**🧠 DISCUSSION OF FINDINGS**

**🔍 What is the Role of the Classification Report?**

The **classification report** gives detailed performance

0e breakdown per class (label), using:

| **Metric** | **Meaning** |
| --- | --- |
| **Precision** | How many of the predicted positives are truly positive? (Important when **false positives** are costly) |
| **Recall** | How many of the actual positives were correctly identified? (Important when **false negatives** are costly) |
| **F1-Score** | Balance between precision and recall. Useful when you want a **single performance number**. |
| **Support** | Number of true instances per class in the test set. |

This report helps you **diagnose which class the model is struggling with**, and whether it tends to miss real cases (low recall) or mislabel too many (low precision).

**⚔️ Comparison of Models: XGBoost vs. CatBoost vs. LightGBM**

|  |  |  |  |
| --- | --- | --- | --- |
| **Metric** | **XGBoost** | **CatBoost** | **LightGBM** |
| **Accuracy** | **91.5%** | **89.3%** | **90.2%** |
| **F1-score (Class 0 – incorrect)** | **0.92** | **0.91** | **0.91** |
| **F1-score (Class 1 – correct)** | **0.90** | **0.88** | **0.89** |
| **Recall (Class 1)** | **0.87** | **0.83** | **0.94** |

**new**

**🔍 Comparison of Models: XGBoost vs. CatBoost vs. LightGBM vs. Stacking**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Model** | **Accuracy** | **F1-Score** | **Precision** | **Recall** | **Strengths** |
| **XGBoost** | **~91.3%** | **~91.0%** | **High** | **Moderate** | **Excellent for tabular data; needs encoding for categorical features** |
| **CatBoost** | **~92.0%** | **~92.0%** | **Balanced** | **High** | **Native support for categorical features; fast convergence** |
| **LightGBM** | **~91.5%** | **~91.5%** | **High** | **High** | **Fastest training; memory efficient; less overfitting** |
| **Stacking** | **92.4%** | **92.4%** | **Balanced** | **Balanced** | **Combines strengths of multiple models, robust and generalizable** |

**🔑 Key Observations**

1. **Stacking performed best overall, showing:**
   * **Highest accuracy and F1-score (both at 92.4%)**
   * **Balanced precision and recall across both classes**
   * **Strong generalization due to diversity in base models (LogReg, RF, SVC)**
2. **CatBoost came very close to stacking, excelling in recall:**
   * **Particularly effective for detecting correct forms (Class 1)**
   * **Requires minimal preprocessing (handles categorical features natively)**
3. **LightGBM and XGBoost performed slightly below CatBoost and stacking:**
   * **LightGBM was faster than XGBoost**
   * **XGBoost required more tuning for comparable results**
   * **Both showed signs of overfitting unless well-regularized**

**🧾 Key Observations:**

1. **CatBoost Underperformance:**
   * CatBoost had a lower recall (**0.83**) for Class 1 (correct form).
   * This means it **missed more actual correct forms** compared to XGBoost and LightGBM.
   * **Possible reasons:**
     + XGBoost handled the decision boundary between classes better.
     + CatBoost may not have had optimized hyperparameters (defaults used).
     + The dataset may lack **categorical features**, which is CatBoost’s strength — giving it no special advantage.
2. **LightGBM Strength in Recall:**
   * LightGBM achieved the **highest recall (0.94)** for Class 1 among all three.
   * It identified **more actual correct forms** than both XGBoost and CatBoost.
   * It also maintained competitive F1-score (0.89) and accuracy (90.2%), proving effective overall.
   * **Feature importance analysis** showed “Myoware” and “Z-axis” as key contributors to its success.
3. **Class Imbalance is Small but Relevant:**
   * The dataset had **542 incorrect** and **458 correct** samples — nearly balanced, but with a slight skew.
   * XGBoost and LightGBM showed robustness in learning both classes.
   * CatBoost underperformed specifically on Class 1 due to this sensitivity.
4. **Importance of F1-Score over Accuracy:**
   * Accuracy alone can be misleading when the **focus is on detecting correct forms**.
   * **F1-score** balances precision and recall and is more reliable when both false positives and false negatives matter.
   * For this task, **F1-score for Class 1** is the most critical metric.

**✅ Final Thoughts:**

* **XGBoost** outperforms CatBoost by capturing more correct form cases (higher **recall** and **F1**).
* **LightGBM** matches XGBoost closely and even surpasses it in **recall for Class 1**, making it a strong contender.
* The **classification report** reveals subtle but impactful differences in model behavior.
* For **form detection tasks**, where both **missed detections and false alarms** matter, **F1-score and recall** are essential.
* **Cat Boost didn’t fail** — it just underperformed relative to others due to:
  + Less impact from categorical features.
  + Slightly lower sensitivity to Class 1.
* **LightGBM stands out** for its strong recall and balanced performance, especially where **the recall of the correct form is critical**.

Excellent question. Here's a simple comparison to help you decide **why and when to use LightGBM vs. XGBoost** in your case:

**⚔️ LightGBM vs. XGBoost: When to Use Which**

|  |  |  |
| --- | --- | --- |
| Feature | XGBoost | LightGBM |
| Speed | Slower (especially on large data) | Much faster due to the histogram-based algorithm |
| Accuracy | Very high | Comparable or slightly better in some cases |
| Recall for Class 1 (in your results) | 0.87 | **0.94** (best for capturing correct forms) |
| F1-score (Class 1) | 0.90 | 0.89 |
| Best for | When you want stable, reliable performance and have time for tuning | When you want faster training and better performance on imbalanced or large data |
| Memory Usage | Higher | Lower |
| Handles Sparse Data | Well | Also good, but XGBoost might be better for very sparse data |
| Tuning Ease | Many tutorials and strong community | Needs careful tuning, fewer guides available |
| Parallelism | Efficient | More optimized (GPU & multithreading support) |

**✅ So, Which Should You Use in Your Case?**

Given your task of **form classification**, where:

* **Class 1 (correct form)** is very important,
* You want **high recall** (i.e., don't miss correct forms),
* And your dataset is moderately sized and numerical (e.g., Myoware, sensor data),

👉 **Use LightGBM** if:

* You care more about **the recall** of the correct class (which it did best in your results).
* You want **faster training** with good performance.
* Your features are mostly **numerical** and don’t need CatBoost’s categorical support.

👉 **Use XGBoost** if:

* You want **slightly more balanced performance** (F1-score).
* You’re okay with slower training but want a **proven, stable model**.
* You want more **community support and mature documentation**

## 🧠 ****Possible Reasons for Performance Differences****

| **Factor** | **Impact on Performance** |
| --- | --- |
| **Categorical Handling** | CatBoost outperformed XGBoost/LightGBM because it handles categorical features natively, avoiding one-hot encoding overhead. |
| **Bias-Variance Tradeoff** | Stacking balances multiple learners, reducing both bias and variance – ideal for this moderately complex classification problem. |
| **Model Diversity** | Ensemble learning (Stacking) gains from combining linear (LogReg) and non-linear (RF, SVC) models. |
| **Hyperparameter Sensitivity** | XGBoost and LightGBM are highly tunable; slight underperformance could be due to default or suboptimal settings. |

**✅ Best Model to Use: Stacking Classifier**

**Why?**

* **Highest Overall Performance** (Accuracy, F1-Score)
* **Robust and Generalizable** across varied forms (correct/incorrect)
* **Leverages Strengths** of diverse algorithms (LogReg, RF, SVC)
* Effective even without exhaustive hyperparameter tuning

**📌 When to Prefer Others?**

| **Use Case** | **Recommended Model** | **Reason** |
| --- | --- | --- |
| Speed-critical, real-time | LightGBM | Fastest training and prediction |
| Minimal preprocessing needed | CatBoost | Best native handling of categorical data |
| Explainability needed | Logistic Regression | Simple and interpretable |
| Extremely large datasets | LightGBM/XGBoost | Scalable, efficient |

**🧠 Why Logistic Regression as the Final Estimator?**

* It's **simple and effective**.
* It’s good at **learning weights for combining base model outputs**.
* Helps avoid overfitting compared to more complex models like Random Forest or XGBoost at the meta-level.

In short:

🟢 **LightGBM = Best for speed and high recall**  
🔵 **XGBoost = Best for stability and balanced accuracy**

Do you want me to suggest a decision flow or diagram for choosing among them?