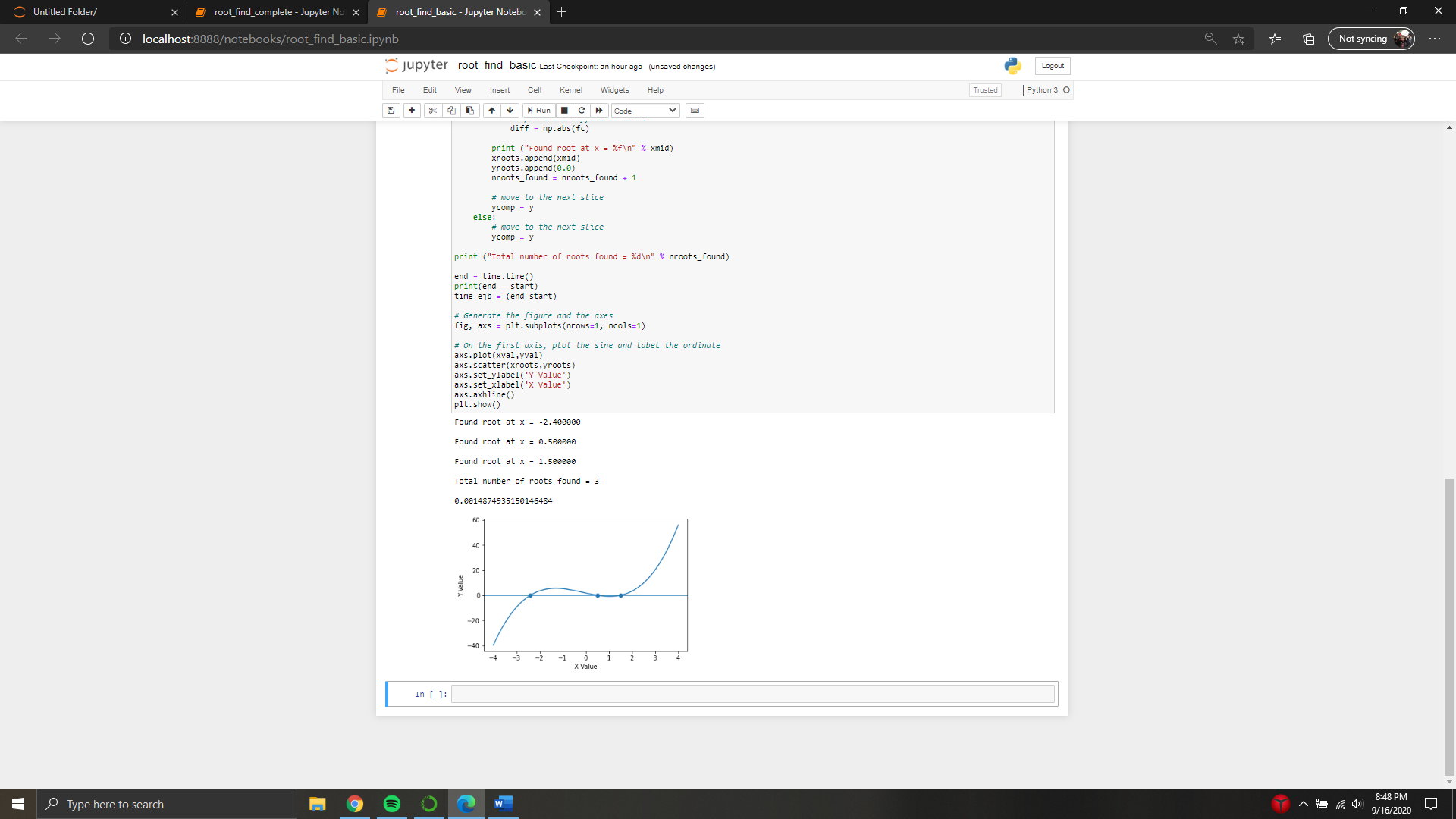
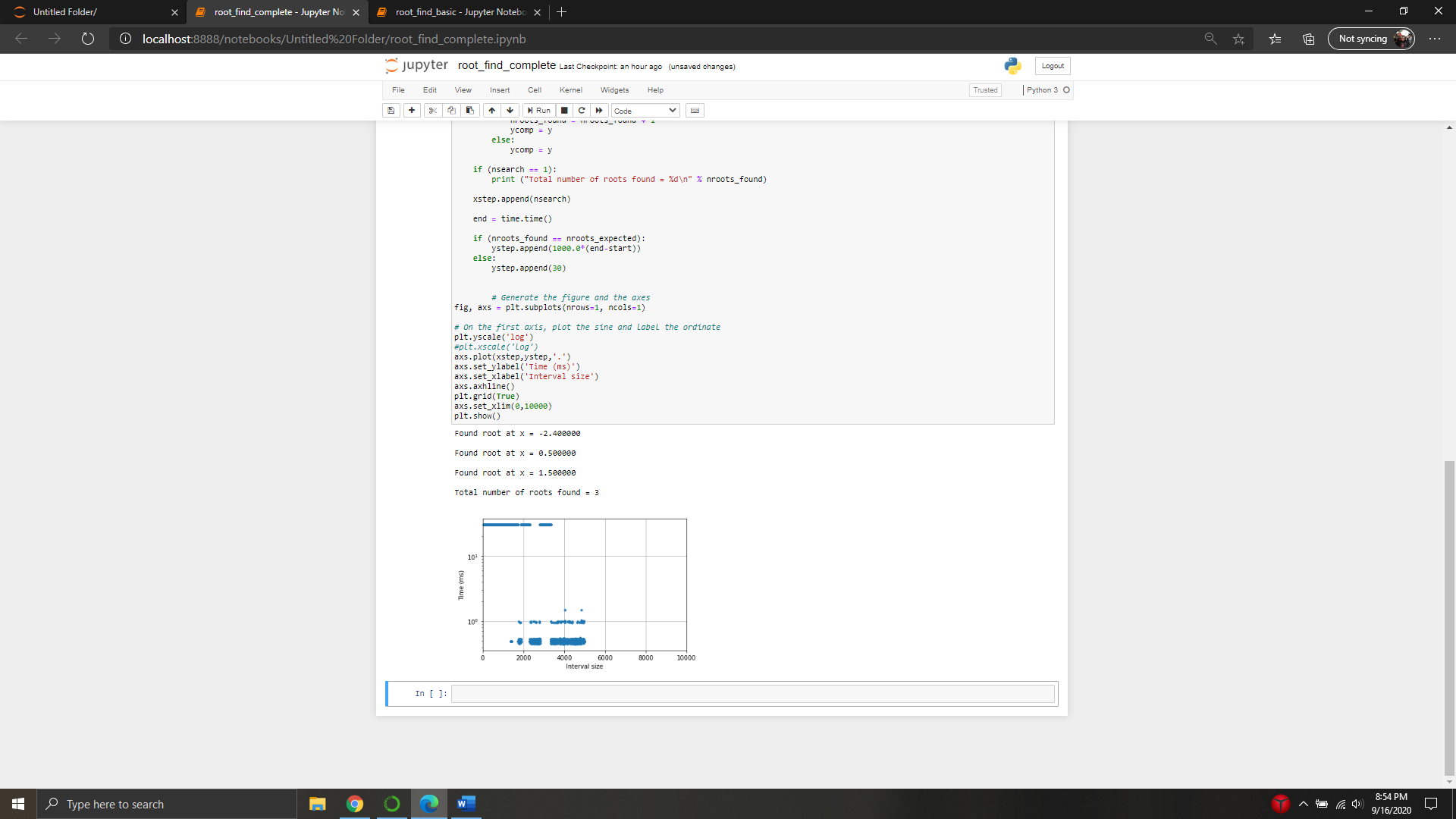
## Question 1 - Root finding using bisection method

a) Calculate the roots of this polynomial using the bisection algorithm. You can start with the file called root\_find\_basic.ipynb and then modify as appropriate. The program should fill an array of 10,000 data points between x=-4.0 and x=4.0. Then, the program should do a sparse search of this array, i.e. once every n\_search elements, to look for sign changes, and subsequently using the bisection algorithm to find the root. The program should also keep track of the time taken to find the roots.



b) Create a plot of the time taken to find the roots vs. the sparsification size (n\_search), for n\_search values between 1 and 10,000. You can start with root\_find\_complete.ipynb and then modify it.

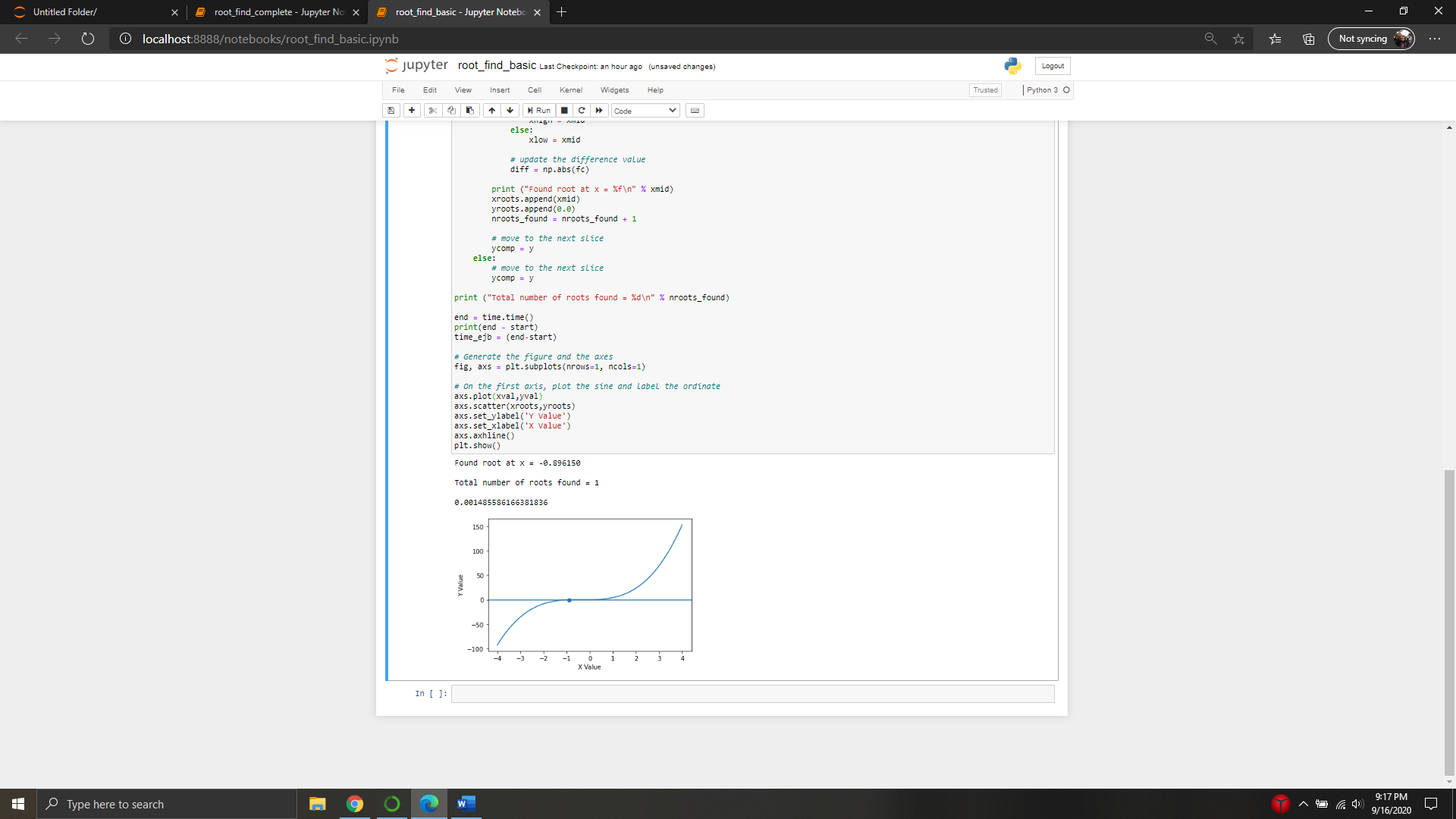


c) Deduce the source of the various patterns that you observe in part b). Hint: You might consider adjusting the number of data points in the original array, as well as the coefficients of the polynomial, slightly.

The various patterns, or outliers, seen in the above graph are based on the roots. When there is a single root you only see one constant line, that requires more time.

d) Modify the code created in part a) to allow for the possibility to find the roots of other polynomials or functions.

Outputs for the function



## Question 2 - Fitting data

1. Go to the following site and retrieve data on the density of air as a function of altitude:

Done and placed into airdensity.txt

1. Modify the Jupyter Notebook that we went through in class for Linear Regression to fit the density data as a function of altitude using a function of the form
2. Determine the parameters A, B, and C, as well as the uncertainties on these parameters.
3. Plot the fit, including 95% confidence interval error bands, along with the data. What is your evaluation of the quality of the fit? Can you think of a different approach that might yield a better fit?