Respuesta Exámen Code Rut Lay Abad

Parte 1

1. ¿Qué es un DataFrame de pandas y para qué se utiliza?

Un DataFrame es una estructura de datos bidimensional de la librería pandas que permite almacenar y manipular datos en forma de tabla, similar a una hoja de cálculo. Se utiliza para análisis de datos, transformación, limpieza y visualización.

2. ¿Cuál es la diferencia entre np.nan y None en Python?

None es un tipo de Python que representa la ausencia de valor o nulo en estructuras genéricas.

np.nan es un valor especial de NumPy que representa un número faltante (not a number) en contextos numéricos. np.nan permite operaciones matemáticas y estadísticas, mientras que None no.

3. ¿Qué significa una p-value menor que 0.05 en una prueba de hipótesis?

Indica que hay evidencia estadísticamente significativa para rechazar la hipótesis nula al 5% de nivel de significancia. Es decir, el resultado observado es poco probable que ocurra por azar.

4. Explica qué es un modelo de regresión logística y cuándo se utiliza.

Un modelo de regresión logística es un modelo estadístico que estima la probabilidad de ocurrencia de un evento binario (0 o 1) en función de variables predictoras. Se utiliza para clasificar observaciones y calcular el efecto (odds ratio) de las variables independientes sobre un resultado binario.

5. ¿Qué diferencia hay entre la prueba de Chi-cuadrado y la prueba exacta de Fisher?

La prueba de Chi-cuadrado se usa con tablas grandes y frecuencias esperadas altas.

La prueba exacta de Fisher se usa cuando los tamaños de muestra son pequeños y las frecuencias esperadas son bajas, ya que es más precisa en esos casos.

6. ¿Qué representa el Odds Ratio en un modelo de regresión logística?

Representa cuánto se incrementa o reduce la probabilidad del evento cuando cambia la variable independiente, manteniendo las demás constantes. Un OR > 1 indica mayor probabilidad (mayor riesgo o asociación positiva).; OR < 1, disminuye la probabilidad del evento (menor riesgo o asociación negativa).

7. ¿Qué es y para qué sirve un intervalo de confianza en las pruebas estadísticas?

Es un rango de valores dentro del cual se espera que se encuentre el valor real de un parámetro (como el OR) con una determinada probabilidad (por ejemplo, 95%). Sirve para expresar la incertidumbre de una estimación.

8. ¿Qué mide el pseudo R^2 de McFadden en un modelo de regresión logística?

Mide la calidad del ajuste del modelo. Es una analogía del R^2 de la regresión lineal, donde valores más altos indican mejor ajuste, aunque sus valores suelen ser más bajos (por ejemplo, 0.2 ya se considera aceptable).

Parte 2

9. ¿Qué significa el argumento dropna() en esta línea?

groups = df[group_col].dropna().unique()

dropna() elimina los valores nulos (NaN) solo de la columna group_col. unique() devuelve los valores distintos restantes.

Se usa para obtener solo los grupos válidos sin datos faltantes.

10. ¿Qué se está haciendo y por qué?

if 3 <= len(df[df[group_col]==g]) <= 5000: p = shapiro(df.loc[df[group_col]==g, col])[1] else: p = 1.0

El test de Shapiro-Wilk que se emplea para comrobar si una distribución sigue un comportamiento paramétrico Este solo es válido para tamaños de muestra entre 3 y 5000. Fuera de ese rango, los resultados pueden ser incorrectos o la prueba no se puede aplicar por lo tanto: en estas líneas de código se evalúa si el número de muestras en el grupo g está entre 3 y 5000. Si cumple, se aplica el test de Shapiro para verificar si los datos tienen distribución normal. Como el test de Shapiro-Wilk no es válido para más de 5000 observaciones, se asigna p = 1.0 para asumir normalidad por el Teorema Central del Límite y seguir con pruebas paramétricas como el t-test. Esto evita errores por tamaños de muestra inválidos para el test de Shapiro.

11. Completa esta línea con el valor-p del t-test:

test_name, p_val = ('t-test', stats.ttest_ind(df[df[group_col]==groups[0]][col], df[df[group_col]==groups[1]][col], equal_var=False).pvalue)

Se usa stats.ttest_ind con equal_var=False (Welch's t-test, que no asume igual varianza).

12. Explica qué representa esta fórmula en statsmodels:

formula = "Diagnostico_bin ~ Edad + C(Sexo) + GlucosaValor"

Esta es s una fórmula de regresión logística. donde:

Diagnostico_bin es la variable dependiente (0 o 1).

Las variables independientes son:

Edad: numérica.

C(Sexo): categórica (se codifica internamente con variables dummies).

GlucosaValor: numérica.

El modelo estima el efecto de cada variable sobre la probabilidad de diagnóstico.

13. ¿Qué hace este bloque y por qué se usa try-except?

try: model = smf.logit(formula, data=df_test).fit(disp=0) except (LinAlgError, ValueError) as e: print(f"Error: {e}") continue

Este bloque intenta ajustar un modelo de regresión logística. Se usa try-except para capturar errores, como: LinAlgError, ValueError que representan datos inválidos o mal codificados. Así el programa no se detiene si falla un modelo, y continúa con las demás variables.

Parte 4

19. ¿Qué limitaciones o riesgos tiene este análisis estadístico para evaluar la relación entre variables y un diagnóstico binario?

Algunas de las principales limitaciones y riesgos son:

Aunque una variable tenga un p-value bajo u odds ratio alto, no significa que cause el diagnóstico. Puede haber variables ocultas (confusores).

Si varias variables independientes están correlacionadas entre sí, pueden distorsionar los resultados del modelo logístico.

Si no se manejan bien los valores NaN, se pueden excluir muchos casos válidos o sesgar los resultados.

El uso de t-tests o regresión logística requiere que ciertos supuestos se cumplan (como independencia o distribución adecuada) y esto puede arrojar valores que estén alejados de la realidad.

Por otro lado existe el riesgo de que variables verdaderamente influyentes no se hayan medido o no se hayan incluido en el análisis, lo que puede ocultar relaciones importantes y producir resultados parciales o erróneos.

```
20. Si un alumno quisiera automatizar este flujo de análisis
para múltiples archivos CSV, ¿qué cambios debería hacer al
código?
```

Se debería usar un bucle para procesar varios archivos, por ejemplo:

import os

for file in os.listdir("Data"):

```
  if file.endswith(".csv"):

     df = pd.read_csv(os.path.join("Data", file),
encoding='latin-1')

     # ... aplicar flujo completo
```

Crear funciones modulares:

Una función para preprocesar los datos (crear Diagnostico_bin, eliminar columnas inválidas).

Una función para aplicar summarize_and_test.

Una función para entrenar el modelo logístico y graficar los resultados.

Guardar los resultados automáticamente, por ejemplo, como archivos .csv o imágenes .png.

Permitir configuraciones externas, como una lista de variables dependientes o columnas a ignorar, usando un archivo .json o .yaml.

Parte 5

```
21. Cambiar la variable dependiente para que incluya
Musculoesqueleticos y Diagnostico
```

Para esto se creó una nueva variable dependiente llamada Diagnostico_Musculo_bin que será 1 si hay diagnóstico musculoesquelético o si hay algún diagnóstico diferente de “Ninguno”, y 0 en otro caso.

```
filename = 'data_basal_musculoesqueleticos.csv'
folder = 'Data'
df_test = pd.read_csv(f"{folder}/{filename}", encoding='latin-1')
# Nueva variable dependiente combinada solucion a parte 
df_test['Diagnostico_Musculo_bin'] = np.where(
    (df_test['Musculoesqueleticos'] == 1) | (df_test['Diagnostico'] != 'Ninguno'),
    1, 0)
#df_test['Diagnostico_bin'] = np.where(df_test['Diagnostico'] != 'Ninguno', 1, 0)
#R: Diagnostico_bin = Musculoesqueleticos or Diagnostico_bin
#df_test = df_test.drop(columns=['Diagnostico'])
#summary_table = summarize_and_test(df_test, group_col='Diagnostico_bin')
summary_table = summarize_and_test(df_test, group_col='Diagnostico_Musculo_bin')
```

Para comprobar si funcionaba se imprimió una tabla con datos del 15 al 40 con las columnas diagnóstico, Musculoesquelético y la nueva variable creada

```
# Visualizar los cambios en una tabla (filas 15 a 40)
tabla_cambios = df_test[['Diagnostico', 'Musculoesqueleticos', 'Diagnostico_Musculo_bin']].copy()
print(tabla_cambios.iloc[15:41])
```

✓ 0.0s

	Diagnostico	Musculoesqueleticos	Diagnostico_Musculo_bin
15	Ninguno	Ninguno	0
16	Ninguno	Ninguno	0
17	OA	SDR	1
18	Ninguno	Ninguno	0
19	Ninguno	Ninguno	0
20	Ninguno	Ninguno	0
21	Ninguno	Ninguno	0
22	OA	Ninguno	1
23	Ninguno	Ninguno	0
24	Ninguno	Ninguno	0
25	Ninguno	SDR	0
26	Ninguno	Ninguno	0
27	Ninguno	Ninguno	0
28	Ninguno	Ninguno	0
29	Ninguno	Ninguno	0
30	Ninguno	SDR	0
31	Ninguno	Ninguno	0
32	Ninguno	Ninguno	0
33	Ninguno	Ninguno	0
34	Ninguno	SDR	0
35	Ninguno	SDR	0
36	AR	Ninguno	1
37	Ninguno	Ninguno	0
38	Ninguno	Ninguno	0
39	Ninguno	Ninguno	0
40	Ninguno	Ninguno	0

22. Cambiar la combinación de variables independientes para encontrar el mejor OR

Para ello se modificaron las variables contenidas en kept_vars y se observó el comportamiento del modelo de regresión logistica para predecir la probabilidad de presentar un diagnóstico musculoesquelético o cualquier otro diagnóstico. Este reveló una

asociación estadísticamente significativa con cuatro variables clínicas clave: historial de dolor, funcionalidad física (HAQ), edad y presencia de comorbilidades.

```
[35]  sig_vars
      ✓ 0.0s

...   array(['Edad', 'Estudios', 'HijosCuantos', 'PersonasViven', 'Presion_Sys',
             'GlucosaValor', 'HorasTrabajas', 'DolorSieteDias_ocaciones',
             'DolorHistoirico_ocaciones', 'Enfermedades_numero_medicamentos',
             'Enfermedades_costo_medicamentos', 'DejaHacer', 'HAQval',
             'Eurocol', 'vivienda_tienen_3', 'vivienda_tienen_9',
             'Comorbilidad_2', 'Comorbilidad_6', 'Comorbilidad_14',
             'DolorSieteDias_mediananivel', 'DolorHistoirico_mediananivel',
             'Musculoesqueleticos', 'HablasEspanolId', 'EscribesEspanolId',
             'ParedesId', 'ElectricidadId', 'ReumaFamiliaId',
             'TenidoDolorSieteDiasId', 'TenidoDolorHistoiricoId',
             'MedicamentoDolorId', 'HAQvalbin', 'TipoTransporte', 'MovilidadId',
             'CuidadoPersonalId', 'ActividadesDiariasId',
             'DolorCalidaddeVidaId', 'EstadoSaludId'], dtype=object)

      drop_vars = ['ElectricidadId', 'Musculoesqueleticos', 'EstadoSaludId'] #eliminando columnas que se que estan mal

      kept_vars = [var for var in sig_vars if var not in drop_vars]

      kept_vars = ['TenidoDolorHistoiricoId', 'HAQval','Edad', 'Comorbilidad_14']#eliminando todas menos las que mostraron mejor relacion
[110] ✓ 0.0s
```

Entre los predictores, la comorbilidad 14 mostró el efecto más fuerte, con un odds ratio (OR) de 11.89 (IC 95%: 4.71–30.03), lo que indica que los pacientes con esta condición tienen casi 12 veces más probabilidad de recibir un diagnóstico. Asimismo, quienes reportaron historial de dolor tuvieron un riesgo más de tres veces mayor (OR = 3.25; IC 95%: 1.75–6.03) respecto a quienes no lo reportaron.

El índice HAQ, que evalúa la discapacidad funcional, también se asoció significativamente con el diagnóstico: por cada unidad de aumento en HAQ, la probabilidad de diagnóstico se multiplicó por 2.15 (IC 95%: 1.36–3.39). Finalmente, la edad presentó un efecto moderado pero significativo, con un OR de 1.03 (IC 95%: 1.015–1.052), lo que implica un incremento del 3% en el riesgo por cada año adicional de vida.

Las métricas de ajuste del modelo (McFadden's $R^2 = 0.2591$, Nagelkerke's $R^2 = 0.3381$) indican un buen poder explicativo, apropiado para estudios clínicos observacionales.

En conjunto, estos resultados permiten identificar factores relevantes que podrían utilizarse para orientar estrategias de detección temprana o intervención preventiva en pacientes con riesgo de padecimientos musculoesqueléticos u otros diagnósticos relacionados.

```
Fitting model with formula: Diagnostico_Musculo_bin ~ C(TenidoDolorHistoiricoId) + HAQval + Edad + Comorbilidad_14
McFadden's R2: 0.2591
Cox & Snell's R2: 0.1835
Nagelkerke's R2: 0.3381
                  variable     level                          method  \
0  TenidoDolorHistoiricoId       SÃ  Multivariable Logistic regression
1                   HAQval  per unit  Multivariable Logistic regression
2                     Edad  per unit  Multivariable Logistic regression
3          Comorbilidad_14  per unit  Multivariable Logistic regression

   odds_ratio  ci_lower   ci_upper       p_value
0    3.247768  1.750677   6.025097  1.869031e-04
1    2.151255  1.363374   3.394447  9.948746e-04
2    1.033102  1.014820   1.051713  3.503577e-04
3   11.893563  4.710612  30.029396  1.608727e-07
```

R2 originales con todas las significativas Fitting model with formula:

Diagnostico_bin ~ Edad + Estudios + HijosCuantos + PersonasViven + Presion_Sys + GlucosaValor + HorasTrabajas + DolorSieteDias_ocaciones + DolorHistoirico_ocaciones + Enfermedades_numero_medicamentos + Enfermedades_costo_medicamentos + DejaHacer + HAQval + Eurocol + vivienda_tienen_3 + vivienda_tienen_9 + Comorbilidad_2 + Comorbilidad_6 + Comorbilidad_14 + DolorSieteDias_mediananivel + DolorHistoirico_mediananivel + C(Musculoesqueleticos) + C(HablasEspanolId) + C(EscribesEspanolId) + C(ParedesId) + C(ElectricidadId) + C(ReumaFamiliaId) + C(TenidoDolorSieteDiasId) + C(TenidoDolorHistoiricoId) + C(MedicamentoDolorId) + C(HAQvalbin) + C(TipoTransporte) + C(MovilidadId) + C(CuidadoPersonalId) + C(ActividadesDiariasId) + C(DolorCalidaddeVidaId) + C(EstadoSaludId)

McFadden's R2: 0.4652 Cox & Snell's R2: 0.3051 Nagelkerke's R2: 0.5622

Aunque el modelo completo logra un mayor poder explicativo, su complejidad y riesgo de sobreajuste lo hacen menos práctico para implementación clínica. El modelo reducido, con solo cuatro variables significativas y métricas de ajuste adecuadas, representa una alternativa eficiente, interpretable y clínicamente relevante para identificar pacientes con riesgo de diagnóstico musculoesquelético o general.

El gráfico de OR quedó de la siguiente manera: