KINGDOM OF SAUDI ARABIA
King Abdutylaziz University
FACULTY OF COMPUTING
& INFORMATION TECHNOLOGY

المملكة العربية السعودية
جامعة الملك عبدالعزيز
كلية الحاسبات
وتقنية المعلومات

Predicting Weather Condition

CPIT440 Project Report

Section AAC

Group 6

Proposed to

Dr. Manal Kalkatawi

Prepared by:

Jomana Sameer Sayadi **(Leader)**	2005725
Rana Mohammed Alshehri	2005625

Table of Contents

1. Introduction

Weather prediction is a crucial field of study and practice that endeavors to forecast the atmospheric conditions and phenomena that will prevail at a specific location or region over a given period of time in the future. This prediction is based on analyzing various factors, including temperature, humidity, air pressure, wind speed, and other atmospheric variables.

2. Problem Statement

2.1. Problem that Project Addresses.

The Weather Prediction project addresses a critical challenge in the field of meteorology — the need for heightened accuracy in weather forecasts to mitigate the impact of unforeseen weather events. Much like the issue of false-negative cases in fire prediction, inaccuracies in weather forecasts can lead to significant losses and disruptions across diverse sectors.

2.2. Project Goals

The primary objective of the Weather Prediction project is to develop a model that significantly enhances the accuracy of weather forecasts. By focusing on accurate prediction, the results are expected to be accurate and reliable so that they can be used to make informed decisions about activities and plans.

3. Data Exploration and Visualization

3.1. Which data mining algorithm you should use?

Classification.

3.2. What is the data type?

Numerical and Categorical.

3.3. Are there any splits (Train-test)?

No, our dataset is not splitted.

3.4. Where do you get the data?

The data is available on Kaggle.

3.5. Provide description of the dataset and comment on the following:

3.5.1. What is the number of the instances?

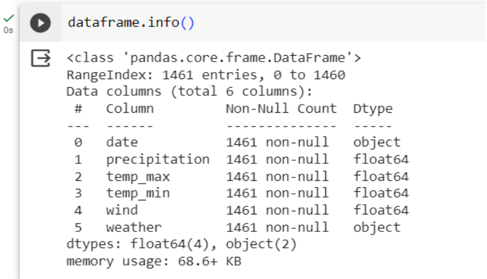
The dataset includes 1461 instances.

3.5.2. Are there any Missing values?

There are no missing values.

3.5.3. What is the Number of the Attributes? What is the Data Type of Each Attribute?

The dataset has 6 attributes. The "precipitation" , "temp_min", "temp_min" and "wind" are numerical, and the "weather" attribute is categorical.



```
dataframe.info()
```

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

Figure 1 dataset information

3.5.4. Categorical Attributes

We have one categorical attribute which is the class attribute (Weather). And there are the categories of the weather attribute.

3.6. Visualization

3.6.1. Histogram

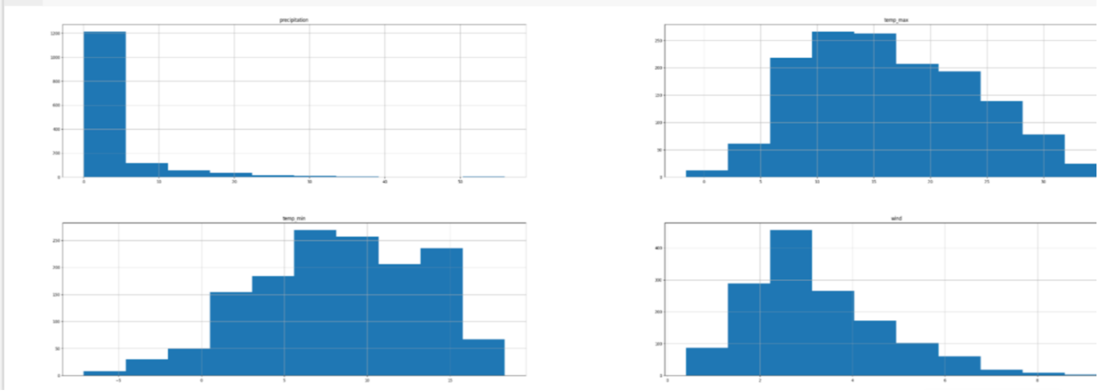


Figure 2 Histogram

3.6.2. Boxplot

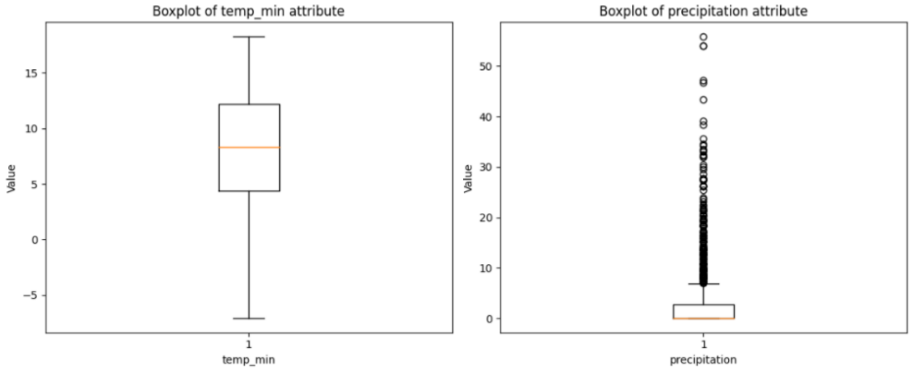


Figure 3 Boxplot

3.6.3. Study the Correlation Between the Attributes to Class Attribute

To study the correlation between attributes and the class attribute, we convert the class attribute type from categorical to numerical.

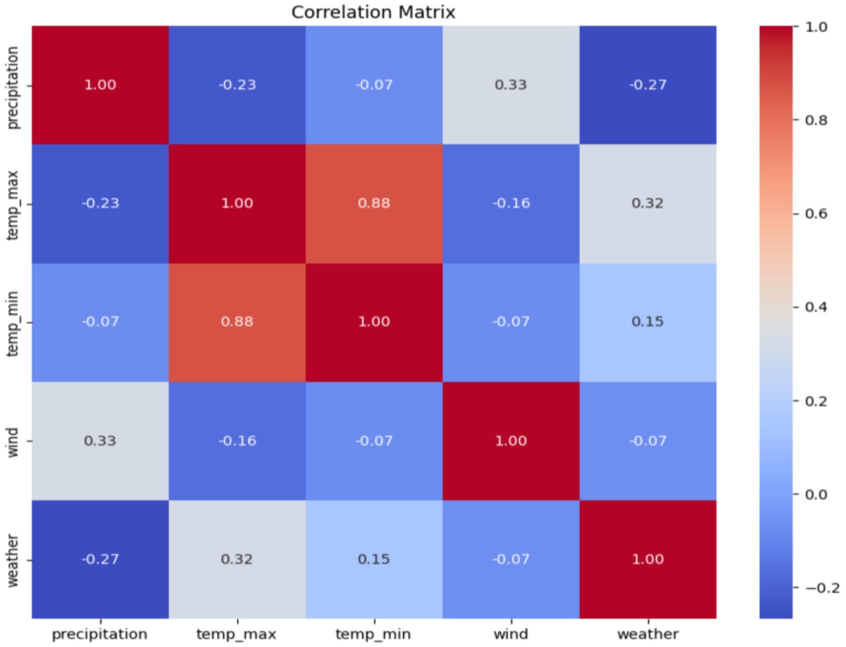


Figure 4 Correlation

Examples for Positive Correlation:
Temp-min & Weather (0.15).
Temp-max & Weather (0.32).

Examples for Negative Correlation:
Precipitation & Weather (-0.27).
Wind & Weather (-0.07) .

4. Data Preprocessing:

4.1. Details About What You Have Done for Preparing Data.

During the data preparation phase, essential measures were implemented to optimize the dataset for machine learning models, focusing on quality and relevance:

- **Column Removal:** The "date" column was eliminated from the dataset, as its inclusion was deemed unnecessary for the specific objectives of the weather prediction task.

- **Missing Values Check:** A meticulous examination was conducted to detect and address any missing values within the dataset. Notably, no missing values were identified, ensuring the dataset's completeness.

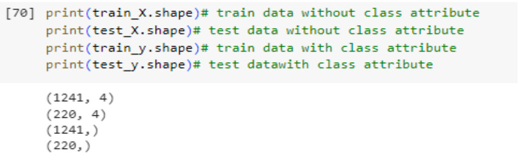
- **Categorical to Numerical Conversion:** the class attribute "weather" type was converted from categorical to numerical.

- **Duplicate Instance Removal:** Instances with identical data points were pinpointed and subsequently removed from the dataset. This step was crucial to uphold data integrity, prevent biases in model training.

4.2. Train/Test Splits.

. Train/test splits the dataset is splitted with 85:15 ratio into two sets:

- Train: 1241 samples/entries
- Test: 220 samples/entries



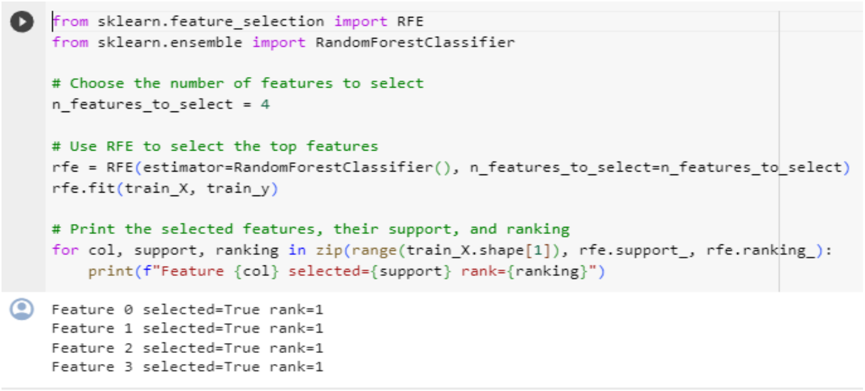
```
[70] print(train_X.shape)# train data without class attribute
     print(test_X.shape)# test data without class attribute
     print(train_y.shape)# train data with class attribute
     print(test_y.shape)# test datawith class attribute

     (1241, 4)
     (220, 4)
     (1241,)
     (220,)
```

Figure 5 Split Train/Test Data

4.3. Feature Selection.

Since all four attributes are crucial for predicting the class attribute, we will utilize all of them as input features for our model. Based on the RFE with a RandomForestClassifier, features 0, 1, 2, and 3 are important for the classification task.



```
from sklearn.feature_selection import RFE
from sklearn.ensemble import RandomForestClassifier

# Choose the number of features to select
n_features_to_select = 4

# Use RFE to select the top features
rfe = RFE(estimator=RandomForestClassifier(), n_features_to_select=n_features_to_select)
rfe.fit(train_X, train_y)

# Print the selected features, their support, and ranking
for col, support, ranking in zip(range(train_X.shape[1]), rfe.support_, rfe.ranking_):
    print(f"Feature {col} selected={support} rank={ranking}")

Feature 0 selected=True rank=1
Feature 1 selected=True rank=1
Feature 2 selected=True rank=1
Feature 3 selected=True rank=1
```

Figure 6 Feature selection

5. Models Training

5.1. Algorithms That You Have Used

To address the weather prediction task effectively, we utilized a trio of powerful machine learning algorithms: Logistic Regression, Gradient Boosting, and Random Forests. Each algorithm contributes unique strengths, collectively offering a comprehensive approach to classification challenges. This diverse ensemble aims to enhance the accuracy and reliability of our weather predictions by leveraging the distinct capabilities of these models.

1. **Logistic Regression Model:**

   Logistic Regression model calculates the probability of an instance belonging to a class using a sigmoid function. It provides interpretable coefficients and assumes a linear relationship between features and predicted probabilities.

2. **Gradient Boosting Model:**

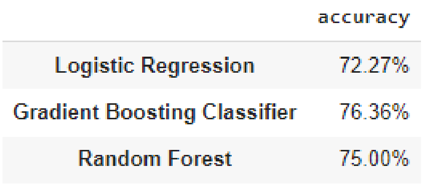
   Gradient Boosting is an ensemble learning technique that builds a series of weak learners sequentially to correct errors made by the previous ensemble. It achieves high predictive accuracy and captures complex relationships in the data.

3. **Random Forests Model**:

   Random Forests is an ensemble learning method that builds multiple decision trees using bootstrap sampling and feature randomization. It combines their predictions through voting (for classification) or averaging (for regression), providing robustness to overfitting and noisy data.

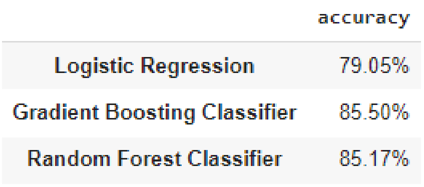
5.2. Cross Validation

Cross-validation is a technique for assessing a model's performance by systematically splitting the dataset into subsets for training and testing. It helps provide a more reliable estimate of the model's generalization ability, crucial for avoiding overfitting and understanding overall performance.



	accuracy
Logistic Regression	72.27%
Gradient Boosting Classifier	76.36%
Random Forest	75.00%

Figure 8 Accuracy Before Cross Validation



	accuracy
Logistic Regression	79.05%
Gradient Boosting Classifier	85.50%
Random Forest Classifier	85.17%

Figure 7 Accuracy After Cross Validation

6. Models Evaluation

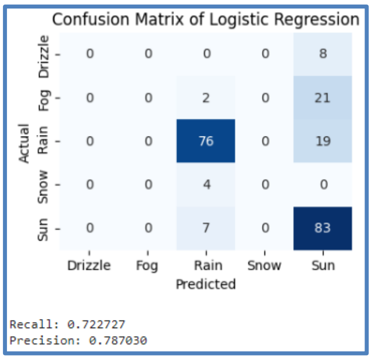
6.1. Performance Metrics You Have Used

Since our data is imbalanced, we prioritize using the recall metric. We choose the "weighted" average (since we have multiclass) to calculate recall for each model. This helps us evaluate how well the models identify positive instances, considering the imbalance in our data.

We calculated recall & precision in each model for test data:

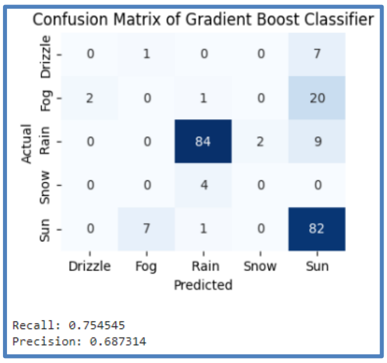
- **Logistic Regression Model:** Recall= 0.722727 , Precision= 0.787030
- **Gradient Boosting Models:** Recall= 0.754545 , Precision= 0.687314
- **Random Forest Classifier:** Recall= 0. 754545 , Precision= 0.707895

Confusion Matrix, Recall & Precision of each Model:



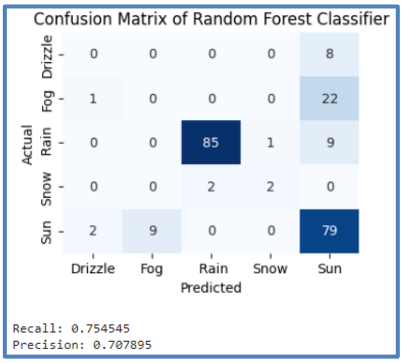
```
Recall: 0.722727
Precision: 0.787030
```

Figure 10 Logistic Regression Model



```
Recall: 0.754545
Precision: 0.687314
```

Figure 9 Gradient Boosting Model



```
Recall: 0.754545
Precision: 0.707895
```

Figure 11 Random Forests Model

6.2. Comparison of Models' Performance Using Table or Plots .

Here are a table that shows different values of accuracy score of each model:

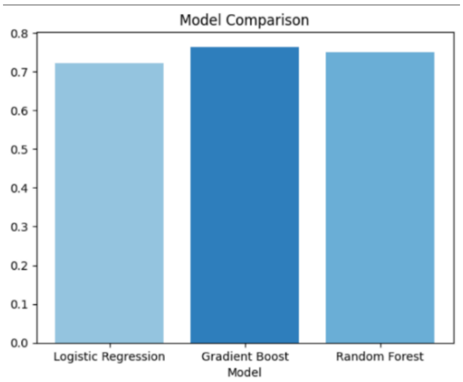


Figure 12 Comparison of Models' Performance

6.3. Which Model You Select and Why?

We chose the Gradient Boosting Model because it exhibited the highest accuracy score among the considered models. This decision is especially influenced by the imbalanced dataset, where correctly identifying positive instances is crucial. The model's strength in capturing complex relationships contributes to its superior performance.

7. Tools

Tool: Google Colab (programming language: python)

The used libraries:

pandas	seaborn	matplotlib.pyplot	train_test_split	StandardScaler
LabelEncoder	RFE	RandomForestClassifier	LogisticRegression	mean_absolute_error
GradientBoostingClassifier	cross_val_score	confusion_matrix	recall_score	

8. Difficulties You Have Faced and Challenges

We had limited expertise in managing data using Python initially. Additionally, we encountered challenges in identifying the most suitable model for our dataset. Given the relatively small size of our dataset, achieving high accuracy posed difficulties. However, through consistent practice, we endeavored to implement the concepts learned in our course.

9. Future Work

We would like to improve our model moreover and apply the model to a large dataset and measure the model's performance.

10. Work Division

A table describing the work division among the group members.

Jomana Sayadi	Rana Alshehri
50%	50%

11.Conclusion

In conclusion, our Weather Prediction project aimed to improve forecast accuracy using machine learning models, including Logistic Regression, Gradient Boosting, and Random Forests. We analyzed a Kaggle dataset with 1461 instances, performed preprocessing, and selected the Gradient Boosting Model for its superior recall in handling the imbalanced data. Our work utilized Python in Google Colab, facing initial challenges but overcoming them through even workload distribution.

12. References

GradientBoostingClassifier. Retrieved from scikit learn: https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.GradientBoostingClassifier.html

LogisticRegression. Retrieved from scikit learn: https://scikit-learn.org/stable/modules/generated/sklearn.linear_model.LogisticRegression.html

RandomForestClassifier. Retrieved from scikit learn: https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.RandomForestClassifier.html

WEATHER PREDICTION. Retrieved from kaggle: https://www.kaggle.com/datasets/ananthr1/weather-prediction