Homework 1 Report

Kapoor, Kartik 2462
Nham, Bryan 2494
Panyala, Sukrutha 8740
Vohra, Vedant 2889

Input Data

The input data we chose is the Cancer dataset. A preliminary look at the dataset tells us that the dataset has d = 30 attributes labeled from x1 - x30, and a binary class attribute (Y) labeled g. There is also an attribute labeled patient_id that serves as an index for each observation. We have chosen to omit the patient_id attribute for this homework project since the attribute does not give us any useful information to predict the class attribute g. We initially tested our models with roughly n = 100k patients and then with roughly n = 1 million patients on the provided Linux server. We will report the results we got with 1 million patients.

Pre-processing

The dataset did not have any missing values, so we did not have to modify the dataset to fill in these values. The summary statistics and the histograms of each of the attributes shows that most of the attributes are right-skewed in relation to the observations. Upon taking a closer look at these statistics, it was surprising to see that the maximum values of most attributes are roughly ten times higher than the mean of the attribute, indicating extreme outliers in the dataset. We also normalized the dataset using z-score normalization in order to improve the performance of our models.

PCA

For this section, we will compare the results of PCA done by the built-in library sklearn and our manual implementation.

1. Built-in Implementation

Initially, we wanted to aim for an explained variance of about 90%. After seeing the explained variances for all PCs, we chose to have 6 PCs since that gave us a total explained variance of 88.7% while also having a small amount of attributes to work with. The results of the built-in PCA are shown in the table below. The time taken by this implementation is roughly 3.56 seconds. What surprised us with the results is that x8, the attribute with the most impact on PC1, has a fairly low standard deviation and range of numbers in relation to the other attributes.

PC Number	Attribute with highest absolute weight	Weight value	Explained variance %
1	x8	0.26092494642421954	44.24%
2	x10	0.36596893334576175	18.96%
3	x12	0.3761348286382672	9.40%
4	x22	-0.6336241925240392	6.60%
5	x5	-0.3647151732656096	5.52%
6	x29	0.5020383949573348	4.01%

2. Manual Implementation

We will now discuss the results for the manual implementation of PCA. We chose an explained variance threshold of 1.0 and ended up with 6 PCs, which is unusual since the sum of the explained variances of all 30 PCs should add up to 1. Our manual implementation ended up taking 5.56 seconds, which is about 2 seconds higher than the built-in implementation. The results of the PCA are shown in the table below. We ended up getting completely different attributes with the most impact and weight values, but the same explained variances for each PC, which we found odd. These errors we obtained may be due to a coding error calculating the gamma matrix or from the implementation of the PCA.

PC Number	Attribute with highest absolute weight	Weight value	Explained variance %
1	x30	-0.7026508246728506	44.24%
2	x4	0.6030653578331762	18.96%
3	x30	0.6895726948636929	9.40%
4	x27	0.47716750559215715	6.60%
5	x14	-0.4555235174077809	5.52%
6	x20	0.4741180976630929	4.01%

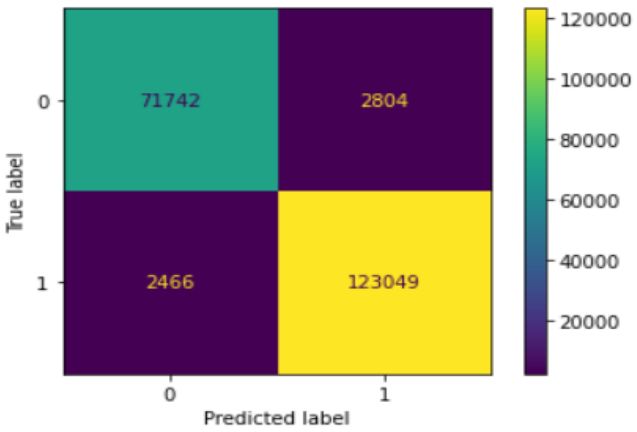
Classification

For this section, we will compare the results of Logistic Regression performed by the built-in library sklearn and our manual implementation of Class Decomposition using K-Means clustering.

Built-in Implementation

```
Built-in Logistic Regression:

Time taken by built-in Logistic Regression:  1.103055100000347
Accuracy of built-in Logistic Regression:  0.9736580342995387
Confusion Matrix of built-in Logistic Regression:
[[ 71742   2804]
 [  2466 123049]]
```



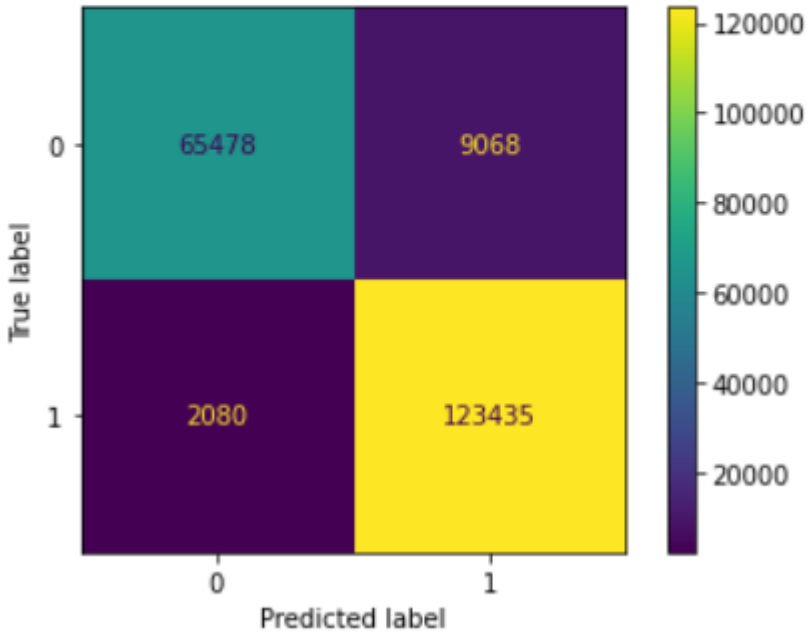
The Built-in implementation has high accuracy here as Logistic Regression performs well when the dataset is linearly separable.

Manual Implementation

In our implementation of Class Decomposition using K-Means clustering, we first divided the input dataset based on the 2 target classes (0 and 1) and then proceeded onto the clustering. We have considered 5 clusters for our model. We have used randomly selected datapoints as the initial centroid. K-Gamma Matrices (K=5) were used to calculate means for each cluster. Finally, we have chosen to perform a single iteration to strike a fair balance between performance and accuracy. The results we obtained are given below.

```
Gamma-based K-Means:

Time taken by gamma-based K-Means:  17.763017499999478
Accuracy of gamma-based K-Means:  0.9442769955163675
Confusion Matrix of gamma-based K-Means:
[[ 65478   9068]
 [  2080 123435]]
```



Responsibilities

Kapoor, Kartik: CLI, Testing
Nham, Bryan: Gamma computation, Analysis, Report
Panyala, Sukrutha: Analysis, Report
Vohra, Vedant: Model implementation, Documentation, Video