

## Lesson 41 Summary: Model Serialization

Data Science with Python – Key Concepts

Data Science Program

## Model Serialization for Deployment

### Why Serialize?

- Save trained models
- Skip retraining
- Share with others

### Pickle

- Python standard
- `pickle.dump/load`
- Simple but large

### Joblib

- sklearn optimized
- Compression built-in
- Faster for arrays

### Best Practices

- Version metadata
- Test after loading
- Document inputs

### Security

- Never unpickle
- untrusted files
- Verify source

### Workflow

- Train -> Save
- Load -> Predict
- Deploy to prod

Save trained models for deployment

# Why Serialize Models?

## Key benefits:

- **Persistence:** Save trained models to disk
- **Efficiency:** Skip expensive retraining
- **Sharing:** Distribute to production/colleagues

---

Essential for any ML deployment pipeline

## Python standard library:

- **Save:** `pickle.dump(model, file)`
- **Load:** `model = pickle.load(file)`
- **Note:** Use binary mode ('wb'/'rb')

---

Simple but can produce large files

## Optimized for sklearn:

- **Save:** `joblib.dump(model, 'model.pkl')`
- **Load:** `model = joblib.load('model.pkl')`
- **Compression:** `compress=3`

---

Recommended for sklearn models with large arrays

## Critical security risk:

- **Never** unpickle untrusted files
- **Pickle** can execute arbitrary code
- **Verify** model source before loading

---

Only load models from trusted sources

### Essential Commands:

Task	Code
Pickle save	<code>pickle.dump(model, open('m.pkl', 'wb'))</code>
Pickle load	<code>pickle.load(open('m.pkl', 'rb'))</code>
Joblib save	<code>joblib.dump(model, 'model.pkl')</code>
Joblib load	<code>joblib.load('model.pkl')</code>

---

Serialization enables model deployment at scale