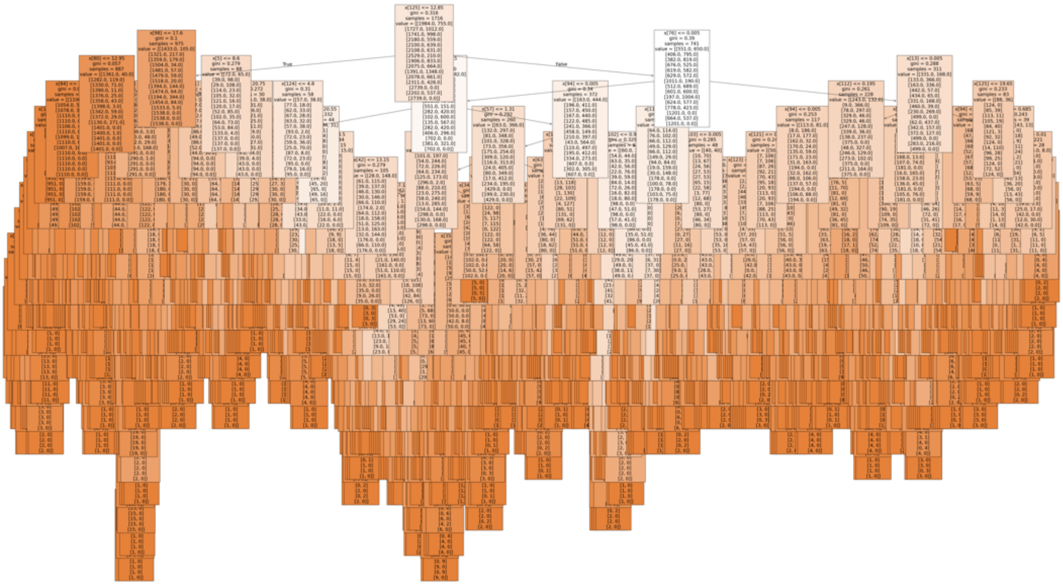
Giada Griso
September 14, 2024
Machine Learning Course
Exercise 2.4

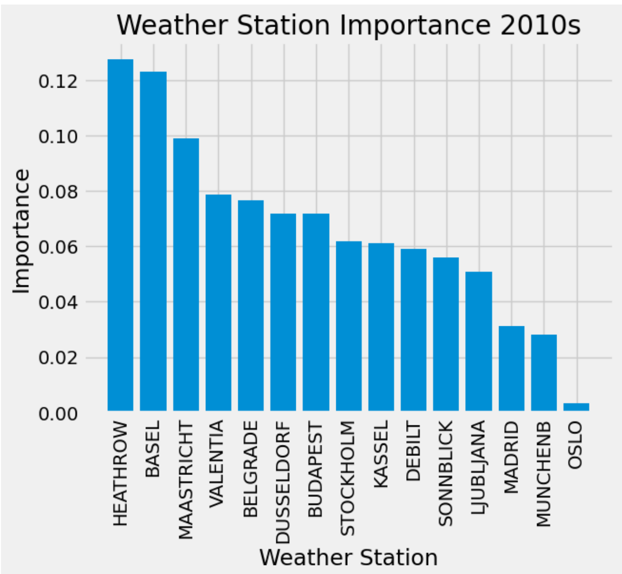
Evaluating Hyperparameters for Model Optimization

Part 1 – Random Forest Model

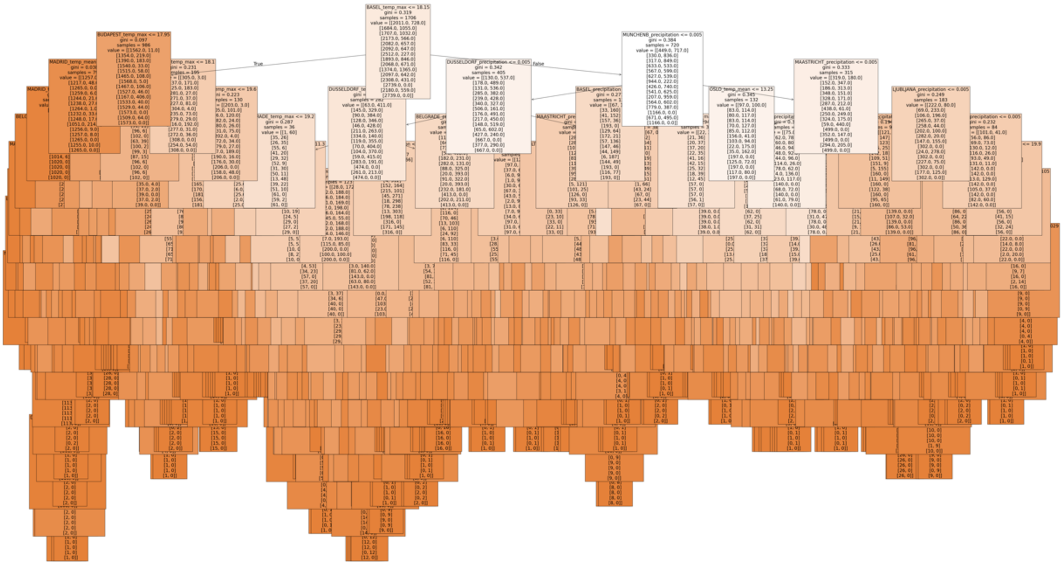
Data Subset	Accuracy Before Optimization (Task 2.3)	Accuracy After Optimization
All Weather Stations (2010 – 2019)	58,7%	66,3%
Heathrow All Years	100%	100%

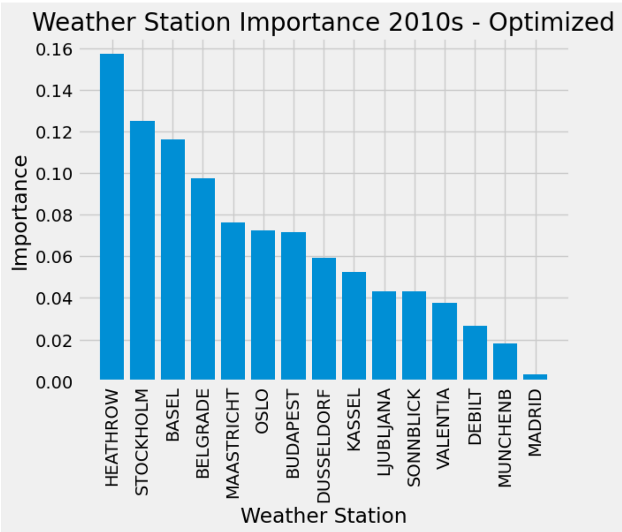
All Weather Stations Before Optimization





All Weather Stations After Optimization



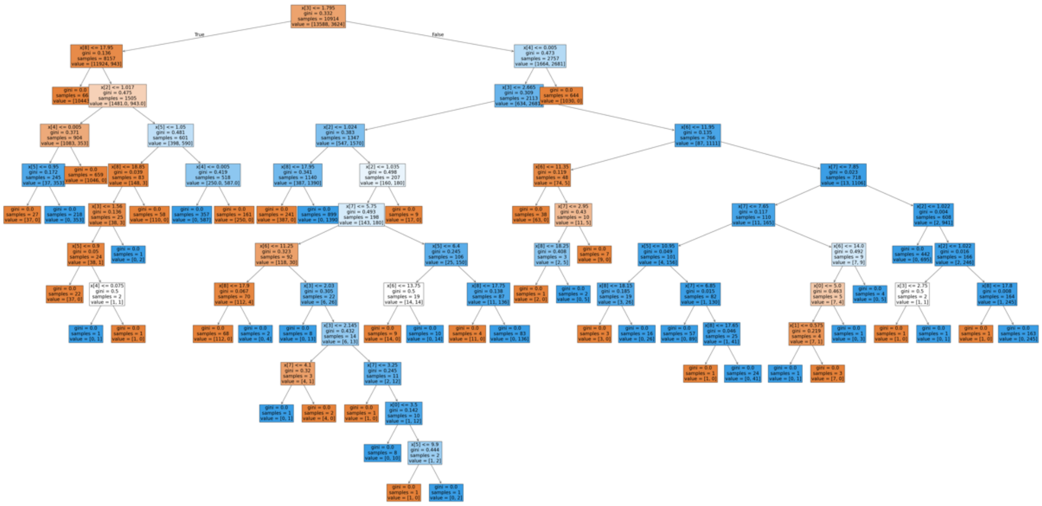


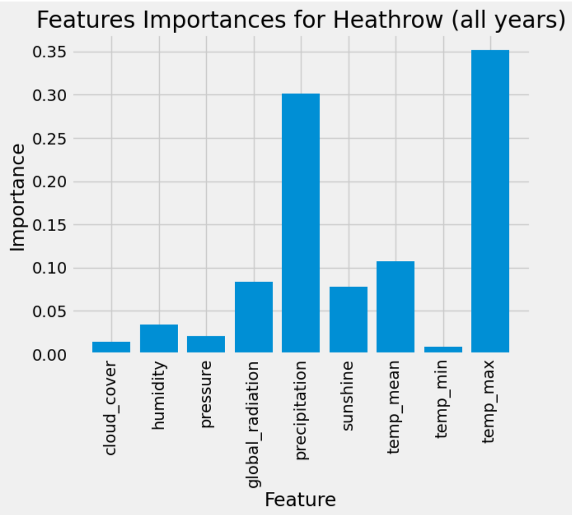
The model accuracy improved of +7,6% after the optimization. The decision trees from both models are complex to interpret, but while the unoptimized tree is deeper, with more layers and splits, the optimized version is simpler, less deep, more balanced, and has fewer splits, helping to prevent overfitting and **improve generalization** by focusing on the most impactful features and splits.

Some notable differences in the bar charts are:

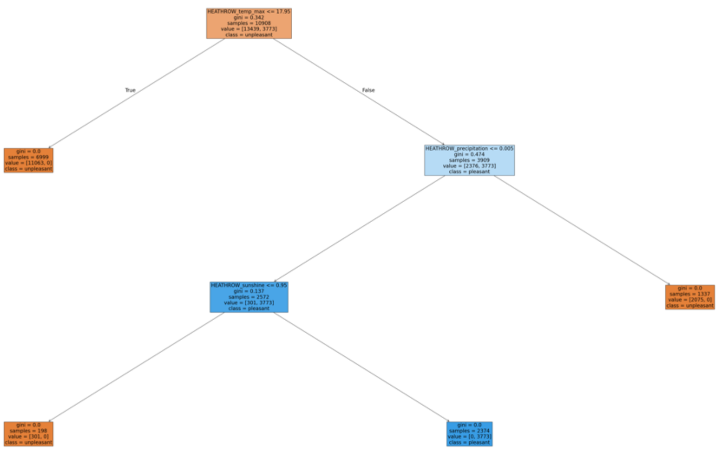
- Heathrow remains the most important station in both charts, but its importance increases slightly after optimization.
- Basel drops from second place to third after optimization, while Stockholm, which is not highly ranked in the first chart, moves to second place in the optimized chart.
- Maastricht falls from third to fifth after optimization, showing that it becomes less important.
- The optimized chart reflects a rebalancing of weather station importance, giving more weight to dominant stations like **Heathrow** and **Stockholm**, while further reducing the relevance of lower-ranked stations.

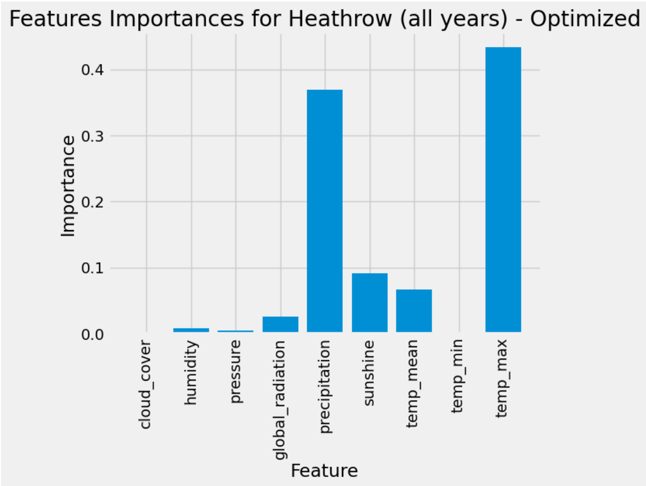
Heathrow Before Optimization





Heathrow After Optimization





After optimizing the hyperparameters for Heathrow, the accuracy maintains stable at 100%. Notable differences are observed in the decision trees: the optimized version of the model has a more straightforward and concise tree structure, likely leading to better generalization, easier interpretation, and improved performance, while the unoptimized version is more complex and prone to overfitting.

In addition, while the weight of the most significant features (**temp_max** and **precipitation**) remain largely unchanged, the least important ones now carry even less weight than before, indicating the optimized model is likely more efficient and focused on the most relevant factors for predicting weather at Heathrow. Sunshine and temp_mean retain their moderate feature importance, but **sunshine** appears to gain slightly more weight after optimization.

Part 2 – Deep Learning – CNN Model

Accuracy Before Optimization (Task 2.2): 12,3%, loss grows exponentially

Accuracy After Optimization: 92,2%, loss = 0.2319

Parameter	Before Optimization	After Optimization
Neurons/n_hidden	256	61
epochs	30	47
batch_size	32	460
learning_rate	-	0,763
kernel	2	2
activation	relu	softsign
optimizer	Adam	Adadelta
layers 1	-	1
layers 2	-	2
normalization	-	0,771
dropout	-	0,730
dropout_rate	-	0,191

Pred	BASEL	BELGRADE	BUDAPEST	DEBILT	DUSSELDORF	HEATHROW	KASSEL	\
True								
BASEL	3517	93	11	4	3	3	1	
BELGRADE	105	987	0	0	0	0	0	
BUDAPEST	28	36	148	2	0	0	0	
DEBILT	14	6	27	35	0	0	0	
DUSSELDORF	5	2	6	5	5	5	0	
HEATHROW	10	2	4	3	2	60	0	
KASSEL	0	3	1	0	1	1	2	
LJUBLJANA	11	6	4	0	0	4	0	
MAASTRICHT	8	0	0	0	0	0	0	
MADRID	26	19	13	1	2	25	0	
MUNCHENB	5	1	0	0	0	0	0	
OSLO	1	0	0	1	0	0	0	
STOCKHOLM	1	0	1	0	0	1	0	
VALENTIA	0	0	0	0	0	1	0	

Pred	LJUBLJANA	MAASTRICHT	MADRID	MUNCHENB	OSLO
True					
BASEL	5	1	44	0	0
BELGRADE	0	0	0	0	0
BUDAPEST	0	0	0	0	0
DEBILT	0	0	0	0	0
DUSSELDORF	0	0	1	0	0
HEATHROW	0	0	1	0	0
KASSEL	2	1	0	0	0
LJUBLJANA	34	0	2	0	0
MAASTRICHT	0	1	0	0	0
MADRID	11	0	361	0	0
MUNCHENB	0	0	1	1	0
OSLO	0	0	0	0	3
STOCKHOLM	0	0	0	0	1
VALENTIA	0	0	0	0	0

After optimizing the model, its accuracy sees a dramatic increase of +79,9%, rising from 12,3% to 92,2%. This improvement is largely driven by key hyperparameter adjustments. The optimizer switches from Adam to Adadelta, the learning rate is fine-tuned to 0,763, and the batch size expands from 32 to 460. Additionally, the number of neurons in the hidden layers reduces from 256 to 61, and the activation function transitions from “relu” to “softsign”. New layers and kernels are introduced, with dropout regularization and normalization added to enhance performance. Collectively, these changes lead to a **more accurate** but **less efficient** model, capable to recognize only 12 out of 15 weather stations.

Part 3 – Iteration

For the purpose of optimizing weather predictions for the Air Ambulance company, I recommend breaking down the dataset into smaller, more manageable components by **locations**, **time intervals** or **weather features**. For locations, the data should be segmented by single weather station – as we did for the random forest optimization – or groups of weather stations, focusing on stations with similar weather patterns (e.g., continental vs. coastal stations) and allowing the model to focus on local weather patterns for each area. Breaking the dataset into different time intervals (like seasons, months, or years) could help capture seasonal changes and trends. Additionally, grouping data by specific weather metrics (like temperature, precipitation, and sunshine) could be useful to test how each factor contributes to flight safety predictions.

When it comes to which model to pick, both have their pros and cons. An **optimized random forest** offers the advantages of transparency and interpretability, as it uses decision trees based on feature importance, making it easier to explain predictions. It requires fewer hyperparameters to tune compared to deep learning models and, in general, it prevents overfitting by averaging the results from multiple trees, leading to more generalizable predictions. However, it is limited in capturing complex, non-linear relationships in data, and it struggles to generalize well to time frames outside of the training data. It can also become computationally expensive with large datasets. On the other hand, an **optimized CNN** excels at detecting intricate patterns and relationships (such as spatial or temporal patterns), making it ideal for handling large and complex datasets. It also scales well and can learn directly from raw data, reducing the need for feature engineering. As a downside, however, it lacks in transparency, making it harder to interpret, and it demands significant computational resources and hyperparameter tuning, increasing complexity and risk of overfitting if not optimized carefully.

For all these reasons, I believe the **optimized random forest** should be used as a starting point for the weather predictions (especially when we want to break down the data by single weather stations, as the model could reach 100% accuracy), and then eventually introduce the optimized CNN for exploring more complex relationships and identifying temporal patterns.

In term of **key weather variables** critical to flight safety, the optimized random forest confirmed the importance of **temperature** (especially its maximum) and **precipitation** as the main factors influencing weather predictions, therefore I suggest Air Ambulance to focus on those primarily.