Vehicle Price Prediction Using Tree-Based Models: SHAP and PDP Interpretability

Avinash Bunga

Master of Science in Information Systems and Business Analytics

Park University

CIS625HOS2P2025 Machine Learning for Business

Professor: Abdelmonaem Jornaz

April 12, 2025

Vehicle Price Prediction Using Tree-Based Models: SHAP and PDP Interpretability

Predicting the resale value of vehicles is a key challenge in the automotive industry. Buyers, sellers, and pricing strategists rely on robust models that provide accurate predictions and offer interpretability. This report expands the previous regression work using tree-based ensemble learning methods. Specifically, this analysis utilizes a Random Forest Regressor (bagging) and an XGBoost Regressor (boosting) to model vehicle prices. It interprets model outputs using SHAP (SHapley Additive explanations) and PDP (Partial Dependence Plot).

The dataset used is a cleaned version of the CarDekho vehicle listing dataset, consisting of numeric vehicle specifications and the target variable Price. The data was preprocessed to retain only numeric columns and remove rows with missing values. The dataset was split into 80 percent training and 20 percent testing subsets for consistent evaluation across models (see Appendix A for the code).

Random Forest Regressor: Feature Importance

The Random Forest Regressor model was trained using the training subset, and predictions were evaluated on the test set. The model yielded a root mean squared error (RMSE) of ₹241,946 and an R-squared (R^2) score of 0.9488, indicating a high level of accuracy. Feature importance analysis showed that Width is the most influential factor in predicting vehicle price, followed by Length, Height, and Car_Age (Panpatil, 2025).

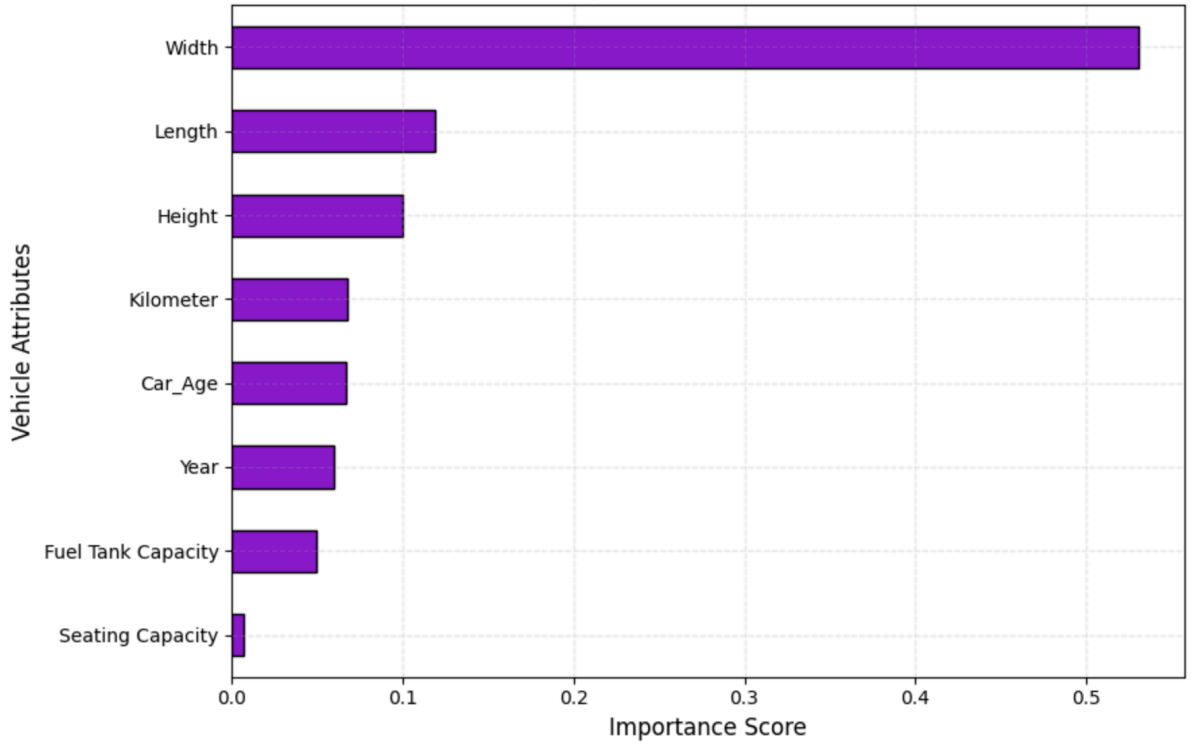
Table 1

Random Forest Evaluation Metrics

Metric	Value
RMSE	₹241,946
R^2	0.9488

Figure 1

Feature Importance: Random Forest Regressor



Note: See Appendix B for the code used in this section.

XGBoost Regressor: Feature Importance

The XGBoost model, known for its efficiency and performance, was trained similarly. It achieved an RMSE of ₹251,219 and an R-squared score of 0.9448 on the test set. Like Random Forest, the most important feature was again Width, confirming its strong correlation with price. Year, Height, and Fuel Tank Capacity were also significant (Ramana et al., 2024).

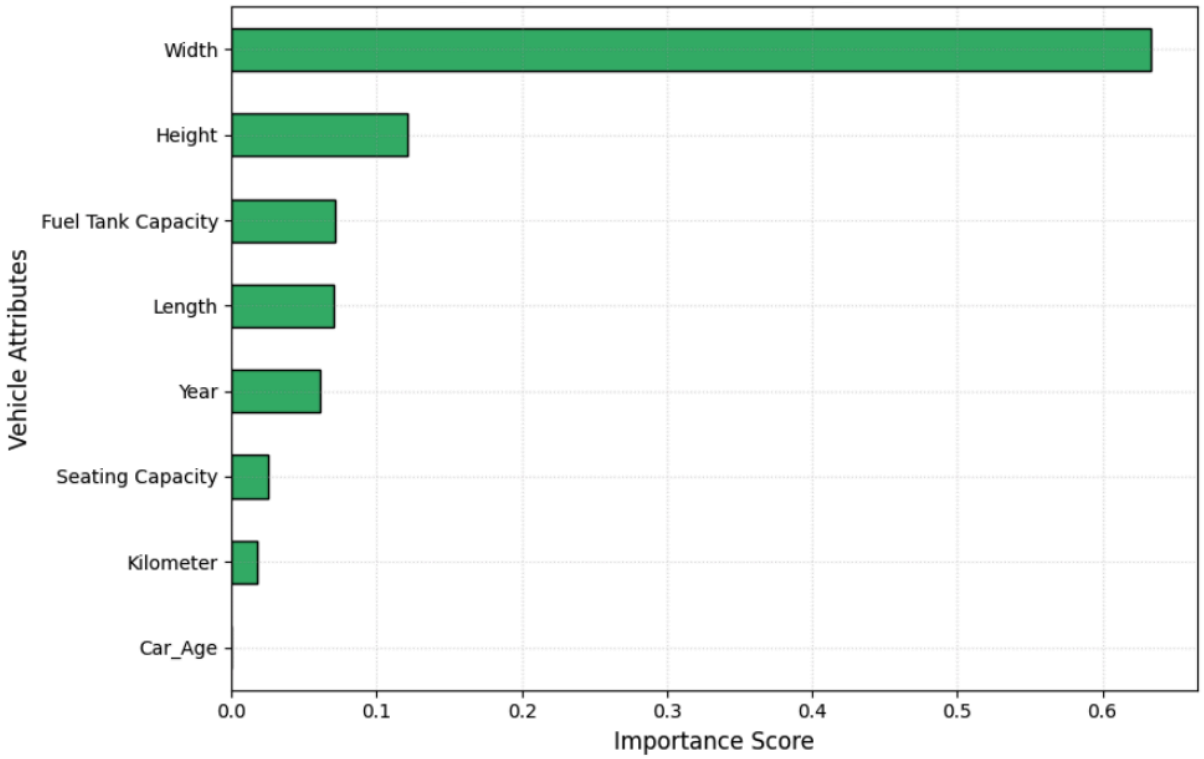
Table 2

XGBoost Evaluation Metrics

Metric	Value
RMSE	₹251,219
R^2	0.9448

Figure 2

Feature Importance: XGBoost Regressor



Note: See Appendix C for the code used in this section.

SHAP Summary Analysis

To explain the predictions, SHAP summary plots were created for both models. SHAP helps visualize how much each feature contributes to the prediction output.

- In the Random Forest model, Width was far ahead of other features in terms of impact.
- In the XGBoost model, Width remained dominant, with Year and Length also contributing significantly.

These results align with earlier regression analysis and provide a consistent interpretability layer (Iakubovskyi, 2022).

Table 3

Top Features by SHAP Value (Random Forest)

Feature	SHAP Value (avg impact)
Width	Highest
Length	High
Height	Moderate
Car_Age, Year	Moderate

Figure 3

SHAP Summary Plot: Random Forest

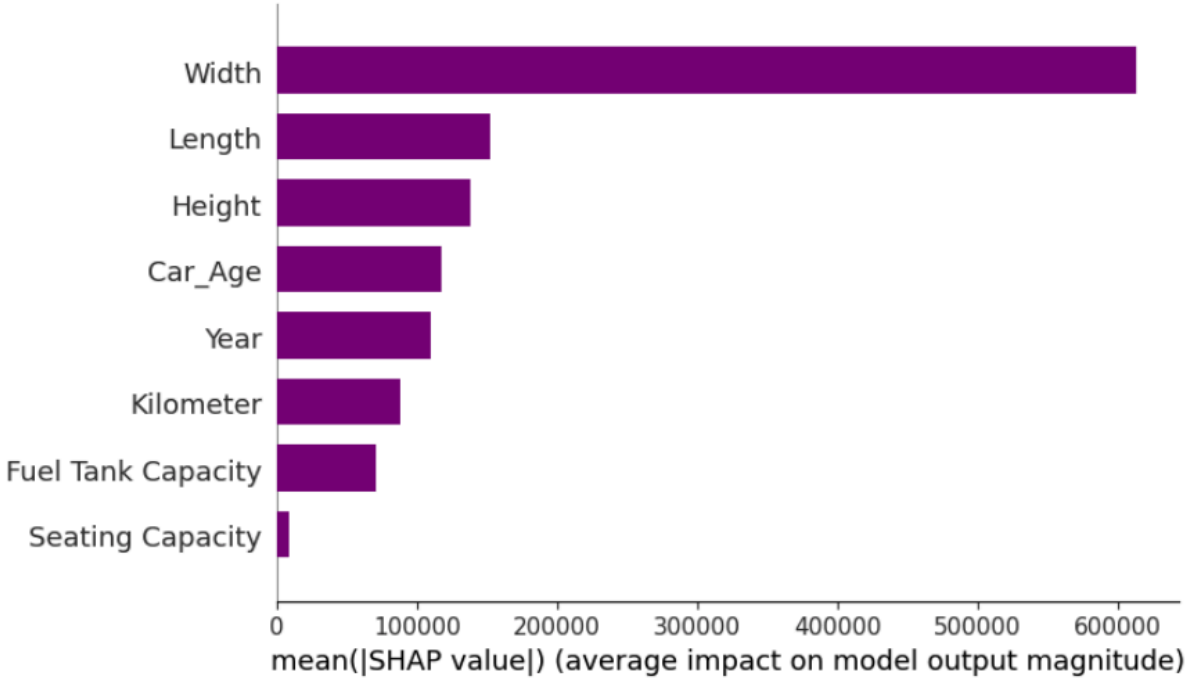


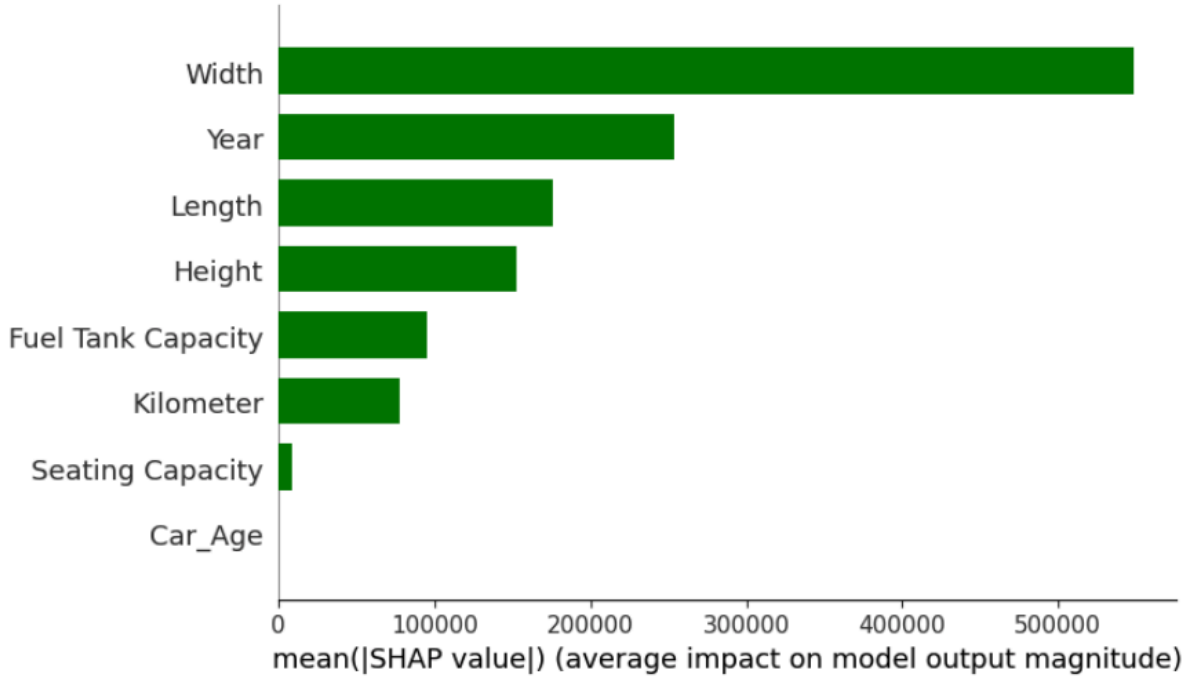
Table 4

Top Features by SHAP Value (XGBoost)

Feature	SHAP Value (avg impact)
Width	Highest
Year	High
Length	Moderate
Height, Fuel Tank	Moderate

Figure 4

SHAP Summary Plot: XGBoost



Note: See Appendix D for the code used in this section.

Partial Dependence Plot (PDP) for Width

To further investigate how Width influences predicted price, PDPs were generated for both models. Both plots show a sharp increase in predicted price beyond 1750 millimeters of width, implying that wider cars (typically larger vehicles or luxury models) are priced higher (Galli, 2024).

- The Random Forest PDP showed a sharp, stepped curve, with more noise and sensitivity.
- The XGBoost PDP was smoother but confirmed the same inflection zone after approximately 1750 millimeters.

Table 5

PDP Observations for Width

Model	Key Observation
Random Forest	Price rises steeply after 1750 mm width
XGBoost	Similar trend, more gradual and smooth

Figure 5

Partial Dependence Plot: Width (Random Forest)

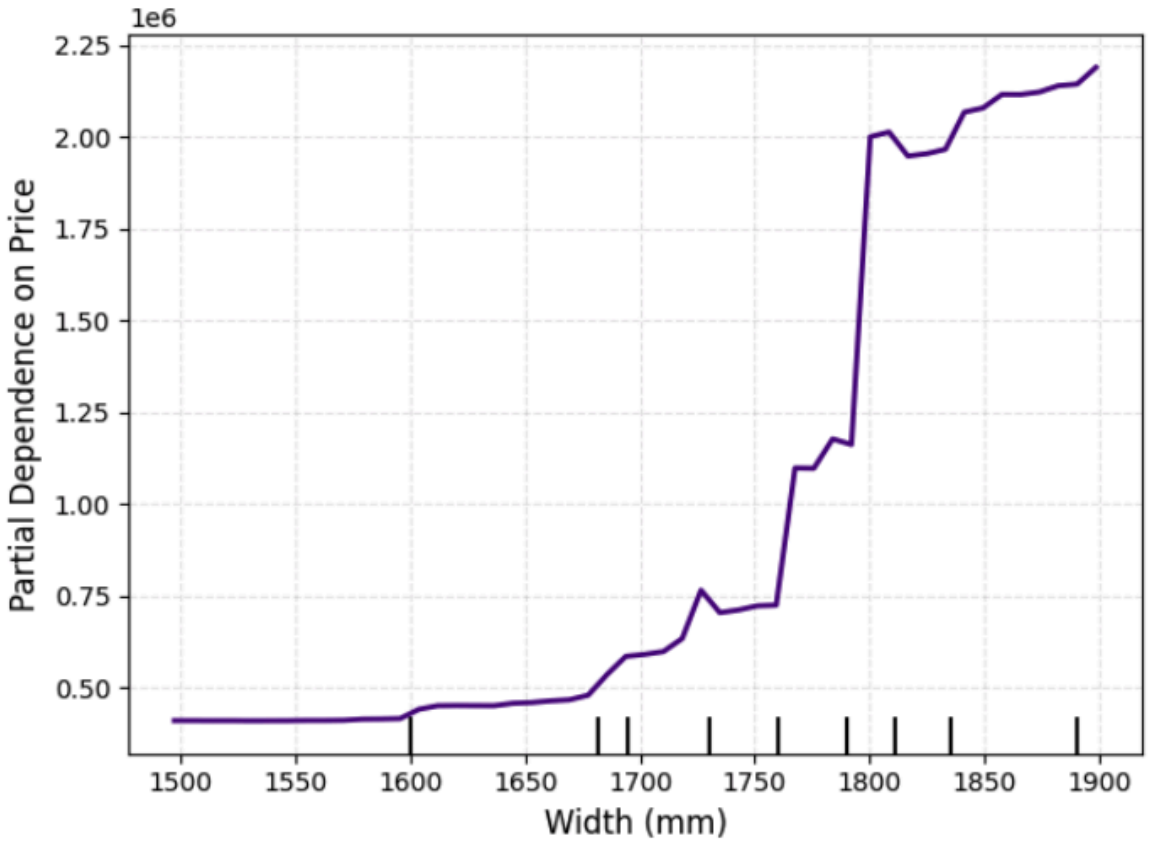
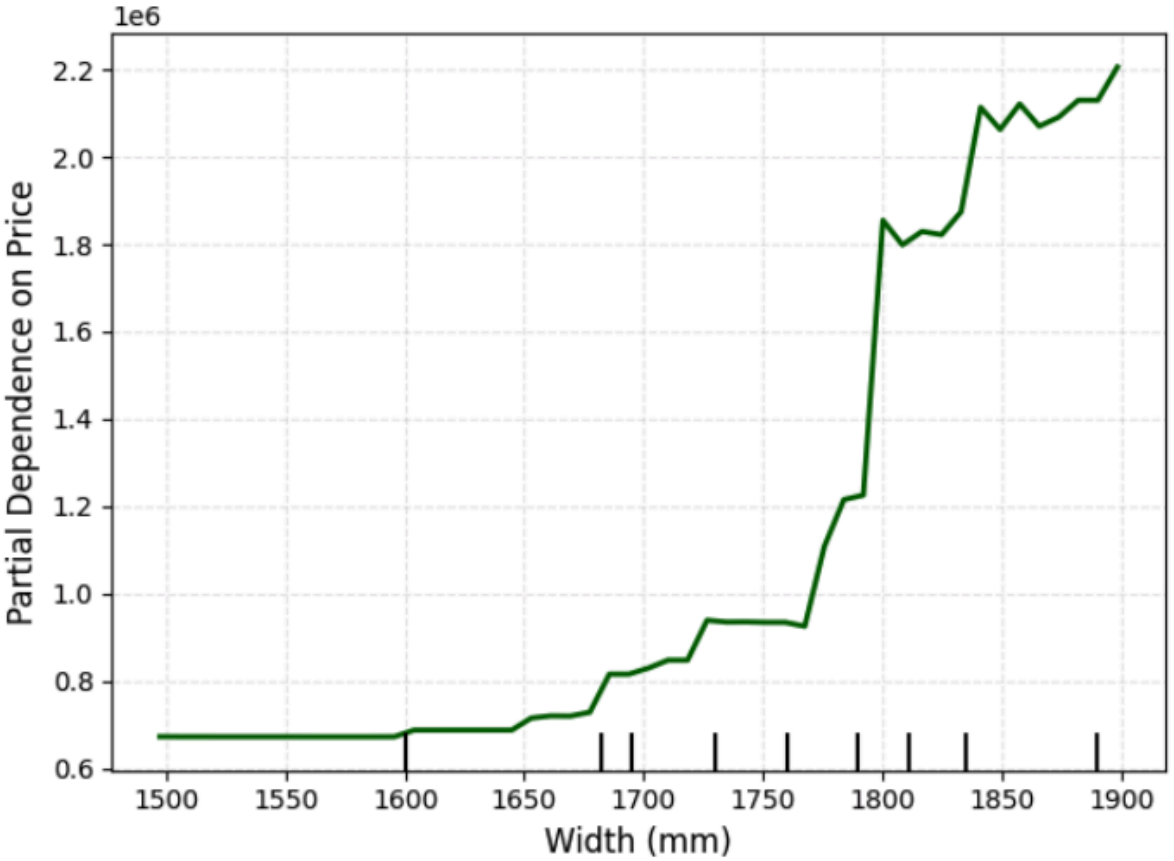


Figure 6

Partial Dependence Plot: Width (XGBoost)



Note: See Appendix E for the code used in this section.

Conclusion

Tree-based ensemble models offer highly accurate vehicle price predictions and robust feature interpretability. Both Random Forest and XGBoost confirmed that Width is the most important predictor, supporting findings from previous regression work.

With the addition of SHAP and PDP, the analysis gained transparent and intuitive explanations of feature behavior. These insights are invaluable in real world applications like pricing algorithms, dealership valuation tools, and car resale platforms.

The combination of predictive performance and visual storytelling through SHAP and PDP enhances both technical rigor and business communication, making this workflow ideal for decision-making.

References

Birla, N. (n.d.). *Vehicle dataset: Used cars data from websites*. Kaggle. Retrieved April 5, 2025, from https://www.kaggle.com/datasets/nehalbirla/vehicle-dataset-from-cardekho

Galli, S. (2024, January 15). *Partial dependence plots with Python: A comprehensive guide*. Train in Data Blog. Retrieved April 12, 2025, from https://www.blog.trainindata.com/partial-dependence-plots-with-python/

Iakubovskyi, D. (2022, September 24). *Price analysis of 100,000 new and used cars from 2022: SHAP values of car age, mileage, brand, and more*. Medium. Retrieved April 12, 2025, from https://medium.com/data-and-beyond/price-analysis-of-100-000-new-and-used-cars-from-2022-effects-of-car-age-mileage-brand-and-7b798960c986

Panpatil, A. (2025, March 1). *Car price prediction using random forest*. LinkedIn. Retrieved April 12, 2025, from https://www.linkedin.com/pulse/car-price-prediction-using-random-forest-ankitaa-panpatil-jrogf/

Ramana, B. V., Govardhini, A., Tejaswini, A. D., Reddy, B. Y. K., Aditya, N. V., & Aravind, A. (2024). Car price prediction using random forest algorithm. *Journal of Nonlinear Analysis and Optimization, 15*(1), 1639–1642. Retrieved April 12, 2025, from https://jnao-nu.com/Vol.%2015%2C%20Issue.%2001%2C%20January-June%20%3A%202024/207_online.pdf

Shafi, A. (2024, October 1). *Random forest classification with Scikit-Learn*. DataCamp. Retrieved April 12, 2025, from https://www.datacamp.com/tutorial/random-forests-classifier-python

Tuychiev, B. (2023, February 22). *Using XGBoost in Python tutorial*. DataCamp. Retrieved April 12, 2025, from https://www.datacamp.com/tutorial/xgboost-in-python

Appendix A

```
#Prepare Car Data for ML

#Only numeric attributes matter for tree models
numeric_car_specs = car_data_raw.select_dtypes(include='number')

#Removing records with incomplete values
car_data_ready = numeric_car_specs.dropna()

#Target Price, Features, everything
vehicle_features = car_data_ready.drop(columns=['Price'])
price_target = car_data_ready['Price']

#Split data to create testing environment
from sklearn.model_selection import train_test_split

carX_train, carX_test, price_train, price_test = train_test_split(
    vehicle_features, price_target, test_size=0.2, random_state=42
)

print("Training features shape:", carX_train.shape)
print("Testing features shape :", carX_test.shape)
```

Appendix B

```
#Train Random Forest Regressor
from sklearn.ensemble import RandomForestRegressor
from sklearn.metrics import mean_squared_error, r2_score
import numpy as np
import matplotlib.pyplot as plt

forest_of_prices = RandomForestRegressor(n_estimators=100, random_state=42)
forest_of_prices.fit(carX_train, price_train)

#Predict
price_predictions_forest = forest_of_prices.predict(carX_test)

#Evaluation
forest_rmse = np.sqrt(mean_squared_error(price_test, price_predictions_forest))
forest_r2 = r2_score(price_test, price_predictions_forest)

print("Random Forest Results")
print(f"RMSE: ₹{int(forest_rmse):,}")
print(f"R² Score: {forest_r2:.4f}")
#Feature Importance Visualization
feature_importance = pd.Series(forest_of_prices.feature_importances_, index=vehicle_features.columns)
feature_importance_sorted = feature_importance.sort_values(ascending=True)
#Plot
plt.figure(figsize=(9,6))
feature_importance_sorted.plot(kind='barh', color='darkviolet', edgecolor='black')
plt.title("Feature Importance: Random Forest ", fontsize=15, color='darkblue', weight='bold')
plt.xlabel("Importance Score", fontsize=12)
plt.ylabel("Vehicle Attributes", fontsize=12)
plt.grid(True, linestyle='--', alpha=0.3)
plt.tight_layout()
plt.show()
```

Appendix C

```
#Train XGBoost Regressor

from xgboost import XGBRegressor

#boosting model
xboost_model = XGBRegressor(n_estimators=100, random_state=42)
xboost_model.fit(carX_train, price_train)

#Predict using test data
price_predictions_boost = xboost_model.predict(carX_test)

#Evaluate model
boost_rmse = np.sqrt(mean_squared_error(price_test, price_predictions_boost))
boost_r2 = r2_score(price_test, price_predictions_boost)

print("XGBoost Model Results")
print(f"RMSE: ₹{int(boost_rmse):,}")
print(f"R² Score: {boost_r2:.4f}")

#Feature importances for XGBoost
boost_importance = pd.Series(xboost_model.feature_importances_, index=vehicle_features.columns)
boost_sorted = boost_importance.sort_values(ascending=True)

#Visualize Boosting impact
plt.figure(figsize=(9,6))
boost_sorted.plot(kind='barh', color='mediumseagreen', edgecolor='black')
plt.title("Boosting Impact: Feature Strength via XGBoost", fontsize=15, color='darkgreen', weight='bold')
plt.xlabel("Importance Score", fontsize=12)
plt.ylabel("Vehicle Attributes", fontsize=12)
plt.grid(True, linestyle=':', alpha=0.4)
plt.tight_layout()
plt.show()
```

Appendix D

```
#SHAP Insights
import shap
import matplotlib.pyplot as plt
shap.initjs()

#Random Forest
shap_engine_forest = shap.TreeExplainer(forest_of_prices)
forest_shap_values = shap_engine_forest.shap_values(carX_test)

#SHAP Plot Random Forest
print("SHAP Breakdown: Random Forest")
shap.summary_plot(
   forest_shap_values, carX_test,
   plot_type="bar",
   color="purple"
)
#XGBoost
shap_engine_boost = shap.TreeExplainer(xboost_model)
boost_shap_values = shap_engine_boost.shap_values(carX_test)
```

```
#SHAP Plot XGBoost
print("SHAP Breakdown: XGBoost")
shap.summary_plot(
    boost_shap_values, carX_test,
    plot_type="bar",
    color="green"
)
```

Appendix E

```
#Partial Dependence Plot (PDP)

from sklearn.inspection import PartialDependenceDisplay

#PDP for Random Forest
print("PDP: Random Forest – How Width influences Price")
PartialDependenceDisplay.from_estimator(
    forest_of_prices, carX_test, ['Width'],
    kind='average', grid_resolution=50,
    line_kw={'color': 'indigo', 'linewidth': 2},
)
plt.xlabel("Width (mm)", fontsize=12)
plt.ylabel("Partial Dependence on Price", fontsize=12)
plt.grid(True, linestyle='--', alpha=0.3)
plt.tight_layout()
plt.show()

#PDP for XGBoost
print("PDP: XGBoost – How Width influences Price")
PartialDependenceDisplay.from_estimator(
    xboost_model, carX_test, ['Width'],
    kind='average', grid_resolution=50,
    line_kw={'color': 'darkgreen', 'linewidth': 2},
)
plt.xlabel("Width (mm)", fontsize=12)
plt.ylabel("Partial Dependence on Price", fontsize=12)
plt.grid(True, linestyle='--', alpha=0.3)
plt.tight_layout()
plt.show()
```