eduAutoML
=========

eduAutoML

[PyPI version](https://badge.fury.io/py/eduautoml)

[License: MIT](https://opensource.org/licenses/MIT)

eduAutoML is a beginner-friendly AutoML wrapper built for students, educators, and early-stage machine learning practitioners.
It simplifies the ML pipeline by automating task detection (classification/regression), preprocessing, model selection, training, and evaluation - all with a single command.

What is eduAutoML?

This tool automatically:

- Detects your ML problem type (classification or regression)
- Preprocesses data (handle missing values, encode categories, scale features)
- Selects and trains appropriate models
- Evaluates and displays performance metrics
- (Coming Soon) Provides a Gradio-based GUI for a no-code interface

Ideal for:

- AI & Data Science students
- ML beginners

- Hackathon prototypes
- Teaching automation in ML

Installation

Install via pip:

```
    pip install eduautoml
```

CLI Usage

Once installed, run from terminal:

```
    eduautoml run --file path/to/your/data.csv --target TargetColumnName
```

Example:

```
    eduautoml run --file data/iris.csv --target species
```

Available CLI options:

```
    --file        Path to input CSV file
    --target      Name of target/output column
    --test-size   Proportion of test split (default=0.2)
    --model       Optional: force model type (e.g., 'logistic', 'tree')
```

GUI Usage (Coming Soon)

A user-friendly Gradio GUI is being developed to allow drag-and-drop CSV uploads, configuration, and visual model evaluation - with zero code.

To launch:

```
    eduautoml gui
```

Stay tuned for updates and screenshots.

Features

- Auto task detection (classification/regression)
- Automated preprocessing
- Smart model selection and training
- Performance evaluation with clear metrics
- Model export (upcoming)
- Gradio GUI (coming soon)
- Sample datasets for learning

Examples

```
from eduautoml import AutoML

automl = AutoML("data/iris.csv", target="species")
automl.run()
```

Expected Output:

```
Detected task: Classification
Best model: RandomForestClassifier
Accuracy: 0.96
Precision: 0.95
...
```

You can find full examples in the examples/ folder.

Contributing

We welcome contributions from the community! To contribute:

1. Fork this repo
2. Create your feature branch (git checkout -b feature/awesome-feature)
3. Commit your changes (git commit -am 'Add awesome feature')
4. Push to the branch (git push origin feature/awesome-feature)
5. Open a pull request

Please make sure to update tests as appropriate.

License

Contact

For suggestions, feedback, or collaboration opportunities, feel free to connect via GitHub Discussions or raise an issue.