Truong An Truong Lam
STAT 410
12/12/2024

Final Project Report

Indoor Positioning System

Summary

This report delves into the pioneering development of an Indoor Positioning System (IPS) that accurately forecasts device locations within a single-floor environment using Wi-Fi signal strength. By employing K-Nearest Neighbors (KNN) Regression and Trilateration with Least Squares Regression, the project uncovers the relationship between signal strength and distance, revealing patterns of signal attenuation and enabling predictive analysis on positioning. In essence, we aim to predict the distance between the device and the router, and then use trilateration to determine the device's location based on the signal strength of the access points.

Introduction

Traditional GPS systems are challenged in indoor environments due to signal obstruction by walls and furniture. This project aims to develop an IPS that leverages Wi-Fi signals and advanced algorithms to predict objects' locations. Specifically, this system is designed for a single-floor setup with six Wi-Fi access points.

Background

An **Indoor Positioning System (IPS)** is a cutting-edge technology used to track the real-time location of objects and people within indoor environments where GPS signals are weak or unavailable. IPS, which typically relies on Wi-Fi/Bluetooth frequency signals, is a crucial tool in various settings such as malls, airports, hospitals, and warehouses. The significant challenge addressed in this project is achieving high accuracy in location prediction despite signal noise and environmental obstacles, underscoring the practical relevance and importance of our work.

Data

Data Collection

The project is meticulously set up in a 15m x 36m floor plan. Testing devices are connected to the network, taken to various locations, and oriented in multiple directions. The testing device generates data by recording the signal strength of all access points, ensuring a comprehensive and robust data collection process.

- **Offline Data (Training):** Contains 166 locations, sampled 880 times per location across eight orientations.

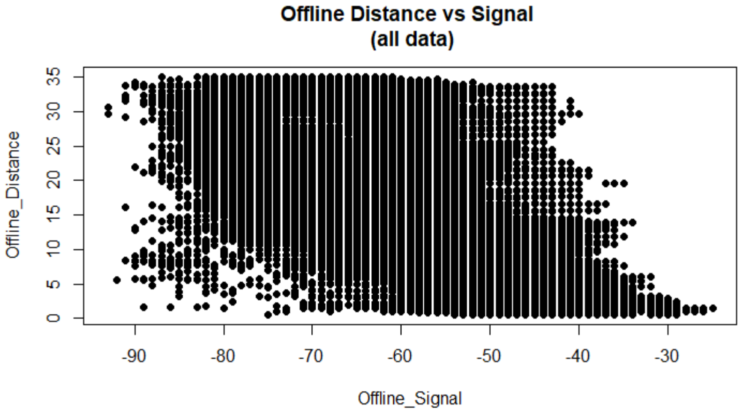
- **Online Data (Testing):** Simulates real-world data with random orientations and a total of 6,600 measurements.

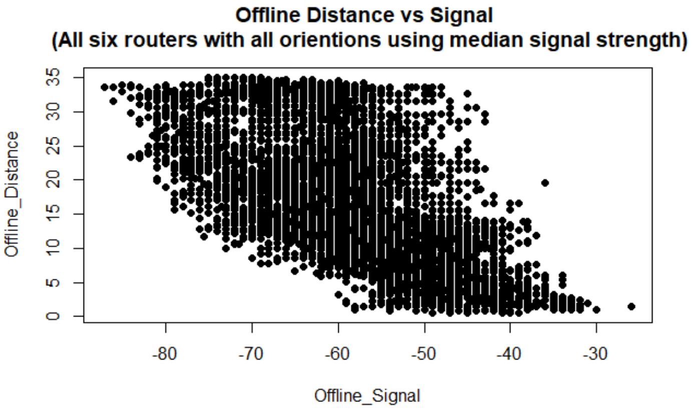
Preprocessing

- Removing observations with missing values
- Converting timestamps to readable formats
- Grouping device orientations into bins : 0°, 45°, 90°, 135°, 180°, 225°, 270°, 315°,
- Remove unnecessary/redundant information
- Filtering access points to retain only relevant data

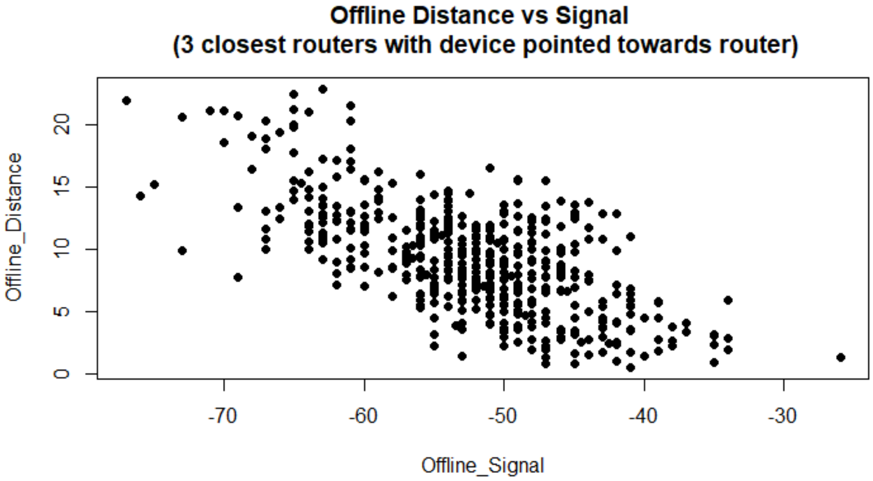
The most influential step in this part is removing noise in the Offline dataset.
We used four different versions of the data to improve the accuracy of our prediction model. The graphs below demonstrate the relationship between signal strength and distance differences when using differently adjusted data. In this case, the variable distance represents the Euclidean distance between the device and the router at the time of the measurement.

Note that there is significant variability in the first graph (all data). We can reduce noise by using the median signal strength from the 110 recordings taken at each position and orientation. For the second graph (using median signal strength), we can observe that the variation increases beyond 10 meters. So, choosing closer access points will yield more accurate predictions.

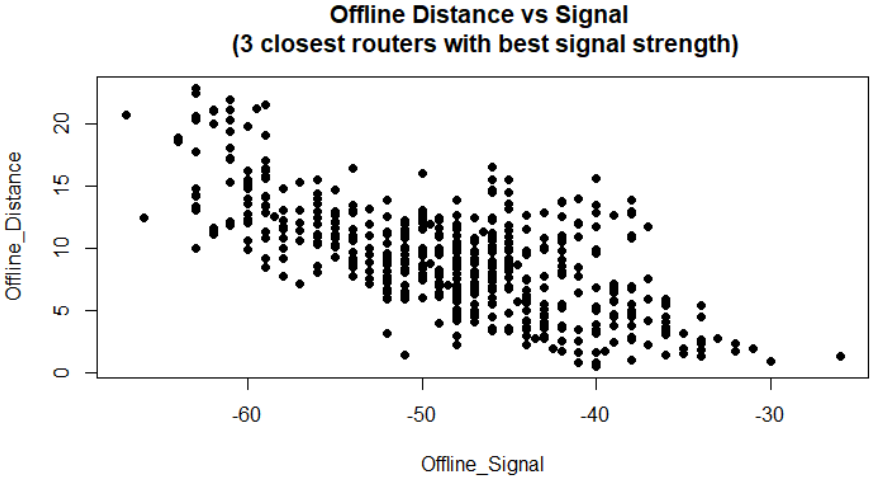




Offline Distance vs Signal
(All six routers with all orientions using median signal strength)



Offline Distance vs Signal
(3 closest routers with device pointed towards router)



Offline Distance vs Signal
(3 closest routers with best signal strength)

Using the three closest access points for each recording, with the device pointed at the router, produces the best result for our model. Further investigation is necessary to determine what orientation or other factors would improve our model.

Methods

Algorithms

K-Nearest Neighbor Regression: A simple, intuitive algorithm for classification and regression tasks. It predicts outcomes based on the similarity of input data points to their closest neighbors in a dataset. In **indoor positioning systems**, KNN Regression estimates the distance between the device and the router based on signal strength. How it works:

- Compute the distance (e.g., Euclidean) between the target and all training observations.
- Identify the **k nearest neighbors** (observations with the smallest distances).
- Predict the distance by averaging the positions of the k neighbors.
- Finding the best K value gives the most accurate prediction.

Trilateration Using Least Squares Regression: A geometric approach to determine an object's position using distances from multiple reference points (e.g., Wi-Fi access points). For training the mode, we used the three closest routers based on distance, and for testing the mode, we used the routers with the top three best signal strengths for location and orientation position. We can approximate the device's location using three reference points in a coordinate system.

Least Squares Regression is used in trilateration when measurements are noisy or inconsistent. It finds the best-fit solution by minimizing the error between observed and predicted distances based on the estimated position.

- Input: Known positions of access points (APs) and distances (calculated from signal strength) to the APs.
- Linearize:
  - Reformulate as a linear system Ax=b, where:
  - A: From the derived system of equations (right side)
  - x: A 1x2 matrix with the x and y coordinate predictions
  - b: From the derived system of equations (left side)
- Solve for Position:
  - Use least squares regression (pseudoinverse or direct matrix solution) to minimize the error between observed and predicted distances.

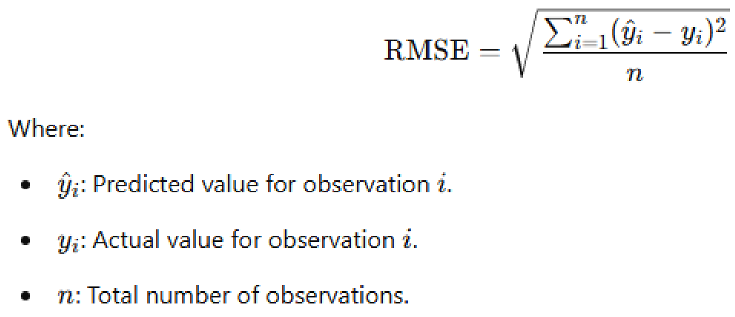
Root Mean Square Error (RMSE):
RMSE is a commonly used metric for evaluating predictive model accuracy. It measures the average error between predicted and actual values and provides a single value

summarizing how well a model performs, with lower RMSE values indicating better accuracy.

How RMSE is Calculated:

1. Compute the differences (errors) between the predicted and actual values.
2. Square each difference to eliminate negative values and emphasize larger errors.
3. Take the average of these squared differences.
4. Calculate the square root of the average to return the error to the original units of measurement.

Formula:



$$\text{RMSE} = \sqrt{\frac{\sum_{i=1}^{n}(\hat{y}_i - y_i)^2}{n}}$$

Where:

- $\hat{y}_i$: Predicted value for observation i.
- y_i: Actual value for observation i.
- n: Total number of observations.

Why RMSE is Useful:

- **Interpretability:** The result is in the same unit as the target variable, making it easy to interpret.
- **Focus on Larger Errors:** Squaring the errors gives more weight to larger errors, making RMSE sensitive to significant prediction mistakes.
- **Model Comparison:** The RMSE metric allows us to compare the performance of different models directly. This comparison is crucial for determining which model performs better for a given data set, and thus, which model is most suitable for our IPS development.

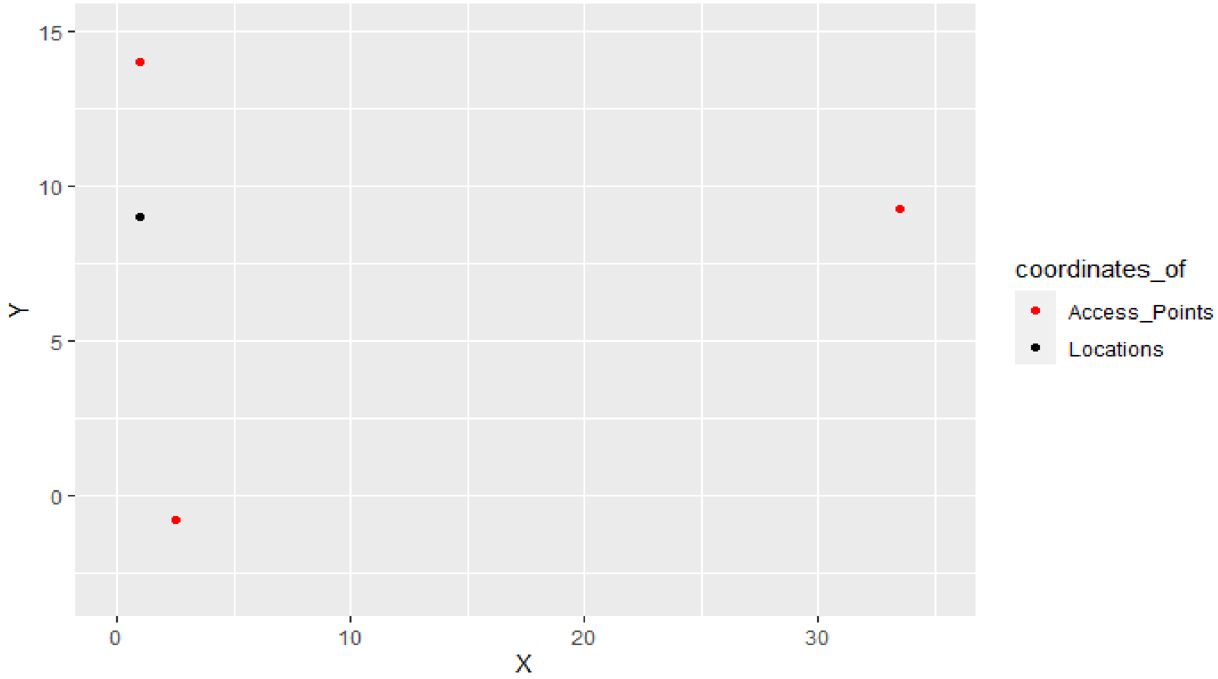
Results

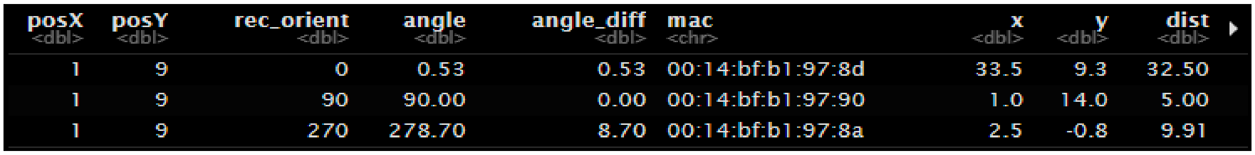
Model Performance: We used four different methods of reducing noise within the data to find the best RMSE. The result is that the data containing the three closest routers when the device is facing the router performed the best with RMSE for the KKN model, which is 3.18 meters, and for the least square method, which resulted in an RMSE of

4.81 meters. It is yet to be determined why this method yielded the best results, so we will need to explore this further.

We took a sample from the data set to explore. Using the sample, we proved that orientation recordings are based on polar coordinates. Also, there is a strong correlation between distance and signal strength (distance = space between the device and router). We also found that orientation influences signal strength. However, further exploration of the orientation's potential role in predicting distance is needed.

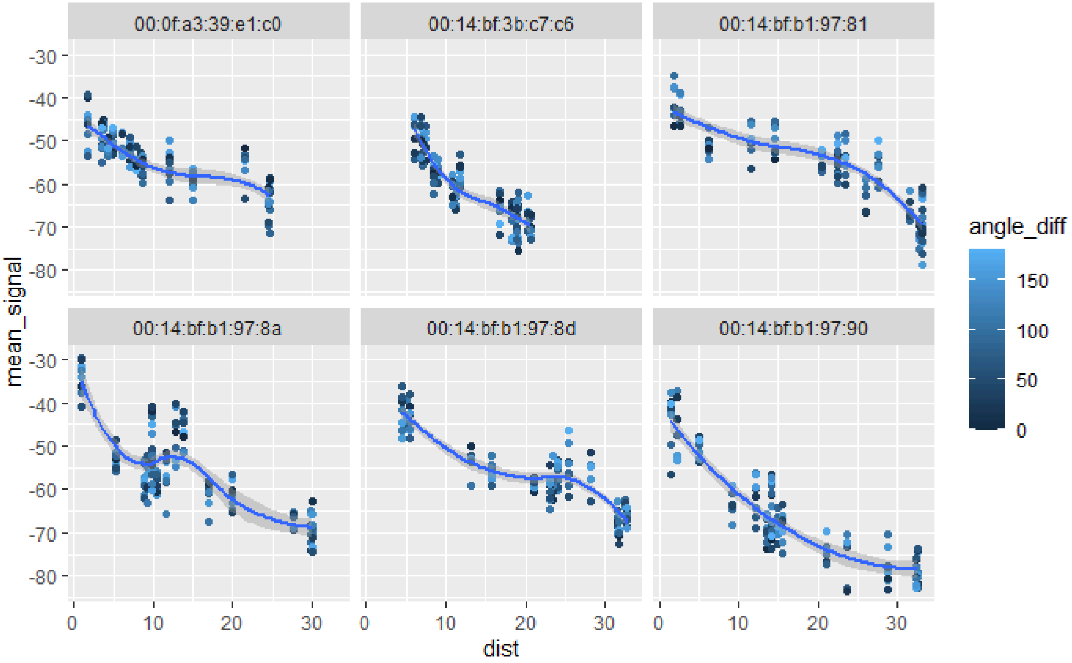
Visualizations:

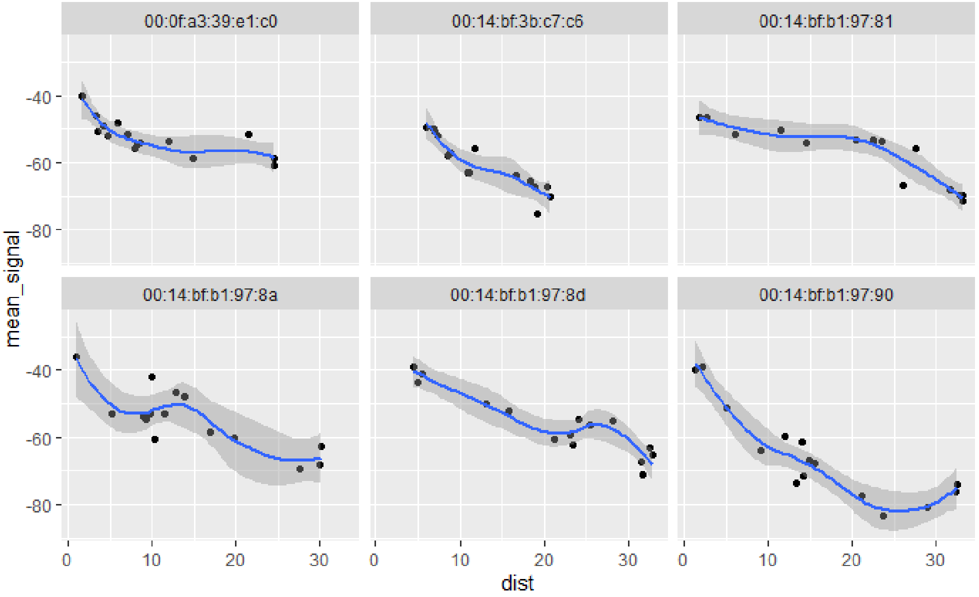




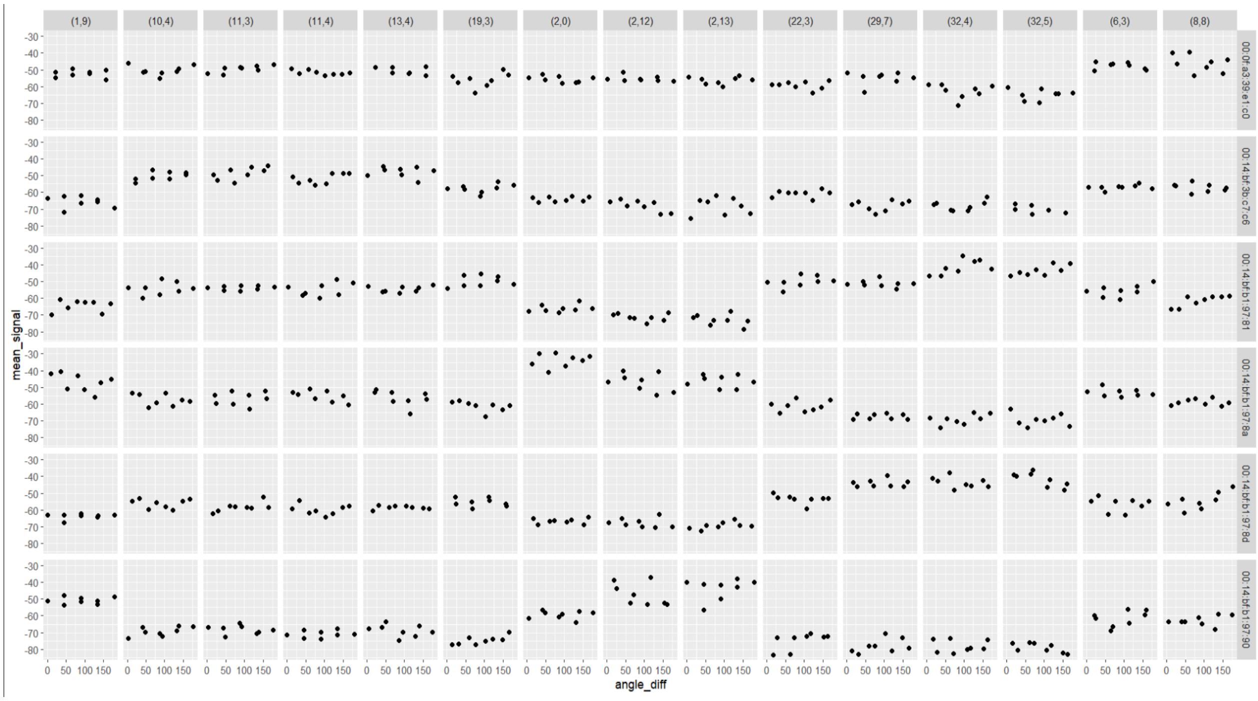
```
 posX  posY   rec_orient   angle   angle_diff  mac                      x      y    dist
<dbl> <dbl>        <dbl>   <dbl>        <dbl>  <chr>                <dbl>  <dbl>   <dbl>
    1     9            0    0.53         0.53  00:14:bf:b1:97:8d     33.5    9.3   32.50
    1     9           90   90.00         0.00  00:14:bf:b1:97:90      1.0   14.0    5.00
    1     9          270  278.70         8.70  00:14:bf:b1:97:8a      2.5   -0.8    9.91
```

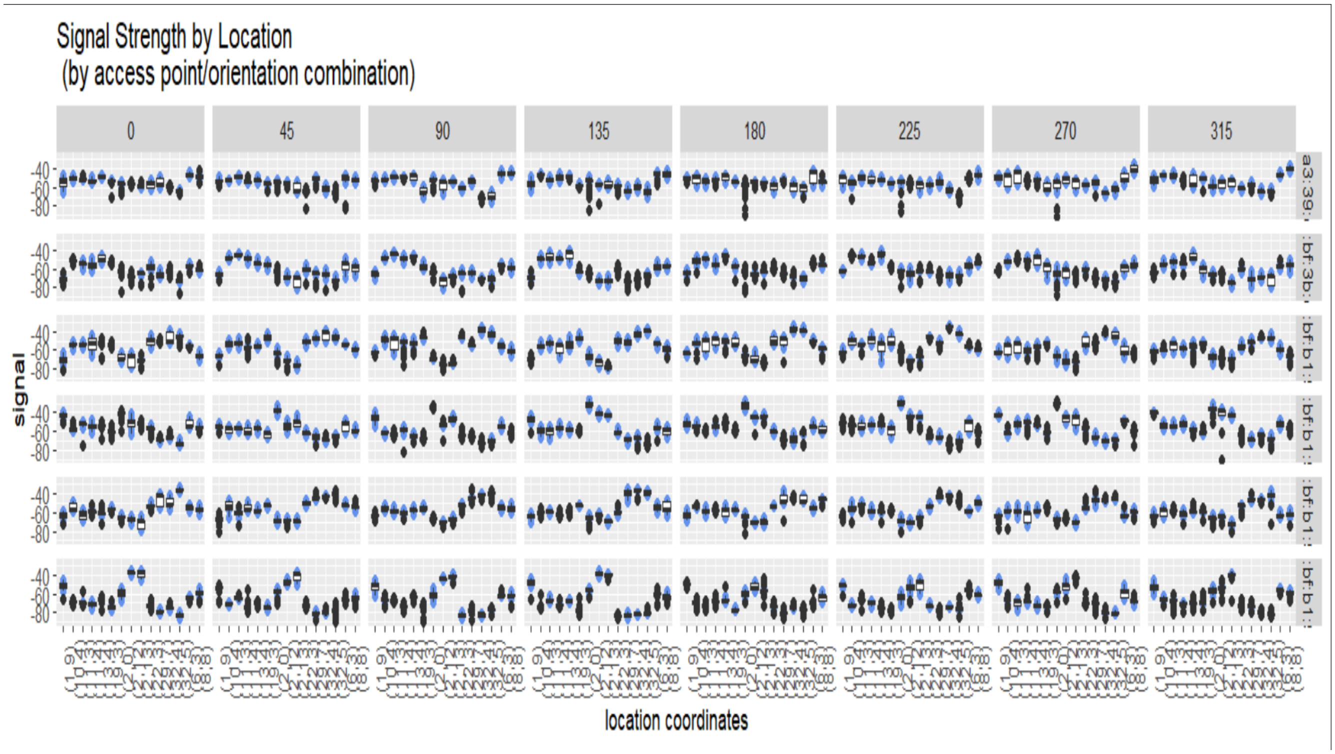
This shows the positional relationships between points on a graph. The variable Angle measures the actual angle from the device to the specified router in polar coordinate space. Rec_orient measures the angle at which the measurement was taken. We can see here that the actual angle between the two points (1,9) and (1,14), respectively, is 90 degrees, which matches the angle that the variable rec_orient recorded. Thus, we have shown that the angle measurement from rec_orientant complies with standard graphical measurements where 0 degrees fall on the X-axis and 90 degrees fall on the Y-axis.





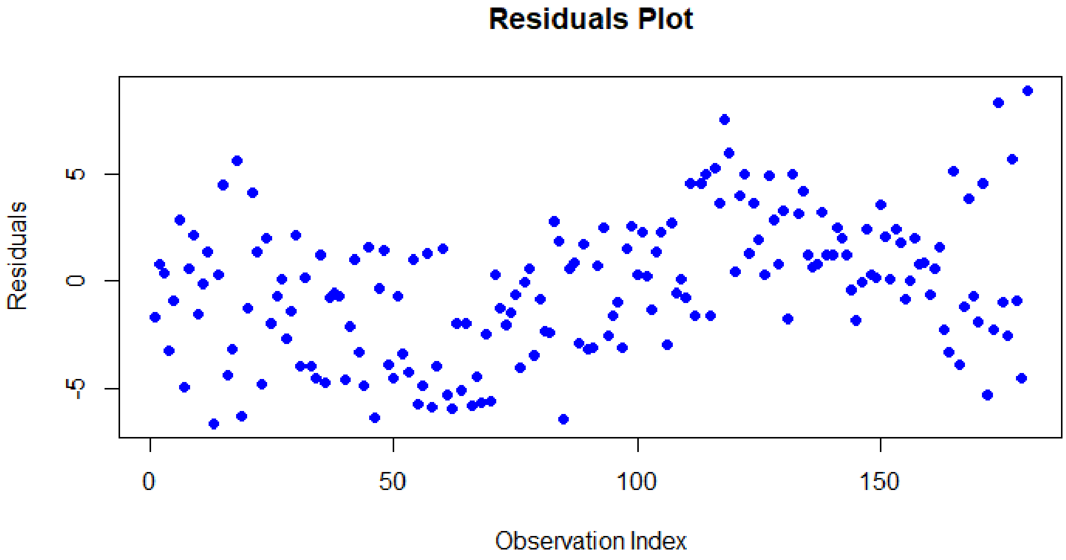
The graph above shows the relationship between distance and mean signal strength (from about 110 recordings). As the distance between the device and the router increases, so does the signal strength. This marked the first significant step in unlocking our predictive modeling capabilities. Note that both graphs look at the relationship between mean signal strength and distance for each access point individually. The top set of graphs is based on the data using all of the mean signal measurements for each orientation, whereas the bottom set of graphs chooses only the measurement where the device is pointing at the router. Note that the above angle_diff shows how far away or towards the device was pointed at the router. We attempted to use graphs with less variability. In this case, measurements were used to determine where the device was pointed at the router.

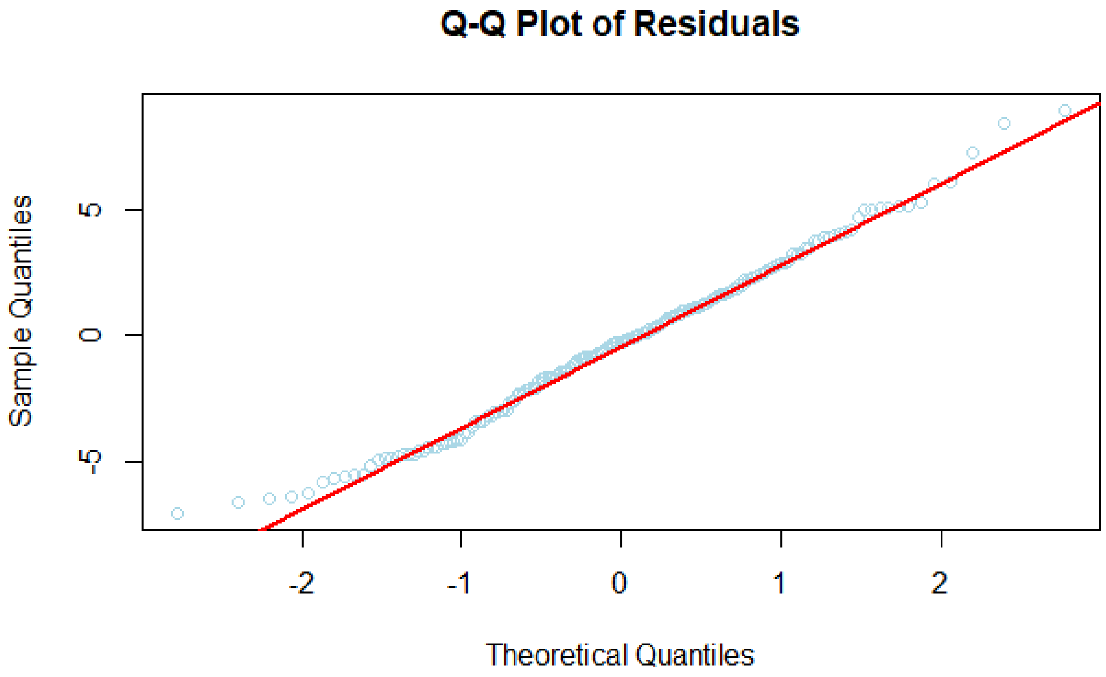




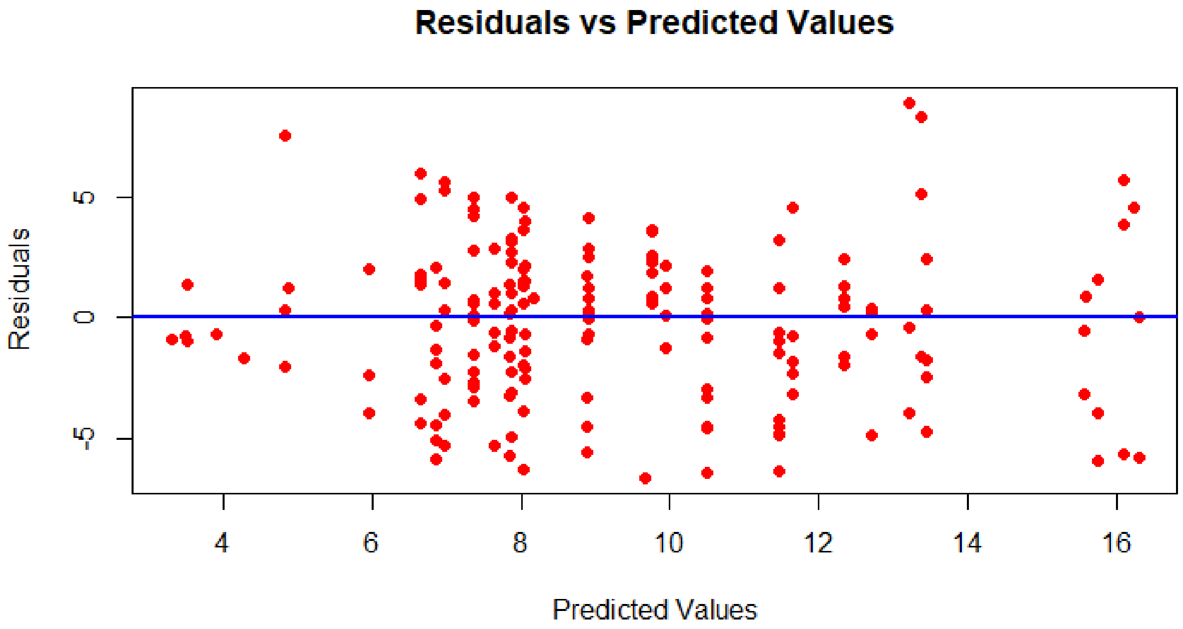
These graphs show the impact that orientation has on signal strength. In other words, how does the way the device was pointed affect the signal recording? While not dramatic, something is going on here that we need to investigate further.

K-Nearest Neighbor Regression with K = 15



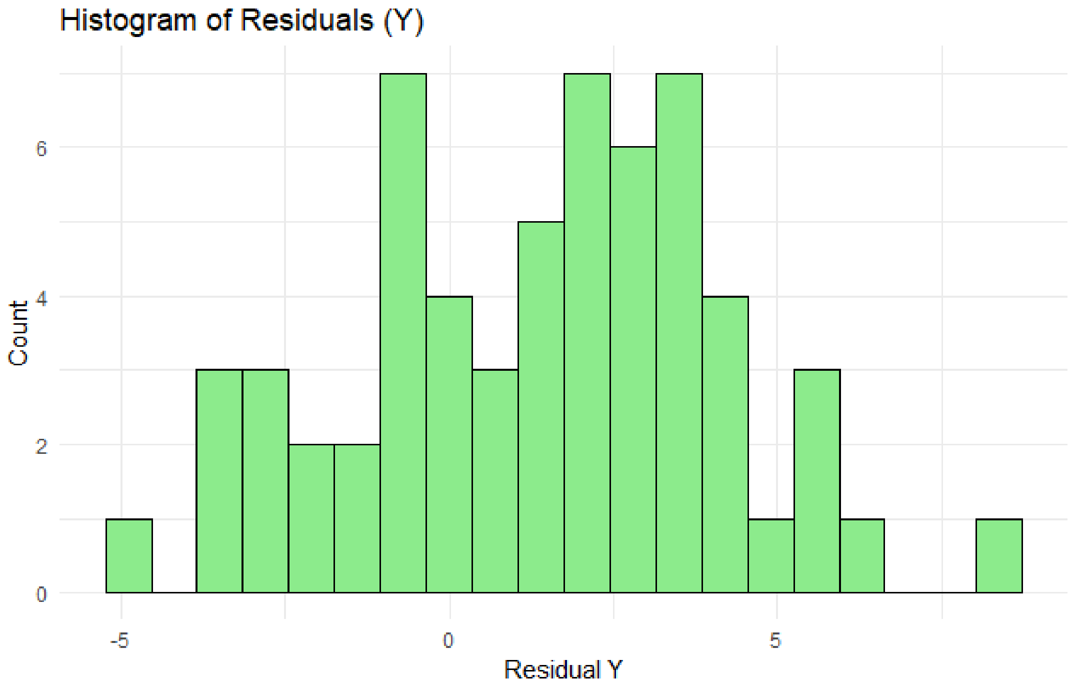

Q-Q Plot of Residuals

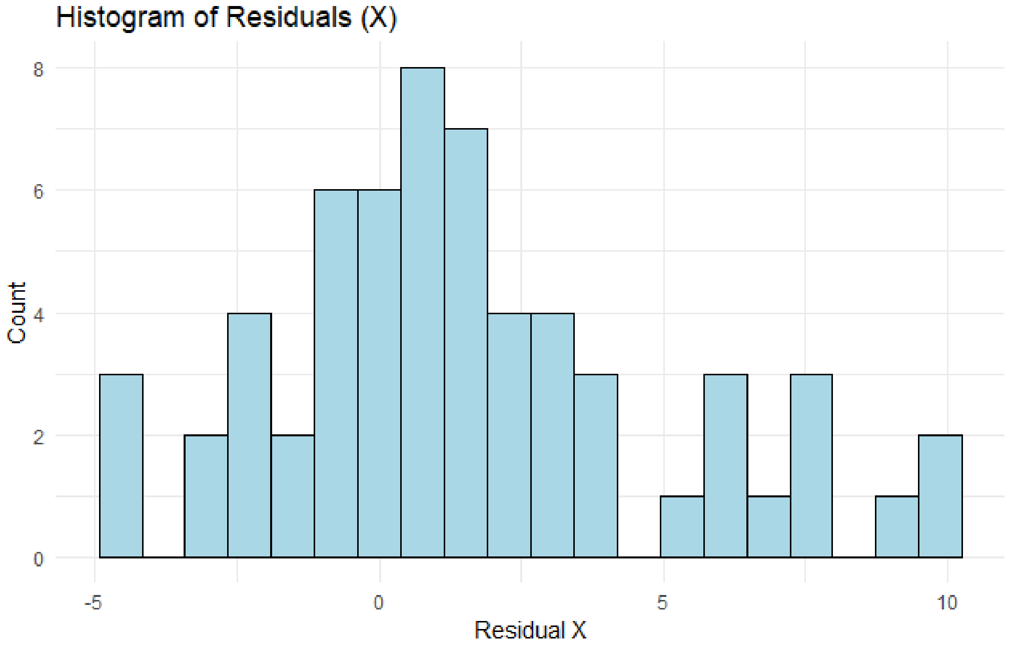
These graphs check the distribution of model residuals for the KNN Regression. In the graph above, there is a point where there is less variability in the residuals. If the observations/recordings were taken at points close to each other, this could mean that the model was more accurate for a specific region in the building. The QQ plot indicates that the model's residuals exhibit a roughly normal distribution, although the tails are slightly skewed. Are there extreme values on each side of the graph?


Residuals vs Predicted Values

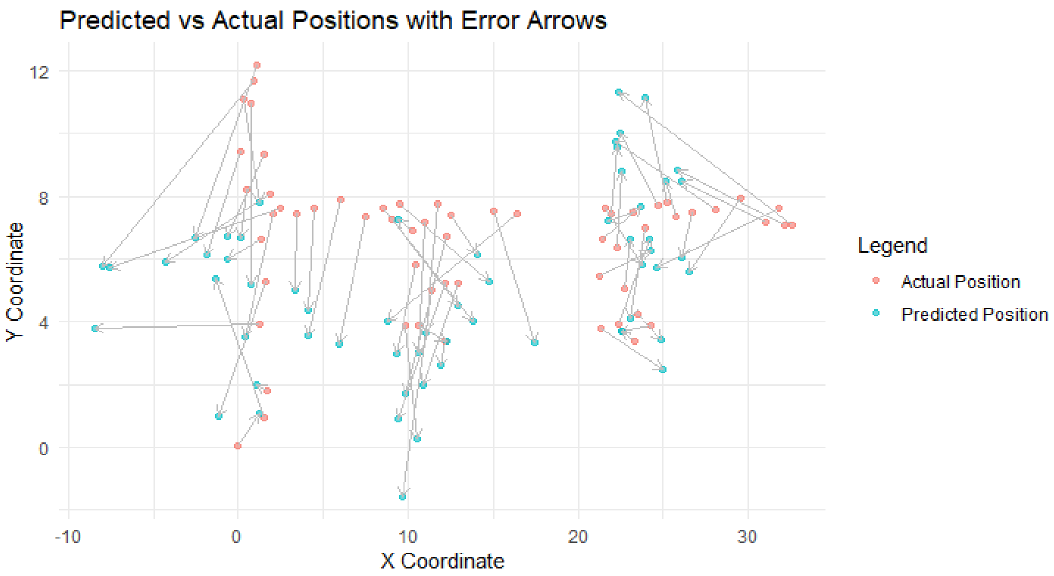
Is there equal variance for the residuals of the model? Mostly, but there are a few gaps. We need to explore this more.

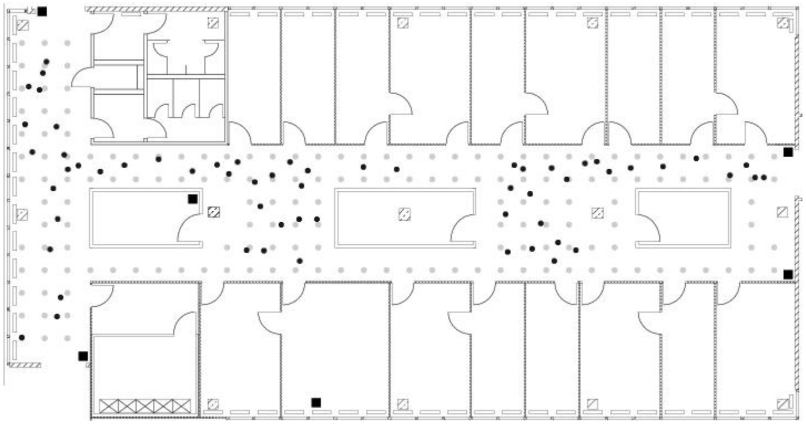
Least Squares Regression

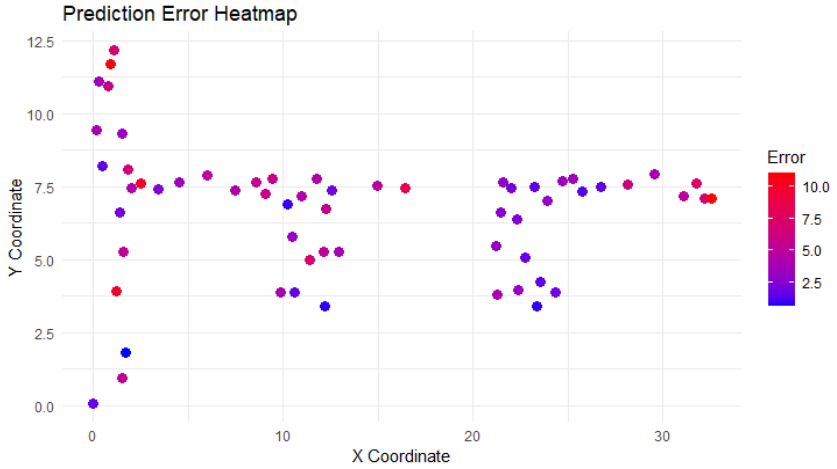




Using a histogram, we show the distribution of the model's residuals for the predicted x and y coordinates. There are some issues here, and using different bin sizes or a QQ plot will likely help us gain more insight.







The first graph shows how far (in distance) our model's predictions were off. The last graph shows how significant our model's error was for each unique measurement location. Using these two graphs against the floor layout, we can check to see what obstacles or other physical features could be contributing to our model's errors. Hopefully, any insight from this will improve the accuracy of the location prediction model.

Conclusion and Recommendations

Key Insights

- A strong inverse relationship exists between signal strength and distance, confirming theoretical expectations.
- Noise filtering, such as removing irrelevant and distant access points, improved the predictive accuracy of both the KNN regression and the Least Squares models.
- Environmental factors like obstacles and device orientation introduce variability, but the general trend remains intact.
- We were able to predict the location by up to 4.8m (16ft), which is not very accurate. Further adjustments will need to be made.
- KNN is simple to implement and effective for scenarios with clean, well-labeled data.
- Trilateration using Least Squares Regression works well in many scenarios but is sensitive to outliers.
- Our limiting factors for this project were time, initial knowledge, and skills.

Concerns

- How do we ensure it is representative of the entire dataset? Are there biases or limitations in the sample that could affect the model's performance?
- How can we reduce noise in the training dataset to improve the model's predictive accuracy? What preprocessing techniques might help?
- What factors might be causing inaccuracies in the model? How could physical obstacles in the building interfere with signal strength?
- Are there assumptions in our data we didn't test for, like Heteroscedasticity (i.e., unequal variance in the signal data), or assumptions in the models we used?
- Could other variables, such as channel frequency or timestamp, be used more effectively than signal strength alone to predict locations?
- What is the relationship between orientation and signal strength? How does this interaction affect our model's predictions, and how can we account for this if necessary?

Recommendations

- **Explore Combining KNN Regression and Trilateration:**
  A hybrid approach could use trilateration to narrow down a general area and then use KNN for finer predictions.

- **Better Understand How Wifi Signals, Devices, and Networking Systems Work**:
  Identify and map how potential obstacles (e.g., walls, furniture, etc.) affect signal attenuation by researching signal strength (RSSI), signal-to-noise ratio (SNR), and interference. Investigate the functioning of WiFi modules, antennas, and

access points.

- **Ensure the Appropriateness of Samples for Data Exploration**:
  For our initial data explorations, we used a given sample. However, we need to ensure that the sample is statistically appropriate for the actual data we are working with.

- **Trialing with Other Location Predicting Methods**:
  For our Indoor Positioning System, we used the Trilateration method based on distances between points. Yet other methods could be explored as well. For example, a Time of Arrival (TOA) algorithm can be used to predict location based on the time it takes for a signal to travel from a transmitter to a receiver.

- **Experiment with Other Machine Learning Algorithms:**
  While KNN regression and Least Squares regression provided us with a fundamental model, other algorithms might work better with our data.

  Some potential algorithms:

  - Random Forest: Combines multiple decision trees to improve accuracy and handle noisy data effectively.
  - Neural Networks: Can model complex, nonlinear relationships in the data and adapt to diverse environments.
- **Gain feedback and insights from and knowledge from other sources**:

  An invaluable asset to this project would be sharing some of our research using online communities like Stack Exchange or speaking with other professionals in the field to troubleshoot and problem-solve our goals and methods.

References

Teixeira-Pinto, A. (2022, July 23). *Machine learning for biostatistics. 2 K-nearest neighbors regression.* Retrieved from https://bookdown.org/tpinto_home/Regression-and-Classification/k-nearest-neighbours-regression.html

IEEE. (n.d.). *Uniwide WiFi-based positioning system.* Retrieved from https://ieeexplore.ieee.org/abstract/document/5514639/

IEEE. (n.d.-a). *An adaptive K-nearest neighbor algorithm.* Retrieved from https://ieeexplore.ieee.org/abstract/document/5569740/

Holtz, Y. (n.d.). *Help and inspiration for R charts.* The R Graph Gallery. Retrieved from https://r-graph-gallery.com/

International Journal of Innovative Technology and Exploring Engineering (IJITEE). (n.d.). Retrieved from https://www.ijitee.org/wp-content/uploads/papers/v5i8/H2255015816.pdf

MathWorks. (n.d.). *Object tracking using time difference of arrival.* Retrieved from https://www.mathworks.com/help/fusion/ug/object-tracking-using-time-difference-of-arrival.html