

Phase 4

1 Phase 4: Deployment, Ethics & Final Assessment

course: Machine Learning Algorithms (MAAI).

Student Name: Mina Ezach Naeem Faltos

Student Number: 34388

1.1 A. Introduction

During Phase 3 Random Forest Regressor was successfully trained and achieved a **Mean Absolute Error (MAE) of €34.71** and R^2 score of 0.48.

Phase 4 aims at meeting the final assessment criteria by: 1. **Saving the model** to use in the future (Model Persistence). 2. **Showing Deployment** through the building of a prediction pipeline. 3. **Analyzing Ethics** namely, possible bias during Airbnb pricing. 4. **Defining a Monitoring Plan** to help with continuous improvement.

```
[22]: import pandas as pd
import numpy as np
import joblib
from sklearn.ensemble import RandomForestRegressor

# 1. Loading Data
# Wrapping this in a try-except block to handle file path variations
try:
    train_df = pd.read_csv('work_MLA_phase2_34388_train.csv')
except FileNotFoundError:
    print("Error: Train file not found. Ensure CSV is in the same folder.")

# 2. Applying Phase 3 Filter
# strictly removing outliers > 500 to match the Phase 3 success criteria.
train_df = train_df[train_df['price'] <= 500]

# 3. Preparing Features
# Converting 'room_type' to binary like in Phase 3
train_df['is_private_room'] = train_df['room_type'].apply(lambda x: 1 if x == 'Private room' else 0)

# Cleaning feature selection
```

```

features = ['accommodates', 'bathrooms', 'bedrooms', 'beds',
            ↪ 'review_scores_rating', 'latitude', 'longitude', 'is_private_room']
X = train_df[features]
y = train_df['price']

# 4. Training the Final Production Model
# Using parameters in Phase 3 Grid Search: n_estimators=100, max_depth=20
print("Training final Production Model...")
final_model = RandomForestRegressor(n_estimators=100, max_depth=20,
            ↪ random_state=42)
final_model.fit(X, y)
print("Success: Model trained on filtered data (Price <= 500).")

```

Training final Production Model...

Success: Model trained on filtered data (Price <= 500).

1.2 B. Model Persistence (Requirement 6a)

In order to use the model in a practical application (such as a web site for example) the model can't retrain each time a customer or a user visits. It is necessary to save the trained model to a file, with the help of joblib.

```

[23]: # Saving the model to a file
model_filename = 'airbnb_price_model.pkl'
joblib.dump(final_model, model_filename)

print(f"Model saved to disk as '{model_filename}'")

```

Model saved to disk as 'airbnb_price_model.pkl'

1.3 C. Deployment Simulation (Requirement 6b)

Next a **Production API** will be simulated. for example, if a user enters their apartment details on a website, The function `predict_price()` which loads the model and provides a response of a Fair Market Value acts as the backend service.

```

[24]: def predict_price(accommodates, bathrooms, bedrooms, beds, rating, lat, lon,
            ↪ room_type):
    """
    Simulates an API endpoint.
    Takes raw user input, preprocesses it, and returns a price prediction.
    """
    # 1. Loading the saved model (Simulating a server load)
    loaded_model = joblib.load('airbnb_price_model.pkl')

    # 2. Preprocess Input (Matching the training features)
    is_private = 1 if room_type == 'Private room' else 0

    input_data = pd.DataFrame({

```

```

        'accommodates': [accommodates],
        'bathrooms': [bathrooms],
        'bedrooms': [bedrooms],
        'beds': [beds],
        'review_scores_rating': [rating],
        'latitude': [lat],
        'longitude': [lon],
        'is_private_room': [is_private]
    })

    # 3. Predict
    prediction = loaded_model.predict(input_data)[0]
    return round(prediction, 2)

# TESTING THE DEPLOYMENT
# Scenario A: A Private Room in Gràcia (Should be cheaper)
price_room = predict_price(2, 1.0, 1.0, 1.0, 95.0, 41.40, 2.16, 'Private room')

# Scenario B: An Entire Apartment in Eixample (Should be expensive)
price_aprt = predict_price(4, 2.0, 2.0, 3.0, 98.0, 41.39, 2.17, 'Entire home/
↳ apt')

# Clean Print Statements
print(f"Predicted Price (Private Room): €{price_room}")
print(f"Predicted Price (Entire Apt):   €{price_aprt}")

```

Predicted Price (Private Room): €54.18

Predicted Price (Entire Apt): €242.48

1.4 D. Interpretability & Ethical Considerations (Requirement 7)

1.4.1 A. Interpretability Recap

According to the Phase 3 analysis, the predictions of the model are driven mainly by: 1. **Capacity:** `accommodates` (Strongest predictor). 2. **Location:** `longitude` and `latitude`. 3. **Quality:** `review_scores_rating`. This openness gives us an opportunity to justify to hosts why a certain price was recommended (For example, the reason behind the high price is that the location is central).

1.4.2 B. Ethical Risks & Bias Mitigation

1. **Gentrification Risk: Problem:** When the model is systemically predicting a higher price of popular neighborhoods, it strengthens its price increases, and it may drive away locals. **Mitigation:** predictions will be marked with the “Markets Estimates” and will show local rental cap laws beside the price.
2. **Bias in Ratings: Problem:** `review_scores_rating` can reflect implicit bias against minority hosts sometimes. **Mitigation: Fairness Audit.** the audit will be conducted quarterly to ensure that the model consistently underprice the listing of certain demographic groups.

In a case of bias, its possible to re-weight or remove the rating feature.

1.5 E. Continuous Improvement (Requirement 8)

1.5.1 1. Monitoring Plan

- **Drift Detection:** The baseline RMSE is €58.13. Assuming the RMSE on new data exceeds €70.00**, that would indicate a big market drift (e.g., inflation or regulatory changes), which causes a call to review.
- **Outlier Alerts:** When a user inputs `price > 500`, the system will mark it as an out of distribution due to the fact that the model is not able to predict luxury listing well.

1.5.2 2. Retraining Strategy

Schedule: Quarterly (every 3 months). **Data Source:** Use new data dumps from “inside airbnb” to reflect changes that are based on seasonal trends (like Summer vs. Winter pricing).

1.6 F. Final Project Conclusion

1.6.1 A. Project Overview

The project was able to put in place a successful implementation of an end-to-end machine learning pipeline to predict prices of Airbnb rental in Barcelona. Based on the MAAI framework, the process involved defining the problem, cleaning of the data, training of the model, evaluation, simulation of deployment and ethical auditing.

1.6.2 1. Model Performance & Strategy

The final **Random Forest Regressor** reached the following results on the unseen test data set:
* **RMSE:** €58.13 * **MAE:** €34.71 * R^2 **Score:** 0.48

These findings show that this performance in predicting is moderate but realistic in consideration of the high variability and subjective nature of the Airbnb market. In order to guarantee stability, listings above the price of at least **€500** were left out of the model, which is then focusing the model on the best possible representation of the market segment, and reducing the impact of extreme outliers.

1.6.3 2. Deployment & Consistency

In the case of deployment, a simplified implementation of the model was done with the most influential features (Capacity, Location, Room Type). This design ensures: * **Computational Efficiency:** Minimizing input complexity for real time inference. * **Consistency:** Preprocessing steps (like, binary encoding) mirror the training pipeline to prevent inference errors.

1.6.4 3. Ethics & Continuous Improvement

- **Ethical Mitigation:** it’s agreed that there are bias in location-based pricing and user ratings. The predictions are clearly denoted as being merely a Market Estimates and not objective truths to avoid strengthening gentrification or inequality.

- **Monitoring:** There is an effective maintenance strategy, which sends an alarm when the **RMSE is more than €70.00 or when there is a data drift. The scheduled quarterly retraining will make sure that the model is able to adjust with the seasonal trends.

1.6.5 4. Final Verdict

Generally, the project has met all the technical, methodological and ethical requirements. It offers a strong, clear and understandable basis that can be used in practice and continued improvement in the future.