

# iris\_Dicision\_tree\_classification\_assignment

January 19, 2023

## 1 Multilinear Reression

```
[ ]: #Import Libraris
import pandas as pd
import seaborn as sns
from sklearn.tree import DecisionTreeClassifier
import matplotlib.pyplot as plt
from sklearn.model_selection import train_test_split
import numpy as np
```

```
[ ]: #Import dataset
iris = sns.load_dataset('iris')
```

```
[ ]: iris.head()
```

```
[ ]:      sepal_length  sepal_width  petal_length  petal_width  species
0           5.1           3.5           1.4           0.2  setosa
1           4.9           3.0           1.4           0.2  setosa
2           4.7           3.2           1.3           0.2  setosa
3           4.6           3.1           1.5           0.2  setosa
4           5.0           3.6           1.4           0.2  setosa
```

```
[ ]: iris.isnull().sum() / len(iris)*100
```

```
[ ]: sepal_length    0.0
     sepal_width    0.0
     petal_length    0.0
     petal_width    0.0
     species        0.0
     dtype: float64
```

```
[ ]: iris.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 150 entries, 0 to 149
Data columns (total 5 columns):
 #   Column          Non-Null Count  Dtype
---  -
 0   sepal_length    150 non-null   float64
```

```

1  sepal_width    150 non-null    float64
2  petal_length   150 non-null    float64
3  petal_width    150 non-null    float64
4  species        150 non-null    object
dtypes: float64(4), object(1)
memory usage: 6.0+ KB

```

```
[ ]: iris = iris[['sepal_length', 'sepal_width', 'petal_length', 'petal_width']]
iris.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

```

```
[ ]: X = iris[['sepal_length', 'sepal_width', 'petal_length', 'petal_width']]
y = iris['species']
```

```
[ ]: model = DecisionTreeClassifier().fit(X, y)
model.predict([[6.3, 3.5, 1.4, 0.2]])
```

```

c:\Users\muham\AppData\Local\Programs\Python\Python311\Lib\site-
packages\sklearn\base.py:409: UserWarning: X does not have valid feature names,
but DecisionTreeClassifier was fitted with feature names
  warnings.warn(

```

```
[ ]: array(['setosa'], dtype=object)
```

```
[ ]: model.score
```

```
[ ]: <bound method ClassifierMixin.score of DecisionTreeClassifier(>
```

```
[ ]: # Model evaluation
X_train, X_test, y_train, y_test = train_test_split(X, y, train_size=0.8)

model = DecisionTreeClassifier().fit(X_train, y_train)
prediction = model.predict(X_test)
```

```
[ ]: model.score(X_test, y_test)
```

```
[ ]: 1.0
```

```
[ ]: from sklearn.metrics import accuracy_score
import joblib
score = accuracy_score(y_test, prediction)
model = DecisionTreeClassifier().fit(X, y)
```

```
joblib.dump(model, 'Species')
```

```
[ ]: ['Species']
```

```
[ ]: plt.figure(figsize=(25,20))  
model = DecisionTreeClassifier().fit(X, y)  
tree.plot_tree(model, filled=True)  
plt.show()
```

