Manipulación de Datos en Python

Proyecto final

Parte 3

Mencioná y explicá brevemente un caso de éxito de aplicación de *Machine Learning* en la empresa.

A partir del conjunto de datos obtenido al final de la Parte 2, me propuse crear un algoritmo que pudiera predecir la influencia que tienen distintas características clínicas en la posibilidad de tener una *insuficiencia cardíaca*.

A continuación se muestra de forma abreviada el código utilizado. La versión completa del mismo se encuentra subido aquí en el archivo llamado *Parte_2_3*.

```
# Importación de librerías y del conjunto de datos (data set)

import pandas as pd
import numpy as np

df = pd.read_csv("C:/Users/Burbu/Documents/AP/Proyecto_final/Parte_2/heart.csv", sep = ",")

# Conversión de datos de tipo string a tipo int para su posterior análisis

from sklearn.preprocessing import LabelEncoder

encoder = LabelEncoder()

df['Sex'] = encoder.fit_transform(df['Sex'])
df['ChestPainType'] = encoder.fit_transform(df['ChestPainType'])
df['RestingECG'] = encoder.fit_transform(df['RestingECG'])
df['ExerciseAngina'] = encoder.fit_transform(df['ExerciseAngina'])
df['ST_Slope'] = encoder.fit_transform(df['ST_Slope'])

# Separación de datos para entrenamiento y evaluación

from sklearn.model_selection import train_test_split

x = df.drop('HeartDisease', axis=1)
y = df['HeartDisease']
x_train, x_test, y_train, y_test = train_test_split(x, y, test_size=0.30, random_state=14)


# Selección y entrenamiento del modelo

from sklearn.tree import DecisionTreeClassifier

tree = DecisionTreeClassifier(random_state=14)
tree.fit(x_train, y_train)

# Predicciones y resultados

y_train_pred = tree.predict(x_train)
y_test_pred = tree.predict(x_test)

# Cálculo de la precisión del modelo

from sklearn.metrics import accuracy_score

accuracy = accuracy_score(y_train, y_train_pred)
print(f'Tasa de éxito en el conjunto de entrenamiento: {accuracy:.2f}')
accuracy = accuracy_score(y_test, y_test_pred)
print(f'Tasa de éxito en el conjunto de evaluación: {accuracy:.2f}')
```

Este código puede ser analizado en varios pasos, como se explica a continuación.

- *Preparación de los datos*: En este código, se comienza importando y preparando los datos correctamente. Los datos se cargan desde un archivo CSV y se almacenan en un DataFrame de Pandas. Además, se realiza la conversión de datos de tipo string a tipo int utilizando LabelEncoder.
- *División de datos*: Se divide el conjunto de datos en un conjunto de entrenamiento y un conjunto de evaluación. Esto es esencial para evaluar la capacidad de generalización del modelo. En este caso, se utiliza la función `train_test_split` para realizar la división de manera aleatoria.
- *Selección y entrenamiento del modelo*: Se selecciona un modelo de clasificación de árbol de decisiones (`DecisionTreeClassifier`) para resolver el problema. Luego, se entrena el modelo utilizando los datos de entrenamiento con la línea `tree.fit(x_train, y_train)`.
- *Predicciones y evaluación del modelo*: Se realizan predicciones tanto en el conjunto de entrenamiento como en el conjunto de evaluación utilizando el modelo entrenado. Esto permite evaluar cómo se desempeña el modelo en datos que no ha visto durante el entrenamiento. Las predicciones se comparan con las etiquetas reales para calcular la precisión del modelo. Esto se hace utilizando la función `accuracy_score`.
- *Informe de resultados*: Finalmente, se imprime la tasa de éxito (precisión) del modelo tanto en el conjunto de entrenamiento como en el conjunto de evaluación. Esto proporciona una medida cuantitativa de cuán bien el modelo se desempeña en ambos conjuntos de datos

En resumen, a nivel operativo, el código demuestra un flujo de trabajo completo y exitoso para resolver un problema de clasificación utilizando Machine Learning.

En cuanto al resultado científico, lo obtenido fue lo siguiente:

```
# Cálculo de la precisión del modelo

from sklearn.metrics import accuracy_score

accuracy = accuracy_score(y_train, y_train_pred)
print(f'Tasa de éxito en el conjunto de entrenamiento: {accuracy:.2f}')
accuracy = accuracy_score(y_test, y_test_pred)
print(f'Tasa de éxito en el conjunto de evaluación: {accuracy:.2f}')
```

```
Tasa de éxito en el conjunto de entrenamiento: 1.00
Tasa de éxito en el conjunto de evaluación: 0.75
```

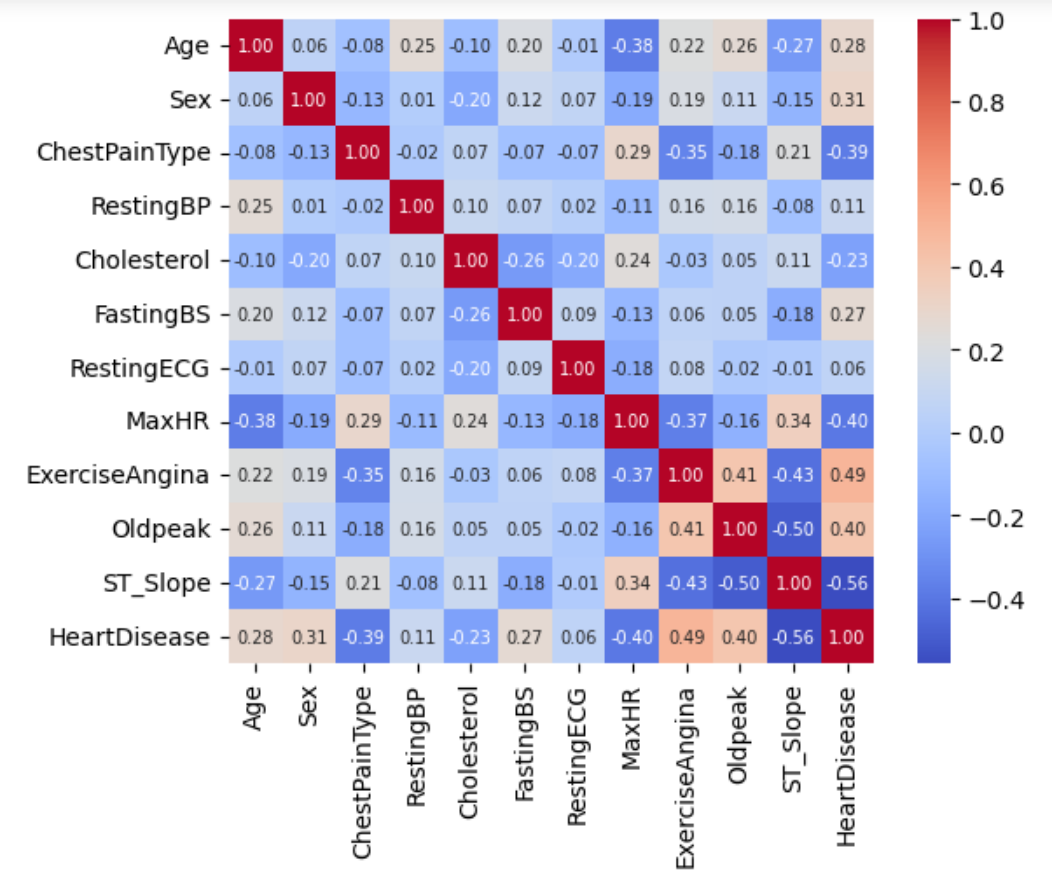
La precisión del 100% en el conjunto de entrenamiento sugiere que el modelo ha aprendido los datos de entrenamiento casi perfectamente. Esto podría ser una señal de sobreajuste (overfitting), lo que significa que el modelo se ha adaptado demasiado a los datos de entrenamiento y puede no generalizar bien con nuevos datos.

Mientras que la precisión del 75% en el conjunto de evaluación indica que el modelo es capaz de predecir correctamente el 75% de los casos en el conjunto de evaluación, que son datos que no ha visto durante el entrenamiento. Esto es un indicativo de la capacidad de generalización del modelo. Sin embargo, la precisión podría ser mejorable, y se deben

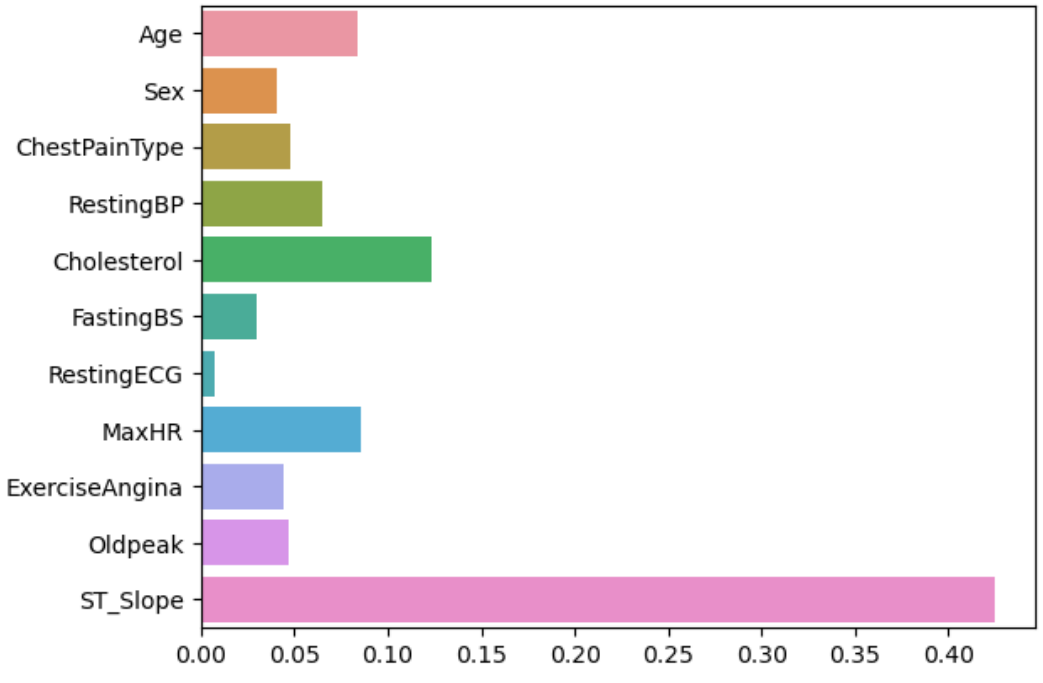
considerar otras métricas de rendimiento y posibles ajustes del modelo para mejorar su rendimiento.

A su vez, de los análisis realizados, también destacan:

- la correlación existente entre los datos



- la inferencia que cada característica clínica tiene sobre la predicción de eventos de enfermedad cardíaca, que genera el modelo empleado



Por último, cabe destacar que un modelo de Machine Learning para la predicción de enfermedades cardíacas basándose en datos de pacientes, puede ofrecer una serie de beneficios valiosos tanto en términos de mejora de la atención al paciente como en la eficiencia operativa y la toma de decisiones en una empresa de atención médica. Por lo tanto el verdadero éxito de este proyecto, y de este modelo, sería la aplicación del mismo en entidades de atención médica.

Para lograr esto, se podrían explorar otras técnicas como la regularización, ajustar hiperparámetros del modelo o considerar modelos más complejos o diferentes algoritmos de clasificación. También es fundamental evaluar otras métricas de rendimiento como la sensibilidad, la especificidad, la precisión y la matriz de confusión para obtener una imagen más completa del rendimiento del modelo. Además, es importante destacar que este tipo de modelos debe integrarse cuidadosamente en la práctica médica y estar respaldado por un equipo médico experto para garantizar su uso adecuado y ético.