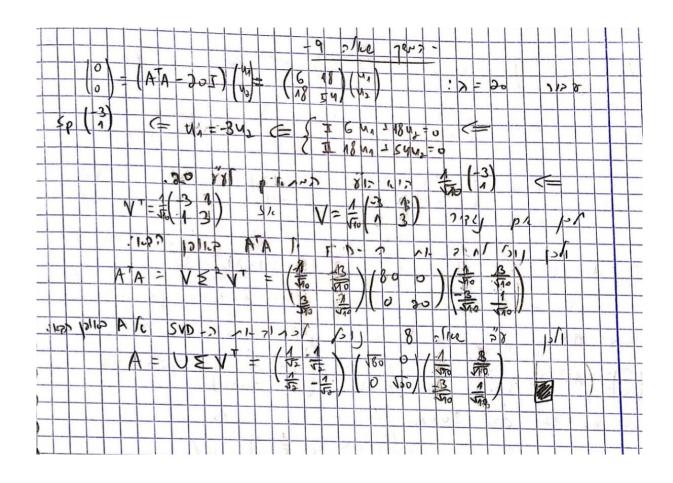
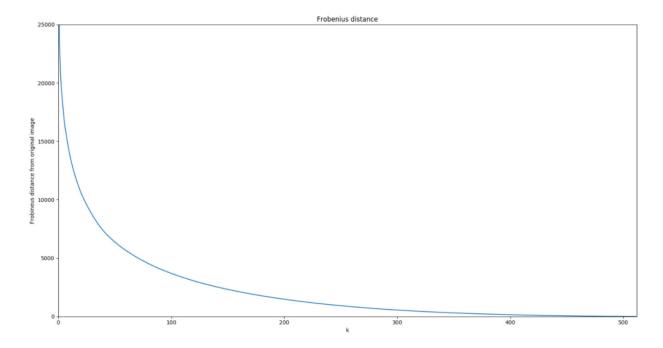


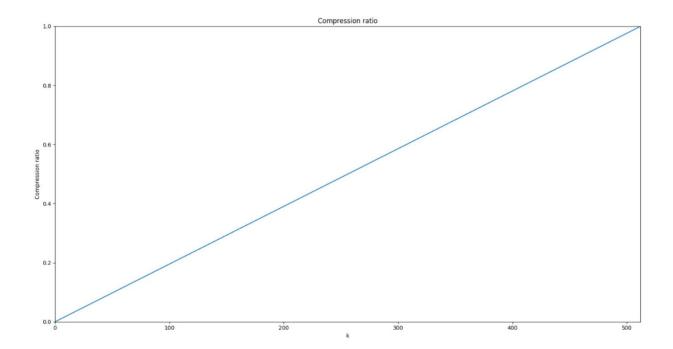
202 e M2×1 (9 10) - (50 30) (26 18) AAT /c xx 1-(50-) 30 - (50-) - 30° 5 = 2500-200x - x2 - 900 = x2 - 100 x = 1600 = (x-80)(x-20) -1 68 1.00. (48 1 - 1 EN CONIN & 1/2 AAT /c sis 3/1/3 50 (1) 30 X NO= 30X2 X = X2 AAT ( ) TE ( ) (30 30) (91) : >= 20 = (30×1-30×2) 1127 11/15 131621 AAT 10 END ATA 11/26/6 6. 14 8/11 76,000 C. 7/128 700 pels ATA Te is so pos ATA les AAT 1c NL 1.3NJ. 80 1 20 817 : X = 80 2124 14/15 0 = 18 V1 - 6 V2 0 = 54 V1 - 18V2 54V = 18V3 THE 80 XXI 61-10 XI 110

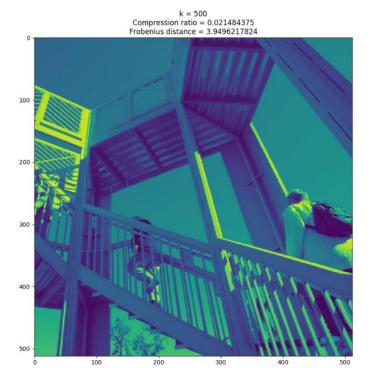


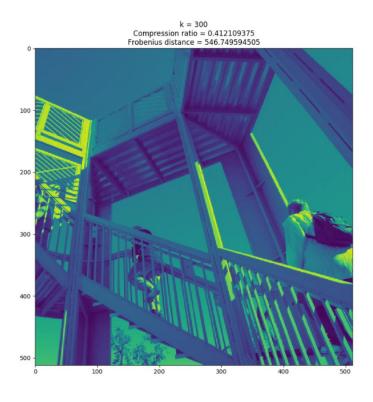
## 10. Frobenius distance as a function of k, for 512>=k>=0:

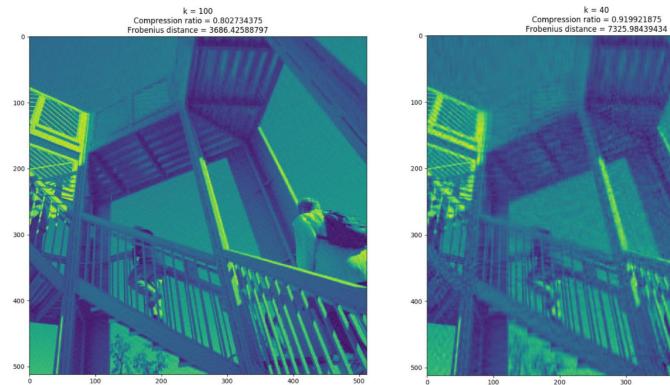


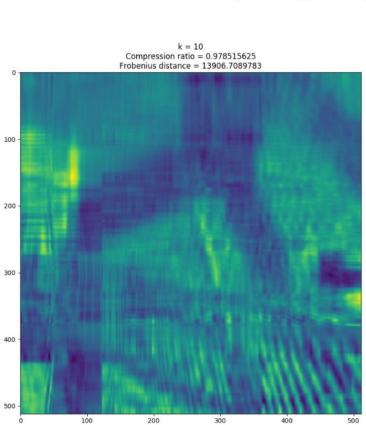
## Compression ratio as a function of k, for 512>=k>=0:



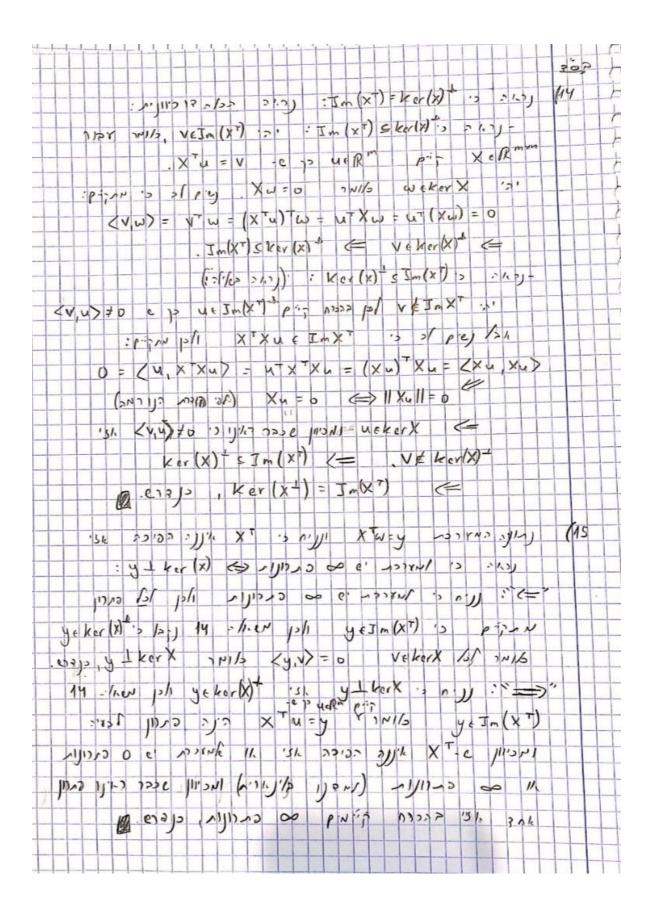


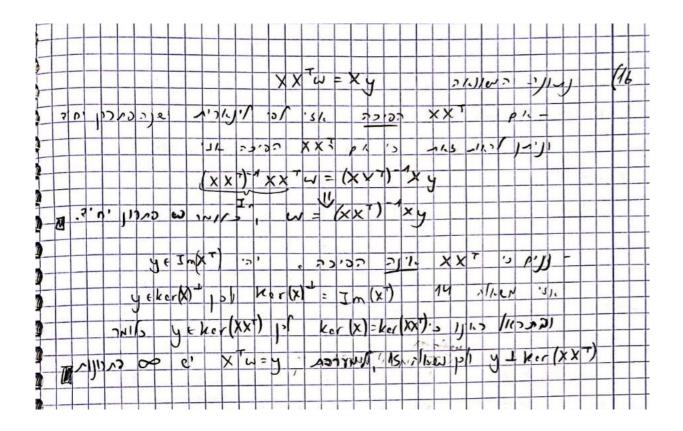






12. First let's notice that 'k' stands for the amount of singular values left, as such we see the the compression ratio is linear to the singular values, as expected. Interesting enough we see in the forbenius distance graph that there is an exponential drop within 100>=k>=0 due to this, we can see in the images that I chose that while k>=100 all though there are very big drops of data we see that the difference in sharpness between k=500 and k=300 is almost non existnat and when k=100 we can tell the difference but we see it is small and the picture is still clear. But for k<100, we see that the loss of sharpness is much more obvious all though the 'leaps' between k of 500,300,100 are bigger than the leaps between 100,40,10 yet the forbenius distance is MUCH bigger, and the sharpness much duller.

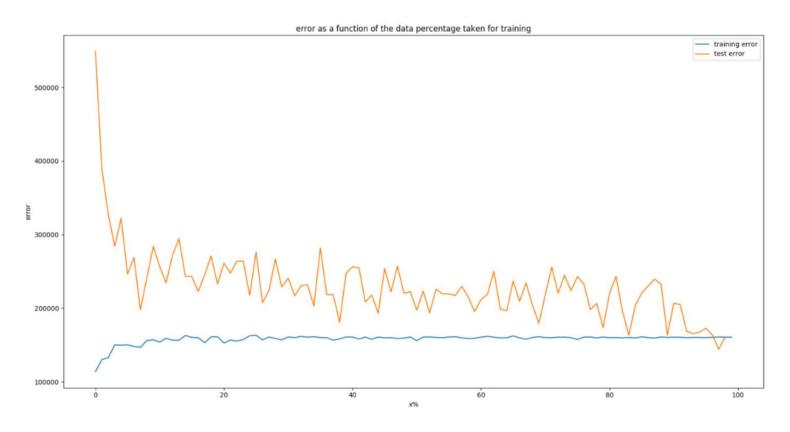


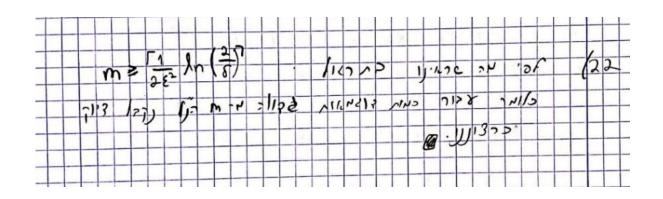


- 17. After looking in depth at the data I realized that there are irrelevant negative prices and 'sqft lot 15' and also houses with MUCH more bedrooms than the majority so I dropped all those. Next I dropped any house which had any NAN in one of its parameters.
- 18. I decided that ID and Date were irrelevant for the learning process so I dropped those entire columns. I also realized that the 'sqft living' colomn is linearly dependent with 'sqft above' and 'sqft basement' so I dropped 'sqft living'. Finally, I realized that there is a limited amount of zipcodes and that it is hard to understand any numeric value from a zipcode so I used the 'get\_dummies' function on it and changed it into a binary option of the different zipcode possibilities.

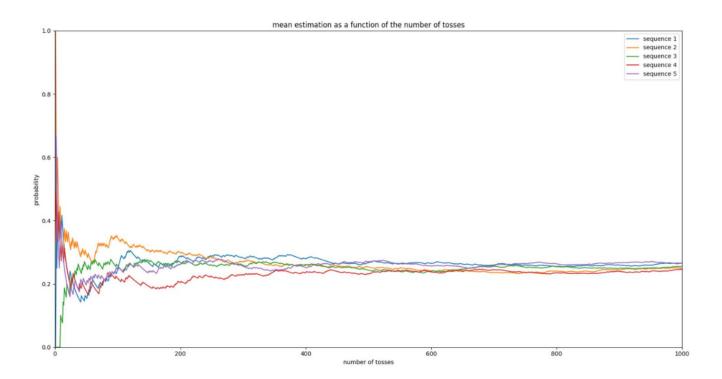
19+20: see attached code files.

## 21. Train and test errors as functions of x (data percentage):

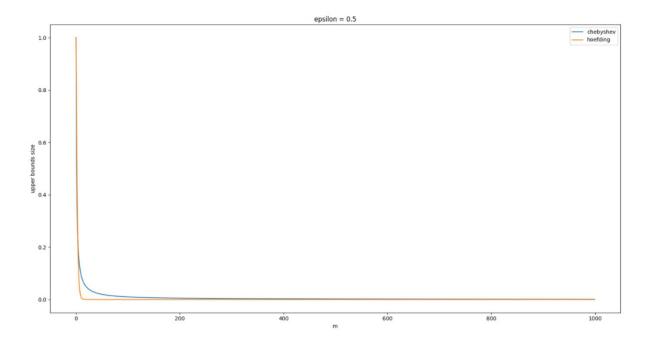


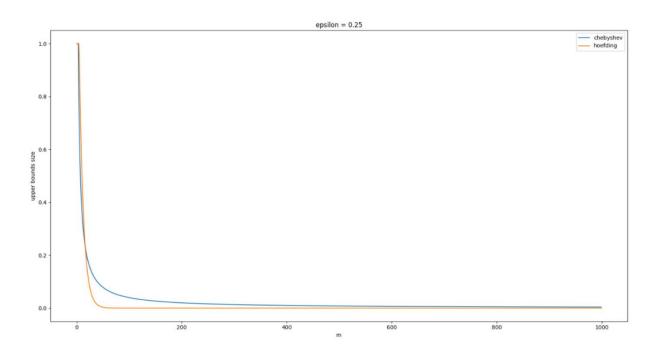


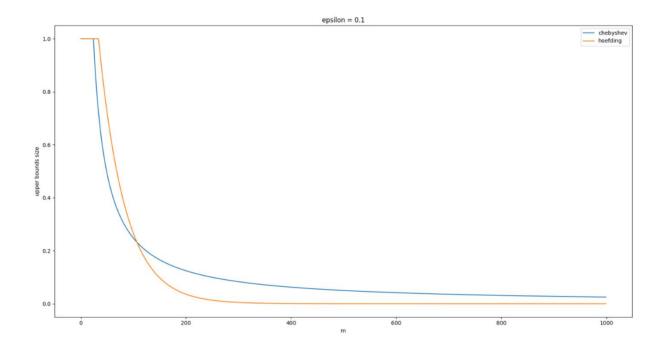
23. a) As m grows, I expect to see that the graph of each sequence should converge to the given statistical bias of the coins of p=0.25 (and this does happen as seen in the following graph):

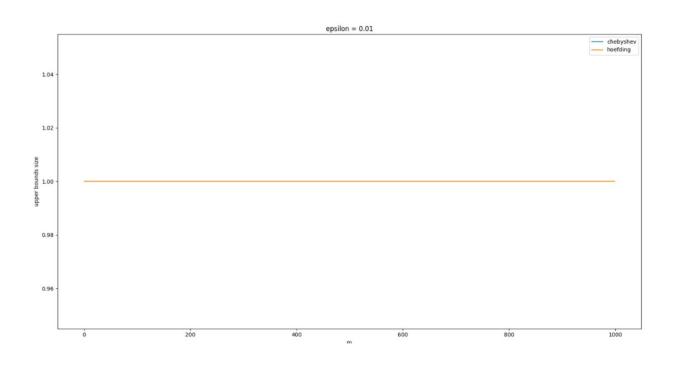


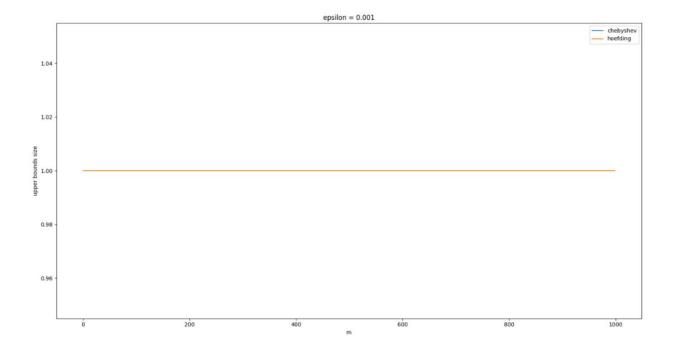
23 b) The next 5 figures show the upper bounds of the Hoeffding and Chebyshev innequlities, as tought in class, per epsilon (while epsilon varies):











23. c) The next 5 figures show the percentage of sequences that satisfy per epsilon (varies), beside the previous upper bound graphs, and while we now know that  $\mathbb{E}\left[X\right] = p = 0.25$ 

$$\left|\overline{X}_m - \mathbb{E}\left[X\right]\right| \geqslant \epsilon$$

As such and based on section (a) I would expect to see that as m (the number of tosses) grows then the  $|\overline{X}_m - \mathbb{E}[X]| \geqslant \epsilon$  difference grows smaller (because of the converging as seen in section a) and so the percent of sequences satisfying such should grow smaller. But as epsilon grows smaller, the tolerance is more strict and as such the percent of sequences should grow larger per m (in comparison to the same spot of m when calculated with larger epsilons):

