

**CSE422 : Artificial Intelligence**

**Project Report**

**Project Title : Rain or snow prediction in Hungary**

| **Group: 13, Lab Section : 09, Summer 2023** | |
| --- | --- |
| **ID** | **Name** |
| 21301610 | Shihab Muhtasim |
| 21301648 | Raiyan Wasi Siddiky |

**Table of Contents**

| **Section**  **No** | **Content** | **Page No** |
| --- | --- | --- |
| 1 | Introduction | 2 |
| 2 | Dataset description | 3 |
| 3 | Correlation of all features | 4 |
| 4 | Imbalanced Dataset | 5 |
| 5 | Dataset Preprocessing | 6 |
| 6 | Feature scaling | 12 |
| 7 | Model Selection Comparison Analysis | 13 |

**Introduction: Weather Prediction for Rain or Snow in Hungary**

Weather prediction is an extremely critical aspect of modern life, influencing various sectors ranging from agriculture to transportation. This is especially true in the case of extreme weather conditions such as rain or snow, which can bring commercial life to a halt. It is therefore essential to have accurate weather forecasts, as they can help individuals and organizations make informed decisions and plan their activities accordingly. This project, "Weather Prediction: Rain or Snow in Hungary," aims to address the challenge of predicting whether it will rain or snow in Hungary based on various meteorological variables.

Project Objectives:

The primary objective of this project is to develop a predictive model that can forecast whether a specific location in Hungary will experience rain or snow within a given time frame. By utilizing historical weather data and applying machine learning techniques, we seek to create a model that offers reliable predictions with a high degree of accuracy. This project will contribute to enhancing the precision of weather forecasts, providing valuable insights for individuals, businesses, and authorities in Hungary.

Problem Statement:

Hungary, like many regions, experiences unpredictable weather conditions that can have significant implications for day-to-day activities and planning. Traditional forecasting methods may fall short in providing accurate predictions for rapidly changing weather patterns, particularly in cases of rain and snow. This project addresses the challenge of improving the accuracy of weather predictions for rain and snow, enabling residents, farmers, transportation services, and other stakeholders to make well-informed choices based on reliable forecasts.

Motivation:

The motivation behind this project lies in the potential positive impact on individuals' lives and societal activities. Accurate weather predictions can empower people to optimize their schedules, mitigate risks associated with adverse weather, and even enhance agricultural practices. Additionally, sectors such as transportation and emergency services can benefit from timely information about rain and snowfall, ensuring public safety and efficient resource allocation. By harnessing the power of data and machine learning, this project aims to contribute to the well-being and productivity of Hungary's population.

**Dataset description**

**Source:** Kaggle

**Link:** [weather prediction ,regression , neural model | Kaggle](https://www.kaggle.com/code/salmamaamouri/weather-prediction-regression-neural-model/input)

**Reference:** Salma Maamouri

**Dataset description**

1. **How many features:**

There are 10 features initially. Such as: Summary , Apparent Temperature , Humidity , Wind Speed (km/h) ,Wind Bearing (degrees), Visibility (km), Loud Cover , Pressure (millibars), Daily Summary, Temperature (C) .

1 class: Precip type with two unique values rain, snow

After pre- processing: 35 features

1. **Is this a classification or regression problem ? Why do you think so?**

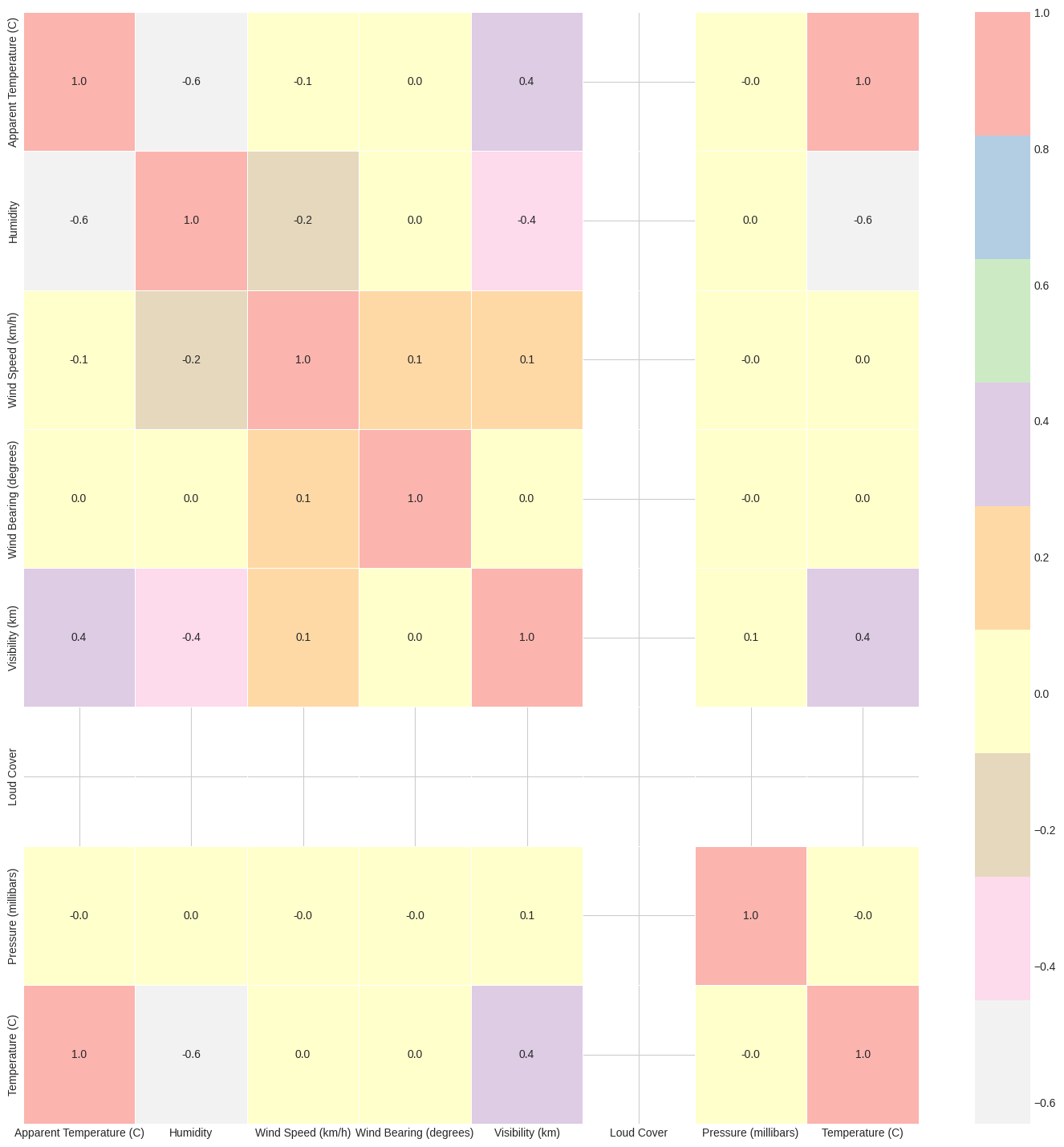
In this project, we are addressing a classification problem. Classification involves the task of predicting categories or classes based on given input features. In the context of weather prediction in Hungary, our aim is to predict whether the weather conditions will lead to the occurrence of rain or snow. These outcomes, "rain" and "snow," are distinct categories that we are trying to predict based on various weather-related features. Therefore, the nature of our task, which involves predicting a categorical label (rain or snow), signifies that our project falls within the domain of classification problems.

1. **Data points:** 96453
2. **What kind of features are in your dataset? Quantitative or categorical?**

The dataset contains a mixture of both quantitative and categorical features. Quantitative features are those that involve numerical measurements, such as temperature, humidity, wind speed, and pressure. These features have values that can be quantified and compared.

On the other hand, categorical features represent different categories or labels. For instance, features like "Summary" and "Daily Summary" provide descriptions of the weather conditions in categorical terms. Similarly, "Precip Type" is a categorical feature indicating whether it's raining or snowing. These features help classify weather conditions into specific categories based on their descriptions.

1. **Correlation of all features:**



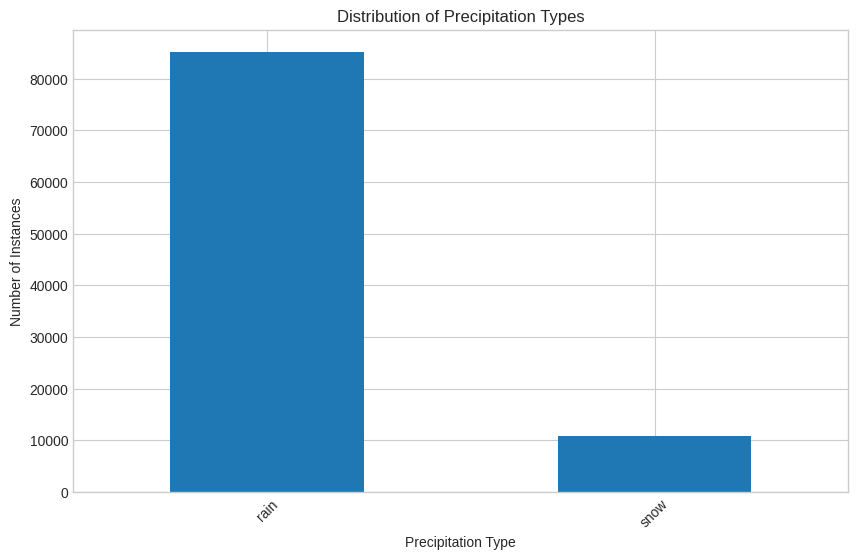
**Imbalanced Dataset**

1. For the output feature, do all unique classes have an equal number of instances or not?

* No

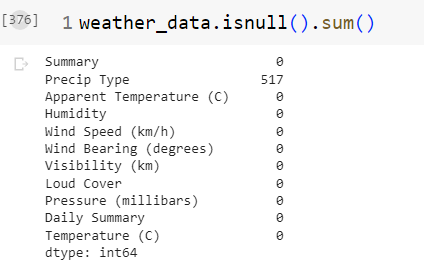
1. Represent using a bar chart of N classes (N=number of classes you have in

your dataset).



**Dataset Preprocessing**

**Faults: Null values**



**Drop null values rows:**

print("Shape of dataframe before dropping:", weather\_data.shape)

weather\_data = weather\_data.dropna(axis = 0, subset = ['Precip Type'])

print("Shape after dropping:", weather\_data.shape)

**Drop null value column:**

weather\_data.drop('Loud Cover', axis=1, inplace=True)

weather\_data.head()

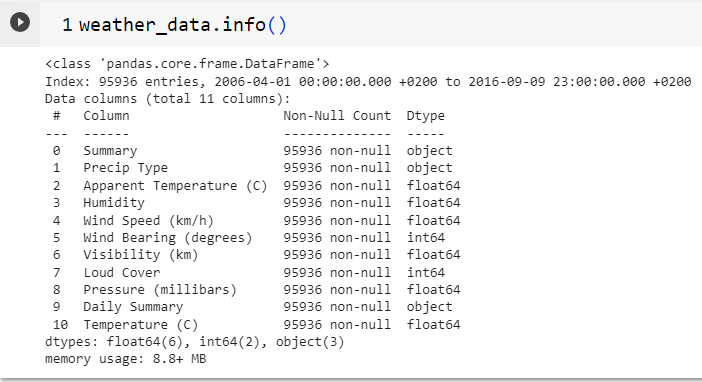
**Impute values:**

imputer = SimpleImputer(strategy='mean')

X = weather\_data.drop('Precip Type', axis=1)

X\_imputed = imputer.fit\_transform(X)

**Fault: Categorical values:**



The object types are qualitative data while the float and int type data are quantitative.

**Encoding**

We need to replace the categorical data with numerical data so that the computer can understand and notice a pattern so that it can make a prediction. We do this via:

1. **Binary encoding:**

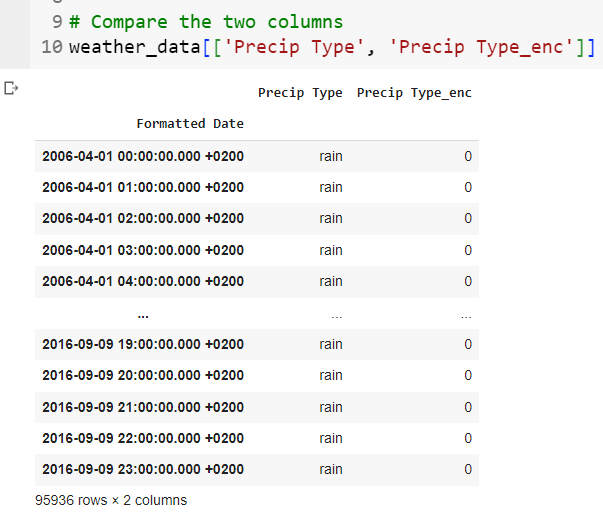
from sklearn.preprocessing import LabelEncoder

# Setting up the LabelEncoder object

enc = LabelEncoder()

# Apply the encoding to the "Precip Type" column

weather\_data['Precip Type\_enc'] = enc.fit\_transform(weather\_data['Precip Type'])



1. **Mapping:**

weather\_data['Daily Summary'] = weather\_data['Daily Summary'].map({

'Partly cloudy throughout the day.': 1,

'Mostly cloudy throughout the day.': 2,

'Foggy in the evening.': 3,

'Foggy overnight and breezy in the morning.': 4,

'Overcast throughout the day.': 5,

'Partly cloudy until night.': 6,

'Mostly cloudy until night.': 7,

'Foggy starting overnight continuing until morning.': 8,

'Foggy in the morning.': 9,

'Partly cloudy until evening.': 10,

'Partly cloudy starting in the morning.': 11,

'Mostly cloudy starting overnight continuing until night.': 12,

'Partly cloudy starting in the afternoon.': 13,

'Partly cloudy starting overnight.': 14,

'Mostly cloudy starting overnight.': 15,

'Mostly cloudy until night and breezy in the afternoon.': 16,

'Mostly cloudy until evening.': 17,

'Foggy throughout the day.': 18,

'Partly cloudy starting in the morning.': 19,

'Partly cloudy starting in the morning continuing until evening.': 20,

'Foggy until morning.': 21,

'Partly cloudy starting in the morning continuing until night.': 22,

'Mostly cloudy starting in the morning.': 23,

'Foggy starting in the evening.': 24,

'Partly cloudy starting in the afternoon continuing until evening.': 25,

'Foggy overnight.': 26,

'Clear throughout the day.': 27,

'Partly cloudy starting overnight continuing until night.': 28,

'Partly cloudy overnight.': 29,

'Partly cloudy starting overnight continuing until evening.': 30,

'Foggy until night.': 31,

'Partly cloudy in the morning.': 32,

'Foggy starting overnight continuing until afternoon.': 33,

'Foggy until afternoon.': 34,

'Breezy and mostly cloudy overnight.': 35,

'Partly cloudy overnight and breezy starting in the morning continuing until afternoon.': 36,

'Breezy in the morning and foggy in the evening.': 37,

'Mostly cloudy until evening and breezy in the evening.': 38,

'Mostly cloudy starting in the evening.': 39,

'Mostly cloudy throughout the day and breezy starting overnight continuing until afternoon.': 40,

'Breezy starting in the morning continuing until night.': 41,

'Overcast throughout the day and breezy starting overnight continuing until morning.': 42,

'Breezy starting overnight continuing until morning and foggy in the evening.': 43,

'Light rain until morning.': 44,

'Mostly cloudy until night and breezy starting in the afternoon continuing until night.': 45,

'Mostly cloudy starting in the morning continuing until afternoon.': 46,

'Breezy until afternoon and overcast throughout the day.': 47,

'Partly cloudy until evening and breezy in the afternoon.': 48,

'Breezy starting overnight continuing until morning and partly cloudy starting overnight continuing until evening.': 49,

'Light rain starting overnight.': 50,

'Partly cloudy starting overnight continuing until evening and breezy starting in the morning continuing until evening.': 51,

'Foggy starting in the morning continuing until evening and breezy in the evening.': 52,

'Partly cloudy throughout the day and breezy in the afternoon.': 53,

'Mostly cloudy starting overnight continuing until evening and breezy starting overnight continuing until morning.': 54,

'Partly cloudy starting overnight continuing until evening and breezy in the morning.': 55,

'Overcast throughout the day and breezy overnight.': 56,

'Light rain in the morning.': 57,

'Rain until morning.': 58,

'Breezy in the morning and mostly cloudy starting in the evening.': 59,

'Mostly cloudy starting in the morning and breezy overnight.': 60,

'Partly cloudy starting overnight and breezy starting in the morning continuing until afternoon.': 61,

'Partly cloudy starting in the morning and breezy starting in the afternoon continuing until evening.': 62,

'Partly cloudy starting in the morning continuing until evening and breezy in the afternoon.': 63,

'Foggy starting overnight continuing until morning and breezy in the afternoon.': 64

})

1. **One hot encoding:**

summary\_enc = pd.get\_dummies(weather\_data['Summary'])

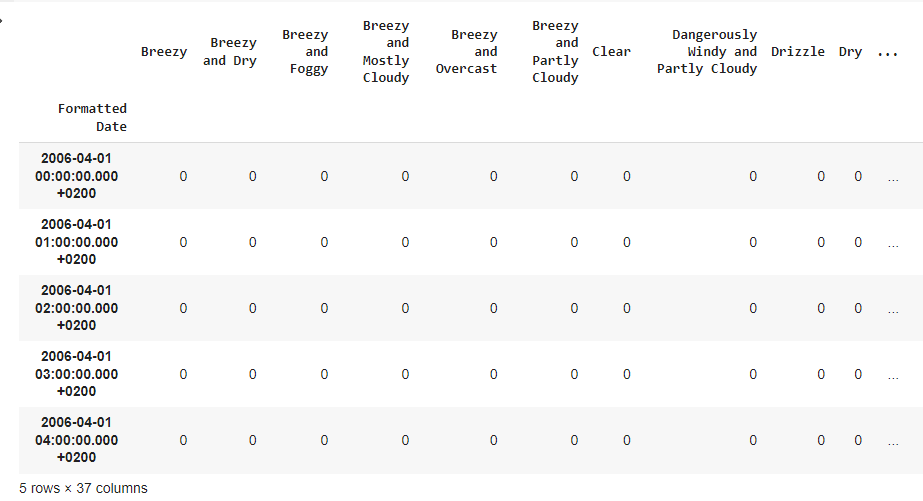
# Drop original 'Summary' column

weather\_data.drop('Summary', axis=1, inplace=True)

# Concatenate with the original dataframe

weather\_data = pd.concat([summary\_enc, weather\_data], axis=1)

weather\_data.head()



1. **Feature scaling:**

One of the problems that the algorithm often faces is that it gets biased towards larger values. To avoid this problem, we scale the numerical data to bring it in a small range.

from sklearn.preprocessing import StandardScaler

# Separate the target variable

a = weather\_data.drop('Precip Type', axis=1)

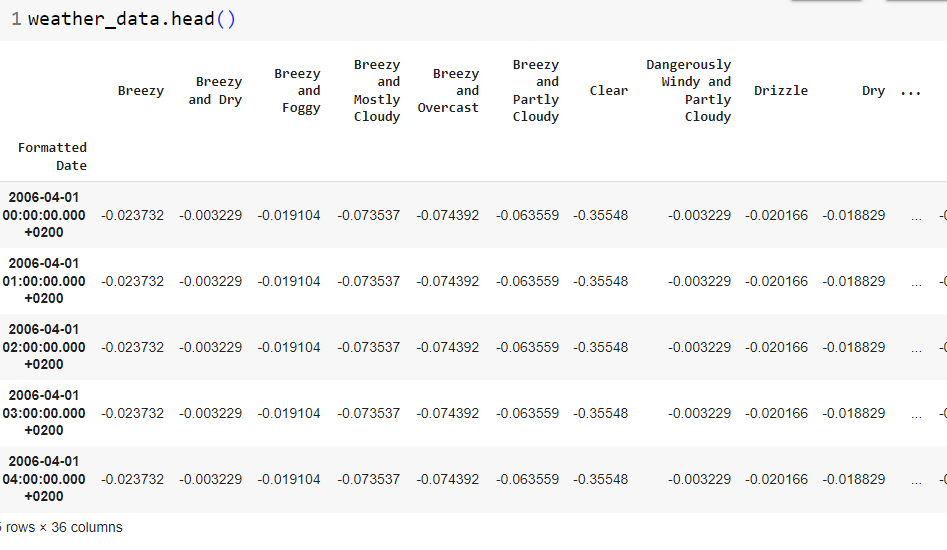
# Initialize the StandardScaler

scaler = StandardScaler()

scaled\_features = scaler.fit\_transform(a)

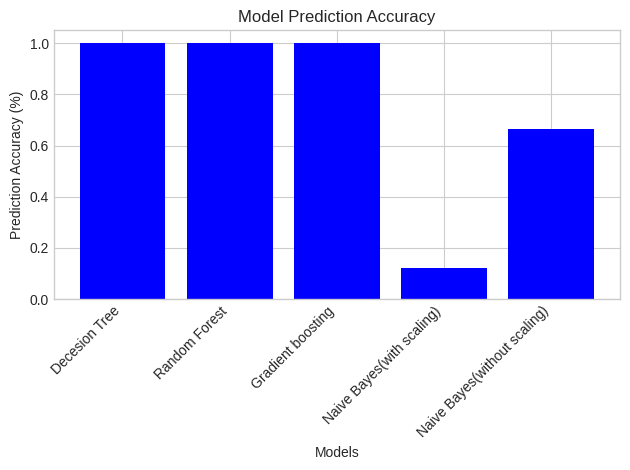
# Update the original weather\_data with the scaled

weather\_data[a.columns] = scaled\_features



7. **Model Selection Comparison Analysis:**

**Bar chart showcasing prediction accuracy of all models:**



**Precision, recall comparison of each model:**

Precision emphasizes the accuracy of positive predictions among all positive predictions. Recall emphasizes the model's ability to correctly identify all positive instances among all actual positive instances.

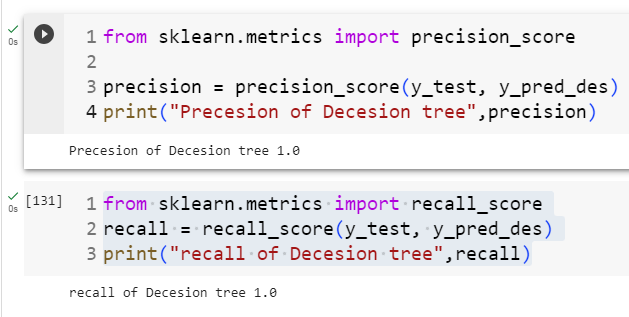
Precision = (True Positive)/(True Positive + False Positive)

Recall = (True Positive)/(True Positive + False Negative)

**Decision Tree:**

Precision = 1

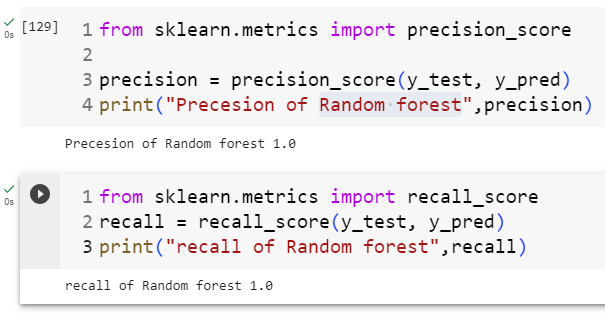
Recall = 1



**Random Forest:**

Precision = 1

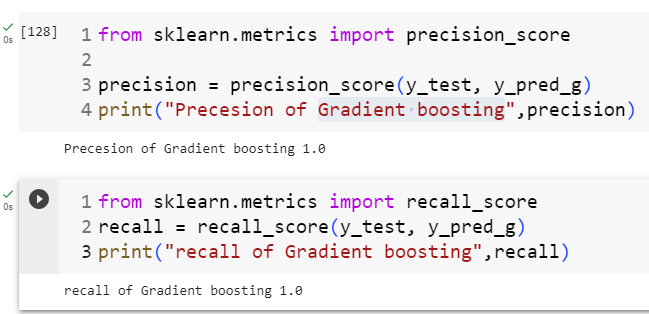
Recall = 1



**Gradient Boosting:**

Precision = 1

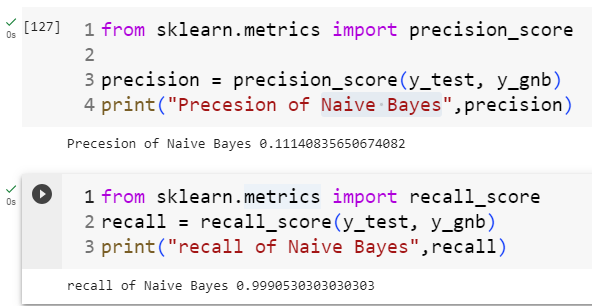
Recall = 1



**Naive Bayes(with Scaling):**

Precision = 0.111

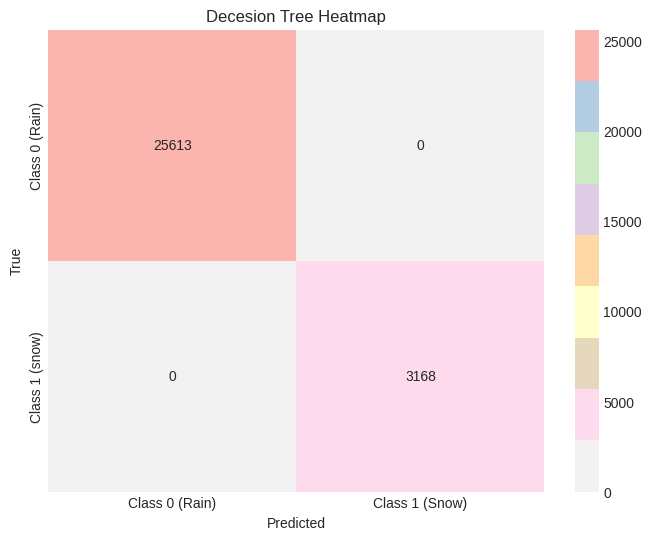
Recall = 0.999



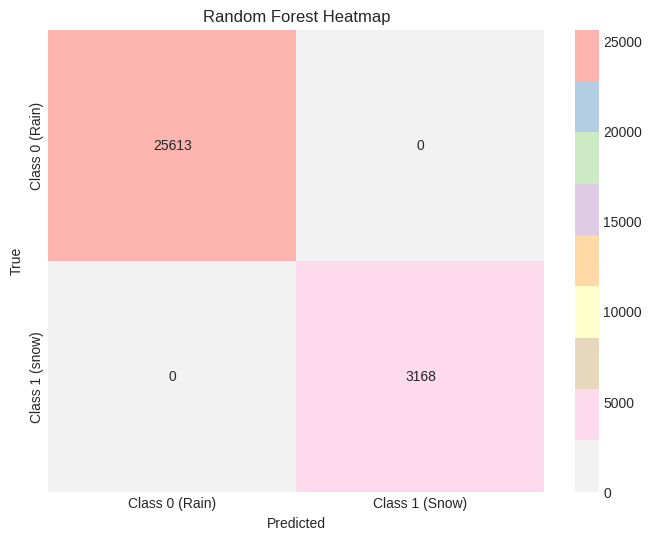
As we can see, the first three models all give extremely accurate predictions, with 0 error so both precision and recall are 1, meaning there are no false positives or false negatives. However, the Naive Bayes model is extremely inaccurate in this case, with a very low precision meaning that there is a huge amount of false positives, which for our data means that it is extremely likely to predict that it is going to rain on days when it will snow, which is likely to inconvenience anyone following this model. This shows that the naive bayes model is not a good algorithm for this particular dataset.

**Confusion matrix:**

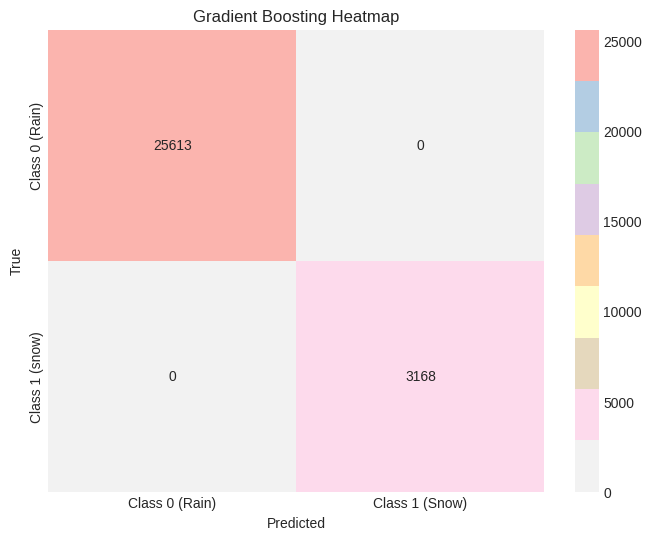
**Decision tree:**



**Random Forest:**



**Gradient Boosting:**



**Naive Bayes(with scaling):**

