Problem 1 – linear regression (30%)

15% credit: Compute the OLS solution on the training set and apply it to the test set. Your results should include the R2 on test data, and a scatter plot of ground truth vs predictions. Please discuss your findings.

Following conversion of the according csv files to pandas dataframes and separating them into the according training and test sets, I used sklearn’s linear_model and sklearn.linear_models’s LinearRegression package to compute the OLS solution of the training set through use of the fit() function. The matrix for the training set was 50 rows by 26 columns, and accordingly, the solution returned a matrix containing 26 weights for each feature in the training set.

Sklearn’s linear regression model essentially serves to abstract the process of OLS, but essentially, when we solve for the solution using the training set, the following equation is used to determine the weights of the predictor, or the solution: $(X^T X)^{-1} X^T Y$. In this case, we use the training X matrix of predictors (x1.csv) and the training Y of the dependent variable (y1.csv) as the according matrices X and Y.

As such, we get the following OLS solution and R2: (note: R2 was calculated using the score() function from sklearn’s linear model module. In this case, score() is analogous to R2, as documentation states that it is calculated using the R2 formula) for the training set:

```
Regression score for OLS: 0.9606822396194843

Regression Model Coefficients/Weights [ 0.01406343 -0.00738955 -0.00339786
-0.08081296 0.01664003 -0.00393456 0.03066151 0.22738778 0.0151364 0.04779523
0.03128287 -0.53692256 -0.7335415 -0.00242663 -0.03690684 -0.03767773 0.01942357
-0.21207741 -0.27932138 0.00520287 0. 0.04130665 0.02241085 -0.03904865
-0.03296109 0.00116588]
```

Discussion of findings:

Analysis of the weights matrix:

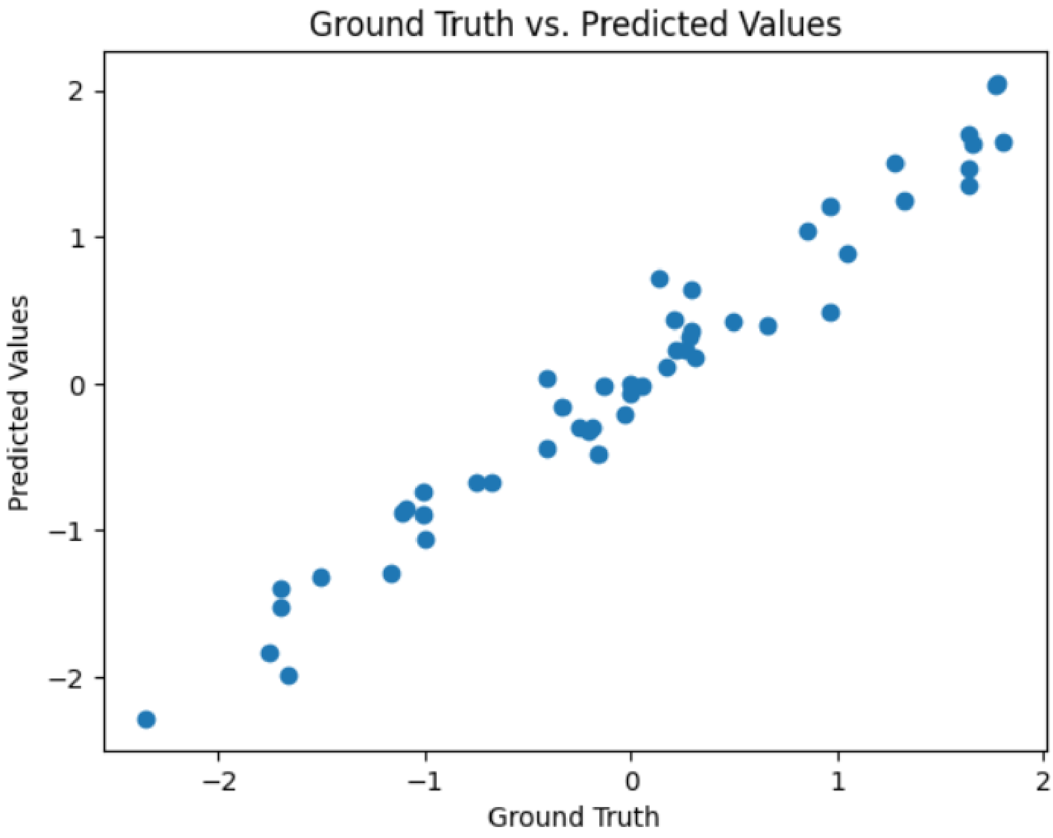
- Based on the weight matrix above, it is interesting to note that the maximum regression coefficient is 0.227, which measures the strength of the 8th predictor or feature in the dataset. This is especially notable because the average regression coefficient in this weight matrix is -0.05899, which is very close to 0. It is also interesting to note that many of the predictors don’t have a lot of predictive strength, and that there is one predictor that has no predictive power at all over the output, as it has a weight of 0.

- This observation brings up the question of the curse of dimensionality, as there are some features that have little to no influence as a predictor.

Analysis of the R^2 Value:

- For the OLS solution using the already existing training set, we get an R2 value of 0.96068, which tells us the matrix Xw explains 96.068% of the variance in Y. This is a very high R2 value; thus we can infer that this model explains the data relatively well. However, it is important to note that this value does not indicate whether the regression model is adequate or not, as R2 does not measure the predictive error.
- In general, the R2 value aligns with the below scatter plot of ground truth values vs. Predicted values, as it appears that the model created by OLS reduces error and generally makes predictions that are close to the predicted value.

Applying the solution to the test data gives us the following predictions, and when graphed against the ground truth values, we get the following (i.e. the actual values of Y):



Discussion of findings:

Analysis of the Scatter Plot:

- The scatter plot of the predictions plotted against the ground truth values appears to generally match expectations, especially following interpretation of the R2 value. The more linear the model is, the more we can expect that the model correctly predicts the ground truth values, as if it is a perfect prediction, the ground truth and predicted value should be the same value. Thus, the ideal that we aim to achieve is a linear function that models similarly to y=x. Based on the graph above, it appears that the model developed using OLS had a tendency to predict values higher than that of the ground truth, but overall, the model either correctly predicted the ground truth values or closely predicted the value.

15% credit: Merge the training and test sets ([x1; x2], [y1; y2]) and re-split them to generate new training and test partitions. Then repeat part a. multiple times and discuss the distribution of R2 values from split to split.

First, I merged the training and test sets, which gave me the following dimensions:

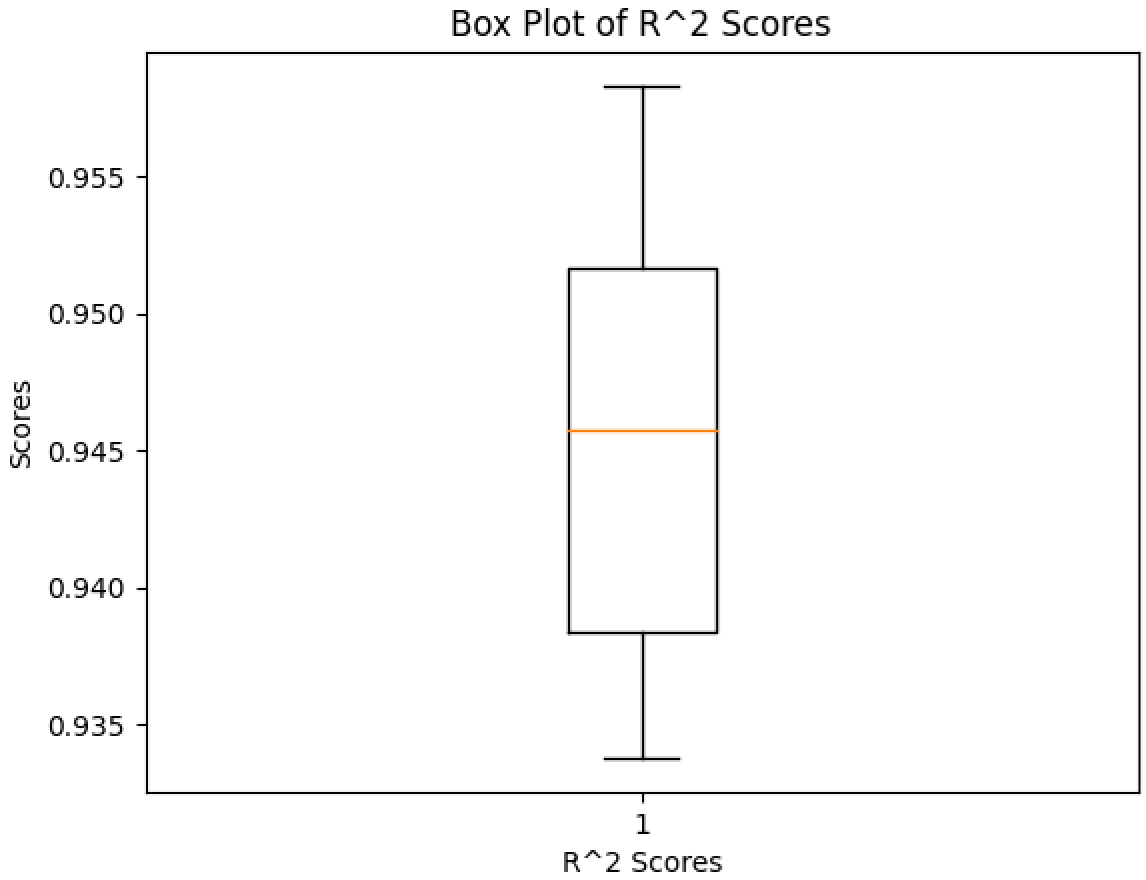
- `Training set after concatenation: (100, 26)`
- `Testing set after concatenation: (100, 1)`

This is expected, as when inspecting the csv files manually, I can see that csv files x1, x2, y1, and y2 have 50 rows, with the y_.csv files having 1 column and the x_.csv files having 26 columns. Thus, I proceed onto the splitting of training and testing partitions.

For this part of the problem, I interpreted this as resplitting the training and testing partitions to be the same size as before (i.e. 50/50) but shuffling the values around so we get unique testing/training sets.This split was repeated 10 times, and for each split, the OLS solution on the training set was calculated, then applied to the test set. For every split, a scatter plot and R^2 value was generated. R^2 values were tracked and used to generate a Box Plot of scores.

For the sake of succinctness and length, as many scatter plots and R^2 values were generated, the results are shown on the last pages of this report, as the instructions say to repeat part a. Multiple times. I interpreted this as providing a scatter plot for every split. The summarized report will be discussed here.

When Interpreting the distribution of R^2 values from split to split, a box plot was generated in order to better visualize the distribution of values.



Further inspection on the R^2 values gives us the following:

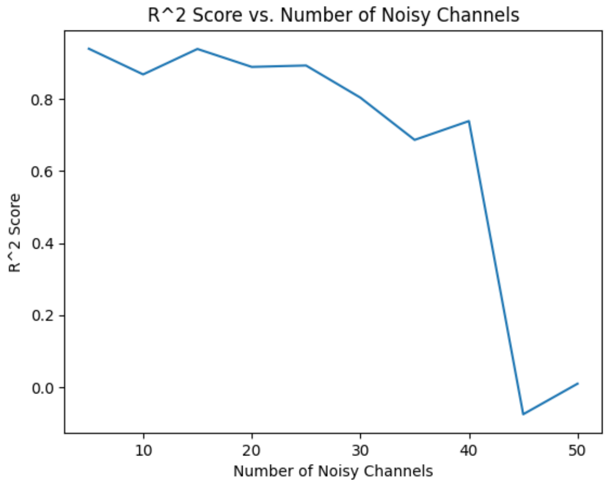
- Average R^2: 0.945
- Standard Deviation: 0.0089
- Minimum R^2: 0.933
- Maximum R^2: 0.958
- First Quartile: 0.938
- Second Quartile: 0.946
- Third Quartile: 0.952

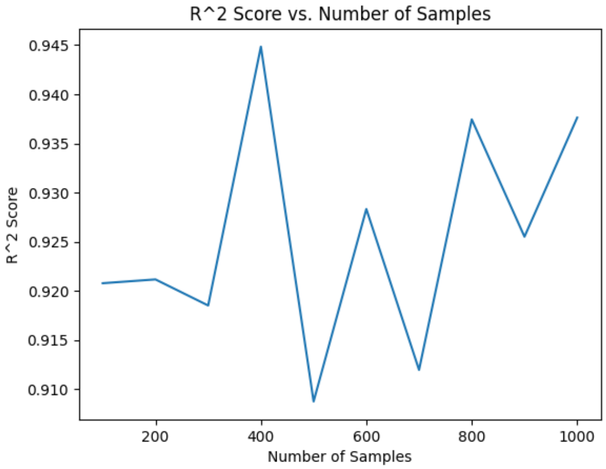
Discussion of findings:

- In general, following the merging of the training and test sets, we get the above distribution of R2 values from split to split. In general, the average R2 value is relatively high, meaning that on average, the models performed relatively well against the validation sets for each split. Even the minimum R2 value was still above 0.9, which leads me to believe that the OLS model overall performs well and is able to explain a large portion of variance in the data.
- On average, when applying OLS, the model explains 94.5% of the variance ub the dataset, and the small standard deviation tells me that in general, the R^2 values are

relatively closer to the mean. This is important because this means that regardless of the splitting of the data, the R^2 values for a model are relatively close to the mean and there isn't much spread.

5% extra credit: Generate your own data by converting the code in Appendix I to Python. Then, experiment with a different number of noisy channels (dd) and samples (n). How does R2 change as a function of these two parameters? Discuss this and other findings.





Discussion of findings:

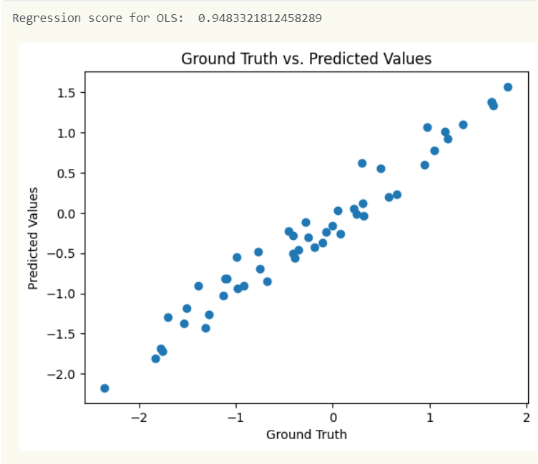
Noisy Channels:

- As Noisy Channels increase, the R^2 Score appears to gradually decrease until it reaches a noisy channel of 45. This tells me that as we increase noisy channels, more complexity and noise is introduced to the data, making it difficult for OLS to generalize predictions. The sudden drop in R^2 where the noisy channel variable equals to around 45 shows that at a certain point, the model can no longer make meaningful predictions that can explain the variance in the data.
- This makes sense because as more noise is added to the data, the predictions become non-linear, and the features become less predictive of response. In terms of linear regression, when we introduce noise, this can lead to overfitting the model to the training set, which explains the reduced performance when checking the model against the validation set. At this point, the weights become very sensitive to the noise, resulting in decreased performance. Because OLS is a closed form solution, choosing a different model might be more ideal in the case of increased weights. For example, gradient descent paired with a regularization technique may allow for better results that stand up against noisy channels.

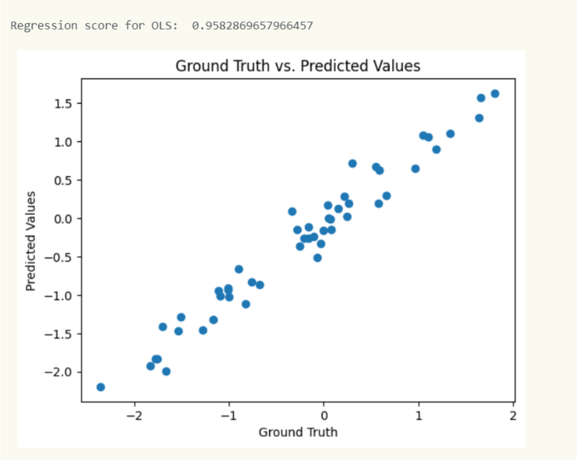
Samples:

- There appears to be no trend regarding R^2 and sample size, as the R^2 value appears to be independent of the sample size. When leaving all other variables untouched, and increasing the sample size, the performance of the model fluctuates between R2 values of 0.91 and 0.945. This makes sense, as though more samples can allow for us to get a better understanding of the real world data, in the end, what determines a high R^2 value depends on how the model holds up against the validation set. If we get data that doesn't have a lot of variability, and is almost perfectly linear, it doesn't matter how many samples we end up having, as in this situation the R^2 value will be high nonetheless.

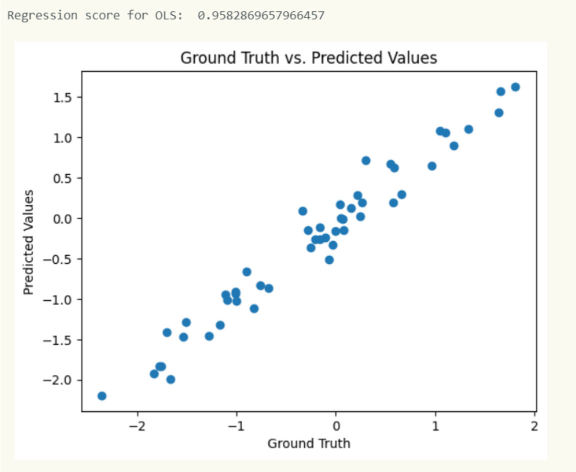
Graphs for Part B.



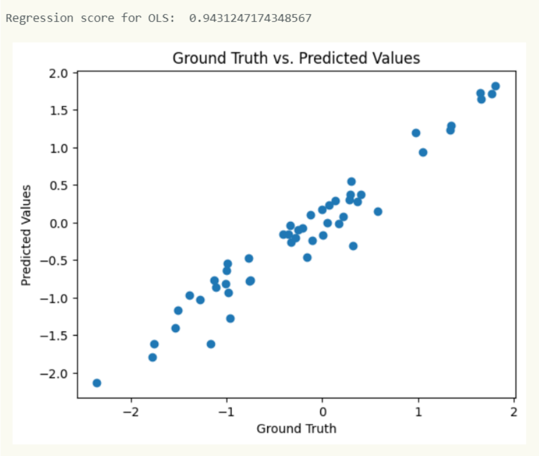
```
Regression score for OLS:  0.9483321812458289
```



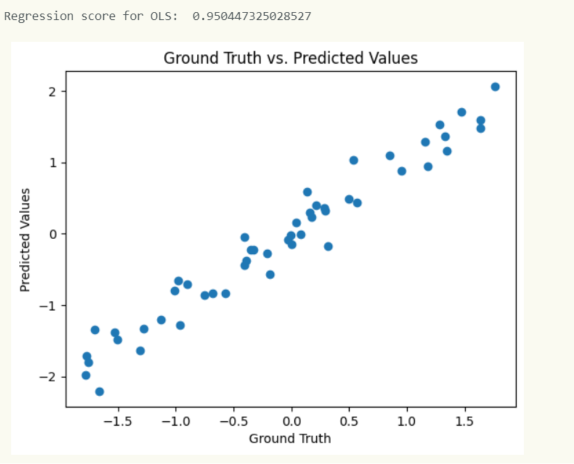
```
Regression score for OLS:  0.9582869657966457
```



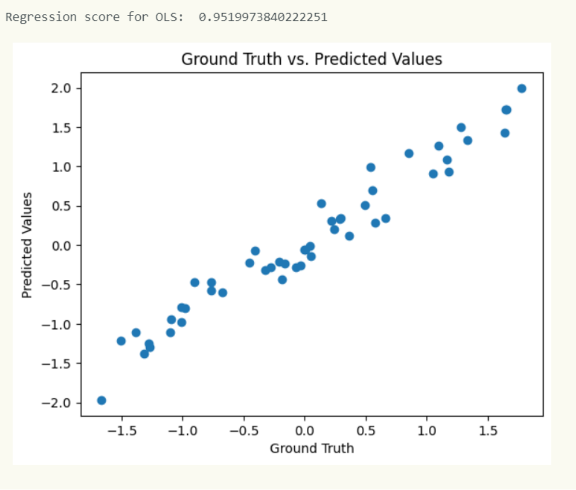
```
Regression score for OLS:  0.9582869657966457
```



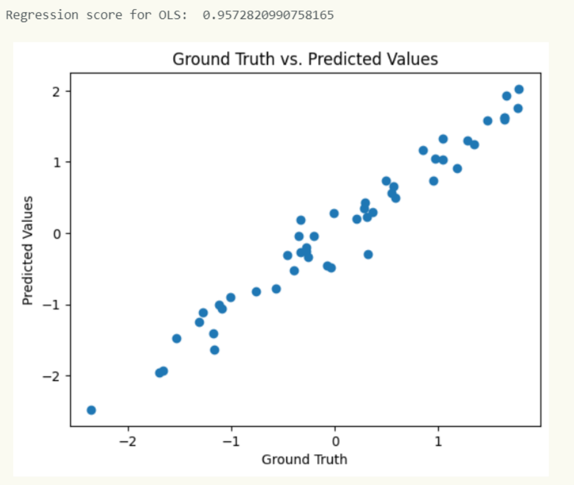
```
Regression score for OLS:  0.9431247174348567
```



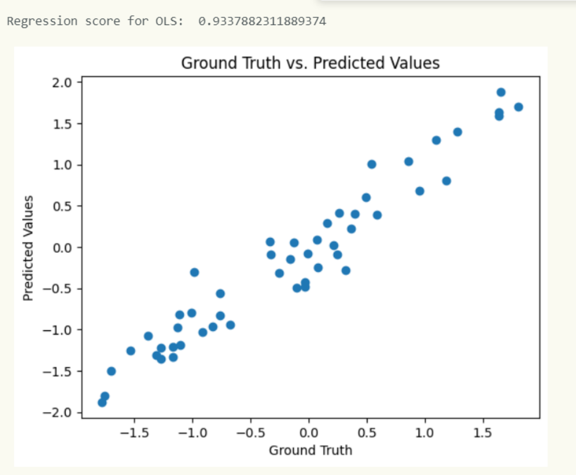
```
Regression score for OLS:  0.950447325028527
```



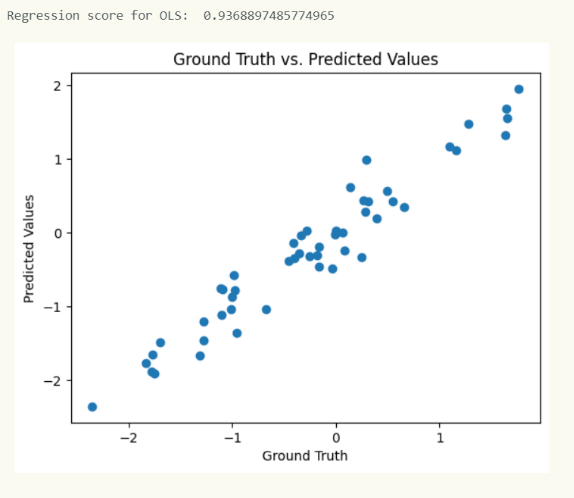
```
Regression score for OLS:  0.9519973840222251
```



```
Regression score for OLS:  0.9572820990758165
```



```
Regression score for OLS:  0.9337882311889374
```



```
Regression score for OLS:  0.9368897485774965
```