day_18_Assignment02

January 19, 2023

Subject of notebook :Decision tree classification on Iris dataset
Name of the auther : Qadir Shahbaz
Where to contact : qadir_shahbaz@yahoo.co.uk
date : 19/01/2023

1 Decision tree classification on Iris dataset

[ ]:
```
# Import libraries
import pandas as pd
import numpy as np
import seaborn as sns
import matplotlib.pyplot as plt
```

[ ]:
```
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LogisticRegression
```

[ ]:
```
df = sns.load_dataset("iris")
```

[ ]:
```
df.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

[ ]:
```
X = df[["sepal_length","sepal_width","petal_length","petal_width"]]
y = df["species"]
```

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

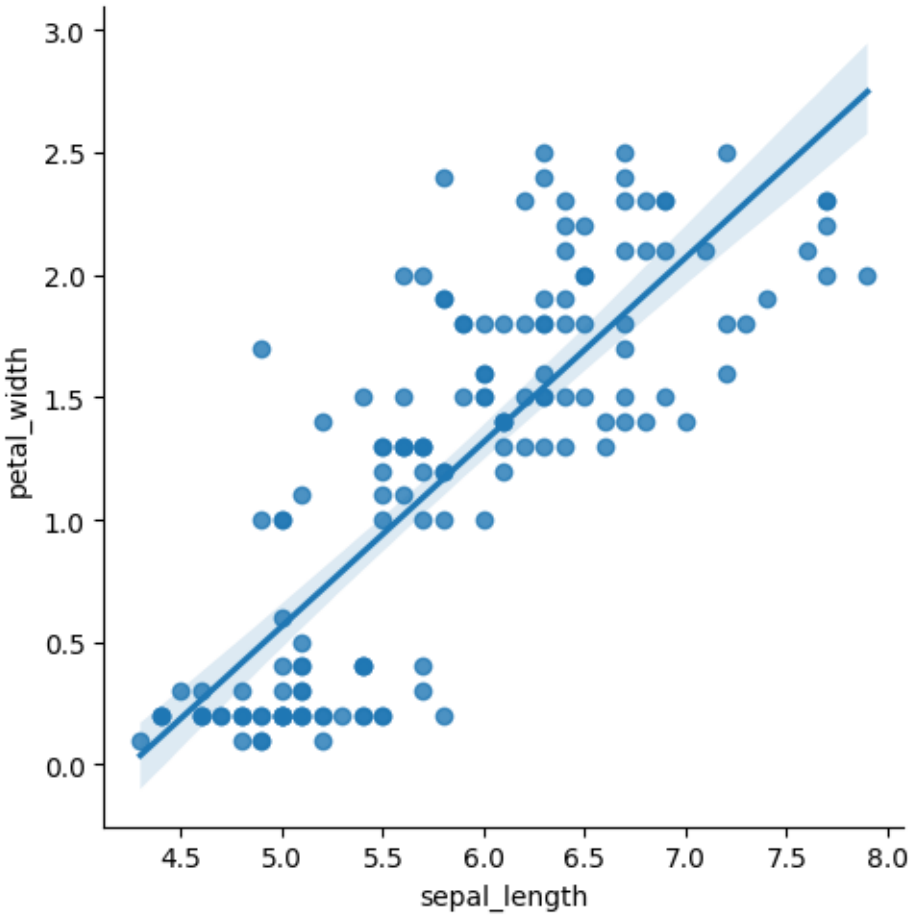
[ ]:
```
sepal_length    0.0
sepal_width     0.0
petal_length    0.0
petal_width     0.0
dtype: float64
```

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

[ ]: 0.0

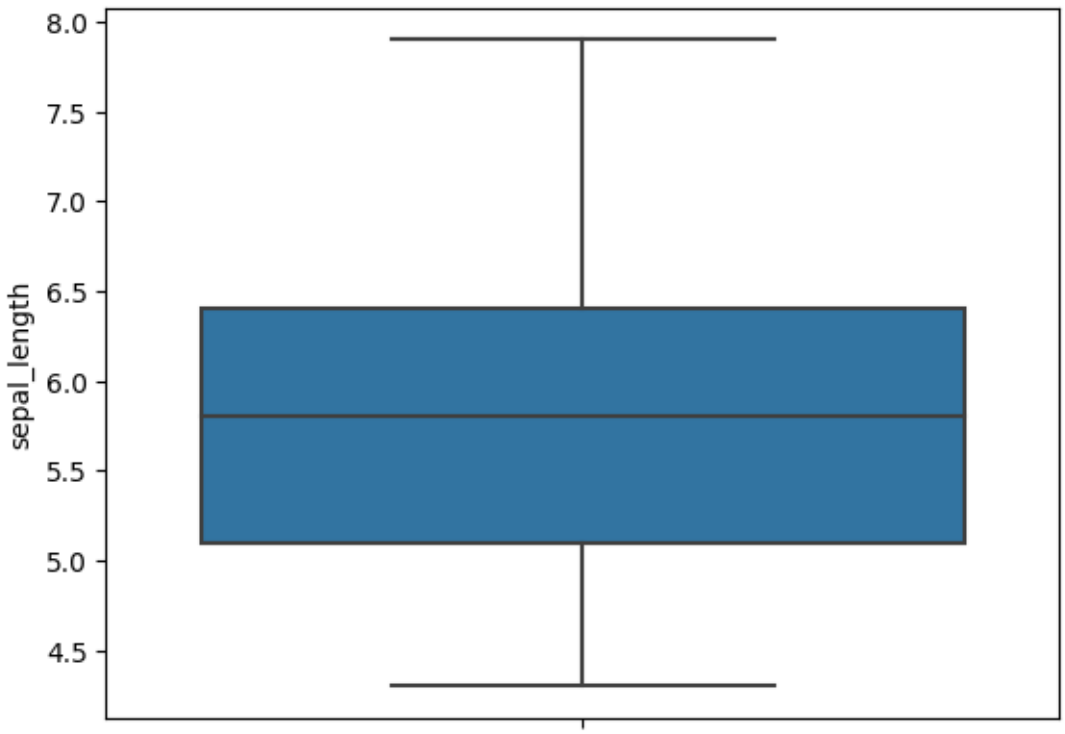
[ ]:
```
sns.lmplot(x="sepal_length", y="petal_width", data=df)
```

[ ]: <seaborn.axisgrid.FacetGrid at 0x1e5cb568d30>



[ ]:
```
sns.boxplot(y="sepal_length", data=df)
```

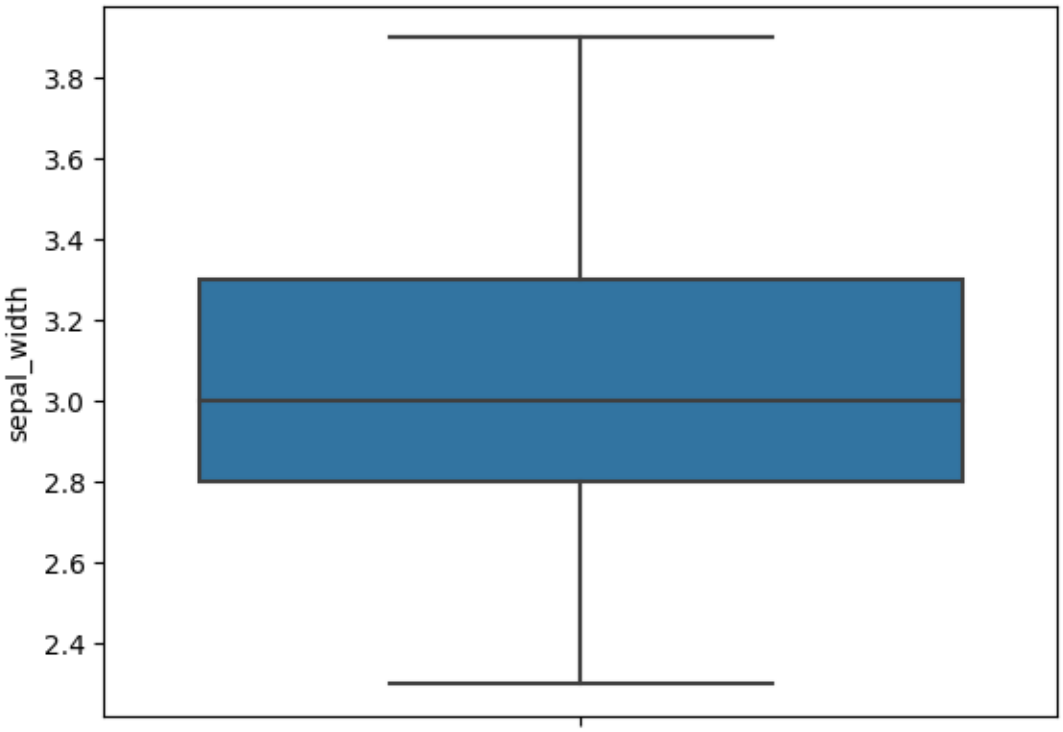
[ ]: <AxesSubplot: ylabel='sepal_length'>



[ ]:
```
df = df[(df["sepal_width"]>2.2)&(df["sepal_width"]<4)]
```

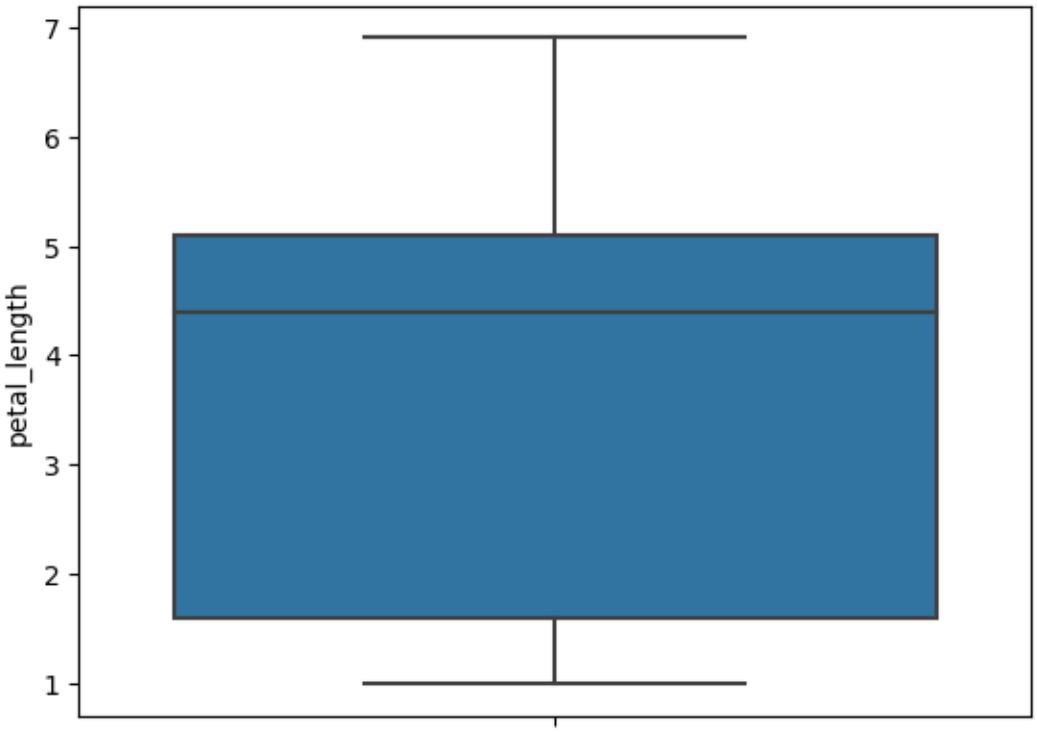
[ ]:
```
sns.boxplot(y="sepal_width", data=df)
```

[ ]:
```
<AxesSubplot: ylabel='sepal_width'>
```



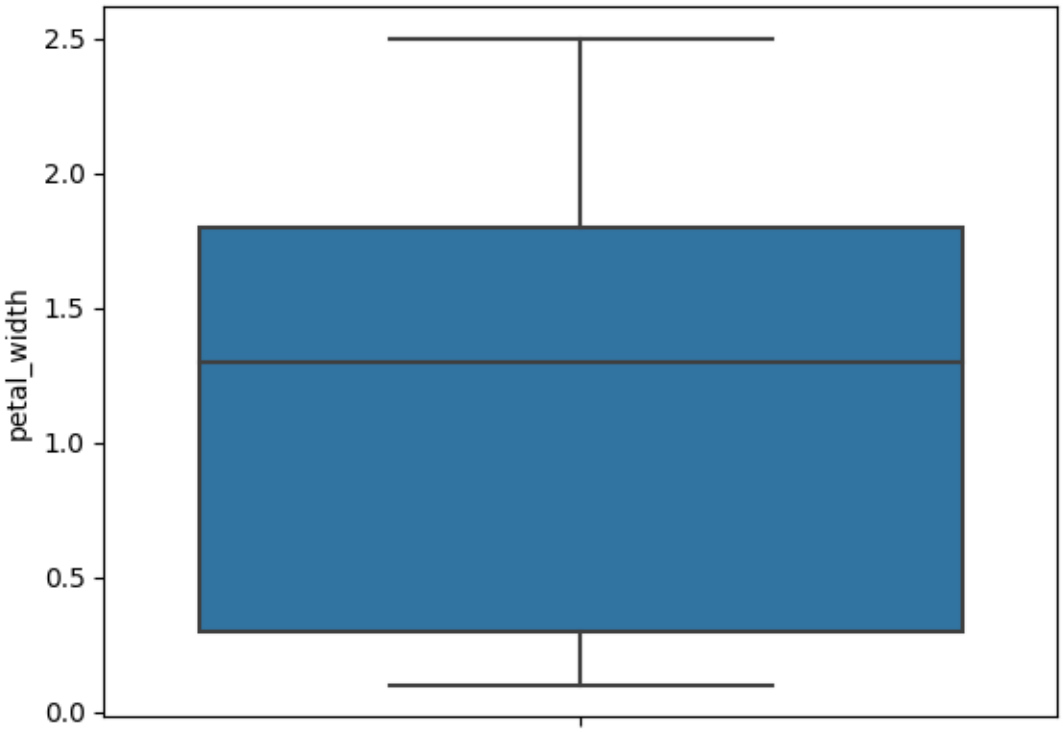
[ ]:
```
sns.boxplot(y="petal_length", data=df)
```

[ ]:
```
<AxesSubplot: ylabel='petal_length'>
```



[ ]:
```
sns.boxplot(y="petal_width", data=df)
```

[ ]:
```
<AxesSubplot: ylabel='petal_width'>
```



[ ]:
```
X = df[["sepal_length","sepal_width","petal_length","petal_width"]]
y = df["species"]
```

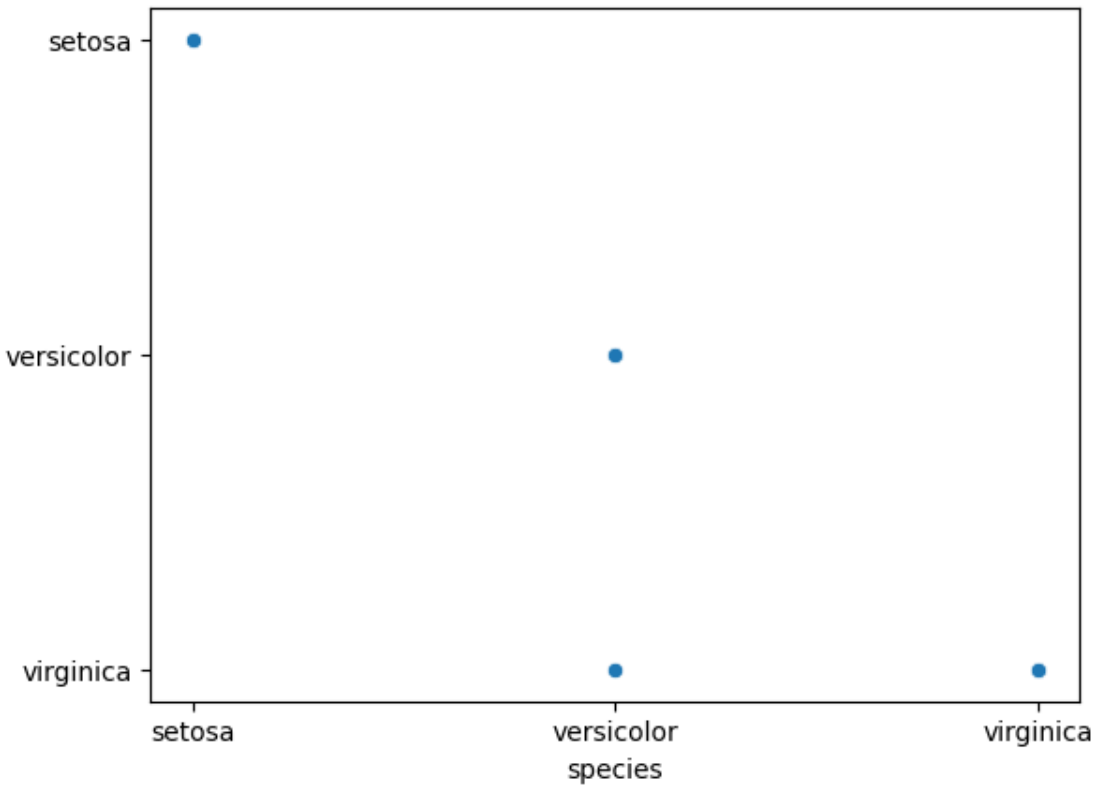
[ ]:
```
from sklearn.model_selection import train_test_split
from sklearn.metrics import accuracy_score
from sklearn.tree import DecisionTreeClassifier

# Prepare data
# X = df[["sepal_length","sepal_width","petal_length"]]
# y = df["petal_width"]

X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2)

# Initialize model
model = DecisionTreeClassifier()

# Fit model to training data
model.fit(X_train, y_train)

# Evaluate model on test data
score = model.score(X_test, y_test)
print("Test score: ", score)
```

```
# Make predictions on new data
predictions = model.predict(X_test)
```

```
Test score:  0.9655172413793104
```

[ ]:
```
#compare
sns.scatterplot(x=y_test, y=predictions)
```

[ ]:
```
<AxesSubplot: xlabel='species'>
```



[ ]:
```
from sklearn.metrics import accuracy_score
```

[ ]:
```
# Generate predictions for the test data
y_pred = model.predict(X_test)

# Compute accuracy
accuracy = accuracy_score(y_test, y_pred)
print("Accuracy: ", accuracy)
```

```
Accuracy:  0.9655172413793104
```