Mini-Proyecto - Informe

Semana: 7

Nombre del estudiante:

Daniel Isaac Juarez Funes - 12141153

Diego Andre Molina Valladares - 12141157

Felix Omar Dominguez - 12141043

Pamela Giselle Ramírez - 12141141

Serlio Alejandro Giron Paz - 12141146

Sede de estudio:

UNITEC TGU

Docente:

Ing. Claudia Cortes

Sección:

CCC405 – Lenguajes de Programación

Fecha de entrega:

Sábado 2 de Marzo 2024

Índice

INTRODUCCIÓN

El algoritmo de análisis de componentes principales es un que es de alta utilidad hoy en día. Esta es una herramienta estadística que nos ayuda en varios sectores de la vida, desde la computación a la biología, está presente en casi cualquier lugar. En cierta forma esta es un directo resultado la evolución de la tecnología y como debemos de seguir innovando con todos los aspectos. Entre más tecnológico se ha vuelto el mundo más data este ha tenido que procesar, con miles de millones de personas, generando data constantemente, es importante saber cómo gestionar y estudiar esta, pero algunas veces esto es demasiado. Por ello algoritmos como ACP, son muy importantes, porque reducen la cantidad de información a base de su relación con otros datos dentro de la misma sección de datos, sin que se pierda mucha información así teniendo una más manejable, pero ligeramente menos exacta selección de datos. Por ello conocer que es lo que hace y como es que lo hace es de suma importancia en el mundo de la informática.

DESCRIPCIÓN DEL PROBLEMA

Para este proyecto se tuvo que crear el análisis de componentes principales, dentro de dos lenguajes de programación, los cuales debían ser de tipos, para ver la diferencia de estos comparándolos con un mismo problema. El problema en sí, el análisis de componentes principales es una herramienta de la estadística que se usa en varias partes de la ciencia. Este consiste en tomar una colección de datos y comprimirla. Para hacer esto, los datos de esta colección deben de tener algún tipo de relación con otros datos de la colección, estos que este relacionados el uno con el otro actuaran de forma para que estos se puedan comprimir, dejando así solo los datos que no tiene nada que ver el uno con el otro. Al hacer estos va a ver una cierta perdida de información, pero esta va a ser mínima.

JUSTIFICACIÓN Y EXPLICACIÓN

C++: Utilizamos C++, primero porque cumplía con la característica de ser un programa débil y estático que contrasta el otro lenguaje elegido. Además, este es uno de los lenguajes con los que todos estamos familiarizados. Se nos enseñó este lenguaje en la clase de programación 3, y aunque este no se ha empleado tanto desde esa clase, ocasionalmente se ha utilizado en ciertos proyectos de clases de la carrera, así no perdemos practica con ello y por medio de ello, podríamos realizar un mejor trabajo usando este lenguaje.

Python: En directa oposición de C++ usamos Python, el cual es un lenguaje fuerte y dinámico, con lo cual cumplíamos con los requerimientos del proyecto. Este si no se nos fue enseñado a nosotros dentro de la carrera, pero siempre es uno de los mas populares y por ello es muy llamativo. Por ello siento que es muy importante conocerlo y tomar esta oportunidad para conocerlo y no estar atrás de la curva, en donde muchas personas ya utilizan Python.

DOCUMENTACIÓN

Python

Este código implementa el Análisis de Componentes Principales (ACP) en Python. Aquí está una explicación paso a paso de las funciones principales y su implementación:

Funciones Principales:

Cargar Archivo y Crear Matriz (cargarArchivo(filename)):

Lee un archivo CSV y devuelve una matriz que contiene los datos.

Centrar y Reducir (centrarReducir(matriz)):

Calcula las medias y desviaciones estándar de las columnas de la matriz.

Centra y reduce la matriz de datos.

Calculo de Matriz de Correlaciones (calcMatrizCorrelaciones(matrizCentradaReducida)):

Calcula la matriz de correlaciones entre las variables centradas y reducidas.

Calculo de Valores y Vectores Propios (calcValoresPropios(matrizCorrelaciones), calcVectoresPropios(matrizCorrelaciones)):

Calcula los valores y vectores propios de la matriz de correlaciones.

Ordenamiento de Valores y Vectores Propios (orderValoresPropios(valoresPropios), orderVectoresPropios(vectoresPropios, orderedValoresPropios)):

Ordena los valores y vectores propios en orden descendente.

Construcción de Matriz V (matrizV):

Crea la matriz de vectores propios ordenados.

Calculo de Matriz de Componentes Principales (calcMatrizComponentesPrincipales(matrizCentradaReducida, matrizV)):

Calcula la matriz de componentes principales utilizando la matriz centrada y reducida y la matriz de vectores propios.

Otros Cálculos y Visualizaciones:

Matriz de Calidades de Individuos, Coordenadas de Variables, Calidades de Variables, Vector de Inercias de los Ejes.

Gráficos de Círculo de Correlación y Plano Principal.

El código utiliza las bibliotecas numpy, matplotlib y csv para realizar los cálculos y visualizaciones necesarios. Los resultados se imprimen en la consola y se muestran gráficamente para una fácil interpretación.

C++

Este código en C++ implementa un análisis de componentes principales (PCA) para un conjunto de datos. Aquí tienes una explicación detallada de cómo se realiza cada paso junto con las bibliotecas utilizadas:

1. Bibliotecas Utilizadas:

iostream: Para entrada/salida estándar.

fstream: Para operaciones de archivo.

vector: Para manejar vectores.

string: Para manipulación de cadenas.

sstream: Para operaciones de cadena más avanzadas.

cmath: Para funciones matemáticas.

algorithm: Para algoritmos de ordenamiento y otras operaciones en contenedores STL.

eigen-3.4.0: Librería para álgebra lineal, en particular para la descomposición de valores propios y vectores propios.

2. Funciones Principales:

cargarArchivo: Lee un archivo CSV y devuelve los datos en forma de una matriz de doble vector.

calcMean: Calcula la media de cada columna de la matriz.

calcStdDev: Calcula la desviación estándar de cada columna de la matriz.

centrarReducir: Centra y reduce la matriz de datos.

calcCorrelaciones: Calcula la correlación entre dos columnas de datos.

calcMatrizCorrelaciones: Calcula la matriz de correlaciones entre todas las columnas de datos.

calcValoresPropios: Calcula los valores propios de la matriz de correlaciones.

calcVectoresPropios: Calcula los vectores propios correspondientes a los valores propios de la matriz de correlaciones.

orderValoresPropios y orderVectoresPropios: Ordena los valores y vectores propios en orden descendente.

calcMatrizComponentesPrincipales: Calcula la matriz de componentes principales utilizando los vectores propios ordenados.

calcMatrizCalidadesIndividuos: Calcula la matriz de calidades de los individuos.

calcMatrizCoordenadasVariables: Calcula la matriz de coordenadas de las variables.

calcMatrizCalidadesVariables: Calcula la matriz de calidades de las variables.

calcVectorInerciasEjes: Calcula el vector de inercias de los ejes.

3. Funciones Auxiliares:

imprimirVector e imprimirMatriz: Funciones para imprimir vectores y matrices.

imprimirParesValores: Función para imprimir pares de valores propios y vectores propios.

4. Función main:

Carga los datos desde un archivo CSV.

Realiza el análisis de componentes principales.

Imprime los resultados de cada paso del análisis.

Este código proporciona una implementación completa de PCA en C++ utilizando la biblioteca Eigen para operaciones de álgebra lineal. Es importante tener en cuenta que se espera que los datos estén en formato CSV y que las operaciones se realizan en memoria, lo que puede no ser eficiente para conjuntos de datos muy grandes.

Para Graficos (Hechos en Python)

1. Bibliotecas Utilizadas:

    csv: Para leer archivos CSV.

    numpy: Para operaciones numéricas.

    matplotlib.pyplot: Para visualización de datos.

2. Funciones Definidas:

fileHeadings(filename): Lee los encabezados de las columnas del archivo CSV.

fileNames(filename): Extrae los nombres de los individuos del archivo CSV.

3. Variables:

OrdenadosVectoresPropios: Matriz de vectores propios ordenados.

matrizComponentesPrincipales: Matriz de componentes principales.

4. Gráfico del Círculo de Correlación:

Se crea una figura con un círculo y se establece un sistema de coordenadas.

Se trazan flechas desde el origen hasta las puntas de los vectores propios.

Se etiquetan las flechas con los nombres de las variables.

Se muestra el gráfico.

5. Gráfico del Plano Principal:

Se toman las coordenadas de los componentes principales.

Se crea un gráfico de dispersión de estos componentes.

Se etiquetan los puntos con los nombres de los individuos.

Se muestra el gráfico.

Ambos gráficos son esenciales para visualizar y comprender los resultados del análisis de componentes principales. El primero muestra cómo las variables originales contribuyen a las nuevas dimensiones (componentes principales), mientras que el segundo muestra cómo los individuos están representados en estas nuevas dimensiones. Esto proporciona una comprensión visual de la estructura de los datos en el espacio de las nuevas variables.

EJEMPLOS Y DEMOSTRACIONES

Pyhton

```
Paso 1 - Matriz Centrada y Reducida:
[[ 0.23263076 -0.7529862   1.78848525  0.65792263  0.65858084]
 [ 0.78651352  1.14584856 -0.53899555 -0.84590053 -0.47690337]
 [ 0.89729007  1.01489444  0.31849737  0.09398895  0.09083874]
 [-1.98290027 -0.7529862  -1.51898747 -0.84590053  1.79406505]
 [-0.87513476 -1.0803715   0.07349939  0.93988948 -0.13625811]
 [ 1.11884317  1.27680268 -0.0489996   0.09398895 -1.04464547]
 [-0.5428051  -0.81846326  0.56349535  1.03387842 -0.24980653]
 [ 1.22961972  1.34227974 -0.29399757  0.09398895 -1.61238758]
 [-0.87513476 -1.0803715  -1.51898747 -2.25573474  1.45341979]
 [ 0.01107766 -0.29464677  1.1759903   1.03387842 -0.47690337]]

Paso 2 - Matriz de Correlaciones:
[[ 1.          0.85407878  0.38457424  0.20719425 -0.78716269]
 [ 0.85407878  1.         -0.02005218 -0.02153942 -0.68772056]
 [ 0.38457424 -0.02005218  1.          0.82091619 -0.36554342]
 [ 0.20719425 -0.02153942  0.82091619  1.         -0.50800132]
 [-0.78716269 -0.68772056 -0.36554342 -0.50800132  1.        ]]
```

```
Paso 3 - Valores y Vectores Propios:
Valor propio: 2.893249673417941
Vector propio: [-0.52664397 -0.42493622 -0.35914704 -0.35269747  0.53730181]

Valor propio: 1.628650424977314
Vector propio: [-0.2704963  -0.50807221  0.56208159  0.58648985  0.09374599]

Valor propio: 0.3465960485145297
Vector propio: [-0.43820071 -0.04049491 -0.56227583  0.39418032 -0.57862603]

Valor propio: 0.12261245959725252
Vector propio: [-0.62387762  0.32538951  0.48374732 -0.42043348 -0.30679407]

Valor propio: 0.008891393492955172
Vector propio: [-0.26121779  0.67362724 -0.07008647  0.44664495  0.52305619]

Paso 4 - Matriz V:
[[-0.52664397 -0.2704963  -0.43820071 -0.62387762 -0.26121779]
 [-0.42493622 -0.50807221 -0.04049491  0.32538951  0.67362724]
 [-0.35914704  0.56208159 -0.56227583  0.48374732 -0.07008647]
 [-0.35269747  0.58648985  0.39418032 -0.42043348  0.44664495]
 [ 0.53730181  0.09374599 -0.57862603 -0.30679407  0.52305619]]
```

```
Paso 5 - Matriz de Componentes Principales:
[[-0.32306263  1.7725245 ]
 [-0.66544057 -1.63870215]
 [-1.00254705 -0.51569247]
 [ 3.17209481 -0.26278201]
 [ 0.48886797  1.3654021 ]
 [-1.70863322 -1.02170044]
 [-0.06758577  1.46233642]
 [-2.01185516 -1.27586457]
 [ 3.04203029 -1.25488069]
 [-0.92386867  1.3693593 ]]

Paso 6 - Matriz de Calidades de Individuos:
[[1.68038276e-01 5.05847111e+00]
 [2.29249053e-01 1.39023766e+00]
 [5.47696895e-01 1.44914663e-01]
 [2.23659704e+00 1.53492327e-02]
 [1.23633757e-01 9.64439593e-01]
 [1.01297430e+00 3.62199538e-01]
 [4.73586747e-03 2.21709136e+00]
 [1.22147031e+00 4.91245566e-01]
 [4.78719334e+00 8.14626950e-01]
 [9.81756240e+00 2.15684044e+01]]
```

```
Paso 7 - Matriz de Coordenadas de las Variables:
[[-1.28504534  0.8690127  -1.24029964  0.19885774 -0.8378326 ]
 [ 0.16434188 -1.49106758  0.28887291  0.32480421  0.85196071]
 [-0.96900967 -0.81805377  0.10821138  0.40535531  0.41164551]
 [ 1.97268053  2.21997271  0.60806121  0.81594519  0.94083443]
 [ 0.17013049  1.13184984  0.12993706 -0.75201709 -0.97364381]
 [-0.69888837 -1.79527935  0.46207178 -0.17119775  0.17396051]
 [-0.31942744  0.791814   -0.06429212 -0.61270756 -1.14228151]
 [-0.51927685 -2.24814478  0.63663764 -0.52201925  0.08442035]
 [ 2.44638563  1.22736521 -0.63193565  0.67382496  1.75969995]
 [-0.96189086  0.11253101 -0.29726456 -0.36084576 -1.26876353]]

Paso 8 - Matriz de Calidades de las Variables:
[0.28932497 0.16286504 0.0346596  0.01226125 0.00088914]

Paso 9 - Vector de Inercias de los Ejes:
[57.86499347 32.5730085   6.93192097  2.45224919  0.17782787]
```

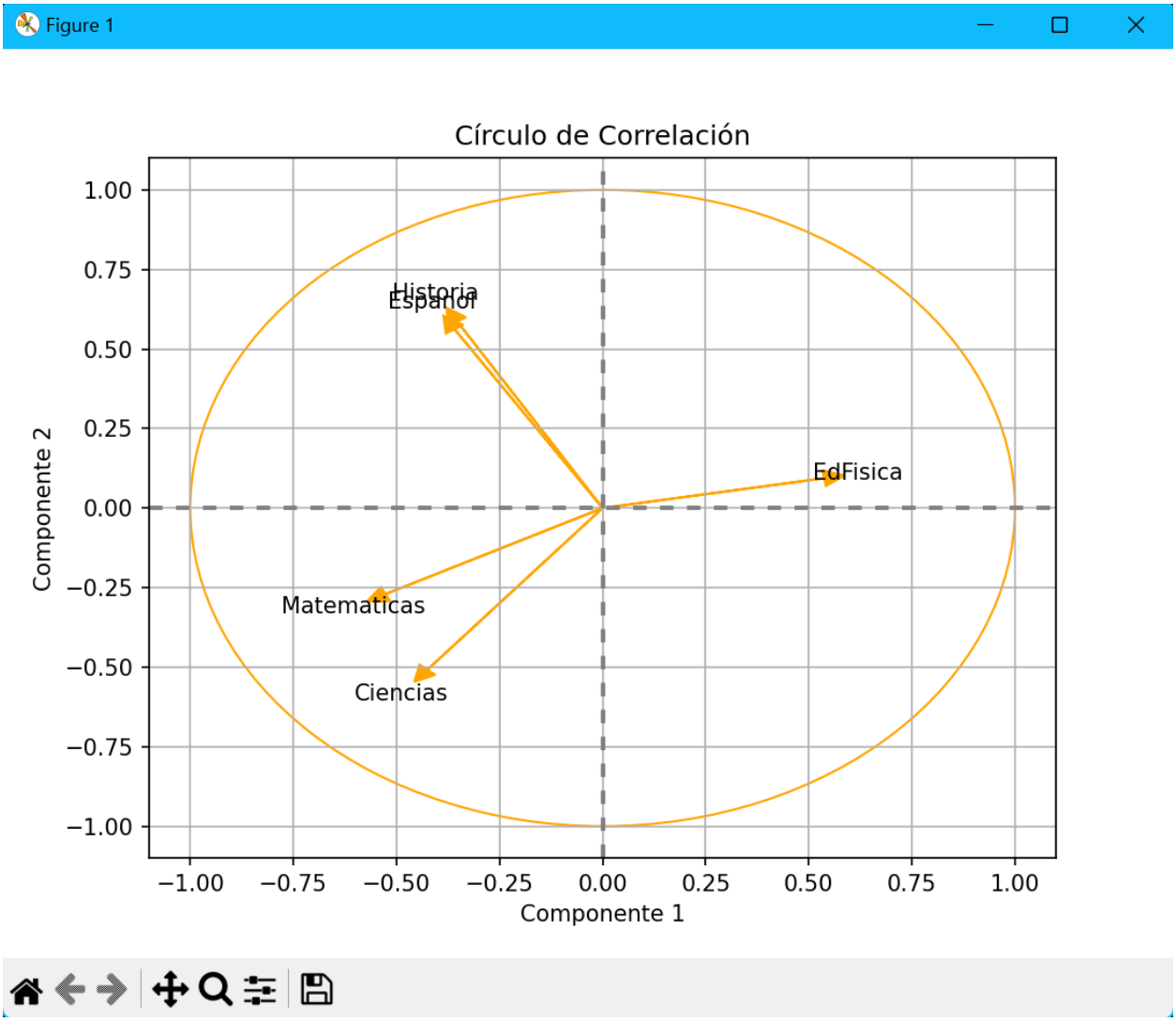


Figure 1
Círculo de Correlación
Historia
Español
EdFisica
Matematicas
Ciencias
Componente 2
Componente 1
1.00
0.75
0.50
0.25
0.00
−0.25
−0.50
−0.75
−1.00

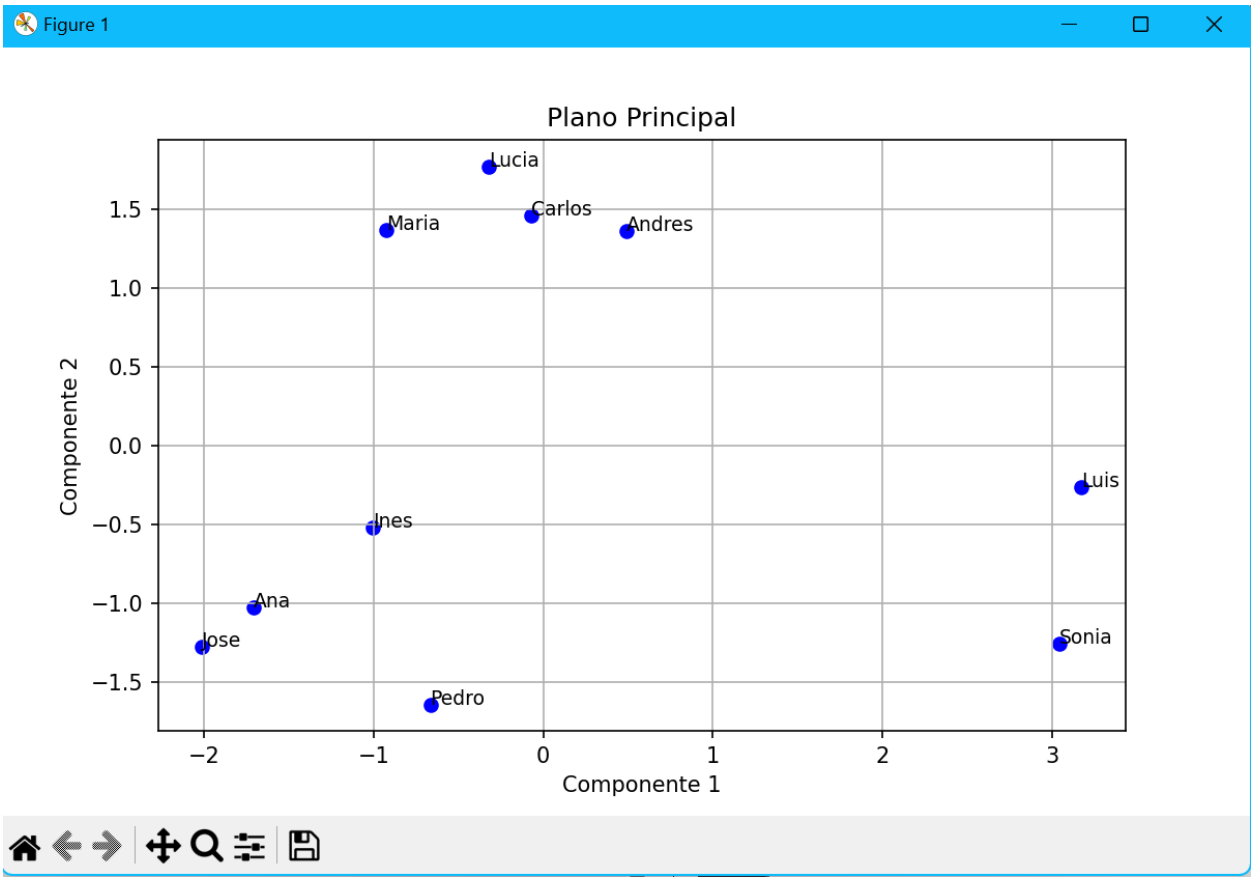


Figure 1
Plano Principal
Lucia
Carlos
Maria
Andres
Luis
Ines
Ana
Jose
Sonia
Pedro
Componente 2
Componente 1
1.5
1.0
0.5
0.0
−0.5
−1.0
−1.5
−2
−1
0
1
2
3

C++

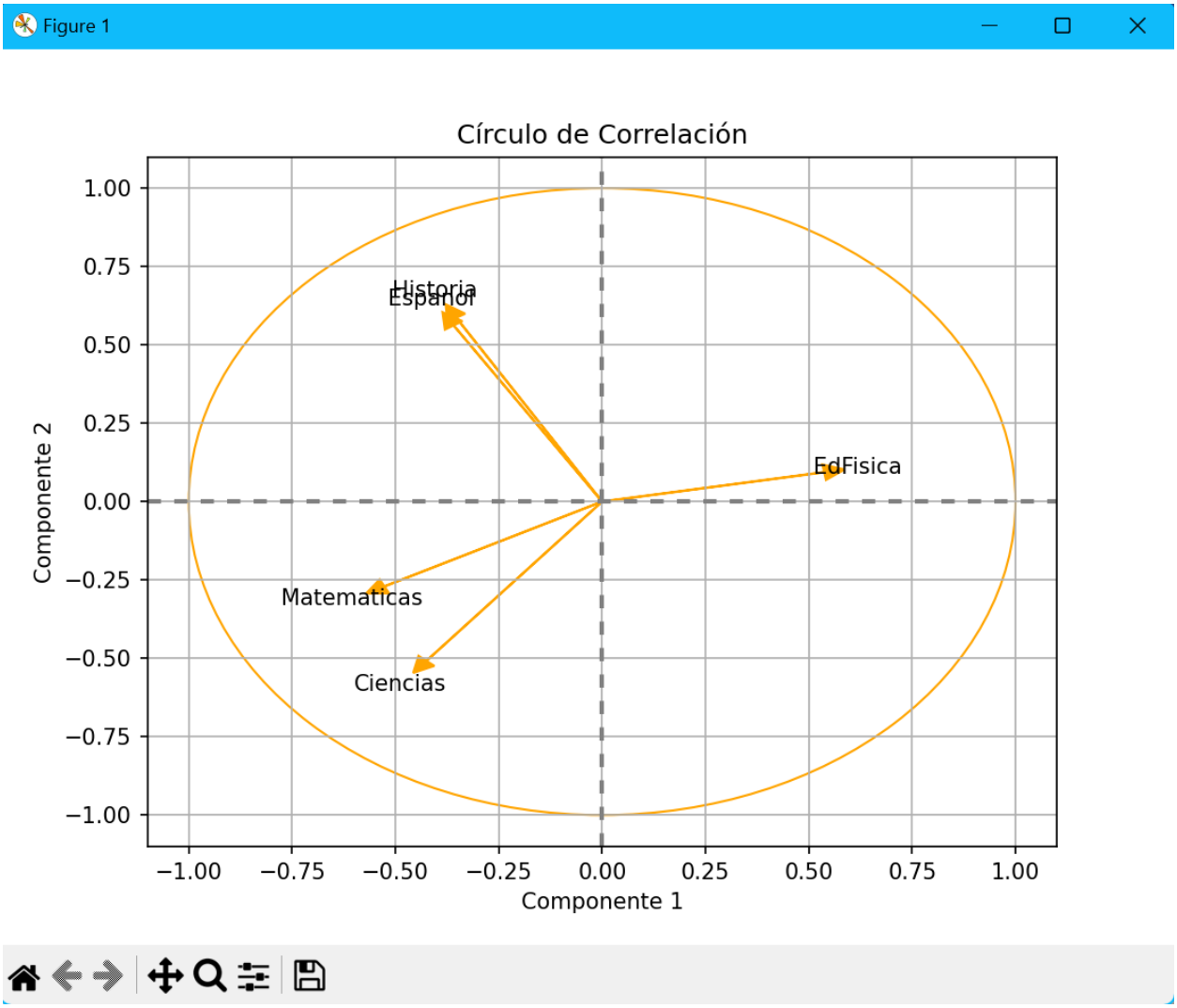
```
PS C:\Users\diego\OneDrive - Universidad Tecnologica Centroamericana\Trabajos Diego Unitec\CLASES Q1 2024\LENGUAJES DE PROGRAMACION\
Miniproyecto Compartido\MiniProyectoLenguajes> cd "c:\Users\diego\OneDrive - Universidad Tecnologica Centroamericana\Trabajos Diego
Unitec\CLASES Q1 2024\LENGUAJES DE PROGRAMACION\Miniproyecto Compartido\MiniProyectoLenguajes\cmasmas\" ; if ($?) { g++ Miniproyecto
.cpp -o Miniproyecto } ; if ($?) { .\Miniproyecto }

Paso 1 - Matriz Centrada y Reducida:
[[0.232631] [-0.752986] [1.78849] [0.657923] [0.658581]
[0.786514] [1.14585] [-0.538996] [-0.845901] [-0.476903]
[0.89729] [1.01489] [0.318497] [0.0939889] [0.0908387]
[-1.9829] [-0.752986] [-1.51899] [-0.845901] [1.79407]
[-0.875135] [-1.08037] [0.0734994] [0.939889] [-0.136258]
[1.11884] [1.2768] [-0.0489996] [0.0939889] [-1.04465]
[-0.542805] [-0.818463] [0.563495] [1.03388] [-0.249807]
[1.22962] [1.34228] [-0.293998] [0.0939889] [-1.61239]
[-0.875135] [-1.08037] [-1.51899] [-2.25573] [1.45342]
[0.0110777] [-0.294647] [1.17599] [1.03388] [-0.476903]
]

Paso 2 - Matriz de Correlaciones:
[[1] [0.854079] [0.384574] [0.207194] [-0.787163]
[0.854079] [1] [-0.0200522] [-0.0215394] [-0.687721]
[0.384574] [-0.0200522] [1] [0.820916] [-0.365543]
[0.207194] [-0.0215394] [0.820916] [1] [-0.508001]
[-0.787163] [-0.687721] [-0.365543] [-0.508001] [1]
]
```

```
Paso 3 - Valores y Vectores Propios:
Valor propio: 2.89325
Vector propio: 0.526644 0.424936 0.359147 0.352697 -0.537302

Valor propio: 1.62865
Vector propio: -0.270496 -0.508072 0.562082 0.58649 0.093746

Valor propio: 0.346596
Vector propio: 0.623878 -0.32539 -0.483747 0.420433 0.306794

Valor propio: 0.122612
Vector propio: 0.438201 0.0404949 0.562276 -0.39418 0.578626

Valor propio: 0.00889139
Vector propio: -0.261218 0.673627 -0.0700865 0.446645 0.523056

Matriz de Vectores Propios:
[[0.526644] [-0.270496] [0.623878] [0.438201] [-0.261218]
[0.424936] [-0.508072] [-0.32539] [0.0404949] [0.673627]
[0.359147] [0.562082] [-0.483747] [0.562276] [-0.0700865]
[0.352697] [0.58649] [0.420433] [-0.39418] [0.446645]
[-0.537302] [0.093746] [0.306794] [0.578626] [0.523056]
]
Matriz de Valores Propios:
[[2.89325] [1.62865] [0.346596] [0.122612] [0.00889139] ]
```

```
Paso 4 - Matriz V:
[[0.526644] [-0.270496] [0.623878] [0.438201] [-0.261218]
[0.424936] [-0.508072] [-0.32539] [0.0404949] [0.673627]
[0.359147] [0.562082] [-0.483747] [0.562276] [-0.0700865]
[0.352697] [0.58649] [0.420433] [-0.39418] [0.446645]
[-0.537302] [0.093746] [0.306794] [0.578626] [0.523056]
]

Paso 5 - Matriz de Componentes Principales:
[[0.323063] [1.77252]
[0.665441] [-1.6387]
[1.00255] [-0.515692]
[-3.17209] [-0.262782]
[-0.488868] [1.3654]
[1.70863] [-1.0217]
[0.0675858] [1.46234]
[2.01186] [-1.27586]
[-3.04203] [-1.25488]
[0.923869] [1.36936]
]
```

```
Paso 6 - Matriz de Calidades de Individuos:
[[0.168038] [5.05847]
[0.229249] [1.39024]
[0.547697] [0.144915]
[2.2366] [0.0153492]
[0.123634] [0.96444]
[1.01297] [0.3622]
[0.00473587] [2.21709]
[1.22147] [0.491246]
[4.78719] [0.814627]
[9.81756] [21.5684]
]

Paso 7 - Matriz de Coordenadas de las Variables:
[[1.55826] [0.369751] [-0.88109] [0.42718] [1.07828]
[-0.478101] [-0.428082] [0.745067] [0.843249] [-1.21944]
[0.414188] [-0.172987] [0.785121] [1.04913] [-0.187362]
[-2.62758] [1.2085] [-1.00196] [-0.644868] [0.977747]
[0.32466] [0.0993879] [-0.419087] [-1.34273] [0.864054]
[0.527358] [-0.857223] [1.26926] [0.61921] [-0.988518]
[0.805379] [-0.124584] [-0.328746] [-0.953663] [0.855366]
[0.563442] [-1.14614] [1.50416] [0.340097] [-1.41403]
[-2.48443] [1.55901] [-1.55696] [-0.0425891] [-0.642092]
[1.39683] [-0.507634] [-0.11577] [-0.295016] [0.675995]
]
```

```
Paso 8 - Matriz de Calidades de las Variables:
[[0.289325] [0.162865] [0.0346596] [0.0122612] [0.000889139] ]


Paso 9 - Vector de Inercias de los Ejes:
[[57.865] [32.573] [6.93192] [2.45225] [0.177828] ]
```



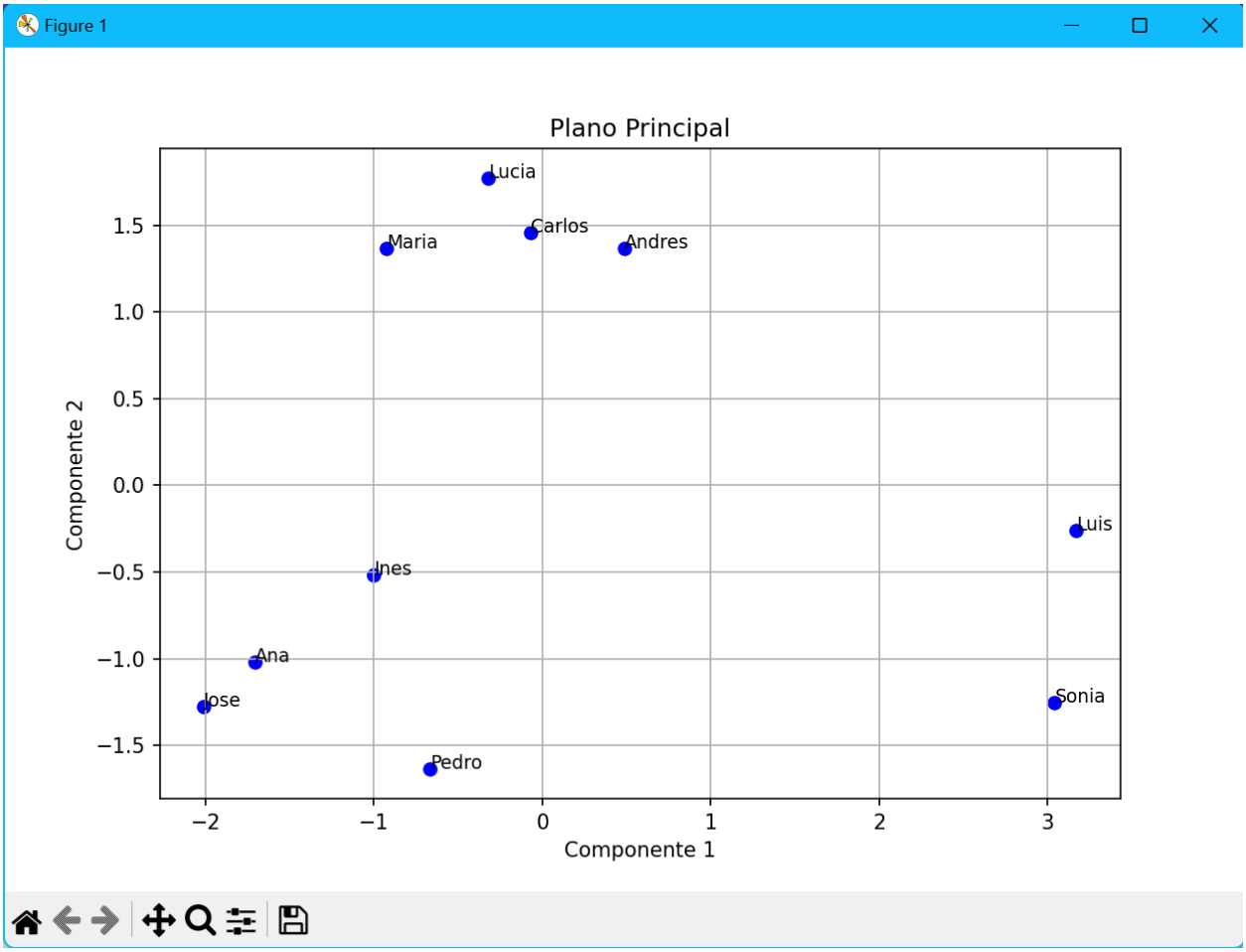


Figure 1
Plano Principal
Componente 1
Componente 2
Lucia
Carlos
Maria
Andres
Luis
Ines
Ana
Jose
Sonia
Pedro
−2
−1
0
1
2
3
1.5
1.0
0.5
0.0
−0.5
−1.0
−1.5

ANALISIS

A. **¿Considera que el lenguaje influyó al construir la solución?**

Si, debido a que si pienso que se pueden realizar ciertos problemas con mayor facilidad en ciertos lenguaje que en otros. En este casos logramos terminar el codigo en python antes que el de c++, por ciertos cosas que tenia el lenguaje en si, que lo hacian mas dificiles de trabajar con.

B. **¿Experimento la diferencia de generar una solución para un mismo problema en lenguajes de paradigmas diferentes? ¿Cuáles?**

Si, ya que los lenguaje pueden tener formas diferentes de trabajar con varibales, tipos, funciones, talvez uno tenga ciertas cosas ya adentro del lenguaje que hagan la solucion mucho mas sencilla mientras que otras no lo tengan. Esto hace que una traduccion directa de un lenguaje a otro sea imposible de hacer, ya que siempre se tienen que hacer cambios cruciales.

C. **¿Cuál sería el mayor aprendizaje que obtuvo de este proyecto?**

Aprender a saber usar las cosas que se nos proveen con cada lenguaje al maximo, porque talvez al quedarnos pensado en como se haria cierto problema en un lenguaje de nuestra preferencia, podriamos estarnos complicando mas y perdiendo el tiempo, a pesar de que eso es lo que intentabamos evitar desde el principio.

D. **Cualquier otro comentario u observación relevante**

Nada mas.

DIFICULTADES ENCONTRADAS

- Trabajar con pyhton al cual no todos estamos acostumbrados
- Entendimiento del algoritmo a realizar
- Errores especificos de C++.

CONCLUSIONES

- Al realizar un mismo problema con dos lenguajes de diferentes tipos uno logra apreciar más las diferencias entre cada uno. En nuestro caso con la comparación entre c++ y Python es una muy interesante, ya que muchas personas no suponen que un lenguaje como c++ seria uno que sea estático y débilmente tipado, debido a como se trabaja con este. Además de otra forma es raro ver como se categoriza a Python como uno dinámico, pero fuertemente tipado debido al poco énfasis que uno ve en los tipos de datos, por lo menos al comienzo. Pero al realizar problemas como estos las diferencias se hacen más notable y por medio de ello se logran apreciar más los lenguajes por sus diferentes cualidades.

- Por medio de un proyecto como este el análisis de componentes principales se va entendiendo como algo mucho más importante para el área de la informática y también para el manejo de data a gran escala. Es importante conocer lo gran y difícil de manejar que se ha vuelto la data con los nuevos servicios en línea que hay hoy en día y con la cantidad de personas que hay generando data cada segundo y por ello este tipo de algoritmos cuyo propósito es compactar la data mientras se intenta perder la menor cantidad de data original posible, hace el análisis de datos mucho más manejable. Es gracias a cosas como estos que muchas cosas que tenemos hoy en día funcionan como funcionan.

BIBLIOGRAFIAS Y REFERENCIAS

disaac21/MiniProyectoLenguajes (github.com)

REFERENCIAS

Rodríguez, O. (29 de mayo de 2008). *Análisis en Componentes Principales*. Obtenido de StudyLib: https://studylib.es/doc/5124728/cap%C3%ADtulo-2---oldemar-rodr%C3%ADguez-rojas

Zakaria, J. (1 de Abril de 2021). *A Step-by-Step Explanation of Principal Component Analysis (PCA)*. Obtenido de BuiltIn: https://builtin.com/data-science/step-step-explanation-principal-component-analysis