About Linear-Regression Algorithm in ML Model

Linear Regression is a machine learning model that is used to analyze the relationship between a dependent variable and one or more independent variables. In the context of electric vehicles (EVs), Linear Regression can be used to model and predict various aspects of EV performance, energy consumption, and range estimation.

Linear Regression model for an EV project, the dependent variable (also called the target variable) is typically a numerical value that you want to predict, such as energy consumption, driving range, or battery performance. The independent variables (also called features or predictors) are the factors that influence the dependent variable, such as battery capacity, charging time, driving conditions, or temperature.

The goal of Linear Regression is to find the best-fitting linear relationship between the independent variables and the dependent variable.

Once the model is trained and the coefficients are determined, it can be used to make predictions on new, unseen data. Given the values of the independent variables for a specific EV instance, the model can calculate the predicted value of the dependent variable (e.g., predict energy consumption or driving range).

It's important to note that Linear Regression assumes a linear relationship between the independent variables and the dependent variable. If the relationship is non-linear, Linear Regression may not provide accurate predictions. In such cases, more advanced regression techniques like polynomial regression, support vector regression, or decision tree-based regression models may be more appropriate.

Dataset View

```
In [7]: data = pd.read_csv("car data.csv")
```

```
In [8]: data.head()
```

Out[8]:

	Car_Name	Year	Selling_Price	Present_Price	Kms_Driven	Fuel_Type	Seller_Type	Transmission	Owner
0	ritz	2014	3.35	5.59	27000	Petrol	Dealer	Manual	0
1	sx4	2013	4.75	9.54	43000	Diesel	Dealer	Manual	0
2	ciaz	2017	7.25	9.85	6900	Petrol	Dealer	Manual	0
3	wagon r	2011	2.85	4.15	5200	Petrol	Dealer	Manual	0
4	swift	2014	4.60	6.87	42450	Diesel	Dealer	Manual	0

```
In [9]: data.tail()
```

Out[9]:

	Car_Name	Year	Selling_Price	Present_Price	Kms_Driven	Fuel_Type	Seller_Type	Transmission	Owner
296	city	2016	9.50	11.6	33988	Diesel	Dealer	Manual	0
297	brio	2015	4.00	5.9	60000	Petrol	Dealer	Manual	0
298	city	2009	3.35	11.0	87934	Petrol	Dealer	Manual	0
299	city	2017	11.50	12.5	9000	Diesel	Dealer	Manual	0
300	brio	2016	5.30	5.9	5464	Petrol	Dealer	Manual	0

```
In [10]: data.shape
Out[10]: (301, 9)
```

Final Conclusion & Insights

Final conclusion and insights gained from research or analysis work using Linear Regression in the context of electric vehicles (EVs) can vary depending on the specific objectives and variables considered. Here are some general conclusions and insights that could be obtained:

Linear Regression can provide insights into the relationships between independent variables (features) and the dependent variable (target) in an EV project. By analyzing the coefficients of the regression model, you can determine the magnitude and direction of the impact each independent variable has on the dependent variable. For example, it can reveal whether battery capacity, charging time, or other factors significantly affect energy consumption or driving range.

The evaluation of the Linear Regression model's performance, you can assess how well it fits the data and makes accurate predictions. Evaluation metrics such as mean squared error (MSE), root mean squared error (RMSE), or R-squared can provide insights into the model's predictive accuracy and goodness of fit. This evaluation helps determine the model's reliability and guides further improvements or adjustments if necessary.

The final conclusion and insights from research and analysis work using Linear Regression in an EVs project depend on the specific objectives, variables considered, data quality and the depth of analysis performed. The obtained insights can inform decision-making, optimization efforts, and future research in the field of electric vehicles.

The Market Segmentation Project given additional time & some budget to purchase data and improve it:

The Market Segmentation Project using Linear Regression in the context of electric vehicles (EVs), here are some steps that can be taken with additional time and a budget to purchase data:

Data collections: Allocate a budget to purchase additional data relevant to the EV market and consumer preferences. This could include data on EV sales, customer demographics, charging infrastructure, government incentives, and other variables that may influence market segmentation.

Data pre-processing: Once the new data is acquired, pre-process it by cleaning, integrating, and transforming it into a suitable format for analysis. Handle missing values, outliers, and inconsistencies in the data. Perform feature engineering techniques to extract meaningful information from the acquired data.

Visualization and reporting: Utilize data visualization techniques to present the findings in a clear and compelling manner. Create visually appealing charts, graphs, and dashboards that communicate the market segmentation insights effectively. Prepare a comprehensive report summarizing the analysis, methodology, and key findings to share with stakeholders.

Model refinement: With additional time, refine the Linear Regression model by experimenting with different variations and adjustments. Explore different forms of regularization techniques (such as Ridge or Lasso regression) to mitigate overfitting and improve generalization. Adjust hyperparameters, including learning rate, regularization strength, or feature transformation methods, to enhance the model's performance.

Above these steps and leveraging additional time and budget to purchase data, you can enhance the Market Segmentation Project using Linear Regression in the EV domain. This will lead to more accurate and insightful market segmentation analysis, enabling business to make informed decisions and tailor their strategies to specific customer segments in the EV market.

The Estimated Market Size

As an AI language model, I don't have real-time data or the capability to perform calculations. I can provide you with general information.

To estimate the market size for the electric vehicle (EV) market, linear regression alone might not be the most suitable method. Market size estimation typically involves various factors, such as historical data, growth rates, industry trends, consumer behaviour, and more.

To estimate the market size of EVs, you would need to consider factors like the number of EVs sold over a certain period, growth rates in the EV market, projections for future adoption, and any specific data or research available for the region or market segment you are interested in.

You could analyze historical data on EV sales and growth rates, and then use regression analysis or other statistical techniques to project future market size. Additionally, you might need to consider factors like government policies, infrastructure development, technological advancements, and consumer preferences.

It's important to note that market size estimation can be complex and requires a comprehensive analysis of multiple variables. Consulting industry reports, market research firms, or specialized analysts would be advisable to obtain more accurate and up-to-date information for your specific market domain and geographical area of interest.

Name top 4 Variables/features which can be used to create most optimal Market Segments for your Market Domain

To create optimal market segments for the electric vehicle (EV) market using linear regression, you would typically consider variables or features that have a significant impact on the demand for EVs. While the selection of variables depends on the available data and the specific context, here are four commonly used variables/features that can help create market segments for the EV domain:

Price: The price of an EV is a crucial factor influencing consumer behaviour and market demand. By analyzing the relationship between price and demand using linear regression, you can identify price ranges that attract different market segments.

Range: The range or driving distance of an EV on a single charge is another vital variable. Many consumers have range anxiety, so understanding the relationship between range and demand can help identify segments that prioritize longer ranges.

Charging Infrastructure: The availability and accessibility of charging infrastructure can significantly impact the adoption of EVs. Including variables related to the density of charging stations or the ease of finding charging points can help identify segments that are more likely to adopt EVs based on infrastructure availability.

Government Incentives: Government policies and incentives, such as tax credits, subsidies, and rebates, can greatly influence the demand for EVs. Including variables related to the presence and extent of government incentives in different regions can help identify market segments that are more responsive to such policy interventions.

These variables are just a starting point, and depending on your specific research or data availability, you may need to consider additional variables, such as vehicle specifications, consumer demographics, environmental consciousness, and brand preferences, to create more accurate and optimal market segments.

ML Model using linear regression algorithm

Analysis:

```
In [10]: data.shape

Out[10]: (301, 9)

In [11]: data.dtypes

Out[11]: Car_Name          object
         Year               int64
         Selling_Price    float64
         Present_Price    float64
         Kms_Driven         int64
         Fuel_Type         object
         Seller_Type       object
         Transmission      object
         Owner              int64
         dtype: object

In [12]: data.columns

Out[12]: Index(['Car_Name', 'Year', 'Selling_Price', 'Present_Price', 'Kms_Driven',
                'Fuel_Type', 'Seller_Type', 'Transmission', 'Owner'],
               dtype='object')

In [13]: data.info()

         <class 'pandas.core.frame.DataFrame'>
         RangeIndex: 301 entries, 0 to 300
         Data columns (total 9 columns):
          #   Column         Non-Null Count  Dtype
         ---  ------         --------------  -----
```

Checking Null Values:

```
In [14]: data.isnull().sum()

Out[14]: Car_Name         0
         Year             0
         Selling_Price    0
         Present_Price    0
         Kms_Driven       0
         Fuel_Type        0
         Seller_Type      0
         Transmission     0
         Owner            0
         dtype: int64
```

```
In [15]: data.describe()
```

Out[15]:

	Year	Selling_Price	Present_Price	Kms_Driven	Owner
count	301.000000	301.000000	301.000000	301.000000	301.000000
mean	2013.627907	4.661296	7.628472	36947.205980	0.043189
std	2.891554	5.082812	8.644115	38886.883882	0.247915
min	2003.000000	0.100000	0.320000	500.000000	0.000000
25%	2012.000000	0.900000	1.200000	15000.000000	0.000000
50%	2014.000000	3.600000	6.400000	32000.000000	0.000000
75%	2016.000000	6.000000	9.900000	48767.000000	0.000000
max	2018.000000	35.000000	92.600000	500000.000000	3.000000

Create Dummy Variables:

```
In [28]: data.replace({"Fuel_Type":{"Petrol":0, "Diesel":1, "CNG":2}}, inplace = True)
         dummy_data = pd.get_dummies(data, columns = ["Seller_Type", "Transmission"], drop_first = True)

In [29]: dummy_data.head()
```

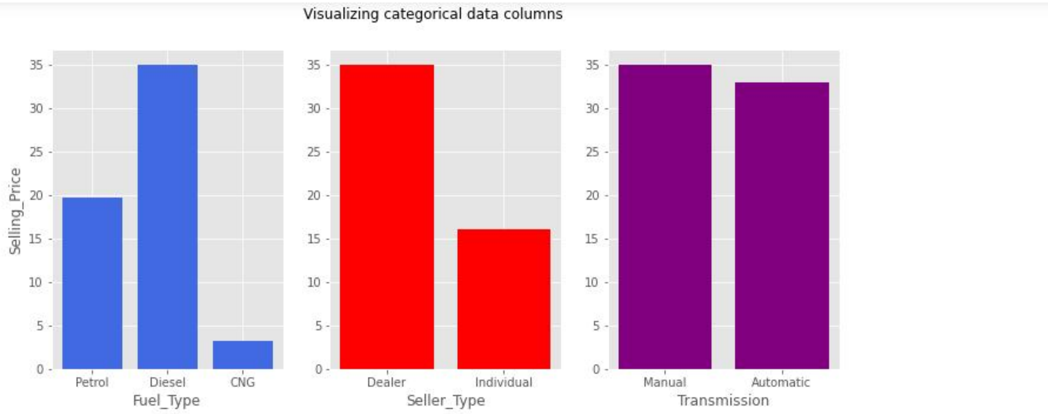
Out[29]:

	Car_Name	Year	Selling_Price	Present_Price	Kms_Driven	Fuel_Type	Owner	Seller_Type_Individual	Transmission_Manual
0	ritz	2014	3.35	5.59	27000	0	0	0	1
1	sx4	2013	4.75	9.54	43000	1	0	0	1
2	ciaz	2017	7.25	9.85	6900	0	0	0	1
3	wagon r	2011	2.85	4.15	5200	0	0	0	1
4	swift	2014	4.60	6.87	42450	1	0	0	1

Visualizations:

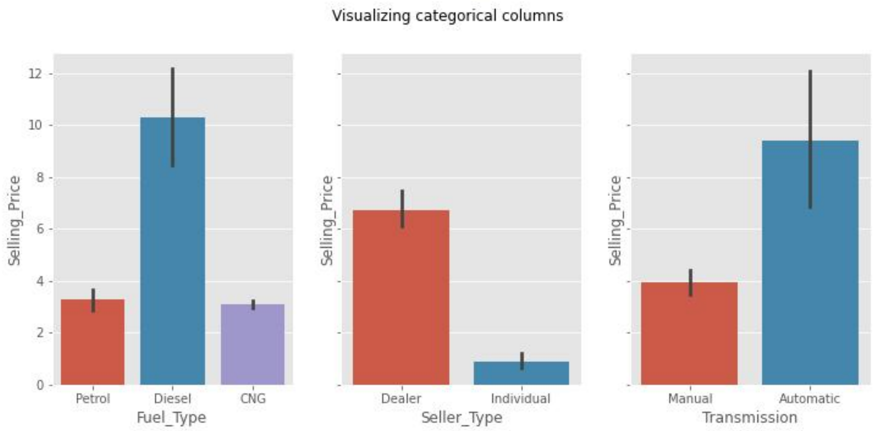
```
In [22]: from matplotlib import style
```

```
In [25]: style.use("ggplot")
fig = plt.figure(figsize = (12,5))
fig.suptitle("Visualizing categorical data columns")
plt.subplot(1,3,1)
plt.bar(fuel_type, selling_price, color = "royalblue")
plt.xlabel("Fuel_Type")
plt.ylabel("Selling_Price")
plt.subplot(1,3,2)
plt.bar(seller_type, selling_price, color = "red")
plt.xlabel("Seller_Type")
plt.subplot(1,3,3)
plt.bar(transmission, selling_price, color = "purple")
plt.xlabel("Transmission")
plt.show()
```

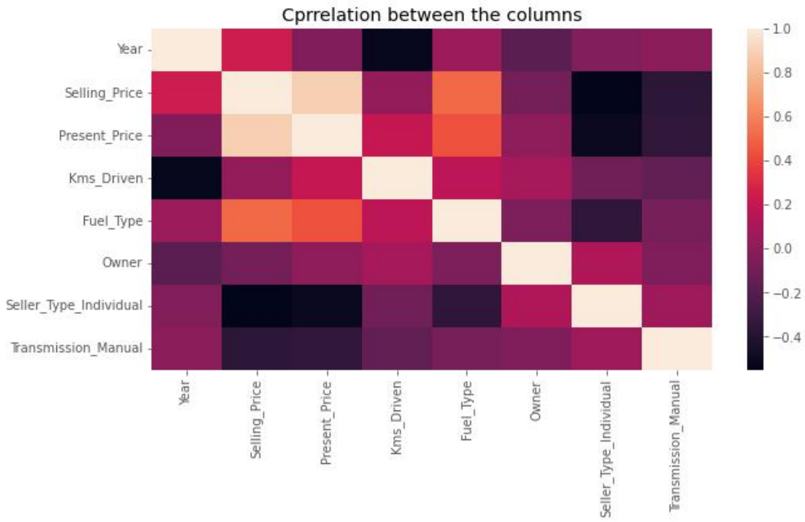



```
In [26]: fig, axes = plt.subplots(1,3, figsize = (12,5), sharey = True)
fig.suptitle("Visualizing categorical columns")
sns.barplot(x = fuel_type, y = selling_price, ax = axes[0])
sns.barplot(x = seller_type, y = selling_price, ax = axes[1])
sns.barplot(x = transmission, y = selling_price, ax = axes[2])
```

```
Out[26]: <AxesSubplot:xlabel='Transmission', ylabel='Selling_Price'>
```

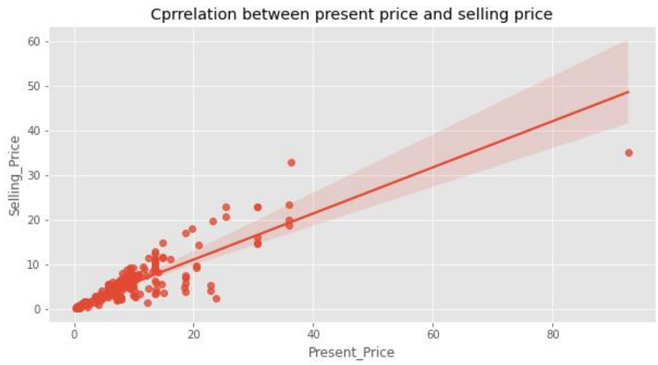


In [31]:
```
plt.figure(figsize = (10,5))
sns.heatmap(dummy_data.corr(), annot = False)
plt.title("Cprrelation between the columns")
plt.show()
```



In [32]:
```
plt.figure(figsize = (10,5))
plt.title("Cprrelation between present price and selling price")
sns.regplot(x ='Present_Price', y = 'Selling_Price', data = dummy_data )
```

Out[32]:
```
<AxesSubplot:title={'center':'Cprrelation between present price and selling price'}, xlabel='Present_Price', ylabel='Selling_Pr
ice'>
```



Train Test Split:

```
In [33]: x = data.drop(["Car_Name", "Selling_Price"], axis = 1)
         y = data["Selling_Price"]

In [34]: print("Shape of x is: ",x.shape)
         print("Shape of y is: ",y.shape)

         Shape of x is:  (301, 7)
         Shape of y is:  (301,)

In [35]: x_train, x_test, y_train, y_test = train_test_split(x, y, test_size=0.2, random_state=42)

In [36]: print("x_test shape:", x_test.shape)
         print("x_train shape:", x_train.shape)
         print("y_test shape:", y_test.shape)
         print("y_train shape:", y_train.shape)

         x_test shape: (61, 7)
         x_train shape: (240, 7)
         y_test shape: (61,)
         y_train shape: (240,)

In [37]: scaler = StandardScaler()

In [38]: x_train = scaler.fit_transform(x_train)
         x_test = scaler.transform(x_test)
```

ML Model:

```
In [31]: model = LinearRegression()

In [32]: model.fit(X_train, y_train)

 Out[32]: LinearRegression()

In [33]: pred = model.predict(X_test)

In [34]: from sklearn.metrics import mean_absolute_error, mean_squared_error, r2_score

In [35]: print("MAE: ", (metrics.mean_absolute_error(pred, y_test)))
         print("MSE: ", (metrics.mean_squared_error(pred, y_test)))
         print("R2 score: ", (metrics.r2_score(pred, y_test)))

         MAE:  1.2581404706473371
         MSE:  3.4932860262251473
         R2 score:  0.8294933369778816
```