Assignment 3

Principles of Machine Learning

Name: Rutwik Borole | Id: 22224253

1. In the ML package, select two different regression algorithms that you will apply to the dataset to learn two different regression models. In your report, include a brief, clear description of both algorithms. Ensure that you acknowledge all your sources of information.

We may use the Scikit learn package for our Electricity Consumption Dataset; Sklearn is one of the most helpful and capable libraries for machine learning in Python. It offers a set of efficient tools for machine learning and statistical modelling, such as classification, regression, clustering, and dimensionality reduction, via a Python interface. Scikit Learn includes several features, including the ability to partition data into training and testing sets, multiple built-in datasets, and a variety of regression and classification models, including support vector machines (SVM), decision trees, logistic regression, and linear regression, which the user can customize and use as needed.

For our electricity consumption datasets, I am making use of two regression algorithms from Scikit Learn package, namely Multiple linear regression, and Random Forest Regressor.

- A simple linear regression has a single dependent variable and one independent variable. If "electricity consumption" is explained by the "temperature" of an individual observation, then the regression is expressed in terms of simple linear regression as follows:
  electricity consumption *= b0+ b1** temperature *+ e*

  Multiple regression takes one step further, using two or more independent variables instead of one. If we add another variable (say, "humidity") to the equation above, it becomes a multiple regression:
  electricity consumption *= b0+ b1** temperature *+ b2** humidity *+ e*

  Multiple linear regression predicts the result of a dependent variable using multiple variables. It can account for nonlinear correlations and variable interactions in ways that simple linear regression cannot. And it does so with greater accuracy.
  The formula for Multiple linear regression is as follows:

  $y_i=\beta_0+\beta_1 x_{i1}+\beta_2 x_{i2}+\ldots+\beta_p x_{ip}+\epsilon$

  where,

  for $i=n$ observations:

  y_i=dependent variable

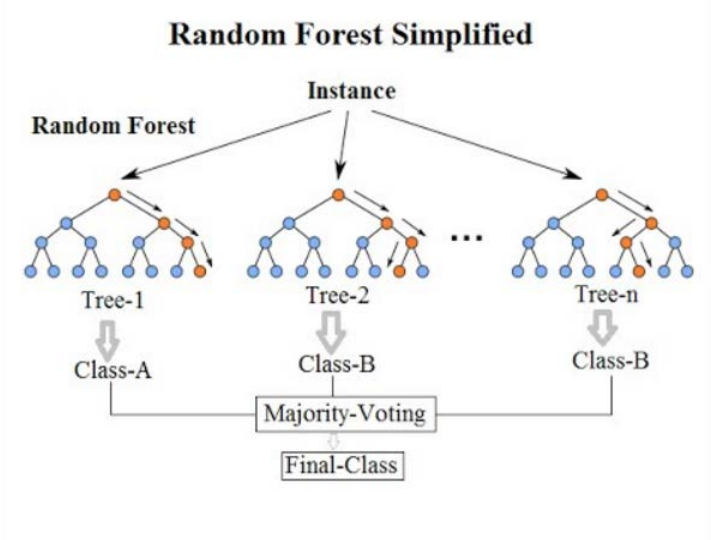
  x_i=explanatory variables

  β_0=y-intercept (constant term)

  β_p=slope coefficients for each explanatory variable

  ϵ=the model's error term (also known as the residuals)

- Random Forest Regression is a supervised learning algorithm that uses ensemble learning method for regression. Ensemble learning method is a technique that combines predictions from multiple machine learning algorithms to make a more accurate prediction than a single model. A Random Forest operates by constructing several decision trees during training time and outputting the mean of the classes as the prediction of all the trees. The averaging makes a Random Forest better than a single Decision Tree hence improves its accuracy and reduces overfitting. A prediction from the Random Forest Regressor is an average of the predictions produced by the trees in the forest.



Source: [wikimedia]

2. Describe the process you followed while developing each model. Be sure to include and justify the final values selected for all parameter settings and describe the process you followed while searching through possible parameter settings. Describe each of the models, using graphics if appropriate.

For both models, I started with preprocessing of the train dataset. Firstly, I split the 'datetime' column into date and time columns using the DATE module. Further, I divided the date into days, months, and years. Next, I used one-hot encoding on the 'var2' column of the dataframe stored the result in the train and test dataframe. Using one hot encoding is necessary because for categorical variables where no ordinal relationship exists, the integer may not be enough, at best or can mislead the model. In this case, a one-hot encoding can be applied to the ordinal representation. This is where the integer encoded variable is removed and one new binary variable is added for each unique integer value in the variable. I dropped columns like 'ID', 'datetime' & 'electricity consumption' from the train dataset as they weren't required for training the models.

- For the linear regression model, I have used the intercept parameter to False as most of the data is centred in all independent variables. As we are using standard scaler from SkLearn package to normalise the numerical values in the datasets, I decided to set the normalize parameter to false. All other parameters of the linear model were not applicable in our predictions, so they were left to their default values, which gave the expected results.
- For random forest regressor, the first parameter I am tuning is the n_estimators as this parameter decides how many trees would be in the forest. I tried with low number of

trees (10,50,100,500) at the beginning, but error value was quite high. Although if n_estimator is set to 1000, it works adequately and doesn't put a lot of stress on the CPU as well. The next parameter I set was random_state = 1, this decides the sampling of the features to consider when looking for best split at each node. Lower number for random state was ideal as higher number were giving ambiguous results. And lastly, I set the min_samples_leaf to 50 as this will allow 50 data points to be placed in a node before the node is split, this parameter will have a smoothing effect and will help in battling overfitting.

3. Discuss how you divided the dataset into training and validation sets and monitored for possible overfitting or underfitting. Can we use cross validation in this scenario, if yes, how did you used it, if not then explain why?

- In our case of electricity consumption dataset, both the datasets, i.e train and test have datetime as one of the columns which makes it a time series forecasting dataset. I have split the data in two ways for making training and validation sets.
  Firstly, I checked if the data is in ascending order by date and then I have split the train dataset by using the TrainTestSplit module of Scikit learn but set the parameter 'Shuffle' to 'False' so the train data will not fit the model on future data and predict on the past data.
  For the second way I have used a simple for loop to split the data according to the days of each month. So, the training data comprises of Day 1 to Day 16 of each month and the validation set comprises of Day 17 to Day 23 of each month. In this way we make sure to not fit the model on future data and predict on past data.
  If we take a look at the windspeed data, we can see that it has very high variance which can possibly lead to overfitting of data, to battle overfitting I have used StandardScalar for my multiple linear regression model and used MinMaxScaler for my random forest regressor.
  Also, to combat overfitting in random forest regressor I have tweaked the value of min_samples_leaf to 50.

- Because our data is projected over time, we cannot use cross validation in this instance (hours in our case). Cross-validation is not straightforward in the case of time series. We cannot select random samples and assign them to either the test or train sets since it makes no sense to forecast values in the past using values from the future. In a nutshell, we want to avoid looking forward when training our model. There is a time dependency between observations, which must be maintained during testing.

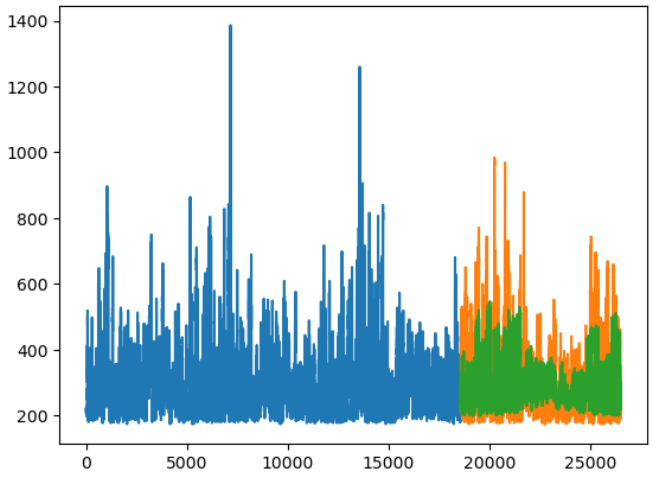
4. Evaluate the performance of the two regression models using appropriate metrics (e.g., RMSE, MAE, R, R2 etc.). Discuss whether the two models give very similar or significantly different results while training and then on testing set, and why

I have evaluated both models twice depending on the above two methods of creating training and validations sets. For evaluation I have used R2 score, mean squared error, root mean squared error and mean absolute error. If we take a look below at the R2 score and error values of both models then we can clearly see that random forest regressor has performed much better than linear regression, R2 score for Random Forest is approximately twice as much higher than the linear regression model. This huge difference is because the linear

regression model fails to adapt to the independent variables. This shows that the independent variables don't have higher correlation with our dependant variable(electricity consumption).

```
#Evaluation for Linear regression based on first 16 days in train set and remaining in validation set

r2 score for Linear Regression model is = 0.22924678886260808
mean_sqrd_error of Linear Regression model is = 10976.28992055669
root_mean_squared error of Linear Regression model is = 104.7677904728199
Mean Absolute error of Linear Regression model is = 74.81480634039269
_____________________________________________________________________________________________
#Evaluation for Linear regression using TrainTestSplit with Shuffle = False

r2 score for Linear Regression model is = 0.19144196167839844
mean_sqrd_error of Linear Regression model is = 9994.821692900772
root_mean_squared error of Linear Regression model is = 99.97410511177767
Mean Absolute error of Linear Regression model is = 69.30328307672265
r2 score for Linear Regression model is = 0.19144196167839844
_____________________________________________________________________________________________
#Evaluation for Random Forest Regressor using TrainTestSplit with Shuffle = False

r2 score of Random Forest Regressor is  0.3233233974327353
mean_sqrd_error of Random Forest Regressor is = 9636.545738518667
root_mean_squared error of Random Forest Regressor is = 98.16590924816347
Mean Absolute error of Random Forest Regressor is = 66.67174528289873
_____________________________________________________________________________________________
#Evaluation for Random Forest Regressor based on first 16 days in train set and remaining in validation set

r2 score of Random Forest Regressor is  0.42159264195858803
mean_sqrd_error of Random Forest Regressor is = 7149.862020401213
root_mean_squared error of Random Forest Regressor is = 84.55685673203098
Absolute error of Random Forest Regressor is = 57.88974485451631
```

Here I have plotted my predictions with the original labels for the best performing model i.e Random Forest Regressor.



References

https://towardsdatascience.com/random-forest-regression-5f605132d19d

https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.RandomForestRegressor.html?highlight=randomforestregressor

https://scikit-learn.org/stable/modules/linear_model.html?highlight=linear+regression

https://www.investopedia.com/terms/m/mlr.asp

https://www.javatpoint.com/multiple-linear-regression-in-machine-learning

https://machinelearningmastery.com/one-hot-encoding-for-categorical-data/