# “Park My Bike” Application Overview

Authors: Annie Lee and Roiana Reid

## Table of Contents

# W205-Final-Project/

The “Park My Bike” application recommends bike parking locations based on the user’s current location and the relative safety of the bike racks which are situated nearby. The safety of bike parking racks are measured by the prevalence of crimes in the vicinity since 2013. This application is built for use in San Francisco and Chicago.

The Github repository for this project contains all the code we used to obtain, clean, explore and analyze the data and the tools used to implement the solution. We have also included examples of the layout of the serving layer for this application. Our final presentation and instructions on how to replicate the analysis and design are also located in the repository.

# data/

The “data” folder contains some of the pertinent data used in our analysis. These include the raw bike parking locations data, files containing the count of crimes committed near the bike racks since 2013 and the safety scores assigned to each of the bike parking locations. The raw crime data is not located in this folder due to github file size limitations. However, the setup scripts provide details on how to download the data.

## chi_bike_data.csv

This file contains bike parking data for Chicago that consists of the rack ID, address, and the angular distances of the bike racks (latitude and longitude). The data was obtained from the City of Chicago (link).

## chi_parking_scores.csv

This file contains the number of crimes committed since 2013 within a 0.1 mile radius of each bike parking location in Chicago (with at least five parking spaces). The three crimes counted were theft, robbery and motor vehicle theft.

## chi_score_f.csv

This file contains the risk score for each of the bike parking locations analyzed in Chicago. The scores are measured on a scale from 0 to 10, with 10 being the safest location and 0 being the most dangerous location.

## sf_bike_data.csv

This file contains bike parking data for San Francisco that consists of the address, location name and the angular distances of the bike racks (latitude and longitude). The data was obtained from the City and County of San Francisco (link) and then cleaned up with the format_sf_bike_parking_data.sh script.

### sf_parking_scores.csv

This file contains the number of crimes committed since 2013 within a 0.1 mile radius of each bike parking location in San Francisco (with at least three parking spaces). The three crimes counted were theft, robbery and vehicle theft.

### sf_score_f.csv

This file contains the risk score for each of the bike parking locations analyzed in San Francisco. The scores are measured on a scale from 0 to 10, with 10 being the safest location and 0 being the most dangerous location.

## screenshots/

This folder contains example images of the user query responses provided by the serving layer of the “Park My Bike” application.

### Chicago Example Screenshot.png

This file is an image of what the user query result would look like for a user in Chicago. The image shows the information that is returned when a user enters his/her current location. The API displays five recommended bike parking locations within a quarter-mile radius. For each location, the address, rack ID, distance from provided location, and the safety score are returned. The five locations are listed in order of safety, from the safest location (highest score) to the least safe location (lowest score).

### SF Example Screenshot.png

This file is an image of what the query result would look like for a user in San Francisco. The image shows the information that is returned when a user enters his/her current location. The API displays five recommended bike parking locations within a quarter-mile radius. For each location, the address, rack ID, distance from provided location, and the safety score are returned. The five locations are listed in order of safety, from the safest location (highest score) to the least safe location (lowest score).

## scripts/

Folder containing the shell, python and SQL scripts used to set up the EC2 environment, clean, load and explore the data, conduct the scoring analysis and create the serving layer for the “Park My Bike” application.

## Setting up the programming environment

### setup_env.sh

This shell script installs Hadoop, Spark SQL and Postgres using scripts provided by the UCB W205 course. It also sets up the Python 2.7 virtual environment and installs the necessary Python packages that are used to implement the solution.

### requirements.txt

This file contains a list of pertinent Python libraries for the project. The setup_env.sh script uses this file to install these Python libraries.

## Data downloading, cleaning and exploration

### explore_sf_data.sql

This SQL script was used to load the raw crime and bike datasets for San Francisco into Hive. It contains the queries we used to perform the initial exploratory analysis on the crime and bike data for San Francisco which enabled us to identify various trends within the data and fine-tune the focus of our project.

### setup_data_tables.sh

This shell script is used to download the raw datasets and load the final scores into Hive. The raw datasets downloaded here are the crime data for San Francisco and Chicago, and the bike parking data for Chicago and San Francisco. These data are downloaded, formatted and loaded into HDFS. Also runs various scripts to load pre-aggregated data and compute scores for bike parking data.

### format_sf_bike_parking_data.sh

This shell script is called by the setup_data_tables.sh file. It contains code to clean up the San Francisco bike parking data as the csv file had a messy schema. In particular, the latitude and longitude coordinates were stored as a pair in one column and there were random line breaks and extra text. As such, this script is designed to remove extra line breaks, remove extra text, and split the location column into two separate columns, one for the longitude coordinates and the other for the latitude coordinates.

## Analytical component, merging of data

### aggregates_chi.py

This  Python script contains the schema for loading the Chicago crime and bike parking installments datasets into tables. The script also contains code that ensures the data are well formatted before loading them into tables and then further cleans and filters out entries missing critical data or malformed data. Another table is then created for areas within a 0.1 mile radius of the bike parking locations (creating search boundaries). A Cartesian product is then used to join the aforementioned table (containing the search boundaries) and the crime data table. Code in the script is then used to count the total number of theft, robbery, and motor vehicle theft crimes occurring near each bike parking location within the search boundaries. The results are saved to a csv file “chi_parking_scores”. Note that only bike parking locations in Chicago with at least five parking spaces are used.

### aggregates_sf.py

This  Python script contains the schema for loading the San Francisco crime and bike parking installments datasets into tables. The script also contains code that ensures the data are well formatted before loading them into tables. Another table is then created for areas within a 0.1 mile radius of the bike parking locations (creating search boundaries). A Cartesian product is then used to join the aforementioned table (containing the search boundaries) and the crime data table. Code in the script is then used to count the total number of theft, robbery, and motor vehicle theft crimes occurring near bike parking locations within the search boundaries. The results are saved to a csv file “sf_parking_scores”. Note that only bike parking locations in San Francisco with at least three parking spaces are used.

### calculate_score_sf.sql

This SQL script contains the schema for creating tables from the sf_parking_scores.csv file described above. The table contains the address of bike racks in San Francisco, the rack ID and the number of crimes in three different categories ('VEHICLE THEFT', 'LARCENY/THEFT', 'ROBBERY') committed within .1 mile of each bike parking location since 2013. Queries are then used to rank each location (in descending order), for each category of crime, by the number of crimes  which have occurred within the vicinity of the bike rack. The ranks for each of the three crimes are then averaged to create an overall rank for each of the bike parking locations. In order to create a safety score, the ranks are normalized to fall between 0 and 10, where 10 is the safest location and 0 is the most dangerous location.

### calculate_score_chi.sql

This SQL script contains the schema for creating tables from the chi_parking_scores.csv file described above. The table contains the address of bike racks in Chicago, the rack ID and the number of crimes in three different categories ('MOTOR VEHICLE THEFT', 'THEFT', 'ROBBERY') committed within .1 mile of each bike parking location since 2013. Queries are then used to rank each location (in descending order), for each category of crime, by the number of crimes  which have occurred within the vicinity of the bike rack. The ranks for each of the three crimes are then averaged to create an overall rank for each of the bike parking locations. In order to create a safety score, the ranks are normalized to fall between 0 and 10, where 10 is the safest location and 0 is the most dangerous location.

### upload_scores.sh

This shell script is used to upload the aggregate data (*_parking_scores.csv) into Hadoop and to run the SQL scripts (calculate_score_*.sql) used to create the scoring tables in Hive.

## Serving Component with Real Time Processing

### app.py

This Python script creates the application interface for accessing the bike parking data. The user enters a location and queries are used to find the scores of nearby bike racks. The locations are sorted by score and the top five locations - in terms of score - are returned to the user along with the location's score, address and the distance to the user. This script creates our serving layer.

# README.md

Markdown document containing instructions on how to run the code provided in the Github repository in order to replicate our analysis and run the application.

# final_presentation.pdf

PDF of the presentation we made on the "Park My Bike" app. This presentation contained a description of the problem we are solving, its importance, the data sources and architecture used, and results from exploratory analysis of the San Francisco data. We also described how we processed the data and gave the class a snapshot of our serving layer. In the presentation we discussed the challenges we faced and near-term and future goals.