**REPORT(Jiayi\_QIAN** **87297, Shujuan\_ZHU87313)**

1. **Introduction**

The goal of this project is to develop a machine learning model that predicts car prices based on user input. We utilized a dataset from Kaggle, performed data exploration and preprocessing, trained multiple machine learning models, and deployed an API using FastAPI.

1. **Performance Metric**

To evaluate the predictive power of our model, we chose Root Mean Squared Error (RMSE) and R² Score as the performance metrics:

* RMSE (Root Mean Squared Error): Measures the average error magnitude between the predicted and actual car prices. A lower RMSE indicates better performance.
* R² Score: Represents the proportion of variance explained by the model. A higher value (closer to 1) indicates a better fit.

These metrics were chosen because car prices have a continuous distribution, and minimizing the RMSE helps improve prediction accuracy.

1. **Performance of algorithm**

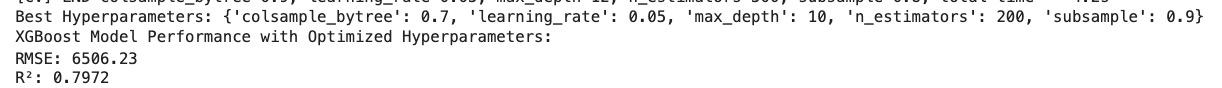
A screenshot of a computer code

Description automatically generatedA screenshot of a computer program

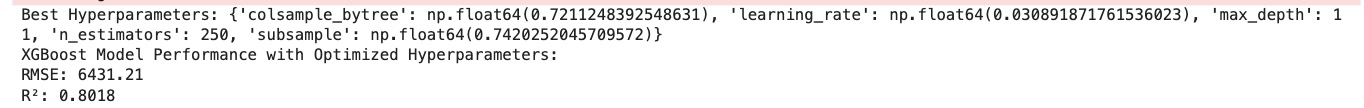
Description automatically generated

* Baseline Model: RMSE: 11218.57 R²: 0.3970 The baseline model is a simple model (linear regression). RMSE of 11218.57 means the model’s predictions deviate from the actual values by about 11218 units on average. R² of 0.3970 suggests that the model can explain about 39.7% of the variance in the target variable (Price). This is a relatively low R², indicating that the model is not doing a very good job of explaining the target variable's variation, which is typical for a simple baseline model.
* Random Forest Model: RMSE: 7476.36 R²: 0.7322 The Random Forest model, a more complex ensemble method, performs better: RMSE of 7476.36: The average deviation between predicted and actual values is much lower than the baseline model, meaning the model's predictions are significantly closer to the true values. R² of 0.7322: This model explains about 73.22% of the variance in the target variable, which is a substantial improvement over the baseline model. This suggests that Random Forest is capturing much more of the underlying relationships in the data.
* XGBoost Model: RMSE: 6753.39 R²: 0.7815 The XGBoost model is performing better than Random Forest, but only slightly: RMSE of 6753.39: The deviation between predicted and actual values is lower than both the baseline and Random Forest models, indicating that XGBoost is providing the most accurate predictions of all the models tested. R² of 0.7815: This model explains 78.15% of the variance in the target variable, which is the highest R² among the three models. XGBoost has clearly learned more from the data and has a stronger predictive power compared to both the baseline and Random Forest models.

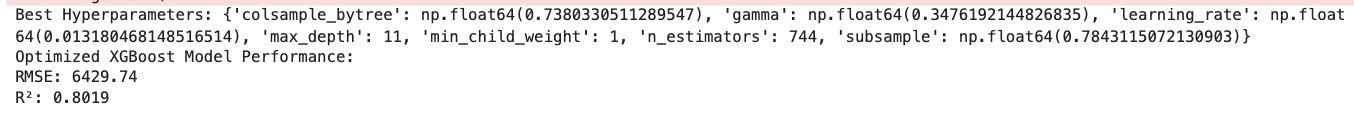
1. **Hyperparameter Optimization Rounds**



* First round of tuning：Through GridSearchCV, you optimized parameters such as n\_estimators, max\_depth, and learning\_rate in a smaller search space, which improved the performance of the model.



* Second round of tuning:A wider search was performed using RandomizedSearchCV, resulting in a lower learning\_rate, an additional layer of max\_depth, and further tuning of subsample and colsample\_bytree to improve generalization.



* Final optimization results: This search was more refined, introducing gamma and min\_child\_weight, and significantly improving n\_estimators, ultimately achieving the lowest RMSE (6429.74) and highest R² (0.8019), indicating that the model has better fitting ability and generalization performance.

1. **API Part:**

Our FastAPI-based prediction API enables users to input car features in JSON format and receive an estimated price. When a request is sent to the /predict/ endpoint, we first preprocess the input by handling missing values, converting categorical features using **one-hot encoding**, and scaling numerical variables with **StandardScaler**, ensuring consistency with our trained model. The processed data is then passed to our **XGBoost model (**best\_xgb\_model.pkl**)**, which predicts the car price based on the given attributes. Finally, we return the predicted price in JSON format. This API provides real-time predictions, aligning with our data preprocessing pipeline to ensure accuracy and reliability in deployment

1. **The link of the different resources of the project (notebook, github, url of the API on render).**

Github：

<https://github.com/shujuan12/Data_science>

1. **Conclusion of the project.**