```
1 import pandas as pd
2 df = pd.read_csv("/content/cars.csv")
3 df.head()
```

	Car_ID	Brand	Model	Year	Kilometers_Driven	Fuel_Type	Transmission	Owner_
0	1	Toyota	Corolla	2018	50000	Petrol	Manual	
1	2	Honda	Civic	2019	40000	Petrol	Automatic	Se
2	3	Ford	Mustang	2017	20000	Petrol	Automatic	
3	4	Maruti	Swift	2020	30000	Diesel	Manual	
4	5	Hyundai	Sonata	2016	60000	Diesel	Automatic	Se

Next steps: **Generate code with df**   **View recommended plots**

```
1 import pandas as pd
2 from sklearn.ensemble import GradientBoostingRegressor
3 from sklearn.preprocessing import LabelEncoder
4 from sklearn.metrics import mean_squared_error, mean_absolute_error, r2_score
5 import matplotlib.pyplot as plt
```

```
1 # Separate the independent and dependent variables
2 X = df[['Brand', 'Model', 'Kilometers_Driven', 'Fuel_Type', 'Transmission', 'Owner_Type', 'Mileage', 'Engine', 'Power', 'Seats']]
3 y = df['Price']
```

```
1    # Create label encoders for the nominal variables
2    label_encoders = {
3        'Brand': LabelEncoder(),
4        'Model': LabelEncoder(),
5        'Fuel_Type': LabelEncoder(),
6        'Transmission': LabelEncoder(),
7        'Owner_Type': LabelEncoder()
8    }
```

```
1    # Encode the nominal variables
2    for variable in label_encoders.keys():
3        X[variable] = label_encoders[variable].fit_transform(X[variable])
```

Show hidden output

```
1    # Create a GradientBoostingRegressor object
2    regressor = GradientBoostingRegressor(random_state=0)
```

```
1    # Train the regressor
2    regressor.fit(X, y)
```

```
▾        GradientBoostingRegressor
GradientBoostingRegressor(random_state=0)
```

```
1    # Make predictions on the training data
2    y_pred = regressor.predict(X)
```

```
1 # Calculate model fit indices
2 mse = mean_squared_error(y, y_pred)
3 rmse = mean_squared_error(y, y_pred, squared=False)
4 mae = mean_absolute_error(y, y_pred)
5 r2 = r2_score(y, y_pred)
```

```
1    # Print model fit indices
2    print("Mean Squared Error:", mse)
3    print("Root Mean Squared Error:", rmse)
4    print("Mean Absolute Error:", mae)
5    print("R-squared:", r2)
```

```
Mean Squared Error: 498791785.9667061
Root Mean Squared Error: 22333.646947301422
Mean Absolute Error: 16261.170457291008
R-squared: 0.9994964364457937
```