****

**WEATHER**

**CLASSIFICATION**

**REPORT**

**Forecast with Precision: Using Simulated Dataset for Weather Types Prediction**

**Presented by:**

**Falola Caleb Oluwapelumi**

**(March Cohort, 2024)**

**TABLE OF CONTENTS**

* Introduction
* Objective
* Problem Statement
* Data Collection Methodology
* Data Features
* Data Preprocessing
* Data Visualization
* Model Selection
* Model Optimization
* Challenges Faced
* Classification Metric Evaluation
* Conclusion

**INTRODUCTION**

Data Link: <https://www.kaggle.com/datasets/nikhil7280/weather-type-classification>

In an era of climate change and increasingly unpredictable weather patterns, accurate and timely weather forecasting has become more crucial than ever. This study explores the application of machine learning techniques to enhance weather prediction capabilities, potentially transforming how we understand and prepare for atmospheric conditions.

Traditional weather forecasting methods, while valuable, often struggle to capture the complex, non-linear relationships present in essential atmospheric dynamics. Machine learning offers a promising solution by leveraging math, statistics and vast amounts of historical and real-time data to identify subtle patterns and correlations that might escape conventional analysis.

This study focuses on the development and evaluation of a machine learning model designed to predict key weather parameters, including temperature, precipitation, wind speed, and humidity. By harnessing the power of algorithms such as Random Forests, Gradient Boosting, and Ensemble techniques, we aim to improve both the accuracy and the lead time of weather forecasts.

The simulated dataset used in this project comprises of various key weather parameters such as Wind Speed, Temperature, Precipitation, Location, Humidity, Season, Cloud Cover, UV Index, Visibility and Atmospheric pressure to predict a multinomial target column (Weather Type) which is classified into Sunny, Cloudy, Rainy and Snowy.

****

**OBJECTIVE**

The primary objective of this study is to develop and implement an advanced machine learning model for accurate and reliable weather forecasting.

Throughout this report, we will:

1. Detail the data preprocessing and feature engineering techniques employed to prepare our dataset for machine learning algorithms.

2. Explain the architecture and thought process behind our chosen machine learning models.

3. Present the results of our models, comparing their performance against traditional forecasting methods.

4. Discuss the implications of our findings for the field of meteorology and potential real-world applications.

5. Address the challenges encountered during the study, as well as future directions for research.

By the conclusion of this report, we aim to demonstrate the potential of machine learning in enhancing weather forecasting capabilities, ultimately contributing to improved decision-making in areas ranging from agriculture and urban planning to disaster preparedness and daily life.

**PROBLEM STATEMENT**

The challenge is to accurately classify weather conditions due to the changeable climate, which poses a difficulty in assigning classified labels. Traditional forecasting methods have limitations, and the dynamic nature of weather conditions requires exploring machine learning techniques such as, Logistic Regression, Support Vector Classifier, K Neighbors Classifier, Random Forest Classifier etc. The problem is to develop Machine Learning models that can classify multinomial weather conditions, which requires experimentation and exploring various machine learning techniques and algorithms to identify the most effective combination for accurate classification between the models.

**DATA COLLECTION METHODOLOGY**

The dataset used was synthetically generated to mimic weather data for classification tasks. It includes various weather-related features and categorizes the weather into four types: Rainy, Sunny, Cloudy, and Snowy. This dataset was artificially designed for practicing classification algorithms, data preprocessing, and outlier detection methods.

The dataset has a sample size of 13200 rows and 11 columns, which featured various numerical and categorical columns. It was stored in a comma-separated values(csv) format file.

Dataset Update Frequency - Annually (Last Updated June 2024)

**NOTE**: The values, ranges, and distributions may not accurately represent real-world conditions, and the data should primarily be used for educational and experimental purposes.

**DATA FEATURES:**

* Temperature (numeric): The temperature in degrees Celsius, ranging from extreme cold to extreme heat.
* Humidity (numeric): The humidity percentage, including values above 100% to introduce outliers.
* Wind Speed (numeric): The wind speed in kilometers per hour, with a range including unrealistically high values.
* Precipitation (%) (numeric): The precipitation percentage, including outlier values.
* Cloud Cover (categorical): The cloud cover description.
* Atmospheric Pressure (numeric): The atmospheric pressure in hectopascal(hpa), covering a wide range.
* UV Index (numeric): The UV index, indicating the strength of ultraviolet radiation.
* Season (categorical): The season during which the data was recorded.
* Visibility (km) (numeric): The visibility in kilometers, including very low or very high values.
* Location (categorical): The type of location where the data was recorded.
* Weather Type (categorical): The target variable for classification, indicating the weather type.

**DATA PREPROCESSING**

Data preprocessing is the process of preparing data for machine learning models. It includes several techniques to clean, transform, and organize raw data into a suitable format for analysis.

The aim of data preprocessing is to improve data quality, make the data suitable for machine learning algorithms, reduce or transform noise and irrelevant information and ultimately enhance model performance and accuracy.

The data preprocessing techniques used in this study includes:

1. Data Cleaning:

* Handling missing values
* Handling duplicate values
* Convert values into consistent metric for better interpretation
* Outlier detection and removal

1. Data Transformation:

* Standardization (Standard scaling)
* Encoding categorical variables

1. Data Reduction:

* Dimensionality Reduction

**Dimensionality reduction:** This is a technique that helps us reduce the number of dimensions or features in our dataset while still preserving most of the important information or patterns present in the original dataset. By employing dimensionality reduction, it becomes much easier to visualize data, spot patterns and perform tasks like clustering of classification more effectively.

The Dimensionality reduction technique used in this project is called Principal Component Analysis (PCA)

**PCA:** Principal component analysis, or PCA, is a dimensionality reduction method that is often used to reduce the dimensionality of large data sets, by transforming a large set of variables into a smaller one that still contains most of the information in the large set. PCA identifies the directions (vectors) in the feature space along which the data varies the most. To reduce dimensions, only the top k principal components are kept. The choice of k depends on how much of the original variance you want to retain. During this study, k was specified as 5.

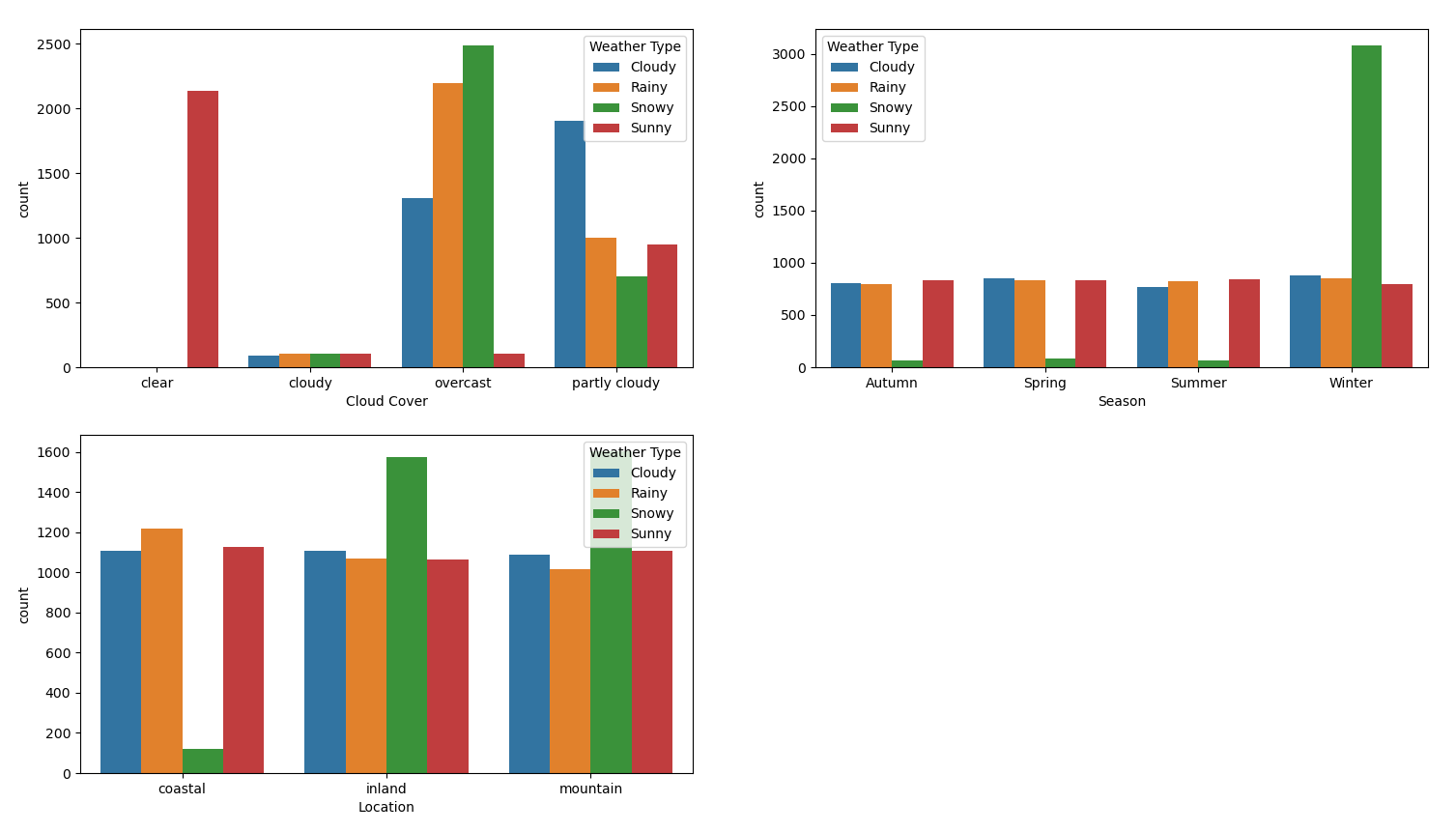
**Data Pipeline:** A pipeline in machine learning is a sequence of data processing components connected in series, where the output of one component becomes the input of the next. It typically transforms multiple steps of a machine learning workflow into a single unit of code. It can beused for multiple transformations of the same columns. A data pipeline improves workflow organization, prevents data leakage, ensures reproducible results, simplicity and is easier deployed.

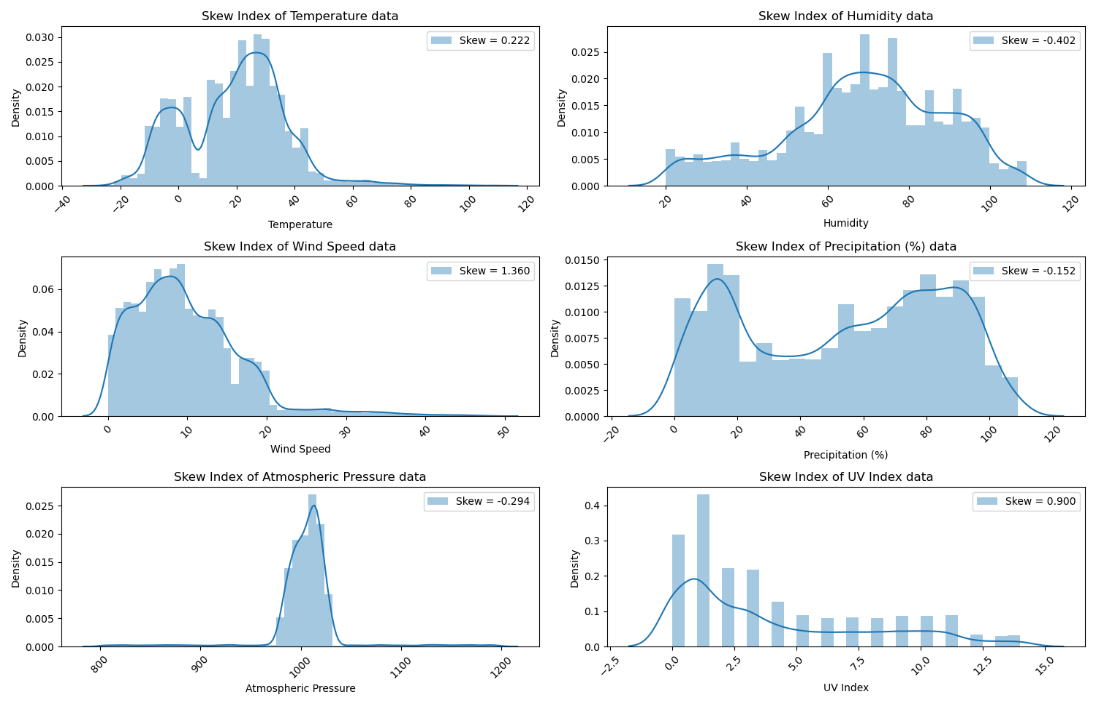
**Column Transformers:** allows us to create and apply different transformations to specific columns of our data. The main advantage of a column transformer is that it allows you to handle different types of features (numerical, Categorical) in a convenient and organized way, helps you avoid repeating the same transformation code for different subsets of features, making your code more readable and maintainable.

The pipeline and column transformers were applied in this study for Standardization of Numerical values and Encoding of Categorical Values

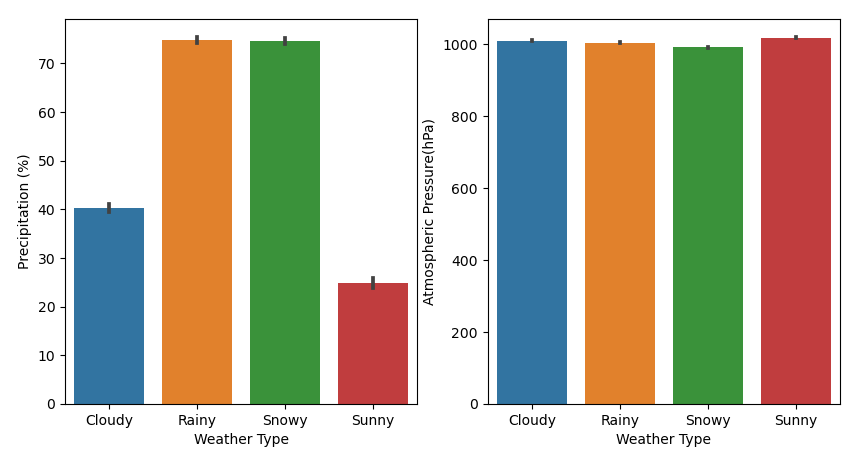
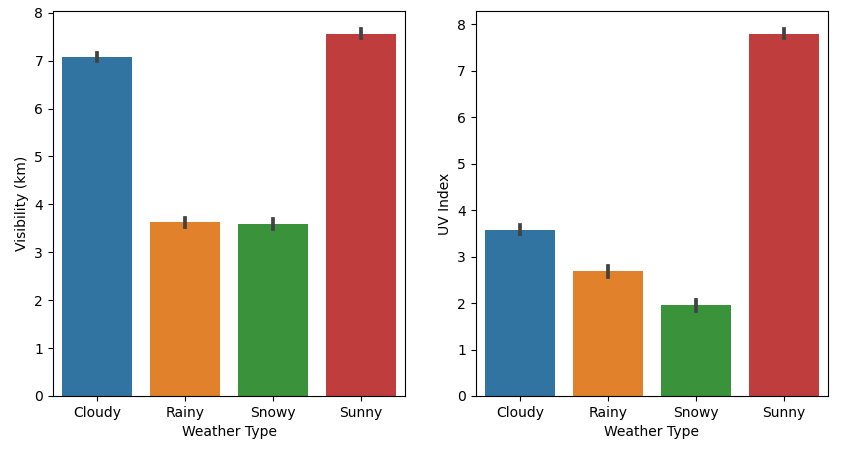
**DATA VISUALIZATION**

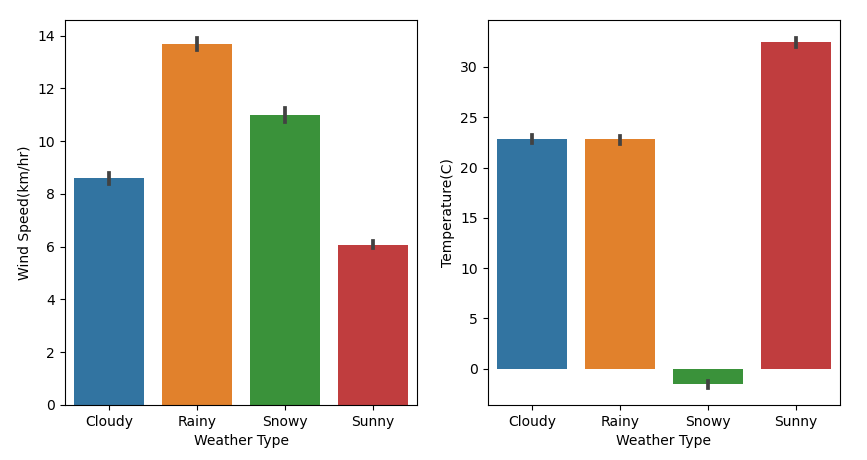
Data Visualization helps uncover hidden patterns and trends in the dataset, it also helps to understand the data better and overall improves model evaluation.

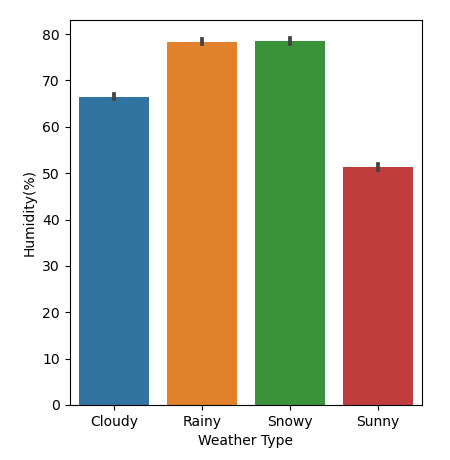
****

****

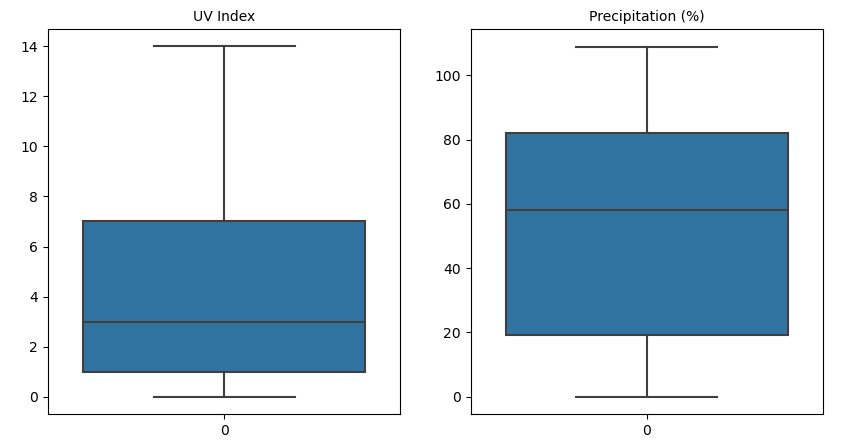
**Kernel Density Plot(kde):** A kernel density plot is used to detect the skewness of the dataset. It is particularly useful for understanding the shape of a data distribution. The kde plot above shows some positive values indicating right skew and negative values indicating left skew.

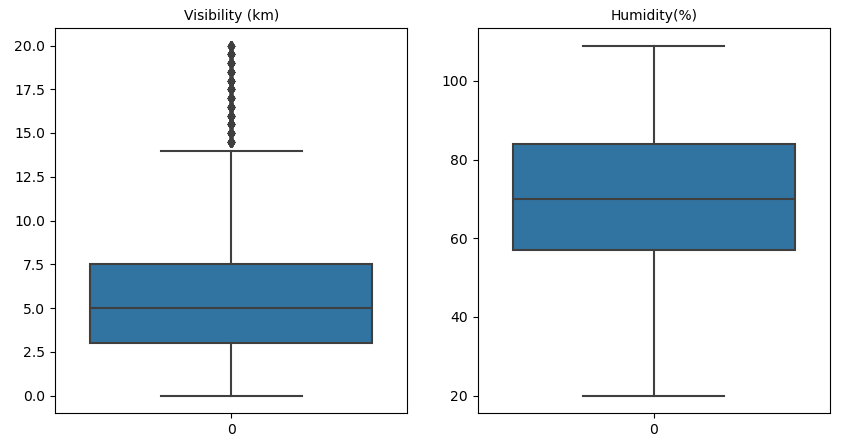
****

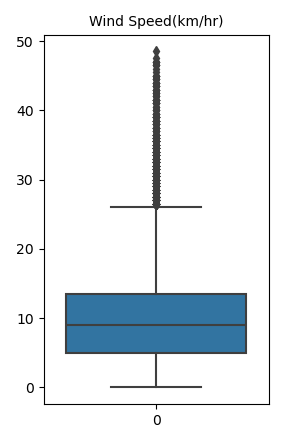
****

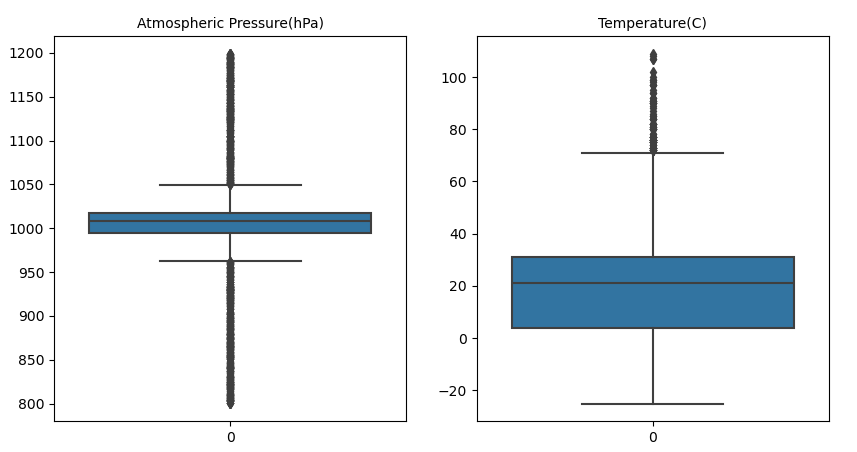
****

**Bar Plot:** This is used to detect and visualize trends and patterns in the dataset. The above bar chart shows the relationship between the target variable (Weather Type) and every other feature in the dataset.

****

****

****

****

**Box Plot:** This is used to detect and Visualize Outliers in the dataset. From the chart above, Outliers can be seen in the most columns of the dataset e.g. Atmospheric Pressure, Temperature, Wind speed etc. it is represented by dark circles on the lines in the plot.

**MODEL SELECTION**

**SUPERVISED LEARNING:** Supervised machine learning is a technique where a model learns from labelled data to make predictions or classifications. The labelled data includes input features and the corresponding desired outputs or target values.

Supervised learning can be used for regression tasks, where the output is a continuous variable, or classification tasks, where the output is a discrete variable/category. In the context of this weather and climate forecasting dataset, Supervised learning was used in the classification of the output which is a multinomial classification variable. There are various algorithms used in supervised learning, such as decision trees, support vector machines, random forests, logistic regression, xgboost, linear regression etc.

It is advisable to approach various and diverse models to increase accuracy and performance.

Here are an Overview of the Classification supervised models used during this study:

**Random Forest**: is a supervised machine learning algorithm implemented from the sckit-learn library used for classification and regression problems. It is an ensemble learning method that combines multiple decision trees to generate a robust and accurate model. A decision tree is a predictive model that uses a tree-like graph to map out different possible outcomes and the decisions that lead to them. Each internal node represents a decision, while each leaf node represents a predicted outcome. The algorithm randomly selects a subset of features and data points to create each decision tree, which reduces overfitting and improves the model's generalization ability. Random Forests has several advantages, including its ability to handle high-dimensional data, non-linear relationships, and missing values. It is also insensitive to noise and outliers in the data.

**K-Neighbors:** The KNN algorithm is another sckit-learn supervised machine learning algorithm that can be used for both classification and regression problems. It works based on the principle of similarity. The KNN algorithm classifies or predicts a new data point by comparing it to the labeled data points in the training dataset. The algorithm measures the distance between the new data point and its K nearest neighbors in the feature space. The K nearest neighbors is determined based on a distance metric, commonly the Euclidean distance. Once the neighbors are identified, the algorithm assigns the majority class label for classification or calculates the average value for regression. KNN has some limitations, including its sensitivity to outliers and its inability to handle high-dimensional data.

**Logistic Regression:** Logistic Regression model can also be used to solve multiclass problems. It is a scikit-learn library that works well for binary classification and also multiclass classification by using extensions such as One vs Rest Approach, multiclass strategy etc. This technique uses math to find the relationships between data factors. It then uses this relationship to predict the value of one of those factors based on the others. The prediction usually has a finite number of outcomes, like binary or multiclass.

**SVM:** The SVM algorithm is a sckit learn classifier that aims to find the optimal hyperplane that maximizes the margin between two classes of data points. The margin is defined as the distance between the hyperplane and the nearest data points from each class. SVM tries to maximize this margin while minimizing the classification error. The SVM algorithm has several advantages over other machine learning algorithms. For instance, it can handle high-dimensional data efficiently and is less prone to overfitting than other algorithms such as decision trees and neural networks. SVM can also handle noisy data and is insensitive to outliers.

In its most simple type, SVM doesn't support multiclass classification natively. It supports binary classification and separating data points into two classes. For multiclass classification, the same principle is utilized after breaking down the multiclassification problem into multiple binary classification problems.

**Gradient Boosting:** Gradient Boosting is a sckit-learn library that uses functional gradient algorithm to repeatedly select a function that leads in the direction of a weak hypothesis or negative gradient so that it can minimize loss function. Gradient boosting classifier combines several weak learning models to produce a powerful predicting model. It is also insensitive to outliers in the dataset.

NOTE: Keep in mind not all classification models work directly well with multinomial classification dataset, most of them have to be adjusted.

**MODEL TRAINING:**

Machine Learning Datasets was split into two sets, one for training the model and the other for testing the model’s accuracy. This helps to maximize the performance of the models. Each model was trained using 70% of the dataset and tested on the remaining 30%.

**MODEL OPTIMIZATION**

The Model optimization was done using the ensemble technique (voting classifier), cross validation and hyper parameter optimization.

**Hyper Parameter Optimization:** First, a grid containing each model parameters were researched on and created to increase model’s accuracy. Then they were cross validated using k-fold cross validation technique with a value of K equals to 5 for all models.

The K-fold cross-validation used is a technique to evaluate the performance of the machine learning model by dividing the dataset into 5 equal folds. The model is trained on 4 of the folds and tested on the remaining one-fold. This process is repeated 5 times, with each fold used once as the testing set. The performance of the model is then averaged over the 5 folds

A randomized search was also employed in going through the parameters grid to randomly find the best parameters on average, with number of iterations specified as 10, 10 different parameter grids would be randomly selected 5 times, i.e. 10\*5 = 50 fits. The best results obtained from each of the 50 K-fold iterations were used to determine the optimal hyperparameters for each model. By using this method, we ensured that the models were optimized for the best possible performance.

**Ensemble Technique (Voting Classifier):** Ensemble is the process of combining two or more models to produce a combined model that is more accurate than any of the individual models, it takes the best characteristics of all the models combined to produce a model more accurate than any individual model. It balances out the weakness of individual models and reduces the risk of overfitting. The ensemble method used during this study is the Voting Classifier. It was used because of its flexibility (can effectively ensemble diverse types of models), simplicity (compared to boosting or stacking) and Effectiveness.

Here are the individual model results after optimization and Ensemble:

|  |  |
| --- | --- |
| **MODEL** | **ACCURACY** |
| RANDOM FOREST CLASSIFIER | 0.9648 |
| GRADIENT BOOSTING CLASSIFIER | 0.9626 |
| SUPPORT VECTOR CLASSIFIER(SVC) | 0.9683 |
| LOGISTIC REGRESSION | 0.9635 |
| K-NEIGHBORS CLASSIFIER | 0.9673 |
| VOTING CLASSIFIER(ENSEMBLE) | 0.9670 |

**CHALLENGES FACED**

**Outliers:** These are values which fall far from the expected variations and normal range of the data distribution. They are anomalies within a dataset which could result from data entry error, measurement errors, intentional outliers, unit or scale issues etc.

The dataset used for this study had a wide range of outliers in various columns in the data. This would undoubtedly impact data quality, analysis and model performance. The outliers were detected and displayed using a boxplot for the numerical columns and a value count was done for the categorical columns.

All the outliers present in the dataset were in the numerical columns, seeing as the dataset had a wide range or samples, I decided to drop the Samples with the outliers using the Z-score method

The z-score is a statistical measure that quantifies how many standard deviations a data point is from the mean of a dataset and its relationship with the threshold.

z-score was used because of its simplicity, flexibility and standardization i.e. it puts all the variables on the same scale. The z-score and standardization technique were also used to handle skewing in the dataset.

**Feature scaling:** This is a preprocessing technique that normalizes data, ensuring all features contribute equally by converting them to a common scale, thereby improving algorithm performance. The dataset had a wide scale of data, from Atmospheric pressure (800-1199) to UV Index (0-14), thereby feature scaling was necessary.

The scaling method used in the in the study is called Standardization or Standard Scaling, this process centers the values around the mean 0 with a unit standard deviation of 1.

Standard scaling was used because its less affected by outliers, preserves data distribution, centers the data around 0 and it highly compatible with most machine learning models and PCA.

**Encoding:** Encoding techniques are methods used to convert categorical (non-numeric) data into a format that machine learning algorithms can understand and process. Most machine learning algorithms work with numerical data. Proper encoding preserves the information and relationships within the data.

Common Encoding techniques includes Label encoding, One-Hot encoding, Target encoding etc.

The dataset had a total of four non-numeric data which had to be converted into numerical data for the of the machine learning model, One-Hot encoding was using in this study, it was selected because it works best with non-ordinal categorical data i.e. categories that have no natural order (e.g. cities, season etc.), keeps features interpretable, preserves data relationship, and is perfect with dataset where number of unique categories is relatively small.

**CLASSIFICATION MODEL EVALUATION METRIC**

The Evaluation metrics used for the model is the classification report, this is a collection of different evaluation metrics. This was done on our chosen model, The voting classifier.

**Accuracy:** The accuracy of the model in decimal form, perfect accuracy is 1.0

**F1 Score:** A combination of precision and recall. A perfect model achieves an F1 score of 1.0. it is a measure of model’s accuracy that takes into account model’s ability to identify relevant data(recall) and ignore irrelevant data(precision) in its results.

**Recall:** Indicates the proportion of actual positives which were correctly classified. A model which produces no false negatives has a recall score of 1.0. This is measure of the model’s ability to identify relevant data.

**Precision:** This indicates the proportion of positive identifications which were actually correct. This is measure of the model’s ability to ignore irrelevant data

**Support:** The number of samples each metric was calculated on.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | PRECISION | RECALL | F1 SCORE | SUPPORT |
| Cloudy | 0.95 | 0.96 | 0.95 | 784 |
| Rainy | 0.97 | 0.97 | 0.97 | 822 |
| Snowy | 0.98 | 0.99 | 0.99 | 846 |
| Sunny | 0.96 | 0.95 | 0.95 | 704 |
| Accuracy |  |  | 0.97 | 3156 |
|  |  |  |  |  |

CLASSIFICATION REPORT ON VOTING CLASSIFIER

**CONCLUSION**

In this study, we developed and evaluated six machine learning models for predicting weather conditions based on experimental data. Our results indicate that the KNN, SVM and RF models performed the best in testing and predicting the data, with accuracy rates of 96.7%, 96.8% and 96.4%, respectively. Logistic Regression and Gradient Classifier had lower accuracy rates of 96.3% and 96.2%, respectively. These findings suggest that machine learning techniques can be highly effective in predicting weather conditions especially when ensembled which would only help increase the performance of the model, it also showed the importance of appropriate feature engineering, outlier detection, skewness, standardization, hyper parameter optimization, and how choice of model or models can significantly impact prediction accuracy and performance. Overall, the results of this study have important implications for weather forecasting. Further research is needed to explore the potential of these models with actual real time data and refine their performance over time.