Proyecto: PCA

Métodos Matemáticos Computacionales para Ciencia de Datos

Nora Isabel Pérez Quezadas

Rubén Sánchez Duque

Equipo 6:

Pérez Martínez Ángel Noel
Rivera Hernández Milena Fernanda
Cruz Mendoza Valentina Ayelen
Monroy Villegas Isaac
Zamora Antiga Ángel Javier

2 de diciembre del 2025

Contenido:

Prediciendo el rendimiento académico en exámenes de estudiantes a partir de sus hábitos de estudio, asistencia, participación de los padres y otros aspectos que influyen en el éxito académico usando Análisis de Componentes Principales.

Cruz Valentina, Monroy Isaac, Pérez Noel, Rivera Milena, Zamora Ángel

Métodos Matemáticos Computacionales para Ciencia de Datos

Diciembre 01, 2025.

Abstracto

Hoy en día el rendimiento académico de los estudiantes en todo el mundo se ve influenciado por una compleja interacción de factores demográficos, sociales y educativos. El presente proyecto utiliza un conjunto de datos sintético de más de 6,000 estudiantes, que incluye 20 características relacionadas con sus horas de estudio, asistencia, hábitos, participación de padres, acceso a recursos, nivel de motivación, entre otras, para identificar los principales factores que influyen en el rendimiento académico con el fin de simplificar y homogeneizar el conjunto de datos original y realizar predicciones. Se empleó el Análisis de Componentes Principales (PCA) para la reducción de dimensionalidad y la interpretación de características, lo que permitió condensar el espacio de variables en 16 componentes que explican el 85% de la varianza acumulada. Los componentes más significativos se asociaron con características relacionadas con el rendimiento académico previo, el género, la motivación académica de los estudiantes, actividades físicas y extracurriculares, horas de estudio, número de sesiones de tutoría, nivel de participación de los padres, entre otras. Posteriormente, estos componentes se utilizaron para realizar una predicción mediante el método de extrapolación visto en el curso junto con integrales numéricas para obtener densidades y visualizar valores esperados e intervalos de probabilidad. La implementación de PCA se compara con la disponible en la biblioteca de Scikit-learn. Este trabajo proporciona un marco analítico sólido para que las instituciones educativas identifiquen a los estudiantes en riesgo y comprendan los determinantes multifacéticos del éxito académico, lo que permite desarrollar estrategias de intervención y selección basadas en datos.

Objetivo: Aplicar PCA al conjunto de datos de rendimiento académico en exámenes de estudiantes para determinar los componentes principales junto con las variables más significativas para realizar una predicción usando extrapolación del rendimiento académico junto con una estimación de la densidad e integral numérica de estas variables.

1. Introducción

Hoy en día muchas instituciones están interesadas en incrementar el rendimiento de sus estudiantes para aumentar su reconocimiento local o bien, para el caso de las admisiones a universidades o instituciones de educación superior, intentar predecirlo con el fin de tener un estimado de eficiencia terminal.

Para tal fin existe un debate sobre cuáles características son realmente las que más influyen en el resultado de los estudiantes en alguna evaluación. Shoukat, A. et al. (2013) afirma que "Los futuros logros académicos están determinados por el rendimiento previo."(pg. 283) Es decir, debemos considerar el historial académico de un estudiante para poder conjeturar algo sobre sus próximas evaluaciones. Sin embargo, es innegable que existen factores externos a los propios estudiantes que influyen en su concentración y motivación para alcanzar ciertos objetivos, por ejemplo, una observación importante de Shoukat, A. et al. (2013) es que "Los niños provenientes de familias de bajos ingresos presentan más resultados en términos de aprendizaje; bajo nivel de alfabetización, baja tasa de retención, problemas de comportamiento escolar y mayores dificultades" (pg. 284). Por ende, podemos observar que son una amplia gama de características o variables las que influyen en el desempeño académico de un solo estudiante.

Es por ello que en este proyecto propondremos una manera de determinar las características principales que influyen en un resultado de un examen específico a partir de un conjunto de datos sintético utilizando el Análisis de Componentes Principales (de sus siglas en inglés PCA) con el objetivo de poder obtener un conjunto de datos que condense las características más influyentes conservando la relación entre ellas, para que sea más sencillo aplicar modelos de regresión si se requiere.

2. Marco Teórico

En este trabajo se utilizaron varias metodologías para obtener resultados cuantitativos, sin embargo, es conveniente indagar un poco en los tipos de rendimiento académico. A continuación se mencionan algunos tipos según González, D. (2015):

- El *rendimiento suficiente*: Es aquel que se obtiene como reflejo de las calificaciones de los exámenes y trabajos realizados.
- El *rendimiento satisfactorio*: no se refiere a lo que ha obtenido realmente el alumno, sino a lo que podría haber obtenido, dadas sus aptitudes y circunstancias; un alumno ha obtenido un rendimiento satisfactorio cuando ha rendido al máximo sus capacidades.
- *Rendimiento insuficiente*: Está referido al no logro o superación del estudiante de las mínimas capacidades que se le exige en el proceso de enseñanza aprendizaje. (s.p)

Análisis de componentes principales

Es fundamental conocer a fondo las estrategias y herramientas a usar en la manipulación de los datos, como el análisis de componentes principales (PCA), consistente en una transformación lineal que permite traducir o caracterizar un conjunto de datos cuyos campos están potencialmente correlacionados a un nuevo conjunto de dimensionalidad reducida cuyas columnas, llamadas componentes principales, carecen mutuamente de correlación.

Los componentes principales generan combinaciones lineales de las variables originales que mantienen cuanta información sea posible maximizando su varianza. (Mackiewicz A., Ratajczak W., 1992, pg. 303)

Para obtener éstos componentes principales se recurre a métodos matemáticos relacionados al álgebra lineal: Podemos percibir nuestro conjunto de datos como una matriz, donde cada registro es un punto (o vector) de variables aleatorias definido en un espacio cuya dimensión corresponde al número de columnas. Dado que se tienen características y registros finitos, es posible construir la matriz de varianzas y covarianzas para describir la relación entre éstas variables o datos.

Posteriormente, para crear una transformación lineal que eficazmente traduzca la información a vectores ortogonales que cumplan con el objetivo de maximizar la varianza, podemos fijarnos en los eigenvectores (o vectores propios) de la propia matriz de varianzas: Matemáticamente son linealmente independientes y son invariantes en dirección bajo una transformación lineal.

En consecuencia, los componentes principales corresponden a los eigenvectores asociados a los mayores eigenvalores, pues son los que explican la mayor proporción de la variabilidad en los datos.

Interpolación

De acuerdo con Burden y Faires (2011), la interpolación es el método mediante el cual se aproxima una función $f(x)$ a través de un polinomio $P(x)$ de grado n utilizando un conjunto discreto de $n+1$ puntos conocidos o nodos $(x_0, x_1, \ldots, x_n)$. El objetivo fundamental de este proceso es determinar un polinomio único, denominado polinomio interpolante, que satisfaga la condición de coincidencia exacta en los nodos dados, es decir, $P(x) = f(x)$ para cada $k = 1,2,\ldots,n$. La justificación teórica para el uso de polinomios en la interpolación reside en el Teorema de Aproximación de Weierstrass, el cual establece que toda función continua definida en un intervalo cerrado puede ser aproximada uniformemente por un polinomio con cualquier grado de precisión deseado.

Para este trabajo, buscamos encontrar un polinomio de grado n que ajuste el conjunto de datos (x_0, y_0), (x_1, y_1) , . . . , (x_m, y_m), usando la matriz de Vandermonde para encontrar los coeficientes de interpolación a partir de las siguientes ecuaciones:

$$
\begin{aligned}
P_n(x_0) = y_0 \quad &\Rightarrow \quad a_o + a_1x_0 + a_2x_0^2 + \cdots + a_{n-1}x_0^{n-1} + a_nx_0^n = y_0 \\
P_n(x_1) = y_1 \quad &\Rightarrow \quad a_o + a_1x_1 + a_2x_1^2 + \cdots + a_{n-1}x_1^{n-1} + a_nx_1^n = y_1 \\
\vdots \quad &\Rightarrow \quad \vdots \\
P_n(x_n) = y_n \quad &\Rightarrow \quad a_o + a_1x_n + a_2x_n^2 + \cdots + a_{n-1}x_n^{n-1} + a_nx_n^n = y_n
\end{aligned}
$$

Ecuaciones 1. Polinomios que representan la interpolación. University of Wisconsin–Madison (2013).

Que se traducen a la matriz de Vandermonde de dimensiones $m \times (n+1)$. Notemos que puede darse el caso $(m > n)$ donde tenemos más datos que el grado del polinomio.

$$
\begin{bmatrix}
1 & x_0 & x_0^2 & \dots & x_0^n \\
1 & x_1 & x_1^2 & \dots & x_1^n \\
1 & x_2 & x_2^2 & \dots & x_2^n \\
\vdots & \vdots & \vdots & \ddots & \vdots \\
1 & x_m & x_m^2 & \dots & x_m^n
\end{bmatrix}
$$

Matriz 1. Matriz de Vandermonde (V).

Y buscamos resolver el sistema $\mathbf{V}\vec{a} = \vec{y}$

Para lograrlo, utilizamos el método de mínimos cuadrados y ecuaciones normales, con lo que llegamos a la siguiente igualdad:

$$(V^TV)a = V^Ty$$

Si definimos $V' = V^TV$ y $b = V^Ty$, como V tiene rango completo, V' es simétrica y definida positiva. Usando la factorización de Cholesky que ya habíamos definido en la primera parte del curso junto con los solucionadores correspondientes podemos resolver $V'a = b$. Obteniendo, $V' = LL^T$.

Con la factorización podemos aplicar los solucionadores "Lx_b(L, b)" para $Ly = b$ y "Ux_b(L.T, y)" para $L^Ta = y$ para con el fin de encontrar a.

La calidad de nuestra implementación se evaluó con la norma euclidiana del residuo:

$$||e||_2 = ||Va - y||_2$$

Extrapolación

La extrapolación consiste en estimar valores objetivos, en nuestro caso, el resultado de un examen para valores fuera de nuestro intervalo de observación original. (Wikipedia, 2025)

Una vez obtenidos los coeficientes $a = (a_0, a_1, \dots, a_n)^T$que definen a nuestro polinomio $P_n(x)$ podemos utilizarlo para obtener predicciones sobre un nuevo conjunto de datos de entrada denotado como x_{futuro} en nuestro código.

Realizaremos lo anterior con el método de 'Extrapolación polinómica', primero encontraremos una nueva matriz de Vandermonde usando los nuevos puntos x_{futuro}.

$$
V_{futuro} = \begin{bmatrix}
1 & x_0 & x_0^2 & \dots & x_0^n \\
1 & x_1 & x_1^2 & \dots & x_1^n \\
1 & x_2 & x_2^2 & \dots & x_2^n \\
\vdots & \vdots & \vdots & \ddots & \vdots \\
1 & x_m & x_m^2 & \dots & x_m^n
\end{bmatrix}
$$

Y calculamos las predicciones como:

$$y_{futuro} = V_{futuro}a$$

Que es equivalente a evaluar cada punto como:

$$y_i = \sum_{j=0}^{n} a_j (x_i)^j$$

Sin embargo, debemos considerar que la extrapolación tiene limitaciones, en especial con polinomios de grados muy grandes ya que son susceptibles a los 'sobreajustes' lo que provoca oscilaciones que no representan realmente los datos. Por lo que interpretaciones lejanas al intervalo original deben interpretarse con cautela. (Chapra & Canale, 2015).

Integral numérica

Una vez obtenida la transformación por componentes principales, es necesario identificar la distribución de los nuevos datos. Con este fin, podemos aplicar integración numérica, que consiste en aproximaciones del área bajo una función o curva.

Cuando una función carece de integral primitiva o la captura de los datos fue realizada de forma empírica y no sigue ninguna función explícita, es posible realizar divisiones sobre el intervalo total de valores para, en cada uno, estimar la integral.

Aunque existen diversas formas de aplicación, nos centraremos particularmente en las sumas de Riemann: A las divisiones antes mencionadas las llamaremos particiones. Sobre cada intervalo de la partición, podemos estimar exactamente en su punto medio x_i el valor de la interpolación sobre los datos transformados. Así, *dibujamos* rectángulos cuya altura será exactamente este punto medio $f(x_i)$, calculamos sus áreas, y las sumamos para obtener un aproximado al área total. Esto es:

$$\int_a^b f(x)\,dx \approx h\,(\sum_{i=0}^{n-1} f((x_i + x_{i+1})/2))$$

Donde la partición es $\{a = x_0, x_1, \ldots, x_{n-1}, b = x_n\}$ y $h = (b-a)/n$.

Como se mencionó anteriormente, cada registro de nuestro dataset representa una variable aleatoria: Conocer su distribución facilita predecir su comportamiento al utilizar, por ejemplo, modelos estadísticos a priori.

De igual forma, si se desea obtener una expresión para la antiderivada, podemos usar un análogo a las sumas y obtener sus valores por intervalo de la partición:

$$F(x_k) = \int_{x_o}^{x_k} f(x)\,dx$$
$$\approx h\,(\sum_{i=0}^{k-1} f((x_i + x_{i+1})/2))$$

De modo que, sobre cada valor parcial obtenido de $F(x_k)$ con $k \in [1, n]$, podemos realizar interpolación nuevamente para así ajustar a una función más explícita.

3. Hipótesis y Metodología

Hipótesis general: A través del Análisis de Componentes Principales se podrán identificar los 3 factores más influyentes en la puntuación obtenida en los exámenes de alumnos de cierto nivel académico.

Hipótesis comprobable: Por conocimiento empírico, mientras más horas de estudio hagan los alumnos, aunado a un buen descanso, y un mayor porcentaje de asistencia, los estudiantes obtendrán mejores resultados en sus evaluaciones, por lo tanto estas variables son las que tendrán una mayor correlación positiva con el rendimiento académico del estudiante (Exam Score), y serán las variables que más contribuyan a explicar la varianza en los principales componentes obtenidos a través de PCA.

Análisis de componentes principales
Previamente se explicó el funcionamiento de PCA, este se puede resumir a un algoritmo con los siguientes pasos:

1. Centrar los datos
2. Calcular la matriz de covarianza
3. Calcular los eigenvalores y eigenvectores (valores y vectores propios)
4. Ordenar los eigenvalores de mayor a menor.
5. Seleccionar los componentes principales y proyectar los datos al nuevo espacio.

Para el propósito de aplicar PCA al dataset, se implementó en Python el algoritmo descrito como una clase. (ver apéndice 1)

Para la implementación de la clase **PCA_propio** es necesaria la utilización de la biblioteca Numpy, dado que las operaciones matriciales requeridas serían computacionalmente ineficientes de realizarse con Python 'puro'.

Parámetros de la clase:

- **n_pca**: Entero, None por defecto. Es el número de componentes (características) a los que se reducirá
- **scale_data**: Booleano, True por defecto. Si es True, estandariza los datos con la estandarización Z-Score($\frac{X-\mu}{\sigma}$). Si es False, únicamente centra los datos restando la media $(X-\mu)$. Esto es útil si los datos vienen previamente estandarizados

Atributos de la clase:

- **w_:** Se inicializa como None. Se convertirá en la **Matriz de Transformación.**
- **mean_:** Almacena el promedio (μ) de cada columna original. Es crucial guardarlo para poder aplicar exactamente el mismo centrado a datos futuros (conjunto de prueba o nuevos datos).
- **std_**: Almacena la desviación estándar (σ) de los datos.
- **var_exp_**: Es la varianza explicada, es decir, el porcentaje de varianza guarda cada componente.

- **cum_var_exp**: Varianza explicada acumulada; útil para decidir cuántos componentes conservar para retener un porcentaje deseado de información (ej. 95%).
- **cov_mat**: Es la matriz de covarianzas calculada.
- eig_vals: Son los valores propios encontrados.
- **eig_vecs**: Vectores propios relacionados a los valores propios.
- **eig_pairs**: Pares ordenados de valores y vectores propios.

Además del constructor, la clase tiene dos métodos: **fit** y **transform**.

- Metodo **fit**: Recibe como parámetros **self** (la instancia del objeto que se está creando o manipulando), y **X** (el conjunto de datos como un arreglo). Este método realiza las siguientes operaciones:

Centrado: Mueve los datos de modo que el centro esté en el origen (0,0).

$$X_{centrado} = X - \mu$$

Estandarización de los datos: Estandariza los datos centrados dividiéndolos por la varianza.

$$X_{std} = \frac{X_{centrado}}{\sigma}$$

Cálculo de la matriz de covarianzas : Se utiliza np.cov para obtener la relación entre todas las variables.

Este paso es fundamental para la obtención de los componentes principales, pues los elementos de su diagonal principal son las varianzas de cada variable individual, mientras que los elementos fuera de ella representan la covarianza entre diferentes pares de variables, es decir, cómo es que dos variables varían conjuntamente.

Cálculo de eigenvectores y eigenvalores: Para el cálculo de los eigenpares se utilizó la función numpy.linalg.eigh, optimizada para calcular los eigenpares de matrices simétricas, como lo es la de covarianza.

Ordenamiento: PCA busca la mayor varianza, es por eso que el método fit ordena los eigenpares obtenidos de mayor a menor.

Selección de componentes: Solo selecciona las n componentes requeridas y las asigna a la matriz de transformación (self.w_).

- Método **transform**: Este método aplica lo aprendido para modificar datos. Recibe X previamente ajustado, si no lo está, no permite transformar. Estandariza aplicando la fórmula $Z = \frac{X - \mu}{\sigma}$. Utiliza la media y desviación estándar calculada previamente en el entrenamiento (método fit). No recalcula la media de los datos nuevos.

Finalmente realiza la **proyección**. Obtiene el producto punto de los datos originales por la matriz W:

$$X_{pca} = X_{std} \cdot W$$

Extrapolación y la interpolación

Función matriz_vandermonde

(Apéndice 2) Esta es una función auxiliar para construir manualmente la matriz necesaria para el ajuste polinómico, elevando los valores de entrada a las potencias correspondientes.

Parámetros:

- **vector_x**: Vector que contiene los datos de la variable independiente con los que se construirá la matriz
- **grado:** Entero que define el grado del polinomio deseado (n).

Retorno:

- **V**: La matriz de vandermonde resultante de dimensiones $m \times (n+1)$.

Desglose de la función:

La función realiza las siguientes operaciones:

1. **Inicialización**: Se define el tamaño de la matriz. El número de filas (m) corresponde a la longitud del vector_x, y el número de columnas es $n+1$ (el grado más el término independiente). Se inicializa una matriz de ceros con estas dimensiones.
2. **Llenado por iteración:** Se recorre cada fila (i) y cada columna (j) mediante ciclos anidados.
3. **Cálculo de potencias:** A diferencia de la implementación estándar que suele ir de menor a mayor grado, esta implementación calcula la potencia de forma descendente como potencia = n - j. Esto genera una estructura de columnas ordenada como: $x^n, x^{n-1}, \cdots, x^0$.
4. **Asignación:** Se asigna a cada celda $V[i, j]$ el valor de x_i elevado a la potencia calculada.

Función interpolación (Apéndice 3.)

Esta función implementa el método de mínimos cuadrados utilizando ecuaciones normales para encontrar el polinomio que mejor ajusta los datos proporcionados.

Parámetros:

- **x**: Vector de variables independientes de dimensión m.
- **y**: Vector de variables dependientes (los valores objetivo a predecir) de dimensión m.
- **grado**: El grado del polinomio (n) que se desea ajustar.

Retorno:

- **a**: Vector de coeficientes óptimos del polinomio, con dimensión $n+1$.
- **norma_error**: Un valor escalar que representa el error residual mínimo del ajuste ($||Va - y||$).

Desglose de la función:

El algoritmo sigue el flujo matemático de las Ecuaciones Normales:

1. **Construcción de la Matriz:** Llama a la función auxiliar matriz_vandermonde(x,grado) para obtener V.

2. **Formulacion de Ecuaciones Normales:** Se prepara el sistema lineal $(V^TV)a = V^Ty$.
    - Calcula la matriz del sistema A_sistema mediante el producto punto V^TV. Esta matriz resultante es simétrica y definida positiva.
    - Calcula el vector independiente b_sistema mediante V^Ty.
3. **Resolución del Sistema:** Se utiliza la factorización de Cholesky (mediante la función externa auxiliar Ax_b_cholesky) para resolver el sistema y encontrar el vector de coeficientes a.
4. **Cálculo de Predicciones:** Una vez obtenidos los coeficientes a, se calculan los valores estimados evaluando el polinomio en los puntos originales: $y_{prediccion} = V@a$.
5. **Evaluación del Error:** Se calcula el residuo como la diferencia entre los valores estimados y los reales ($residuo = y_{prediccion} - y$). Finalmente, se devuelve la norma Euclidiana de este residuo (norma_error) como medida de calidad del ajuste.

Extrapolación: Considerando la función encontrada a través de la interpolación, generamos nuevos puntos discretos y sobre ellos efectuamos el mismo proceso de interpolación.

4. Datos

Se empleó un conjunto de datos encontrado en Kaggle nombrado como: *"Student Performance Factors"*. Este conjunto de datos ofrece un panorama general sobre diversos factores que se piensa afectan el rendimiento estudiantil en los exámenes. Incluye información sobre hábitos de estudio, asistencia, participación de los padres y otros aspectos que influyen en el éxito académico. Está conformado por un total de 6,607 entradas y 20 características, tanto numéricas como categóricas.

El dataset puede encontrarse en el siguiente link: [Student Performance Factors]

A continuación se listan en la siguiente tabla las variables y las descripciones de lo que representan.

Variable	Descripción
1. Hours_Studied	Número de hora dedicadas al estudio po semana.
2. Attendance	Porcentaje de clases a las que asistió.
3. Parental_Involvement	Nivel de participación de los padres en la educación del estudiante (Bajo, Medio, Alto).
4. Access_to_Resources	Disponibilidad de recursos educativos (Bajo, Medio, Alto).
5. Extracurricular_Activities	Participación en actividades extracurriculares (Sí, No).
6. Sleep_Hours	Promedio de horas de sueño por noche.

7.	Previous_Scores	Calificaciones de exámenes anteriores.
8.	Motivation_Level	Nivel de motivación del estudiante (Bajo, Medio, Alto).
9.	Internet_Access	Disponibilidad de acceso a internet (Sí, No).
10.	Tutoring_Sessions	Número de sesiones de tutoría a las que se asistió por mes.
11.	Family_Income	Nivel de ingresos familiares (Bajo, Medio, Alto).
12.	Teacher_Quality	Calidad del profesorado (Baja, Media, Alta).
13.	School_Type	Tipo de escuela a la que asistió (Público, Privado).
14.	Peer_Influence	Influencia de los compañeros en el rendimiento académico (Positiva, Neutra, Negativa).
15.	Physical_Activity	Promedio de horas de actividad física por semana.
16.	Learning_Disabilities	Presencia de dificultades de aprendizaje (Sí, No).
17.	Parental_Education_Level	Nivel educativo más alto de los padres (Bachillerato, Universidad, Posgrado).
18.	Distance_from_Home	Distancia entre el hogar y la escuela (Cerca, Media, Lejos).
19.	Gender	Género del estudiante (Masculino, Femenino).
20.	Exam_Score	Calificación del examen.

Tabla 1. Variables y descripción.

Comenzamos con la exploración de los datos. Iniciamos identificando información general sobre el dataset.

```
data.info()

<class 'pandas.core.frame.DataFrame'>
RangeIndex: 6607 entries, 0 to 6606
Data columns (total 20 columns):
 #   Column                      Non-Null Count  Dtype
---  ------                      --------------  -----
 0   Hours_Studied               6607 non-null   int64
 1   Attendance                  6607 non-null   int64
 2   Parental_Involvement        6607 non-null   object
 3   Access_to_Resources         6607 non-null   object
 4   Extracurricular_Activities  6607 non-null   object
 5   Sleep_Hours                 6607 non-null   int64
 6   Previous_Scores             6607 non-null   int64
 7   Motivation_Level            6607 non-null   object
 8   Internet_Access             6607 non-null   object
 9   Tutoring_Sessions           6607 non-null   int64
 10  Family_Income               6607 non-null   object
 11  Teacher_Quality             6529 non-null   object
 12  School_Type                 6607 non-null   object
 13  Peer_Influence              6607 non-null   object
 14  Physical_Activity           6607 non-null   int64
 15  Learning_Disabilities       6607 non-null   object
 16  Parental_Education_Level    6517 non-null   object
 17  Distance_from_Home          6540 non-null   object
 18  Gender                      6607 non-null   object
 19  Exam_Score                  6607 non-null   int64
dtypes: int64(7), object(13)
memory usage: 1.0+ MB
```

Figura 1. Información sobre el dataset.

Observamos en las columnas de Teacher_Quality,Parental_Education_Level y Distance_from_Home hay nulos. Por lo que será necesario y conveniente rellenarlos con la mediana de cada columna.

```
# Sustitución por columna
for col in data.columns.values:
    falta = np.sum(data[col].isnull())
    if falta:
        print(f'Asignando {falta} valores en columna {col}')
        med = data[col].median()
        data[col] = data[col].fillna(med)

Asignando 78 valores en columna Teacher_Quality
Asignando 90 valores en columna Parental_Education_Level
Asignando 67 valores en columna Distance_from_Home
```

Figura 2. Eliminación de valores nulos.

Además, la mayoría de los tipos de datos son *object*, lo que refleja que son cadenas. Así que se debe hacer el mapeo correspondiente para trabajar únicamente con datos numéricos.

```
medida_map = {'Low': 1, 'Medium': 2, 'High': 3,
              'Yes': 1, 'No': 0,
              'Positive': 2, 'Negative': 0, 'Neutral': 1,
              'Near': 1, 'Moderate': 2, 'Far': 3,
              'Male': 0, 'Female': 1,
              'Public': 0, 'Private': 1,
              'High School': 1, 'College': 2, 'Postgraduate': 3}

for column in data.columns:
  if data[column].dtype == 'object':
    data[column] = data[column].map(medida_map)

data.tail()
```

Figura 3. Mapeo datos categóricos.

Posteriormente, podemos visualizar cada variable contra las demás (solo para las que originalmente eran numéricas), para visualizar si existen correlaciones.

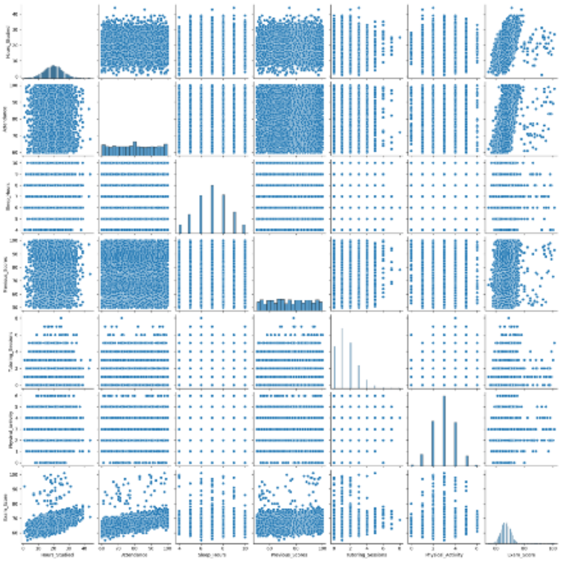

Figura 4. Información sobre el dataset.

Notemos que en la diagonal podemos visualizar ciertas distribuciones que parecieran conocidas, como una uniforme para la asistencia, una normal para horas de estudio, una exponencial para las sesiones de tutoreo, entre otras. Por otro lado, en cuanto a correlaciones con la última columna derecha que representa los resultados del examen, pareciera existir cierta correlación con las horas de estudio y asistencia, junto con resultados previos.

Por otro lado, también es importante identificar anomalías en los datos, comenzamos con una gráfica de bigotes para la columna de Exam_score:

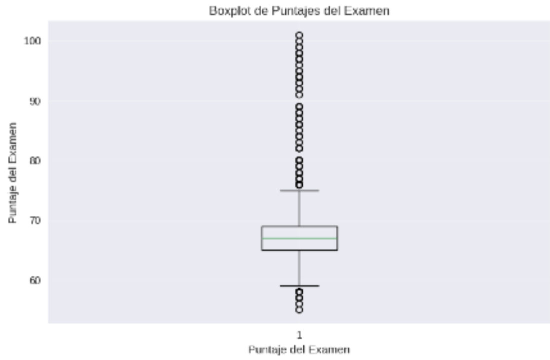

Figura 5. Boxplot Exam_Score.

Se observa que existen varios datos clasificados como anómalos que representan puntajes mayores a 75 aproximadamente, y menores a 60. Esto puede indicar que no es realmente común que los estudiantes obtengan resultados excepcionales en sus evaluaciones.

Posteriormente visualizamos la distribución de la columna *Exam_Score*:

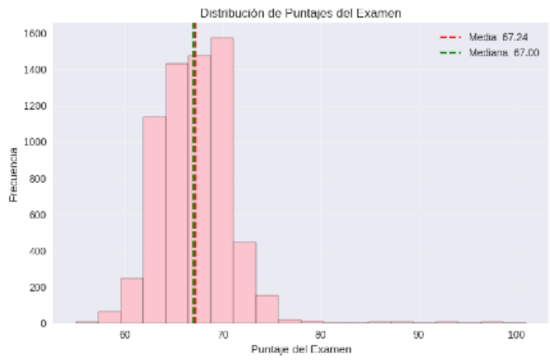

Figura 6. Distribución Exam_Score.

Vemos que se parece bastante a una normal centrada en el 68. Podemos hacer lo mismo para comparar horas de sueño con género.

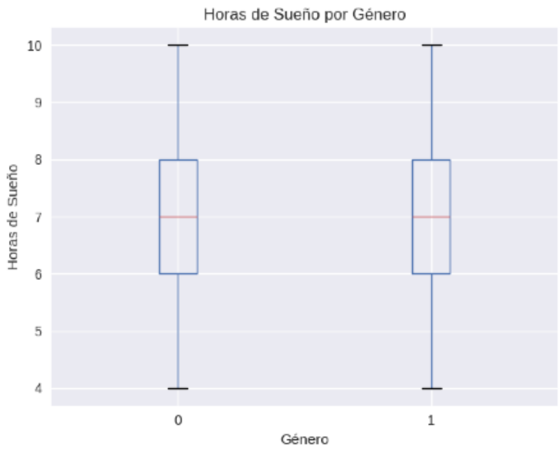

Figura 7. Box plot género con horas de sueño.

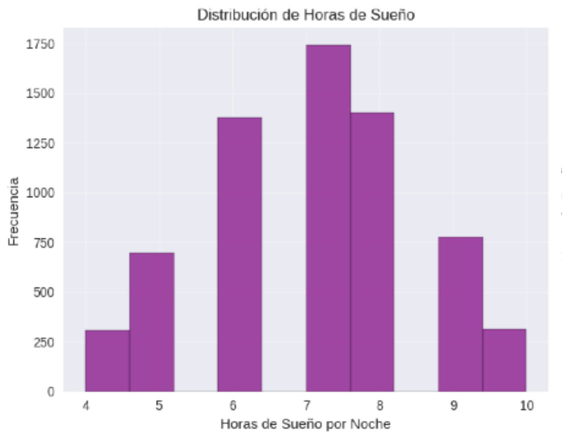

Figura 8. Distribución Horas de Sueño.

Observamos nuevamente que la distribución pareciera una normal centrada en 7 o una exponencial. En cambio, notamos que no tenemos anomalías para las horas de sueño y el género.

Asimismo, calculamos la matriz de correlaciones para visualizar un mapa de calor para las variables numéricas.

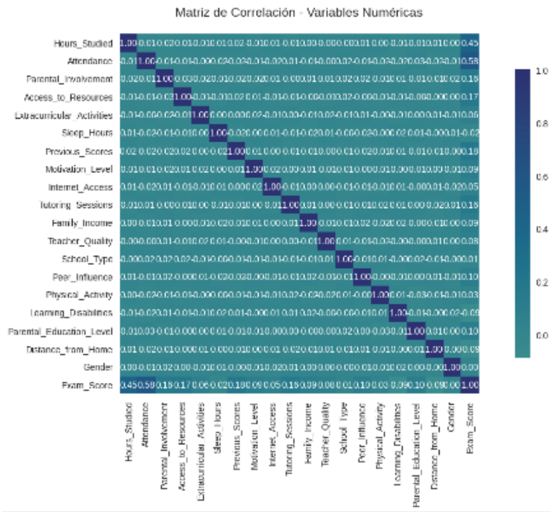

Figura 9. Mapa de calor de las correlaciones de las variables numéricas.

Destacamos que, en efecto, son muy pocas las características que parecieran tener relaciones positivas cercanas a 0.5, en este caso son: *Exam_Score* con *Hours_Studied* y *Attendance* que concuerdan con nuestras observaciones pasadas.

Por otro lado, podemos visualizar también el rendimiento promedio por género y por tipo de escuela.

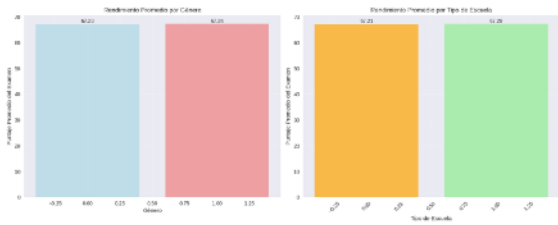

Figura 10. Promedio de rendimiento por género y tipo de escuela.

Y observamos que están bastantes balanceados, lo que podría indicarnos que el conjunto de datos sintético no incluye sesgos en estas 2 características.

Otra comparación relevante son las horas de estudio y la calificación en la evaluación. Las comparamos usando un scatter plot.

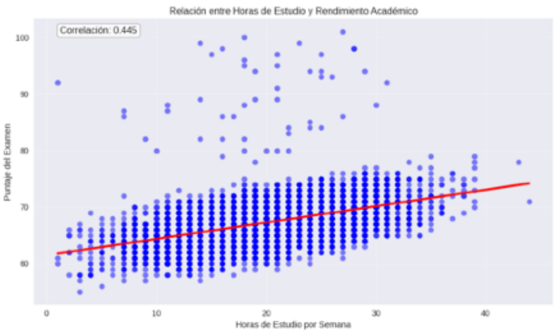

Figura 11. Correlación entre horas de estudio y calificación en exámenes.

Así que podemos observar una correlación positiva de 0.445 entre estas 2 características. Además, para la elección de la cantidad óptima de componentes principales en PCA nos apoyamos en los siguientes resultados:

Primero obtenemos la matriz de Covarianza utilizando la librería de numpy para álgebra lineal. (ver apéndice 4)

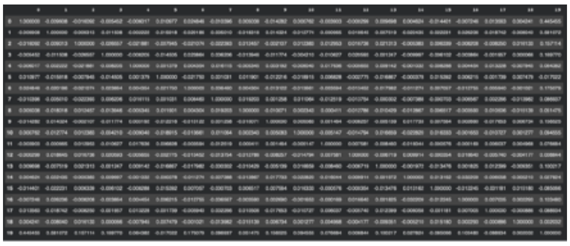

Figura 12. Matriz de Covarianza.

Luego, obtuvimos los vectores propios de esta matriz. (ver apéndice 5)

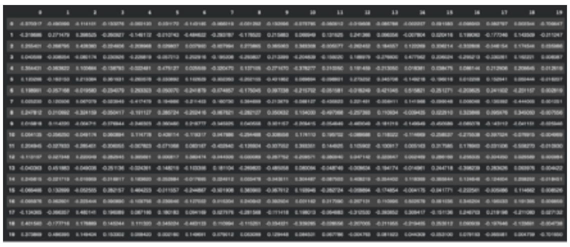

Figura 13. Vectores Propios de la Matriz de Covarianza.

Con esto podemos obtener los siguientes resultados respecto a la importancia de cada variable en cada componente: cada vector representa una Componente Principal y las entradas representan la influencia de cierta variable en esa componente. Asumiendo que aún no reducimos las dimensiones obtenemos lo siguiente:

1. Para la componente 1 la que más influencia es la variable 20 = Exam_scores con -0.70964745.

2. Para la componente 2 la que más influencia es la variable 7 = Previous_Scores con -0.48462183.

3. Para la componente 3 la que más influencia es la variable 3 = Parental_Involvement con 0.42838046.

4. Para la componente 4 la que más influencia es la variable 15 = Physical_Activity con 0.47756247.

5. Para la componente 5 la que más influencia es la variable 6 = Sleep_Hours con 0.47512713.

6. Para la componente 6 la que más influencia es la variable 10 = Tutoring_Sessions con -0.401862.

7. Para la componente 7 la que más influencia es la variable 15 = Physical_Activity con 0.51582127.

8. Para la componente 8 la que más influencia es la variable 19 = Gender con -0.44400501.

9. Para la componente 9 la que más influencia es la variable 12 = Teacher_Quality con -0.49736812.

10. Para la componente 10 la que más influencia es la variable 13 = School_Type con -0.46004853.

11. Para la componente 11 la que más influencia es la variable 18 = Distance_from_Home con -0.59702429.

12. Para la componente 12 la que más influencia es la variable 19 = Gender con -0.5582702.

13. Para la componente 13 la que más influencia es la variable 5 = Extracurricular_Activities con 0.39569092.

14. Para la componente 14 la que más influencia es la variable 10 = Tutoring_Sessions con -0.48505842.

15. Para la componente 15 la que más influencia es la variable 13 = School_Type con 0.40821919.

16. Para la componente 16 la que más influencia es la variable 8 = Motivation_Level con -0.50190751.

17. Para la componente 17 la que más influencia es la variable 15 = Physical_Activity con 0.5025795.

18. Para la componente 18 la que más influencia es la variable 3 = Parental_Involvement con 0.48014101.

19. Para la componente 19 la que más influencia es la variable 1 = Hours_Studied con 0.40155989.

20. Para la componente 20 la que más influencia es la variable 1 = Exam_score con -0.70165044.

Notemos que obtuvimos las siguientes frecuencias:

Variable	**Frecuencia**	**Componentes donde domina**
Physical_Activity	**3**	4, 7, 17
Exam_scores	**2**	1, 20
Parental_Involvement	**2**	3, 18
Tutoring_Sessions	**2**	6, 14
Gender	**2**	8, 12
School_Type	**2**	10, 15
Previous_Scores	**1**	2
Sleep_Hours	**1**	5
Teacher_Quality	**1**	9
Distance_from_Home	**1**	11
Extracurricular_Activities	**1**	13
Motivation_Level	**1**	16
Hours_Studied	**1**	19

Tabla 2. Frecuencia de variables significativas en cada componente principal.

Además, realizamos dos gráficas: La primera para visualizar la varianza acumulada.

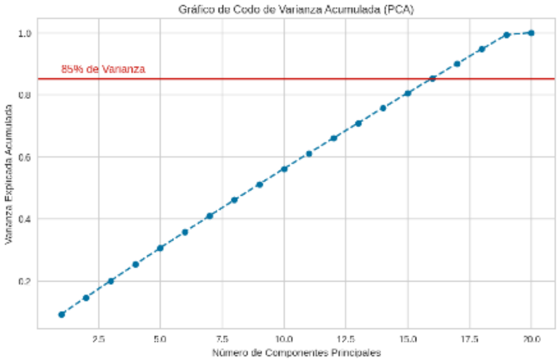

Figura 14. Gráfico de Codo de Varianza Acumulada.

La gráfica de la *Figura 14* muestra cuánta varianza total retienes al usar un número dado de componentes. La línea roja es el umbral que utilizamos como referencia para saber con cuánta varianza nos queremos quedar, el punto donde la línea azul (Varianza Acumulada) cruza por primera vez la línea roja es la cantidad de componentes principales a seleccionar.

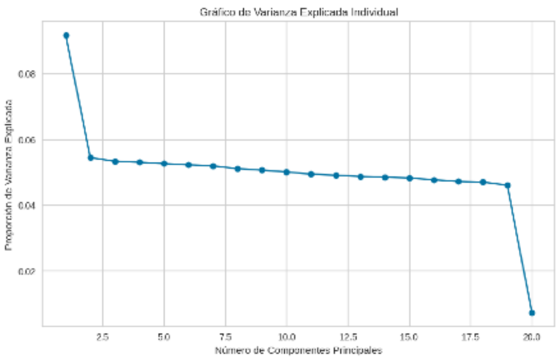

Figura 15. Gráfico de Varianza Individual.

La gráfica de la *Figura 15* muestra cuánta proporción de la varianza total aporta cada componente individualmente. Notemos 3 resultados importantes de la *Figura 15*:

1. Ausencia de componentes dominantes: La primera componente principal (PC1) explica únicamente cerca del 9% de la varianza total. Es decir, nuestro conjunto de datos realmente tiene baja correlación (mucha importancia de la información).
2. Distribución uniforme de la información: La curva presenta una pendiente muy suave (casi plana) desde la PC2 hasta la PC19. Esto indica que cada dimensión aporta una cantidad de información casi equivalente y única.
3. Interpretación de correlación: Ésta pendiente mínima sugiere que las variables originales presentan una baja correlación lineal entre ellas o que el conjunto de datos posee una alta complejidad dimensional (posiblemente ruido), lo que dificulta la reducción de dimensiones sin una pérdida considerable de información.

5. Resultados

Dado que hicimos el análisis con la gráfica de la varianza acumulada, concluimos que debíamos reducir a 16 características para conservar el 85% de la varianza acumulada. Con la implementación propia obtuvimos las siguientes Componentes Principales:

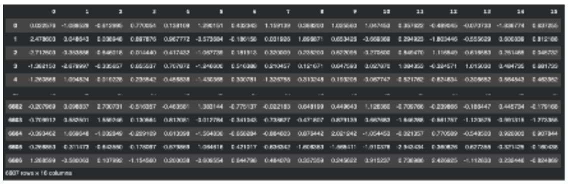

Figura 16. PCA propio con 16 componentes.

Y con sklearn lo siguiente:

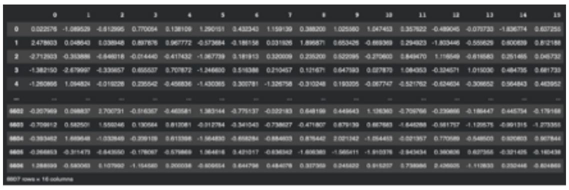

Figura 17. PCA de sklearn con 16 componentes.

Al comparar con la implementación de sklearn notamos que la única diferencia notable es el signo en nuestros componentes. Esto se lo atribuimos a que los vectores propios no tienen signo, es decir,

$$Av = \lambda v \iff A(-v) = \lambda(-v)$$

Además, a partir de nuestro análisis de influencia de variables en los vectores propios, obtuvimos que la variable con mayor frecuencia fue la actividad física:

- Componente 4 (0.477)
- Componente 7 (0.515)
- Componente 17 (0.502)

Así que usaremos esa variable para la **interpolación y extrapolación**:

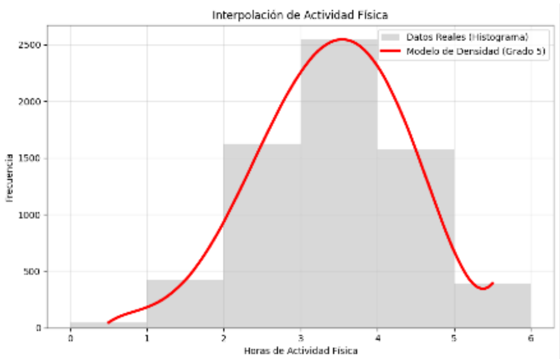

Figura 19. Interpolación de histograma de Horas de Actividad Física.

En la Figura 19 el histograma en gris representa la distribución de densidad de probabilidad de las horas de actividad física observadas. La curva roja representa el modelo de densidad ajustado mediante un polinomio de **quinto grado**.

- Coeficientes obtenidos (a): [31.30833333 -411.39583329 1774.06249975 -2894.01041606 2217.52760354 -536.28515606]
- Error residual: 0.0000

Observamos que los valores obtenidos para los coeficientes presentan magnitudes significativas y signos alternados. Además, elegimos realizar el ajuste con un polinomio de grado 5 ya que probamos con otros grados mayores y menores pero este fue el que nos dió un error residual igual a 0. Sin embargo, somos conscientes de que esto puede involucrar sobreajuste al extrapolar.

Al analizar la gráfica nos es posible ver que la variable *Horas de Actividad Física* aproxima a una distribución normal ligeramente sesgada. Donde la moda, es decir, la mayoría de los estudiantes, realiza entre 3 y 4 horas de actividad física por semana. Respecto a los valores extremos se observa una frecuencia muy baja, indicando que es inusual que los estudiantes sean totalmente sedentarios o, por el contrario, extremadamente activos.

Aplicando extrapolación (*figura 20*) se intenta realizar una predicción de la probabilidad de encontrar la frecuencia de ejercicio que tendrán alumnos futuros. Sin embargo, observamos que la línea azul punteada (la predicción) se dispara drásticamente hacia arriba, lo cual es contraintuitivo, pues se esperaría que la curva se asemejara al modelo de distribución, pues es poco probable encontrar alumnos que se ejerciten en exceso. El modelo predice erróneamente que será altamente probable encontrar casos extremos, validando que —como se mencionó en el marco teórico— la extrapolación polinómica directa no es un método fiable para predecir probabilidades a largo plazo.

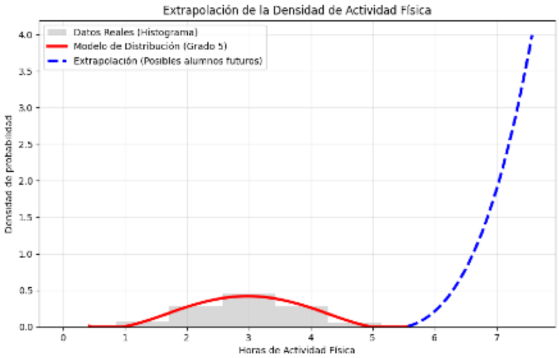

Figura 20. Extrapolación de histograma de Horas de Actividad Física.

Ahora, realizamos interpolación sobre el histograma de nuestro 1° componente principal (que representa la dirección de mayor varianza del conjunto de datos) obtenido con el PCA de implementación propia. Cabe mencionar que elegimos el grado para la interpolación iterando diversos valores y conservamos el que minimizó el error residual.

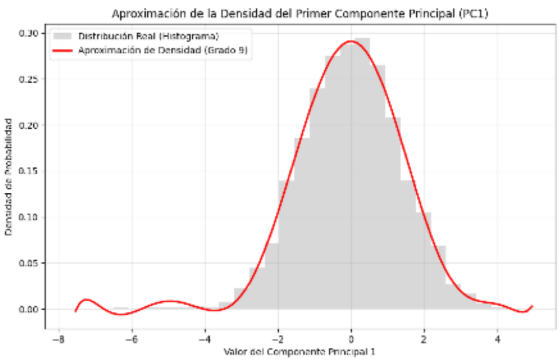

Figura 21. Aproximación de la densidad del Primer Componente Principal (PC1).

- Coeficientes obtenidos (a): [2.69308072e-07 3.05141696e-06 -1.35923239e-05 -2.16877964e-04 2.07006021e-04 5.72546978e-03 -9.11574987e-04 -6.66148998e-02 3.22876655e-04 2.90818606e-01]
- Error residual: 0.0356

Observamos que los coeficientes obtenidos para x_9, x_8 son números muy pequeños, lo que garantiza la estabilidad de la función en los extremos, mientras que los términos de menor grado definen la estructura central de la densidad.

Aquí, vale la pena mencionar la posible interpretación del gráfico de la Figura 22. Recordemos que nos quedamos con 16 de 20 componentes, dado que las variables originales eran estadísticamente independientes (o poco correlacionadas), es decir, cada variable aporta información única e independiente sobre el estudiante, impidiendo una reducción de la dimensionalidad. Recordando también que PC1 es una combinación lineal de todas las variables, y por análisis que habíamos hecho anteriormente, sabemos que la variable que más influye en este componente es *Exam Scores*. Pero si nos fijamos nuevamente en el primer eigenvector:

[-0.37031678, -0.4900989 , -0.11410127, -0.13327622, -0.05012039, 0.03117168, -0.14318455, -0.06601931, -0.03129229, -0.13209611, -0.07378527, -0.06061189, -0.01960847, -0.0857882 , -0.00223736, 0.09158251, -0.09695026, 0.08279691, 0.00234437, -0.70964745]

Vemos que dominan 3 variables:

1. Exam Score: Peso de -0.7096 (Esta variable define casi el 70% del comportamiento).
2. Attendance: Peso de -0.4901 (Muy influyente).

3. Hours_Studied: Peso de -0.3703 (Influyente).

Es decir,

$$PC1 \approx -0.71(ExamScore) -0.49(Attendance) - 0.37(Hours\_Studied)$$

Esto nos dice que un estudiante con ALTAS notas, asistencia y estudio tendrá un valor de PC1 muy negativo. En cambio, un estudiante con BAJAS notas y poca asistencia tendrá un valor de PC1 *positivo*.

Nos habíamos percatado anteriormente que estas tres variables estaban altamente involucradas en los valores de la variable objetivo a través de la matriz de correlación en la exploración de datos. Este análisis nos permitió corroborar esta teoría.

El resto tienen pesos muy pequeños, por lo que se pueden depreciar. Por lo anterior la densidad aproximada, que parece una Campana de Gauss, realmente nos dice que para valores negativos (Izquierda): Estudiantes de Alto Rendimiento tenemos altas notas, alta asistencia, mucho estudio. En cambio, los valores positivos (Derecha) son estudiantes de bajo rendimiento o en riesgo (Bajas notas, faltas, poco estudio). Observamos que la mayoría de los estudiantes se encuentran en esta parte.

Además, aplicamos extrapolación a este PC1.

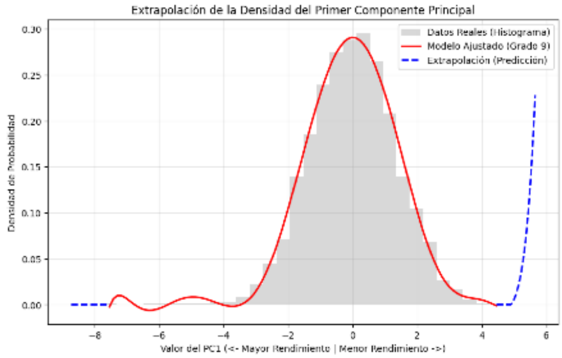

Figura 22. Extrapolación de la Densidad del Primer Componente Principal (PC1).

Notamos que para valores hacia la izquierda tuvimos que limitar los valores de la extrapolación para que no fueran negativos ya que recordemos que es una densidad de probabilidad y la probabilidad nunca puede ser negativa. Aunque ir hacia la izquierda implique encontrar un alumno con mejor rendimiento basado en **mejores notas y mejor asistencia**, vemos que nuestra interpolación forza a los valores a hacerse negativos, por lo que la extrapolación se queda en 0. Esto es consistente con la realidad. Existe un límite físico para la 'perfección académica'. El modelo predice que la probabilidad de encontrar estudiantes con un desempeño 'infinitamente superior' al máximo posible es nula.

En cambio, para los valores hacia la derecha observamos que la predicción (línea punteada azul) se dispara verticalmente hacia arriba, que podemos interpretar como que la probabilidad de encontrar estudiantes con 'pésimo rendimiento' crece hasta el infinito. En realidad, esperaríamos que la curva baje a cero (ya que los estudiantes con rendimiento extremadamente bajo son casos atípicos, no son la mayoría).

Con esto en mente, podemos decir que la interpolación encontrada (polinomio de grado 9) es excelente para interpolar (describir la población actual dentro del rango gris), pero inestable para predecir comportamientos extremos en el lado del bajo rendimiento.

Integral numérica

Posteriormente normalizamos para que en efecto fuera una densidad de probabilidad y obtuvimos lo siguiente:

- Área bajo curva original: 0.9827
- Área después de normalizar: 1.0000
- Valor Esperado (E[X]): 2.9800

El valor esperado resultó en un valor de 2.98 horas de actividad física a la semana.

En la Figura 23 la simetría de la curva roja sugiere que desviaciones positivas o negativas respecto a este promedio son casi igualmente probables. Además, podemos notar una similitud significativa con la campana de Gauss (Distribución Normal).

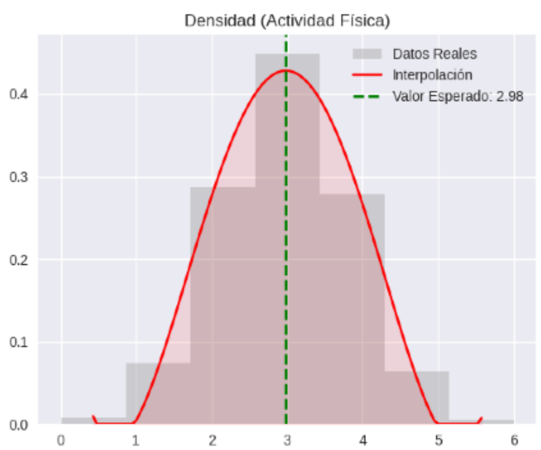

Figura 23. Densidad de Horas de Actividad Física.

Además, la gráfica de la Figura 24 nos muestra la función de distribución acumulada de la variable actividad física. Estadísticamente, se espera que la mayoría de los estudiantes dediquen cerca de 3 horas a la actividad física. En el contexto de la predicción académica, este valor actúa como un punto de referencia. Las desviaciones significativas de este valor (hacia el sedentarismo o hacia el exceso de actividad) son las "señales" que el modelo deberá evaluar para determinar si influyen positiva o negativamente en las calificaciones. Observamos que para valores muy pequeños o muy grandes, la probabilidad que aportan es muy poca, es decir, es muy poco probable que encontremos estudiantes que hacen muy poca o excesiva actividad física.

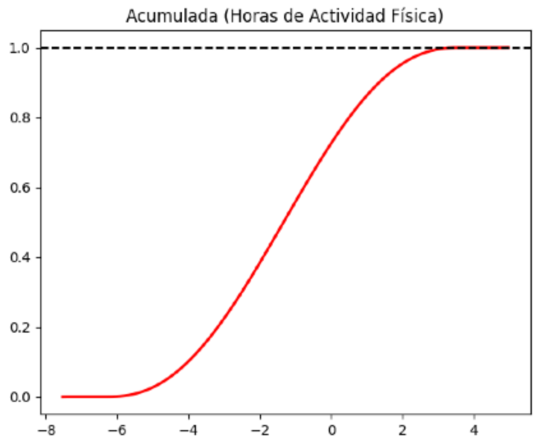

Figura 24. Función de Distribución Acumulada de Horas de Actividad Física.

Para la figura 25, elaborando y comparando las figuras 23 y 24, podemos observar más notoriamente la forma de campana invertida de la interpolación, con un crecimiento y decrecimiento bruscos en $x \approx 1$ y $x \approx 5$, respectivamente. Tanto la función de densidad interpolada como la función de densidad acumulada tienen simetría sobre el mismo eje, y la forma sigmoidal de la integral numérica contrasta con un punto mencionado anteriormente: Existe poca variación y los datos se concentran entre las 2 y 4 horas de actividad física haciendo la función estrictamente creciente (casi lineal entre estos valores), habiendo poca aportación de registros fuera de este rango.

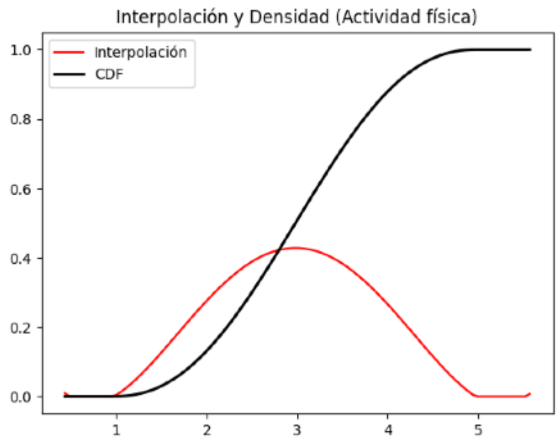

Figura 25. Comparación de interpolación y densidad de Horas de Actividad Física.

Posteriormente, obtuvimos la densidad para la Componente Principal 1.

- Área bajo curva original: 1.0012
- Área después de normalizar: 1.0000
- Valor Esperado (E[PC1]): -0.0277

Que el valor esperado haya sido negativo y cercano a 0 nos indica que cualquier estudiante que obtenga PC1 cercano al valor esperado será un estudiante promedio, es decir, no destacará por tener excelentes o muy malas notas. Es importante notar, que las horas de actividad física en este componente no tienen mucha importancia.

Por otro lado, la figura 26 ilustra la densidad de nuestro primer componente principal, el histograma gris, que muestra la frecuencia empírica de los datos proyectados, haciéndonos notar un centrado en el origen.

Se aplicó el método de interpolación sobre los puntos medios de los intervalos de clase (barras), lo que generó una curva continua (roja) $f(x)$.

Este ajuste actúa como una aproximación de la Función de Densidad de Probabilidad. La silueta resultante exhibe un comportamiento muy parecido al de una distribución Normal.

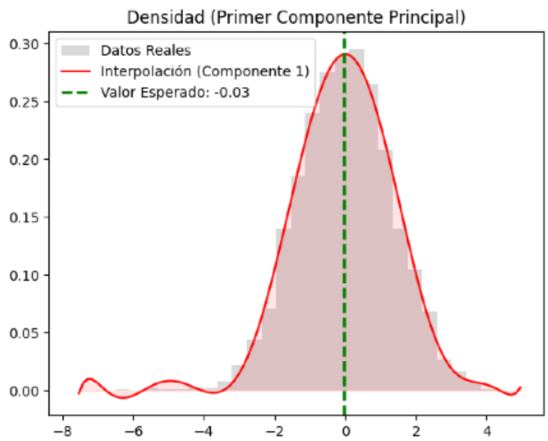

Figura 26. Densidad de PC1.

En la siguiente figura se grafica la función de distribución acumulada F(x), la cual es el resultado de aplicar la integral numérica sobre la función de densidad de la figura 26. El hecho más importante de esta gráfica es que el rango de la función comienza en 0.0 y termina asintóticamente en 1.0, confirmando por la propiedad de la suma de probabilidades que el método de interpolación modeló correctamente el área total, y que el algoritmo de Sumas de

Riemann funcionó con precisión, sin perder ni añadir área significativa.

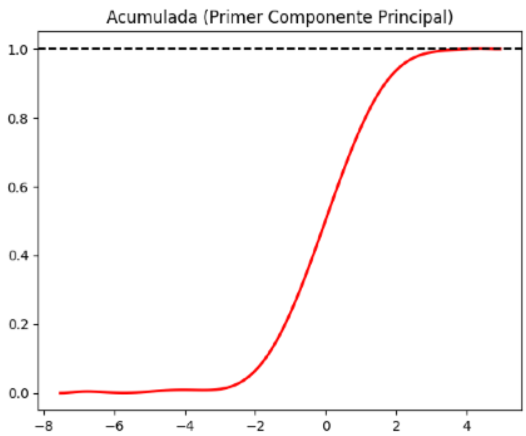

Figura 27. Función de distribución acumulada de PC1.

En la figura 28 se sobreponen la función de densidad de probabilidad (PDF) y la función de distribución acumulada (CDF).

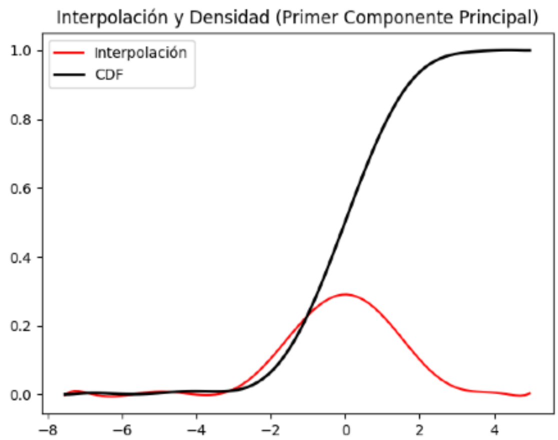

Figura 28. Comparación de interpolación y densidad de PC1.

6. Reflexión

Aunque el presente proyecto se basó en datos sintéticos, los métodos aplicados nos dan una noción de cómo influyen diversos factores en el desempeño de los alumnos en la realidad, pues permitió evidenciar que el rendimiento académico es un fenómeno complejo y multifactorial, que no se determina por una única variable, sino por una red de características que interactúan entre sí. La aplicación del Análisis de Componentes Principales (PCA) permitió revelar que el conjunto de datos posee una alta dimensionalidad efectiva, así —contrario a la expectativa de reducir el sistema a unos pocos factores dominantes— los resultados mostraron necesario la conservación de 16 componentes para explicar el 85% de la varianza. Este hallazgo es significativo, pues indica una baja correlación lineal entre las variables originales y una distribución uniforme de la información, sugiriendo que factores tan diversos como la actividad física, la participación de los padres y los hábitos de estudio también aportan de forma única en los resultados obtenidos por los estudiantes, validando parcialmente la hipótesis sobre la capacidad de condensación del PCA, pero a la vez resaltando la complejidad estructural de los datos, ampliando la comprensión más allá de las variables académicas tradicionales.

Desde una perspectiva metodológica, la implementación propia de algoritmos como PCA, interpolación y extrapolación no sólo validó los resultados obtenidos mediante bibliotecas estándar como *Scikit-learn*, sino que profundizó la comprensión de los métodos matemáticos utilizados. La integración de estos permitió transformar datos discretos en modelos continuos capaces de predecir tendencias, identificando variables clave como la actividad física y el historial académico previo como predictores importantes.

Si bien herramientas como la extrapolación demostraron ser útiles para estimar escenarios futuros, se concluye que deben ser aplicados con cautela, debido a la sensibilidad de los polinomios de alto grado a oscilaciones fuera del intervalo observado.

En suma, este trabajo proporciona un marco analítico robusto que puede auxiliar a las instituciones académicas a identificar patrones de éxito o riesgo, fundamentando la toma de decisiones y estrategias para aumentar su eficiencia terminal en datos cuantitativos precisos.

7. Bibliografía

- Shoukat, A. et al. (2013): *Factors Contributing to the Students Academic Performance: A Case Study of Islamia University Sub-Campus*. [PDF] Recuperado 19 de noviembre, 2025 de, https://www.researchgate.net/profile/Shoukat-Ali-4/publication/277898613_Factors_Contributing_to_the_Students_Academic_Performance_A_Case_Study_of_Islamia_University_Sub-Campus/links/5c604e5a45851582c3dd57c6/Factors-Contributing-to-the-Students-Academic-Performance-A-Case-Study-of-Islamia-University-Sub-Campus.pdf
- González, D. (2015). Investigación acerca del Rol del Valuador y Procedimientos en la Valuación. Sistema de Universidad Virtual de la Universidad de Guadalajara. [PDF]. Recuperado 30 de noviembre, 2025 de, https://www.coursehero.com/es/file/p6nu37t4/Tipos-de-Rendimiento-Acad%C3%A9mico-Es-necesario-distinguir-algunos-tipos-de/
- Carter N. et. al. (2021) *Data Science for Mathematicians*. Taylor & Francis Group. Recuperado 20 de noviembre, 2025 de, https://ebookcentral.proquest.com/lib/unam/reader.action?docID=6309040&c=UERG&ppg=2&nextModalAction=dW5hbSVyZWFkZXIlNjMwOTA0MCVyZXF1ZXN0RG93bmxvYWRNb2RhbCVzc29Mb2dpbkF0dGVtcHRlZA%3D%3D
- Mackiewicz A., Ratajczak W. (1993) Computers & Geosciences Vol. 19, No. 3. *PRINCIPAL COMPONENTS ANALYSIS (PCA)*. (pp 303 - 342). Recuperado el 20 de noviembre, 2025 de, https://www.sciencedirect.com/science/article/pii/009830049390090R
- Wikipedia (2025): *Extrapolación (matemática)*. Recuperado 20 de noviembre, 2025 de, https://es.wikipedia.org/wiki/Extrapolaci%C3%B3n_(matem%C3%A1tica)
- Brezinski C., Redivo-Zaglia M. (1994): Applied Numerical Mathematics 15. *Extrapolation Methods*. (pp 123 - 131). Recuperado el 19 de noviembre, 2025 de https://www.research.unipd.it/bitstream/11577/2495384/1/10.1016-0168-9274%2894%2900015-8.pdf
- Erik Iván Hernández Cosain Dr. Gonzalo Arreola Medina. (agosto de 2021). *EL RENDIMIENTO ACADEMICO Y SU RELACIÓN CON ALGUNOS FACTORES ASOCIADOS AL APRENDIZAJE EN ALUMNOS DE EDUCACIÓN SUPERIOR* (Universidad Pedagógica de Durango, Ed.). https://www.upd.edu.mx/PDF/Libros/RendimientoAcademico.pdf
- **Burden, R. L., & Faires, J. D. (2011).** *Numerical Analysis* (9th ed.). Brooks/Cole, Cengage Learning.
- University of Wisconsin–Madison (2013). Interpolation using the Vandermonde matrix. [PDF]. Recuperado 30 de noviembre, 2025 de,
- https://pages.cs.wisc.edu/~sifakis/courses/cs412-s13/lecture_notes/CS412_12_Feb_2013.pdf
- Chapra, S. C., & Canale, R. P. (2015). *Numerical Methods for Engineers* (7th ed.). McGraw-Hill Education.
- Dataset: https://www.kaggle.com/datasets/lainguyn123/student-performance-factors/code

8. Apéndice

1. Implementación de PCA con clases

```
import numpy as np

class PCA_propio():
 """ PCA_propio
 Implementación propia del PCA basado en Programación Orientada a Objetos.
 -----
 Params:
   n_pca: int
   Número de componentes a conservar; default = None
 -----
 Atributos:
   var_exp_ : 1d_array
     Varianza explicada individual
   cum_var_exp_ : 1d_array
     Varianza explicada acumulativa
   w_ : 2d_array
     Matriz de transormación con los vectores propios de los n_pca primeras
     componentes ordenadas
   cov_mat:
     Matriz de covarianzas
   eig_vals:
     valores propios encontrados
   eig_vecs:
     vectores propios relacionados a los eig_vals
   eig_pairs:
     pares ordenados de valores y vectores propios
 """
 def __init__(self, n_pca: int = 3, scale_data: bool = True):

   self.n_pca = n_pca
   self.scale_data = scale_data
   self.w_ = None
   self.mean_ = None
   self.std_ = None
   self.var_exp_ = None
   self.cum_var_exp_ = None
   self.cov_mat = None
   self.eig_vals = None
   self.eig_vecs = None

 def fit(self, X):
   # Preprocesamiento
   self.mean_ = X.mean(axis=0)
   X_centered = X - self.mean_

   if self.scale_data:
       self.std_ = X.std(axis=0,ddof=0)
       self.std_[self.std_ == 0] = 1.0
       X_processed = X_centered / self.std_
   else:
       self.std_ = np.ones(X.shape[1])
       X_processed = X_centered

   # Matriz de covarianzas
   self.cov_mat = np.cov(X_processed.T)
   # Cálculo de eigenvalores y eigenvectores
   self.eig_vals, self.eig_vecs = np.linalg.eigh(self.cov_mat)

   # Ordenamiento de Eignenpares
   # Crear pares (eigenvalor, eigenvector)
   self.eig_pairs = [[np.abs(self.eig_vals[i]), self.eig_vecs[:, i]] for i in range(len(self.eig_vals))]
```

2. Funcion Matriz de Vandermonde

```
def matriz_vandermonde(vector_x, grado):
  """matriz_vandermonde
  Funcion auxiliar que construye la matriz de Vandermonde manualmente.
  Filas: m (número de datos)
  Columnas: n + 1 (grado del polinomio + término independiente)
  ---
  params:
    vector_x: vector x con la que la construiremos
    grado: el grado de la matriz a contruir
  ---
  return:
    V: matriz de Vandermonde
  """
  m = len(vector_x)
  n = grado

  # Inicializamos una matriz de ceros de tamanio m x (n+1)
  V = np.zeros((m, n + 1))

  for i in range(m): # Recorremos cada fila
      for j in range(n + 1): # Recorremos cada potencia (columna)
          # x^n, x^(n-1), ..., x^0
          potencia = n - j
          V[i, j] = vector_x[i] ** potencia

  return V
```

3. Función de interpolación

```
def interpolacion(x, y, grado):
  """interpolacion
  Funcion que implementa la interpolacion.
  ---
  params:
    x: Vector de variables independientes. Dimension (m,).
    y: Vector de variables dependientes (valores a predecir). Dimension (m,).
    grado: Grado del polinomio deseado (n).
  ---
  return:
    a: Vector de coeficientes optimos del polinomio. Dimension (n+1,).
    norma_error: El valor escalar del error residual mínimo: ||Va - y||.
  """
  # Construimos V
  V = matriz_vandermonde(x, grado)

  # Resolvemos la Ecuación Normal (V^T V)a = V^T
  A_sistema = V.T @ V # Esta matriz es simetrica y def. positiva
  b_sistema = V.T @ y

  # a son los coeficientes y es el vector que buscamos
  # resolvemos con cholesky
  a = Ax_b_cholesky(A_sistema, b_sistema)

  # Calculamos el error ||Va - y||
  # y calculamos la prediccion con los puntos originales
  y_prediccion_puntos = V @ a

  # El residuo es la diferencia vector real - vector estimado
  residuo = y_prediccion_puntos - y

  norma_error = np.sqrt(np.sum(residuo**2))

  return a, norma_error
```

4. **Matriz de Covarianzas**
5. **Vectores propios (son las filas)**

	0	1	2	3	4	5	6	7	8	9
0	1.000151	-0.009909	-0.016094	-0.005453	-0.006018	0.010978	0.024850	-0.010397	0.005039	-0.014284
1	-0.009909	1.000151	-0.009315	-0.011510	-0.002223	-0.015920	-0.020189	-0.005011	-0.018320	0.014326
2	-0.016094	-0.009315	1.000151	-0.026561	-0.021885	-0.007946	-0.021077	-0.022397	0.012459	-0.002107
3	-0.005453	-0.011510	-0.026561	1.000151	-0.008206	-0.014507	0.023868	0.006207	-0.013948	-0.011776
4	-0.006018	-0.002223	-0.021885	-0.008206	1.000151	0.001379	0.004355	0.016118	-0.005346	0.003193
5	0.010978	-0.015920	-0.007946	-0.014507	0.001379	1.000151	-0.021754	0.001031	0.011903	-0.012218
6	0.024850	-0.020189	-0.021077	0.023868	0.004355	-0.021754	1.000151	0.006461	0.004305	-0.013124
7	-0.010397	-0.005011	-0.022397	0.006207	0.016118	0.001031	0.006461	1.000151	0.019206	0.001259
8	0.005039	-0.018320	0.012459	-0.013948	-0.005346	0.011903	0.004305	0.019206	1.000151	-0.010072
9	-0.014284	0.014326	-0.002107	-0.011776	0.003193	-0.012218	-0.013124	0.001259	-0.010072	1.000151
10	0.000762	-0.012776	0.012385	-0.004211	-0.009042	-0.018918	-0.013663	0.011066	0.002343	0.005084
11	-0.003904	-0.000965	0.012955	-0.010629	0.017639	0.006829	-0.003594	-0.012521	0.000411	0.001464
12	-0.000299	0.018647	0.016738	0.020596	-0.005656	-0.002776	-0.013454	-0.013756	-0.012788	-0.008259
13	0.009700	-0.007520	0.021316	-0.001247	0.009143	-0.016869	-0.017985	-0.000302	-0.010431	-0.005140
14	0.004625	-0.022438	-0.005384	-0.009998	-0.001032	-0.000378	-0.011275	-0.007389	-0.012869	0.017736
15	-0.014403	-0.022235	0.006340	-0.006103	-0.008290	0.015395	0.007058	-0.000703	0.006518	0.007565
16	-0.007247	0.026240	-0.008209	-0.003865	0.004454	0.006216	-0.012757	-0.006568	-0.003590	0.002690
17	0.013565	-0.018745	-0.008252	-0.001958	0.013230	-0.001739	-0.005941	0.002267	0.010508	-0.017656
18	0.004241	-0.008042	0.016135	0.000066	-0.007946	0.007480	-0.001021	-0.013984	-0.015141	0.006735
19	0.445522	0.581160	0.157138	0.169796	0.064391	-0.017024	0.175106	0.086951	0.051483	0.156549

	10	11	12	13	14	15	16	17	18	19
0	0.000762	-0.003904	-0.000299	0.009700	0.004625	-0.014403	-0.007247	0.013565	0.004241	0.445522
1	-0.012776	-0.000965	0.018647	-0.007520	-0.022438	-0.022235	0.026240	-0.018745	-0.008042	0.581160
2	0.012385	0.012955	0.016738	0.021316	-0.005384	0.006340	-0.008209	-0.008252	0.016135	0.157138
3	-0.004211	-0.010629	0.020596	-0.001247	-0.009998	-0.006103	-0.003865	-0.001958	0.000066	0.169796
4	-0.009042	0.017639	-0.005656	0.009143	-0.001032	-0.008290	0.004454	0.013230	-0.007946	0.064391
5	-0.018918	0.006829	-0.002776	-0.016869	-0.000378	0.015395	0.006216	-0.001739	0.007480	-0.017024
6	-0.013663	-0.003594	-0.013454	-0.017985	-0.011275	0.007058	-0.012757	-0.005941	-0.001021	0.175106
7	0.011066	-0.012521	-0.013756	-0.000302	-0.007389	-0.000703	-0.006568	0.002267	-0.013984	0.086951
8	0.002343	0.000411	-0.012788	-0.010431	-0.012869	0.006518	-0.003590	0.010508	-0.015141	0.051483
9	0.005084	0.001464	-0.008259	-0.005140	0.017736	0.007565	0.002690	-0.017656	0.006735	0.156549
10	1.000151	-0.005148	-0.014796	0.016662	-0.022823	0.016335	-0.001654	-0.010728	0.001277	0.094569
11	-0.005148	1.000151	0.007582	-0.008461	-0.016046	-0.000576	-0.000169	0.006038	0.004968	0.076696
12	-0.014796	0.007582	1.000151	-0.006720	0.009916	-0.000354	0.016642	-0.005741	-0.004178	0.008845
13	0.016662	-0.008461	-0.006720	1.000151	-0.001973	-0.013478	0.001825	0.012401	-0.009352	0.100232
14	-0.022823	-0.016046	0.009916	-0.001973	1.000151	0.013164	-0.032214	-0.006059	-0.005211	0.027829
15	0.016335	-0.000576	-0.000354	-0.013478	0.013164	1.000151	-0.012247	-0.001181	0.015182	-0.085079
16	-0.001654	-0.000169	0.016642	0.001825	-0.032214	-0.012247	1.000151	0.007006	0.002294	0.103496
17	-0.010728	0.006038	-0.005741	0.012401	-0.006059	-0.001181	0.007006	1.000151	-0.000886	-0.088948
18	0.001277	0.004968	-0.004178	-0.009352	-0.005211	0.015182	0.002294	-0.000886	1.000151	0.002033
19	0.094569	0.076696	0.008845	0.100232	0.027829	-0.085079	0.103496	-0.088948	0.002033	1.000151

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19
0	-0.370317	-0.490099	-0.114101	-0.133276	-0.050120	0.031172	-0.143185	-0.066019	-0.031292	-0.132096	-0.073785	-0.060612	-0.019608	-0.085788	-0.002237	0.091583	-0.096950	0.082797	0.002344	-0.709647
1	-0.318686	0.271479	0.398525	-0.260927	-0.146172	-0.010743	-0.484622	-0.293787	-0.178520	0.215883	0.066949	0.131625	0.241366	0.066356	-0.007804	0.020416	0.199060	-0.177246	0.143509	-0.011247
2	0.255401	-0.268795	0.428380	-0.224606	-0.208968	0.029807	0.037950	-0.007994	0.273865	0.065063	0.383308	-0.005577	-0.262452	0.184557	0.122269	0.306214	-0.332806	-0.046154	0.174544	0.035986
3	0.040599	-0.008054	-0.080174	0.230926	-0.226819	-0.057213	0.202916	-0.195208	-0.293837	0.213399	-0.204839	-0.159220	0.188979	-0.276600	0.477562	0.206024	-0.295213	-0.330261	0.162221	0.008387
4	0.354431	-0.063822	0.100664	-0.138793	-0.022481	0.475127	0.025559	-0.320470	0.137105	-0.271670	-0.378277	0.310050	0.191459	-0.213050	0.018381	0.038475	0.086144	0.212906	0.206645	0.012619
5	0.120266	-0.163153	0.213384	0.361631	-0.260578	-0.333692	0.102629	-0.302263	-0.202155	-0.401862	0.089694	-0.098801	0.273252	0.345708	-0.149218	-0.196016	0.012206	0.152641	0.050444	-0.016207
6	0.198991	-0.057168	-0.019580	-0.234079	0.293323	-0.050070	-0.241879	-0.074657	-0.175045	0.097238	-0.215702	-0.051581	-0.018249	0.421045	0.515821	-0.251271	-0.203625	0.241502	-0.231157	0.002819
7	0.025230	0.120506	0.067079	-0.023949	-0.417479	0.194866	-0.211453	0.160730	0.394699	-0.213879	-0.088127	-0.435823	0.221491	-0.056411	0.141988	-0.099048	0.006066	-0.135392	-0.444005	0.001251
8	0.247812	0.010692	-0.324159	-0.050417	-0.191127	0.285724	-0.202416	-0.067821	-0.282127	0.050632	0.154030	-0.497368	-0.257393	0.110934	-0.039435	0.022910	0.323866	0.095976	0.345050	-0.007556
9	0.016818	0.114220	-0.004711	-0.376844	-0.246323	-0.260460	0.218777	-0.345925	0.040556	0.001157	-0.209415	-0.054849	-0.460049	-0.181219	-0.149949	-0.450589	-0.085078	-0.143112	-0.041151	-0.025946
10	0.054135	-0.056250	-0.049174	0.060894	0.114778	0.409114	-0.119317	0.047886	-0.294488	-0.308558	0.174110	0.195702	-0.088688	0.118322	-0.114669	-0.258537	-0.275538	-0.597024	-0.076915	-0.004969
11	0.204945	-0.027933	-0.285451	-0.206055	-0.007823	-0.071068	0.083187	-0.402840	-0.126924	-0.007552	0.393351	0.144925	0.105902	-0.100917	0.005163	0.317585	0.178900	-0.031506	-0.558270	-0.010930
12	-0.113197	0.027348	0.220049	-0.282645	0.395691	0.000817	0.380474	-0.044509	-0.030089	-0.267752	-0.209571	-0.380940	0.047142	0.223647	0.002469	0.286169	0.256335	-0.304350	0.026589	0.000984
13	-0.040363	0.451683	-0.049036	-0.251136	-0.024391	-0.148219	-0.103356	0.181104	-0.269820	-0.485058	0.080094	-0.048749	-0.008634	-0.194174	-0.014961	0.244718	-0.398239	0.283626	0.093976	0.004422
14	0.245615	-0.221719	-0.016959	-0.316817	0.183620	-0.252684	-0.017695	0.224912	0.033478	-0.043511	0.304487	-0.087503	0.408219	-0.354402	0.118399	-0.365644	0.124646	-0.124004	0.258252	-0.018451
15	-0.066466	0.132699	-0.052555	0.282157	0.464223	-0.011557	-0.244887	-0.501908	0.383900	-0.067612	0.193946	-0.282724	-0.058894	-0.174854	-0.004175	-0.041771	-0.222561	-0.005086	0.114662	0.008526
16	-0.095976	0.062601	-0.223444	0.090890	-0.109756	-0.239946	-0.127022	0.015204	0.240942	-0.392504	0.031162	0.317390	-0.257131	0.110995	0.502579	0.061035	0.345204	-0.195533	0.191395	0.009859
17	-0.134365	-0.066357	0.480141	0.196089	0.087166	0.180183	0.094169	0.027976	-0.281568	-0.111418	0.198013	-0.064883	-0.312530	-0.393852	0.306417	-0.151136	0.246703	0.219198	-0.211080	0.027132
18	0.401560	-0.177716	0.178889	0.145244	0.111320	-0.345024	-0.463133	0.110994	-0.115251	-0.034321	-0.339285	-0.028556	-0.207005	-0.211955	-0.219455	0.253512	0.090939	-0.197646	-0.133691	-0.004738
19	0.373869	0.486395	0.149434	0.153302	0.058420	0.002160	0.149691	0.079012	0.053099	0.129448	0.084531	0.067786	-0.004763	0.081923	0.044309	-0.053100	0.079193	-0.065581	0.004759	-0.701650