Random Forest para Clasificación

In [5]:

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
from sklearn.ensemble import RandomForestClassifier
```

In [2]:

```
df = pd.read_csv("../../Data-Sets/datasets/iris/iris.csv")
```

In [3]:

```
colnames = df.columns.values.tolist()
predictors = colnames[:4]
target = colnames[4]
X = df[predictors]
Y = df[target]
```

In [10]:

```
forest = RandomForestClassifier(n_jobs=2, oob_score=True, n_estimators=500)
forest.fit(X,Y)
```

Out[10]:

```
RandomForestClassifier(n_estimators=500, n_jobs=2, oob_score=True)
```

In [11]:

```
forest.oob_decision_function_
```

Out[11]:

```
array([[1.        , 0.        , 0.        ],
       [1.        , 0.        , 0.        ],
       [1.        , 0.        , 0.        ],
       [1.        , 0.        , 0.        ],
       [1.        , 0.        , 0.        ],
       [1.        , 0.        , 0.        ],
       [1.        , 0.        , 0.        ],
       [1.        , 0.        , 0.        ],
       [1.        , 0.        , 0.        ],
       [1.        , 0.        , 0.        ],
       [1.        , 0.        , 0.        ],
       [1.        , 0.        , 0.        ],
       [1.        , 0.        , 0.        ],
       [1.        , 0.        , 0.        ],
       [0.97474747, 0.02525253, 0.        ],
       [0.98918919, 0.01081081, 0.        ],
       [1.        , 0.        , 0.        ],
       [1.        , 0.        , 0.        ],
       [0.98      , 0.02      , 0.        ],
       [1.        , 0.        , 0.        ],
       [1.        , 0.        , 0.        ],
       [1.        , 0.        , 0.        ],
       [1.        , 0.        , 0.        ],
       [1.        , 0.        , 0.        ],
       [1.        , 0.        , 0.        ],
       [1.        , 0.        , 0.        ],
       [1.        , 0.        , 0.        ],
       [1.        , 0.        , 0.        ],
       [1.        , 0.        , 0.        ],
       [1.        , 0.        , 0.        ],
       [1.        , 0.        , 0.        ],
       [1.        , 0.        , 0.        ],
       [1.        , 0.        , 0.        ],
       [1.        , 0.        , 0.        ],
       [1.        , 0.        , 0.        ],
       [1.        , 0.        , 0.        ],
```

```
[0.9516129 , 0.0483871 , 0.        ],
[1.        , 0.        , 0.        ],
[1.        , 0.        , 0.        ],
[1.        , 0.        , 0.        ],
[1.        , 0.        , 0.        ],
[0.96446701, 0.03045685, 0.00507614],
[1.        , 0.        , 0.        ],
[1.        , 0.        , 0.        ],
[1.        , 0.        , 0.        ],
[1.        , 0.        , 0.        ],
[1.        , 0.        , 0.        ],
[1.        , 0.        , 0.        ],
[1.        , 0.        , 0.        ],
[1.        , 0.        , 0.        ],
[0.        , 0.9689441 , 0.0310559 ],
[0.        , 0.9893617 , 0.0106383 ],
[0.        , 0.78857143, 0.21142857],
[0.        , 0.98453608, 0.01546392],
[0.        , 0.98412698, 0.01587302],
[0.        , 1.        , 0.        ],
[0.        , 0.9516129 , 0.0483871 ],
[0.        , 0.79661017, 0.20338983],
[0.        , 1.        , 0.        ],
[0.        , 0.9704142 , 0.0295858 ],
[0.00529101, 0.91005291, 0.08465608],
[0.        , 0.99459459, 0.00540541],
[0.        , 0.9039548 , 0.0960452 ],
[0.        , 0.984375  , 0.015625  ],
[0.        , 1.        , 0.        ],
[0.        , 1.        , 0.        ],
[0.        , 1.        , 0.        ],
[0.        , 0.99425287, 0.00574713],
[0.        , 0.89673913, 0.10326087],
[0.        , 1.        , 0.        ],
[0.        , 0.04395604, 0.95604396],
[0.        , 0.98888889, 0.01111111],
[0.        , 0.55675676, 0.44324324],
[0.        , 0.98369565, 0.01630435],
[0.        , 1.        , 0.        ],
[0.        , 1.        , 0.        ],
[0.        , 0.86705202, 0.13294798],
[0.        , 0.03389831, 0.96610169],
[0.        , 1.        , 0.        ],
[0.        , 1.        , 0.        ],
[0.        , 1.        , 0.        ],
[0.        , 1.        , 0.        ],
[0.        , 0.99465241, 0.00534759],
[0.        , 0.14606742, 0.85393258],
[0.04123711, 0.91752577, 0.04123711],
[0.01463415, 0.97560976, 0.0097561 ],
[0.        , 1.        , 0.        ],
[0.        , 0.97927461, 0.02072539],
[0.        , 1.        , 0.        ],
[0.        , 0.99450549, 0.00549451],
[0.        , 0.98224852, 0.01775148],
[0.        , 0.98941799, 0.01058201],
[0.        , 0.98816568, 0.01183432],
[0.00473934, 0.94312796, 0.0521327 ],
[0.        , 1.        , 0.        ],
[0.        , 1.        , 0.        ],
[0.        , 1.        , 0.        ],
[0.        , 0.98809524, 0.01190476],
[0.        , 0.9742268 , 0.0257732 ],
[0.        , 1.        , 0.        ],
[0.        , 0.0097561 , 0.9902439 ],
[0.        , 0.02380952, 0.97619048],
```

```
[0.        , 0.        , 1.        ],
[0.        , 0.        , 1.        ],
[0.        , 0.        , 1.        ],
[0.        , 0.        , 1.        ],
[0.        , 0.86592179, 0.13407821],
[0.        , 0.00526316, 0.99473684],
[0.        , 0.01036269, 0.98963731],
[0.        , 0.        , 1.        ],
[0.        , 0.02631579, 0.97368421],
[0.        , 0.00537634, 0.99462366],
[0.        , 0.        , 1.        ],
[0.        , 0.08474576, 0.91525424],
[0.        , 0.01176471, 0.98823529],
[0.        , 0.        , 1.        ],
[0.        , 0.        , 1.        ],
[0.        , 0.        , 1.        ],
[0.        , 0.        , 1.        ],
[0.        , 0.77777778, 0.22222222],
[0.        , 0.        , 1.        ],
[0.        , 0.13265306, 0.86734694],
[0.        , 0.        , 1.        ],
[0.        , 0.08522727, 0.91477273],
[0.        , 0.        , 1.        ],
[0.        , 0.01666667, 0.98333333],
[0.        , 0.2244898 , 0.7755102 ],
[0.        , 0.05612245, 0.94387755],
[0.        , 0.        , 1.        ],
[0.        , 0.45180723, 0.54819277],
[0.        , 0.        , 1.        ],
[0.        , 0.        , 1.        ],
[0.        , 0.        , 1.        ],
[0.        , 0.62745098, 0.37254902],
[0.        , 0.5026455 , 0.4973545 ],
[0.        , 0.        , 1.        ],
[0.        , 0.01075269, 0.98924731],
[0.        , 0.00613497, 0.99386503],
[0.        , 0.47619048, 0.52380952],
[0.        , 0.        , 1.        ],
[0.        , 0.        , 1.        ],
[0.        , 0.05102041, 0.94897959],
[0.        , 0.02298851, 0.97701149],
[0.        , 0.        , 1.        ],
[0.        , 0.        , 1.        ],
[0.        , 0.01507538, 0.98492462],
[0.        , 0.07471264, 0.92528736],
[0.        , 0.        , 1.        ],
[0.        , 0.02873563, 0.97126437],
[0.        , 0.12121212, 0.87878788]])
```

Con ese array podemos ver con qué probabilidad se ha votado que cada elemento sea cual, el array tiene tantas columnas como targets existan para clasificar. A veces pueden haber nan, lo que significa que las variables que ha tomado el árbol para decidir no habrá incluido ninguna variable relevante para clasificar, no habrá ninguna que minimice la entropía. Mientras más árboles decidamos poner, entonces menos probable es que aparezcan nans.

In [12]:
```
forest.oob_score_
```

Out[12]:
```
0.9533333333333334
```

Como se vé, obtenemos inclusive más presición que cuando utilizamos validación cruzada y poda de árbol en el método de clasificación, donde obtuvimos para una profundidad de 3 es de **0.94** como valor máximo. A base de ir creando árboles, maximizamos la presición del modelo.