FINAL PROJECT

**Categorization of Houses into Different Price Range using ML Algorithms from American Housing Survey 2017 Dataset**

Prof. Sergul Aydore

EE-551A Engineering Programming Python

02 December 2019

## Project By:

## Monami Mukhopadhyay

Student ID - 10442486

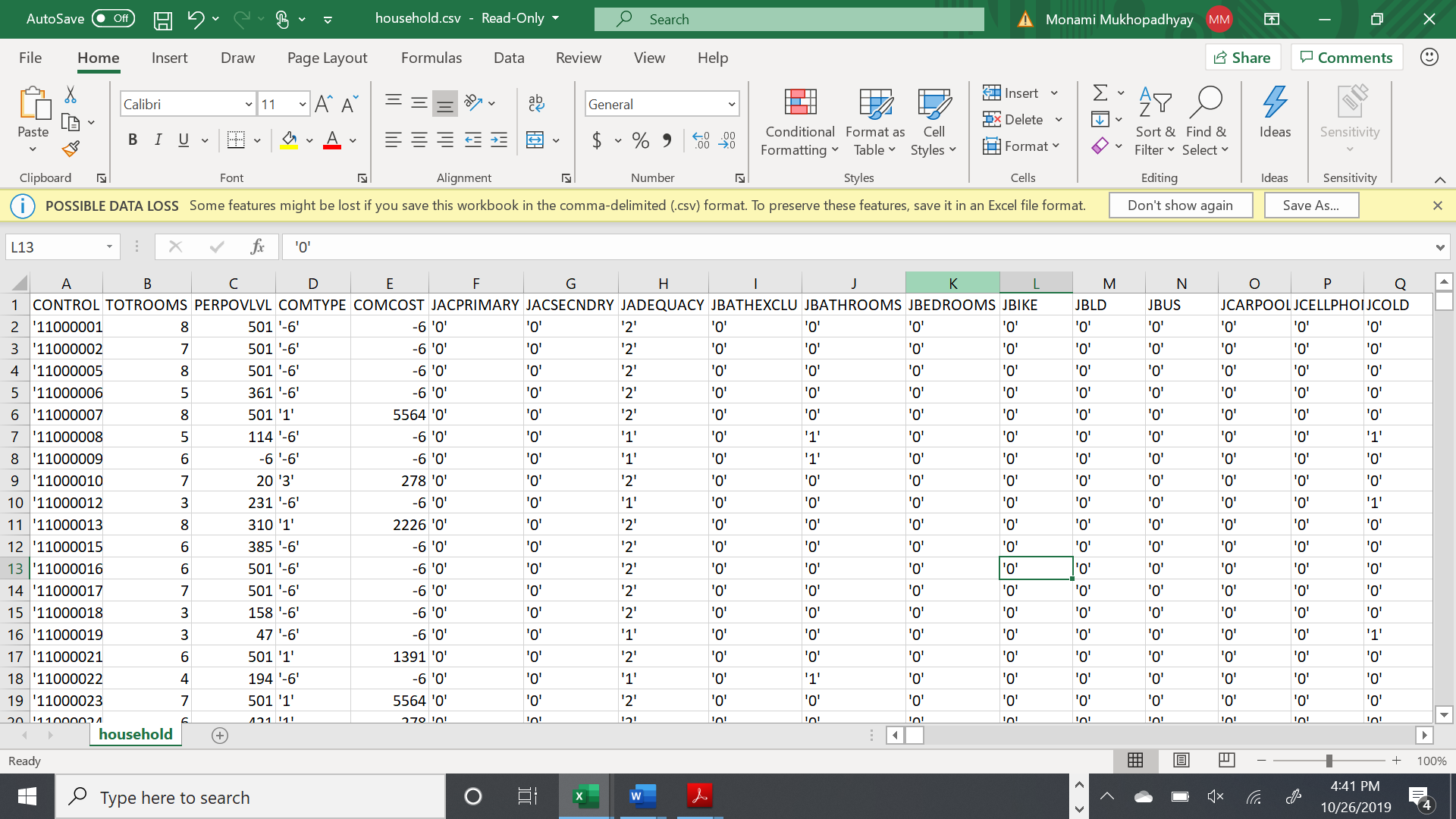
**Problem Statement:**

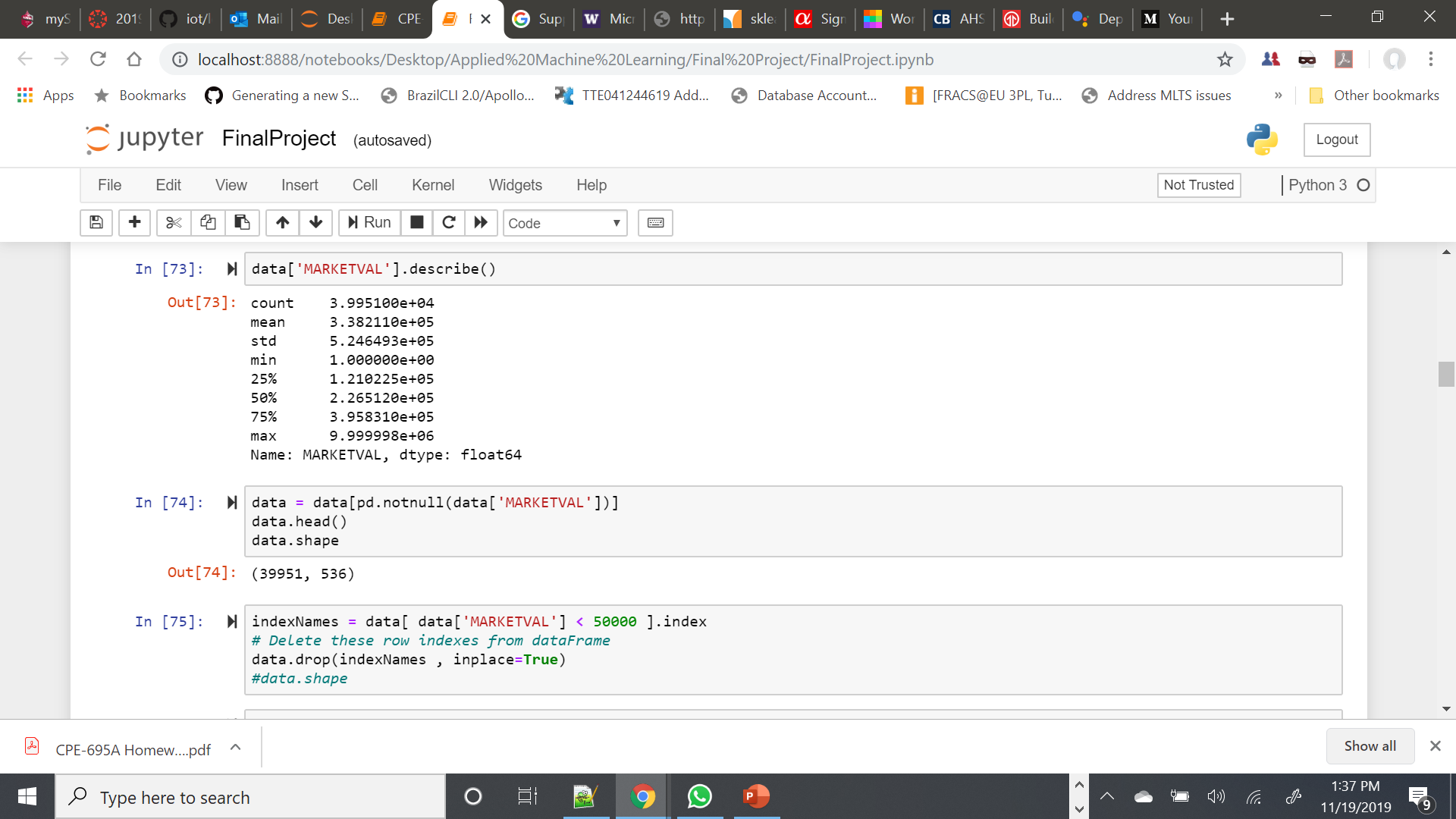
The main goal of this project is to predict the range of selling price of house with a high degree of predictive accuracy using various Machine Learning methods. Given house sale data or explanatory variable such as number of bedrooms, number of bathrooms in unit, housing cost, annual commuting cost etc, the machine learning model is built. Next, the model is evaluated with respect to test data, and plot the prediction and coefficients.

**Data:**

I am using American Housing Survey 2017 data <https://www.census.gov/programs-surveys/ahs/data/2017/ahs-2017-public-use-file--puf-/ahs-2017-national-public-use-file--puf-.html> (household.csv in AHS 2017 National PUF v3.0 CSV.zip). Since the dataset is very big (441 MB), I am just providing the link. It could not be uploaded in github repo. There is another csv file called AHSDICT\_15NOV19\_21\_17\_31\_97\_S.csv that consist of the mapping information of each feature name to their actual meaning and data type information. This file is already present in github repo. In the AHS microdata, the basic unit is an individual housing unit. Each record shows most of the information associated with a specific housing unit or individual, except for data items that could be used to personally identify that housing unit or individual. The dataset comprises of housing data features like TOTROOMS(Number of rooms in unit), PERPOVLVL(Household income as percent of poverty threshold (rounded)), COMCOST(Total annual commuting cost), JBATHROOMS(Number of bathrooms in unit), UNITSF(Square footage of unit), JGARAGE(Flag indicating unit has a garage or carport), JFIREPLACE(Flag indicating unit has a useable fireplace) etc., and target column as MARKETVAL(Current market value of unit) to evaluate model and also check which amongst all features is the most correlated feature for price prediction.

Two primary datasets used are:  
AHS 2017 Data - <https://www.census.gov/programs-surveys/ahs/data.2017.html> (household.csv in AHS 2017 National PUF v3.0 CSV.zip)  
AHS Codebook Feature Name Mapping - <AHSDICT_15NOV19_21_17_31_97_S.csv>





**Pre-requisites:**

* Python 3.7.0
* Numpy
* Pandas
* Scipy
* Scikit-learn
* Matplotlib

**Project Implementation:**

I defined machine learning models such as Random Forest, kNN, Decision Tree and Logistic Regression using most of the explanatory variables describing every aspect of residential homes and predict the price range of each home.

**Determine Independent Variable, Dependent Variable**

**Preprocessing Data**

**Collect Data**

**Calculate Prediction on Test**

**Get the Best Price Range, then calculate coefficient using ML Algorithms**

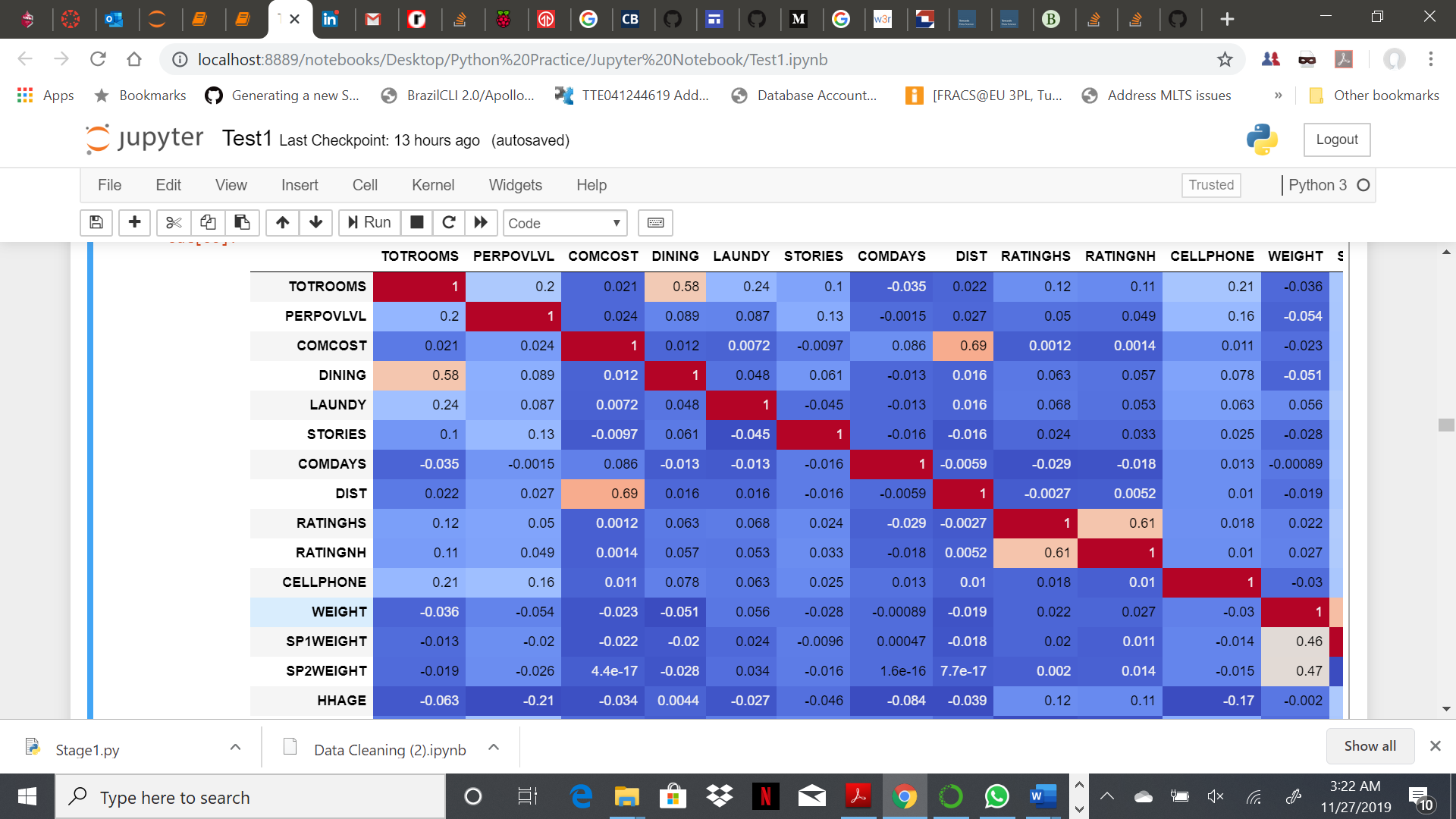
**Get Predictions**

**Calculating the Influential Variables**

**Data Cleaning:**

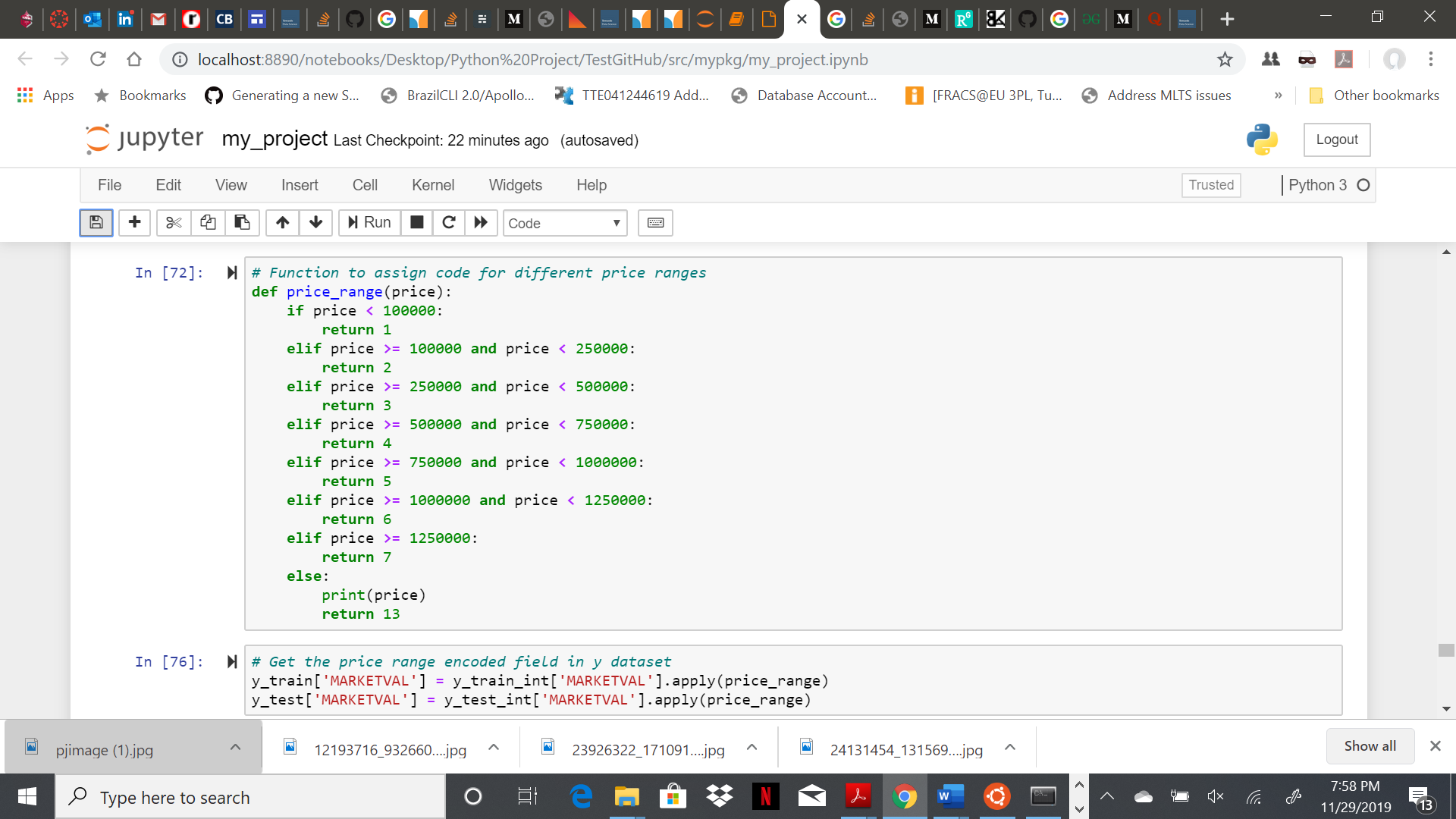
The dataset was cleaned to make it free from erroneous or irrelevant data. By filling up missing values, removing rows, and reducing data size, the final dataset was (36358 rows X 1007 columns). All the features were evaluated, and the results were used to reduce the dimensionality of the dataset and to check which amongst all features is the most correlated feature for price prediction.

Finally, the data was critically analyzed for the distribution of data, and derived relevant statistics. Then the data was checked for inconsistencies and removed accordingly.



**Feature Encoding:**

To predict the price range for houses, I divided the entire range of MARKETVAL values into smaller ranges and encoded them into set of classes as [1,2,3,4,5,6,7,13]. The function used for the same is as below:



**Algorithms Implemented:**

In this project, my aim was to implement algorithms which will be able to learn and classify the new observations to correct house price ranges. I decided to use below machine learning algorithms for the same-

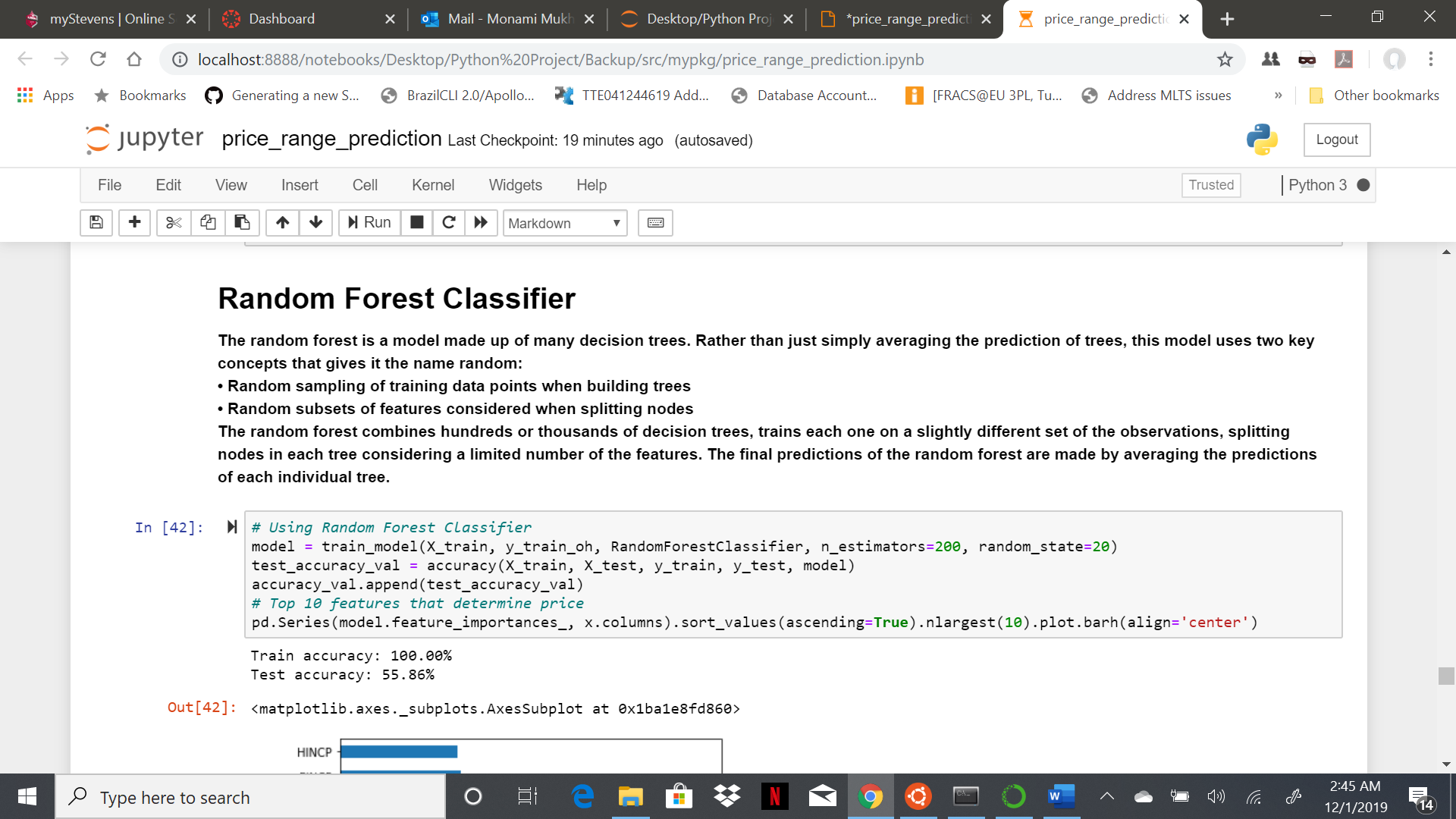
* Random Forest (RandomForestClassifier)
* Logistic Regression (LogisticRegression)
* K-Nearest Neighbor (KNeighborsClassifier)
* Decision Tree (DecisionTreeClassifier)

**Algorithm 1 – Random Forest:**

The random forest is a model made up of many decision trees. Rather than just simply averaging the prediction of trees, this model uses two key concepts that gives it the name random:

* Random sampling of training data points when building trees
* Random subsets of features considered when splitting nodes

The random forest combines many decision trees, trains each one on a slightly different set of the observations, splitting nodes in each tree considering a limited number of the features. The final predictions of the random forest are made by averaging the predictions of each individual tree.

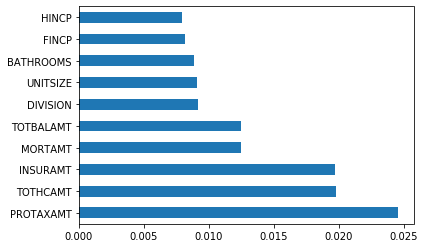


Results: With RandomForestClassifier, the accuracy score were as below:

Training Accuracy – 100.00%

Testing Accuracy – 55.86%

I also plotted a bar graph representing the top 10 features based on their importance in determining the house price range.



These 10 features are as below:

PROTAXAMT- Monthly Property Tax Amount

TOTHCAMT- Monthly Total Housing Cost

INSURAMT- Monthly Homeowner or Renter Insurance Amount

MORTAMT- Monthly Total Mortgage Amount (all mortgages)

TOTBALAMT- Total Remaining Debt Across all Mortgages or Similar Debts for this Unit

DIVISION- Census Division

UNITSIZE- Unit Size (Square Feet)

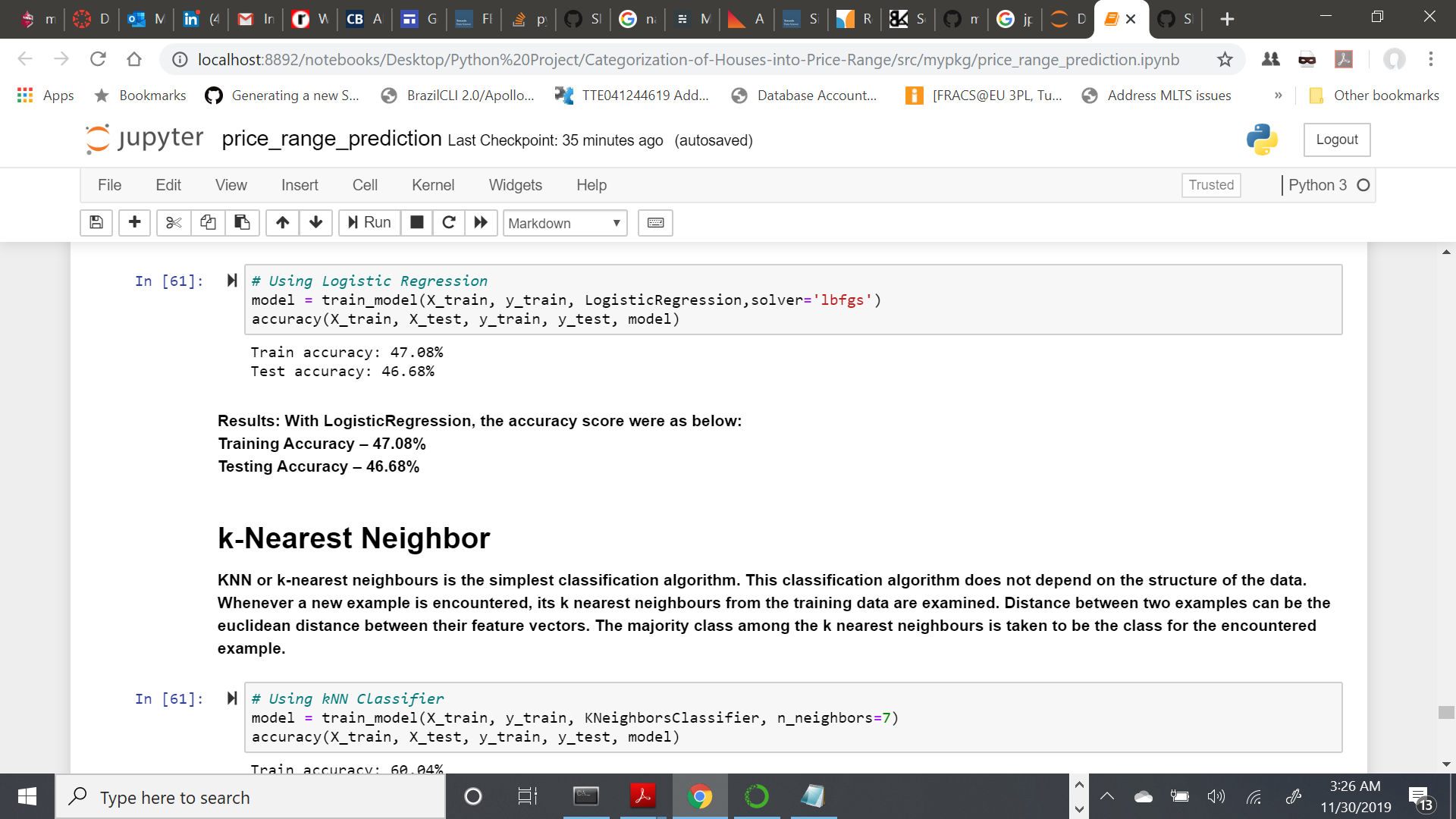
BATHROOMS- Number of Bathrooms in Unit

FINCP- Family Income (Past 12 Months)

HINCP- Household Income (Past 12 Months)

**Algorithm 2 – Logistic Regression:**

Logistic regression is one of the most fundamental and widely used Machine Learning Algorithms. Logistic regression is not a regression algorithm but a probabilistic classification model. Multi class classification is implemented by training multiple logistic regression classifiers, one for each of the K classes in the training dataset.



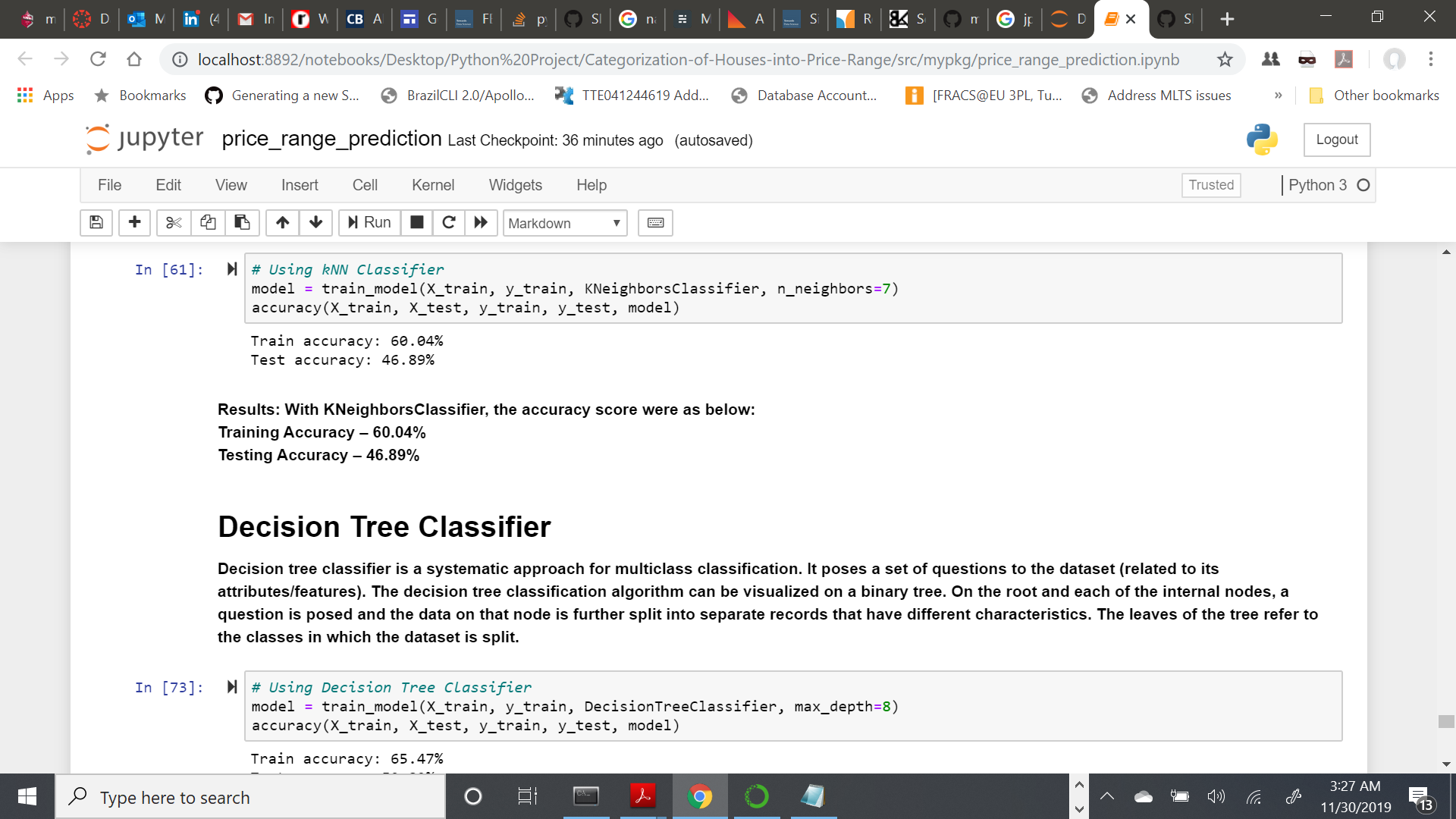
Results: With LogisticRegression, the accuracy score were as below:

Training Accuracy – 47.08%

Testing Accuracy – 46.68%

**Algorithm 3 – k-Nearest Neighbors:**

KNN or k-nearest neighbours is the simplest classification algorithm. This classification algorithm does not depend on the structure of the data. Whenever a new example is encountered, its k nearest neighbours from the training data are examined. Distance between two examples can be the euclidean distance between their feature vectors. The majority class among the k nearest neighbours is taken to be the class for the encountered example.



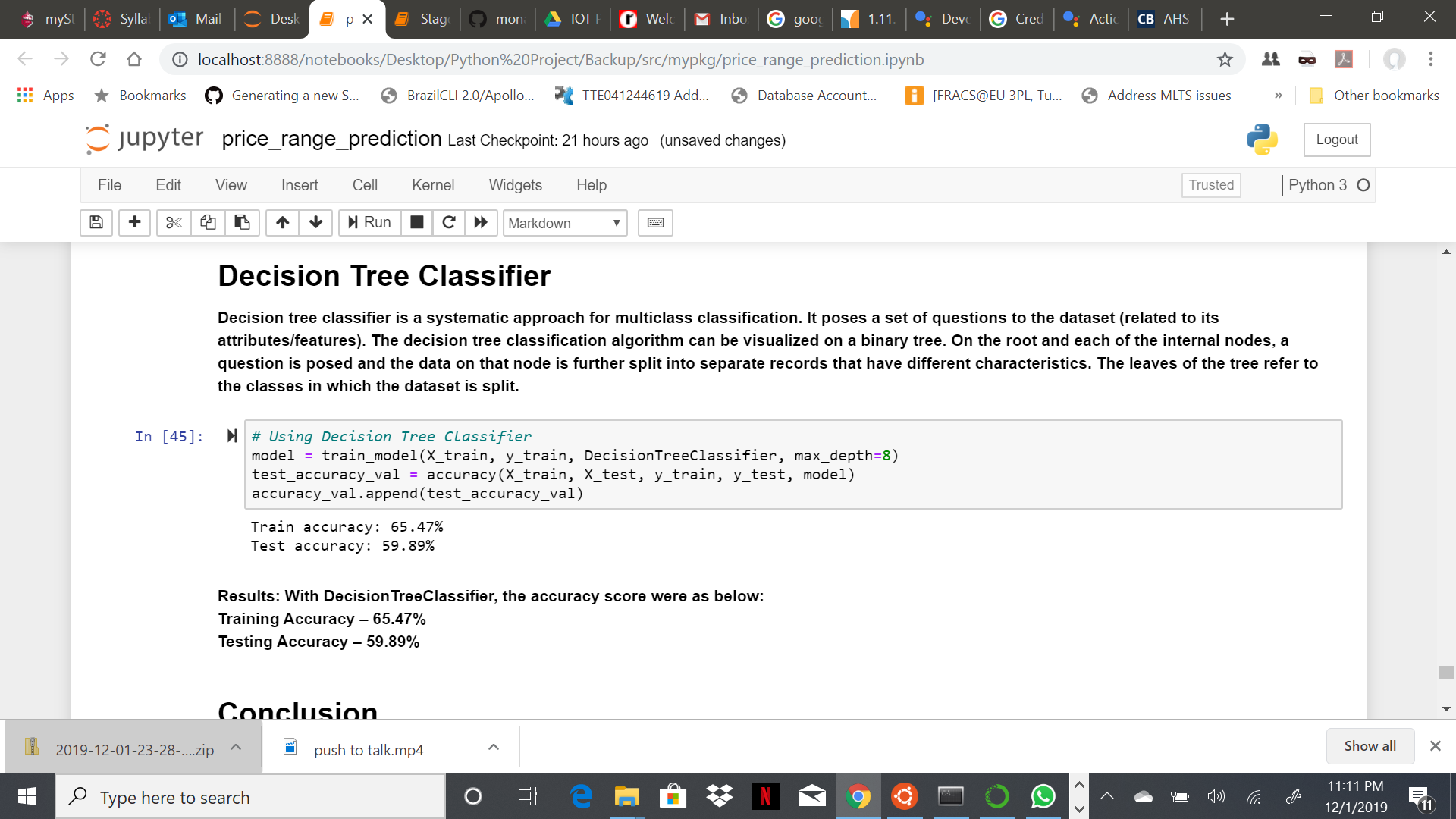
Results: With KNeighborsClassifier, the accuracy score were as below:

Training Accuracy – 60.04%

Testing Accuracy – 46.89%

**Algorithm 4 – Decision Tree:**

Decision tree classifier is a systematic approach for multiclass classification. It poses a set of questions to the dataset (related to its attributes/features). The decision tree classification algorithm can be visualized on a binary tree. On the root and each of the internal nodes, a question is posed and the data on that node is further split into separate records that have different characteristics. The leaves of the tree refer to the classes in which the dataset is split.

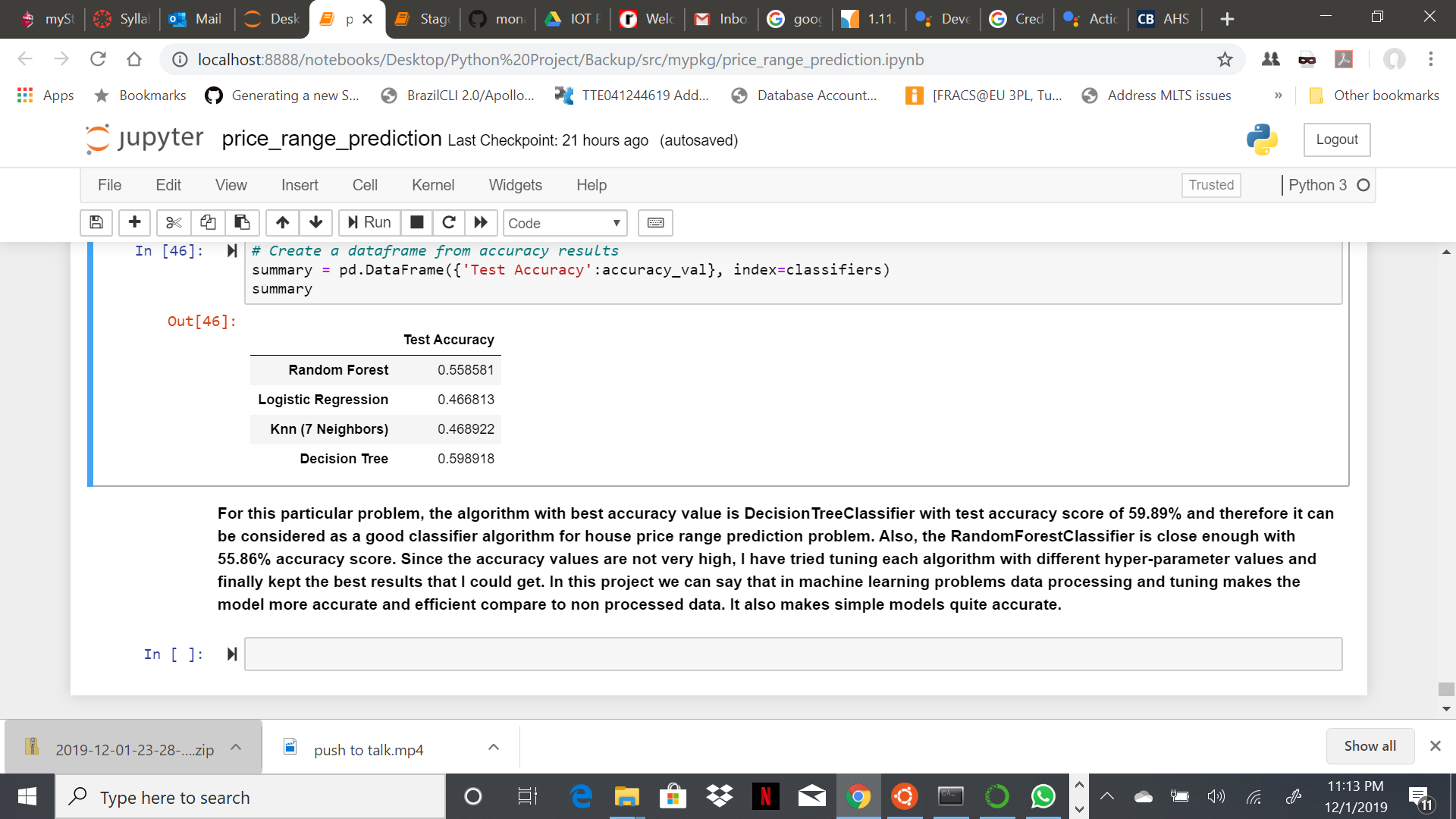


Results: With DecisionTreeClassifier, the accuracy score were as below:

Training Accuracy – 65.47%

Testing Accuracy – 59.89%

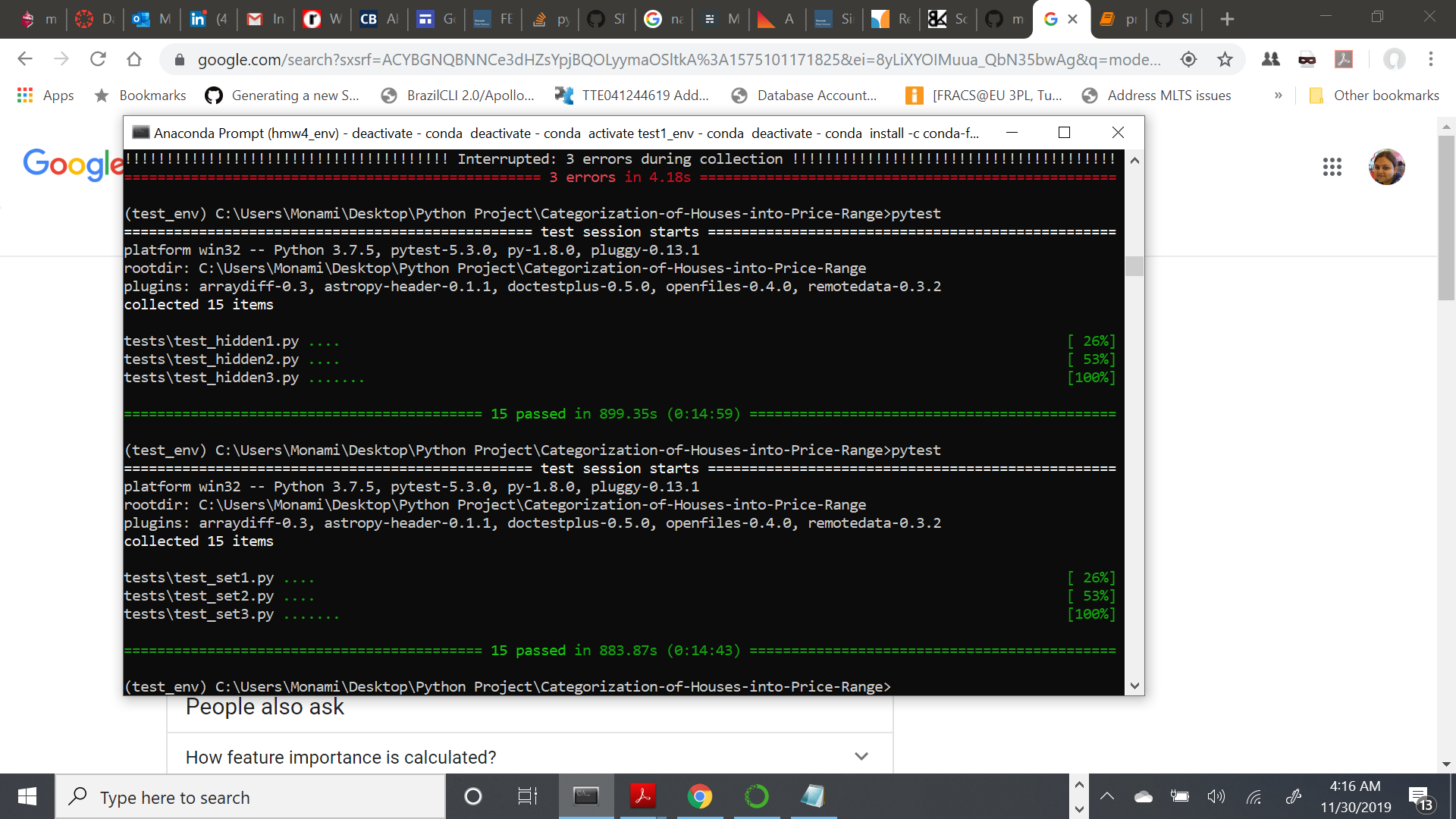
**Conclusions:**

****

For this particular problem, the algorithm with best accuracy value is DecisionTreeClassifier with test accuracy score of 59.89% and therefore it can be considered as a good classifier algorithm for house price range prediction problem. Also, the RandomForestClassifier is close enough with 55.86% accuracy score. Since the accuracy values are not very high, I have tried tuning each algorithm with different hyper-parameter values and finally retained the best results that I could get. In this project it can be said that in machine learning problems data processing and tuning makes the model more accurate and efficient compare to non-processed data. It also makes simple models quite accurate.

**Code Execution Details:**

Since the input AHS 2017 dataset is very big (around 441 MB), and I am executing four different ML algorithms, the entire code takes 15 minutes (approx..) to execute. I have also prepared 15 test codes for my code, which can be verified using pytest.



For my project, I have prepared two types of file for the same code - one .py and other .ipynb. The .py version is for testing using pytest. I am applying different machine learning algorithms and using a big dataset (441MB). Therefore, my .ipynb file became too large (around 90MB) which cannot be uploaded in github repo as it is. Therefore, I prepared a PDF copy of .ipynb file with all outputs that got generated, so that outputs of program are visible. Also, I cleared all outputs for .ipynb file and uploaded that as well. All the relevant documents along with the .ipynb with all generated outputs is present in google drive - <https://drive.google.com/drive/u/0/folders/1Or1xQ5GVPU1sCB3hY7V5pAKYYp-aP2Nd>

Github Repo –

<https://github.com/monamim1989/Categorization-of-Houses-into-Price-Range>

**References:**

Koehrsen, Will. “Beyond Accuracy: Precision and Recall.” Medium, Towards Data Science, 10

Mar. 2018, <https://towardsdatascience.com/beyond-accuracy-precision-and-recall-3da06bea9f6c>.

Jain, Rishabh. “Decision Tree. It Begins Here.” Medium, Medium, 24 June 2017,

<https://medium.com/@rishabhjain_22692/decision-trees-it-begins-here-93ff54ef134>.

Nielsen, & A., M. (1970, January 1). Neural Networks and Deep Learning. Retrieved from

<http://neuralnetworksanddeeplearning.com/chap2.html>.

MOAWAD, A. (2019, October 8). Neural networks and backpropagation explained in a simple

way. Retrieved from [https://medium.com/datathings/neural-networks-and- backpropagation-explained-in-a-simple-way-f540a3611f5e](https://medium.com/datathings/neural-networks-and-%20%20%20%20backpropagation-explained-in-a-simple-way-f540a3611f5e).