Problem Set 8

Dario Loprete

November 5, 2024

GitHub link

https://github.com/dariolop76/phys-ga2000

1 Problem 1

In this problem, we have data from a survey which consists of a "yes or no" answer to the question "Do you recognize the phrase 'Be Kind, Rewind', and know what it means?". We assume that the distribution function of our data can be modeled by the logistic function

$$p(x) = \frac{1}{1 + e^{-\beta_0 - \beta_1 x}}, \tag{1}$$

and want to find the values of the parameters β_0 and β_1 that provide the best fit. We do this by maximizing the likelihood

$$L(\beta_0, \beta_1) = \prod_{\text{s in } y_i=1} p(x_i) \prod_{\text{s in } y_i=0} (1 - p(x_i)), \tag{2}$$

where (x_i, y_i) is our data, with x_i being the age of the respondent and y_i being either 0 (answer: "no") or 1 (answer: "yes"), and s is a sample (namely a specific value for the pair (x_i, y_i)). We can rewrite the likelihood as follows:

$$L(\beta_0, \beta_1) = \prod_{i=1}^{N} p(x_i)^{y_i} (1 - p(x_i))^{1-y_i}, \tag{3}$$

where N is the dimension of our sample of data. Maximizing the likelihood is equivalent to minimizing the function $-\log(L)$, which is equal to

$$-\log(L) = -\sum_{i=1}^{N} \Big[y_i \log p(x_i) + (1 - y_i) \log(1 - p(x_i)) \Big]. \tag{4}$$

In the code `prob_1.py`, we perform this minimization procedure using the package `scipy.optimize.minimize`, thus obtaining the optimal values for the parameteres β_0 and β_1.

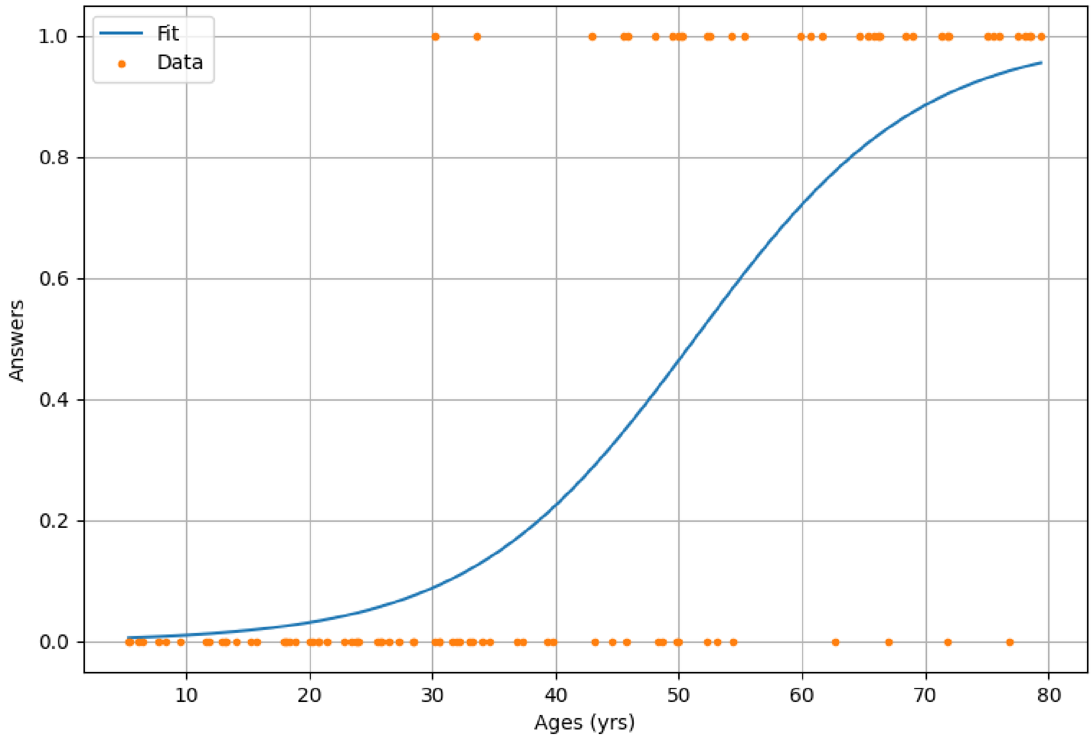

Figure 1: Plot of the data and its fit.

Additionally, we compute the covariance matrix, being the inverse of the hessian matrix for the negative log-likelihood. Figure 1 shows the fit of the data. The results are reported below.

Parameters:
$\beta_0 = -5.6202$,
$\beta_1 = 0.1096$.

Covariance matrix:

$$\begin{bmatrix} 1.1186 & -2.1119 \times 10^{-2} \\ -2.1119 \times 10^{-2} & 4.3467 \times 10^{-4} \end{bmatrix}$$

Errors:
$\sigma_{\beta_0} = 1.058$,
$\sigma_{\beta_1} = 0.021$.

Maximum likelihood value:
$L_{max} = 8.3 \times 10^{-16}$

Final estimate of the parameters
$\beta_0 = -6 \pm 1$,
$\beta_1 = 0.11 \pm 0.02$.

2 Problem 2

In this problem, we analyze the waveform of a single note played by a piano and a trumpet and study its spectrum by means of the Fourier transform. Figures 2 and 3 show the waveform recorded from these two instruments.

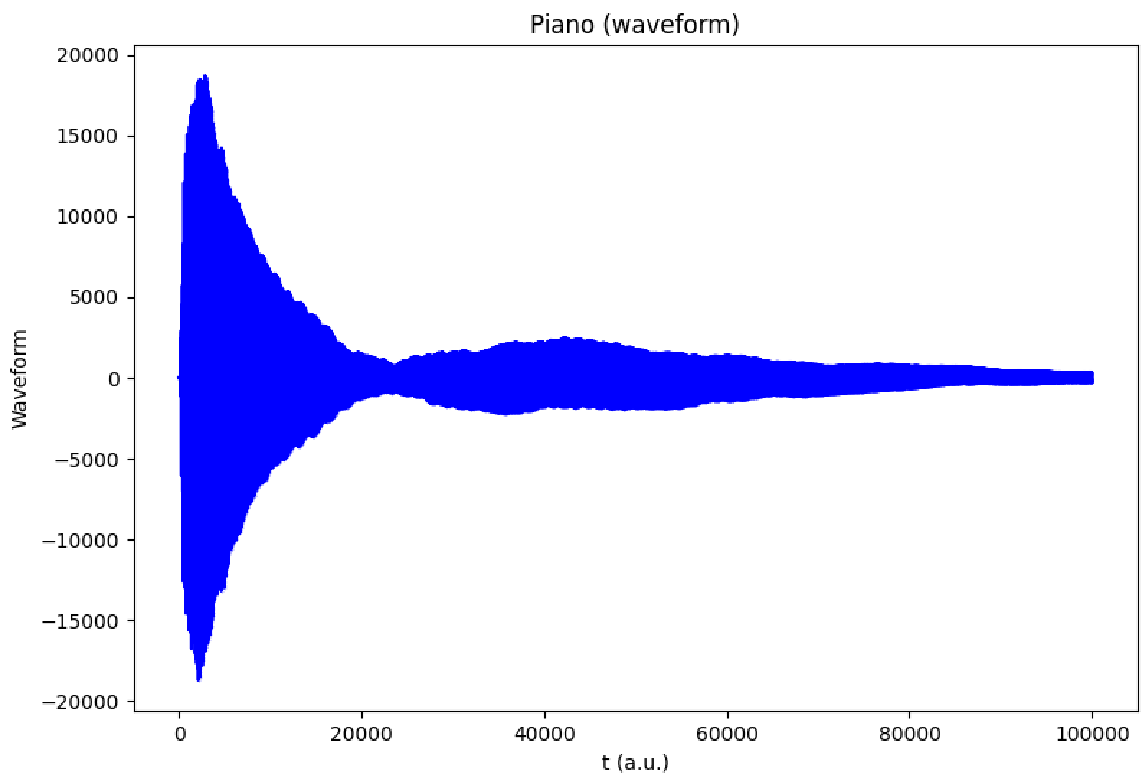

Figure 2: Waveform of a single note played by a piano.

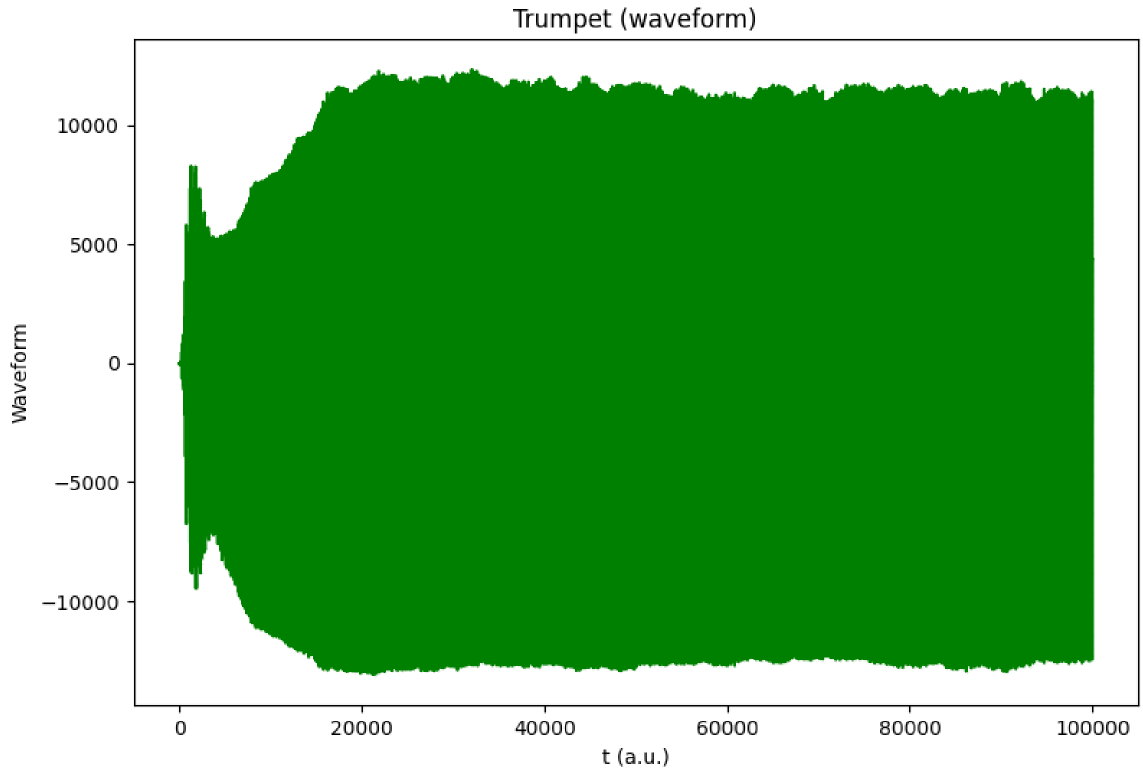

Figure 3: Waveform of a single note played by a trumpet.

We use the package `numpy.fft` to perform a Fast Fourier Transform of the data and plot the magnitudes of the first 10000 coefficients, (Figures 4 and 5), which are defined as

$$c_k = \sum_{n=0}^{N-1} y_n e^{-i\frac{2\pi kn}{N}}, \tag{5}$$

where y_n is our data, N is the dimension of the sample and k is proportional to the frequency of the wave.

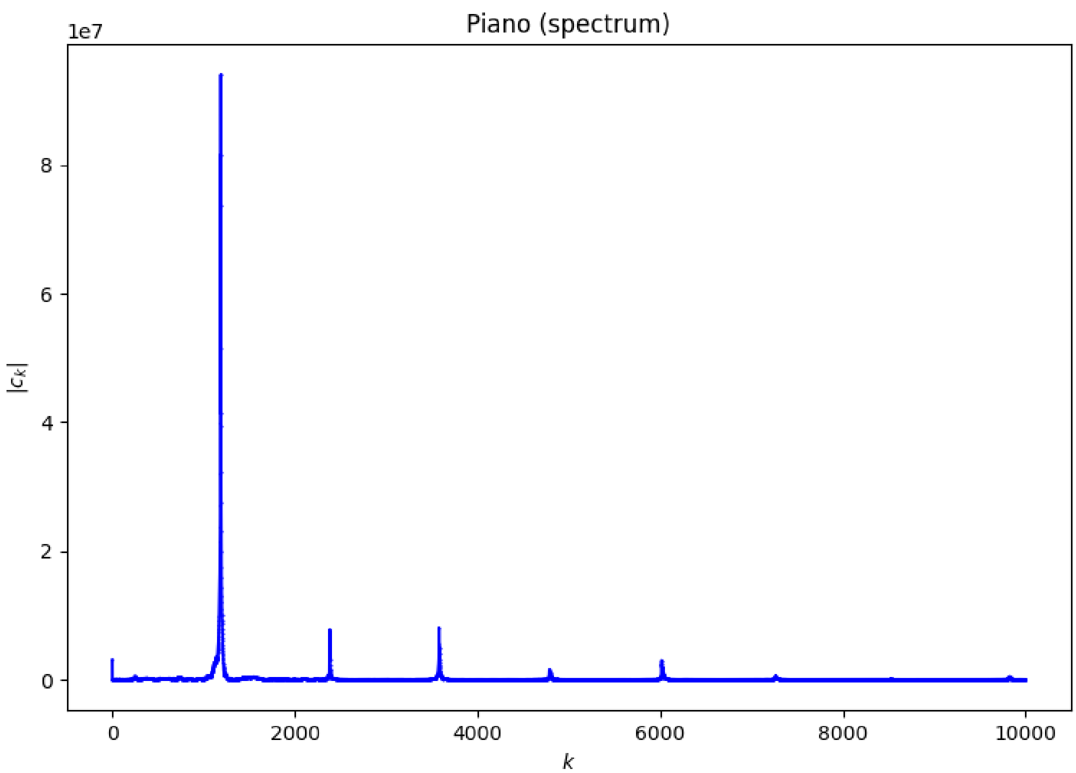

Figure 4: Fourier transform of the signal from the piano.

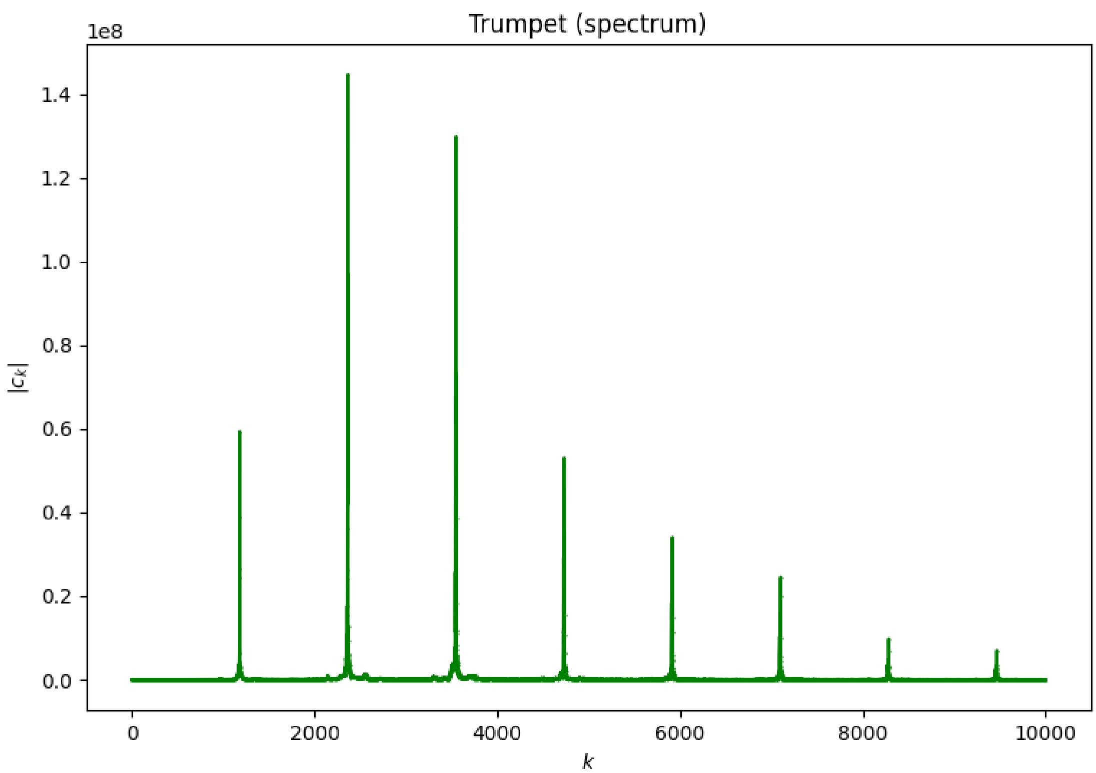

Figure 5: Fourier transform of the signal from the piano.

The first spike corresponds to the frequency of the note these instruments were playing, the fundamental frequency, while the remaining spikes correspond to overtones, namely multiples of the fundamental frequency. Knowing that the waveforms were recorder at a rate of 44100 samples per second, we can rescale the k axis and show the coefficients in terms of the frequency. Figure 6 focuses on the initial part of the spectrum of both waveforms. The frequency of the first spike is ~ 525 Hz, which corresponds to the note C5.

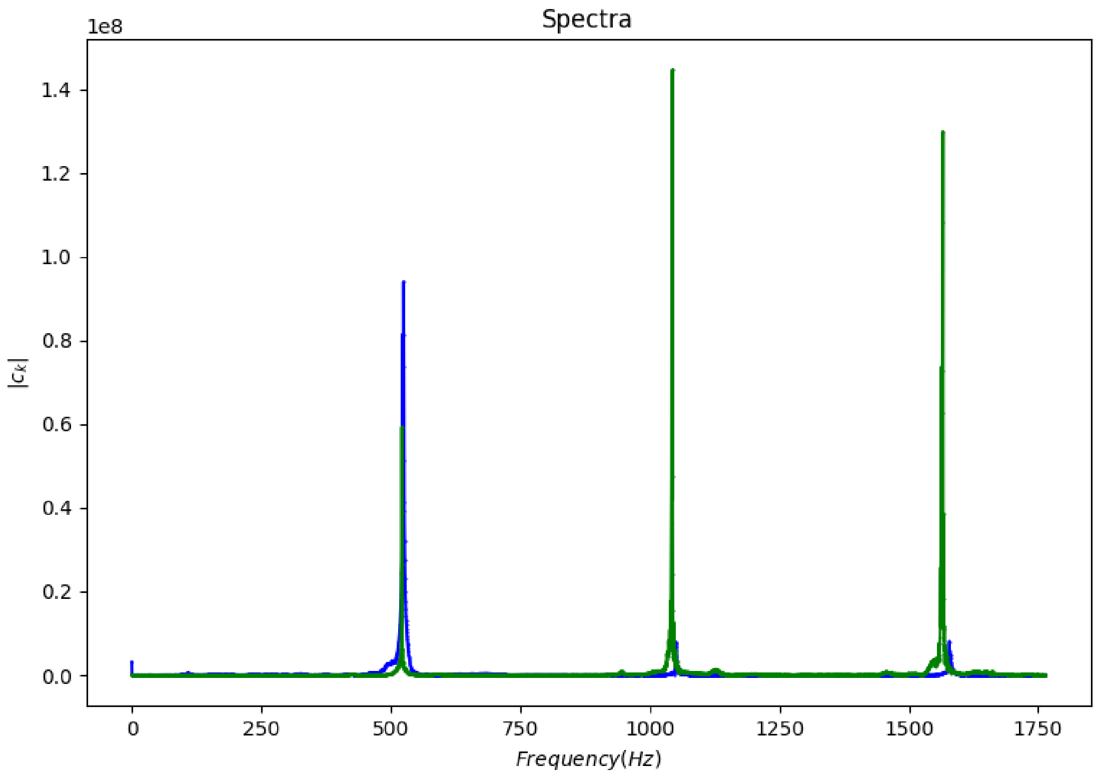

Figure 6: Fourier coefficients for the signals from both instruments VS frequency.

3 Problem 3

In this problem, we perform a Fast Fourier Transform (FFT) analysis as in Problem 2, this time on data regarding the daily closing value for each business day from late 2006 until the end of 2010 of the Dow Jones Industrial Average. Figure 7 shows the plot of the data.

We compute the coefficients and take the inverse FFT of the first 10% of the obtained coeffiecients. We then do the same with the first 2% of the coefficients. Figures 8 and 9 show the reconstructed data compared to the original data.

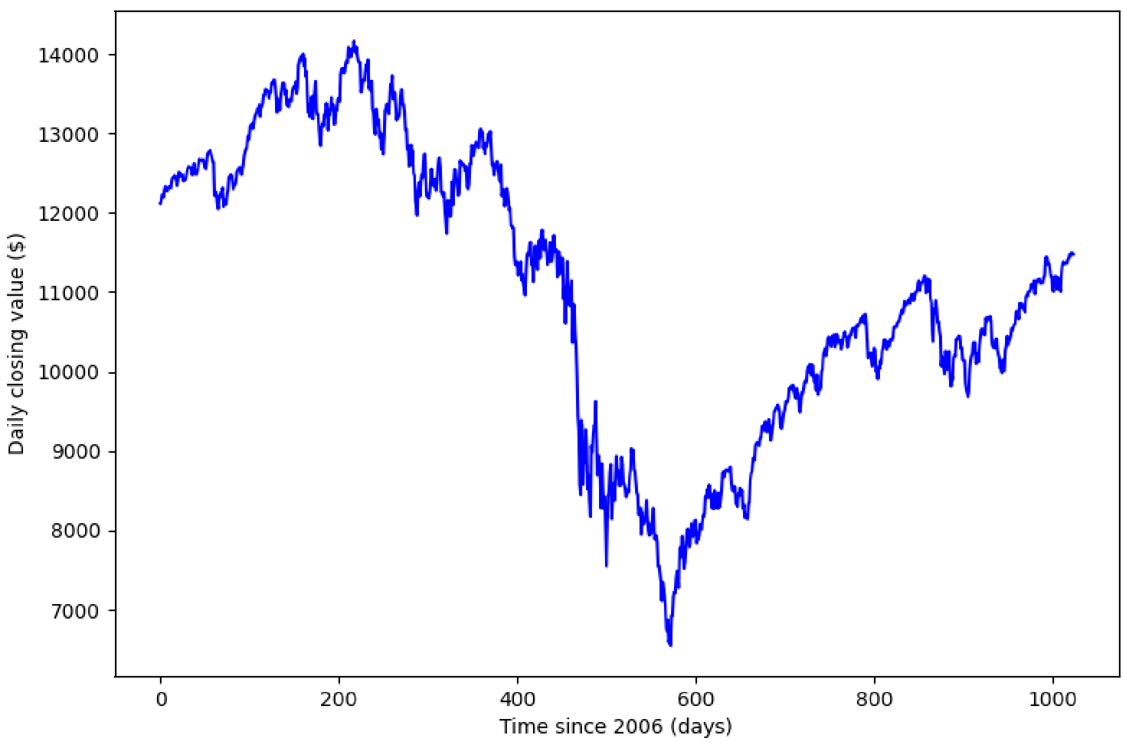

Figure 7: Data of the daily closing value of the Dow Jones Industrial Average.

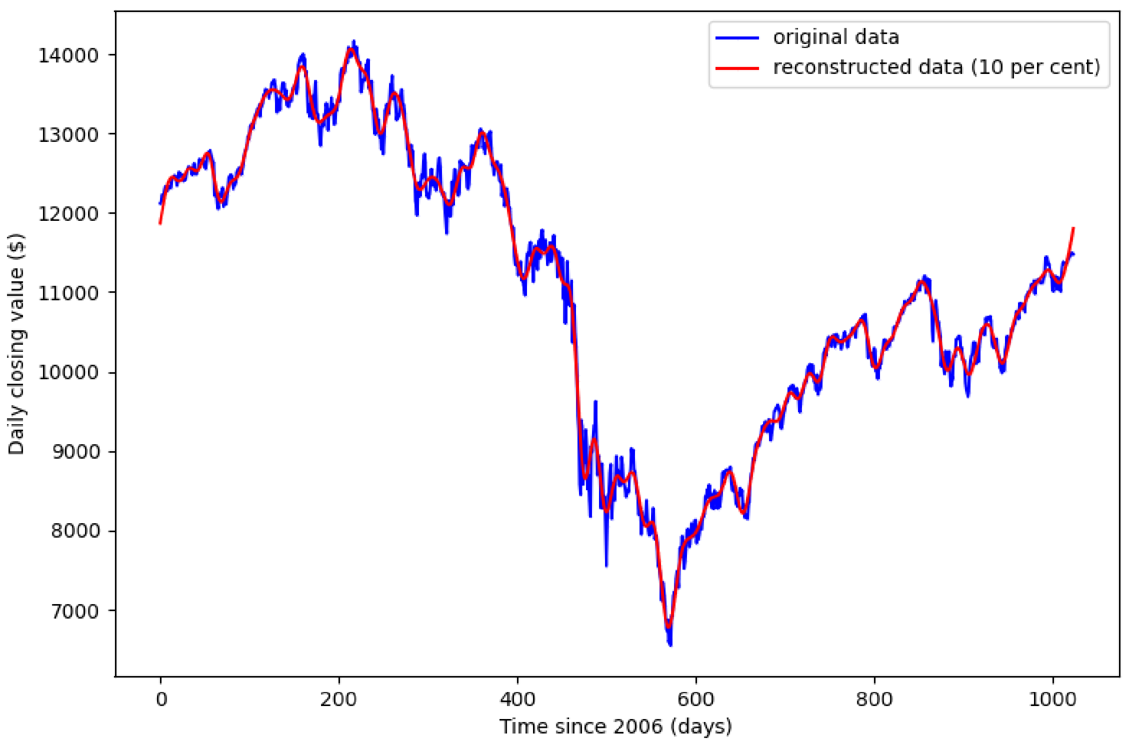

Figure 8: Original data compared to reconstructed data keeping 10 % of the Fourier coefficients.

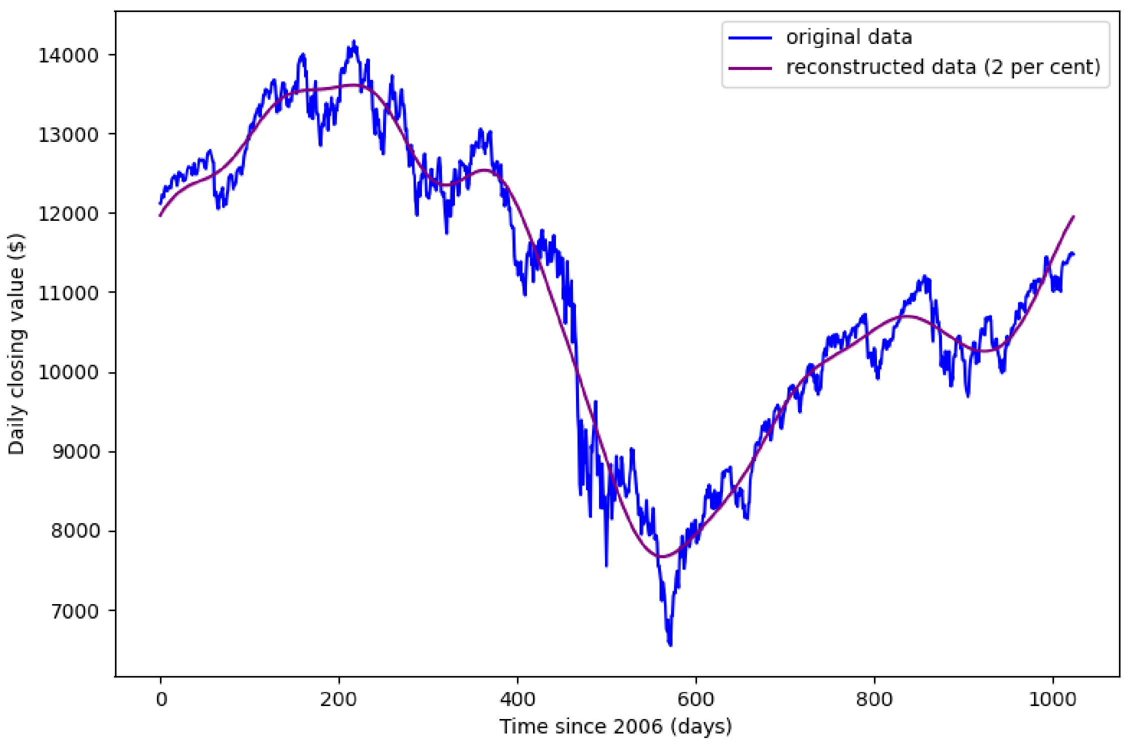

Figure 9: Original data compared to reconstructed data keeping 2 % of the Fourier coefficients.

As we can see form these plots, when we set part of the Fourier coefficients to zero, we lose information about the data. Indeed the reconstructed signal will be similar but not exactly equal to our original data. Additionally, by comparing the two plots, it is clear that the more coefficients we retain, the more accurate the reconstruction becomes.