Climate Risk Classification Report Using Machine Learning

By: Epifanio Solano, Nolan

Table Of Contents

Motivation

Climate change is one of the most urgent challenges we face today. While we often talk about it globally, its effects are not distributed evenly. Some cities are experiencing faster and more extreme temperature changes than others.

Our project aims to dig into this local perspective. We want to determine which cities are most at risk using historical temperature data from Berkeley Earth, which tracks monthly temperatures from the 1700s to 2013. By engineering features that capture temperature trends, variability, and change rates, we will use machine learning models to classify cities into different climate risk levels.

Unlike previous work that focuses on global temperature averages, we want to zoom in and highlight city-level patterns, providing insights that could be more directly useful for urban planning, policy decisions, or raising public awareness.

Prior Research

The dataset used in this study was previously analyzed by researchers at the Goddard Institute for Space Studies (GISS) to monitor and evaluate global surface temperature changes. Their work focused on statistical methods rather than machine learning techniques. Specifically, they used satellite-observed night lights to identify and correct for urban heat biases in temperature measurement stations, ensuring more accurate trend analysis. They combined sea surface temperature records with meteorological station data and tested different methods of integrating ocean data, particularly in polar regions where direct observations are sparse. To improve the data in their temperature graphs, they applied simple 12-month

and multi-year running means. Their findings affirmed that the global warming trend has continued to rise at a consistent rate over recent decades, despite short-term fluctuations caused by El Niño and La Niña cycles.

Goals

Our goal is to build on the prior research by incorporating machine learning models to classify cities based on their risk level due to climate change. While the original GISS study relied solely on statistical analysis to assess global temperature trends, our approach is distinct in that it applies supervised learning techniques to identify patterns and make predictive classifications at the city level. This allows us to go beyond global averages and address localized climate vulnerability, offering a more targeted and actionable understanding of climate risk. By leveraging features such as temperature trends and variability, our model aims to identify cities that may require urgent adaptation measures, an advancement not explored in the original statistical approach.

Dataset

The dataset used in our project is a subset of the Berkeley Earth Surface Temperature dataset, focusing specifically on city-level temperature data from around the world. It contains 8,599,212 entries spanning the years 1743 to 2013, covering 3,448 cities globally. Each entry includes the following columns:

- **dt:** Date of the temperature recording.
- **AverageTemperature:** The monthly average land temperature (in **°C**).
- **AverageTemperatureUncertainty:** The uncertainty associated with the temperature measurement.
- **City:** The name of the city.
- **Country:** The country in which the city is located.
- **Latitude:** Geographic latitude of the city.
- **Longitude:** Geographic longitude of the city.

This dataset provides detailed temperature data over time for thousands of cities around the world, which makes it useful for identifying patterns and trends. Because it includes both geographic and historical information, it allows us to train machine learning models that can classify cities based on their climate risk, an approach that was not explored in the previous research.

Exploratory Data Analysis

We used **head()** to inspect the structure of the dataset and confirmed the presence of relevant columns such as **AverageTemperature, City, Country,** and geographic coordinates (**Latitude, Longitude**). The **info()** output showed over 8.5 million entries across seven columns, with some missing values in temperature-related fields.

Using **count()** and **isnull().sum()**, we found that while location data is largely complete, both **AverageTemperature** and **AverageTemperatureUncertainty** contain 364,130 missing values, particularly concentrated in earlier years where measurements were more sparse.

The **describe()** method provided summary statistics for the temperature-related columns. The global average temperature is approximately 16°C, with a standard deviation of 10°C, reflecting the wide range of climate conditions across cities and over time.

Figure (1): First five entries of the dataset (**head()**).

	dt	AverageTemperature	AverageTemperatureUncertainty	City	Country	Latitude	Longitude
0	1743-11-01	6.068	1.737	Århus	Denmark	57.05N	10.33E
1	1743-12-01	NaN	NaN	Århus	Denmark	57.05N	10.33E
2	1744-01-01	NaN	NaN	Århus	Denmark	57.05N	10.33E
3	1744-02-01	NaN	NaN	Århus	Denmark	57.05N	10.33E
4	1744-03-01	NaN	NaN	Århus	Denmark	57.05N	10.33E

Figure (2): Dataset information, including total rows, column names, and data types (**info()**)

```
RangeIndex: 8599212 entries, 0 to 8599211
Data columns (total 7 columns):
 #   Column                         Dtype
---  ------                         -----
 0   dt                             object
 1   AverageTemperature             float64
 2   AverageTemperatureUncertainty  float64
 3   City                           object
 4   Country                        object
 5   Latitude                       object
 6   Longitude                      object
dtypes: float64(2), object(5)
memory usage: 459.2+ MB
```

Figure (3): Summary of non-null entries per column (**count()**) and missing values per column (**isnull().sum()**).

```
dt                                0
AverageTemperature           364130
AverageTemperatureUncertainty    364130
City                              0
Country                           0
Latitude                          0
Longitude                         0
```

```
dt                                8599212
AverageTemperature                8235082
AverageTemperatureUncertainty     8235082
City                              8599212
Country                           8599212
Latitude                          8599212
Longitude                         8599212
```

Figure (4): Statistical summary of Average Temperature and Average Temperature Uncertainty (**describe()**).

	AverageTemperature	AverageTemperatureUncertainty
count	8.235082e+06	8.235082e+06
mean	1.672743e+01	1.028575e+00
std	1.035344e+01	1.129733e+00
min	-4.270400e+01	3.400000e-02
25%	1.029900e+01	3.370000e-01
50%	1.883100e+01	5.910000e-01
75%	2.521000e+01	1.349000e+00
max	3.965100e+01	1.539600e+01

Data Preprocessing

Before training our machine learning model, we conducted preprocessing on the global land temperature dataset to transform it into a structured format. The raw dataset consisted of over 8.5 million entries with monthly temperature readings for thousands of cities dating back to the 1700s. We began by converting the date column (**dt**) to a datetime format and extracting the **Year** as a new column.

Additionally, the coordinate columns (**Latitude** and **Longitude**) were initially stored as directional strings (e.g., "35.5N", "120.9W"). We converted these to numeric values, enabling accurate spatial analysis and visualization on geographic plots.

A key step in our preprocessing was grouping the data by city. This reduced the dataset from over 8 million rows to just 3,448, one row per city, each summarizing that city's long-term climate data.

We then engineered a set of features designed to capture key climate indicators: the average temperature (**AverageCityTemperature**), temperature volatility (**StandardDeviation**), minimum and maximum observed temperatures, and temperature range (TemperatureRange). We also calculated the warming trend (**RateOfChange**) using linear regression to determine how quickly each city is heating up.

Finally, we added a new feature called **YearsUntil1.5°C**, which estimates how long it will take each city to reach a 1.5°C rise in temperature based on its warming rate. This is based on the Paris Agreement's goal to keep global warming below 1.5°C by around 2050. With this and other features, our cleaned dataset was ready for assigning climate risk labels and training machine learning models.

Figure (5): The finalized dataset after feature engineering

RateOfChange	AverageCityTemperature	StandardDeviation	MinTemperature	MaxTemperature	TemperatureRange	YearsUntil1.5TemperatureChange
0.010225	13.393040	3.856365	5.062	21.913	16.851	146.705257
0.010910	9.132275	6.191364	-7.720	22.812	30.532	137.483358
0.009922	8.018004	6.369246	-7.816	20.883	28.699	151.178383
0.007759	26.786108	1.054214	24.009	30.036	6.027	193.335143
0.013546	25.237664	8.351773	7.964	38.531	30.567	110.736204
...	...	...	...	...	...	...
0.009922	8.018004	6.369246	-7.816	20.883	28.699	151.178383
0.007166	13.599576	6.821893	-0.966	26.774	27.740	209.317852
0.007639	10.250976	7.469154	-5.475	24.577	30.052	196.372628
0.015857	1.611088	13.070158	-28.720	22.229	50.949	94.593668
0.014836	6.204619	14.127610	-21.688	26.092	47.780	101.102106

Climate Risk Labeling

To classify cities by their climate risk, we created a labeling system based on five key features: the rate of temperature increase (RateOfChange), average temperature (AverageCityTemperature), temperature volatility (StandardDeviation), maximum observed temperature (MaxTemperature), and the estimated years until a 1.5°C temperature rise (YearsUntil1.5TemperatureChange). Each of these features captures a different aspect of climate risk, such as how fast a city is warming, how hot it already is, how unstable its climate has been, and how soon it may exceed the 1.5°C global warming threshold outlined in the Paris Agreement.

We chose specific thresholds for each feature based on climate science and the statistical distribution of our data. For example, we labeled cities as **High Risk** if their warming trend exceeded 0.015°C per year, which would lead to a 1.5°C increase in less than 100 years. Cities were also flagged as high risk if their **average temperature** was above 26°C or their **maximum recorded temperature** exceeded 34°C, thresholds associated with increased public health risks such as heat stress and strain on infrastructure. High volatility, measured by a **standard deviation** greater than 10°C, also contributed to a high-risk classification. Additionally, cities projected to exceed a 1.5°C rise within 25 years were labeled as high risk due to the urgency of intervention needed.

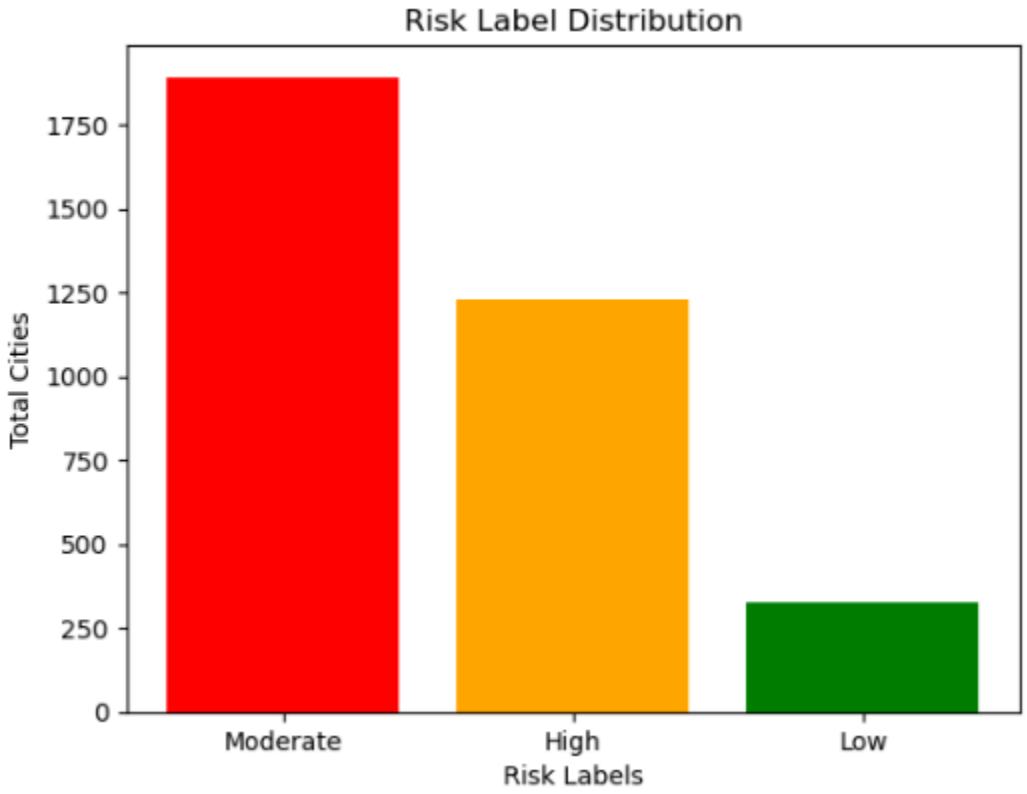
Cities that met more moderate thresholds—such as a warming rate between 0.01–0.015°C per year, average temperatures above 20°C, or years to 1.5°C between 25 and 75 years—were classified as Moderate Risk. All others were labeled Low Risk.

After applying this logic, our final dataset contained:

- 1,227 High Risk cities
- 1,892 Moderate Risk cities
- 329 Low Risk cities

This distribution reflects the reality that many cities are already warming significantly or will likely exceed climate targets within this century, highlighting the urgency of understanding and addressing localized climate risk.

Figure (6): A bar graph showing the distribution of assigned climate risk labels across 3,448 cities. The majority of cities fall into the Moderate Risk category, followed by High Risk, with a smaller number classified as Low Risk based on their temperature trends and climate conditions.



Data Splitting, Label Encoding, and Normalization

To prepare the data for modeling, we first converted the climate risk labels (**Low, Moderate, High**) into numeric values using **LabelEncoder**. This allowed the models to work with the labels more easily. We then split the dataset into 60% training and 40% testing using stratified sampling, so that each risk level stayed balanced in both sets. Since some models like Logistic Regression and Support Vector Machines (SVM) are sensitive to the scale of the input features, we used StandardScaler to normalize the data. We fit the scaler on the training data and used it to transform both the training and testing sets. This made all the features have a similar scale, which helps the models train more effectively.

Model Development and Evaluation

We trained three models to predict climate risk: Logistic Regression, Support Vector Machine (SVM), and Random Forest. To get the best performance, we used **GridSearchCV** to try different settings for each model and used 5-fold cross-validation to check how well they worked. Logistic Regression and SVM were trained using scaled features, while Random Forest was trained on the original features because it doesn't need scaling. Although we looked at several metrics, our main focus was on recall, how well the model identifies cities in each risk category. This was important because missing high-risk or moderate-risk cities could lead to serious consequences in real-world climate planning.

Logistic Regression

We chose Logistic Regression as our first model because it's simple, fast, and works well for multiclass classification problems. It also gives us clear insights into how each feature impacts the prediction. Since it's sensitive to the scale of input features, we normalized the data using **StandardScaler**.

To improve performance, we used **GridSearchCV** with 5-fold cross-validation to tune the hyperparameters. We tested different values of **C** (which controls regularization strength) and tried several solvers like '**lbfgs**', '**saga**', '**sag**', and '**newton-cg**'. The best combination was:

- C = 1
- solver = 'saga'
- multi_class = 'multinomial'
- Penalty = 'l2'

We then tested the model on both the training and testing sets, but our main focus was on recall, we wanted the model to catch as many High Risk and Moderate Risk cities as possible.

Figure(7): A graph showing hyperparameter tuning for Logistic Regression using GridSearchCV. The plot displays how different values of the regularization parameter C impacted model performance across various solvers. The best results were achieved with C = 1 using the saga solver.

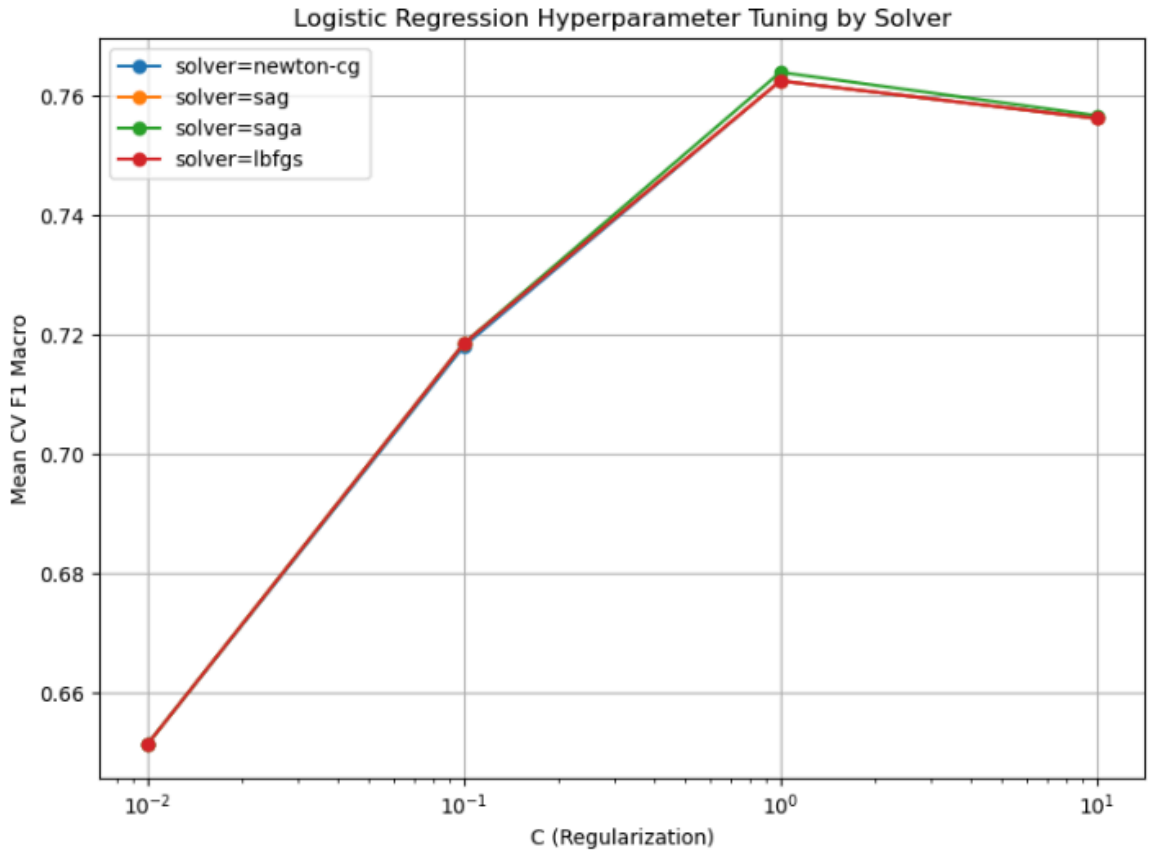


Figure (8): Classification reports for Logistic Regression on the training set and testing set. The model had high recall for Low and Moderate Risk cities, correctly identifying over 80% in both sets. Recall for High Risk cities dropped slightly from 0.69 to 0.66, meaning the model missed a few more in the test set. Still, the model did a good job overall at catching most at-risk cities.

Classification Report for Logistical Regression: Testing Set:

	precision	recall	f1-score	support
Low	0.72	0.83	0.77	491
Moderate	0.52	0.82	0.64	132
High	0.83	0.66	0.73	757
accuracy			0.74	1380
macro avg	0.69	0.77	0.71	1380
weighted avg	0.76	0.74	0.74	1380

Classification Report for Logistical Regression: Testing Set:

	precision	recall	f1-score	support
Low	0.72	0.83	0.77	491
Moderate	0.52	0.82	0.64	132
High	0.83	0.66	0.73	757
accuracy			0.74	1380
macro avg	0.69	0.77	0.71	1380
weighted avg	0.76	0.74	0.74	1380

Support Vector Machines (SVM)

We chose Support Vector Machine (SVM) as one of our models because it works well with complex, non-linear data and performs strongly in classification tasks, especially when the number of features is

small to medium. It's also known for creating clear decision boundaries, which is helpful when trying to separate overlapping risk categories like Moderate and High.

Before training, we scaled all input features using **StandardScaler**, since SVM is sensitive to feature magnitude. We then used **GridSearchCV** to tune two important hyperparameters:

C: controls the trade-off between accuracy and regularization.

gamma: controls how far the influence of a single training point reaches.

We tested a range of values for both **C** and **gamma** using the **RBF** kernel, and found that the best performance came from:

- **C** = 10
- **gamma** = 1

We focused on recall, and the final model performed extremely well. It achieved nearly perfect recall on the training set and maintained very high recall across all three classes in the testing set, especially for High Risk and Moderate Risk cities. This shows the model was able to generalize well without overfitting, making it our strongest performer for recall-focused classification.

Figure (9): SVM hyperparameter tuning using the RBF kernel. The plot shows the effect of different values of the regularization parameter C and kernel coefficient gamma on mean cross-validation accuracy. The best performance was achieved with gamma=1 and C=10, where the model reached nearly 96% accuracy.

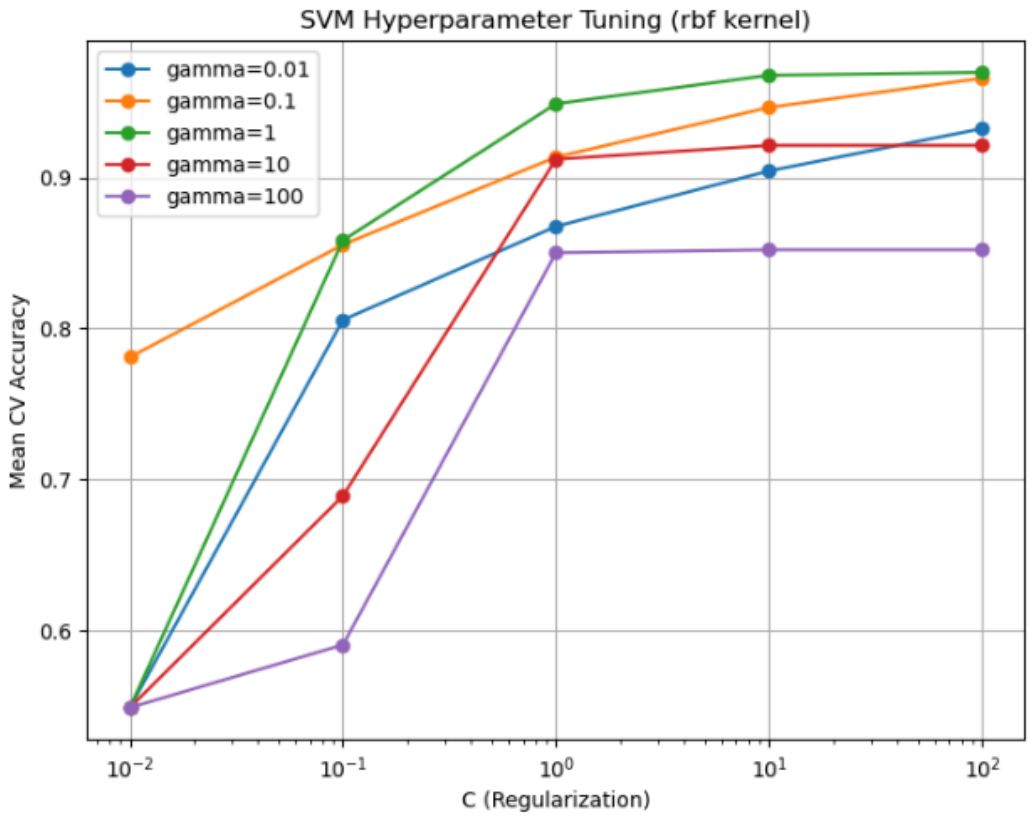


Figure (10): Classification reports for the Support Vector Machine model on the training set and testing set. The model maintained high recall across all risk categories, especially for High-Risk cities, making it the strongest performer in terms of recall so far.

Classification Report for Support Vector Machine: Training Set:

	precision	recall	f1-score	support
Low	1.00	1.00	1.00	736
Moderate	1.00	0.99	1.00	197
High	1.00	1.00	1.00	1135
accuracy			1.00	2068
macro avg	1.00	1.00	1.00	2068
weighted avg	1.00	1.00	1.00	2068

Classification Report for Support Vector Machine: Testing Set:

	precision	recall	f1-score	support
Low	0.98	0.95	0.96	491
Moderate	0.98	0.96	0.97	132
High	0.96	0.98	0.97	757
accuracy			0.97	1380
macro avg	0.97	0.96	0.97	1380
weighted avg	0.97	0.97	0.97	1380

Random Forest Classification

We chose Random Forest because it's a powerful, flexible model that handles complex, non-linear relationships well and performs strongly on tabular data. It's also less sensitive to outliers and missing values, works well with categorical labels, and does not require feature scaling. These qualities made it a strong candidate for our dataset, which includes a mix of numeric climate features and potentially overlapping class boundaries.

To optimize the model, we used **GridSearchCV** with 5-fold cross-validation, tuning three key hyperparameters:

- **n_estimators** = [100, 200] (number of trees)
- **max_depth** = [None, 10, 20] (tree depth)
- **min_samples_split** = [2, 5, 10] (minimum samples to split a node)

The best combination was:

- **n_estimators** = 100
- **max_depth** = 20
- **min_samples_split** = 2

We evaluated the model using recall as our primary metric, since our goal was to correctly identify at-risk cities, especially in the High and Moderate categories.

Figure (11): Random Forest hyperparameter tuning results using **GridSearchCV**. The plot shows how different combinations of **max_depth**, **min_samples_split**, and **n_estimators** affect the mean cross-validated F1 macro score.

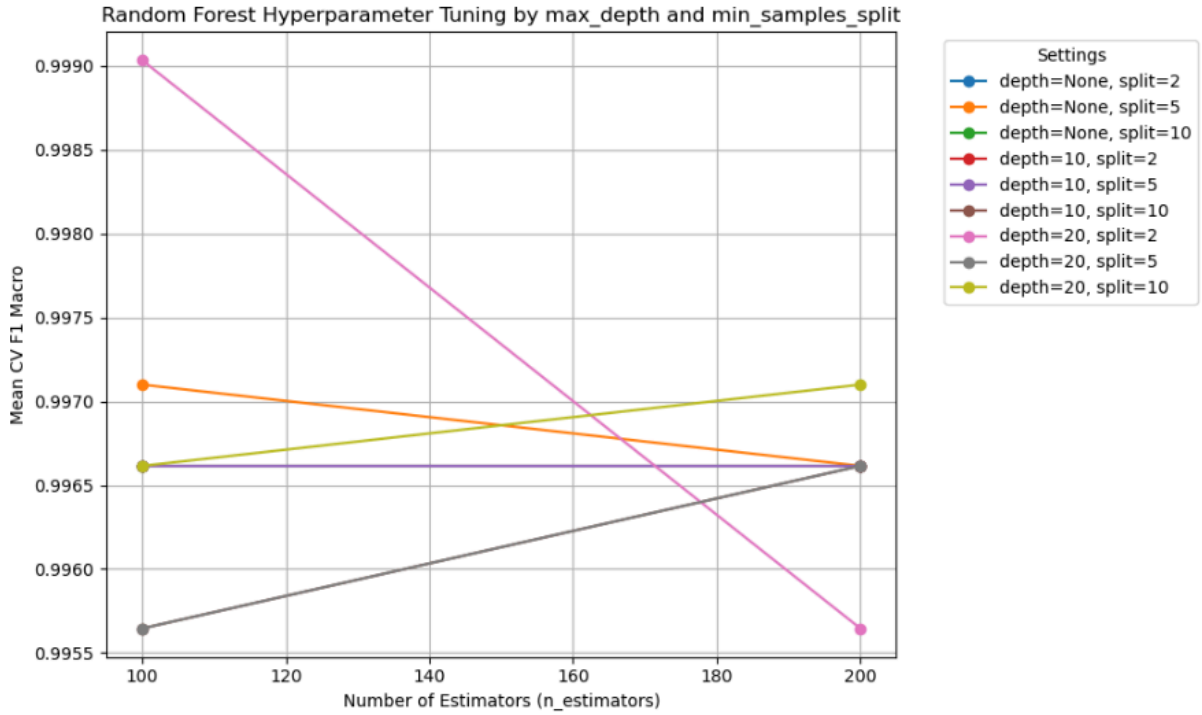


Figure (12): Classification reports for Random Forest on the training set and testing set. The model maintained perfect or near-perfect recall across all risk categories, making it the most accurate and reliable option in our testing.

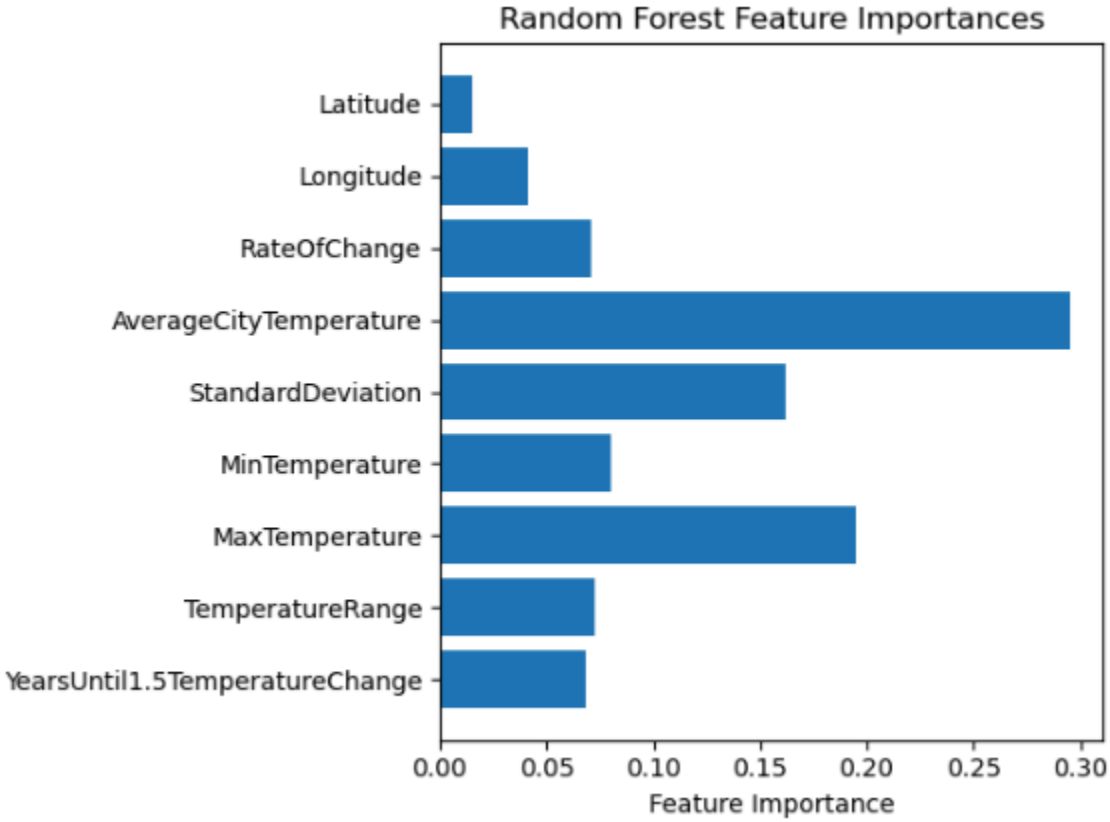
Classification Report for Random Forest: Training Set:

	precision	recall	f1-score	support
Low	1.00	1.00	1.00	736
Moderate	1.00	1.00	1.00	197
High	1.00	1.00	1.00	1135
accuracy			1.00	2068
macro avg	1.00	1.00	1.00	2068
weighted avg	1.00	1.00	1.00	2068

Classification Report for Random Forest: Testing Set:

	precision	recall	f1-score	support
Low	1.00	1.00	1.00	491
Moderate	1.00	0.98	0.99	132
High	1.00	1.00	1.00	757
accuracy			1.00	1380
macro avg	1.00	0.99	1.00	1380
weighted avg	1.00	1.00	1.00	1380

Figure (13): Random Forest feature importances. Features are ranked based on how much they contributed to the model's decision-making process.



Result Analysis

All three models, Logistic Regression, Support Vector Machine (SVM), and Random Forest, gave good results, but some performed better than others.

- Logistic Regression did well for High and Low Risk cities, but had trouble with the Moderate Risk group.
- SVM had very high recall on both the training and testing sets, showing that it was able to generalize well without overfitting.
- Random Forest gave the best overall performance, with perfect recall during training and almost perfect recall on the test set. It consistently found High-Risk cities and was very reliable.

From the feature importance chart, we saw that the most helpful features were Average Temperature, Max Temperature, and Standard Deviation. This means the model focused mostly on actual temperature trends, not just location.

Future Improvements

- Improve how we assign labels by using real-world climate risk data.
- Add more features like population, recent heatwaves, or extreme weather data.
- Try more advanced techniques like Deep Learning.
- Update the dataset with recent years to keep predictions up to date.

Contributions

Epifanio worked on all three machine learning models (Logistic Regression, SVM, and Random Forest), helped clean and process the dataset, and contributed to both the written report and final presentation.

Google Colab