GENESYS : ASHFAQUE AZAD (CODING ROUND)

Requirement:

Build a new product for the housing market. The service will be used by property websites to suggest a price to a customer adding a new listing based on the details of the property.

The end product is shown in figure 1.

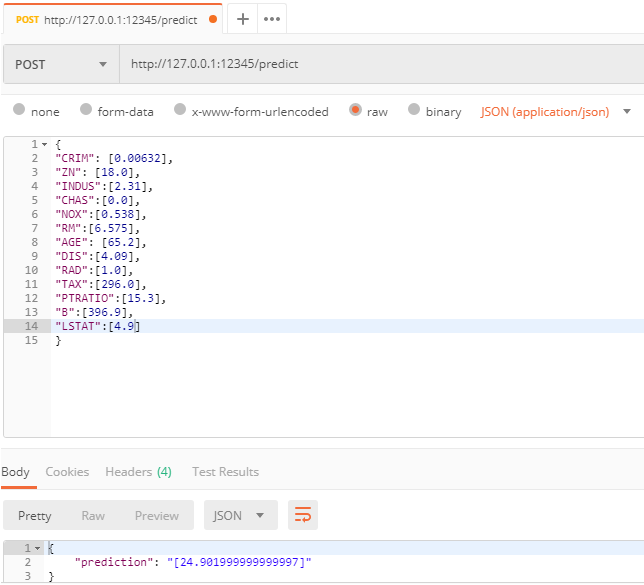


Fig. 1.

Figure 1. demonstrates the final expected output. [ Tool: <https://www.getpostman.com/> ]

Algorithms :

* Linear Regression
* Random Forest (Regression)

DATA (INPUT FOR THE MODEL):

Boston house prices dataset.

No. of instances : 506

No. of attributes: 13 (mixture of numeric and categorical variables)

Tools: Python, sklearn (library for ML), Flask (web framework), and POSTMAN.

There were no null values.

Scaling:

Data was standardized or scaled by removing the mean and scaling it to unit variance.

Linear Regression:

**FILE NAME : LinearRegression-AAzad.ipynb**

Reason for choosing: simplicity of the algorithm.

Training data size: 80% of the data. Rest test data.

Performance Index: R^2 coefficient. : 0.75

*Note:*

*Model was separately trained with few features excluded (this was done after finding its relative importance from the Random Forest algorithm) . However, there was no significant difference produced by the Linear Regression algorithm. R^2 coefficient remained 0.75 up to two decimal places.*

*Random Forest Performed better than Linear Regression.*

Random Forest (with all default setting except n\_estimators):

**FILE NAME : RandomForest-AAzad.ipynb**

n\_estimators was varied and the one with the best R^2results was selected.

Coefficient of determination R^2 came to be: 0.89

About 18.6% improvement in performance compared to Linear Regression.

Next through feature importance, the models with the best scores (above a manual threshold set) was selected and a new RF model was trained.

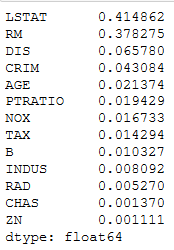


Fig. 2.

Figure 2 shows the features with their importance scores. We see that ‘LSTAT’ and ‘RM’ are the most important features.

The new model trained with fewer features however did not give any significant difference in results.R^2 scored before and after feature selection differed only by 0.7%.

Since the data (instances) available is only about 500 rows and the difference in the performance score was low, it isn’t prudent to go with few selected features. Hence, the model was trained with all of the features.

Random Forest (Hyperparameter optimization ):

**FILE NAME : Random Forest (Hyperparameter Optimisation).ipynb**

Hyperparameters such as ‘max\_depth’ , ‘max\_features’ were set using GridSearchCV , which uses exhaustive search over specified parameter values for an estimate.

n\_estimators was found once against out-of-bag score and once against R^2 score.

However, the model, with the default settings, performed better (based on R^2 score) by 3.4% with respect to one with hyperparameter optimization.

The model used for the API is with the default settings (the only hyperparameter changed was n\_estimators).

**FILE NAME FOR THE *API* USED : RF\_API.py**

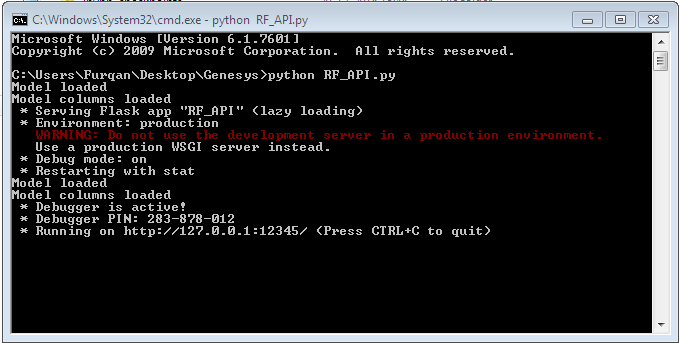
****

Fig 3. Shows the successful compilation.

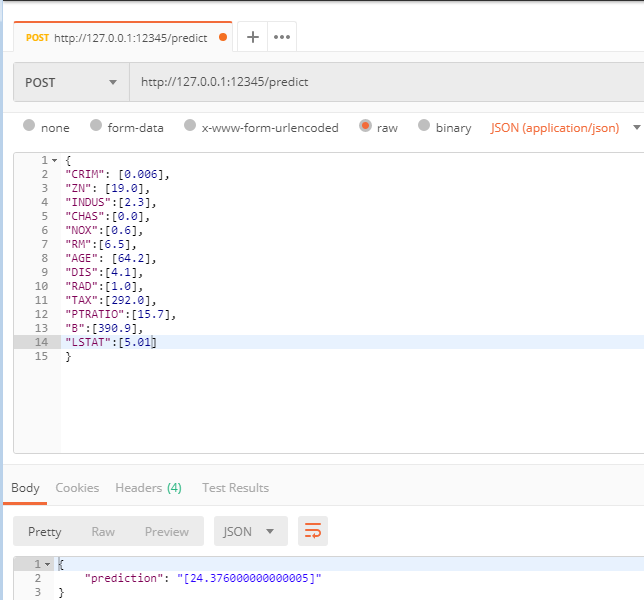
**Saved Models:**

****

**Saved DataFrames:**

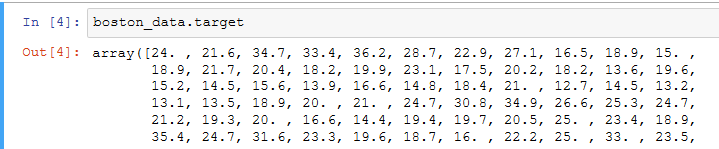
** (training)**

**(testing)**

****

**Note:**

The prediction is in the original format as the following:



The data has not been multiplied by 10000.