SUSTech Southern University of Science and Technology

# 01

# Introduction

# Introduction

- The theme of this data set is "Smart Traffic prediction in the era of Mobile Internet", and the goal is to help social smart travel and urban traffic intelligent control, accurately predict the travel time of each key road section in a certain period, and realize the prediction of traffic state fluctuations.

# Introduction

- **Dataset**

| 属性 | 类型 | 说明 |
|---|---|---|
| link_ID | string | 每条路段（link）的唯一标识 |
| length | double | link 长度（米） |
| width | double | link 宽度（米） |
| link_class | int | link 道路等级，例如 1 代表主干道。 |

表 1

| 属性 | 类型 | 说明 |
|---|---|---|
| link_ID | string | 每条路段（link）的唯一标识 |
| in_links | string | link 的直接上游 link，linkID 之间以#分割 |
| out_links | string | link 的直接下游 link，linkID 之间以#分割 |

表 2

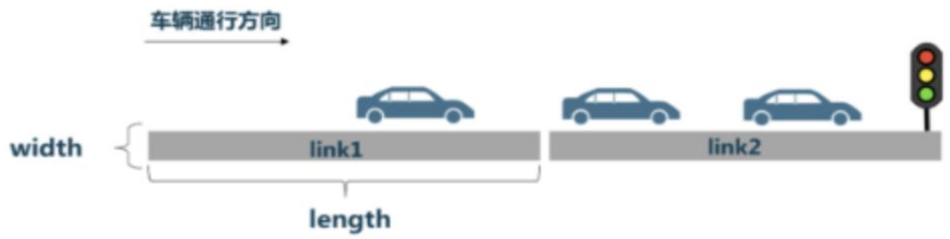




图 2：link2 的直接上游（左）；link2 的直接下游（右）

# Introduction

- **Dataset**
  - date_time: year, month, day
  - more about date_time: weekend, weekday
  - time_interval: divided day time into serveral part
  - travel_time: the average travel time they spend

| 属性 | 类型 | 说明 |
| --- | --- | --- |
| link_ID | string | 每条路段（link）的唯一标识 |
| date_time | date | 日期，例如'2015-10-01' |
| time_interval | string | 时间段，例如[2015-09-01 00:00:00', 2015-09-01 00:00:10) |
| travel_time | double | 车辆在路段上的平均旅行时间（秒） |

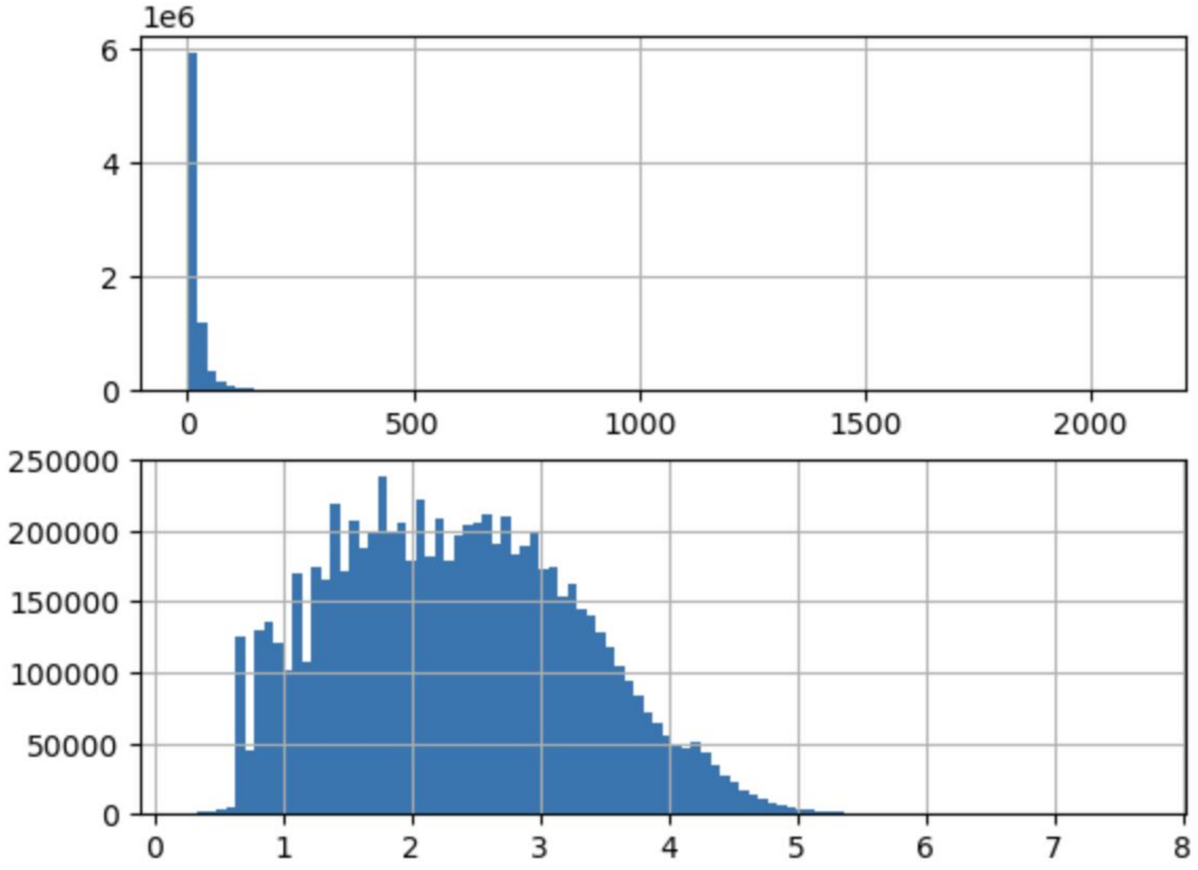
表 3

SUSTech Southern University of Science and Technology

# 02

# Data preprocess

# Visualization

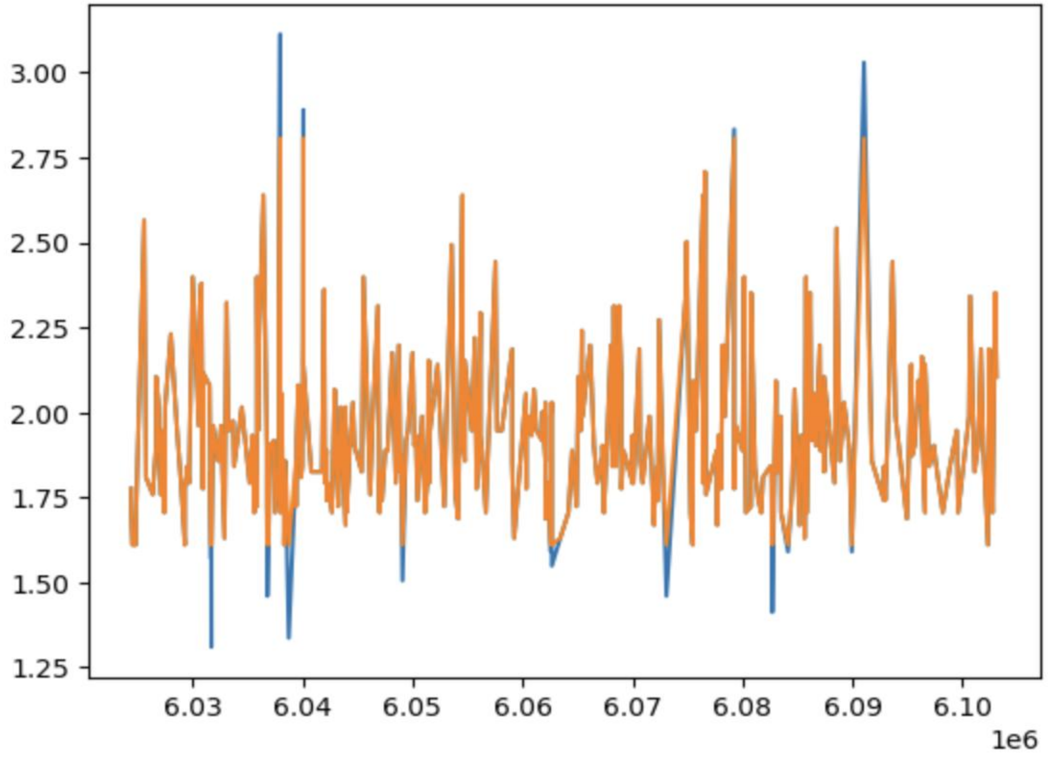



Average travel time distribution

Normal Distribution

# Data preprocess

- **Remove outliers**




Filter the travel time of day for each link ID

```
group[group < group.quantile(0.05)] = group.quantile(0.05)
group[group > group.quantile(0.99)] = group.quantile(0.99)
```

# Data preprocess

- **Complete the missing value**
  - Complete the date range and merge the Link ID
  - Seasonal date trend + Daily hour trend
  - Linear predict
  - Xgboost predict

SUSTech Southern University of Science and Technology

03

# Feature Extraction

# Road Feature

Southern University of Science and Technology

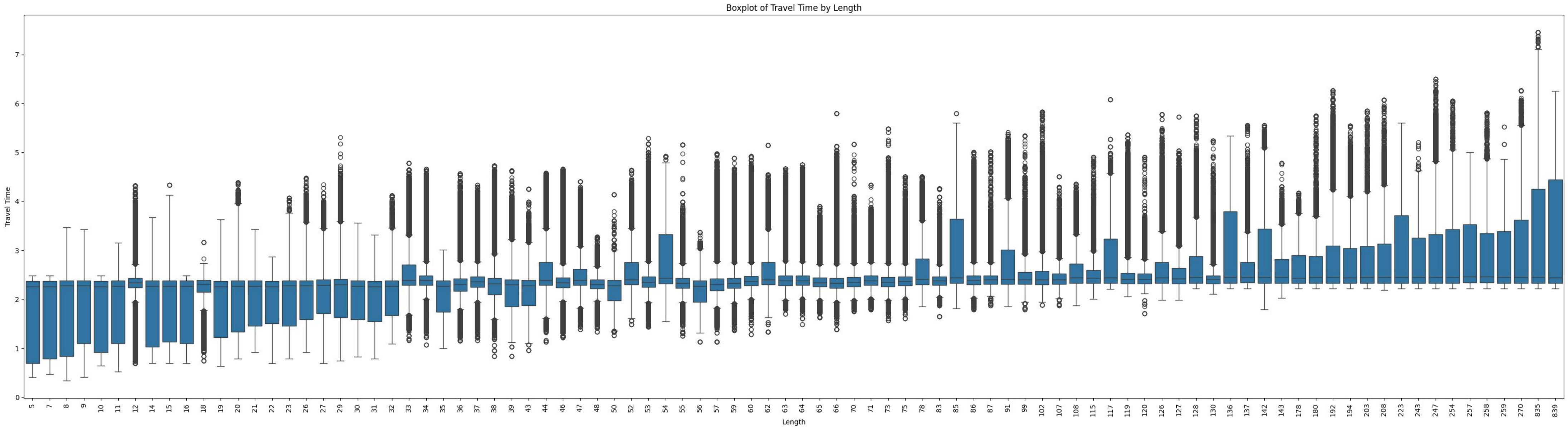
- **Dataset**
  - length
  - width
  - link_class
  - in_links
  - out_links

| 属性 | 类型 | 说明 |
| --- | --- | --- |
| link_ID | string | 每条路段（link）的唯一标识 |
| length | double | link 长度（米） |
| width | double | link 宽度（米） |
| link_class | int | link 道路等级，例如 1 代表主干道。 |

| 属性 | 类型 | 说明 |
| --- | --- | --- |
| link_ID | string | 每条路段（link）的唯一标识 |
| in_links | string | link 的直接上游 link，linkID 之间以#分割 |
| out_links | string | link 的直接下游 link，linkID 之间以#分割 |

# Road Feature

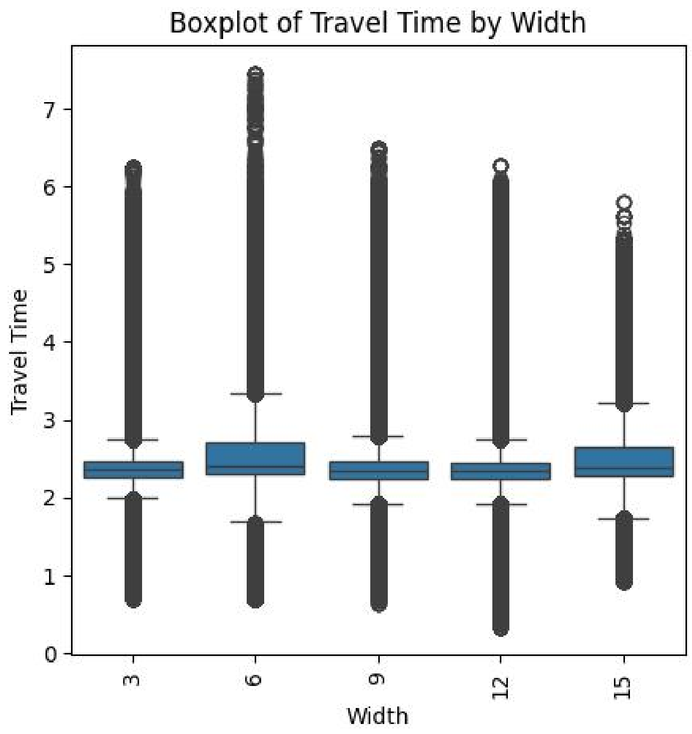
- **Travel Time by Length**




Showing the distribution of travel time using box plots and revealing the trend as road length increases.

# Road Feature

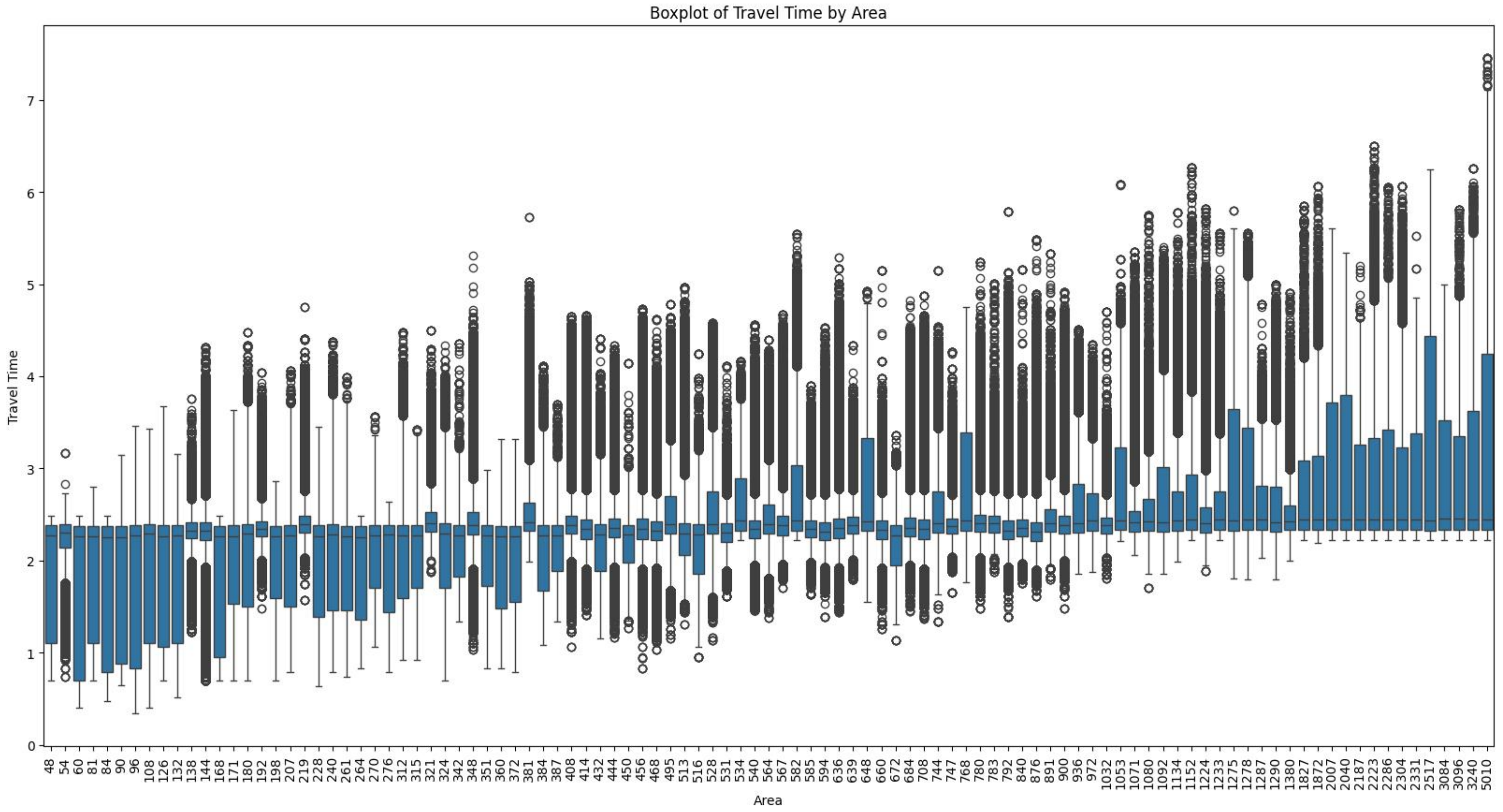
- **Travel Time by Width**




Travel time varies with different road widths.

# Road Feature

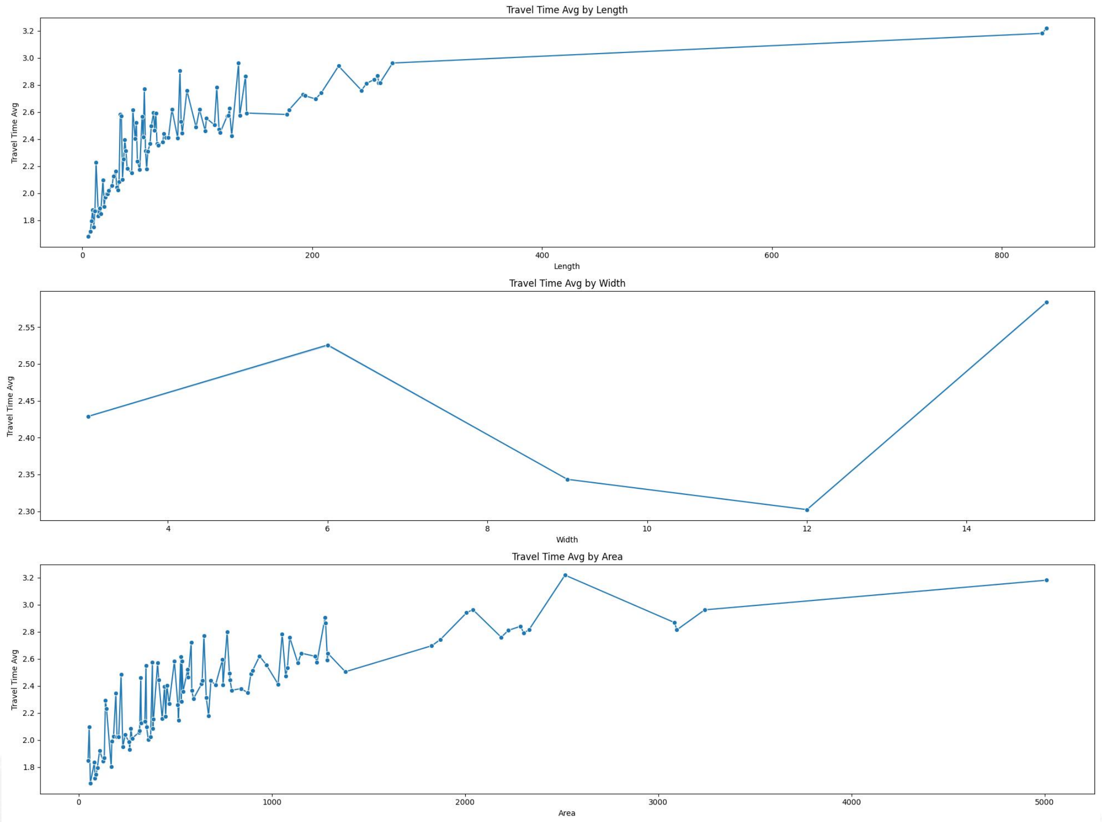
- ## Travel Time by Area




Showing the distribution of travel time using box plots and revealing the trend as road area increases.

# Road Feature

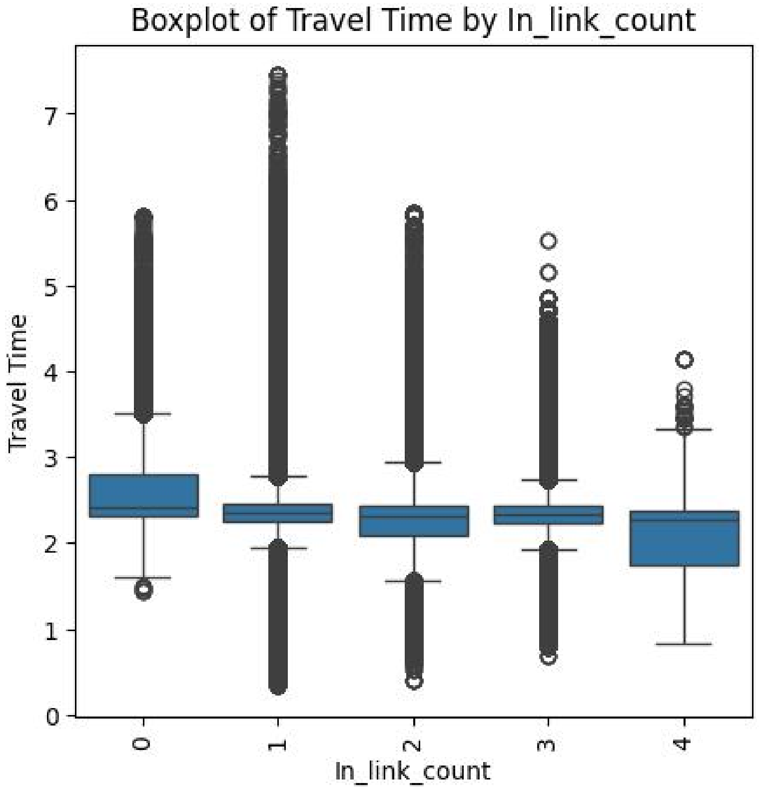
- **Travel Time Avg by Length, Width, Area**

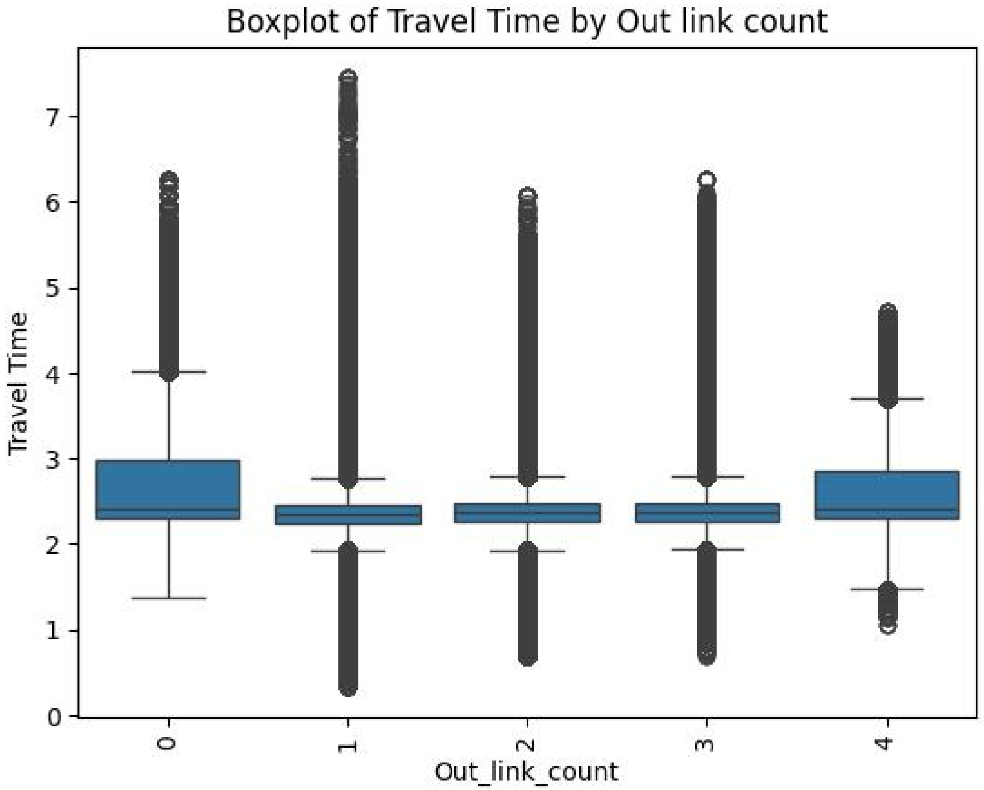



- The average travel time generally increases with road length.
- The average travel time first increases with road width up to 6 meters, then decreases, and finally rises again.
- The average travel time tends to increase with road area, with some fluctuations along the way.

# Road Feature

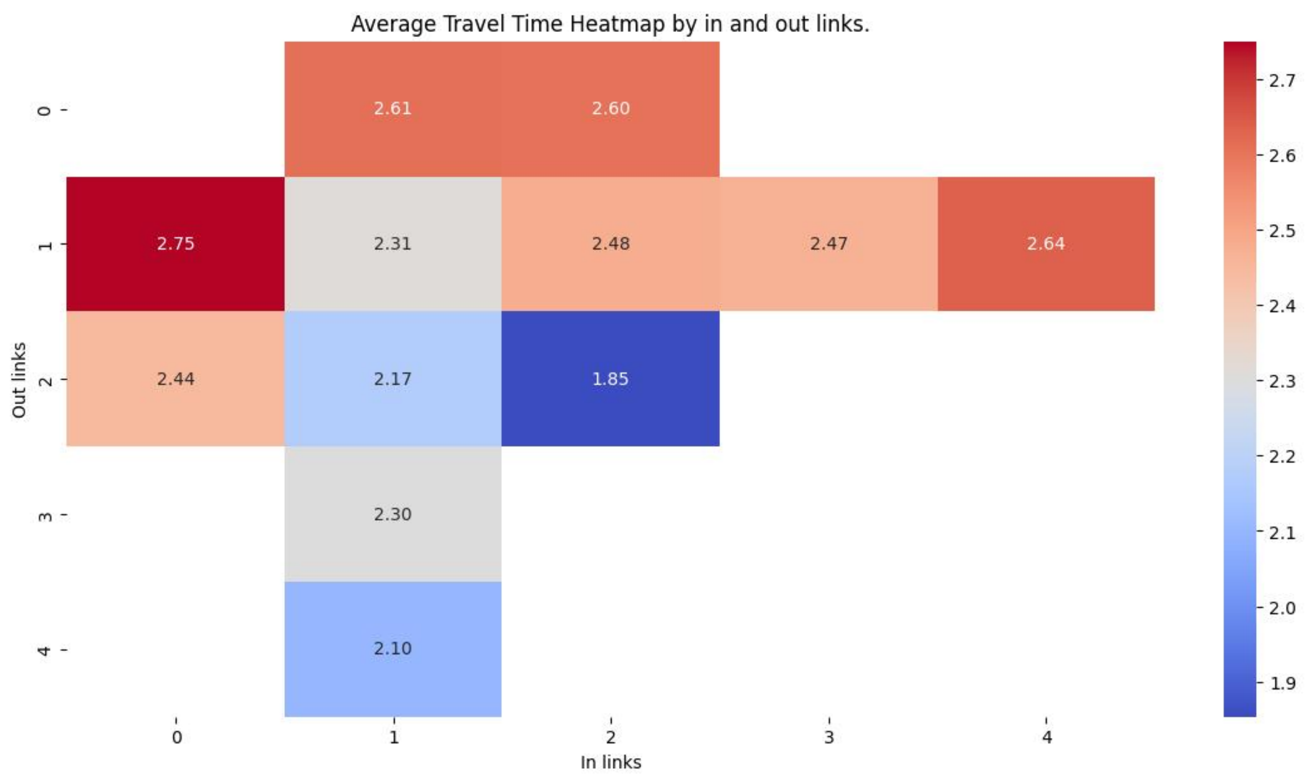
- **Travel Time by in and out links**



# Road Feature

- **Travel Time by in and out links**




- The highest average travel time (2.75) occurs with one outgoing link and no incoming links.
- The lowest average travel time (1.85) is observed with two incoming links and two outgoing links.
- Generally, travel time tends to be higher with fewer incoming or outgoing links, while intermediate values of links tend to have lower average travel times.

# Road Feature

- **Final Result**
  - linkID: The ID of road.
  - travel_time_avg: The average travel time of each road.
  - area: The area of each road.
  - in_link_count: The count of in-links.
  - out_link_count:  The count of out-links.
  - link_count_combination: The combination of in and out links.

# Time Feature

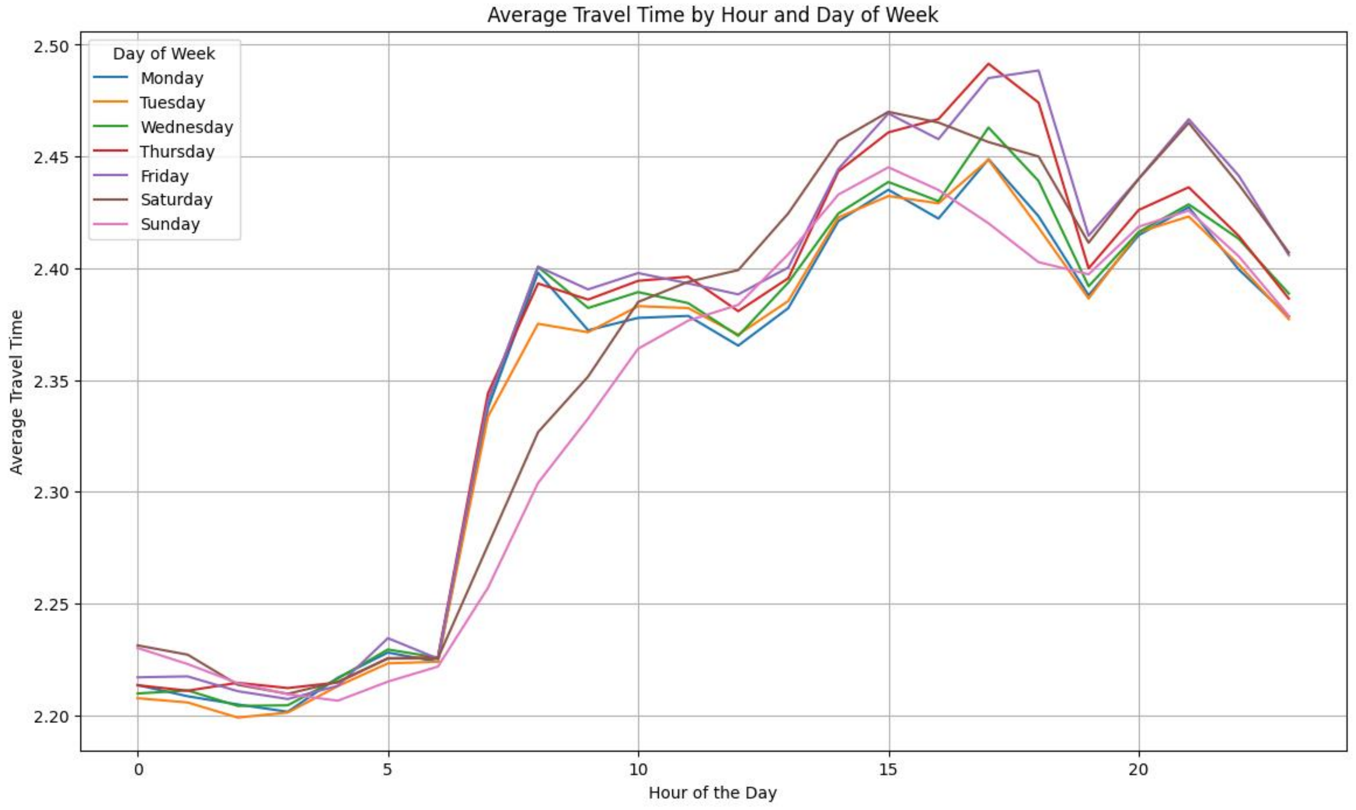
- **Dataset**
  - date_time: year, month, day
  - more about date_time: weekend, weekday
  - time_interval: divided day time into serveral part
  - travel_time: the average travel time they spend

| 属性 | 类型 | 说明 |
|---|---|---|
| link_ID | string | 每条路段（link）的唯一标识 |
| date_time | date | 日期，例如'2015-10-01' |
| time_interval | string | 时间段，例如[2015-09-01 00:00:00',<br>2015-09-01 00:00:10) |
| travel_time | double | 车辆在路段上的平均旅行时间（秒） |

表 3

# Time Feature

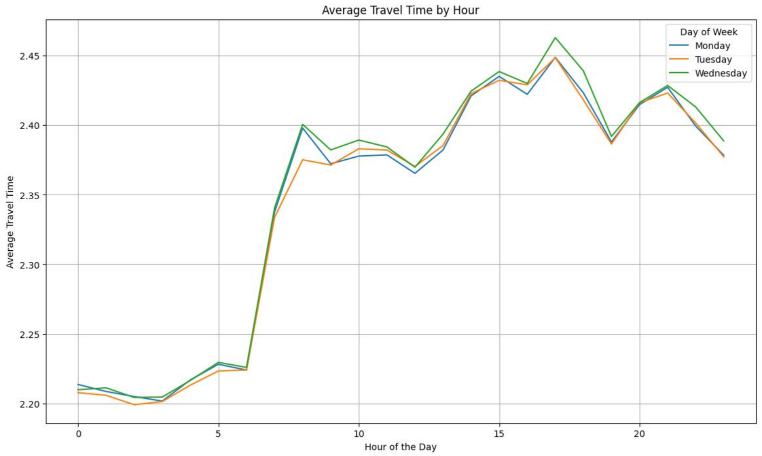
- **Average Travel Time by Hour and Day of Week**



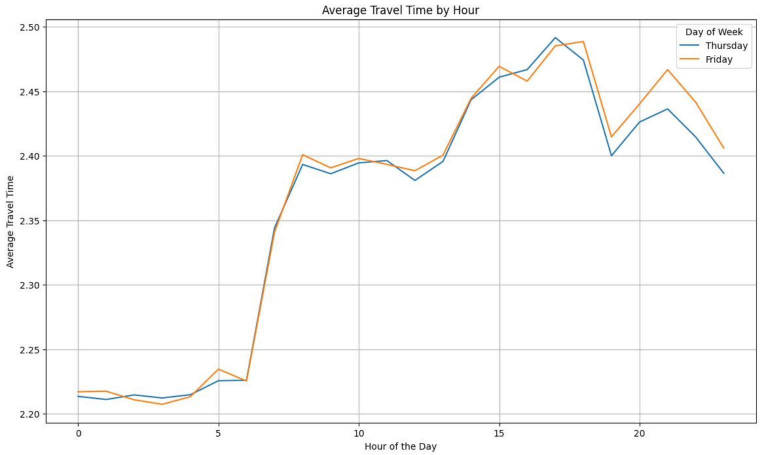

- Visualize the travel time trend of days in a week.
- Complicate -> Divide into groups

# Time Feature

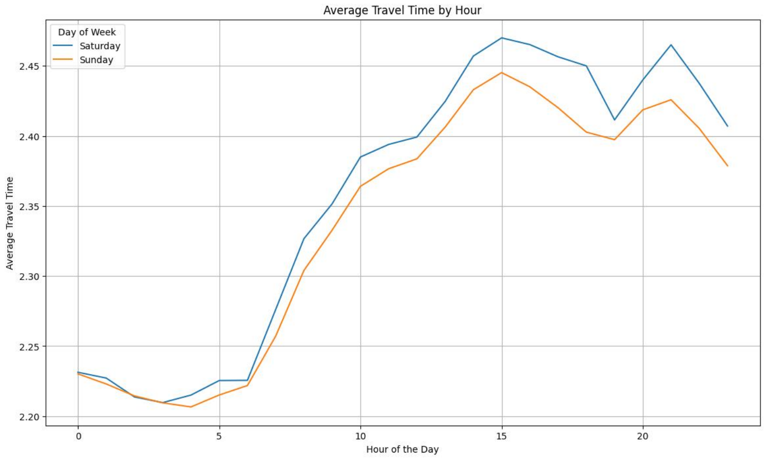
- **Divide into 3 parts according to daily trend**




Monday to Wednesday




Thursday to Friday

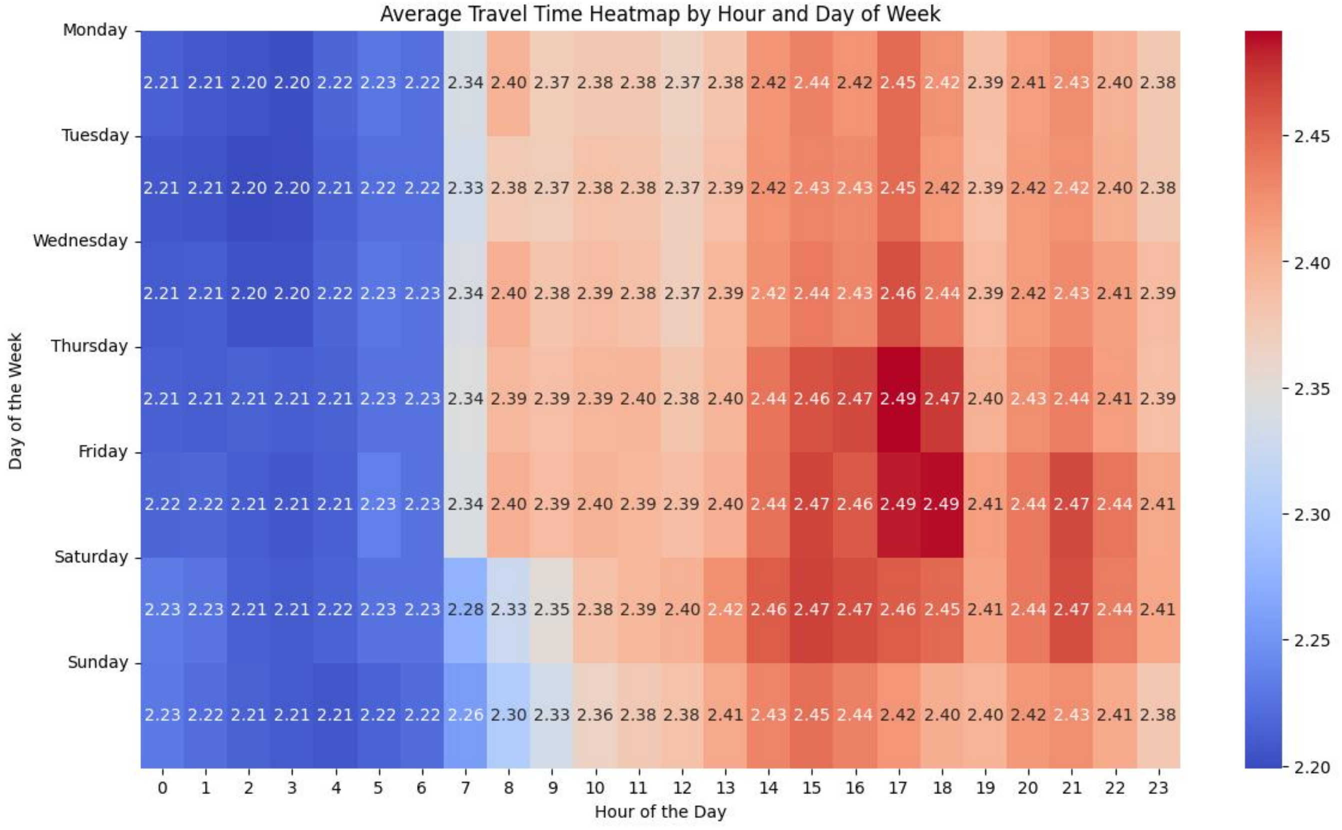



Saturday to Sunday

So we can divide it into 3 parts

# Time Feature

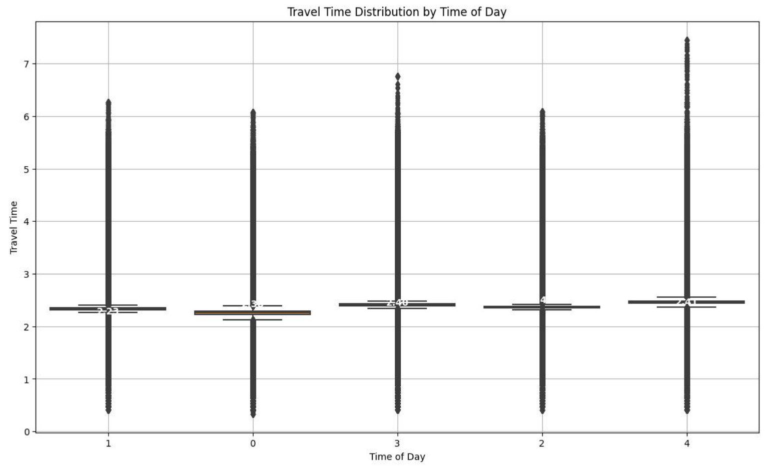
## • Average Travel Time Heatmap by Hour and Day of Week

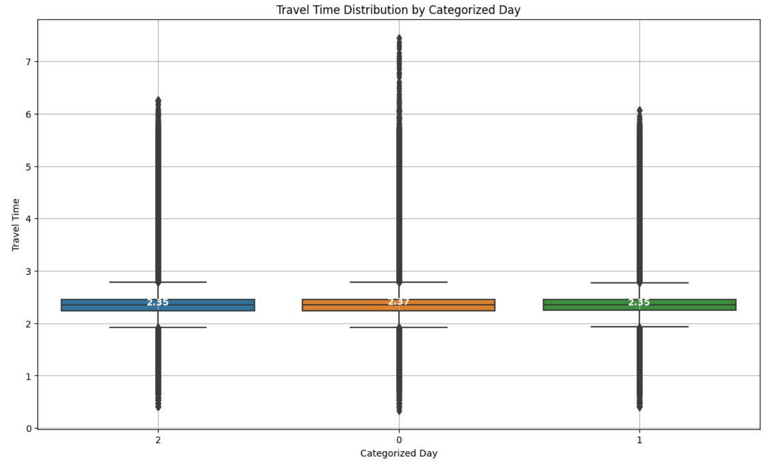


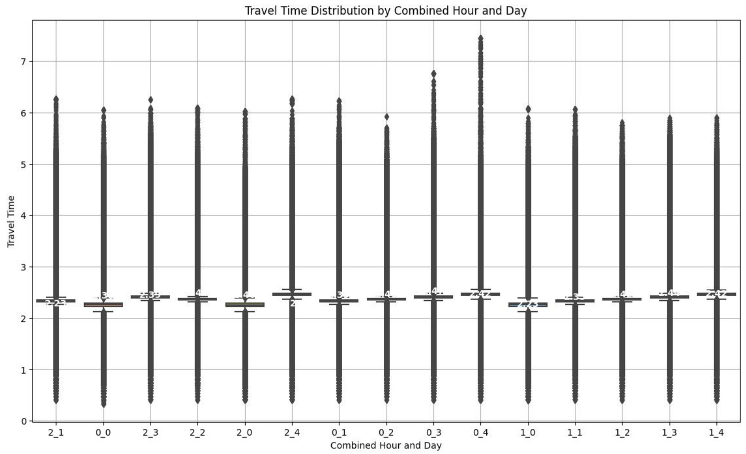

- Split aytime into 5 parts;
  - 0-7: low travel time
  - 8-11: normal
  - 12-15: increasing
  - 15-19: very busy time
  - 19-24: busy time

# Time Feature

- **Combine**



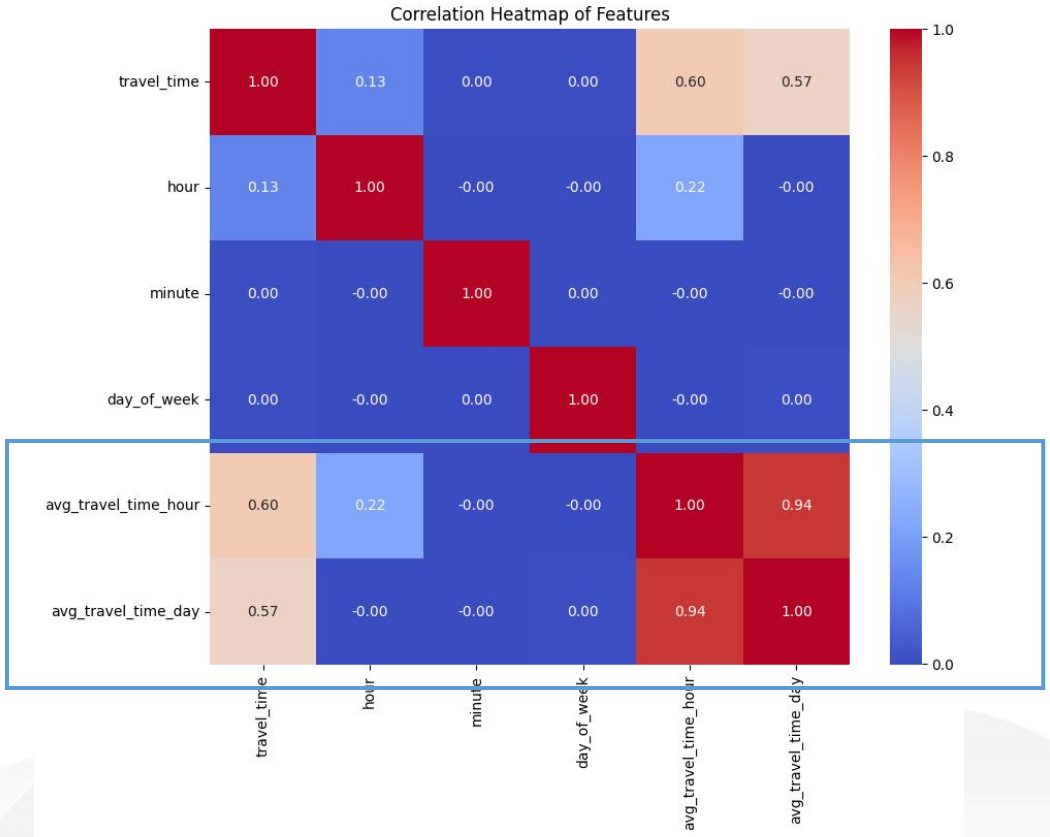




The greater the difference in means, the more pronounced the difference in travel times between categories and the easier it is for the model to learn useful information.

# Time Feature

## • Correlation Heatmap of Features




- Correlation Heatmap: measures the linear relationship between features
- Result: avg_travel_time_day and avg_travel_time_hour both show high relation to travel time, and they are similar to each other -> They represent same mode
- So we keep avg_travel_time_hour

# Time Feature

- **Final Result**
  - linkID: The ID of road
  - start_time: The Original Data
  - travel_time: The Result We Want
  - combined_hour_day: Day(group 7 days into three group)_Hour(group 24hour into 4 group)
  - avg_travel_time_hour:  Average travel time in hour

# 04

# Experiment

# Model

- **XGBoost (eXtreme Gradient Boosting)**
  - **Introduction:** an efficient machine learning algorithm that is based on decision tree
  - **Advantages:** efficient, accurate, flexible and highly interpretable

- **Hyperparameter tuning**
  - **Method:** grid search with cross-validation
  - **Scoring:** negative mean squared error (-MSE)
  - **Optimum:** {'gamma': 0, 'learning rate': 0.2, 'max depth': 8, 'n_estimators': 150}

# Ablation Experiment

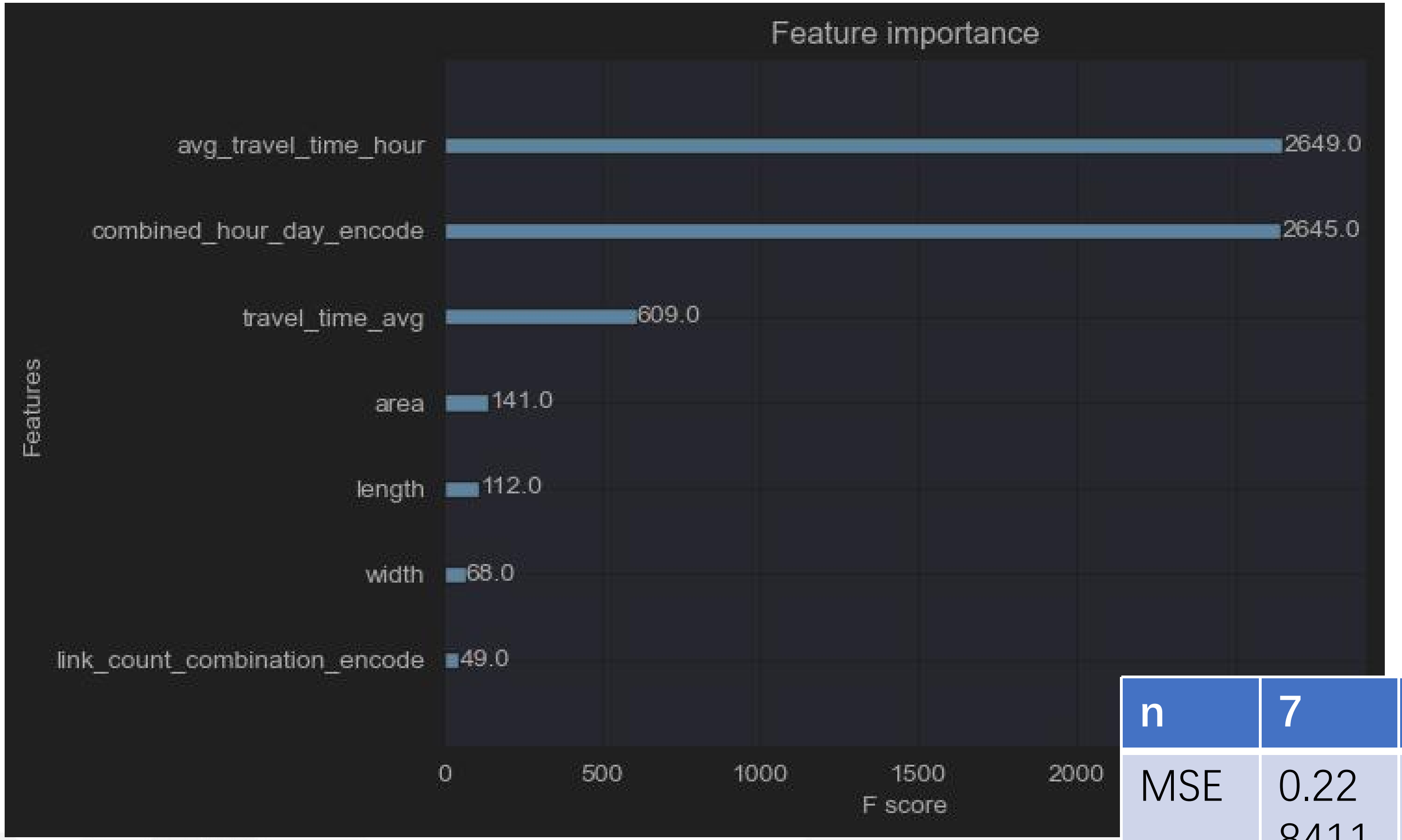
| | Complete Features | Without Temporal Features | Without Spatial Features |
|---|---|---|---|
| MSE | 0.23 | 0.36 | 0.23 |

- **Conclusion**

  temporal features is crucial for enhancing the predictive performance of our model

# Model

## XGBoost



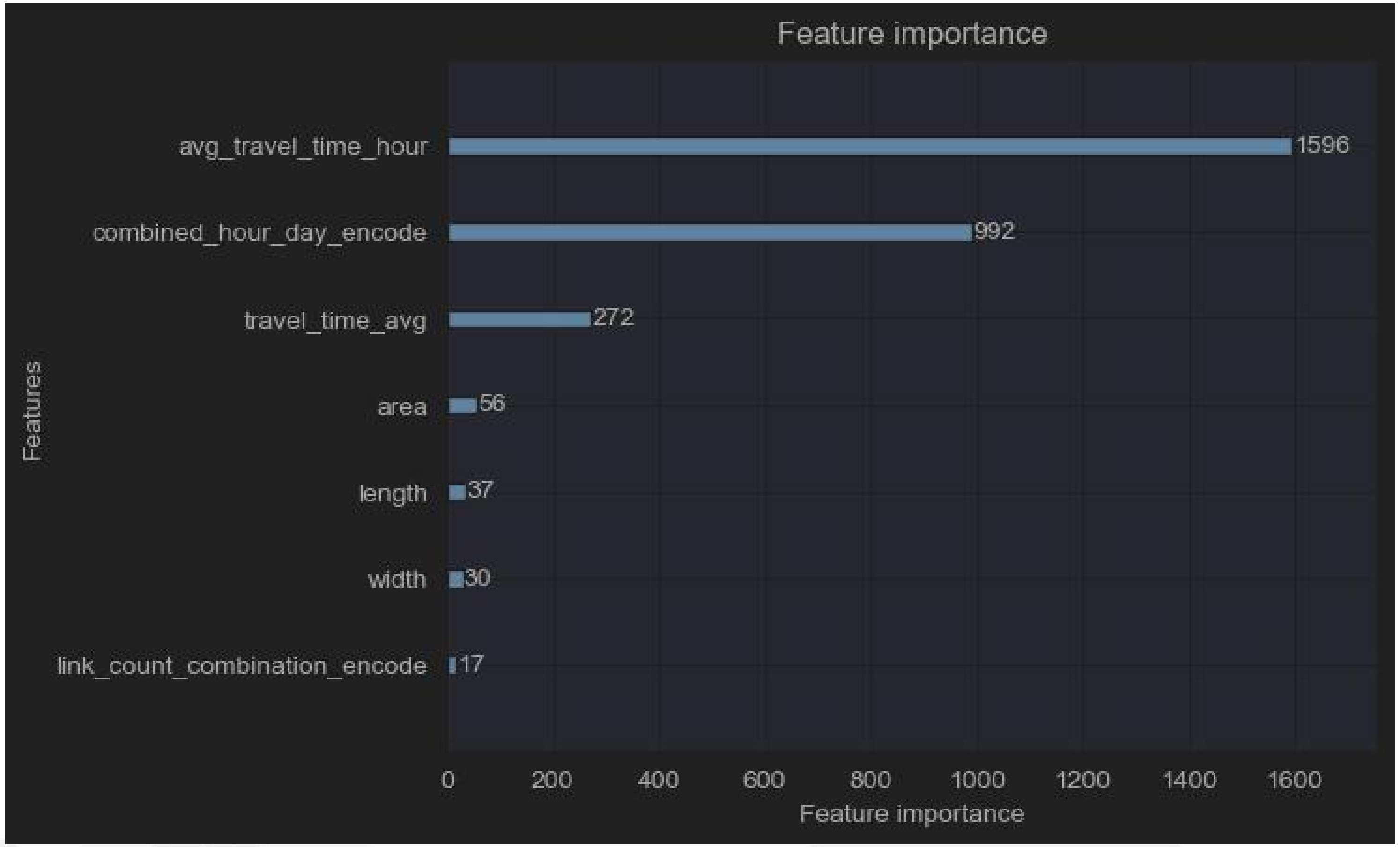

- **Feature importance**

- Time characteristics have a more obvious role
- avg_travel_time_hour and combined_hour_day_encode nearly same importance

| n | 7 | 6 | 5 | 4 | 3 | 2 | 1 |
|---|---|---|---|---|---|---|---|
| MSE | 0.2284115 | 0.2284084 | 0.2284135 | 0.2284167 | 0.2287174 | 0.2287165 | 0.2287131 |

# Model

## Lightgbm




- **Feature importance**
- Time characteristics still have a more obvious role
- this model avg_travel_time_hour more improtance

# Model mixing

| | xgboost | lightgbm | simple_average | weighted_average | stacking |
|---|---|---|---|---|---|
| MSE | 0.22841158189028937 | 0.22844817843263165 | 0.2284214615399772 | 0.2284254579390 7068 | 0.2284042149418 2697 |

- **Conclusion**

**stacking is the points where we can try further**

# THANK YOU