Machine Learning Mini Project 1
Humidity Prediction Based on Weather History [1]

Faisal Ahmed Mustafizur Rahman Hilaly

July 25, 2021

[1] Instructor: Anthony Sander

Contents

Abstract

In this project, we are predicting humidity based on historical weather data. To do so, we initially performed Exploratory Data Analysis (EDA) on the dataset to select appropriate features for the machine learning models. We evaluated our dataset on five different classifiers and observed prediction scores from the classifiers. We also plotted Learning curves and Validation curves of 5 different classifiers. To cross-check our findings, we fitted our data to AutoML to get the best estimator from AutoML.

From this experiment, we came up with the best performing estimator for our dataset and then used Grid Search to tune its hyperparameters to get the most optimum result from the model. We then plotted the ROC-AUC curve to visualize the performance of different estimators. Finally, we applied Principal Component Analysis PCA to analyze dimensionality reduction.

1 Exploratory Data Analysis (EDA) & Data Preparation

Initially, we checked the shape, information, and statistics of the dataset to have a better understanding of the data. That helped us to figure out the features and target variables. Before proceeding further we had to clean up the data for use. To do so, we first identified and removed all the unnecessary columns from the dataset, then we checked null values and fill them with approximate values. In our data set, there was a categorical value ("Precip Type") which we encoded to numeric value using one-hot encoding technique.

After the initial processing, we plotted box and scatter plots to have more understanding of the data. We then split the whole dataset into 80:20 split for training and testing respectively.

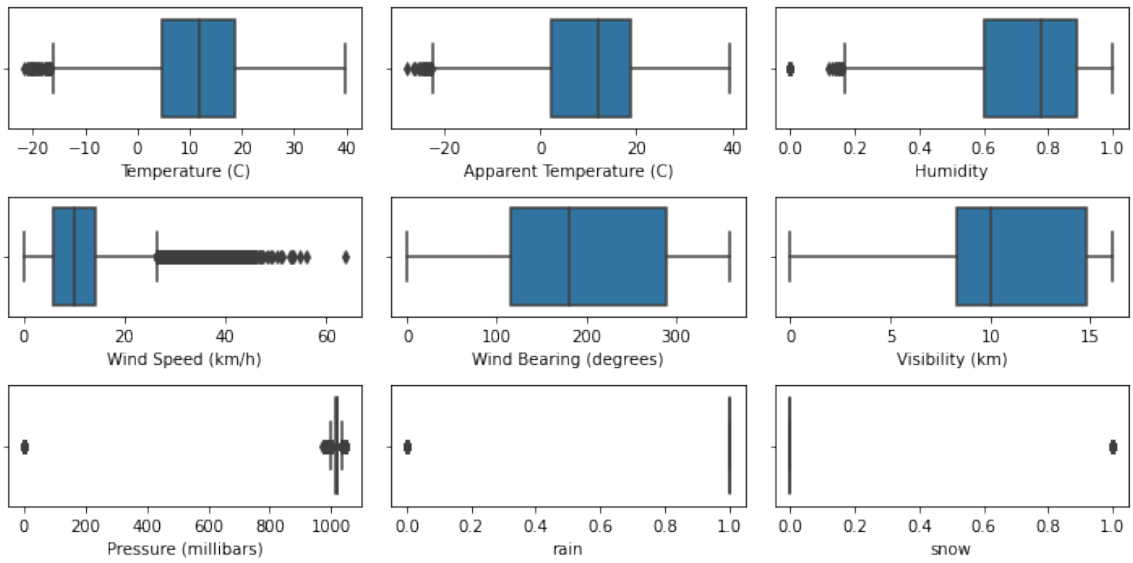


Figure 1: Box Plot

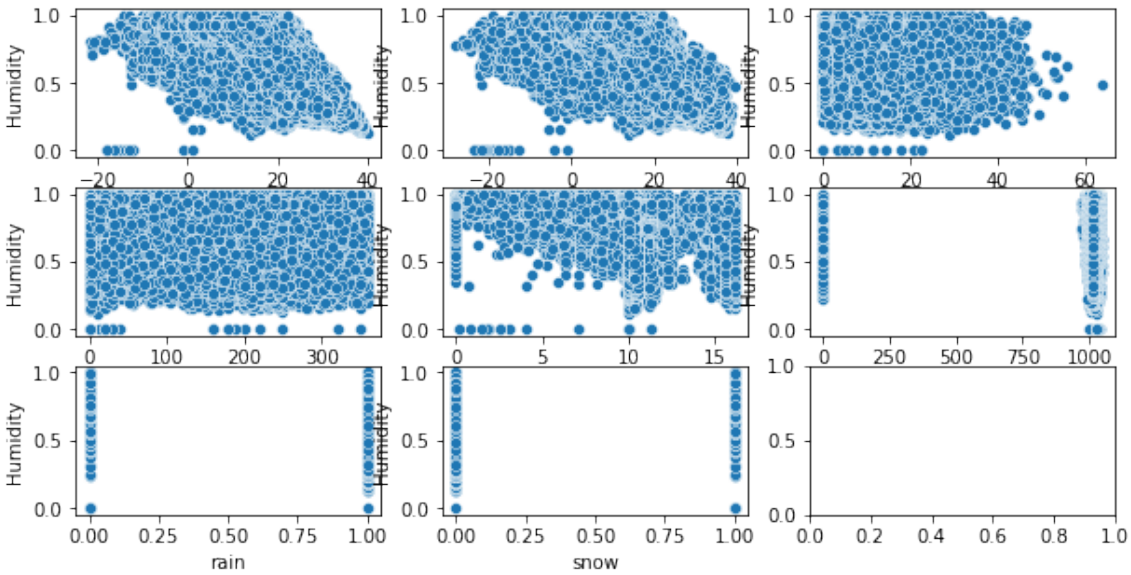


Figure 2: Scatter Plot

2 Experiment with Different Classification Models

To perform our experiment, we selected five different classification modes.

1. Logistic Regression
2. Decision Tree Classifier
3. K Nearest Neighbor
4. Gradient Boosting (we didn't study this one in the class)
5. Random Forest

For every model, we fitted training data and measured their performance against test data. We also plotted validation and learning curves for each model to understand their behavior better.

As every model is biased, we fitted all of the previously mentioned models inside the Voting Classifier to get a more realistic result.

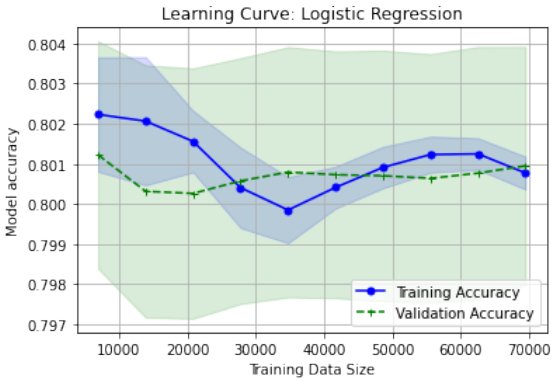
To get the best out of the individual model, we selected the best-performing model from the list of five models and applied Grid Search to tune its hyperparameters.

Finally, we fitted our dataset into AutoML to get the best model for our selected dataset. We used TPOT as the AutoML library.

2.1 Logistic Regression

We observed 80.05

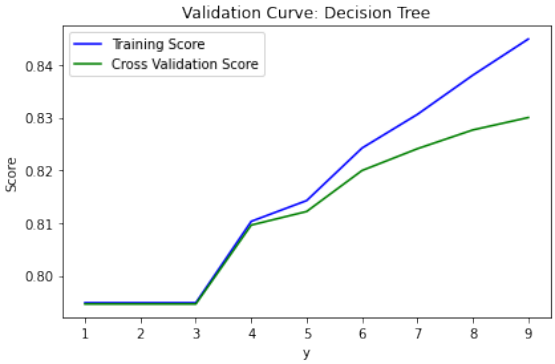
(a) Logistic Regression Validation Curve.



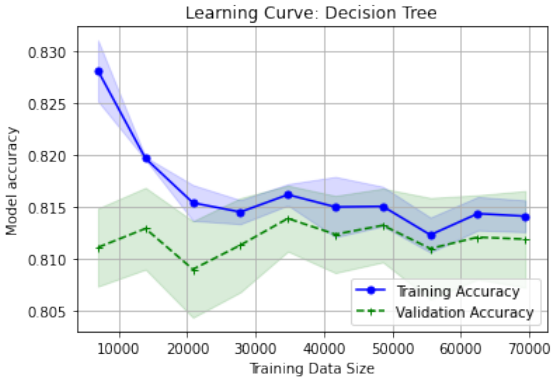
(b) Logistic Regression Learning Curve.

2.2 Decision Tree Classifier

We observed 81.04



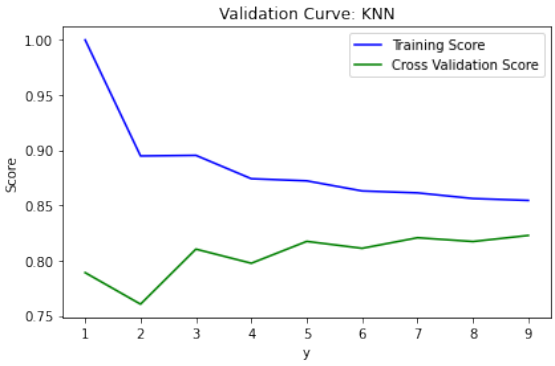
(a) Decision Tree Classifier Validation Curve.



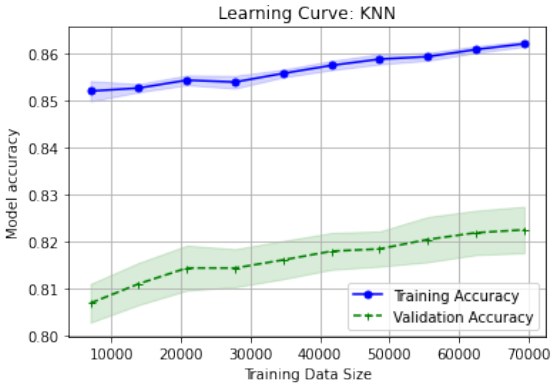
(b) Decision Tree Classifier Learning Curve.

2.3 K Nearest Neighbor

We observed 82.62



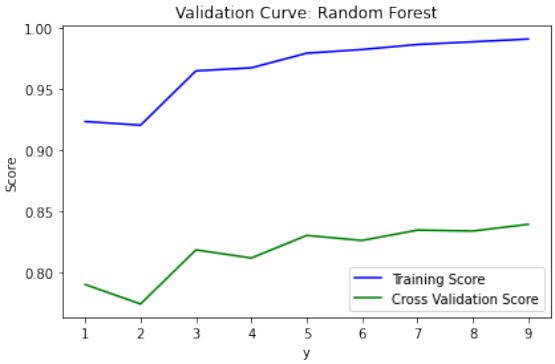
(a) K Nearest Neighbor Validation Curve.



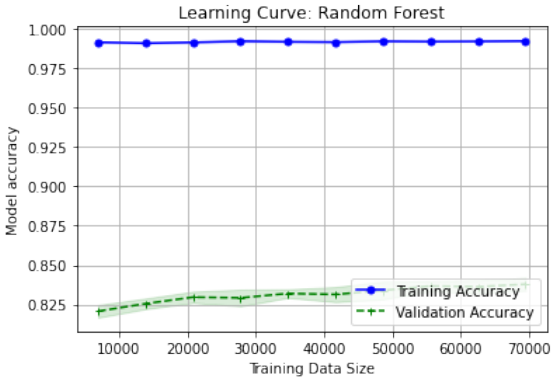
(b) K Nearest Neighbor Learning Curve.

2.4 Random Forest

We observed 83.94



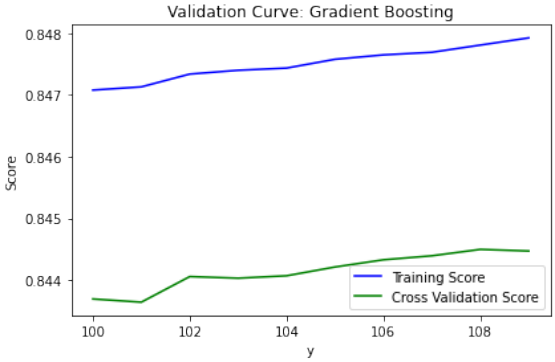
(a) Random Forest Validation Curve.



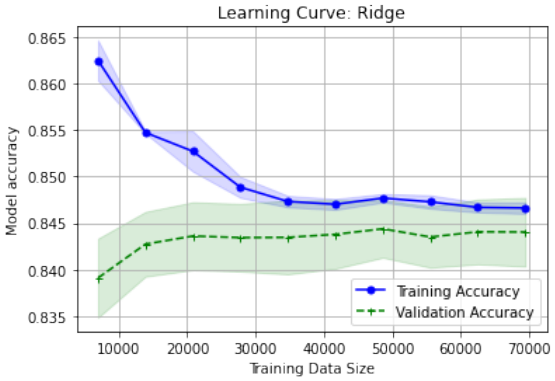
(b) Random Forest Learning Curve.

2.5 Gradient Boosting

Gradient Boosting wast the best performing model of all and we We observed 84.72



(a) Gradient Boosting Validation Curve.



(b) Gradient Boosting Curve.

2.6 Voting Classifier

When we combined all the models and fitted into voting classifier we observed slightly better result than the individual best performing model (Gradient Boosting). We observed 84.97

2.7 AutoML with TPOT

As it takes very long time to figure out the right model automatically using AutoML, we interrupted the TPOT in the middle of its execution. Even we didn't let AutoML to finish still we got Random Forest Classifier as the best performing model. Which is pretty close to our final result. Random Forest Classifier was the second best performing model for our data set.

2.8 Grid Search to Tune Gradient Boosting

We ran grid search with selective hyper parameters of Gradient Boosting to figure out best possible hyper parameters for our problem

```
GradientBoostingClassifier(max_depth=4, min_samples_leaf=25,
    min_samples_split=500, random_state=1)
```

2.9 AUC-ROC Curve

We plotted AUC-ROC curve to visualize our result

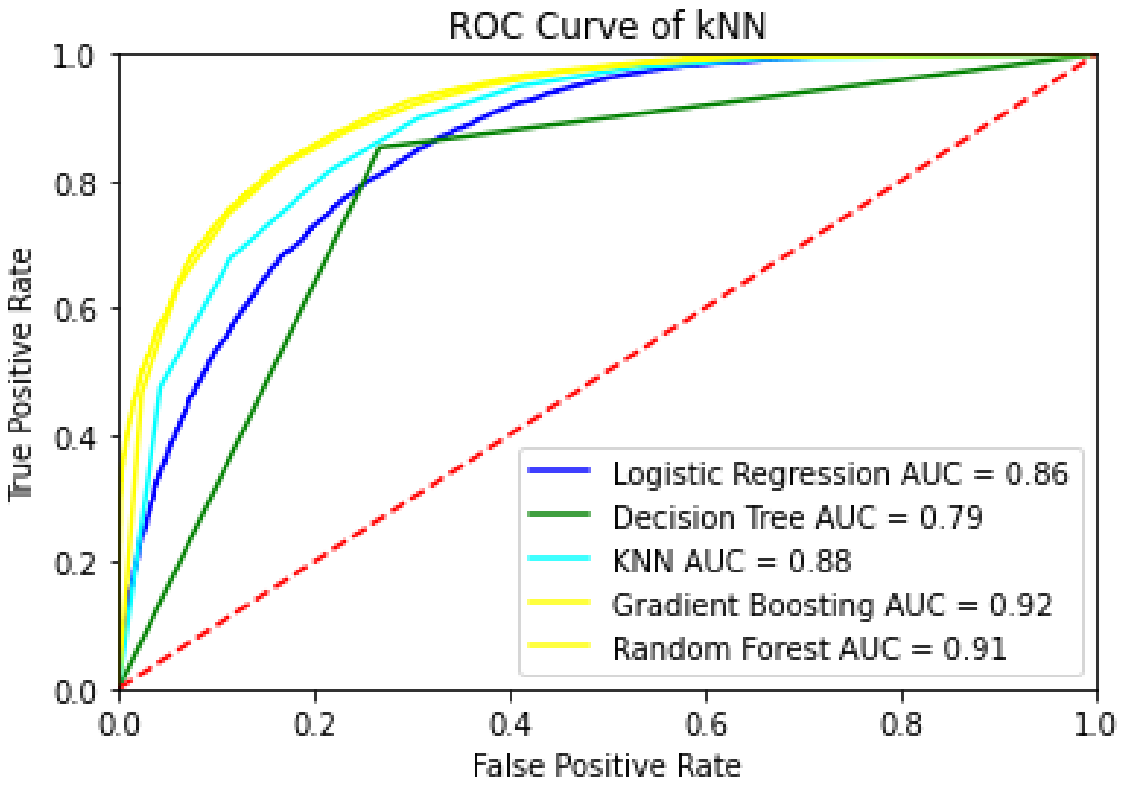


Figure 8: AUC-ROC Curve

2.10 Principal Component Analysis(PCA)

Finally we performed PCA for dimensionality reduction.

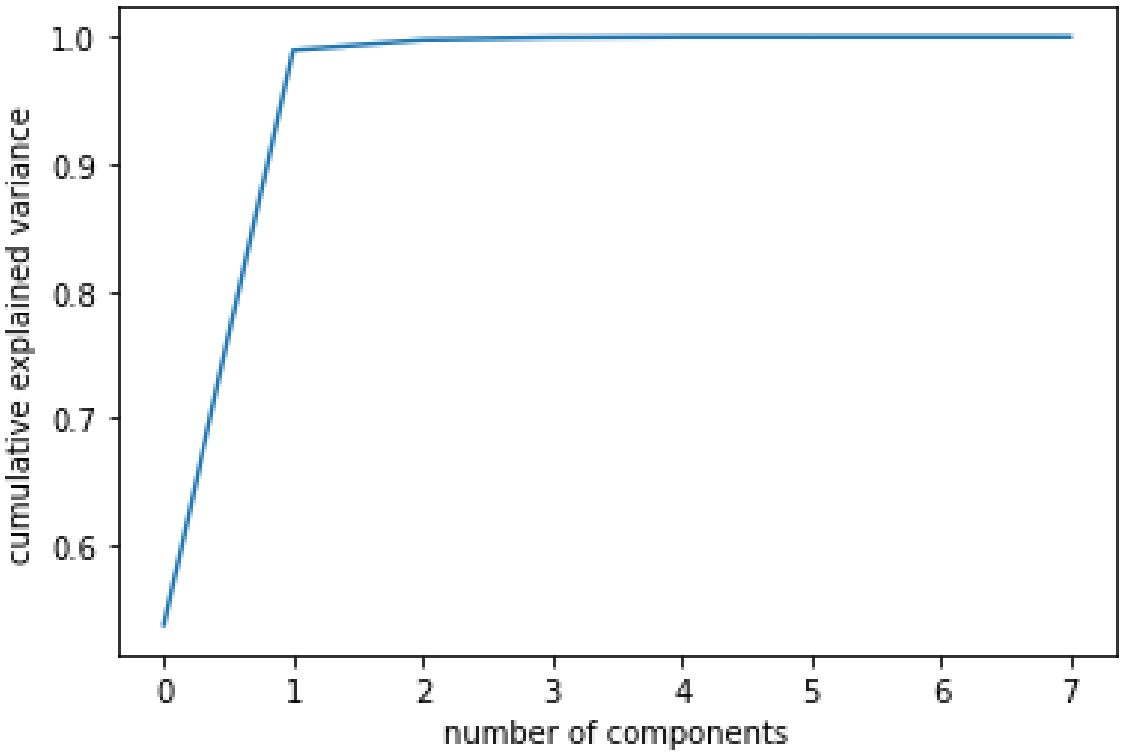


Figure 9: PCA

3 Conclusion

From this experiment we found that Gradient Boosting's performance is very good but it takes long time to train the model. We also found that Voting Classifier works very well to increase overall performance. And Finally, AutoML is very powerful but it requires lot of time and resources to generate best performing estimator.