CAR AUCTION PRICE PREDICTION

Simon Coessens
Student BDMA
simon.coessens@estudiantat.upc.edu

Rana Islek
Student BDMA
rana.islek@estudiantat.upc.edu

Professor : Marta Arias Vicente

June 3, 2024

ABSTRACT

This project explores the application of machine learning techniques to predict the final prices of used cars at auctions. Our goal is to develop a robust price prediction model for used car auctions using the "Used Car Auction Prices" dataset from Kaggle. The dataset comprises a comprehensive collection of vehicle attributes and auction details, providing a rich resource for analysis.

1 Introduction

The primary goal of this project is to develop a machine learning model that can accurately predict the final prices of used cars at auctions. This involves building a price prediction model based on auction listing attributes.

Our objective is to establish a robust model that forecasts final auction prices using various vehicle attributes. The successful implementation of this model will not only improve price prediction accuracy but also offer valuable insights into the factors influencing used car auction prices.

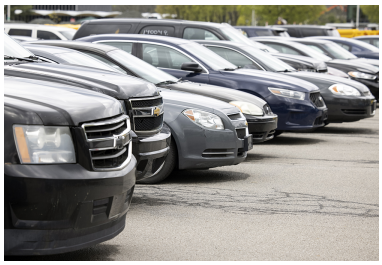

Figure 1: Used Car Auction Prices dataset from Kaggle.

1.1 Data Available

For this project, we utilize the "Used Car Auction Prices" dataset sourced from Kaggle. This dataset contains historical car auction sales prices scraped from various internet sources and was collected in 2015. It is a substantial dataset with 558,811 rows and 16 columns, offering a comprehensive collection of information relevant to used car auctions. Key attributes in the dataset include:

- **Year**: The year of the auction sale.
- **Make**: The manufacturer of the vehicle (e.g., Ford, Chevrolet).
- **Model**: The specific model of the vehicle (e.g., Altima, F-150).
- **Trim**: The trim level of the vehicle.

- **Body**: The body type of the vehicle (e.g., Sedan, SUV).
- **Transmission**: The type of transmission (e.g., automatic, manual).
- **VIN**: The vehicle identification number.
- **State**: The state where the auction took place.
- **Condition**: The condition of the vehicle.
- **Odometer**: The mileage of the vehicle.
- **Color**: The color of the vehicle.
- **Interior**: The interior color of the vehicle.
- **Seller**: The entity selling the vehicle.
- **MMR**: The Manheim Market Report value, which is a valuation metric.
- **Selling Price**: The final selling price at the auction.
- **Sale Date**: The date of the sale.

This rich dataset provides the foundational information necessary to train and test our predictive models. Data preprocessing, including cleaning and feature engineering, is a critical step to ensure the quality and relevance of the inputs to our models.

1.2 Additional Information Used

We utilized locational information of US states to determine the state for each sale in the dataset. This was particularly useful in enriching our dataset and improving the accuracy of our models by providing additional context regarding regional market trends and conditions.

Additionally, we supplemented missing information in the dataset by leveraging external knowledge available online about various cars and their brands. This involved filling in gaps for specific attributes using credible sources to ensure our dataset was as comprehensive and accurate as possible.

2 Previous Work

In our research, we examined two notable previous works related to predicting auction prices for used cars. The first work provided a comprehensive exploration and preparation of the dataset, offering valuable insights into data cleaning and feature engineering. This work effectively demonstrated how to handle missing values, outliers, and various transformations to improve model performance. The detailed exploration set a solid foundation for subsequent modeling efforts. This work can be found here Car Auction: Data Cleansing and Insight.

The second work we reviewed implemented a regression model on the data to predict the final auction prices. While the approach and methodology were solid, we noted that the model included the 'mmr' variable (Manheim Market Report value) as an input feature. The 'mmr' variable represents an estimated price, which, in real-life scenarios, is not available prior to the auction since it essentially serves as the target variable we aim to predict. Developing a model without the 'mmr' variable is more interesting and valuable, as it avoids information leakage and aligns more closely with practical application. This approach ensures that the model remains realistic and applicable in real-world scenarios where the 'mmr' is not available beforehand. You can find this work Regression previous work).

By addressing these limitations and focusing on realistic data inputs, our project aims to develop a more practical and applicable machine learning model for predicting used car auction prices.

3 Data Exploration

The data exploration phase is crucial to understand the dataset's structure, identify patterns, and prepare the data for modeling. The following steps were taken during the data exploration process:

3.1 Loading and Cleaning the Data

The initial step involved loading the dataset and performing basic cleaning operations. As a summary, the dataset has 10301, 10399, 10651, 13195, 653353, 11794, 94, 749 and 749 null values in "make", "model", "trim", "body",

"transmission", "condition", "odometer", "color" and "interior" columns respectively. We investigated them one by one and declared unlogical values as "outliers". Missing values were handled appropriately, and irrelevant columns were dropped. For instance, the 'vin' column, which represents the vehicle identification number, was not considered useful for prediction and was excluded from further analysis. We also made sure about "word uniformity" for string values and dropped duplicate values.

3.2 Feature Engineering

Several new features were created to enhance the dataset and provide more predictive power to the model:

- **Body Information of the Car**: We checked the unique values in the "body" column. We see that several values are similar, therefore we applied the string manipulation in the body column to uniform the strings that have the same references (e.g. "COUPE and "KOUP" both should reference "Coupe").

- **Brand Information of the Car**: Similarly, we checked the unique values in the "make" column. We observed that several values are the same brand but written differently like we just saw in the "body" column, therefore we manipulated the data to uniform them (e.g. "MERCEDES-B" and "MERCEDES" both should reference "MERCEDES-BENZ").

- **Locational Information**: The dataset included the 'state' column indicating the US state where each sale occurred. This information was used to enrich the dataset with regional trends and conditions. We grouped each brand depending on the company's country of origin and created a new column named as "made_in".

- **Time-related Information**: From the 'saledate' column, multiple time-related features were extracted:

  - **Day of the Week**: Extracted as a categorical feature to capture weekly patterns in auction prices.
  - **Is Weekend**: A binary feature indicating whether the sale occurred on a weekend.
  - **Hour of the Day**: Extracted to analyze the time of day when sales occurred.
  - **Month and Year**: Combined feature to observe monthly and yearly trends in the data.

- **Car Age Information**: Calculated as the difference between the year of production and the year of sale. Negative values were identified and removed to ensure logical consistency.

- **Is Sold Below MMR**: A binary feature indicating whether the sale price was below the Manheim Market Report (MMR) value, so we can easily compare the price with the market.

- **Longitude and Latitude Information**: We imported a new dataset to get information about the longitude and the latitude coordinates of cities. Since some states do not have the longitude and latitude information, we have done some manual manipulation. Then we merged two datasets to have an enriched final dataset.

3.3 Exploratory Data Analysis (EDA)

The EDA phase included various analyses to uncover insights and patterns in the data:

- **Distribution of Selling Prices**: Visualizations such as histograms were used to understand selling price distribution. As shown in the figure below, the average selling price of the car is approximately $12,000.

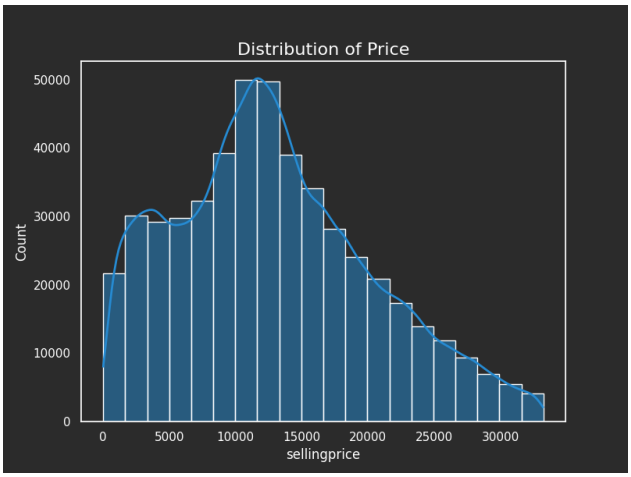

Figure 2: Distribution of Price

- **Sales below MMR based on Car Production Year**: As shown in the plot below, the percentage of sales tends to decrease as the year of production increases.

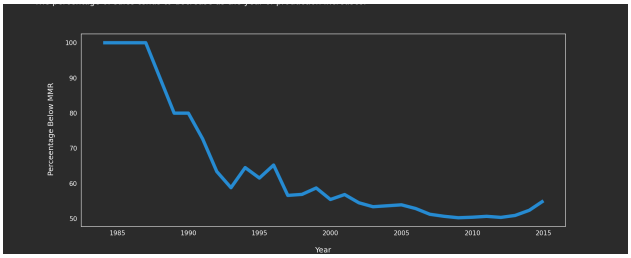

Figure 3: Sales below MMR based on Car Production Year

- **Total Sales based on Day and Hour**: As shown in the heatmap below, most transactions occur on weekdays and between 1AM and 3AM.

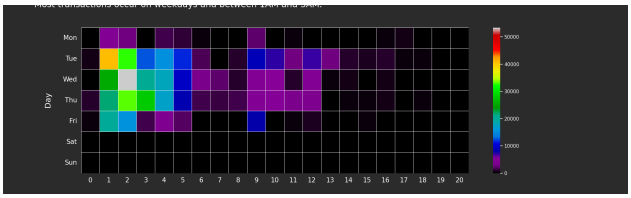

Figure 4: Heatmap for Total Sales based on Day and Hour

- **Total Transaction based on Car Age**: After checking the plot below, one can investigate that cars aged 4 years and under have the highest interest seen from the number of transactions which is quite high, then the number of transactions decreases as the car gets older.

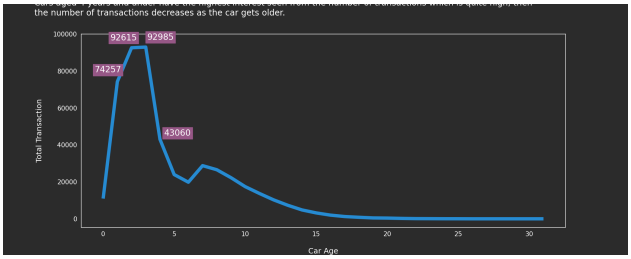

Figure 5: Total Transaction based on Car Age

As we observed from the plot, buyers are more likely to buy cars aged 4 years and under from a demand perspective. On the other hand, from a supply perspective, car owners tend to auction their vehicles despite their age.

We can clearly see that, cars aged 4 years and under have the highest interest as seen from the number of transactions which is quite high and then the number of transactions decreases as the age of the car increases.

The above plot shows an unstable transaction history. We need to remind the fact that the data is the result of scrapping from various resources, so that the monthly transaction history cannot be used as a benchmark.

- **Brand Analysis**: With the histogram below, the distribution of different car brands in the dataset was explored. It was found that Ford and Chevrolet were the most common brands, with significant numbers of sales below the MMR value. Moreover, the top 10 brands represent more than 50% of total transactions.

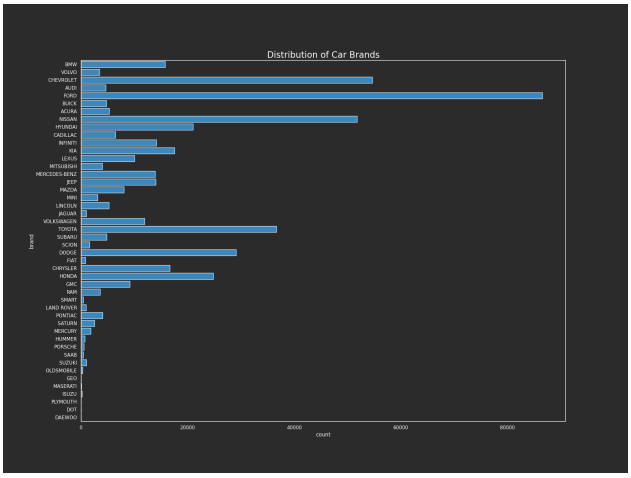

Figure 6: Distribution of Car Brands

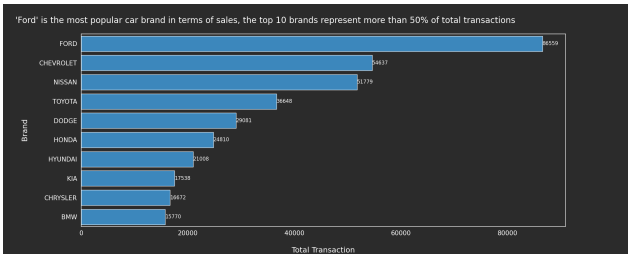

Figure 7: Top 10 Best Selling Brand

- **Body Type Analysis**: With the help of the histogram below, the popularity of different body types, such as Sedans and SUVs, was visualized and analyzed. Sedans and SUVs are drastically more popular than others, in terms of sales. This is also due to the increasing number of brands using these two body types as the basis for the cars they make because of their popularity.

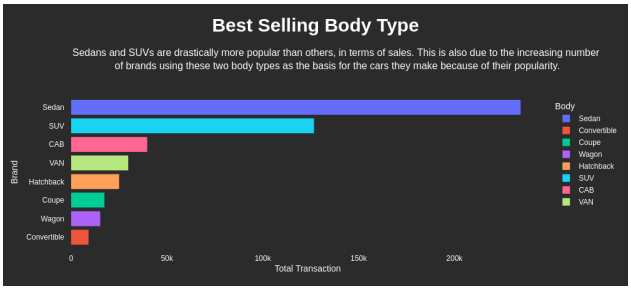

Figure 8: Best Selling Body Type

- **Correlation of Numerical Features**: Looking at the below numerical features analysis, we see that car_age, odometer, and MMR are strong features to predict selling price. So we'll continue with them in our models except MMR. Since MMR is the Market Record, the correlation is very high by its nature.

Figure 9: Correlation Heatmap

- **Seller Analysis based on States**: The diversity of sellers was examined. Most of the auctioned cars come from California and Florida.

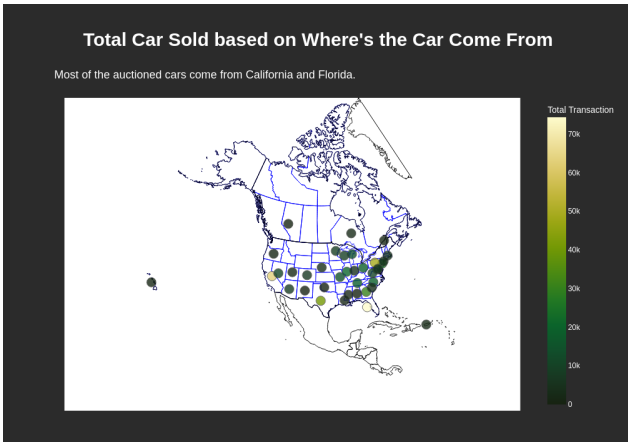

Figure 10: Total Car Sold based on States

3.4 Key Findings

From the data exploration, several key findings emerged:

- Only about 48% of all cars at auction sell at or above the MMR price. Older cars are more likely to sell below the MMR.
- Most transactions tend to occur on weekdays early in the morning, likely because many sellers are companies that close on weekends. Also online auction closings are quite often done at midnight or early in the morning.
- Sedans and Convertibles are the two most popular body types, with Ford, Chevrolet, and Nissan being the most popular brands.
- Around 245 models have a high percentage of sales below MMR, with Ford and Chevrolet being the largest contributors.
- After analysis we also see that most of the features are not really relevant for the selling price prediction, so it is more appropriate to drop them before proceeding to prediction tasks.
- Since MMR is strongly correlated with selling price because of its definition, we will drop that feature to have a more useful selling price prediction model.

These insights from the data exploration phase provided a solid foundation for building and refining the predictive models.

4 Modeling methods

In the following sections, we will present the modeling methods explored in our project. It is important to note that we experimented with multiple configurations and tuning options, and we will discuss only those configurations that yielded the best model performance.

Models were saved to disk after training to enable future use, including deployment and further evaluation:

- **Model Saving**: Each trained model, regardless of its type, was saved using the Joblib library. This method ensures compatibility and efficiency in loading models for subsequent operations.

4.1 General Workflow for the model training and evaluation

We follow a structured approach to ensure systematic development and evaluation of predictive models. This general workflow is applied consistently across different models to maintain standardization and comparability of results.

4.1.1 Data Preprocessing

Data preprocessing is a critical first step in our workflow:

- **Dropping Irrelevant Columns**: We remove columns that do not contribute to the predictive power of the models. This includes metadata or redundant information.
- **Encoding Categorical Variables**: We apply one-hot encoding to categorical variables to convert them into a format suitable for machine learning models. This ensures that categorical data is properly utilized in the models.
- **Scaling Numerical Features**: Numerical features are scaled to have zero mean and unit variance. This standardization is crucial for models that are sensitive to the scale of the input data, such as linear regression and KNN.

4.1.2 Model Training

Model training is performed with a consistent approach to ensure that all models are comparably evaluated:

- **Feature and Target Separation**: We separate the features (independent variables) from the target variable (dependent variable), which is typically the value we aim to predict.
- **Splitting the Data**: The data is typically split into training and testing sets, with 80% used for training and 20% reserved for testing. This split may vary slightly depending on the specific requirements of the project.

- **Model Initialization and Training**: Each model, specific to its algorithmic requirements, is initialized and trained on the training dataset.

4.1.3 Model Evaluation

- **Evaluation Metrics**: We use the following metrics: Mean Absolute Error (MAE), Mean Squared Error (MSE), Root Mean Squared Error (RMSE), and R-squared (R2).
- **Visualizing Results**: We generate plots such as scatter plots of actual vs. predicted prices and residuals to visually assess model performance and identify potential issues with model assumptions.
- **Cross-Validation**: When applicable, we employ cross-validation techniques to ensure that our evaluation is robust and not overly dependent on the partitioning of the data.

4.2 Linear Regression

Linear regression is the most straightforward method for predicting this data, given that there are multiple numerical variables that likely exhibit a linear relationship with the selling price. This section outlines the steps taken, from data preprocessing to model evaluation.

4.2.1 Model Training

The dataset was split into training and testing sets. The following steps were taken:

- **Splitting the Data**: The data was split into training (80%) and testing (20%) sets.
- **Model Initialization and Training**: A linear regression model was initialized and trained on the training set.
- **Model Adjustment**: To ensure all predicted prices are non-negative, as prices cannot be below zero, we modified our model to correct predictions accordingly. The necessary adjustment is implemented as shown in Listing 1.

```
y_pred[y_pred < 0] = 0
```

Listing 1: Ensuring Non-negative Price Predictions

4.2.2 Model Evaluation

The performance of the linear regression model was evaluated using the testing set. The following metrics were used:

- **MAE (Mean Absolute Error)**: At 1596.61, the model shows a moderate average error in prediction, representing a direct measure of error in the same units as the target variable.
- **MSE (Mean Squared Error)**: The high MSE of 4960402.63 indicates sensitivity to large errors, which might suggest outlier influence or model variance issues.
- **RMSE (Root Mean Squared Error)**: An RMSE of 2227.20 highlights the typical error magnitude, useful in evaluating error distribution and its impact on model reliability.
- **R^2 (R-squared)**: With an R^2 of 90.96%, the model explains a significant portion of the variance, indicating strong predictive power.

4.2.3 Visualizing Results

To understand the performance of the model visually, we show several plots. Additionally some plots were created to check the assumptions for the linear model.

- **Actual vs Predicted Prices**: The scatter plot illustrates the relationship between the actual selling prices and the predicted prices by our model. Each point represents an individual prediction, with the actual values on the x-axis and the predicted values on the y-axis. The close clustering of points around the diagonal red line suggests a strong correlation, indicating that our model generally predicts values close to the actual prices. However, the spread increases for higher value ranges, implying potential model inaccuracies or limitations in predicting higher-priced items.

Figure 11: Scatter plot of actual versus predicted selling prices, demonstrating the model's accuracy across different price ranges.

- **Distribution of Residuals**: A histogram of the residuals to check for normality. The residuals seem to follow a normal distribution what is expected for a good linear model.

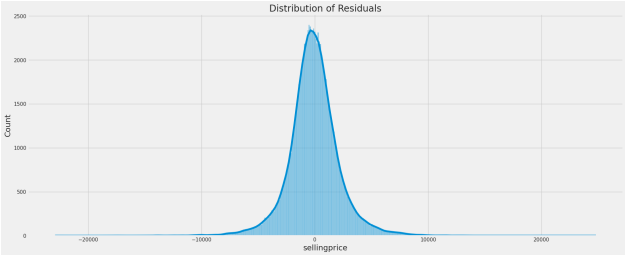

Figure 12: Distribution of Residuals

- **Q-Q Plot**: A Q-Q plot to further assess the normality of residuals. Here we also see that most of the data follows the normality assumption, we only have deviations at the tails.

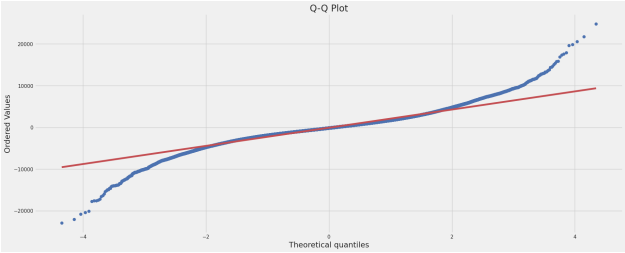

Figure 13: Q-Q Plot of Residuals

4.2.4 Model Evaluation Using Cross-Validation

We used cross-validation to verify the robustness and predictive power of our Linear Regression model. We used a 5-fold cross-validation, yielding the following metrics across all folds:

- **Mean Absolute Error (MAE)**: The model achieved an average MAE of 1716.79, indicating the average absolute error in the model's predictions.
- **Mean Squared Error (MSE)**: With an MSE of 5689755.91, this metric reflects the average squared discrepancies between the predicted and actual values.
- **Root Mean Squared Error (RMSE)**: The RMSE was calculated to be 2385.32, providing a measure of the average magnitude of the predictive error.

- **R-squared (R2)**: The model explained 89.64% of the variance in the dataset, suggesting a strong fit to the data.

These results reinforce the reliability of the model and make it more probable that the model performs well on real world data.

5 Tree-Based Models

The second category of models we looked at are the Decision Tree models. A Decision Trees are well-suited for our dataset as it might be able to capture some relation in the data, that is not linear, and was not captured by the linear model.

5.1 Overview

This section presents the performance outcomes of three tree-based predictive models: Decision Tree Regressor, Random Forest Regressor, and XGBoost. These models were evaluated to predict the selling prices of used cars at auction based on a dataset processed with uniform data preprocessing techniques.

5.2 Data Preprocessing

All models utilized a consistent preprocessing workflow:

- Removing irrelevant columns such as *year*, *vin*, and *mmr*.
- One-hot encoding of categorical features like *brand* and *model*.
- Standardizing numerical features to have zero mean and unit variance.

5.3 Model Evaluations

Each model was trained on an 80-20 train-test split.

5.3.1 Decision Tree Regressor

- **Mean Absolute Error (MAE)**: 1503.64
- **Mean Squared Error (MSE)**: 5252495.88
- **Root Mean Squared Error (RMSE)**: 2291.83
- **R-squared (R2)**: 0.905

We see in Figure 14 that the decision tree captures the relation between the variables in the data quite well, Most predicitions lie close to the real value, with some outliers.

Figure 14: Decision Tree Regressor results

5.3.2 Random Forest Regressor

- **Mean Absolute Error (MAE)**: 2540.92
- **Mean Squared Error (MSE)**: 12480768.95
- **Root Mean Squared Error (RMSE)**: 3532.81
- **R-squared (R2)**: 0.774

5.3.3 XGBoost Regressor

- **Mean Absolute Error (MAE)**: 1972.14
- **Mean Squared Error (MSE)**: 7341382.14
- **Root Mean Squared Error (RMSE)**: 2709.50
- **R-squared (R2)**: 0.867

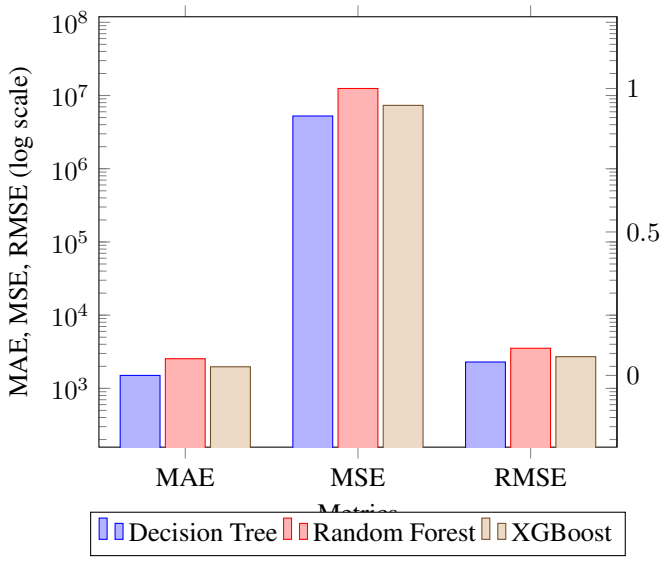

Figure 15: Comparison of Decision Tree, Random Forest, and XGBoost Model Metrics

5.4 Conclusion

As illustrated in Figure 15, the Decision Tree Regressor demonstrates the highest R-squared value and the lowest error metrics, indicating superior performance in fitting to the dataset used. Although we considered implementing cross-validation to enhance the reliability of our findings, the extensive runtime required by these algorithms made it impractical but this would have made for a more reliable conclusion.

6 KNN Model

Altough quite resource intensive the KNN model can provide good performance on prediction datasets. In this section we explore this model.

6.0.1 Model Training

- **Training and Testing Split**: The dataset was divided into training (90%) and testing (10%) sets to evaluate the model's performance accurately. Here we choose a different distribution because the testing takes a significant amount of time for knn.
- **Model Pipeline Creation**: A pipeline comprising the preprocessing steps and the KNN regressor was created. The KNN was configured with 3 neighbors, balancing between overfitting and underfitting.

6.0.2 Model Evaluation

The KNN model's performance was evaluated on the test set using several metrics to assess its predictive accuracy and generalization capability. Here again the computation was to heavy to do cross validation.

- **Mean Absolute Error (MAE)**: Achieved an MAE of 1439.57, indicating the average deviation of the predicted prices from the actual prices.
- **Mean Squared Error (MSE)**: Recorded an MSE of 4374680.88, highlighting the average of the squares of the errors.
- **Root Mean Squared Error (RMSE)**: An RMSE of 2091.57 reflects the standard deviation of the residuals, providing insight into the typical error magnitude.
- **R-squared (R2)**: An R^2 of 0.920 shows that approximately 92% of the variance in the auction prices is predictable from the features, indicating a high level of model effectiveness.

6.0.3 Conclusion

We can conclude that the KNN model performs very well for this prediction dataset.

7 SVM Model

7.1 Data Preprocessing

The preprocessing steps we took:

- **Encoding Categorical Variables**: Important categorical features like *brand* and *model* were one-hot encoded, ensuring that these nominal variables are suitably expressed for the SVM's use.
- **Scaling Numerical Features**: We standardized numerical variables because this is crucial for the SVM algorithm since it depends on the calculation of distances between data points.

7.2 Model Training

A pipeline was constructed to seamlessly integrate preprocessing steps with the SVM model training:

- **Training and Testing Split**: The dataset was split into a training set (80%) and a testing set (20%), with random state control for reproducibility.
- **SVM Pipeline Configuration**: The SVM model was set up within a pipeline that includes both the preprocessor and the SVM with a linear kernel to manage both linear and non-linear relationships in the data.

7.3 Model Evaluation

Following training, the SVM model's performance was evaluated using standard metrics to assess its predictive accuracy and efficacy:

- **Mean Absolute Error (MAE)**: The model reported an MAE of 2377.70, indicating the average magnitude of the errors in predictions.
- **Mean Squared Error (MSE)**: With an MSE of 11583656.04, this metric signifies the average of the squares of the prediction errors.
- **Root Mean Squared Error (RMSE)**: An RMSE of 3403.48, which gives an idea of the magnitude of the typical prediction error.
- **R-squared (R2)**: The R^2 value of 0.789 indicates that approximately 78.9% of the variance in the dependent variable is predictable from the independent variables.

As observed, this SVM model under performs compared to the previously described models. The high dimensionality of the input may overly complicate the model, hindering its ability to effectively capture the underlying data patterns. Further optimization, perhaps through the use of a different kernel function, might improve results.

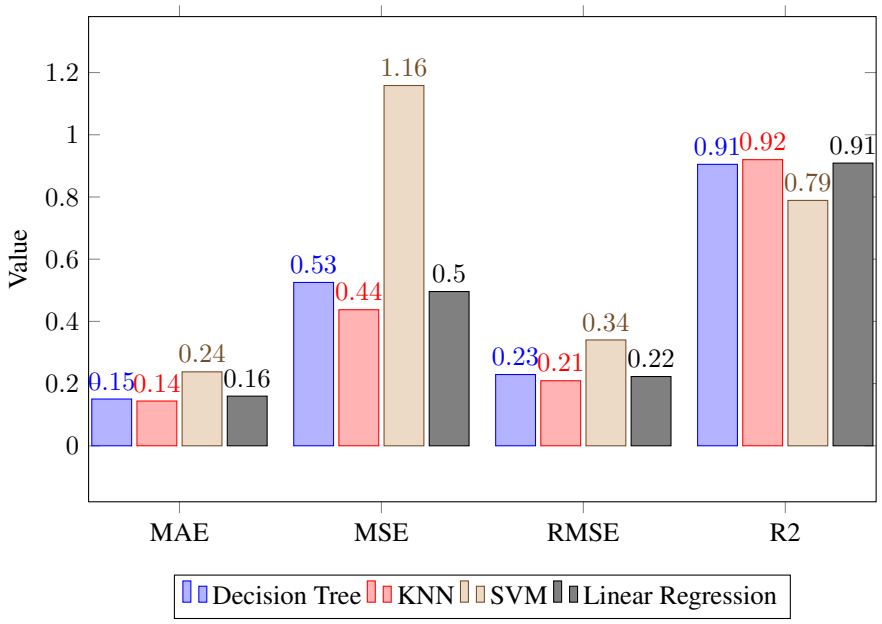

Figure 16: Comparison of MAE, MSE, RMSE, and R^2 for Decision Tree, KNN, SVM, and Linear Regression Models

8 Conlusion

This comprehensive comparison graphically illustrates the strengths and weaknesses of each model in our project. The Decision Tree and KNN models exhibit strong performance, particularly in terms of R^2, demonstrating their ability to accurately predict the variability of the dataset. The Linear Regression model, while not as robust as KNN or Decision Tree in R^2, still performs competently, showing good balance across all metrics. The SVM, however, struggles with higher error metrics and lower R^2.

9 Further Work

For further work it would be interesting to explain which variables have the highest impact on the result. For the Linear Regression model we tried using the `statsmodel` library because it has built in functions that can work with the models coefficients. But using this model gave us a memory error multiple times so we did not continue this path. For the other models as well it could be interesting to look into the explain-ability for each model.