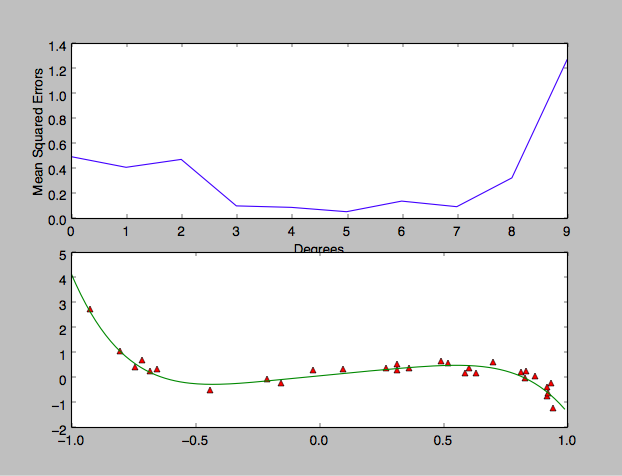
1.

A.)



I used 3 as the number of folds because the total size of the data was 30 – which mean each chunk is 10. This is enough data to test with and not be affected by noisy data.

B.) Best k value was 5 – this works because of the bias/variance relationship. Error associated with bias is high on lower orders since it’s value is determined by how well the regression fits the data and variance is high on higher order regressions because it increases faster as you try to extrapolate the data with higher order regressions. So it makes sense that 5 is the best order because it represents the minimum in the mean squared error plot (the minimum of errors associated with bias and variance)

C.) The value of X=3, plugged into the polynomial is -803.6.

This is probably accurate because cross-validation has told us that a 5th-order polynomial would have the lowest MSE. This model was trained very specifically for data points with x-values from -1 to 1, and didn’t expect a value outside of that range. Extrapolating data values far from this range, definitely does not have good results!

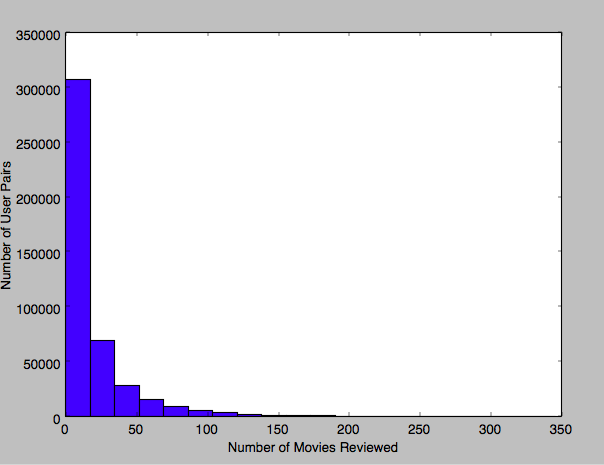
2.

1. See code. X1: W = [-11.0, 4.0586599999999997] iterations =4
2. Perceptron never converges because X2 is not linearly separable. There are 3 clusters of data, which means that there can’t be a single line that separates the data into 2 classifications (-1 and 1).
3. After plotting the values in X2 and Y2, one can see that X2 values near 2 and 0 correspond to a Y value of -1 and X2 values near 1 correspond to a Y value of 1. By transforming the X2 dataset by taking the floor of each value and replacing it with a -1, if it equals 0 or 2, and replacing it with a 1 if the floor is 1, you can make the data linearly separable. This transformation doesn’t change the number of Y classifications so it’s a valid transformation.

X2: W = [0.0, 2.0] iterations = 1

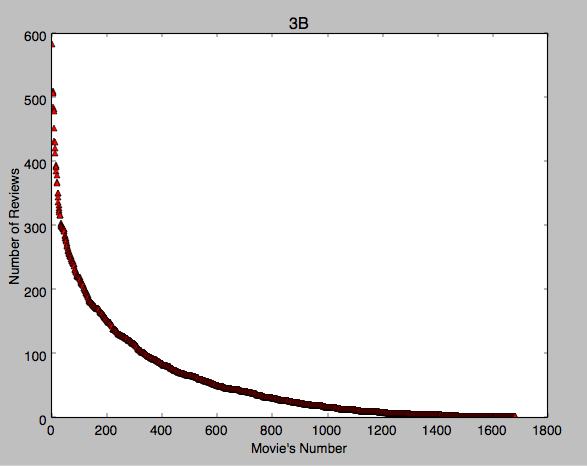
3.

A.) Mean is 18.78, Median is 10.



Choices made: The number of bins was set to 20 and unrated movies were set as zero and not counted.

B.)



Most: Movie #50 had 583 reviews

Least: Movie #1682 had 1 review

Yes, after ranking the movies based on reviews the data follows Zipf’s law – there’s an exponential decay in the number of ratings as movie rank decreases. The law explains that the volume or size of entities is inversely proportional to a power s (s > 0) of their ranking.

4.

A.)

Toy Examples

Zero Fill:

manhattan([1, 4, 0, 2, 1] ,[2, 0, 0, 1, 5]) = 10

manhattan([2, 0, 0, 1, 5],[1, 2, 3, 4, 2]) = 12

manhattan([1, 4, 0, 2, 1],[1, 2, 3, 4, 2]) = 8

Average fill:

manhattan([1, 4, 1.6, 2, 1],[2, 1, 1, 1, 5]) = 9.6

manhattan([2, 1, 1, 1, 5],[1, 2, 3, 4, 2]) = 10

manhattan([1, 4, 1.6, 2, 1],[1, 2, 3, 4, 2]) = 6.4

Filling in the default value based on the average value of the user leads to a more accurate distance measurement. Even though the range of ratings is between 1 and 5 users inherently have their own rating system within that range that could be smaller. By setting the default

B.)

Toy Examples:

pearsons([1, 4, 3, 2, 1],[2, 3, 3, 3, 5]) = 1.2100420126

Euclidean([1, 4, 3, 2, 1],[2, 3, 3, 3, 5]) = 4.35889894354

pearsons ([1, 4, 3, 2, 1],[2, 0, 0, 0, 5]) = 1.73514704411

Euclidean([1, 4, 3, 2, 1],[2, 0, 0, 0, 5]) = 6.78232998313

Pearson’s would be better because it is less affected by the 3 value default compared when compard with Euclidean. This is key because you don’t want to have unrated movies by users affecting the rating of the movie, which makes item based collaborative filtering less accurate. The fact that a user doesn’t rate a movie should not affect the overall rating of the movie and by putting 3 as a default value, which is the median of the ratings range, the rating of a given movie is skewed.

5. See code.

6.

A.)

The error measure I used was a simple right/wrong (Boolean) counter. Every time the algorithm incorrectly predicted the user’s movie rating, I incremented the error counter and summed this value across all folds (50 folds, 100 errors max) per experiment/variant tested.

B.)

For the statistical test, I used a paired independent t-test (scipy.stats.ttest\_ind). This statistical test takes in the errors counted for each fold for each variant and the test measures whether the average (expected) value differs significantly across samples. If we observe a large p-value, for example larger than 0.05 or 0.1, then we cannot reject the null hypothesis of identical average scores. If the p-value is smaller than the threshold, e.g. 1%, 5% or 10%, then we reject the null hypothesis of equal averages.

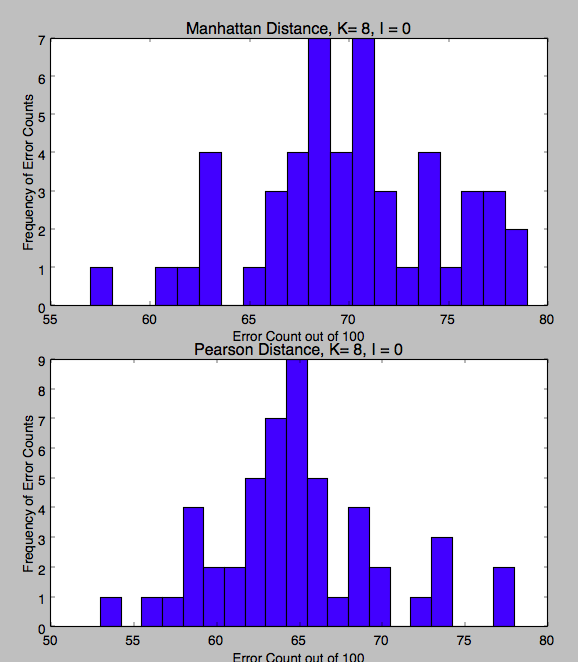
C.)

T-Test: Manhattan, Pearson (array(5.041411909612306), 2.1176631802699955e-06) –

According to the t-test, we reject the null hypothesis in favor of the alternative. The t-statistic is positive meaning that the mean of the first (manhattan) is larger which means that it is less accurate than the second data set (pearson).

Our assumptions for this are that K is set at 8 and that we would ignore zero values. This lends to a more accurate result across the board and properly tests distance measures with real data and a sizable set of nearest neighbors where the mode is somewhat significant.

This does match the intuition we developed in Problem 4,that Pearson is more robust.

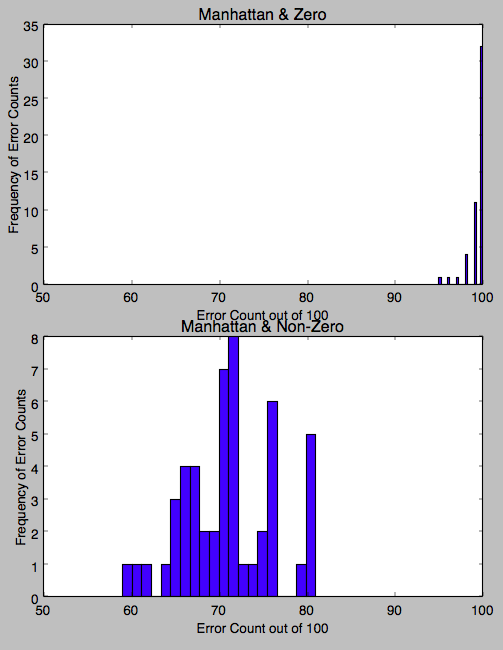


6D.

**Manhattan With Zero & Non-Zero**

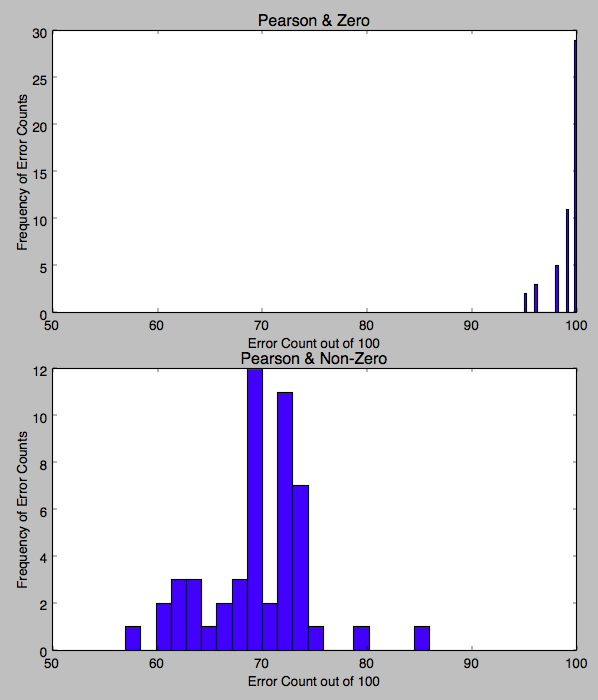
T-Test Result: (array(36.9566181030425), 2.4074074103490135e-59)

Histogram on next page.



**Pearson With Zero & Non-Zero**

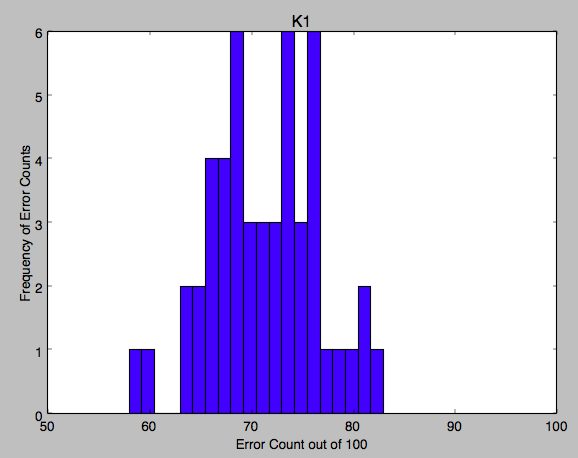
T-Test result: (array(39.647963570392726), 3.7631236797669542e-62)

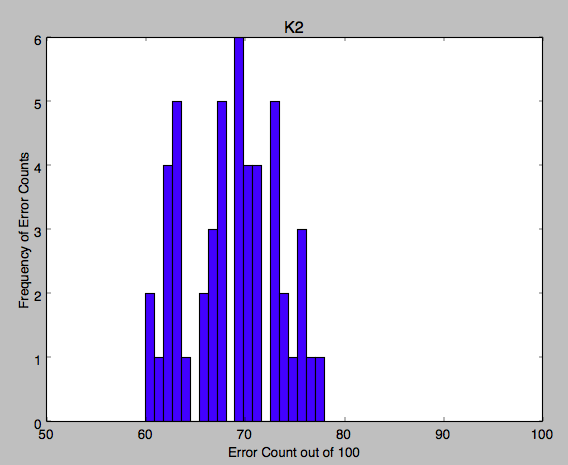


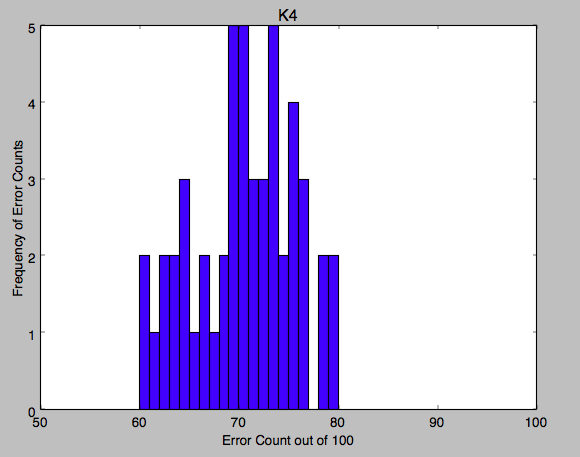
Here we assume a K value of 8 for the same reasons outlined in 6C. Ignoring zeros greatly improves the accuracy of the collaborative filter as evidenced by the statistical t-test and the histograms. The intuition we gained in problem 4 is the same here because zeros greatly reduce the accuracy of the collaborative filter.

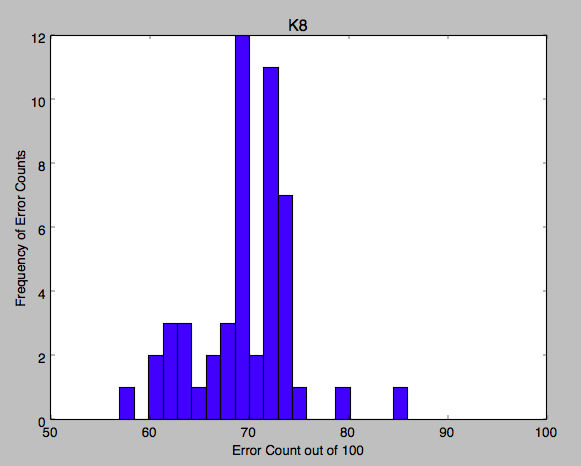
6E

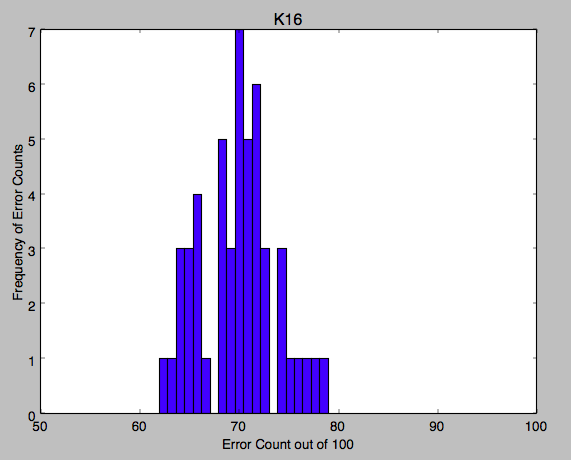
Error Histograms for each K value:

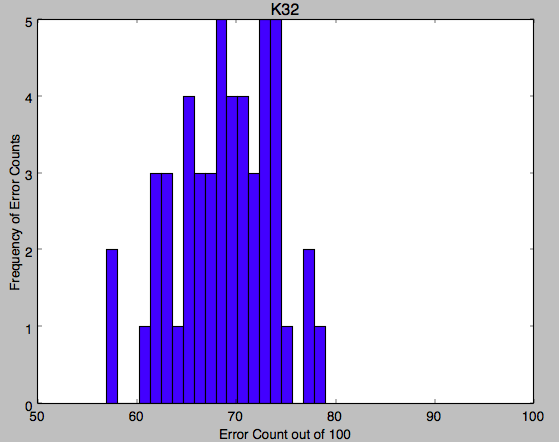










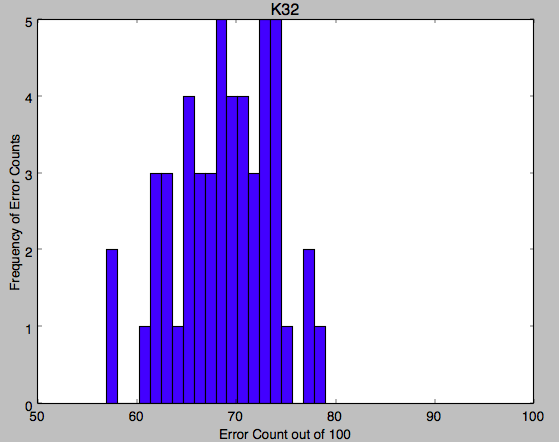


The best settings used here for user-based collaborative filtering are Pearson distance and ignoring zeros.

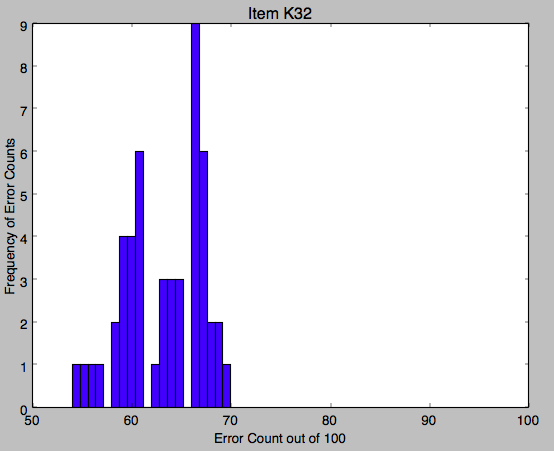
After running the t-tests on all combinations of K values, I found that the null hypothesis was favored among values K=8,16,32. Among those K values, after examining the histograms for these three K values, it looks like 16 and 32 are more accurate with 32 having a lower error average. Therefore, selecting 32 as the optimal K value is justified.

F.)

This is the user-based histogram with a K value of 32, Pearson distance and ignoring zeros:



This is the item-based histogram with a K value of 32, Pearson distance and ignoring zeros.



T-Test result: Item, User (array(-6.250592335771605), 1.0650962712770216e-08)

The item based collaborative filter is better according to the histogram and t-test. The negative t-statistic value means that Item has a lower mean compared to User, which means it is a more accurate collaborative filter. I feel fairly confident in these optimal settings, however, the accuracy of collaborative filters is still weak with an average error rate of over 60%.