Electrical and Computer Engineering Department

Machine Learning and Data Science - ENCS5341

Assignment #2

Rana Musa 1210007

Leyan Burait 1211439

Contents

Table of figure

Dataset

https://www.kaggle.com/datasets/ahmedwaelnasef/cars-dataset/data

The dataset contains approximately 6,750 rows and 9 columns of data scraped from the YallaMotors website using Python's Requests-HTML library. This site provides detailed information about various car models. A great resource for automotive market analysis and predictive modeling, the dataset facilitates Exploratory Data Analysis (EDA) and machine learning tasks.

Features:

1. **Car Name**: The specific name or model of the car.
2. **Price**: The cost of the car, serving as the target variable for predictive modeling.
3. **Engine Capacity**: The size of the engine.
4. **Cylinder**: The number of cylinders in the car's engine.
5. **Horsepower**: The power output of the car's engine, measured in horsepower (HP).
6. **Top Speed**: The maximum speed the car can achieve, usually measured in kilometers or miles per hour.
7. **Seats**: The seating capacity of the car.
8. **Brand**: The manufacturer or make of the car (e.g., Toyota, Ford).

Applications and Objectives:

The primary goal of this dataset is to predict **car prices** using the provided features. This makes it ideal for building and testing **regression models** to uncover the relationships between the target variable (price) and other car attributes.

Processing steps:

First step was done is to standardize all prices to USD, to make prediction process more accurate since dataset includes multiple currencies.

7	Honda HR-	25422.67
8	Peugeot E	22092
9	Peugeot E	20412
10	Renault Kc	31173.33
11	Ford Bronc	63466.67
12	Suzuki Jimi	24502.67
13	Honda HR-	19289.33

Figure 1: Car name and prices in USD

Secondly, data cleaning was performed the missing fields was handled by imputation, by handling numeric fields with mean and non-numeric values with most frequent strategy

In all regression models, we used 5 features, 4 to train (engine capacity, horsepower, cylinder, top speed) and 1 as a target variable (price), we used these features to make it easier for the system to interact with numeric features to perform mathematical operations, we Split the data into training (60%), validation (20%), and test (20%) and handled any non-numeric value may accrue by dropping rows.

Next step was to Encode Categorical Variables by Converting non-numerical (categorical) data into numerical formats using one-hot encoding.

Model Performance on the Validation Set

1. linear regression

We performed linear regression using two methods, first in **closed-form solution**

The closed-form solution is a mathematical approach to solving the regression problem directly using the formula:

$$\theta = (X^T X)^{-1} X^T y$$

Closed form-sloution was implemented from scratch, The matrix operations were implemented using numpy.

Closed-form solution performance on the validation set

```
Model parameters (theta):
[-1.92497614e+05 -1.06888394e+01  8.72729571e+01  6.55449552e+02
  1.81830724e+04]
Validation MSE: 1218314719.41, R2: 0.73, MAE: 23518.04
```

Figure 2Closed-form solution performance on the validation set

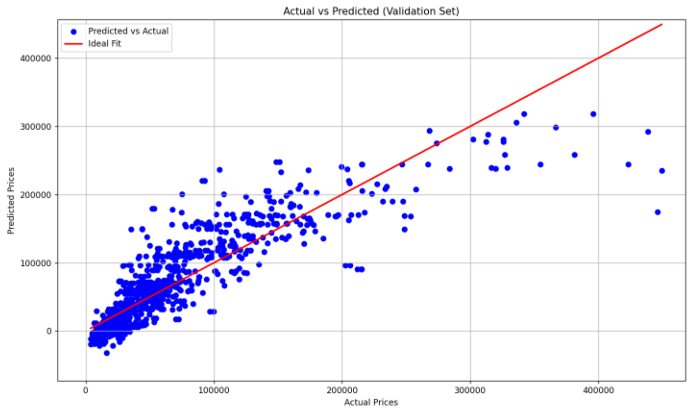


Figure 3plot the validation set Actual vs predicted

Second method to perform linear regression was **gradient-descent**

The gradient descent algorithm is implemented to iteratively minimize a cost function, with the learning rate (α) controlling the update steps. Multiple learning rates are tested, and the model's performance is evaluated on the validation set using metrics such as Mean Squared Error (MSE), Mean Absolute Error (MAE), and R2. The best-performing learning rate is selected based on the lowest validation MSE, and the corresponding model is used to generate predictions for the training, validation, and test sets.

gradient-descent performance on the validation set

```
Best alpha: 0.01
Best theta: [69859.07316712 -5005.36561924 17047.70395609 29944.73094392
 32912.39799834]
```

Figure 4: best result of alpha & theta on the validation set

```
Validation Set MSE: 1217904830.228055
Validation Set MAE: 23513.782712774104
Validation Set R²: 0.7295715181988087
```

Figure 5gradient-descent performance on the validation set

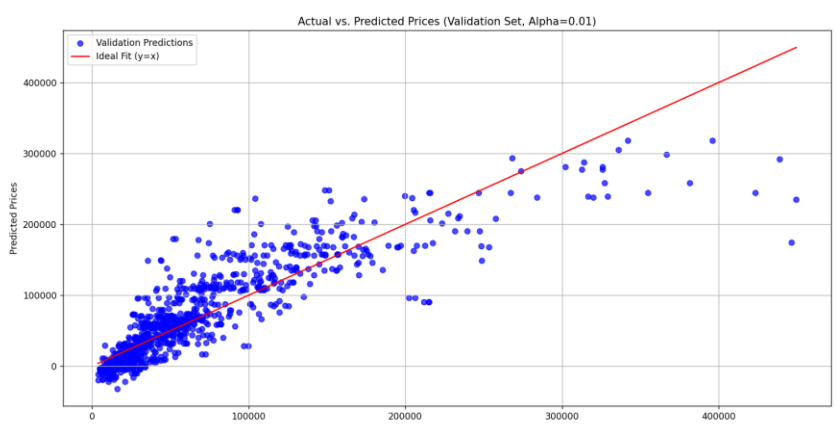


Figure 6:plot of validation set , Alph=0.01

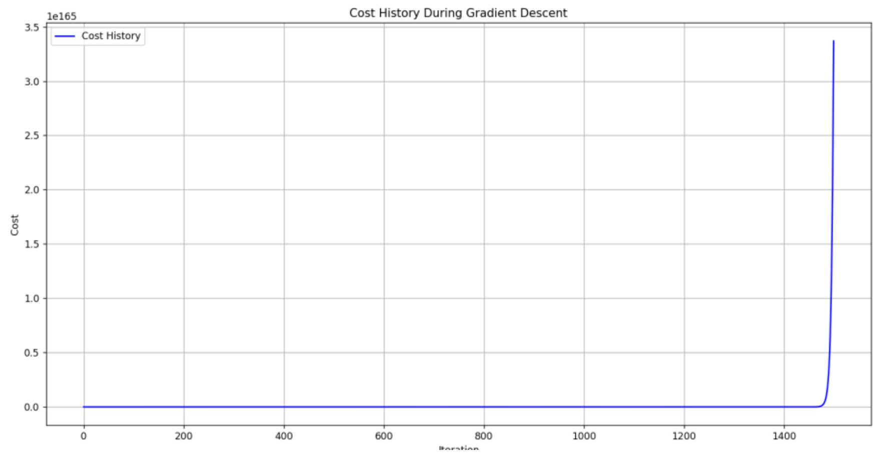


Figure 7: cost history Gradient Descent

As both closed-form solutions and gradient descent aim to minimize the same objective function, the Mean Squared Error (MSE), for linear regression, the results were almost identical. Gradient descent iteratively updates the parameters, resulting in the same solution, as opposed to a closed-form solution. Due to the iterative nature of gradient descent and potential floating-point precision, both methods ultimately produce nearly identical results. As a result of the iterative optimization of gradient descent, both methods produced comparable model performance metrics, confirming that the closed-form solution approximates the precise solution obtained by gradient descent.

2. LASSO regression

we implement a Lasso regression model with a forward feature selection process to predict car prices based on various features such as engine capacity, horsepower, top speed, and cylinder count. After loading and cleaning the dataset, the data is split into training, validation, and test sets. The forward selection process evaluates each feature's contribution to the model's performance by iteratively adding the feature that minimizes the Mean Squared Error (MSE) on the validation set. For each selected feature set, a GridSearchCV is used to find the optimal alpha parameter for Lasso regression, ensuring the model is regularized for the best generalization. After selecting the features, a final model is trained, and the performance is evaluated on the test set using MSE, Mean Absolute Error (MAE), and R^2 score.

LASSO regression performance on the validation set:

```
Adding feature: horse_power with MSE: 1418540779.8808167
Adding feature: top_speed with MSE: 1379055732.8839695
Adding feature: cylinder with MSE: 1240397180.220005
Adding feature: engine_capacity with MSE: 1218314718.1940608
Selected Features: ['horse_power', 'top_speed', 'cylinder', 'engine_capacity']
Best alpha found: 0.0001
Final MSE: 1657084375.5318346
Final MAE: 23967.271893575344
Final R^2: 0.7273241402230416
```

Figure 8: Lasso result all

The model's performance is visualized through feature importance (coefficients), residuals distribution, and a scatter plot comparing actual vs. predicted prices.

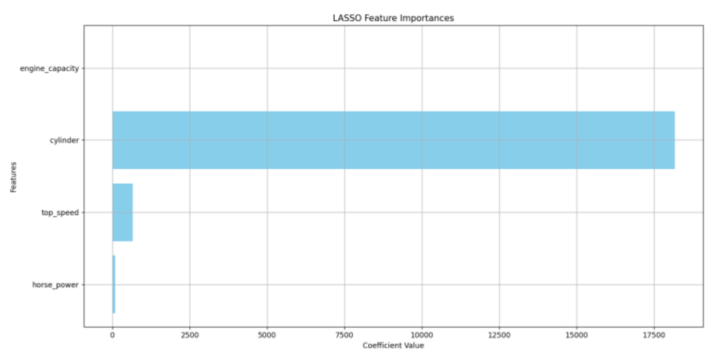


Figure 9: lasso feature importance

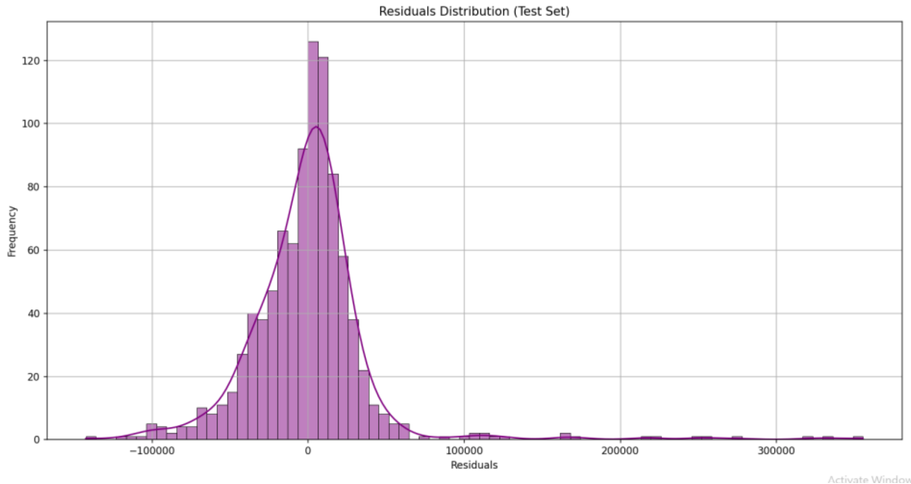


Figure 10: residuals distribution test set

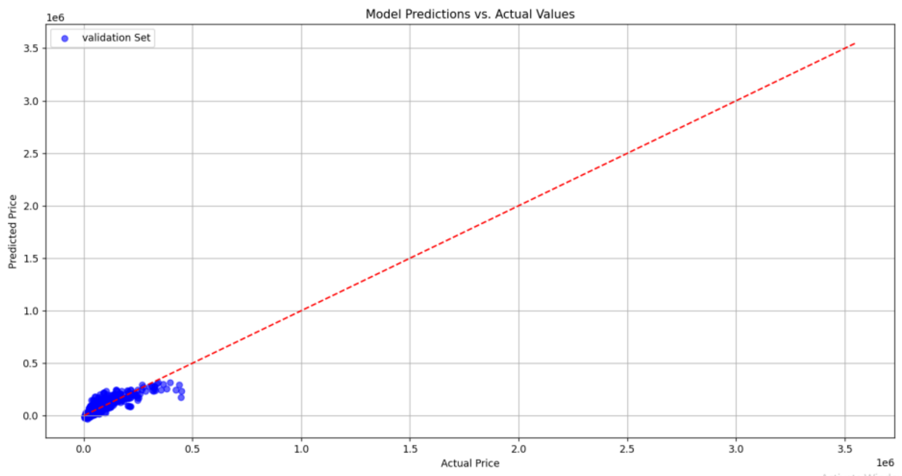


Figure 11: lasso prediction vs actual

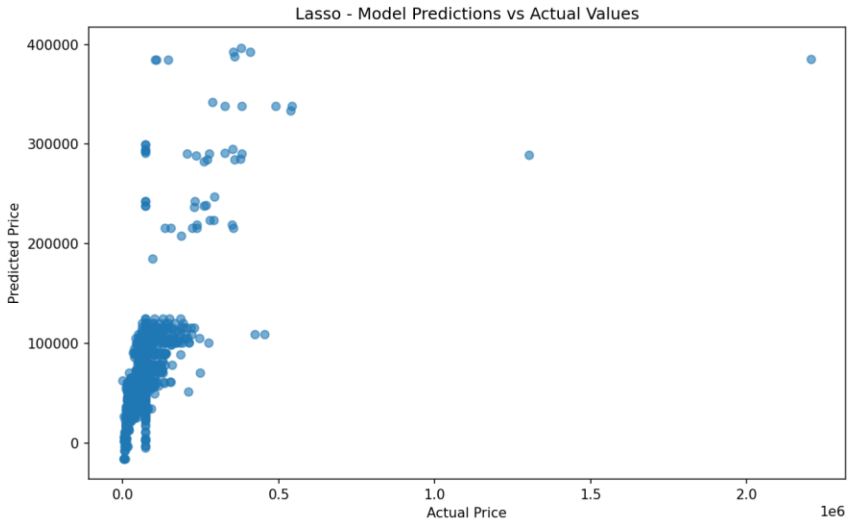


Figure 12another lasso prediction vs actual

3. Ridge Regression

Ridge regression with forward feature selection to predict car prices based on several car features such as engine capacity, horsepower, top speed, and cylinder count. The dataset is first loaded and cleaned by converting relevant columns to numeric and dropping rows with missing values. The data is then split into training, validation, and test sets. In the forward feature selection process, the model is iteratively trained with a set of features and evaluated on the validation set using mean squared error (MSE). The feature that leads to the lowest MSE is added to the model, and the process continues until no further improvement is found. After feature selection, GridSearchCV is used to identify the optimal alpha value for regularization. The Ridge regression model is then retrained using the selected features and best alpha value. Model performance is evaluated with MSE, mean absolute error (MAE), and R^2 scores for the validation set.

```
Selected features: ['horse_power', 'top_speed', 'cylinder', 'engine_capacity']
Best alpha found: 0.0001
Best Model - Validation MSE: 1218314711.679405
Best Model - Test MSE: 1657084372.5034263
Best Model - Validation MAE: 23518.036664144518
Best Model - Test MAE: 23967.27182411905
Best Model - Validation R^2: 0.7294805064745296
Best Model - Test R^2: 0.7273241407213711
```

Figure 13: result of Ridge Regression

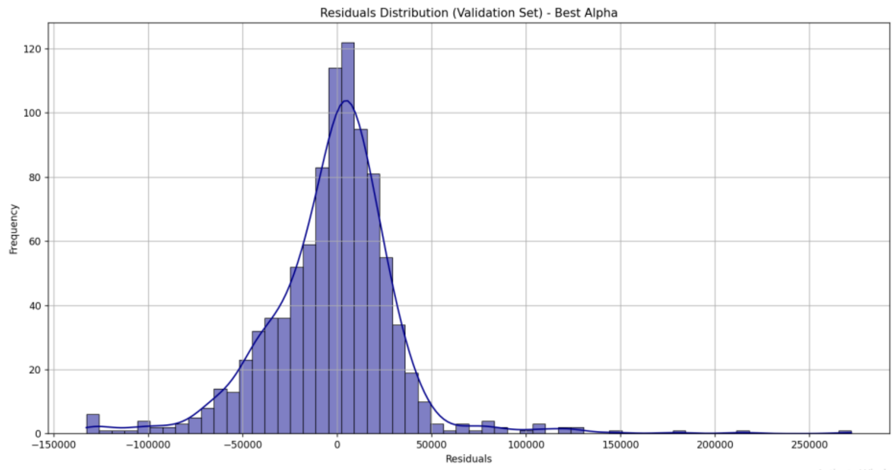


Figure 14result destribution

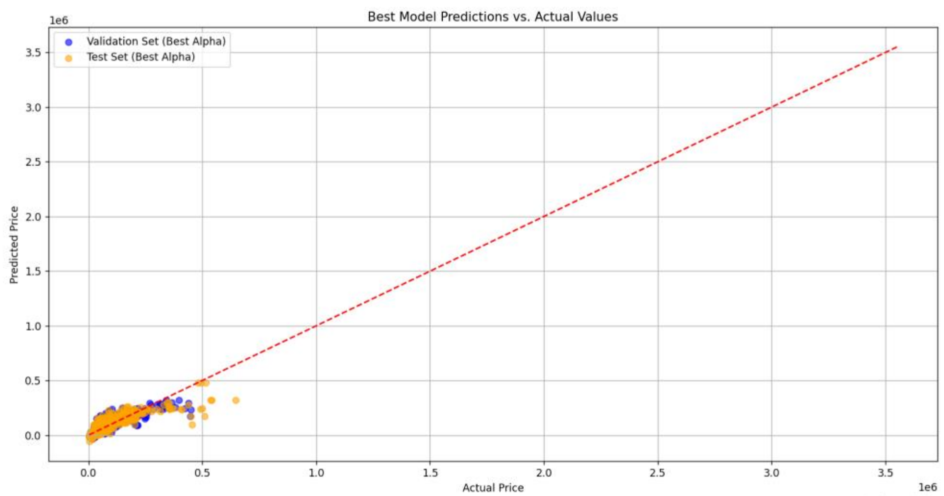


Figure 15: best prediction vs actual

4. Polynomial Regression

polynomial regression was used to predict car prices based on the 'cylinder' feature from a dataset. The process starts by loading the dataset and converting relevant columns to numeric values while handling missing values. The data is then split into training and test sets, with the features ('cylinder') normalized to a range between 0 and 1. The polynomial regression function is defined to transform the data into polynomial features of varying degrees (from 2 to 10), fit a linear regression model to the transformed data, and make predictions. The results are denormalized back to the original scale for meaningful interpretation.

```
Degree 2: MSE=5743374714.6465, MAE=31794.7202, R²=0.4032
Degree 3: MSE=5029925425.1765, MAE=34652.7305, R²=0.4774
Degree 2: MSE=5743374714.6465, MAE=31794.7202, R²=0.4032
Degree 3: MSE=5029925425.1765, MAE=34652.7305, R²=0.4774
Degree 4: MSE=3717251683.9193, MAE=31362.1129, R²=0.6138
Degree 3: MSE=5029925425.1765, MAE=34652.7305, R²=0.4774
Degree 4: MSE=3717251683.9193, MAE=31362.1129, R²=0.6138
Degree 4: MSE=3717251683.9193, MAE=31362.1129, R²=0.6138
Degree 5: MSE=3529474055.3973, MAE=30279.9309, R²=0.6333
Degree 6: MSE=3520694505.2796, MAE=30199.9111, R²=0.6342
Degree 7: MSE=3519952470.8197, MAE=30193.2095, R²=0.6343
Degree 8: MSE=3519952470.8197, MAE=30193.2094, R²=0.6343
Degree 9: MSE=3519952470.8197, MAE=30193.2094, R²=0.6343
Degree 10: MSE=3519952470.8197, MAE=30193.2095, R²=0.6343
```

Figure 16: result 4. Polynomial Regression

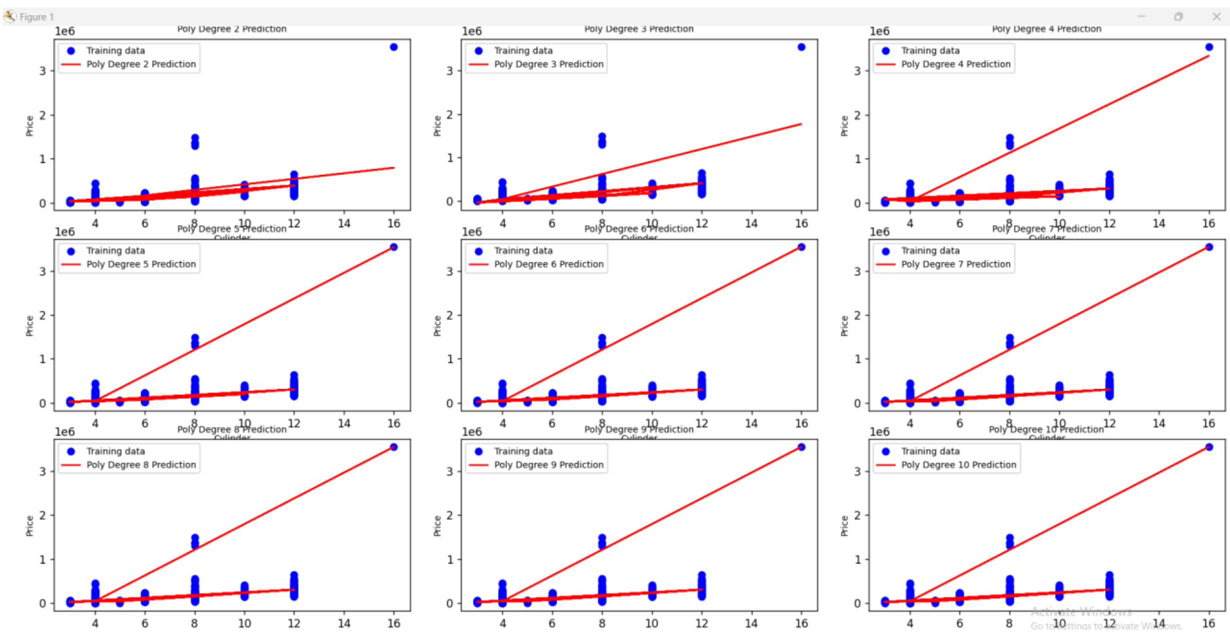


Figure 17: all plot of Polynomial Regression

5. RBF

The RBF is a powerful kernel function used in algorithms such as SVM to transform data into a higher space where classes can be separated using a linear interval as shown bellow figure. The RBF can handle non-linear data efficiently, making it suitable for many applications such as classification and regression.

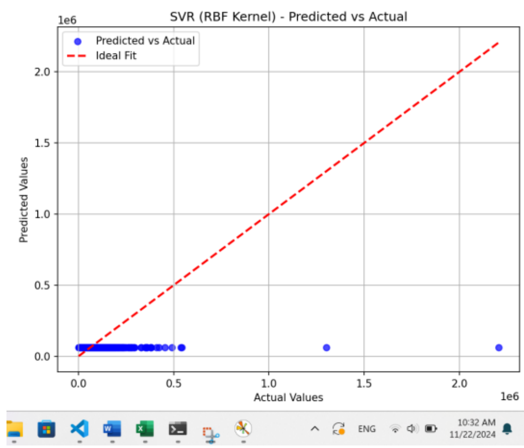


Figure 18:plot of RBF

```
[CV] END ....................................C=100, gamma=1; total time=   0.7s
RBF Kernel Model Performance:
MSE: 806740458.0224
MAE: 16330.4711
R2: 0.8209
```

Figure 19: result of RBF

Link of RBF: https://drive.google.com/file/d/1EGubqpP5f0f2uloLkOI2BTyLvX8FbhB3/view?usp=sharing

Link when check RBF with other model:
https://drive.google.com/file/d/1lZatLfVl2CFyI74iEJJiG9tSnL0ZmNE/view?usp=sharing

Test all model: https://drive.google.com/file/d/1dyBaxxH42Bra5wdB6xsrLQ7Qzy3ibkA/view?usp=sharing

Models performance:

Table 1 .table of all models performance

Regression Model	MSE	MAE	R^2
Closed-form solution	1218314719.41	23518.04	0.73
Gradient decent	1217904830.22	23513.78271	0.72957
LASSO	1657084375.53	23967.27189	0.727
Ridge	1218314711.67	23518.0366	0.7294
Polynomial(degree 10)	3519952470.81	30193.2095	0.6343
RBF	806740458.0224	16330.4711	0.8209

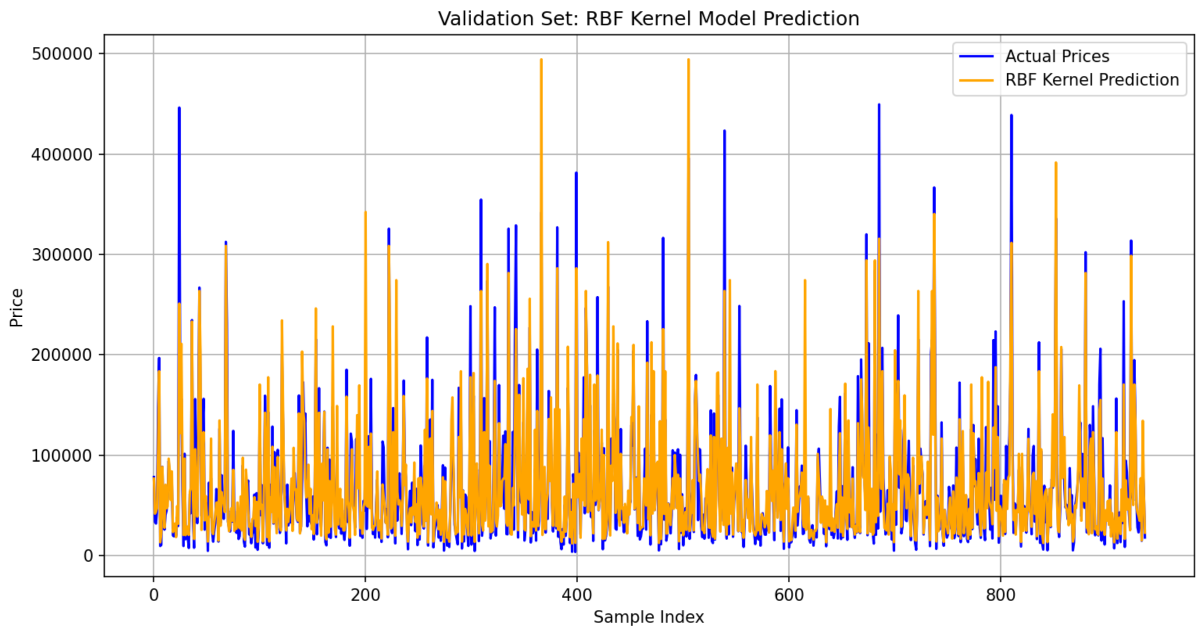


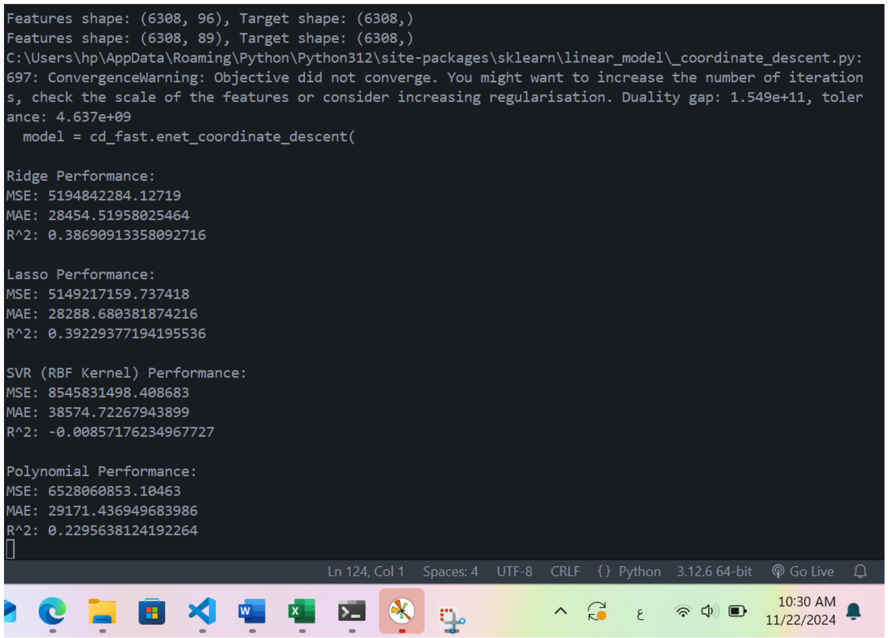
Figure 20:module comparisons

```
Model Performance Comparison:
Closed-Form Solution - MSE: 1218314719.4060, R2: 0.7295
Gradient Descent      - MSE: 1218314719.4060, R2: 0.7295
RBF Kernel (Tuned)    - MSE: 806740458.0224, R2: 0.8209
```

Figure 21: result of Regression Model MSE MAE R^2 of closed-form , Gradient & RBF

```
[CV] END .................................C=100, gamma=0.01; total time=   0.1s
[CV] END .................................C=100, gamma=0.01; total time=   0.1s
[CV] END .................................C=100, gamma=0.01; total time=   0.1s
[CV] END .................................C=100, gamma=0.01; total time=   0.1s
[CV] END ..................................C=100, gamma=0.1; total time=   0.2s
[CV] END ..................................C=100, gamma=0.1; total time=   0.2s
[CV] END ..................................C=100, gamma=0.1; total time=   0.2s
[CV] END ..................................C=100, gamma=0.1; total time=   0.2s
[CV] END ..................................C=100, gamma=0.1; total time=   0.2s
[CV] END ....................................C=100, gamma=1; total time=   0.5s
[CV] END ....................................C=100, gamma=1; total time=   0.6s
[CV] END ....................................C=100, gamma=1; total time=   0.5s
[CV] END ....................................C=100, gamma=1; total time=   0.6s
[CV] END ....................................C=100, gamma=1; total time=   0.5s
RBF Kernel Model Performance:
MSE: 806740458.0224
MAE: 16330.4711
R2: 0.8209
```

Figure 22result of Regression Model MSE MAE R^2 of closed-form , Gradient & RBF



```
Features shape: (6308, 96), Target shape: (6308,)
Features shape: (6308, 89), Target shape: (6308,)
C:\Users\hp\AppData\Roaming\Python\Python312\site-packages\sklearn\linear_model\_coordinate_descent.py:
697: ConvergenceWarning: Objective did not converge. You might want to increase the number of iteration
s, check the scale of the features or consider increasing regularisation. Duality gap: 1.549e+11, toler
ance: 4.637e+09
  model = cd_fast.enet_coordinate_descent(

Ridge Performance:
MSE: 5194842284.12719
MAE: 28454.51958025464
R^2: 0.38690913358092716

Lasso Performance:
MSE: 5149217159.737418
MAE: 28288.680381874216
R^2: 0.39229377194195536

SVR (RBF Kernel) Performance:
MSE: 8545831498.408683
MAE: 38574.72267943899
R^2: -0.00857176234967727

Polynomial Performance:
MSE: 6528060853.10463
MAE: 29171.436949683986
R^2: 0.2295638124192264
```

Figure 23: result of to Ridge, lasso, RBF & polynomial

When plot each form of above

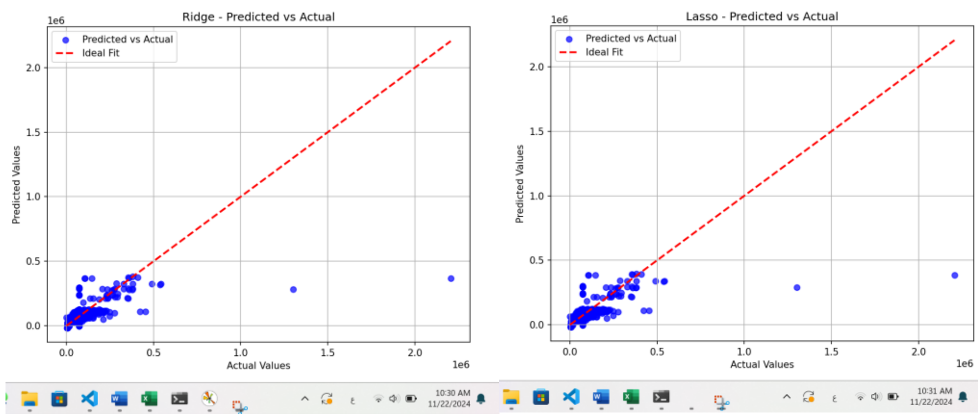


Figure 24: plot of Ridge & lasso

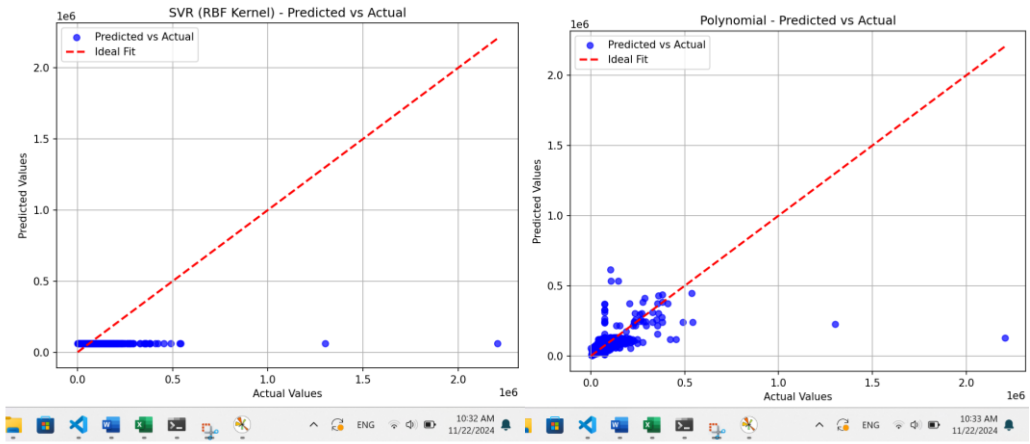


Figure 25:RBF & polynomial

7_ Best Model Based on R² and MSE:

Gradient decent

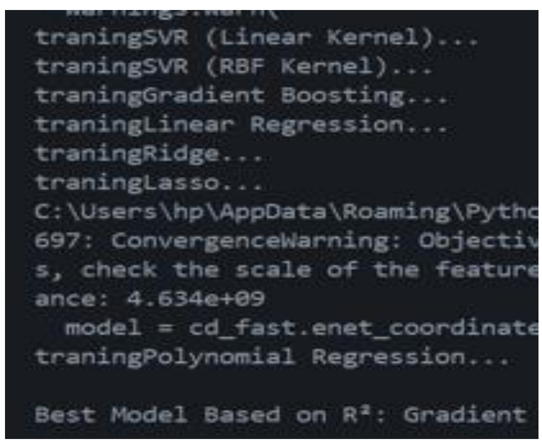


Figure 26Best Model Based on R²: Gradient decent

```
Number of NaN values in each column:
car name                    0
price                    1329
engine_capacity             3
cylinder                  110
horse_power               122
                          ...
country_country_ksa         0
country_country_kuwait      0
country_country_oman        0
country_country_qatar       0
country_country_uae         0
Length: 98, dtype: int64
     Alpha            MSE           MAE            R2                                              Theta
0  0.00001   8.947403e+09  6.843572e+04  -9.867173e-01  [1040.0715570684265, -152.02966051367775, 799....
1  0.00010   6.220880e+09  5.926161e+04  -3.813093e-01  [9731.265417186181, -1322.8089967208782, 6895....
2  0.00100   1.297664e+09  2.597179e+04   7.118614e-01  [54283.11654178715, -4671.788060980361, 21711....
3  0.01000   1.217905e+09  2.351378e+04   7.295715e-01  [69859.07316712322, -5005.365619240192, 17047....
4  0.10000   1.218315e+09  2.351804e+04   7.294805e-01  [69859.0929830705, -5006.073758204423, 17011.3...
5  1.00000  6.487871e+165  6.479323e+82 -1.440593e+156  [-4.0334752586281316e+66, 5.936887344249105e+8...
Best alpha: 0.01
Best theta: [69859.07316712 -5005.36561924 17047.70395609 29944.73094392
 32912.39799834]
Training Set MSE: 6526231785.335762
Training Set MAE: 25959.37579624138
Training Set R²: 0.41321481638237334

Validation Set MSE: 1217904830.228055
Validation Set MAE: 23513.782712774104
Validation Set R²: 0.7295715181988087
```

Figure 27:result of Gradient decent Performance of price attribute

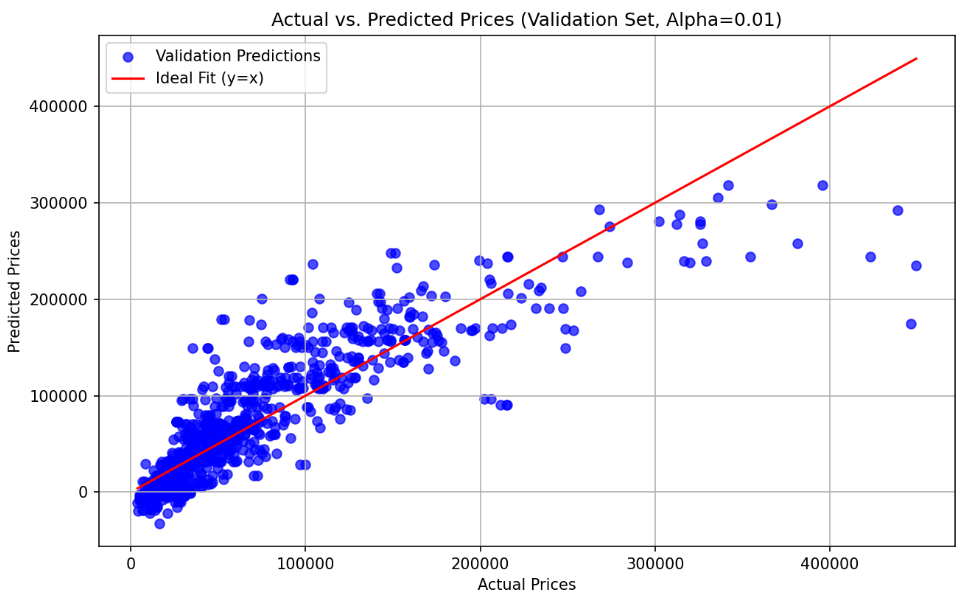


Figure 28: plot of Gradient actual vs predicted of price attribute

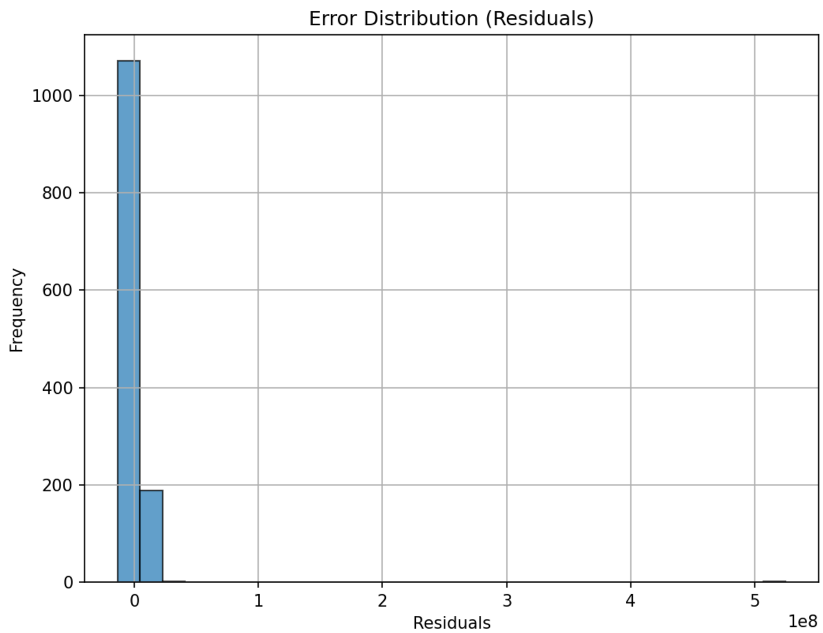


Figure 29: error distribution when apply Gradient descant of price attribute

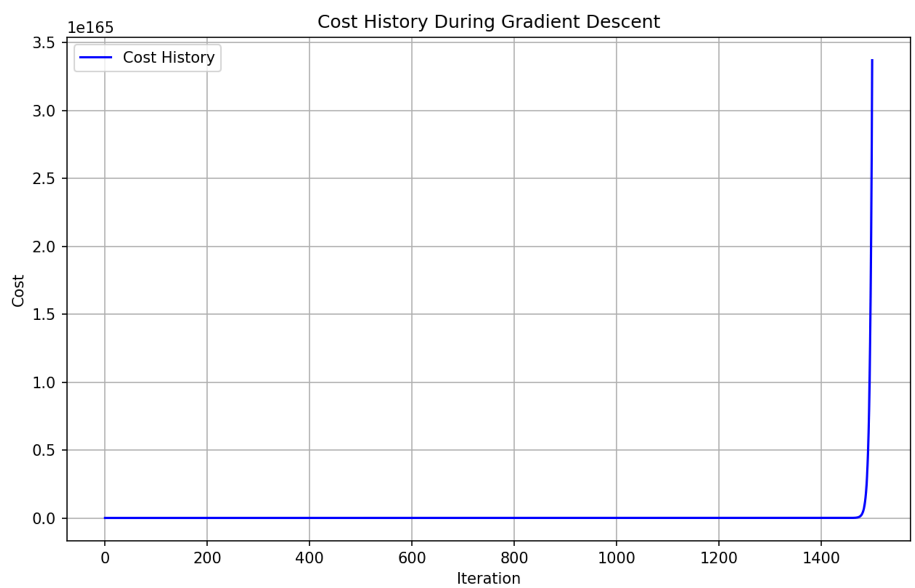


Figure 30:cost of price

Optional:

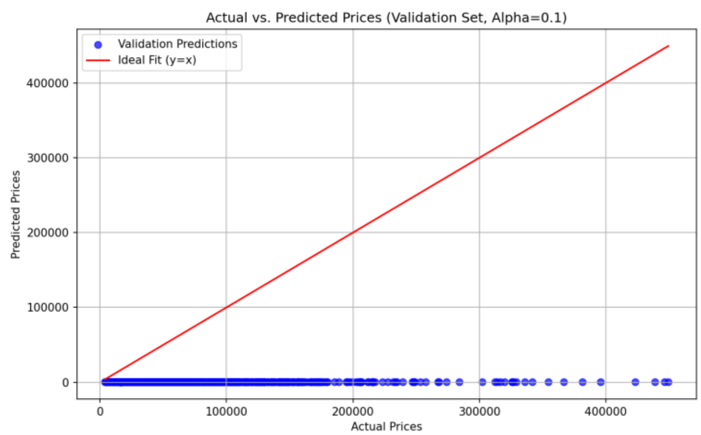
Try identifying another relevant target variable in the dataset and build a regression model to predict its values.

1_Top Speed

```
PS C:\Users\hp\Desktop\ML_Assignment2> & C:/Python312/python.exe c:/Users/hp/Desktop/ML_Assignment2/top_speed.PY
Number of NaN values in each column:
car name                   0
price                   1329
engine_capacity            3
cylinder                 110
horse_power              122
                        ...
country_country_ksa        0
country_country_kuwait     0
country_country_oman       0
country_country_qatar      0
country_country_uae        0
Length: 98, dtype: int64
     Alpha          MSE           MAE           R2                                              Theta
0  0.00001   9.338622e+09  6.953518e+04  -1.073585e+00  [3.306872460124896, -0.029413258240141873, 0.3...
1  0.00010   9.333753e+09  6.950743e+04  -1.072504e+00  [30.940230401994956, -0.22512891361846965, 3.0...
2  0.00100   9.311258e+09  6.936602e+04  -1.067509e+00  [172.59133943413482, 0.0008375091404601122, 6....
3  0.01000   9.304333e+09  6.931733e+04  -1.065971e+00  [222.11456853733722, 0.001139686370235579, 0.0...
4  0.10000   9.304333e+09  6.931733e+04  -1.065971e+00  [222.11463154147373, 1.0337004291384607e-16, 1...
5  1.00000  1.902627e+159  3.508774e+79 -4.224670e+149  [-3.1242519222113373e+63, 3.2150266267553655e+...
Best alpha: 0.1
Best theta: [2.22114632e+02 1.03370043e-16 1.52332414e-14 4.56967386e+01
 1.87563789e-14]
Training Set MSE: 1.276696308343367e-26
Training Set MAE: 1.0619945253881113e-13
Training Set R²: 1.0

Validation Set MSE: 9304332564.56008
Validation Set MAE: 69317.32719490644
Validation Set R²: -1.065971385577157
```

Figure 31: result of Top Speed



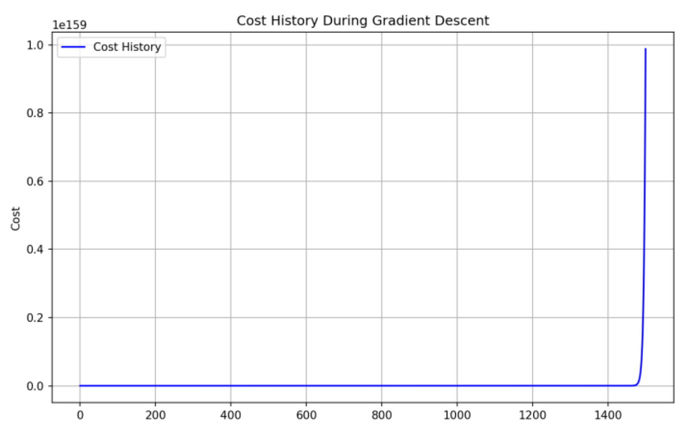


Figure 32: COST History during Gradient Descent

2_ horse power

```
PS C:\Users\hp\Desktop\ML_Assignment2> & C:/Python312/python.exe c:/Users/hp/Desktop/ML_Assignment2/hourse.py
Number of NaN values in each column:
car name                    0
price                    1329
engine_capacity             3
cylinder                  110
horse_power               122
country_country_kuwait      0
country_country_oman        0
country_country_qatar       0
country_country_uae         0
Length: 98, dtype: int64
     Alpha          MSE           MAE            R2                                              Theta
0  0.00001   9.338015e+09  6.953418e+04  -1.073450e+00  [4.171963694484405, -0.24164183229331476, 2.88...
1  0.00010   9.328407e+09  6.949810e+04  -1.071317e+00  [39.0343200388292, -1.9975890556668527, 25.880...
2  0.00100   9.290563e+09  6.931603e+04  -1.062914e+00  [217.7419331359552, -3.621342880574464, 121.39...
3  0.01000   9.282268e+09  6.925398e+04  -1.061072e+00  [280.22063963084923, -0.008662362717254766, 19...
4  0.10000   9.282277e+09  6.925399e+04  -1.061074e+00  [280.2207191171233, -8.711260201140043e-15, 19...
5  1.00000  4.065268e+160  1.621896e+80 -9.026684e+150  [-1.2903593347778092e+64, 1.4861146064985612e+...
Best alpha: 0.01
Best theta: [ 2.80220640e+02 -8.66236272e-03  1.94487052e+02  8.90021381e-02
  3.57061635e-01]
Training Set MSE: 0.11972640299586013
Training Set MAE: 0.1504322223076523
Training Set R²: 0.9999968488277394

Validation Set MSE: 9282267901.61238
Validation Set MAE: 69253.98023672812
Validation Set R²: -1.061072059164859
```

Figure 33: result of horse_power

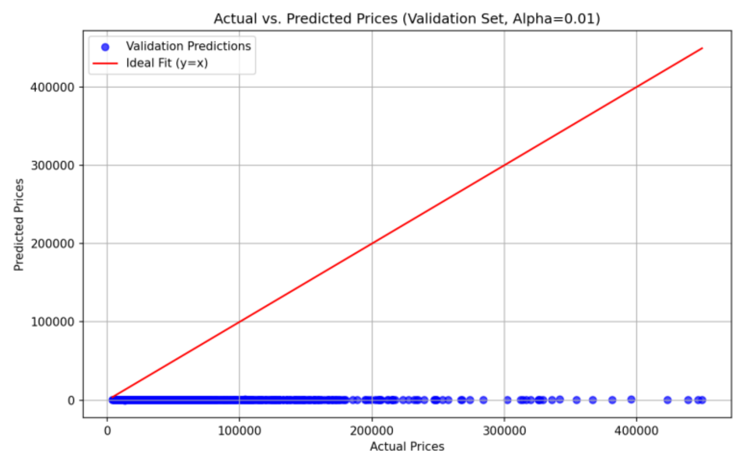


Figure 34: actual vs price of horse_power

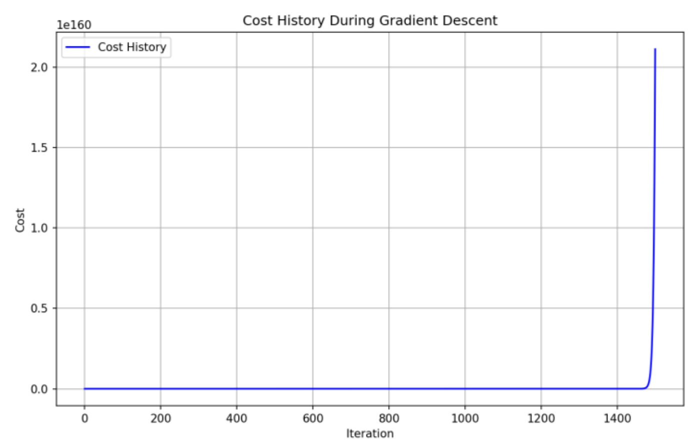


Figure 35: cost of horse_power

3_engine_capacity

```
Number of NaN values in each column:
car name                    0
price                    1329
engine_capacity             3
cylinder                  110
horse_power               122
                          ...
country_country_ksa         0
country_country_kuwait      0
country_country_oman        0
country_country_qatar       0
country_country_uae         0
Length: 98, dtype: int64
     Alpha           MSE           MAE            R2                                              Theta
0  0.00001  9.339292e+09  6.953710e+04  -1.073734e+00  [1.7185931506633245, 6.9720227721252535, -0.58...
1  0.00010  9.339761e+09  6.952527e+04  -1.073838e+00  [16.079745647887947, 65.17530892332815, -4.799...
2  0.00100  9.337496e+09  6.946336e+04  -1.073335e+00  [89.69632103806953, 362.77824633772957, -8.701...
3  0.01000  9.335021e+09  6.944105e+04  -1.072786e+00  [115.43371592153584, 468.34484069780655, -0.02...
4  0.10000  9.335022e+09  6.944105e+04  -1.072786e+00  [115.4337486650053, 468.345868390886, -1.36465...
5  1.00000  7.432983e+159  6.935218e+79 -1.650449e+150  [5.504904462052739e+63, -6.354615613472431e+78...
Best alpha: 0.01
Best theta: [ 1.15433716e+02  4.68344841e+02 -2.08134543e-02  1.16806455e-02
  1.00391800e-02]
Training Set MSE: 0.00026484248166067456
Training Set MAE: 0.006490663742801351
Training Set R²: 0.9999999987925914

Validation Set MSE: 9335021363.503477
Validation Set MAE: 69441.0478374109
Validation Set R²: -1.0727856497959887
```

Figure 36:RESULT OF engine_capacity

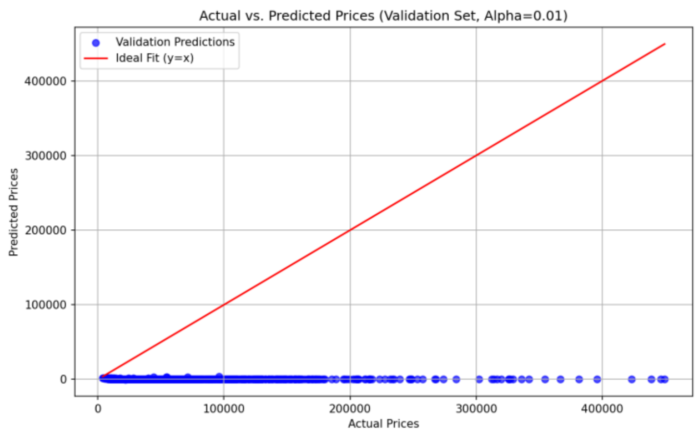


Figure 37actual vs producted of engine_capacity

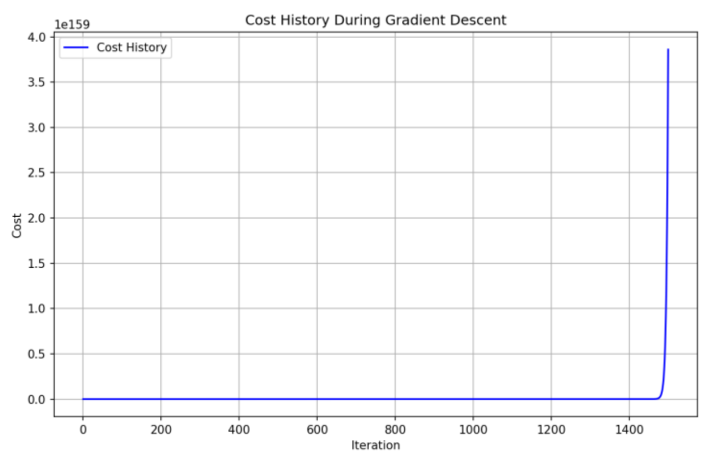
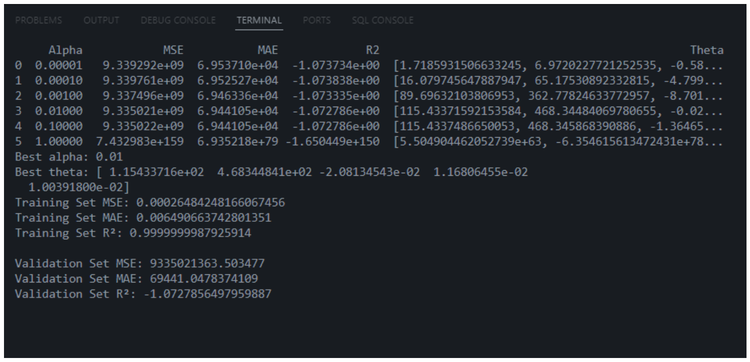


Figure 38: COST History during Gradient Descent

4_cylinder



```
PROBLEMS   OUTPUT   DEBUG CONSOLE   TERMINAL   PORTS   SQL CONSOLE

     Alpha           MSE           MAE            R2                                       Theta
0  0.00001  9.339292e+09  6.953710e+04  -1.073734e+00  [1.7185931506633245, 6.9720227721252535, -0.58...
1  0.00010  9.339761e+09  6.952527e+04  -1.073838e+00  [16.079745647887947, 65.17530892332815, -4.799...
2  0.00100  9.337496e+09  6.946336e+04  -1.073335e+00  [89.69632103806953, 362.77824633772957, -8.701...
3  0.01000  9.335021e+09  6.944105e+04  -1.072786e+00  [115.43371592153584, 468.34484069780655, -0.02...
4  0.10000  9.335022e+09  6.944105e+04  -1.072786e+00  [115.4337486650053, 468.345868390886, -1.36465...
5  1.00000 7.432983e+159  6.935218e+79 -1.650449e+150  [5.504904462052739e+63, -6.354615613472431e+78...
Best alpha: 0.01
Best theta: [ 1.15433716e+02  4.68344841e+02 -2.08134543e-02  1.16806455e-02
  1.00391800e-02]
Training Set MSE: 0.00026484248166067456
Training Set MAE: 0.006490663742801351
Training Set R²: 0.9999999987925914

Validation Set MSE: 9335021363.503477
Validation Set MAE: 69441.0478374109
Validation Set R²: -1.0727856497959887
```

Figure 39: result cylinder

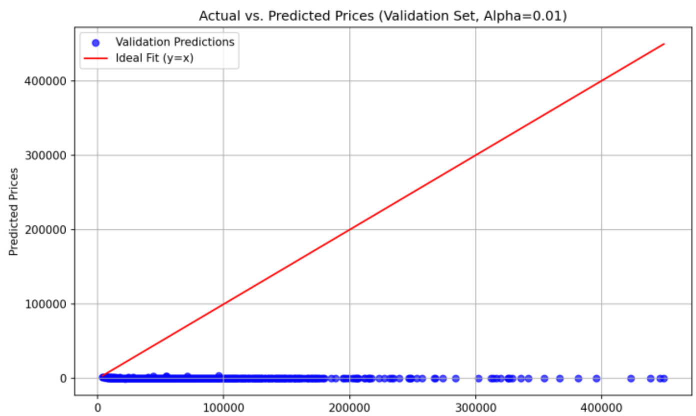


Figure 40actual vs prices cylinder

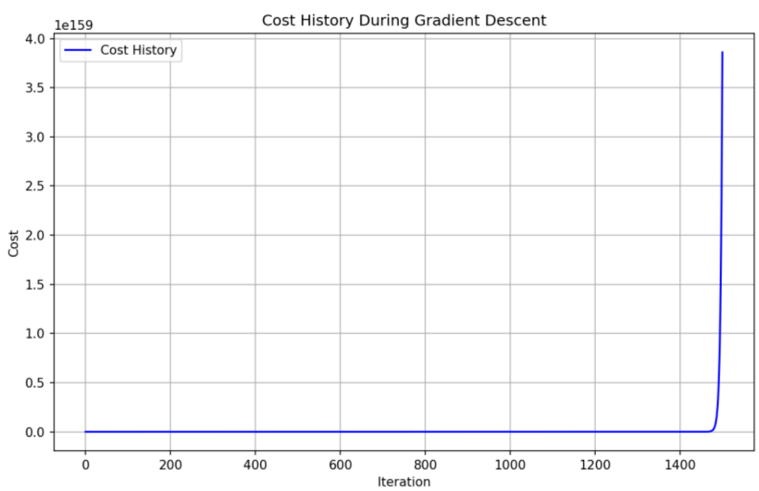


Figure 41:cost

Conclusion:

In this assignment, we applied regression analysis and model selection techniques to a real-world dataset. Using linear and nonlinear regression methods, we developed critical skills in data preprocessing, exploratory data analysis, and predictive modeling.

The project emphasized the importance of feature selection and regularization techniques for preventing overfitting and improving model performance. We gained a deeper understanding of foundational mathematical concepts in machine learning through the application of closed-form solutions and gradient descent for linear regression. Moreover, hyperparameter tuning with grid search ensured optimal model configuration, highlighting the iterative nature of machine learning.

We gained insights into assessing model generalization and performance by evaluating models on validation and test datasets. the inclusion of visualizations and a detailed report allowed us to effectively communicate our findings.

We gained knowledge and tools to effectively tackle real-world regression problems through this assignment, which bridged theoretical concepts and practical applications.

Teamwork distribution:

Rana Musa - 1210007	In this assignment, I implemented the linear regression using close- form solution, gradient descent, LASSO, Ridge and polynomial regression codes. In addition to report until point 5 (polynomial regression)
1211439 Leyan Burait	RBF & TEST OF IT POINT'S 7: PRICE BY GRADIENT _& 8 : TOP SPEED, engine_capacity & cylinder BY GRADIENT . report of RBF to end