

## Phase 5

### 1 Phase 5: Model Monitoring and Lifecycle Management

**course:** Machine Learning Algorithms (MAAI).

**Student Name:** Mina Ezach Naeem Faltos

**Student Number:** 34388

#### 1.1 A. Introduction

A machine learning model is a living asset. The performance of the Random Forest model will automatically degrade with the passing of time as the Barcelona rental market changes. This phase defines the concept of MLOps (Machine Learning Operations) to make sure that the model stays accurate by implementing the automatic auditing and lifecycle management.

**Objectives:** 1. **Monitoring of performance:** detecting when the model's error (RMSE) drifts. 2. **Drift Identification:** differentiating between Data Drift and Concept Drift. 3. **Lifecycle Policy:** Determine retraining schedules and model retirement regulations.

```
[3]: import pandas as pd
import numpy as np
import joblib
from sklearn.metrics import mean_squared_error

# 1. Loading the Champion Model from Phase 4
try:
    model = joblib.load('airbnb_price_model.pkl')
    print("SUCCESS: Production Model Loaded.")
except FileNotFoundError:
    print("ERROR: 'airbnb_price_model.pkl' not found.")

# 2. Thresholds of the performance
BASELINE_RMSE = 58.13
DRIFT_THRESHOLD = 70.00 # Triggering an alert if RMSE exceeds this

def monitor_production_performance(new_data_path):
    """
    Simulates a monitoring service that audits the model against new data.
    """
    # Loading simulated new data
```

```

new_data = pd.read_csv(new_data_path)
new_data = new_data[new_data['price'] <= 500]

# Feature Engineering
new_data['is_private_room'] = new_data['room_type'].apply(lambda x: 1 if x_
↳== 'Private room' else 0)
features = ['accommodates', 'bathrooms', 'bedrooms', 'beds',_
↳'review_scores_rating', 'latitude', 'longitude', 'is_private_room']

# Predictions and Evaluation
X_new = new_data[features]
y_true = new_data['price']
y_pred = model.predict(X_new)

current_rmse = np.sqrt(mean_squared_error(y_true, y_pred))

print(f"--- PERFORMANCE MONITOR REPORT ---")
print(f"Baseline RMSE: €{BASELINE_RMSE}")
print(f"Current RMSE: €{current_rmse:.2f}")

if current_rmse > DRIFT_THRESHOLD:
    print("ALERT: Performance Drift Detected! The model requires retraining.
↳")
else:
    print("STATUS: Model Healthy. No significant drift detected.")

# Executing Monitor
monitor_production_performance('work_MLA_phase2_34388_test.csv')

```

```

SUCCESS: Production Model Loaded.
--- PERFORMANCE MONITOR REPORT ---
Baseline RMSE: €58.13
Current RMSE: €57.54
STATUS: Model Healthy. No significant drift detected.

```

## 1.2 B. Drift Detection Strategy

To safeguard the consistency of the predictions, observing two kinds of shifts is important:

- **Data Drift (Feature Drift):** This happens when the statistical distribution of the inputs changes. As an example, in the event that a large number of high-rating properties suddenly enter the market, the input distribution for `review_scores_rating` shifts.
- **Concept Drift (Relationship Drift):** This happens when the relationship between features and price changes. E.g. when a “Private Room” becomes less valuable compared to an Entire Home because of changing preferences of the travelers, despite the fact that the features did not change.

This can be tracked by measuring the Rolling RMSE. If the error rises steadily beyond the €70

**threshold**, the concept the model learned is no longer valid.

```
[4]: # MLOps Retraining and Versioning Logic

def trigger_retraining_protocol(reason):
    """
    Simulates the documentation of a retraining event.
    """
    new_version = "v1.1_2026_01"
    print(f"INITIATING RETRAINING: {reason}")
    print(f"New Model Version Created: {new_version}")
    print("Action: Model v1.1 will now undergo A/B Testing against the current_↵
    ↵Champion.")

# Performance alert triggering retraining
trigger_retraining_protocol("RMSE exceeded €70 threshold")
```

```
INITIATING RETRAINING: RMSE exceeded €70 threshold
New Model Version Created: v1.1_2026_01
Action: Model v1.1 will now undergo A/B Testing against the current Champion.
```

### 1.3 C. Lifecycle Management Policy.

To make sure that the system remains professional and optimized, these lifecycle rules are followed:

1. **Scheduled Retraining:** The model is retrained every 3 months (scheduled as Quarterly) to reflect seasonal changes (e.g., Summer vs. Winter pricing in Barcelona).
2. **Champion/Challenger Testing:** New models are never deployed directly. They are intentionally tested as “Challengers” against the current (Champion) on a small percentage of data to make sure that they actually perform better.
3. **Model Retirement:** All the models that are above the age of 12 months are automatically archived. This will prevent from the use the old economic assumptions of several years ago or a year ago, as it will mostly be outdated from the present.

### 1.4 D. Phase 5 Summary

Phase 5 defines that the model is not a one time script, but a managed asset. By setting a **€70 RMSE monitoring threshold** and defining a clear **retraining lifecycle**, it can be ensured that the Airbnb price predictor is accurate, ethical, and still worth the business as the Barcelona market changes.