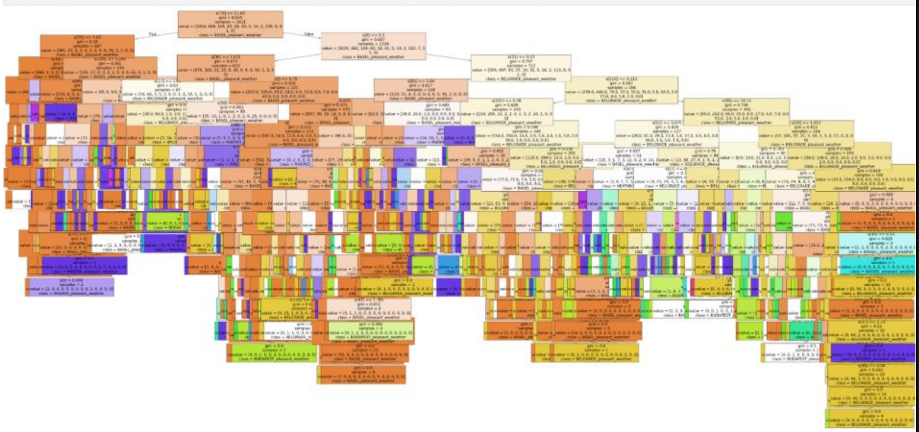
SHRAVANI
11/26/24

2.4: Evaluating Hyperparameters

Hyperparameters Optimization - GridsearchCV/RandomsearchCV - Random Forest

1) All weather stations

```
fig = plt.figure(figsize=(80,40))
plot_tree(clf3.estimators_[15], fontsize = 20, class_names=weather_outcomes, filled=True);
```



Parameters 1:

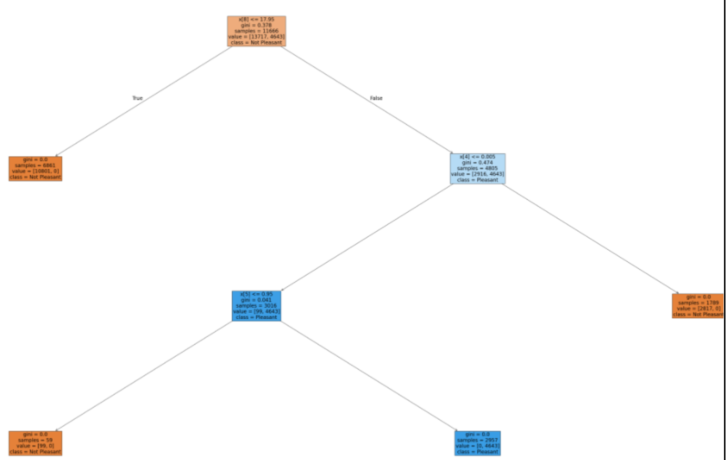
{'max_depth': 3,
'max_features': 7,
'min_samples_leaf': 1,
'min_samples_split': 2,
'n_estimators': 200}

GridsearchCV
SCORE = 0.855/85%

RandomsearchCV
SCORE = 0.849/85%

2) Basel weather station

```
fig = plt.figure(figsize=(80,40))
plot_tree(clf3.estimators_[15], fontsize = 20, class_names=bas_labels, filled=True);
```



Parameters 2:

{'max_depth': 3,
'max_features': 7,
'min_samples_leaf': 1,
'min_samples_split': 2,
'n_estimators': 200}

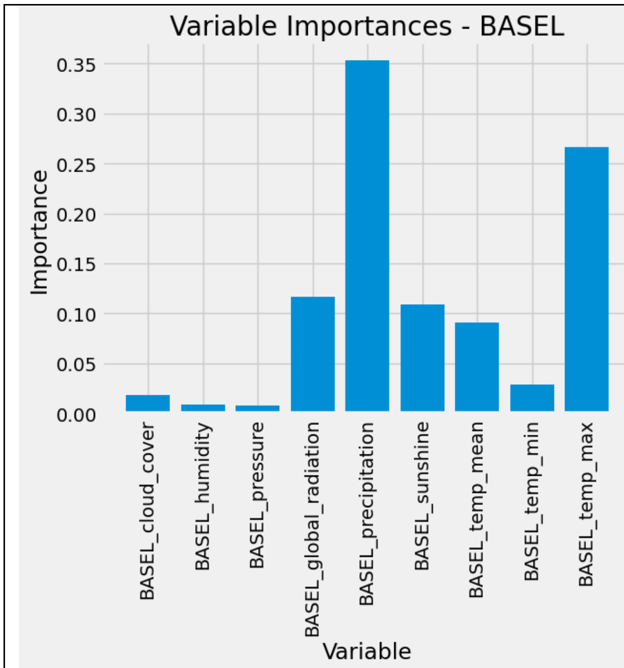
ORIG. SCORE = 1.0

SCORE: 1.0/100%

QNote: very few tree nodes?

Feature Importances Analysis

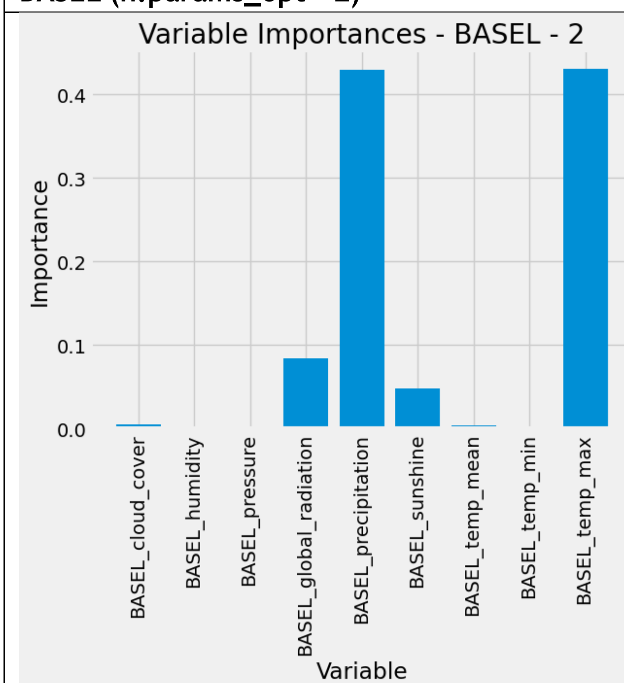
- **BASEL (original plot)**



Top features of importances for BASEL weather station are:

1. Temperature max
2. Temperature mean
3. Precipitation

BASEL (h.params_opt - 2)



Top features of importances for BASEL weather station are still currently:

1. Precipitation
2. Temperature max
3. Temperature mean

Hyperparameters Optimization – Bayesian Optimization Function – Keras CNN model

<table>
<tr><th>CNN model – All weather stations</th><th>Notes:</th></tr>
<tr><td>[28]: # Evaluate
print(confusion_matrix(y2_test, model.predict(X_test)))

144/144 ━━━━━━━━━━━━━━━━━━━━ 1s 5ms/step
Pred BASEL_pleasant_weather
True
BASEL_pleasant_weather 2955
BELGRADE_pleasant_weather 879
BUDAPEST_pleasant_weather 162
DEBILT_pleasant_weather 64
DUSSELDORF_pleasant_weather 25
HEATHROW_pleasant_weather 67
KASSEL_pleasant_weather 9
LJUBLJANA_pleasant_weather 46
MAASTRICHT_pleasant_weather 7
MADRID_pleasant_weather 360
MUNCHENB_pleasant_weather 8
OSLO_pleasant_weather 4
STOCKHOLM_pleasant_weather 3
VALENTIA_pleasant_weather 1</td><td>The CNN model only predicts for Basel weather station(?), also worth noting that the model accuracy was better the original h.params, however, we had a ‘stop iteration’ error. (learn more here)</td></tr>
</table>

Notes: For the random forest model that handles all weather stations and their hyperparameters, the gridsearch and randomsearch reveal that it was 3% less predictive than the original h.params set in 2.3

Observations from Previous Models:

- **Random Forest Importance:** Basel, Belgrade, and Madrid were identified as crucial variables. For each of these stations, the top features varied, indicating the importance of location-specific factors.
- **Basel:** Temperature max and mean, and precipitation were crucial.
- **Belgrade:** Precipitation, temperature max, and mean were important.
- **Madrid:** Temperature max, mean, and precipitation were significant.
- Cloud cover, pressure, and humidity had low importance across all stations.

Recommendations for Air Ambulance:

- Given the importance of temperature, particularly maximum and mean temperatures, it's essential for the Air Ambulance to monitor temperature trends closely.
- Precipitation is another crucial variable, especially for Basel and Belgrade stations. High precipitation levels might indicate adverse weather conditions for flying.
- While cloud cover, pressure, and humidity have low importance overall, they shouldn't be ignored entirely. These variables could still contribute to local weather conditions, especially in combination with other factors.

Iterations: Continue refining the Random Forest model as the baseline, and experiment with CNNs or RNNs if we suspect spatial/temporal patterns are critical. Focus on optimizing hyperparameters and feature selection to improve model performance and interpretability.

Summary: After reevaluating the hyperparameters for the CNN model, there were accuracies as high as 97% on training data but with test data it was around 65% with converging loss below 2% but early stopping was enabled.

The original random forest model [n_estimators=100] is still an optimal choice for predicting pleasant weather days in Europe for ClimateWins; being approx. 90% accurate, utilizing minimal parameter adjustment. 88% is an acceptable score, but in a real-life scenario, the costof error can be life or death; so more model tuning.