A Machine Learning project Report on

Rain Prediction System

SUBMITTED BY

Suprit Giri – 20IT5002
Shrishailendra Patil -19IT1104
Prathmesh Gaikwad -19IT1047

Under the guidance of

Mrs. NILIMA DONGRE

Department of
Information Technology
Dr. D. Y. Patil Group's
Ramrao Adik Institute of Technology
Nerul, Navi Mumbai
(Affiliated to University of Mumbai**)** (2022)

CERTIFICATE

This is to certify that the project entitled **'Rain Prediction System'** being submitted by **Suprit Giri (20IT5002), Prathmesh Gaikwad (19IT1047), Shrishailendra Patil (19IT1104)** to the University of Mumbai in partial fulfillment of the requirement for the award of the degree of 'T. E. I. T' in "ML Mini Project".

Project Guide	**External Examiner**	**Head of Department**
(Mrs. Nilima Dongre)		(Dr. Ashish Jadhav)

DECLARATION

We declare that this written submission represents our ideas in our own words and where others' ideas or words have been included, we have adequately cited and referenced the original sources. We also declare that we have adhered to all principles of academic honesty and integrity and have not misrepresented or fabricated or falsified any idea/data/fact/source in our submission. We understand that any violation of the above will be cause for disciplinary action by the Institute and can also evoke penal action from the sources which have thus not been properly cited or from whom proper permission has not been taken when needed.

Students Name	Roll No.	Signature
1. Suprit Giri	**20IT5002**	**( )**
2. Prathmesh Gaikwad	**19IT1047**	**( )**
3. Shrishailendra Patil	**19IT1104**	**( )**

Date:
Place:

TABLE OF CONTENTS

Introduction:

In today's situation, rainfall is considered to be one of the sole responsible factors for most of the significant things across the world. In India, agriculture is considered to be one of the important factors for deciding the economy of the country and agriculture is solely dependent on rainfall. Apart From that in the coastal areas across the world, getting to know the amount of rainfall is very much necessary.In some of the areas which have water scarcity, to establish rain water harvester, prior prediction of the rainfall should be done. This project deals with the prediction of rainfall using machine learning. The project performs the comparative study of machine learning approach then accordingly portrays the efficient approach for rainfall prediction. First of all, preprocess is performed. Preprocess is the process of representing the dataset in the form of several graphs such as bar graph, histogram etc.

The accurate and precise rainfall prediction is still lacking due to imprecise and inaccurate calculations for the rainfall prediction. Therefore, the need is not to formulate only the rainfall predicting system but also a system that is more accurate and precise as compared to the existing rainfall predictors which could assist in diverse fields like agriculture, water reservation and flood prediction. So, the system which we are proposing will be an end-to-end deployment project (Machine Learning app) where we will create a distinctive and efficient machine learning system for the prediction of rainfall.

Currently, rainfall prediction has become one of the key factors for most of the water conservation systems in and across country. One of the biggest challenges is the complexity present in rainfall data. Most of the rainfall prediction system, nowadays are unable to find the hidden layers or any non-linear patterns present in the system. This project will assist to find all the hidden layers as well as non-linear patterns, which is useful for performing the precise prediction of rainfall. Due to presence of the system which doesn't find the hidden layers and nonlinear patterns accurately, the prediction results to be wrong for most of the times and that may lead to huge losses. So, the main objective for this research work is to find a system that can resolve this issue which will give proper and accurate prediction thereby assisting the country to develop when it comes to agriculture and economy.

Objective:

The main aim of this project is to detect whether it will rain or not on the next day with the help of using weather details that would be based on the previous findings and similarities and will give the output predictions that are reliable and appropriate.

.

Problem statement:

The accurate and precise rainfall prediction is still lacking which could assist in diverse fields like agriculture, water reservation and flood prediction. The issue is to formulate the calculations for the rainfall prediction that would be based on the previous findings and similarities and will give the output predictions that are reliable and appropriate. The imprecise and inaccurate predictions are not only the waste of time but also the loss of resources and lead to inefficient management of crisis like poor agriculture, poor water reserves and poor management of floods. Therefore, the need is not to formulate only the rainfall predicting system but also a system that is more accurate and precise as compared to the existing rainfall predictors.

Literature Survey:

Rainfall prediction is not an easy job especially when expecting the accurate and precise digits for predicting the rain. The rainfall prediction is commonly used to protect the agriculture and production of seasonal fruits and vegetables and to sustain their production and quality in relation to the amount of rain required by them (Lima & Guedes, 2015). The rainfall prediction uses several networks and algorithms and obtains the data to be given to the agriculture and production departments.

Dataset :

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X
1	Date	Location	MinTemp	MaxTem	Rainfall	Evaporat	Sunshine	WindGus	WindGus	WindDir	WindDir	WindSpe	WindSpe	Humidity	Humidity	Pressure	Pressure	Cloud9ar	Cloud3pr	Temp9ar	Temp3pr	RainTod	RainTomorrow	
2	#######	Albury	13.4	22.9	0.6	NA	NA	W	44	W	WNW	20	24	71	22	1007.7	1007.1	8	NA	16.9	21.8	No	No	
3	#######	Albury	7.4	25.1	0	NA	NA	WNW	44	NNW	WSW	4	22	44	25	1010.6	1007.8	NA	NA	17.2	24.3	No	No	
4	#######	Albury	12.9	25.7	0	NA	NA	WSW	46	W	WSW	19	26	38	30	1007.6	1008.7	NA	2	21	23.2	No	No	
5	#######	Albury	9.2	28	0	NA	NA	NE	24	SE	E	11	9	45	16	1017.6	1012.8	NA	NA	18.1	26.5	No	No	
6	#######	Albury	17.5	32.3	1	NA	NA	W	41	ENE	NW	7	20	82	33	1010.8	1006	7	8	17.8	29.7	No	No	
7	#######	Albury	14.6	29.7	0.2	NA	NA	WNW	56	W	W	19	24	55	23	1009.2	1005.4	NA	NA	20.6	28.9	No	No	
8	#######	Albury	14.3	25	0	NA	NA	W	50	SW	W	20	24	49	19	1009.6	1008.2	1	NA	18.1	24.6	No	No	
9	#######	Albury	7.7	26.7	0	NA	NA	W	35	SSE	W	6	17	48	19	1013.4	1010.1	NA	NA	16.3	25.5	No	No	
10	#######	Albury	9.7	31.9	0	NA	NA	NNW	80	SE	NW	7	28	42	9	1008.9	1003.6	NA	NA	18.3	30.2	No	Yes	
11	#######	Albury	13.1	30.1	1.4	NA	NA	W	28	S	SSE	15	11	58	27	1007	1005.7	NA	NA	20.1	28.2	Yes	No	
12	#######	Albury	13.4	30.4	0	NA	NA	N	30	SSE	ESE	17	6	48	22	1011.8	1008.7	NA	NA	20.4	28.8	No	Yes	
13	#######	Albury	15.9	21.7	2.2	NA	NA	NNE	31	NE	ENE	15	13	89	91	1010.5	1004.2	8	8	15.9	17	Yes	Yes	
14	#######	Albury	15.9	18.6	15.6	NA	NA	W	61	NNW	NNW	28	28	76	93	994.3	993	8	8	17.4	15.8	Yes	Yes	
15	#######	Albury	12.6	21	3.6	NA	NA	SW	44	W	SSW	24	20	65	43	1001.2	1001.8	NA	7	15.8	19.8	Yes	No	
16	#######	Albury	8.4	24.6	0	NA	NA	NA	NA	S	WNW	4	30	57	32	1009.7	1008.7	NA	NA	15.9	23.5	No	NA	
17	#######	Albury	9.8	27.7	NA	NA	NA	WNW	50	NA	WNW	NA	22	50	28	1013.4	1010.3	0	NA	17.3	26.2	NA	No	
18	#######	Albury	14.1	20.9	0	NA	NA	ENE	22	SSW	E	11	9	69	82	1012.2	1010.4	8	1	17.2	18.1	No	Yes	
19	#######	Albury	13.5	22.9	16.8	NA	NA	W	63	N	WNW	6	20	80	65	1005.8	1002.2	8	1	18	21.5	Yes	Yes	
20	#######	Albury	11.2	22.5	10.6	NA	NA	SSE	43	WSW	SW	24	17	47	32	1009.4	1009.7	NA	2	15.5	21	Yes	No	
21	#######	Albury	9.8	25.6	0	NA	NA	SSE	26	SE	NNW	17	6	45	26	1019.2	1017.1	NA	NA	15.8	23.2	No	No	
22	#######	Albury	11.5	29.3	0	NA	NA	S	24	SE	SE	9	9	56	28	1019.3	1014.8	NA	NA	19.1	27.3	No	No	
23	#######	Albury	17.1	33	0	NA	NA	NE	43	NE	N	17	22	38	28	1013.6	1008.1	NA	1	24.5	31.6	No	No	
24	#######	Albury	20.5	31.8	0	NA	NA	WNW	41	W	W	19	20	54	24	1007.8	1005.7	NA	NA	23.8	30.8	No	No	
25	#######	Albury	15.3	30.9	0	NA	NA	N	33	ESE	NW	6	13	55	23	1011	1008.2	5	NA	20.9	29	No	No	
26	#######	Albury	12.6	32.4	0	NA	NA	W	43	E	W	4	19	49	17	1012.9	1010.1	NA	NA	21.5	31.2	No	No	
27	#######	Albury	16.2	33.9	0	NA	NA	WSW	35	SE	WSW	9	13	45	19	1010.9	1007.6	NA	1	23.2	33	No	No	
28	#######	Albury	16.9	33	0	NA	NA	WSW	57	NA	W	0	26	41	28	1006.8	1003.6	NA	1	26.6	31.2	No	No	

```
In [3]: df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 145460 entries, 0 to 145459
Data columns (total 23 columns):
 #   Column         Non-Null Count   Dtype  
---  ------         --------------   -----  
 0   Date           145460 non-null  object 
 1   Location       145460 non-null  object 
 2   MinTemp        143975 non-null  float64
 3   MaxTemp        144199 non-null  float64
 4   Rainfall       142199 non-null  float64
 5   Evaporation    82670 non-null   float64
 6   Sunshine       75625 non-null   float64
 7   WindGustDir    135134 non-null  object 
 8   WindGustSpeed  135197 non-null  float64
 9   WindDir9am     134894 non-null  object 
 10  WindDir3pm     141232 non-null  object 
 11  WindSpeed9am   143693 non-null  float64
 12  WindSpeed3pm   142398 non-null  float64
 13  Humidity9am    142806 non-null  float64
 14  Humidity3pm    140953 non-null  float64
 15  Pressure9am    130395 non-null  float64
 16  Pressure3pm    130432 non-null  float64
 17  Cloud9am       89572 non-null   float64
 18  Cloud3pm       86102 non-null   float64
 19  Temp9am        143693 non-null  float64
 20  Temp3pm        141851 non-null  float64
 21  RainToday      142199 non-null  object 
 22  RainTomorrow   142193 non-null  object 
dtypes: float64(16), object(7)
memory usage: 25.5+ MB
```

Proposed Methodology

- We will build an ML model with the help of different ML and Auto ML techniques to predict whether it will rain or not on the next day.
- The dataset which is used in this system have different attributes which will help us in predicting the outcome.

Selected Algorithm :

Random Forest Classifier :-

Random forest is a supervised learning algorithm. The "forest" it builds is an ensemble of decision trees, usually trained with the “bagging” method. The general idea of the bagging method is that a combination of learning models increases the overall result. A random forest is a meta estimator that fits a number of decision tree classifiers on various sub-samples of the dataset and uses averaging to improve the predictive accuracy and control over-fitting.

Random Forest works in two-phase first is to create the random forest by combining N decision tree, and second is to make predictions for each tree created in the first phase.

The Working process can be explained in the below steps and diagram:

Step-1: Select random K data points from the training set.

Step-2: Build the decision trees associated with the selected data points (Subsets).

Step-3: Choose the number N for decision trees that you want to build.

Step-4: Repeat Step 1 & 2.

Step-5: For new data points, find the predictions of each decision tree, and assign the new data points to the category that wins the majority votes.

Random Forest Classifier

In [51]:
```
rfc = RandomForestClassifier()

rfc.fit(X_train, y_train)
```

Out[51]:
```
RandomForestClassifier()
```

In [52]:
```
ypred = rfc.predict(X_test)
```

In [53]:
```
print(confusion_matrix(y_test, ypred))

print("\nAccuracy :",np.round(accuracy_score(y_test, ypred)*100, 2))

print("\n",classification_report(y_test, ypred))
```

```
[[21770   956]
 [ 3345  3021]]

Accuracy : 85.22
```

Gaussian Naïve Bayes :-

Gaussian Naive Bayes is a variant of Naive Bayes that follows Gaussian normal distribution and supports continuous data. Naive Bayes classifiers are a collection of classification algorithms based on **Bayes' Theorem**. It is not a single algorithm but a family of algorithms where all of them share a common principle, i.e. every pair of features being classified is independent of each other.

Bayes' Theorem finds the probability of an event occurring given the probability of another event that has already occurred. Bayes' theorem is stated mathematically as the following equation:

$$P(A|B) = \frac{P(B|A)\,P(A)}{P(B)}$$

where A and B are events and $P(B) \neq 0$.

Basically, we are trying to find probability of event A, given the event B is true. Event B is also termed as evidence.

- P(A) is the priori of A (the prior probability, i.e. Probability of event before evidence is seen). The evidence is an attribute value of an unknown instance(here, it is event B).
- P(A|B) is a posteriori probability of B, i.e. probability of event after evidence is seen.

Gaussian NB

```
In [55]: gnb = GaussianNB()

         gnb.fit(X_train, y_train)

Out[55]: GaussianNB()

In [56]: ypred2 = gnb.predict(X_test)

In [57]: print(confusion_matrix(y_test, ypred2))

         print("\nAccuracy :",np.round(accuracy_score(y_test, ypred2)*100, 2))

         print("\n",classification_report(y_test, ypred2))

         [[19831  2895]
          [ 2716  3650]]

         Accuracy : 80.71
```

K-nearest neighbors:-

- K-nearest neighbors is a classification (or regression) algorithm that in order to determine the classification of a point, combines the classification of the K nearest points. It is supervised because you are trying to classify a point based on the known classification of other points.
- K-Nearest Neighbour is one of the simplest Machine Learning algorithms based on Supervised Learning technique.
- K-NN algorithm assumes the similarity between the new case/data and available cases and put the new case into the category that is most similar to the available categories.
- K-NN algorithm stores all the available data and classifies a new data point based on the similarity. This means when new data appears then it can be easily classified into a well suite category by using K- NN algorithm.
- K-NN algorithm can be used for Regression as well as for Classification but mostly it is used for the Classification problems.

K Nearest Neighbors

In [59]:
```
knn = KNeighborsClassifier(n_neighbors = 3)

knn.fit(X_train, y_train)
```

Out[59]:
```
KNeighborsClassifier(n_neighbors=3)
```

In [60]:
```
ypred3 = knn.predict(X_test)
```

In [61]:
```
print(confusion_matrix(y_test, ypred3))

print("\nAccuracy : ",np.round(accuracy_score(y_test, ypred3)*100, 2))

print("\n",classification_report(y_test, ypred3))
```

```
[[20938  1788]
 [ 3220  3146]]

Accuracy :  82.79
```

XGBoost :-

It works on boosting technique. XGBoost is an ensemble learning algorithm meaning that it combines the results of many models, called base learners to make a prediction.

In [63]:
```
xgb = XGBClassifier(eval_metric='mlogloss')

xgb.fit(X_train, y_train)
```

Out[63]:
```
XGBClassifier(base_score=0.5, booster='gbtree', colsample_bylevel=1,
              colsample_bynode=1, colsample_bytree=1, enable_categorical=False,
              eval_metric='mlogloss', gamma=0, gpu_id=-1, importance_type=None,
              interaction_constraints='', learning_rate=0.300000012,
              max_delta_step=0, max_depth=6, min_child_weight=1, missing=nan,
              monotone_constraints='()', n_estimators=100, n_jobs=8,
              num_parallel_tree=1, predictor='auto', random_state=0,
              reg_alpha=0, reg_lambda=1, scale_pos_weight=1, subsample=1,
              tree_method='exact', validate_parameters=1, verbosity=None)
```

In [64]:
```
ypred4 = xgb.predict(X_test)
```

In [65]:
```
print(confusion_matrix(y_test, ypred4))

print("\nAccuracy Score :", np.round(accuracy_score(y_test, ypred4)*100, 2))

print("\n", classification_report(y_test, ypred4))
```

```
[[21557  1169]
 [ 2957  3409]]

Accuracy Score : 85.82
```

System Requirements

Processor	Minimum: 1.9 gigahertz (GHz) x86 or x64-bit dual core processor with SSE2 instruction set Recommended: 3.3 gigahertz (GHz) or faster 64-bit dual core processor with SSE2 instruction set
Memory	Minimum: 2 GB RAM Recommended: 4 GB RAM
Display	SVGA monitor with a resolution of 1920x1080

Software Requirements

Operating System	Windows 8/ Windows 8.1/ Windows 10/ Windows 11 (Windows 7 is not officially supported but patches are available)
Prerequisites	DirectX 9.0c, .NET Framework 4.1, Microsoft Visual C++ 2005-2013 Redistributable
Others	A stable internet connection with a minimum speed of 25 mbits/sec

8.Implementaiton:

The implementation of the project is divided into seven sections. In the first section, we are going to import the required libraries and then study them. Next, we are going to prepare the dataset with required attributes, then transformations on data are performed, and then data analysis can be made using correlation, followed by splitting of a dataset into train and test sets, finally, model training is done to know the best model that fit(s) our data for predicting rainfall.

Step-1: Import Libraries:

We have imported Numpy, Pandas, Seaborn, Matplot, libraries for evaluating the dataset. Numpy is an open-source module that provides fast mathematical computation on arrays and matrices. We know that Arrays are an integral part of the Machine Learning Ecosystem. Pandas will be useful for performing operations on the data frame. Seaborn and Matplot lib are visualization tools that help us to visualize data in a better way. We have also imported the required algorithms, Random Forest, Decision Tree, and SVM. The Label Encoder is used to convert the categorical variables into numerical variables. The data is trained after it is split into train set and test set.

Step-2: Prepare Dataset:

We have prepared our dataset from various datasets taking the required attributes that are useful for our case study. We should have a basic understanding of our dataset before moving further.

Step-3: Data Preprocessing :

Data Preprocessing is the most vital step while preparing our dataset for model training. Data is often inconsistent, incomplete, and consists of noise or unwanted data. So, preprocessing is required. It involves certain steps like handling missing values,

handling outliers, encoding techniques, scaling. Removing null values is most important because the presence of null values will disturb the distribution of data, and may lead to false predictions. There is very less percent of null values in the dataset.

Missing values:

Imputation is used for replacing missing values. There are few kinds of imputation techniques like Mean imputation, Median imputation, Mode Imputation, Random Sampling imputation, etc. Based on the type of data we have, we can use the required imputation. We have used median imputation to handle missing values.

Handling Outliers:

Outliers are nothing but an extreme value that deviates from the other observations in the dataset. These outliers are either removed or replaced with their nearest boundary value, either upper boundary or lower boundary value.

Label Encoding:

Label Encoding is one of the kinds of encoding techniques that will change categorical variables into numerical variables. It is important to convert the labels because our model can only understand numeric data.

Step-4: Visualization using the technique of Correlation

We need to understand our data. Data visualization is a powerful technique that helps us to know about the trends, patterns that our data follows. There are different techniques to visualize data, one such method is a correlation. Correlation tells us how one or more are related. If two variables are correlated, then we can tell that both are strongly dependent on each other. The variables that are strongly correlated to the target variable, are said to have more influence on the target variable.

Step-5: Splitting Dataset

Dividing the dataset into two sets should be done precisely. The dataset can be divided into the ratio of 80% train set, 20% test set or 70% train set, 30% test set, or any other way. The division of the dataset also affects the accuracy of the training model. A slicing operation can be performed to separate the dataset. We've take care while splitting the dataset, assure that the test set must hold an equivalent features as the train set and also the datasets must be statistically meaningful. Considering the independent variables into 'x' and therefore the target variable into 'y', x = df.iloc [:,:-1].values y = df.iloc [:, 7].values Fig5. Dividing dataset In the above figure, we have divided the dataset into, 80% training dataset 20% testing dataset Ensure that the dataset is split into train and test sets before training the model. Sometimes we may find more accuracy for our models if we involve both datasets.

Step-7: Model Training:

There are several algorithms in machine learning, but we have chosen only three from them to train our model. In Regression, accuracy can be measured by using R2-Score or Mean Squared Error [MSE] or Root Mean Squared Error [RMSE]. The model should be imported from the Sklearn package and then trained. We have to install the Sklearn package and then import it. We have defined a model method to call various models. The model method contains the training and validation statements that will train and test the dataset. MSE value, RMSE value, R2-score are also calculated after training and testing the model. The test dataset will check whether our trained model is efficient for real-time data or not. In Regression, to know the accuracy of the model, we can simply go through R2- score, RMSE, and MSE values. The higher the R2-Score, the efficient the model. The lesser the RMSE and MSE values, the efficient the model. The error value must be less so that our model is more efficient.

9. CONCLUSION AND FUTURE WORK:

This project concentrated on estimation of rainfall and it is estimated that SVR is a valuable and adaptable strategy, helping the client to manage the impediments relating to distributional properties of fundamental factors, geometry of the information and the normal issue of model over fitting. The decision of bit capacity is basic for SVR displaying. We prescribe tenderfoots to utilize straight and RBF piece for direct and non-straight relationship individually. We see that SVR is better than MLR as an expectation strategy. MLR can't catch the non-linearity in a data set and SVR winds up helpful in such circumstances. We additionally process Mean Absolute Error (MAE) for both MLR and SVR models to assess execution of the models. At last, we look at the presentation of SLR, SVR and tuned SVR model. True to form, the tuned SVR model gives the best expectation. We are planning to setup rain prediction system in Latur district.

10.References :

[1] Thirumalai, Chandrasegar, et al. "Heuristic prediction of rainfall using machine learning techniques." 2017 International Conference on Trends in Electronics and Informatics (ICEI). IEEE, 2017.

[2] Geetha, A., and G. M. Nasira. "Data mining for meteorological applications: Decision trees for modeling rainfall prediction." 2014 IEEE International Conference on Computational Intelligence and Computing Research. IEEE, 2014

[3] Parmar, Aakash, Kinjal Mistree, and Mithila Sompura. "Machine learning techniques for rainfall prediction: A review." 2017 International Conference on Innovations in information Embedded and Communication Systems. 2017.

[4] Dash, Yajnaseni, Saroj K. Mishra, and Bijaya K. Panigrahi. "Rainfall prediction for the Kerala state of India using artificial intelligence approaches." Computers & Electrical Engineering 70 (2018): 66-73.

[5] Singh, Gurpreet, and Deepak Kumar. "Hybrid Prediction Models for Rainfall Forecasting." 2019 9th International Conference on Cloud Computing, Data Science & Engineering (Confluence). IEEE, 2019.

[6] Kar, Kaveri, Neelima Thakur, and Prerika Sanghvi. "Prediction of Rainfall Using Fuzzy Dataset." (2019).

[7] Sardeshpande, Kaushik D., and Vijaya R. Thool. "Rainfall Prediction: A Comparative Study of Neural Network Architectures." Emerging Technologies in Data Mining and Information Security. Springer, Singapore, 2019. 19-28.

[8] Chen, Binghong, et al. "Non-Linear Machine Learning Approach to Short-Term Precipitation Forecasting." (2018).

[9] Moon, Seung-Hyun, et al. "Application of machine learning to an early warning system for very short-term heavy rainfall.—Journal of hydrology 568 (2019): 1042-1054.

[10] https://data.gov.in/resources/subdivision-wise-rainfall-andits-departure-1901-2015

http://www.ijstr.org/final-print/jan2020/Prediction-Of-Rainfall-Using-Machine-Learning-Techniques.pdf

https://jespublication.com/upload/2021-V12I6063.pdf

https://ijirem.org/DOC/21_rainfall-prediction-using-machine-learning.pdf

https://www.analyticsvidhya.com/blog/2021/10/a-complete-guide-on-machine-learning-model-deployment-using-Heroku/