

Automating ML with PyCaret

Train & Compare Multiple Models to Find the Best Performer

About Me

Manjunath Janardhan, Principal AI Engineer

21+ years of software engineering experience

Patent Holder

Speaker, Medium Articles

Key expertise areas: AI/ML, Solutions Architect,
Microservices & Cloud



Agenda

- Introduction to ML and PyCaret Fundamentals
- Live Demo: Training Models with the Diabetes Dataset
- Live Demo: Comparing Models for Best Performance
- Live Demo: Model Persistence & Deployment
- Live Demo: Global Freelancers data
- Interactive Q&A Session



Introduction to Machine Learning

Machine Learning (ML): Algorithms enable computers to learn from data and make predictions or decisions.

Typical ML Workflow:

Challenges: Complexity, need for coding, time-consuming setup.

Data preprocessing

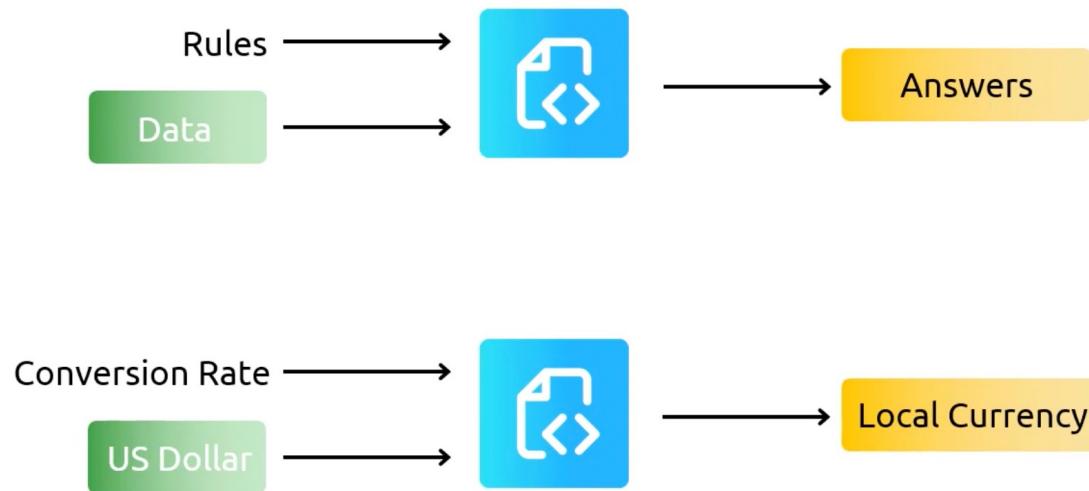
Model selection and training

Evaluation

Deployment

- Machine Learning vs Traditional Coding

Traditional Coding



- Machine Learning vs Traditional Coding

Traditional Coding



Machine Learning



Why PyCaret?

Low-code, fast ML
experimentation in Python

Clean API for training, comparing,
and tuning models

Automation of preprocessing,
imputation, and feature selection

Multiple model benchmarking
with one command

Live Demo - Training Models (Diabetes Dataset)



Dataset: Diabetes dataset with health metrics and outcome label.



Features:

Pregnancies,
glucose, blood
pressure, BMI,
age, etc.



Goal: Predict diabetes onset.



PyCaret Workflow:

Import module and datasets
Setup environment
Train models using multiple algorithms with one line of code

Live Demo - Training Models (Kaggle Global Freelancers Dataset)

Dataset: Global Freelancers dataset of fictional freelancer profiles from around the world

Features:

- name, gender, country, primary skill, age, etc.

Goal: Predict country.

PyCaret Workflow:

- *Import module and datasets*
- *Setup environment*
- *Train models using multiple algorithms with one line of code*

PyCaret vs MLOps Tools

Feature	PyCaret	MLflow	Metaflow
Primary Use	Low-code ML development	Model tracking, deployment	Workflow orchestration
Model Training	✓ Yes	✗ No	✗ No
Experiment Logging	Basic (opt-in)	✓ Full logging	✓ Supports via integrations
Feature Engineering	✓ Basic built-in	✗ Not included	✗ Not included
Scale / Parallel Training	✗ Limited	✓ With plugins	✓ Supports DAGs
Deployment Support	Basic (pickle export)	✓ REST, batch, registry	✓ via AWS, Argo, etc.

Key Takeaways



PyCaret streamlines the entire ML workflow



Enables efficient model development and testing



Supports easy model comparison, persistence, and deployment



Great for rapid prototyping and education!

Resources

Github : <https://github.com/pycaret/pycaret>

Demo :<https://github.com/manjunathshiva/PyData2025>

Docs : <https://pycaret.readthedocs.io/en/stable/>

Dataset : <https://github.com/pycaret/pycaret/blob/master/datasets/diabetes.csv>
<https://www.kaggle.com/datasets/urvishahir/global-freelancers-raw-dataset>

THANK YOU



<https://medium.com/@manjunath.shiva>

LinkedIn