Final Project Report: Predicting Gym Crowdedness at Fitrec

CS 506: Tools for Data Science
Afzal Khan, Haider Khan, Mohammed Murshid, Rashfiqur Rahman, Maaz Shahzad

Project Overview

Boston University's only major gym and recreation center can often be overwhelmed with an alarming amount of visitors. This project explores the factors influencing gym crowdedness at Boston University's Fitrec center. By analyzing data collected through surveys and timestamps, the aim is to predict the optimal times to visit the gym. The project leverages data science techniques, including data preprocessing, visualization, feature extraction, and machine learning modeling, to achieve this goal.

The analysis considers variables such as time of day, day of the week, weather conditions, and student workloads during different periods (e.g., exams). The ultimate objective is to help gym-goers make informed decisions and alleviate overcrowding during peak hours.

1. Data Collection

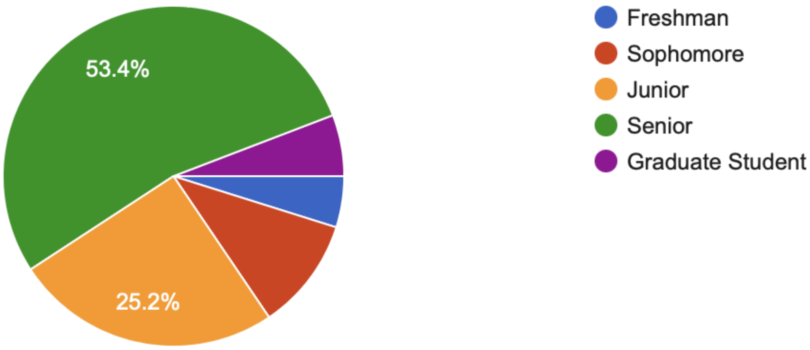
Methodology

The data was collected via a structured survey distributed to Fitrec users. The survey captured the following key elements:

- Timestamp of gym visits.
- Crowdedness rating on a scale of 1 to 5. (1 being not crowded, 5 being really crowded)
- Weekly gym attendance frequency.
- Preferred time slots and days for working out.
- Influence of weather (e.g., rain, snow) on gym attendance.
- Effect of exams or project deadlines on gym visit frequency.

Data Overview

We got a total of 103 respondents. Our initial stage of this project was data collection for which we created a Google form and used it to collect our data. At first it was quite challenging for us to collect data, but we decided to spread our form across various mediums such as Piazza, Discord groups among classes, Reddit, visitors at Fitrec, and among friends. After two weeks of data collection, we were able to get over 100 responses. We were able to get a good split of the various students, mostly being seniors who responded.



2. Data Cleaning and Preprocessing

Steps Taken:

1. **Handling Missing Values:** Missing crowdedness values were imputed with the median of the column.
2. **Timestamp Conversion:** Converted timestamps to datetime objects to extract features like hour and day of the week.
3. **Categorical Encoding:** Converted categorical responses (e.g., "Weekly Frequency") into numerical codes for modeling purposes:
   - Weekly Frequency: {0 - 2 times: 1, 3 - 5 times: 2, More than 5 times: 3}.
   - Timeslot Mapping: {6 AM to 9 AM: 1, 9 AM to 12 PM: 2, and so on}.
4. **Feature Engineering:**
   - Extracted Hour and Day_of_Week from timestamps.
   - Created a binary classification variable for crowdedness (Crowdedness_Binary) to indicate high (4 or above) or low crowdedness.
5. **Handling Multi-Select Responses:**
   - Split responses for days and months into separate rows for accurate frequency counts.

Preprocessed Dataset:

- Features: Year, Weekly Frequency, Hour, Day_of_Week, Timeslot_Num, Crowdedness_Binary.
- Target Variables: Crowdedness (regression) and Crowdedness_Binary (classification).

3. Feature Extraction

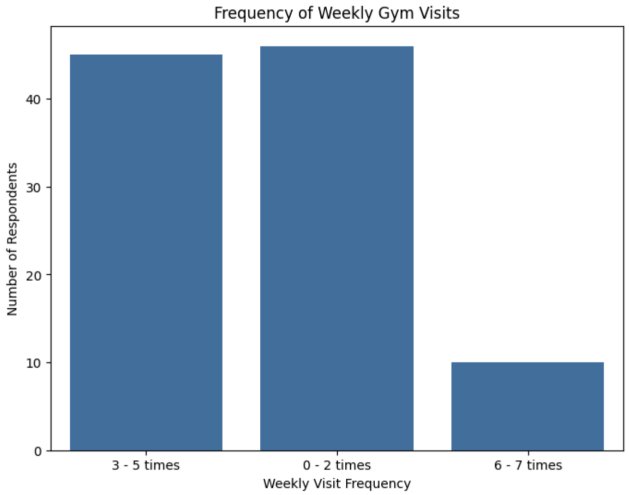
- **Key Features:**

  - **Hour of Day**: Extracted from the timestamp, indicating gym activity during different times of the day.
  - **Day of Week**: Weekday trends affecting gym attendance.
  - **Weekly Frequency**: Students' habitual gym usage patterns.
  - **Weather Influence**: Binary indicator reflecting gym attendance during adverse weather.
  - **Exam Periods**: Captures variations in gym attendance during midterms and finals.
- **Rationale:** These features directly relate to the factors influencing gym crowdedness and provide actionable insights for predictive modeling.

4. Data Visualization

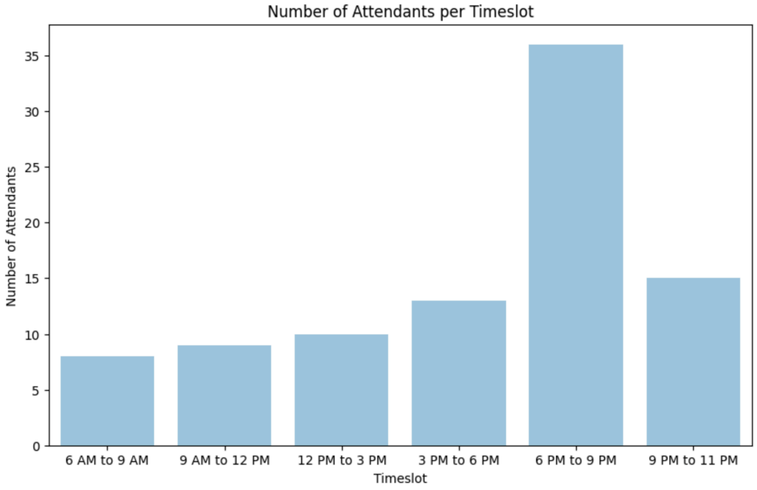
Several visualizations were created to explore patterns and trends in gym usage:

1. **Weekly Gym Visits**
   - Bar chart showing the distribution of weekly attendance frequencies.
   - Insight: Most respondents visited the gym either 3–5 times or 0-2 per week.

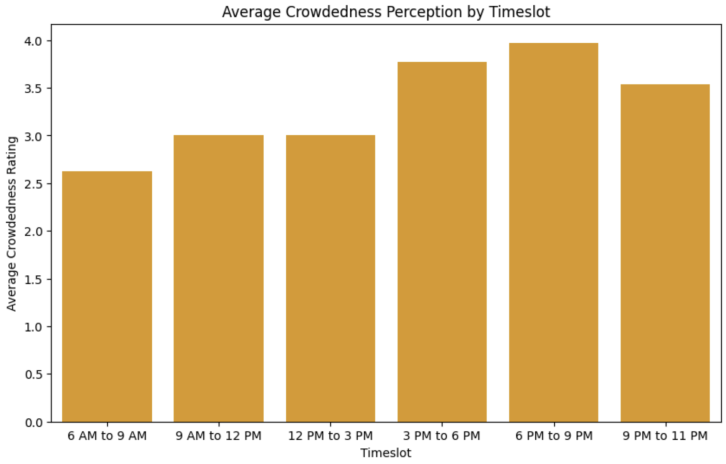


2. **Timeslot Preferences**
   - Bar chart showing attendance per time slot.

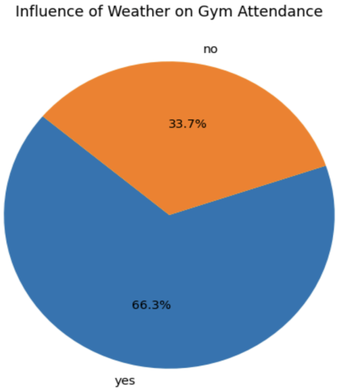
   - Insight: Peak hours were observed between 6 PM and 9 PM.



3. **Average Crowdedness by Timeslot**
   - Bar chart depicting crowdedness ratings per time slot.
   - Insight: The 3 PM to 6 PM and 6 PM to 9 PM slots were rated highest for crowdedness.

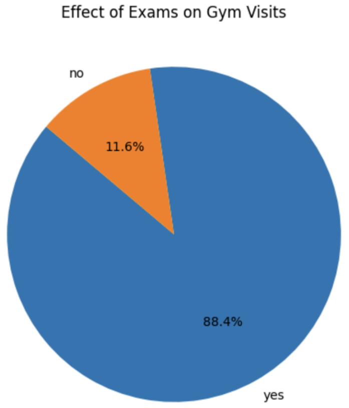


4. **Impact of Weather on Gym Attendance**
   - Pie chart visualizing the percentage of respondents skipping gym due to weather.
   - Insight: Approximately 40% of respondents were influenced by weather.



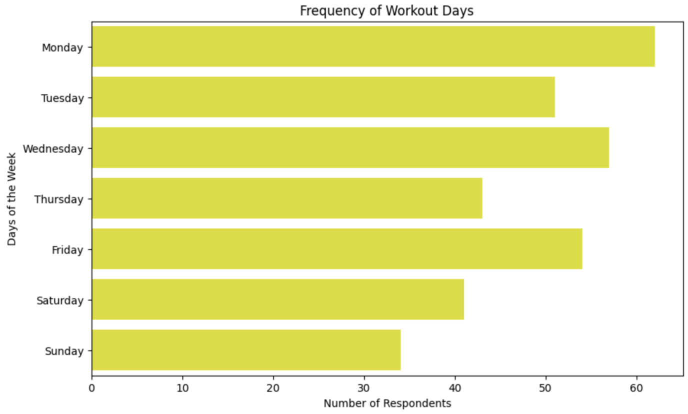
○

5. **Effect of Exam Seasons on Gym Visits**
   - Pie chart showing the percentage of respondents reducing gym frequency during exams.
   - Insight: Exams led to a noticeable dip in gym visits



   - .

6. **Workout Days Preference**
   - Bar chart ranking the frequency of gym visits by day of the week.
   - Insight: Monday and Wednesday were the most popular gym days.

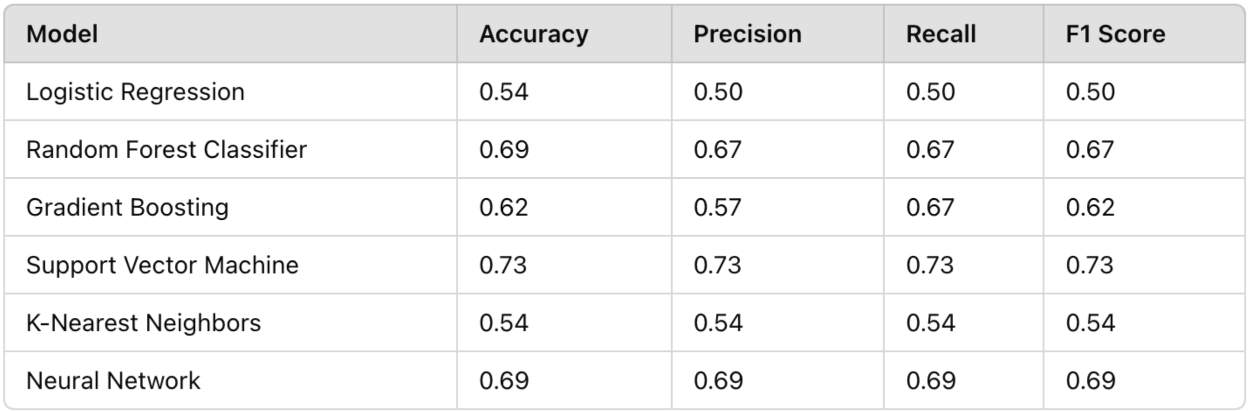


---

5. Model Training

The goal of model training was to classify whether the Fitrec gym is crowded or not based on the survey data. A gym is considered crowded if the crowdedness level is ≥ 4. To achieve this, the dataset was split into a training set (75%) and a testing set (25%). Various machine learning models were evaluated for their performance on this classification task.

Model Performance Table



Model	Accuracy	Precision	Recall	F1 Score
Logistic Regression	0.54	0.50	0.50	0.50
Random Forest Classifier	0.69	0.67	0.67	0.67
Gradient Boosting	0.62	0.57	0.67	0.62
Support Vector Machine	0.73	0.73	0.73	0.73
K-Nearest Neighbors	0.54	0.54	0.54	0.54
Neural Network	0.69	0.69	0.69	0.69

Insights and Model Explanations

1. Logistic Regression

- Logistic regression is a linear model that predicts probabilities for binary classification tasks.
- **Performance:** Accuracy of 54%, with moderate precision, recall, and F1 scores.
- **Challenges:** Logistic regression assumes linear relationships between the features and the target variable, which may not be the case for this dataset. Non-linear interactions (e.g., time of day and gym usage patterns) likely reduced the model's effectiveness.
- **Key Takeaway:** Serves as a baseline model but struggles with complex patterns.

2. Random Forest Classifier

- Random forests are ensemble models that use multiple decision trees to model complex and non-linear relationships.
- **Performance:** Achieved 69% accuracy, performing significantly better than logistic regression.
- **Strengths:** Handles feature interactions well, such as the interplay between weekday and timeslot. The ensemble nature reduces overfitting.
- **Challenges:** Limited by the small dataset, which prevents the model from training deeper trees and fully leveraging its potential.
- **Key Takeaway:** A robust choice for small-to-moderate datasets with non-linear features.

3. Gradient Boosting Classifier

- Gradient boosting iteratively builds decision trees, learning from the errors of previous iterations.
- **Performance:** Accuracy of 62%, with higher recall (67%) than precision (57%).
- **Strengths:** Excels at capturing subtle patterns in the data.
- **Challenges:** Susceptible to noise and overfitting, especially with smaller datasets.
- **Key Takeaway:** While powerful, its potential is limited by the dataset's size and inherent noise.

4. Support Vector Machine (SVM)

- SVMs find a hyperplane that best separates the data into distinct classes.
- **Performance:** Best-performing model with an accuracy of 73%.
- **Strengths:** Focuses on maximizing the margin between classes, making it effective for overlapping data points.
- **Challenges:** Computational complexity increases with larger datasets or when using non-linear kernels.
- **Key Takeaway:** A strong performer due to its robustness to class imbalances and noisy features.

5. K-Nearest Neighbors (KNN)

- KNN assigns classes based on the majority vote of the nearest neighbors.
- **Performance:** Accuracy of 54%, the lowest among the models.
- **Challenges:** Performs poorly on small, imbalanced datasets and is sensitive to feature scaling. The small dataset size likely added noise.
- **Key Takeaway:** KNN is not well-suited for this dataset due to its simplicity and dependence on well-distributed data.

6. Neural Network

- Neural networks can capture complex, non-linear relationships by adjusting weights across multiple layers.
- **Performance:** Matched the random forest with an accuracy of 69%.
- **Strengths:** Suitable for complex patterns and feature interactions.
- **Challenges:** Neural networks require larger datasets to avoid overfitting and achieve better generalization.
- **Key Takeaway:** The small dataset constrained its performance, though it showed promise with balanced precision and recall.

Results Summary

- The **Support Vector Machine (SVM)** was the top-performing model with an accuracy of 73%. Its focus on maximizing the margin between classes proved advantageous for the dataset's characteristics.
- The **Random Forest Classifier** and **Neural Network** also performed well, both achieving 69% accuracy. They effectively captured non-linear patterns but were limited by the dataset size.
- Simpler models like **Logistic Regression** and **KNN** struggled to handle the dataset's complexity, resulting in lower performance.

---

6. Conclusion

This analysis demonstrates that advanced models such as SVM and Random Forests outperform simpler models in predicting gym crowdedness. The findings indicate that crowdedness is influenced by complex, non-linear interactions between features like time of day, day of the week, and weekly attendance frequency.

The insights gained can guide gym-goers to plan their visits during less crowded times and help Fitrec management implement crowd control measures during peak periods.

---

7. Future Work

1. **Dataset Expansion:** Collect more data to improve model performance and generalization.
2. **Feature Engineering:** Incorporate additional features, such as real-time weather data and academic schedules.
3. **Interactive Dashboard:** Create a user-friendly tool for gym-goers to check predicted crowdedness levels.
4. **Advanced Models:** Explore deep learning techniques with a larger dataset for improved performance.

---

8. Workflow

1.) Clone into the repository
2.) Navigate inside the “506_final” directory
3.) Run “make install” - this will install any necessary packages
4.) Run “make run”