EMATM0044 - Introduction to AI

Tarun Kumar Bagadi (ko22684)

May 23, 2023

1 Question 1

1.1 Introduction

The Problem statement is to predict the net hourly energy of the plant (PE) using the data collected over six years(2006-2011) from a combined cycle power plant. The data consists of continuous variables of hourly average ambient Temperature(T)Ambient Pressure (AP), Relative Humidity (RH) and Exhaust Vacuum (V). Regression algorithms predict a continuous value from a set of input variables. In this task, the output variable to predict is the net hourly electrical energy output and the input variables are the ambient temperature, ambient pressure, relative humidity, and exhaust vacuum.**The most suitable algorithm for this task would be Regression**

1.2 Methods

Both the mean squared error (MSE) and the mean absolute error (MAE) are measures used to assess the performance of regression models. The MAE is the average absolute difference between the predicted values and the actual values. At the same time, the MSE is the average squared difference between the predicted values and the actual values. MAE is a more robust measure of error than MSE when there are outliers in the data. So, I chose to mean absolute error(MAE) as the performance metric to measure the algorithms because there are few outliers in the (RH) of the data as shown in Figure 1.

$$MAE = \frac{1}{n}\sum_{i=1}^{n}\left|y_{true}^{(i)} - y_{pred}^{(i)}\right| \tag{1}$$

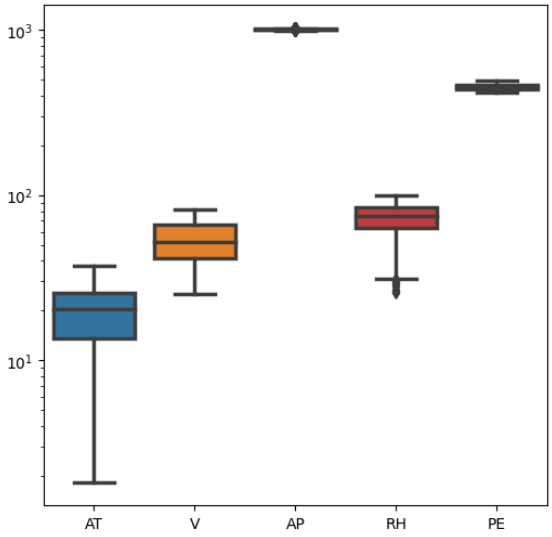

Figure 1: DATA PLOT

The algorithms used for the task Firstly DummyRegressor (Baseline) secondly KNeighborsRegressor and finally DecisionTreeRegressor .

1.2.1 Baseline Model

The Baseline was imported from the sklearn.dummy and it is used to asses the model's performance. The Mean Absolute Error on training data: was 14.78 and Mean Absolute Error on validation data: was 15.03 as shown in the Figure 2

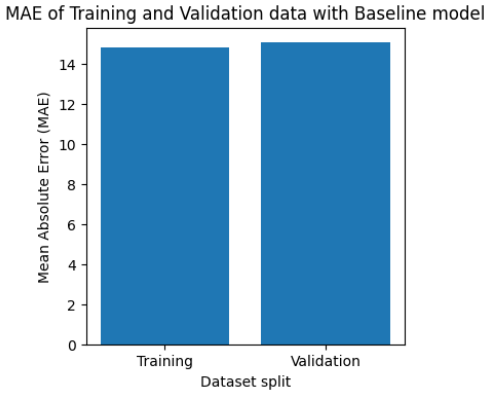

Figure 2: Performance of Baseline Model

1.2.2 KNeighborsRegressor

Imported the regression from KNeighborsRegressor from sklearn.neighbors. For the evaluation of the performance of the model the default values (no values are given to the KNeighborsRegressor). The Figure 4 and FIGURE 5 show the performance of KNN and tuned KNN on training and testing data. Where the tunned KNN over-fits the data and the MAE is zero on the training data.
Hyperparameters The Hyperparameters for KNN regression are

- **n_neighbors**: This hyperparameter sets the number of neighbours to take into account for prediction.The default neighbours are 5.
- **weights**: This hyperparameter sets the weight applied to each neighbour.
- **algorithm**: This hyperparameter determines the algorithm used to compute the nearest neighbours.
- **leaf_size**: This hyperparameter is used for the "ball_tree" or "kd_tree" algorithm. It determines the size of the leaves in the tree data structure used for efficient neighbour finding.

The Figure 3 shows a heatmap of mean absolute error (MAE) with the number of neighbours and weights as the tuned hyperparameters. The tunned KNN parameters are obtained from **GridSearchCV** imported from scikit-learn to determine the optimal set of hyperparameters.

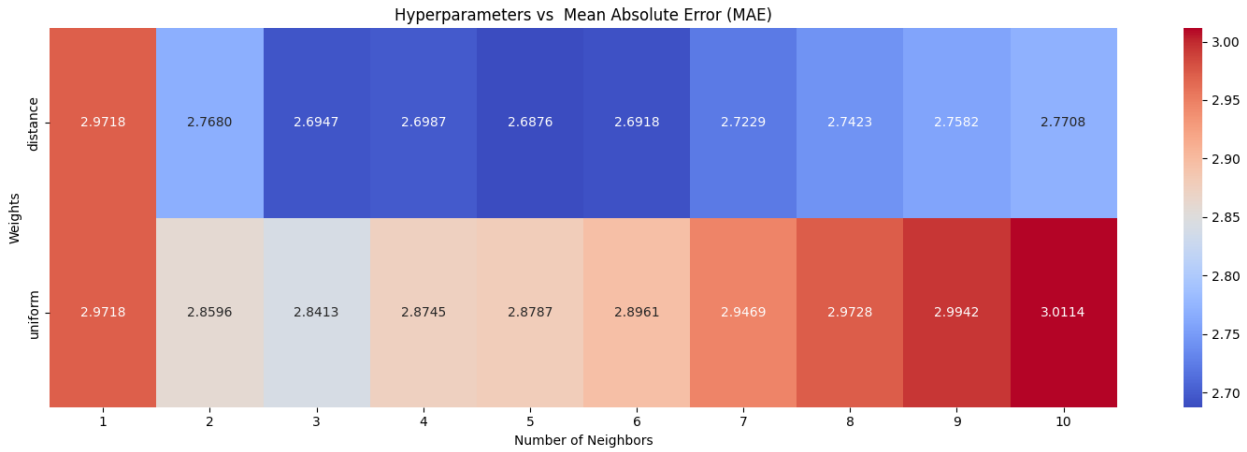

Figure 3: Heat map of Hyperparameter for KNN

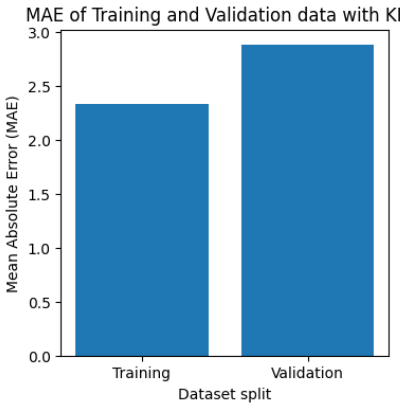

Figure 4: Perfomance of KNN

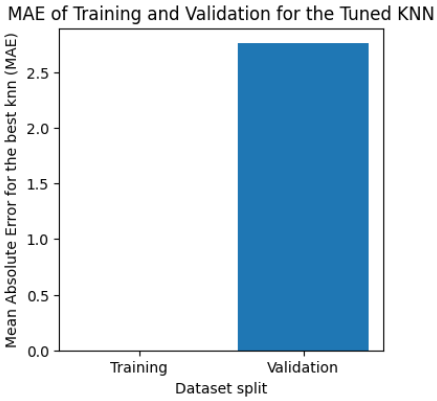

Figure 5: Performance of Tuned KNN

1.2.3 DecisionTreeRegressor

Imported the regression model from DecisionTreeRegressor from sklearn.tree. Predicts the target variable's value using simple decision rules deduced from of the data. They are many other hyperparameters associated with the model but the hyperparameters mentioned below are tuned and the heatmap shows the hyperparameters shown in Figure 6. GridSearchCV is used to find the optimal hyperparameters. The Figure 7 and Figure 8show the performance of the Decision Tree and Tuned Decision Tree on training and test data.

- **max_depth**: This hyperparameter controls the maximum depth of the tree.
- **min_samples_split**: This hyperparameter controls the minimum number of samples required to split a node in the tree.
- **min_samples_leaf**: This hyperparameter controls the minimum number of samples required to be in a leaf node.

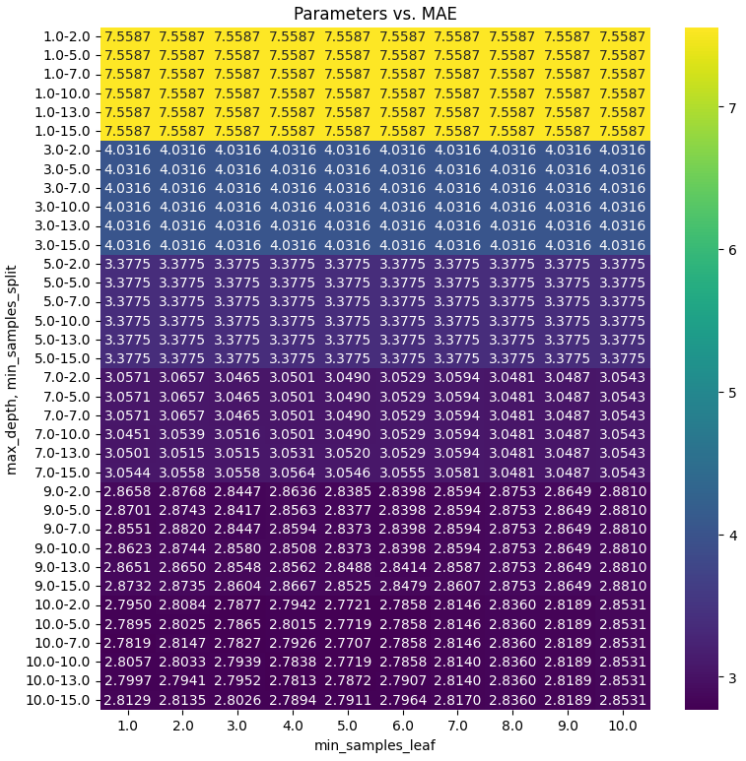

Figure 6: Heat map of Hyperparameter for Decision Tree

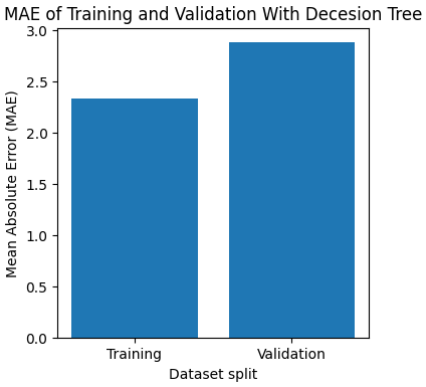

Figure 7: Performance of Decision Tree

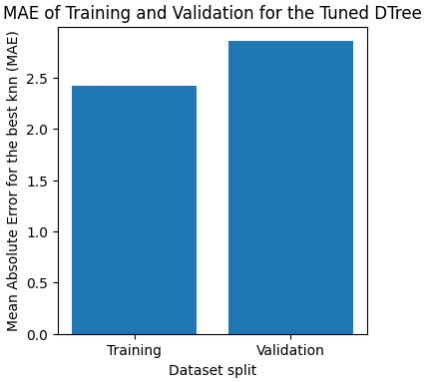

Figure 8: Performance of TunedDTree

1.3 Results and Analysis

From Figure 9 on comparing the MAE values, it was observed that the tuned KNN model achieved the lowest (MAE 2.75), indicating better predictive performance than the normal KNN model (MAE: 2.87) and baseline model (MAE: 15.03).
From Figure10 it is evident that the tuned decision tree regression model outperforms both the baseline and decision tree models. The tuned decision tree model achieved an MAE of 2.853 compared to the untuned decision tree (MAE: 3.088) and baseline (MAE: 15.03), which indicates a high level of accuracy.
From Figure 11 it is clear that the tuned KNN model and the tuned decision tree (DT) model outperform the baseline model. While the tuned KNN model achieved the lowest MAE compared to the tuned decision tree model. This suggests that adjusting the hyperparameters of the algorithm might result in a more accurate model.

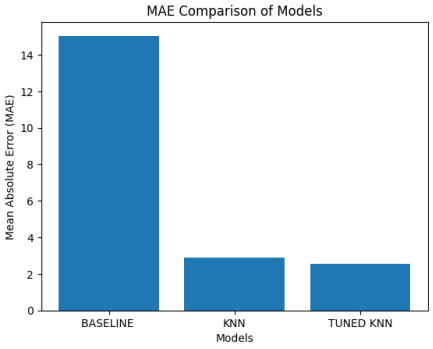

Figure 9: Performance of KNN Models

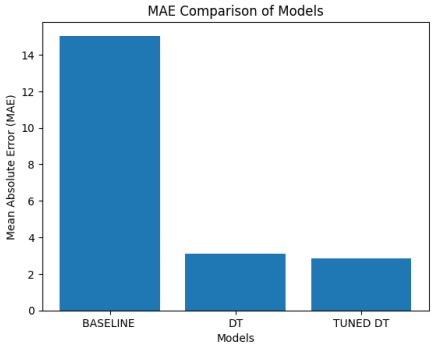

Figure 10: Performance of DT Models

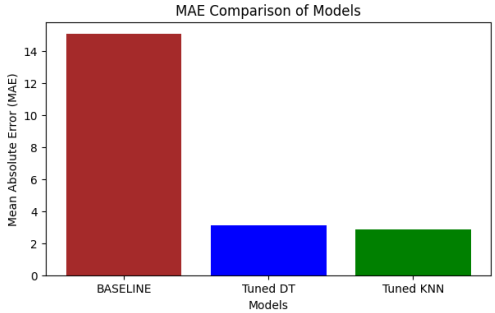

Figure 11: Comparison of Baseline with TunedKNN and TunedDT

2 Question 2: Creating a datasheet

2.1 Composition

1. What do the instances that comprise the dataset represent (e.g., documents, photos, people, countries)? Are there multiple types of instances (e.g., movies, users, and ratings; people and interactions be- tween them; nodes and edges)?

   Ans: The data set contains a variety of adjectives, antonyms, and noun triplets. The transformation was described as a combination of an adjective and a noun. There is a wide range of objects (e.g., fish, persimmon, room) and attributes (e.g., mossy, deflated, dirty)

2. How many instances are there in total (of each type, if appropriate)?

   Ans: The dataset has 245 object classes(nouns), 115 attribute classes(adjectives) and 63,440 images . On average, each object instance is modified by one of the 9 attributes it affords.

3. Does the dataset contain all possible instances or is it a sample (not necessarily random) of instances from a larger set?

   Ans: No, the dataset does not contain all possible instances. Yes, it is a sample from a large number of datasets. A public image database and the internet were used to gather the images for the collection.

4. What data does each instance consist of?

   Ans: Each instance in the dataset consists of the name of the object and different states of the object with the type of transformation applied to the object and image of the object.

5. Is there a label or target associated with each instance? If so, please provide a description.?

   Ans: Yes, there is a label associated with each instance. The directory structure is "A, N", where A is an adjective and N is a noun. Directories where A="adj" contain images where the search string was just the noun "N".The labels are drawn from a set of adjectives.

6. Are there recommended data splits (e.g., training, development/validation, testing)?

   Ans: There are no recommended data splits. The general data split 80-20, can be used where 80% of the data is used for training and 20% is used for testing.

7. Are there any errors, sources of noise, or redundancies in the dataset? If so, please provide a description.

   Ans: The main causes of noise are (adj, noun) pairs being either a product name, a rare combination, or a hard concept to visualised. The dataset contains some errors, such as inaccurate labelling and missing data.

As the data does not relate to people So, skipping the remaining questions in this section.

2.2 Collection Process

1. What mechanisms or procedures were used to collect the data (e.g., hardware apparatuses or sensors, manual human curation, software programs, software APIs)? How were these mechanisms or procedures validated?

   Ans: Used Microsoft's Web N-gram Services to measure the probability of each adj noun phrase that could be created from our lists of adjectives and nouns. Also used these techniques such as Searching for images on the web, web scraping tools and manually searching for images to collect images.

2. Who was involved in the data collection process (e.g., students, crowd workers, contractors) and how were they compensated (e.g., how much were crowd workers paid)?

   Ans: Used online crowdsourcing services to find and label images, web scraping to collect images and manually curated labels for images.

3. Were any ethical review processes conducted (e.g., by an institutional review board)?

   Ans: No review processes were conducted with respect to the collection and annotation of this data

As the data does not relate to people So, skipping the remaining questions in this section

2.3 Uses

1. Has the dataset been used for any tasks already? If so, please provide a description.

   Ans: The dataset is used to study different transformed states of objects, scenes and materials. It is used to study object recognition:

2. Is there a repository that links to any or all papers or systems that use the dataset? If so, please provide a link or other access point.

   Ans: The MIT-States dataset is used by a variety of articles and systems, and there is a repository that links to all `http://web.mit.edu/phillipi/Public/states_and_transformations/index.html`.

3. What (other) tasks could the dataset be used for?

   Ans: The dataset was used for a variety of other tasks such as dynamic scene generation, Object detection and Robot Learning for Manipulation

4. Is there anything about the composition of the dataset or the way it was collected and preprocessed/cleaned/labelled that might impact future uses?

   Ans: Yes, the composition of the dataset does not contain all possible transformations and states images of the objects. So new models cannot rely on the dataset. The dataset is not accurate. Some of the labels are wrong.