Data Sources:
The data used for this project was sourced from a Kaggle dataset. The specific dataset used can be found at kaggle .Dataset was chosen as per the requirement of assignment.

Feature selection:
To ensure the data was suitable for the model, various feature engineering techniques were employed. This involved addressing missing values, handling outliers, performing encoding, variable transformations, and scaling. In particular, feature selection was conducted using a combination of methods such as the Variance Inflation Factor (VIF), correlation analysis, and leveraging domain knowledge. Features exhibiting high multicollinearity were identified and only those independent variables were selected.

Model Architecture:
For this project, the primary model architecture employed was linear regression. This choice yielded high accuracy results. However, to further enhance the model, alternative architectures such as Lasso and Ridge regression can be explored. Ridge regression, for instance, can effectively address multicollinearity and mitigate the impact of irrelevant features on the model's performance.

Evaluation Matrices:
To assess the performance of the model, various evaluation metrics were calculated, including mean squared error (MSE), mean absolute error (MAE), and R-squared. These metrics provide insights into the accuracy, precision, and goodness of fit of the model. Additionally, different graphs like scatter,regplot were plotted to visualize the relationship between the predicted and actual values, enabling a comprehensive understanding of the model's predictive capabilities.

Deployed Model Instruction

This documentation provides instructions on how to use the deployed model as a web service or API. The model has been deployed using FastAPI framework and can be accessed via the local server at http://127.0.0.1:8000/. The API allows you to post input values and receive the corresponding model predictions.

Endpoint:
The API has a single endpoint available for making predictions.

Endpoint URL: http://127.0.0.1:8000/docs

Request Format:
The model expects a JSON payload containing the input data. The required input parameters and their formats are as follows:

```
{
  "sales": float,
  "horsepower": float,
  "width": float,
  "fuel_efficiency": float,
  "manufacturer": "string",
  "vehicle_type": "string"
}
```

p.s
All the code files are attached in mail.