Phase 5 Report: Model Deployment

Artur Zagitov, Ildar Zalialiev, Artem Nazarov

Contents

1 Chapter 6: Model Deployment

1.1 Deployment Strategy

The deployment of the advertisement demand prediction model is a critical step in realizing its practical value for Avito's users. Our deployment strategy focuses on providing both an API endpoint for programmatic access and a user-friendly interface for direct interaction.

1.1.1 API Endpoint Deployment

To ensure scalability and maintainability, we have chosen to deploy the model as an API endpoint using two distinct approaches:

1. **Docker Deployment:** MLflow, a platform for managing the end-to-end machine learning lifecycle, is used to package the trained MLP model, along with its dependencies, into a Docker image. This containerized approach simplifies deployment and ensures consistency across different environments. The MLflow deployment offers a REST API endpoint, allowing for integration with other services or applications.
2. **Flask API Deployment:** For direct and localized deployment, we developed a custom API using Flask, a lightweight web framework in Python. This API wraps the trained model and exposes an endpoint to process incoming prediction requests. While this approach allows for greater customization and control over the API's behavior, it may require additional configuration and management compared to the containerized MLflow deployment.

1.1.2 Gradio UI for User Interaction

To improve user experience and provide direct access to the model's predictions, we developed a graphical user interface (GUI) using Gradio, a Python library for building machine learning demos. The GUI enables users to input information about their advertisement listing, such as title, description, price, and category. Upon submission, the GUI interacts with the deployed model API (either using API from Docker or Flask) to retrieve and display the predicted deal probability.

1.2 Inference Hardware

The model can be deployed on a relatively modest hardware configuration. The size of the model does not exceed 200 KB and can be inferenced on CPU without the need of GPU. Our deployment environment consists of:

- Processor: Intel Core i5
- Memory: 16 GB RAM
- GPU: NVIDIA RTX 3050 with 4GB VRAM

1.3 Model Evaluation Under Production Conditions

To ensure the model's accuracy in a deployment setting, we performed an additional evaluation using a new sample of 10,000 advertisement listings unseen during model training. These evaluations were conducted by using the MLflow Docker containerized model.

Model	RMSE	MSE	MAE
MLP	0.2415	0.0583	0.1618

Table 1: Model Performance in Production Environment

The results in Table 1 indicate that the model maintains consistent performance in production setting and that deployment was correct, with RMSE values closely aligned with those observed during the model validation phase (Chapter 5).

1.3.1 Meeting Business Success Criteria

The consistent RMSE scores below the target threshold of 0.25 demonstrate that the deployed model successfully meets the project's primary business success criterion of providing accurate demand predictions.

1.3.2 Economic Success Criteria

While the model's accuracy in predicting deal probability may be an indicator of its potential economic impact, direct measurement of its effect on Avito's revenue requires further monitoring and analysis of user behavior post-deployment. This analysis should focus on:

- **Transaction Volume:** Tracking changes in the number of successful transactions on platform after the model's deployment.
- **User Engagement:** Analyzing how sellers are utilizing the demand predictions, including any observed changes in pricing strategies, listing quality, and overall engagement with the platform.

2 Conclusion

This project addressed the challenge of predicting demand for online advertisements on the Avito platform, aiming to enable sellers with insights to optimize their listings and improve their chances of successful transactions. Through a systematic approach consisting of data understanding, preparation, model engineering, evaluation, and deployment, we successfully developed and implemented a robust machine learning solution.

Key Achievements:

- Developed and deployed a Multi-layer Perceptron (MLP) model that accurately predicts advertisement deal probability, achieving an RMSE consistently below the target threshold of 0.25.
- Implemented an efficient and reproducible machine learning pipeline using industry-standard tools like DVC, Apache Airflow, and ZenML.
- Validated the model and the data using Giskard and Great Expectations.
- Deployed the model through both a containerized MLflow API and a custom-built Flask API.
- Created a user-friendly Gradio interface that allows sellers to directly interact with the model.

Future Improvements:

- Investigate and incorporate additional features, such as image data and user-specific information, to potentially enhance the model's predictive power further.
- Explore more complex architectures, like Recurrent Neural Networks (RNNs) or Transformer, to capture information from the text data more efficiently.
- Monitor the model's performance in the production environment and analyze its impact on key business metrics, such as transaction volume and user engagement.