CLIMATEWINS
PROJECT SUMMARY

CLIMATE WINS

Keanu Gomes March, 2024
CareerFoundry Student

I. PROJECT OBJECTIVE AND HYPOTHESES

Tableau Dashboard

GOAL OBJECTIVE

Apply optimization algorithms, supervised and unsupervised machine learning techniques to predict the consequences of climate change as a data analyst at the European non profit organization, ClimateWins.

HYPOTHESES THAT CAN BE PROPOSED FROM THIS DATA:

1. Which algorithm predicts pleasant weather days best?
2. Will warmer temperatures correlate positively with the occurrence of pleasant weather days?
3. Does higher global radiation correspond to increased temperatures in cities?

II. DATA ETHICS

DATA SOURCE

https://www.ecad.eu

BIAS TYPES

Selection Bias

Only18 out of 26321 weather stations were chosen as sample data

DATA ACCURACY

The data selected for this analysis comes from reliable and trustable sourcing, as is it from 87 participants from verified meteorological stations across Europe totaling 26321 weather stations and 13 characteristics to be analyzed.

DATA DIMENSIONS

22,951 rows x 170 columns

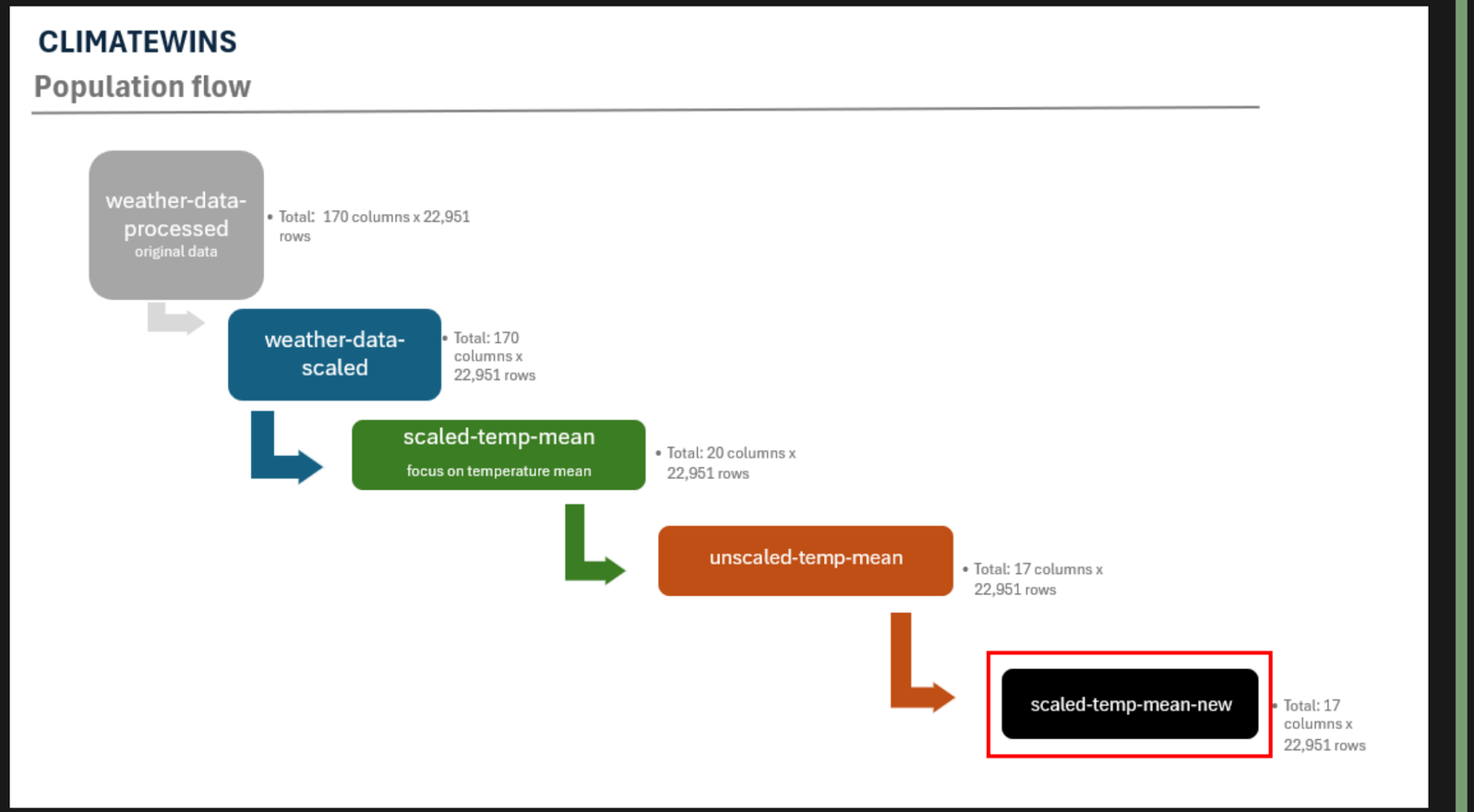
18

Total weather stations

III. DATA WORKFLOW

TEMPERATURE (MEAN) FOCUSED ANALYSIS

- Displayed on the right, is the population flow of my analysis through the (already) processed & cleaned data received from the weather stations.
- In order to feed the data into our supervised learning algorithms, we must removed non-pertinent columns & scale the data in order to normalize it for a more accurate analysis.



GRADIENT DESCENT

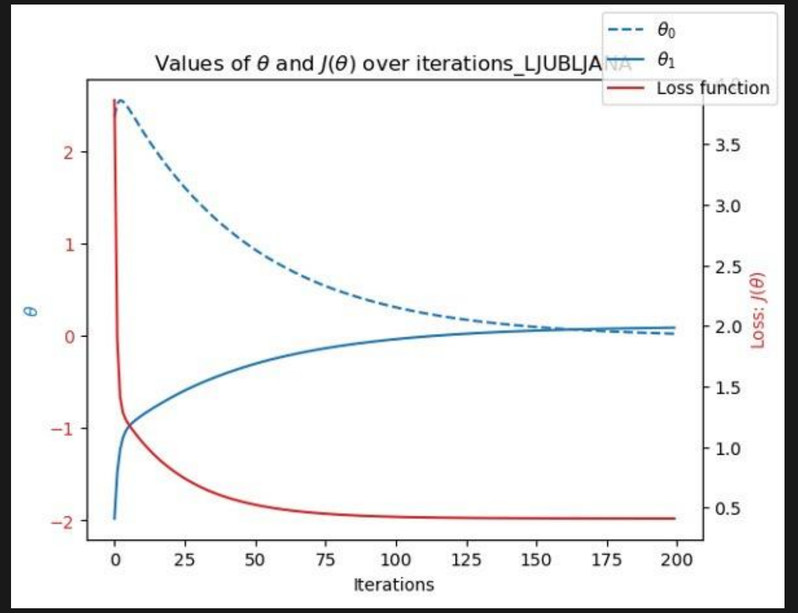
IV. How was optimization used to determine the features of this data set?

- Figured top right, indicates that the model has reached a point of optimization where it has likely found a good set of parameters that minimize the prediction error.

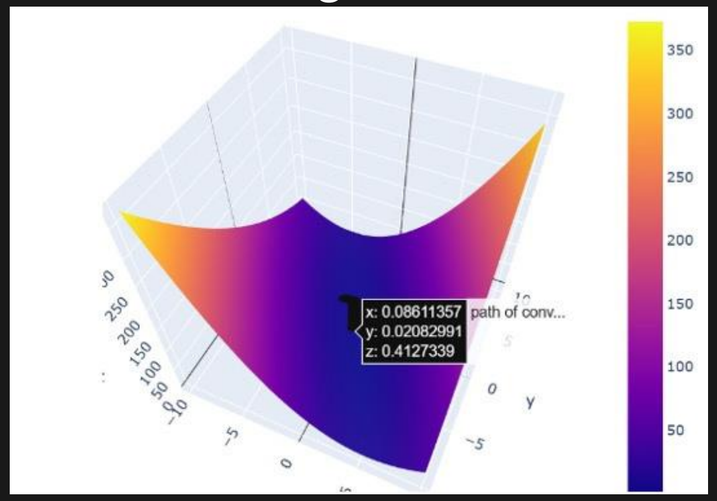
- Figured bottom left, the path converged at a minimum value of 0.4 for the cost function, and the parameters (theta) also converged towards a minimum.

- Figured below, is a screenshot of the ending parameters I used for optimizing the algorithm.

Loss function



Path of convergeance



Parameters used

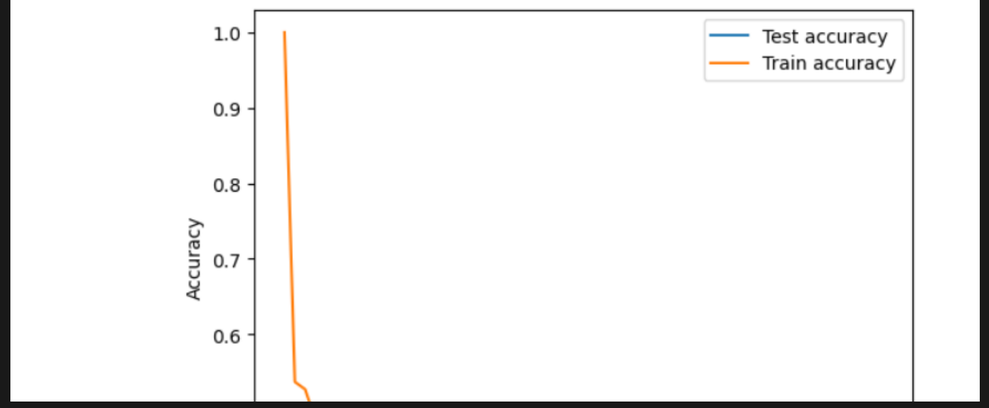
```
num_iterations=200 #<---use same iterations
theta_init=np.array([[2],[-3]]) #<---make a guess [x],[y]
alpha= 0.1 #<---use the same step size
theta3, J_history3, theta0_history3, theta1_history3 = gradient_descent(X,y, theta_init,
                                                                        alpha, num_iterations)
```

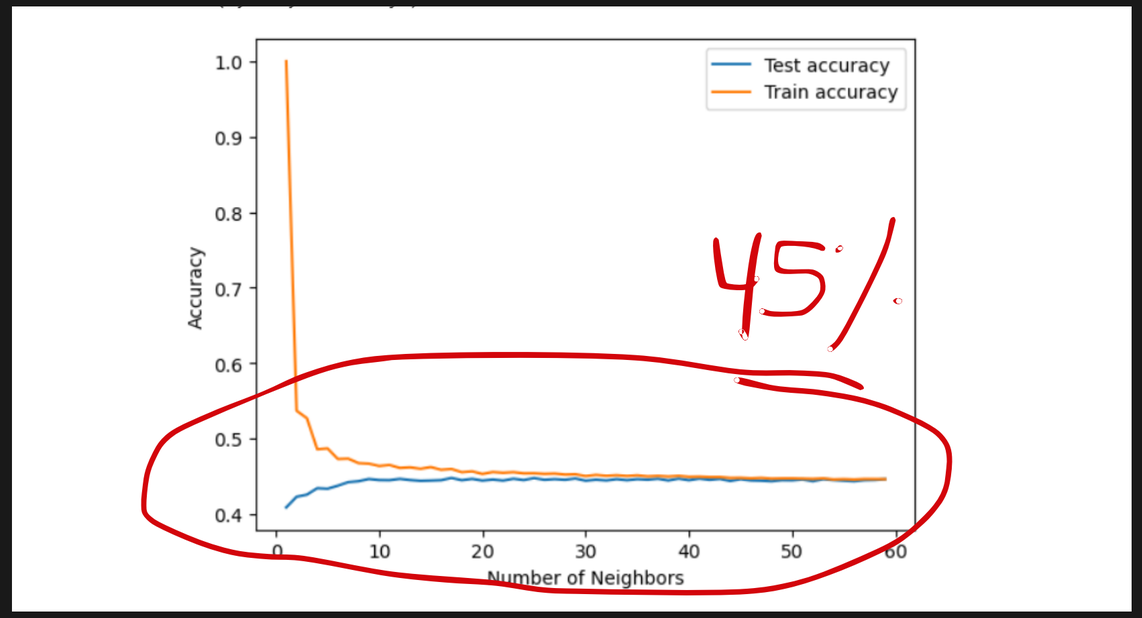
V. ACCURACY SCORES

KNN
{K-NEAREST NEIGHBORS}

```
In [20]:
#plot the relationship between number of neighbors and accuracy
plt.plot(k_range, test_acc, label = 'Test accuracy')
plt.plot(k_range, train_acc, label = 'Train accuracy')
plt.legend()
plt.xlabel('Number of Neighbors')
plt.ylabel('Accuracy')

Out[20]: Text(0, 0.5, 'Accuracy')
```





- Overall Testing Accuracy: approx. 0.45 or 45%
- Individual Station Accuracy: 0.82 - 0.95 or 82-95%
- Interpretation: Potential overfitting; lower overall accuracy compared to individual station scores.

V.I. ACCURACY SCORES

{DECISION TREE}

In [12]:
```
#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:  0.40519344719414 43
```

Out[12]:
```
array([[[3735,  603],
        [ 555,  845]],

       [[3143,  633],
        [ 622, 1340]],
```

- Overall Testing Accuracy: approx. 0.405 or 40%
- Individual Station Accuracy: 0.82 - 0.95 or 82-95%
- Interpretation: Potential overfitting; lower overall accuracy compared to individual station scores.

V.II. ACCURACY SCORES

ANN

{ARTIFICIAL NEURAL NETWORK}

```
2 mlp = MLPClassifier(hidden_layer_sizes=(20, 10, 10), max_iter=1000, tol=0.0001) #increasing hidden layers
3 #Fit the data to the model
4 mlp.fit(X_train, y_train)
```

Out[31]:

MLPClassifier

MLPClassifier(hidden_layer_sizes=(20, 10, 10), max_iter=1000)

In [32]:

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

```
0.45044155240529865
0.4527710003485535
```

- Overall Testing Accuracy: approx. 0.452 or 45%
- Individual Station Accuracy: 0.82 - 0.95 or 82-95%
- Interpretation: Potential overfitting; lower overall accuracy compared to individual station scores.

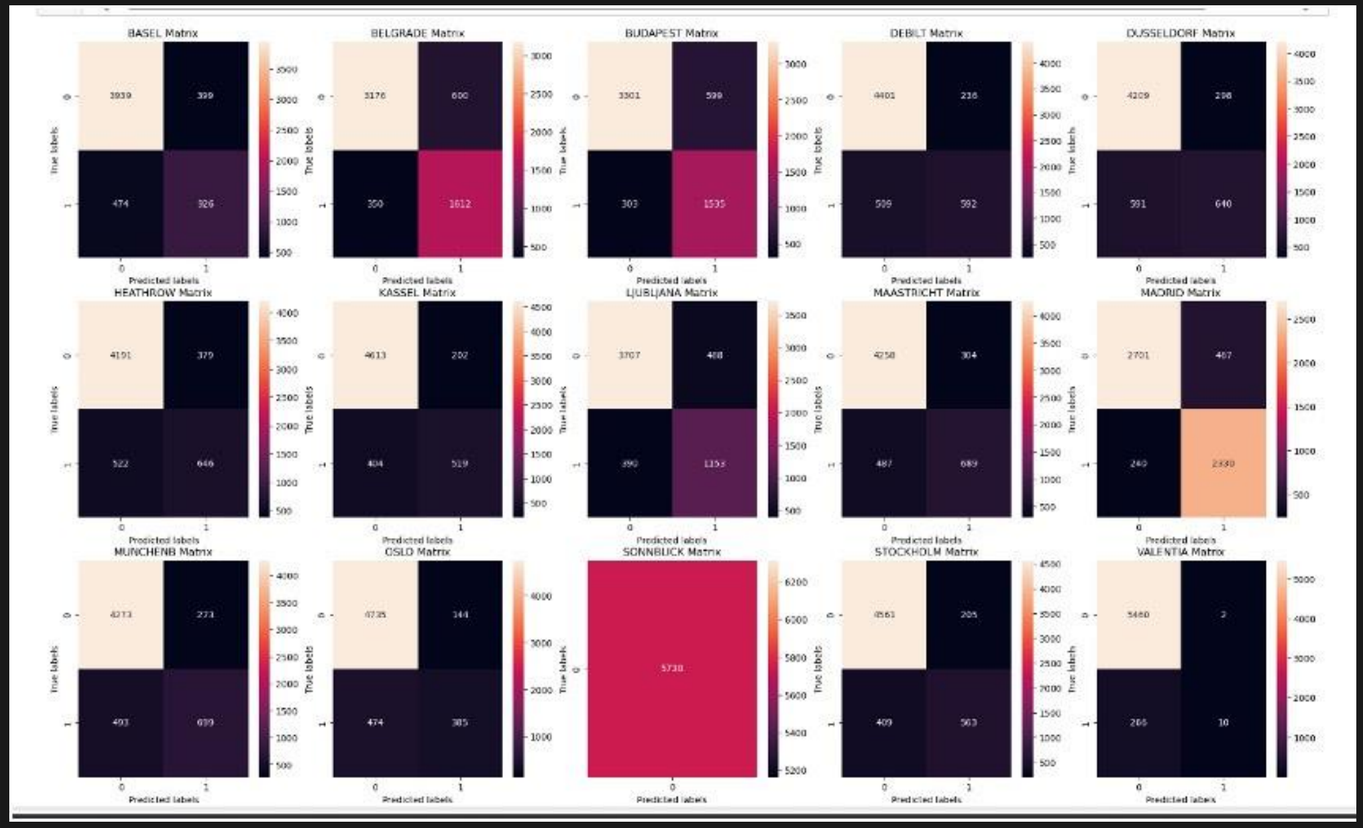
KNN/ANN/ OR DECISION TREE?

VI. 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)



- 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.

VII. Which supervised learning algorithm types will be most effective for our hypotheses?

PRIMARY RECOMMENDATIONS

CONSISTENT TREND:

40-45%
Overall Accuracy

82-100%
Individual Station Accuracy

VALENTIA STANDS OUT:

95%

Achieves high accuracy scores consistently around

RECOMMENDATIONS:

- Investigate data quality and potential biases.
- Conduct feature importance analysis to leverage Valentia's strengths.
- Continue model refinement for improved accuracy.

SUMMARY:

- Engage stakeholders to discuss implications and actions.
- All models show potential overfitting with lower overall accuracy compared to individual station scores.
- Further analysis and model refinement are necessary to address overfitting and improve generalization performance.

THANKS FOR FOLLOWING ALONG!

Any Questions?
Please contact me below at
keanudatatech@gmail.com

or visit:
https://keanudatatech.github.io/portfolio