A Study on the performance of Four Regression Models in Predicting Weather Temperature Based on Python

Taobei Li

Faculty of Science and Engineering，University of Nottingham Ningbo, Ningbo, China

Abstract. For industries like agriculture and disaster management, weather forecasting is essential. This study assesses how well four regression models—linear regression, random forest regression, support vector regression (SVR), and K-Nearest Neighbors (KNN)—predict weather temperatures using a dataset from England. Standardizing and expanding features were part of the data preprocessing process to capture non-linear interactions. Performance metrics were used to evaluate the models' predictive capacity. With the highest R2 value and the lowest error metrics, Random Forest Regression fared better than the other models, suggesting higher predictive accuracy, according to the data. KNN exhibited greater sensitivity to local fluctuations compared to SVR, which performed slightly better overall. linear Regression was the least effective, struggling with non-linear data and exhibiting higher error metrics. This study offers a thorough comparison of weather prediction regression models, emphasizing the performance of the Random Forest regression.

1 Introduction

Weather prediction significantly benefits society by aiding agricultural planning and providing warnings for extreme weather. Because of improved observations, models, data assimilation, and associated techniques that better integrate these, weather prediction has rapidly increased in recent decades[1]. Machine learning is widely used in many domains to tackle difficult problems that are difficult for computer-based techniques to solve[2]. And it is commonly used in weather prediction. Sharapov examined the use of linear regression models in weather observation in recent research conducted over the previous five years using a dataset provided by weather data observations in Nizhny Novgorod. According to the research, precise weather forecasting may be accomplished by utilizing a little quantity of data from the previous day as opposed to depending on a big number of features over a long period of time[3]. Shivam and associates also employed random forest models to forecast weather in order to regulate a variety of activities that are weather-dependent either directly or indirectly[4]. In addition, Taneja et al introduces a novel weather prediction algorithm, finding that Artificial Neural Networks outperform Linear Regression in accuracy[5]. These

Corresponding author: scytl7@nottingham.edu.cn

studies typically focus on a single machine learning model for weather prediction without providing sufficient justification or supporting data for their model choices. This study compares the outcomes of data visualization and the accuracy of various models when presented with the same dataset, focusing on four popular machine learning models for weather prediction. Linear Regression, Random Forest Regression, Support Vector Regression, and K-Nearest Neighbors and compare their respective performances. By doing this study, a more comprehensive evaluation of models' performance can be provided to weather prediction field.

2 Methodology

2.1 Preparation of the dataset

A CSV file containing meteorological data for England from January 1, 2006, to December 31, 2016 was downloaded from Kaggle website [6]. The dataset includes columns such as Formatted Date, Summary, Precip Type, Temperature (C), Pressure (millibars), and Humidity. For example, the first few rows of the dataset are as shown in Figure 1.

EnglandWeather

Formatted Date	Summary	Precip Type	Temperature (C)	Wind Speed (km/h)	Pressure (millibars)	Humidity
2006-01-01 00:00:00.000 +0100	Partly Cloudy	rain	0.577777778	17.1143	1016.66	0.89
2006-01-01 01:00:00.000 +0100	Mostly Cloudy	rain	1.161111111	16.6152	1016.15	0.85
2006-01-01 02:00:00.000 +0100	Mostly Cloudy	rain	1.666666667	20.2538	1015.87	0.82
2006-01-01 03:00:00.000 +0100	Overcast	rain	1.711111111	14.49	1015.56	0.82
2006-01-01 04:00:00.000 +0100	Mostly Cloudy	rain	1.183333333	13.9426	1014.98	0.86
2006-01-01 05:00:00.000 +0100	Mostly Cloudy	rain	1.205555556	15.9068	1014.08	0.85
2006-01-01 06:00:00.000 +0100	Mostly Cloudy	rain	2.222222222	20.5597	1013.93	0.79
2006-01-01 07:00:00.000 +0100	Mostly Cloudy	rain	2.072222222	25.4219	1013.24	0.81
2006-01-01 08:00:00.000 +0100	Mostly Cloudy	rain	2.2	23.1679	1013.19	0.8
2006-01-01 09:00:00.000 +0100	Mostly Cloudy	rain	2.733333333	22.8781	1013.29	0.82

Fig. 1. Dataset of England Weather (2006-2016)

2.2 Data preprocessing

Since the target prediction data for temperature is numerical, non-numeric columns were removed to ensure consistency and simplicity. After that, this study took the cleaned data and retrieved the temperature goal variables and feature variables. Ultimately, standardization was used in this work to transform the feature data into a standard normal distribution with a variance of 1 and a mean of 0, which accelerated the convergence and stability of the model training. This investigation further produced quadratic polynomial features, which broadened the feature space and assisted the model in better capturing the nonlinear interactions between the characteristics. To better understand the temperature data in this dataset, this study created a scatter plot using Python. As seen in Figure 2, there was a noticeable increase in temperature between January 1 and January 3, rising from roughly 1 degree to over 7 degrees. Between 01 and 03, the temperature decreases, and between 01 and 04, it stabilizes at 2 to 4 degrees. Between January 01 and January 02, the temperature rises significantly and shows a non-linear rapid ascent. The temperature rapidly drops from 01-02 to 01-03, showing a non-linear falling pattern. There are small and relatively constant temperature fluctuations from January

to March, with no obvious linear trend toward rising or falling temperatures. On 01–02, there is a great deal of temperature variation, with certain locations being substantially warmer than others. Temperature swings are minimal after 01-04, with most data points falling between 2 and 4 degrees. In conclusion, the temperature data in this dataset seems to show non-linear variations based on the details.

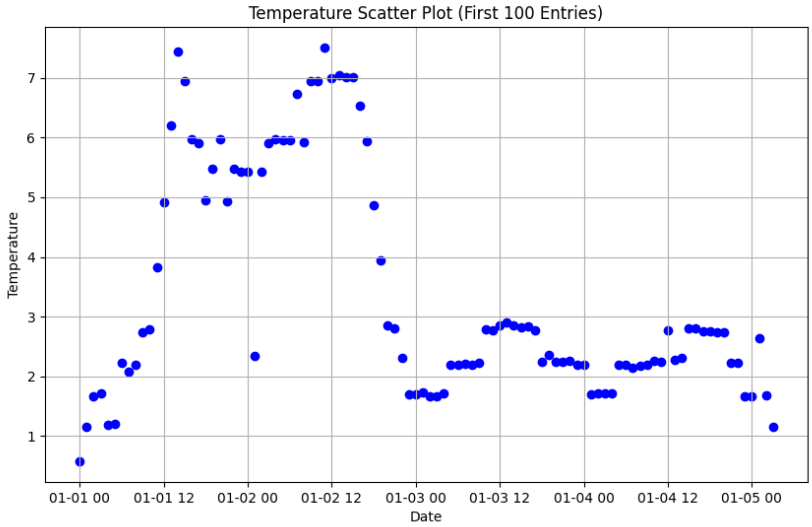

Fig. 2. Temperature Scatter Plot (First 100)

2.3 Linear regression (LR)

Machine learning is widely used in many domains to tackle difficult problems that are difficult for computer-based techniques to handle. Linear regression is among the most widely used and fundamental machine learning algorithms[1]. Sharapov focuses on the use of linear regression in weather prediction. A test data set was created using Nizhny Novgorod weather observations, and the features that had the biggest influence on the predicted value were determined. These characteristics include the preceding several days' highest, lowest, and average temperature as well as the maximum, minimum, and average dew point values[3]. Despite the non-linear properties of the temperature data gathered, we nevertheless developed a linear regression model for comparison and prediction, given the frequent use of linear regression models in machine learning and prediction. And it should be noted that we were able to express non-linear relationships in the model by extending the features to include quadratic polynomial terms. When it is possible to quantify and model the prediction outcomes in connection input variables, linear regression is a frequently employed technique in predictive analysis[3]. It is a method for determining linear correlations between dependent and independent variables via modeling and data analysis[3]. Given this, by analyzing and learning from the existing training data, this technique may anticipate correlations between independent and dependent variables[3]. The following equation serves as the foundation for the linear regression model[7]:

$$\hat{y} = \beta_0 + \beta_1 \cdot x_1 + \beta_2 \cdot x_2 + \cdots + \beta_{p-n} \cdot x_{p-n} + \theta \quad (1)$$

A free machine learning toolkit for Python called Scikit-learn includes features including clustering algorithms, regression, and classification in addition to a number of interoperable Python methods[8]. This project used the Scikit-learn library's built-in linear regression function to create the linear regression model.

2.4 Random Forest Regression (RF)

When developing weather prediction models, researchers frequently favor random forest models over linear regression models due to the nonlinear characteristics displayed by weather data. The well-known random forest model in machine learning is made up of decision trees and the ensemble learning approach[9]. The Random Forest Algorithm's main advantage is its applicability to both regression and classification. It is based on supervised learning[4]. Numerous decision trees (training samples) that are built randomly or in an ensemble make up a Random Forest. Different bootstrap datasets that are obtained from the original source data are used to create these decision trees[4]. When a test dataset is applied to each decision tree, it yields a class prediction; the class with the highest number of votes is chosen as the model class.

After using the bagging approach to create a collection of trained classifiers, this study used voting to classify newly added data points. Voting might be based on the classifiers' decisions or predictions being weighted differently[4]. By selecting and voting at random, the random forest model may completely cover the trained set, resulting in robust decision trees that manage correlations. The random forest classifier is then validated or calibrated using the test set.

This study established the random forest regression model with the training data using the Python language and the scikit-learn package. There are four modifiable variables that impact modeling and need to be set beforehand. The number of trees to be constructed is specified by n_estimators. While more trees typically result in better model performance, they also add to the cost and duration of training. In the code, n_estimators are set to [50, 100], indicating that this study will attempt to train the model using 50 and 100 trees in order to properly strike a balance between computational economy and model performance. The code sets min_samples_split to [2, 5], which indicates that this study will attempt to split nodes using minimum sample sizes of 2 and 5 in order to avoid overfitting. Furthermore, this study tries to employ minimum sample sizes of 1 and 2 in leaf nodes in the hopes of guaranteeing that each leaf node has an adequate number of samples, thereby lowering the variance of the model.

2.5 Support Vector Regression (SVR)

In addition to linear regression and random forest algorithms, support vector regression is also a commonly used machine learning algorithm in the field of weather forecasting. Recently, a study that analyzes synoptic data such as wind speed, temperature, and humidity to predict rainfall used ANFIS and Support Vector Regression (SVR)[10]. By obtaining accurate forecasts, communities will be able to plan ahead for weather anomalies thanks to this technique. The MSE of 0.0928 for the SVR model is promising[10].

When there is a distinct difference between the classes, the SVM method performs very well, exhibiting increased efficiency in higher-dimensional spaces and the capacity to manage scenarios with more dimensions than sample values[11]. It is a good option for many applications because it is also quite memory efficient[11]. SVM does not perform well on huge datasets, though, and frequently underperforms when there is a lot of noise present, like target class overlap, which occurs in real-world scenarios[11]. It also encounters difficulties when the feature values for each data point have larger numerical values than the training

samples, and it does not provide a probabilistic justification for the classifications it generates[11].

The scikit-learn library offers a Support Vector Regression (SVR) class. This study uses this class to create an SVR model. This study uses this class to build an SVR model. What's more, this study specifies the use of the Radial Basis Function (RBF) kernel in the actual construction. This is because the RBF kernel works well in the majority of application cases, is especially good at handling non-linear data, and doesn't require the user to make a lot of assumptions about the data beforehand. As Data preprocessing part has mentioned, the characteristic of data this study select is non-linear, so it is suitable to use RBF kernel.

2.6 K-Nearest Neighbors (KNN)

The K-Nearest Neighbors (KNN) model is another machine learning model that may be used for weather prediction in addition to the three previously mentioned machine learning approaches.

A simple supervised machine learning method that works well for both regression and classification issues is the KNN algorithm. Because the KNN approach does not necessitate an understanding of the relationship between the characteristics and outcomes, it is more effective than parametric time series models[12]. Furthermore, the K-closest Neighbors (KNN) regression model understandably calculates the value of a new observation by using the K closest data points (most comparable in input characteristics) in the training dataset[14]. KNN regression is widely used in a wide range of applications, including as stock price prediction, house price computation, and weather pattern predictions.

The separation between each fresh observation and each observation in the training set is computed by the KNN regression model. The most often used distance metric is the Euclidean distance, which may be calculated using the method shown in equation (2).

$$d(xi, xj) = \sqrt{\sum_{k=1}^{p} \left(x_{ik} - x_{jk}\right)^2} \tag{2}$$

In this instance, p represents how many features are input, x_{ik} represents the kth input feature value for the i^{th} observation, and x_{jk} represents the kth feature value for the j^{th} observation[13].

The KNN technique determines which K neighbors have the shortest distances after calculating the distances. The target variable values of these K nearest neighbors are then averaged (or medianted) to get the predicted value of the new observation[14].

For a variety of issues, the K-Nearest Neighbors (K-NN) algorithm is a basic and user-friendly approach. It is robust in noisy situations because it exhibits a notable tolerance and resilience to noise within the training data. Moreover, K-NN is quick and simple to understand, and it keeps working well even with very huge datasets. Choosing a suitable number for K, however, can be difficult because it has a big impact on the outcome. Since the method involves figuring out the Euclidean distance between every data point in the training dataset, the computation cost may also be substantial.

In this project, this study uses the built-in KNeighborsRegressor class from scikit-learn to build the model, which has three key parameters:

1: 'n_neighbors': This is a core parameter in the KNN model that specifies how many neighbors there are used for prediction. Common choices include 3, 5, 7, 9, and 11, which will be tested one by one during model training to determine the best K value. It is important to remember that the model's accuracy is impacted by the K value selection. Finding the best K value for precise predictions requires testing a variety of values.

2: 'weight's: This parameter determines the weighting strategy for neighbors. This study can choose 'uniform', which means all neighbors have equal weight, or 'distance', It indicates

that each neighbor's weight is inversely related to its distance, with neighbors that are closer together having heavier weights.

3:'Algorithm': This parameter specifies the algorithm used for computing neighbors. The options this study choose are 'auto', 'kd_tree', 'ball_tree' and 'brute'.

In conclusion, by designing a parameter grid, this study use GridSearchCV to improve the hyperparameters of the KNN regression model. This study examines various K values, weighting schemes, and search algorithms in an attempt to determine the optimal model settings that will maximize the model's performance.

3 Result and Discussion

3.1 Visualization

To give a simple but effective understand of the model's prediction outcomes, this study presented the comparison between the model's predicted values and the true values after the model was constructed (as shown in Figures 3-6).

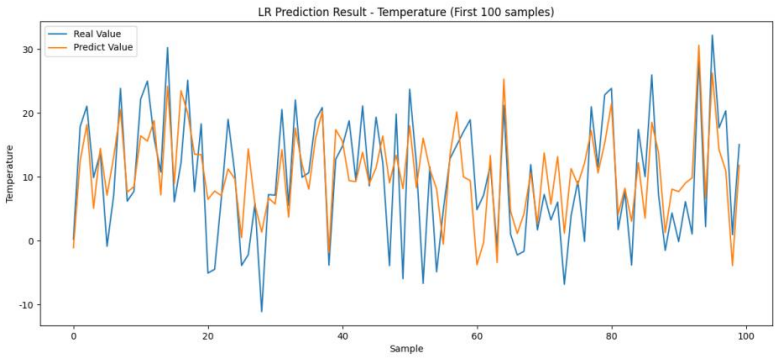

Fig. 3. linear Regression

The real values (blue line) in Figure 3, which shows the linear regression model, illustrate notable volatility with large temperature changes. In contrast, the predicted values (orange line) show observable differences from the real values but often follow the same pattern, particularly at locations where temperature fluctuates quickly. The linear regression model fails to adequately depict the real temperature variations because it has trouble capturing some peaks and valleys. This is due to the fact that when working with complicated, nonlinear data, linear regression frequently produces poor outcomes since it expects a linear relationship between variables. The linear regression model can nevertheless fairly represent the general trend of temperature variations in spite of these biases. Its capacity to manage extremely erratic and quickly changing data is constrained.

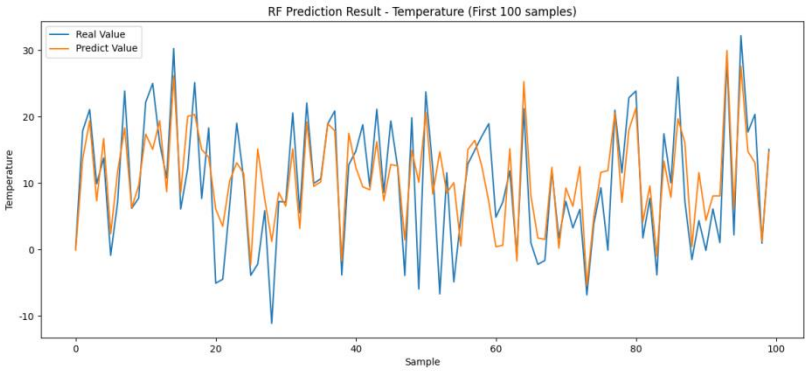

Fig. 4. Random Forest

In comparison to the linear regression model, the random forest model's projected values (orange line) in figure 4 better depict temperature variations, especially during periods of fast temperature increases and declines. The random forest model performs more well at managing complex nonlinear data, as seen by the projected values being closer to the true values. Even with a few outliers here and there, the overall performance beats the linear regression model. Overfitting, however, is a possibility because of the random forest model's intricacy. This means that while the model may perform remarkably well on training data, it may not perform well on test or fresh data. Even though it might not be immediately apparent from the figure, this is a crucial factor to thoroughly analyze and examine when it comes to the model assessment stage.

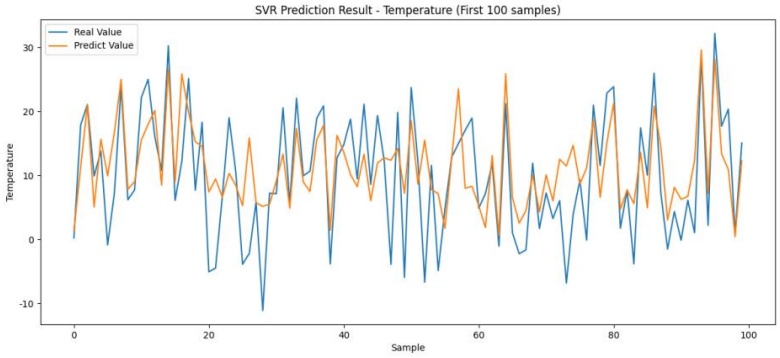

Fig. 5. Support Vector Regression

Figure 5, which displays the actual temperature values (blue line) and the temperature values predicted by the SVR model (orange line) for the first 100 samples, illustrates the SVR prediction results. The real values (blue line) show notable temperature variations between sample locations. With the fluctuation trends of the predictions matching the real values for most sample points, the SVR model's anticipated values (orange line) closely match the general trend of the actual data. According to error analysis, the SVR model performs well in terms of prediction because its forecasts are generally quite near to the actual values for the majority of sample points. The discrepancies between the actual and anticipated values at various sample points (such as approximately points 10, 40, 70, and 90) are present, however they are not very significant. All things considered, the SVR model successfully

tracks the variations of the real values and catches the temperature change trends, especially in regions with rapid temperature change.

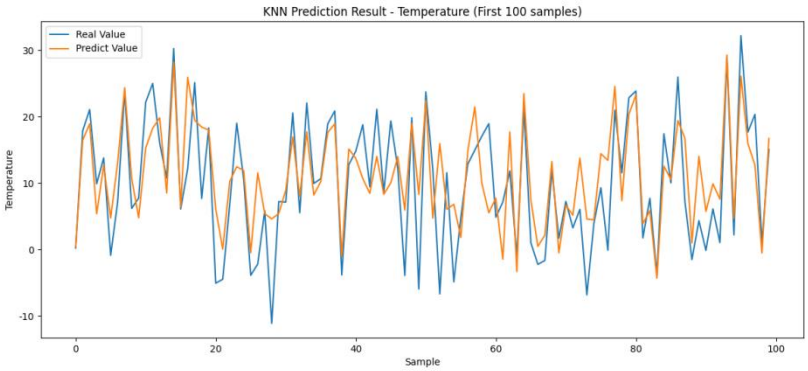

Fig. 6. K-Nearest neighbors

For the first 100 samples, the temperature values predicted by the KNN model (orange line) and the actual temperature values (blue line) are shown in the figure 6. The orange line displays the predicted values of the KNN model, which largely match the overall trend of the actual values but show notable variances in some places. The blue line depicts significant temperature changes between sample points. Error analysis shows that the KNN model performs well in terms of prediction at specific sample points (such as approximately 10, 20, 50, and 80), where the projected values are relatively close to the actual values. Nevertheless, there are greater differences between the expected and actual values at other sample points (such as approximately 30, 60, and 90), which suggests worse predictive performance at these points. Because the KNN model is more sensitive to swings in local data, it exhibits larger errors overall in places with more abrupt temperature changes.

3.2 Model Performance Evaluation

To offer a thorough assessment of the four models' contributions to this project, this study documented the Mean Squared Error (MSE), Root Mean Squared Error (RMSE), Mean Absolute Error (MAE), and R-squared (R^2) for every model. The result is showed as Table 1:

Table 1. Result of MSE, RMSE, MAE, and R^2

	MSE	**RMSE**	**MAE**	**R^2**
Linear regression model	43.499	6.595	5.335	0.522
Random forest regression	35.909	5.992	4.766	0.606
Support Vector Regression	45.599	6.752	5.418	0.499
K-Nearest neighbors	39.657	6.297	4.955	0.564

The model that performed the best was the random forest regression model, which showed the lowest values of MSE, RMSE, and MAE in addition to the highest value of R^2, indicating superior performance in terms of explaining data variance and predictive accuracy. Although its MSE and RMSE were slightly higher but still suggestive of good overall

performance, the KNN regression model trailed the random forest model closely, with strong R2 and MAE metrics right behind it. While outperforming the SVR model, the random forest and KNN models outperformed the linear regression model. Its R2 score was 0.523, slightly higher than the SVR model but lower than the random forest and KNN models, with error metrics (MSE, RMSE, and MAE) lying between those of the SVR and KNN models. The SVR regression model outperformed the other models in terms of both prediction accuracy and data variance explanation, as evidenced by its lowest R^2 value and highest error metrics (MSE, RMSE, and MAE).

4 Conclusion

In order to predict weather temperatures, this study examines and contrasts the accuracy of four regression models. We assessed these models' applicability and predictive accuracy using a UK meteorological dataset. Furthermore, this study's contribution is a comprehensive comparison of regression models' accuracy in weather temperature prediction. This not only provides useful references for further study and applications, but it also demonstrates the advantages and disadvantages of various models for processing meteorological data.

There are still limits even though this study offers a thorough assessment of the four regression models' performances. First, the dataset only includes meteorological information from the United Kingdom, which might not be universally applicable to other areas or climate conditions. Second, other factors like wind speed and atmospheric pressure that might have an impact on weather forecasts are not taken into account by the models used in this study. Even though performance optimization was the goal of the model tuning process, there might be more space for advancement.

Future studies might concentrate on a number of crucial aspects to enhance the model's flexibility and generalization: combining data from different regions and climate conditions; adding more meteorological variables to improve comprehensive prediction and comprehend The impact of multiple factors on forecasting; examining the effectiveness of deep learning models, like neural networks, to reveal complex nonlinear relationships; and figuring out how to use these models for real-time weather forecasting while dynamically modifying them to account for constantly changing climate conditions.

References

1. Alley, R. B., Emanuel, K. A., & Zhang, F. Advances in weather prediction. Science, 363(6425), 342–344 (2019) https://doi.org/10.1126/science.aav7274
2. Maulud, D., & Abdulazeez, A. M. A Review on Linear Regression Comprehensive in Machine Learning. Journal of Applied Science and Technology Trends, 1(4), 140–147 (2020) https://doi.org/10.38094/jastt1457
3. Using Linear Regression for Weather Prediction. IEEE Conference Publication | IEEE Xplore (2022) https://ieeexplore.ieee.org/abstract/document/9803493/authors#authors
4. S. Mishra, A. Shukla, S. Arora, H. Kathuria and M. Singh, Controlling Weather Dependent Tasks Using Random Forest Algorithm, 2020 Third International Conference on Advances in Electronics, Computers and Communications (ICAECC), Bengaluru, India, pp. 1-8 (2020) doi: 10.1109/ICAECC50550.2020.9339508.
5. Taneja, H., Jain, V. J. A., Dubey, A. K., Taplamacioglu, M. C., & Demirci, M. WEATHER PREDICTION USING REGRESSION ALGORITHM AND NEURAL NETWORK TECHNIQUE. International Journal on Technical and Physical Problems of Engineering, 16(59) (2024)

6. EnglandWeather. (2022, August 15). Kaggle. https://www.kaggle.com/datasets/zohrehtofighizavareh/englandweather?resource=download
7. H.-I. Lim, A Linear Regression Approach to Modeling Software Characteristics for Classifying Similar Software, in 2019 IEEE 43rd Annual Computer Software and Applications Conference (COMPSAC), pp. 942-943 (2019)
8. A. McQuistan, Using Machine Learning to Predict the Weather: Part 2 (2024) https://stackabuse.com/using-machine-learning-to-predict-the- weather-part-2/
9. M. Suresh Kumar, V. Soundarya, S. Kavitha, E.S. Keerthika, E. Aswini "Credit Card Fraud Detection using Random Forest Algorithm," 2019 3rd International Conference on Computing and Communication Technologies (ICCCT) , pp. 149-153 (2019)
10. Novitasari, D. C. R., Rohayani, H., Junaidi, R., Setyowati, R. D., Pramulya, R., & Setiawan, F. Weather parameters forecasting as variables for rainfall prediction using adaptive neuro fuzzy inference system (ANFIS) and support vector regression (SVR). In Journal of Physics: Conference Series (Vol. 1501, No. 1, p. 012012). IOP Publishing (2020)
11. M. Bansal, S. Malik, M. Kumar, N. Meena, Arduino based smart walking cane for visually impaired people, in: 2020 Fourth International Conference on Inventive Systems and Control (ICISC), pp. 462–465 (2020) http://dx.doi.org/10.1109/ ICISC47916.2020.9171209.
12. Faysal, I. SHORT TERM TRAFFIC FLOW PREDICTION USING MACHINE LEARNING - KNN, SVM AND ANN WITH WEATHER INFORMATION. International Journal for Traffic and Transport Engineering, 10(3), 371–389. (2020) https://doi.org/10.7708/ijtte.2020.10(3).08
13. Sumayli, A. Development of advanced machine learning models for optimization of methyl ester biofuel production from papaya oil: Gaussian process regression (GPR), multilayer perceptron (MLP), and K-nearest neighbor (KNN) regression models. Arabian Journal of Chemistry, 16(7), 104833 (2023) https://doi.org/10.1016/j.arabjc.2023.104833
14. Bansal, M., Goyal, A., & Choudhary, A. A comparative analysis of K-Nearest Neighbor, Genetic, Support Vector Machine, Decision Tree, and Long Short Term Memory algorithms in machine learning. Decision Analytics Journal, 3, 100071 (2022) https://doi.org/10.1016/j.dajour.2022.100071