Project: Predicting Fuel Efficiency of Indian Vehicles Using Deep Learning

Project Overview

Your challenge is to build a robust deep learning model that predicts the fuel efficiency (e.g., km/litre or mileage) of vehicles in the Indian context. This end-to-end project will take you through the entire data science and machine learning workflow, culminating in a deployed application. The project is designed for a few weeks, with each phase building on the previous one and requiring thoughtful, original work.

Project Roadmap

1. Data Gathering

Goal: Create or collect a dataset that accurately reflects the Indian automotive market.
Tasks:

- Research the types of vehicles, fuel types, and driving conditions common in India.
- If no suitable dataset exists, design and generate a semi-synthetic or fully synthetic dataset. This should reflect Indian vehicle models, fuel types (petrol, diesel, CNG, electric), engine capacities, transmission types, typical road conditions, and climate factors.
- Document your data generation process, including all assumptions, sources, and rationale for chosen features.
- Justify how your dataset addresses the unique aspects of the **Indian context**.

2. Data Cleaning

Goal: Ensure your dataset is accurate, consistent, and ready for analysis.
Tasks:

- Identify and handle missing values, outliers, and inconsistencies.
- Standardize categorical variables (e.g., manufacturer names, fuel types).
- Validate that all features have plausible ranges and values.
- Summarize your cleaning process with before-and-after statistics and visualizations.

3. Data Analysis

Goal: Understand the structure, trends, and relationships in your data.
Tasks:

- Perform exploratory data analysis (EDA) using summary statistics and visualizations.
- Investigate how features like engine size, vehicle weight, and fuel type correlate with fuel efficiency.
- Identify any surprising trends or anomalies, and hypothesize possible explanations.
- Document your findings with plots and concise interpretations.

4. Feature Engineering

Goal: Enhance your dataset with new, meaningful features.
Tasks:

- Derive additional features (e.g., power-to-weight ratio, city vs. highway mileage, average annual kilometers driven).
- Justify each new feature with domain knowledge or supporting analysis.
- Evaluate the impact of your engineered features on model performance in later steps.

5. Feature Scaling

Goal: Prepare your features for deep learning models.
Tasks:

- Decide which features require scaling (e.g., normalization, standardization).
- Apply appropriate scaling techniques and explain your choices.
- Compare model training results before and after scaling to highlight its importance.

6. Model Benchmarking

Goal: Identify the best model for predicting fuel efficiency.
Tasks:

- Implement at least three different models: a baseline regression model (e.g., linear regression), a classical machine learning model (e.g., random forest), and a deep learning model (e.g., a neural network).
- Train and evaluate each model using appropriate metrics (e.g., MAE, RMSE, R^2).

- Visualize and compare the performance of each model using Python plots.
- Write a comparative analysis discussing which model works best and why.

7. Final Accuracy Assessment

Goal: Validate the performance of your chosen model on unseen data.
Tasks:

- Use a hold-out test set or cross-validation to assess final model accuracy.
- Interpret results in the context of real-world application (e.g., what does a certain RMSE mean for a car buyer?).
- Discuss any trade-offs or limitations observed.

8. Overfitting & Underfitting Analysis

Goal: Ensure your model generalizes well.
Tasks:

- Plot training vs. validation loss curves.
- Experiment with model complexity, regularization, and dropout.
- Document steps taken to address overfitting or underfitting, and their effects on performance.

9. App Development

Goal: Make your model accessible through a user-friendly application.
Tasks:

- Design and build a web app (e.g., using Streamlit) where users can input vehicle parameters and receive a fuel efficiency prediction.
- Ensure the app is intuitive, visually appealing, and robust.
- Test the app with a variety of inputs and document the testing process.

10. Version Control & Github

Goal: Maintain a clean, reproducible, and collaborative workflow.
Tasks:

- Organize your code, data, and documentation in a clear directory structure.
- Use meaningful commit messages and maintain a detailed project history.
- Push your complete project to a public or private Github repository.

11. Deployment

Goal: Make your app accessible to others.
Tasks:

- Deploy your app on a cloud platform (e.g., Streamlit Cloud, Heroku, or AWS).
- Ensure the deployed app is functional and accessible.
- Document the deployment process, including any challenges faced and how you resolved them.

12. Comprehensive Documentation

Goal: Provide a complete, professional record of your project.
Tasks:

- Write a detailed report covering all stages: problem statement, data generation, methodology, results, and conclusions.
- Include code snippets, visualizations, and explanations for key decisions.
- Prepare a user manual for your app and a README for your Github repository.
- Reflect on challenges faced, lessons learned, and potential future improvements.

Submission Guidelines

- Originality: All code, analysis, and documentation must be your own. Plagiarism or direct copy-pasting from online sources will result in disqualification.
- Documentation: Each section must include clear explanations, code comments, and visualizations where appropriate.
- Reflection: For each major step, include a brief reflection on challenges faced and how you overcame them.
- Presentation: Submit your Github repository link, deployed app URL, and final report as per the provided format.