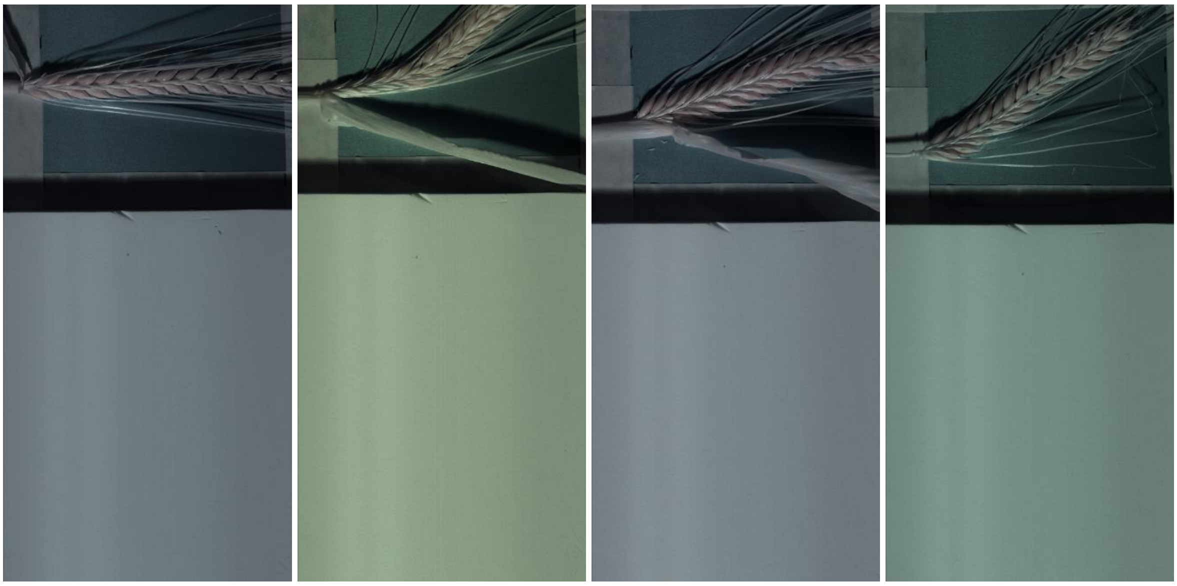
CONTENTS

RESULT OF SEGMENTATION ON ALL IMAGES

Segmentation works for all **SWIR images** except one image excluding five other damages images (images which are not taken from a crop). **So it works for 99.8 % of images**.

For **VNIR it works poorly, it was working only for around 80 % of images**. The reason was that some images had highly different values even for the same part of the images. It can be clearly seen from below images.



Based on this change in colors while exactly same spectrums was used for all images, the segmentation was working poorly. In order to have solved this issue, **we took use of 6 channels by preparing two RGB images** and as a result of segmentation of both images with different kind of parameters and **subsequently taking union of the segmented images** we got somewhat better result which is **working for almost 97% of images**.

VNIR SEGMENTATION:

Below is the channels and parameters used for segmentation.

- First RGB image:
  - Channels used for first RGB image: 23, 31, 39
  - Lower bounds = (0, 0, 70)
  - Upper bounds = (12, 255, 255)
- Second RGB image:
  - Channels used for first RGB image: 159, 140, 120
  - Lower bounds = (0, 0, 0)
  - Upper bounds = (60, 255, 255)

After taking the union of both images, same procedure applied as SWIR images.

INFO ON IMAGES:

	SWIR	**VNIR**
Number of images uploaded in cloud	485	484
Downloaded	485	482

Damaged (refer to respective folder)	5	5
Segmented (refer to respective folder)	480	477
well segmented (refer to respective folder)	477	463
improperly segmented (refer to respective folder)	3	14
Images that dataset built based on (two duplicates removed)	460	460

Figure 1: damaged images

You can take a look on segmented images. Using below link

https://drive.google.com/open?id=1pnGVAzpg3ryjDiXc4QqN_k0UcW_HmJsX

DATASET:

Overall **460 images** composed our dataset. The dataset is prepared **based on mean value** of each spectral for each images.

Since VNIR and SWIR magnitudes were varied, so **we normalized each images individually in the range of zero and one**. And **still there was a sharp change in 160 and 160+1 spectral**. So **we transformed the values of VNIR (we selected VNIR because its values was variated) as much as the difference of each particular 160 and 160+1 spectral of that image by multiplying that difference to all its values**. Following is the result of merging images.

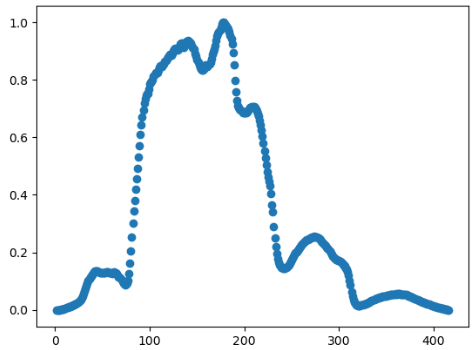


Figure 2: 038367

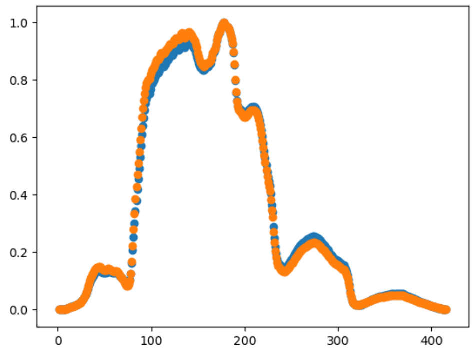


Figure 3: 038367 and 038368

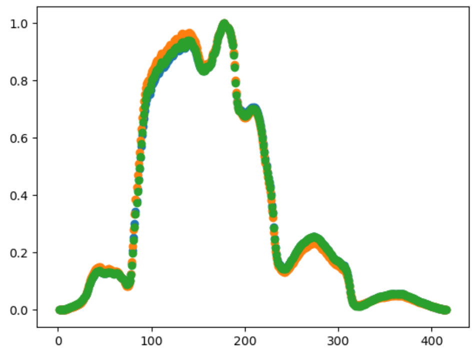


Figure 5: 038367, 038368 and 038369

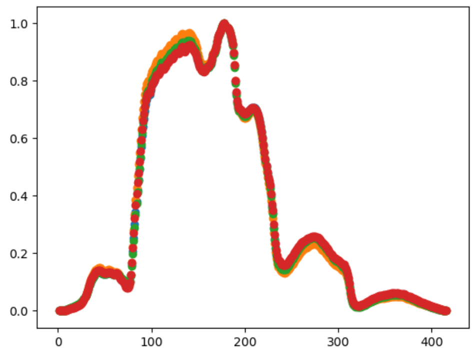


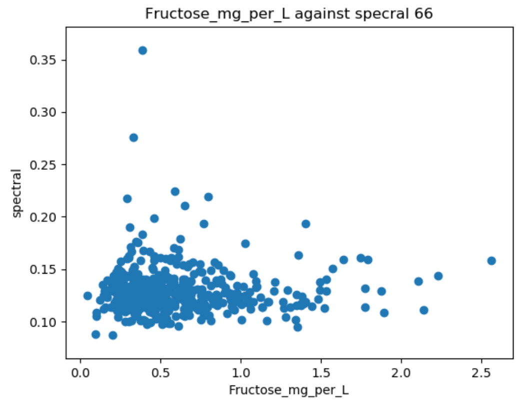
Figure 4: 038367, 038368, 038369 and 038370

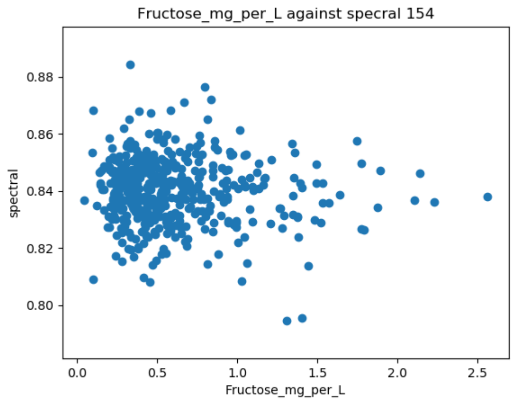
Regarding the other dataset (mapping SWIR and VNIR based on coordinates of images) that we previously discussed in the meeting, we didn't proceeded further for the moment though the coding is ready. We decided to work on this one (based on means) because, it's smaller in size so it is easier to be interpreted and handled.

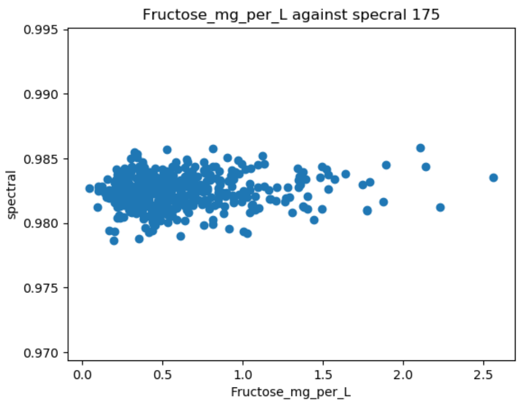
ANALYSING OUR DATA

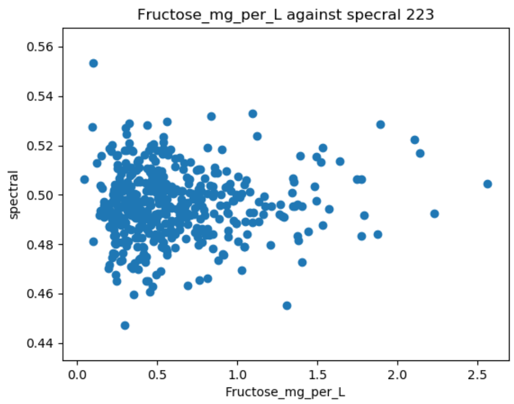
Only keep single target variable against each spectral variable in order to have a look on the pattern and possibly keep and start with the best spectral.

Target values against spectral values are plotted as following:







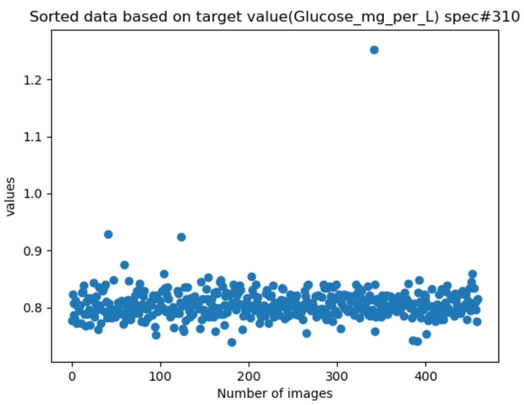


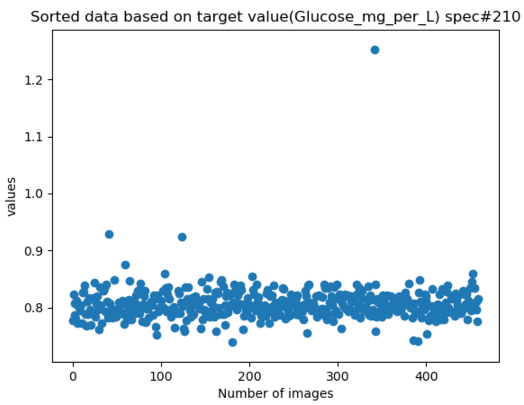
You can take a look on the rest of target value against spectra in below link

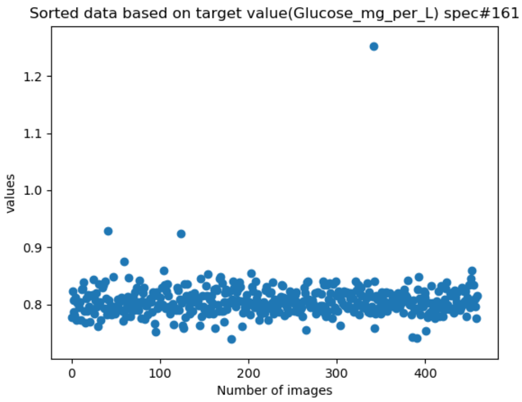
https://drive.google.com/open?id=150PLxesolx5qVrpvLGDD2S-L56hzNk2U

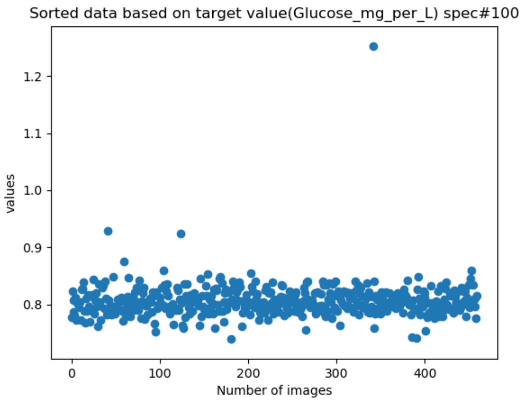
The represented data is not looking to give slope, so I would like to keep the target variable sorted in ascending order. So that I can see how each spectral behave.

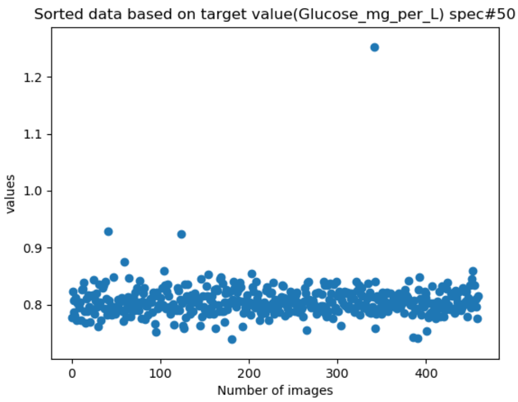
In below scatter plots, the spectral values of sorted images based on target value “glucose” is plotted. And **expected to increase or decrease, but actually they are randomly distributed thorough x axis**.

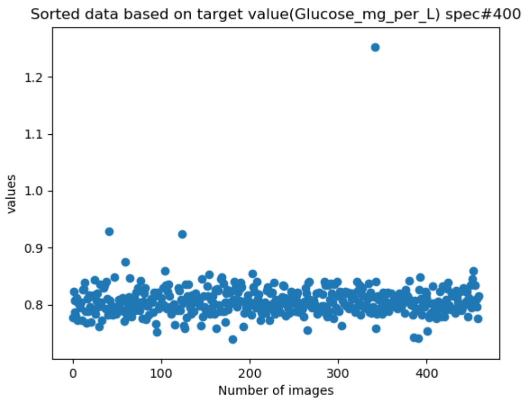












None of the spectral values gives a slope to be able to estimate y values based on given x. therefore expected that the linear regression doesn't give a solid result.

LINEAR MODELS:

LIBRARIRIES AND STEPS TAKEN

The most simple linear model is based on the equation of a rect with the two parameters “a and b” to characterize it. These parameters will be calculated so as to make the sum of squared residuals as small as possible.

y = a*x + c

In this expression, x is the training set, y is the target, b is the slope, and c is the intercept of the rect represented by the model.

- Using scikit-learn library, linear_model module is imported and the constructor ”LinearRegression()” is created to build our predictive model.
- Training data and testing data is 80% and 20 each respectively which randomly will be selected by importing “train_test_split” from sklearn.model_selection module. Using train_test_split() function and passing dataset, target values and percentage of test_ size as parameters, we can split our dataset. After that, fit() function of linear model called with x_train and y_train parameters.
- Now our dataset is trained and the coefficient values can be calculated by accessing coef_ attribute of our LinearRegression() constructor.
- By applying the prediction model on test dataset, we can get a series of targets which can be compared with the values actually observed. Prediction is done by calling predict() function and passing x_test dataset.
- A good indicator of how predicted values are is the variance. The closer the variance is to 1 the more perfect the prediction is. Its done by calling score() function and passing x_test, y_test as parameters.

Methods used from scikit learn:

`fit`(self, X, y[, sample_weight])	Fit linear model.
`get_params`(self[, deep])	Get parameters for this estimator.
`predict`(self, X)	Predict using the linear model.
`score`(self, X, y[, sample_weight])	Return the coefficient of determination R^2 of the prediction.
`set_params`(self, **params)	Set the parameters of this estimator.

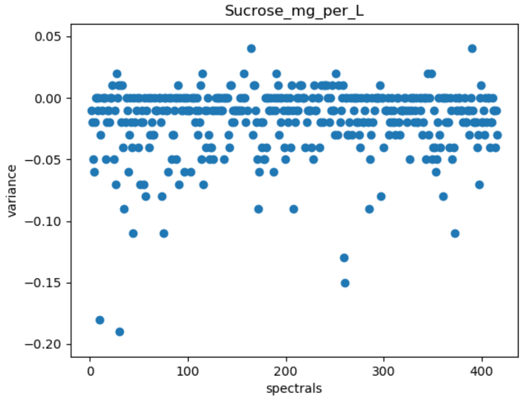
Note: The coefficient R^2 is defined as (1 - u/v), where u is the residual sum of squares ((y_true - y_pred) ** 2).sum() and v is the total sum of squares ((y_true - y_true.mean()) ** 2).sum(). The best possible score is 1.0 and it can be negative (because the model can be arbitrarily worse). A constant model that always predicts the expected value of y, disregarding the input features, would get a R^2 score of 0.0.

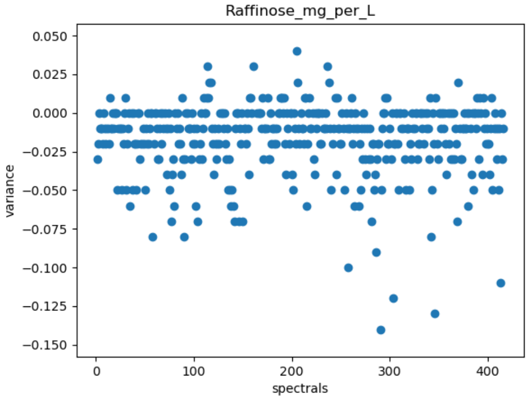
ORDINARY LEAST SQUARE LINEAR REGRESSION

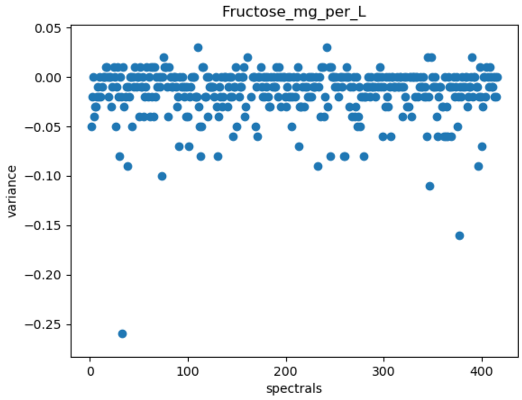
I performed **stepwise regression and for each target value against each spectra separately**.

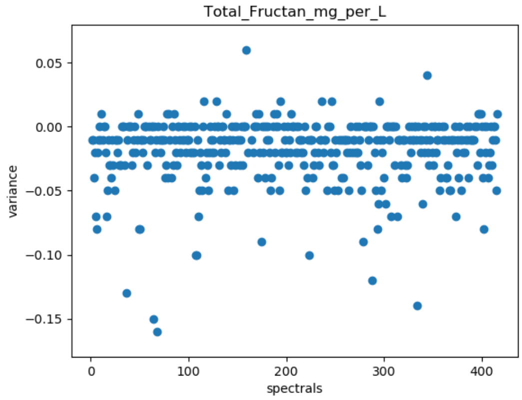
The coefficient R^2 is defined as (1 - u/v), where u is the residual sum of squares ((y_true - y_pred) ** 2).sum() and v is the total sum of squares ((y_true - y_true.mean()) ** 2).sum(). **The best possible score is 1.0 and it can be negative (because the model can be arbitrarily worse).**

The result seems not to be useful at all because many predicted values are below zero.







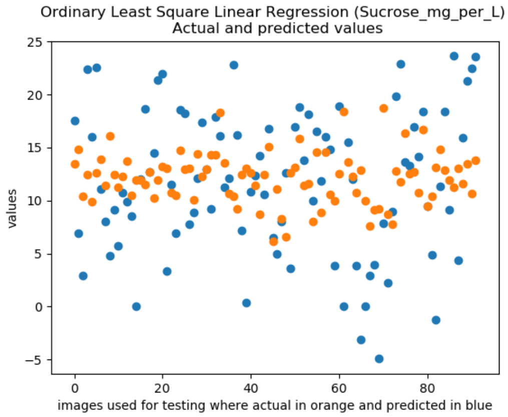


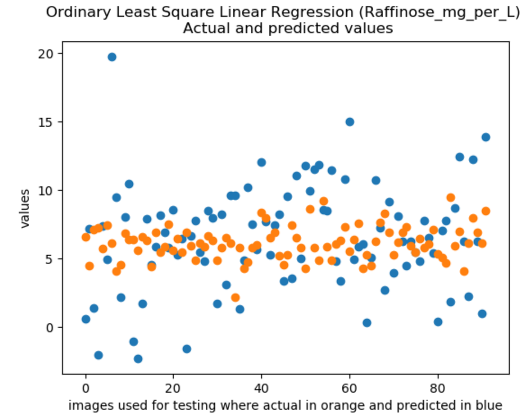
Summary table for finding best channels based on high variance	coefficient R^2	Better channels are
Glucose_mg_per_L	0.054376	160, 165, 237
Fructose_mg_per_L	0.047747	163, 252
Sucrose_mg_per_L	0.063737	162
Raffinose_mg_per_L	0.0415	241
1_Kestose_mg_per_L	0.059166	160
Maltose_mg_per_L	0.042248	160, 164
Nystose_mg_per_L	0.065962	163
1_1_1_Kestopentaose_mg_per_L	0.084035	160
Total_Fructan_mg_per_L	0.057786	159

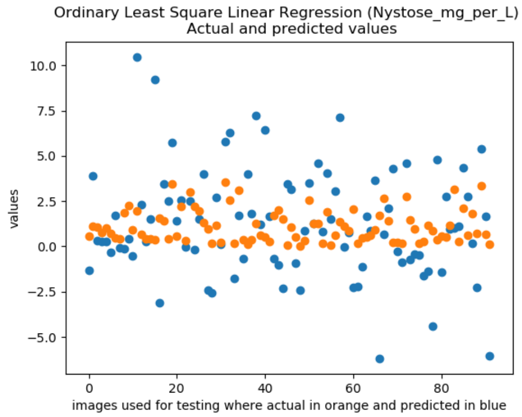
The coefficient R^2 are very low.

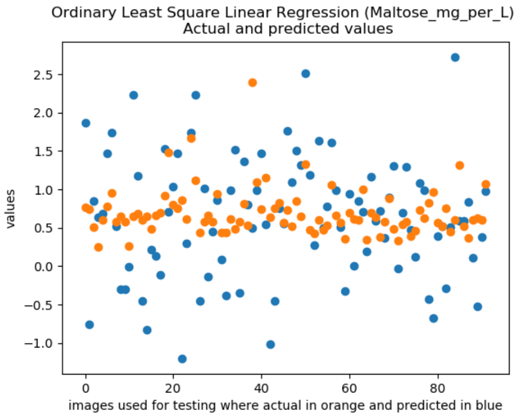
If we train **all spectra together with each single target variable separately, the result is not useful since still coefficient R^2 is very low even for the best channels and many predicted values are below zero**.

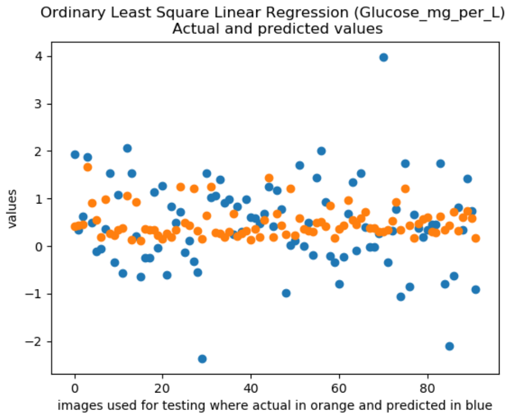
In the following scatter plot, predicted and actual values of test dataset are represented.

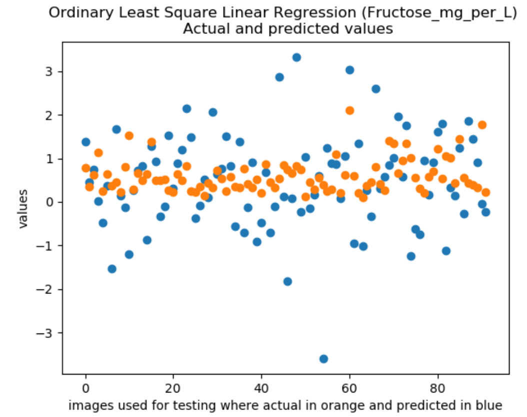

Ordinary Least Square Linear Regression (Sucrose_mg_per_L)
Actual and predicted values
values
images used for testing where actual in orange and predicted in blue

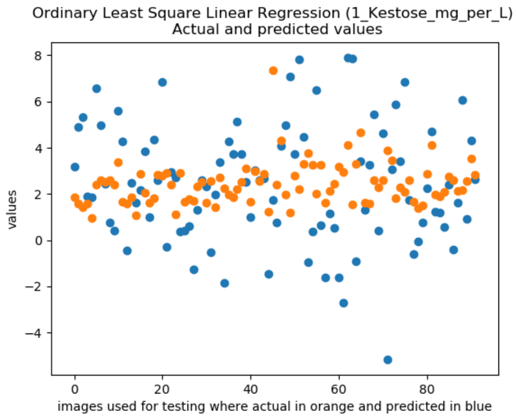

Ordinary Least Square Linear Regression (Raffinose_mg_per_L)
Actual and predicted values
values
images used for testing where actual in orange and predicted in blue

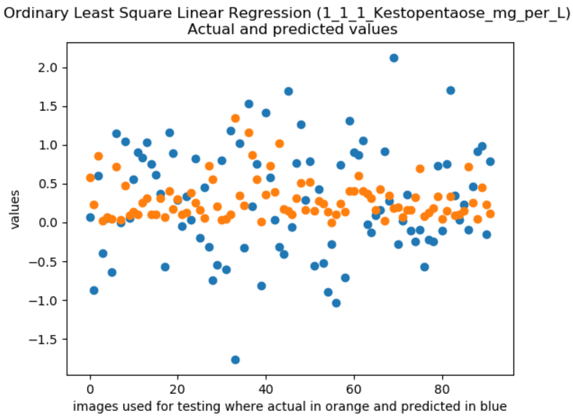

Ordinary Least Square Linear Regression (Nystose_mg_per_L)
Actual and predicted values
values
images used for testing where actual in orange and predicted in blue

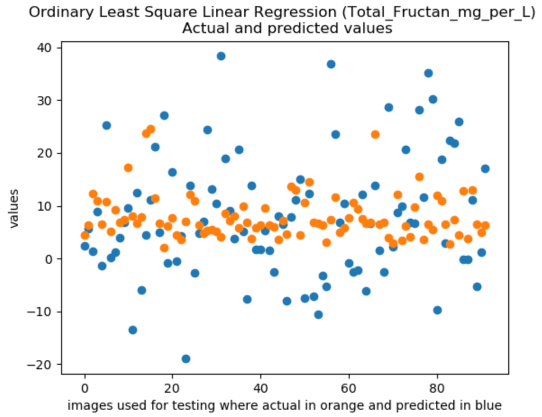

Ordinary Least Square Linear Regression (Maltose_mg_per_L)
Actual and predicted values
values
images used for testing where actual in orange and predicted in blue


Ordinary Least Square Linear Regression (Glucose_mg_per_L)
Actual and predicted values
values
images used for testing where actual in orange and predicted in blue


Ordinary Least Square Linear Regression (Fructose_mg_per_L)
Actual and predicted values
values
images used for testing where actual in orange and predicted in blue


Ordinary Least Square Linear Regression (1_Kestose_mg_per_L)
Actual and predicted values
values
images used for testing where actual in orange and predicted in blue

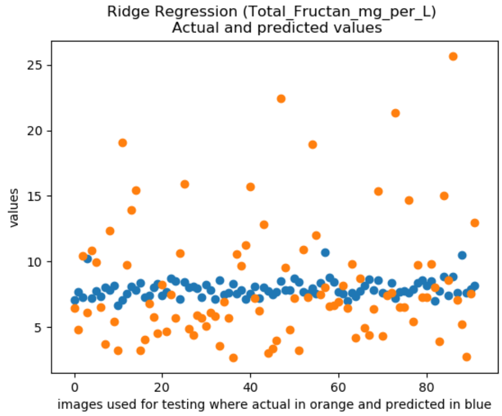

Ordinary Least Square Linear Regression (1_1_1_Kestopentaose_mg_per_L)
Actual and predicted values
values
images used for testing where actual in orange and predicted in blue

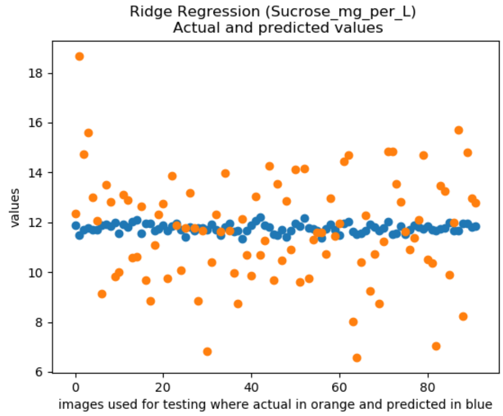


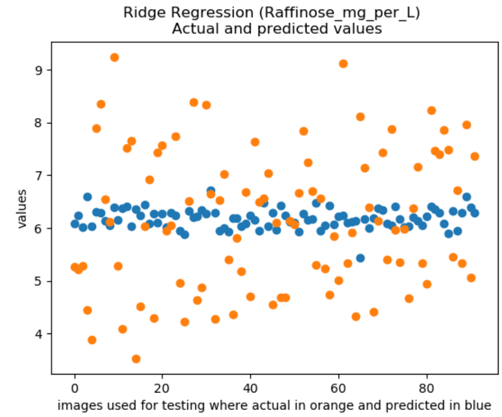
Still the **coefficient R^2 is around -5 and -6**, therefore, it cannot be used at all.

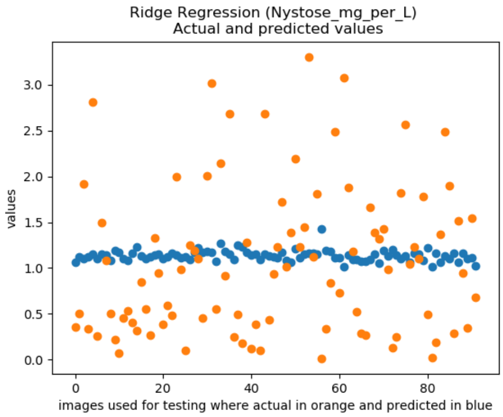
RIDGE REGRESSION

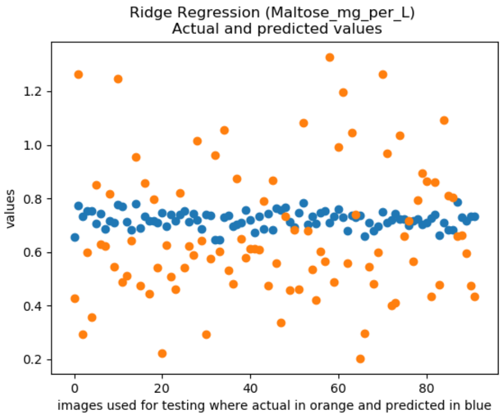
In the following scatter plots, predicted and actual values of test dataset are represented.

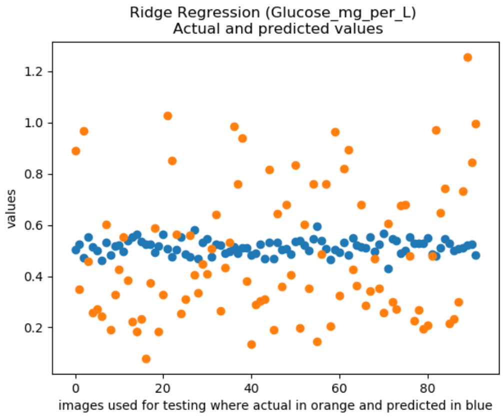


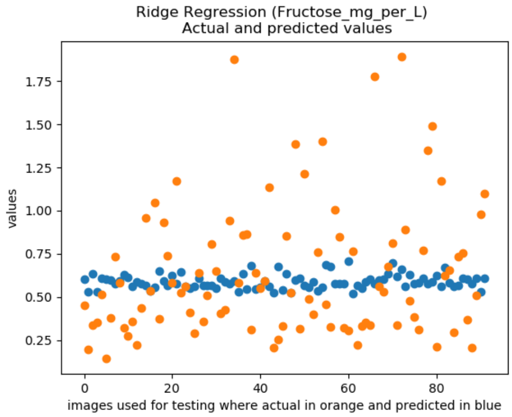


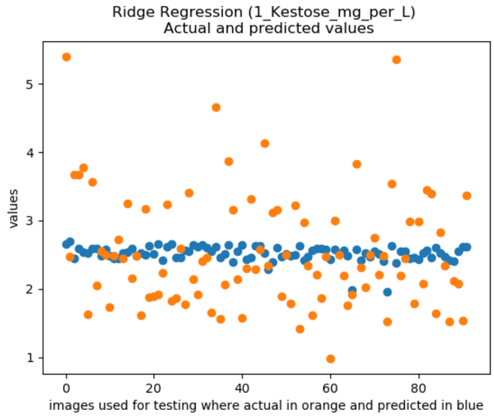


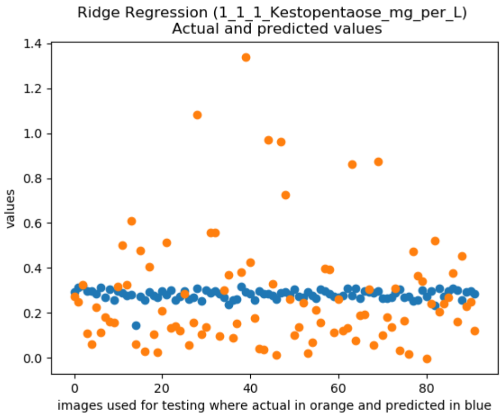










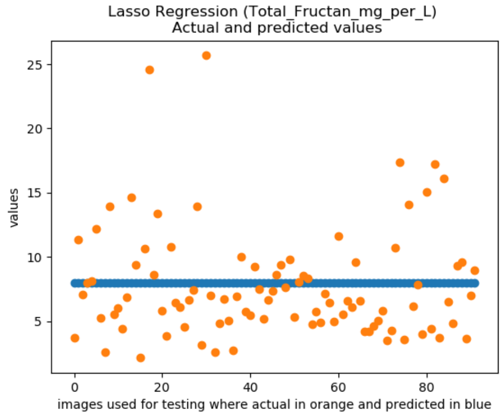


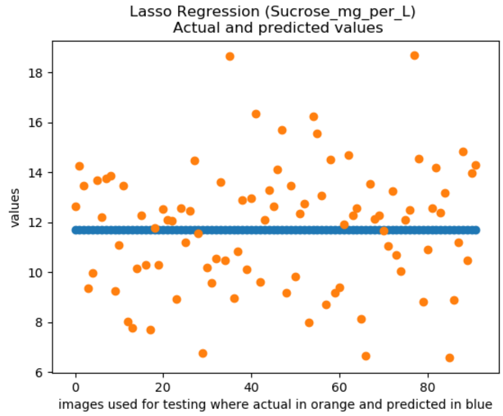
Still the **coefficient R^2 is around -0.02 and -0.06**, though it's far better than Ordinary Least Square Regression but still cannot be used.

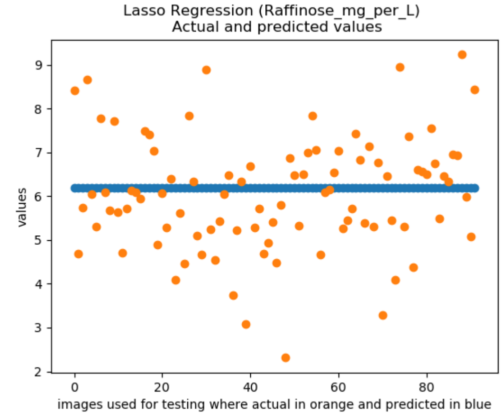
Applying different penalty terms also do not bring significant changes in the results.

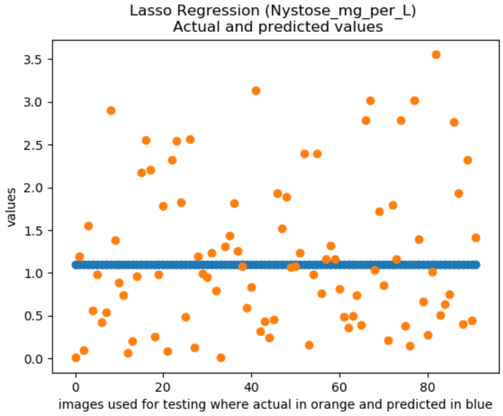
LEAST ABSOLUTE SHRINKAGE AND SELECTION OPERATOR (LASSO)

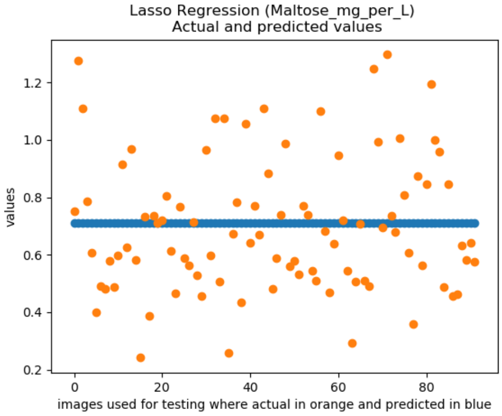
In the following scatter plots, predicted and actual values of test dataset are represented.

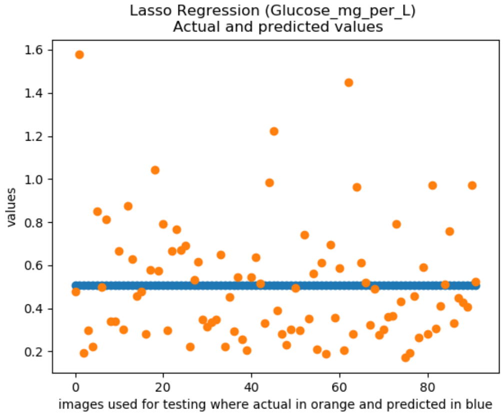


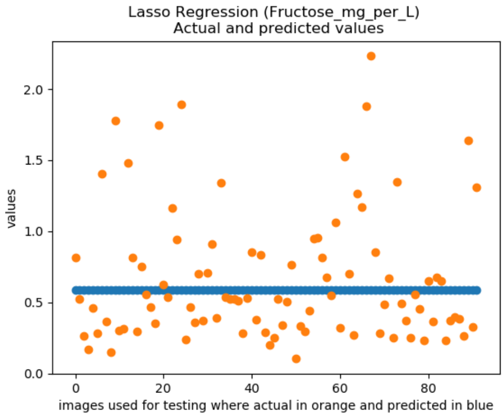


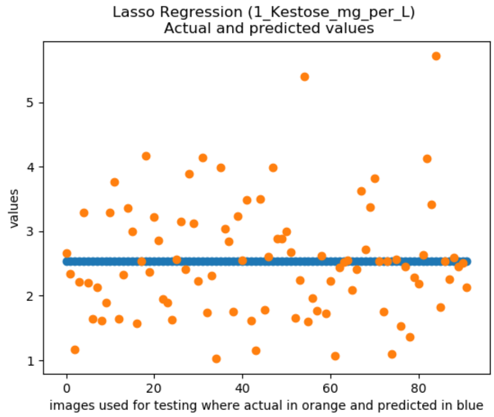


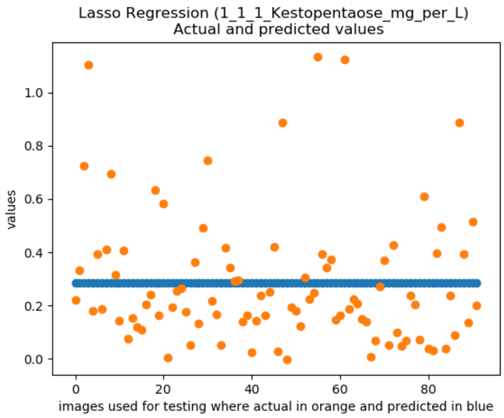










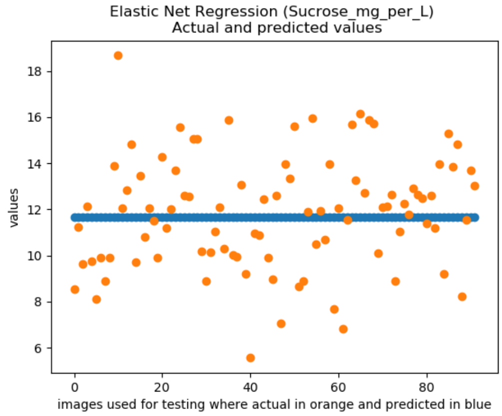


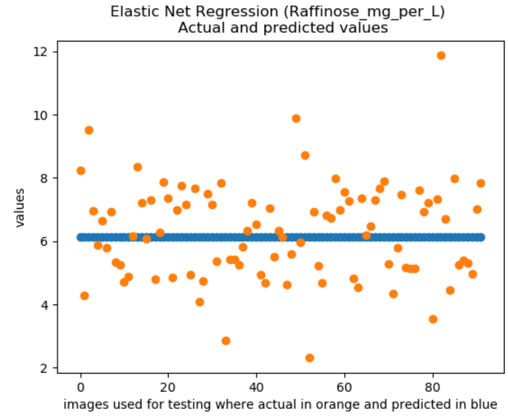
Still the **coefficient R^2 is around -0.005 and -0.01**, though it's far better than Ordinary Least Square Regression and also somewhat better that Ridge regression but still cannot be used.

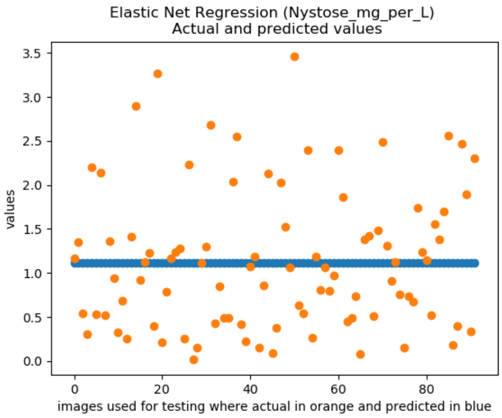
Applying different penalty terms also do not bring significant changes in the results (slope is already zero).

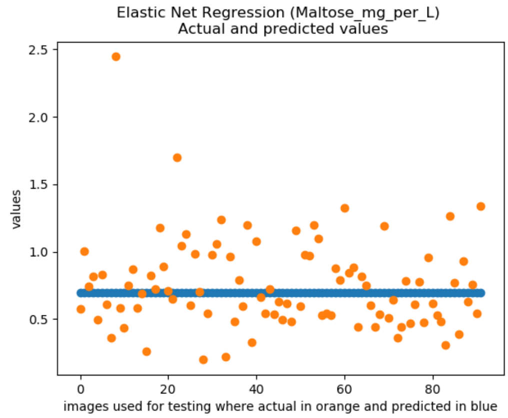
ELASTIC NET

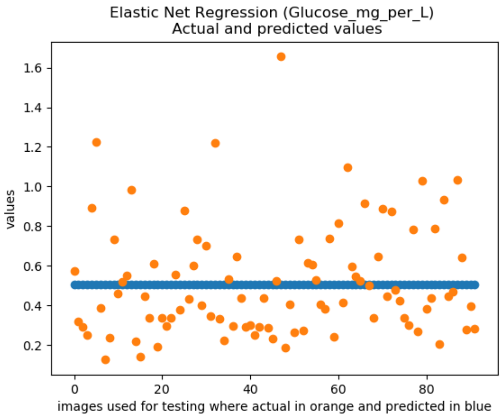
In the following scatter plots, predicted and actual values of test dataset are represented.

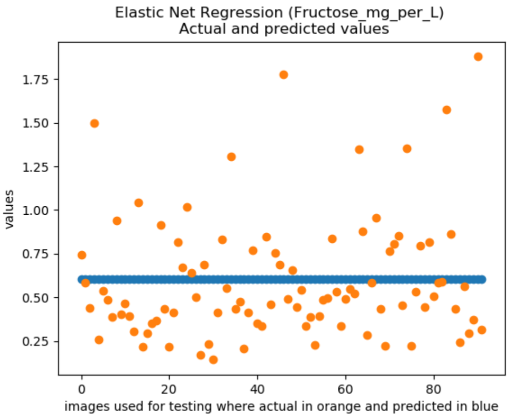


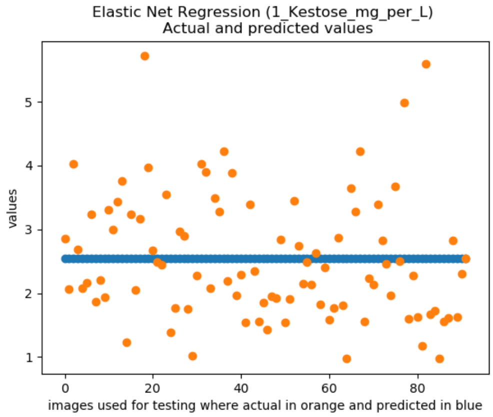


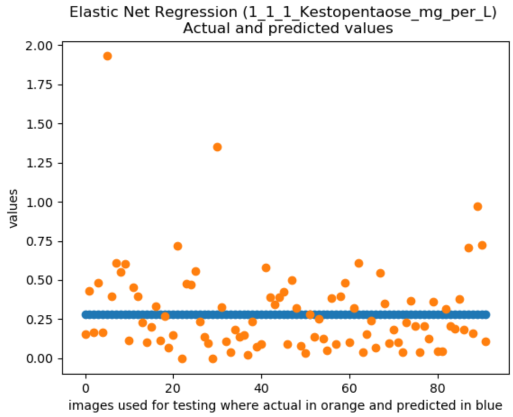


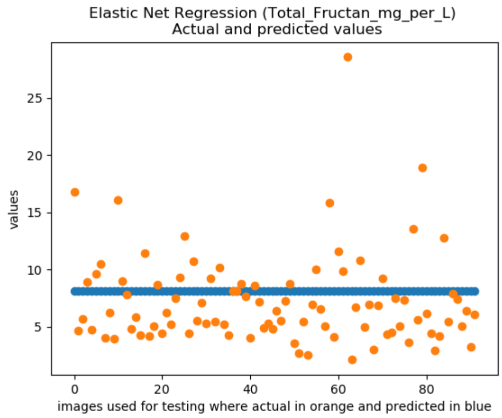












Still the **coefficient R^2 is around -0.0002 and -0.04**, though it's far better than Ordinary Least Square Regression and also somewhat better that Ridge regression and Lasso but still cannot be used.

Applying different penalty terms also do not bring significant changes in the results (slope is already zero).

PROJECT STATUS

Tasks	Expected Date	Done	Assigned to
Literature Research	12.Dec.2019	Yes	All team members
Segmentation	01.Mar.2020	Yes	Saied
Management of Data (downloading images and resizing VNIR images)	05.Mar.2020	Yes	Ramkishore
Merging images	10.Mar.2020	Yes	Saied
Preparing Dataset	15.Mar.2020	Yes	Saied
OLSLR, Ridge, Lasso, Elastic Net	5.Apr.2020	Yes	Saied
SVR	07.Apr.2020		Raman
PLSR	07.Apr.2020		Amit, Ramkishore,
RFR	07.Apr.2020		Devish, Sudheer
Comparing Algorithms	…		Saied, Amit, Raman
Documentation	…		Amit, Raman, Ramkishore, Sudheer, Devish