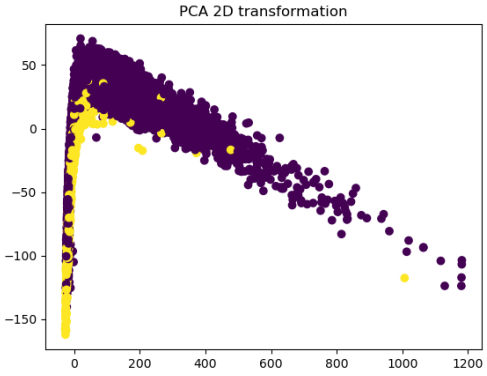
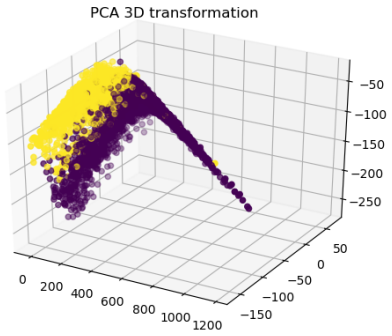
CS 4641 Machine Learning
Bojun Yang — Section B Homework 3 Writeup

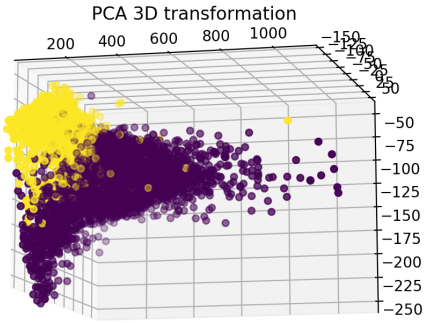
1. PCA Analysis

- **HTRU2**





It is not apparent that the transformed data is linearly separable from the 2d data of the PCA transformation. It appears that the data set is blended and but there is a separation between the two labels. Taking a look at the 3d data, we can tell that there may exist plane that separates the differently colored data points.



Looking at the 3d data from another angle, we can see another view that also seems to be separable with a plane.

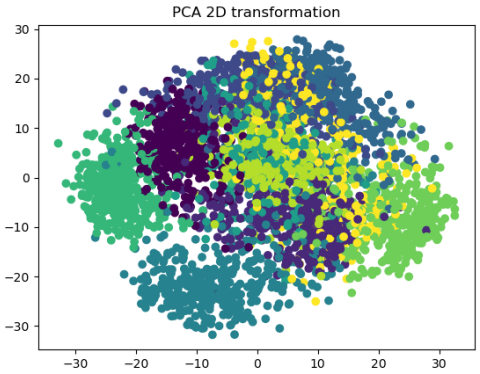
- original data **fit time**: 0.18409967422485352
- 2d transformed data **fit time**: 0.03162217140197754
- 3d transformed data **fit time**: 0.03455996513366699

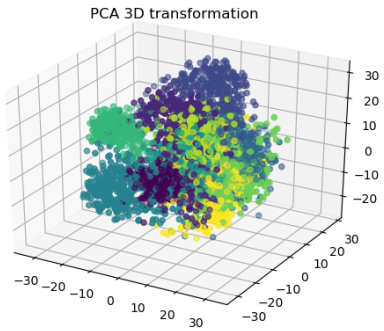
- 2d **train** score: 0.9779298784746473
- 2d transformed **train** score: 0.9193323089817014
- 3d transformed **train** score: 0.9709456628020673

- 2d **test** score: 0.9826815642458101
- 2d transformed **test** score: 0.9164804469273743
- 3d transformed **test** score: 0.9745810055865922

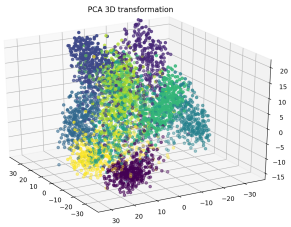
The results of the LogReg tests make sense based on our observations. The 3d transformed data performed better than the 2d transformed data. The performance on the 3d transformed data almost matches the performance on the original data, and a good performance at that. In addition to having similar performance, it is worth noting that the time to fit the data was significantly faster for the transformed data than the original data.

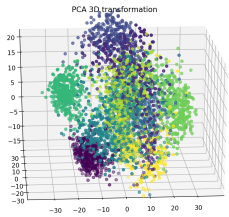
- **digits**

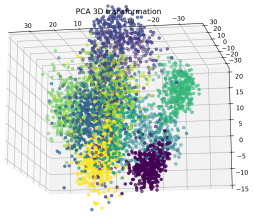


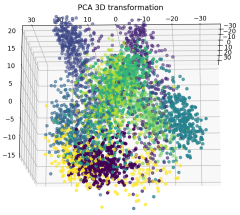


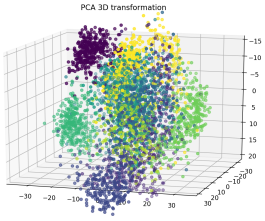
It isn't imediately apparent whether the PCA transformed dataset for digits is linearly separable from either the 2d or 3d data. However if we look at the transformed data at different angles in the 3d data as shown below, we can see some separation of the differently colored data points.











It looks like that all the data groups except for yellow look somewhat linearly separable. However, from the different views, you can tell that the yellow data points form a disk-shaped cloud volume in the middle of the other data points. However, it needs to be noted that each group of points seems overlap with another's groups. Running LogReg Tests on the data:

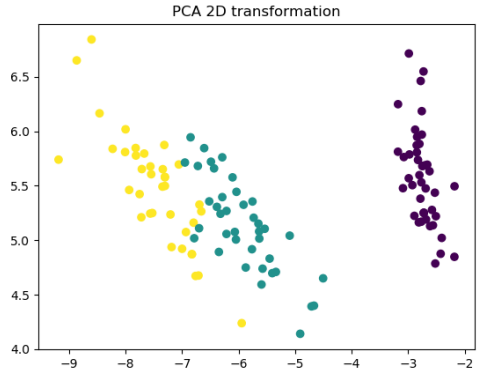
- original data **fit time**: 7.577993869781494
- 2d transformed data **fit time**: 0.7825779914855957
- 3d transformed data **fit time**: 0.6092972755432129

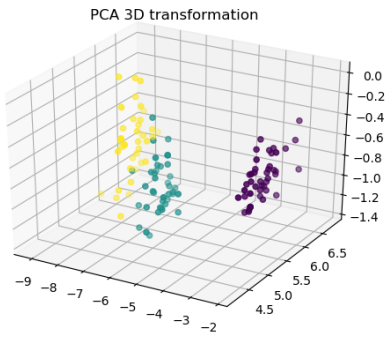
- 2d **train** score: 0.999738425320429
- 2d transformed **train** score: 0.612607899555323
- 3d transformed **train** score: 0.7501961810096782

- 2d **test** score: 0.9488035614913745
- 2d transformed **test** score: 0.5820812465219811
- 3d transformed **test** score: 0.7089593767390094

Based on our observations, these results make sense. LogReg did a lot better on the 3d transformed data than the 2d transformed data. However, none of the results from transformed data LogReg were as good as LogReg on the original dataset. These results confirm our suspicion that LogReg will not work well on the transformed data. However, it is worth noticing that the time to fit the transformed data was significantly lower than the time to fit the original data.

- **iris**





From the the 2d and 3d data of the PCA transformation, it is pretty clear that the transformed data is linearly separable. We can see 3 distinct and spearated groups of data points. Therefore, logistic regression will perform well to classify the transformed data.

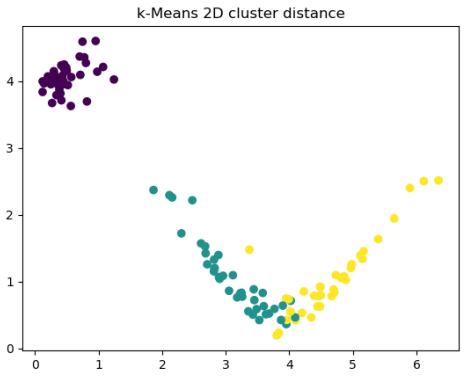
- original data **fit time**: 0.041589975357055664
- 2d transformed data **fit time**: 0.011603116989135742
- 3d transformed data **fit time**: 0.01744532585144043

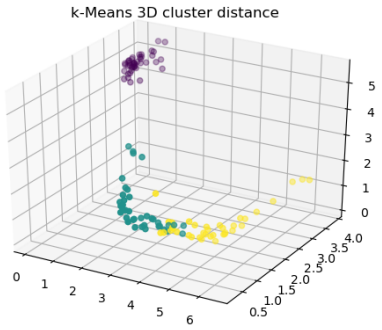
- 2d **train** score: 0.9666666666666667
- 2d transformed **train** score: 0.95
- 3d transformed **train** score: 0.9583333333333334

- 2d **test** score: 0.9666666666666667
- 2d transformed **test** score: 0.9666666666666667
- 3d transformed **test** score: 1.0

These results confirm our observation that the transformed data will perform well with LogReg. The performances are very similar and within 0.01 difference.

2. **K-Means Analysis**
   - **iris**





- original data **fit time**: 0.0294342041015625
- 2d transformed data **fit time**: 0.0076847076416015625
- 3d transformed data **fit time**: 0.014444112777709961

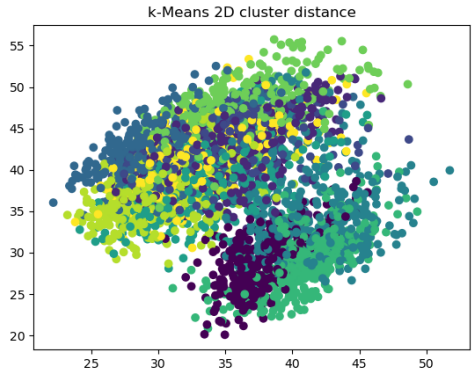
- 2d **train** score: 0.9666666666666667
- 2d transformed **train** score: 0.9416666666666667
- 3d transformed **train** score: 0.9416666666666667

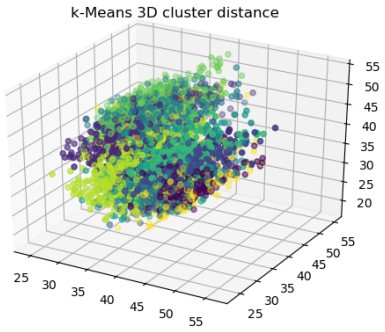
- 2d **test** score: 0.9666666666666667
- 2d transformed **test** score: 0.8333333333333334
- 3d transformed **test** score: 0.8666666666666667

Using k-Means as a feature transform compared to PCA did not perform as well as PCA. It would be better to use PCA for both 2d and 3d data sets as the performance of PCA transformed data is higher than that of k-Means. The graphs of k-Means on this dataset seems to produce data that is less linearly separable than that of PCA's.

The normalized mutual information was 0.59090106038302613 wtih 3 clusters. This makes sense because NMI is higher when points belonging to the same cluster have the same label. Looking at our graphs of the transformed data, we can imagine three centers where each center's cluster contains mostly purple, green, and yellow points, respectively.

- **digits**





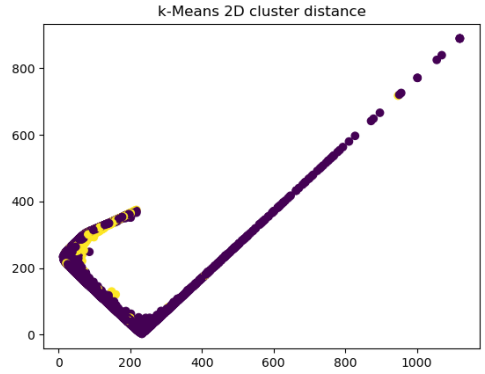
  - original data **fit time**: 6.619047164916992
  - 2d transformed data **fit time**: 1.1980550289154053
  - 2d transformed data **fit time**: 2.3269810676574707

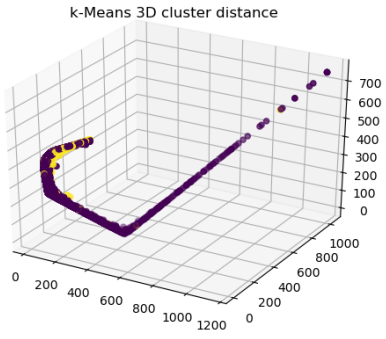
  - 2d **train** score: 0.999738425320429
  - 2d transformed **train** score: 0.46089458540413286
  - 3d transformed **train** score: 0.4969918911849333

  - 2d **test** score: 0.9488035614913745
  - 2d transformed **test** score: 0.4323873121869783
  - 3d transformed **test** score: 0.4802448525319978

Using k-Means as a feature transform is is not better than PCA. Comparing the LogReg scores with k-Means transformed data against the LogReg score for PCA transformed data, the scores for k-Means transformed is much lower. The graphs of the 2d and 3d data points also confirm this. The data points look to be more clustered together, making it harder for LogReg to find lines/planes of separation. The normalized mutual information was 0.7449158709032907 with 10 clusters. This makes sense because NMI is higher when points belonging to the same cluster have the same label. The graphs of the transformed data show points that can be sectioned of into 10 different volumes. It's easy to see that each cluster will correspond to one color of points.

- **HTRU2**





  - original data **fit time**: 0.1594712734222412
  - 2d transformed data **fit time**: 0.030047178268432617
  - 2d transformed data **fit time**: 0.04154205322265625

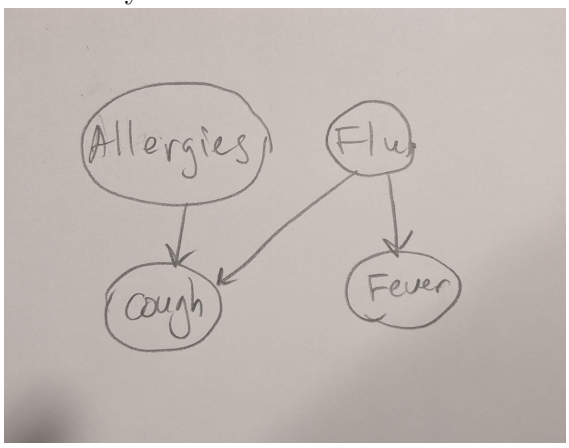
  - 2d **train** score: 0.9779298784746473
  - 2d transformed **train** score: 0.906201983517251
  - 3d transformed **train** score: 0.9088559854728314

  - 2d **test** score: 0.9826815642458101
  - 2d transformed **test** score: 0.9041899441340782
  - 3d transformed **test** score: 0.9044692737430168

Using k-Means as a feature transform is not better than PCA. The scores of LogReg on the transformed data using k-Means is close, but lower than the score of using LogReg on transformed data using PCA. The graphs become more complex and less so linearly separable. The transformed data using k-Means seem to be pushed closer together to form a line of some sorts. This brings the differently labeled data points closer together, increasing the chances of two differently labeled data being in the same cluster.
The normalized mutual information was 0.024972522126462778 with 2 clusters. This makes sense because NMI is higher when points belonging to the same cluster have the same label. For HTRU2, there are over 17,000 points and only 2 labels. This makes it very hard to get two clusters that encompass only one label, which is why the NMI is so low.

3. **Bayes Nets**

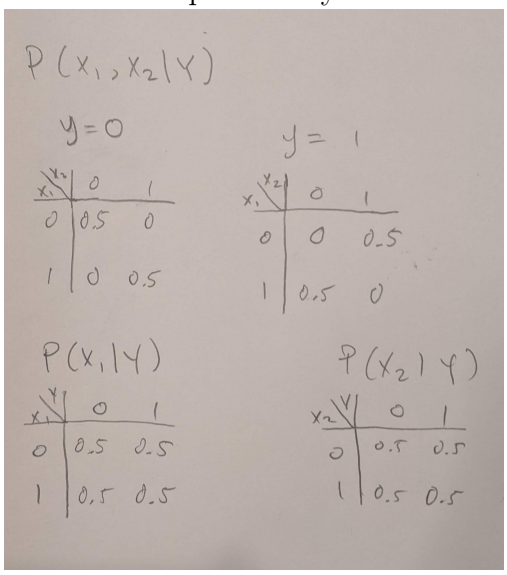
(a) Draw Bayes Net



(b) $p(cough, fever, allergy, flu) = p(cough|allergy, flu)p(fever|flu)p(allergy)p(flu)$

(c) Transition: $p(Z_t = k|Z_{t-1}, u_t = i)$
Observation: $p(Y_t = l|Z_t = j, u_t = i)$
Prior: $p(Z_0 = j) = \Pi_j$

(d) Consider the probability dsitributions below:



Explantion of probability dsitributions:
Given $y = 0$, the two possible combinations of (x_1, x_2) are $[(0, 0), (1, 1)]$.
Given $y = 1$, the two possible combinations of (x_1, x_2) are $[(0, 1), (1, 0)]$.
Given $y = 1$ x_1 can be either 0 or 1. Given $y = 0$ x_2 can be either 0 or 1.
Thus let $x_1 = 0, x_2 = 1, y = 1 \rightarrow p(x_1, x_2|y) = p(0, 1|1) = 0.5$
$p(x_1|y) = p(0|1) = 0.5$
$p(x_2|y) = p(1|1) = 0.5$
$\rightarrow p(x_1|y)p(x_2|y) = p(0|1)p(1|1) = 0.5 * 0.5 = 0.25 \rightarrow p(x_1, x_2|y) \neq p(x_1|y)p(x_2|y)$