Artificial Intelligence Project Report

Section: A

Submitted by:

Fiza Manzoor 2023-CS-612

Samar Noor 2023-CS-630

Submitted to:

Mam. Shanfa Irum

Dated: 4th May 2025

Department of Computer Science

University of Engineering and Technology Lahore, New Campus

🚗 Car Price Prediction – Project Report

Introduction:

The objective of the Car Price Prediction System is to develop a machine learning model that predicts car prices based on various features such as brand, model, year, and mileage. The system aims to provide accurate and reliable price estimates by analyzing historical data and applying algorithms like regression. It utilizes tools such as Pandas, NumPy, and Scikit-learn for data processing and model training, offering a user-friendly interface via Streamlit for real-time price predictions. This tool helps users make informed decisions in buying and selling cars.

This project is divided into **two main Python files**:

1. Model_Training.py – for training the machine learning model.
2. Price_Pridictor.py – for building the GUI and using the trained model to predict car prices.

File 1: Model_Training.py:

This script automates a complete Machine Learning (ML) pipeline for:

- **Regression** (predicting car prices)
- **Classification** (categorizing prices into classes)
- **Clustering** (grouping cars)
- It uses the **car_data.csv** dataset, which must have **at least 3,000 entries** and **20+ features**.
- The models and preprocessing tools are saved for future inference.

Project Directory Setup:

MODEL_DIR = "models"

Creates a folder models/ to store all trained models and preprocessing artifacts:

- Regression models
- Classification models
- Clustering model

- Scaler object
- Encoders
- Column names

Dataset Requirements Fulfilled:

- ✓ **7 Classes**
- ✓ **at least 3K Instances**
- ✓ **28 Features**

Functions and Features:

1. load_data(file_path)

Features:

- Loads CSV file using pandas.
- Trims or raises error based on row count.
- Checks for essential columns.
- Returns cleaned and validated DataFrame.

2. preprocess_data(data)

Features:

- Creates a **price class** (for classification) using pd.qcut.
- Performs **target encoding** for categorical features make and model.
- Handles missing values
- **Scales** numerical features using StandardScaler.
- Returns: preprocessed data, scaler object, encoders dictionary

3. train_regression_models(X, Y)

Models:

- **KNN Regressor (Classification)**
- **Gradient Boosting Regressor (Regression)**

- **Naïve Bayes (Clustering)**

4. train_classification_models(X, Y)

Models:

- **KNN Classifier**
- **Naïve Bayes**
- **Gradient Boosting Classifier**

Features:

- Uses **GridSearchCV** where applicable.
- Evaluates with:
    - **Accuracy**
    - **Precision**
    - **Recall**
    - **F1 Score**
- Returns best-tuned classifiers.

5. perform_clustering(X, n_clusters=7)

Features:

- Uses **K-Means Clustering** with n_clusters=7
- Evaluates using **Silhouette Score**
- Returns fitted KMeans object.

6. save_models(models, scaler, encoders, X_train_cols)

Features:

- Saves all models (joblib) to disk:
    - Regression

- Classification
- Clustering
- Scaler
- Encoders
- Feature columns used in training

7. main() Function

Features:

- Orchestrates the full ML workflow:
    1. Loads dataset
    2. Preprocesses it
    3. Prepares data for regression/classification
    4. Trains models
    5. Performs clustering
    6. Saves all objects to disk

Machine Learning Concepts Used:

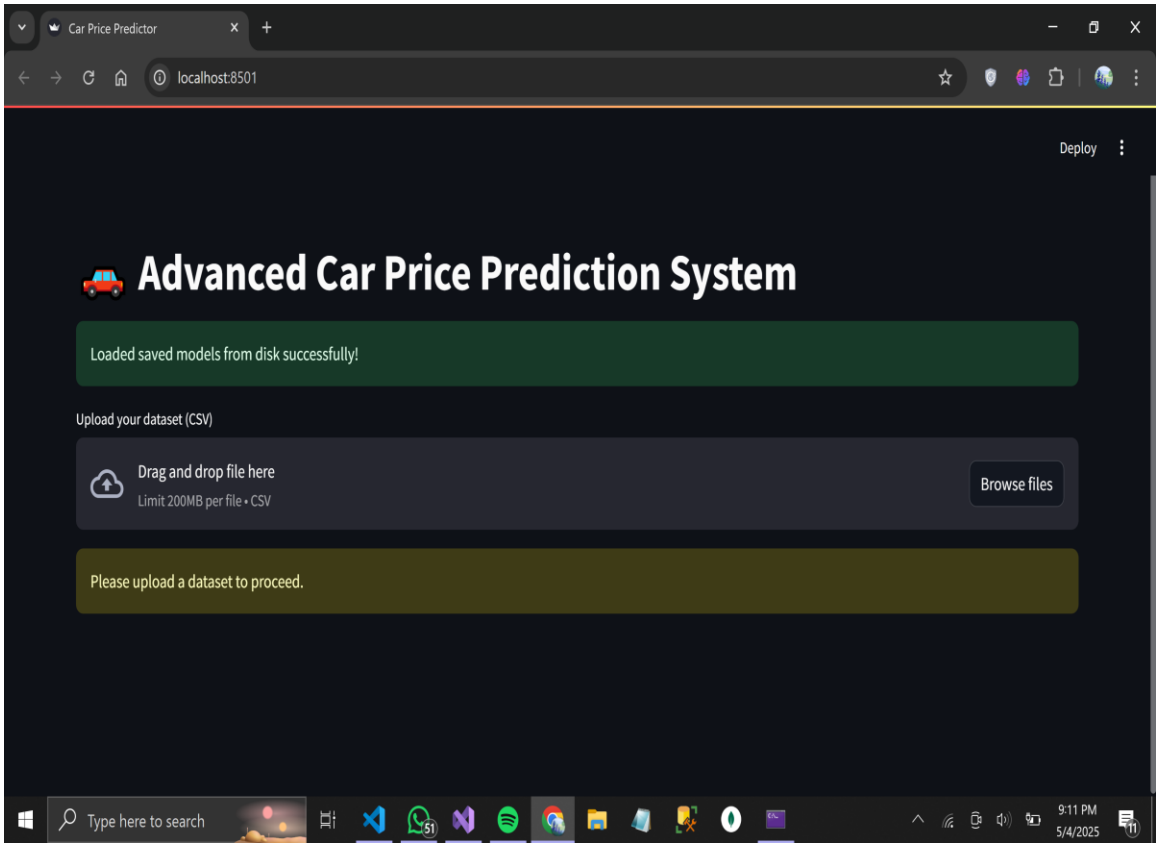
Concept	Usage
Supervised Learning	✓ Regression & Classification
Unsupervised Learning	✓ Clustering
Model Selection	✓ GridSearchCV
Feature Scaling	✓ StandardScaler
Encoding	✓ Target Encoding, Label Encoding

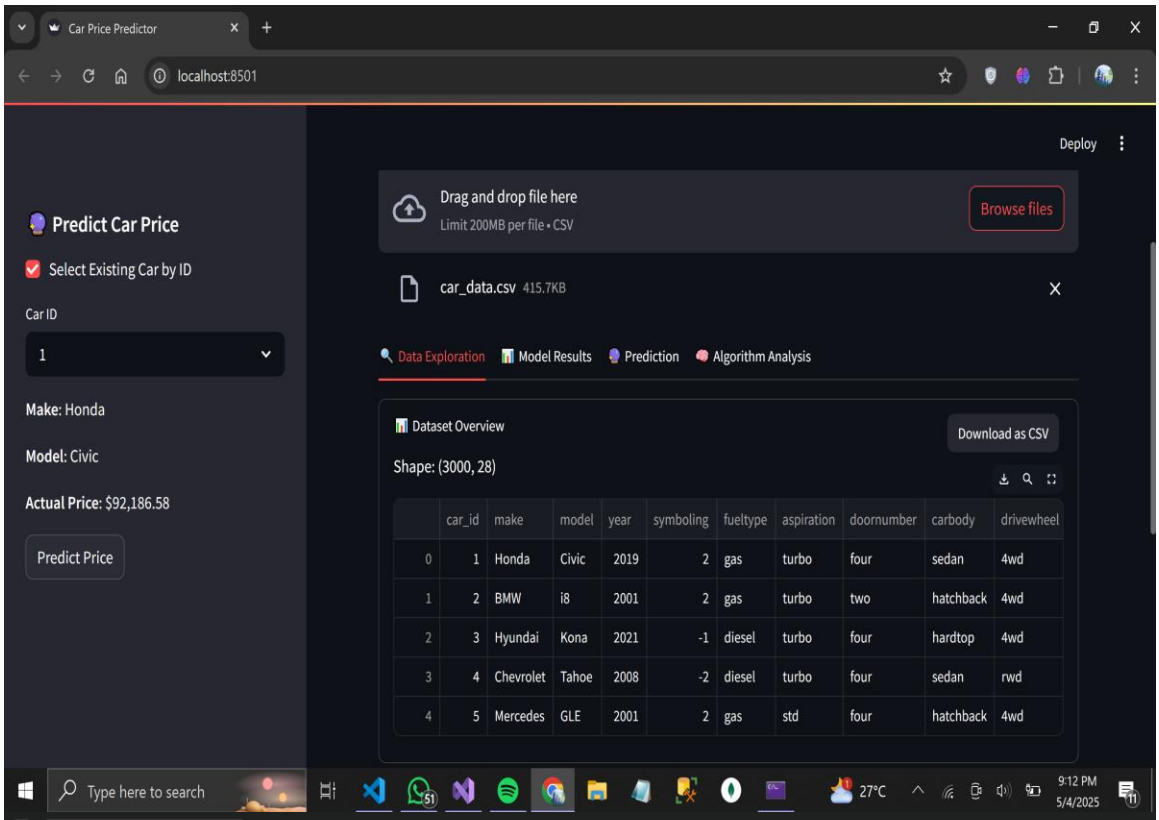
Output Files:

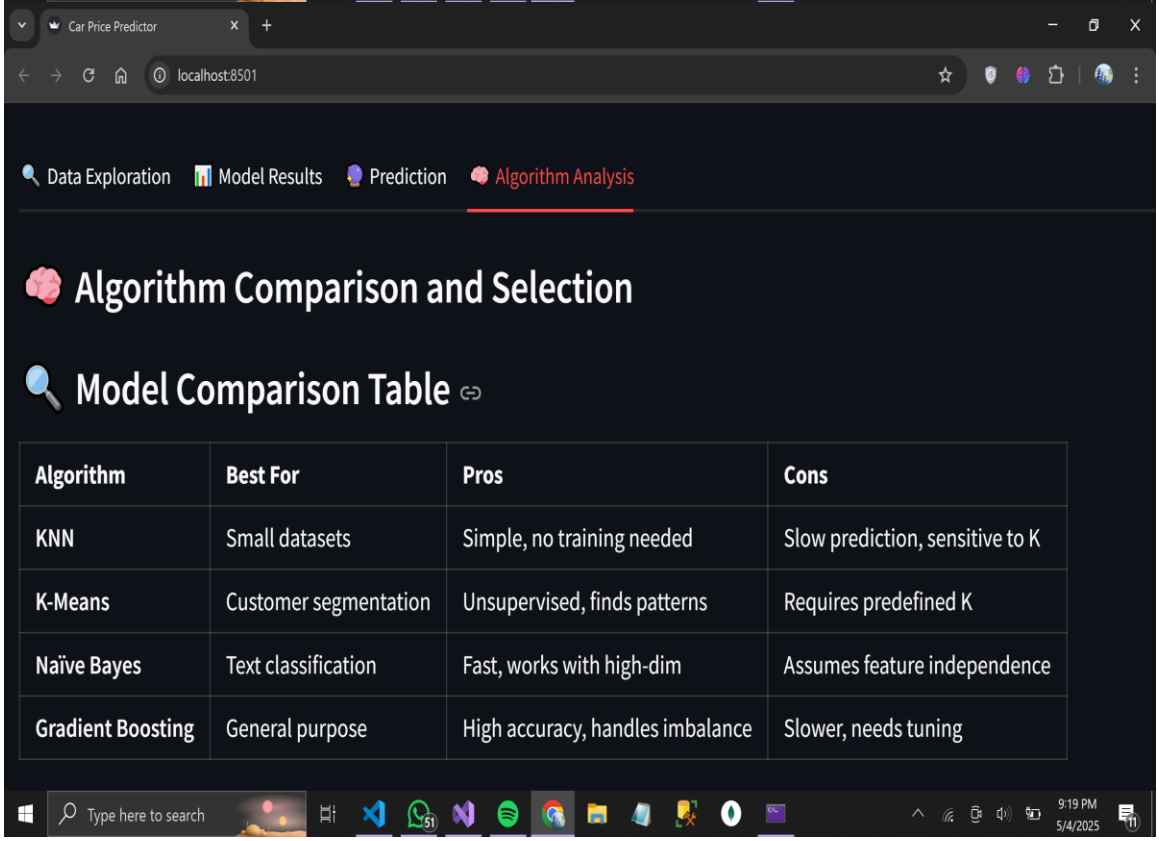
Saved inside models/:

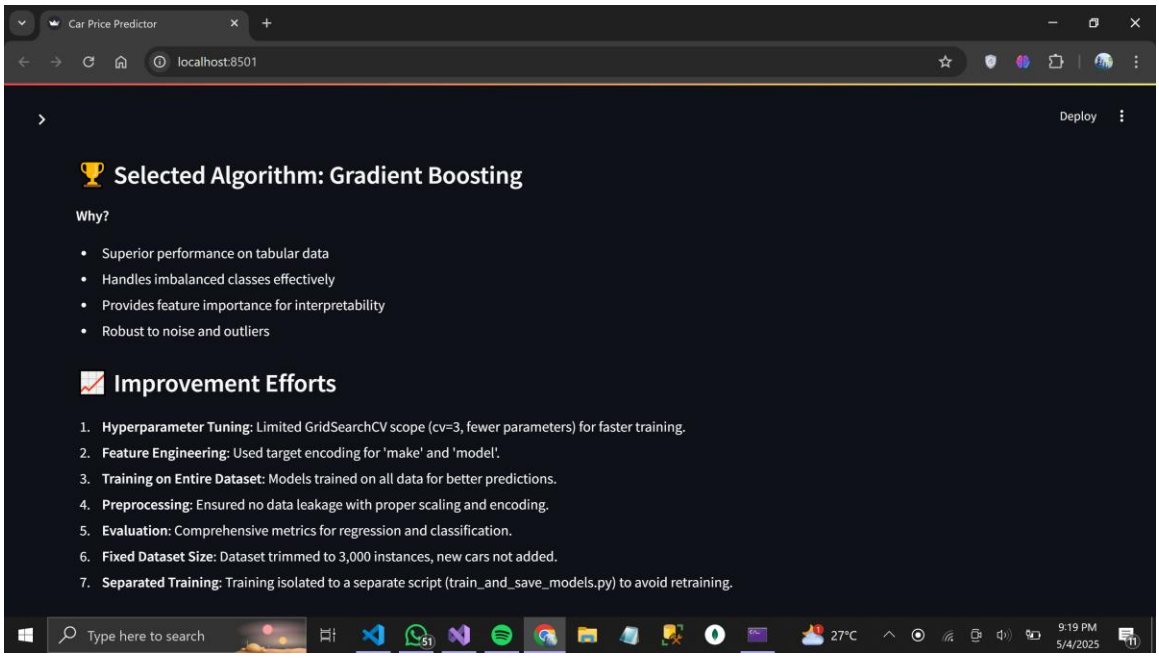
- regression_models.joblib
- classification_models.joblib
- clustering_model.joblib
- scaler.joblib
- encoders.joblib
- X_train_cols.joblib

Project Flow:




Car Price Predictor
localhost:8501
Deploy
Predict Car Price
Select Existing Car by ID
Car ID
1
Make: Honda
Model: Civic
Actual Price: $92,186.58
Predict Price
Drag and drop file here
Limit 200MB per file • CSV
Browse files
car_data.csv 415.7KB
Data Exploration
Model Results
Prediction
Algorithm Analysis
Dataset Overview
Download as CSV
Shape: (3000, 28)
car_id make model year symboling fueltype aspiration doornumber carbody drivewheel
0 1 Honda Civic 2019 2 gas turbo four sedan 4wd
1 2 BMW i8 2001 2 gas turbo two hatchback 4wd
2 3 Hyundai Kona 2021 -1 diesel turbo four hardtop 4wd
3 4 Chevrolet Tahoe 2008 -2 diesel turbo four sedan rwd
4 5 Mercedes GLE 2001 2 gas std four hatchback 4wd
Type here to search
27°C
9:12 PM
5/4/2025


Car Price Predictor
localhost:8501
Data Exploration
Model Results
Prediction
Algorithm Analysis
Algorithm Comparison and Selection
Model Comparison Table
Algorithm | Best For | Pros | Cons
KNN | Small datasets | Simple, no training needed | Slow prediction, sensitive to K
K-Means | Customer segmentation | Unsupervised, finds patterns | Requires predefined K
Naïve Bayes | Text classification | Fast, works with high-dim | Assumes feature independence
Gradient Boosting | General purpose | High accuracy, handles imbalance | Slower, needs tuning
Type here to search
9:19 PM
5/4/2025



File 2: Price_Predictor.py:

1. Imports:

- Pandas (pd): Data manipulation, especially for reading and transforming data.
- NumPy (np): Essential for numerical data, arrays, and math functions.
- Matplotlib (plt) & Seaborn (sns): Data visualization and creating plots.
- Scikit-learn (sklearn): Machine learning tools:

- Metrics: Evaluate model performance (e.g., accuracy, MSE).
- PCA: Dimensionality reduction.
- LabelEncoder: Converts categorical labels to numeric.
- Streamlit (st): Builds interactive web apps for UI and user input/output.
- Warnings: Suppresses specific warnings.
- UUID: Generates unique identifiers (e.g., car IDs).
- OS: Handles file paths and directories.
- Joblib: Saves and loads Python objects (e.g., ML models).
- Counter (from collections): Counts occurrences of elements, used in classification tasks.

2. Constants

- MODEL_DIR: Directory where the models are stored.

- MODEL_PATHS: Paths to various saved models (e.g., regression models, classification models, encoders, scaler).

3. load_data Function

This function loads a dataset from a CSV file and performs a series of validation checks:

- It ensures the dataset has at least 3,000 instances and at least 20 features.
- Normalizes column names to lowercase.
- Checks for required columns like car_id, make, model, and price. If these are missing, it shows an error message.
- If any of these conditions fail, an error or warning message is displayed in the Streamlit interface.

4. explore_data Function

This function provides a comprehensive overview of the dataset:

- **Shape**: Displays the dimensions (rows and columns) of the dataset.
- **Description**: Provides descriptive statistics like mean, standard deviation, etc., for numeric columns.
- **Missing Values**: Checks for and displays any missing values in the dataset.
- **Price Distribution**: Uses Seaborn to plot a histogram with a KDE of the price column to visualize its distribution.
- **Price Class Ranges**: Divides the price column into 7 quantile-based price classes and shows the minimum and maximum values for each class.
- **Correlation Matrix**: Visualizes the correlation between numeric features using a heatmap. Categorical features are encoded using LabelEncoder before correlation calculation.

5. preprocess_data Function

This function prepares the dataset for machine learning models:

- **Price Classification**: If the dataset is used for training, it creates a price_class based on the price column, dividing it into 7 categories.
- **Data Imputation**: Fills missing values in numerical columns with the median and in categorical columns with the mode.

- **Target Encoding**: For the make and model columns, it maps them to their target means (the average price for each make/model).
- **Scaling**: Scales the numerical columns using a provided scaler (standard scaler or any other type), except for the price column.

6. perform_clustering Function

- **PCA**: Reduces the dimensionality of the feature set to two components for visualization.
- **KMeans**: Predicts the clusters using a K-Means model and calculates the silhouette score, which measures the quality of the clusters (higher is better).

7. plot_results Function

This function visualizes and displays the results of the machine learning models:

- **Regression Results**: Displays RMSE (Root Mean Squared Error) and R^2 (coefficient of determination) comparisons between different models in bar charts.
- **Classification Results**: Displays Accuracy, Precision, Recall, and F1 Score for classification models using bar charts.
- **K-Means Clustering**: Visualizes the clustering results after dimensionality reduction using PCA. The silhouette score is also shown to indicate the quality of the clustering.

8. prediction_interface Function

This is an interactive Streamlit interface that allows users to predict car prices:

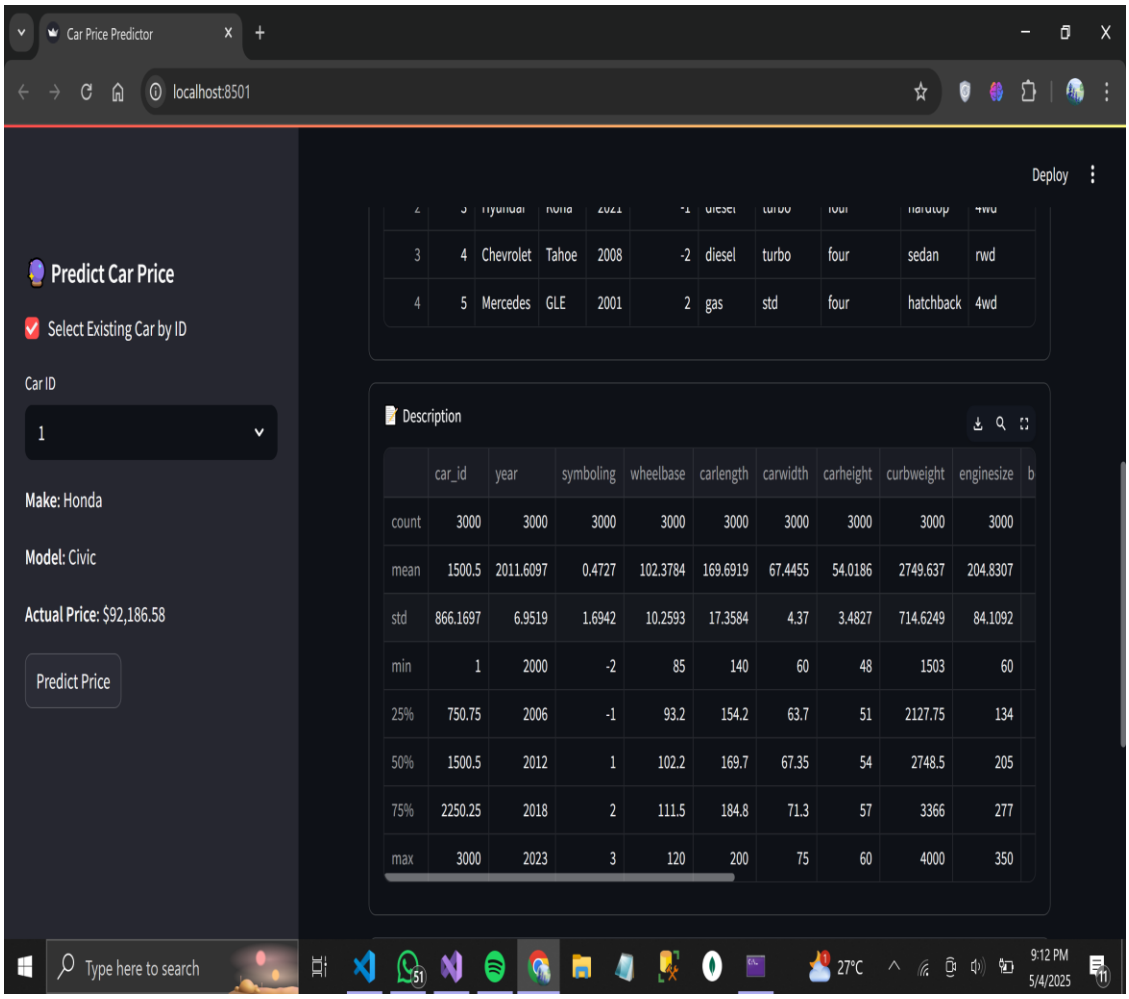
- Users can either select an existing car from the dataset or enter details for a new car.
- It gathers information like make, model, and other numerical features from the user.
- The input is processed (scaled, encoded) before being passed to the regression models for price prediction.
- The function also provides an ensemble prediction (average of all regression models' predictions) and displays the difference between the actual and predicted prices if the car is from the existing dataset.

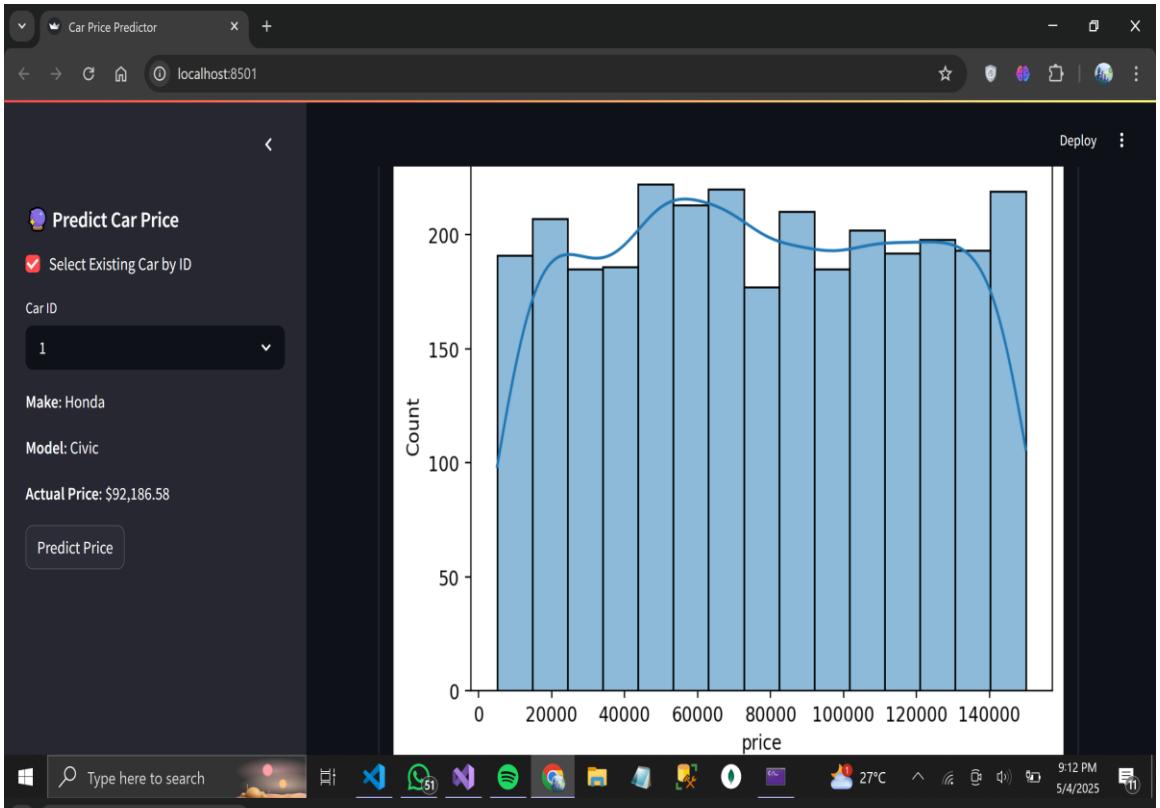
- Additionally, it predicts the car's price class (e.g., "High", "Medium") using the classification models.

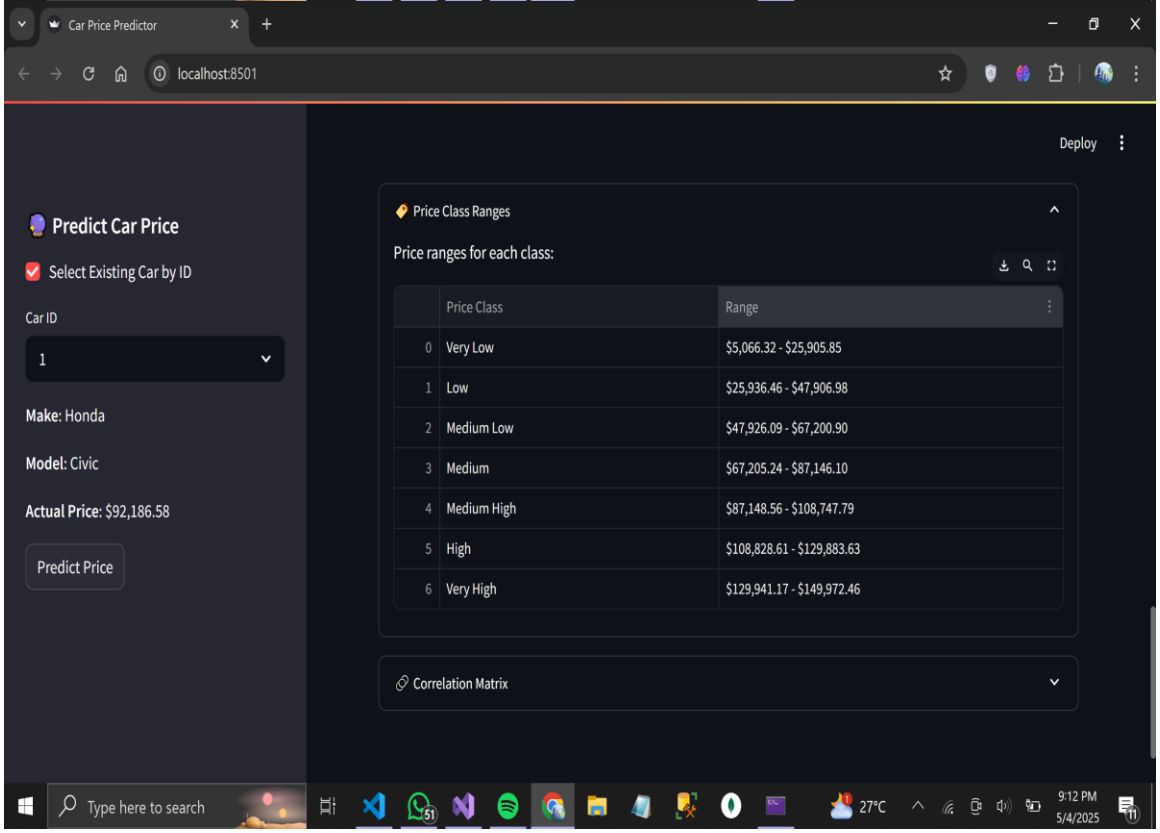
9. algorithm_comparison Function

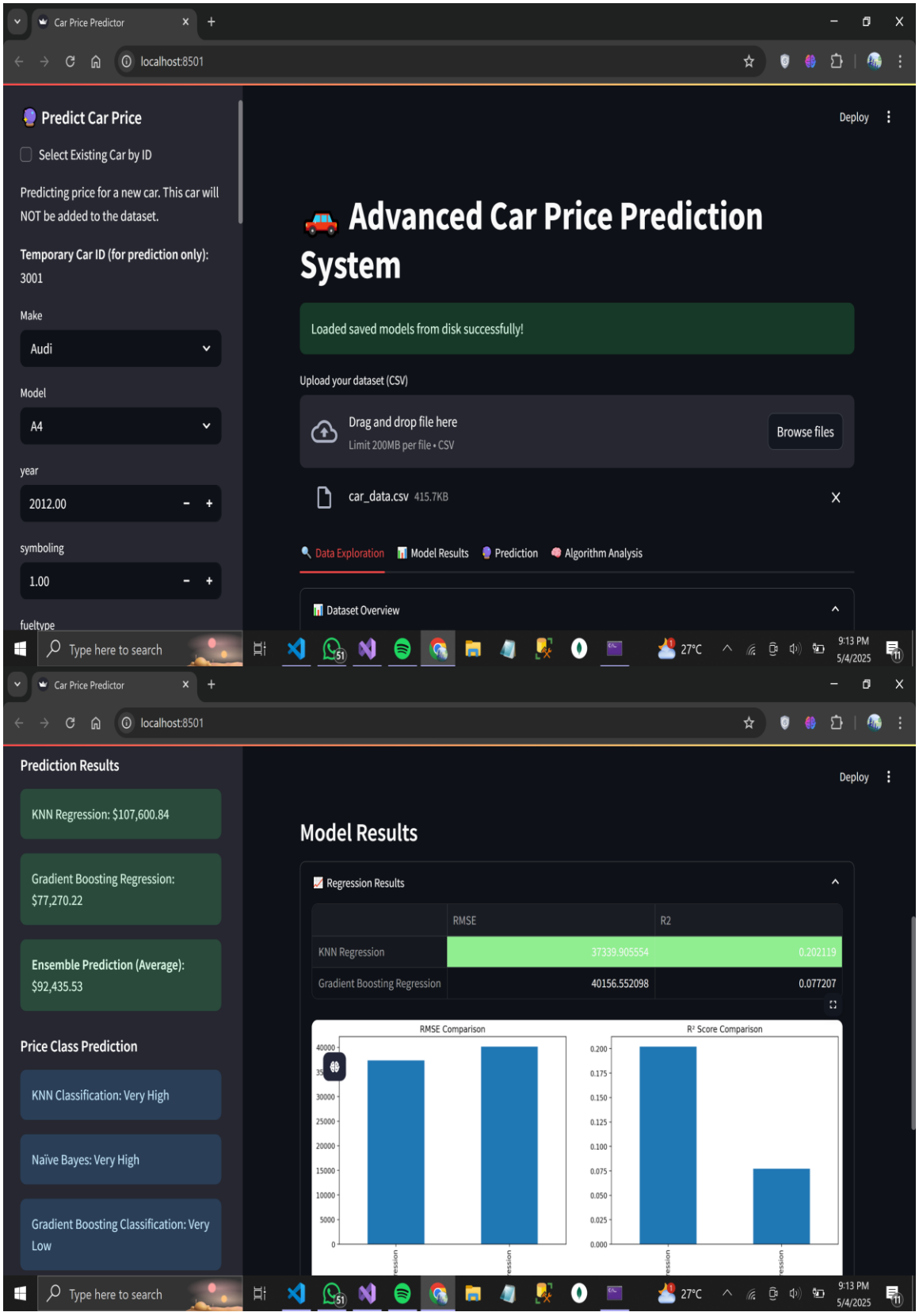
This function compares different machine learning algorithms and presents them in a table format, highlighting the strengths and weaknesses of each algorithm. For example, it mentions:

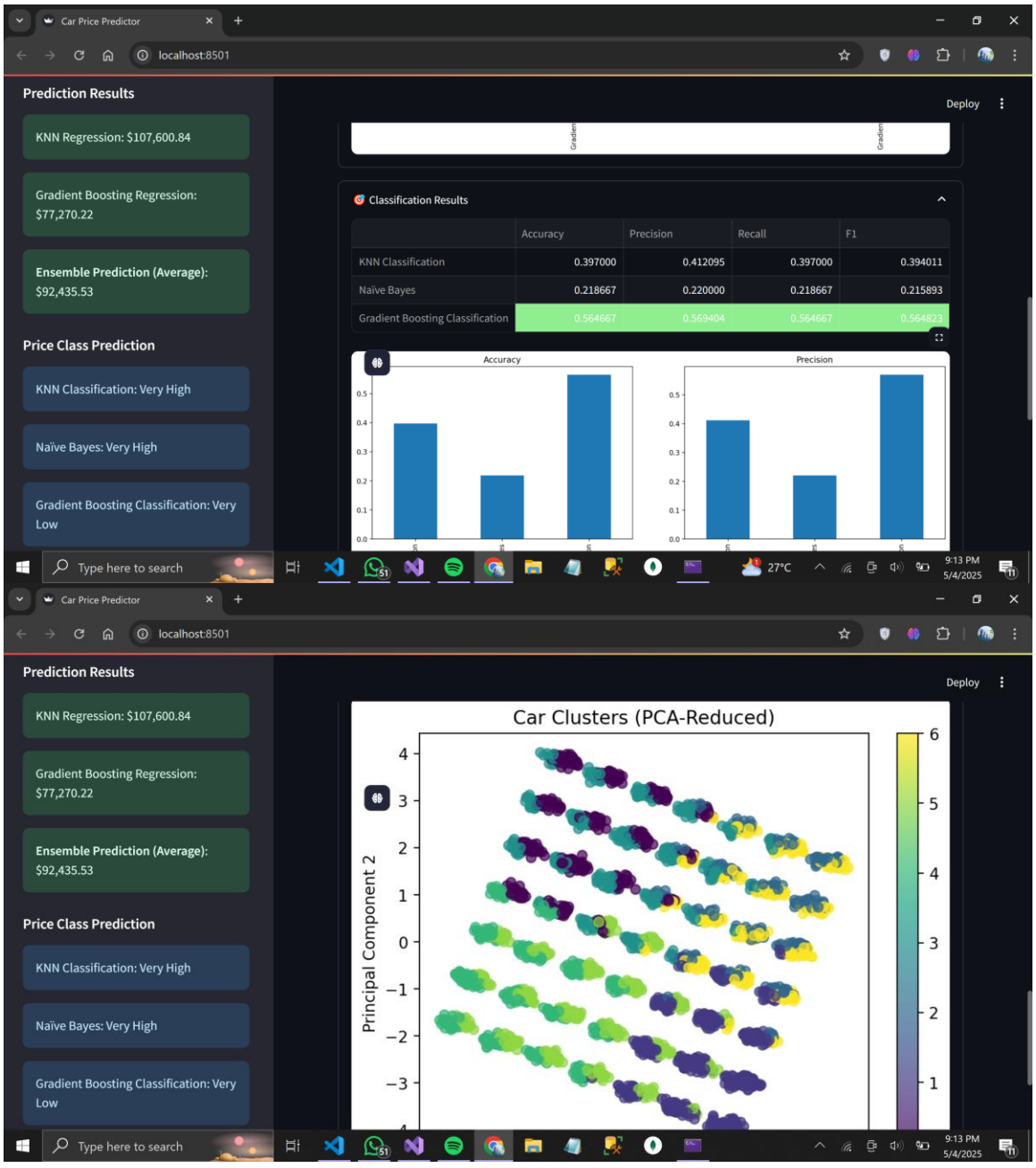
- **KNN**: Good for small datasets but slow during predictions.
- **K-Means**: Useful for clustering but requires predefined cluster numbers.
- **Naive Bayes**: Ideal for text classification but assumes feature independence.




Car Price Predictor
localhost:8501
Deploy
Predict Car Price
Select Existing Car by ID
Car ID
1
Make: Honda
Model: Civic
Actual Price: $92,186.58
Predict Price
Count
price
0
50
100
150
200
0
20000
40000
60000
80000
100000
120000
140000
Type here to search
27°C
9:12 PM
5/4/2025


Car Price Predictor
localhost:8501
Deploy
Predict Car Price
Select Existing Car by ID
Car ID
1
Make: Honda
Model: Civic
Actual Price: $92,186.58
Predict Price
Price Class Ranges
Price ranges for each class:
Price Class
Range
0 Very Low $5,066.32 - $25,905.85
1 Low $25,936.46 - $47,906.98
2 Medium Low $47,926.09 - $67,200.90
3 Medium $67,205.24 - $87,146.10
4 Medium High $87,148.56 - $108,747.79
5 High $108,828.61 - $129,883.63
6 Very High $129,941.17 - $149,972.46
Correlation Matrix
Type here to search
27°C
9:12 PM
5/4/2025


Car Price Predictor
localhost:8501
Predict Car Price
Select Existing Car by ID
Predicting price for a new car. This car will NOT be added to the dataset.
Temporary Car ID (for prediction only):
3001
Make
Audi
Model
A4
year
2012.00
symboling
1.00
fueltype
Deploy
Advanced Car Price Prediction System
Loaded saved models from disk successfully!
Upload your dataset (CSV)
Drag and drop file here
Limit 200MB per file • CSV
Browse files
car_data.csv 415.7KB
Data Exploration
Model Results
Prediction
Algorithm Analysis
Dataset Overview
Type here to search
27°C
9:13 PM
5/4/2025
Car Price Predictor
localhost:8501
Prediction Results
KNN Regression: $107,600.84
Gradient Boosting Regression: $77,270.22
Ensemble Prediction (Average): $92,435.53
Price Class Prediction
KNN Classification: Very High
Naïve Bayes: Very High
Gradient Boosting Classification: Very Low
Deploy
Model Results
Regression Results
RMSE
R2
KNN Regression
37339.905554
0.202119
Gradient Boosting Regression
40156.552098
0.077207
RMSE Comparison
R² Score Comparison
Type here to search
27°C
9:13 PM
5/4/2025



Summary: This code integrates multiple components:

- **Data Loading & Preprocessing**: Handles loading, cleaning, and preparing data for machine learning.
- **Modeling & Evaluation**: Includes regression, classification, and clustering, with evaluation metrics.
- **Interactive UI**: Streamlit is used to create a dynamic and user-friendly interface for data exploration, model predictions, and visualizations.