

# Assignment 2

Nicholas Pysklywec

March 5, 2023

## 1 Generative Models, Naive Bayes Classifier

### Part A

We are asked to compute  $p(\text{Water} = \text{warm} | \text{Play} = \text{yes})$  and  $p(\text{Water} = \text{warm} | \text{Play} = \text{no})$ . In doing this we achieve the following:

$$p(\text{Water} = \text{warm} | \text{Play} = \text{yes}) = \frac{2}{3}$$

$$p(\text{Water} = \text{warm} | \text{Play} = \text{no}) = 1$$

### Part B

We are asked to compute  $p(\text{Play} = \text{yes} | \text{Water} = \text{warm})$  and  $p(\text{Play} = \text{no} | \text{Water} = \text{warm})$ . In doing this we achieve the following:

$$p(\text{Play} = \text{yes} | \text{Water} = \text{warm}) = \frac{2}{3}$$

$$p(\text{Play} = \text{no} | \text{Water} = \text{warm}) = \frac{1}{3}$$

### Part C

We are asked to compute  $p(\text{Play} = \text{yes} | \text{Forecast} = \text{same})$  and  $p(\text{Play} = \text{yes} | \text{Forecast} = \text{change})$ . In doing this we achieve the following:

$$p(\text{Play} = \text{yes} | \text{Forecast} = \text{same}) = 1$$

$$p(\text{Play} = \text{yes} | \text{Forecast} = \text{change}) = \frac{1}{2}$$

## Part D

We are asked to compute  $p(\text{Play} = \text{yes} | \text{Water} = \text{warm})$  and  $p(\text{Play} = \text{no} | \text{Water} = \text{warm})$ , however this time we apply Laplace smoothing. We add 1 to the numerator, and C categories to the denominator. In doing this we achieve the following:

$$p(\text{Water} = \text{warm} | \text{Play} = \text{yes}) = \frac{2+1}{3+6} = \frac{1}{3}$$

$$p(\text{Water} = \text{warm} | \text{Play} = \text{no}) = \frac{1+1}{1+6} = \frac{2}{7}$$

## 2 Kernels

### Part A

$k_1(x, z)$  and  $k_2(x, z)$  are valid kernels. Valid kernels are defined as being positive semidefinite and symmetric. Positive semidefinite matrices are defined as having non-negative eigenvalues. In the question,  $k(x, z)$  is an invalid kernel.

Start with the defined kernel function:

$$k(x, z) = a_1 k_1(x, z) - a_2 k_2(x, z) \quad (1)$$

Set  $a_1 = 1$  and  $a_2 = 2$ , and let  $k_1(x, z) = k_2(x, z)$ :

$$k(x, z) = k_1(x, z) - 2k_1(x, z) \quad (2)$$

$$k(x, z) = -k_1(x, z) \quad (3)$$

Step 3 demonstrates the kernel as invalid. Since  $k_1$  was positive definite,  $-k_1$  would render the matrix not positive semidefinite, and therefore the kernel  $k$  invalid.

Positive semidefinite matrices have a positive trace value. The trace is defined as the sum of diagonal elements of the matrix. With the negative sign application to  $k_1$  effectively, we are also reversing the sign of the trace, which since  $k_1$  is positive semidefinite, must be positive, and in reversing it, make it negative. The trace of a matrix, is also the sum of its eigenvalues. If a trace was negative at least one eigenvalue would be negative.  $-k_1$  therefore has at least one negative eigenvalue, which would make it certainly not positive semi-definite. Since  $k(x, z) = -k_1(x, z)$ ,  $k(x, z)$  is an invalid kernel.

### Part B

We construct the feature mapping to demonstrate the kernel is valid. For kernels, we must be able to write it as a dot product of vectors in a high dimensional feature space defined by  $\theta$ . We set out in doing this:

$$k(x, z) = f_1(x)f_1(z) + f_2(x)f_2(z) \quad (1)$$

The feature mapping consists of  $\theta(x), \theta(z)$ .

$$\theta(x) = (f_1(x), f_2(x)), \theta(z) = (f_1(z), f_2(z)) \quad (2)$$

We write the kernel in the form  $k(x, z) = \theta(x)^T \theta(z)$ :

$$k(x, z) = (f_1(x), f_2(x))^T (f_1(z), f_2(z)) \quad (3)$$

Since we explicitly can write the kernel function as a feature mapping directly, so it is therefore a valid kernel.

## Part C

We are given the following kernel which is valid:

$$k_1(x, z) = e^{\frac{x^T z}{\sigma^2}} \quad (1)$$

We want to prove this gaussian kernel is valid:

$$k_2(x, z) = e^{-\frac{\|x-z\|_2^2}{2\sigma^2}} \quad (2)$$

We know  $k_1$  is a valid kernel, we want to demonstrate how the gaussian kernel is equivalent to show the gaussian kernel is also valid. We start by expanding  $\|x-z\|_2^2$  in  $k_2$ , which is the euclidean distance:

$$\|x-z\|_2^2 = \left( \sqrt{\sum_{i=1}^m (x_i - z_i)^2} \right)^2 = \left( \sqrt{\sum_{i=1}^m (x_i^2 + z_i^2 - 2x_i z_i)} \right)^2 \quad (3)$$

We have the  $x, z$  norms and the dot product of  $x, z$  above which we can replace:

$$\|x-z\|_2^2 = (\|x\| + \|z\| - 2x^T z) \quad (4)$$

$$\|x-z\|_2^2 = x^T x + z^T z - 2x^T z \quad (5)$$

Replace into equation:

$$k_2(x, z) = \exp\left(-\frac{1}{2\sigma^2}(x^T x + z^T z - 2x^T z)\right) \quad (6)$$

$$k_2(x, z) = \exp\left(-\frac{x^T x}{2\sigma^2}\right) \exp\left(\frac{x^T z}{\sigma^2}\right) \exp\left(-\frac{z^T z}{2\sigma^2}\right) \quad (7)$$

There exists kernel properties such that  $k(x, z) = f(x)k_3(x, z)f(z)$ . We use this on the above, and therefore only need to demonstrate  $k_3$  is valid.  $k_3$  here is the middle item from above, so we are left with:

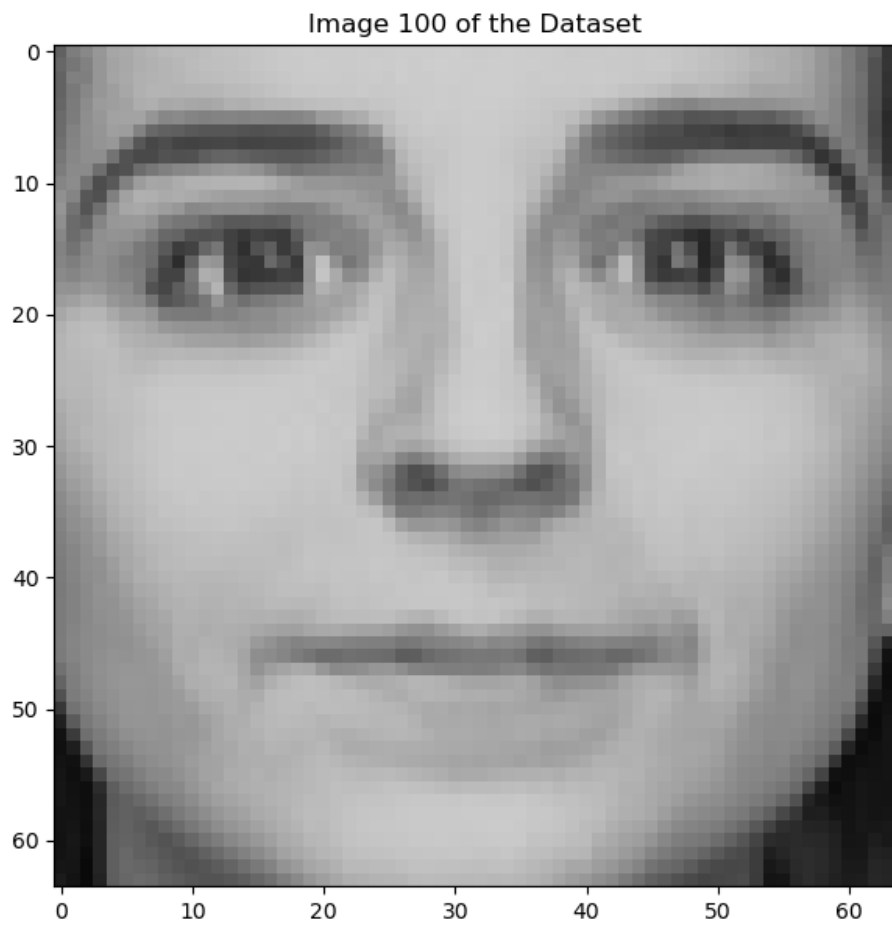
$$k_3(x, z) = e^{\left(\frac{x^T z}{\sigma^2}\right)} \quad (8)$$

This happens to be  $k_1$ , which we were told was a valid kernel in the question.  $k_3$  is a valid kernel making  $k_2$  a valid kernel. Therefore the gaussian kernel is a valid kernel.

### 3 PCA and Eigenface

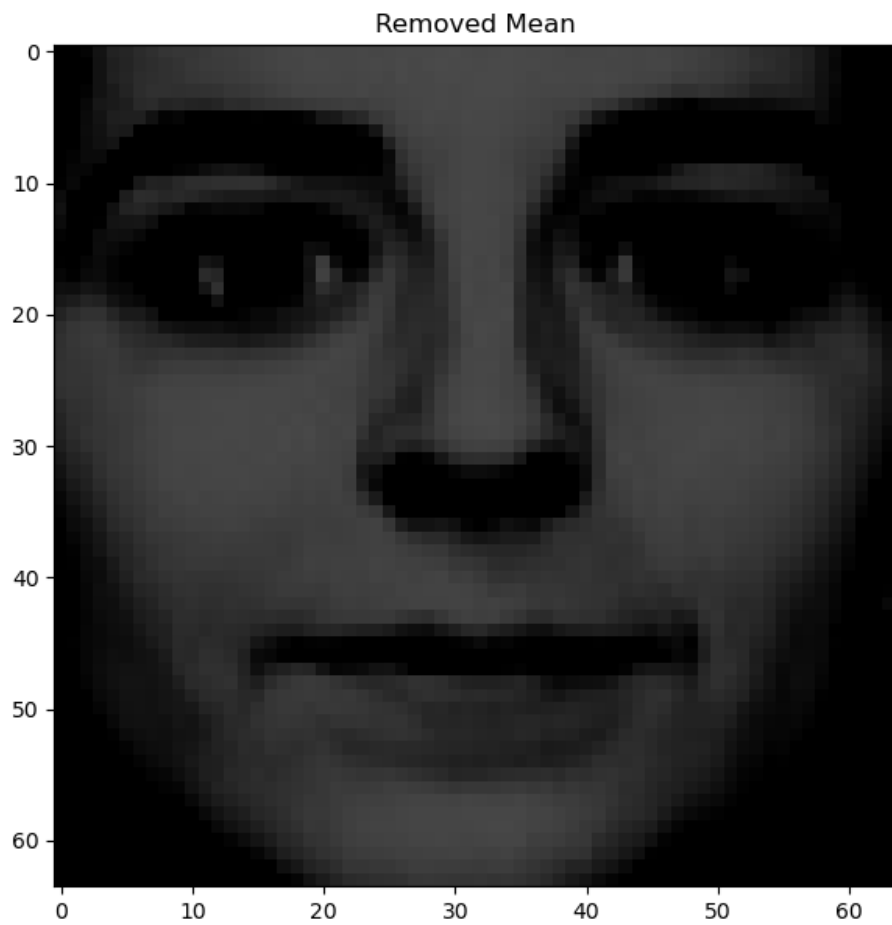
#### Part A

We display the 100th image in the Python notebook:



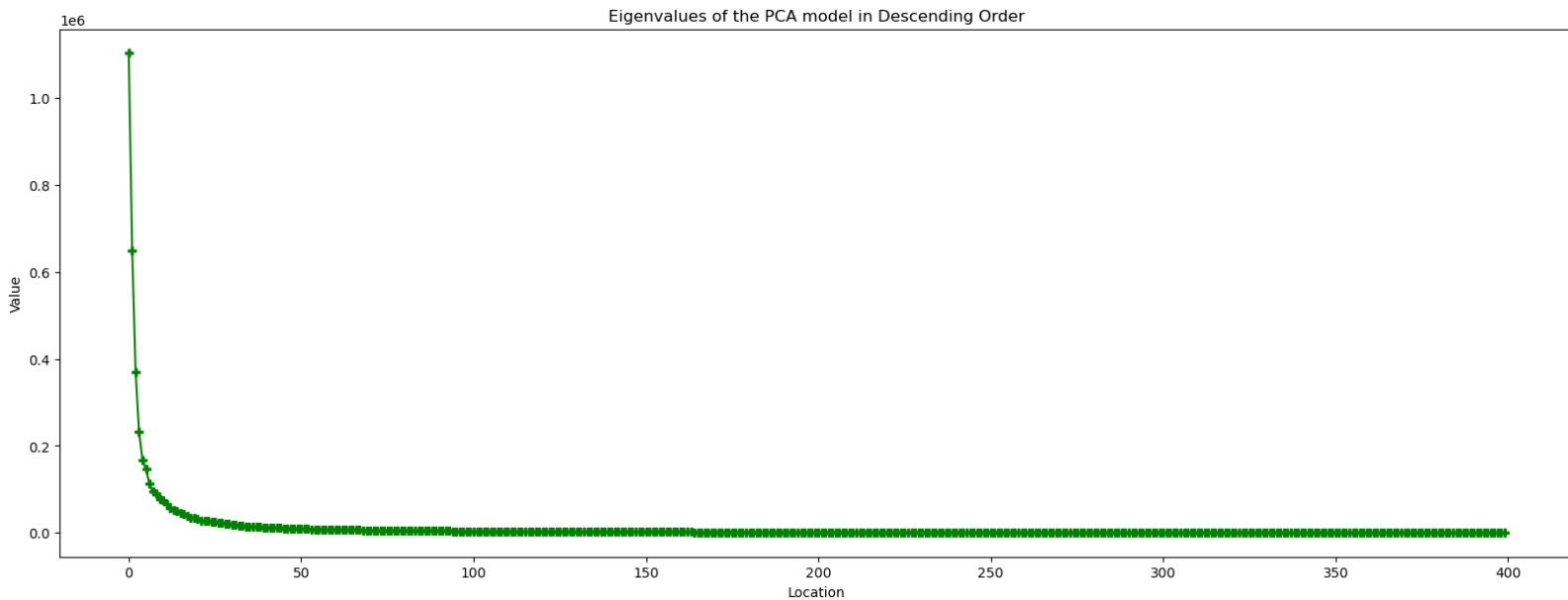
## Part B

We remove the mean of the images, then demonstrated the change in images by having another look at image 100:



### Part C

PCA is done using an existing ML toolbox Sklearn. Specifically we make use of [this](#) part of the library. The eigenvalues are collected after performing PCA, and then plotted in descending order.



## Part D

The last eigenvalue is zero because there exists no variance captured in this component. Specifically, we have a correlation/covariance matrix  $X^T X$  for the sample data here, and  $n$  eigenvalues. We sorted the data, and the last value here demonstrates the component provides the least information relative to all other components.

```
#Look at value 400
print("400th Eigenvalue:", reverse_array[399])
```

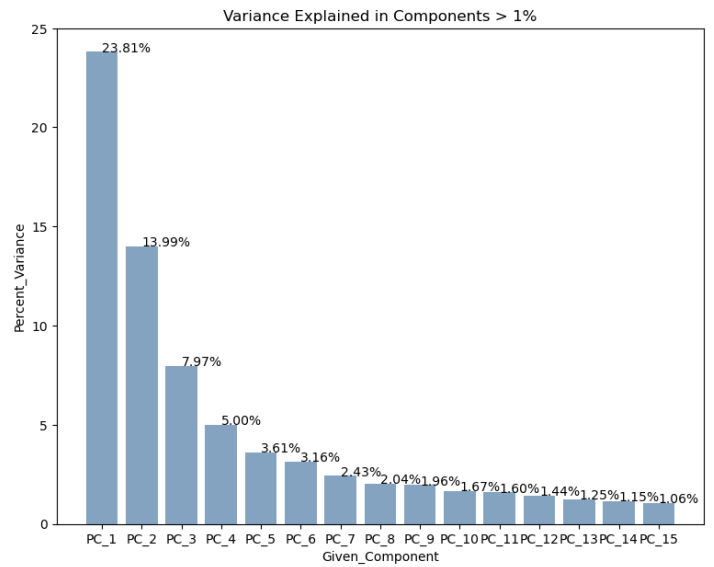
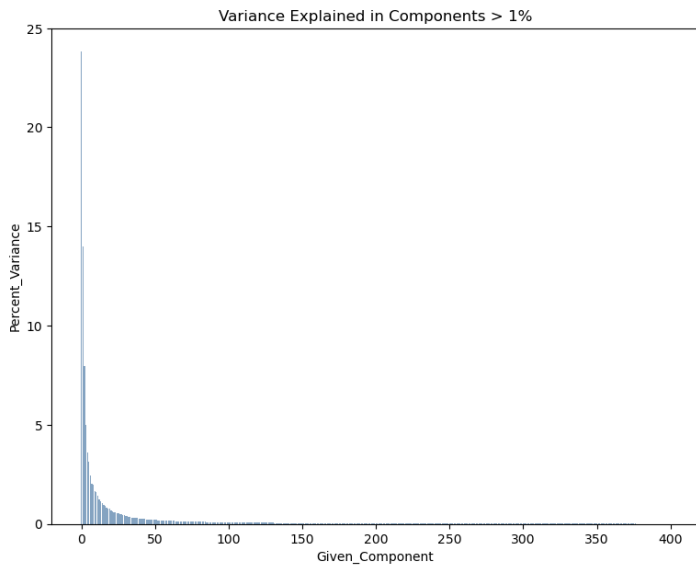
✓ 0.0s

400th Eigenvalue: 1.6076397455156887e-26

## Part E

The variance captured in components is plotted amongst each component on the left. The most relevant components (with captured variance  $\geq 1\%$ ) are plotted on the right to demonstrate some of the largest components without too much redundancy)

## Analysis of Variance in Components

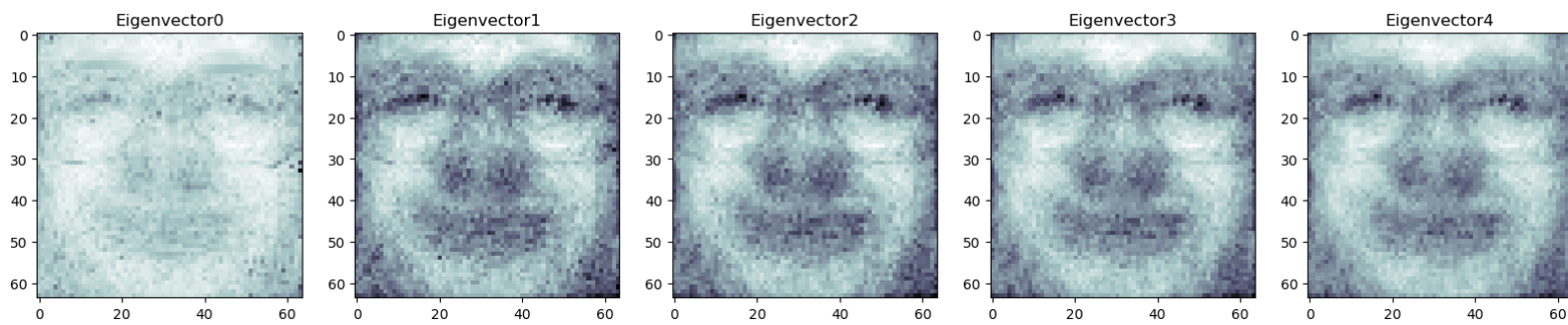


85% of the variance is captured in the first 40 components of the data(done in python). The number of components we use is slightly subjective on the use case. If we perhaps don't need as much information we could get away only with the first four components, which account for 50% of the variance in the data.

## Part F

The five top eigenvectors are plotted below from largest eigenvector to smallest. These should demonstrate a lot of the variance in the data.

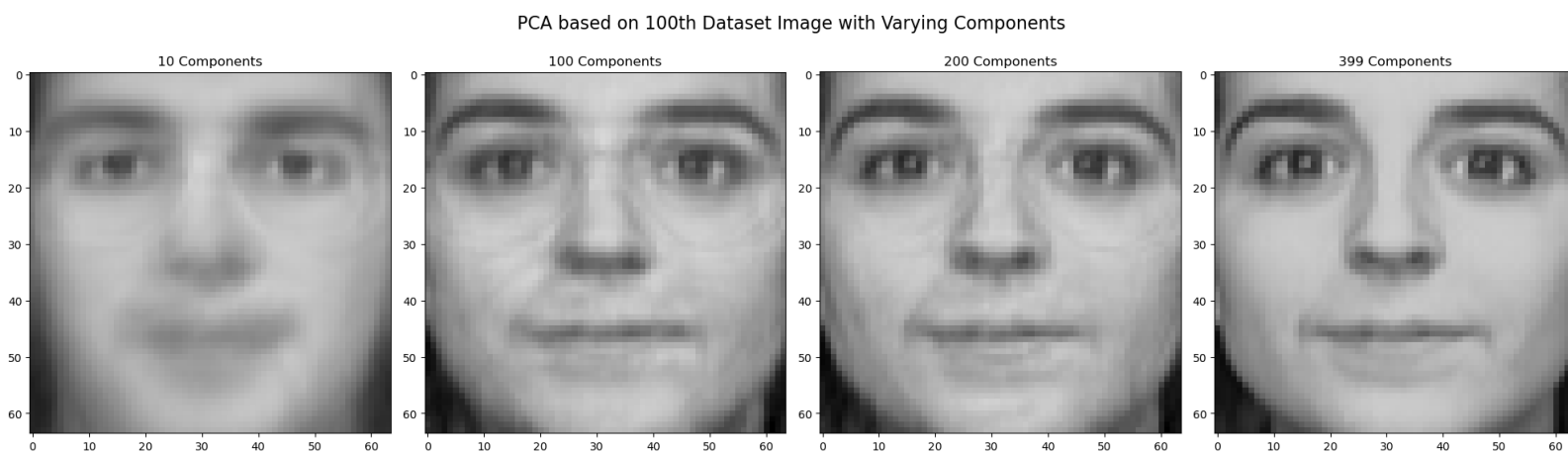
### Top Five Eigenvectors





## Part G

Using the PCA toolbox from Sklearn, we perform PCA with 10, 100, 200, and 399 components on the dataset. To demonstrate the additions of each set of component to the data, we will plot image 100 over each use of components from least to greatest.



Greater components clearly add more information regarding image 100. We see more detail, starting with a relatively plain human face to a more detailed human women.