day_18_Assignment01

January 19, 2023

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

1 Multilinear Regression 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 LinearRegression
```

```
[ ]: 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
```

```
[ ]: df_ml = df[["sepal_length","sepal_width","petal_length","petal_width"]]
```

```
[ ]: df_ml.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 = df_ml[["sepal_length","sepal_width","petal_length"]]
     y = df_ml[["petal_width"]]
```

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

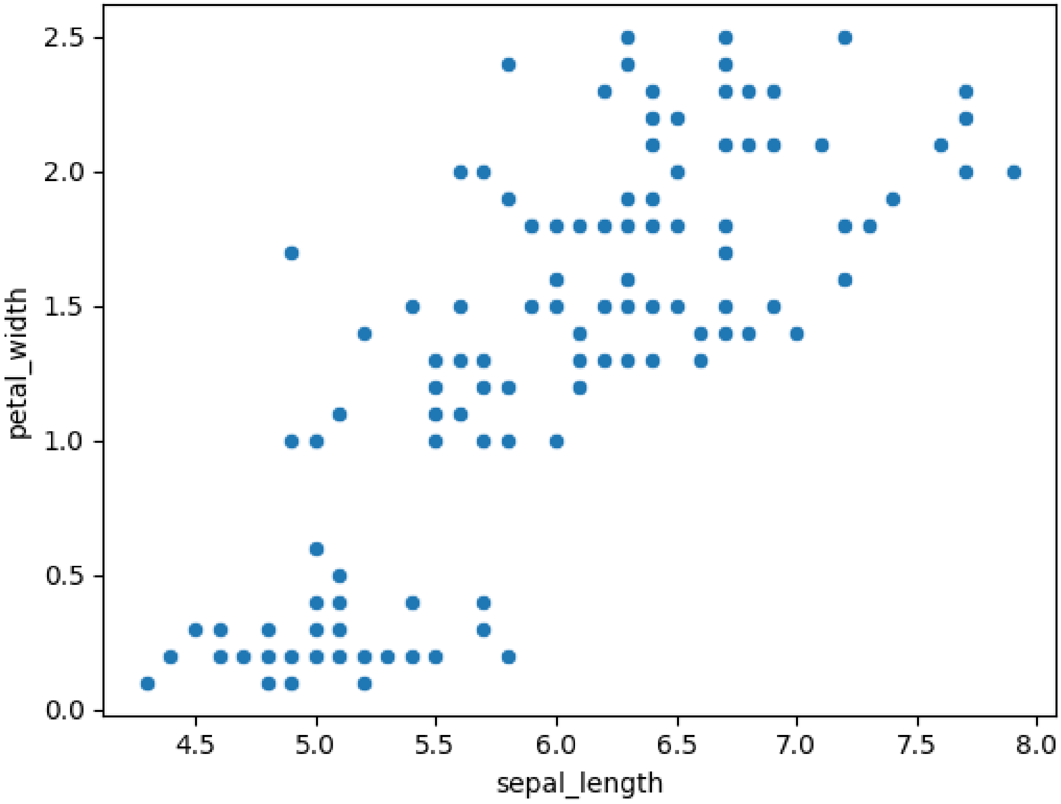
```
[ ]: sepal_length    0.0
     sepal_width     0.0
     petal_length    0.0
     dtype: float64
```

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

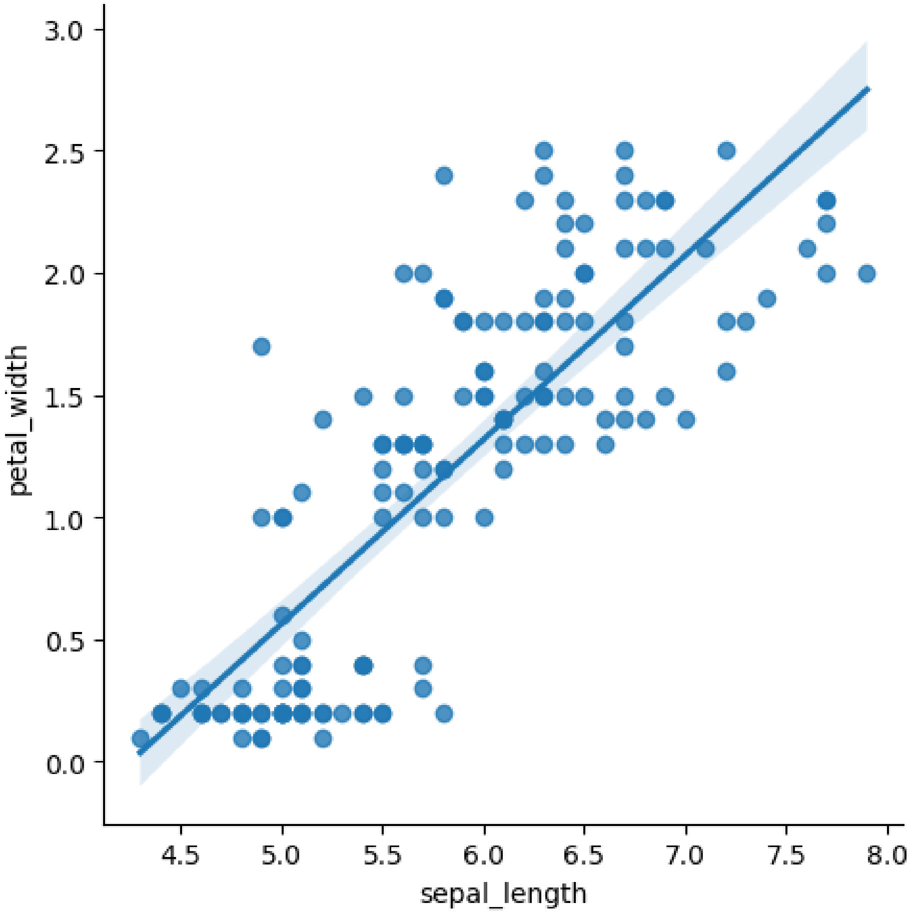
```
[ ]: petal_width    0.0
     dtype: float64
```

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

```
[ ]: <AxesSubplot: xlabel='sepal_length', ylabel='petal_width'>
```

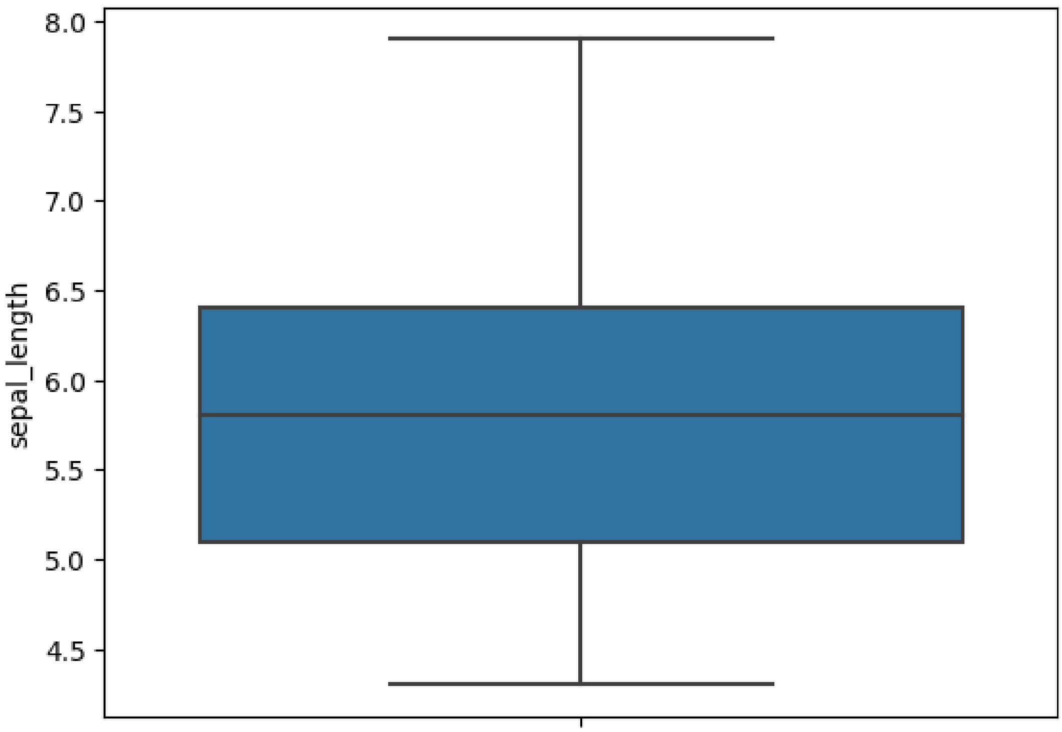


```
[ ]: sns.lmplot(x="sepal_length", y="petal_width", data=df)
[ ]: <seaborn.axisgrid.FacetGrid at 0x242387b0280>
```



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

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

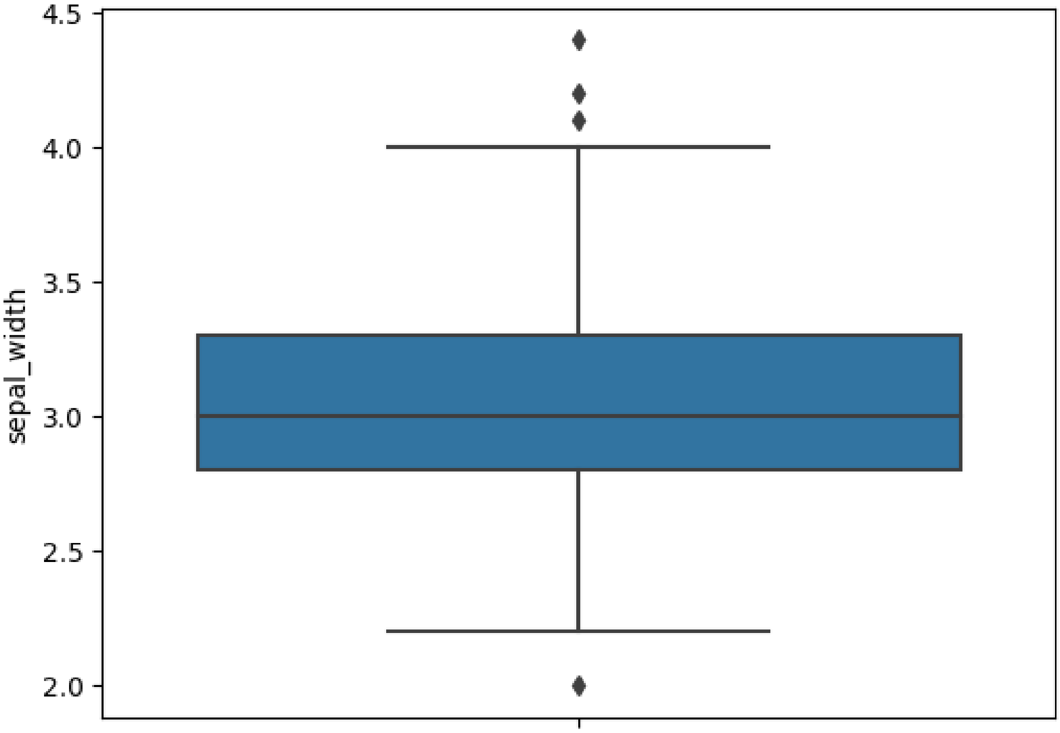


```
[ ]: df.columns
```

```
[ ]: Index(['sepal_length', 'sepal_width', 'petal_length', 'petal_width',
            'species'],
           dtype='object')
```

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

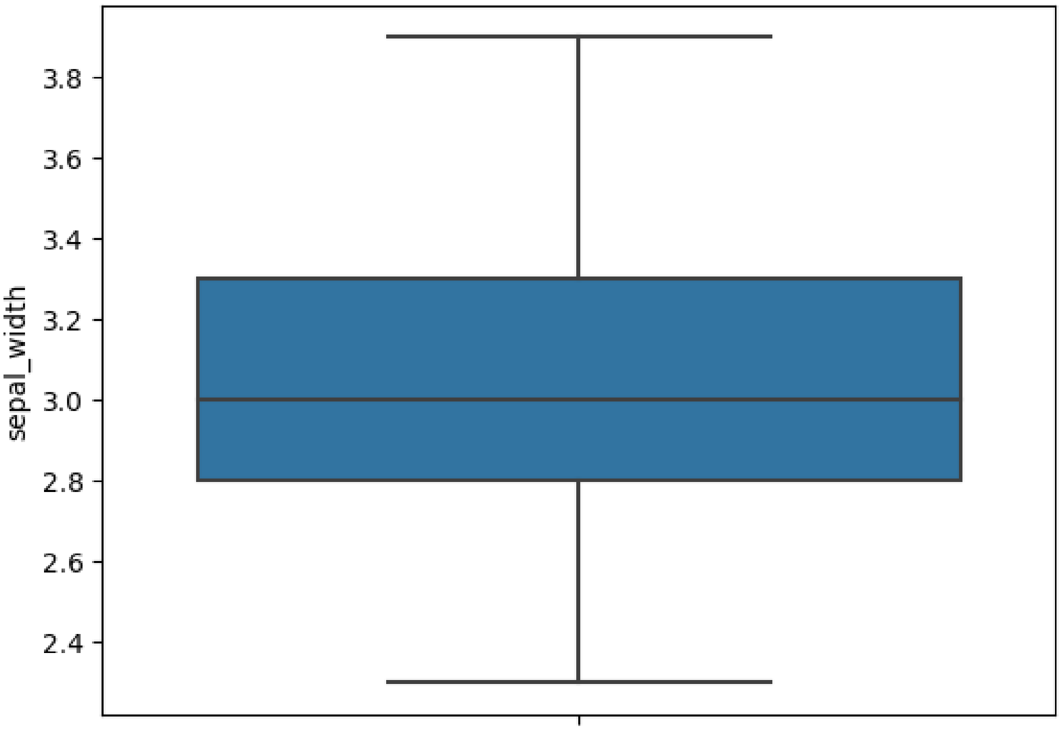
```
[ ]: <AxesSubplot: ylabel='sepal_width'>
```



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

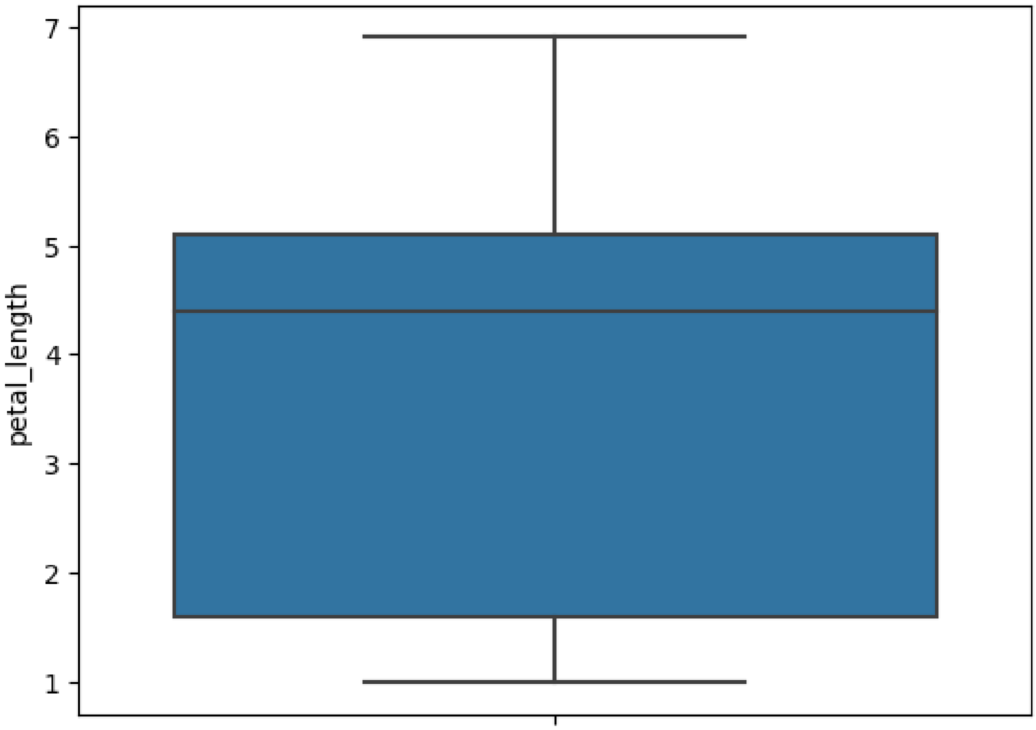
[ ]:
```
sns.boxplot(y="sepal_width", data=df_ml)
```

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



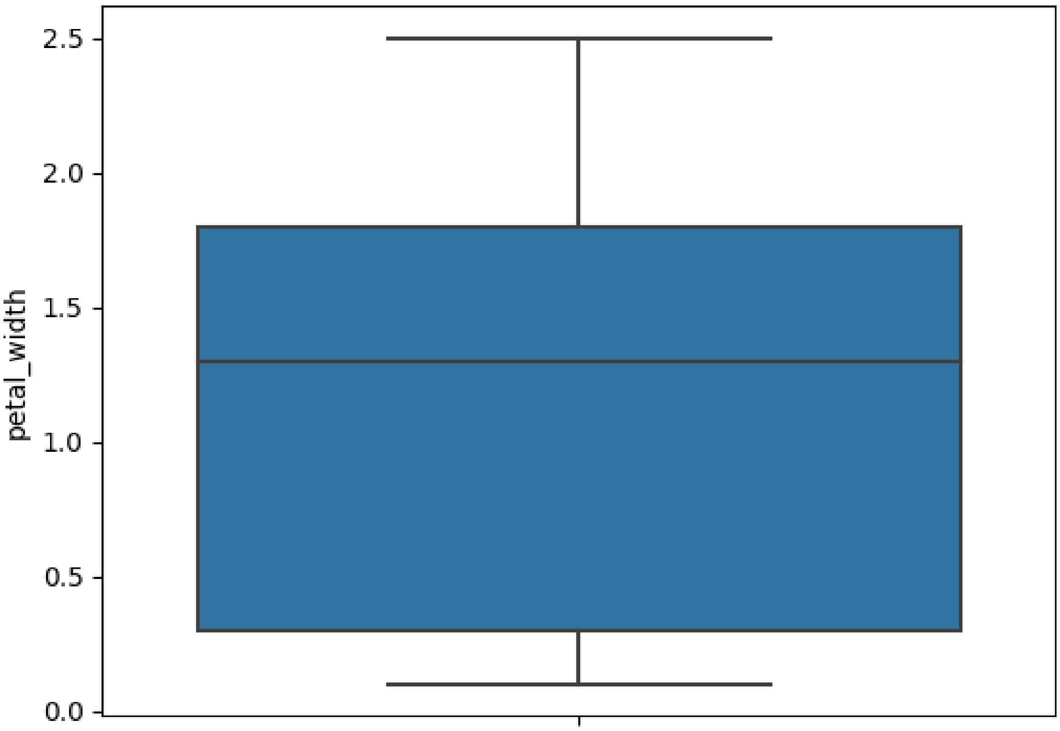
[ ]:
```
sns.boxplot(y="petal_length", data=df_ml)
```

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



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

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



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

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

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

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

# Initialize model
model = LinearRegression()

# 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
```

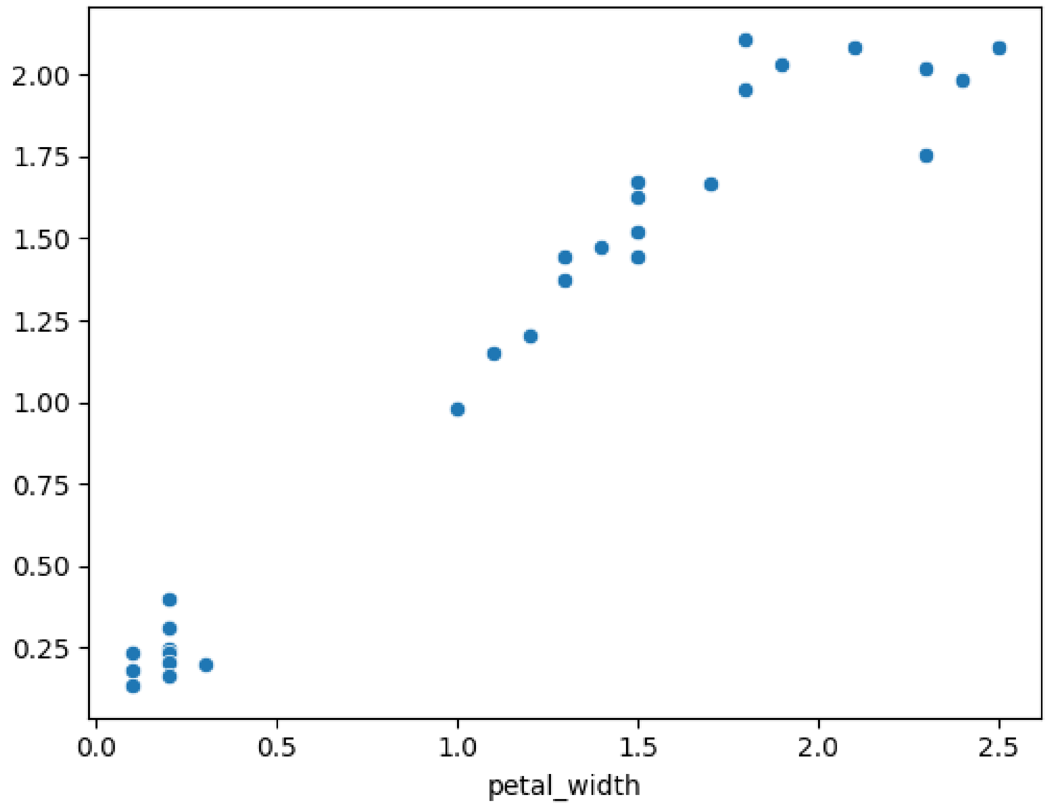
```
predictions = model.predict(X_test)
```

```
Test score:  0.944449575172331
```

[ ]:

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

[ ]: <AxesSubplot: xlabel='petal_width'>



[ ]:

```
from sklearn.metrics import mean_squared_error,mean_absolute_error,r2_score
MAE = mean_absolute_error(y_true=y_test, y_pred=predictions)
MSE = mean_squared_error(y_true=y_test, y_pred=predictions)
RMSE = mean_squared_error(y_true=y_test, y_pred=predictions, squared = False)
R2= r2_score(y_true=y_test, y_pred=predictions)

print("MAE: ", MAE)
print("MSE: ", MSE)
print("RMSE: ", RMSE)
print("R2_squared:", R2)
```

```
MAE:  0.13260928149350626
MSE:  0.03582440952382236
```

```
RMSE:  0.18927337246380527
R2_squared: 0.944449575172331
```

[ ]:

```
```