← Go Back to Elective Project

Course Content

Project FAQ - Boston House Price Prediction

How to implement OLS() for Boston house price prediction?

You can use the OLS() method and pass the necessary parameters to create the model and use .fit() on top of the model to fit the data. For example,

```
model1 = sm.OLS().fit()
```

In the OLS() function, pass the parameters y_train as the first parameter and X_train as the second parameter.

After the model is created and fitted, one can access the summary() method. In the presence of heteroskedasticity, the main consequence for the least-squares estimator is that the least-squares estimator is still a linear and unbiased estimator, but it might no longer be the best, i.e., there might be another estimator with a smaller variance.

What is Goldfeld Quandt test?

The Goldfeld Quandt test is a test used in regression analysis to test for homoscedasticity. It compares variances of two subgroups; one set of high values and one set of low values. If the variances differ, the test rejects the null hypothesis that the variances of the errors are not constant.

Is there any reason why we choose a random state in the formulation?

The random_state is a non-negative integer that ensures that the results that you generate are reproducible. The random state that you provide is used as a seed to the random number generator. This ensures that the random numbers are generated in the same order every time you run the code. For example, scikit-learn uses random permutations to generate the train and test splits.

Why did you create a log transformation for MEDV and not other ones like NOX, for example?

It is important to note here that the predicted values are log(MEDV) and, therefore, coefficients have to be converted accordingly by taking their exponent to understand their influence on price. However, if we take the transformation of an independent variable, then it would be harder to interpret the model coefficients for that variable.

For example, the house price decreases with an increase in NOX (nitric oxide concentration). When everything else is constant, a 1 unit increase in the NOX leads to a decrease of in a home value of $2,800 (MEDV is measured in 1000 dollars). This is fairly easy to understand as more polluted areas are not desirable to live in and, hence, cost less.

What are the similarities and differences between the training set and the test set? Is there a decision process for inputting training and test data sets?

Train set:

Test set: A subset to test the trained model.

Training the model means letting the model learn from the training data and testing the model means evaluating the performance of the model on unseen data.

In a dataset, a training set is used to build a model, while a test set is used to validate the model built. Data points in the training set are different from the test set. There is one more type of set called **validation set** that we use to tune the model.

1) **Training Set:** A subset to train a model. Here, you have the complete training dataset where you can perform EDA, data preprocessing, feature engineering, etc. and use it to fit the model.

2) **Validation Set:** This is crucial to choose the optimal values of hyperparameters for the model. We can divide the training set into a train set and validation set. Based on the results on the validation set, the model can be tuned, which helps to get the most optimized model.

3) **Testing Set:** Once the model is trained and tuned, you can test the performance of the model on the data which is not observed before by the model.

. **When do we use randomization?**

You can use randomization before splitting train and test datasets, except if you have time series data or if the order is important. In that case, there are ways to do it. If you have an imbalanced dataset, then you can do oversampling, undersampling, or synthetic data generation techniques like SMOTE before randomly splitting it.

Both training and testing datasets must reflect the original data distribution. The original dataset must be randomly shuffled before the split phase in order to avoid a correlation between subsequent elements.

. **The relationship between RAD and TAX saw no pattern. So to remove outliers, we calculated a new df1 where df[TAX]<600, i.e., we created a df1 where values less than 600 for TAX were stored. In that block, in the last print statement, we have added [0] at the end. What is its purpose?**

It will return two values: Pearson's correlation coefficient at the 0th index and the Two-tailed p-value at the 1st index. To show the Pearson correlation coefficient, we used the 0th index.

The Pearson correlation is a metric used to find the correlation between any two variables. In the code, we have seen that there is a high correlation between the TAX and RAD variables. This high correlation might be due to outliers in the data. So after removing outliers, we need to check whether the correlation has decreased or not. In order to check this, the Pearson correlation coefficient is used to show the correlation between TAX and RAD variables.

Note: There is no need to remember the functions, it is just to demonstrate that correlation was decreased after removing the outliers from the data. Whenever you want to know whether any 2 variables are highly correlated or not, Pearson correlation coefficient would help.

. **In linear regression, we have four assumptions:**
1) The mean of residuals should be 0
2) No Heteroscedasticity
3) Linearity of variables
4) Normality of error terms
How many of them need to be true in order for us to know that we can go ahead with the model?

The linear regression does not depend upon the number of assumptions satisfied, it's about how well the model would perform in different scenarios.

The model will be less accurate if the model fails to meet the assumptions. The amount of inefficiency of the model will depend upon the extent it deviates from the assumption. However, we can try some transformations to satisfy the above conditions. For example, if the model fails to meet heteroscedasticity, we can solve it by transforming the target variable (i.e., y) using square root, log, reciprocal square root, or reciprocal transformations.

- **While trying to run the code to create the linear regression model, it gives an error:**
  ValueError: shapes (354,12) and (354,12) not aligned: 12 (dim 1) != 354 (dim 0)

You need to pass y_train instead of Y in question 5. You are getting an error while using y_train because you have dropped the 'TAX' column from the X dataframe instead of X_train in question 4. Please replace X with X_train in question 4 to resolve the error, and then you can use y_train in question 5 to get the desired results.

Please refer to the below code for your reference to create the linear regression model:

```
# Create the model
model1 = sm.OLS(y_train, X_train).fit()
```

- **The medv log was right skewed and applying log transformation will change it almost to normal distribution. Is applying log will change the values of MEDV which may result in an incorrect model?**

While we apply the logarithmic transformation, the higher values certainly get reduced, and hence the prediction is also used to be log-transformed. This certainly reduces the error because log-transformed values are less than non-transformed values. So, it is important to note here that the predicted values are log (MEDV) and, therefore, coefficients have to be converted accordingly by taking their exponent to understand their influence on price.

For example, the house price decreases with an increase in NOX (nitric oxide concentration). When everything else is constant, a 1 unit increase in the NOX leads to a decrease of exp(1.056225) ~ 2.88 times the price of the house. This is fairly easy to understand as more polluted areas are not desirable to live in and, hence, cost less.

- **What should be included in the conclusions and recommendations at the end of the project?**

The conclusions and recommendations should be totally based on the key findings from your entire project. It can be based on the most important variables as per the model outputs/coefficients, and any important observations you have made during the exploratory data analysis.

- **How do we build up our Python coding skills?**

Constant practicing is the most important factor in improving the coding skills. Go through the case studies and practice project provided in this course, and try to code them yourself. You develop coding skills with time, keep doing more hands-on.

Happy Learning!

Previous Next