Iteration 3

Xiaoyang Fei, Jack Carpini, Maalolan Bharaniraj

March 11, 2025

1 Flowchart

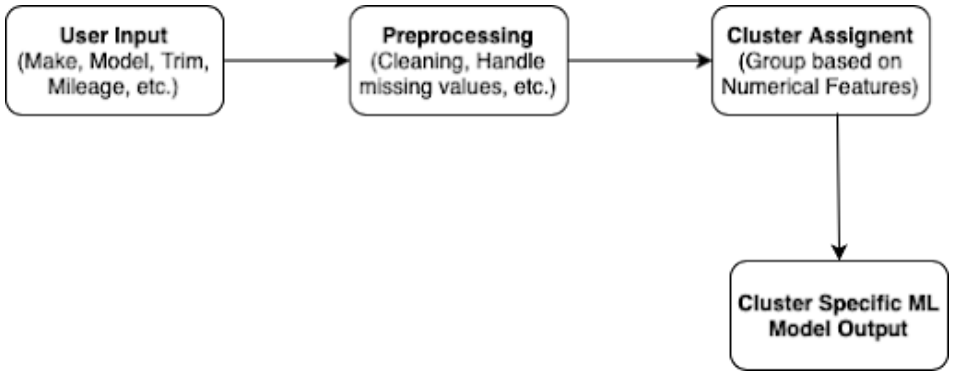

Figure 1: Flowchart

2 Purpose of Methodology

We aim to predict fair car prices by adopting a hybrid approach that combines clustering and regression techniques. By segmenting car bargains into clusters and training individualized models for each cluster, we can capture different market trends, improving prediction accuracy. This method helps identify overpriced or underpriced cars, informing purchasing decisions.

3 Problem Statement

3.1 Regression Task

Predict used car prices based on historical transaction data.

3.2 Challenges

Car pricing varies significantly due to factors such as location, mileage, and brand-specific depreciation. A "one-size-fits-all" model fails to capture the nuances across different car segments.

3.3 Solution

- **Clustering First:** Group cars based on numerical attributes (e.g., mileage, age, condition) to create distinct market segments.
- **Cluster-Specific Models:** Train tailored models (e.g., LightGBM, Neural Networks) for each cluster to enhance predictive accuracy.
- **Impact:** This method empowers buyers with data-driven price estimates, reducing reliance on biased or incomplete listings.

4 Data Collection and Preparation

4.1 Datasets

Both datasets were sourced from Kaggle:

- **vehicle-sales-data** 558,837 rows, reduced to 12 features.
- **us-used-car-sales-data** 122,144 rows, reduced to 8 features.

4.2 Original Features

- **Dataset 1:** year, make, model, trim, body, transmission, vin, state, condition, odometer, color, interior, seller, mmr, selling_price, sale_date.
- **Dataset 2:** ID, price_sold, year_sold, zipcode, mileage, make, model, year, trim, engine, body_type, num_cylinders, drive_type.

4.3 Merged Features

- year, make, model, trim, state, condition, odometer, mmr, selling_price, zipcode, mileage, price_sold.

4.4 Data Cleaning

- Merged both datasets to increase the total number of records.
- Records missing make, model, or trim were removed, as these were too difficult to impute.
- Filled missing numerical values with the median and categorical values with the mode.
- The numerical columns exhibited a heavy right skew, which aligns with expectations.

5 Selection of Machine Learning Models

We selected a variety of machine learning models, including:

- **Traditional Regression Models:** Lasso regression, Ridge regression, and LightGBM.
- **Deep Learning:** Fully connected feedforward neural networks.

Support Vector Machine (SVM) was considered but was dropped due to computational constraints. Hyperparameter tuning is ongoing, with RMSE used as the evaluation metric.

6 Model Development and Training

6.1 Clustering

- KMeans was used to segment the data set into three clusters.
- The data set was divided into 80% training and 20% testing within each group.
- Separate models were trained on each cluster to determine the best predictive approach.

6.2 Hyperparameter Tuning

- LightGBM: Grid search optimization for `num_leaves` and `learning_rate`.
- Additional hyperparameter tuning is in progress.

7 Evaluation and Comparison

7.1 Raw Data Results

The information was initially grouped into five groups using KMeans, but two of the clusters were deleted as outliers and were not used for comparisons. Various regression models were subsequently fitted to each of the remaining clusters, including Lasso regression, Ridge regression, and LightGBM. The models were compared on RMSE, and no hyperparameter optimization was performed at this stage. PCA visualizations were made to obtain improved representations of cluster distributions.

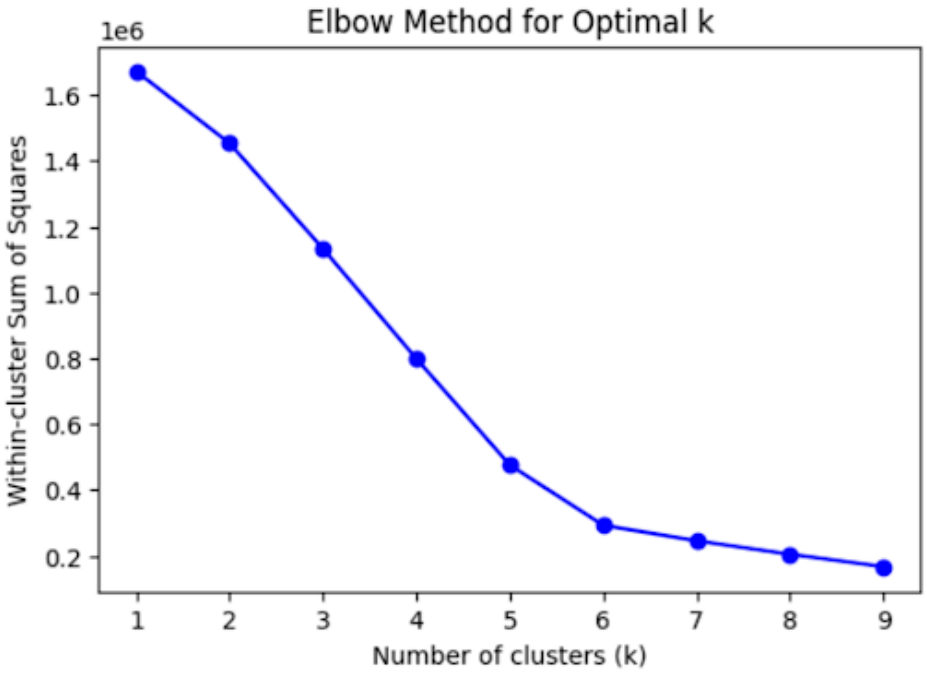

Figure 2: Elbow Method for Optimal k

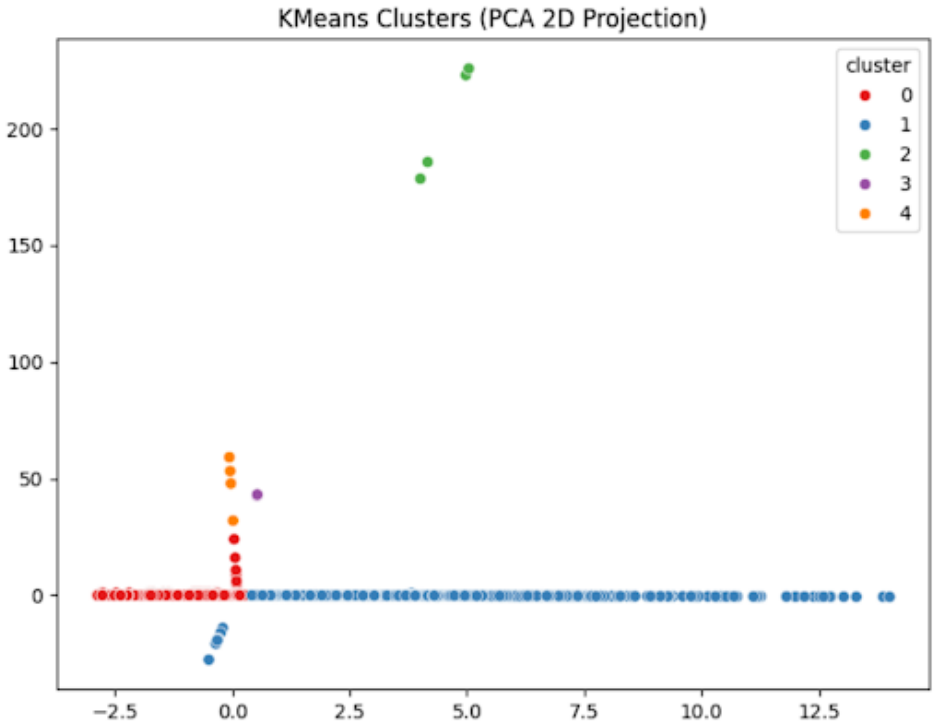

Figure 3: KMeans Clusters (PCA 2D Projection)

Cluster	Model	RMSE	R^2
0	Lasso Regression	7412.82	0.29
0	Ridge Regression	7412.82	0.29
1	Lasso Regression	1737.58	0.97
1	Ridge Regression	1737.58	0.97
4	Lasso Regression	9866.98	-0.37
4	Ridge Regression	9764.99	-0.34
0	LightGBM	5133.30	0.66
1	LightGBM	1733.18	0.97
4	LightGBM	8388.16	0.01

Table 1: Raw Data Results

7.2 2024 Inflation Adjustment Column

This data was clustered into 3 clusters with KMeans using numeric columns alone. Various regression models were applied to each cluster following the clustering. Models attempted here are lasso and ridge regression, lightgbm, and feedforward neural network. Models were compared in terms of RMSE and hyperparameter tuning has not been attempted yet.

Cluster	Model	RMSE	R^2
0	Lasso Regression	2520.36	0.961
0	Ridge Regression	2520.34	0.961
1	Lasso Regression	17302.984	0.0037
1	Ridge Regression	17302.988	0.0037
2	Lasso Regression	2085.54	0.935
2	Ridge Regression	2085.56	0.935
0	LightGBM	2472.27	0.962
1	LightGBM	11013.39	0.596
2	LightGBM	1665.10	0.958

Table 2: 2024 Inflation Adjustment Column

7.3 2024 Inflation Adjustment Row

To account for inflation, we normalized the 2024 sale price for every combination of year, make, model, and trim. We then grouped the data set into two groups based on KMeans considering only numerical features. We used XGBoost and LightGBM models after clustering. These models were compared on RMSE and R^2 and hyperparameter tuning was performed to enhance their performance.

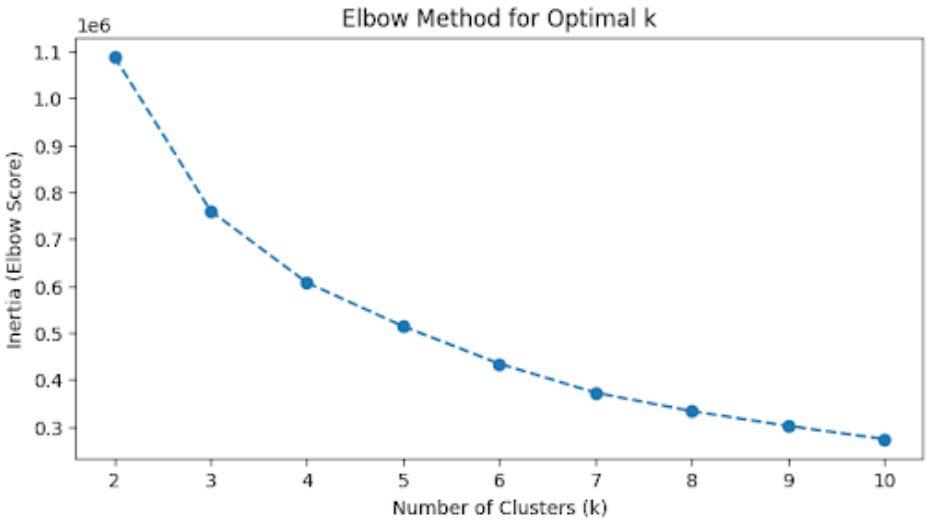

Figure 4: Elbow Method for Optimal k

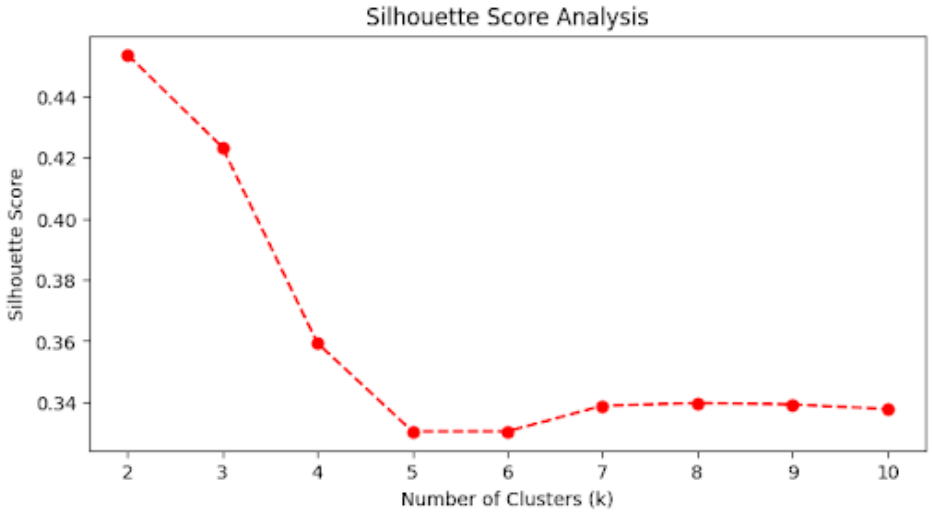

Figure 5: Silhouette Score Analysis

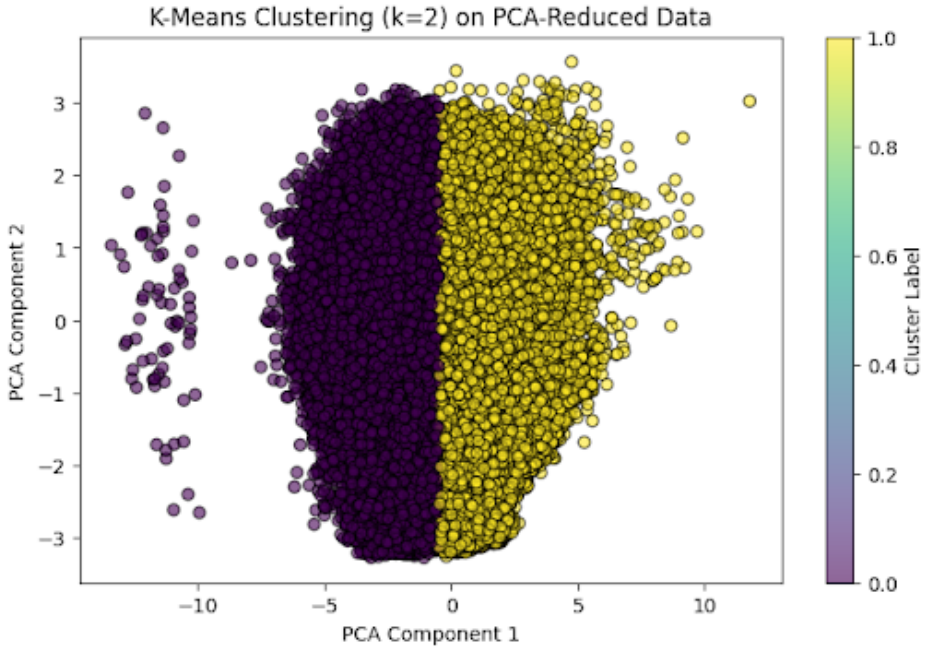

Figure 6: K-Means Clustering (k=2) on PCA-Reduced Data

Cluster	Model	RMSE	R^2
0	XGBoost	2525.35	0.912
0	LightGBM	2520.24	0.965
1	XGBoost	2084.13	0.872
1	LightGBM	2587.11	0.921
2	XGBoost	2084.13	0.932
2	LightGBM	2087.20	0.933

Table 3: 2024 Inflation Adjustment Row

8 Comparison and Final Decision

LightGBM outperformed other models, achieving the lowest RMSE in most cases. XGBoost performed well, but had slightly higher errors. Traditional regression models struggled with clusters that had higher price variance. Future improvements will focus on improving clustering techniques, improving feature engineering, and improving model tuning.