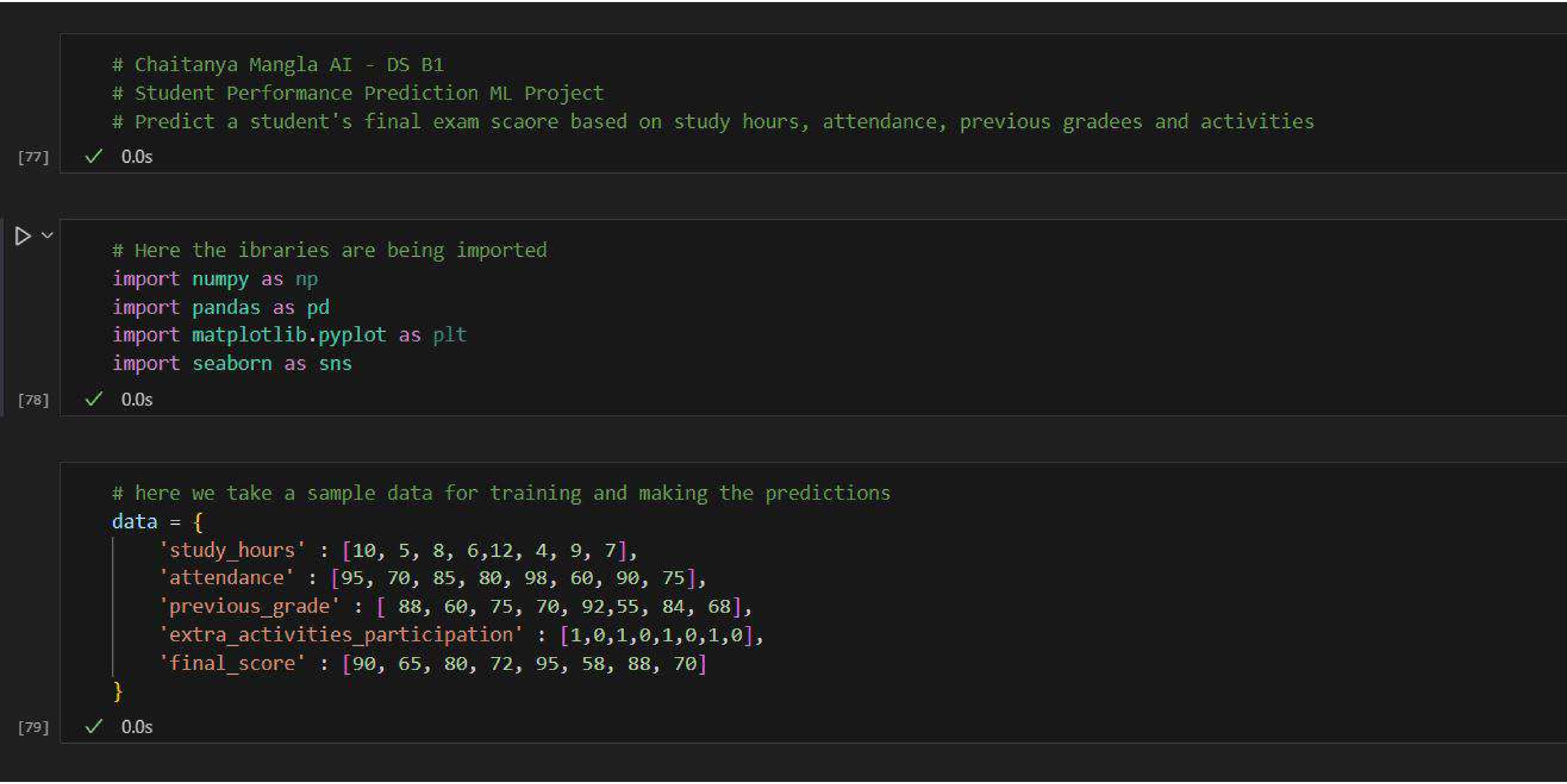


```
# Chaitanya Mangla AI - DS B1
# Student Performance Prediction ML Project
# Predict a student's final exam scaore based on study hours, attendance, previous gradees and activities
```

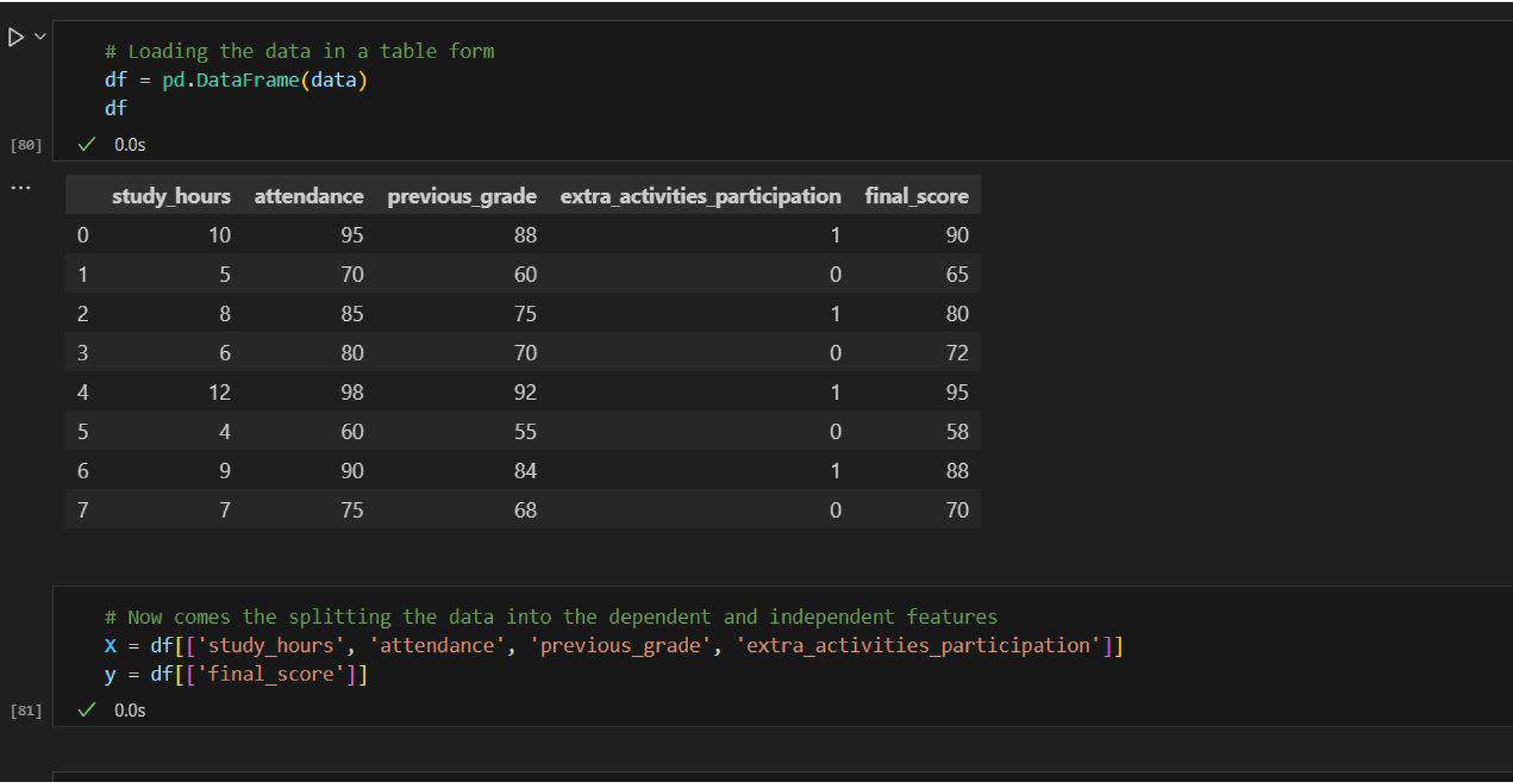
[77] ✓ 0.0s

```
# Here the ibraries are being imported
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
```

[78] ✓ 0.0s

```
# here we take a sample data for training and making the predictions
data = {
    'study_hours' : [10, 5, 8, 6,12, 4, 9, 7],
    'attendance' : [95, 70, 85, 80, 98, 60, 90, 75],
    'previous_grade' : [ 88, 60, 75, 70, 92,55, 84, 68],
    'extra_activities_participation' : [1,0,1,0,1,0,1,0],
    'final_score' : [90, 65, 80, 72, 95, 58, 88, 70]
}
```

[79] ✓ 0.0s



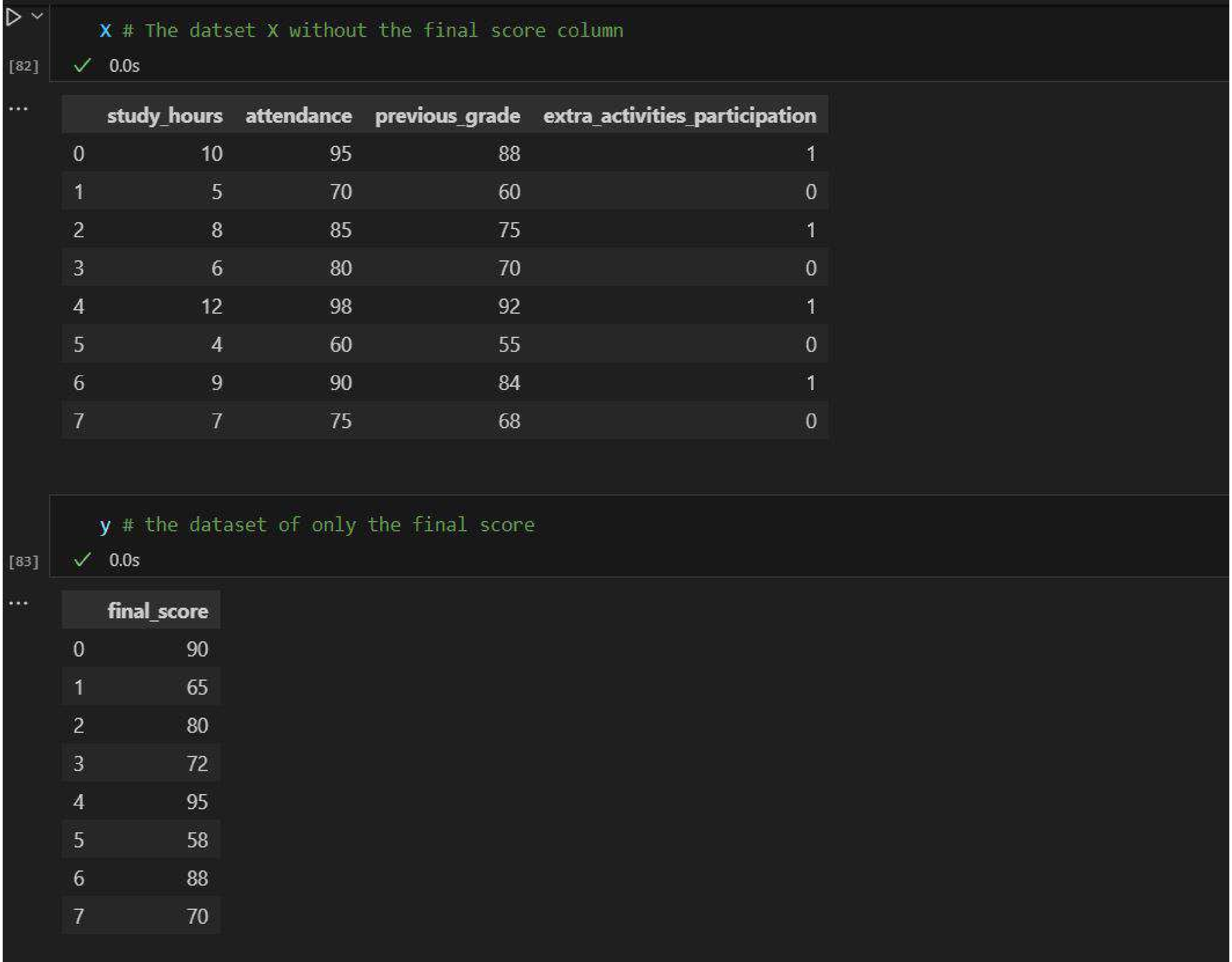
```python
# Loading the data in a table form
df = pd.DataFrame(data)
df
```

[80] ✓ 0.0s

	study_hours	attendance	previous_grade	extra_activities_participation	final_score
0	10	95	88	1	90
1	5	70	60	0	65
2	8	85	75	1	80
3	6	80	70	0	72
4	12	98	92	1	95
5	4	60	55	0	58
6	9	90	84	1	88
7	7	75	68	0	70

```python
# Now comes the splitting the data into the dependent and independent features
X = df[['study_hours', 'attendance', 'previous_grade', 'extra_activities_participation']]
y = df[['final_score']]
```

[81] ✓ 0.0s



```
X # The datset X without the final score column
```

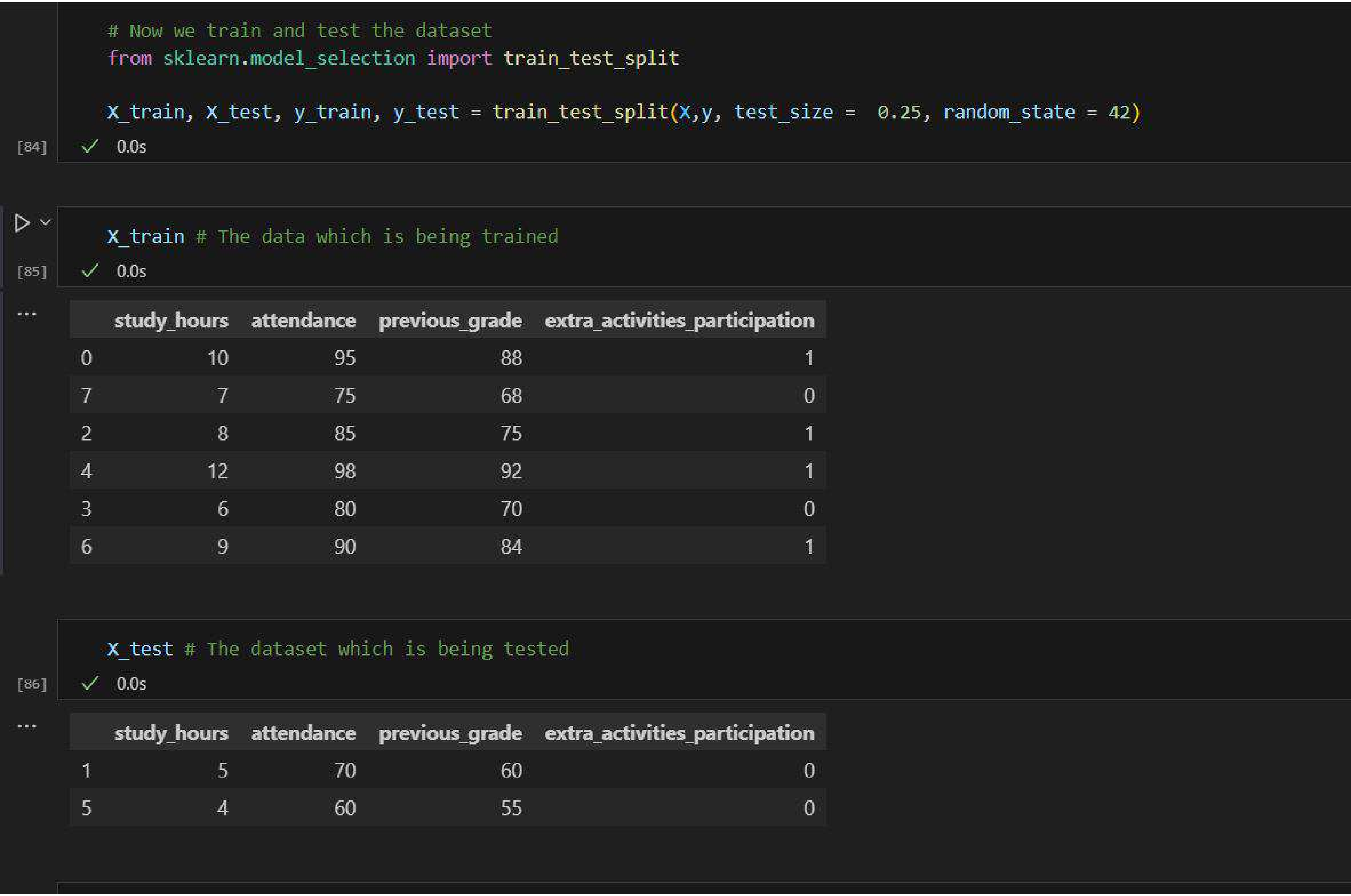
[82] ✓ 0.0s

	study_hours	attendance	previous_grade	extra_activities_participation
0	10	95	88	1
1	5	70	60	0
2	8	85	75	1
3	6	80	70	0
4	12	98	92	1
5	4	60	55	0
6	9	90	84	1
7	7	75	68	0

```
y # the dataset of only the final score
```

[83] ✓ 0.0s

	final_score
0	90
1	65
2	80
3	72
4	95
5	58
6	88
7	70



```python
# Now we train and test the dataset
from sklearn.model_selection import train_test_split

X_train, X_test, y_train, y_test = train_test_split(X,y, test_size =  0.25, random_state = 42)
```

[84] ✓ 0.0s

```python
X_train # The data which is being trained
```

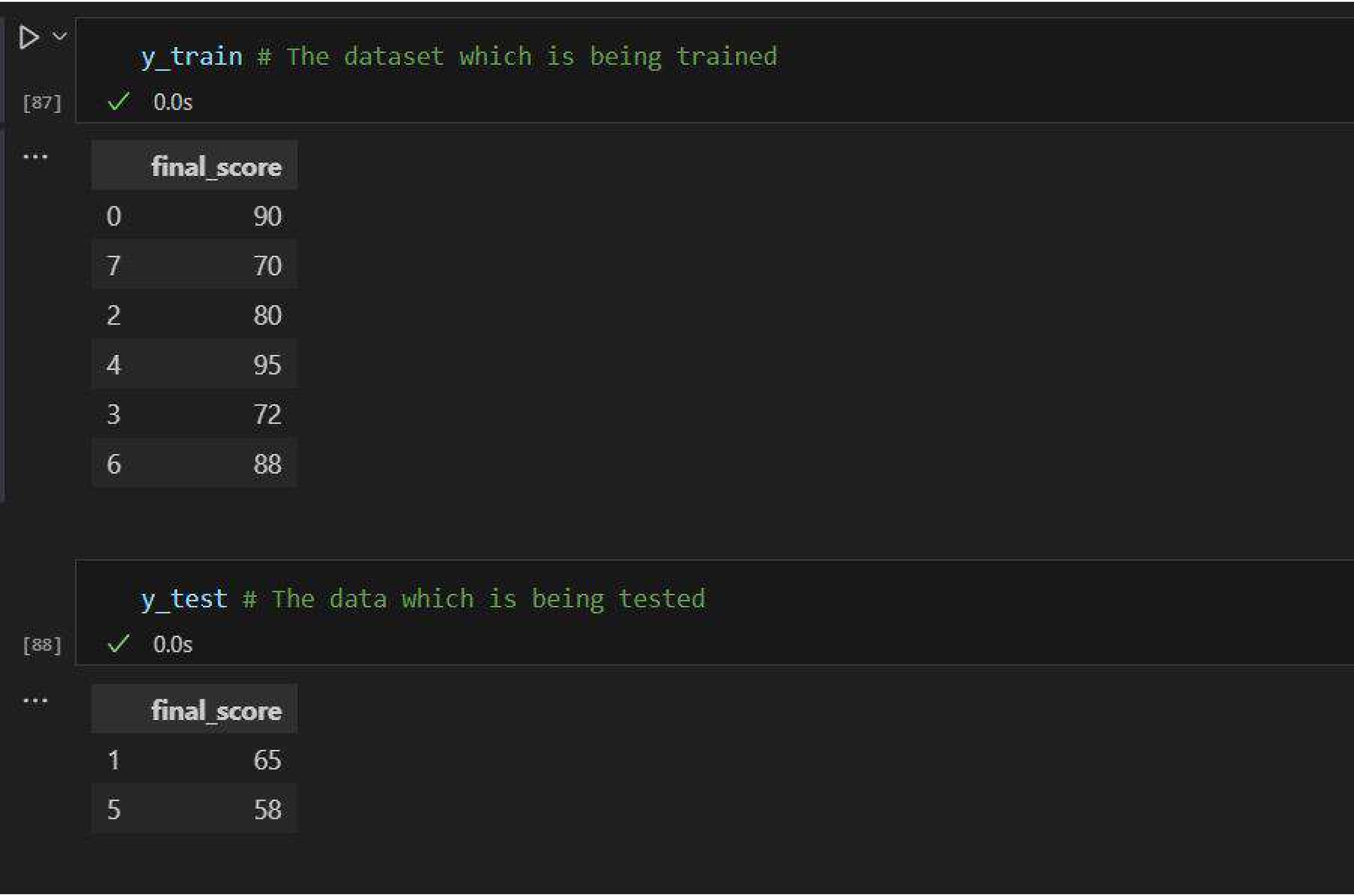
[85] ✓ 0.0s

	study_hours	attendance	previous_grade	extra_activities_participation
0	10	95	88	1
7	7	75	68	0
2	8	85	75	1
4	12	98	92	1
3	6	80	70	0
6	9	90	84	1

```python
X_test # The dataset which is being tested
```

[86] ✓ 0.0s

	study_hours	attendance	previous_grade	extra_activities_participation
1	5	70	60	0
5	4	60	55	0



```
y_train # The dataset which is being trained
```

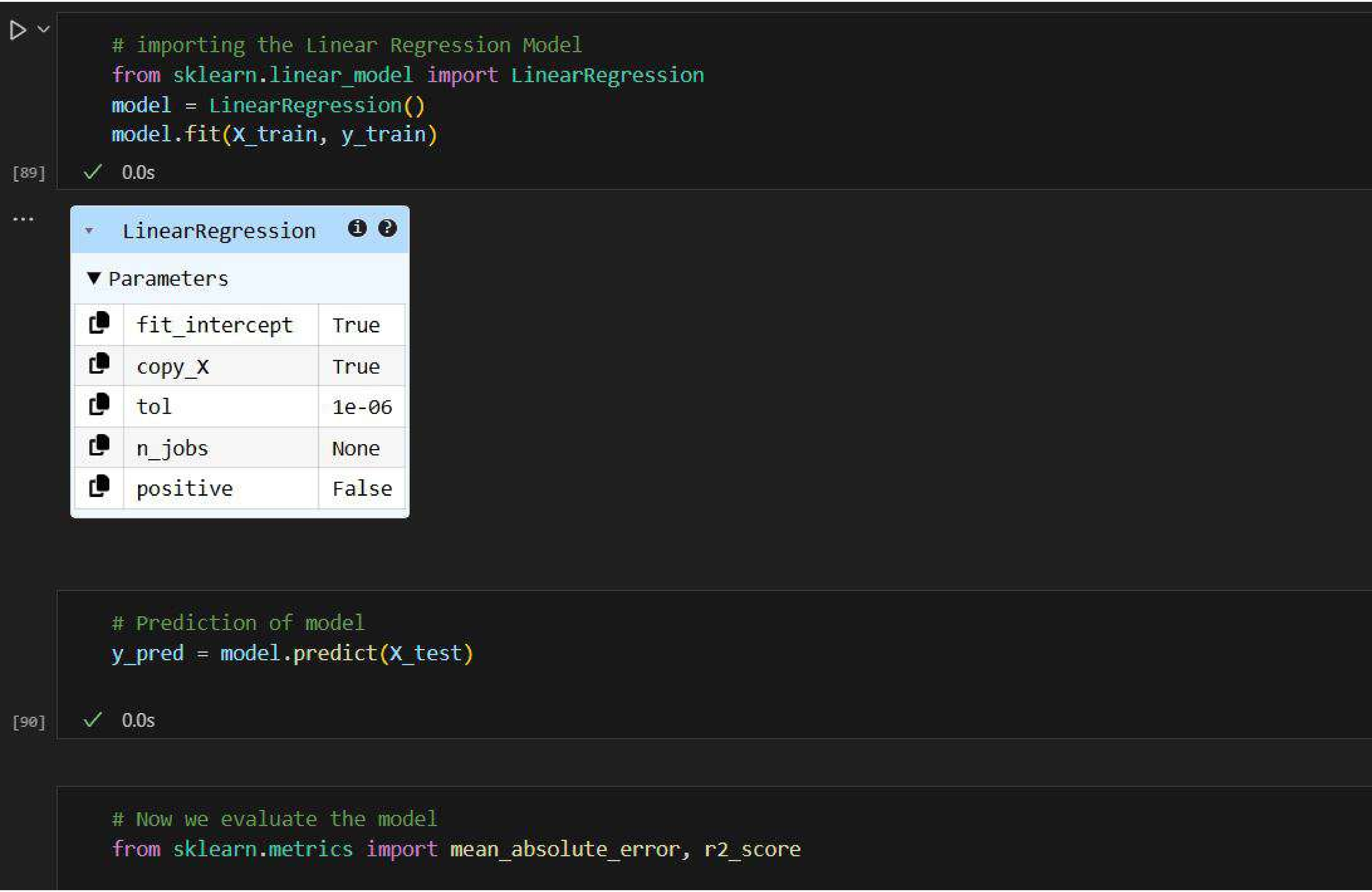
[87] ✓ 0.0s

```
   final_score
0           90
7           70
2           80
4           95
3           72
6           88
```

```
y_test # The data which is being tested
```

[88] ✓ 0.0s

```
   final_score
1           65
5           58
```



```python
# importing the Linear Regression Model
from sklearn.linear_model import LinearRegression
model = LinearRegression()
model.fit(X_train, y_train)
```

[89] ✓ 0.0s

LinearRegression

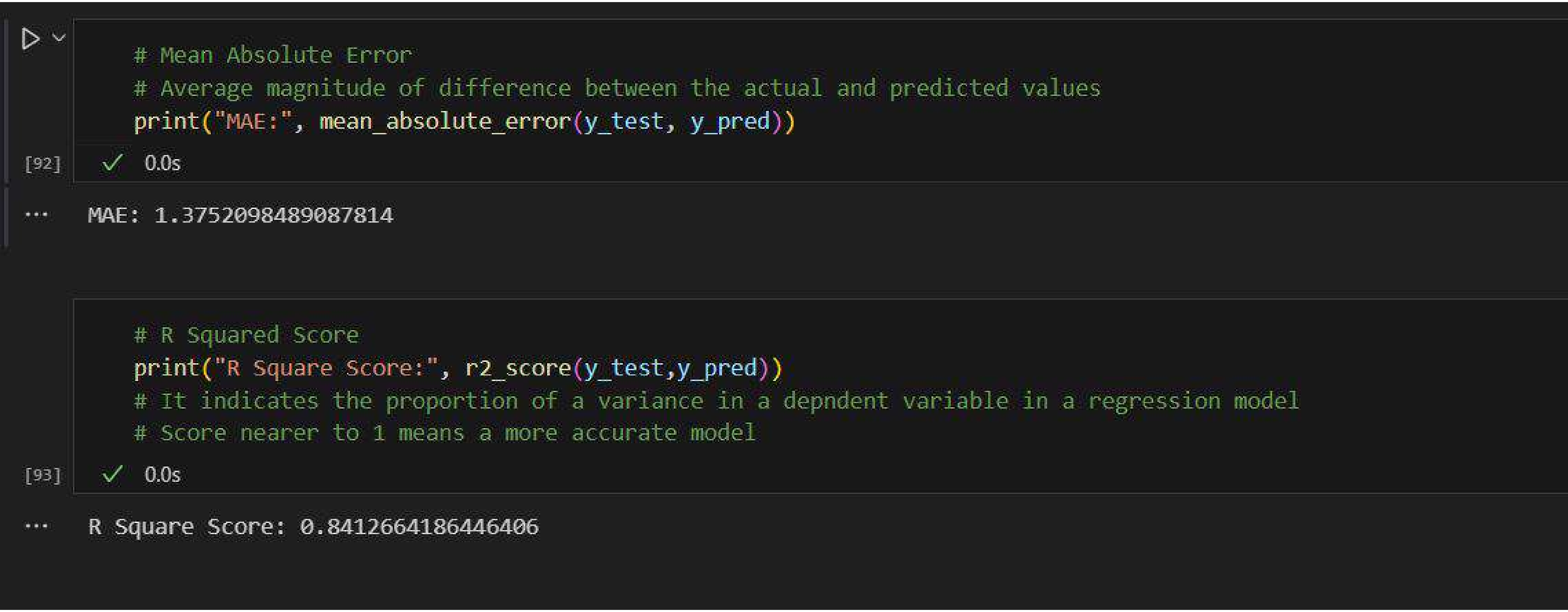
▼ Parameters

	fit_intercept	True
	copy_X	True
	tol	1e-06
	n_jobs	None
	positive	False

```python
# Prediction of model
y_pred = model.predict(X_test)
```

[90] ✓ 0.0s

```python
# Now we evaluate the model
from sklearn.metrics import mean_absolute_error, r2_score
```

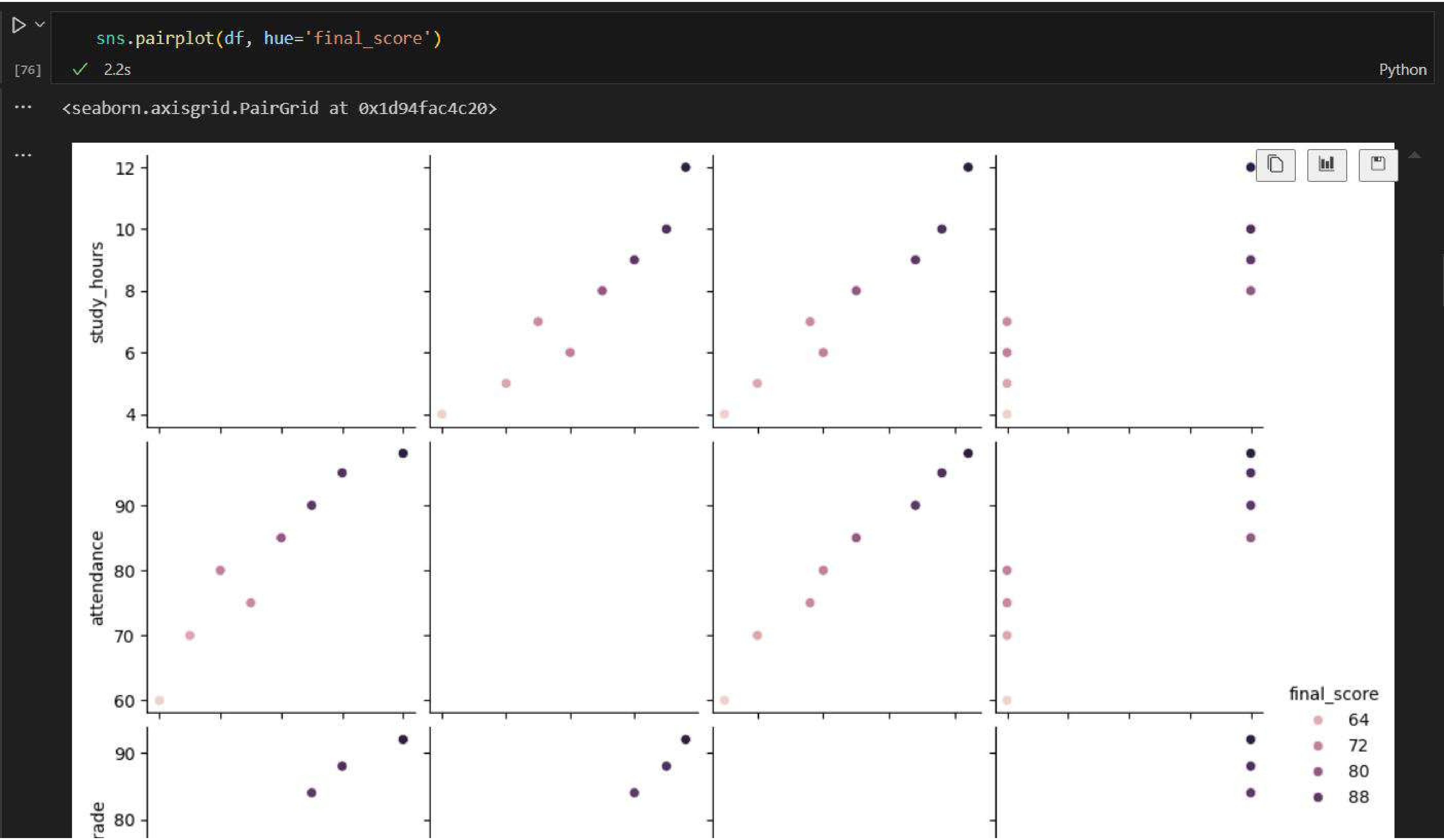


```
# Mean Absolute Error
# Average magnitude of difference between the actual and predicted values
print("MAE:", mean_absolute_error(y_test, y_pred))
```

```
MAE: 1.3752098489087814
```

```
# R Squared Score
print("R Square Score:", r2_score(y_test,y_pred))
# It indicates the proportion of a variance in a depndent variable in a regression model
# Score nearer to 1 means a more accurate model
```

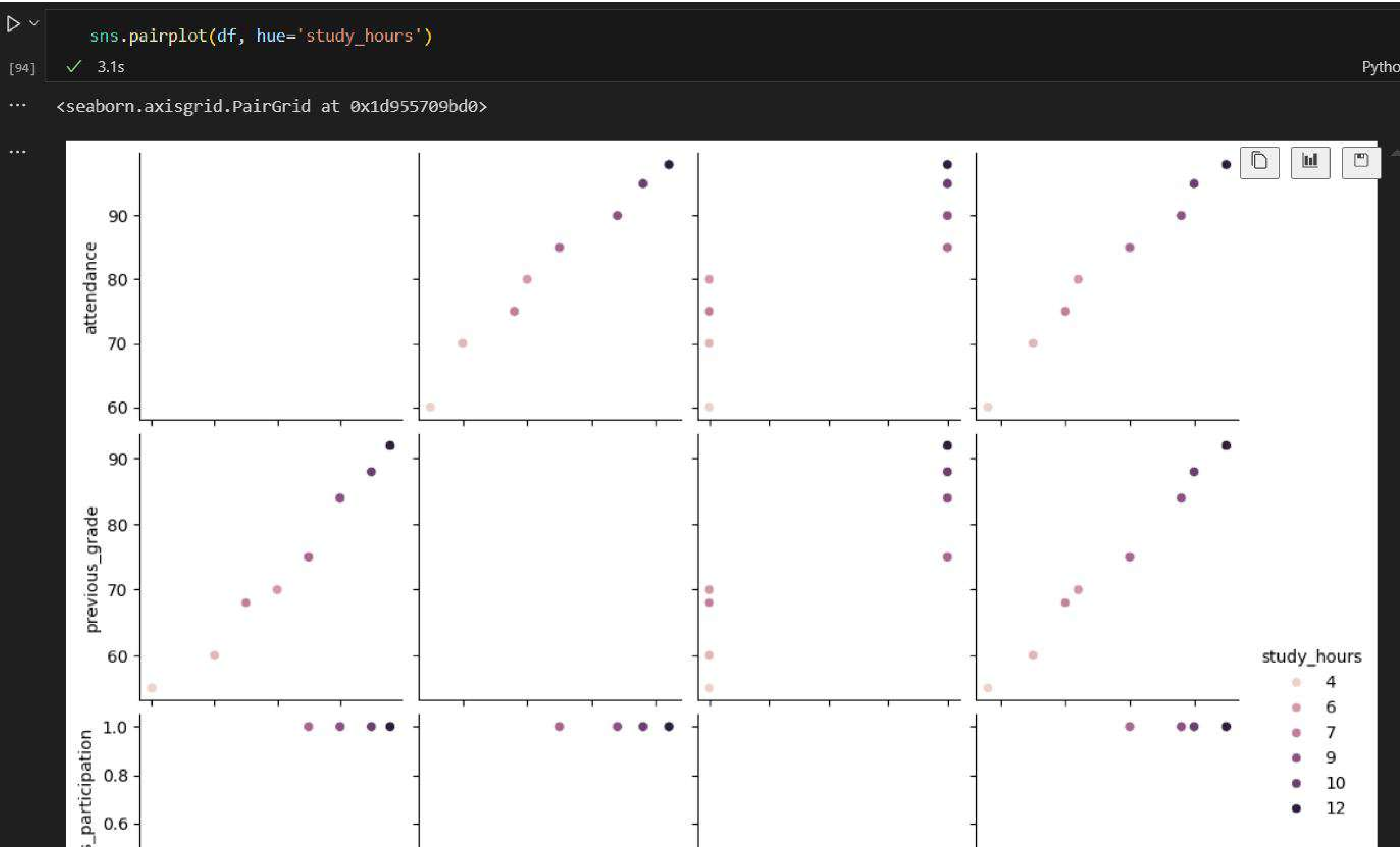
```
R Square Score: 0.8412664186446406
```



```
sns.pairplot(df, hue='final_score')
```

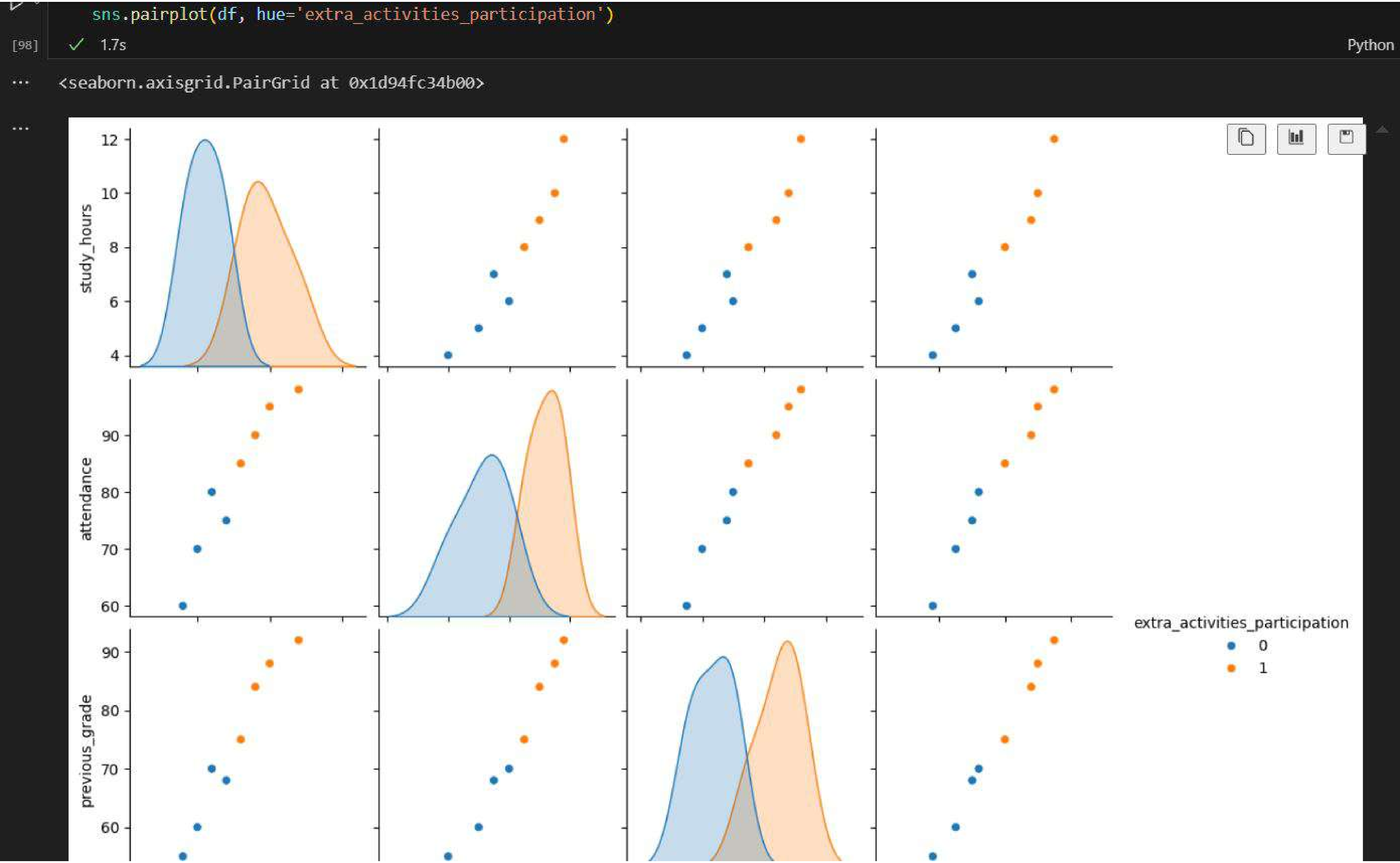
[76] ✓ 2.2s Python

```
<seaborn.axisgrid.PairGrid at 0x1d94fac4c20>
```



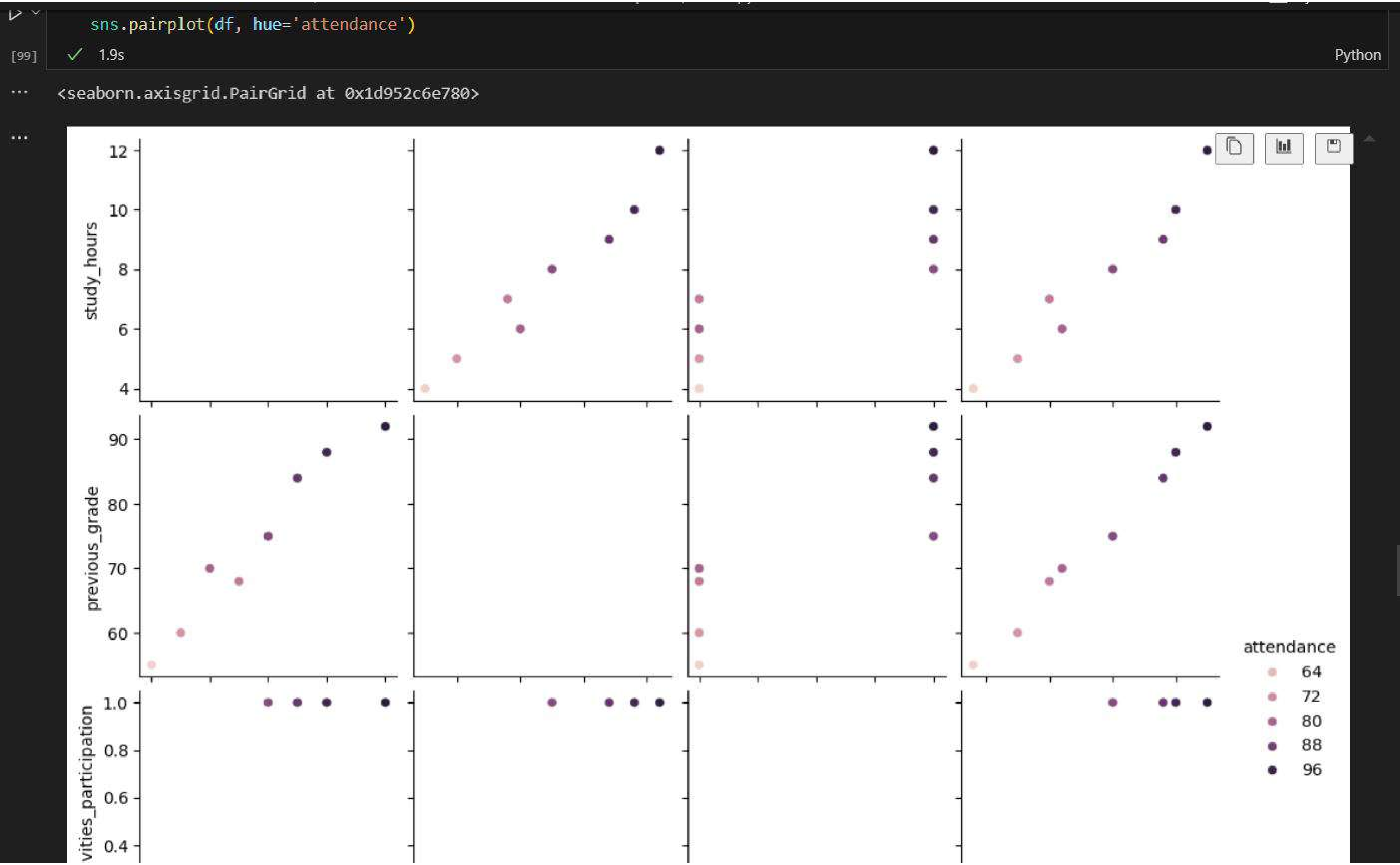
```
sns.pairplot(df, hue='study_hours')
```

```
<seaborn.axisgrid.PairGrid at 0x1d955709bd0>
```



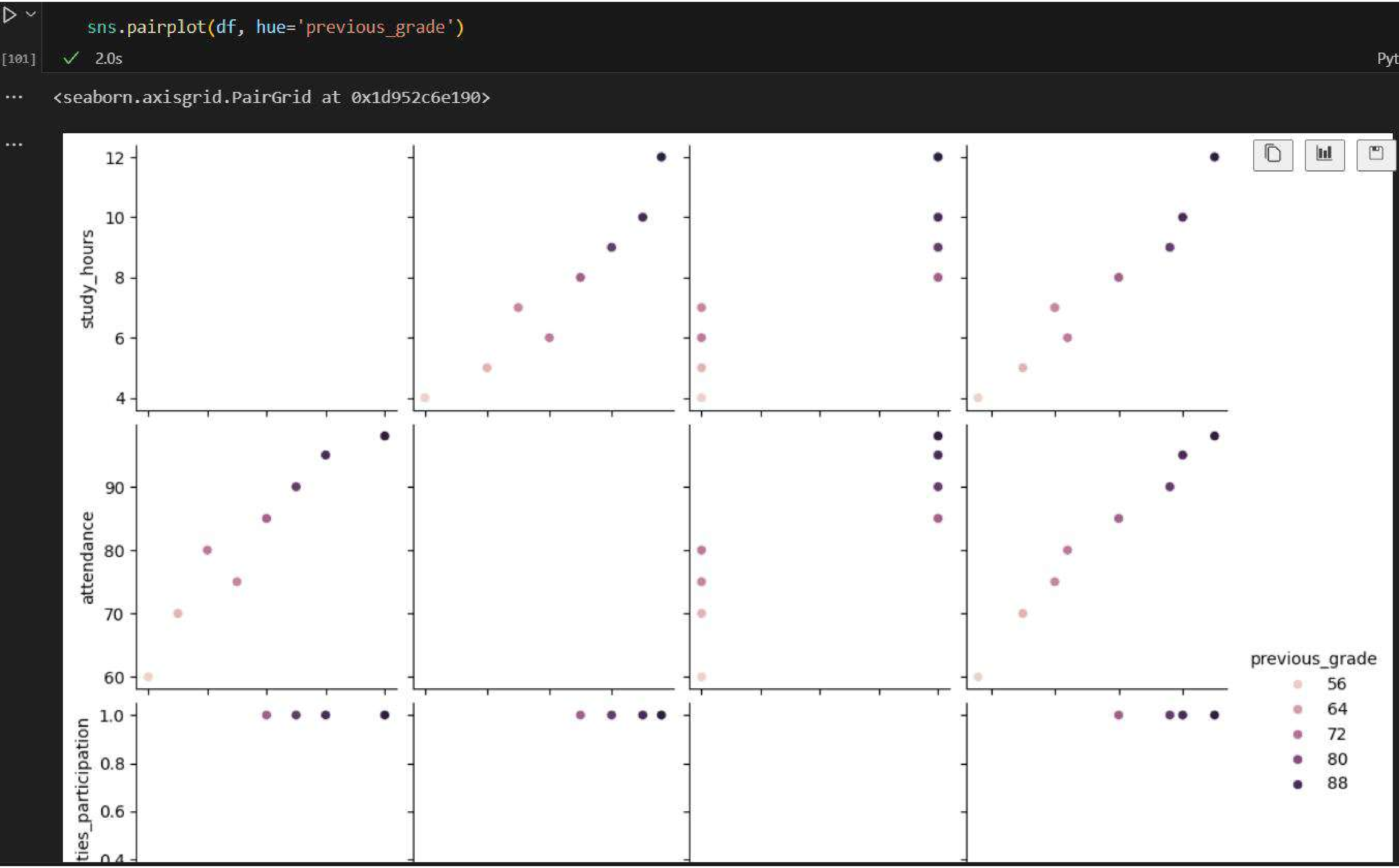
```
sns.pairplot(df, hue='extra_activities_participation')
```

```
<seaborn.axisgrid.PairGrid at 0x1d94fc34b00>
```



```
sns.pairplot(df, hue='attendance')
```

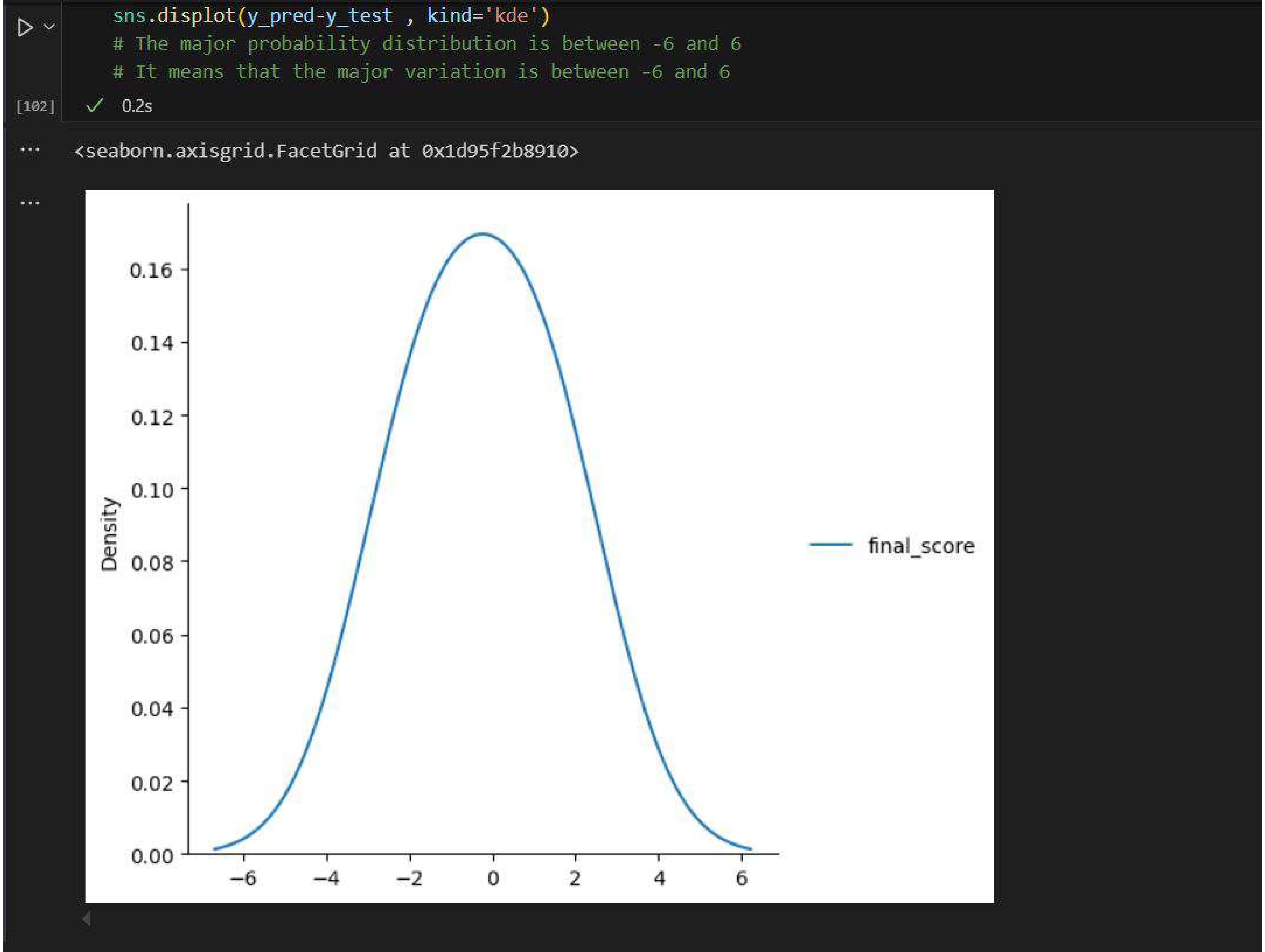
```
<seaborn.axisgrid.PairGrid at 0x1d952c6e780>
```



```
[101]
sns.pairplot(df, hue='previous_grade')
```

✓ 2.0s

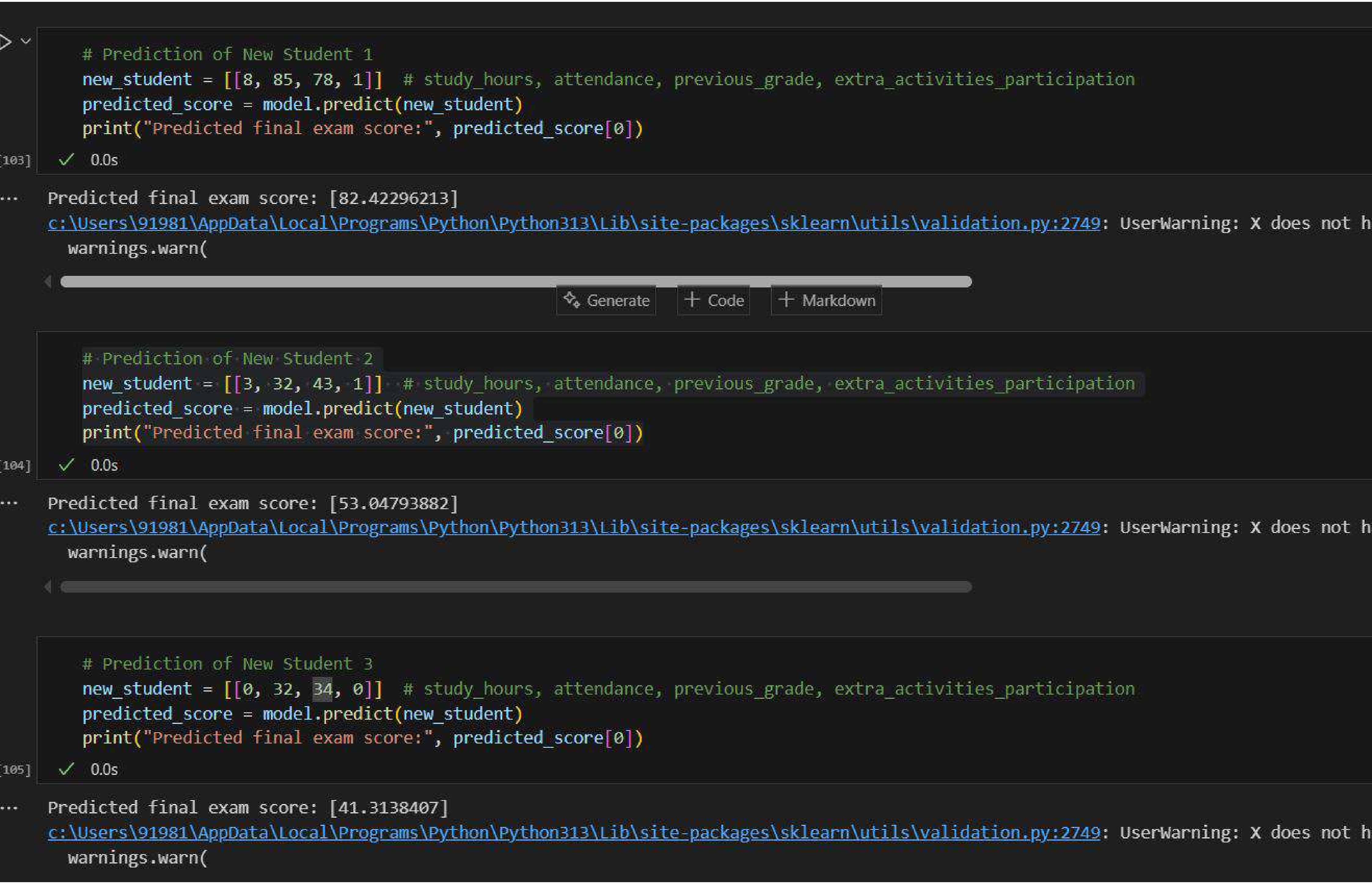
```
<seaborn.axisgrid.PairGrid at 0x1d952c6e190>
```



```
sns.displot(y_pred-y_test , kind='kde')
# The major probability distribution is between -6 and 6
# It means that the major variation is between -6 and 6
```

[102] ✓ 0.2s

```
<seaborn.axisgrid.FacetGrid at 0x1d95f2b8910>
```



```
# Prediction of New Student 1
new_student = [[8, 85, 78, 1]]  # study_hours, attendance, previous_grade, extra_activities_participation
predicted_score = model.predict(new_student)
print("Predicted final exam score:", predicted_score[0])
```

[103] ✓ 0.0s

```
Predicted final exam score: [82.42296213]
c:\Users\91981\AppData\Local\Programs\Python\Python313\Lib\site-packages\sklearn\utils\validation.py:2749: UserWarning: X does not h
  warnings.warn(
```

```
# Prediction of New Student 2
new_student = [[3, 32, 43, 1]]  # study_hours, attendance, previous_grade, extra_activities_participation
predicted_score = model.predict(new_student)
print("Predicted final exam score:", predicted_score[0])
```

[104] ✓ 0.0s

```
Predicted final exam score: [53.04793882]
c:\Users\91981\AppData\Local\Programs\Python\Python313\Lib\site-packages\sklearn\utils\validation.py:2749: UserWarning: X does not h
  warnings.warn(
```

```
# Prediction of New Student 3
new_student = [[0, 32, 34, 0]]  # study_hours, attendance, previous_grade, extra_activities_participation
predicted_score = model.predict(new_student)
print("Predicted final exam score:", predicted_score[0])
```

[105] ✓ 0.0s

```
Predicted final exam score: [41.3138407]
c:\Users\91981\AppData\Local\Programs\Python\Python313\Lib\site-packages\sklearn\utils\validation.py:2749: UserWarning: X does not h
  warnings.warn(
```