

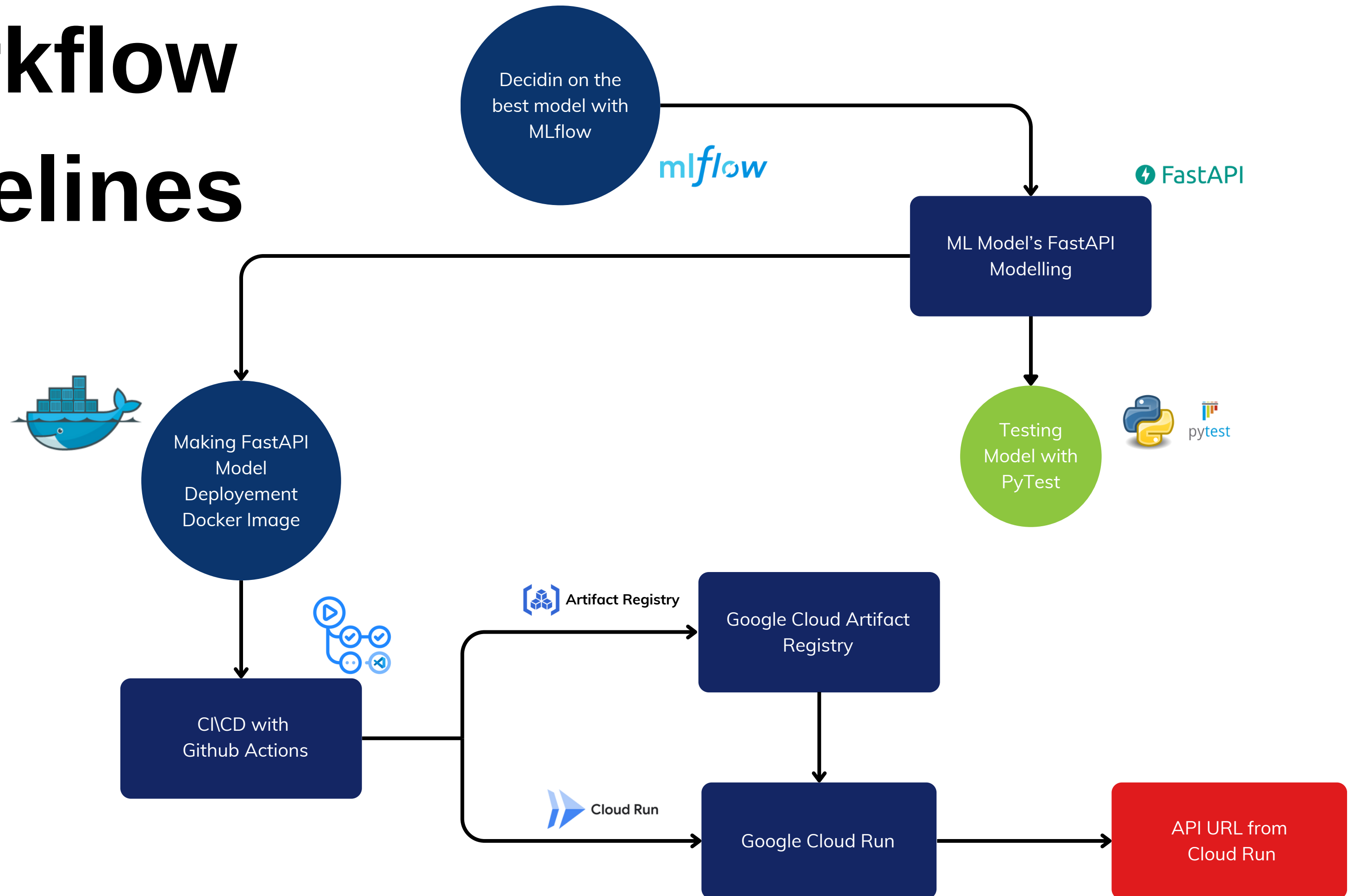


MLOps End-to-End Project

Presented by Ezgi Demirer

ISTDSA-MLOPS Final Project

Workflow Pipelines





Credit Approval Prediction with MLOps

Vision

- The Credit Approval Prediction model aims to streamline loan approval processes by leveraging machine learning for accurate and fair credit risk assessment, helping financial institutions make data-driven decisions efficiently.



Mission

- The Credit Approval Prediction model is designed to enhance the efficiency and accuracy of loan approval processes by leveraging CI/CD pipelines with GitHub Actions and deploying on Google Cloud Run. This ensures seamless integration, scalability, and real-time credit risk assessment for financial institutions.



Goals

- Improve the credit approval prediction process using machine learning to enhance financial institutions' risk assessment procedures.
- Provide a reliable, transparent, and accessible service to accelerate financial decision-making.
- Promote data-driven decision-making to establish fair and objective credit evaluation systems.



Data

	income	credit_score	debt_ratio	age	employment_years	approved
0	9270	765	0.79	30	27	0
1	2860	833	0.74	57	17	0
2	7390	704	0.25	25	33	1
3	7191	812	0.34	22	21	1
4	13964	532	0.73	26	10	0

This dataset contains 1000 rows and 6 columns.

Full dataset:
https://github.com/EzgiDemirer/mlops_endtoend

Approved: Credit approval status (0 = Rejected, 1 = Approved)	Debt Ratio: Ratio of debt to income
Income: Applicant's income	Age: Applicant's age
Credit Score: Credit rating score	Employment Years: Length of employment in years

MLflow

Model Selection Evaluation

Overall Performance (Accuracy, F1 Score, Recall)

- XGBoost achieved the highest Accuracy (0.985) score.
- It also performed well in F1 Score (0.96) and Recall (0.93).
- AdaBoost and DecisionTree showed competitive performance in Recall and F1 Score.

Selection Based on Business Requirements

- If reducing false negatives is crucial, models with high Recall should be preferred.
- If overall accuracy is the priority, XGBoost is the best choice.
- If balance is important, F1 Score should be considered (XGBoost and AdaBoost perform well here).

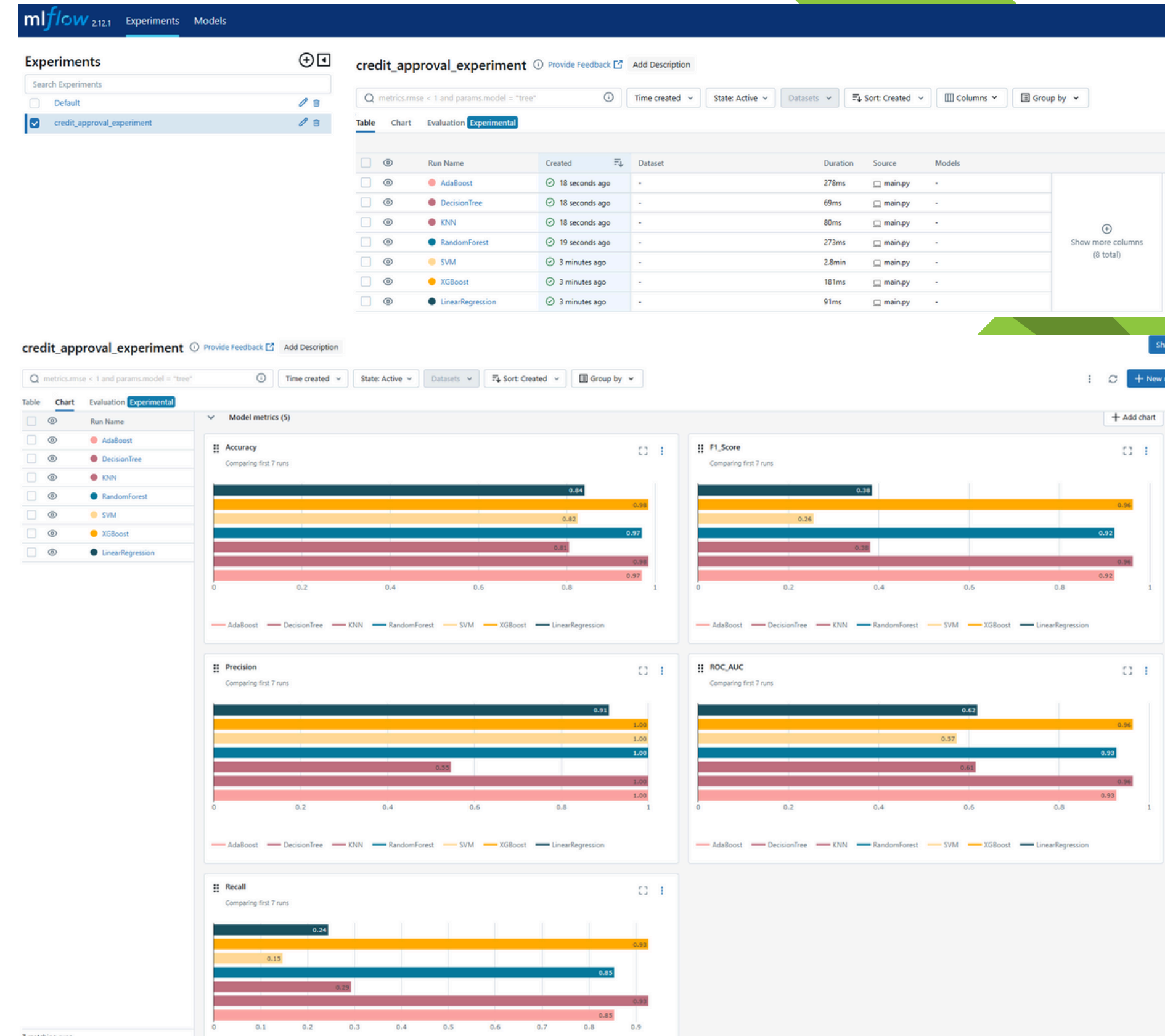
MLOps and Production Suitability

- XGBoost is a fast and efficient model, well-suited for production environments.
- AdaBoost and DecisionTree train faster but may not scale as well as XGBoost for large datasets.

Conclusion:

For an end-to-end MLOps project, the best choice is **XGBoost**.

- High overall performance
- Lower error costs
- Optimized for production deployment
- Alternatively, if speed and model size are priorities, AdaBoost or DecisionTree can be considered, but XGBoost delivers the best results.



FastAPI

Credit Approval Prediction API with FastAPI

FastAPI is a modern, high-performance web framework for building APIs with Python.

This API predicts credit approval based on applicant details such as income, credit score, debt ratio, age, and employment years.

Example Predictions:

- Approved (Green Check)
- Rejected (Red Cross)

Users input their financial details.

- The API processes the data and makes a prediction.
- The result is either Approved or Rejected.

The image displays two screenshots of a web application titled "Kredi Onay Tahmin Sistemi".

Top Screenshot: The browser address bar shows "astapi-117197913281.us-central1.run.app". The form contains the following input fields and values: Gelir: 7390, Kredi Skoru: 704, Borç Oranı: 0.25, Yaş: 25, Çalışma Yılı: 33. Below the inputs is a "Tahmin Yap" button. The prediction result is displayed as "Tahmin Sonucu: Onaylandı".

Bottom Screenshot: The browser address bar shows "fastapi-117197913281.us-central1.run.app". The form contains the following input fields and values: Gelir: 2860, Kredi Skoru: 833, Borç Oranı: 0.74, Yaş: 57, Çalışma Yılı: 17. Below the inputs is a "Tahmin Yap" button. The prediction result is displayed as "Tahmin Sonucu: Reddedildi".

CI/CD PROCESSES

x
x x
x x
x x
x x
x x
x x
x

x
x x
x x
x x
x x
x x
x x
x

GitHub Actions

Build Processes in GitHub Actions

This build workflow automates the MLOps pipeline, ensuring efficient and repeatable deployments.

Key steps in the GitHub Actions build process:

- Set up the environment (Python, dependencies, Google Cloud CLI)
- Run unit tests to validate the code
- Build and push the container image to Artifact Registry
- Deploy the model/service to Google Cloud Run

This setup ensures continuous integration and deployment (CI/CD) for machine learning applications.

← CI/CD MLOps

✓ commit #15

🏠 Summary

Jobs

✓ build

Run details

🕒 Usage

📄 Workflow file

build

succeeded 7 minutes ago in 3m 39s

- > ✓ Set up job
- > ✓ Checkout Repo
- > ✓ Set up Python
- > ✓ Install dependencies
- > ✓ Run Tests
- > ✓ Set up gcloud CLI
- > ✓ Build the container image and push to Artifact Registry
- > ✓ Deploy to Cloud Run
- > ✓ Post Set up gcloud CLI
- > ✓ Post Set up Python
- > ✓ Post Checkout Repo
- > ✓ Complete job

Artifact Registry

Google Cloud

My First Project

Search (/) for resources, docs, products, and more

Search

Artifact Registry

Repositories

Settings

+ CREATE REPOSITORY

EDIT REPOSITORY

DELETE

SETUP INSTRUCTIONS

Filter

Enter property name or value

<div><div></div></div>	Name <div></div>	Format	Type	Location	Scanning	Description	Labels	Version policy <div></div>	Encryption <div></div>	Encryption key	Immutable image tags <div></div>	Created
<div><div></div></div>	<div></div> mlops	<div></div> Docker	Standard	us-central1 (Iowa)	Active			—	Google-managed	—	False	5 days ago

Google Cloud Artifact Registry

- Artifact Registry is a fully managed repository for storing and managing Docker images and other artifacts used in CI/CD pipelines.
- In MLOps, Artifact Registry helps store containerized machine learning models, ensuring secure and scalable deployments.
- It integrates seamlessly with CI/CD workflows, enabling automated builds and deployments with GitHub Actions.

Cloud Run

Deploying FastAPI on Google Cloud Run

Google Cloud Run is a fully managed serverless platform that allows deploying containerized applications with automatic scaling. Our FastAPI application has been successfully deployed and is now accessible via a unique URL.

Key Features of Cloud Run Deployment:

- Scalability: Automatically scales up and down based on traffic.
- Security: Supports authentication and networking configurations.
- Continuous Deployment: Integrated with GitHub Actions for seamless updates.

FastAPI is live at: <https://fastapi-117197913281.us-central1.run.app>

The screenshot displays the Google Cloud Run console for a project named 'My First Project'. The 'Services' tab is active, showing a table with one service: 'fastapi'. The service is a container, located in the 'us-central1' region, and is accessible via the URL 'https://fastapi-117197913281.us-central1.run.app'. The console also shows options to 'EDIT & DEPLOY NEW REVISION' and 'SET UP CONTINUOUS DEPLOYMENT'. Below the service details, there are tabs for 'METRICS', 'SLOS', 'LOGS', 'REVISIONS', 'TRIGGERS', 'NETWORKING', 'SECURITY', and 'YAML'. A 'Predefined' dropdown and a '+ CREATE UPTIME CHECK' button are also visible.

Name	Deployment type	Req/sec	Region	Authentication	Ingress	Recommendation	Last deployed	Deployed by
fastapi	Container	0	us-central1	Allow unauthenticated	All	—	8 minutes ago	117197913281-compute@developer.gserviceaccount.com

Kredi Onay Tahmin Sistemi

Gelir:

Kredi Skoru:

Borç Oranı:

Yaş:

Çalışma Yılı:



Thank You

Presented by Ezgi Demirer

ISTDSA-MLOPS Final Project