Optimization

GRID	Hyperperameter	Input	Results	Input - Took 4.4 minutes to run	Results
	max_depth	3, 10, none	None	5, 10, 15	10
	n_estimators	10, 100, 100	100	100, 200, 300	300
	max_features	1, 3, 5, 7	7	2, 4, 6, 8	8
	min_samples_leaf	1, 2, 3	1	4, 5, 6	4
	min_samples_split	2, 3	3	5, 10	5
			53.50%		52.60%
				Took 16 minutes	
RANDOM	max_depth	10, 100 step 10	60	10, 75 step 5	45
	n_estimators	10, 500 step 50	360	100, 500 step 10	470
	max_features	1, 7	6	5, 10	9
	criterion	gini, entropy	gini	gini, entropy	entropy
	min_samples_leaf	1, 4,	1	1, 4	2
	min_samples_split	2, 10	4	2, 10 step 1	3
			52.90%		55.60%

n_estimators – every time I expanded the estimators upwards the optimized result came close to the max hyperparameter. The Random Grid found 360 in a range from 10-500 with a step of 50 and 470 in a range of 100-500 with a step of 5. The grid returned the maximum amount of the given range in both attempts as the optimized value. I am going to use the highest number returned **470.**

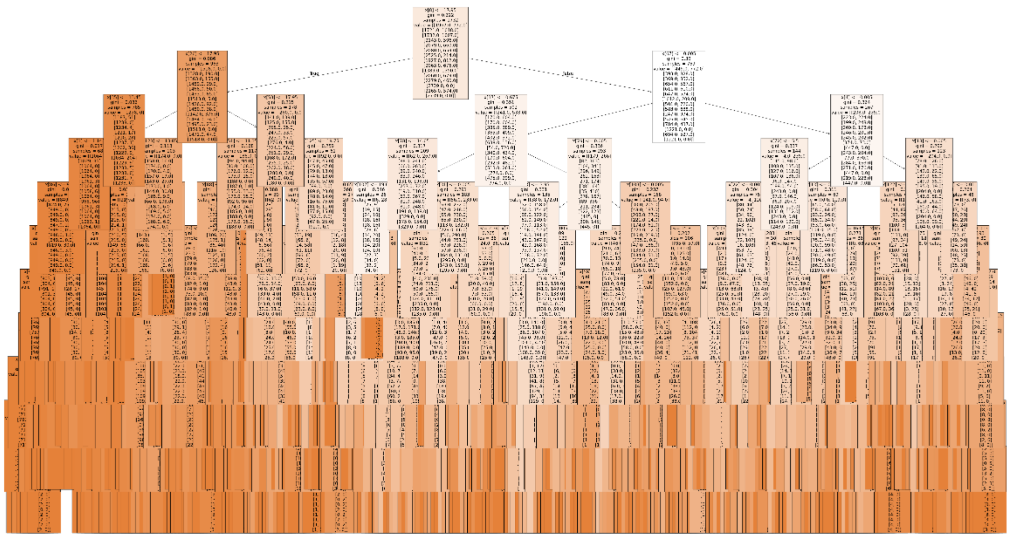
Max_depth – Every time the optimized max depth (except when none was given as option) was found, it was close to the average of the range that I input. I'm concerned with overfitting so I am going to take the lowest max_depth that was found 10.

Min_samples_leaf – always returned the lowest value I gave as an option. Going to set at 1.

Max_features – The optimized result were always at the top of the range that was input. I'm going to take the average of the optimized results 7.

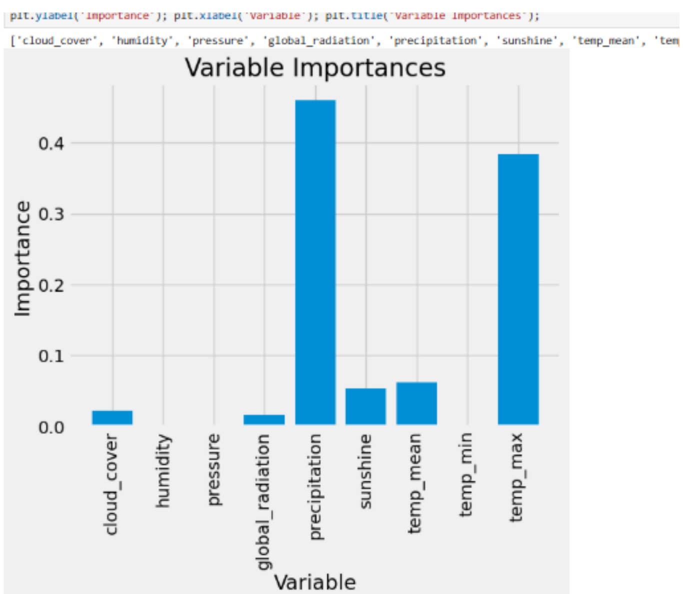
Min_samples_split – When I used a step of 1 the optimized result was 3.

Using these values dropped the accuracy several percent age points to 53.5%. The variables of greatest significance were Basel, Dusseldorf, and Maastricht. These results are similar to the results from the earlier lesson.



Dusseldorf

When I ran the same parameters on Dusseldorf, I once again got 100% accuracy with precipitation as the variable of greatest importance at 46%. Max temperature had the second highest influence at 38%. I believe the model is looking for moderate temperatures without rain to define pleasant weather conditions. This is a very high reliance on these two variables. I'm concerned that the high reliance on these two variables is causing overfitting.



I attempted to find the f-score and precision for this model based on the comment from the last assignment. I think I am doing something wrong as my precision was at 100%.

```
[57]: precision = precision_score(y_test, y_pred)
      recall = recall_score(y_test, y_pred)
      f1 = f1_score(y_test, y_pred)
      accuracy = accuracy_score(y_test, y_pred)
```

```
[59]: print(f"Precision: {precision}")
      print(f"Recall: {recall}")
      print(f"F1-Score: {f1}")
      print(f"Accuracy: {accuracy}")

      Precision: 1.0
      Recall: 1.0
      F1-Score: 1.0
      Accuracy: 1.0
```

I would like to learn more about finding the precision. If I am experience low precision but high accuracy then my model is over fitting and I need to change the hyperparameters to fix this issue.

Deep Learning

The CNN model greatly improved from my first attempt in exercise 2.2. In that exercise, my losses were exponentially huge and I had very poor accuracy. This exercise shows me how important the correct hyperparameters are. At first the CNN model appeared to be incompatible with the ClimateWins data using the Bayseian optimization, I was able to find hyperparameters that make the model work. Using the optimized hyperparameters shown below, the model improv ed to 90.8% accuracy with a loss of .2706.

```
{'activation': 'softsign',
 'batch_size': 460,
 'dropout': 0.7296061783380641,
 'dropout_rate': 0.19126724140656393,
 'epochs': 47,
 'kernel': 1.9444298503238986,
 'layers1': 1,
 'layers2': 2,
 'learning_rate': 0.7631771981307285,
 'neurons': 61,
 'normalization': 0.770967179954561,
 'optimizer': <keras.src.optimizers.adadelta.Adadelta at 0x270d4096060>}
```

#06 CNN with Optimized Paramaters

```
#07.4 Evaluate matrix
print(confusion_matrix(y_test, y_pred, stations))
```

```
Pred        BASEL  BELGRADE  BUDAPEST  DEBILT  DUSSELDORF  HEATHROW  KASSEL  \
True
BASEL        3450       128        27      13          11        13       1
BELGRADE       91       899        49       5           2         5       0
BUDAPEST       29         8       142       8           4         5       2
DEBILT         11         2         2      47           6         8       1
DUSSELDORF      5         0         1       1           7         9       0
HEATHROW        9         1         1       0           4        45       0
KASSEL          0         0         1       0           0         0       6
LJUBLJANA       6         1         1       0           0         0       0
MAASTRICHT      5         0         0       0           0         1       0
MADRID         50         1        10       0           0         2       1
MUNCHENB        6         0         0       0           0         0       0
OSLO            0         0         0       0           0         0       0
STOCKHOLM       1         0         0       0           0         0       0
VALENTIA        1         0         0       0           0         0       0

Pred        LJUBLJANA  MAASTRICHT  MADRID  MUNCHENB  OSLO
True
BASEL               2           0      33         2     2
BELGRADE            5           0      34         0     2
BUDAPEST            5           0      11         0     0
DEBILT              1           1       3         0     0
DUSSELDORF          2           0       4         0     0
HEATHROW            1           0      19         0     2
KASSEL              1           1       2         0     0
LJUBLJANA          36           0      15         1     1
MAASTRICHT          0           1       2         0     0
MADRID              1           0     392         0     1
MUNCHENB            0           0       0         1     1
OSLO                0           0       0         0     5
STOCKHOLM           0           0       0         1     2
VALENTIA            0           0       0         0     0
```

Iterations

If I was breaking down this dataset I would strongly consider geography as a category that should define smaller groups. I would have to look more closely if the groups should be weather stations that are near each other or weather stations that are on the same latitude. I would need to look into weather patterns before decided. I also think it might be good to divide into groups based on average precipitation levels. Since precipitation keeps dominating the variables, creating groups based on precipitation might stop the model from overfitting. Before recommending anything to Air Ambulance I need to study what weather impacts helicopters ability to fly. I assume heavy rain, wind, and cloud build up are important factors. I'm not sure that temperature matters.