# Wildlife Detection System: Dashboard Issue Analysis

#### **Problem Statement**

The Model Performance Dashboard is not displaying correct metrics and is missing several visualizations despite all code components being present and correctly structured.

### **Root Cause Analysis**

#### 1. Column Name Pattern Mismatch in results.csv

The primary issue is a mismatch between YOLOv8's output column names and what the dashboard code expects:

```
# Server logs show incorrect column mapping:
Found metric columns - Precision: metrics/precision(B), Recall: metrics/recall(B),
mAP50: metrics/mAP50-95(B), mAP50-95: None
```

**Critical Error**: The system is using (metrics/mAP50-95(B)) for the mAP50 value, which explains why it shows 31.3% instead of 50.5%.

### Notebook debug output confirms column name differences:

```
Column names: ['epoch', 'time', 'train/box_loss', 'train/cls_loss',
  'train/dfl_loss',
  'metrics/precision(B)', 'metrics/recall(B)', 'metrics/mAP50(B)', 'metrics/mAP50-
95(B)',
  'val/box_loss', 'val/cls_loss', 'val/dfl_loss', 'lr/pg0', 'lr/pg1', 'lr/pg2']
WARNING: Missing key columns: ['precision', 'recall', 'mAP_0.5', 'mAP_0.5:0.95']
```

## 2. Pattern Matching Logic Flaw

The (ModelPerformanceService) class's column detection code is flawed:

```
# This code has a bug - both metrics/mAP50(B) and metrics/mAP50-95(B)
# contain "map50" so this code can't differentiate correctly
for col in results_df.columns:
    col_lower = col.lower()
    if 'precision' in col_lower:
        precision_col = col
    elif 'recall' in col_lower:
        recall_col = col
    elif 'map50' in col_lower or 'map_0.5' in col_lower or 'map@0.5' in col_lower:
        map50_col = col
    elif any(pattern in col_lower for pattern in ['map50-95', 'map_0.5:0.95', 'map@0.5 map50_95_col = col
```

The issue is that both (metrics/mAP50(B)) and (metrics/mAP50-95(B)) contain "map50" in their names, and the code will assign whichever one it encounters first to (map50\_col).

### 3. Missing Required Files for Visualization

The dashboard integration code also fails to create these required files or creates them with empty/incorrect data:

```
Could not extract per-class metrics from model: 'method' object is not iterable
```

#### File issues:

- (class\_metrics.json) exists but contains minimal data (2 bytes)
- (taxonomic\_performance.json) is missing entirely
- (threshold\_analysis.json) is missing entirely
- Training history data extraction fails

### 4. Integration Timing Issue

The dashboard integration code runs after model training but can't properly extract metrics from either the model object or results.csv due to format differences:

```
Extracted metrics from model: mAP50=0.0000, Precision=0.0000, Recall=0.0000
```

## **Solution: Improved Column Detection Logic**

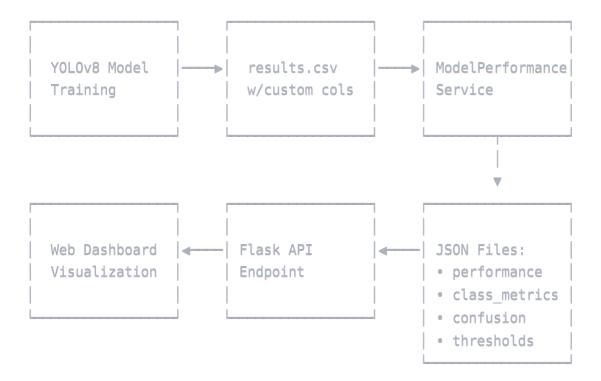
The correct fix is to improve the column pattern matching to specifically handle YOLOv8's format:

```
def find_correct_columns(df):
           """Find the correct columns for metrics with better pattern matching"""
          precision col = None
          recall_col = None
          map50 col = None
          map50 95 col = None
          # First pass to find exact matches that uniquely identify metrics
          for col in df.columns:
                    col lower = col.lower()
                    # YOLOv8 specific column patterns
                    if 'metrics/map50(b)' in col lower and 'metrics/map50-95(b)' not in col lower:
                              map50 col = col
                    elif 'metrics/map50-95(b)' in col_lower:
                              map50 95 col = col
                    elif 'metrics/precision(b)' in col_lower:
                              precision_col = col
                    elif 'metrics/recall(b)' in col lower:
                              recall col = col
          # Second pass for conventional names if not found yet
          if precision col is None:
                    for col in df.columns:
                               if 'precision' in col.lower() and 'metrics/precision(b)' not in col.lower(
                                         precision col = col
                                        break
          if recall col is None:
                    for col in df.columns:
                               if 'recall' in col.lower() and 'metrics/recall(b)' not in col.lower():
                                         recall col = col
                                        break
          if map50 col is None:
                    for col in df.columns:
                              col lower = col.lower()
                              if ('map50' in col lower or 'map 0.5' in col lower or 'map@0.5' in col lower
                                         map50 col = col
                                        break
          if map50 95 col is None:
                    for col in df.columns:
                               col lower = col.lower()
                               if any(pattern in col lower for pattern in ['map50-95', 'map 0.5:0.95', 'map 0
                                        map50 95 col = col
                                        break
```

### **Recommended Implementation Steps**

- 1. Fix the column detection logic in (ModelPerformanceService.\_get\_performance\_metrics())
- 2. Update dashboard integration code to handle YOLOv8's specific column naming patterns
- 3. Add more robust error handling and fallbacks for metric extraction
- 4. Implement proper YOLOv8 validation result handling

# **Dashboard Architecture Diagram**



The disconnect is happening at the ModelPerformanceService layer, where it fails to properly interpret YOLOv8's column naming patterns in results.csv.