Project Report

On

Regression of Used Car Prices

Submitted in partial fulfilment of the requirements for the award of

BACHELOR OF TECHNOLOGY

in

COMPUTER SCIENCE & ENGINEERING

(Artificial Intelligence & Machine Learning)

by

Ms. G NIKHITA (22WH1A6601)

Ms. G REVATHI (22WH1A6606)

Ms. R GEETIKA SRI (22WH1A6654)

Ms. B SRI VAISHNAVI (22WH1A6662)

Under the esteemed guidance of

Ms. A Naga Kalyani

Assistant Professor, CSE(AI&ML)

Department of Computer Science & Engineering

(Artificial Intelligence & Machine Learning)

BVRIT HYDERABAD COLLEGE OF ENGINEERING FOR WOMEN

(Approved by AICTE, New Delhi and Affiliated to JNTUH, Hyderabad)

Accredited by NBA and NAAC with A Grade

Bachupally, Hyderabad – 500090

2023-24

Department of Computer Science & Engineering

(Artificial Intelligence & Machine Learning)

BVRIT HYDERABAD COLLEGE OF ENGINEERING FOR WOMEN

(Approved by AICTE, New Delhi and Affiliated to JNTUH, Hyderabad)

Accredited by NBA and NAAC with A Grade

Bachupally, Hyderabad – 500090

2023-24

CERTIFICATE

This is to certify that the major project entitled **"Regression Of Used Car Prices"** is a bonafide work carried out by **Ms. G. Nikhita (22WH1A6601), Ms. G. Revathi (22WH1A6606)**, **Ms. R. Geetika Sri (22WH1A6654), Ms. B. Sri Vaishnavi (22WH1A6662**) in partial fulfilment for the award of B. Tech degree in **Computer Science & Engineering (AI&ML)**, **BVRIT HYDERABAD College of Engineering for Women, Bachupally, Hyderabad**, affiliated to Jawaharlal Nehru Technological University Hyderabad, Hyderabad under my guidance and supervision. The results embodied in the project work have not been submitted to any other University or Institute for the award of any degree or diploma.

Supervisor

Ms. A Naga Kalyani
Assistant Professor
Dept of CSE(AI&ML)

Head of the Department

Dr. B. Lakshmi Praveena
HOD & Professor
Dept of CSE(AI&ML)

External Examiner

DECLARATION

We hereby declare that the work presented in this project entitled **"Regression Of Used Car Prices"** submitted towards completion of Project work in III Year of B.Tech of CSE(AI&ML) at **BVRIT HYDERABAD College of Engineering for Women,** Hyderabad is an authentic record of our original work carried out under the guidance of **Ms. A Naga Kalyani, Assistant Professor, Department of CSE(AI&ML).**

Sign with Date:

G. Nikhita

(22WH1A6601)

Sign with Date:

G. Revathi

(22WH1A6606)

Sign with Date:

R. Geetika Sri

(22WH1A6654)

Sign with Date:

B. Sri Vaishnavi

(22WH1A6662)

ACKNOWLEDGEMENT

We would like to express our sincere thanks to **Dr. K. V. N. Sunitha, Principal, BVRIT HYDERABAD College of Engineering for Women**, for her support by providing the working facilities in the college.

Our sincere thanks and gratitude to **Dr. B. Lakshmi Praveena, Head of the Department, Department of CSE(AI&ML), BVRIT HYDERABAD College of Engineering for Women,** for all timely support and valuable suggestions during the period of our project.

We are extremely thankful to our Internal Guide, **Ms. A Naga Kalyani, Assistant Professor, CSE(AI&ML), BVRIT HYDERABAD College of Engineering for Women,** for her constant guidance and encouragement throughout the project.

Finally, we would like to thank our Major Project Coordinator, all Faculty and Staff of CSE(AI&ML) department who helped us directly or indirectly. Last but not least, we wish to acknowledge our **Parents** and **Friends** for giving moral strength and constant encouragement.

G. Nikhita (22WH1A6601)

G. Revathi (22WH1A6606)

R. Geetika Sri (22WH1A6654)

B. Sri Vaishnavi (22WH1A6662)

ABSTRACT

This project aims to develop a predictive model for estimating used car prices through exploratory data analysis (EDA) on a dataset containing detailed information about vehicles and their associated features. The dataset includes attributes such as make, model, year of manufacture, mileage, fuel type, engine size, and additional features influencing car prices. A thorough analysis of the data is performed using visualizations, statistical summaries, and pattern recognition to identify key factors driving vehicle valuation. The project employs techniques such as data cleaning, handling missing values, feature engineering, and scaling to prepare the dataset for accurate predictive modelling. Machine learning algorithms, including linear regression, random forests, and gradient boosting methods, are applied to predict car prices effectively. The outcomes of this project aim to provide insights into market dynamics, assist buyers and sellers in making informed decisions, and enhance understanding of factors affecting used car pricing.

PROBLEM STATEMENT

To develop a highly accurate and reliable regression model for predicting the price of used cars, leveraging advanced machine learning techniques to analyze a diverse range of vehicle features and historical data. This model will examine key attributes such as the make, model, year of manufacture, mileage, fuel type, engine capacity, overall condition, and market trends to uncover complex patterns and relationships that influence resale value. By integrating these insights, the model will provide a robust and intelligent pricing solution, catering to both buyers and sellers in the used car market. Buyers can utilize this tool to make well-informed purchasing decisions based on fair market evaluations, while sellers can set competitive prices that reflect the true value of their vehicles. This approach aims to bring greater transparency, accuracy, and efficiency to the used car market, addressing the challenges of price disparities and subjective assessments while fostering trust and confidence among stakeholders.

DATASET

car_ID	symboling	CarName	fueltype	aspiration	doornumb	carbody	drivewhee	enginelocа	wheelbase	carlength	carwidth	carheight	curbweigh	enginetyp
1	3	alfa-rome	gas	std	two	convertibl	rwd	front	88.6	168.8	64.1	48.8	2548	dohc
2	3	alfa-rome	gas	std	two	convertibl	rwd	front	88.6	168.8	64.1	48.8	2548	dohc
3	1	alfa-rome	gas	std	two	hatchback	rwd	front	94.5	171.2	65.5	52.4	2823	ohcv
4	2	audi 100 l	gas	std	four	sedan	fwd	front	99.8	176.6	66.2	54.3	2337	ohc
5	2	audi 100ls	gas	std	four	sedan	4wd	front	99.4	176.6	66.4	54.3	2824	ohc
6	2	audi fox	gas	std	two	sedan	fwd	front	99.8	177.3	66.3	53.1	2507	ohc
7	1	audi 100ls	gas	std	four	sedan	fwd	front	105.8	192.7	71.4	55.7	2844	ohc
8	1	audi 5000	gas	std	four	wagon	fwd	front	105.8	192.7	71.4	55.7	2954	ohc
9	1	audi 4000	gas	turbo	four	sedan	fwd	front	105.8	192.7	71.4	55.9	3086	ohc
10	0	audi 5000	gas	turbo	two	hatchback	4wd	front	99.5	178.2	67.9	52	3053	ohc
11	2	bmw 320i	gas	std	two	sedan	rwd	front	101.2	176.8	64.8	54.3	2395	ohc
12	0	bmw 320i	gas	std	four	sedan	rwd	front	101.2	176.8	64.8	54.3	2395	ohc
13	0	bmw x1	gas	std	two	sedan	rwd	front	101.2	176.8	64.8	54.3	2710	ohc
14	0	bmw x3	gas	std	four	sedan	rwd	front	101.2	176.8	64.8	54.3	2765	ohc
15	1	bmw z4	gas	std	four	sedan	rwd	front	103.5	189	66.9	55.7	3055	ohc
16	0	bmw x4	gas	std	four	sedan	rwd	front	103.5	189	66.9	55.7	3230	ohc
17	0	bmw x5	gas	std	two	sedan	rwd	front	103.5	193.8	67.9	53.7	3380	ohc
18	0	bmw x3	gas	std	four	sedan	rwd	front	110	197	70.9	56.3	3505	ohc
19	2	chevrolet	gas	std	two	hatchback	fwd	front	88.4	141.1	60.3	53.2	1488	l
20	1	chevrolet	gas	std	two	hatchback	fwd	front	94.5	155.9	63.6	52	1874	ohc

DataSet Link: DataSet

Code

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn.preprocessing import StandardScaler, LabelEncoder
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LinearRegression
from sklearn.metrics import mean_squared_error,r2_score
import warnings
warnings.filterwarnings('ignore')
```

Code

```
df = pd.read_csv('/CarPrice_Assignment.csv')
```

Code

```
df.isnull().sum()
```

Output

car_ID	0
symboling	0
CarName	0
fueltype	0
aspiration	0
doornumber	0
carbody	0
drivewheel	0
enginelocation	0
wheelbase	0
carlength	0
carwidth	0
carheight	0

dtype: int64

Code

df.duplicated().sum()

Output

```
0
```

Code

df.dtypes

Output

car_ID	int64
symboling	int64
CarName	object
fueltype	object
aspiration	object
doornumber	object
carbody	object
drivewheel	object
enginelocation	object
wheelbase	float64
carlength	float64
carwidth	float64
carheight	float64

dtype: object

Code

df.nunique()

Output

car_ID	205
symboling	6
CarName	147
fueltype	2
aspiration	2
doornumber	2
carbody	5
drivewheel	3
enginelocation	2
wheelbase	53
carlength	75
carwidth	44

dtype: int64

Code

df.describe()

Output

	car_ID	symboling	wheelbase	carlength	carwidth	carheight	curbweight	enginesize
count	205.000000	205.000000	205.000000	205.000000	205.000000	205.000000	205.000000	205.000000
mean	103.000000	0.834146	98.756585	174.049268	65.907805	53.724878	2555.565854	126.907317
std	59.322565	1.245307	6.021776	12.337289	2.145204	2.443522	520.680204	41.642693
min	1.000000	-2.000000	86.600000	141.100000	60.300000	47.800000	1488.000000	61.000000
25%	52.000000	0.000000	94.500000	166.300000	64.100000	52.000000	2145.000000	97.000000
50%	103.000000	1.000000	97.000000	173.200000	65.500000	54.100000	2414.000000	120.000000
75%	154.000000	2.000000	102.400000	183.100000	66.900000	55.500000	2935.000000	141.000000
max	205.000000	3.000000	120.900000	208.100000	72.300000	59.800000	4066.000000	326.000000

Code

```
categorical_columns = ['fueltype','aspiration','doornumber','carbody','drivewheel','enginelocation','enginetype',
    'cylindernumber',
    'fuelsystem'
]
for col in categorical_columns:
    print(f"Category in {col} is : {df[col].unique()}")
```

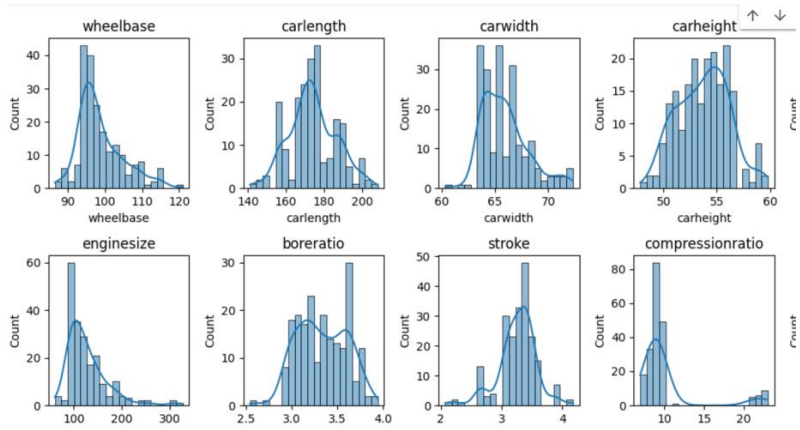
Output

```
Category in fueltype is : ['gas' 'diesel']
Category in aspiration is : ['std' 'turbo']
Category in doornumber is : ['two' 'four']
Category in carbody is : ['convertible' 'hatchback' 'sedan' 'wagon' 'hardtop']
Category in drivewheel is : ['rwd' 'fwd' '4wd']
Category in enginelocation is : ['front' 'rear']
Category in enginetype is : ['dohc' 'ohcv' 'ohc' 'l' 'rotor' 'ohcf' 'dohcv']
Category in cylindernumber is : ['four' 'six' 'five' 'three' 'twelve' 'two' 'eight']
Category in fuelsystem is : ['mpfi' '2bbl' 'mfi' '1bbl' 'spfi' '4bbl' 'idi' 'spdi']
```

Code

```
numerical_features = ['wheelbase', 'carlength', 'carwidth', 'carheight', 'curbweight',
                      'enginesize', 'boreratio', 'stroke', 'compressionratio', 'horsepower',
                      'peakrpm', 'citympg', 'highwaympg', 'price']
plt.figure(figsize=(12, 8))
for feature in numerical_features:
    plt.subplot(3, 5, numerical_features.index(feature) + 1)
    sns.histplot(data=df[feature], bins=20, kde=True)
    plt.title(feature)
plt.tight_layout()
plt.show()
```

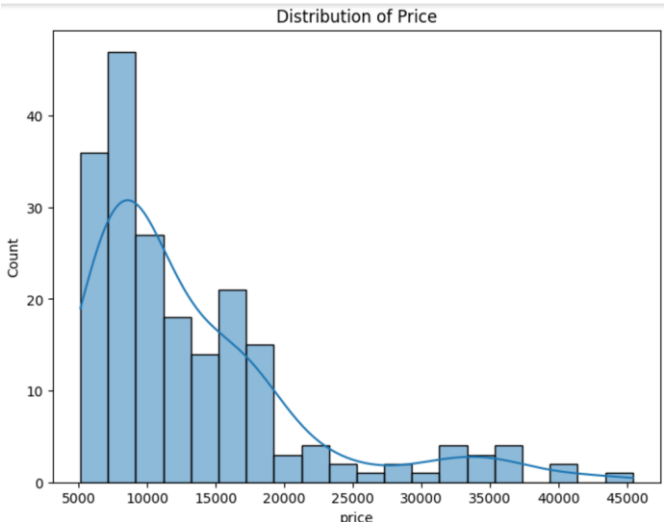
Output



Code

```
plt.figure(figsize=(8, 6))
sns.histplot(data=df['price'], bins=20, kde=True)
plt.title('Distribution of Price')
plt.show()
```

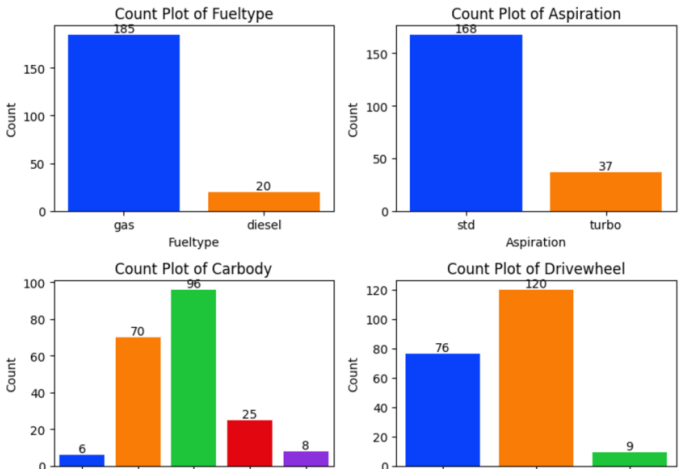
Output



Code

```
categorical_columns = ['fueltype', 'aspiration', 'doornumber', 'carbody',
'drivewheel',
                       'enginelocation', 'enginetype', 'cylindernumber', 'fuelsystem']
fig, axes = plt.subplots(nrows=3, ncols=3, figsize=(12, 9))
axes = axes.ravel()
for i, column in enumerate(categorical_columns):
    sns.countplot(x=df[column], data=df, palette='bright', ax=axes[i],
saturation=0.95)
    for container in axes[i].containers:
        axes[i].bar_label(container, color='black', size=10)
    axes[i].set_title(f'Count Plot of {column.capitalize()}')
    axes[i].set_xlabel(column.capitalize())
    axes[i].set_ylabel('Count')
plt.tight_layout()
plt.show()
```

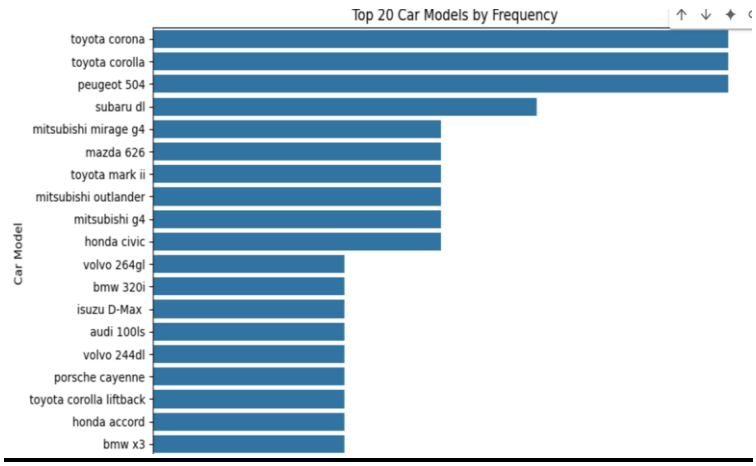
Output



Code

```
n = 20
top_car_models = df['CarName'].value_counts().head(n)
plt.figure(figsize=(10, 6))
sns.barplot(x=top_car_models.values, y=top_car_models.index)
plt.title(f'Top {n} Car Models by Frequency')
plt.xlabel('Frequency')
plt.ylabel('Car Model')
plt.tight_layout()
plt.show()
```

Output

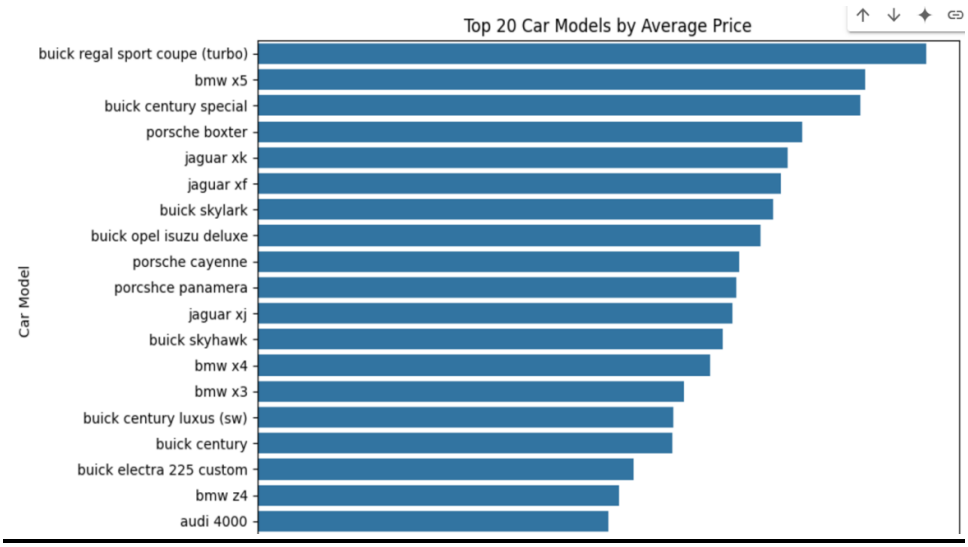


Code

```
avg_prices_by_car = df.groupby('CarName')['price'].mean().sort_values(ascending=False)
top_car_models = avg_prices_by_car.head(n)
plt.figure(figsize=(10, 6))
sns.barplot(x=top_car_models.values, y=top_car_models.index)
plt.title(f'Top {n} Car Models by Average Price')
```

```
plt.xlabel('Average Praice')
plt.ylabel('Car Model')
plt.tight_layout()
plt.show()
```

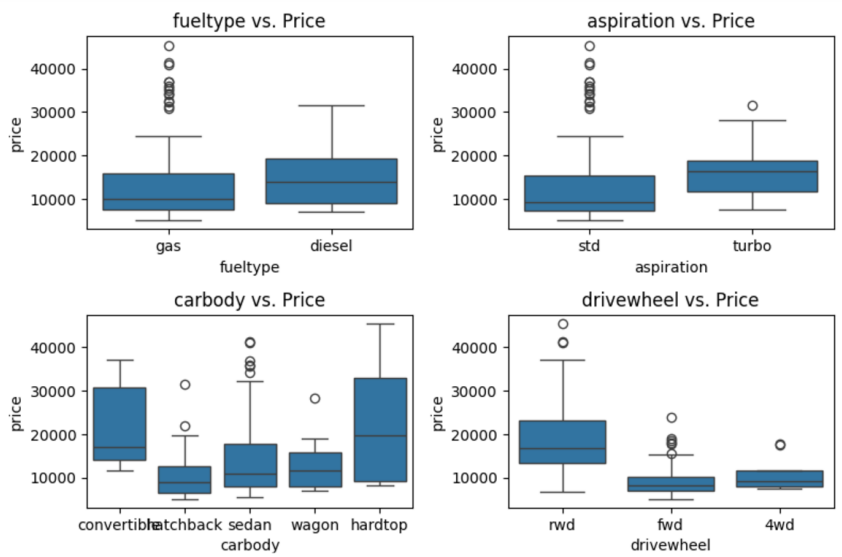
Output



Code

```
plt.figure(figsize=(12, 8))
for feature in categorical_columns:
    plt.subplot(3, 3, categorical_columns.index(feature) + 1)
    sns.boxplot(data=df, x=feature, y='price')
    plt.title(f'{feature} vs. Price')
plt.tight_layout()
plt.show()
```

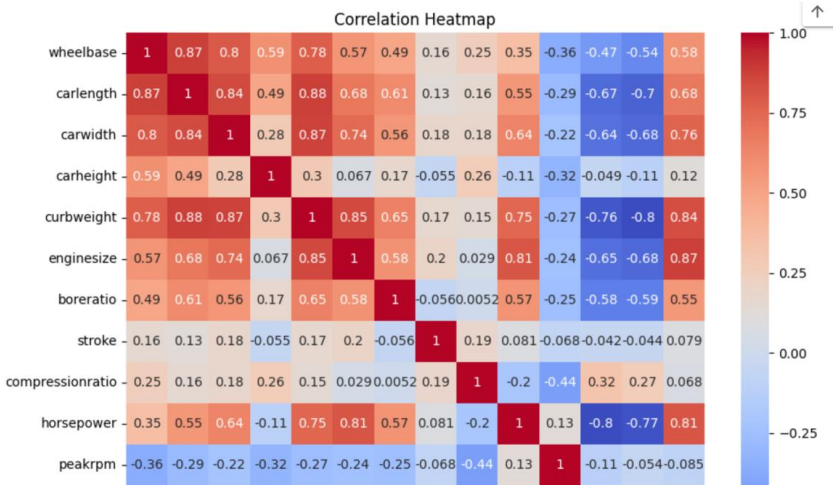
Output



Code

```
correlation_matrix = df[numerical_features].corr()
plt.figure(figsize=(10, 8))
sns.heatmap(correlation_matrix, annot=True, cmap='coolwarm')
plt.title('Correlation Heatmap')
plt.show()
```

Output



Code

```
df['brand'] = df['CarName'].apply(lambda x: x.split(' ')[0])
df['model'] = df['CarName'].apply(lambda x: ' '.join(x.split(' ')[1:]))
categorical_columns = ['fueltype', 'aspiration', 'doornumber', 'carbody',
'drivewheel',
                       'enginelocation', 'enginetype', 'cylindernumber', 'fuelsystem',
'brand', 'model']
numerical_columns = ['wheelbase', 'carlength', 'carwidth', 'carheight', 'curbweight',
                     'enginesize', 'boreratio', 'stroke', 'compressionratio', 'horsepower',
                     'peakrpm', 'citympg', 'highwaympg']
label_encoder = LabelEncoder()
for column in categorical_columns:
    df[column] = label_encoder.fit_transform(df[column])
df['power_to_weight_ratio'] = df['horsepower'] / df['curbweight']
for column in numerical_columns:
    df[f'{column}_squared'] = df[column] ** 2
df['log_enginesize'] = np.log(df['enginesize'] + 1)
scaler = StandardScaler()
df[numerical_columns] = scaler.fit_transform(df[numerical_columns])
```

Code

```
X = df.drop(['price', 'CarName'], axis=1)
y = df['price']
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2,
random_state=42)
model = LinearRegression()
model.fit(X_train, y_train)
y_pred = model.predict(X_test)
```

Code

```
mse = mean_squared_error(y_test, y_pred)

r2_square = r2_score(y_test,y_pred)

print(f" R-squared: {r2_square}")

print(f'Mean Squared Error: {mse}')
```

Output

```
 R-squared: 0.8615670883413198
Mean Squared Error: 10928450.668414652
```

Code

```
pred_df=pd.DataFrame({'Actual Value':y_test,'Predicted Value':y_pred,'Difference':y_test-y_pred})

pred_df
```

Output

	Actual Value	Predicted Value	Difference
15	30760.000	24644.498946	6115.501054
9	17859.167	19770.234284	-1911.067284
100	9549.000	8323.732428	1225.267572
132	11850.000	13885.801565	-2035.801565
68	28248.000	26718.615839	1529.384161
95	7799.000	7079.308256	719.691744
159	7788.000	10106.850946	-2318.850946
162	9258.000	6570.571035	2687.428965
147	10198.000	10418.264330	-220.264330
182	7775.000	10904.998472	-3129.998472
191	13295.000	13284.208305	10.791695
164	8238.000	6407.502162	1830.497838
65	18280.000	17635.683226	644.316774
175	9988.000	10737.804545	-749.804545
73	40960.000	43281.426089	-2321.426089

GitHub Repository: GitHub