**Car Purchase Recommendation System Design Report**

**1. Introduction**

**1.1 Problem Statement**

The car purchasing process often overwhelms consumers due to the multitude of choices and factors, such as budget constraints, brand preferences, fuel types, and specific features. Many potential buyers experience decision fatigue, leading to suboptimal purchases. The objective of this project is to develop a car purchase recommendation system that assists users by providing tailored car suggestions based on their unique preferences and requirements.

**1.2 Objectives**

The key objectives of this project include:

* **Development of a Web-Based Application**: A streamlined and intuitive web application will be developed using Streamlit to collect user preferences, such as budget, car type, brand, and fuel type.
* **Implementation of a Deep Learning Model**: The system will utilize a Multi-Layer Perceptron (MLP) neural network model trained on an extensive dataset to predict car recommendations effectively.
* **Ensuring Scalability and Security**: The system will be designed to handle increased user demand and data volume securely and efficiently, allowing for future growth.

**1.3 Relevance and Impact**

This system is designed to simplify the car-buying process, enabling users to make informed decisions with greater ease. By leveraging machine learning algorithms, the system aims to enhance user satisfaction and confidence in their car purchase choices.

**2. System Architecture Overview**

**2.1 Components and Interactions**

The architecture of the system is composed of several interconnected components, each serving a specific purpose:

* **User Interface (UI) and Experience Design**: Built using Streamlit, this component offers a user-friendly interface for inputting preferences and viewing car recommendations.
* **Data Processing Module**: Responsible for cleaning and preprocessing user inputs, converting them into a format suitable for model prediction.
* **Model Prediction Engine**: The core of the system, utilizing a pre-trained MLP neural network to analyze processed inputs and generate car recommendations.
* **Feedback Mechanism** (Optional): Collects user feedback on the recommendations, which can be used to enhance the model's predictive accuracy over time.
* **Database Module** (Optional): A database stores user inputs, car data, and prediction history for ongoing analysis and system improvement.

**3. Detailed Design and Methodology**

**3.1 Data Preparation and Feature Engineering**

1. **Data Collection**:
   * The dataset includes 20,000 entries with various car attributes, such as price, brand, model, fuel type, and engine size. This dataset serves as the foundation for training the recommendation model.
2. **Data Cleaning**:
   * **Handling Missing Values**: Missing values are filled with the most common value for categorical data (e.g., car brand) or the average for numerical data (e.g., price).
   * **Removing Duplicates and Outliers**: Duplicate entries and outliers are identified and removed to maintain data quality and ensure accurate model training.
3. **Feature Engineering**:
   * **One-Hot Encoding**: Categorical variables, like car brand and fuel type, are transformed into numerical format using one-hot encoding, allowing the model to process these features effectively.
   * **Normalization**: Numerical features, such as price and engine size, are normalized using Min-Max scaling. This standardization ensures all input features are on a comparable scale, enhancing the model's learning process.
4. **Feature Selection**:
   * Features are selected based on their correlation with the target variable and domain relevance. This step reduces model complexity and improves predictive performance.

**3.2 Model Selection and Training**

1. **Model Architecture**:
   * A Multi-Layer Perceptron (MLP) neural network is utilized, structured as follows:
     + **Input Layer**: Receives user input features, such as budget and preferred car attributes.
     + **Hidden Layers**: Comprises multiple layers of neurons that identify complex patterns in the data. Each layer progressively refines the understanding of how different features impact the car recommendation.
     + **Output Layer**: Produces a probability score for each car option, ranking their suitability based on the user’s input.
2. **Training Process**:
   * **Loss Function (Cross-Entropy Loss)**: This function measures the model's prediction accuracy. It penalizes incorrect predictions, prompting the model to adjust and improve.
   * **Optimizer (Stochastic Gradient Descent)**: The optimizer iteratively updates the model parameters to minimize the loss. A learning rate of 0.01 and momentum of 0.9 guide the speed and stability of these updates.
   * **Batch Size and Epochs**: The model is trained using batches of 32 data points over 50 epochs, allowing for gradual improvement through multiple learning cycles.
3. **Model Evaluation**:
   * The data is divided into training (70%), validation (15%), and test (15%) sets. Performance is evaluated using metrics such as accuracy, precision, recall, F1-score, and a confusion matrix to provide a comprehensive assessment of the model's effectiveness.

**3.3 System Workflow and Data Flow**

1. **User Input and Preprocessing**:
   * Users input their preferences through the Streamlit interface. These inputs are then preprocessed to match the format required by the model.
2. **Model Prediction**:
   * The processed data is passed to the MLP model, which generates a ranked list of car recommendations.
3. **Output Display**:
   * The recommendations are displayed in a ranked list on the Streamlit app, with options to sort and filter results by various criteria, such as price or brand.
4. **Feedback Collection** (Optional):
   * Users can provide feedback on the recommendations, which can be used to refine the model and improve its accuracy over time.

**3.4 User Interface and Experience Design**

1. **Homepage**:
   * The homepage provides an overview of the application’s purpose and a call-to-action button to start the recommendation process.
2. **Input Form Page**:
   * This page contains a form where users enter their preferences, such as budget, car type, preferred brands, and fuel type. The form uses intuitive elements like dropdown menus and checkboxes.
3. **Results Page**:
   * The results page displays a ranked list of recommended cars, including key attributes like price, brand, and fuel efficiency. Users can sort and filter the recommendations and provide feedback on their usefulness.

**3.5 Database Design (Optional)**

1. **User Inputs Table**:
   * A table for storing user input data, including preferences and interaction history.
2. **Car Data Table**:
   * A table containing detailed information about each car, such as brand, model, price, fuel type, and specifications.
3. **Prediction History Table**:
   * A table recording the recommendations provided to users and any feedback received, which can be used to improve the model’s accuracy over time.

**3.6 Deployment Strategy**

1. **Deployment Platform**:
   * The application will initially be deployed using Streamlit Sharing for quick deployment. For larger-scale usage, platforms like AWS or Google Cloud Platform (GCP) will be considered to ensure scalability and performance.
2. **CI/CD Pipeline**:
   * A Continuous Integration/Continuous Deployment (CI/CD) pipeline will be set up using GitHub Actions to automate testing and deployment, minimizing errors and facilitating smooth updates.
3. **Environment Management**:
   * Docker will be used to create consistent environments across development and production, ensuring reliable application performance across different platforms.

**3.7 Security and Privacy**

1. **Data Encryption and Privacy**:
   * All data transmission between the user and the server will be encrypted using HTTPS to protect sensitive information. Access controls will be implemented to ensure only authorized personnel can access sensitive data.
2. **Regular Security Audits**:
   * Regular security audits will be conducted to identify and address any potential vulnerabilities, maintaining the system's integrity and security.

**4. Risk Management**

**4.1 Technical and Business Risks**

1. **Model Overfitting**:
   * **Risk**: The model might perform well on training data but poorly on new, unseen data.
   * **Mitigation**: Techniques such as dropout regularization and early stopping will be used to prevent overfitting and ensure the model generalizes well.
2. **Data Quality and Bias**:
   * **Risk**: Biases in the dataset could skew the model’s predictions.
   * **Mitigation**: Rigorous data cleaning and preprocessing will be performed. Additionally, techniques such as oversampling underrepresented categories will be used to balance the dataset.
3. **Scalability and Market Acceptance**:
   * **Risk**: The system may face challenges in handling a growing number of users or fail to meet market expectations.
   * **Mitigation**: The system will be deployed on a scalable cloud platform, and user feedback will be actively collected and incorporated to continuously improve system functionality.

**5. Project Timeline**

**5.1 Gantt Chart and Task Allocation**

* A Gantt chart will be used to outline the project timeline, detailing each phase, including data preparation, model training, UI development, testing, and deployment. This timeline will help track progress and ensure that all tasks are completed on schedule.

**5.2 Team Roles and Responsibilities**

* Clear roles and responsibilities will be assigned to each team member to optimize collaboration and task management. For example, one member may focus on data preparation while another is responsible for model development and testing.

**6. Conclusion**

The car purchase recommendation system is designed to assist users in making informed car-buying decisions by providing personalized recommendations based on their preferences. This report presents a detailed design that is ready for immediate implementation, ensuring no technical gaps. The next steps will involve developing the application, deploying it, and continuously refining it based on user feedback and performance metrics.

**7. References**

A comprehensive list of references will be provided to support the research and design process, ensuring all sources are properly acknowledged.