EVERMORE

Cloud-based platform that provides machine learning tools and services

Supervisor: Ali Seydi KEÇELİ
Ataberk Asar, Ayça Akyol, Gamze Ergin
Department of Computer Science, Hacettepe University, Ankara, Türkiye

MOTIVATION:

Our motivation for developing the Evermore MLaaS web application stemmed from the recognition of the significant challenges faced by individuals and businesses in effectively harnessing the power of machine learning. As machine learning continues to permeate various industries, there exists a growing demand for accessible and user-friendly platforms that cater to users of all skill levels.

We were inspired to create a solution that democratizes access to machine learning, empowering beginners to learn and explore the field while providing advanced tools for experienced practitioners to streamline their workflow. By addressing the complexities and barriers associated with traditional ML development, we aimed to accelerate innovation, enhance productivity, and drive meaningful impact across diverse domains.

AIM:

Throughout the project, the team focused on designing and implementing features that promote learning, experimentation, and collaboration. The integration of automated tools aimed to guide users through the machine learning process, while interactive environments provided hands-on experience and immediate feedback. Additionally, the inclusion of a marketplace encouraged knowledge sharing and resource exchange within the ML community, further enhancing the platform's utility and accessibility.

DESIGN:

*O*ur MLaaS web application is designed with a microservices architecture, leveraging Spring Boot, Flask, Docker, and AWS infrastructure to create a scalable and flexible system.

System Architecture:

1. User Microservice: Manages user authentication, authorization, and profile information.

2. Payment Microservice: Handles payment processing, subscription management, and financial transactions.

3. Model Microservice: Manages machine learning models, including their deployment and execution.

4. Frontend: Provides the user interface and interactions, developed using the React framework.

Cloud Infrastructure

The application is hosted on AWS, utilizing the following services:

1. AWS RDS: PostgreSQL instance for managing relational database needs.

2. AWS S3: File storage for user-uploaded datasets, model files, and other resources.

3. Docker Containers: Each ML model runs in a separate Docker container, ensuring isolation and compatibility across various environments.

Model Development and Deployment

Model Development Process:

1. **Model Training:** Utilizes datasets uploaded to AWS S3, allowing training either locally or on cloud-based instances.
2. **Model Packaging:** Trained models are packaged into Docker containers for consistent execution environments, ensuring deployment compatibility across various infrastructures.
3. **Model Deployment:** Deployed models are accessible via RESTful APIs, facilitating seamless integration with other services and user applications.
4. **Model Evaluation:** Ensures the quality and reliability of deployed models through key evaluation metrics:
   - Accuracy: Measures the proportion of correct predictions.
   - Precision and Recall: Evaluates performance, particularly in imbalanced datasets.
   - F1 Score: Combines precision and recall into a single metric.
   - Mean Squared Error (MSE): Used for regression models, indicating the average squared difference between observed and predicted values

Algorithms

Our application supports a wide range of ML algorithms, developers are free to improvise as long as they are compatible with our containerization standards. Algorithms including but not limited to:

- Supervised Learning
- Unsupervised Learning
- Semi-Supervised Learning
- Anomaly Detection
- Deep Learning
- Reinforcement Learning
- Generative AI

Website:

https://aag-bitirme-projesi.github.io/tanitim/

Conclusion:

In conclusion, our Machine Learning as a Service (MLaaS) web application stands as a testament to our technical achievements in the realm of machine learning accessibility and usability.

One of our key technical achievements lies in the architecture of our application. By adopting a microservices-based approach, we have ensured scalability, flexibility, and maintainability. This architectural decision enables seamless integration of various components while allowing independent development and deployment, thus enhancing the agility of our platform.

Furthermore, our cloud-native infrastructure, hosted on AWS, leverages cutting-edge services such as RDS and S3 to provide reliable storage, efficient data management, and high availability. Docker containers play a pivotal role in ensuring consistent execution environments for our models, facilitating deployment across diverse infrastructures with ease.

References:

- *AWS Documentation. (n.d.). Retrieved from https://aws.amazon.com/documentation/*
- *Docker Documentation: https://docs.docker.com/*