e-ISSN: 2582-5208
International Research Journal of Modernization in Engineering Technology and Science
(Peer-Reviewed, Open Access, Fully Refereed International Journal)
Volume:05/Issue:10/October-2023 Impact Factor- 7.868 www.irjmets.com

ASSESSING THE BEST REGRESSION APPROACH FOR PRECIPITATION ANALYSIS WITH METEOROLOGICAL DATA IN LUCKNOW USING PYCARET

Devasheesh Krishan[*1], A. Singh[*2]

[*1]M.Tech Student, Department Of Civil Engineering, Institute Of Engineering And Technology, Lucknow, Uttar Pradesh- 226021, India.

[*2]Department Of Civil Engineering, Institute Of Engineering And Technology, Lucknow, Uttar Pradesh- 226021, India.

*Corresponding Author : Devasheesh Krishan

DOI : https://www.doi.org/10.56726/IRJMETS45024

ABSTRACT

Accurate precipitation forecasting is of paramount importance in various fields, including agriculture, hydrology, environmental engineering and disaster management. This study investigates the suitability and performance of different regression models in predicting precipitation based on meteorological variables. Seven meteorological factors, including mean PBLH (Planetary Boundary Layer Height), wind speed, humidity, evapo-transpiration,, NDVI (Normalized Difference Vegetation Index), surface radiation and temperature, are considered as potential predictors of rainfall in this study.

The objective is to identify the most effective regression model for quantifying the dependency of precipitation on these meteorological parameters. To achieve this, a comprehensive comparative analysis of regression models available in the PyCaret framework is conducted. Models such as Linear Regression, Lasso Regression, Random Forest Regression, K-Neighbors Regression and XGradient Boosting Regression, among others, are evaluated and compared. The best model is chosen based on certain parameters. The assessment includes metrics such as Mean Absolute Error (MAE), Mean Squared Error (MSE), R-squared (R2) and Root Mean Squared Error (RMSE) to measure predictive accuracy. Additionally, residual analysis, learning curves, and error plots are employed to gain insights into model behaviour.

Results from this study offer valuable guidance for selecting the most appropriate regression model for precipitation forecasting in different scenarios. Understanding the strengths and weaknesses of various models can lead to improved accuracy in predicting this critical meteorological variable, ultimately benefiting applications ranging from agriculture to disaster preparedness.

Keywords: Precipitation Forecasting; Regression Models; Meteorological Variables; PyCaret; Residual Analysis.

I. INTRODUCTION

Precipitation forecasting is an important part of meteorological research and has far-reaching implications in various areas, including agriculture, urban planning, management of water resources, and disaster readiness. Accurate prediction of precipitation events, their intensity, and timing is essential for informed decision-making and risk mitigation in these areas. In the context of India, where monsoonal rains significantly impact daily life and regional economies, precise precipitation forecasting is of paramount importance.

The northern Indian city of Lucknow, the capital of the state of Uttar Pradesh, is one such place affected by the heavy rains. Famous for its historical and cultural heritage, Lucknow enjoys a semi-arid climate characterized by distinct rainy and dry seasons. The rainy season, which usually starts in June and lasts until September, plays an important role in the city's climate. Adequate rainfall is essential for agricultural production [1] and water recharge [2, 3], while excessive or unseasonal rainfall can cause flooding and related complications [4]. So we decided to focus our study on Lucknow.

In recent years, advances in meteorological data collection and computational capabilities have opened up new avenues for improving precipitation forecasting. Traditional statistical methods have been complemented and, in some cases, replaced by machine learning and data-driven regression models, offering the potential for greater accuracy and reliability [5, 6, 7, 8].

A study carried out by Chaudhari and Choudhari [9] suggests that temperature, cyclone, and wind are some vital components of the atmosphere over the Indian landmass that effect rainfall. But the study didn't measure the correlations between the features to determine the effect of the independent characteristics on the rainfall.

Another paper by Thirumalai et al. [10] describes solar radiation, water vapour in air and some other diurnal features as important factors required for rainfall. In another study by Ganasekran et al. [11], temperature, wind speed, relative humidity, etc. are identified as bringers of precipitation. These parameters are common in another study conducted in Kerala as well [12].

This study focuses on Lucknow as an example of a region with a distinct monsoonal climate, making it an ideal candidate for investigating the effectiveness of various regression models in predicting precipitation. Assessing a big dataset of meteorological variables, including Planetary Boundary Layer Height (PBLH), wind speed, humidity, evapo-transpiration, and others, we aim to determine which regression model(s) predict the most precise and reliable relationship of precipitation with other variables in this context. With the increase in availability of data and computing prowess recently, machine learning has become an important part of the upcoming generation of rainfall forecasting models. Modern machine learning methods provide a means to learn temporal dynamics in a purely data-driven manner [13].

By conducting a comparative analysis of regression models, which have worked well in predicting rainfall in a study in Senegal [14] using the PyCaret framework, we seek to contribute valuable insights into the selection of appropriate models for precipitation forecasting. The findings of this study are expected to enhance our understanding of the relationship between meteorological variables and precipitation in Lucknow and have broader applications in monsoonal regions across India.

Through a rigorous evaluation of model performance, we aim to assist meteorologists, researchers, and policymakers in making more informed decisions related to precipitation forecasting and its impact on various sectors in Lucknow and beyond.

II. MATERIALS AND METHODOLOGY

2.1 Study area

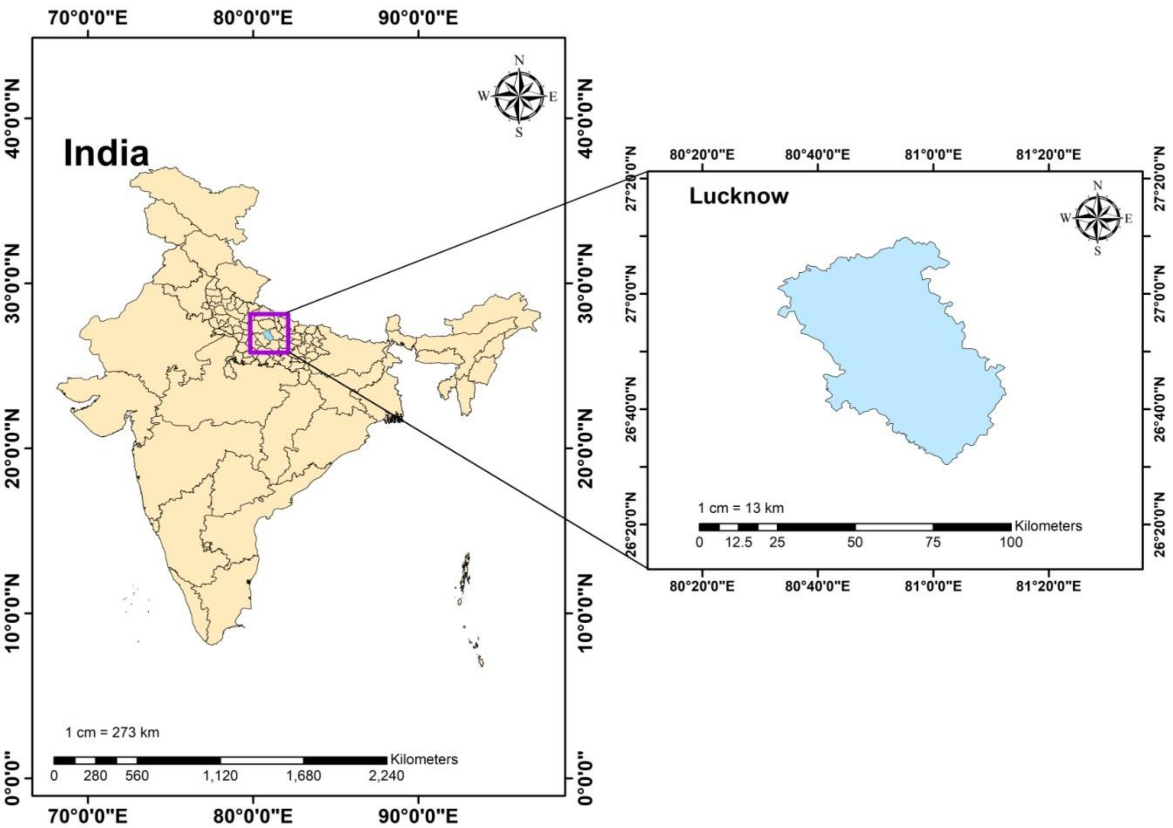

Fig. 1 Study area (Lucknow)

e-ISSN: 2582-5208
International Research Journal of Modernization in Engineering Technology and Science
(Peer-Reviewed, Open Access, Fully Refereed International Journal)
Volume:05/Issue:10/October-2023 Impact Factor- 7.868 www.irjmets.com

We chose Lucknow, the capital city of the second largest state of India, Uttar Pradesh, as the site for this study. The city is located between latitudes 26^0 44' 08" and 26^0 57' 57" north and longitudes 80^0 49' 50" and 81^0 03' 14" east. Lucknow is also the most populated city of the state, with the population being around 28 lakhs when the last census was conducted in 2011. Surrounded on the east side by Barabanki, on the west side by Unnao, on the south corner by Rae-bareli and in the north side by Sitapur and Hardoi, Lucknow is situated on the north-western shore of the Gomti River. Infact the Gomti river passes through the middle of the city.

Lucknow is located at an average height of about 123 metres from the Mean Sea Level. Being far from the sea and located in a landlocked state, Lucknow has a continental type of climate. The rainy season in Lucknow lasts for about four months, with the monsoon clouds reaching Lucknow around 20^{th} of June each year and staying until the end of September. There is sparse rain in summers and winters but that is mainly due to cyclonic effects (from Arabian Sea or Bay of Bengal) or due to prevalence of western disturbances, respectively.

2.2 Data Collection

The data for this study was collected from Goddard Earth Sciences Data and Information Services Center, or GES DISC, Interactive Online Visualization and Analysis Infrastructure, which is an acronym for GIOVANNI. It is an online tool developed by NASA wherein we can find satellite data for various atmospheric variables like precipitation, surface radiation, humidity, transpiration, Aerosols, etc. GIOVANNI provides free access to data and thus is helpful to researchers and students of climatic science around the world.

The precipitation data used in this study was provided by Tropical Rainfall Measuring Mission, or TRMM. It was a research satellite launched by JAXA, the Japanese space agency in collaboration with NASA. It gives both daily and monthly average precipitation values and we have taken the monthly average precipitation values in this study.

The other parameters used in this study to identify their relationship with precipitation are mean PBLH (Planetary Boundary Layer Height), wind speed, humidity, evapo-transpiration,, NDVI (Normalized Difference Vegetation Index), surface radiation and temperature.

The PBLH data is provided by Atmospheric Infrared Sounder, which is a hyperspectral infrared sounder on the NASA's Aqua and Merra satellites. The Aqua satellite contains, among other instruments, Moderate Resolution Imaging Spectroradiometer (MODIS), which is responsible for providing a range of datasets including surface temperatures, reflectance, precipitation, evapo-transpiration and NDVI. It is used in this study to provide data for NDVI. The Modern-Era Retrospective analysis for Research and Applications, Version 2 (MERRA-2), or Merra 2 satellite is used to provide data for Specific Humidity and PBLH. The Merra 2 satellite enables assimilation of modern hyperspectral radiance and microwave observations, along with GPS-Radio Occultation datasets. NASA Global Land Data Assimilation System Version 2 (GLDAS-2), is another satellite that is used to provide three critical datasets in this study, namely potential evapo-transpiration, surface air pressure and instantaneous wind speed.

2.3 PyCaret

PyCaret is a Python open source machine learning library designed to make performing standard tasks in a machine learning project easy. It is a Python version of the Caret machine learning package in R, and is quite famous these days as it permits models to be analysed, contrasted, and adjusted on any given set of data with just some lines of python code.

The PyCaret library provides these aspects, permitting the machine learning coder in Python to check on the spot a number of standard machine learning algorithms on a classification or regression dataset with a single call of the function.

PyCaret stands as a remarkable Python library, tailor-made to simplify and expedite the entire machine learning journey, including model deployment. It's a tool that takes the complexity out of the process, allowing data scientists and analysts to focus on the creative aspects of predictive modeling, rather than getting caught in intricate technicalities.

Fundamentally, PyCaret gives a wide range of functionalities, covering everything from handling data pre-processing and feature engineering to model selection, hyper-parameter tuning, and model evaluation. What

e-ISSN: 2582-5208
International Research Journal of Modernization in Engineering Technology and Science
(Peer-Reviewed, Open Access, Fully Refereed International Journal)
Volume:05/Issue:10/October-2023 Impact Factor- 7.868 www.irjmets.com

makes it different is its user-friendly design, making it equally accessible to amateurs and professionals in the machine learning area.

PyCaret starts off by automating those routine data pre-processing works like filling in missing data, managing outliers, and normalizing features. It even supports advanced techniques such as categorical variable encoding and feature selection, ensuring that our data is primed for modelling.

One standout feature is PyCaret's ability to effortlessly compare and fine-tune multiple machine learning models. Just specify your target variable, and PyCaret takes care of the rest, training and evaluating a suite of algorithms to provide a comprehensive view of model performance. This simplifies the process of selecting the best-performing models for further refinement.

Hyper-parameter tuning is another area where PyCaret shines. It simplifies the optimization process by automating the search for the best combination of hyper-parameters, enhancing model performance without the need for extensive manual tuning.

In addition to its modelling capabilities, PyCaret offers insightful visualizations and reports to aid in model interpretation and is thus easily explainable. It provides feature importance plots, confusion matrices, learning curves, and more, helping users understand how models arrive at their predictions.

When the ideal model is identified, PyCaret offers a straightforward path to deploying it in production. Saving, loading, and deploying models in various formats becomes easy, making it practical for real-world applications.

In essence, PyCaret is a versatile and efficient machine learning tool that makes the machine learning pipeline accessible to all. It empowers data professionals to create, fine-tune, and deploy sophisticated models with minimal hassle, making it an invaluable asset for accelerating the development of predictive analytics solutions, both in research and industry contexts.

Firstly, we checked the data manually for any missing values. It was noticeable that the PBLH data had records starting from February 2000. That meant it was missing a single value as all other datasets were starting from January 2000. So it was imperative that we fill the values for proper functioning of the code. So we interpolated the missing value from the next available twelve month data. Now the dataset was complete.

The csv file was uploaded into the code with the help of 'upload' function. The steps followed were as follows:

Data Loading: We began by importing the necessary libraries, including pandas for data manipulation and matplotlib.pyplot for plotting. We used the pd.read_csv function from the pandas library to load the CSV file. After loading the data, we converted the "time" column to a datetime format using pd.to_datetime. This likely transformed a date-time string into a datetime object. Next, we filtered the data based on the "year" in the "time" column. For this, we created two separate data-frames:

train_data: This dataframe contains data from years 2000 to 2015.

test_data: This dataframe contains data from years 2016 to 2019.

Next, we initiated the PyCaret experiment setup using the setup function. We specified that the target variable is "mean_TRMM_3B43_7_precipitation", which is the mean monthly precipitation data. This step prepares the data for modelling within the PyCaret framework. It handles data pre-processing, splitting, and other setup tasks. Once we had our data set up, we could use PyCaret to compare and select the best-performing regression models. This is often done using the compare_models function, which is available within the PyCaret library. In our case, since we're studying the dependency of precipitation on several variables, we selected regression models like Linear Regression, Extra Trees Regression, Lasso Regression, Random Forest Regression, and more.

After selecting initial models, you can further improve their performance through hyper-parameter tuning. PyCaret provides the 'tune_model' function for this purpose. Tuning involves finding the best combination of hyper-parameters for each model to enhance their predictive capabilities. Thereafter, we evaluated the tuned models to understand how well they performed on our dataset. PyCaret offers various evaluation metrics for regression tasks, such as Mean Absolute Error (MAE), Mean Squared Error (MSE), Root Mean Squared Error (RMSE), R-squared (R2), and more.

Finally, visualizations are crucial for gaining insights into our data and model performance. Hence we used PyCaret's plot_model function to create various types of visualizations.

III. RESULTS AND DISCUSSION

The various regression models' performance is illustrated in the table below:

	Model	MAE	MSE	RMSE	R2	RMSLE	MAPE	TT (Sec)
knn	K Neighbors Regressor	34.0342	2819.5798	52.0659	0.6975	1.4136	44.9612	0.0950
lightgbm	Light Gradient Boosting Machine	35.0635	3087.0447	54.2806	0.6895	1.2888	35.4515	0.2590
et	Extra Trees Regressor	34.2399	3107.6804	54.8962	0.6804	1.1668	30.1242	0.3260
rf	Random Forest Regressor	34.9519	3224.4137	55.7701	0.6769	1.1978	56.2306	0.5700
ada	AdaBoost Regressor	40.0005	3260.2341	56.2676	0.6655	1.6961	124.2579	0.2250
lr	Linear Regression	41.5262	3333.5797	56.8438	0.6577	1.6525	109.3231	0.4180
lar	Least Angle Regression	41.5843	3357.3107	57.0848	0.6539	1.6467	106.7247	0.0940
gbr	Gradient Boosting Regressor	35.5308	3399.3654	57.3790	0.6492	1.2191	22.7178	0.2470
lasso	Lasso Regression	43.7266	3564.4277	58.7115	0.6329	1.7615	130.2020	0.1560
llar	Lasso Least Angle Regression	43.7260	3564.3956	58.7112	0.6329	1.7617	130.2106	0.0930
en	Elastic Net	44.4079	3609.3889	58.9413	0.6319	1.7924	138.0301	0.0970
ridge	Ridge Regression	43.5099	3542.2995	58.6780	0.6305	1.7150	128.9064	0.1550
br	Bayesian Ridge	45.3801	3792.8175	60.5207	0.6115	1.8213	157.5272	0.0900
xgboost	Extreme Gradient Boosting	36.3715	3899.3168	61.1911	0.5963	1.1383	9.9615	0.3070

Fig. 2 Model Comparison using PyCaret

As is evident from the table, K-Neighbors regressor outperformed all the other models in most of the parameters except two, in RMSLE and MAPE, in which XGBoost regressor had the best values. But since MSE, MAE, R-square (R^2) are better predictors of a model's capabilities, K-Neighbors regressor is the best available model for determining relationships between precipitation and other atmospheric variables.

Thereafter we proceeded to plot the validation curve, residual curve, prediction error curve and learning curve for the best model based on R^2 values. They are as shown below:

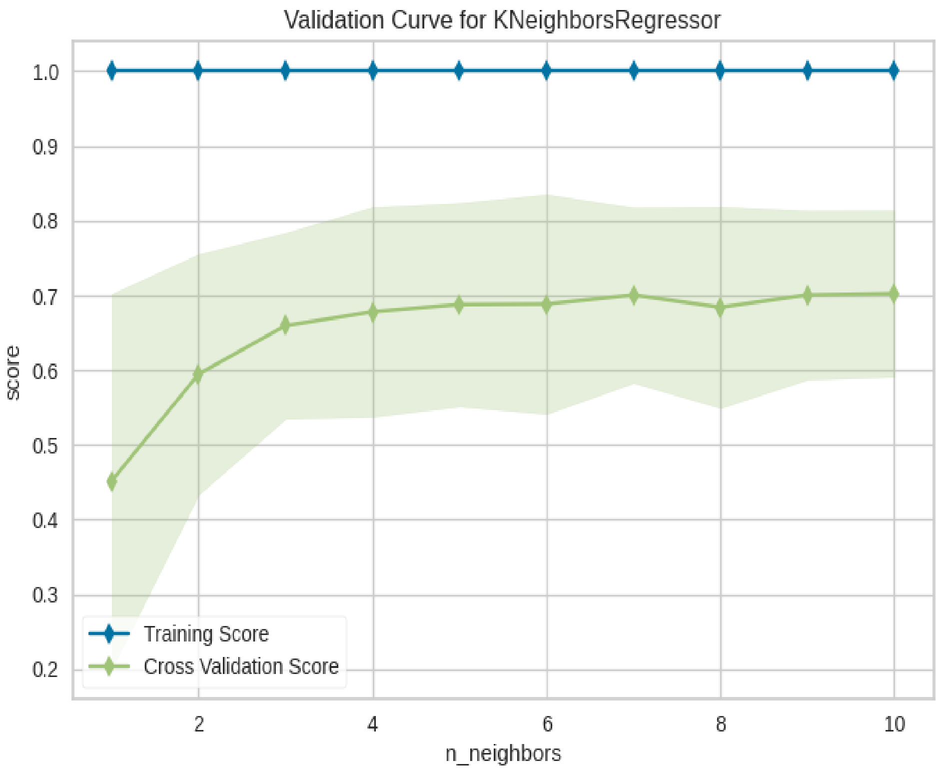

Fig. 3 Validation Curve for K-neighbors regression

e-ISSN: 2582-5208
International Research Journal of Modernization in Engineering Technology and Science
(Peer-Reviewed, Open Access, Fully Refereed International Journal)
Volume:05/Issue:10/October-2023 Impact Factor- 7.868 www.irjmets.com

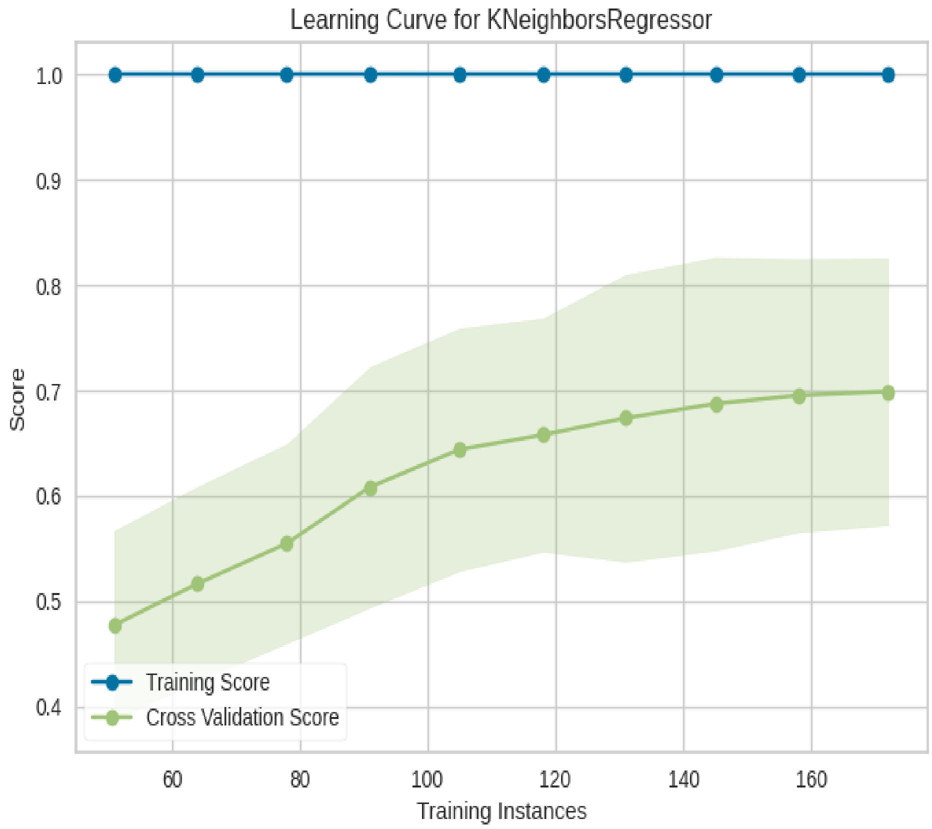

Fig. 4 Learning Curve for K-neighbors regression

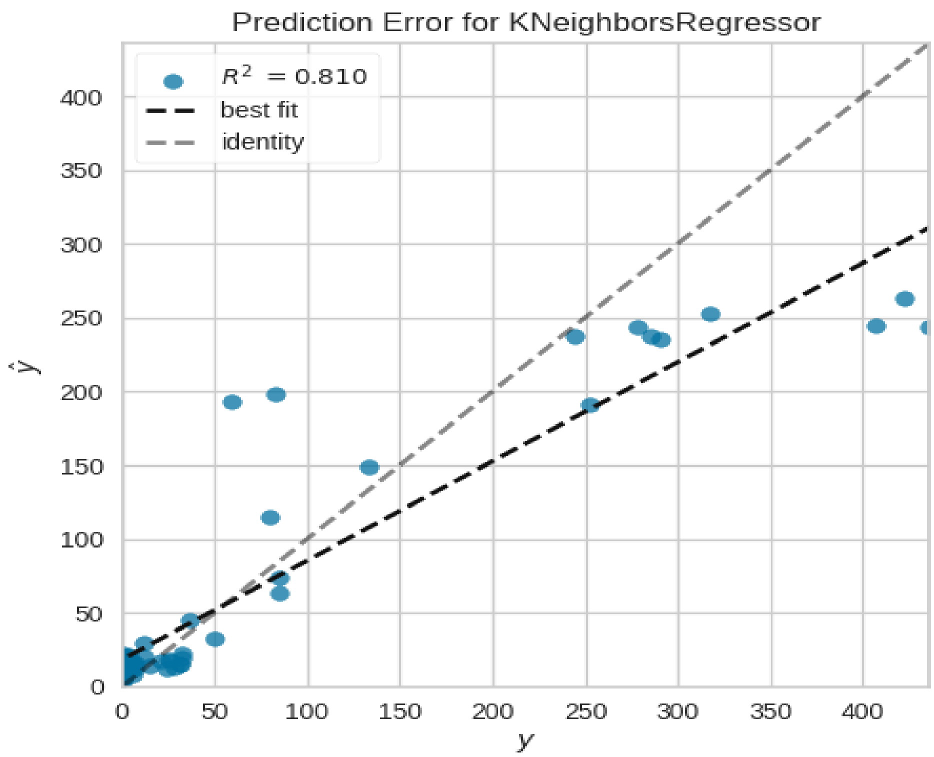

Fig. 5 Prediction Error Curve for K-neighbors regression

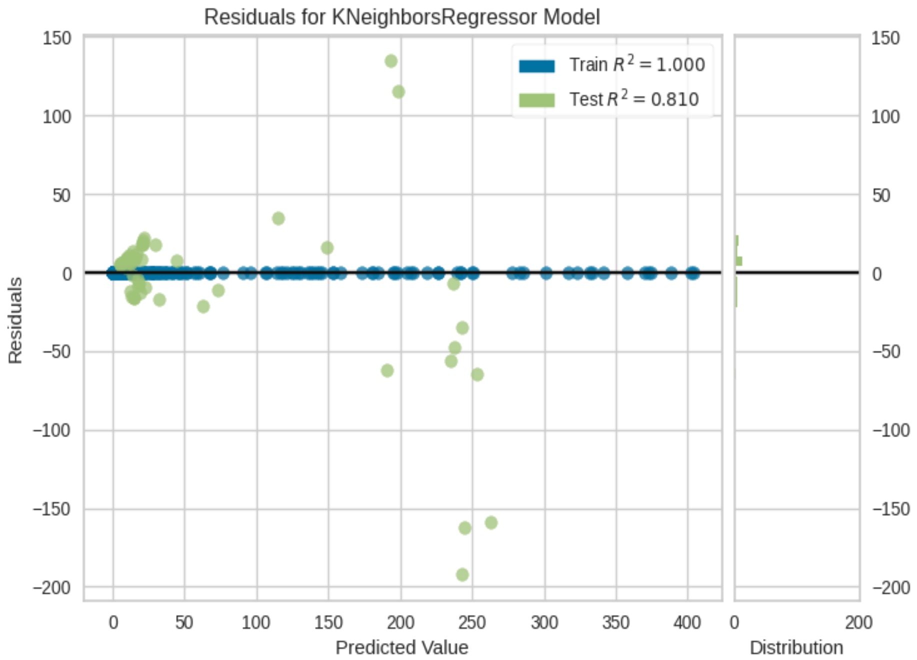

Fig. 6 Residuals curve for K-neighbors regression

To understand the plots, we'd need some context regarding what these plots indicate. So let us elaborate one by one.

A residuals plot is a graphical representation of the differences between the predicted values and the actual values (residuals) of the target variable in a regression model. It helps to assess the quality of the model's predictions by examining the distribution and patterns of the residuals. In a good model, the residuals should be randomly scattered around the horizontal zero line, indicating that the model captures the underlying patterns in the data, which is the case in our residuals plot for K-Neighbors regressor.

The learning curve plot shows the performance of the machine learning model in the training and validation sets as a function of the training data size. It helps to see how effective or stable a model's performance is by using more training data. The plot typically shows the training and validation errors (such as mean squared error or accuracy) on the y-axis and the size of the training data on the x-axis. This can help detect models' over-fitting (high training errors, low validation error) or whether there is under-fitting (high training and validation errors). Meeting the training sequence and cross-validation determines whether or not the model can benefit from additional training data. If the curves are consistent and flat, it would mean that the model does not benefit much from more training information. Our lines are not converging, suggesting that the model would have benefited from more training data. So, we can say that to increase the accuracy of the model, we can increase the size of the data set.

Error plots show model performance in terms of error (such as mean squared error) on the y-axis and model complexity on the x-axis. It helps to understand how model performance changes with different extreme parameter settings or model configurations. The plot usually shows a curve or line representing error values for different model settings or hyper-parameters. This can help achieve a desired configuration that reduces error and guides the selection of over-parameters. The objective is to identify the point at which the validation error is minimized while keeping the model complexity in check. This case represents a good balance between under-fitting (high training and validation errors) and over-fitting (low training errors, large validation errors), resulting in a model that performs well on unseen data

A validation curve plot showing model performance metrics (e.g., accuracy, mean squared error) on the y-axis and hyper-parameter values on the x-axis helps to examine how different values of a particular hyper-parameter affect model performance influence on The plot usually shows a curve or line representing the performance metric for various extreme parameter values. Identifying the point of efficiency can help determine the optimal value for the over-parameter.

IRJMETS
e-ISSN: 2582-5208
International Research Journal of Modernization in Engineering Technology and Science
(Peer-Reviewed, Open Access, Fully Refereed International Journal)
Volume:05/Issue:10/October-2023 Impact Factor- 7.868 www.irjmets.com

In the end, we determined relationships between the variables with the help of R^2 values and presented it in form of a heat map.

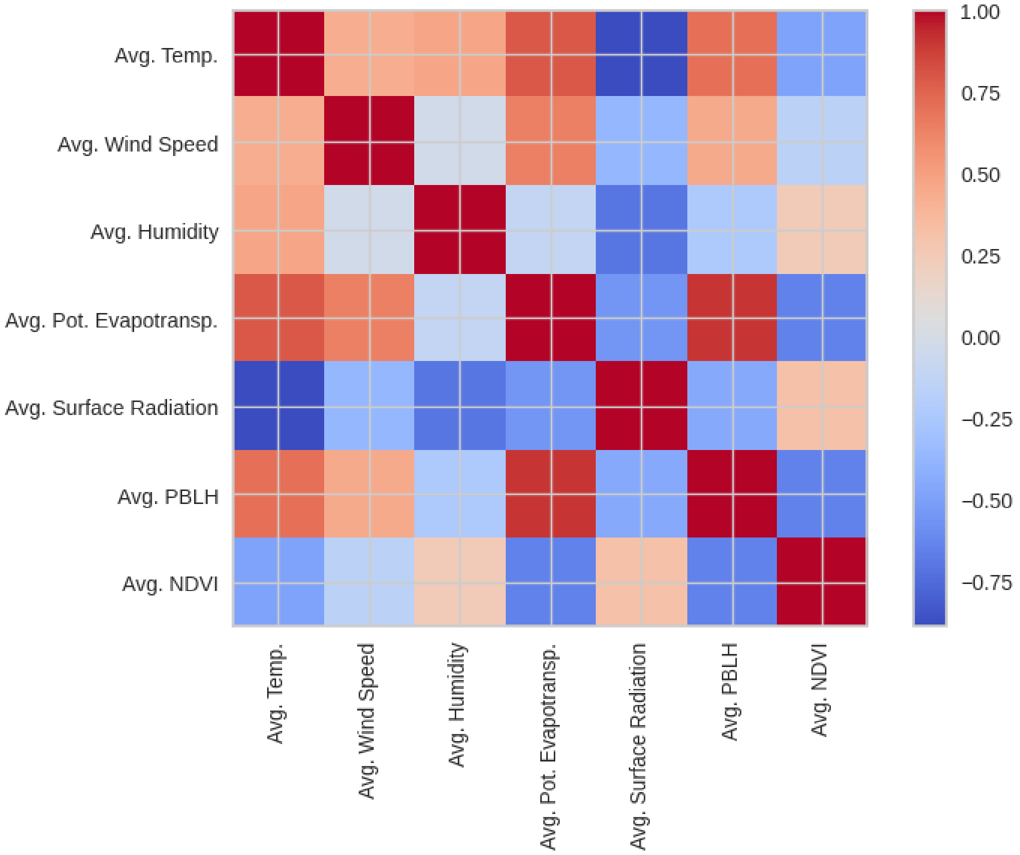

Fig. 7 Relationship between variables

IV. CONCLUSION

In this study, we explored the regression modelling techniques using the PyCaret framework to examine the dependency of precipitation on various meteorological variables in the Lucknow region. Our research aimed to identify the most suitable regression model for accurately predicting precipitation, an important component of regional climate assessment and water resource management.

Our findings demonstrated that while several regression models displayed strong predictive capabilities, the K-Neighbors Regression model consistently outperformed others, exhibiting the lowest values for key evaluation metrics such as R^2, RMSE and MAE. This model's superior performance signifies its potential for accurate precipitation predictions.

The visualizations, including line charts and correlation charts, offer valuable understanding of the impact of variables such as temperature, wind speed, humidity, evapo-transpiration, surface radiation, and planetary boundary layer height on precipitation patterns.

Our research underlines the importance of PyCaret as a powerful tool for regression analysis, enabling model selection and providing meaningful visualizations to enhance our understanding of complex meteorological relationships. This study serves as a foundation for further research into regional climate modeling and informs decision-makers in water resource management, agriculture, and disaster preparedness.

ACKNOWLEDGEMENTS

The authors are thankful to the team at Earth Sciences Data and Information Services Center, or GES DISC, Interactive Online Visualization and Analysis Infrastructure (GIOVANNI) NASA for providing the satellite data. The authors are also thankful to the Department of Civil Engineering, Institute of Engineering & Technology, Lucknow for supporting the work.

IRJMETS

e-ISSN: 2582-5208

International Research Journal of Modernization in Engineering Technology and Science

(Peer-Reviewed, Open Access, Fully Refereed International Journal)

Volume:05/Issue:10/October-2023 Impact Factor- 7.868 www.irjmets.com

Conflict of interest:

The authors declare no conflicts of interest regarding the publication of this paper.

Ethical Approval: The present research work does not contain any studies performed on animals or human subjects by any of the authors

V. REFERENCES

[1] A. Kusiak, X. Wei, A. P. Verma and E. Roz, "Modeling and Prediction of Rainfall Using Radar Reflectivity Data: A Data-Mining Approach," in IEEE Transactions on Geoscience and Remote Sensing, vol. 51, no. 4, pp. 2337-2342, April 2013. https://doi.org/10.1109/TGRS.2012.2210429.

[2] Chowdari K.K, Girisha R and K. C. Gouda, "A study of rainfall over India using data mining," 2015 International Conference on Emerging Research in Electronics, Computer Science and Technology (ICERECT), Mandya, India, 2015, pp. 44-47. https://doi.org/10.1109/ERECT.2015.7498985.

[3] Namitha K, Jayapriya A, SanthoshKumar G. Rainfall prediction using artificial neural network on map-reduce framework. ACM. 2015. https://doi.org/10.1145/2791405.2791468.

[4] Liyew, C.M., Melese, H.A. Machine learning techniques to predict daily rainfall amount. J Big Data 8, 153 (2021). https://doi.org/10.1186/s40537-021-00545-4

[5] Garg, Arnav & Pandey, Himanshu. (2019). Rainfall Prediction Using Machine Learning. doi:10.13140/RG.2.2.26691.04648. https://www.researchgate.net/publication/333223383_Rainfall_Prediction_Using_Machine_Learning

[6] S. Aswin, P. Geetha and R. Vinayakumar, "Deep Learning Models for the Prediction of Rainfall," 2018 International Conference on Communication and Signal Processing (ICCSP), Chennai, India, 2018, pp. 0657-0661, https://doi.org/10.1109/ICCSP.2018.8523829.

[7] C. Z. Basha, N. Bhavana, P. Bhavya and S. V, "Rainfall Prediction using Machine Learning & Deep Learning Techniques," 2020 International Conference on Electronics and Sustainable Communication Systems (ICESC), Coimbatore, India, 2020, pp. 92-97, doi: https://doi.org/10.1109/ICESC48915.2020.9155896

[8] Vijayan, R, V. Mareeswari, P MohanKumar and Gundu. Srikar. "Estimating Rainfall Prediction using Machine Learning Techniques on a Dataset." International Journal of Scientific & Technology Research 9 (2020): 440-445. https://api.semanticscholar.org/CorpusID:220279217

[9] Chaudhari MM, Choudhari DN. Study of various rainfall estimation & prediction techniques using data mining. Am J Eng Res. 2017;6(7):137–9. http://ajer.org/papers/v6(07)/Q0607137139.pdf

[10] C. Thirumalai, K. S. Harsha, M. L. Deepak and K. C. Krishna, "Heuristic prediction of rainfall using machine learning techniques," 2017 International Conference on Trends in Electronics and Informatics (ICEI), Tirunelveli, India, 2017, pp. 1114-1117, doi: https://doi.org/10.1109/ICOEI.2017.8300884

[11] N. Gnanasankaran, E. Ramaraj. (2020). A Multiple Linear Regression Model To Predict Rainfall Using Indian Meteorological Data. International Journal of Advanced Science and Technology, 29(8s), 746 - 758. Retrieved from http://sersc.org/journals/index.php/IJAST/article/view/10816

[12] I. R. P, J. Anitha, J. Brema, D. S. Juliet and S. S, "Modeling Precipitation Patterns in Alappuzha -Kerala: An Analysis of Regression and Time Series Approaches," 2023 3rd International Conference on Intelligent Technologies (CONIT), Hubli, India, 2023, pp. 1-6, doi: https://doi.org/10.1109/CONIT59222.2023.10205530.

[13] Ahmed NK, Atiya AF, Gayar NE, El-Shishiny H. An Empirical Comparison of Machine Learning Models for Time Series Forecasting. Econometric Reviews. 2010;29(5-6):594–621. https://doi.org/10.1080/07474938.2010.481556

[14] Nyasulu, C., Diattara, A., Traore, A., Deme, A., Ba, C. (2023). Exploring Use of Machine Learning Regressors for Daily Rainfall Prediction in the Sahel Region: A Case Study of Matam, Senegal. In: Ngatched Nkouatchah, T.M., Woungang, I., Tapamo, JR., Viriri, S. (eds) Pan-African Artificial Intelligence and Smart Systems. PAAISS 2022. Lecture Notes of the Institute for Computer Sciences, Social Informatics and Telecommunications Engineering, vol 459. Springer, Cham. https://doi.org/10.1007/978-3-031-25271-6_5