**TASK 2**

1. **INTRODUCTION**

This report is centered on building a regression model which predicts the prices of house in the given data set. Here, Decision tree regressor was used to determine the important features as well as predict the house prices. The result of the prediction was then compared with the true value and a plot of this was generated at the end. It is also important to note that Python tool was used to solve these problems.

**2.0 Setup**

To start with, libraries such as pandas (used to work with data set), seaborn (for counting and plotting), matplotlib (for plotting) and numpy (for working with arrays) were used in this process. And for the feature importance and prediction, the respective library was imported from sklearn.

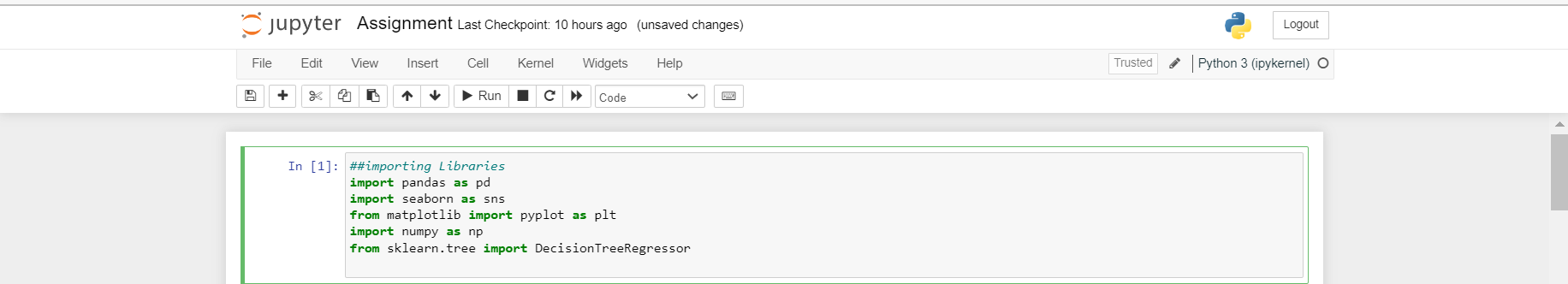


Fig 1: setup

**3.0 Importing Data set.**

The training and test data sets were imported using the pandas library and were named ‘housing\_train’ and ‘housing\_test’ respectively. The head of the 2 data sets was examined after importing them.

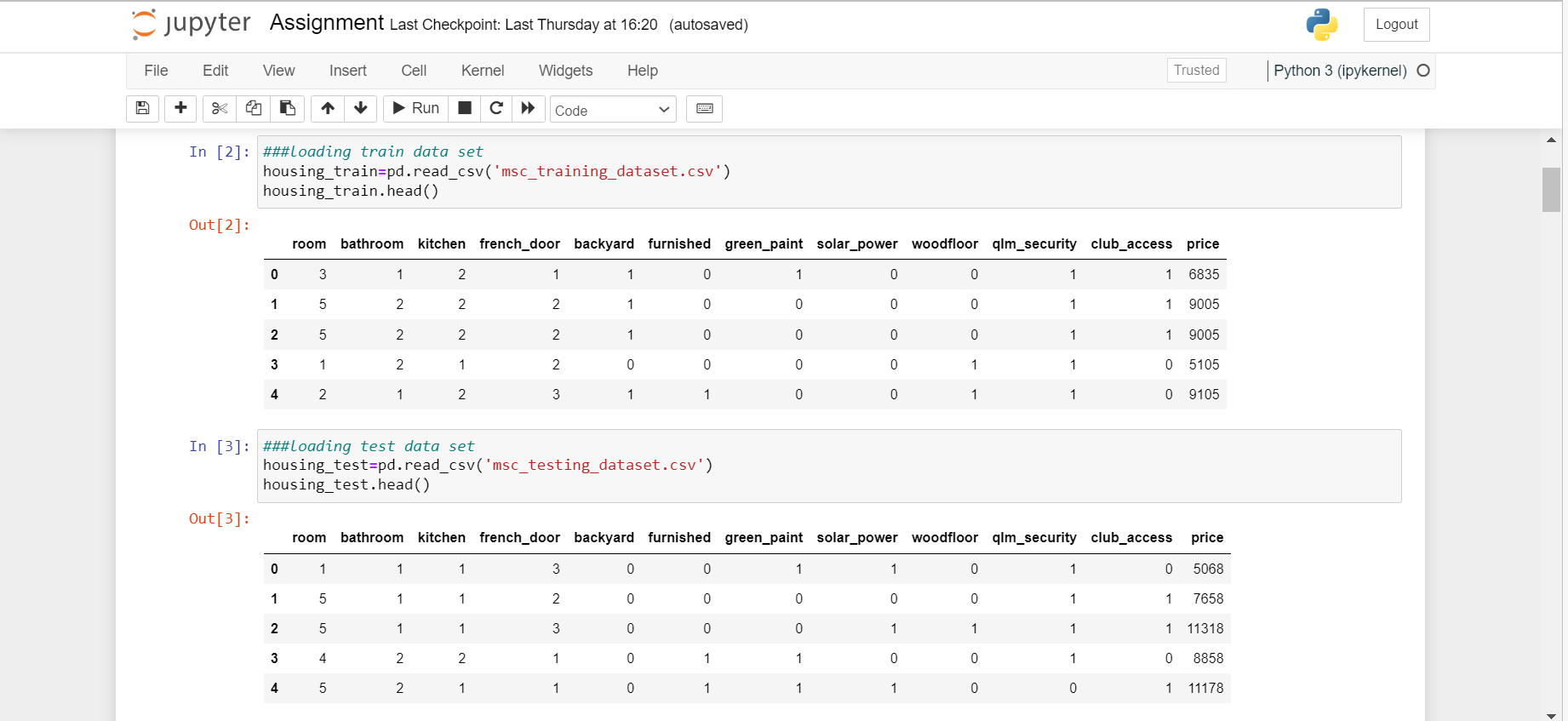


Fig 2: Importing data set.

**4.0 Exploratory Data Analysis.**

The train data set was explored to check for null values using the ‘isnull’ command and it was observed that the data set has no null values. It can also be observed that there is a total of 12 attributes in the data set.

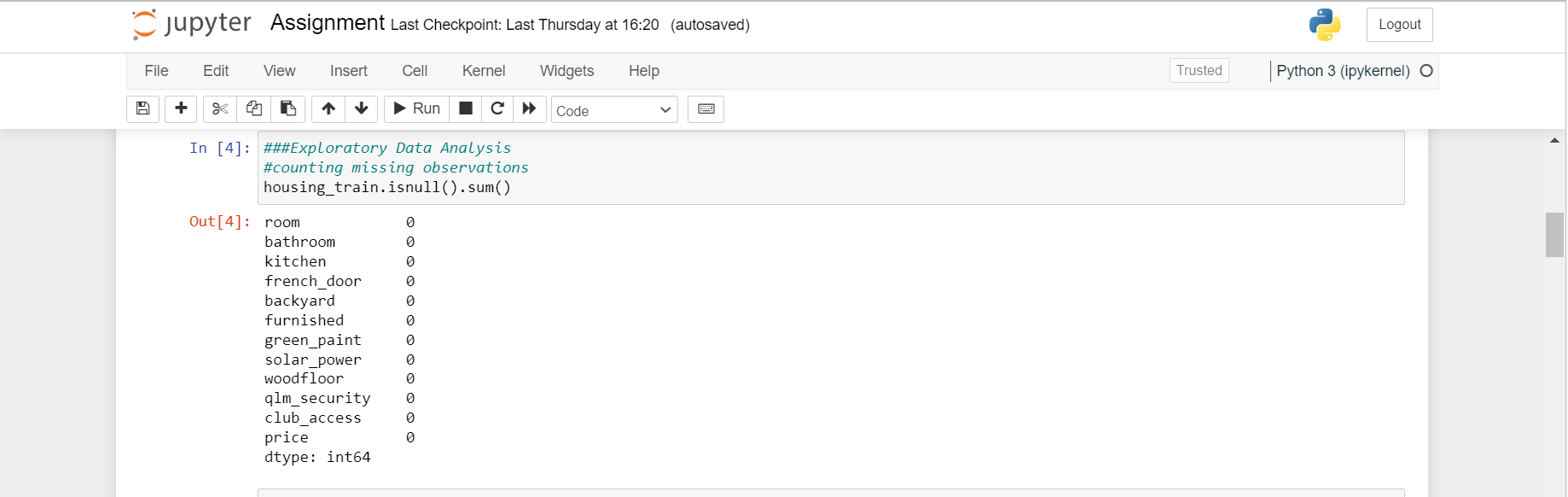


Fig 3: counting missing observations.

The descriptive statistics of the data set was also accounted for using the ‘describe’ command. From the result, we can see that there is a total of 3000 observations. Taking attribute ‘room’ as a case study, the column has an average value of 2.9, a standard deviation of 1.424281, minimum value of 1, and maximum value of 5. The first quartile value of this attribute is 2, the median value is 3 and the third quartile is 4. This explanation also applies to other attributes too.

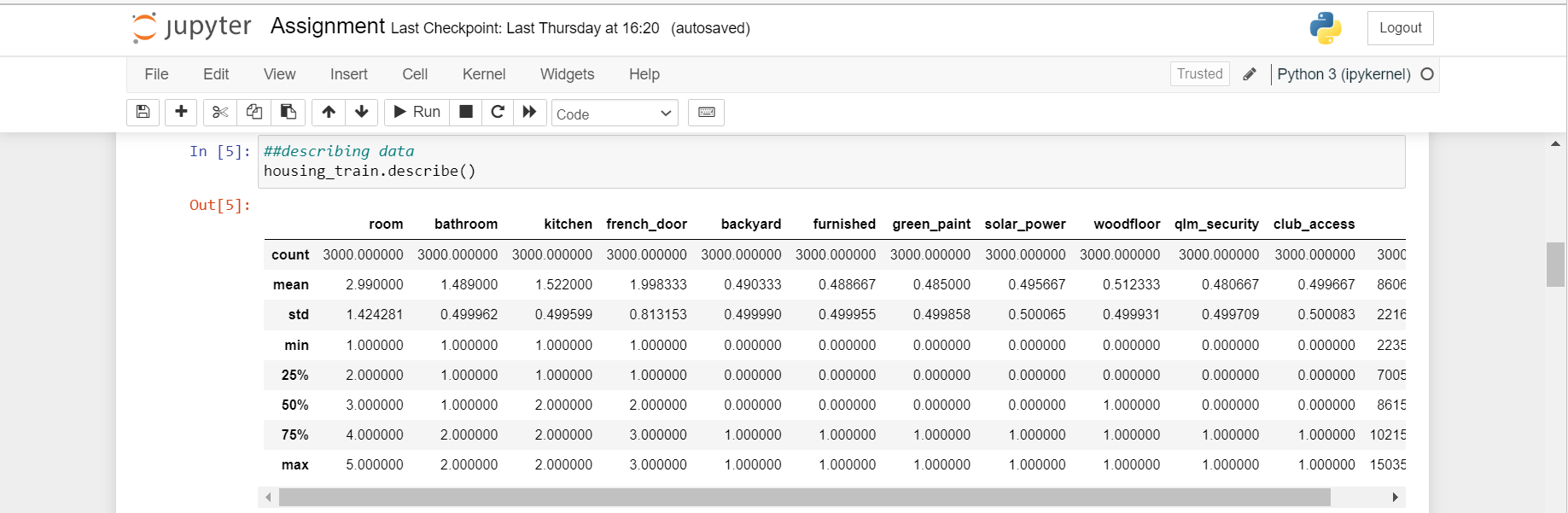


Fig 4: Descriptive analysis of dataset.

I also looked into how the attributes correlates to themselves using the ‘.corr’ command and then rounded the figures into 2 decimal place. It can be observed that the attributes have a very low correlation.

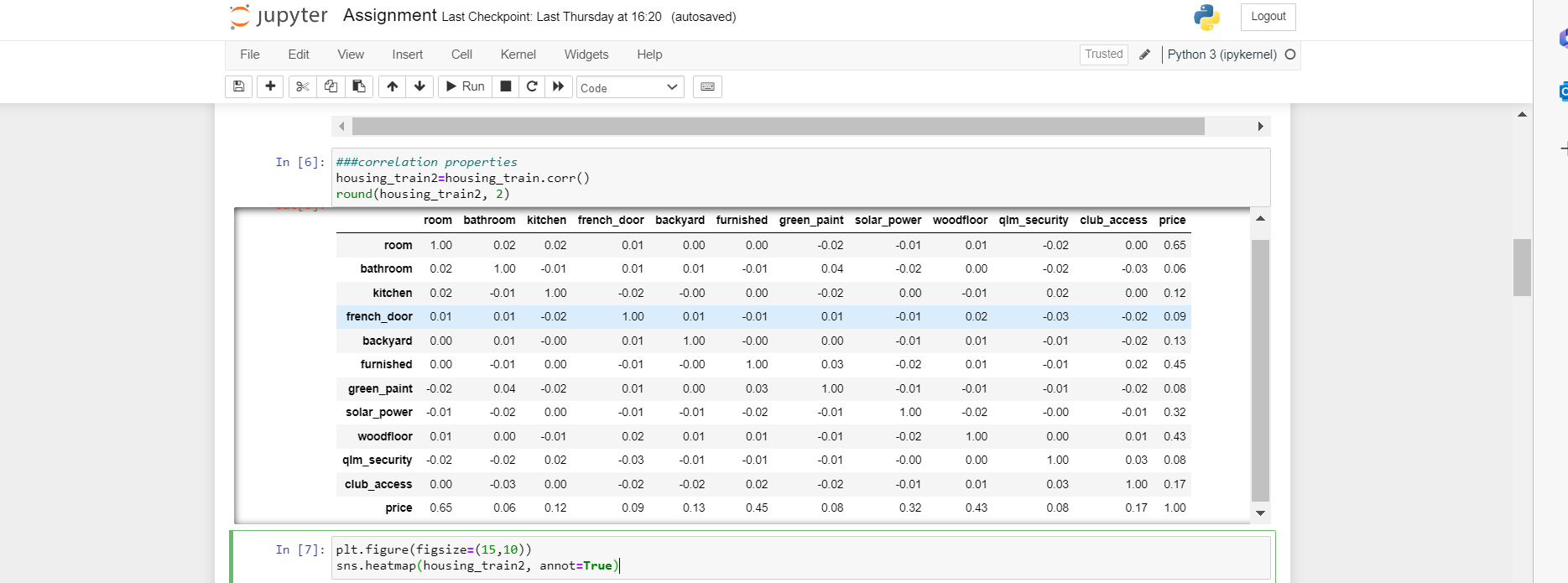


Fig 5: correlation properties

This is more evident in the correlation plot as values that are perfectly correlated have light colors and as the correlation reduces, the color gets deeper. However, the highest correlated value here is 0.65 which is still low, while others have a very low value.

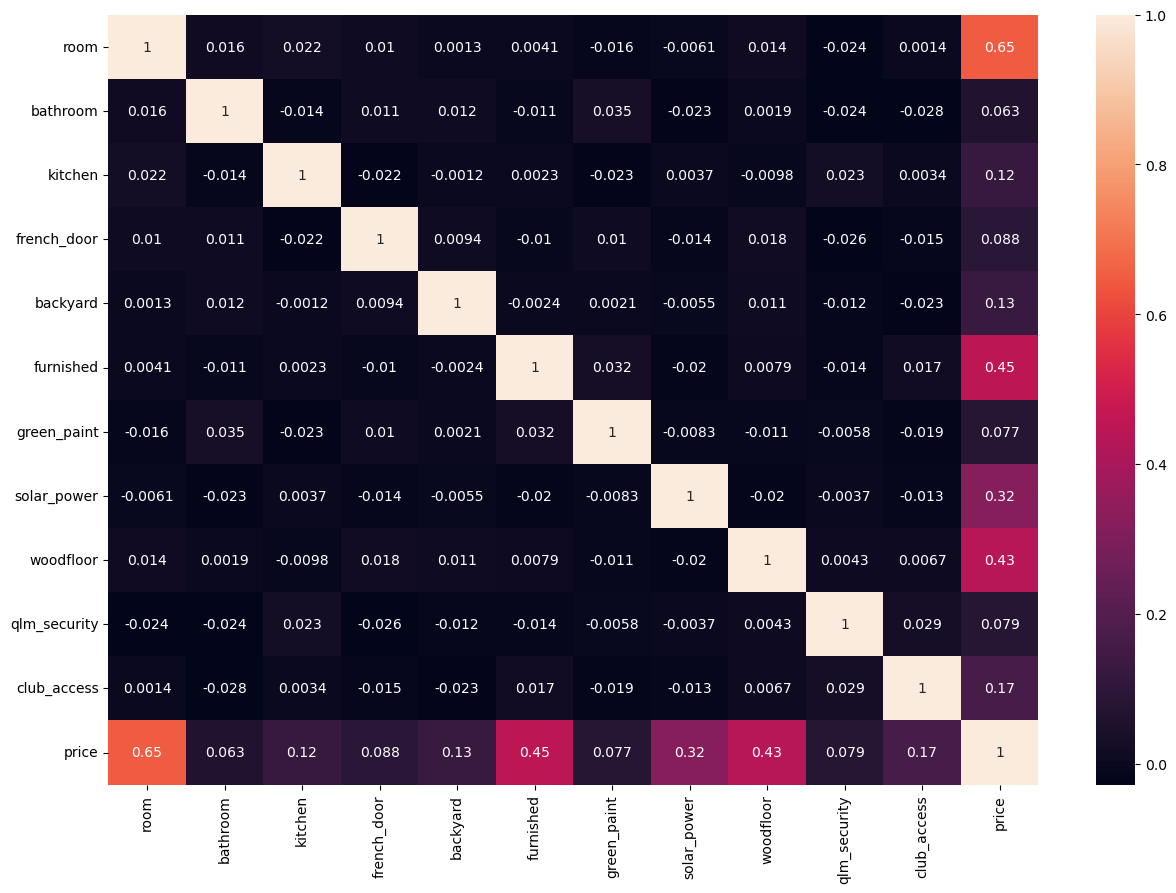


Fig 6: correlation plot

Lastly, since the train and test set has been splitted initially, the input and out put columns of each of the data sets were defined as seen in the code snippet.

This was achieved by dropping the price attribute and naming the resulting data as X\_train and X\_test respectively. Then the price column was selected and named the y\_train and y\_test for each data respectively.

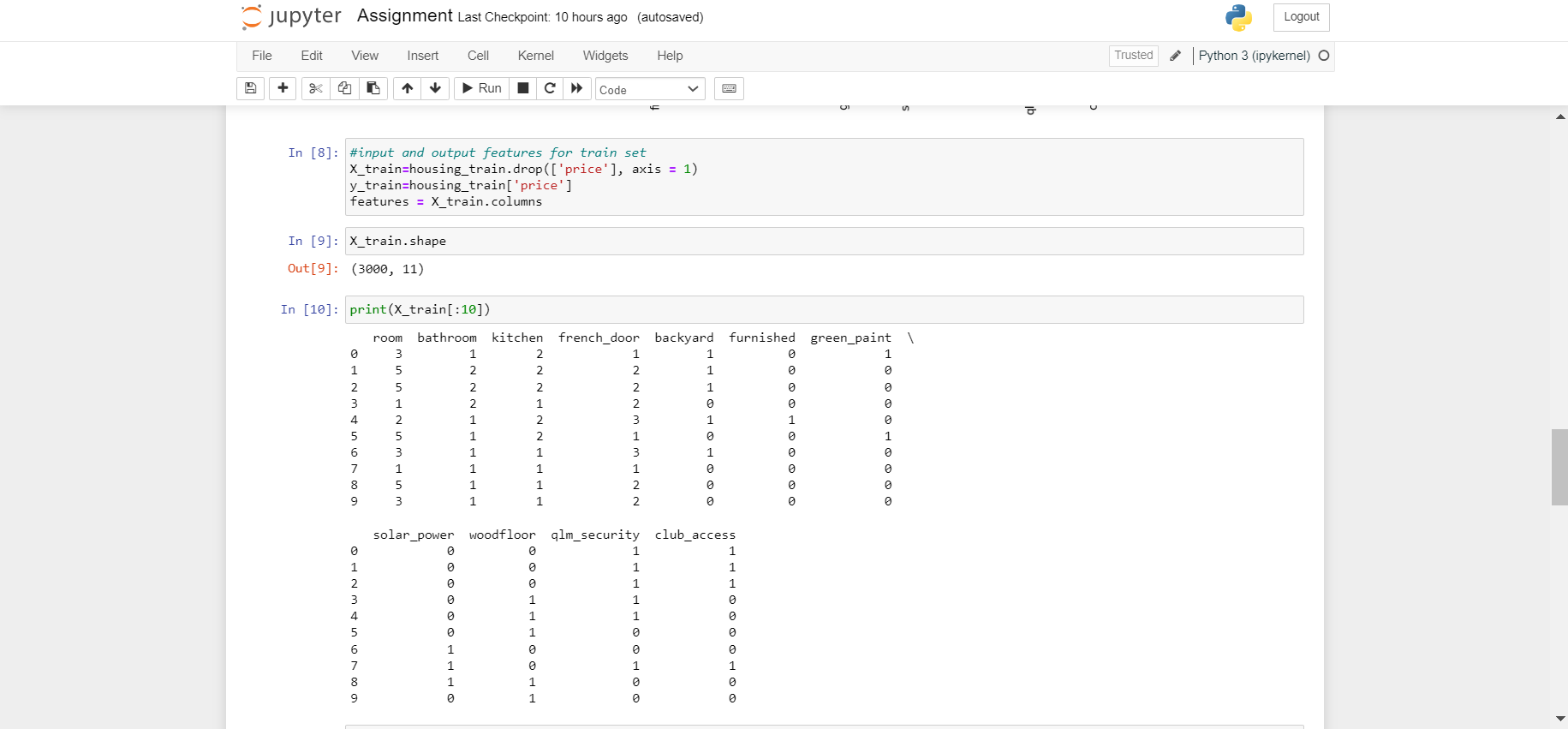


Fig 7: defining X and y train set.

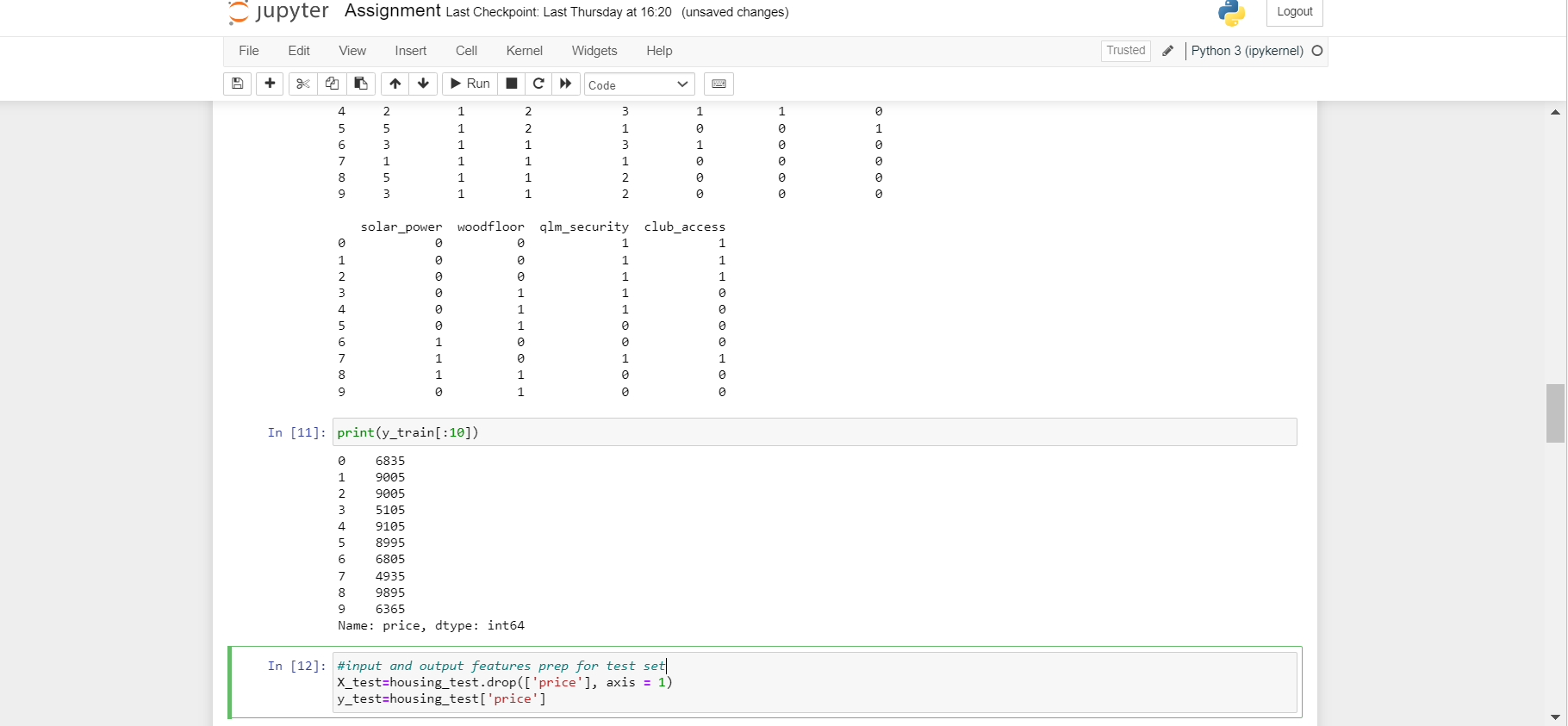


Fig 8: Defining X and y test set.

**5.0 Feature Importance.**

Here, we want to determine the feature(s) that are highly associated with the outcome, which are considered as important. Since we are dealing with numerical data here, decision tree regressor was used to derive the important features.

Hence, decision tree regressor was imported and the model was defined. This model was then fitted on the initially defined X\_train and y\_train and called DTR.

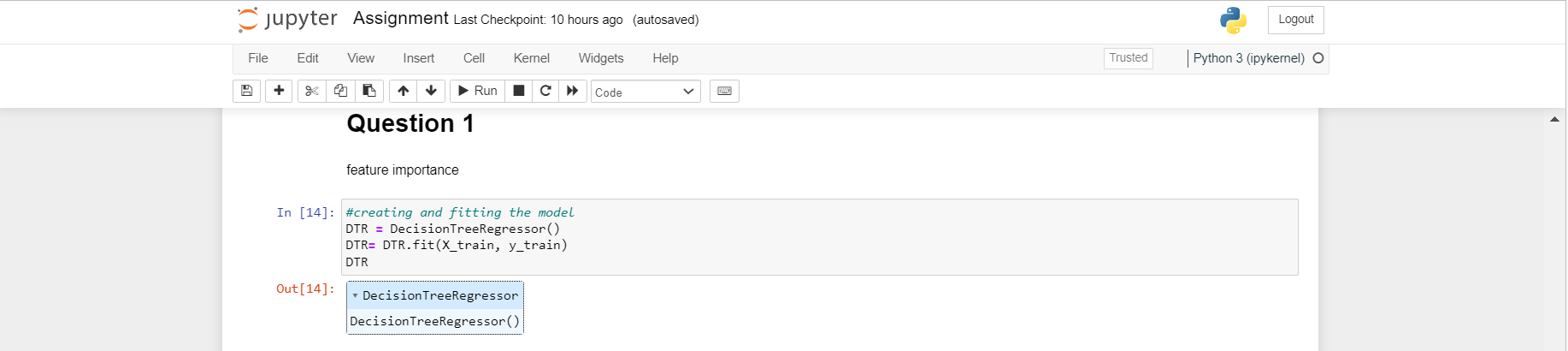


Fig 9: Model building and fitting.

Lastly, the feature importance was printed by recalling the regressor model, DTR on ‘feature\_importance\_’. A bar chart of this importance was plotted in ascending order, and it can be seen from this plot that ‘room’ had the highest importance with a value of 0.42670578, next to ‘furnished’ with a value of 0.20082385, and the third being ‘woodfloor’ with a value of 0.17567573.

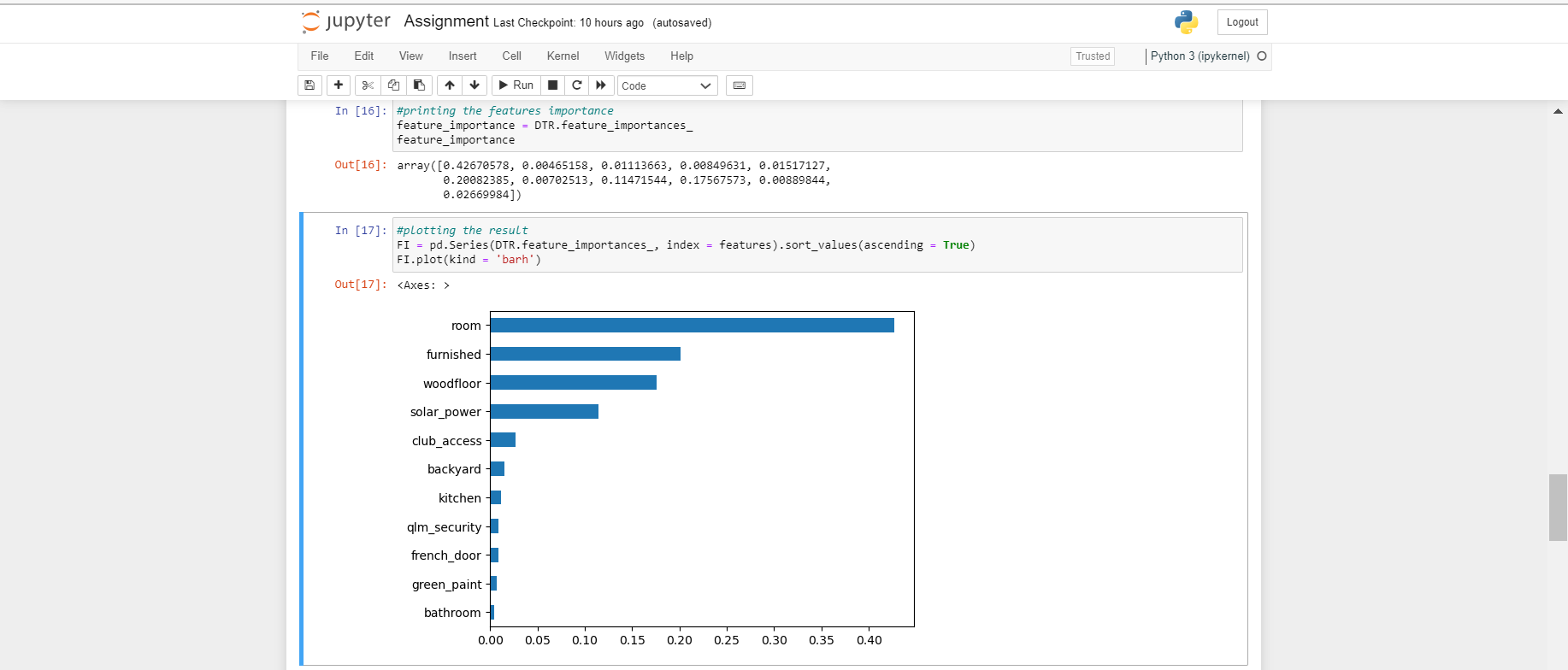


Fig 10: Features printing and plot.

**6.0 Predicting house prices.**

Here, a decision tree regressor was also used to predict the house prices. First, the model was built while specifying random state as 0, this was then fitted on the X\_train and y\_train set and named regressor. The resulting regressor was then predicted on the test set and named y\_pred.

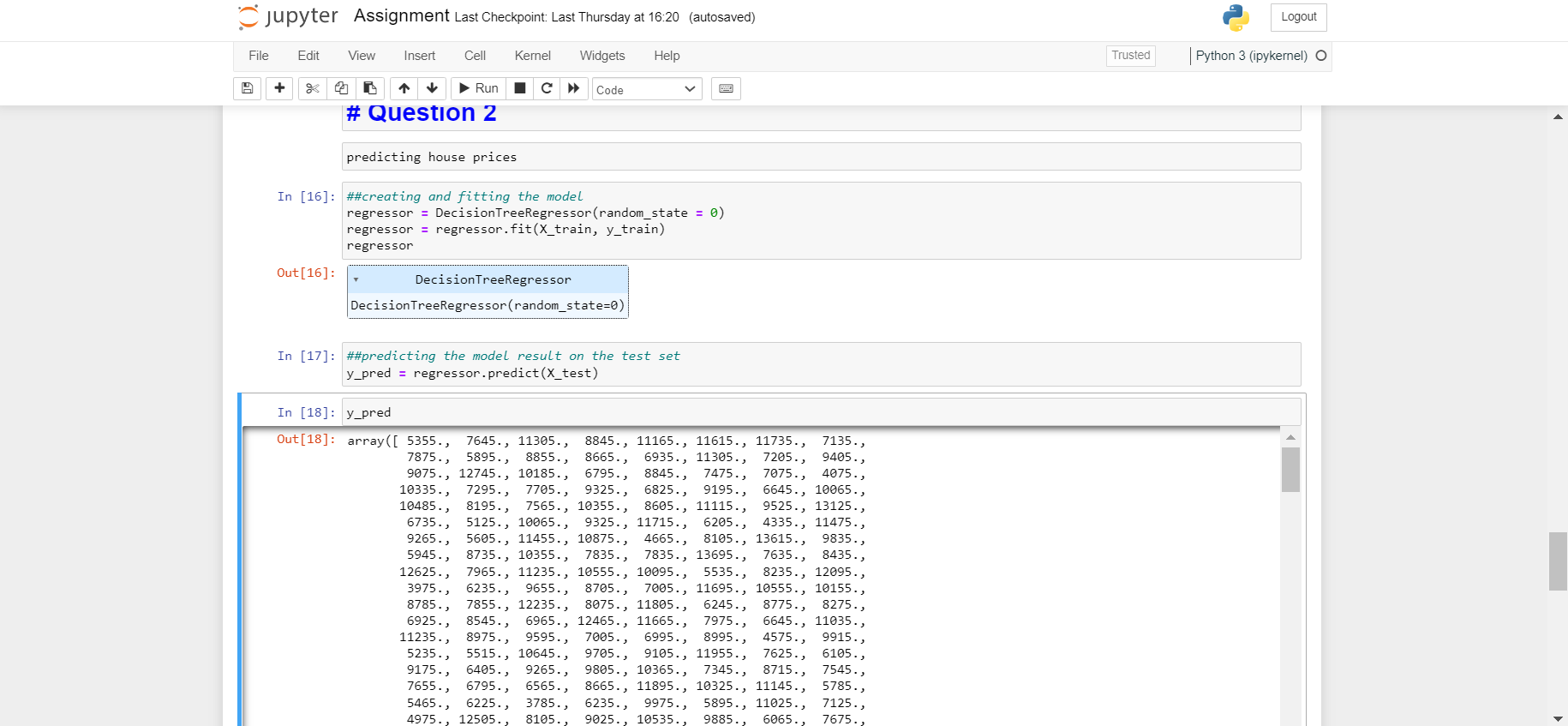


Fig 11: Regressor model building and Prediction.

The R-squared co-efficient of determination of the model was checked on both the train and test set. It can be seen below that the model is slightly overfitting on the training set (R2= 1.000) but has a decent performance on the testing set (R2 = 0.978).

Finally, with the help of pandas dataframe, the true value of house prices was compared with the predicted values, and this was then exported in csv format. At the end, a scatter plot of the real values in comparison to the predicted values was plotted. It can be observed that the results of the predicted values are almost perfectly fitted on the true values, which shows that the decision tree regressor is a good model for this data set.

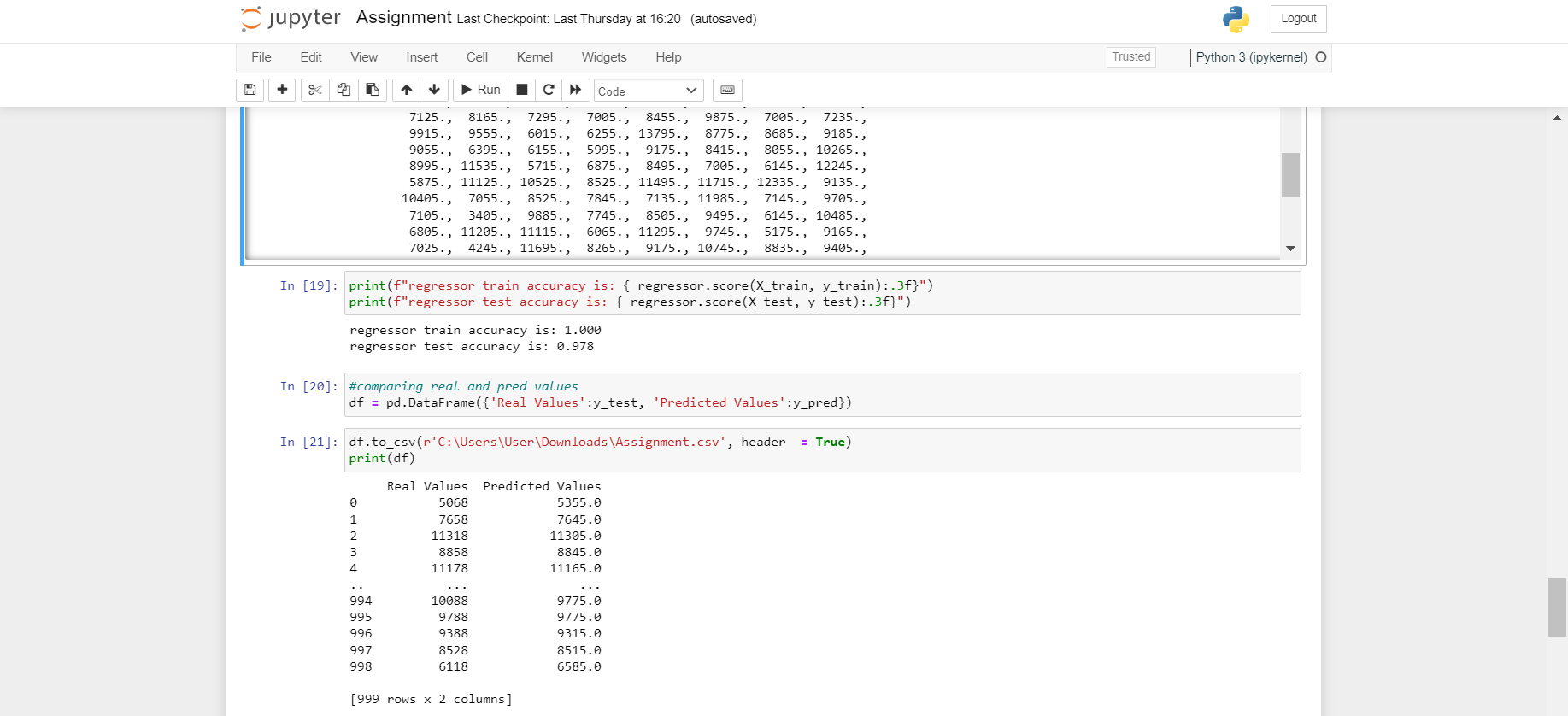


Fig 12: Comparing real and predicted values.