

Assignment - 3 [SKLearn & Matplotlib]

```
In[1]: from sklearn.datasets import load_iris
```

```
In[2]: iris = load_iris()
```

```
print(iris)
```

```
{'data': array([5.1, 3.5, 1.4, 0.2],
```

```
[4.9, 3., 1.4, 0.2],
```

```
[4.7, 3.2, 1.3, 0.2],
```

```
[4.6, 3.1, 1.5, 0.2],
```

```
[5.0, 3.6, 1.4, 0.2],
```

```
-----
```

```
-----
```

```
[4.8, 3.4, 1.9, 0.2], }
```

```
In[3]: import pandas as pd
```

```
In[4]: data = pd.DataFrame(iris.data, columns = iris.feature_names)
```

```
print(data)
```

	sepal length(cm)	sepal width(cm)	petal length(cm)	petal width(cm)
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
---	-----	----	----	----
148	6.2	3.4	5.4	2.3
149	5.9	3.0	5.1	1.8

In[5]: data.head()

	Sepal length(cm)	sepal width(cm)	petal length(cm)	petal width(cm)
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

In[6]: data['Species'] = pd.DataFrame(iris.target)
print(data)

	Sepal length(cm)	Sepal Width(cm)	Petal Length(cm)	Petal Width(cm)
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
--	--	--	--	--
148	6.2	3.4	5.4	2.3
149	5.9	3.0	5.1	1.8

	Species
0	0
1	0
2	0
--	--
147	2
148	2
149	2

In[7]: data.head()

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

In[8]: x = data.iloc[:, :-1]
y = data.iloc[:, -1]

In[9]: from sklearn.model_selection import train_test_split
x_train, x_test, y_train, y_test = train_test_split(x, y, test_size=0.3)

In[10]: from sklearn.tree import DecisionTreeClassifier
model = DecisionTreeClassifier(criterion="entropy", splitter="best")

In[11]: model_fit = model.fit(x_train, y_train)
y_pred = model.predict(x_test)

In[12]: from sklearn.metrics import accuracy_score
print(accuracy_score(y_pred, y_test).round(2)*100)
96.0

```

In[13]: from matplotlib import pyplot as plt
        %matplotlib inline
        from sklearn.tree import plot_tree

```

```

In[14]: plt.figure(figsize=(25,20))
        plot_tree(model_fit, feature_names=iris.feature_names,
                  class_names=iris.target_names,
                  filled=True)
        plt.show()

```

