Predicting Iris species using Linear Regression Algorithmn

February 15, 2023

[1]:
```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
import warnings
warnings.filterwarnings('ignore')
```

[4]:
```
df=pd.read_table('https://archive.ics.uci.edu/ml/machine-learning-databases/
↪iris/iris.data',sep=',',header=None,names=['sepal length','sepal␣
↪width','petal length','petal width','class'])
```

[5]:
```
df.head()
```

[5]:

	sepal length	sepal width	petal length	petal width	class
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

[6]:
```
df.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   class         150 non-null    object
dtypes: float64(4), object(1)
memory usage: 6.0+ KB
```

[7]:
```
df.describe()
```

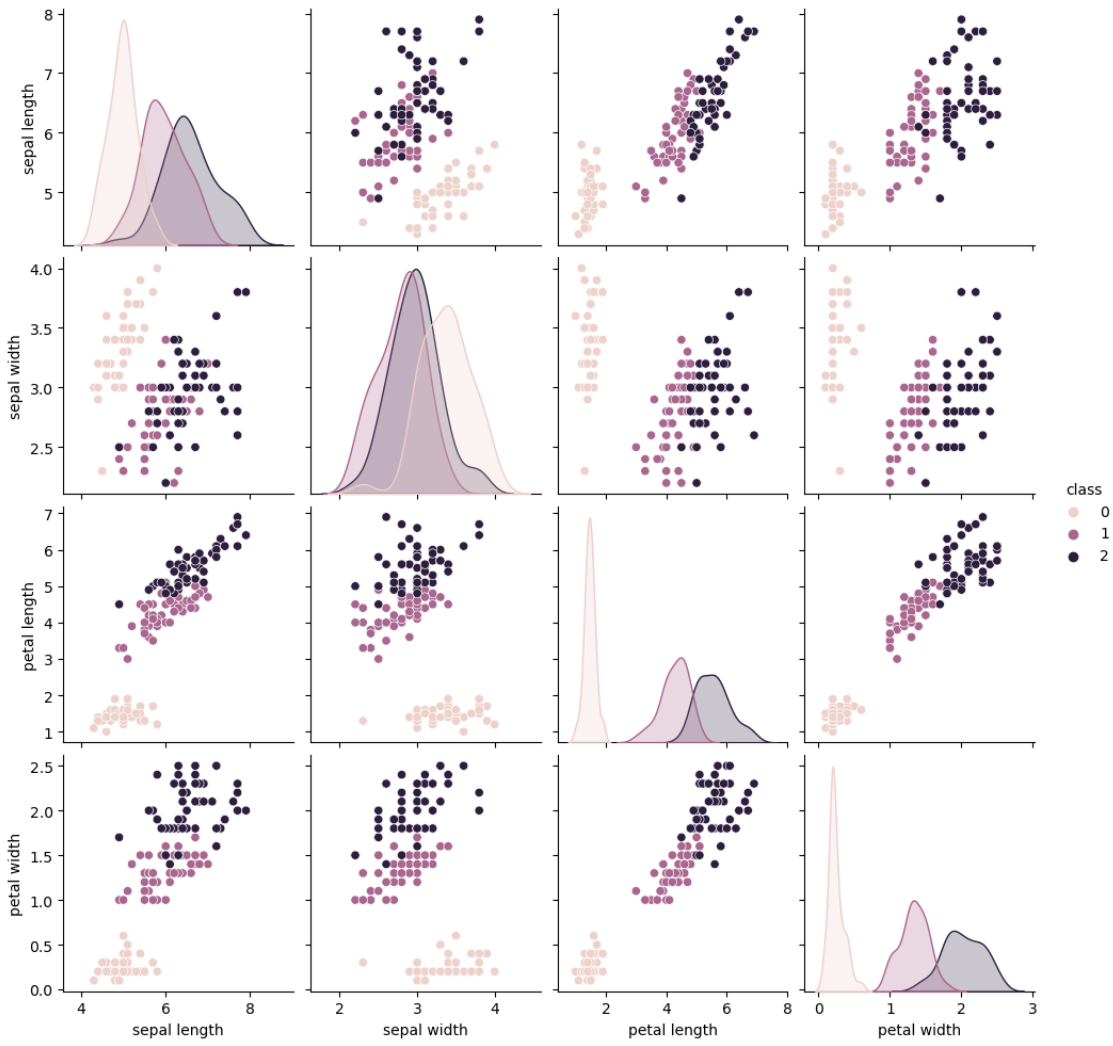
[7]:
```
       sepal length  sepal width  petal length  petal width
count    150.000000   150.000000    150.000000   150.000000
mean       5.843333     3.054000      3.758667     1.198667
std        0.828066     0.433594      1.764420     0.763161
min        4.300000     2.000000      1.000000     0.100000
25%        5.100000     2.800000      1.600000     0.300000
50%        5.800000     3.000000      4.350000     1.300000
75%        6.400000     3.300000      5.100000     1.800000
max        7.900000     4.400000      6.900000     2.500000
```

[8]:
```
df.isna().sum()
```

[8]:
```
sepal length    0
sepal width     0
petal length    0
petal width     0
class           0
dtype: int64
```

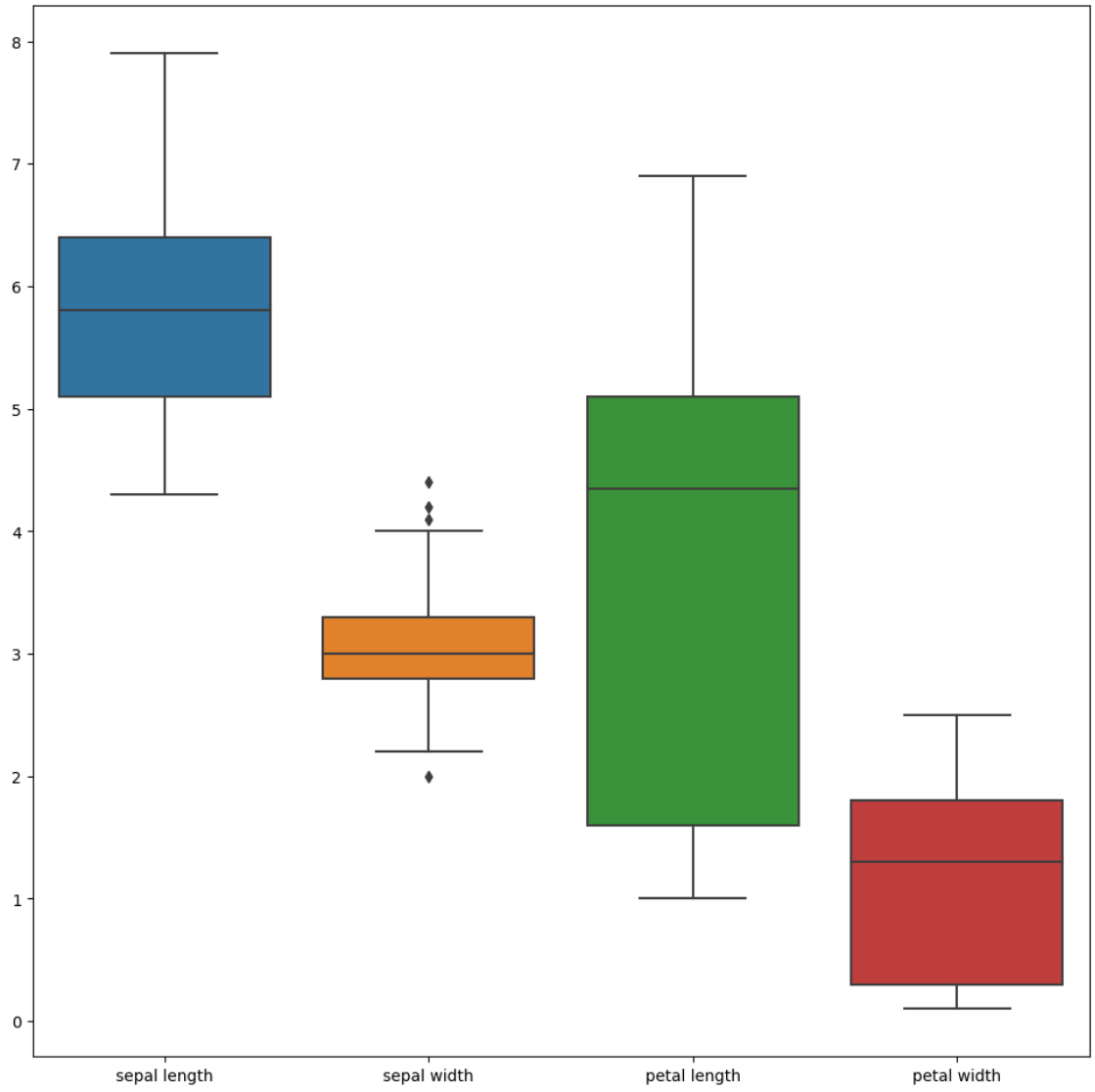
[36]:
```
sns.pairplot(df,hue='class')
```

[36]:
```
<seaborn.axisgrid.PairGrid at 0x20fe8253520>
```



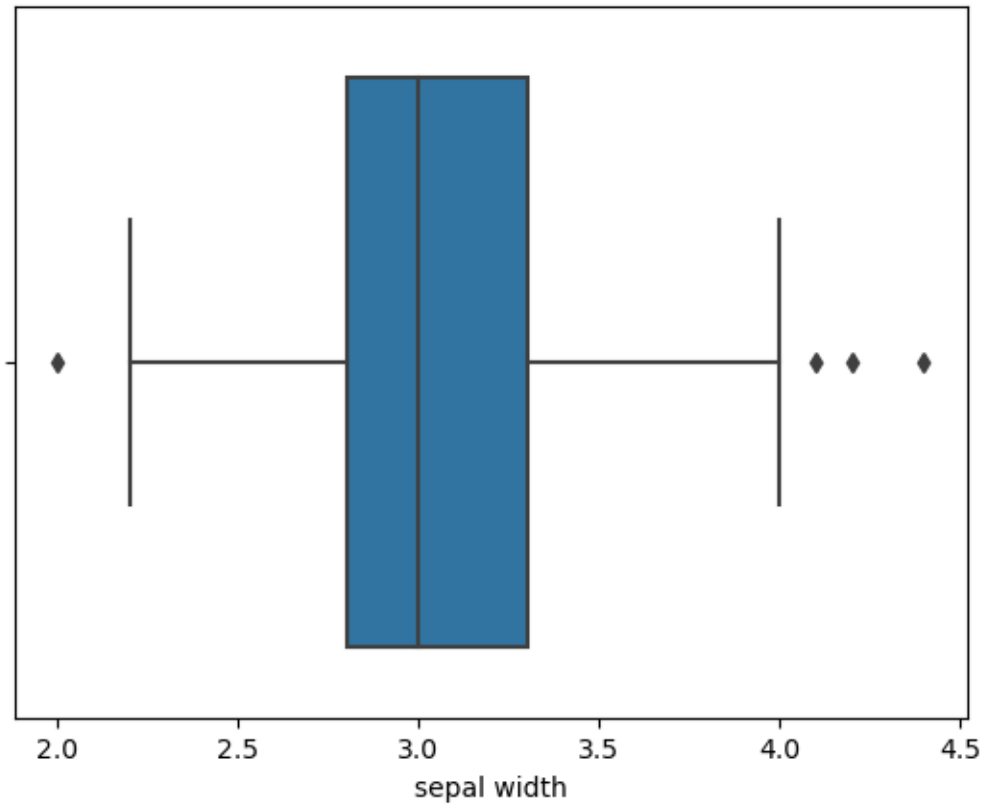
[9]:
```
plt.figure(figsize=(12,12))
sns.boxplot(data=df)
```

[9]: <AxesSubplot:>



[10]:
```
sns.boxplot(x='sepal width',data=df)
```

[10]: <AxesSubplot:xlabel='sepal width'>



[14]:
```
df[df['sepal width']>4.0]
```

[14]:

	sepal length	sepal width	petal length	petal width	class
15	5.7	4.4	1.5	0.4	Iris-setosa
32	5.2	4.1	1.5	0.1	Iris-setosa
33	5.5	4.2	1.4	0.2	Iris-setosa

[15]:
```
df[df['sepal width']<2.1]
```

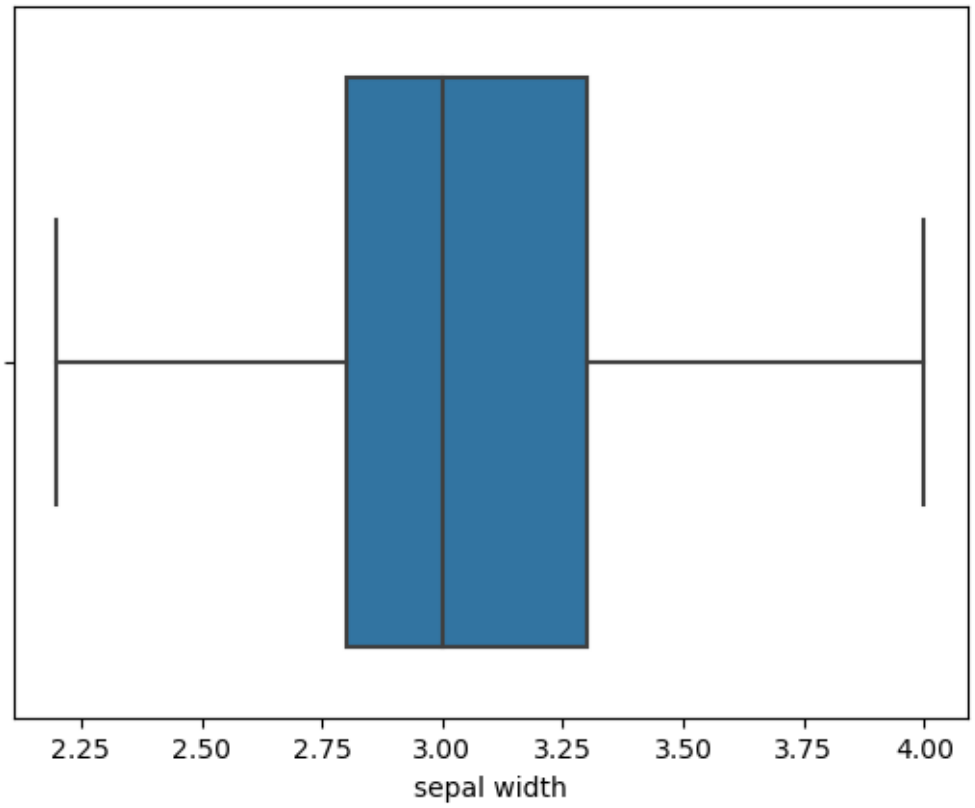
[15]:

	sepal length	sepal width	petal length	petal width	class
60	5.0	2.0	3.5	1.0	Iris-versicolor

[16]:
```
df.drop([15,32,33,60],axis=0,inplace=True)
```

[17]:
```
sns.boxplot(x='sepal width',data=df)
```

[17]:
```
<AxesSubplot:xlabel='sepal width'>
```



[18]:
```
df.head()
```

[18]:

	sepal length	sepal width	petal length	petal width	class
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

[20]:
```
from sklearn.preprocessing import LabelEncoder
```

[21]:
```
le=LabelEncoder()
```

[22]:
```
df['class']=le.fit_transform(df['class'])
```

[23]:
```
df.head()
```

[23]:

	sepal length	sepal width	petal length	petal width	class
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
```

[24]:
```
x=df.iloc[:,:-1]
y=df.iloc[:,-1]
```

[25]:
```
from sklearn.model_selection import train_test_split
```

[26]:
```
xtrain,xtest,ytrain,ytest=train_test_split(x,y,test_size=0.3,random_state=1)
```

[27]:
```
from sklearn.linear_model import LinearRegression
```

[28]:
```
linreg=LinearRegression()
```

[29]:
```
linreg.fit(xtrain,ytrain)
```

[29]:
```
LinearRegression()
```

[30]:
```
ypred=linreg.predict(xtest)
```

[31]:
```
from sklearn.metrics import r2_score
```

[32]:
```
r2=r2_score(ytest,ypred)
```

[33]:
```
print(f'accuracy:{r2}')
```

```
accuracy:0.9207877266545594
```

[34]:
```
train=linreg.score(xtrain,ytrain)
test=linreg.score(xtest,ytest)
print(f'training accuracy:{train}\ntesting accuracy:{test}')
```

```
training accuracy:0.9315971931894438
testing accuracy:0.9207877266545594
```

[ ]:
```

```