Examen de Aprendizaje Supervisado

Instrucciones: Utiliza los archivos `hospital_data.csv` y `students_data.csv` para responder las siguientes preguntas. Escribe tu código en las celdas correspondientes.

1. Cargar los datos

Carga ambos archivos CSV en dos DataFrames de pandas.

```
import pandas as pd
# Cargar los datos
df_hospital = pd.read_csv('hospital_data.csv')
df_students = pd.read_csv('students_data.csv')
```

2. Exploración inicial

Muestra las primeras 5 filas y las estadísticas descriptivas de cada conjunto de datos.

```
# Exploración inicial hospital
display(df_hospital.head())
display(df_hospital.describe(include='all'))

# Exploración inicial students
display(df_students.head())
display(df_students.describe(include='all'))
```

{"columns":[{"name":"index","rawType":"int64","type":"integer"},
{"name":"Age","rawType":"int64","type":"integer"},
{"name":"Blood_Pressure","rawType":"int64","type":"integer"},
{"name":"Cholesterol","rawType":"int64","type":"integer"},
{"name":"Heart_Rate","rawType":"int64","type":"integer"},
{"name":"Gender","rawType":"object","type":"string"},
{"name":"Diagnosis","rawType":"object","type":"string"},
{"name":"Smoker","rawType":"object","type":"string"},
{"name":"Exercise","rawType":"object","type":"unknown"},
{"name":"Risk_Level","rawType":"object","type":"string"}],"ref":"3d931ab8-67bb-4fe7-b2e5-e0fedd0d26d4","rows":
[["0","51","113","225","96","Male","Diabetes","No",null,"High Risk"],
["1","92","87","270","63","Female","Healthy","No","Occasional","High Risk"],["2","14","119","195","89","Male","Heart Disease","No","Occasional","Low Risk"],
["3","71","162","229","65","Male","Heart Disease","Yes",null,"High Risk"],["4","60","121","203","78","Female","Heart Disease","Yes","Occasional","High Risk"]],"shape":
{"columns":9,"rows":5}}

```
{"columns":[{"name":"index","rawType":"object","type":"string"},
{"name":"Age","rawType":"float64","type":"float"},
{"name":"Blood_Pressure","rawType":"float64","type":"float"},
{"name":"Cholesterol","rawType":"float64","type":"float"},
{"name":"Heart_Rate","rawType":"float64","type":"float"},
{"name":"Gender","rawType":"object","type":"unknown"},
{"name":"Diagnosis","rawType":"object","type":"unknown"},
{"name":"Smoker","rawType":"object","type":"unknown"},
{"name":"Exercise","rawType":"object","type":"unknown"},
{"name":"Risk_Level","rawType":"object","type":"unknown"}],"ref":"e1bd
e441-532b-4b5f-a343-4321489e7514","rows":
[["count","1000.0","1000.0","1000.0","1000.0","1000","1000","1000","67
7","1000"],["unique",null,null,null,null,"2","4","2","2","2"],
["top",null,null,null,null,"Male","Heart
Disease","No","Occasional","High Risk"],
["freq",null,null,null,null,"518","269","505","350","503"],
["mean","49.128","128.948","226.64","79.928",null,null,null,null,null]
,
["std","29.573505172843618","29.133040178292926","44.69606401170172","
11.492819536602012",null,null,null,null,null],
["min","0.0","80.0","150.0","60.0",null,null,null,null,null],
["25%","23.0","104.0","188.0","70.0",null,null,null,null,null],
["50%","50.0","128.0","228.0","80.0",null,null,null,null,null],
["75%","74.0","154.0","267.0","90.0",null,null,null,null,null],
["max","99.0","179.0","299.0","99.0",null,null,null,null,null]],"shape
":{"columns":9,"rows":11}}

{"columns":[{"name":"index","rawType":"int64","type":"integer"},
{"name":"GPA","rawType":"float64","type":"float"},
{"name":"Attendance","rawType":"int64","type":"integer"},
{"name":"Study_Hours","rawType":"int64","type":"integer"},
{"name":"Projects_Completed","rawType":"int64","type":"integer"},
{"name":"Major","rawType":"object","type":"string"},
{"name":"Year","rawType":"object","type":"string"},
{"name":"Scholarship","rawType":"object","type":"string"},
{"name":"Extracurricular","rawType":"object","type":"unknown"},
{"name":"Result","rawType":"object","type":"string"}],"ref":"3677241b-
7e88-40eb-a401-bfa9780dc4d9","rows":
[["0","3.52","66","37","0","Business","Junior","Yes","Music","Fail"],
["1","2.06","70","13","9","Business","Senior","Yes",null,"Fail"],
["2","2.16","95","18","7","Engineering","Senior","No",null,"Pass"],
["3","1.02","65","29","8","Science","Freshman","No","Music","Pass"],
["4","3.07","95","4","6","Business","Sophomore","No","Sports","Fail"]]
,"shape":{"columns":9,"rows":5}}

{"columns":[{"name":"index","rawType":"object","type":"string"},
{"name":"GPA","rawType":"float64","type":"float"},
{"name":"Attendance","rawType":"float64","type":"float"},
{"name":"Study_Hours","rawType":"float64","type":"float"},
{"name":"Projects_Completed","rawType":"float64","type":"float"},
```

```
{"name":"Major","rawType":"object","type":"unknown"},
{"name":"Year","rawType":"object","type":"unknown"},
{"name":"Scholarship","rawType":"object","type":"unknown"},
{"name":"Extracurricular","rawType":"object","type":"unknown"},
{"name":"Result","rawType":"object","type":"unknown"}],"ref":"b4619ab3
-5c72-4671-932c-5a88962f4488","rows":
[["count","1000.0","1000.0","1000.0","1000.0","1000","1000","1000","74
0","1000"],["unique",null,null,null,null,"4","4","2","3","2"],
["top",null,null,null,null,"Business","Sophomore","No","Music","Fail"]
,["freq",null,null,null,null,"271","258","510","269","508"],
["mean","1.98941","73.735","19.248","4.674",null,null,null,null,null],
["std","1.1707044463914882","14.298994238625184","11.400722453970735",
"2.8554995846397664",null,null,null,null,null],
["min","0.0","50.0","0.0","0.0",null,null,null,null,null],
["25%","0.95","62.0","10.0","2.0",null,null,null,null,null],
["50%","2.01","73.0","19.0","5.0",null,null,null,null,null],
["75%","2.94","86.0","29.0","7.0",null,null,null,null,null],
["max","4.0","99.0","39.0","9.0",null,null,null,null,null]],"shape":
{"columns":9,"rows":11}}
```

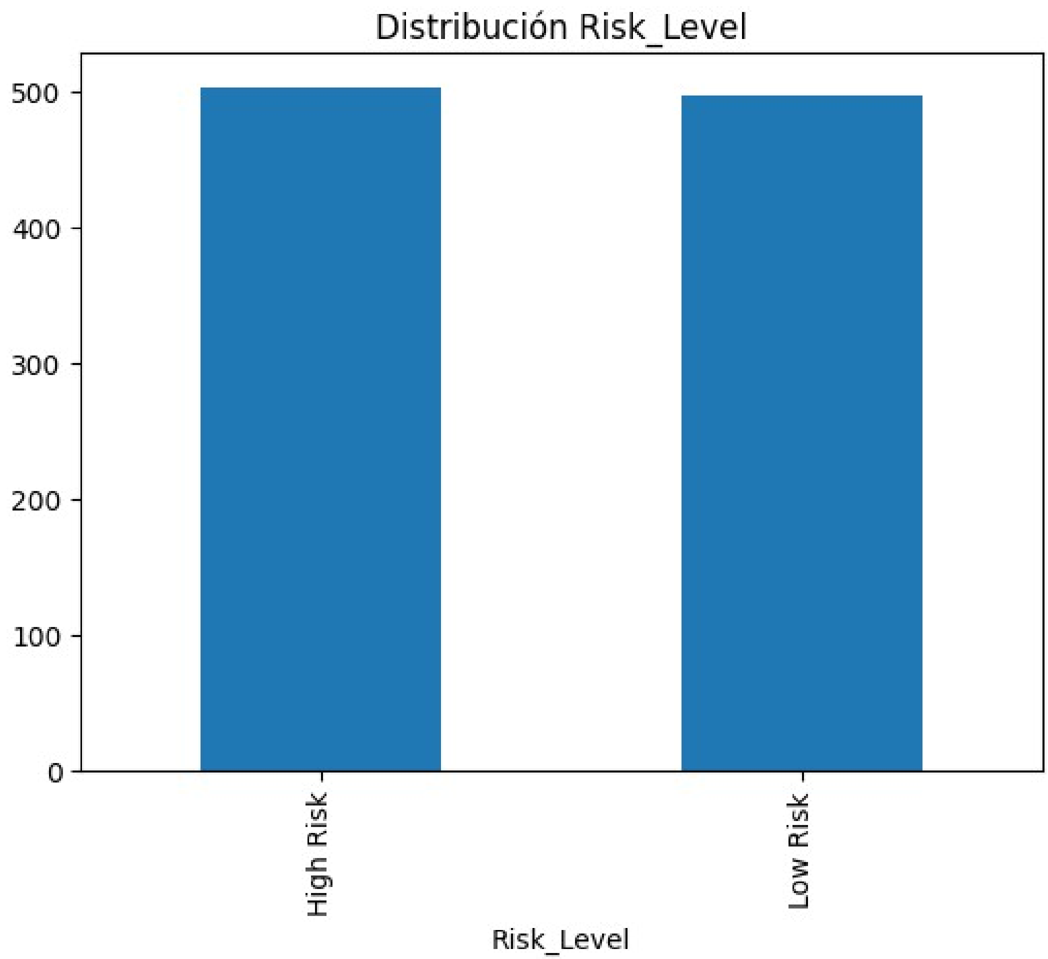

3. Análisis de la variable objetivo

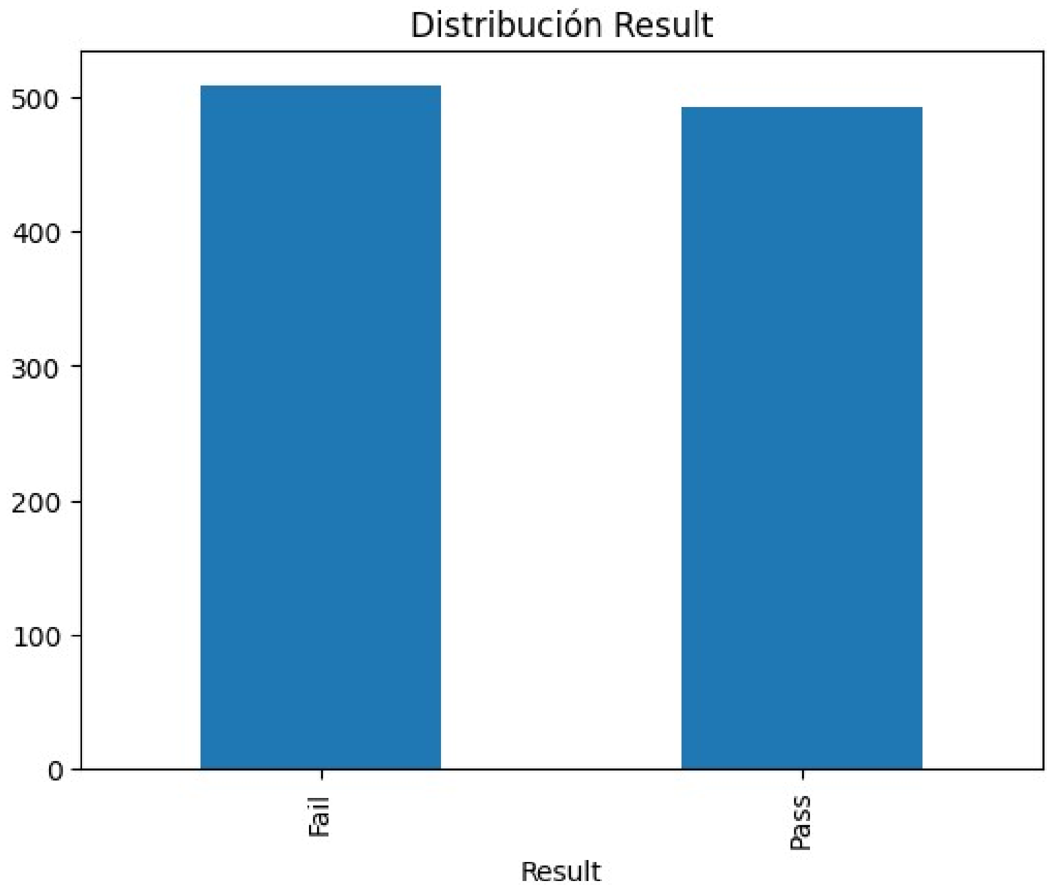
Muestra la distribución de la variable Risk_Level en hospitaldf y Result en studentsdf.

```
import matplotlib.pyplot as plt

# Distribución Risk_Level
df_hospital['Risk_Level'].value_counts().plot(kind='bar',
title='Distribución Risk_Level')
plt.show()

# Distribución Result
df_students['Result'].value_counts().plot(kind='bar',
title='Distribución Result')
plt.show()
```


Distribución Risk_Level
500
400
300
200
100
0
High Risk
Low Risk
Risk_Level



4. Preprocesamiento

- Codifica las variables categóricas.
- Normaliza las variables numéricas si es necesario.

```
from sklearn.preprocessing import LabelEncoder, StandardScaler

# Copias para preprocesar
df_hosp = df_hospital.copy()
df_stud = df_students.copy()

# Codificar categóricas
def encode_categoricals(df, target):
    for col in df.select_dtypes(include='object').columns:
        if col != target:
            df[col] = LabelEncoder().fit_transform(df[col])
    return df

df_hosp = encode_categoricals(df_hosp, 'Risk_Level')
df_stud = encode_categoricals(df_stud, 'Result')

# Codificar variable objetivo
df_hosp['Risk_Level'] =
```

```
LabelEncoder().fit_transform(df_hosp['Risk_Level'])
df_stud['Result'] = LabelEncoder().fit_transform(df_stud['Result'])

# Normalizar numéricas
def normalize(df, exclude):
    scaler = StandardScaler()
    num_cols = df.select_dtypes(include=['int64',
'float64']).columns.difference([exclude])
    df[num_cols] = scaler.fit_transform(df[num_cols])
    return df

df_hosp = normalize(df_hosp, 'Risk_Level')
df_stud = normalize(df_stud, 'Result')
```

5. División de datos

Divide los datos en conjuntos de entrenamiento y prueba (80/20).

```
from sklearn.model_selection import train_test_split

# Hospital
y_hosp = df_hosp['Risk_Level']
X_hosp = df_hosp.drop('Risk_Level', axis=1)
Xh_train, Xh_test, yh_train, yh_test = train_test_split(X_hosp,
y_hosp, test_size=0.2, random_state=42)

# Students
y_stud = df_stud['Result']
X_stud = df_stud.drop('Result', axis=1)
Xs_train, Xs_test, ys_train, ys_test = train_test_split(X_stud,
y_stud, test_size=0.2, random_state=42)
```

6. Entrenamiento de modelos

Entrena al menos dos modelos de clasificación (por ejemplo, Regresión Logística y Árbol de Decisión) para cada conjunto de datos.

```
from sklearn.linear_model import LogisticRegression
from sklearn.tree import DecisionTreeClassifier

# Hospital
log_hosp = LogisticRegression(max_iter=1000)
log_hosp.fit(Xh_train, yh_train)
dt_hosp = DecisionTreeClassifier(random_state=42)
dt_hosp.fit(Xh_train, yh_train)

# Students
log_stud = LogisticRegression(max_iter=1000)
```

```
log_stud.fit(Xs_train, ys_train)
dt_stud = DecisionTreeClassifier(random_state=42)
dt_stud.fit(Xs_train, ys_train)

DecisionTreeClassifier(random_state=42)
```

7. Evaluación de modelos

Evalúa los modelos usando métricas como precisión, recall, F1-score y matriz de confusión.

```
from sklearn.metrics import accuracy_score, precision_score,
recall_score, f1_score, confusion_matrix, classification_report

def eval_model(model, X, y, nombre):
    y_pred = model.predict(X)
    print(f"\nEvaluación para {nombre}:")
    print("Accuracy:", accuracy_score(y, y_pred))
    print("Precision:", precision_score(y, y_pred,
average='weighted'))
    print("Recall:", recall_score(y, y_pred, average='weighted'))
    print("F1-score:", f1_score(y, y_pred, average='weighted'))
    print("Matriz de confusión:\n", confusion_matrix(y, y_pred))
    print(classification_report(y, y_pred))

print("--- Modelos Hospital Data ---")
eval_model(log_hosp, Xh_test, yh_test, 'Logistic Regression
(Hospital)')
eval_model(dt_hosp, Xh_test, yh_test, 'Decision Tree (Hospital)')

print("\n--- Modelos Students Data ---")
eval_model(log_stud, Xs_test, ys_test, 'Logistic Regression
(Students)')
eval_model(dt_stud, Xs_test, ys_test, 'Decision Tree (Students)')

--- Modelos Hospital Data ---

Evaluación para Logistic Regression (Hospital):
Accuracy: 0.47
Precision: 0.48214562192697985
Recall: 0.47
F1-score: 0.4652127659574468
Matriz de confusión:
 [[53 38]
 [68 41]]
              precision    recall  f1-score   support

           0       0.44      0.58      0.50        91
           1       0.52      0.38      0.44       109

    accuracy                           0.47       200
```

```
   macro avg       0.48      0.48      0.47       200
weighted avg       0.48      0.47      0.47       200


Evaluación para Decision Tree (Hospital):
Accuracy: 0.515
Precision: 0.5154408212560386
Recall: 0.515
F1-score: 0.5152076250912846
Matriz de confusión:
 [[43 48]
 [49 60]]
              precision    recall  f1-score   support

           0       0.47      0.47      0.47        91
           1       0.56      0.55      0.55       109

    accuracy                           0.52       200
   macro avg       0.51      0.51      0.51       200
weighted avg       0.52      0.52      0.52       200


--- Modelos Students Data ---

Evaluación para Logistic Regression (Students):
Accuracy: 0.475
Precision: 0.47225112199102415
Recall: 0.475
F1-score: 0.4730491165095174
Matriz de confusión:
 [[58 49]
 [56 37]]
              precision    recall  f1-score   support

           0       0.51      0.54      0.52       107
           1       0.43      0.40      0.41        93

    accuracy                           0.47       200
   macro avg       0.47      0.47      0.47       200
weighted avg       0.47      0.47      0.47       200


Evaluación para Decision Tree (Students):
Accuracy: 0.55
Precision: 0.5506516290726817
Recall: 0.55
F1-score: 0.5502709755118427
Matriz de confusión:
 [[61 46]
 [44 49]]
```

```
              precision    recall  f1-score   support

           0       0.58      0.57      0.58       107
           1       0.52      0.53      0.52        93

    accuracy                           0.55       200
   macro avg       0.55      0.55      0.55       200
weighted avg       0.55      0.55      0.55       200
```

8. Comparación de modelos

Compara el rendimiento de los modelos y justifica cuál elegirías para cada conjunto de datos.

Análisis de resultados para Hospital Data

Regresión Logística:

- Accuracy: 0.47
- Precision: 0.48
- Recall: 0.47
- F1-score: 0.47
- La matriz de confusión muestra que el modelo tiene dificultades para distinguir entre las clases, con un desempeño apenas superior al azar. El recall para la clase 1 (positiva) es bajo (0.38), lo que indica que muchos casos positivos no son detectados.

Árbol de Decisión:

- Accuracy: 0.52
- Precision: 0.52
- Recall: 0.52
- F1-score: 0.52
- El árbol de decisión mejora ligeramente el desempeño respecto a la regresión logística, pero aún así el modelo no logra una buena discriminación entre clases. El accuracy y las demás métricas apenas superan el 50%, lo que sugiere que los datos pueden ser complejos o que se requiere mayor preprocesamiento o ajuste de hiperparámetros.

Conclusión:

- Para el conjunto de datos hospital, el árbol de decisión es preferible sobre la regresión logística, aunque ambos modelos muestran un desempeño limitado. Se recomienda explorar más el preprocesamiento, ingeniería de variables o probar otros modelos para mejorar los resultados.