Project Report: Car Price Prediction

Introduction

This project is part of COMP 5212, focusing on predicting car prices for various brands using machine learning techniques. The project involves data preprocessing, model training, and evaluation using different machine learning algorithms.

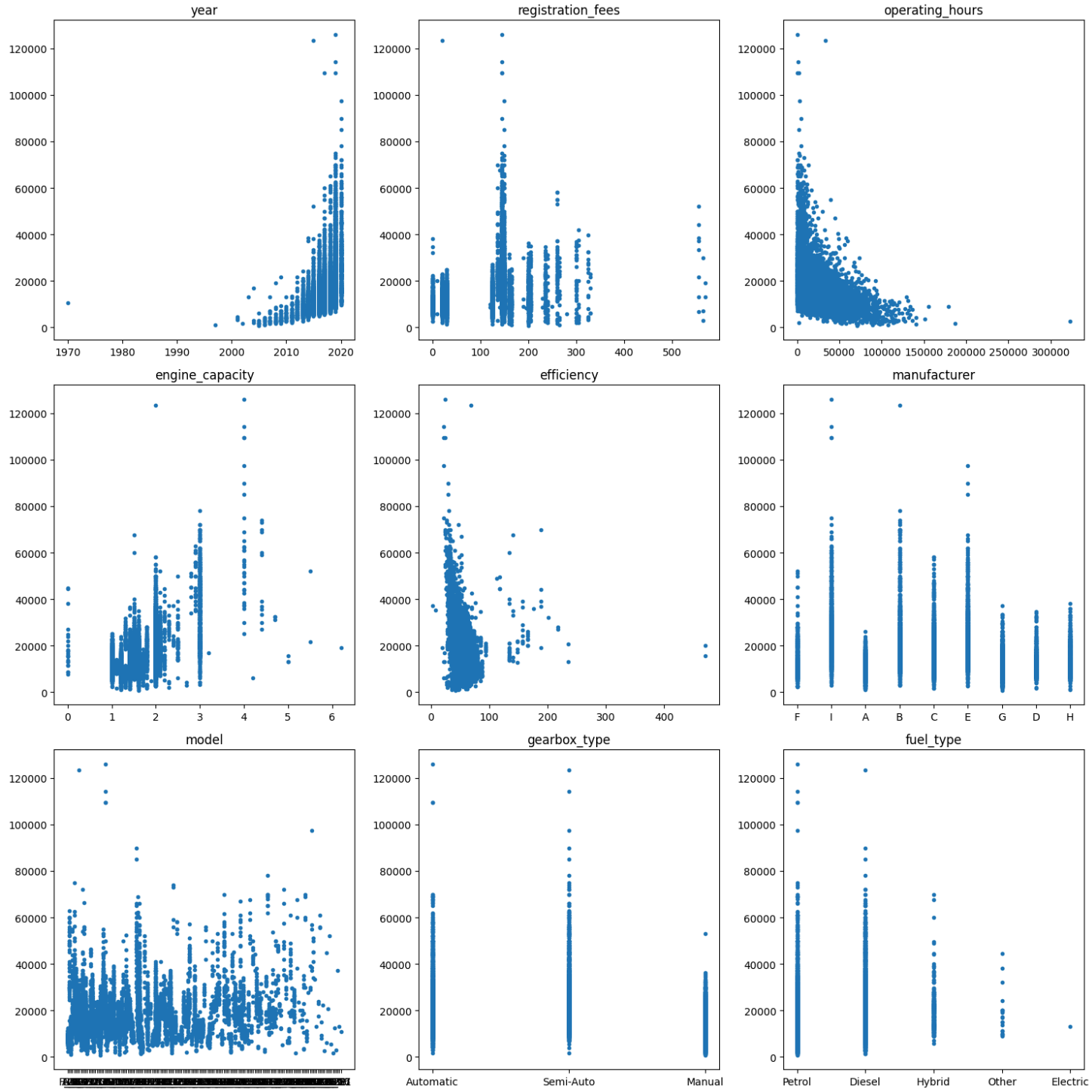
Project Structure

The workspace is organized as follows:

```
data/
    test.csv
    train.csv
data_analy.ipynb
encoder.pkl
inference.py
isoforest_model.pkl
mlp_model.pth
mlp.ipynb
model.pkl
output.csv
preprocessing.py
run.sh
scaler.pkl
```

Data analysis

Data analysis is performed in data_analy.ipynb, our outcome is the scatter graph of the price of the car and the various attributes of the car.



We can see tha the price of the car is correlated to the attributes, but it is more complicated than a linear relationship.

Also, the manufacturer of the car is a redundant attribute, as it is included in the model of the car.(e.g. model A_1 is from brand A)

Moreover, we can classify the attributes into two categories, discrete and continuous attributes. Labeled data fuel_type, gearbox_type, model, and manufacturer are discrete attributes, others like engine_capacity, registration_fees, years seems also discrete. While the operating_hours and efficiency are continuous attributes. More information will be revealed in the data preprocessing part.

Data Preprocessing

Data preprocessing is handled in preprocessing.py. Key steps include:

Model Training

Linear Regression

The linear regression model is implemented in linear_regression.ipynb. Key steps include:

- *Preparing Data*: The data is split into training and validation sets.
- *Training the Model*: The model is trained using the training data.
- *Evaluating the Model*: The model is evaluated using Mean Squared Error (MSE).
- *Saving Predictions*: The predictions are saved to `output.csv`.

Random Forest Regression

The random forest regression model is implemented in legacy/random_forest_regression.ipynb. Key steps include:

- *Preparing Data*: The data is split into training and validation sets.
- *Training the Model*: The model is trained using `RandomForestRegressor` from `sklearn`.
- *Hyperparameter Tuning*: Hyperparameter tuning is performed using `GridSearchCV`.
- *Evaluating the Model*: The model is evaluated using cross-validation.
- *Saving Predictions*: The predictions are saved to `output.csv`.

Multi-Layer Perceptron (MLP)

The MLP model is implemented in mlp.ipynb. Key steps include:

- *Defining the Model*: The MLP architecture is defined using `torch.nn`.
- *Training the Model*: The model is trained using the training data.
- *Evaluating the Model*: The model is evaluated using training and validation loss.
- *Saving the Model*: The best model is saved to `mlp_model.pth`.

Attempts to Improve the Model

Discrete features

Outliers

Label encoding

One hot encoding

Standardization

Hyperparameter tuning

Holdout validation

Model Evaluation

Model evaluation is performed using various metrics:

- *Linear Regression*: Evaluated using Mean Squared Error (MSE).
- *Random Forest Regression*: Evaluated using cross-validation scores.
- *MLP*: Evaluated using training and validation loss. The results are visualized using plots in mlp.ipynb.

Inference

The inference script inference.py is used to make predictions on new data using the trained models. The predictions are saved to `output.csv` .

Conclusion

This project demonstrates the application of different machine learning algorithms for car price prediction. The models are trained and evaluated using various techniques, and the results are saved for further analysis.

For more details, refer to the individual notebooks and scripts in the workspace.