

Examen de reposición

Python for Finance

Universidad Francisco Marroquín

17 de noviembre de 2025

Nombre y apellidos: _____

Duración: 1 hora y 30 minutos

Puntuación total: 10 puntos

Recursos permitidos: Ninguno (sin ordenador, sin apuntes)

Ejercicio 1 – Python y matemáticas financieras (2 puntos)

Una renta financiera consiste en pagos periódicos que crecen en progresión geométrica. El primer pago es C_1 , el segundo es $C_1 \cdot (1 + g)$, el tercero $C_1 \cdot (1 + g)^2$, y así sucesivamente durante n períodos. La tasa de descuento es r por período.

El valor presente de esta renta es:

$$VP = \sum_{t=1}^n \frac{C_1 \cdot (1 + g)^{t-1}}{(1 + r)^t}$$

1. (1 punto) Escribe una función `valor_presente_renta(C1, g, r, n)` que:

- Reciba como parámetros: C_1 (primer pago), g (tasa de crecimiento), r (tasa de descuento), y n (número de períodos).
 - Calcule y retorne el valor presente de la renta usando un bucle `for` o una comprensión de lista.
 - Incluye validación: si $r \leq 0$ o $n \leq 0$, debe retornar `None` y mostrar un mensaje de error.
2. (0.5 puntos) Escribe el código para:
 - Llamar a la función con $C1=1000$, $g=0.02$, $r=0.05$, $n=10$.
 - Mostrar el resultado formateado con 2 decimales (ejemplo: “Valor presente: 8,863.25”).
 3. (0.5 puntos) Explica en texto:
 - Qué sucede con el valor presente si g es mayor que r y por qué.
 - Cómo cambiarías la función para calcular el valor presente de una renta perpetua (infinitos pagos) cuando $g < r$.

Ejercicio 2 – Pandas: rendimientos diarios y mensuales (2 puntos)

Supón que tienes la siguiente tabla de precios diarios de una acción (cierre ajustado):

fecha	close_adj
2025-01-02	100.0
2025-01-03	101.0
2025-01-06	100.5
2025-01-07	102.0
2025-01-08	101.0

1. (0.5 puntos) Escribe el código en Python para:

- Importar pandas.
- Crear un DataFrame llamado `df` con esos datos.
- Convertir la columna `fecha` a tipo fecha y dejarla como índice.

2. (0.5 puntos) Escribe el código para añadir una columna `ret_log` con el rendimiento logarítmico diario, definido como:

$$\text{ret_log}_t = \ln \left(\frac{\text{close_adj}_t}{\text{close_adj}_{t-1}} \right)$$

3. (0.5 puntos) Escribe una línea de código que calcule el rendimiento medio diario y lo guarde en una variable llamada `ret_medio_diario`.

4. (0.5 puntos) Explica en 2–3 frases:

- Cómo cambiarías el código para obtener rendimientos mensuales a partir de esta misma serie (indica qué método de resampling usarías y sobre qué frecuencia).
- Qué diferencia conceptual hay entre trabajar con precios y trabajar con rendimientos.

Ejercicio 3 – Series temporales con Darts (2 puntos)

Tienes una serie temporal mensual de ventas de una empresa almacenada en un CSV con las columnas:

- fecha (formato YYYY-MM-DD)
- ventas (en miles de €)

Quieres usar la librería `darts` para hacer predicciones con un modelo exponencial suavizado.

1. (0.5 puntos) Escribe el código para:

- Cargar el CSV `ventas.csv` en un DataFrame de pandas.
- Convertir el DataFrame a un objeto `TimeSeries` de darts, especificando la columna de tiempo y la columna de valores.

2. (0.5 puntos) Escribe el código para:

- Hacer un split temporal: entrenar con los primeros 36 meses y testear con los últimos 12 meses.
- Indica cómo harías este split usando métodos de `TimeSeries` de darts.

3. (0.5 puntos) Escribe el código para:

- Crear un modelo `ExponentialSmoothing` de darts.
- Entrenar el modelo con los datos de entrenamiento.
- Generar predicciones para los próximos 12 meses.

4. (0.5 puntos) Explica en texto:

- Qué ventaja tiene usar un split temporal (y no aleatorio) en series temporales.
- Qué métrica usarías para evaluar el modelo (por ejemplo, MAE, RMSE, MAPE) y por qué.

Ejercicio 4 – Regresión lineal múltiple (2 puntos)

Tienes un dataset de viviendas con las siguientes variables:

- precio (variable objetivo, en miles de €)
- superficie (m^2)
- habitaciones (número de habitaciones)
- antigüedad (años desde construcción)
- distancia_center (km al centro de la ciudad)

1. (0.5 puntos) Escribe el código para:

- Cargar el CSV `viviendas.csv` en un DataFrame `df`.
- Separar las variables explicativas (`X`) de la variable objetivo (`y`).
- Hacer un `train_test_split` (80 % train, 20 % test, con `random_state=42`).

2. (0.5 puntos) Escribe el código para:

- Entrenar un modelo de regresión lineal (`LinearRegression`) con los datos de entrenamiento.
- Obtener los coeficientes del modelo y el intercepto.

3. (0.5 puntos) Escribe el código para:

- Hacer predicciones sobre el conjunto de test.
- Calcular el R^2 (coeficiente de determinación) y el RMSE (Root Mean Squared Error) en el conjunto de test.

Indica qué funciones de `sklearn.metrics` utilizarías.

4. (0.5 puntos) Interpreta en texto:

- Qué significa un R^2 de 0.75 en este contexto.
- Si el coeficiente de `distancia_center` fuera -5.2, ¿qué implicaría esto sobre la relación entre la distancia al centro y el precio?

Ejercicio 5 – Clasificación y métricas (árbol de decisión) (2 puntos)

Dispones de un dataset macroeconómico mensual con:

- inflacion (inflación interanual, %)
- crecimiento_PIB (crecimiento intertrimestral anualizado, %)
- tasa_paro (%)
- recesion (0 = no recesión, 1 = recesión)

Quieres entrenar un modelo de clasificación que prediga si hay recesión (recesion) a partir de las otras variables.

1. **(0.5 puntos)** Escribe el código necesario para:

- Cargar el CSV `macro.csv` en un DataFrame `df`.
- Definir `X` (variables explicativas) y `y` (variable objetivo).

2. **(0.5 puntos)** Escribe el código para:

- Hacer un `train_test_split` (70 % train, 30 % test).
- Entrenar un `DecisionTreeClassifier` con una profundidad máxima (`max_depth`) de 3.

3. **(0.5 puntos)** Escribe el código para calcular en el conjunto de test:

- La **accuracy** del modelo.
- La **matriz de confusión**.

Indica qué funciones de `sklearn.metrics` utilizarías.

4. **(0.5 puntos)** Explica en texto:

- Qué significa exactamente la **accuracy** en este problema.
- Por qué podría ser engañosa si la proporción de meses con recesión es muy pequeña en comparación con los meses sin recesión.