Universidad Tecnológica Centroamericana
Facultad de Ingeniería

CC405 – Lenguajes de Programación
Docente: Claudia Cortés.

Mini-Proyecto: Análisis de Componentes Principales.

El objetivo de este mini proyecto es comprender y poner en práctica las diferencias sintácticas y semánticas de lenguajes de programación que pertenecen a diferentes categorías.

Contexto

El Análisis en Componentes Principales (ACP), también conocido como
PCA por sus siglas en inglés (Principal Component Analysis), es una técnica estadística utilizada para reducir la dimensionalidad de un conjunto de datos. Se utiliza para identificar
patrones y relaciones en los datos y representarlos en un espacio de menor dimensión, mientras
se mantiene la mayor parte de la información original.

El propósito central del Análisis de Componentes Principales (ACP) radica en descubrir un conjunto de variables latentes (ocultas) denominadas **componentes principales**, las cuales no están correlacionadas entre sí y son formadas a través de combinaciones lineales de las variables originales. Estas componentes principales son dispuestas según la cantidad de variabilidad que explican en los datos originales, de manera que las primeras componentes principales capturan la mayor cantidad de información disponible.
De manera generalizada, el algoritmo ACP consta de los siguientes pasos:

1. Proceso de normalización/estandarización de datos: Se lleva a cabo una estandarización de las variables para garantizar que todas tengan la misma escala y evitar que las variables con mayor variabilidad ejerzan una influencia dominante en el análisis.
2. Cálculo de la matriz de varianzas-covarianzas o matriz de correlación: Se procede a calcular la matriz de varianzas-covarianzas. Cuando las variables están centradas y estandarizadas, las entradas de esta matriz representan las correlaciones entre las variables originales.
3. Cálculo de las componentes principales: Las componentes principales se determinan a partir de la matriz de correlación. Estas componentes son obtenidas mediante combinaciones lineales de las variables originales y se seleccionan de manera que capturen la máxima variabilidad presente en los datos.
4. Elección de las componentes principales: Se decide cuántas componentes principales serán utilizadas para representar los datos. Esta selección puede basarse en la

inspección de los valores propios o en la cantidad de varianza explicada por cada componente.

5. Transformación/proyección de datos al nuevo espacio: Como paso final, los datos originales son proyectados en el espacio definido por las componentes principales seleccionadas. Esto implica la multiplicación de la matriz de datos por la matriz de vectores propios de la matriz de correlaciones. Mediante esta proyección, los datos se representan en un espacio de menor dimensión.

El algoritmo ACP puede ser útil para identificar las variables más importantes en un conjunto de datos y para eliminar la multicolinealidad en el análisis de regresión.

Problema

Su grupo de trabajo deberá realizar dos implementaciones del algoritmo ACP en dos lenguajes de diferentes paradigmas/ tipos. Ejemplo: un lenguaje fuertemente tipado con definición estática de tipos vs un lenguaje débilmente tipado con revisión dinámica de tipos.
Sus programas deberán seguir una estructura como la descrita a continuación:

- **Entradas de datos:** Tabla de datos o matriz $X \in M_{nxm}$
- **Salida de satos:** Matriz de componentes principales $C \in M_{nxm}$, Matriz de calidades de los individuos (matriz de cosenos cuadrados) $Q \in M_{nxm}$,Matriz de coordenadas de variables ($T \in M_{mxm}$), Vector de inercias de los Ejes $I \in M_{1xm}$ y Matriz de calidades de variables (matriz de cosenos cuadrados) $S \in M_{mxm}$.
- **Gráficos para mostrar:** Plano principal y circulo de correlación.

Pasos de aplicación:

1. Centrar y reducir la tabla original de datos X.
2. Realizar el cálculo de la matriz de correlaciones $R \in M_{mxm}$
3. Calcular y ordenar (de mayor a menor) los vectores y valores propios de la Matriz $R \in M_{mxm}$
4. Considerando que los valores propios ordenados se representan como $\lambda_1, \lambda_2, \ldots, \lambda_m$ y los vectores propios serian $\mathbf{u_1, u_2, \ldots, u_m}$, , se deberá construir una matriz $V \in M_{mxm}$ de la siguiente manera:
$$V = [\mathbf{u_1}|\mathbf{u_2}|\cdots|\mathbf{u_m}]$$
5. Calcular la matriz de componentes principales $C \in M_{nxm}$ de la siguiente manera:
$$C = X.V$$
6. Cálculo de la matriz de calidades de individuos Q $\in M_{nxm}$:

$$Q_{ir} = \frac{(C_{i,r})^2}{\sum_{j=1}^{m}(X_{ij})^2} \quad \text{para } i = 1, 2, \ldots, n; \quad r = 1, 2, \ldots, m.$$

7. Calcular la matriz de coordenada de las variables $T \in M_{mxm}$
8. Calcular la matriz de calidades de las variables $S \in M_{mxm}$
9. Calcular el vector de inercias de los ejes $I \in M_{1xm}$ de la siguiente manera:

$$I = (100 \cdot \frac{\lambda_1}{m}, 100 \cdot \frac{\lambda_2}{m}, \ldots, 100 \cdot \frac{\lambda_m}{m})$$

Más información sobre el ACP se puede encontrar en el siguiente documento.

Requisitos

Los principales requerimientos de este mini proyecto son los siguientes:

1. Implementación de ACP en el primer lenguaje seleccionado y sus gráficos (40%).
2. Implementación de ACP en el segundo lenguaje seleccionado y sus gráficos (40%) .
3. Un reporte en (20%), describiendo lo siguiente:
    a. **Portada.**
    b. **Introducción.**
    c. **Descripción del problema** en sus propias palabras.
    d. **Justificación y explicación:** Listar los lenguajes seleccionados para resolver el problema además de proveer y sustentar su decisión técnica describiendo las características de los lenguajes.
    e. **Documentación de los códigos.** Se debe describir como se realizó la implementación del algoritmo en cada lenguaje, incluir diagrama de clases de ser necesario.
    f. **Ejemplos y demostraciones:** Imágenes selectas de la ejecución de sus soluciones.
    g. **Análisis.**
        i. ¿Considera que el lenguaje influyó al construir la solución?
        ii. ¿Experimento la diferencia de generar una solución para un mismo problema en lenguajes de paradigmas diferentes? ¿Cuales?
        iii. ¿Cuál sería el mayor aprendizaje que obtuvo de este proyecto?
        v. Cualquier otro comentario u observación relevante.
    g. **Dificultades encontradas**.

h. **Conclusiones.**

i. **Bibliografías o referencias.**

Otras Políticas

1. Este mini proyecto está diseñado para trabajar y entregarse **Grupalmente.**
2. Se permite el uso de **cualquier lenguaje de programación** (EXCEPTO JAVA) para implementar este mini proyecto.
3. De conocer el lenguaje, se recomienda escribir el reporte/ informe final en **LaTeX**.
4. La entrega será mediante Canvas. El estudiante debe subir el Zip file con el reporte en PDF y código fuente. Se recomienda el uso colaborativo de github.
5. **NO debe usar repositorios públicos**.
6. El **plagio** será penalizado de manera severa.
7. Los estudiantes que entreguen mini proyectos 100% originales recibirán una nota parcial a pesar de errores existentes en la funcionalidad y otras imperfecciones en el reporte. En cambio, los estudiantes que presenten tareas que contengan material plagiado (código, texto, estadísticas) recibirán 0% automáticamente, y además serán remitidos al **comité de ética**.
8. Se tomará en cuenta el estilo de programación y **se substraerán puntos por código desordenado**.
9. Mini proyectos entregados después de la fecha de entrega solamente podrán recibir la mitad de la calificación final. Por esta razón, es posible que **un trabajo incompleto pero entregado a tiempo termine recibiendo mejor calificación que uno completo entregado un minuto tarde**.