Benemérita Universidad Autónoma de Puebla - FCC

Minería de Datos

Implementación de Ensamble Parte I. Aplicación de PCA

ABRIL - 2020

Emiliano Carrillo Moncayo

Tabla de Contenidos

1. Introducción 3
2. Importación de Datos 4
3. Preprocesar Datos 5
4. PCA 6
    - 4.1 Aplicación del PCA 6
    - 4.2 Análisis de resultados 6
    - 4.3 Visualización del PCA 2D 7
5. Detección de Outliers 8
6. Clasificación 10
    - 6.1 Clasificar 10
    - 6.2 Predicción y Análisis 11

PARA UNA EXPLICACIÓN PASO A PASO, VER EL JUPITER NOTEBOOK:

HTTPS://EMILIANOCARRILLO.GITHUB.IO/ENSAMBLE-PARTE-1/#TABLA-DE-CONTENIDOS

1. Introducción

PCA es un algoritmo de reducción de dimensionalidad no supervisado. En este ejemplo lo emplearemos para satisfacer 3 objetivos:

- **Hacer un análisis**

  Analizaremos la cantidad de componentes principales que son necesarios para resumir nuestros datos de una manera que no haya tanta pérdida de información. Este objetivo se establece únicamente como fase de experimentación.

- **Facilitar la visualización de nuestros datos**

  Ocuparemos 2 componentes principales para poder graficar nuestros datos multidimensionales en una gráfica de dispersión. Así podremos entender mejor nuestra data y podemos detectar posibles outliers.

- **Como paso de preprocesamiento para la clasificación de elementos**

  Por último ocuparemos PCA para resumir nuestros datos y eficientizar el entrenamiento de nuestro clasificador sin perder efectividad de clasificación.

2. Importación de Datos

El primer paso es importar los datos que ocuparemos para el análisis. El archivo de entrada debe ser un archivo de texto plano con el formato siguiente:

```
No. Elementos
No. Atributos
No. Clases
atrib_0, atrib_1, ..., atrib_n, clase
atrib_0, atrib_1, ..., atrib_n, clase
... ... ...
atrib_0, atrib_1, ..., atrib_n, clase
```

Preprocesar archivo¶

Primero preprocesamos el archivo para obtener los metadatos de No. de elementos, atributos y clases que éste contiene en el encabezado y así construir nuestro dataset.

Podemos obtener un pequeño vistazo de cómo se ve nuestro dataset hasta ahora.

	atrib_1	atrib_2	atrib_3	atrib_4	atrib_5	atrib_6	atrib_7	atrib_8	atrib_9	atrib_10	atrib_11	atrib_12	atrib_13	atrib_14	atrib_15	atrib_16	atrib_17	atrib_18	atrib_19	clase
0	140	125	0	0.0	0.0	0.277778	0.062963	0.666667	0.311111	6.185185	7.333334	7.666666	3.555556	3.444444	4.444445	-7.888889	7.777778	0.545635	-1.121818	0
1	188	133	0	0.0	0.0	0.333333	0.266667	0.500000	0.077778	6.666666	8.333334	7.777778	3.888889	5.000000	3.333333	-8.333333	8.444445	0.538580	-0.924817	0
2	105	139	0	0.0	0.0	0.277778	0.107407	0.833333	0.522222	6.111111	7.555555	7.222222	3.555556	4.333334	3.333333	-7.666666	7.555555	0.532628	-0.965946	0
3	34	137	0	0.0	0.0	0.500000	0.166667	1.111111	0.474074	5.851852	7.777778	6.444445	3.333333	5.777778	1.777778	-7.555555	7.777778	0.573633	-0.744272	0
4	39	111	0	0.0	0.0	0.722222	0.374074	0.888889	0.429629	6.037037	7.000000	7.666666	3.444444	2.888889	4.888889	-7.777778	7.888889	0.562919	-1.175773	0

DATASET ORIGINAL

3. **Preprocesar Datos**

Como PCA se soporta de la desviación estándar de los datos para calcular la nueva proyección de nuestros datos, una variable con una desviación estándar alta tendrá un peso mayor para el cálculo de la proyección que una variable con una desviación estándar baja. Si normalizamos los datos, todas las variables tendrán la misma desviación estándar, por lo tanto, el cálculo no estará cargado.

Además, como no tenemos conocimiento del dominio del conjunto de datos de ejemplo, no sabemos si las unidades de medida de sus variables son distintas. Otra razón por la cual normalizar nuestros datos.

PCA se considera como un algoritmo no supervisado, esto quiere decir que se apoya únicamente del set de datos sin las clases asignadas. Por esto, el primer paso de preprocesamiento será dividir nuestro set en dos: el set con los atributos y el set de las clases de asignación. Paso continuo sería estandarizar los datos sin la columna de las clases.

	atrib_1	atrib_2	atrib_3	atrib_4	atrib_5	atrib_6	atrib_7	atrib_8	atrib_9	atrib_10	atrib_11	atrib_12	atrib_13	atrib_14	atrib_15	atrib_16	atrib_17	atrib_18	atrib_19
0	-0.058049	-0.285629	0.0	-0.338097	-0.199868	-0.621918	-0.110554	-0.463761	-0.090389	-0.784836	-0.693751	-0.796105	-0.848481	1.511535	-0.712136	-0.070207	-0.849913	0.757935	-0.001032
1	0.598297	-0.153891	0.0	-0.338097	-0.199868	-0.598559	-0.103935	-0.509463	-0.093776	-0.769504	-0.659068	-0.793056	-0.837138	1.680258	-0.772656	-0.106443	-0.831181	0.724362	0.119387
2	-0.536634	-0.055087	0.0	-0.338097	-0.199868	-0.621918	-0.109110	-0.418060	-0.087325	-0.787194	-0.686044	-0.808304	-0.848481	1.607948	-0.772656	-0.052089	-0.856157	0.696035	0.094247
3	-1.507478	-0.088021	0.0	-0.338097	-0.199868	-0.528480	-0.107184	-0.341891	-0.088024	-0.795450	-0.678336	-0.829651	-0.856043	1.764620	-0.857382	-0.043030	-0.849913	0.891177	0.229748
4	-1.439109	-0.516170	0.0	-0.338097	-0.199868	-0.435042	-0.100444	-0.402826	-0.088669	-0.789553	-0.705312	-0.796105	-0.852262	1.451277	-0.687929	-0.061148	-0.846791	0.840188	-0.034012

SET DE ATRIBUTOS

4. PCA

4.1 Aplicación del PCA

A continuación aplicaremos el PCA con tantos componentes principales como especifique el usuario. Se calculan 2 por defecto para que a continuación podamos graficar nuestros datos.

```
nComponentes = int(input())
pca = PCA(n_components=nComponentes)
```

```
2
```

INPUT DE NO. DE COMPONENTES

	PC1	PC2
0	-2.299677	-0.343837
1	-2.371925	-0.403867
2	-2.389590	-0.266572
3	-2.514506	-0.096834
4	-2.236746	-0.124341

DATASET DESPUÉS DE APLICAR POCA CON 2 COMPONENTES PRINCIPALES

4.2 Análisis de resultados

A continuación calcularemos la razón de varianza para cada componente. Ésta relación es la varianza causada por cada componente princiapl. Esta nos sirve para observar que tan bien los componentes principales calculados representan a nuestros datos originales.

Estos son las razones para los 19 componentes principales.

```
[4.13814900e-01, 1.65640288e-01, 1.06345188e-01, 6.27965215e-02,
5.78674248e-02, 4.70498598e-02, 4.20205907e-02, 3.83644353e-02,
2.77241028e-02, 1.90903343e-02, 1.22596944e-02, 4.70502749e-03,
2.29038862e-03, 3.12439197e-05, 1.49188547e-16, 1.03612940e-16,
9.12790822e-17, 7.73403277e-17, 9.78771803e-34]
```

RAZONES PARA LOS 19 COMPONENTES

Podemos observar que el CP1 es responsable de 41.3% de la varianza. Similarmente, el CP2 causa el 16.5% de la varianza en nuestro set de datos. Por lotanto, podemos decir que

colectivamente, los dos primeros componentes principales capturan el 57.8% (41.3% + 16.5%) de la información de nuestro dataset, lo cual no es tan óptimo pero nos puede ser útil, por ejemplo, para graficar nuestros datos.

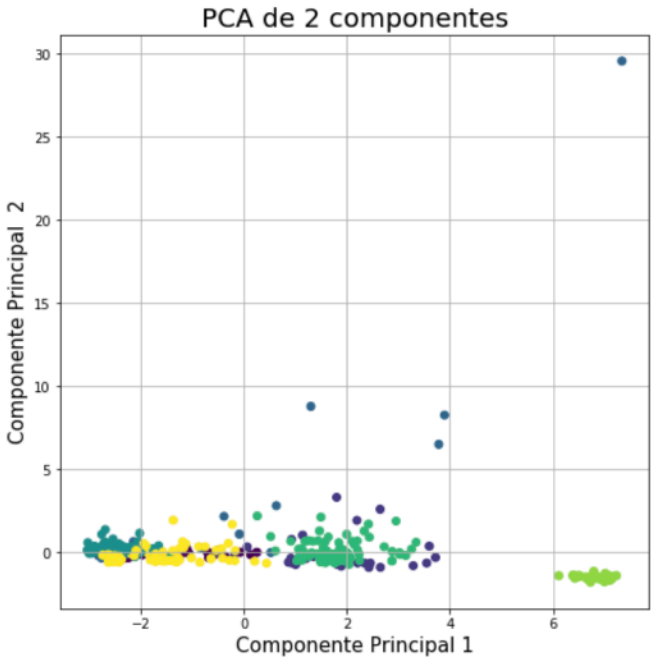
Como ejemplo, si quisieramos representar un 85% de nuestros datos, tendríamos que ocupar los 6 Componentes picipales primarios.

```
PC1: 0.414
PC2: 0.166
Total: 57.95%
```

EXPLAINED_VARIANCE COMPUESTA DEL PC1 Y PC2

4.3 Visualización del PCA 2D

Nuestro dataset original contenía 19 dimensiones las cuales, a través del PCA, logramos reducir a 2. Esto lo hicimos para poder graficar nuesta data y poder recuperar patrones



GRÁFICA DE DISPERSIÓN DE LOS DATOS EN DOS DIMENSIONES

5. Detección de Outliers

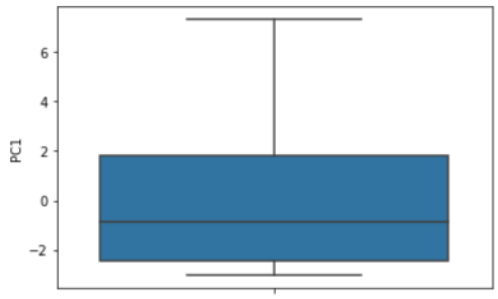
Al graficar observamos que hay elementos en nuestro dataset que se comportan de manera extraña y están muy alejados de los demás.

Se puede ver mucha varianza en el componente 2, con algunos elementos muy alejados de la media. Éstos son posibles elementos anómalos o outliers que, sin el dominio del dataset, no podríamos evaluar con certeza.

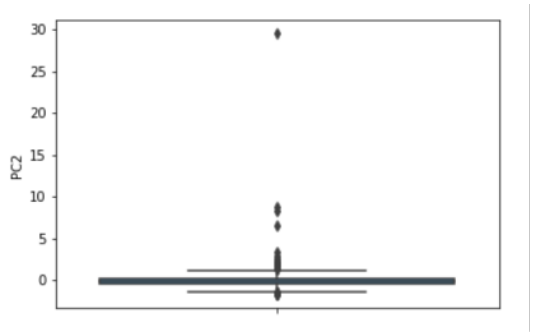
Primero, nos apoyaremos de la técnica de graficación por bigotes para detectar outliers en los dos componentes.

En el PC 1 no vemos elementos graficados fuera de los bigotes, lo que nos indica que no hay casos anómalos.

A diferencia del PC1, en el PC2 podemos observar que existen elementos muy alejados de la media y de los bordes superior e inferior (Q1 y Q3).



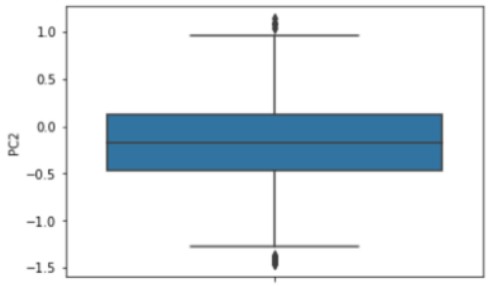
GRÁFICA DE BIGOTES PARA EL PC1



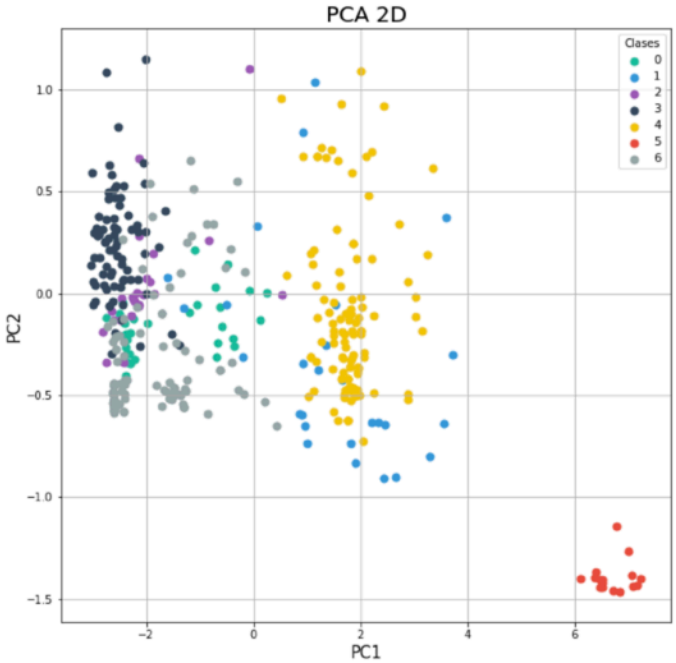
GRÁFICA DE BIGOTES PARA EL PC2

Para efectos de la experimentación, supondremos que sí son outliers y queremos recortarlos. Para esto nos apoyaremos del método del Rango Inter-Cuartil (IQR) que es el mismo que emplean las gráficas de bigotes para su graficación. Este método observa la dispersión estadística de nuestros datos. Con esto podremos detectar aquellos elementos que están muy alejados de la media y sobrepasan los límites superior e inferior, y eliminarlos.

Calculamos los límites inferior y superior. Una vez identificados los límites podemos observar aquellos elementos que los exceden. Una vez identificados nuestros outliers, procederemos a eliminarlos.



GRÁFICA DE BIGOTES PARA EL PC2 DESPUÉS DE RECORTAR OUTLIERS



GRÁFICA DE DISPERSIÓN DE LOS DATOS EN 2D DESPUÉS DE RECORTAR OUTLIERS

Una vez que limpiamos nuestros datos, procederemos a graficar de nuevo nuestros dos primeros componentes principales para observar de mejor manera como se dispersan nuestras clases.

6. Clasificación

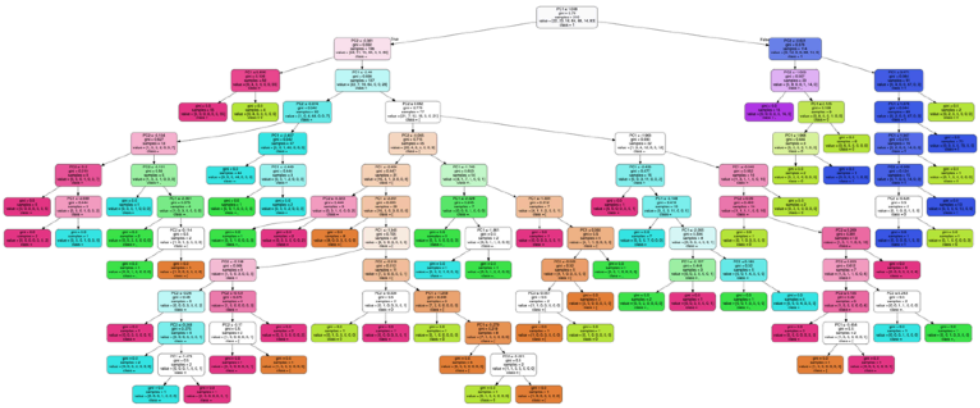
Tener una gran cantidad de atributos en un dataset afecta el rendimiento y la precisión de los algoritmos de clasificación. Nuestro dataset original contenía 19 atributos, los cuales, a través de la técnica de reducción de dimensionalidad de PCA, logramos reducir a 2.

En este ejemplo entrenaremos un árbol de decisión con nuestros datos reducidos con PCA. En seguida, analizaremos la precisión de éste cuando es entrenado con distintas cantidades de Componentes Principales. El objetivo es ver el número óptimo de Componentes Principales que nos permitan reducir el tiempo de entrenamiento del clasificador al resumir nuestros datos adecuadamente, y conservar un elevado porcentaje de precisión.

El método de clasificación por árbol de decisión es un método de aprendizaje supervisado, por esto, debemos entrenarlo con el set de atributos y su clasificación inicial. Además, para probar la precisión de éste, necesitamos un set de prueba. Por ello procederemos a partir nuestro set de datos en 2 secciones: Un set para entrenar a nuestro clasificador, y uno para entrenarlo.

6.1 Clasificar

Procederemos a entrenar nuestro clasificador con el set de datos de entrenamiento y sus respectivas clases. A continuación se muestra el árbol de decisión generado. El objetivo es generar un árbol no tan profundo para que la toma de decisiónes sea rápida.



ÁRBOL DE DECISIÓN GENERADO TRAS ENTRENAR CLASIFICADOR CON LOS 2 COMPONENTES PRINCIPALES

6.2 Predicción y Análisis

Por último predeciremos la clasificación de los datos de prueba que apartamos del set original antes de clasificar y compararemos, a través de una matriz de confusión, que tanto éstos se alejan de su verdadera clasificación.

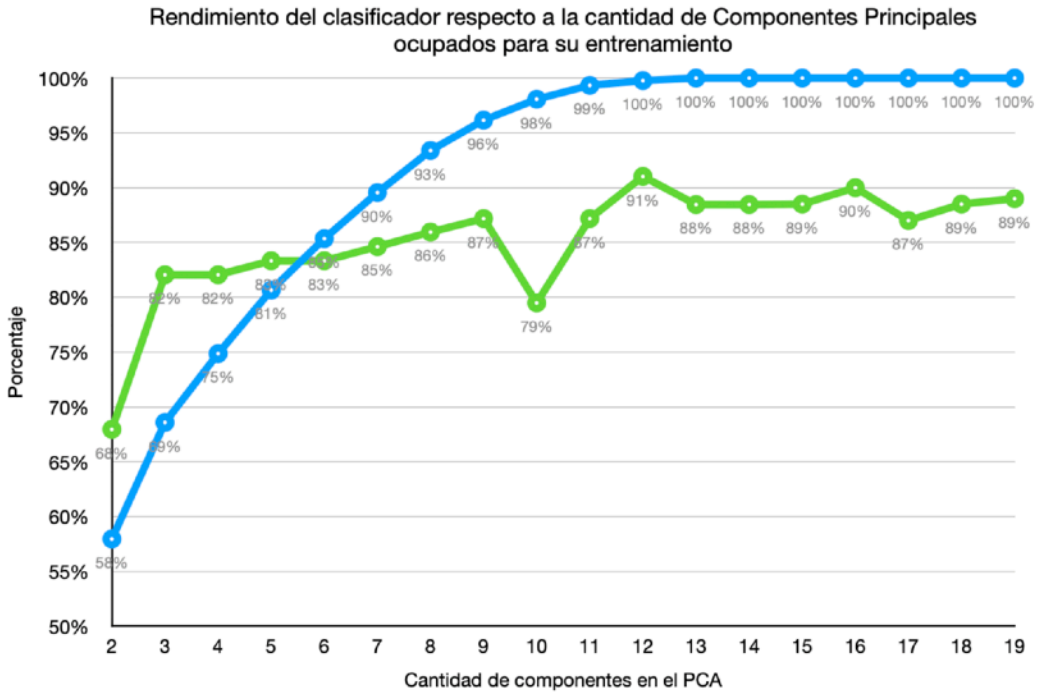
```
Número de Componentes Principales ocupados:
2
Explained_Variance:
57.95%

Matriz de confusión:
[[ 4  0  1  1  0  0  2]
 [ 0  1  0  0  3  0  0]
 [ 2  0  3  0  0  0  3]
 [ 0  0  0 11  0  0  2]
 [ 0  4  0  0 21  0  0]
 [ 0  0  0  0  0  2  0]
 [ 0  2  2  0  0  0 14]]

Precición de clasificación:
71.79%
```

MÉTRICAS Y MATRIZ DE CONFUSIÓN TRAS PROBAR NUESTRO CLASIFICADOR CON 2 COMPONENTES

Podemos obeservar que al ocupar 2 componentes principales, estamos representando el 57.95% de nuestros datos originales y la precisión de nuestro clasificador es de 70.51%, lo cual no es óptimo.

Para observar la variación en la precisión del clasificador dependiendo de la cantidad de componentes principales que ocupemos realizamos múltiples experimentos, entrenando el clasificador el resultado del PCA tras alterar la cantidad de componentes principales que deseabamos.

A continuación se presenta la gráfica que relaciona la cantidad de componentes principales ocupados para el entrenmamiento del clasificador y su respectiva calificación de precisión de clasificación.



Como se observa en la gráfica, a partir de los 11-12 componentees principales, se tiene una casi perfecta representación de los datos principales. A su vez, es en estos valores que el clasificador tiene su máximo en cuanto a grado de precisión. Esto quiere decir que reducir nuestra data a 12 dimensiones, en este ejmplo, sería óptimo para el clasificador.