FindParking@NJIT
Ricky Palaguachi, Steven Kyritsis, Austin Fontaine

Project for CS 301 - Data Science at NJIT

Quick Links

Check out the overview on YouTube here.
Github Repo here.

Background

Over 70% of NJIT students commute. Some by train, most by car. NJIT's Parking decks fill up too fast--this is frustrating for commuters. Our goal is to predict parking levels for a given day and time.

Related Papers

There is similar, more sophisticated, published research on the topic already. Authors Klandev, et. al. have implemented a smart parking system providing predictions about real-time parking occupancy. They used traffic congestion information and regression models to predict availability. The full paper is available here.

Data Handling

1. Scrape parking data from http://mobile.njit.edu/parking using Python
2. Automate this on a Raspberry Pi to collect data every 60 seconds
3. Clean the data and save it to Google Sheets
4. Repeat forever

Methods and Experimentation

How did we gather insights from the data?
We grouped data “vertically” along the day of week. So for each Monday, we took each parking decks’ average availability as a function of time.

Alternatives Considered
We could also aggregate the data “horizontally”. This means, for a particularly busy week (e.g., finals week) the availability might be drastically reduced and the model would be more conservative.

Parking congestion could also be influenced by events and holidays, such as midterm exams, campus tours, concerts, sporting events, parties, and more.

The number of commuters each year fluctuates slightly. As the student body increases so does the nominal amount of commuters. Since the number of parking spots remains constant, if more commuters come each year, congestion will rise. Additionally, some on-campus residents might have an overnight parking spot as well, which would increase directly with the number of students at NJIT.

What was the design of our system?

The system took as input

(i) the day of the week, and
(ii) the location of the user,

which would then be mapped to NJIT to determine an **arrival time** (e.g., Monday @ 12:30 PM).

Based on the commute duration, we must predict what the parking levels would be in the next 24 minutes. Our time series analysis prediction works by analyzing

a) Average trends for the last 7 same days-of-week (e.g., past 7 Mondays).
~~b) Trends for the last 15, 30, and 60 minutes using an EWMA.~~

What existing tools/libraries were used?
The usual suspects of Data Science were used to solve this problem. Namely,

- Pandas
- NumPy
- SciKit Learn
- Google Sheets API
- Google Maps API

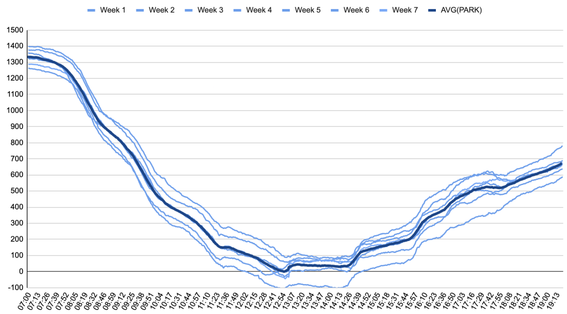
What were the benefits and drawbacks of each approach?
There was high precision for our dataset. Every 60 seconds, the Raspberry Pi pinged NJIT’s Parking Sensors to gather a reading. This allowed for 1440 readings per day per parking deck. Over 30 days, we’ve gathered over 43,000 readings.

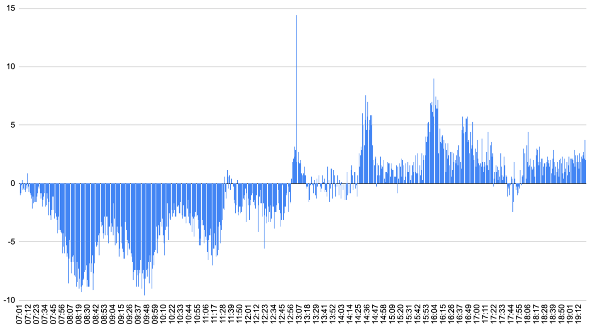
A considerable drawback was the lack of dimensionality with our dataset. The only reading we could make was the count of available spots. No finer details were measurable, such as the per-floor availability.

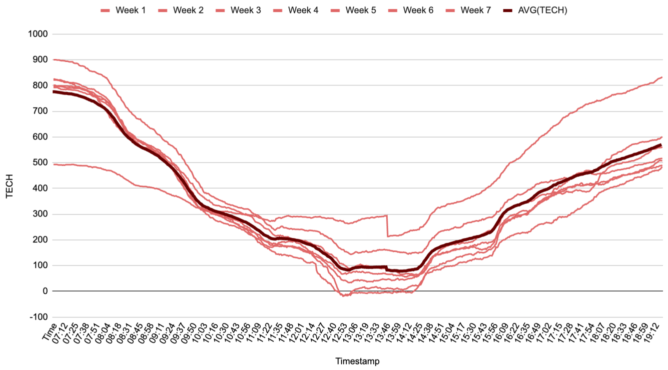
Conclusion

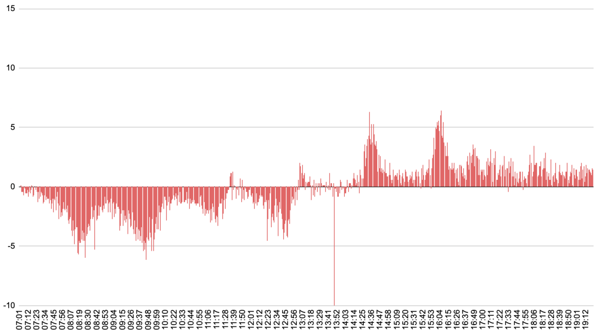
Our project was able to give the user an accurate prediction of the amount of available spots in the selected parking deck given the user inputted ETA. Additionally, we were able to successfully notify whether the parking decks were filling or emptying based on positive or negative rates of change in the spot availability.

MONDAYS


PARK on Mondays

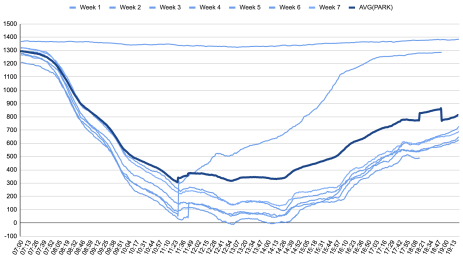

First Derivative or PARK on Mondays


TECH on Mondays

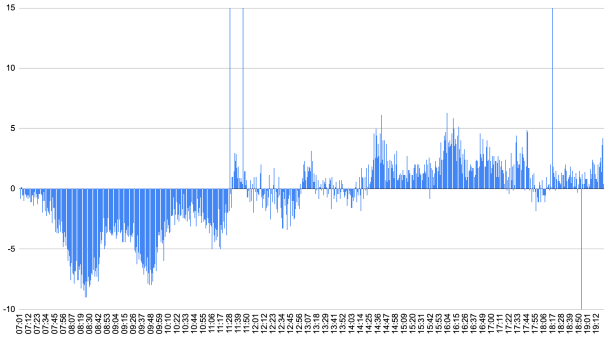

First Derivative for TECH on Mondays

TUESDAYS

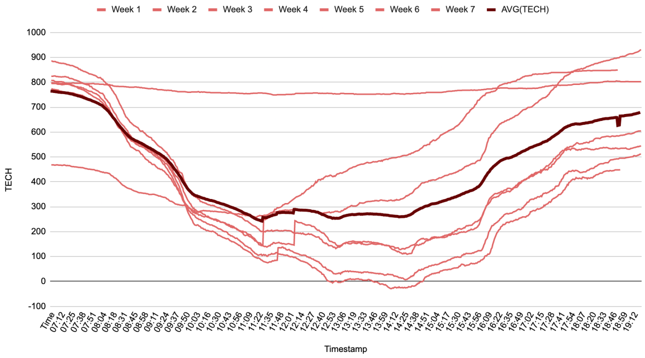
PARK on Tuesdays



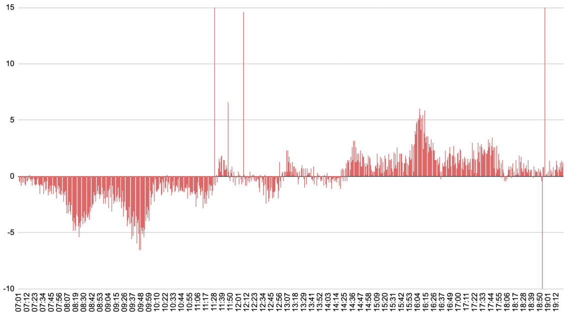
First Derivative or PARK on Tuesdays



TECH on Tuesdays

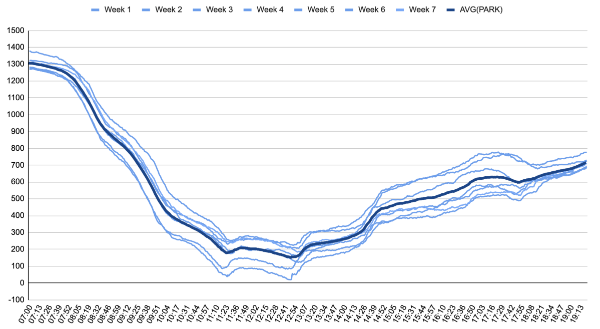


First Derivative for TECH on Tuesdays

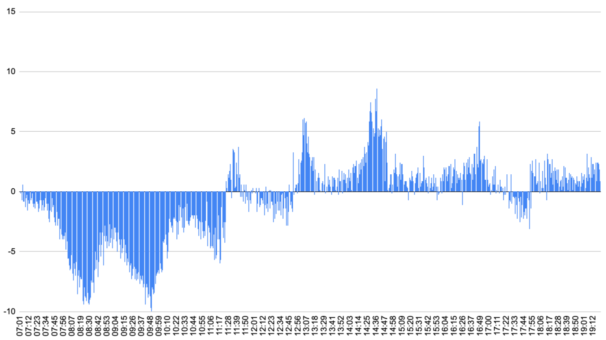


WEDNESDAYS

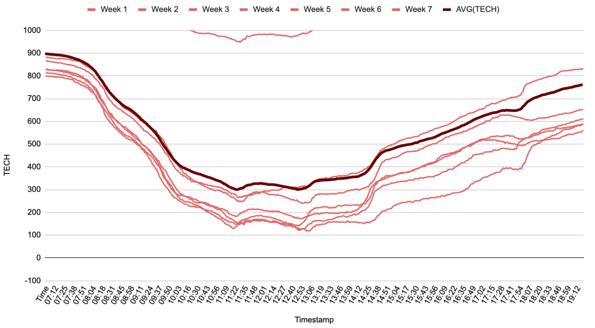
PARK on Wednesdays



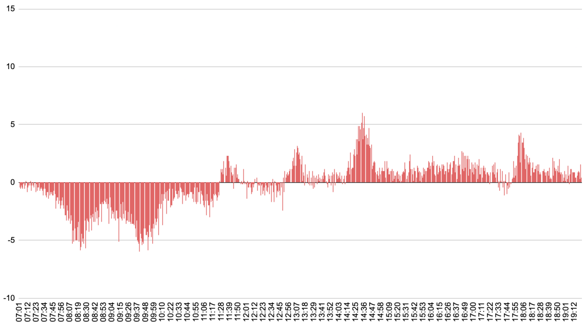
First Derivative or PARK on Wednesdays



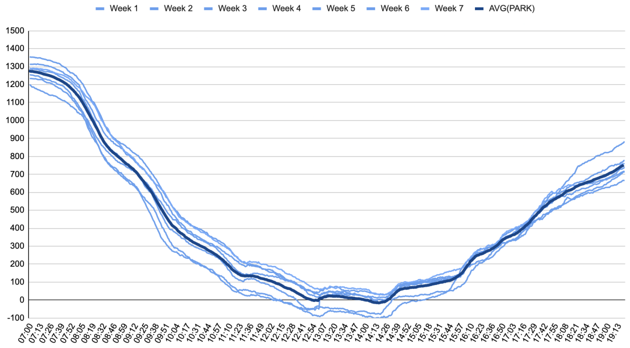
TECH on Wednesdays

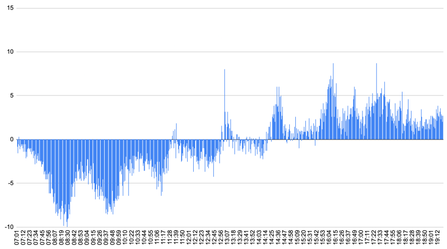


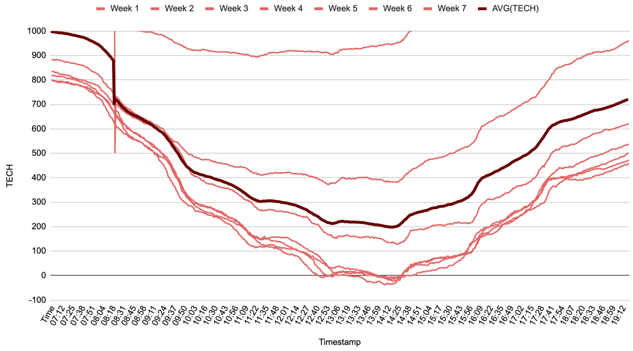
First Derivative for TECH on Tuesdays

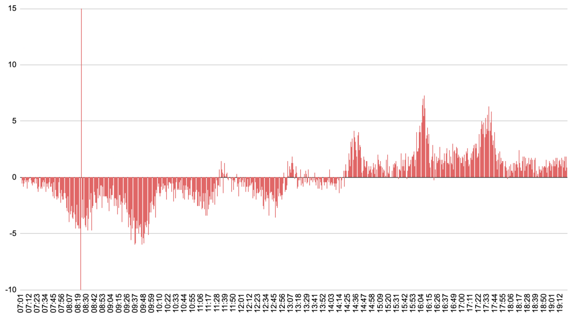


THURSDAYS


PARK on Thursdays

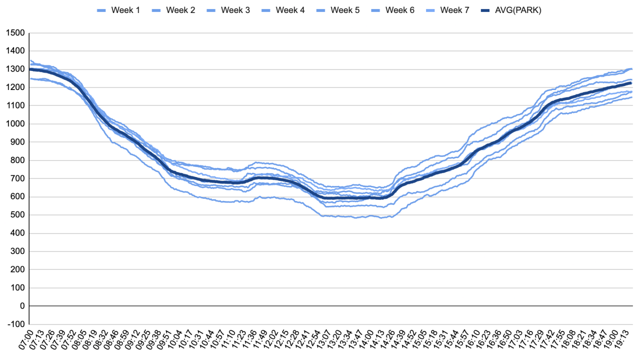

First Derivative or PARK on Thursdays


TECH on Thursdays

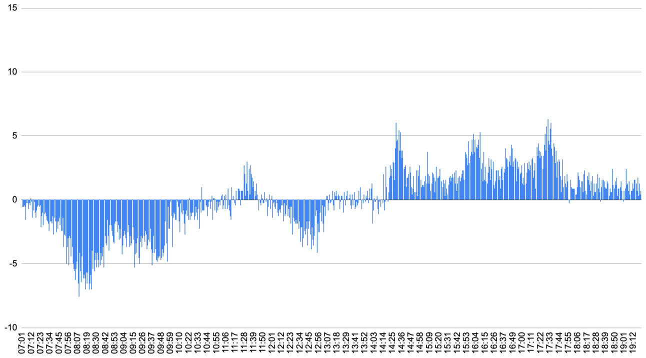

First Derivative for TECH on Tuesdays

FRIDAYS

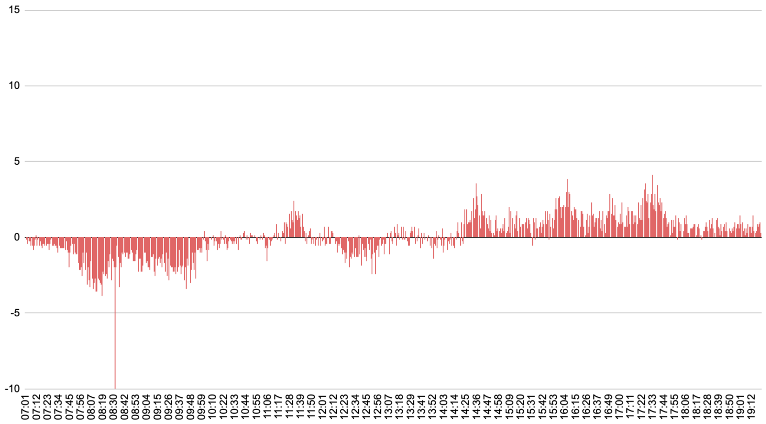
PARK on Fridays

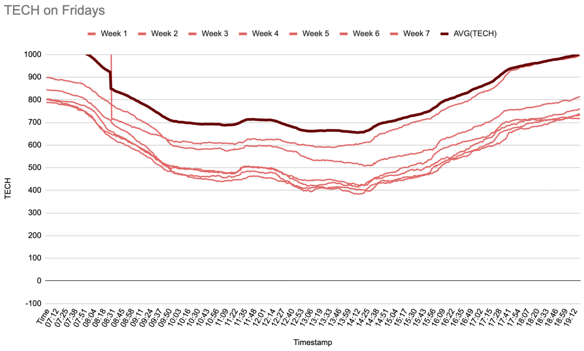


First Derivative or PARK on Fridays



First Derivative for TECH on Fridays




TECH on Fridays
Week 1
Week 2
Week 3
Week 4
Week 5
Week 6
Week 7
AVG(TECH)
1000
900
800
700
600
500
400
300
200
100
0
-100
TECH
Time
07:12
07:25
07:38
07:51
08:04
08:18
08:31
08:45
08:58
09:11
09:24
09:37
09:50
10:03
10:16
10:30
10:43
10:56
11:09
11:22
11:35
11:48
12:01
12:14
12:27
12:40
12:53
13:06
13:19
13:33
13:46
13:59
14:12
14:25
14:38
14:51
15:04
15:17
15:30
15:43
15:56
16:09
16:22
16:35
16:49
17:02
17:15
17:28
17:41
17:54
18:07
18:20
18:33
18:46
18:59
19:12
Timestamp