Comparison of CatBoost and LightGBM Models for Air Humidity Prediction

Tangkas Surya Wibawa [1]*, Novita Kurnia Ningrum [2], Ahmad Syahreza [3]***
* Informatics Engineering, Dian Nuswantoro University
tangkassurya2803@gmail.com [1], novita.kn@dsn.dinus.ac.id [2], zamysyah@gmail.com [3]

Article Info

Article history:

Received 2025-04-30
Revised 2025-06-13
Accepted 2025-06-16

Keyword:

Air Humidty,
CatBoost,
LightGBM,
Machine Learning,
Weather Prediction.

ABSTRACT

This study uses historical weather data from the Badan Meteorologi, Klimatologi, dan Geofisika (BMKG) to evaluate the performance of two combination machine learning models, LightGBM and CatBoost, in predicting air humidity. Daily weather data including temperature, humidity, rainfall, daylight duration, and wind characteristics are included in the dataset. Among the preprocessing procedures were label encoding, normalization with MinMaxScaler, and managing missing values. Date fields' temporal information was extracted using feature engineering. Both models were optimized using GridSearchCV with three-fold cross-validation after being trained with an 80/20 split. Using R², MAE, and RMSE, the model's performance has been evaluated. CatBoost outperformed LightGBM, which received an R² score of 0.7981, with a better R² score (0.8191) and smaller prediction errors (MAE = 0.0570, RMSE = 0.0744). While feature importance analysis indicated that temperature and seasonal features were important predictors, residual plots validated the models low bias and good generalization. Both models can help with strategic decision-making in climate-sensitive businesses and salt production, according to the results, and are suitable for humidity forecasting.

I. INTRODUCTION

Indonesia is one of the largest archipelagic countries in the world with a coastline length of 82,290 km (Ministry of Marine Affairs and Fisheries of the Republic of Indonesia, 2018). The potential of Indonesia's coastal resources is very large, both from biological and non-biological resources, one of which is salt[1]. Salt is divided into two types, namely consumption salt and industrial salt, depending on the level of sodium chloride required by users[2]. As a strategic commodity, salt has an important role in meeting the needs of industrial raw materials and household consumption[3]. In Indonesia, potential land for the development of the salt industry reaches 33,625 hectares, with 60% of the 20,821 hectares already utilized by around 19,503 salt farmers (Center for Statistics and Information Data, 2018). Salt production centers are scattered in various regions such as Cirebon, Indramayu, Sumenep, Pati, and other coastal areas[4].

Indonesia's vast coastal areas possess significant potential for salt production, which is crucial for meeting national demands for domestic use and industrial raw materials. However, environmental factors, especially air humidity, have a significant impact on how well salt is produced[5]. Weather variability frequently leads to less than ideal output results. Consequently, in order to estimate air humidity with great precision, a trustworthy prediction method is required. As technology develops, machine learning presents interesting instruments for environmental factor modeling and forecasting. The purpose of this study is to formulate the problem of which machine learning model CatBoost or LightGBM performs better in predicting air humidity in order to support the efficiency of salt production in Indonesia's coastal regions. In order to increase the productivity of the salt business, the research seeks to answer this question and advance the use of artificial intelligence in environmental prediction[6].

The application of machine learning techniques for predicting air humidity one of the crucial environmental parameters affecting salt production is the main focus of this study. In order to support effective salt harvesting, persistent and precise estimation of air humidity is crucial, especially considering the unpredictable weather in Indonesia's coastal regions. The comparison of two machine learning models, CatBoost and LightGBM, in terms of how well they forecast air humidity is the only focus of this study.

The study's comparison of CatBoost and LightGBM is based on how each model handles tabular data, which is the typical format of meteorological datasets like temperature, humidity, and sunshine intensity. Because CatBoost handles missing values and categorical variables automatically, it offers significant preprocessing advantages over other methods like One-Hot Encoding. The histogram-based learning and leaf-wise tree growth techniques of LightGBM, on the other hand, are well-known for their high model training efficiency, enabling quicker and more precise training on big datasets. Both models in this study showed good predictive performance, with low error rates and R2 values near 1, which suggests minimal overfitting and good generalization. Additionally, LightGBM tends to excel in training speed and result stability, while CatBoost provides better interpretability, especially for categorical features. Therefore, comparing these two models is important to determine the most appropriate algorithm for data-driven air humidity prediction systems that support strategic decision-making in salt production. Salt farmers can use the results of the CatBoost and LightGBM models with good predictions to make decisions such as scheduling drying or determining the right harvest time.It is expected that the use of these data-based predictions will increase production efficiency and reduce the risk of losses caused by unpredictable weather changes.

The LightGBM method, developed by Microsoft, offers effective Gradient Boosting algorithm capabilities. This model is designed to improve computational efficiency and prediction accuracy, especially on large datasets and with high feature dimensions[7]. With a histogram-based learning approach, LightGBM is able to speed up the training process compared to conventional boosting methods[8].

Meanwhile, CatBoost is one of the algorithms in the Gradient Boosting Decision Tree (GBDT) family that uses a decision tree as a base predictor and is specifically designed to handle categorical variables efficiently. The algorithm was introduced by Prokhorenkova and Dorogush with the aim of reducing information loss in large and complex datasets[9]. The advantages of CatBoost lie in its ability to handle large volumes of data, support parallel processing, and prevent overfitting through sophisticated internal tuning techniques[10].

In the context of salt production, CatBoost has the potential to provide high accuracy in predicting light intensity based on complex environmental data[11], especially when the data contains many categorical features such as region, land type, or time. Therefore, the utilization of CatBoost becomes relevant as one of the alternative models that can adaptively improve salt production efficiency[12].

This research is expected to answer the main problem of the salt industry, namely fluctuations in production yields due to uncertainty in weather conditions[13]. The results of this research can be used by salt farmers and stakeholders in the maritime industry as a consideration in a more accurate and data-based decision-making process. Thus, this research not only contributes to the development of predictive technology in the salt sector, but also supports the sustainable increase in productivity and resilience of the maritime-based economy.

II. METHODS

As shown in Figure 1, the flowchart of this research outlines in detail the four main stages, namely data collection, pre- processing, processing, result, and testing, which together form the foundation of the research methodology[14].

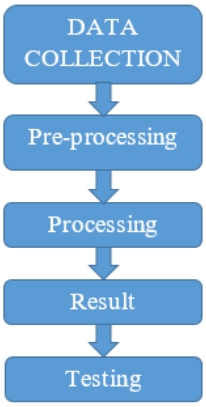


Figure 1. Research Stages

A. Hardware and Software

The process of training and testing machine learning models in this study was carried out using a laptop with the following specifications: Intel Core i5-9300H processor, 8 GB RAM, 256 GB SSD storage, NVIDIA GeForce GTX 1650 4GB graphics card, and Windows 10 Home operating system. The 15.6-inch screen allows optimal visualization of data and prediction results during the evaluation process. The selection of this device is tailored to the computational needs of the CatBoost and LightGBM algorithms, especially in handling datasets with medium to large sizes[14].

The cloud computing service Google Collaboratory, also known as Google Colab, was used to handle the data for this study. This service enables efficient code processing without relying entirely on local device specifications.

B. Data Collection and Preparation

The research data was stored and organized using Microsoft Excel to make it easier in the pre-processing and initial analysis stages, because the Python programming language has flexibility in data processing and various machine learning support libraries, such as scikit-learn, pandas, and special libraries CatBoost[15] and LightGBM[16]. The data was obtained from the official website Badan Meteorologi, Klimatologi, dan Geofisika (BMKG) (using the website https://dataonline.bmkg.co.id/) and was recorded at the Ahmad Yani Meteorological Station, located in Semarang, Central Java, Indonesia. It consists of 1,832 daily records collected over the period from 21 February 2020 to 22 February 2025, covering 11 meteorological features: measurement date (TANGGAL), minimum temperature (Tn), maximum temperature (Tx), average temperature (Tavg), average humidity (RH_avg), rainfall (RR), sunshine duration (ss), maximum wind speed (ff_x), wind direction at maximum wind speed (ddd_x), average wind speed (ff_avg), and predominant wind direction (ddd_car).

To ensure robustness against outliers and maintain the overall distribution of the data that are important for the requirements of predictive models in salt production, missing or null values in the dataset were handled prior to analysis using median imputation, which fills in any absent values with the median value of the corresponding variable.

The data collection and preparation process was the starting point of this study. A daily dataset recording air humidity (RH_AVG), along with other weather variables, including a recording date column, was used. Agencies responsible for monitoring meteorological conditions conduct field measurements on a regular basis. This data is derived from these measurements. The dataset is used to train and test predictive models that aim to estimate air humidity values based on recorded weather factors. Therefore, the data contained in the dataset is very important. Data loading is done using the pandas Python library[17], which is known to be powerful for managing tabular data[18], and has commands for reading csv files[19].

TABLE I.
DATA DESCRIPTION TABLE

Name	Data Type	Total
Measurement date (TANGGAL)	Integer	1832
Min Temperature (Tn)	Integer	1832
Max Temperature (Tx)	Integer	1832
Average Temp (Tavg)	Integer	1832
Humidity (RH_avg)	Integer	1832
Rainfall (RR)	Integer	1832
Sunlight (ss)	Integer	1832
Wind direction (ff_x)	Integer	1832
Wind velocity (ff_avg)	Integer	1832
Most wind direction (ddd_car)	Integer	1832
Wind direction Max (ddd_x)	Integer	1832

C. Pre-processing

The initial step of data cleansing is performed after the data has been entered into the programming environment. It handles invalid values. The presence of the character "-" in some cells in a data set is one type of irregularity found. This character semantically indicates that data is missing or unavailable. This "-" value is not recognized by Python in its original form as NaN (Not a Number), so it must be replaced explicitly for the applied imputation method to work correctly in the future[20]. The applymap() function is used to implement this replacement thoroughly. This function ensures that any element of type string containing "-" is converted to an empty value that can be identified by the system as a missing value.

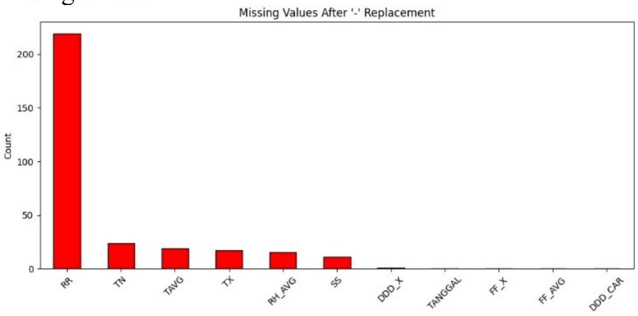


Figure 2. Missing Values after Replacement

To improve guarantee compatibility with the modeling techniques, categorical variables were encoded using the LabelEncoder approach[21]. In addition to making the data structure better, this step helped the model to numerically analyze categorical data. The dataset was normalized using MinMaxScaler, which is necessary for gradient-based methods like LightGBM or CatBoost, in order to bring all feature values into the same scale.

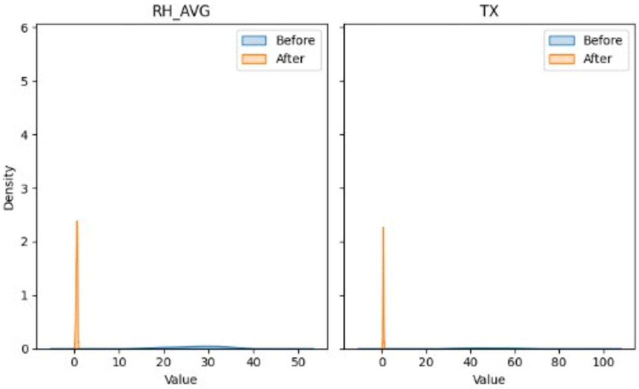


Figure 3. After Normalization RH_AVG & TX

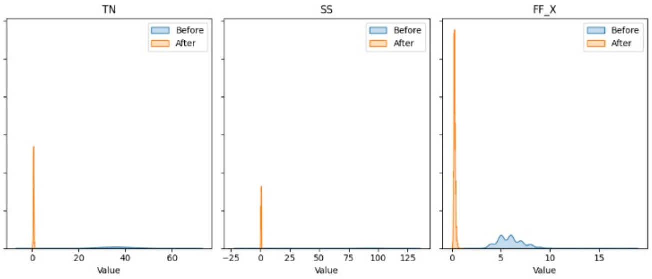
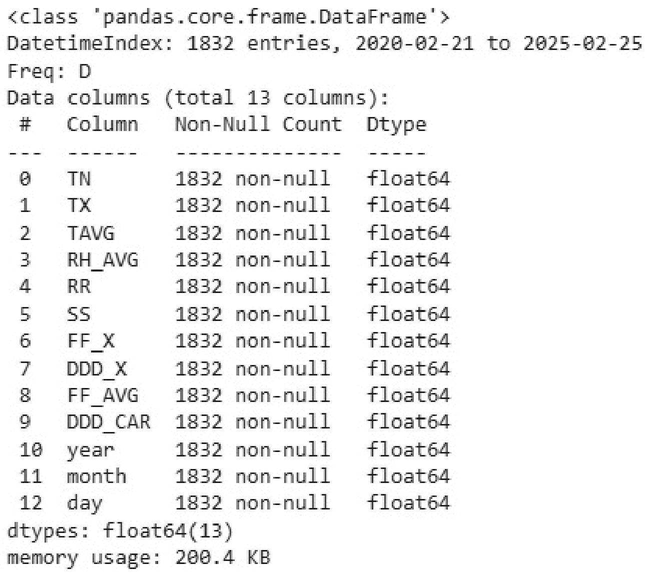


Figure 4. After Normalization TN, SS, FF_X

D. Feature Engineering

In addition, time data is also specially processed. By using the to_datetime function from the pandas library[22], the TANGGAL column can be converted from text to datetime format. This conversion is essential to enable systematic manipulation of time, such as the extraction of information about the year, month, and day. After the date column is converted, a split is made into three new columns, year, month, and day, each of which represents a more specialized time component. With this split, you can perform temporal analysis such as observing seasonal patterns or changes in air humidity each month. To avoid data redundancy, the original TANGGAL column was removed. This transformation results in a dataset structure that is more structured, consistent, and ready for further analysis.



```
<class 'pandas.core.frame.DataFrame'>
DatetimeIndex: 1832 entries, 2020-02-21 to 2025-02-25
Freq: D
Data columns (total 13 columns):
 #   Column   Non-Null Count  Dtype  
---  ------   --------------  -----  
 0   TN       1832 non-null   float64
 1   TX       1832 non-null   float64
 2   TAVG     1832 non-null   float64
 3   RH_AVG   1832 non-null   float64
 4   RR       1832 non-null   float64
 5   SS       1832 non-null   float64
 6   FF_X     1832 non-null   float64
 7   DDD_X    1832 non-null   float64
 8   FF_AVG   1832 non-null   float64
 9   DDD_CAR  1832 non-null   float64
 10  year     1832 non-null   float64
 11  month    1832 non-null   float64
 12  day      1832 non-null   float64
dtypes: float64(13)
memory usage: 200.4 KB
```

Figure 5. Data using datetime

E. Modelling

1) LightGBM: After data preparation, the dataset was divided into training and testing subsets using an 80/20 ratio to evaluate the model's generalization performance. The LightGBM regressor was trained on 80% of the data, while the remaining 20% was reserved for testing the model's predictive accuracy. To optimize model performance, hyperparameter tuning was conducted using the GridSearchCV method, focusing on four parameters: n_estimators, learning_rate, max_depth, and num_leaves. Although a single predefined set of hyperparameters was evaluated, GridSearchCV systematically tested combinations using 3-fold cross-validation (cv=3). In this process, the training data was partitioned into three subsets; the model was trained on two of them and validated on the third in a rotating sequence, ensuring more robust performance estimation and reducing the risk of overfitting to a specific subset.

TABLE II.
LIGHTGBM MODEL DESCRIPTION

Aspect	Description
Model	LightGBM Regressor
Train-Test Split	80% Training / 20% Test
Hyperparameter Tuning	GridSearchCV
Parameters Tuned	n_estimators, learning_rate, max_depth, num_leaves
Combinations	1 (single predefined combination)
Cross-Validation	3-Fold (cv=3)
CV Process	Train on 2 folds, validate on 1 fold (rotated 3 times)
Evaluations Metrics	R², Mean Absolute Error (MAE), Root Mean Squared Error (RMSE)
Visualizations Used	Scatter Plot, Residual Plot, Histogram, Feature Importance

2) CatBoost: To be able to further explore the reliability as well as efficiency of the LightGBM model, CatBoost was tested as a different gradient boosting technique. A structured hyperparameter tuning method was used to train the CatBoost regressor after the same 80/20 train-test split. 3-fold cross-validation was implemented to validate the tuning, even though only one preset set of hyperparameters—covering iterations, learning_rate, depth, and l2_leaf_reg—was used.

TABLE III.
CATBOOST MODEL DESCRIPTION.

Aspect	Description
Model	CatBoost Regressor
Train-Test Split	80% Training / 20% Test
Hyperparameter Tuning	GridSearchCV
Parameters Tuned	Iterations, learning_rate, depth, 12_leaf_reg
Combinations	1 (single predefined combination)
Cross-Validation	3-Fold (cv=3)
CV Process	Train on 2 folds, validate on 1 fold (rotated 3 times)
Evaluations Metrics	R², Mean Absolute Error (MAE), Root Mean Squared Error (RMSE)
Visualizations Used	Scatter Plot, Residual Plot, Feature Importance

This guaranteed that the model's setup was dependable and constant across various training data segments. On the test set, the CatBoost model showed acceptable outcomes despite its restricted hyperparameter search space. Evaluation criteria including R², mean absolute error, and root mean squared error showed that the model could successfully identify patterns in the data. The model's dependability was further reinforced by visual examinations using scatter plots, time series comparisons, and residual diagnostics. In addition to providing a useful performance benchmark, this complementary application of CatBoost showed how adaptable tree-based ensemble algorithms are for predicting air humidity, especially when little modification is required to achieve high predictive accuracy.

III. RESULT AND DISCUSSION

To figure out how well a trained model captures real-world patterns, model evaluation is an essential stage in predictive data analysis. right after preprocessing, an 80/20 train-test split was used to divide the dataset, guaranteeing that test data was minimized during training. Using the standard regression measures of R² (coefficient of determination), MAE (mean absolute error), and RMSE (root mean squared error), the LightGBM and CatBoost models were both calibrated. The quantitative evaluation was further supported by visualization approaches such as feature importance rankings, scatter plots, and residual plots.

Using GridSearchCV, the LightGBM regressor was adjusted throughout a specific set of hyperparameters: num_leaves, max_depth, learning_rate, and n_estimators. Threefold cross-validation was used during this tuning process to guarantee stability and prevent overfitting. The following results were obtained by the optimized model on the test set.

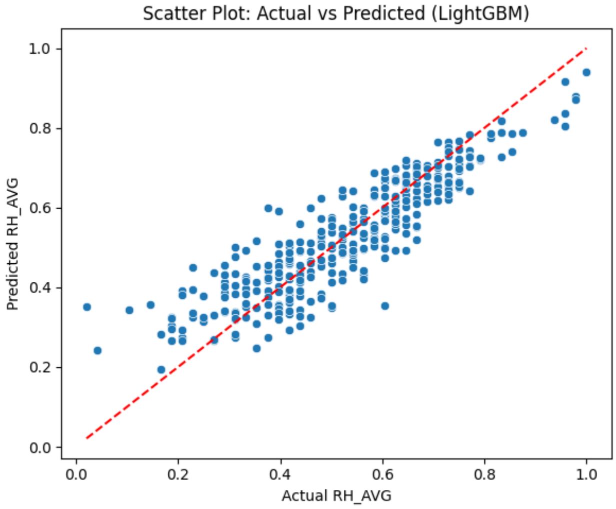


Figure 6. ScatterPlot for LightGBM

A residual plot was created after prediction for the purpose to analyze the LightGBM regressor's outcomes in more detail. The predicted values were displayed against the residuals, which are the difference between the actual values (y_test) and the anticipated values (y_pred). The residuals of an acceptable regression model should be scattered randomly around zero with no obvious pattern. The LightGBM residual plot in this particular case showed a symmetrical and uniformly distributed spread around the zero line, showing no bias in predictions and pointing to good generalization of the model.

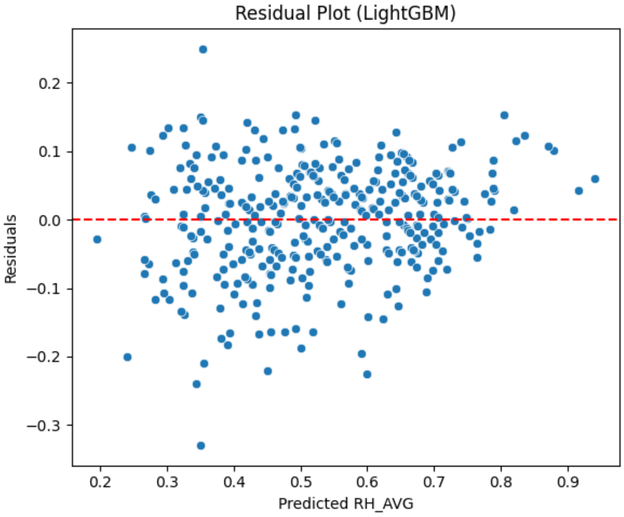


Figure 7. ResidualPlot for LightGBM

The evaluation data collected throughout testing provide additional support for this visual observation. A Tuned R² score of 0.7981 was obtained by the LightGBM model after it was tuned with parameters like learning_rate=0.01, max_depth=5, min_child_samples=10, n_estimators=500, and num_leaves=31. This means that the model explains around 80% of the variation in the test data. Moderate prediction errors are indicated by the Root Mean Squared Error (RMSE) of 0.0786 and the Mean Absolute Error (MAE) of 0.0617. These numbers imply that although the model makes predictions that are passably accurate, its accuracy could be increased.

Table IV. LIGHTGBM METRICS RESULT

Metric	Value
R²	0.7981
Mean Absolute Error (MAE)	0.0617
Root Mean Squared Error (RMSE)	0.0786

The CatBoost model, that had been modified with iterations=500, learning_rate=0.05, depth=6, and l2_leaf_reg=5, achieved a Tuned R² score of 0.8191, which was slightly more accurate in terms of variance explanation. With an RMSE of 0.0744 and an MAE of 0.0570, the results showed better prediction accuracy and less average errors than LightGBM. These findings are in line with the CatBoost residual plot, which shows less bias and dependable generalization due to the prediction errors' close clustering around zero and good distribution.

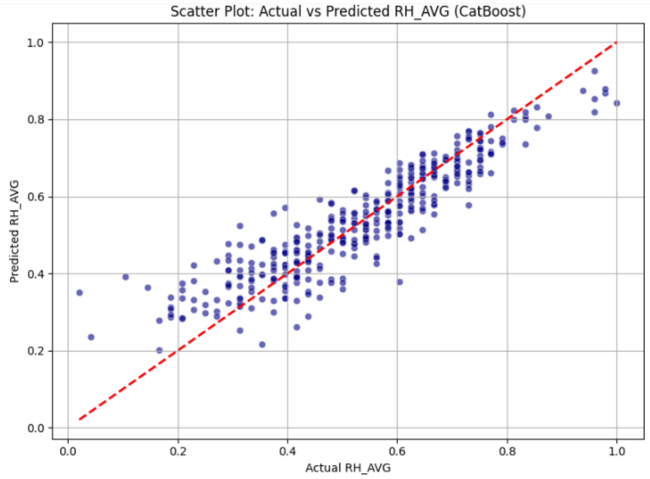


Figure 8. Scatterplot for CatBoost

A scatter plot comparing real and anticipated outcomes was used to assess the CatBoost model. It revealed a generally strong alignment along the ideal prediction line. Despite a certain apparent spread, this alignment shows that the model was able to follow the trend of actual RH_AVG values very well. A residual plot was looked at in order to evaluate these projections reliability even more. According to the results, there were no noticeable trends or heteroscedasticity in the residuals, which indicate the difference between expected and actual values, and they were distributed very symmetrically around the zero line. The residual distribution's randomness indicates that the CatBoost model generalizes well across a range of expected values and is not significantly impacted by systematic bias.

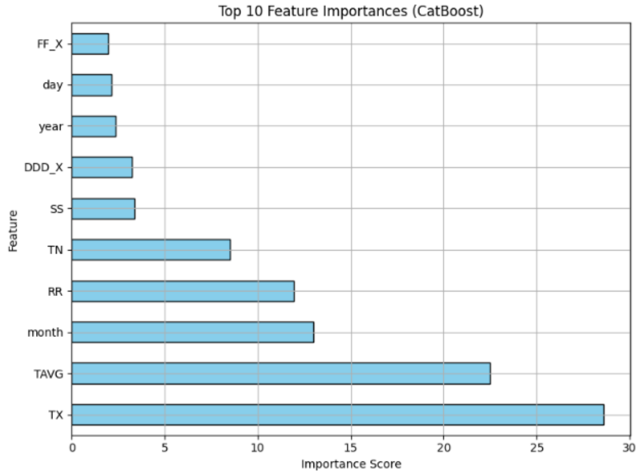


Figure 9. Feature Importance using CatBoost

A feature importance analysis was performed with the CatBoost model to improve interpretability. Maximum temperature (TX), average temperature (TAVG), and month were the top three contributors, and each had a significant effect in predicting relative humidity. These characteristics are consistent with known meteorological principles since temperature directly impacts the atmosphere's ability to retain moisture and seasonal patterns, which are represented by the month to have an impact on humidity levels. Smaller but still significant contributions were made by TN (minimum temperature) and RR (rainfall). The outcomes support the model's accuracy and applicability of its predictions by confirming that it caught significant environmental influences.

IV. CONCLUSION

In conclusion, this study effectively shows that how well the machine learning algorithms LightGBM and CatBoost predicted air humidity based on BMKG weather data. Following preprocessing and normalization, GridSearchCV was used to improve both models using 3-fold cross-validation with 80/20 train-test splits. According to the results, CatBoost outperformed LightGBM in generalization, as demonstrated by a higher R2 score (0.8191 vs. 0.7981) and lower error metrics (MAE and RMSE). The predictive behavior and interpretability of the models were further confirmed by visualization approaches such as scatter plots, residual plots, and feature importance. For example, the most important factors were maximum temperature, average temperature, and month. These results show that both models can be effectively implemented into air humidity prediction systems, especially for weather-sensitive sectors and decision-making processes that affect the effectiveness of salt production.

BIBLIOGRAPHY

[1] R. Hartati, W. Widianingsih, B. W. RTD, M. B. Puspa, and E. Supriyo, “Analisa Air Tambak Desa Kaliwlingi sebagai Bahan Baku Produksi Garam Konsumsi,” *J. Mar. Res.*, vol. 11, no. 4, pp. 657–666, 2022, doi: 10.14710/jmr.v11i4.35353.

[2] S. Redjeki, “Produksi Garam Industri Dari Garam Rakyat Industrial Salt Production From People’s Salt.”

[3] I. Sulistiyawati, N. L. Rahayu, M. Falah, and W. M. Endris, “Konsumsi Garam Beryodium Sebagai Upaya Preventif Penyakit Gaky Di Masyarakat.”

[4] O. Putri and T. Sugiarti, “Perkembangan dan Faktor yang Mempengaruhi Permintaan Volume Impor Garam Industri di Indonesia,” *J. Ekon. Pertan. dan Agribisnis*, vol. 5, no. 3, pp. 748–761, Jul. 2021, doi: 10.21776/ub.jepa.2021.005.03.13.

[5] R. Sunoko, A. Saefuddin, R. Syarief, and N. Zulbainarni, “Proteksionisme dan Standardisasi Garam Konsumsi Beryodium,” *J. Kebijak. Sos. Ekon. Kelaut. dan Perikan.*, vol. 12, no. 2, p. 101, Dec. 2022, doi: 10.15578/jksekp.v12i2.11077.

[6] P. : Jurnal *et al.*, “Hasnawati Amqam 147 | P a g e Kelimpahan dan Karakteristik Mikroplastik pada Produk Garam Tradisional di Kabupaten Jeneponto Abundance and Characteristic of Microplastics in Traditional Salts in Jeneponto”.

[7] E. Febriantoro, E. Setyati, and J. Santoso, “Pemodelan Prediksi Kuantitas Penjualan Mainan Menggunakan LightGBM,” *SMARTICS J.*, vol. 9, no. 1, pp. 7–13, Apr. 2023, doi: 10.21067/smartics.v9i1.8279.

[8] P. Septiana Rizky, R. Haiban Hirzi, U. Hidayaturrohman, U. Hamzanwadi Selong Jl TGKH Muhammad Zainuddin Abdul Madjid Pancor, and L. Timur, “Perbandingan Metode LightGBM dan XGBoost dalam Menangani Data dengan Kelas Tidak

Seimbang," 2022. [Online]. Available: www.unipasby.ac.id

[9] A. Darmawan *et al.*, "Implementasi Catboost Menggunakan Hyper-Parameter Tuning Bayesian Search Untuk Memprediksi Penyakit Diabetes."

[10] Andrian Febriansyah Istianto, Fajri Rakhmat Umbara, and Asep Id Hadiana, "Prediksi Curah Hujan Menggunakan Metode Categorical Boosting (Catboost)," Jul. 2023. doi: https://doi.org/10.36040/jati.v7i4.7304.

[11] O. Pahlevi, D. Ayu, N. Wulandari, L. K. Rahayu, H. Leidiyana, and Y. Handrianto, "Bulletin Of Computer Science Research Model Klasifikasi Risiko Stunting Pada Balita Menggunakan Algoritma CatBoost Classifier," *Media Online)*, vol. 6, no. 4, pp. 414–421, 2024, doi: 10.47065/bulletincsr.v4i6.373.

[12] E. Mumpuni, "Implementasi Shap Pada Catboost Untuk Meningkatkan Akurasi Prediksi Temperatur Udara Di Kota Pekanbaru," 2024.

[13] Ali Armadi *et al.*, "Pengabdian Budidaya Garam Dan Dampak Dari Peluasan Wilayah Tambak Garam Beserta Penanaman Pohon Di Desa Galis Kec. Gili Genting," *J. Pengabdi. Masy. Nusant.*, vol. 5, no. 3, pp. 147–152, Sep. 2023, doi: 10.57214/pengabmas.v5i3.359.

[14] M. D. Firmansyah, I. Rizqa, F. A. Rafrastara, and W. Ghozi, "Balancing CICIoV2024 Dataset with RUS for Improved IoV Attack Detection," vol. 9, no. 2, pp. 250–257, 2025.

[15] M. T. Syamkalla, S. Khomsah, and Y. S. R. Nur, "Implementasi Algoritma Catboost Dan Shapley Additive Explanations (SHAP) Dalam Memprediksi Popularitas Game Indie Pada Platform Steam," *J. Teknol. Inf. dan Ilmu Komput.*, vol. 11, no. 4, pp. 777–786, Aug. 2024, doi: 10.25126/jtiik.1148503.

[16] S. Diantika, "Penerapan Teknik Random Oversampling Untuk Mengatasi Imbalance Class Dalam Klasifikasi Website Phishing Menggunakan Algoritma LightGBM," 2023.

[17] L. Pappalardo, F. Simini, G. Barlacchi, and R. Pellungrini, "scikit-mobility: A Python Library for the Analysis, Generation, and Risk Assessment of Mobility Data," *J. Stat. Softw.*, vol. 103, no. 4, 2022, doi: 10.18637/jss.v103.i04.

[18] A. Kotelnikov, D. Baranchuk, I. Rubachev, and A. Babenko, "TabDDPM: Modelling Tabular Data with Diffusion Models," *Proc. Mach. Learn. Res.*, vol. 202, pp. 17564–17579, 2023.

[19] I. Arora, "Improving Performance of Data Science Applications in Python," *Indian J. Sci. Technol.*, vol. 17, no. 24, pp. 2499–2507, 2024, doi: 10.17485/ijst/v17i24.914.

[20] S. Jäger, A. Allhorn, and F. Bießmann, "A Benchmark for Data Imputation Methods," *Front. Big Data*, vol. 4, no. July, pp. 1–16, 2021, doi: 10.3389/fdata.2021.693674.

[21] S. Khedkar, S. Lambor, Y. Narule, and P. Berad, "Categorical Embeddings for Tabular Data using PyTorch," *ITM Web Conf.*, vol. 56, p. 02002, 2023, doi: 10.1051/itmconf/20235602002.

[22] S. Masuda, T. Tateishi, and T. Takahashi, "Datetime Feature Recommendation Using Textual Information," *Procedia Comput. Sci.*, vol. 225, pp. 617–625, 2023, doi: 10.1016/j.procs.2023.10.047.