Project Report: Car Depreciation Prediction Application

1. Overview

This project focuses on developing a machine learning-based web application that predicts the **current market value** of a used car and forecasts its **depreciation over the next three years**. The objective is to help users—such as individual buyers, sellers, or dealerships—make informed decisions about car purchases or sales.

2. Application Features

Current Price Prediction

The user provides the following inputs:

- Manufacturer
- Model
- Title Status
- Car Age
- Odometer Reading

Based on these inputs, the application estimates the **current market price** of the car using a trained regression model.

Future Price Forecasting

The app also predicts how the car's value will change over time:

- Price is forecasted for **Year +1**, **Year +2**, and **Year +3**
- The logic adjusts the car's age and odometer reading each year and applies the same trained model for each future prediction

This feature gives users a **3-year depreciation projection**, which is helpful for financial planning or resale timing.

Prediction Logic for Depreciation

- **Model Used:**
  Random Forest Regressor (baseline model chosen for its performance and

interpretability)

- **Pipeline Flow:**
  1. Input data is collected via a web form
  2. Categorical features (like model, manufacturer, title status) are encoded
  3. The processed data is passed into the trained model
  4. The model outputs the predicted price
- **Future Forecasting Logic:**
  Car age is incremented by 1 year at a time; odometer reading is also increased based on average mileage assumptions, and the model is re-applied for each year.

5. Deployment

- **Backend:** Flask web application with a RESTful API that accepts user inputs and returns predictions
- **Hosting Platform:** AWS Elastic Beanstalk
- **User Access:** Simple web interface (HTML form) that allows real-time interaction and result display

The application is currently deployed and accessible on both laptop and mobile (HTTP only). HTTPS/SSL integration and improved UI/UX are planned for future releases.

4. Planned Improvements

Due to **local system limitations**, the current model was trained on a **subset of only 50,000 rows** (10% of the full dataset) to avoid memory issues. To enhance model performance and generalizability, the following improvements are planned:

Feature Expansion

Include additional features that influence car price and depreciation:

- `condition`
- `cylinders`
- `drive`

- `paint color`
- `size`
- `transmission`

Model Experimentation

Experiment with advanced regression models:

- **XGBoost**
- **LightGBM**
- **Gradient Boosting Regressor**
- **Support Vector Regression (SVR)**

These models may outperform Random Forest in terms of predictive accuracy.

Hyperparameter Tuning

Use tools like:

- `GridSearchCV` and `RandomizedSearchCV` for exhaustive tuning

Cross-Validation

Implement **k-fold cross-validation** (e.g., k=5 or 10) to ensure the model performs consistently across different data splits.

Cloud-Based Training

To overcome hardware limitations, future versions will:

- Train the model on **full dataset (500k rows)**
- Use **cloud resources** (e.g., AWS EC2, Google Colab Pro, or Databricks) to handle large-scale training and tuning efficiently