FINAL PROJECT REPORT

**Classifying Houses into Different Price Range using Neural Network and Tree Classifiers from American Housing Survey 2017 Dataset**

Prof. Shucheng Yu

CPE-695A Applied Machine Learning

11 December 2019

## Project By:

## Darp Raithatha

## Likith Nandigam

## Monami Mukhopadhyay

**Problem Statement:**

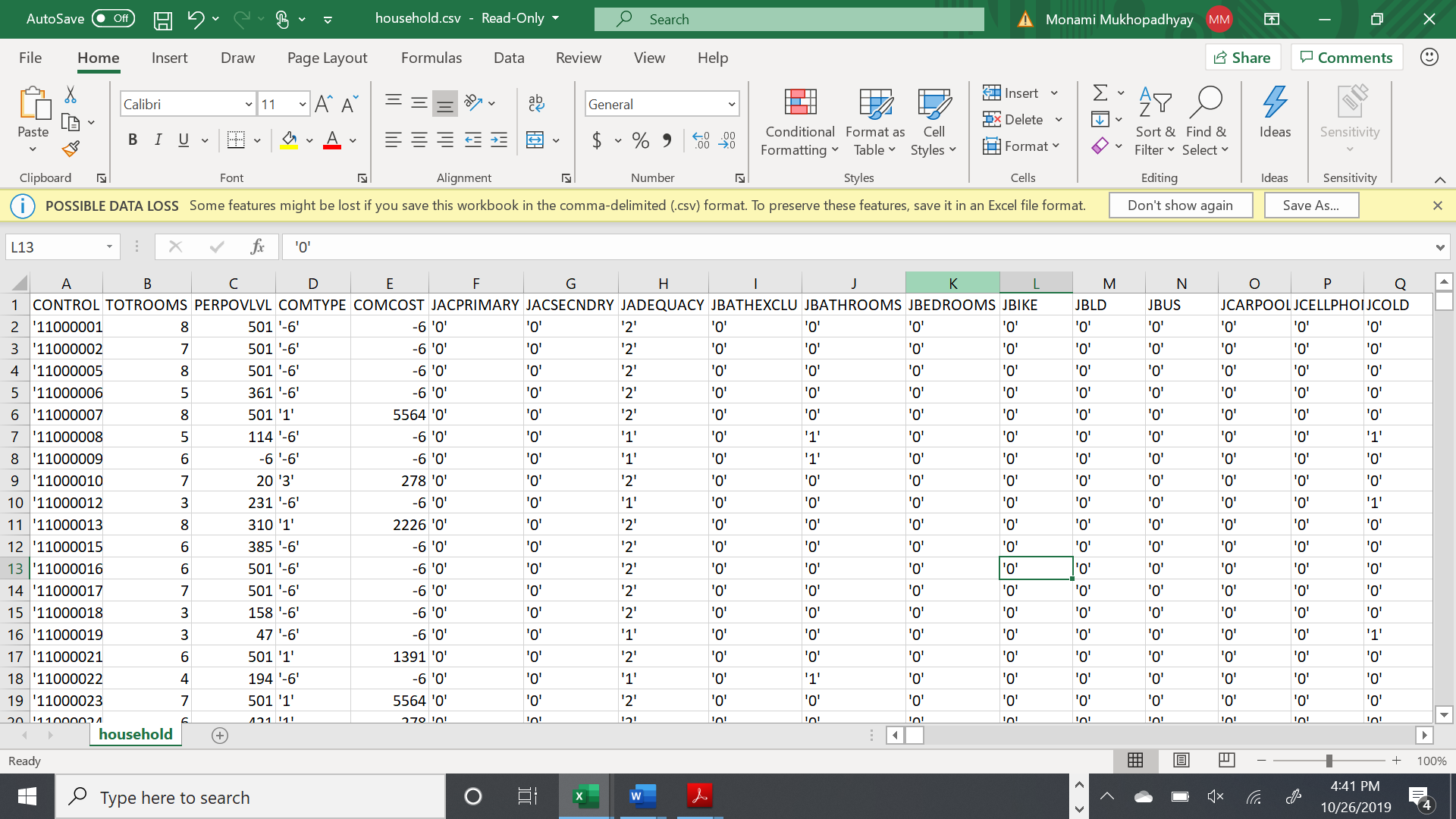
The main goal of this project is to predict the range of selling price of house with a high degree of predictive accuracy using various Machine Learning methods. Given house sale data or explanatory variable such as number of bedrooms, number of bathrooms in unit, housing cost, annual commuting cost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

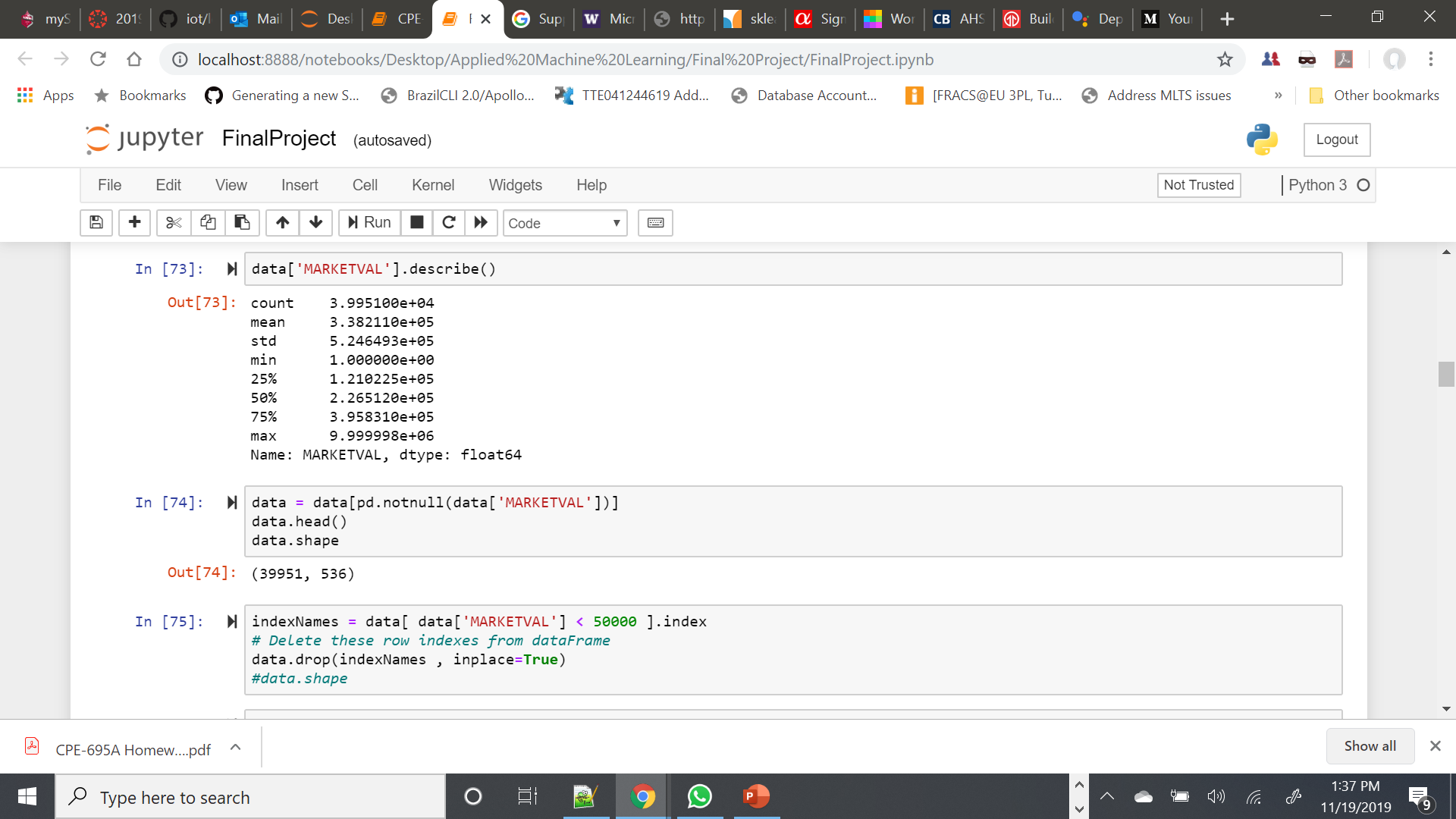
**Data:**

We are using American Housing Survey 2017 (household.csv in AHS 2017 National PUF v3.0 CSV.zip). Since the dataset is very big, we have uploaded it in our google drive. 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. Our 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 – <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:**

We defined machine learning models such as Random Forest, kNN, Decision Tree, Multi-layer Perceptron Classifier, AdaBoost, and, Ensemble Method (Stacking) 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**

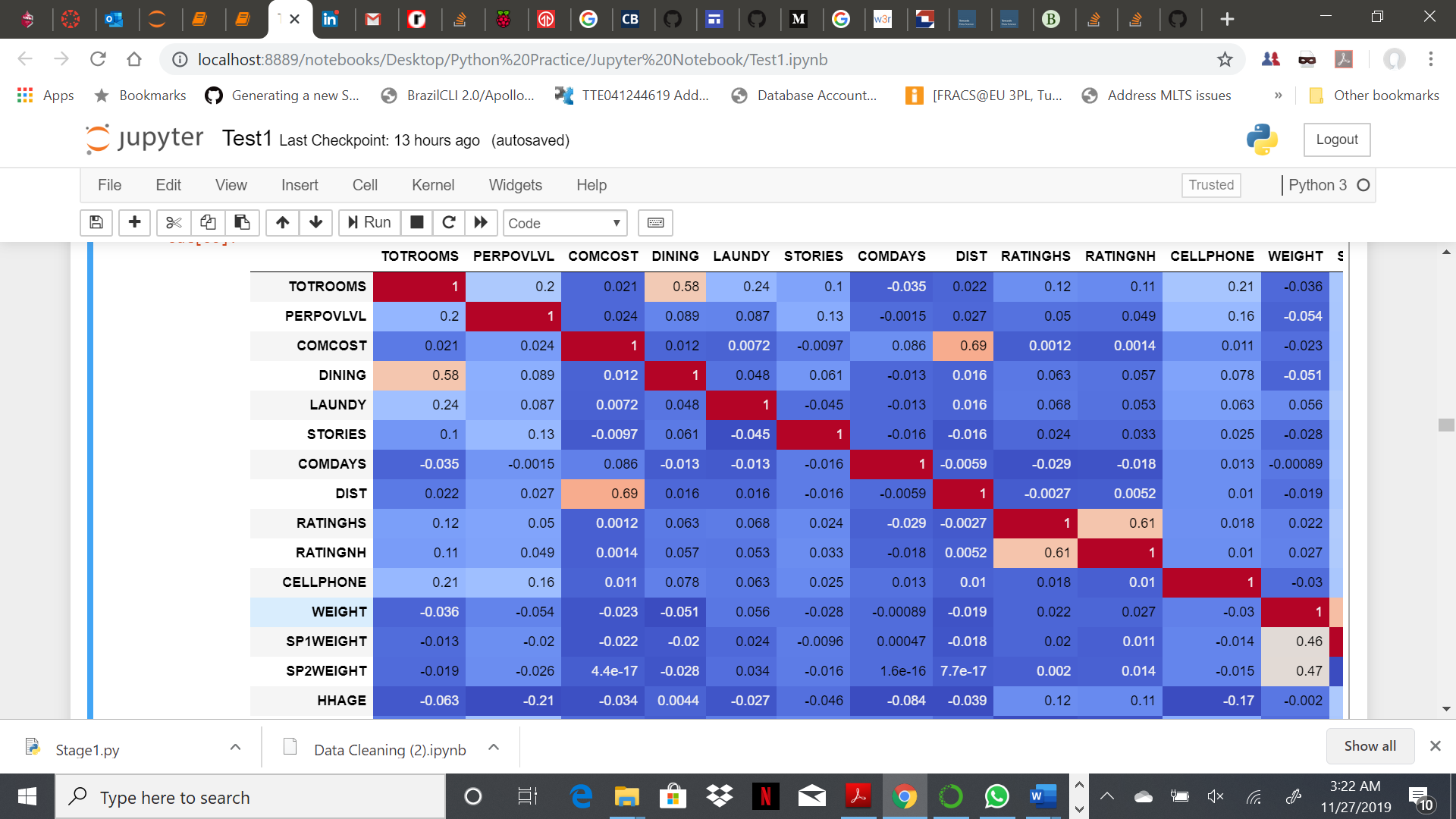
**Get Predictions**

**Calculating the Influential Variables**

**Data Cleaning:**

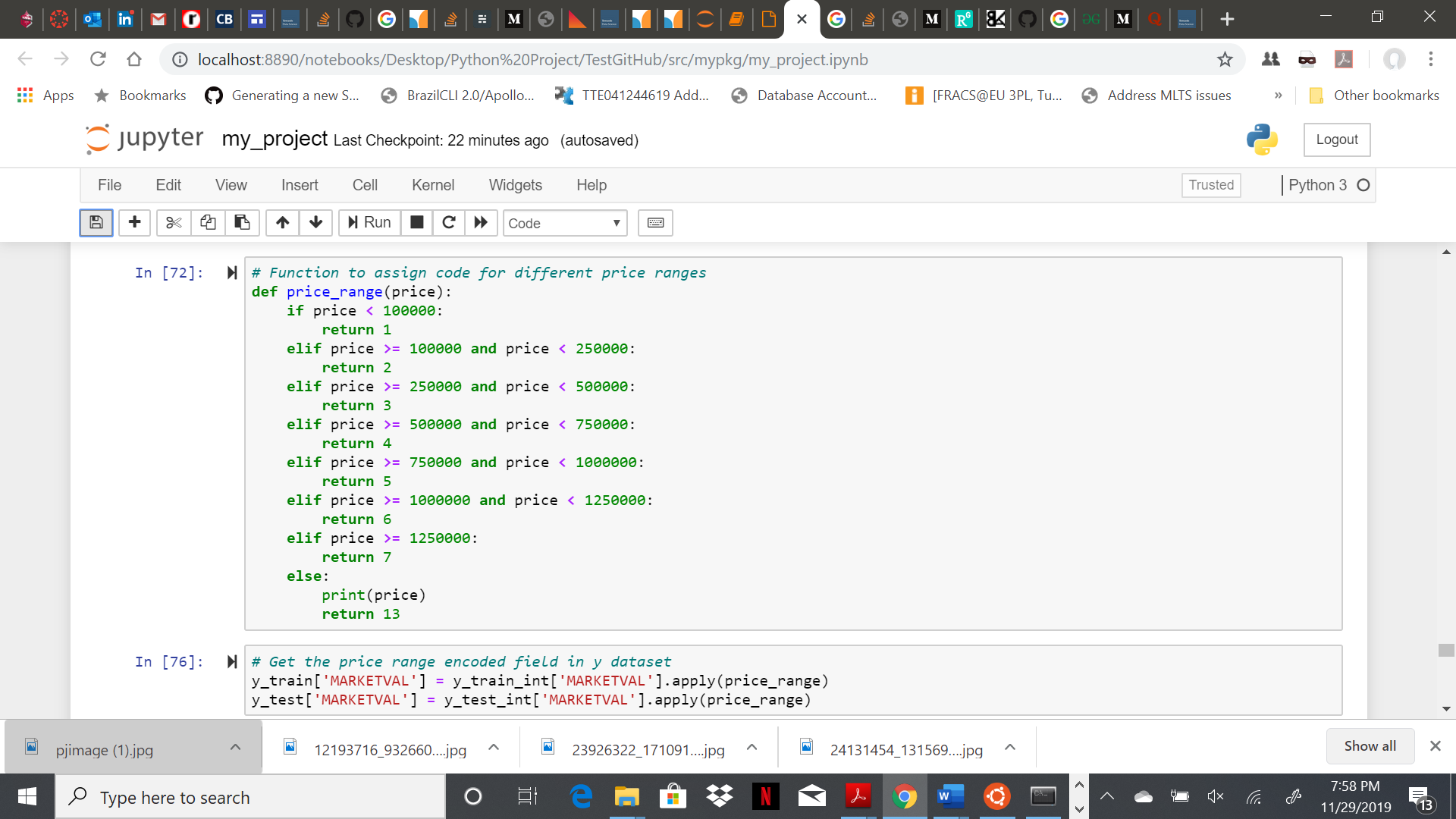
The dataset was cleaned to make it free from erroneous or irrelevant data. By filling up missing values, removing rows, and reducing data size, the final dataset was (36358 rows X 526 columns). All the features were evaluated, and the results were used to reduce the dimensionality of the dataset and 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, we 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:**

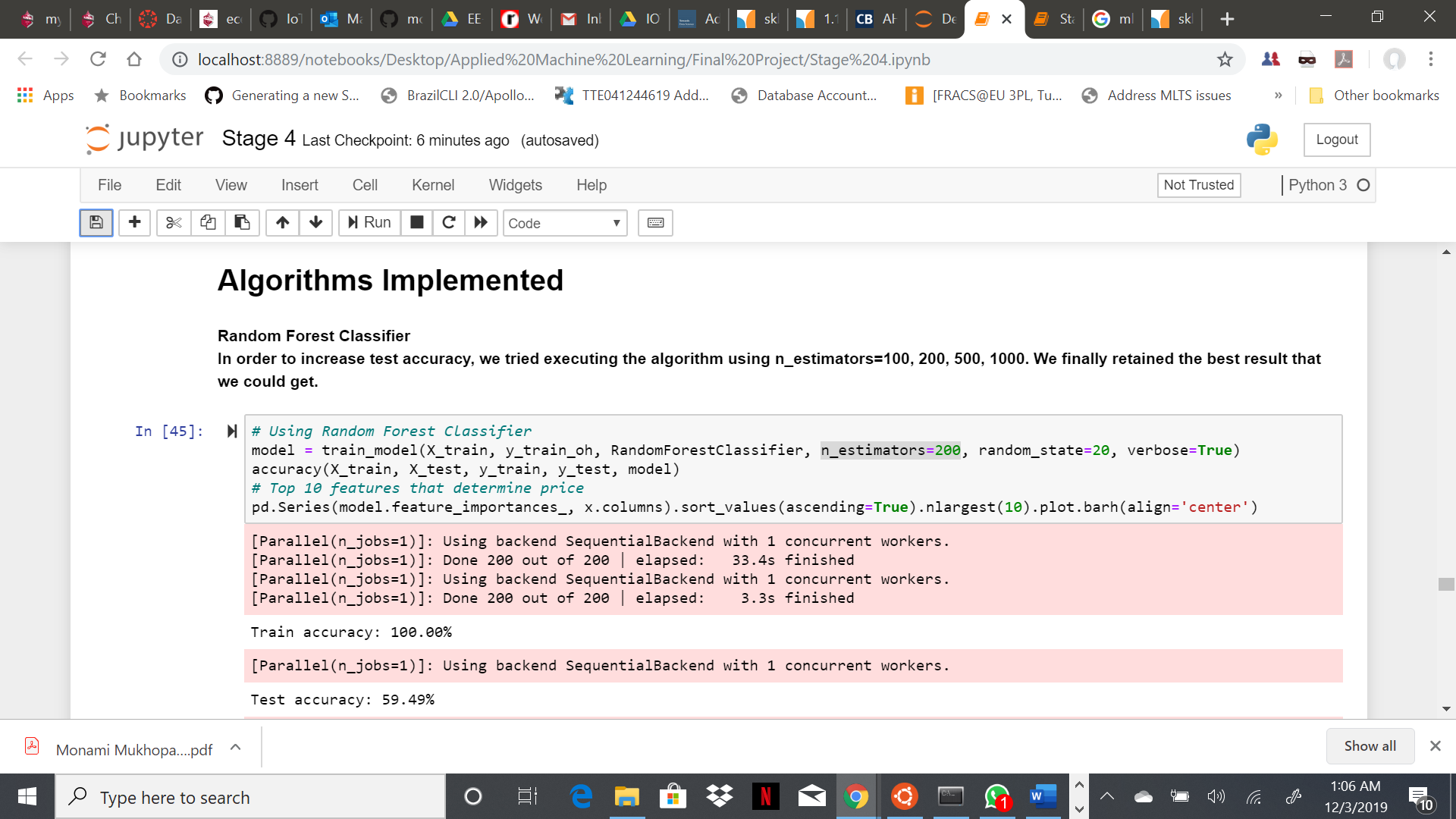
In this project, our aim was to implement algorithms which will be able to learn and classify the new observations to correct house price ranges. So, we decided to use below machine learning algorithms for the same-

* Random Forest (RandomForestClassifier)
* K-Nearest Neighbor (KNeighborsClassifier)
* Decision Tree (DecisionTreeClassifier)
* Multi-layer Perceptron Classifier (MLPClassifier)
* AdaBoost Classifier (AdaBoostClassifier)
* Ensemble Method (Stacking)

**Algorithm 1 – Random Forest:**

**The random forest is a classification algorithm consisting of many decision trees. It uses bagging and feature randomness when building each individual tree to try to create an uncorrelated forest of trees whose prediction by committee is more accurate than that of any individual tree.**

**In order to increase test accuracy, we tried executing the algorithm using n\_estimators=100, 200, 500, 1000. We finally retained the best result that we could get.**

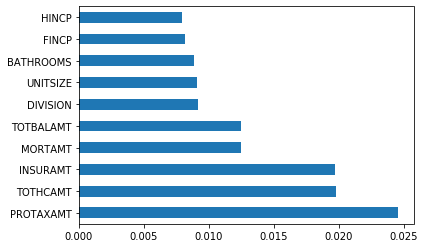


Results: With RandomForestClassifier, the accuracy score were as below:

Training Accuracy – 100.00%

Testing Accuracy – 59.49%

We 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

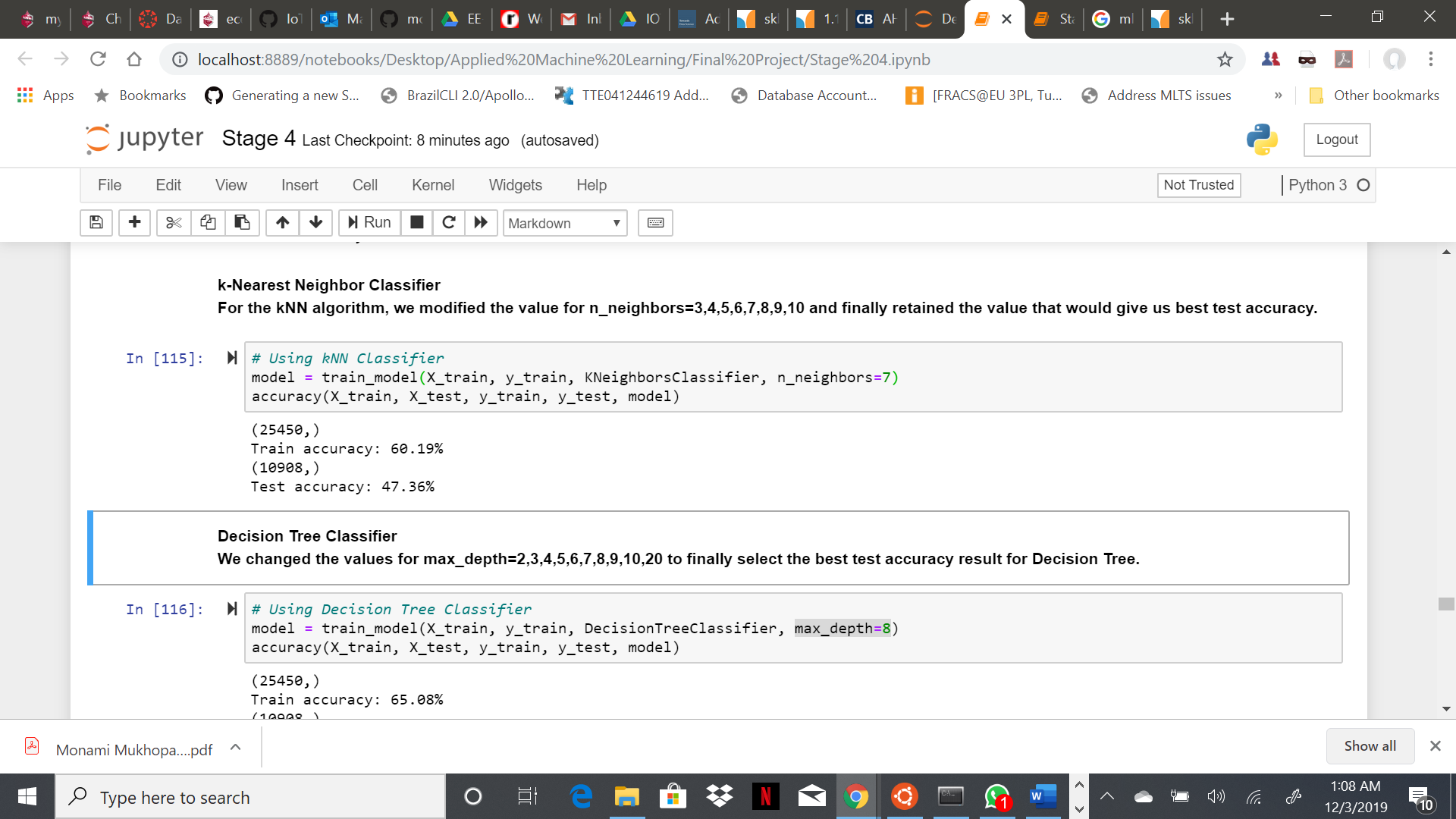
FINCP- Family Income (Past 12 Months)

HINCP- Household Income (Past 12 Months)

**Algorithm 2 – 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.**

**For the kNN algorithm, we modified the value for n\_neighbors=3,4,5,6,7,8,9,10 and finally retained the value that would give us best test accuracy.**



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

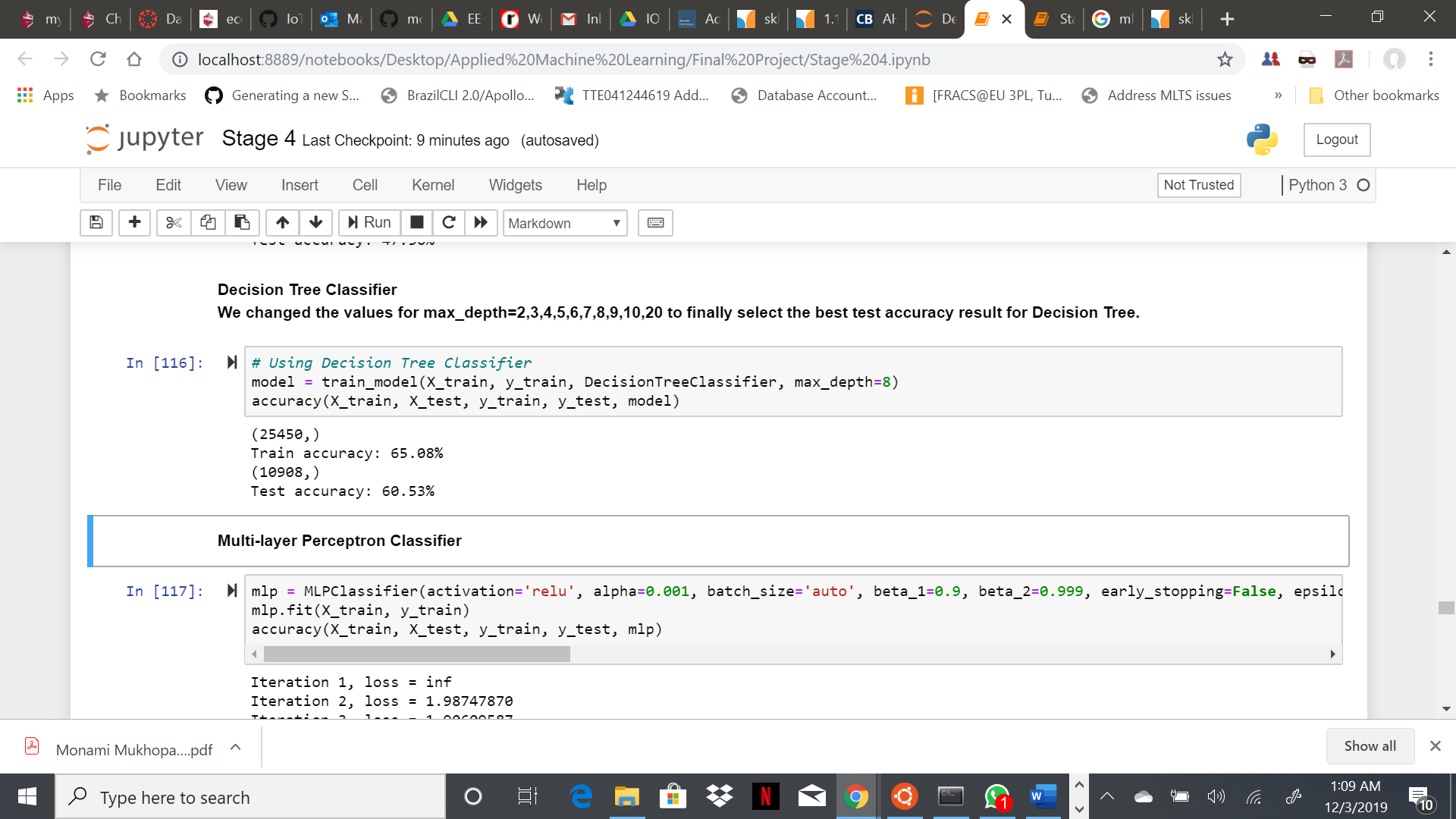
Training Accuracy – 60.19%

Testing Accuracy – 47.36%

**Algorithm 3 – Decision Tree:**

Decision Tree Classifier is a simple and widely used classification technique. Decision Tree Classifier poses a series of carefully crafted questions about the attributes of the test record. Each time it receives an answer, a follow-up question is asked until a conclusion about the class label of the record is reached.

**We changed the values for max\_depth=2,3,4,5,6,7,8,9,10,20 to finally select the best test accuracy result for Decision Tree.**



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

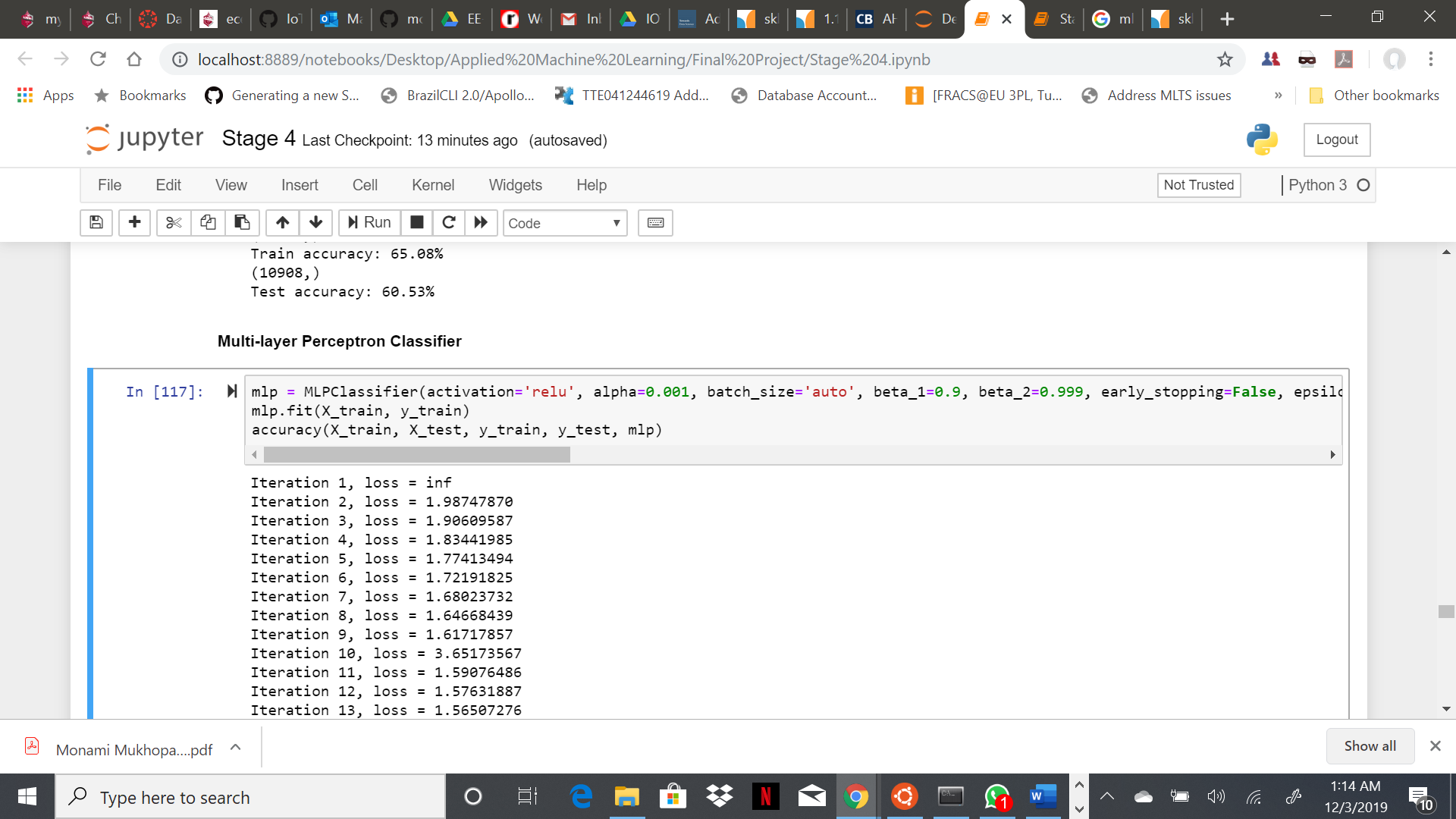
Training Accuracy – 65.08%

Testing Accuracy – 60.60%

**Algorithm 4 – Multi-layer Perceptron:**

MLPClassifier relies on an underlying Neural Network to perform the task of classification. A multilayer perceptron (MLP) is a feedforward artificial neural network model that maps sets of input data onto a set of appropriate outputs.

**We tried different values for alpha, max\_iter and solver to get the best accuracy results.**



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

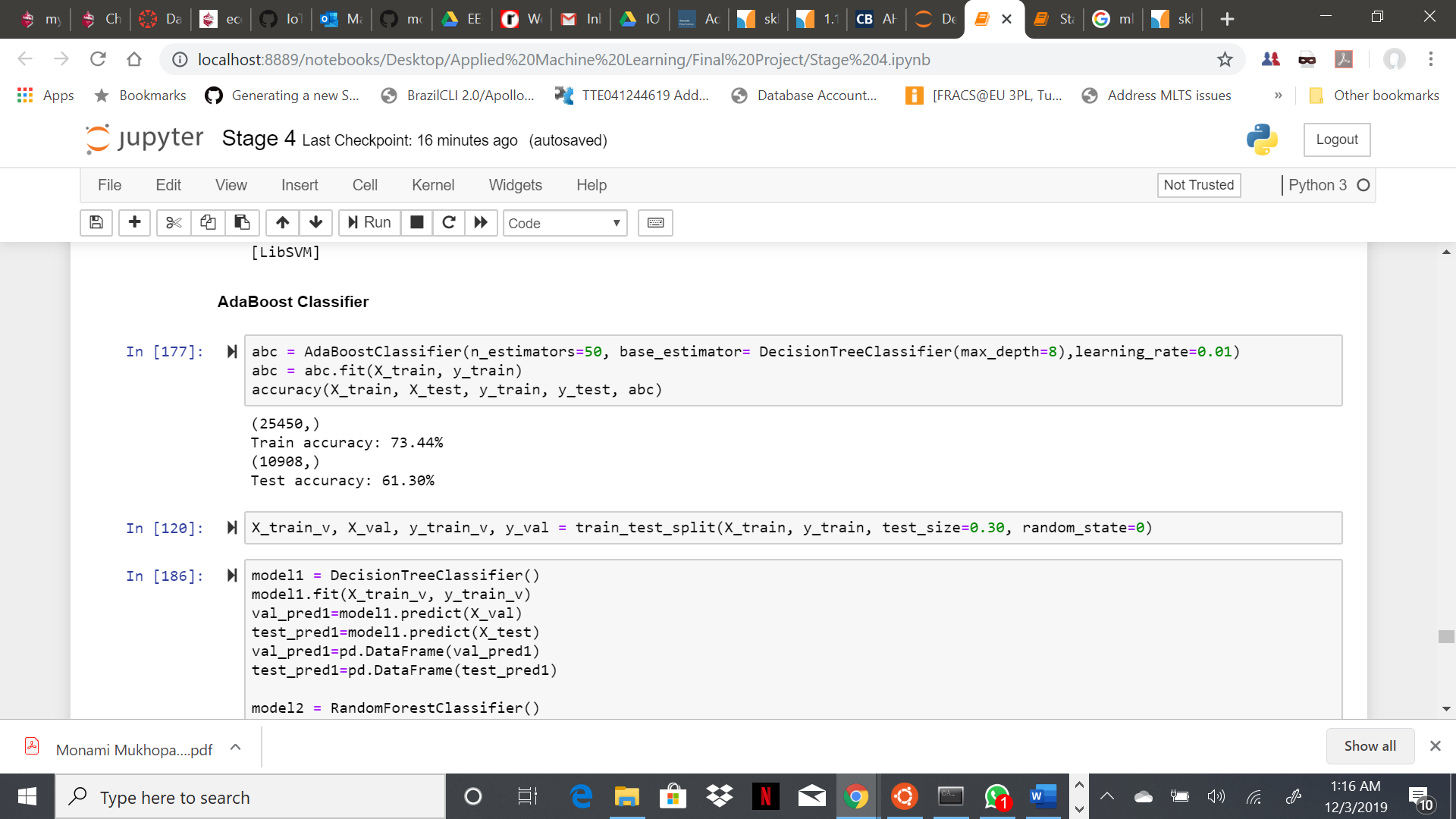
Training Accuracy – 37.65%

Testing Accuracy – 37.40%

**Algorithm 5 – AdaBoost:**

Ada-boost classifier combines weak classifier algorithm to form strong classifier. A single algorithm may classify the objects poorly. But if we combine multiple classifiers with selection of training set at every iteration and assigning right amount of weight in final voting, we can have good accuracy score for overall classifier.

For AdaBoostClassifier, we used n\_estimators=20,50,70,100,300,500,1000, max\_depth= 3,4,5,7,8,20,15,20, learning\_rate=0.1,0.3,0.5,0.01,0.08,0.003. Finally, we got below best accuracy values.



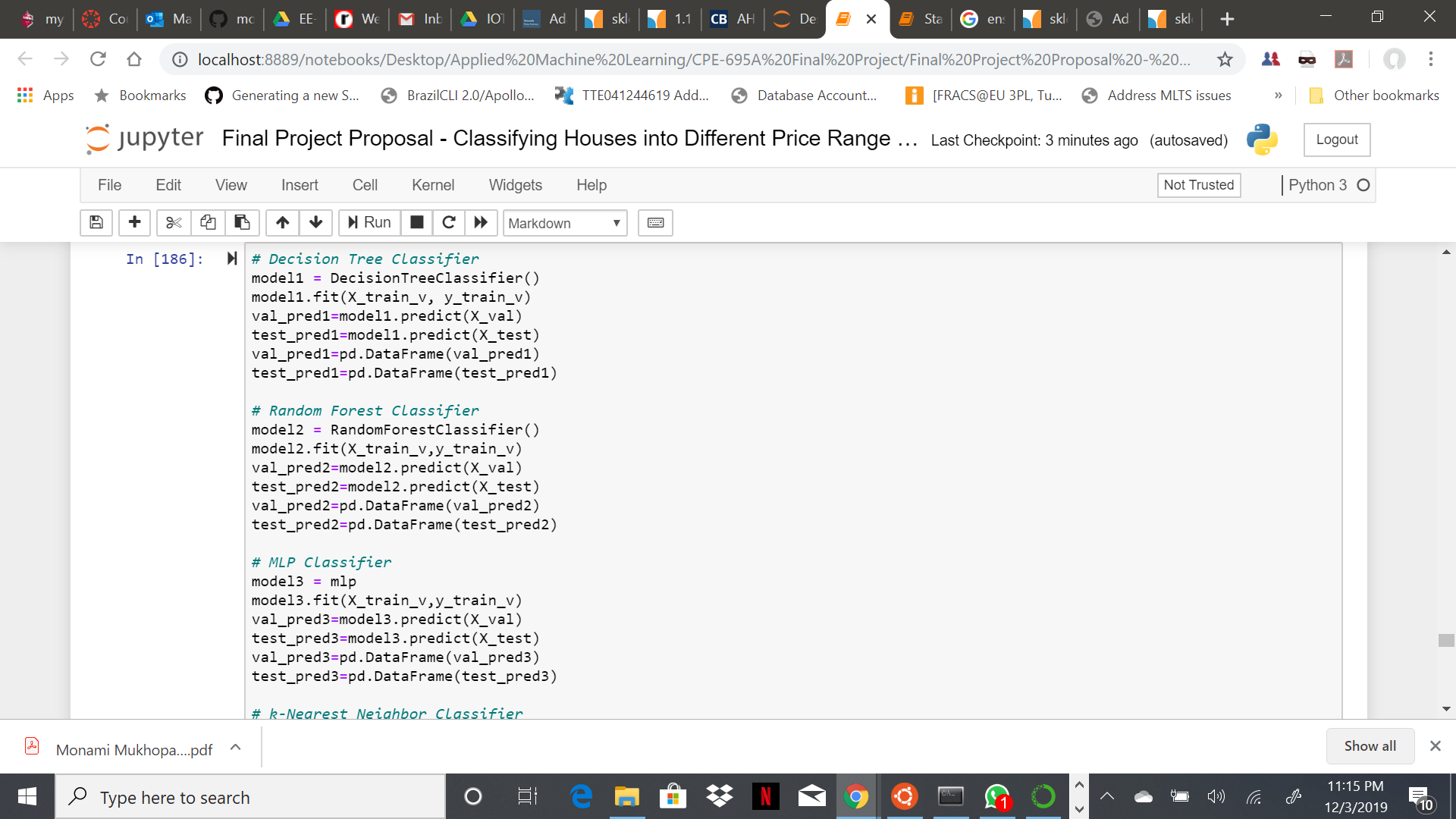
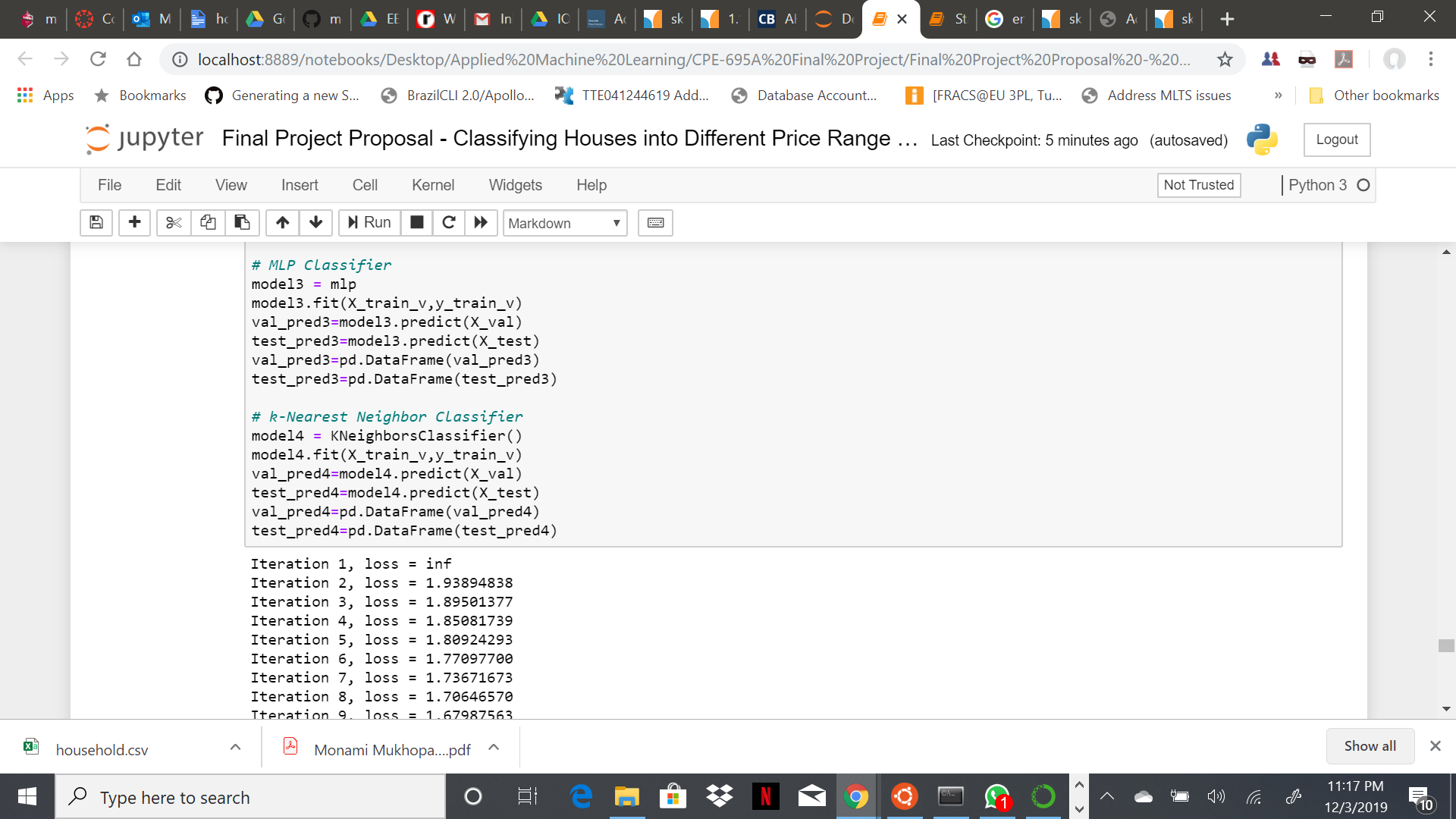
Results: With DecisionTreeClassifier, the accuracy score were as below:

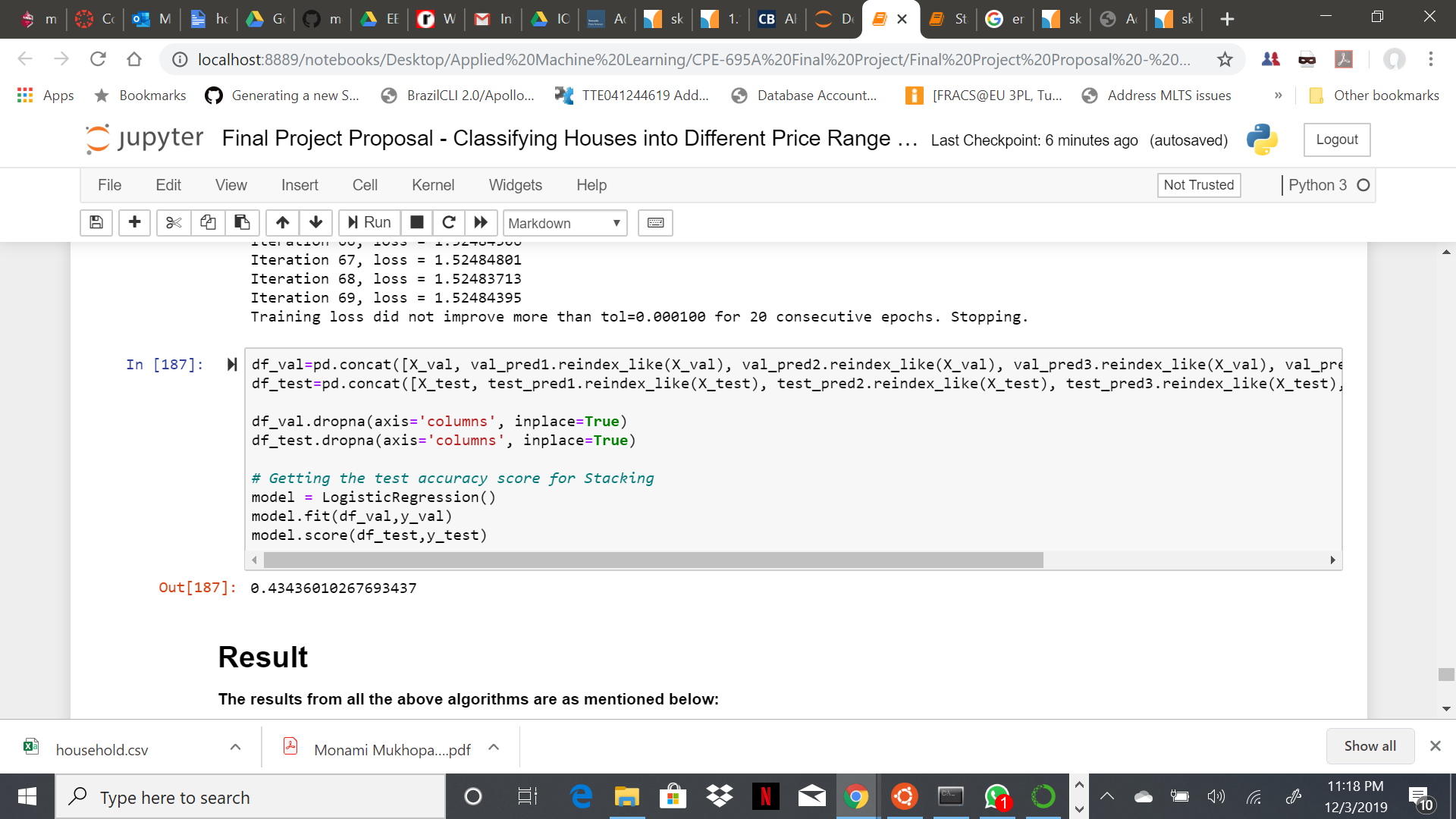
Training Accuracy – 73.54%

Testing Accuracy – 61.29%

**Algorithm 6 – Ensemble Method (Stacking):**

Stacking is an ensemble learning technique that combines multiple classification or regression models via a meta-classifier or a meta-regressor. The base level models are trained based on a complete training set, then the meta-model is trained on the outputs of the base level model as features.



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

Testing Accuracy Score – 0.43436010267693437

**Conclusions:**

A range of different models were explored, from relatively simple to more complex ones, with the goal of best predicting home price ranges but also utilizing different data science skills in the process. For this particular problem, the algorithm with best accuracy value is AdaBoost Classifier with test accuracy score of 61.29% and therefore it can be considered as a good classifier algorithm for house price range prediction problem. Also, the Decision Tree Classifier is close enough with 60.60% accuracy score. We have tried tuning each algorithm with different hyper-parameter values and finally kept the best results for each.

**Contribution:**

Each of the members had unique contributions towards the project. While three of us participated equally in Data Cleaning process, we worked on different ML algorithms as below.

* Darp Raithatha is responsible for preliminary research, gathering information, correlation determination, external materials to refer to, model evaluation and code testing. He worked on AdaBoost Classifier and Ensemble Method (Stacking) algorithms.
* Likith Nandigam is responsible for cleaning the dataset, compiling the code, code testing, result extraction, and documentation. He worked on Random Forest Classifier and k-Nearest Neighbor Classifier algorithms.
* Monami Mukhopadhyay is responsible for code testing, result extraction, project compilation, documentation, gathering and implementing references. She worked on Decision Tree Classifier and Multi-layer Perceptron Classifier algorithms.

**Code Execution Details:**

For our project, we applied four different machine learning algorithms and using a big dataset (around 161 MB). Therefore, all the relevant documents presentation slides, project report and source code .ipynb file is present in google drive.

Google Drive-

[***https://rb.gy/jdzjde***](https://rb.gy/jdzjde)

Github Repo –

[***https://github.com/darpraithatha/House-Price-Range-Prediction***](https://github.com/darpraithatha/House-Price-Range-Prediction)

**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).