

Machine Learning Engineer (0-2 Years Experience) – Assessment Task

Objective

The purpose of this assessment is to evaluate the candidate's ability to preprocess data, train and evaluate machine learning models, deploy a simple model as an API, and follow basic MLOps practices.

Task: Build and Deploy a Machine Learning Model for Predicting House Prices

Instructions:

You are provided with a dataset containing house attributes and their corresponding prices. Your task is to:

1. **Preprocess the data** (handle missing values, feature engineering, scaling, encoding, etc.).
 2. **Train and evaluate a regression model** that predicts house prices.
 3. **Optimize the model** using hyperparameter tuning.
 4. **Deploy the model as a REST API** using Flask or FastAPI.
 5. **Write a short report** explaining your approach, decisions, and model performance.
-

Dataset

You can use any publicly available dataset such as **Kaggle's House Price Prediction Dataset** or **California Housing Dataset (from Scikit-learn)**.

Part 1: Data Preprocessing

- Load the dataset and perform exploratory data analysis (EDA).
 - Handle missing values appropriately.
 - Perform feature engineering (scaling, encoding categorical variables, feature selection).
 - Visualize correlations between features and the target variable.
-

Part 2: Model Training & Evaluation

- Split the dataset into **training and testing sets**.
- Train a **regression model** (e.g., Linear Regression, Decision Tree, Random Forest, XGBoost).
- Evaluate the model using **RMSE, MAE, and R² scores**.

- Optimize the model using **GridSearchCV** or **RandomizedSearchCV**.
 - Save the trained model using **Pickle** or **Joblib**.
-

Part 3: Model Deployment

- Build a simple **Flask** or **FastAPI** application to serve predictions.
 - Create an endpoint `/predict` that takes input features as JSON and returns the predicted price.
 - Test the API using **Postman** or **CURL**.
 - Containerize the application using **Docker** (optional for bonus points).
-

Part 4: Report & Documentation

- Provide a brief report explaining:
 - Steps taken for data preprocessing and feature engineering.
 - Model selection and optimization approach.
 - Deployment strategy and API usage guide.
 - Include **clear and well-commented code** in a GitHub repository or a Jupyter Notebook.
-

Bonus Points (Optional)

- Implement **logging and error handling** in the API.
 - Deploy the API on **AWS/GCP/Azure** or **Render**.
 - Use **DVC** or **MLflow** for model versioning.
 - Create a simple **frontend UI** to interact with the model.
-

Submission Guidelines

- Submit a **GitHub repository** containing:
 - Jupyter Notebook (`.ipynb`) or Python scripts (`.py`).
 - A `README.md` with clear instructions on running the project.
 - Model file (`.pkl` or `.joblib`).
 - Flask/FastAPI script (`app.py`).
 - Dockerfile (if applicable).
- Provide a link to a **hosted API** (if deployed).