First Semester 2025/2026

Faculty of Engineering and Technology

Electrical and Computer Engineering Department

ENCS5341, MACHINE LEARNING AND DATA SCIENCE

Assignment#3

Prepared by:

Name: Abdulrahman Atyani Id:1221808

Name: Tareq Nazzal Id:1221899

Supervised by: Dr. Yazan Abu Farha

Section: 1

January-2026

Table of Contents

Introduction

A . Task Description

In this project, the goal is to predict the **weather condition** associated with an image using a set of contextual attributes. Specifically, each example includes information about the **country**, **time of day**, and **season**, and the task is to use these features to classify the weather into one of four categories: **Sunny, Cloudy, Snowy, or Rainy**.

This problem is formulated as a **supervised multi-class classification task**, since the target variable contains more than two possible classes. Each row in the dataset represents a single photograph with its corresponding metadata and a ground-truth weather label. Before training the models, the dataset was cleaned by removing entries with missing values, and the data was split into training and test sets using a stratified approach to maintain the original class distribution.

B. Models Explored

To study the effectiveness of different machine learning approaches, several models were explored and compared.

As a baseline, **K-Nearest Neighbours (KNN)** was used with two different values of kkk (k = 1 and k = 3). KNN was chosen because of its simplicity and its ability to exploit similarity between samples once categorical features are transformed using one-hot encoding. Testing multiple values of **k** helped assess the trade-off between sensitivity to noise and generalization. The best-performing KNN configuration, based on weighted F1-score, was selected as the baseline model.

To improve upon this baseline, two more advanced models were implemented. The first was a **Support Vector Machine (SVM)** with a Gaussian (RBF) kernel. This model was selected because it can capture non-linear relationships between features. Class imbalance was addressed by using balanced class weights, and key hyperparameters were tuned using cross-validation.

The second model was a **Random Forest classifier**, which is an ensemble-based approach that combines multiple decision trees. Random Forest was chosen for its robustness and its ability to model complex feature interactions. Similar to the SVM, class imbalance was handled through class weighting, and several hyperparameters such as the number of trees and tree depth were tuned.

C. Evaluation Metrics

Multiple evaluation metrics were used to assess model performance. **Accuracy** was reported to provide an overall measure of correct predictions; however, since the dataset is imbalanced (with "Sunny" being the dominant class), accuracy alone is not sufficient. Therefore, the **weighted F1-score** was used as the primary evaluation metric, as it balances precision and recall while accounting for class frequencies.

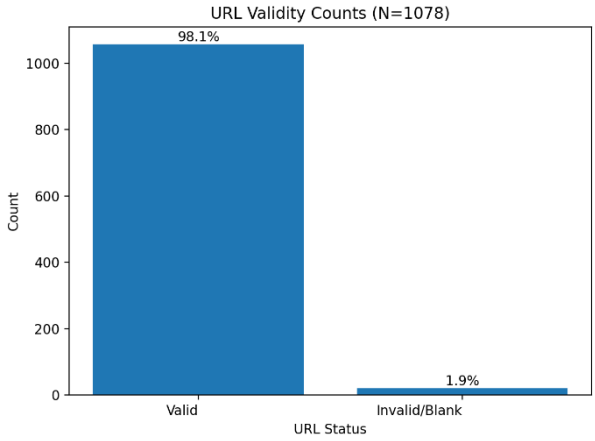
In addition, **confusion matrices** were analysed to better understand the types of errors made by each model and to identify which weather categories were more difficult to predict.

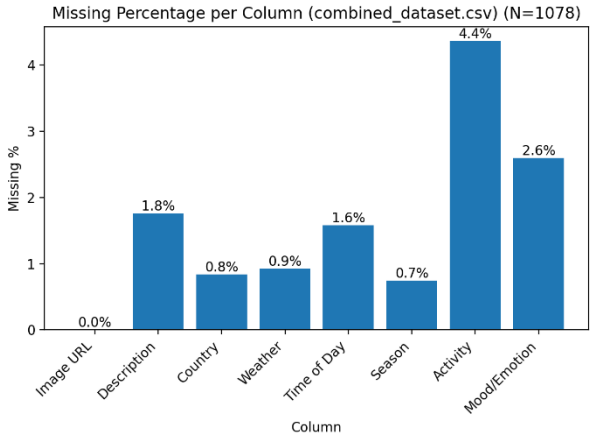
Exploratory Data Analysis

A. Exploring data

We made some exploring to the combined_dataset.csv file which contain all the data combined in one csv file, which contained 1078 sample, then we found the following stats and information about the data:

1- Missing values and URL validity:

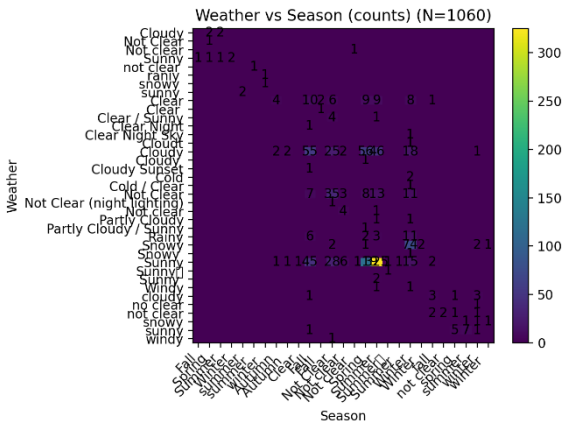


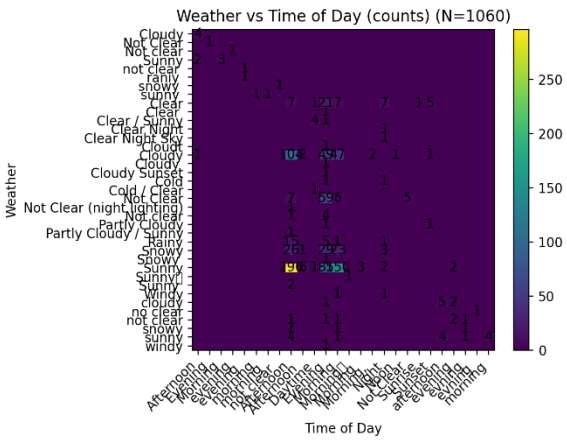


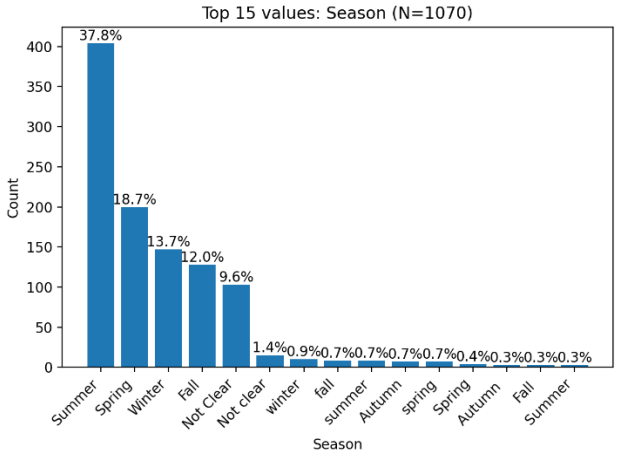
These stats shows that the dataset has a percentage of missing data, which represented in blank cells.

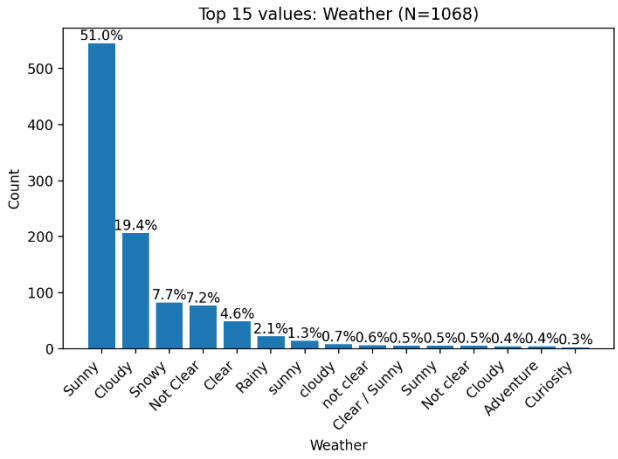
2- data distribution per feature:

Here are some of the data distribution and heatmaps which give a brief understanding about dirty the data is, and tell us how some column names in the combined dataset before cleaning are repetitive due to a small differences in words characters.









B. Data Preprocessing

After combining all csv files in one file (combined_dataset.csv), we started to preprocess and clean the data to prepare it to be suitable for our machine learning models, preprocessing took the following steps:

1. After deciding that **Weather** would be the prediction target, and using **Country, Season, and Time of Day** as the main input features, we began by **cleaning the labels** to reduce noise. In particular, we removed any rows where **Weather** or **Season** was labeled **"Not Clear"**, since these values do not provide reliable supervision for training. This filtering step reduced the dataset from **1,078 samples to 807 samples**, producing a cleaner and more consistent set for modeling.

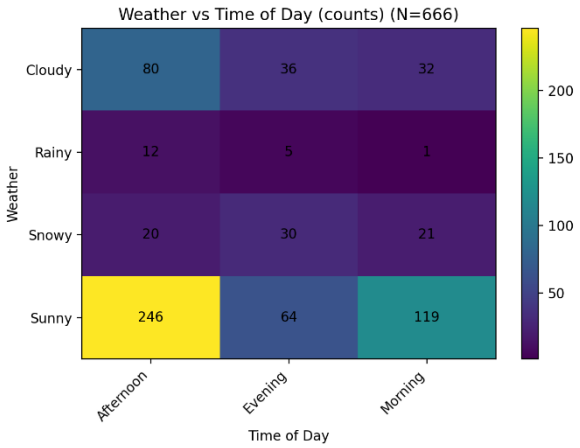
2. Next, we needed to ensure that every remaining sample could be linked to an actual image. We therefore generated an **ID** for each sample and used it as a stable reference for the image downloading stage. The script attempted to download the image from the provided URL for every sample and saved each file using the **corresponding ID as the filename**. This approach guarantees that each downloaded image can be matched correctly to its row in the dataset. Because some URLs were invalid or did not provide a usable image, not all downloads succeeded. After completing the download stage, we obtained **667 images**, each named by the ID of its matching sample.

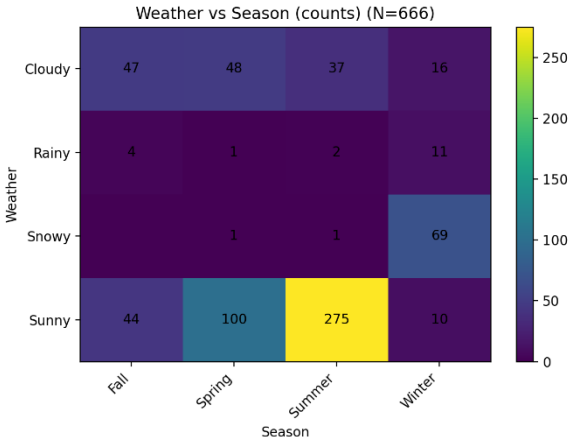
3. Throughout the pipeline, we kept detailed logs to track data loss and diagnose issues. Specifically, we recorded:

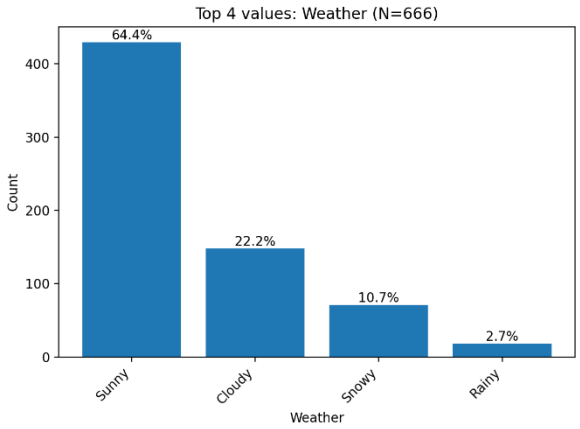
- **failed files** that could not be read or did not contain the required columns during the merging step, and
- **failed image downloads** (including the URL and failure reason) during the image collection step.

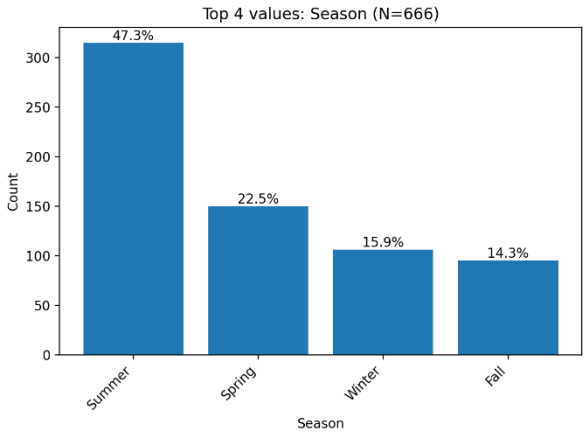
These logs are important for transparency, reproducibility, and understanding how the dataset size changes across stages.

4. Finally, we prepared the final dataset (the **676 samples with available images**) for machine learning by applying **one-hot encoding** to the categorical variables. We encoded **Country, Season, Time of Day, and Weather** to transform them into a numeric format that models can use effectively. We intentionally limited encoding to these columns because they are the most relevant to our objective and keeping the feature space smaller helps reduce complexity and avoid unnecessary sparsity.









We can see the results clearly after cleaning, cleaner labelling, less noise and better data distribution.

Experiments and Results

A .Dataset Characteristics

The dataset contains **676 samples** distributed across four weather classes, the dataset is **strongly imbalanced**, with *Sunny* representing the majority of samples (441 instances), while *Rainy* is severely underrepresented (19 instances). This imbalance motivates the use of weighted evaluation metrics and class-balanced models.

B. Baseline and Proposed Models

As a starting point, K-Nearest Neighbours (KNN) was used as the baseline model with two neighbourhood sizes, k=1k = 1k=1 and k=3k = 3k=3. KNN was selected because it is simple and intuitive, making it a useful reference for evaluating whether more complex models provide real improvements. With k=1k = 1k=1, the model achieved an accuracy of approximately 0.69 and a weighted F1-score of 0.67. However, this setting proved to be sensitive to noise, since each prediction depends on a single nearest neighbour, leading to unstable predictions for minority classes such as *Rainy* and *Cloudy*. Increasing the neighbourhood size to k=3k = 3k=3 improved performance, with the model achieving an accuracy of 0.71 and a weighted F1-score of 0.68. By considering multiple neighbours, the model became more robust and less affected by individual noisy samples, and therefore **KNN (k = 3)** was selected as the baseline model for comparison.

To improve upon the baseline, two more advanced models were explored: a Support Vector Machine (SVM) with a Gaussian (RBF) kernel and a Random Forest classifier. The SVM was chosen for its ability to model non-linear relationships between contextual features such as country, season, and time of day. Balanced class weights were applied to address class imbalance, and hyperparameters including the regularization parameter CCC and kernel parameter γ\gammaγ were tuned using grid search and stratified cross-validation. The tuned SVM achieved the highest cross-validation weighted F1-score and reached a test accuracy of approximately 0.70 with a weighted F1-score of 0.70. The Random Forest model, selected for its ability to capture complex feature interactions, achieved a similar test accuracy but obtained the highest weighted F1-score on the test set (≈ 0.705). This indicates that while both proposed models slightly underperformed the baseline in terms of accuracy, they provided more balanced and reliable performance across all weather classes, particularly when compared to the baseline KNN model.

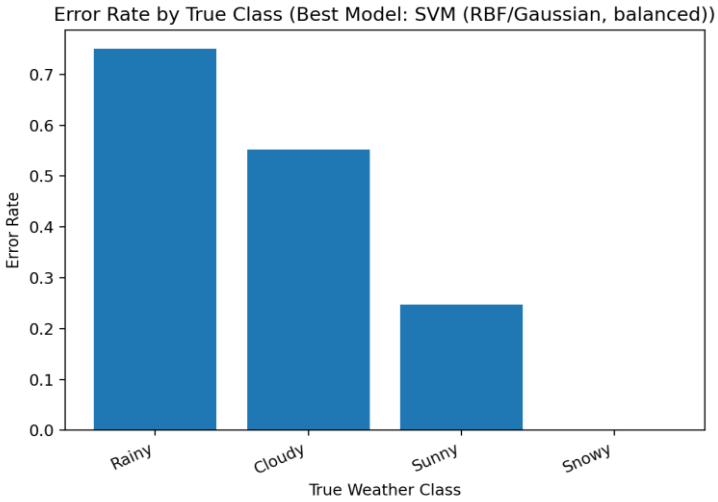
C. Model Performance Comparison

Although the KNN baseline achieved the highest raw accuracy, both the **SVM** and **Random Forest** models achieved **higher weighted F1-scores**, indicating better handling of class imbalance. The SVM achieved the best cross-validation F1-score, while the Random Forest achieved the highest test F1-score.

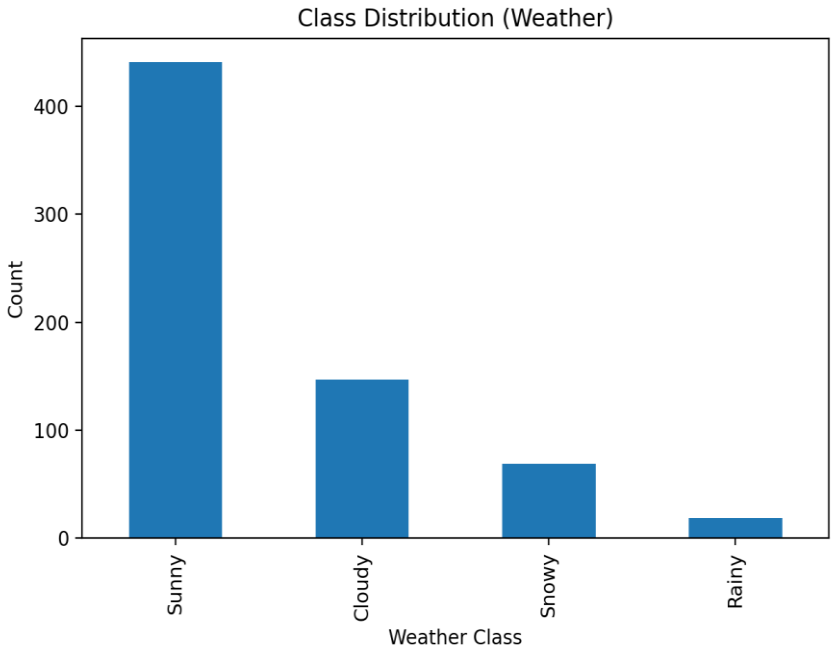
These results highlight that accuracy alone is not sufficient in imbalanced datasets and that weighted metrics provide a more reliable evaluation.

D. Error Analysis

The classification errors of the best-performing model (SVM) were analysed in detail. As shown in **below figure**, the *Rainy* class exhibits the highest error rate (75%), followed by *Cloudy*. In contrast, *Snowy* achieved perfect classification, and *Sunny* showed relatively strong performance.



The most common misclassification patterns involved confusion between *Sunny* and *Cloudy*, suggesting that these classes share similar contextual attributes such as season and time of day. The high error rate for *Rainy* is largely attributed to its limited number of training samples, as also indicated by the class imbalance in **this figure**.



Analysis

To analyse the behaviour of the best-performing model, I focused on the **SVM (RBF, balanced)** because it achieved the highest cross-validation weighted F1-score. After training and selecting the best hyperparameters, I evaluated the model on the held-out test set and then performed an error analysis using the test predictions. Concretely, I created a table that contains each test example with its **true label**, **predicted label**, and whether the prediction was correct. I also saved all **misclassified examples** into a separate file (misclassified_examples.csv) so I could inspect them directly. In addition, I computed a **confusion matrix** and measured the **error rate per true class** to identify which weather categories were most problematic. Finally, I grouped the test errors by feature values (Country, Time of Day, Season) to look for patterns such as whether certain conditions consistently produce more mistakes.

The results show clear and meaningful patterns. First, the model struggles most with the minority classes. **Rainy** had the highest error rate: only **25% accuracy** on the test set (3 out of 4 rainy examples were misclassified), which is expected because Rainy is extremely underrepresented in the dataset (only 19 total samples). **Cloudy** was also difficult, with an accuracy of about **44.8%**, and it was frequently confused with **Sunny**. The confusion matrix confirms this: the largest error pairs were **Sunny → Cloudy (19 cases)** and **Cloudy → Sunny (13 cases)**. This suggests that using only contextual metadata (country, season, time of day) is often not enough to separate these two labels, since many sunny and cloudy photos share similar metadata. On the other hand, **Snowy** was classified perfectly (**100% accuracy**, 14/14 correct), which likely indicates that snowy conditions correlate strongly with certain seasons and locations (e.g., winter-related contexts), making them easier to recognize even without image pixels. When analysing errors by feature values, **Spring** showed the highest error rate among seasons (≈0.486), and **Afternoon/Evening** were harder than Morning (≈0.333 vs. 0.233). However, some "hardest countries" had an error rate of 1.0 but appeared only 1–2 times in the test set, meaning those values are **not reliable conclusions** and reflect small sample sizes rather than consistent model failure. Overall, the error analysis indicates that the main limitations come from **class imbalance** (especially for Rainy) and **overlap between Sunny and Cloudy** under the same contextual attributes. This supports the idea that future improvement would likely require either more balanced data (more Rainy examples) or incorporating image information to capture visual cues that metadata cannot represent.

Conclusions and Discussion

In this task, the goal was to predict weather conditions using contextual information such as country, time of day, and season. Several machine learning models were implemented and compared, including K-Nearest Neighbours as a baseline and more advanced models such as Support Vector Machines and Random Forests. The experiments showed that while the KNN baseline achieved the highest raw accuracy, the more advanced models provided better class-balanced performance, as reflected by higher weighted F1-scores. In particular, the SVM achieved the best cross-validation performance, while the Random Forest achieved the highest weighted F1-score on the test set. These results demonstrate that evaluating models using multiple metrics is important, especially in imbalanced classification tasks.

Despite these encouraging results, several limitations were identified. The most significant limitation is the **class imbalance** in the dataset, particularly for the *Rainy* class, which contains very few samples. This imbalance led to high error rates for minority classes and influenced the behaviour of all models. In addition, the models relied only on **contextual metadata**, which proved insufficient for distinguishing visually similar weather conditions such as *Sunny* and *Cloudy*. Furthermore, some feature values (e.g., certain countries) appeared only a few times in the dataset, making it difficult to draw reliable conclusions about their impact. Although weighted F1-score helps mitigate imbalance effects, it cannot fully compensate for the lack of representative data in minority classes.

Several directions could be explored to improve this work in the future. Collecting a more **balanced dataset**, especially with additional *Rainy* examples, would likely improve performance. Another promising direction is to incorporate **image-based features** using computer vision models, which could help capture visual cues that contextual metadata alone cannot provide. Additionally, experimenting with more advanced ensemble methods or cost-sensitive learning techniques could further improve performance on minority classes. Overall, this project highlights both the strengths and limitations of metadata-based weather classification and provides a solid foundation for future improvements.