Iris Species

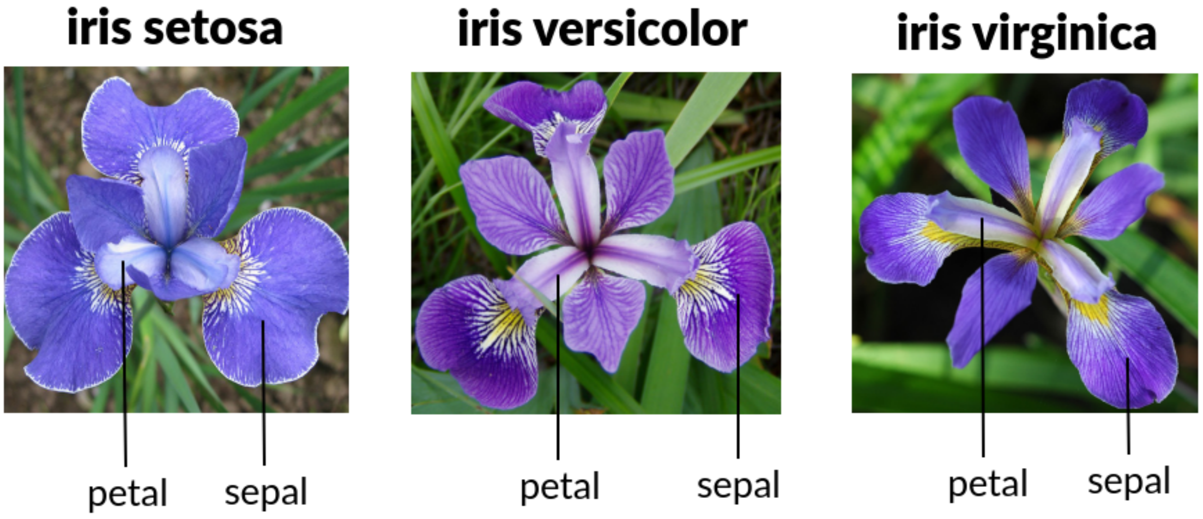
Classify iris plants into three species in this classic dataset

The Iris dataset was used in R.A. Fisher's classic 1936 paper, The Use of Multiple Measurements in Taxonomic Problems (http://rcs.chemometrics.ru/Tutorials/classification/Fisher.pdf), and can also be found on the UCI Machine Learning Repository (http://archive.ics.uci.edu/ml/).

It includes three iris species with 50 samples each as well as some properties about each flower. One flower species is linearly separable from the other two, but the other two are not linearly separable from each other.

The columns in this dataset are:

- **Id:** SPL-SPW-PTL-PTW(CM)
- **SepalLengthCm:** Length of the sepal (in cm)
- **SepalWidthCm:** Width of the sepal (in cm)
- **PetalLengthCm:** Length of the petal (in cm)
- **PetalWidthCm:** Width of the petal (in cm)
- **Species:** Species name



In [1]:

```python
import pandas as pd

col = ['sepal length (cm)', 'sepal width (cm)',
       'petal length (cm)', 'petal width (cm)','Target']
iris_df = pd.read_csv('uci-iris/iris.data.csv',names=col)

iris_df.head()
```

Out[1]:

	sepal length (cm)	sepal width (cm)	petal length (cm)	petal width (cm)	Target
0	5.1	3.5	1.4	0.2	Iris-setosa
1	4.9	3.0	1.4	0.2	Iris-setosa
2	4.7	3.2	1.3	0.2	Iris-setosa
3	4.6	3.1	1.5	0.2	Iris-setosa
4	5.0	3.6	1.4	0.2	Iris-setosa

In [2]:

```python
pd.get_dummies(iris_df['Target'],drop_first=True).head()
# Convert categorical variable into dummy/indicator variables
```

Out[2]:

	Iris-versicolor	Iris-virginica
0	0	0
1	0	0
2	0	0
3	0	0
4	0	0

In [3]:

```python
from sklearn.preprocessing import LabelEncoder

labEnc = LabelEncoder()

Y = labEnc.fit_transform(iris_df['Target'])
Y
```

Out[3]:

```
array([0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
       0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
       0, 0, 0, 0, 0, 0, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,
       1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,
       1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2,
       2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2,
       2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2])
```

- 0 Iris-Setosa
- 1 Iris-Versicolour
- 2 Iris-Virginica

In [4]:

```
import seaborn as sns

sns.pairplot(iris_df,hue='Target')
```

Out[4]:

```
<seaborn.axisgrid.PairGrid at 0x26a0034a860>
```

In [5]:

```
X = iris_df[['petal length (cm)','petal width (cm)']]
Y = labEnc.fit_transform(iris_df['Target'])
```

In [6]:

```
from sklearn.linear_model import LogisticRegression

log_reg = LogisticRegression()

log_reg.fit(X,Y)
```

```
C:\Users\Jesus\Anaconda3\lib\site-packages\sklearn\linear_model\logistic.py:
433: FutureWarning: Default solver will be changed to 'lbfgs' in 0.22. Speci
fy a solver to silence this warning.
  FutureWarning)
C:\Users\Jesus\Anaconda3\lib\site-packages\sklearn\linear_model\logistic.py:
460: FutureWarning: Default multi_class will be changed to 'auto' in 0.22. S
pecify the multi_class option to silence this warning.
  "this warning.", FutureWarning)
```

Out[6]:

```
LogisticRegression(C=1.0, class_weight=None, dual=False, fit_intercept=True,
          intercept_scaling=1, max_iter=100, multi_class='warn',
          n_jobs=None, penalty='l2', random_state=None, solver='warn',
          tol=0.0001, verbose=0, warm_start=False)
```

In [7]:

```
log_reg.predict([[1.4,0.2]])
```

Out[7]:

```
array([0])
```

In [8]:

```
log_reg.predict_proba([[1.5,0.5]])
```

Out[8]:

```
array([[0.75926153, 0.1935966 , 0.04714187]])
```

In [9]:

```
Y_pred = log_reg.predict(X)
```

In [10]:

```
from sklearn.metrics import confusion_matrix,accuracy_score

print(confusion_matrix(Y,Y_pred))

print(accuracy_score(Y,Y_pred))
```

```
[[50  0  0]
 [ 0 35 15]
 [ 0  4 46]]
0.8733333333333333
```