CMPT 353 - Project

Anh Truong

August 2025

1 Introduction and Motivation

1.1 Main Question

While many devices today can use multiple advanced sensors to understand a user's environment, this often comes at the cost of higher complexity, energy consumption, and data processing requirements. If similar classification performance could be achieved using fewer sensors, devices could be made more efficient, affordable, and easier to integrate into broader safety systems. This context leads me to the main question of the this project: "**Can a dual-sensor approach, reliably tell the difference between indoor and outdoor environments?**"

1.2 Motivation

There is a growing opportunity to develop context-aware systems that can recognize a user's surroundings thanks to the development of new and sophisticated electronic sensors. In addition to being useful tools for researchers, such systems can improve their capacity to offer beneficial direction in potentially hazardous outdoor situations. Environmental scientists could use environmental classification data to correlate noise levels or air quality with particular indoor or outdoor contexts, while urban planners could use it to study pedestrian movement patterns and healthcare researchers could use it to track patient activity levels in various settings.

I only concentrate on using just two sensors at this point in the project for two reasons. First, it makes the system straightforward and simpler to test, allowing one to observe clearly how each sensor helps determine whether a person is indoors or outdoors. Second, low-cost and low-power solutions are more feasible in real-world deployment, particularly for wearable or mobile devices. By demonstrating that accurate classification can be achieved with only a dual-sensor configuration, we set the stage for future sensor additions without needlessly complicating the system from the beginning.

The "Physical Toolbox" sensor suite, which consists of several environmental sensors, served as the starting point for this project. After trying it out, I discovered some intriguing variations in the light meter readings between walking in open spaces on campus and being outside, like beneath a tree. This observation made me wonder if it would still be possible to distinguish between indoor and outdoor environments using fewer sensor combinations.

2 Data and Methodology

2.1 Data Collection

I collected three kinds of data on 2 different days, August 4th and August 5th, at noon, in the afternoon, and at night, for a total of nearly 6 hours of audio recording and saved them as separate files:

2.1.1 Phone sensor logs (.csv):

Recorded with the *Physical Toolbox Suite* app on (Android / Google Play). I enabled all motion and environmental sensors: accelerometer (ax, ay, az), gyroscope (wx, wy, wz), gravity (gFx, gFy, gFz), magnetometer (Bx, By, Bz), barometer (p), and light sensor (I). The app writes timestamped CSVs.

2.1.2 Audio recordings files (.wav):

The app's built-in "sound meter" only reports valid values while it's in the foreground; as soon as I switch apps, it drops to $-\infty$ dB. To avoid that, I recorded audio separately using the headset microphone (Sony WH-CH720N). Using the headset mic reduces breath/proximity artifacts compared to the phone's built-in mic and keeps recording stable while I was using other apps. Due to privacy concerns, I cannot share the raw audio recordings in the folder I submitted but can only share the merged files containing the feature RMS for audio.

2.1.3 Ground-truth labels files (.txt):

I manually wrote timestamps with "in/out" in the phone's Notes app as I moved. I define inside as "inside a building," which includes covered/underground transit stations (e.g., SkyTrain). When the train arrived at a station, I labeled that as in; when the train was running outdoors between stations, that was out and of course, the timestamp along with those.

2.1.4 Why some sensors weren't used?

GPS: I excluded GPS because (a) it can make the task trivial (satellite lock often implies outdoors and I can easily lay that on the map and see my location with no effort) and (b) the app was showing an approximation of 11 m error, which is not great in urban canyons. Omitting GPS keeps the problem focused on on-device, low-power, privacy-friendlier signals.

2.1.5 Timing & synchronization

Each sensor CSV has its own timestamps from the app. I turned on the clock when I was noting down the "in"/ "out" status. There might be some errors due to manual recordings here.
Each .wav file encodes its own start/end time in the filename, and I align per-second RMS to those times.
The label TXT files contain human-entered timestamps. During preprocessing, I align labels to sensor/audio by nearest time-of-day with a small tolerance window.

2.1.6 Environments covered

Data obtained from walking outdoors, riding the SkyTrain (outdoors between stations, indoors at stations), and being inside campus buildings, at the mall, inside my friend's house and at different time of the day. This mix intentionally stresses edge cases (e.g., covered areas with daylight leakage) so we can see whether non-GPS sensors still separate in vs. out.

Since one is in audio file, I cannot tell the exact number of data points. However, after all necessary filterings, the total of data points for training and validation is 10,348 data points.

2.2 Data Filtering

2.2.1 Expectations before Filterings

The barometer is generally stable indoors and outdoors as it should be, but occasionally shows spikes due to app reporting errors. The barometer measurement for what I know in Physics class, cannot exceed 1200 hPa, or fall below 700 or 800. This can provide an early indication of indoor/outdoor transitions: lower pressure outdoors vs. indoors (slightly), although the difference is small, something like a few hPa.

The light meter gives me clear difference between open daylight and covered areas, though under trees or skylights, readings can be intermediate. Indoors, light is generally lower but varies with lamps and window placement. I actually missed recording the timestamp when I arrived and departed from 29th Avenue Station, but the area was quite sunny and bright, so I did not exclude that from the true labels to do semi-supervised learning and create noise in my data. This could also be a bit noisy since sometimes, I put my phone in pocket.

The accelerometer, gyroscope, g-Force meter in the app (to measure gravity) generally show movement patterns, e.g., walking, standing still, or train vibrations and these are useful for separating stationary indoor vs. moving outdoor sequences, but noisy when the device shifts in the pocket or bag.
Magnetometer is sensitive to surrounding metal and electronics; indoors near large structures shows more variation.
The audio recording measures the background noise varies by environment like low hum indoors, higher wind, traffic outdoors. Can distinguish quiet indoor spaces from louder outdoor areas, but may be affected by talking, train announcements, or headphone proximity noise.

Recordings do not always start or stop at the same time across sensors and audio. I also know something about the file name such as timestamps are in the filenames (e.g., MyRec_MMDD_HHMM.wav) give the end time of the recording, which allows back-calculating per-second timestamps. This is important to align sensor and audio data correctly, especially since manual notes were taken for labeling indoor vs. outdoor events.

For night and day time, from the weather channel app from my phone, the sunset was at 20:45 p.m., so I set the data points with time after that as night and the other are day and did one-hot encoding on that.

2.2.2 Filtering

I decided to filter using a Butterworth filter because it provides a smooth, monotonic frequency response in the passband, effectively removing unwanted noise without causing ripples or signal distortion. Calculating important properties like RMS requires preserving the natural signal structure, especially for my sensor and audio data. For smoothing time series with different assumptions, the Kalman and LOWESS filters, which I mostly studied in class, are better suited (Kalman for recursive state estimation, and LOWESS for non-parametric trend fitting). The Butterworth filter is more appropriate in this case because it offers precise control over frequency ranges like low-pass for slow-varying sensors like accelerometer trends, and band-pass for audio to isolate the audible range while preserving the physical meaning of the signal. The filtering actually provides good results and captured a lot of subtle trends rather than just smoothing out the data. Below is the filtering applied to the audio file and the light meter sensor. The luminance filtering has a really high spike at around thee 1600th index. That could be from the fact that I walked by the window and the sun shone directly on it. Other filtering outputs are also provided in the `analysis_output` folder for you to look at and evaluate since it is not practical to add 39 plots in a 5-page report.

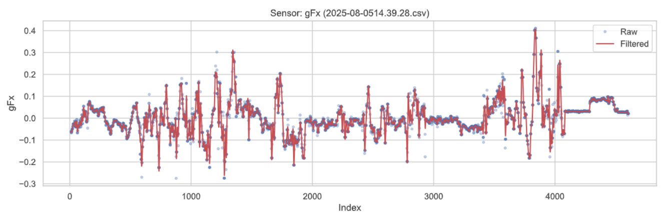


Figure 1: g-Force, x-axis direction, filtering

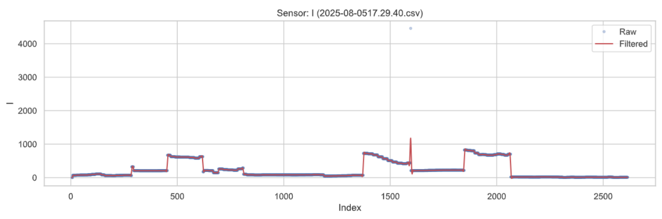


Figure 2: Luminance filtering

After filtering and joining multiple data frames, including the label data frame, and the sensor and audio data frame using the common time, I also realized that some first rows' output are weird since the luminance and output from the barometer were weird, refer to `merged_dataset.csv` file, so I have to filter and drop them out to reduce the outliers.

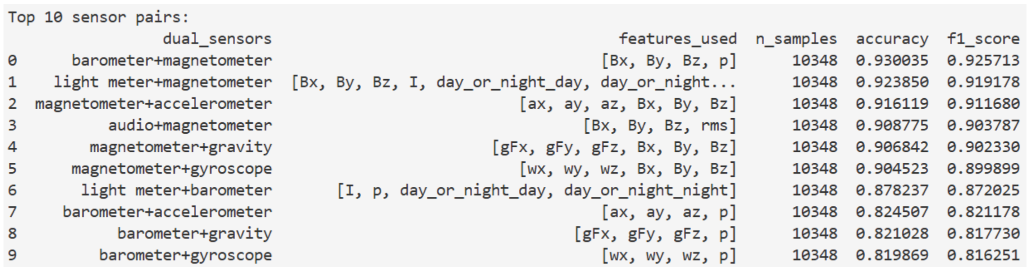
2.3 Training the Model

2.3.1 Process

To classify inside vs. outside, I focused on evaluating dual-sensor combinations to identify the most informative pairs while keeping the model interpretable. I used a Random Forest classifier with 200 trees and a minimum leaf size of 30 to avoid overfitting. The dataset was split into training and test sets with a 75/25 stratified split. I trained the classifier on all candidate features from the chosen sensor pairs and evaluated performance using accuracy, F1-score, and cross-validation. I also noticed that there is a slight imbalance in the data set, 3634 in's and 6714 out's so I set the parameter `class_weight` to be "balanced" so it can handle the weights and predict the minority class. And in this report, I will mostly use the F1-score instead of the accuracy score due to the credibility of the two metrics on the imbalance dataset. At the moment, I do not want to tune the model so that the errors are analyzed in a data-centric style instead.

2.3.2 Result

The result after I train is as below.



```
Top 10 sensor pairs:
                 dual_sensors                                      features_used  n_samples  accuracy  f1_score
0       barometer+magnetometer                                   [Bx, By, Bz, p]      10348  0.930035  0.925713
1     light meter+magnetometer  [Bx, By, Bz, I, day_or_night_day, day_or_night...      10348  0.923850  0.919178
2   magnetometer+accelerometer                          [ax, ay, az, Bx, By, Bz]      10348  0.916119  0.911680
3           audio+magnetometer                                 [Bx, By, Bz, rms]      10348  0.908775  0.903787
4         magnetometer+gravity                        [gFx, gFy, gFz, Bx, By, Bz]      10348  0.906842  0.902330
5       magnetometer+gyroscope                          [wx, wy, wz, Bx, By, Bz]      10348  0.904523  0.899899
6        light meter+barometer       [I, p, day_or_night_day, day_or_night_night]      10348  0.878237  0.872025
7      barometer+accelerometer                                   [ax, ay, az, p]      10348  0.824507  0.821178
8            barometer+gravity                                [gFx, gFy, gFz, p]      10348  0.821028  0.817730
9          barometer+gyroscope                                   [wx, wy, wz, p]      10348  0.819869  0.816251
```

Figure 3: The dual sensors and the corresponding F1-score

One finding from the result is that at least one environment-based sensor in the combination of this top 10 and the best dual sensors are two environment-based sensors rather than motion-based. Another one is the small difference between accuracy and F1 score suggests that the classifier performs consistently across classes, without favoring one class over another. This indicates that the model is robust even for minority classes in the dataset.

2.4 Data Analysis

One question that can answer my curiosity about whether a model that trains on all sensors, i.e., all features, can gives a better prediction and is sacrificing the simplicity in a 2-sensor setup worth it. Therefore, I just did an extra step on training that model and compare the F1-scores of both models. I found that the optimal dual-sensor model—in this case, the one that takes into account characteristics from the magnetometer and barometer had an F1 score that was roughly 1% higher than the all-sensors setup. This suggests that simply adding additional features does not always guarantee improved performance and can be explained possibly by that the additional sensors added redundant or noisy data, which would have limited the model's capacity to generalize. This is in line with the "curse of dimensionality," which states that adding irrelevant predictors might reduce the predictive ability of strong characteristics. Because it better balances informativeness and noise, the dual-sensor setup works better in this case.

The first one is quite trivial. Another question is that is the model making sensible mistakes? This question is inspired by 'Exercise 8' when we are trying to cluster the cities. From the statistical analysis, the p-value associated with p, the barometer measurement, was 0.0103 using a Mann-Whitney U test. This indicates a statistically significant difference in barometric pressure between the two true classes among the misclassified points. In other words, even among errors, the true classes have distinguishable barometric patterns, suggesting that the model's misclassifications are not entirely random.

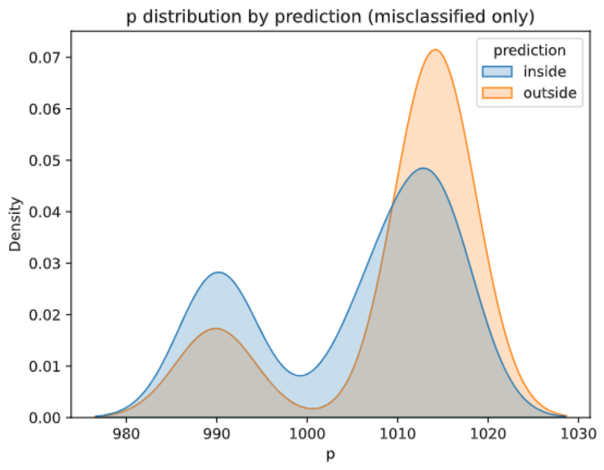


Figure 4

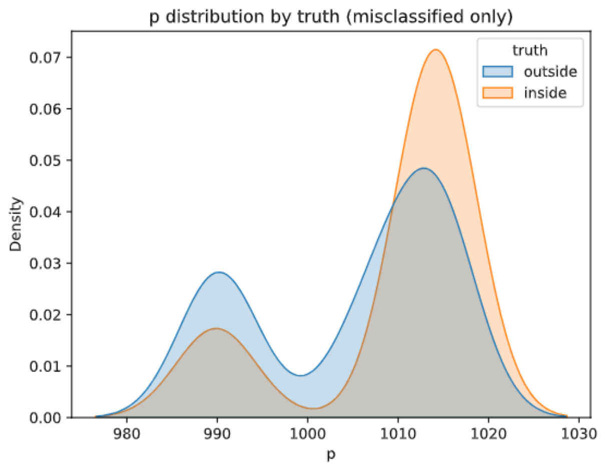


Figure 5

The KDE plot also shows that misclassifications occur primarily in the overlapping regions of the distributions for each class. The first plot shows the distribution grouped by the true labels (truth), while the second plot shows the distribution grouped by the predicted labels (prediction). We notice that the distributions are quite similar in shape, which is expected because these points are misclassified: the predicted label often does not match the truth. In the truth-based KDE, the "inside" points (orange) have a slightly higher peak around 1015 hPa, while "outside" points (blue) have a broader spread around 990–1015 hPa. This separation suggests that pressure (p) has some predictive signal, but misclassifications occur in regions where the distributions overlap.

The similarity between the truth and prediction KDE plots also indicates that the model's errors are not extreme: the model tends to predict labels in regions where the features themselves are ambiguous, rather than making wildly incorrect predictions.

Overall, this analysis demonstrates that the misclassifications are informative: they highlight ranges of feature values where the model is less certain, and they confirm that the barometer (p) is a meaningful predictor even when errors occur. By focusing on misclassified points, we gain targeted insight into the limitations of the model without the noise of correctly classified points dominating the analysis.

3 Limitations and Discussion

One challenge I faced early on was deciding which audio feature to extract. More complex representations like MFCCs or spectral centroid might capture finer details of sound, but they also add computational cost and are sensitive to microphone position. I ultimately chose RMS because it is simple, efficient, and consistent with the per-second resolution of the other sensors, but this decision reflects the broader tradeoff of interpretability versus complexity that runs throughout the project. RMS (Root Mean Square) provides a measure of a signal's average energy, which correlates well with environmental context such as indoor versus outdoor settings. Unlike more complex features like MFCCs or spectral centroid, RMS is robust to irrelevant variations in pitch or timbre, less sensitive to microphone position or headphone proximity, and computationally efficient for long recordings. Its per-second aggregation also aligns naturally with the time resolution of other sensors, making it suitable for merging and classification. This makes RMS particularly relevant to the project, as differences in ambient sound energy can help distinguish contexts like being inside a building versus outside on campus, complementing other sensor readings for environmental detection.

That said, there are clear limitations in what I was able to accomplish. The first one is data coverage. I only collected a few hours of recordings across two days, which is not nearly enough to capture the full range of environments someone might encounter. For example, "indoors" could mean a quiet library, a crowded gym, or a transit tunnel with heavy echoes, all of which are very different acoustically and visually. Similarly, "outdoors" could be a sunny park, a noisy intersection, or a shaded street. My dataset touched on some of these, but not all. This makes it hard to know how well the model would generalize in the wild.

Another limitation is labeling. Because I manually recorded timestamps while moving, there are inevitable errors of a few seconds. In some cases, like entering a train station or walking under a canopy, even a two–three second misalignment could place the label in the wrong environment. That introduces noise that the classifier has to deal with. It's encouraging that the model still performed well despite this, but future work should aim for more precise labeling, either by designing an app that lets me tap directly to log transitions, or by syncing video recordings with sensor data.
From the modeling side, I deliberately kept things simple by focusing on dual-sensor combinations and Random Forests without heavy parameter tuning. This was a conscious choice: I wanted to see how far I could get with minimal complexity. The fact that the best two-sensor pair actually outperformed the all-sensors setup suggests that more sensors are not always better, and that careful selection matters. Still, it's possible that more advanced models (e.g., gradient boosting, temporal neural networks) could squeeze out higher performance, especially if fed more training data.

Finally, the most "human" limitation: I was the only participant. My walking speed, the way I hold or pocket my phone, and even the headphones I used all shape the sensor readings. For a system like this to be robust, it would need to handle many different users, devices, and environments.

Despite these limitations, the project has shown something important: with just two carefully chosen, low-power sensors, we can already get meaningful separation between indoor and outdoor contexts. This suggests that real-world applications, from fitness trackers to safety systems which don't necessarily need an arsenal of sensors to be effective. For me, that is the most exciting takeaway: simplicity can work, and sometimes, less really is more.

Looking forward, the most natural next steps would be to expand the dataset in both scale and diversity — collecting data from multiple users, at different locations, and under more varied conditions (e.g., libraries, gyms, train tunnels, night-time outdoors). With a richer dataset, I could also explore semi-supervised methods to handle noisy labels, and test whether temporal models (such as LSTMs or transformers) capture transitions better than per-second classifiers. Finally, building a lightweight app that synchronizes sensors and labels automatically would make the pipeline more practical and reproducible. Together, these improvements would move the project closer to a deployable system while still holding onto the goal of simplicity and low-power operation.

4 Project Experience Summary

Indoor/Outdoor Classification from Smartphone Sensor & Audio Data

I developed a GPS-free indoor/outdoor detection system by collecting and analyzing smartphone barometer, magnetometer, and audio data; applied filtering, feature extraction (e.g., RMS), and multimodal synchronization to create a clean dataset.

I built and evaluated classification models (Random Forest, PCA analysis), showing that dual-sensor setups can outperform all-sensor combinations, demonstrating critical insights into feature relevance and model performance.

I gained hands-on experience with data cleaning, multimodal data integration, and applied machine learning for real-world signal processing challenges.