AN ARCHITECTURE FOR MACHINE LEARNING IN DJANGO

PYCON 2016 ROSE CITY PORTLAND, OREGON MAY 28TH – JUNE 5TH

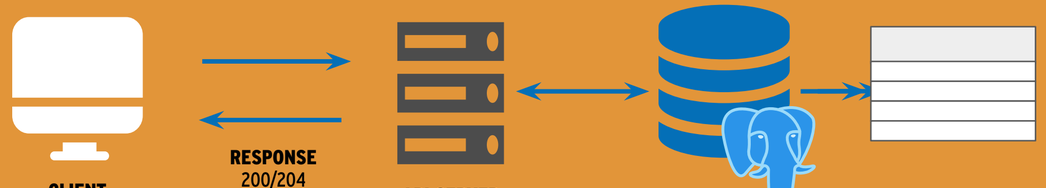
Web Applications that Learn by Example

By Benjamin Bengfort and Rebecca Bilbro

DistrictDataLabs

pycon.districtdatalabs.com

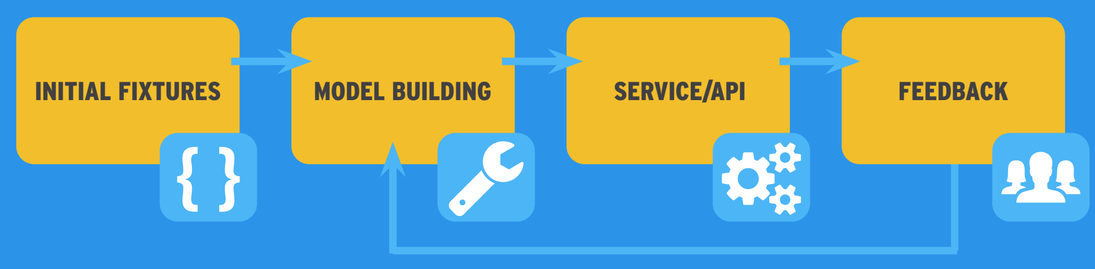
REST API DESIGN

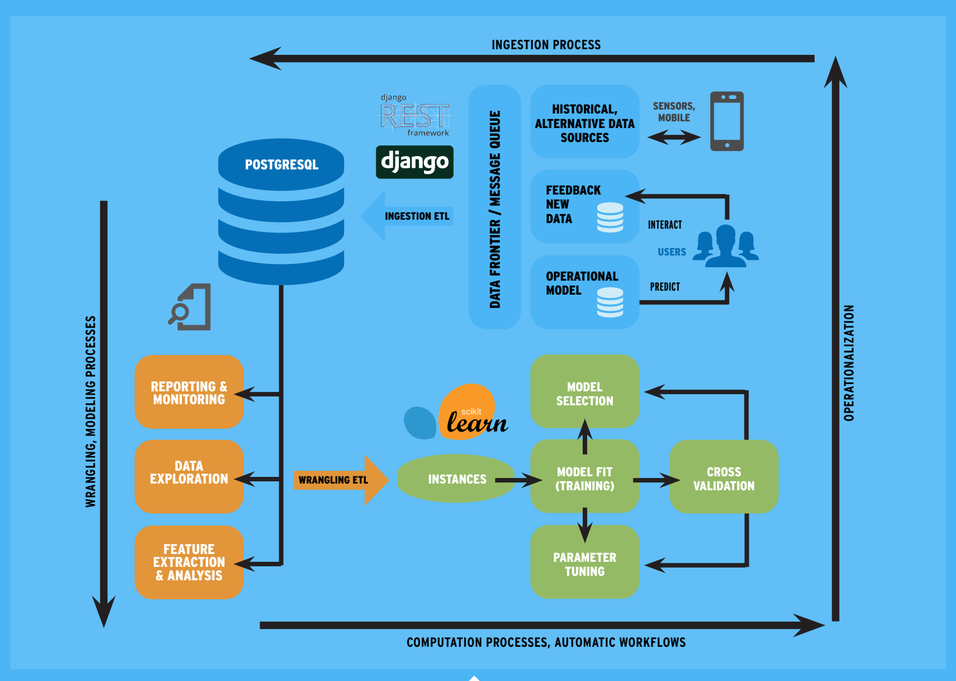


MACHINE LEARNING:
- Batch analysis: learns by iterating over a single table
- Must be routinely updated to take advantage of feedback
- Once trained, is able to make predictions on new data
- Can be evaluated via cross-validation

WEB APPLICATIONS:
- Transactional, deal with only a few rows at a time
- Are live, and perform computation on demand
- Must make immediate responses to requests
- Can be interactive and solicit feedback

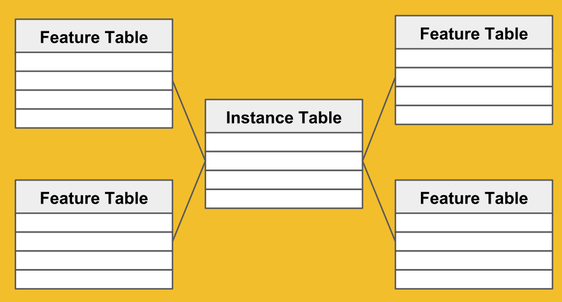
SIMPLE VIEW OF INTEGRATION



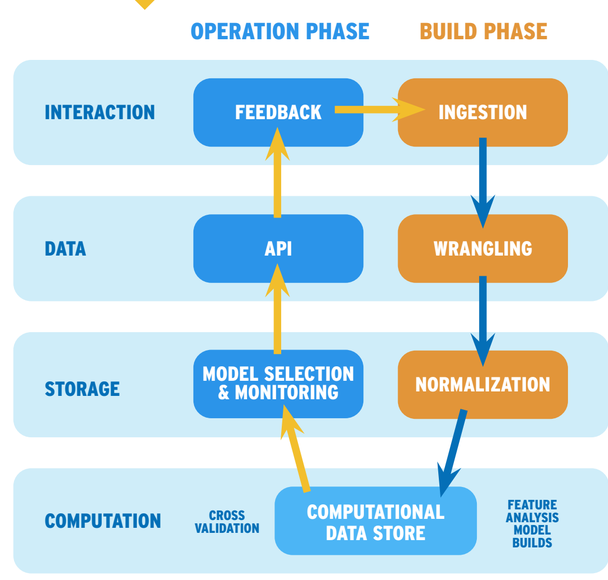


COMPUTATIONAL DATA STORE (ALSO "DATA MANAGEMENT")

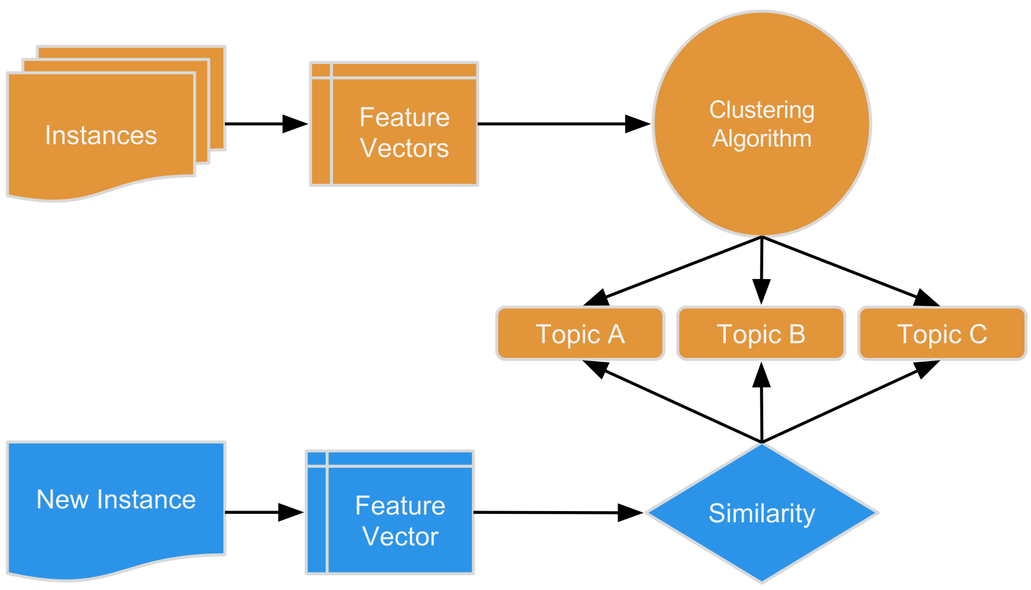
Integration happens at the database layer, which we will call a "computational data store". The Web app manages transactions on normalized feature tables, which are joined into an instance table for machine learning.



DATA PRODUCT PIPELINE



MACHINE LEARNING PIPELINES



SUPERVISED MACHINE LEARNING – CLUSTERING

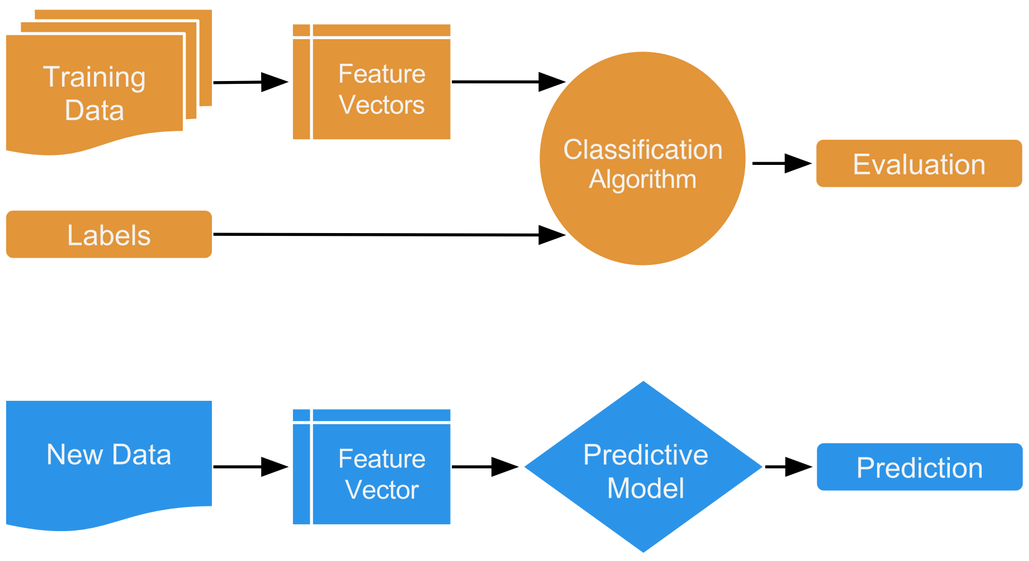
TWO PHASES IN DJANGO

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 GET request to model `predict()`.
- Save/update predictions to database and return the predicted response.
- Store feedback and update feature tables on POST/PUT/PATCH.
- Remove predictions on DELETE.



SUPERVISED MACHINE LEARNING – CLASSIFICATION/REGRESSION

MODEL STORAGE (ALSO "MODEL MANAGEMENT")

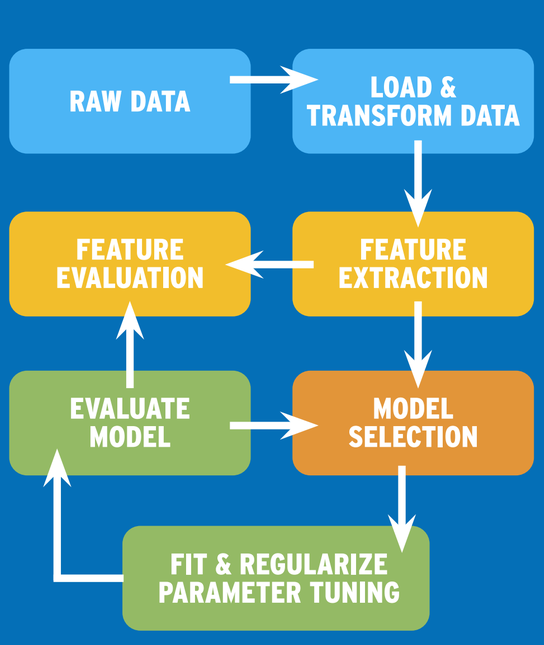
Additionally, models are also stored in the database as pickles, and can be retrieved and loaded by the web application.

ID	Model	Hyperparameters	Build Time	Score	Pickle
1	Naive Bayes	{"alpha": 1.0}	235.32	.832	BLOB
2	SVC	{"C": 1.0, "kernel": "linear"}	20.312	.861	BLOB
3	KNN	{"k": 5, "weights": "distance"}	482.129	.821	BLOB

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): a visual diagnostic tool for machine learning.