

# PS8

Shihong Pan

<https://github.com/PSH-hub24/phys-ga2000>

November 5, 2024

## Q1

Let  $\vec{x}$  denotes the age array and  $\vec{y}$  denotes the respond array. Let  $B_i := -(\beta_0 + \beta_1 x_i)$ . The negative log likelihood  $l(\vec{x}) = -\sum_i [y_i \ln p(x_i) + (1 - y_i) \ln(1 - p(x_i))]$  can be written as

$$l(x) = -\sum_i y_i \ln\left(\frac{1}{1 + e^{B_i}} \frac{1 + e^{B_i}}{e^{B_i}}\right) + \ln\left(\frac{e^{B_i}}{1 + e^{B_i}}\right) = -\sum_i [(1 - y_i)B_i - \ln(1 + e^{B_i})] \quad (1)$$

Note that this expression avoids taking the log of zero since  $1 + e^{B_i} \neq 0$ . Then I minimized  $l(\vec{x})$  using `scipy.optimize.minimize` (with `jax.grad` to get the gradient as part of the input of `minimize`). The covariance matrix was obtained by taking the inverse of the Hessian (`jax.jacfwd(jax.grad(f))`) of  $l(\vec{x})$ , and the formal errors were obtained by taking square roots of diagonal entries of the covariance matrix. Fig 1 shows these values. Fig 2 plots the logistic model and the survey data together. It makes sense as the growth in the probability of answering yes shown by the model matches the change in the density of the black dots along the age axis.

## Q2

- a) I used `numpy.fft.rfft` to calculate the DFT of the two given files. Fig 3, 4 plot the waveforms, while Fig 5, 6 plot the magnitudes of the first 10000 coefficients of their DFT. From the plots of Fourier coefficients, it can be deduced that while the fundamental frequency of both instruments (the first major peak) is the same, trumpet produces sound with stronger harmonics (shown by the subsequent higher peaks with equal spacing, unlike the single dominant peak in the piano case).
- b) Let  $f_s = 44100$  be the sampling rate. The actual frequency corresponds to the  $k$ -th coefficient is  $f_k = k f_s / N$  where  $N$  is the number of samples. The calculation in the codes shows that, the fundamental frequency, which is the frequency represented by the first dominant peak, is 524.79Hz. The closest musical note I found is note C in Octave 5, 523.25Hz.

### Q3

- a) Fig 7 plots dow.txt
- b) One-line code, `dowFFT = np.fft.rfft(dow)`.
- c) `N = dowFFT.size`  
`dowFFT[int(N/10):] = 0`
- d) Fig 8 plots the new dow data. I see that the new curve is slightly smoother because only the first 10% of the Fourier coefficients are left, i.e., the high-frequency components are eliminated when the corresponding Fourier coefficients are set to 0.
- e) Fig 9 plots the new dow data. This time only the first 2% is left, the new curve is much smoother as only the low-frequency components are presented.

```
WARNING:absl:No GPU/TPU found, falling back to CPU. (Set TF_CPP_MIN_LOG_LEVEL=0 and rerun for more info.)
Maximum likelihood values of beta0 and beta1:
[-5.62027395  0.10956337]
Number of function evaluations: 73
Number of iterations: 10
Covariance matrix:
[[ 1.1186510e+00 -2.1119379e-02]
 [-2.1119379e-02  4.3467587e-04]]
Formal errors:
[1.057663  0.02084888]
[Finished in 5.3s]
```

Figure 1: The maximum likelihood values, covariance matrix, and formal errors of  $\beta_0$  and  $\beta_1$ .

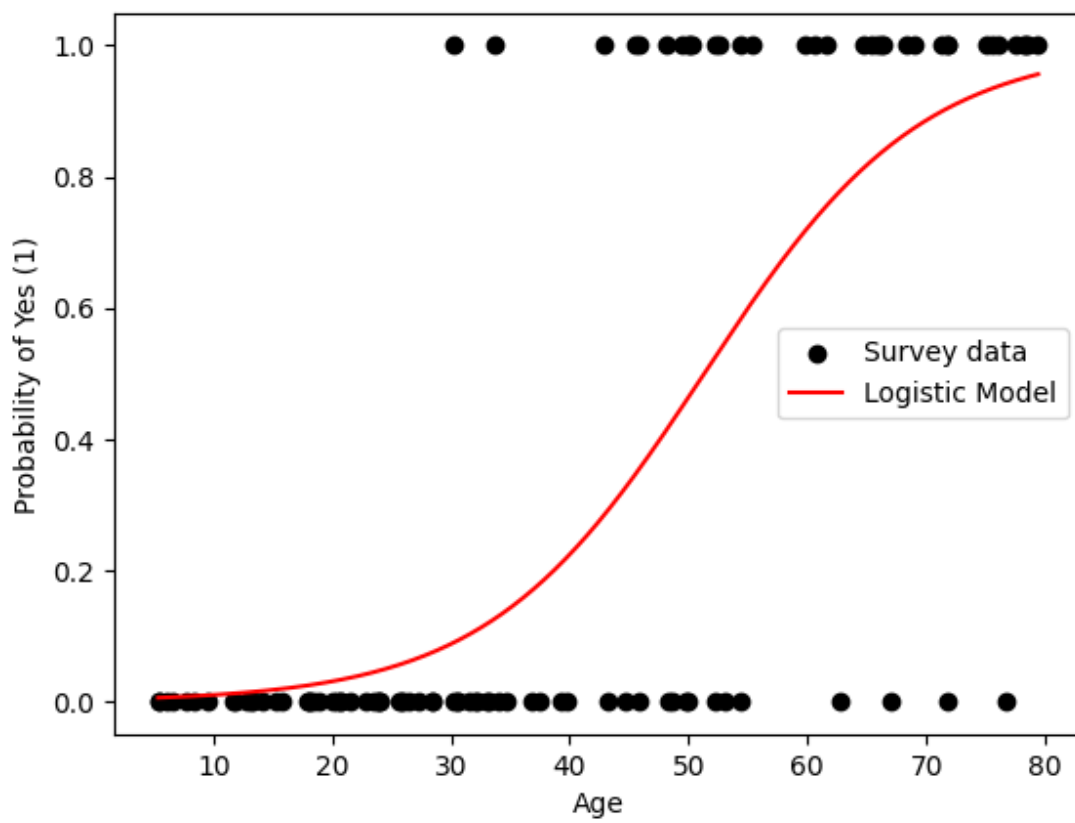


Figure 2: Plot the logistic model and the survey data on the same plot.

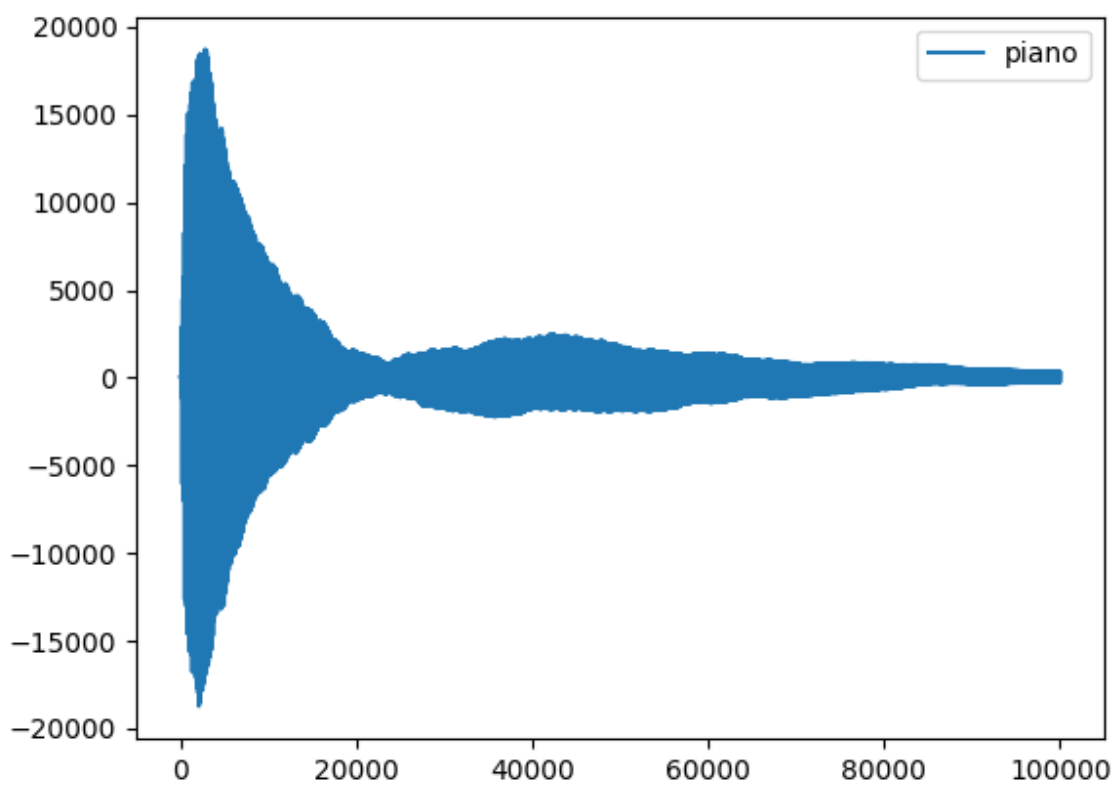


Figure 3: Waveform of piano.txt.

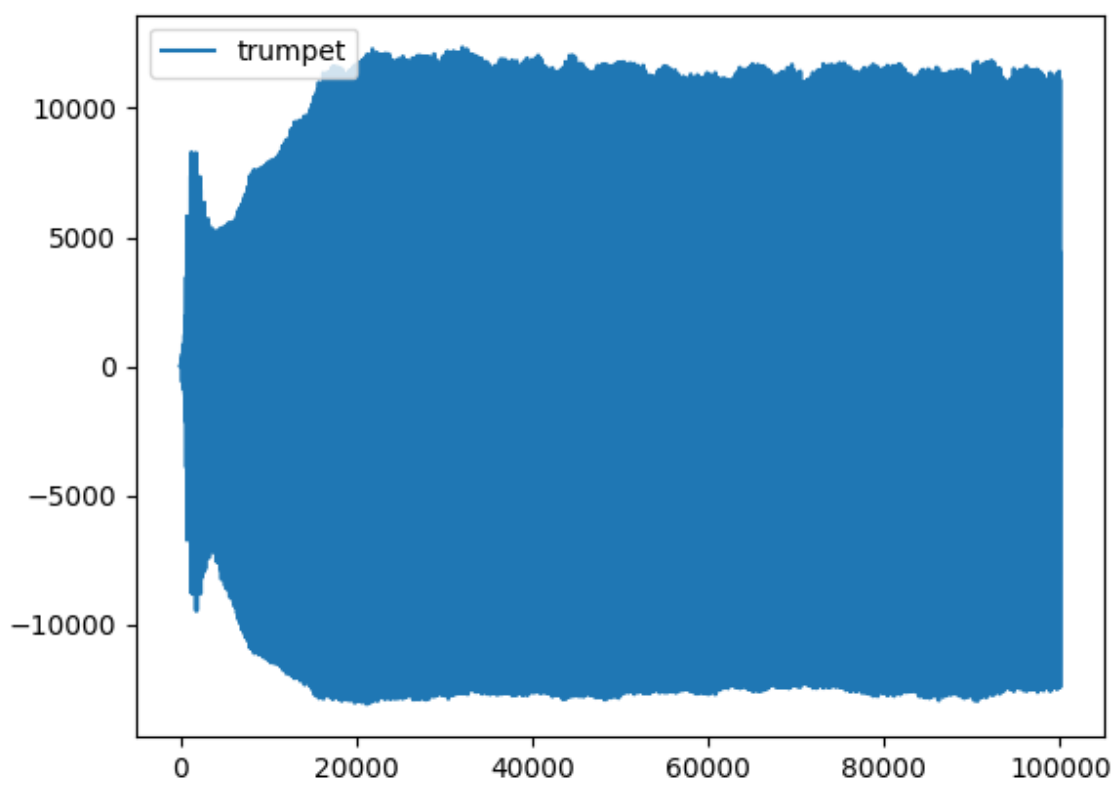


Figure 4: Waveform of trumpet.txt.

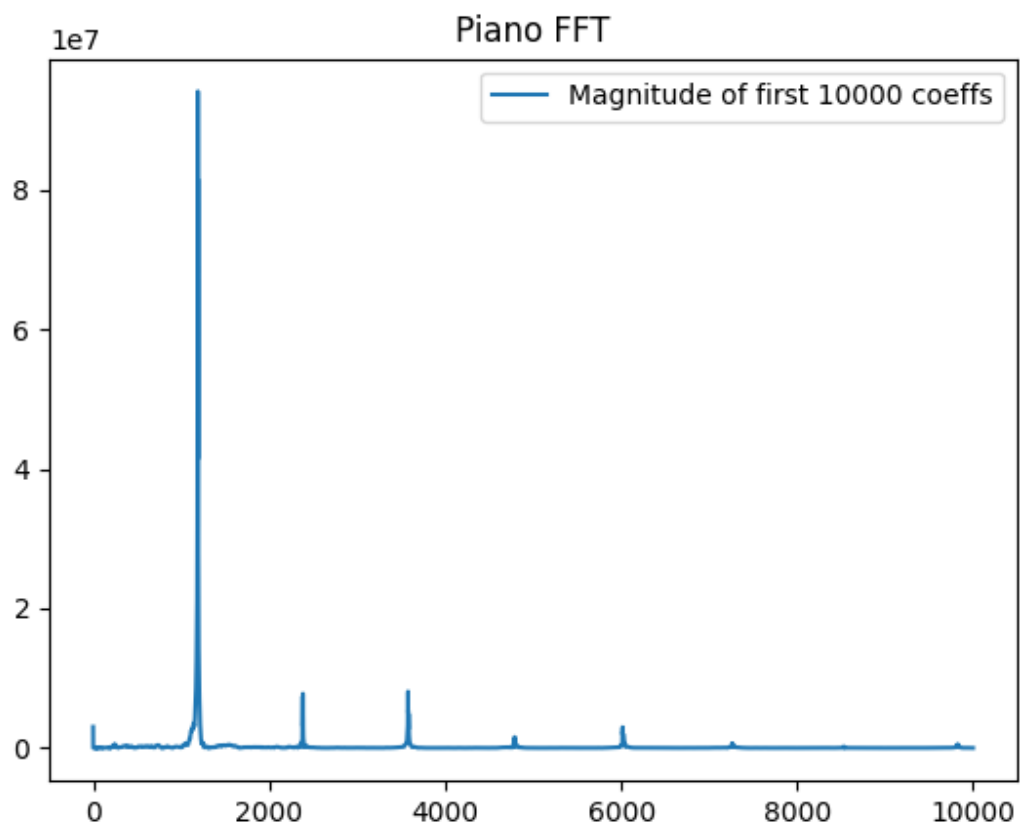


Figure 5: Plot the logistic model and the survey data on the same plot.

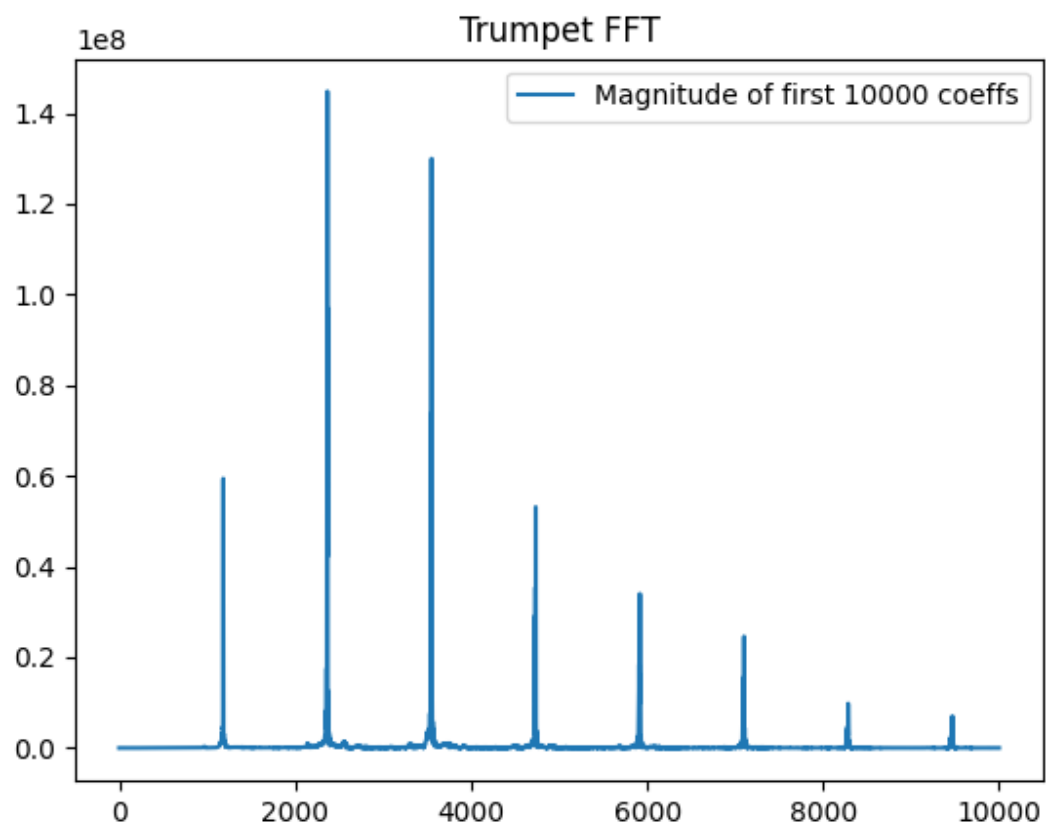


Figure 6: Plot the logistic model and the survey data on the same plot.

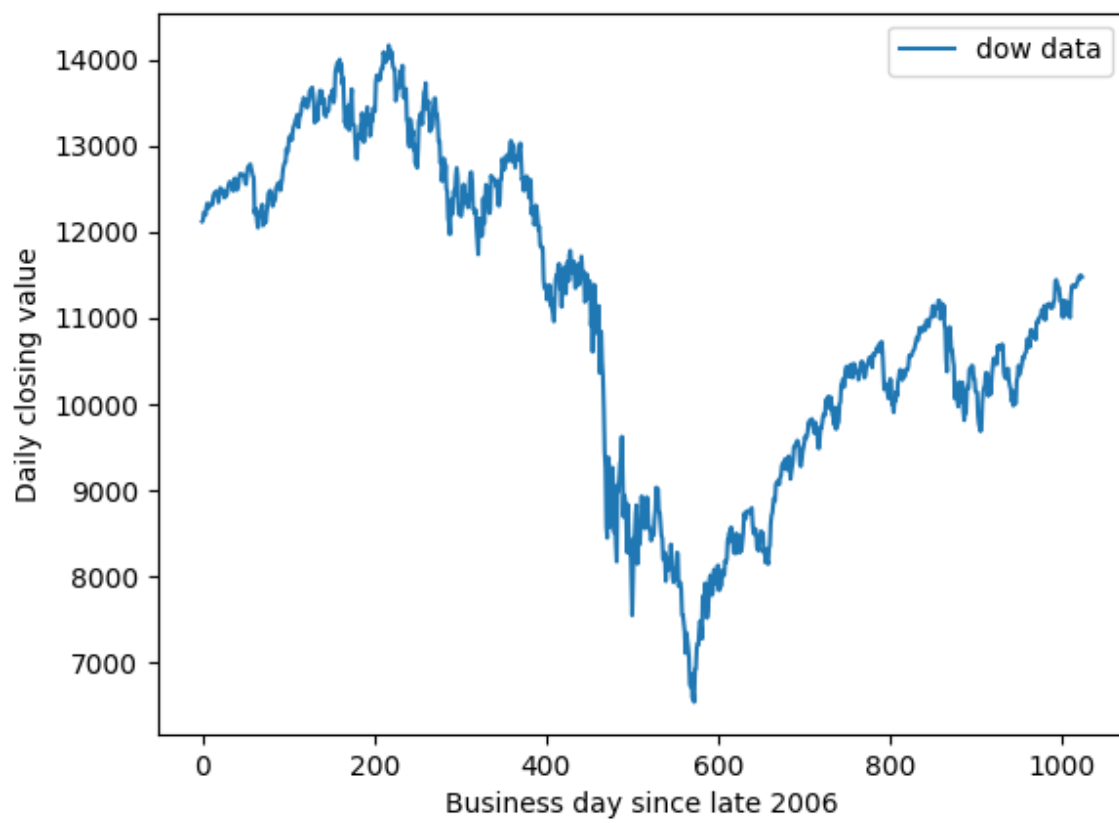


Figure 7: Plot dow.txt.



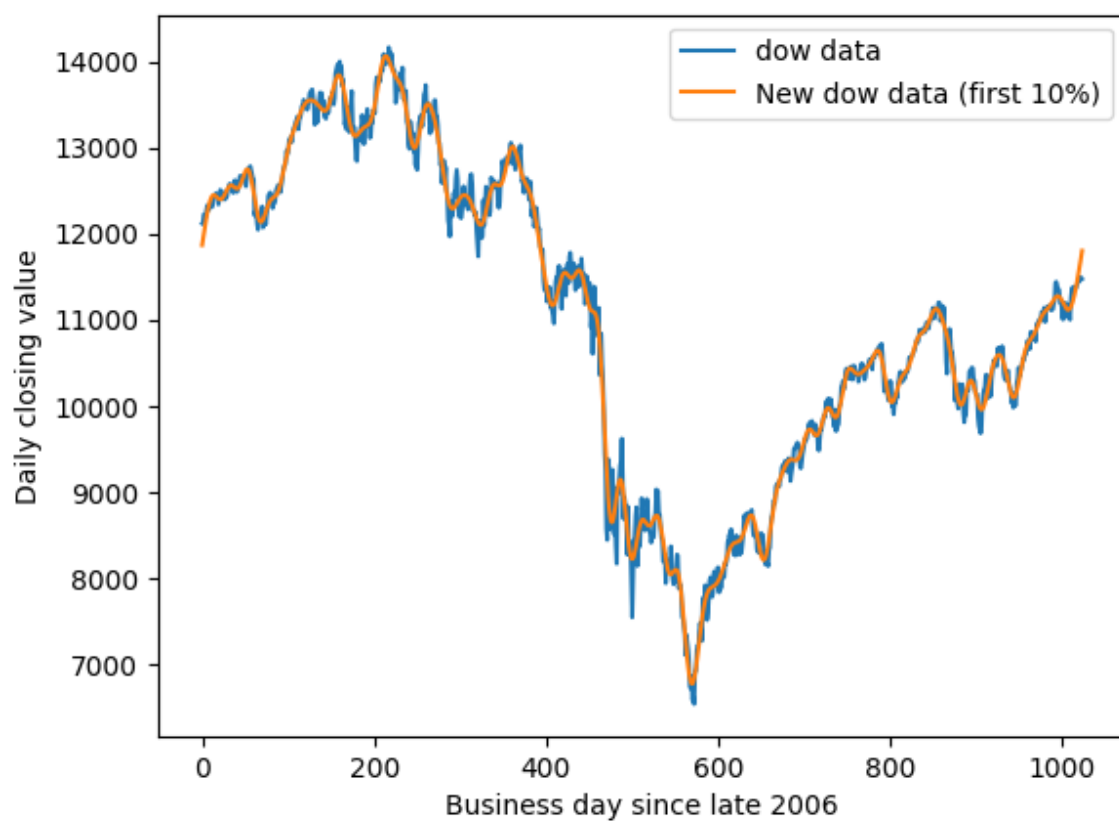


Figure 8: Plot the new dow data after keeping only the first 10% of the Fourier coefficients.

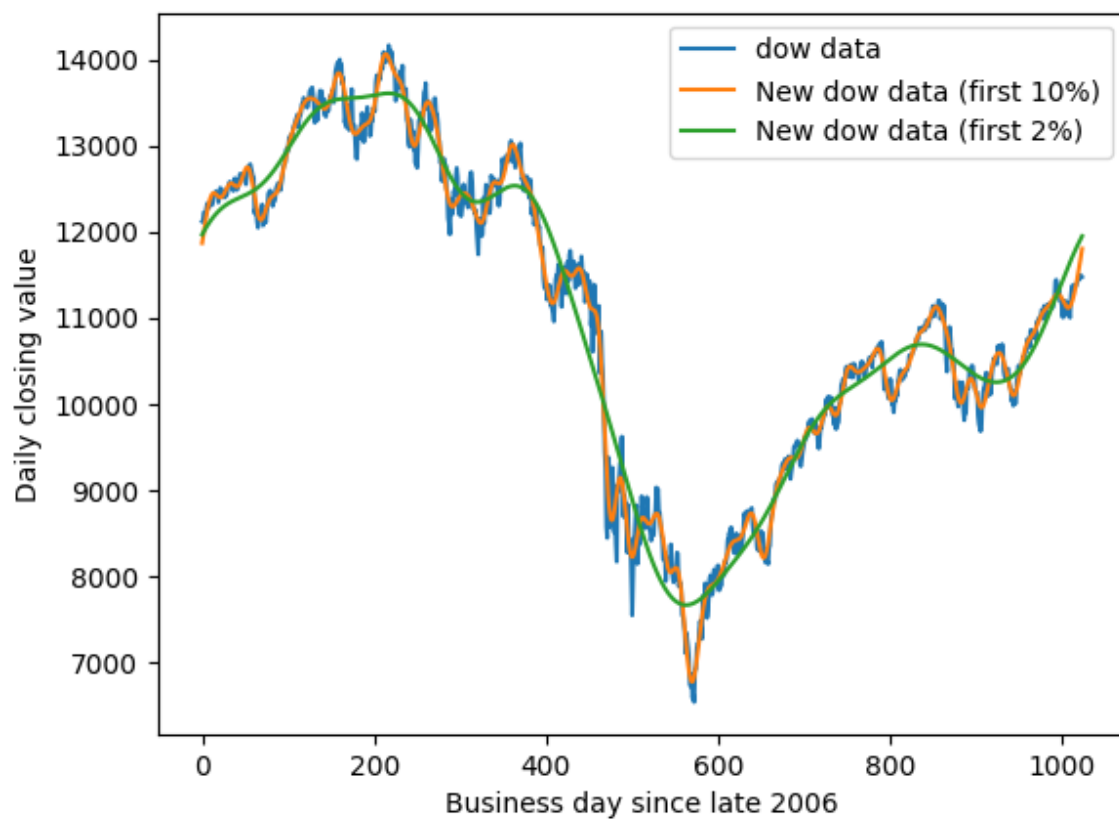


Figure 9: Plot the new dow data after keeping only the first 2% of the Fourier coefficients.