Worksheet 9 - Nov 14, 2025

Intro to Machine Learning

Dataset: spotify_mini.csv

🧠 Goal

Today, you'll explore how computers *learn from data*!
We'll use a small Spotify dataset to understand:

- Train your first machine learning model
- Make predictions using real Spotify data
- Visualize the pattern the computer learned

🧩 Step 1: Load & Explore the Data

1. Download **spotify_mini.csv** from GitHub (Week 6) and upload it to your Colab sidebar (Files ➜ Upload).
2. Run this code 👇:

```
import pandas as pd

spotify = pd.read_csv("spotify_mini.csv")
spotify.head()
```

1. 📝 **Question 1:**
   What columns do you see in the dataset? List three columns you find interesting.

column 1	column 2	column 3

📊 Step 2: Choose a Feature and Label

We'll try to predict how many **streams** a song gets based on one feature (for example, skip_rate).

```
X = df[['skip_rate']]      # Feature (input)
y = df['streams']          # Label (output)
```

📝 **Question 2:**

- What is the **feature** in this example?
- What is the **label**?

Feature	Label

✂️ Step 3: Split the Data into Training and Testing Sets

We want to see if the computer can learn patterns — so we'll train it on part of the data and test it on the rest.

```
from sklearn.model_selection import train_test_split

X_train, X_test, y_train, y_test = train_test_split(
    X, y, test_size=0.2, random_state=42
)
```

📝 **Question 3:**

- How much of the data is used for training?
- How much is used for testing?
- Why do we split the data?

Training %	Testing %	Why do we split?

👀 Step 4: Check Your Split

Run this to see how many examples are in each part:

```
print("Training examples:", len(X_train))
print("Testing examples:", len(X_test))
```

📝 **Question 4:**
Write down how many rows your training and testing sets have:

Data Type	Number of Rows
Training	
Testing	

🧠 Step 5: Train the Model

We'll use a simple model called **Linear Regression**.
Think of it as: "Draw the smoothest line through the data."

Run this code:

```
from sklearn.linear_model import LinearRegression

model = LinearRegression()
model.fit(X_train, y_train)
```

📝 Question 5:

In your own words, what did the computer do when we called **model.fit()?**

(Example: "It learned the pattern between skip rate and streams.")

🎯 Step 6: Make Predictions

Now let's see what the model thinks about the test data.

```
predictions = model.predict(X_test)

for pred, real in zip(predictions, y_test):
    print(f"Predicted: {pred:.0f}   Actual: {real}")
```

📝 Question 6:

Look at the predicted vs actual values.
Write down one pair:

Predicted Streams	Actual Streams

📈 Step 7: Visualize the Pattern

A graph helps you *see* what the computer learned.

```
import matplotlib.pyplot as plt

plt.scatter(X, y, color='blue')                          # real song data
plt.plot(X, model.predict(X), color='red')               # model's line
plt.xlabel('Skip Rate')
plt.ylabel('Streams')
plt.title('Spotify Streams vs. Skip Rate')
plt.show()
```

📝 Question 7:

Look at your graph:

- Does the **red line** match the general direction of the data?
- What does it tell you about skip rate and streams?

💬 Step 8: Reflection

Answer in 1–2 sentences:

1. How is this similar to how Spotify guesses whether a song will be popular?

   ---

2. What do you think would make the model more accurate?
   (More data? More features?)

⭐ Bonus Challenge

Try replacing skip_rate with a new feature like minutes_listened:

```
X = df[['minutes_listened']]
```

Did your predictions improve?
Why do you think that happened?