**AN ARCHITECTURE FOR MACHINE LEARNING IN DJANGO**

**Web Applications that Learn by Example**

By Benjamin Bengfort and Rebecca Bilbro

---

**Simple View of Integration**

- **Data Exploration & Analysis**
  - Feature Analysis
  - Hyperparameter Tuning
  - Model Selection
  - Evaluation: Visual, Iteration
  - Cross-Validation

**Model Storage (Also "Model Management")**

- Models are stored in the database as pickles.
- Models can be retrieved and loaded by the web application.

**Computational Data Store (Also "Data Management")**

Integration happens at the database layer. The web application manages transactions on normalized feature tables, which are joined into an instance table for machine learning.

---

**Machine Learning Pipelines**

**Supervised Machine Learning - Clustering**

**Build Phase**
- Routinely (nightly/weekly) join feature tables into an instance table to create a static snapshot of the data to learn on.
- Engage the model selection triple to fit one or more models.
- Evaluate models using cross-validation.
- Pickle models and save them back to the database.

**Operation Phase**
- Initialize API by loading "best" model from the database into memory (time consuming, so must be done before request).
- Pass request to `model.predict()`.
- Store predictions to database and return the predicted response.
- Store feedback and update feature tables on POST/PUT/PATCH.
- Redraw predictions as DOLCE.

---

**Model Selection Triple**

- Feature Analysis
- Hyperparameter Tuning
- Model Selection

Evaluation: Visual Evaluation, Cross-Validation

For more on the model selection triple, check out Yellowbrick:
[https://github.com/DistrictDataLabs/yellowbrick](https://github.com/DistrictDataLabs/yellowbrick)

A visual diagnostic tool for machine learning.