To

IITD-AIA Foundation of Smart Manufacturing

Date:09-07-2023

Subject: ***Weekly Progress Report for Week-5.***

Dear Sir,

Following is the required progress report of this week dated from 03-07-2023 to 09-07-2023.

Weekly Progress:

July 03 2023 :

Topics covered:

- I have started training the machine learning model.
- I have modeled and fitted the dataset into a Linear Regression model.
- I have also found the MSE (Mean square Error) and R2 Score of the model.
- Mean Squared Error (MSE): 1.3886596104682267
- Coefficient of Determination (R2 Score): 0.7033756090297395
- I have split the dataset into training and testing sets using the train_test_split() function from scikit-learn. We allocate 80% of the data for training and 20% for testing.
- The predictions are stored in y_pred whose output is [4.00713465 3.62812479 3.6926767 7.40379233 3.94391245 8.06488213 5.21710267 8.43969874 3.59861032 6.54045616 7.74909447 6.68367359].

```
[4.00713465 3.62812479 3.6926767  7.40379233 3.94391245 8.06488213
 5.21710267 8.43969874 3.59861032 6.54045616 7.74909447 6.68367359]
Mean Squared Error (MSE): 1.3886596104682267
Coefficient of Determination (R2 Score): 0.7033756090297395
```

July 04 & July 05:

Topics covered:

- I have performed model fine tuning.
- We use the RandomForestRegressor from scikit-learn as the base model.
- The parameter grid specifies different combinations of hyperparameters we want to explore during the grid search.

- The GridSearchCV class performs the grid search with cross-validation, where we define the number of folds for cross-validation using the cv parameter.
- The scoring metric is set to 'neg_mean_squared_error', but we can change it to a different evaluation metric based on our specific task.
- I have performed merging of the datasets.
- We start by loading the vibration data from the 60 Excel files into separate dataframes using a list comprehension. Each file is read using pd.read_excel and stored in a list called dfs.
- The pd.concat function is then used to concatenate all the dataframes in dfs into a single dataframe called vibration_data. This creates a single dataframe containing all the vibration data from the 60 files.
- Next, we load the cutting parameter data from the summary file using pd.read_excel and store it in a dataframe called cutting_params.
- Finally, we merge the vibration data (vibration_data) with the cutting parameter data (cutting_params) using the pd.merge function. The on parameter specifies the common identifier column between the two dataframes, which is used for the merging operation. Replace 'common_identifier' with the actual column name that serves as the identifier in your dataset.
- The merged data is stored in a new dataframe called merged_data, which will contain the combined information from the vibration data and cutting parameters.

July 06:

Topics covered:

- I have tried using Neural Network for training the model.
- The data is preprocessed by normalizing the input features using minmaxScaler. Then, the data is split into training and testing sets using train_test_split.
- A neural network model is built using the Sequential class from Keras. The model consists of three dense layers with ReLU activation. The model is compiled with the Adam optimizer and mean squared error loss.
- Finally, predictions are made on the test set using model_predict, and the model is evaluated using mean squared error (MSE) and R-squared score.
- The model is trained using model_fit on the training data for a specified number of epochs and batch size.

July 07:

Topics covered:

- I have tried using Gradient Boosting algorithm for training the model.
- There are various implementations of Gradient Boosting, such as XGBoost, LightGBM, and scikit-learn's GradientBoostingRegressor and GradientBoostingClassifier.
- Gradient Boosting uses the gradient descent optimization algorithm to minimize a loss function during each iteration. The loss function is typically a measure of the error between the predicted and actual values.

- The key advantage of Gradient Boosting is that it can handle complex relationships between variables and perform well even with high-dimensional data.

July 08 & July 09:

Topics covered:

- I have performed model selection.
- I am working on selecting the correct and efficient model.
- Today, I tried to compare random forest or gradient boosting with other models.
- These methods are called Ensemble methods.
- Ensemble methods, like gradient boosting, are known for their robustness to noise and ability to handle complex relationships in the data.
- These models can capture nonlinear patterns, interactions, and handle noisy observations effectively.
- They work well with large datasets and are generally more resistant to overfitting.
- Random forest is another ensemble method that combines multiple decision trees to make predictions.
- It is known for its robustness to noise and ability to handle high-dimensional datasets.
- Random forest models can provide feature importance rankings, which can help in understanding the relevance of different parameters.
- Both random forest and gradient boosting are powerful machine learning algorithms that can handle complex relationships and noise in the data.
- They can automatically learn the important patterns and provide accurate predictions without explicitly knowing the underlying relationships between parameters.