HANOI UNIVERSITY OF SCIENCE AND TECHNOLOGY

Machine Learning

Weather Forecasting

Phan Vĩnh Đăng - 20214955

Trịnh Thái Dương - 20214954

Hoàng Sơn Tùng - 20214979

Nguyễn Hoàng Anh - 20214945

Nguyễn Quốc Trung - 20214976

Supervisor:	Prof. Thân Quang Khoát
Department:	Computer Science
School:	School of Information and Communications Technology

HANOI, 07/2023

ABSTRACT

Weather forecasting plays a crucial role in various domains, such as agriculture, transportation, and disaster management. Accurate predictions of weather conditions can help individuals and organizations make informed decisions, mitigate risks, and optimize resource allocation. Traditional weather forecasting methods rely on physical models and historical data analysis. However, these methods often face challenges in capturing complex and nonlinear relationships in weather patterns. Binary classification and regression techniques have been applied in this problem, with different approaches. In this work, we predict the weather in the 3 hour, the dataset consists of 8512 records. The experimental results for both SVM, RF and RidgeRegression algorithms applied to the dataset showed that the accuracy in the case of classification achieves an accurate prediction reach to approximately 90%, while in regression, the number is even higher.

TABLE OF CONTENTS

CHAPTER 1. INTRODUCTION 1

1.1 Motivation.. 1

1.2 Objectives 1

1.3 Organization... 1

CHAPTER 2. LITERATURE REVIEW 3

2.1 Algorithms 3

2.2 Preprocessing Methods 4

2.3 Evaluation Metrics 4

CHAPTER 3. DATA ANALYSIS 6

3.1 Diversity 6

3.2 Features.. 6

3.2.1 Feature Distribution 6

3.2.2 Feature Correlation 7

3.2.3 Feature Importance.. 8

CHAPTER 4. METHODOLOGY 10

4.1 Random Forest 10

4.2 SVM (Support Vector Machine)... 20

4.3 Ridge Regression 22

CHAPTER 5. CONCLUSION.. 25

CHAPTER 6. REFERENCES.. 26

CHAPTER 1

INTRODUCTION

1.1 Motivation

Weather forecasting has traditionally relied on numerical weather prediction models that utilize complex mathematical equations to simulate atmospheric behavior. However, these models often struggle with short-term predictions due to the inherent uncertainties and chaotic nature of weather systems. Machine learning offers an alternative approach by leveraging historical weather data to learn patterns and make predictions based on observed relationships between weather features and future weather conditions. This research focuses on developing a machine learning model specifically tailored for short-term weather forecasting, with a prediction horizon of three hours.

1.2 Objectives

The main objective of this research is to develop a machine learning-based model that accurately predicts weather conditions three hours ahead. The model will utilize various weather features, such as temperature, humidity, wind speed, and atmospheric pressure, etc... as input variables. By training the model on historical weather data, it will learn the patterns and relationships between these features and the corresponding weather conditions, enabling it to make accurate predictions for the future.

1.3 Organization

This report is organized into several chapters, each focusing on a specific aspect of the weather forecasting project, or to be more specific, on each steps as well as methods of tackling this problem. The organization of the report is as follows:

- Chapter 2: provides a comprehensive literature review on weather forecasting techniques, machine learning algorithms, and relevant research in the field.
- Chapter 3: provides a brief description on the dataset, the relation between the features, the importance of them as well as their importance or their effect on the label
- Chapter 4: presents the methodology used in this project, including the description of machine learning algorithms, model training, and evaluation techniques.
- Chapter 5: concludes the thesis by summarizing the findings, discussing the limitations of the project, and suggesting future research directions.
- Chapter 6: references

By following this organization, we aim to provide a comprehensive understanding of the research conducted and the results obtained in this machine learning-based weather forecasting project.

CHAPTER 2

LITERATURE REVIEW

2.1 Algorithms

- Random Forests: Random Forests is an ensemble learning algorithm that combines multiple decision trees to make predictions. Its ability to handle complex datasets and capture non-linear relationships has made it popular in predictive modeling tasks. In the context of our project, Random Forests can be employed to build a predictive model using various input features. Researchers have demonstrated the effectiveness of Random Forests in predicting medical diagnoses, stock market trends, and customer behavior. Evaluating the performance of Random Forests using metrics such as AUC, accuracy, and F1 score provides insights into the overall effectiveness of the model.

- Ridge Regression: Ridge Regression is a linear regression technique that incorporates regularization to handle multi-collinearity and prevent overfitting. It introduces a penalty term to the regression objective function, which shrinks the coefficient estimates. In our project, Ridge Regression can be used as a baseline predictive model, capturing the overall trend in the data. While Ridge Regression may not capture complex non-linear patterns, it provides interpretable results and can serve as a benchmark for comparison. Metrics such as AUC, accuracy, and F1 score can be employed to evaluate the performance of the Ridge Regression model and compare it against more advanced algorithms like Random Forests and Neural Networks.

- SVM (Support Vector Machine): Support Vector Machines (SVM) is a popular supervised learning algorithm used for classification and regression tasks. It finds an optimal hyperplane that separates classes or predicts a target variable. SVM can handle linear and non-linear data by using a kernel function. It has been successfully used in various domains such as text categorization and image classification. SVM offers good generalization and robustness.

2.2 Preprocessing Methods

- Data Cleaning: Describe the techniques used to clean the data, such as handling missing values, outliers, and inconsistencies. Highlight the importance of data cleaning in ensuring the quality and reliability of the dataset.
- Feature Selection/Extraction: Discuss the approaches employed for feature selection or feature extraction. These methods help identify the most relevant features or transform the original features into more informative representations. Explain how feature selection/extraction techniques contribute to improving model performance and reducing dimensionality.
- Data Normalization/Scaling: Explain the techniques used to normalize or scale the data, such as min-max scaling or standardization. Discuss the rationale behind these techniques and how they help in mitigating the impact of different scales or distributions of the features on the model's performance.
- Handling Imbalanced Data: If your research deals with imbalanced datasets, discuss techniques for handling class imbalance, such as oversampling, undersampling, or using class-weighted approaches. Emphasize the importance of addressing class imbalance to prevent biased model predictions.
- Data Partitioning: Briefly mention the strategies used for data partitioning, such as splitting the dataset into training, validation, and testing sets. Discuss the rationale behind the chosen partitioning approach and its implications for model evaluation and generalization.
- K-fold Cross-Validation: If cross-validation is used in the previous studies you review, mention the specific techniques employed (e.g., k-fold cross-validation) and highlight its importance in assessing the generalizability of the models.

2.3 Evaluation Metrics

To assess the performance of the machine learning models in our project, several evaluation metrics will be considered. The Area Under the Receiver Operating Characteristic Curve (AUC) provides a measure of the model's ability to discriminate between positive and negative instances. Accuracy is a commonly used metric that measures the overall correctness of the predictions. F1 score combines precision and recall and is particularly useful when dealing with imbalanced datasets. These metrics provide a comprehensive evaluation of the models' predictive capabilities and aid in model selection and comparison.

CHAPTER 3

DATA ANALYSIS

3.1 Diversity

Our dataset is not a complex dataset with 13 different features, and the number of records is approximately 10,000, which is enough to ensure the diversity of the data. The table below is a brief description of our features.

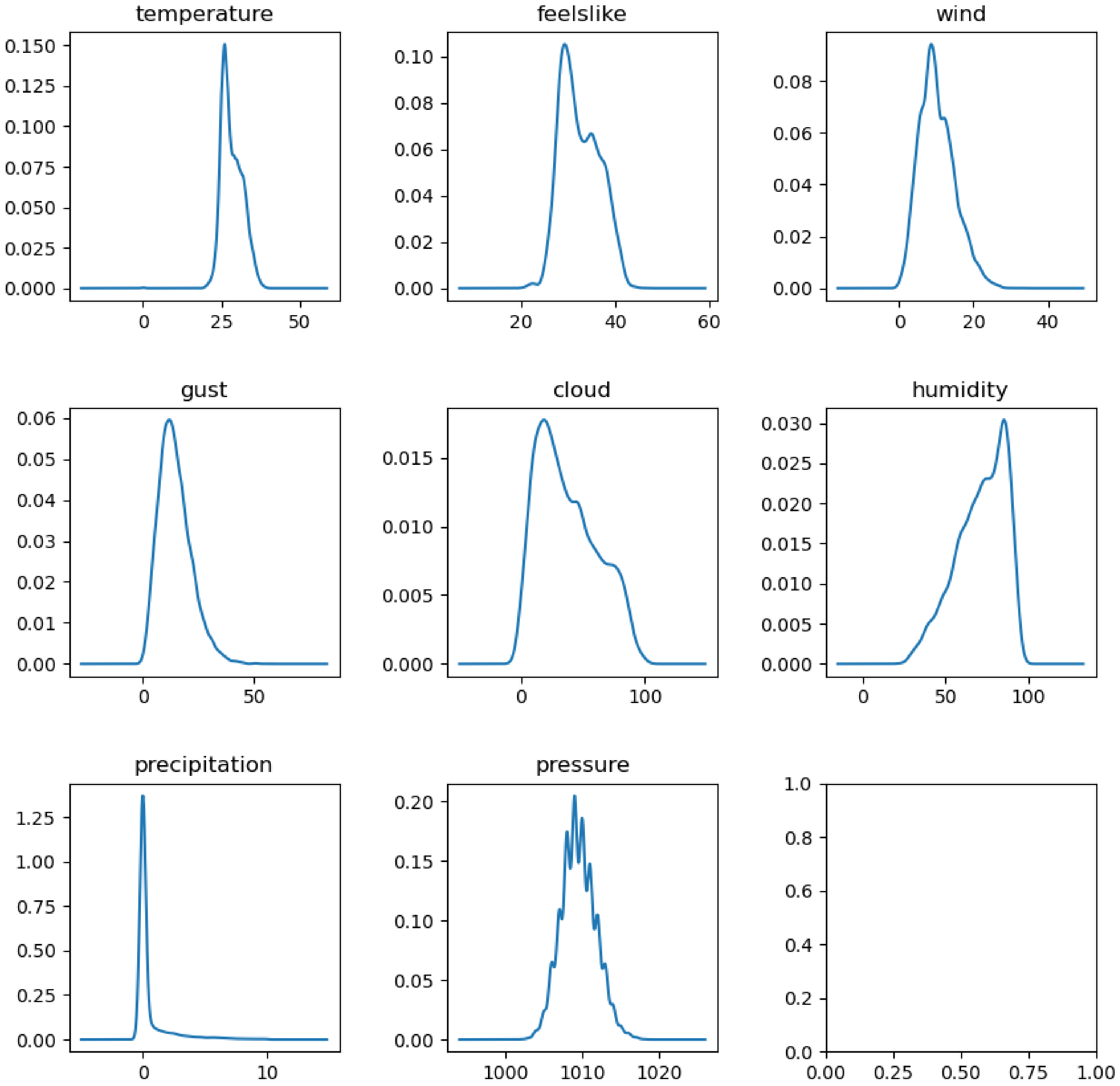
Dataset Overview

Features	Description	Unit
time	time of weather features	string - hh:mm
month	month of weather features	float
temperature	environment temperature	float - celsius
feelslike	temperature we feel	float - celsius
wind	wind speed	float - km/h
direction	wind direction	string
gust	maximum wind speed	float - km/h
cloud	cloud cover	float - %
humidity	air humidity	float - %
precipitation	amount of rain	float - mm
pressure	air pressure	float - bm
weather	overall weather	string

3.2 Features

3.2.1 Feature Distribution

We use KDE (Kernel Density Estimation) to have an overview of the distribution of all the numerical features.

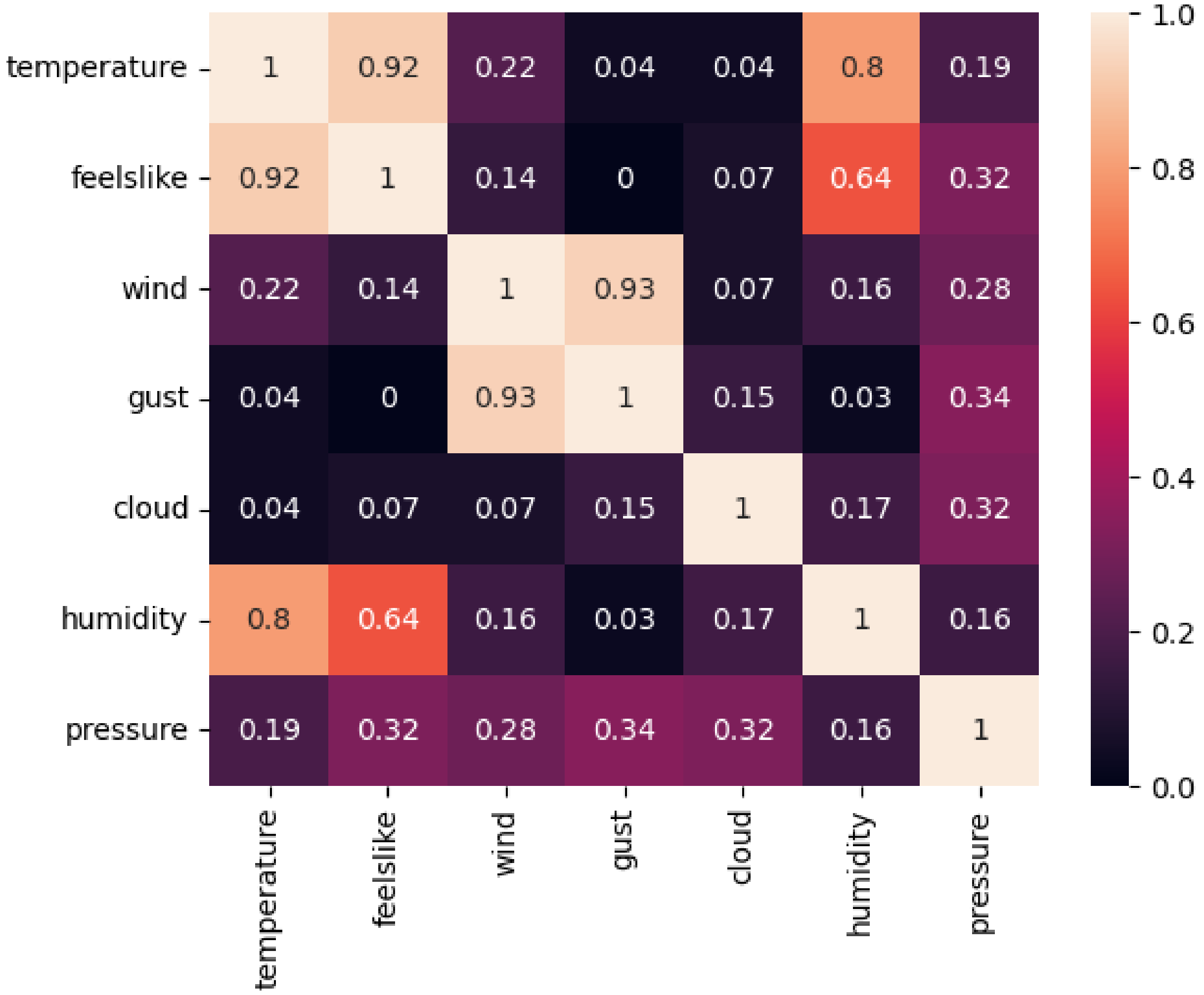


Data Distribution with KDE

Judging by the KDE, we decided to delete one feature which is the ’precipitation’ since their values are mostly ’0.0’, which might lead to the problem of Overfitting.

3.2.2 Feature Correlation

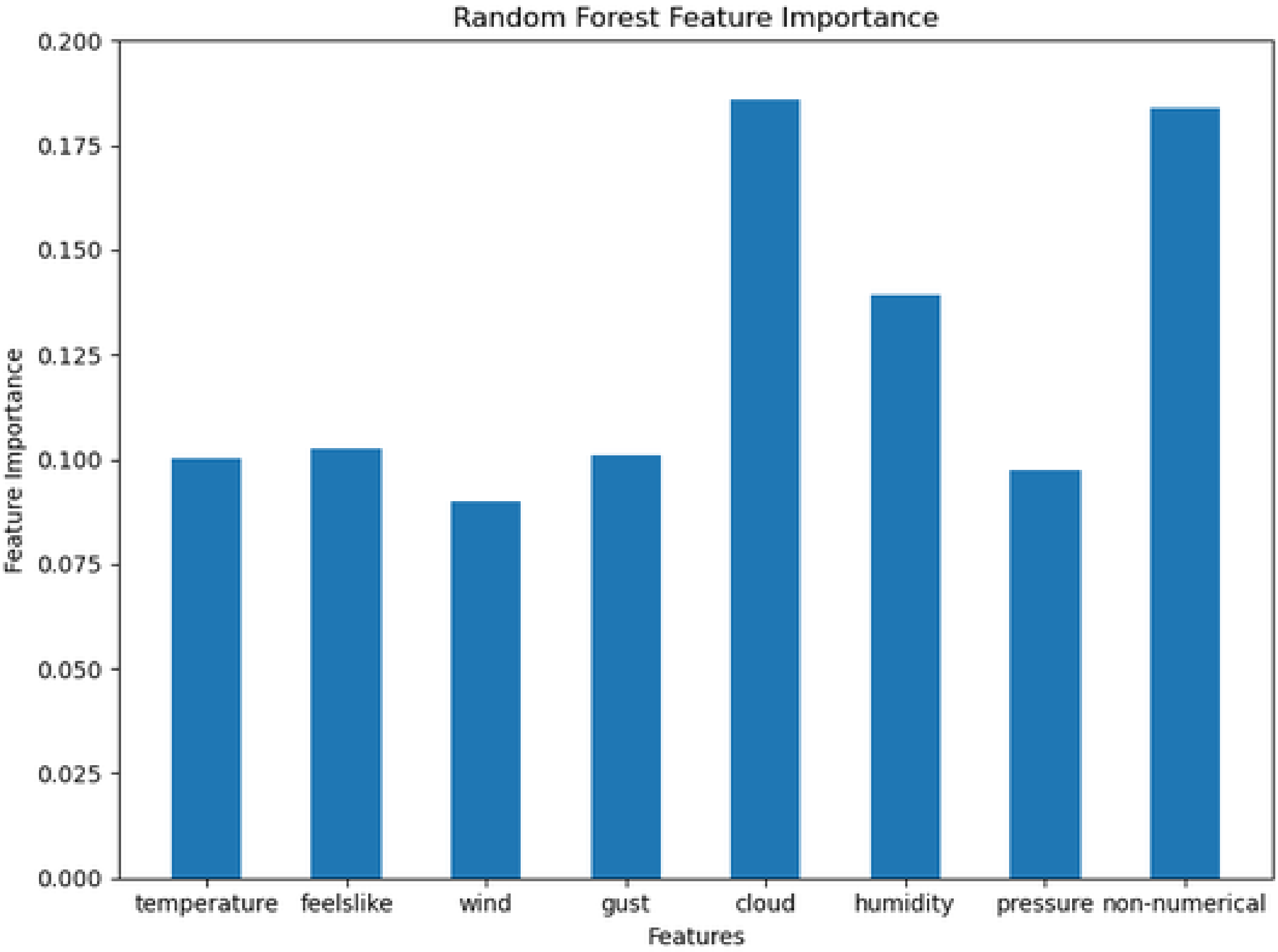
After that, we use the heatmap function from the ‘sns‘ library to provide a detailed and graphic view of the relation of every feature with the rest of them.



Correlation Heatmap

3.2.3 Feature Importance

Finally, we want to see the effect of each feature on the labels, so we use a bar graph to have a further decision on dropping any column to reduce the dimension of the data.



Feature Importance Graph

CHAPTER 4

METHODOLOGY

To solve this machine-learning based problem, we use 3 different approaches which are 4 different common algorithms. In this chapter, we will have a detailed view on how we preprocess data in each algorithm, test their accuracy, and draw some conclusions from the results.

4.1 Random Forest

Data Preprocessing

Firstly, we shift the dataset 1 row up so that the label of each row now is the label 3 hours later. And we need to drop the last row since it has no label.

Random Forest is known to use numerical data rather than categorical data, so we need to find the types of all columns and start encoding columns that are not 'int' or 'float'. We can see from Table **??** that there are 2 features that are not of 'float' type or 'int' type, excluding the label weather with which we will work later.

Feature: Direction

There are 16 different directions in the dataset, and we think that this number will make the data too complex to study. Therefore, we decide to change it into 8 common directions only:

Direction Change

Before change	**After change**
ENE, NE, NNE	NE
ESE, SE, SSE	SE
NNW, NW, WNW	NW
WSW, SW, SSW	SW
S	S
W	W
N	N
S	S

After that, we apply One-Hot Encoding to change these string ’directions’ into different dummy variables, which will appear in the complete pipeline at the end of the Data Processing.

Feature: Time

Since we have shifted our label 1 row up, which means that the label of each record is now the weather 3 hours later, there is no need to keep all those time values. So we also change values in time into 2 values: ’earlier’ or ’later’:

Time Change

Before change	**After change**
0:00, 3:00, 6:00, 9:00	earlier
12:00, 15:00, 18:00, 21:00	later

Feature: Month

We also change this feature since there are 12 months which are a bit too many. Therefore, we decide to use the season instead, which is also related to the weather:

Month Change

Before change	**After change**
1, 2, 3	spring
4, 5, 6	summer
7, 8, 9	autumn
10, 11, 12	winter

Label: Weather

There are exactly 20 labels in this dataset, which makes it very hard for the model to predict precisely. Additionally, the number of labels is considerably uneven. Therefore, we decide to change them into 6 labels only, making it easier for the model to learn, predict, and for the user to understand the weather:

Bảng 4.1: Weather Change

Before Change	After Change
Heavy rain at times, Heavy rain, Moderate rain at times, Moderate rain	Heavy/moderate rain
Mist, Light drizzle, Patchy light drizzle	Mist/drizzle
Light rain, Patchy light rain with thunder, Patchy light rain, Light rain shower	Light rain
Thundery outbreaks possible, Overcast, Moderate or heavy rain shower, Patchy rain possible, Partly cloudy	Cloudy
Sunny	Sunny
Clear	Clear

After splitting the labels into 6 new labels, we see that the number of labels is considerably unequal, which leads to the use of SMOTE (Synthetic Minority Oversampling Technique) to balance the dataset:

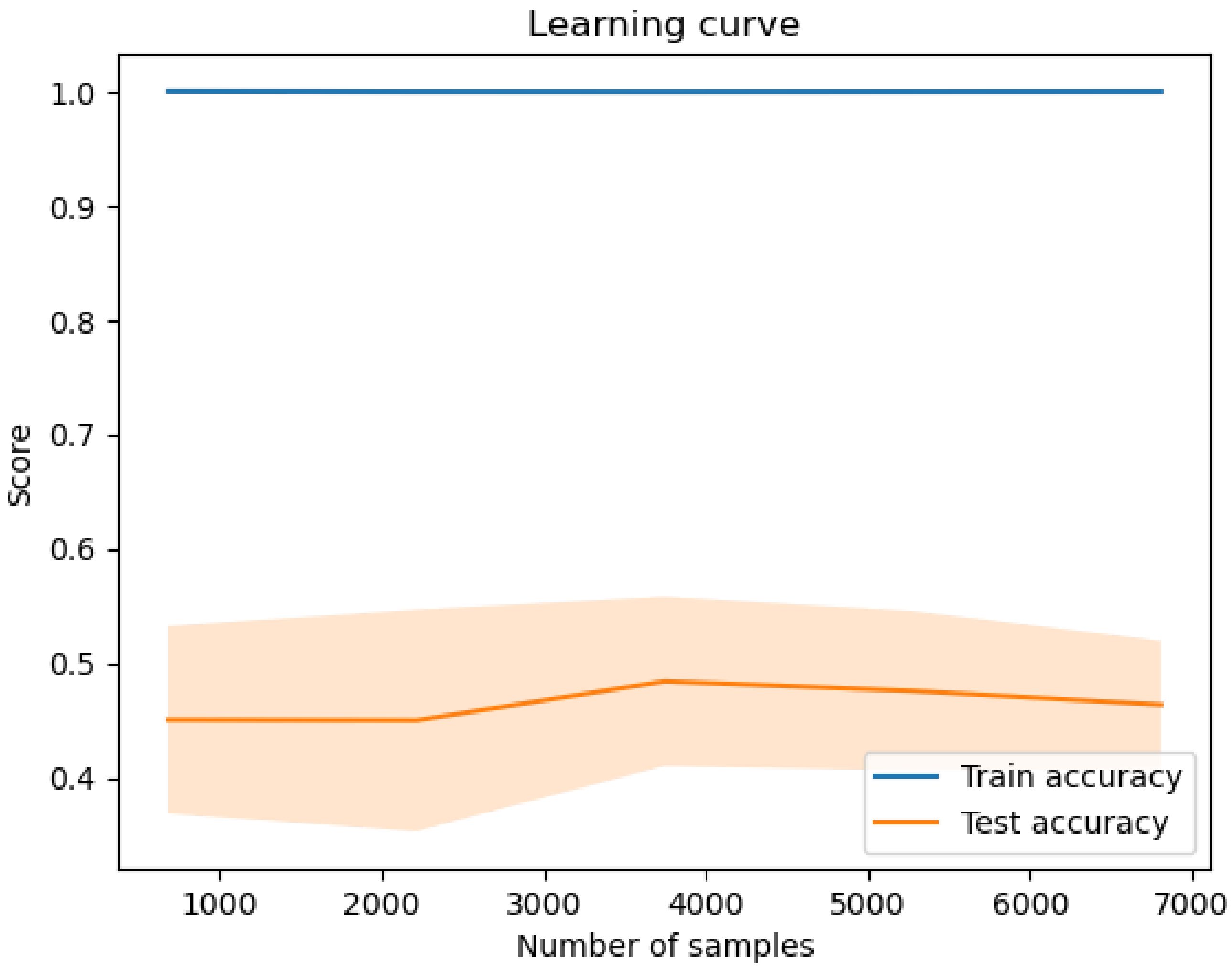
Label	Count
Cloudy	4294
Light rain	1179
Clear	1102
Sunny	1003
Heavy/moderate rain	864
Mist/drizzle	69

↓

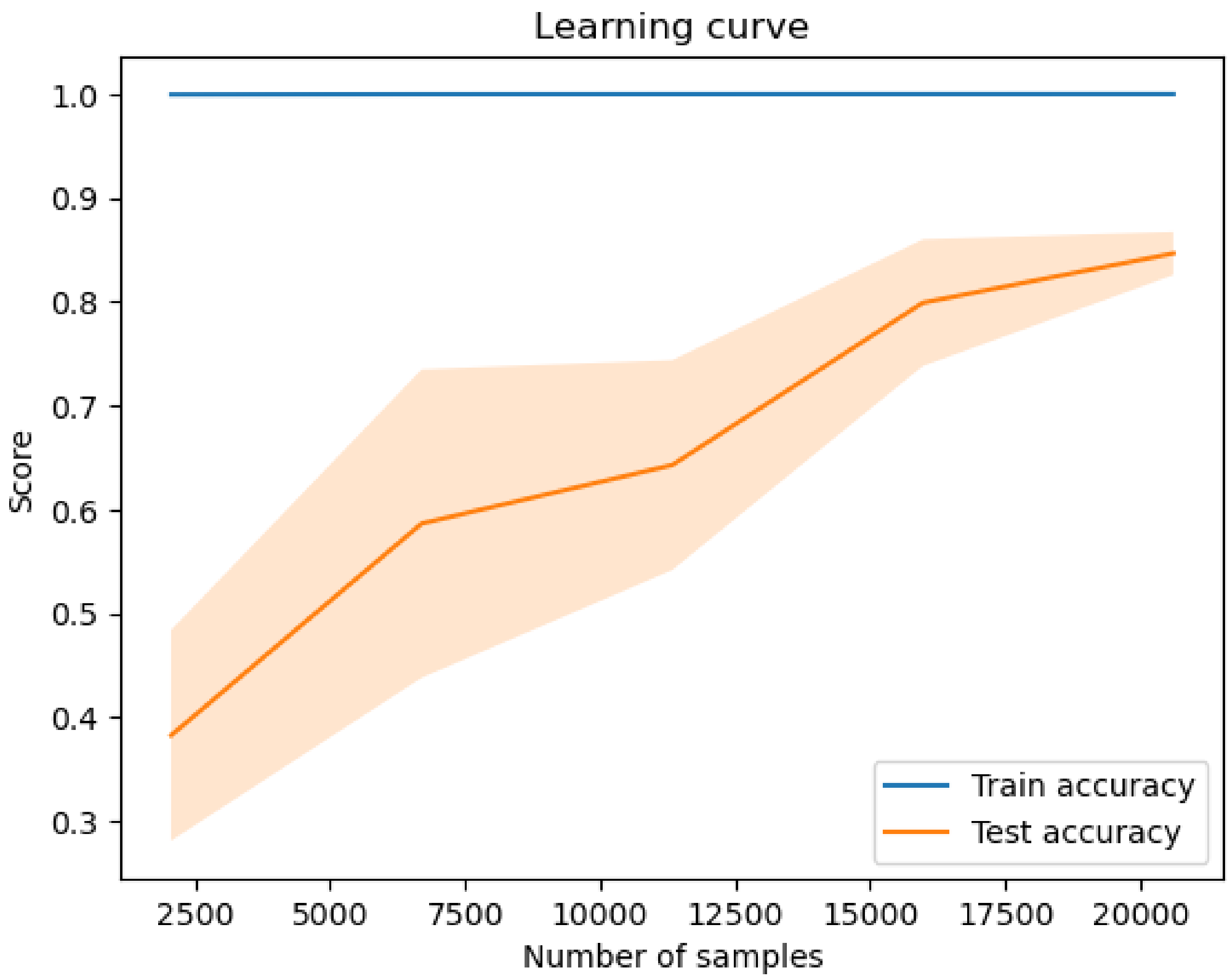
Label	**Count**
Clear	4294
Sunny	4294
Cloudy	4294
Light rain	4294
Heavy/moderate rain	4294
Mist/drizzle	4294

Before Label Sampling and After Label Sampling

And to see the effect of the act of oversampling, we show 2 different learning curves of the dataset before sampling and after sampling to compare:



Before Sampling



After Sampling

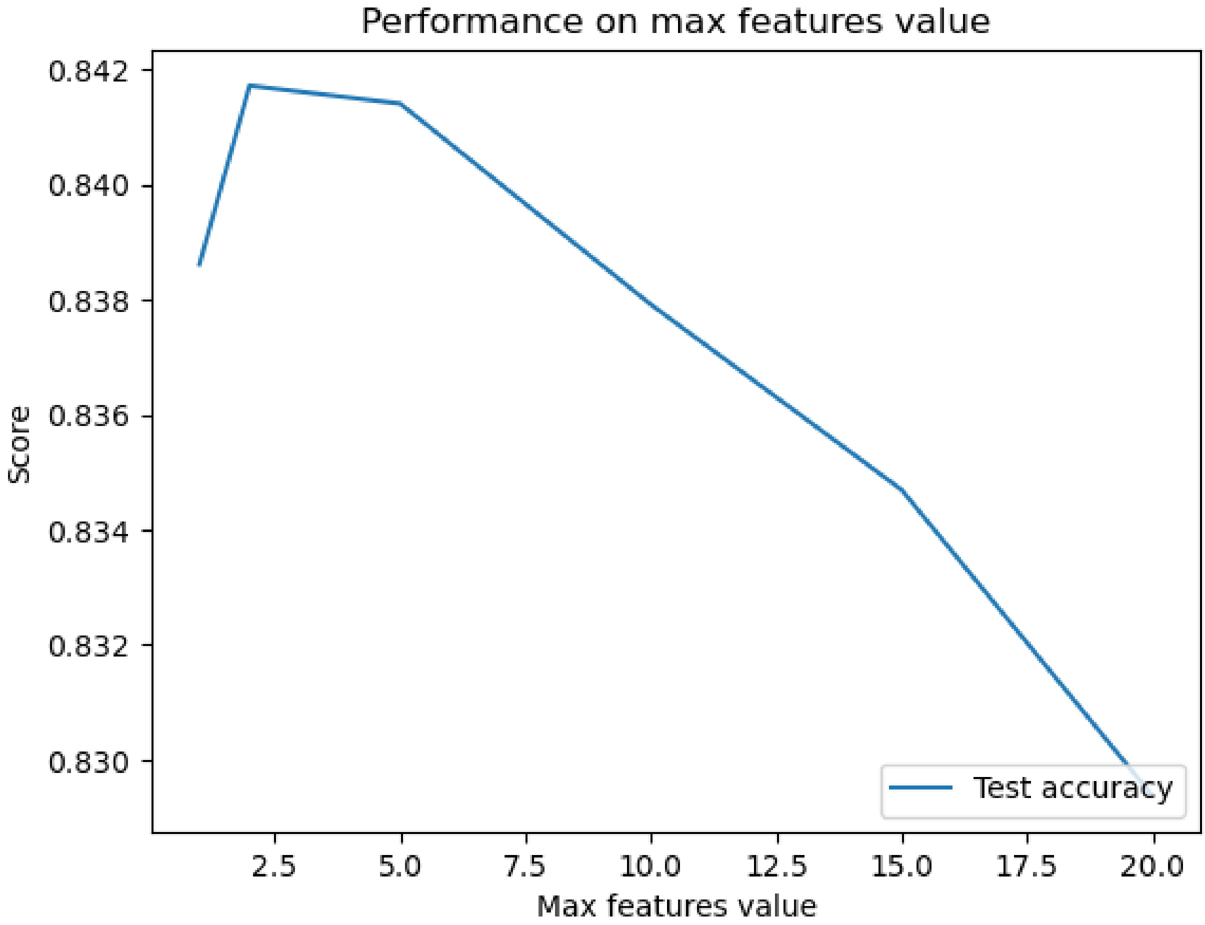
We also create a pipeline that includes the use of One-hot Encoding for the categorical features and the StandardScaler for the numerical feature of which the purpose is to make the precision of the prediction higher.

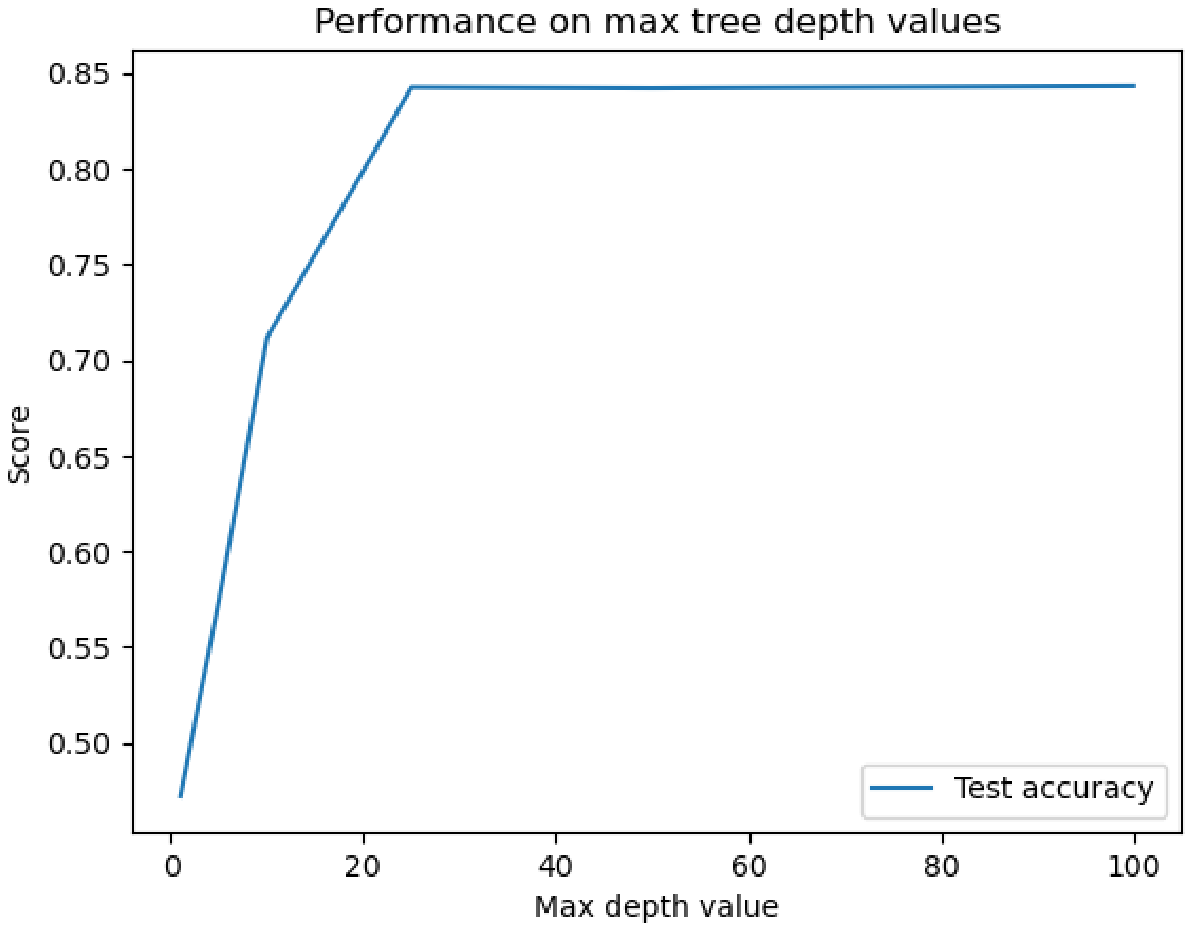
Train-Test Split

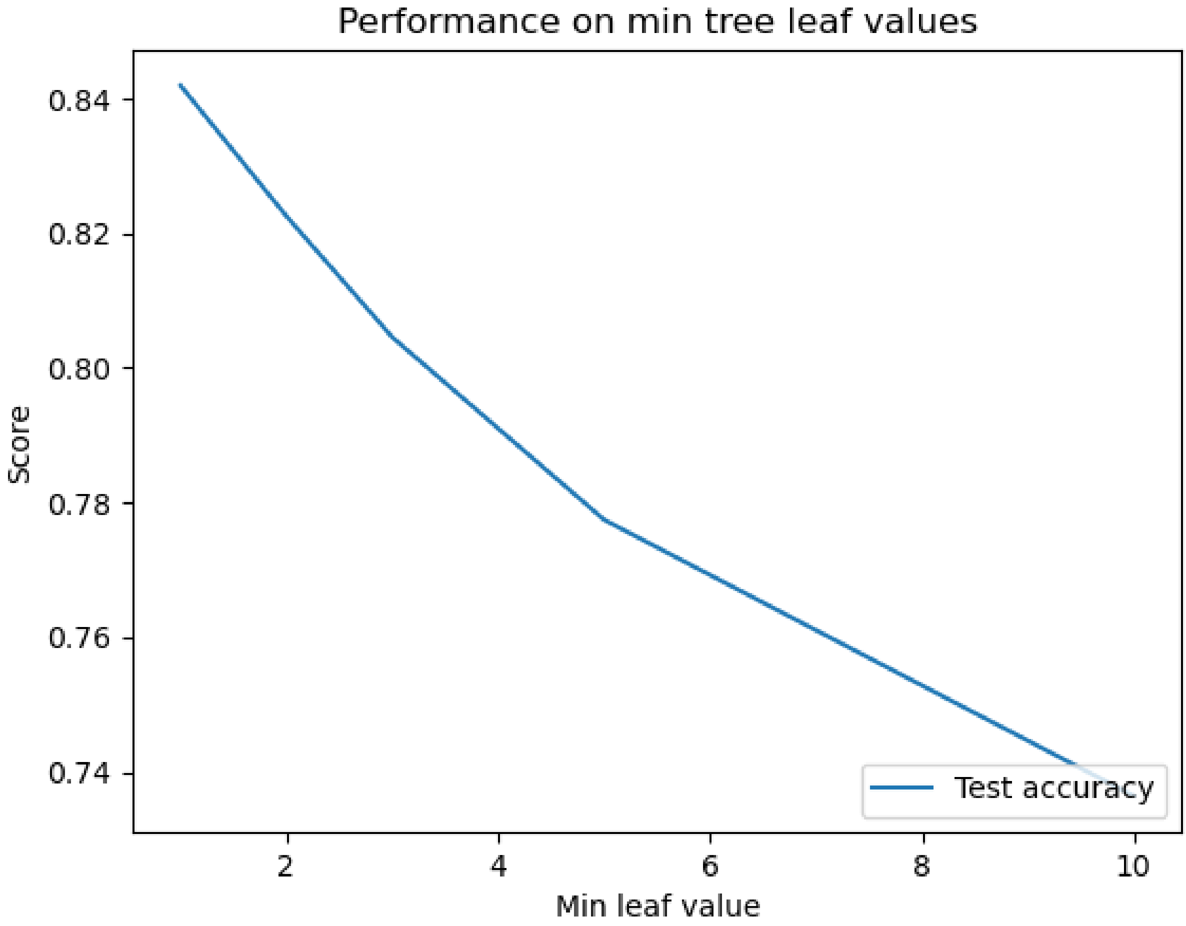
After preprocessing the data, we need to split the dataset into two sets: the train set and the test set. We split the dataset with a ratio of 2:8, which means that 80% of the dataset will be used as the train set, and the remaining 20% will be the test set.

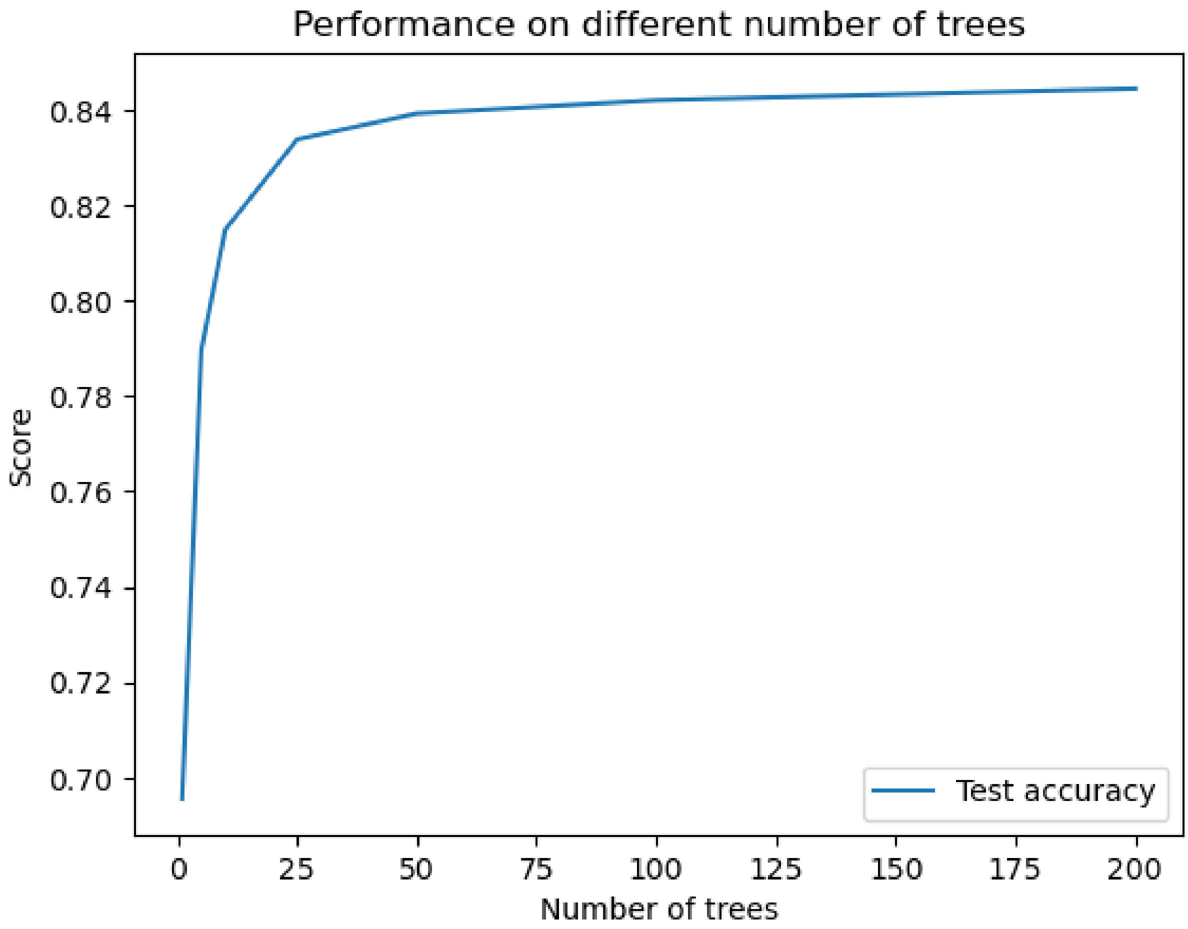
Parameter Choosing

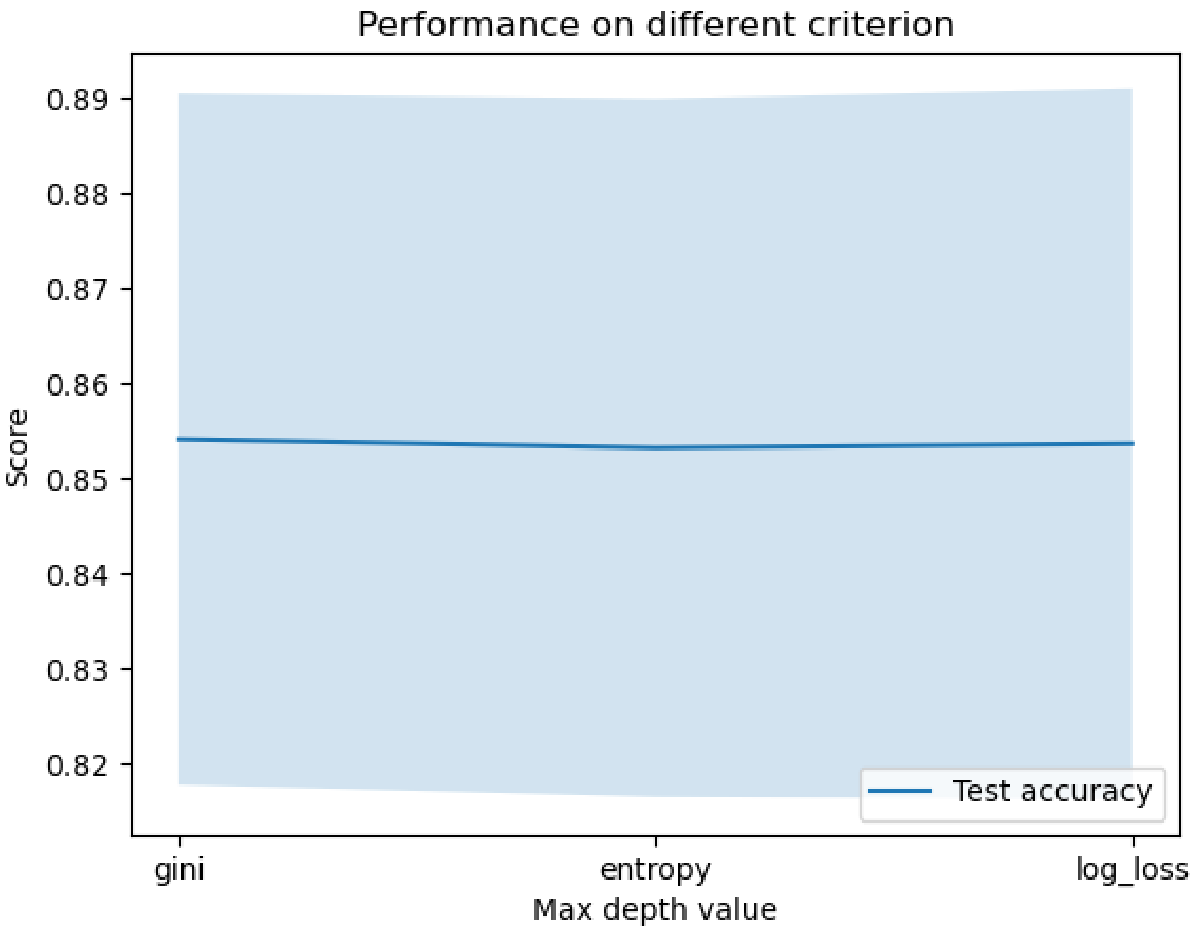
In order to choose the best parameter for Random Forest, we use K-fold Cross Validation as well as different functions to test a list of parameters and receive these results:


Performance on max features value
Score
0.842
0.840
0.838
0.836
0.834
0.832
0.830
Max features value
2.5
5.0
7.5
10.0
12.5
15.0
17.5
20.0
Test accuracy


Performance on max tree depth values
Score
0.85
0.80
0.75
0.70
0.65
0.60
0.55
0.50
Max depth value
0
20
40
60
80
100
Test accuracy


Performance on min tree leaf values
Score
0.84
0.82
0.80
0.78
0.76
0.74
Min leaf value
2
4
6
8
10
Test accuracy


Performance on different number of trees
Score
0.70
0.72
0.74
0.76
0.78
0.80
0.82
0.84
Number of trees
0
25
50
75
100
125
150
175
200
Test accuracy



According to those graphs, we decide to choose the following as parameters for out RandomForestClassifier:

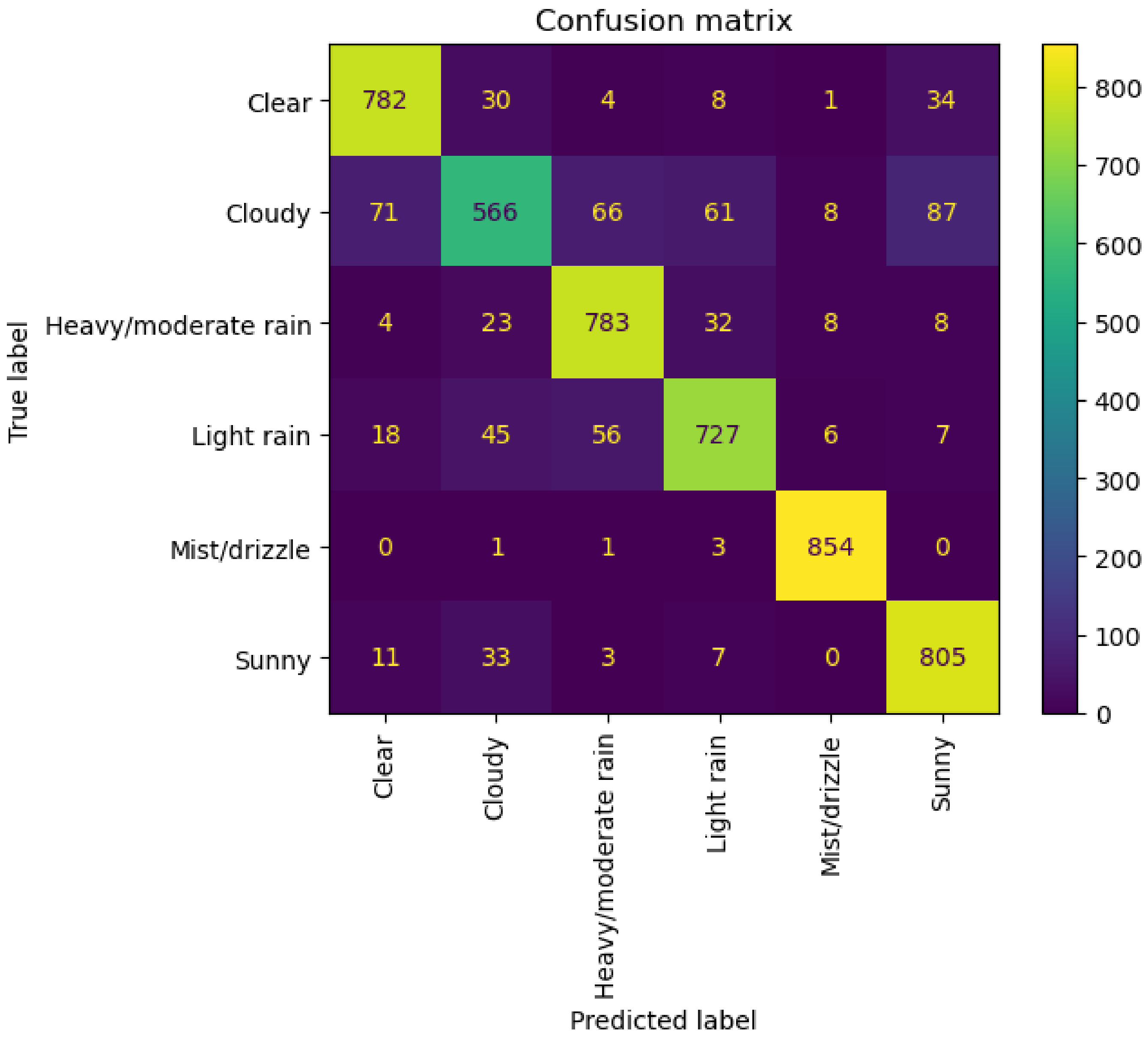
Parameter	Choice
n_estimators	150
max_depth	25
criterion	log_loss
max_features	4

Results and Conclusion

These are the results when we use Random Forest:

Metric	Result
Training accuracy score	1.00
Accuracy score	0.87
Recall score	0.87
Precision score	0.87
F1 score	0.87
ROC AUC score	0.98

And the Confusion Matrix for an easier analysis:



By analyzing the results, we can see that Random Forest is very efficient in solving problems that have both numerical and categorical data like this. It has most of the labels predict correctly, unless the ’Cloudy’ weather seems to be the only flaw.

4.2 SVM (Support Vector Machine)

Data Preprocessing

The data preprocessing here is the same as in the section 4.1. You can go back to the 4.1 to see the Data Preprocessing.

Train-Test Split

The Train-Test Split here is the same as in the section 4.1. You can go back to the 4.1 to see the Train-Test Split.

Parameter Choosing

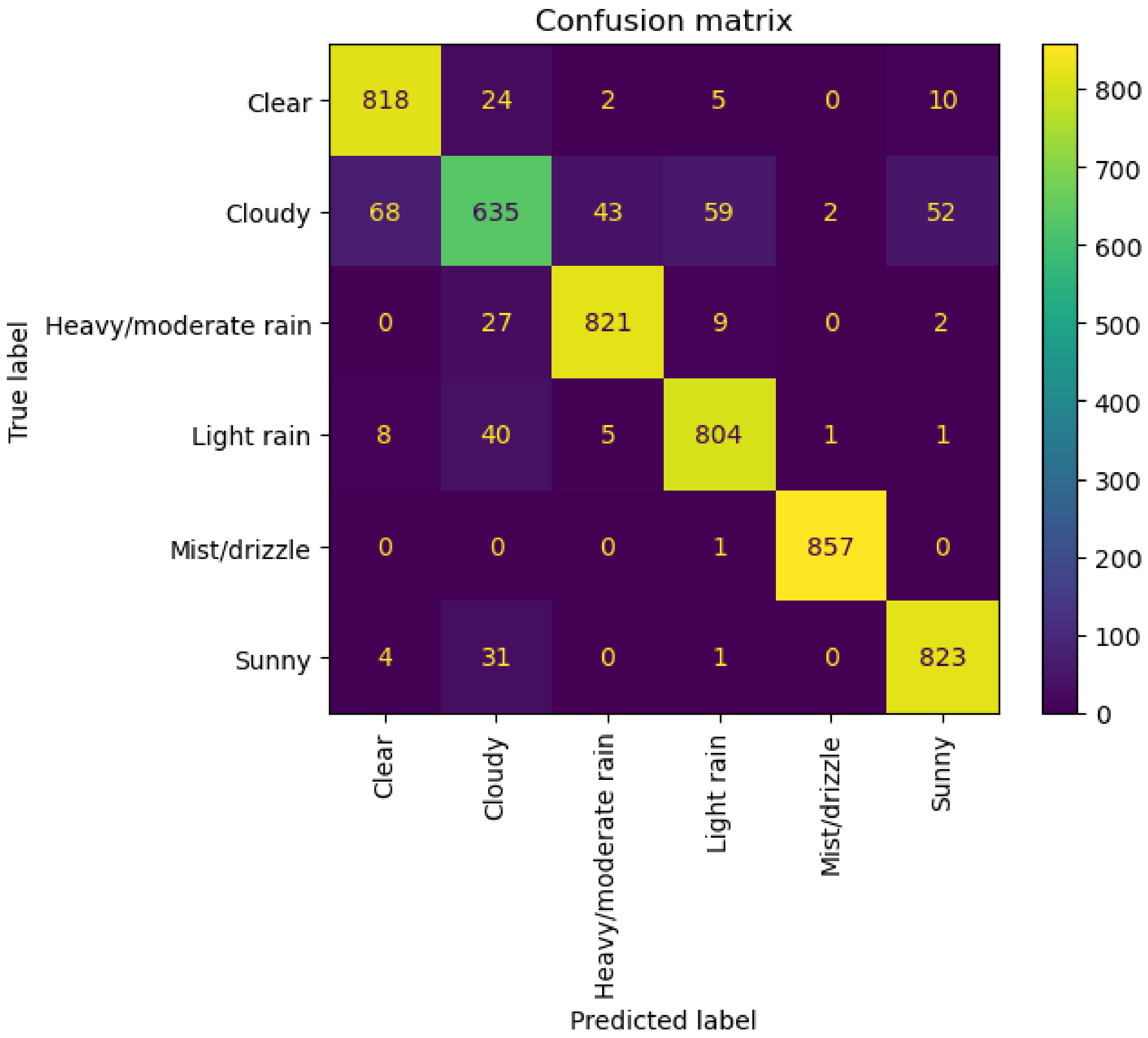
With this algorithm, we also use some met.

Results and Conclusion

These are the results when we use SVM:

Metric	**Result**
Training accuracy score	1.00
Accuracy score	0.92
Recall score	0.92
Precision score	0.92
F1 score	0.92
ROC AUC score	0.99

And the Confusion Matrix for an easier analysis:



By analyzing the results, we can see that SVM is even better than Random Forest in solving this problem. It has most of the labels predict almost flawlessly, still the 'Cloudy' weather seems to be the one flaw.

4.3 Ridge Regression

Since this is a Classifier problem and we should not use Ridge Regression, we want to approach this problem a different way. Since the problem is predicting the overall weather, a number of people will be more pleased to know the temperature than to know the weather. That is why we want to tackle a new problem, which is 'Predicting the temperature 3 hours later using the numerical features'

Data Preprocessing

Firstly, we extract from the dataset all the essential features for the RidgeRegression, which are the numerical features. Then we drop the outliers as well as shift the dataset 1 row up so that the label of each row now is the label 3 hours later. And we also have to drop the last row since it has no label.

Train-Test Split

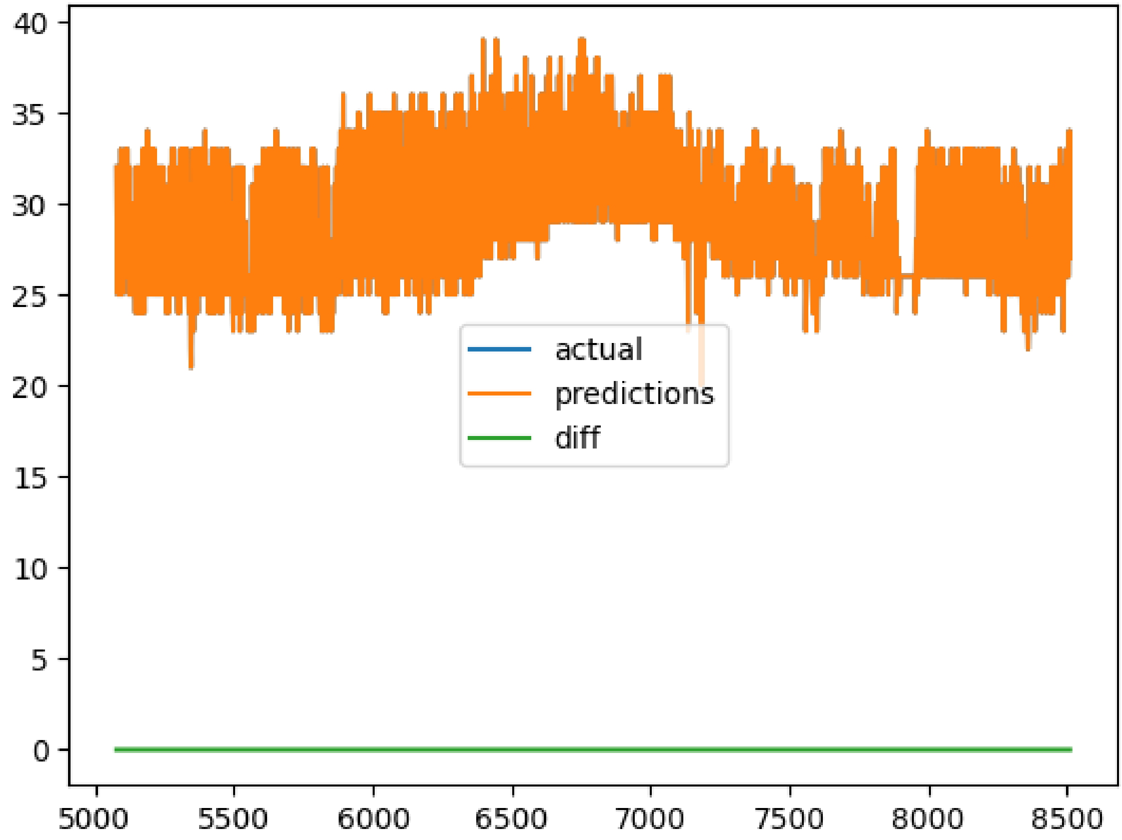
After preprocessing the data, we need to split the dataset into two sets: the train set and the test set. We split the dataset with a ratio of 4:6, which means that 60% of the dataset will be used as the train set, and the remaining 40% will be the test set.

Parameter Choosing

Here we choose the parameter Alpha = 0.1 since in Ridge regression model, the parameter alpha controls the regularization strength. A higher value of alpha can decrease the complexity of the model and increase bias but reduce variance. On the other hand, a lower value of alpha can increase complexity, decrease bias but increase variance.

Results and Conclusion

To better see the results, we have a plot to see the difference between the actual values and the predicted ones:



By analysing this plot, we can conclude that with all those chosen numerical features, RidgeRegression will perform excellently and predict almost every values correct

CHAPTER 5

CONCLUSION

After using all 3 different algorithms, we can now compare them to have the final conclusion As the numerical and graphical results have been shown, the Random Forest can handle this problem very well with the all the results of different metrics reaching nearly 90%, and SVM even exceeds those numbers with 92% of most of metrics. On the other hand, approaching the problem in a diffent perspective, the RidgeRegression also performs very well with an excetionally high accuracy. However, there are still many improvement can be made, such as preprocessing data in more detail, use different methods to modify data,...

CHAPTER 6

REFERENCES

leena Leena H. Alamri, Ranim S. Almuslim, Mona S. Alotibi, Dana K. Alkadi, Irfan Ullah Khan, Nida Aslam, "Predicting Student Academic Performance using Support Vector Machine and Random Forest" in *ICETM '20: Proceedings of the 2020 3rd International Conference on Education Technology Management*, December 2020, pp. 100–107.

sklearn RandomForestClassifier. [Online]. Available: https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.RandomForestClassifier.html (visited on 06/30/2023).

sklearn Support Vector Machine. [Online]. Available: https://scikit-learn.org/stable/modules/svm.html (visited on 06/28/2023).

qingcai Qingyuan Cai, Peng Li, Ruchuan Wang, "Electricity theft detection based on hybrid random forest and weighted support vector data description" in *International Journal of Electrical Power Energy Systems*. Volume 153. 2023.

Jason Brownlee Jason Brownlee PhD, *Hypertext transfer protocol securityl (HTTPS)*. [Online]. Available: https://machinelearningmastery.com/framework-for-imbalanced-classification-projects/ (visited on 09/25/2010)

Pham Dinh Khanh DeepAI KhanhBlog, *Hypertext transfer protocol securityl (HTTPS)*. [Online]. Available: https://phamdinhkhanh.github.io/deepai-book/ch_appendix/appendix_matplotlib.html#density (visited on 09/25/2010)