CAR RECOMMENDATION SYSTEM

Submitted by:
Jitesh Maurya (102317244)
Kaavya Dhir (102317251)

BE Third Year

CSE/COE

Submitted to:

Dr. Anjula Mehto

Assistant Professor

THAPAR INSTITUTE
OF ENGINEERING & TECHNOLOGY
(Deemed to be University)

Computer Science and Engineering Department

Thapar Institute of Engineering and Technology, Patiala

November 2025

TABLE OF CONTENTS

Introduction or Project Overview

The automobile market in India has expanded rapidly in recent years, offering customers a wide variety of vehicle options across different price ranges, brands, and technical specifications. As a result, selecting an appropriate car that satisfies a user's budget, preferences, and performance expectations has have become complex.

To address this challenge, the present project proposes a **Machine Learning–based Car Recommendation System** that assists users in identifying suitable vehicles using data-driven analysis.

The system employs **K-Nearest Neighbors (KNN)**, a similarity-based learning algorithm, to recommend cars based on user inputs such as car name, budget, or fuel type. The model processes relevant automobile attributes—including price, mileage, engine capacity, transmission type, and brand—to identify vehicles closest in profile to a user's requirements.

A **Flask-based backend API** and an interactive **frontend interface** further make the system practical and user-friendly. The overall objective of this project is to demonstrate how ML techniques can improve consumer decision-making through personalized recommendations.

Problem Statement

The process of purchasing a car involves evaluating numerous factors such as vehicle price, fuel efficiency, engine specifications, brand popularity, fuel type, and the age of the vehicle. Due to the large volume of available options in the used and new car markets, manually comparing these attributes is often time-consuming and may lead to suboptimal decisions.

Therefore, there is a need for an automated recommendation system capable of:

- Analyzing multiple car features simultaneously
- Identifying vehicles similar to a user's preferred model
- Suggesting cars within a specified specification
- Highlighting popular or commonly purchased models

The main problem addressed by this project is the lack of an intelligent, data-driven tool that simplifies car selection using machine learning techniques.

Overview of the Dataset used

The system uses a car dataset (car_data.csv) that contains multiple attributes relevant for recommendation tasks. The dataset includes information such as:

- **Car Name**
- **Brand**
- **Fuel Type**
- **Transmission Type**
- **Selling Price**
- **Mileage**
- **Kilometers Driven**
- **Vehicle Age**
- **Engine CC**
- **Max Power**
- **Seating Capacity**
-

Before training the model, the dataset undergoes preprocessing steps, including:

- Handling missing values and duplicates
- Encoding categorical variables (Brand, Fuel Type, Transmission) using Label Encoding
- Normalizing numerical features using StandardScaler
- Organizing feature vectors for similarity computation

The processed dataset provides a reliable foundation for accurate recommendations.

Project Workflow

The workflow of the Car Recommendation System consists of the following major stages:

Step 1: Data Loading and Cleaning
The dataset is imported and cleaned using pandas. Missing values, inconsistencies, and duplicate rows are removed to ensure high-quality input for the model.

Step 2: Feature Engineering
Categorical features (brand, fuel type, transmission) are label-encoded, while numerical attributes (price, mileage, age, engine capacity, kilometers driven) are normalized. The model uses these transformed features to calculate car similarities.

Step 3: Model Training (K-Nearest Neighbors)
A KNN model is trained on the scaled feature matrix. The algorithm identifies the closest cars to a given vehicle or requirement by calculating Euclidean distances in high-dimensional feature space.
The number of neighbors (k=6) ensures diverse yet relevant recommendations.

Step 4: Model Saving and Deployment
The trained model, along with encoders and scaler, is stored in a serialized form (car_model.pkl) for efficient reuse.

Step 5: Flask API Development
A lightweight Flask backend (flask_app.py) exposes three major endpoints:

- /recommend/similar – recommends cars similar to a given model
- /recommend/budget – recommends best-value cars within a budget
- /recommend/popular – suggests popular cars based on dataset frequency

Step 6: Frontend Interface
An HTML-based user interface (index.html) provides:

- Tabs for Similar Cars, Budget-Based Search, and Popular Cars
- Interactive forms
- Visual cards displaying vehicle details
- Real-time results fetched via API calls

This frontend enhances usability and offers a complete end-to-end system.

Results

The system successfully generates relevant recommendations across multiple categories:

1. Similar Cars Recommendation
When the user inputs a car name (e.g., “Swift”), the system identifies vehicles with similar attributes such as price, fuel type, engine capacity, and mileage. The results include similarity scores indicating how closely each recommendation matches the chosen vehicle.

2. Custom Recommendations
For a given budget (e.g., ₹5,00,000), the system ranks cars based on a computed “value score,” which considers:

- Mileage
- Maximum price
- Fuel Type
- Seats
- Transmission
- Engine Power

This helps users find the most economical and performance-efficient vehicles within their budget.

3. Popular Cars
Using brand frequency counts, the system identifies and recommends the most common and widely preferred models in the dataset.
Across all three modes, the system produces accurate and meaningful results due to well-structured preprocessing and similarity learning.

Conclusion

This project demonstrates the practical application of machine learning in building an intelligent Car Recommendation System that assists users in making informed automobile purchasing decisions. By leveraging KNN-based similarity analysis, the system identifies cars that closely align with user preferences and budgets.

The integration of Flask APIs and a modern web interface ensures that the solution is accessible and user-friendly.

Future improvements may include:

- Incorporating advanced ML models (e.g., clustering, neural networks)
- Adding images and additional metadata
- Providing real-time price updates via APIs
- Building a mobile application interface

Overall, the project successfully meets its objective of showcasing how machine learning can be used to enhance user decision-making in the automotive domain.