长安大学

二〇二〇届毕业设计

基于交通大数据的出租车时长预测

——以纽约市为例

学　　院：汽车学院

专　　业：汽车运用工程

姓　　名：赵博宇

学　　号：2016904636

指导教师：高扬

完成时间：2020 年 6 月

二〇二零年六月

长安大学

Bachelor Graduation Thesis

Taxi Trip Duration Prediction Based on Traffic Big Data: A Case Study of New York City

by

Boyu Zhao

u2016904636

Supervisor: Prof. Yang Gao

School of Automobile

June 2020

Abstract

In the modern era, cities have emerged as hubs of transportation development, with transportation serving as the lifeblood of urban areas. The efficiency of urban transportation systems has always been a major concern in urban planning. Taxis, as a crucial component of urban public transportation, have a significant impact on the overall functioning of urban transport networks. The length of taxi trips, to a great extent, reflects the state of urban traffic infrastructure. However, accurately predicting the optimal and most time-efficient routes for taxis, as well as estimating the duration of taxi trips, remains a challenging task. This article focuses on researching the prediction of taxi trip duration in New York City. The main objectives of this study are as follows:

Firstly, the existing trip duration data of New York City taxis was analyzed to verify its authenticity and eliminate any missing, duplicate, or invalid data. After an initial cleaning process, the data were visualized and analyzed for correlations, primarily considering factors such as time, number of passengers, and coordinates. Through this analysis, key variables influencing trip duration were identified.

Secondly, the advantages and disadvantages of different machine learning algorithms were analyzed and a prediction algorithm was developed. In this thesis, the XGBoost algorithm was used to study the prediction performance under different parameter combinations, and the optimal parameter combination was selected. Based on the algorithm and its parameter combination, a taxi trip duration prediction algorithm was designed.

Finally, the cleaned and processed data was fed into the XGBoost algorithm specifically designed for this study to conduct prediction and validation, ultimately obtaining the results. The performance of the prediction was evaluated using appropriate evaluation metrics, with the root mean square error (RMSE) of the predicted results calculated as 0.36923.

Following the above steps, this model successfully predicts the duration of taxi trips based on the available data. The experimental results are deemed satisfactory and provide valuable insights for future research and applications.

Keywords: data cleaning, machine learning, duration prediction, boosting

Contents

装 订 线

Chapter 1 Introduction

1.1. Background

Since the beginning of the new century, urbanization and the number of family cars have experienced a significant increase, resulting in a substantial rise in the number of motor vehicles. Technological advancements have brought convenience to people's daily commute. However, the existing urban transportation infrastructure struggles to meet the growing demands of travelers. Traffic congestion has become a severe issue in major cities worldwide, leading to frequent traffic accidents that endanger the safety of urban residents and impede economic development. Research indicates that vehicles emit higher levels of emissions when driving at low speeds during congestion. For instance, when the speed is below 5km/h, the exhaust NOx emission coefficient is 7; whereas, when the speed exceeds 35km/h, the coefficient drops to 3. This demonstrates the significant impact of traffic conditions on the economy, environment, and overall urban development issues, as well as the happiness of residents.

In this context, predicting the running time of transportation modes has become a key concern in traffic infrastructure development. It holds immense potential and practical value as a novel approach to enhance urban transportation construction. Taxis, being both convenient and fast, serve as crucial components of urban public transportation and are among the most frequent users of urban roads. Their diverse roles determine their significance in urban transportation, and the efficiency of taxi trips reflects the rationality and scientific planning of urban transportation design. The problems faced by taxis in their daily trips represent areas for future improvement in urban transportation construction.

This thesis selected New York City, one of the largest and most complex cities globally in terms of traffic conditions, as the research subject. By studying taxi running hours in the city and utilizing machine learning techniques, the thesis aimed to predict taxi trip hours using innovative research methods. Additionally, it explored prediction methods that have universal applicability for estimating the operating hours of various transportation modes.

New York City, situated on the east coast of the United States, stands as the country's primary port and holds the distinction of being the most populous and renowned city in the nation, as well as globally. The city's population is approximately 8.51 million people, which can reach 20 million when considering the entire metropolitan area. The total area of New York City spans about 12,144.4 square kilometers. Given the enormous population

and urban area, it is easy to comprehend the immense pressure faced by urban road traffic in the city.

From an administrative perspective, New York City is divided into five county boroughs: the Bronx, Brooklyn, Manhattan, Queens, and Staten Island. Manhattan serves as the city's downtown area and forms the focus of the study in this thesis. The city exhibits a unique layout, primarily consisting of one-way streets with a high volume of pedestrian and vehicular traffic. The presence of disjointed and narrow roads results in intermittent traffic congestion, imposing significant strain on New York City's transportation system. This detrimental traffic condition leads to severe economic losses, substantial environmental damage, and a decrease in the quality of travel, overall quality of life, and happiness of the city's residents. Consequently, it also affects the efficiency and quality of transportation within Manhattan.

With its massive population and expansive urban area, New York City's urban transportation system faces complexity and stress unparalleled by most other cities. It has progressively become the most congested city in the United States, witnessing a constant influx of people and traffic during peak hours and holidays. The movement of people and vehicles traversing downtown Manhattan and the five boroughs resembles a tidal wave. As depicted on the map, New York City comprises separate islands interconnected by a limited number of bridges. Particularly in Manhattan, the city's central district, the streets follow a distinct design, primarily consisting of one-way roads arranged in a checkerboard pattern. Avenues, named from the west to east as the first to twelfth avenue (with several non-numbered avenues interspersed), bear the primary responsibility of traffic flow in the Manhattan district. The condition of these avenues significantly influences traffic quality and efficiency within Manhattan. In addition to the avenues, east-west streets are numbered from south to north, with Fifth Avenue serving as the symmetrical center dividing the east and west. These streets play a crucial role in connecting the avenues and alleviating the heavy traffic pressure they bear. Notably, Broadway Avenue, which cuts diagonally across Manhattan, presents a different arrangement compared to the orderly pattern of avenues and streets. While this road layout imparts a sense of orderliness to the city, it often leads to severe traffic problems, hindering travel efficiency. While Uptown and Midtown primarily adopt the checkerboard design, Downtown, located south of Hairston Street, features streets with individual names, lacking specific naming rules. Consequently, concentrated travel demand in these areas leads to congestion on major arterials and bridges. This results in wastage of resources and reduced travel efficiency.

Given such challenging road traffic conditions, rational urban transportation planning is of utmost importance to facilitate swift journeys to destinations. Taxis, as integral components

of urban public transportation, carry a significant volume of traffic. Ensuring their smooth and efficient trip is crucial for maintaining the stability and timeliness of urban transportation. The following section provides a brief overview of the taxi situation in New York City.

Yellow taxis are a common sight in the New York area, and to operate on the road, they require licensing from the New York City Taxi and Limousine Commission. Private companies are responsible for the trip and management of these taxis. The New York City area boasts approximately 13,000 taxis. Taxi fares vary depending on the time of day. Typically, the starting fare is $2.50, with an additional charge of $0.40 per mile. Starting from 8 p.m., the fare increases to $3.00, while during holiday rush hours (4 p.m. to 8 p.m.), the starting fare rises to $3.50. In addition to these standard costs, passengers are responsible for additional charges when taxis encounter traffic jams or pass through tolled bridges and tunnels. The high cost of taxi services reflects the high demand for urban road transport. Given the substantial demand for travel and the complex road conditions, the pressure and responsibilities faced by the New York City transportation system are immense. The intricate road design poses challenges for conducting smooth research. When confronted with a complex and extensive dataset, the task at hand is to fully, scientifically, and effectively utilize the data to extract the relevant variables that influence taxi operating hours accurately.

In light of this background, this thesis employed New York City as a case study. By analyzing existing data and designing a taxi trip duration prediction algorithm, the thesis aimed to forecast taxi operating hours in New York City.

1.2. Domestic and international development status

In recent years, urbanization and population growth have led to a significant demand for upgrading city infrastructure. However, the existing old and inefficient urban transportation systems are struggling to meet the rapidly increasing traffic demand. Unreasonable traffic planning has resulted in problems such as congestion, accidents, and resource waste, which hinder economic development. Therefore, the construction of modern transportation facilities is urgently needed.

The prediction of operating hours for transportation vehicles has received little attention in both domestic and international research. Traditional research methods often prove inadequate for the complex real-world conditions, necessitating more convenient, accurate, and advanced research tools. The emergence and development of artificial intelligence has provided new ideas and methods for predicting transportation operating hours. The advantages of high speed and intelligence offered by artificial intelligence can significantly

reduce the workload of researchers and uncover overlooked areas or correct misconceptions from past experiments.

This thesis aimed to use machine learning as a method to predict the time required for a taxi to complete a journey. The predictions could inform urban road traffic planning and provide scientific and reasonable suggestions for residents' travel, helping them make decisions to reduce unnecessary trip duration. It then promoted urban transportation development and reduced unnecessary economic losses. Additionally, this model can serve as a framework for analyzing and researching the running time of urban transportation, providing valuable insights for transportation planning by identifying facilities that lag or have unreasonable designs.

However, since there is little involvement in this field at home and abroad, this section introduces short-time traffic flow prediction methods, artificial intelligence techniques and time series data prediction, which have been developed in the industry. In the field of short-time traffic flow prediction, researchers have proposed research methods as follows:

The literature [7] proposed an improved model for traffic flow prediction based on least squares support vector machines.

The literature [8] argues that chaos exists in traffic states, and this thesis uses chaos and fractal theory to recover the dynamical system of traffic flow sequences and uses multivariate local prediction method to predict the time series and collects data in the field to apply the model for analysis and verification, and achieves a high prediction accuracy.

A hybrid AGO-SVM approach for highway traffic flow prediction was proposed in the literature [9].

In the thesis [10], a multi-method combination prediction model based on Bayesian network is proposed for the uncertain information problem in short-time traffic flow prediction, and its prediction accuracy is better than that of a single prediction model.

The literature [11] proposes a support vector machine prediction model with wavelet analysis. The model firstly performs wavelet decomposition of traffic flow to smooth the traffic flow, obtains the high-frequency and low-frequency parts of the traffic flow signal, uses a support vector machine for prediction, and finally reconstructs the prediction results of the high-frequency and low-frequency parts using wavelets to obtain the final prediction value.

The literature [12] used a cuckoo search algorithm to optimize the BP neural network parameters for the short-time prediction model.

The literature [13] combines the idea of integrated learning to improve the GMDH

algorithm.

In the literature [14], for the characteristics of short-time traffic flow change periodicity and randomness, time and spatial sequence traffic observations are selected as the training samples of support vector machine for training, and the spatial sequence prediction values are used to correct the traffic flow time series prediction results, and its impact on future prediction is dynamically adjusted by analyzing the historical time and space series prediction results.

In the literature [15], a traffic flow prediction method by vehicle type is proposed for the problem of prediction accuracy, and good prediction accuracy is achieved experimentally.

The data used in this thesis cover the entire first half of 2016, including temporal, seasonal, and holiday variables, which are essential for time series data prediction. Time series data forecasting involves acknowledging the development trend of the forecast object through statistical and mathematical analysis of known data, considering the influence of chance factors that produce randomness and can lead to errors in predictions. To eliminate the influence of random fluctuations, mathematical processing and statistical analysis of known data are performed to obtain stable time data. Trend forecasting is then conducted to forecast future values.

Time series data forecasting is a category of quantitative forecasting, and as a regression forecasting method, its basic idea of forecasting is: firstly, to acknowledge the continuity of the development of the forecast object, which is mainly through the statistical and mathematical analysis of the known time series data, so as to obtain the time series development trend; secondly, to also comprehensively consider the influence of chance factors on the time series data, which will produce randomness, and this part Chance factors can make the prediction results produce instability and uncertainty, making the prediction results produce errors. In the process of time series data forecasting, the influence generated by random fluctuations should be eliminated, and the specific method is to perform certain mathematical processing through statistical analysis of known data to clean off the random data of chance and obtain cleaned and stable time data. Trend forecasting is then performed after cleaning off this random factor.

In studies targeting time series data forecasting, the academic community has traditionally considered machine learning algorithms as having no advantage. In Kasun Bandara, Christoph Bergmeir et al.'s article "Forecasting across time series databases using recurrent neural networks on groups of similar series: A clustering approach", it is argued that complex methods do not necessarily produce better forecasting results than simple methods when forecasting time series data. In the M3 prediction competition, machine

algorithms did not produce as good results as most statistical algorithms. In the NN3 and NN5 competitions held specifically for knapsack learning, only one machine learning algorithm outperformed the damped exponential smoothing method, and none of the methods were able to outperform the simple exponential smoothing method. It is evident that traditional machine learning algorithms still have great room for improvement in the field of time series data prediction. However, with the rapid development of artificial intelligence, researchers in the field of machine learning continue to develop and improve time series data prediction algorithms. Machine learning prediction of time-series data has become an effective tool.

While the discipline of artificial intelligence emerged early, the concept of machine learning came a little later, with the "Turing Test" proposed in 1950 by Alan Turing, the British scientist known as the father of the world's computers. Then, until the beginning of 1986, machine learning began a period of rapid development. During the development process, machine learning has experienced two major advances. The first advance came in 1986. The "shallow learning", or BP neural network algorithm, proposed by Rumelhart and McClelland and others, which solved the XOR Gate problem, aroused worldwide attention, and the world began to research on shallow learning. In this process, many machine learning algorithms emerged, such as iterative algorithms, support vector machine algorithms, etc. These algorithms have promoted great progress in many scientific and industrial fields. At the same time, a special type of RNN algorithm proposed by Hochreiter et al. is the long-term memory network algorithm LSTM, which can effectively learn long-term dependencies and is a highly efficient sequential model. The second advance is deep learning. Deep learning as a concept first appeared in 2006 by Hinton et al. and has opened the research on deep learning since then. Deep learning provides new ideas for solving optimization challenges related to deep structures. In the 2012 ImageNet Large Scale Visual Recognition Challenge, the AlexNet model based on Deep Convolutional Neural Networks (DCNN) proposed by Hinton et al. reduced the Top 5 error rate of the "image classification" task to 15.3%, which was much better than the second place's 26.2%. The industry's interest in deep learning was greatly triggered. This world record was broken in 2013 by a deep CNN designed and optimized by Fergu et al. with an error rate reduced to 11.7%. The error rate was reduced to 3.6%. In 2016, Google's AlphaGo Go system, which combines Monte Carlo tree search and deep neural networks, defeated former Go world champion Lee Sedol, causing widespread and far-reaching discussions around the world; and in 2017, the upgraded version of AlphaGo defeated Ke Jie, the world's top-ranked Go professional at the time; in the same year, Oxford University, in collaboration with DeepMind, released LipNet, a model that maps variable-length video sequences into text, which combines convolutional neural networks with LSTM recurrent

neural networks to achieve automatic lipreading at the utterance level, and in the GRID corpus, the model achieved 93.4% accuracy, a result that far exceeded the then best accuracy of 79.6%. In 2018, the Google AI team created a new bidirectional language model, BERT, which differs from other language representation models by pre-training deep bidirectional representations mainly by coming to jointly adjust the context in all layers, and the method achieved a record of 11 natural language processing (NLP) tasks. Since 2012, deep learning has been evolving rapidly, with new revolutionary advances and breakthroughs in many different areas. In just a few years, deep learning has become the most forward-looking machine learning approach in the industry today, helping researchers in various fields such as computing, mechanics, civil engineering and biology to overcome difficulties and achieve new breakthroughs.

Machine learning technology is evolving rapidly, and new algorithms have improved the limitations of previous methods. Researchers now have a broader range of choices for predicting time series data. Machine learning algorithms can not only mine existing features but also discover new relevant features. With reasonable parameter tuning, machine learning can provide more accurate and usable results. Machine learning has flourished in various fields and has become an indispensable tool for scientific research.

Traditional research methods, such as simulation, fieldwork, and questionnaires, are costly, inefficient, and often produce superficial data, making it challenging to obtain accurate and scientific conclusions. Given the long construction period and social impact of transportation projects, it is crucial to employ scientific methods and rigorous analysis to avoid erroneous conclusions that could lead to significant economic losses. Therefore, careful consideration and the use of more scientific analysis methods are necessary for traffic-related issues.

Hourly forecasting is an essential aspect of transportation construction, as construction is based on demand. Scientific methods are crucial in forecasting taxi durations, which serve as representatives of urban public transport and often reflect multifaceted issues. Although there is an abundance of data in this field, past research methods have struggled to handle the complexity of the data, limiting their utilization and potential. The emergence and development of data analysis and machine learning disciplines have provided a more scientific approach to analyzing and extracting insights from vast amounts of data. Machine learning reduces human effort and produces comprehensive research results, making the conclusions more intuitive and easily understandable. Thus, since the birth of machine learning, it has been widely applied not only in trip duration prediction but also in traffic flow prediction and traffic safety, with practical success.

In this thesis, we employed the XGBoost algorithm of machine learning to design a model

for predicting taxi durations in New York City. Considering the limitations of experimental conditions and the focus on a single vehicle type, we build upon previous methods and experiences to develop this model.

1.3. Technology roadmap

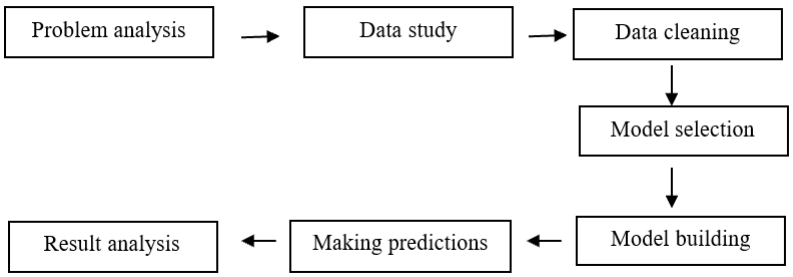

Figure 1 Technology roadmap

As depicted in the figure above， the technical approach adopted in this research follows the subsequent steps: initial data observation, data cleaning to eliminate unreasonable, duplicated, and invalid data, data preprocessing to obtain statistically manageable data, and visualization of the processed data for further analysis. Variables relevant to hour prediction are selected through the steps, and dummy variables are generated for unique one-hot encoding. An algorithm for predicting taxi trip durations is designed, debugged, and subjected to multiple simulations to identify the optimal parameter combination. The final dataset, after one-hot encoding, is inputted into the machine learning model, and the parameters are tuned to make the ultimate prediction. Evaluation metrics are designed to assess the results, and the conclusion is derived accordingly.

The first chapter of this thesis serves as an introduction, providing background information on the topic, discussing the development of short-time traffic flow prediction research both domestically and internationally, and presenting the research methodology employed in this thesis. The second chapter focuses on problem analysis and data pre-processing, outlining the composition, sources, and types of data used in the project, as well as the research approach. The latter half of the thesis is dedicated to the cleaning of temporal, spatial, and ridership data in the dataset. The cleaned data is then analyzed and visualized, and variables are uniquely encoded while removing irrelevant and duplicated ones, resulting in a dataset suitable for the final machine learning process.

The third chapter delves into taxi trip duration prediction, including the design of a taxi trip duration prediction algorithm. In this thesis, the XGBoost algorithm is utilized for this purpose. The previously pre-processed data is inputted, and after a series of parameter tuning, the algorithm is executed to obtain the best parameter combination, finalizing the

装 订 线

model and generating the prediction results. The obtained results are evaluated using the root mean square error, with positive evaluation outcomes signifying the achievement of the experiment's objectives. The fourth chapter comprises the conclusion. The fifth and sixth sections consist of acknowledgments and references, respectively.

Chapter 2 Problem analysis and data preprocessing

2.1. Data introduction

The objective of this problem is to predict the length of taxi trips in New York City. Starting from taxis, the data is observed and studied from the existing taxi trip status data, cleaned and processed to predict the future city taxi trip. This project uses the 2016 New York City taxi trip status as an example. The data was collected by the New York City Taxi and Limousine Commission and integrated and provided by Kaggle. The data includes sample set, training set, and test set.

- Training set: a total of 1458644 trip records.
- Test set: a total of 525,134 trip records.

	A	B	C	D	E	F	G	H	I	J	K
1	id	vendor_id	pickup_datetime	dropoff_datetime	passenger_count	pickup_longitude	pickup_latitude	dropoff_longitude	dropoff_latitud	store_and_fwd_flag	trip_duration
2	id2875421	2	2016/3/14 17:24	2016/3/14 17:32	1	-73.98215485	40.76793671	-73.96463013	40.76560211	N	455
3	id2377394	1	2016/6/12 0:43	2016/6/12 0:54	1	-73.98041534	40.73856354	-73.9994812	40.73115158	N	663
4	id3858529	2	2016/1/19 11:35	2016/1/19 12:10	1	-73.97902679	40.7639389	-74.00533295	40.71008682	N	2124
5	id3504673	2	2016/4/6 19:32	2016/4/6 19:39	1	-74.01004028	40.7199707	-74.01226807	40.70671844	N	429
6	id2181028	2	2016/3/26 13:30	2016/3/26 13:38	1	-73.97305298	40.79320908	-73.97292328	40.78252029	N	435
7	id0801584	2	2016/1/30 22:01	2016/1/30 22:09	6	-73.98285675	40.74219513	-73.99208069	40.74918365	N	443
8	id1813257	1	2016/6/17 22:34	2016/6/17 22:40	4	-73.96901703	40.7578392	-73.95740509	40.76589584	N	341
9	id1324603	2	2016/5/21 7:54	2016/5/21 8:20	1	-73.96927643	40.79777908	-73.92247009	40.76055908	N	1551
10	id1301050	1	2016/5/27 23:12	2016/5/27 23:16	1	-73.9994812	40.73839951	-73.98578644	40.73281479	N	255
11	id0012891	2	2016/3/10 21:45	2016/3/10 22:05	1	-73.98104858	40.74433899	-73.97299957	40.78998947	N	1225
12	id1436371	2	2016/5/10 22:08	2016/5/10 22:29	1	-73.98265076	40.76383972	-74.00222778	40.73299026	N	1274
13	id1299289	2	2016/5/15 11:16	2016/5/15 11:34	4	-73.99153137	40.74943924	-73.95654297	40.77062988	N	1128
14	id1187965	2	2016/2/19 9:52	2016/2/19 10:11	2	-73.96298218	40.75667953	-73.98440552	40.7607193	N	1114
15	id0799785	2	2016/6/1 20:58	2016/6/1 21:02	1	-73.95630646	40.76794052	-73.96611023	40.76300049	N	260

Figure 2 Train set data header

	A	B	C	D	E	F	G	H	I
1	id	vendor_id	pickup_datetime	passenger_count	pickup_longitude	pickup_latitude	dropoff_longitude	dropoff_latitude	store_and_fwd_flag
2	id3004672	1	2016/6/30 23:59	1	-73.98812866	40.73202896	-73.99017334	40.75667953	N
3	id3505355	1	2016/6/30 23:59	1	-73.96420288	40.67999268	-73.95980835	40.65540314	N
4	id1217141	1	2016/6/30 23:59	1	-73.99743652	40.73758316	-73.98616028	40.72952271	N
5	id2150126	2	2016/6/30 23:59	1	-73.95606995	40.77190018	-73.98642731	40.73046875	N
6	id1598245	1	2016/6/30 23:59	1	-73.97021484	40.76147461	-73.9615097	40.75588989	N
7	id0668992	1	2016/6/30 23:59	1	-73.99130249	40.74979782	-73.98051453	40.78654861	N
8	id1765014	1	2016/6/30 23:59	1	-73.97830963	40.74155045	-73.95207214	40.71700287	N
9	id0898117	1	2016/6/30 23:59	2	-74.01271057	40.70152664	-73.98648071	40.71950912	N
10	id3905224	2	2016/6/30 23:58	2	-73.99233246	40.73051071	-73.87561798	40.87521362	N
11	id1543102	2	2016/6/30 23:58	1	-73.99317932	40.74876022	-73.97930908	40.76131058	N
12	id3024712	1	2016/6/30 23:58	4	-73.96852875	40.67843246	-73.96659088	40.63571167	N
13	id3665810	2	2016/6/30 23:58	1	-73.98277283	40.75690842	-73.9746933	40.75333023	N
14	id1836461	1	2016/6/30 23:58	1	-73.92110443	40.76729202	-73.93685913	40.77404404	N
15	id3457080	2	2016/6/30 23:57	1	-73.98680115	40.73491669	-73.97589874	40.75689316	N

Figure 3 Test set data header

As seen from the above figure, the test set lacks dropoff_datetime and trip_duration variables compared with the training set. That is, the duration of the taxi trip to be predicted by the experiment. The data in the training and test sets are classified and described as follows.

- Id: the unique identifier of each trip.

- Vendor_id: the code of the taxi operator providing the trip, with two operators, 1 and 2.
- Pickup_datetime: The date and time when the meter was started, accurate to the second.
- Dropoff_datetime:The date and time, to the second, when the meter was stopped. This data is not available in the test set.
- Passenger_count: Number of passengers in the taxi (value entered by the driver).
- Pickup_longitude: Longitude at the time the meter was started.
- Pickup_latitude: latitude at the time the meter was started.
- Dropoff_longitude: the longitude at which the meter was stopped.
- Dropoff_latitude: the latitude at which the meter was stopped.
- Store_and_fwd_flag: indicates whether the trip record is stored in the onboard recorder before sending to the server. There are two states, Y means stored and N means not stored.
- Trip_duration: Trip duration (in seconds). This data is not available in the test set.

The above is the introduction of each variable in the dataset. In the following, the above dataset is observed, cleaned, processed and visualized in order to extract the characteristics of the variables that affect the prediction of taxi trip duration.

2.2. Data analysis

Since there may be various defects in the dataset, such as duplicates, missing values, residual values, invalid values, etc., these data are not only useless for the final taxi duration prediction results, but also affect the quality of the prediction results. At the same time, for machine learning topics, the data need to be treated as raw materials, and the success of the project results depends heavily on the quality of the data and how they are processed. Therefore, in this thesis, the data is first processed to obtain cleaned data, after which accurate and valuable results can be obtained when machine learning is performed. Once all the data are prepared, they are transformed into a format suitable for use by duration prediction algorithms. This stage includes processes such as filtering, aggregation, input and transformation. The data processing method chosen at this stage depends on the type of data, the data processing library and the type of algorithm. Data cleaning is a pre-requisite for subsequent work, and only data that have been purposefully cleaned are of research value, and subsequent conclusions based on that data are meaningful. Therefore, this section focuses on the specific analysis of each variable in the dataset and the methods and processes chosen to process the data for this project.

2.2.1. Data review

Firstly, in Figure 4 and Figure 5, various statistical measures are calculated, such as count, mean, standard deviation (std), minimum (min), 25th percentile, median, 75th percentile, and maximum values, for each variable in the training and test sets, respectively.

	vendor_id	passenger_count	pickup_longitude	pickup_latitude
count	1.458644e+06	1.458644e+06	1.458644e+06	1.458644e+06
mean	1.534950e+00	1.664530e+00	-7.397349e+01	4.075092e+01
std	4.987772e-01	1.314242e+00	7.090186e-02	3.288119e-02
min	1.000000e+00	0.000000e+00	-1.219333e+02	3.435970e+01
25%	1.000000e+00	1.000000e+00	-7.399187e+01	4.073735e+01
50%	2.000000e+00	1.000000e+00	-7.398174e+01	4.075410e+01
75%	2.000000e+00	2.000000e+00	-7.396733e+01	4.076836e+01
max	2.000000e+00	9.000000e+00	-6.133553e+01	5.188108e+01

	dropoff_longitude	dropoff_latitude	trip_duration
count	1.458644e+06	1.458644e+06	1.458644e+06
mean	-7.397342e+01	4.075180e+01	9.594923e+02
std	7.064327e-02	3.589056e-02	5.237432e+03
min	-1.219333e+02	3.218114e+01	1.000000e+00
25%	-7.399133e+01	4.073588e+01	3.970000e+02
50%	-7.397975e+01	4.075452e+01	6.620000e+02
75%	-7.396301e+01	4.076981e+01	1.075000e+03
max	-6.133553e+01	4.392103e+01	3.526282e+06

Figure 4 Training set data overview

	vendor_id	passenger_count	pickup_longitude	pickup_latitude
count	625134.000000	625134.000000	625134.000000	625134.000000
mean	1.534884	1.661765	-73.973614	40.750927
std	0.498782	1.311293	0.073389	0.029848
min	1.000000	0.000000	-121.933128	37.389587
25%	1.000000	1.000000	-73.991852	40.737392
50%	2.000000	1.000000	-73.981743	40.754093
75%	2.000000	2.000000	-73.967400	40.768394
max	2.000000	9.000000	-69.248917	42.814938

	dropoff_longitude	dropoff_latitude
count	625134.000000	625134.000000
mean	-73.973458	40.751816
std	0.072565	0.035824
min	-121.933327	36.601322
25%	-73.991318	40.736000
50%	-73.979774	40.754543
75%	-73.963013	40.769852
max	-67.496796	48.857597

Figure 5 Test set data overview

Figure 4 provides an overview of the training set data, while Figure 5 presents an overview of the test set data. From these figures, it can be observed that the distribution trends of the training and test sets are relatively similar, except for the count item. Notably, the training set contains a significantly larger amount of data compared to the test set.

To further investigate the dataset for residual values, missing values, or invalid values,

Figure 6 and Figure 7 illustrate the data volume in the training and test sets, respectively.

```
RangeIndex: 1458644 entries, 0 to 1458643
Data columns (total 11 columns):
id                  1458644 non-null object
vendor_id           1458644 non-null int64
pickup_datetime     1458644 non-null object
dropoff_datetime    1458644 non-null object
passenger_count     1458644 non-null int64
pickup_longitude    1458644 non-null float64
pickup_latitude     1458644 non-null float64
dropoff_longitude   1458644 non-null float64
dropoff_latitude    1458644 non-null float64
store_and_fwd_flag  1458644 non-null object
trip_duration       1458644 non-null int64
dtypes: float64(4), int64(3), object(4)
memory usage: 122.4+ MB
```

Figure 6 Training set

```
RangeIndex: 625134 entries, 0 to 625133
Data columns (total 9 columns):
id                  625134 non-null object
vendor_id           625134 non-null int64
pickup_datetime     625134 non-null object
passenger_count     625134 non-null int64
pickup_longitude    625134 non-null float64
pickup_latitude     625134 non-null float64
dropoff_longitude   625134 non-null float64
dropoff_latitude    625134 non-null float64
store_and_fwd_flag  625134 non-null object
dtypes: float64(4), int64(2), object(3)
memory usage: 42.9+ MB
```

Figure 7 Test set

Based on the above figures, it can be concluded that there are no missing values in the dataset, and the number of data points for each type is the same in both the training and test sets. Consequently, further analysis can be conducted.

However, due to the dataset's enormous size, it becomes challenging to visually examine the relationship between variables and the factors influencing taxi runtime. Therefore, additional observation using mathematical tools is necessary in subsequent sections.

The dataset includes several features, such as id, vendor_id, pickup_datetime, dropoff_datetime, passenger_count, pickup_longitude, dropoff_longitude, pickup_latitude, dropoff_latitude, store_and_fwd_flag, trip_duration, among others. Each of these variables is analyzed as follows.

- Traffic conditions vary throughout the day, with morning and evening peaks, as well as distinctions between weekdays and weekends. Furthermore, traffic conditions can be affected by factors such as months and seasons. For instance, slippery roads during winter can lead to lower traffic speeds and longer trip durations. Such information can be extracted from the pickup_datetime variable.
- The number of passengers also impacts taxi trips, as an increase in passenger count may result in more stops, thereby increasing the trip duration. This information is captured by the passenger_count variable.
- The vendor_id and store_and_fwd_flag variables indicate the presence of two taxi companies, labeled as 1 and 2.
- The latitude and longitude variables, such as pickup_latitude, pickup_longitude, dropoff_latitude, and dropoff_longitude, can be utilized for clustering and calculating the operating distance and driving direction simultaneously.

The above analysis pertains to each feature of the acquired data. In the subsequent sections, the author conducted a specific analysis of these variables to examine their effects on trip durations. The data items that do not meet the experimental requirements, as observed in the data, are cleaned and analyzed to extract variables relevant to taxi trip duration prediction.

2.2.2. Time data cleaning

During the initial steps, it was observed that some passenger counts in the training set exceeded 6, while others were 0. Additionally, the minimum trip duration was 1 second, and the maximum was 3,526,828 seconds. Such anomalous data needs to be removed. In temporal data processing, the following cleaning rules are applied:

1) The end time of the meter is earlier than the start time.
2) Trip time and id duplicate or null values.

3) The number of passengers is 0 or greater than 5.
4) Extreme data points are eliminated by removing values that deviate from the mean by two standard deviations.

After removing all the data above, the initial cleaning of the time and number of passengers results in the dataset.

Figure 8 illustrates the extreme values of boarding and alighting times in the cleaned training set and test set.

```
2016-06-30 23:59:39
2016-01-01 00:00:17

2016-06-30 23:59:58
2016-01-01 00:00:22

2016-06-30 23:59:58
2016-01-01 00:00:17
```

Figure 8 Extreme values of pick-up and drop-off times

Based on Figure 8, the latest nighttime value in the training set is 23:59:39 seconds on June 30, 2016, while the earliest boarding time is 0:17 seconds on January 1, 2016. In the test set, the latest nighttime value is 23:59:58 seconds on June 30, 2016, and the earliest boarding time is 0:22 seconds on January 1, 2016. The overall latest ride time for both the training and test sets is 23:59:58 seconds on June 30, 2016, and the earliest ride time is 0:17 seconds on January 1, 2016. Thus, the temporal data meets the experiment's requirement for the year 2016.

By applying the cleaning rules above to the time and number of passengers, the preliminary processed data is obtained. The subsequent analysis focuses on the spatial data.

2.2.3. Spatial data cleaning

The _longitude and _latitude terms exist in the training set and test set, respectively. hese variables provide crucial information for analyzing the direction and distance of taxi trips, making them essential features in the taxi duration prediction model. To ensure the data's accuracy and relevance to the prediction results, it is necessary to clean this part of the data based on the known latitude and longitude coordinates of New York City.

We know that New York City is situated between 40.63° and 40.85°N latitude and 73.75° and 74.03°W longitude. Hence, the data should be inspected to determine if the boarding

and alighting locations exhibit anomalies and if the distribution is consistent between the two datasets.

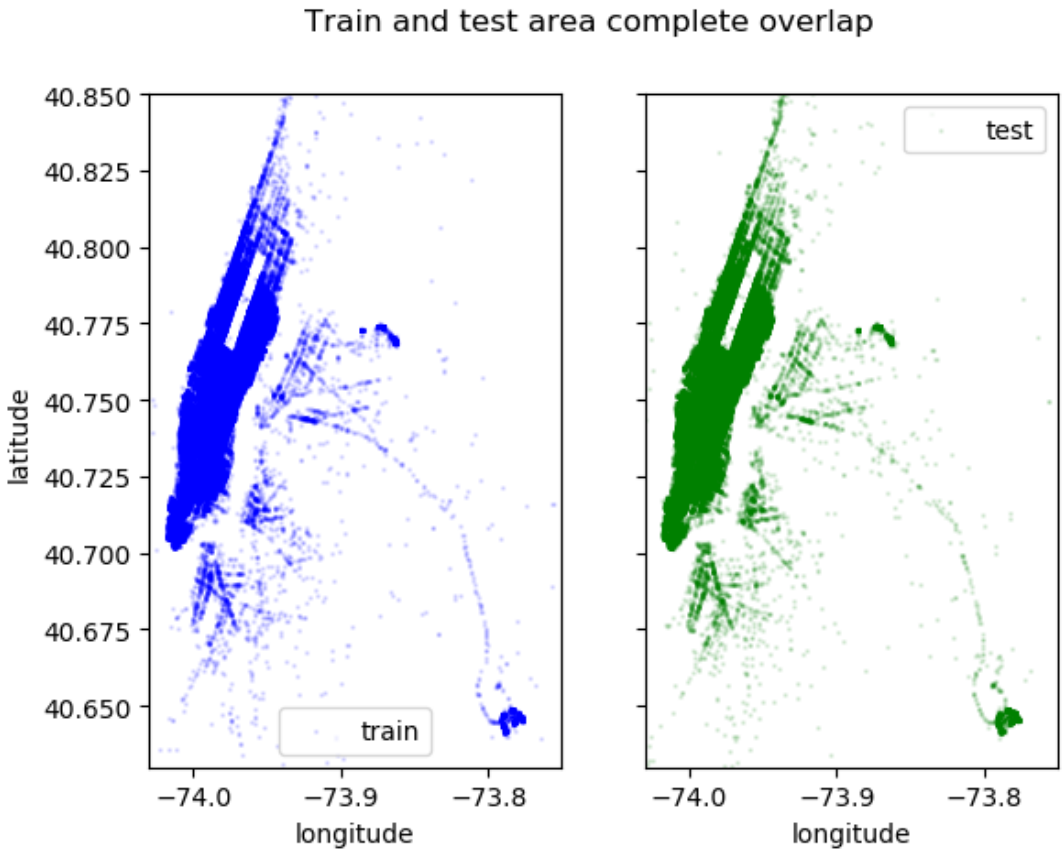

Figure 9 The distribution of coordinates in training set and test set.

Next, the maximum and minimum values of latitude and longitude for boarding and alighting locations are examined, as shown in Figure 10.

```
51.88108444213867    43.92102813720703
34.359695434570305   32.1811408996582
-61.33552932739258   -61.33552932739258
-121.93334197998048  -121.9333038330078
```

Figure 10 Extreme values of pick-up and drop-off coordinates

As shown in Figure 10, in the training set, the maximum value of the coordinates of the pick-up coordinates is about 51.88°N, 61.33°W, and the minimum value is about 34.35°N, 121.93°W. The maximum value of the coordinates of the drop-off location is 43.92°N, 61.33°W, and the minimum value is 32.18°N, 121.93°W. It is evident that these extreme values deviate from the geographical range of New York City. Therefore, data with

significant deviations need to be eliminated. By referring to the map of New York City, data located outside the geographical range are removed.

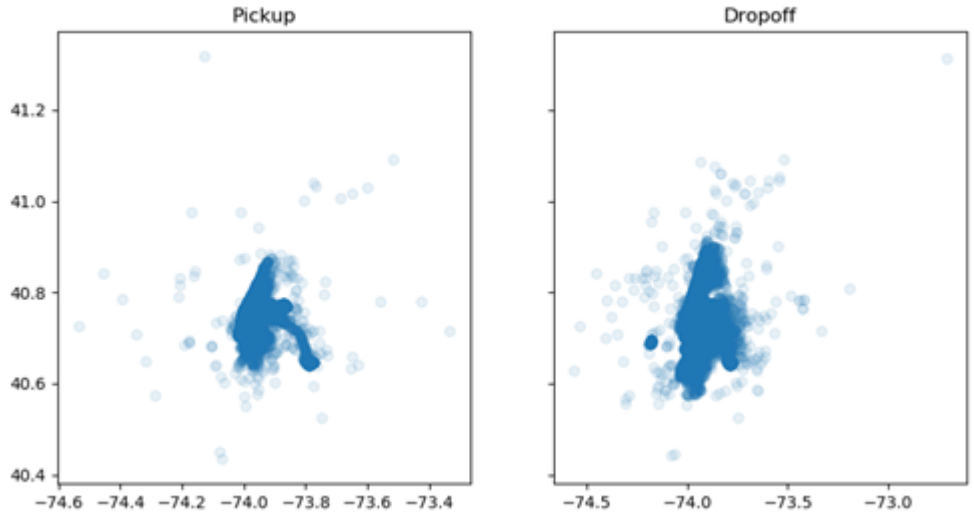

Figure 11 Distribution of the pick-up and drop-off locations in training set

Figure 11 illustrates the distribution of pick-up and drop-off locations in the training set after mapping. Although the data is relatively concentrated, some pruning is still necessary to enhance the effectiveness of model training. Certain trip records are associated with excessively long durations and are far from the concentration distribution area. To prevent data with large deviations from affecting the prediction results, outliers beyond the range of 0.001% and 99.999% are deleted. This process helps improve the training effectiveness. The region is then replotted and enlarged to observe changes in the distribution of outlier points, as depicted in Figure 12. After removing several outliers, the data becomes more concentrated. Thus, the cleaning of spatial data in the training set is completed. The figure allows for observation of the distribution of latitude and longitude in the training set.

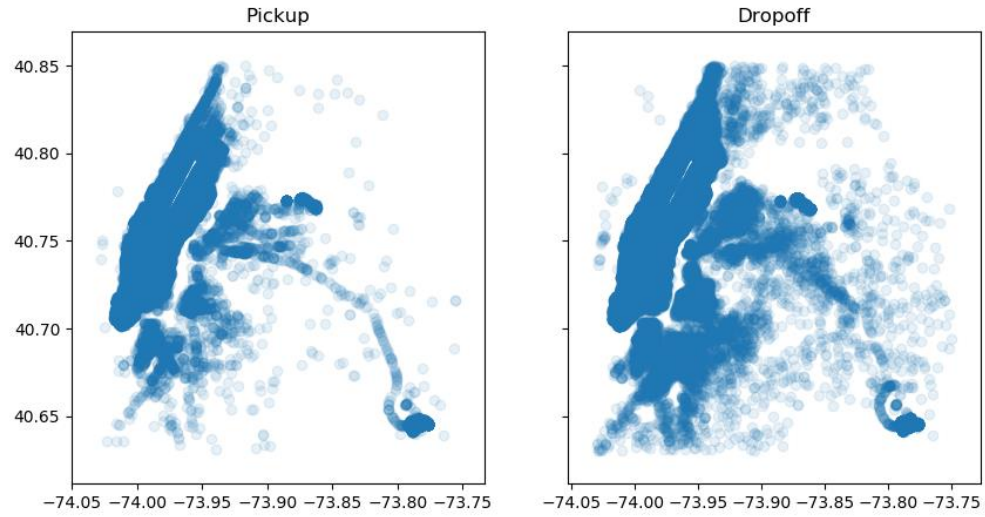

Figure 12 Distribution of the pick-up and drop-off locations after cleaning

With the completion of spatial data cleaning, it is important to note that only the coordinates of pick-up and drop-off are insufficient for duration prediction. Therefore, the cleaned coordinate dataset requires mathematical calculations and visualization in the subsequent sections to derive features related to taxi trip duration.

2.3. Data pre-processing and visualization

The initially cleaned time, ridership, and spatial data above are still complex and diverse, and it is difficult to discern the variables that ultimately affect taxi hours based on numbers alone. Further processing of these data is needed to filter the relevant variables. The visualization processing in this section consists of the following main parts.

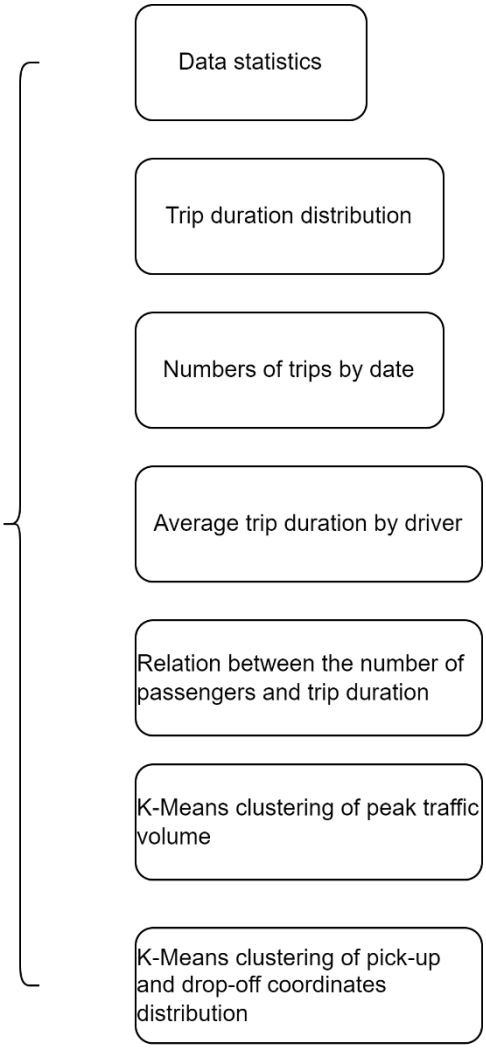

Figure 13 Visualization

2.3.1. Time data pre-processing

In this section, the data pre-processing performed on the given dataset. Due to the limited information in the dataset, only rough dates are available. However, considering that traffic volumes vary across different days and times, treating all data equally will not provide an objective analysis. K-means algorithm was used to identify daily peak hours and compared them with the regular 24-hour trip data. By incorporating the cleaned data from the first step, the final time data could be obtained.

First, the distribution of trip durations was plotted in the figure below.

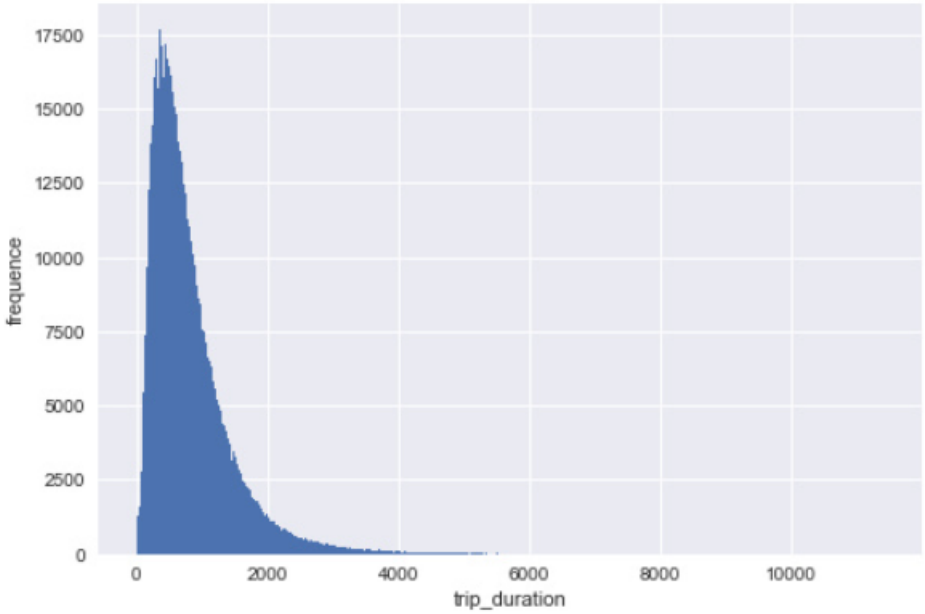

Figure 14 Statistical distribution of trip duration

Figure 14 illustrates the statistical distribution of trip duration for taxi trips in the training set. The horizontal axis represents the trip duration, and the vertical axis represents the frequency. It can be observed that the trip durations are primarily concentrated in the range of 0 to 2000, with the highest frequency within this interval. The data roughly follows a power law distribution, which is not suitable for direct statistical analysis. To address this, the logarithmic transformation was applied to the data, resulting in a distribution that approximates a normal distribution.

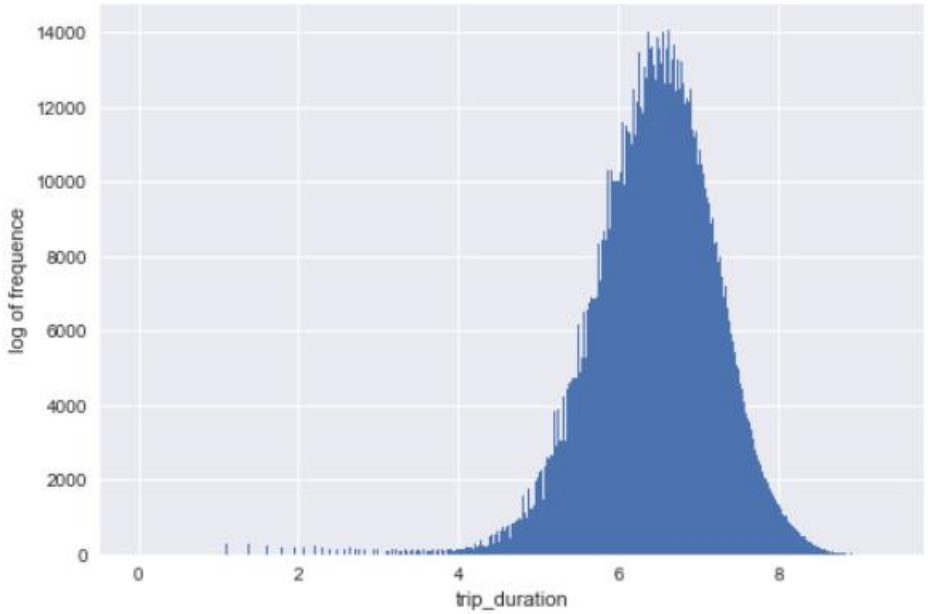

Figure 15 Logarithmic processing of trip duration

As seen in Figure 15, after logarithmic transformation, the taxi trip durations basically obey normal distribution, which is convenient for the observation.

Considering the long statistical period and potential variations in taxi trip durations at different times and seasons, the distribution of trips across different dates was examined.

Figure 16 Numbers of trips by date in training set and test set

装 订 线

Figure 16 shows the distribution of the numbers of trips by date in the training and test sets. It can be found that the trend in the training and test sets is similar. This suggests that the data from both sets can be considered uniformly and hold analytical value. However, anomalies are noticeable at the end of January and May, corresponding to significant declines in the number of trips. After investigation, it was found that:

1) January 23 and 24, 2016. The United States was hit by a snowstorm.
2) May 30, 2016. Memorial Day in the United States (the last Monday of May).

It can be concluded that the data changes are reasonable and can be further processed.

Third, considering that there are two drivers in the dataset, since there may be differences in route choice, driving speed, etc., between the drivers, to account for potential differences in driving styles between two drivers, their working hours were compared. Figure 17 displays the average trip durations of the two drivers, indicating that Driver 2 spends more time on average than Driver 1. To incorporate this information into the model, we introduce a dummy variable for driver identification during subsequent analysis.

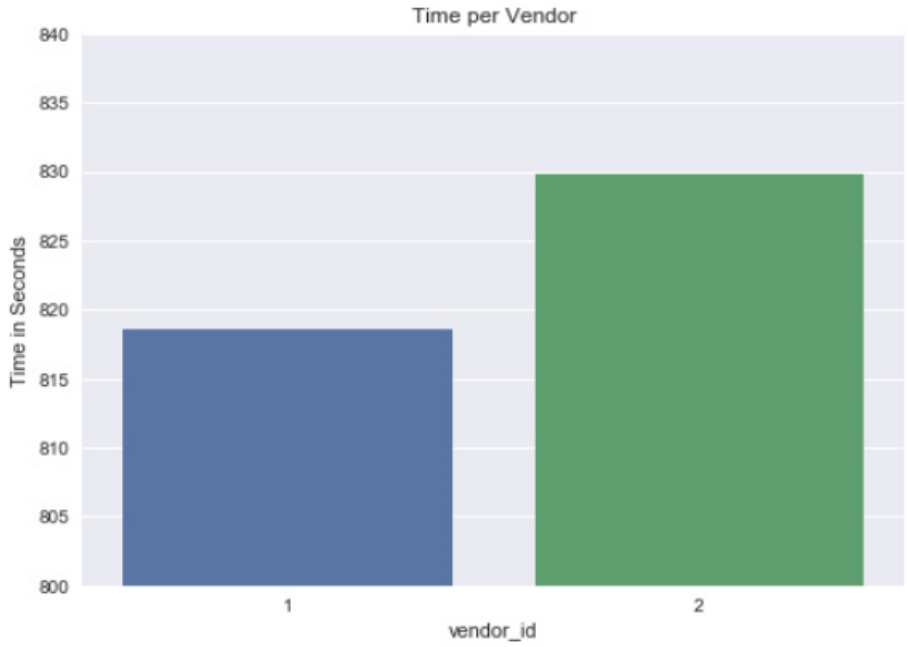

Figure 17 Average trip duration by driver

The dataset includes the indicator for the number of passengers (passenger_count). The impact of different number of passengers on the taxi trip duration should be taken into consideration. Figure 18 visualizes the relationship between the number of passengers and trip duration. The trip durations are approximately the same across different passenger counts. Thus, it can be concluded that while there is a relationship between the number of

装 订 线

passengers and trip duration, its effect is not significant.

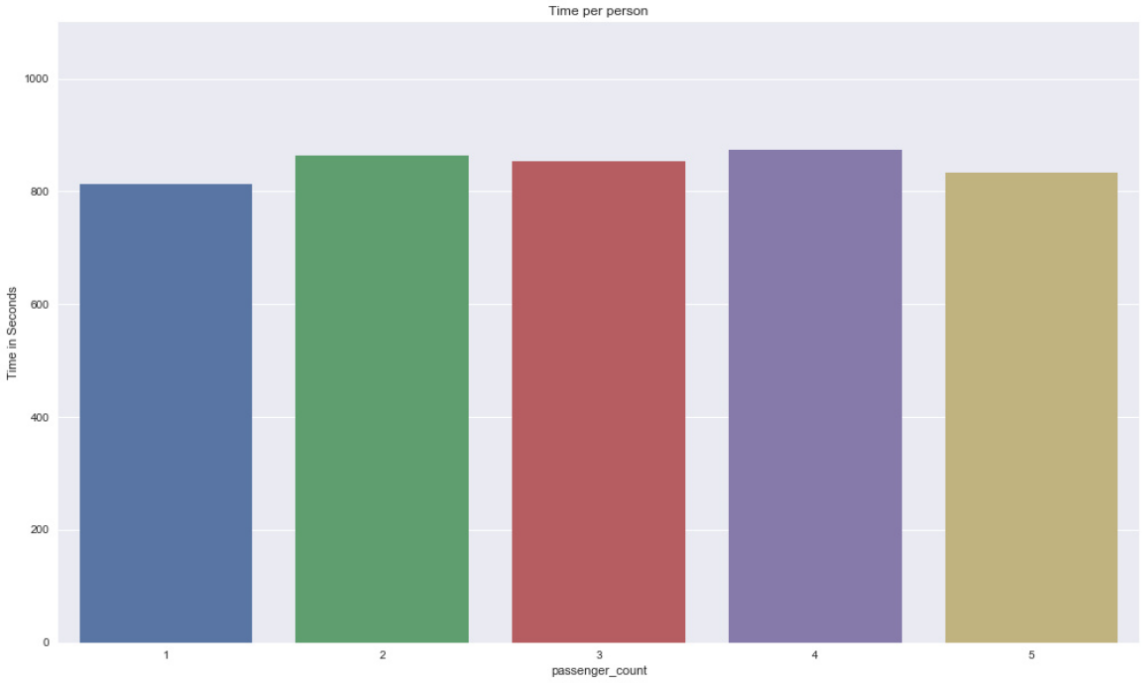

Figure 18 Relationship between number of passengers and trip duration

Finally, the number of peak hours per day should be considered. This is because there is a difference in traffic volume between the morning and evening peak hours and between day life and night life. Therefore, the impact of peak hour traffic volume on urban traffic has to be considered when predicting the trip duration.

Because the hourly data in the obtained dataset are individual hours and their distribution frequencies are difficult to be specifically counted and compared, K-Means clustering in visualizing peak hour trip duration was adopted in this thesis. As an unsupervised learning method, the main idea of K-Means is to distribute samples into different clusters according to their different characteristics, so that individual data points can be clustered into data sets with some identical characteristics. The algorithm divides the given sample set into K clusters according to the difference in the size of the distance of a certain feature of different data between the samples. The Means refers to the mean of all samples within the same cluster, i.e., with some of the same characteristics, and calculates their center of gravity. The steps are as follows.

1) Find the center of gravity of each cluster by using the method of finding the mean.
2) Based on the obtained center of gravity, calculate the distance from the sample to the center of gravity and determine which cluster the sample belongs to.

Therefore, using K-Means clustering can easily observe the distribution of trip durations

and compare it with the standard hours, so that the visualization of peak hour trip duration can be realized.

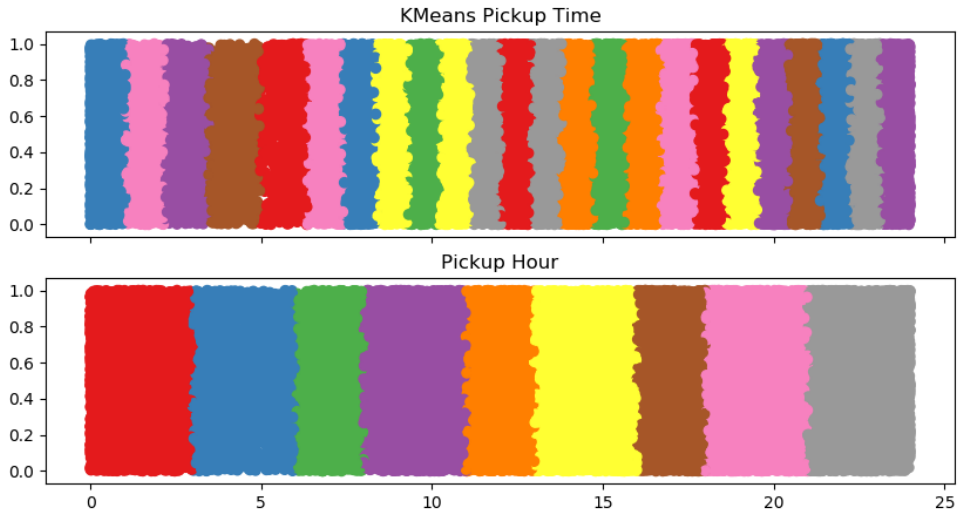

Figure 19 K-Means clustering of peak traffic volume distribution.

By dividing the dataset into 24 clusters (corresponding to 24 hours in a day), the distribution of peak traffic volume can be found. The peak hours are concentrated between 8 and 10 a.m. and around 3 p.m., representing the commuting times for New York City residents. Furthermore, peak hours after 7:00 p.m. and around 3:00 a.m. indicate the nightlife hours for the city's residents.

2.3.2. Spatial data pre-processing

The spatial data provided in the dataset includes latitude and longitude coordinates of pick-up and drop-off. Firstly, Figure 9 above provides an overview of the coordinates distribution. It can be observed that the pick-up and drop-off coordinates represented by the two plots in the training and test sets are similar, with the only significant difference being the larger amount of data in the training set. The utilization of the data in the dataset is further described in the following section.

2.3.2.1. Distance and direction

Given the known coordinate positions of the pick-up and drop-off in the dataset, the distance and direction of the trip can be calculated. Four methods are employed for this purpose: Manhattan distance, Euclidean distance, cosine distance and Haversine.

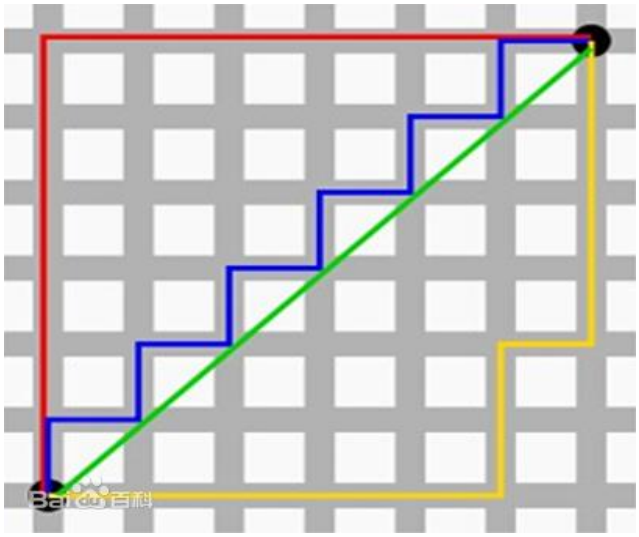

Figure 20 Distance algorithm

Figure 20 depicts the distance diagram, where the red curve represents the Manhattan distance, the blue and yellow lines represent the equivalent Manhattan distance, and the green line represents the Euclidean distance.

1) The Manhattan distance, also known as the taxi distance, is the sum of the length and width of a rectangle drawn with two points as diagonals. Assuming two coordinates (x_1, y_1) and (x_2, y_2) in a two-dimensional space, the Manhattan distance D is calculated as follows,

$$D = |x_1 - x_2| + |y_1 - y_2| \quad (1.)$$

2) Euclidean distance. the Euclidean distance represents the straight-line distance between two points on the diagonal of a rectangle. It is calculated using the Pythagorean theorem,

$$D = \sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2} \quad (2.)$$

3) Cosine distance, also referred to as cosine similarity, measures the distance and direction of two points based on the cosine of the angle between two vectors in three-dimensional space. Unlike the Euclidean distance, the cosine distance incorporates a measure of the difference between the two vectors in terms of direction.

4) Haversine function. In cases where a larger area needs to be considered or the direction of the taxi trip is important, the haversine function is utilized. Since the Earth is a sphere, the haversine function provides an accurate calculation of the distance between two points on the Earth when given their latitude and longitude coordinates. In this thesis, the haversine function is applied to both the test set and training set by converting the given latitude and longitude into radians. This function calculates the

half-positive vector distance between two points on the sphere. The resulting values are stored in the dataset as separate variables.

In summary, the haversine function was used to calculate the distance between the pick-up and drop-off location, and the cosine distance was used to calculate and represent the direction of this displacement. The calculated results are stored in the dataset as separate variables.

2.3.2.2. Location Clustering

To further process the data, location clustering is performed on the pick-up and drop-off locations. As the dataset contains discrete points, clustering is necessary to identify regions of interest for subsequent data utilization. Similar to the method used for handling the temporal distribution of peak traffic volume, the K-Means algorithm is employed to cluster the locations. In this case, a K value of 15 is chosen, resulting in the division of the locations into 15 clusters. The clustering results are illustrated in Figure 21, where the left figure represents the clustering of pick-up locations, and the right figure represents the clustering of drop-off locations. Each color module corresponds to a specific area where boarding and alighting locations are clustered. This clustering process effectively divides the Manhattan area into distinct regions. The clustered locations obtained can be utilized as variables in the subsequent taxi trip duration prediction algorithm.

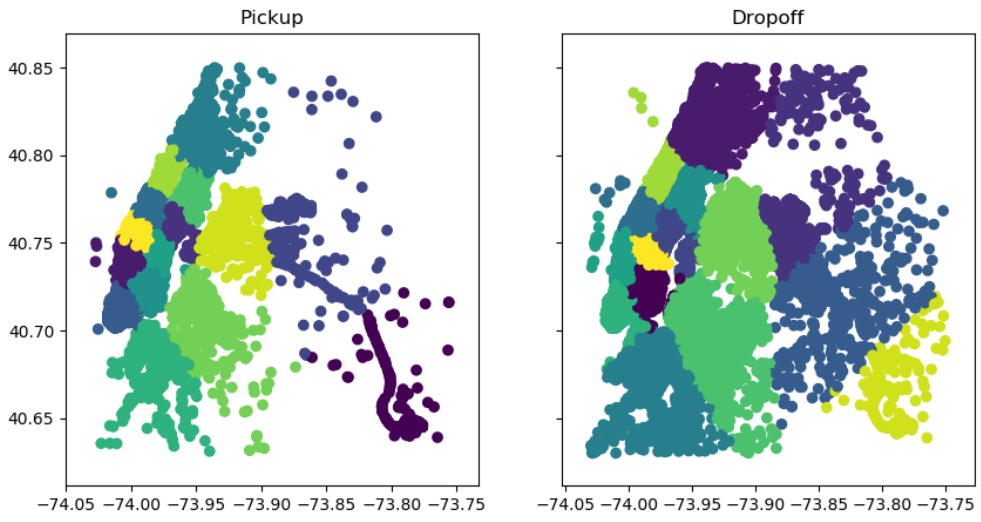

Figure 21 K-Means clustering of pick-up and drop-off locations.

After performing K-Means algorithm, pickup location clusters (pickup_cluster) and dropoff location clusters (dropoff_cluster) are created as features. These features are used as dummy variables in the later stages of the thesis, specifically in the one-hot encoding

input model for taxi trip duration prediction.

2.3.3. One-hot encoding

One-hot encoding is a technique used to represent categorical variables as binary vectors. It utilizes 0s and 1s to indicate different states. In this method, N-bit status registers are employed to encode N states, with each state having a separate register bit. When reading or writing, only one register bit corresponding to the state is set as valid. This approach provides a solution for dealing with discrete, unordered, and irregular data, which can be challenging for computers to recognize. In machine learning, one-hot encoding is frequently used to convert categorical features into a numerical format suitable for analysis. In this study, the processed data were encoded to generate machine learning-ready data.

To prepare the cleaned and processed data for direct usage in machine learning models, it is crucial to perform one-hot encoding on discrete variables, transforming them into dummy variables for computational purposes. The primary objective of this section is to extract various pieces of information such as driver, date, number of passengers, and pick-up/drop-off coordinates clustering for one-hot encoding. The pandas.get_dummies function can be employed to implement one-hot encoding in the thesis. To ensure that the extracted information can be utilized as features, it is necessary to confirm whether the date data in both datasets have the same size, including months, days, hours, and so on.

2.3.3.1 Time series data analysis

When predicting the duration of taxi trips, the impact of time series data needs to be considered. Time series data analysis involves understanding the patterns and characteristics of the given data, identifying trends within the data, and predicting whether these trends will continue in the future. The fundamental features of time series data analysis include:

- The assumption that trends observed in the past will persist in the future.
- The presence of irregularity in the data on which the prediction is based.
- The absence of consideration for causal relationships among variables.

Hence, data that changes over time can be identified as time series data. Prior to inputting the data into the model, the periodicity of time should be extracted as a feature. This can be achieved by converting the different state quantities of each sample into binary variables, representing zero or one. For instance, the seven days of the week (Monday to Sunday) can be transformed into seven variables.

The temporal data analyzed in this study spans a significant period, encompassing the entire first half of 2016 without distinguishing between different seasons, weekdays, or

holidays. Therefore, it is crucial to account for seasonality, periodicity, and holidays when handling time series data. Consequently, this section initially examines the data to filter out processed data that can be one-hot encoded as dummy variables. These variables must satisfy the condition of having the same number of occurrences in both the training and test sets in order to be encoded as dummy variables for doxastic coding. Upon retrieval, it is observed that both the training and test sets encompass six months, enabling months to be coded as dummy variables. Additionally, both sets include 31 days, allowing days to also be encoded as dummy variables. Similarly, both sets encompass 24 hours, permitting hours to be encoded as dummy variables, and both sets cover seven days per week, allowing weeks to be coded as dummy variables.

In summary, different parts of the date, including month, days of month days of week, and hour, can be safely utilized as features in the model. By incorporating time series data, the machine learning prediction model can consider temporal influences, resulting in more accurate predictions.

装

2.3.3.2 Creating dummy variables.

After the above analysis, the relevant variables that affect the trip duration of taxis can be obtained. These variables are uniquely and thermally coded and input into the model.

The data classified in the training set and test set above are converted into dummy variables, including vendor_id, passenger_count, store_and_fwd_flag, pick-up cluster, drop-off cluster, month, hour, day of month, day of week data.

订

Once the above data is converted into dummy variables, all categorical variables in the training and test sets are eliminated. This is because the information contained in these variables has already been replaced by the corresponding dummy variables. The categorical variables that are removed include id, vendor_id, passenger_count, store_and_fwd_flag, month, dayofmonth, hour, and dayofweek. Furthermore, the log_trip_duration feature, which was created in the previous section as a logarithmic transformation of one of the features for statistical observation and pattern identification, is also eliminated. Finally, all the processed dummy variables from the previous section are added.

线

After pre-processing of the data, the training set consists of 1,437,128 data points with 288 feature variables, while the test set comprises 525,134 data points with 287 feature variables. The training set has one additional feature compared to the test set, which is the target feature, i.e., the required trip duration. The processed dataset, along with the relevant features, is fed into the designed taxi trip duration prediction algorithm to obtain the prediction results.

Chapter 3 Design of Taxi Duration Prediction Algorithm

3.1. Prediction algorithm selection

In this chapter, an appropriate algorithm is selected and trained on the data to predict taxi trip duration while avoiding overfitting. Overfitting not only hampers the accuracy of duration prediction but also leads to misclassification. In this thesis, the XGBoost algorithm has been chosen. Here is a brief introduction to it.

XGBoost is a boosting algorithm that combines the idea of boosting with a parallel regression tree model. It has gained widespread usage in the field of data mining in recent years. The algorithm begins by training a base learner on the initial training set. The training samples are then adjusted based on the base learner's predictions. The updated base learner is used to train the previously misclassified samples. This iterative learning and adjustment process continues, allowing the algorithm to focus more on the previously misclassified instances. The final outcome is a set of weighted base learners that are combined to make predictions.

XGBoost is a type of boosting model that consists of K binary tree models, forming an additive algorithm. When predicting the score of a sample, the algorithm traverses each tree based on the sample's characteristics. Each leaf node in the tree corresponds to a score, and the scores from all trees are aggregated to obtain the final predicted value for the sample. By combining multiple binary tree models, XGBoost effectively mitigates the risk of overfitting. Additionally, having multiple trees leads to more accurate and scientifically sound classification and prediction results compared to a single base learner.

One notable feature of XGBoost is its ability to automatically assess feature importance. It continuously identifies important features while ignoring or downplaying non-important ones, which could potentially affect the results negatively. This feature allows the algorithm to improve the accuracy of its predictions.

In summary, the advantages of XGBoost can be summarized as follows: it incorporates regularization, parallel processing, high flexibility, pruning, built-in cross-validation, and so on. It effectively reduces overfitting, improves computation speed, enables customization of optimization objectives and evaluation criteria, and determines the optimal number of iterations, among other benefits. Compared to the Gradient Boosting Decision Tree (GBDT) model, XGBoost overcomes limitations in computational speed and

accuracy. It introduces regularization to the gradient boosting decision tree model, thereby mitigating overfitting. While the traditional gradient boosting decision tree model performs a first-order Taylor expansion on the loss function, taking negative gradient values as residuals, the XGBoost algorithm performs a second-order Taylor expansion, resulting in more accurate model outcomes. Additionally, XGBoost can continue training on the results from previous rounds, and its parallel computation is achieved by analyzing the importance of each feature based on the budgeted results of the previous round. This allows for parallel decision-making, reduces resource usage, speeds up computations, and shortens the overall computing time. Considering these advantages of the XGBoost algorithm, it has been chosen to perform the taxi trip duration prediction in this thesis.

3.2. Prediction algorithm design

3.2.1. Evaluation metrics

After constructing the model and obtaining predictions, it is crucial to evaluate the model's performance. The choice of evaluation method depends on the data type and model used and aims to measure the discrepancy between predicted and actual values. Common evaluation metrics include MSE, RMSE and RMSLE.

1) Mean square error (MSE)

$$\mathrm{MSE} = \frac{1}{n}\sum_{i=1}^{n}(yi - \bar{y}i)^2 \tag{3.}$$

2) Root mean square error (RMSE):

$$\mathrm{RMSE} = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(yi - \bar{y}i)^2} \tag{4.}$$

3) Root mean square log error (RMSLE):

$$\mathrm{RMSLE} = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(\log(\bar{y}i + 1) - \log(yi + 1))^2} \tag{5.}$$

Here, n represents the total number of observation sets, y_i represents the predicted value, and $\bar{y}$ represents the true value. RMSE is chosen as the evaluation metric for taxi trip duration prediction in this thesis which assigns higher penalties to poor predictions.

3.2.2. Parameter selection and optimization

The selection of parameters directly impacts the accuracy of the machine learning model. Common methods for parameter tuning include the expert empirical method and the grid

search method. The expert empirical method heavily relies on subjective judgment, while the grid search method suffers from narrow parameter selection ranges, making it difficult to obtain optimal parameters. The XGBoost algorithm includes three major types of parameters: general parameters, booster parameters, and target learning parameters. After several experimental simulations and optimizations, the following parameter combinations are selected.

General parameters: booster, silent, and nthread.

- The model booster is chosen based on gbtree, which represents the tree model for each iteration.
- Silent was set to 1, enabling silent mode to suppress information output.
- nthread is not specified, allowing the utilization of all CPU cores for multi-threaded control.

Booster parameters: eta, min_child_weight, max_depth, subsample, colsample_bytree, and lambda.

- eta was set to 0.5 to reduce the weight of each step, thereby improving the model's robustness.
- min_child_weight determines the minimum weight of a leaf node sample and helps prevent overfitting. In this thesis, it was set to 1.
- max_depth represents the maximum depth of the constructed tree. It is used to control overfitting. Setting a large value allows the model to learn specific samples, resulting in overfitting. In this thesis, it was set to 6.
- subsample controls the number of randomly sampled samples in each tree. A value of 0.9 was chosen to mitigate overfitting when the subsample size is small.
- colsample_bytree controls the proportion of randomly sampled columns (features) in each tree. It was set to 0.9 in this thesis.
- lambda was set to the default value of 1. This parameter controls the regularization part of the model, reducing overfitting.

Objective learning parameters: objective and eval_metric.

- The objective parameter defines the loss function to be minimized. The default value reg:linear was used in this thesis.
- The eval_metric parameter selects the evaluation method for the data. In this project, RMSE was chosen as the evaluation metric.

3.3. Training of the prediction algorithm

After completing the data processing and designing the taxi trip duration prediction

algorithm, the processed final dataset is input into the prediction algorithm for training, and the results are obtained through computation. The hardware platform used in this thesis consists of an Intel i5-7300HQ CPU and an NVIDIA GeForce GTX1050Ti GPU.

The XGBoost algorithm requires three sets of data: a training set, a test set, and a validation set. The validation set is used for continuous model accuracy evaluation and is eventually used to make predictions on the test set. Therefore, the resulting dataset is partitioned accordingly. Initially, the training set is divided into a sub-training set and a sub-validation set. For this thesis, 1,000,000 data points are selected from the training set, with 80% used as the training set and 20% as the validation set.

Before inputting the data into the model, the data were de-indexed and restructured to ensure accurate referencing of each data row. Indexes were removed to recreate for XGBoost. The newly created training and test sets were fed into the model, which then predicted the test set and outputs the prediction results. As shown in the table below, 49 iterations were performed and the following results were obtained.

Table 1 RMSE of the model prediction results.

Iterations	RMSE of the training set	RMSE of the validation set
1	1.56126	1.56132
2	0.86811	0.86812
3	0.57378	0.57384
4	0.46897	0.46926
5	0.43527	0.43579
6	0.42333	0.42406
7	0.41586	0.41673
8	0.40709	0.40799
9	0.40397	0.40503
10	0.40084	0.40226
11	0.39818	0.39988
12	0.39616	0.39799
13	0.39318	0.39526
14	0.39011	0.39238
15	0.38828	0.39068
16	0.38702	0.38958
17	0.38567	0.38849
18	0.38443	0.38737
19	0.38302	0.3862

20	0.38166	0.38521
21	0.38055	0.38424
22	0.37952	0.38343
23	0.37807	0.38242
24	0.37652	0.38106
25	0.37564	0.38047
26	0.37474	0.3797
27	0.37404	0.37924
28	0.3734	0.37874
29	0.37247	0.37796
30	0.37182	0.37761
31	0.37117	0.37708
32	0.3707	0.37686
33	0.36957	0.376
34	0.36872	0.37539
35	0.3681	0.37492
36	0.3673	0.37428
37	0.36684	0.37404
38	0.36608	0.37343
39	0.36552	0.37292
40	0.36502	0.3726
41	0.36427	0.37212
42	0.3639	0.37192
43	0.3633	0.37153
44	0.36264	0.37097
45	0.3622	0.37066
46	0.36167	0.37041
47	0.36118	0.37015
48	0.36055	0.36968
49	0.35986	0.36923
Modeling RMSE	0.36923	

After performing the prediction algorithm, the RMSE of the training set improved from 3.02531 to 0.35986, representing an 88.10% improvement. Similarly, the RMSE of the validation set improved from 3.02547 to 0.36923, reflecting an 87.80% improvement. Although the results of the training and validation sets showed continued improvement, the

enhancement effect diminished. The model achieved stable results, and further increasing the number of iterations or modifying relevant parameters might lead to overfitting. The RMSE of the algorithm model on validation set is 0.36923 after 49 iterations. The machine learning algorithm significantly improved the prediction results. The prediction results of the test set will be stored in the dataset pred.

In conclusion, the prediction results for New York City taxi hours in the first half of 2016 have been obtained, with a root mean square error of 0.36923 in the validation set.

Chapter 4 Conclusions and future work

4.1. Conclusions

The core focus of this thesis lies in the processing of big data and the application of machine learning techniques. The research methodology involved several crucial steps. Initially, the collected data underwent inspection, classification, and cleaning processes to extract valid information and augment it with relevant data, resulting in a refined dataset. Subsequently, mathematical preprocessing techniques were employed to transform the data into a usable and meaningful format. The visualization of the data aided in uncovering hidden patterns, facilitating the creation of new features for machine learning purposes. Finally, a machine learning model was selected, and the processed data, along with the newly created features, were inputted into the model. Parameter tuning and the design of evaluation metrics were carried out to derive the final prediction results.

In this thesis, a data processing as well as machine learning model has been constructed. For a large amount of complex data, matplotlib, seaborn and other packages have been used to operate and process the data, and visually observed the data features to extract relevant variables. XGBoost algorithm has been chosen to develop the predictions of trip durations of taxis. This algorithm has the advantages of multi-threading and can effectively reduce overfitting which was also reflected in the training process. In the case of limited experimental conditions, the experimental results can be obtained quickly and accurately in a timely manner, which reduces the experimental cost. In this thesis, by cleaning and visualizing the data, the feature variables that affect the trip duration of the taxi successfully were successfully found, which provided a great help to control the number of variables in the input model within a reasonable range in the following.

4.2. Future work

While this thesis has made notable progress, several areas for improvement and further research exist. In the design of the taxi trip duration prediction algorithm, only XGBoost algorithm was performed. The absence of comparative analyses among algorithms hampered the comprehensive understanding of the taxi trip duration prediction domain. In future research, introducing different algorithms for comparison purposes would enable a more accurate and profound comprehension of the topic, leading to scientifically rigorous and objective conclusions.

Moreover, when extracting variables related to taxi trip duration prediction, some variables

were not considered comprehensively enough or overlooked, potentially impacting the accuracy of the prediction results. For example, a lack of understanding regarding the statistical data, the actual operating conditions of taxis in New York City and incomplete analysis of the "store_and_fwd_flag" variable. Further research in this domain necessitates a thorough and accurate understanding of various categorical variables in big data to ensure scientifically grounded predictions.

The research on transportation often deals with massive volumes of data, encompassing taxis, buses, subways, passenger cars, freight vehicles, and even larger quantities of private cars. The generation of such vast amounts of data presents challenges for traditional research methods, often resulting in overlooked data and ignored variables, thus impeding the exploration of valuable insights hidden within the data, which hampers effective solutions to real-world traffic problems and leads to significant resource wastage. However, the development and maturity of artificial intelligence technologies can change this situation. Supercomputers have provided researchers with methods and tools to address big data challenges. Machine learning, with its significant value across diverse fields, holds immense potential in transportation research. With the aid and influence of new technologies, many previously intractable traffic problems now have novel approaches and breakthroughs. Continuously exploring and researching machine learning techniques is warranted to discover more accurate prediction and simulation methods.

Acknowledgement

Time flies, four years of college time passes quickly. Four years ago, I entered the campus with a dream, and university life has made me grow up a lot. There is not only a full campus life, bright sunshine, spacious classrooms, but also a strong learning atmosphere, harmonious teacher-student relationship, and I have been growing up under the stern but kind teachers.

Thanks to the learning environment provided by the school, I was able to successfully complete my four years of study, broaden my horizons and acquire knowledge. I would like to thank all the teachers who taught me when I encountered difficult problems and patiently guided me to overcome them; I would like to thank my counselors who cared for me in my life and showed me the way to my life.

I would like to thank Professor Gao Yang for his guidance during my graduation design process in these six months. From topic selection, simulation, framework, thesis writing, etc., I am grateful to Prof. Gao Yang for his weekly guidance without interruption. From the initial voice calls to the offline meetings after the school year started, I was given the timeliest attention and reliable layman's answers to every question I had. In half a year, I went from being at a loss with the topic to successfully writing a complete machine learning algorithm without Prof. Gao's help and encouragement.

Thank you to my teachers, my classmates, and Chang'an University, where I spent four years of my youth and sublimated my dreams.

References

[1]徐健锋,汤涛,严军峰,刘真.基于多机器学习竞争策略的短时交通流预测[J].交通运输系统工程与信息,2016,16(04):185-190+198.
[2]丁洁,刘晋峰,杨祖莧,阎高伟.基于深度学习的交通拥堵检测[J/OL].重庆大学学报:1-9[2020-04-14].
[4]徐周波,杨健,刘华东,黄文文.基于 XGboost 与拓扑结构信息的蛋白质复合物识别算法[J/OL].计算机应用:1-5[2020-04-21].
[3]陈屹.神经网络与深度学习实战[M].北京:机械工业出版社,2019.
[4]王斌会,王术.Python 数据挖掘方法及应用[M].北京:电子工业出版社,2019.
[5]Alexander T. Combs. Python Machine Learning Blueprints[M].北京:人民邮电出版社,2017.
[6]杨连贺,董禹龙,房超.Python 程序设计使用教程[M].北京:清华大学出版社,2018.
[7]赵亚萍,张和生,周卓楠,等. 基于最小二乘支持向量机的交通流量预测模型[J].北京交通大学学报,2011 (02):114—117+136.
[8]郭敏,蓝金辉,肖翔，等. 基于混沌理论对北京二环路进行短时交通流量预测的研究[J]. 交通运输系统工程与信息,2010,l0(2):106—111.
[9]张通,张骏,杨霄.基于混合 AGO—SVM 的高速公路短时交通量预测研究[J] 交通运输系统工程与信息，2011,11(1):157-162.
[10]王建,邓卫,赵金宝.基于贝叶斯网络多方法组合的短时交通流量预测[J].交通运输系统工程与信息,2011,11(4):147—153.
[11]崔艳,程跃华.小波支持向量机在交通流量预测中的应用[J].计算机仿真,2011(7):353—356.
[12]高述涛.cs 算法优化 BP 神经网络的短时交通流量预[J].计算机工程与应用，2013(9)：106—109.
[13]王明月,王晶,齐瑞云,等.基于改进 GMDH 算法的道路短时交通流量预测[J].计算机应用,2015(s1):101-103+134.
[14]李巧茹,赵蓉,陈亮.基于 SVM 与自适应时空数据融合的短时交通流量预测模型[J].北京工业大学学报,2015(4):597—602.
[15]周桐,杨智勇,孙棣华,等.分车型的高速公路短时交通流量预测方法研究[J].计算机应用研究，2015(7):1996-1999.
[16]李嘉. 基于卷积神经网络的心律失常自动分类关键技术研究[D].吉林大学,2019.
[17]蹇松雷,卢凯.复杂异构数据的表征学习综述[J].计算机科学,2020,47(02):1-9.
[18] 邓顺熙, 陈爱侠, 曹申存. 高速公路汽车污染物排放因子的测试与研究[J].中国公路学报,1999(S1): 96-102.
[19]陈雪明.纽约的公共交通系统和规划经验谈[J].国际城市规划,2015,30(S1):84-88.
[20]杨敏明,王雪松.纽约市道路交通安全改善经验与启示[J].交通与运输 2019,35(02):37-40.
[21]叶景,李丽娟,唐臻旭.基于 CNN-XGBoost 的短时交通流预测[J].计算机工程与设计,2020,41(04):1080-1086.
[22]Kasun Bandara,Christoph Bergmeir,Slawek Smyl. Forecasting across time series databases using recurrent neural networks on groups of similar series: A clustering approach[J]. Expert Systems With Applications,2020,140.

装 订 线