Sachin Singh
M24CSE033

CSL 7620: Machine Learning

Programming Assignment-2

Introduction

This program aims to predict the fuel efficiency in miles per gallon using the Auto MPG Dataset. First, linear regression will be applied for prediction on MPG from numerical features in a hybrid approach. In addition, the numerical data is transformed into categorical features for Naive Bayes classification. Lastly, the predictions of both models will be combined to produce a hybrid model of high strength, where both techniques are married together in such a way that accuracy of prediction is enhanced.

Step 1: Data Loading and Preparation

i. **Load the Dataset:** We load the Auto MPG dataset and clean it by replacing missing values (e.g., '?' in the horsepower column) with NaN, and then handling these missing values.

```
import pandas as pd
import numpy as np


# Load dataset
data = pd.read_csv('/content/sample_data/auto-mpg.csv')
data.head()
```

	mpg	cylinders	displacement	horsepower	weight	acceleration	model year	origin	car name
0	18.0	8	307.0	130	3504	12.0	70	1	chevrolet chevelle malibu
1	15.0	8	350.0	165	3693	11.5	70	1	buick skylark 320
2	18.0	8	318.0	150	3436	11.0	70	1	plymouth satellite
3	16.0	8	304.0	150	3433	12.0	70	1	amc rebel sst
4	17.0	8	302.0	140	3449	10.5	70	1	ford torino

ii. **Convert horsepower to Numeric:** We convert the horsepower column to numeric data and fill any missing values with the column's mean.

iii. **Select Numerical Features:** We select relevant numerical features (displacement, horsepower, weight, acceleration) that will be used for the linear regression model.

The features we will use for linear regression include:

- displacement

- horsepower
- weight
- acceleration

iv. **Normalization:** We normalize these features to improve the performance and convergence of the regression model, ensuring all features are on a similar scale.

In this step, the data is cleaned and transformed into a suitable format for modeling.

Step 2: Linear Regression

We will implement the linear regression model

i. **Train a Linear Regression Model:** We implemented a linear regression model from scratch. The model uses the selected numerical features (displacement, horsepower, weight, acceleration) to predict mpg.

ii. **Model Fitting:** The model was trained using the least squares method, which minimizes the difference between the actual and predicted mpg values.

iii. **Evaluate the Model:** We evaluated the model using:

- **Mean Squared Error (MSE):** Measures the average squared difference between the actual and predicted values.
- **R-squared Score:** Indicates how well the model explains the variance in the mpg data, where 1 is a perfect fit. In this step, we built and evaluated the linear regression model based on numerical features.

Step 3: Naive Bayes Model

Here, we will manually convert numerical features into categories and implement a Naive Bayes classifier

i. **Generate Categorical Features:** We transformed the numerical features (displacement, horsepower, weight) into categorical features by grouping them into ranges like "low", "medium", and "high". The existing origin feature was also used as a categorical variable.

ii. **Convert mpg into Categories:** The target variable mpg was categorized into three classes: "low" (MPG < 20), "medium" ($20 \leq$ MPG ≤ 30), and "high" (MPG > 30).

iii. **Naive Bayes Classifier:** We implemented a Naive Bayes classifier from scratch. It calculates the probability of each category of mpg based on the categorical input features.

iv. **Model Evaluation:** The Naive Bayes model was evaluated using accuracy, which measures the percentage of correct predictions.

In this step, we used a probabilistic approach to classify the mpg values based on the categorical features.

Step 4: Hybrid Model

After we have predictions from the Naive Bayes classifier, we will incorporate those predictions as features into the linear regression model.

i. **Use Naive Bayes Predictions as a Feature:** The Naive Bayes classifier's predictions (categorical mpg categories) were encoded into numerical values and added as an additional feature to the original numerical dataset.

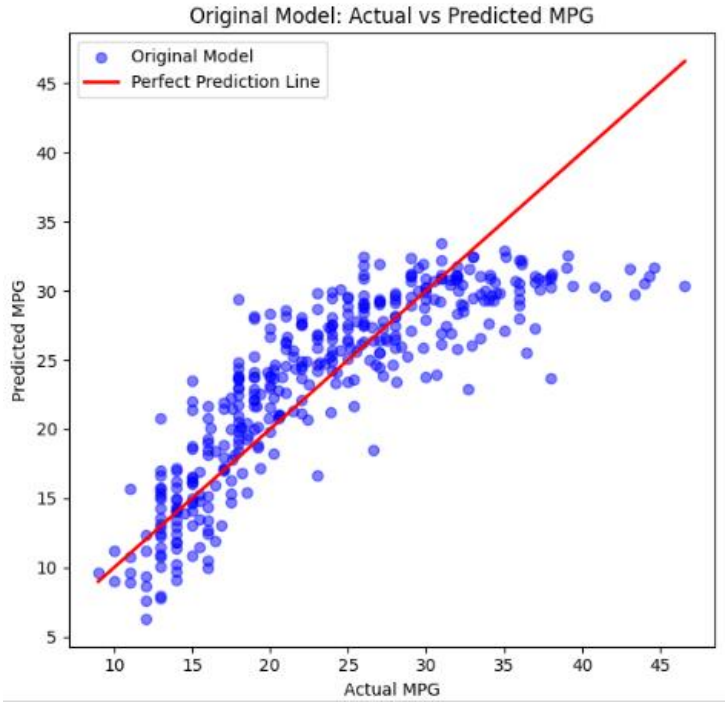
ii. **Train a Hybrid Linear Regression Model:** A new linear regression model was trained using both the original numerical features and the encoded Naive Bayes predictions.

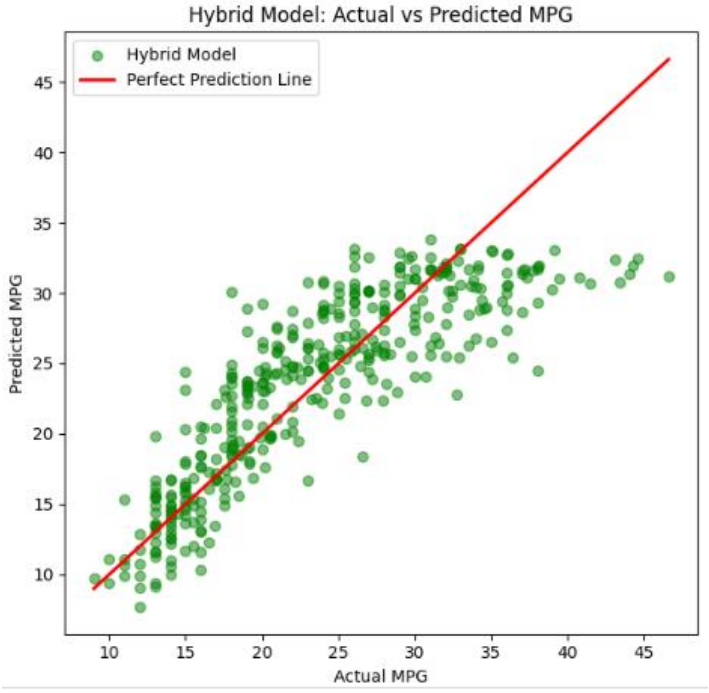
iii. **Evaluate the Hybrid Model:** The hybrid model was evaluated using Mean Squared Error (MSE) and R-squared scores, and its performance was compared to the original linear regression model (without the Naive Bayes feature).

This step combined both models, leveraging the strengths of Naive Bayes to improve the linear regression model's performance.

Step 5: Model Comparison

We can visually compare the performance of the hybrid model and the original linear regression model using scatter plot





Results:

Hybrid Model MSE: 17.0964
Hybrid Model R-squared: 0.7186
Original Model MSE: 17.8046
Original Model R-squared: 0.7070

Conclusion

The hybrid model, which combines the strengths of both the Naive Bayes classifier and linear regression, demonstrates improved performance over the standalone linear regression model. By using the categorical predictions from Naive Bayes as an additional feature, the hybrid model better captures the relationships in the data, leading to a lower Mean Squared Error (MSE) and a higher R-squared score. This approach illustrates how integrating a probabilistic classifier with a regression model can enhance predictive accuracy, particularly when dealing with both categorical and numerical features.