O'REILLY®

Databricks Machine Learning Associate Certification Prep

Dr. Yasir Khan
CEO, 38 Labs

www.38labs.com

How Long Have You Been Using Databricks at Work?

- Never used
- Less than 6 months
- 6 months to 1 year
- 1 to 2 years
- More than 2 years

Course Objectives

- Learn about Databricks Lakehouse Platform
- Explore advanced ML concepts: AutoML, MLflow, Spark ML, Pandas
- Set up ML clusters & run end-to-end ML workflows
- Put ML models into production with industry best practices
- Prepare for the Databricks Certified Machine Learning Associate exam

Course schedule

Day 1: Architecture and ML Concepts

- Introduction
- Databricks UI
- ML Key Concepts
- Databricks Runtime for ML
- Classification, Regression, Forecasting

Day 2: Advanced ML

- Feature Store
- Managed MLflow
- MLflow Model Registry
- Exploratory Data Analysis
- Feature Engineering

Day 3: ML Workflows

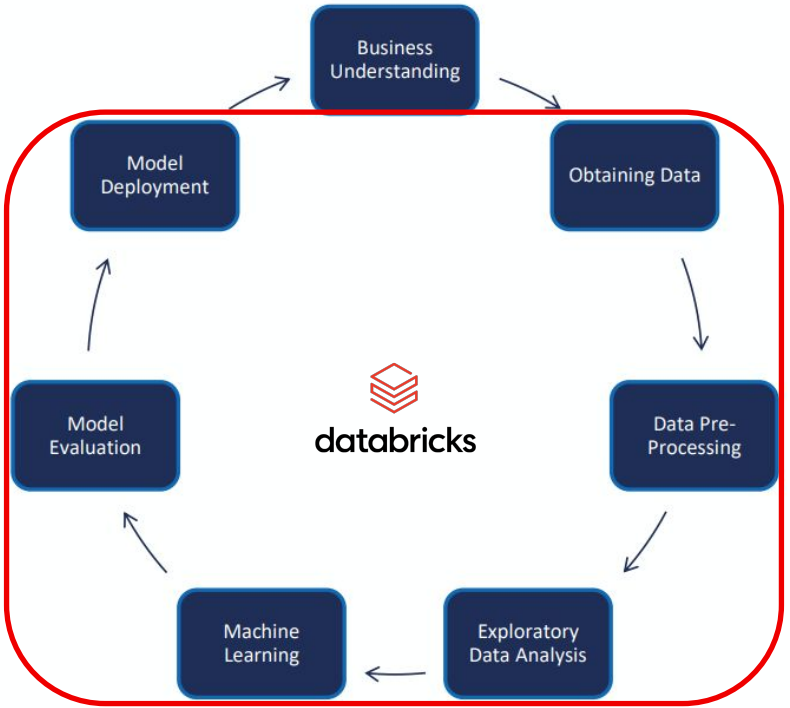
- Hyperparameter Tuning
- Evaluation and Selection
- Binary Classification, Regression & Decision Trees
- Spark ML Modeling APIs

Day 4: ML Scalability and Pandas

- Scaling ML Models
- Pandas on Databricks
- Pandas API on Spark
- Pandas Function APIs
- Pandas User Defined Functions (UDFs)

Today’s Schedule & Learning Objectives

- Introduction
- Databricks UI
- ML Key Concepts
- Databricks Runtime for ML
- Classification, Regression, Forecasting



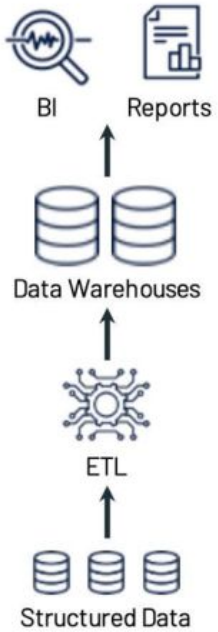
Day 1
Architecture and ML Concepts

Introduction to Databricks

Big Data Platform Architecture Evolution

Data Warehouse

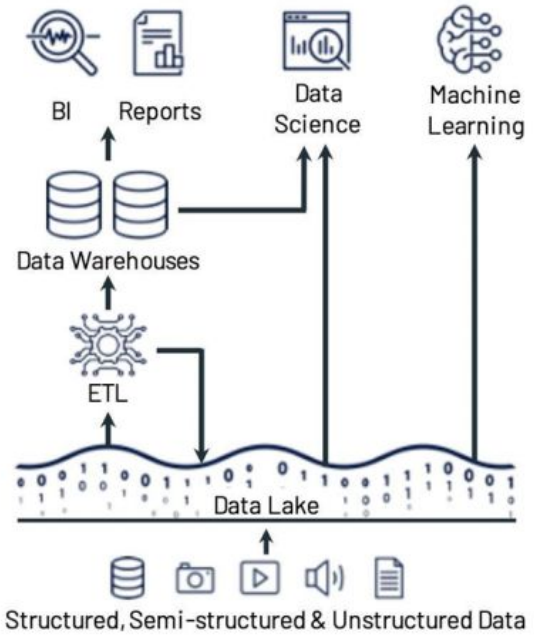
- Needs Extract Transform Load (ETL)
- Schema-on-write design
- Underlying layer is database technology
- Better data management when writing
- Limited to commonly used analysis scenarios
- Cannot support semi-structured and unstructured data



Big Data Platform Architecture Evolution

Data Lake + Data Warehouse

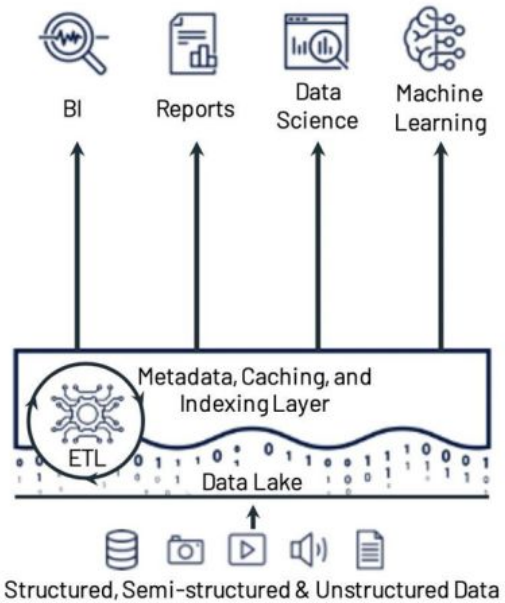
- Open data format
- Support for structured/unstructured data
- Schema-On-Read design
- Complex data governance
- Complex database management
- Need for performance optimization



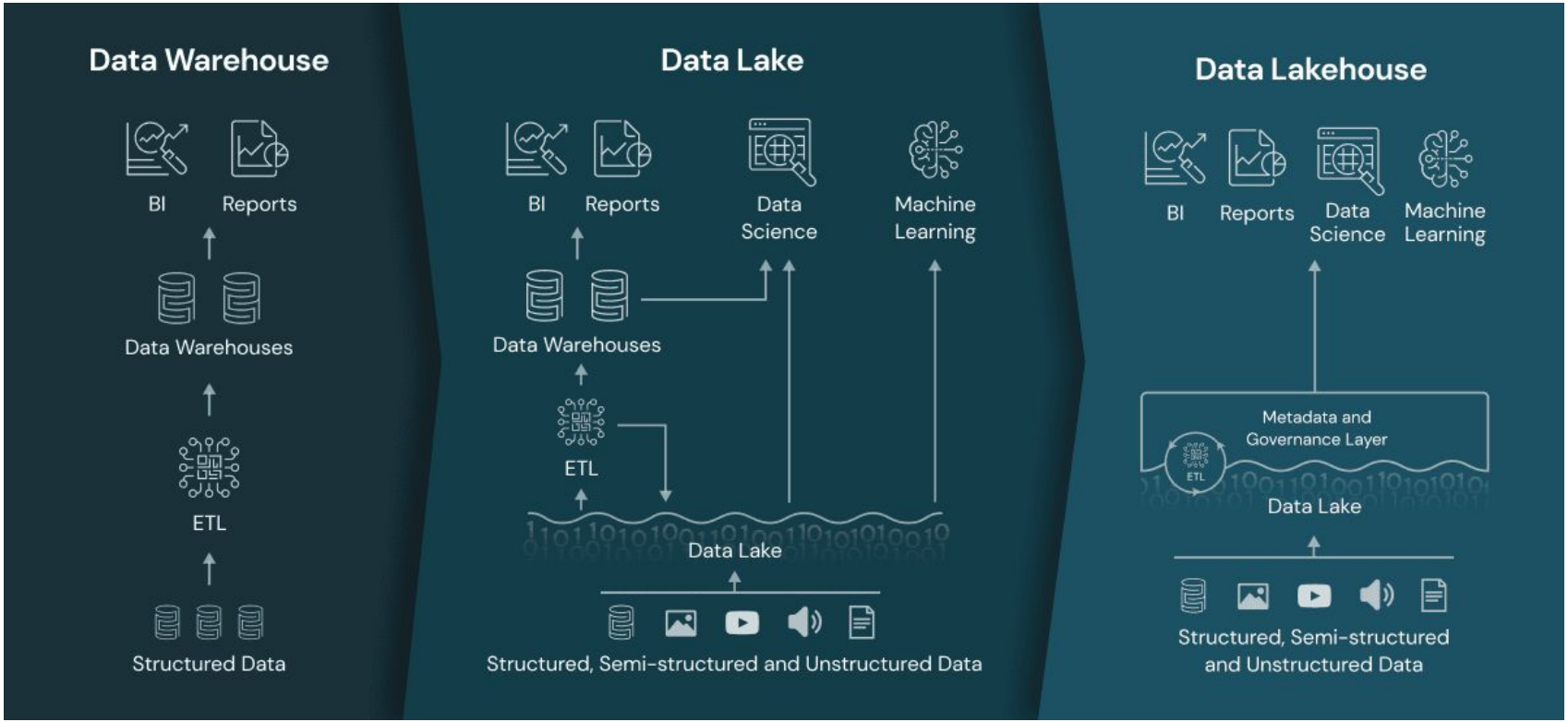
Big Data Platform Architecture Evolution

Lake House

- Abstracts transaction management layer on top of data lake
- Provides Data Management features of data warehouse
- Optimizes data performance on cloud object storage
- Supports complex analysis big data scenarios
- Supports both batch and stream processing

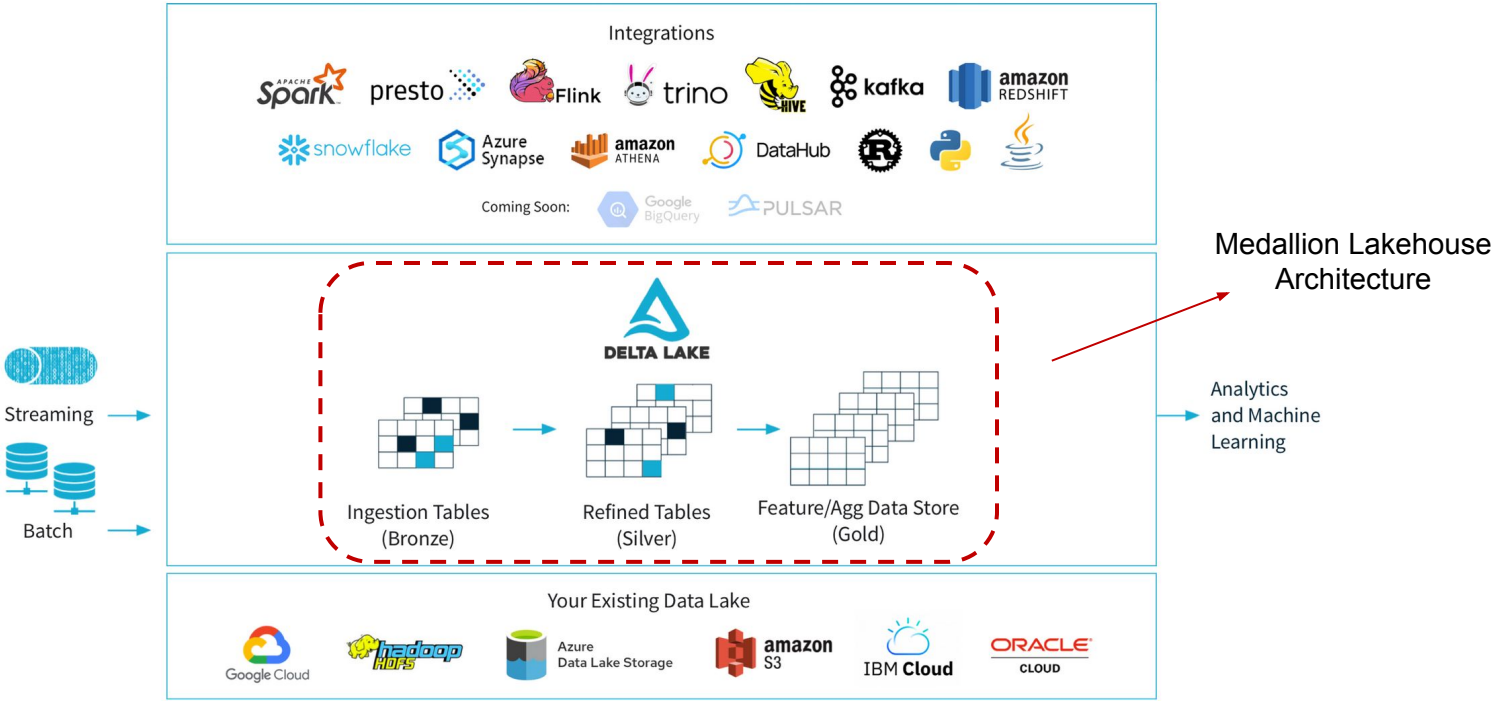


Data Warehouse vs. Data Lake vs. Data Lakehouse

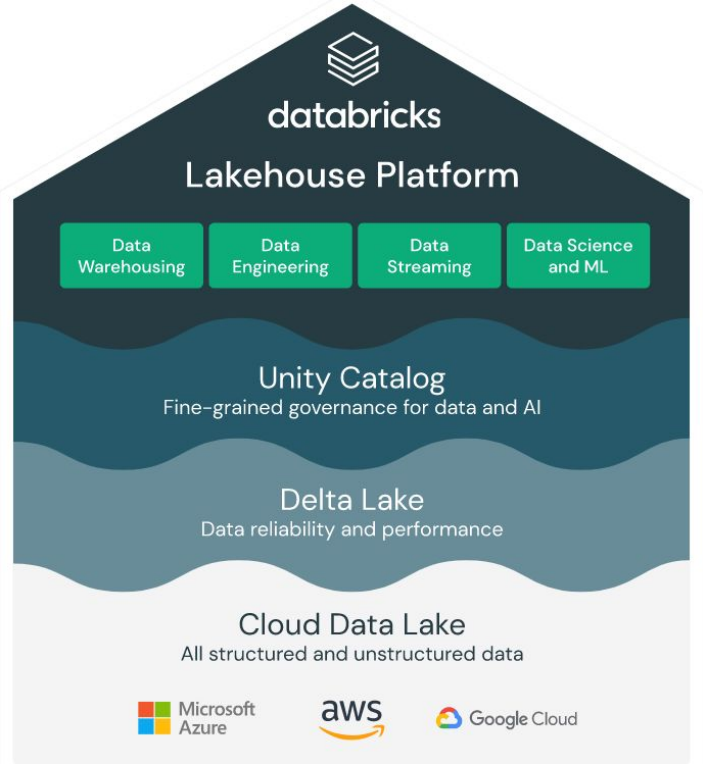


Big Data Platform Architecture Evolution

Delta Lake, an Open-Source Project is Born

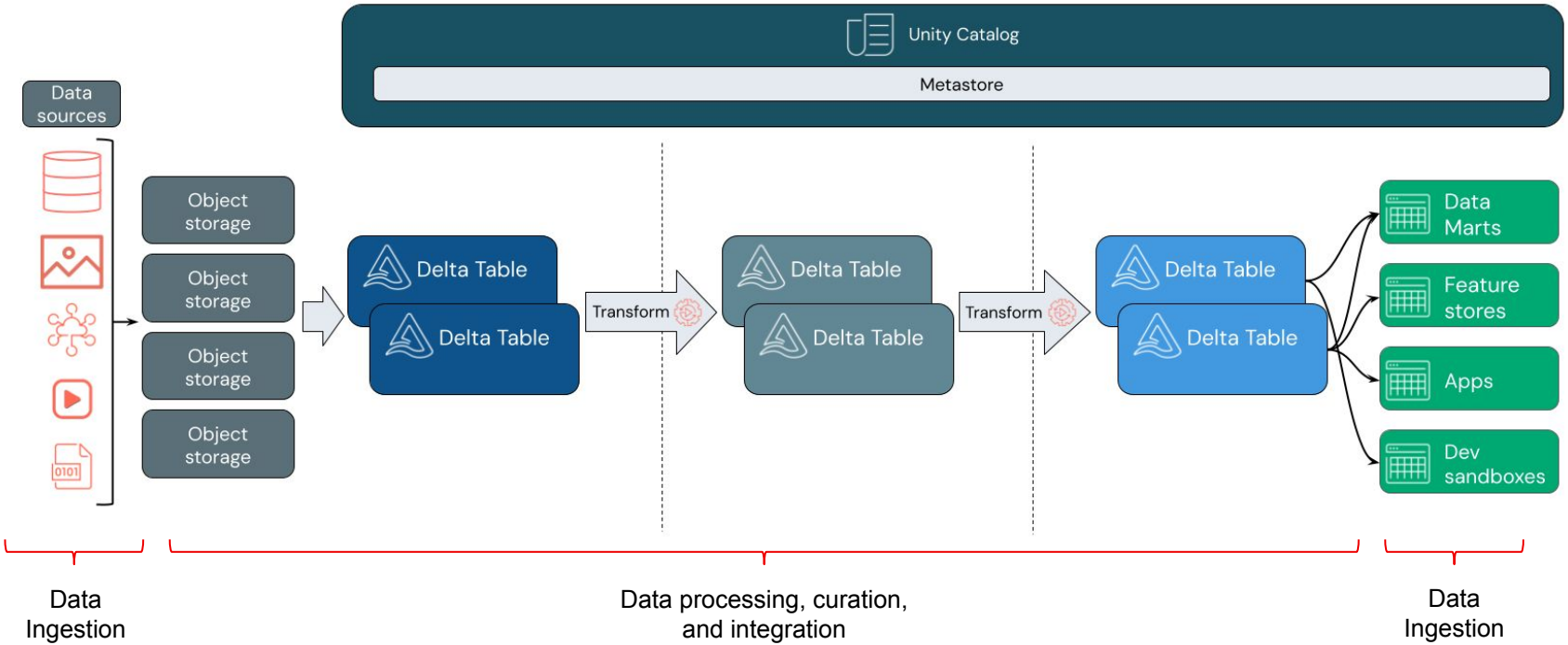


What is Databricks Lakehouse?



- Databricks is a unified, open analytics platform for building, deploying, sharing, and maintaining enterprise-grade data, analytics, and AI solutions at scale.
- Founded by creators of Apache Spark
- Based on
  - Delta Lake - optimized storage layer, supports ACID transactions, schema enforcement
  - Unity Catalog - unified, fine-grained governance solution for data and AI

What is Databricks Lakehouse?

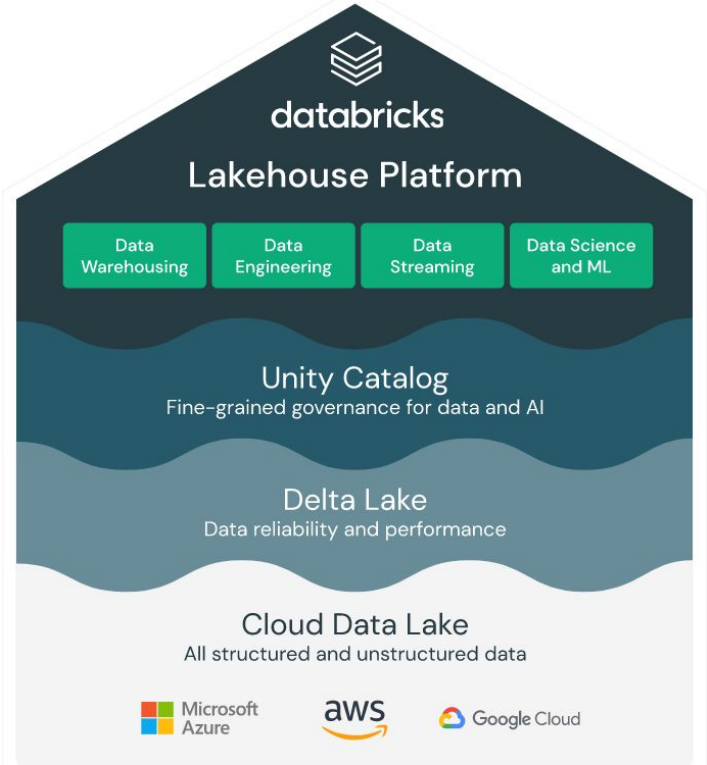


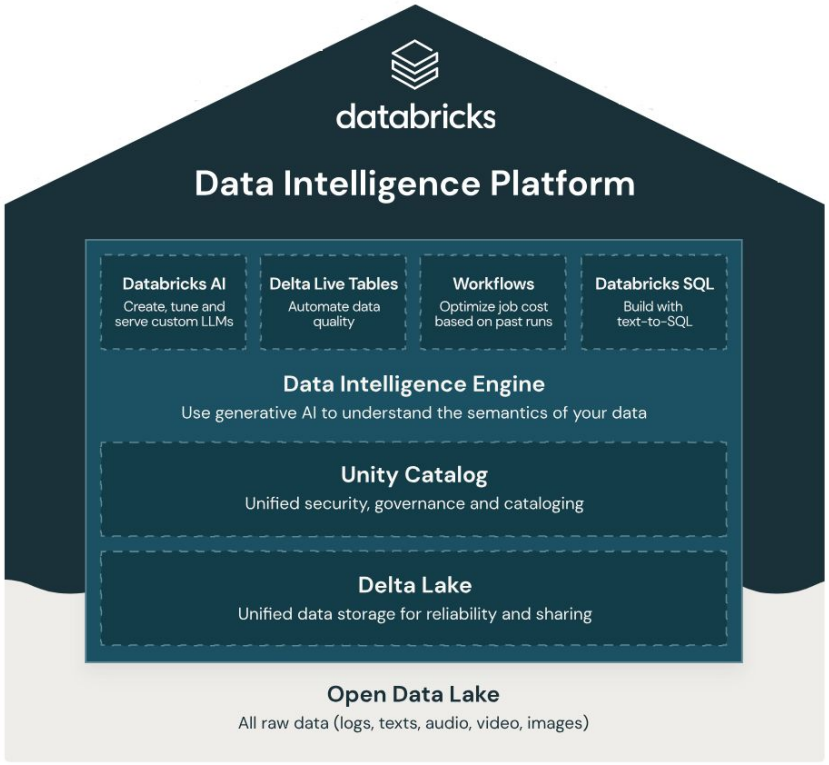
Databricks Lakehouse Performance

Lightning-Fast performance

- Delta Lake on Databricks delivers massive scale and speed, with data loads and queries running up to 1.7x faster than with other storage formats.

Databricks Lakehouse as Data Intelligence Platform





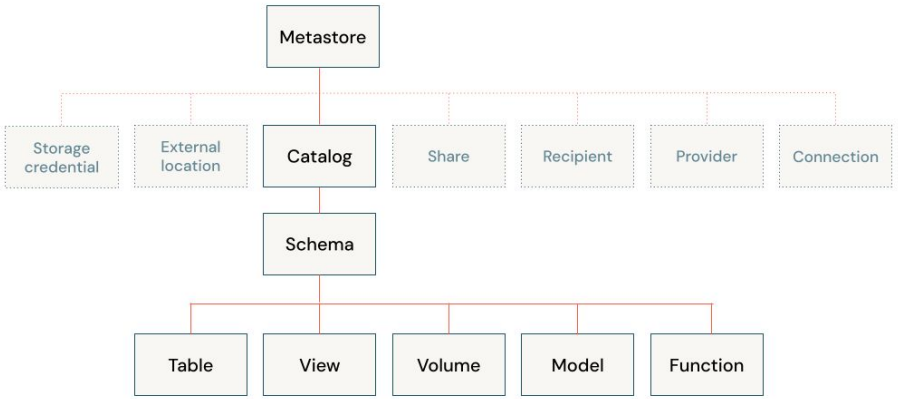
Features and Functionalities

- Control transactions and data access
- Manage access to production data
- Leverage views
- Share data with collaborators
- Manage permissions at scale
- Easy data discovery
- Accelerate time to production

ACID Guarantees on Databricks

- **Atomicity** - All transactions either succeed or fail completely.
- **Consistency** - How a given state of the data is observed by simultaneous operations.
- **Isolation** - Simultaneous operations potentially conflict with one another.
- **Durability** - Committed changes are permanent.

Data Objects in Databricks



- **Catalog:** Grouping of databases.
- **Schema:** Grouping of objects in a catalog. Contains tables, views, and functions.
- **Table:** Collection of rows and columns stored as data files in object storage.
- **View:** Saved query typically against one or more tables or data sources.
- **Function:** Saved logic that returns a scalar value or set of rows.

Integrations

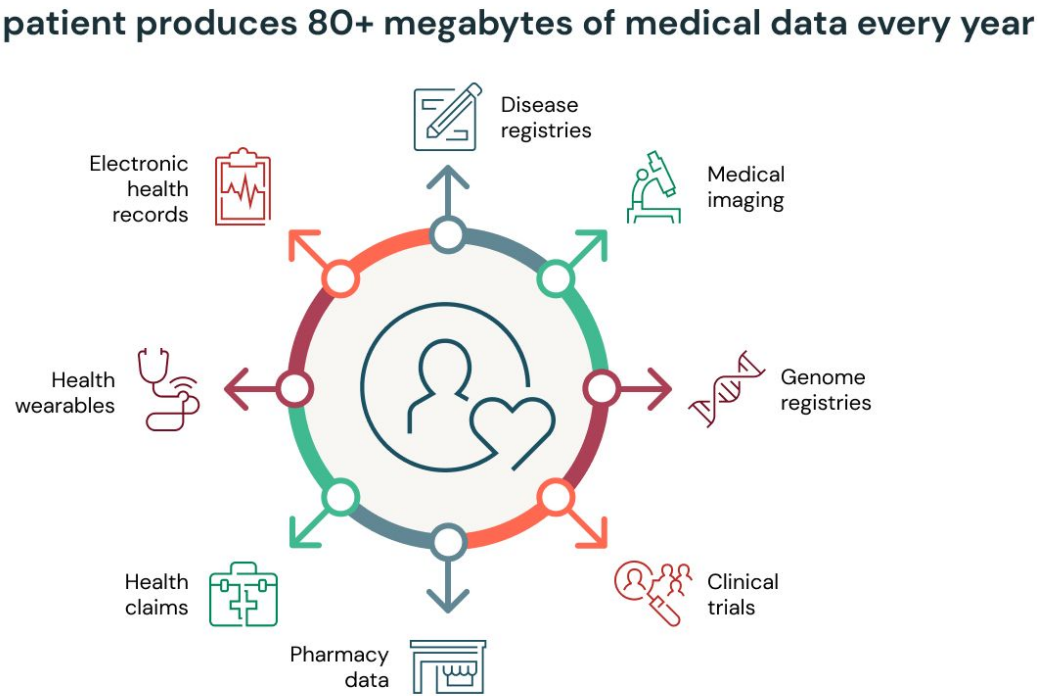
Integrations	Description
Partner Connect	User interface that allows to integrate more quickly and easily with Databricks clusters and SQL warehouses.
Formats	CSV, Delta Lake, JSON, Parquet, XML etc.
Data Storage	Amazon S3, Google BigQuery etc.
Platforms	Snowflake, and other providers
BI & Visualization	Power BI, Tableau etc.
ETL/ELT tools	dbt, Prophecy, and Azure Data Factory
Data pipeline orchestration	Airflow
SQL database	DataGrip, DBeaver, and SQL Workbench
IDEs & Dev Tools	DataGrip, IntelliJ, PyCharm, Visual Studio Code, Git

Capabilities of Databricks Lakehouse?

Enterprise Data Lakehouse	Combines the strengths of enterprise data warehouses and data lakes to accelerate, simplify, and unify enterprise data solutions.
ETL and Data Engineering	Backbone for data-centric companies by making sure data is available, clean, and stored in data models that allow for efficient discovery and use
ML, AI, and data science	Databricks machine learning expands the core functionality of the platform with a suite of tools tailored to the needs of data scientists and ML engineers
Warehousing, analytics, and BI	User-friendly UIs with cost-effective compute resources and infinitely scalable, affordable storage to provide a powerful platform for running analytic queries
Data gov. secure data sharing	The lakehouse makes data sharing within your organization as simple as granting query access to a table or view.
DevOps, CI/CD, Orchestration	Easy development lifecycles for ETL pipelines, ML models, and analytics dashboards along with versioning, automating, scheduling, deploying code and production resources
Real-time streaming analytics	Databricks leverages Apache Spark Structured Streaming to work with streaming data and incremental data changes.

Use Case

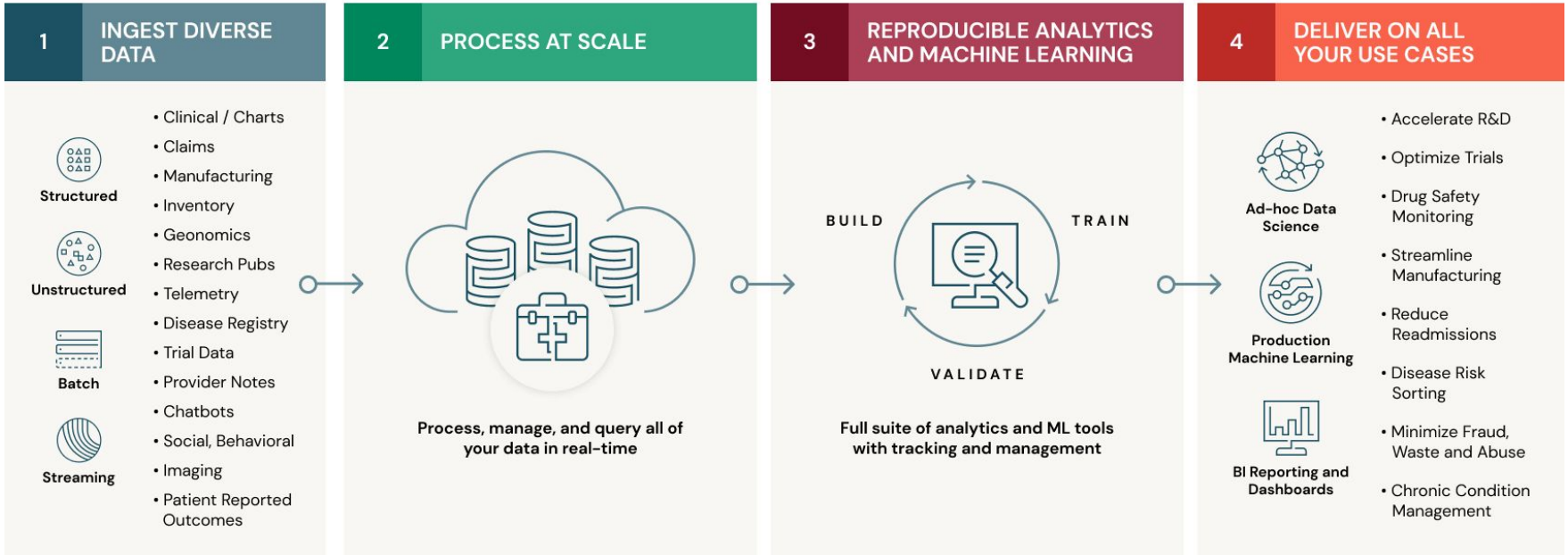
Unlocking the Power of Health Data With Databricks



Use Case

Unlocking the Power of Health Data With Databricks

Building a Lakehouse for Healthcare and Life Science



Q&A

Lesson Review

Q. Which among Data Warehouse, Data Lake and Delta Lake cannot support semi structured, unstructured data?

A. Data Warehouse

Q. Which component of Databricks Lakehouse Platform is responsible for Data reliability and performance?

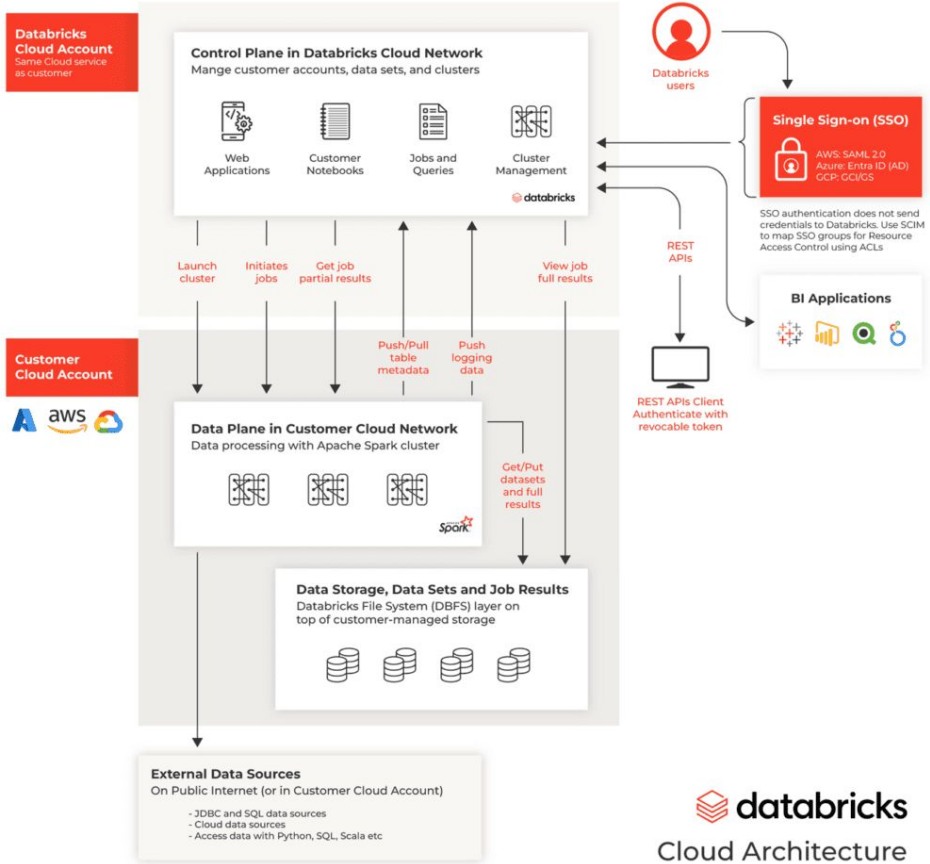
A. Delta Lake

Q. Which component of Databricks Lakehouse Platform is responsible for Data & AI governance?

A. Unity Catalog

Databricks UI

Databricks and Cloud Providers



- You can configure features to control access and provide private connectivity between users and their Azure Databricks workspaces
- Classic compute resources, such as cloud clusters connect to the control plane.
- Configure private and dedicated connections from serverless compute to storage in the serverless compute plane.

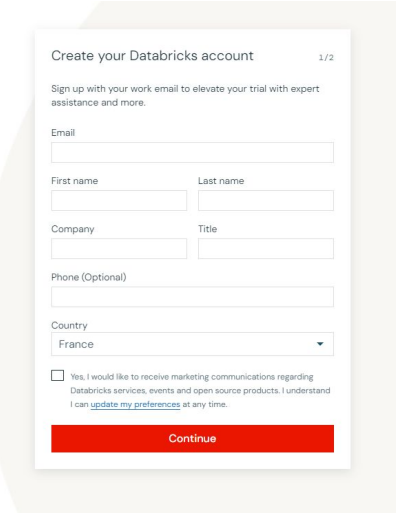
Databricks Community Edition

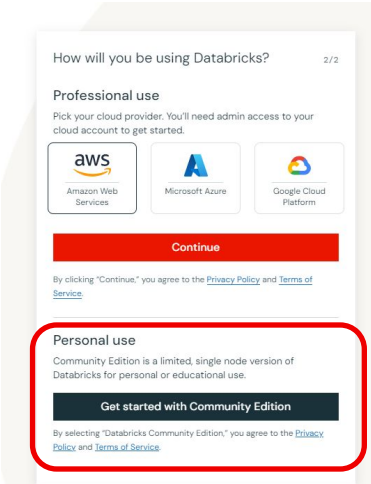
- Free cloud-based big data platform
- Features micro-cluster, cluster manager, notebook environment
- Users can share and host notebooks for free
- Rich portfolio of training resources available
- Ideal for developers, data scientists, data engineers, IT professionals
- Hosted on Amazon Web Services
- No AWS costs incurred with Community Edition

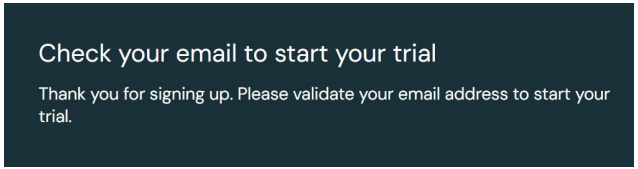
Databricks Community Edition

- Sign up here
- Click on "Get started with Community Edition" for "Personal Use"
- Validate your email address to start your Databricks Community Edition trial
- Login to your account from community.cloud.databricks.com/login.html

Instructions are also provided in the Bitbucket repository



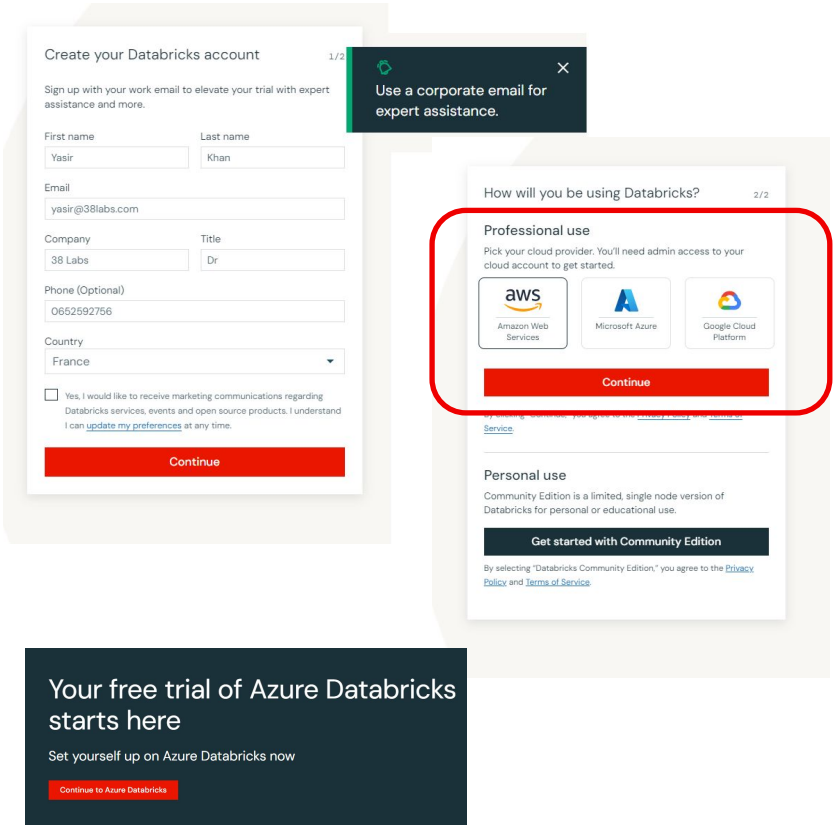




Lab/Demo: Databricks Workspace with Community Edition

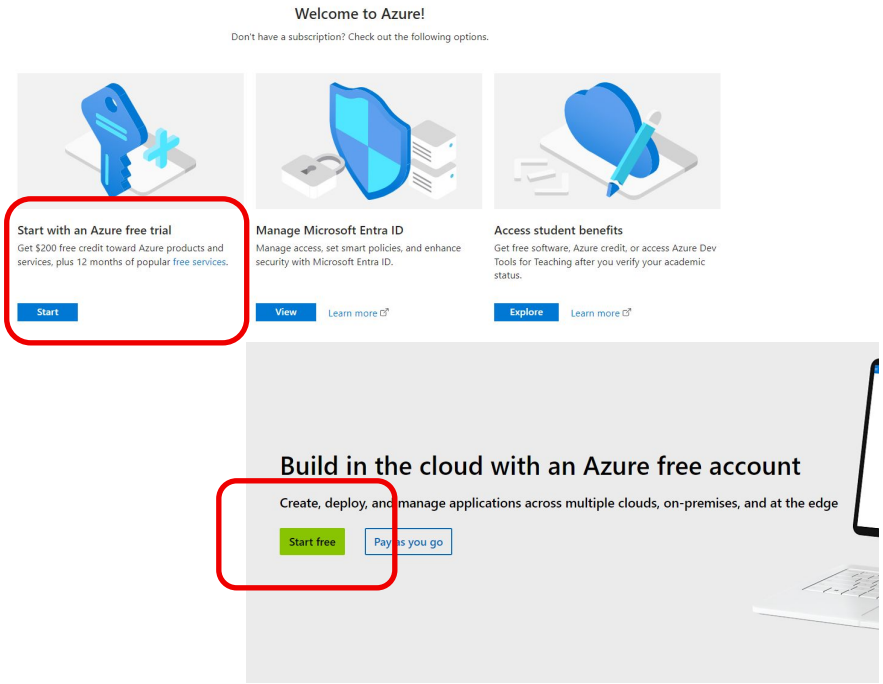
Databricks On Azure Cloud

- Sign up here
- Click on your choice of cloud provider under "Professional Use“ and click “Continue”
- Click on Continue to “Azure Databricks”



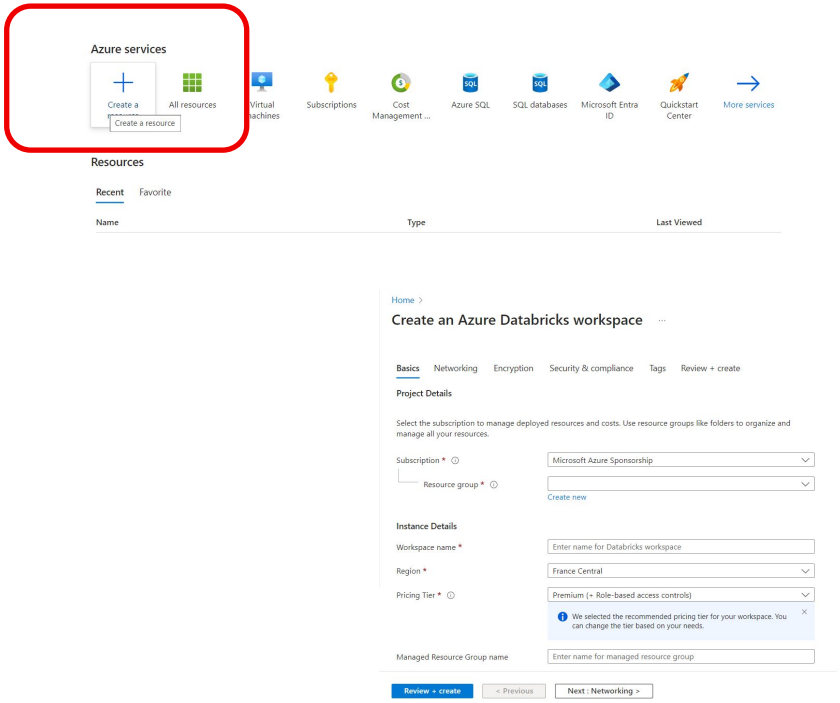
Databricks On Azure Cloud

- If you don’t have any Azure credits, you will be taken to the Azure startup page to sign up for free trial
- Otherwise you will be take to the page to create resources



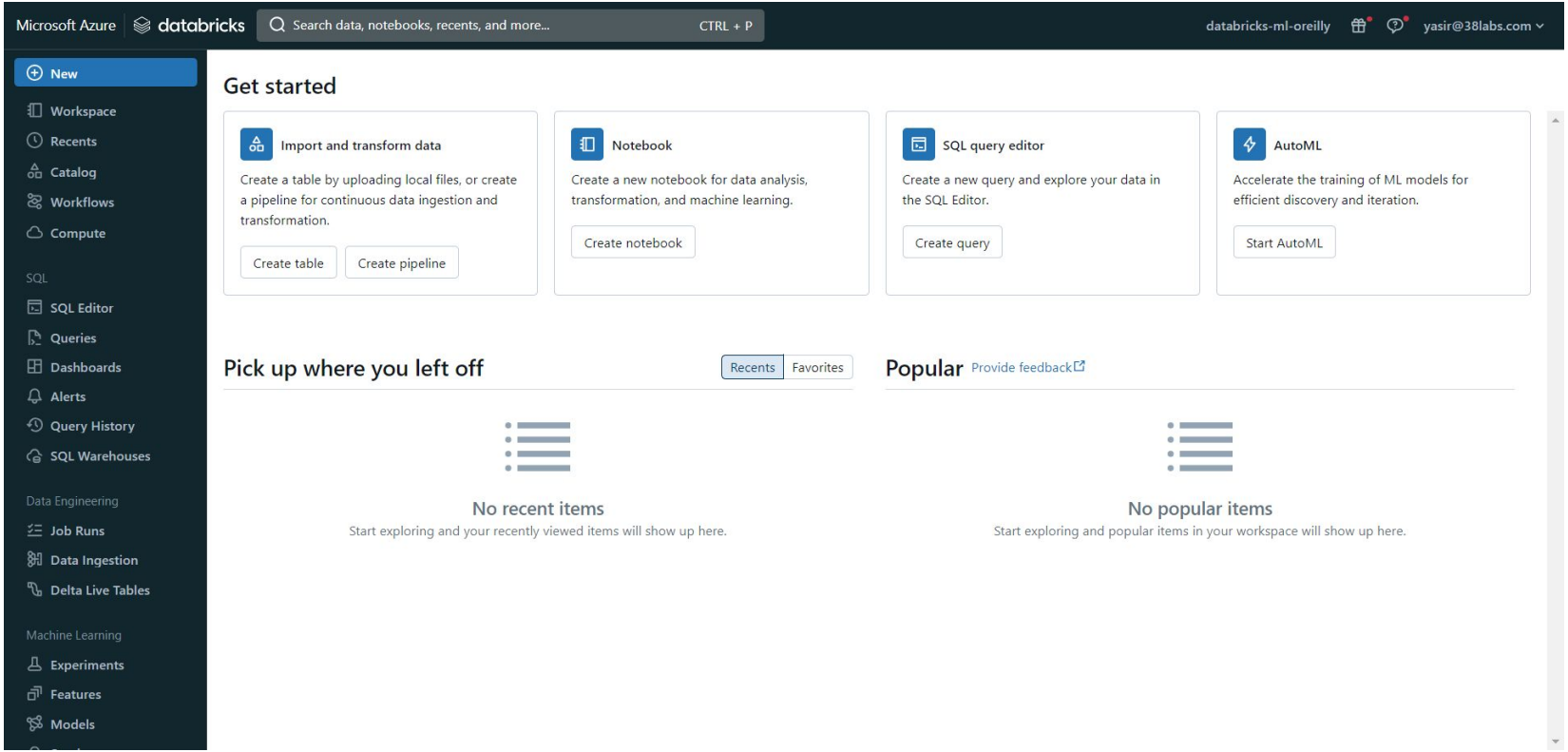
Databricks On Azure Cloud

- Start by creating resources and select Azure Databricks as the resource type



Lab/Demo: Databricks Workspace with Azure Cloud

Databricks UI



Lab/Demo: Databricks UI Overview

Q&A

Lesson Review

Q. What is the primary role of Databricks Control Plane?

A. Manages customer accounts, data sets and clusters

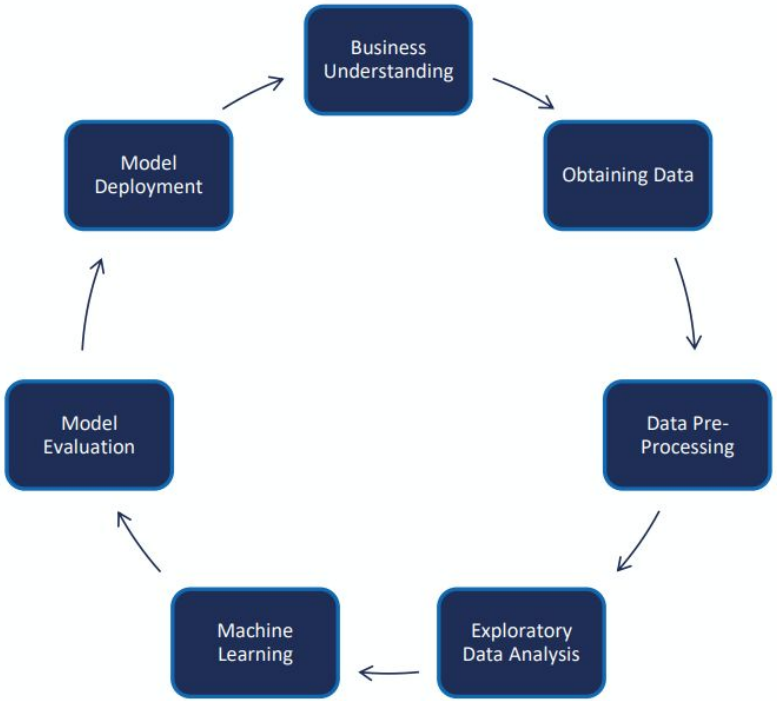
Q. Which are the 3 largest providers on which Databricks can be used?

A. AWS, GCP and Microsoft Azure

Break (5 min)

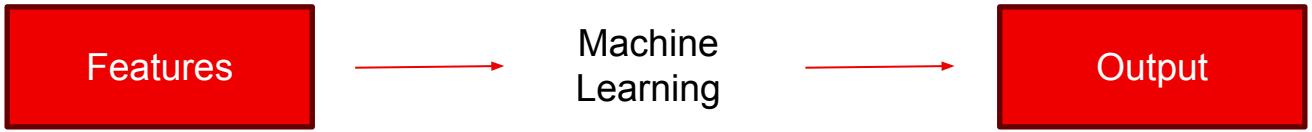
Databricks Machine Learning
Key Concepts

Machine Learning Workflow



What is Machine Learning?

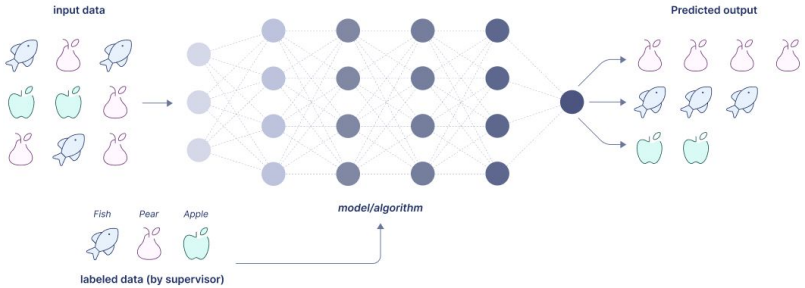
- Build **patterns** and **relationships** in your data without explicitly programming them
- Derive an approximate function to map **features** to an **output** or relate them to each other



Types of Machine Learning?

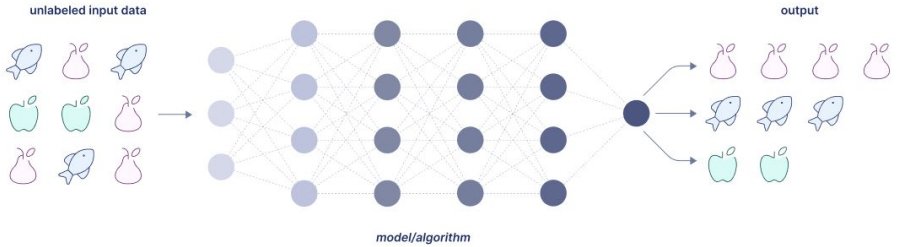
Supervised Machine Learning

- Labeled data (known function output)
- Regression (continuous/ordinal-discrete output)
- Classification (categorical output)



Unsupervised Machine Learning

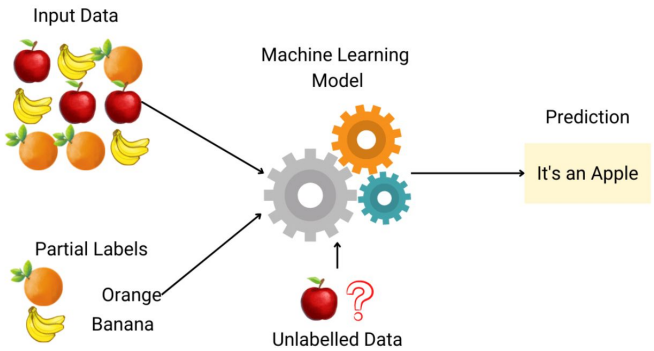
- Unlabeled data (no known function output)
- Clustering (categorize records based on features)
- Dimensionality reduction (reduce feature space)



Types of Machine Learning?

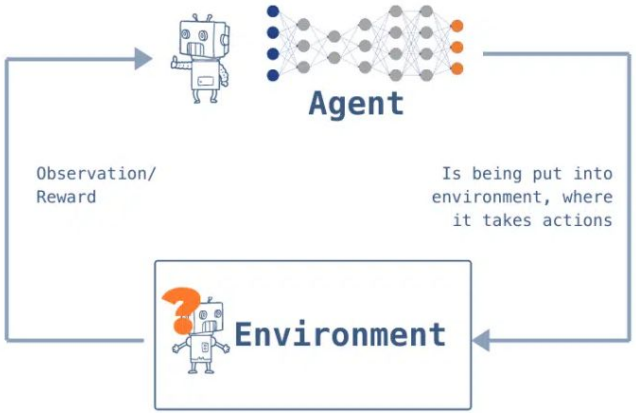
Semi-supervised Learning

- Labeled and unlabeled data, mostly unlabeled
- Combines supervised and unsupervised learning
- Commonly trying to label the unlabeled data to be used in another round of training

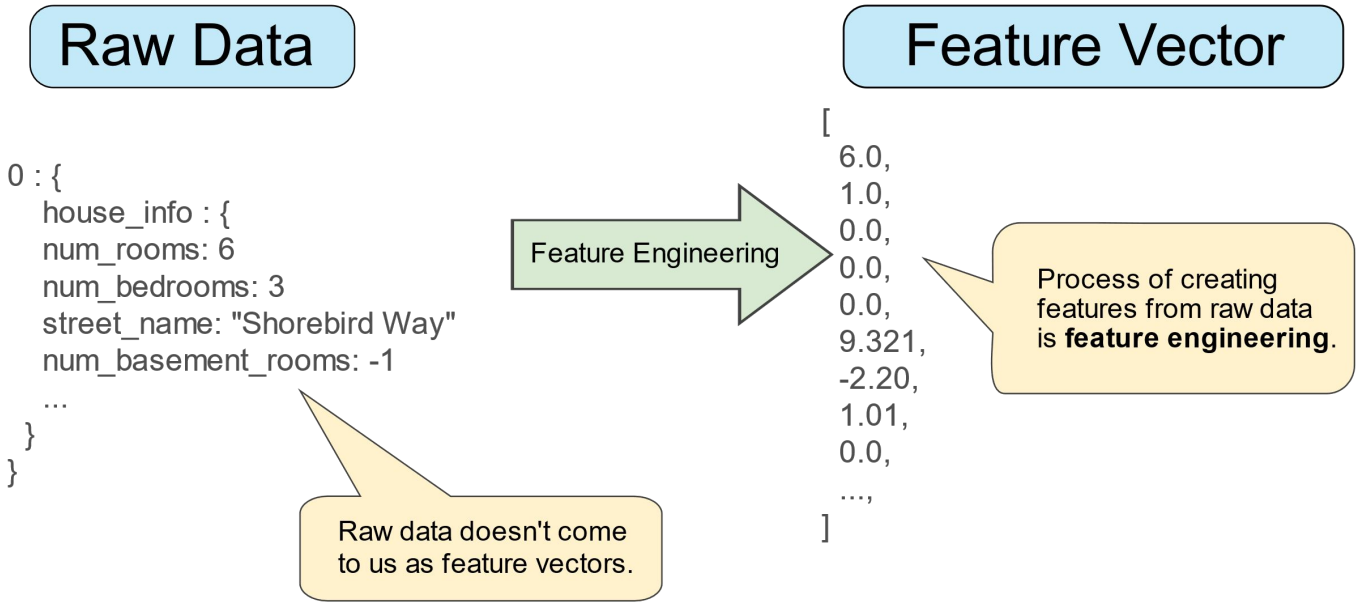


Reinforcement Learning

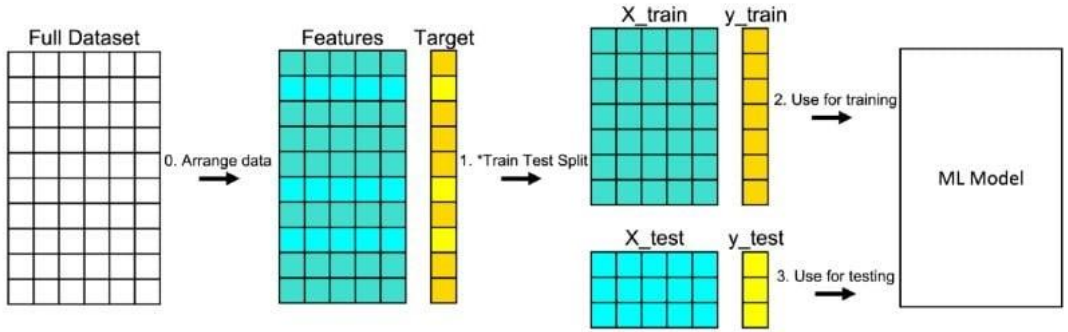
- States, actions, and rewards
- Useful for exploring spaces and exploiting information to maximize expected cumulative rewards
- Frequently utilizes neural networks and deep learning

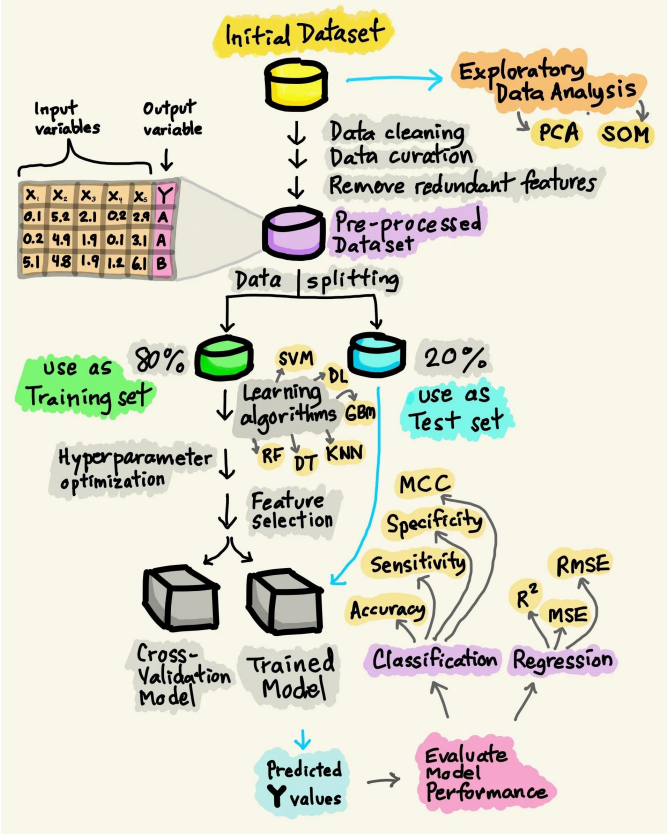


Feature Engineering

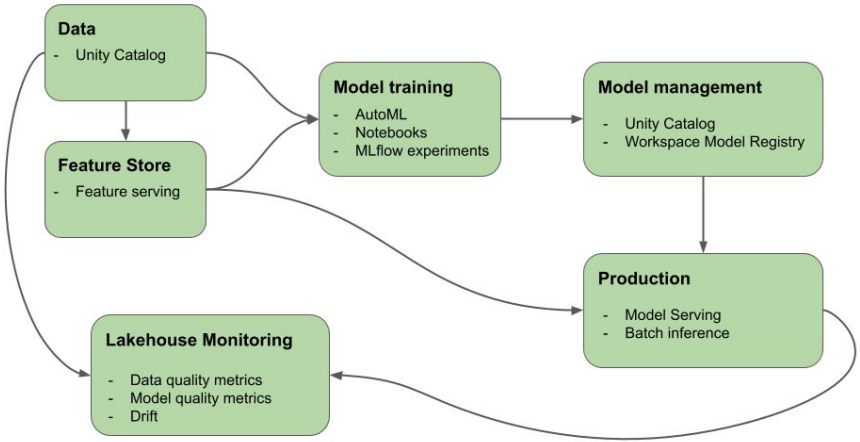


Building and Evaluating ML Models



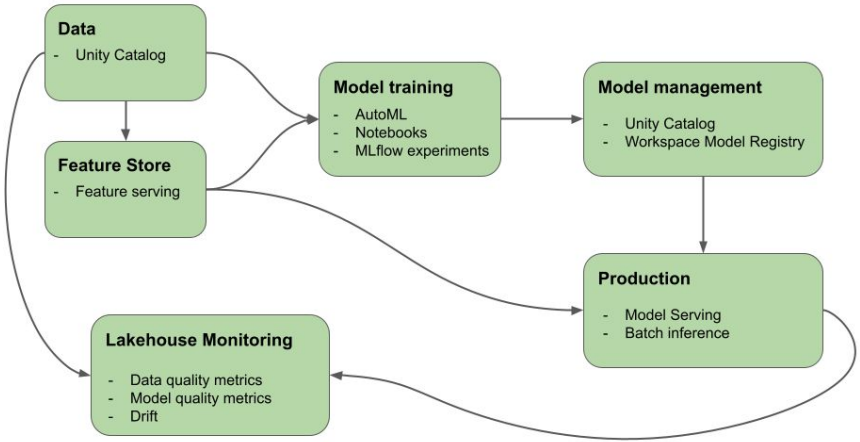


AI and Machine Learning on Databricks



- Unity Catalog - governance, discovery, versioning, and access control for data, features, models, and functions.
- Lakehouse Monitoring - data monitoring.
- Feature engineering and serving.
- Databricks Workflows for automated workflows and production-ready ETL pipelines.
- Databricks Git folders for code management and Git integration.

AI and Machine Learning on Databricks



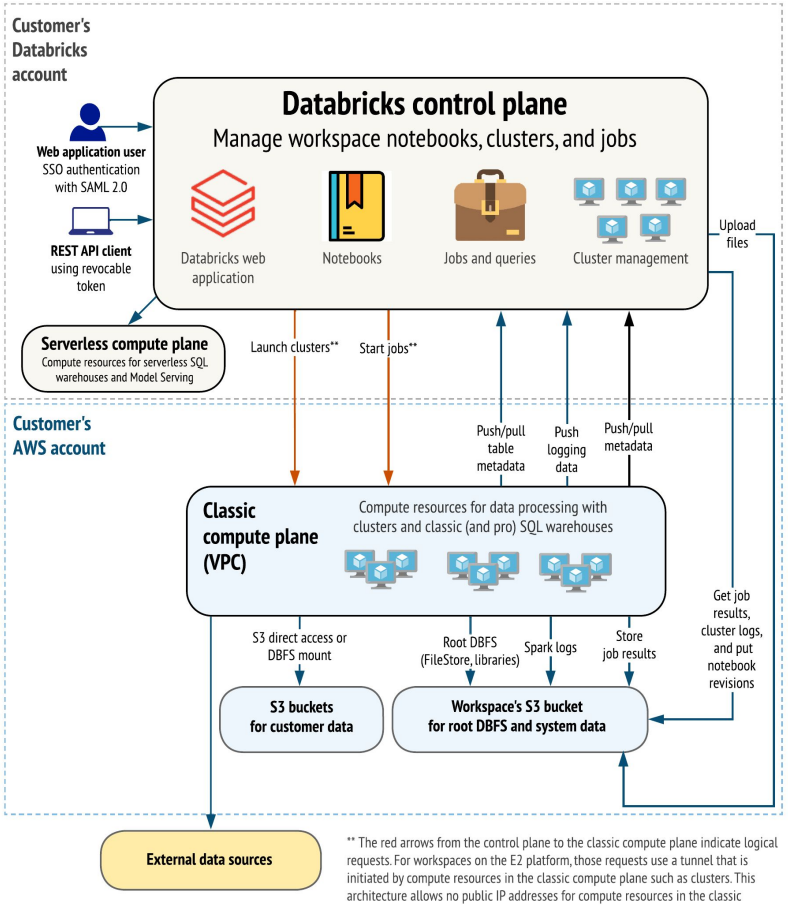
Support for the model lifecycle:

- Databricks AutoML for automated model training.
- MLflow for model development tracking.
- Unity Catalog for model management.
- Databricks Model Serving for high-availability, low-latency model serving. LLMs support using:
  - Foundation Model APIs to access/query state-of-the-art open models from a serving endpoint.
  - External models to access models hosted outside of Databricks.
- Lakehouse Monitoring to track model prediction quality

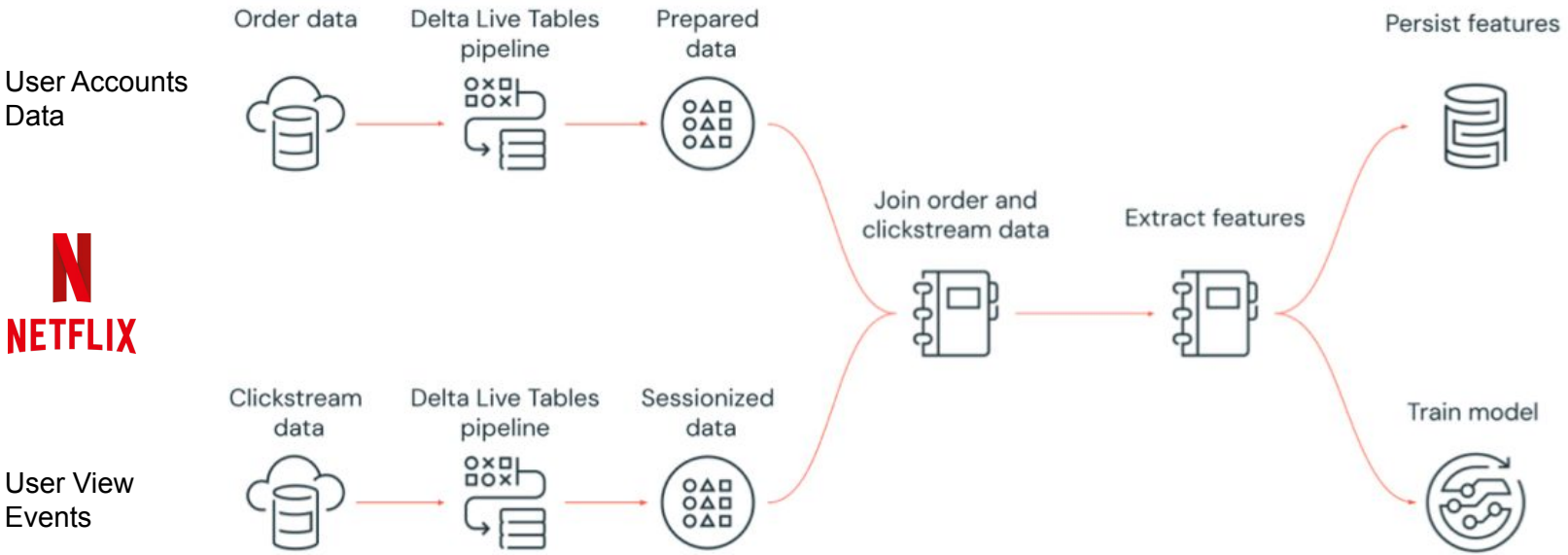
Databricks Architecture

Databricks operates out of

- Control Plane: Includes the backend services that Databricks manages in your Databricks account.
- Compute Plane: Where data processing occurs
  - Classic Compute Plane
  - Serverless Compute Plane



An Example Databricks Workflow



Q&A

Lesson Review

Q. What are the 2 primary types of Machine Learning?

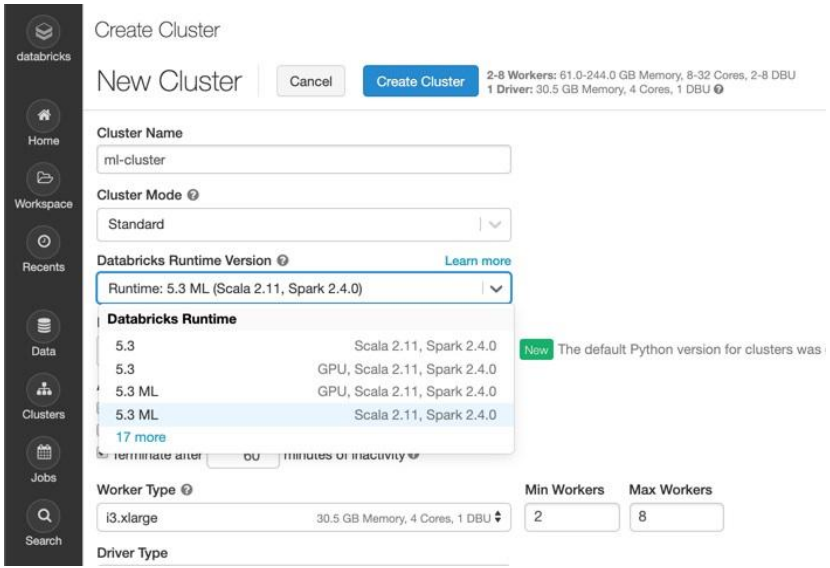
A. Supervised and Unsupervised Machine Learning

Q. What are the primary differences between a Control plane and Compute Plane?

A. Control Plane includes the backend services that Databricks manages in your Databricks account. Compute Plane is where data processing occurs

Databricks Runtime for Machine Learning

Databricks Runtime for AI and ML



- Ready-to-use clusters with built-in ML and AI frameworks
- Automates cluster creation with pre-built machine learning and deep learning infrastructure including the most common ML and DL libraries
- Built-in popular AI libraries like Hugging Face Transformers and LangChain
- Built-in popular ML libraries such as TensorFlown PyTorch, Keras and XGBoost
- Compatible versions of installed libraries

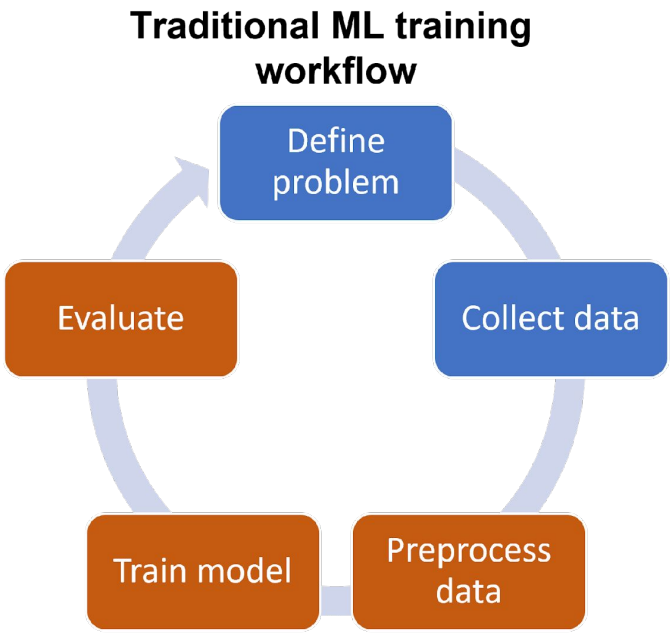
Lab/Demo: Creating Databricks ML Cluster

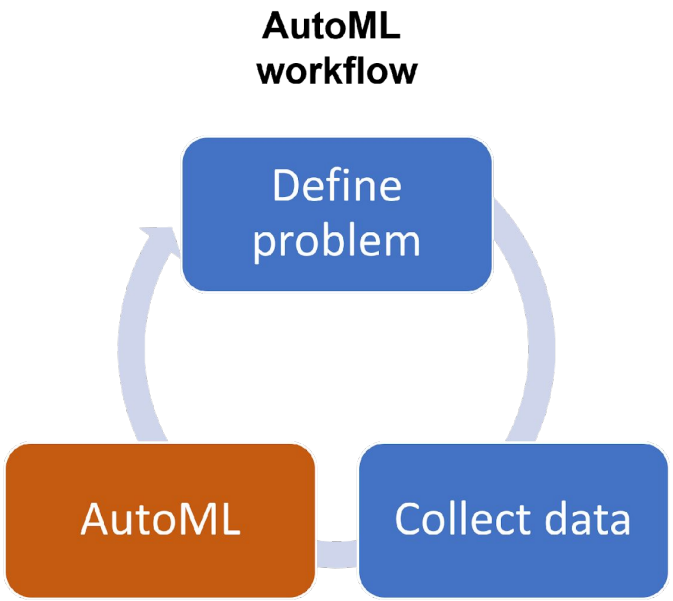
Lab/Demo: Explore Cluster Features from UI

Break (5 min)

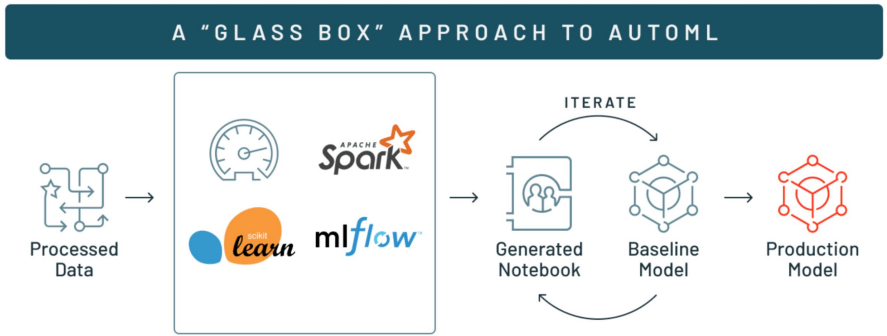
Classification, Regression and Forecasting with AutoML

Databricks AutoML



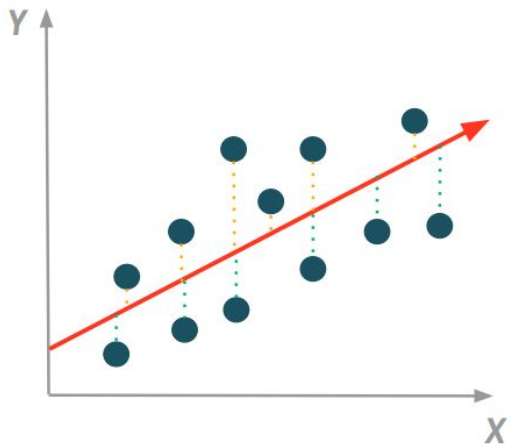


Advantages of Using Databricks AutoML



- Existing AutoML tools are opaque - users unaware of how models are trained.
- Especially difficult for domain-specific modifications e.g., auditability for regulatory compliance
- Reverse engineering these opaque models consumes time and resources
- Databricks AutoML offers a transparent approach, Auto-generating Python notebooks for every trained model.
- Data scientists can leverage their expertise to customize or add cells to these notebooks.

Linear Regression

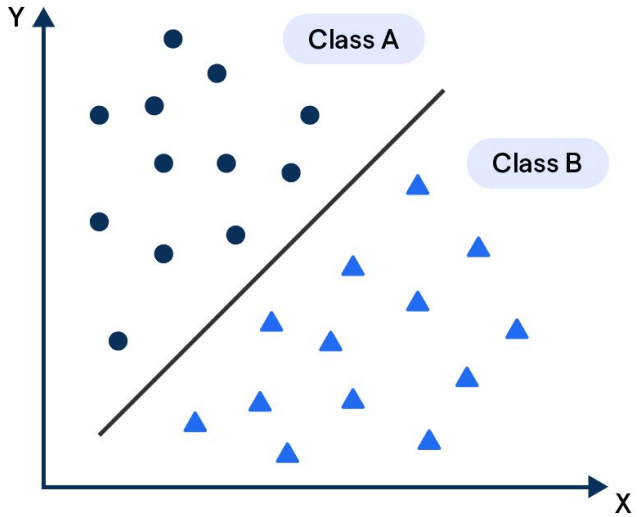


- A supervised machine learning method
- Provides a linear relationship between an independent variable and a dependent variable to predict the outcome of future events

Using **Databricks AutoML** our goal will be to train a **Linear Regression Model** - draw a line that minimizes the sum of the squared residuals.

Lab/Demo: Linear Regression Using AutoML

Classification

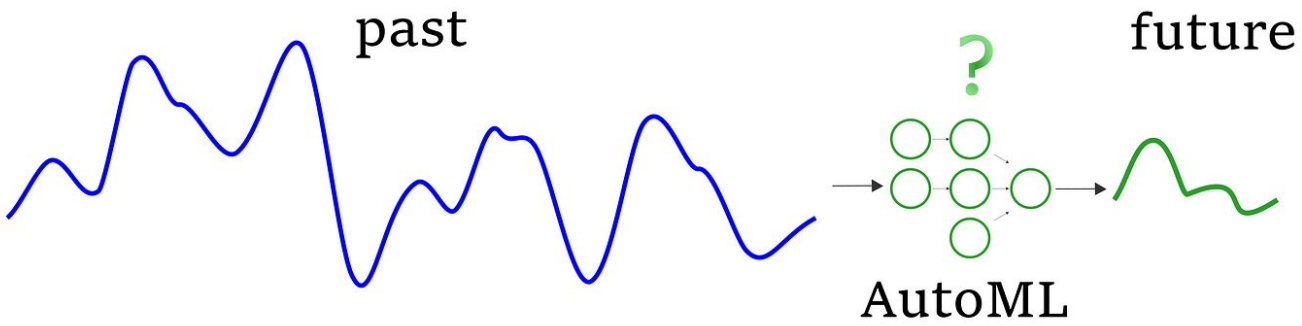


- A supervised machine learning method
- Provides a class relationship and is used to predict the category or class of an object based on its features

Using **Databricks AutoML** our goal will be to train a **Classification Model** - draw a line that minimizes the sum of the squared residuals.

Lab/Demo: Classification Using AutoML

Forecasting



Using **Databricks AutoML** our goal will be to train a **Forecasting Model** – a machine learning model capable of predicting time-series data

Lab/Demo: Forecasting Using AutoML

Q&A

Lesson Review

Q. Name at least three tasks that Databricks AutoML can automate

A. Feature engineering, Model selection, Hyperparameter tuning

Q. How is Databricks AutoML different from other AutoMLs?

A. Databricks AutoML offers a transparent approach, Auto-generating Python notebooks for every trained model.

Q. List 3 Machine Learning Algorithms handled by AutoML?

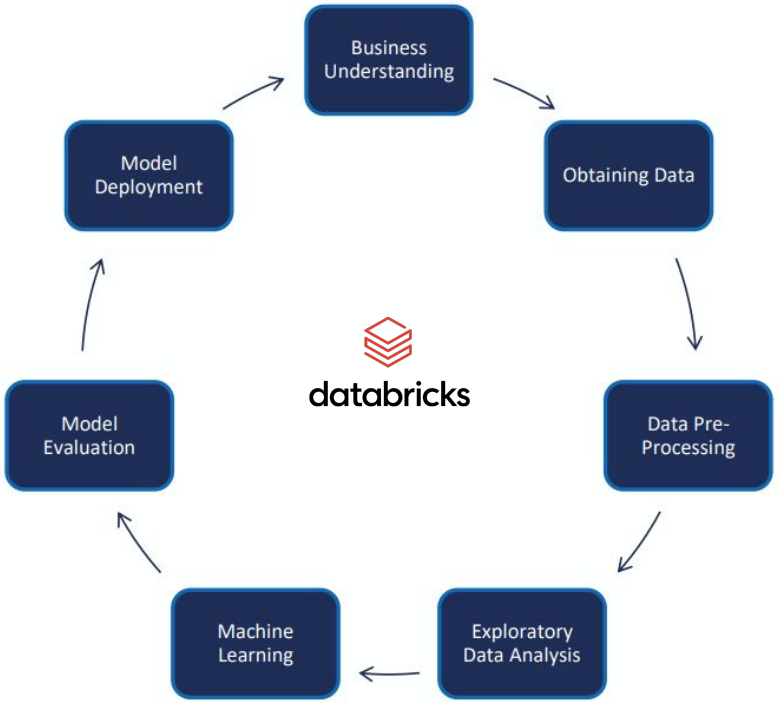
A. Linear Regression, Classification, and Forecasting

End of Day 1

Day 2
Advanced ML with Databricks

Today's Schedule & Learning Objectives

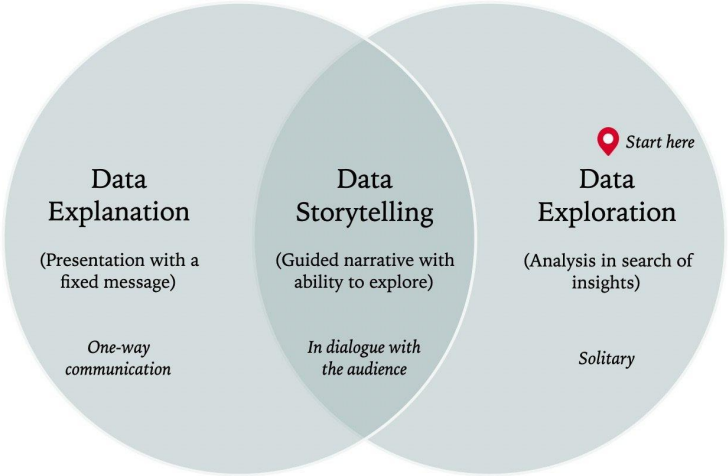
- Exploratory data analysis
- Feature engineering
- Feature Store
- Managed MLflow
- MLflow model registry



Exploratory Data Analysis

Data Exploration

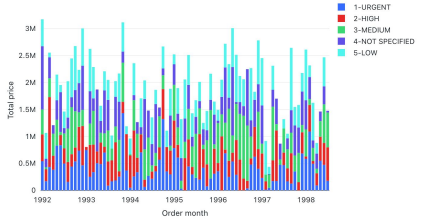
- EDA explores data, revealing characteristics and quality.
- Uses statistics and visualizations to inform preparation and algorithm selection.
- Ensures data readiness, influences ML model success.
- Two tools for EDA in Databricks
  - Using Databricks SQL
  - Using Databricks Runtime – using notebooks

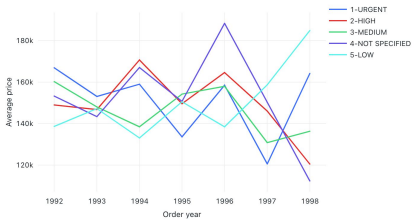


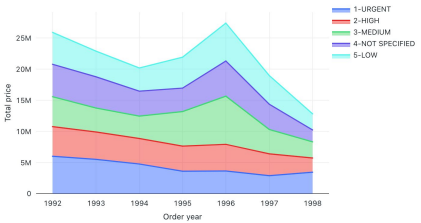
Lab/Demo: Data Visualization in Databricks SQL

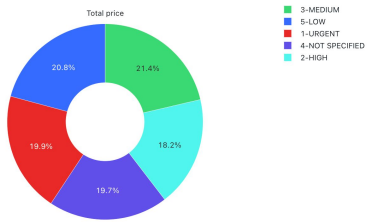
Lab/Demo: Data Visualizations in Notebooks

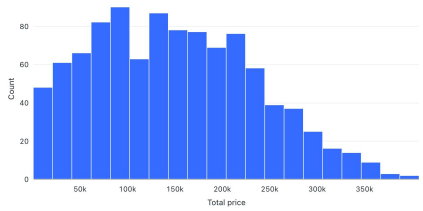
Visualization Types

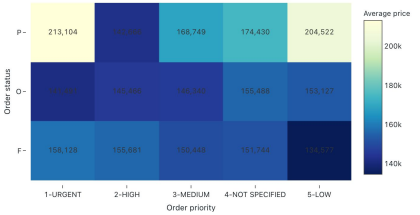


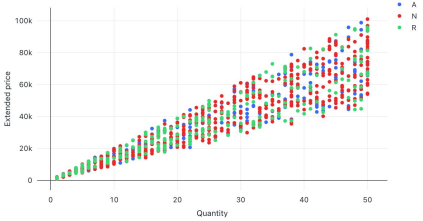


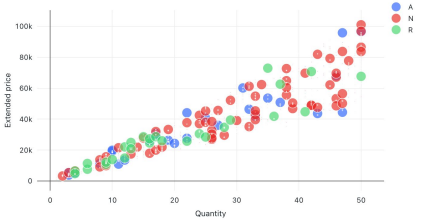


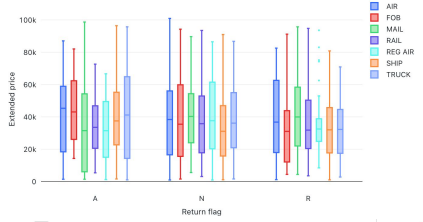


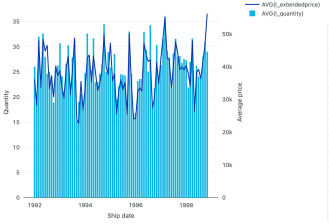


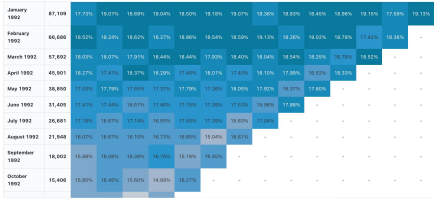




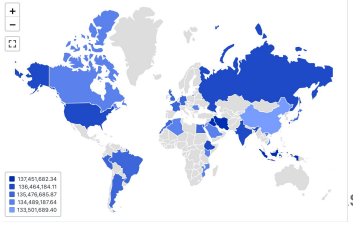


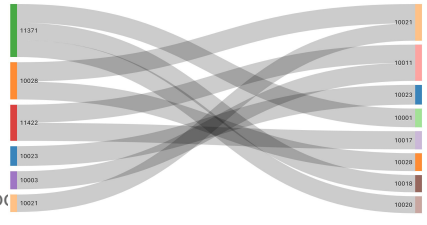


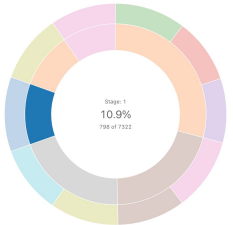


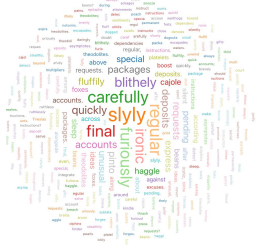


Steps	Value	% Max	% Previous
F	549,181,919,365.27	100%	100%
O	548,923,692,869.55	99.95%	99.95%
P	35,333,603,011.43	6.43%	6.44%









Data Profiling

- Data profiling - process of collecting statistics and summaries of data to assess its quality and other characteristics.
- A well-rounded data profiling process encompasses four main components:
  - Data Overview - e.g., number of records
  - Univariate Analysis & Feature Statistics – e.g., range, mean, median of a numeric feature
  - Multivariate Analysis & Correlation Assessment – e.g., correlation between 2 features
  - Data Quality Evaluation – e.g., alerts such as high correlation, class imbalance

Lab/Demo: Data Profiling with ydata-profiling

Q&A

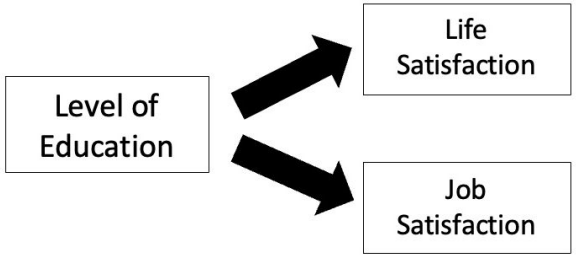
Lesson Review

Q. What are the two tools that can be used for EDA in Databricks?

A. Databricks SQL and Databricks Runtime (using notebooks)

Q. Researchers are examining the impact of Level of Education on two outcomes Life Satisfaction and Job Satisfaction. Is this an example of Multivariate Analysis or Univariate Analysis?

A. Multivariate Analysis. The results can tell us whether an individual who completed graduate school showed higher life AND job satisfaction than an individual who completed only high school or college.



Feature Engineering

Lab/Demo: Missing Value Imputation

Lab/Demo: Outlier Removal

Lab/Demo: Feature Creation

Lab/Demo: Feature Scaling

Lab/Demo: One-hot Encoding

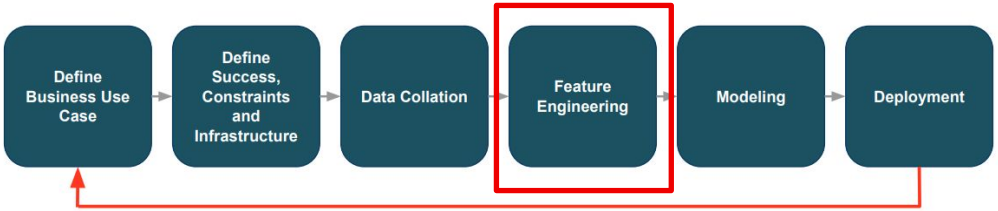
Lab/Demo: Feature Selection and Transformation

Lab/Demo: Dimensionality Reduction

Break (5 min)

Introduction to Feature Store

Feature Store

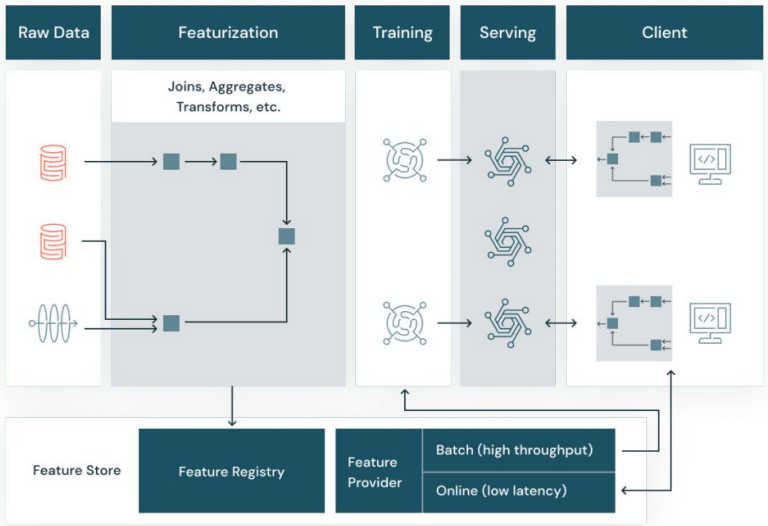


- Features are crucial for machine learning model performance.
- Feature engineering transforms raw data into predictive features.
- How to manage features? Inconsistencies in feature computation between training & inference?

- Feature store is a Centralized repository for sharing and accessing features.
- Advantages
  - Streamlines feature engineering processes.
  - Promotes collaboration and knowledge sharing.
  - Improves model accuracy and reduces errors.

Why Use Databricks Feature Store?

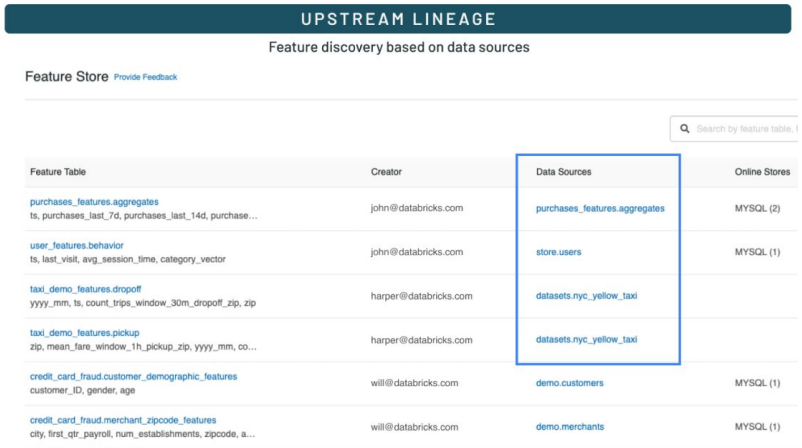
Empower Data Teams with Feature Management



- Provide data teams with the ability to
  - create new features,
  - explore and reuse existing ones,
  - publish features to low-latency online stores,
  - build training data sets and
  - retrieve feature values for batch inference.

Why Use Databricks Feature Store?

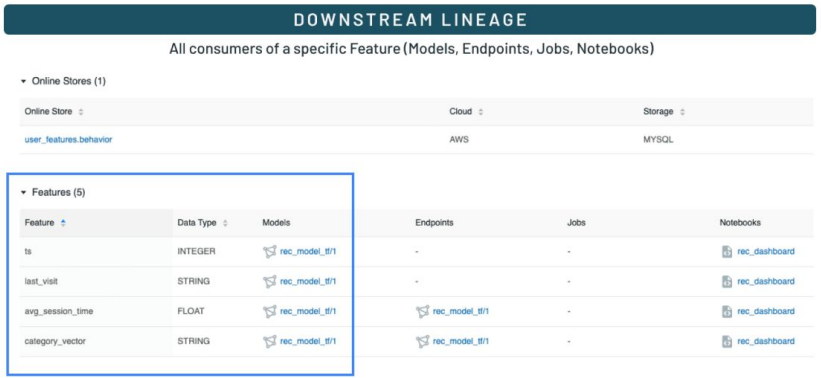
Features as Reusable Assets



- Feature Registry provides a searchable record of all features, their associated definition, source data, and their consumers
- Search for features based on the consumed raw data, use features directly or fork existing features.

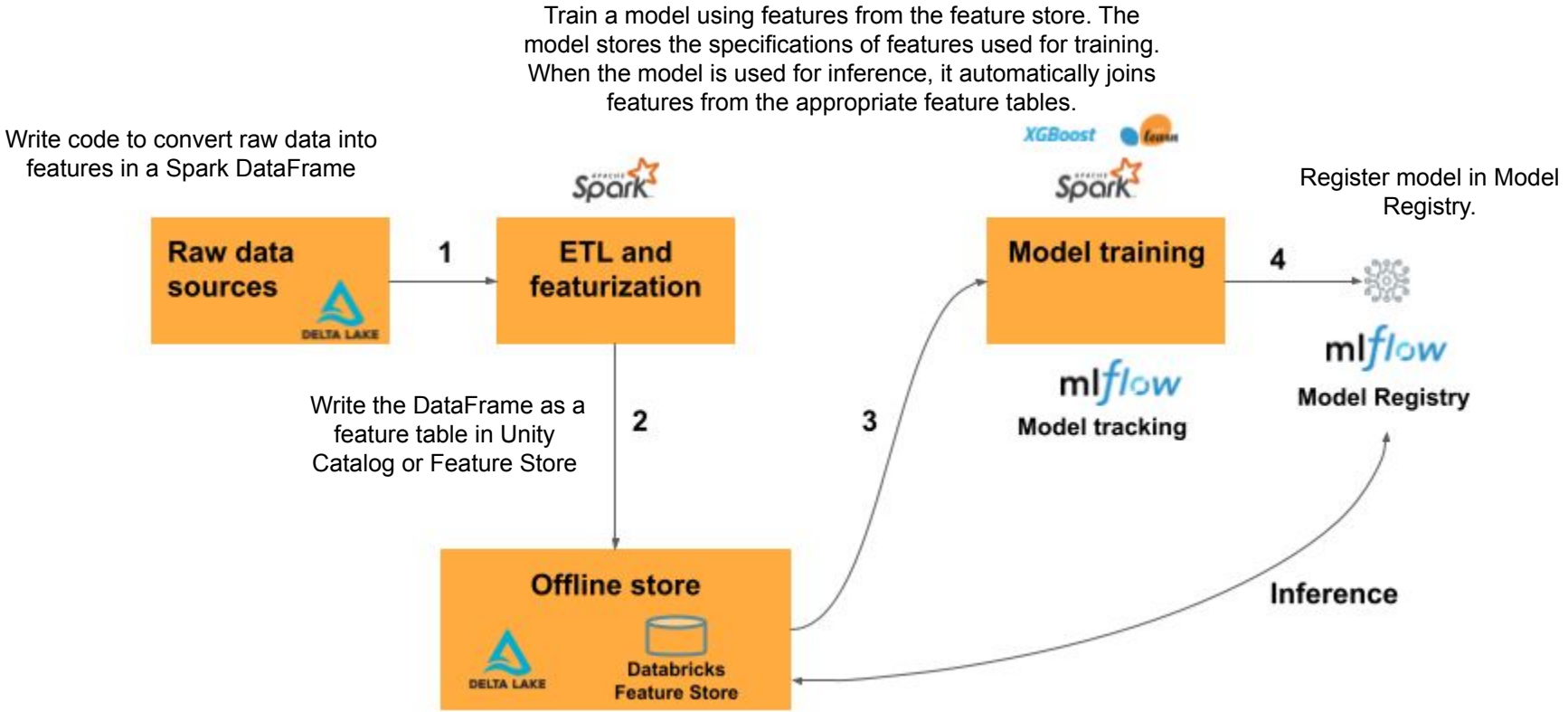
Why Use Databricks Feature Store?

Consistent Features for Training and Serving



- Feature Provider serves the features in two modes:
  - Batch mode provides features at high throughput for training ML models or batch inference.
  - Online mode provides features at low latency for serving ML models or for the consumption of the same features in BI applications.
- Features used in model training are automatically tracked with the model

How Does Databricks Feature Store Work?

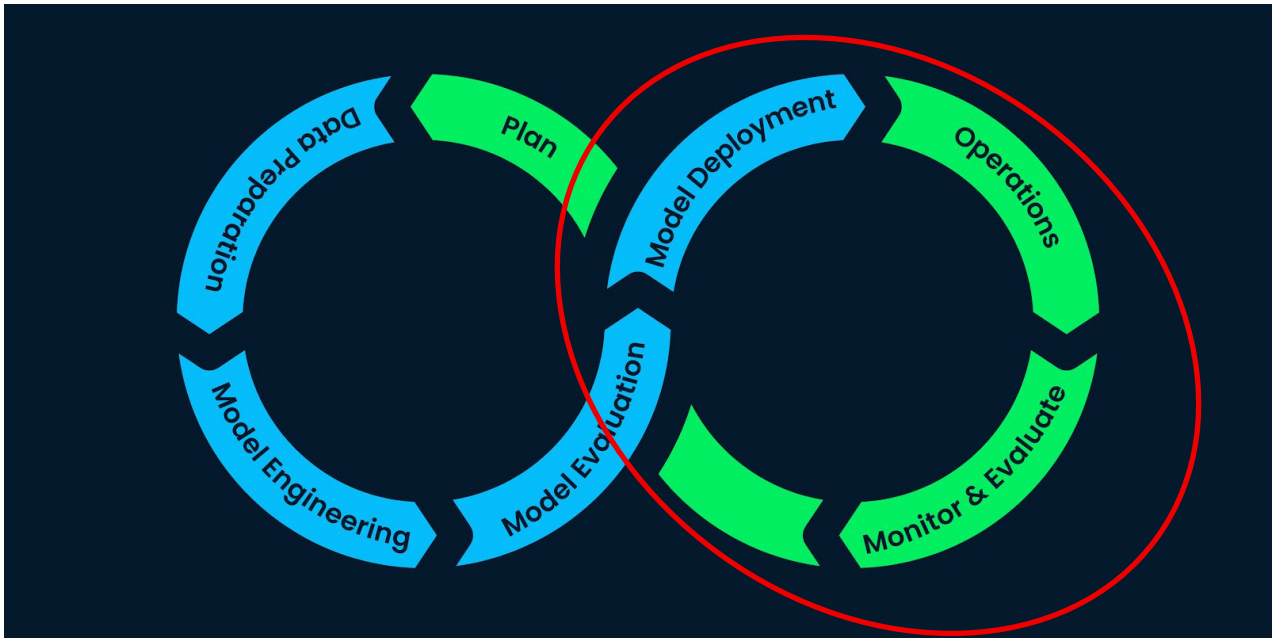


Lab/Demo: Feature Engineering in Unity Catalog

Lab/Demo: Basic Workspace Feature Store

Managed ML flow in Databricks

Model Lifecycle Management



Concerns When Deploying Models

Availability

- How will my end users or application use the model
- Where do I need to put my model for making it accessible
- Will the model be easy to use and understand

Evaluation

- Are my users using the model?
- Is my model still performing?
- Do I need to retrain my model?
- Do I need a new model that is better?

MLflow

TRACKING

Record and query experiments: code, data, config, and results.

PROJECTS

Package data science code in a format that enables reproducible runs on many platforms

MODEL REGISTRY

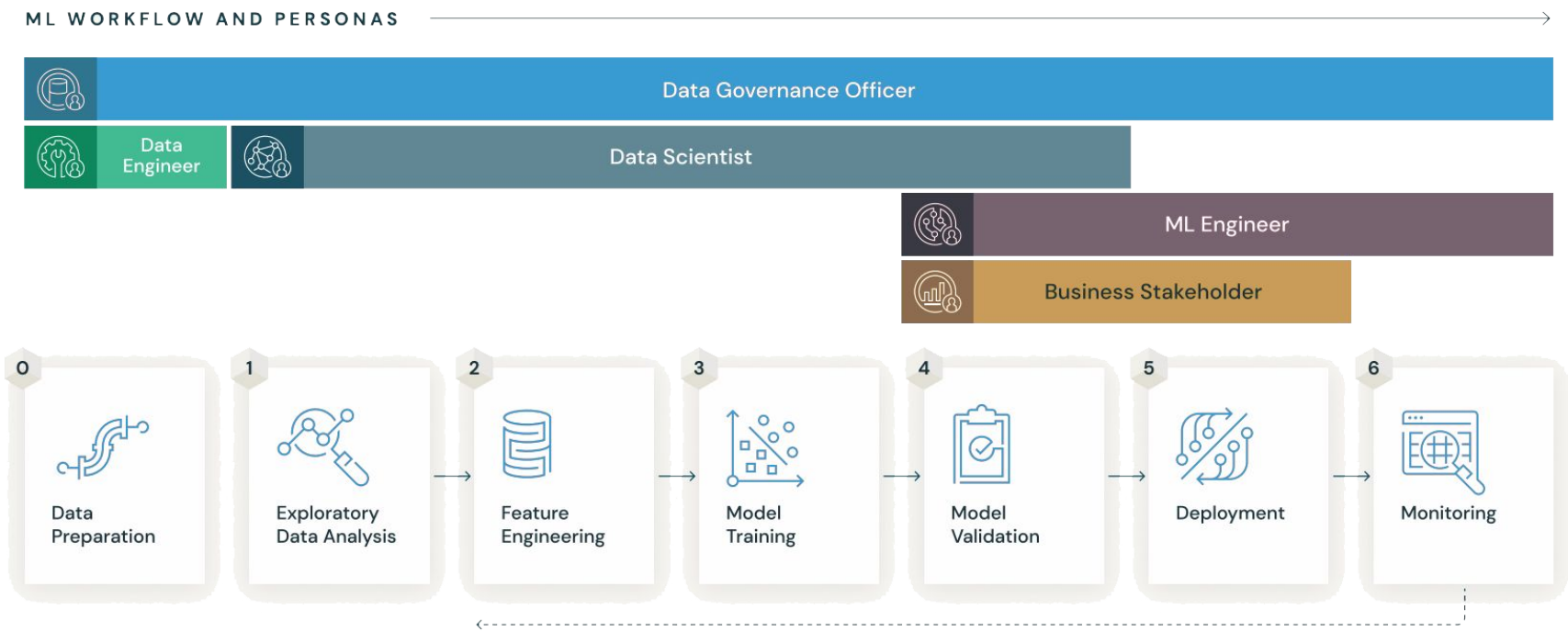
Store, annotate, and manage models in a central repository

MODELS

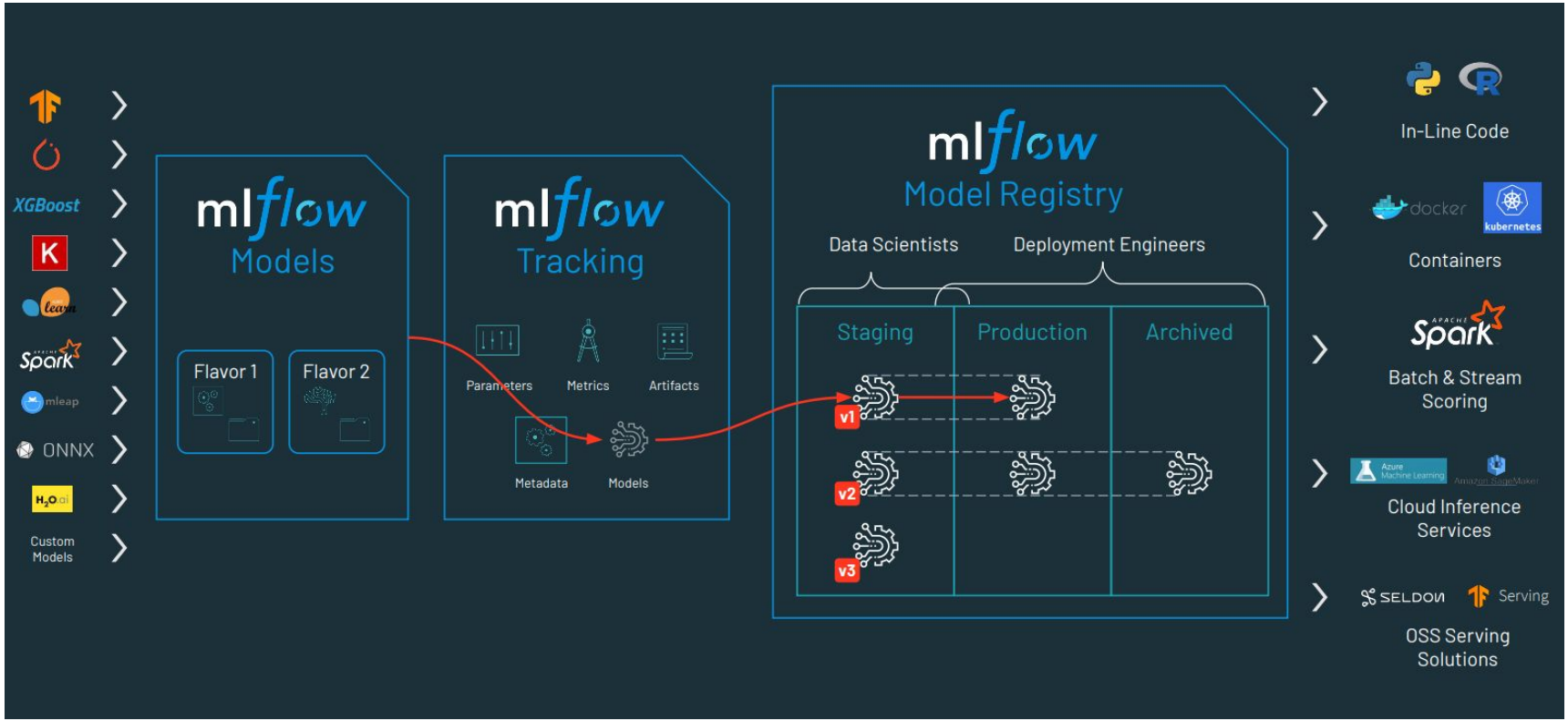
Deploy machine learning models in diverse serving environments

- Open-source platform for machine learning lifecycle
- Operationalizing machine learning
- Developed by Databricks
- Pre-installed on the Databricks Runtime for ML
- APIs: CLI, Python, R, Java, REST

MLflow


ML WORKFLOW AND PERSONAS
Data Governance Officer
Data Engineer
Data Scientist
ML Engineer
Business Stakeholder
0
Data Preparation
1
Exploratory Data Analysis
2
Feature Engineering
3
Model Training
4
Model Validation
5
Deployment
6
Monitoring

Model Lifecycle Management



Lab/Demo: Explore ML End-to-End Example

Q&A

Lesson Review

Q. Which MLflow component is used for Storing, Annotating, and managing models in a central repository?

A. Model Registry

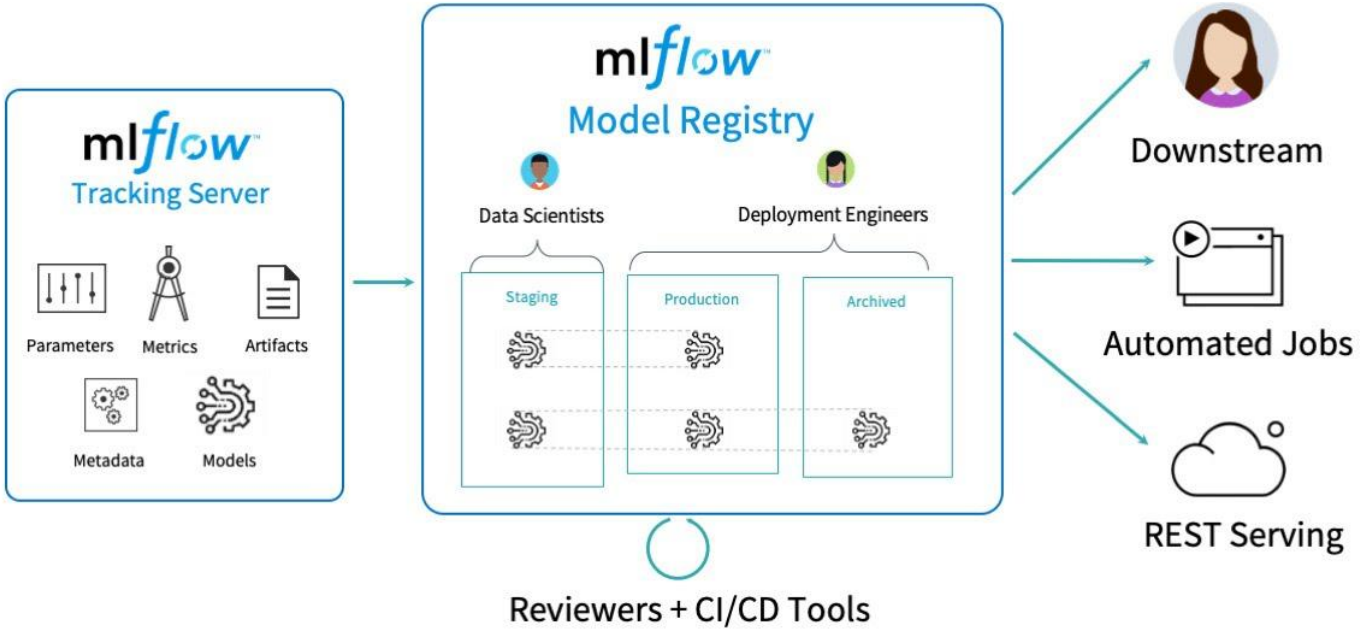
Q. Which MLflow component is used for deploying ML models in diverse serving environments?

A. Models

Break (5 min)

MLflow Model Registry

MLflow Model Registry



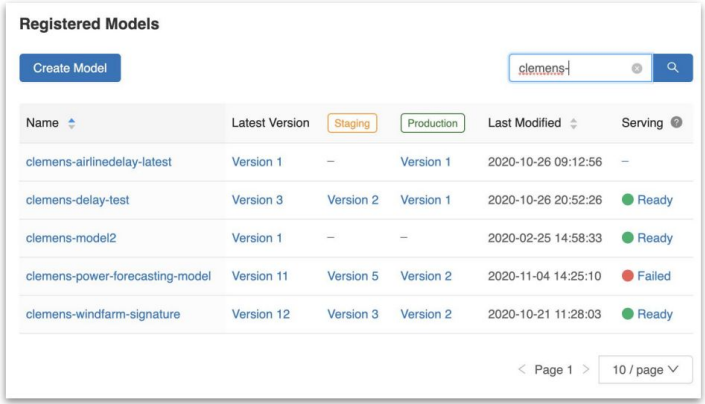
MLflow Model Registry

- Collaborative, centralized model hub
- Facilitate experimentation, testing, and production
- Integrate with approval and governance workflows
- Monitor ML deployments and their performance

Introducing the MLflow Model Registry Blog Post

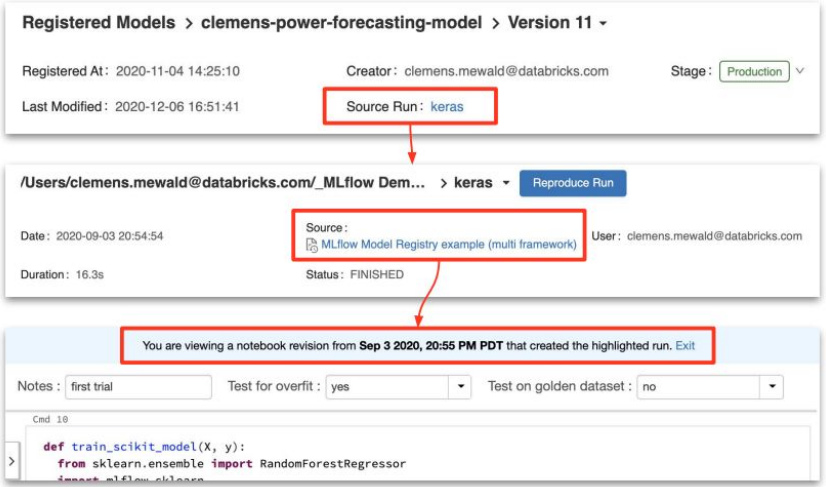
MLflow Model Registry

One Collaborative Hub for Model Management



Centralized Model Management and Discovery

- Overview of all registered models, their versions at Staging and Production
- Search by name, tags, etc.
- Model-based ACLs

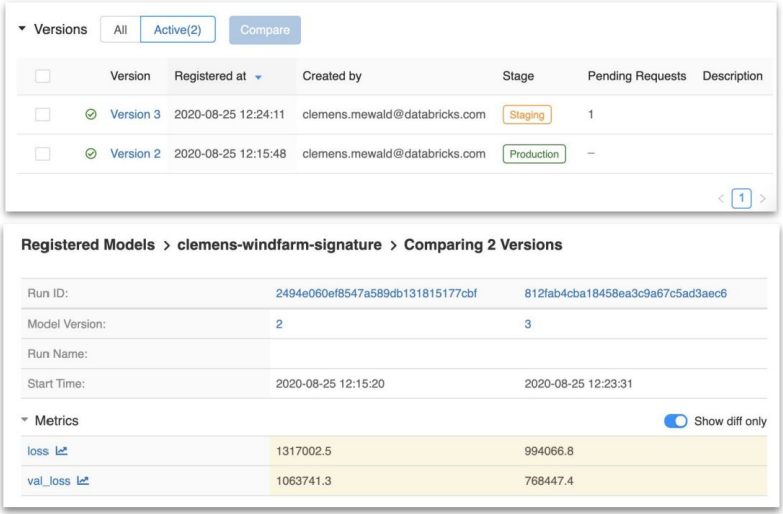


Full lineage from deployed models to

- Run that produced the model
- Notebook that produced the run
- Exact revision history of the notebook that produced the run

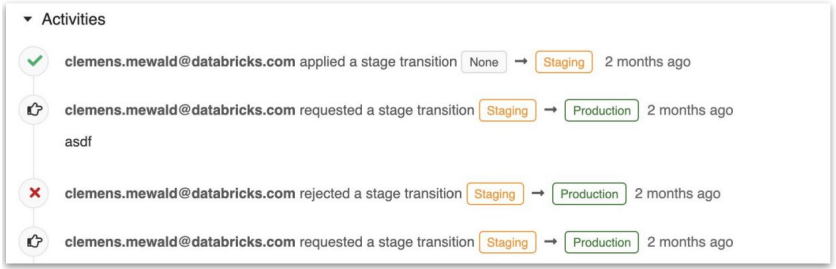
MLflow Model Registry

Version Control and Visibility into Deployment Process



Versioning of ML artifacts

- Overview of active model versions & deployment stage
- Comparison of versions & their logged metrics, parameters, etc.

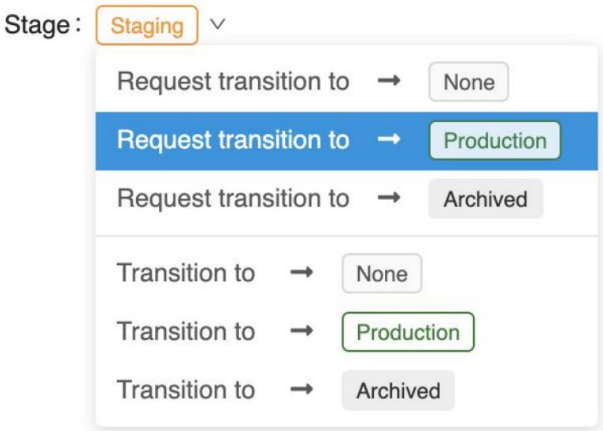


Visibility and auditability of the deployment process

- Audit log of stage transitions and requests per model

MLflow Model Registry

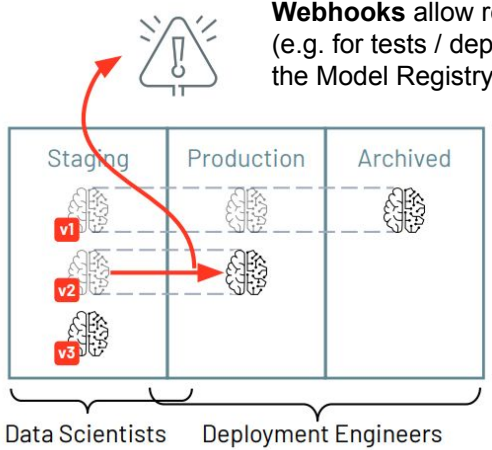
Review Processes and CI/CD Integration



Manual review process

- Stage-based Access Controls
- Request and approval workflow for stage transitions

Webhooks allow registering of callbacks (e.g. for tests / deployment) on events in the Model Registry



Automation through CI/CD integration

- Webhooks for events like model creation, version creation, transition request, etc.
- Mechanisms to store results / metadata through Tags and Comments

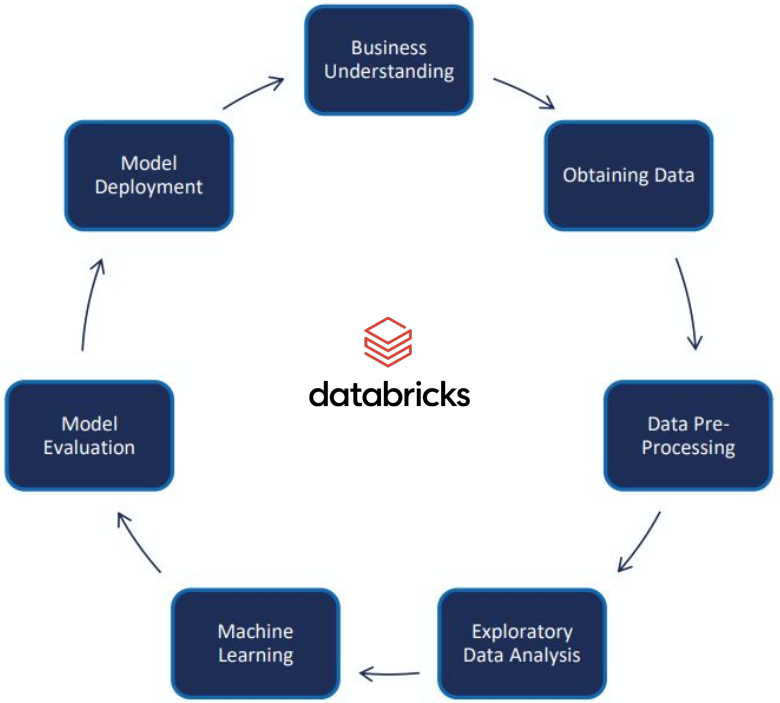
Lab/Demo: Manage Model Lifecycle Using the Workspace Model Registry

End of Day 2

Day 3
ML Workflows

Today's Schedule & Learning Objectives

- Hyperparameter tuning
- Evaluation and selection
- Binary Classification, Regression & Decision Trees
- Spark ML modeling APIs



Hyperparameter Tuning

Hyperparameter Basics

What Is a Hyperparameter?

- Examples for Random Forest:
  - Tree depth
  - Number of trees
  - Number of features to consider

A parameter whose value is used to control the training process.

Hyperparameter Basics

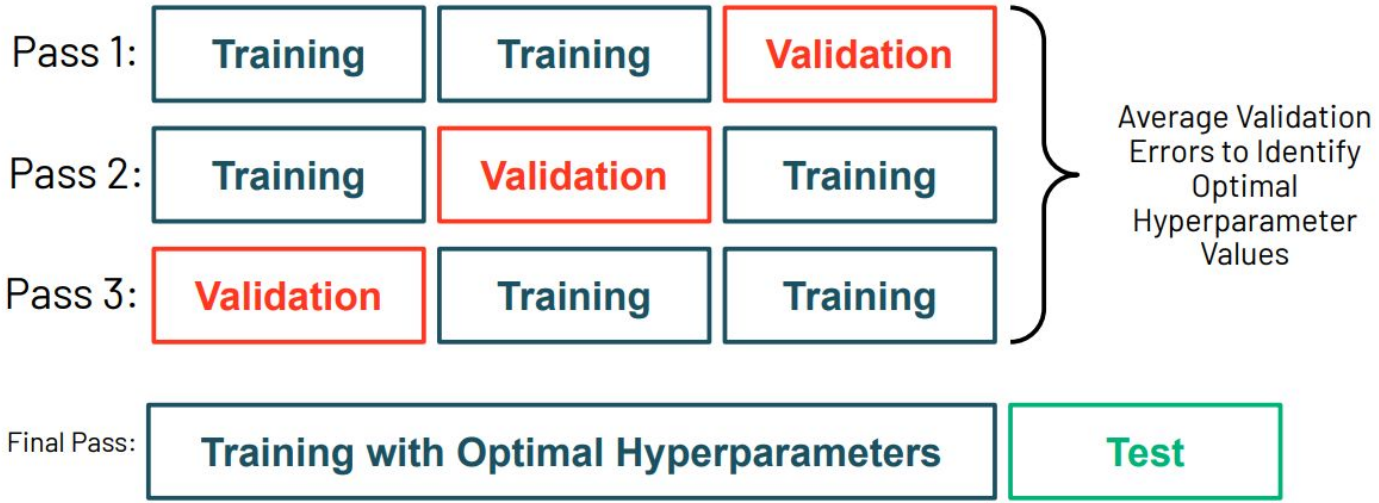
Selecting Hyperparameter Values

- Build a model for each hyperparameter value
- Evaluate each model to identify the optimal hyperparameter value
- What dataset should we use to train and evaluate?

What if there isn't enough data to split into three separate sets?

Hyperparameter Basics

K-Fold Cross Validation



Hyperparameter Basics

Optimizing Hyperparameter Values

- Train and validate every unique combination of hyperparameters

Tree Depth	Number of Trees
5	2
8	4

Tree Depth	Number of Trees
5	2
5	4
8	2
8	4

Question: With 3-fold cross validation, how many models will this build?

Hyperparameter Tuning with Hyperopt

Hyperparameter Tuning with Hyperopt

Problems with Grid Search

- Exhaustive enumeration is expensive
- Manually determined search space
- Past information on good hyperparameters isn't used
- So, what do you do if…
  - You have a training budget
  - You have a non-parametric search space
  - You want to pick your hyperparameters based on past results

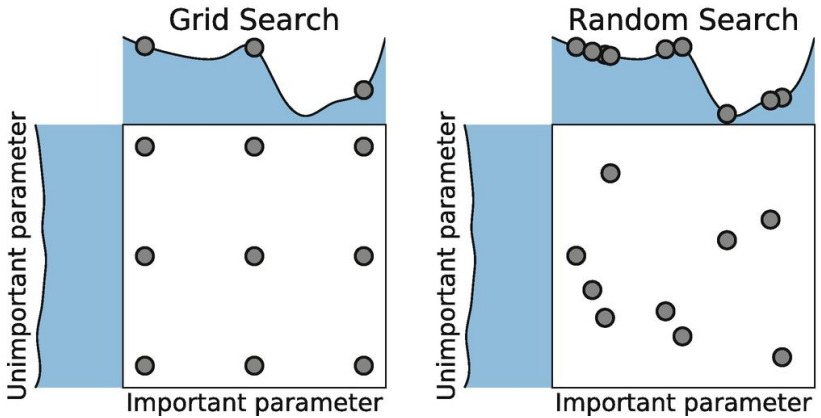
Hyperparameter Tuning with Hyperopt

What Is Hyperopt?

- Open-source Python library
- Optimization over awkward search spaces
- Serial
- Parallel
- Spark integration
- Three core algorithms for optimization:
  - Random Search
  - Tree of Parzen Estimators (TPE)
  - Adaptive TPE

Hyperparameter Tuning with Hyperopt

Optimizing Hyperparameter Values



- Generally, outperforms grid search
- Can struggle on some datasets (e.g. convex spaces)

Hyperparameter Tuning with Hyperopt

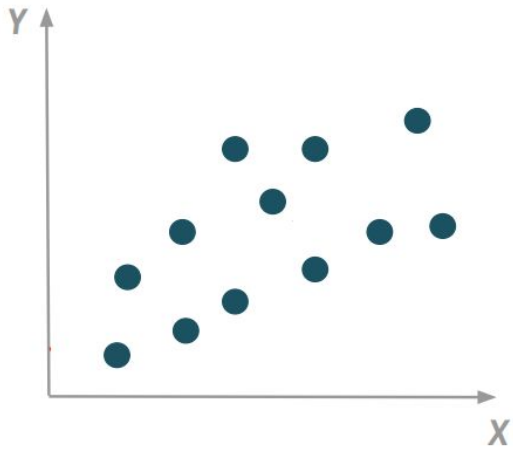
Tree of Parzen Estimators

- Meta-learner, Bayesian process
- Non-parametric densities
- Returns candidate hyperparameters based on best expected improvement
- Provide a range and distribution for continuous and discrete values
- Adaptive TPE better tunes the search space
  - Freezes hyperparameters
  - Tunes number of random trials before TPE

Break (5 min)

Classification, Regression, and Decision Trees

Linear Regression

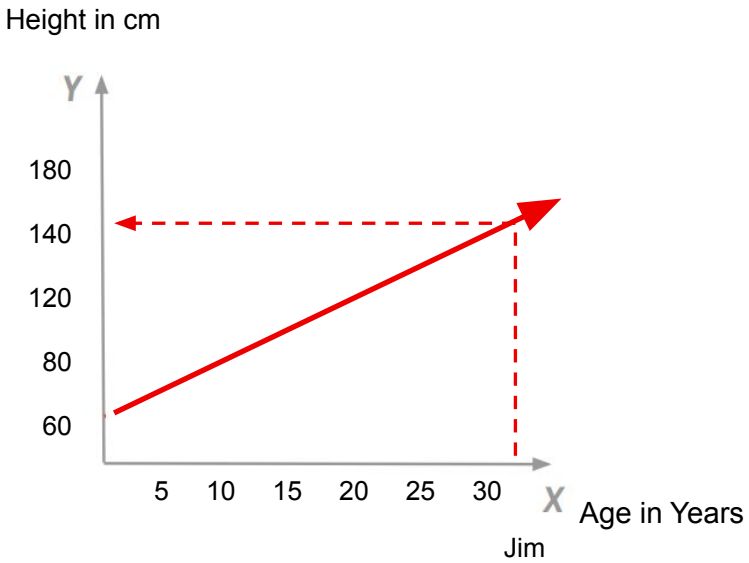


- A supervised machine learning method
- Provides a linear relationship between an independent variable and a dependent variable to predict the outcome of future events

Using **Databricks AutoML** our goal will be to train a **Linear Regression Model** - draw a line that minimizes the sum of the squared residuals.

Linear Regression

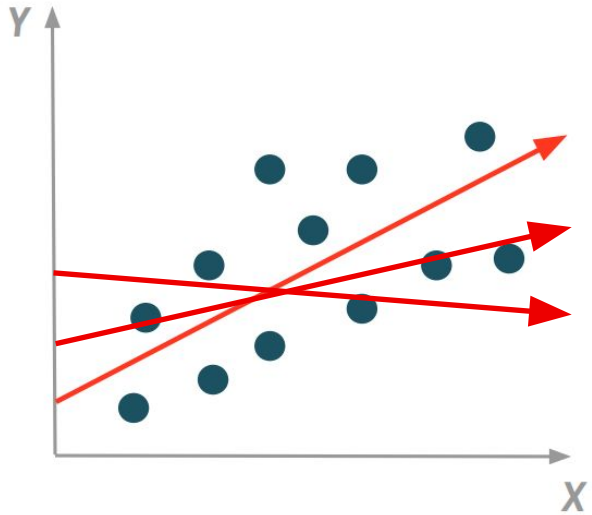
A Quick Example



Name	Age	Height
Alfred	5	60
Tom	12	100
Zack	20	140
Cathy	7	180
Ravi	30	160
Johnathan	25	140
Jim	32	??

Linear Regression

Goal : Find the Line of Best Fit



$$\hat{y} = w_0 + w_1 x$$

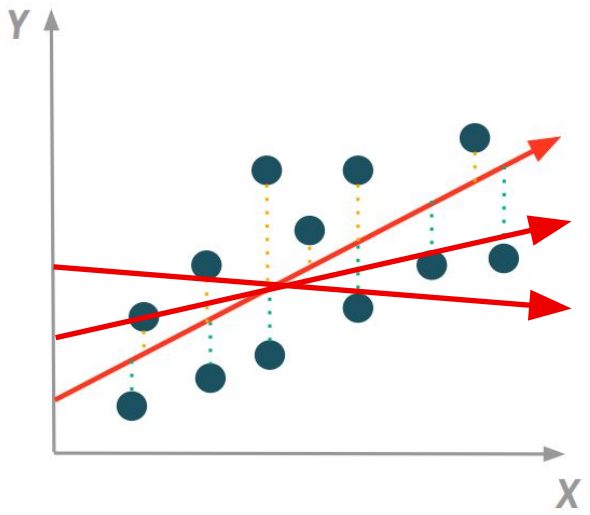
$$y \approx \hat{y} + \epsilon$$

where...

x: feature
y: label
w_0: y-intercept
w_1: slope of the line of best fit

Linear Regression

Goal : Find the Line of Best Fit While Minimizing the Residuals



- **Blue Point:** True Value
- **Green-Dotted Line**: Positive Residual
- **Orange-Dotted Line**: Negative Residual
- **Red Line**: Line of best fit

Evaluation Metrics

Loss: $(y - \hat{y})$
Absolute loss: $|y - \hat{y}|$
Squared loss: $(y - \hat{y})^2$

The goal is to draw a line that minimizes the sum of the squared residuals.

Linear Regression

Evaluation Metric: Root Mean-Squared-Error (RMSE) & R^2

$$Error = (y_i - \hat{y}_i)$$

$$SE = (y_i - \hat{y}_i)^2$$

$$SSE = \sum_{i=1}^{n} (y_i - \hat{y}_i)^2$$

$$MSE = \frac{1}{n} \sum_{i=1}^{n} (y_i - \hat{y}_i)^2$$

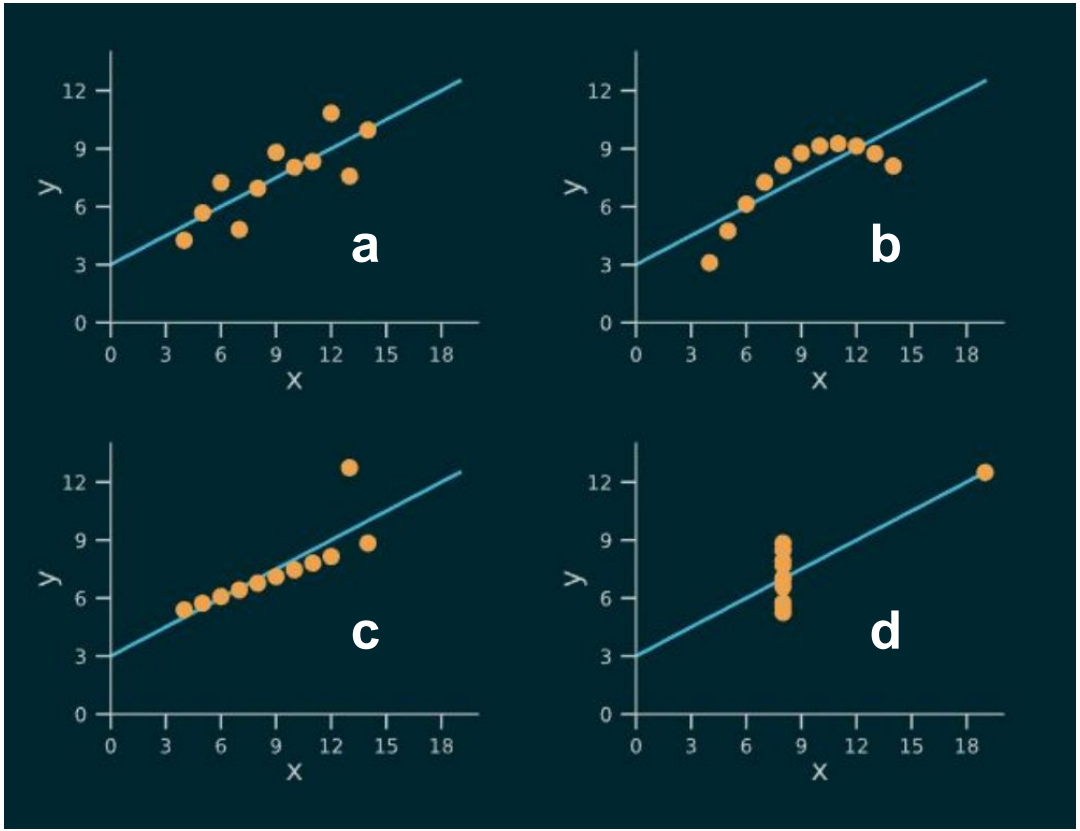
$$RMSE = \sqrt{\frac{1}{n} \sum_{i=1}^{n} (y_i - \hat{y}_i)^2}$$

$$SS_{tot} = \sum_{i=1}^{n} (y_i - \bar{y})^2$$

$$SS_{res} = \sum_{i=1}^{n} (y_i - \hat{y}_i)^2$$

$$R^2 = 1 - \frac{SS_{res}}{SS_{tot}}$$

Which Dataset Is Best Suited for Linear Regression?



Machine Learning Libraries

Scikit-learn is a popular single-node machine learning library

But what if our data or model gets too big for a single-node?

Machine Learning In Spark

Machine learning in Spark allows us to work with **bigger data** and train models **faster** by **distributing the data** and **computations** across **multiple workers**

Scale Out and **Speed Up**

MLLib

- Original ML API for Spark
- Based on RDDs

Spark ML

- Newer ML API for Spark
- Based on DataFrames
- Supported API

Non-Numeric Features

Two primary types of non-numeric features

Ordinal Features

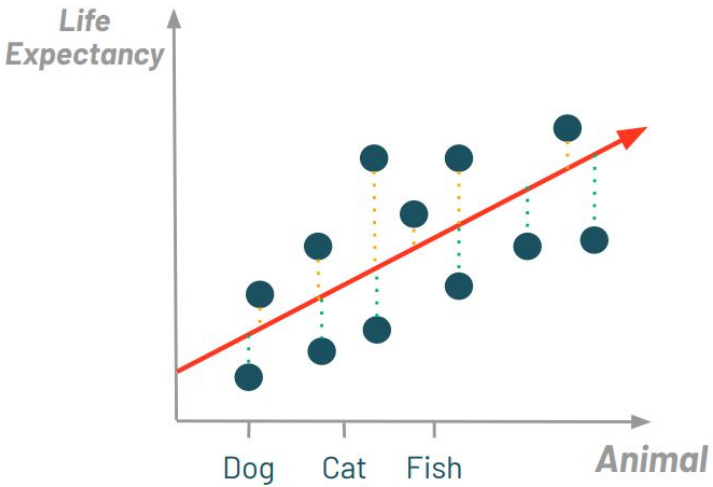
- A series of categories of a single feature
- Relative ordering, but not necessarily consistent spacing e.g. Infant, Toddler, Adolescent, Teen, Young Adult, etc.

Categorical Features

- A series of categories of a single feature
- No intrinsic ordering e.g. Dog, Cat, Fish

Non-Numeric Features

The Case of Linear Regression



How do we handle non-numeric features for linear regression?

- X-axis is numeric, so features need to be numeric
- Convert our non-numeric features to numeric features?

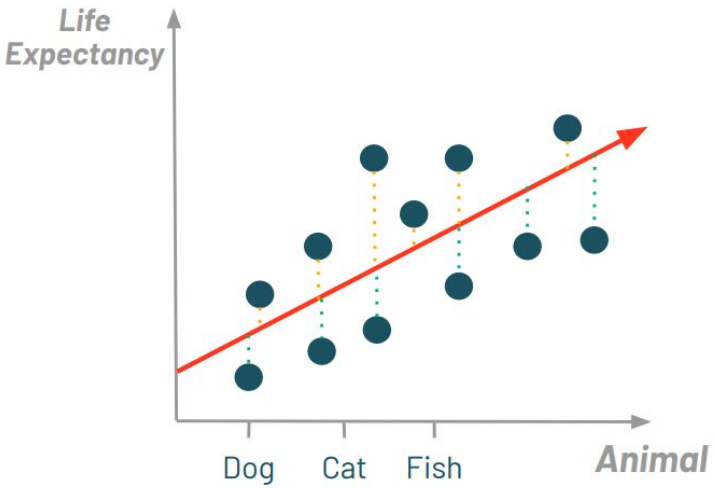
Could we assign numeric values to each of the categories?

- “Dog” = 1, “Cat” = 2, “Fish” = 3, etc.
- Does this make sense?

Non-Numeric Features

The Case of Linear Regression

This implies 1 Cat is equal to 2 Dogs!



How do we handle non-numeric features for linear regression?

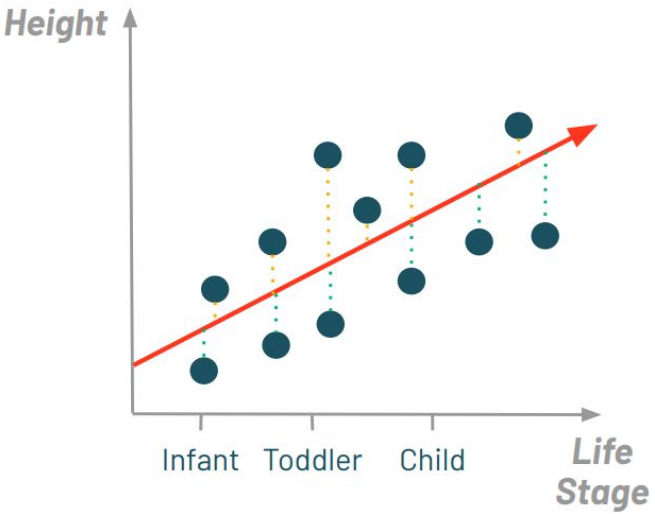
- X-axis is numeric, so features need to be numeric
- Convert our non-numeric features to numeric features?

Could we assign numeric values to each of the categories?

- “Dog” = 1, “Cat” = 2, “Fish” = 3, etc.
- Does this make sense?

Non-Numeric Features

The Case of Linear Regression



What about with ordinal variables?

- Since ordinal variables have an order just like numbers, could this work?
- “Infant” = 1, “Toddler” = 2, “Child” = 3, etc.
- Does this make sense?

Ordinal categories aren’t necessarily evenly spaced, so it’s still not perfect and not particularly scalable

Non-Numeric Features

Use of One-Hot Encoding (OHE)

Animal
Dog
Cat
Fish

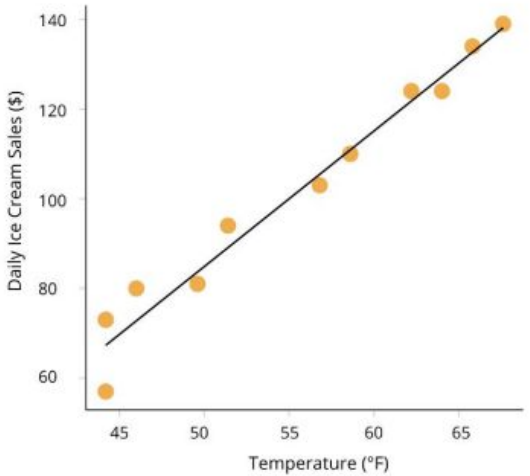
OHE →

Dog	Cat	Fish
1	0	0
0	1	0
0	0	1

- Creates a binary “dummy” feature for each category
- Doesn’t force a uniformly-spaced, ordered numeric representation

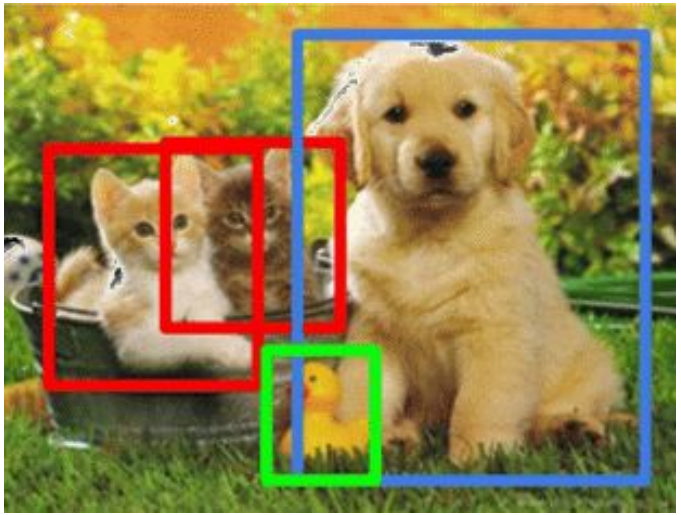
Types of Supervised Learning

Regression



Predicting a continuous output

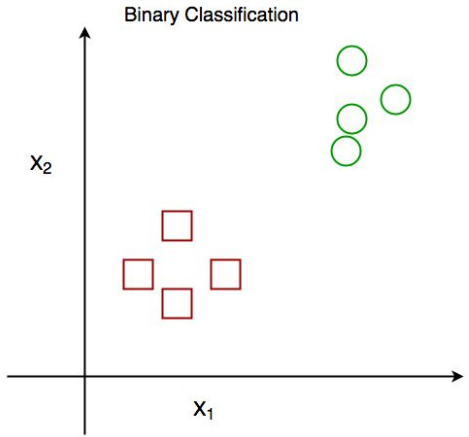
Classification



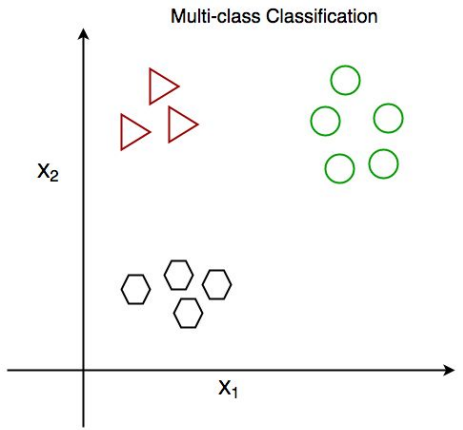
Predicting a categorical/discrete output

Types of Classification

Binary Classification (2 label class)



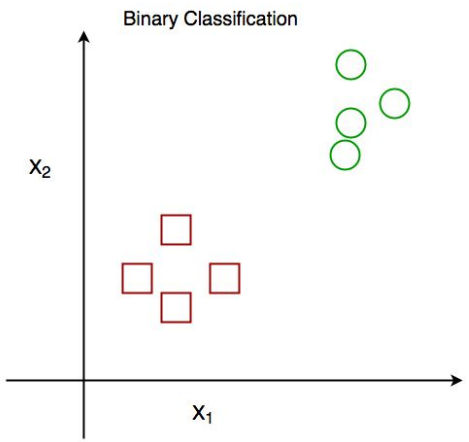
Multiclass Classification (>3 label class)



Model output is commonly the probability of a record belonging to each of the classes

Binary Classification

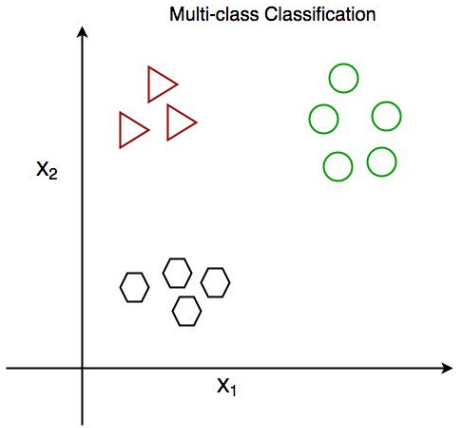
Binary Classification (2 label class)



- Outputs:
  - Probability that the record is **Red Square** given a set of features
  - Probability that the record is Green Circle given a set of features
- Reminders:
  - Probabilities are bounded between 0 and 1
  - And linear regression returns any real number

Multiclass Classification

Multiclass Classification (>3 label class)

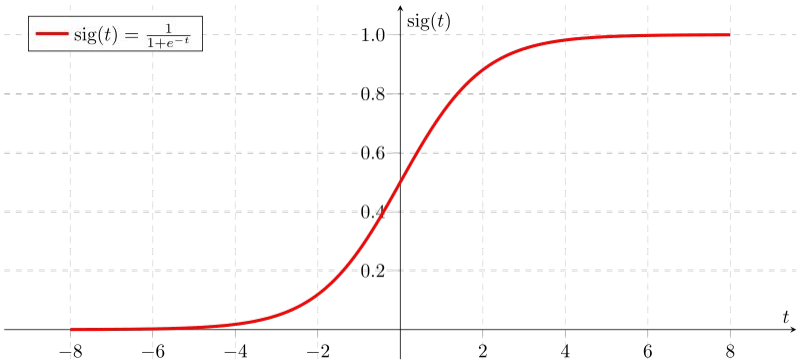


- Creates a binary “dummy” feature for each category
- Doesn’t force a uniformly-spaced, ordered numeric representation

Bounding Binary Classification Probabilities

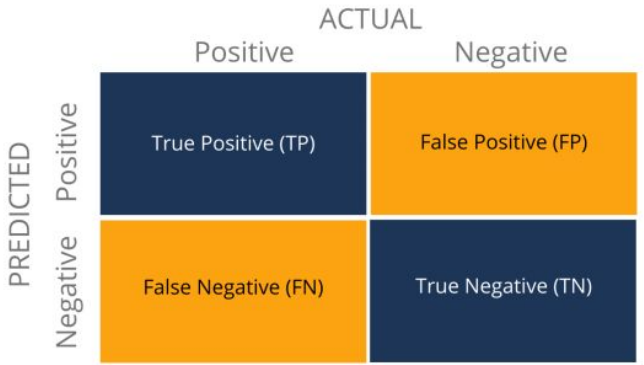
How can we keep model outputs between 0 and 1?

- Logistic Function:
  - Large positive inputs → 1
  - Large negative inputs → 0

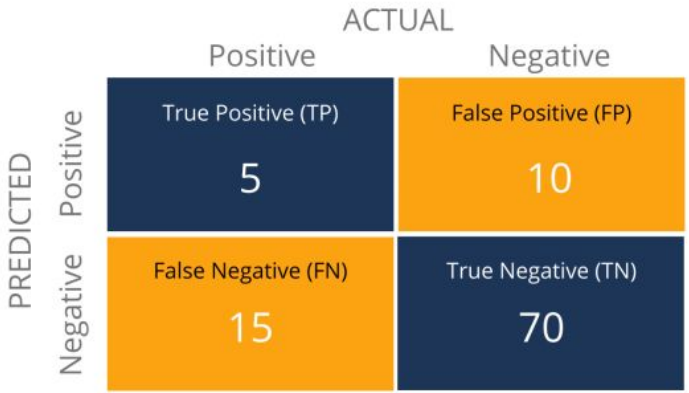


Evaluating Binary Classification Models

- How can the model be wrong?
  - Type I Error: False Positive
  - Type II Error: False Negative
- Representing these errors with a confusion matrix



Binary Classification Metrics



$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN} = 0.75$$

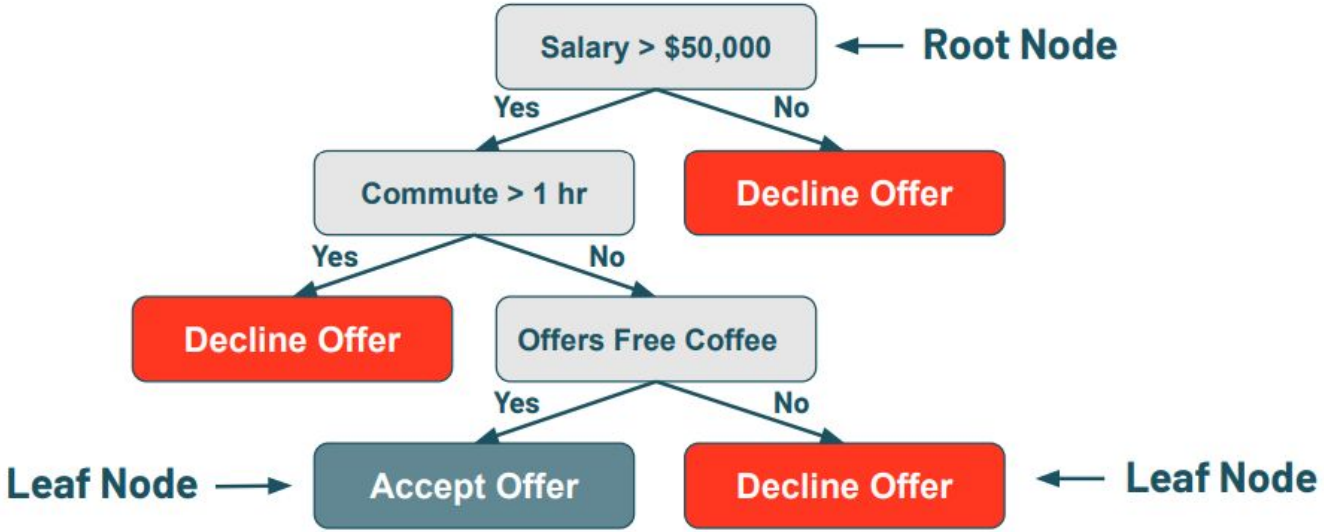
$$\text{Recall} = \frac{TP}{TP + FN} = 0.25$$

$$\text{Precision} = \frac{TP}{TP + FP} = 0.33$$

$$\text{F1 Score} = \frac{2 * \text{Precision} * \text{Recall}}{\text{Precision} + \text{Recall}} = 0.28$$

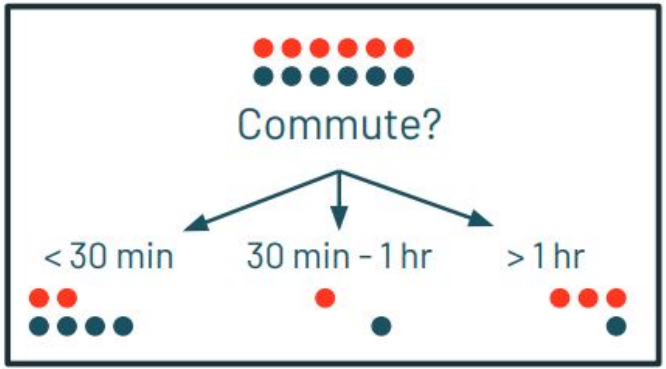
Decision Trees

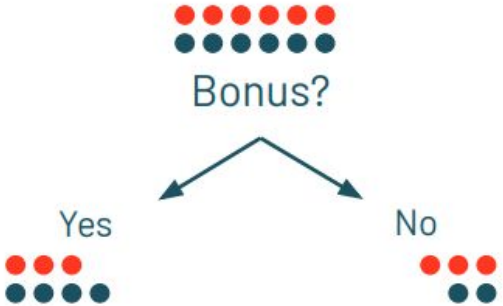
Decision Making



Decision Trees

Determining Splits

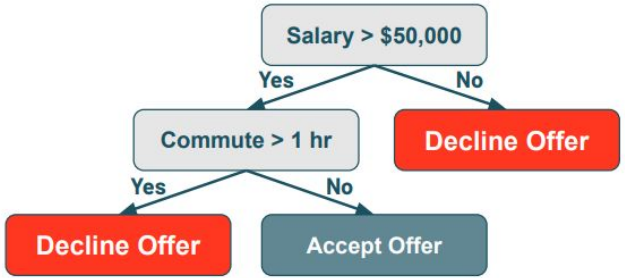


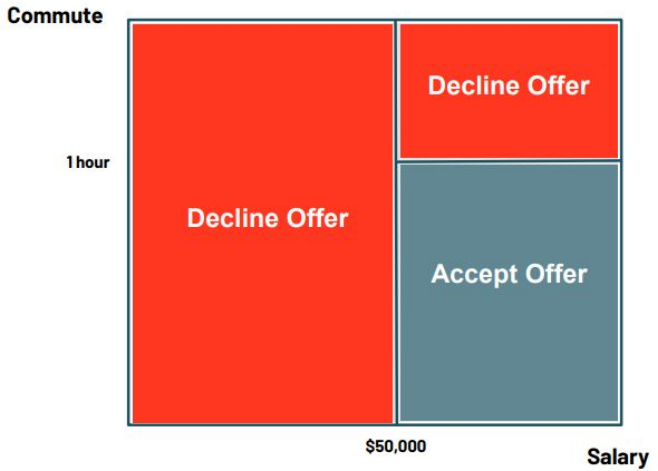


Commute is a better choice because it provides information about the classification.

Decision Trees

Creating Decision Boundaries

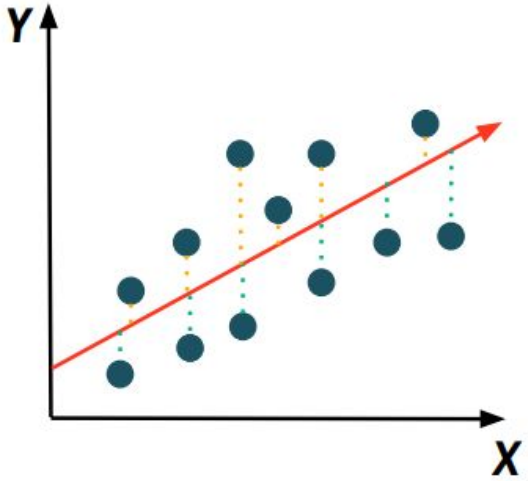


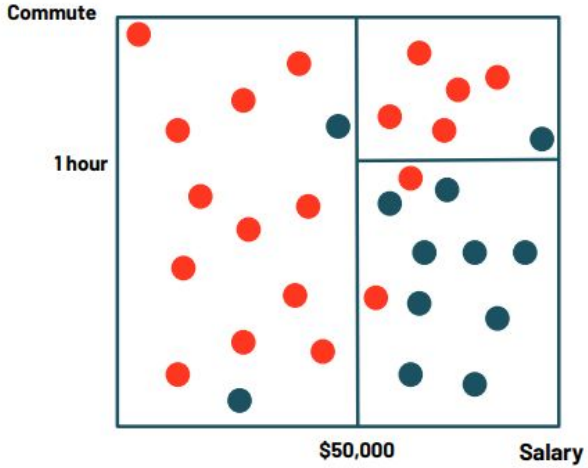


Commute is a better choice because it provides information about the classification.

Decision Trees

Lines vs. Boundaries





Linear Regression

- Lines through data
- Assumed linear relationship

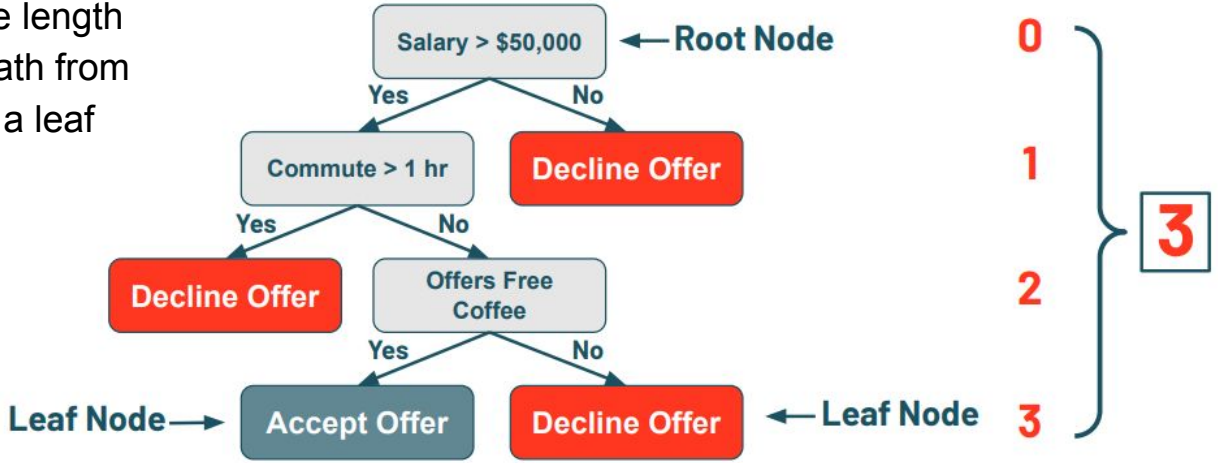
Decision Trees

- Boundaries instead of lines
- Learn complex relationships

Decision Trees

Tree Depth

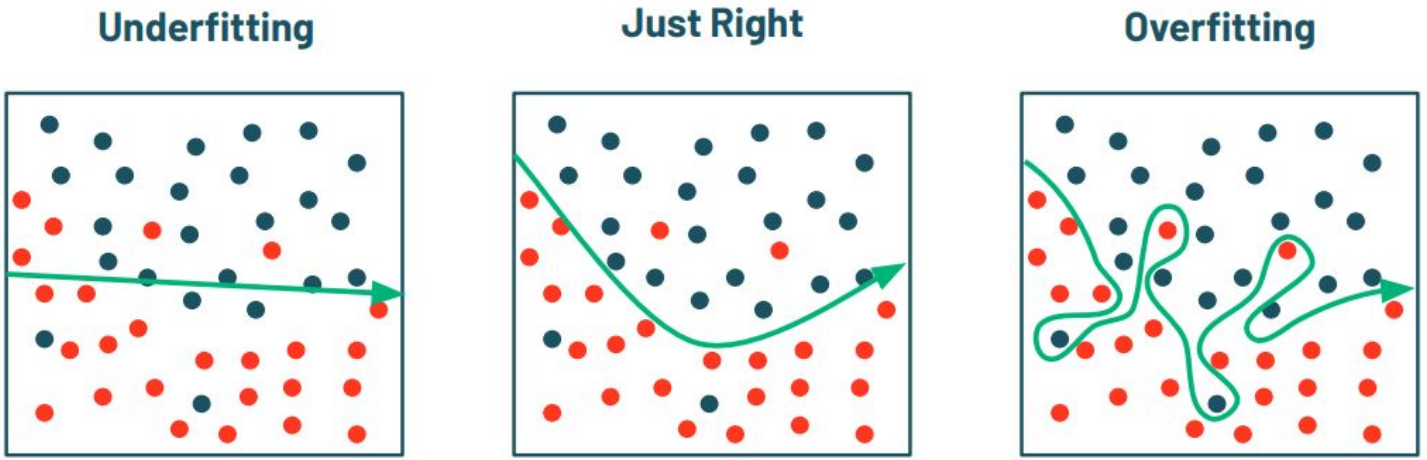
Tree Depth: the length of the longest path from a root note to a leaf node



Note: shallow trees tend to underfit, and deep trees tend to overfit

Decision Trees

Underfitting vs. Overfitting



Q&A

Lesson Review

Q. What type of machine learning task aims to predict categories or classes?

A. Classification

Q. Which machine learning task involves predicting continuous values?

A. Regression

Q. In decision trees, what is the name of the process where the tree splits the data into smaller subsets based on feature values?

A. Splitting

Break (5 min)

Spark ML Modeling APIs

Lab/Demo: Binary Classification Using Spark ML Modelling APIs

Lab/Demo: Regression Using Spark ML Modelling APIs

Lab/Demo: Decision Trees Using Spark ML Modelling APIs

End of Day 3

Day 4
ML Scalability and Using Pandas

Today’s Schedule & Learning Objectives

- Scaling ML models
- Pandas on Databricks
- Pandas API on Spark
- Pandas function APIs
- Pandas User Defined Functions (UDFs)

Scaling ML Models

Ensemble Learing

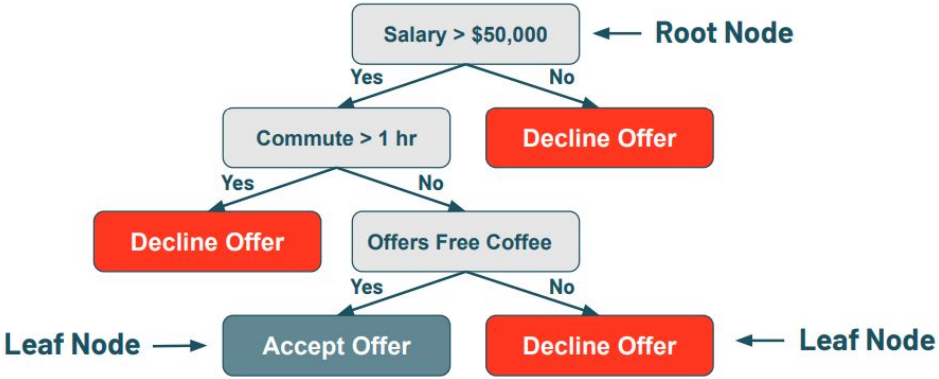
- Ensemble methods use multiple learning algorithms to obtain better predictive performance than could be obtained from any of the constituent learning algorithms alone.
- 3 types of Ensemble Learning
  - Bagging
  - Boosting
  - Stacking

Ensemble Learning

The Decision Making Dilemma

But Decision Trees suffer from bias and variance issues

- To make model more flexible (low bias), increase the number of tunable parameters (depth, split, features)
- But more flexibility = high variance, i.e., small changes in the training data can result in large changes in the model's predictions.
- Reducing Variance will increase Bias = model less flexible



Ensemble Learning

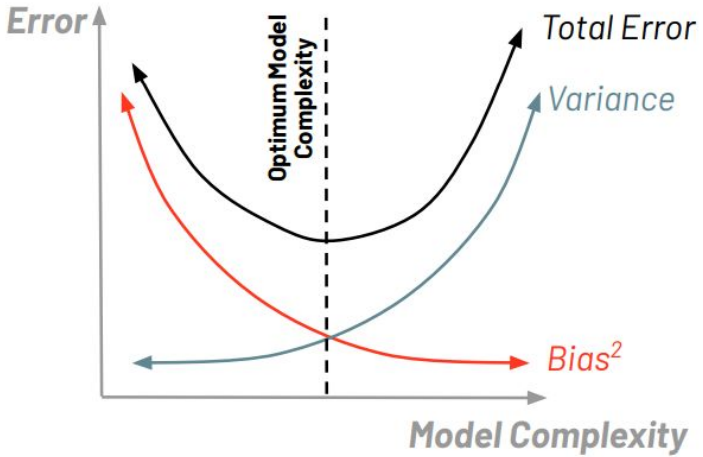
Bias-Variance Tradeoff

- Low-Bias, Low-Variance:
  - Ideal model.
  - Not possible practically.
- Low-Bias, High-Variance:
  - Overfitting. Predictions inconsistent and accurate on average.
  - Predicted values accurate(average) but scattered.
- High-Bias, Low-Variance:
  - Underfitting. Predictions consistent but inaccurate on average.
  - Predicted values inaccurate but will be not scattered.
- High-Bias, High-Variance:
  - Predictions inconsistent and also inaccurate on average.

Ensemble Learning

Bias-Variance Tradeoff

$$Error = Variance + Bias^2 + Noise$$



How can you lower BOTH Bias and Variance at the same time?

- Reduce Bias : Build more complex models
- Reduce Variance : Use a lot of data

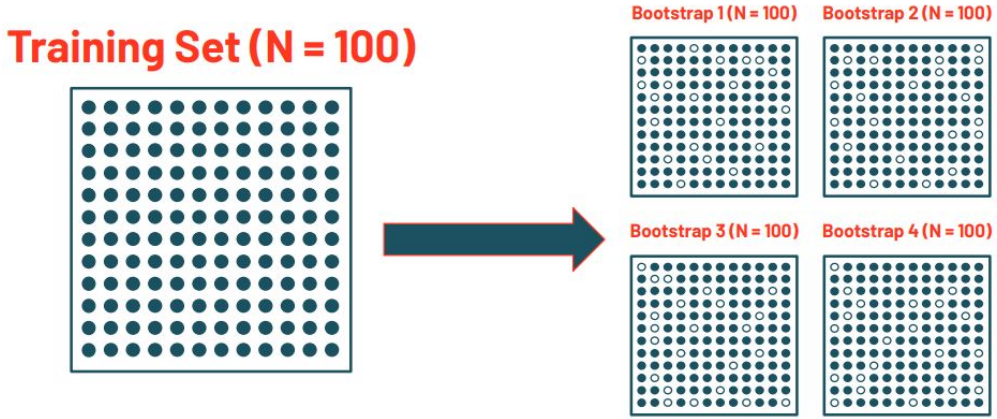
Solution

- Let us build 100 complex decision trees
  - Using more data reduces variance for one model
  - Averaging more predictions reduces prediction variance
- But we only have one training set … or do we?

Ensemble Learning

Bootstrap Sampling

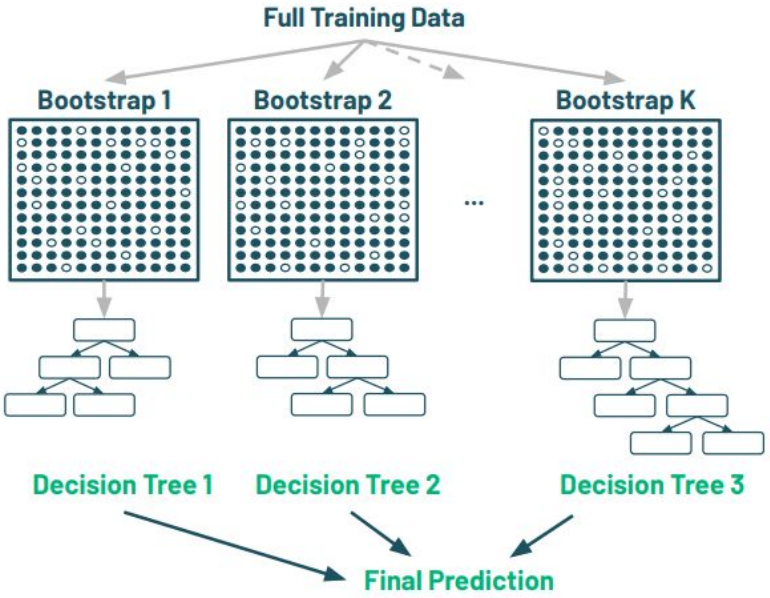
- A method for simulating N new datasets:
  1. Take sample with replacement from original training set
  2. Repeat N times



Ensemble Learning

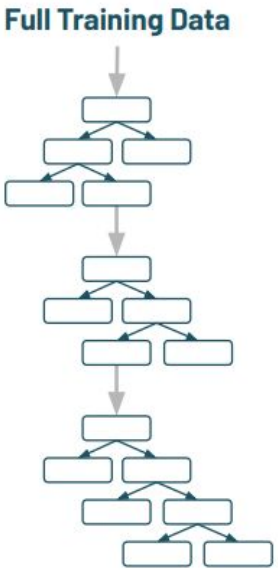
Bagging (Bootstrap Aggregating)

- Train a tree on each of sample, and average the predictions
- This is bootstrap aggregating, commonly referred to as bagging



Ensemble Learning

Boosting



- Sequential (one tree at a time)
- Each tree learns from the last
- Sequence of trees is the final model

Ensemble Learning

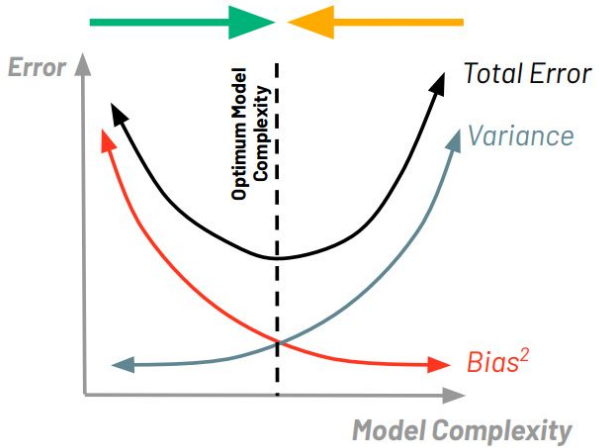
Bagging vs. Boosting

Boosting

- Starts with high bias, low variance
- Works right
- E.g., Gradient Boosted Decision Trees

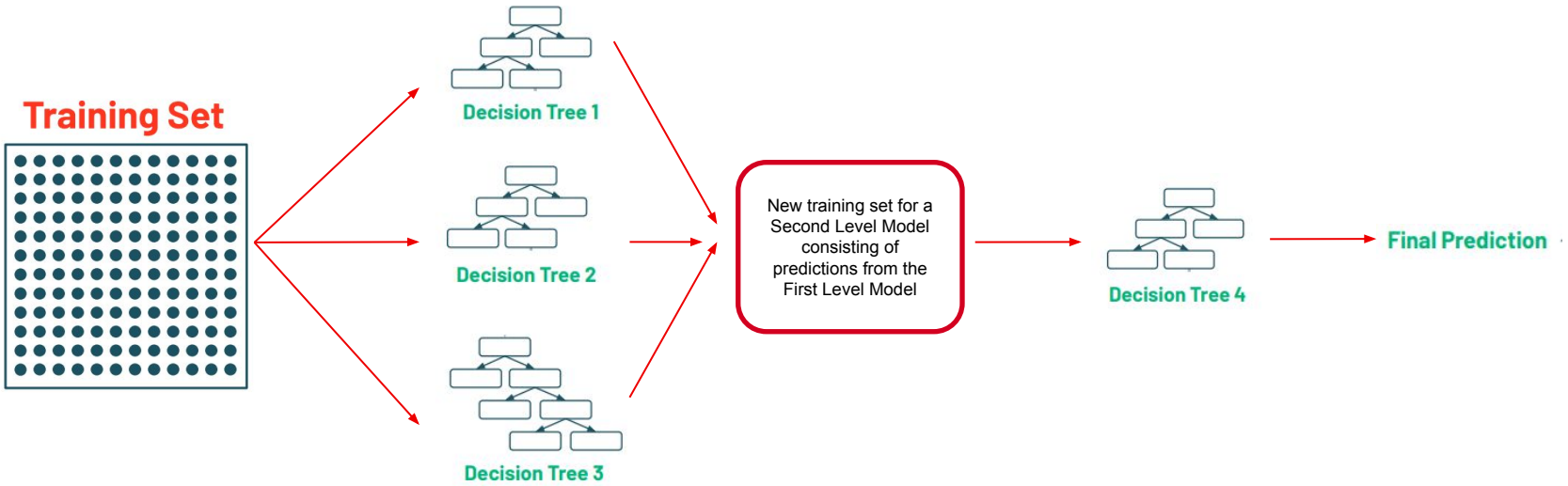
Bagging

- Starts with high variance, low bias
- Works left
- E.g., Random Forest



Ensemble Learning

Stacking



Q&A

Lesson Review

Q. What are the primary concerns when minimizing error in machine learning models?
A. Bias and Variance

Q. What method sequentially trains weak learners to improve overall model performance?
A. Boosting

Q. What ensemble method involves training multiple models independently on different subsets of the training data?
A. Bagging

Q. What ensemble technique involves training a meta-learner on the predictions of multiple base learners?
A. Stacking

Pandas on Databricks

Introduction to Pandas on Databricks

- Databricks Runtime includes pandas as one of the standard packges
  - Allows use of pandas DataFrames in Databricks notebooks and jobs
  - Pandas API on Spark on top of PySpark DataFrames (Databricks Runtime >=10.0)
  - Convert DataFrames between pandas and PySpark
- Apache Spark
  - Includes Arrow-optimized execution of Python logic in the form of panda's function APIs
  - Allows users to apply panda transformations directly to PySpark DataFrames
  - Supports pandas UDFs, which uses Arrow-optimizations for arbitrary UDFs in python

Lab/Demo: Store & Load Data with Pandas

Lab/Demo: Working with Files on Databricks

Lab/Demo: Accessing Data via Access Key & SAS Token

Lab/Demo: Mounting ADLS to DBFS

Lab/Demo: Mount Storage Container using f-strings

Lab/Demo: Multi-hop Architecture

Break (5 min)

Pandas API on Spark

Lab/Demo: Object Creation (Series, DataFrame, View Data, Data Selection)

Lab/Demo: Applying Python Function with Pandas-on-Spark Object

Lab/Demo: Grouping & Plotting Data

Lab/Demo: Type Conversion and Native Support for Pandas Object

Lab/Demo: Distributed Execution for Pandas and Using SQL in Pandas API

Lab/Demo: Conversion to/from Pyspark Dataframe & Spark Execution Plans

Lab/Demo: Caching Dataframes

Break (5 min)

Pandas Function APIs

Pandas Function APIs

- Directly apply a Python native function to a PySpark DataFrame
  - Native function should take input and output pandas instances
- Panda's function APIs are like Panda's UDFs
- Uses Apache Arrow to transfer data, pandas to work with the data
- 3 types of function APIs : Grouped Map, Map, Cogrouped Map
- Leverage the same internal logic that Pandas UDF execution uses
- Share characteristics such as PyArrow, supported SQL types and the configurations

Lab/Demo: Pandas Function API – Grouped Map

Lab/Demo: Pandas Function API – Map

Lab/Demo: Pandas Function API – Cogrouped Map

Pandas User Defined Functions (UDFs)

Lab/Demo: Pandas User Defined Functions

Lab/Demo: Series to Series UDF

Lab/Demo: Iterator of Series to Iterator of Series UDF

Lab/Demo: Iterator of Multiple Series to Iterator of Series UDF

Lab/Demo: Series to Scalar UDF

Wrap-up

Get Ready for the Exam

Exam Format

- **Type:** Proctored certification
- **Total number of questions:** 48
- **Time limit:** 90 minutes
- **Registration fee:** $200
- **Question types:** Multiple choice
- **Test aides:** None allowed
- **Languages:** English
- **Delivery method:** Online proctored
- **Prerequisites:** None, but related training highly recommended
- **Validity period:** 2 years
- **Recertification:** Recertification is required to maintain your certification status. Databricks Certifications are valid for two years from issue date.
- **Unscored content:** Exams may include unscored items to gather statistical information for future use. These items are not identified on the form and do not impact your score.

Get Ready for the Exam

Steps Involved

1. Review the Machine Learning Associate Exam Guide to understand what will be on the exam
2. Practice the labs provided during this live event
3. Refer to the Machine Learning Associate Exam Guide
4. Practice Mock Exams (First 3 set are free)
5. Register for the exam
6. Review the technical requirements and run a system check
7. Review the exam guide again to identify any gaps
8. Study to fill in the gaps
9. Take your exam!

Get Ready for the Exam

How to Pass this Cert?

You'll need to answer 48 MCQ questions within 90 minutes with an accuracy of over 70%.

These questions are segmented into four Pillars:

- Databricks Machine Learning
- Data Preprocessing
- Model Development
- Model Deployment

Contact Your Instructor for Any Additional Questions

Dr. Yasir Khan

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

yasir@38labs.com

https://www.38labs.com


O'REILLY®