Module Code & Module Title

CU6051NI - Artificial Intelligence

Assessment Weightage & Type

75% Individual Coursework

Year and Semester

2023-24 Autumn

Student Name: Aayusha Lamichhane

London Met ID: 21039801

College ID: NP01CP4A210095

Assignment Due Date: January 17th, 2024

Assignment Submission Date: January 17th, 2024

I confirm that I understand my coursework needs to be submitted online via Google Classroom under the relevant module page before the deadline in order for my assignment to be accepted and marked. I am fully aware that late submissions will be treated as non-submission and a marks of zero will be awarded

Abstract

Weather forecasting plays a vital role in societies, influencing decisions ranging from daily activities to disaster preparedness. The integration of machine learning algorithms in providing accurate forecasts has enhanced the quality of living. This abstract provides an overview of the key machine learning algorithms used in weather prediction, including Naïve Bayes Classifier, K-nearest neighbors (KNN), and Support Vector Machines (SVM). The performance metrics are accuracy, precision, recall, F1-score and error rate. Further work involves the development of a pseudocode based on classification algorithm with the relevant dataset. The report further explores the significance of machine learning integration with weather prediction, addressing their advantages, disadvantages, and areas of improvement.

Table of Contents

Table of figures

Table of Tables

Table of Equations

1 Introduction

1.1 Explanation of topic and concepts

Artificial intelligence (AI), the ability of computer systems to carry out the task that require human intelligence. The goal of AI is to possess the characteristics of intelligent entities, such as the ability to reason, speech recognition, decision-making, or learn from past experience (Copeland). AI is being used across multiple industries from health care to education industries. Since, AI advancement it has been able to solve the problem related to human cognitive functions such as playing games, interpreting speech, and identifying patterns. They learn it by processing huge amount of data or information, looking for a pattern to model their own decision-making (Schroer). The most common application of AI in our life includes, ChatGPT, Netflix, Google Translate (translates text from one language to another), Tesla (self-driving car through computer vision) and so on.

Defining intelligence can be challenging, in order to avoid confusion AI experts have differentiated AI between strong AI, and weak AI:

Strong AI or General: Strong AI possesses human-like cognitive abilities across various tasks. It is still a goal for the future research and development as it requires a machine who performs a task as an intellectual being. Within the context of strong AI philosophy, the concept claims that there is no inherent distinction between software, such as AI, imitating the cognitive processes of the human brain and the behaviors exhibited by an individual. This resemblance encompasses not only comprehension but also awareness (Rouse, 2023). The key trait of Strong AI includes problem-solving skills, system is made proficient in complex problem-solving, reasoning, and decision-making. The effort to create strong artificial intelligence continues to be a persistent and demanding search within the field of AI.

Weak AI or Narrow: Weak AI are trained to do a particular task or they are guided on performing specific tasks, such as virtual personal assistants like Siri or Alexa whose task is to assist human being, or a recommendation algorithm which learns the preference of a client and shows the content based on the engagement of the client. Weak AI will just follow the rules that are set for it; it will not be able to think for itself (Rouse, 2023). These are created to solve a specific problem and are designed to thrive in a certain area with clearly defined tasks. It lacks flexibility in its cognitive capacities because the tasks are predetermined.

Weak AI can be seen in the following examples:

- Chatbots
- Autonomous Vehicles
- Email filtering
- Virtual personal assistant like Siri or Alexa and so on.

Machine Learning

Machine learning (ML) is a subset of artificial intelligence (AI) which specializes in the development of computer algorithms and models that enables computers to learn from data. It enables computers to learn from data and make decisions or predictions without being explicitly programmed to do so. Machine learning revolves around the development and application of algorithms, to enhance the performance of the model to be more precise and accurate.

For instance, if we want a computer to be able to recognize images of dogs, we provide it images of dogs and let the machine learning algorithm to figure out the common patterns and features that define a dog (Crabtree, 2023). As the algorithm analyzes large volume of images containing dog, its ability to identify dogs improves, even when presented with images it has not encountered before.

Deep Learning

Deep learning is a subset of machine learning model that is based on artificial neural networks (ANN) with multiple layers that is inspired by functioning and structure human brain. This network consists of nodes (neuron) that process the data into a meaningful output. Countless artificial intelligence (AI) services and applications that increase automation by carrying out physical and analytical operations without the need for human intervention are powered by deep learning (IBM, 2024). Deep learning is applied in Image Recognition, Speech Recognition, Autonomous Vehicles and so on.

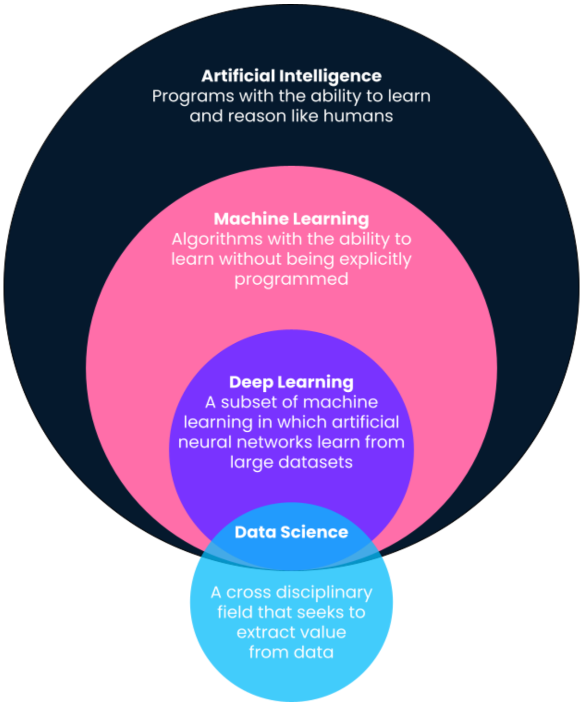

Artificial Intelligence
Programs with the ability to learn and reason like humans
Machine Learning
Algorithms with the ability to learn without being explicitly programmed
Deep Learning
A subset of machine learning in which artificial neural networks learn from large datasets
Data Science
A cross disciplinary field that seeks to extract value from data

Figure 1: Comparing different fields (Crabtree, 2023).

Types of Machine Learning

Machine learning is classified into three different types, they are defined below:

Supervised learning

The most common type of machine learning is supervised learning, the model is trained on a labeled dataset (Crabtree, 2023). The aim of this algorithm is to learn the mapping from input to outputs, making predictions or classifications when the data provided has not been encountered before. The common task of supervised learning includes regression and classification.

- **Classification:** Classification is an algorithm that addresses classification problems where the output variable is categorical; male or female, yes or no (Kanade, 2023). The objective of classification is to predict the category of new, unseen data based on the training dataset. The algorithm that classification uses commonly include Decision Trees, Random Forest, Support Vector Machines (SVM), K-Nearest Neighbors (k-NN), and Naïve Bayes.

- **Regression:** Regression is an algorithm that works on the inputted data to predict the numerical values, it tries to find a relationship between the input variables and the output variable by fitting a mathematical model to the data (Pandey, 2022). The objective of regression is to make accurate prediction of the numeric value for the new variable, or unseen data. The common algorithms used by regression include Linear Regression, Polynomial Regression, Random Forest, and Decision Trees.

Unsupervised Learning

Unsupervised learning as the name suggests, involves working on an unlabeled dataset. Based on the data the model is left to find the patterns and relationships on its own. This algorithm tried to find the patterns, relationships, within the data without the external guidance. An unsupervised learning algorithm, for instance, might scan through a large dataset containing information on various fruits, find patterns in the data points, and then sort the data according to similar patterns, such as the length and breadth of the data. The unsupervised learning methods include clustering and dimensionality reduction.

Reinforcement Learning:

Reinforcement learning involves taking suitable action to maximize reward in particular situation, the objective to use this is to find best possible behavior or path to find answers to a specific situation (geeksforgeeks). It is used while playing games such as Go or teaching an agent to play a chess. The objective of reinforcement learning is to identify the optimal action model that maximizes the overall cumulative reward for its agent. In the absence of a training dataset, the reinforcement learning problem is resolved through the agent's independent actions, guided by input from the environment (Brooks, 2023).

1.1.1 Explanation of the chosen problem domain/topic

Weather prediction, helps in estimating the condition of the atmosphere at a specific location over a time. To predict the weather of a certain location or geography involves using data analysis, scientific principles, and computational models. The goal of weather prediction is to help individuals, businesses, and government make appropriate decision based on the information provided. The absence of weather forecasting could cause serious damage in various sectors such as businesses, human life, society, and the economy. The basic area which could be affected include:

- **Transportation Sectors**: Airlines, road way transport, and shipping rely on weather forecasts to plan their route and to run their operation smoothly.
- **Energy Sector:** The renewable energy sources such as wind and solar are highly dependent on weather to generate power.
- **Agriculture:** Farmers rely on weather prediction to plan their planting and harvesting schedules.
- **Natural Disaster Management:** Weather is associated with various natural disasters such as flood, tornadoes therefore timely weather prediction is needed in order to respond to the natural disasters.

The absence of weather forecasting could cause hinder in various sectors, above mentioned were only the key sectors.

Some Facts and Figures

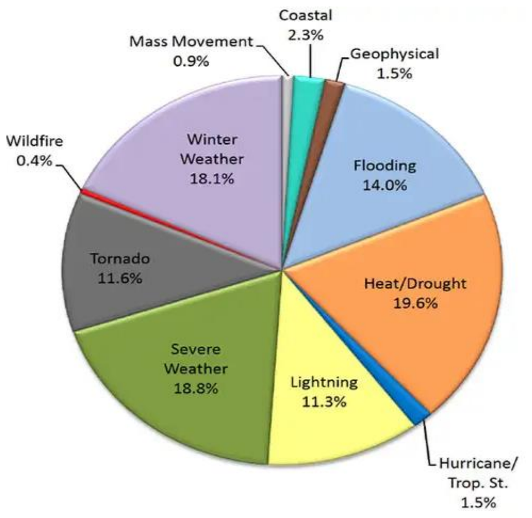


Figure 2: Natural disaster caused by weather conditions (Borden & Cutter, 2023).

- The highest concern of natural disaster is heat/Drought, as we can see 19.6% death is caused by it, this is the report of natural hazards death across the US.
- The other severe weather conditions include floods, winter weather, hurricanes and wildfires, according to geographers Kevin Borden and Susan Cutter, of the university of South Carolina in Columbia.

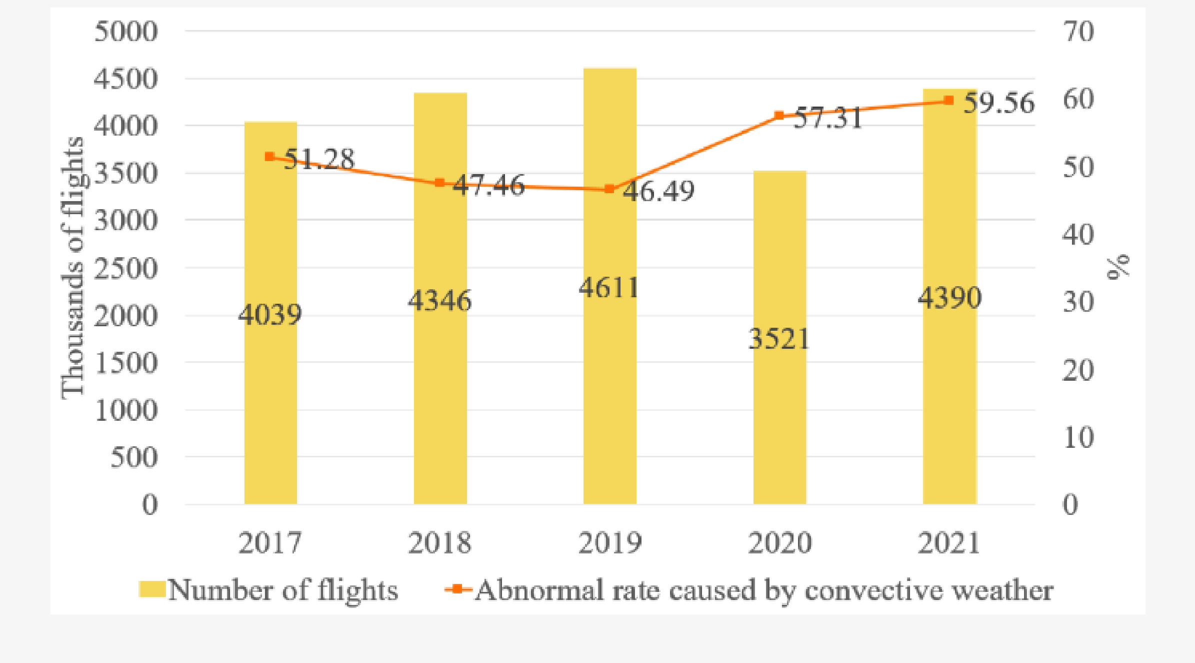


Figure 3: Number of flights and abnormal rate caused by convective (Wang, et al., 2023)

weather from 2016 to 2022.

Convective weather refers to atmospheric conditions such as rising of warm air, its cooling, thunderstorms, turbulence's and so on. The above figure shows the abnormal rate caused by convective weather.

- Knowing the weather conditions in advance helps to save the life of every person involved in that flight from pilot to passengers.
- Weather prediction or forecasting is important, since it helps to re-route the aircraft or settle for different plan.

2 Background

2.1 Research work done on the chosen topic/problem domain

2.1.1 Weather Prediction

Weather Prediction involves the ability to predict the upcoming weather conditions in advance. This is achieved by analyzing vast amount of real-time meteorological data collected worldwide, which is then integrated with numerical weather prediction models using sophisticated computing machines (Rautela & Karki, 2015). The traditional forecasting methods had limitations in terms of accuracy and efficiency, and the rise of machine learning has opened up fresh opportunities to improve the precision of weather forecasting through the automated analysis of extensive and complex datasets. Machine learning algorithms, including artificial neural networks (ANN), support vector machines (SVM), k-nearest neighbors (KNN), and random forest (RF), offer the potential to develop models that can accurately predict weather conditions. These machine learning techniques can automatically and precisely identify hidden patterns and relationships between inputs and outputs, contributing to the improvement of weather forecast accuracy (Safia, et al., 2023).

Figure 4: Weather forecast (Adobe Stock, 2023).

Weather forecasting has evolved significantly over the past century, challenging and ever-changing nature of weather condition has made forecasting a persistently difficult task. Traditionally weather forecasting relied on physical factors such as humidity, temperature, wind, and cloud. However, these models have limitations, especially in terms of accuracy and efficiency due to complexity of computational resources (Safia, et al., 2023). The emergence of machine learning has bought new opportunities to enhance forecasting accuracy by automatically analyzing large and complex datasets (Safia, et al., 2023). Different techniques are employed for prediction and analysis of weather, such as Statistical Weather Prediction, Numerical Weather Prediction and Graphical Weather Prediction.

2.1.2 Machine Learning in Weather Classification

Machine learning in weather prediction has played a pivotal role, revolutionizing the traditional forecasting methods. Unlike the conventional approaches, machine learning models have harnessed the advance algorithms and computational power, machine learning can analyze large and complex datasets related to various meteorological parameters. The models such as artificial neural networks (ANN), decision trees, and support vector machines (SVM) helps to identify the intricate patterns, correlations, relationships within the data that might be challenging for human experts to discern, and help to make accurate predictions based on historical weather data.

2.1.3 Advantages of Problem Domain

The problem domain of weather prediction offers several advantages, driven by advancement in technology. Some key advantages include:

- **Improved Accuracy:**

  Machine learning techniques, such as artificial neural networks, ensemble methods, and decision trees help in identifying patterns and relationship within the datasets. This leads to more accurate weather forecasts and helps societies to navigate the challenges of unpredictable weather conditions.

- **Early Warning Systems:**
  The natural disasters such as hurricanes, tornadoes, or floods causes discomfort in various regions, their impact can be mitigated by early warning systems by providing timely alerts to affected regions. This helps the affected region to opt for proactive measures and evacuations.

- **Agricultural Optimization:**
  In agriculture sector, weather prediction helps farmers in optimizing their agricultural practices such as crop planning, pest control, and so on which can lead to agricultural productivity and sustainability.
- **Resource Management:**
  Resource management include the water resource management sector, since weather condition directly affect the energy production (electricity), water reservoir management, and distribution, that can help in optimizing energy production.
- **Transportation Planning:**
  Transportation management industry include, aviation, road way transportation, and shipping and it is vital for the weather prediction to plan the routes, schedules, and operations. The accurate weather forecast help reducing the impact caused by weather conditions.

- **Climate Research:**

  The accumulated weather data over the period of time help to understand the trend, pattern in weather condition ultimately contributing to the study of climate variations, assessing environmental impacts, and contributing to climate research.

2.1.4 Disadvantages of Problem Domain

Even with all the advantages of weather prediction, there are still some challenges and disadvantages associated with it and they are as follows:

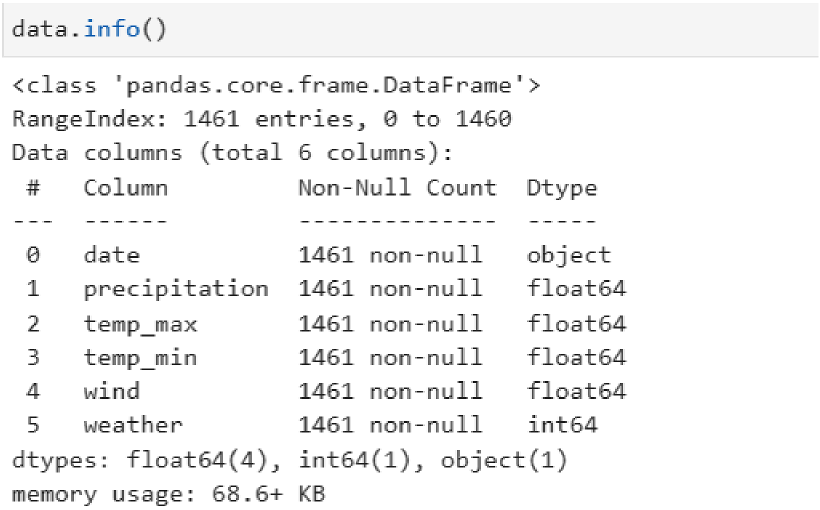
- **Extreme Weather Events:**
  Extreme weather condition includes tornadoes, floods, hurricanes, and other challenges, these events are complex to understand hence their changes are difficult to model accurately.
- **Changing Climate Patterns:**
  Even with all the advancements ad achievements, the process to model and understand the uncertainties still is a daunting task. The Earth's atmosphere is a complex thing to understand, there are various factors to keep in mind that can influence the weather at any moment.
- **Computational Intensity:**
  To predict the advance weather condition, we need substantial computational resources that has high-resolution simulations that can process vast datasets, which can be expensive, particularly in less economically developed area.
- **Data Availability:**
  Data Accuracy or accurate weather prediction depends upon the quality and availability of data. Gaps in data coverage results in inaccuracy in measurement that directly hinders the performance of prediction models.

2.1.5 Dataset

The dataset for the proposed work on weather prediction was thoroughly research in various sectors such as Kaggle, Open Weather Map, NOAA National Centers for Environmental Information (NCEI) and many more. After the thorough research the dataset model is taken from Kaggle. The selected dataset contains six columns, and needs to be prepared.

Name	**Description**	**Datatype**
Date	Date of a day on which data was collected. Such as mm/dd/yyyy.	object
Precipitation	Precipitation has various forms like rain, snow, and drizzle. It is a factor in weather pattern.	float
Max temperature	Maximum temperature of a certain region or a day.	float
Min temperature	Minimum temperature of a certain region or a day.	float
wind	The direction from which it originates, or a certain pattern in the wind.	float
weather	It defines as a type of weather in a certain region based on the dataset.	object

Table 1: Dataset Description.



```
data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1461 entries, 0 to 1460
Data columns (total 6 columns):
 #   Column         Non-Null Count  Dtype  
---  ------         --------------  -----  
 0   date           1461 non-null   object 
 1   precipitation  1461 non-null   float64
 2   temp_max       1461 non-null   float64
 3   temp_min       1461 non-null   float64
 4   wind           1461 non-null   float64
 5   weather        1461 non-null   int64  
dtypes: float64(4), int64(1), object(1)
memory usage: 68.6+ KB
```

Figure 5: Dataset Information.

2.2 Review and analysis of work in the Problem Domain

2.2.1 Research 1

Title: Weather Prediction and Classification Using Neural Networks and k-Nearest Neighbors

Authors: Rhea Mantri, Kulkarni Rakshit Raghavendra, Harshita Puri, Jhanavi Chaudhary, Kishore Bingi

Published in: https://ieeexplore.ieee.org/xpl/conhome/9528077/proceeding

Journal: https://ieeexplore.ieee.org/document/9528115/

Summary:

This research paper, focuses on classification algorithms for weather prediction, using neural networks and K-NN. The data that was feed to the model included visibility, dew point temperature, wind speed, and wind direction, and was trained using Levenberg-Marquard algorithm to predict the output parameter of one year. The aim of this paper was to develop a model to predict temperature and humidity. The raw dataset consisting of Delhi's weather from 1996 to 2017 was taken into the account for the analysis. The data set was divided into 80% and 20% for training and testing purposes. The accuracy during training was 76.30% where as in testing was 46.29% for classification's model.

Citation: (Mantri, et al., 2021)

2.2.2 Research 2

Title: Weather Prediction Using Machine Learning Algorithms

Authors: Aiswarya Shaji, A.R Amritha, V.R Rajalakshmi

Published in: https://ieeexplore.ieee.org/xpl/conhome/9862308/proceeding

Publisher: IEEE

Summary

This paper takes various machine learning model for comparison such as Random Forest, Decision Tree, Gaussian Naïve Bayes, Linear Regression with MLP classifier, and it was found to outperform more complex hybrid models in predicting weather more accurately. The analysis of Gaussian Naïve, Random Forest and MLP classifiers gives an accuracy of 51%. The dataset was collected from Kaggle and was processed using python, the factors such as temperature, humidity, wind speed and cloud cover was taken into the account. The hybrid model was suggested for the analysis, where 70% of the dataset was split for training and 30% was for testing dataset.

Citation: (Aiswarya Shaji, 2021)

2.2.3 Research 3

Title: A Comprehensive Study on Weather Forecasting using Machine Learning

Authors: Deepti Mishra, Pratibha Joshi

Published in: https://ieeexplore.ieee.org/xpl/conhome/9596064/proceeding

Publisher: IEEE

Summary

This paper focuses on prediction and analysis of weather forecasting using the algorithm, by applying the concept of linear regression and artificial neural network. In linear regression process, it shows the relationship between the variables is direct indicator used for parameter modelling which were further selected from the data. The dataset was used from Kaggle. The main aim of this paper was to apply the concept of prediction in weather forecasting, The prediction was applied in a trained dataset through supervised learning approach. In this paper, linear regression was chosen for its simplicity and effectiveness in predicting relationships between variables.

Citation: (Mishra & Joshi, 2021)

2.2.4 Research 4

Title: Real Time Weather Prediction System using Ensemble Machine Learning

Authors: D.Dhilipkumar, P.S.Yaswanth Bala, T.Yogeshwaran

Published in: Second International Conference on Augmented Intelligence and Sustainable Systems (ICAISS 2023)

Publisher: IEEE

Summary

This paper uses ensemble learning methods that combines several different algorithm models. This model, the goal of using this technique was to improve overall performance of different models and mitigating their individual weaknesses. Each algorithm's result was computed independently. The paper focuses on weather prediction using K-nearest neighbor (KNN), Gaussian Naïve Bayes (GNB), Gradient Boosting Classifier, and Support Vector Classifier (SVC). The ensemble method demonstrates superior results compared to individual algorithms, achieving better accuracy and precision. The features used to predict weather contains average temperature, humidity, moisture content, wind velocity, cloud cover, and rainfall. Evaluation metrics such as Classification Error Rate (CER), Prediction Time, Precision, Recall and specificity are used to assess the performance of the algorithms.

Citation: (D.Dhilikumar, et al., 2023)

2.2.5 Research 5

Title: Multi-Class Weather Classification Using Machine Learning Techniques

Authors: Adolf Fenyi, Micheal Asante

Journal: Journal of Theoretical and Applied Information Technology

This research paper classifies weather parameters such as sunny, cloudy, rainy, snowy and hazy using a novel algorithm. For the segmentation purpose the researcher used edge detection algorithm, and Support Vector Machine (SVM) was used for classification. In Holdout Cross Validation System 70% of the images were used for training while the remaining 30% were used for testing purposes. RGB images were used to examine the weather conditions. The worst performance was recoded in XGBoost classifier with an accuracy of 70.5% whereas Convolution Neural Network had an accuracy of 83.9%. The challenged faced by the researcher was the high energy consumption that affects the computer vision weather recognition system.

Citation: (Fenyi & Asante, 2023)

2.2.6 Summarized Review and Analysis

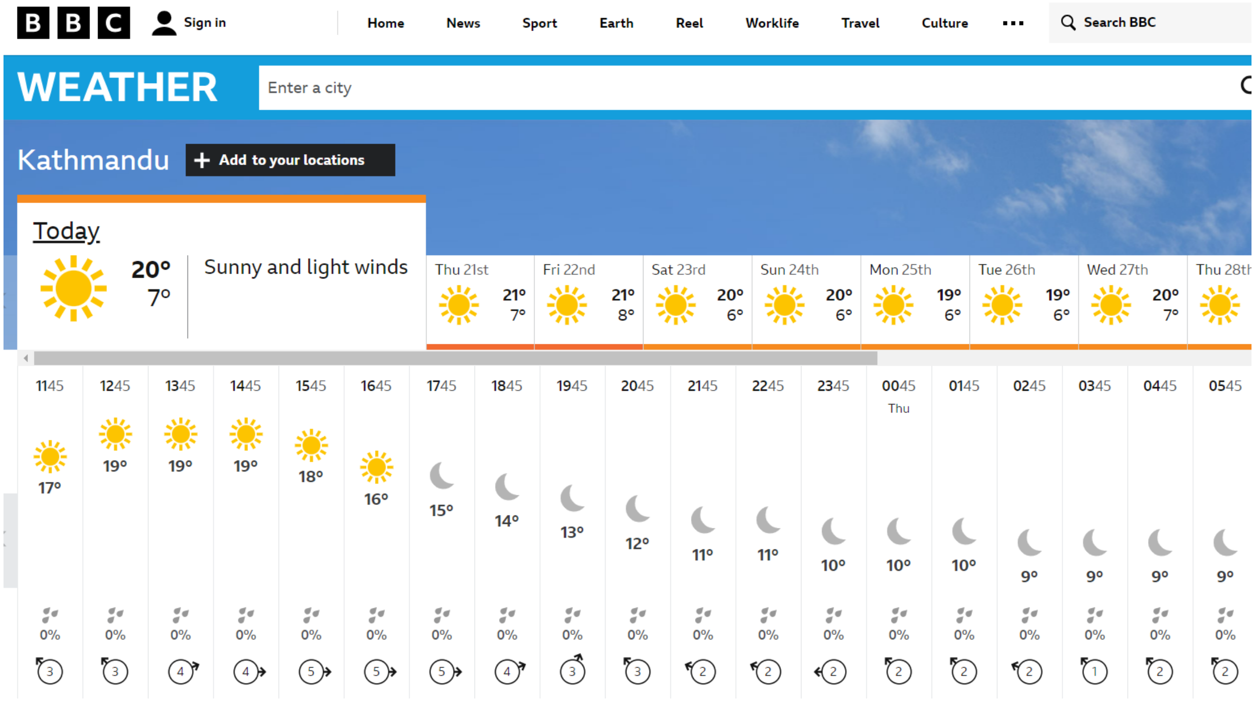
Machine learning algorithm that is suitable for weather prediction depends on various factors, including the nature of the data, the parameters of the data to be predicted, and the complexity of the problem. In the paper, a combination of algorithms is used to enhance the prediction accuracy. After the careful experimentation and validation done on the papers proved that the accuracy of Naïve bayes was highest as compared to other algorithms. Whereas, the combination of algorithm prediction was greater as compared to other algorithms.

The effectiveness of algorithm depends upon the specific characteristics of the weather data and the objective behind performing the task. Naïve Bayes is used for its computational efficiency, since it is useful for probabilistic weather predictions. Neural Networks (Deep learning) is suitable for complex, non-linear relationships in the data, that requires a large amount of data for training. K-NN was used for both regression and classification tasks. SVM was used for smaller dataset, it proves it efficiency for high-dimensional spaces.

Based on the analysis of all five-research paper, the algorithm mentioned above has its own benefits and drawbacks based on the dataset that was used and the output that the researcher wanted. It highlights the importance of using appropriate algorithms for achieving accurate results.

2.3 Review of similar systems

2.3.1 BBC Weather Channel



Today
20°
7°
Sunny and light winds

Thu 21st	Fri 22nd	Sat 23rd	Sun 24th	Mon 25th	Tue 26th	Wed 27th	Thu 28th
21° 7°	21° 8°	20° 6°	20° 6°	19° 6°	19° 6°	20° 7°	

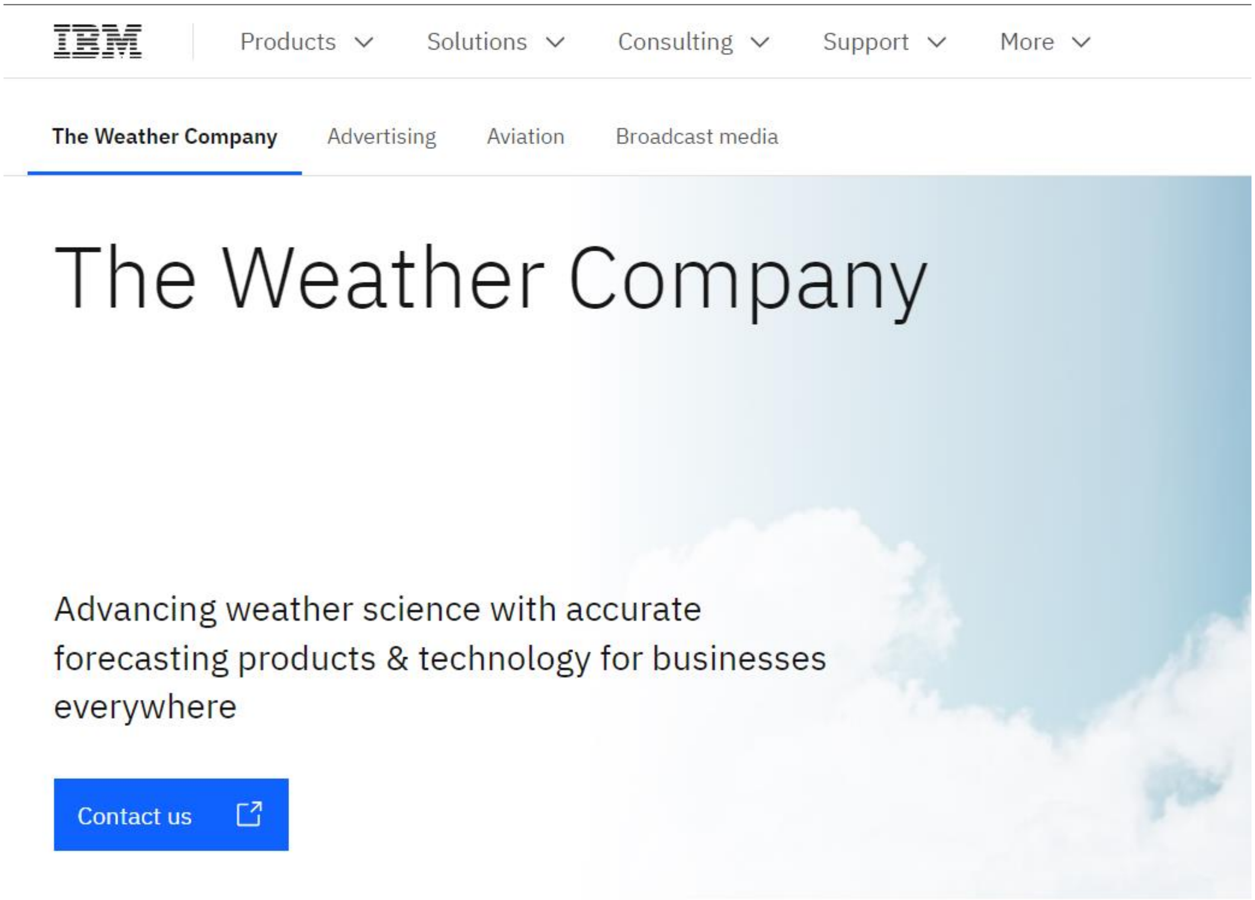
1145	1245	1345	1445	1545	1645	1745	1845	1945	2045
17°	19°	19°	19°	18°	16°	15°	14°	13°	12°
0%	0%	0%	0%	0%	0%	0%	0%	0%	0%
3	3	4	4	5	5	5	4	3	3

2145	2245	2345	0045 Thu	0145	0245	0345	0445	0545
11°	11°	10°	10°	10°	9°	9°	9°	9°
0%	0%	0%	0%	0%	0%	0%	0%	0%
2	2	2	2	2	2	1	2	2

Figure 6: Weather forecasting used in BBC (BBC Weather, 2023).

BBC is a media organization which relies on weather forecasting to provide accurate and timely weather prediction to its audience. BBC is one of many examples of media outlets that uses weather forecasting.

2.3.2 IBM



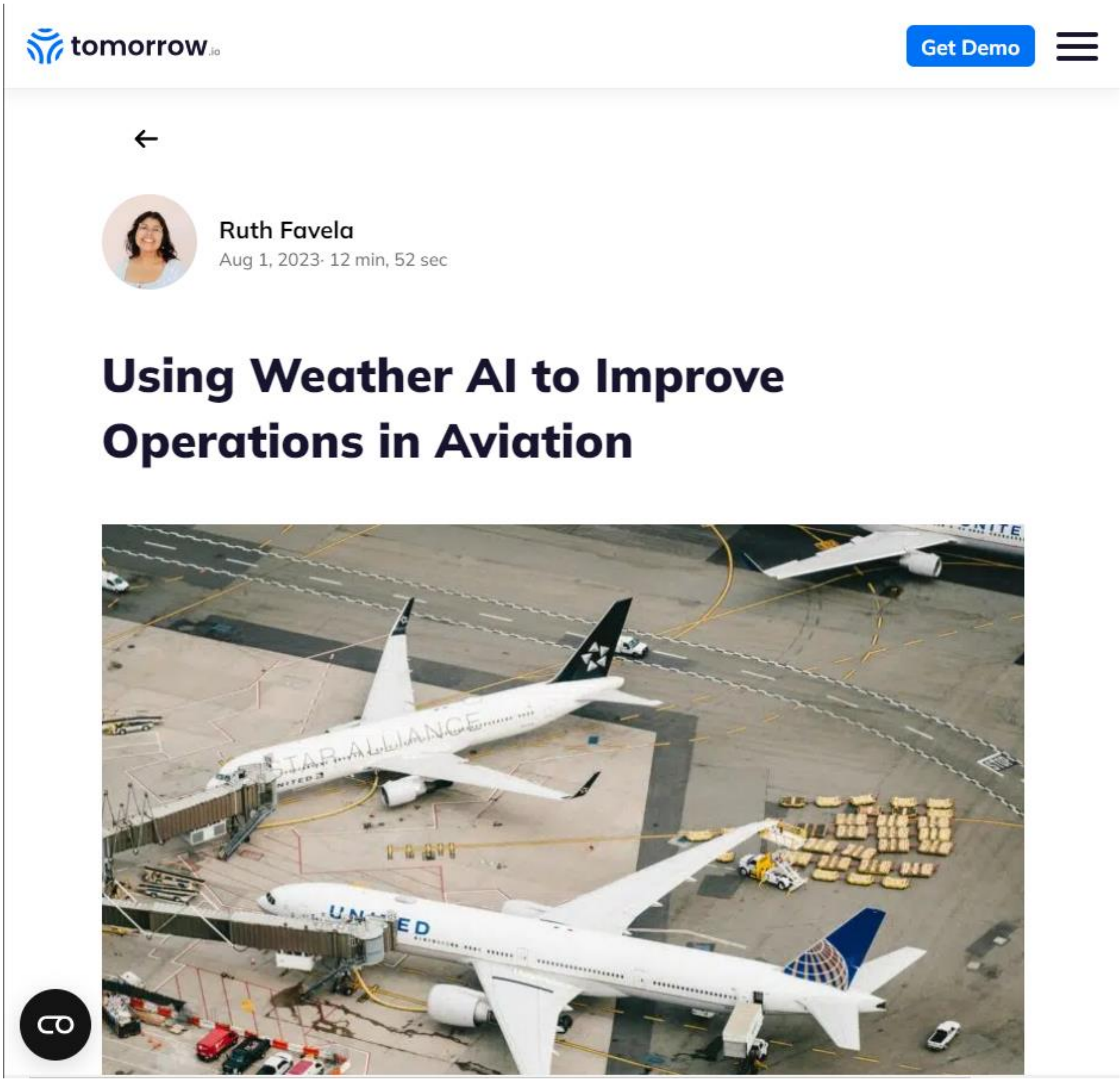
The Weather Company

Advancing weather science with accurate forecasting products & technology for businesses everywhere

Figure 7: IBM (the weather company) (IBM, 2023).

IBM Business, is the world’s leading weather provider, helping people and businesses make more informed decisions and take action in the face of weather (IBM, 2023).

2.3.3 Aviation Industry



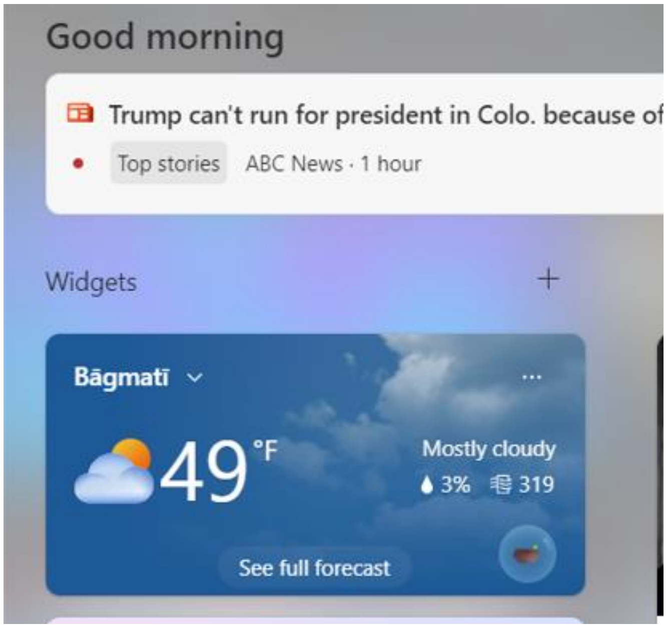
Ruth Favela
Aug 1, 2023· 12 min, 52 sec

Using Weather AI to Improve Operations in Aviation

Figure 8: Aviation industry using AI for weather prediction (Favela, 2023).

The aviation industry using AI to predict weather forecasting, to deliver accurate, and insight delivery for the aviation industry. This data driven approach help in predicting accurate weather which is important for flight safety, passenger experience and so on.

2.3.4 Personal Desktop



Good morning

Trump can't run for president in Colo. because of

Top stories ABC News · 1 hour

Figure 9: In-bult system in desktop.

Desktop has in build feature that shows the weather based on the region we are in. This is the example from mine own laptop that has weather forecasts.

3 Solution

3.1 Proposed Approach to solving the problem

A supervised learning approach was followed for solving the problem. The proposed approach in solving the problem of weather prediction is to use machine learning algorithms that works well on the selected dataset to predict the accurate weather condition. After all the research done from the research paper and websites, Naïve Bayes and K-nearest neighbors seem to provide more accuracy than other algorithms. Support Vector Machine (SVM) works well on small to medium-sized dataset and is effective in high dimensional spaces. There these three were selected for further work.

3.1.1 Elaboration of the AI algorithm used

3.1.1.1 Naïve Bayes

The Naïve Bayes classifier is a probabilistic classification method based on Bayes' theorem. It is particularly effective for classification tasks, it is said to be "naïve" given the categorization because it believes the attributes are separated from one another (D, et al., 2023). It has shown its uses in different applications, including spam filtering, text classification, and weather prediction also. Here's an overview of Naïve Bayes being used in weather prediction:

3.1.1.1.1 Overview of Naïve Bayes Classifier:

- **Bayes Theorem:**

  The algorithm in Bayes theorem, that calculates the probability of a hypothesis given the evidence. In the context of classification:

$$p\left(\frac{y}{x}\right) = \frac{p\left(\frac{x}{y}\right) * P(y)}{P(x)}.$$

Equation 1: Bayes theorem

Where,

P(y/x) is Posterior probability: The probability of hypothesis Y on the observed event x.

P(x/y) is Likelihood probability: The probability of the evidence given that the probability of hypothesis is true.

P(y) is Prior Probability: The probability of hypothesis before observing the evidence.

P(x) is Marginal Probability: The probability of evidence (Java T point).

- **Assumption of Feature Independence:**

The Naïve Bayes algorithm functions under the basic assumption that, the given class label, and the features that describes an instance are conditionally independent. This assumption being naïve actually contributes to the algorithm's computational efficiency.

3.1.1.2 K-nearest neighbors (KNN)

The KNN algorithm is a supervised machine learning algorithm that is used for both classification and regression problem. It identifies the new data point based on the available data, which simply means when the new data is introduced it can be easily classifies into a well-suited category by using K-NN algorithm (Java T point, 2023).

3.1.1.2.1 Overview of K-nearest neighbors (KNN) algorithm:

- **KNN Equation:**

$$d(x, y) = \sqrt{\sum_{i=1}^{n} (y_i - x_i)^2}$$

Equation 2: Euclidean distance equation.

- KNN an instance-based learning: KNN model does not construct a distinct model in the training phase; rather, it memorizes an entire training dataset.
- Distance Metric: KNN uses Euclidean distance mostly to measure the similarity between the data points.
- Majority Voting (Classification): The class of a data point is determined by majority voting among its k-nearest neighbors.

3.1.1.2.2 Application in Weather Prediction:

- Nearest Neighbors: KNN feature is to select the k-nearest data points measured by the distance.
- Feature Selection: Helps in identifying the relevant weather features such as wind, temperature, condition and so on.

- Distance Calculation: Helps in distance calculation of current weather conditions and all historical data points.
- Prediction: Its prediction is based on the majority voting of the weather condition.

3.1.1.3 Support Vector Machines (SVM)

SVM is mostly used in weather prediction, since it has the ability to handle complex datasets and nonlinear relationships. It can handle the multidimensional nature of weather data, considering the factors like temperature, wind, humidity and so on. SVM is not only used in weather prediction, it is also applied in image classification, spam detection, handwriting identification, and so on. Based on the nature of decision boundary, SVM is classified in two groups:

- **Linear SVM**: Linear SVMs is used to separate the data points of different classes using linear decision boundary. It means when a straight line in 2D or a hyperplane in higher dimensions can entirely divide the data points into their respective classes (GeeksforGeeks).
- **Non-linear SVM:** Non-linear SVM can be used to classify data when it cannot be separated into two classes by a straight line in the case of 2D (GeeksforGeeks, 2023). To separate the data linearly, the original input data is transformed by kernel functions into a higher-dimensional feature space.

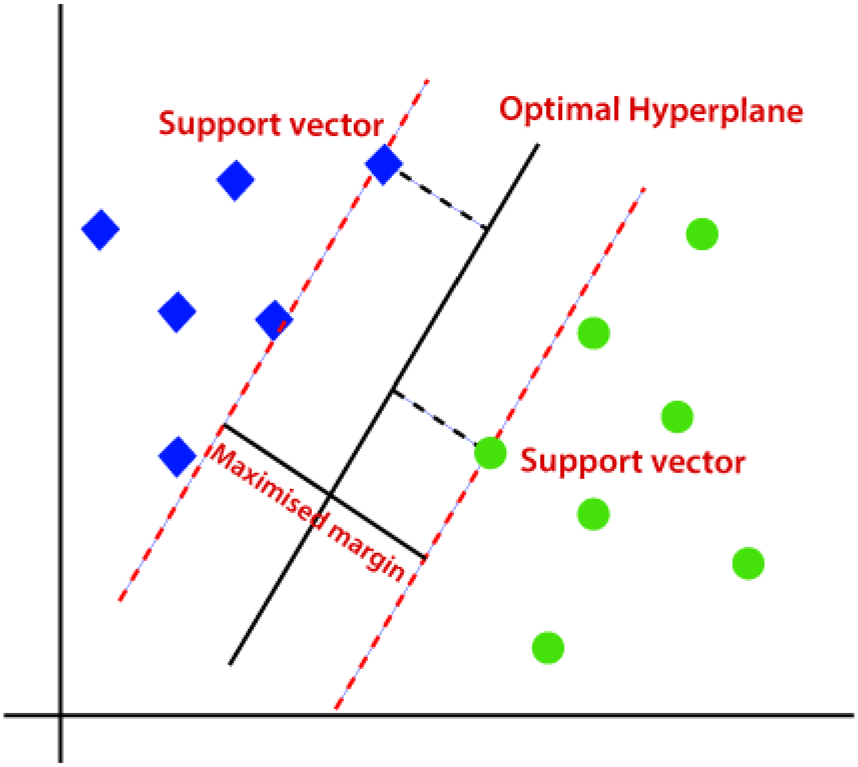


Figure 10: SVM example (Java T point, 2023).

3.1.2 Implementation methodology of used Algorithm in the system

The implementation process after applying the chosen algorithm as a methodology involves several steps, they are described below:

Data Collection:

The data is gathered from Kaggle. The data set contains the information related to date, maximum temperature, minimum temperature, wind, and weather.

Data Preprocessing:

The pre-processing is an essential step where we handle missing data or encode categorical variables. After that we split the dataset into train and test datasets.

Implementation of Algorithms

Import the Naïve Bayes algorithm, KNN algorithm, and SVM algorithm. After that we have to train the model using the training and testing dataset.

Model Evaluation:

Model Evaluation involves making prediction based on the test set. The model evaluation is made using the metrics like accuracy, precision, recall, and F-1 score.

3.2 Pseudocode

The pseudocode for the proposed solution is as follows:

```
START
        IMPORT libraries
        IMPORT dataset
        ANALYSE dataset
        GET dataset INTO data shape
        PRE-PROCESS dataset
                REMOVE null-values
                CONVERT data into numerical value
                CONVERT date format
        CHOOSE a Classification Algorithm
        SPLIT dataset into train and test data
        TRAIN the model by fitting the train dataset
        TEST the model with test dataset
        VALIDATE the model
        CALCULATE accuracy, precision, F1-score, and re-call score of the model.
        INTERPRET Result
END
```

3.3 Flowchart

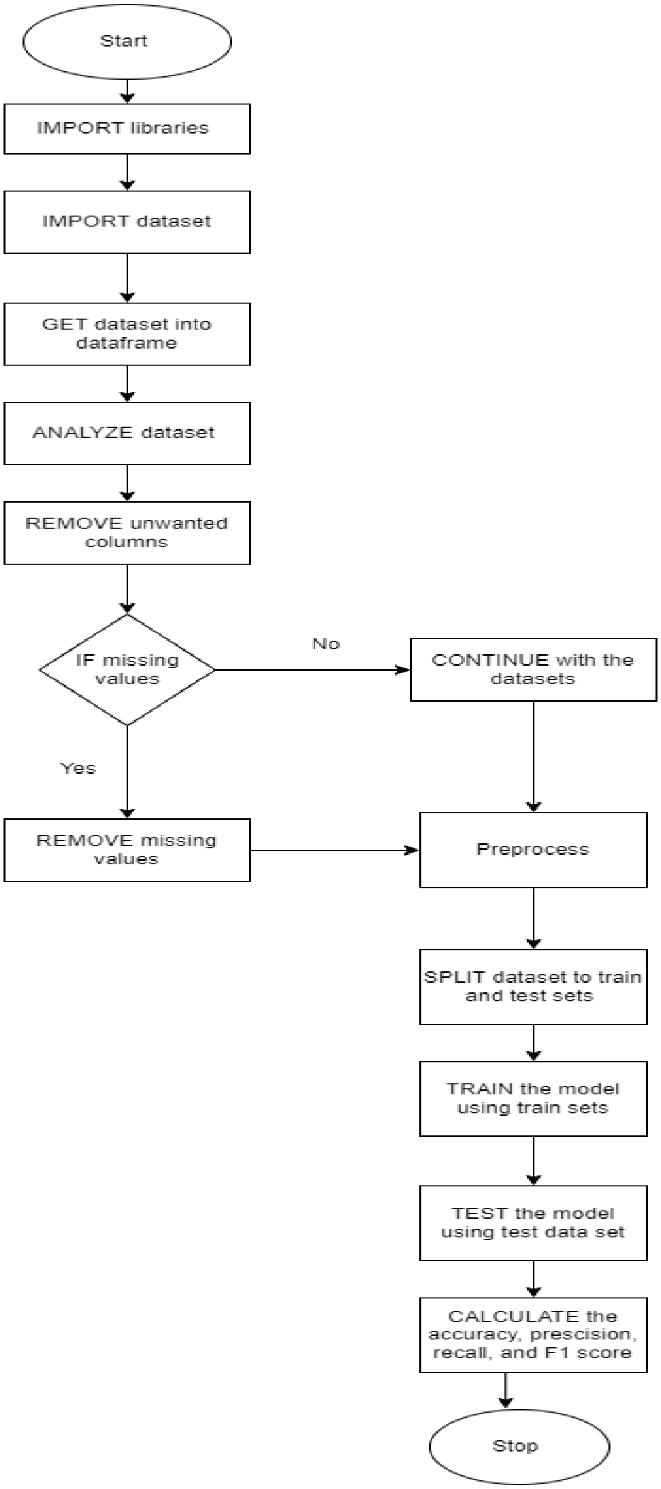

Start
IMPORT libraries
IMPORT dataset
GET dataset into dataframe
ANALYZE dataset
REMOVE unwanted columns
IF missing values
No
CONTINUE with the datasets
Yes
REMOVE missing values
Preprocess
SPLIT dataset to train and test sets
TRAIN the model using train sets
TEST the model using test data set
CALCULATE the accuracy, prescision, recall, and F1 score
Stop

Figure 11: Flowchart

3.4 Development process

3.4.1 Explanation of used tools and technologies

3.4.1.1 Anaconda Navigator

Anaconda Navigator is a graphical user interface (GUI) that is integrated into the widely used Anaconda distribution, it enables to work with packages and environments without needing to type conda comments in a terminal window (Anaconda Navigator, 2018). It can help us to access popular integrated development environments (IDEs) like Jupyter Notebook, JupyterLab and so on.

3.4.1.2 Jupyter Notebook

Jupyter Notebook is an application for generating notebooks that fall under the Project Jupyter area. Jupyter Notebook, which is built upon the capabilities of the computational notebook format, provides interactive, real-time alternatives for code prototyping and explanation, data exploration and visualization, and idea sharing (Jupyter Notebook, 2015).

3.4.1.3 Microsoft Excel

Excel is a spreadsheet application developed by Microsoft, which is classified as an Office product division designed to cater to business needs. Spreadsheet data can be formatted, organized, and calculated using Microsoft Excel (Gillis, 2024). Data analysts and other users of excel can organize data, make information easier to organize and check weather the data has been added or changed.

3.4.1.4 Python

Python, which is described as a dynamic, high-level, object-oriented programming language that is interpreted, displays itself to be a viable choice throughout the process of developing a weather prediction system. Moreover, Python functions efficiently as a bridging language or scripting language, facilitating the seamless integration of numerous pre-existing components within the system.

3.4.2 Libraries used

3.4.2.1 Explanation of used libraries

- **NumPy:**

  NumPy, being a powerful numerical computing library in Python, helps in handling the mathematical computations involved in weather prediction algorithms, and it also uses NumPy for multidimensional array objects, which help in the representation of data such as temperature, precipitation, and so on.

- **Pandas**

  Pandas introduces the DataFrame, which helps to present the data in a structured format. It can also be used for handling missing values and removing duplicate values. It also integrates with data visualization libraries such as Matplotlib and Seaborn to analyze and visualize weather data patterns and trends.

- **Matplotlib**

  Matplotlib is a Python library that is used for data visualization. It is used for plotting options such as line plots, scatter plots, bar charts, and histograms.

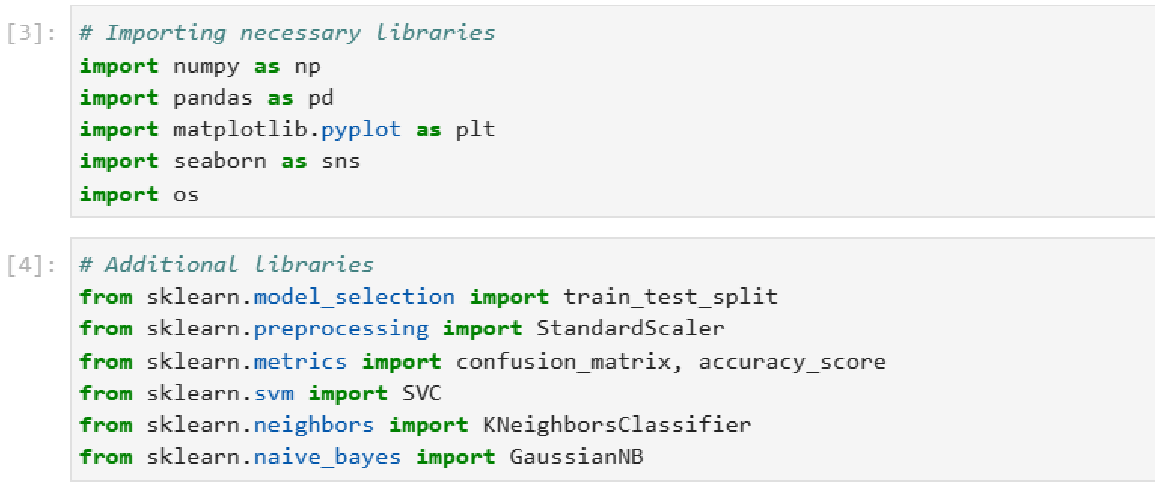
- **Seaborn**

  Seaborn is constructed on top of Matplotlib, which facilitates statistical data visualization. It also works with Pandas DataFrames to store and manipulate weather data.

3.4.3 Explanation of development

3.4.3.1 Importing Libraries

The first step in solution is to import the necessary libraries.

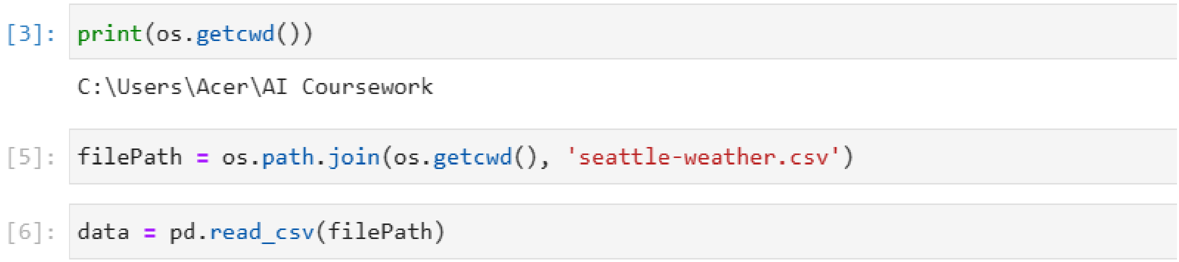


```
[3]: # Importing necessary libraries
     import numpy as np
     import pandas as pd
     import matplotlib.pyplot as plt
     import seaborn as sns
     import os

[4]: # Additional libraries
     from sklearn.model_selection import train_test_split
     from sklearn.preprocessing import StandardScaler
     from sklearn.metrics import confusion_matrix, accuracy_score
     from sklearn.svm import SVC
     from sklearn.neighbors import KNeighborsClassifier
     from sklearn.naive_bayes import GaussianNB
```

Figure 12: Importing Libraries.

3.4.3.2 Importing the dataset

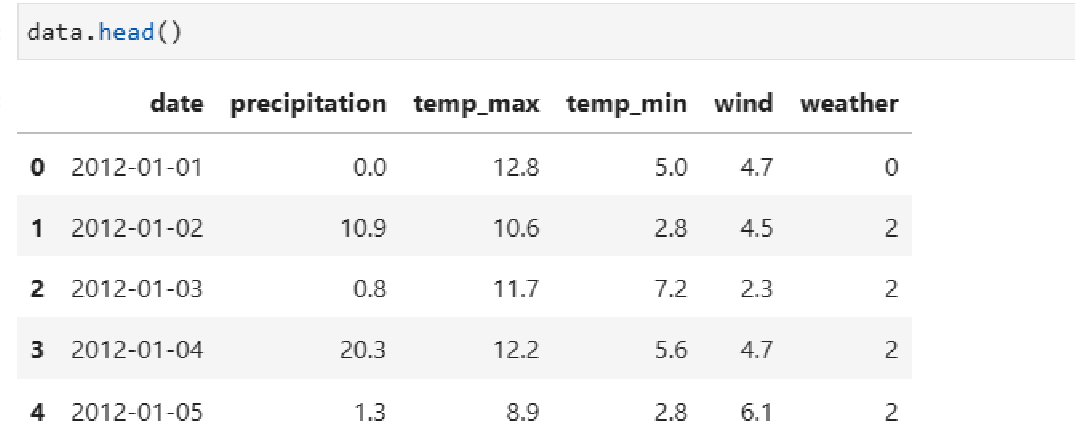
The data set was loaded into a Pandas DataFrame from a csv file.



```
[3]: print(os.getcwd())

     C:\Users\Acer\AI Coursework

[5]: filePath = os.path.join(os.getcwd(), 'seattle-weather.csv')

[6]: data = pd.read_csv(filePath)
```

Figure 13: Importing dataset.

3.4.3.3 Pre-Processing the data



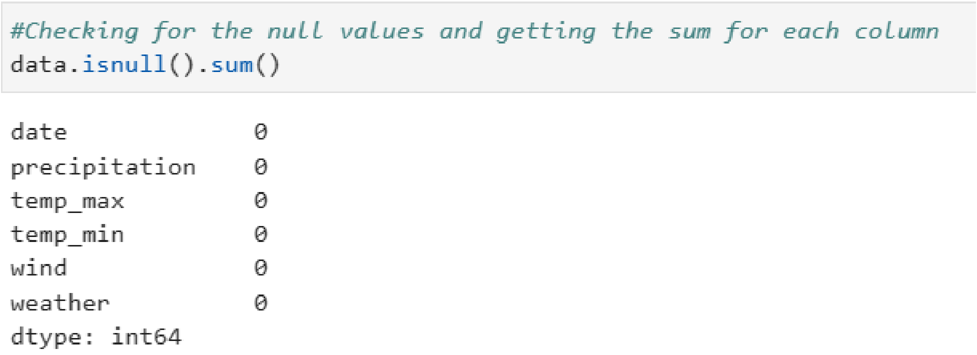
```
data.head()
```

	date	precipitation	temp_max	temp_min	wind	weather
0	2012-01-01	0.0	12.8	5.0	4.7	0
1	2012-01-02	10.9	10.6	2.8	4.5	2
2	2012-01-03	0.8	11.7	7.2	2.3	2
3	2012-01-04	20.3	12.2	5.6	4.7	2
4	2012-01-05	1.3	8.9	2.8	6.1	2

Figure 14: Dataset.

This is the visual representation of the raw data.

3.4.3.4 Checking for the null values



```
#Checking for the null values and getting the sum for each column
data.isnull().sum()
```

```
date             0
precipitation    0
temp_max         0
temp_min         0
wind             0
weather          0
dtype: int64
```

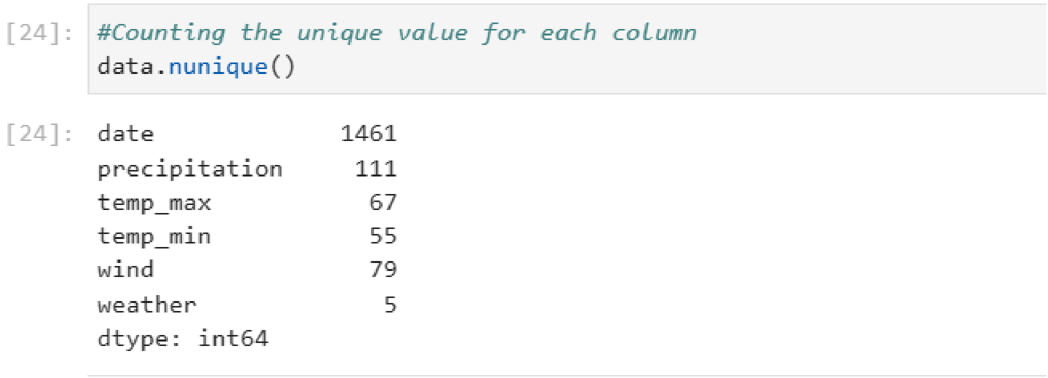
Figure 15: Checking for the null values.

The result shows that there are no null values. Since the count of missing values in each column is '0'.

- If the result had shown the value '1' or '2' instead of '0'. We need to remove ant row that contained at least one missing values. We would use 'data.dropna()' to remove rows with missing values.

3.4.3.5 Checking for the unique values

data.nunique() calculates the unique values in each column, it us done to know the diversity of values in our dataset.

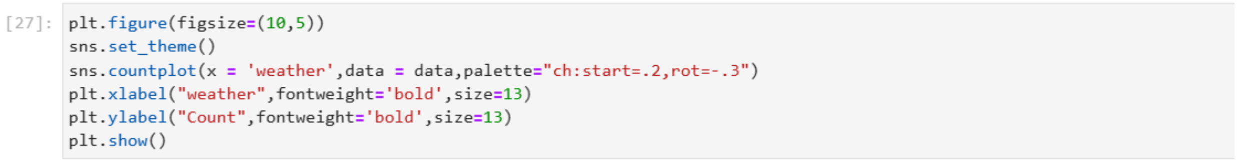


```
[24]: #Counting the unique value for each column
      data.nunique()
```

```
[24]: date             1461
      precipitation     111
      temp_max           67
      temp_min           55
      wind               79
      weather             5
      dtype: int64
```

Figure 16: Unique values

3.4.3.6 Bar Chart



```
[27]: plt.figure(figsize=(10,5))
      sns.set_theme()
      sns.countplot(x = 'weather',data = data,palette="ch:start=.2,rot=-.3")
      plt.xlabel("weather",fontweight='bold',size=13)
      plt.ylabel("Count",fontweight='bold',size=13)
      plt.show()
```

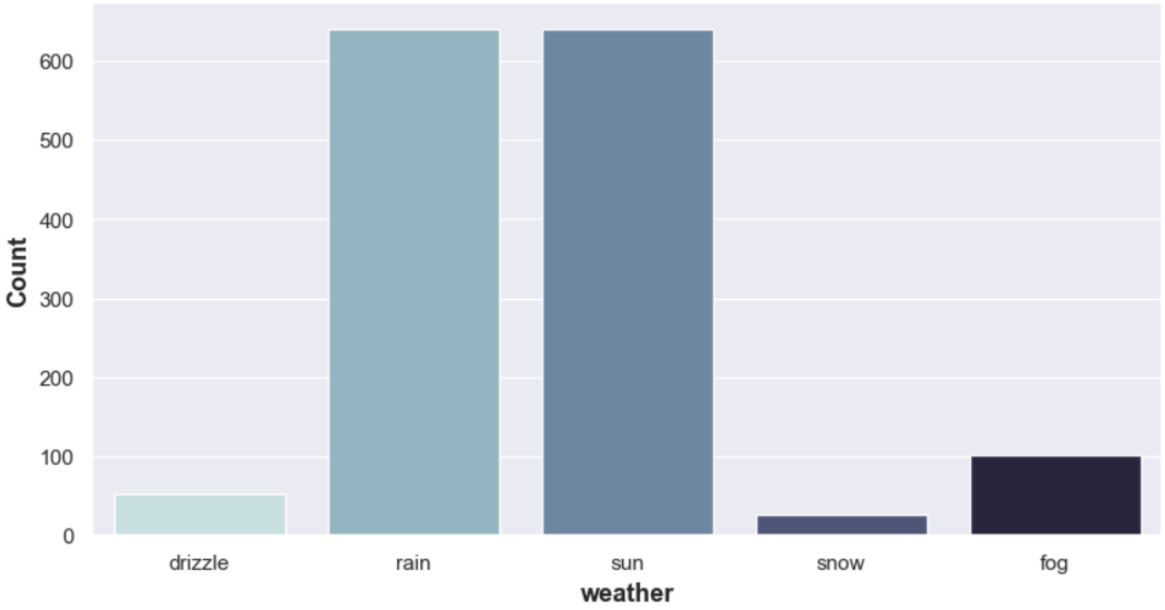
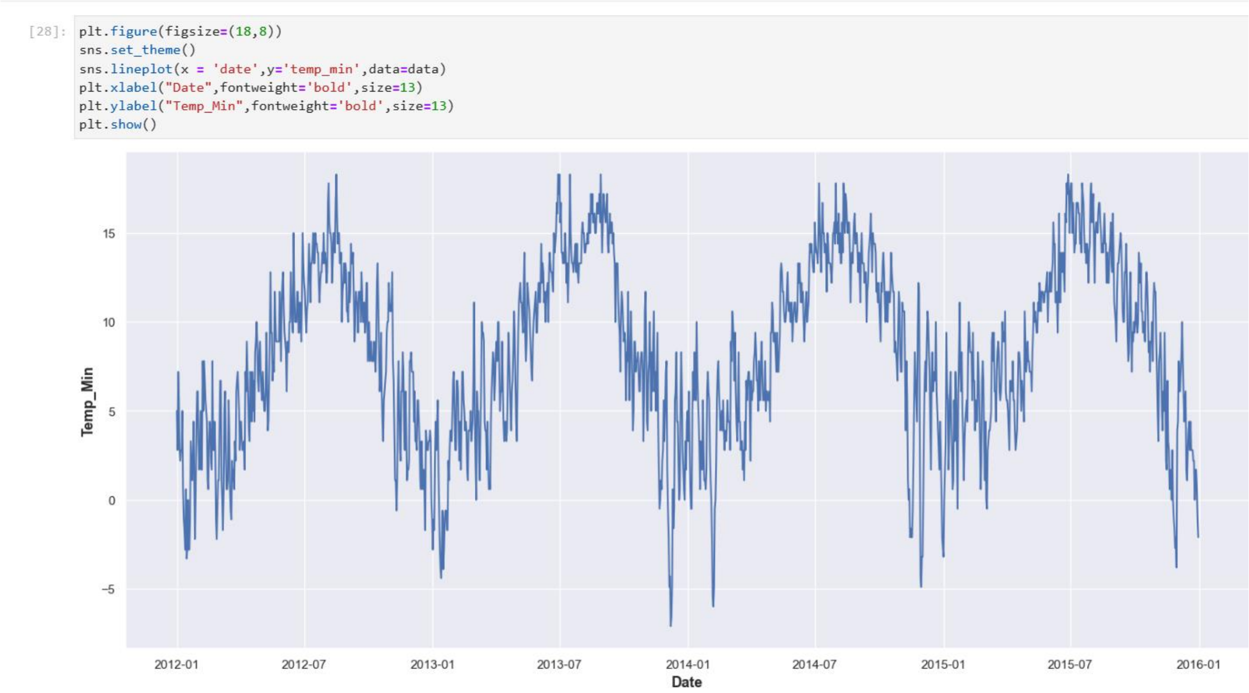


Figure 17: Weather Count

Using the count plot function from the seaborn library to plot a bar that displays the count of different categories in the weather column.

3.4.3.7 Line plot



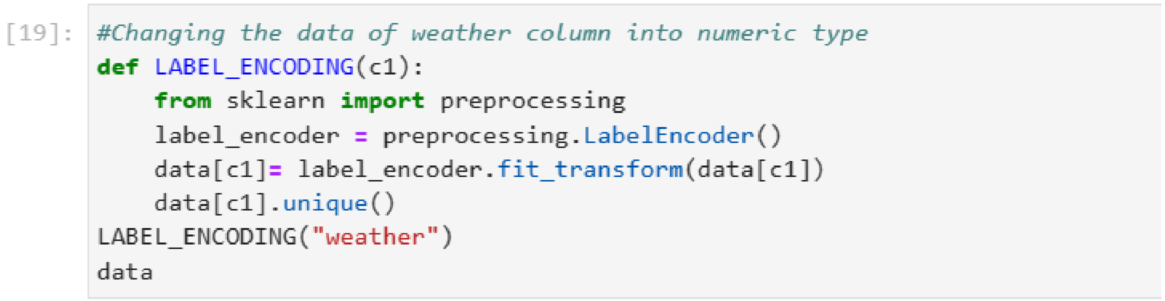
```
[28]: plt.figure(figsize=(18,8))
      sns.set_theme()
      sns.lineplot(x = 'date',y='temp_min',data=data)
      plt.xlabel("Date",fontweight='bold',size=13)
      plt.ylabel("Temp_Min",fontweight='bold',size=13)
      plt.show()
```

Figure 18: Line plot.

The result shows the line chart with the variation of the minimum temperature over different dates. It is useful to identify trends, patterns, and fluctuation in the temperature data across different dates.

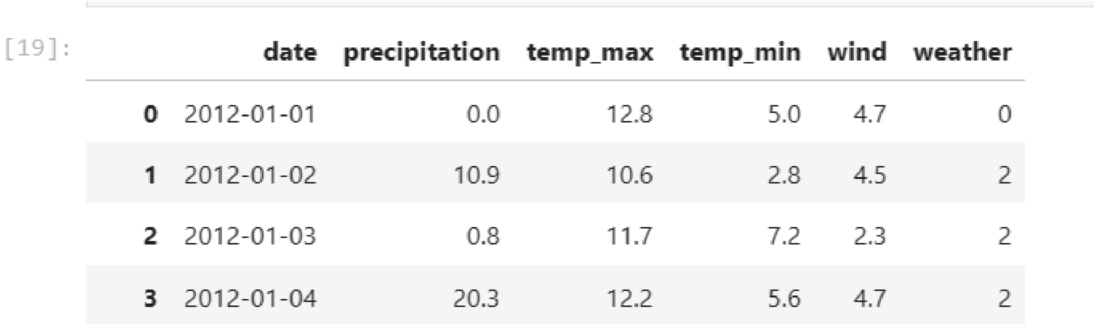
3.4.3.8 Changing the data type

In scikit-learn, the preprocessing module helps to transform and scale data before feeding it into machine learning models. We have used the ‘LabelEncoder’ to convert the label of the weather column into numerical format.



```
[19]: #Changing the data of weather column into numeric type
      def LABEL_ENCODING(c1):
          from sklearn import preprocessing
          label_encoder = preprocessing.LabelEncoder()
          data[c1]= label_encoder.fit_transform(data[c1])
          data[c1].unique()
      LABEL_ENCODING("weather")
      data
```

Figure 19: Changing the data type.



[19]:

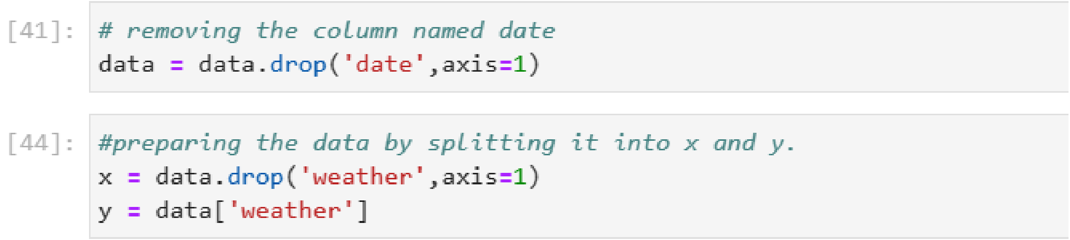
	date	precipitation	temp_max	temp_min	wind	weather
0	2012-01-01	0.0	12.8	5.0	4.7	0
1	2012-01-02	10.9	10.6	2.8	4.5	2
2	2012-01-03	0.8	11.7	7.2	2.3	2
3	2012-01-04	20.3	12.2	5.6	4.7	2

Figure 20: Changed Data

As shown in the above figure, the weather data has been converted into a numerical data type.

3.4.3.9 Preparing the data

To begin, we eliminate the column labeled 'date' by use the drop method. After that, we split the DataFrame into two distinct components, namely 'x' and 'y', to preprocess the data for a machine learning algorithm. In column 'x', we have all the data except for the weather column. Within the dataset labeled 'y', the only available column is weather. This is because we specifically chose to include just weather data.

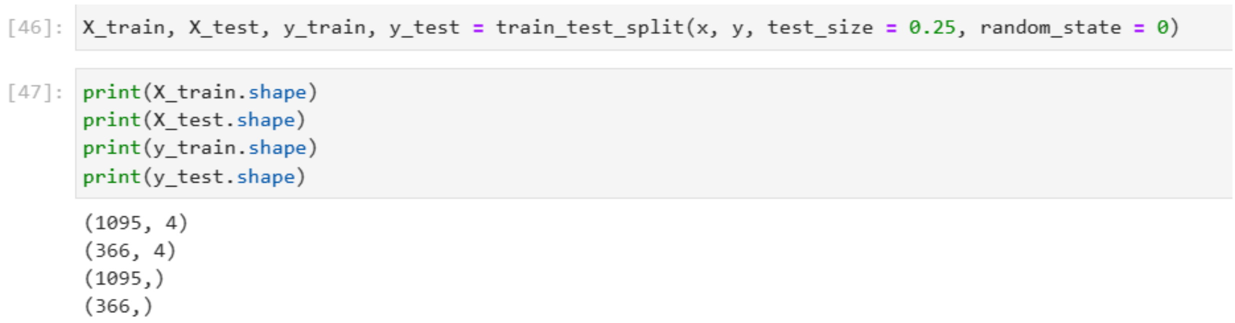


```
[41]: # removing the column named date
      data = data.drop('date',axis=1)
```

```
[44]: #preparing the data by splitting it into x and y.
      x = data.drop('weather',axis=1)
      y = data['weather']
```

Figure 21: Split the data into x and y.

3.4.3.10 Splitting the dataset into train and test

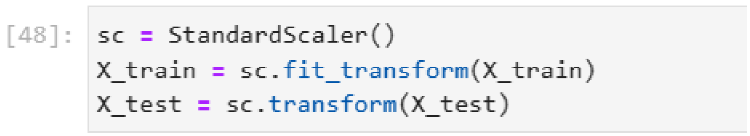


```
[46]: X_train, X_test, y_train, y_test = train_test_split(x, y, test_size = 0.25, random_state = 0)

[47]: print(X_train.shape)
      print(X_test.shape)
      print(y_train.shape)
      print(y_test.shape)

      (1095, 4)
      (366, 4)
      (1095,)
      (366,)
```

Figure 22: Splitting into train and test dataset.

We have previously imported the train_test_split function from the model_selection module within scikit-learn. The function facilitates the division of the data into separate training and testing sets. 25% of the data will be used for testing, and the remaining 75% will be used for training. The random state is set to zero to ensure that we get the same split again.

3.4.3.11 Feature Scale

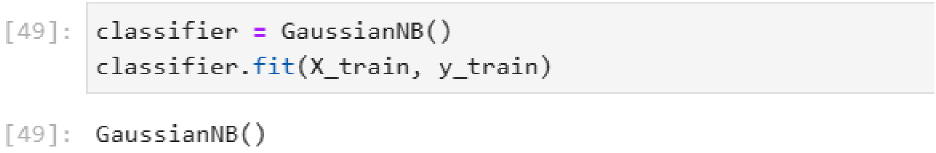


```
[48]: sc = StandardScaler()
      X_train = sc.fit_transform(X_train)
      X_test = sc.transform(X_test)
```

Figure 23: Feature Scale

We have already utilized the 'StandardScaler' module to normalize the characteristics in both our training and testing datasets. This feature is utilized to eliminate the average value and normalize it to have a variance of a unit. It is utilized for algorithms that are sensitive to the scale of the input features, such as K-Nearest Neighbors (KNN) and Support Vector Machines (SVM) in our specific scenario.

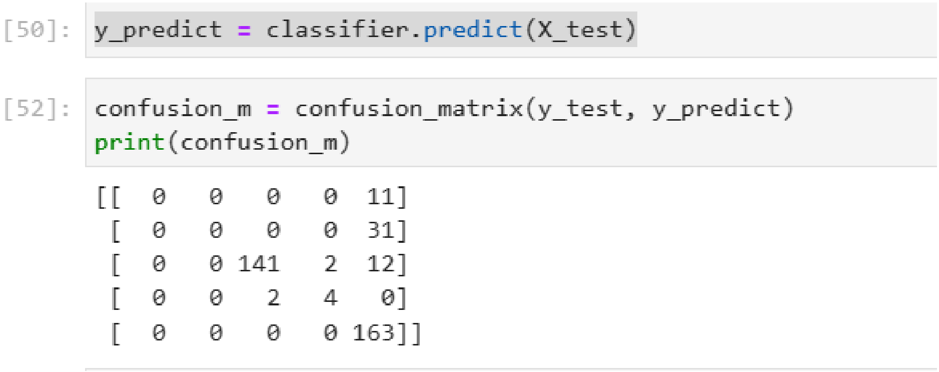
3.4.3.12 Training the dataset (Naïve Bayes)



```
[49]: classifier = GaussianNB()
      classifier.fit(X_train, y_train)

[49]: GaussianNB()
```

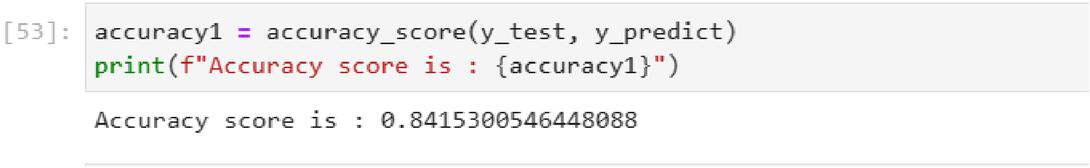
Figure 24: Gaussian NB instance.

Creating an instance of the 'GaussianNB' class.



```
[50]: y_predict = classifier.predict(X_test)

[52]: confusion_m = confusion_matrix(y_test, y_predict)
      print(confusion_m)

      [[  0   0   0   0  11]
       [  0   0   0   0  31]
       [  0   0 141   2  12]
       [  0   0   2   4   0]
       [  0   0   0   0 163]]
```

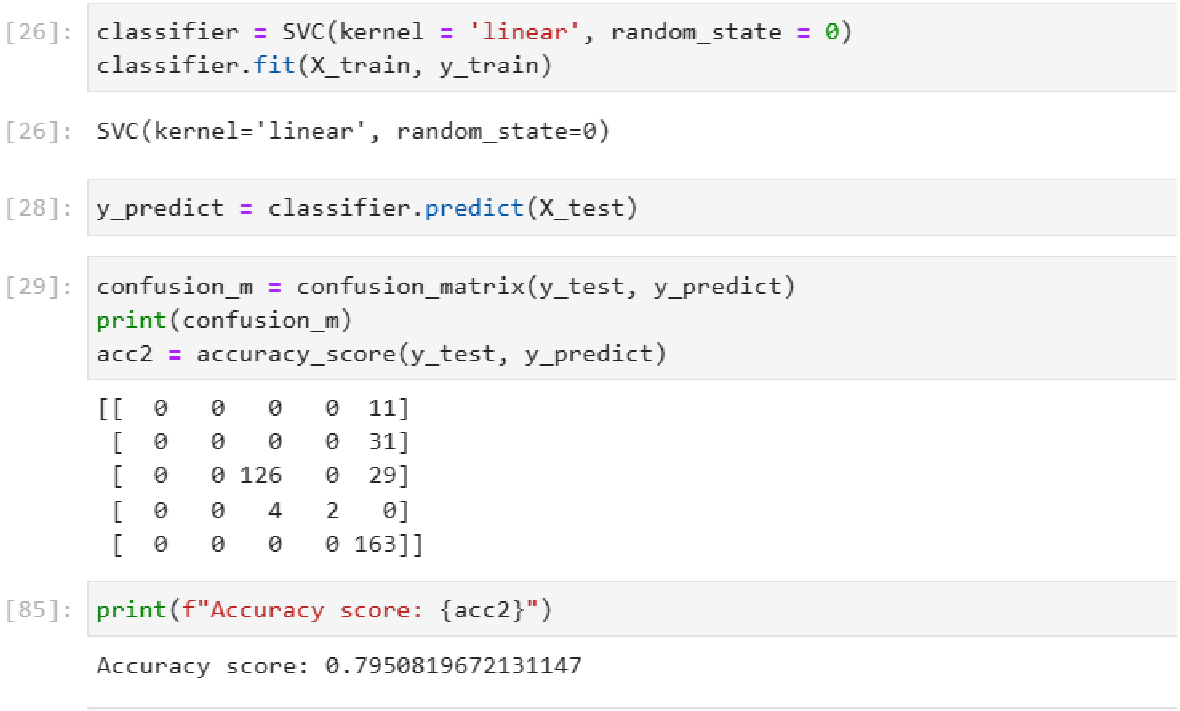
Figure 25: Confusion matrix



```
[53]: accuracy1 = accuracy_score(y_test, y_predict)
      print(f"Accuracy score is : {accuracy1}")

      Accuracy score is : 0.8415300546448088
```

Figure 26: Accuracy score

3.4.3.13 Test set result (SVM)



```
[26]: classifier = SVC(kernel = 'linear', random_state = 0)
      classifier.fit(X_train, y_train)

[26]: SVC(kernel='linear', random_state=0)

[28]: y_predict = classifier.predict(X_test)

[29]: confusion_m = confusion_matrix(y_test, y_predict)
      print(confusion_m)
      acc2 = accuracy_score(y_test, y_predict)

      [[  0   0   0   0  11]
       [  0   0   0   0  31]
       [  0   0 126   0  29]
       [  0   0   4   2   0]
       [  0   0   0   0 163]]

[85]: print(f"Accuracy score: {acc2}")

      Accuracy score: 0.7950819672131147
```

Figure 27: Without hyperparameter tuning.

- Accuracy Calculation:

  Accuracy is a common metric used to assess the performance of classification models. It is a measure of the model's ability to properly predict the class labels of the instances in the dataset.

  The accuracy formula is given by:

$$\text{Accuracy} = \frac{\text{Number of correct Predictions}}{\text{Total Number of Predictions}} \times 100$$

Equation 3: Accuracy Formula

- Precision calculation

  Precision is a classification metric that evaluates the accuracy of a model's positive predictions. It calculates the percentage of real positive predictions among all instances shown as positive.

  The precision formula is given by:

$$\text{Precision} = \frac{True\ Positives}{True\ Positives + False\ Positives}$$

Equation 4: Precision Calculation

- Recall calculation

  Recall measures the proportion of true positives predictions among all instances that are actually positive.

  The recall formula is given by:

$$\text{Recall} = \frac{True\ Positives}{True\ Positives + False\ Negatives}$$

Equation 5: Recall calculation.

- F1 score calculation

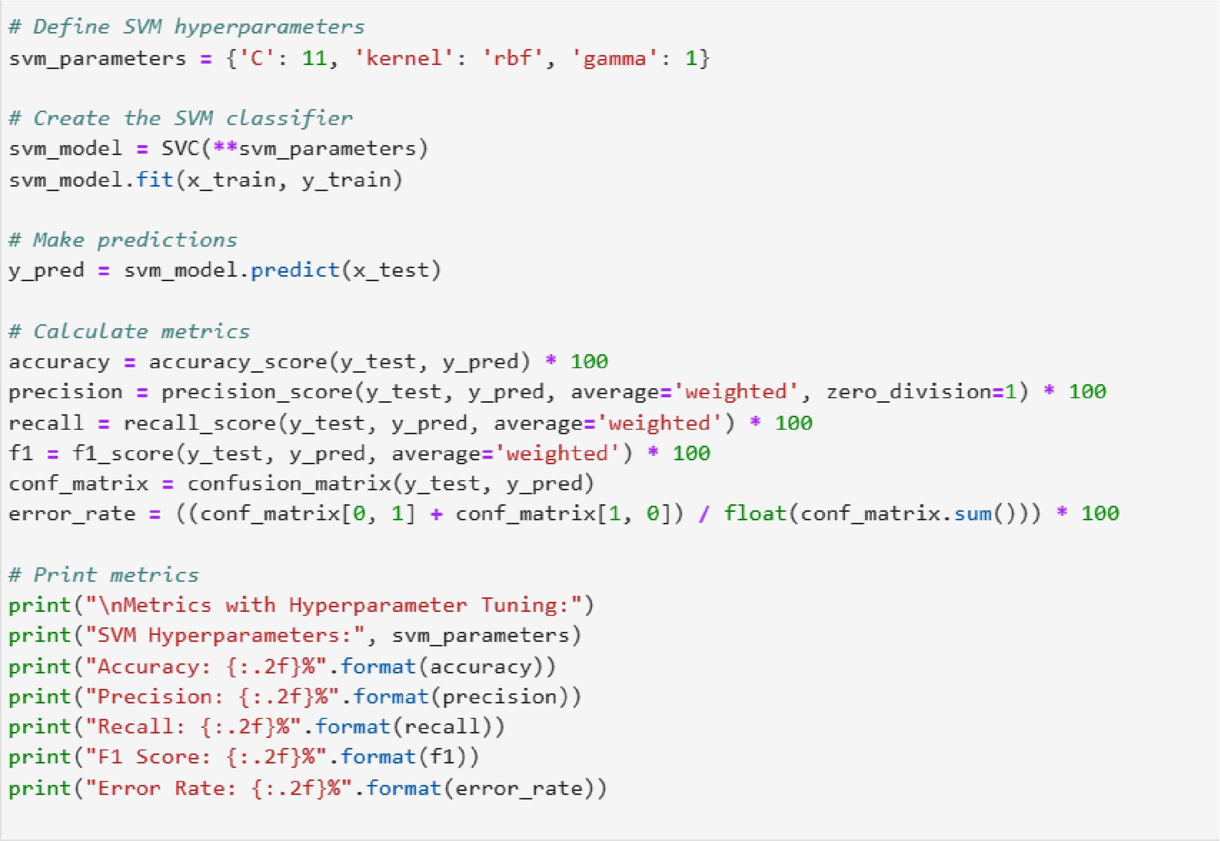
  The F1 score calculated the measure of a model’s accuracy by combining the precision and recall mean value. Higher F1 score indicated better overall performance.

  The F1 Score is calculated using the following formula:

$$\text{F1 Score} = 2\times \frac{Precision*Recall}{Precision+Recall}$$

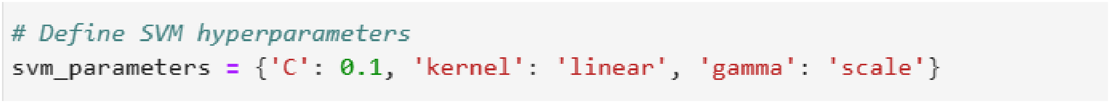
Equation 6: Recall Calculation

3.4.3.14 SVM hyperparameter tuning



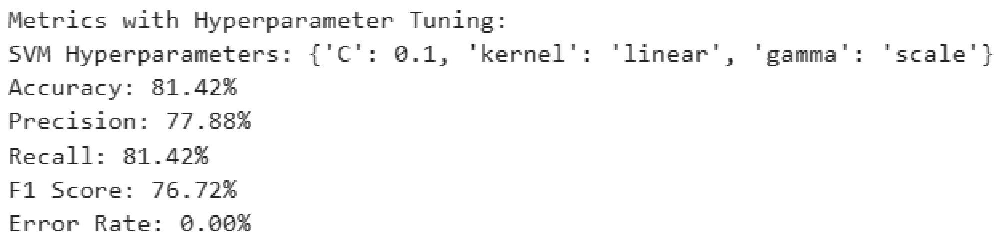
```
# Define SVM hyperparameters
svm_parameters = {'C': 11, 'kernel': 'rbf', 'gamma': 1}

# Create the SVM classifier
svm_model = SVC(**svm_parameters)
svm_model.fit(x_train, y_train)

# Make predictions
y_pred = svm_model.predict(x_test)

# Calculate metrics
accuracy = accuracy_score(y_test, y_pred) * 100
precision = precision_score(y_test, y_pred, average='weighted', zero_division=1) * 100
recall = recall_score(y_test, y_pred, average='weighted') * 100
f1 = f1_score(y_test, y_pred, average='weighted') * 100
conf_matrix = confusion_matrix(y_test, y_pred)
error_rate = ((conf_matrix[0, 1] + conf_matrix[1, 0]) / float(conf_matrix.sum())) * 100

# Print metrics
print("\nMetrics with Hyperparameter Tuning:")
print("SVM Hyperparameters:", svm_parameters)
print("Accuracy: {:.2f}%".format(accuracy))
print("Precision: {:.2f}%".format(precision))
print("Recall: {:.2f}%".format(recall))
print("F1 Score: {:.2f}%".format(f1))
print("Error Rate: {:.2f}%".format(error_rate))
```

Figure 28: SVM Hyperparameters



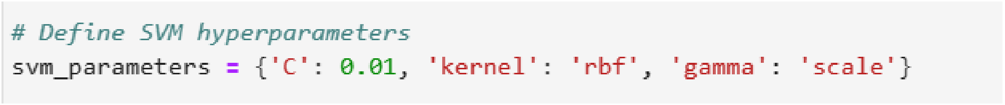
```
# Define SVM hyperparameters
svm_parameters = {'C': 0.1, 'kernel': 'linear', 'gamma': 'scale'}
```

Figure 29: Model training



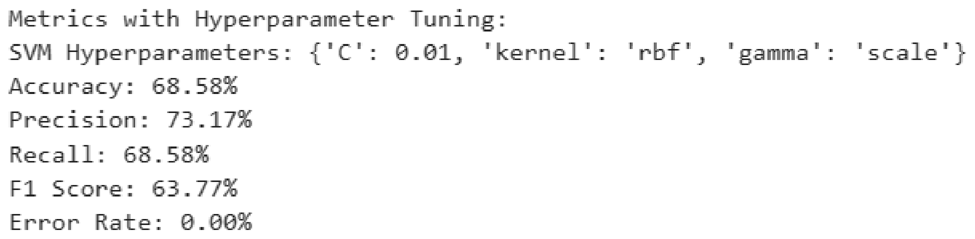
```
Metrics with Hyperparameter Tuning:
SVM Hyperparameters: {'C': 0.1, 'kernel': 'linear', 'gamma': 'scale'}
Accuracy: 81.42%
Precision: 77.88%
Recall: 81.42%
F1 Score: 76.72%
Error Rate: 0.00%
```

Figure 30: Model training results



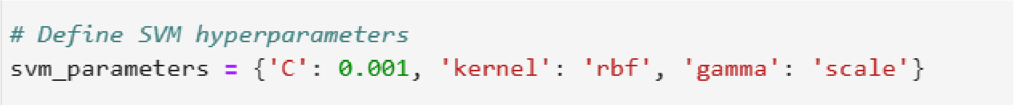
```
# Define SVM hyperparameters
svm_parameters = {'C': 0.01, 'kernel': 'rbf', 'gamma': 'scale'}
```

Figure 31:Model training



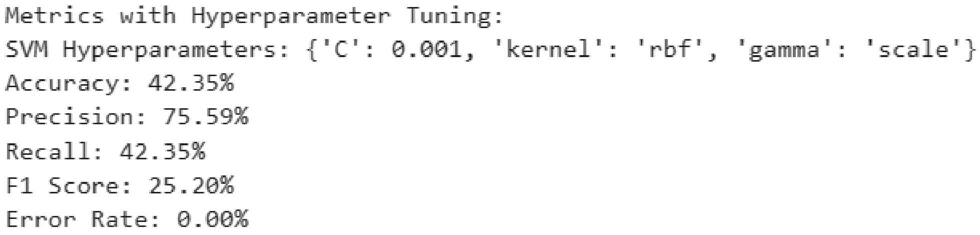
```
Metrics with Hyperparameter Tuning:
SVM Hyperparameters: {'C': 0.01, 'kernel': 'rbf', 'gamma': 'scale'}
Accuracy: 68.58%
Precision: 73.17%
Recall: 68.58%
F1 Score: 63.77%
Error Rate: 0.00%
```

Figure 32:Model training result



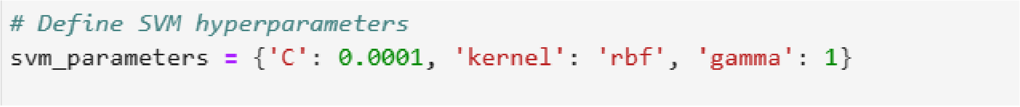
```
# Define SVM hyperparameters
svm_parameters = {'C': 0.001, 'kernel': 'rbf', 'gamma': 'scale'}
```

Figure 33:Model training



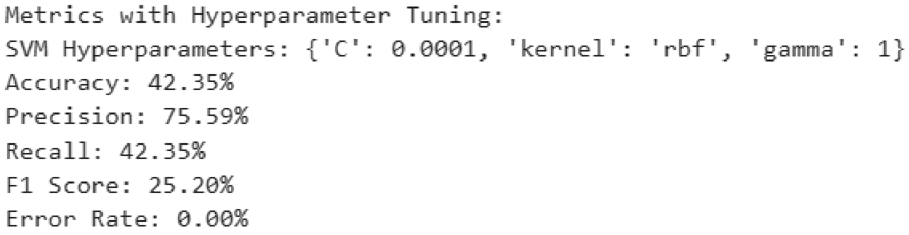
```
Metrics with Hyperparameter Tuning:
SVM Hyperparameters: {'C': 0.001, 'kernel': 'rbf', 'gamma': 'scale'}
Accuracy: 42.35%
Precision: 75.59%
Recall: 42.35%
F1 Score: 25.20%
Error Rate: 0.00%
```

Figure 34:Model training results



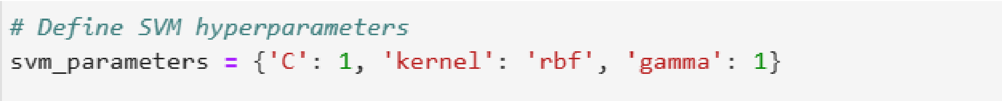
```
# Define SVM hyperparameters
svm_parameters = {'C': 0.0001, 'kernel': 'rbf', 'gamma': 1}
```

Figure 35:Model training



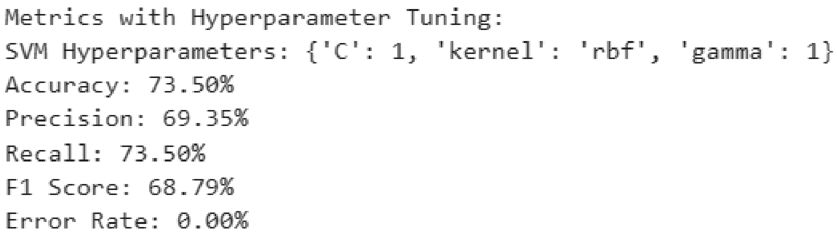
```
Metrics with Hyperparameter Tuning:
SVM Hyperparameters: {'C': 0.0001, 'kernel': 'rbf', 'gamma': 1}
Accuracy: 42.35%
Precision: 75.59%
Recall: 42.35%
F1 Score: 25.20%
Error Rate: 0.00%
```

Figure 36:Model training results



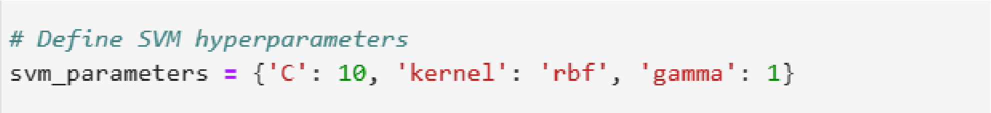
```
# Define SVM hyperparameters
svm_parameters = {'C': 1, 'kernel': 'rbf', 'gamma': 1}
```

Figure 37:Model training



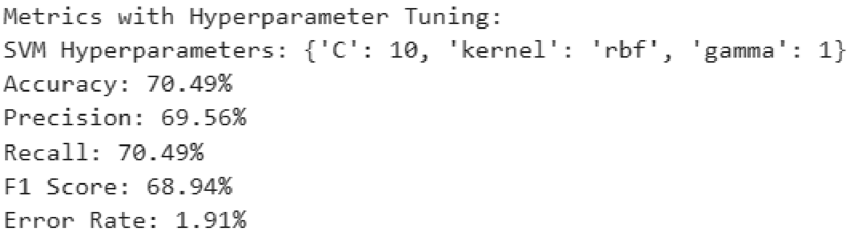
```
Metrics with Hyperparameter Tuning:
SVM Hyperparameters: {'C': 1, 'kernel': 'rbf', 'gamma': 1}
Accuracy: 73.50%
Precision: 69.35%
Recall: 73.50%
F1 Score: 68.79%
Error Rate: 0.00%
```

Figure 38:Model training results



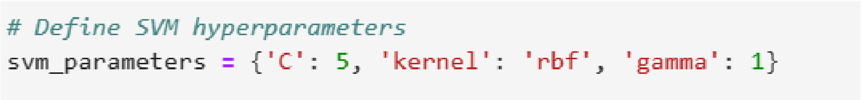
```
# Define SVM hyperparameters
svm_parameters = {'C': 10, 'kernel': 'rbf', 'gamma': 1}
```

Figure 39:Model training



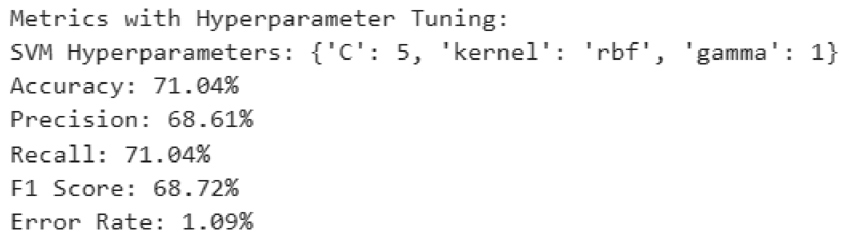
```
Metrics with Hyperparameter Tuning:
SVM Hyperparameters: {'C': 10, 'kernel': 'rbf', 'gamma': 1}
Accuracy: 70.49%
Precision: 69.56%
Recall: 70.49%
F1 Score: 68.94%
Error Rate: 1.91%
```

Figure 40:Model training results



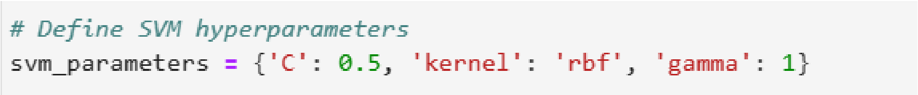
```
# Define SVM hyperparameters
svm_parameters = {'C': 5, 'kernel': 'rbf', 'gamma': 1}
```

Figure 41:Model training



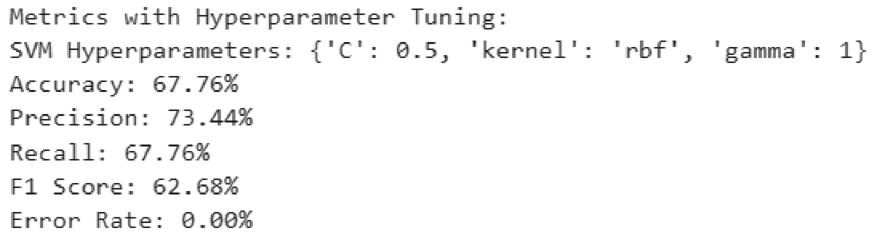
```
Metrics with Hyperparameter Tuning:
SVM Hyperparameters: {'C': 5, 'kernel': 'rbf', 'gamma': 1}
Accuracy: 71.04%
Precision: 68.61%
Recall: 71.04%
F1 Score: 68.72%
Error Rate: 1.09%
```

Figure 42:Model training results



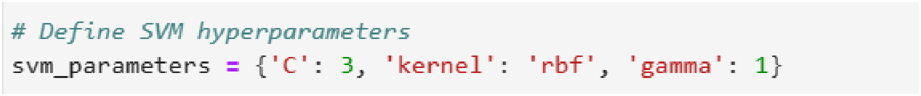
```
# Define SVM hyperparameters
svm_parameters = {'C': 0.5, 'kernel': 'rbf', 'gamma': 1}
```

Figure 43:Model training



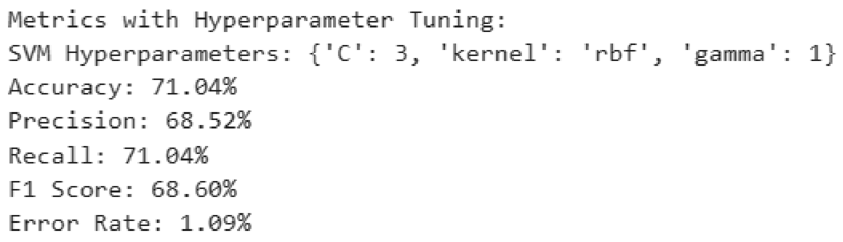
```
Metrics with Hyperparameter Tuning:
SVM Hyperparameters: {'C': 0.5, 'kernel': 'rbf', 'gamma': 1}
Accuracy: 67.76%
Precision: 73.44%
Recall: 67.76%
F1 Score: 62.68%
Error Rate: 0.00%
```

Figure 44: Model training results



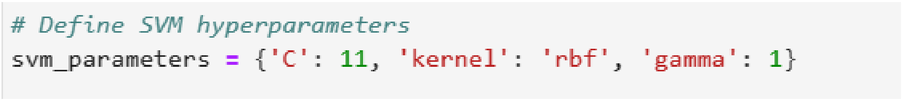
```
# Define SVM hyperparameters
svm_parameters = {'C': 3, 'kernel': 'rbf', 'gamma': 1}
```

Figure 45: Model training



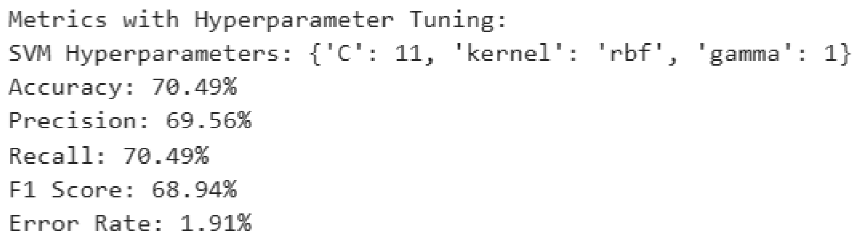
```
Metrics with Hyperparameter Tuning:
SVM Hyperparameters: {'C': 3, 'kernel': 'rbf', 'gamma': 1}
Accuracy: 71.04%
Precision: 68.52%
Recall: 71.04%
F1 Score: 68.60%
Error Rate: 1.09%
```

Figure 46: Model training results



```
# Define SVM hyperparameters
svm_parameters = {'C': 11, 'kernel': 'rbf', 'gamma': 1}
```

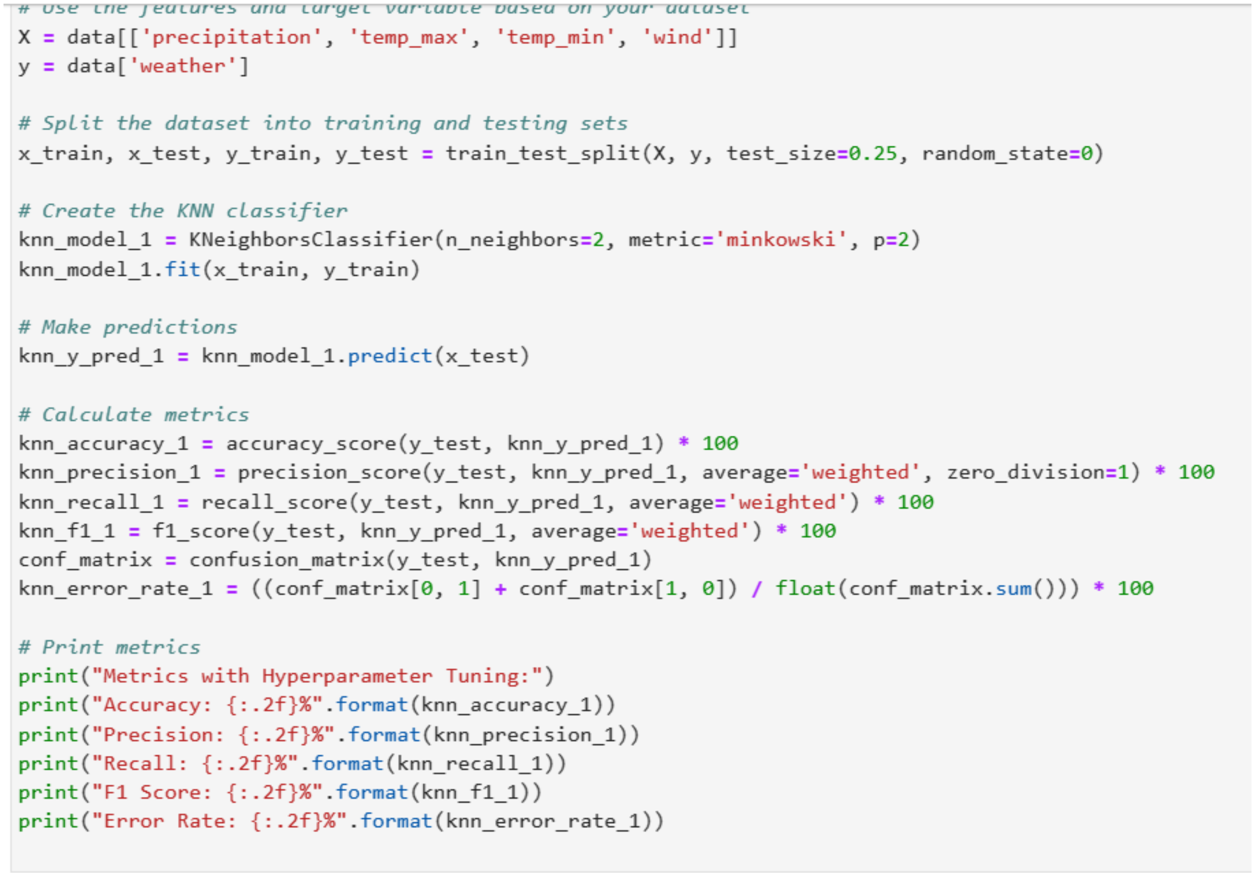
Figure 47: Model training



```
Metrics with Hyperparameter Tuning:
SVM Hyperparameters: {'C': 11, 'kernel': 'rbf', 'gamma': 1}
Accuracy: 70.49%
Precision: 69.56%
Recall: 70.49%
F1 Score: 68.94%
Error Rate: 1.91%
```

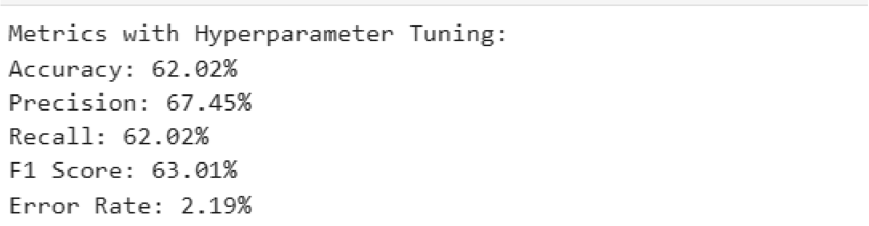
Figure 48: Model training results

3.4.3.15 KNN hyperparameter tuning



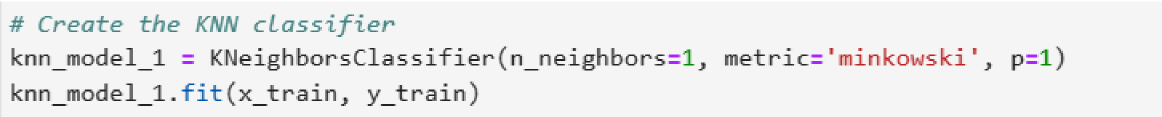
```
# Use the features and target variable based on your dataset
X = data[['precipitation', 'temp_max', 'temp_min', 'wind']]
y = data['weather']

# Split the dataset into training and testing sets
x_train, x_test, y_train, y_test = train_test_split(X, y, test_size=0.25, random_state=0)

# Create the KNN classifier
knn_model_1 = KNeighborsClassifier(n_neighbors=2, metric='minkowski', p=2)
knn_model_1.fit(x_train, y_train)

# Make predictions
knn_y_pred_1 = knn_model_1.predict(x_test)

# Calculate metrics
knn_accuracy_1 = accuracy_score(y_test, knn_y_pred_1) * 100
knn_precision_1 = precision_score(y_test, knn_y_pred_1, average='weighted', zero_division=1) * 100
knn_recall_1 = recall_score(y_test, knn_y_pred_1, average='weighted') * 100
knn_f1_1 = f1_score(y_test, knn_y_pred_1, average='weighted') * 100
conf_matrix = confusion_matrix(y_test, knn_y_pred_1)
knn_error_rate_1 = ((conf_matrix[0, 1] + conf_matrix[1, 0]) / float(conf_matrix.sum())) * 100

# Print metrics
print("Metrics with Hyperparameter Tuning:")
print("Accuracy: {:.2f}%".format(knn_accuracy_1))
print("Precision: {:.2f}%".format(knn_precision_1))
print("Recall: {:.2f}%".format(knn_recall_1))
print("F1 Score: {:.2f}%".format(knn_f1_1))
print("Error Rate: {:.2f}%".format(knn_error_rate_1))
```

Figure 49: K-NN metrics



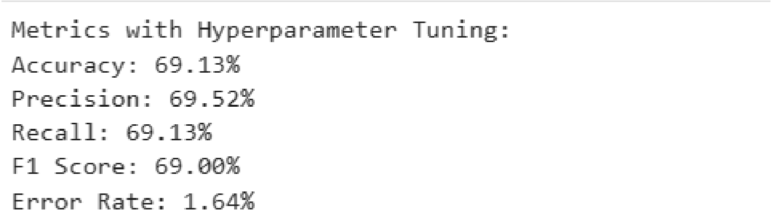
```
Metrics with Hyperparameter Tuning:
Accuracy: 62.02%
Precision: 67.45%
Recall: 62.02%
F1 Score: 63.01%
Error Rate: 2.19%
```

Figure 50: Model training results



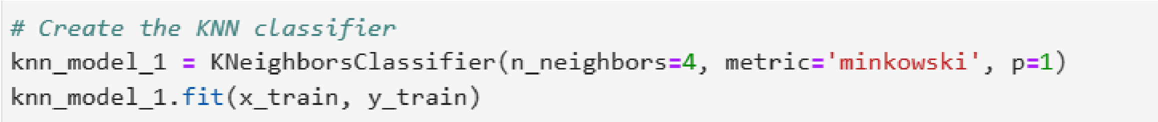
```
# Create the KNN classifier
knn_model_1 = KNeighborsClassifier(n_neighbors=1, metric='minkowski', p=1)
knn_model_1.fit(x_train, y_train)
```

Figure 51: Model training



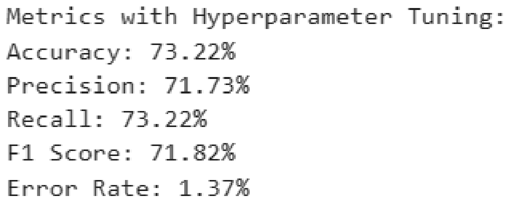
```
Metrics with Hyperparameter Tuning:
Accuracy: 69.13%
Precision: 69.52%
Recall: 69.13%
F1 Score: 69.00%
Error Rate: 1.64%
```

Figure 52:Model training results



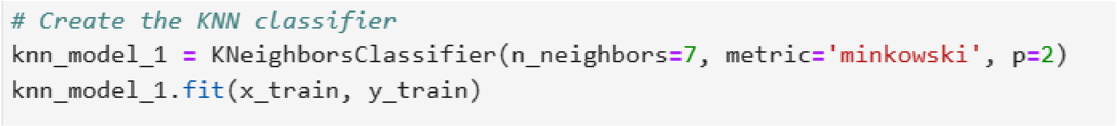
```
# Create the KNN classifier
knn_model_1 = KNeighborsClassifier(n_neighbors=4, metric='minkowski', p=1)
knn_model_1.fit(x_train, y_train)
```

Figure 53: Model training



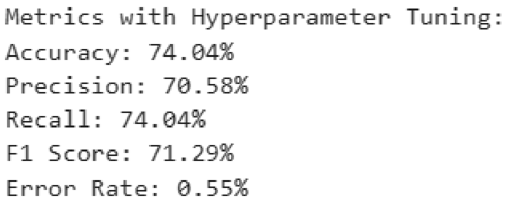
```
Metrics with Hyperparameter Tuning:
Accuracy: 73.22%
Precision: 71.73%
Recall: 73.22%
F1 Score: 71.82%
Error Rate: 1.37%
```

Figure 54: Model training results



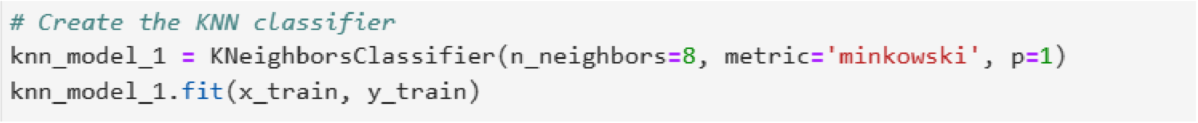
```
# Create the KNN classifier
knn_model_1 = KNeighborsClassifier(n_neighbors=7, metric='minkowski', p=2)
knn_model_1.fit(x_train, y_train)
```

Figure 55: Model training



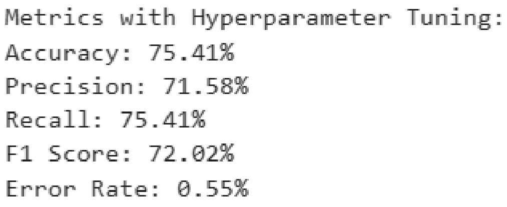
```
Metrics with Hyperparameter Tuning:
Accuracy: 74.04%
Precision: 70.58%
Recall: 74.04%
F1 Score: 71.29%
Error Rate: 0.55%
```

Figure 56: Model training results



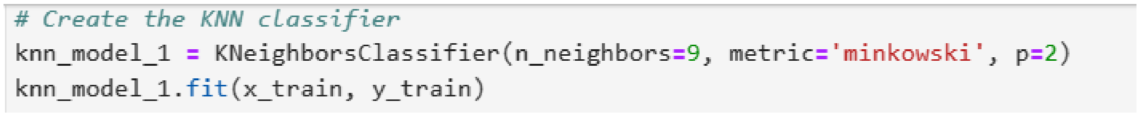
```
# Create the KNN classifier
knn_model_1 = KNeighborsClassifier(n_neighbors=8, metric='minkowski', p=1)
knn_model_1.fit(x_train, y_train)
```

Figure 57: Model training



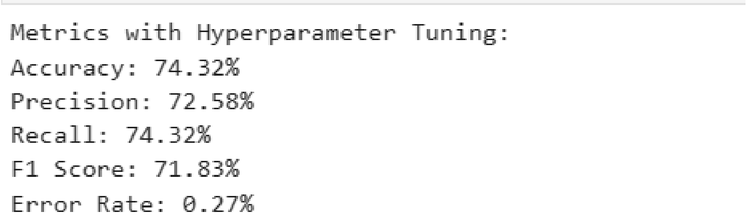
```
Metrics with Hyperparameter Tuning:
Accuracy: 75.41%
Precision: 71.58%
Recall: 75.41%
F1 Score: 72.02%
Error Rate: 0.55%
```

Figure 58: Model training results



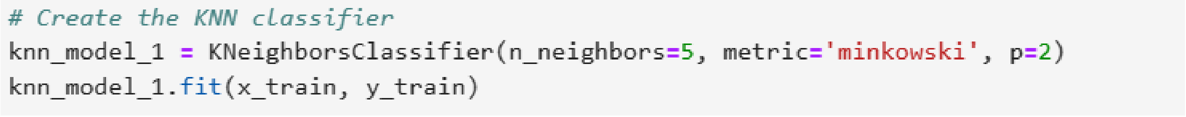
```
# Create the KNN classifier
knn_model_1 = KNeighborsClassifier(n_neighbors=9, metric='minkowski', p=2)
knn_model_1.fit(x_train, y_train)
```

Figure 59: Model training



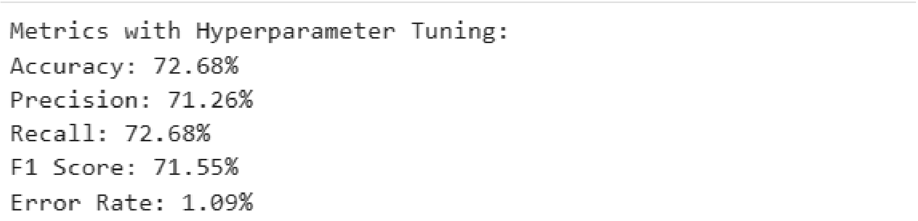
```
Metrics with Hyperparameter Tuning:
Accuracy: 74.32%
Precision: 72.58%
Recall: 74.32%
F1 Score: 71.83%
Error Rate: 0.27%
```

Figure 60: Model training results



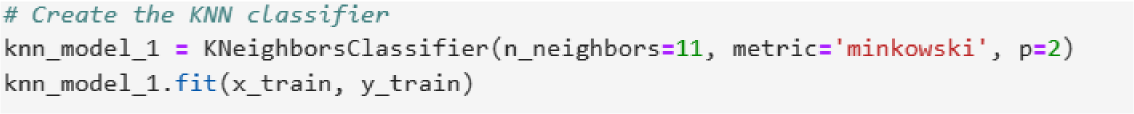
```
# Create the KNN classifier
knn_model_1 = KNeighborsClassifier(n_neighbors=5, metric='minkowski', p=2)
knn_model_1.fit(x_train, y_train)
```

Figure 61:Model training



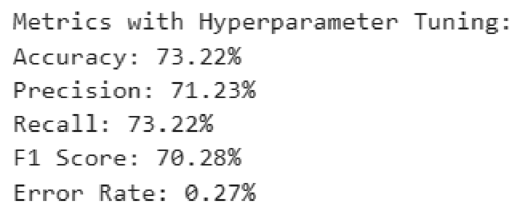
```
Metrics with Hyperparameter Tuning:
Accuracy: 72.68%
Precision: 71.26%
Recall: 72.68%
F1 Score: 71.55%
Error Rate: 1.09%
```

Figure 62: Model training results



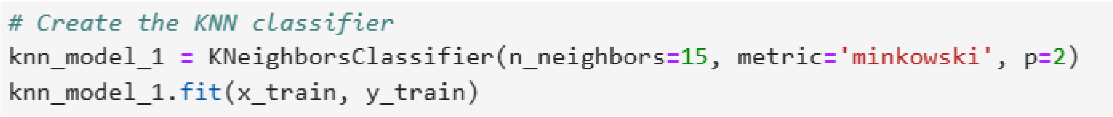
```
# Create the KNN classifier
knn_model_1 = KNeighborsClassifier(n_neighbors=11, metric='minkowski', p=2)
knn_model_1.fit(x_train, y_train)
```

Figure 63: Model training



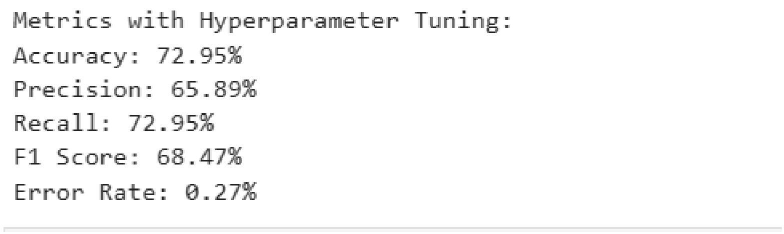
```
Metrics with Hyperparameter Tuning:
Accuracy: 73.22%
Precision: 71.23%
Recall: 73.22%
F1 Score: 70.28%
Error Rate: 0.27%
```

Figure 64: Model training results



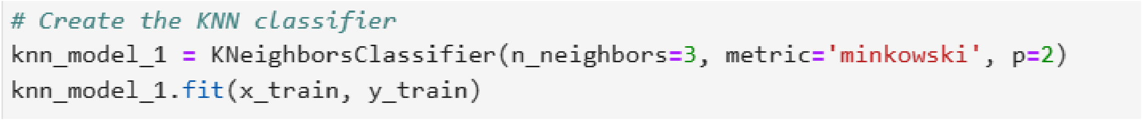
```
# Create the KNN classifier
knn_model_1 = KNeighborsClassifier(n_neighbors=15, metric='minkowski', p=2)
knn_model_1.fit(x_train, y_train)
```

Figure 65: Model training



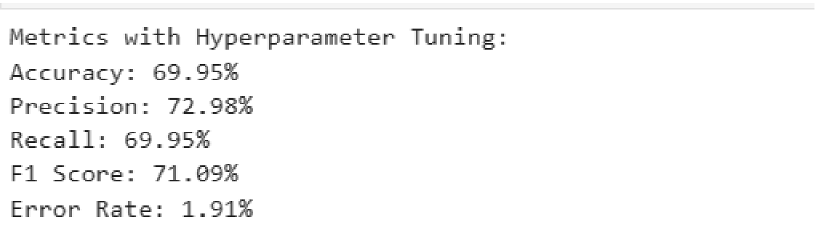
```
Metrics with Hyperparameter Tuning:
Accuracy: 72.95%
Precision: 65.89%
Recall: 72.95%
F1 Score: 68.47%
Error Rate: 0.27%
```

Figure 66: Model training results



```
# Create the KNN classifier
knn_model_1 = KNeighborsClassifier(n_neighbors=3, metric='minkowski', p=2)
knn_model_1.fit(x_train, y_train)
```

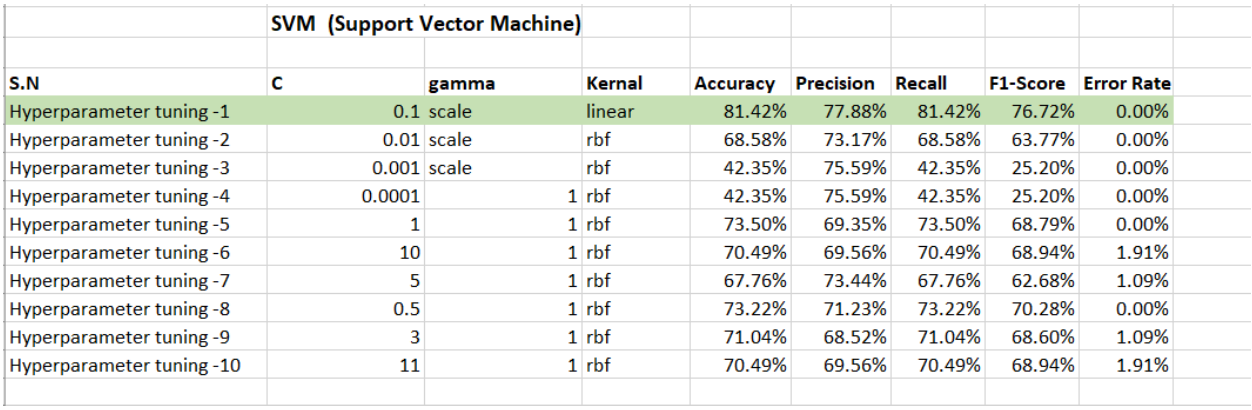
Figure 67: Model training



```
Metrics with Hyperparameter Tuning:
Accuracy: 69.95%
Precision: 72.98%
Recall: 69.95%
F1 Score: 71.09%
Error Rate: 1.91%
```

Figure 68: Model training results

3.4.4 Achieved Results

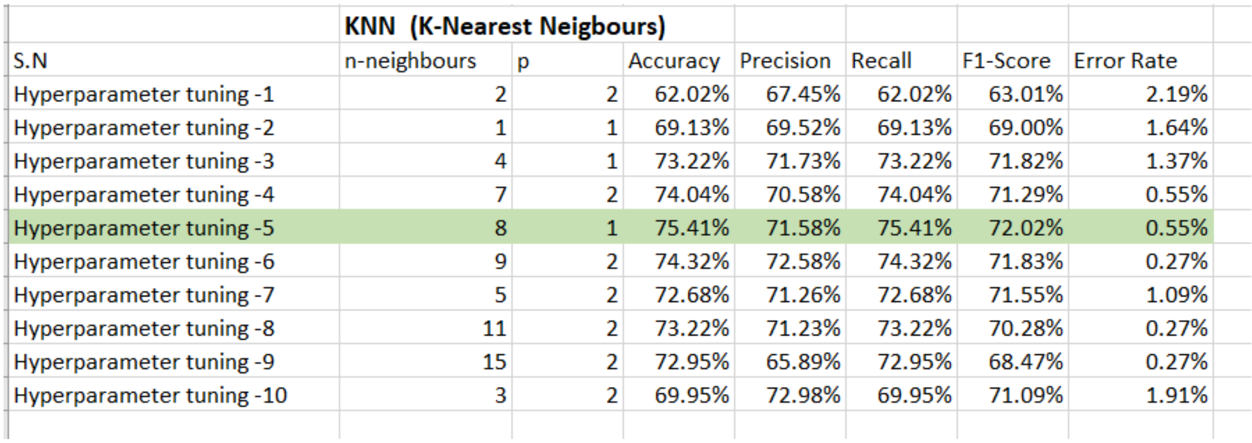


SVM (Support Vector Machine)

S.N	C	gamma	Kernal	Accuracy	Precision	Recall	F1-Score	Error Rate
Hyperparameter tuning -1	0.1	scale	linear	81.42%	77.88%	81.42%	76.72%	0.00%
Hyperparameter tuning -2	0.01	scale	rbf	68.58%	73.17%	68.58%	63.77%	0.00%
Hyperparameter tuning -3	0.001	scale	rbf	42.35%	75.59%	42.35%	25.20%	0.00%
Hyperparameter tuning -4	0.0001	1	rbf	42.35%	75.59%	42.35%	25.20%	0.00%
Hyperparameter tuning -5	1	1	rbf	73.50%	69.35%	73.50%	68.79%	0.00%
Hyperparameter tuning -6	10	1	rbf	70.49%	69.56%	70.49%	68.94%	1.91%
Hyperparameter tuning -7	5	1	rbf	67.76%	73.44%	67.76%	62.68%	1.09%
Hyperparameter tuning -8	0.5	1	rbf	73.22%	71.23%	73.22%	70.28%	0.00%
Hyperparameter tuning -9	3	1	rbf	71.04%	68.52%	71.04%	68.60%	1.09%
Hyperparameter tuning -10	11	1	rbf	70.49%	69.56%	70.49%	68.94%	1.91%

Figure 69: Best result obtained (SVM)

The table illustrates the impact of hyperparameter choices on the SVM's model performance. The row with the highest accuracy among the other data configuration has been highlighted (Hyperparameter tuning -1). The performance metrics are accuracy, precision, recall, F1-score and error rate. 81.42% is the highest accuracy observed with 'gamma': scale, 'C': 0.1 value and kernel: 'linear'.



KNN (K-Nearest Neigbours)

S.N	n-neighbours	p	Accuracy	Precision	Recall	F1-Score	Error Rate
Hyperparameter tuning -1	2	2	62.02%	67.45%	62.02%	63.01%	2.19%
Hyperparameter tuning -2	1	1	69.13%	69.52%	69.13%	69.00%	1.64%
Hyperparameter tuning -3	4	1	73.22%	71.73%	73.22%	71.82%	1.37%
Hyperparameter tuning -4	7	2	74.04%	70.58%	74.04%	71.29%	0.55%
Hyperparameter tuning -5	8	1	75.41%	71.58%	75.41%	72.02%	0.55%
Hyperparameter tuning -6	9	2	74.32%	72.58%	74.32%	71.83%	0.27%
Hyperparameter tuning -7	5	2	72.68%	71.26%	72.68%	71.55%	1.09%
Hyperparameter tuning -8	11	2	73.22%	71.23%	73.22%	70.28%	0.27%
Hyperparameter tuning -9	15	2	72.95%	65.89%	72.95%	68.47%	0.27%
Hyperparameter tuning -10	3	2	69.95%	72.98%	69.95%	71.09%	1.91%

Figure 70: Best result obtained (K-NN)

The table illustrates the impact of hyperparameter choices on the K-NN's model performance. The row with the highest accuracy among the other data configuration has been highlighted (Hyperparameter tuning -5). The performance metrics are accuracy, precision, recall, F1-score and error rate. 75.41% is the highest accuracy observed with n-neighbours: 2, and p:1.

4 Conclusion

4.1 Analysis of the work done

Ultimately, the coursework revolved around the fundamental principles of Artificial Intelligence (AI) and Machine Learning (ML). Examined the diverse range of research conducted on the subject utilizing classification or regression algorithms. The dataset utilized for this project was obtained from Kaggle. The methodology employed for this study includes the application of Naïve Bayes, KNN, and SVM.

Development started from importing the libraries and dataset. Then we pre-processed the data, converted the data into the data shape. Then we selected the percentage of the train and test set using the mentioned algorithm. While hyperparameter tuning using the SVM model highest accuracy was observed which is 81.42%.

The objective of this study was to enhance the precision of weather condition predictions. The precision is dependent upon the selection of the appropriate parameter, dataset, and algorithm.

The integration of machine learning in weather prediction not only enhances the accuracy of forecasts but also helps various businesses and individuals in risk mitigation and resource optimization. Several algorithms can be employed for weather prediction since each has its strengths and weaknesses. The way to select the suitable algorithm depends on factors such as the size of the dataset, the complexity of relationships in the data, and the nature of the prediction task. In this paper, the Naïve Bayes Classifier, Support Vector Machines (SVM), and K-Nearest Neighbors (KNN) are selected.

KNN predicts the weather conditions by considering the historical patterns of the neighboring locations with similar features; SVM predicts weather by finding a hyperplane, which helps in separating the instances of different weather conditions and patterns; and Naïve Bayes uses a probability model based on the past or historical data in weather prediction.

4.2 Solution that addresses the real-world problems

Weather being a crucial factor for various sectors since it impacts on daily activities, industries and safety. Weather somehow impacts human life, if we plan on going out for vacations, or attend event it is better if we know about weather conditions as it helps us to make informed decisions. Here are some prominent applications of weather prediction in our day-to-day life:

Agriculture:

Farmers depend on weather forecasts to plant, harvest crops, regulate irrigation, and pest control. The accurate prediction helps farmer to manage their harvesting schedules to optimize crop yields and irrigation.

Aviation:

Airlines have to plan their flight ahead they have to factors such as turbulence, wind patterns and thunderstorms into account to ensure passenger safety and operational efficiency.

Transportation and Logistics:

There are companies such as shipping companies, trucking firms, and logistics provider who rely on weather forecasts to schedule their routes, in order to avoid disruption and ensure timely deliveries.

Emergency Management:

Natural disaster such as hurricanes, tornadoes, floods, and wildfires can occur at any time, early warnings can save lives and help authorities to plan, coordinate evacuation efforts. Weather forecast is crucial for responding to natural disasters.

The above mentioned were few applications of weather forecasts to solve real-world problems, the accurate prediction of weather condition has significant impact on social, economic, and environmental aspects of human being, industries, and communities.

4.3 Further work

There are several areas for further work related to weather prediction using machine learning. Here are some potential areas where further works are to be done include:

1. Implementing more sophisticated machine learning algorithms that can provide precise weather forecasts regarding the possibilities of various weather conditions or situation such as rain, snow, drizzle or hail and so on.
2. By creating more sophisticated algorithm and integrating to agricultural sector so that the farmers can protect their crops from adverse weather conditions.
3. Predicting precise weather patterns can help in managing health risks, since there are case of infectious diseases and air quality that can cause heat-related illness and vector-borne diseases.
4. Insurance companies can benefit from it, since it can help in assessing and pricing weather-related risks by creating loss prevention strategies for businesses and individuals.
5. Water resource management can benefit by knowing the forecast as it aids in water reservoir management by predicting drought condition to facilitate water management.

Weather prediction will continue to evolve, offering solutions or helping a certain sector. The new technology will facilitate to continue research and innovation in meteorology, data analytics, and modeling techniques that will eventually contribute to unlock the full potential.

5 References

Adobe Stock, 2023. *Adobe Stock.* [Online]
Available at: https://stock.adobe.com/search?k=weather+forecast
[Accessed 20 December 2023].

Aiswarya Shaji, A. A. V. R., 2021. Weather Prediction Using Machine Learning Algorithms. *IEEE Xplore.*

Anaconda Navigator, 2018. *Anaconda.* [Online]
Available at: https://docs.anaconda.com/free/navigator/index.html
[Accessed 15 January 2024].

Anon., n.d. [Online].

BBC Weather, 2023. *BBC Weather.* [Online]
Available at: https://www.bbc.com/weather/1283240
[Accessed 20 December 2023].

Borden, K. & Cutter, S., 2023. *New Scientist.* [Online]
Available at: https://www.newscientist.com/article/dn16287-death-map-usa-natural-disaster-hotspots-revealed/
[Accessed 18 December 2023].

Brooks, R., 2023. *University of York.* [Online]
Available at: https://online.york.ac.uk/what-is-reinforcement-learning/#:~:text=Reinforcement%20learning%20(RL)%20is%20a,using%20feedback%20from%20its%20actions.
[Accessed 14 January 2024].

Copeland, B., 2023. *Britannica.* [Online]
Available at: https://www.britannica.com/technology/artificial-intelligence
[Accessed 17 December 2023].

Crabtree, M., 2023. *Datacamp.* [Online]
Available at: https://www.datacamp.com/blog/what-is-machine-learning
[Accessed 18 December 2023].

D.Dhilikumar, P. Y. B., T. Y. & K., 2023. Real Time Weather Prediction System using Ensemble Machine Learning. *IEEE Xplore,* Issue 23749233.

D, D., Y. b. & Y., 2023. *Real Time Weather Prediction System using.* s.l., ieee Xplore, pp. 1-8.

Favela, R., 2023. *tomorrow.io.* [Online]
Available at: https://www.tomorrow.io/blog/using-weather-ai-to-improve-operations-in-aviation/
[Accessed 20 December 2023].

Fenyi, A. & Asante, M., 2023. MULTI-CLASS WEATHER CLASSIFICATION USING MACHINE LEARNING TECHNIQUES. *Journal of Theoretical and Applied Information Technology,* 101(1992-8645).

GeeksforGeeks, 2023. *Geeks for Geeks.* [Online]
Available at: https://www.geeksforgeeks.org/support-vector-machine-algorithm/
[Accessed 16 December 2023].

geeksforgeeks, 2023. *geeksforgeeks.* [Online]
Available at: https://www.geeksforgeeks.org/what-is-reinforcement-learning/
[Accessed 19 December 2023].

Gillis, A. S., 2024. *TechTarget.* [Online]
Available at: https://www.techtarget.com/searchenterprisedesktop/definition/Excel
[Accessed 15 January 2024].

IBM, 2023. *IBM.* [Online]
Available at: https://www.ibm.com/weather
[Accessed 20 December 2023].

IBM, 2024. *IBM.* [Online]
Available at: https://www.ibm.com/topics/deep-learning
[Accessed 14 January 2024].

Java T point, 2023. *Java T point.* [Online]
Available at: https://www.javatpoint.com/k-nearest-neighbor-algorithm-for-machine-learning
[Accessed 15 December 2023].

Java T point, 2023. *javatpoint.com.* [Online]
Available at: https://www.javatpoint.com/machine-learning-naive-bayes-classifier
[Accessed 15 12 2023].

JavaTpoint, 2023. *javaTpoint.* [Online]
Available at: https://www.javatpoint.com/machine-learning-support-vector-machine-algorithm
[Accessed 20 December 2023].

Jupyter Notebook, 2015. *The Jupyter Notebook.* [Online]
Available at: https://jupyter-notebook.readthedocs.io/en/stable/notebook.html
[Accessed 15 January 2024].

Kanade, V., 2023. *spiceworks.* [Online]
Available at: https://www.spiceworks.com/tech/artificial-intelligence/articles/what-is-ml/
[Accessed 19 December 2023].

Mantri, R. et al., 2021. Weather Prediction and Classification Using Neural Networks and k-Nearest Neighbors. *IEEE Xplore,* Issue 21137870.

Mishra, D. & Joshi, P., 2021. A Comprehensive Study on Weather Forecasting using Machine Learning. *IEEE Xplore,* Issue 21381365.

Pandey, A. K., 2022. *Medium.* [Online]
Available at: https://medium.com/@arunp77/regression-algorithms-29f112797724#:~:text=Regression%20algorithms%20are%20a%20type%20of%20machine%20learning%20algorithm%20used,mathematical%20model%20to%20the%20data.
[Accessed 18 December 2023].

Rautela, P. & Karki, B., 2015. Weather Forecasting: Traditional Knowledge of the People of Uttarakhand Himalaya. *Research Gate,* III(19016), pp. 1-14.

Rouse, M., 2023. *Techopedia.* [Online]
Available at: https://www.techopedia.com/definition/31622/strong-artificial-intelligence-strong-ai
[Accessed 14 January 2024].

Rouse, M., 2023. *Techopedia.* [Online]
Available at: https://www.techopedia.com/definition/31621/weak-artificial-intelligence-weak-ai
[Accessed 14 January 2024].

Safia, M., Abbas, R. & Aslani, M., 2023. Classification of Weather Conditions Based on Supervised Learning for Swedish Cities. *MDPI Open access journals.*

Schroer, A., 2023. *builtin.* [Online]
Available at: https://builtin.com/artificial-intelligence
[Accessed 18 December 2023].

Wang, S., Chu, J., Li, J. & Duan, R., 2023. *MDPI Open Access Journals.* [Online]
Available at: https://www.mdpi.com/2226-4310/9/4/189
[Accessed 13 DEcember 2023].