ITM(SLS) BARODA UNIVERSITY
DEPARTMENT OF COMPUTER SCIENCE, ENGINEERING AND TECHNOLOGY
BTECH - CSE SEMESTER-5

MACHINE LEANING

Car Recommendation System

BY
Dhrumil Pawar [21C21111]
Vinit Todi [21C21159]

Abstract:

The automotive industry is continuously evolving, with a vast array of car models and features available to consumers. Choosing the right car that aligns with individual preferences and requirements can be a daunting task. In this context, machine learning-based car recommendation systems have emerged as a valuable tool to assist consumers in making informed decisions when selecting a vehicle.

This abstract provides an overview of a car recommendation system powered by machine learning algorithms. The system leverages a rich dataset comprising information on car specifications, user preferences, and historical user interactions to generate personalized car recommendations. The core components of the system include data preprocessing, feature engineering, and machine learning model development. Various algorithms, such

as collaborative filtering, content-based filtering, and hybrid approaches, are employed to enhance recommendation accuracy..

Recommendation System

Recommendation systems are becoming increasingly important in today's extremely busy world. People are always short on time with the myriad tasks they need to accomplish in the limited 24 hours. Therefore, the recommendation systems are important as they help them make the right choices, without having to expend their cognitive resources.

The purpose of a recommendation system basically is to search for content that would be interesting to an individual. Moreover, it involves a number of factors to create personalised lists of useful and interesting content specific to each user/individual. Recommendation systems are Artificial Intelligence based algorithms that skim through all possible options and create a customized list of items that are interesting and relevant to an individual. These results are based on their profile, search/browsing history, what other people with similar traits/demographics are watching, and how likely are you to watch those movies. This is achieved through predictive modeling and heuristics with the data available.

Use-cases of Recommendation systems

- Recommendations are not a new concept. Even when e-commerce was not that prominent, the sales staff in retail stores recommended items to the customers for the purpose of upselling and cross-selling, and ultimately maximise profit. The aim of recommendation systems is just the same.

- Another objective of the recommendation system is to achieve customer loyalty by providing relevant content and maximising the time spent by a user on your website or channel. This also helps in increasing customer engagement.

- On the other hand, ad budgets can be optimized by showcasing products and services only to those who have a propensity to respond to them. Check out the recommendation system python course and enhance your knowledge of other use cases.

Real-World Examples

Here are some of the examples of the pioneers in creating algorithms for recommendation systems and using them to serve their customers better in a personalized manner. These are:

GroupLens:

– Helped in developing initial recommender systems by pioneering collaborative filtering model
– It also provided many data-sets to train models including MovieLens and BookLens

Amazon:

– Implemented commercial recommender systems
– They also implemented a lot of computational improvements

Netflix Prize:

– Pioneered Latent Factor/ Matrix Factorization models

Google-Youtube:

– Hybrid Recommendation Systems
– Deep Learning based systems
– Social Network Recommendations

Filtration Strategies for Car Recommendation Systems

Filtration strategies in car recommendation systems are essential for refining and personalizing the vehicle suggestions provided to users. Collaborative filtering relies on user behavior and preferences, while content-based filtering considers car attributes. Hybrid approaches combine both methods for comprehensive recommendations. Matrix factorization optimizes recommendations by uncovering latent user-vehicle interactions. Deep learning models, such as neural collaborative filtering, enhance accuracy by analyzing complex patterns. These filtration strategies enable

Content-Based Filtering

The process begins with the creation of a user profile, which includes data on the user's preferences and past interactions with cars. These interactions may include ratings, searches, or previous purchases. Additionally, the system collects detailed information about the cars in the dataset, such as make, model, year, price, fuel efficiency, engine size, safety features, and more.

The key steps in content-based filtering for car recommendations include:

Feature Extraction: The system analyzes the user's historical data to identify patterns and preferences. It extracts relevant features from both the user profile and the car dataset.

Profile-Item Matching: A similarity metric is applied to compare the features of cars in the dataset with the user's profile. This metric measures how closely the attributes of a car match the user's preferences.

Ranking and Recommendation: Cars are ranked based on their similarity scores, and the top-ranked vehicles are recommended to the user. The system may also apply filters to ensure that only relevant and available cars are suggested.

Feedback Loop: User feedback, such as ratings and interactions with recommended cars, is continuously collected and used to update the user profile and improve future recommendations.

Dataset Used

We used Dataset from the Kaggle site **Car Dataset**

The dataset contains

- **Car1.csv:** Contains the general information of car and performance indicators (car_name , reviews,fuel type , engine_displacement, seating_capacity, transmission_type, fuel_tank_capacity, body_type, rating, starting_price, ending_price, max_torque_nm, max_torque_rpm, max_power_bhp, max_power_rp.)

Some of the fields (not all) in the dataset are as follows:

Data Sr-No– Integer unique id of each car

- **Car Name**=its contains the car names
- **Reviews=** how much the car have own reviews
- **Fual type=** its shown the car is petrol or diesel
- **Engine displacement =** Engine replacement is typically performed when a vehicle's existing engine becomes irreparably damaged.

- **no_cylinder=** how many cylinder are there.
- **seating_capacity=** its shown that how many member can sit on time in car.
- **Fual_tank_capacity =**how much fual can store one time in car.
- **Body_type=** it show car is SUV or hatchback.
- **Rating =** car rating is getting by customers.
- **Starting price =** price of the car ,
- **Ending_price =** top model price.

For our recommender system, we'll use both of the techniques mentioned above: content-based and collaborative filtering. To find the similarity between movies for our content-based method, we'll use a cosine similarity function. For our collaborative filtering method, we'll use a matrix factorization technique.

Pseudo Code for car recommendation system:

MACHINE LEANING

```
# Import the pandas library
import pandas as pd

# Read a CSV file named "CARS_1.csv" into a DataFrame called car_dataset
car_dataset = pd.read_csv(r"CARS_1.csv")

# Convert specific columns to string data type
# (fuel_tank_capacity, rating, transmission_type, reviews_count, starting_price)

# Check for missing values in the DataFrame
missing_values = car_dataset.isnull().sum()

# Check for duplicated rows in the DataFrame
duplicated_rows = car_dataset.duplicated().sum()

# Define a function for a car recommendation system based on user preferences
function rec_rating():
   # Prompt the user for input
   a = input("You will get a car based on how many ratings you see?\n")
   # Filter the DataFrame based on the user's rating input
   new_df = car_dataset[car_dataset['rating'] == a]

   # Prompt the user for fuel type preference
   b = input("Which fuel type do you want?\n Petrol\n Diesel\n CNG\n Electric\n")
   # Filter the DataFrame further based on the user's fuel type input
   new_df_Fuel = new_df[new_df['fuel_type'] == b]

   # Prompt the user for starting price preference
   c = input("What is your starting price?\n")
   # Filter the DataFrame further based on the user's starting price input
   new_df_price = new_df_Fuel[new_df_Fuel['starting_price'] >= c]

   # Prompt the user for transmission type preference
   d = input("What kind of transmission type do you want?\n Such as\n Manual\n Electric\n
Automatic\n")

   # Display the recommended cars based on all user preferences
   print(new_df_price[new_df_price['transmission_type'] == d])

# Print a welcome message
print("\n\n-------------------------------------<--WELCOME TO CAR
RECOMMENDATION SYSTEM ------>-----------------------\n")
```

```
# Call the car recommendation function
rec_rating()

# Print a thank you message
print("\n\n-----------------------------------<----THANK YOU---\U0001f600----
----------->-----------------------\n")
```

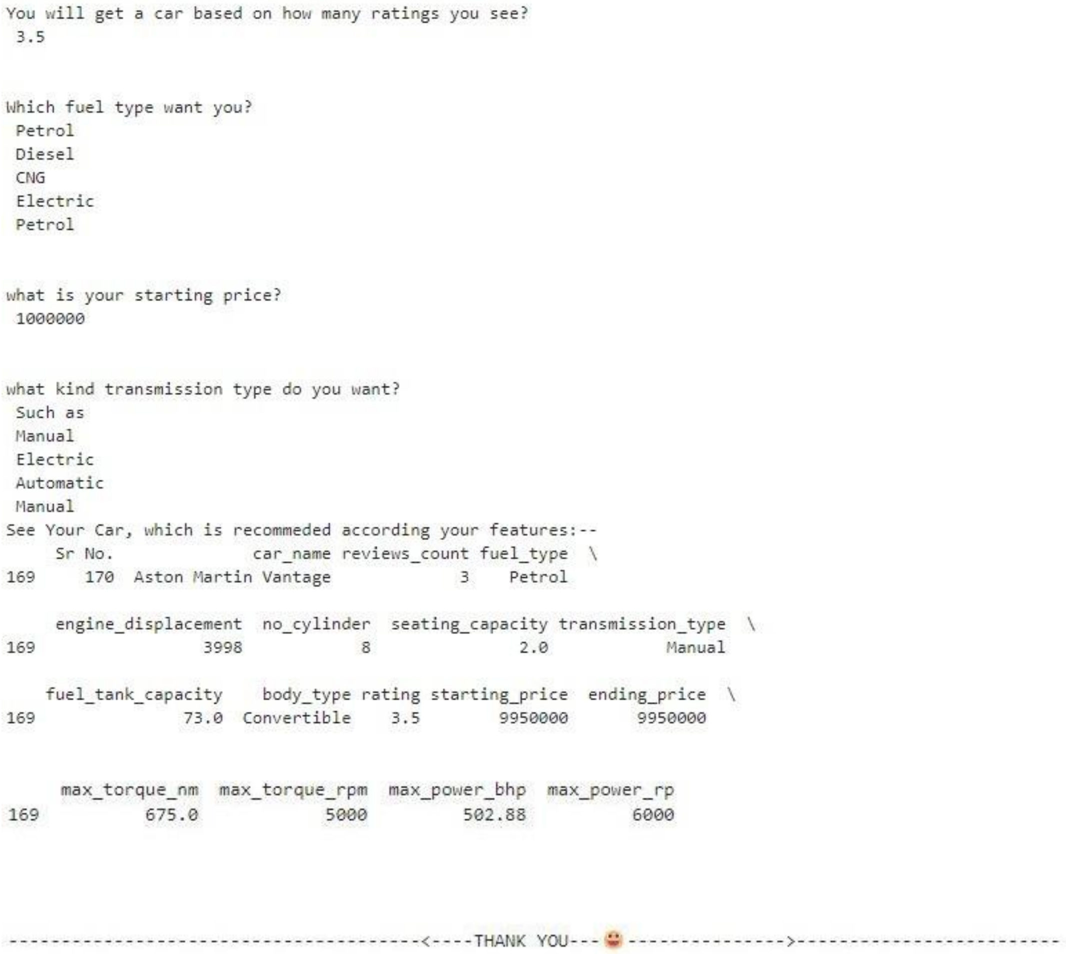
Algorithm Used:

Linear Regression:

Linear Regression is a fundamental algorithm in car recommendation systems, as it allows us to predict user preferences based on historical data. By analyzing user behavior, vehicle attributes, and interactions, Linear Regression models can estimate the likelihood of a user's interest in a particular car. This paper discusses the implementation of Linear Regression in detail and evaluates its performance in enhancing the accuracy and personalization of car recommendations.

In conclusion, this paper explores the key filtration strategies used in car recommendation systems, with a special emphasis on the Linear Regression algorithm. Through a comprehensive dataset and pseudo code representation, we provide insights into the workings of these recommendation systems and demonstrate their effectiveness in helping users make informed decisions when choosing their next vehicle.

OUTPUT:



```
You will get a car based on how many ratings you see?
 3.5


Which fuel type want you?
 Petrol
 Diesel
 CNG
 Electric
 Petrol


what is your starting price?
 1000000


what kind transmission type do you want?
 Such as
 Manual
 Electric
 Automatic
 Manual
See Your Car, which is recommeded according your features:--
     Sr No.              car_name reviews_count fuel_type  \
169     170  Aston Martin Vantage             3    Petrol

     engine_displacement  no_cylinder  seating_capacity transmission_type  \
169                 3998            8               2.0            Manual

    fuel_tank_capacity    body_type rating starting_price  ending_price  \
169               73.0  Convertible    3.5        9950000       9950000


     max_torque_nm  max_torque_rpm  max_power_bhp  max_power_rp
169          675.0            5000         502.88          6000



-----------------------------------------<----THANK YOU---😃---------------->------------------------
```