Aprendizaje automático

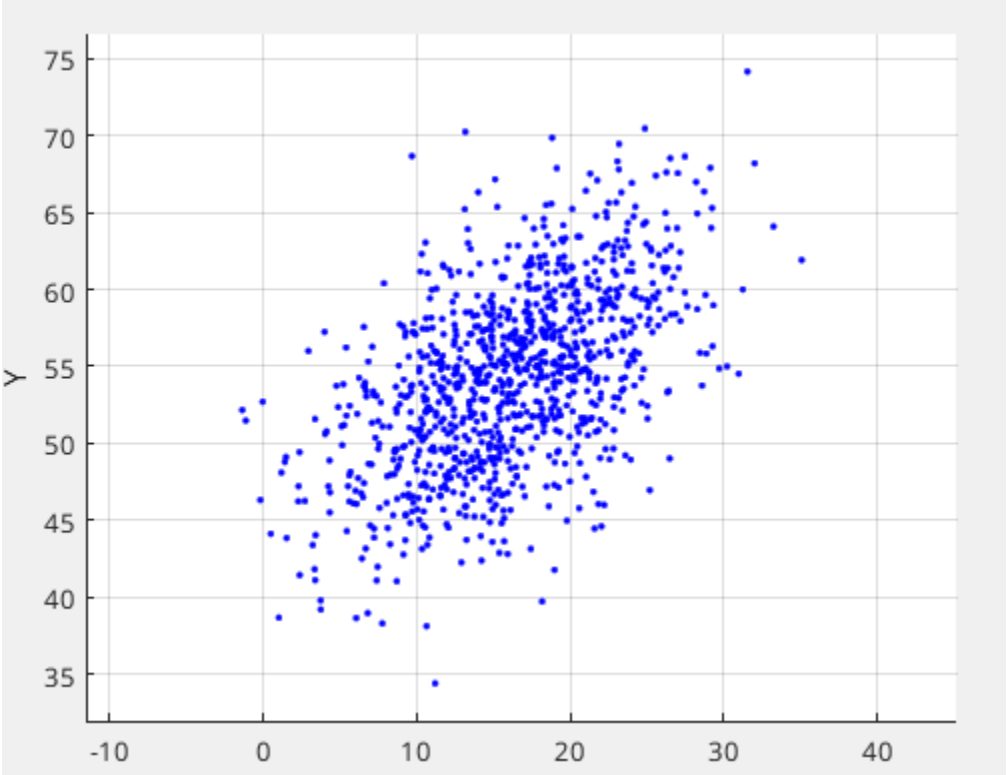
- práctica 6 -

Por:

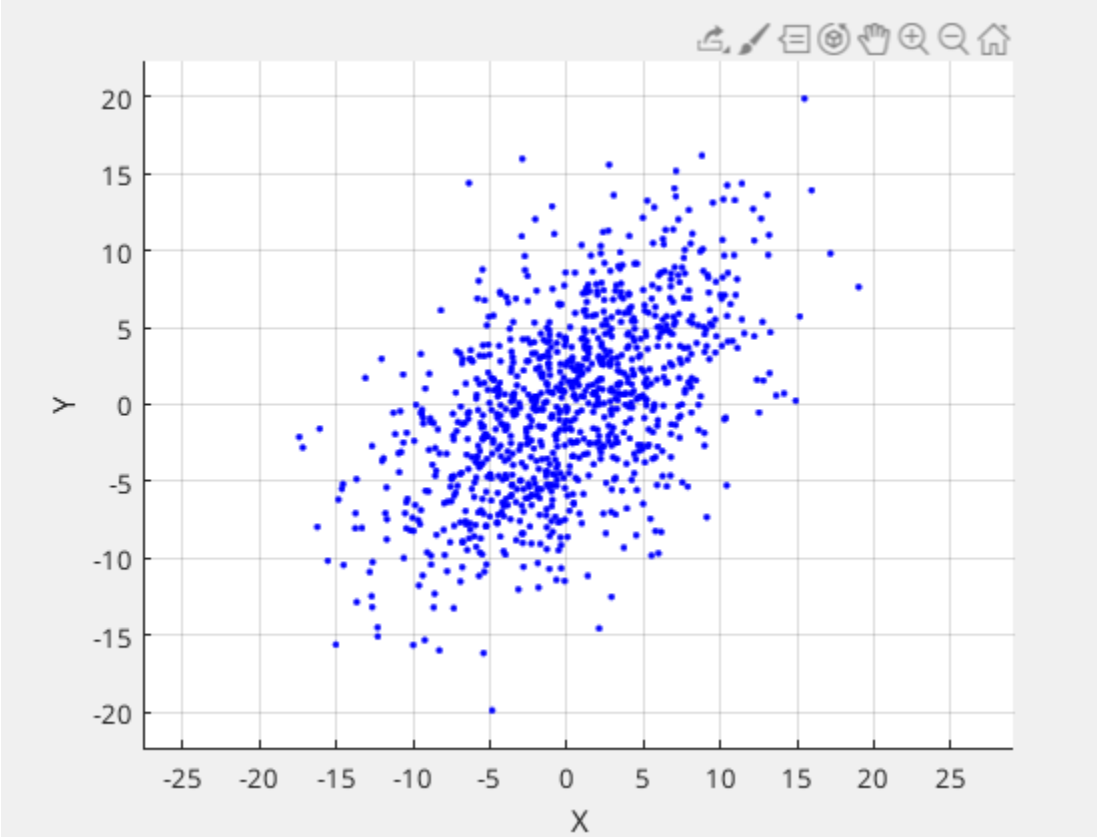
Diego Caballé Casanova (738712)

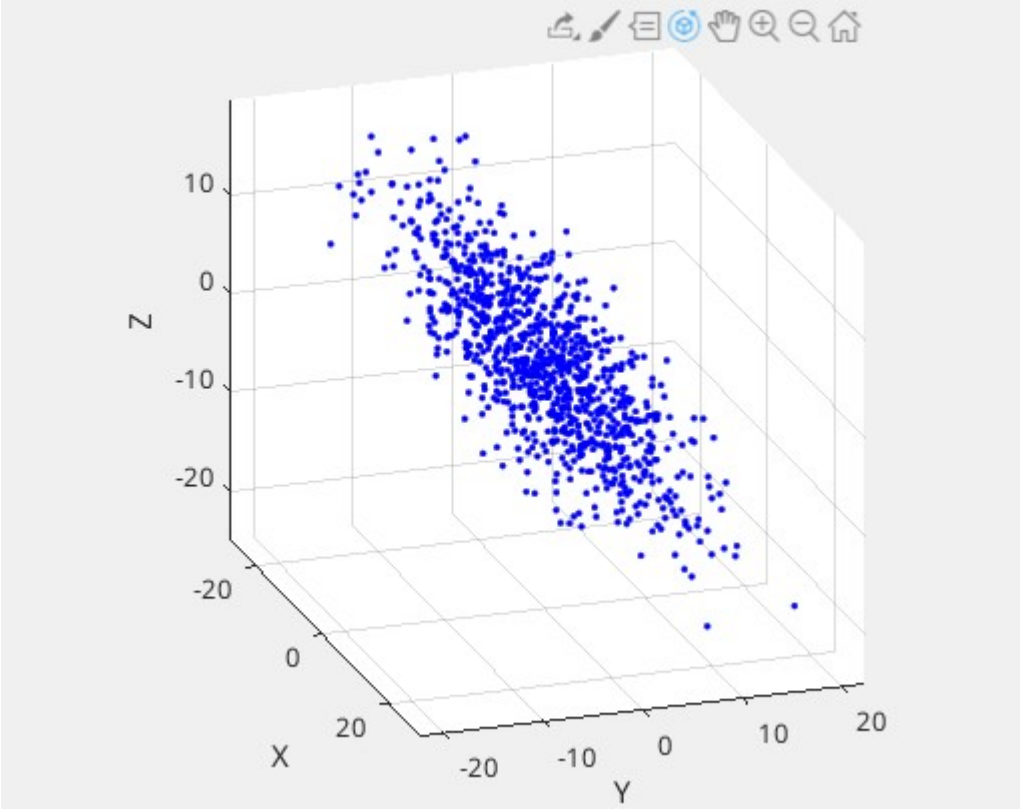
Apartado 1 (P61)

Estos son los datos originales:

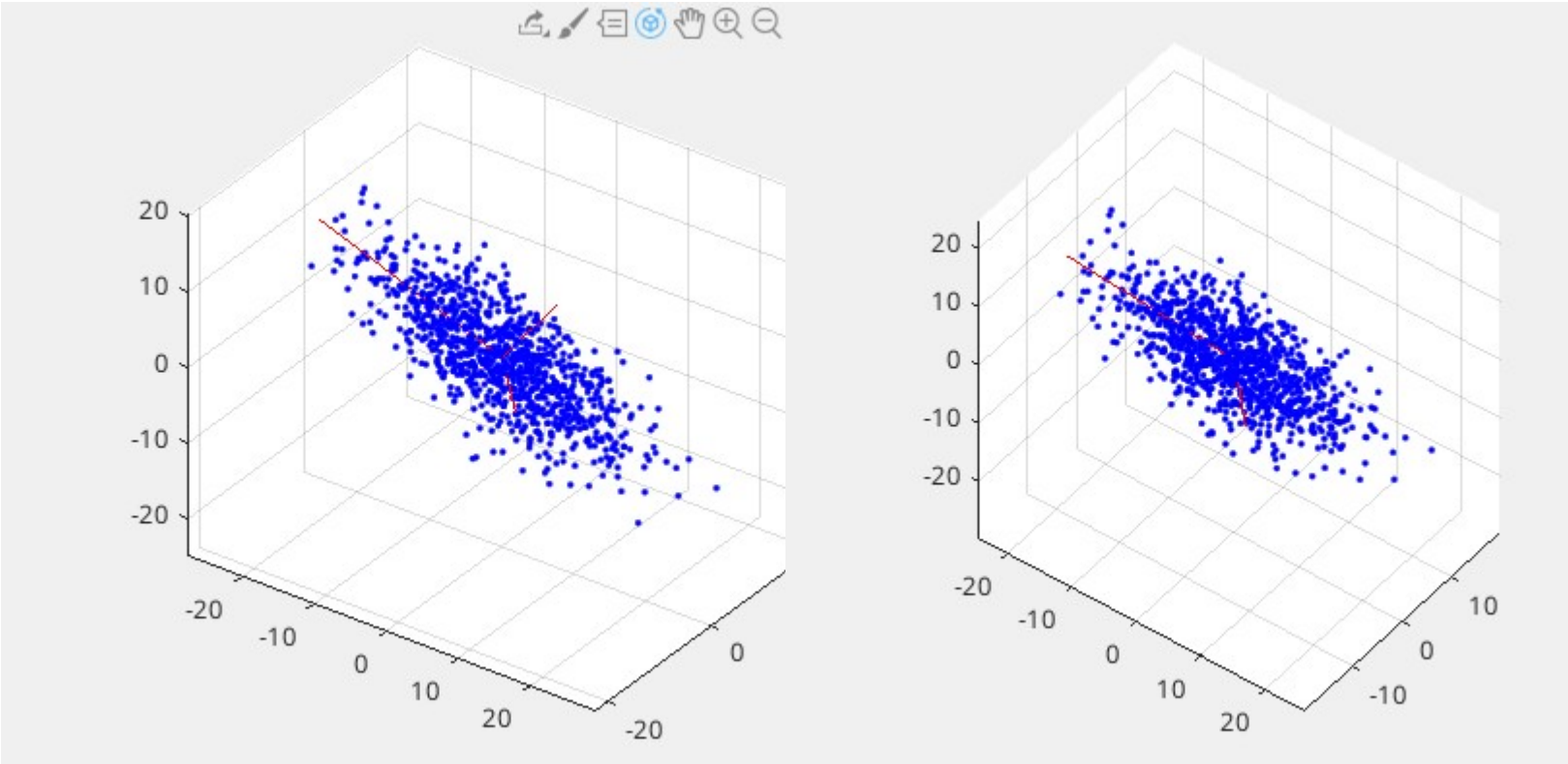


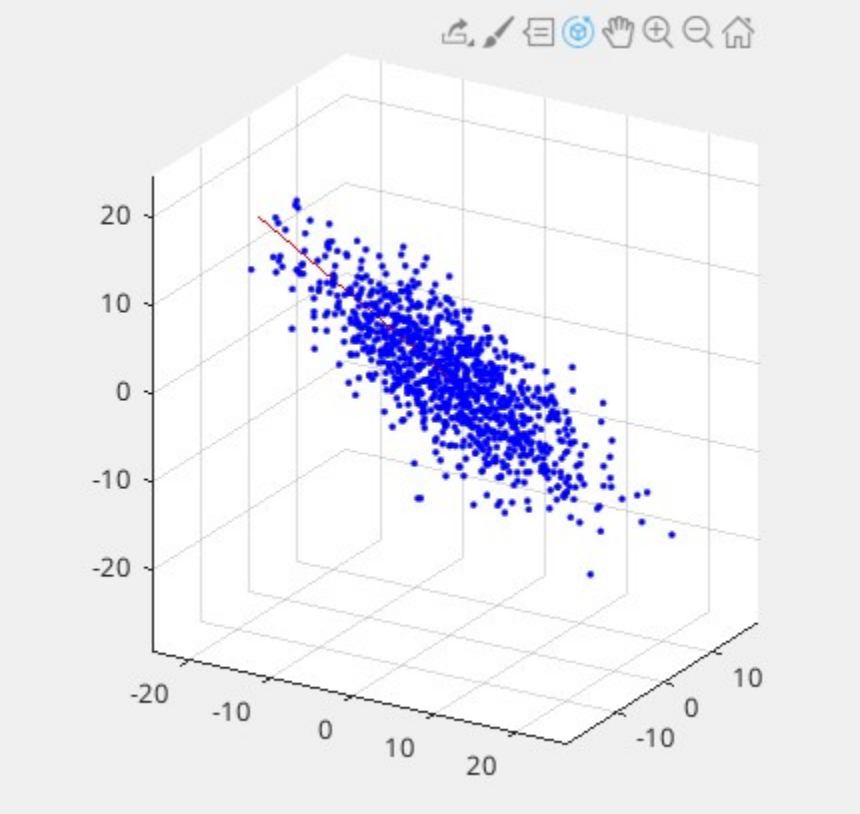
El primer paso para hacer realizar PCA es estandarizar los datos, centrandolos, la gráfica estandarizada quedaría así:



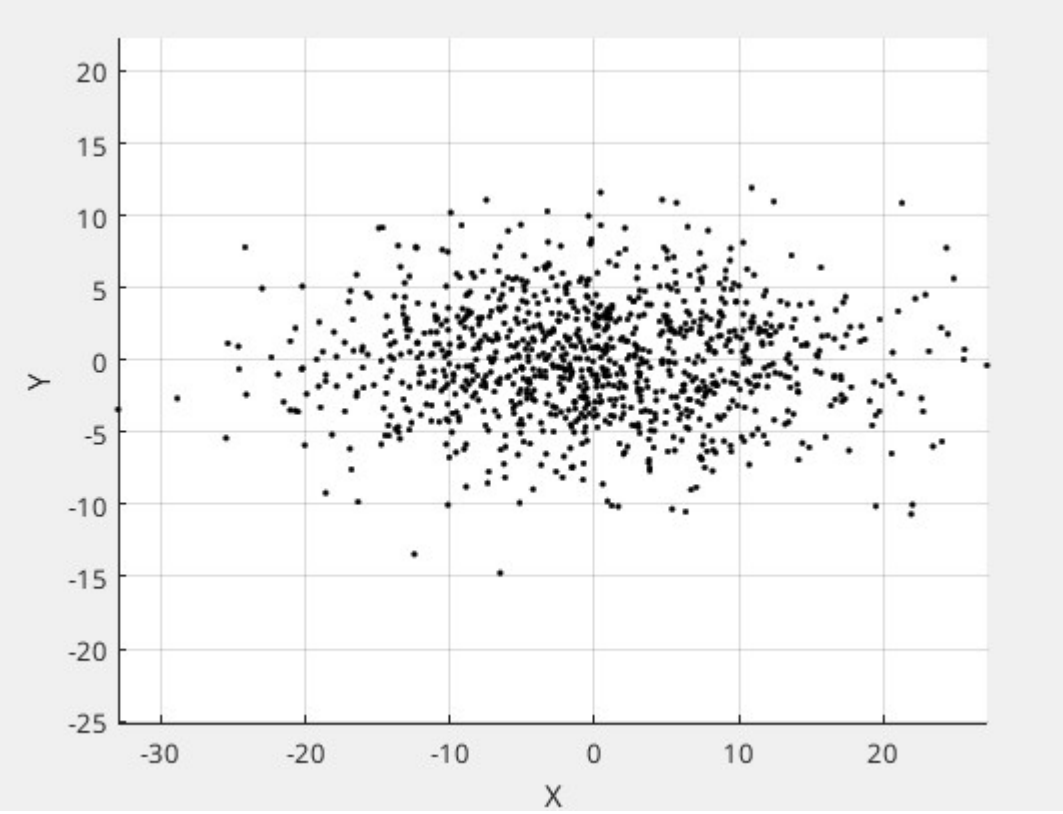


Para poder realizar la reducción de dimensión, necesitamos obtener los vectores propios a los que podremos reproyectar los datos. Los vectores propios son estos, de más a menos importantes.

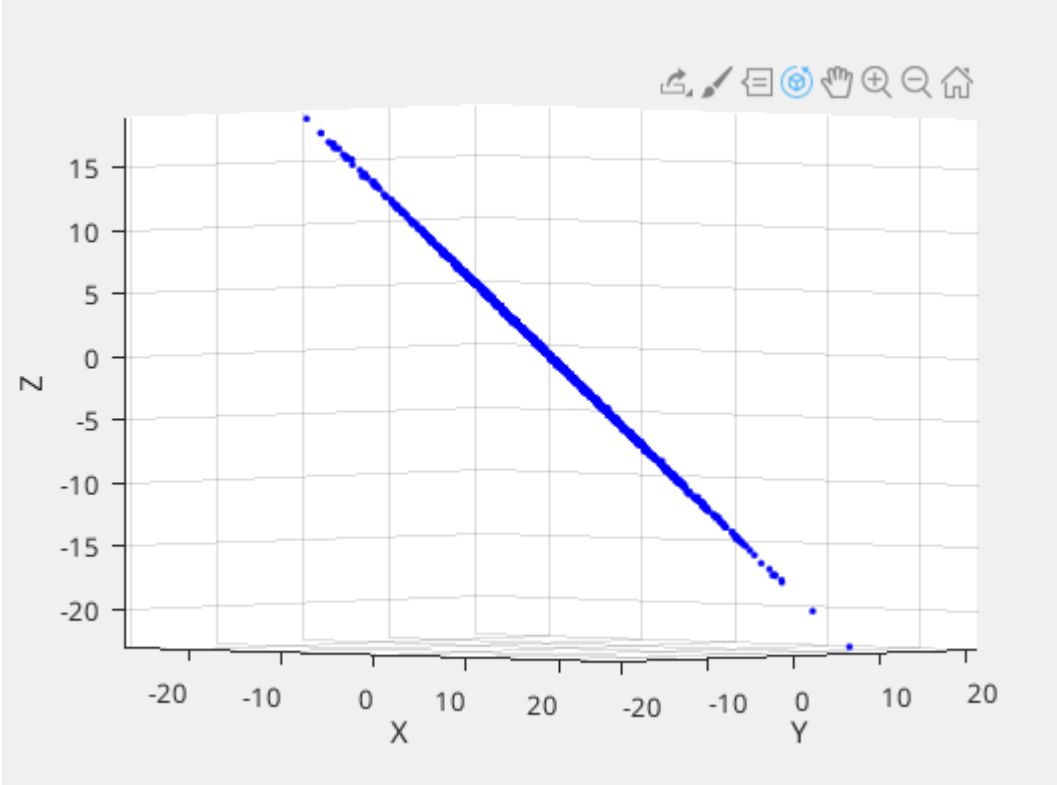


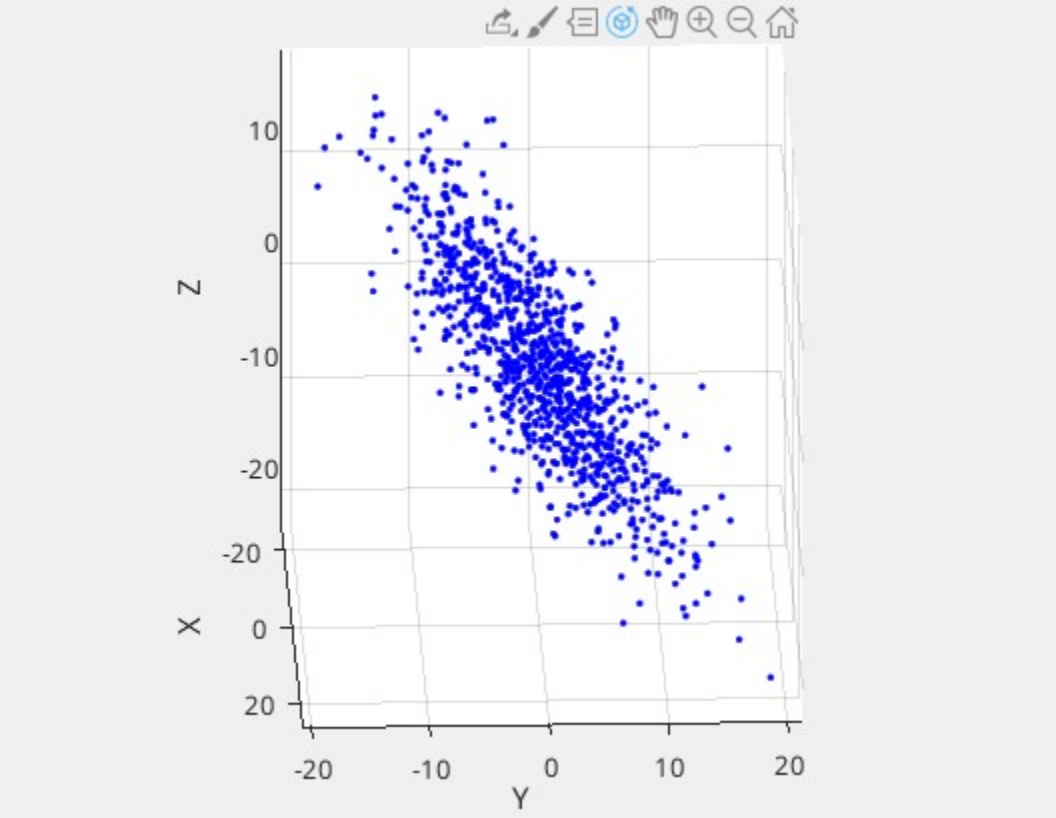


Una vez hemos visto los vecotres propios, vemos que los dos últimos son los más importantes, sobre estos dos vectores propios, crearemos la nueva base sobre la que transformaremos cada dato, obteniendo Z, como la transformación de cada dato en esta nueva base.



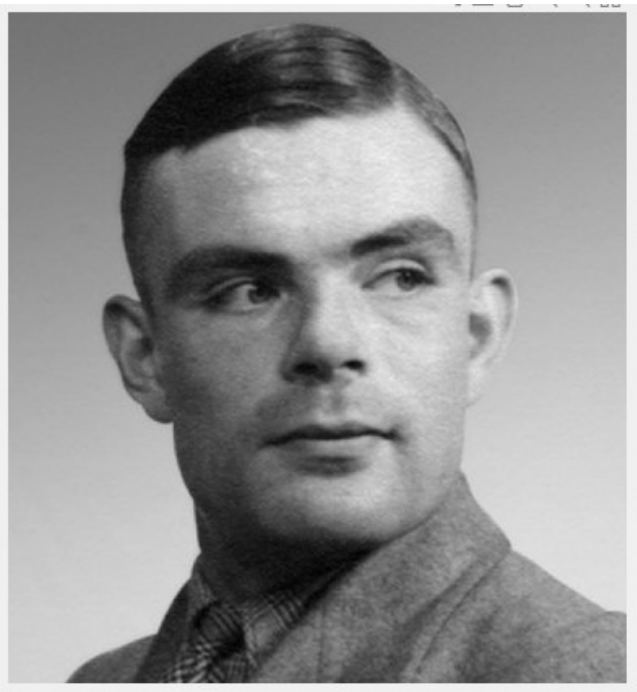
Seguidamente,utilizando esta formula Xgorrot=U*Zt; podemos conseguir reconstruir los datos, obteniendo los datos con una dimensión menos, pero representada en el espacio tridimensional inicial.



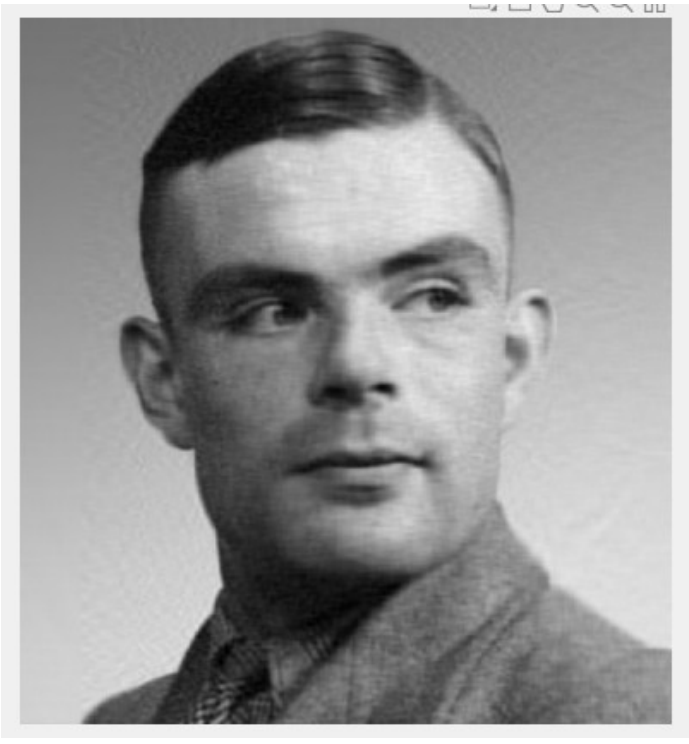


Apartado 2 (P62)

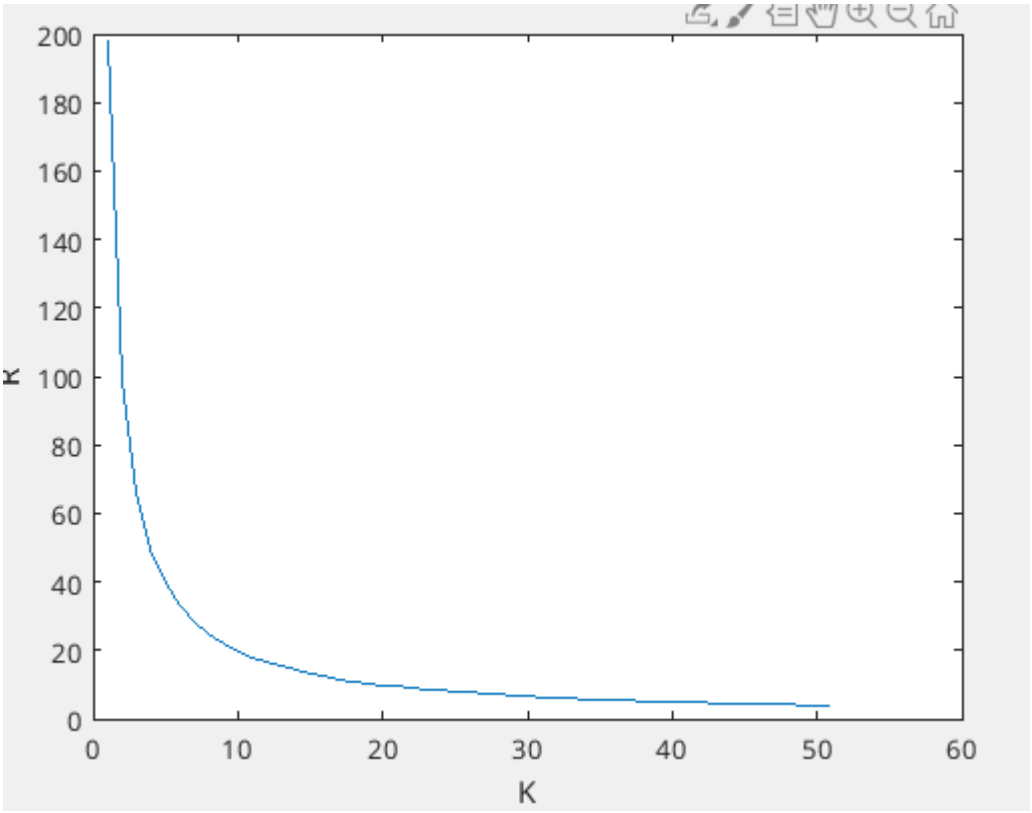
En este apartado comprimiremos esta imagen mediante SVD



Esta imagen, la podemos descomponer en vectores utilizando la función de matlab SVD(), que contienen el espacio de las filas, el espacio de las columnas y los valores singulares de la matriz de esta foto. A partir de aquí, si elegimos los suficientes valores singulares, junto con las filas y las columnas, podemos conseguir una foto similar que ocue menos espacio. El número de valores singulares a escoger lo realizaremos como en el apartado anterior, decidiendo que 51 es suficiente para tener una semajanza del 90% con la foto anterior. La foto resultante de elegir 51 valores singulares es:



Aunque a simple vista parecen idénticas, es cierto que la cálidad de la imagen ha disminuido, pero lo que hemos conseguido es comprimir la imagen en un ratio de 3,876, es decir, la segunda imagen es 4 veces más ligera. Para visualizar la relación de compresión frente a los valores propios obtenemos la siguiente gráfica:

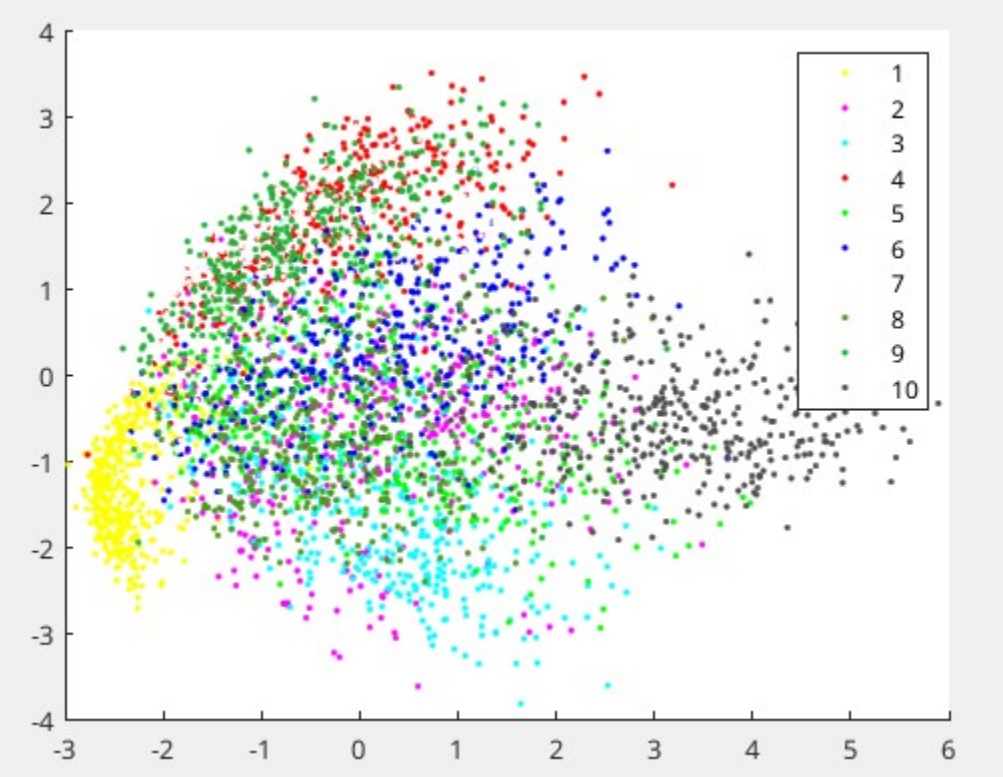


Como vemos, el valor de nuestra k, 51, es justo cuando la imagen se empieza a ver bien, pero se consigue maximizar el ratio de compresión. Obviamente, si eligieramos 5 valores singulares la foto sería mucho más ligera, pero el resultado sería este:

Apartado 3 (P63)

En este apartado, tendremos que repetir la clasificación de números de la práctica 5, pero reduciendo la dimensión de los atributos de estos números.

Repetiremos los mismos pasos del PCA, obteniendo de nuevo la transformación de cada dato a la nueva base. En este caso, dado que los atributos son los pixeles, elegimos los dos pixeles más representativos y mostramos la distribución de los datos sobre estos dos atributos.



En base a esta distribución, podemos decidir 2 clases que se podrán diferenciar bien y dos clases que no. Esto lo sabemos en función de lo juntos o dispersos que están los números. Por ejemplo, los datos a clasificar podrían ser el 1 y el 10 que están muy distantes, y el 4 y el 8 que están muy mezclados. Después de clasificarlos obtenemos la siguiente matriz de confusión para:

Clasificar 1 y 10

```
matrizDeConfusion =

   100     0     0     0     0     0     0     0     0     0   100
     0     0     0     0     0     0     0     0     0     0     0
     0     0     0     0     0     0     0     0     0     0     0
     0     0     0     0     0     0     0     0     0     0     0
     0     0     0     0     0     0     0     0     0     0     0
     0     0     0     0     0     0     0     0     0     0     0
     0     0     0     0     0     0     0     0     0     0     0
     0     0     0     0     0     0     0     0     0     0     0
     0     0     0     0     0     0     0     0     0     0     0
     1    99     0     0     0     0     0     0     0     0   100
   101    99     0     0     0     0     0     0     0     0   200
```

clasificar 4 y 8

```
matrizDeConfusion =

     0     0     0     0     0     0     0     0     0     0     0
     0     0     0     0     0     0     0     0     0     0     0
     0     0     0     0     0     0     0     0     0     0     0
    70   130     0     0     0     0     0     0     0     0   200
     0     0     0     0     0     0     0     0     0     0     0
     0     0     0     0     0     0     0     0     0     0     0
     0     0     0     0     0     0     0     0     0     0     0
     0     0     0     0     0     0     0     0     0     0     0
     0     0     0     0     0     0     0     0     0     0     0
     0     0     0     0     0     0     0     0     0     0     0
    70   130     0     0     0     0     0     0     0     0   200
```

ciertamente, podemos ver como la clasificación del 1 y el 10 ha sido casi perfecta, mientras que la clasificación del 4 y 8, no sólo se ha confundido muchisimo, como nos daba a intuir la gráfica anterior.

Pero ahora nos surge la duda de si estos datos comprimidos influyen a la hora de clasificar todos los datos. Después de realizar la prueba obtenemos un f1 de 0,74, que es ciertamente peor al 0,94 que obteniamos con la matriz de covarianzas completa, y no tan malo como el 0,84 que se obtenía con las matrices en bayes ingenuo. Así quedaría la matriz de confusiones:

```
matrizDeConfusion =

     0     2     1     1     1     1     0    30     2     0    38
     0    74     2     1     4     2     0    26     6     0   115
     0     1    55     1     0     2     1    21     4     1    86
     1     8     0    40     2     3     0    21     3     0    78
     0     4     0     3    58     1     1    23     8     0    98
     0     3     0     0     2    80     0    22     4     1   112
     0     4     1     2     2     0    68    32     2     1   112
     0     4     1     1     2     1     0   101     4     0   114
     0     3     5     2     4     0     1    23    91     1   130
     0     2     0     0     1     1     2    28     6    77   117
     1   105    65    51    76    91    73   327   130    81  1000
```