
taxfix
What was your job status in Germany in 2023?
Employed
Minijob (up to €450 per month)
Student
Trainee
On parental leave
Unemployed

taxfix Senior Machine Learning Engineer

Context

At Taxfix, understanding user behavior during the tax filing process is essential for enhancing user experience and achieving business goals. This case study presents a simulated problem and dataset that mirrors real challenges we face at Taxfix. A key challenge is identifying users at risk of dropping off before completing their tax filing. By accurately predicting these potential drop-offs, we can proactively improve user engagement.

Additional Context: The goal is to predict whether a user will complete their tax filing based on their activity patterns, demographics, and engagement with the platform. The provided dataset includes **5,000 rows** and contains the following variables:

- **age**: Age of the user.
- **income**: Annual income in Euros.
- **employment_type**: Type of employment (e.g., full_time, part_time, unemployed).
- **marital_status**: User's marital status (e.g., single, married, divorced).
- **time_spent_on_platform**: Total time spent on the platform (in minutes).
- **number_of_sessions**: Total number of sessions.
- **fields_filled_percentage**: Percentage of tax fields filled by the user.
- **previous_year_filing**: Whether the user filed taxes in the previous year (binary: 0/1).
- **device_type**: Device type used by the user (e.g., mobile, desktop, tablet).
- **referral_source**: Source of referral (e.g., friend_referral, organic_search, social_media_ad).
- **completed_filing** (Target Variable): Whether the user completed the tax filing process (binary: 0/1).

Objective

Develop a deployable end-to-end machine learning solution to predict whether a user will complete their tax filing. The focus is on clean, modular, production-ready code, API deployment, and scalability.

Tasks

1. **End-to-End ML Pipeline with FastAPI Deployment** Build a modular and scalable Python codebase that includes:

    - **Data Ingestion and Preprocessing**:
        - Read and prepare the provided clean dataset (CSV with 5,000 rows).
        - Perform necessary preprocessing, including feature encoding, scaling etc.
        - Split the dataset into proper training, testing, and inference datasets to simulate real-world conditions.
    - **Model Training**:
        - Train a simple classification model (e.g., Logistic Regression or Random Forest) to predict user drop-off.
        - Evaluate the model using metrics like precision, recall, or F1-score.
    - **Model Serving with FastAPI**:
        - Develop a FastAPI service that:
            - Loads the trained model.
            - Provides a REST API endpoint for new data points.

Note: Do not focus on identifying the best model or improving the evaluation metric. A simple working training and inferencing pipeline is sufficient.

2. **Dockerization**
    - Pack the entire solution into a Docker container for ease of deployment.
3. **Deployment and Integration Strategy Document**
    - Prepare a concise document (2 pages max) that includes:

    - **Integration and Deployment Strategy**:
        - How the solution integrates into the Taxfix product workflow.
        - A diagram explaining the design and flow of the system.
        - Steps to deploy the ML solution to a cloud environment (AWS/GCP/Azure), including:
            - CI/CD pipeline setup for automated deployments.
            - Monitoring and retraining strategies to ensure model performance over time.
            - Scalability considerations (e.g., load balancing, autoscaling).

Deliverables

1. **GitHub Repository**:
    - Modular Python code implementing the full ML training and inference pipeline along with FastAPI service.
    - A README with clear setup instructions to run the pipeline and launch the FastAPI locally.
2. **Dockerized Solution**:
    - The entire ML pipeline and FastAPI service are packed into a Docker container.
3. **Deployment Document**:
    - Steps to containerize and deploy the solution in the cloud.
    - CI/CD, monitoring, and scaling strategies.

Time Guidelines

- Spend a maximum of 6 hours on this case study.
- Prioritize clean, modular code, scalability, and deployment readiness.
- If any part is incomplete, document your approach, assumptions, and next steps in the document.

Notes

- A clean dataset with 5,000 rows will be provided for this task.
- Focus on deploying a simple, working pipeline rather than achieving the most accurate model.

Good luck! 🚀