Abstract

The automotive industry is dynamic and prone to frequent market fluctuations, it is consistently difficult to anticipate vehicle sales numbers with any level of accuracy. Manufacturers, dealerships, and other stakeholders need to be able to predict future sales in order to plan marketing campaigns, optimise manufacturing, manage inventories, and make well-informed decisions. As a result, we have decided to use the dataset "vehicle sales and market trend." The method of exploratory data analysis is applied, together with data cleansing and transformation. Given that the dataset contains both numerical and categorical forms, we employed scalable algorithms to convert categorical data into their numerical equivalents. Finally, predictive models are developed to estimate car costs based on a variety of factors and provide information on pricing strategies, market demand, and consumer preferences.

1. Introduction

The automotive industry stands as a cornerstone of modern society, driving economic growth, mobility, and technological innovation. Within this dynamic landscape, the accurate prediction of car prices emerges as a fundamental challenge with profound implications for manufacturers, dealerships, consumers, and ancillary sectors. As the automotive market evolves amidst shifting consumer preferences, technological advancements, and economic uncertainties, the ability to anticipate and effectively manage car prices assumes heightened significance.

Predicting car prices has wide applications including:

- Manufacturer decision-making: Predictive models can help manufacturers understand market demand, competitive pricing, and consumer preferences to optimize pricing strategies.
- Consumer Decision-Making: Predictive models can provide consumers with estimates of fair market value, helping them negotiate prices with dealerships and make budget-conscious decisions
- Risk Assessment: Financial institutions, such as banks and lending agencies, use predictive models to assess the risk associated with auto loans and leases.
- Inventory Management: Predictive models can help dealerships determine the optimal pricing for their inventory based on factors such as market demand, seasonality, and vehicle features.
- Resale Value Estimation: It is crucial for consumers who plan to sell or trade in their vehicles, as it helps them understand the potential depreciation and make financial plans accordingly.
- Automotive R&D: Predicting car prices serves as crucial input for R&D aimed at innovation, product differentiation, improving design and performance to resonate with consumer preferences.
- Policy making: Governments can craft effective tax and emission regulation, driving positive impact on society.

The rest of the paper is structured as follows. Section 2 describes the methodology used including data cleaning, transformation, feature selection and exploratory data analysis. Section 3 describes the different predictive modeling techniques. Section 4 illustrates a result on model implementation and evaluate each of the model performances. Section 5 presents a discussion on the future work to improve models and its application.

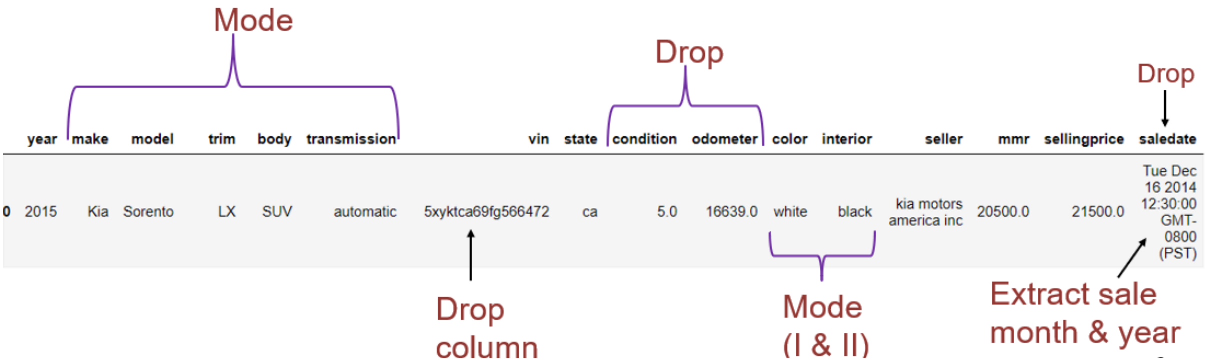
2. Methodology

Data is collected from Kaggle: car_prices. It consists of 558837 rows and 16 features including:

- Car details: make, model, body, transmission, vin, trim, colour, interior and manufacturing year
- Transaction information: state, seller, selling prices and sale dates
- Market trends: MMR
- Condition and Odometer

2.1. Data cleaning

The data cleaning strategy is shown in Fig.1. consist of first row of the dataset to sample its structure. We extract the required information including month and year from “saledate” and then dropped it. Also, “vin” is the unique identification number of vehicle serving irrelevant information to predict selling price and is consequently dropped. We noticed that manufacturing year is greater than selling year for 0.04 % of data, potentially indicating the inclusion of reservation costs in the selling price. So, we dropped it to remove bias in data. Table 1. provides null percentage of each feature, where “transmission” has maximum null values and other features range from 0-2.36 %. In order to solve this, a function that groups data according to the parameters "year," "make," "model," "trim," "body," "colour," "interior," and "state" is created. This allows the mode values for each of the parameters to be filled in for the missing values for "make," "model," "trim," "body," and "transmission." The reason for this approach's 5% fill for "transmission" is that null values for "make," "model," "trim," and "body" overlap. As a result, the dataset is cleansed of these values as well as null entries for "condition" and "odometer". Additionally, to ensure a balanced representation of consumer preferences in the cleaned dataset, 50% of the missing values in the categories "colour" and "interior" are filled with the first colour option made by customers, and the remaining 50% are filled with the second choice. Lastly, we examined distinct entries in every feature and address variations in case sensitivity. For instance, in the “body” unique values such as "sedan" and "Sedan" may exist, representing the same category but with different capitalization. By standardizing the case variation, we unify these entries to enhance the uniformity of the dataset representation.



	year	make	model	trim	body	transmission	vin	state	condition	odometer	color	interior	seller	mmr	sellingprice	saledate
0	2015	Kia	Sorento	LX	SUV	automatic	5xyktca69fg566472	ca	5.0	16639.0	white	black	kia motors america inc	20500.0	21500.0	Tue Dec 16 2014 12:30:00 GMT-0800 (PST)

Fig.1. Data cleaning strategy

Table 1. Null percentage of features

Features	Null percentage
make	1.84
model	1.86
trim	1.9
body	2.36
transmission	11.69
condition	2.11
odometer	0.02
color	0.13
interior	0.13
mmr	0.006
sellingprice	0.002
saledate	0.002

2.2. Exploratory data analysis

We explored different questions which helps at gaining insights, understanding patterns, and identifying trends within a dataset.

Key findings:

- What is the impact of odometer, condition and MMR on selling price?

  Higher the value in the condition column, the more is the selling price. Cars with lower odometer readings and a condition rating of 30 or higher tend to command higher selling prices. Conversely, when the odometer reading is higher, the condition rating is predominantly less than 30, resulting in a decrease in the selling price as shown in Fig.2. Also, selling price increases with increase in “mmr” as shown in Fig. 3.

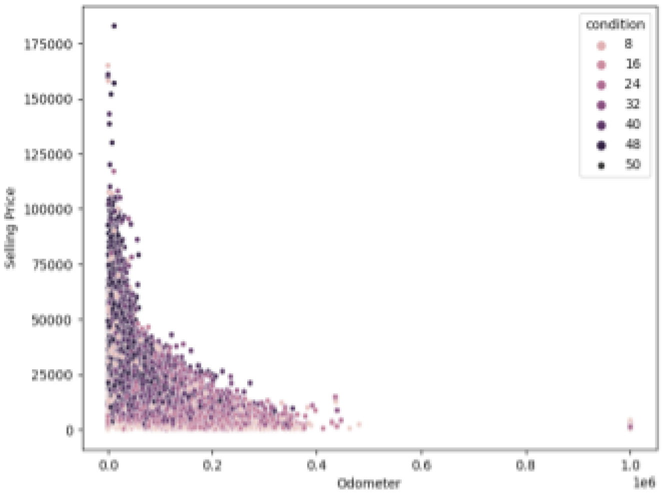


Fig.2. Selling price variation with odometer

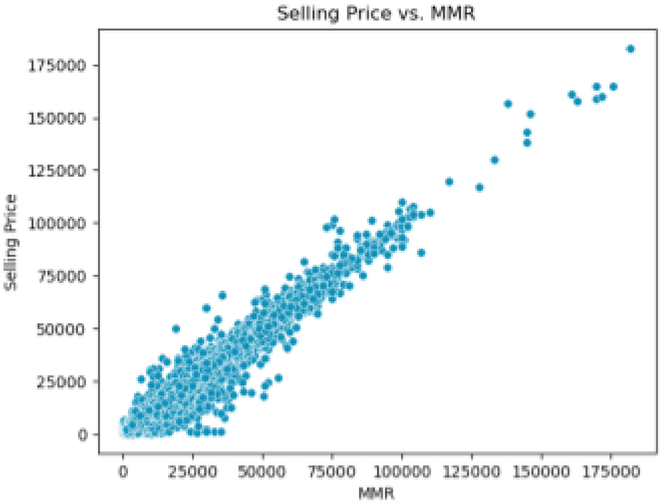


Fig.3. Selling price variation with mmr

- Which manufacturer is the best seller and most preferred combination of model, body, color and transmission type of their cars.

BMW, Ford, Lexus, Chevrolet, Chrysler, Mazda, Audi, Toyota, Ram, Infinity are the top companies in car manufacturing as shown in Fig. 4. Table. 2 shows the different combination of car features of top companies preferred by customer

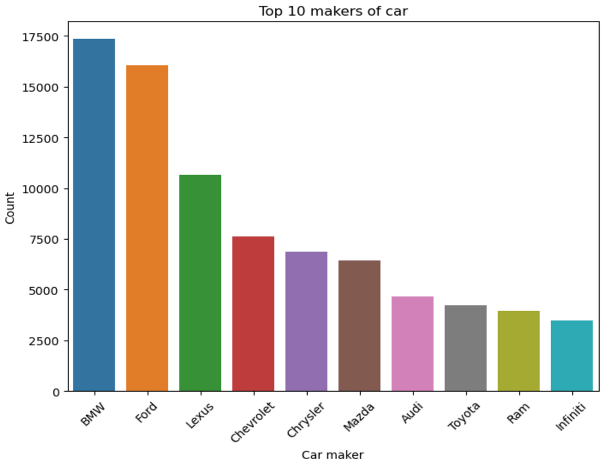


Fig.4. Top companies of cars

Top Makers	Preferred model	Preferred body	Preferred color
BMW	3 Series	Sedan	Black
Ford	F-150	Super Crew	White
Lexus	RX 350	SUV	Black
Chevrolet	Impala	Sedan	Silver
Chrysler	Town and Country	Minivan	White
Audi	A4	Sedan	Black

Table.2. Most preferred feature combination

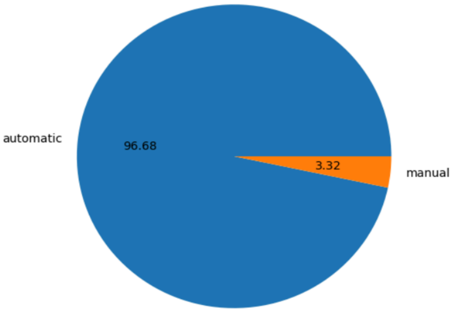


Fig.5. Transmission distribution

- Which body has maximum demand and how do the selling prices vary across the top 10 car body types?

  SUVs, Coupes, and Sedans are noted for their higher selling prices, as indicated in Fig.7. However, when comparing demand, Coupes exhibit lower popularity relative to SUVs and Sedans, as depicted in Fig.6. Consequently, we can infer that SUVs and Sedans are the preferred choices among customers.

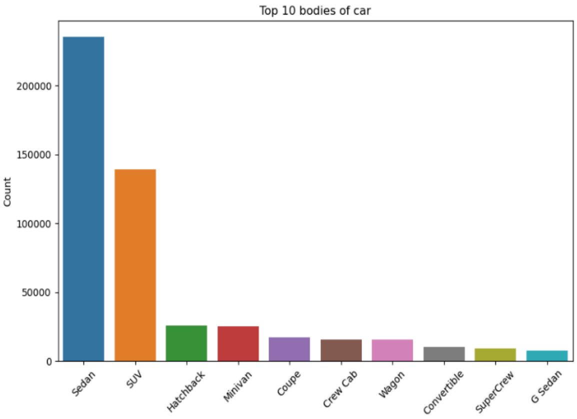


Fig.6. Top bodies of cars

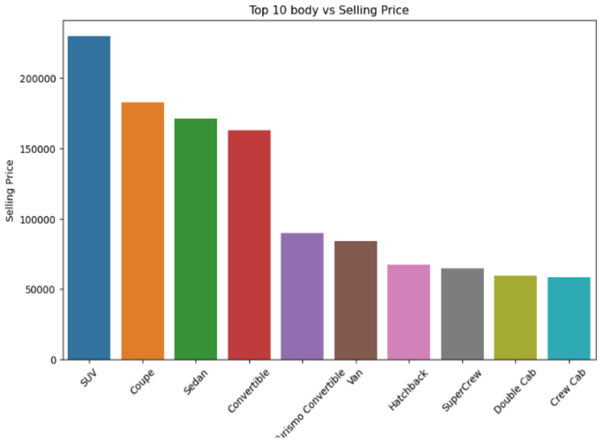


Fig.7. Selling price variation with car body

- Which are the most preferred colors in car and how the preference changes according to selling price?

  Green, red, white black and grey are noted for their higher selling prices, as indicated in Fig.9. However, when comparing demand, green and red exhibit lower popularity relative to white, black and grey, as depicted in Fig.8. Consequently, we can infer that neutral colors are the preferred choices among customers. Understanding this market trend can provide valuable insights for startup companies, guiding their design and product development efforts to align with consumer preferences. By focusing on neutral color options, these companies can enhance their appeal to a broader customer base and capitalize on market demand more effectively.

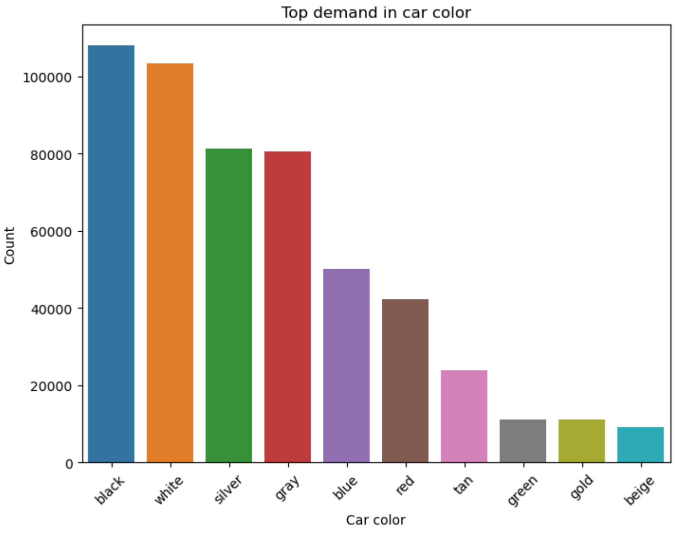


Fig.8. Top car colors

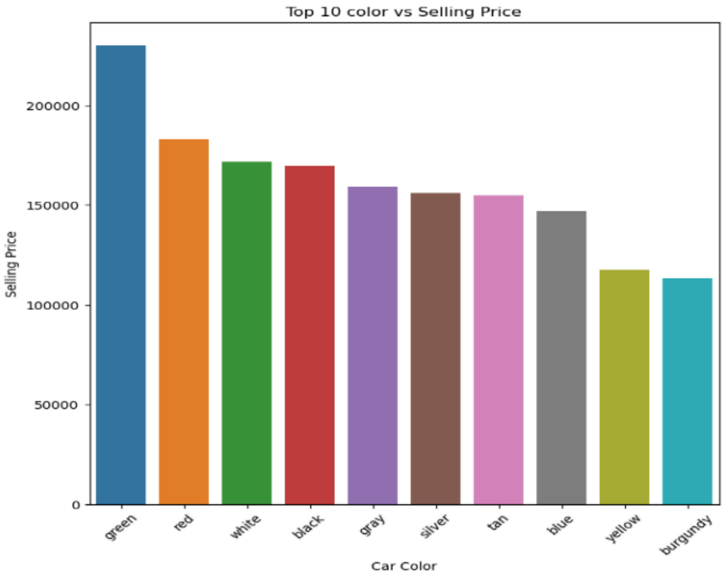


Fig.9. Selling price variation with colors

- Which month has the maximum demand?

The observed pattern in Fig.10. suggests that there may be seasonal variations in car sales. The peaks in sales during the first two months could indicate higher demand at the beginning of the year, possibly due to factors like tax returns, year-end bonuses, or new model releases. Similarly, increased sales in the 5th, 6th, and 12th months might be influenced by seasonal events or promotions, such as end-of-year sales or holiday discounts. Dealerships may adjust their inventory management strategies based on seasonal demand patterns. Higher sales in certain months could lead to adjustments in inventory levels and pricing strategies to optimize sales and maximize profits.

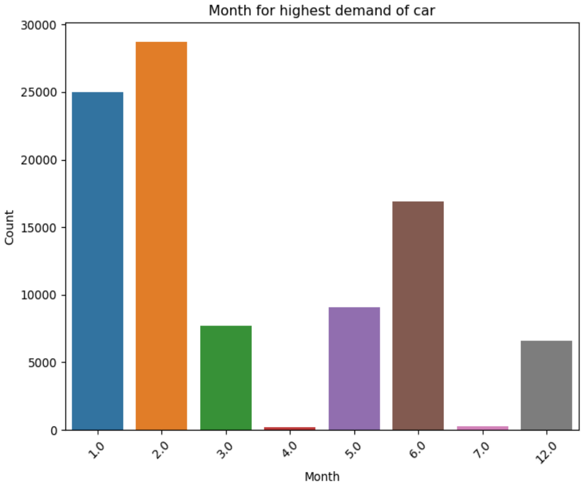


Fig.10. Demand variation with months

- What are the demographics for different companies?

  Florida and California have the maximum demand in Fig.11. although the selling price is highest in these states. The reason can be high population density and taxes are less in Florida and Texas. Table.3. shows the market capture by companies in different top states. This information can be valuable for understanding market dynamics and competitive positioning, helping companies maintain their strategies to better serve customers and capture market share effectively.

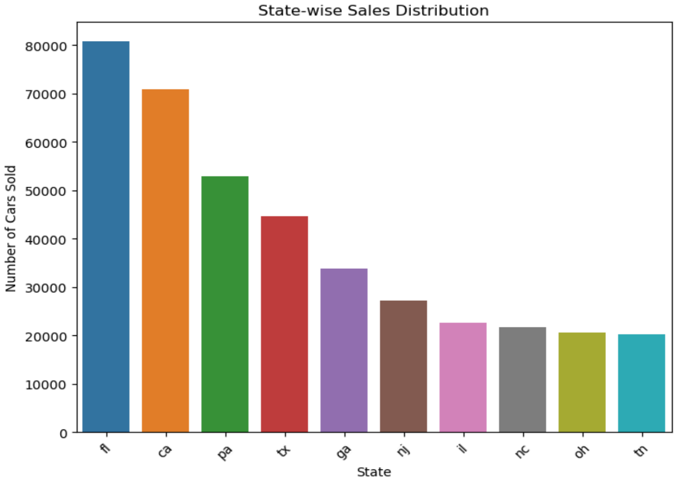


Fig 11: Number of cars sold per state

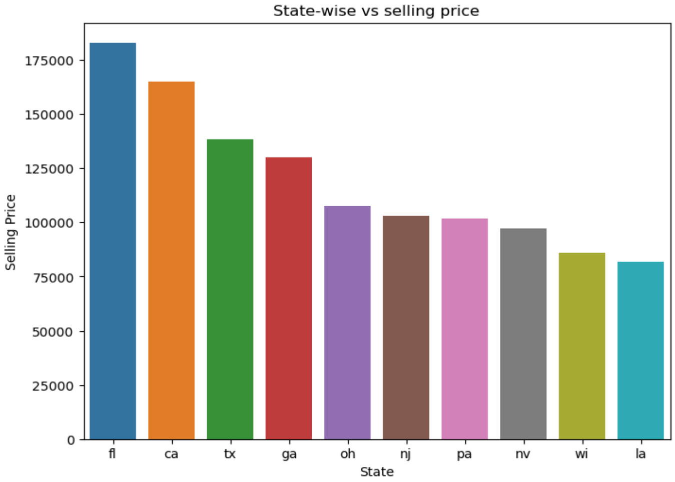


Fig 12: Selling price per state

Table 3. Top companies in state

State	Top manufacturer
CA	BMW
FL	Lexus
TX	Ford
GA	BMW
NC	BMW

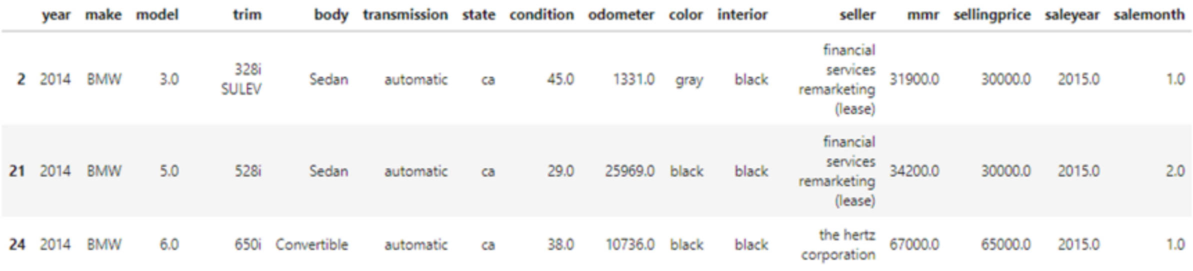
2.3. Data Transformation

2.3.1. Models Implemented

1. Label Encoding

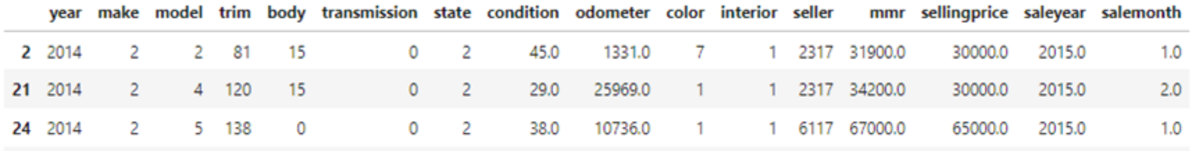
This is a common technique in machine learning to convert categorical data into numerical value. This method changes every category in a categorical variable into a distinct numerical code. When working with numerical data, categorical data is typically more difficult. In our dataset, several categorical labels such as ‘trim’, ‘make’, ‘model’, ‘trim’, ‘body’, ‘transmission’, ‘state’, ‘color’, interior’, and ‘seller’ are present. These labels contain categorical data that need to be transformed into numerical values for analysis. Label encoding is used to accomplish this transformation, giving each category a numerical code within a given range determined by its categorical value.

Before applying label encoding, the data for these categorical labels appears as shown in Figure 13. For instance, the ‘make’ label has values such as ‘BMW’, while ‘body’ label has categories like ‘Sedan’, and so forth. Once label encoding is applied, the categorical data, such as ‘BMW’ and ‘Sedan’ are converted into numerical data 2 and 15, respectively as illustrated in Figure 14.



	year	make	model	trim	body	transmission	state	condition	odometer	color	interior	seller	mmr	sellingprice	saleyear	salemonth
2	2014	BMW	3.0	328i SULEV	Sedan	automatic	ca	45.0	1331.0	gray	black	financial services remarketing (lease)	31900.0	30000.0	2015.0	1.0
21	2014	BMW	5.0	528i	Sedan	automatic	ca	29.0	25969.0	black	black	financial services remarketing (lease)	34200.0	30000.0	2015.0	2.0
24	2014	BMW	6.0	650i	Convertible	automatic	ca	38.0	10736.0	black	black	the hertz corporation	67000.0	65000.0	2015.0	1.0

Fig.13. Datasets Before Transformed with Label Encoding



	year	make	model	trim	body	transmission	state	condition	odometer	color	interior	seller	mmr	sellingprice	saleyear	salemonth
2	2014	2	2	81	15	0	2	45.0	1331.0	7	1	2317	31900.0	30000.0	2015.0	1.0
21	2014	2	4	120	15	0	2	29.0	25969.0	1	1	2317	34200.0	30000.0	2015.0	2.0
24	2014	2	5	138	0	0	2	38.0	10736.0	1	1	6117	67000.0	65000.0	2015.0	1.0

Fig.14. Datasets After Transformed with Label Encoding

2. Feature Engineering

Feature engineering is the process of transforming unprocessed data into a format that improves machine learning model performance. The goal is to extract meaningful features that effectively contribute to the predictive accuracy of the models. In this project, two engineering features, ‘Car_age’ and ‘Milesperyear’, have been developed for this project.

‘Car_age’ helps customers make decisions by offering useful information about the age of the used car. However, ‘Milesperyear’ measures each used car’s travel distance and provides important details about how it was utilized. These engineered features not only enrich the dataset but also improve the predictive capabilities of the models, which increases the model’s effectiveness in producing well-informed predictions.

3. Isolation Forests

Isolation forest is an unsupervised anomaly detection algorithm that operates on the principle of isolating outliers within a dataset. The technique finds anomalies (outliers) which are simpler to isolate than typical data points by dividing the data space at random. It assumes that outliers need fewer splits to separate from the majority of the data. Based on the number of splits needed, an anomaly score is assigned to each data point; higher scores indicate a higher probability of being an outlier.

4. Collinearity Test with Pearson Correlation Coefficient

Collinearity test is needed to identify the significant degree of correlation between two or more independent variables. High correlation could lead to redundancy in the dataset and affect the predictive model’s performance. In this project, we use Pearson correlation. This method assesses the direction and intensity of the linear relationship between two features. The values are ranging from -1 to 1. A coefficient of -1 indicates perfect negative correlation, 0 signifies no correlation, and 1 means perfect positive correlation. We use a a threshold of 0.9, which means that if the correlation coefficient is higher than this value, one of the attributes will be excluded.

5. Feature Scaling Using StandardScaler

Feature scaling is used to standardize the range of features because they have different ranges. In this project, we use the StandardScaler method for this purpose. It scales each feature to possess a mean of 0 and a standard deviation of 1, creating features with a standard normal distribution. This approach adeptly resolves issues associated with feature scales, biases, and outliers, all while maintaining the integrity of the data’s relationships and interpretability.

6. ANOVA and Mutual Information

ANOVA (Analysis of Variance) is a statistical method to compare means across two or more groups. It calculates the F-statistics, which is the ratio of variance between groups and variance within groups. Associated with the F-statistics is the p-value, which signifies the likelihood of observing the F-statistics under the null hypothesis that features hold no influence over the target variable. In feature selection, both F-statistics and p-value play crucial roles in assessing the significance of individual features concerning the target variable. Higher

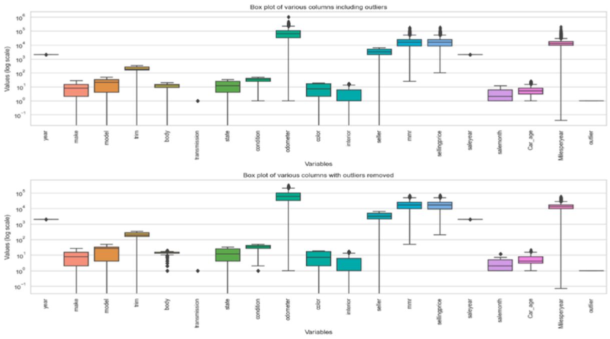
F-statistics added with low p-value indicate strong associations between features and the target.

Mutual information (MI) is a machine learning technique employed to gauge the dependence between features and the target. It quantifies the mutual dependence, measuring how much information one variable provides about the other. MI adeptly captures non-linear relationships between variables with high MI values indicating a strong non-linear association with the target. In this project, we combine F-statistics, p-value, and MI value to identify which features have strong association with the target. This will help the predictive models' performance.

2.3.2 Features Preprocessing

1. Outlier removal

We have designated 'auto' as the contamination or outlier value, resulting in the removal of approximately 6000 samples. Figure 15 below illustrates the distribution of dataset both before and after outlier removal. The box plots depict that there is minimal discrepancy between the two distributions.


Box plot of various columns including outliers

Box plot of various columns with outliers removed

Fig.15. Datasets After Transformed with Label Encoding

2. Collinearity test

By employing a threshold of 0.9, we identify a strong correlation between the 'Car_age' and 'year' features. Consequently, we eliminate the 'Car_age' feature from consideration. Initially, the number of selected features stands at 17. However, after removing 'Car_age' feature, the count reduces to 16 selected features. The collinearity test result can be observed in figure 16.

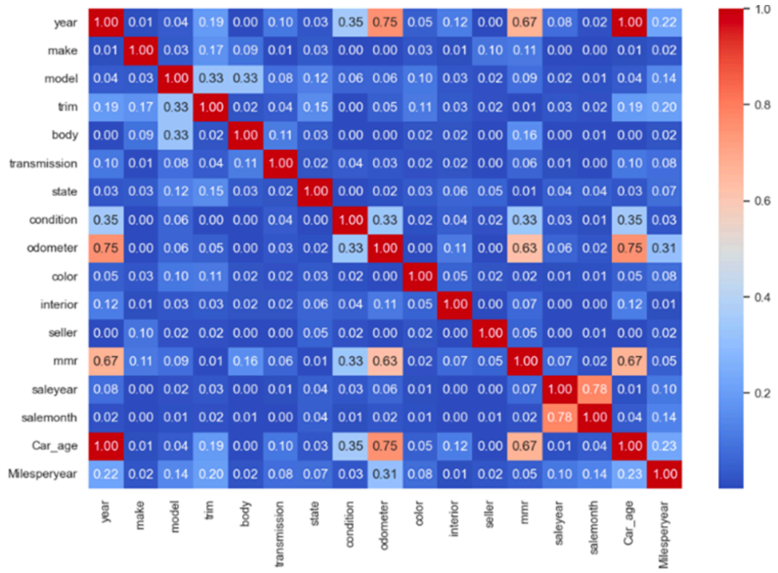


Fig.16. Correlation Matrix Result

3. Feature selection

We categorize 16 features into two distinct groups: continuous features and categorical features. Within each category, we conduct evaluation based on F-statistics and MI-values, employing varying thresholds tailored to the characteristics of each category as seen in figure 17.. Through this analysis, we identify six features that have strong linear and non-linear associations with the target variable. These features are ‘Mmr’, ‘Year’, ‘Model’, ‘Trim’, ‘Condition’, and ‘Seller’.

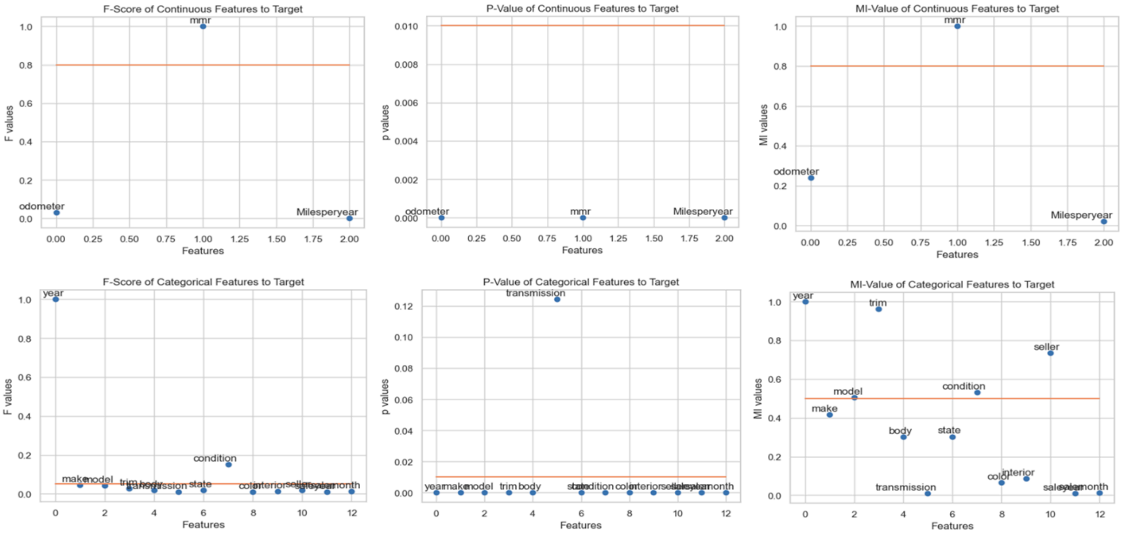


Fig.17. F-Score, P-Values, and MI-Values for each Category

3. Predictive modeling techniques

3.1. Models Implemented

1. Linear Regression: It is a deterministic statistical method used to model the relationship between a dependent variable and one or more independent variables. It assumes a linear relationship between the independent variables and the dependent variable. The basic form of a linear regression model with one independent variable can be represented as:

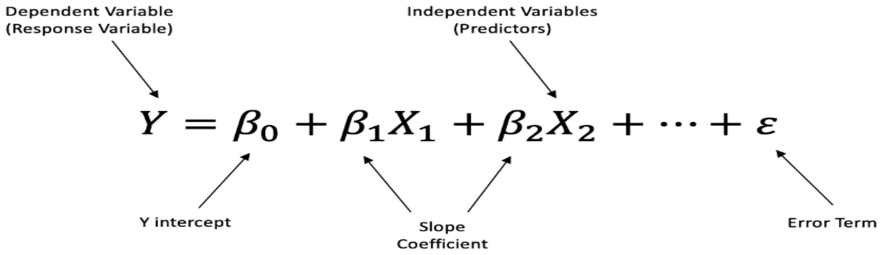


Fig 18: Linear Regression Formula

It aims to find the values of the coefficients that minimize the sum of the squared differences between the observed and predicted values of the dependent variable.

2. Ridge Regression: It is also known as L2 regularization, is a regularization technique used in linear regression to address multicollinearity (high correlation between independent variables) and overfitting (when the model fits the training data too closely, leading to poor generalization to new data). The penalty term is added to the least squares objective function to shrink the coefficients of the regression model towards zero. This penalty term is proportional to the square of the magnitude of the coefficients, resulting in smaller coefficient values and a more stable model.

$$L(w) = RSS + \lambda \sum_{j=1}^{N} w_j^2 = \sum_{p=1}^{P} (y_p - w_0 - \sum_{j=1}^{N} w_j x_{pj})^2 + \lambda \sum_{j=1}^{N} w_j^2$$

Fig 19: Ridge Regression Cost Function Formula

3. Lasso Regression
Lasso regression is a linear regression which uses L1 penalty term to the ordinary least squares (OLS) objective function. With this penalty, Lasso regression solves overfitting issues and facilitates feature selection by encompassing the absolute values of all the coefficients within the model. As L1 penalty increases, models with larger coefficients are penalized, such that certain coefficients shrink to zero. The features with zero coefficients are excluded from the model entirely. This

process yields a sparse model with fewer features, thereby diminishing complexity and mitigating overfitting.

$$L(w) = RSS + \lambda \sum_{j=1}^{N} |w_j| = \sum_{p=1}^{P} (y_p - w_0 - \sum_{j=1}^{N} w_j x_{pj})^2 + \lambda \sum_{j=1}^{N} |w_j|$$

Figure 20: Lasso Regression Cost Function Formula

4. Elastic Net Regression
   Elastic Net regression uses two regularization methods which are Lasso regression and Ridge regression within the linear regression framework. With these techniques, Elastic Net can solve multicollinearity and feature selection issues. This combination is achieved through the manipulation of a hyperparameter, alpha, which controls the balance between the influences of L1 and L2 penalties. It is commonly used in a condition where there are many predictors which some of them may be correlated. Moreover, it proves particularly advantageous in handling high-dimensional datasets.

$$J(w, \lambda_1, \lambda_2) = ||y - Xw||^2 + \lambda_2 ||w||_2^2 + \lambda_1 ||w||_1$$

Figure 21: Elastic Net Regression Cost Function Formula

5. Random Forests Regression
   A formidable ensemble learning approach, Random Forests operates by building several decision trees during training and returning the mean prediction of each individual tree. To encourage variability among the trees, each tree in the forest is constructed using a sample from the training data and a randomly chosen feature set. This sampling method is called bagging. This unpredictability contributes to the model's increased robustness and reduces overfitting. Compared to other methods, Random Forests have a number of advantages, including as their inherent ability to manage non-linear correlations and interactions between features, their capacity to handle big datasets with high dimensionality, and their ability to handle missing values well. They can also provide estimations of feature relevance. Furthermore, they are easier to tune and less prone to overfitting than other models like Support Vector Machines since they are less sensitive to hyperparameters.

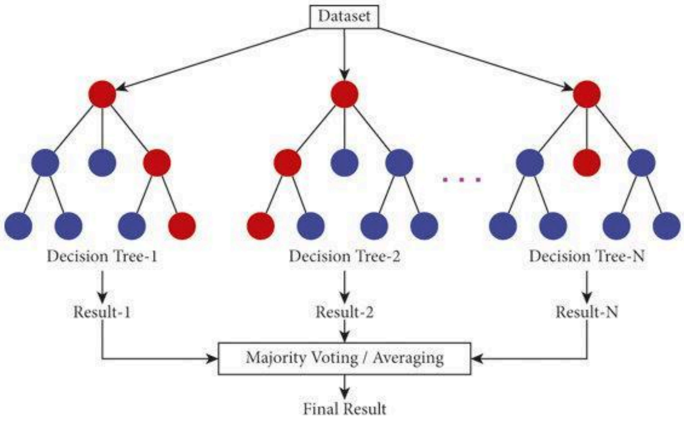


Figure 22: random Forests Workflow Diagram

6. Extreme Gradient Boosted Regression

Extreme Gradient Boosting, or XGBoost, is an incredibly effective ensemble learning method that excels at tasks using structured or tabular data. It creates a sequence of decision trees one after the other, training each one to fix the mistakes of the preceding one. By adding new trees that minimize the overall loss, XGBoost optimizes a global objective function in contrast to Random Forests, which construct trees independently. With every extra tree added, this iterative process improves the predicted accuracy of the model. XGBoost improves its generalization capabilities by including regularization techniques like shrinkage (learning rate) and pruning to control overfitting. Moreover, XGBoost is scalable and appropriate for large datasets because it allows distributed and parallel processing. Other strengths include its robustness against outliers, automatic handling of feature interactions, and internal handling of missing values. With a large range of hyperparameters to fine-tune model performance, XGBoost is extremely flexible.

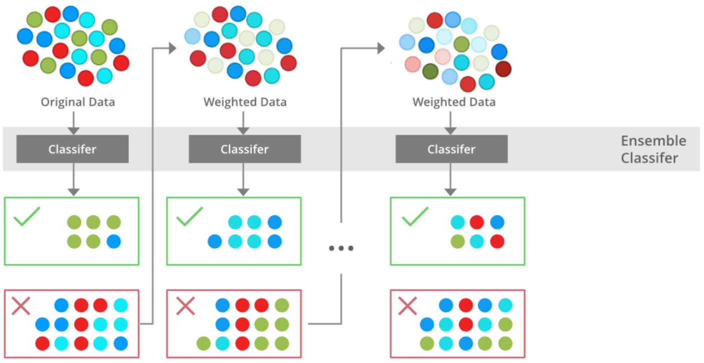


Figure 23: Extreme Gradient Boosting Workflow Diagram

7. K-Nearest Neighbors Regression

For both regression and classification applications, the K-Nearest Neighbors (KNN) algorithm is a straightforward but powerful non-parametric supervised learning technique. The underlying premise of it is that similar data points typically belong to the same class or have similar output values. Based on a distance metric (such as the Euclidean distance) in the feature space, KNN finds the K nearest neighbors when predicting the class of a new data point in classification. The anticipated class for the new data point is the one that most of these neighbors belong to. Similar to this, KNN predicts the output value in regression by averaging or using the output values of its closest neighbors as a proxy for majority vote. KNN is a popular option for quick prototyping and for beginners because to its ease of understanding and intuitiveness. But because KNN needs calculating distances between each new data point and all of the training data points, its primary disadvantage is computational inefficiency, particularly when dealing with huge datasets. Additionally, the success of KNN is highly dependent on the value of K and the selection of the distance measure, both of which require careful tuning.

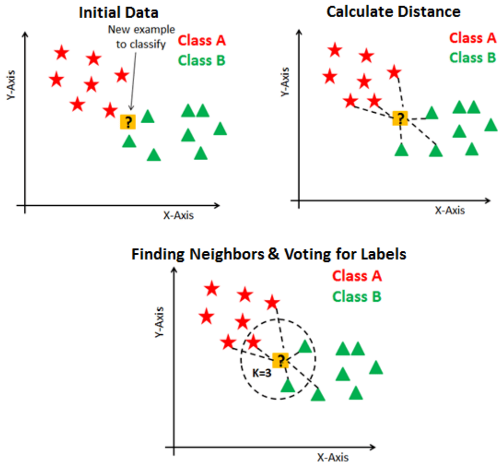
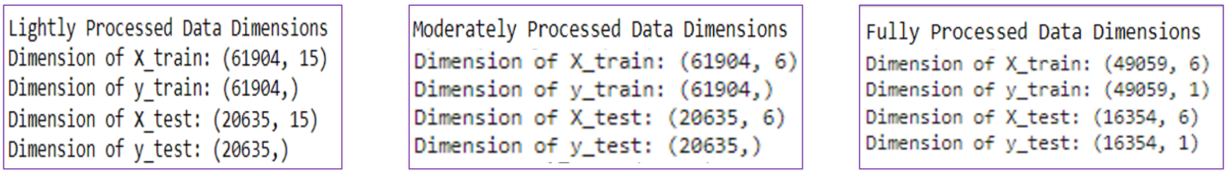


Figure 24: K Nearest Neighbors Workflow Diagram

3.2 Modelling Results

In order to properly investigate the properties of the data, as well as to assemble the best model we implemented three sets of predictive models, each consisting of the seven models listed above. The first set was implemented using data that had undergone only the basic steps of preprocessing, namely label encoding and elimination of duplicated/redundant values. The second set was implemented using data that had undergone feature selection using the ANOVA algorithm, dropping the number of attributes from 15 to 6. The final set was trained on data that had undergone outlier removal using the isolation forests algorithm, which dropped around 50% of the dataset. One of the biggest reasons for implementing several models was due to the murkiness surrounding outlier removal when it comes to categorical data, as with such data outliers are wholly based on context. As such, we found it prudent to investigate whether the model performs better with or without the presumed outliers.



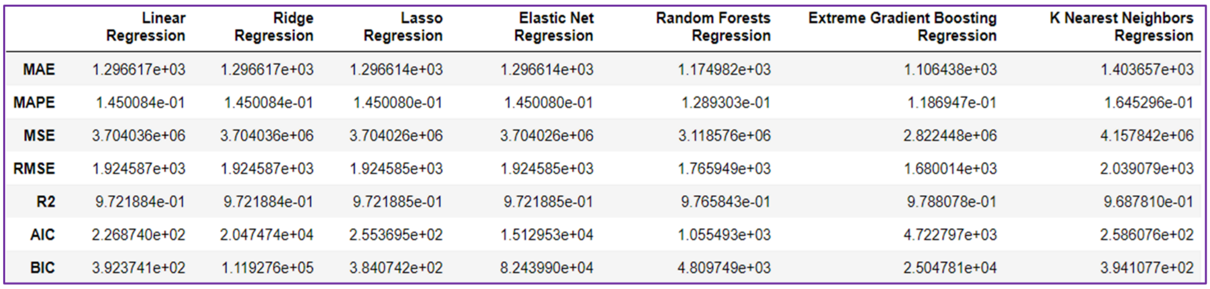
```
Lightly Processed Data Dimensions
Dimension of X_train: (61904, 15)
Dimension of y_train: (61904,)
Dimension of X_test: (20635, 15)
Dimension of y_test: (20635,)
```

```
Moderately Processed Data Dimensions
Dimension of X_train: (61904, 6)
Dimension of y_train: (61904,)
Dimension of X_test: (20635, 6)
Dimension of y_test: (20635,)
```

```
Fully Processed Data Dimensions
Dimension of X_train: (49059, 6)
Dimension of y_train: (49059, 1)
Dimension of X_test: (16354, 6)
Dimension of y_test: (16354, 1)
```

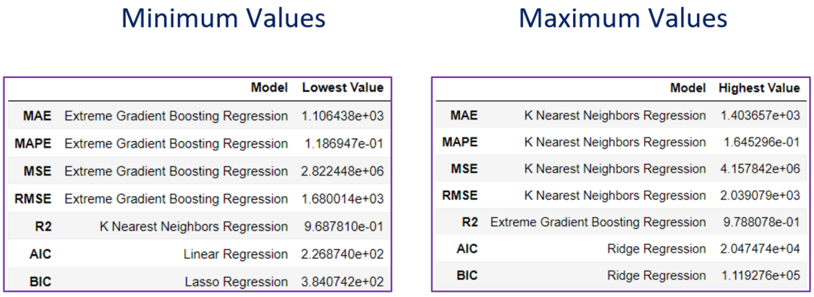
Figure 25: Dimensions of data used to train and validate predictive models

All of the models are tuned using the cross-validation model and implemented. The results from the implementation is interesting, as all of the models perform extremely well and return very similar results. Even across the different sets we implemented, there is negligible variation between the results of the models.



	Linear Regression	Ridge Regression	Lasso Regression	Elastic Net Regression	Random Forests Regression	Extreme Gradient Boosting Regression	K Nearest Neighbors Regression
MAE	1.296617e+03	1.296617e+03	1.296614e+03	1.296614e+03	1.174982e+03	1.106438e+03	1.403657e+03
MAPE	1.450084e-01	1.450084e-01	1.450080e-01	1.450080e-01	1.289303e-01	1.186947e-01	1.645296e-01
MSE	3.704036e+06	3.704036e+06	3.704026e+06	3.704026e+06	3.118576e+06	2.822448e+06	4.157842e+06
RMSE	1.924587e+03	1.924587e+03	1.924585e+03	1.924585e+03	1.765949e+03	1.680014e+03	2.039079e+03
R2	9.721884e-01	9.721884e-01	9.721885e-01	9.721885e-01	9.765843e-01	9.788078e-01	9.687810e-01
AIC	2.268740e+02	2.047474e+04	2.553695e+02	1.512953e+04	1.055493e+03	4.722797e+03	2.586076e+02
BIC	3.923741e+02	1.119276e+05	3.840742e+02	8.243990e+04	4.809749e+03	2.504781e+04	3.941077e+02

Figure 26: Results of the models trained on the first dataset



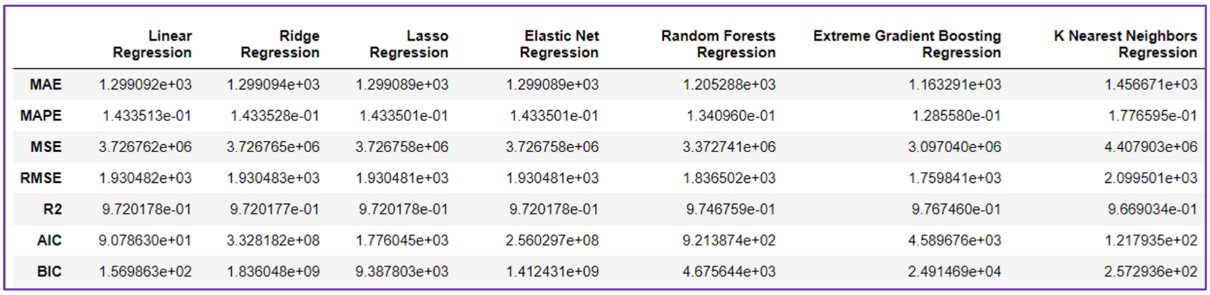
Minimum Values

	Model	Lowest Value
MAE	Extreme Gradient Boosting Regression	1.106438e+03
MAPE	Extreme Gradient Boosting Regression	1.186947e-01
MSE	Extreme Gradient Boosting Regression	2.822448e+06
RMSE	Extreme Gradient Boosting Regression	1.680014e+03
R2	K Nearest Neighbors Regression	9.687810e-01
AIC	Linear Regression	2.268740e+02
BIC	Lasso Regression	3.840742e+02

Maximum Values

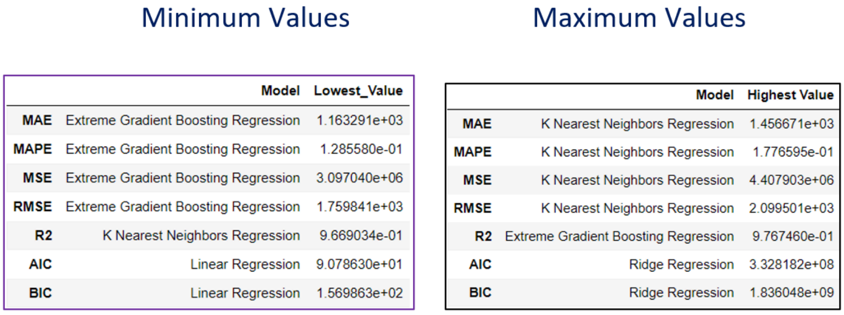
	Model	Highest Value
MAE	K Nearest Neighbors Regression	1.403657e+03
MAPE	K Nearest Neighbors Regression	1.645296e-01
MSE	K Nearest Neighbors Regression	4.157842e+06
RMSE	K Nearest Neighbors Regression	2.039079e+03
R2	Extreme Gradient Boosting Regression	9.788078e-01
AIC	Ridge Regression	2.047474e+04
BIC	Ridge Regression	1.119276e+05

Figure 27: The minimum and maximum validation results for the models trained on the first dataset



	Linear Regression	Ridge Regression	Lasso Regression	Elastic Net Regression	Random Forests Regression	Extreme Gradient Boosting Regression	K Nearest Neighbors Regression
MAE	1.299092e+03	1.299094e+03	1.299089e+03	1.299089e+03	1.205288e+03	1.163291e+03	1.456671e+03
MAPE	1.433513e-01	1.433528e-01	1.433501e-01	1.433501e-01	1.340960e-01	1.285580e-01	1.776595e-01
MSE	3.726762e+06	3.726765e+06	3.726758e+06	3.726758e+06	3.372741e+06	3.097040e+06	4.407903e+06
RMSE	1.930482e+03	1.930483e+03	1.930481e+03	1.930481e+03	1.836502e+03	1.759841e+03	2.099501e+03
R2	9.720178e-01	9.720177e-01	9.720178e-01	9.720178e-01	9.746759e-01	9.767460e-01	9.669034e-01
AIC	9.078630e+01	3.328182e+08	1.776045e+03	2.560297e+08	9.213874e+02	4.589676e+03	1.217935e+02
BIC	1.569863e+02	1.836048e+09	9.387803e+03	1.412431e+09	4.675644e+03	2.491469e+04	2.572936e+02

Figure 28: Results of the models trained on the second dataset



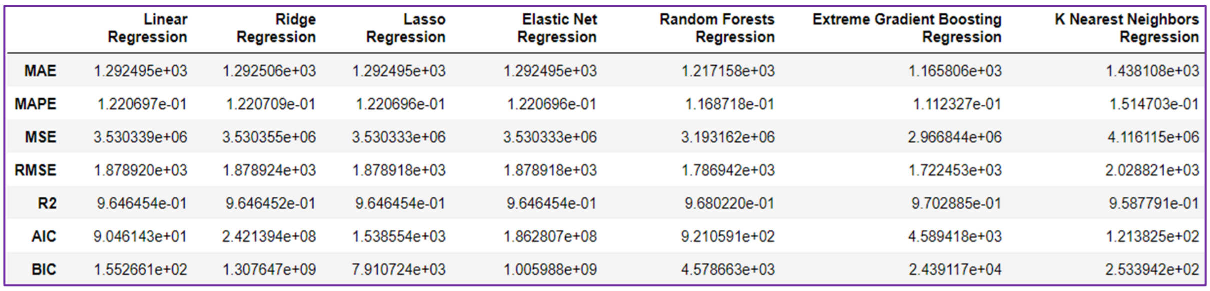
Minimum Values

	Model	Lowest_Value
MAE	Extreme Gradient Boosting Regression	1.163291e+03
MAPE	Extreme Gradient Boosting Regression	1.285580e-01
MSE	Extreme Gradient Boosting Regression	3.097040e+06
RMSE	Extreme Gradient Boosting Regression	1.759841e+03
R2	K Nearest Neighbors Regression	9.669034e-01
AIC	Linear Regression	9.078630e+01
BIC	Linear Regression	1.569863e+02

Maximum Values

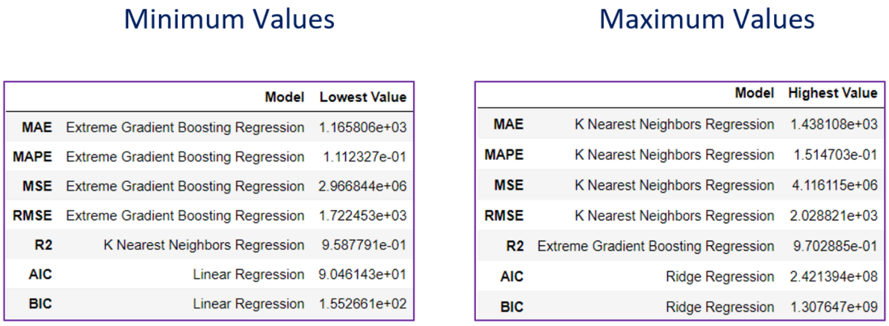
	Model	Highest Value
MAE	K Nearest Neighbors Regression	1.456671e+03
MAPE	K Nearest Neighbors Regression	1.776595e-01
MSE	K Nearest Neighbors Regression	4.407903e+06
RMSE	K Nearest Neighbors Regression	2.099501e+03
R2	Extreme Gradient Boosting Regression	9.767460e-01
AIC	Ridge Regression	3.328182e+08
BIC	Ridge Regression	1.836048e+09

Figure 29: Minimum and maximum validation results for the models trained on the second dataset



	Linear Regression	Ridge Regression	Lasso Regression	Elastic Net Regression	Random Forests Regression	Extreme Gradient Boosting Regression	K Nearest Neighbors Regression
MAE	1.292495e+03	1.292506e+03	1.292495e+03	1.292495e+03	1.217158e+03	1.165806e+03	1.438108e+03
MAPE	1.220697e-01	1.220709e-01	1.220696e-01	1.220696e-01	1.168718e-01	1.112327e-01	1.514703e-01
MSE	3.530339e+06	3.530355e+06	3.530333e+06	3.530333e+06	3.193162e+06	2.966844e+06	4.116115e+06
RMSE	1.878920e+03	1.878924e+03	1.878918e+03	1.878918e+03	1.786942e+03	1.722453e+03	2.028821e+03
R2	9.646454e-01	9.646452e-01	9.646454e-01	9.646454e-01	9.680220e-01	9.702885e-01	9.587791e-01
AIC	9.046143e+01	2.421394e+08	1.538554e+03	1.862807e+08	9.210591e+02	4.589418e+03	1.213825e+02
BIC	1.552661e+02	1.307647e+09	7.910724e+03	1.005988e+09	4.578663e+03	2.439117e+04	2.533942e+02

Figure 30: Results of the models trained on the third dataset



Minimum Values

	Model	Lowest Value
MAE	Extreme Gradient Boosting Regression	1.165806e+03
MAPE	Extreme Gradient Boosting Regression	1.112327e-01
MSE	Extreme Gradient Boosting Regression	2.966844e+06
RMSE	Extreme Gradient Boosting Regression	1.722453e+03
R2	K Nearest Neighbors Regression	9.587791e-01
AIC	Linear Regression	9.046143e+01
BIC	Linear Regression	1.552661e+02

Maximum Values

	Model	Highest Value
MAE	K Nearest Neighbors Regression	1.438108e+03
MAPE	K Nearest Neighbors Regression	1.514703e-01
MSE	K Nearest Neighbors Regression	4.116115e+06
RMSE	K Nearest Neighbors Regression	2.028821e+03
R2	Extreme Gradient Boosting Regression	9.702885e-01
AIC	Ridge Regression	2.421394e+08
BIC	Ridge Regression	1.307647e+09

Figure 31: Minimum and maximum validation results for the models trained on the third dataset

Through visualizations, we can also see that all of the models, across the sets, yield extremely accurate results on the validation set. However, none of the models are able to accurately capture the variation present in the dataset.

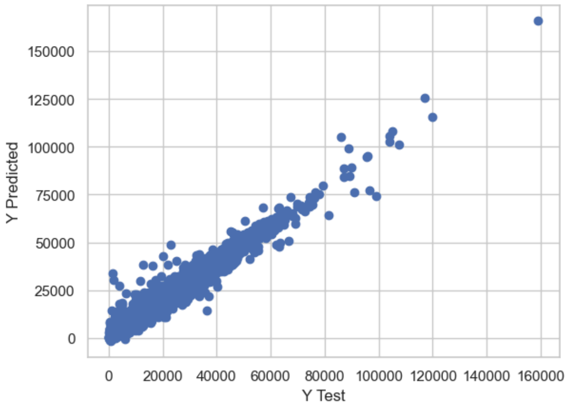


Figure 32: Accuracy of Predictions

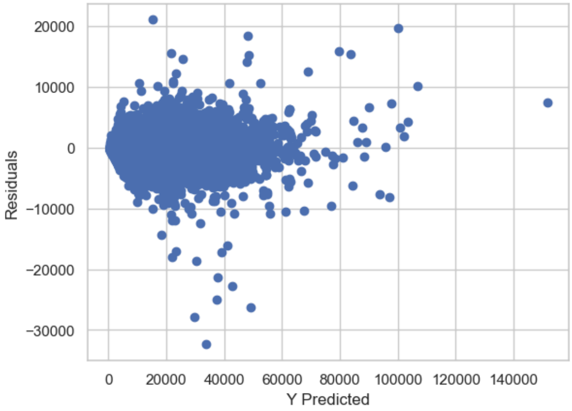


Figure 33: Variation Captured

After plotting the accuracy of the predictions and the variation captured for all 21 models, we get near identical plots. There could be two reasons for why all of the models yield similar results. The first is that none of the models are complex enough to capture the underlying patterns present in the data. This is not very likely as we have used a fairly wide range of models in terms of complexity ranging from simple models like K-nearest neighbors and complex models like extreme gradient boosting. Additionally, the data we are processing concerns car sales which is unlikely to contain the level of complexity that would cause this error. The other, more likely explanation is that the pattern is extremely prevalent throughout the data, to the point where regardless of the methodology or complexity of the model, the implemented model will pick up on that thread and follow through on it. This is the most likely explanation as there are very strong trends that prevail within the domain of car sales.

4. Results

We have implemented several machine learning models such as linear regression, random forests regression and k-nearest neighbors in order to correctly predict the selling price of a car based on the features of the car, such as the make and condition of the car. After preprocessing and modelling the data, we have found that linear regression performs the best when fed data that has undergone feature selection and data cleaning, but not outlier removal; with a root mean square error value of 1,930.48, an R squared score of 0.97, an AIC score of 90.78 and a BIC score of 156.98.

5. Conclusions

In this project, we have conducted several operations in order to create a model that best predicts the selling price of a car based on its make and condition. We conducted several preprocessing steps such as filling in all of the null values that we could and dropping the rest, eliminating duplicate values and converting the categorical values to integer values for easier procession. We then conducted feature selection using the ANOVA algorithm and eliminated outliers using the isolation forests. Throughout this process, we saved different instances of the data in order to see which of them would provide the most amount of information to the model. We then implemented several machine learning models such as the random forests algorithm, elastic net regression and k-nearest neighbors in order to find the model most suited to the data at hand. After this process, we found that due to a pervasive pattern in the data almost all of the models perform extremely well. We found that the linear regression model was the best out of the lot, when trained with data that included the best features and most of the rows; due to it having the best fit while simultaneously being the least complex model.

6. Future work

As every model performs consistently, it is possible that there is not enough variability in the data, producing results that are similar. We plan to investigate more sophisticated approaches, such deep learning models, to find hidden patterns in the data in order to improve model performance. To determine whether the tendencies we've seen are unique to our dataset or a sign of more general trends in the automotive sector, we also intend to repeat the research with a bigger and more varied dataset. In addition, we want to improve user experience by incorporating prediction models into applications that interact with users. Customers will be able to input their desired features and get an estimate of the selling pricing as a result, making informed decisions easier.

7. Reference

[1] Guha, Sharmistha. (2024). " Lecture Notes: Model Selection and Regularization." STAT 654, Texas A&M University.

[2]Guha, Sharmistha. (2024). "Lecture Notes: Regression Analysis Part 1." STAT 654, Texas A&M University.

[3]Guha, Sharmistha. (2024). "Lecture Notes: Regression Analysis Part 3." STAT 654, Texas A&M University.

[4] https://www.researchgate.net/figure/Illustration-of-random-forest-trees_fig4_354354484

[5]https://medium.com/@techynilesh/xgboost-algorithm-explained-in-less-than-5-minutes-b561dcc1ccee

[6] https://towardsdatascience.com/knn-visualization-in-just-13-lines-of-code-32820d72c6b6

[7] https://scikit-learn.org/stable/modules/generated/sklearn.linear_model.LinearRegression.html

[8] https://www.ibm.com/topics/ridge-regression

[9]https://www.analyticsvidhya.com/blog/2017/06/a-comprehensive-guide-for-linear-ridge-and-lasso-regression/

[10]https://www.axalta.com/content/dam/New%20Axalta%20Corporate%20Website/Documents/Publications/Axalta-2015-Global-Color-Popularity-Report.pdf

[11] https://scikit-learn.org/stable/modules/cross_validation.html

[12] https://www.kaggle.com/datasets/syedanwarafridi/vehicle-sales-data/data