📆 DAY 27 — Model Persistence (Saving & Loading Models)

🎯 Day 27 Goal

Learn how to:

```
Trainonce → Savemodel → Loadlater → Predictwithoutretraining
```

1️⃣ What Is Model Persistence? (CORE IDEA)

> Model persistence means storing a trained model to disk so it can be reused later.

Why this matters:

- Training can be expensive
- Production systems must load models instantly
- Retraining every time is impossible

2️⃣ What Needs to Be Saved? (VERY IMPORTANT)

⚠️ **Not just the model**

You must save:

- The **entire pipeline** (preprocessing + model)

Why?

- Preprocessing must match training exactly
- Scaling, encoding, feature order must be identical

📌 **Always save the pipeline, not just the model**

3️⃣ Tools for Model Persistence in Python

Option 1 — `joblib` (RECOMMENDED ✅)

- Faster for large NumPy objects
- Used by scikit-learn

Option 2 — `pickle`

- General Python serialization
- Slightly slower, less safe

📌 **Use `joblib` for ML**

4️⃣ Saving a Model (IMPORTANT)

Example:

```
import joblib

joblib.dump(pipeline,"model.joblib")
```

This stores:

- Scaler
- Encoder
- Model
- Parameters

All together.

5️⃣ Loading a Model (IMPORTANT)

```
loaded_pipeline = joblib.load("model.joblib")
```

Now you can:

```
loaded_pipeline.predict(new_data)
```

No training needed.

6️⃣ Why Pipelines + Persistence = POWER 🔥

If you saved only the model:

- You’d need to remember scaling
- Feature order matters
- Easy to make mistakes

Pipeline saves:

- Everything
- In correct order
- Safely

📌 This is how **production ML works**.

7️⃣ Common Mistakes (VERY IMPORTANT)

❌ Saving model before fitting

❌ Saving only classifier, not pipeline

❌ Changing feature names later

❌ Using different preprocessing

✅ Save after training

✅ Load before prediction

✅ Use same input schema

8️⃣ Where Is This Used in Real Life?

- Flask / FastAPI APIs
- Batch prediction jobs
- Scheduled pipelines
- Model versioning systems
