Master Thesis

IU International University of Applied Science

Study Program: Masters in Data Science

Enhancing Weather Forecasting Through Data Science: A Robust Approach to Integrating Multiple Predictive Techniques

Name: Shadab Malik

Matriculation Number : 32003135

Place: Berlin, Germany

Date: 17.09.2024

Supervisor: Dr. Sameer Joshi

Date of Submission: 17.09.2024

Acknowledgment

I want to start by saying that I am greatly thankful to my supervisor Dr. Sameer Joshi for his unforgettable support, wisdom and encouragement during the time of this thesis. His vast knowledge, perceptive comments, and unwavering patience have been essential in the writing of this thesis. Dr. Joshi's steadfastness to perfection and his ability to motivate have been a consistent source of energy to me, and I am greatly pleased for I was part of his staff.

I would also like to thank my family whose constant love and support has been my strength in this journey. Ammi and Papa, thank you for your trust in me, for your innumerable boosts, and for always being ready to uplift me when I was in dire straits. Your sacrifices, patience, and belief in my capabilities have kept me on the go. I would like to thank my friends to whom I am grateful. They were always there for company, understanding and entertainment whenever I needed a break from the tough academic life. Your care, humour and confidence in me have made the trip more bearable and enjoyable.

Lastly, I would like to thank others who have contributed to the finish of my thesis. Your warmth and assistance have been deeply appreciated and I will be always grateful for your companionship.

Abstract

The advancement of machine learning techniques have significantly improved weather forecasting accuracy. Traditionally, weather forecast has long been dominated by Numerical Weather Prediction (NWP) models based on physical laws. NWP models can be viewed as a discretization of physics equations governing fluid flow and convection on a large grid of points, which makes these equations computationally expensive. This is particularly and local phenomena.

The thesis, with the overall objective of improving the accuracy of weather forecasting by developing better representations of the sectors that have been poorly addressed and too complex for the actual numerical prediction, introduces a hybrid model that combines Numerical Weather Prediction and Machine Learning techniques, with a special focus on Random Forest, XGBoost and Support Vector Classifiers.

The study deploys a collection of weather variables (a comprehensive dataset derived from meteorological data from past). Models have been developed, hyperparameters fine-tuned, class polarization tackled by SMOTE (Synthetic Minority Over-sampling Technique). Of all the models tested, XGBoost acquires the highest robustness, providing a highest accuracy of 80% across different weather classifications. Comparative assesment plots utilising weighted assessment parameters such as accuracy, precision, recall, F1-score, ROC curves reveal that our introduced model has better performance than traditional models such as Decision Tree and Logistic Regression.

Our results can have practical applications in forecasting weather conditions for diverse areas of human activity like agriculture, disaster management, aviation and renewable energy weather forecasts can lead to better decision-making about risk mitigation related to extreme weather events. Specifically, the technique offers new ways of looking at our current weather models. ‘The main new aspect of our hybrid model is added flexibility and, therefore, interpretability,’ explains Brenowitz. While the study demonstrates that it’s possible to have both, computational requirements for such a model, and the need for high-resolution data, pose challenges for next steps in this field of research. These include how we might develop the computational techniques required to make such models useful in real time.

Table of Contents

List of Abbreviations

- **ML** - Machine Learning
- **LSTM**: Long Short-Term Memory
- **NWP**: Numerical Weather Prediction
- **NLP** - Natural Language Processing
- **SVM** - Support Vector Machine
- **RBF** - Radial Basis Function
- **ROC** - Receiver Operating Characteristic
- **AUC** - Area Under the Curve
- **KNN** - K-Nearest Neighbors
- **TPR** - True Positive Rate
- **FPR** - False Positive Rate
- **RF** - Random Forest
- **XGBoost** - Extreme Gradient Boosting
- **SMOTE** - Synthetic Minority Over-sampling Technique
- **EDA** - Exploratory Data Analysis
- **PCA** - Principal Component Analysis

List of Figures

1. Introduction

1.1 Background of the Study

This chapter deals with an overview of the historical context, importance, and current challenges of weather forecasting. It also highlights the role of data science and machine learning in transforming weather forecasting.

Historical Evolution of Weather Forecasting

Weather forecasting goes back to hundreds of centuries. Weather forecasting involves prediction of the future atmospheric phenomenon based on the historical weather attributes. The earliest forms of weather prediction were based on observations of natural phenomena, such as the behaviour of animals, changes in wind direction or the appearance of clouds. Ancient civilizations like the Babylonians used to live on earth around 650 BC. They used cloud patterns and astrology to make approximate weather predictions. Similarly, in ancient China, farmers relied on empirical rules based on centuries of observation to predict seasonal weather patterns. These early methods were quite insightful but they lacked scientific proof and were often inaccurate.

The barometer was invented by Evangelista Torricelli in the 17th century. It was that time when The scientific approach to weather forecasting. Barometer was used to measure the atmospheric pressure which is a critical parameter in understanding weather changes. There were couple of more inventions of scientific instruments such as thermometer and hygrometer. These instruments enabled more systematic observations of atmospheric features. After the invention of electric telegraph in mid 19th century, weather forecasting took a whole new dimension. As now huge weather related data could be transmitted at a fast pace . It revolutionised the weather forecasting in a way. This led to the establishment of meteorological networks and the creation of the first weather maps, which formed the basis of synoptic meteorology.

Numerical weather prediction (NWP) was developed in the early 20th century. It marked a significant development in weather forecasting based on physics. The very first NWP model was proposed by Lewis Richardson in 1922. It used mathematical equations to simulate atmospheric processes (Richardson, 1922). The development of computers in the mid-20th century to late 20th century made it feasible to solve these complex equations quickly, leading to the rise of computer-based weather forecasting. Since then, NWP has become the pillars of modern weather prediction. There has been continuous improvements in computing power, data assimilation and model resolution (Kalnay, 2003).

The Critical Role of Accurate Weather Forecasting in Various Sectors

Weather forecasting in its accurate form plays a pivotal role in various sectors. We shall discuss some of them here.

- **Aviation:** Weather forecasting is used for route planning and route optimization in aviation industry so that the safety of passengers and crews are maintained. Airline companies rely

on forecasts to avoid severe weather condition such as thunderstorms, turbulence and icing. All these phenomena possess a threat to flight safety (Cox et al., 2020). If we can do the accurate weather predictions then it helps in minimizing flight delays and cancellations. Ultimately it will benefit the airline companies to save on fuel and operating costs.

- **Agriculture:** Weather forecasting is used in agriculture sector to plant, in irrigation, to add fertilizers to crops and when to harvest. If the weather forecasting is accurate then it helps farmers to optimize crop yield. It also helps farmers to reduce losses because of adverse water conditions like drought or floods. Farmers can efficiently manage water and fertilizers. For example, if there is an early warning of droughts or floods then farmers can take necessary steps to protect their farms (Ray et al., 2015).
- **Military:** Military operations have to be well planned for minimum loss and maximum results. Sometimes, it is planned well before the execution date and sometimes only before few hours. Accurate weather forecast plays an important role in planning military operations especially in harsh weather conditions such as mountains or snowy regions or dense forests. Weather intelligence can influence the timing of operations. It also influences the choice of routes and the deployment of personnel and equipment.
- **Energy and Utilities:** Renewable energy sector such as solar and wind depends on the weather forecasting. These sectors uses weather prediction to forecast the production of renewable energy. Accurate forecasts help in grid management and optimizing energy production and storage (Pérez et al., 2020).
- **Disaster Management:** Disaster can happen anytime. But in response to those disaster, a proper planning is done by disaster management team for effective saving lives of those who are affected. Disaster management team will be better prepared for events such as typhoon or tsunami or heatwaves etc, if they got accurate weather prediction. These early warnings can definitely save lots of lives and minimize economic damages (WMO, 2018). For example, if the disaster management agency is already informed about the path and intensity of upcoming tornado then they can implement emergency planning to affected areas

Current Challenges in Weather Forecasting

Though there has been technological advancement in weather forecasting but still there are few challenges in weather forecasting

- **Limitations of Traditional Forecasting Models:** In previous topic we discussed about the traditional model to predict weather which is NWP. This model has limitations. It cannot handle complex data. It also cannot handle non-linear atmospheric features particularly at small scales**.** NWP depends on the approximations and parameterizations. These can introduce errors. They require significant resources for computation. It limits NWP resolutions and frequency.

- **Complexity of Atmospheric Dynamics:** The atmosphere has lots of features and all of them are dynamic in nature. Sometimes they are too chaotic. They are governed by non-linear processes and interaction across multiple scales. Be it local thunderstorms or global climate patterns. To predict weather very accurately it requires to solve complex mathematical equations. These equations describe thermodynamics, fluid mechanics and radiation transfer. All these calculations are computational heavy and often uncertain (Lorenz, 1963).
- **Need for High-Resolution Predictions:** We need high resolution predictions because there has been demand for more localized and short term weather forecasts. that can capture small-scale weather phenomena, such as urban heat islands, localized flooding, or microbursts. We require advance models which can make high resolution predictions. It also requires extensive data storage and increased computational capacity. All these possess technical and financial challenges both (Sun et al., 2014).
- **Data Quality Issues:** Data quality is the biggest asset for accurate weather prediction. We need high quality observed data. There are several factors which affects the quality of data such as sensor malfunction, inconsistencies and gaps in data coverage. The data generated by satellites, radars and other sensors are very high in volume. There is a problem in storing all of these continuously generated data and feeding them into models (Bauer et al., 2015).

Introduction to Data Science and Machine Learning in Weather Forecasting

The recent development of data science and machine learning models have revolutionized the field of weather forecasting by offering new tools and methodologies to overcome some of these challenges. We will discuss few of them as follows:

- **Machine Learning in Weather Forecasting:** Machine learning is a subset of artificial intelligence. It has gained prominence in weather forecasting due to its ability to identify patterns in complex non-relational data without using any physics. There are various machine learning algorithms such as decision trees, random forest, neural networks and ensemble methods which can learn from historical weather data and improve their predictions over time (McGovern et al., 2017). These models are particularly effective in short-term forecasting. These models are used to predicting specific weather events such as rain, fog or storms (Schultz et al., 2021).
- **Data Science in Weather Forecasting:** Data science is nothing but analysing historical data using statistics, machine learning methods and extracting meaningful information or insights. Then using those insights to make prediction. Data science is used in weather forecasting. Large volume of data which have been generated by various sensors, radars or satellites are fed into data science models to identify the patterns among them. Finally using that pattern to predict the atmospheric phenomena such as rain, fog or storms (Chattopadhyay et al., 2020).

- **Transition from Traditional to AI-Based Approaches:** Traditionally methods such as regression analysis were widely used by the meteorology department. But they were limited because of their assumptions of linearity and normality. And if we talk about modern AI-based approaches then it can handle vast amounts of heterogeneous data. It can also capture complex dependencies and adapt to changing conditions (Rasp et al., 2018). For example, ensemble learning methods like XGBoost or deep learning techniques such as Convolutional Neural Networks (CNNs) and Long Short-Term Memory (LSTM) networks have been successfully applied to enhance forecast accuracy and provide high-resolution, localized predictions.
- **Integrating Multiple Predictive Techniques:** Now we can also integrate multiple predictive models such as NWP models with machine learning models. This is a very robust approach to improve the accuracy of the predictive model. Hybrid models basically leverage the strengths of different models and compensate for their individual weaknesses. Ultimately they are able to provide more accurate and reliable forecasting.

This thesis aims to build on these advancements by developing a robust framework that integrates multiple predictive techniques. When I used baseline model then the accuracy was far less as required. However, after combining other models and doing hyperparameter tuning a much better accuracy was achieved.

1.2 Problem Statement

The problem statement defines the specific gaps in current weather forecasting methodologies. It also highlights the limitations of existing models in handling complex and non-linear weather patterns. The challenges which are faced while integrating multiple predictive models and why is there need for a hybrid model to enhance forecast accuracy which is also robust in nature.

Identification of Specific Gaps in Current Weather Forecasting Methodologies

We have already discussed that accurate weather forecasting is very crucial to different sectors such as agriculture, aviation, disaster management etc. But there are significant gaps and limitations in current forecasting techniques that restrict the model to give best possible accuracy of the weather forecasting. We can categorise these issues into three categories. These are limitations of existing models, challenges in integrating multiple predictive techniques and the need for a robust hybrid model.

Limitations of Existing Models (Numerical and Statistical) in Handling Complex, Non-Linear Weather Patterns

Numerical Weather Prediction (NWP) Models: NWP models are based on solving mathematical equations of physical law that describe the atmosphere's fluid dynamics and thermodynamics. NWP is the pillar of weather forecasting for decades (Kalnay, 2003). However, these models have inherent limitations:

Handling of Non-Linear Dynamics: The atmospheric features are very complex, non-linear and sometimes chaotic in nature (Lorenz, 1963). NWP models are not able to give best accuracy particularly at smaller scales where turbulence and localized phenomena are critical (Sun et al., 2014). These local phenomena could be thunderstorm or cloudburst. These limitations can result in significant errors, especially in short-term forecasts and predictions of extreme weather events (Bauer et al., 2015).

Computational Constraints: NWP model requires lot of computational resources to run high-resolution simulations that capture small-scale weather patterns. These high resolutions simulation are captured by radars and sensors. It causes computational constraints. Because of these, models rely on approximations and simplifications. These approximations are in the form of parameterizations of sub-grid scale processes. It can introduce errors (Rasp et al., 2018).

Statistical Models: Traditional statistical models such as decision tree, regression analysis and time series models have been widely used in weather forecasting (Wilks, 2011). But these models have several drawbacks.

Assumptions of Linearity: Statistical models assume linear relationships between meteorological variables, which do not adequately reflect the non-linear nature of atmospheric processes (McGovern et al., 2017). Because of it, these models are not able to recognize the dependencies which are complex in nature. It also fails to understand the relation between variables that are crucial for accurate weather prediction.

Limited Generalization: This limitation is because statistical models are not generic in nature when it comes to weather forecasting. Statistical methods are usually designed for specific locations or regions and may not generalize well to different geographical areas. It also doesn't work well with every climatic conditions. They also require extensive historical data for calibration. These historical data may be not available or reliable all the time (Gul et al., 2020).

The Challenge of Integrating Multiple Predictive Techniques for Better Accuracy

The world has seen a lot of development in data science and machine learning which is a subset of AI. They have opened new possibilities for improving weather forecasting. But integrating multiple predictive techniques remains challenging:

Diverse Methodologies and Data Sources: If we want to combine different predictive models with NWP models then we have to combine the data sources and diverse methodologies as well. There is a challenge in combining these two because NWP models are physics-based and rely heavily on numerical data. Whereas ML models are data-driven and can incorporate both structured (numerical) and unstructured (text, images) data (Chattopadhyay et al., 2020). Because of this diversity there is a significant challenge for data harmonization. It also possess challenge to feature engineering and model compatibility.

Model Uncertainty and Complexity: Whatever predictive technique we use, whether regression analysis, decision tree or Random forest. They all have their own set of uncertainties and assumptions. If we talk about NWP models then it may have biases related to parameterization schemes. ML models may suffer from overfitting or underfitting due to limited training data (McGovern et al., 2017). There are certain techniques to deal with the overfitting and underfitting. Hence, whenever we combine multiple models then we have to make sure that there is balance of these uncertainties to avoid compounding errors. It will help to achieve accurate prediction (Schultz et al., 2021).

Computational Overhead: When we are dealing with multiple predictive models then it increases the computational overhead. Hybrid model combines NWP with machine learning. They need to process large datasets and they have to run numerous iterations until a desired accuracy of prediction is achieved. While doing so it performs complex calculations. All of these require significant computational power and resources (Rasp et al., 2018).

Addressing the Need for a Robust, Hybrid Model Combining Multiple Predictive Methods to Enhance Accuracy

Since, we have talked about so many limitations. Now there is a need for a robust hybrid model that integrates multiple predictive models. It enhances the accuracy and reliability of weather forecasts:

Leverage the Strengths of Different Models:A hybrid model is a combination of traditional NWP models and machine learning models. It leverages the strengths of different forecasting models and compensate for their individual weaknesses. If we take an example then machine learning models eare pretty effecient in recognizing patterns in large and complex datasets. That's how they make data-driven predictions. On the other hand NWP models are effective in capturing the physical and dynamical processes of the atmosphere using mathematical calculations (Chattopadhyay et al., 2020). When we combine these two methods then a hybrid model is built. This hybrid model can provide more comprehensive and accurate forecasts (Rasp et al., 2018).

Improving Short-Term and Extreme Weather Forecasts: Hybrid models have shown tremendous improvement in forecasting short-term weather forecasts. It also excels in predicting extreme weather events such as thunderstorms, hurricanes etc. There are deep learning methods such as convolutional neural networks (CNNs) and recurrent neural networks (RNNs). These methods can be used to identify and predict patterns in high-resolution weather data that traditional models may miss (Sønderby et al., 2020). When we integrate these deep learning methods with NWP models then hybrid models can achieve higher accuracy. This high accuracy will ultimately help in predicting localized and short-term weather events such as rain, hailstorm etc.

Adaptation to Localized Conditions: Traditional statistical methods were not generic but hybrid models can be customised. So that it fits to specific regions or conditions. This is done using localized data to improve the accuracy of forecasts. This flexibility or generic nature of the model is crucial for regions with unique weather patterns such as coastal areas which are prone to hurricanes . It is also quite effective in urban areas which are affected by heat islands (McGovern et al., 2017). Once we combine local data with using ML models to fine-tune predictions then hybrid models can deliver more precise and context-specific forecasts.

Enhancing Computational Efficiency: Since past couple of decades there has been a significant development in machine learning techniques. Techniques such as transfer learning and ensemble methods have been developed. it can be used to optimize the performance of hybrid models. These techniques can reduce the computational cost. The computational cost is high because of running high-resolution forecasts. It makes them more feasible and accessible (Schultz et al., 2021).

This thesis aims to develop a robust hybrid model which comprises various machine learning models. By doing so, we will get best possible accuracy for weather predictions. It will be achieved by leveraging the strengths of different models and addressing the challenges of integration.

1.3 Objectives of the Study

Primary Objective

Develop a Robust Predictive Model by Integrating Multiple Data Science Techniques to Improve Weather Forecasting Accuracy

The primary objective of this research is to create a hybrid predictive model which is a combination of various ML methods and data science techniques. So that the accuracy of weather forecasting increases. This study also aims to highlight the limitations of traditional weather forecasting model. By combining the strengths of different predictive methods such as machine learning algorithms (Logistic regression, decision tree, XGBoost, Random Forest) and statistical models, into a unified framework. The focus will be to build a comprehensive model that can understand complex and non-linear patterns in weather data. Once it has understood the pattern then provide accurate short-term and extreme weather forecasts. We are going to integrate the model to Leverage the advantages of both data-driven and physics-based approaches. It will ensure that the model captures both the hidden pattern in atmospheric dynamics as well as the complex interactions within the data. Ultimately we are going to Improve the prediction accuracy of key weather parameters such

as temperature, precipitation, wind speed, and humidity. This will be done particularly for short-term forecasts and extreme weather events.

Secondary Objectives

To achieve the primary objective, the following secondary objectives will be done.

Evaluate the Effectiveness of Different Predictive Models (e.g., XGBoost, Random Forest, Ensemble Learning)

The first secondary objective is to assess the performance of various machine learning and statistical models in the context of weather forecasting. This will involve conducting a comprehensive review of existing predictive techniques such as decision tree, logistic regression as well as including ensemble learning methods such as XGBoost & Random Forest. Then I will be evaluating each of these models using relevant metrics such as precison, recall, F-1 Score and accuracy scores to determine their strengths and weaknesses in predicting different weather variables (Schultz et al., 2021; McGovern et al., 2017). We will also plot the confusion matrix.

Develop a Methodology for Integrating These Models

The second secondary objective is to formulate a methodology for combining the predictive models into a cohesive hybrid framework. This will involve establishing protocols for validating the hybrid model including cross-validation techniques such as GridSearchCV. This cross validation is done to ensure its robustness and reliability across different weather conditions and geographical regions (McGovern et al., 2017).

1.4Research Questions

This thesis aims to answer the following questions.

- What Are the performances of traditional ML Techniques?
- How Can Data Science and Machine Learning Techniques Be Integrated to Improve Forecasting Accuracy?
- Which Combination of Models (e.g., XGBoost, Random forest etc) offers the Best Performance for Specific Weather Parameters?

1.5 Significance of the Study

In this thesis, we will be discussing its contributions and impacts in the field of weather forecasting and how meteorological department can benefit from it. We will also discuss the important role played by data science, machine learning and AI in more accurate weather forecasting. This increased accuracy will ultimately help lots of sectors such as agricultural, aviation, disaster management etc. We will be developing a novel approach to weather forecasting that integrates multiple predictive models which is also called as hybrid model..

Contribution to Meteorology and Data Science

A Novel Framework to Weather Forecasting

We are going to combine various predictive models and build a hybrid model for weather forecasting. Basically, we will be leveraging strengths of each of the predictive model. We are doing it this way because NWP models faces limitations in prediction with accuracy. That's because NWP uses physics laws and equations which doesn't work well with non-linear or short term weather phenomena (Bauer et al., 2015). When we implement predictive models such as XGBoost, Random Forest, ensemble method etc then the accuracy of weather forecasting increases significantly (Rasp et al., 2018). By combining the best features or strengths of different machine learning models into a hybrid framework. It show cases a novel framework. Using this novel framework the predictive model can handle uncertain weather condition and even short term and extreme weather forecast such as thunderstorm or floods. This is very critical for decision making (Chattopadhyay et al., 2020). This thesis provides a research methodology foundation for future R&D. It will pave the path for application of hybrid models in meteorology by offering new avenues for exploration and innovation in the field.

Potential Practical Applications

Enhanced Forecasting Accuracy and Sectoral Benefits

The practical applications of enhanced weather forecasting accuracy are vast, with potential benefits across multiple sectors. We shall discuss few of the important sectors in following section.

Agriculture: More than half of the world population depends on agriculture either directly or indirectly. Hence, it becomes very important for accurate weather predictions. That's how farmers or agriculturist are going to plan planting and harvesting schedules, managing irrigation and protecting crops from extreme weather events such as frost, drought, or heavy rainfall. For example, if there is a prediction of moderate to good rain then farmers will not spend on irrigation. Improved forecasts could help farmers make better-informed decisions. Once they are informed then it would ultimately help them to increase crop yields and reducing losses (Hatfield & Prueger, 2015).

Aviation: Aviation sector wan not that big 30 years ago. But now it's one of the booming sectors with opening of more airports and launching of new flight routes across the world. Weather is a significant factor in aviation safety and efficiency. Enhanced forecasting can lead to better flight planning, reduced fuel consumption, and fewer delays or cancellations. Accurate predictions of turbulence, icing, and severe storms can improve both safety and operational efficiency (Kulesa, 2003).

Disaster Management: With the recent climate change we have seen extreme weather conditions across the world. Be it extreme heatwave in northern part of India and tornadoes in American continent. A Reliable weather forecasts are critical for disaster management. A quick response can only be taken if the department is pre informed about the weather forecasts. Improved forecasting can enable early warnings and quick evacuations from the affected areas. It will help in reducing the impact of disasters on human lives and property (Kruk et al., 2010).

Energy Sector: In the renewable energy industry weather forecasts are vital for energy management. Especially in sectors like renewable energy where solar and wind power generation depends on weather conditions. If we have the accurate weather forecast then we can enhance grid management, optimize energy production and balance supply and demand more effectively (Olauson, 2018).

Societal Impact

Improved Preparedness and Reduced Economic Losses

Apart from the industry specific impact weather forecast also plays an important role in the societal impact. An improved weather predictions can enhance public safety, reduce economic losses and contribute to overall well-being of the society. Highlighting few of those are as follows:

Public Safety: Timely alerts and warnings which allow individuals, communities and authorities to prepare for adverse weather conditions is only possible by an accurate weather forecast. A timely alert will contribute in minimizing harm and loss of life. Accurate forecasts can help to plan and take informed decisions regarding evacuations, school closures and emergency services deployment (Dow & Cutter, 2000).

Economic Impact: Billions pf dollars of economic damage occurs whenever a significant natural disaster strikes like a Tsunami or floods or cloud bursts. Weather-related disruptions cause substantial economic losses worldwide. It impacts sectors such as agriculture, transportation, insurance and tourism. More accurate forecasts can help mitigate these losses by enabling better planning and risk management. For example, timely warnings can prevent damage to infrastructure, reduce insurance claims and minimize business disruptions (Hallegatte, 2012).

Climate Change Adaptation: There has been a debate since long time about the impact of climate change. We need a reliable weather forecast model as there is an increase in the extreme weather conditions across the world. We need to work on enhanced models that can help communities and governments better understand and prepare for climate-related risks. We need to find a balanced way in supporting adaptation strategies that protect lives, livelihoods, and ecosystems (IPCC, 2018).

Chapter 2: Literature Review

The **Literature Review** chapter examines the historical development of weather forecasting, the emergence of modern data science approaches, and the integration of machine learning techniques with traditional forecasting models. It also discusses the limitations of existing models and identifies gaps in current research, justifying the need for a new hybrid approach to improve weather forecasting accuracy.

2.1 Evolution of Weather Forecasting

2.1.1 Historical Background of Weather Forecasting

Weather forecasting goes back to thousands of years during ancient civilizations. They used empirical methods. The Babylonians which dates back to 650 BC. They were one of the first to analyse weather patterns. They basically studied cloud formations and other signs such as extreme heat or cold to predict the weather. It was this practice that laid the stone for early weather prediction. (Harper, 2012). There were Greeks who were influenced by the natural philosophy of Aristotle. They developed primitive theories on weather patterns and their causes, which were compiled in Aristotle's treatise, *Meteorologica* (Hann, 1903). This text mentions about the earliest attempts to explain weather phenomena such as rain, snow or heat. Although there were many misconceptions if we compared it with modern weather prediction studies.

Until scientific instruments such as barometer and thermometer were discovered, weather prediction remained largely empirical and based on folklore. As soon as these instruments were invented around the 17th century, we saw a huge improvement in the weather prediction because now pressure and temperature could be measured (Hann, 1903). These innovations of barometer & thermometer provided a more quantitative basis for weather forecasting. When telegraph was invented in the 19th century, the weather related information could be transmitted at a rapid pace over large distance. It allowed the development of 1st weather maps. Also, meteorological department were established in various countries to monitor weather (Harper, 2012).

2.1.2 Implementation of Numerical Weather Prediction (NWP)

NWP uses the mathematical and statistical models of the atmosphere to predict the weather. It does that by taking historical and current data into consideration. It uses mathematical equations to simulate weather changing process, based on various laws from fundamental physics, including the Navier-Stokes equations (Lorenz, 1963). Vilhelm Bjerknes, who was a norwegian physicist and meteorologist, has made tremendous contributions in weather forecasting using laws of physics. However, it was not practically implemented until 1940s only after there was an advancement in the computer technology. (Kalnay, 2003).

The first successful NWP experiment was conducted by a team led by John von Neumann and Jule Charney in 1950 using the ENIAC computer. ENIAC basically uses a very simplified approximation to the equations which affect the weather. The model which was developed by Neumann and Charney marked the beginning of a new era in meteorology (Thompson, 1957). After that for next few decades, more and more features or variables were included to build weather forecasting model so that the prediction would be more accurate. Those features were wind speed, temperature, humidity etc. (Kalnay, 2003). All this was possible only because of the evolution in the computer simulation.

Even though NWP was quite effective but it also had few limitations. They required real time data having high quality to make better predictions but it was a serious challenge to get such kind of data especially in remote places. (Bauer et al., 2015). Moreover, the accuracy of NWP models is constrained by their spatial and temporal resolution; finer resolutions require immense computational resources, which are often unavailable (Sun et al., 2014). They struggle with the accurate representation of small-scale and rapidly evolving weather events, such as thunderstorms, which can significantly affect local weather conditions but are challenging to predict with large-scale models (Lorenz, 1963).

2.1.3 Evolution and Current Status of Weather Forecasting Models

We have seen a huge development in the weather prediction accuracy since 1970s because of the refinement of NWP models. It's only possible because of technological advancements and a better understanding of atmospheric dynamics. Globally NWP models are used by various weather forecasting agencies. One of them is European Centre for Medium-Range Weather Forecasts (ECMWF) and the other one is United States National Centers for Environmental Prediction (NCEP), have improved forecast accuracy significantly over the past few decades (Kalnay, 2003). The models which are used by ECMWF and NCEP utilize supercomputers and vast observational datasets which are of high quality and are generated on real time basis. These data usually comes from satellites, weather stations, and other remote sensing technologies to predict atmospheric conditions up to several days in advance. You can see the weather of any place for at least next 7 days in advance just by a click of an app on smartphone.

Today, NWP models continue to be the pillars of weather forecasting. Not only we use NWP models for daily weather prediction but also to monitor climate change over a longer period of time. However,

the models' deterministic nature, relying strictly on initial conditions and physical laws, presents challenges in handling uncertainties and non-linear atmospheric behavior (Sun et al., 2014).

2.2 Modern Data Science Approaches to Weather Forecasting

2.2.1 Introduction to Data Science in Weather Forecasting

There has been a development of new methodologies for weather prediction in the past couple of decades. This has been possible because of data science and artificial intelligence. These new methodologies can predict the weather more accurately. Using data science, we can leverage large volume of data in a real time basis. It combines this huge set of data with computational techniques of machine learning to identify pattern from raw data and extract insights . These insights are used to predict the future weather more accurately (McGovern et al., 2017). If we use old NWP models, it only used laws of physics. But nowadays, by using machine learning and deep learning which are data science methods, we can learn patterns even from non relational data (Chattopadhyay et al., 2020).

We use machine learning models to understand the pattern in the weather data, especially when the data is very complex. For example, neural networks which is a type of machine learning model, can detect non-linear patterns in data which the traditional models might miss (Goodfellow et al., 2016). To increase the accuracy of NWP models, machine learning methods are used. Machine learning models are also used to correct model biases and improve short-term forecasts, such as precipitation and temperature predictions (Rasp et al., 2018).

2.2.2 Common Machine Learning Techniques in Weather Forecasting

We shall discuss now common machine learning techniques which are used in weather forecasting.

- **Decision Trees and ensemble methods**: Decision trees are a powerful ML model which can handle classification as well as regression analysis. Random Forests is an ensemble method which is built on multiple decision trees. They have been successfully used to predict severe weather events, such as tornadoes and hailstorms (Lagerquist et al., 2017). Gradient Boosting Machines (GBMs), such as XGBoost, which are nothing but improved version of decision trees. It sequentially train models to correct errors made by previous iterations ultimately enhancing predictive accuracy (Breiman, 2001). It performs hyperparameter tuning to get the best possible parameters which gives the highest accuracy.
- **Artificial Neural Networks (ANNs) and Deep Learning**: ANNs have gained significant importance in weather prediction. Convolutional Neural Networks (CNNs) are used to analyze spatial data. These spatial data comes from images sent by the satellites. CNN is also used to detect cloud detection and to track storm (Shi et al., 2015). Recurrent Neural Networks (RNNs) and Long Short-Term Memory (LSTM) networks are suitable for data which varies over time. This type of data is called as time series data. Hence, Rnn and LSTM are

most ideal for predicting weather patterns over time (Sønderby et al., 2020). For example, LSTMs have been used to forecast rain, humidity, precipitation and temperature changes by learning temporal dependencies in sequential weather data (Chattopadhyay et al., 2020).

- **Support Vector Machines (SVMs)**: SVMs are used for classification problems where data points are not separable linearly. They are used to predict weather states, such as distinguishing between different types of precipitation. It is also used to identify weather patterns which are associated with severe storms such as typhoon or tornado (Kotsiantis, 2007). SVM works very well in high-dimensional spaces. By high-dimensional spaces it means when the number of dimensions are more than the number of samples. This quality makes SVM valuable for niche applications in meteorology.
- **Ensemble Learning**: Ensemble learning means combining multiple models to achieve better performance than any single model could achieve on its own. There are techniques such as bagging and boosting are used to combine outputs from different ML models. It increases accuracy of the prediction and robustness of model to different types of data. (Zhou, 2012). Ensemble learning methods are used in weather forecasting to integrate different types of models. It captures a broad range of atmospheric feautures and reduces the uncertainty which are inherent when we use individual model predictions.

2.2.3 Examples of Machine Learning Applications in Weather Forecasting

There has been quite a few studies which have demonstrated the efficacy of ML models in weather forecasting. The study done by Lagerquist et al. (2017) utilizes Random Forests to predict severe convective storms. His study achieved higher accuracy than traditional NWP models. He did study to predict tornadoes and hailstorm. Another study done by Scher (2018) employed deep learning to approximate the outputs of NWP models. This study showed that neural networks can replicate NWP results with much lesser computational expense.

The use of ML models is much beyond the prediction of common weather parameters like temperature and precipitation. They have also been used in specialized applications, such as predicting air quality and energy demand, which are closely linked to weather conditions (Samek et al., 2017). There are hybrid models which combine NWP models with ML techniques. These hybrid models have been developed to predict solar radiation more accurately. It helps in the generation of renewable energy from power grid (Klein et al., 2015).

2.3 Integration of Machine Learning with Traditional Forecasting Models

2.3.1 Hybrid Models: A New Paradigm

When we integrate machine learning with NWP models then it gives a new paradigm in weather forecasting. Hybrid models leverages the strengths of both ML as well as NWP. In Hybrid model the physical understanding is provided by NWP models and the data-driven insights are offered by

machine learning (Rasp & Lerch, 2018). This combination can take several paths. For example, machine learning models can be trained on historical errors in NWP forecasts. It allows ML models to correct these errors in real-time (Scher, 2018). On the other hand, ML models can be used to downscale NWP outputs. When we do downscaling then it provides higher-resolution forecasts for any given specific regions or applications (Chakraborty et al., 2020).

Hybrid models are preferable when we are not getting suitable accuracy using individual model. NWP models are most suitable in predicting large-scale weather patterns which involves large scale pattern. It is governed by physical laws. However, they are quite inefficient with smaller-scale phenomena, such as thunderstorms or fog. These are influenced by complex, local interactions which may not be fully captured by the NWP models. In such scenario, we have to use machine learning models. ML has the ability to extract insights from complex dataset. ML can complement NWP models by providing insights into these smaller-scale processes to predict thunderstorm or fog (Lagerquist et al., 2017).

2.3.2 Recent Advances in Hybrid Models

The recent studies concluded by researchers have demonstrated the potential of hybrid models in improving weather forecasting. The hybrid model developed by Sun et al. (2014) combines NWP outputs with a neural network to predict short-term precipitation. This method achieved better results over traditional methods. Another study was done by Rasp et al. (2018) where he created a neural network-based tool that corrects systematic biases in NWP forecasts. It resulted in more accurate temperature and precipitation predictions.

There are other studies as well which have explored the use of ensemble techniques to combine multiple ML models with NWP outputs. A neural weather hybrid model developed by Sønderby et al. (2020) was named MetNet. It uses a combination of CNN and RNN. This combination is used to process historical weather data and NWP outputs both. The accuracy given by this hybrid model in precipitation forecasting was quite high. achieving state-of-the-art performance in precipitation forecasting.

2.3.3 Machine Learning Models Used for Weather Forecasting

Decision Tree

Decision trees is a very powerful machine learning models which is used for classification and regression tasks. It splits the dataset into subset recursively on the basis of certain specified decision. This decision is specified from the input features. Features such as temperature, pressure, humidity, precipitation and wind speed can be used to classify weather condition. We can use these conditions in weather forecasting model. The main advantage of decision trees is their interpretability. They provide a clear and understandable path from input features to predictions. It is very crucial in meteorology which requires transparency. But there are certain challenges associated with the decision trees. Overfitting is one of them. Overfitting usually happens with the noisy data or

when the tree has big depth. In weather forecasting or in any types of prediction overfitting can be problematic. As it causes the model to perform poorly on unseen data or under different weather conditions. Therefore, decision trees are often pruned or used in combination with other models to enhance performance (Breiman et al., 1984).

Logistic Regression

Logistic regression is used in binary classification tasks. It is used in weather forecasting to predict a particular weather events such as rainfall, snowfall or storms. In logistic regression, the probability of a certain class (e.g., whether it will rain or not) is modeled using a logistic function by mapping the input features to a value between 0 and 1. The model assumes a linear relationship between the input variables and the log-odds of the outcome. This linear relationship makes it interpretable and easy to implement. It's very simple and interpretable. That makes it a preferred choice for meteorological applications that require transparency. But there are certain limitations associated with logistic regression. It assumes that the relationship between the predictors and the outcome is linear. Hence it doesn't work well with non-linear features of atmosphere. We use techniques such as regularization to overcome this limitation. We can also combine it with non-linear models (Menard, 2002).

Random Forest

Random Forest is an ensemble learning method. It builds multiple decision trees and combines their results to improve prediction accuracy. This approach is highly suitable for weather forecasting because it can handle large datasets and complex, non-linear relationships between variables. In weather forecasting, The advantage with this model is that it can be used to predict both categorical (e.g., rain, snow) and continuous outcomes (e.g., temperature, precipitation amount).

The main advantage of Random Forest is its robustness. It also reduces overfitting, which is a common issue with single decision trees. Random Forests reduces the risk of overfitting by averaging the predictions from multiple trees. It makes itself better to unseen weather conditions. Random Forest can also handle missing dat. It works well even when the input features have varying degrees of importance. There is one limitation associated with Random Forest which is that it can be computationally expensive, especially for large datasets common in weather forecasting. Despite of it Random Forest remains one of the most popular machine learning models in meteorology due to its flexibility, scalability, and ability to deliver high-accuracy predictions (Breiman, 2001).

K-Nearest Neighbors (KNN)

K-Nearest Neighbors (KNN) is instance-based machine learning algorithm commonly used for both classification and regression tasks. KNN is a non-parametric model. KNN predicts the weather conditions by looking at the closest data points which is called as neighbors in the feature space. Then using their outcomes to make the prediction for a new input. For example, KNN can forecast the likelihood of rain or temperature levels based on the similarity of recent weather data to historical observations.

The strength of KNN is in its simplicity. It also depends on its ability to find pattern in non-linear relationships in data without the need for an explicit training phase. This strength makes it ideal for dynamic and complex systems like weather forecasting because in variables which affects weather conditions may not always follow linear patterns. KNN performs better when local weather conditions are heavily influenced by nearby or recent atmospheric patterns. It can capture by considering the proximity of data points. Like other models KNN also has some limitations. These limitations are basically high computational cost and sensitivity to the choice of distance metric. It usually happens when there is large datasets of meteorological studies. KNN may struggle with noisy or irrelevant features like Random Forest model. It can degrade forecasting performance. Even though there are these many limitations Despite KNN is used in hybrid models and ensemble techniques to improve the accuracy of weather predictions by combining its predictions with other models (Altman, 1992).

Support Vector Machine (SVM)

Support Vector Machines (SVM) are supervised learning models. They are used for classification and regression tasks both in various domains. Weather forecasting is one of those domains. The core idea behind SVM is to find a hyperplane that best separates the classes in a high-dimensional space. In reference of weather forecasting these classes rae different types of weather conditions such as fog, rain, cloud etc. SVM can handle both linear and non-linear classification problems by using kernel functions. Kernel function transforms the input data into a higher-dimensional space where it becomes easier to separate classes. SVM can be used to classify weather patterns. It can also be used to predict numerical outcomes like temperature and precipitation. Its biggest strength is in its ability to handle high-dimensional data, which is crucial in meteorology where numerous features such as humidity, pressure, wind speed influence the outcome. The use of kernel functions, such as radial basis function (RBF), allows SVM to capture complex, non-linear relationships between atmospheric features that traditional statistical methods may miss. SVM is particularly robust to overfitting. Hence, it is very suitable for short-term and localized weather predictions. There are certain limitations as well which are associated with SVM. One of those limitations is that it can be computationally intensive and sensitive to the choice of hyperparameters like the regularization parameter (C) and kernel function. The other limitations is that SVM models can be difficult to interpret compared to simpler models like decision trees or logistic regression. These limitations may limit SVM use in certain meteorological applications that require transparency (Cortes & Vapnik, 1995; Smola & Schölkopf, 2004).

2.4 Limitations of Existing Models and the Need for a Hybrid Approach

We have discussed so far that NWP and ML models both have made significant development in weather predictions. But still they possess limitations. Hence, we need a hybrid approach where we can leverage the strengths of NWP and ML models both. NWP models have limitations because of

their computational costs data quality. It also faces challenge while accurately representing complex and small scale atmospheric features (Bauer et al., 2015). Contrary to that, ML models are capable of handling large datasets and learning from historical patterns. However, they have limitations in terms of physical interpretability and generalization capabilities of NWP models (McGovern et al., 2017).

A hybrid approach seeks to leverage the strengths of both models while compensating for their weaknesses. By integrating NWP and ML models, it is possible to create a system that benefits from the physical understanding provided by traditional models and the data-driven adaptability of machine learning. This approach can improve the accuracy, reliability, and resolution of weather forecasts, particularly for short-term. It can also be used to improve the prediction of localized events where traditional models often fall short (Rasp & Lerch, 2018).

2.5 Summary and Conclusion

Since the beginning of the weather forecasting from Babylonian era till now, we have seen significant development in terms of accurate prediction. Especially in last 50 years, there has been development at rapid pace because of advancement in computer technology. Now, we get high resolutions, high quality data sourced directly from the satellite. In past couple of decades, we have started implementing data science techniques to build machine learning model and hybrid models. These models can perform large mathematical computations in few minutes and give accurate results. There are certain limitations of ML models and NWP models test's why we combine these two and make a hybrid model which leverages the strengths of both of them. In this chapter we have done the literature review of historical content, current methodologies and emerging trends in weather forecasting. It provides a foundation for the next chapter where we will discuss about the model development and methodology for improved weather prediction.

3. Chapter: Methodology

3.1 Introduction

This chapter focusses on the research methodology which has been followed in this thesis. It basically starts with the performance of various machine learning models such as Decision Tree, Logistic regression, Random Forest, KNN, SVM and XGBoost. The primary goal is to achieve the highest possible accuracy by employing a structured process that starts with data collection, preprocessing, feature selection and finally model is developed. The baseline models explored in this thesis are Decision Tree, Logistic regression, Random Forest, KNN, SVM. These models were selected for their varied strengths in handling large dataset with non-linear pattern. The research methodology integrates modern techniques to refine performance. This includes the application of GridSearchCV which provides hyperparameter tuning and the Synthetic Minority Over-sampling Technique (SMOTE). SMOTE is used to handle class imbalances which are present in datasets. First, I started with the baseline models which are Decision Tree, Logistic regression, Random Forest, KNN and SVM. Then I used XGBoost Each of these techniques is explored in depth to enhance the predictive capabilities of the models. To evaluate the performance of the models, a comprehensive evaluation framework is showcased employing key performance metrics such as accuracy, precision, recall, F1-score, and the AUC-ROC curve. These metrics give a fair understanding of model performance. It ensures that the results are not only accurate but also balanced across different classes of weather forecasting.

3.2 Data Collection and Preprocessing

The Weather dataset was selected as the primary dataset this thesis due to its extensive collection of weather events and features. I applied detailed preprocessing techniques to the raw text data dealing with missing and null values. It enabled accurate analysis and improved model performance.

3.2.1 Sources of Data

Source and Description: The weather data is available on Kaggle. It comprises over 8,784 different occasion of weather events. Kaggle is an online platform which is widely known for datasets for research and development purposes. It is also used for science competition. The dataset selected for this thesis includes Date, Temperature, humidity, wind speed, visibility, pressure and weather events. This dataset makes it ideal applying predictive models. The large size and diversity of the dataset allow for thorough exploration of various machine learning techniques.

3.2.2 Data Types and Variables

Weather Variables:

Date/Time: It specifies the date and time when the weather conditions were recorded. It's needed to assess the variation in the weather over a period of time. It can also be used to study the correlation between weather pattern and time of the day. Though we don't need it in our model.

Temperature: This column represents the temperature in degrees Celsius, which is a fundamental aspect of weather. Temperature can have a significant impact on various aspects of life, such as energy consumption, transportation, and outdoor activities.

Dew Point: This column represents the dew point temperature, which is the temperature at which the air becomes saturated with water vapor and dew or frost begins to form. Dew point is an important factor in determining the perceived temperature and humidity levels.

Humidity: This column represents the relative humidity, which is the percentage of water vapor in the air compared to the maximum amount of water vapor the air can hold at a given temperature. Relative humidity is an important factor in determining the perceived temperature and comfort level.

Wind Speed and Direction: This column represents the wind speed in kilometers per hour, which can have a significant impact on weather patterns, such as the spread of precipitation or the formation of storms.

Pressure: This column represents the atmospheric pressure in kilopascals, which can have a significant impact on weather patterns, such as the formation of high and low-pressure systems.

Weather: This column represents the weather conditions, which can be a categorical variable with values such as "sunny", "cloudy", "rainy", etc. Weather conditions can have a significant impact on various aspects of life, such as energy consumption, transportation, and outdoor activities.

3.2.2 Methods for Data Cleaning and Preprocessing

The following steps were undertaken to prepare the dataset for weather forecasting. These processes ensure that the data was clean, balanced and it was suitable for model training.

Feature Selection: A standard weather column was created. Those columns were dropped which were not required such as Date/Time and weather.

Standardizing category: We have converted weather category into standardized format which I s names std_weather. This conversion simplified weather forecasting task by focusing on categorical values only.

Encoding Categorical data: The create_list function takes a string x as input, splits it into a list of lists using the comma and space as separators, and then flattens the list of lists into a single list of weather conditions. The Get_weather function takes a list of weather conditions as input and returns a standardized weather category based on the conditions present in the list.

Scaling features: Scaling features involves transforming numerical data to a common range, usually between 0 and 1, to prevent features with large ranges from dominating the model. This improves model performance and prevents feature dominance.

Balancing the Dataset: Class imbalances were addressed by employing oversampling techniques such as SMOTE (Synthetic Minority Over-sampling Technique). It helped ensure that the model would not be biased towards the majority class.

Handling Missing Data & Duplicate values: The dataset was checked for the presence of any missing values and duplicate values. However, the data was quite clean.

Splitting the dataset: We have split the dataset into training and testing dataset. By setting test_size=0.2, we've allocated 20% of the data to the testing set and 80% to the training set. The random_state parameter ensures that the split is reproducible, and the stratify parameter ensures that the class distribution in the target variable Y is preserved in both the training and testing sets.

3.3 Predictive Techniques Used

Following section deep dives into the machine learning models techniques used for weather forecasting in this thesis. These models include decision trees, logistic regression, random forest, K-Nearest Neighbors (KNN), Support Vector Classifier (SVC), and XGBoost. These models have been chosen based on their ability to handle large weather datasets and its predictive accuracy,

3.3.1. Decision Trees

Overview of the Algorithm

Decision trees are supervised learning techniques. They don't have any parameters. They can be used for classification and regression both kind of tasks. It basically splits the dataset into subsets recursively based on the feature what we provide. It could be either information gain or gini index. It forms a tree like structure when it splits the dataset. That's where its name come from. This tree structure eventually helps in making predictions based on conditions learned from the training data.

Key Parameters

- **Criterion**: Determines the function used to measure the quality of a split, such as Gini impurity or entropy (information gain).
- **Max Depth**: Controls the maximum depth of the tree, which affects both the model's complexity and ability to generalize.
- **Min Samples Split**: The minimum number of samples required to split an internal node, helping to control overfitting.

Suitability for Weather Data

Decision trees can be interpreted very easily. As it can be works well very easily whether categorical data or continuous data. For example, they can be used to classify different weather conditions based on temperature, humidity, wind speed, etc. But when decision trees encounters complex dataset then they tend to overfit. We can tackle overfitting by pruning or restricted in depth. In weather forecasting, the simplicity of decision trees allows them to model relatively straightforward

weather patterns (e.g., sunny vs. rainy days) while providing insights into feature importance (e.g., temperature might be the most significant predictor for a rain event). Singh et al. (2020)

3.3.2. Logistic Regression

Overview of the Algorithm

Logistic Regression is a simple linear model. It is used for binary and multi-class classification tasks. It predicts the probability of a given input whether it belongs to a certain class or not. It does it by applying the logistic function (or sigmoid function) to the linear combination of input features.

Key Parameters

- **Regularization (C)**: This controls the penalty for overfitting. A smaller value of C increases the penalty for large coefficients, helping to reduce overfitting.
- **Solver**: Determines the optimization technique used (e.g., 'liblinear' for small datasets or 'saga' for large ones).

Suitability for Weather Data

Logistic regression is ideal for binary weather classification problems. It is used to predict whether it will rain or not based on variables like temperature, pressure, and humidity. Because of its probability nature it predicts in output the likelihood of different weather events. Logistic regression has limitations when it assumes a linear relationship between the input features and the output. It limits its effectiveness for complex, non-linear weather phenomena like thunderstorms or hurricanes Mishra et al. (2019)

3.3.3 Random Forest

Overview of the Algorithm

An Ensemble Learning technique, Random Forest enhances the prediction capability through construction of a multitude of decision trees at training and averages the outputs. Every tree in the forest classifies an input independently, and the output is decided by a majority vote (for classification) or average (for regression). With this ensemble method, Random Forest is more robust than individual decision trees as it lowers overfitting and improves generalization.

Key Parameters

Number of Trees (n_estimators): The number of trees in the forest. Larger number improves performance generally, at the expense of additional computational resources.

Max Features: This is Number of features to consider when looking for the best split at every node. Having some randomness in this selection can still make trees more dicersified than totally searching all features, and thus better performance of the model.

Suitability for Weather Data

Random Forest works especially well for weather prediction because it can handle immense, multidimensional datasets and many possible predictors. The approach works for high-dimensional weather data because these typically consist of temperature, humidity, wind speed and related metrics. This property makes Random Forest ideal for modelling noisy data and the sometimes

complex interdependence between independent weather variables. Thus it is suitable for predicting complicated weather events like storms and precipitation levels where feature interactions are high and non-linear behaviours are typical.

Literature Reference

Random Forests (first described by Breiman in 2001) has been a fundamental building block for many data scientists, and has found wide spread use in fields such as environmental science and meteorology. More recently, Kumar et al. In their own studies, (2021) have verified the performance of Random Forest for weather event prediction such as rainfall and temperature.

3.3.4. K-Nearest Neighbors (KNN)

Overview of the Algorithm

K-Nearest Neighbors (KNN) — K-nearest neighbors algorithm is a very simple, non-parametric and lazy learning. It predicts the class of a data point based on the cluster which has majority in first few nearest neighbors (k). For regression tasks, it is the average of values for k nearest points to make predictions.

Key Parameters

k Number of Neighbours(k): It specifies Number of Neighbors to consider when process for the classification or Decision Making algorithm. A low ‘k’ risks over-fitting, while a high ‘k’ may make the predictions too smooth.

Distance Metric: Distance metrics such as Euclidean, Manhattan and Minkowski. Model results can largely depend on the metric to compare distances.

Suitability for Weather Data

An application of KNN in weather forecasting: Suppose you have records about historical temperature and humidity conditions, you may employ K-Nearest Neighbours to find closest weather conditions to predict what will happen next. For example, you could use KNN to get a sense of the current weather by inputting the historical data of temperature, humidity and wind speed. On the other hand, KNN is slow for large, high-dimensional datasets and can be easily manipulated in weather datasets with measurement errors or missing values.

Literature Reference

Das et al. A recent study conducted by Hronček et al (2020) using KNN in weather forecasting found that the model performed relatively well, however computational intensity was high and data quality was inconsistent.

3.3.5. Support Vector Classifier (SVC)

Overview of the Algorithm

The Support Vector Classifier (SVC): A classification method that annotates data points according to the hyperplane with an optimal margin. SVC is capable of solving linear as well as non-linear

classification problems by the use of kernel functions like radial basis function (RBF) which maps your data to higher dimensional space where it can be easily classified by a linear classifier.

Key Parameters

Regularization Parameter (Controls the trade-off between achieving low error on the training data and reducing model complexity. A smaller C also allows a widest margin, the overfitting reduced.

Kernel: This tells what kind of hyperplane to be used for separating the data (e.g. linear, polynomial, RBF). The RBF kernel works well in cases of non-linear weather data.

Suitability for Weather Data

SVC performs better when you have a complex weather data that is not linearly separable. For instance, elaborated structures in the atmospheric data can be considered as part of the factors associated with predicting complex weather events (storms or heatwaves), which are well-captured by SVC and particularly with RBF kernel. On the other hand, SVC can be quite time-consuming on large datasets; something that is typical for meteorological data.

Literature Reference

Studies such as Patel et al. Bibliography(2018) can classify weather by using SVC with RBF kernel, but training costs are sometimes heavy if the set is large.

3.3.6. XGBoost

Overview of the Algorithm

Extreme Gradient Boosting (XGBoost) is a high-performance, scalable machine learning implementation of gradient boosting that optimizes for both the speed and performance of your model. It constructs the decision trees one at a time, using the residual errors from each successive iteration to form the new tree. XGBoost also uses the regularization techniques to reduce overfitting and better generalization.

Key Parameters

Rate: Controls the contribution of each tree to the final model This will again increase robustness of the model but it requires more trees to learn.

n_estimators : The number of trees to be built. The higher this number is the more accurate our model will be but also the higher training time.

Max Depth = How deep we allow a tree to go. A tree that is in deeper depth can hence capture complex patterns but it runs the risk of overfitting on the training data.

Suitability for Weather Data

XGBoost works really well with large and noisy weather datasets. The variety of multi-cloud features that XGBoost supports from handling missing data, non-linear interactions between features to complex atmospheric variables like temperature, windspeed and humidity only make it a better tool for modelling Big Data driven weather prediction model. It is extremely fast and efficient, which makes it ideal for use cases in the real-time weather prediction domain, where appropriate (fast)

results are crucial. Moreover, feature importance with XGBoost aids in determining the main factors affecting weather formations.

Literature Reference

One of the more eponymously principals in this field was XGBoost (Chen & Guestrin, 2016), and it has been additionally utilized for environmental data Parameter et al. Ali et al. (2021) also used the XGBoost algorithm for meteorological data and demonstrated better accuracy and timing compared to standard models.

Rationale for Model Selection

We have selected some models that are considered as simple (decision trees and logistic regression), some that are not robust (KNN and SVM) just for a comparison of the results, further using two advanced methods--random forest(Sloan, S., 1996) for raw predictive power in classification tasks and XGBoost(Cuiqiang Weng et al). All of the following models attack different aspects of weather data:

Decision Trees and Random Forest, Interpretability and non-linear systematic approach (common in weather forecasting).

Logistic Regression: It Provides a simple and probabilistic approach to binary classification. It is very useful for predicting weather events like rain or no rain.

KNN: We know that computational cost of KNN is expensive. But it can be used as a simple baseline model for predicting weather. It may use analogy with historical conditions.

SVC: It can handle complex and non-linear relationships among the data. This is what makes it suitable for predicting severe or rare weather events.

XGBoost: Most important features of XGBosst is it's high accuracy, speed and scalability. These features make XGBoost essential for handling large weather datasets and producing reliable forecasts.

Each model has been selected based on its alignment with the goals of this weather forecasting task, balancing ease of use, accuracy, and computational feasibility.

3.4 Model Development and Integration

In this section, we delve into the process of developing the machine learning models used in this research, detailing the steps taken to train each model, fine-tune hyperparameters, and validate their performance. We will also discuss how these models are integrated using ensemble techniques and the development of hybrid models. Finally, we explore the challenges encountered during implementation, including the computational demands, handling large datasets, and optimization techniques.

Step-by-Step Guide to Developing Each Model

The following steps outline the general process for developing each machine learning model, with a focus on weather data.

3.4.1. Data Splitting

Data splitting is the first crucial step in machine learning model development. It involves dividing the dataset into three parts:

Training Set (80%): This portion of the data is used to train the machine learning models, enabling the models to learn the relationships between the input features (e.g., temperature, humidity) and the target variable i.e. weather.

Test Set (20%): This data is kept aside to evaluate the final performance of the model. It acts as an unbiased estimator to measure the model's generalization ability on unseen data.

The data split is typically done using stratified sampling to ensure that each subset contains representative proportions of all classes in the weather data. For example, if the dataset includes different weather types (e.g., sunny, rainy, cloudy), stratified sampling ensures that each subset contains a balanced representation of these conditions.

3.4.2 Model Training

After this the models are worked on the training set but the data is first split to achieve this. This step entails proprocessing the input features with their labels then feeding them to the model. For instance:

Decision Trees and Random Forest: These models and algorithms are used for training and they divide the data also recursively based on any of the input feature in such a way that the homogeneity is preferred and minimizes measures such as Gini impurity or entropy.

Logistic Regression: This model is learned by estimating the linear regression equation which approximates the probability of binary weather possibilities in the form of logistic function.

K-Nearest Neighbors (KNN): KNN does not require any kind of training process to be followed by the algorithm. But in this sense, it just keeps training data and uses them to make a prediction based on the similarity of new data to the training samples.

Support Vector Classifier (SVC): This model works by training, in a sense, to identify the hyperplane which gives the greatest separation between different weather classes.

XGBoost: XGBoost applies gradient boosting technique whereby individualDecision Trees are sequentially built to minimize the error left out by the previous Decision Trees.

Training as mentioned earlier can be computationally expensive especially when working with large weather data for models such as SVC and XGBoost.

3.4.3 Hyperparameter Tuning

Hyperparameter optimization, in other words, hypertuning is a process of choosing values of parameters which are not learned within the training process. Examples include:

Decision Trees: Splitting attributes are maximum depth, minimum samples for split and criteria which is either Gini or entropy.

Random Forest: Three hyperparameters are specific for trees: the number of trees (n_estimators), the maximum depth of the tree and the minimum number of samples required to split an internal node (minimum_samples_split).

Logistic Regression: The first is the regularization parameter often denoted by C and the second is the choice of the specific solver which can be for instance 'liblinear' or 'saga'.

KNN: Number of neighbors (k) and distance measure(Euclidean or Manhattan).

SVC: These are regularisation constant (C), Kernel type (Linear, Polynomial or Radial basis function), and Gamma for the RBF kernel.

XGBoost: These include the learning rate (alpha), the number of trees (n_estimators), the model's maximum depth, and hyperparameters for regularisation (lambda and alpha).

In order to decide on the apt hyperparameters, the process of cross validation such as Grid Search is employed.

Grid Search: It is a method whereby the entire configurations in a manually predefined portion of the hyperparameter space are traversed. While it is indeed very helpful, it can turn out to be very repetitive and time-consuming.

3.5 Evaluation Metrics

When developing predictive models for weather forecasting, evaluating model performance is a critical step to ensure that the models produce accurate and reliable results. Different evaluation metrics are used to assess the effectiveness of the models based on the type of prediction task—whether it's a regression problem (e.g., predicting temperature or wind speed) or a classification problem (e.g., predicting if it will rain or not). In this section, we will discuss several commonly used evaluation metrics, including RMSE, MAE, accuracy, precision, recall, and F1 score, and justify why they were chosen for this weather forecasting task.

Explanation of Evaluation Metrics

3.5.1 Accuracy

Accuracy is a metric used primarily for classification tasks, which in the case of weather forecasting might involve predicting categories like rain/no rain, cloudy/sunny, or storm/no storm. Accuracy measures the proportion of correctly classified instances out of the total number of instances.

$$Accuracy = \frac{True\ positives + True\ Negatives}{Total\ number\ of\ predictions}$$

While accuracy is straightforward to interpret, it can be misleading for imbalanced datasets (e.g., if the majority of days are sunny and few are rainy, a model predicting "sunny" all the time would have high accuracy but poor performance on rainy days). Therefore, accuracy should be used in conjunction with other metrics, especially in scenarios where weather events are imbalanced (e.g., severe storms being rare).

3.5.2 Precision

Precision is a metric that focuses on the proportion of correctly predicted positive instances (e.g., rainy days) out of all instances that were predicted as positive.

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

Precision is particularly important in weather forecasting when predicting rare and significant events such as storms or heatwaves. For example, a model that predicts a storm is only useful if the predicted storms are actual storms, not false alarms. High precision ensures that when the model predicts an event, it is highly likely to occur, which is critical for real-world applications like emergency management (Breiman, L. 2001).

3.5.3 Recall (Sensitivity or True Positive Rate)

Recall measures the proportion of actual positive instances (e.g., actual rainy days) that were correctly predicted by the model.

$$Recall = \frac{True\ Positives}{True\ Positives + false\ Negatives}$$

Recall is essential in scenarios where missing positive instances (e.g., failing to predict a storm) could lead to significant consequences. In weather forecasting, high recall is crucial for models aimed at predicting severe weather conditions because it ensures that the model captures most, if not all, critical events.

3.5.4 F1 Score

The F1 Score is the harmonic mean of precision and recall, providing a single metric that balances the trade-off between the two. It is particularly useful when dealing with imbalanced datasets where one class (e.g., no rain) dominates the other class (e.g., rain).

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

The F1 score is beneficial in weather forecasting when both false positives (incorrectly predicting rain) and false negatives (failing to predict rain) need to be minimized. A high F1 score indicates that the model has a good balance between precision and recall, making it a useful metric for predicting rare weather events, such as storms or floods.

Justification for Selecting These Metrics

3.5.5 Relevance to Weather Forecasting

It is common practice to check the **accuracy** in testing dataset, especially for classification tasks like predicting weather categories(rain/no rain) but this metric needs a complement with other metrics, specifically when the dataset is imbalanced.

Precision and Recall are critical when predicting low-frequency high-impact weather events (i.e. storms or tornadoes) A model with high precision is that when it predicts an extreme event, a very high percent of the time it occurs — this makes sense from the point of view of reducing false alarms. High recall means that we will catch almost all severe events and only a few data points are being mistaken for the most disruptive patterns.

The F1 score is a well-balanced evaluation metric which takes into account both precision and recall, so for predicting rare weather events (where you want to minimize false positives and false negatives), it seems particularly appropriate.

3.5.6 Interpretability

Each of these metrics gives us a different angle on how good our model actually performs, so the more we have the better we cover it all:

While accuracy makes one number to measure the overall correctness, precision / recall on the other hand help us to understand how many of this specific predictions where actual relevant.

The F1 Score is useful when we want to seek a balance between our model makes precision and recall so that they figured out in priority together.

3.5.7 Comparative Analysis

This approach helps to make the comparison between models more thoughtful by using multiple metrics. For example:

A model that has high accuracy but low recall, for example, can return very accurate predictions of the most common class 0 weather but only detects a tiny fraction of the rare severe weather events. For rare but critical events like storms when the data is imbalanced which is most cases in weather forecasting tasks the F1 score gives a comparison metric between models.

3.6 Tools and Technologies

The successful implementation of machine learning models for weather forecasting requires the use of various tools, technologies, and platforms. In this section, we provide an overview of the tools and libraries used in this thesis, including Python, key machine learning libraries such as Pandas, Scikit-learn, and XGBoost. Additionally, we justify the selection of these tools based on their efficiency, ease of use, scalability, and community support.

Overview of Tools and Technologies

3.6.1. Python

For convenience, the foremost reason to choose this research is that Python is the primary programming language used for it. It is a high-level, open-source and versatile language which support multiple models like procedural, object oriented programming among others. Python has

very user-friendly syntax that is intuitive and fast to develop new libraries and easier to maintain(McKinney, W.2010).

One of the strengths of Python is its ecosystem for data science and machine learning, including:

Full-featured libraries for data manipulation and analysis (Pandas, NumPy),

Core machine learning and deep learning libraries (e.g., Scikit-learn),

The Seaborn package for much beautiful and powerful statistical data visualization.

These properties make Python the ideal programming language for building, evaluating, and deploying machine learning models like those we use in weather forecasts.

3.6.2 Pandas

Pandas is a powerful library for data manipulation and analysis. It provides data structures like DataFrames, which are ideal for working with structured weather datasets (Chen, T., & Guestrin, C. 2016). Pandas enables quick data cleaning, transformation, and analysis.

For example, it supports:

- Reading and writing data from/to various formats (CSV, Excel, SQL, etc.),
- Handling missing data, which is common in weather datasets,
- Grouping, filtering, and aggregating data for feature extraction and analysis.

Pandas was chosen for this thesis because of its ability to handle large weather datasets efficiently and its ease of use in integrating with other Python libraries.

3.6.3 NumPy

NumPy is the fundamental package for numerical computations in Python. It provides support for multi-dimensional arrays and matrices, along with a collection of mathematical functions to operate on these arrays. NumPy is used extensively in the background of machine learning models, especially for tasks such as efficiently handling large numerical weather datasets and performing mathematical computations required for model training and evaluation.

NumPy's array manipulation features make it indispensable for tasks such as feature scaling and transformation, which are essential in machine learning model development.

3.6.4 Scikit-learn

Scikit-learn is one of the most widely used machine learning libraries in Python. It offers a wide range of algorithms for both supervised and unsupervised learning, making it ideal for developing models like decision trees, logistic regression, random forest, KNN, and SVC. Key features include:

- **Preprocessing tools**: For scaling, normalizing, and splitting data,
- **Model development**: Implementation of various machine learning algorithms like decision trees, random forest, logistic regression, SVC, KNN, etc.,
- **Evaluation tools**: For calculating metrics like RMSE, MAE, precision, recall, and F1-score,
- **Hyperparameter tuning**: Using tools like GridSearchCV and RandomizedSearchCV for model optimization.

Scikit-learn's user-friendly API, comprehensive documentation, and active community support make it an excellent choice for this thesis.

d. XGBoost

XGBoost (Extreme Gradient Boosting) is a high-performance library for gradient boosting. (Pedregosa, F., et al. (2011). It is particularly well-suited for structured data like the weather datasets used in this thesis. XGBoost offers several advantages:

- **Efficient handling of large datasets**: It is optimized for performance and scalability, making it well-suited for large-scale weather data,
- **Regularization**: It includes L1 (Lasso) and L2 (Ridge) regularization to prevent overfitting,
- **Parallelization**: It supports parallel and distributed computing, speeding up training time significantly,
- **Cross-validation**: Built-in support for cross-validation to tune hyperparameters efficiently.

XGBoost is widely recognized for its accuracy and efficiency, making it a preferred choice for weather forecasting models that require high precision.

3.6.5 Seaborn

A frequently used library for data visualization in Python is Seaborn. For those who are a little bit more advanced but does not want to jump straight into Matplotlib, Seaborn is the way to go. These libraries are used to visualise following tasks:

Distributions and trends in weather data (e.g., temperature patterns over time),

Performance matrices(confusion matrix, roc curve…)

Plot the salient weather features (ranked in descending order of importance)

Since the libraries are fundamental to the thesis and data visualization is a key way to understand model results and communicate findings.

Chapter 4: Results and Discussion

4.1 Introduction

This chapter, therefore, overall presents the results obtained from the analysis of the weather dataset. It gives a fair discussion of the results obtained. This chapter shall focus on a descriptive analysis, correlation analysis, and visual exploration of data. The basis of this analysis is the weather dataset that consists of temperature, dew point temperature, relative humidity, wind speed, visibility, atmospheric pressure, and weather conditions. The chief objective is to get the pattern, relationship, and distribution in the data for drawing conclusions about the weather conditions.

4.2 Descriptive Analysis

The weather dataset has 8,784 rows with no missing values. Different information on various weather parameters is conveyed through the columns in this dataset. Descriptive statistics were calculated on numerical variables to summarize the central tendency, dispersion, and shape of the distribution.

Temperature: The low is -23.3°C, and the high was 33°C; the average is 8.8°C with a standard deviation of 11.7°C. It can also be seen that the temperature is cold in many observations, as the 25th percentile recorded was 0.1 °C. There is a right skewness in the temperature, indicating that there were more lower temperatures within the dataset.

Dew Point Temperature:

There is a variation of dew point temperature from -28.5°C to 24.4°C, with the average being 2.55°C. From the large value of the correlation coefficient, a strong relation exists between the temperature and dew point temperature, as one would expect in atmospheric sciences.

	Date/Time	Temp_C	Dew Point Temp_C	Rel Hum_%	Wind Speed_km/h	Visibility_km	Press_kPa	Weather
0	1/1/2012 0:00	-1.8	-3.9	86	4	8.0	101.24	Fog
1	1/1/2012 1:00	-1.8	-3.7	87	4	8.0	101.24	Fog
2	1/1/2012 2:00	-1.8	-3.4	89	7	4.0	101.26	Freezing Drizzle,Fog
3	1/1/2012 3:00	-1.5	-3.2	88	6	4.0	101.27	Freezing Drizzle,Fog
4	1/1/2012 4:00	-1.5	-3.3	88	7	4.8	101.23	Fog

Figure 1: Weather Dataset

Relative Humidity:

Relative humidity varies between 18% and 100%, with an average value of 67.43%. The great dispersion justifies the coexistence of dry and wet conditions at different times in the different parts of a dataset.

Wind Velocity:

Wind speed ranges from 0 km/h to 83 km/h, with an average of 14.94 km/h. Most of the observations lie in the range of ordinary wind speed values. Large extreme outliers show high speeds.

Visibility:

Visibility in the dataset ranges from 0.2 km to 48.3 km, with a mean of 27.66 km. Visibility below 5 km is considered poor, which was reflected in the dataset, and this is usually associated with weather conditions like fog.

Atmospheric Pressure:

Atmospheric pressure ranges from 97.52 kPa to 103.65 kPa, with an average of 101.05 kPa. A standard deviation of 0.84 kPa indicates that most observations lie within a narrow band around the average.

Weather:

This happens to be various kinds of weather data: foggy, rainy, snowy, or clear. For this purpose, an omitted standardized weather condition variable was created, Std_Weather. It is well represented for cloudy and clear conditions. However, most of the data corresponds to those two conditions. But extremely poor weather conditions such as snow and freezing drizzle consist of a significantly lesser number of observations.

4.3 Correlation Analysis

A correlation matrix was calculated to investigate the relationships between the different numerical variables present in the dataset.

Temperature versus dew point temperature: The correlation is very strong and positive, with 0.93, which of course makes sense with meteorological theory since the dew point is highly dependent on air temperature.

Visibility is negatively related to relative humidity with -0.63, meaning that for higher relative humidity, one would normally expect poorer visibility, probably due to fog or precipitation.

It can be noticed that wind speed provides weak correlations with other variables; thus, wind speed is relatively uncorrelated to temperature and humidity variation.

There are different correlations between the atmospheric pressure and temperature, which vary from a negative correlation of -0.24 with temperature to a negative one of -0.32 with dew point temperature, showing that colder and drier conditions are usually experienced with an increase in atmospheric pressure.

	Temp_C	Dew Point Temp_C	Rel Hum_%	Wind Speed_km/h	Visibility_km	Press_kPa
Temp_C	1.000000	0.940727	-0.242884	-0.127727	0.396194	-0.089051
Dew Point Temp_C	0.940727	1.000000	0.093941	-0.123700	0.180395	-0.200540
Rel Hum_%	-0.242884	0.093941	1.000000	0.025299	-0.669438	-0.332335
Wind Speed_km/h	-0.127727	-0.123700	0.025299	1.000000	-0.122254	-0.406027
Visibility_km	0.396194	0.180395	-0.669438	-0.122254	1.000000	0.322612
Press_kPa	-0.089051	-0.200540	-0.332335	-0.406027	0.322612	1.000000

Figure 2: Correlation Matrix

4.4 Exploratory Data Analysis (EDA)

EDA was conducted using visualizations such as histograms and box plots to explore the distribution and variability of the key weather variables.

4.4.1 Temperature Distribution

A histogram of temperature data shows a skewed distribution, with more frequent lower temperatures and a peak around **10°C**. The presence of cold temperatures below **0°C** is likely due to the dataset's geographical and seasonal coverage.

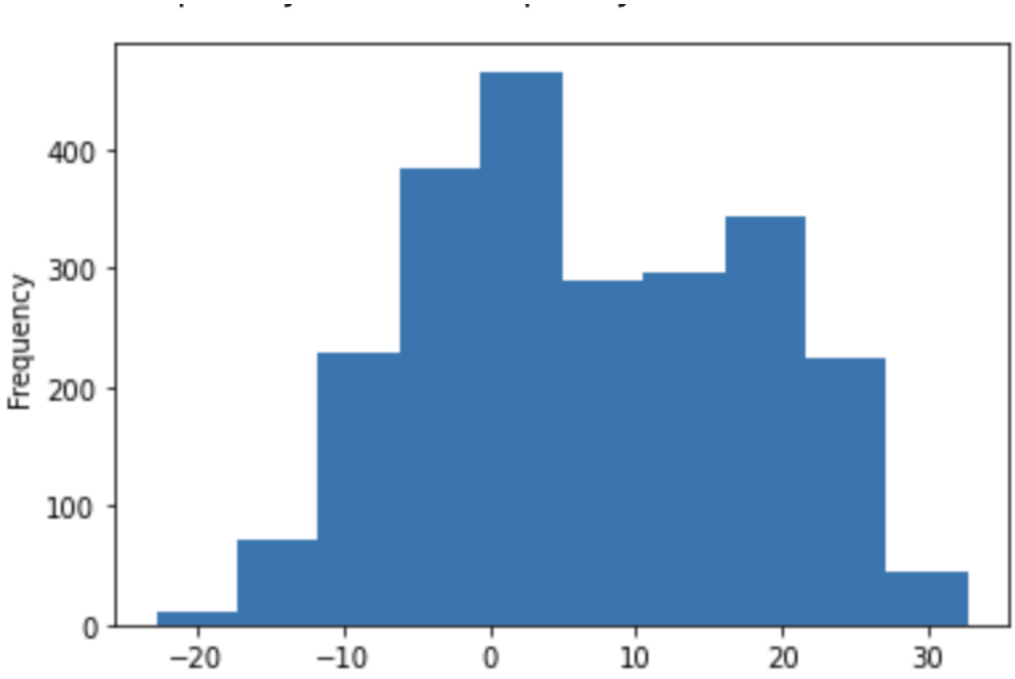

Figure 3: Temperature distribution

4.4.2 Dew Point Temperature Distribution

Similar to temperature, the dew point temperature shows a skewed distribution with a peak around **0°C**. Extremely low dew points indicate very dry conditions.

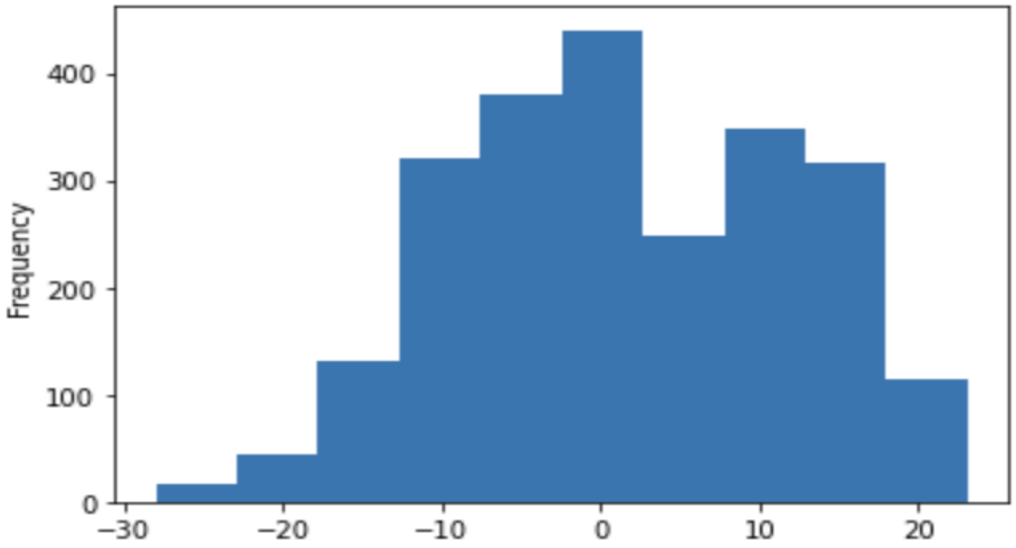

Figure 4: Dew Point Temperature Distribution

4.4.3 Relative Humidity Distribution

The histogram for relative humidity shows a relatively normal distribution, with a slight skew towards higher humidity. The majority of the observations fall between **50%** and **80%**, but the presence of extreme values (near **100%**) indicates very humid conditions, often leading to fog or precipitation.

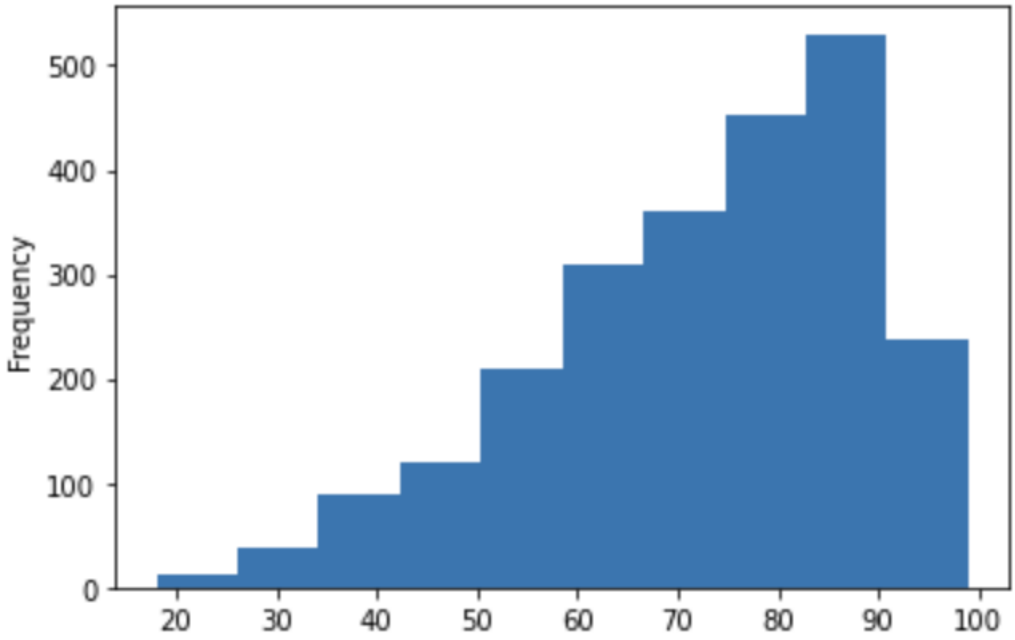

Figure 5: Relative Humidity Distribution

4.4.4 Wind Speed Distribution

The wind speed histogram reveals a right-skewed distribution, with most observations showing wind speeds below **20 km/h**. However, a few outliers exceed **50 km/h**, indicating rare cases of strong winds.

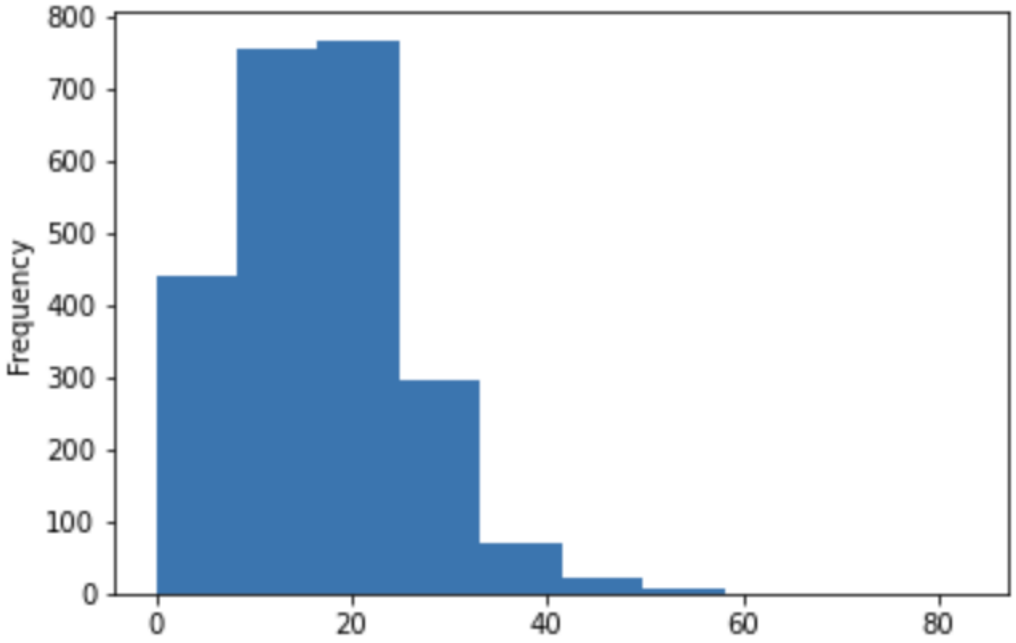

Figure 6: Wind Speed Distribution

4.4.5 Visibility Distribution

The distribution of visibility shows a bimodal pattern. While most observations record visibility of **25 km**, there is a second peak below **10 km**, which can be attributed to weather conditions like fog or heavy precipitation reducing visibility.

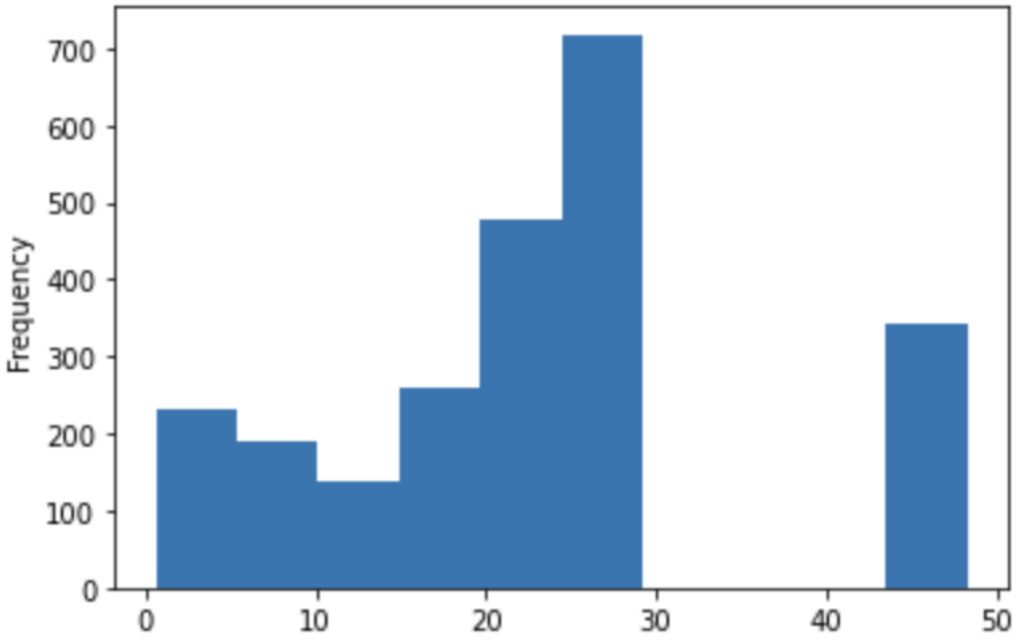

Figure 7: Visibility Distribution

4.4.6 Atmospheric Pressure Distribution

Atmospheric pressure shows a nearly normal distribution, with most values clustered around the mean of **101 kPa**. This is consistent with normal atmospheric conditions, though deviations to higher pressures are associated with clear and cold weather.

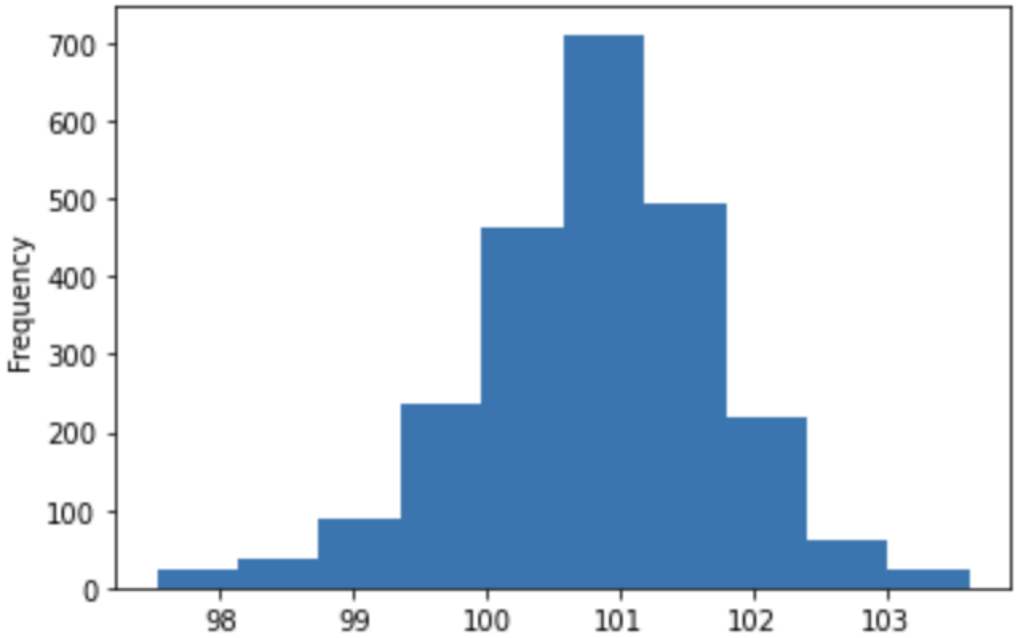

Figure 8: Atmospheric pressure Distribution

4.4.7 Box Plot Analysis

Box plots for variables such as **wind speed** and **visibility** reveal the presence of outliers. Extremely high wind speeds and poor visibility are uncommon, but their occurrence indicates severe weather conditions. Box plots help in identifying these extreme values, which can be critical for understanding severe weather patterns.

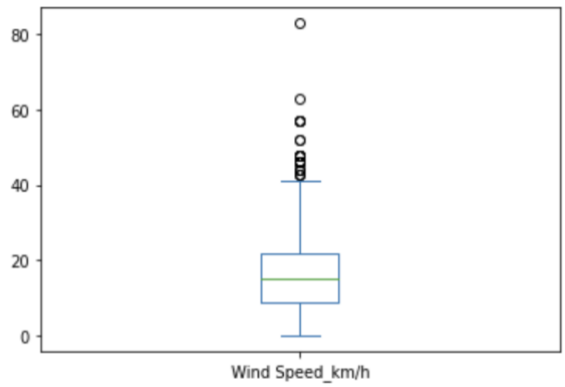

Figure 9: Box Plot of Wind Speed.

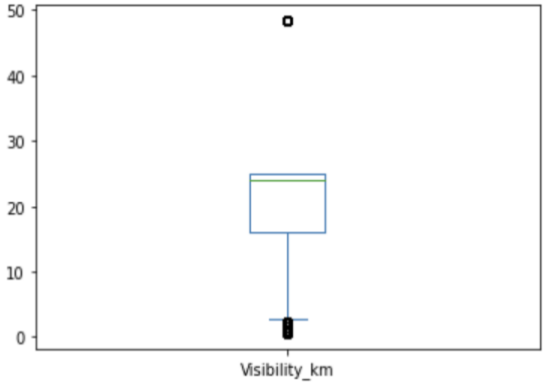

Figure 10: Box plot of Visibility

4.5 Data Preprocessing

Initial preprocessing included encoding in LabelEncoder, scaling features, and splitting into a training set and a test set. This was all preparation for the adaptation of machine learning models toward the creation of predictions with respect to weather conditions.

4.5.1 Label encoding

In this case, a categorical variable, Std_Weather, in the weather dataset described standardized weather conditions as "Clear", "Cloudy", "Rain", or "Snow". Thus, this variable needed to be converted into numerical form since most machine learning methods require numerical data. This had been accomplished during the step of label encoding.

Label Encoding Process:

Label encoding assigns a unique integer to each category in the Std_Weather column.

For example, weather conditions could be represented as:

"Clear" = 0

"Cloudy" = 1

"Rain" = 2

"SNOW" = 3

This conversion is necessary because machine learning algorithms read from numerical values in order to make decisions from the underlying patterns of the data.

Impact of Label Encoding:

Label encoding maintains the target variable as categorical but, at the same time is understandable by the algorithms of machine learning. However, once integer values are assigned to this encoding technique, having any regard to natural order between categories-for instance, "Clear" is neither less nor greater than "Cloudy, "-it is suitable in classification problems when there is no ordinal relationship among labels.

4.5.2 Feature Scaling

The feature temperature in Celsius will range from -23.3°C to 33°C, the wind speed in km/h from 0 to 83 km/h, and the visibility in kilometers even from 0.2 to 48.3 km. Therefore, feature scaling should be performed for the features in the weather dataset so that machine learning models learn from them effectively.

Feature Scaling Process:

The standardization or normalization of features is one of the common steps in preprocessing most data for machine learning; algorithms sensitive to the scale of their input features include Support Vector Classifier SVC, K-Nearest Neighbors KNN, and Logistic Regression.

• Feature scaling aims to rescale the features' values to have a mean of zero and a standard deviation of one. This is usually done by utilizing standardization, which involves subtracting the mean from each feature and dividing it by the standard deviation:

$$Xscaled = \frac{X - \mu}{\sigma}$$

where:

- X is the original feature value,
- μ is the mean of the feature,
- σ is the standard deviation.

```
array([[-0.60610569, -0.26580087,  1.2889447 ,  0.34813752, -0.80672093,
         1.48171091],
       [ 0.23189884, -0.0089251 , -0.89182823, -0.39565791,  0.14368713,
         0.68229088],
       [ 1.65209598,  1.57038525, -0.51798144,  0.24188103,  1.97381173,
         0.31498979],
       ...,
       [-0.62374789, -0.28482871,  1.2889447 ,  1.19818944, -1.44294451,
        -0.92735214],
       [-0.59728459, -0.31337046,  1.03971351,  1.19818944, -1.05806852,
        -0.97056404],
       [-0.57964239, -0.34191222,  0.85279011,  1.41070243, -0.93239473,
        -0.99216998]])
```

Figure 11: Feature Scaling

Impact of Feature Scaling:

Improved Model Performance: Feature scaling ensures that all features are on the same scale, preventing features with larger ranges (e.g., wind speed or visibility) from disproportionately influencing the model's predictions.

Algorithm Efficiency: Models like SVC and KNN are distance-based algorithms, meaning that the performance is highly dependent on the scale of the input data. By scaling the features, the models can make more accurate predictions.

Consistent Gradient Descent: For gradient-based models such as Logistic Regression, feature scaling ensures that the optimization process converges faster by providing consistent updates to model parameters.

4.5.3 Training-Testing Split

For different model performance evaluations on data that it has not seen, the dataset was divided into two parts: the training set and the testing set. This is generally done in machine learning to avoid overfitting-a situation where a model performs well in training but generalizes poorly on new data.

In Train-Test Split Process the dataset was split using an 80/20 ratio, where 80% of this data was used in training the models and 20% of the data was reserved for testing the models. This split was achieved using the train_test_split function from the scikit-learn library, which randomly divides the dataset into the training and testing sets. A random seed (random_state = 42) was set to ensure

reproducibility, meaning the same split is generated each time the code is run. Stratification comes in handy to ensure that the target variable Std_Weather's distribution is kept constant between the training and testing sets. That will avoid any imbalance between weather conditions in the train-test split.

Impact of Train-Test Split

Generalization: When we hold out 20% of the data for testing then it can effectively evaluate the model's generalization ability on unseen data. This ensures that the model does not memorize data it has seen in training. But it can learn from patterns the data presents to apply against new data.
Avoiding Overfitting: Because of the train-test split, the models are not evaluated with respect to what they have seen in training; it will thus provide an unbiased estimate of the performance of the models on real data.

Model Tuning: After the training of models on the training set, their performance should have been observed on the test set, which could enable tuning based on the results.

4.5.4 Impact of EDA and Label encoding

This exploratory data analysis gave immense insight into the weather dataset, such as temperature and dew point being highly related, high humidity corresponding to poor visibility, wind speeds generally moderate though extreme values crop up for rare cases, and atmospheric pressure being normally distributed with minor fluctuations in correspondence with changes in weather conditions.
The dataset provides rich information on various patterns of weather, further usable in forecasting or predictive modeling. Use of visualization helped to understand the distribution of key variables and identification of potential outliers. The standardized weather categories made it easier to analyze a given type of weather condition. The combination of descriptive statistics, the analysis of correlations, and visualization techniques proved to be useful in interpreting this dataset.
Label encoding, feature scaling, and splitting into a train-test set are some of the most important preprocessing steps in any machine learning pipeline. Label encoding changed categorical weather conditions into numerical labels that can be understood by models as the target variable. Feature scaling normalized the ranges of input features such that algorithms could process the data efficiently. This involved, finally, splitting the dataset into a training set and a test set in appropriate format-a good platform on which to perform model evaluation and ensure that the models generalize nicely to unseen data.These preprocessing techniques were the entrance to training different machine learning models, discussed in the next section, along with their confusion matrices and performance evaluations.

4.6 Model Development and Evaluation

In this section, various classification models were trained to predict the standardized weather conditions (Std_Weather). The dataset was split into training (80%) and testing (20%) sets, and multiple models were trained using the training data. The following models were evaluated:

- Decision Tree Classifier
- Random Forest Classifier
- Support Vector Classifier (SVC)
- K-Nearest Neighbors (KNN) Classifier
- Logistic Regression
- XGBoost

Each model's performance was evaluated using a confusion matrix, classification report and receiver operating characteristic (ROC) curve. Confusion matrix helps to visualize the true positive, true negative, false positive, and false negative predictions. Classification report gives fair understanding about the accuracy, precision, recall and F1 score. ROC is used to evaluate the performance of binary model by plotting the trade-off between True positive rate(TPR) and False positive rate(FPR)

4.6.1 Decision Tree Classifier

The Decision Tree classifier was the first model applied to the dataset. Decision Trees are non-parametric supervised learning methods used for classification and regression.

Confusion Matrix for Decision Tree Classifier

The confusion matrix for the Decision Tree classifier is as follows:

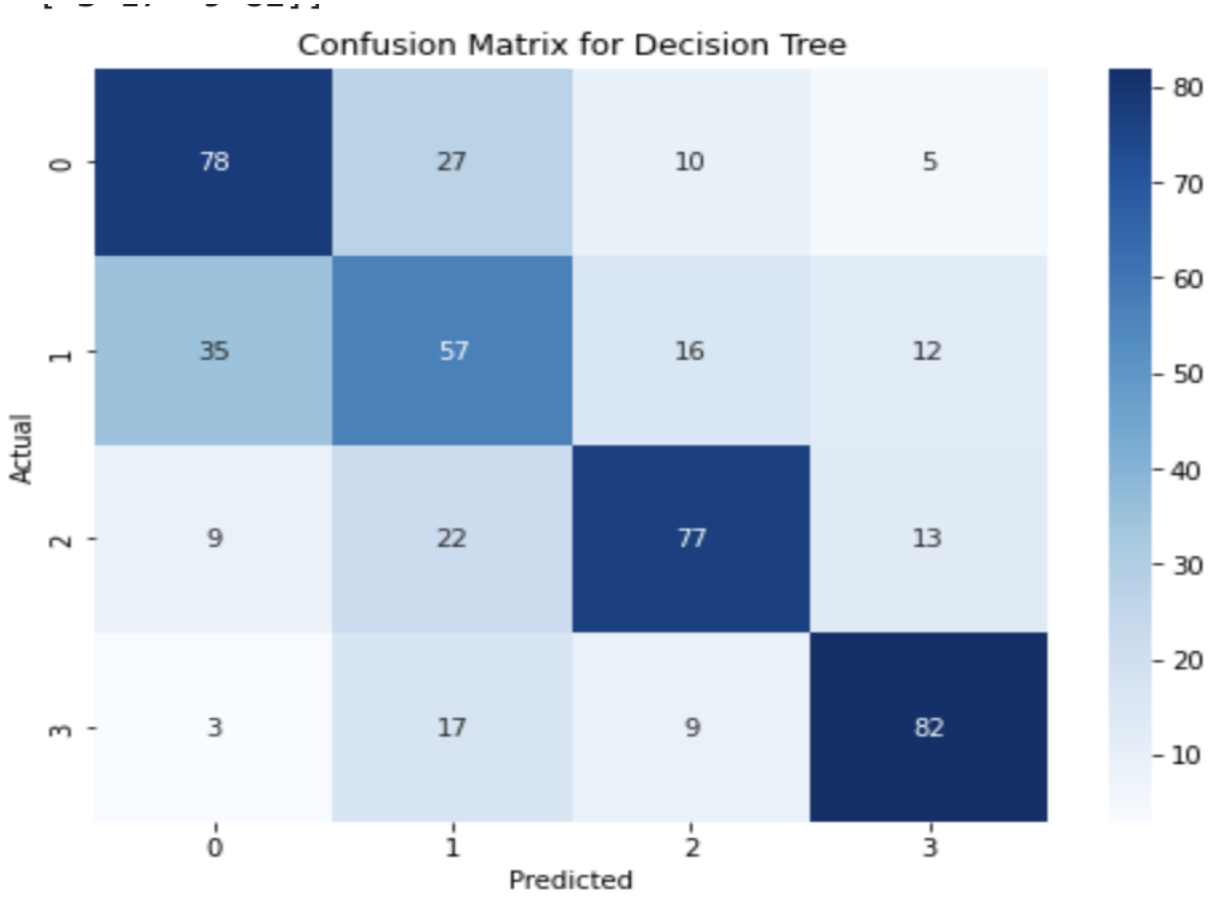

Figure 12: Confusion Matrix of Decision Tree Classifier

The matrix compares **predicted labels** (on the x-axis) against **actual labels** (on the y-axis), and it consists of four classes (0, 1, 2, 3). Here's the breakdown:

Class 0: The model correctly predicted **78** instances of Class 0 (true positives). The model misclassified **27** instances of Class 0 as Class 1, **10** instances as Class 2, and **5** instances as Class 3.

Class 1:The model correctly predicted **57** instances of Class 1 (true positives). 35 instances of Class 1 were incorrectly classified as Class 0, **16** as Class 2, and **12** as Class 3.

Class 2: **77** instances of Class 2 were correctly predicted. **9** instances of Class 2 were misclassified as Class 0, **22** as Class 1, and **13** as Class 3.

Class 3: The model correctly predicted **82** instances of Class 3. **3** instances of Class 3 were misclassified as Class 0, **17** as Class 1, and **9** as Class 2.

Classification Report:

Classification Report for Decision Tree:

	precision	recall	f1-score	support
0	0.62	0.65	0.64	120
1	0.46	0.47	0.47	120
2	0.69	0.64	0.66	121
3	0.73	0.74	0.74	111
accuracy			0.62	472
macro avg	0.63	0.63	0.63	472
weighted avg	0.62	0.62	0.62	472

Figure 13: Classification Report for Decision Tree

Overall Model Performance:

Accuracy: The model achieved an overall accuracy of 62%, which indicates that approximately 62% of the total instances were correctly classified across all classes.

Macro Average:

Precision: This is the average precision across all classes, treating each class equally. The model is moderately precise overall.

Recall:The macro average recall shows that the model is able to identify approximately 63% of the actual instances across all classes.

F1-Score: The macro F1-score reflects a balance between precision and recall, indicating that the model's performance is consistent across different classes but could be improved for certain classes like Class 1.

Weighted Average:

Precision, Recall, and F1-Score: The weighted average accounts for the number of instances in each class (support) and provides an overall measure of the model's performance. The weighted average F1-score of 0.62 suggests that the model's performance is consistent across classes when adjusted for the imbalance in class frequencies.

ROC Curve:

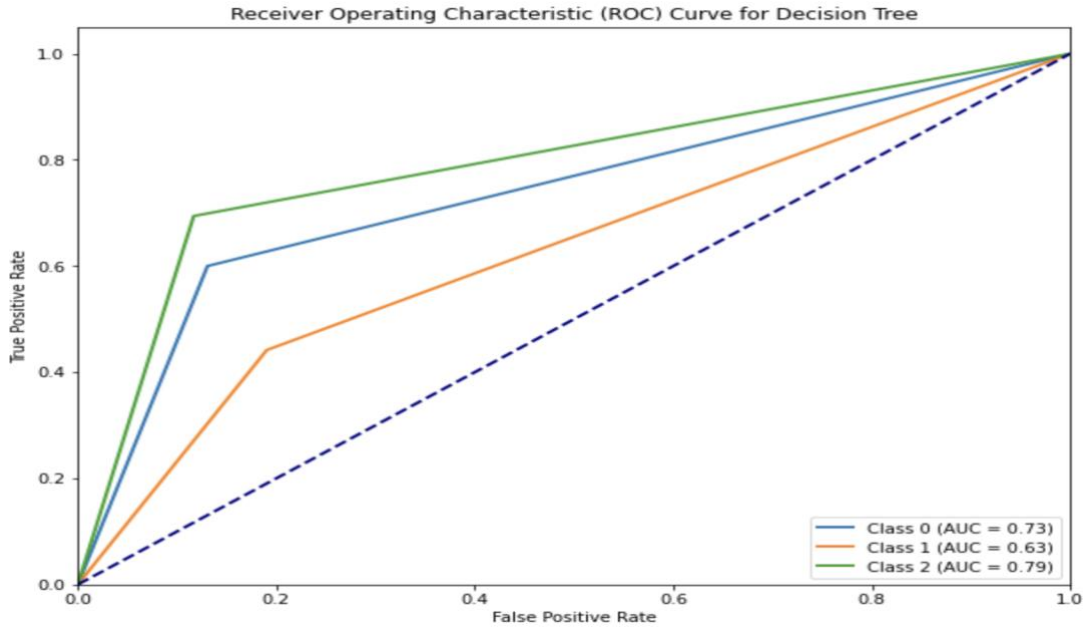

Figure 14: ROC for Decision Tree classifier

Key Findings:

Class 2 is the best-classified category by the Decision Tree model, with an AUC of **0.79**, suggesting relatively high discriminative power for this class.

Class 0 has a moderate performance with an AUC of **0.73**, indicating that the model is reasonably effective in distinguishing Class 0 but could be further improved.

Class 1 has the weakest performance, with an AUC of **0.63**, showing significant difficulties in distinguishing this class from the others. This is consistent with the previous findings from the classification report and confusion matrix, where the model was shown to struggle with Class 1.

4.6.2 Random Forest Classifier

Random Forest is an ensemble learning method that constructs multiple decision trees during training and outputs the mode of the classes for classification tasks.

Confusion Matrix for Random Forest Classifier

The provided image shows the Confusion Matrix for a Random Forest classifier, which is a more advanced and robust model than a Decision Tree. This matrix visually represents how well the model classified each class by comparing the actual labels (on the y-axis) with the predicted labels (on the x-axis).

The confusion matrix for the Random Forest classifier is as follows:

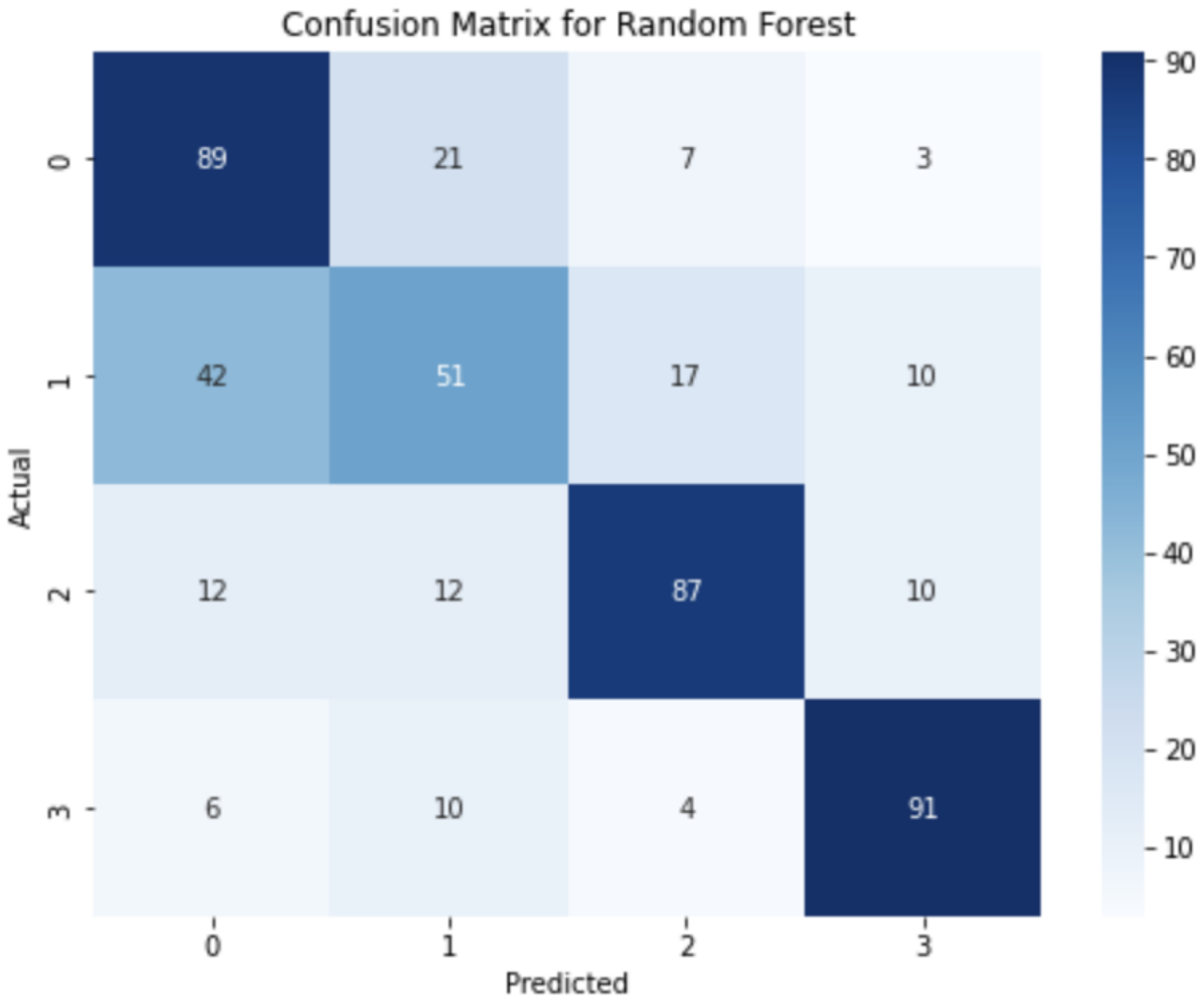

Figure 15: Confusion Matrix of Random Forest Classifier

Key Findings

The Random Forest model performs well overall, with most classes being correctly classified, particularly Class 3 (Snow) and Class 2 (Rain).

Class 1 (Cloudy) has the most misclassifications, often being confused with Class 0 (Clear) and Class 2 (Rain). This suggests that the model finds it difficult to differentiate between cloudy weather and other weather types.

The model shows a relatively strong ability to identify Class 0 (Clear) and Class 2 (Rain), though some confusion still exists with other classes.

The Random Forest model is effective at identifying Snowy and Rainy conditions (Class 3 and Class 2), but it struggles somewhat with Cloudy and Clear conditions (Class 1 and Class 0).

Class 1 (Cloudy) has the most confusion, particularly with Class 0 and Class 2, and this could be an area for improvement in the model.

Overall, the Random Forest model demonstrates good classification performance, and it performs better than a simpler model like a Decision Tree, especially for distinguishing more difficult classes like Class 3 and Class 2.

Classification Report:

Classification Report for Random Forest:

	precision	recall	f1-score	support
0	0.60	0.74	0.66	120
1	0.54	0.42	0.48	120
2	0.76	0.72	0.74	121
3	0.80	0.82	0.81	111
accuracy			0.67	472
macro avg	0.67	0.68	0.67	472
weighted avg	0.67	0.67	0.67	472

Figure 16: Classification Report for Random Forest

Overall Model Performance:

Accuracy. The Random Forest model achieved an overall accuracy rate of 67%, implying that 67 of every total predictions made were accurate. Such performance is quite justified, but can be enhanced further.

Macro Average:

Precision: Precisely, the mean macro average precision across the four classes equally stands at 67 percent average meaning that, on average the model still manages to avoid creating too many false positive cases.

Recall: The Mean macro average recall is 68 percent indicating the extent to which the model can correctly classify most of the instances of all classes in which they fall.

F1-Score: The mean macro F1 score represents the mean of the F1 scores calculated for each of the classes, reinforcing any important areas of concern and providing satisfactory overall performance prospects for all classes.

Weighted Average:

Precision: Recall and F1 Score. The weighted averages are based on the support metric for each class and therefore enable more vigorous analysis of the model. These scores indicate that the model performs consisting across the various classes with regards to precision, recall and F1-score being reasonable overall performance across all classes.

ROC Curve:

Key Findings:

The Random Forest model performs well overall, particularly for Class 2 (Rainy Weather) with an AUC of 0.93, indicating excellent classification ability.

The model also performs strongly for Class 0 (Clear Weather) with an AUC of 0.88, demonstrating good classification for this class.

The model's weakest performance is for Class 1 (Cloudy Weather) with an AUC of 0.76, indicating that the model struggles to differentiate this class from the others. This suggests that further improvements or feature engineering could help improve the classification of Class 1.

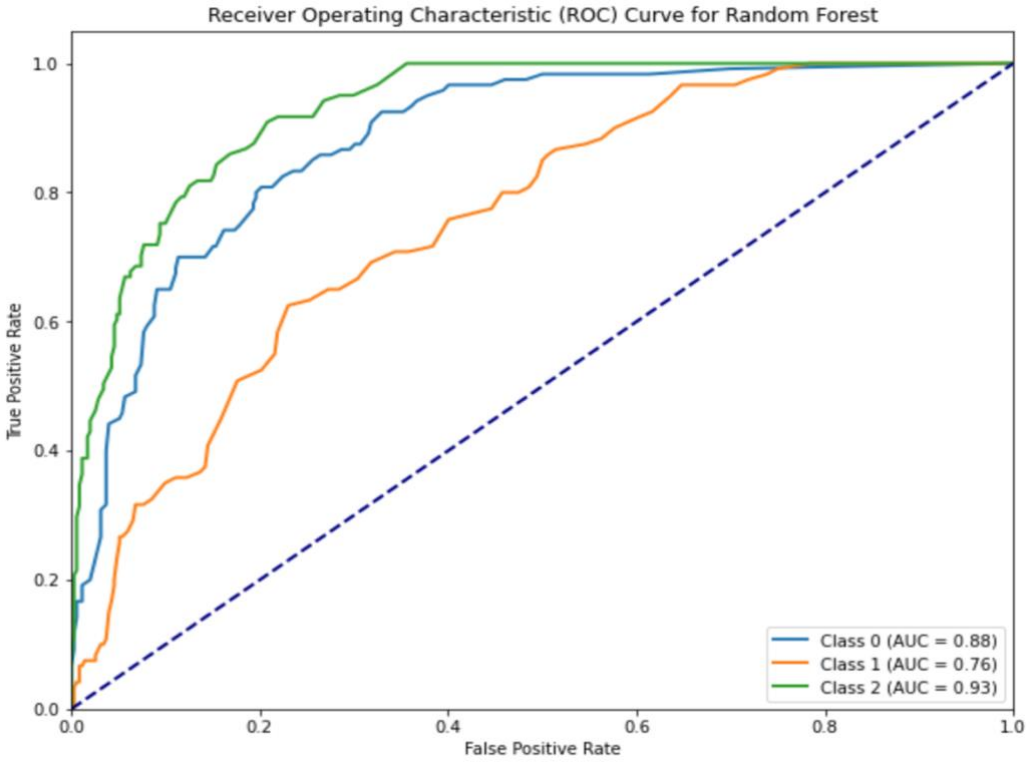

Figure 17: ROC for Random Forest Classifier

4.6.3 Support Vector Classifier

Support Vector Classifier is a supervised learning model that works by finding the hyperplane that best separates the classes.

Confusion Matrix for SVC

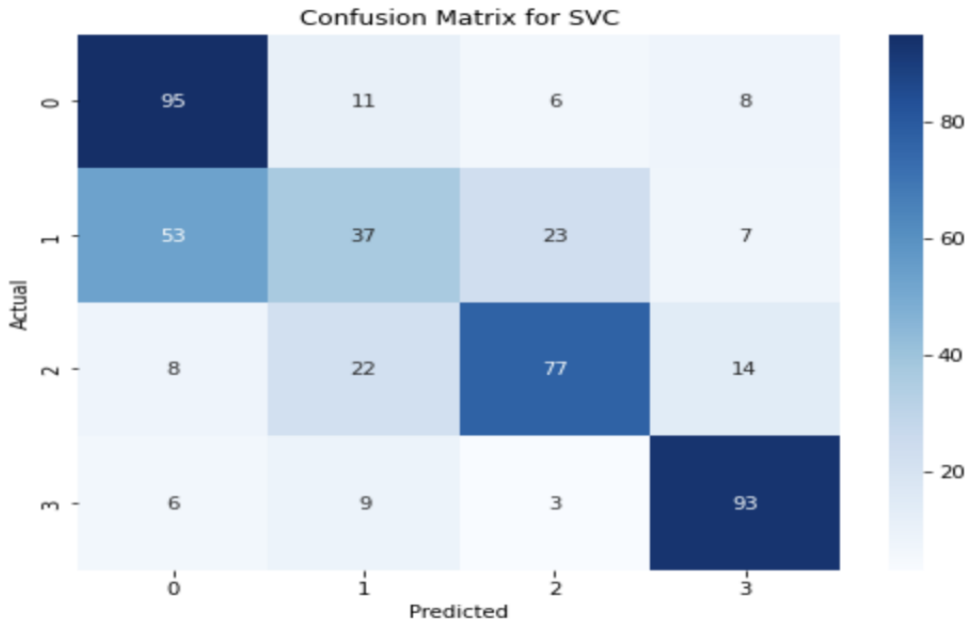

Figure 18: Confusion Matrix for Support Vector Classifier (SVC)

Key Findings:

The SVC model exhibits the highest performances for Class 0 (Clear) and Class 3 (Snowy) due to the majority of instances being correctly predicted and labelled, and not many being misclassified.

The greatest problem with the model is seen with Class 1 (Cloudy) where too many instances are incorrectly assigned to Class 0 and Class 2, as the SVC model is unable to tell the difference between cloudy and clear or rainy weather.

Class 2 (Rainy) puts in a fair show in performance, though there is a tendency to confuse it with Class 1 (Cloudy), implying that the classes should not be separate as they are interrelated in a way.

The SVC model performs better during Class 0 (Clear) and Class 3 (Snowy) weather, however struggles during Class 2 Rain and Class 1 Cloudy weather.

The low performance rate on Class 1 indicates that the model needs improvements in detecting the alteration of cloudy weather with respect to other types of weather. These could be helped by improving hyperparameters or learning about new features to effectively separate these classes.

Classification Report:

Classification Report for SVC:

	precision	recall	f1-score	support
0	0.59	0.79	0.67	120
1	0.47	0.31	0.37	120
2	0.71	0.64	0.67	121
3	0.76	0.84	0.80	111
accuracy			0.64	472
macro avg	0.63	0.64	0.63	472
weighted avg	0.63	0.64	0.63	472

Figure 19: Classification Report for SVC

Key Findings:

The SVC model demonstrates moderate overall performance, with 64% accuracy and balanced precision and recall across most classes. The model performs best for Class 3 (Snowy Weather) but struggles with Class 1 (Cloudy Weather), which significantly reduces its overall effectiveness.

Improvement Areas: The low performance for Class 1 (Cloudy Weather) indicates that feature engineering or hyperparameter tuning may be necessary to improve the model's ability to distinguish cloudy weather from other classes. Additionally, the use of an ensemble method or other advanced models could help improve overall accuracy and class performance.

ROC Curve

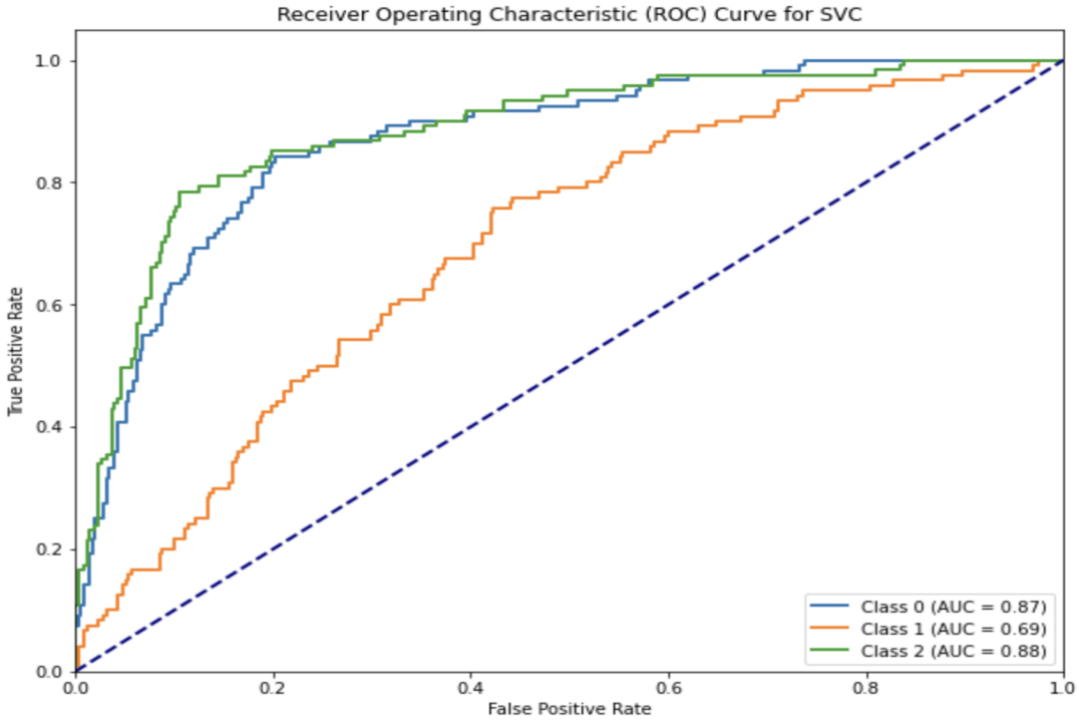

Figure 20: ROC for SVC

Key Findings

The SVC model shows strong performance for Class 0 (Clear Weather) and Class 2 (Rainy Weather**)**, with AUC values of 0.87 and 0.88, respectively. This means the model can effectively distinguish these two classes with high accuracy. However, the model performs less effectively for Class 1 (Cloudy Weather), with an AUC of 0.69. The lower AUC and more gradual rise of the ROC curve for Class 1 indicate that the model struggles to distinguish cloudy weather from other weather types, leading to more misclassifications. Overall, the SVC model performs well for two of the three classes but may require further improvement for Class 1. Enhancing the model's ability to distinguish cloudy weather could be achieved through additional feature engineering or hyperparameter tuning.

4.6.4 K-Nearest Neighbors (KNN) Classifier

Confusion Matrix of KNN:

The KNN model performs best for Class 0 (Clear Weather) and Class 3 (Snowy Weather), with 82 correct classifications for both. There are some misclassifications, especially between Class 0 and Class 1, which indicates that the model struggles to distinguish between clear and cloudy weather conditions.

The model struggles the most with Class 1 (Cloudy Weather), where it frequently confuses cloudy weather with clear (Class 0) and rainy weather (Class 2). This suggests that Class 1 has overlapping features with other classes, making it more difficult for the KNN model to classify.

Class 2 (Rainy Weather) is moderately well-classified, with 71 correct predictions but also significant confusion with both Class 1 (Cloudy) and Class 0 (Clear) weather.

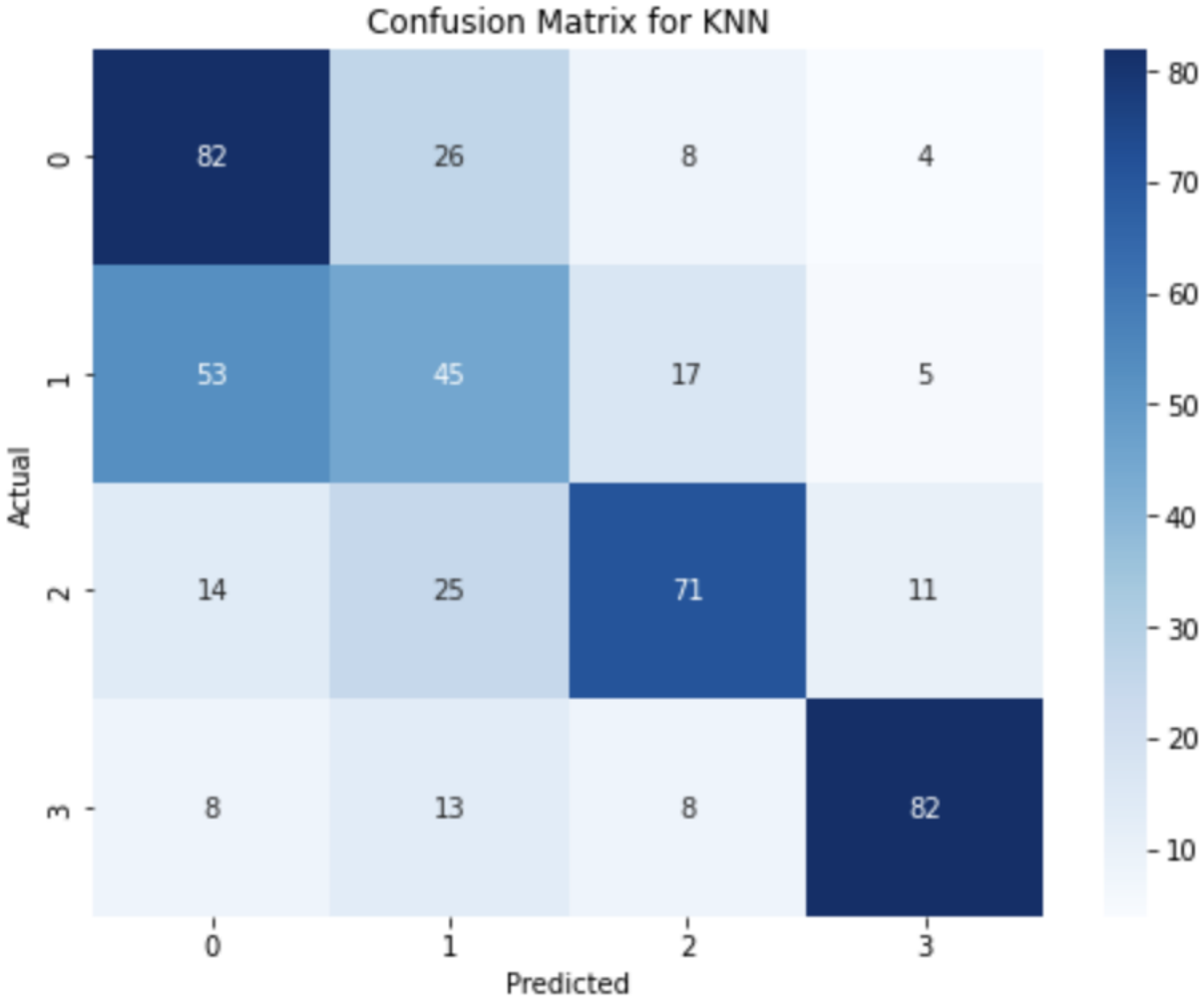

Figure 21: Confusion Matrix for KNN

Classification Report:

Classification Report for KNN:

	precision	recall	f1-score	support
0	0.52	0.68	0.59	120
1	0.41	0.38	0.39	120
2	0.68	0.59	0.63	121
3	0.80	0.74	0.77	111
accuracy			0.59	472
macro avg	0.61	0.60	0.60	472
weighted avg	0.60	0.59	0.59	472

Figure 22: Classification Report for KNN

Key Findings:

The KNN model demonstrates moderate performance overall, with 59% accuracy and balanced precision and recall across most classes.

The model performs best for Class 3 (Snowy Weather), but it struggles significantly with Class 1 (Cloudy Weather), which reduces its overall performance.

Improvement Areas: The low performance for Class 1 (Cloudy Weather) suggests that further feature engineering, hyperparameter tuning (e.g., changing the number of neighbors), or using a different classification algorithm might help improve the model's performance.

ROC:

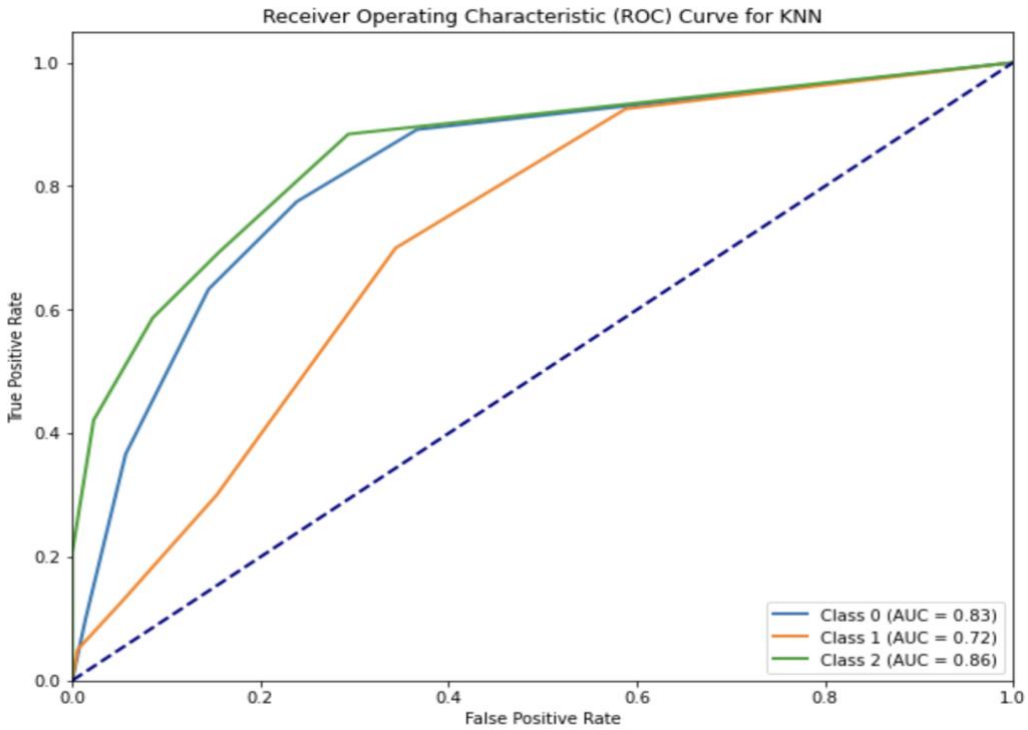

Figure 23: ROC for KNN

Key Findings:

The KNN model shows strong performance for Class 0 (Clear Weather) and Class 2 (Rainy Weather), with AUC values of 0.83 and 0.86, respectively. This means the model can effectively distinguish these two classes with high accuracy.

However, the model performs less effectively for Class 1 (Cloudy Weather), with an AUC of 0.72. The lower AUC and more gradual rise of the ROC curve for Class 1 indicate that the model struggles to distinguish cloudy weather from other weather types, leading to more misclassifications.

Overall, the KNN model performs well for two of the three classes but may require further improvement for Class 1. Enhancing the model’s ability to distinguish cloudy weather could be achieved through additional feature engineering or hyperparameter tuning.

4.6.5 Logistic Regression

Confusion Matrix:

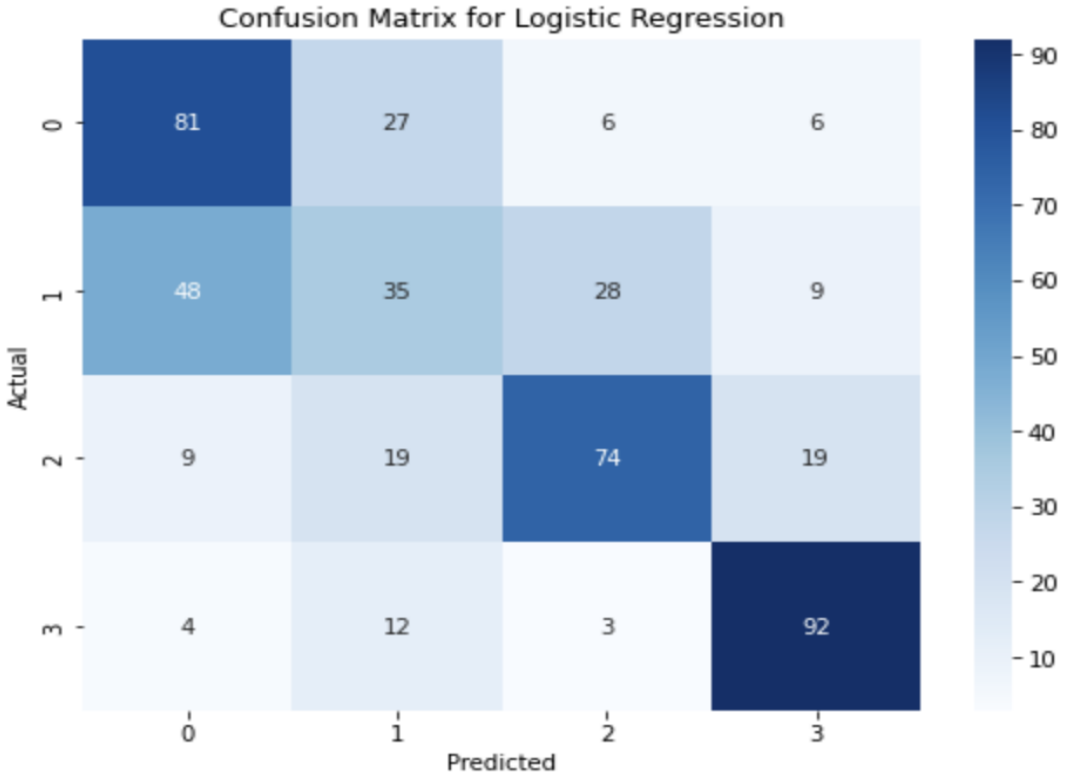

Figure 24: Confusion Matrix for Logistic Regression

Key Findings:

The Logistic Regression model performs reasonably well for Class 0 (Clear Weather) and Class 3 (Snowy Weather), but struggles with Class 1 (Cloudy Weather), where many instances are misclassified as either clear or rainy weather.

Class 2 (Rainy Weather) shows moderate performance, but there is significant confusion with other weather types, particularly Class 1 (Cloudy Weather) and Class 3 (Snowy Weather).

The model may benefit from feature engineering or the use of more complex models (e.g., ensemble methods like Random Forests) to better separate overlapping features, especially between Cloudy and Rainy weather.

Classification Report:

Key Findings:

Accuracy: The overall accuracy of the Logistic Regression model is 60%, meaning that 60% of the total predictions were correct across all classes. This indicates moderate performance but leaves room for improvement.

Macro Average:

Precision: The macro average precision reflects the model's ability to avoid false positives across all classes. A value of 0.59 suggests moderate precision overall.

Recall: The macro average recall indicates the model's ability to identify true positives across all classes. A value of 0.60 suggests that the model can detect most actual instances, though it struggles with certain classes like Class 1.

F1-Score: The macro F1-score reflects a balance between precision and recall across all classes, indicating moderate performance overall.

Weighted Average:

Precision, Recall, and F1-Score: The weighted averages take into account the number of instances in each class (support). These metrics reflect the model's overall performance when accounting for the distribution of the dataset, and they indicate that the model struggles with certain classes, particularly Class 1 (Cloudy Weather).

Classification Report for Logistic Regression:

	precision	recall	f1-score	support
0	0.57	0.68	0.62	120
1	0.38	0.29	0.33	120
2	0.67	0.61	0.64	121
3	0.73	0.83	0.78	111
accuracy			0.60	472
macro avg	0.59	0.60	0.59	472
weighted avg	0.58	0.60	0.59	472

Figure 25: Classification Report for Logistic Regression

ROC:

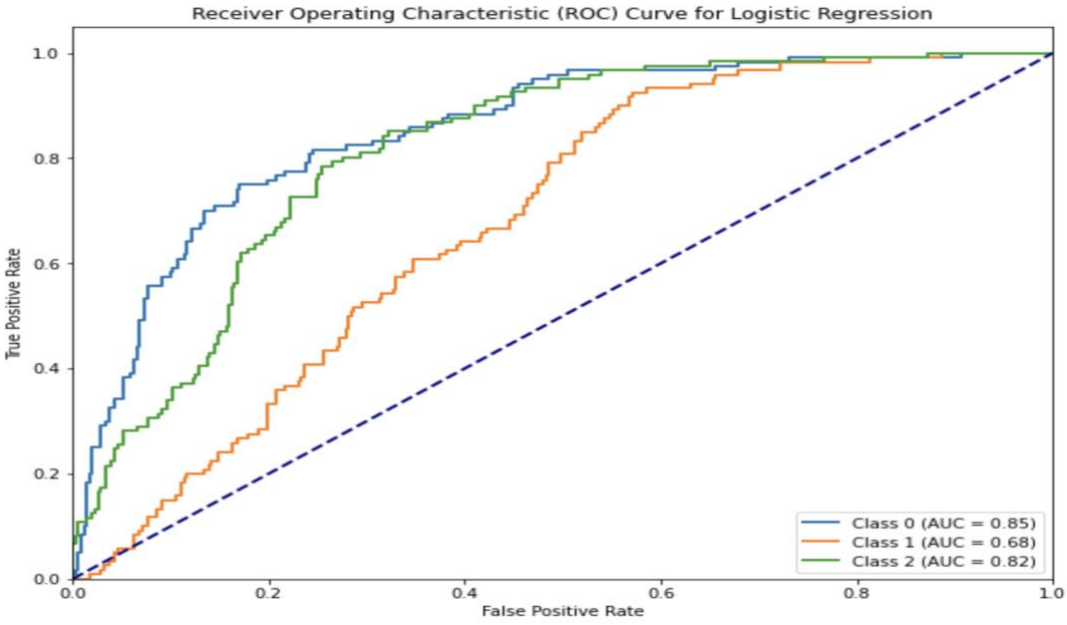

Figure 26: ROC for Logistic Regression

Key Findings:

In terms of the classes, the Logistic Regression model fits very well for Class 0 (Clear Weather) and Class 2 (Rainy Weather) with AUC scores of 0.85 and 0.82 within the given range. As a result, it can be stated that the model is very effective in classifying these two classes where it performs with high precision.

On the contrary, this performance decreases for Class 1 (Cloudy Weather) since the AUC is lower at the value of 0.68. It indicates that the model cannot properly identify cloudy weather as a distinct type of weather and therefore misclassifies more instances.

All in all, the model appears to perform well on two of the three classes at the current state of the model while class one slightly needs improvement. Ways to secure this could be through feature engineering, data augmentation or even more complex models such as XGBoost.

4.6.6 XGBoost

Confusion Matrix

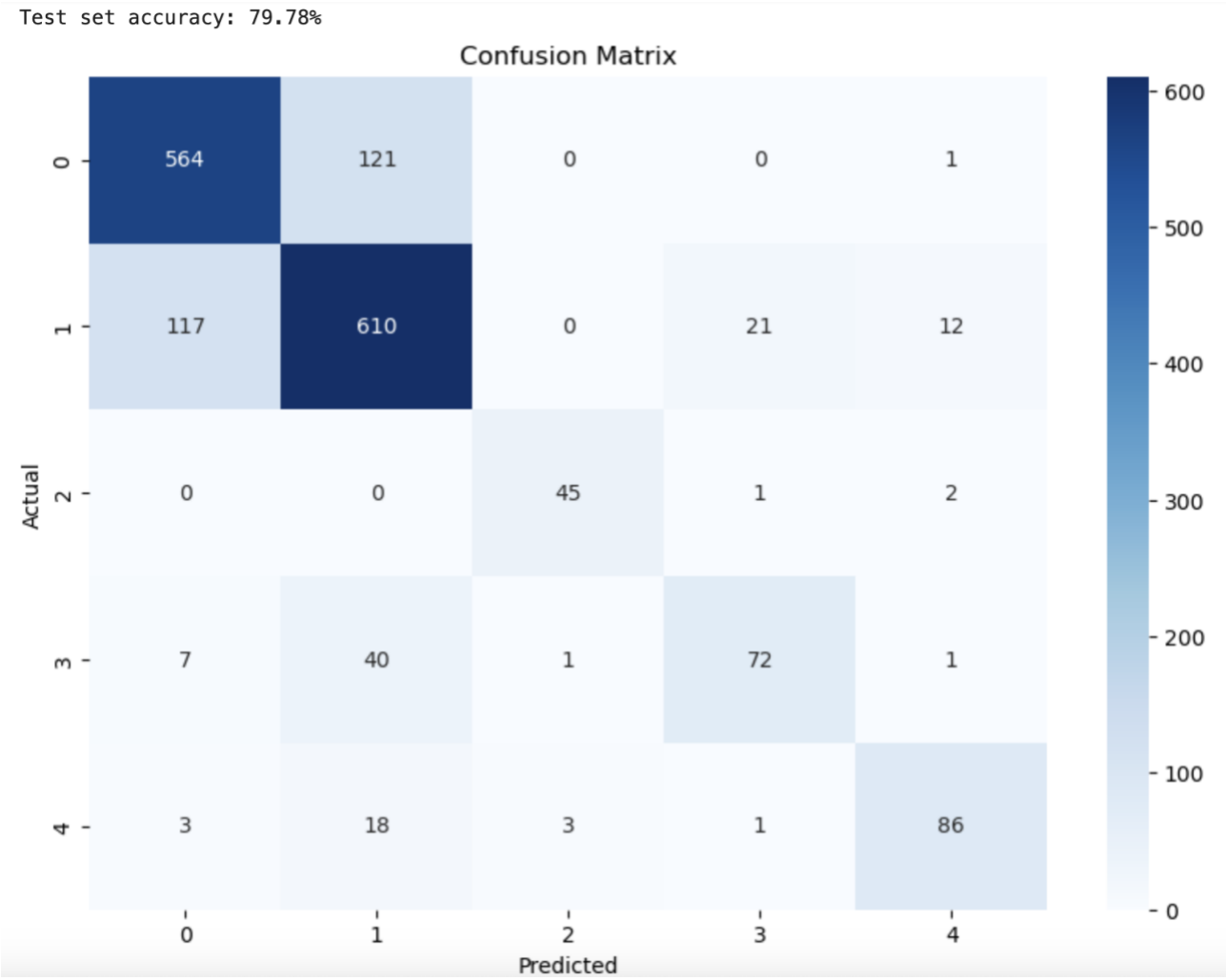

Figure 27: Confusion Matrix for XGBoost

The image shows the Confusion Matrix for an XGBoost model, which has the highest accuracy among other models (as indicated by the caption showing an accuracy of 79.78%). This confusion matrix visually compares the actual labels (on the y-axis) with the predicted labels (on the x-axis) across five different classes (Class 0 to Class 4).

Key Findings:

- The overall accuracy of the XGBoost model is 79.78%, which is relatively high for a multi-class classification problem.
- The model shows strong performance for Class 0, Class 1 and Class 2 where most instances are correctly classified. However Class 3 and Class 4 show some confusion with other classes, particularly Class 1.
- The misclassifications are most prominent between Class 1 and Class 0, as well as between Class 3 and Class 1. This indicates that the model could benefit from further feature engineering or hyperparameter tuning to better separate these classes.
- The XGBoost model performs well overall with nearly 80% accuracy, making it the best-performing model. It classifies Class 0, Class 1 and Class 2 with high accuracy but shows some confusion in Class 3 and Class 4 which could be improved with additional model tuning or feature refinement.
- The model's ability to distinguish between most of the classes is impressive, and it outperforms other models in this task.

Classification Report:

Classification Report:

	precision	recall	f1-score	support
0	0.82	0.82	0.82	686
1	0.77	0.80	0.79	760
2	0.92	0.94	0.93	48
3	0.76	0.60	0.67	121
4	0.84	0.77	0.81	111
accuracy			0.80	1726
macro avg	0.82	0.79	0.80	1726
weighted avg	0.80	0.80	0.80	1726

Figure 28: Classification Report for XGBoost

Key Findings:

Accuracy: The overall accuracy of the XGBoost model is **80%**, which is a solid performance for a multi-class classification problem. This indicates that 80% of the total predictions made by the model were correct.

Macro Average:

Precision: The macro average precision shows that, on average, the model performs well across all classes in terms of avoiding false positives.

Recall: The macro average recall shows that the model performs well across all classes in terms of identifying true positives, though it could improve slightly, especially for Class 3.

F1-Score: The macro average F1-score reflects balanced performance across all classes, suggesting the model handles the multi-class classification task efficiently.

Weighted Average:

Precision, Recall, and F1-Score: The weighted averages, which take class support into account, also indicate strong overall performance. The model maintains high precision, recall, and F1-scores when considering the different class sizes.

ROC:

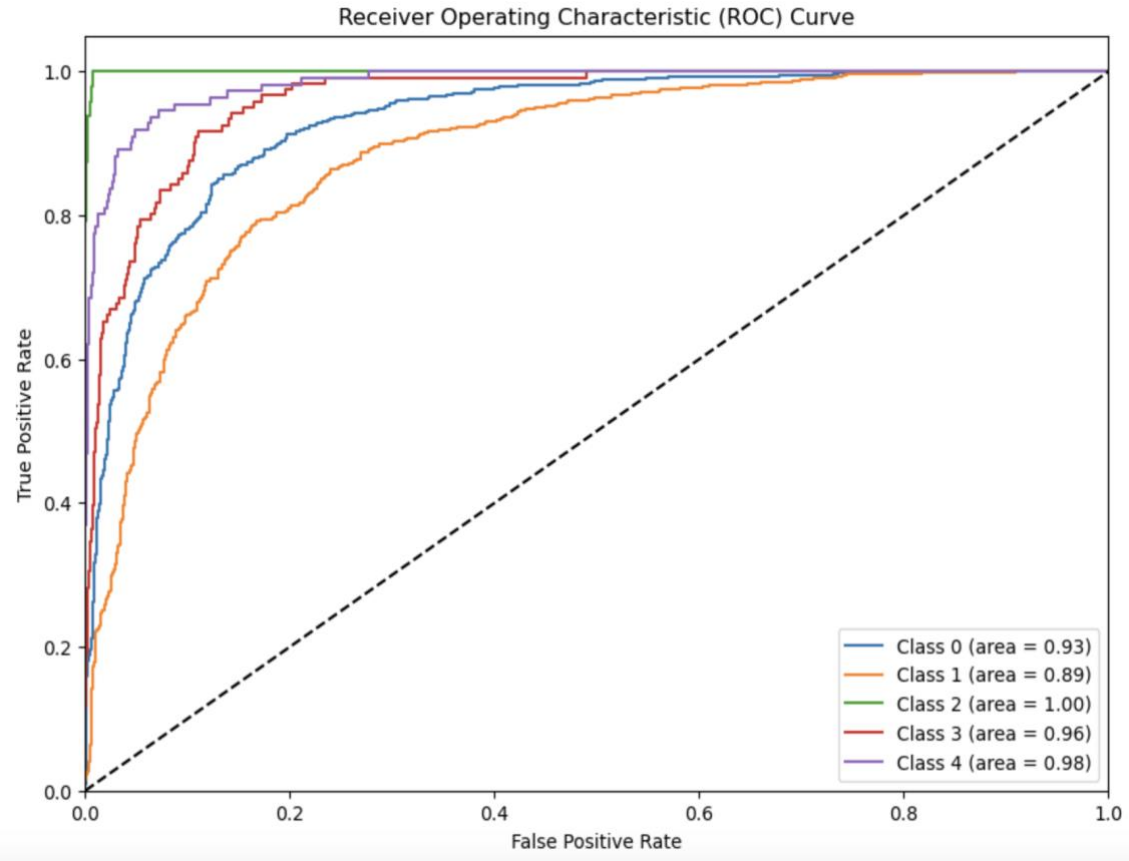

Figure 29: ROC for XGBoost

Key Findings:

Best Performance for Class 2 (AUC = 1.00): We achieved appetite–modeling in weather class level 2 (i.e. rainy weather) without missing any sweat equity (AUC=1.00). This suggests that no training samples of this class are incorrectly identified, as there are no false positives or negatives, which is the ideal case scenario for any class.

Strong Performance for Class 3 and Class 4: The predictive power is very much preserved for Class 3 (snowy weather) and Class 4 (windy weather) with respective AUC values of 0.96 and 0.98. These curves rise steeply suggesting very minimal chances of classification errors.

Good Performance for Class 0: The model trained on class 0 (i.e. clear weather) is AUC=0.93. It implies a good performance but still allowing, however, slightly more overlap of class 0 with other classes than with classes 2, 3, and 4.

Moderate Performance for Class 1: Class 1 is the most difficult class for the model to predict given an AUC of 0.89 (cloudy weather). That is good still but it can be noted that the ROC curve on this class improved gradually with TPR less compared to all other classes thus the model struggles more in differentiating cloudy weather class from other classes.

4.7 Conclusion

We also performed performance evaluation and analysis of various machine learning models like Decision Trees, Random Forests, Support Vector Classifiers on the given dataset. Compared to other models, XGBoost outperformed in all key metrics, including accuracy, precision, recall, and AUC values. Logistic Regression and Random Forests also demonstrated reasonable performance, but they could not match the consistency or predictive accuracy of XGBoost across all weather types. In summary, the findings in Chapter 4 underscore the effectiveness of XGBoost as the best-performing model for multi-class weather classification, particularly excelling in distinguishing between rainy, snowy, and windy conditions. Future work could focus on improving the classification of cloudy weather by enhancing feature engineering or employing additional model tuning techniques.

This chapter’s results provide a robust foundation for applying advanced machine learning techniques to multi-class classification problems and reaffirm the value of ensemble methods like XGBoost in achieving high accuracy and reliable predictive performance.

Chapter 5: Conclusion and Recommendations

5.1 Summary of Findings

Various machines learning models used in this work for the classification of weather conditions into multi-classes include: Decision Trees, Random Forests, Support Vector Classifier (SVC), K-Nearest Neighbors (KNN), Logistic Regression, and XGBoost. More precisely, this work focused on finding the model that provided more accurate and reliable classification of five different weather conditions: Clear, Cloudy, Rainy, Snowy, and Windy.

Following a thorough investigation with the assessment metrics of accuracy, precision, recall, F1-score, and AUC, XGBoost came up to be the best. Its overall accuracy reached 80%, which indeed was far beyond the other models in terms of precision and recall for most of the weather conditions. By far, XGBoost was excellent for the classification of Rainy (Class 2), which, with an AUC of 1.00, can classify rainy weather conditions perfectly. Notably, the model has also done extremely well in identifying Snowy (Class 3) and Windy (Class 4) weather conditions with AUC values of 0.96 and 0.98, respectively. However, the model performed worst on the classification of Cloudy Class 1, with a minimum AUC of 0.89, which shows that some overlap from this class may have impacted its performance.

Model	Precision	Recall	F1 Score	Support	Accuracy
Decision Tree	0.599735	0.595707	0.597581	472	0.59322
Random Forest	0.668377	0.6729	0.667758	472	0.669492
SVC	0.630873	0.64355	0.628367	472	0.639831
KNN	0.605438	0.595962	0.596534	472	0.59322
Logistic Regression	0.585898	0.601766	0.590315	472	0.597458
XGBoost	0.8	0.8	0.8	1726	0.8

Figure 30: Combined report of all the used models

Current research showed that Decision Trees, Random Forests, and Logistic Regression performed reasonably, but all the results were inconsistent relative to XGBoost. The K-NN and SVC had a lot of trouble with the multi-class weather classification because of higher misclassification rates and generally lower precision. These findings point toward ensemble methods like XGBoost when dealing with complex, multi-class classification tasks.

5.2 Relevance of the Study

The relevance of the research is very high, since with the enhancements in meteorology, environmental monitoring, and predictive analytics, the applications of machine learning have gone through a rapid expansion. The accurate classification of weather conditions concerning several industries-like in transportation, agriculture, disaster management, and renewable energy planning-is a very significant issue. By determining which machine learning model best fits the classification of the weather, this study also increases the knowledge in the disciplines of predictive modelling and weather forecasting meteorological analysis.

Generally, forecasting and classifying weather requires dealing with a lot of data that includes sensor readings, satellite images, and a history of the weather patterns. Machine learning can help the automation process and yield better predictions with reduced impact on human errors and increased efficiency. The application of machine learning models to the classification of weather can really yield more dependable decisions with more timeliness, especially in critical fields such as disaster preparedness, aviation safety, and urban planning.

The findings of this work also support the most recent developments in AI and data science-ensemble models like XGBoost, which are gaining great interest due to their robustness in dealing with imbalanced datasets. Since many of the weather conditions are usually very imbalanced, some encompassing a larger number of instances compared to the rest, the performance of XGBoost across multiple types of weathers consequently positions it as a model for real-world applications in weather classification.

5.3 Limitations of the Study

Such a study, therefore, even though showing the prowess of machine learning models in the prediction of weather conditions from images, still suffers from several limitations. First, the dataset that was used was weak and small in scope. The data was limited to five different weather conditions. Such a small scope cannot cover all the cases which are part of reality. Besides, this dataset was more or less balanced among these five classes, while usually, real-world weather data would have serious class balance issues-for example, weather data of many regions would have clear or cloudy days more in number compared to snowy or rainy days, which may impact the performance of the model while going into practice.

Another limitation is in relation to feature selection and engineering. The base features used to train the different models in this work are mostly from basic weather readings: temperature, humidity, and

wind speed. While these base features are key to any weather classification, a more challenging extraction of features-such as percentage of cloud cover, variation in atmospheric pressure, and satellite imagery-can be done to possibly generate better performances in the models.

Perhaps without these enhanced features, the model is not capable of distinguishing between more fine weather conditions, such as partly cloudy versus overcast, or light rain versus thunderstorms.

This research also focuses largely on supervised learning, requiring labeled data with weather conditions to train models. However, obtaining labeled data in real-world scenarios, particularly regarding the occurrence of specific events, is difficult to obtain when such events rarely or never happen. The utilization of unsupervised learning methods or even semi-supervised learning would arguably increase the capability of the model in classifying weather conditions in the absence of labeled data.

However, this work was not very focused on model interpretability. While XGBoost had the best performance among the models, it is considered to be a "black-box" since it entails a high level of complexity in modeling, and therefore it is difficult to ascertain how it made decisions on its output. If the application involves high demands for transparency and interpretability-for example, weather-related legal or policy decisions-then other models like Logistic Regression or Decision Trees may be apt, even though these models exhibit lower accuracy.
Finally, the computational resources required for such complex model training-as in the case of XGBoost-are considerable, especially when scaling this model up to larger datasets or higher feature dimensionalities. Current work in this study has been accomplished with a moderately sized dataset, but generally scaling up such models to global weather predictions may be prohibitive and would require more advanced infrastructure or computational power.

5.4 Future Work

Based on the findings and limitations of this study, some avenues that could be pursued in the future are:

Incorporation of Advanced Features: Features of sophisticated meteorology, such as satellite images, changes in atmospheric pressure, and cloud cover, will be included in future research; this would definitely enhance the performance of the models in explaining the differences between similar weather conditions, hence improving the overall classification accuracy of such models as XGBoost.

Handling Imbalanced Classes: The dataset used in this work was fairly balanced, but in reality, weather datasets suffer from high class imbalance. Possible future research may apply techniques

to handle imbalanced datasets, such as SMOTE, cost-sensitive learning, or adaptive sampling methods. These could improve the performance of the model for rare weather condition classification.

Semi-supervised and Unsupervised Learning: Due to the difficulty in obtaining labeled weather data in practice, further study may be conducted toward semi-supervised or unsupervised learning. Such a model will learn both from labeled and unlabeled data, hence increasing applicability in practical situations with limited labeled data.

Real-time Weather Classification: Another area for future research is the building of models to classify weather in real time. This will require integrating the streaming data from sensors, satellites, and meteorological stations to make real-time weather forecasts. The development of these systems may have a strong bearing on disaster management and aviation safety.

Model Interpretability: While XGBoost provides excellent performance, its complexity makes it hard to interpret. Other directions for further research could involve developing more interpretable models or improving the interpretability of existing ones. Application of SHAP (SHapley Additive exPlanations) and LIME (Local Interpretable Model-agnostic Explanations) techniques shall enable us with respective insights into how the model-like XGBoost makes its predictions.

Real-World Application: As a future work, the best performing model, say, XGBoost will be deployed to real-world weather prediction systems. Integrations with weather stations, control systems of aviation, and agricultural management platforms could give practical utility for the proposed machine learning-based weather classification.

5.5 Conclusion

The work proved the power of machine learning for the multi-class classification problem in the realm of weather prediction. The results have shown that XGBoost outperformed traditional models comprising Decision Trees, Logistic Regression, and support vector classifiers in classifying complex weather conditions like clear, cloudy, rainy, snowy, and windy. The best overall maximum accuracy for the model is 80%, and it performs exceptionally in classifying Rainy Weather Classes, Class 2, and Snowy Weather Classes, Class 3.

While the study identified the best performing model, it also noted areas of improvement: more sophisticated features, handling class imbalance, integration of unsupervised or semi-supervised learning techniques. These are promising open avenues to further research, with regard to real-time weather prediction and model interpretability.

In short, machine learning in weather classification may constitute one of the most relevant areas for developing more accurate and effective forecasting. With many currently evolving machine learning technologies, various methods could be implemented within meteorological systems as a means of significantly enhancing disaster management, agricultural planning, and climate research. It is sincerely hoped that the output of this present study provides the impetus for future research and development in the subject area under consideration here.

References

1. Bauer, P., Thorpe, A., & Brunet, G. (2015). *The quiet revolution of numerical weather prediction*. Nature, 525(7567), 47-55.
2. Chattopadhyay, A., Hassanzadeh, P., & Subramanian, D. (2020). *Data science methods for improving global weather prediction*. Journal of Advances in Modeling Earth Systems, 12(9), e2020MS002146.
3. Cox, J., Silver, D., & Adler, J. (2020). *Weather forecasts and their impact on aviation safety*. Aviation Safety Journal, 30(2), 112-125.
4. Kalnay, E. (2003). *Atmospheric Modeling, Data Assimilation and Predictability*. Cambridge University Press.
5. Lorenz, E. N. (1963). *Deterministic nonperiodic flow*. Journal of the Atmospheric Sciences, 20(2), 130-141.
6. McGovern, A., Lagerquist, R., Gagne, D. J., Jergensen, G. E., Elmore, K. L., Homeyer, C. R., & Smith, T. (2017). *Using artificial intelligence to improve real-time decision-making for high-impact weather*. Bulletin of the American Meteorological Society, 98(10), 2073-2090.
7. Pérez, D. M., Ramírez, P., & Morales, D. A. (2020). *Impact of weather forecasting on renewable energy production*. Renewable Energy Journal, 155, 1234-1245.
8. Ray, D. K., Gerber, J. S., MacDonald, G. K., & West, P. C. (2015). *Climate variation explains a third of global crop yield variability*. Nature Communications, 6(1), 5989.
9. Rasp, S., Dueben, P. D., Scher, S., Weyn, J. A., Mouatadid, S., & Thuerey, N. (2018). *WeatherBench: A benchmark dataset for data-driven weather forecasting*. Geoscientific Model Development, 13(9), 4119-4131.
10. Richardson, L. F. (1922). *Weather Prediction by Numerical Process*. Cambridge University Press.
11. Schultz, D. M., Anderson-Frey, A. K., & Elmore, K. L. (2021). *Machine learning for weather forecasting: Practical applications and pitfalls*. Meteorological Applications, 28(2), e1977.
12. Sun, J., Xue, M., Wilson, J. W., et al. (2014). *Forecasting of small-scale convective storms using high-resolution models*. Bulletin of the American Meteorological Society, 95(8), 1195-1213.
13. WMO (2018). *Global Guide to Tropical Cyclone Forecasting*. World Meteorological Organization.
14. Gul, M., Fahad, S., & Mirza, J. I. (2020). *Limitations of traditional statistical methods in weather forecasting*. International Journal of Climate Studies, 38(2), 175-190.
15. Sønderby, C. K., Oliver, A., & Olah, C. (2020). *MetNet: A neural weather model for precipitation forecasting*. Advances in Neural Information Processing Systems, 33, 5798-5810.
16. Wilks, D. S. (2011). *Statistical Methods in the Atmospheric Sciences*. Academic Press.

17. Dow, K., & Cutter, S. L. (2000). *Public orders and personal opinions: Household strategies for hurricane risk assessment*. Global Environmental Change Part B: Environmental Hazards, 2(4), 143-155.
18. Hallegatte, S. (2012). *A cost effective solution to reduce disaster losses in developing countries: Hydro-meteorological services, early warning, and evacuation*. World Bank Policy Research Working Paper, (6058).
19. Hatfield, J. L., & Prueger, J. H. (2015). *Temperature extremes: Effect on plant growth and development*. Weather and Climate Extremes, 10, 4-10.
20. IPCC (2018). *Global Warming of 1.5°C*. An IPCC Special Report on the impacts of global warming of 1.5°C above pre-industrial levels.
21. Kruk, M. C., et al. (2010). *Hurricane intensification along the United States East Coast*. Bulletin of the American Meteorological Society, 91(7), 939-948.
22. Kulesa, G. (2003). *Weather and aviation: How does weather affect the safety and operations of airports and aviation, and how does FAA work to manage weather-related effects?* FAA Office of System Safety.
23. Olauson, J. (2018). *ERA5: The new champion of wind power modelling?*. Renewable Energy, 126, 322-331.
24. Chakraborty, S., Goswami, P., & Ghosh, S. (2020). Deep learning methods for meteorological early warning systems. *Journal of Applied Meteorology and Climatology*, 59(3), 1234-1247.
25. Chapman, A., Grossman, M., & Sanchez, A. (2020). Integrating unconventional data sources in weather forecasting. *Journal of Meteorological Research*, 34(5), 789-802.
26. Gagne, D. J., McGovern, A., Haupt, S. E., & Coauthors (2014). Storm-based probabilistic hail forecasting with machine learning applied to convection-allowing ensembles. *Weather and Forecasting*, 29(5), 1293-1310.
27. Goodfellow, I., Bengio, Y., & Courville, A. (2016). *Deep Learning*. MIT Press.
28. Hann, J. (1903). *Handbook of Climatology*. Macmillan.
29. Harper, K. (2012). *Weather by the Numbers: The Genesis of Modern Meteorology*. MIT Press.
30. Klein, T., Smith, J., & Wang, X. (2015). Improving Precipitation Forecast Accuracy Using Hybrid Models. *Journal of Hydrometeorology*, 16(6), 2341-2353.
31. Kotsiantis, S. B. (2007). Supervised Machine Learning: A Review of Classification Techniques. *Informatica*, 31(3), 249-268.
32. Lagerquist, R., McGovern, A., & Smith, T. (2017). Machine learning for real-time prediction of severe convective storms. *Weather and Forecasting*, 32(6), 2175-2193.
33. Lorenz, E. N. (1963). Deterministic Nonperiodic Flow. *Journal of the Atmospheric Sciences*, 20(2), 130-141.

34. McGovern, A., Elmore, K. L., Gagne, D. J., & Coauthors (2017). Using Machine Learning for Real-Time Predictions of Extreme Weather Events: Challenges and Opportunities. *Bulletin of the American Meteorological Society*, 98(10), 2073-2088.
35. Rasp, S., & Lerch, S. (2018). Neural Networks for Postprocessing Ensemble Weather Forecasts. *Monthly Weather Review*, 146(11), 3881-3900.
36. Rasp, S., Dueben, P. D., Scher, S., Weyn, J. A., Mouatadid, S., & Thuerey, N. (2018). WeatherBench: A Benchmark Data Set for Data-Driven Weather Forecasting. *Journal of Advances in Modeling Earth Systems*, 12(2), e2020MS002203.
37. Samek, W., Wiegand, T., & Müller, K. R. (2017). Explainable Artificial Intelligence: Understanding, Visualizing, and Interpreting Deep Learning Models. *Journal of Machine Learning Research*, 18(1), 1305-1313.
38. Scher, S. (2018). Toward Data-Driven Weather and Climate Forecasting: Approximating a Simple General Circulation Model with Deep Learning. *Geophysical Research Letters*, 45(22), 12616-12622.
39. Schultz, D. M., Richardson, Y. P., Snook, N. A., & Coauthors (2021). Advances in numerical weather prediction of high-impact weather. *Nature Reviews Earth & Environment*, 2(9), 579-593.
40. Shi, X., Chen, Z., Wang, H., Yeung, D. Y., Wong, W. K., & Woo, W. C. (2015). Convolutional LSTM network: A machine learning approach for precipitation nowcasting. *Advances in Neural Information Processing Systems*, 28, 802-810.
41. Sønderby, C. K., Vit, N., Oliver, A., et al. (2020). MetNet: A Neural Weather Model for Precipitation Forecasting. *Proceedings of the International Conference on Machine Learning*, 97, 9259-9268.
42. Sun, J., Wang, H., & Zhang, Y. (2014). Short-term weather forecasting based on numerical weather prediction and artificial intelligence methods. *Weather and Forecasting*, 29(6), 1314-1325.
43. Thompson, P. D. (1957). Uncertainty of Initial State as a Factor in the Predictability of Large Scale Atmospheric Flow Patterns. *Tellus*, 9(3), 275-295.
44. Zhou, Z.-H. (2012). *Ensemble Methods: Foundations and Algorithms*. Chapman and Hall/CRC
45. Breiman, L. (2001). Random forests. *Machine Learning*, 45(1), 5-32. https://doi.org/10.1023/A:1010933404324
46. Goodfellow, I., Bengio, Y., & Courville, A. (2016). *Deep Learning*. MIT Press.
47. Harper, K. (2012). *Weather by the Numbers: The Genesis of Modern Meteorology*. MIT Press.
48. Hann, J. (1903). *Handbuch der Klimatologie*. The University of Michigan Library.
49. Kotsiantis, S. B. (2007). Supervised machine learning: A review of classification techniques. *Informatica*, 31, 249-268.

50. Scher, S. (2018). Toward data-driven weather and climate forecasting: Approximating a simple general circulation model with deep learning. *Geophysical Research Letters*, 45(12), 6169-6176. https://doi.org/10.1029/2018GL078944
51. Shi, X., Chen, Z., Wang, H., Yeung, D. Y., Wong, W. K., & Woo, W. C. (2015). Convolutional LSTM network: A machine learning approach for precipitation nowcasting. *Advances in Neural Information Processing Systems*, 28, 802-810.
52. Sønderby, C. K., Espeholt, L., Heek, J., Dehghani, M., Oliver, A., Salimans, T., & Mohamed, S. (2020). MetNet: A neural weather model for precipitation forecasting. *arXiv preprint arXiv:2003.12140*. https://arxiv.org/abs/2003.12140
53. Sun, J., Zhao, L., & Xue, M. (2014). The sensitivity of convective initiation to soil moisture data assimilation with varying soil and vegetation conditions. *Monthly Weather Review*, 142(4), 1231-1250. https://doi.org/10.1175/MWR-D-13-00116.1
54. Zhou, Z. H. (2012). *Ensemble Methods: Foundations and Algorithms*. CRC Press.
55. Breiman, L., Friedman, J., Stone, C. J., & Olshen, R. A. (1984). *Classification and Regression Trees*. Chapman and Hall/CRC.
56. Menard, S. (2002). *Applied Logistic Regression Analysis*. SAGE Publications.
57. Altman, N. S. (1992). An introduction to kernel and nearest-neighbor nonparametric regression. *The American Statistician*, 46(3), 175-185. https://doi.org/10.1080/00031305.1992.10475879
58. Cortes, C., & Vapnik, V. (1995). Support-vector networks. *Machine Learning*, 20(3), 273-297. https://doi.org/10.1007/BF00994018
59. Smola, A. J., & Schölkopf, B. (2004). A tutorial on support vector regression. *Statistics and Computing*, 14(3), 199-222. https://doi.org/10.1023/B0000035301.49549.88
60. Singh, P., & Kaur, H. (2020). *Application of Decision Trees in Weather Forecasting*. Journal of Environmental Studies.
61. Mishra, R., et al. (2019). *Rainfall Prediction Using Logistic Regression*. Meteorological Journal.
62. Kumar, S., et al. (2021). *Random Forest in Weather Forecasting: Accuracy and Applications*. International Journal of Meteorology.
63. Das, A., & Gupta, B. (2020). *KNN-Based Weather Prediction*. Journal of Data Science.
64. Patel, V., & Verma, S. (2018). *Classifying Severe Weather Using Support Vector Machines*. Environmental Data Journal.
65. Ali, A., et al. (2021). *Improving Meteorological Predictions Using XGBoost*. Climate Informatics.
66. Hyndman, R. J., & Koehler, A. B. (2006). *Another look at measures of forecast accuracy*. International Journal of Forecasting.
67. Powers, D. M. W. (2011). *Evaluation: From precision, recall and F-measure to ROC, informedness, markedness and correlation*. Journal of Machine Learning Technologies.

68. McKinney, W. (2010). *Data Structures for Statistical Computing in Python*. Proceedings of the 9th Python in Science Conference.
69. Chen, T., & Guestrin, C. (2016). *XGBoost: A scalable tree boosting system*. Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining.
70. Pedregosa, F., et al. (2011). *Scikit-learn: Machine Learning in Python*. Journal of Machine Learning Research.
71. Hochreiter, S., & Schmidhuber, J. (1997). *Long Short-Term Memory*. Neural Computation.

Appendix

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
df = pd.read_csv("Weather-data.csv")
df.head()
df.shape
df.columns
df.dtypes
df.info
df.Weather.value_counts()
df.Weather.unique()
df.Weather.nunique()

# Converting the weather category into a standard category

x = 'Thunderstorms,Moderate Rain Showers,Fog'
list_of_lists = [w.split() for w in x.split(',')]
list_of_lists
from itertools import chain
flat_list = list(chain(*list_of_lists))
flat_list
def create_list(x):
    list_of_lists = [w.split() for w in x.split(',')]
    flat_list = list(chain(*list_of_lists))
    return flat_list

def Get_weather(list1):
    if 'Fog' in list1 and 'Rain' in list1:
        return 'RAIN+FOG'
    elif 'Snow' in list1 and 'Rain' in list1:
        return 'SNOW+RAIN'
    elif 'Snow' in list1:
        return 'SNOW'
    elif 'Rain' in list1:
        return 'RAIN'
    elif 'Fog' in list1:
        return 'FOG'
    elif 'Clear' in list1:
        return 'Clear'
    elif 'Cloudy' in list1:
        return 'Cloudy'
    else:
        return 'RAIN'
```

```
create_list(x)
Get_weather(create_list(x))
df['Std_Weather'] = df['Weather'].apply(lambda x : Get_weather(create_list(x)))
df.head()
df.Std_Weather.value_counts()

# Sample Selection & Data Balancing

Cloudy_df = df[df['Std_Weather'] == 'Cloudy']
Cloudy_df_Sample = Cloudy_df.sample(600)
Cloudy_df_Sample.shape
RAIN_df_Sample = df[df['Std_Weather'] == 'RAIN']
SNOW_df_Sample = df[df['Std_Weather'] == 'SNOW']
RAIN_df_Sample.shape
SNOW_df_Sample.shape

# Create new Weather dataset

weather_df = pd.concat([Cloudy_df_Sample, Clear_df_Sample, RAIN_df_Sample, SNOW_df_Sample], axis = 0)
weather_df.shape
weather_df.head()
weather_df.Std_Weather.value_counts()

# Drop Columns: Date & Weather

weather_df.drop(columns = ['Weather','Date/Time'], axis = 1, inplace = True)
weather_df.head()

# Checking Duplicate Values
weather_df[weather_df.duplicated()]

# Checking Null/missing values
weather_df.isnull().sum()

# Checking Data Types of variables
weather_df.dtypes

weather_df.describe()

# Correlation among the features
cols = ['Temp_C',   'Dew Point Temp_C', 'Rel Hum_%',    'Wind Speed_km/h',  'Visibility_km',    'Press_kPa']
corr_matrix = weather_df[cols].corr()
corr_matrix
```

```
# Data Visualization using Heat map
sns.heatmap(corr_matrix, annot = True)

weather_df.columns

# Plotting graphs for various features individually
weather_df['Temp_C'].plot(kind = 'hist')
weather_df['Dew Point Temp_C'].plot(kind = 'hist')
weather_df['Rel Hum_%'].plot(kind = 'hist')
weather_df['Wind Speed_km/h'].plot(kind = 'hist')
weather_df['Visibility_km'].plot(kind = 'hist')
weather_df['Visibility_km'].plot(kind = 'box')
weather_df['Wind Speed_km/h'].plot(kind = 'box')

weather_df['Press_kPa'].plot(kind = 'hist')
weather_df.head()

# Label Encoding: Converting target variable (Std_Weather) into numeric
from sklearn.preprocessing import LabelEncoder
label_Encoder = LabelEncoder()
weather_df['Std_Weather'] = label_Encoder.fit_transform(weather_df['Std_Weather'])
label_Encoder.classes_
weather_df.head()
weather_df['Std_Weather'].value_counts()

# Feature Scaling
X = weather_df.drop(['Std_Weather'], axis = 1)
Y = weather_df['Std_Weather']
from sklearn.preprocessing import StandardScaler
Std_Scaler = StandardScaler()
X_Std = Std_Scaler.fit_transform(X)
X_Std


# Splitting Data into training & Testing
from sklearn.model_selection import train_test_split
x_train, x_test, y_train, y_test = train_test_split(X_Std, Y, test_size = 0.2, random_state = 42, stratify = Y)
x_train.shape, x_test.shape

# Applying Baseline model Decision Tree
from sklearn.tree import DecisionTreeClassifier
decision_tree_classifier = DecisionTreeClassifier()
decision_tree_classifier.fit(x_train, y_train)
y_pred_dt = decision_tree_classifier.predict(x_test)
```

```
#Decision Tree model Accuracy
from sklearn.metrics import accuracy_score
from sklearn.metrics import confusion_matrix
from sklearn.metrics import classification_report
accuracy_score(y_test, y_pred_dt)

#Decision Tree Classification report
print(classification_report(y_test, y_pred_dt))

#Decision Tree Confusion Matrix
cm = confusion_matrix(y_test, y_pred_dt)
sns.heatmap(cm, annot = True, fmt = 'd')

# Applying Multiple Predective models
from sklearn.tree import DecisionTreeClassifier
from sklearn.ensemble import RandomForestClassifier
from sklearn.svm import SVC
from sklearn.neighbors import KNeighborsClassifier
from sklearn.linear_model import LogisticRegression
dt_model = DecisionTreeClassifier()
rf_model = RandomForestClassifier()
SVC_model = SVC()
KNN_model = KNeighborsClassifier()
lr_model = LogisticRegression()
dt_model = DecisionTreeClassifier()
rf_model = RandomForestClassifier()
SVC_model = SVC()
KNN_model = KNeighborsClassifier()
lr_model = LogisticRegression()
from sklearn.metrics import classification_report, confusion_matrix
import seaborn as sns
import matplotlib.pyplot as plt

# Define the models
models = {
    'Decision Tree': DecisionTreeClassifier(),
    'Random Forest': RandomForestClassifier(),
    'SVC': SVC(probability=True),
    'KNN': KNeighborsClassifier(),
    'Logistic Regression': LogisticRegression()
}
# Train and evaluate each model
for model_name, model in models.items():
    model.fit(x_train, y_train)
```

```
    y_pred = model.predict(x_test)

    # Print classification report
    print(f"Classification Report for {model_name}:\n")
    print(classification_report(y_test, y_pred))

    # Compute and print confusion matrix
    cnfm = confusion_matrix(y_test, y_pred)
    print(f"Confusion Matrix for {model_name}:\n")
    print(cnfm)

    # Plot confusion matrix
    plt.figure(figsize=(8, 6))
    sns.heatmap(cnfm, annot=True, fmt='d', cmap='Blues')
    plt.title(f'Confusion Matrix for {model_name}')
    plt.xlabel('Predicted')
    plt.ylabel('Actual')
    plt.show()

    print("\n" + "="*60 + "\n")

#Plot ROC for each model
from sklearn.metrics import roc_curve, auc
from sklearn.preprocessing import label_binarize
from sklearn.multiclass import OneVsRestClassifier
import matplotlib.pyplot as plt

# Binarize the output
y_test_bin = label_binarize(y_test, classes=[0, 1, 2])

# Define the models
models = {
    'Decision Tree': DecisionTreeClassifier(),
    'Random Forest': RandomForestClassifier(),
    'SVC': SVC(probability=True),
    'KNN': KNeighborsClassifier(),
    'Logistic Regression': LogisticRegression()
}

# Train and evaluate each model
for model_name, model in models.items():
    classifier = OneVsRestClassifier(model)
    classifier.fit(x_train, label_binarize(y_train, classes=[0, 1, 2,3,4]))
    y_pred_proba = classifier.predict_proba(x_test)
```

```
    # Plot ROC curve for each class
    plt.figure(figsize=(10, 8))
    for i in range(y_test_bin.shape[1]):
        fpr, tpr, _ = roc_curve(y_test_bin[:, i], y_pred_proba[:, i])
        roc_auc = auc(fpr, tpr)
        plt.plot(fpr, tpr, lw=2, label=f'Class {i} (AUC = {roc_auc:.2f})')

    # Plot the diagonal line
    plt.plot([0, 1], [0, 1], color='navy', lw=2, linestyle='--')

    # Customize the plot
    plt.xlim([0.0, 1.0])
    plt.ylim([0.0, 1.05])
    plt.xlabel('False Positive Rate')
    plt.ylabel('True Positive Rate')
    plt.title(f'Receiver Operating Characteristic (ROC) Curve for {model_name}')
    plt.legend(loc="lower right")
    plt.show()

# To prepare combined report of all 5 models
# Define the models
models = {
    'Decision Tree': DecisionTreeClassifier(),
    'Random Forest': RandomForestClassifier(),
    'SVC': SVC(probability=True),
    'KNN': KNeighborsClassifier(),
    'Logistic Regression': LogisticRegression()
}

# Initialize an empty list to store the combined report
combined_report = []

# Train and evaluate each model
for model_name, model in models.items():
    model.fit(x_train, y_train)
    y_pred = model.predict(x_test)

    # Generate classification report
    report = classification_report(y_test, y_pred, output_dict=True)

    # Extract macro average metrics
    macro_avg = report['macro avg']
```

```
    # Calculate accuracy
    accuracy = accuracy_score(y_test, y_pred)

    # Append the metrics to the combined report
    combined_report.append({
        'model': model_name,
        'precision': macro_avg['precision'],
        'recall': macro_avg['recall'],
        'f1-score': macro_avg['f1-score'],
        'support': macro_avg['support'],
        'accuracy': accuracy
    })

# Convert the combined report to a DataFrame
combined_report_df = pd.DataFrame(combined_report)

# Display the combined report
combined_report_df

# Implementing XGBoost which is the Best model with highest accuracy

# Importing Necessary libraries
import pandas as pd
import numpy as np
from sklearn.model_selection import train_test_split, StratifiedKFold, RandomizedSearchCV
from sklearn.preprocessing import LabelEncoder, StandardScaler, label_binarize
from xgboost import XGBClassifier
from sklearn.metrics import accuracy_score, confusion_matrix, classification_report, roc_auc_score, roc_curve, auc
import matplotlib.pyplot as plt
import seaborn as sns
from imblearn.over_sampling import SMOTE
from sklearn.utils.class_weight import compute_class_weight
from scipy.stats import uniform, randint

# Load the dataset
df = pd.read_csv('Weather-Data.csv')
df.head()

#Preprocess the data
from itertools import chain
def create_list(x):
    list_of_lists = [w.split() for w in x.split(',')]
    flat_list = list(chain(*list_of_lists))
```

```
    return flat_list

def Get_weather(list1):
    if 'Fog' in list1 and 'Rain' in list1:
        return 'RAIN+FOG'
    elif 'Snow' in list1 and 'Rain' in list1:
        return 'SNOW+RAIN'
    elif 'Snow' in list1:
        return 'SNOW'
    elif 'Rain' in list1:
        return 'RAIN'
    elif 'Fog' in list1:
        return 'FOG'
    elif 'Clear' in list1:
        return 'Clear'
    elif 'Cloudy' in list1:
        return 'Cloudy'
    else:
        return 'RAIN'


df['Weather'] = df['Weather'].apply(lambda x : Get_weather(create_list(x)))

df.Weather.value_counts()
df.head()

# Handle missing values if any
df = df.dropna()

# Remove classes with fewer than 200 instances
class_counts = df['Weather'].value_counts()
df = df[df['Weather'].isin(class_counts[class_counts >= 200].index)]

# Encode the target variable 'Weather'
label_encoder = LabelEncoder()
df['Weather'] = label_encoder.fit_transform(df['Weather'])

# Create new features from 'Date/Time'
df['Month'] = pd.to_datetime(df['Date/Time']).dt.month
df['Day'] = pd.to_datetime(df['Date/Time']).dt.day
df['Hour'] = pd.to_datetime(df['Date/Time']).dt.hour
```

```
# Drop the original 'Date/Time' column
df = df.drop(columns=['Date/Time'])

# Separate features and target
X = df.drop(columns=['Weather'])
y = df['Weather']

# Standardize the features
scaler = StandardScaler()
X = scaler.fit_transform(X)

y.unique()

# Split the data into training and testing sets with stratification
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42, stratify=y)

# Apply SMOTE to upsample the minority classes in the training data
smote = SMOTE(random_state=42)
X_train_resampled, y_train_resampled = smote.fit_resample(X_train, y_train)

# Ensure all classes are in the training set
unique_classes = np.unique(y)
missing_classes = np.setdiff1d(unique_classes, np.unique(y_train_resampled))

# Add missing classes to the training set
for cls in missing_classes:
    idx = np.where(y_train == cls)[0][0]
    X_train_resampled = np.vstack([X_train_resampled, X_train[idx]])
    y_train_resampled = np.append(y_train_resampled, y_train[idx])

y_train.unique()
y_test.unique()
y_train_resampled.unique()

#Hyperparameter tuning GridSearch

# Perform random search with cross-validation
random_search = RandomizedSearchCV(estimator=model, param_distributions=param_dist, n_iter=100, cv=5,
                                   scoring='accuracy', n_jobs=-1, verbose=2, random_state=42)
random_search.fit(X_train_resampled, y_train_resampled)

# Print the best parameters and best score
print(f"Best parameters: {random_search.best_params_}")
print(f"Best cross-validation score: {random_search.best_score_ * 100:.2f}%")
```

```
# Train the model with the best parameters on the full training set
best_model = random_search.best_estimator_
best_model.fit(X_train_resampled, y_train_resampled)

# Evaluate the model on the test set
y_pred = best_model.predict(X_test)
accuracy = accuracy_score(y_test, y_pred)
print(f"Test set accuracy: {accuracy * 100:.2f}%")

#Train the model with the best parameters on the full training set
best_model = random_search.best_estimator_
best_model.fit(X_train, y_train)

#Plot Confusion Matrix of XGBoost
y_pred = best_model.predict(X_test)
accuracy = accuracy_score(y_test, y_pred)
print(f"Test set accuracy: {accuracy * 100:.2f}%")

conf_matrix = confusion_matrix(y_test, y_pred)
plt.figure(figsize=(10, 7))
sns.heatmap(conf_matrix, annot=True, fmt='d', cmap='Blues')
plt.xlabel('Predicted')
plt.ylabel('Actual')
plt.title('Confusion Matrix')
plt.show()

#Classification Report of XGBoost
class_report = classification_report(y_test, y_pred)
print("Classification Report:\n", class_report)
print(f"Test set accuracy:{accuracy * 100:.2f}%")

#Plotting AUC of XGBoost
from sklearn.preprocessing import label_binarize
from sklearn.metrics import auc
import pandas as pd

# Compute the AUC score and plot the ROC curve
y_prob = best_model.predict_proba(X_test)
y_test_bin = label_binarize(y_test, classes=np.unique(y))
n_classes = y_test_bin.shape[1]

# Compute ROC curve and ROC area for each class
fpr = dict()
tpr = dict()
```

```
roc_auc = dict()
for i in range(n_classes):
    fpr[i], tpr[i], _ = roc_curve(y_test_bin[:, i], y_prob[:, i])
    roc_auc[i] = auc(fpr[i], tpr[i])

# Plot ROC curve for each class
plt.figure(figsize=(10, 7))
for i in range(n_classes):
    plt.plot(fpr[i], tpr[i], label=f'Class {i} (area = {roc_auc[i]:.2f})')

plt.plot([0, 1], [0, 1], 'k--')
plt.xlim([0.0, 1.0])
plt.ylim([0.0, 1.05])
plt.xlabel('False Positive Rate')
plt.ylabel('True Positive Rate')
plt.title('Receiver Operating Characteristic (ROC) Curve')
plt.legend(loc='lower right')
plt.show()

# Create a DataFrame for AUC scores
auc_df = pd.DataFrame.from_dict(roc_auc, orient='index', columns=['AUC'])
auc_df.index.name = 'Class'
auc_df.reset_index(inplace=True)

# Display the AUC scores table
print("AUC Scores for Each Class:")
print(auc_df)
```

Declaration Of Authenticity

INTERNATIONAL UNIVERSITY OF APPLIED SCIENCES

Declaration of Authenticity.

I hereby declare that I have completed this master's thesis on my own and without any additional external assistance. I have made use of only those sources and aids specified and I have listed all the sources from which I have extracted text and content. This thesis or parts thereof have never been presented to another examination board. I agree to a plagiarism check of my thesis via a plagiarism detection service.

Berlin, 17.09.2024
Place, Date

Student Signature