A Data-Driven Approach to Weather Forecasting

Using Machine Learning to Predict Picnic Suitability

MARIKA MACAWILE 08.20.2024

Objective and Hypotheses

OBJECTIVE:

- **TO DEVELOP AND EVALUATE MACHINE LEARNING MODELS THAT PREDICT PICNIC SUITABILITY BASED ON WEATHER DATA FROM VARIOUS STATIONS**

- **HYPOTHESES:**
  1. **CERTAIN WEATHER STATIONS PROVIDE MORE ACCURATE PREDICTIONS DUE TO THEIR GEOGRAPHIC LOCATIONS AND DATA QUALITY.**
  2. **THE DECISION TREE MODEL WILL OUTPERFORM K-NEAREST NEIGHBORS (KIN) AND ARTIFICIAL NEURAL NETWORKS (ANN) IN PREDICTING PICNIC SUITABILITY DUE TO ITS ABILITY TO HANDLE COMPLEX INTERACTIONS IN THE DATA.**
  3. **SCALING THE DATA WILL SIGNIFICANTLY IMPROVE THE PERFORMANCE OF MODELS LIKE KEN AND ANN.**

Data Sources, Biases, and Accuracy

- DATA SOURCES:
  - WEATHER DATA: PROVIDED BY MULTIPLE WEATHER STATIONS, CAPTURING VARIABLES SUCH AS TEMPERATURE, HUMIDITY, AND WIND SPEED.
  - PICNIC SUITABILITY DATA: LABELS INDICATING WHETHER CONDITIONS WERE SUITABLE FOR PICNICS, BASED ON HISTORICAL OBSERVATIONS.
- POTENTIAL BIASES:
  - GEOGRAPHICAL BIASES: STATIONS IN DIFFERENT REGIONS MAY HAVE VARYING DATA QUALITY AND RELEVANCE.
  - TEMPORAL BIAS: SEASONAL VARIATIONS MAY INTRODUCE BIASES IF NOT PROPERLY ACCOUNTED FOR IN THE MODEL.
  - LABELING BIAS: SUBJECTIVITY IN DETERMINING “SUITABILITY” FOR PICNICS COULD INFLUENCE THE ACCURACY OF LABELS.
- ACCURACY OF THE DATA:
  - WEATHER DATA: GENERALLY ACCURATE BUT MAY VARY BY STATION.
  - SUITABILITY LABELS: POTENTIAL INACCURACIES DUE TO HUMAN JUDGMENT.

Optimization and Feature Selection

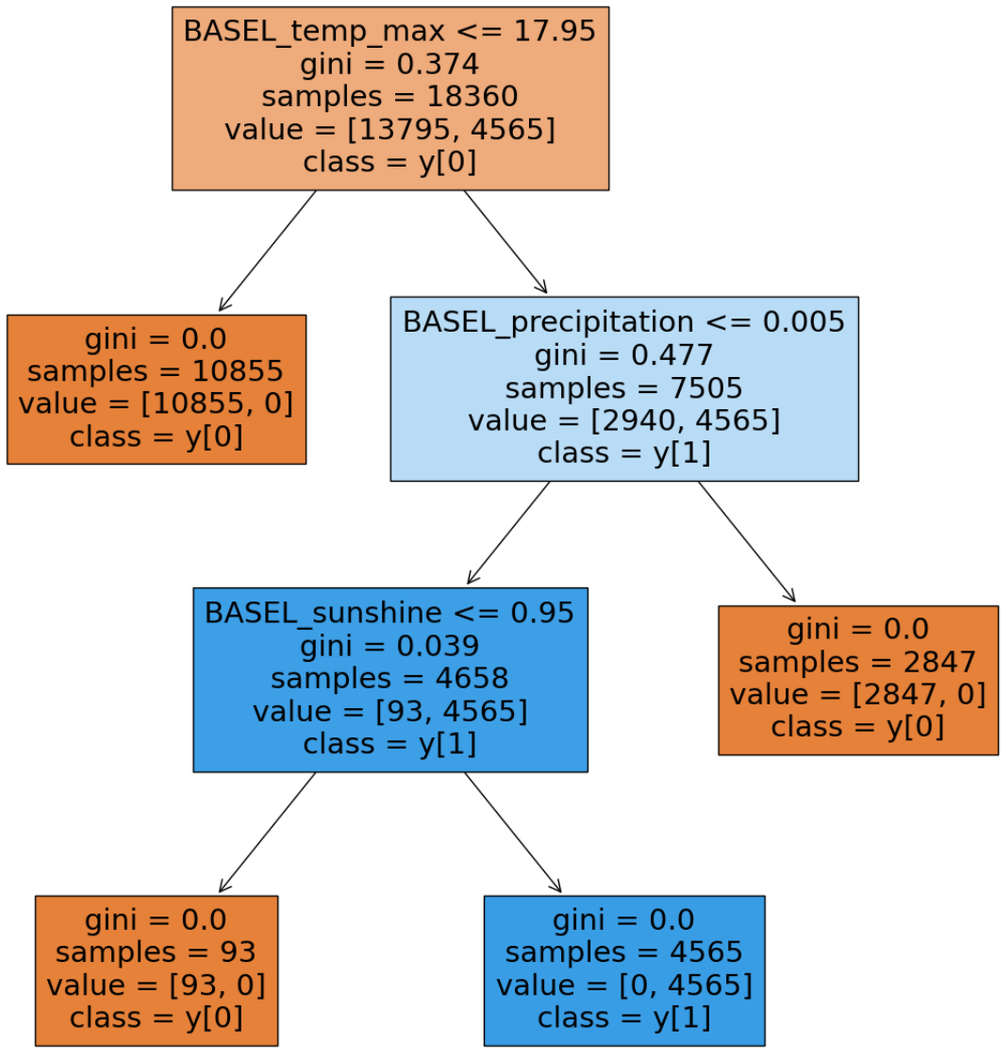
- Feature Optimization:
  - Tested scaling techniques to determine the impact on model performance.
  - Evaluated different configurations (number of layers, nodes, max iterations, tolerance) for the ANN model to optimize performance.
- Outcome:
  - Scaling improved performance for KNN and ANN models.
  - The optimal configuration for ANN was a single hidden layer with 50 nodes, 1000 max iterations, and 0.0001 tolerance.

```
/opt/anaconda3/lib/python3.11/site-packages/sklearn/neural_network/_multilayer_perceptron.py:686: ConvergenceWarnin
g: Stochastic Optimizer: Maximum iterations (200) reached and the optimization hasn't converged yet.
  warnings.warn(

Layers: (50,), Max Iter: 200, Tol: 0.0001 - Training Accuracy: 0.6886710239651416, Test Accuracy: 0.612200435729847
5

/opt/anaconda3/lib/python3.11/site-packages/sklearn/neural_network/_multilayer_perceptron.py:686: ConvergenceWarnin
g: Stochastic Optimizer: Maximum iterations (200) reached and the optimization hasn't converged yet.
  warnings.warn(

Layers: (50,), Max Iter: 200, Tol: 0.001 - Training Accuracy: 0.6886710239651416, Test Accuracy: 0.6122004357298475
Layers: (50,), Max Iter: 200, Tol: 0.01 - Training Accuracy: 0.6202614379084967, Test Accuracy: 0.5751633986928104

/opt/anaconda3/lib/python3.11/site-packages/sklearn/neural_network/_multilayer_perceptron.py:686: ConvergenceWarnin
g: Stochastic Optimizer: Maximum iterations (500) reached and the optimization hasn't converged yet.
  warnings.warn(

Layers: (50,), Max Iter: 500, Tol: 0.0001 - Training Accuracy: 0.7671568627450981, Test Accuracy: 0.64880174291939

/opt/anaconda3/lib/python3.11/site-packages/sklearn/neural_network/_multilayer_perceptron.py:686: ConvergenceWarnin
g: Stochastic Optimizer: Maximum iterations (500) reached and the optimization hasn't converged yet.
  warnings.warn(

Layers: (50,), Max Iter: 500, Tol: 0.001 - Training Accuracy: 0.7671568627450981, Test Accuracy: 0.64880174291939
Layers: (50,), Max Iter: 500, Tol: 0.01 - Training Accuracy: 0.6202614379084967, Test Accuracy: 0.5751633986928104
Layers: (50,), Max Iter: 1000, Tol: 0.0001 - Training Accuracy: 0.8133986928104575, Test Accuracy: 0.67581699346405
23
Layers: (50,), Max Iter: 1000, Tol: 0.001 - Training Accuracy: 0.7986383442265795, Test Accuracy: 0.667973856209150
3
Layers: (50,), Max Iter: 1000, Tol: 0.01 - Training Accuracy: 0.6202614379084967, Test Accuracy: 0.5751633986928104

/opt/anaconda3/lib/python3.11/site-packages/sklearn/neural_network/_multilayer_perceptron.py:686: ConvergenceWarnin
g: Stochastic Optimizer: Maximum iterations (200) reached and the optimization hasn't converged yet.
  warnings.warn(

Layers: (100,), Max Iter: 200, Tol: 0.0001 - Training Accuracy: 0.7766339869281046, Test Accuracy: 0.60370370370370
37

/opt/anaconda3/lib/python3.11/site-packages/sklearn/neural_network/_multilayer_perceptron.py:686: ConvergenceWarnin
g: Stochastic Optimizer: Maximum iterations (200) reached and the optimization hasn't converged yet.
  warnings.warn(

Layers: (100,), Max Iter: 200, Tol: 0.001 - Training Accuracy: 0.7766339869281046, Test Accuracy: 0.603703703703703
7
Layers: (100,), Max Iter: 200, Tol: 0.01 - Training Accuracy: 0.7058823529411765, Test Accuracy: 0.5880174291938998

/opt/anaconda3/lib/python3.11/site-packages/sklearn/neural_network/_multilayer_perceptron.py:686: ConvergenceWarnin
g: Stochastic Optimizer: Maximum iterations (500) reached and the optimization hasn't converged yet.
  warnings.warn(

Layers: (100,), Max Iter: 500, Tol: 0.0001 - Training Accuracy: 0.8662854030501089, Test Accuracy: 0.61481481481481
48
Layers: (100,), Max Iter: 500, Tol: 0.001 - Training Accuracy: 0.836764705882353, Test Accuracy: 0.6080610021786492
```

Supervised Learning Models Used

- **Models Evaluated:**

1. **K-Nearest Neighbors (KNN):**
   - Easy to implement but prone to overfitting with low k.
   - Test accuracy plateaued around 47%, indicating limited generalization.
2. **Decision Tree:**
   - Achieved perfect training accuracy but showed overfitting with a test accuracy of 64%.
   - Recommended pruning to reduce overfitting and improve generalization.
3. **Artificial Neural Network (ANN):**
   - Best configuration achieved 67.58% test accuracy.
   - Required careful tuning to avoid convergence issues.

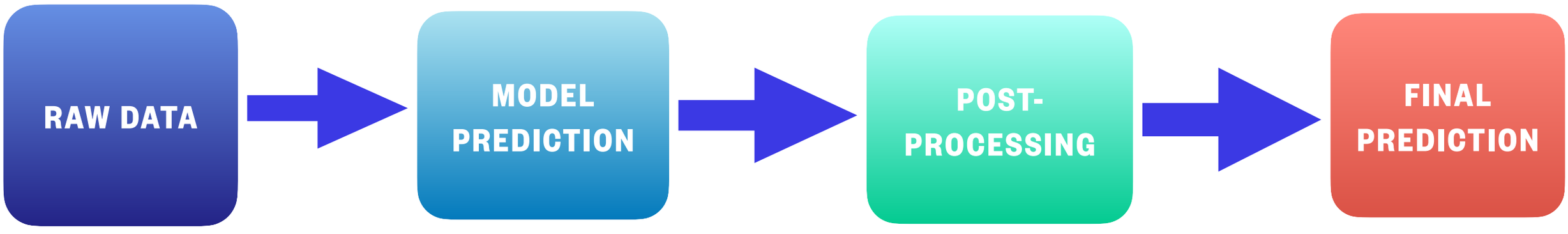


Decision Tree

- **Most Effect Model:**
  - Decision Tree, with pruning, due to its balance between accuracy and interpretability.

Post-Processing and Efforts Required

- **Post-Processing Efforts:**
    - **Bias Mitigation:** Addressing geographical and temporal biases by including additional features or adjusting data collection methods.
    - **Model Pruning:** Essential for the Decision Tree to reduce overfitting.
    - **Data Cleaning:** Ensuring data is consistent across all stations to improve model accuracy.
    - **Hyperparameter Tuning:** Ongoing effort to optimize models, particularly for ANN, to balance training time and accuracy.



Summary and Next Steps

- Summary of Findings:
    - Hypotheses Supported: The Decision Tree model, with appropriate pruning, is most effective for predicting panic suitability.
    - Optimization Success: Scaling and careful hyper parameter tuning significantly impacted model performance.
    - Challenges: Overfitting, convergence issues, and data biases need ongoing management.
- Next Steps:
    - Implement pruning on the Decision Tree model to improve generalization.
    - Perform cross-validation to further validate the models.
    - Explore additional data sources or features to enhance model accuracy.
- Future Analysis:
    - Investigate the impact of incorporating real-time data and extending the model to predict longer-term suitability trends.

Thank you

MARIKA MACAWILE