fit( )

vs

fit_transform( )

vs

fit_predict( )

1. fit() — Just Learn 📚

What it does:

- Fits the model or transformer to the data
- Learns parameters ***(e.g., mean and std in scalers, weights in models)***
- Doesn't return transformed data

Sample Python code

```
from sklearn.preprocessing import StandardScaler
import numpy as np
X = np.array([[1], [2], [3]])
scaler = StandardScaler()
scaler.fit(X)  # Calculates mean and std
print(scaler.fit(X))
print("Mean:", scaler.mean_)
print("Var :", scaler.var_)
print("Std  :", np.sqrt(scaler.var_))
```

📤 Output:

```
StandardScaler()
Mean: [2.]
Var : [0.66666667]
Std  : [0.81649658]
```

🔁 2. fit_transform() — Learn & Apply 🚀

🔁 What it does:

- Fits the data ***(learns parameters)***
- Transforms the data ***(applies the operation)***
- One-liner for fit() + transform()

Sample Python code

```
scaler = StandardScaler()
scaled = scaler.fit_transform([[1], [2], [3]])
print(scaled)
```

📤 Output:

[[-1.22], [0.], [1.22]]

🔍 3. fit_predict() — Learn & Guess 🧠

🔮 What it does:

- Fits the model to the data
- Predicts labels from ***unsupervised models (like clustering)***
- Common in algorithms like ***KMeans, DBSCAN***

Sample Python code

```
from sklearn.cluster import KMeans
X = [[1], [2], [10], [12]]

kmeans = KMeans(n_clusters=2)
labels = kmeans.fit_predict(X)
print(labels)
```

📤 Output:

[1 1 0 0]

📌 Quick Summary:

Method	Learns	Transforms	Predicts
fit()	✅	❌	❌
fit_transform()	✅	✅	❌
fit_predict()	✅	❌	✅

🧠 When to Use What:

Situation	Method
You only want to learn parameters	fit()
You want to transform data after learning	fit_transform()
You want to cluster or label data without supervision	fit_predict()

Let's connect
&
grow together.

Thank you.