Capsule

Capsule is a production-ready Python library that gives machine learning models superpowers in production environments. It provides immutable model wrappers with built-in drift detection, performance monitoring, secure serialization, and visualization capabilities for both classification and regression tasks.

Unlike traditional ML libraries that focus on training, Capsule specializes in production deployment concerns—wrapping your trained models with enterprise-grade monitoring and security features while maintaining a clean, unified API.

✨ Key Features

- 🔒 **Secure Model Serialization**: Optional AES-GCM encryption for model persistence and deployment
- 📊 **Built-in Drift Detection**: Automatic univariate drift monitoring using NannyML integration
- 📈 **Rich Visualizations**: Out-of-the-box plotting for model evaluation (ROC curves, scatter plots, etc.)
- 🎯 **Unified API**: Consistent interface for both classification and regression models
- 🔧 **Framework Agnostic**: Works with any scikit-learn compatible model
- 🛡️ **Production Ready**: Immutable wrappers prevent accidental model modification
- 📦 **Type Safe**: Full type hints and runtime validation with Pydantic
- 🔍 **Performance Monitoring**: Track model performance degradation over time
- 🎨 **Extensible**: Easy to extend with custom monitoring and visualization capabilities

🚀 Quick Start

```
from capsule import ClassificationCapsule, RegressionCapsule
from sklearn.ensemble import RandomForestClassifier
from sklearn.model_selection import train_test_split

# Train your model as usual
model = RandomForestClassifier()
X_train, X_test, y_train, y_test = train_test_split(X, y)
model.fit(X_train, y_train)

# Wrap it in a Capsule for production superpowers
capsule = ClassificationCapsule(model, X_test, y_test)

# Make predictions with drift detection
predictions = capsule.predict(X_new)

# Visualize performance
capsule.plots.roc_curve()

# Monitor for drift
drift_results = capsule.detect_drift(X_production)
```

Installation

Install from GitHub

```
pip install git+https://github.com/vitorbezzan/capsule.git
```

Development Setup

To set up Capsule for development, follow these detailed steps:

1. Clone the Repository

```
git clone https://github.com/<your-username>/capsule.git
cd capsule
```

2. Create a Virtual Environment (Recommended)

```
# Using venv
python -m venv venv
source venv/bin/activate  # On Windows: venv\Scripts\activate

# Or using conda
conda create -n capsule python=3.8+
conda activate capsule
```

3. Install in Development Mode

This project uses `pyproject.toml` for dependency management. Install the package in editable mode with all dependencies:

```
# Install the package in development mode
pip install -e .

# Install with development dependencies (if specified in pyproject.toml)
pip install -e ".[dev]"
```

4. Verify Installation

```
# Run tests to verify everything is working
pytest

# Or run tests with coverage
pytest --cov=capsule
```

5. Development Workflow

- The source code is located in `src/capsule/`
- Tests are in the `tests/` directory
- Documentation files are in `docs/`
- Use `pytest` to run tests during development
- The project configuration is managed through `pyproject.toml`

License

See LICENSE for details.