AN ARCHITECTURE FOR MACHINE LEARNING IN DJANGO

Web Applications that Learn by Example
By Benjamin Bengfort and Rebecca Bilbro

Machine learning algorithms differ from traditional data mining in that they utilize pattern recognition techniques to learn from historical data in an attempt to identify or predict outcomes for new information. As the algorithm makes predictions, feedback is utilized to adapt or correct the model so that it learns from new experience as well as the old. This makes web and mobile applications an ideal place to employ or deploy machine learning models.

REST API DESIGN

- **REVERSE**
  - GET
  - POST
  - PUT
  - DELETE

API SERVER
- Machine Learning
    - Batch analytics: learn by batching over a single table
    - Must be readily updated to take advantage of feedback
    - Been 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
- As new, and perform computation on demand
- Must make immediate responses to requests
- Can be interactive and solicit feedback

COMPUTATIONAL DATA STORE
(SAAS "DATA MANAGEMENT")

Ingestion controls of the database layer, which we will call a "computational data store". The Web app manages ODS/ETL-style transformations on normalized feature tables, which are joined into an instance table for machine learning.

DATA PRODUCT PIPELINE
- Interactions
  - Operation Phase
    - Model Selection
      - Feature Analysis
      - Hyperparameter Tuning
      - Model Selection
    - Model Evaluation
      - Cross-Validation
    - Model Storage (also "model management")
      - Store and manage machine learning models
- Ingestion
  - Data
    - API
    - Wrangling
- Storage
  - Machine Learning Model Storage
  - Normalization
- Computation
  - Computational Data Store
  - Feature Table

How do you operationalize machine learning models in an application?

Web applications are becoming smarter and more personalized through the clever use of automatic optimization and machine learning algorithms. In this paper, we present an architecture for Django web applications that utilizes machine learning models trained via Scikit-Learn to predict and tune experiences for specific users. Web application developers will discover that the tools they currently use like Django, REST Frameworks, and PostgresOL are also the framework required for online machine learning.

Machine learning methods have two phases: a training/validation phase and an operational phase. For this reason, traditional web architectures that employ an application server and backend database need to be adapted to fit into the ML lifecycle. In this paper, we present a machine learning architecture for directly using models inside web applications, particularly via a RESTful API.