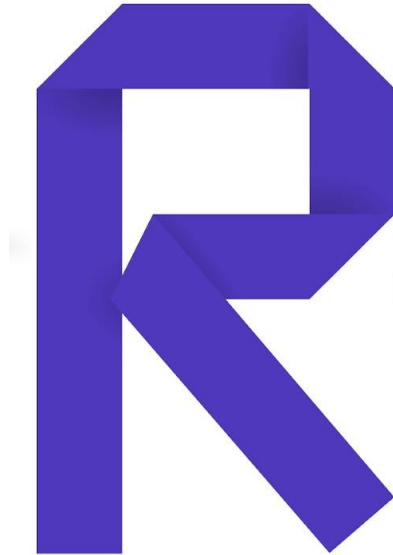


**FORECASTING KECEPATAN KENDARAAN PADA WAKTU TERTENTU
MENGUNAKAN MACHINE LEARNING RANDOM FOREST REGRESSOR DAN
XGBOOST**



**Fauzan Ihza Fajar
Mohamad Faza Fauzan
Roissyah Fernanda Khoiroh**

**FafaAnalytics
RISTEK Datathon 2023**

DAFTAR ISI

I. PENDAHULUAN.....	3
1.1 Latar Belakang.....	3
1.2 Rumusan Masalah.....	3
1.3 Tujuan Penelitian.....	3
II. DATA PREPROCESSING.....	4
2.1 Scraping Data.....	4
2.2 Penggabungan Data.....	4
2.3 Penanganan missing value.....	4
III. EXPLORATORY DATA ANALYSIS.....	5
3.1 Variabel waktu setempat.....	6
3.2 Target Rerata Kecepatan.....	7
3.3 Variabel name, maxspeed, dan highway.....	8
3.4 Analisis Multivariat.....	9
IV. FEATURE ENGINEERING.....	10
IV.1 Ekstrak Fitur Tunggal.....	10
IV.2 Fitur Jarak.....	10
IV.3 Analisis Multivariat.....	11
IV.4 Feature Selection.....	11
V. MODELING.....	12
5.1 Pemodelan Machine Learning.....	12
5.2 Hyperparameter Tuning.....	12
VI. EVALUASI.....	13
VII. ANALISIS HASIL PREDIKSI.....	14
VIII. KESIMPULAN.....	14
IX. REFERENSI.....	15

I. PENDAHULUAN

1.1 Latar Belakang

Forecasting kecepatan rata-rata kendaraan merupakan penggunaan bidang ilmu data dan *machine learning* untuk memprediksi kecepatan rata-rata kendaraan pada waktu tertentu dengan berbagai variabel. Dengan pemanfaatan *machine learning* kita bisa memprediksi secara akurat jalan mana yang macet atau lancar dan mengukur rata-rata kecepatan kendaraan pada waktu tertentu. Adapun model algoritma *machine learning* yang digunakan pada kasus *forecasting* ini, yakni random forest dan XGBoost (*Extreme Gradient Boosting*).

Random forest merupakan jenis algoritma *machine learning* yang masuk dalam kategori *supervised learning*. Random forest dapat diterapkan pada tugas klasifikasi maupun regresi. Penerapannya bisa menangani kumpulan data yang berisi variabel kategoris pada tugas klasifikasi sekaligus variabel kontinu pada tugas regresi. Sebab, algoritma ini menerapkan teknik ansambel. Teknik ansambel sendiri berarti menggabungkan banyak penggolong (*classifiers*) untuk bisa memberikan solusi terhadap masalah yang kompleks.

Extreme Gradient Boosting (XGBoost) merupakan suatu metode pada *machine learning* dimana XGBoost merupakan algoritma regresi dan klasifikasi dengan metode *ensemble*. XGBoost ini juga merupakan suatu varian dari algoritma *Tree Gradient Boosting* yang dikembangkan dengan optimasi 10 kali lebih cepat dibandingkan *Gradient Boosting* (Chen & Guestrin, 2016).

Pada kompetisi ini, kami ingin memprediksi rata-rata kecepatan kendaraan di berbagai jalan pada waktu tertentu menggunakan data historis dengan mengukur faktor-faktor seperti dimensi waktu, jenis serta kondisi jalan, peraturan kecepatan maksimal, dan lain-lain.

1.2 Rumusan Masalah

1. Bagaimana performa model *random forest* dalam memprediksi kecepatan rata-rata kendaraan pada waktu tertentu?
2. Bagaimana performa model *Extreme Gradient Boosting* (XGBoost) dalam memprediksi kecepatan rata-rata kendaraan pada waktu tertentu?

1.3 Tujuan Penelitian

1. Untuk mengukur performa model *random forest* dalam memprediksi kecepatan rata-rata kendaraan pada waktu tertentu.
2. Untuk mengukur performa model *Extreme Gradient Boosting* (XGBoost) dalam memprediksi kecepatan rata-rata kendaraan pada waktu tertentu.

II. DATA PREPROCESSING

2.1 Scraping Data

Scraping data merupakan aktivitas pengambilan informasi atau data dari sebuah sumber daring, seperti halaman web, secara otomatis. Dalam konteks ini, langkah-langkah yang kami lakukan dalam proses scraping melibatkan tiga tugas pokok, yaitu mengumpulkan informasi yang sangat penting untuk tiga kolom di data train dan test, yakni `id_jalan`, `id_titik_mulai`, dan `id_titik_akhir`. Namun dari ketiga kolom tersebut, kami hanya mengambil nilai unique agar mengefisiensi waktu dan tenaga. Informasi untuk kolom `id_jalan` akan kami ambil melalui tautan <https://www.openstreetmap.org/?way=>, sementara untuk kolom `id_titik_mulai` dan `id_titik_akhir` akan kami peroleh melalui tautan <https://www.openstreetmap.org/?node=>. Dari proses scraping ini, hasil yang kami peroleh akan terperinci sebagai berikut:

1. kolom `id_jalan`

Data yang dihasilkan terdapat 20 baris dan 33 kolom ('Unnamed: 0', 'highway', 'lit', 'maxspeed', 'name', 'not:name', 'ref', 'lanes', 'lanes:backward', 'lanes:forward', 'operator', 'source:name', 'source:ref', 'bicycle', 'foot', 'lane_markings', 'sidewalk', 'cycleway:left', 'cycleway:right', 'cycleway', 'surface', 'ref:GB:usrn', 'sidewalk:both', 'source:maxspeed', 'wikidata', 'maxspeed:type', 'pavement:minimum_width', 'source', 'class:bicycle:commute', 'horse', 'oneway', 'busway', 'turn:lanes:forward')

2. kolom `id_titik_mulai` dan `id_titik_akhir`

Data yang dihasilkan terdapat 488 baris dan 3 kolom ('Unnamed: 0', 'Latitude_awal', 'Longitude_awal')

2.2 Penggabungan Data

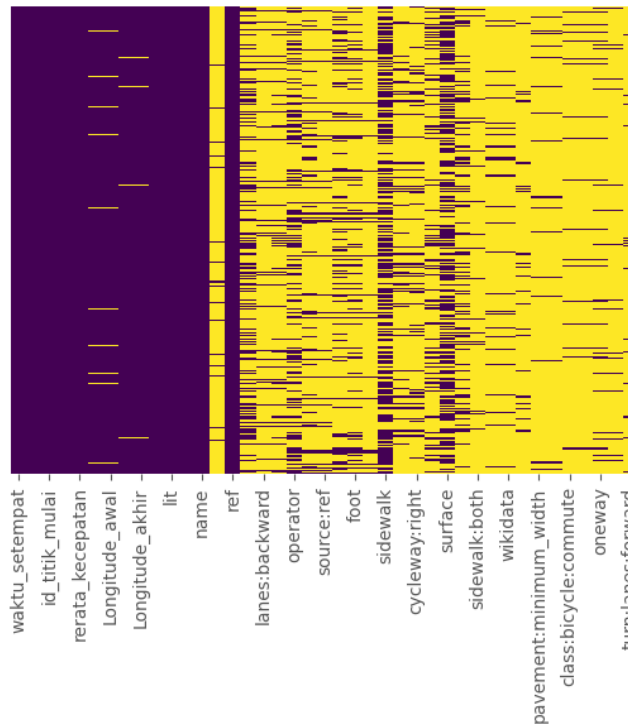
Pada tahap *feature engineering*, kami akan memanfaatkan informasi yang terdapat dalam data train dan data test, khususnya pada kolom `id_jalan`, `id_titik_mulai`, dan `id_titik_akhir`. Oleh karena itu, perlu dilakukan penggabungan antara data train dan data test utama dari ketiga kolom tersebut. Tujuan dari penggabungan ini adalah untuk memastikan bahwa data yang dihasilkan dapat digunakan secara efektif dalam proses pemodelan yang akan dilakukan nantinya.

2.3 Penanganan missing value

Berdasarkan Gambar II.1, dapat diamati bahwasannya sebagian besar kolom pada dataset hasil scraping didominasi oleh missing value seperti pada kolom 'class:bicycle:commute', 'horse', 'not:name', 'lanes', 'lanes:backward', 'lanes:forward', 'operator', 'source:name', 'source:ref', 'bicycle', 'foot', 'lane_markings', 'cycleway:left', 'cycleway:right', 'cycleway', 'ref:GB:usrn', 'sidewalk:both', 'source:maxspeed', 'wikidata', 'maxspeed:type', 'pavement:minimum_width', 'source', 'oneway', 'busway', 'turn:lanes:forward'. Kolom 'Latitude_awal', 'Longitude_awal', 'Latitude_akhir', 'Longitude_akhir' juga memiliki missing value, namun tidak sebanyak missing value

pada kolom-kolom di sebelah kanan. Penanganan missing value akan dibagi pada 3 tahap:

1. Kolom yang memiliki missing value lebih dari 50% akan didrop dari dataset, diantaranya: 'not:name', 'lanes', 'lanes:backward', 'lanes:forward', 'operator', 'source:name', 'source:ref', 'bicycle', 'foot', 'lane_markings', 'cycleway:left', 'cycleway:right', 'cycleway', 'ref:GB:usrn', 'sidewalk:both', 'source:maxspeed', 'wikidata', 'maxspeed:type', 'pavement:minimum_width', 'source', 'class:bicycle:commute', 'horse', 'oneway', 'busway', 'turn:lanes:forward', 'id'.
2. Kolom 'Latitude_awal', 'Longitude_awal', 'Latitude_akhir', 'Longitude_akhir' akan diisi dengan nilai median.
3. Kolom 'sidewalk' dan 'surface' yang memiliki missing value akan diinput dengan 'tidak_tersedia'.

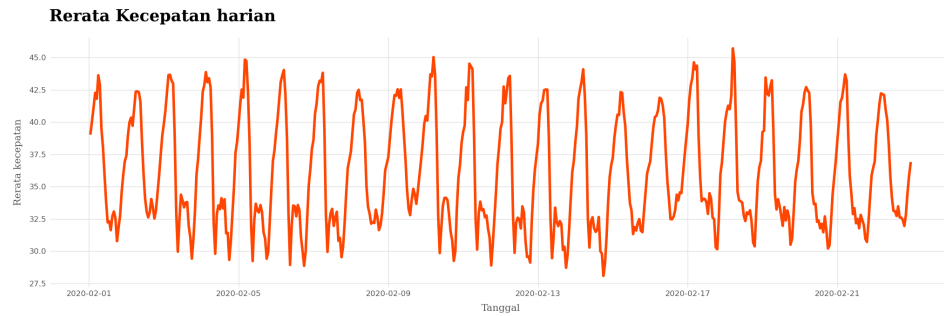


Gambar II.1 heatmap missing value pada data hasil scraping

III. EXPLORATORY DATA ANALYSIS

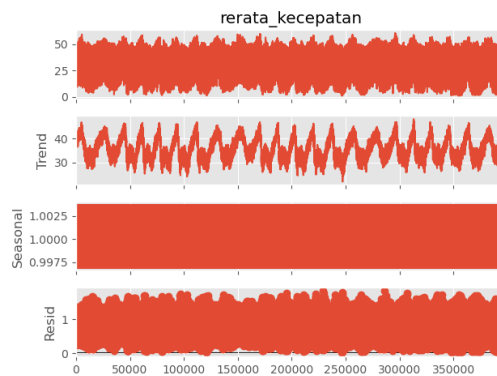
Fitur-fitur pada dataset yang telah dibersihkan, selanjutnya dianalisis untuk mendapatkan temuan. Kami fokus menggali distribusi dari fitur serta target, serta hubungan antara keduanya. Temuan yang diperoleh akan menjadi pertimbangan dalam tahapan selanjutnya, yaitu tahapan *feature engineering*.

3.1 Variabel waktu setempat



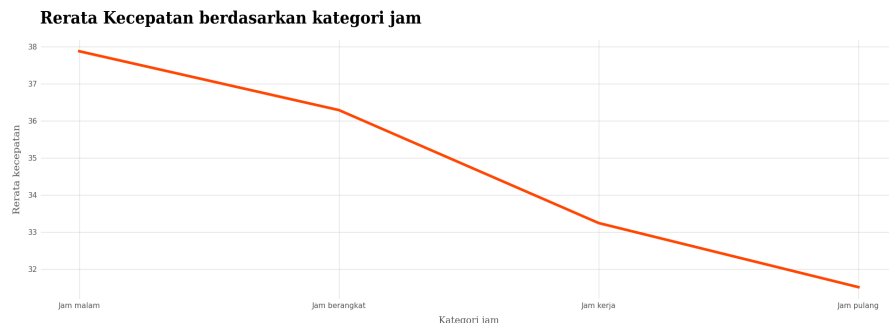
Gambar III.1 plot time series rerata kecepatan

Di atas merupakan plot *time series* dari rerata kecepatan dalam satuan hari. Dalam rentang waktu 1 Februari 2020 - 29 Februari 2020, rerata kecepatan mengalami tren naik lalu turun, yang terjadi secara berulang.



Gambar III.2 plot dekomposisi musiman untuk rerata kecepatan

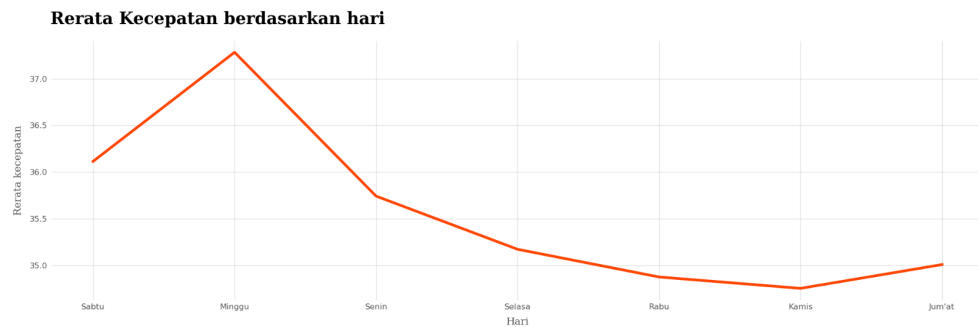
Untuk mengetahui periode musiman dari *series* tersebut, dapat menggunakan metode dekomposisi. Berdasarkan Gambar III.2, dengan menggunakan periode=30, target rerata kecepatan ketika dipecah memiliki pola musiman harian. Ketika ditelisik lebih dalam lagi, khususnya saat variabel waktu_setempat dipecah dari format hari, ke dalam format jam diperoleh temuan terdapat tren rerata kecepatan pada kategori waktu dan kategori hari tertentu.



Gambar III.3 plot rerata kecepatan berdasarkan kategori jam

Gambar III.3 di atas, diperoleh dengan mengelompokkan jam ke dalam 4 kategori yaitu jam berangkat (pukul 7-10 pagi), jam kerja (pukul 10 pagi-4 sore), jam

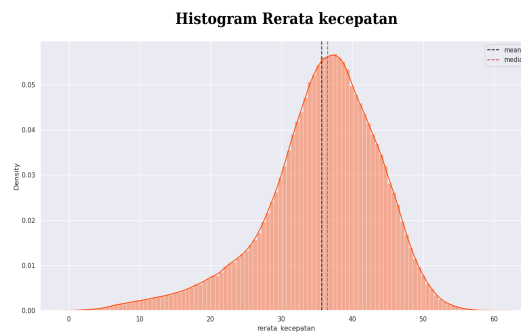
pulang (pukul 4-7 sore), serta jam malam (pukul 7 sore hingga sebelum jam berangkat). Terlihat bahwasannya kendaraan cenderung bergerak lambat saat jam pulang kerja. Lalu, kendaraan juga memiliki perbedaan rerata kecepatan yang cukup signifikan, antara *weekend* dan *weekday* sebagaimana yang terlampir pada Gambar III.4. Rerata kecepatan yang cukup tinggi pada hari minggu dan sabtu mengindikasikan lalu lintas yang lancar dikarenakan penurunan aktivitas, sedangkan pada senin - jum'at rerata kecepatan cenderung lambat akibat hari aktif kerja.



Gambar III.4 plot rerata kecepatan berdasarkan hari

3.2 Target Rerata Kecepatan

Rerata kecepatan sebagai target pada data tersebut, perlu diperiksa distribusinya. Pada Gambar III.5 rerata kecepatan memiliki ekor yang lebih panjang ke kiri daripada ke kanan, lalu mean juga lebih besar daripada median. Hal tersebut mengindikasikan, rerata kecepatan menceng ke kiri.



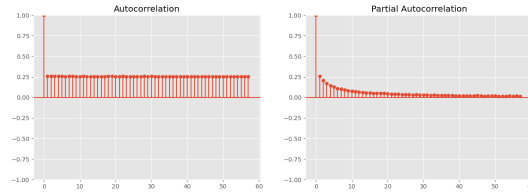
Gambar III.5 histogram rerata kecepatan

Rerata kecepatan sebagai variabel yang akan diprediksi perlu diuji stasioneritasnya. Hal ini dimaksudkan untuk mencegah terjadinya *spurious regression* yaitu fenomena ketika regresi memiliki R^2 yang tinggi, namun sebetulnya hubungan tersebut tidak berarti. Pemeriksaan stasioneritas dapat dilakukan melalui Uji *Augmented Dickey-Fuller Test* (ADF Test), dengan hipotesis sebagai berikut:

- H_0 : data tidak stasioner
- H_1 : data stasioner

Digunakan $\alpha=0.05$, dengan kriteria penolakan uji akan menolak H_0 bila p-value lebih kecil atau sama dengan alpha. Diperoleh p-value sebesar 0.041, artinya H_0 ditolak sehingga rerata kecepatan sudah stasioner. Untuk pembuktian lebih lanjut,

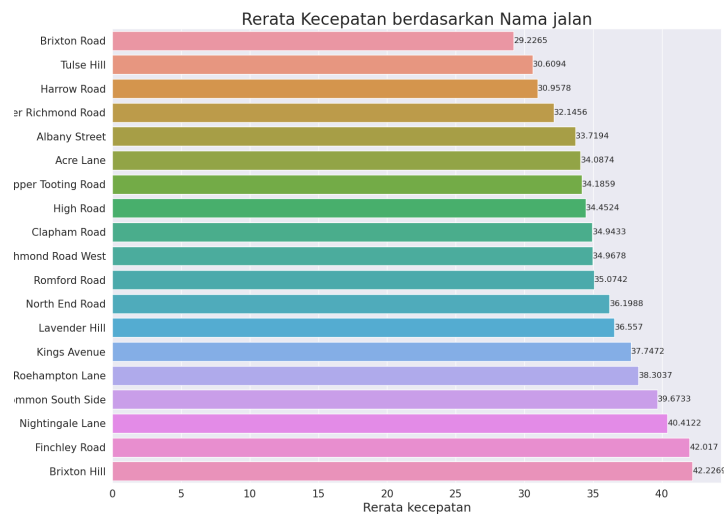
stasioneritas juga dapat diperiksa melalui plot ACF dan PACF. Pada Gambar III.6, melalui plot ACF dan PACF tersebut dapat ditelusuri bahwasannya *series* sudah berada pada nilai mean yang konstan (garis horizontal) sehingga rerata kecepatan sudah stasioner dalam mean.



Gambar III.6 plot ACF dan PACF

3.3 Variabel name, maxspeed, dan highway

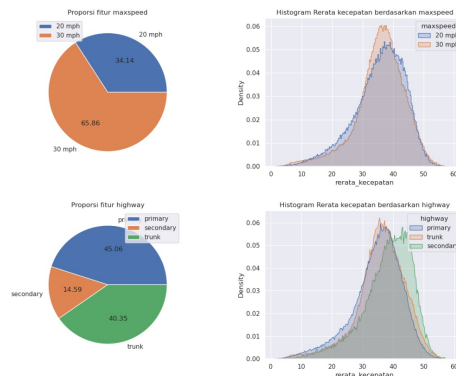
Rerata kecepatan selain berkaitan erat dengan dimensi waktu, juga berhubungan dengan jarak. Jarak mengukur perpindahan kendaraan dari titik awal menuju titik akhir, jalur yang dilalui dalam hal ini nama jalan, serta kondisi jalan perlu dieksplor lebih lanjut. Gambar III.7 menunjukkan hasil perhitungan mean dari target rerata kendaraan berdasarkan nama jalan. Terdapat 19 nama jalan, dengan Brixton Road, Tulse hill, dan Harrow Road menjadi 3 jalan dengan rerata kecepatan terendah.



Gambar III.7 bar chart rerata kecepatan berdasarkan nama jalan

Karakteristik yang dimiliki oleh suatu jalan seperti *maxspeed*, dan *highway* juga dapat diperiksa proporsi nilai uniknya serta keterkaitannya dengan target rerata kecepatan. Gambar III.8 menunjukkan proporsi masing-masing nilai unik untuk variabel *maxspeed*, dan *highway*. Diperoleh temuan, 65.86% jalan memiliki kecepatan maksimal 30 mph, lalu sebanyak 45.06% jalan merupakan *primary highway*. Terkait hubungan antara *highway* dan *maxspeed* dengan rerata kecepatan, pada histogram di bawah dapat diinterpretasikan bahwa jalan dengan *maxspeed* 30mph tentu memiliki

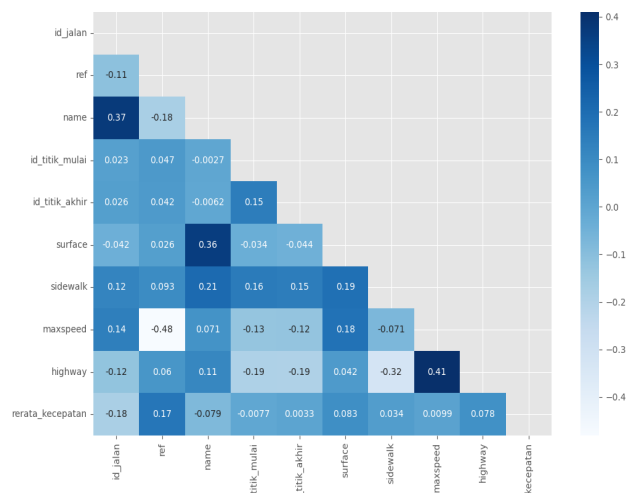
rerata kecepatan lebih tinggi daripada jalan dengan *maxspeed* 20mph, lalu *primary* dan *trunk highway* memiliki rerata kecepatan lebih tinggi daripada *secondary highway*.



Gambar III.8 pie chart dan histogram pada fitur maxspeed, dan highway

3.4 Analisis Multivariat

Kekuatan serta arah hubungan antara banyak variabel dapat diukur dengan menghitung koefisien korelasi *spearman*. Untuk lebih memudahkan interpretasi, koefisien yang dihasilkan disajikan dalam bentuk matriks korelasi. Pada data, sebagian besar fitur merupakan fitur kategorik sehingga perlu melakukan Labelencoder() terlebih dahulu, sebelum menghitung korelasinya. Gambar III.9 berikut menyajikan keterkaitan antara fitur-fitur dalam data. Dapat diamati, sebagian besar *cell* memiliki koefisien korelasi kurang dari 0.1, hanya 3 *cell* dengan koefisien korelasi mencapai angka 0.3-0.41 yaitu 'id_jalan' dengan 'name', 'name' dengan 'surface', dan 'maxspeed' dengan 'highway'. Dapat disimpulkan, hubungan antara fitur-fitur, maupun hubungan antara fitur dengan target dalam data tersebut tergolong lemah.



Gambar III.9 matriks korelasi pada data hasil scraping

IV. FEATURE ENGINEERING

IV.1 Ekstrak Fitur Tunggal

Waktu setempat diekstrak dari format hari menjadi format jam. Lalu, dilakukan rekayasa fitur untuk mengelompokkan hari menjadi fitur baru bernama 'weekend', dimana hari senin-jum'at diencode sebagai 0, sedangkan hari sabtu dan minggu diencode sebagai 1. Rekayasa fitur selanjutnya, dilakukan untuk mengelompokkan jam menjadi fitur baru bernama 'kategori_jam' dengan rincian: jam berangkat (pukul 7 - 10 pagi) diencode dengan 1, jam kerja (pukul 10 pagi - 4 sore) di encode dengan 2, jam pulang (pukul 4 - 7 sore) diencode dengan 3, serta jam malam (pukul 7 sore hingga sebelum jam berangkat kerja) diencode dengan 4.

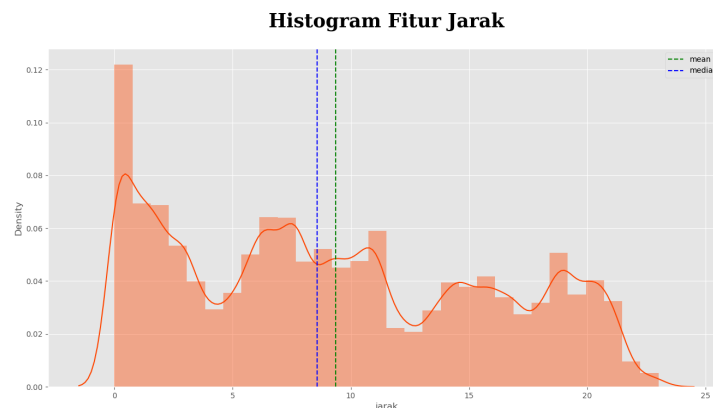
IV.2 Fitur Jarak

Fitur jarak dihasilkan dengan menghitung *Haversine Distance*, menggunakan rumus pada Gambar IV.1 berikut. 'Latitude_awal', 'Longitude_awal', 'Latitude_akhir', 'Longitude_akhir' menjadi fitur yang dibutuhkan untuk menghitung jarak antara titik awal dan titik akhir. Untuk mendapatkan jarak dalam satuan *km*, nilai d yang diperoleh harus dikalikan dengan radius bumi dalam kilometer ($R=6371$).

$$d = 2r \arcsin \left(\sqrt{\sin^2 \left(\frac{\phi_2 - \phi_1}{2} \right) + \cos(\phi_1) \cos(\phi_2) \sin^2 \left(\frac{\lambda_2 - \lambda_1}{2} \right)} \right)$$

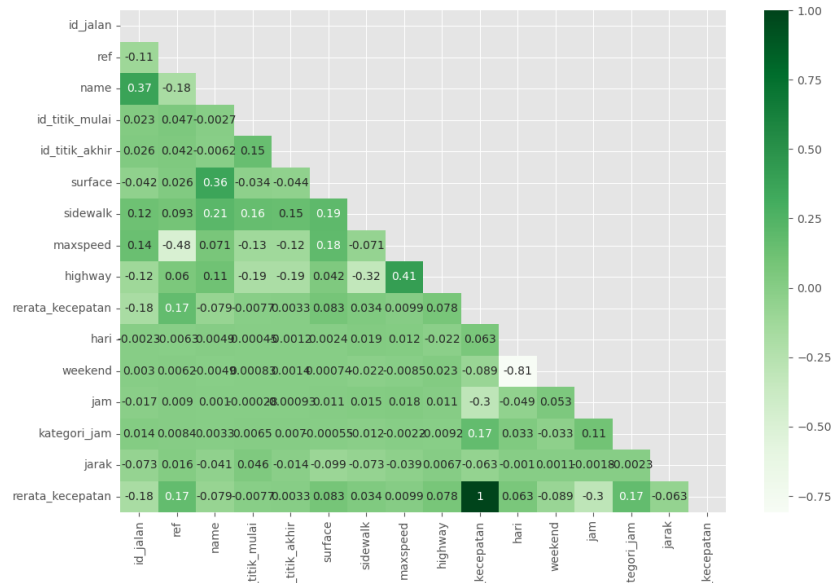
Gambar IV.1 rumus menghitung Haversine Distance

Gambar IV.2 menunjukkan distribusi dari fitur jarak. Terlihat bahwasannya jarak menyebar mulai dari 0 km hingga 23 km, jarak 0 km menjadi jarak yang paling sering ditempuh. Distribusi dari fitur jarak ini terlihat tidak beraturan namun masih dapat diamati bahwa distribusinya menceng ke kanan, yang dibuktikan dengan nilai median lebih besar daripada nilai mean.



Gambar IV.2 histogram fitur jarak

IV.3 Analisis Multivariat



Gambar IV.3 matriks korelasi setelah penambahan fitur baru

Matriks Korelasi di atas diperoleh setelah melakukan rekayasa fitur, khususnya terhadap fitur tunggal waktu setempat dan fitur yang berguna untuk menghitung jarak yaitu Latitude awal, Latitude akhir, Longitude awal, dan Longitude akhir. Gambar IV.3 di atas menunjukkan seberapa kuat hubungan antara banyak fitur dalam data. Fitur weekend dan fitur hari memiliki koefisien korelasi sebesar 0.81, dengan tanda negatif. Fitur weekend merupakan fitur baru yang dihasilkan dengan mengelompokkan hari ke dalam 2 kategori yaitu *weekday* dan *weekend*, sehingga tidak heran bila koefisien korelasi antara hari dan weekend cukup besar. Untuk fitur-fitur lainnya memiliki koefisien korelasi yang kecil yaitu kurang dari 0.18.

IV.4 Feature Selection

Berdasarkan hasil analisa pada poin sebelumnya, fitur-fitur yang akan dipakai untuk model diantaranya: 'id_jalan', 'id_titik_mulai', 'id_titik_akhir', 'hari', 'jam', 'kategori_jam', dan 'weekend'. Alasan pemilihan tersebut, karena 'id_jalan' sudah mewakili 'name' dan 'ref', yang merupakan nama jalan. 'id_titik_mulai', dan 'id_titik_akhir' sudah mewakili 'jarak'. Lalu untuk kolom-kolom 'hari', 'jam', 'kategori_jam', dan 'weekend' merepresentasikan waktu. Meskipun terdapat unsur multikolinearitas pada fitur 'weekend' dan 'hari', hal tersebut tidak menjadi masalah karena model yang kami gunakan merupakan model non-linear.

V. MODELING

5.1 Pemodelan Machine Learning

Pemodelan dilakukan dengan melatih 2 model gradient boosted machine learning yaitu Extreme Gradient Boosting (XGBoost) dan Random Forest Regressor. Untuk setiap model, dilakukan hyperparameter tuning guna meminimumkan nilai

SMAPE (Symmetric Mean Absolute Percentage Error). Masing masing dari model akan diinput `random_state=42`, agar model dapat dipanggil kembali dan terhindar dari *randomize*.

5.2 Hyperparameter Tuning

Hyperparameter tuning adalah proses mencari kombinasi nilai yang optimal untuk hyperparameter dari sebuah model machine learning atau algoritma. Pada penelitian ini, metode yang digunakan adalah Bayesian Optimization, yaitu metode untuk mencari kombinasi optimal dari hyperparameter suatu model dengan memanfaatkan pendekatan probabilistik. Tujuannya adalah meminimalkan atau memaksimalkan fungsi objektif yang biasanya mewakili kinerja model.

Berikut adalah konfigurasi hyperparameter untuk tiap model

Tabel V.1 konfigurasi hyperparameter Random Forest

No.	Parameter	Range Nilai Parameter
1.	n_estimators	(10,200)
2.	max_depth	(5,15)
3.	min_samples_split	(2,10)
4.	min_samples_leaf	(1,10)
5.	max_features	(0.1,1.0)
6.	min_impurity_decrease	(0.0,1.0)

Tabel V.2 konfigurasi hyperparameter XGBoost

No.	Parameter	Range Nilai Parameter
1.	n_estimators	(10,200)
2.	learning_rate	(0.01, 1.0)
3.	max_depth	(1, 20)
4.	min_child_weight	(0.1, 10)
5.	subsample	(0.1, 1.0)
6.	colsample_bytree	(0.1, 1.0)

Tahap selanjutnya, dilakukan 30 iterasi untuk melakukan hyperparameter tuning, sehingga didapat nilai SMAPE terbaik dengan konfigurasi hyperparameter sebagai berikut.

Tabel V.3 hasil hyperparameter tuning Random Forest

No.	Parameter	Nilai Parameter
1.	n_estimators	200
2.	max_depth	20
3.	min_samples_split	10
4.	min_samples_leaf	2
5.	max_features	1
6.	min_impurity_decrease	0

Tabel V.3 hasil hyperparameter tuning XGBoost

No.	Parameter	Nilai Parameter
1.	n_estimators	87
2.	learning_rate	0.4041252174051859
3.	max_depth	11
4.	min_child_weight	7.798752694870117
5.	subsample	0.67629599233451
6.	colsample_bytree	0.6094043421124198

VI. EVALUASI

Setelah dilakukan hyperparameter tuning pada seluruh data train, dilakukan evaluasi model menggunakan rata-rata dari K-fold Cross Validation dengan n_splits=5 dan didapatkan hasil sebagai berikut

Tabel VI.1 hasil SMAPE

No.	Model	CV SMAPE (Symmetric Mean Absolute Percentage Error)
1.	Random Forest Regressor	9.429
2.	Random Forest Regressor Tuning	8.612

3.	XGBoost	9.097
4.	XGBoost Tuning	8.445

VII. ANALISIS HASIL PREDIKSI

Tabel VI.2 hasil Train Test Split (75:25) SMAPE

No.	Model	Train Test Split (75:25) SMAPE	
		Train	Test
1.	Random Forest Regressor	6.42	9.42
2.	Random Forest Regressor Tuning	7.61	8.62
3.	XGBoost	9.03	9.14
4.	XGBoost Tuning	7.84	8.49

Tabel diatas menunjukkan hasil evaluasi antara train set dan test set menggunakan SMAPE. XGBoost menjadi model dengan SMAPE terbesar di antara ketiga model lainnya, namun selisih SMAPE antara kedua set tersebut tergolong kecil hanya selisih 0.09 yang mengindikasikan model mampu belajar dengan baik dalam memprediksi rerata kecepatan. Untuk Random Forest Regressor, meskipun memiliki SMAPE terkecil namun selisih antara SMAPE pada train set dan test set menjadi yang paling besar di antara ketiga model, dengan selisih sebesar 3. Selanjutnya, selisih SMAPE pada train set dan test set pada Random Forest Regressor Tuning yaitu sebesar 1.01, sedangkan pada XGBoost Tuning selisihnya sebesar 0.65. Keempat model tersebut mengalami *overfitting* namun masih dalam batas wajar karena selisih antara SMAPE di train set, dan test set relatif kecil.

VIII. KESIMPULAN

Pada penelitian ini model random forest regressor digunakan karena kemampuannya yang cocok dalam menangani data non linear, seperti waktu, hari dalam seminggu, dan variabel target (kecepatan kendaraan). Sedangkan XGBoost dalam penelitian ini digunakan karena memiliki teknik boosting yang berfokus pada mengurangi kesalahan model regresi.

Berdasarkan hasil pembuatan model machine learning yang telah dilakukan menggunakan variabel hari/jam, id jalan, id titik awal dan id titik akhir. Didapatkan XGBoost Tuning sebagai model dengan performa terbaik dalam memprediksi rerata

kecepatan dikarenakan memiliki SMAPE terkecil yaitu 8.445 saat evaluasi model menggunakan rata-rata dari K-fold Cross Validation.

IX. REFERENSI

- [1] Bhuiyan, M., Kabir, A., & Ahasan Kabir, M. (2020). Medical Image and data Processing Using Machine Learning algorithm View project Vehicle Speed Prediction based on Road Status using Machine Learning. *Advanced Research in Energy and Engineering*, 2(1), 1–9. www.fazpublishing.com/aree
- [2] Chada, S. K., Görges, D., Ebert, A., & Teutsch, R. (2022). *Deep Learning-Based Vehicle Speed Prediction for Ecological Adaptive Cruise Control in Urban and Highway Scenarios*. <http://arxiv.org/abs/2212.00149>
- [3] Gaikwad, T. D. (2020). *Vehicle Velocity Prediction Using Artificial Neural Networks and Vehicle Velocity Prediction Using Artificial Neural Networks and Effect of Real-World Signals on Prediction Window Effect of Real-World Signals on Prediction Window*. https://scholarworks.wmich.edu/masters_theses.
- [4] Rombe, Yoris. (2021). PENGGUNAAN METODE XGBOOST UNTUK KLASIFIKASI STATUS OBESITAS DI INDONESIA. *Journal Unhas*. <http://repository.unhas.ac.id/id/eprint/13027/>
- [5] Yasir, Rafed Muhammad, dkk,. (2022). *Traffic Congestion Prediction Using Machine Learning Techniques*. *Cornell University Journal*. <https://arxiv.org/abs/2206.10983>.
- [6] Zhou, Z., Yang, Z., Zhang, Y., Huang, Y., Chen, H., & Yu, Z. (2022). *iScience A comprehensive study of speed prediction in transportation system: From vehicle to traffic*. <https://doi.org/10.1016/j.isci>