Installing Scikit-learn and Making Your First Machine Learning Model

Step 1: Installing Scikit-learn

Open your terminal (or use Jupyter Notebook / Google Colab) and run:

```
pip install scikit-learn
```

Or in Google Colab:

```
!pip install scikit-learn
```

Step 2: Make Your First Machine Learning Model (in 5 minutes!)

Here’s a **“Can we predict if a fruit is an apple or an orange?”** example:

The idea:

We have some fruits.

- Apples are **heavy** and **red**
- Oranges are **lighter** and **orange-colored**

At the first glance, it seems easy to tell them apart, right? You can always compare Orange pixels and red pixels, or check the weight. But as you keep getting more fruits, it becomes harder to write rules for every single fruit and with the messy real world, rules you write starts to break. Let’s train a machine to learn this!

Code Example:

```python
from sklearn import tree

# Step 1: Collect Data
# Features: [weight (grams), color (0 = red, 1 = orange)]
features = [
    [150, 0],  # Apple
    [170, 0],  # Apple
    [130, 1],  # Orange
    [120, 1],  # Orange
]
labels = ["apple", "apple", "orange", "orange"]

# Step 2: Train a model
clf = tree.DecisionTreeClassifier()
clf = clf.fit(features, labels)

# Step 3: Predict!
print(clf.predict([[160, 0]]))  # Heavy and red-ish? Probably an apple!
print(clf.predict([[115, 1]]))  # Light and orange? Probably an orange!
```

What we Learned:

- **We just taught a computer to recognize fruits!**
- It looked at the weight and color to decide.
- We didn’t write “if-else” — the computer *learned* by itself!
- We’ve just done **machine learning**!

Step 3: Test Your Model

Now, let’s see how well our model works!

- “Try `[140, 1]` – what does it think?”
- “Try changing colors and weights randomly and test the model!”

Visualize It

You can show how the decision tree looks like:

```python
from sklearn.tree import plot_tree
import matplotlib.pyplot as plt

plt.figure(figsize=(10,6))
plot_tree(clf, feature_names=["weight", "color"], class_names=["apple", "orange"], filled=True)
plt.show()
```