

Outdoor Air Temperature Prediction

Phase Three

CSCE 3602 Fundamentals of Machine Learning

Amr Abdelbaky
Computer Science
The American University In Cairo
Cairo, Egypt
amrkhaled122@aucegypt.edu

Ali Eissa
Computer Science
The American University In Cairo
Cairo, Egypt
alieissa@aucegypt.edu

Before Starting This Milestone

According to the provided feedback from milestone 2, editions were made to our dataset. To be precise, we changed the method of scaling used from a mixture of MinMax and Normal scaling to only Normal scaling, in addition to that, also we did one hot encoding to the Region feature and concatenated it with our dataset as opposed to removing it from the dataset, so currently we have a total of 29 features as of the start of this milestone (it was 22 but the regions column had 7 unique values) and we have 8,392,299 sample in the whole dataset.

A full description experimental analysis for each of the supervised machine learning models you tried (experimental setup, parameter choice, and performance evaluation)

a. Experimental Setup and Parameter Choices:

- General assumptions :

1. We split the dataset into testing and training with 80% given to training and 20% given to testing
2. For using GridSearch with cross validation, the k-folds were set to a value of 5.
3. When Using K-NN algorithm, the optimal K value was assumed to be the square root of the sample size (\sqrt{n}).
4. When Using Random Forests, the n-estimator was set to the default value of 100 and we tried another run with 200 estimators..
5. During transforming the problem from a regression to classification, the range of the target variable was [-9,47] so we binned the target variable into 6 different classes which we saw fit

for classifying the weather with the following ranges:

6. For Multilayer Perceptron (Neural Network) the number of Epochs was set to 10 and 30 and other values were defaulted .
 - [-10- 0] = Very Cold.
 - [0-10]= Cold.
 - [10-20] = Moderate.
 - [20-30] = Warm.
 - [30-40] = Hot.
 - [40-50]=Very Hot.
- **Important Notes :**
1. SVM model (SVR for regression and SVC for classification) was not used because even if we installed the packages that utilize the GPU which should be faster than 80% of the normal CPU implementation, it took 6 hours running the model and it never finished due to a crash usually happening after 5 or more hours have passed since starting the model fitting, we have later discovered that with such a large sample size (8,229,000 sample) SVM is not feasible and might even take several days to fit.
2. For Testing KNN-s Models, a random sample of 15,000 was taken from the testing set which comprises 20% of the whole dataset split due to the long time it takes for testing KNN-s and it was evaluated based on this sample.
3. Models where we can utilize GPU implementations were used to reduce the time for training.

b. Performance evaluation :

- Regression Models :

- Regression Models were all evaluated according to Their MSE(Mean Squared Error) and R2 (Goodness of fit).

1. Linear Regression :

MSE	R2
1.2976516446600896	0.8988394259704199

2. KNN-s Regressor (test sample 15,000 only) :

MSE	R2
2.4946026690537204	0.8041505931377732

3. Decision Trees Regressor :

MSE	R2
0.10388761598921503	0.9919012695654646

4. Random Forests Regression :

MSE	R2
0.1077971791738604	0.9915964931198051

5. MultiLayer Perceptron (Neural network) with linear activation :

MSE	R2
0.23509007857745792	0.9816731652170146

- **Classification Models :**


1. Logistic Regression (Multinomial Classification):

Confusion Matrix:					
[0	0	0	0	521]
[0	0	0	0	310646]
[0	0	0	0	31193]
[0	0	0	0	4]
[0	0	0	0	33]
[0	0	0	0	1336063]
Classification Report:					
	precision	recall	f1-score	support	
Cold	0.00	0.00	0.00	521	
Hot	0.00	0.00	0.00	310646	
Moderate	0.00	0.00	0.00	31193	
Very Cold	0.00	0.00	0.00	4	
Very Hot	0.00	0.00	0.00	33	
Warm	0.80	1.00	0.89	1336063	
accuracy			0.80	1678460	
macro avg	0.13	0.17	0.15	1678460	
weighted avg	0.63	0.80	0.71	1678460	

2. KNN-s Classifier (test sample 15,000 only) :

```
Precision: 0.7922935528942436
Recall: 0.8122
Accuracy: 0.8122
```

3. Decision Tree Classifier :



Confusion Matrix:

[

[

[

[

[

[

512

0

10

2

0

5

0

304034

11

0

22

6449

0

9

30354

0

0

818

2

0

1

0

0

0

0

13

0

0

11

0

2]

6590]

817]

2]

0]

1328791]]

Classification Report:

precision

recall

f1-score

support

Warm

Hot

Moderate

Very Hot

Cold

Very Cold

0.97

0.98

0.97

0.00

0.46

0.99

0.98

0.97

0.00

0.33

0.99

0.98

0.98

0.97

0.00

0.39

0.99

521

310646

31193

4

33

1336063

accuracy

macro avg

weighted avg

0.73

0.99

0.71

0.99

0.99

0.99

1678460

1678460

1678460

4. Random Forests Classifier :

Confusion Matrix:						
[503	3	12	1	0	2]
[0	302395	31	0	0	8220]
[5	5	29584	0	0	1599]
[2	0	0	0	0	2]
[0	33	0	0	0	0]
[0	8489	702	0	0	1326872]]
Classification Report:						
	precision	recall	f1-score	support		
Cold	0.99	0.97	0.98	521		
Hot	0.97	0.97	0.97	310646		
Moderate	0.98	0.95	0.96	31193		
Very Cold	0.00	0.00	0.00	4		
Very Hot	0.00	0.00	0.00	33		
Warm	0.99	0.99	0.99	1336063		
accuracy			0.99	1678460		
macro avg	0.65	0.65	0.65	1678460		
weighted avg	0.99	0.99	0.99	1678460		

5. MultiLayer Perceptron (Neural network) with Softmax activation :

```
Precision: 0.7707001955834194
Recall: 0.7960428011391395
F1-score: 0.7057212545787996
Accuracy: 0.7960428011391395
```

6.

7. Naive bayes Classifier :

Confusion Matrix:						
[489	5	14	0	0	13]
[0	270663	419	5	5073	34486]
[563	1627	10942	0	56	18005]
[2	0	0	1	0	1]
[0	10	0	0	21	2]
[1272	156716	49332	68	4355	1124320]
Classification Report:						
	precision	recall	f1-score	support		
Warm	0.21	0.94	0.34	521		
Hot	0.63	0.87	0.73	310646		
Moderate	0.18	0.35	0.24	31193		
Very Hot	0.01	0.25	0.03	4		
Cold	0.00	0.64	0.00	33		
Very Cold	0.96	0.84	0.89	1336063		
accuracy			0.84	1678460		
macro avg	0.33	0.65	0.37	1678460		
weighted avg	0.88	0.84	0.85	1678460		

A comment on each of the model's performance, and its fit for the problem at hand (pros, cons, etc.)

- Regression Models :

- The problem is highly attributed as a regression model due to the fact that the label is a continuous value (Air temperature (instant) in celsius degree) and also about 21 of the features are continuous values and 7 values were one hot encoded , this will make us favor given that its results are fairly acceptable.
1. **Linear Regression :** A linear regression model is simple and looks fairly acceptable in the scope of our problem, however, due to the fact that it assumes a linear relationship between target variable and features , which may not always be true , this resulted in it ranking at the 4th place compared to other regression models that were used .
 2. **KNN-s Regressor (test sample 15,000 only) :** easy to implement and captures complex relationships between features and target variable ,However , Computationally expensive during prediction, especially with large datasets which was our case and why we chose to test on a random 15,000 sample from the whole test sample which took about 3 hours to test , this model came in the last place.
 3. **Decision Trees Regressor :** one of its major pros is that it can capture non-linear relationships and interactions , however one of its major problems is that it is prone to overfitting ,especially with deep trees.
 4. **Random Forests Regression :** Reduces overfitting compared to a single decision tree by aggregating predictions from multiple trees. It handles high-dimensional data well and is robust to outliers. However , It can be computationally expensive and memory-intensive, especially with a large number of trees (we couldn't test with $n\text{-estimators} = 300$) .
 5. **MultiLayer Perceptron (Neural network) with Relu activation :** Can capture complex non-linear relationships between features and the target variable. It's highly flexible and can handle large amounts of data,However, It requires tuning of hyperparameters such as learning rate, number of layers, and neurons per layer. It is prone to overfitting in case of small datasets (which is not our case) .
- ### - Classification Models :
- Our target variable which was a continuous variable was replaced by a categorical variable

[Temperature Category] which has 6 classes as we have divided the dataset .

- We had to bin every feature that was binnable in order to use the Naive Bayes model .
1. **Logistic Regression(Multinomial Classification):**similar to linear regression , it is simple and works well for binary and multi-class classification problems. It's suitable for problems with linear decision boundaries.However . Similar to linear regression , it assumes a linear relationship between features and target variable(s), which might not hold true for complex datasets. It may not perform well if the classes are not well-separated.
 2. **KNN-s Classifier (test sample 15,000 only) :**
The same limitations and advantages that were mentioned in KNN-s Regressor .
 3. **Decision Tree Classifier :**
The same limitations and advantages that were mentioned in Decision Tree Regressor .
 4. **Random Forests Classifier :**
The same limitations and advantages that were mentioned in Random Regressor .
 5. **MultiLayer Perceptron (Neural network) with Softmax activation :**
The same limitations and advantages that were mentioned in MultiLayer Perceptron (Neural network) with Relu activation .
 6. **Naive Bayes Classifier :** Simple and computationally efficient. Performs well with high-dimensional data and large datasets. Robust to irrelevant features.Cons: Assumes independence between features, which might not hold true for all datasets. May not capture complex relationships between features and target variables. Limited expressiveness compared to more complex models like neural networks.

A comparative analysis of the performance of the top 3 models :

a. top 3 models:

1. Decision Tree Regressor
2. Random Forest Regressor
3. Neural Network with Relu Activation Model
- 4.

b. Parameter Choices:**Decision Tree Regressor**

Depth	20
criterion	squared_error
splitter	best
min_samples_split	2
min_samples_leaf	1

Random Forest Regressor

n_estimators	100
max_depth	16
max_features	'auto', which means it considers all features for each split.
min_samples_split	2
min_samples_leaf	1
bootstrap	True

Neural Network Relu activation

n_layers	64
activation	relu
batch_size	32
optimizer	Adam
epochs	30
learning rate	0.001

c. Performance evaluation :**Decision Tree Regressor**

MSE	R2
0.10388761598921503	0.9919012695654646

Random Forest Regressor

MSE	R2
0.1077971791738604	0.9915964931198051

Neural Network linear activation

MSE	R2
0.21225622438259276	0.9834532159780687

Since Regression Models are best suited from our problem, and on avg they were the most performant, we chose the top 3 from that category.

There is a very slight performance advantage of decision Trees vs Random Forest Regressors.

Both however have a noticeable advantage compared to a Neural Network with Linear activations.

A choice for a machine learning technique from the top 3 models, weighing in the performance, the nature of the model, and its fit to the problem you chose.

a. The Choice for a machine learning technique

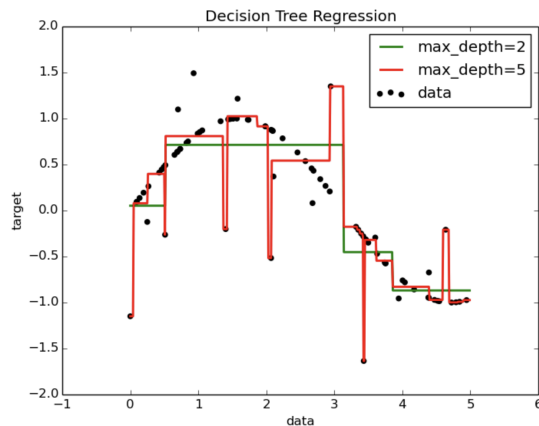
As the Decision Tree model is the best performing, in addition to it being not as memory intensive as the Random forest. it stands as the most suitable choice for our model. However, since we are aware of its tendency to over-fit we will keep that in mind when approaching our strategy,

possibly pruning or having some hybrid between Random Forests and Decision Tree strategies.

b. The nature of the model

Decision trees in regression problems are used to estimate the positions on theoretically continuous functions.

This example shows it being fitted to approximate data following a sine wave [1]:



As we can see however, The higher the max depth of the tree, the more it is prone to overfitting.

c. How it Fits the problem

The Decision Tree Regression model is a good fit for our problem due to its versatility in handling different types of data and its ability to capture complex relationships between features and the target variable, and is good enough for our goal of estimating air temperature based on other weather features.

Its interpretability makes it easy to understand and explain predictions. Moreover, its computational efficiency makes it suitable for scenarios with limited resources or requiring real-time predictions. By addressing overfitting

concerns, we can leverage the strengths of Decision Trees to achieve accurate and robust predictions for our machine learning tasks.

Note for the Notebooks submitted:

There are two submitted notebooks with the models output , the primary notebook is ML_Pilot_Study and the secondary one is ML_Pilot_Study_Another_models , the secondary contains some models that that were mentioned here in this report but are not included in the primary notebook due to a difference in the running environment.

References

- [1] "Decision Tree Regression." Scikit, scikit-learn.org/0.16/auto_examples/tree/plot_tree_regression.html#example-tree-plot-tree-regression-py. Accessed 8 Apr. 2024.