A sales forecast model

DATA MINING FINAL PROJECT

Oscar Iannucci s4843592, Nikita Matsnev s1082442´

Radboud University, The Netherlands

May 9, 2020

# ABSTRACT

In this paper we build a model for sales predictions based on a list of video games with sales greater than 100,000 copies. We use 2 algorithms in order to achieve our goal, namely gradient boosting and random forest regression.We evaluate the performance improvement in a medical image classification task which arises from using a rotation-equivariant block in a convolutional neural network, as opposed to using traditional convolutions. We find that performance is not improved much by using rotation equivariance: it yields an AUC score of 0.8949 while traditional convolutions yield 0.8823. Our work, complete with a clean Python implementation of rotation-equivariant networks, is made available online at

[https://github.com/gabrielraya/Data-Mining/tree/master/Project.](https://github.com/gabrielraya/Data-Mining/tree/master/Project)

*K*eywords Convolutional Neural Netowrks · Equivariant CNNs · Histopathology · Medical Imaging

## 1 Introduction

Companies are increasingly using data to boost their sales and be more competitive. One of the challenges that they might encounter is to select the right data and attributes to predict their sales and especially formulate accurate predictions. In order to achieve the aforementioned result, different methods can be applied, from easier to more complex ones, but some methods can be better than others.

In this paper we are proposing 2 types of algorithms in order to predict sales from a dataset with video games sales: gradient boosting and random forest regression (described in section 2). Our predicted variable is specifically EU sales and our predicting variables are: Platform, Year of making, Genre and Publisher.

We are using these 2 algorithms and comparing them since they both use different ways of prediction and, additionally, random forest regression can be sensitive to noisy data and create overfitting results. The backup in that case would be gradient boosting.

## 2 Application domain, research problem & methods description

2.1 Application domain & research problem

Our application domain is the video games industry.

We will apply gradient boosting and random forest regression in order to predict EU\_sales and compare the predicted results with actual results to see which algorithm performs better. We will train our gradient boosting tree and random forest regression trees based on the aforementioned attributes and validate the data via 10 fold cross validation.

2.2 Research methods

* Gradient boosting:

Gradient boosting is a ML technique which is highly recommended for prediction purposes. Specifically, this algorithm belongs to the tree-based algorithm categories and its main strength is that is suitable for weak learners, and it can reduce bias. In other words, this algorithm produces a strong learner with high accuracy by using an additive training method which implies the combination of multiple trees defined as “weak learners”. A graphic example of a gradient boosting tree algorithm can be found in figure 1.

Diagram, shape

Description automatically generated

(Source: https://www.researchgate.net/profile/Ivanna-Baturynska/publication/340524896/figure/fig3/AS:878319096569859@1586418999392/Schematical-representation-of-gradient-boosting-regression-in-regards-to-algorithm.png)

Furthermore, the algorithm is fast, does not need any preprocessing and shows robust results.

* Random forest regression:

Random forest regression is a model which combines many decision trees, not related to each other, into one unique model, therefore we can define it a meta-estimator. Moreover, this is a bagging technique, therefore it does not use any boosting method. As the name suggests, random forest is based on the classical concept of decision trees algorithm. However, we do know that decision trees generally tend to overtrain data. Random forest solves this issue by averaging multiple depths trees belonging to different parts of the same set, by having the final result of reduced variance. An example of random forest regression algorithm can be found in figure 2.

Diagram

Description automatically generated

(Source: https://www.researchgate.net/publication/313489088/figure/fig3/AS:864415041732616@1583104014685/Fig-A10-Random-Forest-Regressor-The-regressor-used-here-is-formed-of-100-trees-and-the.jpg)

This algorithm advantages include a high accuracy, large data handling and effective estimation of missing data.

## 3 Related work, similar problems

## Several studies already tried to achieve what we are trying to achieve, either with classification problems or regression problems. For example (reference Joanna Henzel) use gradient boosting in order to forecast and optimize promotion efficiency in retail. Similarly, ( referenceAlessandro Massaro) use gradient boosting in order to predict sales including promotion conditions. The prediction was applied to customer’s segments with personalized services according to their purchasing behaviour. Also (reference Maksim Korolev, Kurt Ruegg) and (reference Ankur Jain) used XGBoost to predict sales in the pharmaceutical industry based on historical data.

## We also have examples of related papers trying to use random forest regression in order to predict sales. For example (reference Jimenez F., Sanchez, G.) used historical available data in order to forecast sales based on random forest regression. Also (reference Suleka Helmini1) provides a comparison among different methods for forecasting sales in the retail sector, including random forest regression and gradient boosting.

## 4 Data set, data collection & preprocessing methods

4.1 Data Set

The dataset includes a list of video games with sales greater than 100,000 copies and the fields included are:

* Rank - Ranking of overall sales
* Name - The games name
* Platform - Platform of the games release (i.e. PC,PS4, etc.)
* Year - Year of the game's release
* Genre - Genre of the game
* Publisher - Publisher of the game
* NA\_Sales - Sales in North America (in millions)
* EU\_Sales - Sales in Europe (in millions)
* JP\_Sales - Sales in Japan (in millions)
* Other\_Sales - Sales in the rest of the world (in millions)

4.2 Data collection

Data was collected by an archive on Kaggle.com. The dataset of interest was previously generated by a scrape of vgchartz.com. There are originally 16,598 records and 2 records were dropped due to incomplete information. Therefore, we have in total 16,598 rows and 10 columns.

For our research problem we will focus specifically on:

* Nominal attributes: Platform, Year of making, Genre and Publisher.
* Continuous attributes: EU\_sales, NA\_sales

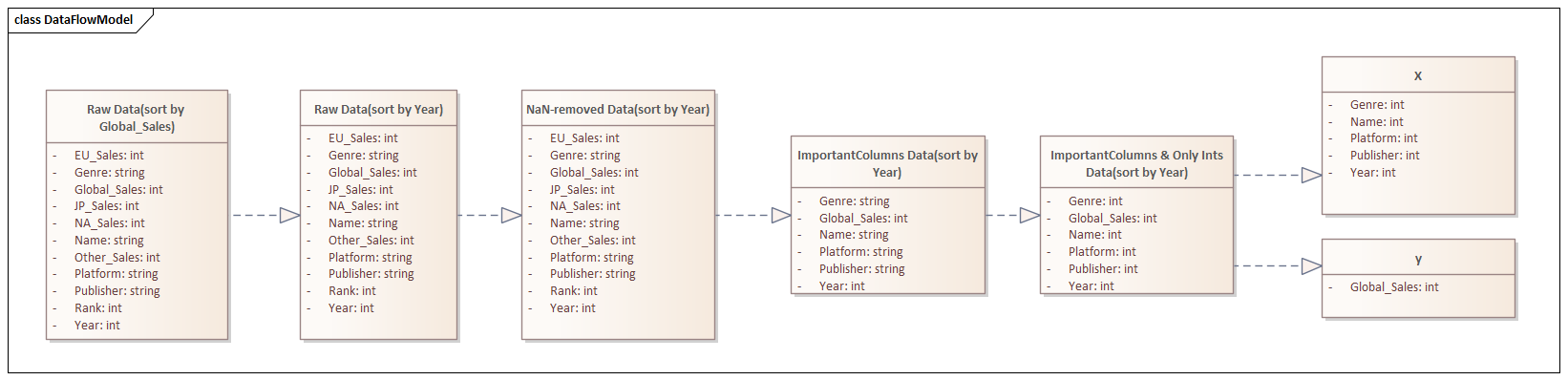
4.3 Data preprocessing

Before running our 2 selected algorithms, we need to make sure that our data is properly preprocessed:

We selected from the entire set only the attributes we were interested in. Since gradient boosting and random forest regression support only numbers, we converted the string values into integer values and made sure that both training and test data set are in the same numeric format.

The resulting subset would assume that transformed string values are having a unique integer value assigned. We also centered and standardized the data to give better reliability to the whole model.

We have removed redundant attributes such as regional sales and sorted everything due to the year in ascending order. And then we obtain data set X and expected output set y.



## 5 Approach/method(s)

5.1 Description

We perform random forest regression and gradient boosting regression on a dataset containing revenues per different world regions of gaming industry, dataset is from Kaggle. As already specified, this is a regression task where we take 4 columns as predictors (namely platform, year, genre, publisher) in order to predict values of 2 columns Y (EU/NA sales), with a set consisting of 16598 rows. When plotting the results, we use sales on the Y axis and year of publishing on the X axis, in order to see the sales trends per year of publishing and check how the performance YoY. Define further how we split test set, training set and validation set. We use the utility of SciKit Learn library “train\_test\_split” to split given data set in random train and test sets, where we have set up the size of testing set to be 17% as in the large datasets such as we have, it is more relevant to have more training than the default value suggests.

5.2 Training set

5.3 Test set

5.4 Validation

**6 Results**

6.1 Random forest regression

By performing random forest regression and plotting the results we can see that trends of predicted values mostly match the actual values. With the regression task, we also defined the importance of each predictor in defining the final prediction:

Variable: years Importance: 0.4

Variable: genres Importance: 0.25

Variable: publishers Importance: 0.22

Variable: platforms Importance: 0.13

6.2 Gradient boosting regression

By performing gradient boosting regression we can see that it is based on the publishers instead.

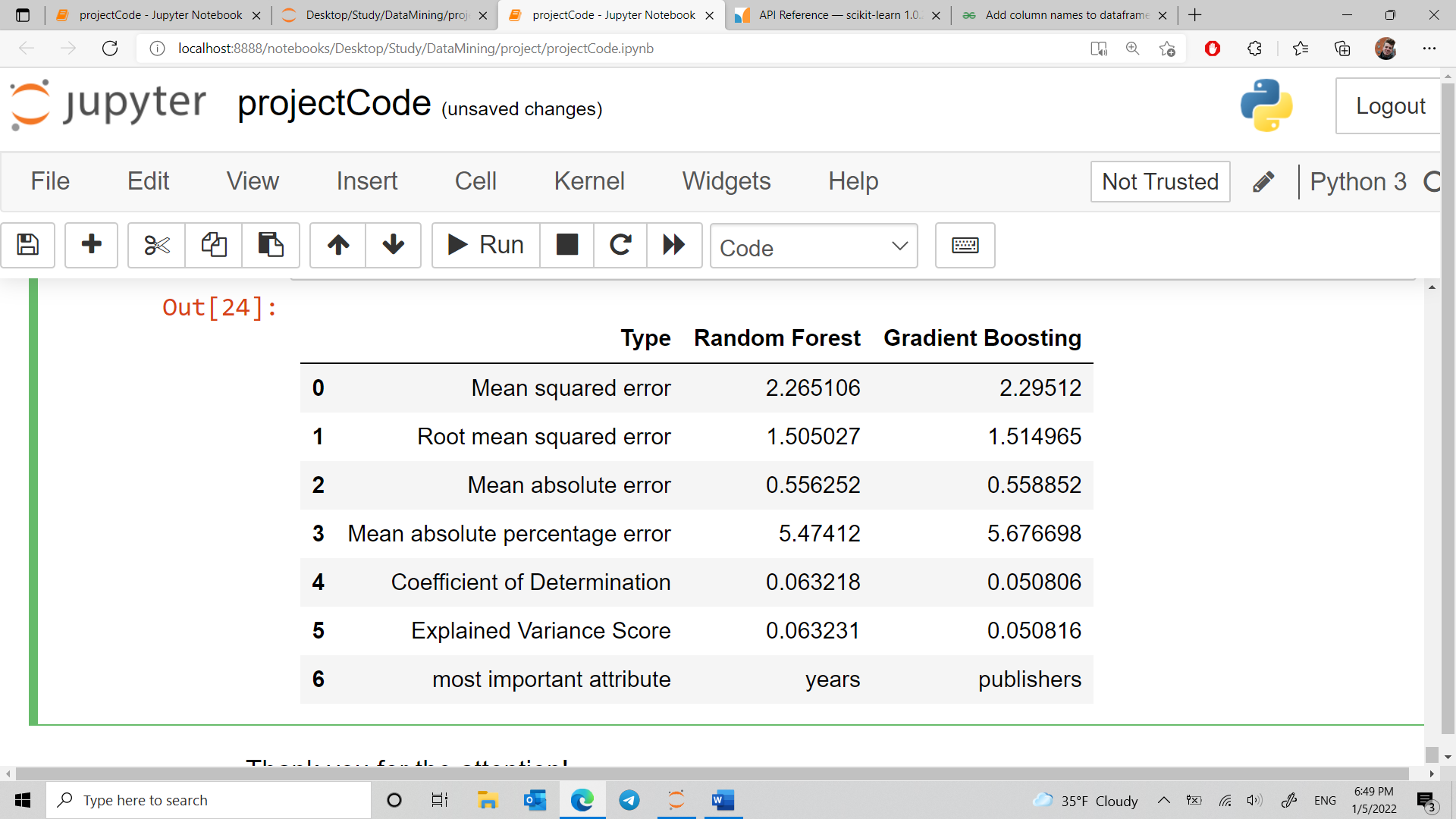
Variable: years Importance: 0.11

Variable: genres Importance: 0.04

Variable: publishers Importance: 0.82

Variable: platforms Importance: 0.03

Surprisingly, both methods perform quiet similar results in all of the tests we have managed with some deviations in Coefficient of determination and Explained variance score and also minor fluctuations in MAPE.



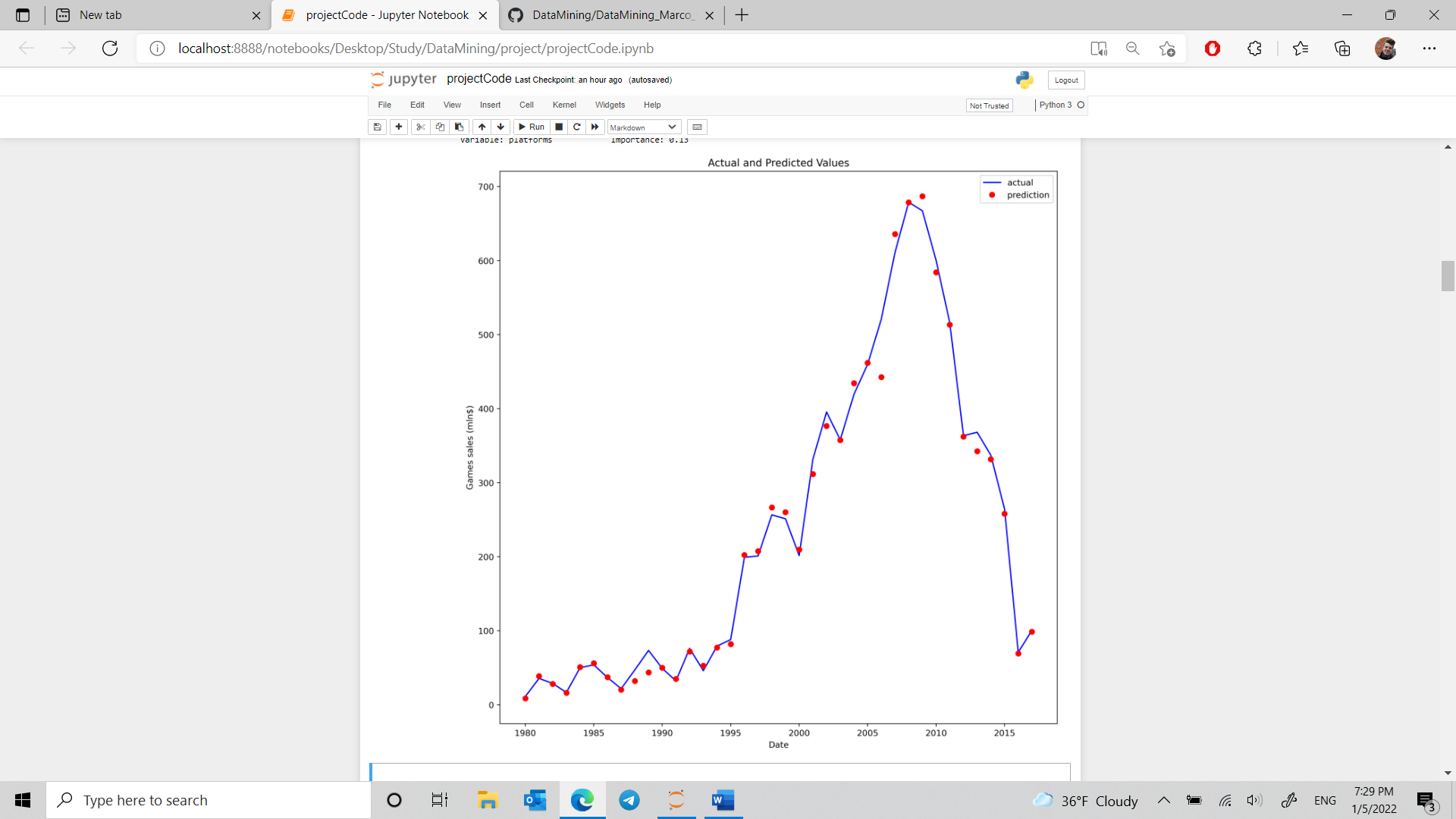


Figure 1. Random Forest Predictions vs. the actual data

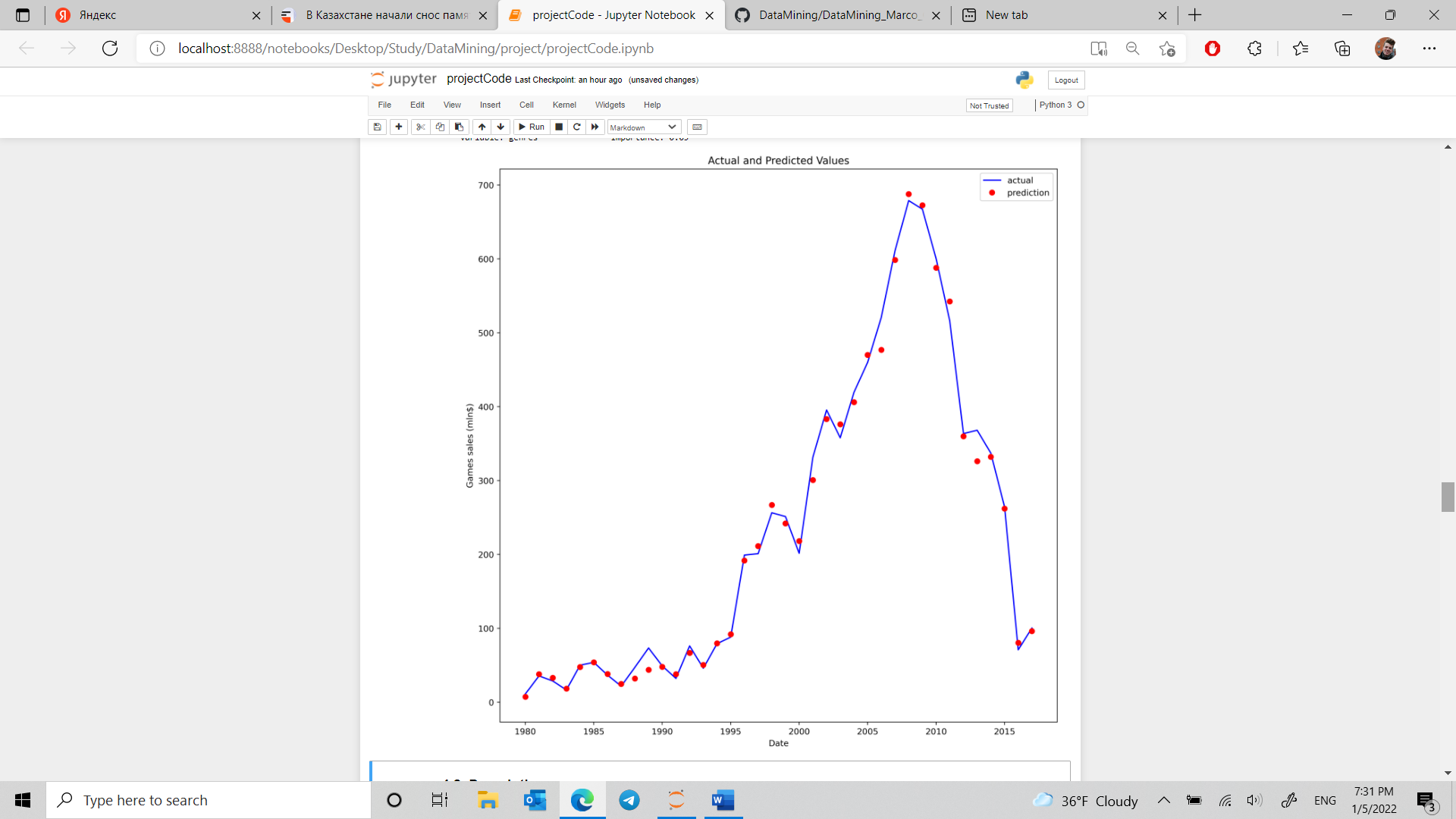
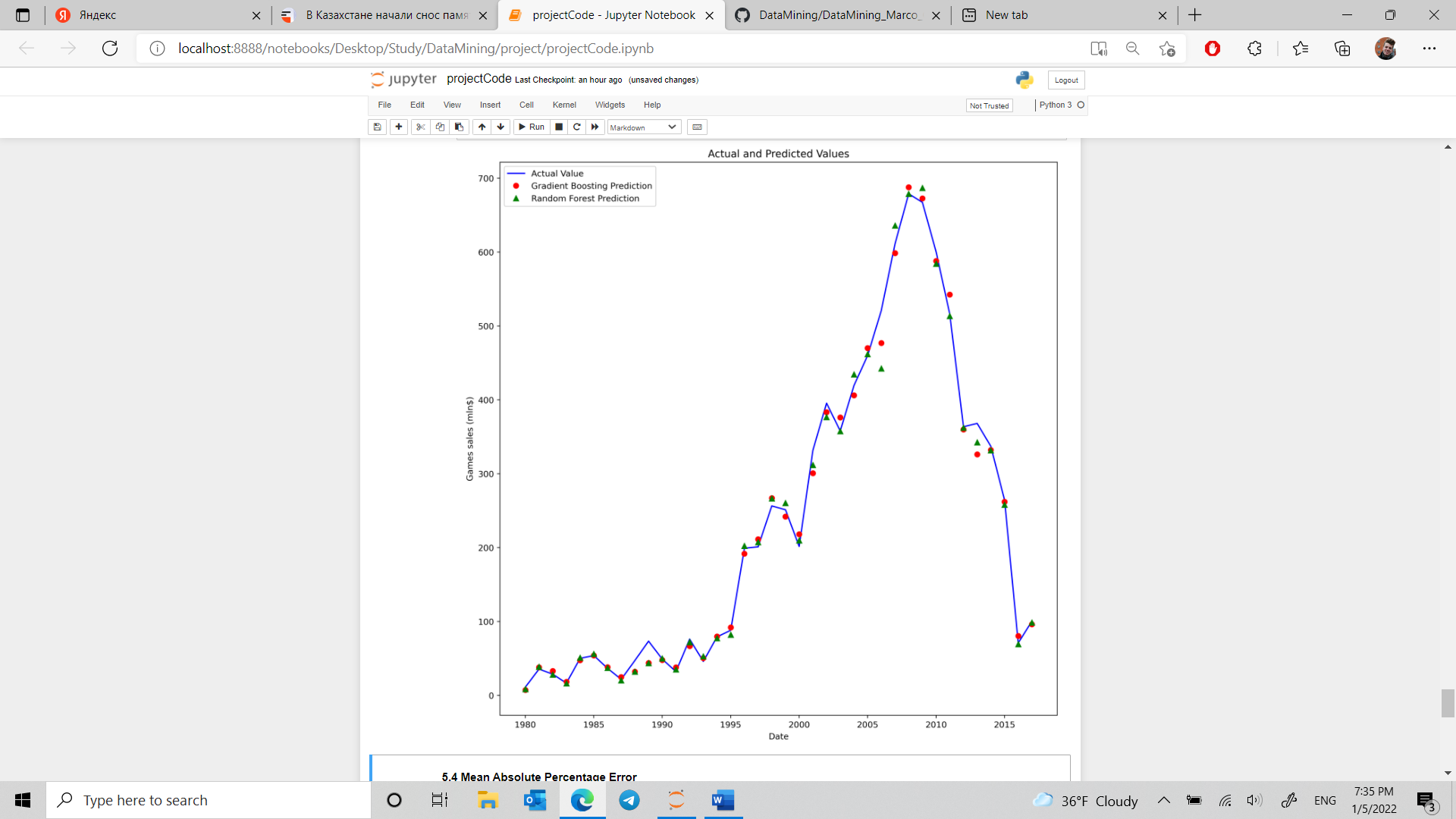


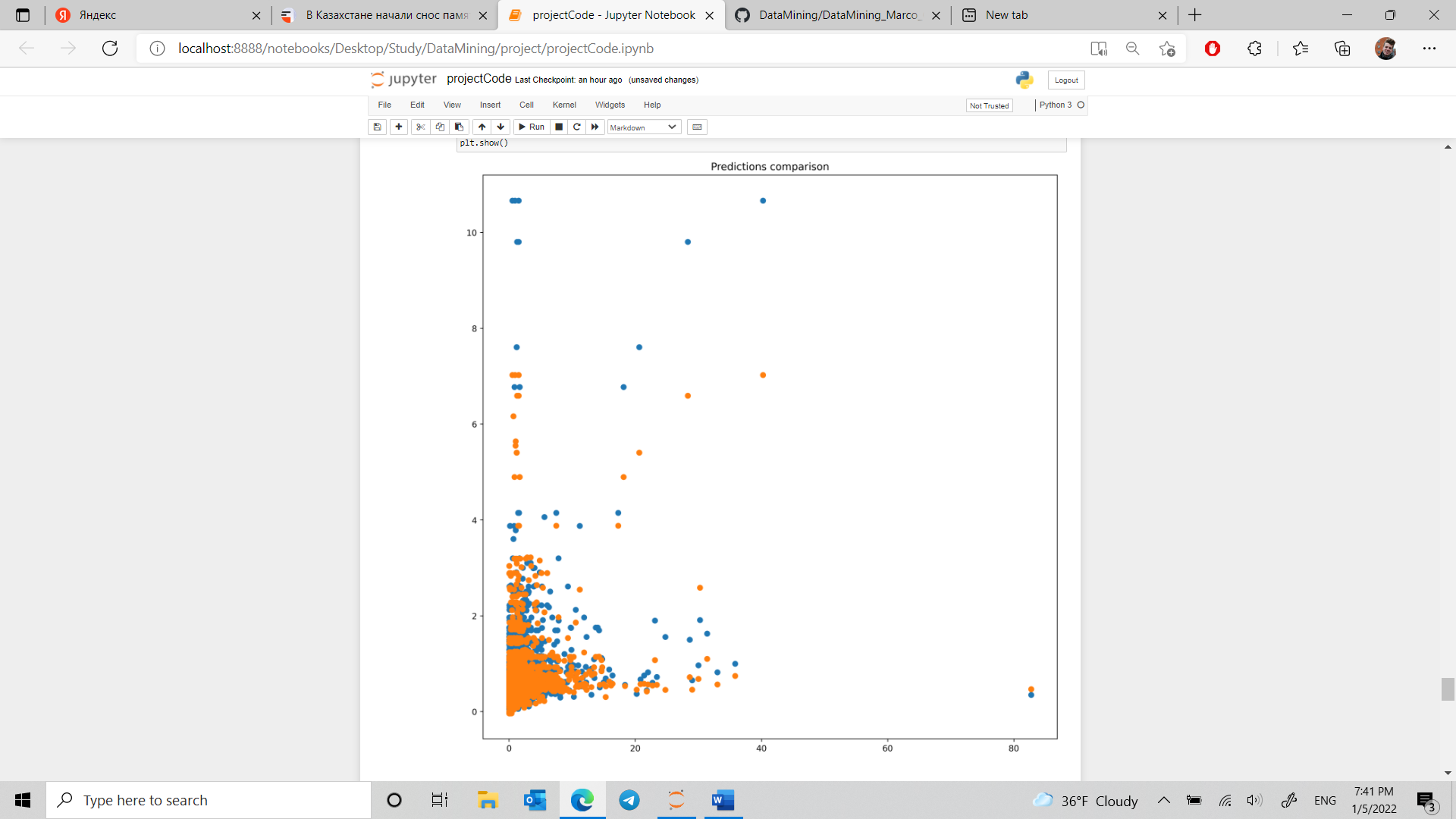
Figure 2. Gradient Boosting Predictions vs. the actual data

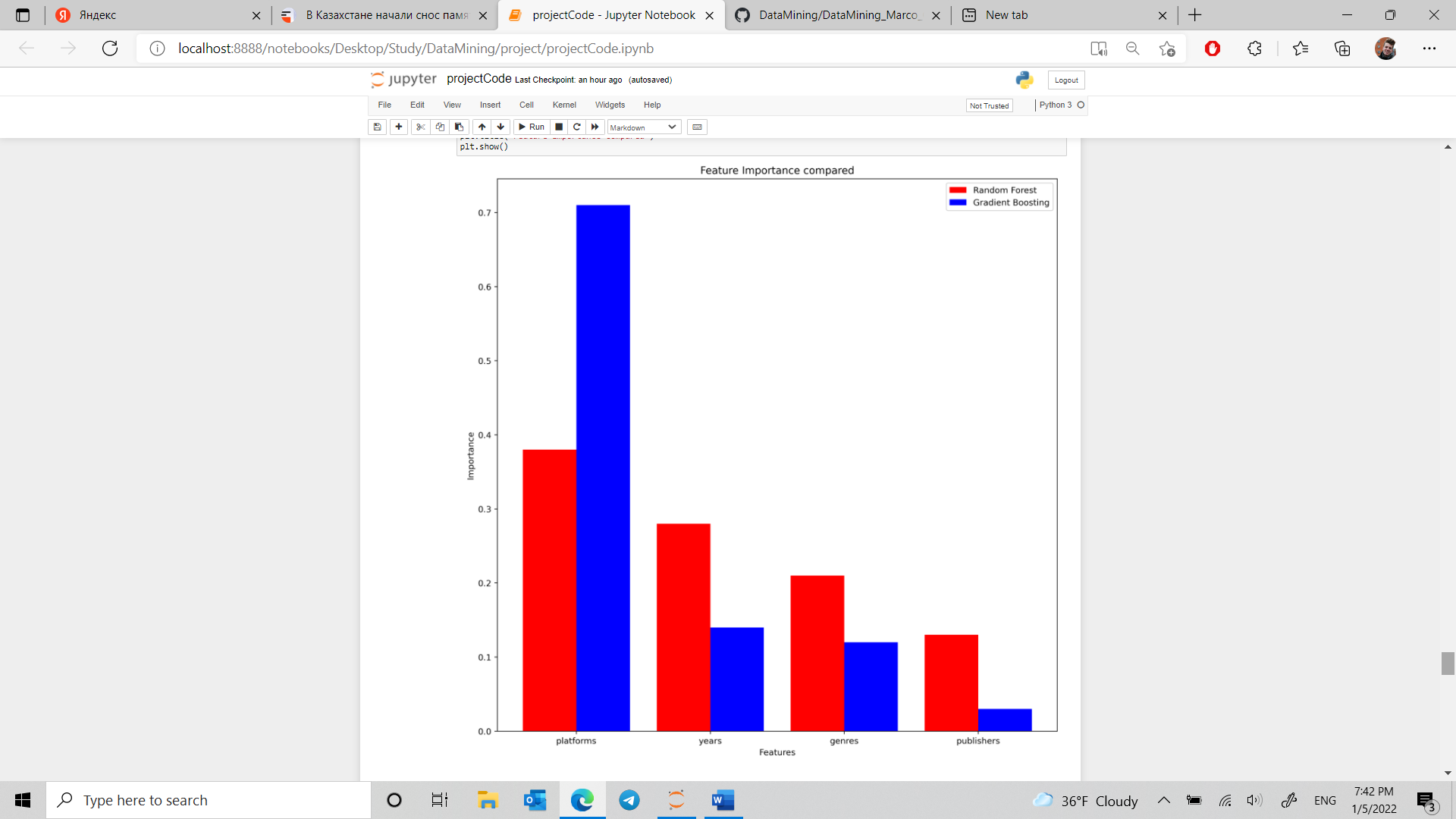
To be more specific, let’s take a look at these graphs combined:



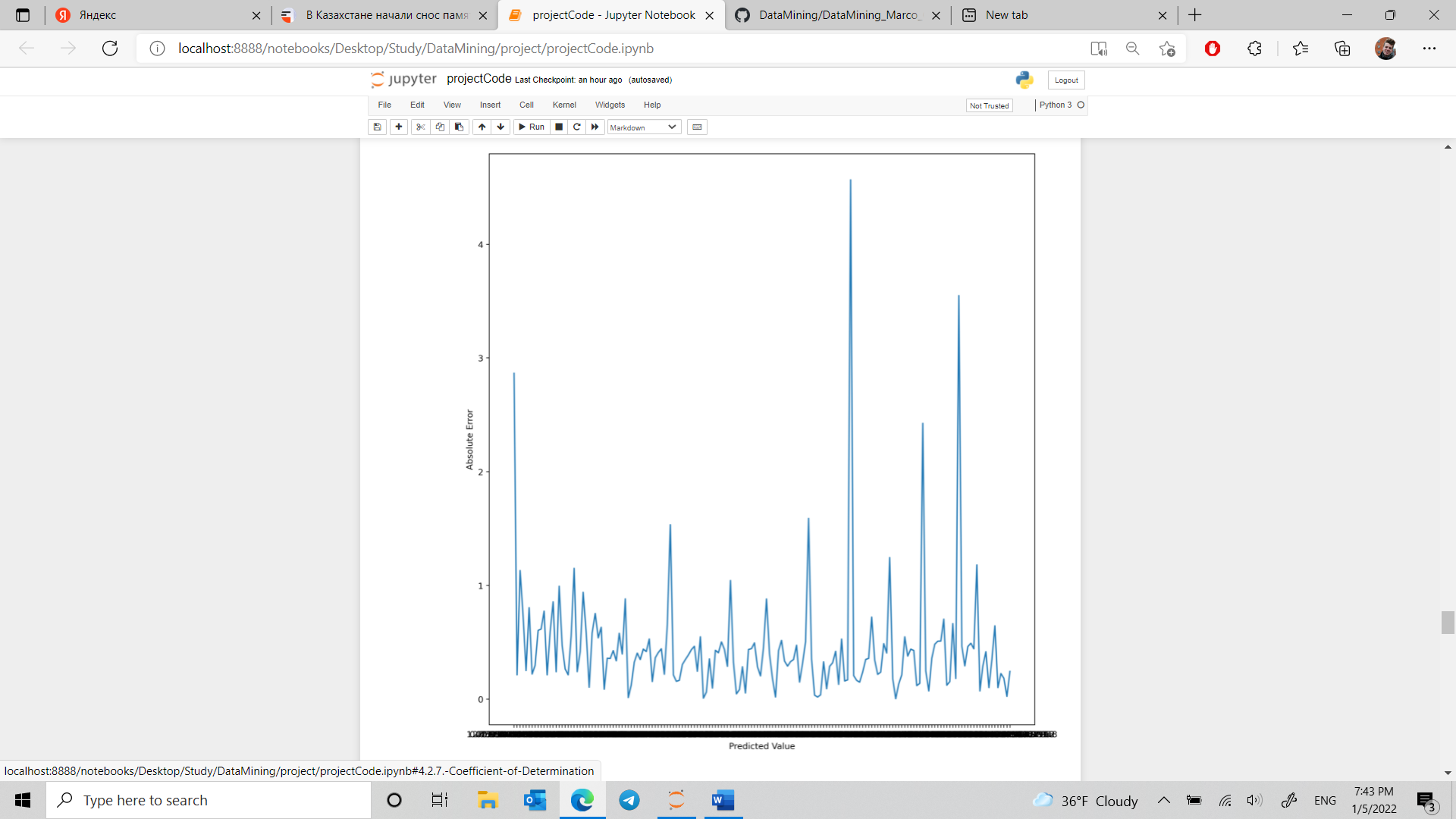
We can clearly see that Gradient Boosting, if miscalculating, tends to assume bigger value rather than Random Forest, and that is interesting, and you will elaborate on that.

We will have to explain all the stuff from the project, I ve added the plots here:

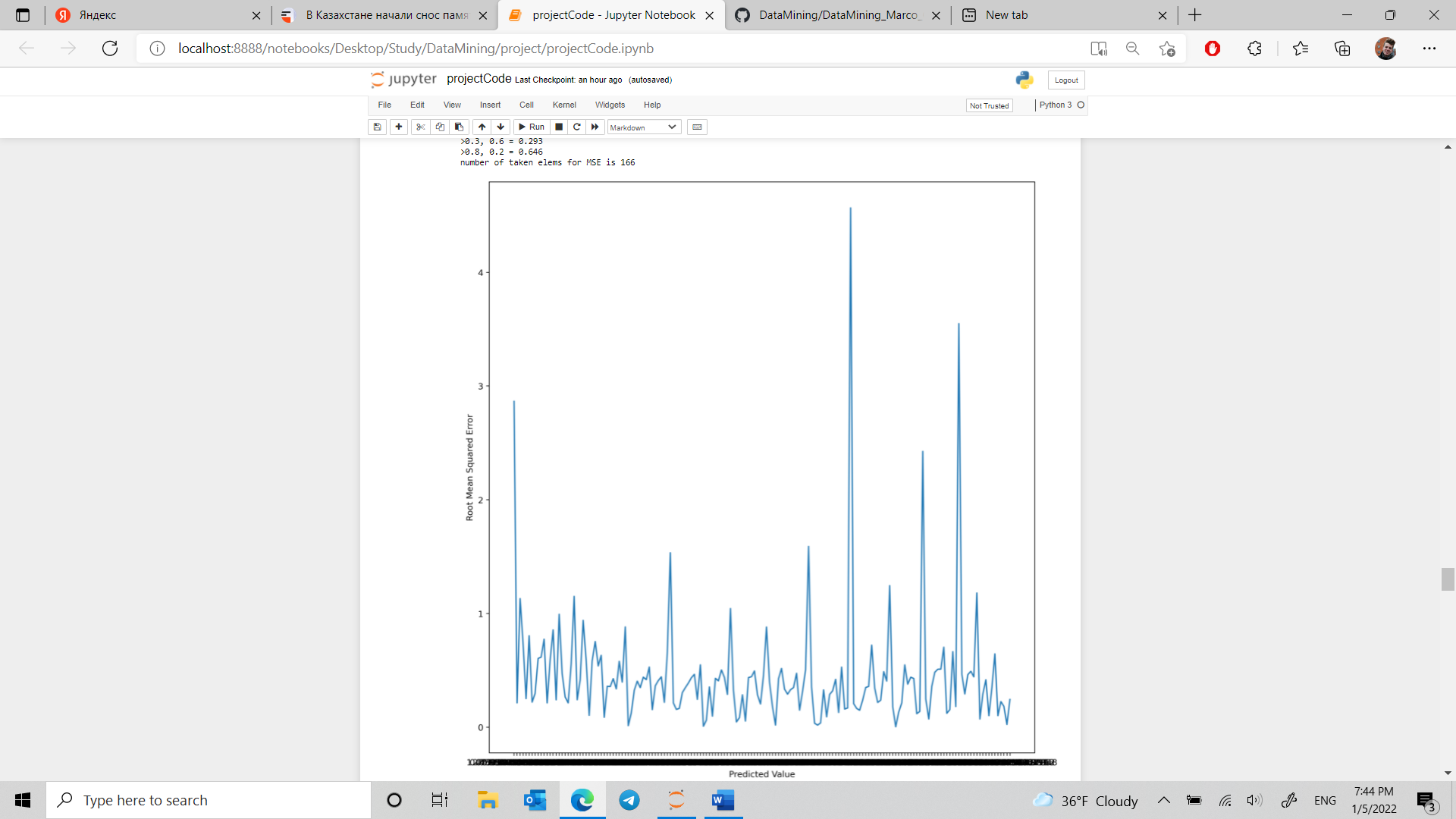




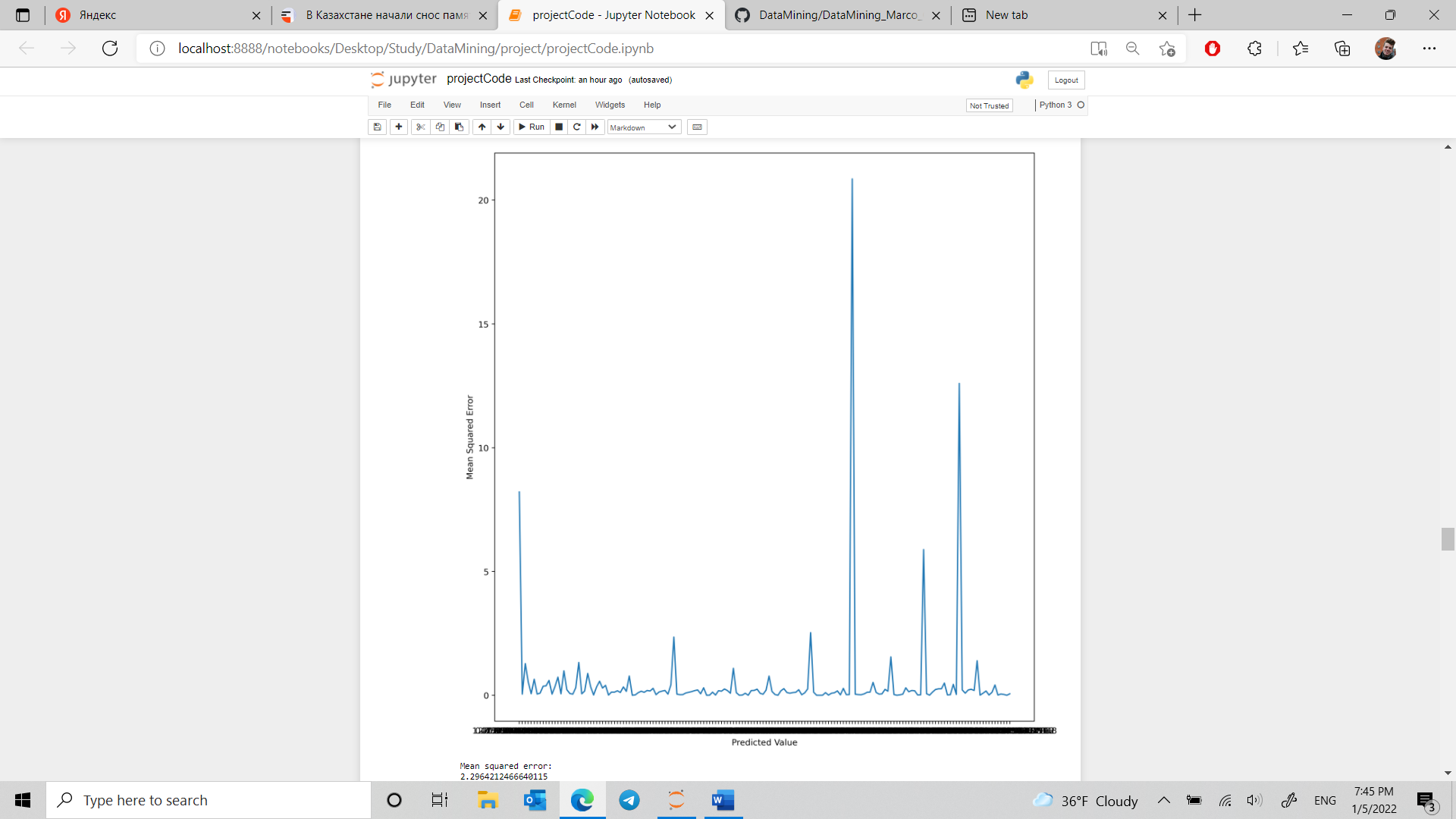
MAE of GB



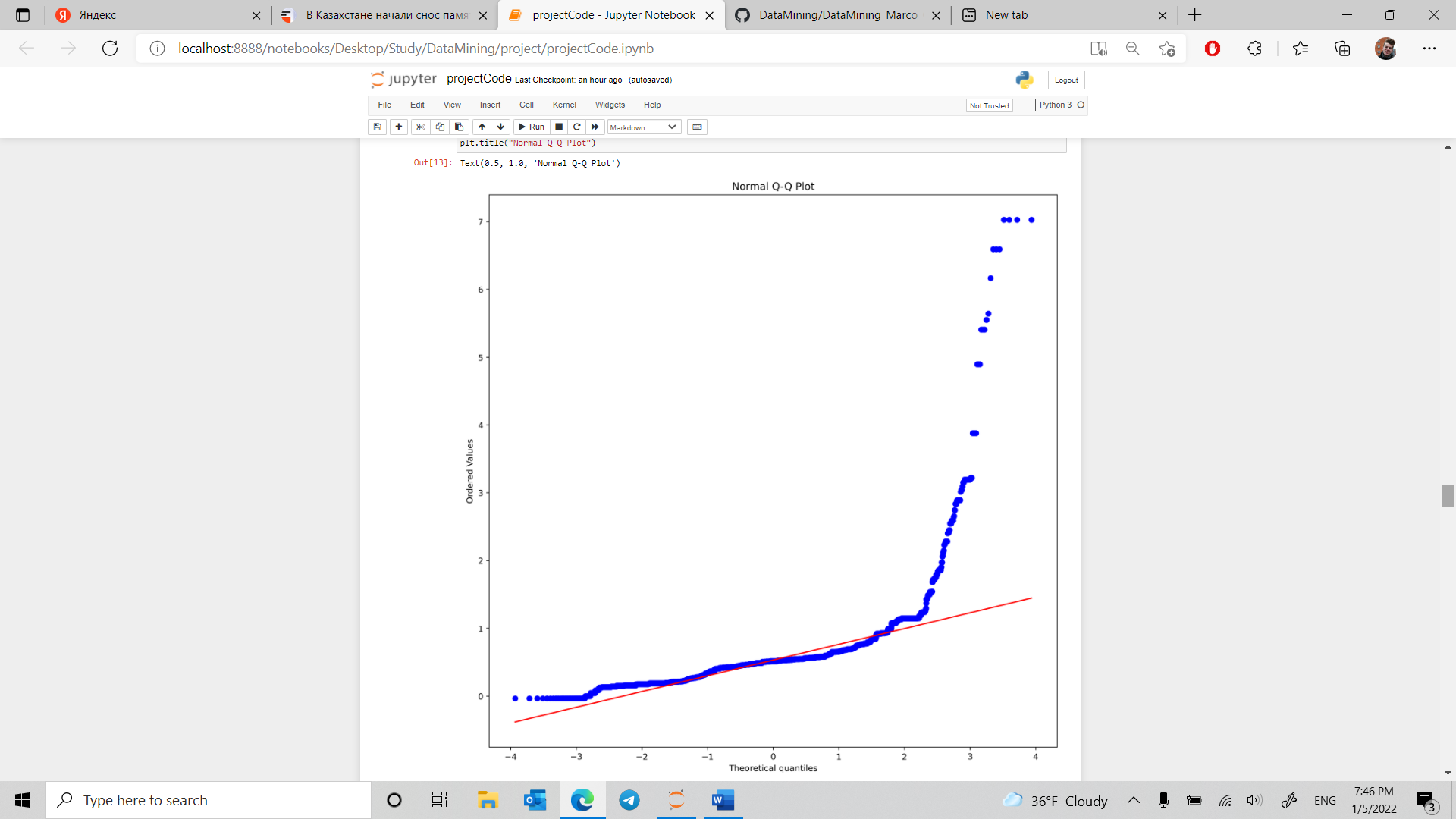
Root MSE



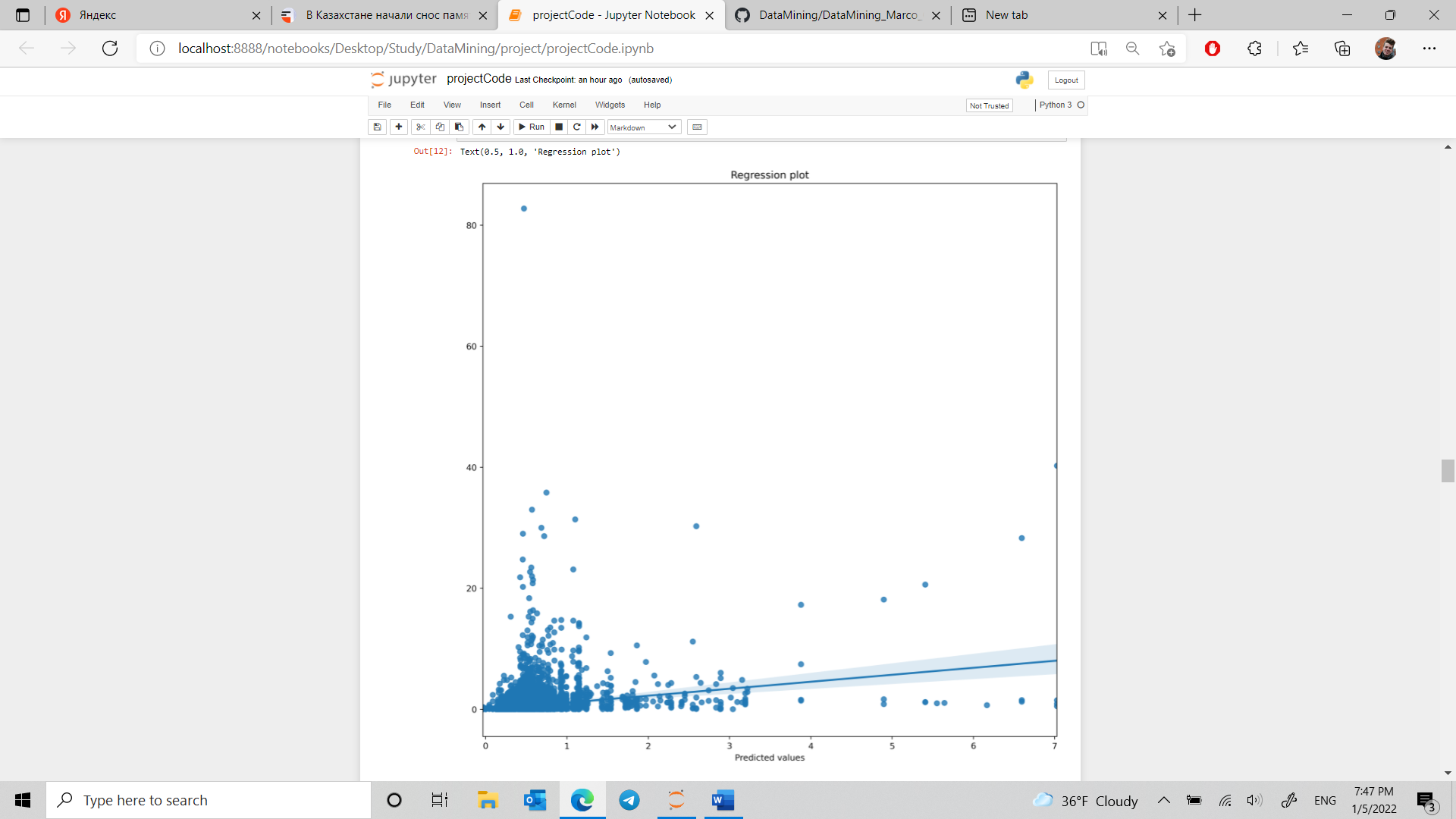
MSE of GB



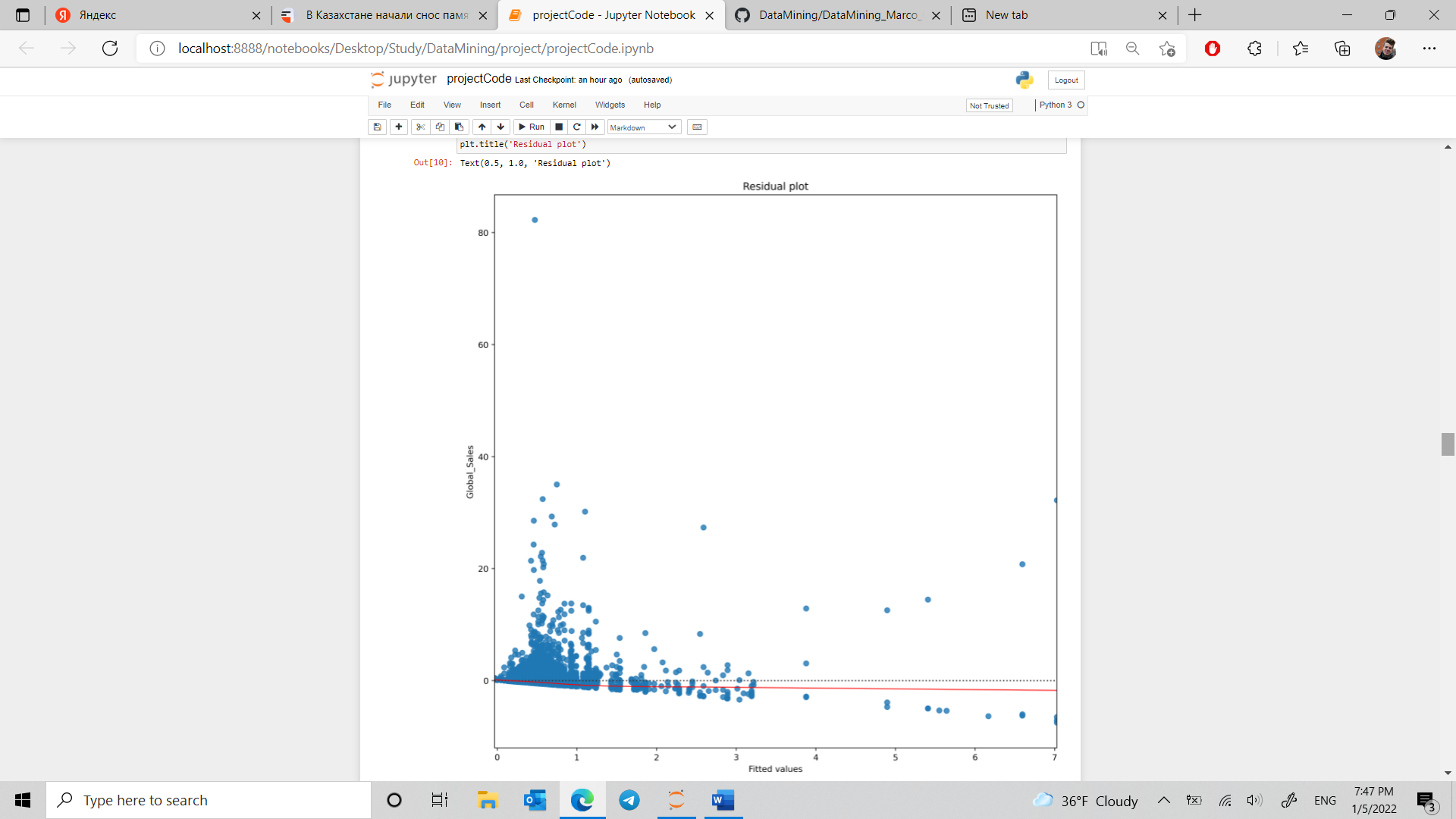
QQ plot of GB



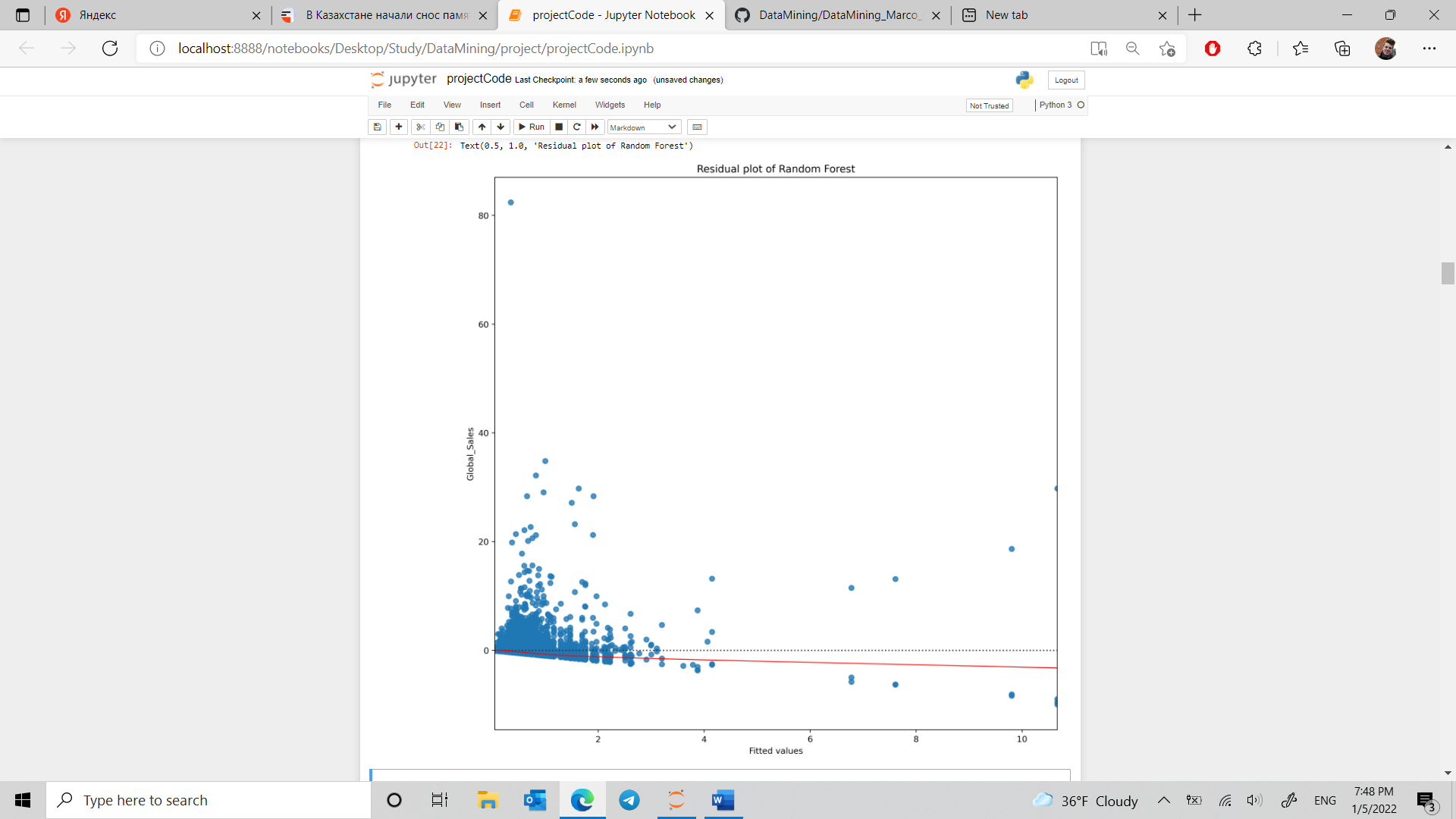
Regression plot GB



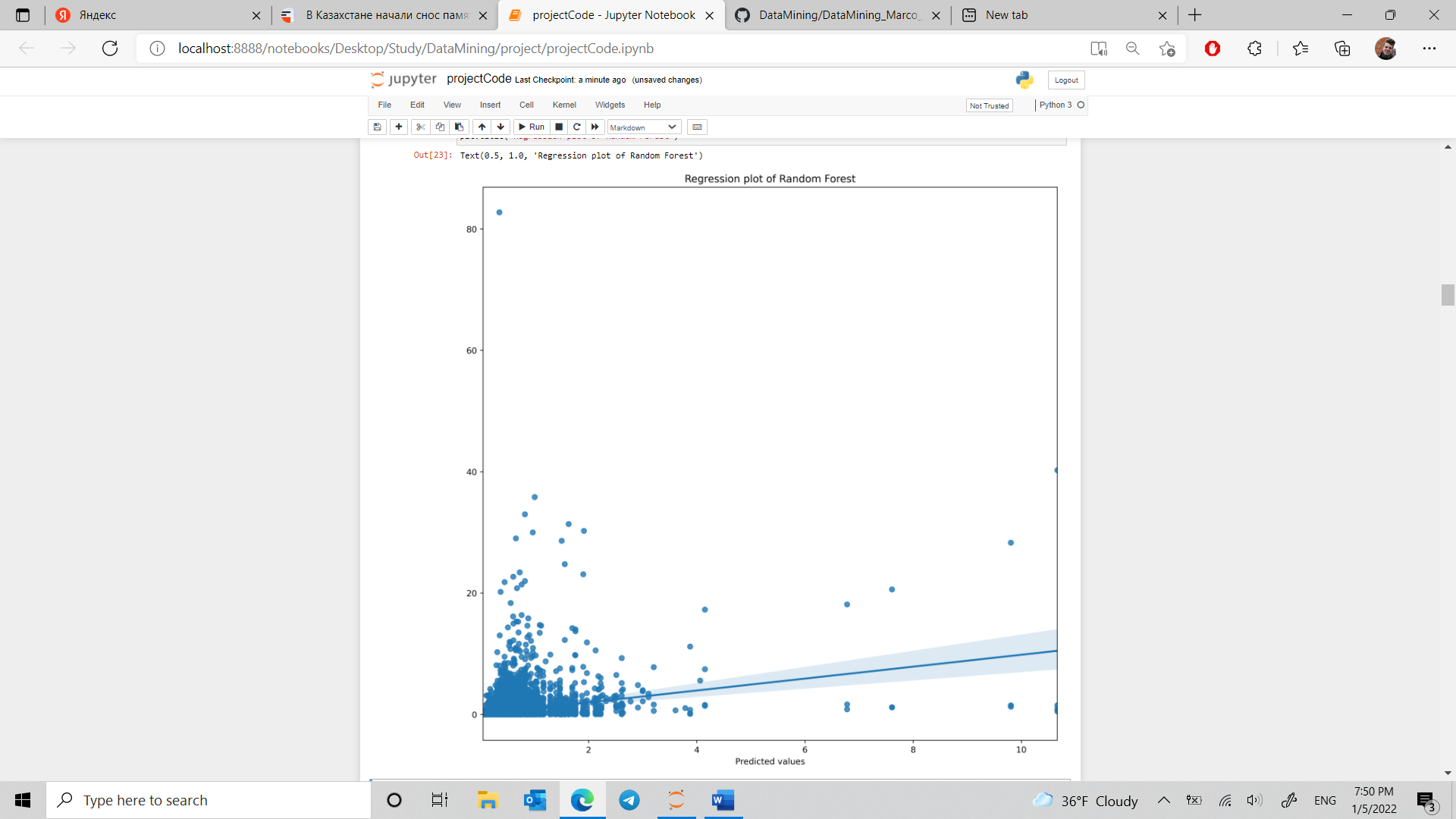
Residual plot



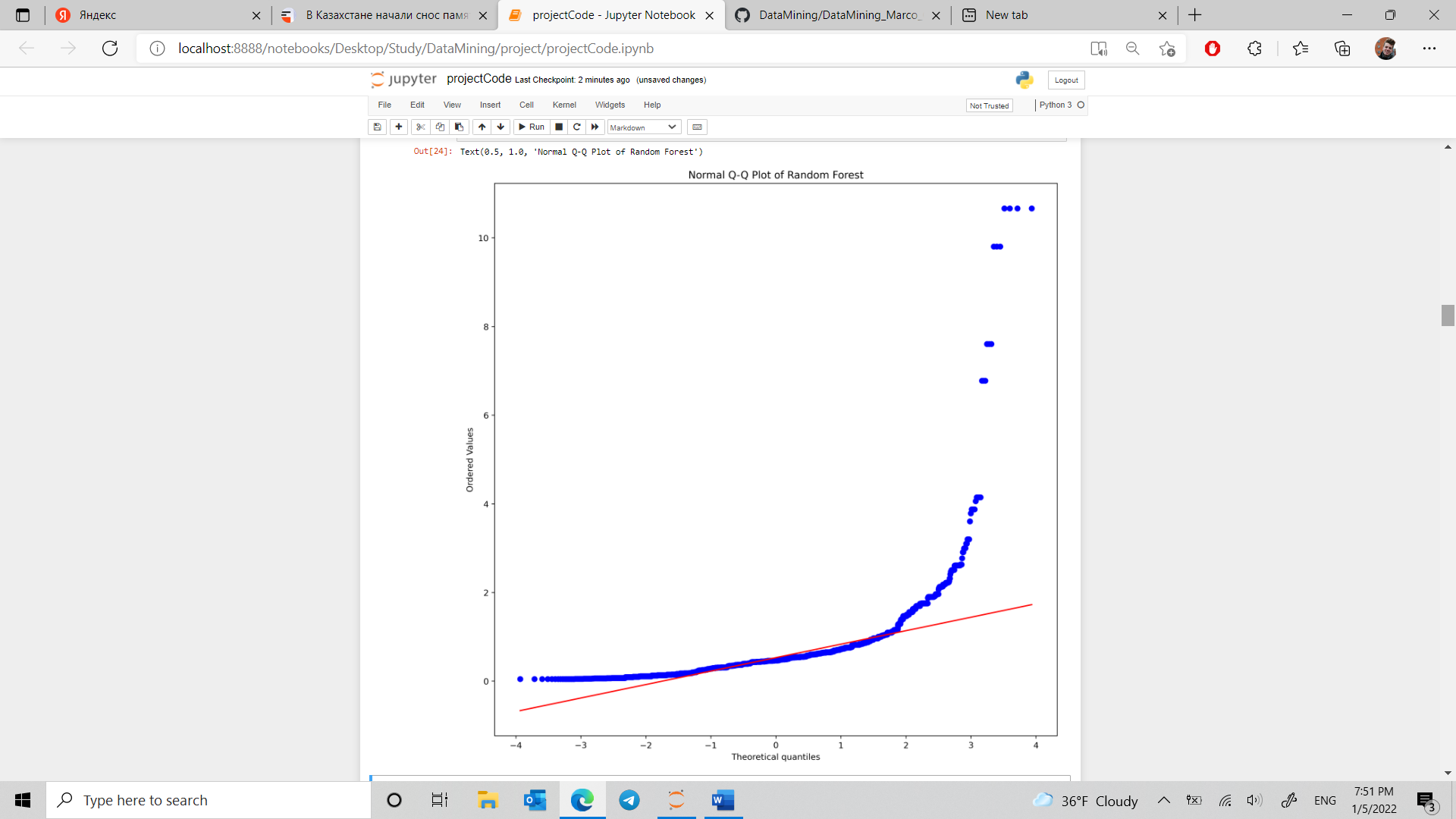
Residual plot of Random Forest



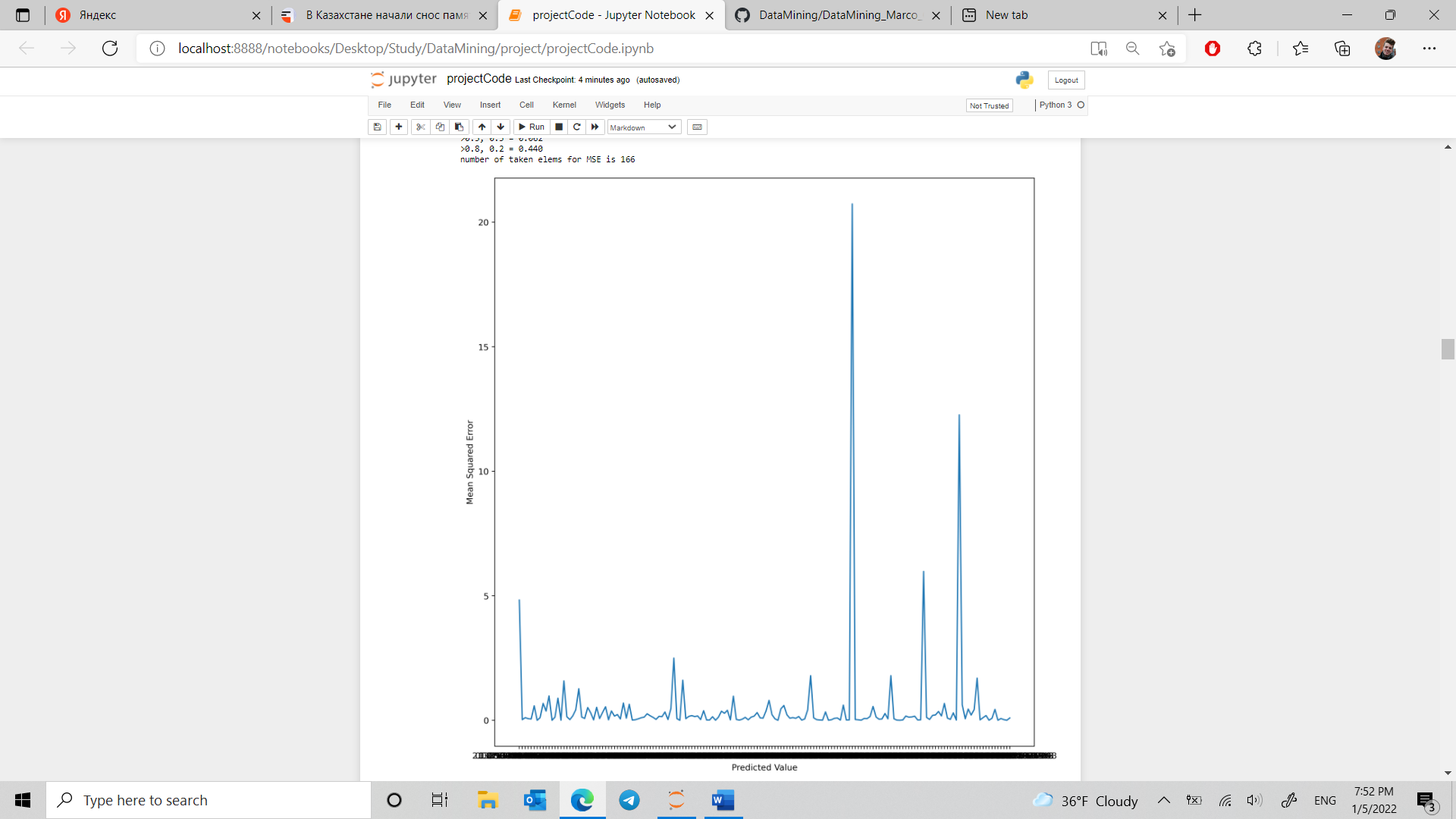
Regression plot of RF



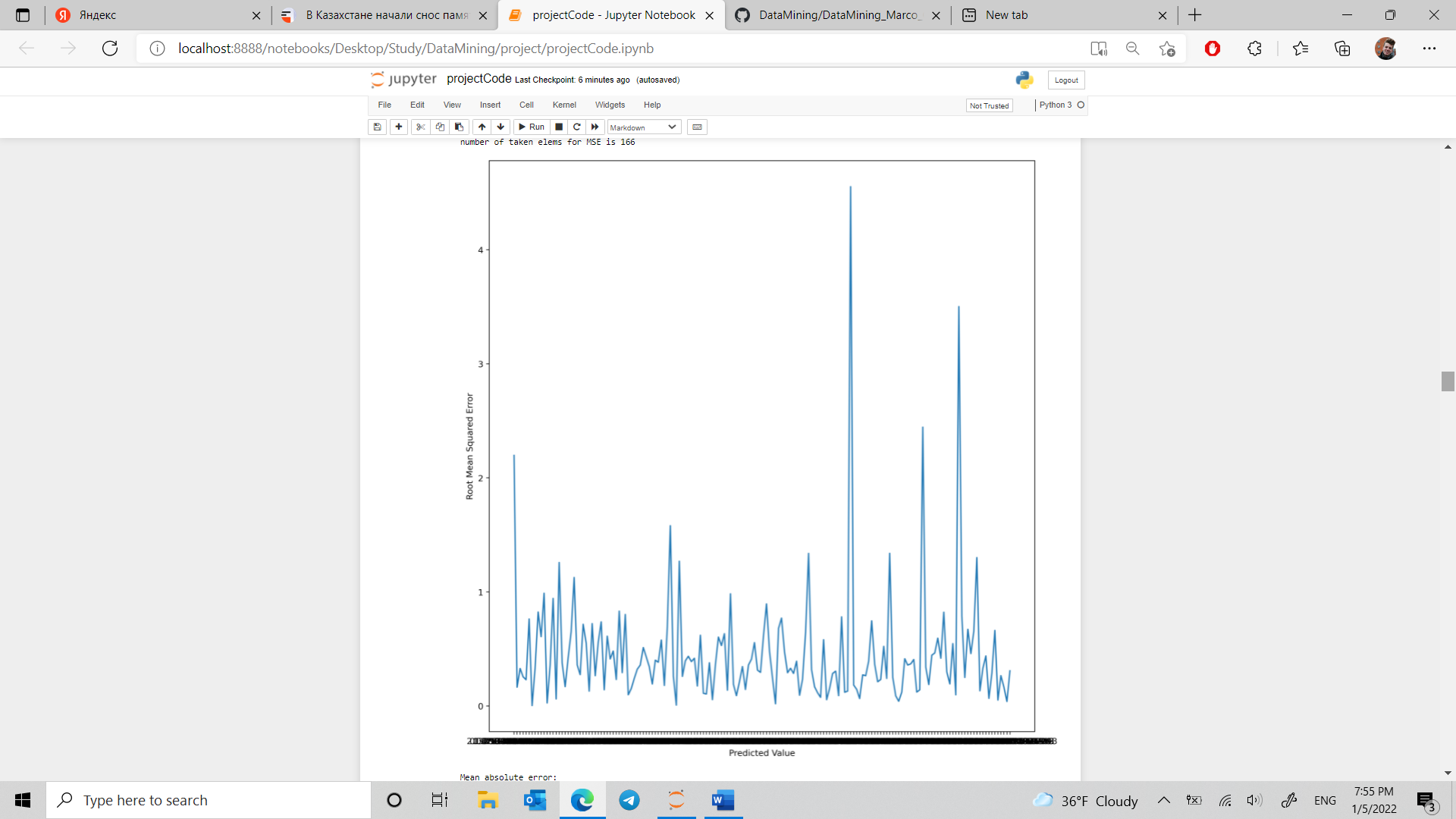
QQ plot of RF



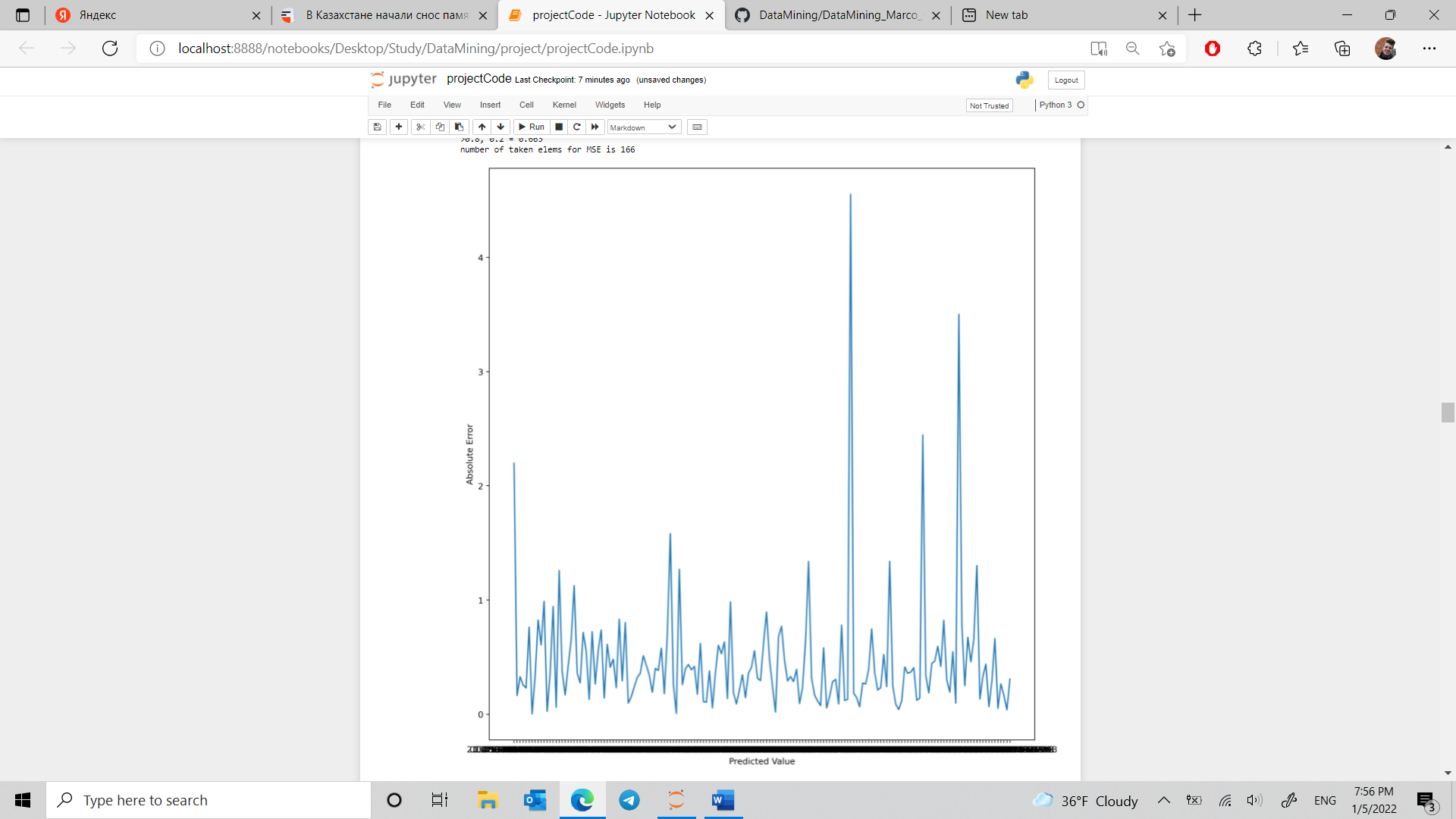
MSE RF



Root MSE RF



MAE RF



To more objectively evaluate this difference, we perform a sign test (*H*0: the equivariant model performs the same or worse as the standard model, *HA*: the equivariant model outperforms the standard model) and obtain an insignificant p-value 0*.*105.

For completeness, in figure 6, we also present the confusion matrices for the each of the models at a cut-off threshold of 50% confidence for the "tumor" class. As can be seen, these matrices do not differ significantly.

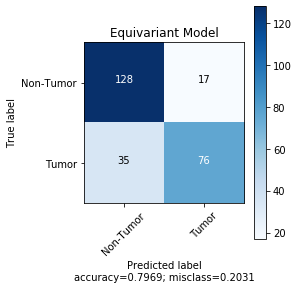
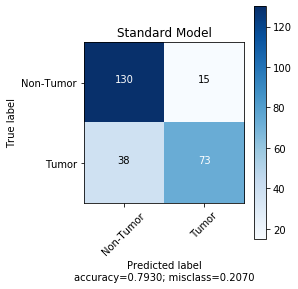


Figure 6: Confusion matrix comparison

To investigate if there are, in fact, any significant differences between the models in terms other than performance, we plot the confidence score for the "tumor" class as given by the equivariant model against that given by the standard model in figure 7. As can be seen, the scores assigned to samples by the models are quite similar, meaning that they are likely paying attention to similar aspects of the image.

Chart, scatter chart

Description automatically generated

Figure 7: Probability of belonging to "tumor" class, as estimated by different models.

To further investigate the differences between the models, we visualize their internal activations at different layers when viewing an example image - see figure 8. Again, it can be seen that they are processing images in similar ways - they focus on the same groups of cells.

A picture containing text, display

Description automatically generated

Standard model

A picture containing text, display

Description automatically generated

Equivariant model

Figure 8: Internal model activations (green, pooled over channels) when looking at example image (pink).

## 5 Conclusions & future work

On this particular dataset, and for this particular architecture, rotation equivariance proved not to provide a particularly significant boost in network performance. However, to fully evaluate the strengths and weaknesses of the rotationequivariant layer, one would need to perform an evaluation on many more datasets and using many more architectures. Especially the architectures would need scrutiny; we suspect that the observed lack of difference between the equivariant and standard models is due to the fact that the amount of parameters in the standard model was enough to examine the image under all rotation/flip combinations. This is room for future work.

To facilitate this, we publish our code, which contains a clean Python implementation of rotation-equivariant convolutional neural networks. This implementation utilizes Pytorch[3]; we use Scikit-Learn[4] for evaluation.

## References

*Video Game Sales*. (2016). Kaggle. <https://www.kaggle.com/gregorut/videogamesales>

Henzel J., & Sikora M. (2020). Gradient Boosting Application in Forecasting of Performance Indicators Values for Measuring the Efficiency of Promotions in FMCG Retail. *Silesian University of Technology*. Retrieved from <https://arxiv.org/pdf/2006.04945.pdf>

Massaro, A., Panarese, A., Giannone, D., & Galiano, A. (2021). Augmented Data and XGBoost Improvement for Sales Forecasting in the Large-Scale Retail Sector. *Applied Sciences, 11(17)*, 7793. <https://doi.org/10.3390/app11177793>

Jain A., Manghat Nitish M., & Saurabh C. (2015). Sales Forecasting for Retail Chains. Retrieved from <https://cseweb.ucsd.edu/classes/wi15/cse255-a/reports/fa15/004.pdf>

Jiménez, F., Sánchez, G., García, J., Sciavicco, G., & Miralles, L. (2017). Multi-objective evolutionary feature selection for online sales forecasting. *Neurocomputing, 234*, 75–92. <https://doi.org/10.1016/j.neucom.2016.12.045>

Helmini, S., Jihan, N., Jayasinghe, M., & Perera, S. (2019, May 8). *Sales forecasting using multivariate long short term memory network models*. PeerJ Preprints. <https://peerj.com/preprints/27712/>

## <https://arxiv.org/pdf/2006.04945.pdf>

## <https://www.mdpi.com/2076-3417/11/17/7793/pdf>

## <https://cs229.stanford.edu/proj2015/193_report.pdf>

## <https://cseweb.ucsd.edu/classes/wi15/cse255-a/reports/fa15/004.pdf>

## <https://www.researchgate.net/profile/Luis-Miralles-2/publication/311730529_Multi-objective_evolutionary_feature_selection_for_online_sales_forecasting/links/5a15aa5e0f7e9bc6481c70cf/Multi-objective-evolutionary-feature-selection-for-online-sales-forecasting.pdf>

## <https://peerj.com/preprints/27712.pdf>