**Introduction:**

The used car market is a dynamic and essential sector, influencing both individual consumers and the broader economy. For my final project, I aim to develop a predictive model capable of estimating used car prices by leveraging two main datasets: a “micro-level” dataset from Kaggle (<https://www.kaggle.com/datasets/austinreese/craigslist-carstrucks-data>) and a “macro-level” dataset derived from my midterm project (<https://github.com/sg7667/DBFinal>). The micro-level dataset includes variables such as price, year, manufacturer, model, and odometer, while the macroeconomic dataset incorporates variables such as consumer sentiment, unemployment, and auto inventory. These datasets will be used to analyze and predict the factors driving used car prices, potentially providing insights for buyers and sellers of the used car market. The predictive model will enable stakeholders to make informed decisions, optimize pricing strategies, and better understand the interplay between micro-level and macro-level factors in the used car market. Successful development of the model will hopefully enhance predictive capabilities for used car prices.

**MICRO-LEVEL DATASET**

**Data Pre-Processing & Preliminary Examination**

The data compromises of 'url', 'region', 'region\_url', 'price', 'year', 'manufacturer', 'model', 'condition', 'cylinders', 'fuel', 'odometer', 'title\_status', 'transmission', 'VIN', 'drive', 'size' , 'type', 'paint\_color', 'image\_url', 'description', 'county', 'state', 'lat', 'long', 'posting\_date'.

First, I selected relevant columns for car price prediction after checking the number of data (there were some with 0 data):

A screenshot of a computer program

Description automatically generatedA screenshot of a computer

Description automatically generated

Second, I identified columns with missing values:

A computer screen shot of a missing values

Description automatically generated A screenshot of a graph

Description automatically generated

I decided that since the original dataset is large, the missing data will not be replaced but removed before proceeding with the analysis.

**Categorical Data Encoding**

I used LabelEncoder to process categorical data, and final data looks like as it is below:

A screenshot of a computer screen

Description automatically generated A screenshot of a computer

Description automatically generated

**Exploratory Data Anlysis**

To begin with EDA, I have visualized the correlation among the variables with used car price using the correlation heatmap.

**A screenshot of a graph

Description automatically generated**

Among these,

Price and Year: There is a positive correlation between price and year. This suggests that newer manufacturing years are more likely to have higher prices.

Price and Odometer: There is a negative correlation between price and odometer. This indicates that higher mileage is likely associated with lower prices.

Price and Cylinders: There is a positive correlation between price and cylinders. This indicates that more cylinders are likely associated with higher prices.

Other Variables (e.g., Manufacturer, type, color): The correlations with price are relatively weak for these variables.

To understand the correlations better, I have visualized the relationships using scatter plot, bar plot, and box plot.

A graph of blue dots

Description automatically generated

As mileage increases, the price tends to decrease.

A graph of blue dots

Description automatically generated

The newer the vehicles, the higher the prices. While there are some outliers (older vehicles with abnormally high prices), most data points follow a positive correlation between year and price.

A graph showing a bar graph

Description automatically generated

As the number of transmissions increases, the price increases. Higher transmission speeds often improve performance and efficiency, which involve advanced technology and manufacturing costs, making the vehicle more expensive.

A graph of a bar graph

Description automatically generated

As the number of cylinders increases, the price increases. 0-1 cylinders, however, do not follow the pattern - could imply a niche market. Vehicles with more cylinders generally provide greater power and performance, and they are more expensive due to higher production costs and fuel consumption.

A graph of blue and black bars

Description automatically generated

High-Priced Vehicles: Porsche, Mercedes-Benz, Lexus, BMW, etc., generally exhibit higher price ranges.

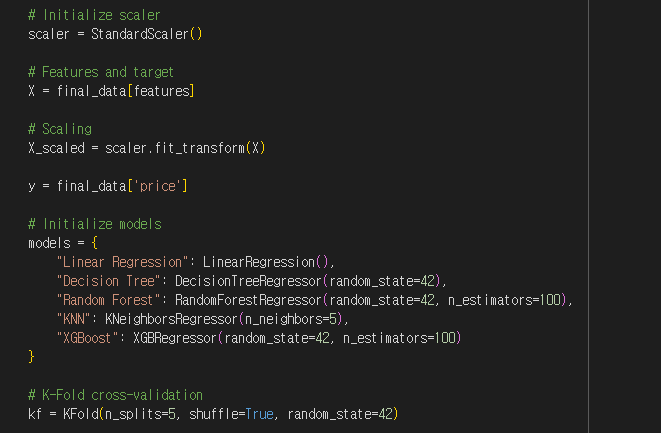
Low-Priced Vehicles: Kia, Hyundai, Ford, etc., are positioned in relatively lower price ranges.

Mid-Priced Vehicles: Toyota, Honda, Nissan, etc., show a middle price distribution.

**Modeling & Interpretations**

The code below implements a comparison of various machine learning models to predict prices. The primary goal was to evaluate the models using key metrics, including Mean Squared Error (MSE), Root Mean Squared Error (RMSE), and R^2 across multiple cross-validation folds.

First, the dataset is split into features (X) and target (y), with the target being the price. Features are scaled using StandardScaler to normalize them.



A 5-fold cross-validation is employed to split the data into five subsets, ensuring that each subset is used as a test set once while the rest are used for training. This helps evaluate the models' generalization abilities.

The used car price prediction is a regression problem that requires models capable of outputting continuous values. Among the various regression models, the following five models were selected for the task:

1. Linear Regression

Linear Regression is a simple, intuitive model that serves as a baseline for evaluating linear relationships in the data. It allows for quick training and straightforward interpretation of variable importance.

2. Decision Tree

The Decision Tree model captures basic nonlinear relationships and serves as a simple, interpretable alternative. It splits data into hierarchical rules, making it easy to understand how features impact used car prices. This model acts as a bridge between Linear Regression and more advanced methods like Random Forest.

3. Random Forest

Random Forest is a robust model that captures nonlinear relationships and interactions between variables. It prevents overfitting and calculates variable importance, identifying key drivers of used car prices. This model tests for nonlinear patterns and compares them to Linear Regression results.

4. K-Nearest Neighbors (KNN)

KNN is a distance-based model that predicts prices using local patterns in the data. It performs well with properly scaled data, capturing relationships based on proximity to similar instances.

5. XGBoost

XGBoost is a high-performance boosting model capable of learning intricate data relationships and reducing prediction errors. It offers advanced optimization and regularization features, making it ideal for maximizing accuracy.

A screenshot of a computer program

Description automatically generated

The evaluate\_model function calculates:

* MSE: Measures the average squared difference between predicted and actual values. Lower values indicate better accuracy.
* RMSE: Square root of MSE, providing an error measure in the same units as the target variable.
* R^2: Indicates the proportion of variance in the target variable explained by the model. Values closer to 1 imply better performance.

A screenshot of a computer screen

Description automatically generated

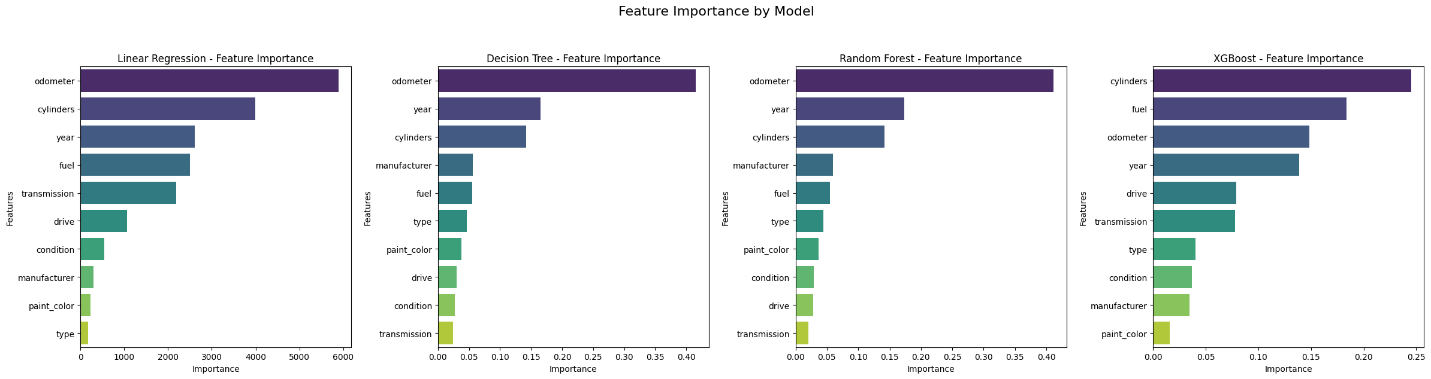
The results indicate significant differences in model performance. Linear Regression has the highest error (MSE: 95,421,006.87, RMSE: 9,766.46) and the lowest R2 (0.4756), showing that it struggles to capture the underlying relationships in the data, which are likely not purely linear.

Decision Tree performs better, with lower errors (MSE: 58,070,188.72, RMSE: 7,618.41) and a higher R2 (0.6807), indicating its ability to handle nonlinear patterns, though it still lacks robustness.

**Random Forest emerges as the best model**, achieving the lowest errors (MSE: 32,198,039.91, RMSE: 5,671.18) and the highest R2 (0.8230), demonstrating its effectiveness in capturing complex nonlinear relationships and interactions.

**XGBoost also performs well**, with competitive error metrics (MSE: 41,653,722.95, RMSE: 6,451.95) and a strong R2 (0.7710), making it a solid alternative to Random Forest, particularly for datasets with high complexity.

KNN shows intermediate performance, with errors (MSE: 48,147,045.34, RMSE: 6,936.75) and R2 (0.7354) slightly worse than Random Forest and XGBoost, but better than Linear Regression and Decision Tree.



The feature importance analysis reveals that odometer is the most influential predictor in most models, including Linear Regression, Decision Tree, and Random Forest, highlighting its strong relationship with used car prices. In contrast, XGBoost identifies cylinders as the most important feature, followed by fuel and odometer. Across all models, year consistently shows moderate importance, reflecting its relevance in determining vehicle value. Less impactful features, such as paint\_color, type, and transmission, contribute minimally to the predictions, indicating limited influence on price variation. The differences in feature rankings, particularly in XGBoost, underline the varying ways models prioritize relationships and interactions in the data.

Overall, odometer, cylinders, year, and fuel emerge as critical factors in predicting used car prices, with models like XGBoost and Random Forest providing nuanced insights into feature interactions.

**Grid Search**

Further, GridSearchCV is employed to optimize the hyperparameters for two machine learning models: Random Forest and XGBoost, aimed at minimizing the mean squared error (MSE) through a 3-fold cross-validation.

For Random Forest, the grid search explores combinations of hyperparameters such as the number of estimators (n\_estimators), maximum tree depth (max\_depth), and minimum samples required for splitting (min\_samples\_split) or as leaf nodes (min\_samples\_leaf). Similarly, for XGBoost, the grid search tests values for the number of estimators, tree depth, learning rate, and subsample ratio.

After fitting the grid search, the code extracts the best hyperparameter combinations and their corresponding RMSE values for each model, storing the results in a dictionary (optimization\_results) for easy comparison.

A screenshot of a computer

Description automatically generated

For XGBoost, the "Best RMSE" after grid search (7135.56) is worse than the pre-optimization mean RMSE (6451.95).

For Random forest, A screenshot of a computer

Description automatically generated

the "Best RMSE" after grid search (7333.84) is worse than the pre-optimization mean RMSE (6451.95).

These were unexpected outcomes, but differences in results between the two codes can be attributed to a few factors. First, I used 5-fold cross-validation in the first code and 3-fold cross-validation in the second code, which could introduce slight variance in evaluation due to the different validation splits. Second, in this case, the initial parameter (n\_estimators=100) already performed well, so the optimized hyperparameters may not show significant improvement. As a result, the performance advantage could fluctuate depending on the training run. Third, there could have been potential overfitting.

**MACRO-LEVEL DATASET**

This dataset incorporates all the data analyzed in the midterm project. However, the model now includes a new dataset on auto inventory.

Dataframe looks like below:

A screenshot of a graph

Description automatically generated

**Exploratory Data Anlysis**

Most work on EDA has already been done in the midterm project, however, this final project gives a quick EDA as well.

A diagram of a graph

Description automatically generated with medium confidence

Used car prices are negatively influenced by inventory levels, consumer sentiment, and unemployment rate, with inventory having the strongest impact.

A graph of a graph with blue dots

Description automatically generated

The analysis shows a negative correlation (-0.362) between unemployment rates and used car prices, with prices generally decreasing as unemployment rises. At low unemployment rates (3-4%), prices are highly volatile, ranging from 240 to 360, and some outliers suggest unique market conditions. As unemployment exceeds 7%, price volatility decreases, stabilizing between 220 and 280. Most data clusters in the 4-7% range, indicating a complex, non-linear relationship between unemployment and used car prices.

A graph of a used car prices

Description automatically generated

The analysis shows a negative correlation (-0.344) between consumer sentiment and used car prices, with prices generally decreasing as sentiment improves. At low sentiment levels (50-70), higher prices (300-360) are observed with significant volatility, while at higher sentiment levels (80-110), prices stabilize within a narrower range (220-280). High consumer sentiment could shift consumers’ demand towards new cars rather than used cars, explaining the relationship.

A graph of sales and inventory

Description automatically generated

The analysis reveals a strong negative correlation (-0.564) between inventory levels and used car prices, where prices decrease sharply as inventory rises. At low inventory levels (200-400), prices are high (300-360) and volatile, while at higher inventory levels (800-1600), prices stabilize between 220-280. The relationship is non-linear: prices drop steeply when inventory is below 400 but decline more gradually beyond 800. This pattern highlights the direct impact of supply on prices, where low inventory drives prices up due to supply shortages.

**Modeling**

I have used similar models to micro-level data: Linear Regression, Random Forest, and XGBoost models. After scaling the data and splitting it into training and test sets, the models are trained, and performance is measured using R2, MSE, and cross-validation scores.

**A screenshot of a computer

Description automatically generated**

A comparison of blue bars

Description automatically generated with medium confidence

XGBoost demonstrated the best performance with an R2 of 0.7781 (78% explanatory power) and the lowest MSE of 157.13, along with stable cross-validation results (R2 = 0.7211 ± 0.1166). Random Forest also performed well, achieving an R2 of 0.7589 and an MSE of 170.76, with slightly higher variability in cross-validation (R2 = 0.7405 ± 0.1695). In contrast, Linear Regression showed significantly lower performance, with an R2 of 0.4665, the highest MSE of 377.78, and unstable cross-validation results (R2 = 0.2087 ± 0.4948). Overall, XGBoost proved to be the most accurate and reliable model for predicting used car prices.

A close-up of a graph

Description automatically generated

Across all models, Inventory emerged as the most critical feature for predicting used car prices (Random Forest: 88.12%, XGBoost: 89.62%), followed by Unemployment Rate (RF: 6.85%, XGB: 5.43%), while Consumer Sentiment had the least impact (RF: 5.03%, XGB: 4.95%). These results suggest that the relationship between used car prices and the features is non-linear, as evidenced by the superior performance of non-linear models (XGBoost and Random Forest) compared to Linear Regression. Inventory's dominance highlights the significant influence of supply-side factors on pricing. The high cross-validation variance in Linear Regression (±0.4948) further indicates its inability to capture the complex patterns in the data effectively.

**Grid Search**

Lastly, a hyperparameter grid is defined for XGBoost, including parameters like n\_estimators, max\_depth, learning\_rate, and subsample. Using GridSearchCV with 3-fold cross-validation and the negative mean squared error scoring metric, the best hyperparameters are identified, and the corresponding RMSE is calculated.

A screenshot of a computer program

Description automatically generated

A screenshot of a computer program

Description automatically generated

The results demonstrate that XGBoost, after hyperparameter tuning with GridSearchCV, outperformed all other models for predicting used car prices. The optimized XGBoost model achieved an R2 score of 0.7939 and an MSE of 145.9565 on the test set, showing improvement over its pre-tuning R2 score of 0.7781 and MSE of 157.1315.

**Conclusion**

The results of the analysis highlight that non-linear models, particularly Random Forest and XGBoost, outperform simpler models like Linear Regression when predicting used car prices. For the micro-level dataset, Random Forest achieved the best overall performance with an R2 score of 0.8230 and the lowest RMSE of 5,671.18, effectively capturing complex relationships in the data. XGBoost followed closely, while Linear Regression performed the worst. For the macro-level dataset, XGBoost again demonstrated superior performance with an R2 score of 0.7939 and an MSE of 145.96 after hyperparameter tuning, improving upon its pre-optimization results. Random Forest also performed well, with an R2 of 0.7589 and an MSE of 170.76, while Linear Regression significantly underperformed.

Key Findings

1. Performance of Non-Linear Models:  
   Random Forest and XGBoost demonstrated superior predictive power, effectively handling the non-linear relationships within the used car pricing data.
2. Dominant Features:  
   For micro-level, odometer, cylinder, fuel, and year seemed to have the most significant influence on used car prices, odometer seeming to have the biggest influence. For macro-level, inventory seemed to have the highest influence.
3. Comparison between micro-level and macro-level models

The comparison might be difficult because 1) the two datasets have different number of total data (micro-level has larger dataset) 2) the two datasets are based on different time period 3) the target variable (used car prices) is different in that one is about used car prices of particular “car models,” whereas the other is the average used car prices across the economy.

In any case, for **some** **comparison**,

The micro-level data and models achieved a higher R2 score, indicating a better ability to explain the variance in used car prices. However, the macro-level models had lower RMSE values, meaning they provided more precise predictions in terms of error magnitude.

In short, findings confirm that non-linear models, particularly XGBoost and Random Forest, are far more effective for predicting used car prices due to their ability to capture the complex interplay between features. The micro-level models achieved a higher R², explaining variance better, while the macro-level models had lower RMSE, providing more precise predictions, though differences in dataset size, time period, and target variables make direct comparison challenging.

**Further Improvements & Next Steps**

Data Improvements:

* Data Augmentation and Collection:
  + Expand Data Range: Integrate external datasets like weather data, used car demand trends, and regional economic indicators.
  + Increase Sample Size: Include recent market data and data from diverse regions to improve model robustness.
* Feature Engineering:
  + Create Derived Variables:
    - Generate interactions between features (e.g., Unemployment Rate \* Consumer Sentiment).
  + Incorporate Time Features:
    - Include variables reflecting seasonality and cyclicality (e.g., year, month, season).

Model Improvements:

* Enhance Model Performance:
  + Ensemble Methods: Combine predictions from multiple models (e.g., Weighted Averaging, Stacking) for improved accuracy.
* Optimize Specific Models:
  + Improve hyperparameter optimization
  + Random Forest: Remove unimportant variables to improve efficiency.
  + XGBoost: Apply feature interaction constraints to eliminate meaningless interactions.
* Add Time Series Models:
  + Introduce models like LSTM or Prophet to capture seasonality and cyclical trends in used car prices.
* Leverage Deep Learning:
  + For larger datasets, use deep learning-based regression models to capture complex, non-linear relationships.