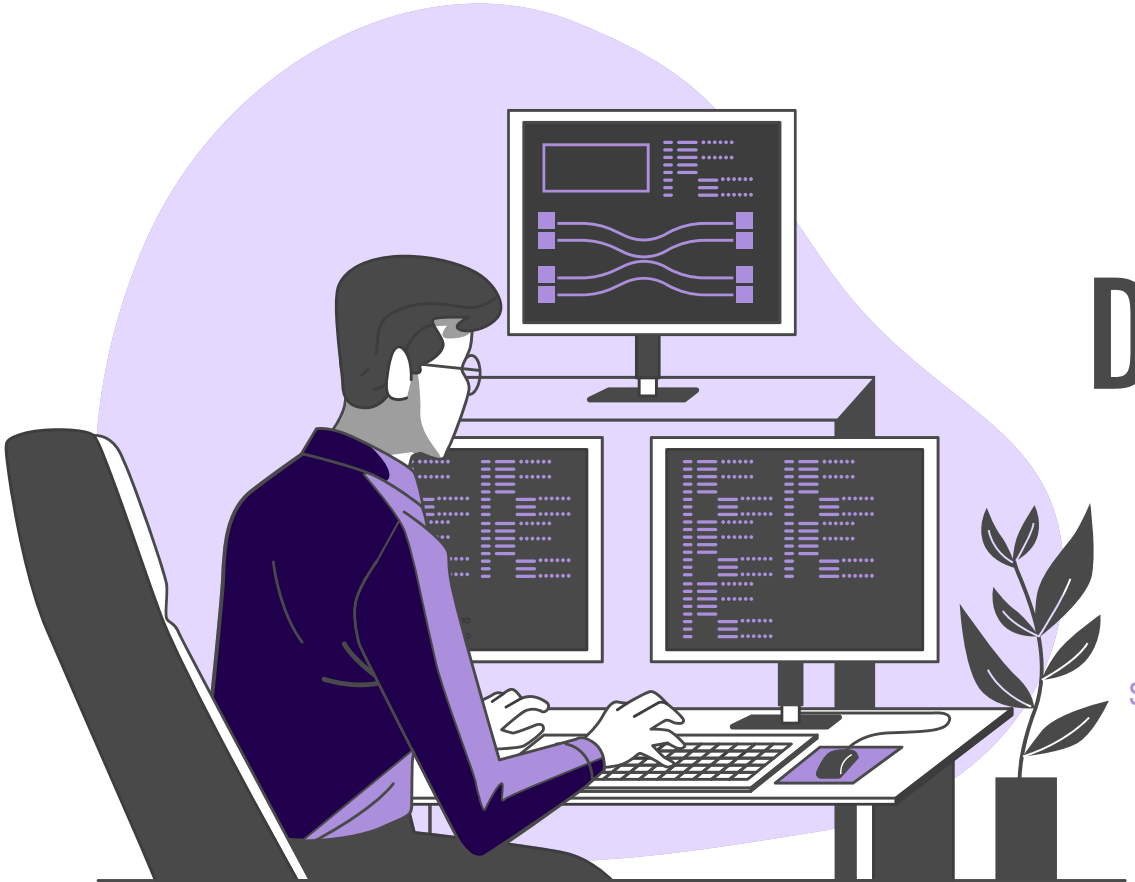


Demystifying MLOps

Rashmi Nagpal

Software Engineer (ML) @Cactus Communications

Table of Contents

Agenda of the talk:

01

Motivation: Why MLOps?

VB Events GamesBeat Data Pipeline The MetaBeat Event

The Machine
Making sense of AI

Sponsored

Why do 87% of data science projects never make it into production?

MIT Technology Review

Featured Topics Newsletters Events Podcasts

ARTIFICIAL INTELLIGENCE

The Dark Secret at the Heart of AI

No one really knows how the most advanced algorithms do what they do. That could be a problem.

02

Basics of Machine Learning Models

ARTIFICIAL INTELLIGENCE

Any technique that enables computers to mimic human behavior

MACHINE LEARNING

Ability to learn without explicitly being programmed

DEEP LEARNING

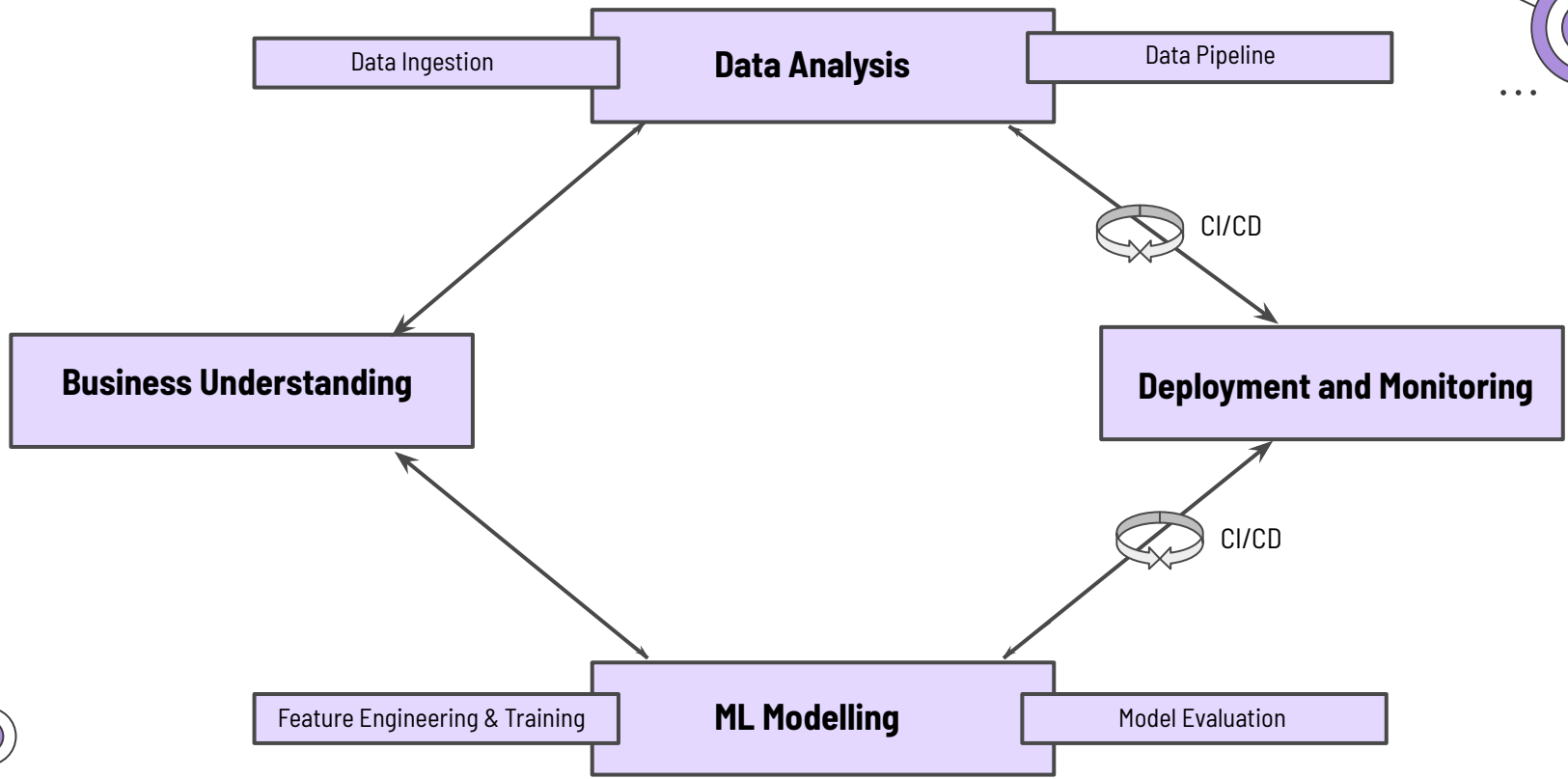
Extract patterns from data using neural networks

3 1 3 5 6 7

1 4 5 9 2 3

03

Characteristics of ML Problem


Data Ingestion
Data Analysis
Data Pipeline
CI/CD
Business Understanding
Deployment and Monitoring
CI/CD
Feature Engineering & Training
ML Modelling
Model Evaluation

Types of ML models

01

Learning Models

Supervised Learning
Unsupervised Learning

02

Statistical Models

Inductive Learning
Deductive Learning

03

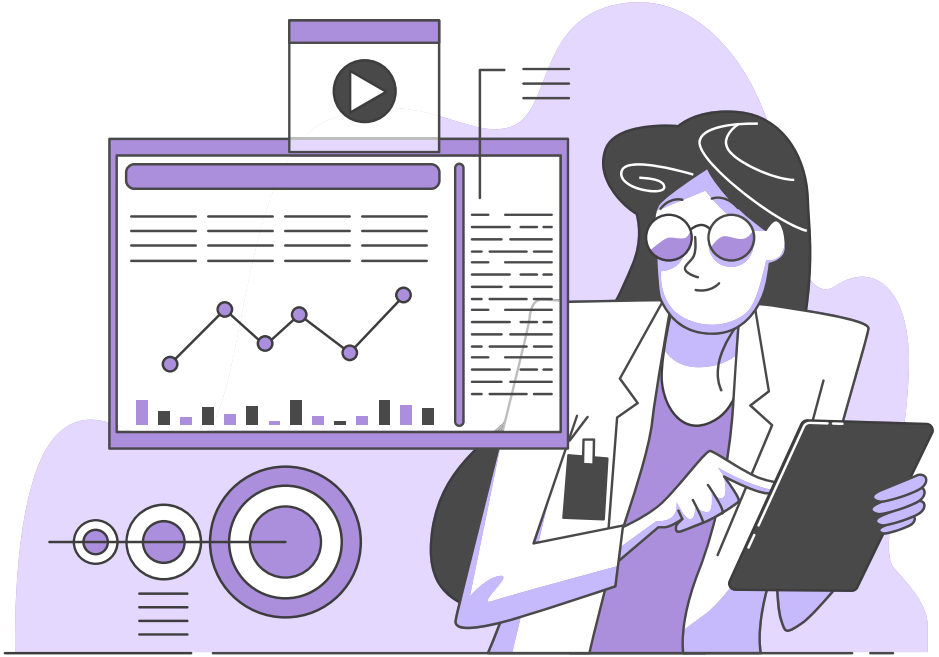
Hybrid Models

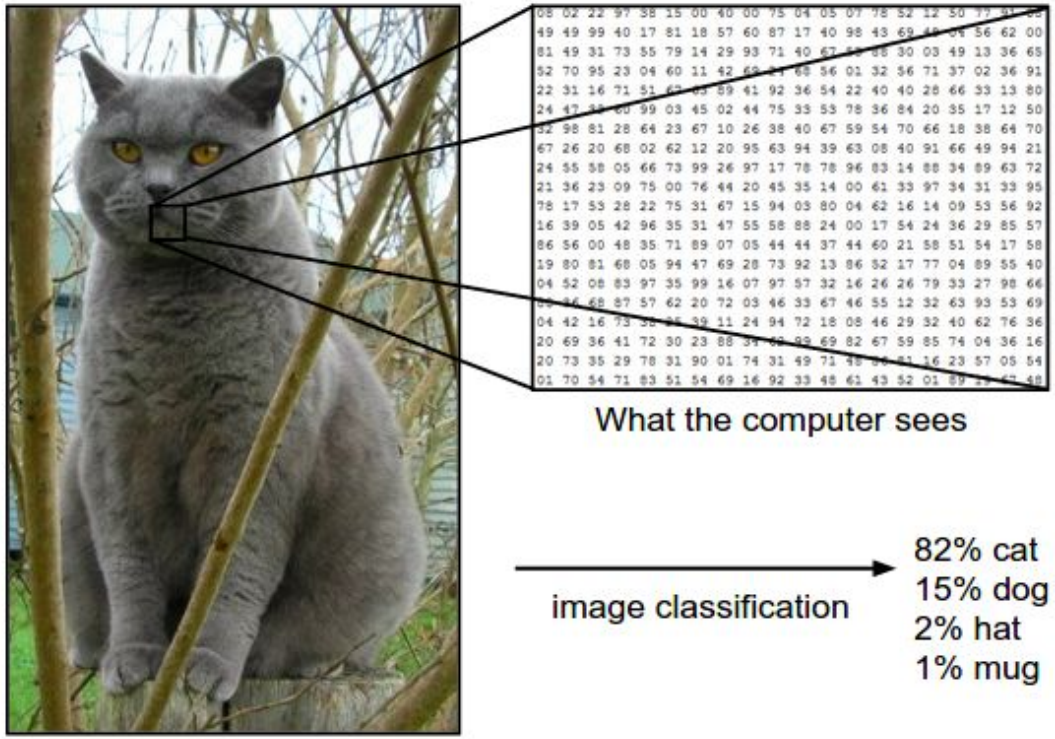
Semi Supervised Learning
Ensemble Learning
Transfer Learning

04

Human in the Loop

Active Learning
Human Reinforcement Learning




What the computer sees
image classification
82% cat
15% dog
2% hat
1% mug

Demo

Import Dataset

```
[ ] dataset = pd.read_excel("FlightDataset.xlsx")
    dataset.head()
```

	Airline	Date_of_Journey	Source	Destination	Route	Dep_Time	Arrival_Time	Duration	Total_Stops	Additional_Info	Price
0	IndiGo	24/03/2019	Banglore	New Delhi	BLR → DEL	22:20	01:10 22 Mar	2h 50m	non-stop	No info	3897
1	Air India	1/05/2019	Kolkata	Banglore	CCU → IXR → BBI → BLR	05:50	13:15	7h 25m	2 stops	No info	7662
2	Jet Airways	9/06/2019	Delhi	Cochin	DEL → LKO → BOM → COK	09:25	04:25 10 Jun	19h	2 stops	No info	13882
3	IndiGo	12/05/2019	Kolkata	Banglore	CCU → NAG → BLR	18:05	23:30	5h 25m	1 stop	No info	6218
4	IndiGo	01/03/2019	Banglore	New Delhi	BLR → NAG → DEL	16:50	21:35	4h 45m	1 stop	No info	13302

Preprocessing

Preprocessing

```
[ ] dataset.isnull().sum()
    dataset.dropna(inplace = True)
```

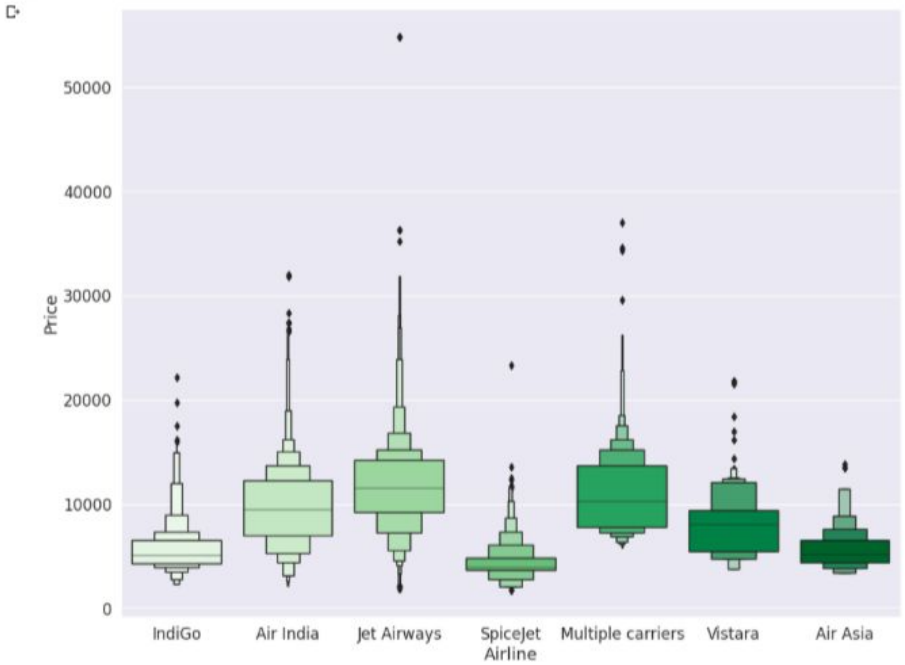
```
dataset["Airline"].value_counts()
```

```
Jet Airways                          3849
IndiGo                               2053
Air India                            1751
Multiple carriers                    1196
SpiceJet                              818
Vistara                               479
Air Asia                              319
GoAir                                 194
Multiple carriers Premium economy      13
Jet Airways Business                    6
Vistara Premium economy                 3
Trujet                                  1
Name: Airline, dtype: int64
```

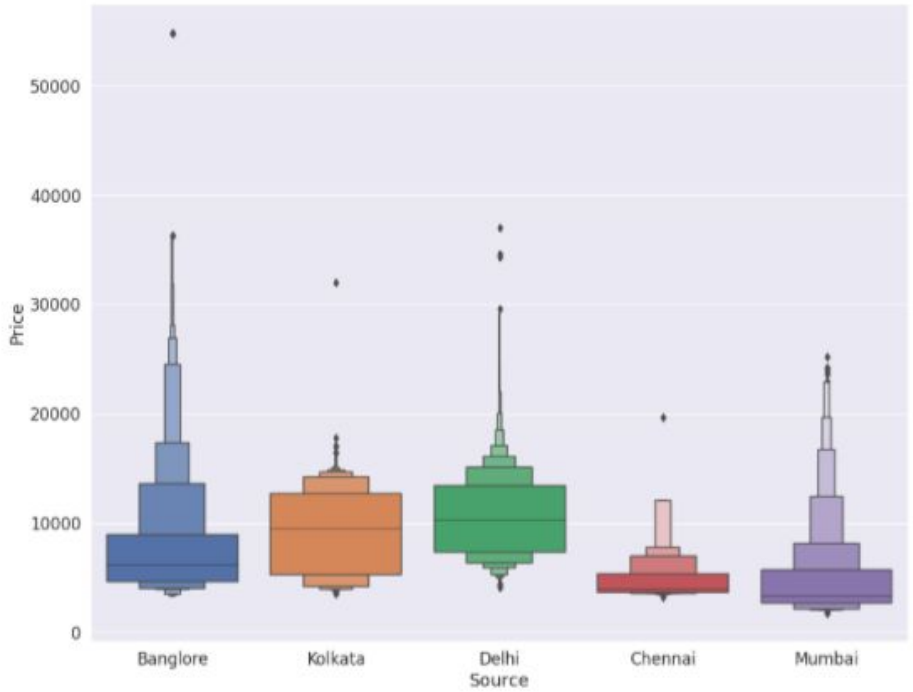
```
[ ] dataset = dataset[dataset["Airline"] != 'Trujet']
    dataset = dataset[dataset["Airline"] != 'Vistara Premium economy']
    dataset = dataset[dataset["Airline"] != 'Jet Airways Business']
    dataset = dataset[dataset["Airline"] != 'Multiple carriers Premium economy']
    dataset = dataset[dataset["Airline"] != 'GoAir']
```

Exploratory Data Analysis

```
import matplotlib.pyplot as plt
import seaborn as sns
import matplotlib as plt
from matplotlib import rcParams
import numpy as np
sns.set_style("darkgrid")

sns.set(font_scale = 1.5)
rcParams['figure.figsize'] = 15,12
sns.set_style("darkgrid")
colors = ["g", "o"]

ax = sns.boxenplot(x = 'Airline', y = 'Price', hue="Airline", data=dataset, dodge =False, palette="Greens")
ax.get_legend().set_visible(False)
```



```
import matplotlib.pyplot as plt
import seaborn as sns
import matplotlib as plt
from matplotlib import rcParams
import numpy as np
sns.set_style("darkgrid")

sns.set(font_scale = 1.5)
rcParams['figure.figsize'] = 15,12
sns.set_style("darkgrid")
colors = ["g", "o"]

ax = sns.boxenplot(x = 'Source', y = 'Price', hue="Source", data=dataset, dodge =False)
ax.get_legend().set_visible(False)
```



Destination

```
Airline = pd.get_dummies(dataset[["Airline"]], drop_first= True)
Source = pd.get_dummies(dataset[["Source"]], drop_first= True)
Destination = pd.get_dummies(dataset[["Destination"]], drop_first= True)
```

```
dataset = pd.concat([dataset, Airline, Source, Destination], axis = 1)
```

```
dataset = dataset.drop(["Additional_Info", "Arrival_Time", "Dep_Time", "Duration", "Date_of_Journey", "Route", "Airline", "Source", "Destination"], axis = 1)
```

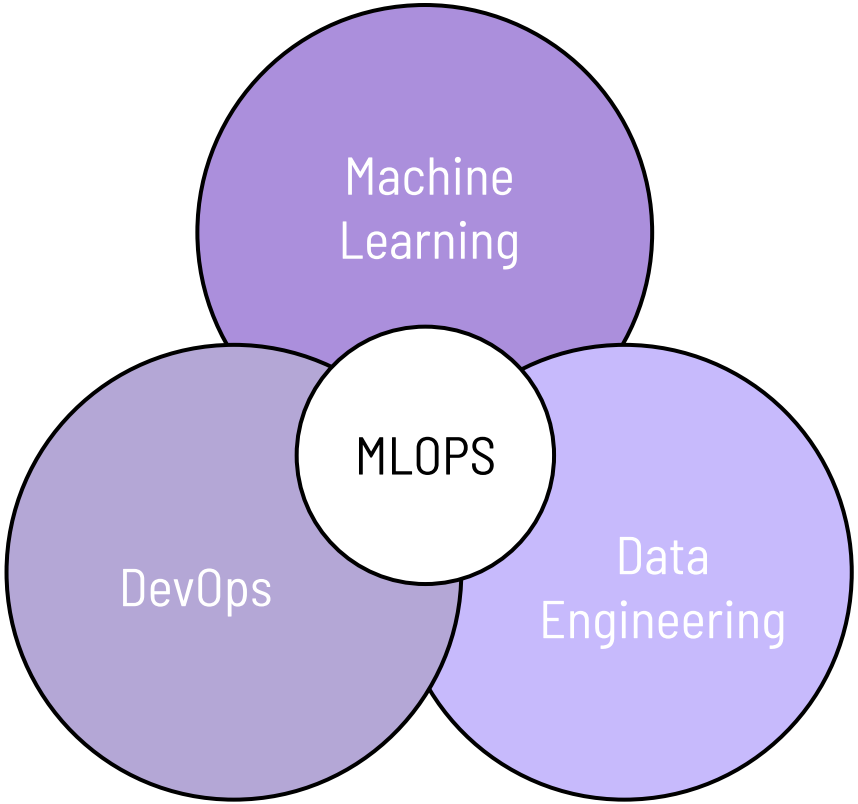
```
labels = np.array(dataset['Price'])
features= dataset.drop('Price', axis = 1)

feature_list = list(features.columns)
features = np.array(features)

# Split the data into training and testing sets
train_features, test_features, train_labels, test_labels = train_test_split(features, labels, test_size = 0.25, random_state = 42)

print('Training Features Shape:', train_features.shape)
print('Training Labels Shape:', train_labels.shape)
print('Testing Features Shape:', test_features.shape)
print('Testing Labels Shape:', test_labels.shape)
```

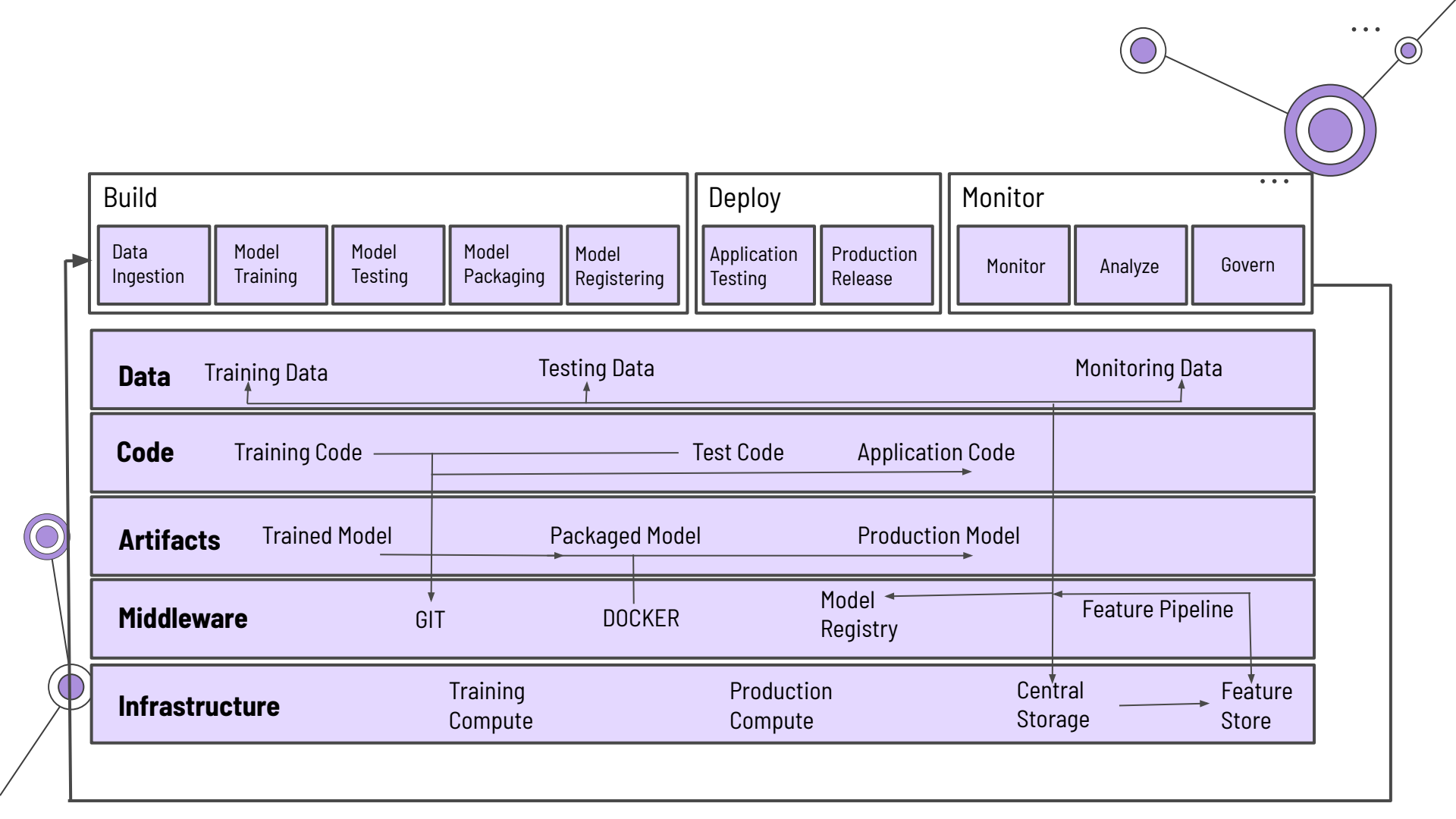
```
Training Features Shape: (7848, 24)
Training Labels Shape: (7848,)
Testing Features Shape: (2617, 24)
Testing Labels Shape: (2617,)
```

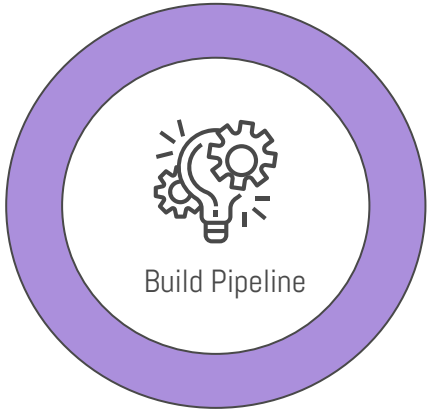
```
#Random Forest Classifier

rf = RandomForestClassifier(n_estimators = 100, random_state = 42)
rf.fit(train_features, train_labels)


# Use the forest's predict method on the test data
predictions = rf.predict(test_features)
# Calculate the absolute errors
errors = abs(predictions - test_labels)
# Print out the mean absolute error (mae)
print('\n\nMean Absolute Error:', round(np.mean(errors), 2))


# Calculating Accuracy
print("\n\nAccuracy Score:",accuracy_score(test_labels, predictions))


# Calculating Recall
print("\n\nRecall Score:",recall_score(test_labels, predictions, average='macro'))


# Calculating F1 Score
print("\n\nF1 Score:",f1_score(test_labels, predictions, average='macro'))
```

04

Framework of MLOps Workflow

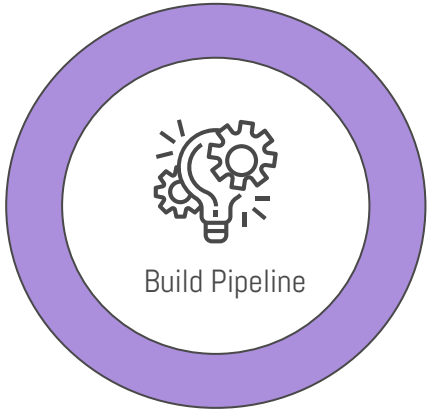

Machine
Learning
MLOPS
DevOps
Data
Engineering


Build
Data Ingestion
Model Training
Model Testing
Model Packaging
Model Registering
Deploy
Application Testing
Production Release
Monitor
Monitor
Analyze
Govern
Data
Training Data
Testing Data
Monitoring Data
Code
Training Code
Test Code
Application Code
Artifacts
Trained Model
Packaged Model
Production Model
Middleware
GIT
DOCKER
Model Registry
Feature Pipeline
Infrastructure
Training Compute
Production Compute
Central Storage
Feature Store



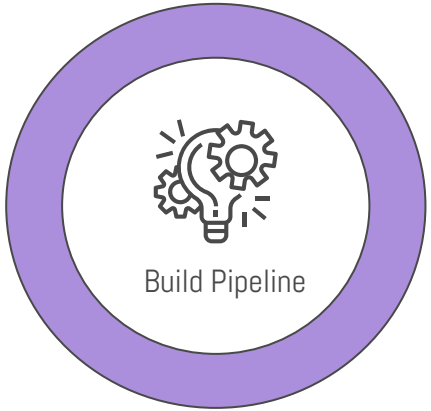
Is data good enough for ML?

- Accuracy : Inaccurate data - poor ML performance
- Reliability : Redundancy in data - unreliable decisions
- Relevance : Irrelevant information - inappropriate contextualization
- Timeliness : Obsolete data - costs business time and money



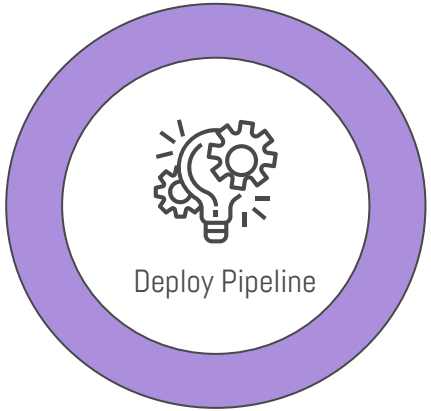
Why Data Ingestion so important?

- To make data consumable in new system before loading it (eg, masking PII)
- To speed up the availability of data for innovation and growth



How to choose which model to train?

- Performance : Depends on business use case
- Explainability : Understanding the interpretation of the results
- Complexity : More complex, harder to explain
- Dataset : NN's best for tons of data, while KNN for smaller dataset
- Dimensionality : not every model scales well with higher-dimensional data
- Training time and cost : Budget available
- Inference time : Depends on time to run a model & make predictions

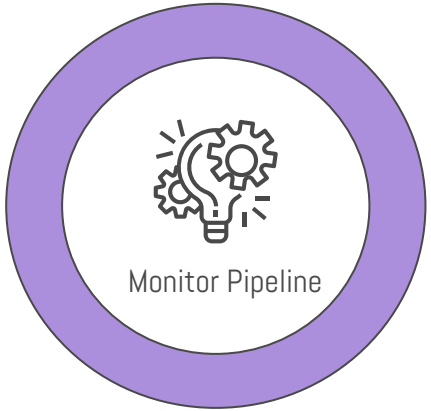


What metrics to use for model testing?

- Measure what really matters. For eg, for classification problems - PRF
- Track metrics after few iterations by using performance charts
- Testing on real unseen data to check for loss/error function
- Unit tests for model robustness on the real data

Evaluation Taxonomy

Model Performance Metrics				
Learning Models		Hybrid Models	Statistical Models	HITL Models
Supervised Learning	**Unsupervised Learning**	Rewards Return	R-square	Optimal Policy
Cross Validation	Rand Index	SHAP	Std Deviation	Risk Rate
Precision	Purity	Turing Test	Mean	Human Bias
Recall	Silhouette Coefficient	Human Vs Machine	Bias	
F-Score			Variance	
Confusion Matrix				



Why to package ML models?

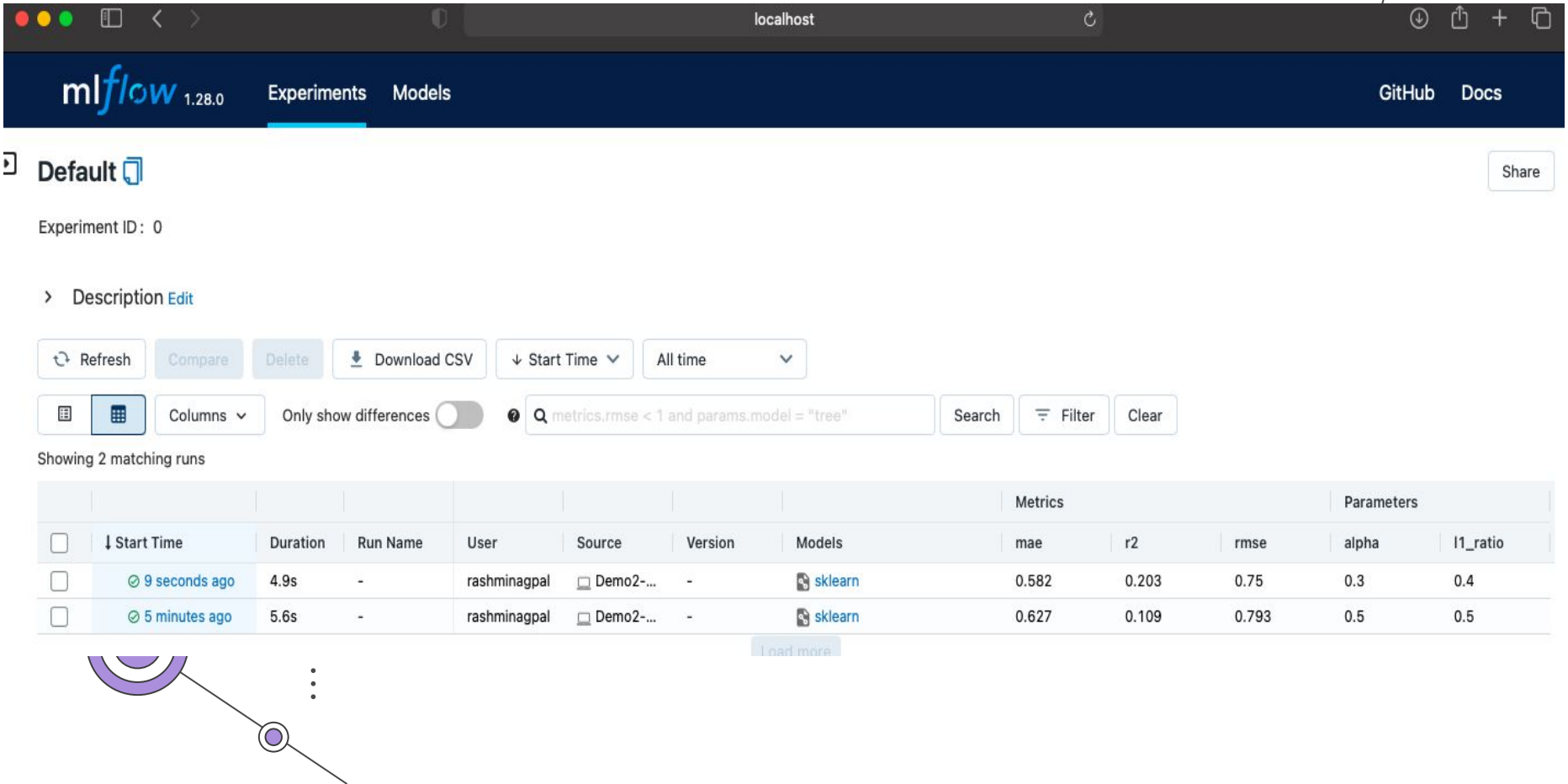
- Portability : Replicate models in various setups (VM or serverless)
- Inference : Serve ML models in real-time
- Interoperability : To fine-tune, retrain & adapt to various environments

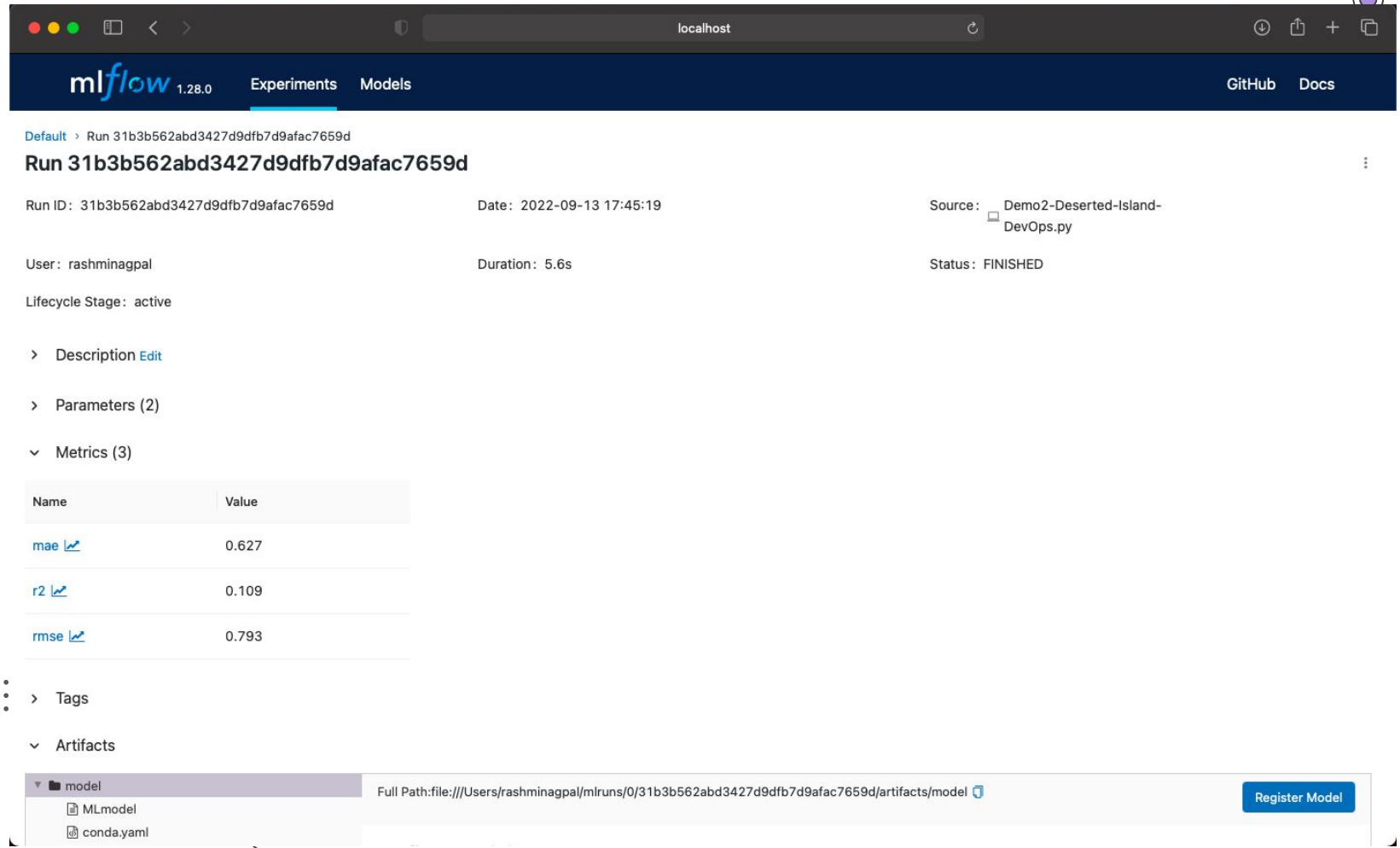
Demo

```
import os
import warnings
import sys

import pandas as pd
import numpy as np
from sklearn.metrics import mean_squared_error, mean_absolute_error, r2_score
from sklearn.model_selection import train_test_split
from sklearn.linear_model import ElasticNet
from urllib.parse import urlparse
import mlflow
import mlflow.sklearn

import logging

logging.basicConfig(level=logging.WARN)
logger = logging.getLogger(__name__)
```

```python
if __name__ == "__main__":
    warnings.filterwarnings("ignore")
    np.random.seed(40)

    # Read the wine-quality csv file from the URL
    csv_url = (
        "http://archive.ics.uci.edu/ml/machine-learning-databases/wine-quality/winequality-red.csv"
    )
    try:
        data = pd.read_csv(csv_url, sep=";")
    except Exception as e:
        logger.exception(
            "Unable to download training & test CSV, check your internet connection. Error: %s", e
        )

    # Split the data into training and test sets. (0.75, 0.25) split.
    train, test = train_test_split(data)

    # The predicted column is "quality" which is a scalar from [3, 9]
    train_x = train.drop(["quality"], axis=1)
    test_x = test.drop(["quality"], axis=1)
    train_y = train[["quality"]]
    test_y = test[["quality"]]

    alpha = float(sys.argv[1]) if len(sys.argv) > 1 else 0.5
    l1_ratio = float(sys.argv[2]) if len(sys.argv) > 2 else 0.5
```

```
with mlflow.start_run():
    lr = ElasticNet(alpha=alpha, l1_ratio=l1_ratio, random_state=42)
    lr.fit(train_x, train_y)

    predicted_qualities = lr.predict(test_x)

    (rmse, mae, r2) = eval_metrics(test_y, predicted_qualities)

    print("Elasticnet model (alpha=%f, l1_ratio=%f):" % (alpha, l1_ratio))
    print("  RMSE: %s" % rmse)
    print("  MAE: %s" % mae)
    print("  R2: %s" % r2)

    mlflow.log_param("alpha", alpha)
    mlflow.log_param("l1_ratio", l1_ratio)
    mlflow.log_metric("rmse", rmse)
    mlflow.log_metric("r2", r2)
    mlflow.log_metric("mae", mae)

    tracking_url_type_store = urlparse(mlflow.get_tracking_uri()).scheme

    # Model registry does not work with file store
    if tracking_url_type_store != "file":

        # Register the model
        # There are other ways to use the Model Registry, which depends on the use case,
        # please refer to the doc for more information:
        # https://mlflow.org/docs/latest/model-registry.html#api-workflow
        mlflow.sklearn.log_model(lr, "model", registered_model_name="ElasticnetWineModel")
    else:
        mlflow.sklearn.log_model(lr, "model")
```


localhost
mlflow 1.28.0
Experiments
Models
GitHub
Docs
Default
Share
Experiment ID: 0
Description Edit
Refresh
Compare
Delete
Download CSV
Start Time
All time
Columns
Only show differences
metrics.rmse < 1 and params.model = "tree"
Search
Filter
Clear
Showing 2 matching runs
Metrics
Parameters
Start Time
Duration
Run Name
User
Source
Version
Models
mae
r2
rmse
alpha
l1_ratio
9 seconds ago
4.9s
-
rashminagpal
Demo2-...
-
sklearn
0.582
0.203
0.75
0.3
0.4
5 minutes ago
5.6s
-
rashminagpal
Demo2-...
-
sklearn
0.627
0.109
0.793
0.5
0.5
Load more


localhost
mlflow 1.28.0
Experiments
Models
GitHub
Docs
Default › Run 31b3b562abd3427d9dfb7d9afac7659d
Run 31b3b562abd3427d9dfb7d9afac7659d
Run ID: 31b3b562abd3427d9dfb7d9afac7659d
Date: 2022-09-13 17:45:19
Source: Demo2-Deserted-Island-DevOps.py
User: rashminagpal
Duration: 5.6s
Status: FINISHED
Lifecycle Stage: active
Description Edit
Parameters (2)
Metrics (3)
Name
Value
mae
0.627
r2
0.109
rmse
0.793
Tags
Artifacts
model
MLmodel
conda.yaml
Full Path:file:///Users/rashminagpal/mlruns/0/31b3b562abd3427d9dfb7d9afac7659d/artifacts/model
Register Model

05

Principles for Monitoring ML Pipeline

Key Principles

01

Drift

Check on the degradation of the predictive model's performance eg, Feature, Data, Model drift

02

Bias

Error which leads to skewed outcomes, low accuracy or analytical errors.

03

Transparency

AI is non-deterministic in in nature, hence need to build trust in AI systems

04

Explainable AI

Ethics at the forefront, hence vital to understand the decisions the model is making.

06

Resources

Resources

Articles

- MIT Review Article : The Dark Secret at the Heart of AI
- Venturebeat Article : Why do 87% models never make into production?

Books

- Practical MLOps by Alfredo Deza and Noah Gift
- Engineering MLOps by Emmanuel Raj

Research Paper

- NeurIPS, Hidden Technical Debts in ML systems

Code

- My github, https://github.com/RN0311/Deserted-Island-DevOps-Demo

Thanks!

Happy to take any questions!

@iamrashminagpal