Weather Conditions and Climate Change with ClimateWins

Shravani I.S

November 2024

Introduction

- ClimateWins is interested in using machine learning to help predict the consequences of climate change around Europe and, potentially, the world. The data used in this presentation was collected through hurricane predictions from The National Oceanic and Atmospheric Administration (NOAA) in the U.S., typhoon data from The Japan Meteorological Agency (JMA) in Japan, world temperatures, and a great deal of other data.

OBJECTIVES

ClimateWins is interested in using machine learning to help predict the consequences of climate change around Europe and, potentially, the world.

HYPOTHESES

- Can machine learning predict significant temperature increases in Europe over the next decade?
- Which machine learning models are most effective for forecasting extreme weather events like heatwaves and storms?
- What patterns or correlations can machine learning identify between heatwaves and variables like air quality, urbanization, or economic impact?

Data Sets

The data set based on weather was collected from 18 different weather stations across Europe.

The data set contain information that rang from the late 1800s to 2022,

The records were made almost every day, including values as wind speed, temperate, global ration, between others.

This data was collected by the European Climate Assessment & Data Set Project.

Data Set Link.

Data Bias

1. Collection Bias

- Data is from **18 weather stations** across Europe, while over **26,321 stations** exist. This limited coverage may not represent Europe's diverse climates.
- Changes in **instrumentation** or **station locations** over time can introduce inconsistencies.

2. Location Bias

- Data focuses on Europe and the Mediterranean. Predictions may not generalize to regions like Brazil or Canada due to different climatic systems.

3. Temporal Bias

- Data spans from the **late 1800s to 2022**. Older records may no longer represent current conditions, potentially misleading machine learning models.

4. Sampling Bias

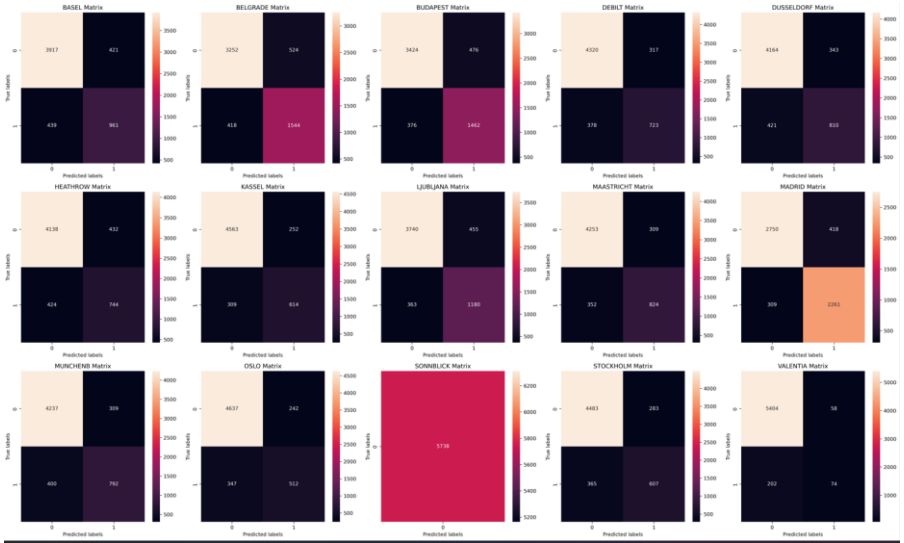
- Specific selection of stations can skew results, missing broader climate patterns across Europe.

Data Optimization

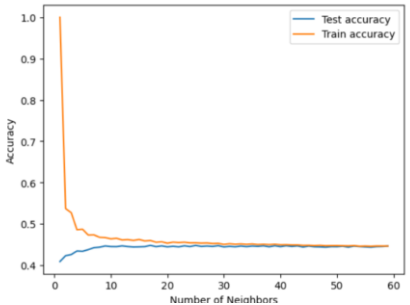
- This data set was optimized through the Gradient Descent.
- Through the application of gradient descent is one of the simplest ways to find a local minimum (or valley) and can be used in linear and nonlinear cases.
- In this case, we applied the gradient descent to find the minimum error, through number of iterations, as well as the number of steps (alpha), which varied as the case.
- It was possible to get a result near to 0, as can be seen in the next slide, after adjusting the theta0 and theta1, as well as the number of iterations and alpha.

K-Nearest Neighbour

Weather Station	Accurate Predictions	False Positive	False Negative	Accuracy Rate
Basel	3917	961	421	439	85%
Belgrade	3252	1544	524	418	84%
Budapest	3424	1462	476	376	85%
Debilt	4320	723	317	378	88%
Dusseldorf	4164	810	343	421	87%
Heathrow	4138	744	432	424	85%
Kassel	4563	614	252	309	90%
Ljubljana	3740	1180	455	363	86%
Maastricht	4253	824	309	352	88%
Madrid	2750	2261	418	309	87%
Munchenb	4237	792	309	400	88%
Oslo	4637	512	242	347	90%
Sonnblick	5738	0	0	0	100%
Stockholm	4483	607	283	365	89%
Valentia	5404	74	58	202	96%
				Average	88%

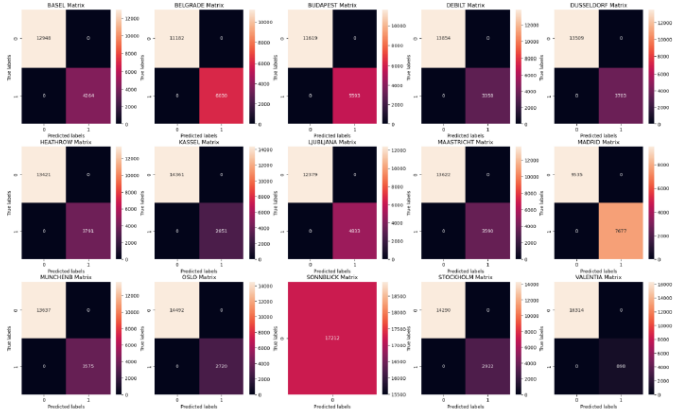


- Promising Weather Stations: Basel, Belgrade, Budapest, Ljubljana, and Madrid show promising weather predictions, as accurate predictions are higher than false positives.
- Sonnblick: Achieved 100% accuracy for unpleasant weather, inflating the overall average to 88%. This may suggest overfitting due to imbalanced data or noise in the model.
- Valentia: Delivered the most reliable accuracy rate of 96%, exceeding the 88% average.
- Belgrade & Budapest: Recorded the lowest accuracy (84% and 85%), falling below the average accuracy of 88%.

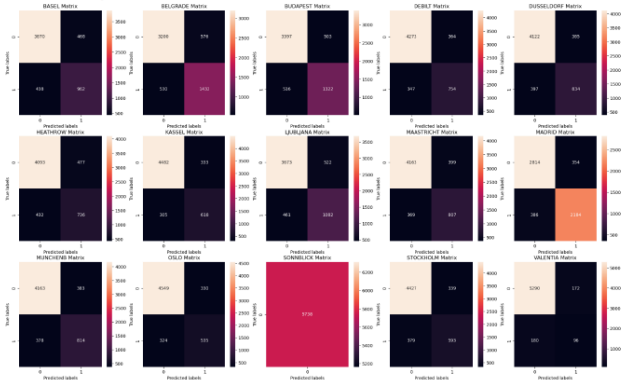


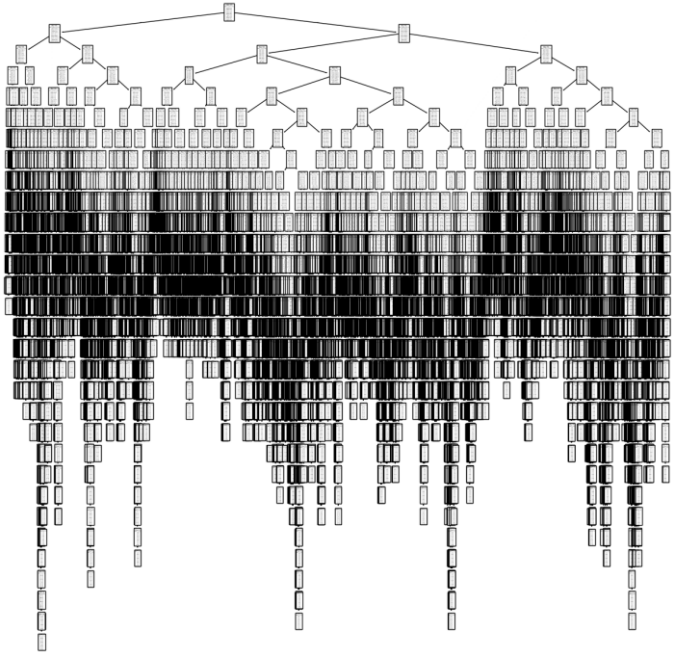
Decision Tree

Training Accuracy:



Test Accuracy:





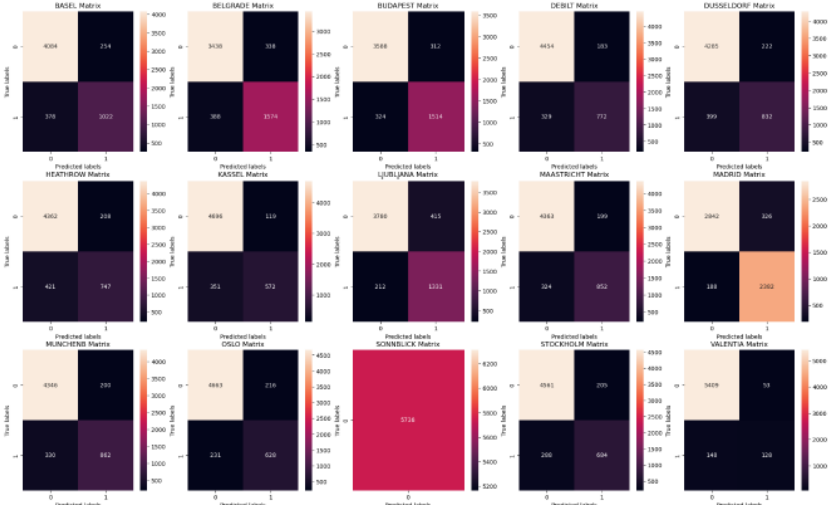
```
]: #What is the testing accuracy score? Using the cross-validation method
   y_pred = weather_dt.predict(X_test)
   print('Test accuracy score: ', accuracy_score(y_test, y_pred))
   multilabel_confusion_matrix(y_test, y_pred)
```

```
Test accuracy score:  0.47385848727779717
```

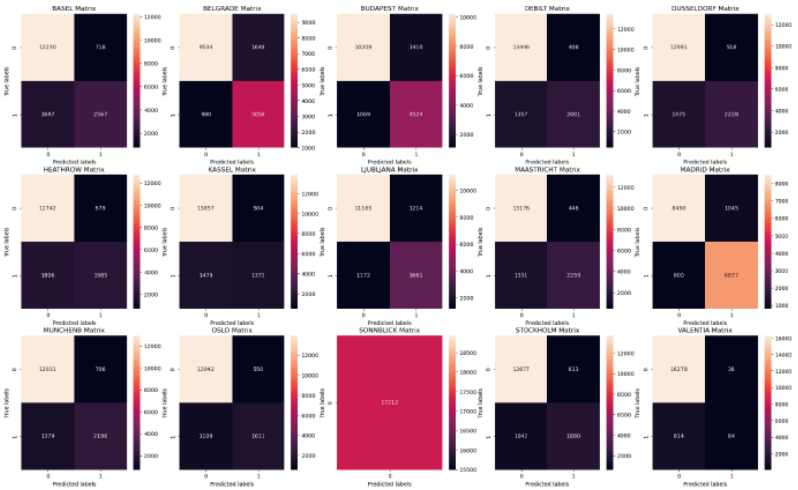
Artificial Neural Network (first scenario)

Weather ANN Model 1 - Confusion Matrix

Test Accuracy:



Training Accuracy:




```
#testing ANN accuracy
y_pred = mlp.predict(X_train)
print(accuracy_score(y_pred, y_train))
y_pred_test = mlp.predict(X_test)
print(accuracy_score(y_pred_test, y_test))
```

46% testing accuracy

```
0.4571810364861724
0.46357615894039733
```

```
#Create the ANN
#hidden_layer_sizes has up to three layers, each with a number of nodes. So (5, 5) is two hidden layers with 5 nodes each,
#and (100, 50, 25) is three hidden layers with 100, 50, and 25 nodes.
mlp = MLPClassifier(hidden_layer_sizes=(5, 5), max_iter=500, tol=0.0001)
#Fit the data to the model
mlp.fit(X_train, y_train)
```

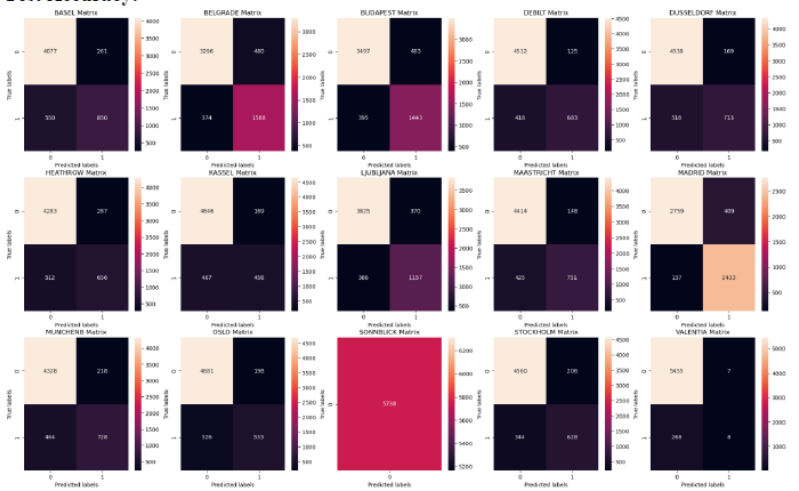
MLPClassifier

```
MLPClassifier(hidden_layer_sizes=(5, 5), max_iter=500)
```

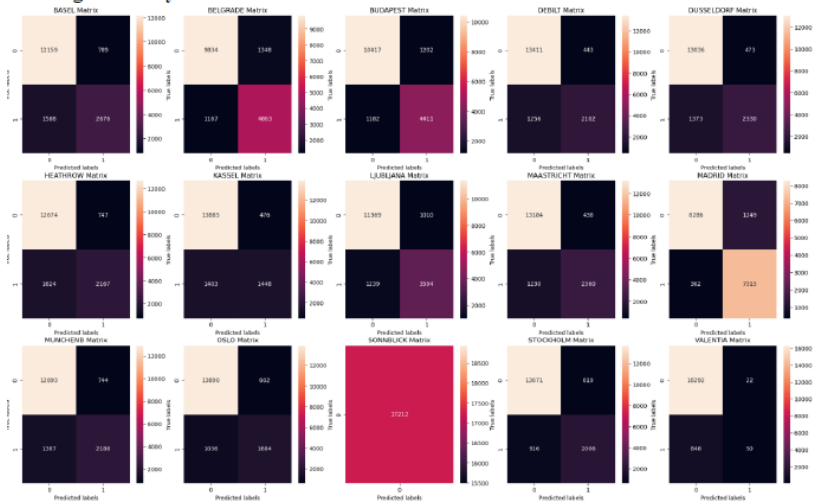
Artificial Neural Network (second scenario)

Weather ANN Model 2 - Confusion Matrix

Test Accuracy:



Training Accuracy:




```
y_pred = mlp.predict(X_train)
print(accuracy_score(y_pred, y_train))
y_pred_test = mlp.predict(X_test)
print(accuracy_score(y_pred_test, y_test))
```

- 47% test accuracy.

```
0.4699047176388566
0.47518299058905
```

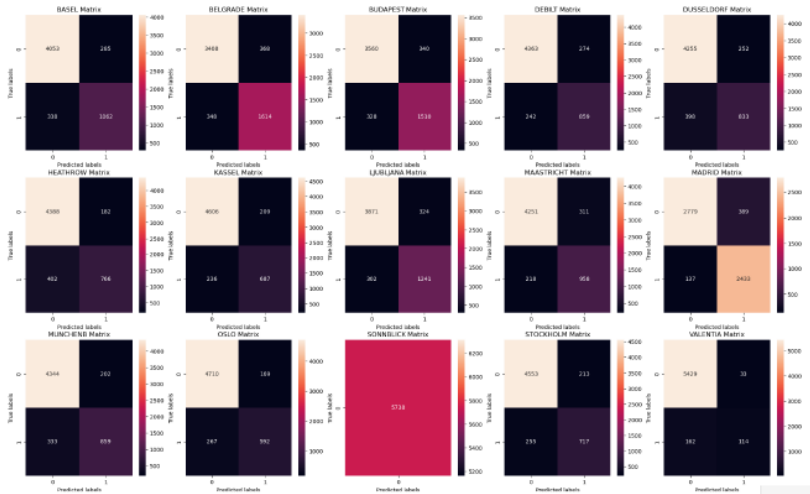
```
[20]: #Create the ANN
#hidden_layer_sizes has up to three layers, each with a number of nodes. So (5, 5) is two hidden layers with 5 nodes each,
#and (100, 50, 25) is three hidden layers with 100, 50, and 25 nodes.
mlp = MLPClassifier(hidden_layer_sizes=(10, 5), max_iter=500, tol=0.0001)
#Fit the data to the model
mlp.fit(X_train, y_train)
```

```
[20]: MLPClassifier
MLPClassifier(hidden_layer_sizes=(10, 5), max_iter=500)
```

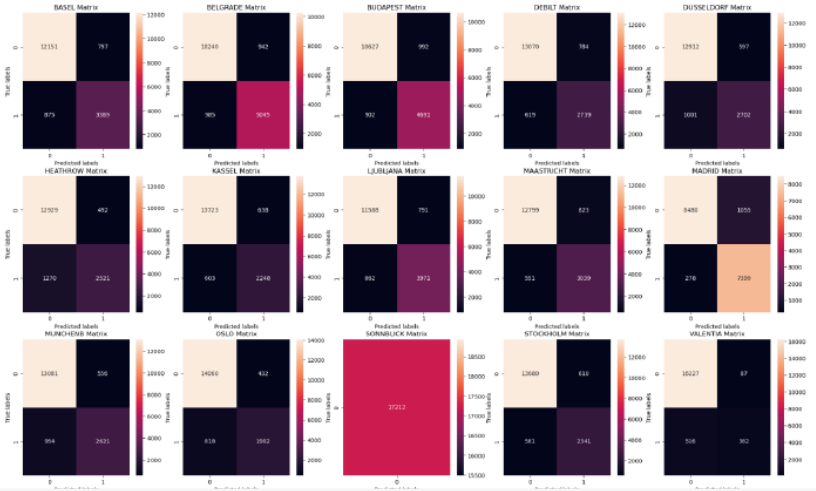
Artificial Neural Network (third scenario)

Weather ANN Model 3 - Confusion Matrix

Test Accuracy:



Training Accuracy:




```
y_pred = mlp.predict(X_train)
print(accuracy_score(y_pred, y_train))
y_pred_test = mlp.predict(X_test)
print(accuracy_score(y_pred_test, y_test))
```

```
0.5153962351847549
0.4949459742070408
```

49% testing accuracy

```
#Create the ANN
#hidden_layer_sizes has up to three layers, each with a number of nodes. So (5, 5) is two hidden layers with 5 nodes each,
#and (100, 50, 25) is three hidden layers with 100, 50, and 25 nodes.
mlp = MLPClassifier(hidden_layer_sizes=(70, 60, 60), max_iter=1000, tol=0.0001)
#Fit the data to the model
mlp.fit(X_train, y_train)
```

```
MLPClassifier
MLPClassifier(hidden_layer_sizes=(70, 60, 60), max_iter=1000)
```

KNN/ANN/OR DECISION TREE?

How accurately do the algorithms predict pleasant and non-pleasant days per weather station?

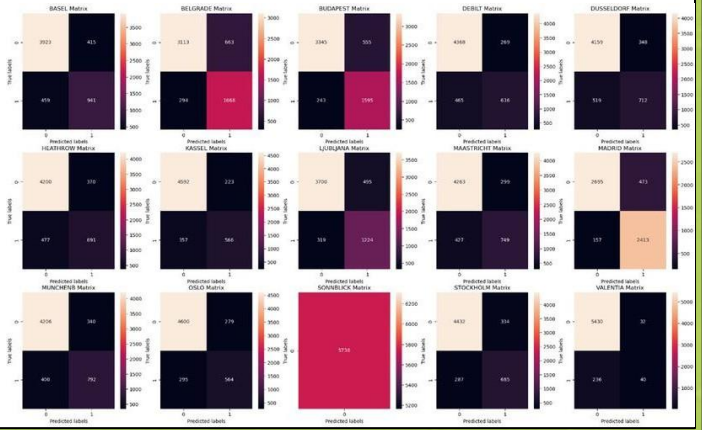
VALENTIA PREDICTION METRICS for 60 neighbors

1. **Accuracy:** 95.34%
2. **Precision:** 99.96%
3. **Recall (Sensitivity):** 95.37%
4. **F1 Score:** 97.61%

Confusion Matrix Scores (pleasant vs non-pleasant weather)

```
print("Accuracy scores for each group:")
for i, accuracy in enumerate(accuracy_scores):
    print(f"Group {i + 1}: {accuracy:.4f}")
```

```
Accuracy scores for each group:
Group 1: 0.8520
Group 2: 0.8294
Group 3: 0.8513
Group 4: 0.8717
Group 5: 0.8536
Group 6: 0.8487
Group 7: 0.9001
Group 8: 0.8526
Group 9: 0.8761
Group 10: 0.8890
Group 11: 0.8736
Group 12: 0.8996
Group 13: 1.0000
Group 14: 0.8961
Group 15: 0.9540
```



- VALENTIA seems to have the least false positives and negatives, & the highest number of true positives out of every station and algorithm used, this indicates that it may be the most accurate at the individual level.

Data Accuracy

- **Decision Tree**: Accuracy not assessed due to the need for pruning.
- **ANN**:
  - **Test Accuracy**: 46% - 49%
  - **Training Accuracy**: 45% - 51%
  - Moderate performance, with a slight risk of overfitting.
- **KNN**:
  - **Test Accuracy**: 88%
  - Best performing model, indicating it is well-suited for this dataset.

Conclusion:

- **KNN** is the top performer with 88% accuracy.
- **ANN** shows stable but lower accuracy (46% - 49%).
- **Decision Tree** requires further adjustments before accuracy can be determined.

1) How is machine learning used, and is it applicable to weather data?

Machine learning algorithms work inductively, identifying patterns and rules from data to solve problems. In this project:

- **K-Nearest Neighbor (KNN):** Uses proximity between data points to classify and predict groupings.
- **Decision Tree Algorithm:** A hierarchical structure guiding decisions based on input data.
- **Artificial Neural Network (ANN):** A network of interconnected nodes, with thresholds to activate specific layers for deeper pattern recognition

2) What ethical concerns exist for this project?

Ethical issues include:

- **Privacy Risks:** Potential monitoring of private properties or activities under the guise of climate change data collection.
- **Political Bias:** Climate data can be influenced by personal beliefs, affecting collection and algorithmic training.
- **Positives in This Case:** Since no personal information is linked to the dataset, privacy violations are unlikely.

3) What are the historical maximums and minimums in temperature?

Based on the European Climate Assessment dataset:
•**Minimum Temperature:** -34.3ºC at Sonnblick on January 13, 1968.
•**Maximum Temperature:** 43.6ºC at Belgrade on July 24, 2007.

4) Can machine learning predict whether weather conditions will be favorable on a certain day?

•The **K-Nearest Neighbor (KNN)** As could be seen in this slide, the machine learning algorithm KNN was able to predict in 88% that the weather was pleasant in that day.
•**With further training** and more comprehensive data, the model's accuracy can increase, allowing it to more reliably predict favorable or hazardous weather conditions based on historical patterns.

Conclusion

The current data was better predicted by the KNN algorithm, considering that it had an average of 88% in the test set.

The decision tree needs to be prune for better accuracy.

As for the ANN, besides the fact that is unpredictable, it has shown an accuracy of around 46% for the test set, which is a lot lower than the KNN algorithm that.

So, for now, since this project it is still incomplete, I'd consider the KNN algorithm the best option, as this one had 88% of accuracy predicting the climate temperature.

VALENTIA STANDS OUT: **95%**

Achieves high accuracy scores consistently around

Next Steps

Continue testing through supervised and unsupervised algorithm, after optimizing them.

Further prune the decision tree for better accuracy.

Search for other options, such as new algorithms, or a combination of algorithms that were already used, to potentially lead us to patterns that were not defined before.

Combine both supervised and unsupervised methods to create a complete climate model that predicts both specific weather events an wider climate trends.

Conduct feature importance analysis to leverage Valentia's strengths.

THANKS!

Do you have any questions?

Please contact me below at:

Or visit :