IMEC-2001 AC1: Percepción de Conocimientos

1. TENER EN CUENTA

La entrega de la **Actividad Clase 1** es en formato PDF y se debe enviar vía Bloque Neón con el desarrollo escrito de los ejercicios con las siguientes condiciones:

- El nombre del archivo debe ser **AC1_NombreApellido.pdf**.
- La fecha límite de entrega en Bloque Neón es: **14:00 de agosto 14 de 2023**.
- La entrega es individual.

2. DESCRIPCIÓN DE LA ACTIVIDAD

La actividad tiene como objetivo percibir el conocimiento general para solucionar ejercicios de ingeniería. Cada estudiante debe responder en forma de **texto** el procedimiento a seguir para dar solución a los ejercicios planteados. Este debe incluir las herramientas utilizadas en el proceso de solución (por ejemplo, 'en Excel utilizo tal comando…', 'en MATLAB utilizó tal función…', 'en Python empleo el código…').

3. EJERCICIOS

Ítem 1: Ajuste de Datos

ENUNCIADO

En la ruta **week1 > data > engines.xlsx** dispone datos de las variables `Mass (Kg)` y `Revolutions per Minute (RPM)`. A partir de estos datos se requiere:

1. Realizar la regresión lineal (línea de tendencia).
2. Extraer los valores de pendiente e intercepto.
3. Estimar el valor de R^2.

SOLUCIÓN

<Escriba su respuesta aquí>.

Todo lo que explicaré está en código de Python y fue visto en la clase de métodos computacionales 1 del semestre pasado:

1. Método 1: Mínimos cuadrados (método de regresión logística), lo que se hace es definir una función costo ji-cuadrado con cierto modelo matemático que describa los datos (en este caso una recta). Cabe recalcar que esta función costo se puede interpretar en este caso (sin incertidumbres entre los datos) como el error total entre el modelo y los datos. Básicamente es encontrar los parámetros que minimicen la función costo mediante cosas como descenso del gradiente (machine-learning) o expresarlo analíticamente si es posible. De hecho, en el caso de la regresión lineal, se pueden despejar relativamente fácil los parámetros (de forma que no hay que usar regresión del gradiente para minimizar numéricamente la función costo).

Método 2: Mínimos cuadrados (método de regresión matricial), lo que se hace es utilizar herramientas matemáticas de álgebra lineal para calcular los parámetros que se ajustan a los datos. Este es un método analítico mucho más simple para realizar estos tipos de regresiones polinómicas. Sin embargo, no funciona en el caso de funciones que requieren encontrar parámetros que no pertenecen a un conjunto de subfunciones linealmente independientes.

Método 3: Bootstrapping (método de Bradley Efron 1979). Este método es mucho más sofisticado porque convierte los datos en un histograma haciendo uso de números aleatorios, lo que permite hallar los parámetros de ajuste con su correspondiente incertidumbre con el uso de la librería corner (algo que con la mayoría de métodos no se puede). No obstante, esto es para obtener resultados más rigurosos al nivel de una tesis.

2. Con cualquiera de los tres métodos se pueden hallar los parámetros (pendiente e intersección) porque eso es lo que retornan las funciones de cada método.

3. Tengo tres formas de calcular el Coeficiente de determinación R^2:
   - Usando valores del eje y: Sumatoria del cuadrado de la diferencia entre lo predicho y lo observado, dividido entre la sumatoria del cuadrado de la diferencia entre los observado y la media. Luego, restar eso al número uno.
   - Usando valores del eje y: Sumatoria del cuadrado de la diferencia entre lo predicho y media, dividido entre la sumatoria del cuadrado de la diferencia entre los observado y la media.
   - Usando coordenadas de los puntos: El cuadrado de la covarianza dividido entre la multiplicación de las varianzas de cada variable (aplica solo para regresiones lineales).

   Cada una de estas formas se pueden hacer relativamente fácil mediante *for loops* o funciones de numpy y evaluando el modelo cada vez que sea necesario con sympy.

Ítem 2: Interpolación

ENUNCIADO

Se realizaron las siguientes cinco mediciones de datos:

Y	0	6	12	20	26
X	655	645	620	560	455

A partir de estos datos se requiere conocer el valor de Y cuando X toma los valores de 650, 580 y 461.

SOLUCIÓN

<Escriba su respuesta aquí>.

En primer lugar, pondría los datos en array de numpy. Luego, la idea es encontrar una función que represente los datos. Lo puedo realizar mediante interpolación de Lagrange o interpolación de Newton. Básicamente es implementar las fórmulas de estos métodos matemáticos a programación.

Ítem 3: Raíces

ENUNCIADO
Los datos de esfuerzo (σ) y deformación (ε) tomados a partir de una prueba de tracción se disponen en la ruta **week1 > data > strain-stress.csv**. La región elástica se describe con la ecuación $\sigma = 23268\varepsilon + 11$ mientras que la región plástica se describe con la ecuación $\sigma = 26202\varepsilon^3 - 11347\,\varepsilon^2 + 1332\varepsilon + 78$.

El límite elástico es el punto en donde se pasa se la región elástica a la región plástica. En otras palabras, el límite elástico se obtiene en el punto $\sigma_{elástica} = \sigma_{plástica}$.

Estime el límite elástico.

SOLUCIÓN
<Escriba su respuesta aquí>.
Es básicamente solucionar una ecuación de grado 3, algo que puede ser tedioso de realizar a mano. Para este tipo de cosas existe sympy.solve, de manera que solo se pone la ecuación de la forma f(x) y se solucionará. Otra forma es usar el método de Newton-Raphson para hallar las raíces del polinomio, pero es menos inexacto y llega a requerir más recursos computacionales.

Ítem 4: Sistemas de Ecuaciones

ENUNCIADO
Las ecuaciones que describen un sistema mecánico en estado estático son:

$$-A\cos 2.92° - B\sin 33.8° - 0.601L\sin 47.9° + 1.02L\cos 78.8° = 0$$
$$-A\sin 2.92° + B\cos 33.8° - 0.601L\cos 47.9° - 1.02L\sin 78.8° = 0$$

Siendo $L = 15$ N. Estime las fuerzas A y B.

SOLUCIÓN
<Escriba su respuesta aquí>.
Lo mismo que el ítem anterior, se usa sympy.solve para resolver sistemas de ecuaciones con sus respectivas variables.

Ítem 5: Optimización

ENUNCIADO
Los datos de esfuerzo (σ) y deformación (ε) tomados a partir de una prueba de tracción se disponen en la ruta **week1 > data > strain-stress.csv**. La región elástica se describe con la ecuación $\sigma = 23268\varepsilon + 11$ mientras que la región plástica se describe con la ecuación $\sigma = 26202\varepsilon^3 - 11347\,\varepsilon^2 + 1332\varepsilon + 78$.

El esfuerzo último es el valor máximo de esfuerzo de la región plástica: $\max \sigma_{plástica}$.

Estime el esfuerzo último.

SOLUCIÓN

<Escriba su respuesta aquí>.

Una forma sería usar la librería de sympy.diff para hallar la primera derivada de la función en cuestión e igualarla a 0 y usar sympy.solve para resolverlo, lo que daría valores de x. En caso de que haya más de una solución, entonces es solo comparar los valores del eje vertical para ver cuál es el mayor, lo que sería el máximo de la función.

Ítem 6: Incertidumbre

ENUNCIADO

Se realizaron las siguientes cinco mediciones de datos de masa (m) y velocidad (v):

m	3.1	2.8	2.9	3.0	3.2
v	9.9	10.5	10.2	9.6	9.9

A partir de esta información:

1. Estime la incertidumbre asociada a cada variable.
2. Estime la energía cinética ($E_c = \frac{1}{2}mv^2$) y su incertidumbre asociada.

SOLUCIÓN

<Escriba su respuesta aquí>.

En primer lugar, pondría los datos en un array de numpy por separado.

1. La incertidumbre se saca con la desviación estándar. Para esto es simplemente usar numpy.std para obtener el resultado.
2. Para esto se usa la ley de propagación de la incertidumbre. Se puede usar la librería de sympy para las derivadas y evaluaciones de cada variable. Sin embargo, se puede hacer a mano.

Ítem 7: ODE

ENUNCIADO

El comportamiento dinámico de un sistema masa-resorte-amortiguador se puede describir a partir de la masa (m), la constante de rigidez del resorte (k) y la constante de amortiguamiento (c) y una fuerza externa (F) que incita el movimiento.

$$\ddot{x} + \frac{c}{m}\dot{x} + \frac{k}{m}x = \frac{F}{m}$$

Asumiendo que $m = 5$ kg, $k = 33$ N/m, $c = 10$ Ns/m, $F = 90$ N, $\ddot{x}_{t=0} = 0$ y $\dot{x}_{t=0} = 10$:

1. Estime el comportamiento de la velocidad con respecto al tiempo (t) desde $t = 0$ hasta $t = 25$.
2. Estime el comportamiento de la posición con respecto al tiempo (t) desde $t = 0$ hasta $t = 25$.

SOLUCIÓN

<Escriba su respuesta aquí>.

Lo que se puede hacer es solucionar de una vez la ecuación diferencial con sympy.dsolve, sympy.Function, sympy.Symbol, sympy.Eq, sympy.diff y sympy.solve para poder hacer lo demás de la siguiente forma:

1. Derivar la solución de la ecuación diferencial para obtener la velocidad y luego crear un array en numpy para los valores del tiempo. Luego usar sympy.lambdify para evaluar la función en cada valor del tiempo y así graficar en matplotlib.pyplot.
2. Ya teniendo la solución de la ecuación diferencial se hace lo mismo que describí anteriormente.