Mytoxin prediction using Hyperspectral Data

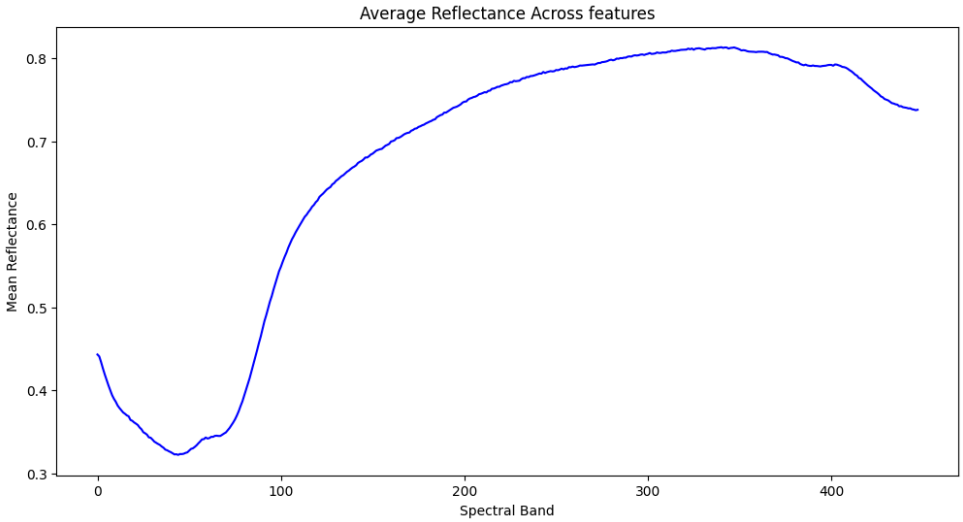
Exploratory data analysis

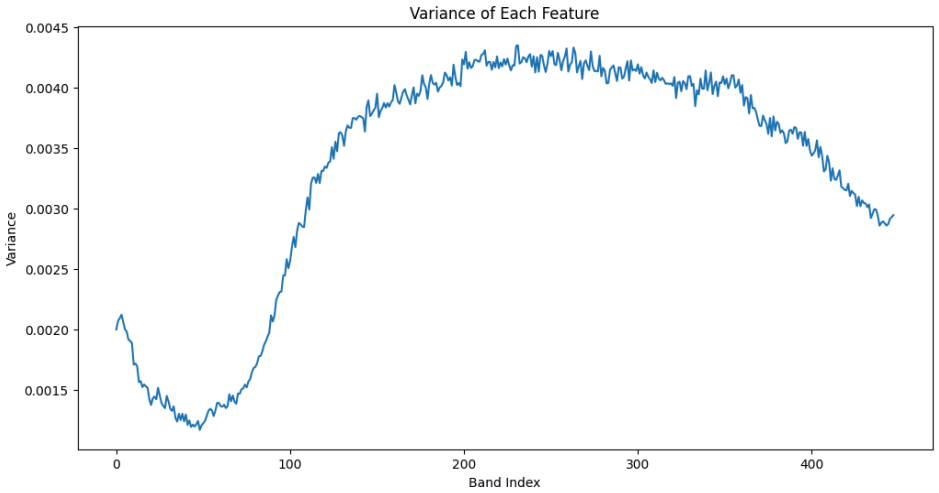
Although the dataset doesn't have any meta-data, and we don't know what the features correspond to, but am expecting that the features represent the wavelengths in increasing order

Exploring the distribution of the features:

- The dataset contained 500 rows, where each row had 450 entries.
- I looked at how the values of each feature (column) was spread.

- I took the mean and variance of all the features, and found an interesting pattern



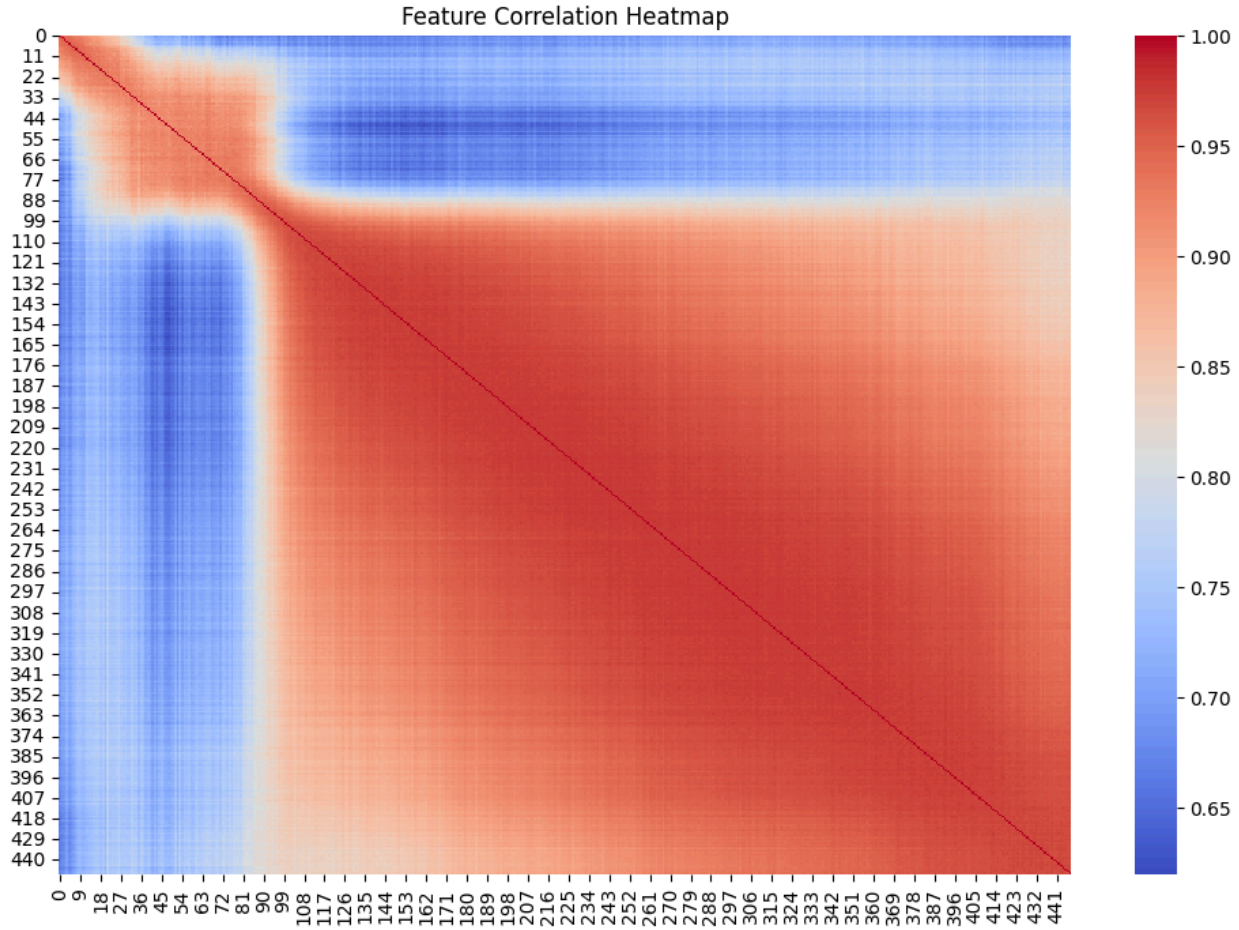


In the above graphs, we can see that both the graphs follow the same pattern. The variance graph has more noise which is obvious as it uses square, and is more noisy.

How are the features related to each other

The dataset had 448 features (removing the first column and the last variable). I looked into how the features are correlated to each other. I noticed that the correlation was more in the feature

with index (>90) with its neighboring features.

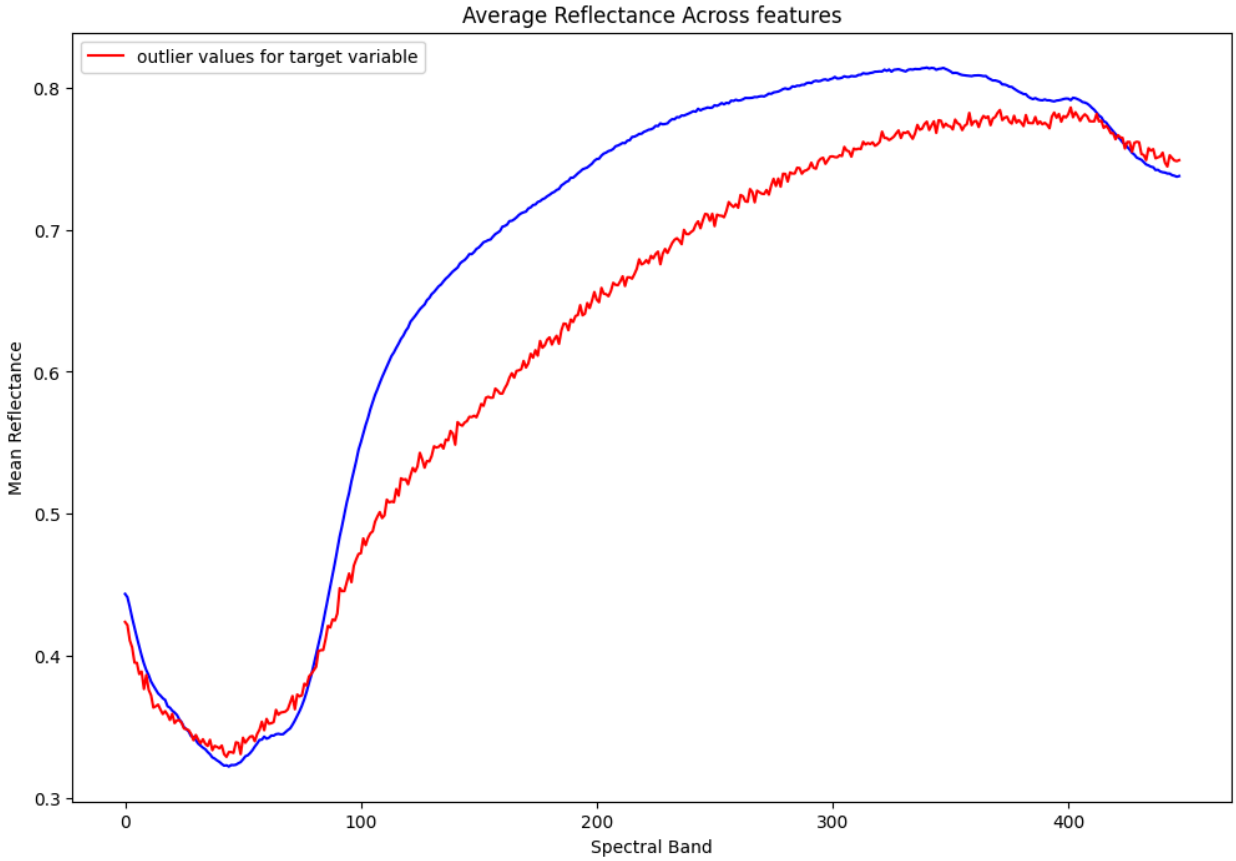


How's the target variable distributed?

The target variable represents deoxynivalenol (DON) concentration. is a mycotoxin which commonly infect grains like corn, wheat, and barley. It is a significant concern in food safety because it can be toxic to both humans and animals if consumed in high amounts. It was interesting how the max value for this variable is as high as 131000, where the safe measure for human consumption is < 750. The distribution was left skewed.

Anomalies in the target variable and how I treated them: Although 95% of the values in the target variable were under 8920, there were some outliers present in the target variable which needed to be treated. There were various ways to treat them, I could have used IQR and removed those samples. But when I went through the distribution of how features are distributed for the samples having these anomalies, and the rest of the samples, I found out that the anomalies had different distribution, and hence I didn't want to remove them as that will result in loss of information. So, I capped the values at a value of 20,000. The image below compares this distribution of samples

with the outlier value of DON concentration, and other samples.

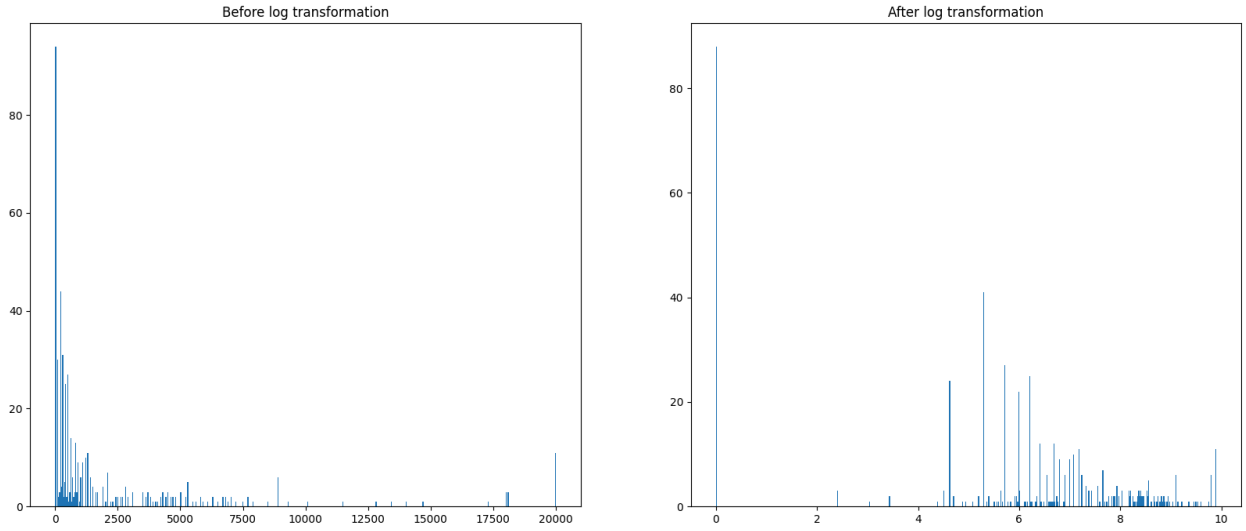


How the features are correlated with the target variable

I looked at how each feature was correlated with the target variable. The graph below shows the distribution. F

Preprocessing raw data

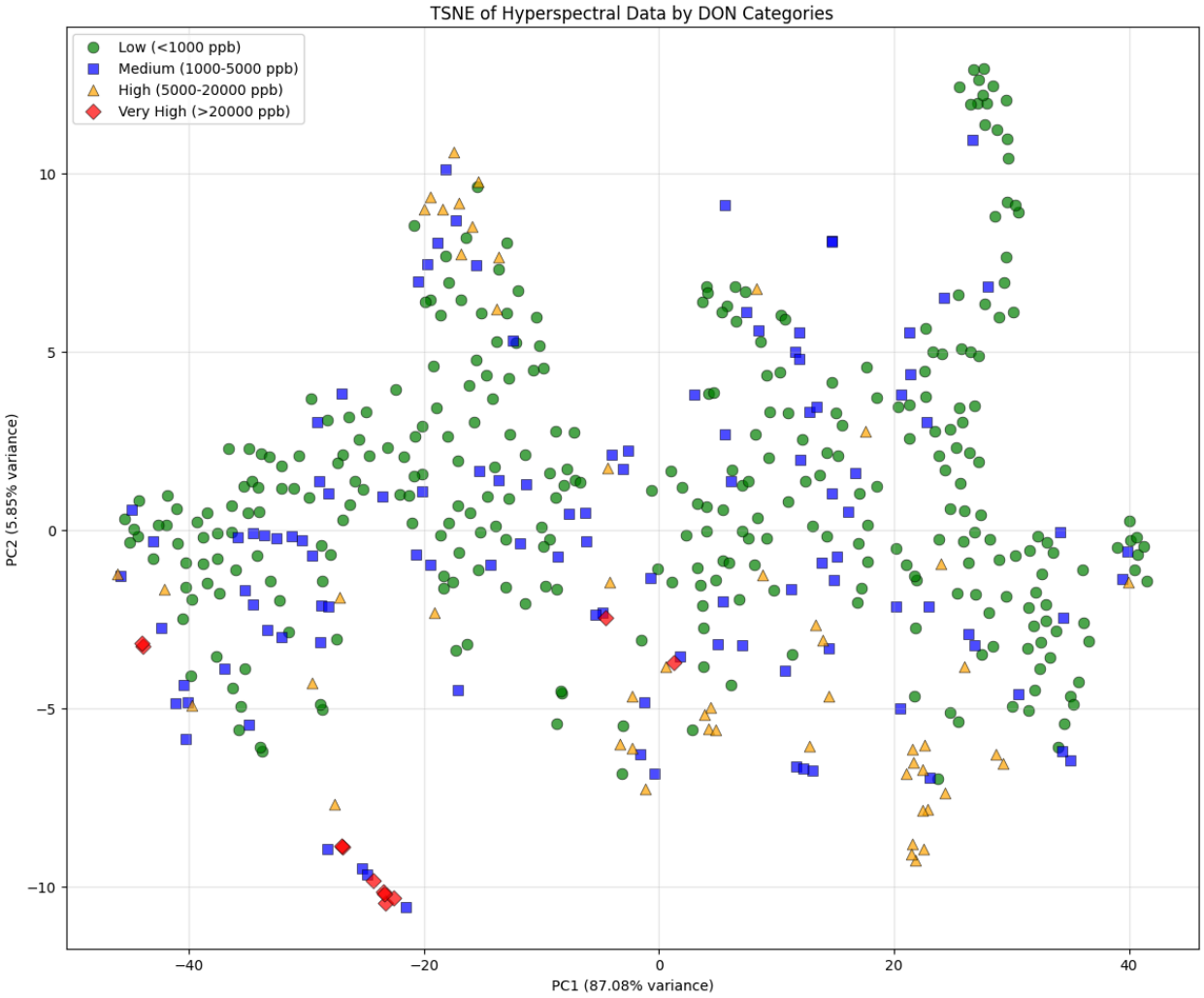
For the features dataset, I normalized the dataset to have 0 mean, and 1 variance. For the dependent variable, I performed log transformation. This was the before and after applying this preprocessing step -



As we can see from the above graph, the skewness is decreased after log transformation. Although we have a big bar at (x=0), and we will see later on how this will cause an issue in our task of predicting the dependent variable.

Visualizing the high dimensional data by converting it into lower dimensions

I converted the 448 dimensional data into 2 dimensional data (as it's not possible (yet) to visualize 448 dimensional data). To get better insights, I converted the target variable into 4 categories. The graph below shows the distribution. We can see some clustering in sample with very high, and also samples with high value of DON concentration.



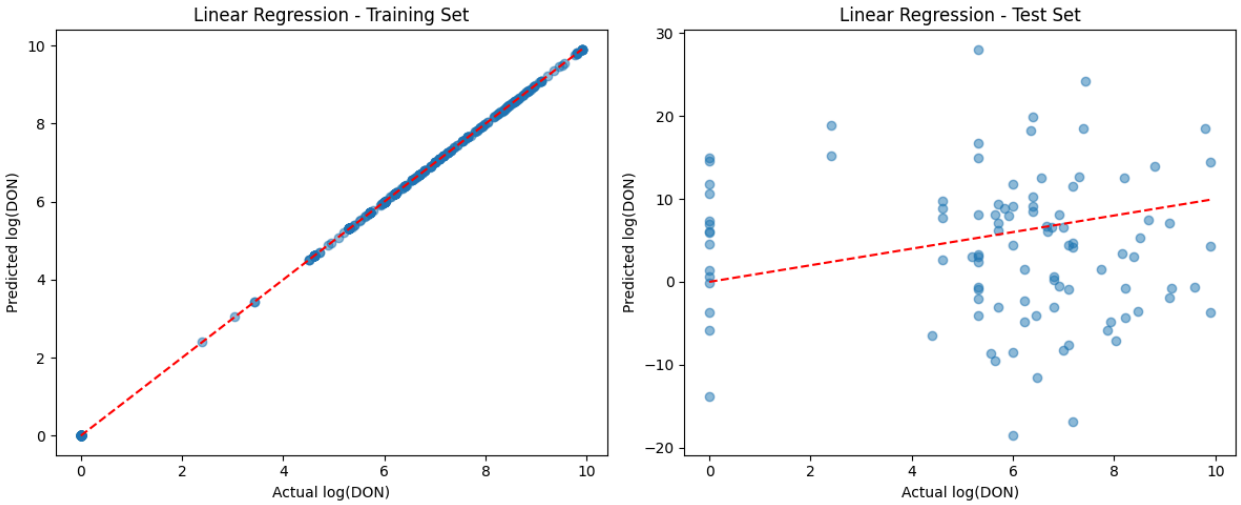
Model Training

I tried 5 models, Linear regression, ridge regression, lasso, neural network, and neural network with a custom function. Here's an overall analysis for the same -

	model	rmse_train	rmse_test	r2_train	r2_test	mae_train	mae_test
0	Linear Regression	6.292157e-14	9.335031	1.000000	-10.399320	4.854450e-14	7.523131
1	Ridge Regression	2.321355e+00	2.524650	0.375741	0.166223	1.800695e+00	1.872137
2	Lasso Regression	2.618642e+00	2.526850	0.205610	0.164770	2.026639e+00	1.876149
3	Neural Network (Optimized)	9.705960e-01	2.857929	0.890866	-0.068441	6.932429e-01	2.040850
4	Weighted Neural Network (Optimized)	1.328457e+00	3.052826	0.795555	-0.219134	9.692876e-01	2.290170

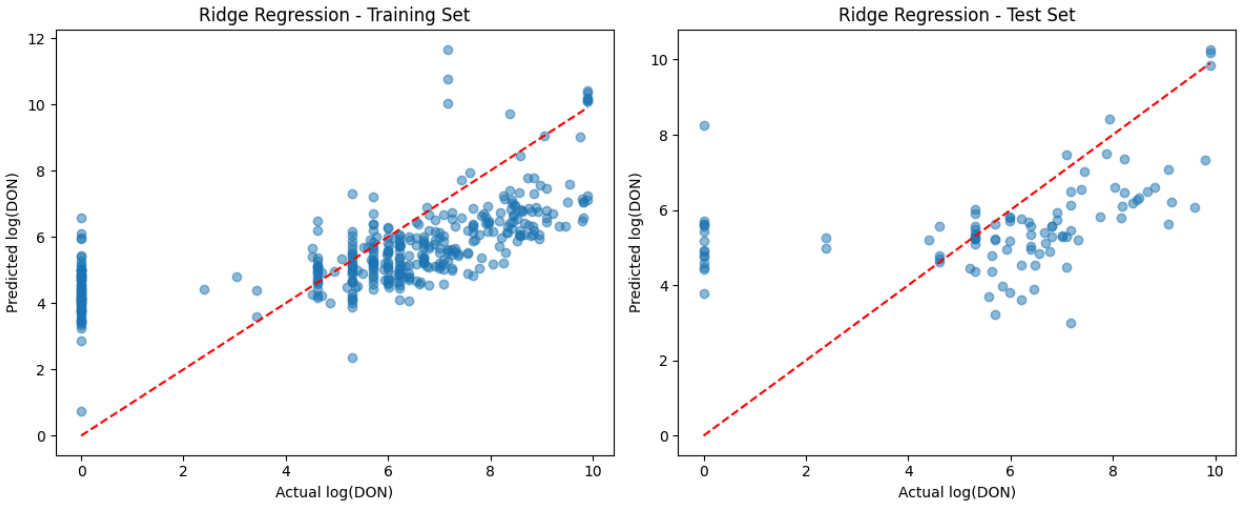
For each of the above mentioned models, we will go in a bit more detail -

Linear Regression:



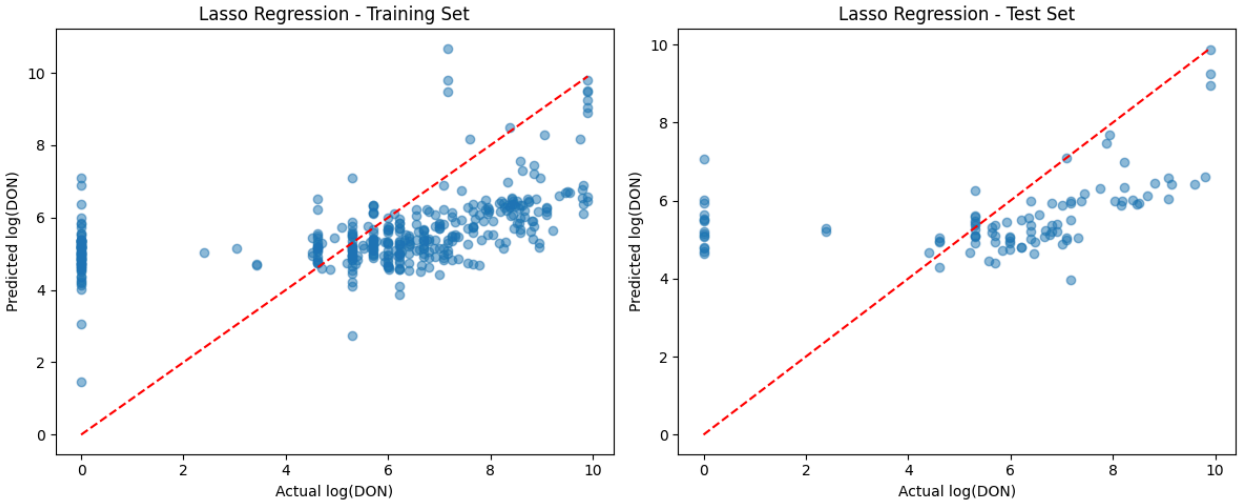
Looking at the left curve, we can state that it's highly overfitting the data. Also, we can assume that adding regularization will improve the results.

Ridge Regression:

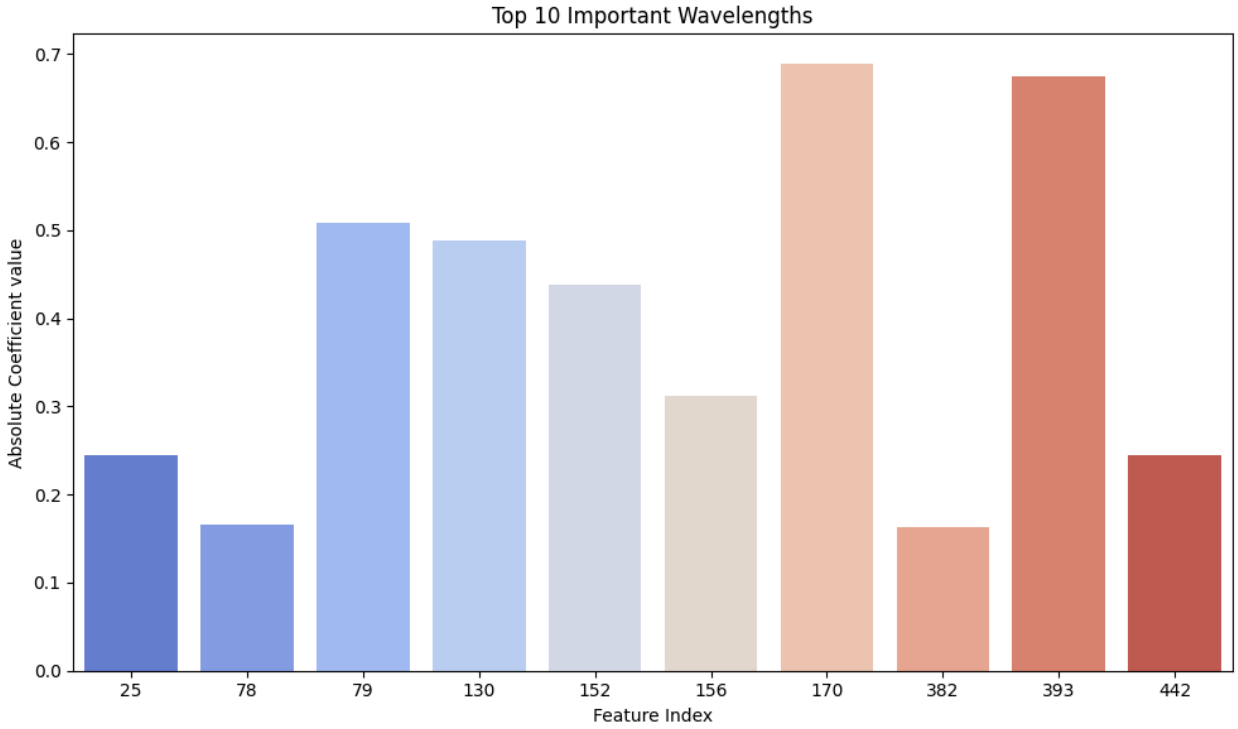


As expected, the performance of linear regression improves with regularization.

Lasso Regression:



Lasso regression also gives a better performance. One more thing possible with lasso regression is feature selection. Here are the features that was selected by lasso -



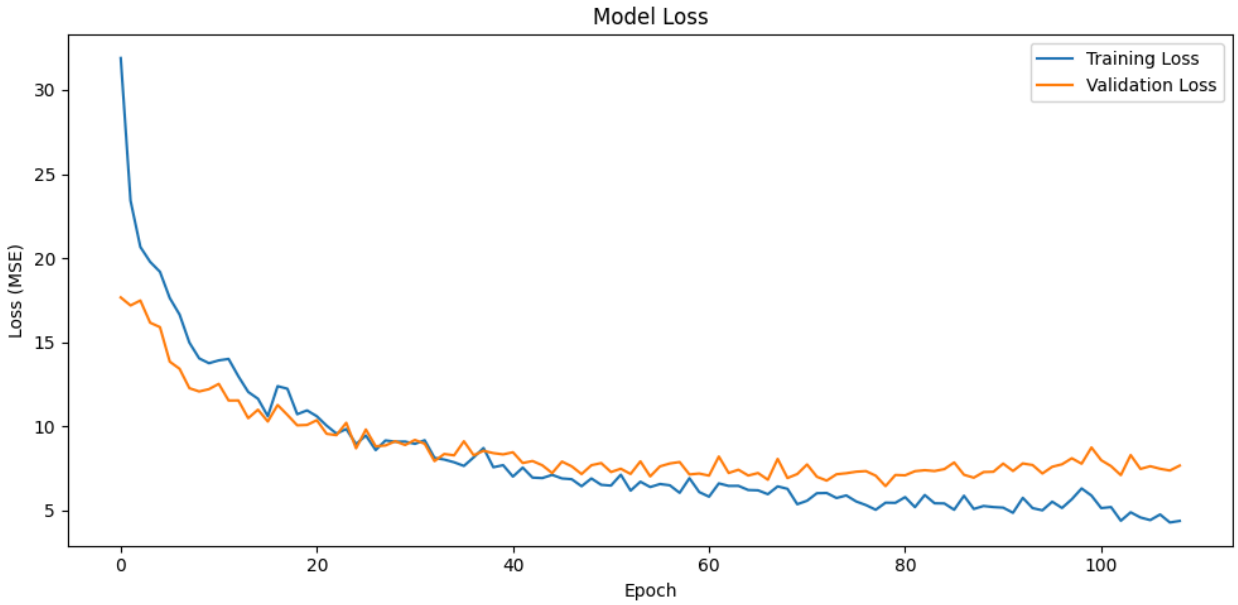
Notice that the feature with max coefficient value also had the most correlation when we looked at how features are correlated in data analysis part.

Neural Network:

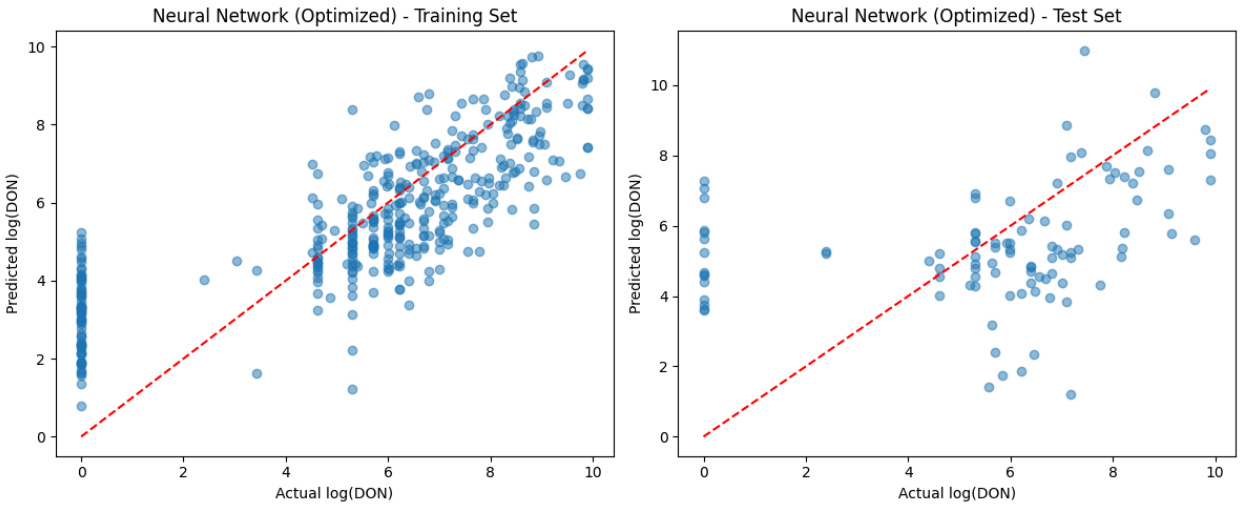
I performed hyperparameter tuning with optuna, and selected the optimum values for the number of neurons in each layer, dropout rate, and learning rate. These are the values I got -

```
Best hyperparameters for Regular Neural Network:
hidden_layer_0: 160
hidden_layer_1: 32
hidden_layer_2: 96
dropout_rate: 0.37613860011805833
learning_rate: 0.006210907736107205 5
```

I trained the model using these values, and got the following training curve -

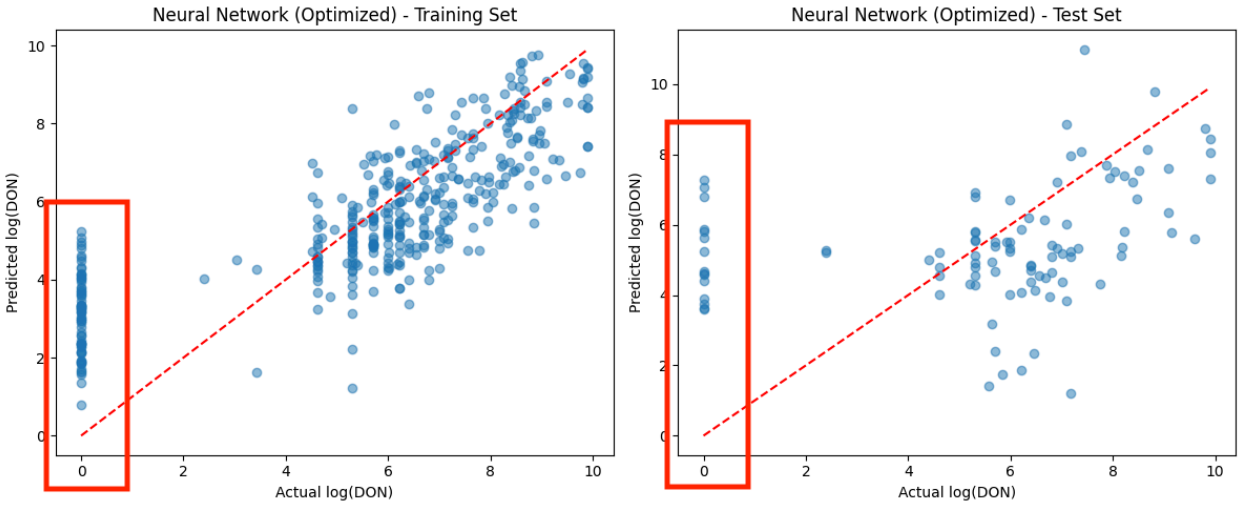


This was the performance of the neural network:



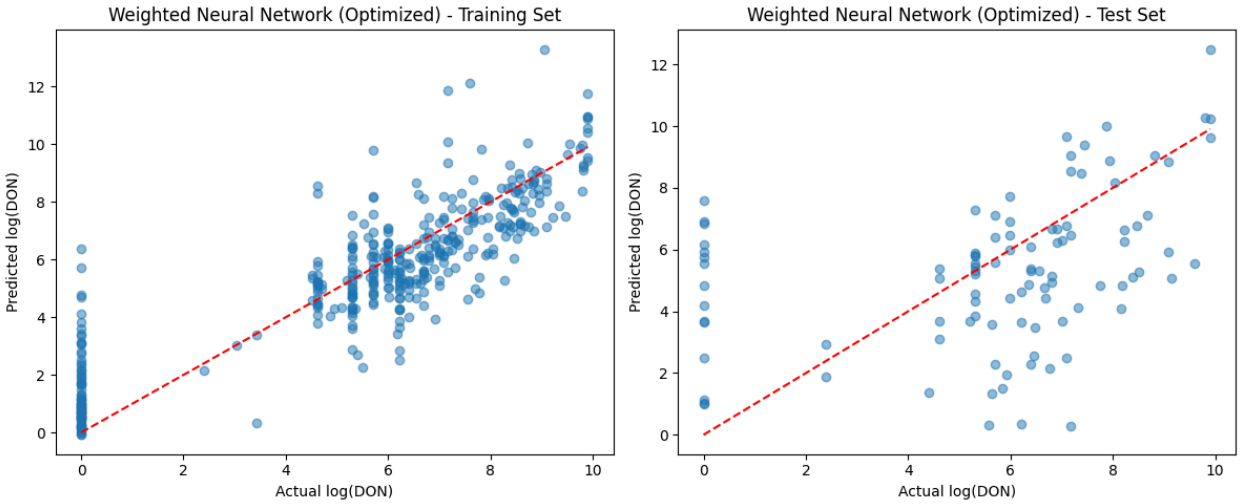
The problem with this model, and how to amend it:

The data has many samples where the value of DON concentration is 0. These are the samples which are completely safe for consumption. Although, the problem with our models above is that they are mispredicting the samples with 0 DON concentration. The interested region is highlighted in the image below -



Neural Network with a custom loss function:

To give more emphasis on the samples with low DON concentration, I trained a neural network with a custom loss function, ie, weighted MSE loss. It did improve the performance of the model for samples with 0 value for DON concentration, which is evident from the graph below -



We can see that the model performs better on samples where DON concentration is 0.

TO-DO:

These are the things I had planned to do but couldn't implement due to time constraints -

- Implement a two phases model, a combination of Classification and regression such that the model performs better on samples with low value of DON concentration
- Model interpretability with SHAP.
- A demo using streamlit.

The code structure:

The codebase follows a modular implementation, with well commented and well documented code, mostly using OOPS concept. This is the overall structure -

```
├── README.md
├── src/
│   ├── config/
│   │   └── config.py         # Configuration parameters
│   ├── data/
│   │   └── data_loader.py    # Data loading and preprocessing
│   ├── models/
│   │   └── models.py         # Model architectures
│   ├── utils/
│   │   ├── training.py       # Training utilities
│   │   └── visualization.py  # Visualization utilities
│   └── main.py              # Main script
├── x_processed.csv          # Input features
└── y_processed.csv          # Target values
```