ANALISIS PREDIKSI CURAH HUJAN BERDASARKAN DATA CUACA HARIAN DAN DATA CUACA PER JAM

ARKAVIDIA 8.0

DATAVIDIA 2023

Tim Kotak Riset SC

Disusun Oleh:

Audrey Shafira Fattima	**(21/472678/PA/20320)**
Rabbani Nur Kumoro	**(21/472599/PA/20310)**
William Hilmy Susatyo	**(21/472585/PA/20308)**

Universitas Gadjah Mada

Daerah Istimewa Yogyakarta

2023

1. LATAR BELAKANG

Cuaca merupakan faktor yang tidak dapat dikendalikan oleh manusia, namun memiliki pengaruh yang signifikan terhadap kehidupan sehari-hari. Secara khusus, curah hujan merupakan parameter meteorologi yang paling signifikan terhadap kehidupan manusia karena dapat mempengaruhi banyak aspek, mulai dari infrastruktur bangunan, kemungkinan bencana alam seperti banjir, hingga transportasi seperti jadwal penerbangan pesawat dan sebagainya. Dengan demikian, prediksi cuaca sangat penting untuk diketahui sebagai bahan pertimbangan dalam mengambil keputusan dan merencanakan aktivitas. Prediksi curah hujan dapat dilakukan dengan mengimplementasikan metode *machine learning* data historis yang tersedia.

2. TUJUAN DAN MANFAAT

Secara umum, tujuan dilakukannya laporan ini adalah memprediksi curah hujan (mm) yang akan terjadi di suatu wilayah dengan menggunakan data cuaca harian dan data cuaca per jam. Adapun manfaat dari laporan ini adalah untuk membantu perencanaan kegiatan yang berkaitan dengan cuaca, seperti kegiatan pertanian, pariwisata, dan transportasi dan mengantisipasi bencana alam yang disebabkan oleh cuaca, seperti banjir atau kekeringan.

3. BATASAN MASALAH

Batasan yang terdapat pada laporan ini diantaranya adalah sebagai berikut:

1. *Dataset* yang digunakan hanya berdasarkan *dataset* yang diberikan oleh panitia Arkavidia,
2. Fitur yang akan diprediksi adalah fitur “rain_sum (mm)” yang merupakan curah hujan pada hari tertentu.

Penentuan *Mean Squared Error* (MSE) berdasarkan *train test split* yang dilakukan terhadap data *train*, dimana dilakukan pembagian sebesar 80% untuk *train*, dan 20% untuk *test*.

4. METODE

a. PREPROCESSING

Dataset yang digunakan terdiri dari empat *dataset* yang berbeda. *Dataset* pertama adalah *train dataset* yang berisi fitur-fitur dan target yang digunakan untuk melatih model. Selain itu, diberikan juga *train_hourly*, yang merupakan dataset tambahan yang berisi fitur-fitur untuk setiap jam. Lalu, selain *train dataset,* diberikan juga *test dataset,* dan *test_hourly dataset*, yang memiliki fitur yang sama seperti *train dataset.* Namun, pada *test dataset*, tidak terdapat fitur yang menjadi target prediksi, yaitu kolom *rain_sum,* yang berisi jumlah curah hujan pada hari tersebut.

Berdasarkan *dataset* yang diberikan, masih terdapat banyak *missing values* di beberapa kolom dan masih banyak distribusi data yang *skewed.* Apabila tidak dibersihkan, hal-hal tersebut akan mengacaukan perhitungan yang dilakukan oleh *machine learning model*, sehingga perlu dilakukan *preprocessing* terhadap *dataset* yang diberikan. Oleh karena itu, tahapan ini akan dibagi menjadi dua langkah, yakni proses *handle* terhadap *missing values* dan distribusi yang *skewed*.

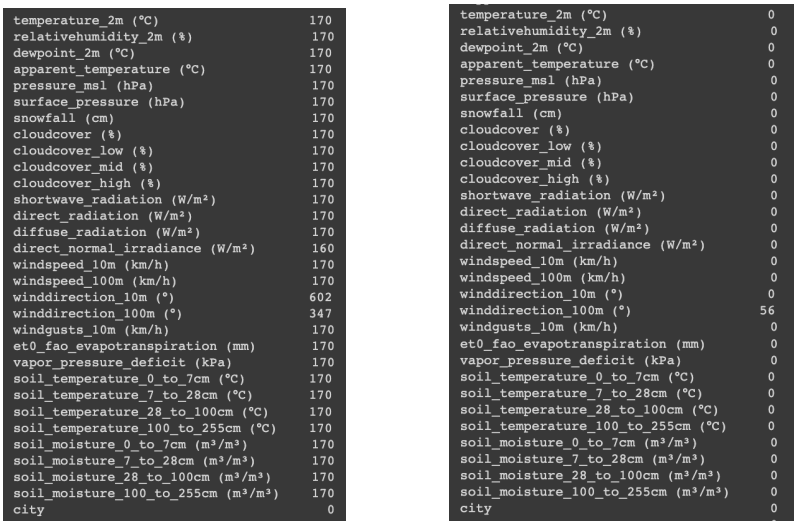
1. Handle Missing Values

Dapat dilihat pada ***Gambar 1.*** dan ***Gambar 2.***, bahwa terdapat beberapa kolom yang memiliki *missing values* pada *dataset train* dan *train_hourly.* Namun, tidak terdapat *missing values* pada *dataset test* dan hanya terdapat *missing values* pada 1 kolom untuk *dataset test_hourly*

temperature_2m_max (°C)	50
temperature_2m_min (°C)	50
apparent_temperature_max (°C)	50
apparent_temperature_min (°C)	50
shortwave_radiation_sum (MJ/m²)	60
rain_sum (mm)	60
snowfall_sum (cm)	60
windspeed_10m_max (km/h)	50
windgusts_10m_max (km/h)	50
winddirection_10m_dominant (°)	466
et0_fao_evapotranspiration (mm)	60
elevation	0
city	0

temperature_2m_max (°C)	0
temperature_2m_min (°C)	0
apparent_temperature_max (°C)	0
apparent_temperature_min (°C)	0
shortwave_radiation_sum (MJ/m²)	0
snowfall_sum (cm)	0
windspeed_10m_max (km/h)	0
windgusts_10m_max (km/h)	0
winddirection_10m_dominant (°)	0
et0_fao_evapotranspiration (mm)	0
elevation	0
city	0

Gambar 1. *Missing Values pada dataset train dan test*

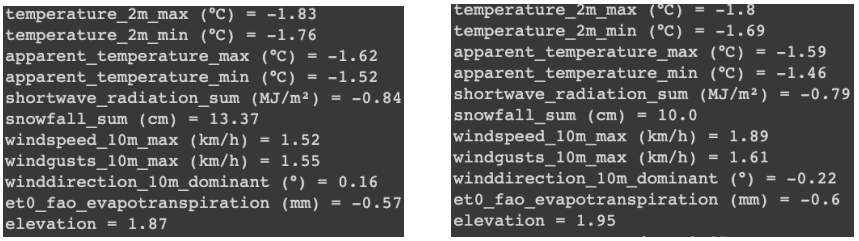


Gambar 2. *Missing Values pada dataset train_hourly dan test_hourly*

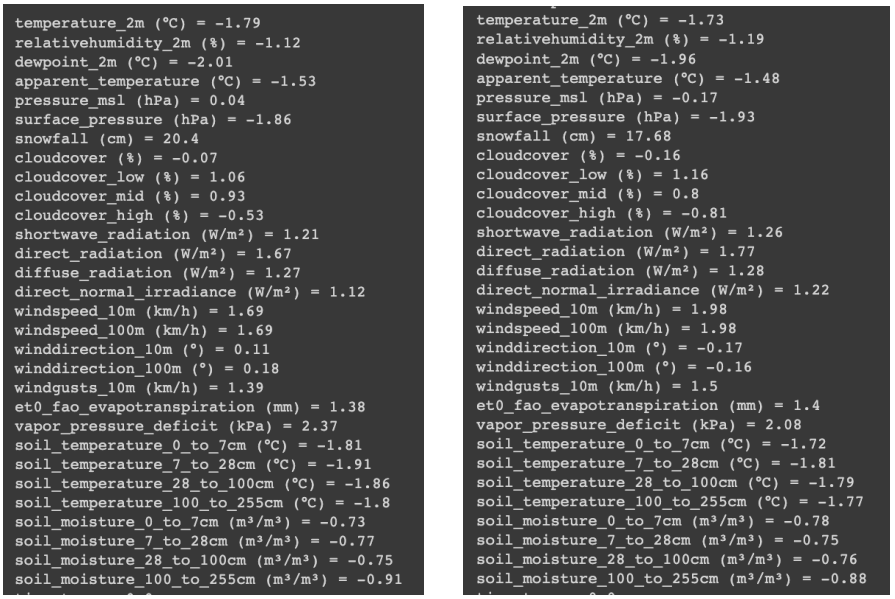
Secara keseluruhan, persentase *missing value* terhadap keseluruhan data pada *dataset train* adalah sebesar 0.42% dan *train_hourly* sebesar 0.032%. Sehingga, dapat disimpulkan bahwa persentase *missing value* tersebut tidak signifikan terhadap keseluruhan data. Oleh karena itu, langkah yang dilakukan untuk mengatasi *missing values* tersebut adalah dengan menghapus baris mengandung *missing values* pada salah satu kolomnya.

2. Handle Skew

Pada penelitian ini, *threshold* (batasan) yang digunakan untuk menentukan apakah suatu kolom *skew* kanan adalah sebesar 1. Sementara itu, untuk *skew* kiri batasannya adalah sebesar -1. Sehingga, seperti yang terlihat pada ***Gambar 3.*** dan ***Gambar 4.***, dapat dikatakan bahwa distribusi dari data yang memiliki *skew* yang lebih besar atau lebih kecil dari batas-batas tersebut cenderung tidak merata.



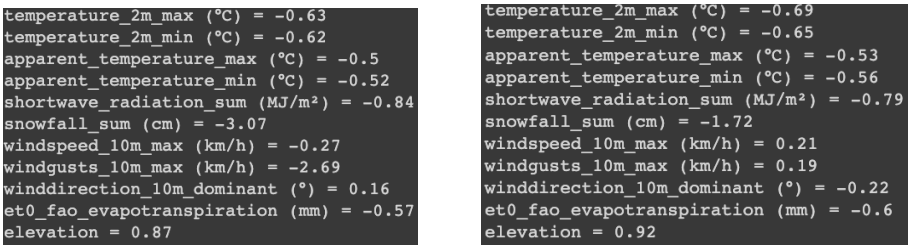
Gambar 3. *Skew pada dataset train dan test (sebelum transformasi)*



Gambar 4. *Skew pada dataset train_hourly dan test_hourly (sebelum transformasi)*

Untuk kolom yang distribusi datanya masih cenderung *skewed* ke kanan atau ke kiri, diimplementasikan transformasi untuk mengurangi *skew* dari kolom tersebut. Distribusi data yang *skew* ke kiri dapat dinormalisasi dengan baik oleh transformasi kuadrat (Square Transform) sedangkan pada data yang *skew* ke kanan dapat dinormalisasi dengan baik oleh transformasi logaritma (Log Transform) **[3]**. Oleh karena itu, data memiliki kecenderungan untuk *skew* ke kiri, maka dapat diaplikasikan transformasi pangkat kuadrat. Sedangkan, jika distribusi data cenderung *skew* ke kanan, dapat diaplikasikan transformasi logaritma.

Seperti yang dapat dilihat pada ***Gambar 5.*** dan ***Gambar 6.***, setelah dilakukan transformasi, *skew* dari setiap kolom yang *skew* awalnya lebih besar dari 1 mengalami penurunan, sedangkan pada kolom yang *skew* awalnya lebih kecil dari -1 cenderung mengalami peningkatan.



Gambar 5. *Skew pada dataset train dan test (setelah transformasi)*

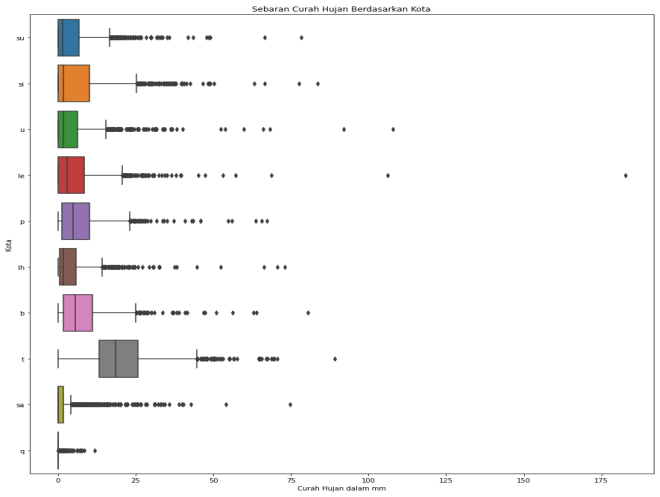
```
temperature_2m (°C) = -0.48
relativehumidity_2m (%) = -0.8
dewpoint_2m (°C) = -0.53
apparent_temperature (°C) = -0.36
pressure_msl (hPa) = -0.17
surface_pressure (hPa) = -1.84
snowfall (cm) = -6.2
cloudcover (%) = -0.16
cloudcover_low (%) = -0.49
cloudcover_mid (%) = 0.8
cloudcover_high (%) = -0.81
shortwave_radiation (W/m²) = 0.19
direct_radiation (W/m²) = 0.46
diffuse_radiation (W/m²) = 0.17
direct_normal_irradiance (W/m²) = 0.3
windspeed_10m (km/h) = -0.37
windspeed_100m (km/h) = -0.48
winddirection_10m (°) = -0.17
winddirection_100m (°) = -0.16
windgusts_10m (km/h) = -0.08
et0_fao_evapotranspiration (mm) = -0.81
vapor_pressure_deficit (kPa) = -0.57
soil_temperature_0_to_7cm (°C) = -0.14
soil_temperature_7_to_28cm (°C) = -0.55
soil_temperature_28_to_100cm (°C) = -0.6
soil_temperature_100_to_255cm (°C) = -0.6
soil_moisture_0_to_7cm (m³/m³) = -0.78
soil_moisture_7_to_28cm (m³/m³) = -0.75
soil_moisture_28_to_100cm (m³/m³) = -0.76
soil_moisture_100_to_255cm (m³/m³) = -0.88
```

```
temperature_2m (°C) = -0.43
relativehumidity_2m (%) = -0.74
dewpoint_2m (°C) = -0.17
apparent_temperature (°C) = -0.33
pressure_msl (hPa) = 0.04
surface_pressure (hPa) = -1.77
snowfall (cm) = -6.27
cloudcover (%) = -0.07
cloudcover_low (%) = -0.49
cloudcover_mid (%) = 0.93
cloudcover_high (%) = -0.53
shortwave_radiation (W/m²) = 0.16
direct_radiation (W/m²) = 0.4
diffuse_radiation (W/m²) = 0.14
direct_normal_irradiance (W/m²) = 0.25
windspeed_10m (km/h) = -0.43
windspeed_100m (km/h) = -0.52
winddirection_10m (°) = 0.11
winddirection_100m (°) = 0.18
windgusts_10m (km/h) = -0.17
et0_fao_evapotranspiration (mm) = -0.78
vapor_pressure_deficit (kPa) = -0.52
soil_temperature_0_to_7cm (°C) = 0.17
soil_temperature_7_to_28cm (°C) = -0.47
soil_temperature_28_to_100cm (°C) = -0.53
soil_temperature_100_to_255cm (°C) = -0.54
soil_moisture_0_to_7cm (m³/m³) = -0.73
soil_moisture_7_to_28cm (m³/m³) = -0.77
soil_moisture_28_to_100cm (m³/m³) = -0.75
soil_moisture_100_to_255cm (m³/m³) = -0.91
```

Gambar 6. *Skew pada dataset train_hourly dan test_hourly (setelah transformasi)*

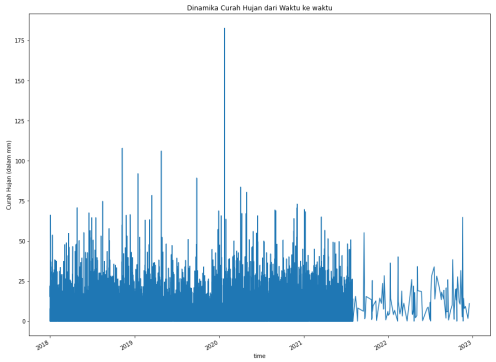
b. EXPLORATORY DATA ANALYSIS

Pada tahapan ini, dilakukan eksplorasi terhadap data untuk menemukan pola, hubungan, dan informasi di antara setiap fitur yang ada yang belum diketahui sebelumnya. Seperti yang bisa dilihat pada ***Gambar 7.***, langkah pertama yang dilakukan adalah memvisualisasikan *box plot* berdasarkan intensitas hujan di setiap kota. Berdasarkan visualisasi yang dilakukan, dapat dilihat nilai minimum, maksimum, rata-rata, serta kuartil dari data intensitas hujan di setiap kota.



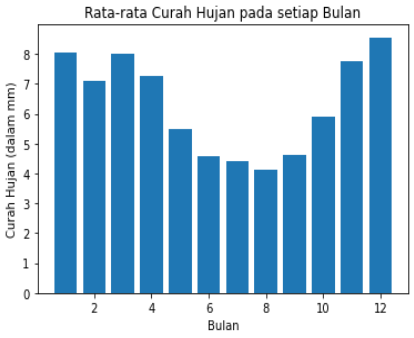
Gambar 7. *Box-Plot Intensitas Curah Hujan Berdasarkan Kota*

Selanjutnya, seperti yang ditunjukkan pada ***Gambar 8.***, dilakukan visualisasi analisis runtun waktu pada curah hujan dari tahun ke tahun. Pada visualisasi ini, dapat dilihat bahwa mulai dari pertengahan tahun 2021 sampai seterusnya, data yang tersedia tidak selengkap tahun-tahun sebelumnya. Artinya terdapat ketidakseimbangan pada data karena banyaknya *missing values.*



Gambar 8. *Line-Plot Analisis Runtun Waktu pada Curah Hujan setiap Tahun*

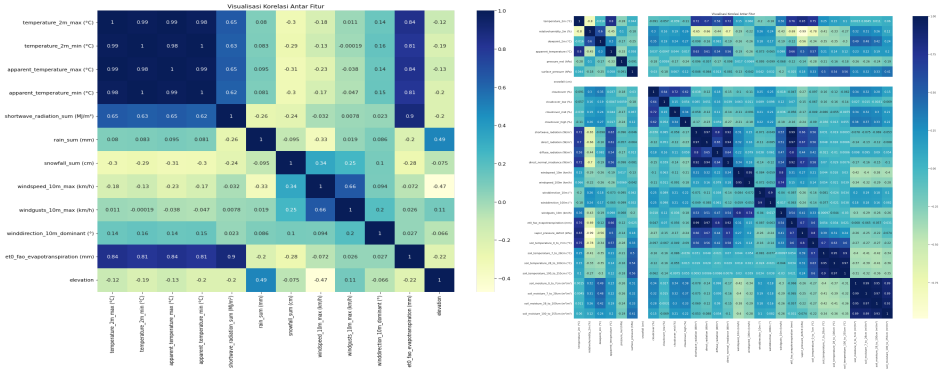
Setelah melakukan visualisasi berdasarkan runtun waktu, dilakukan visualisasi untuk rata-rata curah hujan pada setiap bulannya. Dapat dilihat pada ***Gambar 9***., digunakan rata-rata curah hujan per bulan pada setiap tahun. Dari visualisasi ini dapat disimpulkan bahwa setiap bulan memiliki curah hujan yang berbeda-beda, sehingga dapat disimpulkan bahwa periode waktu mempengaruhi curah hujan.



Gambar 9. *Bar-Plot Rata-rata Curah Hujan setiap Bulan*

Untuk mengevaluasi korelasi antar fitur pada *dataset* yang diberikan, dianjurkan untuk menggunakan *heatmap* sebagai metode visualisasi data. Dari analisis *heatmap* pada ***Gambar 10.***, dapat dilihat bahwa masih terdapat banyak titik yang memiliki warna biru tua, yang menunjukkan bahwa masih terdapat

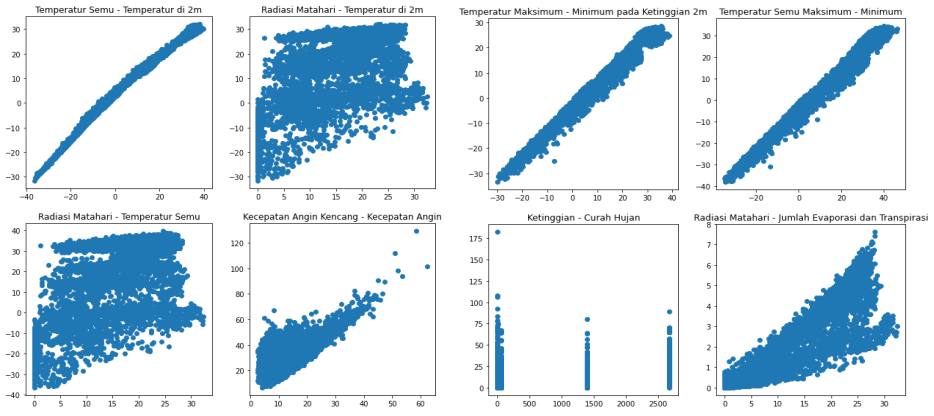
korelasi yang tinggi antar fitur. Sedangkan, pada *heatmap* **(b)**, dapat dilihat bahwa terdapat sel yang berwarna putih, yang menunjukkan bahwa fitur tersebut memiliki nilai korelasi 0 pada semua barisnya.

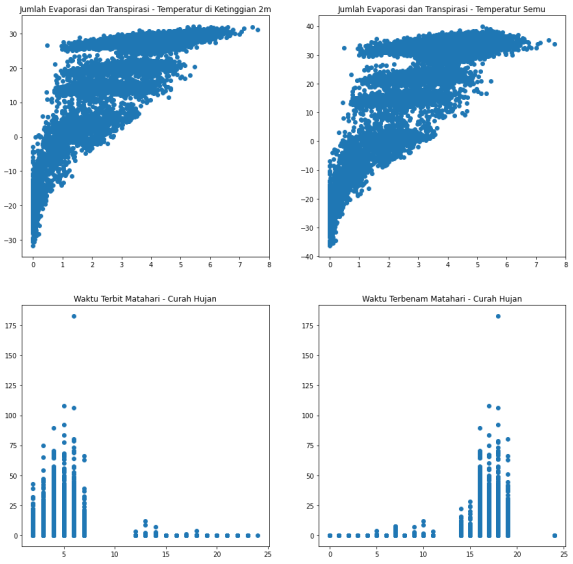


(a) **(b)**

Gambar 10. *Heatmap Korelasi Antar Fitur pada (a) train dan (b) train_hourly*

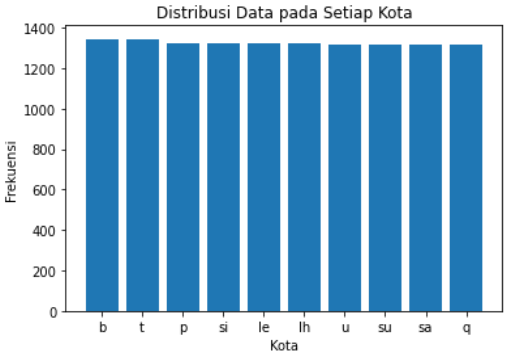
Setelah melakukan analisis menggunakan *heatmap*, selanjutnya adalah membuat visualisasi dalam bentuk *scatter plot. Scatter plot* dan *heatmap* memiliki fungsi yang sama, yaitu untuk mengevaluasi korelasi antar fitur dalam *dataset*. Namun, *heatmap* menampilkan distribusi data dalam bentuk matriks dengan menggunakan warna untuk menunjukkan intensitas data pada setiap sel matriks. Sedangkan *scatter plot* menampilkan hubungan antara dua variabel dengan menggunakan titik-titik yang di *plot* pada koordinat x dan y. Pada visualisasi yang dapat dilihat pada ***Gambar 11.***, dapat dilihat bahwa terdapat beberapa variabel yang memiliki korelasi yang tinggi, sementara variabel lain memiliki tingkat korelasi yang rendah dengan variabel lainnya.





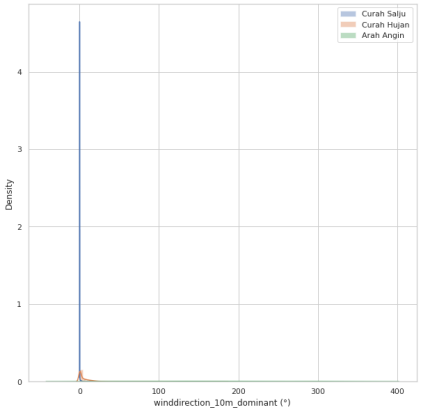
Gambar 11. *Scatter Plot Korelasi Antar Fitur Lanjutan pada train*

Untuk mengevaluasi distribusi data dalam *dataset*, dianjurkan untuk menggunakan metode visualisasi *bar plot*. *Bar plot* dapat digunakan untuk menampilkan perbandingan jumlah data pada setiap kota. Dari analisis visualisasi *bar plot* pada ***Gambar 12.***, dapat dilihat bahwa data terdistribusi dengan merata pada *dataset* ini.



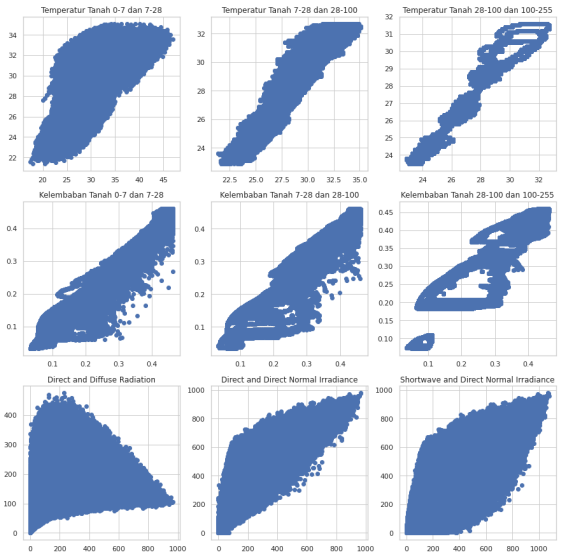
Gambar 12. *Bar Plot Distribusi Data pada Setiap Kota*

Kernel Density Estimate (KDE) *Plot* merupakan salah satu metode yang dapat digunakan untuk mengevaluasi frekuensi kemunculan suatu nilai pada sebuah *dataset*. Metode ini digunakan untuk menggambarkan fungsi densitas probabilitas dari variabel data yang bersifat kontinu atau non-parametrik. Melalui visualisasi yang dihasilkan, dapat dilihat bahwa distribusi data curah salju sangat tidak merata, dengan banyaknya data yang memiliki nilai 0.



Gambar 13. *Distribution Plot pada Fitur dengan Tipe Data Numerik di train*

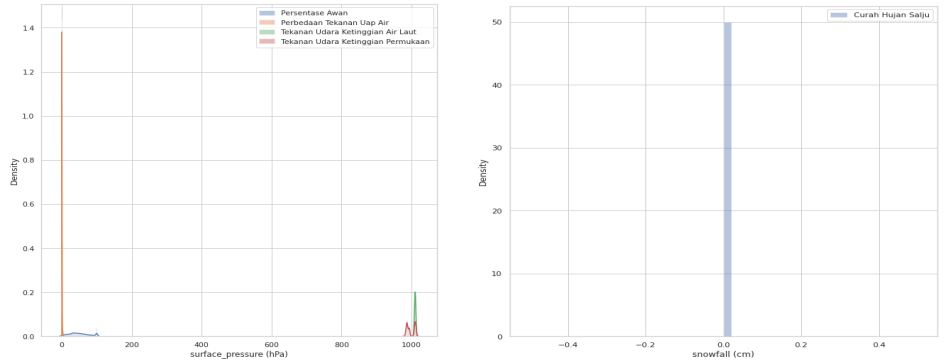
Setelah melakukan analisis terhadap *dataset* train, selanjutnya adalah memvisualisasikan *dataset train_hourly* dengan menggunakan metode *scatter plot*. Seperti yang telah dijelaskan sebelumnya, *scatter plot* merupakan metode visualisasi yang digunakan untuk mengevaluasi korelasi antar variabel dalam *dataset*.



Gambar 14. *Scatter Plot Korelasi Antar Fitur Lanjutan pada train_hourly*

Sama seperti pada *dataset* sebelumnya, visualisasi dilakukan dengan menggunakan metode *Kernel Density Estimate* (KDE) *Plot* untuk mengetahui seberapa sering suatu nilai muncul dalam *dataset*. Dari **Gambar 15.**, dapat diketahui bahwa distribusi data tidak merata pada dua fitur, yaitu perbedaan

tekanan uap air dan curah hujan salju. Analisis ini menunjukkan bahwa kedua fitur tersebut memiliki jumlah data bernilai 0 yang cukup besar.



Gambar 15. *Distribution Plot pada Fitur dengan Tipe Data Numerik di train_hourly*

c. FEATURE ENGINEERING

Untuk mengolah data, setelah melakukan data *preprocessing* dan EDA, langkah selanjutnya yang dilakukan adalah *feature engineering.* Pada tahapan ini, langkah yang dilakukan mencakup modifikasi fitur, melakukan *encoding* terhadap variabel kategorik, dan melakukan *scaling* terhadap fitur yang terdapat pada *dataset.*

1. Modifikasi Fitur pada masing-masing Dataset

Pada tahapan ini dilakukan penambahan kolom baru berdasarkan kombinasi dari beberapa kolom yang terdapat pada *dataset* yang diberikan. Adapun beberapa kolom baru yang ditambahkan mencakup sebagai berikut:

1. ***year***, didapat dengan melakukan pemotongan (*slicing*) dari segmen tahun dalam kolom waktu dari *dataset train* dan *test*.
2. ***month,*** diperoleh dengan mengekstrak segmen bulan dari kolom waktu pada *dataset train* dan *test*.
3. ***sunrise_time,*** diperoleh dengan melakukan pemotongan terhadap segmen jam dan menit dari kolom *sunrise (iso 8601)* pada *dataset train_hourly* dan *test_hourly* dengan ketentuan sebagai berikut:
    - Apabila menit pada suatu jam berada diatas 30, kolom *sunrise time* diisikan dengan waktu 1 jam setelah jam tersebut.

- Sebaliknya, apabila menit pada suatu jam berada dibawah atau sama dengan 30, maka kolom baru akan diisi dengan segmen jam dari kolom terkait.

4. ***sunset_time*** diperoleh dengan melakukan pemotongan terhadap segmen jam dan menit dari kolom *sunset (iso 8601)* pada *dataset train_hourly* dan *test_hourly* dengan ketentuan yang serupa seperti pada kolom *sunrise_time*
5. ***soil_temperature_avg,*** dihasilkan dari rata-rata antara temperatur tanah pada kedalam 0-7 cm, 7-28 cm, 28-100 cm, dan 100-255 cm pada *dataset train_hourly* dan *test_hourly*
6. ***soil_moisture_avg***, mencakup rata-rata dari kelembapan air pada tanah untuk kedalaman 0-7 cm, 7-28 cm, 28-100 cm, dan 100-255 cm pada *dataset train_hourly* dan *test_hourly*

Selanjutnya, untuk mengurangi redundansi fitur, dilakukan penghapusan terhadap kolom yang telah dibuat menjadi fitur baru. Seperti pada kolom *sunrise (iso 8601), sunset (iso 8601)* yang telah digantikan oleh fitur *sunrise_time* dan *sunset_time*. Setelah itu, dilakukan *preprocessing* terhadap fitur yang baru dibuat tersebut dengan langkah sebagaimana yang sudah dijelaskan pada tahapan preprocessing.

Selain itu, berdasarkan hasil dari tahapan EDA, terdapat beberapa fitur yang memiliki korelasi tinggi satu sama lain. Dikarenakan fitur-fitur ini tidak akan memberikan informasi tambahan dan hanya akan meningkatkan tingkat kompleksitas model nantinya, maka dilakukan penghapusan pada beberapa kolom yang memiliki tingkat korelasi yang tinggi. Selain itu, dilakukan juga penghapusan terhadap kolom yang memiliki distribusi yang sangat tidak merata seperti pada kolom curah hujan salju.

Langkah yang dilakukan selanjutnya adalah menggabungkan informasi yang terdapat pada *dataset train* dengan *dataset train_hourly* serta *dataset test* dengan *test_hourly.* Adapun informasi yang digabungkan adalah rata-rata dan standar deviasi untuk setiap hari dan kota pada *dataset train_hourly* dan *test_hourly*. Hal ini disebabkan karena pada dasarnya rata-rata dan standar

deviasi dapat merepresentasikan distribusi data dengan baik.

3. Handling Variabel Kategorik

Berdasarkan *dataset* yang sudah digabungkan pada tahapan sebelumnya, dapat diamati bahwa terdapat beberapa kolom yang distribusinya tersebar ke dalam beberapa nilai saja, seperti *year, month, elevation*, *sunrise time*, dan *sunset time.* Fitur tersebut kemudian diubah ke dalam bentuk kategorik. Selanjutnya, bersama dengan beberapa fitur yang pada kondisi awal sudah bersifat kategorik, seperti *city*, diklasifikasikan apakah fitur tersebut termasuk kategorik nominal atau kategorik ordinal. Didapat bahwa fitur *city* dan *year* merupakan kategorik nominal, sedangkan sisanya merupakan kategorik ordinal.

Untuk fitur kategorik nominal dapat diolah menggunakan metode *one-hot encoding*, sedangkan fitur kategorik ordinal diolah menggunakan *label encoding*. Dalam *one-hot encoding*, setiap kategori dari data diubah menjadi sebuah vektor dengan panjang sama dengan jumlah kategori yang ada, di mana hanya satu elemen di vektor tersebut yang memiliki nilai 1 (hot) dan sisanya 0 (cold) **[2]**. Sementara itu, *label encoding* merupakan teknik yang digunakan untuk mengubah fitur kategori dari sebuah *dataset* menjadi bilangan bulat sehingga fitur tersebut dapat ditangani oleh *machine learning model* **[6]**.

4. Feature Scaling

Berdasarkan langkah yang sudah dilakukan sebelumnya, ditemukan bahwa pada sejumlah fitur masih terdapat pencilan (*outlier*). Oleh karena itu, pada tahapan ini, *Robust Scaler* digunakan untuk menormalisasi data dengan menghilangkan *outlier*. Tahapan ini dilakukan dengan menghitung median dan kuartil dari setiap fitur, kemudian mengurangi median dari setiap nilai dan membagi dengan *interquartile range* (IQR). Hal ini memungkinkan *Robust Scaler* untuk menangani data dengan nilai yang sangat tidak biasa. Berbeda dengan *Standard Scaler* yang menggunakan rata-rata dan standar deviasi, *Robust Scaler* dapat mengatasi *outlier* pada data dengan lebih efektif.

Disamping itu, ditemukan pula bahwa jumlah fitur yang terdapat pada *dataset* yang diolah sangatlah banyak. Untuk mengatasi permasalahan tersebut,

Principal Component Analysis (PCA) diimplementasikan terhadap data yang sudah dinormalisasi untuk mengurangi dimensi dari data dengan menelusuri kombinasi linear dari fitur yang mewakili sebagian besar variansi dari data. PCA mencari komponen utama dari data yang memiliki variansi yang paling besar, yang kemudian digunakan sebagai fitur baru. PCA digunakan untuk mengurangi kompleksitas data dan menemukan pola yang mungkin tidak dapat dilihat dengan mata telanjang.

d. MODELLING

Model terbaik yang digunakan untuk melakukan prediksi curah hujan dalam laporan ini adalah dengan metode *stacking* yang menggabungkan *model* Extra Trees, LGBM, dan CatBoost menggunakan Voting Regressor.

Extra Trees Regressor adalah algoritma *machine learning* yang menggunakan teknik Random Forest untuk menyelesaikan masalah regresi. Secara esensial, algoritma ini adalah varian dari Random Forest yang menggunakan teknik "*extra-trees*" untuk meningkatkan kinerja **[4]**. Perbedaan utama antara Extra Trees Regressor dan Random Forest adalah proses *splitting* yang dilakukan secara acak sehingga lebih efektif dalam mengatasi *overfitting*.

Algoritma *machine learning* selanjutnya yang digunakan adalah Light Gradient Boosting Machine (LGBM) yang berbasis *gradient boosting*. Umumnya, LGBM ideal digunakan untuk dataset yang besar dan kompleks **[5]**. Disamping itu, LGBM juga memiliki fitur untuk menangani data yang tidak seimbang dan *missing value*.

Kemudian, algoritma *machine learning* yang digunakan selanjutnya adalah CatBoost yang juga berbasis *gradient boosting* dan baik digunakan pada dataset yang memiliki *categorical feature*, serta memiliki sejumlah metode yang dapat meningkatkan performa seperti pengambilan kolom yang penting dan *learning rate* yang dapat diatur **[8]**.

Untuk meningkatkan performa dari beberapa *model* tersebut, digunakan Voting Regressor yang menggabungkan ketiga *model* dengan cara memberikan bobot pada masing-masing *model* yang digabungkan dan mengambil rata-rata

dari hasil prediksi dari setiap *model* **[1]**. Dengan cara ini, Voting Regressor dapat mengurangi *overfitting* dan meningkatkan stabilitas dari *model*. Proses implementasi *model* pada *dataset* yang diberikan adalah sebagaimana ditunjukkan pada ***Diagram 1.***

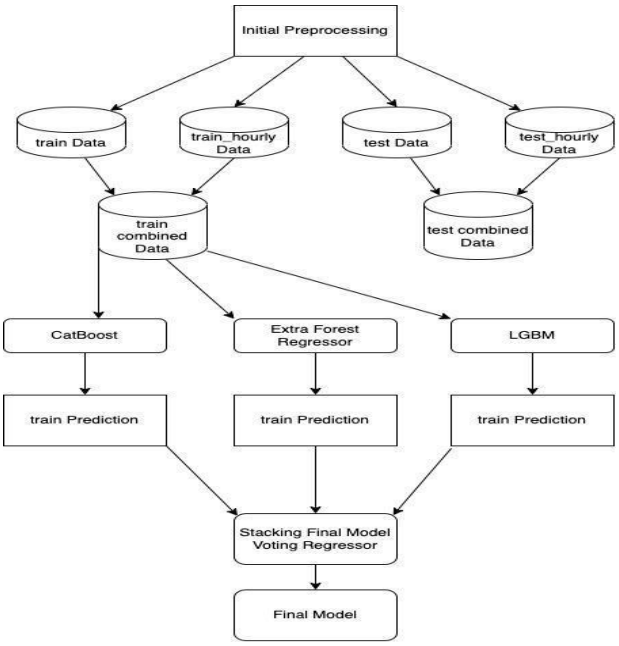


Diagram 1. *Implementasi model pada dataset*

e. VALIDATION

Pemisahan uji-latih, atau yang lebih dikenal dengan istilah *train test split*, merupakan metode yang umum digunakan untuk mengevaluasi kinerja *machine learning model*. Secara umum, metode ini dilakukan dengan membagi data yang tersedia menjadi dua bagian, yaitu data latih yang digunakan untuk melatih model dan data uji yang digunakan untuk mengevaluasi kinerja *model*. Metode *train test split* baik digunakan pada *dataset* yang memiliki jumlah data yang banyak **[7].** Berdasarkan *dataset training* yang diberikan, sebanyak 80% diantaranya akan dijadikan sebagai data latih, sedangkan sisanya digunakan sebagai data uji.

Adapun metrik yang digunakan sebagai acuan dalam laporan ini adalah *mean square error*. *Mean Square Error* (MSE) adalah metrik yang digunakan untuk mengukur kesalahan dalam model prediksi. Rumus yang digunakan untuk menghitung MSE adalah sebagai berikut:

$$MSE = \frac{1}{n}\Sigma(y - \hat{y})^2$$

dengan n menyatakan banyaknya data, y menyatakan nilai aktual yang terdapat pada data, dan $\hat{y}$ menyatakan hasil prediksi. Berdasarkan rumus tersebut, dapat disimpulkan bahwa MSE tidak mungkin bernilai negatif. Nilai MSE yang mendekati 0 mengindikasikan bahwa model yang digunakan adalah model yang dapat melakukan prediksi dengan akurat.

Perbandingan hasil dari implementasi stacking terhadap ketiga model dengan penggunaan model tersebut secara tunggal adalah sebagai berikut:

Model / Score	LGBM	Extra Tree	CatBoost	Stacking
mean square error	23.812	23.914	22.674	22.35

5. KESIMPULAN

Berdasarkan hasil laporan ini, metode *stacking* antara Extra Trees, LGBM, dan CatBoost dapat memprediksi kuantitas curah hujan berdasarkan fitur yang terdapat pada *dataset* yang diberikan dengan MSE sebesar 22.35, sehingga dapat disimpulkan bahwa metode yang digunakan dapat melakukan prediksi kuantitas curah hujan dengan baik. Meskipun demikian, hasil yang didapat masih dapat terus ditingkatkan dengan melakukan penelitian dan pengembangan lebih lanjut terhadap proses analisis data dan pemodelan yang sudah dilakukan. Dengan harapan, metode yang digunakan dalam laporan ini dapat dimanfaatkan oleh pihak terkait untuk merencanakan kegiatan yang berhubungan dengan kondisi cuaca, seperti kegiatan pertanian, pariwisata, dan transportasi serta tindakan preventif terhadap bencana alam yang disebabkan oleh cuaca seperti banjir dan kekeringan.

DAFTAR PUSTAKA

[1] Dietterich, T.G., 2000. *Ensemble Methods in Machine Learning*. LCNS 1857 (pp. 1-15). Available at: https://link.springer.com/chapter/10.1007/3-540-45014-9_1 (Accessed: January

17, 2023)

[2] Dahouda, M.K., Joe, I., 2021. *A Deep Learned Embedding Technique for Categorical Features Encoding*. In IEEE Access (Volume 9) (pp. 114381-114391). IEEE. Available at: https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=9512057 (Accessed: January 17, 2023)

[3] Feng, C., Wang, H., Lu, N., Chen, T., He, H., Lu, Y., Tu, X., 2014. *Log-transformation and its Implications for Data Analysis*. Available at: https://www.ncbi.nlm.nih.gov/pmc/articles/PMC4120293/ (Accessed: January 17, 2023)

[4] Geurts, P., Ernst, D., and Wehenkel, L., 2006. *Extremely Randomized Trees*. *Machine Learning* (2006) 63 (pp. 3-42). Available at: https://link.springer.com/article/10.1007/s10994-006-6226-1 (Accessed: January 18, 2023)

[5] Ke, G., Meng, Q., Finley, T., Wang, T., Chen, W., Ma, W., Ye, Q., and Liu, T.Y., 2017. *LightGBM: A Highly Efficient Gradient Boosting Decision Tree*. 31st Conference on Neural Information Processing Systems (NIPS 2017). Available at:https://proceedings.neurips.cc/paper/2017/file/6449f44a102fde848669bdd9eb6b76fa-Paper.pdf (Accessed: January 17, 2023)

[6] Kelleher, J.A. and Kelleher, B.D., 2018. *A Comparison of Feature Encoding Techniques for Categorical Variable in Decision Tree Models*. IEEE Transaction on Knowledge and Data Engineering (Accessed: January 16, 2023)

[7] Niculescu-Mizil, A., Caruana, C., 2006. *An Empirical Comparison of Supervised Learning Algorithms.* Available at: https://dl.acm.org/doi/abs/10.1145/1143844.1143865 (Accessed: January 17, 2023)

[8] Prokhorenkova, L., Gusev, G., Vorobev, A., Dorogush, A.V., and Gulin, A., 2017. *CatBoost: Unbiased Boosting with Categorical Features*. Moscow Institute of Physics and Technology. Available at: https://arxiv.org/abs/1706.09516 (Accessed: January 18, 2023)

LAMPIRAN

Source Code | Google Colaboratory

1. **Preprocessing**
   - https://colab.research.google.com/drive/1Mve9oNVMmRr-FXds47EU-MURb7DU18kl
2. **Exploratory Data Analysis (EDA)**
   - https://colab.research.google.com/drive/1DWu9nol0Rx90yVtfDQ0Jk7nsZr5ll3ls
3. **Feature Engineering**
   - https://colab.research.google.com/drive/1uZEMAQ7310vRoCt18rCWRtvhRjvNnIlt
4. **Modelling dan Validation**
   - https://colab.research.google.com/drive/1kFk5mpV4Lti6TPNgT3hONCbhBTBt4Ov9