Traffic Jam Prediction

Gibran Brahmanta P.
gibranbrahmanta@gmail.com

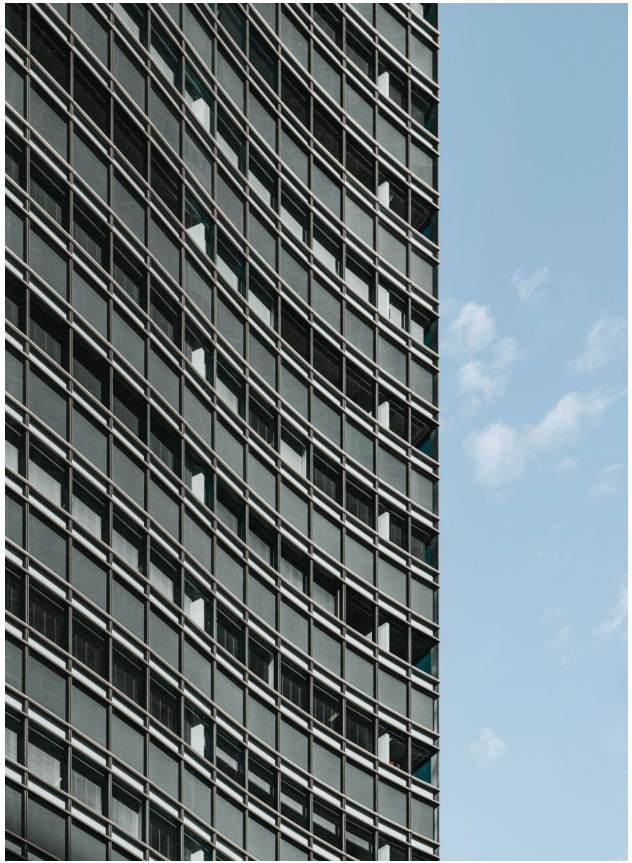


Table of Contents

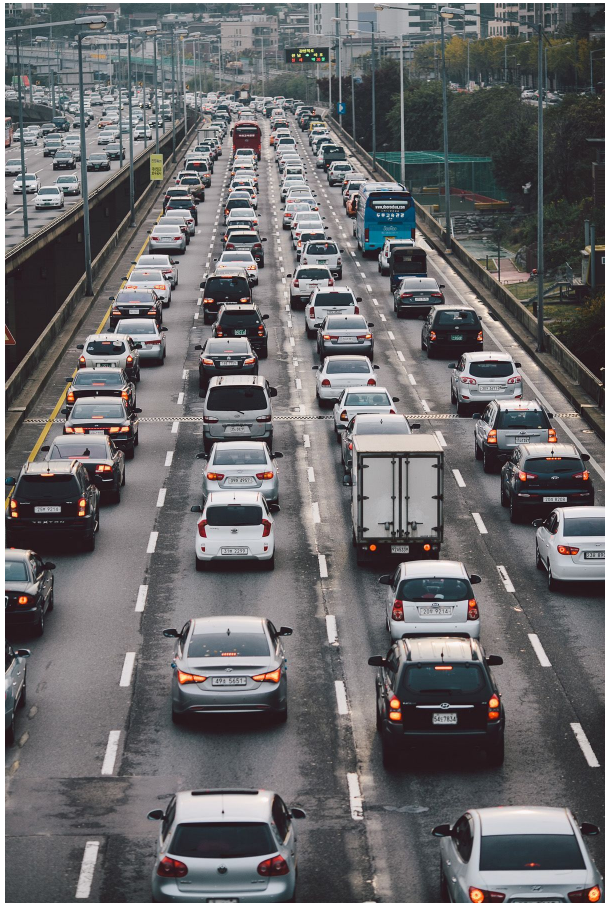
01

Business Understanding

Traffic Jam [noun]

“a large number of vehicles close together and unable to move or moving very slowly”

—Cambridge Dictionary



Traffic Jam Effects on Indonesia

Economic

Cause Losses of **IDR 71.4 Trillion** Per Day

Fuel Consumption

2.2 Million Liters of Fuel 'evaporates' Per Day

Environment

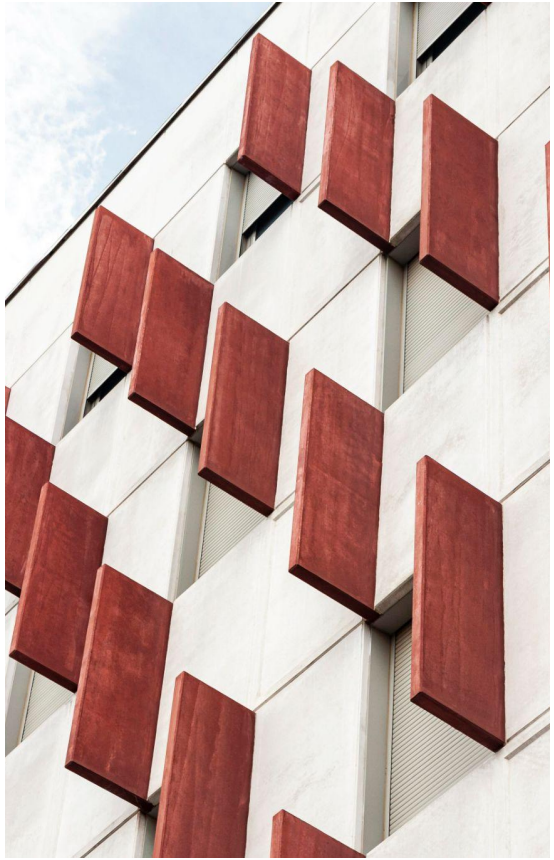
PM2.5 value is **five times** as much as the **PM2.5 annual mean guideline** established by the WHO

Health

People being more vulnerable to **stress, muscle fatigue, and repository problem**

How to reducing the effects of Traffic Jam?

Obviously, by preventing the traffic jam



Problem Statement

How can we **predict** the **traffic jam condition** on **a street** in a city?

Solution Approach

- By using the **historical data of street condition**, **weather condition** (especially rain), and **gap between current date and nearest holiday**, we could **predict the traffic jam condition** on a street in a specific timestamp
- Using the prediction result, **we could know how the condition of the traffic jam on a street before it's being happen**
    - So, the preventive approach to reduce the traffic jam could be done
- For this project, **Bogor is being used as a city for doing the pilot project** before the project is being implemented on many cities
    - By doing that, we could know the impact, advantages, and disadvantages of the project at a lower level
    - So if the project is being implemented on a higher level, the project could create more positive impact

Data Understanding

- To solve the problem, a dataset from satellite navigation software on smartphones (Waze) is provided. **The dataset has three types of data**:
  - **Jams**: Traffic jam data based on current condition
  - **Irregularities**: Street data based on historical data
  - **Alerts**: Flags to indicate whether a special occasions occur or not on a street
- **All of those informations has time-related attribute**, therefore it can concluded that all of those informations were a **time series data**

EDA - Alerts Dataset

- **< 10% of the rows have null value** on one of its attributes
  - **Those rows were being dropped** and not being included in the analysis
- There were **234 streets on this dataset**
- **For several streets**, there were **multiple value on 'avg_location'** attribute
  - In order to achieve the uniqueness on that attribute, mean value of each coordinate were used on each street that has more than one value on that attributes
  - Also, variance related to each 'avg_location' coordinate on each street is being computed to check the consistency of the dataset
- Overall, **variance value that being computed can be considered low**, so we can conclude that the dataset were consistent.
  - Because of that, the **dataset can be the source of truth of the street location**

EDA - Jams Dataset

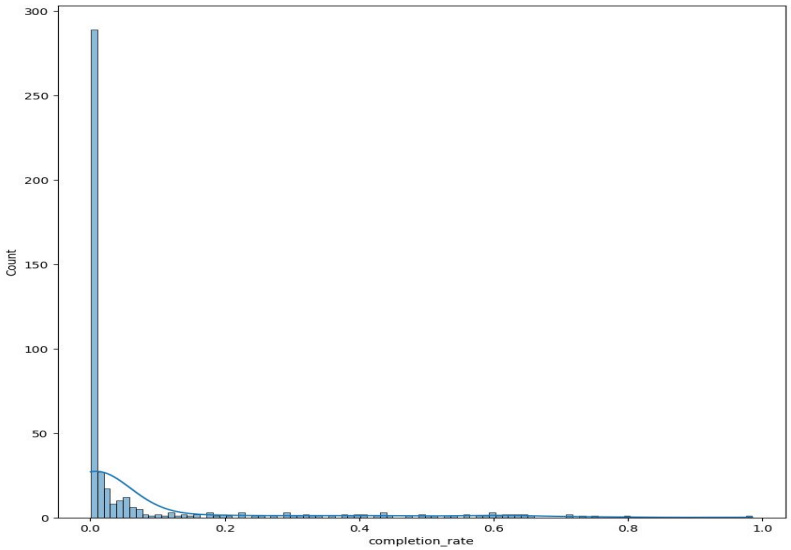
- **+/- 1% of the rows have null value** on one of its attributes
  - **Those rows were being dropped** and not being included in the analysis
- There were **454 streets in the dataset** and contains **100529 rows**
- Timespan:
  - **Min Timestamp**: 2022-07-06 00:00:00
  - **Max Timestamp**: 2022-09-06 00:00:00
- The dataset **can't be considered as a full historical data**, because the dataset **only cover +/- 14%** of the expected full historical data
  - The number of expected full historical data are being computed by multiply number of days, number of hour on each day (24) and number of streets

EDA - Jams Dataset (2)

- There were **several street** that **has more than one row** of data **in a timestamp**
  - **Expectation**: For each location and timestamp, there were only one row of data
  - **In order to meet the expectation**, these steps were conducted:
    - For each location and timestamp that only has 1 row, insert it into the final dataset
    - For each location and timestamp that has more than 1 row, insert the row that has highest number of report. If there were several rows that have the highest number of report, pick randomly one row
    - **Used Assumption**: Higher number of reports indicates the information was more reliable
  - **Remaining rows** after those process: **51848** rows

EDA - Jams Dataset (3)

- **Most of the street** has a **low number of rows**.
  - **Completion Rate**: number of rows on the data/number of rows that is being needed to create a full historical data
  - Completion Rate Stats:
    - min = 0.000672
    - 25% percentile = 0.001344
    - 50% percentile = 0.004032
    - 75% percentile = 0.038306
    - max = 0.984543



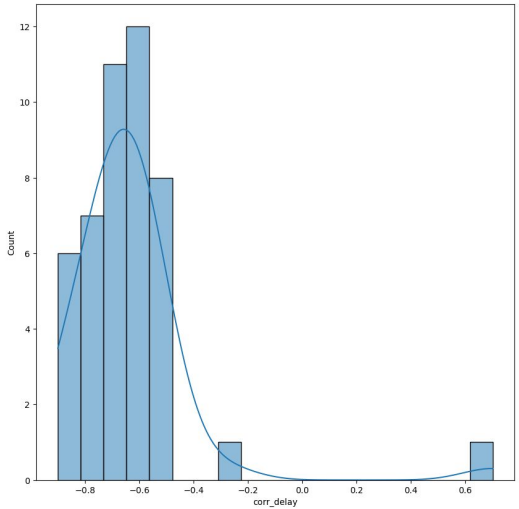
Completion Rate Histogram

EDA - Jams Dataset (4)

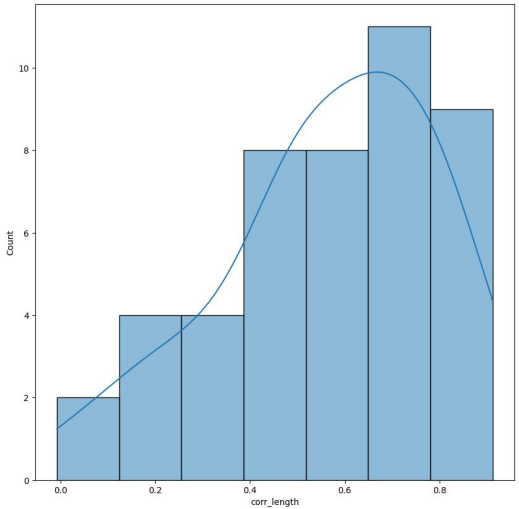
- Related to previous information, **further analysis and modeling only included several streets** that have completion rate **more than 90% percentile of completion rate data**
  - It is related to the objective of the modeling, that is to prevent traffic jams on Bogor
    - To create bigger impact related to that, **the prevention process can be done on several streets that are often jammed**
  - Also, **if all of the data are being used**, then the dataset will be an more **imbalanced dataset**. It is because, on the modeling part, the **jams dataset will be converted into full historical data**. Whereas **the missing data will be considered** as data with **jam level equals to 0**
    - That conditions can give negative effects on the model performance
- After filtering out the data from all of unused street data, **the remaining rows** of the data is **37001**

EDA - Jams Dataset (5)

- There were **several attributes that has medium-high correlation** on each street
  - median_speed_kmh & median_length (with average equals to 0.566587)
  - median_speed_kmh & median_delay (with average equals to -0.637149)



Correlation Histogram Between median_speed_kmh & median_delay



Correlation Histogram Between median_speed_kmh & median_length

EDA - Jams Dataset (6)

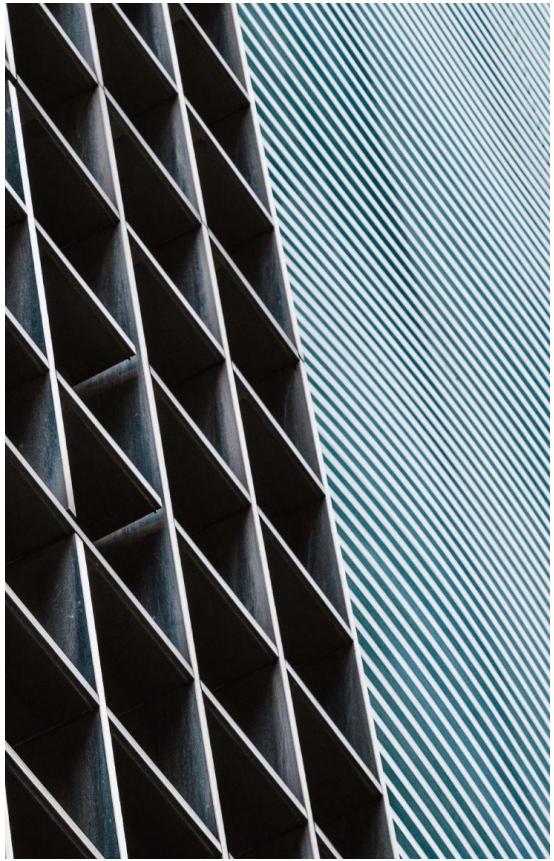
- **Coordinates of each street also being computed** on this dataset
  - For each rows, calculate the mean and variance of longitude and latitude
  - After that, calculate the mean value of mean and variance of longitude and latitude on each street
  - At the final result:
    - The mean value of latitude and longitude data on each street will be treated as the location of the street
    - The variance value of each street will indicates the consistency of the location data
- **Mean value of variance on each street can be considered low**, so we can conclude that the dataset were consistent.
  - Because of that, **the dataset can be the source of truth of the street location**

EDA - Irregularities Dataset

- There were **two useless attributes**:
  - **cause_type**: only has NULL value
  - **median_jam_level**: has exact same value with jam_level on each row
- There were **85 streets** in the dataset and contains **4051 rows**
- The dataset has time span
  - **Min Timestamp**: 2022-07-06 09:00:00.000
  - **Max Timestamp**: 2022-09-04 21:00:00.000
- The dataset **can't be considered as a full historical data**
- **Coordinates of each street also being computed on this dataset** with the exact same process with Jams dataset
  - **Mean value of variance on each street can be considered low**, so we can conclude that the dataset were consistent.
  - Because of that, the **dataset can be the source of truth of the street location**

EDA - Comparison between Dataset

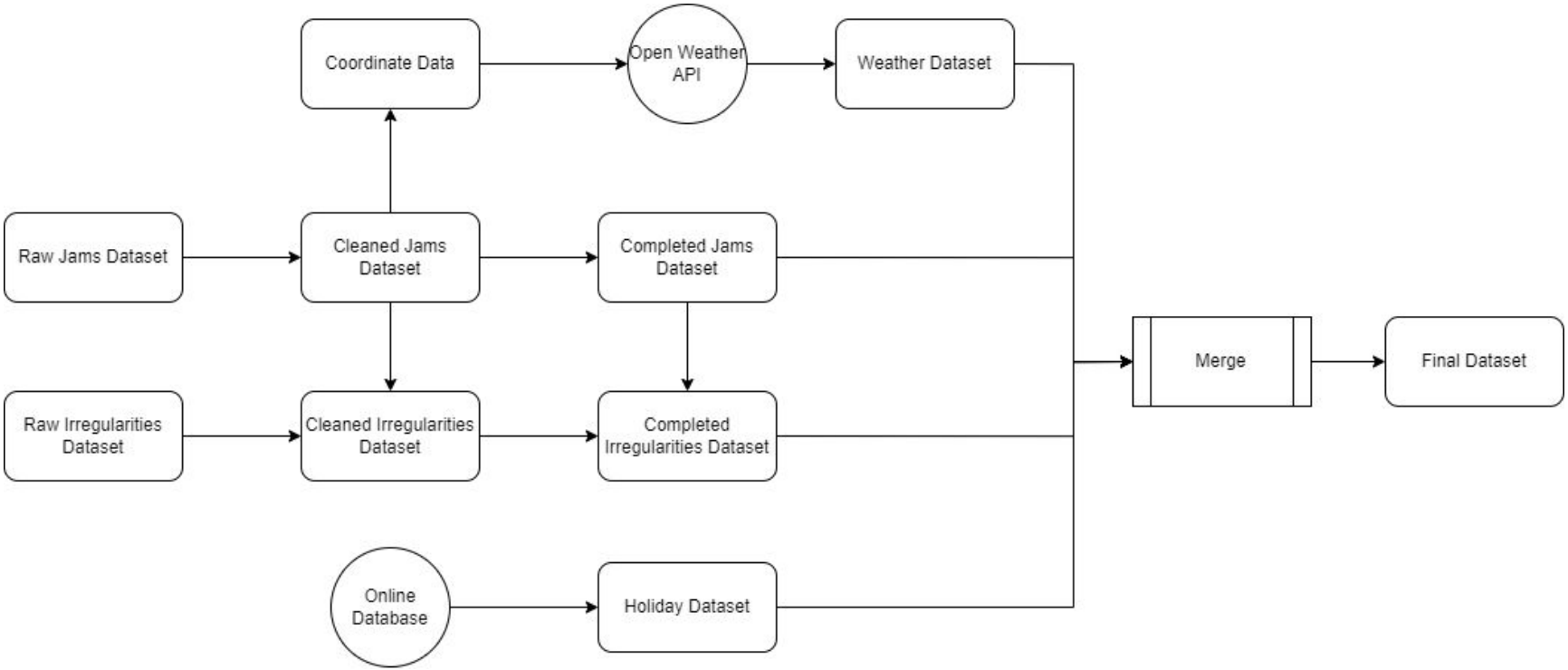
- **There were several street that being used for further analysis** on modeling that **don't appear on the irregularities dataset**. So that dataset **can't be used as source of truth of street location**
- Meanwhile, **alert dataset have all used streets** and it has relatively low difference if compared to location data at jams dataset, but **the variance value on that dataset is bigger than variance value on the jams dataset**
    - So, **the jams dataset will be used as the source of truth about the street location** in order to maintain the consistency
- There were **several differences on median_speed attribute on Jams Dataset and Irregularities Dataset**.
    - For further analysis, **median_speed attribute on Jams dataset will be used because it is more complete** and based on the **modeling objective that closely related to jams**



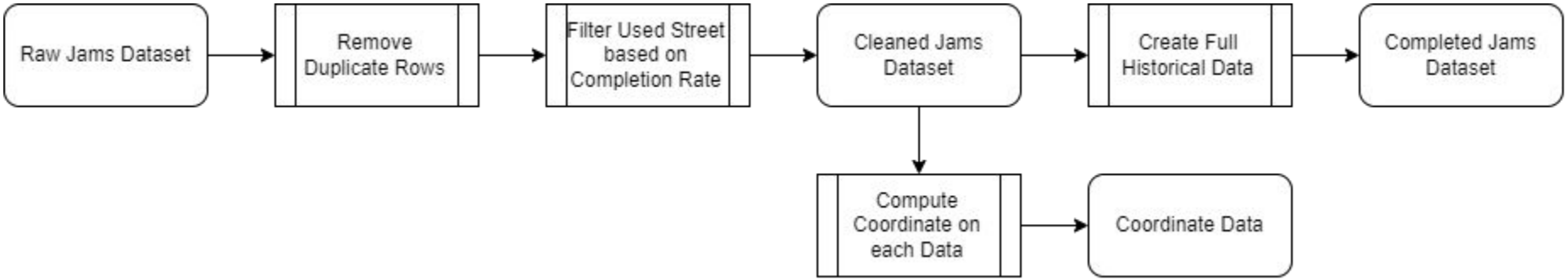
02

Data Modeling

Data Preprocessing Flow



Jams Data Preprocessing Flow



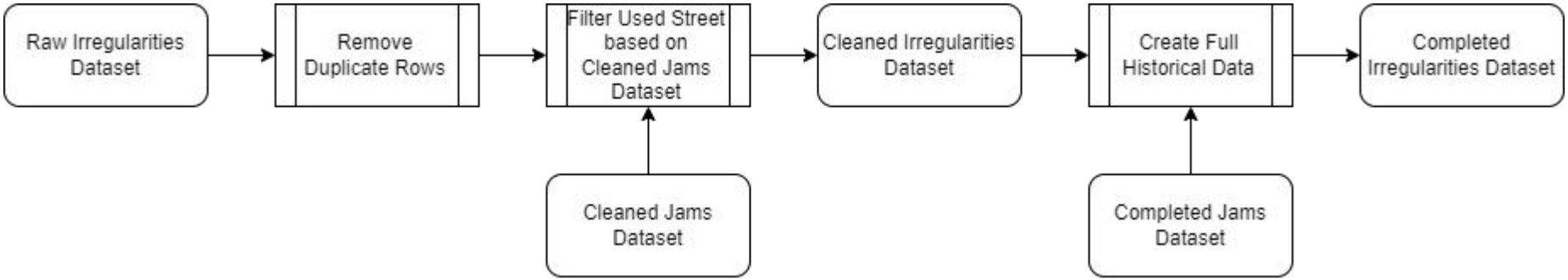
Jams Data Preprocessing

- Remove rows that has duplicated (street, timestamp) data
  - **Note**: If there were duplication, keep row that has highest vote count
- **Filter** used street based on **completion rate**
  - Used street are streets that has **completion rate >= 90 percentile** of completion rate distribution
- Compute the **coordinate of each used street** by calculating the mean value of latitude and longitude on each related data
  - Result from this step will be used later on to **create the weather dataset**

Jams Data Preprocessing (2)

- On each street that are being used, **create data for all of those timestamps that doesn't appear in the original dataset** in order to create a full historical data. Below are the description on each used attributes on the new created data
  - current_timestamp: new timestamp that doesn't appear in the original data
  - street: name of the street
  - level: 0, by using assumption that a timestamp that doesn't appear in the original means that there were no traffic jam on that time
  - median_speed: computed by using the domain knowledge about the speed comparison on each jam level. Randomized value also being included on this step to create more variative dataset
  - median_length: computed by using the comparison between median_speed and median_length on the original dataset
  - median_delay: computed by using the comparison between median_speed and median_delay on the original dataset

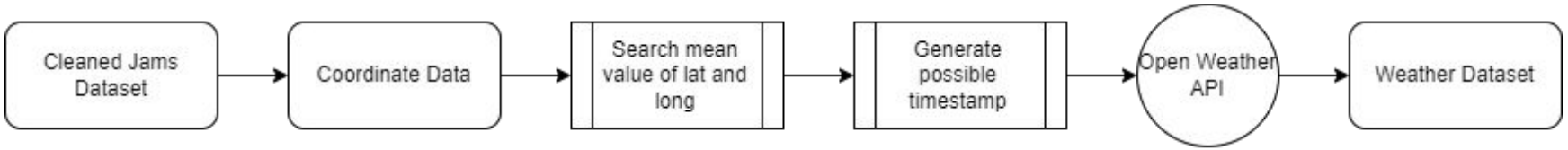
Irregularities Data Preprocessing Flow



Irregularities Data Preprocessing

- Remove rows that has duplicated (street, timestamp) data
  - **Note**: If there were duplication, keep row that has highest vote count
- **Filter** used street based on **Cleaned Jams Data**
- On each street that are being used, **create data for all of those timestamps that doesn't appear in the original dataset** in order to create a full historical data. Below are the description on each used attributes on the new created data
  - current_timestamp: new timestamp that doesn't appear in the original data
  - street: name of the street
  - median_regular_speed: computed by searching the median value of 'median_speed' in completed jams dataset that related to the processed street and has timestamp before or equal the processed timestamp
  - median_delay_seconds: computed by searching the median value of 'median_delay' in completed jams dataset that related to the processed street and has timestamp before or equal the processed timestamp

Weather Data Preprocessing Flow

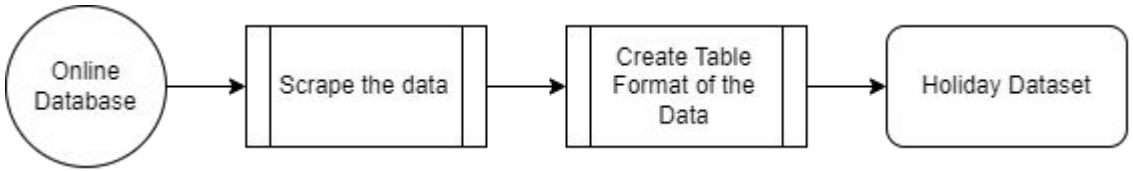


Weather Data Preprocessing

- By using the **coordinate data that has been got from jams dataset**, search for the **mean value of latitude and longitude** between all of the used street
    - These method were used in order to **reduce the number of API Call** that has to be made to get the weather data. The number of API Call has to be limited due to lack of resource (money, because the API has limited number of free request)
    - It is can be considered safe because from the EDA result, we could conclude that **a city tends to have same weather conditions across all of the streets**
- **Generate all possible timestamp** based on start timestamp and end timestamp
- For each timestamp, get the weather dataset using the OpenWeather API

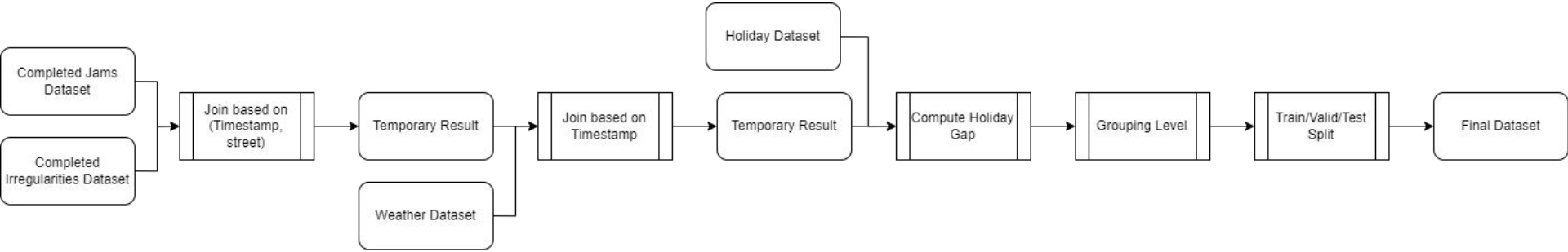
Holiday Data Preprocessing

- Scrape holiday data from this online database, and
- Create the table format of the data



Holiday Data Preprocessing Flow

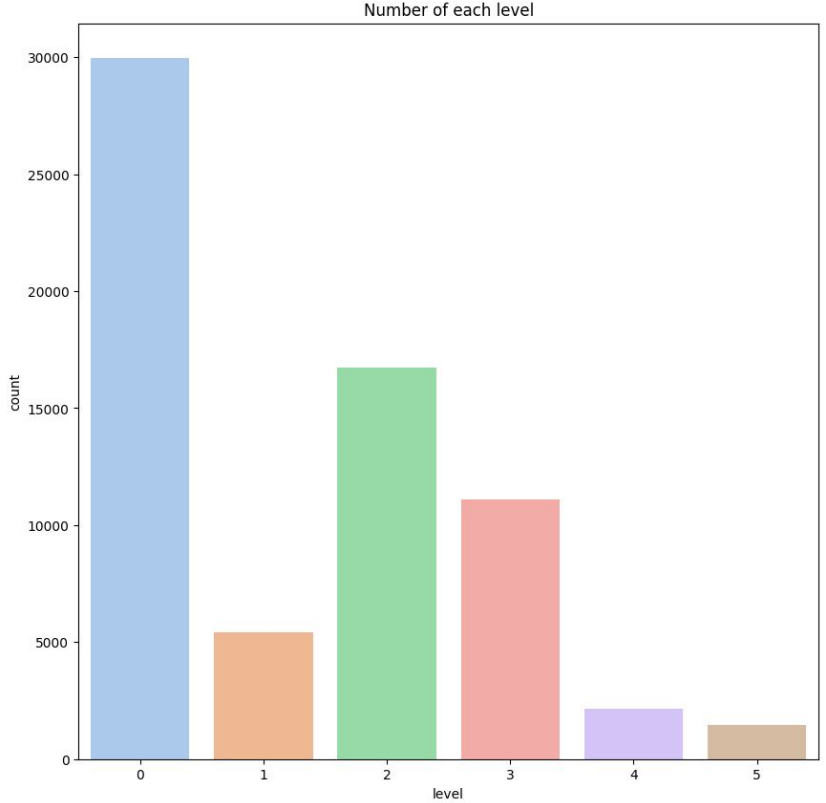
Merging Dataset Flow



Merging Dataset

- **Jams and Irregularities dataset** are being **joined** based on (timestamp, street)
- **Weather data** is being **merged** with the result from previous step **based on timestamp**
- **Holiday data** is being merged with the result from previous step by **computing the day gap between the timestamp and the nearest holiday date**. The result will have range minimum -1 and maximum 7 where it tells how many days before/after the nearest holiday on a timestamp. -1 if current timestamp doesn't have any nearest holiday in a one week time span
- **Level** are **being grouped into several group** in order to reduce the imbalanced
- **Several additional attributes** such as 'time_series_split' and 'classification_split' also being created in order **to indicate the role of each row on the modeling** (train/valid/test)

Why level must be grouped?



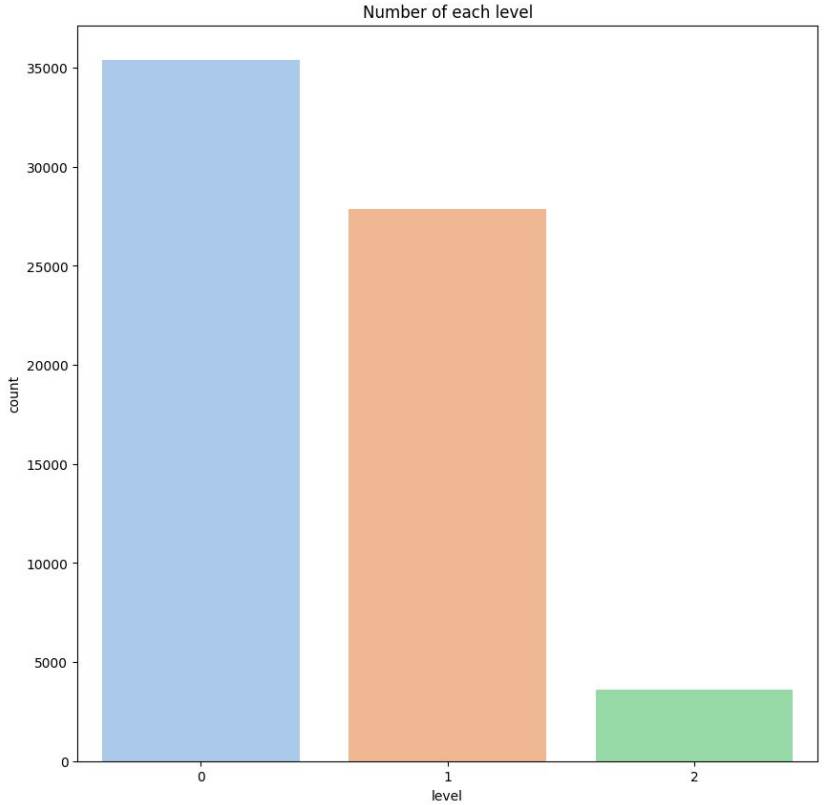

```
Train set
0    24208
2    13471
3     8785
1     4374
4     1671
5     1173
Name: level, dtype: int64
Valid set
0    3411
2    1931
3    1308
1     621
4     289
5     168
Name: level, dtype: int64
Test set
0    2328
2    1340
3    1019
1     430
4     193
5     118
Name: level, dtype: int64
```

Imbalanced Dataset!

Grouping Process

- Due to imbalanced data, target feature on the classification task (level) is being grouped into several group to reduce the number of class that are going to be predicted
- Level grouping
  - Group 0: Low traffic jam (level 0-1)
  - Group 1: Medium traffic jam (level 2-3)
  - Group 2: High traffic jam (level 4-5)

Grouping Result




```
Train set
0    28582
1    22256
2     2844
Name: level, dtype: int64
Valid set
0    4032
1    3239
2     457
Name: level, dtype: int64
Test set
0    2758
1    2359
2     311
Name: level, dtype: int64
```

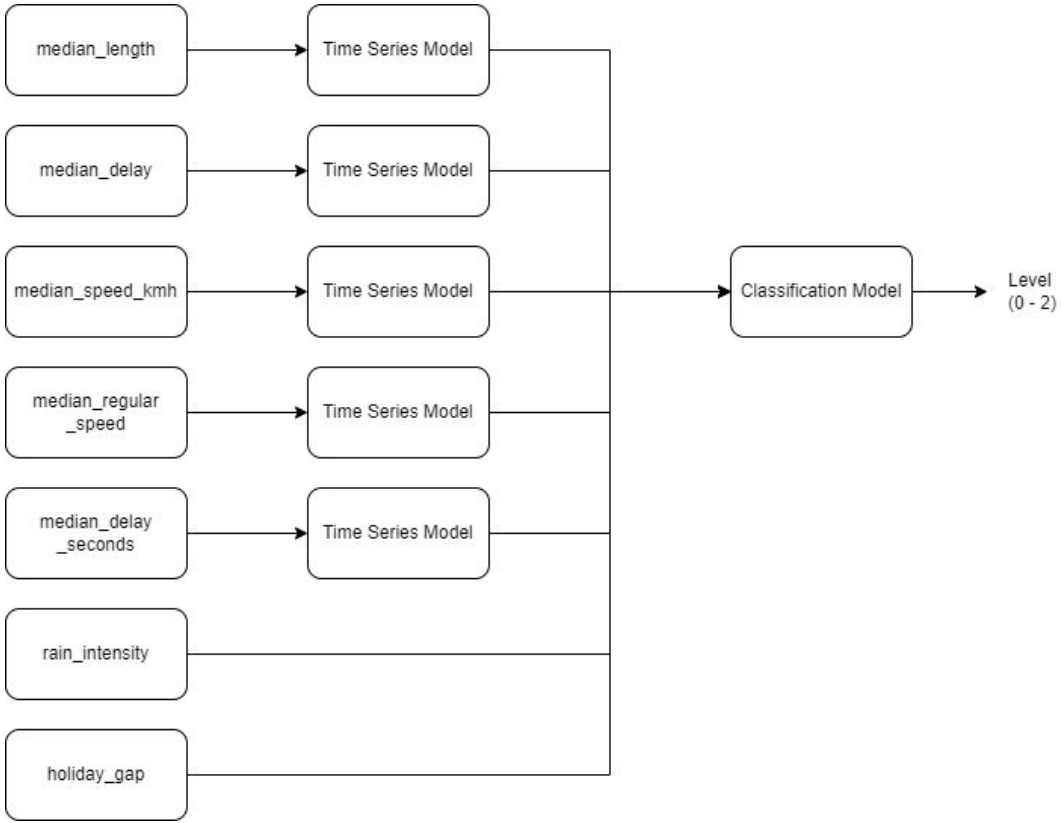
Machine Learning Modeling

- To predict the traffic jam level, **a two-steps model was used**. There were several attributes that are being used to predict traffic jam level such as
  - 'median_length',
  - 'median_delay',
  - 'median_speed_kmh',
  - 'median_regular_speed',
  - 'median_delay_seconds',
  - 'rain_intensity', and
  - 'holiday_gap'.
- Those attributes has various type, **for the 5 first attributes** can be indicated as a **time-series related attributes** and **the other** were **regular numerical attributes**.

Machine Learning Modeling (2)

- The model has **input** a **timestamp** and a **street name** related to a city.
- The **output of the model is traffic jam level** on related street and timestamp that has **3 possible value (0, 1, and 2).**
- Below is the machine learning model flow to predict a traffic jam level
    - At first, the model will predict all of the time-series related attributes on the related street and timestamp.
    - After that, the result from the previous step were used as several attributes to predict the traffic jam level with the other attributes.
        - The non time-series related attribute value can be got from internal database and weather prediction API.

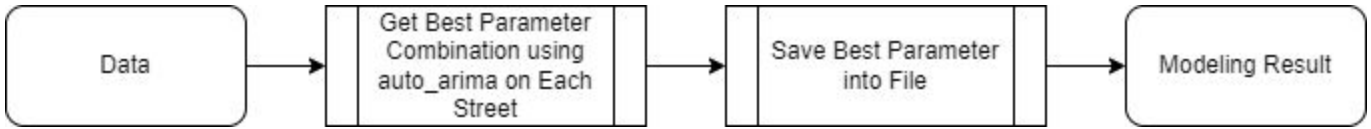
Machine Learning Model Flow



Time Series Modeling

- All of the **time-series related attributes** have **unique model configuration** on each used **street.**
- The used model is **ARIMA**.
  - At first, **SARIMA model also want to be used as a comparison** but **due to lack of computation** resource so the model **couldn't be used**.
  - Below are the specification of **time series model partition** for modeling:
    - train_set: Start from 2022-07-06 09:00:00.000
    - test_set: Start from 2022-08-24 00:00:00.000
- At the **training phase**, **several parameters related to the model** that have best performance were being **searched using the auto_arima package**. The result of the modeling phase are stored into several files to accommodate the inference process

Time Series Modeling Flow



Classification Modeling

- Below are the **several models were used on the experiment** to create the classification model
  - Linear Regression
  - Support Vector Machine
  - Naive Bayes
  - Decision Tree
  - Random Forest
  - LightGBM
  - XGBoost

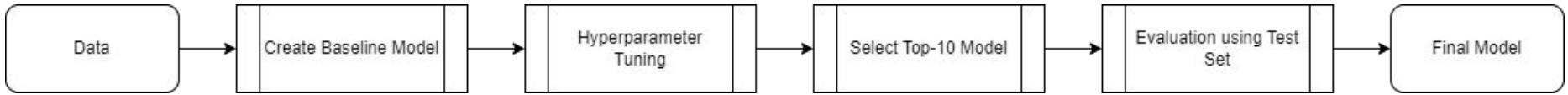
Classification Modeling (2)

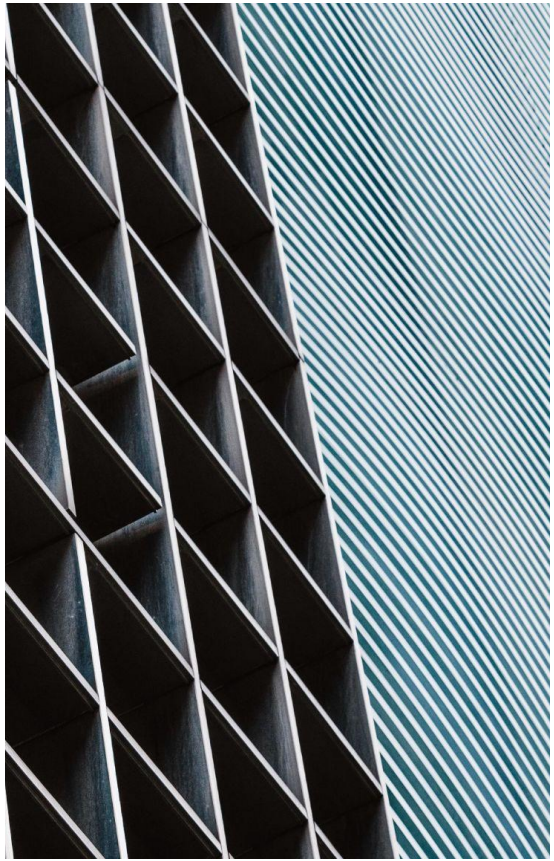
- For all of those models, the **baseline model** were created by train those models only **using the default parameter** that has been determined by the used package and do the **evaluation process using test dataset** to measure the performance of each models in a basic situation.
- After that, the **hyperparameter tuning process** were carried out by **training all of those models** using several hyperparameter combination
    - On the hyperparameter tuning process, all of the models were **trained using the train dataset**, and for all of the models that have been trained, **evaluation process using validation dataset** were carried out to get **top-10 models that have best performance**
    - Those **top-10 models are being used for next evaluation process using test dataset** to determine which **model and parameter combination that has best performance**.

Classification Modeling (3)

- Below are the specification of **classification model partition** for modeling:
  - train_set: Start from 2022-07-06 09:00:00.000
  - valid_set: Start from 2022-08-24 00:00:00.000
  - test_set: Start from 2022-08-31 00:00:00.000
- When doing the **evaluation process using validation and test dataset**, all models were using the **prediction result from the time series model as the input**.
  - From that process, it is expected that the final model **can be robust** because it involves the error that being made by the time series model.
- On the classification modeling, not all of the models are being saved into a .pkl file. **Only baseline and top-10 models that based on hyperparameter tuning process** that are being **saved**.

Classification Modeling Flow





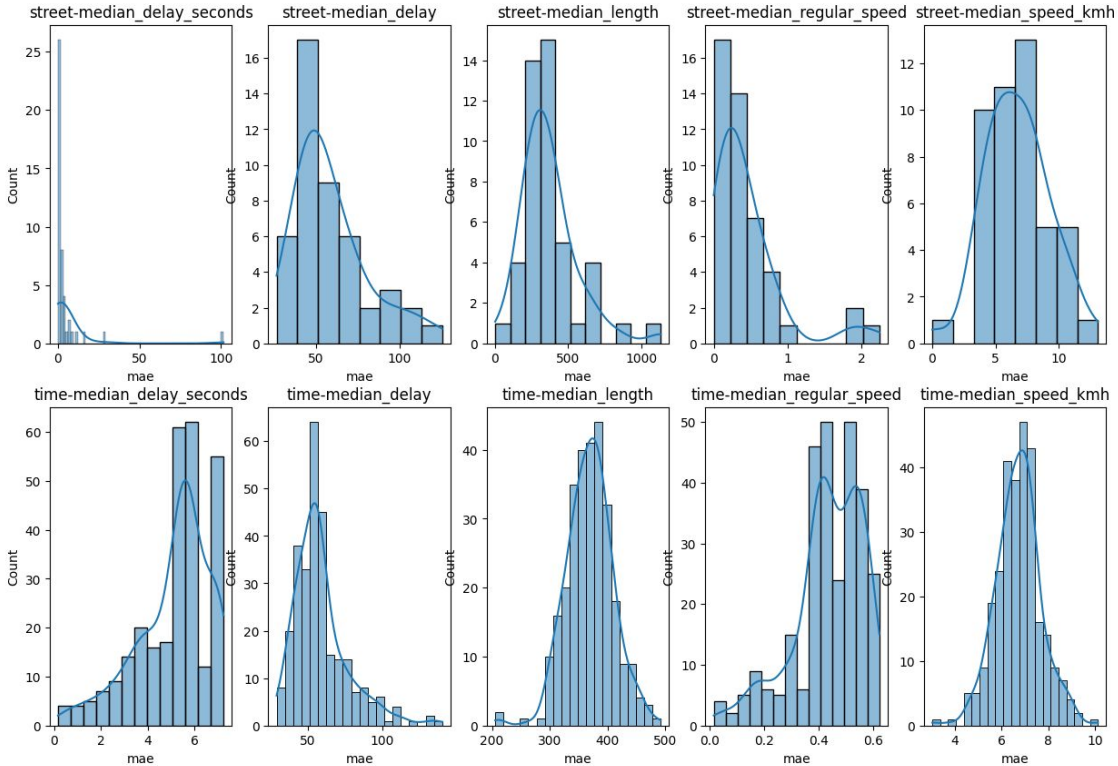
03

Evaluation

Time Series Model Evaluation

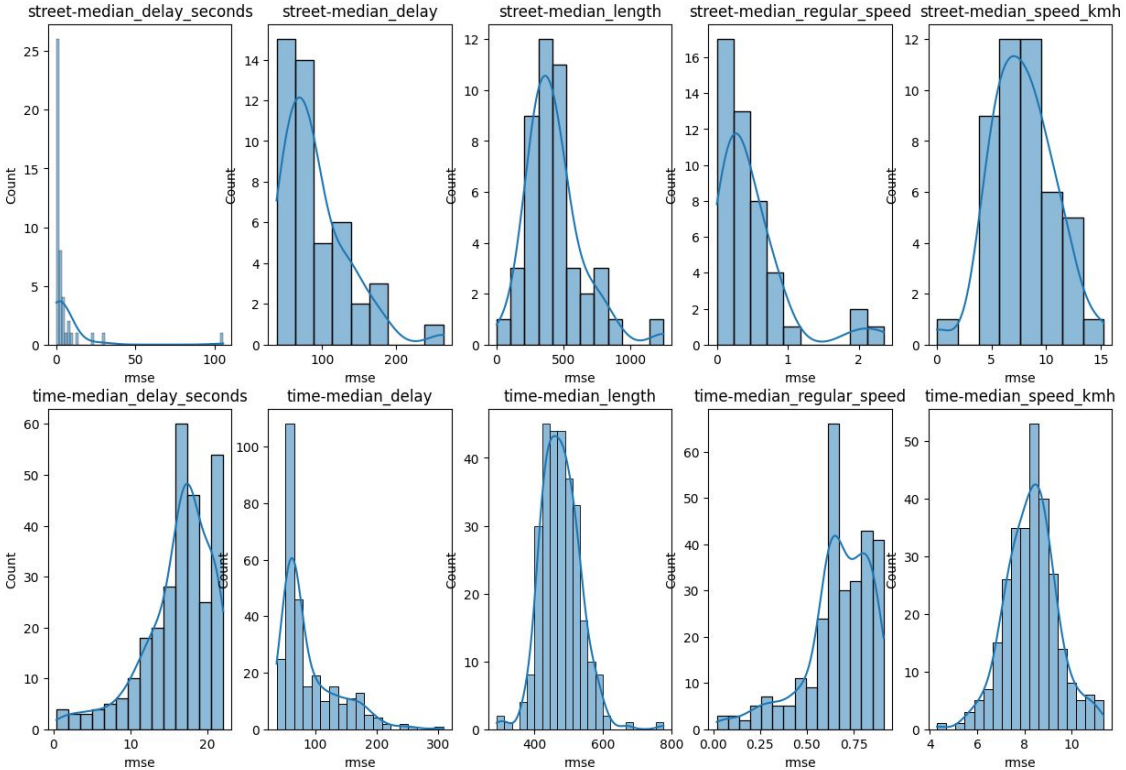
Feature	RMSE	MAE
median_delay	100.823259	58.859931
median_delay_seconds	16.855420	5.136940
median_length	479.631479	369.180811
median_regular_speed	0.690851	0.435304
median_speed_kmh	8.366097	6.729041

Time Series Error Analysis



MAE Value Histogram based on Street and Time

Time Series Error Analysis (2)

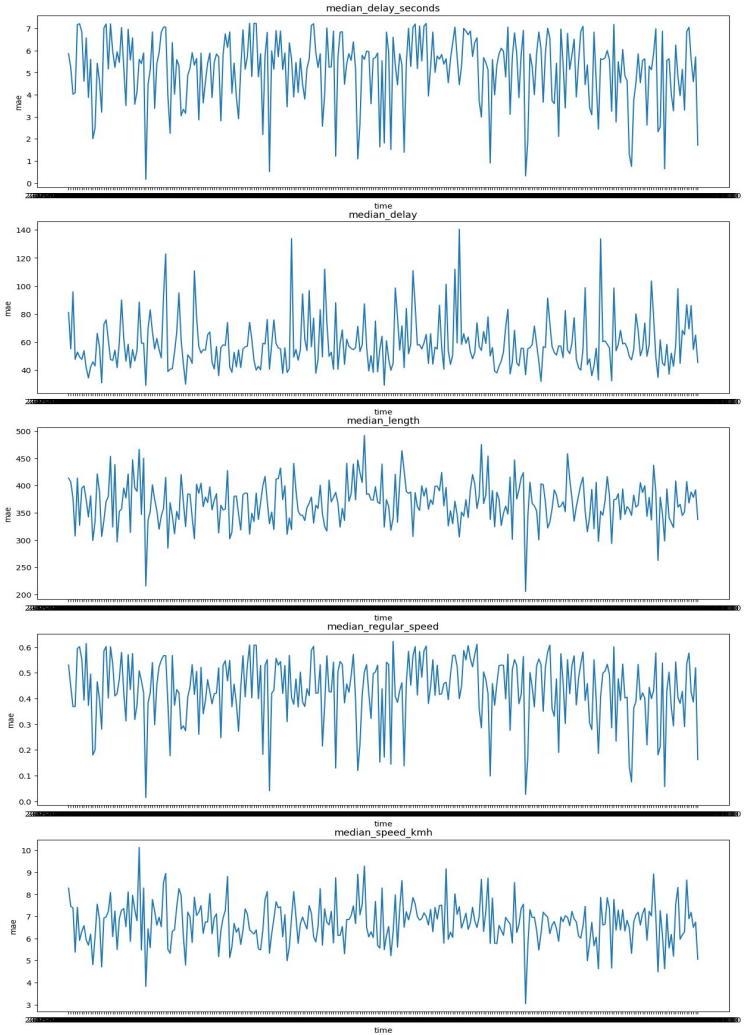


RMSE Value Histogram based on Street and Time

Time Series Error Analysis (3)

Line plot of MAE value (from the top image to the bottom)

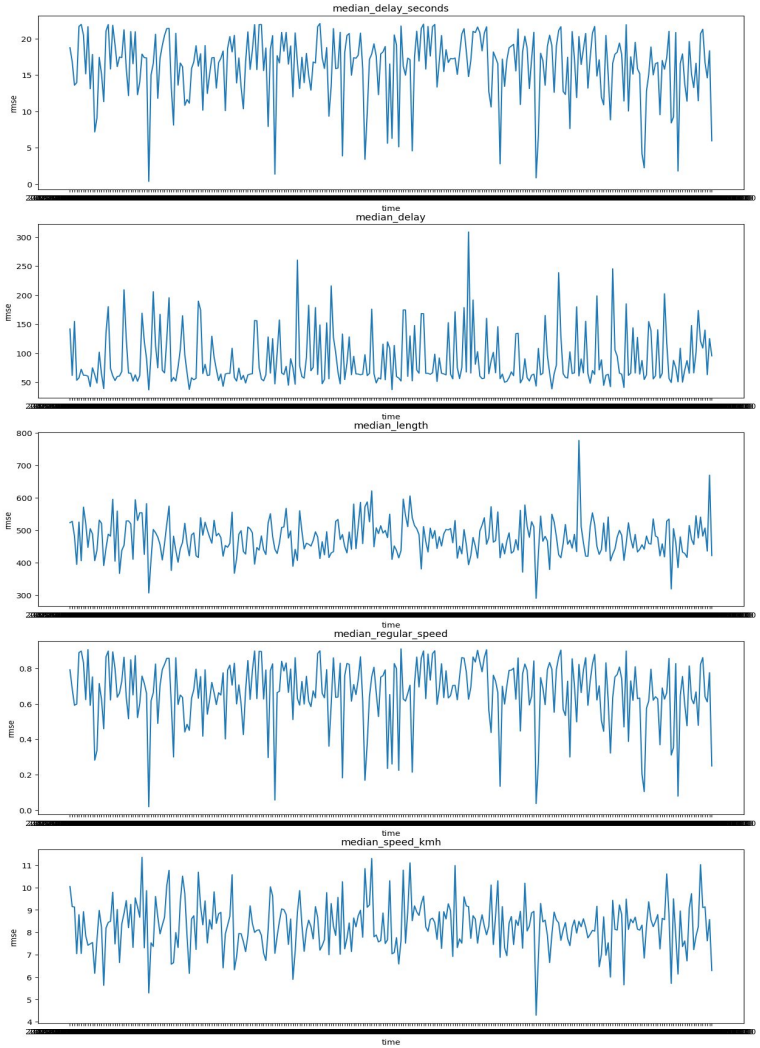
- 'median_length',
- 'median_delay',
- 'median_speed_kmh',
- 'median_regular_speed',
- 'median_delay_seconds'



Time Series Error Analysis (4)

Line plot of RMSE value (from the top image to the bottom)

- 'median_length',
- 'median_delay',
- 'median_speed_kmh',
- 'median_regular_speed',
- 'median_delay_seconds'



Time Series Error Analysis (6)

feature	min	mean	q25	q50	q75	max
median_length	True	False	False	False	False	False
median_delay	True	False	True	False	False	False
median_speed_kmh	True	False	False	False	False	False
median_regular_speed	True	False	False	False	False	False
median_delay_seconds	False	False	False	False	False	False

MAE vs Each Feature Distribution

feature	min	mean	q25	q50	q75	max
median_length	True	False	False	False	False	False
median_delay	True	True	True	True	False	False
median_speed_kmh	True	False	False	False	False	False
median_regular_speed	True	False	False	False	False	False
median_delay_seconds	True	False	True	False	False	False

RMSE vs Each Feature Distribution

Note: True if metric value > distribution value of a feature, otherwise False

Time Series Error Analysis (7)

- For all features, the existing **prediction error isn't a exploding error type of error**. It is because the error graph doesn't show any trend
- At the histogram, **several features has different distribution at time and street aspect**
    - **Conclusion**: The prediction result **wasn't being precise on the time aspect**, there were several features that has big value of error. But for the **majority streets**, they have **several cases that made the prediction being precise**, and it made the overall prediction result went lower
- From the comparison tables, we can conclude that the **error that are being happened in the prediction can still considered safe** because all of the metric value is less than the mean value of each feature

Classification Model Evaluation

Model Name	Accuracy	Precision	Recall	F-1 Score
DecisionTree	0,598	0,724	0,525	0,569
LightGBM	0,610	0,732	0,535	0,582
LogisticRegression	0,599	0,724	0,531	0,580
NaiveBayes	0,503	0,262	0,494	0,341
RandomForest	0,605	0,728	0,532	0,579
SVM	0,571	0,708	0,500	0,531
XGBoost	0,607	0,729	0,534	0,582

Evaluation Result on Baseline Model

Note: Full Result of Classification Model Evaluation can be accessed here

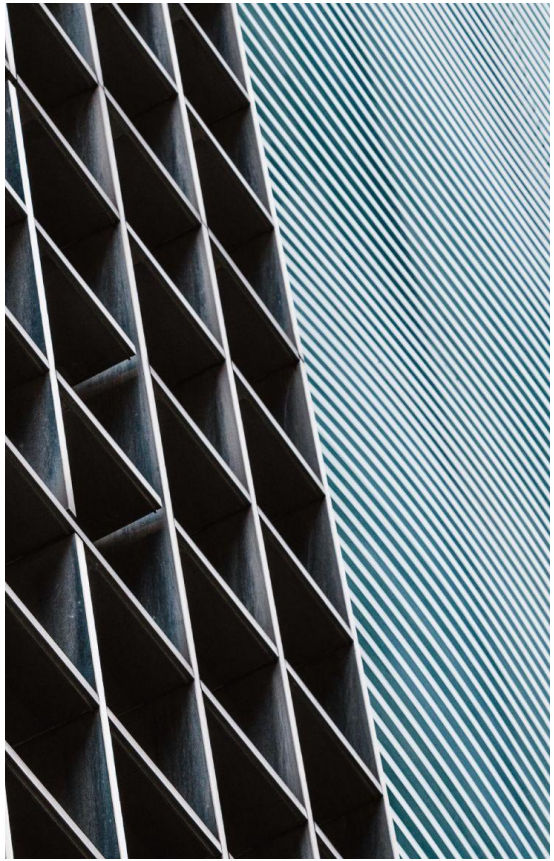
Classification Model Evaluation (2)

Model Name	Best Param	Accuracy	Precision	Recall	F-1 Score
DecisionTree	(criterion_entropy)(splitter_best)(max_depth_10)(max_features_auto)	0,626	0,742	0,550	0,599
LightGBM	(learning_rate_0.01)(n_estimators_100)(max_depth_50)(objective_multiclass)	0,621	0,740	0,544	0,591
LogisticRegression	(penalty_l1)(C_1.0)(solver_saga)	0,606	0,729	0,537	0,586
NaiveBayes	(alpha_1,0)(norm_True)	0,518	0,309	0,474	0,369
RandomForest	(criterion_log_loss)(n_estimators_500)(max_depth_10)(max_features_auto)	0,621	0,740	0,544	0,591
SVM	(C_1.0)(multi_class_crammer_singer)(loss_squared_hinge)	0,600	0,728	0,536	0,584
XGBoost	(n_estimators_500)(max_depth_10)(learning_rate_0.001)	0,628	0,746	0,547	0,593

Classification Model Evaluation Interpretation

The evaluation result of the best model could be interpreted as:

- If there were 100 data point (street, timestamp) that are being predicted, then the model can accurately predict the traffic jam level of 62 data point
- If there were 100 data point (street, timestamp) that have prediction result as high level traffic jam, then 72 data point were really have high level traffic jam
- If there were 100 data point (street, timestamp) that have high level traffic jam, then the model can accurately predict the traffic jam level of 55 data point



04

Result

Prediction Result and How to Use it

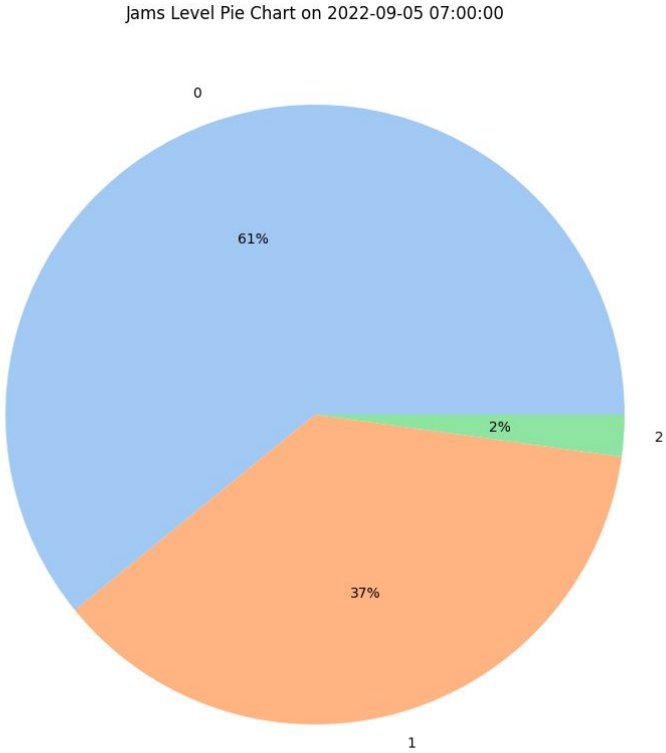
- Final prediction result has form of table, where it have several columns such as
  - street
  - time
  - 'median_length',
  - 'median_delay',
  - 'median_speed_kmh',
  - 'median_regular_speed',
  - 'median_delay_seconds',
  - 'rain_intensity',
  - 'holiday_gap'.
  - Level
- The prediction result could be stored into a Data Warehouse (eg: Google Bigquery) and being used for a dashboard

Prediction Result

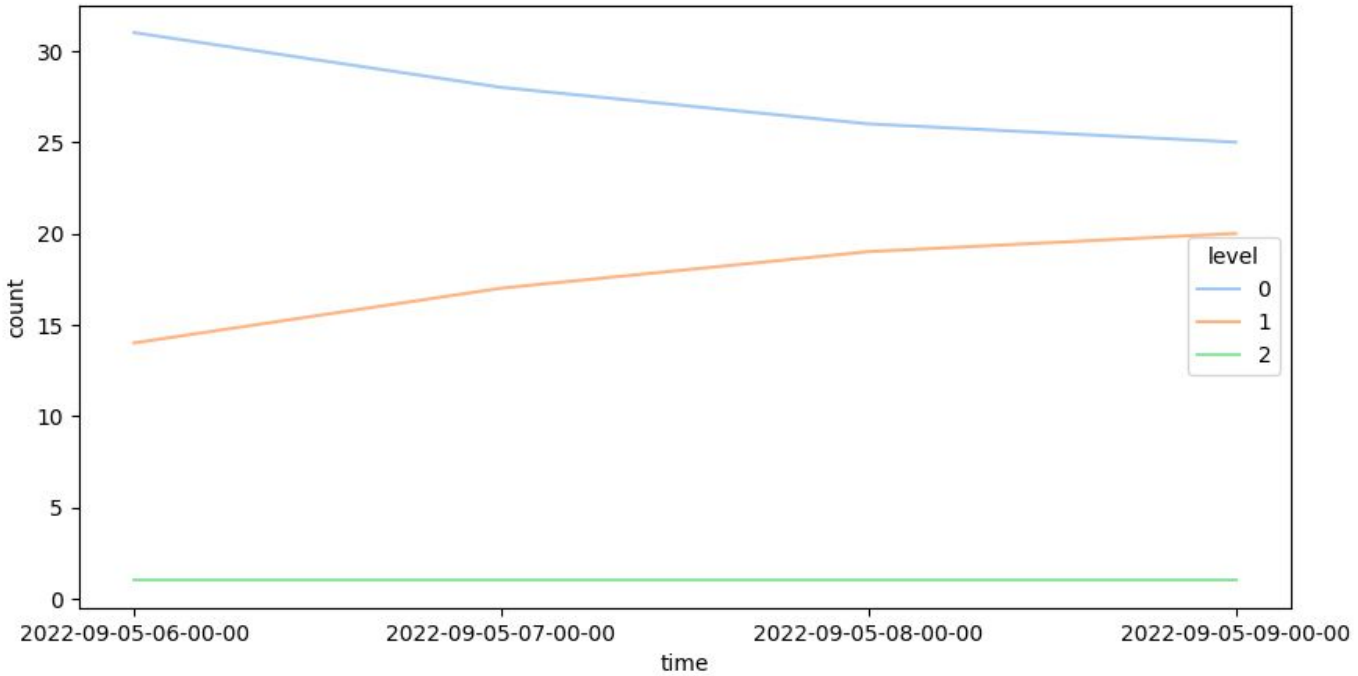
	street	time	median_length	median_delay	median_speed_kmh	median_regular_speed	median_delay_seconds	rain_intensity	holiday_gap	level
0	Binamarga	2022-09-05 06:00:00	944.073021	47.688814	21.158976	22.070456	11.258950	0.0	-1	0
1	N9 KH Soleh Iskandar	2022-09-05 06:00:00	1366.923769	90.454060	34.580495	23.939861	76.192464	0.0	-1	0
2	Pemuda	2022-09-05 06:00:00	1571.778701	30.429110	29.435582	32.308384	10.589722	0.0	-1	0
3	Pangeran Asogiri	2022-09-05 06:00:00	1119.573224	42.858462	17.888612	19.007083	13.017546	0.0	-1	0
4	N9 Ir Haji Juanda	2022-09-05 06:00:00	1830.376447	69.814455	30.579794	27.521334	66.597883	0.0	-1	0
...	...	...	...	...	...	...	...	...	...	...
179	Jenderal Ahmad Yani	2022-09-05 09:00:00	1112.839832	82.561392	20.396286	21.083128	70.323468	0.0	-1	1
180	N9 Jalan Raya Pajajaran	2022-09-05 09:00:00	1335.104995	104.065934	20.905643	22.647561	88.628959	0.0	-1	1
181	Siliwangi	2022-09-05 09:00:00	1370.802413	60.524155	22.886403	26.387502	13.854813	0.0	-1	0
182	RE Abdullah	2022-09-05 09:00:00	843.471092	61.329310	14.726772	22.268389	10.890829	0.0	-1	1
183	N9 Otto Iskandardinata	2022-09-05 09:00:00	790.544255	121.491647	26.927066	31.134088	8.380060	0.0	-1	1

Prediction Result Table

Several Graphs that Could be Made

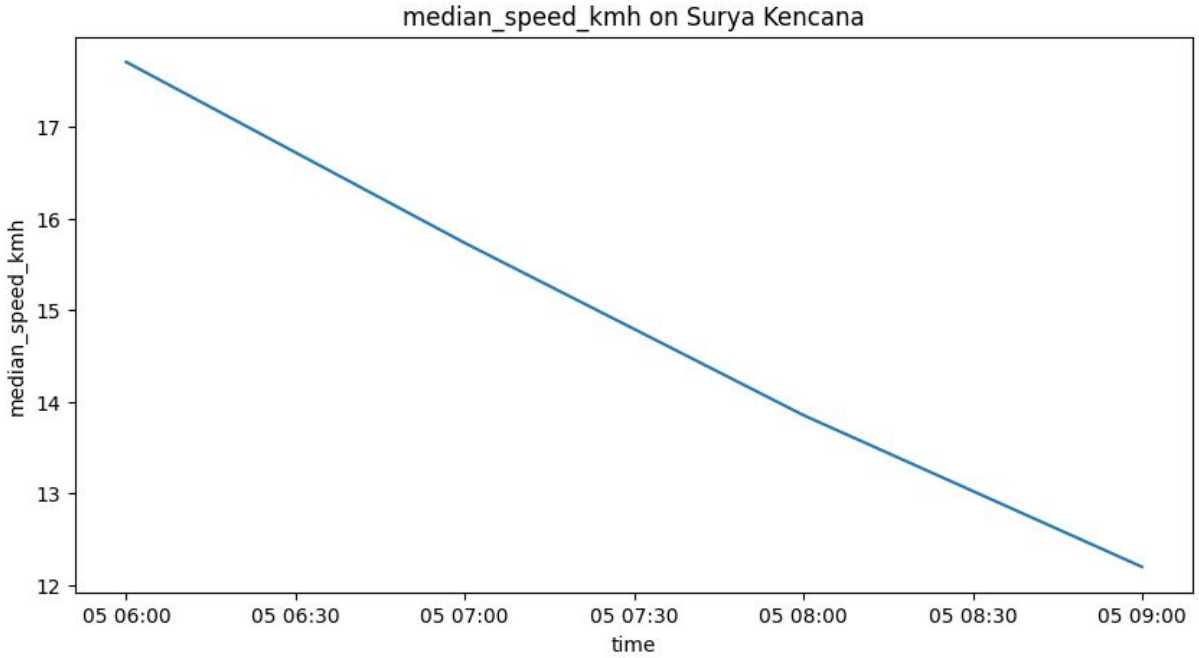


Several Graphs that Could be Made



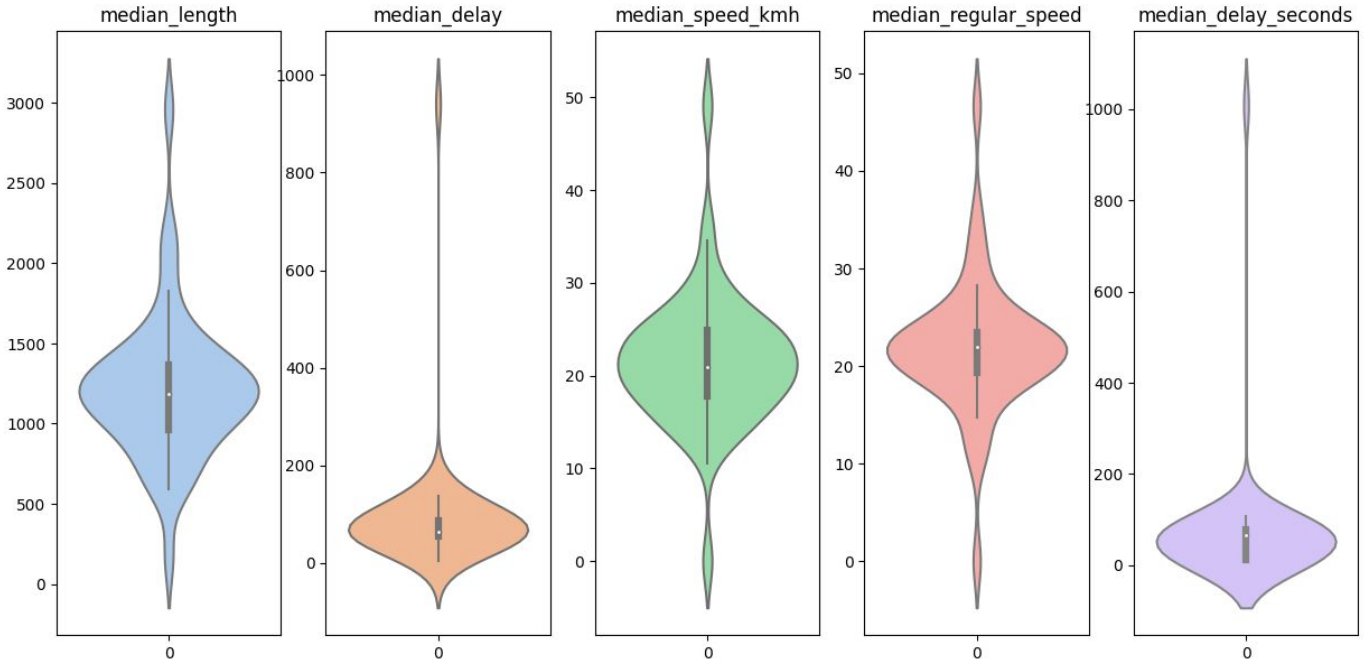
Number of street on each traffic jam level on each timestamp

Several Graphs that Could be Made



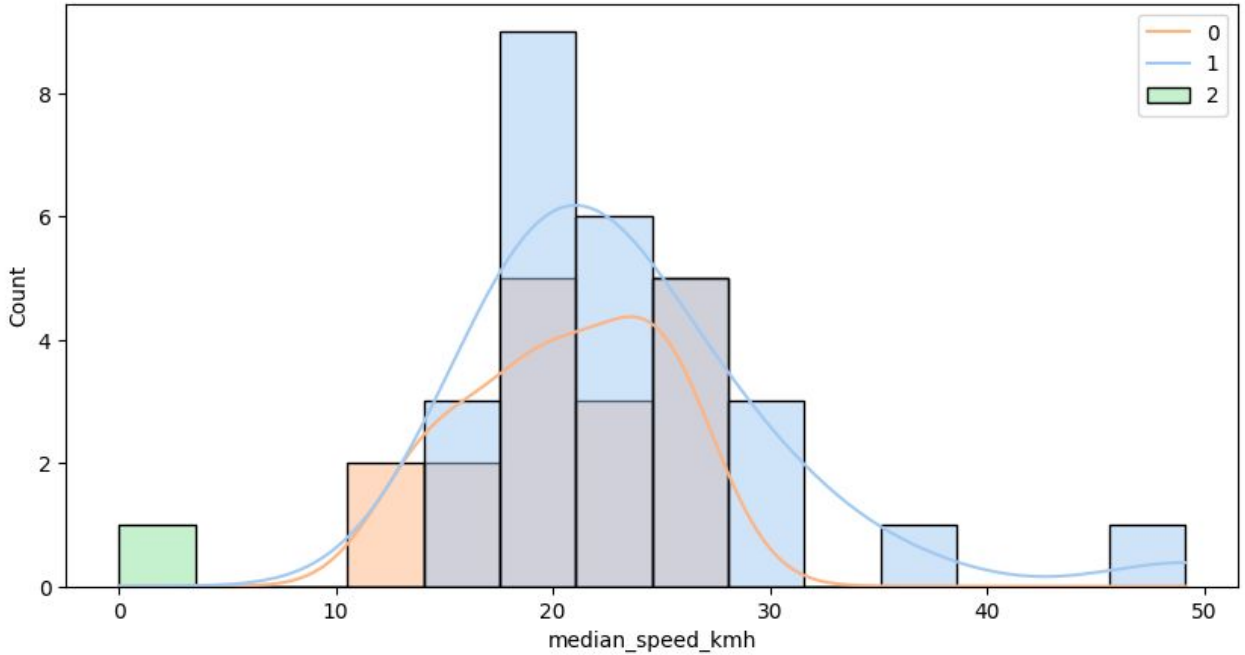
Median Speed on Surya Kencana at each timestamp

Several Graphs that Could be Made



Time Series Feature Violin Plot on a Timespan

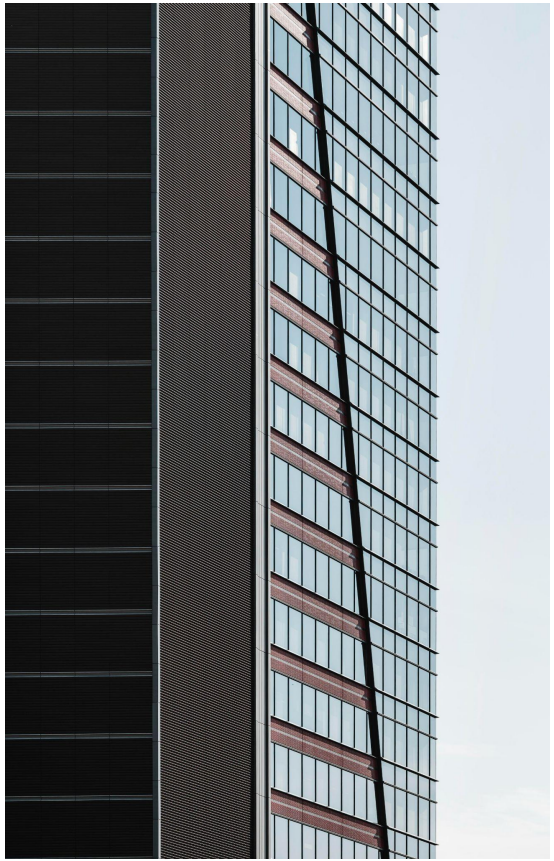
Several Graphs that Could be Made



Median Speed Histogram on each Traffic Jam Level on a timestamp

Future Works

- Add error analysis for Classification model
- Time Series modeling using SARIMA model and see if it’s improve the current model performance



THANK YOU!

Gibran Brahmanta P.

gibranbrahmanta@gmail.com

Reference

- Dauletbak, D., & Woo, J. (2020). Big Data Analysis and prediction of traffic in Los Angeles. KSII Transactions on Internet and Information Systems, 14(2). https://doi.org/10.3837/tiis.2020.02.021