.

3 Functions

Keep in mind that this report is not considering all functions used in the assignment. Only the most important and necessary functions to complete the assignment. The functions that is considered in this report are the following : read_csv, train_test_split, LinearRegression.fit, KNeighboursClassifier.fit, KNeighboursClassifier.predict, plot, scatter and accuracy_score.

3.1 read_csv

```
import pandas as pd
pd.read_csv("data.csv")
```

This function comes from the Pandas library. The read_csv function reads a file in the csv format. Then it puts the data in a Pandas object called DataFrame. This object can be used as a multidimentional array in python. In this case, 2-dimentional.

3.2 train_test_split

```
from sklearn import linear_model
trainingX, testX, trainingY, testY = train_test_split(X, y, test_size)
```

This function splits a dataset into a training set and a testing set. You can specify the size of the sets with the test_size hyperparameter. The function will divide the dataset into two sets randomly. This is to ensure that each sample is a random sample of the original dataset. The function will output the two new sets.

3.3 LinearRegression.fit and KNeighboursClassifier.fit

```
from sklearn import linear_model
linearRegression = linear_model.LinearRegression()
linearRegression.fit(trainingX, trainingY)
```

This function comes from the sklearn library. The fit function fits a linear model to the input training data. This will run the linear regression training algorithm and produce the coefficient m and the constant b in a linear function $f(x) = mx + b$.

```
from sklearn.neighbors import KNeighborsClassifier
neighbours = KNeighborsClassifier()
neighbours.fit(trainingX, trainingY)
```

The same function will be used for training with the KNeighboursClassifier algorithm. This makes it possible to call KNeighboursClassifier.predict in the future.

3.4 KNeighboursClassifier.predict

```
from sklearn.neighbors import KNeighborsClassifier
neighbours = KNeighborsClassifier()
neighbours.fit(trainingX, trainingY)
predictionY = neighbours.predict(testX)
```

This function comes from the sklearn library. The predict function can predict the y value of some x value on a KNeighborsClassifier.

3.5 plot

```
import matplotlib.pyplot as plot
plot.plot([x1, x2], [y1, y2])
```

This function comes from the matplotlib library. The function is used to visualize and draw a line between two points in 2 dimentional space. In this context, it will be used to draw the linear function $f(x) = mx + b$ with the coefficient and constant produced by the LinearRegression.fit function.

3.6 scatter

```
import matplotlib.pyplot as plot
plot.scatter(x, y)
```

This function comes from the matplotlib library. It can be used to create a 2 dimentional scatter plot of the test and training data.

3.7 accuracy_score

```
from sklearn import metrics
metrics.accuracy_score(testY, predictionY)
```

The accuracy_score function is used to calculate the difference between two datasets. Ouputs the difference in percentage (i.e. 0 - 1).

4 Procedures

Both of the tasks in this assignment had the same procedure to solve. The procedure can be explained like this algorithm:

- Read and parse the data file with Pandas.
- Splitting the data in two parts: x and y.
- Split the data in a training and testing set.
- Using the fit function of either LinearRegression or KNeighborsClassifier.
- For LinearRegression; Plot the line. For KNeighborsClassifier; Predict the test values and compare the prediction to the test values.