Istanbul Medipol University
School of Engineering and
Natural Sciences
Computer Engineering

2021 – Assignment #4
Advanced Programming
Assignment Date: 22.05.2021
Due Date: 31.05.2021

Programming Assignment #4

Question I (20 pts):

(Regression) Load the Boston dataset from sklearn. Split the dataset into training and testing parts. Use ridge regression (linear_model.Ridge) to predict the target values. Try with both the original attributes and polynomial features (preprocessing.PolynomialFeatures(2,interaction_only=True)). Determine the best regularization coefficient (alpha) in each case. Plot true value vs. predicted values. Finally, decide on a single attribute, try to predict the values using a single attribute.

Question II (20 pts):

(Classification) Load the MNIST dataset from sklearn. Split the dataset into training and testing parts. Use the KNN classifier. Try different K values and analyze the performances. Write your analyzes as a comment in the code.

Question III (30 pts):

Keras's bundled CIFAR10 dataset contains 32-by-32 color images labeled in 10 categories with 50,000 images for training and 10,000 for testing. Using the convnet techniques you learned in the MNIST case study, build, train and evaluate a convnet for CIFAR10. How accurate are the predictions compared to those you experienced with MNIST given in the textbook?

Question IV (30 pts):

a. A great source of plain text files is the collection of over 57,000 free e-books at Project Gutenberg:

   https://www.gutenberg.org

   Download the text-file version of *Pride and Prejudice* from Project Gutenberg

   https://www.gutenberg.org/ebooks/1343

   Create a script that reads *Pride and Prejudice* from a text file. Produce statistics about the book, including the total word count, the total character count, the average word length, the average sentence length, a word distribution containing frequency counts of all words, and the top 10 longest words. In the "Natural Language Processing (NLP)" chapter, you'll find lots of more sophisticated techniques for analyzing and comparing such texts.

   Each Project Gutenberg e-book begins and ends with some additional text, such as licensing information, which is not part of the e-book itself. You may want to remove that text from your copy of the book before analyzing its text.

b. A word cloud visualizes words, displaying more frequently occurring words in larger fonts. In this exercise, you'll create a word cloud that visualizes the top 200 words in *Pride and Prejudice*. You'll use the open-source wordcloud module's[15] WordCloud class to generate a word cloud with just a few lines of code.
https://github.com/amueller/word_cloud.
To install wordcloud, open your Anaconda Prompt (Windows), Terminal (macOS/Linux) or shell (Linux) and enter the command:

```
conda install -c conda-forge wordcloud
```

You create and configure a WordCloud object as follows:

```
from wordcloud import WordCloud
wordcloud = WordCloud(colormap='prism', background_color='white')
```

Using the techniques from the previous exercise, create a frequencies dictionary containing the frequencies of the top-200 words in *Pride and Prejudice*. Then execute the following statements to generate a rectangular word cloud and save its image to a file on disk:

```
wordcloud = wordcloud.fit_words(frequencies)
wordcloud = wordcloud.to_file('PrideAndPrejudice.png')
```

You can then double-click the PrideAndPrejudice.png image file on your system to view it.