DATA+AI SUMMIT
BY databricks

AI-POWERED FINTECH:
AR FORECASTING WITH DATABRICKS & MLFLOW

Floriant Sturm & Julie Vanackere
12/06/2024

Nice to meet you!

element61
sharing experience & expertise

Julie Vanackere
Data scientist

daidalo.

Floriant Sturm
Co-founder

Outline

2 main topics

Business perspective	Technical perspective
1. Why Accounts receivable (AR) forecasting? 2. How do we approach this? 3. How can AR forecasting become a plug & play solution and what are the **technical requirements**?	1. Automated infra deployments 2. Standardized feature engineering 3. Standardized ML training 4. An automated ML lifecycle 5. Monitoring for customer confidence

Business perspective

Why did we focus on AR forecasting?

What is AR forecasting?

Accounts-Receivable
~ Outstanding invoices

- Currently companies use a <u>reactive approach</u> to chase “late-payers”
  - They contact them after it is too late
  - They know their outstanding invoices
- Big issue
  - “Cash flow is the pulse of the company”

Forecasting

- Predicting <u>when</u> a customer will pay their invoice in the future
- This will help you to anticipate
  - Who will pay late
  - How much cash is to be expected in the next x period
- Contact strategy: incentivize customers that are likely to pay late = <u>proactive approach</u>

Why did we focus on AR forecasting?

Why is it relevant for all companies?

- Quick win!
  We only need the historical invoice data to get started
- Predicting future cash flows reduces:
  - Credit risk
  - Plan expenses, investments & potential savings

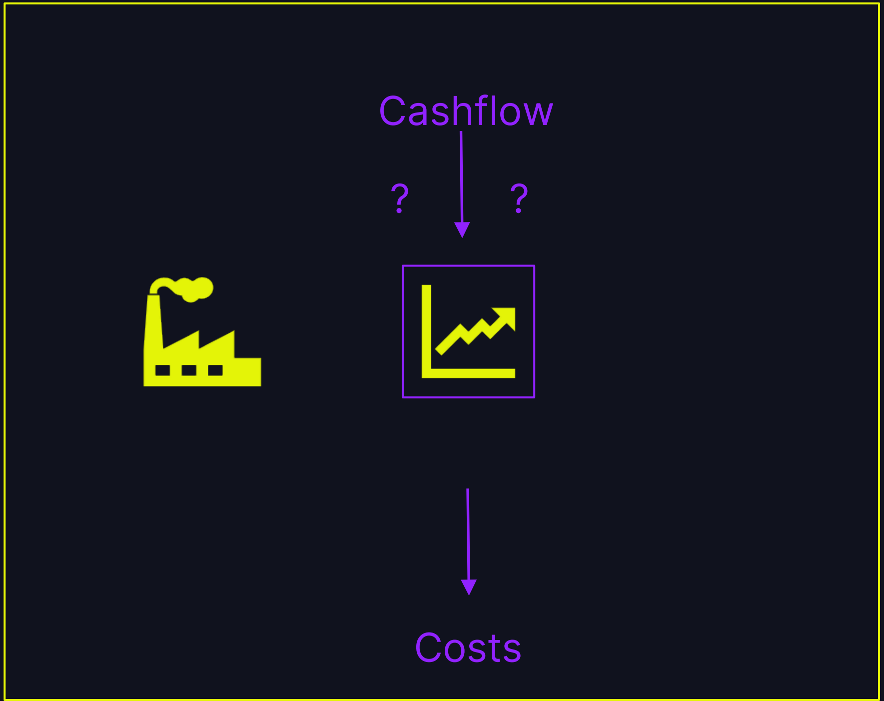
So… the proactive approach is the clear way to go!

Why did we focus on AR forecasting?

A tangible example

Context:

- **Production** company of pharmaceuticals
- Need to expand the production plant and invest in machines
- Do we have the cash flow to cover the costs?



How do we approach this?

How do we provide a sustainable approach?

Where do we generally focus on?

- Identifying the business problem
- Strategy - focus on a sustainable solution
  - Provides direct impact
  - Efficient implementation
  - Easy maintainable by the client
- Business validation
- Coaching & development

How do we look at Data Science?

- We try to go beyond, but how?
  - We keep the baseline structure (gathering data, etc...)
  - ...but the core business use case is tackled by Data Science
- Data Science means ML, AI,... whatever suits the business case best

How can AR forecasting become a plug and play solution?

What were our initial requirements?

1. Automated infra deployments
2. Standardized feature engineering
3. Standardized ML training
4. Automated ML lifecycle
5. Monitoring for customer confidence

Technical perspective

Automated infra deployments

Deploy Azure infrastructure quickly through Terraform

Infrastructure-as-code that deploys:

- Resources
  - Databricks
  - ADF
  - Storage Account
  - Keyvault
- Networking
- Roles and responsibilities
- 2 environments

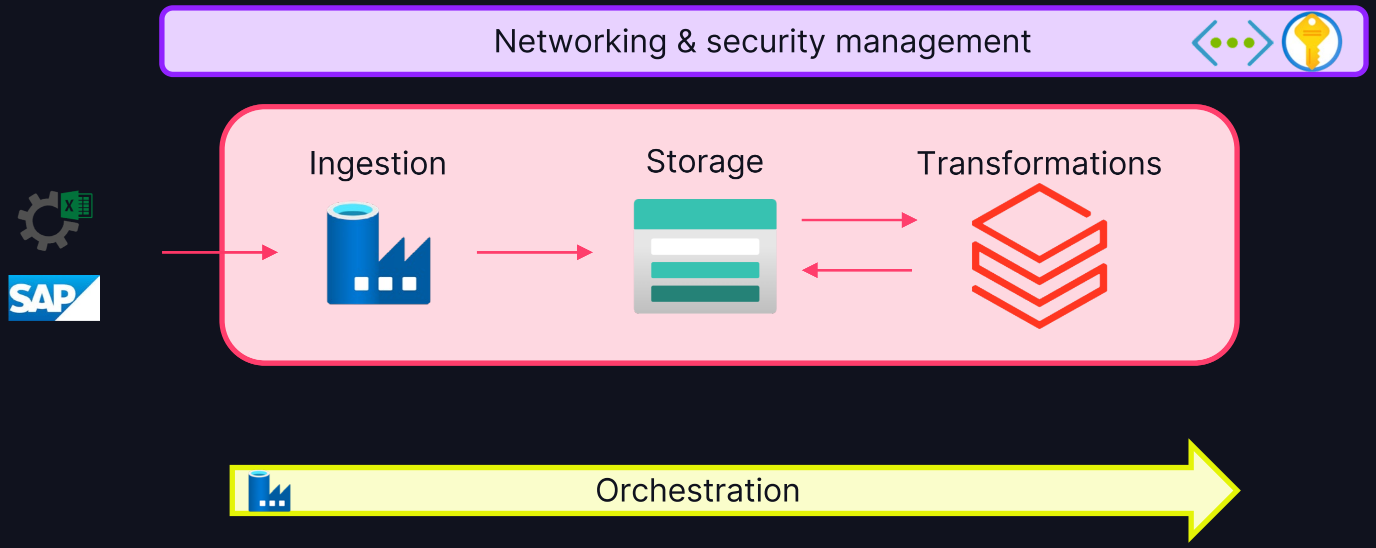
All automated through scripting

What does terraform look like?

```
module "e61-tff" {
  source = "../e61-tif"

  tags            = var.tags
  global_settings = var.global_settings

  resource_groups = var.resource_groups

  networking = {
    vnets                              = var.vnets
    route_tables                       = var.route_tables
    routes                             = var.routes
    network_security_group_definition = var.network_security_group_definition
  }

  security = {
    keyvaults = var.keyvaults
    keyvault_access_policies = var.keyvault_access_policies
  }

  storage_accounts = var.storage_accounts

  analytics = {
    databricks_workspaces = var.databricks_workspaces
  }

  role_mapping = var.role_mapping
```

Automated infra deployments

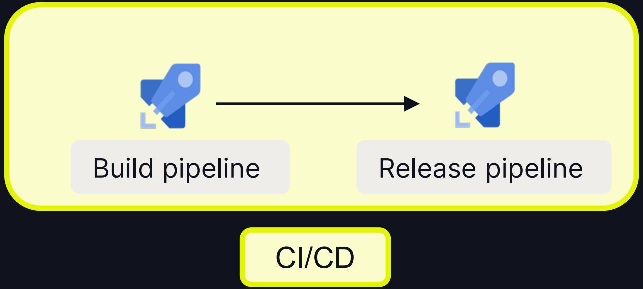
We start with a Modern Data Platform in Azure



Automated infra deployments

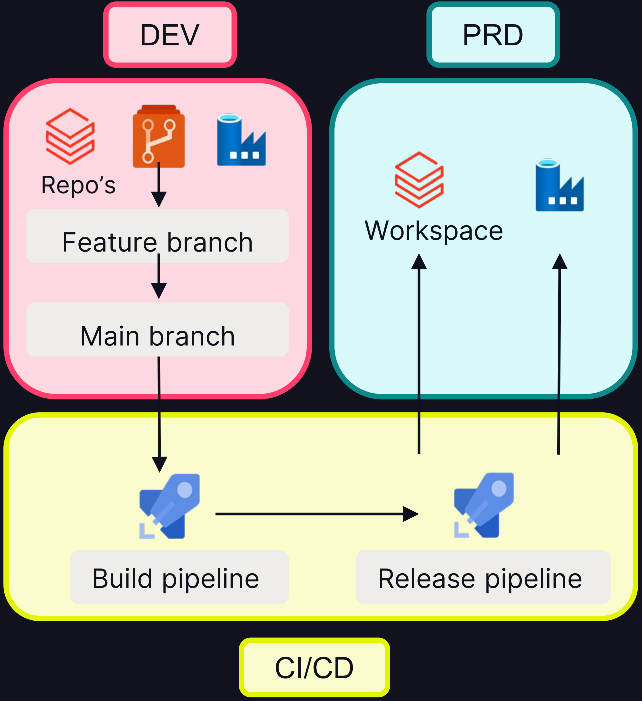
This facilitates a development – (acceptance) – production set up

- Because of **terraform** different environments with the same resources can be easily setup
- Because of the **CICD pipelines**, code can be reproduced in these environments
- But how do we do this practically?



Automated infra deployments

Afterwards, we deploy our code to Databricks and ADF using Devops CI/CD



- Development environment
  - Code is stored in Azure devops
  - Databricks repo code is deployed in Databricks workspace (ML models)
  - ADF GIT is deployed to ADF live mode
- Production environment

  Stable code (finished ML models) runs in prd environment if used for critical business processes = extra layer

Fully managed through scripts

How can AR forecasting become a plug and play solution?

What were our initial requirements?

1. Automated infra deployments ✓
2. Standardized feature engineering
3. Standardized ML training
4. Automated ML lifecycle
5. Monitoring for customer confidence

Standardized feature engineering

Feature table for AR forecasting

Invoice-level features (mandatory)

- Year invoice was created
- Month in which the invoice is due
- Document type
- # Line items in invoice
- The invoice amount

Customer-aggregated features (optional)

- % previous invoices late
- # of previous invoices
- Whether the last invoice was late (0/1)
- Preferred payment date

Data collections (optional)

- When and with what action did we contact the customer?
- At which dunning level?
  - 1: sending reminder
  - 2: calling
  - 3: giving a fee

Standardized ML training

We use the specified features, to make predictions

Features

Invoice-level features

- Year invoice was created
- Month in which the invoice is due
- Document type
- # Line items in invoice
- The invoice amount

Customer-aggregated features

- % previous invoices late
- # of previous invoices
- Whether the last invoice was late (0/1)
- Preferred payment date

Data collections

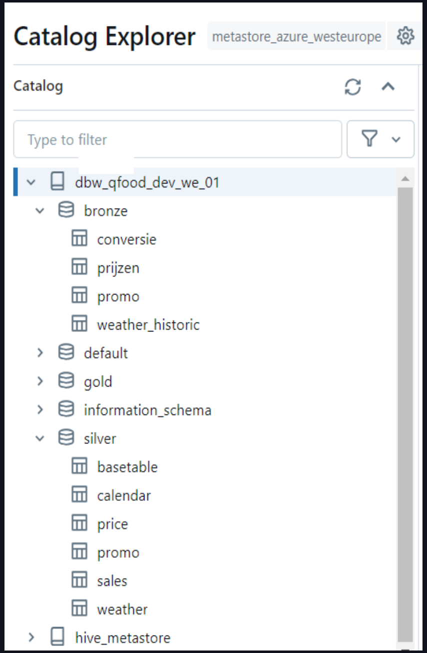
- When and with what action did we contact the customer?
- At which dunning level?
  - 1: sending reminder
  - 2: calling
  - 3: giving a fee

Target

- Regression: # Days late
- Classification: in buckets
  - On time
  - 0-30 days late
  - 30-60 days late
  - > 60 days late

Standardized feature engineering

Where do we store these features & labels?



Delta tables

- Delta files stored on data lake
- ACID
- Natively integrated with Unity Catalog
- Upserts & truncate insert

How can AR forecasting become a plug and play solution?

What were our initial requirements?

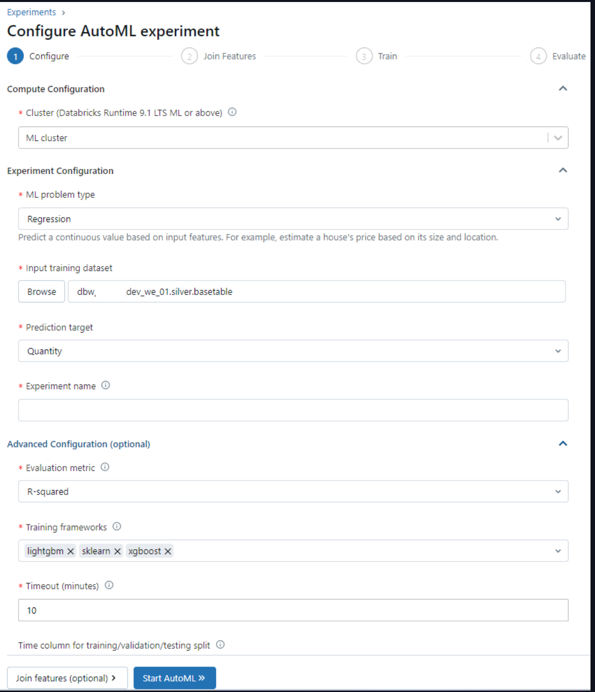
1. Automated infra deployments ✓
2. Standardized feature engineering ✓
3. Standardized ML training
4. Automated ML lifecycle
5. Monitoring for customer confidence

Standardized ML training

What is AutoML and how do we use it?

- AutoML: interface & code based model training in databricks
- Can be used for exploration, but we use it for model training as a whole
- The best model (according to R2) is automatically stored in Mlflow registry
- We track the feature importance to iterate on

```
run_name = "AutoMl Model"
databricks.automl.regress(dataset = training_data_filtered,
                          target_col = "nr_days_late",
                          exclude_frameworks = ["xgboost"] ,
                          experiment_name = "AR forecasting AutoML",
                          primary_metric = "r2",
                          timeout_minutes = 5
    )
```



How can AR forecasting become a plug and play solution?

What were our initial requirements?

1. Automated infra deployments ✔
2. Standardized feature engineering ✔
3. Standardized ML training ✔
4. Automated ML lifecycle
5. Monitoring for customer confidence

An automated ML lifecycle

How do we manage the model lifecycle?

mlflow™

Train a model

Experiments

Model in production

Model registry

Model in staging

Model registry

Versions | All | Active 2 | Compare

Version	Registered at	Created by	Stage
Version 3	2022-11-22 20:00:11	floriant.sturm@element61.be	Staging
Version 2	2022-11-22 19:59:46	floriant.sturm@element61.be	Production
Version 1	2022-11-22 19:46:45	floriant.sturm@element61.be	Archived

- We use MLFlow – natively integrated in Databricks
- Everything we need:
  - Experiments
  - Model registry with lifecycle mgmt.
  - Python SDK (automation ☺)

An automated ML lifecycle

4

How do we choose to update the production model?

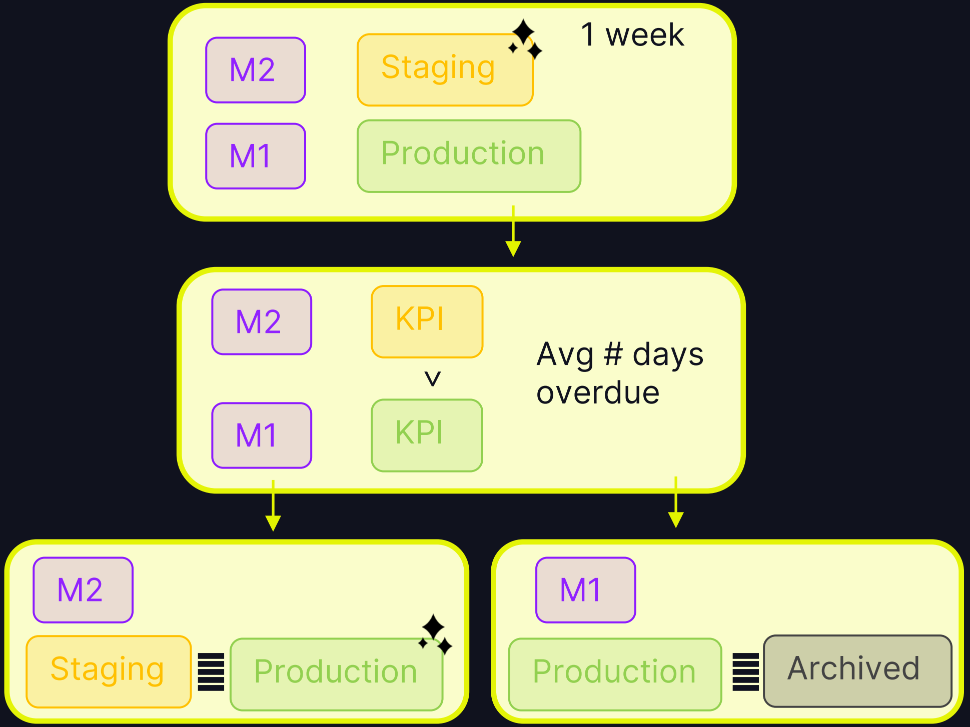
- Difference between technical KPI (R2) and business KPI (# Days late)
- Technical KPI as benchmark
  - No actions towards customers
  - Gradually improves when retraining the model
- Business KPI as benchmark
  - Switch to Business KPI when technical KPI gradually worsens
  - Actions have been taken towards customers
  - The ML model is used in the business for a while now
- Let's say we focus on the second scenario

An automated ML lifecycle

How do we choose to update the production model?

We are solving a business problem, so

- We use a business KPI – RMSE on actuals
- We retrain the model with new data every week.
- The new model becomes a staged model that « shadows » the production model
- We only update the production model when our business KPI improved



How can AR forecasting become a plug and play solution?

What were our initial requirements?

1. Automated infra deployments ✓
2. Standardized feature engineering ✓
3. Standardized ML training ✓
4. Automated ML lifecycle ✓
5. Monitoring for customer confidence

Monitoring for customer confidence

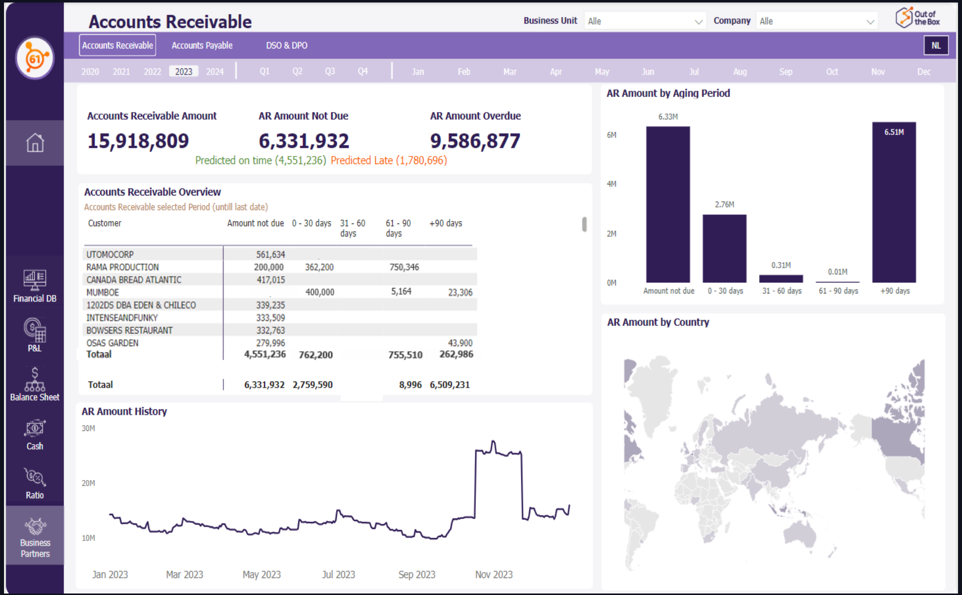
5

How do we enable trust? – the most important ‘KPI’ of an AI-solution

Dashboard for customers:	Dashboard for data scientists:
Build insights in the actuals vs predictions • Amount/invoice that will be overdue per customer • How many days this will be overdue • Action list: which customers to target? • Comparison of cash flows to future investments	Monitor data & models over time: • Model versions - keep track of historic versions • Model performance - technical KPIs • Model performance - business KPIs • Data drift

Monitoring for customer confidence

Example of a customer insights dashboard



- Periodic buckets with amounts
- Actionable dashboard
- Contact those with large amounts with 90+ days predicted

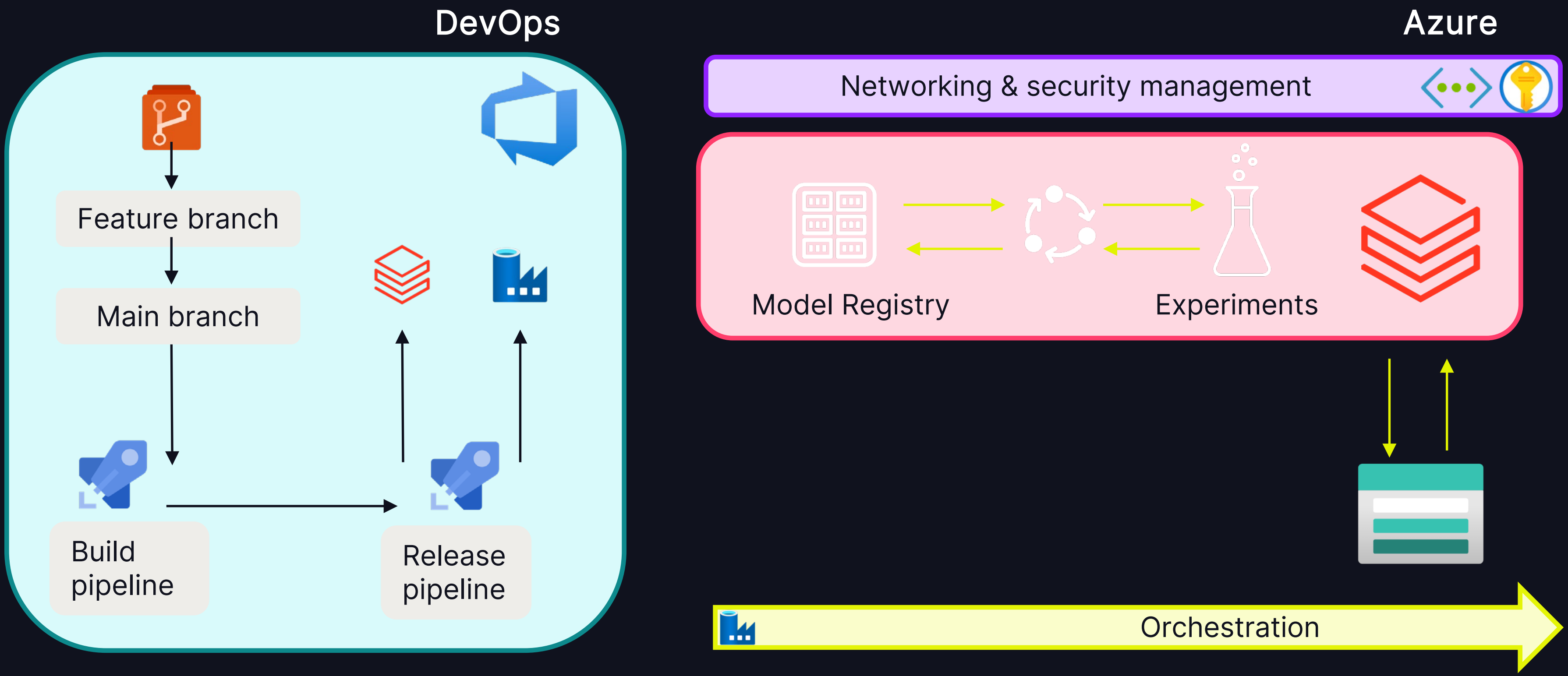
How can AR forecasting become a plug and play solution?

What were our initial requirements?

1. Automated infra deployments ✔
2. Standardized feature engineering ✔
3. Standardized ML training ✔
4. Automated ML lifecycle ✔
5. Monitoring for customer confidence ✔

Great! All requirements fulfilled!

We managed to build a plug -and-play ML solution!



Let's have a chat!

element61
sharing experience & expertise

Julie Vanackere
Data scientist

daidalo.

Floriant Sturm
Co-founder

https://www.linkedin.com/in/julie-vanackere/

https://www.linkedin.com/in/floriantsturm/