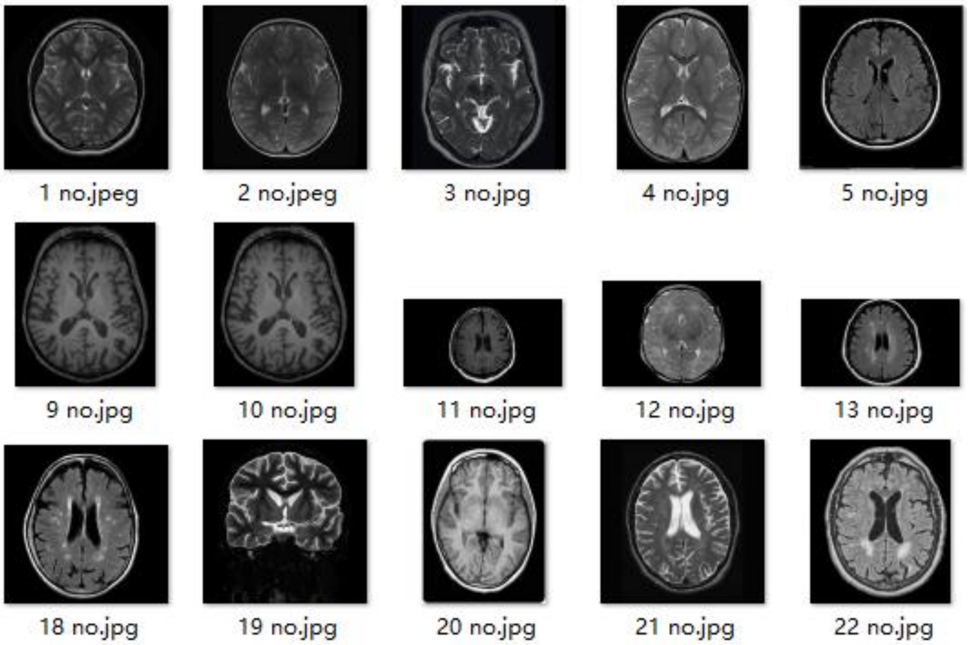
0. Dataset Choose

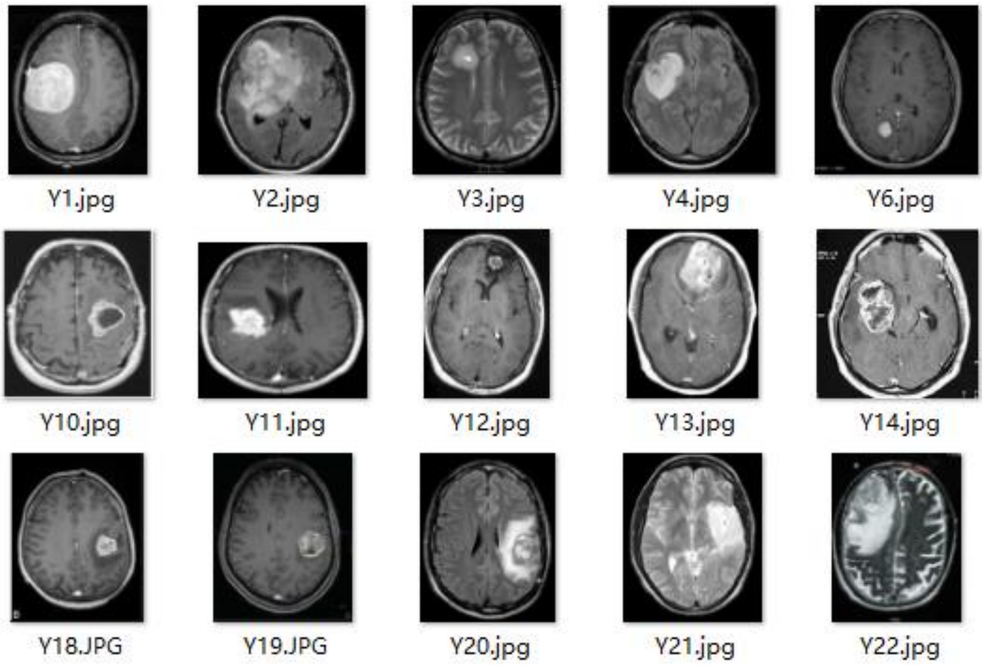
0.1 Classification - Brain Tumor Dataset

This dataset is designed to aid in the development of computer vision models for the early detection and classification of brain tumors. It contains medical images, typically MRI or CT scans, of human brains with and without tumors.

Healthy:

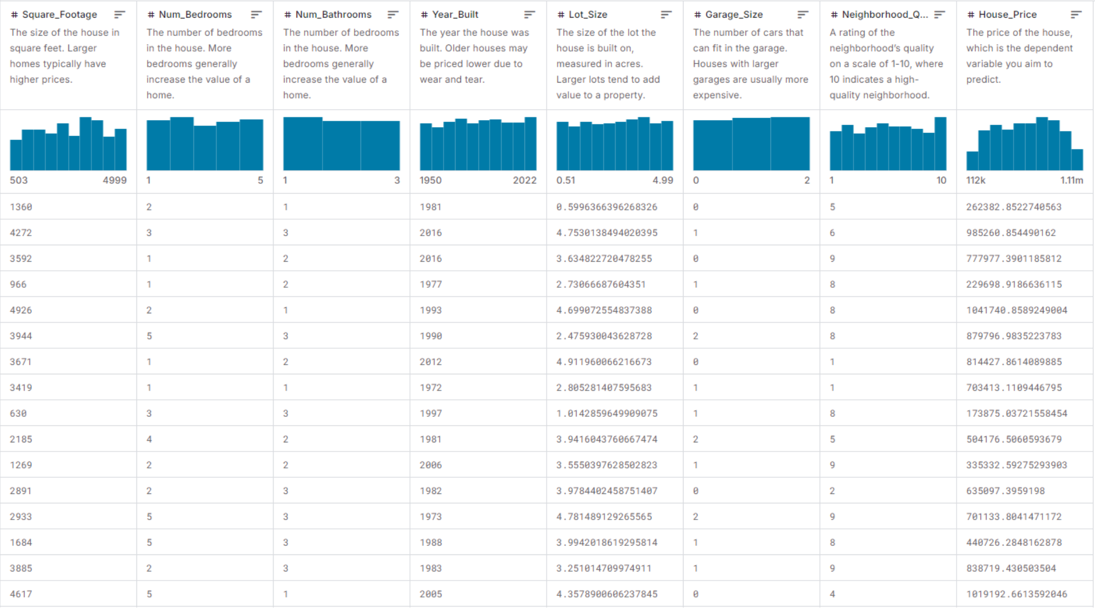


Tumor:



0.2Regression - House Price Regression Dataset

This dataset is designed for beginners to practice regression problems, particularly in the context of predicting house prices. It contains 1000 rows, with each row representing a house and various attributes that influence its price. The dataset is well-suited for learning basic to intermediate-level regression modeling techniques.



# Square_Footage	# Num_Bedrooms	# Num_Bathrooms	# Year_Built	# Lot_Size	# Garage_Size	# Neighborhood_Q...	# House_Price
The size of the house in square feet. Larger homes typically have higher prices.	The number of bedrooms in the house. More bedrooms generally increase the value of a home.	The number of bedrooms in the house. More bedrooms generally increase the value of a home.	The year the house was built. Older houses may be priced lower due to wear and tear.	The size of the lot the house is built on, measured in acres. Larger lots tend to add value to a property.	The number of cars that can fit in the garage. Houses with larger garages are usually more expensive.	A rating of the neighborhood's quality on a scale of 1-10, where 10 indicates a high-quality neighborhood.	The price of the house, which is the dependent variable you aim to predict.
503 4999	1 5	1 3	1950 2022	0.51 4.99	0 2	1 10	112k 1.11m
1360	2	1	1981	0.5996366396268326	0	5	262382.8522740563
4272	3	3	2016	4.7530138494020395	1	6	985260.854490162
3592	1	2	2016	3.634822720478255	0	9	777977.3901185812
966	1	2	1977	2.73066687604351	1	8	229698.9186636115
4926	2	1	1993	4.699072554837388	0	8	1041740.8589249004
3944	5	3	1990	2.475930043628728	2	8	879796.9835223783
3671	1	2	2012	4.911960066216673	0	1	814427.8614089885
3419	1	1	1972	2.805281407595683	1	1	703413.1109446795
630	3	3	1997	1.0142859649909075	1	8	173875.03721558454
2185	4	2	1981	3.9416043760667474	2	5	504176.5060593679
1269	2	2	2006	3.5550397628502823	1	9	335332.59275293903
2891	2	3	1982	3.9784402458751407	0	2	635097.3959198
2933	5	3	1973	4.781489129265565	2	9	701133.8041471172
1684	5	3	1988	3.9942018619295814	1	8	440726.2848162878
3885	2	3	1983	3.251014709974911	1	9	838719.430503504
4617	5	1	2005	4.3578900606237845	0	4	1019192.6613592046

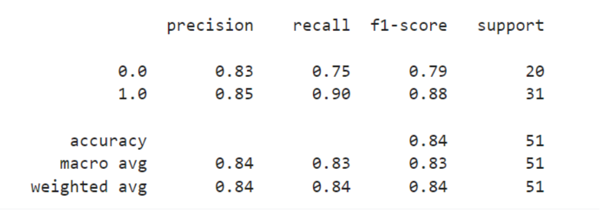
We take use *Square_Footage, num_Bedroomsm, Year_Built, Lot_Size*, etc. as the input X and the *House_Price* as the predict value Y.

1. Classifiers

1.1 Logistic Regression

Metrix:

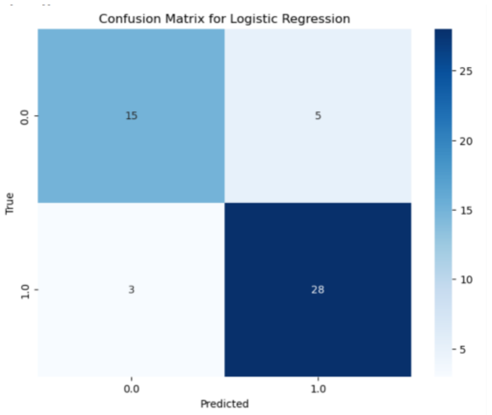
Confusion Matrix:



```
              precision    recall  f1-score   support

         0.0       0.83      0.75      0.79        20
         1.0       0.85      0.90      0.88        31

    accuracy                           0.84        51
   macro avg       0.84      0.83      0.83        51
weighted avg       0.84      0.84      0.84        51
```



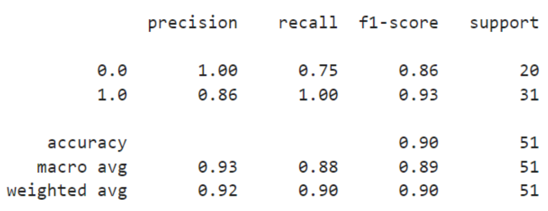
The confusion matrix shows that the classifier performed well with 3 false negatives and 5 false positives, indicating some misclassification but overall good precision and recall.

Model Justification: Logistic Regression is generally a solid baseline model for binary classification tasks. It is the right model here due to its simplicity and interpretability. However, it might not capture non-linear relationships in the data.

1.2. Support Vector Machine (SVM)

Metrix:

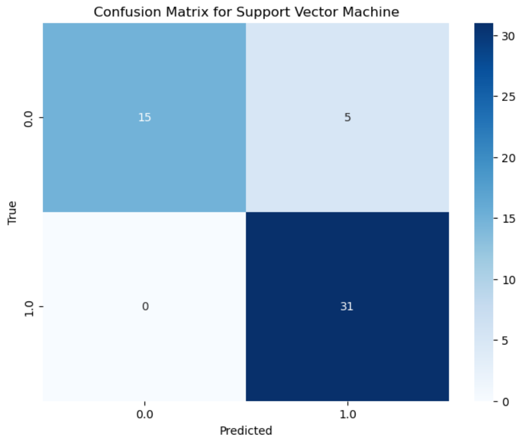
confusion Matrix:



```
              precision    recall  f1-score   support

         0.0       1.00      0.75      0.86        20
         1.0       0.86      1.00      0.93        31

    accuracy                           0.90        51
   macro avg       0.93      0.88      0.89        51
weighted avg       0.92      0.90      0.90        51
```

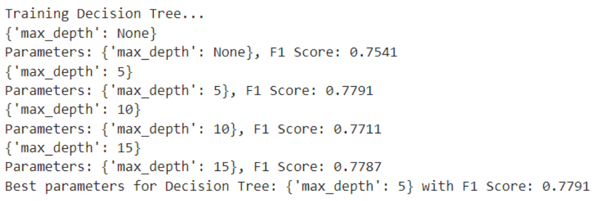


The confusion matrix indicates that the SVM model misclassified 5 instances of class 0 as class 1 but made no false negatives in class 1.

Model Justification: SVM with a non-linear kernel tends to perform well with smaller datasets and non-linear boundaries. It seems to have been a good fit for this task, providing high precision and recall.

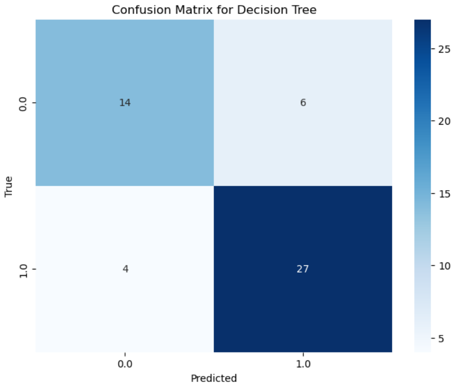
1.3. Decision Tree

Metrix:



```
Training Decision Tree...
{'max_depth': None}
Parameters: {'max_depth': None}, F1 Score: 0.7541
{'max_depth': 5}
Parameters: {'max_depth': 5}, F1 Score: 0.7791
{'max_depth': 10}
Parameters: {'max_depth': 10}, F1 Score: 0.7711
{'max_depth': 15}
Parameters: {'max_depth': 15}, F1 Score: 0.7787
Best parameters for Decision Tree: {'max_depth': 5} with F1 Score: 0.7791
```

Confusion Matrix:

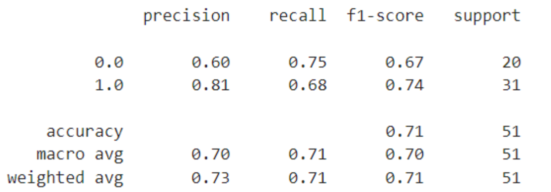


The confusion matrix indicates moderate performance, with some misclassification for both classes. The tree seems to underfit slightly.

Model Justification: While Decision Trees can handle non-linear patterns, they often suffer from overfitting or underfitting based on their depth. For this dataset, the model could have been better with some tuning (e.g., limiting the depth).

1.4. Multi-Layer Perceptron (MLP)

Metrix:

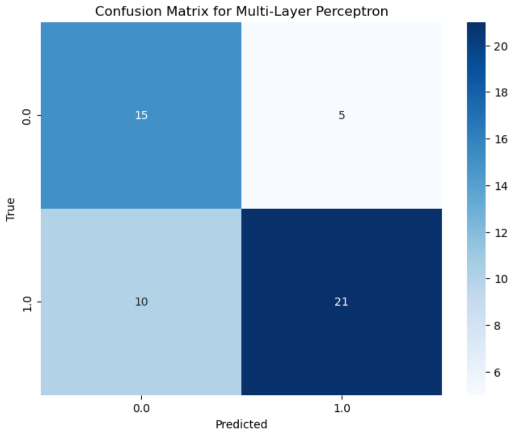


```
              precision    recall  f1-score   support

         0.0       0.60      0.75      0.67        20
         1.0       0.81      0.68      0.74        31

    accuracy                           0.71        51
   macro avg       0.70      0.71      0.70        51
weighted avg       0.73      0.71      0.71        51
```

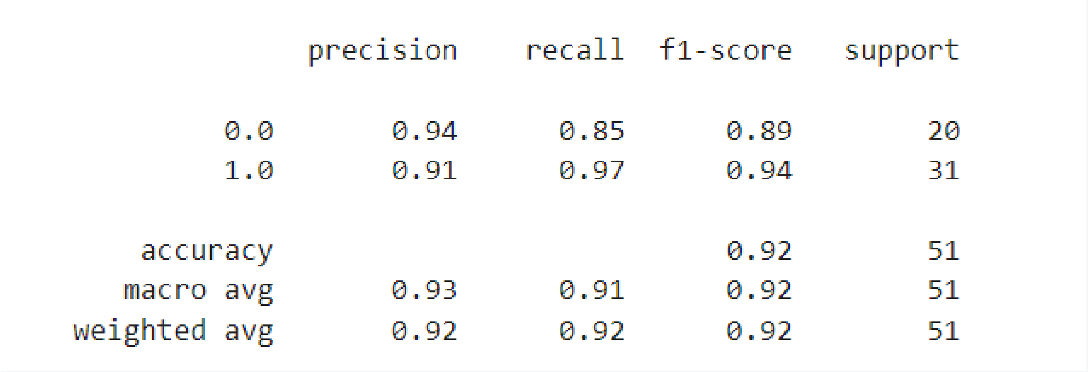
Confusion Matrix:



The MLP model shows a weaker performance compared to others, particularly with a high number of false negatives for class 1.

Model Justification: The MLP is a more complex model, which may explain why it underperformed here. The dataset may not be large or complex enough to benefit from a neural network-based approach.

1.5. Random Forest

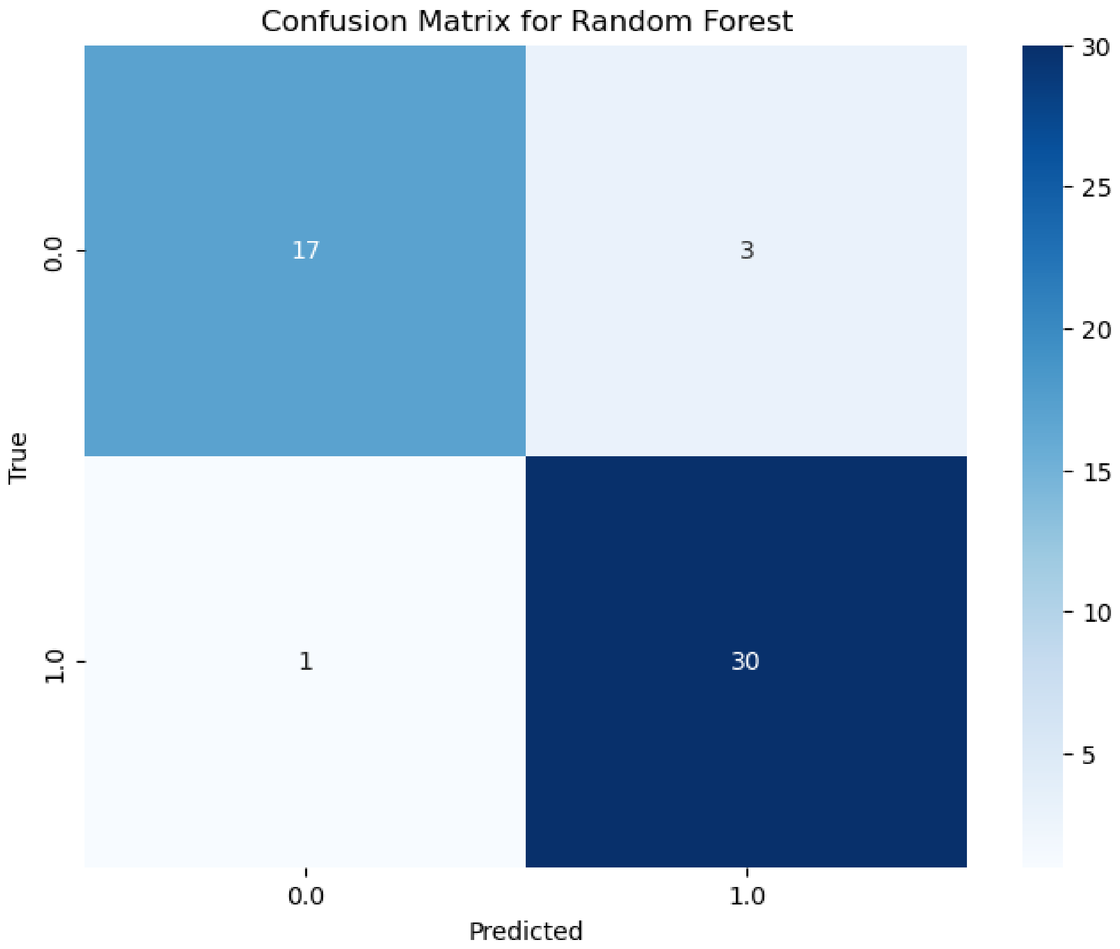


```
              precision    recall  f1-score   support

         0.0       0.94      0.85      0.89        20
         1.0       0.91      0.97      0.94        31

    accuracy                           0.92        51
   macro avg       0.93      0.91      0.92        51
weighted avg       0.92      0.92      0.92        51
```

Confusion Matrix:



The confusion matrix indicates that Random Forest performed very well, with only a few misclassifications.

Model Justification: Random Forest's ability to reduce variance and overfitting through ensemble methods makes it a highly suitable model for this task. Its strong performance justifies its choice.

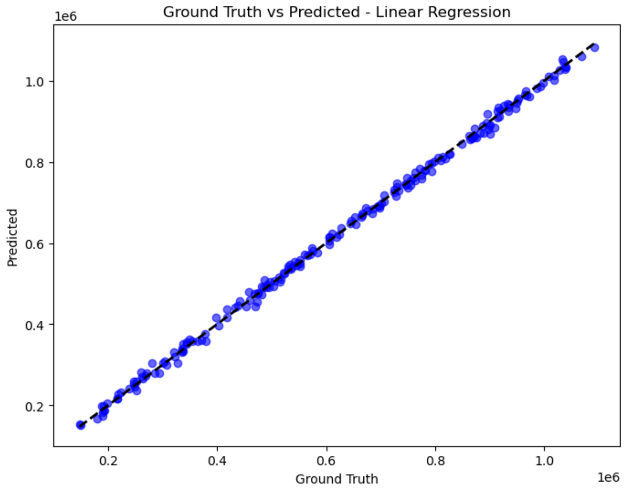
2. Regression Models

2.1 Linear Regression

Mean Squared Error (MSE): 101,434,798.51

Scatter Plot Analysis: The scatter plot shows a linear relationship between predicted and actual values but with some spread, especially at extreme values.

Model Justification: Linear Regression works well if the relationships between features and the target are linear. However, for this dataset, there may be non-linear relationships causing suboptimal performance.

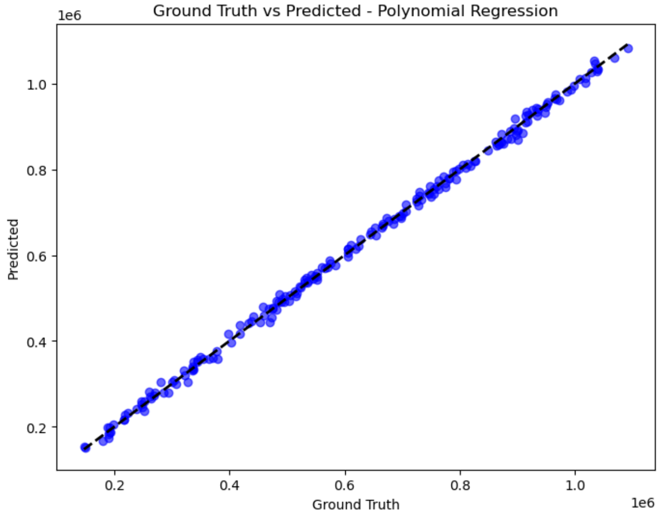


2.2. Polynomial Regression

Mean Squared Error (MSE): 101,434,798.51 (degree = 1)

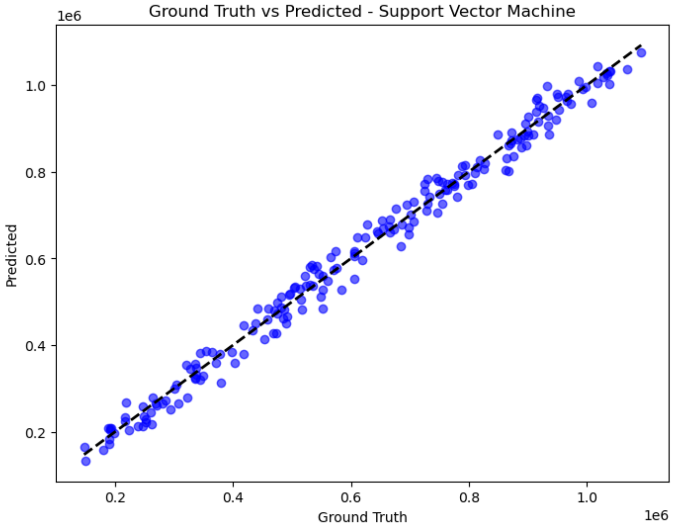
Scatter Plot Analysis: The scatter plot shows a similar performance to Linear Regression, indicating that a polynomial degree of 1 (essentially linear) was the best fit.

Model Justification: The best-performing degree was 1, meaning the data did not benefit from higher polynomial transformations. This suggests that linear relationships dominated the data.



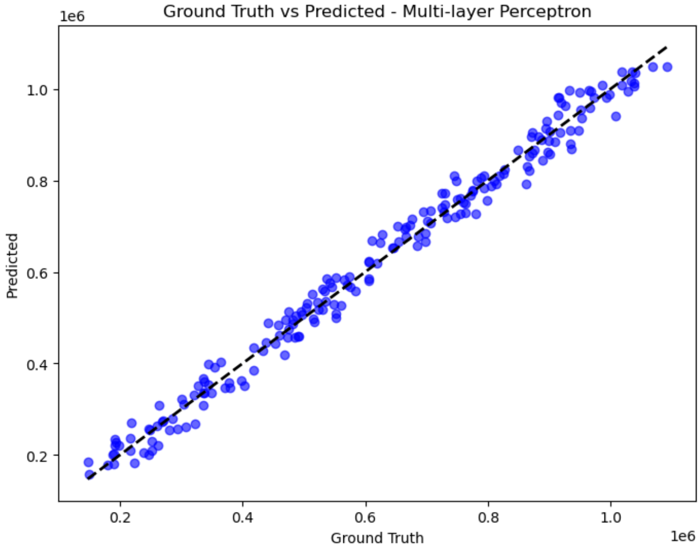
2.3. Support Vector Machine (SVM) for Regression

Mean Squared Error (MSE): 753,919,697.96
Scatter Plot Analysis: The scatter plot shows a wide spread, indicating poor prediction accuracy.
Model Justification: SVM for regression can struggle if the relationship between variables is highly non-linear or if the kernel is not well-tuned. For this task, it does not seem to be the right choice.



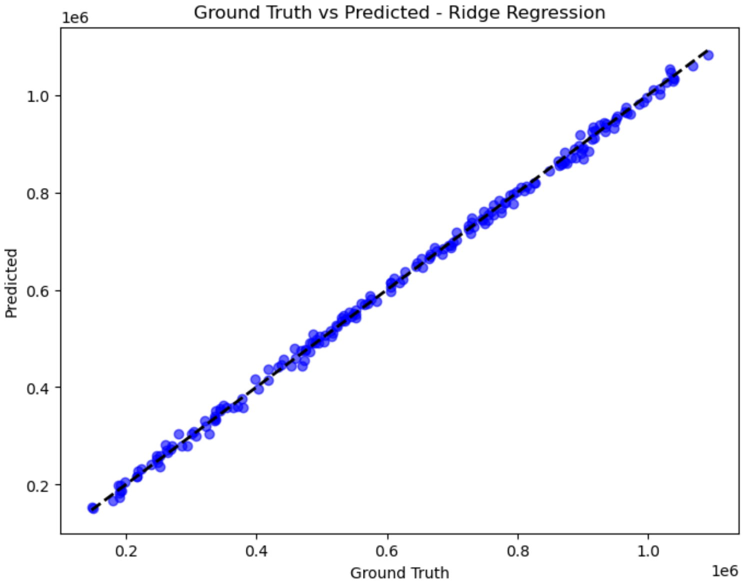
2.4 Multi-layer Perceptron (MLP) for Regression

Mean Squared Error (MSE): 879,859,957.36
Scatter Plot Analysis: The scatter plot shows a significant spread in predictions, especially in extreme values.
Model Justification: Neural networks often require large datasets to perform well. For this task, MLP may not have been the ideal model due to the size and complexity of the data.



2.5. Ridge Regression

Mean Squared Error (MSE): 101,438,129.73
Scatter Plot Analysis: Similar to Linear Regression, the scatter plot shows some spread, but Ridge Regression regularizes the coefficients slightly better.
Model Justification: Ridge Regression is helpful when overfitting is a concern. It performed similarly to Linear Regression, suggesting minimal overfitting in the original model.



3.1. Time Taken:

More than 10 hours.

3.2. Collaborators:

I worked independently, discussing general approaches with classmates but did not share any code.

3.3 Resources Used:

Scikit-Learn Documentation: Used for model implementation guidance.
URL: https://scikit-learn.org
Stack Overflow: For debugging issues with plotting and tuning models.
URL: https://stackoverflow.com

3.4. Challenges:

Tuning hyperparameters, especially for SVM and MLP, was challenging due to the large search space. And to write a generalized test code that works for multiple models is very much a test of basic python skills, and I spent a lot of time on it.

5. Most Rewarding Part:

Seeing the performance improvements after tuning Random Forest and Logistic Regression models was the most rewarding. And there is great delight in seeing that the predictions of your model are very close to the true value.

6. Lessons Learned:

I learned how different models perform under various conditions, how to choose better hyperparameters and how important it is to select the right model based on the dataset's characteristics.