```python
import pandas as pd
import numpy as np
from sklearn import datasets
from sklearn.cluster import KMeans
import matplotlib.pyplot as plt
import matplotlib.patches as mpatches
import sklearn.metrics as sm
%matplotlib inline
```

```python
iris = datasets.load_iris()
print(iris.data)
```

```
    [6.1 3.  4.6 1.4]
    [5.8 2.6 4.  1.2]
    [5.  2.3 3.3 1. ]
    [5.6 2.7 4.2 1.3]
    [5.7 3.  4.2 1.2]
    [5.7 2.9 4.2 1.3]
    [6.2 2.9 4.3 1.3]
    [5.1 2.5 3.  1.1]
    [5.7 2.8 4.1 1.3]
    [6.3 3.3 6.  2.5]
    [5.8 2.7 5.1 1.9]
    [7.1 3.  5.9 2.1]
    [6.3 2.9 5.6 1.8]

    [6.5 3.  5.8 2.2]
    [7.6 3.  6.6 2.1]
    [4.9 2.5 4.5 1.7]
    [7.3 2.9 6.3 1.8]
    [6.7 2.5 5.8 1.8]
    [7.2 3.6 6.1 2.5]
    [6.5 3.2 5.1 2. ]
    [6.4 2.7 5.3 1.9]
    [6.8 3.  5.5 2.1]
    [5.7 2.5 5.  2. ]
    [5.8 2.8 5.1 2.4]
    [6.4 3.2 5.3 2.3]
    [6.5 3.  5.5 1.8]
    [7.7 3.8 6.7 2.2]
    [7.7 2.6 6.9 2.3]
    [6.  2.2 5.  1.5]
    [6.9 3.2 5.7 2.3]
    [5.6 2.8 4.9 2. ]
    [7.7 2.8 6.7 2. ]
    [6.3 2.7 4.9 1.8]
    [6.7 3.3 5.7 2.1]
    [7.2 3.2 6.  1.8]
    [6.2 2.8 4.8 1.8]
    [6.1 3.  4.9 1.8]
    [6.4 2.8 5.6 2.1]
    [7.2 3.  5.8 1.6]
    [7.4 2.8 6.1 1.9]
    [7.9 3.8 6.4 2. ]
    [6.4 2.8 5.6 2.2]
    [6.3 2.8 5.1 1.5]
    [6.1 2.6 5.6 1.4]
    [7.7 3.  6.1 2.3]
```

```
[6.3 3.4 5.6 2.4]
[6.4 3.1 5.5 1.8]
[6.  3.  4.8 1.8]
[6.9 3.1 5.4 2.1]
[6.7 3.1 5.6 2.4]
[6.9 3.1 5.1 2.3]
[5.8 2.7 5.1 1.9]
[6.8 3.2 5.9 2.3]
[6.7 3.3 5.7 2.5]
[6.7 3.  5.2 2.3]
[6.3 2.5 5.  1.9]
[6.5 3.  5.2 2. ]
[6.2 3.4 5.4 2.3]
[5 9 3   5 1 1 8]]
```

iris.target_names

```
array(['setosa', 'versicolor', 'virginica'], dtype='<U10')
```

iris.target

```
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, 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])
```

```
x = pd.DataFrame(iris.data, columns=['Sepal Length', 'Sepal Width', 'Petal Length', 'Petal
y = pd.DataFrame(iris.target, columns=['Target'])
```

x.head()

	Sepal Length	Sepal Width	Petal Length	Petal Width
0	5.1	3.5	1.4	0.2
1	4.9	3.0	1.4	0.2
2	4.7	3.2	1.3	0.2
3	4.6	3.1	1.5	0.2
4	5.0	3.6	1.4	0.2

```
iris_k_mean_model = KMeans(n_clusters=3)
iris_k_mean_model.fit(x)
```

```
KMeans(n_clusters=3)
```

iris_k_mean_model.labels_

```
array([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, 0, 0, 2, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
        0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 2, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
        0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 2, 0, 2, 2, 2, 2, 0, 2, 2, 2,
        2, 2, 2, 0, 0, 2, 2, 2, 2, 0, 2, 0, 2, 0, 2, 2, 0, 0, 2, 2, 2, 2,
        2, 0, 2, 2, 2, 2, 0, 2, 2, 2, 0, 2, 2, 2, 0, 2, 2, 0], dtype=int32)
```

```
iris_k_mean_model.cluster_centers_
```

```
    array([[5.9016129 , 2.7483871 , 4.39354839, 1.43387097],
           [5.006      , 3.428      , 1.462      , 0.246      ],
           [6.85       , 3.07368421, 5.74210526, 2.07105263]])
```

```python
plt.figure(figsize=(12,3))

red_patch = mpatches.Patch(color='red', label='Setosa')
green_patch = mpatches.Patch(color='green', label='Versicolor')
blue_patch = mpatches.Patch(color='blue', label='Virginica')

colors = np.array(['red', 'green', 'blue'])

predictedY = np.choose(iris_k_mean_model.labels_, [1, 0, 2]).astype(np.int64)

plt.subplot(1, 2, 1)
plt.scatter(x['Petal Length'], x['Petal Width'], c=colors[y['Target']])
plt.title('Before classification')
plt.legend(handles=[red_patch, green_patch, blue_patch])

plt.subplot(1, 2, 2)
plt.scatter(x['Petal Length'], x['Petal Width'], c=colors[predictedY])
plt.title("Model's classification")
plt.legend(handles=[red_patch, green_patch, blue_patch])
```
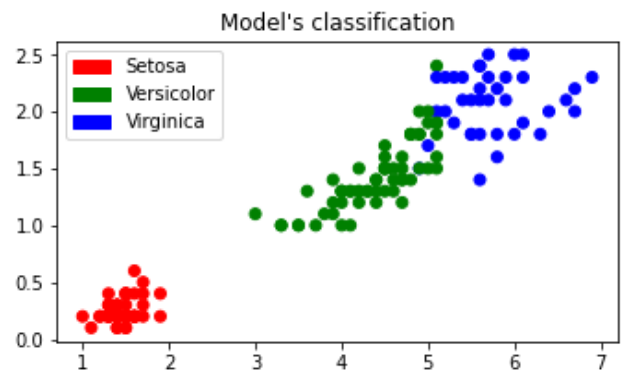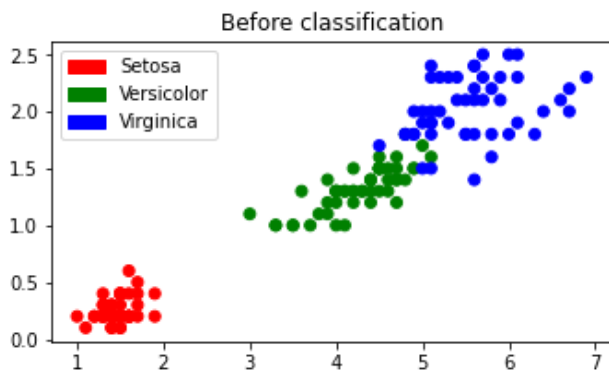
```
    <matplotlib.legend.Legend at 0x7f922ceb3e10>
```



```
sm.accuracy_score(predictedY, y['Target'])
```

```
    0.8933333333333333
```

Colab paid products - Cancel contracts here

✓ 462630h 25m 42s   completed at 11:55 AM   ● ✕