EuroPython 2020

Real Time Machine Learning with Python

Alejandro Saucedo | as@seldon.io

Twitter: @AxSaucedo
Hello, my name is Alejandro Saucedo

Engineering Director
Seldon Technologies

Chief Scientist
The Institute for Ethical AI & ML

Head of Solutions Eng & Sci
Eigen Technologies

Software Engineer & DevX Lead
Bloomberg LP
# Seldon: OSS Production ML Deployment

1. **Package**
   Create REST or gRPC dockerized microservice.

2. **Describe Deployment**
   Create/update Kubernetes resource manifest for deployment graph.

3. **Deploy**
   Manage and analyze the performance of live deployments.

## 2. Seldon Deploy
(UI, Collaboration, Control, Audit)

<table>
<thead>
<tr>
<th>MAB (Multi-Arm Bandits)</th>
<th>Outlier Detection</th>
<th>Explanation</th>
<th>Bias Detection</th>
</tr>
</thead>
</table>

## 1. **Seldon Core**
(runtime ML graph engine)

- Microservices - Istio service mesh (optional)

- **kubernetes**
The Institute for Ethical AI & Machine Learning

We are a UK-based think tank that brings together technologists, policymakers & academics to develop standards & frameworks for Data Governance & Machine Learning.
We are part of the LFAI
Today

- Conceptual intro to stream processing
- Machine learning for real time
- Tradeoffs across tools
- Hands on use-case
Real Time Reddit Processing

- Real time ML model for reddit comments
- 200k comments for training model
- /r/science comments removed by mods

We will be fixing the front page of the internet
A trip to the past present: ETL

E - Extract
T - Transform
L - Load
Variations

- ETL - Extract Transform Load
- ELT - Extract Load Transform
- EL - Extract Load
- LT - Load Transform
- WTF - LOL
Specialised Tools
Nifi
Flume
Oozie
Airflow
Elasticsearch
Data Warehouse
Jupyter notebook?
Batch VS Streaming

The spectrum of data processing
Batch VS AND Streaming

The right tool for the challenge
Unifying Worlds

Massive drive on converging worlds
Streaming Concepts: Windows

- Tumbling windows
- Sliding windows

Processing of batches in real time
Streaming Concepts: Checkpoints

Keeping track of stream progress
Streaming Concepts: Watermarks

Considering data that comes late in windows and stream batches
Some Stream Processing Tools

- Flink (Multiple Languages)
- Kafka Streams (Multiple Languages)
- Spark Stream (Multiple Languages)
- Faust (Python)
- Apache Beam (Python)
Today we’re using

Stream Processing
ML Serving
ML Training

FAUST
spaCy
CORE
kafka
scikit learn
Machine Learning Workflow

Machine Learning Model Creation

- Training data
- Transforming data for input into ML model
- Training machine learning model
- Persisted Training model

Prediction with trained model

- Unseen Data
- Transforming data for input into ML model
- Prediction
- Results
Model Training

```python
clean_text_transformer = CleanTextTransformer()

spacy_tokenizer = SpacyTokenTransformer()

tfidf_vectorizer = TfidfVectorizer(
    min_df=3,
    max_features=1000,
    preprocessor=lambda x: x, tokenizer=lambda x: x,
    token_pattern=None,
    ngram_range=(1, 3), use_idf=1, smooth_idf=1,
    sublinear_tf=1)

lr_model = LogisticRegression(C=1.0, verbose=True)
```
Model Training

"You are a DUMMY!!!!!!"

"You are dummy"

[ PRON, IS, DUMB ]

[ 1000, 0100, 0010 ]

[ 1 ]

```python
x_train_clean = 
    clean_text_transformer.transform(x_train)

x_train_tokenized = 
    spacy_tokenizer.transform(x_train_clean)

tfidf_vectorizer.fit(
    x_train_tokenized[TOKEN_COLUMN].values)

x_train_tfidf = 
    tfidf_vectorizer.transform(
        x_train_tokenized[TOKEN_COLUMN].values)

lr_model.fit(x_train_tfidf, y_train)
pred = lr_model.predict(x_test_tfidf)
```
More on EDA & Model Evaluation

https://github.com/axsaucedo/reddit-classification-exploration/
Overview of Components

Reddit Source

Processor: fetch_stream

kafka

Queue

Topic: reddit_stream
Topic: prediction
Topic: alert

Stream processor

Processor: ml_predict

ML Service

seldon model
spaCy
async def generate_reddit_comments():
    reddit_sample = await fetch_reddit_comment()

    reddit_data = {
        "id": reddit_sample["id"][0],
        "score": int(reddit_sample["score"][0]),
        ... # Cut down for simplicity
    }

    await app.topic("reddit_stream").send(
        key=reddit_data["id"],
        value=reddit_data)
```python
async def predict_reddit_content(tokenized_stream):
    async for key, comment_extended in tokenized_stream.items():
        tokens = comment_extended['body_tokens']

        probability = seldon_prediction_req(tokens)

        data = {
            "probability": probability,
            "original": comment_extended['body']
        }

        await app.topic("reddit_prediction").send(
            key=key,
            value=data)

        if probability > MODERATION_THRESHOLD:
            await reddit_mod_alert_topic.send(
                key=key,
                value=data)
```

ML Model Request Step

```python
sc = SeldonClient(  
gateway_endpoint="istio-ingress.istio-system.svc.cluster.local",  
deployment_name="reddit-model",  
namespace="default")

def seldon_prediction_req(tokens):  
data = np.array(tokens)  
output = sc.predict(data=data)  
return output.response["data"]['ndarray']
```
Overview of Seldon Model Serving

1. Containerise
2. Deploy
3. Monitor

From model binary
Or language wrapper
Into fully fledged microservice

Simple Seldon Core Inference Graph
- Model A
- API (REST, gRPC)

Complex Seldon Core Inference Graph
- Model B
- Model A
- Model C
- Multi Armed Bandit
- Feature Transformation
- Outlier Detection
- Explanation
- Direct traffic to the most optimal model
- Key features to identify outlier anomalies (Fraud, KYC)
- Why is the model doing what it's doing?
Wrapping ML models for Serving with Seldon

```python
import dill

from ml_utils import CleanTextTransformer, SpacyTokenTransformer

class RedditClassifier:
    
    def __init__(self):
        self._clean_text_transformer = CleanTextTransformer()
        self._spacy_tokenizer = SpacyTokenTransformer()

        with open('tfidf_vectorizer.model', 'rb') as model_file:
            self._tfidf_vectorizer = dill.load(model_file)

        with open('lr.model', 'rb') as model_file:
            self._lr_model = dill.load(model_file)

    def predict(self, X, feature_names):
        clean_text = self._clean_text_transformer.transform(X)
        spacy_tokens = self._spacy_tokenizer.transform(clean_text)
        tfidf_features = self._tfidf_vectorizer.transform(spacy_tokens)
        predictions = self._lr_model.predict_proba(tfidf_features)
        return predictions
```
Overview of Components

Reddit Source

Queue

kafka

Topic: reddit_stream

Topic: prediction

Topic: alert

Stream processor

Processor: fetch_stream

Processor: ml_predict

ML Service

seldon model

spaCy
Recap of Today

- Conceptual intro to stream processing
- Machine learning for real time
- Tradeoffs across tools
- Hands on use-case
EuroPython 2020

Real Time Machine Learning with Python

Alejandro Saucedo | as@seldon.io

@AxSaucedo