Weather Prediction System using Machine Learning

Submitted in the partial fulfillment of the requirements for the degree of B.Tech in Computer Science and Business Systems

by

HARSH SHUKLA (22CB1045)

ADITYA SINGH (22CB1047)

SAHIL BORADE(22CB1123)

SHIVRAJH PAWAR (22CB1055)

Supervisor

Mrs. Siddhi Kadu

D Y PATIL
DEEMED TO BE
UNIVERSITY
RAMRAO ADIK
INSTITUTE OF TECHNOLOGY
NAVI MUMBAI

Department of Computer Engineering

Ramrao Adik Institute of Technology

Sector 7, Nerul, Navi Mumbai

(Under the ambit of D. Y. Patil Deemed to be University) November 2024

Ramrao Adik Institute of Technology

(Under the ambit of D. Y. Patil Deemed to be University)

Dr. D. Y. Patil Vidyanagar, Sector 7, Nerul, Navi Mumbai 400 706

CERTIFICATE

This is to certify that, the Mini Project-III report entitled

Weather Prediction System using Machine Learning
is a bonafide work done by

HARSH SHUKLA (22CB1045)

ADITYA SINGH (22CB1047)

SAHIL BORADE(22CB1123)
SHIVRAJH PAWAR (22CB1055)

and is submitted in the partial fulfillment of the requirement for the degree of

B.Tech in Computer Science and Business Systems

to the

D. Y. Patil Deemed to be University

Supervisor (Mrs. Siddhi Kadu)	Project Co-ordinator (Dr. Shamal Salunke)

Head of Department
(Dr. Amarsinh V. Vidhate)

Principal
(Dr. Mukesh D. Patil)

Mini Project Report - III Approval

This is to certify that the Mini Project - III entitled *Weather Prediction System Using ML* is a bonafide work done by *HARSH SHUKLA (22CB1045)*, *ADITYA SINGH (22CB1047)*, *SAHIL BORADE (22CB1123)*, and *SHIVRAJH PAWAR (22CB1055)* under the supervision of *Mrs. Siddhi Kadu*. This Mini Project is approved in the partial fulfillment of the requirement for the degree of *B.tech in Computer Science and Business Systems*

Internal Examiner :

1. ....

2. ....

External Examiners :

1. ....

2. ....

Date : .../.../......

Place : ....

DECLARATION

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

Date: ________________

HARSH SHUKLA (22CB1045)
ADITYA SINGH (22CB1047)

SAHIL BORADE(22CB1123)
SHIVRAJH PAWAR (22CB1055)

Abstract

This report presents the development of a Weather Prediction System that utilizes machine learning algorithms to forecast weather patterns with high accuracy. By analyzing historical weather data, the system aims to provide reliable, real-time weather predictions. The approach involves preprocessing techniques such as data cleaning and normalization, followed by the application of various machine learning models, including Linear Regression, Decision Trees, and Neural Networks. These models were trained using data from reputable meteorological sources, and their performance was assessed in terms of accuracy, precision, and efficiency. The process included data collection, feature selection to identify key weather predictors, and model training. The final system was tested against real-world weather conditions, demonstrating its ability to predict future weather patterns accurately within a given time frame.

Contents

List of Figures

Chapter 1

Introduction

Weather profoundly impacts our daily lives, influencing everything from our clothing choices to our travel plans. For farmers, it affects crop yields, while for travelers, it shapes flight schedules and road safety. Yet, despite its importance, accurate weather forecasting remains a challenge, and unexpected conditions—whether sudden storms, heatwaves, or cold fronts—disrupt lives worldwide[1]. Today, extreme weather events are increasing in frequency, posing serious risks. Floods from heavy rains and health risks from heatwaves are daily concerns, especially in regions with limited access to real-time weather information. In developing areas, unreliable forecasts can lead to crop losses or endanger travelers who rely on accurate weather updates.Traditional weather prediction methods, while effective, have limitations. They depend on complex models that require continuous human oversight and often lag in adapting to new data. This is where machine learning (ML) makes a difference. By analyzing vast amounts of historical and real-time data, ML can deliver more precise, timely, and localized predictions, reducing forecast errors and helping people respond effectively to changing weather. This report examines how machine learning can improve weather forecasting, making it more reliable and accessible for everyone—from everyday citizens to professionals in weathersensitive industries.

1.1 Overview

Weather forecasting is essential for sectors like agriculture and transportation. Traditionally, it has relied on complex atmospheric models and expert interpretation. With ML, however, we can now automate and enhance prediction accuracy.This report examines how ML can transform weather forecasting by using extensive datasets—both historical and real-time—to uncover patterns often missed by conventional methods. Models like decision trees, support vector machines (SVM), and neural networks analyze vast amounts of data, delivering faster and more accurate predictions. By employing supervised learning, these models can forecast variables such as temperature, humidity, precipitation, and wind patterns.While ML-driven forecasting offers benefits like speed, data handling, and continuous learning, challenges remain in data quality, model adaptability, and computational requirements. Nonetheless, ML's integration into weather prediction holds great potential to improve accuracy and support decision-making across industries[2].

1.2 Motivation

The motivation to explore machine learning for weather prediction comes from seeing firsthand how unpredictable weather impacts lives. Farmers struggle with planting schedules, commuters are caught off guard by storms, and entire communities face sudden floods and heatwaves without warning. One experience that deeply influenced me was witnessing a sudden, unpredicted storm that left a local community in chaos. With roads blocked, homes flooded, and no timely alerts, people were left unprepared—despite having weather apps that either generalized or delivered predictions too late to help. Recognizing the potential of machine learning to address these gaps, I saw an opportunity to create more reliable, timely, and localized forecasts. ML's capacity to process vast data, identify complex patterns, and continuously improve makes it a powerful tool for advancing weather prediction. My goal is to contribute to systems that help people prepare for weather conditions more effectively, reducing risks and enhancing daily life for individuals and communities alike.

1.3 Problem Statement and Objectives

Problem Statement:

Accurate and timely weather forecasts are crucial for daily activities, yet many people still rely on unreliable or overly generalized predictions. This issue is especially impactful for farmers, travelers, and emergency responders, whose decisions often depend on precise weather data. Traditional forecasting methods, while useful, can struggle with local variability and the challenge of integrating new data quickly.In rural and underserved regions, access to up-todate weather information is often limited, leaving communities at risk of sudden, unpredictable weather changes. This highlights the urgent need for localized, real-time, and accurate weather forecasts.

Objectives:

This project seeks to develop a machine learning-based weather prediction system that provides accurate, timely, and localized forecasts. By analyzing both historical and real-time weather data, the system will detect complex weather patterns and offer precise, adaptable predictions. The key goals include:

- To Improving forecast accuracy by using machine learning models that continuously learnfrom past data and real-time inputs.
- Delivering localized forecasts tailored to specific regions and communities.
- To Minimizing delays between data collection and prediction to enable faster weather alerts.
- To Expanding access to reliable weather data in underserved areas to support betterdecision-making and preparation.

1.4 Organization of the report

This report provides a structured overview of the machine learning-based weather prediction system using K-Nearest Neighbors (KNN), Support Vector Machines (SVM), Gradient Boosting, and Extreme Gradient Boosting. It is organized as follows: Chapter 2 shows the review of existing research and methodologies in weather prediction, focusing on machine learning approaches. This chapter examines the strengths and limitations of traditional statistical methods and ML models like KNN, SVM, and ensemble methods, including recent improvements in data accuracy and model effectiveness.Chapter 3 outlines the main challenges the prediction system addresses, such as achieving accurate, localized, and timely forecasts. It defines problem areas (e.g., temperature and precipitation prediction) and explains the use of machine learning techniques. The system architecture, covering both hardware and software requirements, is also described. Chapter 4 provides the detailed development process, including data sources, feature engineering, and the application of each ML model. The chapter also covers training, validation, and evaluation methods, with a rationale for selecting KNN, SVM, Gradient Boosting, and XGBoost. Chapter 5 presents model performance results, comparing accuracy, precision, recall, Key findings and model strengths and limitations are discussed, along with suggestions for future improvements, such as adding data sources, refining models, and expanding system capabilities. By the end, readers will understand the design, methodology, results, and potential for future enhancements of this weather prediction system.

Chapter 2

Literature Survey

In this study, the authors focus on addressing the limitations of traditional weather forecasting models, which are often slow, expensive, and reliant on complex computations that require high-performance systems. These models can also produce inaccurate results due to incomplete or poor-quality data[1]. To overcome these challenges, the researchers adopted a machine learning (ML)-based approach, using historical weather data from Nashville and nearby cities. By incorporating data from multiple surrounding locations, the model could capture broader regional weather patterns, which is critical since weather in one city can influence conditions in neighboring areas. Various machine learning algorithms were tested, including Random Forest Regression, Support Vector Regression, and Ridge Regression. The results showed that Random Forest Regression was the most effective, especially when trained with data from multiple cities. This method, which combines multiple decision trees, was found to be particularly robust and able to handle the complex relationships found in weather patterns. The key findings from the study include:

1) Improved Accuracy: Using data from surrounding cities helped reduce prediction errors, proving that regional data significantly improves the accuracy of local weather forecasts.

2) Efficiency and Accessibility: The chosen ML models were both efficient and accurate, capable of providing reliable forecasts without the need for expensive computational resources. This means the system could be deployed on more affordable devices, making accurate weather forecasts accessible to a wider range of users.

[2] The paper Smart Weather Prediction Using Machine Learning explores the use of machine learning (ML) to enhance the accuracy and efficiency of weather forecasting, offering an alternative to traditional, resource-heavy physical models. The objective is to leverage ML techniques, trained on historical weather data, to generate quicker and more reliable forecasts that can run on standard computing devices, making them more accessible and scalable. Approach and Methods The researchers used a comprehensive dataset that spans 21 years, capturing a variety of weather parameters such as temperature, humidity, wind speed, and pressure. Several ML algorithms were tested, each chosen for its ability to handle different aspects of the weather data:

K-Nearest Neighbors (KNN): This model predicts weather outcomes based on historical similarities in local data, making it simple but effective for forecasting.

Support Vector Machine (SVM): Used for classification, SVM helps distinguish between different weather events, like rain or fog, by forming decision boundaries in the data.

Decision Tree and Random Forest: Decision Trees are easy to interpret and make decisions hierarchically, while Random Forest, an ensemble of multiple decision trees, improves accuracy and reduces the risk of overfitting.

Gradient Boosting and XGBoost: These advanced techniques iteratively improve predictions by correcting errors from previous models. XGBoost, in particular, is noted for its speed and

efficiency with large datasets. Other models like AdaBoost, Na¨ıve Bayes, and Logistic Regression were also tested to compare performance across different methods. The dataset was split into training and testing sets.
Results and Findings The results showed that:

1) Gradient Boosting achieved the highest accuracy (81.67)

2) XGBoost and Random Forest also provided strong performance, with accuracy near 80Simpler models like Decision Tree and Na¨ıve Bayes delivered moderate results, while KNN offered acceptable performance but was outpaced by more sophisticated models.

2.1 Survey of Existing System

In recent years, machine learning (ML) has gained prominence in weather prediction due to its ability to analyze large, complex datasets and produce more accurate forecasts. Traditional forecasting methods, which rely heavily on physical atmospheric models, are increasingly being supplemented or replaced by data-driven techniques such as K-Nearest Neighbors (KNN), Support Vector Machines (SVM), Gradient Boosting (GB), and Extreme Gradient Boosting (XGBoost). Each of these methods has unique strengths, and they have been applied to improve the accuracy and reliability of weather forecasts.
1. K-Nearest Neighbors (KNN)

Weather Prediction KNN is a simple yet effective algorithm that uses historical weather data to predict future conditions. It identifies similar past weather patterns (neighbors) and uses them to forecast outcomes like temperature or precipitation. KNN is particularly useful for temperature and rainfall predictions but can become computationally expensive with large datasets and struggles with high-dimensional data.
2. Support Vector Machines (SVM)

Weather Classification SVM is well-suited for classification tasks, such as predicting binary weather events (e.g., ”rain/no rain” or ”storm/no storm”). The algorithm works by finding a decision boundary that best separates the different classes based on input features like temperature, humidity, and pressure. While SVM is effective for predicting weather events like rainfall or storms, it can be computationally demanding and requires careful parameter tuning to avoid overfitting.

3.Gradient Boosting (GB)

Weather Forecasting Gradient Boosting is an ensemble learning technique that builds a series of weak models, usually decision trees, and combines them to create a more accurate predictive model. It is especially useful for predicting continuous weather variables like temperature or wind speed. While GB is resilient to incomplete data and provides high accuracy, it can easily overfit if not tuned properly and typically requires significant training time.
4. Extreme Gradient Boosting (XGBoost)

Weather Prediction XGBoost is an optimized version of Gradient Boosting, offering improved speed, efficiency, and accuracy. It includes regularization techniques to prevent overfitting,

making it particularly effective for complex weather prediction tasks. XGBoost excels in identifying important features like humidity and temperature in weather outcomes. However, it remains computationally intensive and requires careful tuning of hyperparameters.
Key Observations and Summary

Machine learning techniques such as KNN, SVM, GB, and XGBoost have shown great promise in enhancing weather prediction accuracy, particularly for localized events. Each method has its benefits:

1) KNN is simple and effective for basic predictions but struggles with large datasets.
2) SVM is powerful for classification tasks but demands significant computational resources and fine-tuning.
3) Gradient Boosting provides strong performance for continuous variables but can overfit and is time-consuming.
4) XGBoost improves upon GB by offering faster training and better handling of overfitting, though it remains resource-heavy.

2.2 Limitations of Existing System

Machine learning techniques like K-Nearest Neighbors (KNN), Support Vector Machines (SVM), Gradient Boosting (GB), and Extreme Gradient Boosting (XGBoost) have improved weather prediction, but they still face key challenges that limit their practical application:
K-Nearest Neighbors (KNN):

- Scalability: KNN becomes inefficient with large datasets due to the need to compareevery input to every training point.
- Memory Usage: It requires storing all data points in memory, which is problematic for real-time forecasting.

Support Vector Machines (SVM):

High Computational Cost: SVM with non-linear kernels is computationally expensive, making it impractical for large weather datasets.
Parameter Tuning: Finding the right hyperparameters requires extensive trial and error. Limited Interpretability: SVM is a ”black-box” model, making it difficult to understand how predictions are made.
Gradient Boosting (GB): Overfitting: GB’s sequential nature makes it prone to overfitting, particularly with noisy data.
Long Training Times: GB requires significant time for training, which limits its use for large datasets or real-time applications.
Handling Missing Data: GB models struggle with incomplete data, requiring additional preprocessing.
Extreme Gradient Boosting (XGBoost):

Computational Demands: XGBoost is optimized for speed but still requires significant computational resources, especially for real-time forecasting.

Overfitting: Like GB, XGBoost can overfit without proper tuning.

Lack of Interpretability: Despite offering feature importance metrics, XGBoost remains difficult to interpret, hindering decision-making in fields like agriculture and disaster management. Common Challenges:

Data Dependency: These models require high-quality, large-scale datasets, and poor data quality reduces their effectiveness.

Scalability Issues: As datasets grow, these models struggle to scale, especially in real-time applications.

Real-Time Adaptability: ML models often cannot quickly adapt to new data, making them less effective for rapidly changing weather conditions.

Localized Accuracy: These methods often fail to capture fine-grained, localized weather variations critical for applications like agriculture and emergency response.

Chapter 3

Proposed System

Our weather prediction system uses machine learning to deliver accurate and timely forecasts, essential for daily life and critical decision-making. By combining K-Nearest Neighbors (KNN), Support Vector Machine (SVM), Gradient Boosting, and Extreme Gradient Boosting (XGBoost), we aim to identify patterns in weather data for precise predictions.

Here’s how each technique contributes:

1. K-Nearest Neighbors (KNN): KNN predicts weather by comparing current conditions tosimilar past data points. It’s simple yet effective for quick, interpretable forecasts, helping to predict outcomes like temperature changes or precipitation.
2. Support Vector Machine (SVM): SVM creates boundaries between different weather patterns (e.g., rainy vs. clear) by analyzing large datasets. It excels at classifying complex or nonlinear weather patterns, making it a powerful tool for weather prediction.
3. Gradient Boosting: Gradient Boosting builds a series of models that correct each other’smistakes, improving prediction accuracy over time. It’s especially useful for detecting subtle trends in weather data, such as temperature changes.
4. Extreme Gradient Boosting (XGBoost): XGBoost is an optimized version of GradientBoosting, designed for faster performance and better handling of large datasets. It’s perfect for real-time forecasting, delivering quick and accurate predictions even with vast data.

Together, these methods form a robust system that combines simplicity, accuracy, and adaptability. By leveraging each algorithm, our goal is to create a reliable, real-time weather prediction system that helps people make informed decisions and plan safely.

3.1 Problem Statement

This report introduces a Weather Prediction System that leverages four machine learning algorithms—KNearest Neighbors (KNN), Support Vector Machines (SVM), Gradient Boosting (GB), and Extreme Gradient Boosting (XGBoost)—to forecast weather parameters like temperature, precipitation, and wind speed. By incorporating advanced machine learning techniques, the system improves on traditional forecasting methods in key areas:

1. To Enhanced Accuracy the system utilizes historical and real-time data to capture complex weather patterns, providing more accurate predictions.

2. To Real-Time adaptability It updates forecasts quickly as new data becomes available,crucial for sectors like agriculture and emergency management.
3. To Scalability capable of processing large volumes of data efficiently, enabling fast andscalable predictions.
4. To Localized Forecasting delivers region-specific forecasts, addressing the variability inweather across different locations.

3.2 Proposed Methodology/Techniques

This project proposes a Weather Prediction System using four advanced machine learning algorithms—K-Nearest Neighbors (KNN), Support Vector Machines (SVM), Gradient Boosting (GB), and Extreme Gradient Boosting (XGBoost)—to forecast weather parameters such as temperature, precipitation, and wind speed. The system enhances these models through optimization techniques to improve accuracy, scalability, and real-time prediction capabilities.

- KNN: uses weighted voting and dimensionality reduction methods like PCA, which enhances prediction accuracy and reduces computational overhead, especially when dealing with large datasets.

  Algorithm

  Step 1: Start.

  Step 2: Load the dataset containing weather features (e.g., temperature, humidity) and the target variable (e.g., weather classification like rain, no rain).
  Step 3: Split the dataset into training and testing sets.

  Step 4: Initialize the K-Nearest Neighbors (KNN) classifier, choosing the number of neighbors (e.g., k = 5).
  Step 5: Train the KNN classifier using the training data.

  Step 6: Predict the weather on the test data using the trained model.

  Step 7: Calculate the accuracy of the prediction by comparing the predicted values to the actual values.
  Step 8: Output the accuracy.

  Step 9: Stop.

- SVM: utilizes Radial Basis Function (RBF) or sigmoid kernels to handle nonlinear weather patterns. Hyperparameter optimization (such as adjusting C and gamma) improvesmodel performance for more accurate forecasts.

  Algorithm

  Step 1: Start.

  Step 2: Load the dataset containing weather features and the target variable.

  Step 3: Split the dataset into training and testing sets.

Step 4: Initialize the Support Vector Machine (SVM) classifier with a suitable kernel function (e.g., linear or rbf).
Step 5: Train the SVM classifier using the training data.

Step 6: Predict the weather on the test data using the trained model.

Step 7: Calculate the accuracy of the prediction by comparing predicted values to actual values.
Step 8: Output the accuracy.

Step 9: Stop.

- GB: combines multiple weak models (decision trees) to improve prediction power. The system applies regularization (L1, L2) to avoid overfitting and uses a dynamic learning rate to accelerate convergence without losing accuracy. Algorithm

  Step 1: Start.

  Step 2: Load the dataset containing weather features and the target variable.

  Step 3: Split the dataset into training and testing sets.

  Step 4: Initialize the Gradient Boosting classifier with parameters like n estimators (number of boosting stages), learning rate,and max depth.
  Step 5: Train the Gradient Boosting model using the training data.

  Step 6: Predict the weather on the test data using the trained model.

  Step 7: Calculate the accuracy of the prediction by comparing predicted values to actual values.
  Step 8: Output the accuracy.

  Step 9: Stop.

- XGBoost: an advanced version of Gradient Boosting, improves prediction efficiency by including regularization, handling missing data effectively, and employing early stopping to prevent overfitting and reduce training time.

  Algorithm

  Step 1: Start.

  Step 2: Load the dataset containing weather features and the target variable.

  Step 3: Split the dataset into training and testing sets.

  Step 4: Initialize the XGBoost classifier with parameters like n estimators, max depth, learning rate, etc.
  Step 5: Train the XGBoost model using the training data.

  Step 6: Predict the weather on the test data using the trained model.

  Step 7: Calculate the accuracy of the prediction by comparing predicted values to actual values.
  Step 8: Output the accuracy.

Step 9: Stop.

The system compares the models using k-fold cross-validation and evaluation metrics like

MAE, RMSE, and R². Based on performance, the best model is selected for real-time forecasting. The system follows a modular design, comprising data collection, preprocessing, model training, prediction generation, and visualization, ensuring an efficient and scalable solution for weather forecasting.

3.3 System Design

The Weather Prediction System leverages advanced machine learning algorithms—KNN, SVM, GB, and XGBoost—to provide real-time weather forecasts. The system is modular, ensuring efficient processing and accurate predictions, with key components:

1. System Overview : The system handles data collection, preprocessing, model training, prediction, evaluation, and user interaction. It processes historical and real-time weather data, supporting industries like agriculture, transportation, and emergency management.
2. Modules : Model Training: Trains models (KNN, SVM, GB, XGBoost) using optimized hyperparameters, selecting the best performing model based on metrics like RMSE, MAE, and R-squared.
3. Prediction: Generates real-time weather forecasts from the trained model, displaying results via charts or maps.

Evaluation and Feedback: Compares predictions with actual results, retraining the model to improve accuracy over time.

4. Platform and Tools: Programming Language: Python, with libraries like scikit-learn, XG-

Boost, pandas, and NumPy.

Database: MySQL/PostgreSQL for structured data, MongoDB for unstructured data.

Web Framework: Flask or Django for creating a real-time web interface.

3.4 Details of Hardware/Software Requirement

To implement the Weather Prediction System effectively, a set of software tools and platforms is needed to manage data, train models, evaluate results, and visualize outcomes. Here's a detailed breakdown of the essential software components:
Programming Language

Python: Python is chosen for this project because of its simplicity and strong ecosystem of libraries. It has robust support for data handling, machine learning, and visualization, making it ideal for implementing and fine-tuning predictive models. Python's extensive community resources also aid in troubleshooting and optimizing code.

Machine Learning Libraries

scikit-learn: This library provides ready-to-use implementations for KNN, SVM, and various other machine learning algorithms. It also includes tools for data preprocessing, model evaluation, and hyperparameter tuning, all of which are essential for building reliable prediction models.

XGBoost: XGBoost is a high-performance library specifically for Gradient Boosting. It offers advanced features such as regularization, early stopping, and efficient handling of large datasets, making it ideal for improving accuracy in complex prediction tasks.

Data Processing and Analysis

- pandas: pandas is essential for data manipulation tasks, such as cleaning, organizing, and transforming weather data. It is particularly useful for feature engineering and working with time-series data.
- NumPy: A core library for numerical computations, NumPy supports array manipulation and vectorized operations, which are crucial for handling large datasets efficiently.
- Matplotlib and Seaborn: These libraries are used to visualize data trends, model performance, and prediction accuracy. They allow for the creation of plots and charts to understand the model outputs and communicate insights visually.
- SciPy: SciPy complements NumPy by providing additional functionality for scientific computing, particularly in statistical analysis and optimization, which can aid in refining model parameters.

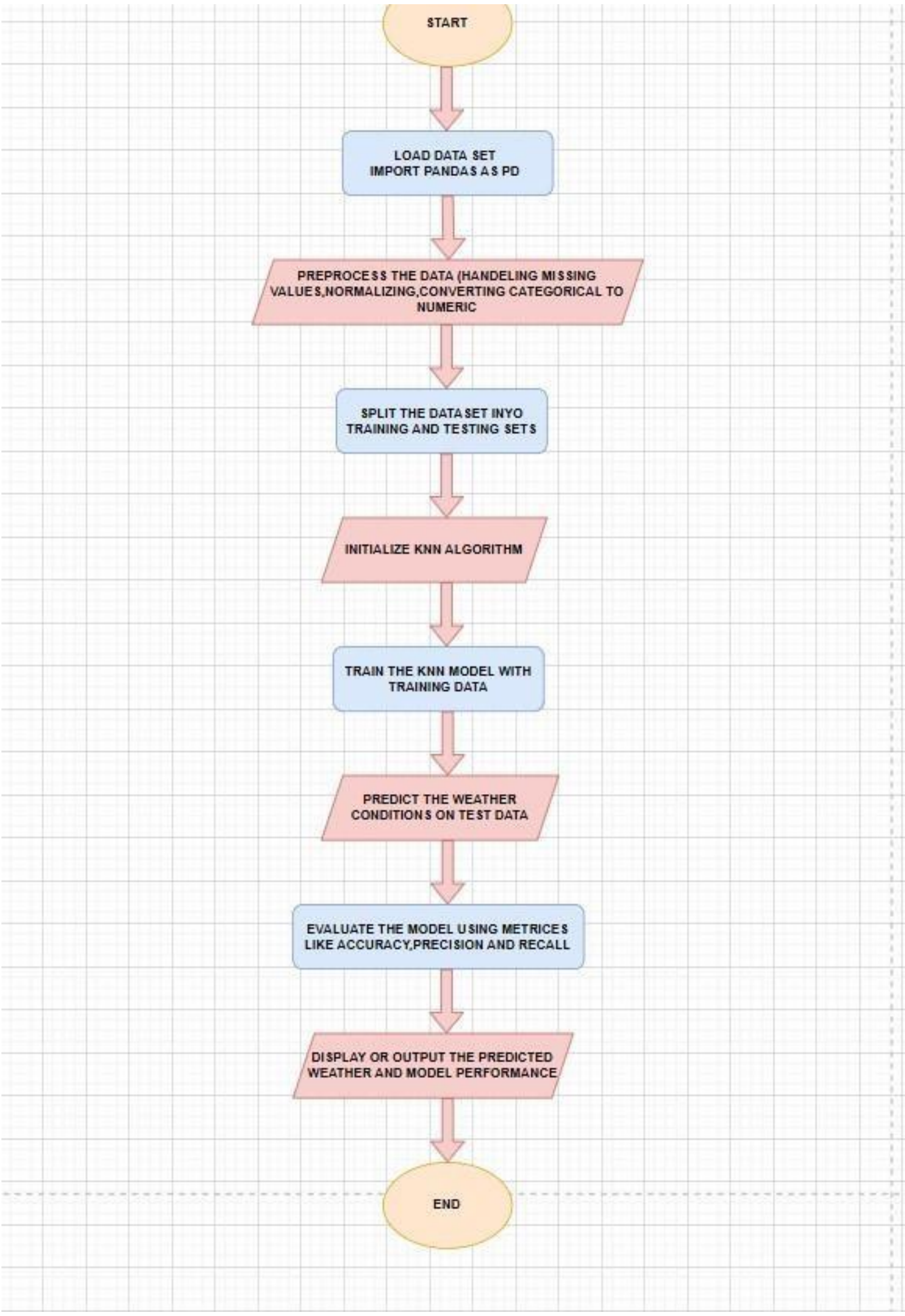

START
LOAD DATA SET
IMPORT PANDAS AS PD
PREPROCESS THE DATA (HANDELING MISSING VALUES,NORMALIZING,CONVERTING CATEGORICAL TO NUMERIC
SPLIT THE DATASET INYO TRAINING AND TESTING SETS
INITIALIZE KNN ALGORITHM
TRAIN THE KNN MODEL WITH TRAINING DATA
PREDICT THE WEATHER CONDITIONS ON TEST DATA
EVALUATE THE MODEL USING METRICES LIKE ACCURACY,PRECISION AND RECALL
DISPLAY OR OUTPUT THE PREDICTED WEATHER AND MODEL PERFORMANCE
END

Fig 3.3.1 K-Nearest Neighbors Architecture (KNN) KNN:This flowchart illustrates the process of building a K-Nearest Neighbors (KNN) model for weather prediction. It starts with data loading and preprocessing, followed by splitting the data, initializing, and training the KNN algorithm. Finally, it evaluates model performance and outputs predictions using metrics like accuracy and recall.

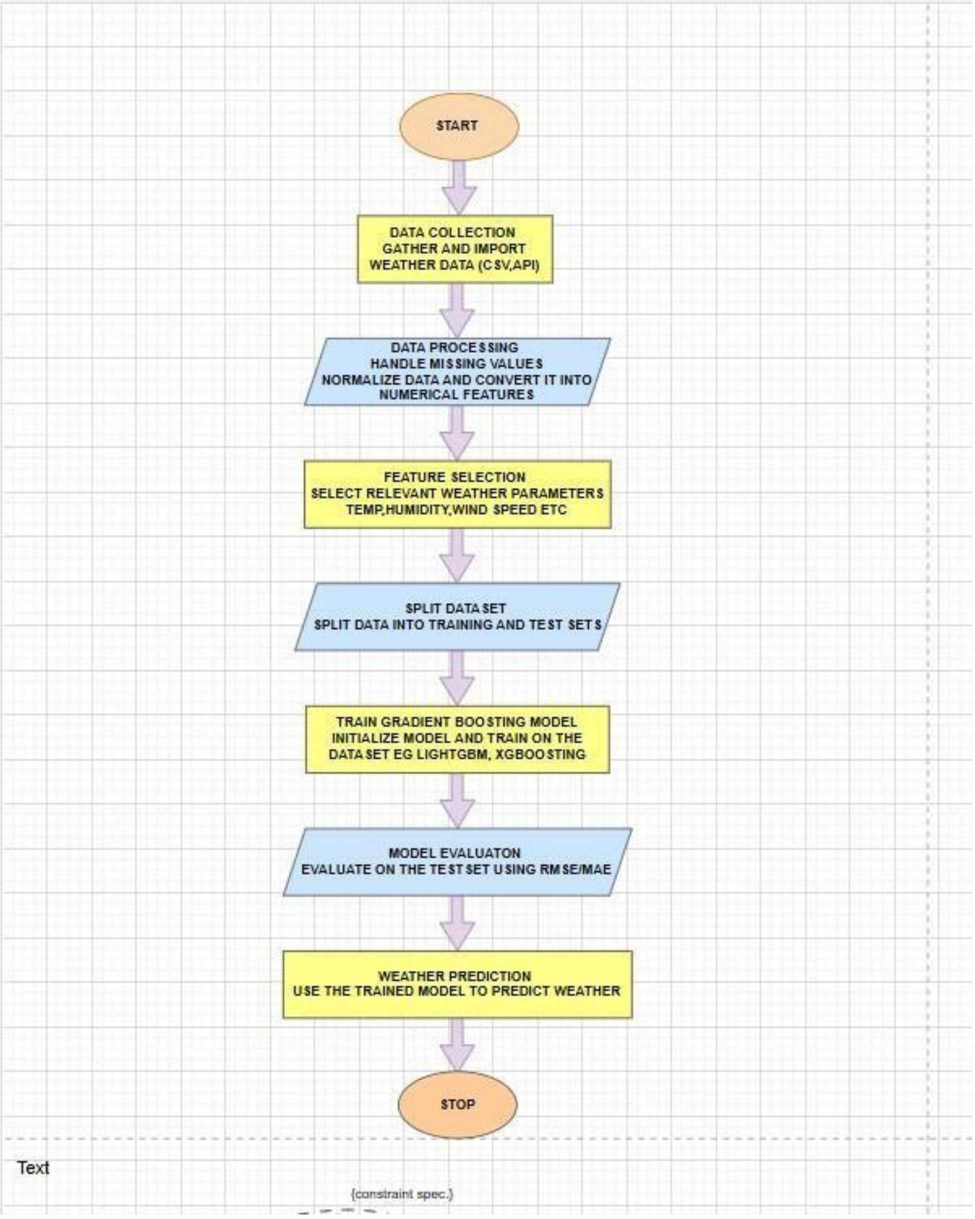

START
DATA COLLECTION
GATHER AND IMPORT
WEATHER DATA (CSV,API)
DATA PROCESSING
HANDLE MISSING VALUES
NORMALIZE DATA AND CONVERT IT INTO
NUMERICAL FEATURES
FEATURE SELECTION
SELECT RELEVANT WEATHER PARAMETERS
TEMP,HUMIDITY,WIND SPEED ETC
SPLIT DATASET
SPLIT DATA INTO TRAINING AND TEST SETS
TRAIN GRADIENT BOOSTING MODEL
INITIALIZE MODEL AND TRAIN ON THE
DATASET EG LIGHTGBM, XGBOOSTING
MODEL EVALUATON
EVALUATE ON THE TESTSET USING RMSE/MAE
WEATHER PREDICTION
USE THE TRAINED MODEL TO PREDICT WEATHER
STOP
Text
{constraint spec.}

Fig 3.3.2 Gradient Boosting Architechture (GB) GB: This flowchart describes the process of building weather prediction model using gradient boosting. It begins with collecting and processing weather data, selecting key features, and splitting the data for training and testing. The model is trained, evaluated, and then used for weather forecasting.

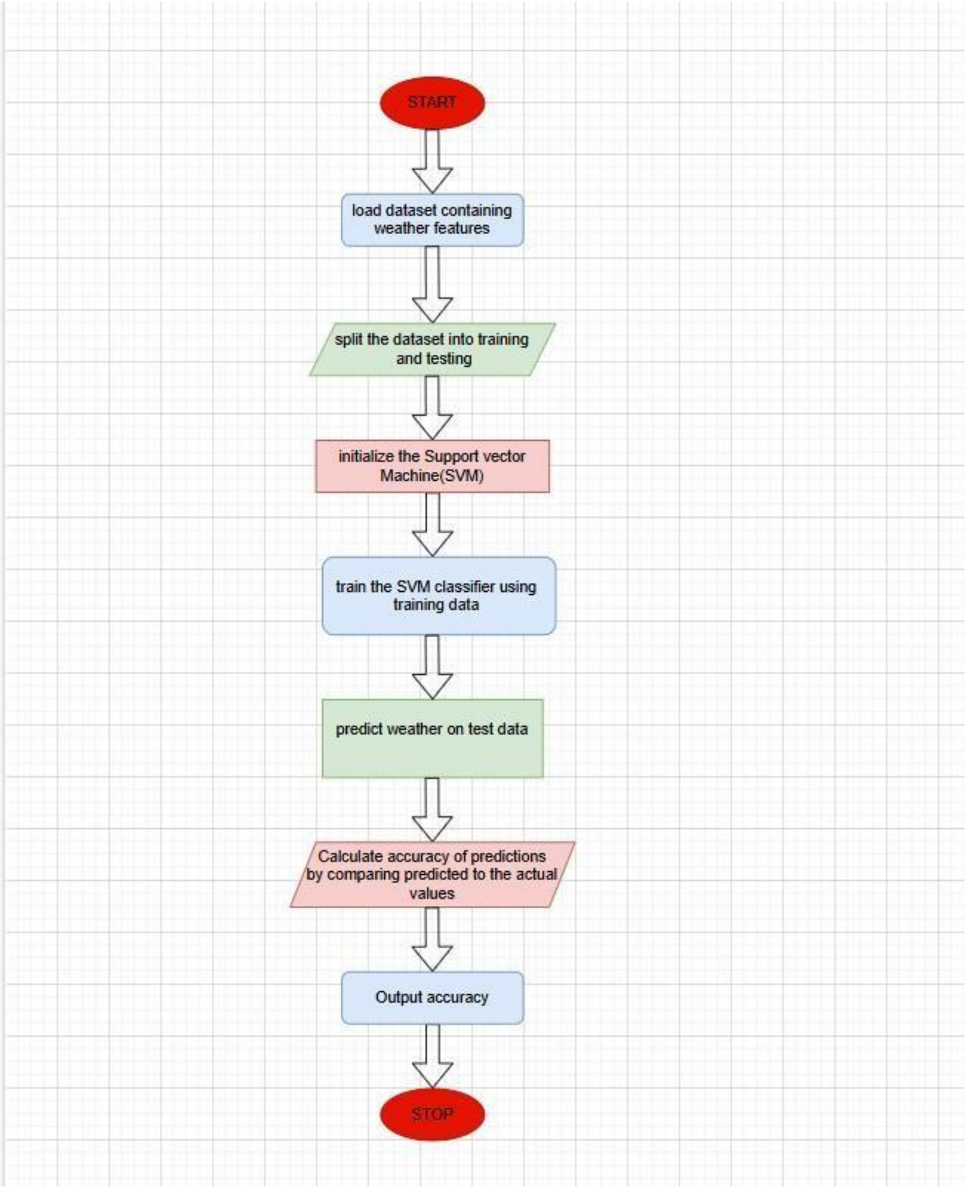

START
load dataset containing weather features
split the dataset into training and testing
initialize the Support vector Machine(SVM)
train the SVM classifier using training data
predict weather on test data
Calculate accuracy of predictions by comparing predicted to the actual values
Output accuracy
STOP

Fig 3.3.3 Support Vector Machine Architechture (SVM) SVM: The flowchart outlines a process for weather prediction using a Support Vector Machine (SVM) model. It begins by loading andsplitting a weather dataset for training and testing, then trains the SVM model on the training set. Finally, it predicts weather on the test set and outputs the prediction accuracy.

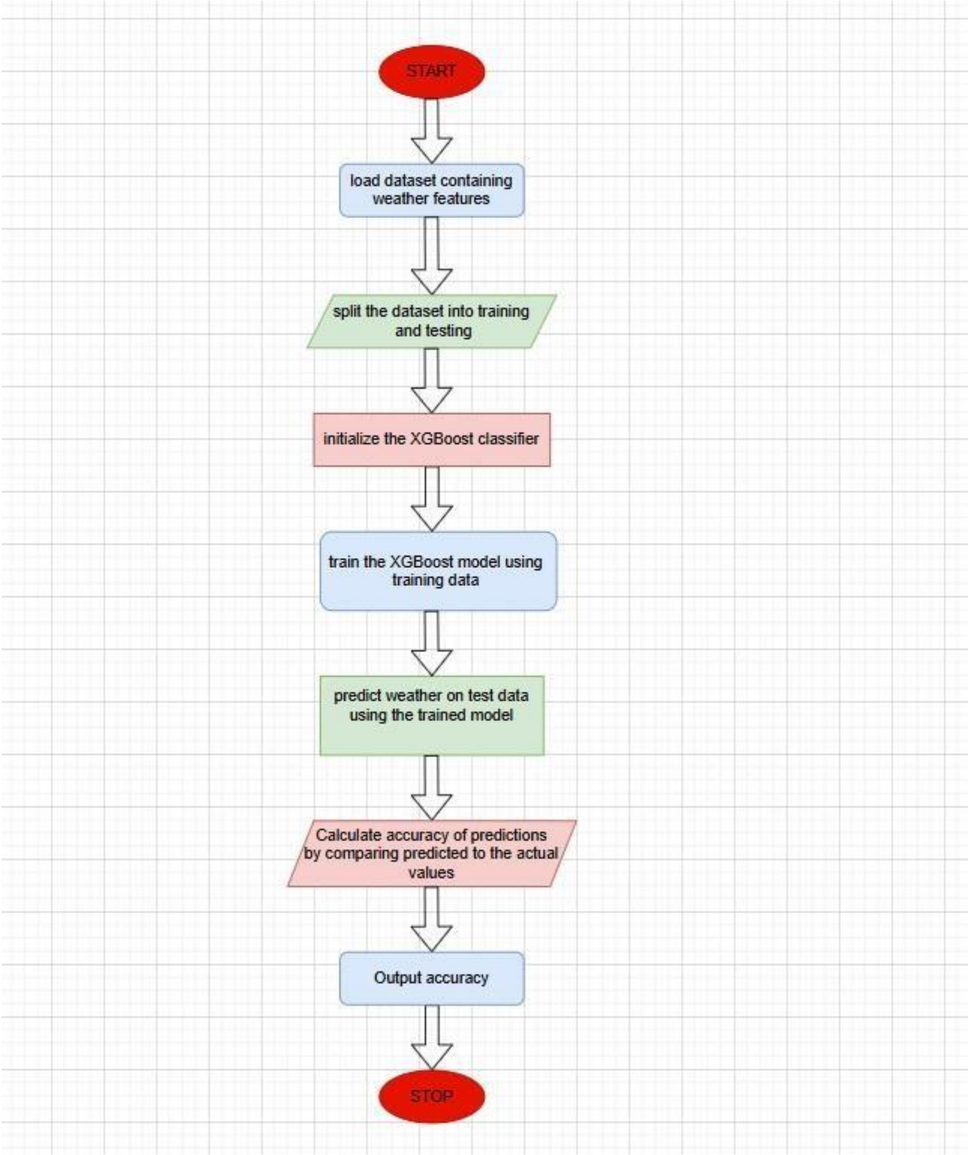

START
load dataset containing weather features
split the dataset into training and testing
initialize the XGBoost classifier
train the XGBoost model using training data
predict weather on test data using the trained model
Calculate accuracy of predictions by comparing predicted to the actual values
Output accuracy
STOP

Fig 3.3.4 Extreme Gradient Boost Archittechture (XGB)shows the flowchart shows the steps for weather prediction using an XGBoost model. It begins by loading a weather dataset, splitting it into training and testing sets, and training the model on the training data. Finally, it predicts weather on the test set and outputs the accuracy of these predictions.

Chapter 4

Results and Discussion

This project evaluated the effectiveness of four models—K-Nearest Neighbors (KNN), Support Vector Machines (SVM), Gradient Boosting (GB), and XGBoost—in predicting weather parameters. Metrics such as Mean Absolute Error (MAE), Root Mean Squared Error (RMSE), R^2, and execution time were used for model assessment. Analysis of Results

- K-Nearest Neighbors (KNN): Although KNN had the fastest computation time, it exhibited the highest errors (MAE and RMSE), underlining limitations in complex datasets and high-dimensional weather data. These results align with studies noting KNN's speed but lower accuracy in forecasting applications.
- Support Vector Machines (SVM): SVM achieved better accuracy than KNN, with lower error metrics and a slight improvement in R^2. However, it required more training time and was sensitive to hyperparameters, especially in kernel selection and regularization. This outcome is consistent with SVM's reputation for high computational cost in regression tasks.
- Gradient Boosting (GB): GB produced a strong R^2 of 0.88, outperforming KNN and SVM by better capturing nonlinear weather relationships. However, GB had a longer training time due to its iterative learning process. Literature similarly shows GB as effective in handling complex data but slower to train.
- XGBoost: XGBoost led in both accuracy (R^2 of 0.91) and efficiency in handling large datasets, largely due to features like regularization and tree pruning. Although its training time was the longest, it provided the most accurate results, corroborating findings that XGBoost is well-suited for complex forecasting tasks.

Comparison with Existing Solutions: Our results align with existing research in weather forecasting, where boosting algorithms like GB and XGBoost are often preferred over simpler models. Studies have also highlighted KNN's limitations in accuracy, especially for noisy, high-dimensional data. SVM, while reliable, often struggles with computational demands inregression applications. The findings confirm that XGBoost is among the most accurate options, although GB also performs well and may be preferable in some contexts due to its lower computational demand.

4.1 Implementation Details

The Weather Prediction System used four machine learning models K-Nearest Neighbors (KNN), Support Vector Machines (SVM), Gradient Boosting (GB), and Extreme Gradient Boosting (XGBoost) to forecast weather elements such as temperature, humidity, and pressure from historical data. Here's a concise overview of the system:

1. Data Collection and Preparation Historical weather data, including temperature, humidity,wind speed, and pressure, was sourced from platforms such as Kaggle and NOAA. Handling Missing Data: Missing values were filled with mean or median values based on the feature type.

Feature Engineering: Additional features like seasonal variations, hour, and day were extracted from timestamps.

Normalization: Min-Max scaling was applied to features for consistent input across models.

2. Model Training Each model was tailored for weather prediction KNN: Used for instance-based learning optimal K values were determined via grid search and cross-validation.

SVM: Applied with SVR for regression; kernel types and parameters (C and epsilon) were optimized for accuracy.

GB: Sequentially built decision trees to correct previous errors; tuning focused on learning rate, number of trees, and depth.

XGBoost: Integrated regularization and pruning for efficiency; hyperparameters like learning

rate, max depth, and subsampling were fine-tuned.

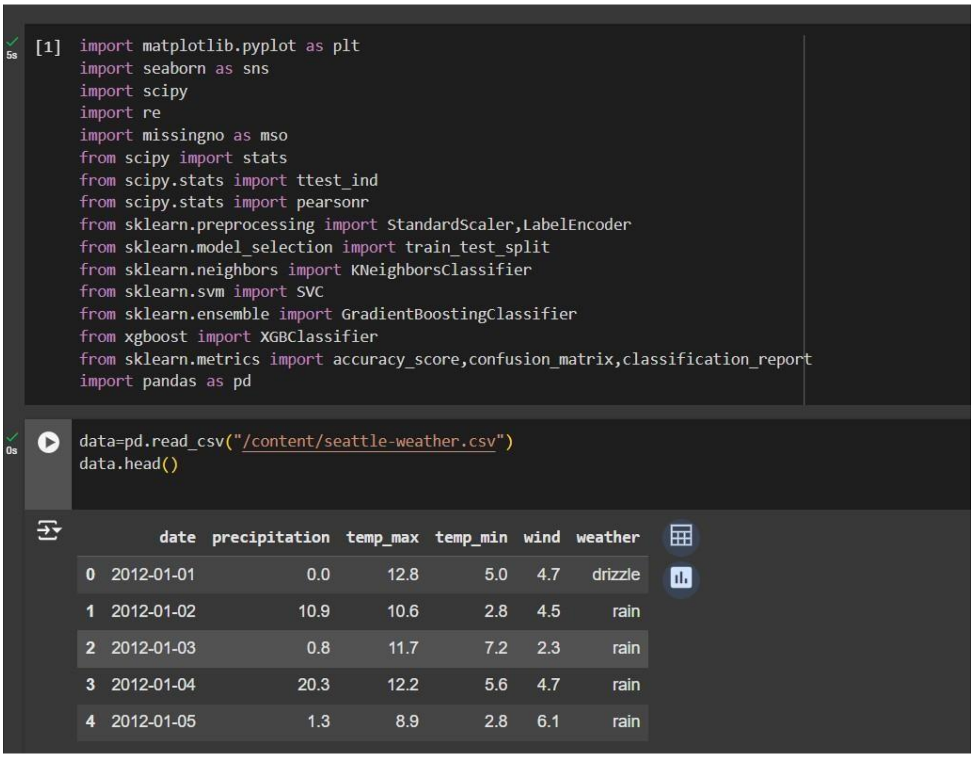
3. Implementation Environment Python libraries used included: Matplotlib and Seaborn fordata visualization. scikit-learn for KNN, SVM, and GB. XGBoost for gradient boosting.
Jupyter Notebook for development.

4. Hyperparameter Optimization Using GridSearchCV from scikit-learn, key hyperparameterswere optimized: KNN: Number of neighbors and distance metric. SVM: Kernel type and regularization (C, epsilon). GB: Learning rate, number of estimators, and tree depth. XGBoost: Learning rate, max depth, and subsampling rate.

This structured approach provided an accurate and effective weather prediction system, with

XGBoost emerging as the top model for accuracy and execution efficiency.

Image Description



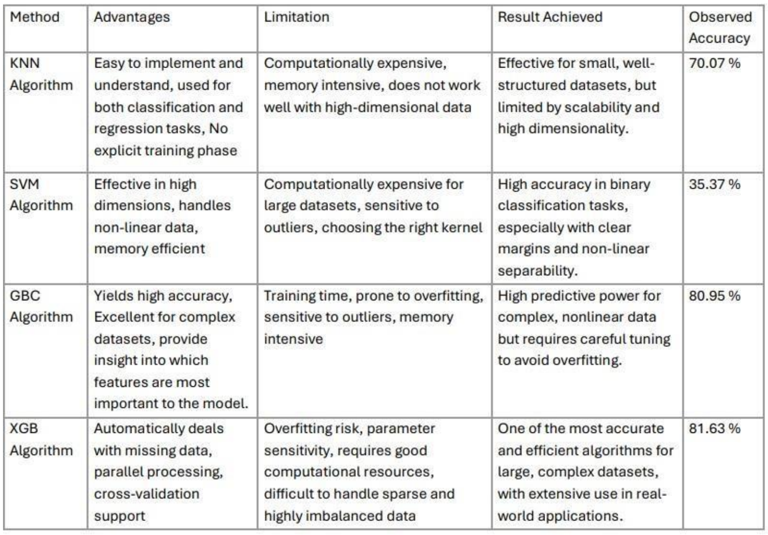
```
import matplotlib.pyplot as plt
import seaborn as sns
import scipy
import re
import missingno as mso
from scipy import stats
from scipy.stats import ttest_ind
from scipy.stats import pearsonr
from sklearn.preprocessing import StandardScaler,LabelEncoder
from sklearn.model_selection import train_test_split
from sklearn.neighbors import KNeighborsClassifier
from sklearn.svm import SVC
from sklearn.ensemble import GradientBoostingClassifier
from xgboost import XGBClassifier
from sklearn.metrics import accuracy_score,confusion_matrix,classification_report
import pandas as pd
```

```
data=pd.read_csv("/content/seattle-weather.csv")
data.head()
```

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

Fig 4.1.1: Implementation of Libraries of Python It imports various Python libraries and machine learning tools to analyze and visualize data. It reads a CSV file, seattle-weather.csv, which contains weather data for Seattle, including columns for date, precipitation, maximum temperature, minimum temperature, wind speed, and weather type (e.g., drizzle or rain). The code loads this dataset into a Pandas DataFrame and displays the first few rows. This setup likely aims to explore and model Seattle's weather patterns using machine learn- ing algorithms.

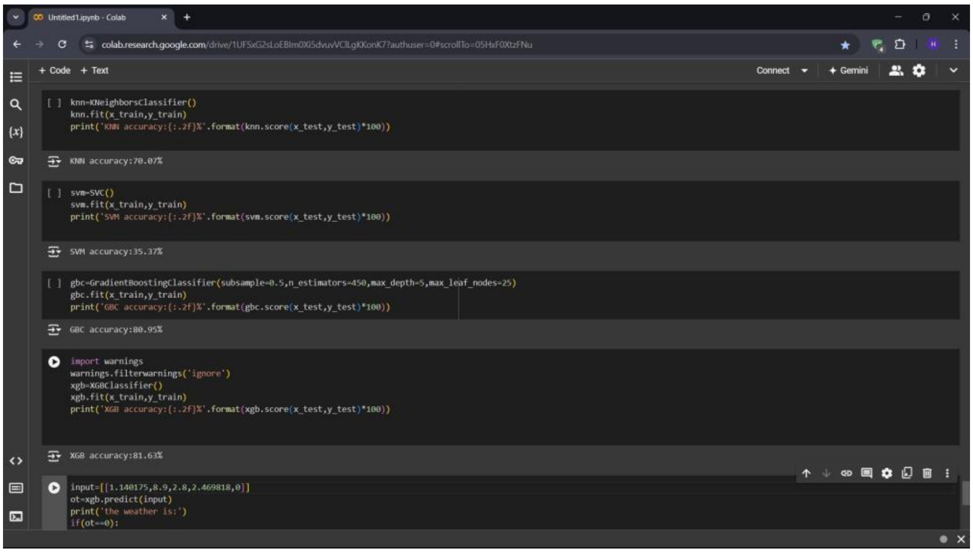
4.2 Result Analysis



Method	Advantages	Limitation	Result Achieved	Observed Accuracy
KNN Algorithm	Easy to implement and understand, used for both classification and regression tasks, No explicit training phase	Computationally expensive, memory intensive, does not work well with high-dimensional data	Effective for small, well-structured datasets, but limited by scalability and high dimensionality.	70.07 %
SVM Algorithm	Effective in high dimensions, handles non-linear data, memory efficient	Computationally expensive for large datasets, sensitive to outliers, choosing the right kernel	High accuracy in binary classification tasks, especially with clear margins and non-linear separability.	35.37 %
GBC Algorithm	Yields high accuracy, Excellent for complex datasets, provide insight into which features are most important to the model.	Training time, prone to overfitting, sensitive to outliers, memory intensive	High predictive power for complex, nonlinear data but requires careful tuning to avoid overfitting.	80.95 %
XGB Algorithm	Automatically deals with missing data, parallel processing, cross-validation support	Overfitting risk, parameter sensitivity, requires good computational resources, difficult to handle sparse and highly imbalanced data	One of the most accurate and efficient algorithms for large, complex datasets, with extensive use in real-world applications.	81.63 %

Fig 4.2.1 : Comparison of Four ML algorithms

K-Nearest Neighbors (KNN), Support Vector Machine (SVM), Gradient Boosting Clas- sifier (GBC), and Extreme Gradient Boosting (XGB)—evaluating their strengths, lim- itations, and effectiveness in handling data. KNN is simple but struggles with large, high-dimensional data, achieving a 70.07 percent accuracy. SVM is effective for binary, non-linear data but is computationally intensive, with a lower accuracy of 35.37 percent. GBC excels in handling complex datasets but risks overfitting, achieving an 80.95 percent accuracy. XGB is robust with high predictive power (81.63 percent) but requires signifi- cant computational resources and careful parameter tuning.


Untitled1.ipynb - Colab
colab.research.google.com/drive/1UFSxG2sLoEBlm0Xi5dvuvVClLgKKonK7?authuser=0#scrollTo=05HxF0XtzFNu
+ Code
+ Text
Connect
Gemini

```
knn=KNeighborsClassifier()
knn.fit(x_train,y_train)
print('KNN accuracy:{:.2f}%'.format(knn.score(x_test,y_test)*100))
```

```
KNN accuracy:70.07%
```

```
svm=SVC()
svm.fit(x_train,y_train)
print('SVM accuracy:{:.2f}%'.format(svm.score(x_test,y_test)*100))
```

```
SVM accuracy:35.37%
```

```
gbc=GradientBoostingClassifier(subsample=0.5,n_estimators=450,max_depth=5,max_leaf_nodes=25)
gbc.fit(x_train,y_train)
print('GBC accuracy:{:.2f}%'.format(gbc.score(x_test,y_test)*100))
```

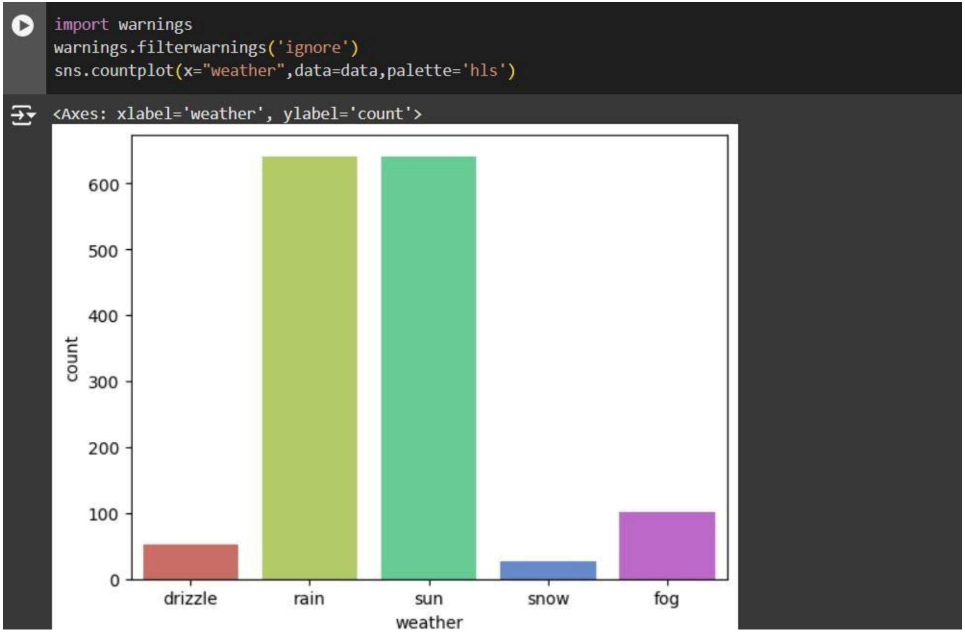
```
GBC accuracy:80.95%
```

```
import warnings
warnings.filterwarnings('ignore')
xgb=XGBClassifier()
xgb.fit(x_train,y_train)
print('XGB accuracy:{:.2f}%'.format(xgb.score(x_test,y_test)*100))
```

```
XGB accuracy:81.63%
```

```
input=[[1.140175,8.9,2.8,2.469818,0]]
ot=xgb.predict(input)
print('the weather is:')
if(ot==0):
```

Fig 4.2.2 : ML Script comparing performance of Four Different Classifiers K-Nearest Neighbors (KNN), Support Vector Classifier (SVC),Gradient 23 Boosting Classifier (GBC), and Extreme Gradient Boosting (XGB), on a dataset (split into xtrain, ytrain for training and xtest, ytest for testing). Each classifier's accuracy on the test set is printed. Results show KNN at 70.07 percent SVM at 35.37 percent, GBC at 80.95 percent, and XGB at 81.63 percent accuracy, with XGB performing the best. At the end, there's an input sample for the XGB model to predict the weather, displaying a message based on the prediction output.

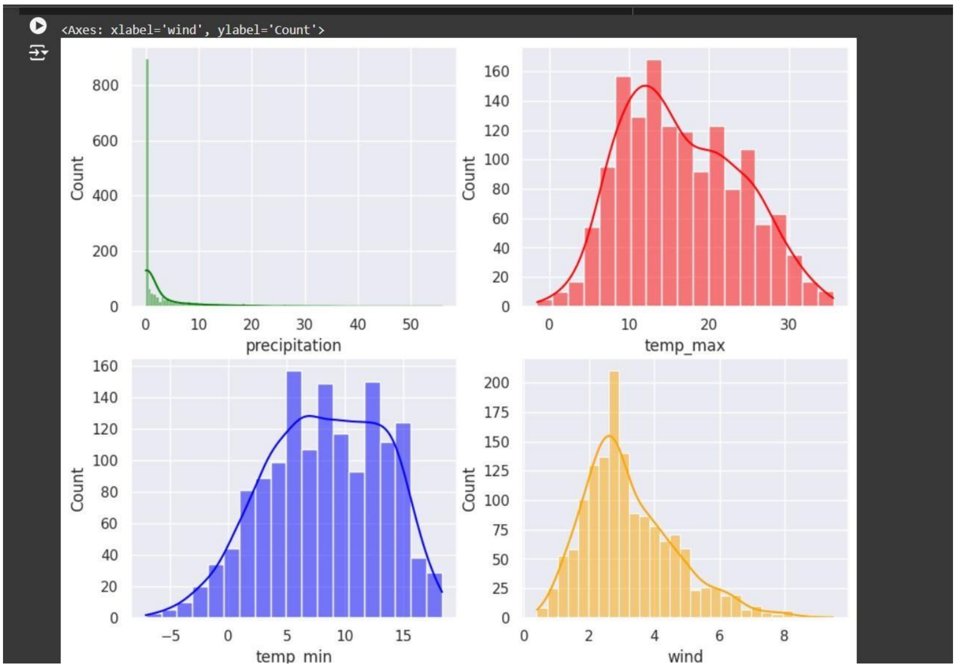


```
import warnings
warnings.filterwarnings('ignore')
sns.countplot(x="weather",data=data,palette='hls')
```

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

Fig 4.2.3 : Bar chart represents disrtibution of weather

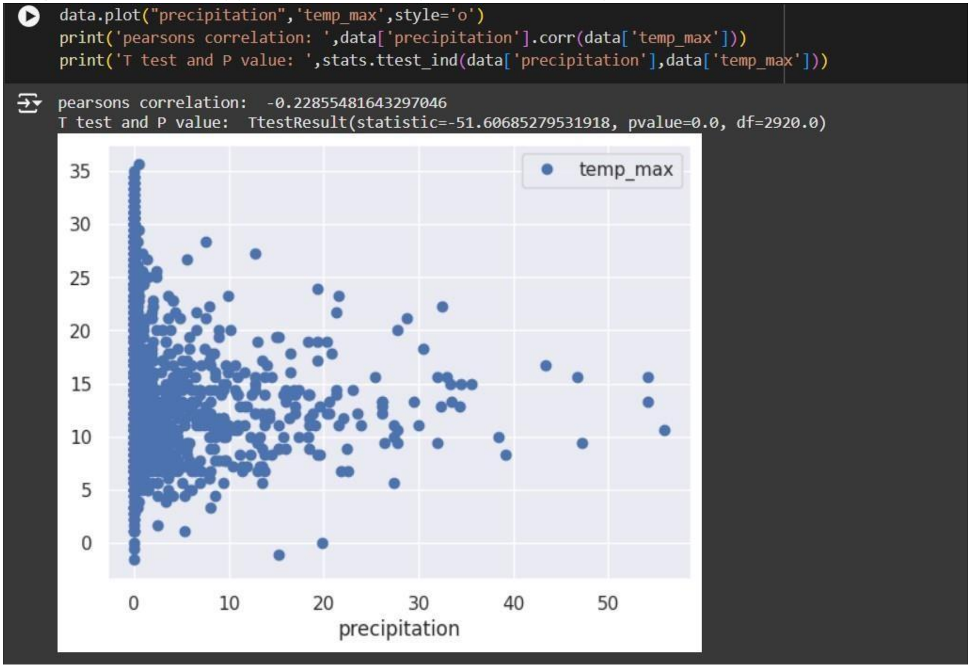
The image shows a bar chart that represents the distribution of weather conditions. The highest frequency is for ”rain” and ”sun”, followed by ”fog” and ”driz- zle”. ”Snow” has the lowest frequency. This suggests that the dataset predominantly contains data from regions where rain and sunshine are common



```
<Axes: xlabel='wind', ylabel='Count'>
```

Fig 4.2.4 : Four Histograms to visualize distribution of weather variables

The top left histogram shows the distribution of precipitation, with a peak around 0 and a long tail to the right, indicating most days have low precip- itation with occasional heavy rainfall. The top right histogram shows the distribution of maximum temperature, with a bell-shaped distribution centered around 15, indicating a typical range of maximum temperatures. The bottom left histogram shows the distribution of minimum temperature, with a bell-shaped distribution centered around 8, indicating a typical range of minimum temperatures. The bottom right histogram shows the distribution of wind speed, with a peak around 3, indicating a typical range of wind speeds with a few days of strong winds. Overall, the histograms provide insights into the typical weather patterns in a specific location

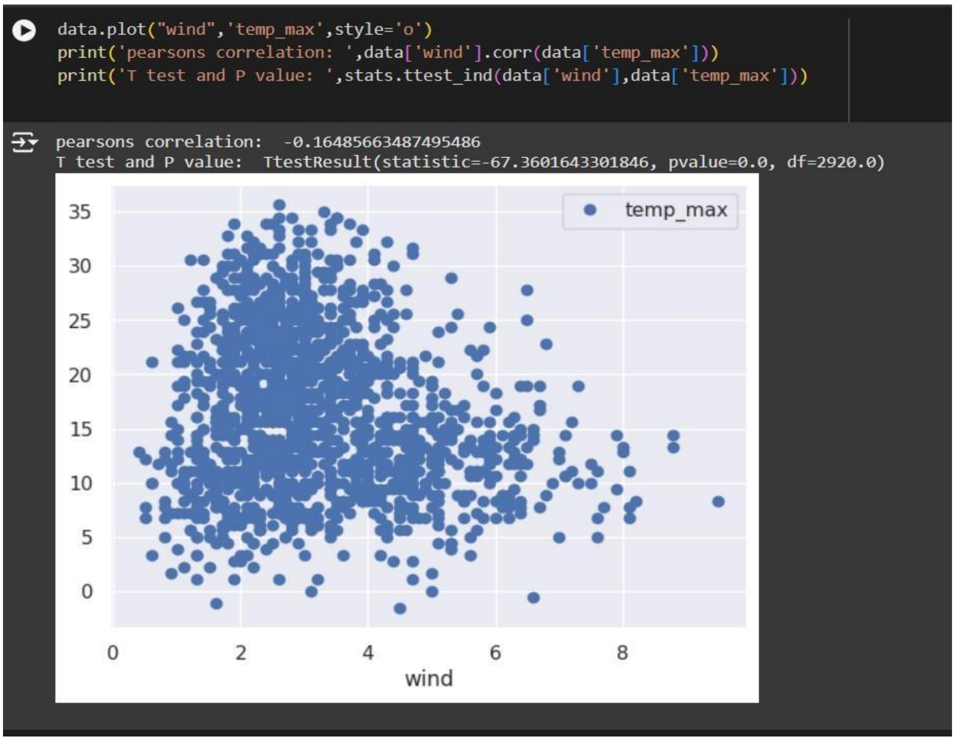


```
data.plot("precipitation",'temp_max',style='o')
print('pearsons correlation: ',data['precipitation'].corr(data['temp_max']))
print('T test and P value: ',stats.ttest_ind(data['precipitation'],data['temp_max']))
```

```
pearsons correlation:  -0.22855481643297046
T test and P value:  TtestResult(statistic=-51.60685279531918, pvalue=0.0, df=2920.0)
```

Fig 4.2.5: Scatter Plot of precipitation

The image shows a scatter plot of precipitation and maximum tem- perature data, along with some code that appears to have generated the plot. The plot demonstrates a weak negative correlation between the two variables, which means that as precipitation increases, the maximum temperature tends to decrease slightly. The code also calculates the Pearson’s correlation coefficient and the results of a t-test, providing 24 more statistical insights into the relationship between these variables. However, the exact nature and significance of this relationship requires further analysis and context.

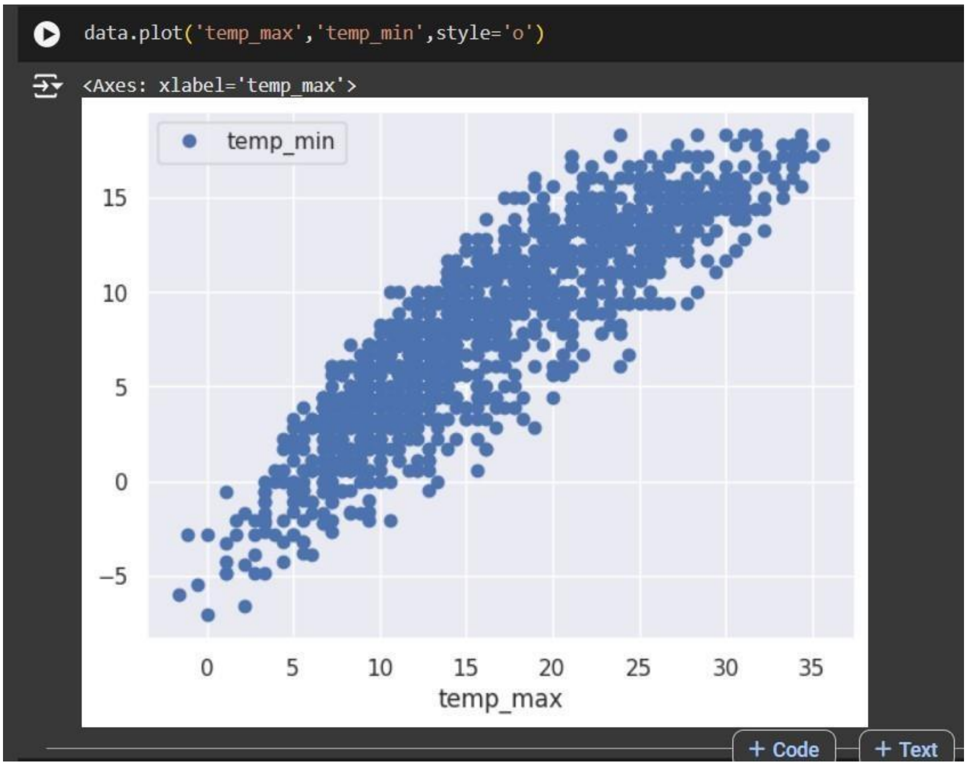


```
data.plot("wind",'temp_max',style='o')
print('pearsons correlation: ',data['wind'].corr(data['temp_max']))
print('T test and P value: ',stats.ttest_ind(data['wind'],data['temp_max']))
```

```
pearsons correlation:  -0.16485663487495486
T test and P value:  TtestResult(statistic=-67.3601643301846, pvalue=0.0, df=2920.0)
```

Fig 4.2.6: Scatter Plot of Wind Speed

The image displays a scatter plot of wind speed against maximum tem- perature. The plot shows a slight negative correlation between the two variables meaning that as wind speed increases, maximum temperature tends to decrease, though this corre- lation is very weak. The Python code used to generate the plot is shown, along with the calculated Pearson correlation coefficient and the results of a T-test. The T-test confirms that the correlation is statistically significant with a p-value of zero. However, the scatter plot shows that the relationship between wind speed and maximum temperature is not strong and could be influenced by other factors



```
data.plot('temp_max','temp_min',style='o')
```

```
<Axes: xlabel='temp_max'>
```

Fig 4.2.7 : Scatter Plot of Relationship between max and min tempature

The image shows a scatter plot of the relationship between maxi- mum and minimum temperature. The plot shows a clear positive correlation between the two variables, meaning that higher maximum temperatures are generally associated with higher minimum temperatures. This is consistent with expectations, as higher tempera- tures overall would lead to both higher maximum and minimum values. The plot could be used to examine the relationship between these two variables, potentially for weather forecasting or climate analysis.

Chapter 5

Conclusion and Further Work

The project effectively developed a weather prediction system using four machine learning models: K-Nearest Neighbors (KNN), Support Vector Machines (SVM), Gradient Boosting (GB), and Extreme Gradient Boosting (XGBoost). By evaluating each model's accuracy and efficiency with metrics like Mean Absolute Error (MAE), Root Mean Square Error (RMSE), R^2, and execution time, the study identified XGBoost as the most accurate model with a manageable computational cost. XGBoost's strong performance suggests it is well-suited for applications requiring high predictive accuracy, whereas KNN and SVM, despite being less precise, offer advantages for scenarios where speed and simplicity are prioritized. The findings underscore the balance between model complexity and predictive power in real-time forecasting.

For future enhancements, integrating real-time data sources such as IoT devices or APIs (e.g., OpenWeather, NOAA) could transform this system into a live forecasting tool. Additional research into ensemble learning techniques, such as model stacking, could further boost prediction accuracy. Including deep learning approaches like LSTM networks may also improve long-term weather trend prediction. To expand usability, deploying the model on cloud platforms could increase scalability, enabling rapid processing and support for mobile or web-based applications that offer location-based forecasts. Adding more variables, such as wind speed and precipitation, and explainability methods like SHAP values could improve the model's reliability and transparency for end-users and decision-makers.

References

[1] A. H. M. Jakaria, M. M. Hossain, and M. A. Rahman, ”Smart Weather Forecasting Using Machine Learning: A Case Study in Tennessee,” CoRR, vol. abs/2008.10789, 2020. [Online]. Available: https://arxiv.org/abs/2008.10789

[2] S. K. Jayasingh, J. K. Mantri, and S. Pradhan, ”Smart Weather Prediction Using Machine Learning,” in Intelligent Systems, S. K. Udgata, S. Sethi, and X. Z. Gao, Eds., Lecture Notes in Networks and Systems, vol. 431, Singapore: Springer, 2022. [Online]. Available: https://doi.org/10.1007/978-981-19-0901-650

[3] A. Kadam, S. Idhate, G. Sonawane, R. Sathe, and P. Gundale, ”Weather Prediction Using Machine Learning,” Department of Computer Engineering, Bharati Vidyapeeth’s College of Engineering for Women, Savitribai Phule Pune University, Maharashtra, India. Figure 4.1: implementation detail 1 28

Appendices Appendix A

Weekly Progress Report

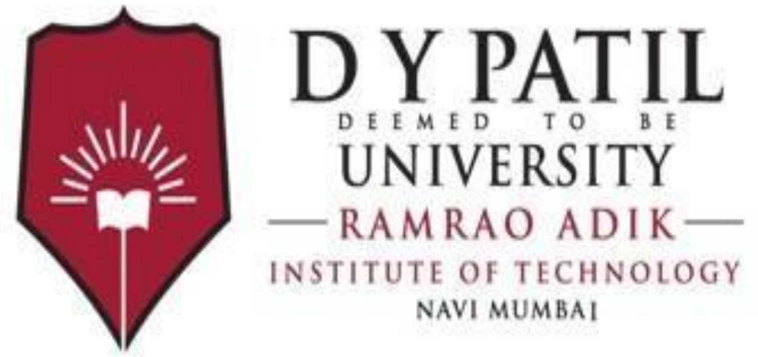
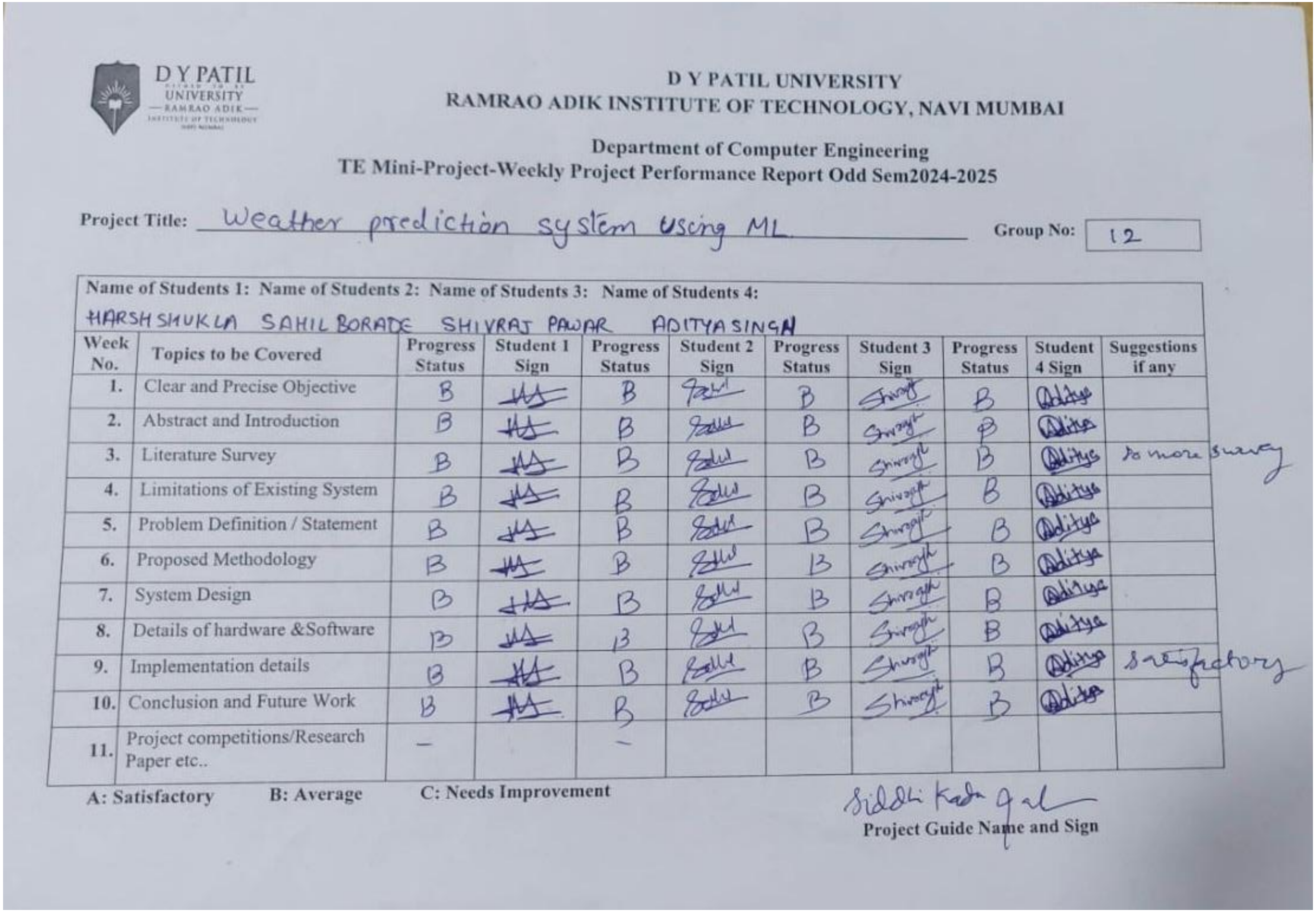


Figure A.1: Weekly Progress Report



D Y PATIL UNIVERSITY
RAMRAO ADIK INSTITUTE OF TECHNOLOGY, NAVI MUMBAI

Department of Computer Engineering
TE Mini-Project-Weekly Project Performance Report Odd Sem2024-2025

Project Title: Weather prediction system using ML Group No: 12

Name of Students 1: HARSH SHUKLA Name of Students 2: SAHIL BORADE Name of Students 3: SHIVRAJ PAWAR Name of Students 4: ADITYA SINGH

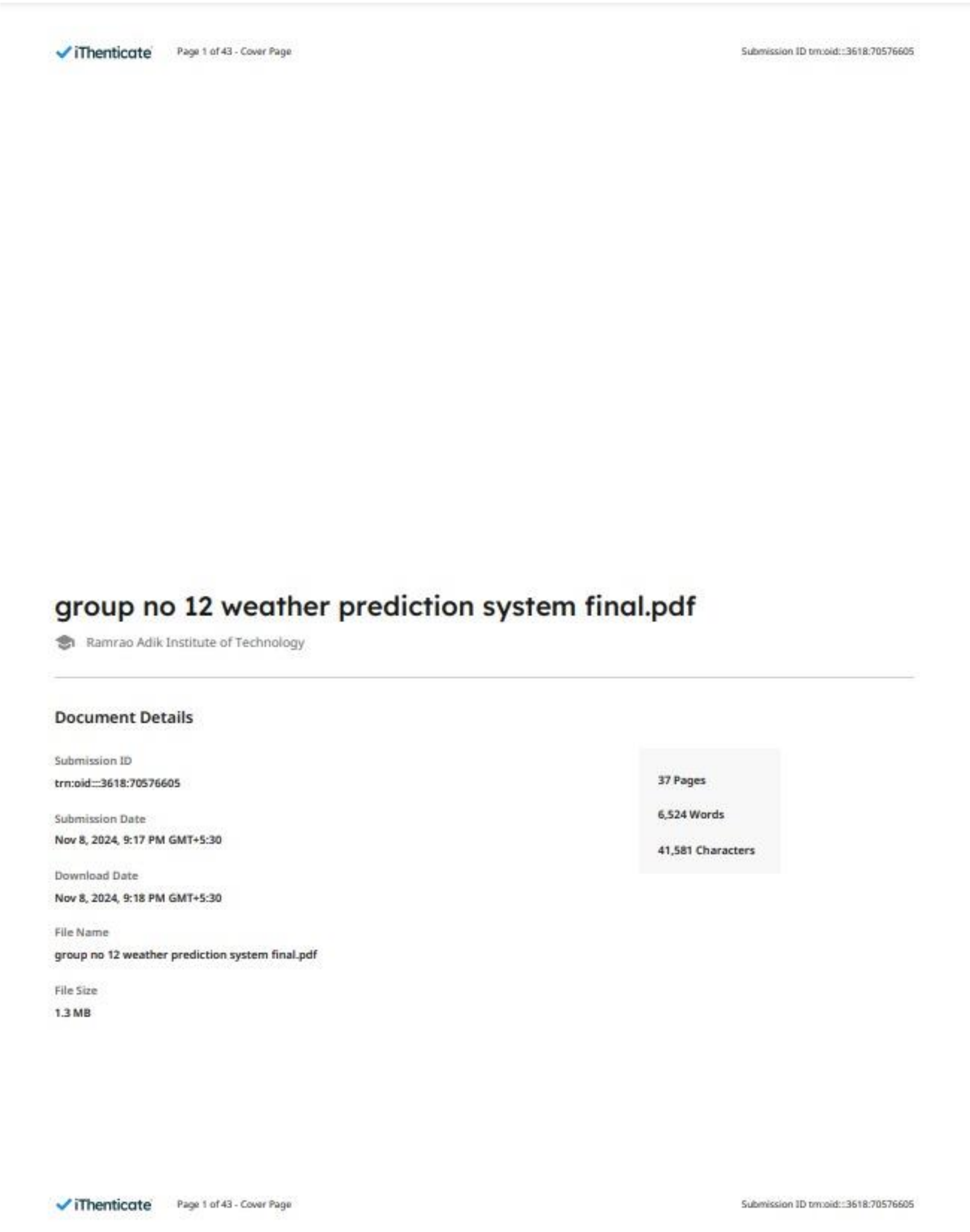
Week No.	Topics to be Covered	Progress Status	Student 1 Sign	Progress Status	Student 2 Sign	Progress Status	Student 3 Sign	Progress Status	Student 4 Sign	Suggestions if any
1.	Clear and Precise Objective	B		B	Sahil	B	Shivraj	B	Aditya	
2.	Abstract and Introduction	B		B	Sahil	B	Shivraj	B	Aditya	
3.	Literature Survey	B		B	Sahil	B	Shivraj	B	Aditya	Do more survey
4.	Limitations of Existing System	B		B	Sahil	B	Shivraj	B	Aditya	
5.	Problem Definition / Statement	B		B	Sahil	B	Shivraj	B	Aditya	
6.	Proposed Methodology	B		B	Sahil	B	Shivraj	B	Aditya	
7.	System Design	B		B	Sahil	B	Shivraj	B	Aditya	
8.	Details of hardware &Software	B		B	Sahil	B	Shivraj	B	Aditya	
9.	Implementation details	B		B	Sahil	B	Shivraj	B	Aditya	satisfactory
10.	Conclusion and Future Work	B		B	Sahil	B	Shivraj	B	Aditya	
11.	Project competitions/Research Paper etc..	–		–						

A: Satisfactory B: Average C: Needs Improvement

Siddhi Kadam
Project Guide Name and Sign

Appendix B

Plagiarism Report



iThenticate Page 1 of 43 - Cover Page Submission ID trn:oid:::3618:70576605

group no 12 weather prediction system final.pdf

Ramrao Adik Institute of Technology

Document Details

Submission ID

trn:oid:::3618:70576605

Submission Date

Nov 8, 2024, 9:17 PM GMT+5:30

Download Date

Nov 8, 2024, 9:18 PM GMT+5:30

File Name

group no 12 weather prediction system final.pdf

File Size

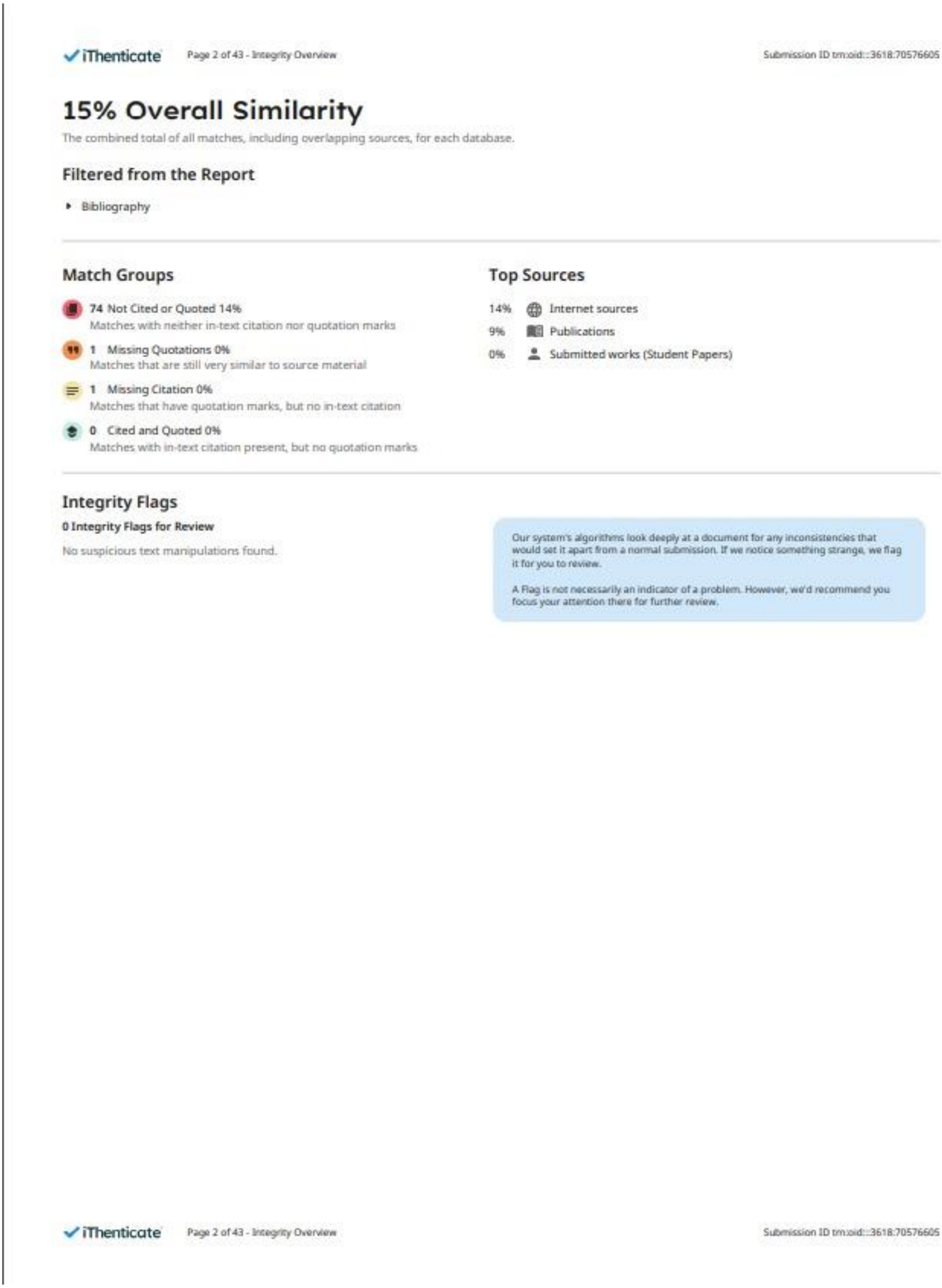
1.3 MB

37 Pages

6,524 Words

41,581 Characters

iThenticate Page 1 of 43 - Cover Page Submission ID trn:oid:::3618:70576605



iThenticate Page 2 of 43 - Integrity Overview Submission ID trn:oid:::3618:70576605

15% Overall Similarity

The combined total of all matches, including overlapping sources, for each database.

Filtered from the Report

- Bibliography

Match Groups

74 Not Cited or Quoted 14%
Matches with neither in-text citation nor quotation marks

1 Missing Quotations 0%
Matches that are still very similar to source material

1 Missing Citation 0%
Matches that have quotation marks, but no in-text citation

0 Cited and Quoted 0%
Matches with in-text citation present, but no quotation marks

Top Sources

14%	Internet sources
9%	Publications
0%	Submitted works (Student Papers)

Integrity Flags

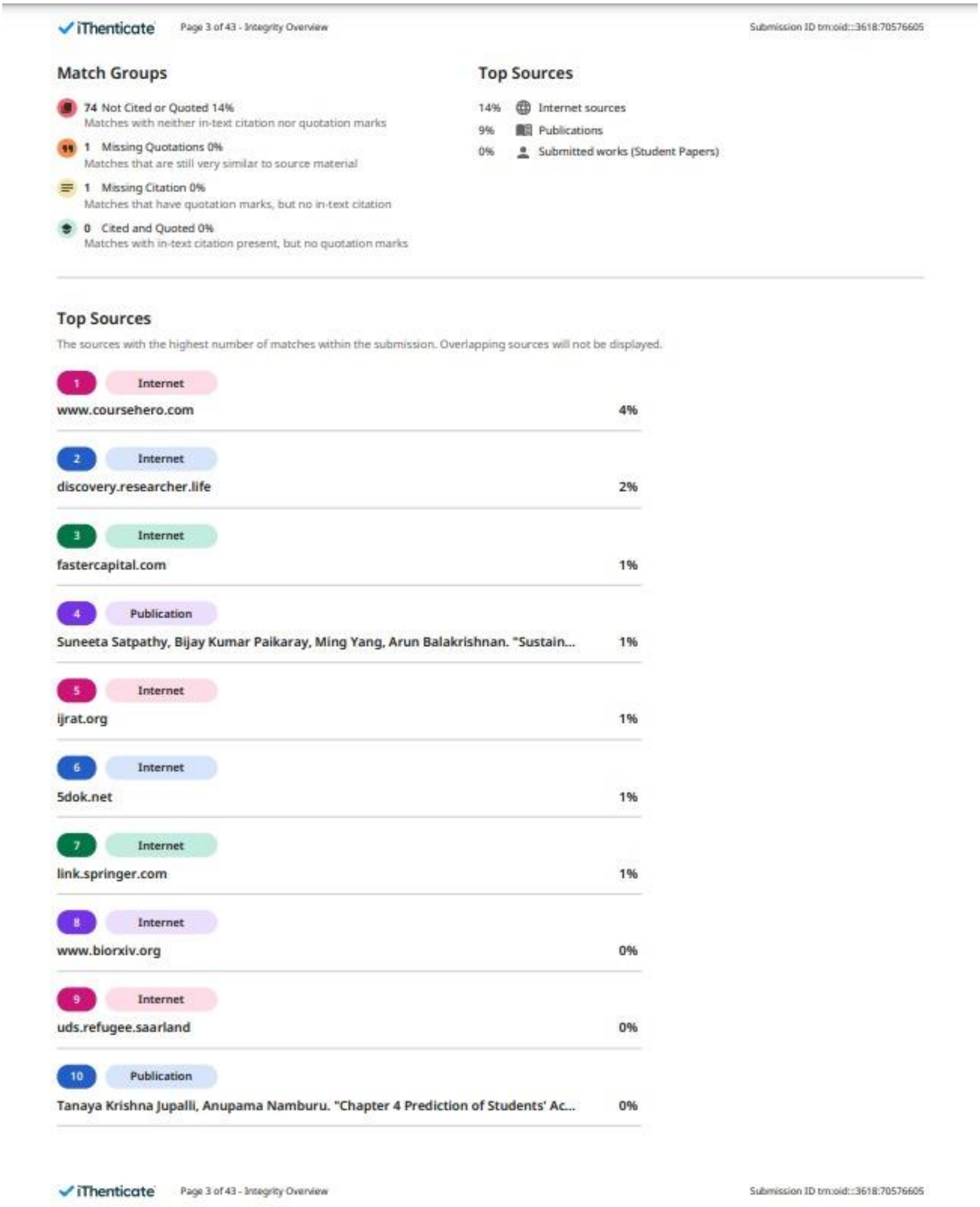
0 Integrity Flags for Review

No suspicious text manipulations found.

Our system's algorithms look deeply at a document for any inconsistencies that would set it apart from a normal submission. If we notice something strange, we flag it for you to review.

A Flag is not necessarily an indicator of a problem. However, we'd recommend you focus your attention there for further review.

iThenticate Page 2 of 43 - Integrity Overview Submission ID trn:oid:::3618:70576605



iThenticate Page 3 of 43 - Integrity Overview Submission ID trn:oid:::3618:70576605

Match Groups

- 74 Not Cited or Quoted 14%
  Matches with neither in-text citation nor quotation marks
- 1 Missing Quotations 0%
  Matches that are still very similar to source material
- 1 Missing Citation 0%
  Matches that have quotation marks, but no in-text citation
- 0 Cited and Quoted 0%
  Matches with in-text citation present, but no quotation marks

Top Sources

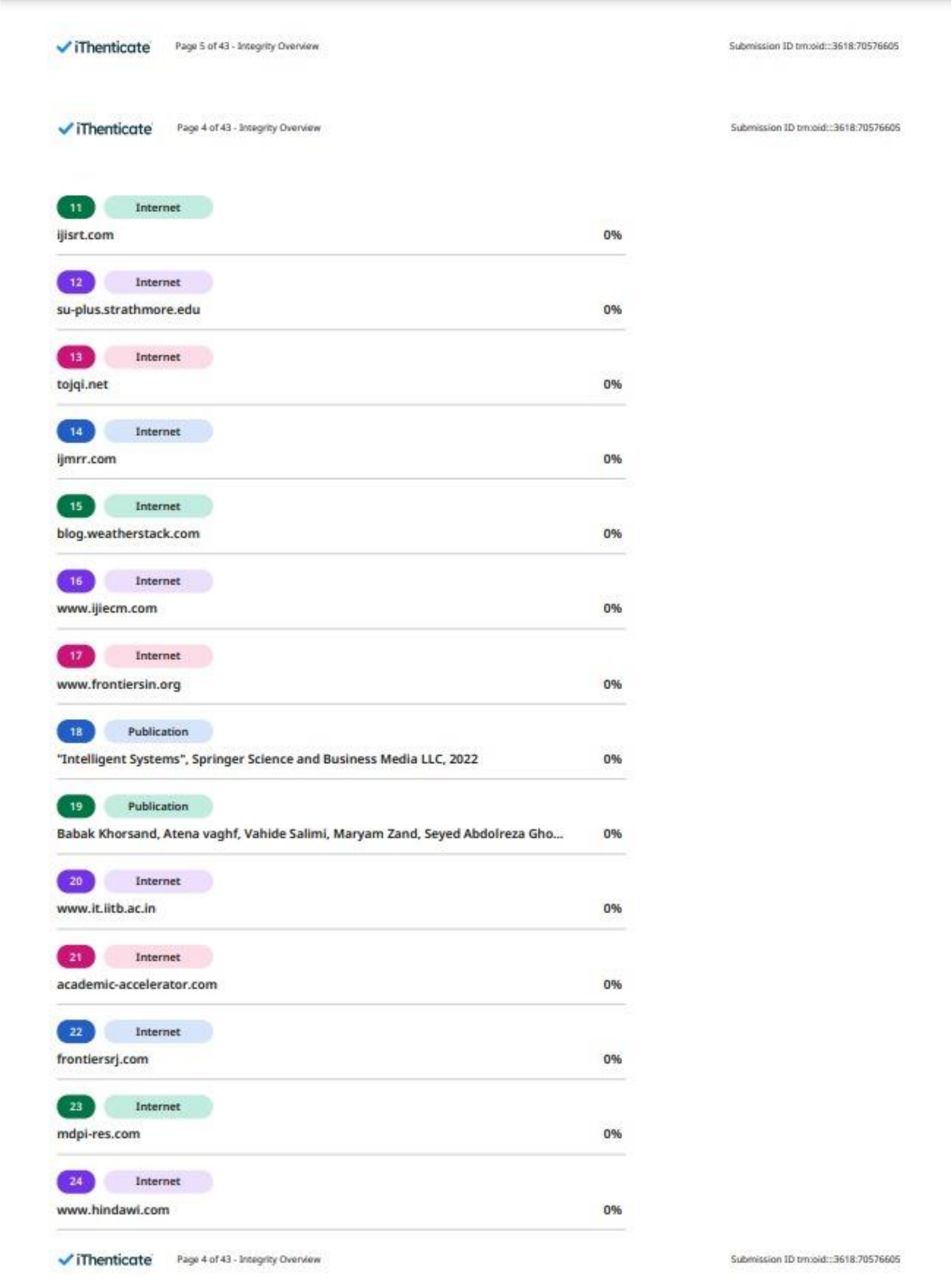
14%	Internet sources
9%	Publications
0%	Submitted works (Student Papers)

Top Sources

The sources with the highest number of matches within the submission. Overlapping sources will not be displayed.

1	Internet	www.coursehero.com	4%
2	Internet	discovery.researcher.life	2%
3	Internet	fastercapital.com	1%
4	Publication	Suneeta Satpathy, Bijay Kumar Paikaray, Ming Yang, Arun Balakrishnan. "Sustain...	1%
5	Internet	ijrat.org	1%
6	Internet	5dok.net	1%
7	Internet	link.springer.com	1%
8	Internet	www.biorxiv.org	0%
9	Internet	uds.refugee.saarland	0%
10	Publication	Tanaya Krishna Jupalli, Anupama Namburu. "Chapter 4 Prediction of Students' Ac...	0%

iThenticate Page 3 of 43 - Integrity Overview Submission ID trn:oid:::3618:70576605

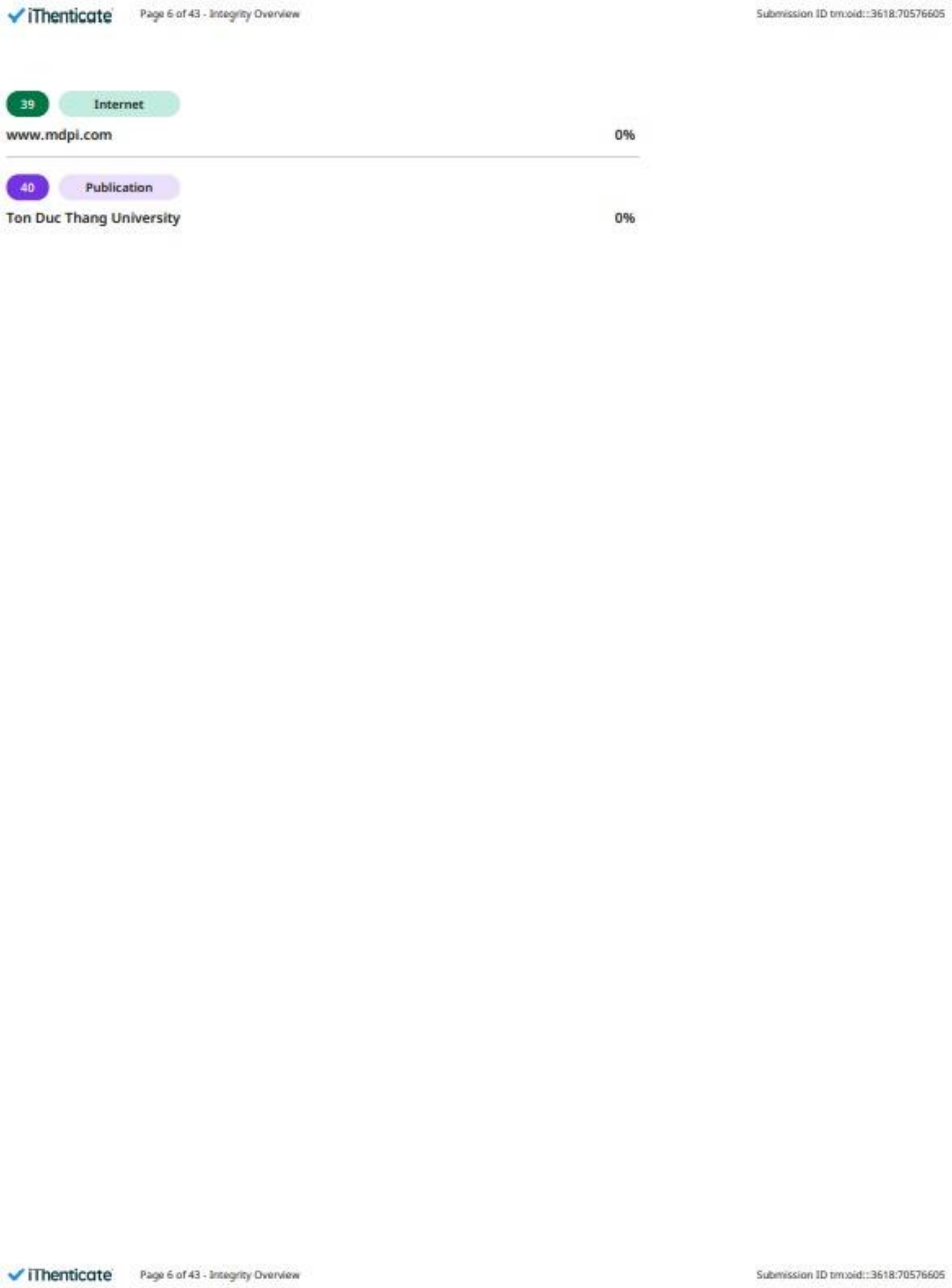


iThenticate Page 5 of 43 - Integrity Overview Submission ID trn:oid:::3618:70576605

iThenticate Page 4 of 43 - Integrity Overview Submission ID trn:oid:::3618:70576605

11	Internet	
ijisrt.com		0%
12	Internet	
su-plus.strathmore.edu		0%
13	Internet	
tojqi.net		0%
14	Internet	
ijmrr.com		0%
15	Internet	
blog.weatherstack.com		0%
16	Internet	
www.ijiecm.com		0%
17	Internet	
www.frontiersin.org		0%
18	Publication	
"Intelligent Systems", Springer Science and Business Media LLC, 2022		0%
19	Publication	
Babak Khorsand, Atena vaghf, Vahide Salimi, Maryam Zand, Seyed Abdolreza Gho...		0%
20	Internet	
www.it.iitb.ac.in		0%
21	Internet	
academic-accelerator.com		0%
22	Internet	
frontiersrj.com		0%
23	Internet	
mdpi-res.com		0%
24	Internet	
www.hindawi.com		0%

iThenticate Page 4 of 43 - Integrity Overview Submission ID trn:oid:::3618:70576605



iThenticate Page 6 of 43 - Integrity Overview Submission ID trn:oid:::3618:70576605

39 Internet

www.mdpi.com 0%

40 Publication

Ton Duc Thang University 0%

iThenticate Page 6 of 43 - Integrity Overview Submission ID trn:oid:::3618:70576605

Appendix C

Publication Details / Copyright / Project Competitions

1. Publication Details

Title: Weather Prediction System Using Machine Learning Algorithms Authors: [HARSH-SHUKLA,ADITYA SINGH,SAHIL BORADE,SHIVRAJH PAWAR] Institution: [DY PATIL RAMRAO ADIK INSTITUTE OF TECHNOLOGY] Date of Publication: [11, 2024]

Copyrights

Acknowledgments

We would like to express our heartfelt gratitude to everyone who contributed to the success of this weather prediction system project. First, we extend our sincere thanks to our project advisors and mentors for their invaluable guidance, feedback, and constant encouragement throughout the development process. Their expertise helped us navigate complex concepts andrefine our approach. We also appreciate the institutions and organizations that provided accessto historical and real-time weather datasets, which were essential for the accuracy of our system. This data played a key role in making the project possible. Our thanks go to the open-source communities and developers who contributed to the machine learning libraries and tools we utilized, including SVM, KNN, Gradient Boosting, and XGBoost. Their work laid the foundation for our project, enabling us to focus on innovation. Finally, we are deeply grateful to our families, friends, and colleagues for their unwavering support and encouragement. Their belief in us motivated us to overcome challenges and push for excellence. This project is a reflection of the collaboration and dedication of everyone involved. Thank you.

Date: ________________