ITESO

DEPARTAMENTO DE MATEMÁTICAS Y FÍSICA

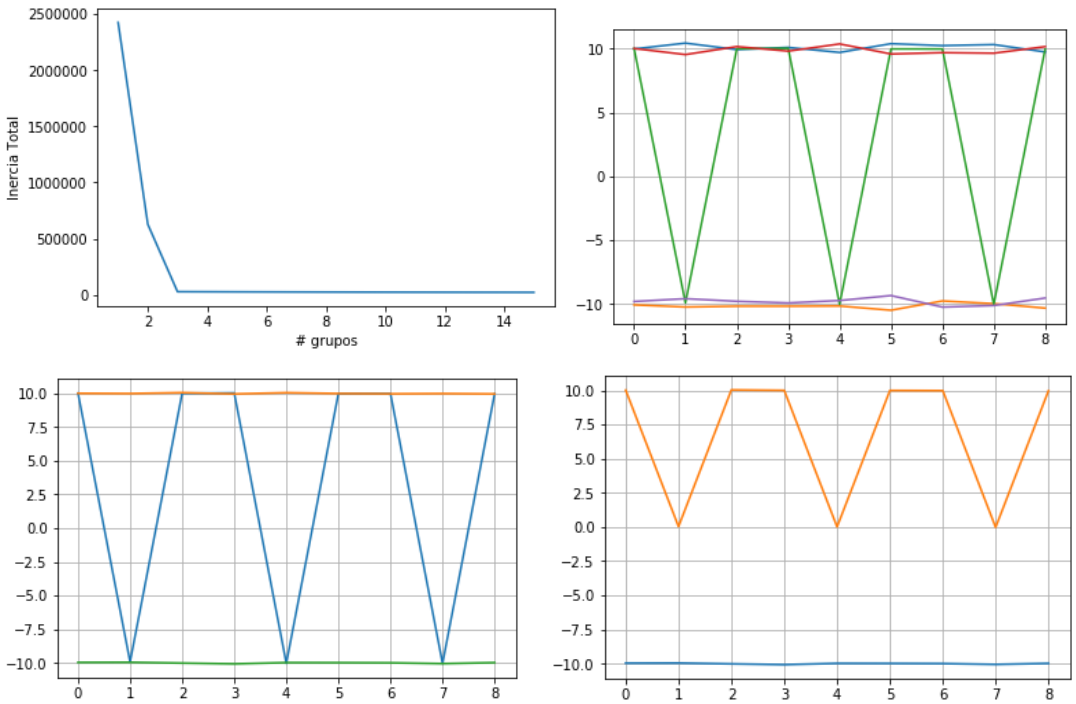
Asignatura: Ciencia de Datos e Inteligencia de Negocios

EXAMEN (Clustering y reducción de características)

Nombre: ___Oscar Eduardo Flores Hernández ___

Lea detenidamente los reactivos y responda con claridad. Si se requiere hacer uso de más hojas para la realización de cálculos, es necesario que se adjunten a este cuando se haga entrega del examen.

1. (3 puntos) En un experimento se logró identificar que 9 variables podían ser consideradas como importantes, las cuales determinaban el comportamiento de este. Después de hacer muchos experimentos se lograron recoger las muestras de diferentes condiciones de trabajo y se encuentran en el archivo ***"ex2c_1_2.csv"***. Determine cuantos grupos o patrones se encuentran en los datos recopilados y justifique su respuesta con código, figuras o mediciones.



En la primer imagen encontramos una gráfica de codos basados en el algoritmo de agrupamiento conocido como k-means, una de las características principales de este algoritmo es que únicamente funciona con datos cuantitativos. Los datos son cuantitativos, por lo tanto los agrupamos con método de k-means. Se puede notar la presencia de un codo cuando el número de grupos es 3.

Para comprobar que el número de grupos es 3 se grafican los centroides con 5 grupos (imagen 2), con 3 grupos (imagen 3) y con 2 grupos (imagen 4). Se puede notar que con 2 grupos se pierde información de clasificación y le da demasiada importancia a los picos (Sin darle propiamente un

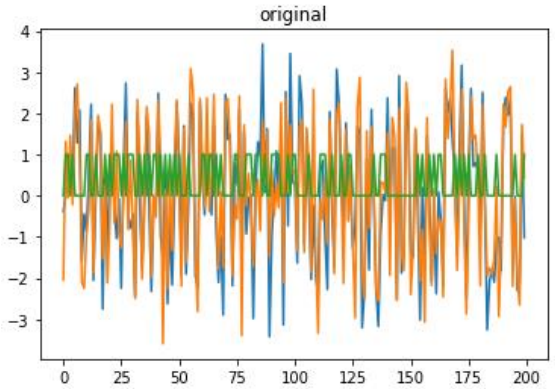
grupo a estos. Con 5 grupos ocurre lo contrario; se tienen 2 grupos 'sobreajustados' en la parte superior, 2 grupos 'sobreajustados' en la parte inferior y 1 grupo en 'zig-zag'. Con 3 grupos no se sobre-ajusta ninguno de los grupos, por lo tanto se considera como la mejor aproximación.

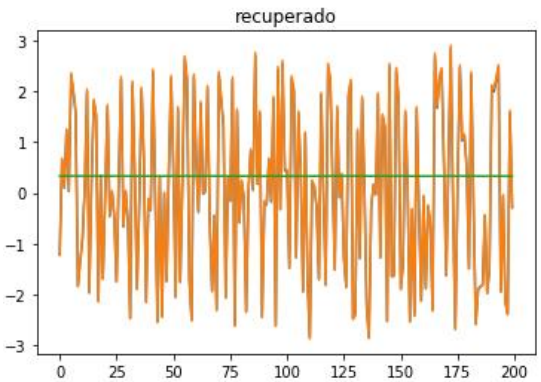
2. (2 puntos) En la base de datos **"ex2c_2_2.csv"** se cuenta una base de datos en 3 dimensiones. Explique y justifique si ¿es posible hacer una reducción de las variables por medio del "PCA" a solo una variable? Muestre en un gráfico, como serían los datos después de la reducción.

No es posible reducir a una sola variable debido a que los nuevos datos (tabla de la derecha) presentados no tienen similitud con los datos originales (tabla de la izquierda), principalmente en los datos binarios, estos datos originalmente variaban demasiado y en los datos recuperados de una proyección a una sola variable los datos que en teoría deberían ser binarios tienen únicamente un valor cercano a .33.

Index	x1	x2	x3
0	-0.382266	-2.04435	0
1	0.00611131	1.31898	1
2	0.279828	-0.0566355	1
3	1.01507	1.46598	0
4	0.307537	-0.21547	1
5	2.61765	2.06124	0
6	1.27805	2.70682	0
7	2.07773	1.15642	0
8	-1.58374	-2.09574	0
9	-0.443345	-2.24242	0
10	-0.877043	-0.929278	1
11	0.51781	-0.353819	1
12	2.21718	1.82437	0

	0	1	2
0	-1.22172	-1.21895	0.332515
1	0.651093	0.685126	0.33406
2	0.0979738	0.122773	0.333604
3	1.22031	1.26384	0.334529
4	0.032193	0.0558941	0.333549
5	2.30592	2.36757	0.335424
6	1.96996	2.026	0.335147
7	1.58808	1.63776	0.334832
8	-1.8382	-1.84572	0.332007
9	-1.35077	-1.35016	0.332409
10	-0.907147	-0.899127	0.332775
11	0.0664241	0.0906967	0.333578
12	1.99058	2.04697	0.335164



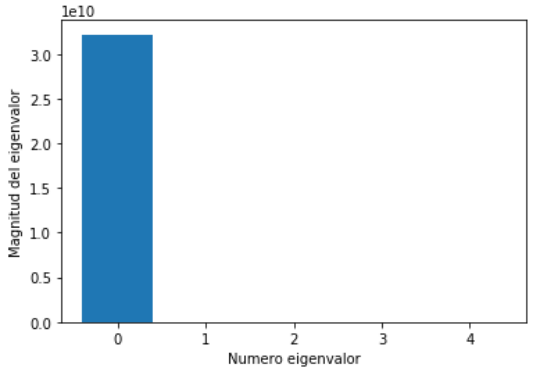


Los datos originales también presentan una diferencia notoria entre la variable en color naranja y la variable en color azul, no obstante, en los datos recuperados la primer variable en color naranja esta sobre la variable en color azul (esto significa que la recuperación de los datos considera que se mueven exactamente igual los datos de la columna 1 y los datos de la columna 2).

3. Considere la siguiente base de datos pequeña (**"*enfermedades.csv*"**):

Nombre	2014	2013	2012	2011	2010
Casos de Dengue	1446	2584	560	175	1171
Casos de Influenza A H1N1	608	14	592	6	108
Casos de VIH/SIDA	506	630	978	665	578
Egresos hospitalarios	221364	155789	180462	220280	199288
Esperanza de vida al nacer	75.36	75.36	75.89	77.28	77.07
Muertes maternas	52	33	35	36	48

Si aplicamos el análisis de componentes principales (PCA) a los datos de los años del 2010 al 2014 y obtenemos los siguientes eigenvalores, $w = [\mathbf{3.22056079e10} \quad \mathbf{702177} \quad \mathbf{49378.7} \quad \mathbf{28691.3} \quad \mathbf{3.63003}]$, los cuales pueden ser graficados como:



a. (1 punto) ¿Es correcto pensar que con solo los datos del año 2014 de todas categorías es suficiente para poder distinguirlas o se conserva más del 90% de información? Justifique su respuesta.

No, los eigenvalores nos muestran que tanto peso tienen los vectores de transformación de nuestros datos a una nueva dimensión. Lo que nos estaría diciendo esta gráfica de eigenvalores sería que es suficiente hacer una proyección a una línea o a una base de datos con una sola columna (que no tendría una interpretación directa), trabajar con ella y posteriormente regresar a las 5 variables que conocemos para darle una interpretación real. Todo esto sin perdida de información.

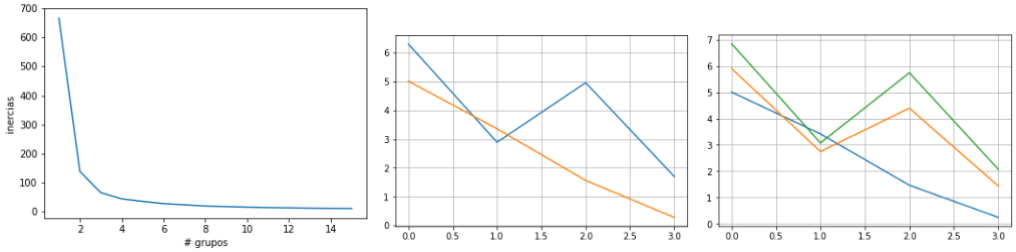
b. (1 punto) Normalmente el PCA se aplica a las columnas de una base de datos. ¿Qué interpretación se le puede dar a los resultados de aplicar el PCA a las filas de la base de datos?

Si se aplica el PCA a la base de datos de enfermedades por columnas estaríamos trasladando los datos de las columnas a nuevas dimensiones en las que compararíamos la importancia y

eliminaríamos (de ser posible) columnas basados en sus eigenvalores. El efecto para esta pequeña base de datos ejecutado sobre las filas en lugar de las columnas sería bastante similar, con la gran diferencia de comparar cada una de las enfermedades con otras en una nueva dimensión. El problema con esto es que no se tendría una interpretación con las proyecciones, sino que se tendría que recuperar previamente para poder hacer una interpretación.

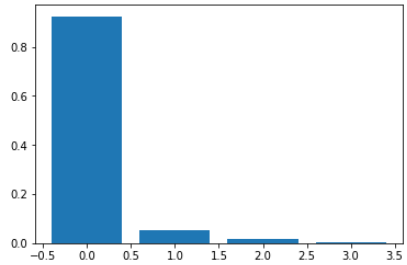
Cabe resaltar que para bases de datos más grandes puede resultar un problema potencial el hacerlo por filas, esto debido a que parte del proceso de PCA implica el calculo de una matriz de covarianzas. Si hay demasiadas filas, una matriz de covarianzas necesitaría demasiado poder computacional.

4. En el archivo llamado **"ex2c_4.csv"** se encuentra un set de datos con información sobre características de flores que fueron capturadas por un botánico. Basado en estos realicé lo siguiente:
    a. (1.5 puntos) Por medio de un algoritmo de clustering determine cuantos grupos se patrones se pueden determinar en este set de datos.

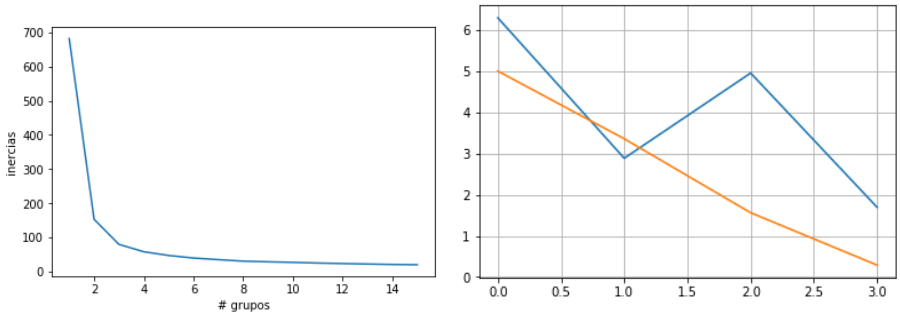


Para los datos originales no estandarizados se utiliza K-means como algoritmo de clustering y se obtiene la gráfica de codos antieror. Basados en la gráfica de codos se podrían haber tomado 2 decisiones de agrupación aparentemente válidas (2 y 3 grupos), no obstante, al ver los centroides de 2 y 3 grupos se puede observar que los centroides naranja y verde del lado derecho son muy similares al azul del lado izquierdo, por lo tanto escojemos tener 2 clusters únicamente.

    b. (1.5 puntos) Aplique el análisis de PCA en la base de datos original y con la base de datos reducida (manteniendo como mínimo 90% de la información), vuelva a determinar el número de patrones por medio de un algoritmo de clustering. ¿Se mantuvo el número de grupos encontrados en el inciso anterior? Explique y valide su respuesta con gráficas y mediciones.



Después de hacer un análisis de PCA encontramos que se puede hacer una transformación a 2 nuevas variables en un espacio 'desconocido' y aún así conservar el 99% de la información. Tomamos la transformación para las 2 variables transformadas y posteriormente recuperamos la información en una tabla que debería tener un comportamiento similar al original. Una vez se tiene la información recuperada repetimos el algorítmo de clustering, los resultados son los siguientes:



Se puede observer que en este caso el número de grupos sugerido es bastante similar al que se tenía originalmente, sin embargo, el trabajo computacional se redujo al utilizar la mitad de las columnas necesarias para ejecutar el algoritmo de clustering.

Codigo:

```python
#!/usr/bin/env python3
# -*- coding: utf-8 -*-
"""
Created on Wed Apr 10 15:23:09 2019

@author: Chelsi
"""

import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
from sklearn.cluster import KMeans

##############################################################################
############################      Ejercicio1      ############################
##############################################################################
#%% Importar datos
data = pd.read_csv('Archivos Examen 2-20190410/ex2c_1_2.csv',index_col=0)

#%% Grafica de codos
inercias = np.zeros(15)
for k in np.arange(len(inercias))+1:
    model = KMeans(n_clusters=k,init='random')
    model = model.fit(data)
    inercias[k-1]= model.inertia_

plt.plot(np.arange(len(inercias))+1,inercias)
plt.xlabel('# grupos')
plt.ylabel('Inercia Total')
plt.show()

#%%
model = KMeans(n_clusters=5,init='k-means++').fit(data)
grupos = model.predict(data)
centroides = model.cluster_centers_
plt.plot(centroides.T)
plt.grid()
plt.show()

#%%
model = KMeans(n_clusters=3,init='k-means++').fit(data)
grupos = model.predict(data)
centroides = model.cluster_centers_
plt.plot(centroides.T)
plt.grid()
plt.show()

#%%
model = KMeans(n_clusters=2,init='k-means++').fit(data)
grupos = model.predict(data)
centroides = model.cluster_centers_
plt.plot(centroides.T)
plt.grid()
plt.show()

#%% Clasificar los datos segun la grafica de codo
model = KMeans(n_clusters=3)
model = model.fit(data)
grupos = model.predict(data)

```

```
##############################################################################
############################      Ejercicio2      ############################
##############################################################################
#%% Importar datos
data = pd.read_csv('Archivos Examen 2-20190410/ex2c_2_2.csv',index_col=0)

#%% PCA
media = data.mean(axis=0)
data_m = data-media
M_cov = np.cov(data_m,rowvar=False)
w,v = np.linalg.eig(M_cov)

#%% Decidir numero de variables a reducir
porcentaje = w/np.sum(w)
porcentaje_acum = np.cumsum(porcentaje)

plt.figure()
plt.bar(np.arange(len(porcentaje)),porcentaje)
plt.show()

#%% Proyectar datos en nuevas dimensiones
limite = .91
indx = porcentaje_acum<=limite
componentes = w[indx]
M_trans = v[:,indx]

data_new = np.array(np.matrix(data_m)*np.matrix(M_trans))

#%% Recuperar imagenes de las variables reducidas
data_r = np.array(np.matrix(data_new)*np.matrix(M_trans.transpose()))
data_r = data_r + media.values

#%%
plt.plot(data[0:200])
plt.title('original')
plt.show()

plt.plot(data_r[0:200])
plt.title('recuperado')
plt.show()




##############################################################################
############################      Ejercicio4      ############################
##############################################################################
#%% Importar Datos
data = pd.read_csv('Archivos Examen 2-20190410/ex2c_4.csv',index_col=0)

#%% Encontrar número de clusters por gráfica de codos.
n_clusters = 15
inercias = np.zeros(n_clusters)
for k in np.arange(n_clusters)+1:
    model = KMeans(n_clusters=k,init='k-means++').fit(data)
    inercias[k-1] = model.inertia_

plt.plot(np.arange(n_clusters)+1,inercias+1) #plt.plot(x,y)
plt.xlabel('# grupos')
plt.ylabel('inercias')
plt.show()

```

```
#%% Clasificar datos segun el codo y graficar los centroides
model = KMeans(n_clusters=2,init='k-means++').fit(data)
grupos = model.predict(data)
centroides = model.cluster_centers_
plt.plot(centroides.T)
plt.grid()
plt.show()

#%% PCA en la base de datos
media = data.mean(axis=0)
data_m = data-media
M_cov = np.cov(data_m,rowvar=False)
w,v = np.linalg.eig(M_cov)

#%% Decidir numero de variables a reducir
porcentaje = w/np.sum(w)
porcentaje_acum = np.cumsum(porcentaje)

plt.figure()
plt.bar(np.arange(len(porcentaje)),porcentaje)
plt.show()

#%% Proyectar datos en nuevas dimensiones
limite = 1
indx = porcentaje_acum<=limite
componentes = w[indx]
M_trans = v[:,indx]

data_new = np.array(np.matrix(data_m)*np.matrix(M_trans))

#%% Recuperar imagenes de las variables reducidas
data_r = np.array(np.matrix(data_new)*np.matrix(M_trans.transpose()))
data_r = data_r + media.values

#%% Encontrar número de clusters por gráfica de codos para datos normalizados
n_clusters = 15
inercias = np.zeros(n_clusters)
for k in np.arange(n_clusters)+1:
    model = KMeans(n_clusters=k,init='k-means++').fit(data_r)
    inercias[k-1] = model.inertia_

plt.plot(np.arange(n_clusters)+1,inercias+1)
plt.xlabel('# grupos')
plt.ylabel('inercias')
plt.show()

#%% Clasificar datos segun el codo y graficar los centroides para datos normaliz
model = KMeans(n_clusters=2,init='k-means++').fit(data_r)
grupos = model.predict(data_r)
centroides = model.cluster_centers_
plt.plot(centroides.T)
plt.grid()
plt.show()
```