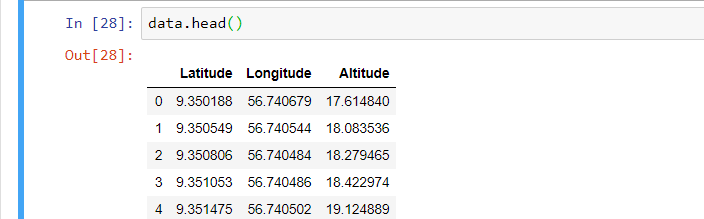
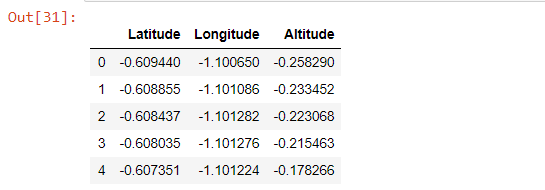
**ASSIGNMENT 2**

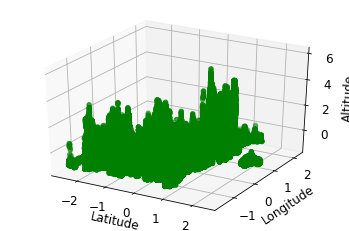
## Data Description:

The data set used in our analysis consists of 3 columns, which are as follows:  
1. Latitude(X1)  
2. Longitude(X2)   
3. Altitude(Y)

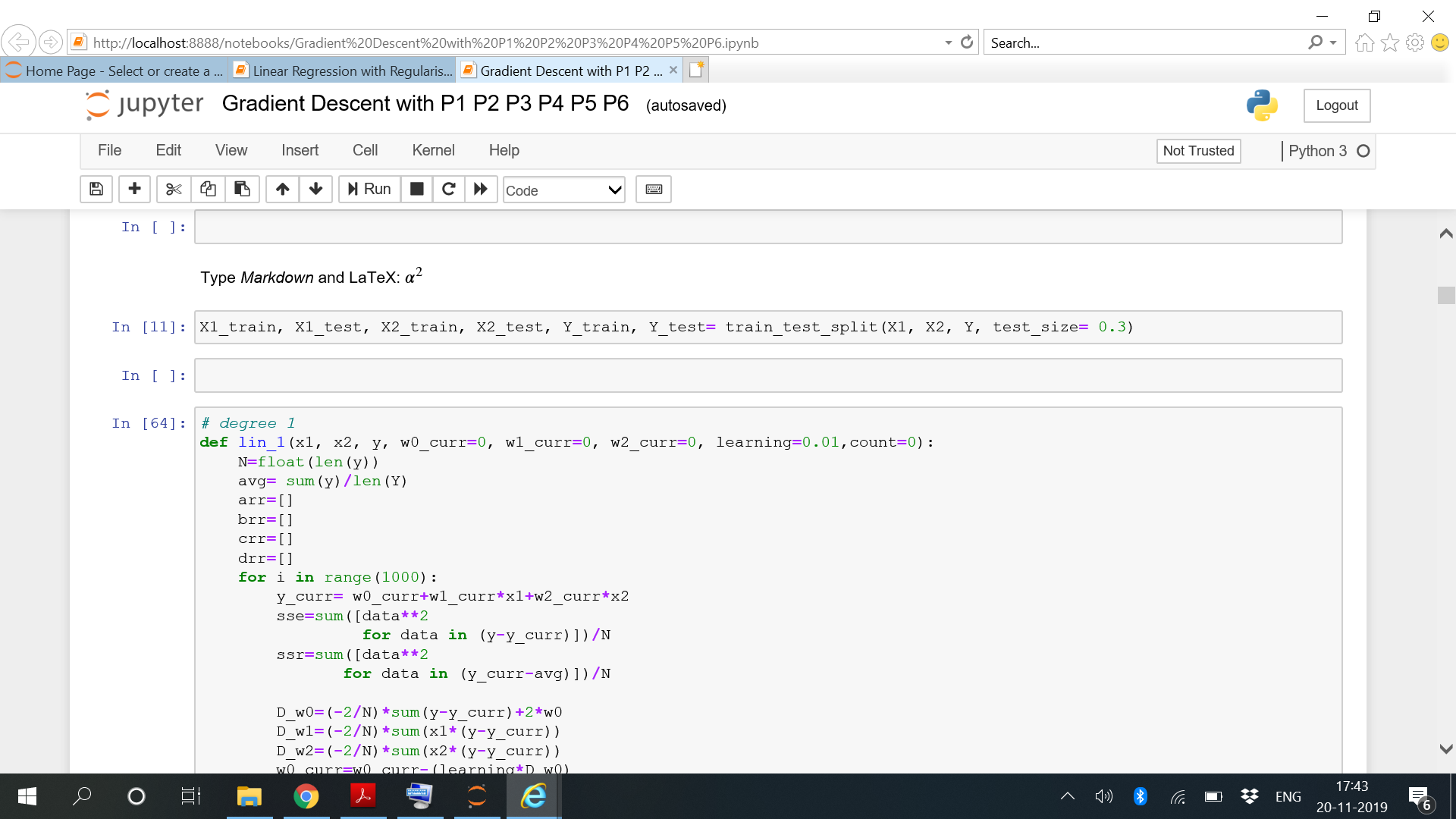


The size of our dataset is **1044628**  points. For more accurate results, we have **normalized** our data set. After normalization we got:

The scatter plot of the data is :



After the above-mentioned step,the dataset was divided into 2 sets: Training Dataset, and Testing Dataset, with 30% of the data for testing purposes and 70% for training purposes.



**Part A: Using Gradient Descent**

# Gradient Descent Method

Gradient descent is an optimization algorithm used to minimize some function by iteratively moving in the direction of steepest descent as defined by the negative of the gradient.

Math

Given the cost function:

f(m,b)=1N∑i=1n(Yi−(mxi+b))2

The gradient can be calculated as:

f′(m,b)=⎡⎣df/dm/df/db⎤⎦=[1N∑−2xi(Yi−(mxi+b))1N∑−2(Yi−(mxi+b))]

To solve for the gradient, we iterate through our data points using our new m and b values and compute the partial derivatives. This new gradient tells us the slope of our cost function at our current position (current parameter values) and the direction we should move to update our parameters. The size of our update is controlled by the learning rate.

Math

Given the cost function:

f(m,b)=1N∑i=1n(Yi−(mxi+b))^2

The gradient can be calculated as:

f′(m,b)=⎡⎣df/dm/df/db⎤⎦=[1N∑−2xi(Yi−(mxi+b))1N∑−2(Yi−(mxi+b))]

To solve for the gradient, we iterate through our data points using our new m and b values and compute the partial derivatives. This new gradient tells us the slope of our cost function at our current position (current parameter values) and the direction we should move to update our parameters. The size of our update is controlled by the learning rate.

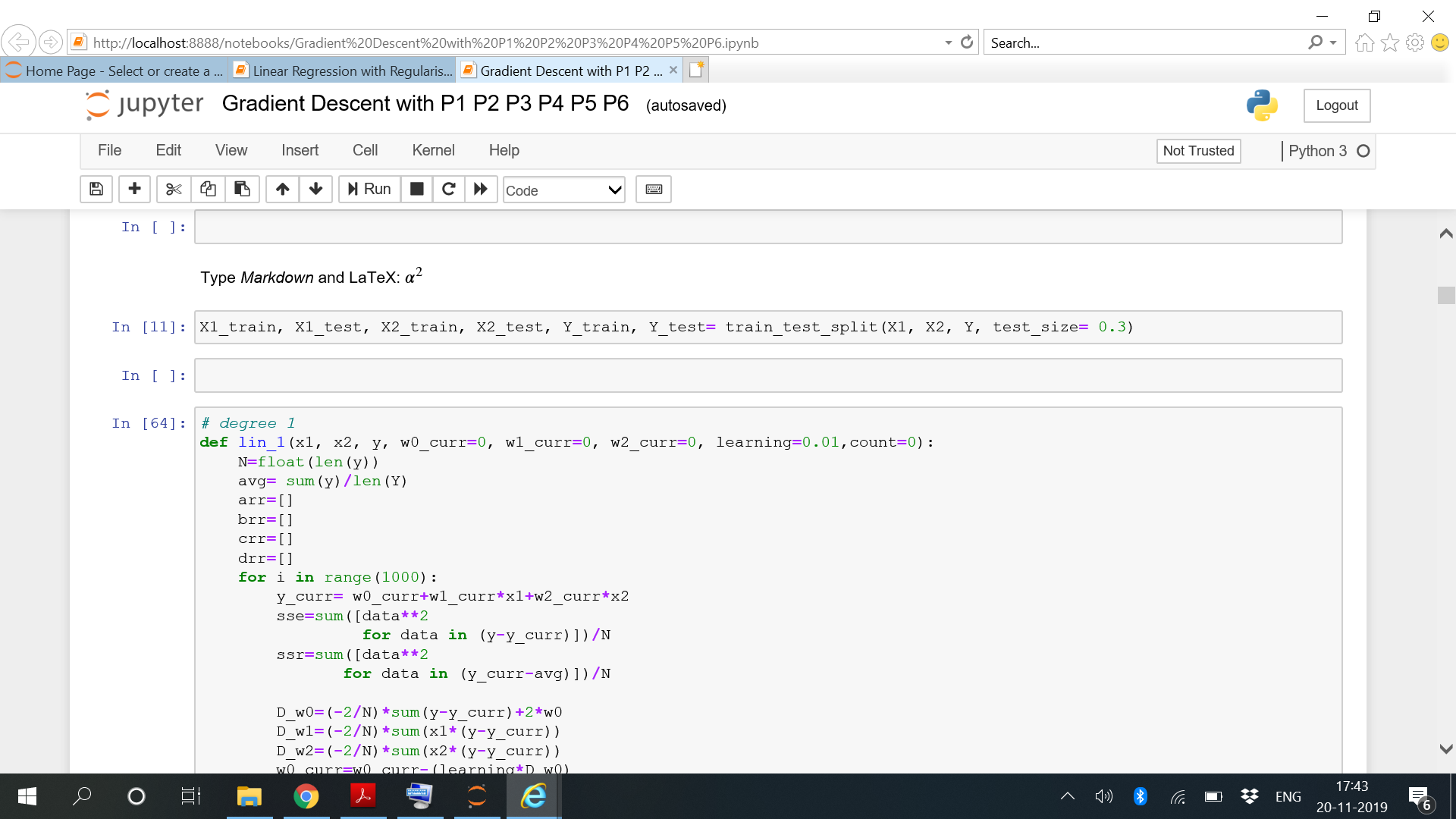
**Degree 2:**

**We have to estimate 5 weights here:**

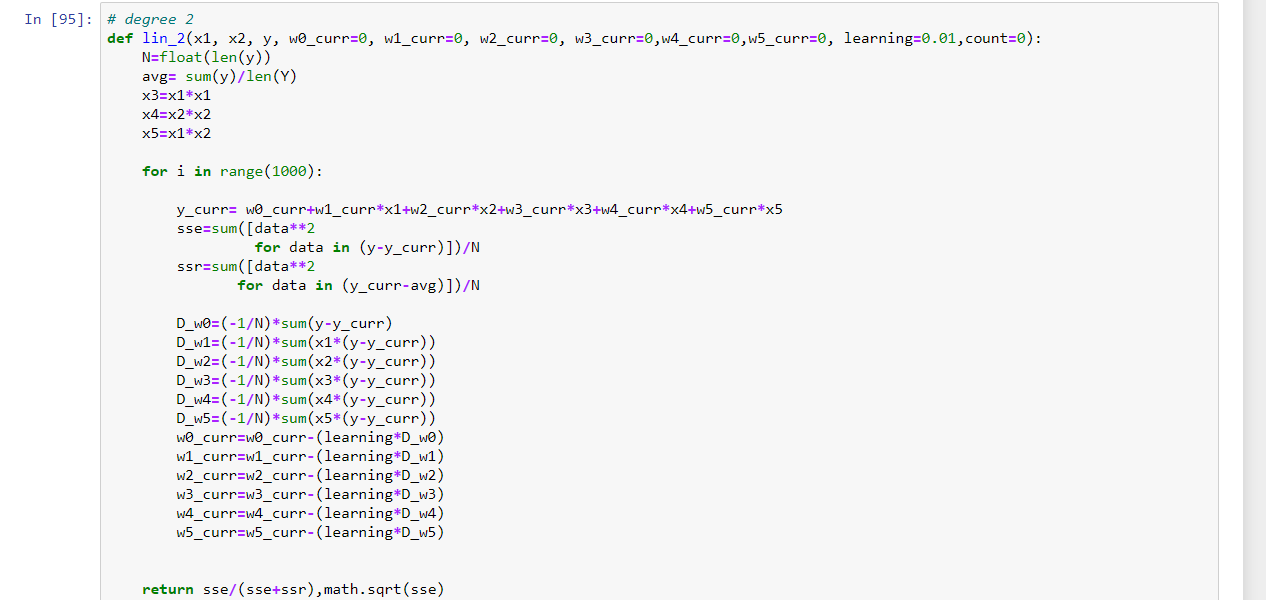
Train: Test Ratio =70:30

**Iterations=1000**

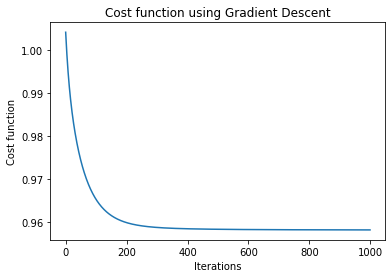
**Learning rate=0.01**



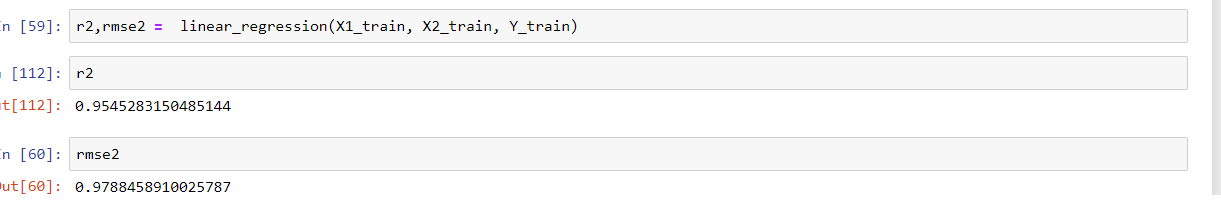
Code for generating the coefficients in degree 2 is:



The cost Function varies with iterations as:



The **results** for Training are:



**The R square for degree 2 is 0.955**

**RMSE for degree 2 is0.978**

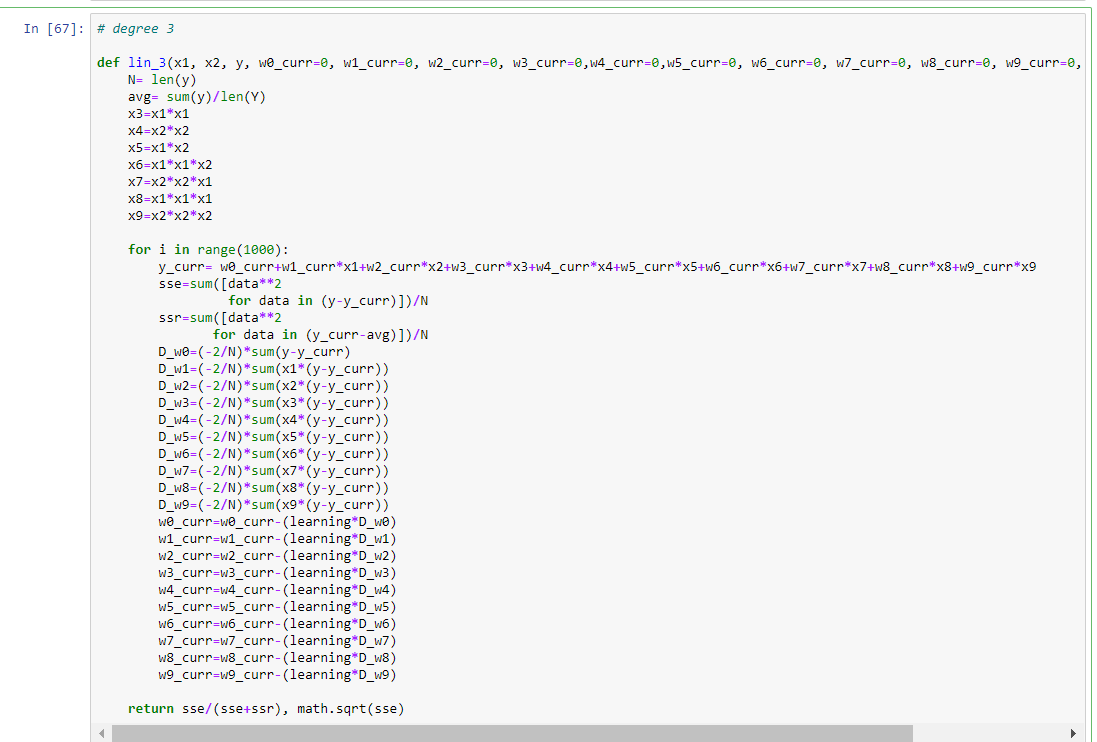
* **Degree 3:**

We have to estimate **9 weights** here:

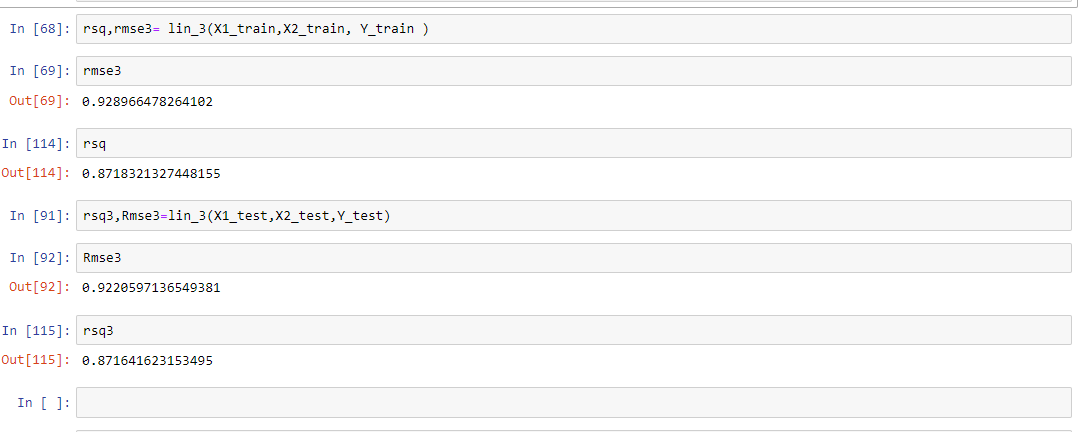
**Iterations=1000**

**Leaning Rate=0.01**

The logic is shown below:



**The results for training and testing data are:**

****

**R square for degree 3 is 0.8718**

**RMSE for degree 3 is 0.9289**

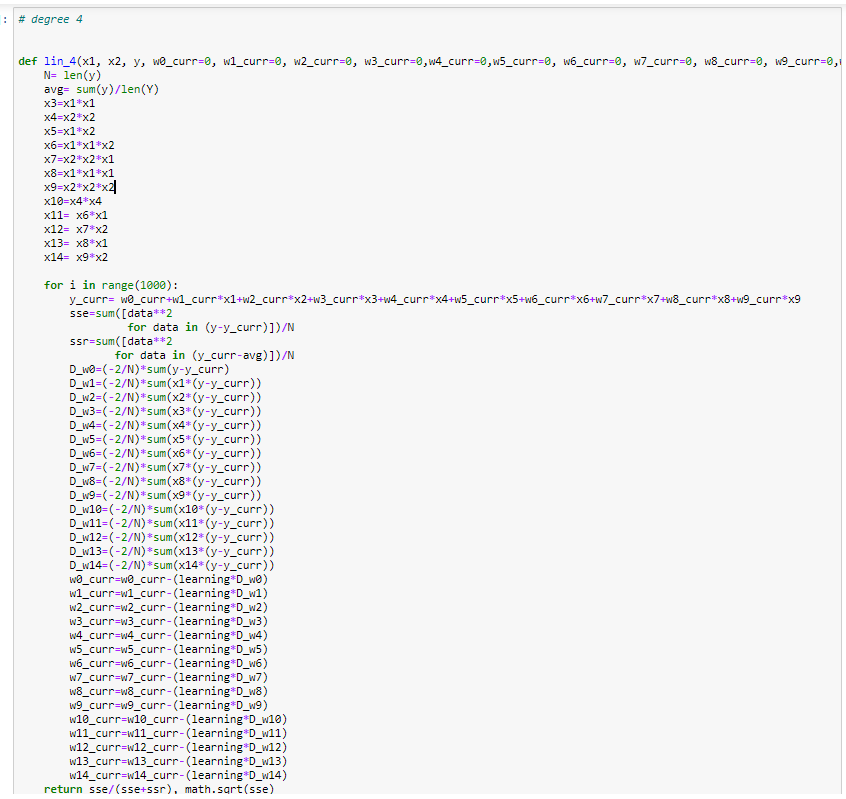
**Degree 4:**

Here we have to estimate  **14 weights.**

**Iterations=1000**

**Learning Rate=0.01**

The logic is shown below



**The results for training and testing are:**

**R square for degree 4 = 0.8375**

**RMSE for degree 4 = 0.9033**

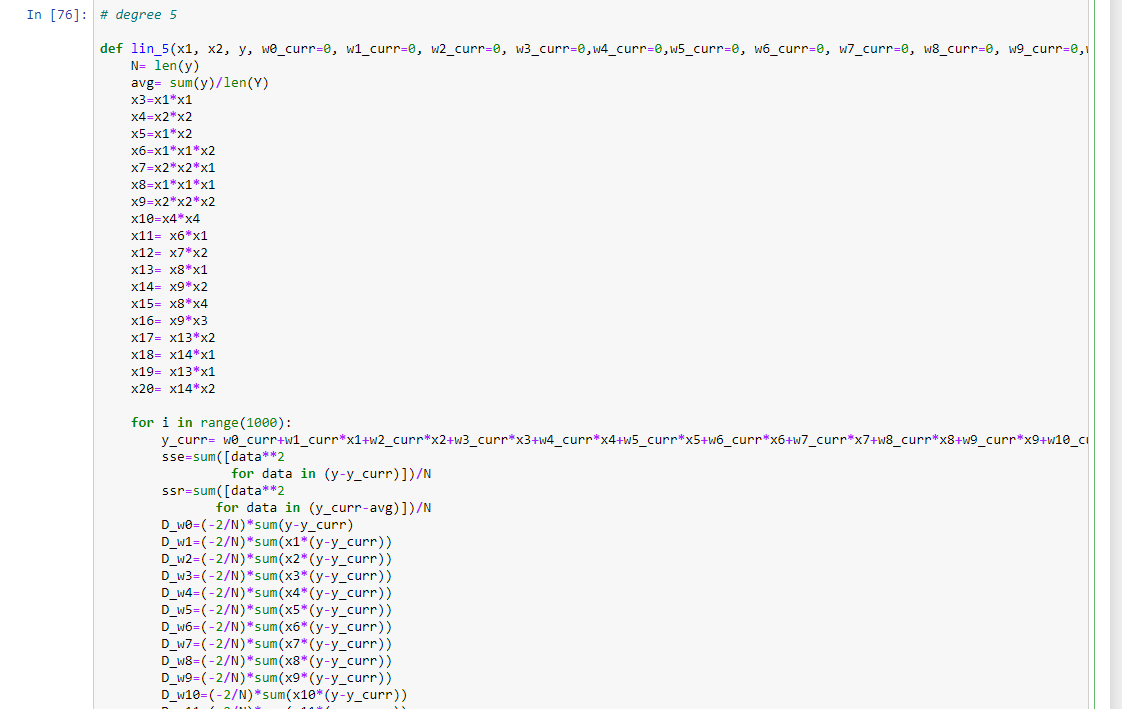
**Degree 5**

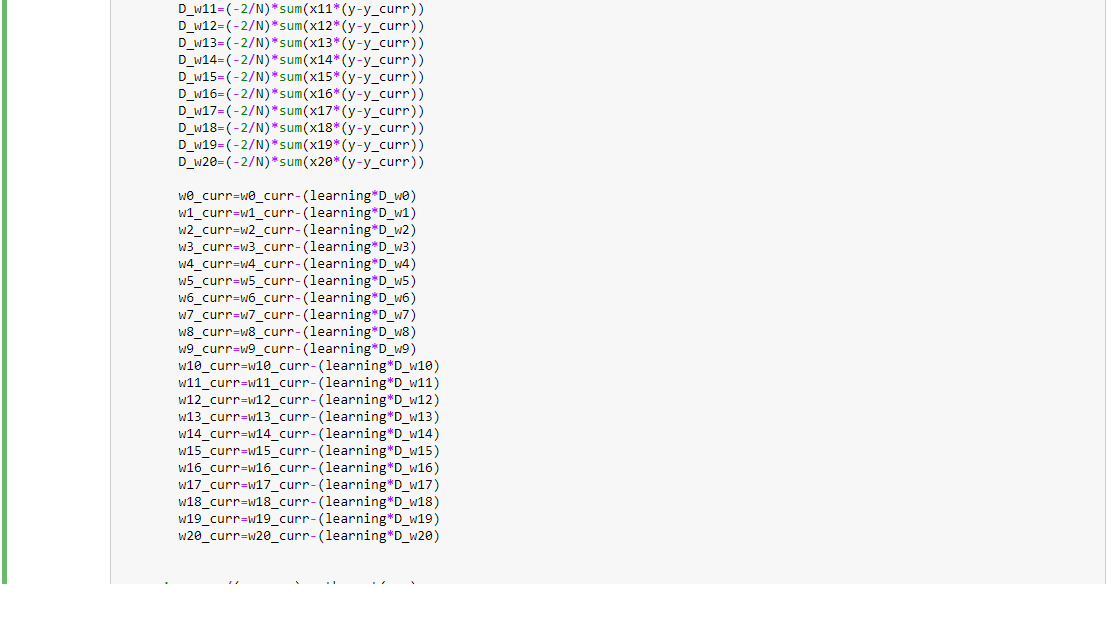
We have to estimate **20 weights** here.

**No of iterations=1000.**

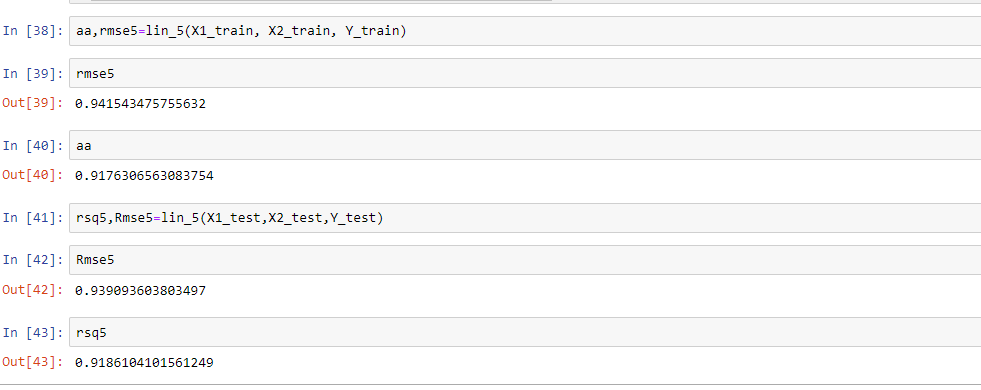
**Learning Rate=0.01**

The logic is shown below:





The results for **training and testing** are:



**R square for degree 5 = 0.9176**

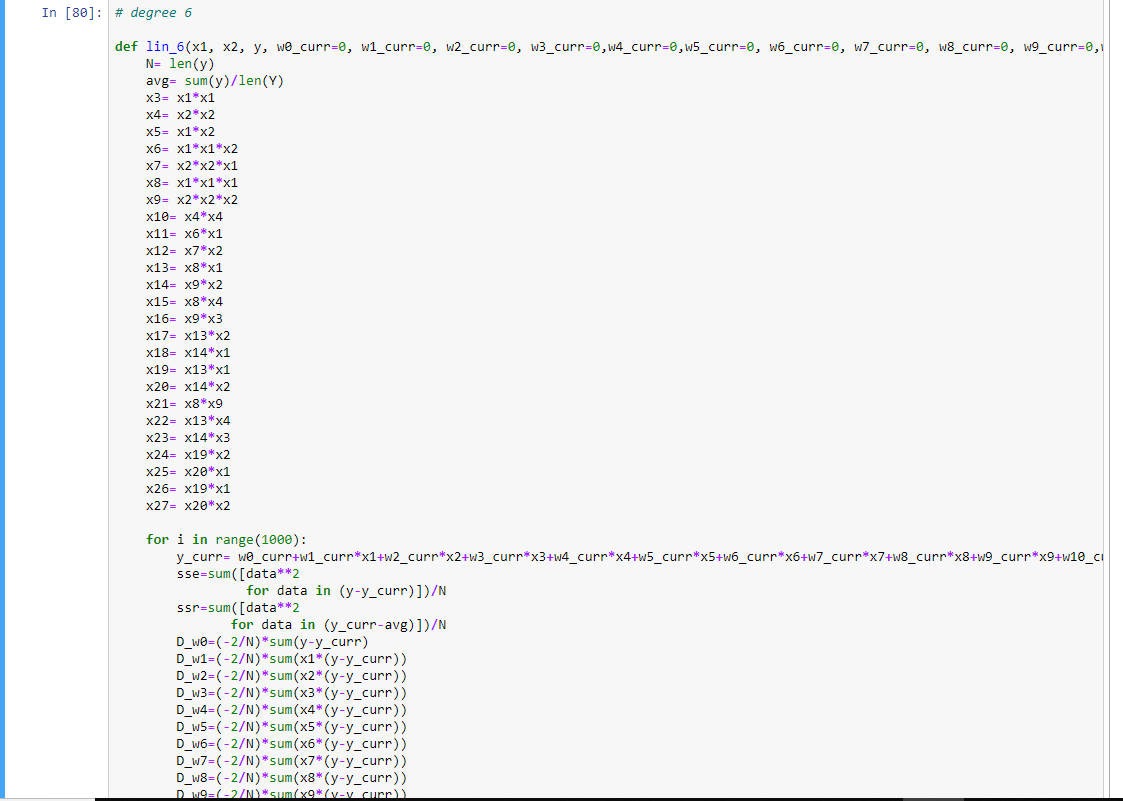
**RMSE for degree 5 = 0.9415**

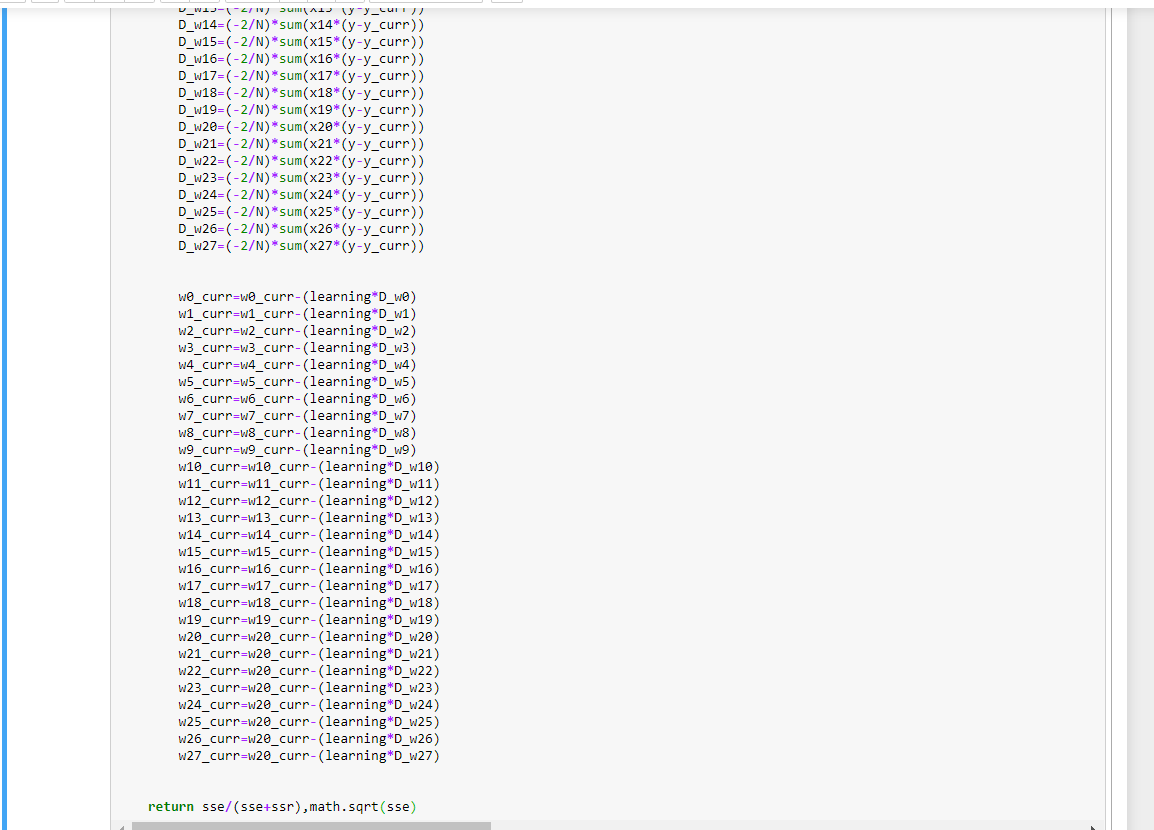
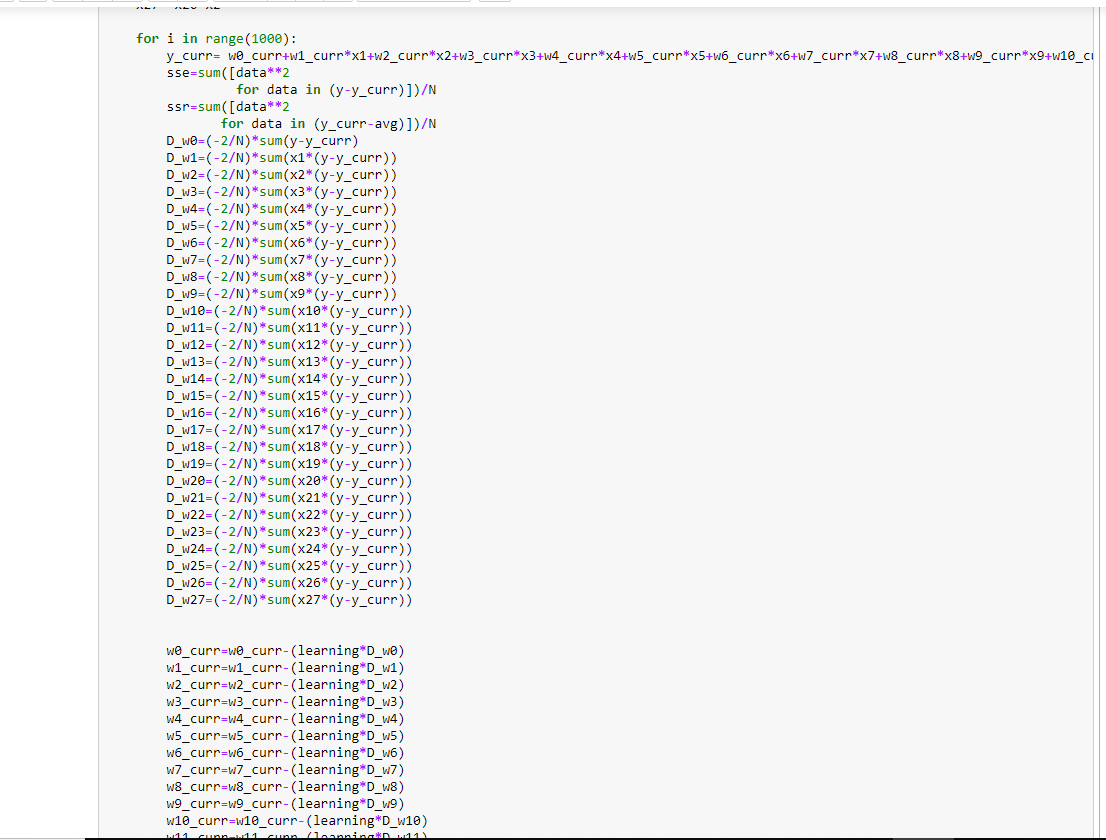
**Degree 6:**

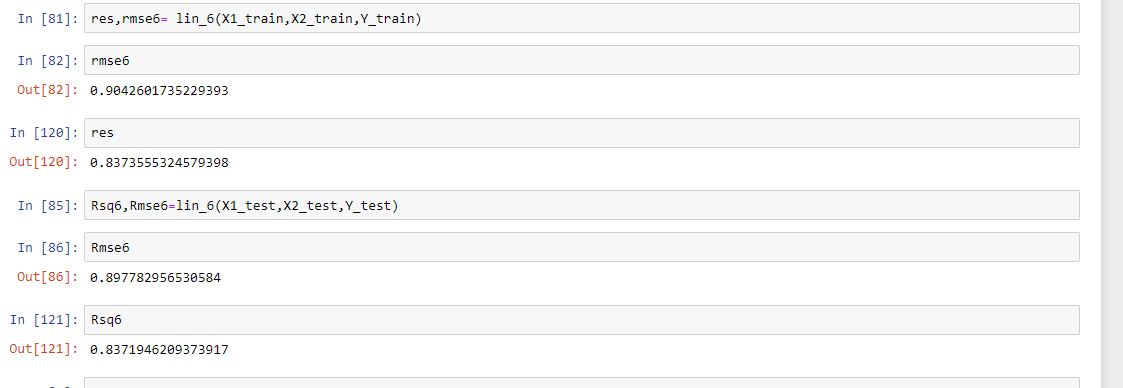
Here we have to estimate **27 coefficients**

**Iterations= 1000**

**Learning rate = 0.0001**





The results for **training and testing error** for degree 6 are:

**R square for degree 6:**

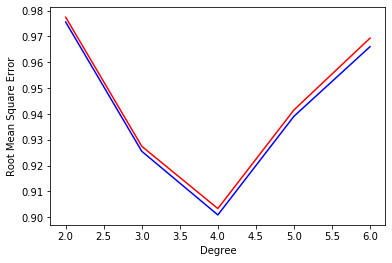
**RMSE for degree 6 :**

**Comparative Study:**

|  |  |  |
| --- | --- | --- |
| **Degree** | **R square** | **RMSE** |
| **2** | **0.955** | **0.978** |
| **3** | **0.8718** | **0.9289** |
| **4** | **0.8375** | **0.9033** |
| **5** | **0.9176** | **0.9415** |
| **6** | **0.8974** | **0.969** |

**Plotting RMSE vs degree for (degree=2,3,4,5,6)**

**(red= training, blue=testing)**



**Conclusion**

The plot clearly suggests that degree 4 is the best model as RMSE is the least.

Overfitting is not observed in the data up to degree 6.

Gradient Descent Method with Regularisation  
  
The above methodology follows similar principle of optimisation as that of gradient descent. The only place where a fundamental difference arrives is in the development of Cost Function, and then subsequently, the Step Size. There are 2 types of regularisation:

1.L1 Regularisation (Lasso Regression)

2. L2 Regularisation (Ridge Regression)

In this case, it was specified to fit a polynomial of degree 6 to the dataset given, which was to be optimised by gradient descent. The results are as follows:

Ridge Regression

After fitting the function on the training datapoints, we obtain the following parameters for accuracy:  
  
R2=0.9227738082826966

RMSE=0.9763918526698111

The coefficients of the training model are as follows:  
W0=0.004422348206681807

W1=0.008861785982315911

W2=-0.0013396265544481942

W3=0.0005232901968194099

W4=0.01544400915946711

W5=0.005664238755928369

W6=0.004044317247380639

W7=0.004719682046403276

W8=0.0034027689309101545

W9= -0.0064368213703902

W10=0.024450533006485687

W11=0.012932563574528355

W12=0.007998069672042057

W13=0.013302804759043226

W14=0.024450533006485684

W15=-0.008341948457650208

W16=-0.00043366071153623915

W17=-0.011733503591989396

W18=0.005123538762527979

W19=-0.012208122719078021

W20=-0.008506424354734258

W21=-0.00842565319372756

W22=-0.008402923403926918

W23=-0.008441229154490002

W24=-0.008347812431455215

W25=-0.008433053509765854

W26=-0.008273446211002348

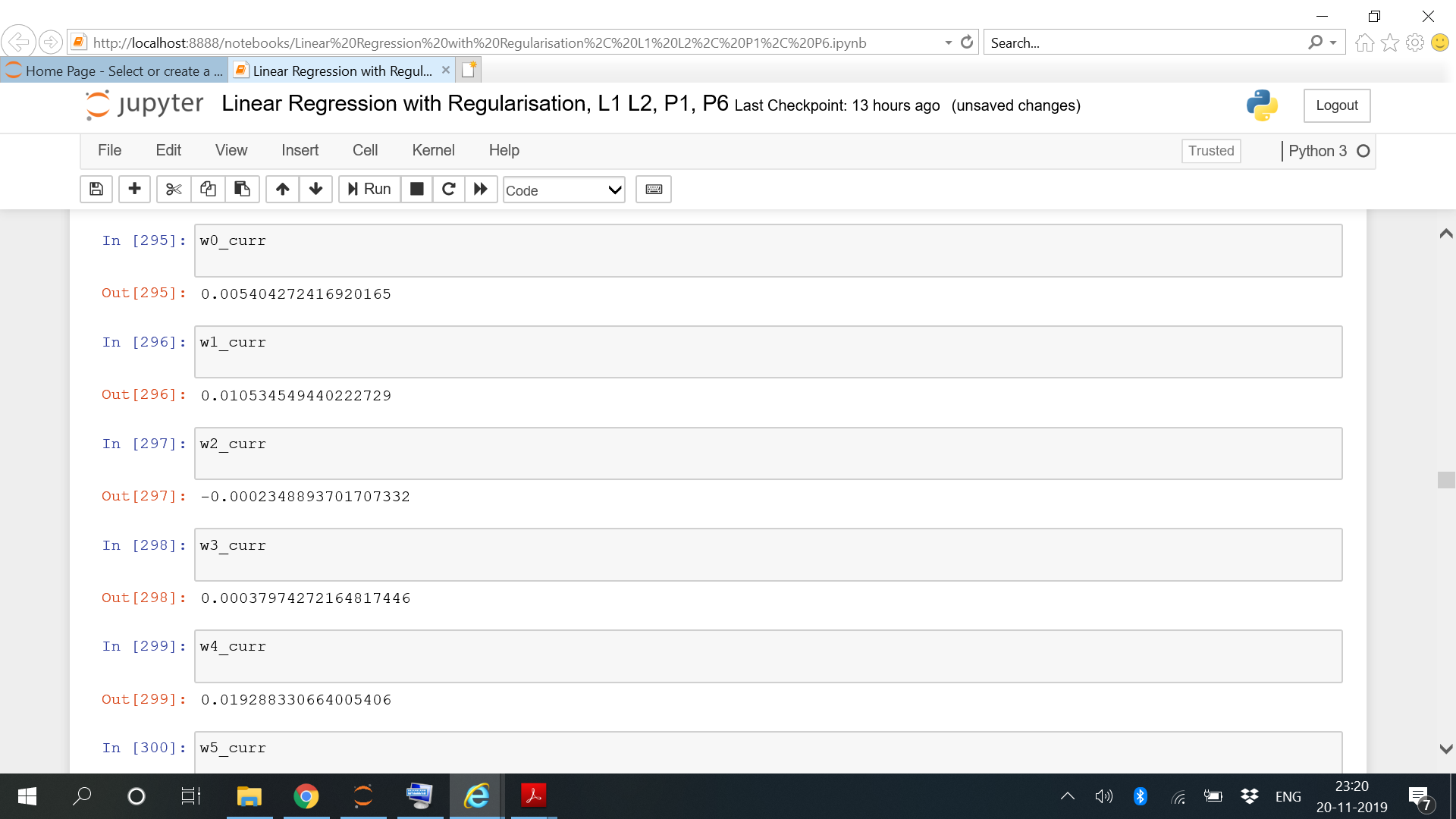
W27=-0.008367259723064477

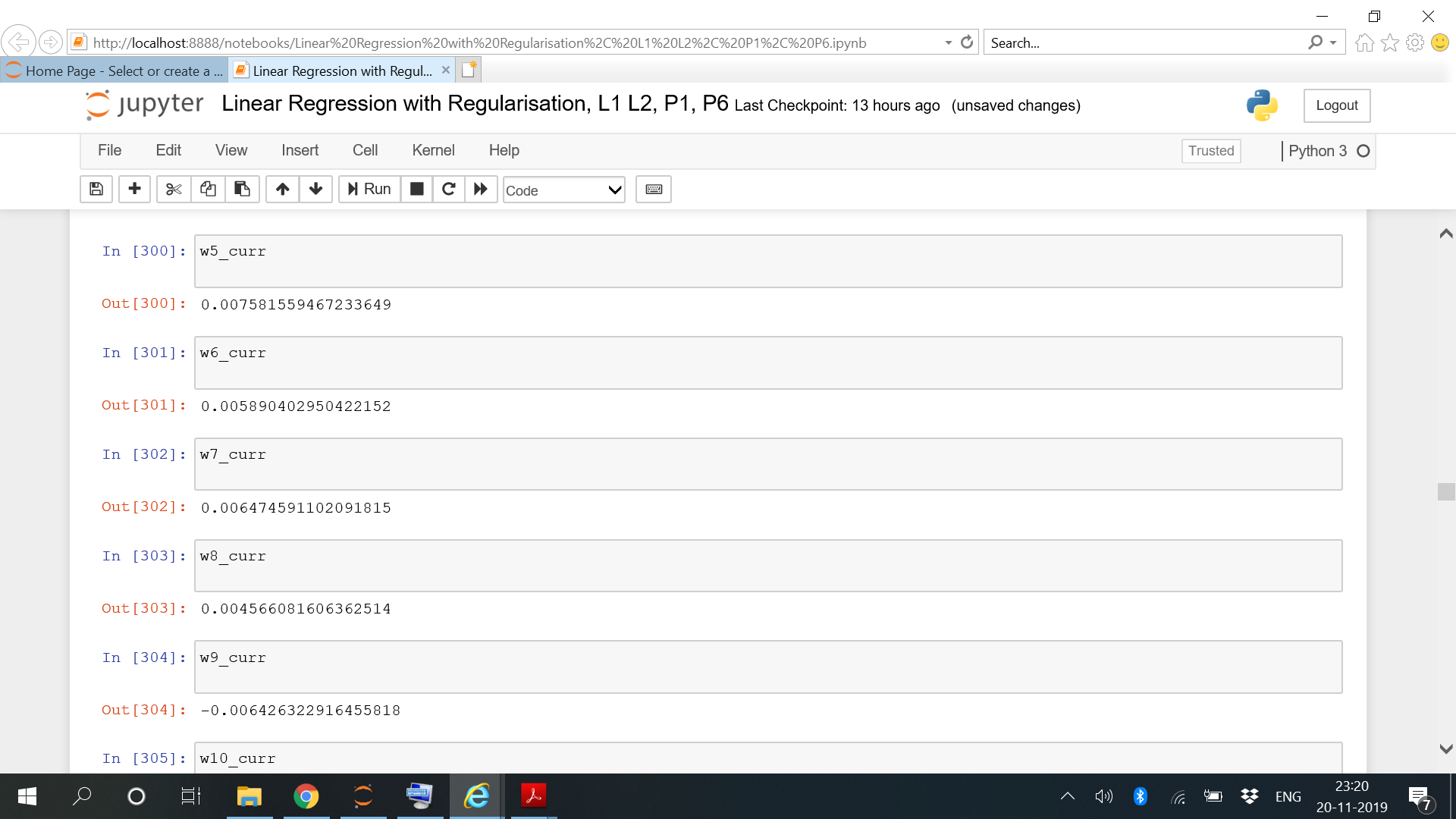
Now, after testing, we obtain the following:

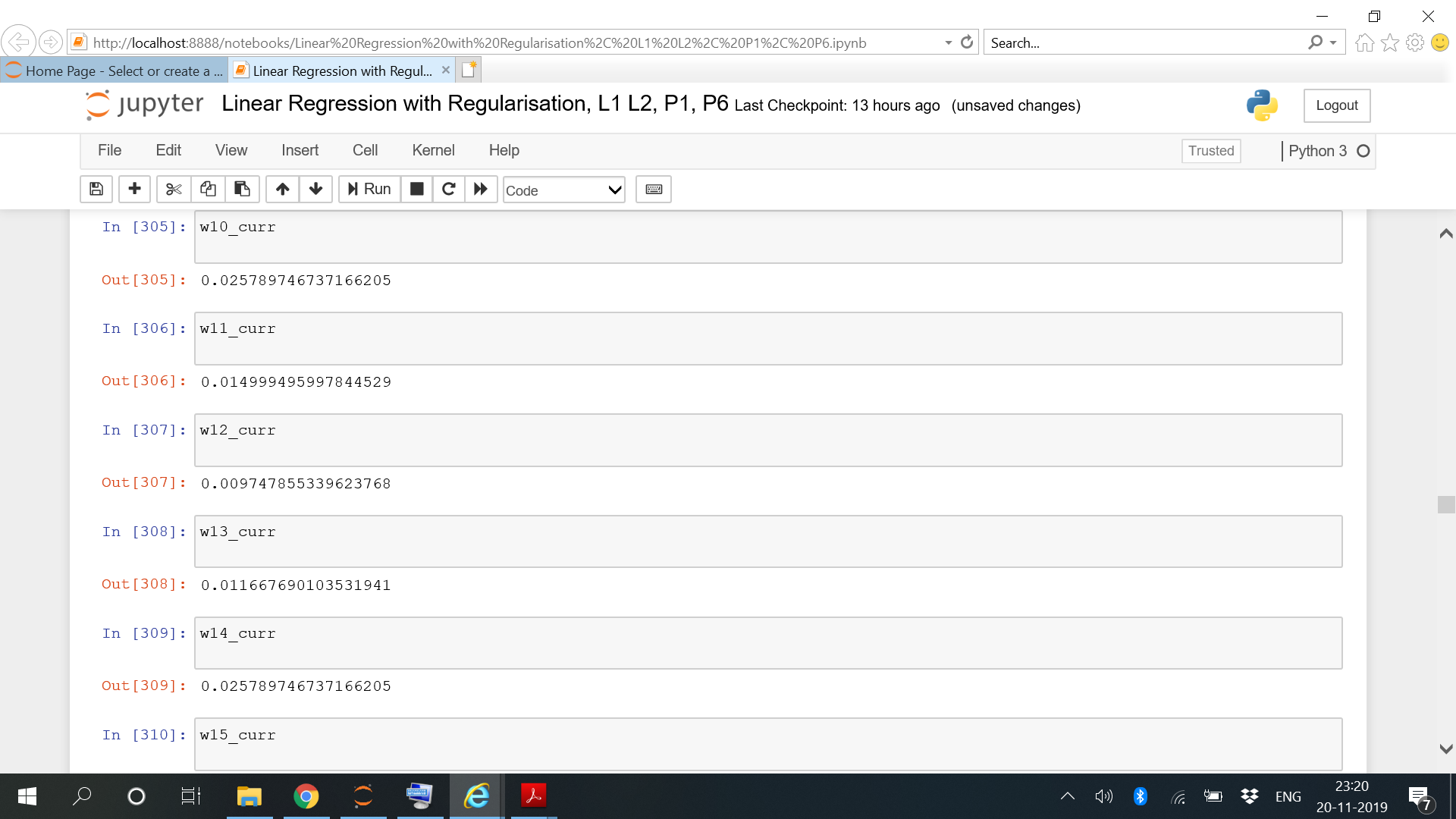
R2=0.9324791398080852

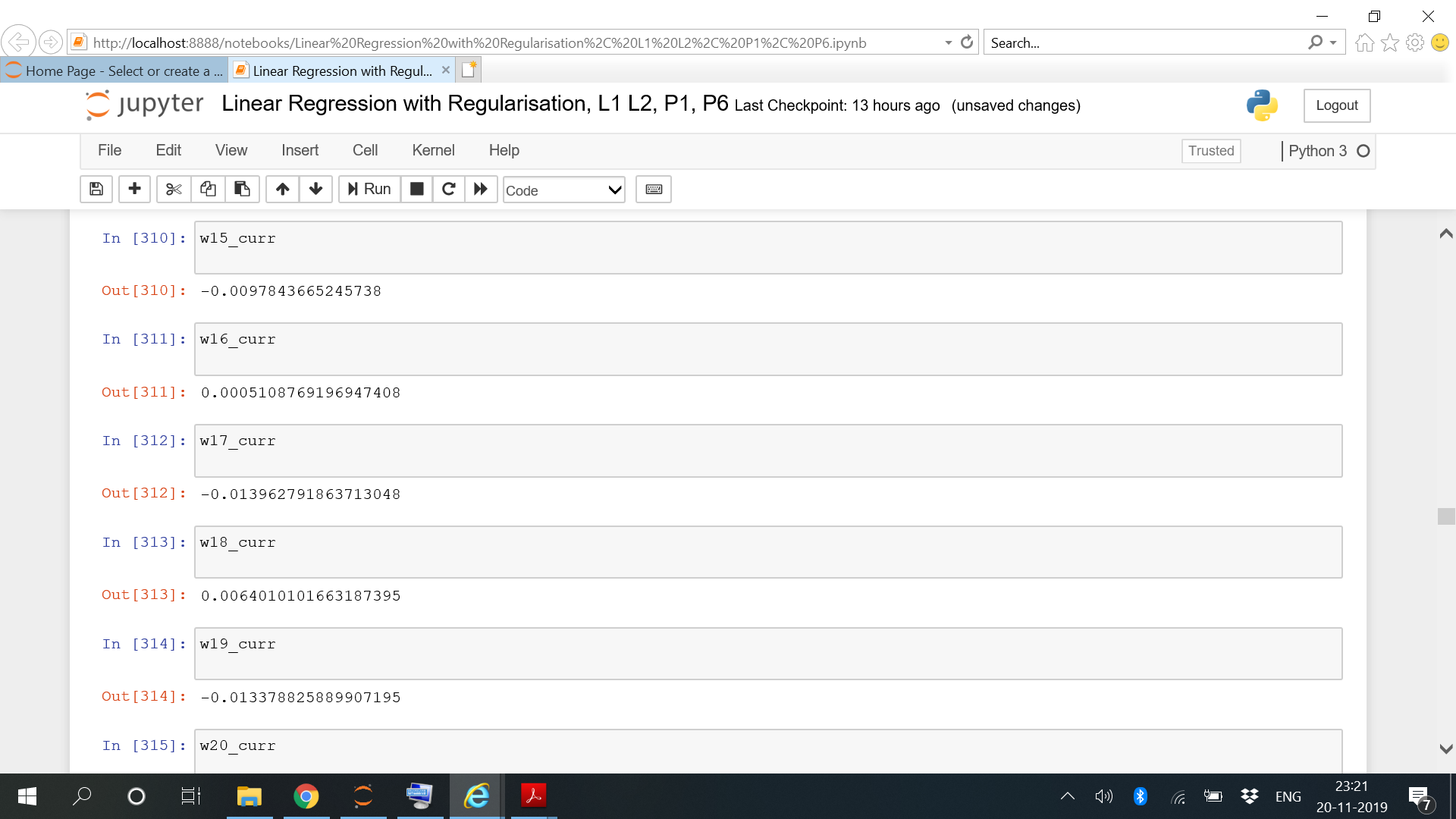
RMSE=0.9962679419847585

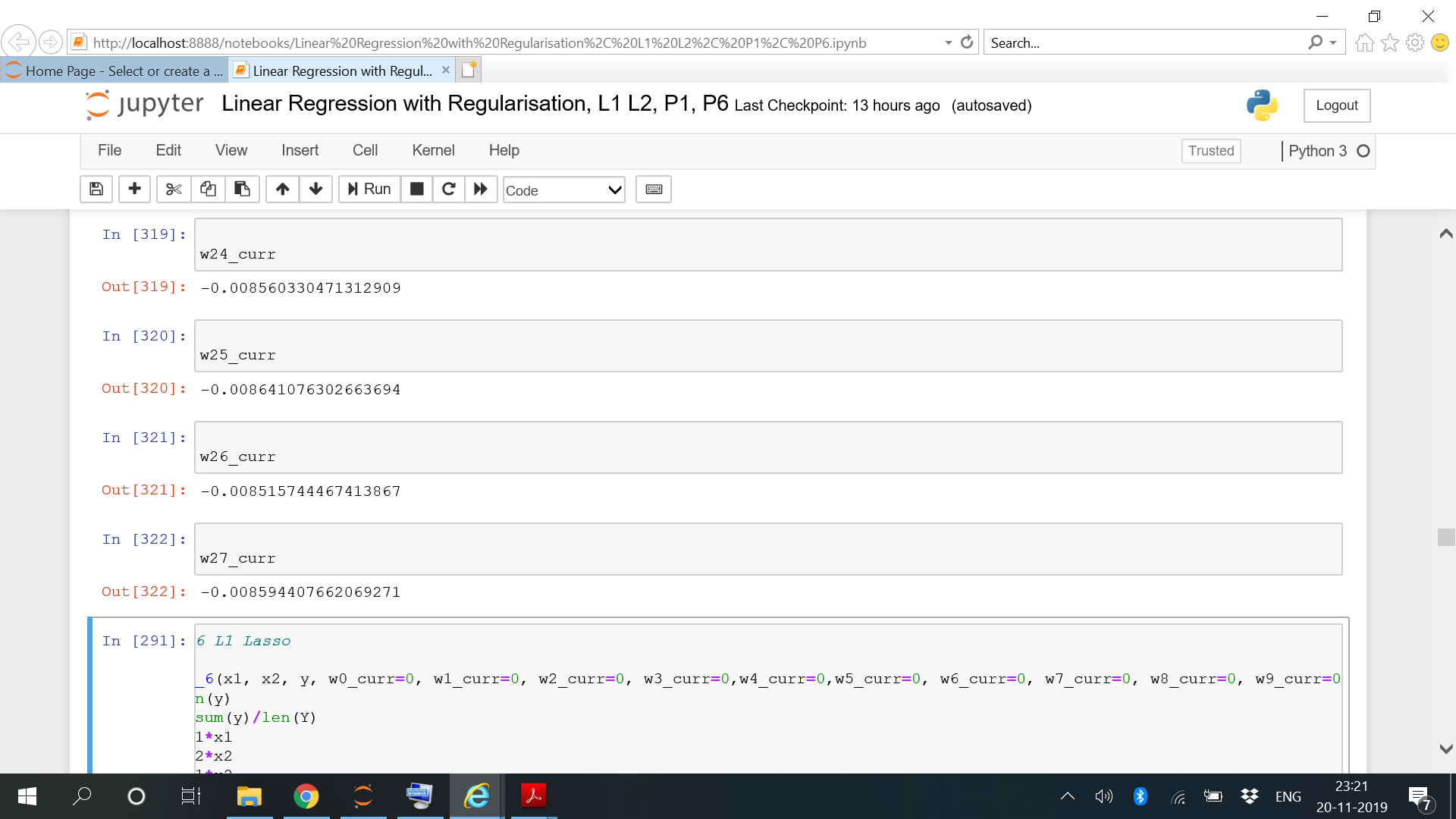
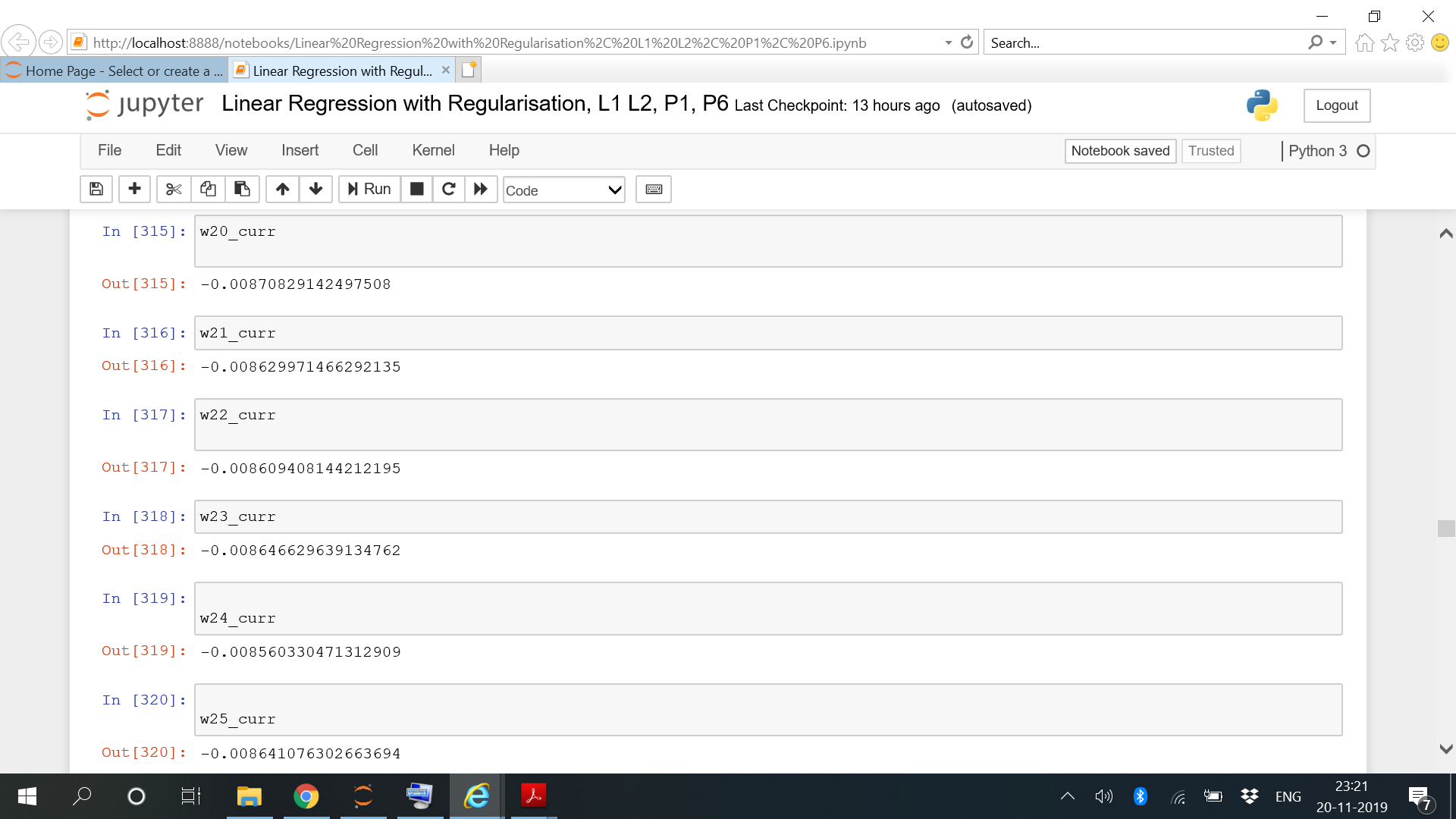
Final Coefficients are as follows:











Lasso Regression

R2=0.5091662244950589

RMSE=8.666801672368104

W0=0.2681308684436814

W1=-1.4958242865261162

W2=-1.3732711707677232

W3=-1.3732711707677232

W4=-1.1562716688695138

W5=-1.4637843804465744

W6=-1.1421949146399288

W7=-1.037544326206513

W8=-1.33891466317565

W9=-0.8182911181359552

W10=-0.6105613877813417

W11=-2.0072274425686056

W12=-1.4543965756648862

W13=-1.6360766206106374

W14=-0.6105613877813417

W15=0.08509192210291763

W16=-0.1844653775204185

W17=0.4939030808582798

W18-0.1169161335149536

W19=0.39164138043079355

W20=0.36076433104615907

W21=0.35646213015927186

W22=0.3553389940718674

W23=0.35736017122670116

W24=0.3543299800242629

W25=0.35811184836734966

W26=0.3559634981220863

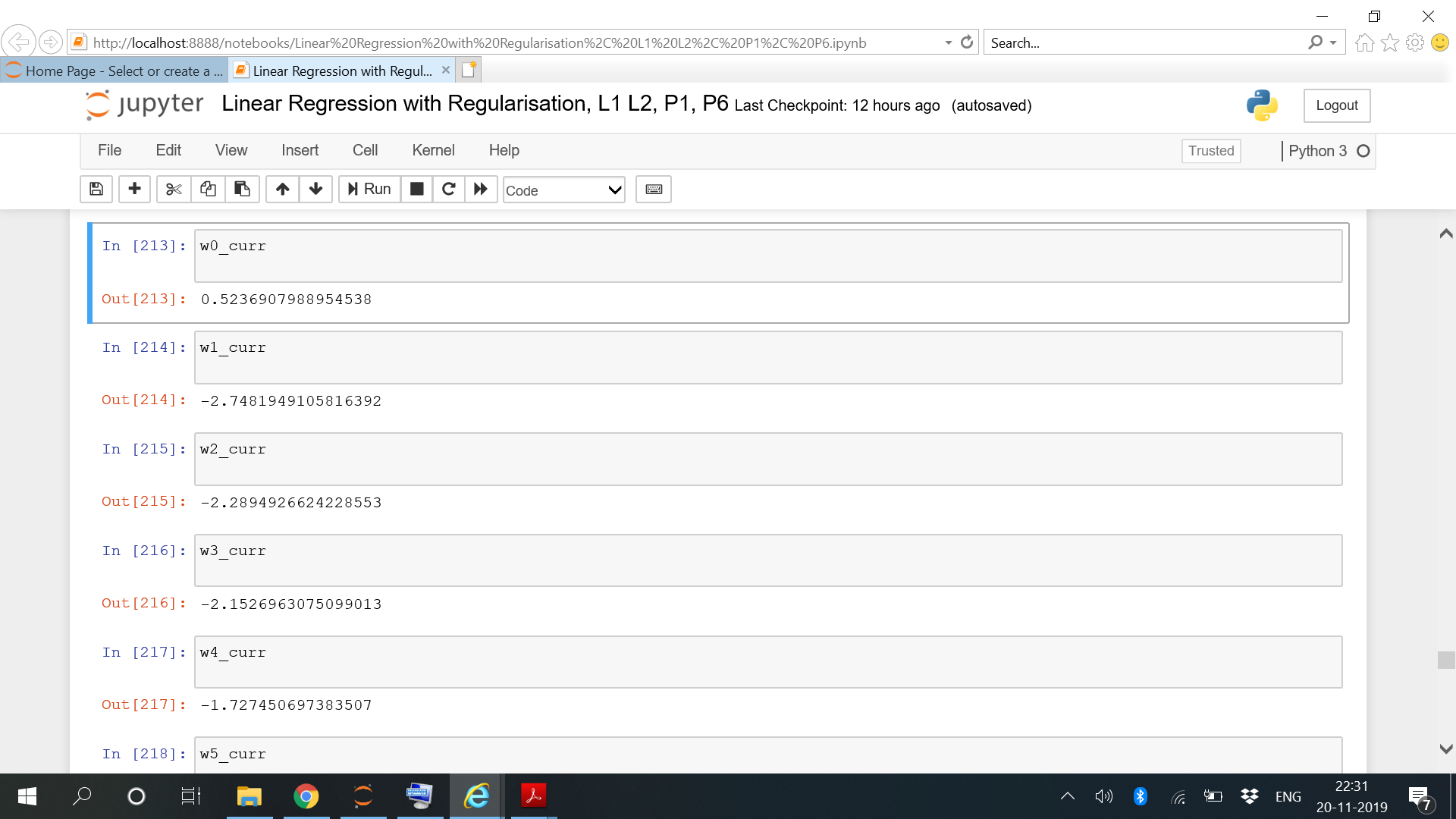
W27=0.3600372624810498

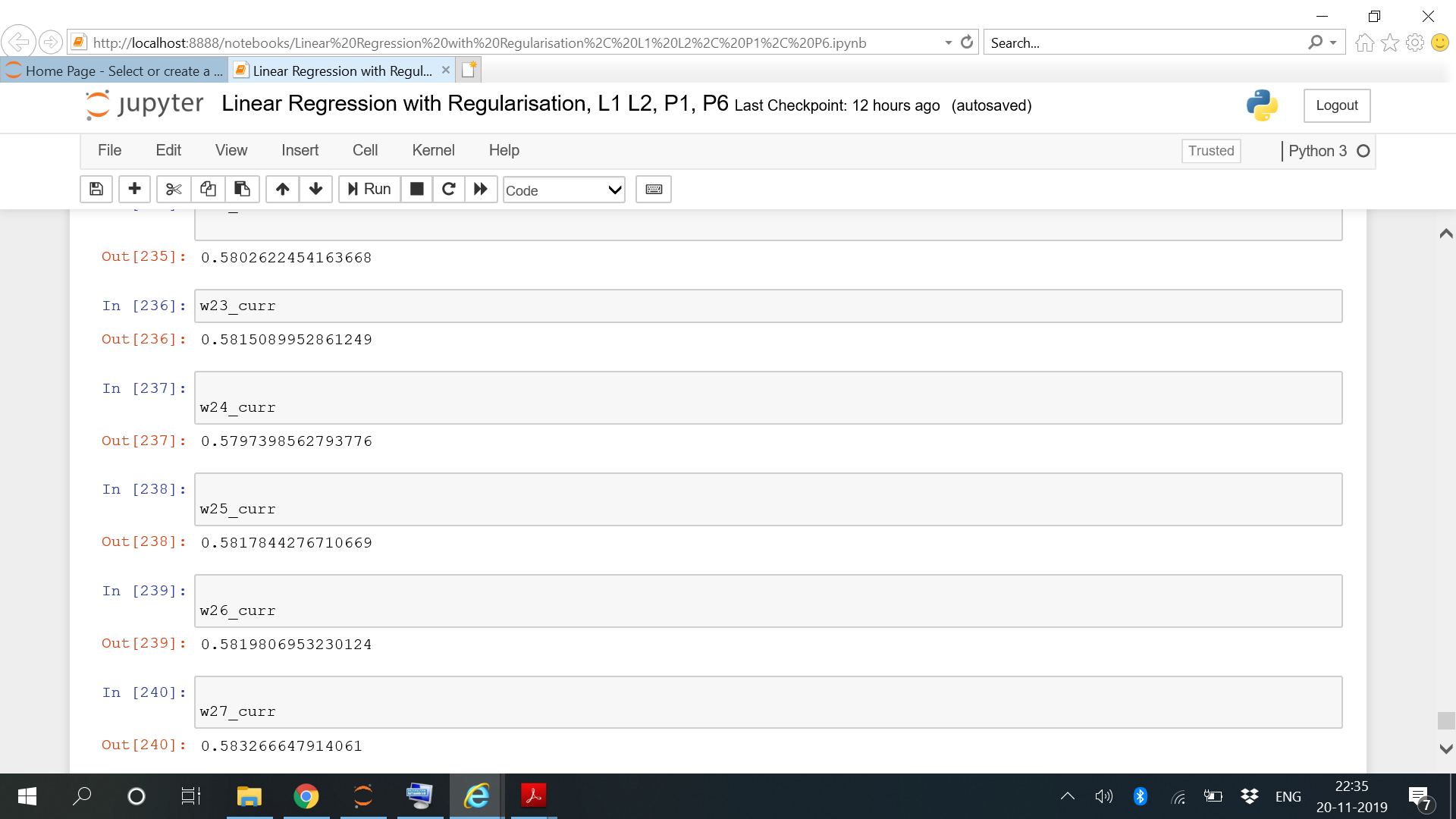
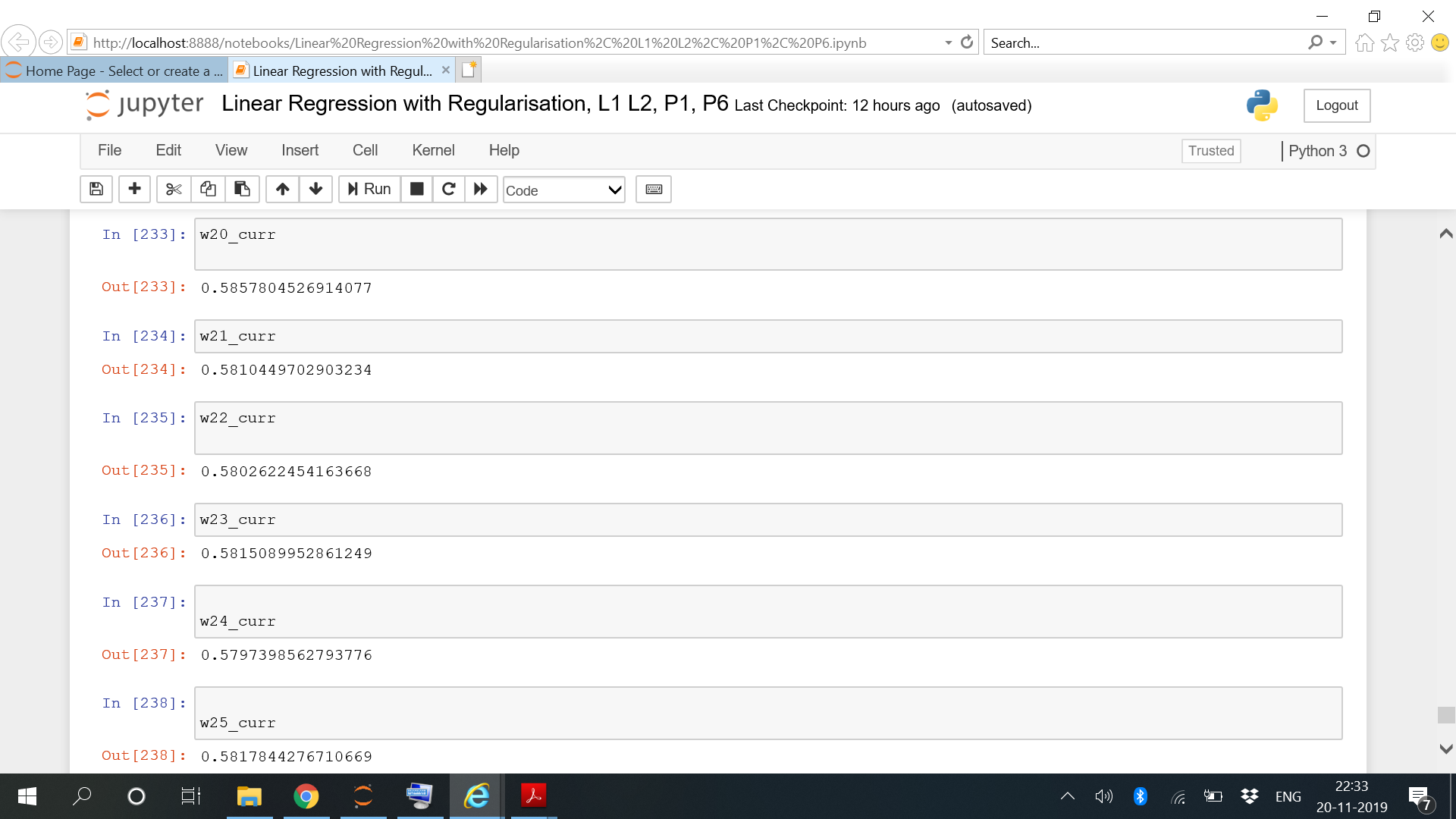
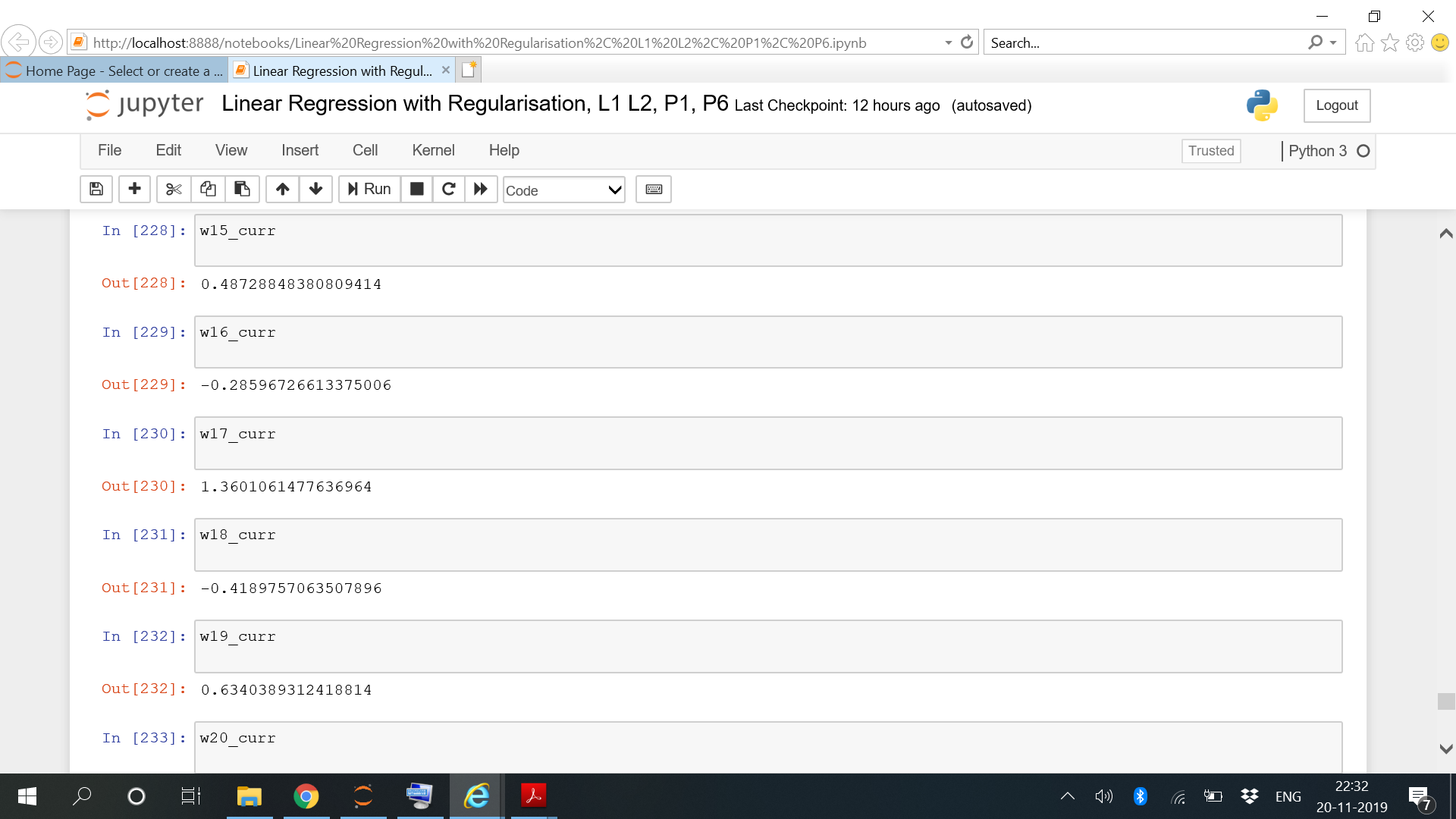
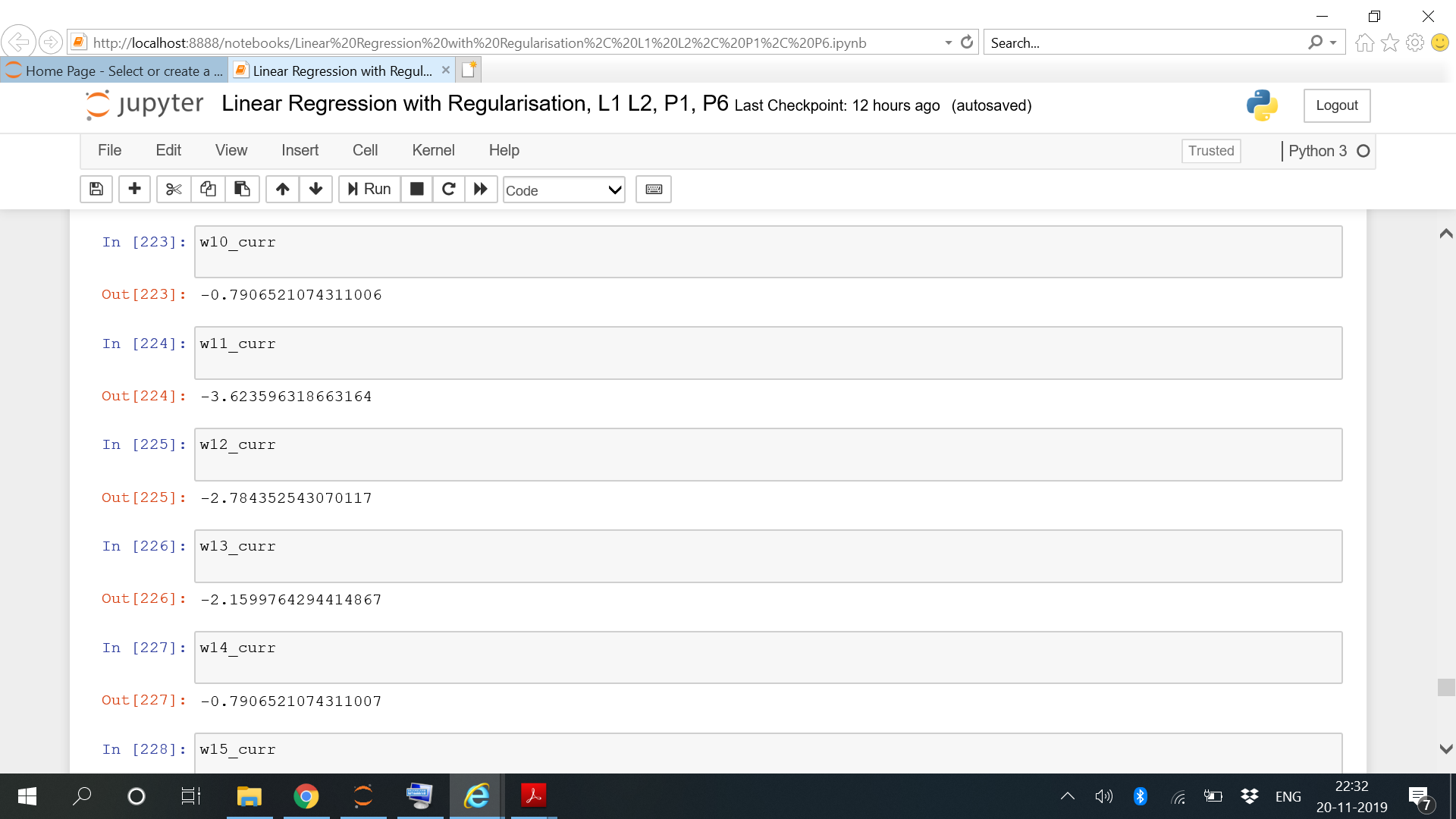
After testing data,

R2=0.5076791275541308

RMSE=12.57553008977919

The final coefficients obtained are as follows:

****

****