

Magic Minutes

and yet... it compiles

Denver Simmons

Olivia Clough

Ryan Mitchell

Abby Wysopal

Cole Bergmann

Why we chose this project

- Disneyland Parks in Anaheim, California attracts over 18.3 million visitors per year
- Each visitor arrives at Disneyland with the risk of waiting in line for multiple hours per attraction
- We believe there should be a way to predict wait times and crowd sizes within the park.

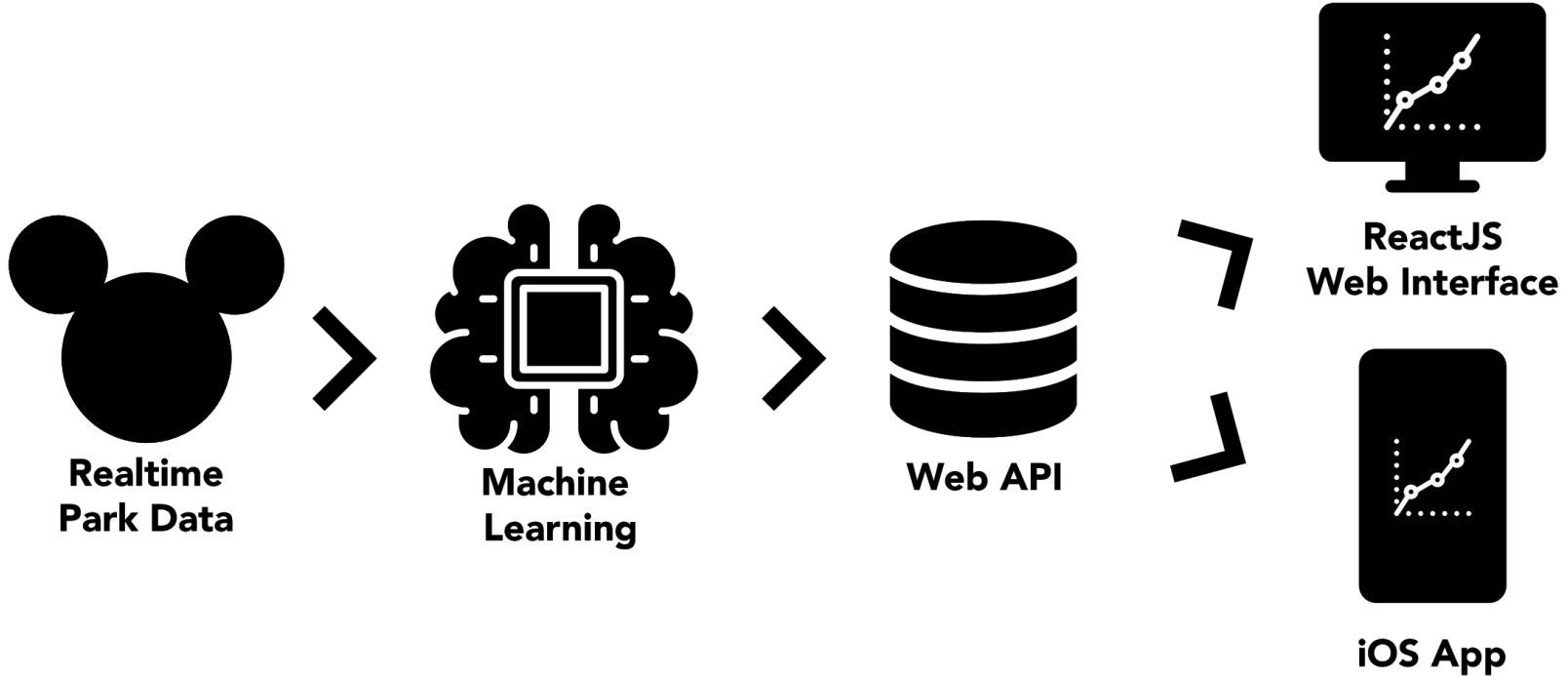
Disney's current technology

- Disneyland currently has a mobile application that contains a variety of features however they focus on advertising premium fast passes.
- We used machine learning to train an algorithm using data points found from their live updates to predict wait times for different rides throughout the day as well as total park population for the next month.

Technologies Used

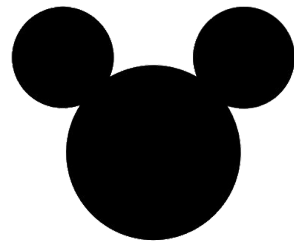
- ReactJS
- Tensorflow
- Java
- Python
- MySQL
- Travis-CI
- Selenium
- Swift
- Ubuntu

Overview



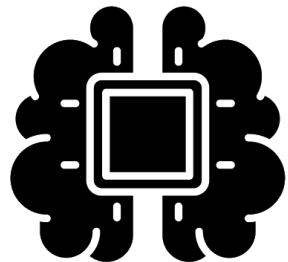
Realtime Data

- Started with historical data from November 2017 to present
- Needed to format historical data so the machine learning model could easily read it
- Created a Java tool to import our old data, remove outliers, add additional parameters (temperature, precipitation probability, etc), and export to desired format
- Created another Java program that runs every 10 minutes to add the latest data to our database

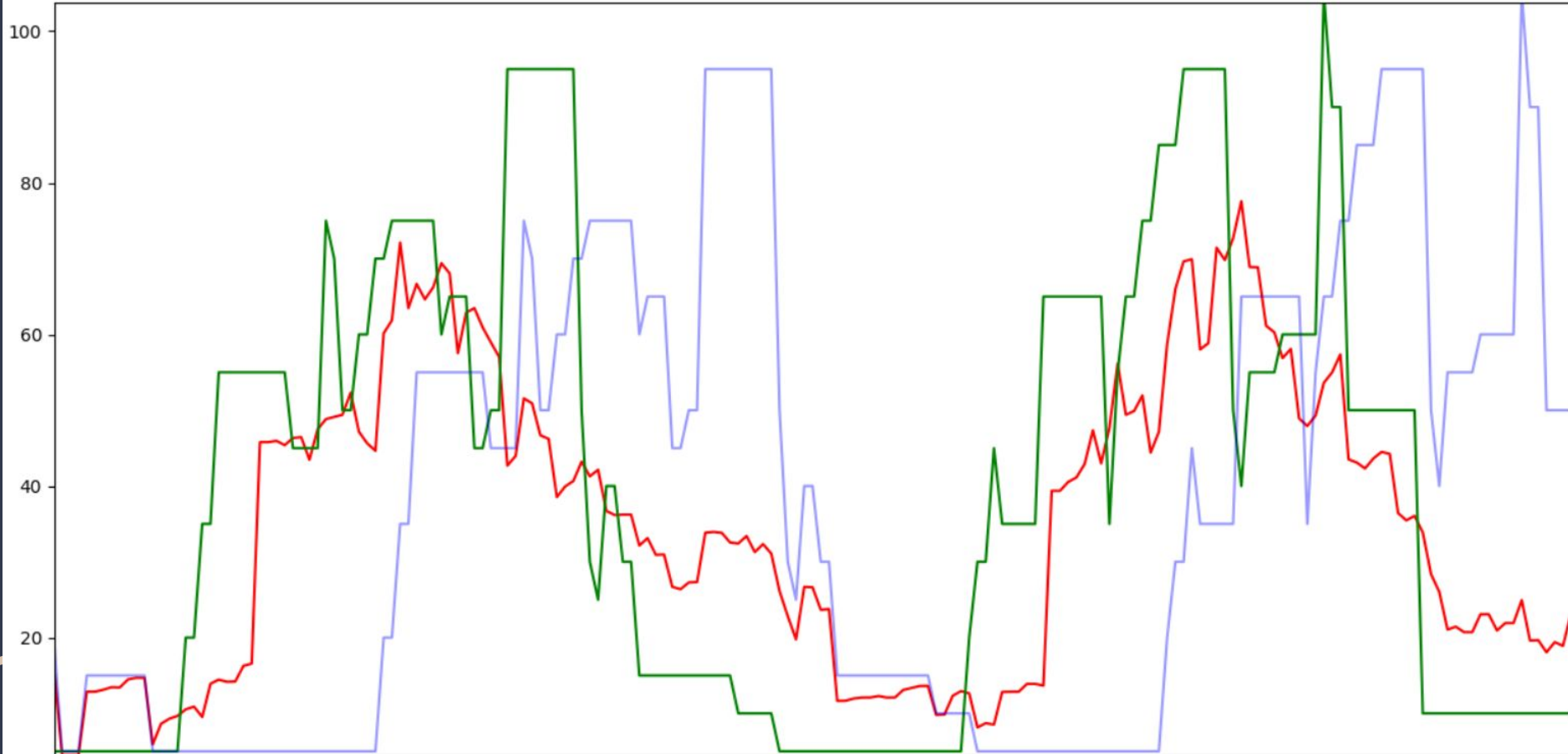


Machine Learning

- Tensorflow
- LSTM Network
- Ride Prediction Model Inputs:
 - All current ride wait times
 - Current temperature
 - Current humidity
- Ride Prediction Model Output:
 - Ride 'A' wait time at time 'X' in the future



Two Days of Splash Mountain Predicting 4 Hours Into the Future



Using the Model

- Creating SQL Database
- Writing/Reading with Database
- Updating Database
- Hosting API Endpoint
- Merging Real Time Data with Predicted Data

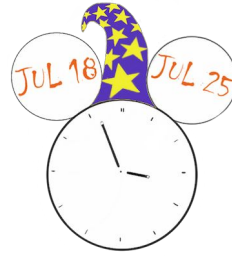
```
+-----+  
| Tables_in_cs48 |  
+-----+  
| population  
| ride0  
| ride1  
| ride2  
| ride3  
| ride4  
| ride5  
| ride6  
| ride7  
| ride8  
| ride9  
+-----+
```



Connecting to the front end

- Fetching the Data
 - Fetching an array of data from API
- Setting the variable state
 - Array of both predictive and historical data
- Displaying data
 - ChartJS

Demo



Magic Minutes

<https://magic-minute.me/>



ReactJS
Web Interface



iOS App

Challenges

- Dividing tasks in the best way possible
- Had not used the languages before, had to use Sprint 1 as a learning curve
- Memory space on the server
- Runtime for machine learning caused errors in fetching
- Learning how to connect different interfaces to transfer data

Future

- Android app
- Add in more rides to the Wait Time section
- Add the capability for the user to set a date and select a list of rides they want to go on. The app will create the best schedule.
- Further improve the model with additional heuristics (% of schools in session, etc)
- With more data, training models to predict any day of the year.
- Feature that would show you how many minutes a fast-pass would save you.

Summary

Main Goal: Provide accurate predictive models for the wait times and park population at Disneyland to allow users to spend their time and plan their trips more efficiently.

Features:

Links:

<https://magic-minute.me/>

<https://trello.com/b/iG3UuHoD/cs48>

<https://github.com/colebergmann/and-yet-it-compiles>