This practical serves to introduce some fundamental techniques in the Python programming language to prepare you for this course. Along with this document you have been provided with skeleton code which contains demarcated areas, numbered accordingly, in which to code answers to the tasks below. Document your answers by including the relevant code in your report together with detailed comments (in the code).
Die doel van hierdie prakties is om 'n paar grondliggende tegnieke van die Python programmeringstaal voor te stel as voorbereiding op die inhoud van die kursus. Hierdie dokument word vergesel deur 'n program raamwerk met genommerde plekke waar u kode moet byvoeg. Dokumenteer u antwoorde deur die relevante kode by u verslag in te sluit, vergesel deur gedetaileerde kommentaar (in die kode).

Tasks to be completed:
Take wat voltooi moet word:

1. Firstly import the libraries `numpy` and `matplotlib`.
   Eerstens moet die biblioteke ***`numpy`*** *en* ***`matplotlib`*** *ingevoer word.*

2. Then read in the MNIST dataset from files `data/train.csv` and `data/test.csv` and store as numpy variables with shapes:
   Lees dan die MNIST datastel van die lêers ***`data/train.csv`*** *en* ***`data/test.csv`*** *en stoor as numpy veranderlikes met vorm:*

   - Train : (60000×784)
   - Test : (10000×784)
   - Train targets : (60000×1)
   - Test targets : (10000×1)

* **Hint**: Open data/train.csv to see how the data is stored. The MNIST data set consists of 28×28 pixel images of handwritten digits where each pixel has a value between 0 and 255.
Wenk*: Kyk na data/train.csv om te sien hoe die data gestoor word. Die MNIST datastel bevat 28×28 pixel beelde van handgeskrewe syfers met elke piuxel 'n waarde tussen 0 en 255.*

3. Print the shapes of the training and test data and targets to verify that task 2 has been correctly achieved.
   Druk die vorms van die afrig- (train) en toets- (test) data en teikens om te verifeer dat Taak 2 reg afgehandel is.

4. Normalise the data by means of linear scaling to lie in the range $[0, 1]$. Verify that the normalisation is successful.
   Normaliseer die data deur middel van lineêre skalering om in die interval $[0, 1]$ *te lê. Verifeer dat die normaliserting suksesvol was.*

5. Using `matplotlib` plot the first example from the training set.
   Gebruik ***`matplotlib`*** *om die eerste item in die afrigstel te plot.*

* **Hint**: You may have to utilise `np.reshape` to produce the correct shape to plot the image.
Wenk*: Dalk moet u* ***`np.reshape`*** *gebruik om die beeld die regte geometrie te gee.*

6. Load in weights from a logistic regression model from the file `data/weights.csv`, and store the weights as a numpy array called `weights`.
Lees die gewigte vir die logistiese regressie model van die lêer `data/weights.csv`, en stoor die gewigte as 'n numpy array met die naam `weights`.

* **Hint**: The shape of the array should be (785×10).
***Wenk**: die geometrie van die array behoort (785×10) te wees.*

7. Uncomment the code in Section 7, to load a `SoftmaxRegression`[1] model and call pythons `help` function to see how to correctly utilise the class's `load` function.
Ontkommenteer die kode in Afdeling 7, om die a `SoftmaxRegression`[2] model te lees. Gebruik die python `help` funksie om te sien hoe om die klas se `load` fubksie te gebruik.

   - Utilising the information from `help`, pass the loaded weights to the model.
   Deur `help` te gebruik, voer die gewigte na die model.
   - Use the model to make a prediction for an example from the test set (by calling `model(example)`). There is an interesting example is at index 7.
   Gebruik nou die model om 'n voorspelling te maak vir 'n voorbeels uit die toesstel (deur `model(example)` te roep). Daar is 'n interresante voorbeeld by indeks 7.

   * **Hint**: When feeding input to the model, make sure the shape of the input is a row vector (1×784) and the result of taking a row from the X matrix whose shape is (N×D).
   ***Wenk**: Maak seker dat die intree wat vir die model gevoer word 'n (1×784) ry vektor is wat verkry word deur 'n ry van die (N×D) matriks X te kopieer.*

   * **Note**: You are not expected to understand the regression model for this practical, only to apply it.
   ***Nota**: Daar word nie in hierdie prakties van u verwag om die werking van die regressiemodel te verstaan nie, u moet dit net toepas.*

8. Plot a bar graph of the model probabilities computed for each possible output digit from the test set sample in Question 7 (index 7).
Plot 'n staafgrafiek van die waarskynlikhede wat die model bereken vir elke moonlike syfer vir die toetsmonster van die toetsstel soos gebruik in Vraag 7 (indeks 7).

9. Define your own python function to calculate the model accuracy, and use it to calculate the average accuracy on the test set.
Skryf u eie python funksie om die model se akkuraatheid te bereken, en gebruik dit om die gemiddelde akuraatheid op die toetsstel te bepaal.

   * **Hint**: The classification result is determined by finding the index of the array element of model outputs containing the maximum probability. This can be accomplished using numpy's `argmax()` function.
   ***Wenk**: Die klassifikasieresultaat word bepaal deur die indeks van die indeks van die modeluitsette te vind wat die maksimum waarskynlikheid bevat. Dit kan met die numpy funksie `argmax()` gedoen word*

[1] `SoftmaxRegression` receives as input a (1×784) vector of image pixels and outputs a (1×10) vector containing the probability that the input vector is an image of the digit 0,1,...,9

[2] `SoftmaxRegression` neem 'n (1×784) vektor beeld pixels as intree en gee 'n (1×10) vektor van waarsknlikhede as uittree wat die waarskynlihed dat die beeldpixsels aan die syfer 0,1,...9 behoort verteenwoordig.