

Creado por:

Isabel Maniega

Decission Tree Classification

```
In [2]: import pandas as pd
from sklearn.tree import DecisionTreeClassifier
from sklearn.model_selection import train_test_split
from sklearn import metrics
```

```
In [3]: # Dataset:
# https://www.kaggle.com/datasets/uciml/pima-indians-diabetes-database?resource=download
```

```
In [5]: col_names = ["pregnant", "glucose", "bp", "skin", "insulin", "bmi", "pedigree", "age", "label"]

pima = pd.read_csv("diabetes.csv", header=None, names=col_names)
```

```
In [6]: pima.head()
```

Out[6]:

	pregnant	glucose	bp	skin	insulin	bmi	pedigree	age	label
0	Pregnancies	Glucose	BloodPressure	SkinThickness	Insulin	BMI	DiabetesPedigreeFunction	Age	Outcome
1	6	148	72	35	0	33.6	0.627	50	1
2	1	85	66	29	0	26.6	0.351	31	0
3	8	183	64	0	0	23.3	0.672	32	1
4	1	89	66	23	94	28.1	0.167	21	0

```
In [8]: pima.drop(index=pima.index[0], axis=0, inplace=True)
pima.head()
```

Out[8]:

	pregnant	glucose	bp	skin	insulin	bmi	pedigree	age	label
1	6	148	72	35	0	33.6	0.627	50	1
2	1	85	66	29	0	26.6	0.351	31	0
3	8	183	64	0	0	23.3	0.672	32	1
4	1	89	66	23	94	28.1	0.167	21	0
5	0	137	40	35	168	43.1	2.288	33	1

```
In [9]: feature_cols = ["pregnant", "glucose", "bp", "insulin", "bmi", "pedigree", "age"]
X = pima[feature_cols]
y = pima.label
```

```
In [10]: X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.3, random_state=1)
```

```
In [11]: # Creamos el clasificador

clf = DecisionTreeClassifier()

clf = clf.fit(X_train, y_train)

y_pred = clf.predict(X_test)
y_pred
```

```
Out[11]: array(['0', '0', '0', '1', '0', '1', '0', '0', '0', '0', '1', '1', '1',
        '1', '0', '0', '1', '0', '1', '0', '1', '0', '1', '0', '1',
        '0', '1', '0', '0', '0', '1', '0', '1', '1', '0', '1', '0',
        '1', '0', '0', '0', '0', '0', '0', '0', '0', '0', '0', '1',
        '1', '0', '1', '0', '1', '0', '0', '1', '0', '0', '1', '0',
        '0', '1', '1', '0', '1', '0', '0', '1', '0', '0', '1', '0',
        '0', '0', '0', '1', '0', '1', '0', '1', '0', '0', '1', '1',
        '0', '0', '0', '1', '0', '0', '0', '0', '0', '0', '0', '0',
        '0', '0', '0', '0', '1', '0', '0', '0', '0', '0', '0', '0',
        '1', '0', '0', '0', '0', '0', '0', '1', '0', '0', '1', '0',
        '1', '0', '0', '0', '0', '0', '0', '1', '0', '1', '0', '1',
        '1', '0', '0', '0', '1', '0', '0', '0', '1', '0', '1', '1',
        '1', '1', '1', '1', '0', '1', '0', '1', '0', '0', '0', '1',
        '0', '0', '0', '0', '0', '1', '0', '0', '0', '0', '0', '1',
        '0', '1', '0', '0', '1', '1', '1', '1', '0', '1', '1', '0',
        '0', '0', '0', '0', '0', '1', '1', '1', '0', '0', '0', '0',
        '0', '0', '1', '0', '0', '0', '0', '0', '1', '0'], dtype=object)
```

```
In [12]: y_test
```

```
Out[12]: 286    0
102     0
582     0
353     0
727     0
..
242     0
600     0
651     0
12      1
215     1
Name: label, Length: 231, dtype: object
```

```
In [13]: print("Accuracy:", metrics.accuracy_score(y_test, y_pred))

Accuracy: 0.6796536796536796
```

```
In [14]: # Otra opción de clasificador

clf = DecisionTreeClassifier(criterion="entropy", max_depth=3)

clf = clf.fit(X_train, y_train)

y_pred = clf.predict(X_test)

print("Accuracy:", metrics.accuracy_score(y_test, y_pred))

Accuracy: 0.7705627705627706
```

Creado por:

Isabel Maniega