## horizontal line



CSTP2301 Final Project V2

16.03.2024

**─**

ML

Jia Xi Lin

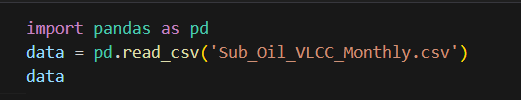
Muochu Hu

# Goals

1. Picking IDs 542236, 67321, 549295, 41108, 54982 to do the calculations.
2. Using a regression model to train and test the data set.
3. Comparing the actual value of y and predicted value of y and calculate the accuracy of them, fill the results in an Excel file.

#### **Step 1: Read the Dataset**

Begin by importing necessary libraries and loading the dataset. Use pandas to read the CSV file containing the data. This step is crucial as it sets up your data for preprocessing and analysis.

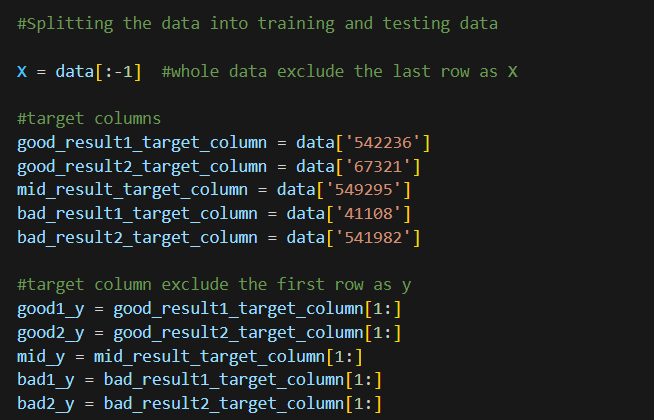


#### **Step 2: Data Preprocessing**

Prepare the dataset for the model. This involves splitting the dataset into features (X) and target variables (y). Since the model requires numerical inputs, ensure that the data is cleaned and appropriately formatted.

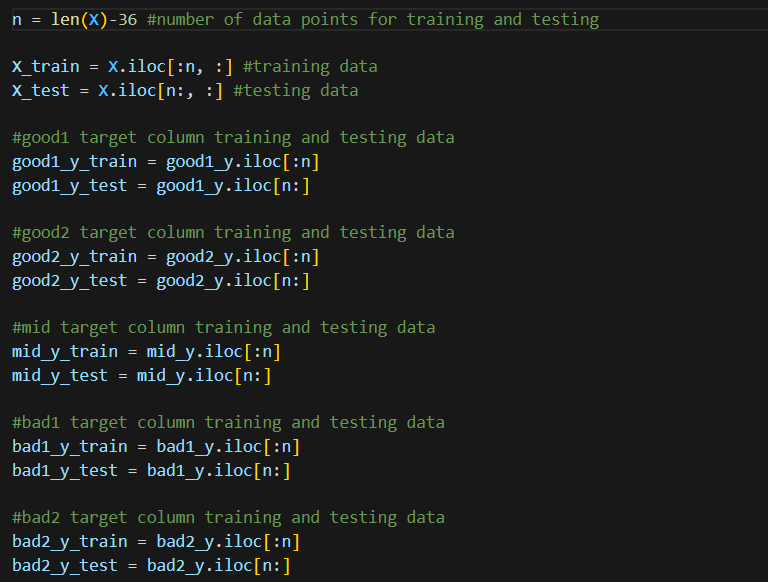
* Exclude the last row from the dataset for features (X).
* Define target columns for various scenarios (good, mid, and bad results).
* Exclude the first row from target columns to align with X.

To generate X and y, we can consider the entire table as X and copy the target column in a vector as y. Remember that the label of sample t in X is in row t+1 in y.

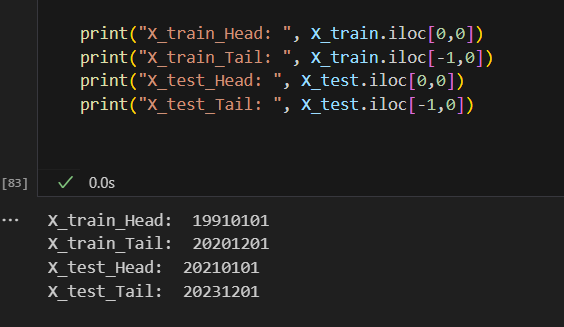


**Step 3: Splitting the Data into Training and Testing Sets**

It's essential to evaluate the model's performance on unseen data. Split the dataset into a training set used for learning and a testing set for evaluation. We aim to assess the model's accuracy over the last three years available in the dataset. To achieve this, consider training the model on all samples from the beginning up to n-36, and then test it on the last 36 samples (n is the total number of samples excluding the last one that doesn’t include a label)



Making sure of the training data are the data from 19910101 to 20201201, and the testing data are from 20210101 to 20231201 by printing them out.

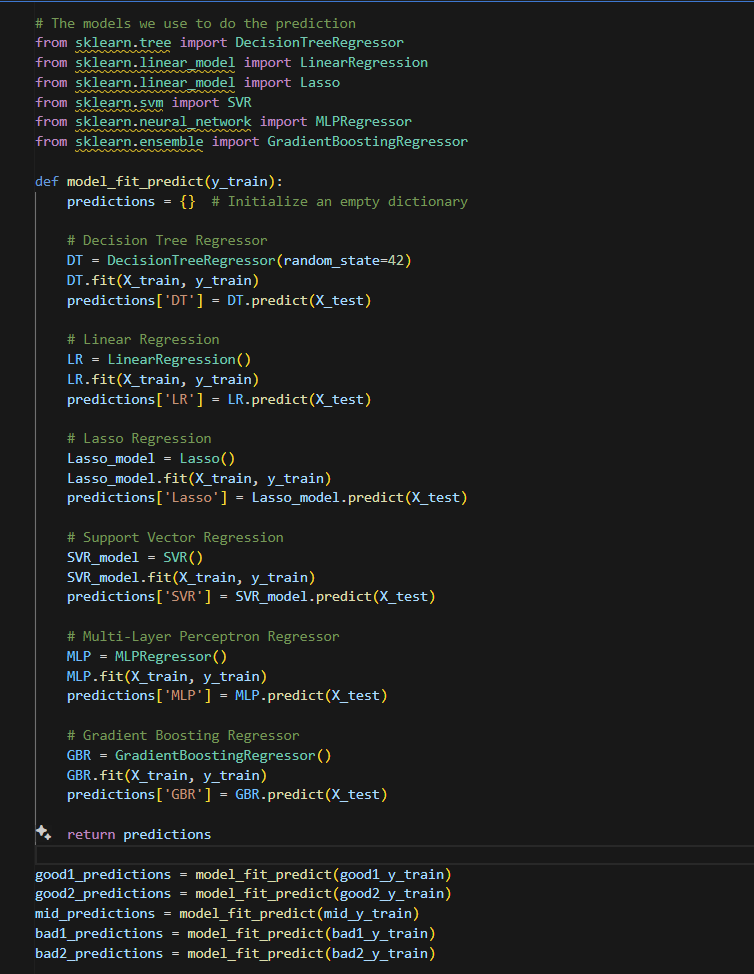


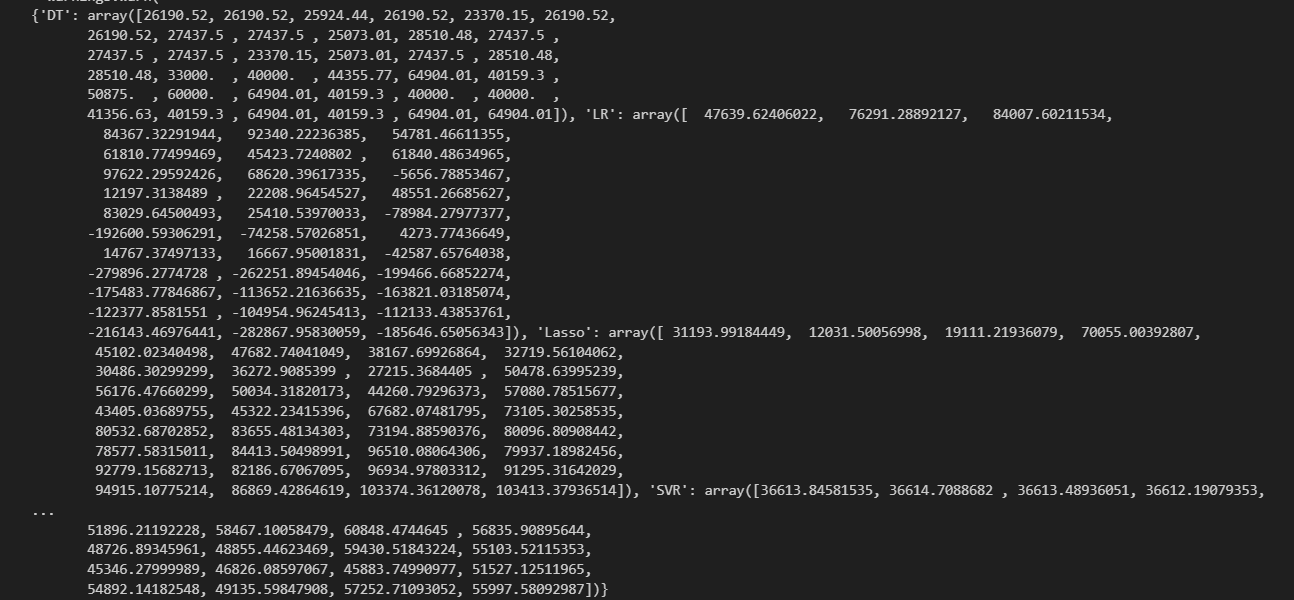
#### **Step 4: Model Selection and Training**

We designed a model\_fit\_predict function to simplify the process of training and predicting with various regression models. This function receives the training data and a target variable, then applies several algorithms to produce a dictionary of predictions.

We tested five models: Decision Tree Regressor, Linear Regression, Lasso Regression, Support Vector Regression, and Multi-Layer Perceptron Regressor, plus a Gradient Boosting Regressor.After fitting each model to the training data, we invoked the predict method using the test set (X\_test) to generate predictions.

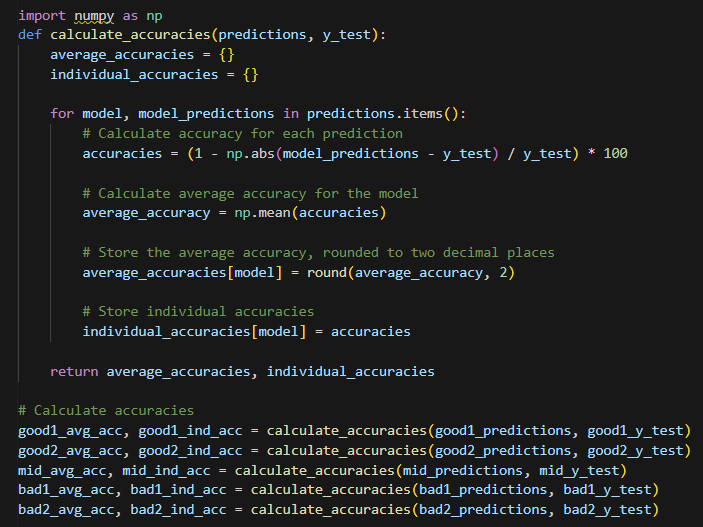
By organizing predictions in a dictionary with model names as keys, we facilitated the subsequent evaluation of model performance based on accuracy metrics. This structured approach was instrumental in identifying the most suitable model for each target column, leading us to a set of predictions that could then be analyzed for accuracy and reliability.



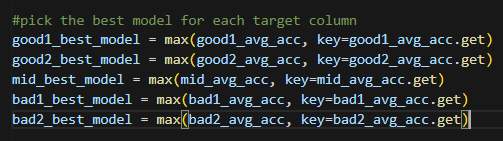


#### **Step 5: Model Prediction and Evaluation**

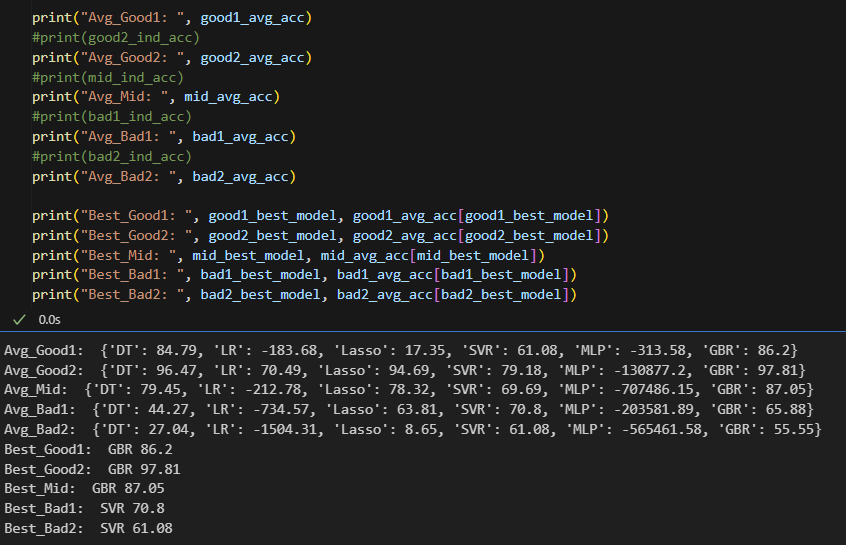
The calculate\_accuracies function measures how well our models predict outcomes. It measures the difference between what the model predicts and the actual results, turning this into a percentage that represents the model's accuracy. This is done for each prediction and then averaged out to get the model's overall accuracy. We've applied this function to several datasets, allowing us to see which model gives the most accurate predictions for each case. The results are kept neat, with individual prediction accuracies and the model's average accuracy neatly recorded and easy to compare.



After the accuracies are calculated, the next step is to find the best accuracy for each model.



We can easily find out the best model for each target ID by printing the results:

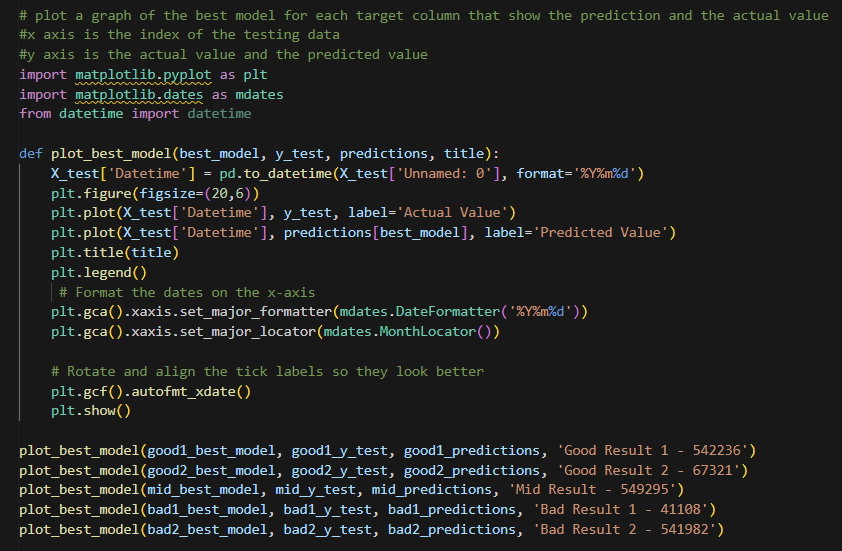


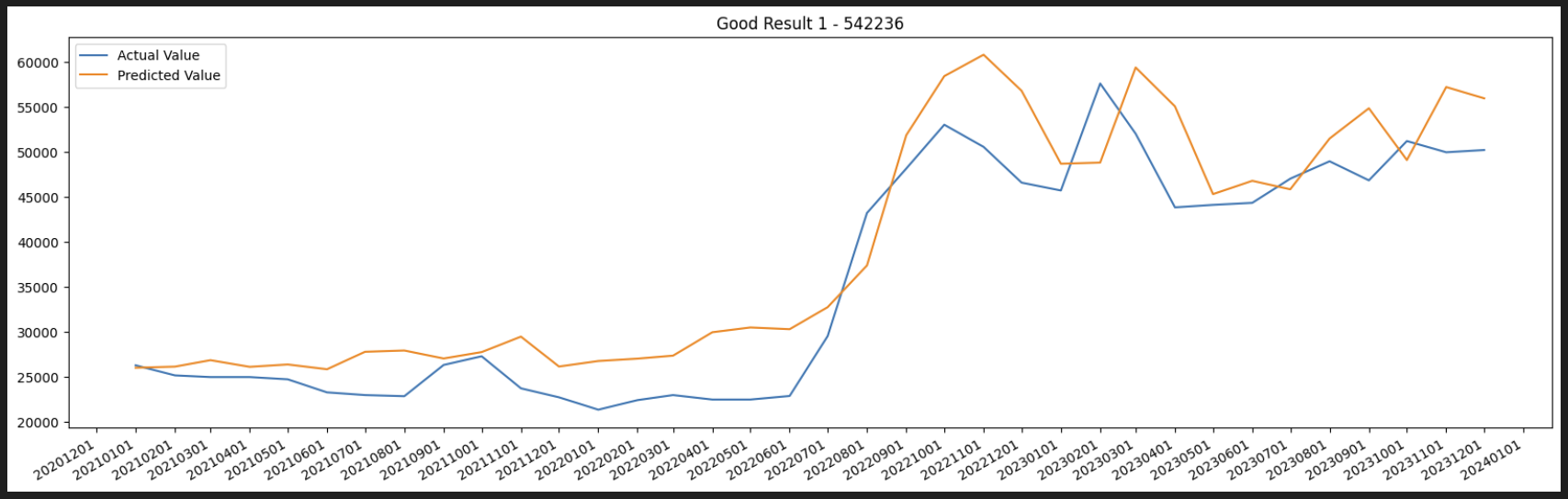
#### 

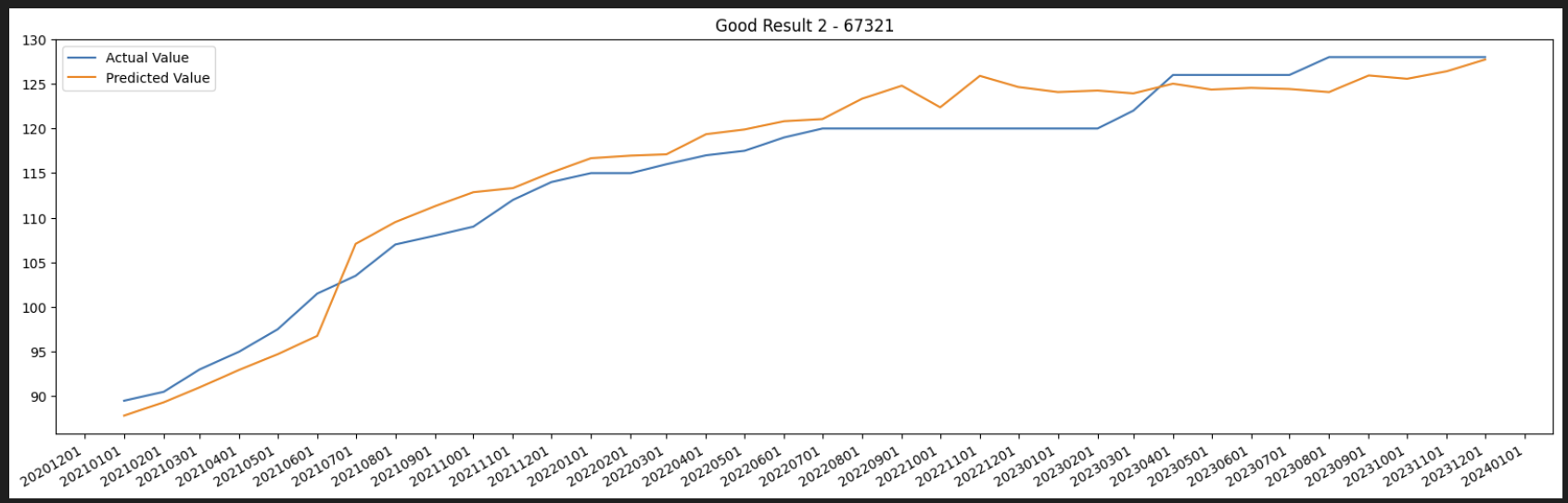
#### 

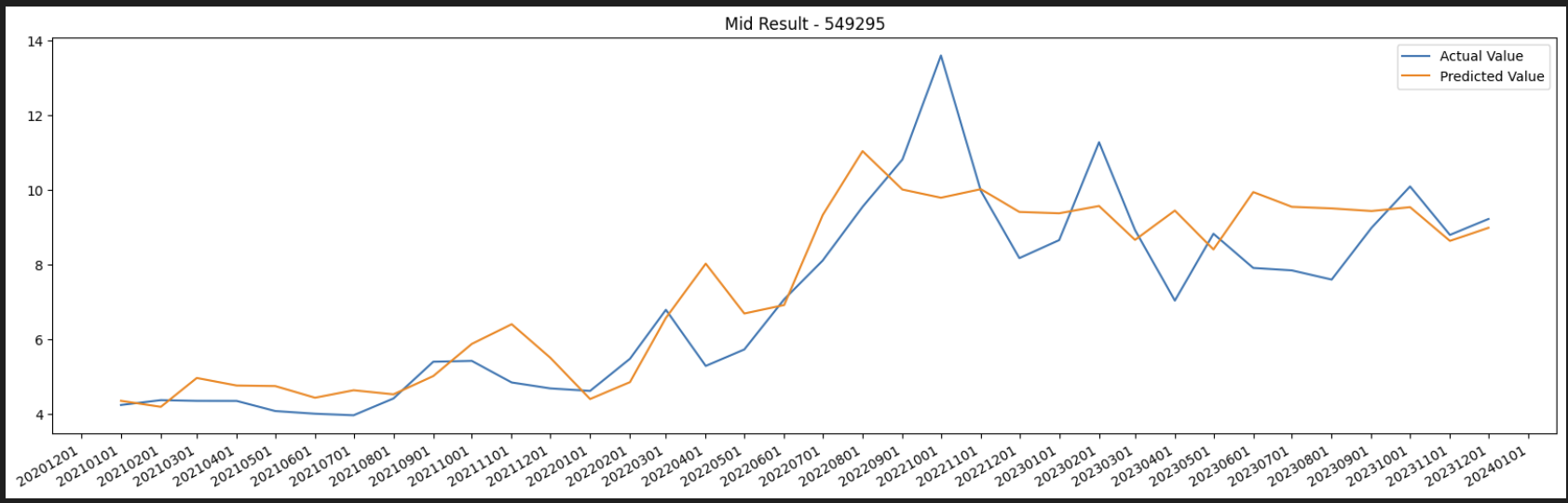
#### **Step 6: Plot Diagrams**

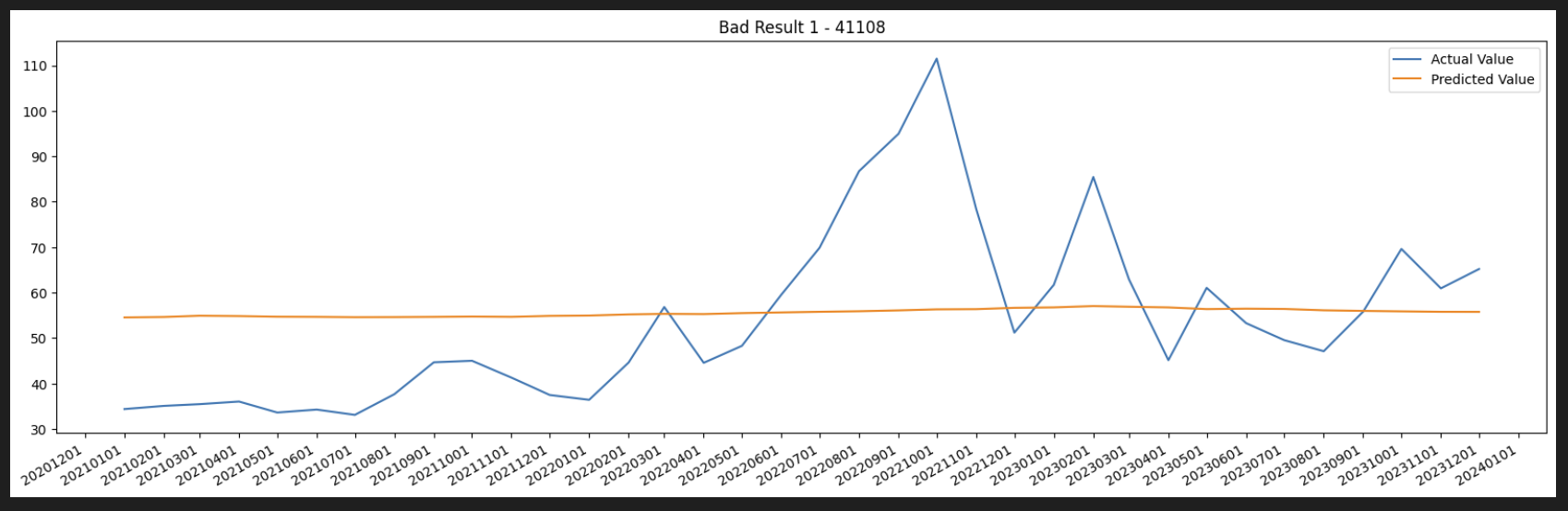
Using the Matplotlib library for plotting the performance of prediction models over time. The script defines a function called plot\_best\_model.These plots are critical for visually assessing how well each model's predictions align with the actual data, which is an essential part of the model evaluation process.

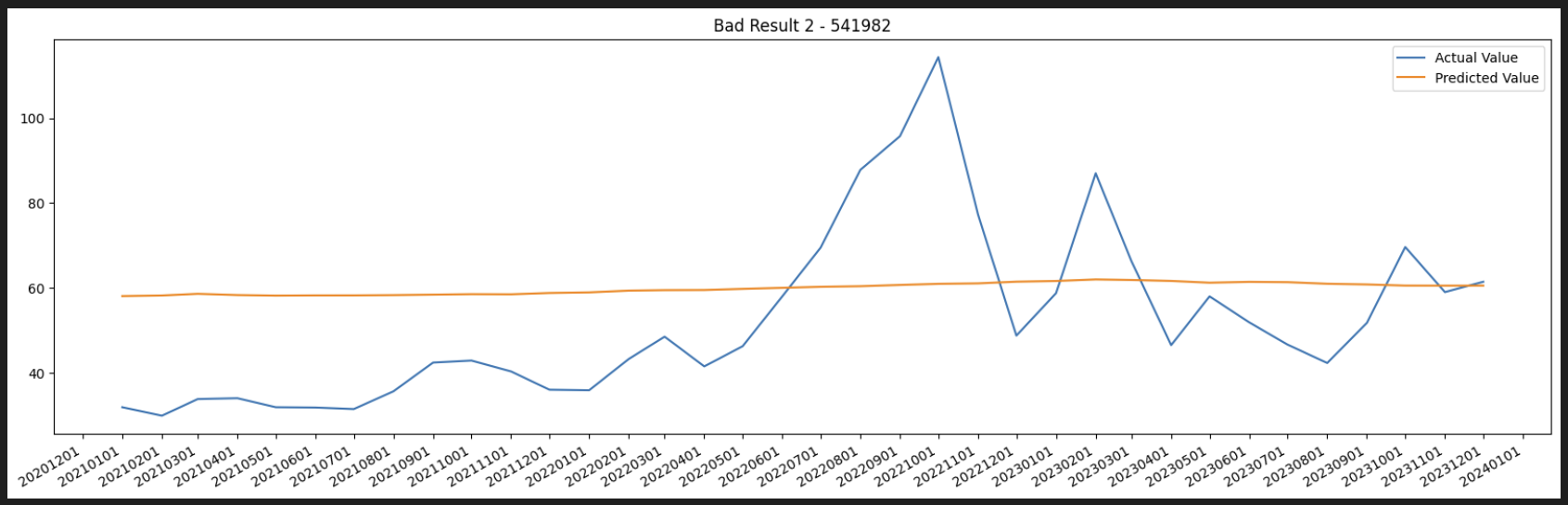








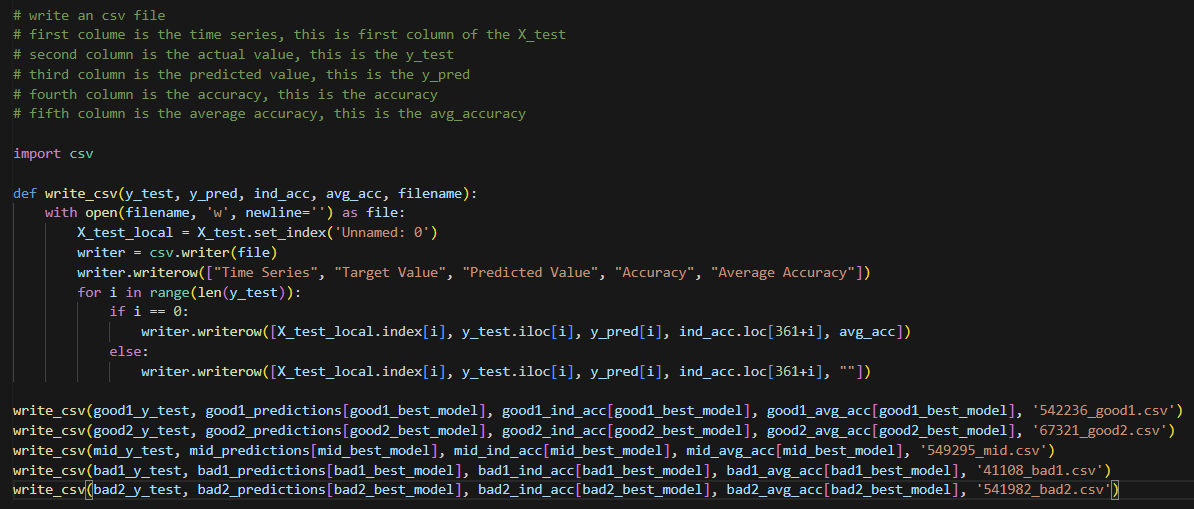




#### **Step 7: Export csv files**

we devised the write\_csv function. This function takes in the actual values, the predicted values, the individual prediction accuracies, the average accuracy, and the desired filename for output.

For every model, we used the function to write a CSV file containing the time series from our test set, along with the corresponding actual and predicted values, and their accuracies. It neatly places the average accuracy at the top of the column, avoiding repetition, for a cleaner look in the data file.



### **Conclusion**

In this project, we delved into a comprehensive study using an array of regression models on the Sub\_Oil\_VLCC\_Monthly.csv dataset to forecast various outcomes that were pre-classified as good, mid, and bad. Our toolkit included a selection of models: the Decision Tree Regressor, Linear Regression, Lasso Regression, Support Vector Regression (SVR), Multi-Layer Perceptron Regressor, and Gradient Boosting Regressor (GBR).

Our structured approach kicked off with meticulous data preprocessing. This set the stage for splitting the dataset into training and testing sets, a critical step for a valid assessment of each model's predictive capabilities. Each model was diligently trained on distinct data segments targeting specific outcome variables.

We gauged the performance through accuracy metrics, which compared the predicted values to the actual outcomes. Comprehensive reports showcased the performance, highlighting time series data, actual versus predicted values, and accuracy figures.

A deep dive into the performance metrics revealed that different models excelled for various target IDs. The GBR model was superior for '542236' with an accuracy rate of 86.2%, and for '67321', it impressively reached 97.81% accuracy. The target ID '549295' also benefited from the GBR, achieving 87.05% accuracy. However, the SVR model showed its strengths for '41108', with a 70.8% accuracy rate, and for '541982', managing a 61.08% accuracy rate.

The selection of the best-suited model for each target ID is a testament to the project's success, illustrating the distinct efficacy of each regression model across a spectrum of scenarios. This not only highlights the GBR's comparative advantage in specific applications but also the strategic potential of SVR in others, setting a benchmark for model selection in predictive analytics and opening avenues for targeted improvements in modeling techniques.