

# Group 104 - ML Final Report

## Introduction / Background

Smart home technologies, including sensors embedded in toilets, chairs, and beds, are gaining traction for health monitoring, particularly for the elderly or those with chronic conditions.

As people age or develop disabilities, their physical capabilities change, and a smart bathroom environment with health monitoring features could play a crucial role in addressing these challenges. Our focus is on using sensors and force readings to monitor physical activity, enabling real-time health assessments and tailored treatments.

These systems can track daily activities, provide feedback, and trigger alerts when abnormal behavior is detected. Existing research shows that sensor-based systems can identify changes in physical activity, offering valuable data for caregivers and healthcare professionals. However, challenges remain in accurately classifying and interpreting movement patterns, especially for people with physical impairments. This study aims to leverage machine learning to enhance sensor data analysis, improve classification accuracy, and reduce false positives.

## Motivation

The purpose of this research is to detect unexpected behaviors such as abnormal sitting or rising patterns in participants with physical disabilities. This offers several key benefits that drive the motivation for our study:

- **Health Monitoring:** Using sensor data for real-time health monitoring to help detect unexpected behaviors such as abnormal sitting or rising patterns in participants indicating health deterioration.
- **Behavioral Insights:** Analyzing long-term movement patterns of those with disabilities to identify patterns or changes in movement that can help understand the needs of patients and develop more effective treatments [3].
- **Preventative Care:** Using alert systems that notify family members or caretakers in the event of a fall or prolonged sitting can help prevent further health deterioration.

## Problem Definition

The problem at hand is to develop a machine learning model that can accurately classify different movement states of an individual sitting on a toilet, using sensor data. These movements include:

- **Sitting:** The person is seated on the toilet.
- **Offboarding:** The person is leaving the seat.
- **Onboarding:** The person is approaching and sitting on the seat.
- **Not on Toilet:** The person is not seated on the toilet.

The primary challenge is to create a model that can accurately classify these states based on sensor data, while minimizing false positives and ensuring that abnormal behaviors, such as prolonged sitting or difficulties in standing, are detected. This is especially important for individuals with physical impairments, as it can enable timely interventions and improved care.

## Dataset Description



Our dataset consists of force sensor readings from 40 people with 20 trials each. Some key features include:

- Timestamps (Unix format): These represent the exact time when the sensor data was captured.
- Load sensor values: These are readings from several sensors scattered around the toilet seat. These numbers will be useful in determining the Center of Pressure (COP).

Dataset Link: <https://tinyurl.com/toiletseatdata>

## Data Preprocessing Methods



## Data Cleaning and Selection

The raw sensor data often contains noise and irrelevant features. Therefore, in order to optimize model performance:

- We removed all unnecessary columns, leaving only the timestamps and the four critical sensor values. For instance, columns such as floor tile data and grab bar data, which were irrelevant to our goal of investigating movement patterns specific to the toilet seat, were excluded.
- The data was filtered to remove outliers that were outside the normal range, as well as any invalid sensor readings, to ensure the integrity of the dataset.
- Timestamps with missing or invalid data were removed.
- The filtered data was then compiled into a single CSV file for consistency and ease of analysis.
- To standardize timestamps across different trials and make them easier to process, Unix timestamps were converted into the number of seconds since data collection began.

## Feature Engineering

To enhance the model's ability to identify patterns and reduce dimensionality:

- **Feature Extraction:** To simplify the analysis while retaining meaningful information, we calculated the **Center of Pressure (COP)** using the four sensor values. The COP provides a single metric that reflects the overall pressure distribution, offering insights into the user's approximate position on the toilet seat. Individual sensor values alone lack this context. Additionally, we incorporated data points from 1-2 seconds prior to the current timestamp to capture transitional states, such as changes in weight or center of pressure (COP), over time.

The COP was calculated using the following formulas:

- **COP X-Axis:**

$$\text{COP}_x = (\text{Sensor2} + \text{Sensor4} - \text{Sensor1} + \text{Sensor3}) / \text{sum of sensor values}$$

- **COP Y-Axis:**

$$\text{COP}_y = (\text{Sensor2} - \text{Sensor4} - \text{Sensor1} + \text{Sensor3}) / \text{sum of sensor values}$$

By calculating COP, we can:

- Monitor balance during activities.
- Detect irregularities in pressure distribution due to external factors or physical conditions.
- Provide meaningful insights for feedback or training systems.

## Data Filtering

To ensure accuracy, data points with COP values outside the range  $([-2, 2])$  were removed, as these suggest that the user is not near the toilet seat. In our grid, the origin represents the center of the toilet seat so points beyond -2,2 mean the user is not neat the seat. This filters out unnecessary datapoints.

ML Models/Algorithms



## Random Forest

The first model we decided to implement was the Random Forest classifier, using scikit-learn's RandomForestClassifier method. Random Forest was chosen for this study to its ability to handle non-linear relationships inherent to movement data which was necessary for our dataset that analyzed complex data like the center of pressure (COP) values and time differences. Additionally, the RF model is able to handle imbalanced datasets when compared to other classification based models; this was necessary for our dataset because we had a lot more data points for sitting and not on toilet states in comparison to off-boarding and onboarding states. Lastly, this model allowed us to adjust class weights to avoid overrepresenting the more frequent classes relative to the lesser ones.

The three features that were used were Time\_Diff\_Milliseconds (tracking how much time as passed), COP\_X (left to right weight distribution), COP\_Y( front and back weight distribution). The hyperparameters specified for the model include estimators at 200, a random\_state at 42 (arbitrary randomness metric), and class\_weights to have balanced results(higher importance for certain movements).

While Training the model, we used 80% of the data for training, while the rest was used as testing data. We also made sure to adjust class weights if needed, as well as cross valdiate to make sure the model performed well.

## Support Vector Machine (SVM)

The second machine learning model we implemented was the Support Vector Machine (SVM). The SVM model was chosen for its ability to handle non-linear data and generalize well in high-dimensional spaces. SVM can effectively separate classes, and its performance with limited data makes it suitable for applications where overfitting needs to be controlled. It is particularly useful for datasets where distinguishing between classes requires finding a decision boundary that maximizes the margin between them.

We used a radial basis function (RBF) kernel, a common kernel function for SVM. This kernel can handle non-linearly separable data by projecting it into a higher-dimensional space where it can be separated using a hyperplane. The RBF function computes the dot product and squared distances between features in the dataset, and then performs classification using linear SVM. Similar to Random Forest, class weights can be adjusted to handle class imbalances.

Another step we implemented was cross-validation, where the dataset was divided into equal splits. For each split, the model was trained on some portions of the data and tested on the remaining split. This helps reduce overfitting and ensures reliable performance. Cross-validation is particularly important for SVM models, as they are sensitive to hyperparameters, and this approach helps evaluate performance in a more robust way.

## XGBoost

The last model we used was XGBoost, which is known for being effective in both regression and classification tasks. The model works by building trees sequentially, with each tree improving on the previous one. The importance of each feature is understood over time as the model identifies which features contribute the most.

XGBoost was our third chosen model because it is highly effective with structured datasets and can mitigate overfitting. We believed that the repetitive process of capturing relationships within the data over each tree (and improving each tree sequentially) made XGBoost a better model than the logistic regression model we originally implemented. Logistic regression assumes a linear relationship in the data, which makes it difficult to recognize non-linear patterns. On the other hand, XGBoost can learn complex relationships between features and handle these non-linear patterns more effectively.

There are several parameters used to build these trees in sequence, including the number of estimators, learning rate, max depth, and weights. The number of estimators determines the number of trees the model will build, which increases the model's ability to learn complex patterns but also opens up the risk of overfitting. The learning rate indicates the size of the update with each sequential tree—lower learning rates result in slower convergence. The max depth parameter limits the depth of each tree, which affects the number of splits in a tree. Deeper trees tend to have more splits but also have a higher chance of overfitting. It is important to note that trees with low depth may lead to underfitting, as they may not be complex enough to capture important patterns.

Results: Final Report



## Visualizations

The confusion matrix is one way we can visualize the results of the model depicting the relationship between the predicted label and the true label of the classification. The values in the matrix that provide the most value to us are along the diagonal. A strong diagonal pattern in the confusion matrix indicates good classification, which is generally true in this case. However, the biggest thing to note is the difference between the values for the stable states and the values for the transition states. The sitting and not sitting states had values of 7509 and 13721, respectively on the diagonal signaling high correlation, but the onboarding and offboarding values were just 1333 and 975 which is far worse relatively indicative of poor correlation there. Overall, the confusion matrix was effective at showing whether or not a participant was sitting on the toilet seat, with some confusion regarding whether or not the participant is in a transition state.

## Analysis

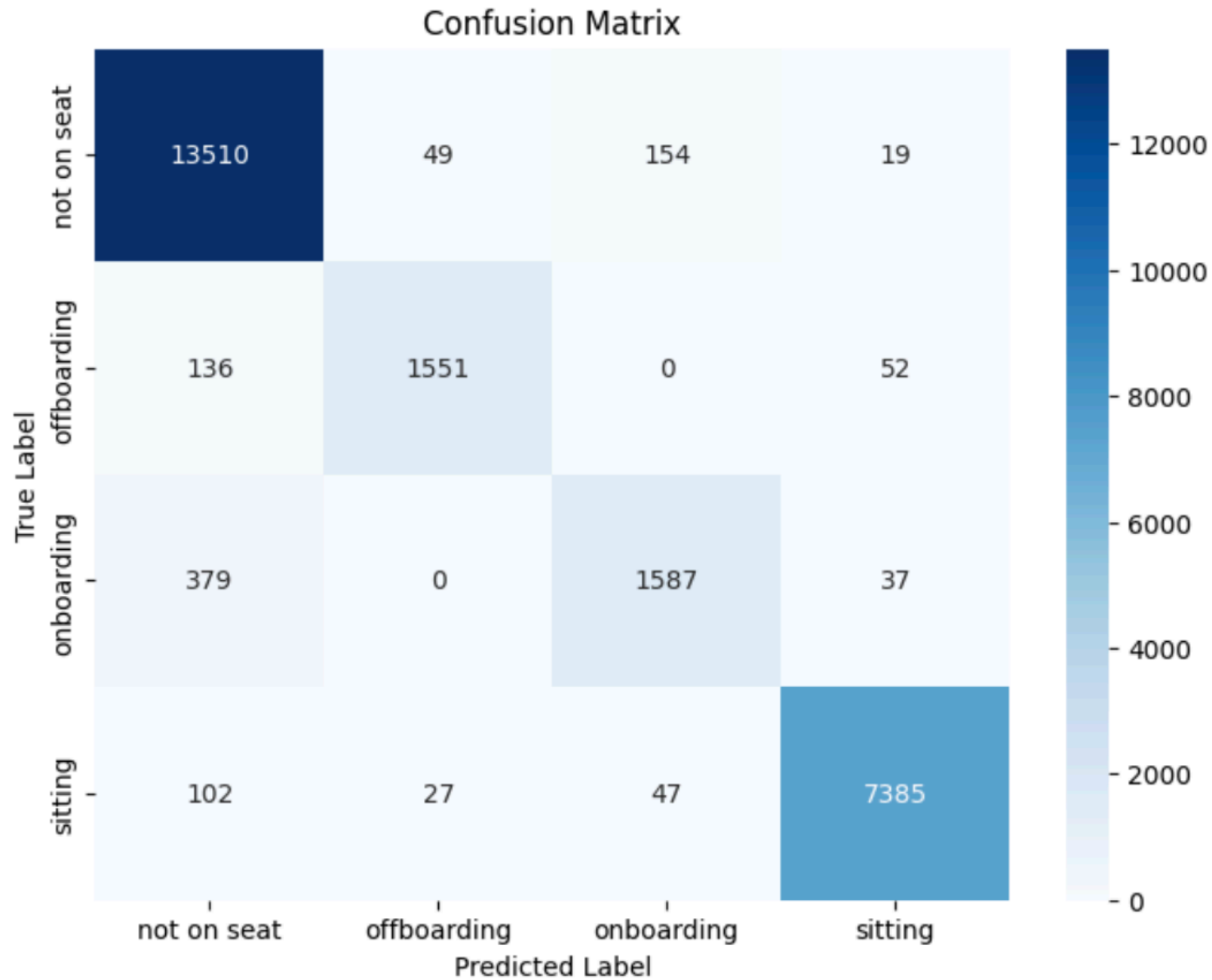
### Random Forest

Accuracy: 96.00%

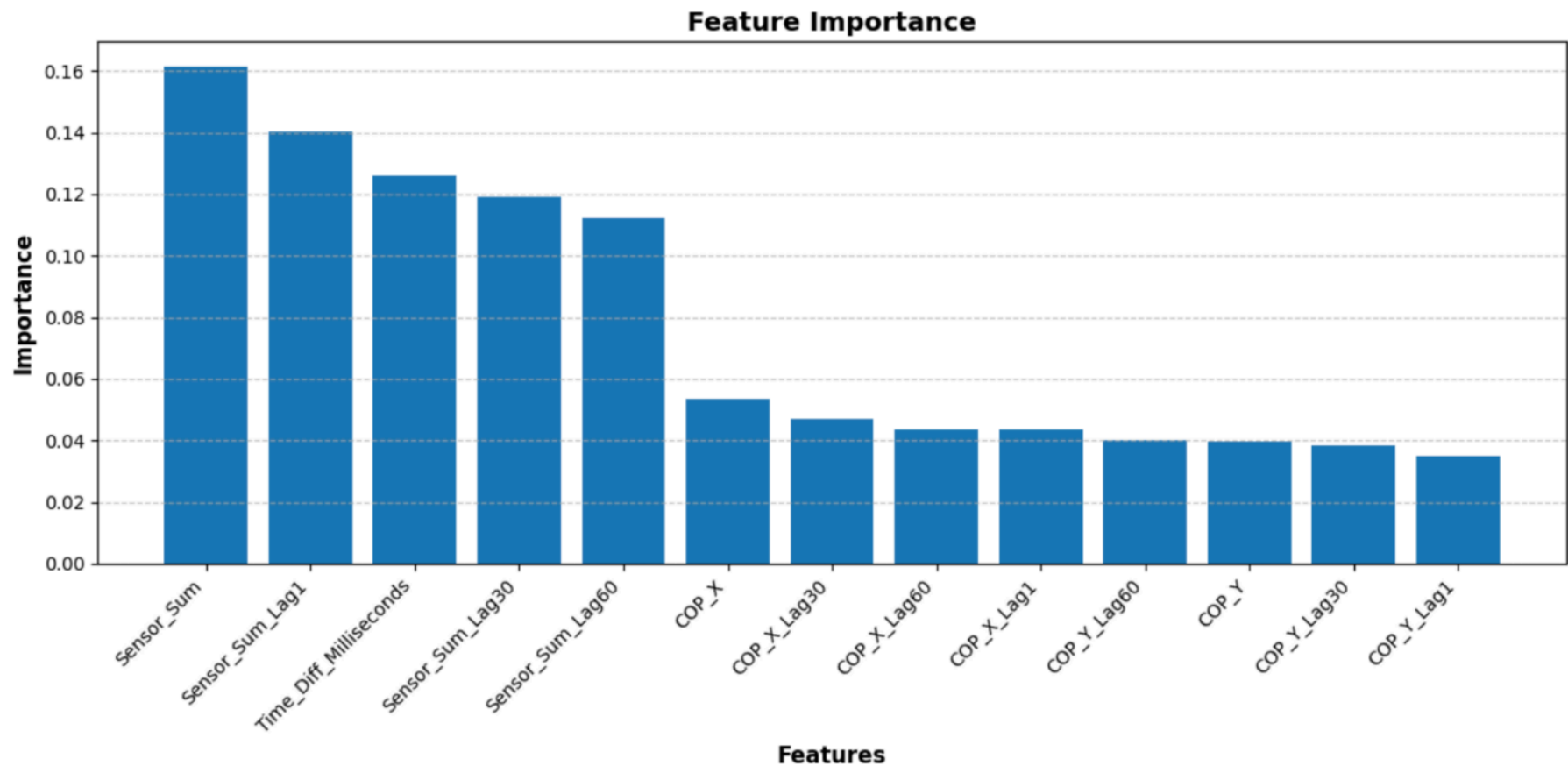
Classification Report:

	precision	recall	f1-score	support
not on seat	0.96	0.98	0.97	13732
offboarding	0.95	0.89	0.92	1739
onboarding	0.89	0.79	0.84	2003
sitting	0.99	0.98	0.98	7561
accuracy			0.96	25035
macro avg	0.95	0.91	0.93	25035
weighted avg	0.96	0.96	0.96	25035

Random Forest Metrics



RF Confusion Matrix



RF Feature Importance

After implementing different class weights as well as lag features to represent measurement over time, the results of the random forest model improved drastically. The Random Forest model achieved an overall accuracy of 96.00%, showing consistent metrics across all classes. The "not on seat" and "sitting" states showed promising results, with F1-scores of 0.97 and 0.98 and precision values of 0.96 and 0.99. Recall rates of 0.98 each show that the model reliably classifies these dominant behaviors with minimal misclassifications. Offboarding and onboarding showed good performance but with slightly lower recall rates, at 0.89 and 0.79, indicating that the model occasionally struggles to identify all instances of these behaviors. For "offboarding," the F1-score of 0.92 and precision of 0.95 highlight the model's ability to make accurate predictions, though misclassifications into "not on seat" and "sitting" remain noticeable. Onboarding showed a significant number of instances misclassified as "not on seat," suggesting overlapping features or insufficient differentiation between these classes. The confusion matrix shows these observations, showing that while the model performs exceptionally well for the non-transition states, it has some difficulty distinguishing transition instances. The model was enhanced through class weight adjustments and new features related to lag pressure measurements. Hyperparameter tuning, such as optimizing `max_depth` or `min_samples_split`, could also be explored. With these improvements, the Random Forest model demonstrates excellent overall performance, with high precision and recall across most metrics, making it a good choice for this classification task.

The feature importance visualization allows us to see which features had the highest importance, and which features were more effective in predicting the state of a participant. The feature importance chart shows that the most important feature in the model is `sensor_sum`, with the highest importance score of approximately 0.16. This indicates that the sum of sensor values

is the most influential feature in driving the model's predictions. Following closely are sensor\_sum\_lag1 and time\_diff\_milliseconds, which also exhibit high importance, making them significant contributors to the model's performance. Moderately important features include COP\_X and COP\_X\_lag30. We can see that the table allows us to see that sensor\_sum was the best at encapsulating the force sensor data in the model. Since certain features can be considered more important, it allows us to see which data changes in sequential movement over time were key in determining different states of the participant.

## SVM

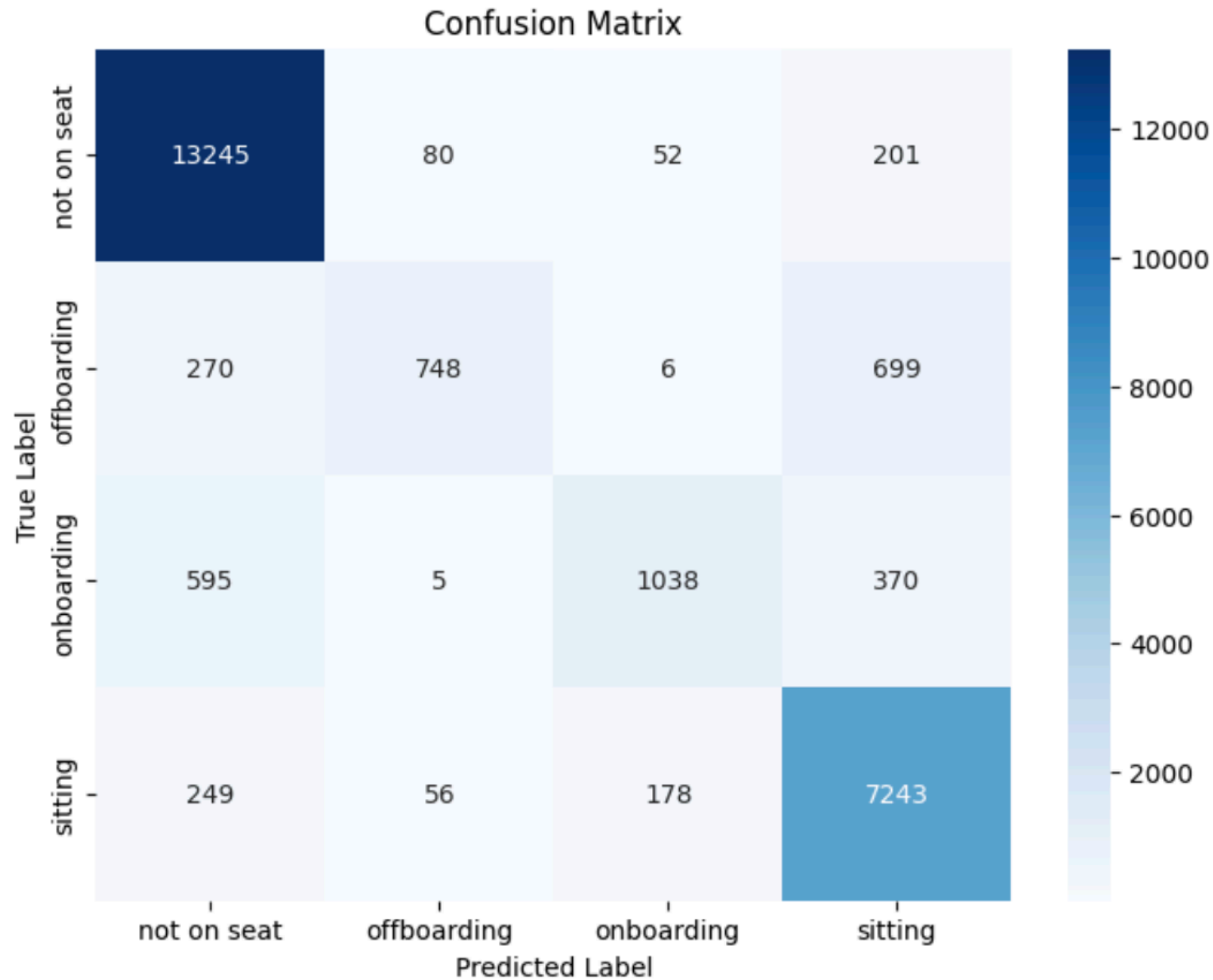
```
Accuracy: 88.97%
Classification Report:
              precision    recall  f1-score   support

not on seat      0.92      0.98      0.95     13578
offboarding      0.84      0.43      0.57      1723
onboarding       0.81      0.52      0.63      2008
sitting          0.85      0.94      0.89      7726

 accuracy              0.89      25035
 macro avg           0.86      0.72      0.76      25035
weighted avg           0.89      0.89      0.88      25035
```

SVM Metrics





SVM Confusion Matrix

The SVM model performed average overall, correctly identifying different behaviors about 89% of the time. It was particularly good at spotting when no one was on the seat - it caught these cases 98% of the time and was right 92% of the time when it made this prediction. The model was also quite good at detecting when someone was sitting, correctly identifying this state 94% of the time. However, the model struggled with the transition states "offboarding" and "onboarding," with F1-scores of 0.57 and 0.63, due to lower recall rates (0.43 for "offboarding" and 0.52 for "onboarding"). These lower recall values indicate that the model often doesn't use these classes correctly. This is probably due to the fact the model had more

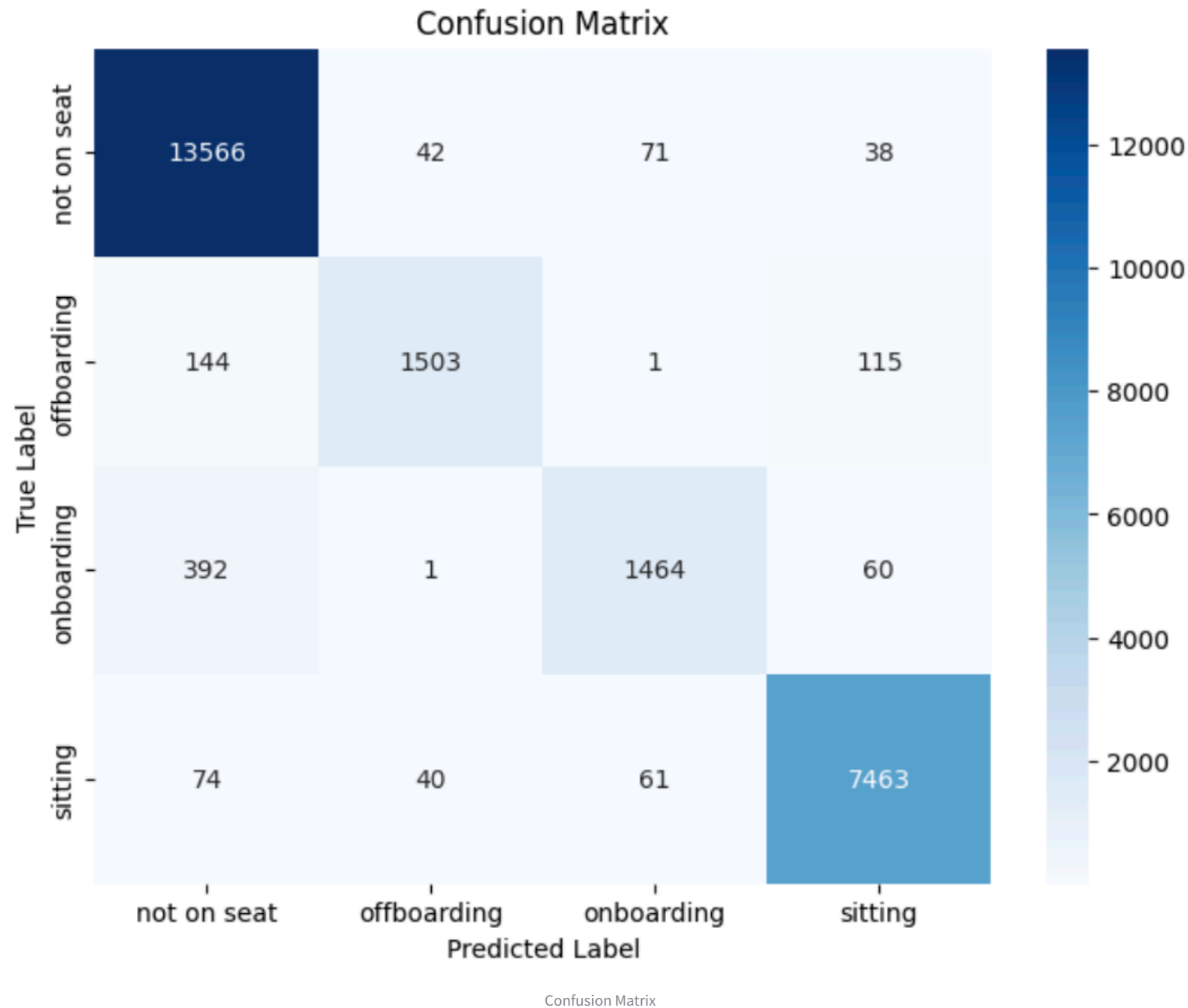
data on instances of sitting and not sitting compared to the number of instances of person onboarding and offboarding the seat. In the future, we can improve this score by adding more onboarding and offboarding data to the training set. The confusion matrix helps show these challenges, as there are misclassifications of "offboarding" as "sitting" and "onboarding" as "not on seat."

When looking at the confusion matrix, we see that most "non on seat" instances were correctly classified (13,245), with minimal error in placing them in other categories. On the other hand the "offboarding" state showed instances being misclassified as "sitting," while some are classified as "not on seat" (270). This suggests overlap in features with these classes and can lead to some error. These findings indicate that while the model performs well on non-transition states, it struggles with transition ones, leading to reduced sensitivity for these behaviors. Performance could be improved by adjusting weights or refining features to better distinguish overlapping behaviors. Despite these challenges, the model's overall robustness is evident, with strong precision and recall for dominant behaviors and potential for improvement in transition classes.

## XGB

Accuracy: 95.85%				
Classification Report:				
	precision	recall	f1-score	support
not on seat	0.96	0.99	0.97	13717
offboarding	0.95	0.85	0.90	1763
onboarding	0.92	0.76	0.83	1917
sitting	0.97	0.98	0.97	7638
accuracy			0.96	25035
macro avg	0.95	0.90	0.92	25035
weighted avg	0.96	0.96	0.96	25035

XGB Metrics



The XGBoost model performed better, achieving an overall accuracy of 95.85%. This was more accurate compared to the SVM model's performance. The model was consistent across all states. The "not on seat" and "sitting" states showed good results, with F1-scores of 0.97 each, precision values of 0.96 and 0.97, and recall rates of 0.99 and 0.98. This exhibits the model's ability to classify these non-transition states with precision. The transition states, "offboarding" and "onboarding," also showed substantial improvements compared to the SVM model,

with F1-scores of 0.90 and 0.83. However, their recall rates, while improved, were lower at 0.85 for offboarding and 0.76 for onboarding, meaning there was some error in correctly identifying every instance. Similar to previous discussion in SVM, this is probably due to the fact that the model has less training in identifying onboarding and offboarding versus sitting and not sitting values. These transitions are also less clear than the other states. The confusion matrix highlighted that the majority of misclassifications for these transition states involved confusion with "not on seat," which may suggest overlapping features. Despite these challenges, the precision for these states was high, at 0.95 for "offboarding" and 0.92 for "onboarding," demonstrating the model's reliability in predictions. Overall, XGBoost was good with accuracy and recall, outperforming the SVM model across these metrics (even in the transition states). To improve its results, we could look into more hyperparameter tuning to reduce misclassification.

## Comparison of Models and Next Steps

After building all three models it is clear the random forest model was the best, closely followed by XGBoost model, and finally SVM. However, in terms of precision and f1-score for the transitional phases(onboard and offboarding), the XGBoost model was the best. Compared to sitting and not sitting classes, the accuracy in identifying the transitional states was low. This is why in our next steps we aim to add more data to capture the onboard/offboard period and maybe include users onboarding and offboarding toilet seat multiple times to increase transitional classes. In addition to that, we want to explore other advanced machine learning models, such as LSTMs or CNNs. They could provide better insights into sequential/spatial data patterns. Another area of improvement is real-time implementation, which can let us build a system that deploys the model in a live environment to classify behaviors in a non static way.

Gantt Chart: Final Report

[View Gantt Chart in Google Sheets](#)

Contribution Table: Final Report

	Name	Contribution
0	Saloni Jain	M1 Data Cleaning, M2 Feature Reduction, M3 Data Visualization
1	Sneha Pal	M1 Coding, M2 Data Visualization, M3 Coding
2	Sooriya Senthilkumar	M1 Feature Reduction, M2 Coding, M3 Results
3	Aneesh Sabarad	M1 Results Evaluation, M2 Results, M3 Data Cleaning
4	Vibha Guru	M1 Data Visualization, M2 Data Cleaning, M3 Feature Reduction

Proposal Video

[View Video on Youtube](#)

Final Video

[View Video on Youtube](#)

## Citations



- [1] B. Jones et al., "Smart Bathroom: Developing a Smart Environment to Study Bathroom Transfers," 2017, [https://www.resna.org/sites/default/files/conference/2017/pdf\\_versions/outcomes/Jones.pdf](https://www.resna.org/sites/default/files/conference/2017/pdf_versions/outcomes/Jones.pdf)
- [2] J. Pohl et al., "Accuracy of gait and posture classification using movement sensors in individuals with mobility impairment after stroke," *Frontiers in Physiology*, vol. 13, Sep. 2022, doi: <https://doi.org/10.3389/fphys.2022.933987>.
- [3] S. Hermesen, V. Verbiest, Marije Buijs, and E. Wentink, "Perceived Use Cases, Barriers, and Requirements for a Smart Health-Tracking Toilet Seat: Qualitative Focus Group Study," *JMIR human factors*, vol. 10, pp. e44850–e44850, Aug. 2023, doi: <https://doi.org/10.2196/44850>.