1) I experimented with different values of k as demonstrated below. The best validation accuracy for “agaricus-lepiota” dataset was 100 percent when k=1. The best validation accuracy for “primary-tumor” dataset was 47.06 percent when k=10.

The changes of validation accuracy here has similar trend compared to the decision tree algorithm. For “agaricus-lepiota” dataset, the validation accuracy increases as the model complexity increases (K gets smaller). This is similar when the complexity of decision tree model increases (higher depth values). Furthermore, in “primary-tumor” dataset, initially the validation accuracy is lower. As the value of k increases, it reaches its maximum around k=10 and then it starts to decrease. Therefore, there is a point (k=10) with maximum validation accuracy which is located between the highest complexity which overfits and the lowest complexity which underfits. There is a similar trend in the decision tree algorithm for “primary-tumor” dataset.

agaricus-lepiota Dataset:

K	Train Accuracy	Validation Accuracy	Test Accuracy
1	100	100	100
3	100	100	100
5	99.95	99.90	99.90
10	99.95	99.85	99.85
20	99.93	99.80	99.85
50	99.78	99.75	99.61
100	98.50	97.83	98.47
1000	88.75	89.02	88.58

primary-tumor Dataset:

K	Train Accuracy	Validation Accuracy	Test Accuracy
1	94.67	34.12	27.06
3	56.21	34.12	38.82
5	47.93	43.53	43.53
10	48.52	47.06	42.35
20	44.97	44.71	40.00
50	43.79	43.53	36.47
100	29.59	23.53	37.65

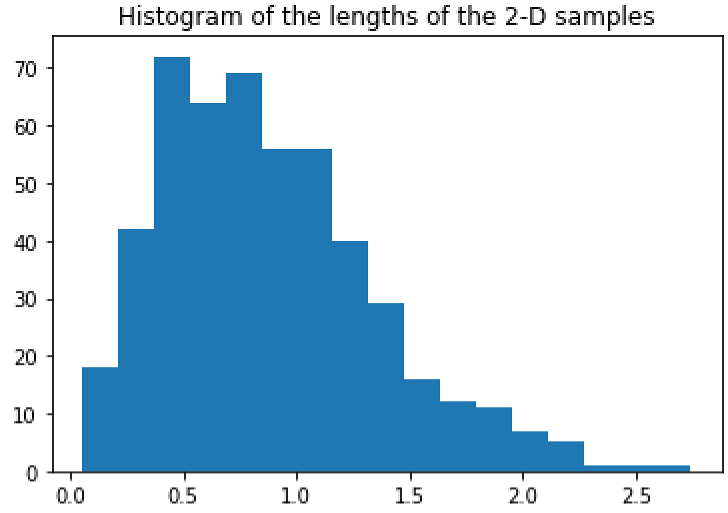
2) As K increases, the complexity of the model decreases. At K=1, the model has the highest complexity and therefore it is more prone to overfitting. However, when K is equal to the number of samples in the training set, the algorithm just picks the majority over the whole set, so it clearly underfits.

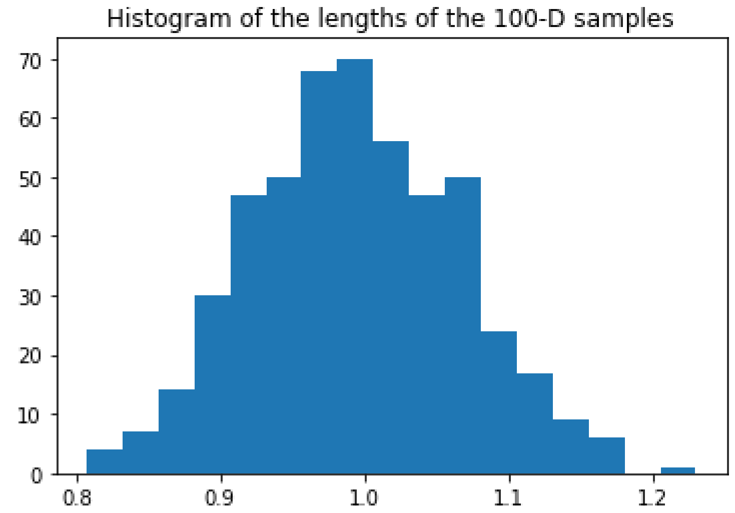
In “agaricus-lepiota” dataset, as K increases, both of the training and validation accuracies slowly start to decrease. There is no sign of overfitting in this dataset because the highest complexity is giving the highest validation accuracy.

In “primary-tumor” dataset, however, the highest complexity (k=1) gives the highest training accuracy, but the validation accuracy is not optimal. The model is apparently overfitting. As we lower the complexity a little bit, the model gives the best validation accuracy around k=10. After that, as we continue to lower the model complexity, the training and validation accuracies decrease as well.

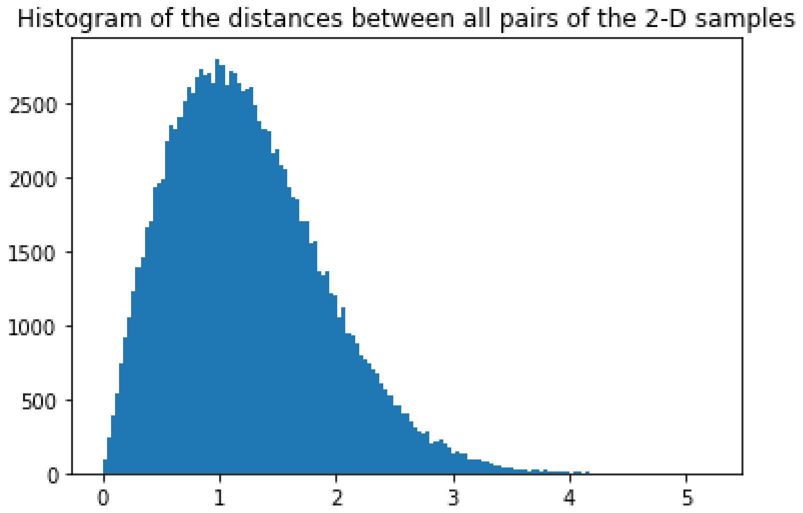
3) There is no training time for the KNN classifier because the actual classification happens at test time. When the number of samples is high and each sample has a high-dimensional feature vector, the classification time is considerably high. This happened for “agaricus-lepiota” dataset. It has 4062 training data points and 2031 validation and test data points. Also, each data point has 117 features. On a computer system with Core i5-7200U CPU at 2.5GHz, it took several minutes to classify the training and test sets. However, classification in “primary-tumor” dataset took less than 5 seconds. It can be due to the fact that it has only 169 training data points and 85 validation and test data points. Each data point has 42 features.

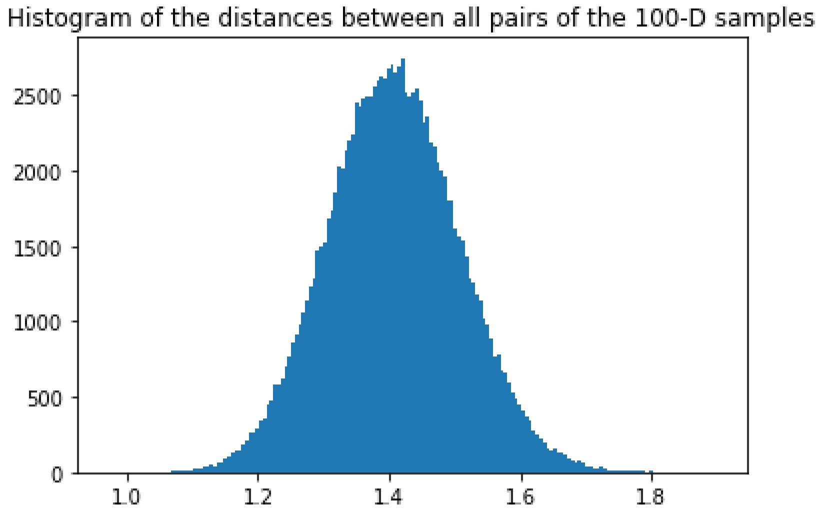
4) Here are the histograms of the lengths of the 2-D and 100-D generated samples:



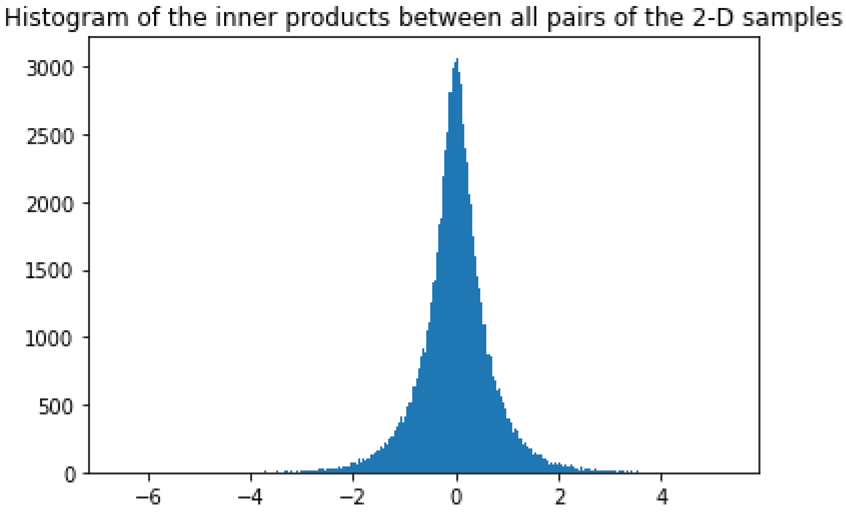


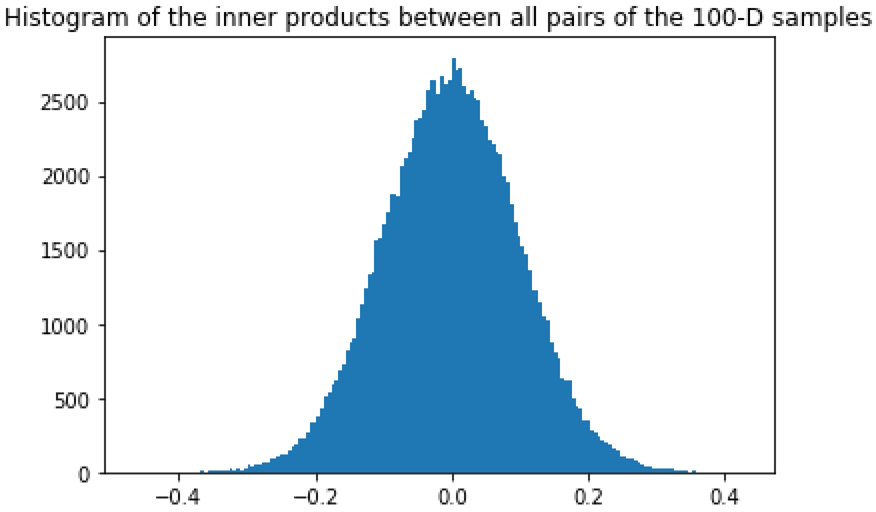
Here are the histograms of the distances between all pairs of the 2-D and 100-D generated samples:





Here are the histograms of the inner products between all pairs of the 2-D and 100-D generated samples:


Histogram of the inner products between all pairs of the 2-D samples


Histogram of the inner products between all pairs of the 100-D samples

* As it is observable in the 100-D samples, multiple unit-variance Gaussians in large dimensions has the following characteristics:

- The average length of the samples are 1 and the variation around the average is small (compare to low-dimensional samples).
- The distance between any pair in the high-dimensional data converges to a fixed value as the dimension becomes larger and larger. In other words, the distance between most of the pairs in a high-dimensional data is the same. For this example, the expected value for the distance between any two samples is √2:
  || v1 - v2 || = √ (||v1|| + ||v2|| - 2<v1, v2>) = √ (1+1-0) = √2

- The value of the inner product of any two samples in high-dimensional data converges to zero as the dimension becomes larger and larger. In other words, most of the samples in high-dimensional data are orthogonal to each other.

***** Because the distance between most of the pairs in a high-dimensional data is the same, the KNN classifier may not have a good performance in such cases.

5) I experimented with different K and d values as demonstrated in the tables in the next page. For “agaricus-lepiota” dataset, K=1 still gives the best performance and d=10 seems a good candidate. The difference between training and validation accuracy for d=10 and for the original one is less than 2 percent and it still effectively reduces the dimension space from 117 to 10. Therefore, it can be a good candidate. For “primary-tumor” dataset, K=10 still gives the best performance and d=20 seems a good option. It reduces the dimensional space from 42 to 20 and also has training and validation accuracy with less than 10 percent difference compared to the original.

Here, we can observe that the dimensionality reduction does not hurt the training and test accuracies considerably if it is done properly. Furthermore, with dimensionality reduction we can considerably improve the run-time of the KNN algorithm for “agaricus-lepiota” dataset. The trade-off is that the training and validation accuracies are slightly lower.

agaricus-lepiota Dataset:

K	d	Train Accuracy	Validation Accuracy	Test Accuracy
1	1	100.00	55.88	55.05
1	5	100.00	83.60	85.28
1	10	100.00	97.14	97.83
1	20	100.00	99.95	100.00
1	50	100.00	100.00	100.00
5	1	73.66	59.23	59.38
5	5	95.62	93.55	93.55
5	10	99.73	99.21	99.46
5	20	99.98	99.85	99.85
5	50	99.98	99.85	99.85
10	1	65.02	54.95	53.13
10	5	94.19	91.58	91.29
10	10	99.56	99.21	98.97
10	50	99.95	99.85	99.85
50	1	62.28	58.94	58.44
50	5	87.49	87.64	87.30
50	10	97.86	97.19	97.74
50	50	99.66	99.70	99.46
100	1	64.57	65.53	64.25
100	5	92.91	91.78	91.29
100	10	96.53	96.21	96.60
100	50	98.57	97.88	98.62

primary-tumor Dataset:

K	d	Train Accuracy	Validation Accuracy	Test Accuracy
1	1	94.67	16.47	16.47
1	5	94.67	23.53	20.00
1	10	94.67	31.76	27.06
1	20	94.67	32.94	24.71
1	50	94.67	40.00	34.12
5	1	39.05	21.18	12.94
5	5	42.60	30.59	28.24
5	10	49.70	30.59	36.47
5	20	50.89	36.47	41.18
5	50	46.75	41.18	32.94
10	1	36.09	18.82	11.76
10	5	41.42	34.12	27.06
10	10	44.38	38.82	30.59
10	20	48.52	42.35	34.12
10	50	52.07	45.88	40.00
50	1	29.59	27.06	35.29
50	5	26.04	31.76	23.53
50	10	33.73	34.12	31.76
50	20	37.87	36.47	37.65
50	50	43.20	40.00	38.82
100	1	21.89	24.71	29.41
100	5	27.22	27.06	34.12
100	10	23.67	30.59	28.24
100	20	28.40	28.24	37.65
100	50	28.99	24.71	32.94

6) Dimensionality reduction can improve runtime of the algorithm. Moreover, it may also be possible to achieve a higher accuracy by doing a dimensionality reduction because it may reduce the noise. It may remove some dimensions that are too noisy or average the noise of several dimensions.