Productionizing your ML code seamlessly

lauris #EuroPython
Connecting people with
great local businesses
Some fun facts about Yelp

🌟 Yelpers have written 155 million reviews since 2004.

💰 We have 74 million desktop and 30 million mobile app monthly unique visitors.

👨‍💻 We have over 500 developers.

📦 We have over 300 services and our monolith yelp-main has over 3 million lines of code!
Agenda for today

- What does running an ML model in production involve?
- How to improve your development workflow to make the path to production easier
Starting Point: your notebook
It solves your problem

- Predicts a desirable behavior
- Recommends the best items to your users
- Detects when an event is about to happen
-Forecasts trends in stocks
- Ranks items in search
Your notebook can be simple

Data Preparation

XGBoost Format

```python
from data_mining.util.libsvm import read_group_local
from data_mining.util.libsvm import read_libsvm_sklearn_local

# Dev set
X_xgboost, y = read_libsvm_sklearn_local('/code/dev_set/', len(feature_cols))
xgboost_group = read_group_local('/code/dev_set/)

# Prod set
X_xgboost, y = read_libsvm_sklearn_local('/code/prod_set/', len(feature_cols))
xgboost_group = read_group_local('/code/prod_set/)
```

Sklearn Format

```python
from sklearn.model_selection import GroupShuffleSplit

group_id_column = to_group_id_column(xgboost_group)
X = join_group_id_column(X_xgboost, group_id_column)
gss = GroupShuffleSplit(n_splits=1)

train_idx, test_idx = [(train, test) for (train, test) in gss.split(X, y, groups=group_id_column) if 0]

X_train = X[train_idx]
y_train = y[train_idx]

X_test = X[test_idx]
y_test = y[test_idx]
```
STARTING POINT: YOUR NOTEBOOK

... or not so much

```
# imports
from yelp_redshift.client import RedshiftClient
import pandas as pd
import yaml
import os
from datetime import date, timedelta
from scipy import stats
import pickle
from sklearn.base import TransformerMixin
from sklearn.preprocessing import FunctionTransformer

from sklearn.linear_model import LogisticRegression
from sklearn.model_selection import cross_val_predict
from sklearn.pipeline import Pipeline
from sklearn.metrics import roc_auc_score
from sklearn.metrics import roc_curve
from sklearn.metrics import confusion_matrix

pd.options.display.max_rows = 200
pd.options.display.max_columns = 200

# a bunch of functions, mostly to duplicate data_science repo functionalities

def get_from_yaml(datastore_name):
    # this will need to change for non-data_science environments
    datastore_yaml_path = "{0}/yelp_[1].yaml".format(os.path.expanduser("~/.data_science_config"), datastore_name)
    with open(datastore_yaml_path, 'r') as f:
        config_opts = yaml.load(f)
    return config_opts

def redshift_run(datastore_name, sql_text, execute=False, chunk_size=200000):
    # runs sql_text on datastore_name.
    # runs as a command if execute is true, otherwise returns results.
    config_opts = get_from_yaml(datastore_name)
    client = RedshiftClient(
        host=config_opts.get('host'),
        port=config_opts.get('port'),
        db=config_opts.get('db'),
    )
```
Making a model work in a notebook is only the first step to make it useful.

- Getting data to predict or train on regularly.
- Evaluate the model.
- Use the predictions in your product.
- Verify that your initial objective is accomplished, regularly.
“Only a small fraction of real-world ML systems is composed of the ML code [...] The required surrounding infrastructure is vast and complex.”

What does running an **ML model** in **production** involve?
WHAT DOES ML IN PROD INVOLVE?

This presentation is not about tooling
WHAT DOES ML IN PROD INVOLVE?

This is a **framework** on how to **tackle** the **problem**
WHAT DOES ML IN PROD INVOLVE?

ML pipeline
A simplified view

- **Data sources**
- **Sampling**
- **Feature Extraction**
- **Model Training**
- **Evaluation**
- **Model**

Training

- **Data sources**
- **Sampling**
- **Feature Extraction**
- **Model Training**
- **Evaluation**
- **Model**

Prediction

- **Data sources**
- **Sampling**
- **Feature Extraction**
- **Predictions**
- **Your product**
WHAT DOES ML IN PROD INVOLVE?

Your data source is **updating**

### Training

- Data sources
- Sampling
- Feature Extraction
- Model Training
- Evaluation
- Model

### Prediction

- Data sources
- Sampling
- Feature Extraction
- Predictions
- Your product
WHAT DOES ML IN PROD INVOLVE?

**Updating the model**

- 🕒 Regular training
- ⏳ Re-run strategies
- ✌️ Scale

---

**Training**

1. Data sources
2. Sampling
3. Feature Extraction
4. Model Training
5. Evaluation
6. Model

**Prediction**

1. Data sources
2. Sampling
3. Feature Extraction
4. Predictions
5. Your product

---

Your product
WHAT DOES ML IN PROD INVOLVE?

Evaluating Model

Training

- Data sources
- Sampling
- Feature Extraction
- Model Training
- Evaluation
- Model

Prediction

- Data sources
- Sampling
- Feature Extraction
- Predictions
- Your product
Evaluating Model

Does my evaluation metric reflect how this model will be used in production?

Consider both a classic metric and something that makes sense business wise.

Think about feedback loops
WHAT DOES ML IN PROD INVOLVE?

**Generating predictions**

- Regular training
- Re-run strategies
- Scale

**Training**

1. Data sources
2. Sampling
3. Feature Extraction
4. Model Training
5. Evaluation
6. Model

**Prediction**

1. Data sources
2. Sampling
3. Feature Extraction
4. Predictions
5. Your product
WHAT DOES ML IN PROD INVOLVE?

Using predictions

Training

Data sources

Sampling

Feature Extraction

Model Training

Evaluation

Model

Prediction

Data sources

Sampling

Feature Extraction

Predictions

Your product
Measuring Success

- Track the business metrics you are trying to move
- Confront your hypothesis to reality
- A/B test different models
- Beware about predicting what will happen anyway. Think about having a hold-out set.
How to **improve** your development workflow to make the path to **production** easier
General Advice

- Use containers, virtualenvs
- Persist your models, logs, code...
- Use prod technologies from the get go
- Rely on already existing tools
MAKE PRODUCTIONIZING EASIER

Feature Extraction

Training
- Data sources
- Sampling
- Feature Extraction
- Model Training
- Evaluation
- Model

Prediction
- Data sources
- Sampling
- Feature Extraction
- Predictions
- Your product
Feature Extraction

- Code should be **the same** between training and prediction

- **Unit Test** the feature extraction code

- Think about **edge cases**

- Write features as **code**
  Giant SQL queries are hard to review, test and maintain
Training

Data sources

Sampling

Feature Extraction

Model Training

Evaluation

Model

Prediction

Data sources

Sampling

Feature Extraction

Predictions

Your product

Evaluating Model

MAKE PRODUCTIONIZING EASIER
Evaluating Model

- Perform **feature importance analysis**, to be able to detect issues or big change between one training and the other.

- Set aside a **sanity check test set** for evaluation
Version
All the things

Keep track of change
**Log**

All the things

And persist it
Monitoring the pipeline

Training

- Data sources
- Sampling
- Feature Extraction
- Model Training
- Evaluation
- Model

Prediction

- Data sources
- Sampling
- Feature Extraction
- Predictions
- Your product

MAKE PRODUCTIONIZING EASIER
Monitoring the pipeline

- Keep track of the **number of prediction generated** (and alert when it’s 0)
- Keep track of **timings**, to be able to see problems earlier
- Alert on **errors** in your pipeline’s code
- Alert on the **business metrics** you are trying to move.
- Write **runbooks**
ML code is code, and all good practices still apply.

- Design for change, and evolution.

- Verify your assumption against reality. Really.
We're Hiring!

www.yelp.com/careers/

Offices @
Hamburg
London
San Francisco
Thank you!
Questions?