WEATHER PREDICTION USING MACHINE LEARNING ⛅

ABSTRACT:

Weather prediction plays a crucial role in various fields such as agriculture, disaster management, transportation, and energy. Traditional weather forecasting relies on numerical weather prediction models, which are computationally intensive and often require significant processing power and time. Machine learning (ML) techniques offer a data-driven approach, enabling faster and potentially more accurate predictions by leveraging historical and real-time data.

This project explores the application of machine learning algorithms for weather prediction, focusing on temperature, precipitation, humidity, and wind speed forecasting. The study involves the collection and preprocessing of meteorological data, feature selection, and the application of supervised learning models such as decision trees, support vector machines (SVM), random forests, and K-neighbors Classifier.

The results demonstrate that ML models can complement traditional forecasting methods, providing efficient and scalable solutions. This study underscores the potential of machine learning to enhance weather prediction accuracy, improve early warning systems, and support decision-making in climate-sensitive sectors.

I. INTRODUCTION:

Weather prediction has always been a crucial aspect of human life, influencing activities such as agriculture, transportation, disaster management, and daily decision-making. Traditionally, meteorological predictions relied on physical models of atmospheric behaviour, leveraging observations of variables such as temperature, humidity, pressure, and wind patterns. While these methods have advanced significantly, they are computationally intensive and often struggle with uncertainty in long-term forecasts.

In recent years, machine learning (ML) has emerged as a transformative tool in weather prediction. By analysing historical data and recognizing complex patterns, ML models can complement traditional meteorological methods, offering faster and sometimes more accurate predictions. Machine learning involves training algorithms on large datasets to learn relationships between input variables (e.g., past weather conditions) and target outputs (e.g., future weather states).

This study investigates the application of machine learning for predicting weather parameters such as temperature, humidity, precipitation, and wind speed. By utilising historical weather datasets, ML algorithms can be trained to predict future conditions with high accuracy. The process begins with data collection from various sources.Preprocessing techniques are applied to clean and standardise the data, ensuring it is suitable for training machine learning models.

A variety of algorithms, including linear regression, decision trees, random forests, support vector machines (SVM), and KNN are employed to tackle both classification and regression tasks. The study concludes with a discussion of the practical implications of ML in weather prediction, future directions for research, and opportunities for real-time deployment in critical applications. This project aims to classify weather conditions into specific categories (e.g., Rain, Fog, Snow, Clear, Cloudy) using a combination of data preprocessing, feature engineering, and machine learning models. By analysing historical weather data, the project implements classification techniques to predict standard weather types efficiently.

II. DATA

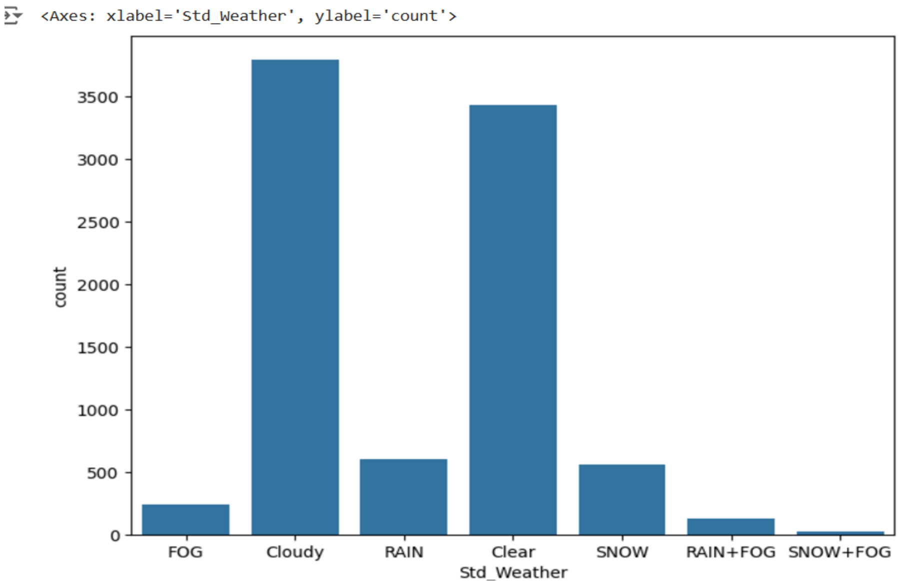
A. Dataset Source:

The dataset utilised in this project is named **"Dataset11-Weather-Data.csv"**,(Link: Dataset11-Weather-Data.csv - Google Drive) containing historical weather observations. This dataset provides a rich set of meteorological variables to analyse and predict different weather conditions effectively.Here is the detailed description of the dataset.

- The dataset includes meteorological variables such as temperature, dew point, relative humidity, wind speed, visibility, and pressure.
- Shape: `(Rows x Columns)`where there are 8784 rows and 8 columns as derived from the initial exploration.

#	Columns	
0	**Date/Time**	The timestamp of the recorded weather data.
1	**Weather**	The observed weather conditions.
2	**Temp_C**	Temperature in degrees Celsius.
3	**Dew Point Temp_C**	Dew point temperature in degrees Celsius.
4	**Rel Hum**	Relative humidity in percentage.
5	**Wind Speed**	Wind speed in kilometres per hour.
6	**Visibility**	Visibility in kilometres.
7	**Press_kPa**	Atmospheric pressure in kilopascals.

I.Data Description



```
<Axes: xlabel='Std_Weather', ylabel='count'>
```

II.Chart Showing weather and counts

B. **Data Balancing**:

The original dataset exhibited an imbalance in the distribution of weather categories. For instance:

- Weather conditions like "Clear" or "Cloudy" were overrepresented.
- Other conditions, such as "Snow" or "Fog," had significantly fewer observations.
- Balance the dataset by sampling an equal number of instances for each weather category (Cloudy, Clear, Rain, Snow).

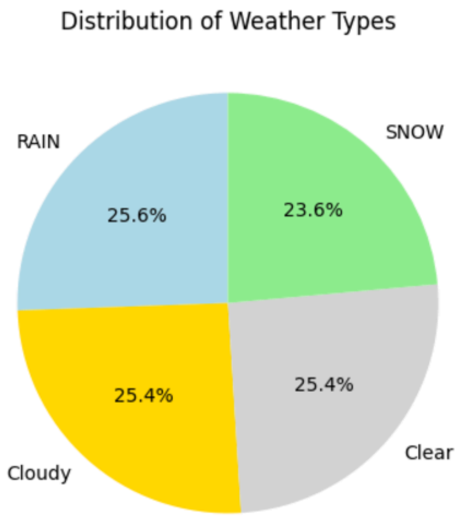
Such imbalance can lead to biased machine learning models that prioritize majority classes while underperforming on minority classes. Addressing this imbalance was crucial to ensure fair and accurate predictions across all weather categories.

Imbalanced Data:

Std_Weather	Value Count
Cloudy	3797
Clear	3432
Rain	603
Snow	556
Fog	241
Rain+Fog	129
Snow+Rain	26

C. Feature Selection and Data Standardization:

Select relevant features for the prediction model, excluding the target variable Std_Weather and standardize the feature set using StandardScaler to ensure that all features contribute equally to the model.



D. Model Training and Evaluation:

Split the dataset into training and testing sets.

- Train several classification models, including Decision Tree, Random Forest, Support Vector Classifier (SVC), Logistic Regression, K-Nearest Neighbors (KNN), and Naive Bayes.
- Evaluate the models using metrics such as accuracy, precision, recall, and F1 score.

III. DESCRIPTION OF METHODS:

A. DECISION TREE CLASSIFIER

A Decision Tree is constructed by asking a series of questions with respect to the dataset. Each time an answer is received, a follow-up question is asked until a conclusion about the class label of the record. The series of questions and their possible answers can be organized in the form of a decision tree, which is a hierarchical structure consisting of nodes and directed edges. It has 3 types of nodes: Root, Internal, and Leaf nodes. In a decision tree, each leaf node is assigned a class label. The non-terminal nodes, which include the root and other internal nodes, contain attribute test conditions to separate records that have different characteristics. Using the decision algorithm, we start at the tree root and split the data on the feature that results in the largest information gain (IG) (reduction in uncertainty towards the final decision). In an iterative process, we can then repeat this splitting procedure at each child node until the leaves are pure. This means that the samples at each leaf node all belong to the same class.The classifier was implemented using `sklearn.tree` package.

B. RANDOM FOREST CLASSIFIER

As its name implies, Random Forest Classifier consists of a large number of individual decision trees that operate as an ensemble. Each individual tree in the random forest spits out a class prediction and the class with the most votes becomes our model's prediction. The low correlation between models is the key as they can produce ensemble predictions that are more accurate than any of the individual predictions, as the trees protect each other from their individual errors. The process of Bagging is used to diversify models as each individual tree is allowed to randomly sample from the dataset with replacement.The classifier was implemented using sklearn.ensemble package.

C. LOGISTIC REGRESSION

Regression analysis is a predictive modelling technique that analyzes the relation between the target or dependent variable and independent variable in a dataset. Regression analysis techniques get used when the target and independent variables show a linear or non-linear relationship between each other, and the target variable contains continuous values. Regression analysis involves determining the best fit line, which is a line that passes through all the data points in such a way that distance of the line from each data point is minimized.

Logistic regression is one of the types of regression analysis technique, which gets used when the dependent variable is discrete. Example: 0 or 1, true or false, etc. This means the target variable can have only two values, and a sigmoid curve denotes the relation between the target variable and the independent variable, by mapping any real value to a value between 0 and 1.The classifier was implemented using `sklearn.linear_model` package.

D. GAUSSIAN NAIVE BAYES

Naive Bayes classifiers are a collection of classification algorithms based on Bayes' Theorem of mathematics. In simple words, the Bayes' theorem describes the probability of an event, based on prior knowledge of conditions that might be related to the event. 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. Naive Bayes is a classification algorithm for binary (two-class) and multi-class classification problems. The technique is easiest to understand when described using binary or categorical input values. It is called naive Bayes because the calculation of the probabilities for each hypothesis are simplified to make their calculation tractable.

Naive Bayes can be extended to real-valued attributes, most commonly by assuming a Gaussian distribution. This extension of Naive Bayes is called Gaussian Naive Bayes. Beside the Gaussian Naive Bayes there are also existing the Multinomial naive Bayes and the Bernoulli naive Bayes. We picked the Gaussian Naive Bayes because it is the most popular one and one of the simplest to implement.The classifier was implemented using sklearn.naive_bayes package.

E. SUPPORT VECTOR CLASSIFIER

SVC is a supervised machine learning model that uses the principles of Support Vector Machines (SVM) for classification tasks. It is commonly employed for binary classification but can be extended to multi-class problems.

SVC works by finding the optimal hyperplane that best separates data points of different classes in a feature space. The hyperplane is a decision boundary that maximizes the margin between data points of different classes. The data points closest to the hyperplane are known as support vectors, and they play a crucial role in defining the position and orientation of the hyperplane.The classifier was implemented using sklearn.svm package.

F. KNEIGHBORS CLASSIFIERS

KNeighborsClassifier is a simple, non-parametric, and lazy learning algorithm used for classification and regression tasks. It is based on the principle that data points that are close in the feature space are more likely to belong to the same class. KNN works by finding the ‘k’ nearest neighbors to a query point and assigning the class based on the majority vote of its neighbors.It is Simple and easy to implement.The classifier was implemented using sklearn.neighbors package.

IV EXPERIMENTS AND RESULTS

For our supervised learning technique analysis, we’ve used Naive Bayes (Gaussian), Logistic Regression, and Decision Tree, RandomForest Classifiers, SVM, and KNeighborsClassifier . In our research, we found that the Decision Tree classifier performed the poorest, whereas the Random Forest Classifier gave the best result in terms of every metric. The Decision Tree Classifier, while interpretable and easy to visualize, struggled with the complexity of the dataset, leading to lower performance metrics.

It wasn’t surprising to see the Random Forest classifier performing the best. The KNeighborsClassifier performed better than Naive Bayes classifier and Logistic Regression. The Random Forest Classifier came out on top in all the performance metrics, which was expected as it is an extension of the Decision Tree classifier, averaging out results of multiple recursions of the same.

The performance of various machine learning models was evaluated using the specified metrics of accuracy, precision, recall, and F1 score.

- Accuracy: The proportion of correctly predicted instances out of all instances.

$$\text{Accuracy}=TP+TN \backslash TP+TN+FP+FN$$

- Precision: The proportion of correctly predicted positive instances out of all predicted positive instances.

$$\text{Precision}=TP\backslash TP+FP$$

- Recall: The proportion of correctly predicted positive instances out of all actual positive instances.

$$\text{Recall}=TP\backslash TP+FN$$

- F1 - Score: The harmonic mean of precision and recall. It provides a balanced measure when the class distribution is uneven.

$$\text{F1 Score}=2 \cdot \text{Precision} \cdot \text{Recall}\backslash \text{Precision}+\text{Recall}$$

	Model	Accuracy	Precision	Recall	F1-Score
0	SVC	0.608051	0.593965	0.608051	0.599027
1	Logistic Regression	0.605932	0.592590	0.605932	0.597579
3	KNN	0.641949	0.642468	0.641949	0.640582
4	Naive Bayes	0.582627	0.546841	0.582627	0.548718
5	Decision Tree	0.584746	0.585854	0.584746	0.584998
6.	Random Forest	0.680085	0.667894	0.680085	0.67211

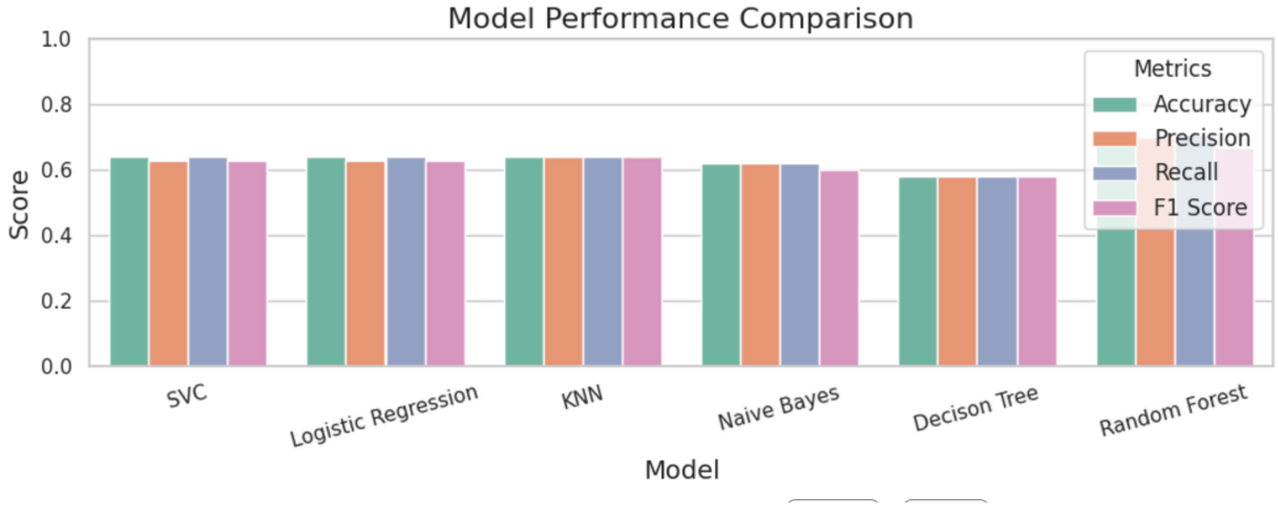


Fig. 1

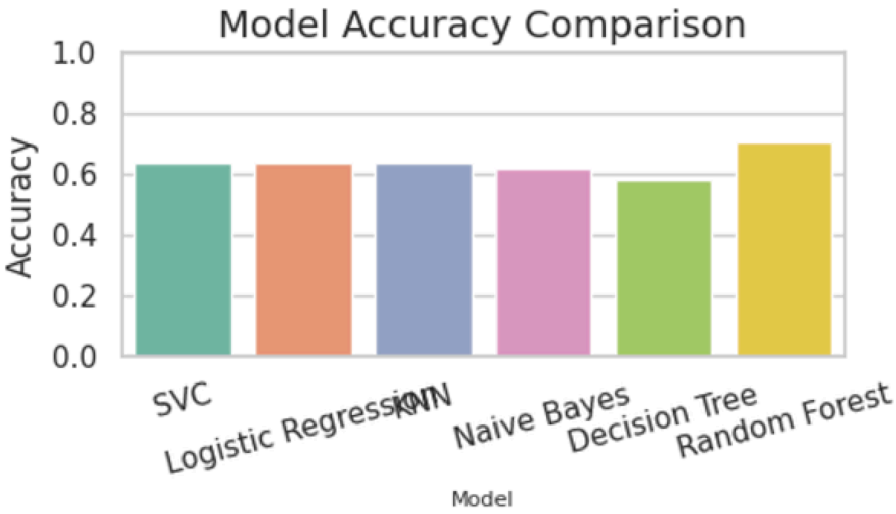


Fig 2

- Fig. 2 indicates the proportion of correctly classified instances out of the total instances. The Random Forest Classifier achieved the highest accuracy of 71%, suggesting that it effectively captured the patterns in the data. Other models, such as Support Vector Classifier (SVC), Logistic Regression, and K-Nearest Neighbors (KNN), achieved an accuracy of 64%, indicating moderate performance. The Decision Tree Classifier exhibited the lowest accuracy of 58%, suggesting that it may be overly simplistic for this dataset.

V CONCLUSION

In conclusion, this project demonstrates the effectiveness of machine learning algorithms in classifying weather conditions based on various meteorological features. It successfully demonstrated the application of machine learning techniques for weather classification, achieving promising results with the Random Forest Classifier. While the performance metrics indicate a solid understanding of weather patterns, there remains room for improvement through feature enhancement and model optimization.Future work could involve incorporating additional features, such as geographical data or historical weather patterns, to further enhance model accuracy.