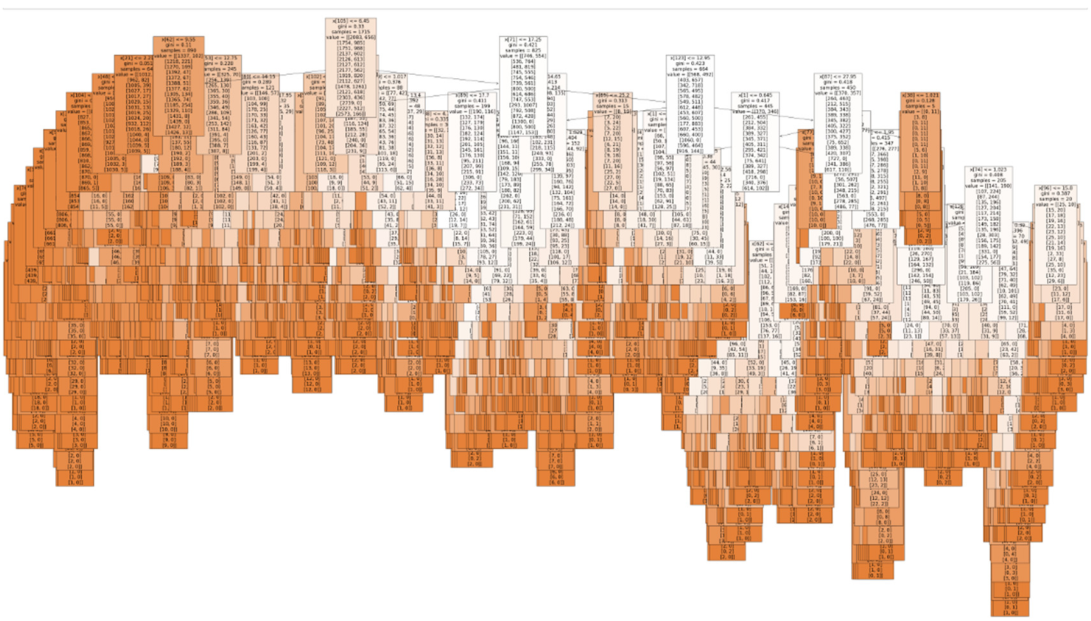
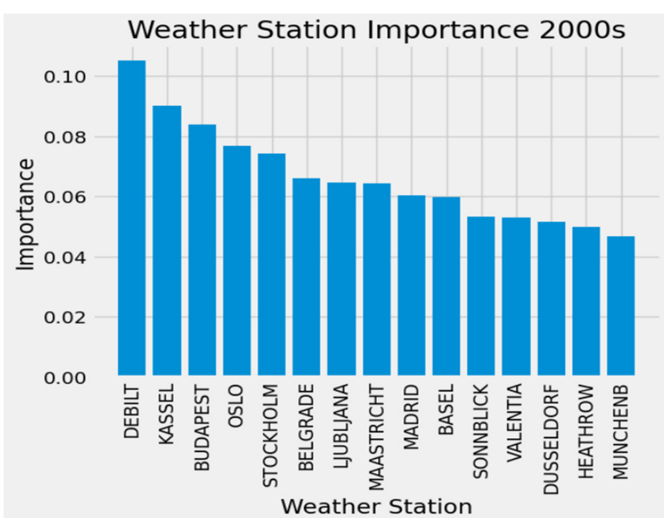
Exercise 2.4: Evaluating Hyperparameters
Timothy Aluko

Part 1: Random Forest

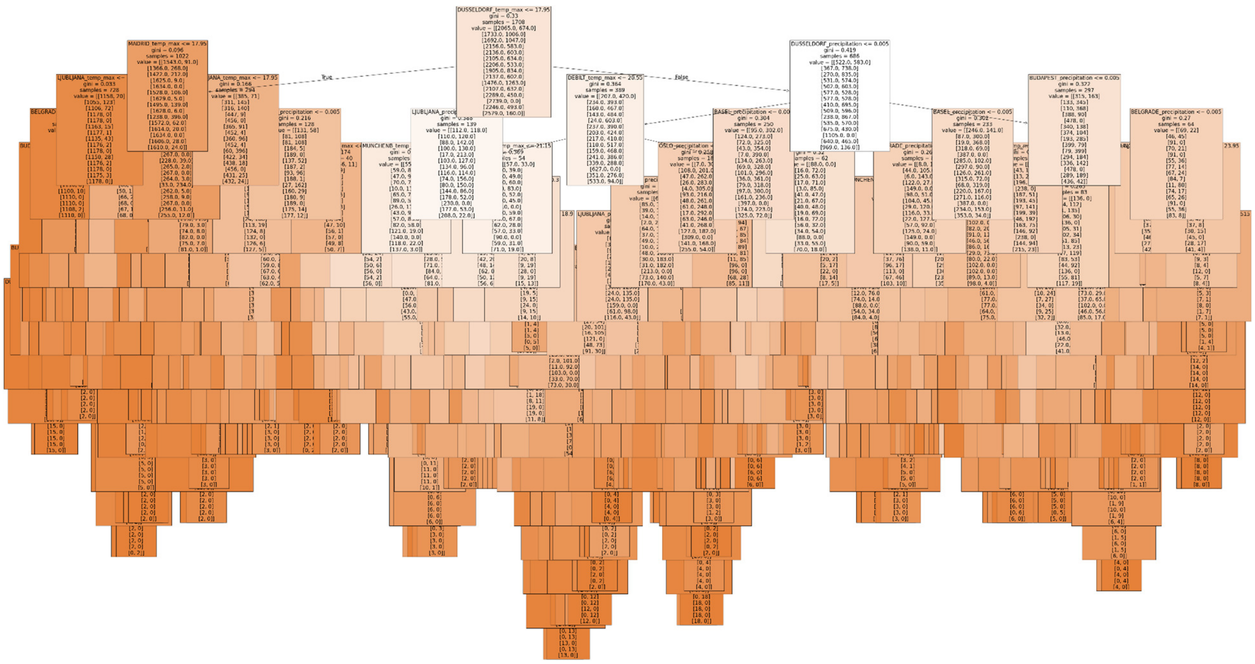
Data subset	Accuracy - Exercise 2.3	Accuracy - Exercise 2.4
All weather stations (2000-2009)		66.958%
Budapest All years	100%	100%

All Weather Stations Before Optimization

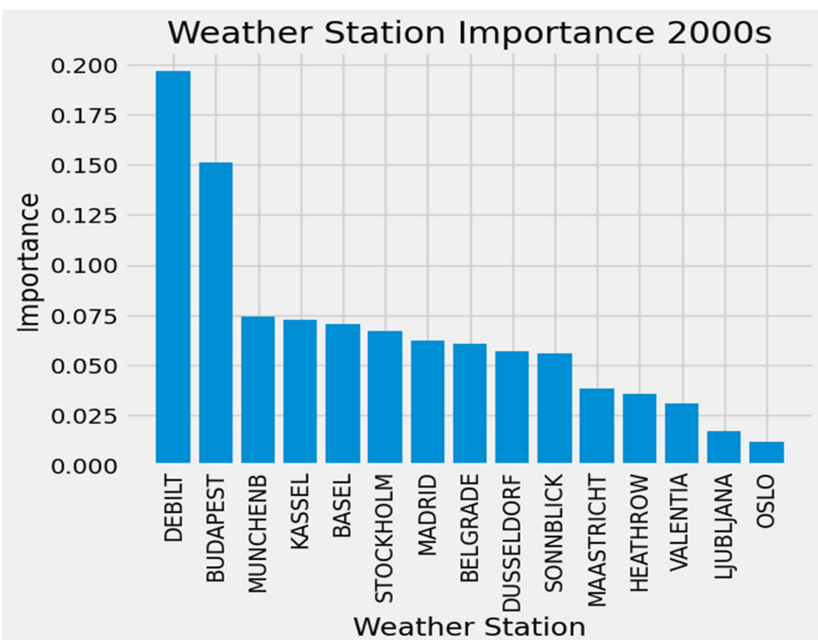




All Weather Stations After Optimization



Decision tree 2000-2009 After Hyperparameter Optimisation



The model accuracy improved by 0.67% after the optimization. The decision trees from both models are complex to interpret, but while the unoptimised tree is deeper, with more layers and splits, the optimised version is simpler, less deep, more balanced, and has fewer splits, helping to prevent overfitting and improve generalisation by focusing on the most impactful features and splits. There are some notable differences in the bar charts.

Debilt remains the most important station in both charts and also increased its importance more after optimization.

Kassel drops from second place to fourth after optimisation, while Budapest, which is in third place in the first chart, moves to second place in the optimised chart.

Munichenb ranked last in the ranking, and rose to the second most important position after optimisation, demonstrating its increasing importance.

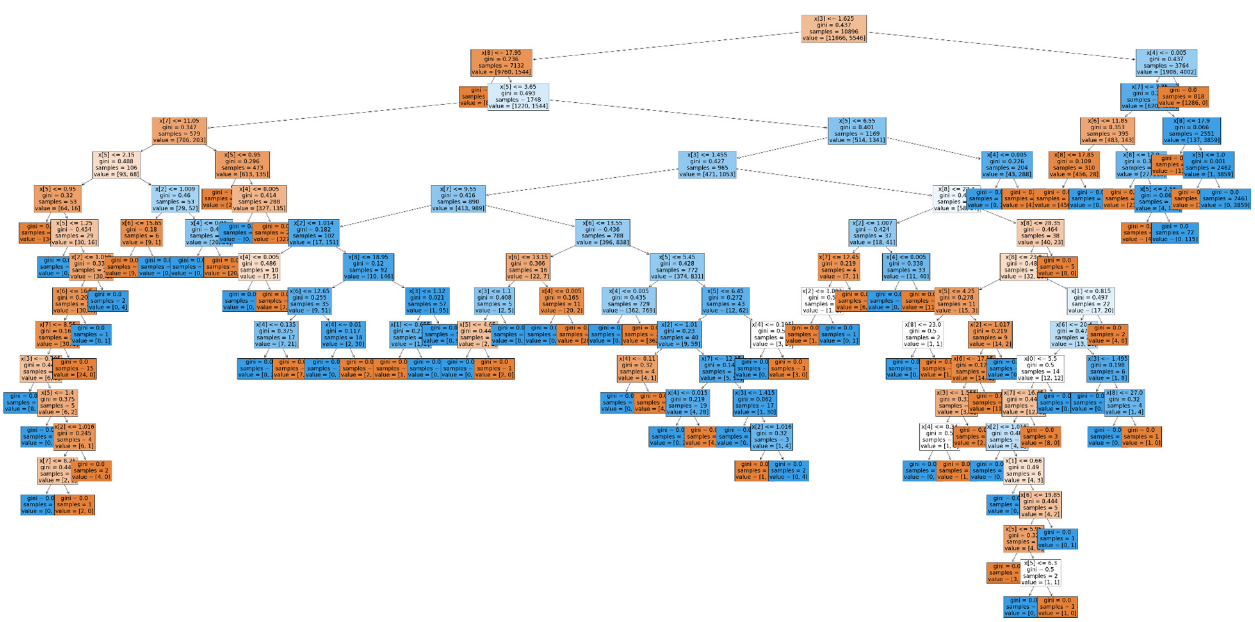
The optimised chart reflects a rebalancing of weather station importance, giving more weight to dominant stations like Debilt Munchenb and Kassel while further reducing the relevance of lower-ranked stations.

Budapest

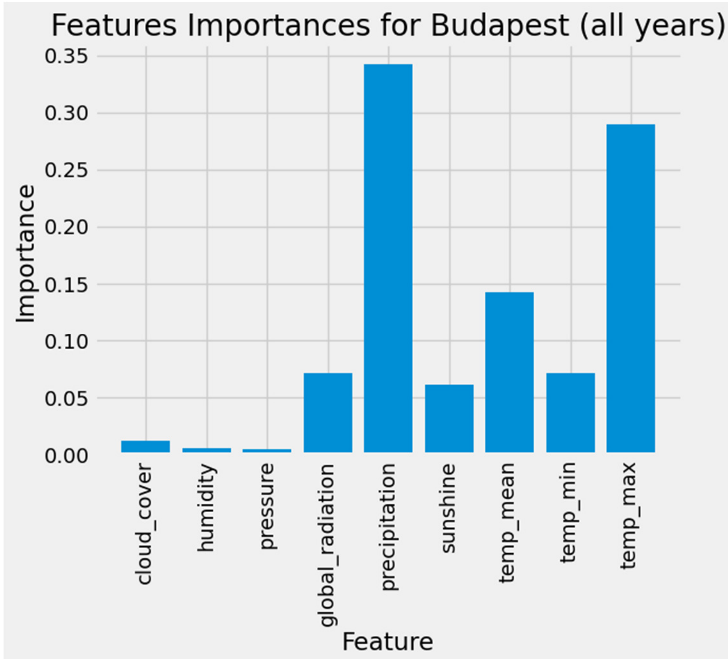
The accuracy of the predictions of the random forest for Budapest in all years remained the same as it was already at 100% in Exercise 2.3. However, the decision tree generated with the optimised hyper-parameters is considerably simpler. The feature importance charts show that only 3 features (precipitation, sunshine and max temperature) are important after optimisation compared to all the features (9 features) in the un-optimised random forest.

Below are images of the decision trees and feature importance graphs.

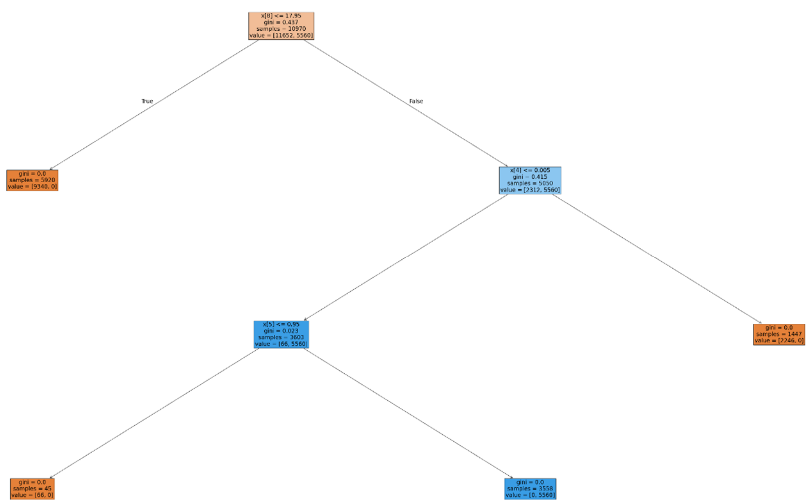
Before optimisation



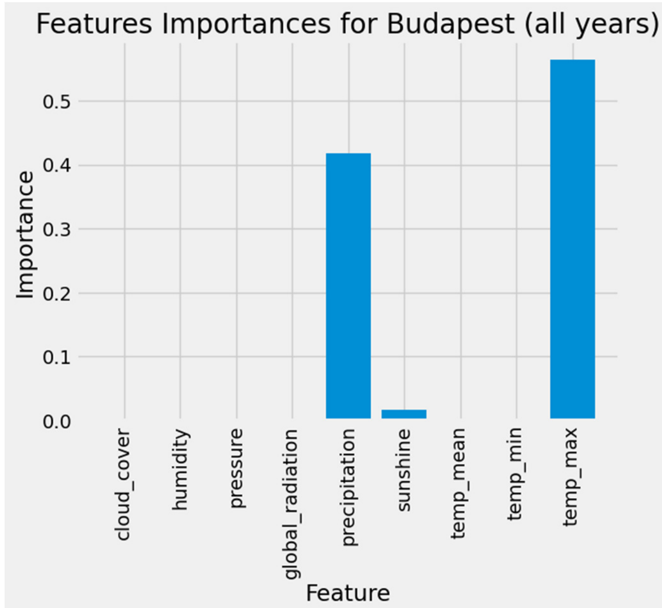
Decision tree Budapest All years



After optimisation



Decision Tree Budapest - All years



Part 2: Deep Learning

Accuracy Before Optimization (Task 2.2): accuracy: 0.1180, loss: 543366464.0000

Accuracy After Optimization: accuracy: 0.9518, loss: 0.1420

Parameter	Before optimisation	After Optimisation
Optimiser	ADAM	Adadelta
Epochs	30	91
Activation	relu	softsign'
Batch size	16	460
Dropout	-	0.7296061783380641
Kernel	2	1.9444298503238986
Dropout rate	-	0.19126724140656393
layers1	-	1
layers2	-	2
Learning rate	-	0.7631771981307285
Neurons	256	61
Normalisation	-	0.770967179954561

Hyper-Parameters used

After optimisation, the models accuracy in predicting weather stations based on the weather data improves by a large margin, going from 11.8% accuracy to 95.18% accuracy. Hence, the optimisation was successfully able to determine the best parameters for this problem. Looking at the confusion matrices shows that the model after optimisation makes a lot of less false predictions than before optimisation; for instance, Basel is predicted to be Dusseldorf, 1233 times before optimisation, but after optimisation, this error occurs 0 times.

Major changes were using the Adadelta optimiser over the ADAM optimiser, increasing the number of epochs from 30 to 91; increasing the batch size and decreasing the number of neurons from 256 to 61. The optimised model is unlikely to be overfitted, based on how accurately the model performs on the test set (see confusion matrix below) and due to the reduction in the number of neurons from 256 in the unoptimised model to 61 in the optimised model. The reduction in number of neurons reduces the possibility of the random noise in the data being learnt by the model.

Pred / True	BASEL	BELGRADE	BUDAPEST	DEBILT	DUSSELDORF	HEATHROW	KASSEL
BASEL	16	85	574	547	1233	33	107
BELGRADE	0	102	74	106	564	0	1
BUDAPEST	0	10	11	41	103	0	0
DEBILT	0	7	0	23	45	0	0
DUSSELDORF	0	0	0	9	16	0	0
HEATHROW	0	1	0	29	34	0	0
KASSEL	0	2	0	2	5	0	0
LJUBLJANA	0	2	2	7	20	0	0
MAASTRICHT	0	0	0	0	6	0	0
MADRID	1	0	17	112	133	1	5
MUNCHENB	0	0	0	0	1	0	0
OSLO	0	0	0	2	2	0	0
STOCKHOLM	0	1	0	1	1	0	0
VALENTIA	0	0	0	1	0	0	0

Pred / True	MAASTRICHT	MADRID	MUNCHENB	OSLO	SONNBLICK	STOCKHOLM	VALENTIA
BASEL	1	954	3	85	6	6	32
BELGRADE	0	236	0	9	0	0	0
BUDAPEST	0	47	0	2	0	0	0
DEBILT	0	6	0	1	0	0	0
DUSSELDORF	0	4	0	0	0	0	0
HEATHROW	0	16	0	2	0	0	0
KASSEL	0	1	0	1	0	0	0
LJUBLJANA	0	30	0	0	0	0	0
MAASTRICHT	0	3	0	0	0	0	0
MADRID	0	171	0	17	0	1	0
MUNCHENB	0	6	0	1	0	0	0
OSLO	0	1	0	0	0	0	0
STOCKHOLM	0	1	0	0	0	0	0
VALENTIA	0	0	0	0	0	0	0

Confusion Matrix - Before Hyperparameter Optimisation

Pred / True	BASEL	BELGRADE	BUDAPEST	DEBILT	DUSSELDORF	HEATHROW	KASSEL
BASEL	3644	22	6	1	0	2	0
BELGRADE	84	999	4	0	0	1	0
BUDAPEST	17	14	173	4	0	1	0
DEBILT	3	3	10	63	2	1	0
DUSSELDORF	4	1	0	3	9	10	0
HEATHROW	7	1	0	1	2	66	0
KASSEL	1	0	0	0	0	0	7
LJUBLJANA	7	3	1	0	1	0	0
MAASTRICHT	4	0	0	0	0	1	0
MADRID	33	3	4	0	0	2	0
MUNCHENB	7	0	0	0	0	0	0
OSLO	1	0	0	0	0	0	0
STOCKHOLM	1	0	0	0	0	0	0
VALENTIA	1	0	0	0	0	0	0

Pred / True	LJUBLJANA	MAASTRICHT	MADRID	MUNCHENB	OSLO	STOCKHOLM
BASEL	1	1	5	0	0	0
BELGRADE	0	0	4	0	0	0
BUDAPEST	0	0	5	0	0	0
DEBILT	0	0	0	0	0	0
DUSSELDORF	0	0	2	0	0	0
HEATHROW	0	0	4	0	1	0
KASSEL	1	1	1	0	0	0
LJUBLJANA	40	0	8	0	1	0
MAASTRICHT	0	4	0	0	0	0
MADRID	1	1	413	0	1	0
MUNCHENB	0	0	0	1	0	0
OSLO	0	0	0	0	4	0
STOCKHOLM	0	0	0	1	0	2
VALENTIA	0	0	0	0	0	0

Confusion Matrix - After Hyperparameter Optimisation

Part 3: Iteration

I would iterate on the data by splitting the data by location and season to determine how the accuracy of the models is affected by these factors. I choose location and season as these are known to be two large determinants of the weather at a specific location. Using models split like these could allow for more accurate predictions of the weather at any location and time of the year.

The random forests models are much simpler to train and interpret than the neural network models, while still being able to represent the nonlinear nature of the data. However, as the complexity of the data increases they become less accurate; this is seen in the results above where the random forest trained on all weather stations has a 66.95% accuracy and the random forest trained on only Budapest data has a 100% accuracy. The CNN approach on the other hand is capable of representing complex data sets as it has an accuracy of 95.18% on all data; but is more expensive to train and does not give any insight into the data as it is a black box. Hence, the random forest should be used initially to gain a sense of the variation in the climate of different stations and what the most important features are; this could help the air ambulance company form policies on how to decide when to fly. The CNN would be useful for investigating more complex weather patterns and when introducing other data sources that might be relevant to the problem.

The feature importance of the optimised random forest says that precipitation and maximum temperature are the most important features for predicting the weather. Hence, ClimateWIns should focus on monitoring these in order to determine if it is safe to fly on a day.