Introduction

18 June 2024 19:51

→ Introduction

→ Code → Jupyter notebook
↘ VS Code

→ Model Registry → Deployment

↓
Local
↓
Model training → Free EC2 t2.micro

t.2 large
↓
costly

MLOPs

Train Predictions

Data → Model → Deployment

Input ↓ .pkl, .joblib

• csv, url → data read.

→ data clearing

→ EDA (Exploratory data analysis)

Hypothesis test , plotting

→ Feature Building → New features

→ FE / Feature transformation

↓

Model (try all the models)

↓

Select the Best model (cross val)

↓

Hyper Parameter Tuning

Baysean search ← Hyper Parameter Tuning

↓ ↙ ↘

Optuna Grid Search Random Search

Hyperopt

↓ Best params.

Model (Best Params)

↓

Predictions ← Serialize

↓

Evaluate → Streamlit → Local deployment

Experimentation Tracking :-

Simple Imputer (mean value, median value, constant value) Experiments

imputer = Simple Imputer (const)

acc = 0.80

Where does ML Flow fits in MLOps

18 June 2024 19:52

Experimentation tracking

Model Registry

Record → Model → location save (Backup)

Advantages:-

1) Best models → Record

2) Best models → Versioning

LGL-bytree = 0.7

Xgboost 0.7 v-1 → 0.8

xgboost 0.8 v-2

3) MLFlow → Docker integrated

Docker image.

Repository → Deploy

Deployment → Errors ↓

xgboost_0.8 v2

Experimentation tracking :-

Project → Experiment ← Same dataset

Runs → Different iterations / combinations

Plot Gradient Boosting (n_estimators, learning_rate)

→ Parameter → n_estimators, lr.

→ Metrics → sklearn_metrics., → scores

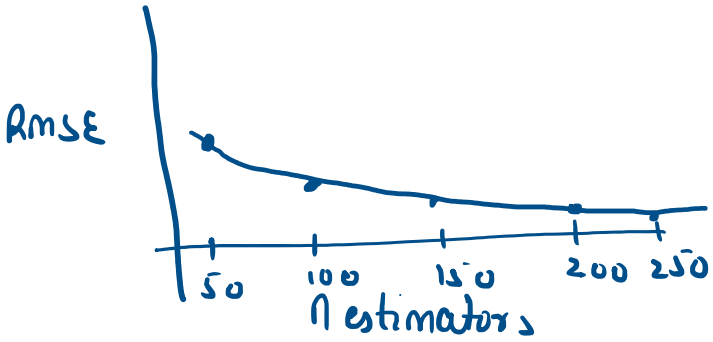
↓ sklearn_losses → loss.

(R2, RMSE)

n_est = [50, 100, 150, 200, 250] 5

lr = [0.1, 0.2, 0.3] 3

Total comb = 15



DVC → Live Experimentation

MLFlow → Server run. (URI)

Very light weight → very less code = More functionality

Model Registry → Docker compatible
↓
Deploy

Prerequisites

18 June 2024 19:52

Need to have

1) Machine learning flow → VS Code
   → notebook.

2) Tracking and Evaluation.

3) Hyperparameter tuning → Hyperopt
   → Optuna
   ↓
   Grid search

Good to have

1) MLOP's

2) Version control

3) Experimentation tracking → DVC

Capabilities

18 June 2024 19:53

1) Experimentation tracking

2) Model Registry.

3) Model Evaluation → Simple code → Reg, Clf

↓ x_test,

Confusion matrix, ROC, Metrics ← model type.

Precision, Recall, Accuracy.

Classification Report.

4) Model logging → Model + metadata.

Requirements → pip.

↓ conda

Parameters

model.get_params()

Env → virtualenv, conda

config →

↓

Prediction code → data.

Prediction code → data.

model predictions

5) Data validation → Fast API (Pydantic)

log input data + Validation

Trying to become a do it all tool

18 June 2024 19:53

DVC → Model versioning.

Deployment

Evaluation

Data Validation → Pydantic

End point → Dockers.

↓

Containerize → Push to registry

↓

Deploy on server