CMPN451
Big Data

Assignment Report #4

Group members:

Mostafa Mohamed Sabry	1162211
Youssef Alaa Mostafa	1180092

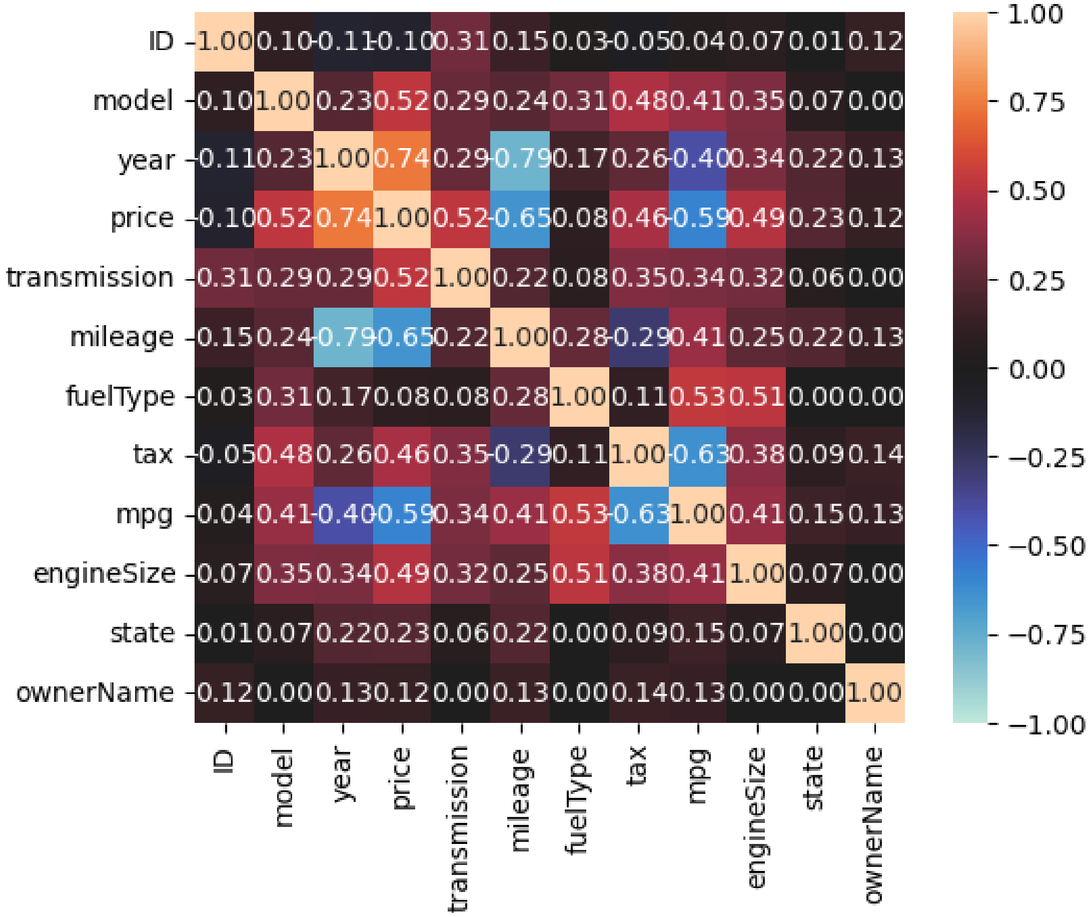
Project Pipeline

1. **Removing irrelevant features:**
    a. Renaming incorrect model categories.
    b. Removing irrelevant variables based on their correlation with the price.
2. **Detecting and dealing with anomalies:**
    a. Detecting anomalies using box plot and histogram plot.
    b. Detecting anomalies using Tukey’s method.
    c. Removing anomalous rows.
3. **Detecting and dealing with Not Assigned values:**
    a. Detecting variables containing not assigned values.
    b. Figuring out ways to fill them without data imputation.
    c. Filling those that couldn’t be figured out with mode (if they are categorial data) or median (if they are quantitative data).
4. **Finding the correlation between the features and figuring out insights from them.**
5. **Feature engineering:**
    a. Encoding categorial variables via One-Hot Encoding.
    b. Scaling quantitative variables using normalization
6. **Model Selection:**
    a. Splitting training data into 2 parts for evaluating different regression models.
    b. Testing different machine learning models.
    c. Selecting the appropriate model with the lowest RMSLE score.
7. **Training the selected model with the preprocessed training dataset.**
8. **Applying steps 1, 3, 5 on the testing dataset.**
9. **Predicting the price of the testing dataset entries**

Data Cleaning Process

First, we looked at the different car model labels, we found additional 2 classes whose name is the same as another 2 classes but with an additional dot. Since the classes contain 1 and 2 cars respectively, we believe that those classes were misspelled while entering their data.

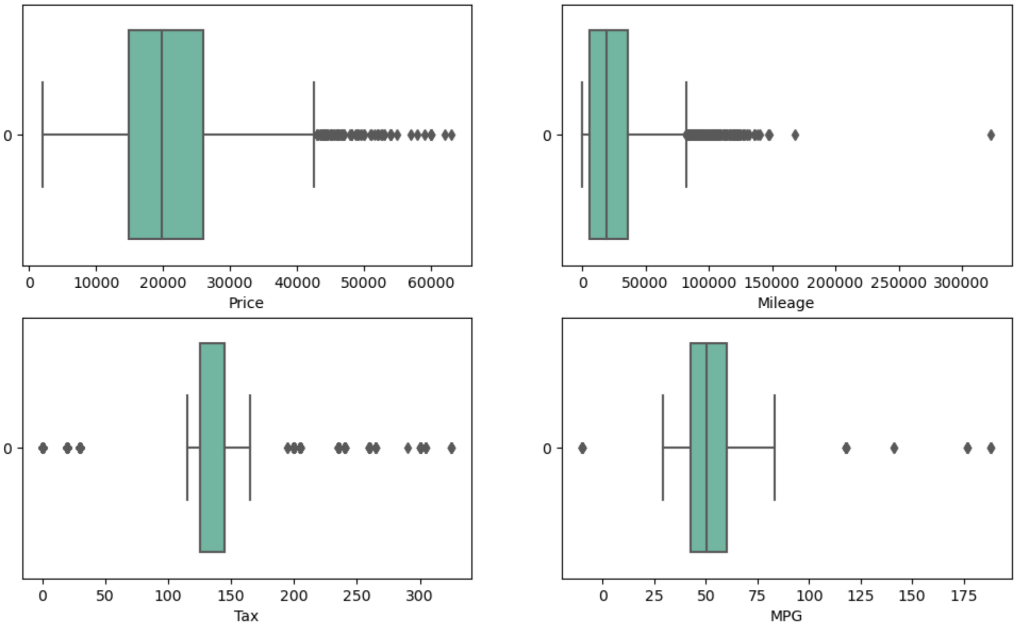
After that we used an implementation of the heatmap that can get the correlation of both categorial and quantitative data to get the correlation values among the dataset variables.

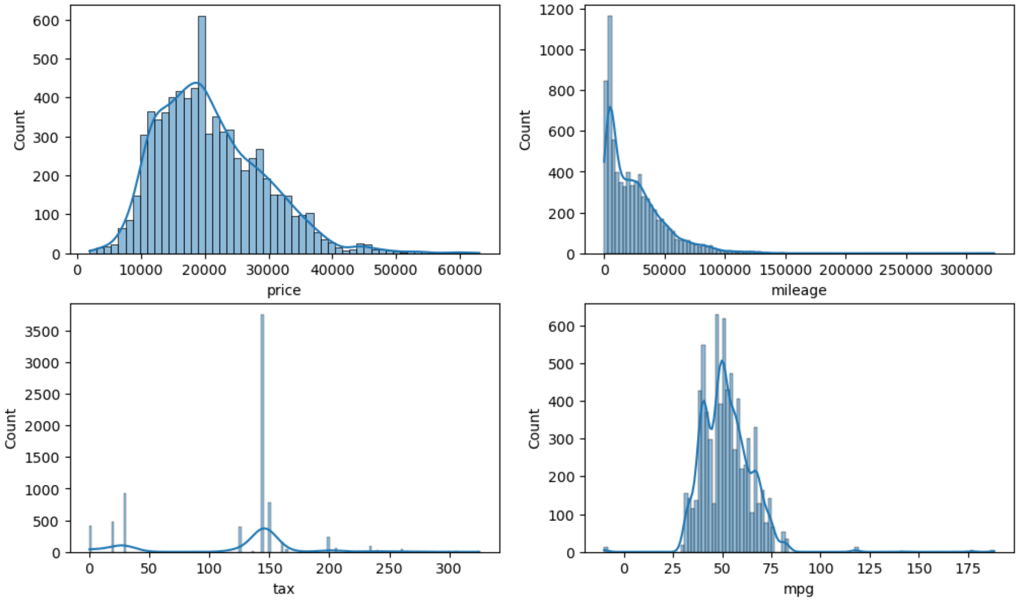


Looking at the **price** row, we noticed that there was nearly no correlation between the price and each of **ID**, **ownerName** and **state**, so we deemed these columns as irrelevant and removed them.

After that we checked if there exist empty entries in the dataset, and when we found empty entries in the **model**, **mileage** and **state**, we elected to search for outliers and deal with them first so that if we applied data imputation techniques on the empty entries, it won't be based on noisy data.

We searched for outliers using the box plot in **price**, **mileage**, **tax** and **mpg** and we found the following:





We investigated **tax** first, but looking at its histogram, we left it alone as the displayed histogram doesn't follow any distribution.

For **price**, **mileage**, and **mpg**, since their distribution shows skewness (especially for mileage and mpg as they show high positive skewness), we elected to use Tukey's method to identify the outliers where we choose to remove the entries that are outside the upper fence (3*IQR) as they have a very high probability of being outliers.

We chose to look at the data first to find logical behaviors that will save us from utilizing data imputation techniques on all the empty values, and we elected to use mode imputation to fill **model** and **state**, and since the mileage distribution is skewed, we chose median imputation to fill the not assigned entries.

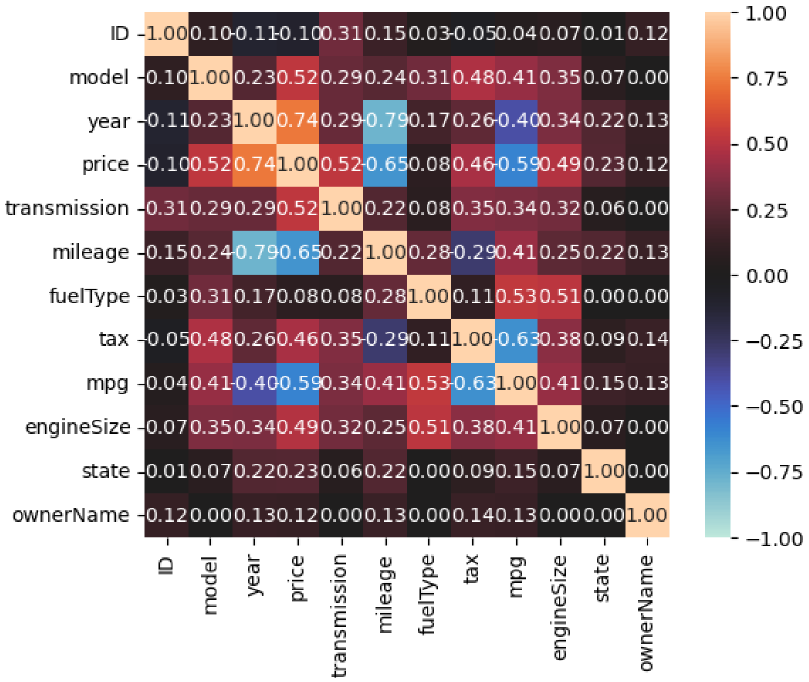
After that we started to deal with the Not Assigned values in **mileage**. After checking all the new cars in the dataset, we found their **mileage** entry to be 0, so for the new cars with not assigned **mileage** values, we filled these with 0, the rest was filled with medium imputation.

The inverse was done with not assigned values in **state**, where we assigned New to the rows with 0 mileage and not assigned state.

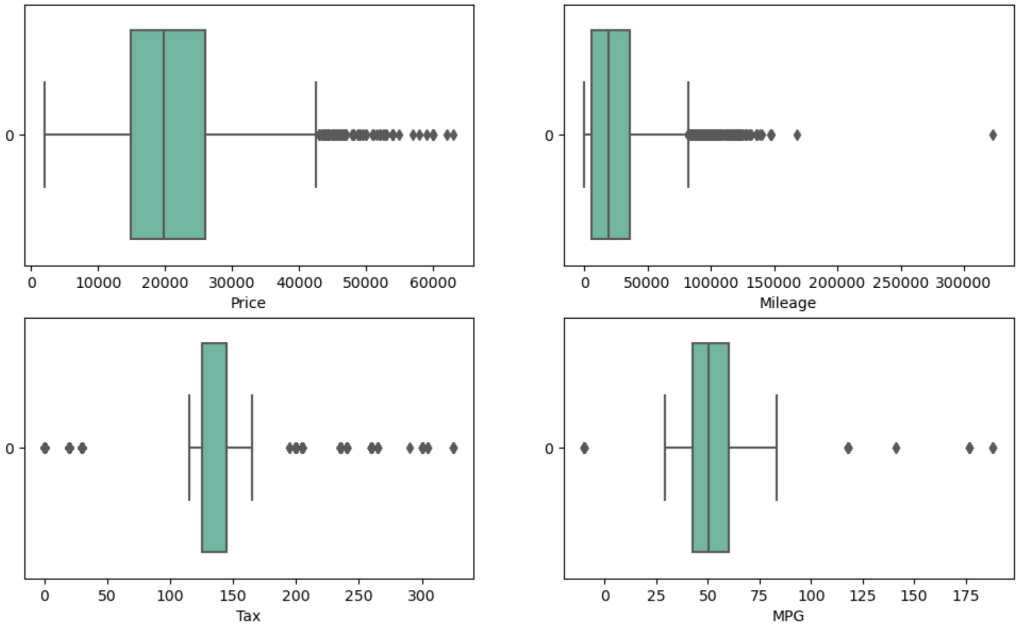
Since we couldn't find a certain pattern in the not assigned **model** entries, we just used mode imputation on them.

Data Visualization

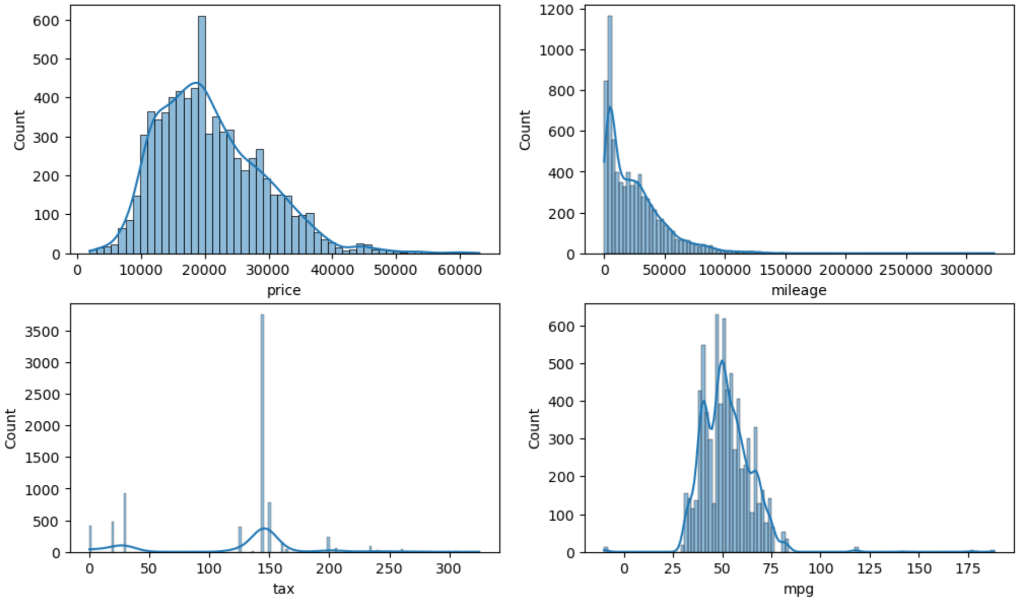
Heatmap to show the correlation between the different values and their relations with one another in **the training data**:



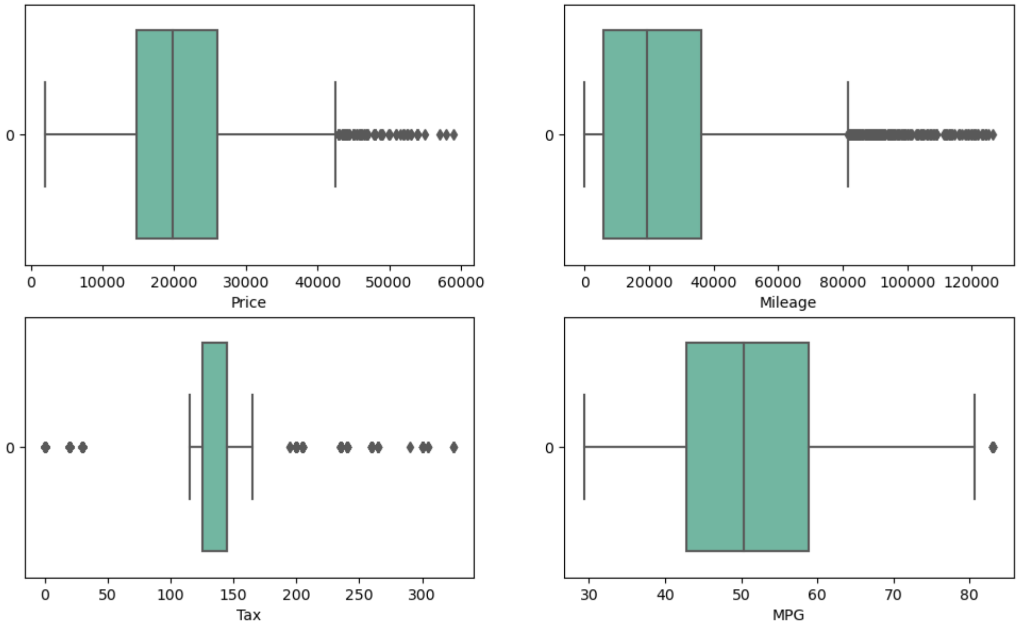
Box plot for the **price**, **mileage**, **tax**, and **mpg** to identify outliers:



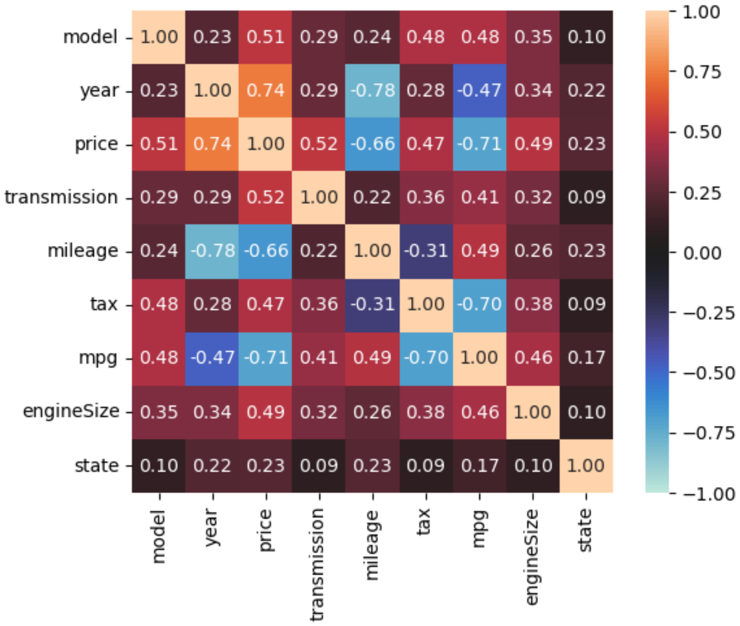
Histogram plots for **price**, **mileage**, **tax**, and **mpg** to view their distribution:



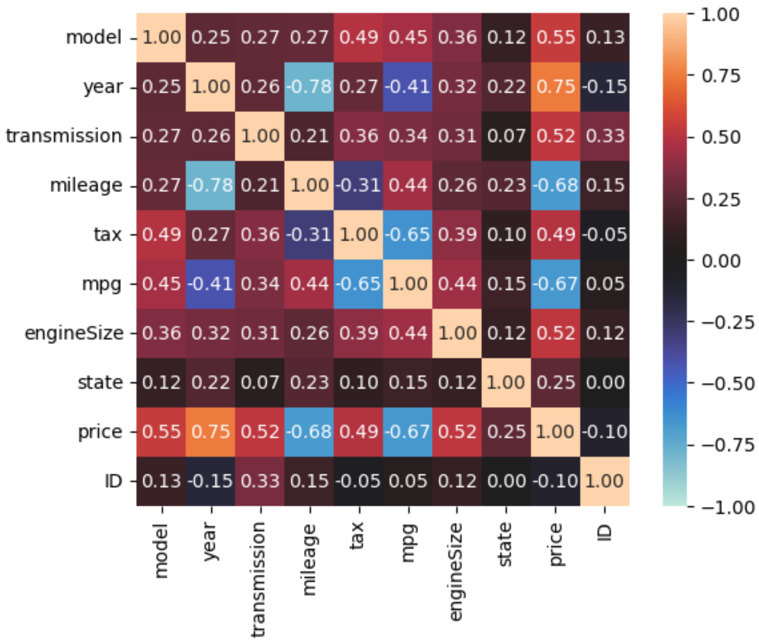
Box plot for the **price**, **mileage**, **tax**, and **mpg** after dealing with outliers:



Heatmap to show the correlation between the different values and their relations in **the training data** with one another after dealing with outliers:

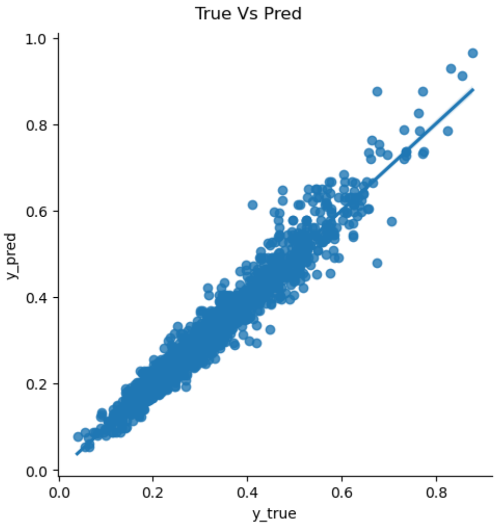


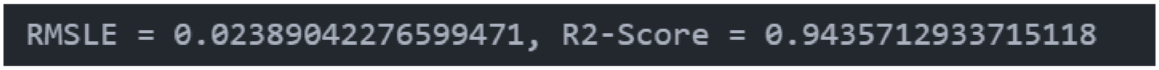
Heatmap to show the correlation between the different values and their relations with one another in **the testing data**:



Trained Models

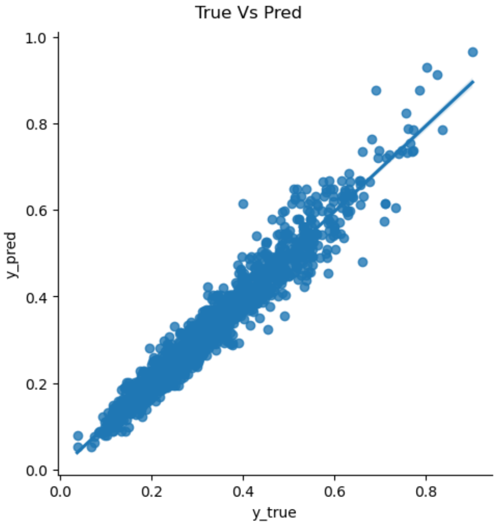
1. Gradient Boosting

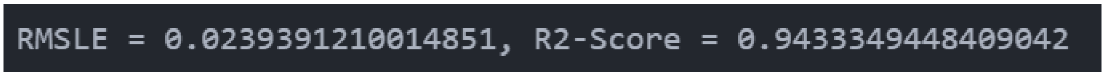




```
RMSLE = 0.02389042276599471, R2-Score = 0.9435712933715118
```

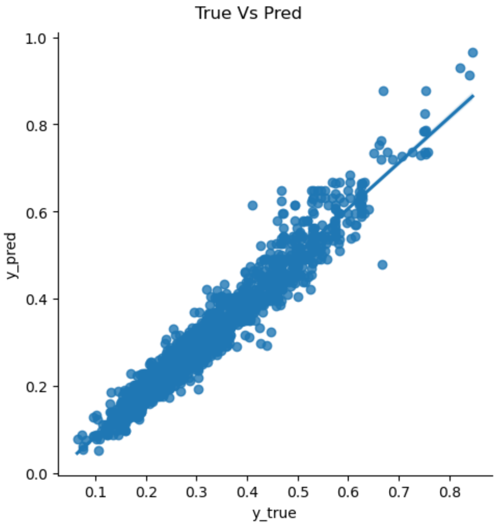
2. Random Forest Regressor

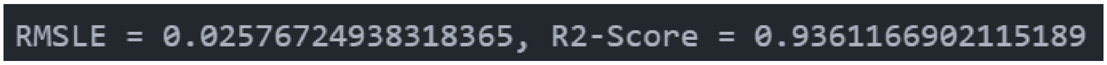




```
RMSLE = 0.023939121001485 1, R2-Score = 0.9433349448409042
```

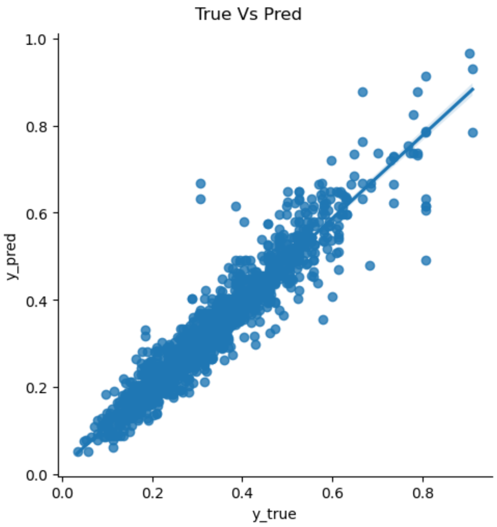
3. XGBoost Regression

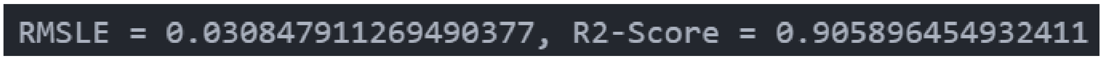




```
RMSLE = 0.02576724938318365, R2-Score = 0.9361166902115189
```

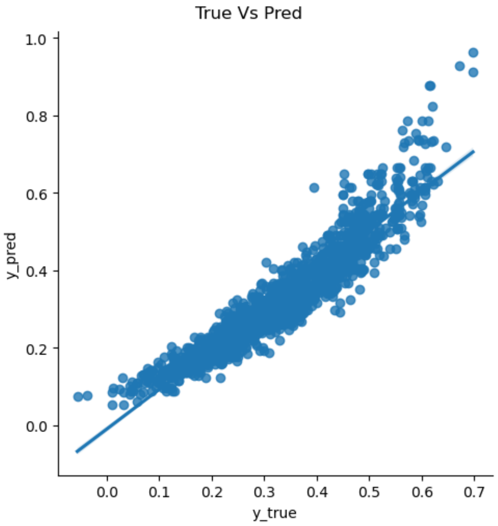
4. Decision Tree Regression

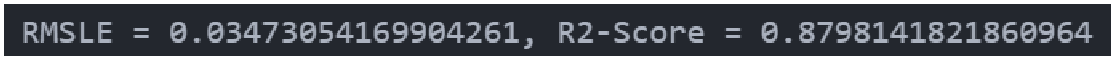




```
RMSLE = 0.030847911269490377, R2-Score = 0.905896454932411
```

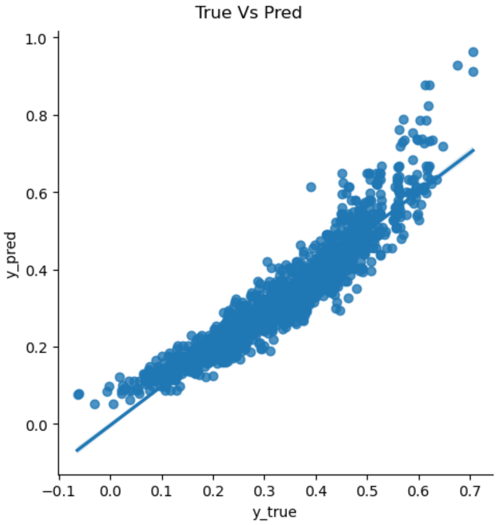
5. Gaussian Process Regression

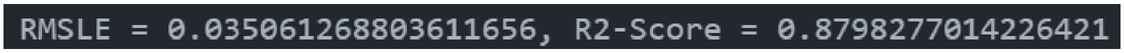




```
RMSLE = 0.03473054169904261, R2-Score = 0.8798141821860964
```

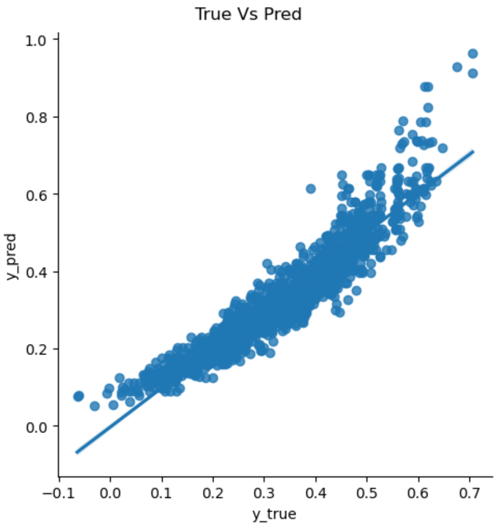
6. Linear Regression

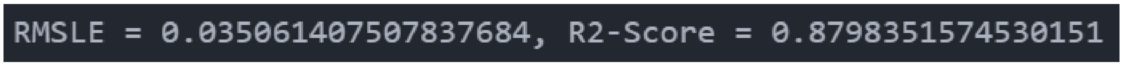




```
RMSLE = 0.035061268803611656, R2-Score = 0.8798277014226421
```

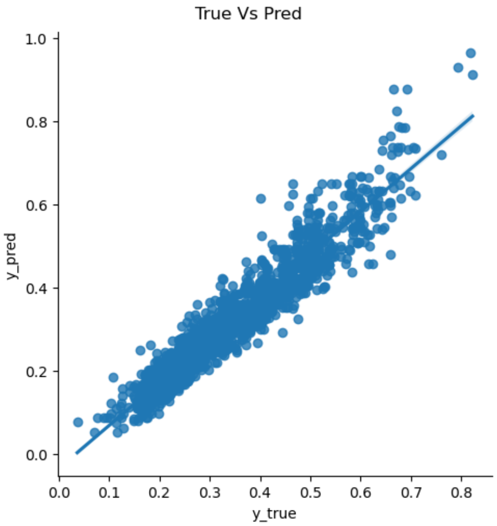
7. Random Sample Consensus Regression

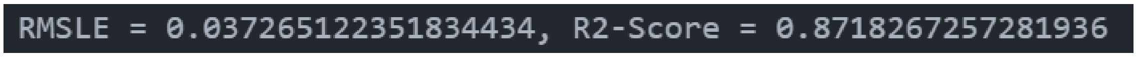




```
RMSLE = 0.035061407507837684, R2-Score = 0.8798351574530151
```

8. Support Vector Regression





```
RMSLE = 0.037265122351834434, R2-Score = 0.8718267257281936
```

Results and evaluations

Model	RMSLE	R2-SCORE
Gradient Boosting	0.023890422765994710	0.9435712933715118
Random Forest	0.02393912100148510	0.9433349448409042
XGBoost Regression	0.025767249383183650	0.9361166902115189
Decision Tree Regression	0.030847911269490377	0.9058964549324110
Gaussian Process Regression	0.034730541699042610	0.8798141821860964
Linear Regression	0.035061268803611656	0.8798277014226421
RANSAC	0.035061407507837684	0.8798351574530151
Support Vector Regression	0.037265122351834434	0.8718267257281936

The model we used is **Gradient Boosting**.