Weekly Progress Report

Submitted by:

Sandesh Pabitwar

Modern Education Society's College of Engineering Pune

Project Title: INTP22-ML-2 Remaining Usable Life Estimation (NASA Turbine dataset)

Objectives:

1. Understand the components and working of Turbo Engine.
2. Understand the dataset and analyze dataset
3. To learn about different Python libraries and their implementation.
4. To develop model for remaining usable life estimation

Task done in this week:

1. Successfully implemented linear regression model.
2. Implemented Random Forest algorithm to get more accuracy.
3. Done hyperparameter tunning on implemented model.
4. Started gathering information about CNN model.

Successfully implemented linear regression model.

Linear regression is a statistical method for modeling relationships between a dependent variable with a given set of independent variables. It is a simplest algorithm in machine learning. Multiple linear regression is one type of linear regression. Multiple linear regression attempts to model the relationship between two or more features and a response by fitting a linear equation to the observed data.

In the given dataset sensors values are acting as a features and RUL is acting as a targeted value (response). Using a linear regression, we can get a linear equation which will be used to predict the RUL.

To check the accuracy of the linear regression I have defined one **evaluate** function which will check the actual and predicted value.

```
def evaluate(y_true, y_hat, label='test'):
    mse = mean_squared_error(y_true, y_hat)
    rmse = np.sqrt(mse)
    variance = r2_score(y_true, y_hat)
    print('{} set RMSE:{}, R2:{}'.format(label, rmse, variance))
```

above image shows the implementation of evaluate function.

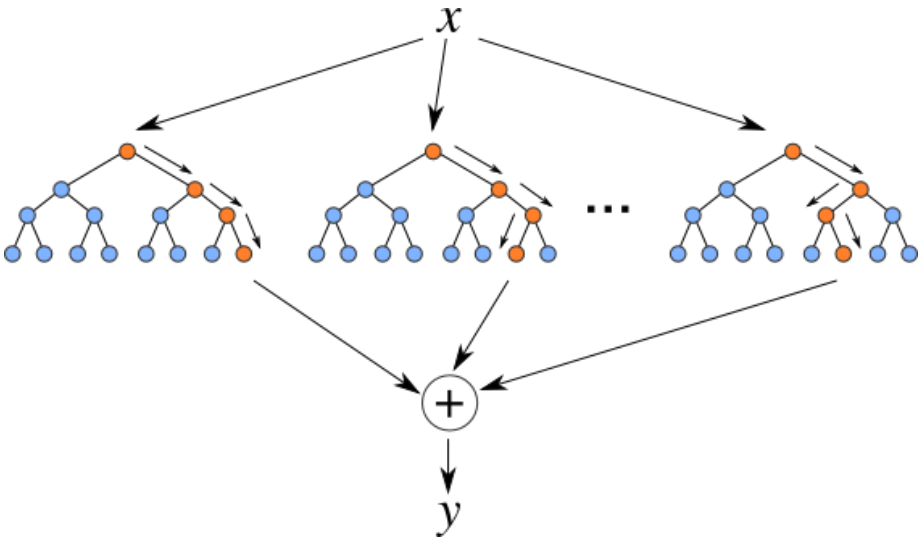
I have implemented the linear regression on train and test data. It was giving 47.42641631507684 % RMSE (root mean square error) on train dataset and 35.43331555318205% RMSE on test data set.

```
train set RMSE:47.42641631507684, R2:0.20094599479348674
test set RMSE:35.43331555318205, R2:0.2729504004868851
```

It means that accuracy is approximately 52%.

Implemented Random Forest algorithm to get more accuracy.

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.



As above image shows to predict the output y model uses multiple decision trees and combines all the decision trees.

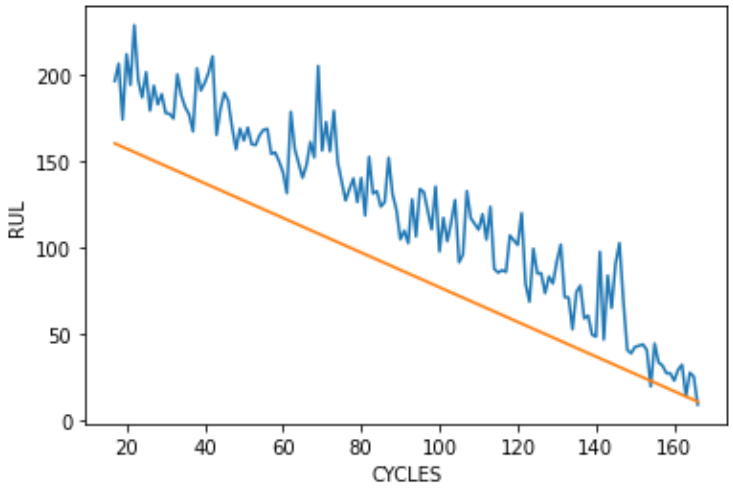
To implement the random forest model algorithm, I have divided the total dataset into two parts
1. Train set

2. Test set

I have divided data in 80:20 part.

After implementing the random forest algorithm, I got `53.11167215342031` RMSE on test set a nd `16.725718955631518` RMSE on train set.

```
RMSE on Test set: 53.11167215342031
RMSE on Training set: 16.725718955631518
```



Above graph shows the predicted values and actual values. Orange colour line is the actual values and blue lines shows the predicted values. It is clear that model is in overfit condition.

Done hyperparameter tunning on implemented model.

I have used Grid search for hyperparameters tunning. Grid-search is used to find the optimal hyperparameters of a model which results in the most 'accurate' predictions.

Used three parameters for hyperparameter tunning i.e., 1. **n_estimators 2. max_depth 3. min_samples_leaf**

n_estimators: The more trees, the less likely the algorithm is to overfit. So, tried to increase this parameter. The lower this number, the closer the model is to a decision tree, with a restricted feature set.

max_depth: This parameter is used to reduce the complexity of the learned models, lowering over fitting risk

min_samples_leaf: This has a similar effect to the max_depth parameter, it means the branch will stop splitting once the leaves have that number of samples each.

Below image shows result after grid search tunning.

```
RMSE on Test set: 84.4757626342506
RMSE on Training set: 16.437382647798177
```

Started gathering information about CNN model.

The construction of a convolutional neural network is a multi-layered feed-forward neural network, made by assembling many unseen layers on top of each other in a particular order.

It is the sequential design that give permission to CNN to learn hierarchical attributes.

In CNN, some of them followed by grouping layers and hidden layers are typically convolutional layers followed by activation layers.

The pre-processing needed in a ConvNet is kindred to that of the related pattern of neurons in the human brain and was motivated by the organization of the Visual Cortex.