E-Commerce Sales Predictions MLOps Presentation

Joren Libunao, Zemin Cai, Theo Kim, Anirav Jain

Project goal

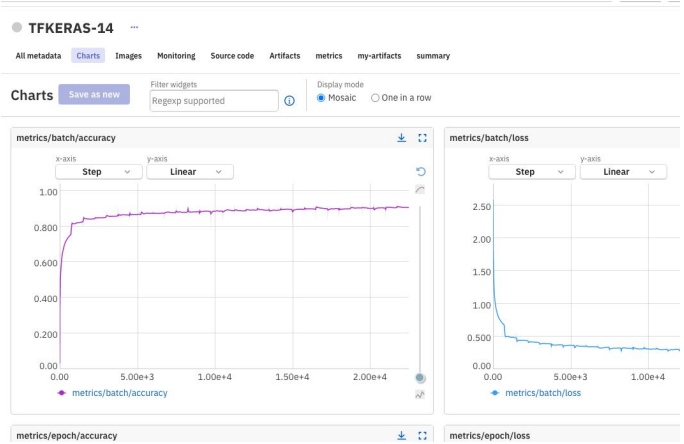
- Company Size: Mid-Size
- Team: Core ML
- Goal: Predicting Sales of Summer Clothes in E-Commerce
- The green check mark indicates we are picking that specific tool. ✅

Experiment Tracking

Comparing Neptune and ML flow

How: Research different aspects of the tools

- Financial costs
- Cost of learning
- Need for infrastructure and skill



mlflow™

neptune.ai

Experiment Tracking

neptune.ai

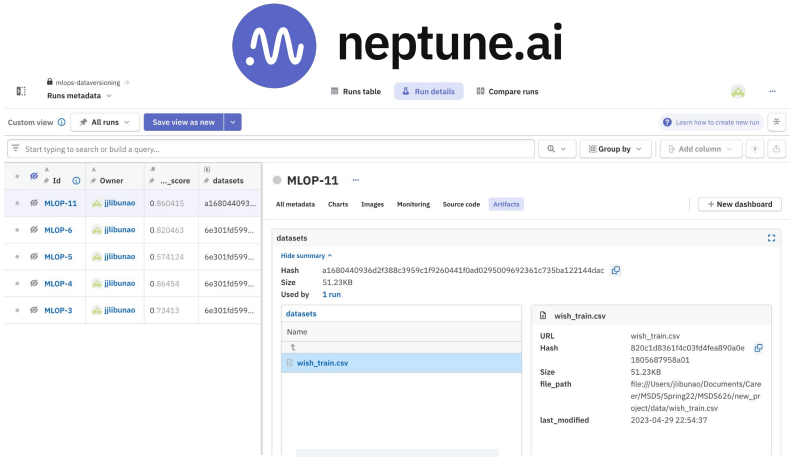
- Neptune is a cloud-based service, meaning that it is accessible from anywhere with an internet connection.
- Neptune has a more polished and modern interface, with a focus on collaboration and sharing.
- Neptune, on the other hand, has integrations with commercial tools such as Hugging Face and Weights & Biases
- Neptune has a free plan that charge 150$/month per team

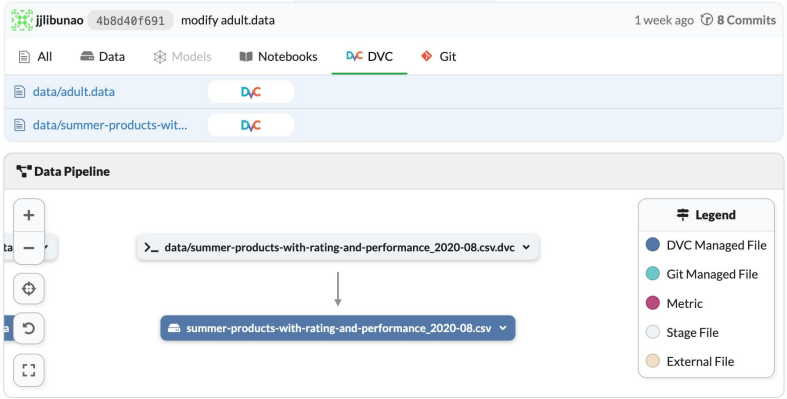
mlflow™

- MLflow can be installed on-premise or used through a cloud service like Databricks.
- MLflow's interface is simpler and more focused on tracking individual experiments.
- MLflow integrates well with other open-source tools, such as TensorFlow and PyTorch
- MLflow is an open-source tool, so there are no direct costs associated with using it. If you choose to use MLflow through a cloud service like Databricks, there will be associated costs.

Data Versioning POC

- DVC vs Neptune AI
- Comparisons:
  - Subscriptions and pricing
  - Ease of use
  - Limitations
- Versioned our dataset using both platforms to compare ease of use and find limitations in use





Data Versioning POC

neptune.ai

- Subscriptions and pricing: $150/month for a team, $600/month for entire organization
- Ease of use: User-friendly interface, built-in support for visualizing data, easy artifact tracking
- Limitations: Self-hosted deployment requires an organization plan, and need to use Neptune-client library as well
- Other:
  - Integrates well with other tools
  - Offers experiment tracking
  - Can collaborate with other team members, with ability to share notebooks with comments

DVC

- Subscriptions and pricing: $40/member/month ($160 total for the team)
- Ease of use: Complicated to set up, less intuitive than Neptune, need to integrate with Git
- Limitations: Must always be connected with Git to use any version control features
- Other:
  - Open source tool - A lot of support including email, chat, and community support for individuals and teams
  - Ability to share datasets and models easily with other team members

Data Quality POC

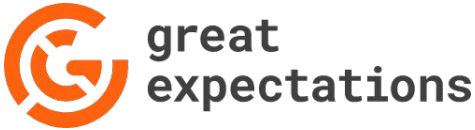
- Tools:
  - Great Expectations
  - Pandera
- Method: Implemented data quality using both tools on the Wish dataset
- Comparisons:
  - Cost
  - Functionalities
  - Ease of Use

great expectations

pandera

Data Quality POC

- Cost: Free (open-source)
- Functionalities:
  - Works with Pandas DataFrames
  - Can validate data
- Ease of Use:
  - Can define schema for individual columns only
  - Less steps
    - Define schema
    - Validate data and see result
  - Python-based syntax



- Cost: Free (open-source)
- Functionalities:
  - Works with Pandas DataFrames
  - Data profiling, documentation
  - Can validate relationships between tables
- Ease of Use:
  - Can define expectations across entire datasets or subsets of data
  - More steps
    - Create DataContext
    - Create Validator
    - Define the expectations
    - Create checkpoint
    - Run checkpoint
    - See result
  - YAML-based syntax

Model Orchestration POC

- Apache Airflow vs MetaFlow
- Comparisons:
  - Workflow Complexity
  - Ease of use
  - Limitations

Apache Airflow

METAFLOW

Model Orchestration POC

Apache Airflow

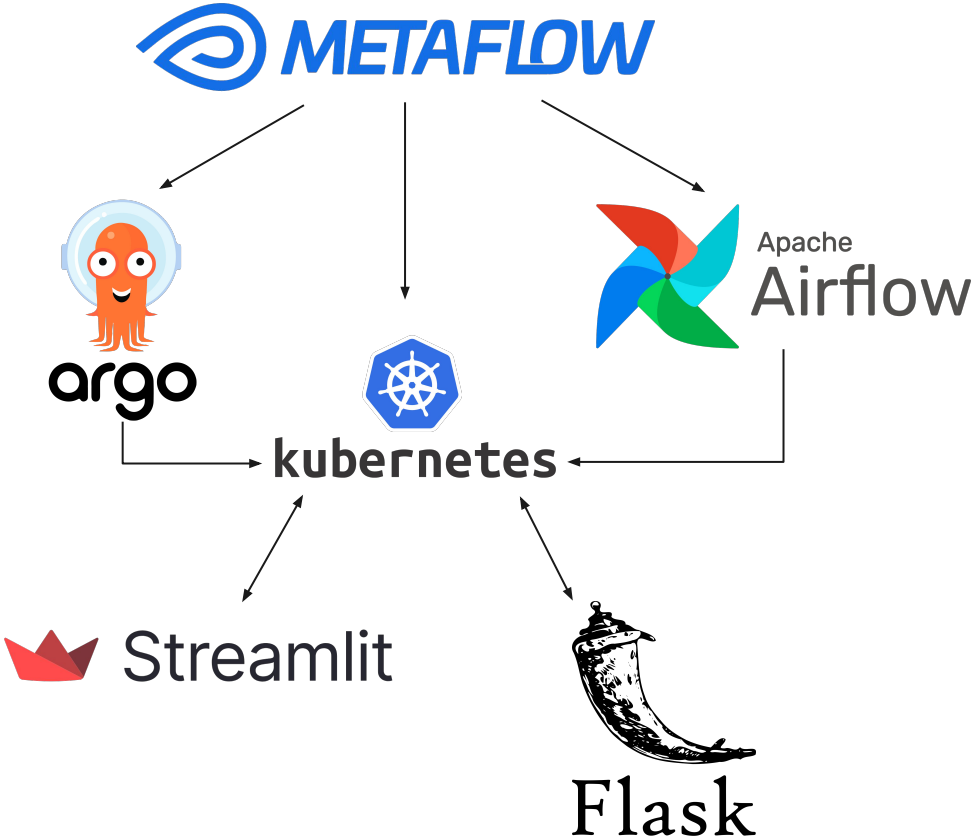
- Workflow Complexity: Extensibility and support for various integrations might be more suitable for complex workflows.
- Ease of use: Can be challenging to learn and set up
- Limitations: Can be resource-heavy and may require significant infrastructure management, especially for large scale deployments.
- Other:
  - Has a large and active community
  - Rich UI for monitoring and managing workflows
  - Developed by Airbnb to programmatically author, monitor and schedule workflows.

METAFLOW

- Workflow Complexity: Lightweight design might be more suitable for simple workflows.
- Ease of use: Simple, user friendly interface which automatically versions all code, data and models.
- Limitations: Fewer community resources and only integrates with AWS.
- Other:
  - Mainly built for python based workflows.
  - Developed by Netflix to improve productivity of data scientists.

Model Deployment and App POC

- Method: Offline deployment
- Tools:
  - Metaflow + Argo Workflows and Kubernetes + Streamlit
  - Metaflow + Airflow and Kubernetes + Flask
- Comparisons:
  - Ease of use
  - Integration with tools, specifically Kubernetes
  - Web app development capabilities
- Using both methods, attempted to build prototype apps using Wish dataset to compare ease of use and process flow between tools
- All tools are free and do not require licensing fees except possibly Metaflow



Model Deployment and App POC

Workflow 1: argo and Streamlit

- Ease of use:
  - Argo Workflows is Kubernetes-based, uses user-friendly interface, command line, and YAML files to define workflows, but can be complex or difficult to configure
  - Streamlit is fairly intuitive and simple, use directly with Python
- Both work well with Kubernetes, Argo also works well with third-party services
- In Argo, you specify the steps of your workflow, inputs/outputs, and dependencies
- Streamlit is a good choice for rapid prototyping and POC projects, but does not offer as much customization as Flask

Workflow 2: Apache Airflow and Flask

- Ease of use:
  - Apache Airflow requires Python to define schedules, but can be fairly difficult and unintuitive to debug
  - Flask requires more setup and configuration, would need to teach web development skills (HTML/CSS) in order to use efficiently
- Both work well with Kubernetes
- Airflow offers more workflow / scheduling flexibility if you're using many different technologies
- Flask allows for more advanced web app development and customization, but makes rapid prototyping difficult

Model Monitoring

1. User interface
2. Integration
3. Features
4. Cost

EVIDENTLY AI

Grafana

Model Monitoring

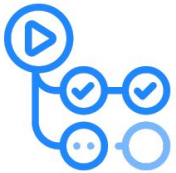
EVIDENTLY AI

- Tool designed specifically for machine learning model monitoring and evaluation
- User-friendly interface that is optimized for machine learning model monitoring
- Integrates with popular machine learning frameworks like TensorFlow, PyTorch, and scikit-learn
- Provides a range of machine learning-specific metrics and visualizations, including model performance metrics, feature drift, and error analysis
- Evidently is a commercial tool, and pricing is based on the number of models and the number of data points being monitored.

Grafana

- More general-purpose data visualization tool that can be used to monitor a wide range of metrics and data sources.
- More flexible and customizable, but it requires more technical expertise to set up and use effectively.
- Provides a range of visualizations and dashboards that can be used to monitor a wide range of metrics and data sources, but it does not include machine learning-specific features.
- Open-source, and there are no direct costs associated with using it.

CI/CD/CT



GitHub Actions 

- Runs on GitHub cloud
- Can define workflow in GitHub repo
- Easier configuration
- Free for small volumes

Jenkins

- Runs on a self-hosted server or a separate cloud service
- Has a larger ecosystem with more flexible and extensive functionality
- Requires a separate interface for configuration
- More mature package with a bigger community

Architecture diagram