Data Driven Kinetics - Manual

Pragneshkumar Rana, Sivaram Ambikasaran, Krithika Narayanaswamy

March 30, 2020

Contents

1 Introduction: **2**
- 1.1 Ignition delay brief intro: . . . 2
- 1.2 Why ignition delay? . . . 2
- 1.3 Key-idea behind framework: . . . 2

2 Setting up for the first time: **7**

3 Run the App: **8**
- 3.1 Available options and flags: . . . 8

List of Figures

1 Division of cluster based on Relative Error . . . 3
2 Division of left and right cluster into sub-clusters . . . 4
3 Conceptual tree diagram of procedure . . . 4
4 Division of data points in different clusters . . . 5
5 Coefficient obtained in different clusters after regression. Highlighted nodes of tree indicates final clusters which are useful of prediction. . . . 6

1 Introduction:

1.1 Ignition delay brief intro:

Chemical reaction and transport process are main components of the combustion process. In combustion, chemical reaction have different time scales which is called as a ignition delay. It is also major combustion property. Ignition delay can be divided into two parts:

1. Physical Ignition Delay

   Physical ignition delay depends on different physical phenomenon of combustion process such as, atomization of fuel, heating, evaporation rate of fuel, etc.

2. Chemical Ignition Delay

   Chemical ignition delay depends on type of fuel, molecular structure of fuel, equivalence ratio etc.

1.2 Why ignition delay?

Ignition delay is crucial parameter for combustor and IC engines. Right amount of ignition delay is essential for efficient working of such devices. Calculation of ignition delay is computationally intensive as it requires time consuming simulations which includes detailed complex chemistry as well. To resolve such issue; accurate, simplified, and efficient framework has been designed to predict the ignition delay. To predict ignition delay, shock tube data of n-alkanes is utilized.

1.3 Key-idea behind framework:

The goal of a framework is to predict ignition delay. For that, machine learning concepts, multiple linear regression is used along with concept of the recursive tree. To divide the data based on error, tertiary tree is utilized. The whole process is summarized briefly below:

1. Fit regression plane on all the fuel data using multiple regression.

2. Calculate relative error of the actual and predicted value for all data.

3. Data points which has relative error less than specified criteria will be combined in one bin called **center cluster**

4. Other data, which has relative error more than specified criteria will be further bifurcated based on absolute error of actual and predicted value.

5. Data points which has positive absolute error has lies on one side of the fitted plane and other data points which has negative sign lies on other side of the fitted plane.

6. Thus, three clusters will be obtained in which,

   Left cluster : Data points which has relative error more than specified error and actual error has positive sign.

   Center Cluster : Data points which has relative error less than specified criteria.

   Right Cluster : Data points which has relative error more than specified error and actual error has negative sign.

Center cluster will give a correlation for the data and will not be divided further. Whereas, Left and Right cluster may also give correlation if they satisfies the specified relative error criteria, otherwise it will be divided recursively by following step:1-6 until specified relative error criteria is not fulfilled.

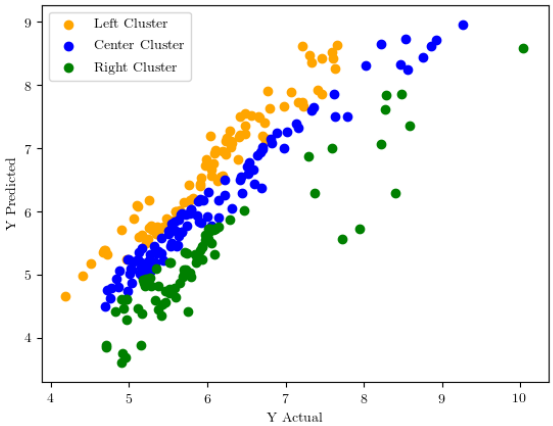
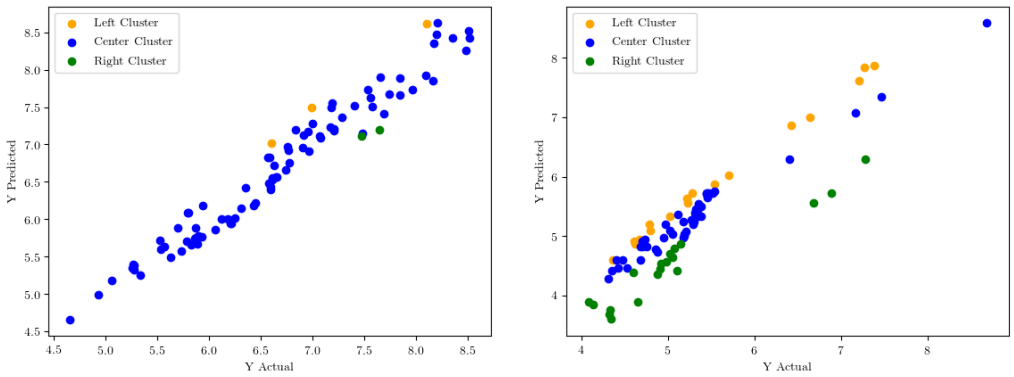


Figure 1: Division of cluster based on Relative Error

For example, after fitting the first regression plane, result of the actual and predicted value is shown in fig 1. Blue colour shows, the data points which has relative error less than 5%. Green and orange colored data points have relative error more than 5%. Apart from relative error, absolute error also plays essential role in clustering. Orange data point have positive absolute error and green data points have negative absolute error. By this way, three clusters is obtained. Further, orange and green data points will be divided into three clusters individually following the same procedure explained above. Obtained result is shown in fig 2a and 2b.



(a) Left Cluster Division (b) Right Cluster Division

Figure 2: Division of left and right cluster into sub-clusters

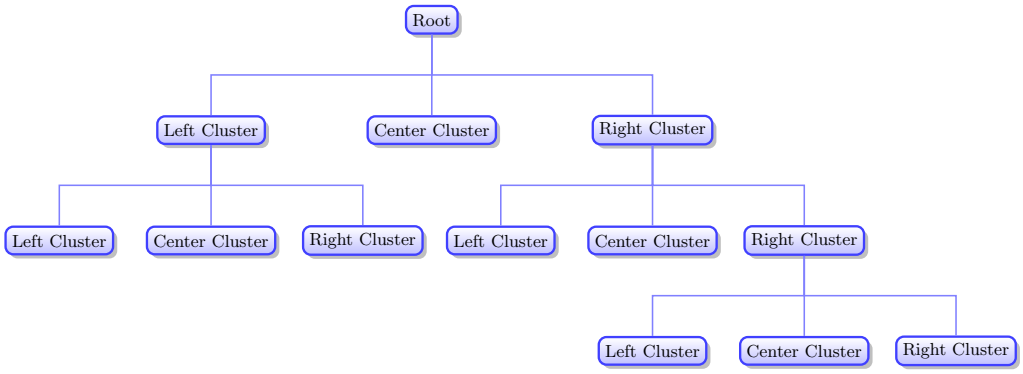


Figure 3: Conceptual tree diagram of procedure

Each center clusters gives one correlation for prediction of ignition delay. Left and right clusters can also give correlation or can generate cluster if it satisfies the specified relative error criteria else, it will further divide into more parts/clusters till relative error criteria is not satisfied for data-points which is given in fig-3. For example distribution of data points in different cluster is shown in fig 4.

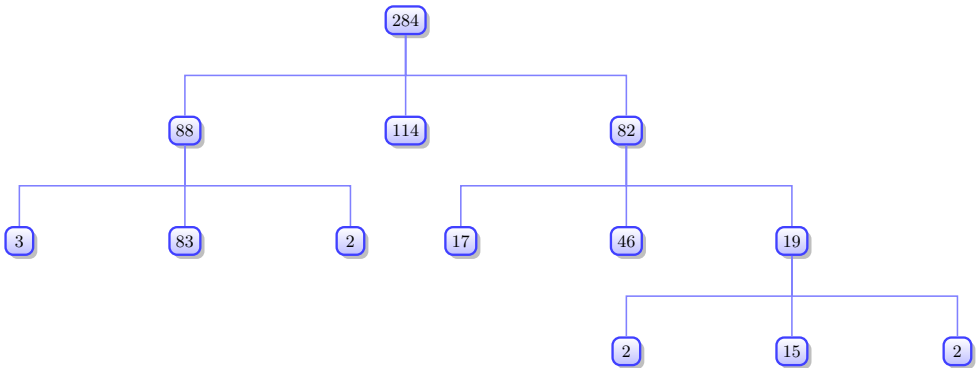


Figure 4: Division of data points in different clusters

Along with that, the regression coefficient obtained after following the whole procedure is given in fig .5. From figure 5, it is clear that the nodes highlighted with red borders shows final clusters. The correlation obtained by highlighted clusters as in fig-5 are useful for prediction of ignition delay value. Calculation of centroid and distance of data points from those centroid plays major role in prediction.

The formulation of multiple linear regression used to find out the correlation is given as,

$$\tau = \beta_0 \frac{P}{P_0}^{\beta_1} X_{fuel}^{\beta_2} X_{oxi}^{\beta_3} exp\left(\beta_4 \frac{T_0}{T} + \beta_5 \frac{T_0}{T \times C_{SH}}\right) \tag{1}$$

$$= \beta_0 \frac{P}{P_0}^{\beta_1} X_{fuel}^{\beta_2} X_{oxi}^{\beta_3} exp\left(\frac{1}{T}\left\{\underbrace{\beta_4 T_0 + \beta_5 \frac{T_0}{C_{SH}}}_{\text{similar to } \frac{E_a}{R}}\right\}\right) \tag{2}$$

Here, all the β_i values indicates the regression coefficients. P-pressure, X_{fuel}-fuel mole fraction,X_{oxi}-oxidizer mole fraction, T-temperature, C_{SH}-number of secondary carbon and hydrogen bonds are the features of the equation. P_0 and T_0 are reference pressure and temperature.

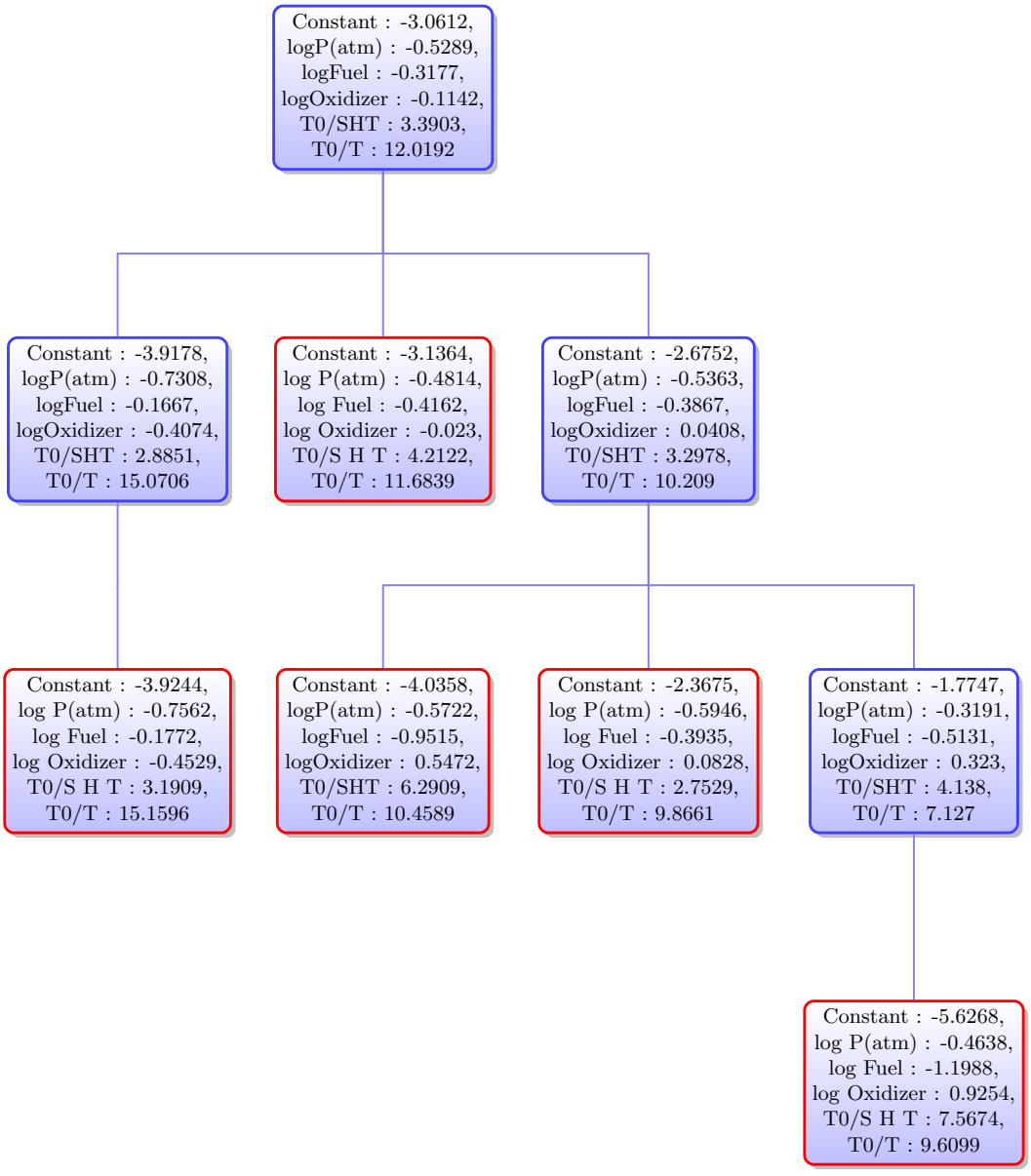


Figure 5: Coefficient obtained in different clusters after regression. Highlighted nodes of tree indicates final clusters which are useful of prediction.

2 Setting up for the first time:

- After downloading the application, put it in the *./home* directory.
- Install all the dependency. Dependency can be directly installed using INSTALL.sh file. They are mentioned below:
  - numpy
  - matplotlib
  - seaborn
  - scikit
  - statmodel
  - rdkit
  - latexpdf - Install manually by following instruction

    To install all dependency run the following command:
  - chmod +x INSTALL.sh
  - ./INSTALL.sh
- After installing all the dependency, open .bashrc file and copy-paste following command at end of the file.
  - ##For DATA DRIVEN FRAMEWORK
  - export IDCODE="~/Data_driven_Kinetic/"
  - export PATH=$PATH:$IDCODE
  - alias IDprediction="pwd ~/Data_driven_Kinetics/filelocation.txt && Run.sh"
- All set! - Run the application following instruction given in section-3

3 Run the App:

After following the procedure given in section-2, it is convenient to use the app from any directory. let's consider filename- **alkane.csv** for explanation of all commands.

- Command to run the code:

  IDprediction -flag File-Name

3.1 Available options and flags:

- **-a : *Analyze* the data-set by selecting range of parameters**

  - command : **IDprediction -a alkane.csv**

  - To analyze and find out data-points having properties in certain range to analyze ignition delay for specific fuel.
    - select fuel by giving corresponding index as input
    - Input equivalence-ratio from available options
    - Input pressure value from available options
    - By default it will considers all the data points having same fuel structure and same equivalence ration within ± 0.5 atm. If you want to change the range pressure input:'y' and input the desired range else input:'n'

  - **Output :**
    - ./result/data_analysis/

      It will generate the plot of τ vs 1000/T, which includes data points having selected ϕ, specified range of pressure for selected fuel.

- **-b : *Bond* types in given fuel**

  - command : **IDprediction -b CCCC**

    To find out type of chemical bonds available in different type of carbon and other atoms.
    Note: Works for n-alkanes and branched alkanes.

  - **Output :**

    * ./result/Bond_details/

      It will generate **SMILE_result.csv** file which contains available bond details for given fuel. Detail in the file includes,

      · **Primary_C** : Total number of primary carbons
      · **Secondary_C** : Total number of secondary carbons
      · **Tertiary_C** : Total number of tertiary carbons
      · **Quaternary_C** : Total number of Quaternary carbons
      · **Other_Atom** : Total number of atoms other than carbon
      · **P_P** : Total number Primary-Primary carbon bonds
      · **P_S** : Total number Primary-Secondary carbon bonds
      · **P_T** : Total number Primary-Tertiary carbon bonds
      · **P_Q** : Total number Primary-Quaternary carbon bonds
      · **S_S** : Total number Secondary-Secondary carbon bonds
      · **S_T** : Total number Secondary-Tertiary carbon bonds
      · **S_Q** : Total number Secondary-Quaternary carbon bonds
      · **T_T** : Total number Tertiary-Tertiary carbon bonds
      · **T_Q** : Total number Tertiary-Quaternary carbon bonds
      · **Q_Q** : Total number Quaternary-Quaternary carbon bonds
      · **P_H** : Total number Primary carbon - Hydrogen bonds
      · **S_H** : Total number Secondary carbon - Hydrogen bonds
      · **T_H** : Total number Tertiary carbon - Hydrogen bonds
      · **Fuel** : Fuel SMILE

      Note : If file already exits then result will be appended to old output file.

- **-h : *Histogram* plots of parameters for each fuel individually**

  - command : **IDprediction -h alkane.csv**

    It will generate the histogram plots of parameters associated with ignition delay, separately for all the fuels.

  - **Output :**

    * ./result/Fuel_Parameter_Histogram/

      Directory contains, folders named by fuel smiles. Each folder contains histogram plot of parameters associated with ignition delay. These plots are useful for visualization and analysis of the parameters.

- **-c : *Criteria* for separation**

  - command : **IDprediction -c 0.05 -m alkane.csv**

    Flag has to be passed with multiple linear regression or error based clustering regression. Criteria value 0.05 indicates relative error of 5%. Criteria plays key role in creation of clusters. Use of criteria is to filter out the data points which has relative error less than specified value to create a cluster.

  - **Default : 0.05**

- **-r : *Remove* feature by back-elimination**

  - command : **IDprediction -c 0.05 -r True -s 0.05 -m alkane.csv**

    This Flag **has to be passed with True or False** .
    True : Backward elimination will be activated
    Flase : Backward elimination will be deactivated

  - **Default : False**

- **-s : *Significance Level* for p-value or Confidence zone criteria**

  - command : **IDprediction -c 0.05 -r True -s 0.05 -m alkane.csv**

    Value passed with this flag is useful for backward elimination. Significance level is used for rejection/acceptance of null hypothesis. Low significance value is generally passed as it indicates higher evidence is needed for rejection of null hypothesis.

  - **Default : 0.05**

- **Null hypothesis :** For multiple linear regression, null hypothesis defined as, there is no statistical relationship between independent and dependent variables or coefficient associated with dependent variable is zero.

- **Backward elimination :** In this procedure, if any feature has p-value more than specified or default value 0.05 (if -r True and -s not passed) then that feature will be eliminated. Result will be obtained running regression again. Same procedure will be repeated till p-value associated with all the regressor are less then specified value.

- **-m : *Multiple* linear regression of data**

  - command : **IDprediction -c 0.05 -r False -m alkane.csv**
    or **IDprediction -m alkane.csv**

    It will do multiple regression on whole data-set and will **generate output as root of the tree.** Details of other flag is mentioned above. If others flags are not passed then it will take default value as mentioned.

  - **Output :**

    * *./object_file/* - useful for prediction

      This directory has three more sub-directory inside it; **./regressor -** object file of trained model. **./x_names** - has feature names of trained model. **./scalar**- object file which has function to scale data (NOT UTILIZED).

* *./result/coefficients/Node_type/*
  stores coefficient obtained after multiple regression

* – *./result/console_output/Node_type/*
  Output printed on console screen also get stored in the **output_result.txt** file

* *./result/Node_type/ID_comparison_i/*
  Directory contains **./ID_comparison_i** folder, which contains file named, **ID_comparison_test_i.csv** which has Ignition delay comparison detail for testing and likewise similar detail of training data is in **ID_comparison_train_i.csv** file. Ignition Delay comparison file contains y_act(Actual Ignition Delay value), y_predicted(Predicted Ignition Delay value), and relative error between those two values.

* *./result/vif/Node_type/*
  VIF(Variation Inflation Factor) - it is useful to check the multi-collinearity of the features
  VIF = 1 means feature has no multi-collinearity
  VIF = 1-5 means feature has moderate correlation
  VIF > 5 means poorly identify the coefficient
  ./vif_i.csv contains features and associated VIF values.

  source:
  Multicollinearity
  VIF

Note: i here indicated index
Node Type includes : {root,left_node, center_node, right_node}

- **-t : *Tree* - Error based clustering regression**
  - command : **IDprediction -c 0.05 -r False -t alkane.csv**
    or **IDprediction -c 0.05 -t alkane.csv**

    The whole idea behind error besed clustering is explained in section-1.3.
  - **Output :**
    - *./object_file/* -useful for prediction
      This directory has Four more sub-directory inside it;
      - **./regressor -** object file of trained model.
      - **./x_names** - feature names of trained model.
      - **./scalar** - object file which is used to scale the data (NOT UTILIZED)
      - **./centroid**- it contains files for final clusters (highlighted nodes as in fig-5). Each file contains average calculated value of features associated with cluster-data.

    - *./plots/*

      Directory contains several tree based plots which helpful for visualization. Along with that it has sub directory named **./cluster_plot_y** which includes comparative plots of y_{actual} vs $y_{predicted}$ for every clusters as shown in fig-1;
      ./plots/ directory has six plots. All plots are generated with the help of latex.
      - **ChildLabel** plot is shows index associated with clusters.
      - **Datasize** plot shows number of data points in each cluster.
      - **Training** plot shows R2 value of training set in each cluster.
      - **Testing** plot shows R2 value of testing set in each cluster.
      - **MaxRelError** plot shows maximum relative error of training set for each cluster.
      - **coefficient** plot shows coefficient obtained after regression for each cluster node.

      ./plots/cluster_plot_y/- it contains plots as in fig-1. Plots are useful to understand role of relative error and absolute error in clustering. Data points with same colour are in one cluster.

* *./result/coefficients/Node_type/*
  stores coefficient obtained after multiple regression

* *./result/cluster_data_before_regression/Node_type*
  Directory contains cluster specific data-file of **useful clusters -** means data which is utilized for generation of cluster.

* *./result/Tree_coefficient_result/Node_type/*
  stores coefficient obtained after multiple regression which is same as copy of ./result/coeffcients/. Stored seperatly as it will not mix its result with Multiple regression result. **It has data of only final clusters.**

* *./result/final_cluster/Node_type*
  contains data file of seperated clusters for all the nodes

* *./result/console_output/Node_type/*
  Output printed on console screen also get stored on the in the output_result.txt file

* *./result/Node_type/ID_comparison/*
  Directory contains ./ID_comparison_i folder, which has Ignition delay comparision files training and testing. Ignition delay comparison file contains y_act(Actual Ignition delay value), y_predicted(Predicted Ignition Delay value), and relative error .

* *./result/vif/Node_type/*
  VIF(Variation Inflation Factor) - To check the multi-collinearity of the features
  VIF = 1 means feature has no multi-collinearity
  VIF = 1-5 means feature has moderate correlation
  VIF > 5 means poorly identify the coefficient
  ./vif_i.csv contains features and associated VIF values.
  source: Multicollinearity, VIF

Note: i here indicated index
Node Type includes : {root,left_node, center_node, right_node}

- **-e : *External* Dataset used for prediction of ignition delay**

  - command : **IDprediction -e testset.csv**

    testset.csv file may contain all the features except ignition delay values. The goal of the process is to predict ignition delay. Procedure behind code is briefly mentioned below:,

    1. Load all the centroid files
    2. Calculate the distance of all the data points from all the centroids and assign the centroid to each data point by least distance.
    3. By assigned cluster, load the respective regressor object of that cluster and predict the ignition delay and process the result.

  - **Output :**

    * *–./external_test_result/console_output/*
      Directory contains **output_result.txt** which contains output printed on console screen.

    * *–./external_test_result/Ignition_delay_comparison/*
      ID_comparison_external_cluster_{Node_index}_{Node_type}.csv file which includes y_predicted - Predicted Ignition delay value, y_actual - actual ignition delay vales and relative error between them. All files are cluster specific means data point in the assigned to one specific cluster.

    * *–./external_test_result/classified_data*
      Directory contains several data files. Each file is associated with one cluster and it contains independent variable values,class of data points, predicted and actual ignition delay values.

    * *–./external_test_result/prediction_comparison_plots/*
      It contains Predicted Ignition Delay vs Actual Ignition Delay plot. Plots are generated using on data points assigned to specific cluster. In short, each plot is related to each cluster.

- **-p : *Plot* frequency of the parameters**

  - command : **IDprediction -p coefficient.csv**

    While running the regression, data points will be splitted randomly. Due to split and change in the data points, obtained coefficient will get affected. To visualize variations in coefficient and find out average coefficient value this command is utilized. Gives result in the form of plot.

    Use this command in the directory where coefficient file exist.

  - **Output:**

    * ./coefficient_histogram_plots
      it contains all histogram plots
    * ./result/
      it contains file output_result.txt which stores output printed on console screen.