# Parking Availability Prediction based on Machine Learning Approaches: A Case Study in the Short North Area

Thesis

Presented in Partial Fulfillment of the Requirements for the Degree Master of Science in the Graduate School of The Ohio State University

By

Yuxiao Zhao

Graduate Program in Computer Science and Engineering

The Ohio State University

2020

Thesis Committee

Dr. Dave Ogle, Advisor

Dr. Thomas Bihari

Abstract

Parking availability information can help drivers make decisions on where to park, reduce road congestion, and balance the parking demand. This study utilizes the historic parking meter transactions to estimate the parking occupancy aggregated by parking zones in the Short North, Columbus. Daily and weekly recurring patterns are identified from the parking availability time series. Clustering algorithms are used to cluster the average weekly time series of each parking zone to identify similar parking patterns. And machine learning algorithms are trained to make predictions for each cluster of zones based on input features, including time of the day, day of the week, and month. The study compares different clustering algorithms and machine learning algorithms to choose the model with the best performance. Agglomerative clustering shows more solid outcomes than k-means clustering. Multilayer perceptron (MLP) with two hidden layers and 50 hidden units each layer outperforms other machine learning algorithms and MLPs with other parameters. This study shows the potential of using historical transaction data and machine learning algorithms to make parking availability predictions. The pipeline of data collection, data cleaning, data exploration, data transformation, feature engineering, model selection, modeling, and model evaluations in this study could be reproduced to apply to other areas.

Dedication

To my parents for their most supportive love.

Acknowledgments

First, I want to thank Prof. Dave Ogle for always being supportive and helpful with my work. If it were not for you, I would not get the opportunity to intern with the Smart Columbus project. I really appreciate you could see my match with this project and help me get involved with their team. And as my advisor, you are always concerned about my coursework, career development, and research. You are always willing and patient to take time to meet with me weekly and check how everything goes. Your guidance about my research and work is very much appreciated.

Second, I want to thank Prof. Tom Bihari for being on my committee. Your advice on my research and proposal writing is of great help.

Third, I want to thank my advisor Prof. Ningchuan Xiao in the Geography department who supported me in pursuing a dual degree and encouraged me to do what suits my best. You have always been so kind and patient about my work. And you are also straightforward and have a very open mind to your students.

Fourth, I want to thank my colleges in Smart Columbus, especially Dr. Hong Li. Without your devoted help, much work could not have been done. I appreciate talking with you about data science problems and sharing thoughts with you.

Fifth, I want to thank my parents and friends for keeping me company, physically or virtually. Meicheng Shen, you are my best friend I have made in the United States. I am

grateful to have met you. Ning Zhang, my great roommate, is a joy to live with and has brought me a lot of good memories. Luyu Liu, you are a lot of fun and can really cheer everybody up. Jialin Li, you have got an amazing talent in saving money, and I look forward to seeing you being rich in the future. Zhiying Li, you are a perfect travel planner, reliable trip partner, great financial guide, and my unforgettable CVS co-visitor. Rebecca Chapman, I have one of the best memories hanging out with you, and you are so kind and charming. Lingwei Li, Scarlett Jin, and Jiayong Liang, my wonderful workout squad, I miss and will miss so much the time we dine together. My parents, thanks for always being caring about my well-being and thesis progress. I wish I could have the chance to show you around the amazingly average but adorable town, Columbus if it were not for the c-virus.

Lastly, I want to thank myself. You are the one who has all sorts of weaknesses but made the way to today with all these amazing, intelligent, and hardworking people around. I wish you all the luck and courage to go for what you want. “We are all in the gutter, but some of us are looking at the stars.”, from Oscar Wilde.

## Vita

2017 …………………………………………………………. B.S in Geography Science,

Wuhan University, Wuhan, China

2018 – 2019 ……………………………………………... Graduate Teaching Associate,

The Ohio State University, Columbus, Ohio

2019 ………………………………………………… Software Product Engineer Intern,

ESRI., Redlands, California

2019 – 2020 ………………………. Data Scientist Intern (Graduate Research Associate),

Smart Columbus, Columbus, Ohio

## Fields of Study

Major Field: Computer Science and Engineering

Table of Contents

List of Tables

List of Figures

# Chapter 1 Introduction

## 1.1 Background and Problem Statement

Parking availability is one of the major concerns in urban transportation systems. As the population density, economic activities, and road construction grow in Columbus, road congestion is becoming more severe, especially in areas like the Short North and downtown. During peak hours, vehicles searching for parking, circling and idling, increase congestion. This leads to longer travel time, lower road use efficiency, and more vehicle emissions. The inefficiency of searching for parking is due to a lack of real-time and near-future information of parking availability that can guide travelers to find a parking space more efficiently (Yang et al., 2019). A better system to manage parking demand and provide parking availability information to drivers would help the City of Columbus solve the problem of parking space shortage. Additionally, location and availability information of parking meters could discourage parking violations and overtime parking.

Moreover, by analyzing and predicting the pattern of parking occupancy, the city could adjust the rate of parking according to how busy the area is. This would balance the parking demand in different areas, increase the parking revenue, and encourage more mobility options.

The Event Parking Management (EPM) project of Columbus aims to help people search for, reserve, and pay for parking space in garages, parking lots or on-street parking meters through mobile or web apps. EPM aims to build a more intelligent parking management system and makes searching for parking easier. One of the essential features needed in such applications is the ability to predict the real-time or near-future availability of parking spaces. On-street short-term paid parking is often managed by parking meters, and thus predicting the availability of on-street parking meters is set as one of the goals of the project.

There have been many studies on making predictions on short-term parking availability or occupancy based on different assumptions and methods. Some studies predict the parking occupancy based on individual drivers' behaviors on the microscale (Caicedo et al., 2012; Millard-Ball et al., 2014; Yang & Qian, 2017). Such studies often analyze the parking behaviors and model the arrival and departure time of the vehicles by some probabilistic distributions. Others apply data-driven methods on the macroscale using the parking occupancy data aggregated in a small area such as a street block (Shao et al., 2019; Tamrazian et al., 2015; Yang et al., 2019). For these types of data-driven approaches, machine learning or deep learning algorithms are used to learn the spatial-temporal pattern of parking from the historical parking data and predict the real-time or near-future parking occupancy.

The development of the smart city project in Columbus makes this type of application possible by providing the platform and the data. Big open data, especially data about urban transportation, has been made available to the public through Smart Columbus

Operating System[1]. These data include important information about how parking meters were historically used and their regulations and locations, which could be used to analyze the parking patterns and predict future trends. They serve as an important backbone of a data-driven parking prediction system to improve the parking services and management, provide more convenient parking services, and better parking experience. Parking meter transaction data have been used in this project to predict a zone-based parking occupancy. For the granularity of parking, many data-driven analyses and predictions of parking are based on the occupancy aggregated into a small geographical unit. In this study, the city of Columbus has pre-defined the parking zones based on street blocks, block faces, and meter configurations. In one parking zone, meters are often in the same street block, on the same side of the same street and with the same operating hours and rate. There are 83 zones and 687 meters in the Short North area.

In this study, the Short North Area is chosen as a pilot area to develop, implement and test predictive models of parking availability as it is one of the busiest areas where parking is in high demand during peak time. Parking availability is defined as the count of unoccupied parking meters out of the total count of meters in a zone, which is geographical units defined by Columbus Parking Services based on street blocks and meter configurations. The terms “parking occupancy” and “parking availability” are used interchangeably in the thesis as one is the opposite to the other, for example, high occupancy indicates low availability of parking.

[1] https://discovery.smartcolumbusos.com/

## 1.2 Thesis objectives

In this thesis, we use publicly available historical data of parking to build machine learning models in order to predict parking availability.

Three sub-objectives have been put forward for this study. The first is to analyze the patterns of parking availability in the Short North area in Columbus. The second is to develop Predictive Models for the Short North using different machine learning methods and historic parking meter transaction data. The third objective is to design a lifecycle of data collecting, data exploration, data cleaning, data transformation, feature engineering, model training, and model evaluation to make the parking prediction expandable to larger areas.

To implement a better system of parking management and to provide real-time predictions on parking availability, we focus on making real-time predictions on the availability of on-street parking meters based on time, day, and month in each parking zone in the Short North area in Columbus.

## 1.3 Thesis Organization

**Chapter 1** introduces the current problem of parking shortage and the need for parking availability prediction, gives an overview of parking occupancy prediction studies based on different data, assumptions, and approaches and describes the major goals of this study.

**Chapter 2** summarizes the benefits of parking availability prediction and discusses previous studies about parking availability prediction from different perspectives.

**Chapter 3** describes the data used in this study and methods of data processing, parking pattern analysis, and predictive modeling.

**Chapter 4** presents the results of parking pattern analysis and parking availability prediction models and evaluates the performance of the models.

**Chapter 5** summarizes the results, findings, and insights from this study and discusses the future directions for model improvement and applications in other areas.

# Chapter 2 Related Work

Much work has been done that contributes to parking availability prediction. In the following sections, firstly, the benefits of providing parking availability are discussed. Secondly, previous modeling methods, data sources, and spatial granularity of parking availability prediction are introduced, summarized, and compared.

## 2.1 Benefits of Parking Availability Prediction

A few existing intelligent parking management systems and related studies show that parking availability information can greatly facilitate online parking reservations and improve customer experience (Wilbur-Smith Associates, 2009). One such example is the parking system, which provides reservation options and real-time parking availability via phones or Internet in Bay Area Rapid Transit in San Francisco or Oakland Metropolitan Area. Studies show that the number of available parking spaces can be an essential factor that affects drivers' parking decisions (Hendricks & Outwater, 1998). Lack of parking availability information may decrease the chance of making a good decision about parking and increase the searching, cruising, and queuing time. However, drivers with parking availability information are 45% more successful in making parking decisions than those without such information at the time of arrival at the destination (Caicedo et al., 2012). Apart from being beneficial to the individual drivers, parking availability information contributes to the whole transportation system by managing the parking demand, reducing traffic congestion, and air pollution (Caicedo et al., 2012).

## 2.2 Approaches to Parking Availability Prediction

Previous studies on parking availability prediction are discussed and compared from different perspectives to provide a comprehensive overview of state-of-art models and approaches.

Many studies apply a theory-driven approach by analyzing individual drivers' parking behavior and simulating the arrival and departure process based on some distribution models to make estimates about parking occupancy status. Caicedo (2012) adopted an aggregated approach to process the parking request and made real-time parking availability predictions based on a discrete choice model at the parking facility in Barcelona. Caliskan et al. (2007) presented a model to predict the parking occupancy of the parking lots based on queuing theory and Markov Chain and evaluated a model based on simulation. Boyles et al. (2015) developed a model to predict the probability of parking availability as a function of searching intensity under an equilibrium framework. Millard-Ball et al. (2014) predicted the parking availability from aggregated parking occupancy data based on a stationary Markovian multiserver queue model and the Poisson distribution. Yang et al. (2017) modeled the parking payment behavior using probability distributions and estimated an aggregated parking occupancy based on parking meter transaction data. Such approaches often rely on a thorough analysis, solid assumptions, and complicate modeling of parking behavior, and the models learned from existing data might not be easily adaptable when parking patterns change due to new circumstances such as road constructions or parking rate changes.

The development of machine learning algorithms, along with the increasing availability and quantity of data, has led to more data-driven methods being applied to prediction or forecasting in the field of transportation. Yang et al. (2003) identified a few factors such as traffic, weather, road conditions, and events and applied principle component analysis and neural networks to predict the availability of parking spaces in the garages. Chen (2014) predicted the parking occupancy in parking lots in San Francisco using different models, including ARIMA, linear regression, support vector regression, and neural networks based on factors like time, day, event, distance, and price. Pflügler et al. (2016) focused on using publicly available data and presented a neural network model to predict the parking availability based on time, day, location, weather, traffic, and users' request data. Shao et al. (2019) proposed an approach by adopting clustering algorithms and recurrent neural network (RNN) with long short-term memory (LSTM) to predict the occupancy rate and parking duration based on sensor data. Yang et al. (2019) considered spatial-temporal correlations between parking occupancy and multi-source data such as traffic-related data in the road networks and applied deep learning models, including the graph neural networks (GNN) and RNN with LSTM to predict block-level real-time parking occupancy. These studies show the great potential in using machine learning and deep learning techniques to identify complicated parking patterns and make parking availability predictions with the abundance of data.

From the perspectives of data sources, many studies use parking sensor data to measure the time-varying parking occupancy (San Francisco Municipal Transportation Agency, 2014; Shao et al., 2019; Vlahogianni et al., 2014). Parking sensors can measure real-time occupancy accurately, but it is expensive to install sensors for all the on-street parking meters, and the data availability relies on real-time data streaming systems. Some other studies estimate parking occupancy based on users' request data with lower operation costs (Caicedo et al., 2012; Nawaz et al., 2013; Pflügler et al., 2016). However, such systems and methods rely on a large amount of user data and may not reflect the real parking information. With the improvement of parking data availability, some studies started to estimate the parking occupancy based on parking transaction data (Yang et al., 2019; Yang & Qian, 2017). Parking transaction data is less expensive than sensor data to obtain and maintain and provide more reliable occupancy information than request data. Additionally, there are two types of predictions based on whether they use real-time data or not. The first type makes current or near-future prediction of parking occupancy based on historical parking data with various input features such as time of day, day of the week, weather and seasonality, which is called offline prediction (Pflügler et al., 2016; Tamrazian et al., 2015). The second type predicts parking occupancy in the near future based on historical and real-time parking data, which is called online prediction (Tamrazian et al., 2015; Yang et al., 2019). Tamrazian et al. (2015) developed prediction models of both types for on-street and off-street parking and made comparisons between the two types of prediction. It is shown from the study that although the online models output more accurate results than offline models, the offline model can successfully make relatively good predictions, outperforming the historical and the previous day's average. Considering the high costs of obtaining real-time data via parking sensors, using an offline model is a cost-effective and feasible approach when there is not enough real-time parking data.

Apart from the modeling methods and data used, the spatial granularity of the prediction should be considered as an important factor affecting the model performance. For on-street parking, parking occupancy is often measured in an aggregated spatial unit. Chen (2014) examined how the aggregation of parking lots may affect the model performance or the prediction error. It shows that aggregating parking lots to a higher level can smooth the pattern and make it more predictable, which indicates higher accuracy. But the higher aggregation level indicates lower spatial granularity and less information. There is also a study showing that effective granularity for the model to be reliable could be different in different areas (Yang & Qian, 2017). Yang et al. (2019) predicted the parking occupancy aggregated to street blocks and proved the feasibility of making a prediction on the block level. The spatial unit used in this project is pre-defined parking zones based on street blocks and regulation rules. There exist parking zones that are smaller than a street block (e.g., when meters in the same street block have different rates or operating hours, they are assigned to different parking zones). The size of a parking zone and the number of meters in a zone vary among the Short North area. The prediction models should be evaluated with considerations of these variations.

# Chapter 3 Data and Methods

This chapter first talks about data collecting, data cleaning, data transformation, and exploratory analysis. Then it introduces the clustering algorithms applied to identify similar parking patterns among different zones. Lastly, it describes the model building, training, and evaluation process of the parking prediction.

## 3.1 Data

The first section covers the data description, data processing, and preliminary analysis of the data. It lays the foundation for the modeling of parking availability.

### 3.1.1 Data Summary

In this study, parking meter transaction data is used to estimate the parking availability. The start and end time of a parking transaction record is assumed to be the arrival and departure times of a vehicle. Parking meter transaction data from 2015 to 2019 and parking meter inventory data in 2017 and 2019 are provided by IPS Group, which is the major parking meter provider in Columbus. Each parking meter transaction record has the following fields: meter ID, start time, end time, transaction type, and total payment. In the parking meter inventory, each record has the following: meter ID, longitude, latitude, and street name. Parking zone information is provided by the City of Columbus. Each record in the parking zone dataset has the following: meter ID, zone ID, latitude, longitude, street name, and area name. Both meter ID and zone ID are unique in the whole system, which makes it possible to link the meters with their zones and aggregate parking meter occupancy to parking zones.

The Short North is the study area of this project. Currently, there are 687 parking meters in the Short North in total. Among them, 45 meters are new, 642 meters are retained, and 201 meters are removed, comparing the meter inventory in 2017 to that in 2019. There are 83 parking zones in the Short North currently. Zones are pre-defined by the City of Columbus for better parking management based on street blocks, block faces, and meter configurations (operating hours, rate, etc.). Parking meters and parking zone distributions in the Short North are shown in Figure 2.1. Different colors represent different parking zones. Most of them are distributed along or near High Street. Table 2.1 shows the summary statistics for the parking transaction duration time in each year from 2015 to 2019. Table 2.2 shows the summary statistics of parking zone size by the number of meters in each zone in the Short North. The zone size varies a lot in the Short North Area, which may lead to huge differences in parking patterns.

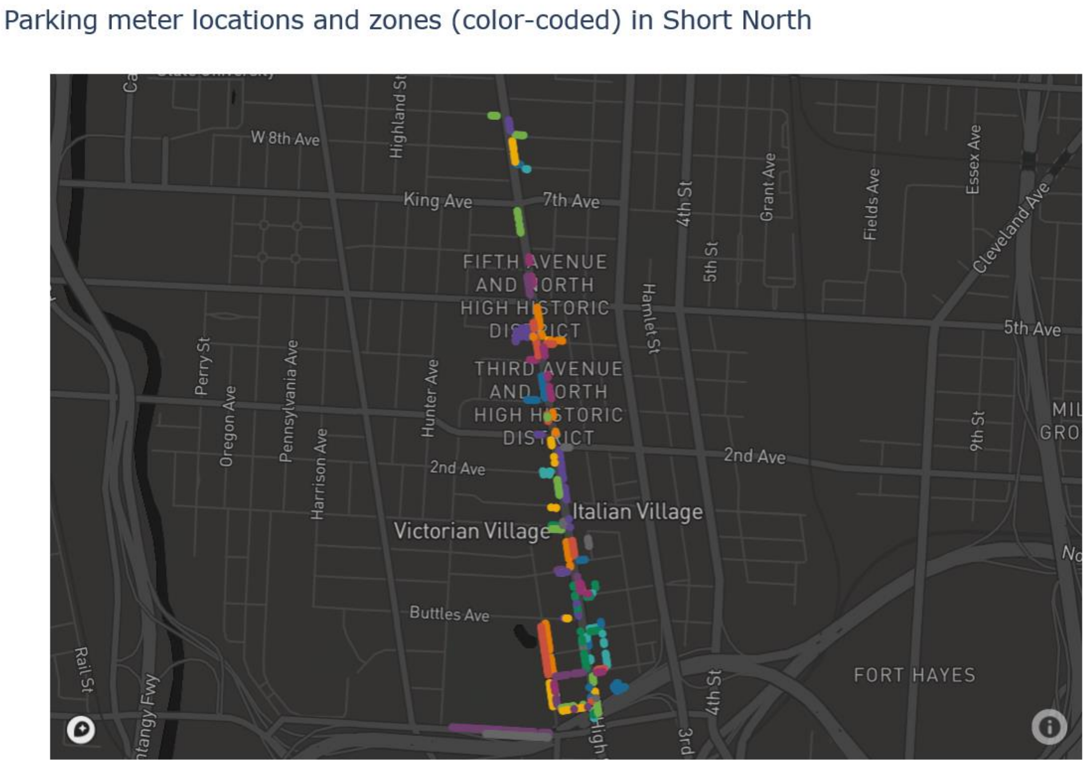


Figure 3.1 Parking Meter Locations with Zones in the Short North in 2019

Table 3.1 Summary Statistics of Parking Meter Transaction Duration Time (Min.) in Each Year from 2015 to 2019

| Year | Count | Max | Min | Mean | Median |
|---|---|---|---|---|---|
| 2015 | 6,465,689 | 783 | 0 | 91 | 80 |
| 2016 | 6,504,351 | 780 | 0 | 98 | 85 |
| 2017 | 6,125,061 | 779 | 0 | 101 | 88 |
| 2018 | 6,089,883 | 778 | 0 | 104 | 91 |
| 2019 | 5,701,536 | 812 | 0 | 101 | 88 |

Table 3.2 Summary statistics of parking zone size (number of meters)

| Count | Mean | Std. Dev. | Min | Median | Max |
|---|---|---|---|---|---|
| 83 | 8.28 | 11.75 | 1.00 | 5.00 | 71.00 |

### 3.1.2 Data Cleaning and Transformation

Data cleaning needs to be done to exclude potential errors from the data production and collection process. Invalid records are excluded if they contain NULL values or invalid date/time, or the same start and end times. The start and end times are rounded to minutes, and the transaction records with duration time smaller than 1 minute or greater than 720 minutes are excluded as these records may not be valid parking and should be considered as outliers. Duplicates are dropped when there are identical records. When there are records with an overlapping time period for the same parking meter, the overlapping part would be excluded. Missing values are handled in this way: if there is no transaction record of a parking meter for a whole month, then this meter would be considered as inactive and be excluded from the total count in its parking zone during that month. Additionally, according to the parking policy, meters in the Short North only need payment from 8:00 am to 10:00 pm Monday to Saturday. On-street parking is free from 10:00 pm to 8:00 am, on Sundays, and city-recognized public holidays. During the free parking time period, transactions occurred cannot reflect the true occupancy level (a small portion of people may still pay for the meters) and are thus excluded from the time series data. Parking predictions on free hours and days are not in the scope of this study. Table 3.1 shows the counts of records before and after cleaning and the dropped ratio.

Meter transactions only record the start and end time and need to be transformed into a measure of parking availability. In this study, parking availability is defined as the proportion of available parking meters out of the total active meters in a parking zone in each half-hour time slot from 8:00 am to 10:00 pm. Each meter transaction record is first converted to the semi-hour availability status. A meter is considered as available in a half-hour time slot if there is no transaction for it during the half-hour. Then the single-meter availability is aggregated to the availability rate in each zone for each semi-hour time slot. Table 3.2 shows examples of the parking availability rate. Data cleaning and transformation are done using the SQL server 2019.

Table 3.3 Counts of Transaction Records before and after Data Cleaning

| **Year** | **Counts before cleaning** | **Counts after cleaning** | **Dropped Percent (%)** |
|---|---|---|---|
| 2015 | 6,465,689 | 5,933,236 | 8.24 |
| 2016 | 6,504,351 | 6,003,503 | 7.70 |
| 2017 | 6,125,061 | 5,654,236 | 7.69 |
| 2018 | 6,089,883 | 5,525,329 | 9.27 |
| 2019 | 5,701,536 | 5,105,345 | 10.46 |
| Total | 30,886,520 | 28,221,649 | 8.63 |

Table 3.4 Examples of Zone-based Parking Availability Rate

| **Zone Name** | **Time slot** | **Available** | **Total** | **Availability** |
|---|---|---|---|---|
| 31024 | 2015-01-16 08:00 – 08:30 | 16 | 17 | 0.94 |
| 31024 | 2015-01-16 08:30 – 08:30 | 16 | 17 | 0.94 |
| 31024 | 2015-01-16 09:00 – 09:30 | 13 | 17 | 0.76 |
| 31024 | 2015-01-16 09:30 – 10:00 | 11 | 17 | 0.65 |
| 31024 | 2015-01-16 10:00 – 10:30 | 6 | 17 | 0.35 |

### 3.1.3 Exploratory Analysis and Visualization

Exploratory analysis is needed to understand the temporal and spatial patterns of parking. Parking availability is calculated for the whole Short North area and aggregated into time slots in a week (10:00 pm to 8:00 am, and Sundays are excluded). There are 28 semi-hour slots each day and six days a week, leading to 168 semi-hour time slots in total in a week. The aggregated weekly time series of parking availability is shown in figure 3.2, and the aggregated daily time series of parking availability is shown in figure 3.3. It shows that there is a daily recurring pattern for the time series. During lunch and dinner time, parking is in high demand and reaches the valley of availability in the Short North Area. Over the week, parking availability is showing an overall decreasing trend from Monday to Saturday. Time of the day and day of the week can significantly affect the availability of parking meters.

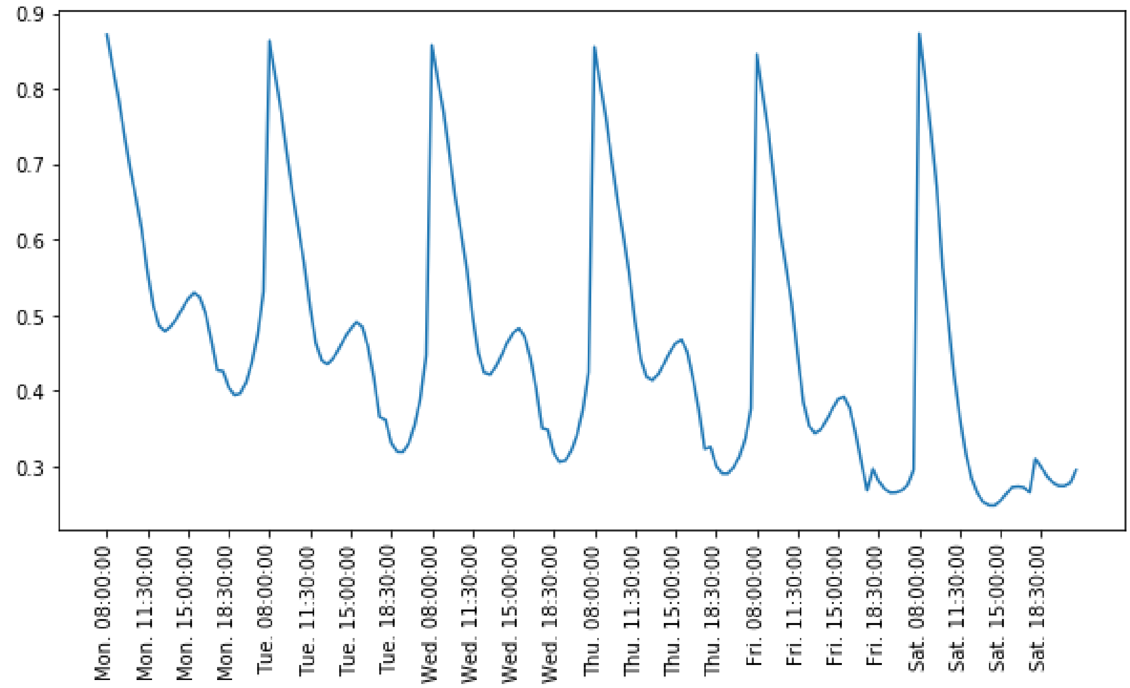


Figure 3.2 Aggregated Weekly Patterns of Parking Availability

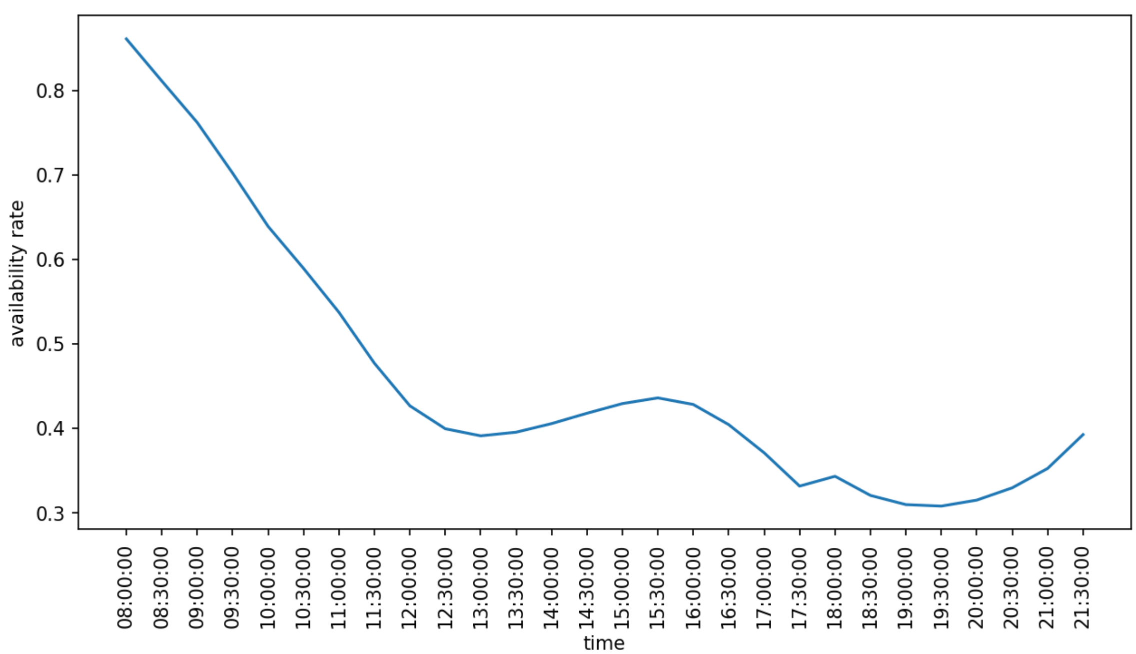


Figure 3.3 Aggregated Daily Patterns of Parking Availability

To test for the seasonality, we calculate the autocorrelation plot in figure 3.4. Autocorrelation is the correlation between a time series and a lagged version of itself. It is important to examine the autocorrelation to identify seasonality. If $T_i$ is the time series and $T_{i-k}$ is the lagged series with $k$ as the time lag, an autocorrelation plot is to plot the autocorrelation between $T_i$ and $T_{i-k}$ for different lag values $k$. Time lag in the x-axis represents the number of time slots that is lagged. Autocorrelation is a value between -1 and 1 and is shown in the y-axis. Higher autocorrelation indicates a higher dependency between the two. When the lag is 28 and the times of 28, the autocorrelation is significantly positive (above the dotted line) and reaches the peak, which indicates the daily recurring pattern is significant or the seasonal period is 28 semi-hour time slots (there are 28 in one day).

To further analyze the seasonality, the daily seasonality is removed by differencing the time series. Subtracting the time series $T_i$ by its lagged series $T_{i-28}$ leads to a new time series $T'_i$. By calculating the autocorrelation plot of this new time series $T'_i$ (figure 3.5), significantly autocorrelation with time lag 168 (or times of 168) is identified, indicating there is a weekly recurring pattern other than the daily pattern, or the seasonal period is 28 × 6 semi-hour time slots (there are 168 in one week).

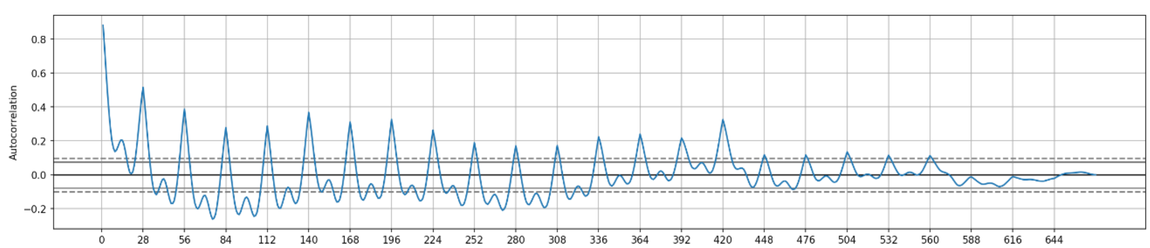


Figure 3.4 Autocorrelation of Parking Availability Time Series

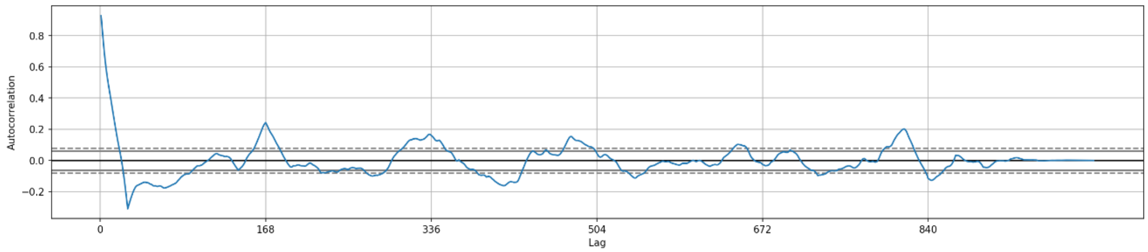


Figure 3.5 Autocorrelation of Differenced Parking Availability Time Series

The yearly trend of the parking occupancy aggregated by month for different years is shown in figure 3.6. For all the years from 2015 to 2018, the parking availability tends to reach a peak in January, June, and October. For the year 2019, the overall parking availability is higher than in other years and does not show a very similar pattern throughout the year. This is because the new parking management system allows people to use mobile apps to pay for the parking from 2019 so that not all transaction is covered by the meter transaction records. The parking occupancy estimated from meter payment only could be lower than the actual rate, and thus the estimated parking availability is higher than it should be. Based on these reasons, only data from 2015 to 2018 will be used in further analysis and modeling.

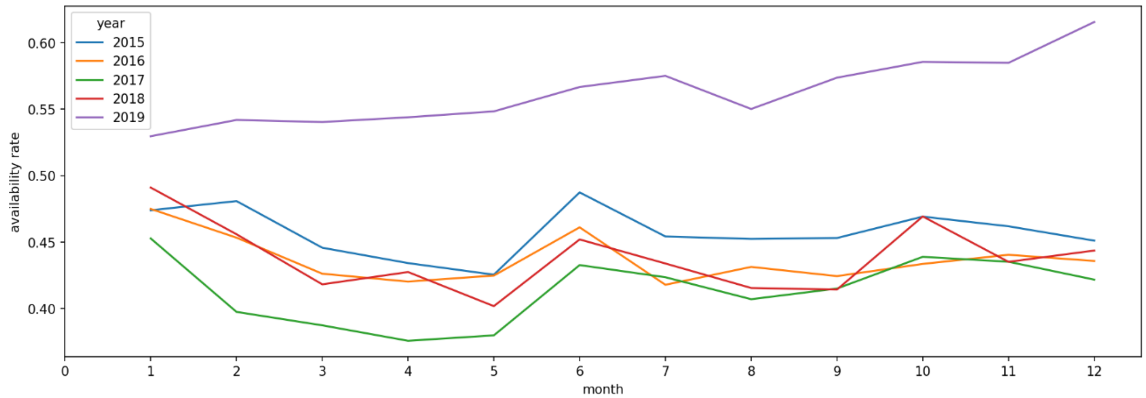


Figure 3.6 Yearly Trend for Parking Availability Aggregated by Month in Different Years

## 3.2 Clustering Algorithms

Studies show that the parking availability of a parking zone may be affected not only by the parking zones nearby but also by those far away (Liu et al., 2017; Wang et al., 2017). There are also some studies applying clustering algorithms to identify similar time-series patterns among different parking units (Chen, 2014; Shao et al., 2019; Tamrazian et al., 2015). Clustering algorithms can group the parking zones to clusters with similar parking patterns, and the parking prediction could be made for each cluster with a similar pattern. Each zone-based parking availability time series is aggregated to an average weekly time series with 168 semi-hour time slots to reduce the noise in the parking pattern. And then, k-means and agglomerative clustering are applied upon these weekly time series of different parking zones, and these two clustering algorithms are compared in clustering time-series data.  Euclidean distance is chosen as a metric to measure the similarity or distance between time series. If time series $T$ is represented as a vector $T = (t_1, t_2, t_3, \ldots, t_k)$

and time series $T' = (t'_1, t'_2, t'_3, \ldots, t'_k)$. The Euclidean distance between $T$ and $T'$ can be calculated as:

$$d_E(T, T') = \sqrt{\sum_i (t_i - t'_i)^2}$$

3.2.1 K-means Clustering

K-means clustering is an iterative clustering algorithm that classifies all the data into k clusters. It starts with k random observations as centroids, and in each iteration, it first assigns all the observations to its closest centroid, and then the new cluster centroids would be recomputed based on the observations that are assigned to each cluster. When there is no more change in centroids' positions, the iteration would be stopped.

K-means algorithm's advantage is that it can handle big data well due to its linear time complexity. However, the k-means algorithm only optimizes the within-cluster variance and often converges to a local optimum instead of the global optimum. The results of k-means depend on the random initial centroids and are hardly reproducible. One common initialization approach is called the Forgy method, which chooses random observations from the data. Another initialization method is called k-means++, which is often used to avoid the poor clustering results by spreading out the initial cluster centroids. It uses a weighted probability distribution to draw the random initial point where the probability of a point been chosen as the next centroid is proportional to the squared distance from it to its closest existing centroid. Moreover, the value of *k* needs prior knowledge and arbitrary decision making. Elbow method, silhouette method, and gap statistic are the common techniques applied to help determine the appropriate *k* value used in k-means. Elbow method and silhouette directly both compute and optimize a metric, which is the total within-cluster sum of squares (WSS) or the average silhouette to find the number of clusters. Gap statistic compares the total within-cluster sum of squares with the expected value under the null reference distribution for different *k* values.

In this study, the Elbow method is used to choose the value of *k*, and k-means++ is used to initialize the centroids. Additionally, the k-means algorithm is repeatedly run a few times with different initializations, and the results are compared in order to examine the effect of random initialization and the optimality of the clustering results.

### 3.2.2 Agglomerative Clustering

Agglomerative clustering is a hierarchical clustering method that tries to build a hierarch of clusters in a bottom-up manner. It treats each observation a singleton cluster and recursively merges pairs of clusters into a new cluster until all the clusters are merged into a single cluster. Based on the way agglomerative clustering works, its result is often shown as a dendrogram, which represents a hierarchy of clusters.

Other than the distance metric, how to choose the pair of clusters to merge at each iteration also depends on the linkage criterion, which determines the distance between two clusters based on pairwise distances of observations. Some common linkage criteria include maximum or complete linkage, minimum or single linkage, average linkage, and Ward's criterion. Maximum (complete) linkage uses the maximum pairwise distance as the cluster distance. Minimum (single) linkage uses the minimum pairwise distance as the cluster distance. Average linkage uses the average pairwise distance as the cluster distance. Ward's criterion, also called Ward's minimum variance criterion, is to choose the pair of clusters, which leads to the minimum increase of total within-cluster variance to merge.

Compared with k-means, agglomerative clustering is more time consuming, but it relies on fewer assumptions about the distribution of data. And it gives a complete picture of the complex relationships between clusters and does not require choosing the number of clusters beforehand. Instead, cutting the dendrogram at a selected height will yield a clustering result with a certain number of clusters. By observing and interpreting the dendrogram, the number of clusters could be picked in a more intuitive way.

In this study, Ward's criterion selected for the purpose of this study because less variance of patterns within each cluster makes parking more predictable. The number of clusters is picked from the dendrogram.

## 3.3 Predictive Models

With different parking zones clustered into different clusters, predictive models of parking availability are developed for the parking zones in each cluster based on the assumption that zones in one cluster share similar parking patterns. The following sections describe the steps of feature engineering, model selection, model training, and evaluation.

### 3.3.1 Feature Engineering

From the exploratory analysis in section 3.1.3, there is a significant daily and weekly seasonality, or a daily and weekly recurring pattern for the parking availability. Also, there is an annual trend that parking availability varies in different months. Categorical features, including time of the day, day of the week and month are considered to be factors affecting the parking availability. One-hot encoding is applied to the data to use the categorical data as input features. One-hot encoding is an encoding technique that maps the categorical value to a few dummy variables with value 0 or 1 to represent the presence of a certain category. There are 28 semi-hour time slots a day, and time of the day is transformed to 27 dummy variables, respectively representing whether the time is during 8:30 - 9:00, 9:00 - 9:30, …, 21:30 - 22:00 or not. There are six days (Sundays excluded) for the week, and day of the week is transformed into five dummy variables, representing whether it is Tuesday, Wednesday, Thursday, Friday, Saturday or not, respectively. There are 12 months and month is transformed to 11 dummy variables, representing whether it is Feb., Mar., ..., Dec. or not. The “time slots” field in the parking availability data (shown in table 3.2) are transformed into these dummy variables, and there are in total 43 binary input features for the prediction models. The output of the model is the parking availability rate, which is a real value between 0 and 1.

### 3.3.2 Model Selection

Different machine learning models, including linear regression, decision tree, and multilayer perceptron, are chosen to make predictions, and the model performance is compared.

Linear regression (OLS) is an algorithm that models the linear relationship between the dependent variable (available rate) and the independent variables (time of the day, day of the week, and month). The formula is shown as follows. The model is fitted to get the least squared error.

$$y_i = \beta_0 + \sum_{k=1}^{27} \theta_k \times [TOD(t_i) = k] + \sum_{p=1}^{5} \lambda_p \times [DOW(t_i) = p]$$

$$+ \sum_{q=1}^{11} \eta_q \times [MON(t_i) = q] + \varepsilon_i$$

Where,

$y_i$ is the available rate of the observation,

$\beta_0$, $\theta_k$, $\lambda_p$, and $\eta_q$, are parameters to be estimated,

$t_i$ is the timestamp of the observation,

$TOD(t_i)$ returns the time of the day of the given timestamp, mapped to an integer from 1 to 27 (8:30-9:00 to 1, 9:00-9:30 to 2, …, 21:30-22:00 to 27)

$DOW(t_i)$ returns the day of the week of the given timestamp, mapped to an integer from 1 to 5 (Tuesday to 1, Wednesday to 2, …, Saturday to 5)

$MON(t_i)$ returns the month of the given timestamp, mapped to an integer from 1 to 11 (February to 1, March to 2, …, December to 11),

$[COND]$ returns 1 if $COND$ is true, 0 if $COND$ is false, and

$\varepsilon_i$ is the error term of each prediction.

Decision tree learning uses a tree to make decisions (predictions) in which the target value is based on input attributes and a set of rules. Decision trees can be used in both classification (target variable is categorical) and regression (target variable is continuous). In decision trees, a non-leaf node is a rule to split the data based on an input feature, a branch is the outcome of the rule or a subset of all data, and a leaf node is a final outcome. In regression, the average outcome of the training data in a leaf node is the predicted value of the node. The algorithm constructs a tree from the top or root, which is the total set of the data. In each step, it selects the best feature and the best cut-off point to split based on a metric, which can be variance for regression or Gini index for classification. The algorithm performs the split recursively until some stop criteria are reached, which could be the minimum number of observations in a non-leaf node or a leaf node. In this study, all the 43 dummy variables are set as input, and a fully unpruned tree is constructed with no early stopping criteria.

Multi-layer perceptron (MLP) is a feedforward neural network model that consists of multiple layers of perceptrons. It at least has one input layer, one hidden layer, and one output layer. Its use of non-linear functions as activation functions makes it able to distinguish non-linearly separable data or learn non-linear models. Backpropagation is used to train the model to minimize the loss function, which is the squared error in MLP regressor. In this study, MLP models with 1 or 2 hidden layers and 50, 100, or 150 hidden units in each layer are trained and evaluated using the 43 dummy input variables. The learning rate is set as 0.001, and the activation function is ReLU (the rectified linear unit function).

For model selection, the data for one cluster from 2015 to 2018 is split into a training set and a test set as a ratio of 7:3. There are 388,374 records in the training set and 166,446 records in the test set. The models are trained on the training set and tested on the test set. Performance metrics for these models are reported. MLP with different hyperparameters (hidden units and hidden layers) are also compared. Root-mean-square error (RMSE) and mean absolute error (MAE) are the metrics used to measure the accuracy of the models. They can be calculated using the following.

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

$$MAE = \frac{1}{n}\sum_{i=1}^{n}|y_i - \hat{y_i}|$$

Where,

$y_i$ is the actual output value and $\hat{y_i}$ is the predicted value.

For a better evaluation of the models and choosing a solid model for prediction, cross-validation is performed. Cross-validation partitions the validation set into several subsets, holds out one set, trains the model on remaining sets, and evaluates on the holdout set. K-fold cross partitions the data into k equal fold and repeats the train-test process when a different fold is held out each time. The average 5-fold cross-validation score is also reported. A higher score indicates overall better model performance. The model with the best overall performance is selected for the prediction.

### 3.3.3 Model Training and Evaluation

Predictive models are built for different clusters. Hence, model training and evaluation are performed independently for each cluster. Parking availability data from 2015 to 2018 are split into training and testing sets in the ratio of 7:3. The models are trained over the training set, tested over test set, and RMSE and MAE are reported for each cluster. The 5-fold cross-validation is also performed for each model trained.

# Chapter 4 Results and Evaluations

This chapter shows the clustering results with two different algorithms and compares these two methods, and then shows the comparisons between different algorithms and parameter settings. The results and evaluations of final models for each cluster are also shown.

## 4.1 Clustering Results

### 4.1.1 Average Weekly Pattern of Parking Availability

Figure 4.1 shows the examples of the average weekly parking pattern for some parking zones. Along x-axis, integers from 1 to 168 are used to represent semi-hour time slots 8:00 – 8:30, 8:30 – 9:00, …, 9:30 – 10:00. It can be found that the weekly patterns vary a lot when the locations and sizes of zones are different. Clustering algorithms are needed to identify similar patterns.

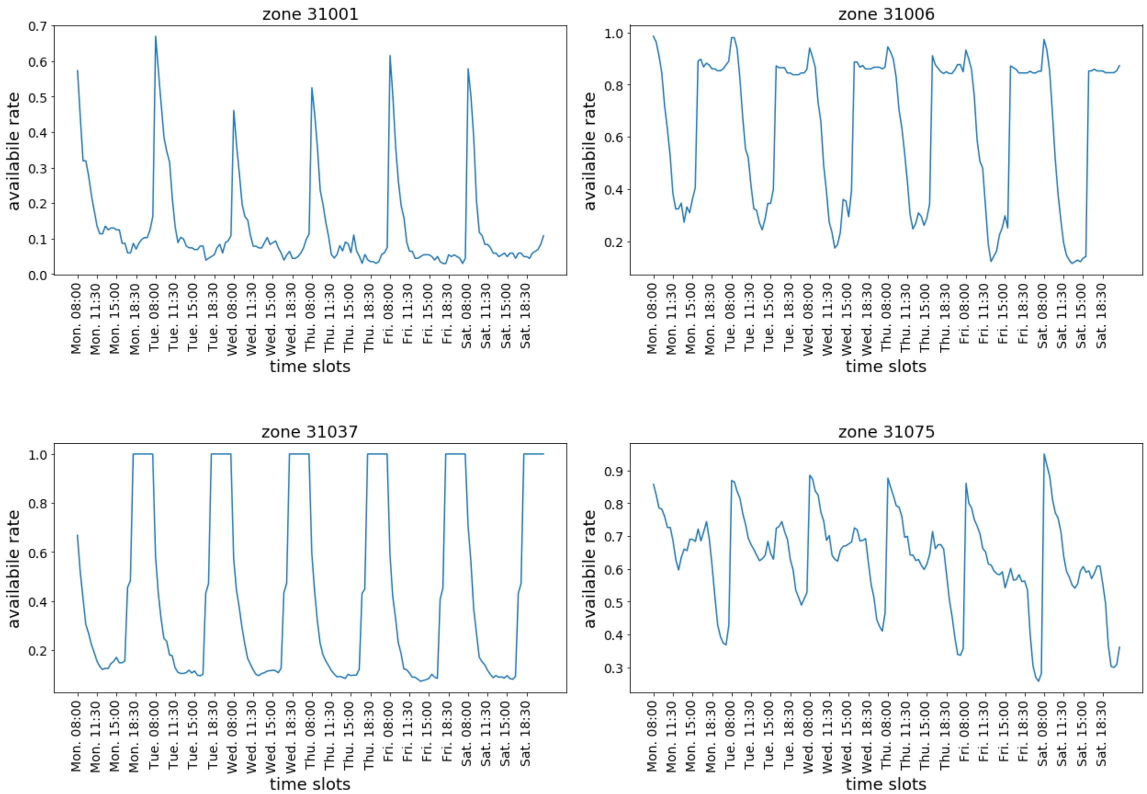


Figure 4.1 Examples of Average Weekly Pattern for Different Parking Zones

### 4.1.2 Results of K-means Clustering

Figure 4.2 shows the curve for the Elbow method in which k value is represented on the x-axis, and the average within-cluster sum of squared error (WSS) is represented on the y-axis. It indicates the appropriate k value should be five as the slope of the curve decreases more slowly after 5. K-means clustering is run with k = 5 over the weekly time series of all the parking zones. Table 4.1 shows the clustering results for the k-means with different initializations. Figure 4.3 shows the clustering result of test #2 with cluster centers in red. Average WSS is used to measure the variance within clusters and the overall goodness of the clustering. It can be shown that although the result in figure 4.3

shows that most of the patterns are close to the centers for each cluster, each run of k-means with different initializations leads to different clustering results, and the goodness is different. It is hard to have a stable clustering result, which is guaranteed to be good.

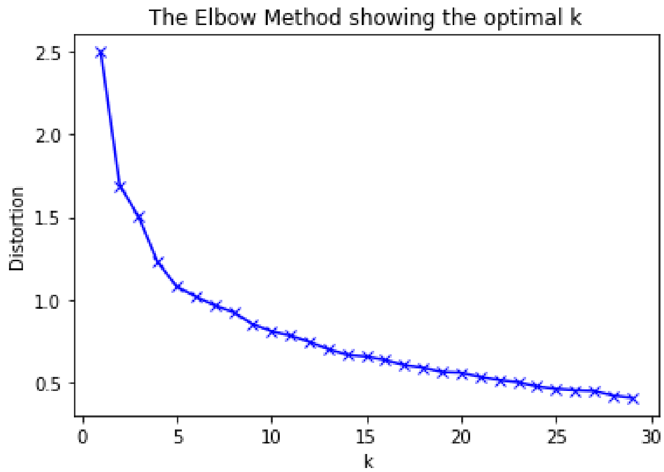


Figure 4.2 Elbow Method

Table 4.1 K-means Clustering Results for Different Initializations

| Test # | Average WSS | Number of Zones Each Cluster |
|---|---|---|
| 1 | 1.1409 | 2, 13, 16, 21, 30 |
| 2 | 1.0795 | 5, 11, 15, 21, 30 |
| 3 | 1.1766 | 3, 5, 12, 21, 41 |
| 4 | 1.0832 | 5, 12, 15, 17, 33 |
| 5 | 1.1599 | 2, 3, 11, 19, 47 |

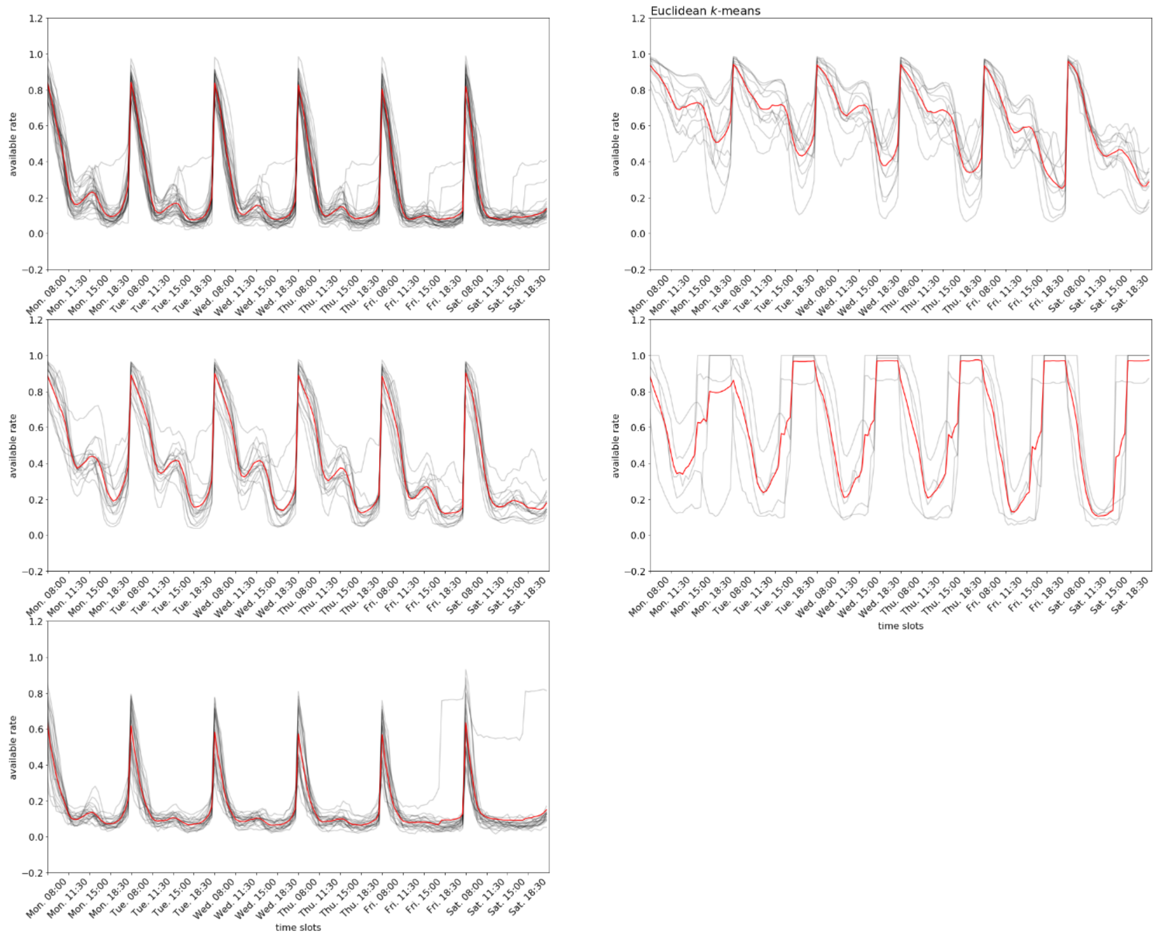


Figure 4.3 Clustering Result (Cluster Centers in Red)

### 4.1.3 Results of Agglomerative Clustering

Agglomerative clustering based on the Ward method and Euclidean distance is shown in figure 4.4. The clustering result is deterministic and shows a whole picture of the hierarchy of relationships between zone patterns. A horizontal line at 4.65 is shown. After observing the dendrogram, this line is selected as the cutoff line, and it ends up with 7 clusters shown in 4.5. Table 4.2 shows the number of zones and statistics of the size of zones in each cluster. It can be shown that in general, larger zones tend to be grouped together (in clusters 1, 4, and 5) while small zones tend to be together (in clusters 3, 6).

However, cluster 2 and 7, there is a big variance of the size of zones within clusters. Figure 4.6 - 4.12 show the patterns for each cluster, respectively. It can be shown that most of the patterns in the same cluster are similar to each other in terms of peak and valley period and shape of the curves. There are only very few exceptions. Among cluster 2, zone 31029 has significantly higher availability than others on the Friday and Saturday nights due to no parking allowed for this zone after 18:00 on these two days. Among cluster 7, the curve of zone 31053 seems to have a shift from others because no parking (full availability) is allowed in zone 31026 and 31006 after 16:00 while no parking is allowed in zone 31053 after 18:00 every day.

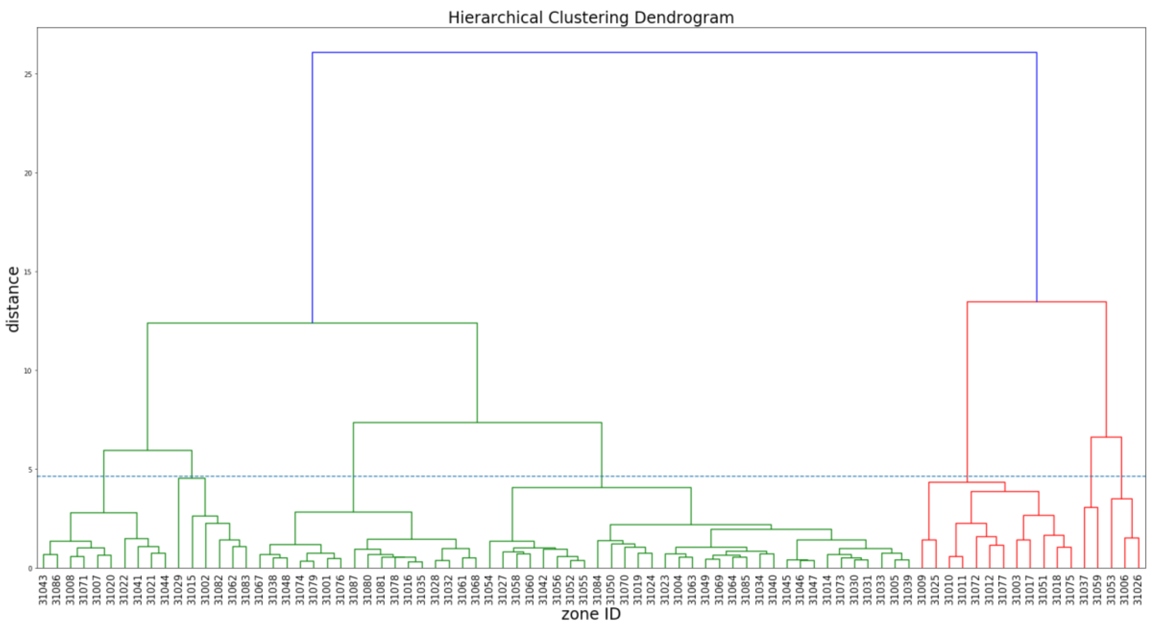


Figure 4.4 Agglomerative Clustering Dendrogram

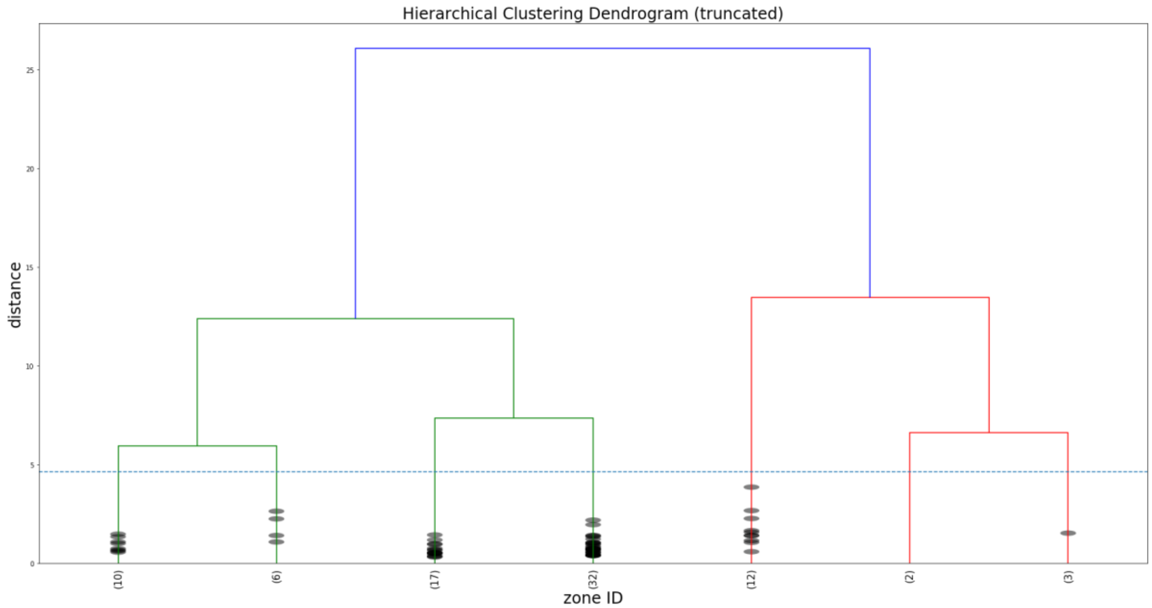


Figure 4.5 Truncated Agglomerative Clustering Dendrogram with 7 Clusters

Table 4.2 Agglomerative Clustering Results with Number of Zones and Size of Zones

| **Cluster ID** | **# Zones** | **Size of Zones (# Meters)** | | | | |
|---|---|---|---|---|---|---|
| | | Mean | Median | Min. | Max. | Total |
| 1 | 10 | 6.6 | 5 | 2 | 13 | 66 |
| 2 | 6 | 6.8 | 2.5 | 2 | 25 | 41 |
| 3 | 17 | 3.9 | 2 | 1 | 10 | 66 |
| 4 | 32 | 7.2 | 6 | 1 | 25 | 231 |
| 5 | 12 | 19.2 | 7.5 | 2 | 71 | 230 |
| 6 | 2 | 2.5 | 2.5 | 2 | 3 | 5 |
| 7 | 3 | 15.3 | 2 | 1 | 43 | 46 |

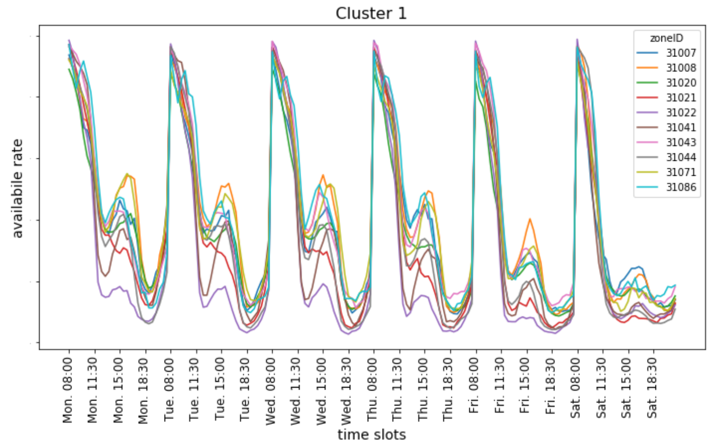


Figure 4.6 Parking Patterns of Cluster 1

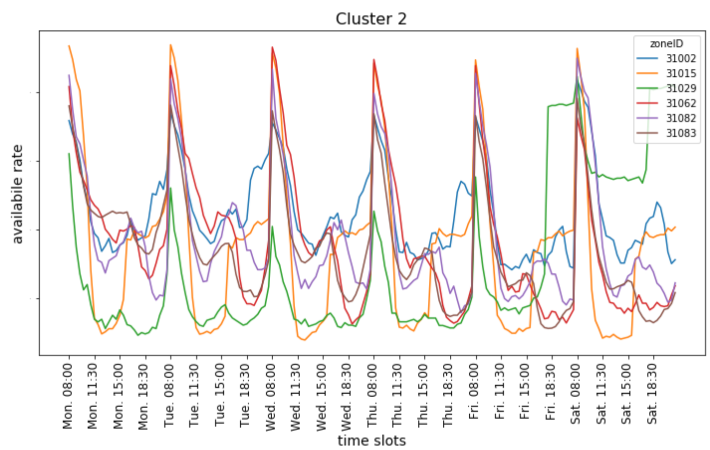


Figure 4.7 Parking Patterns of Cluster 2

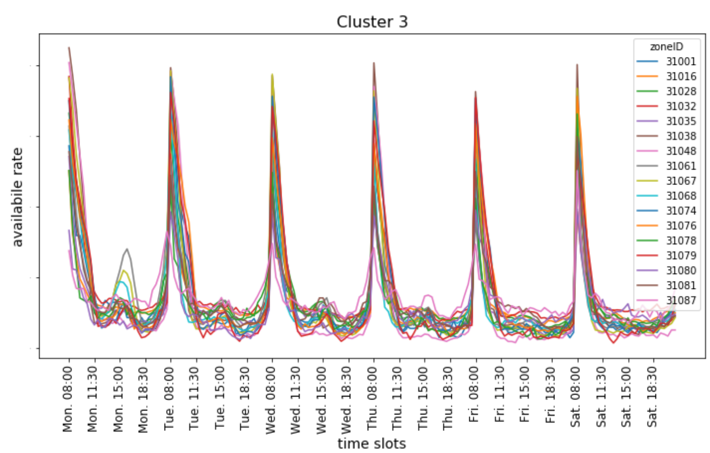


Figure 4.8 Parking Patterns of Cluster 3

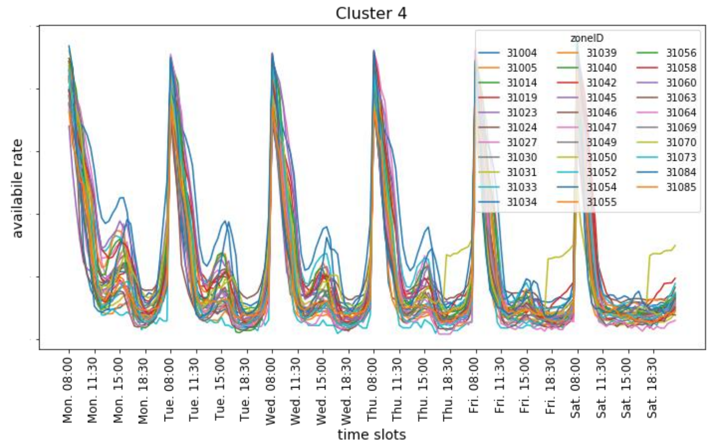


Figure 4.9 Parking Patterns of Cluster 4

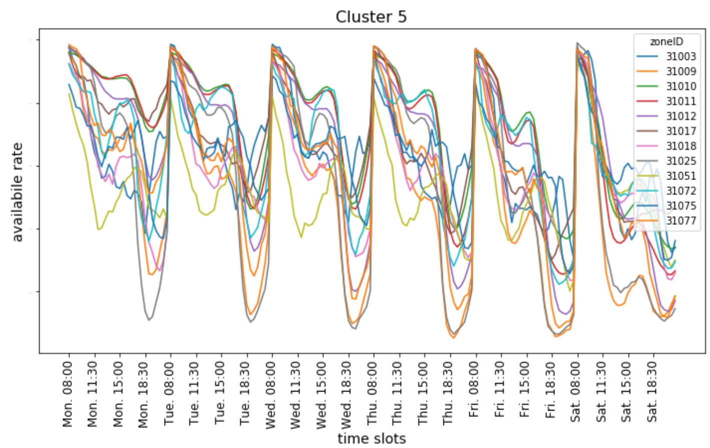


Figure 4.10 Parking Patterns of Cluster 5

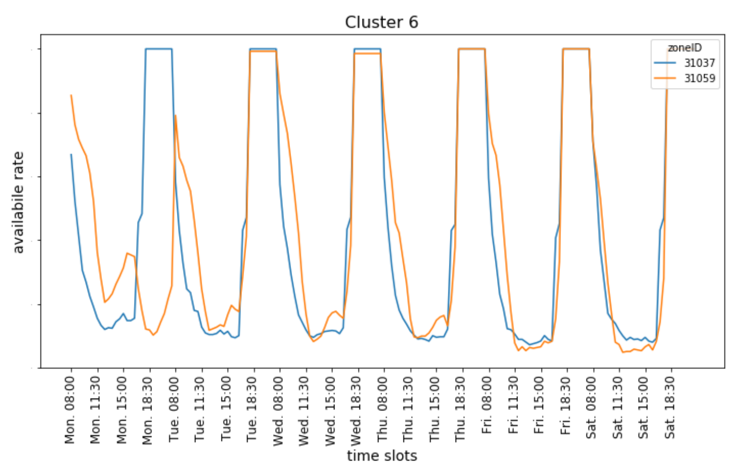


Figure 4.11 Parking Patterns of Cluster 6


Figure 4.12 Parking Patterns of Cluster 7

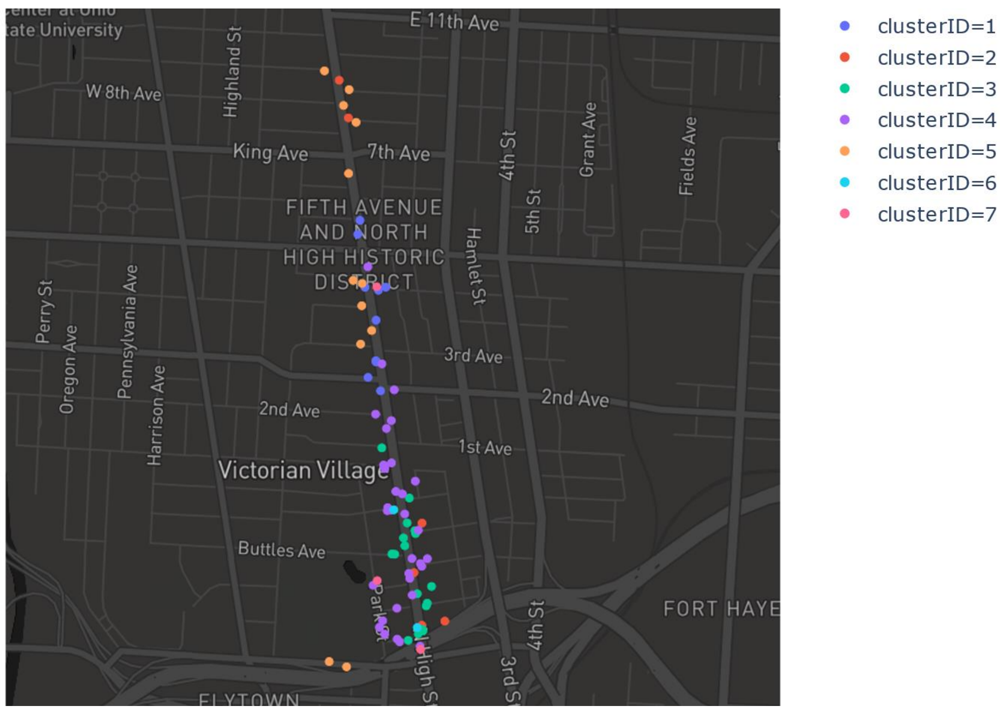


Figure 4.13 Distribution of Clusters of Parking Patterns in the Short North

Figure 4.13 shows the spatial distribution of clusters of parking zones in the Short North. It demonstrates that parking zones may not only be affected by nearby zones but are likely to synchronize with distant zones. Zones in one cluster may not have to be geographically close to each other.

## 4.2 Model Tests and Evaluations Results

### 4.2.1 Model Comparisons

By comparing the metrics of different models with different parameters, the best model could be chosen. As shown in table 3.3, MLP with two hidden layers and 50 hidden units in each layer has the overall best and most robust performance in terms of low error and high cross-validation score. MLP is a flexible model and good at approximating complex non-linear relationships. Hence, MLP with two hidden layers and 50 hidden units in each layer is chosen for the final model training.

Table 4.3 Metrics for Test Data Using Different Models

| | **Hidden Units Each Layer** | **Train RMSE** | **Test RMSE** | **Test MAE** | **CV score** |
|---|---|---|---|---|---|
| Linear Regression | N.A. | 0.2498 | 0.2504 | 0.1659 | 0.1948 |
| Decision Tree | N.A. | 0.2476 | 0.2495 | 0.1648 | 0.1967 |
| Multilayer Perceptron | 50 | 0.2485 | 0.2496 | 0.1690 | 0.1959 |
| | 100 | 0.2475 | 0.2494 | 0.1654 | 0.1956 |
| | 150 | 0.2483 | 0.2496 | 0.1591 | 0.1972 |
| | **50,50** | **0.2480** | **0.2492** | **0.1643** | **0.1985** |
| | 100,100 | 0.2480 | 0.2492 | 0.1654 | 0.1984 |
| | 150,150 | 0.2482 | 0.2492 | 0.1639 | 0.1983 |

### 4.2.2 Model Evaluations

Table 3.4 shows the model training results for each cluster, including the number of samples in train and test set, RMSE for train and test set, MAE for the test set, and CV score. It shows good performance for most of these clusters. Cluster 2 has a higher error and a lower CV score than others. This may be because the parking zones that have ‘no parking’ period at night discussed in 4.1.3 add to the variance within cluster 2 and make parking availability less predictable.

Table 4.4 Model Metrics

| Cluster ID | # Zones | # Train Samples | # Test Samples | Train Set RMSE | Test Set RMSE | Test Set MAE | CV score |
|---|---|---|---|---|---|---|---|
| **1** | **10** | **222,950** | **95,550** | **0.2462** | **0.2484** | **0.1879** | **0.5437** |
| **2** | **6** | **111,249** | **47,679** | **0.3414** | **0.3437** | **0.2906** | **0.1483** |
| 3 | 17 | 388,374 | 166,446 | 0.2480 | 0.2491 | 0.1659 | 0.1996 |
| 4 | 32 | 721,672 | 309,288 | 0.2362 | 0.2380 | 0.1703 | 0.4343 |
| 5 | 12 | 258,092 | 110,612 | 0.2784 | 0.2796 | 0.2189 | 0.3460 |
| **6** | **2** | **46,648** | **19,992** | **0.2628** | **0.2658** | **0.1854** | **0.6335** |
| 7 | 3 | 63,896 | 27,384 | 0.2898 | 0.2952 | 0.2014 | 0.4561 |
| Average | 11.71 | 258,983 | 110,993 | 0.2718 | 0.2743 | 0.2029 | 0.3945 |

# Chapter 5. Conclusions and Discussions

## 5.1 Summary and Conclusions

This project predicts the zone-based on-street meter parking availability based on parking transaction data in the Short North Area in Columbus. Due to the lack of sufficient sensor data for providing real-time information about parking occupancy, parking occupancy, or availability is estimated by the meter transaction start and end time in each zone. Parking availability time series are analyzed, and daily and weekly recurring patterns are identified. Due to the number of parking zones and potential similar parking patterns in the Short North Area, it is inefficient and not necessary to develop a separate model for each parking zone. Instead, clustering algorithms are run to identify groups with similar parking patterns (daily and weekly patterns), and predictive models are made based on clusters. Agglomerative hierarchical clustering has advantages over k-means clustering in this study because it does not need prior knowledge about k, which is the number of clusters and does not rely on any assumptions of data distribution. Moreover, it can output deterministic clustering results and deeper insights about relationships between clusters and avoid the bad results due to random initializations in the k-means. Based on the model comparison, MLP with two hidden layers and 50 units per layer shows the best performance overall and is used to develop models for each cluster.

This project contributes to the existing research by combining agglomerative clustering and machine learning algorithms to make overall good and solid predictive models based on purely historic parking transaction data (offline predictions). It does not rely on expensive sensor data or any real-time communications but demonstrates the potential in estimating aggregated parking occupancy from historic meter transactions.

Additionally, a lifecycle of data collection, data cleaning, data exploration, data transformation, feature engineering, model selection, modeling, and model evaluations can be standardized as a reproducible pipeline when new parking data is available and predictive models need to be updated or rebuilt.

Overall, this project provides a solution to making parking predictions using existing data. The predicted parking availability can help provide better services to the users, mitigate road congestion, and improve the transportation system.

## 5.2 Future Work

### 5.2.1 Model Improvements

The models could be improved, considering the following methods. The ***first*** to use data from different sources. For example, using weather and traffic data as input features for the models might potentially improve the algorithm as parking is highly affected by these factors. The ***second*** is to use real-time data, such as sensor data. Now, Columbus has only about 100 sensors but will install more in the near future. Since parking sensors can provide much more accurate information about meter occupancy status, it can make better estimations about parking availability and provide more up-to-date information. A potential modeling technique is to model the errors or residuals of the current models using real-time or near real-time sensor data in order to improve accuracy. The ***third*** is to make the availability more predictable. There are some small zones in the Short North with only 1 or 2 meters, which does not have a smooth and predictable pattern. Aggregating existing small parking zones to larger zones as prediction units can improve the model but may bring about challenges due to the different regulation policies for different zones in the Short North.  The ***last*** is to improve the algorithms. The current approach selects MLP as the best model to be trained over different clusters. But the model selection decision could be made differently for different clusters as well. Also, ensemble modeling can be applied to combine different machine learning algorithms and make more solid predictions.

### 5.2.2 Application in Other Areas

The prediction approaches can be easily extended to other areas in the City of Columbus to make predictions.  The whole process can be standardized as a pipeline. With the new data, potential new parking patterns could be found. For example, parking demand in the downtown area might be different from that of the Short North, and the peak and valley would be different. But by agglomerative clustering and other machine learning algorithms, the new patterns are learnable and predictable if provided with sufficient data to train the models.

Bibliography


Boyles, S. D., Tang, S., & Unnikrishnan, A. (2015). Parking search equilibrium on a network. *Transportation Research Part B: Methodological*, *81*(P2), 390–409.

Caicedo, F., Blazquez, C., & Miranda, P. (2012). Prediction of parking space availability in real time. *Expert Systems with Applications*, *39*(8), 7281–7290.

Caliskan, M., Barthels, A., Scheuermann, B., & Mauve, M. (2007). Predicting Parking Lot Occupancy in Vehicular Ad Hoc Networks. *2007 IEEE 65th Vehicular Technology Conference - VTC2007-Spring*, 277–281.

Chen, X. (2014). *Parking Occupancy Prediction and Pattern Analysis*.

Hendricks, S., & Outwater, M. (1998). Demand Forecasting Model for Park-and-Ride Lots in King County, Washington. *Transportation Research Record*, *1623*(1), 80–87.

Liu, Y., Liu, C., Lu, X., Teng, M., Zhu, H., & Xiong, H. (2017). Point-of-Interest Demand Modeling with Human Mobility Patterns. *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 947–955.

Millard-Ball, A., Weinberger, R. R., & Hampshire, R. C. (2014). Is the curb 80% full or 20% empty? Assessing the impacts of San Francisco's parking pricing experiment. *Transportation Research Part A: Policy and Practice*, *63*, 76–92.

Nawaz, S., Efstratiou, C., & Mascolo, C. (2013). ParkSense: A smartphone based sensing system for on-street parking. *Proceedings of the 19th Annual International Conference on Mobile Computing & Networking - MobiCom '13*, 75.

Pflügler, C., Köhn, T., Schreieck, M., Wiesche, M., & Krcmar, H. (2016). Predicting the availability of parking spaces with publicly available data. *Informatik 2016*.

San Francisco Municipal Transportation Agency. (2014). *SFpark Pilot Project Evaluation: The SFMTA's Evaluation of the Benefits of the Sfpark Pilot Project*. Retrieved from http://sfpark.org/wp-content/uploads/2014/06/SFpark_Eval_Summary_2014.pdf

Shao, W., Zhang, Y., Guo, B., Qin, K., Chan, J., & Salim, F. D. (2019). Parking Availability Prediction with Long Short Term Memory Model. In S. Li (Ed.), *Green, Pervasive, and Cloud Computing* (pp. 124–137).

Tamrazian, A., Qian, Z. (Sean), & Rajagopal, R. (2015). Where is My Parking Spot?: Online and Offline Prediction of Time-Varying Parking Occupancy. *Transportation Research Record*, *2489*(1), 77–85.

Vlahogianni, E. I., Kepaptsoglou, K., Tsetsos, V., & Karlaftis, M. G. (2014). Exploiting new sensor technologies for real-time parking prediction in urban areas.

*Transportation Research Board 93rd Annual Meeting Compendium of Papers*, 14–1673.

Wang, P., Fu, Y., Liu, G., Hu, W., & Aggarwal, C. (2017). Human Mobility Synchronization and Trip Purpose Detection with Mixture of Hawkes Processes. *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 495–503.

Wilbur-Smith Associates. (2009). *Feasibility study of real time parking information at Metrorail parking facilities -Virginia Stations*. Washington Metropolitan Area Transit Authority.

Yang, S., Ma, W., Pi, X., & Qian, S. (2019). A deep learning approach to real-time parking occupancy prediction in transportation networks incorporating multiple spatio-temporal data sources. *Transportation Research Part C: Emerging Technologies*, *107*, 248–265.

Yang, S., & Qian, Z. (2017). Turning meter transactions data into occupancy and payment behavioral information for on-street parking. *Transportation Research Part C: Emerging Technologies*, *78*, 165–182.

Yang, Z., Liu, H., & Wang, X. (2003). The research on the key technologies for improving efficiency of parking guidance system. *Proceedings of the 2003 IEEE International Conference on Intelligent Transportation Systems*, *2*, 1177–1182 vol.2.