Asignatura	Datos del alumno	Fecha
Aprendizaje Automático y Minería de Datos	Estudiante: Oscar David Bocanegra	03/17/2025
	Estudiante: Laura Marcela Barona	

Trabajo: Modelo sencillo mediante árboles de clasificación en Python

Objetivos

El objetivo principal de esta actividad es que el alumno aplique uno o varios algoritmos de clasificación para predecir la variable respuesta en el conjunto de datos objetivo, evaluar dicho algoritmo e interpretar los resultados.

Descripción de la actividad y pautas de elaboración

Los pasos a seguir para realizar la actividad son los siguientes (pasos orientativos):

- Análisis descriptivo de los datos.
- Determinar el conjunto de modelización y el de validación.
- Tratamiento de *missing* (si lo hay).
- Realizar los análisis previos que consideres oportunos.
- Dividir los datos entre conjunto de modelación y conjunto test.
- Aplicar uno o varios algoritmos para predecir la variable respuesta.
- Comentar los resultados.

La variable respuesta es «Chance of admit» y se debe considerar como «yes» si es ≥0,6 y «no» en caso contrario. El enlace que contiene los datos es el siguiente:

Accede al enlace a través del aula virtual o desde la siguiente dirección:

https://www.kaggle.com/mohansacharya/graduate-admissions

Asignatura	Datos del alumno	Fecha
Aprendizaje Automático y Minería de Datos	Estudiante: Oscar David Bocanegra	03/17/2025
	Estudiante: Laura Marcela Barona	

Criterios de evaluación

- La evaluación y la entrega de actividades se realizará de forma individual.
- Se valorarán especialmente los comentarios sobre la interpretación práctica de los resultados.
- Se puede utilizar Python.
- Se deben comentar los resultados obtenidos y el código.

Extensión máxima: 3 páginas, fuente Calibri 12 e interlineado 1,5.

Asignatura	Datos del alumno	Fecha
Aprendizaje Automático y Minería de Datos	Estudiante: Oscar David Bocanegra	03/17/2025
	Estudiante: Laura Marcela Barona	

Análisis descriptivo de los datos.

El primer paso para desarrollar para esta actividad va a ser la debida importación de las librerías que vamos a usar para este proyecto las cuales son las siguientes

```
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt
from sklearn.model_selection import train_test_split
from sklearn.tree import DecisionTreeClassifier
from sklearn.ensemble import RandomForestClassifier
from sklearn.metrics import classification_report, confusion_matrix, ConfusionMatrixDisplay
```

Después de esto lo que realizamos fue renombrar las columnas para un uso más fácil a la hora de manipular dichos datos, eliminar la columna del identificados del data set recibido ya que este no tiene gran relevancia para esta actividad, e imprimir las primeras filas de este data set.

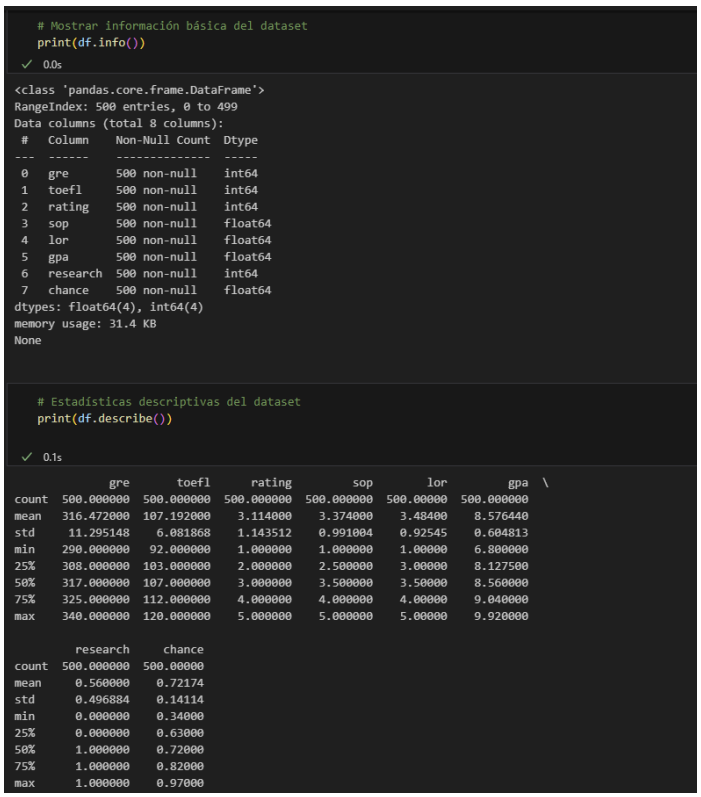
```
# Cargar el conjunto de datos
df = pd.read_csv('./file/Admission_Predict_Ver1.1.csv')

# Renombrar las columnas para facilitar su uso
df = df.rename(columns={'Serial No.': 'no',
                        'GRE Score': 'gre',
                        'TOEFL Score': 'toefl',
                        'University Rating': 'rating',
                        'SOP': 'sop',
                        'LOR ': 'lor',
                        'CGPA': 'gpa',
                        'Research': 'research',
                        'Chance of Admit ': 'chance'})

# Eliminar la columna 'no' ya que es solo un identificad
df.drop(['no'], axis=1, inplace=True)

# Mostrar las primeras filas del dataset para revisión
print(df.head())
```

✓ 0.1s

```
   gre  toefl  rating  sop  lor   gpa  research  chance
0  337    118       4  4.5  4.5  9.65         1    0.92
1  324    107       4  4.0  4.5  8.87         1    0.76
2  316    104       3  3.0  3.5  8.00         1    0.72
3  322    110       3  3.5  2.5  8.67         1    0.80
4  314    103       2  2.0  3.0  8.21         0    0.65
```

Para que luego de esto poder identificar el tipo de data que estaremos usando, obteniendo la siguiente información.

Asignatura	Datos del alumno	Fecha
Aprendizaje Automático y Minería de Datos	Estudiante: Oscar David Bocanegra	03/17/2025
	Estudiante: Laura Marcela Barona	



```
# Mostrar información básica del dataset
print(df.info())
```

✓ 0.0s

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 500 entries, 0 to 499
Data columns (total 8 columns):
 #   Column    Non-Null Count  Dtype
---  ------    --------------  -----
 0   gre       500 non-null    int64
 1   toefl     500 non-null    int64
 2   rating    500 non-null    int64
 3   sop       500 non-null    float64
 4   lor       500 non-null    float64
 5   gpa       500 non-null    float64
 6   research  500 non-null    int64
 7   chance    500 non-null    float64
dtypes: float64(4), int64(4)
memory usage: 31.4 KB
None
```

```
# Estadísticas descriptivas del dataset
print(df.describe())
```

✓ 0.1s

```
              gre       toefl      rating         sop         lor         gpa  \
count  500.000000  500.000000  500.000000  500.000000  500.00000  500.000000
mean   316.472000  107.192000    3.114000    3.374000    3.48400    8.576440
std     11.295148    6.081868    1.143512    0.991004    0.92545    0.604813
min    290.000000   92.000000    1.000000    1.000000    1.00000    6.800000
25%    308.000000  103.000000    2.000000    2.500000    3.00000    8.127500
50%    317.000000  107.000000    3.000000    3.500000    3.50000    8.560000
75%    325.000000  112.000000    4.000000    4.000000    4.00000    9.040000
max    340.000000  120.000000    5.000000    5.000000    5.00000    9.920000

         research     chance
count  500.000000  500.00000
mean     0.560000    0.72174
std      0.496884    0.14114
min      0.000000    0.34000
25%      0.000000    0.63000
50%      1.000000    0.72000
75%      1.000000    0.82000
max      1.000000    0.97000
```

En donde podemos concluir que la data que manejamos es

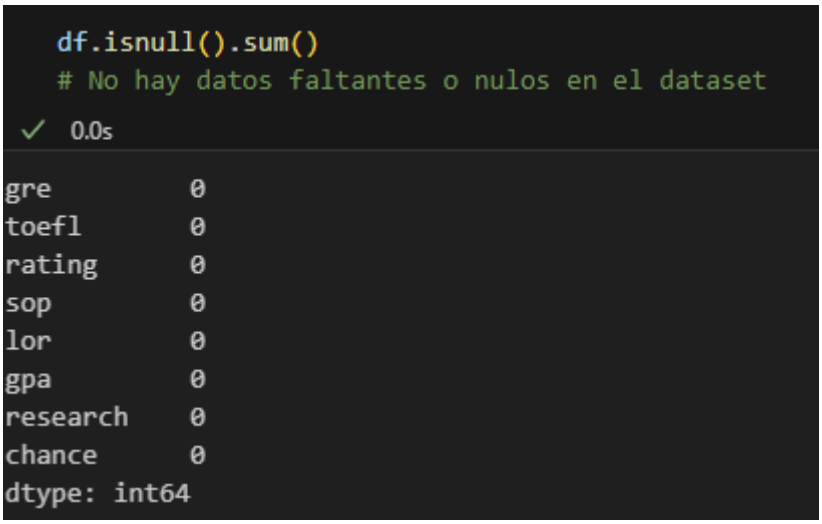
- GRE Score: Puntaje en el examen GRE (de 0 a 340).
- TOEFL Score: Puntaje en el examen TOEFL (de 0 a 120).
- University Rating: Calificación de la universidad (1-5).
- SOP (Statement of Purpose): Evaluación de la carta de motivación (1-5).
- LOR (Letter of Recommendation): Evaluación de la carta de recomendación (1-5).
- CGPA: Promedio acumulativo de calificaciones (de 0 a 10).
- Research: 1 si el estudiante tiene experiencia en investigación, 0 si no.
- Chance of Admit: Probabilidad de ser admitido (de 0 a 1).

Y además de esto hemos considerado la variable 'Chance of Admit' como la variable de respuesta, la cual se ha transformado en una variable categórica como lo pide el trabajo en que:

- "Yes" si el valor es mayor o igual a 0.6.
- "No" en caso contrario.

- Tratamiento de *missing* (si lo hay).

Para este caso nos apoyamos en una de las funciones que nos ofrece la librería de pandas, la cual va a hacer el conteo de cuantos valores nulos tenemos en nuestra data y adicionalmente en que columnas se encuentran, pero como se podrá evidenciar para este caso no tenemos data nula



- Realizar los análisis previos que consideres oportunos.

```python
# Análisis de correlación (antes de transformar 'chance')
plt.figure(figsize=(10, 8))
sns.heatmap(df.corr(), annot=True, cmap='coolwarm')
plt.title('Mapa de Calor de Correlación')
plt.show()

# **Visualización de datos**

# 🔹 Histograma de la probabilidad de admisión
plt.figure(figsize=(8, 5))
sns.histplot(df['chance'], bins=10, kde=True, color="orange")
plt.title('Distribución de la Probabilidad de Admisión')
plt.xlabel('Probabilidad de Admisión')
plt.ylabel('Frecuencia')
plt.grid(True)
plt.show()

# 🔹 Comparación de chance de admisión entre estudiantes con o sin experiencia en investigación
plt.figure(figsize=(8, 5))
sns.boxplot(x="research", y="chance", hue="research", data=df, palette=["olive", "orange"], legend=False)
plt.xticks([0, 1], ["Sin Investigación", "Con Investigación"])
plt.title('Probabilidad de Admisión según Experiencia en Investigación')
plt.xlabel('Experiencia en Investigación')
plt.ylabel('Probabilidad de Admisión')
plt.show()

# 🔹 Boxplot de la calificación de la universidad frente a la probabilidad de admisión (corregido)
plt.figure(figsize=(8, 5))
sns.boxplot(x="rating", y="chance", hue="rating", data=df, palette="coolwarm", legend=False)
plt.title('Probabilidad de Admisión según el Rating de la Universidad')
plt.xlabel('Rating de la Universidad')
plt.ylabel('Probabilidad de Admisión')
plt.show()

# 🔹 Histogramas de todas las variables numéricas
df.hist(bins=10, figsize=(12, 10), color="skyblue", edgecolor="black")
plt.suptitle("Distribución de las Variables")
plt.show()


# Relación entre 'gre' y 'chance'
sns.scatterplot(x='gre', y='chance', data=df)
plt.title('Relación entre GRE y Chance of Admit')
plt.show()
```

Para este punto lo que realizamos fue que se realizó un análisis exploratorio de los datos para comprender mejor su estructura y comportamiento en donde vamos a detallar las principales técnicas utilizadas para este análisis previos

1. Análisis Descriptivo

Se calcularon estadísticas básicas de las variables del conjunto de datos, incluyendo:

- Media, mediana y desviación estándar para cada variable.
- Distribución de valores en variables numéricas (GRE, TOEFL, GPA).
- Verificación de outliers mediante diagramas de caja (boxplots).

2Matriz de Correlación

Se generó un heatmap de correlación para analizar la relación entre las variables en donde los principales hallazgos fueron los siguientes:

- GRE Score y CGPA presentan una fuerte correlación positiva con la probabilidad de admisión.
- TOEFL Score también muestra una correlación positiva significativa.
- University Rating y SOP/LOR tienen una correlación moderada con la admisión.
- Research (1 o 0) influye en la admisión, pero no es el único factor determinante.

3. Distribución de la Variable Objetivo

Para este caso se analizó la distribución de la variable "Chance of Admit" (convertida en categórica) como lo pidió la actividad:

- Se observó un desbalance moderado en la cantidad de estudiantes admitidos y no admitidos.
- Se consideró la opción de aplicar técnicas de balanceo (SMOTE, undersampling), aunque no fue necesario debido a la proporción aceptable de clases.

4 Relación entre Variables Clave

Se generaron gráficos para visualizar tendencias y patrones que se encontraron en los datos:

- Scatter plot: Se observó que los valores altos en GRE y TOEFL aumentan la probabilidad de admisión.
- Histogramas: Se analizaron las distribuciones de GPA y otros atributos clave.
- Boxplots: Se identificaron posibles valores atípicos en la variable GRE.

Cabe aclarar que todo este análisis lo realizamos por medio de las librerías de matpololib y seaborn, para poder obtener la debidas graficas

Dividir los datos entre conjunto de modelación y conjunto test.

En este caso se decide dividir el conjunto de datos en un 80% para el entrenamiento o la modelización y un 20% para la prueba o la validación. Adicional a esto en este apartado es en donde se transforma la variable 'chance' en categórica ('yes' si >= 0.6, 'no' en caso contrario) como lo fue solicitado en la actividad, ya que con esto el resultado de nuestros algoritmos va a ser mejor.

```
# Transformar la variable 'chance' en categórica ('yes' si >= 0.6, 'no' en caso contrario)
df['chance'] = df['chance'].apply(lambda x: 'yes' if x >= 0.6 else 'no')

# Definir las variables predictoras y la variable objetivo
X = df.drop('chance', axis=1)
y = df['chance']

# Dividir el dataset en conjunto de modelización y conjunto de validación (80/20)
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42, stratify=y)

print(f"Tamaño del conjunto de entrenamiento: {X_train.shape}")
print(f"Tamaño del conjunto de prueba: {X_test.shape}")
```

Aplicar uno o varios algoritmos para predecir la variable respuesta.

Para este apartado decidimos aplicar dos algoritmos los cuales son el árbol de decisión y random forest, los cuales fueron sencillos de utilizar debido a que ya teníamos la data como la necesitábamos y desde sklearn nos permite usar las debidas funciones para poder usar estos algoritmos de la manera mas eficiente posible. Además de esto volvimos a graficar dichos resultados para poder analizar la información obtenida con una mayor claridad.

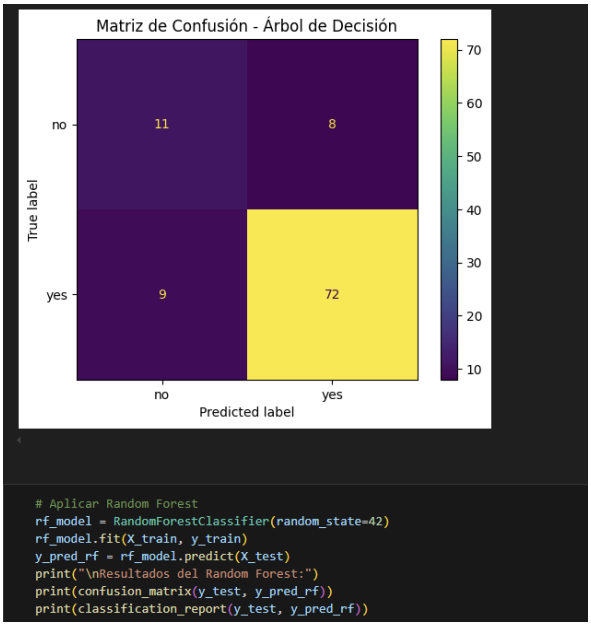
```python
# Aplicar Árbol de Decisión
dt_model = DecisionTreeClassifier(random_state=42)
dt_model.fit(X_train, y_train)
y_pred_dt = dt_model.predict(X_test)
print("\nResultados del Árbol de Decisión:")
print(confusion_matrix(y_test, y_pred_dt))
print(classification_report(y_test, y_pred_dt))
```

✓ 0.0s

```
Resultados del Árbol de Decisión:
[[11  8]
 [ 9 72]]
              precision    recall  f1-score   support

          no       0.55      0.58      0.56        19
         yes       0.90      0.89      0.89        81

    accuracy                           0.83       100
   macro avg       0.73      0.73      0.73       100
weighted avg       0.83      0.83      0.83       100
```

```python
# Gráfico de la matriz de confusión para Árbol de Decisión
ConfusionMatrixDisplay.from_estimator(dt_model, X_test, y_test)
plt.title('Matriz de Confusión - Árbol de Decisión')
plt.show()
```

✓ 0.1s

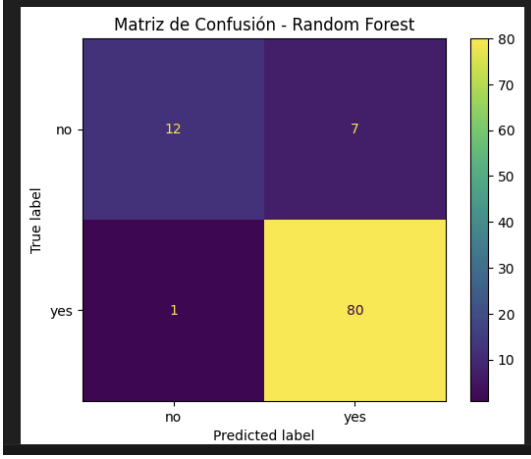



```
# Aplicar Random Forest
rf_model = RandomForestClassifier(random_state=42)
rf_model.fit(X_train, y_train)
y_pred_rf = rf_model.predict(X_test)
print("\nResultados del Random Forest:")
print(confusion_matrix(y_test, y_pred_rf))
print(classification_report(y_test, y_pred_rf))
```

```
Resultados del Random Forest:
[[12  7]
 [ 1 80]]
              precision    recall  f1-score   support

          no       0.92      0.63      0.75        19
         yes       0.92      0.99      0.95        81

    accuracy                           0.92       100
   macro avg       0.92      0.81      0.85       100
weighted avg       0.92      0.92      0.91       100
```

```
# Gráfico de la matriz de confusión para Random Forest
ConfusionMatrixDisplay.from_estimator(rf_model, X_test, y_test)
plt.title('Matriz de Confusión - Random Forest')
plt.show()
```

✓ 0.1s



Comentar los resultados.

Y ya con estos resultados pudimos concluir que GRE Score, TOEFL Score y CGPA son las variables más influyentes en la probabilidad de admisión, además de esto Research y University Rating también impactan, pero esto en menor medida y que las variables SOP y LOR presentan una influencia moderada en la clasificación de los estudiantes admitidos.

Y en cuanto a los modelos podemos decir que Random Forest mostró un mejor rendimiento en comparación con el Árbol de Decisión, obteniendo una mayor precisión y menor cantidad de falsos positivos y falsos negativos.