PROJECT REPORT

ON

Auto Price Pro: Predicting Car Prices with Machine Learning

Group G

Prabhsimran Singh(c0905791)

Vidhamjot Kaur(c0909093)

Anmol Singh Paman(c0908043)

Gurpreet Kaur(c0908454)

Sarabjot Singh(c0909102)

Course Name & Number: 2024W-T2 BDM 2053-Big Data Algorithms and Statistic 02(DSMM Group 2)

Professor's Name: Darcy Gratton

Due Date: 9th April 2024

Introduction

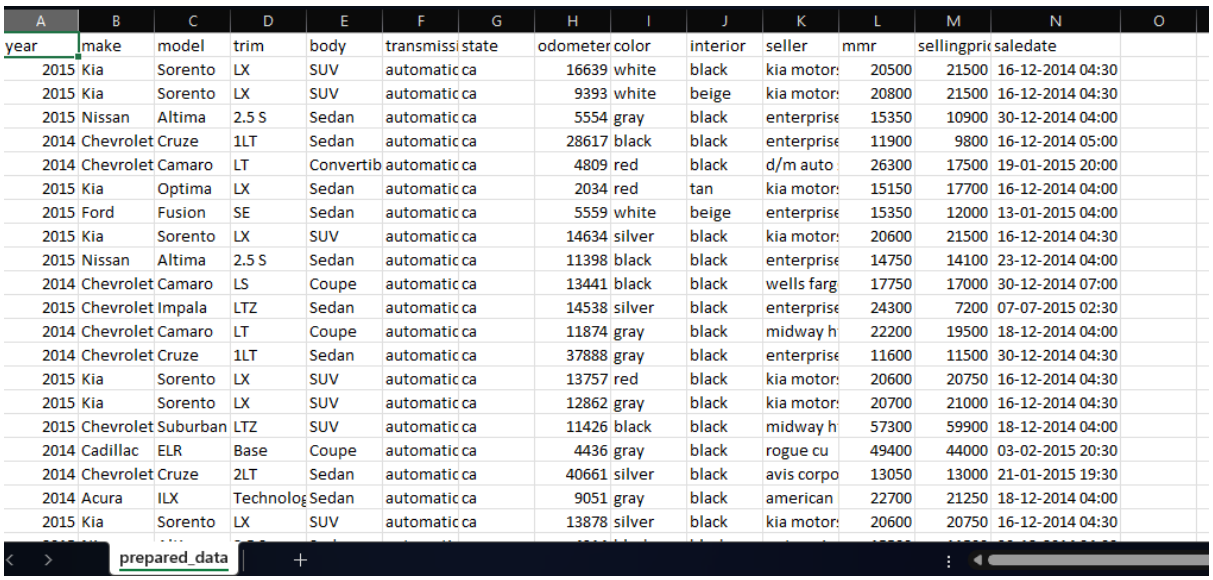
This project aims to develop a car price prediction system using XGBoost, which accurately estimates used car prices based on various features. The primary target audience includes car sellers, car buyers, and industry professionals such as dealerships, appraisers, and insurance companies. Accurate price predictions empower buyers to make informed decisions, avoid overpaying, and negotiate better deals. Sellers can set competitive prices, attract potential buyers, and optimize their profits. Industry professionals, such as dealerships, appraisers, and insurance companies, can utilize the system to determine trade-in values, set inventory prices, and assess vehicle worth accurately. By delivering accurate and reliable predictions, this project promotes transparency, facilitates fair transactions, and enhances decision-making in the automotive market. It has the potential to improve customer satisfaction, streamline operations, and contribute to a trustworthy and efficient industry.

Objective Statement:

The aim of this project is to create an accurate car price prediction system using XGBoost that takes into account multiple features to determine the used car's pricing. In order to help buyers, sellers, and industry professionals make wise decisions, set competitive prices, maximise profit margins, and improve operational efficiency in the automotive market, the system is designed to offer a dependable tool.

Data Collection and Exploratory Data Analysis

Sources of Data: The data for this car price prediction project was collected from Kaggle. The primary source was a dataset obtained from a reputable online marketplace for used cars. The dataset included various attributes such as make, model, year, odometer reading, MMR (Market Mirror Ratio), interior color, exterior color, transmission, trim, and body type. Additionally, external datasets were utilized to enhance the predictive power of the model.

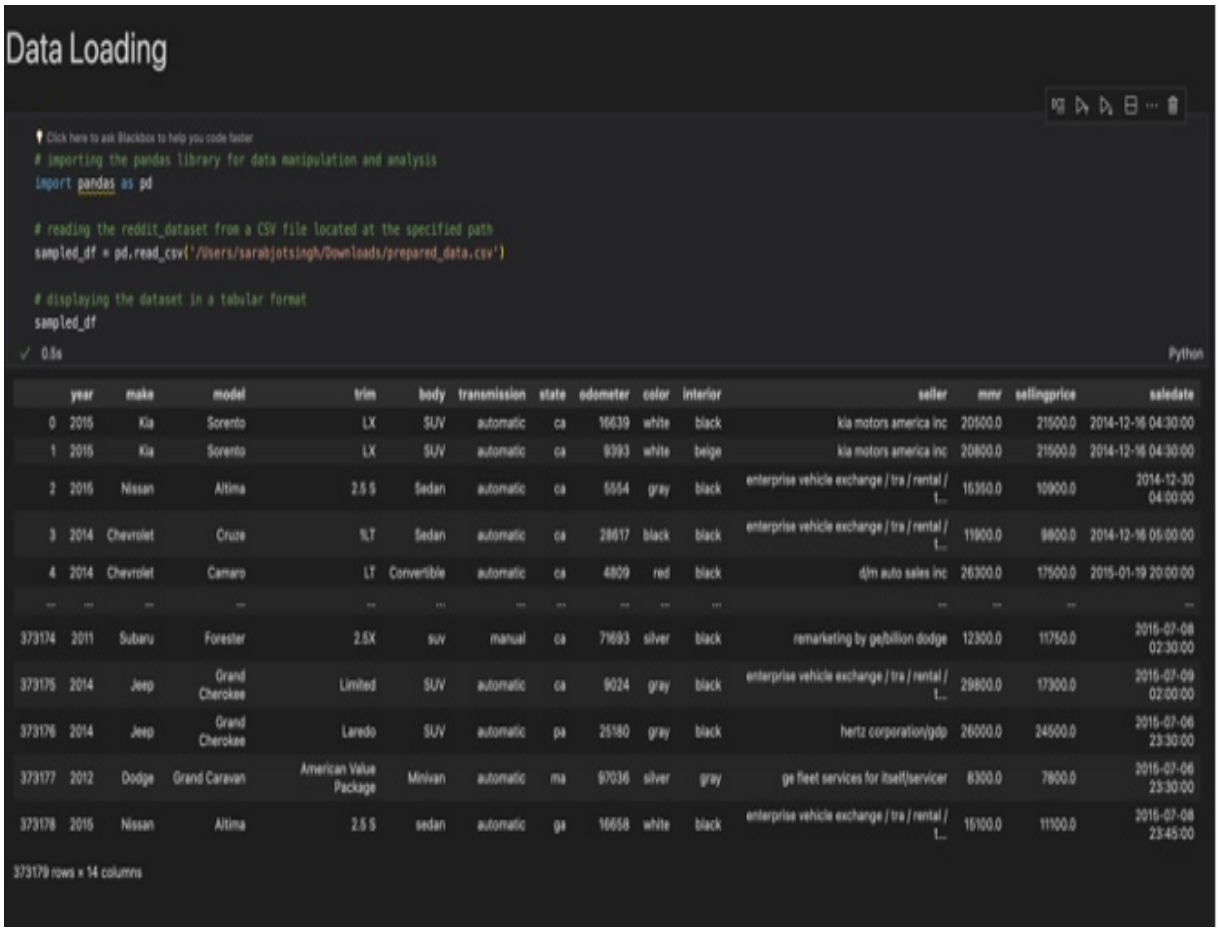


A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
year	make	model	trim	body	transmissi	state	odometer	color	interior	seller	mmr	sellingpric	saledate	
2015	Kia	Sorento	LX	SUV	automatic	ca	16639	white	black	kia motors	20500	21500	16-12-2014 04:30	
2015	Kia	Sorento	LX	SUV	automatic	ca	9393	white	beige	kia motors	20800	21500	16-12-2014 04:30	
2015	Nissan	Altima	2.5 S	Sedan	automatic	ca	5554	gray	black	enterprise	15350	10900	30-12-2014 04:00	
2014	Chevrolet	Cruze	1LT	Sedan	automatic	ca	28617	black	black	enterprise	11900	9800	16-12-2014 05:00	
2014	Chevrolet	Camaro	LT	Convertib	automatic	ca	4809	red	black	d/m auto	26300	17500	19-01-2015 20:00	
2015	Kia	Optima	LX	Sedan	automatic	ca	2034	red	tan	kia motors	15150	17700	16-12-2014 04:00	
2015	Ford	Fusion	SE	Sedan	automatic	ca	5559	white	beige	enterprise	15350	12000	13-01-2015 04:00	
2015	Kia	Sorento	LX	SUV	automatic	ca	14634	silver	black	kia motors	20600	21500	16-12-2014 04:30	
2015	Nissan	Altima	2.5 S	Sedan	automatic	ca	11398	black	black	enterprise	14750	14100	23-12-2014 04:00	
2014	Chevrolet	Camaro	LS	Coupe	automatic	ca	13441	black	black	wells farg	17750	17000	30-12-2014 07:00	
2015	Chevrolet	Impala	LTZ	Sedan	automatic	ca	14538	silver	black	enterprise	24300	7200	07-07-2015 02:30	
2014	Chevrolet	Camaro	LT	Coupe	automatic	ca	11874	gray	black	midway h	22200	19500	18-12-2014 04:00	
2014	Chevrolet	Cruze	1LT	Sedan	automatic	ca	37888	gray	black	enterprise	11600	11500	30-12-2014 04:30	
2015	Kia	Sorento	LX	SUV	automatic	ca	13757	red	black	kia motors	20600	20750	16-12-2014 04:30	
2015	Kia	Sorento	LX	SUV	automatic	ca	12862	gray	black	kia motors	20700	21000	16-12-2014 04:30	
2015	Chevrolet	Suburban	LTZ	SUV	automatic	ca	11426	black	black	midway h	57300	59900	18-12-2014 04:00	
2014	Cadillac	ELR	Base	Coupe	automatic	ca	4436	gray	black	rogue cu	49400	44000	03-02-2015 20:30	
2014	Chevrolet	Cruze	2LT	Sedan	automatic	ca	40661	silver	black	avis corpo	13050	13000	21-01-2015 19:30	
2014	Acura	ILX	Technolog	Sedan	automatic	ca	9051	gray	black	american	22700	21250	18-12-2014 04:00	
2015	Kia	Sorento	LX	SUV	automatic	ca	13878	silver	black	kia motors	20600	20750	16-12-2014 04:30	

prepared_data

Data Preprocessing Steps:

Handling Missing Values: The first step was to identify and handle missing values in the dataset. Missing data can adversely affect the performance of machine learning models. Different strategies were employed depending on the nature and extent of missing values. For numerical features, missing values were often imputed using techniques such as mean, median, or regression-based imputation. Categorical variables with missing values were either imputed with the mode or treated as a separate category.



```
Click here to ask Blackbox to help you code faster
# checking for missing values in each column
missing_values_counts = sampled_df.isnull().sum()

# printing the number of missing values for each column
print("Number of missing values in each column:")
print(missing_values_counts)
```

✓ 0.0s

```
Number of missing values in each column:
year            0
make            0
model           0
trim            0
body            0
transmission    0
state           0
odometer        0
color           0
interior        0
seller          0
mmr             0
sellingprice    0
saledate        0
dtype: int64
```

Encoding Categorical Variables: Categorical variables were transformed into numerical representations for model compatibility. One-hot encoding or label encoding techniques were used depending on the cardinality and nature of the categorical features. One-hot encoding creates binary columns for each category, while label encoding assigns a unique numerical label to each category.

```
1  # Perform one-hot encoding for textual features
2  encoder = OneHotEncoder(handle_unknown='ignore')
3  X_textual_encoded = encoder.fit_transform(X_textual)
```

✓ 0.2s

Scaling Numerical Features: To ensure that numerical features were on a similar scale, they were often standardized or normalized. Standardization involves transforming the data to have zero mean and unit variance, while normalization scales the data to a specified range, such as [0, 1]. This step helps prevent features with larger magnitudes from dominating the model's learning process.

```
1  # Perform standard scaling for numerical features
2  scaler = StandardScaler()
3  X_numerical_scaled = scaler.fit_transform(X_numerical)
```

✓ 0.0s

Challenges Encountered:

Data Quality and Consistency: One of the common challenges during data collection was ensuring data quality and consistency. The dataset obtained from the online marketplace may contain errors, outliers, or inconsistencies in the recorded information. Careful data cleaning and validation were performed to address these issues and ensure the accuracy of the dataset.

Training model to interpret textual data: Dealing with categorical variables can be challenging, especially when they have a large number of unique categories. It requires careful consideration of encoding techniques and potential dimensionality issues. Techniques like feature hashing or dimensionality reduction methods may be employed to address these challenges.

Missing Data Imputation: Missing data is a common occurrence in real-world datasets. Determining the appropriate imputation strategy for missing values requires careful analysis and consideration. Different imputation techniques may be applied depending on the missing data patterns and the impact of missing values on the overall dataset.

Standard scaling for numerical values: Because the features in the automobile price prediction project have different scales and ranges, standard scaling for numerical data is essential. Features with distinct value ranges from others include MMR and odometer reading. It is required to standardise the numerical values because XGBoost is sensitive to feature scales. If this isn't done, the model's learning process will be dominated by features from bigger scales, which can result in biassed predictions. Standard scaling improves the effectiveness and accuracy of the car price prediction model by ensuring fair comparison and equal weighting of all features.

Feature Engineering

The car price prediction model utilizes several key features to estimate the prices of used cars. These features include:

- **Make and Model:** The make and model of a car provide important information about its brand, reputation, and market demand. Different car brands and models may have varying price ranges due to factors such as popularity, reliability, and performance.
- **Year:** The year of the car indicates its age and can significantly impact its price. Generally, newer cars tend to have higher prices compared to older ones.
- **Odometer Reading:** The odometer reading represents the mileage or distance traveled by the car. Lower mileage is often associated with higher prices, as it implies less wear and tear and potential longevity.
- **MMR (Market Mirror Ratio):** The Market Mirror Ratio is a metric that compares a car's price to its estimated market value. It provides a relative measure of the car's pricing competitiveness in the market.
- **Interior and Exterior Color:** The color of a car's interior and exterior can influence its desirability and price. Certain colors may be more popular or have a higher demand, affecting the perceived value of the vehicle.
- **Transmission:** The type of transmission, whether automatic or manual, can impact the price of a car. Automatic transmissions are generally more common and preferred, leading to potential price differences.

- **Trim:** The trim level indicates the specific features, options, and upgrades present in a car. Higher trim levels often come with additional amenities, which can influence the price.
- **Body Type:** The body type refers to the structure and design of the car, such as sedan, SUV, hatchback, or coupe. Different body types may have different price ranges based on their utility, style, and market demand.

```
# Selecting features and target variable
textual_features = ['make', 'model', 'trim', 'body', 'color', 'interior']
numerical_features = ['year', 'odometer', 'mmr']
target_variable = 'sellingprice'
```
✓ 0.0s

To enhance the predictive power of the model, various feature transformations or combinations can be applied. Some common techniques include:

- **Polynomial Features:** Introducing polynomial features can capture non-linear relationships between variables. For example, creating squared or interaction terms between certain features may uncover hidden patterns and improve prediction accuracy.
- **Logarithmic or Exponential Transformations:** Applying logarithmic or exponential transformations to certain features can help normalize their distribution or capture exponential relationships, respectively.
- **Feature Scaling:** Scaling numerical features to a similar range, such as using standardization or normalization techniques, can prevent certain features from dominating the model's learning process.
- **Feature Interaction:** Creating new features by combining two or more existing features can capture synergistic effects and interactions that may affect car prices. For example, combining make and model to create a composite feature can provide additional information about brand-model combinations.

Model Selection and Training

- **Choice of XGBoost:** XGBoost (Extreme Gradient Boosting) was chosen as the modeling algorithm for several reasons. XGBoost is a powerful and popular gradient boosting algorithm known for its high performance and effectiveness in handling structured data. It excels in capturing complex non-linear relationships, handling missing values, and managing a large number of features. XGBoost also incorporates regularization techniques to prevent overfitting and provides feature importance analysis, which aids in understanding the impact of different features on the model's predictions. Due to these advantages, XGBoost is well-suited for car price prediction tasks where the dataset typically consists of diverse features with non-linear relationships.
- **Model Training Process:** The model training process involved several steps. Firstly, the dataset was split into training and testing sets. The training set was used to train the XGBoost model, while the testing set was reserved for evaluating its performance. The train-test split is crucial to assess the model's ability to generalize to unseen data and avoid overfitting.
- **Cross-Validation Techniques:** To further evaluate the model's performance and mitigate issues related to the train-test split, cross-validation techniques were employed. K-fold cross-validation, where the training set is divided into k subsets (folds), was used. The model was

trained and evaluated k times, with each fold serving as the validation set once while the remaining folds were used for training. This process ensures that the model's performance is robust across different subsets of the training data.

- **Hyperparameter Tuning:** Hyperparameter tuning was performed to optimize the model's performance. Techniques such as grid search or random search were applied to explore different combinations of hyperparameters. Hyperparameters control the behavior of the model and affect its performance. Important hyperparameters in XGBoost include the learning rate, maximum depth of trees, subsampling rate, regularization parameters, and the number of boosting rounds. By systematically searching through different hyperparameter values, the best combination that maximizes the model's performance, as measured by a chosen evaluation metric (e.g., mean squared error or root mean squared error), was selected.
- **Selected Hyperparameters:** The specific hyperparameters selected depend on the dataset and the results of the hyperparameter tuning process. The optimal combination may vary, but some commonly tuned hyperparameters in XGBoost include the learning rate, maximum depth of trees, and regularization parameters such as gamma and lambda. The selected hyperparameters aim to strike a balance between model complexity and overfitting, ensuring the best performance on unseen data.

Model Evaluation

- **Evaluation Metrics**: The performance of the car price prediction model was assessed using various evaluation metrics commonly employed for regression tasks. Some of the commonly used metrics include:
  a. **Mean Squared Error (MSE):** MSE measures the average squared difference between the predicted prices and the actual prices. It provides a measure of the model's overall prediction accuracy.
  b. **Root Mean Squared Error (RMSE):** RMSE is the square root of the MSE, making it more interpretable in the original unit of the target variable. It represents the average absolute difference between the predicted and actual prices.
  c. **R-squared (R2):** R-squared measures the proportion of the variance in the target variable that is explained by the model. It indicates how well the model fits the data, with higher values indicating a better fit
  d. **MAE:** In this project using XGBoost, MAE (Mean Absolute Error) can be used as an evaluation metric. It will measure the average absolute difference between the predicted and actual car prices, providing a quantifiable measure of prediction accuracy.

```python
# Predictions on the test set
y_pred = xgb_model.predict(X_test)

# Calculate Mean Squared Error (MSE)
mse = mean_squared_error(y_test, y_pred)

# Calculate Root Mean Squared Error (RMSE)
rmse = np.sqrt(mse)

# Calculate Sum of Squared Residuals (SSR)
ssr = np.sum((y_pred - y_test) ** 2)

# Calculate Mean Absolute Error (MAE)
mae = mean_absolute_error(y_test, y_pred)

# Calculate R-squared (R2)
r2 = r2_score(y_test, y_pred)

print("Mean Squared Error (MSE):", mse)
print("Root Mean Squared Error (RMSE):", rmse)
print("Sum of Squared Residuals (SSR):", ssr)
print("Mean Absolute Error (MAE):", mae)
print("R-squared (R2):", r2)
```

✓ 0.1s

Results and Interpretation:

```
Mean Squared Error (MSE): 3056578.9048032835
Root Mean Squared Error (RMSE): 1748.3074400125636
Sum of Squared Residuals (SSR): 159278326729.2991
Mean Absolute Error (MAE): 1048.3503366994848
R-squared (R2): 0.9517205451302917
```

Mean Squared Error (MSE): MSE of 3056578.90 indicates an average squared difference of 3,056,578 between predicted and actual car prices, penalizing larger errors.

Root Mean Squared Error (RMSE): RMSE of 1748.31, the square root of MSE, measures average prediction error in original price units. A lower RMSE suggests model predictions have an average error of about 1,748.

Sum of Squared Residuals (SSR): SSR of 159278326729.30 represents the sum of squared differences between predicted and actual car prices, indicating the total residual error of the model.

Mean Absolute Error (MAE): MAE of 1048.35 measures the average absolute difference between predicted and actual car prices, capturing the average magnitude of errors regardless of their direction.

R-squared (R2): R2 of 0.9517 suggests the model explains about 95.17% of car price variance. Higher R2 values indicate better model fit to the data.

Feature Importance Analysis

Feature importance analysis using XGBoost can provide valuable insights into the factors that have the most significant impact on predicting car prices. Here's how the analysis can be conducted and interpreted:

- **Calculating Feature Importance:** XGBoost provides a built-in feature importance metric based on the improvement of the model's performance when using a particular feature. This metric quantifies the contribution of each feature in the model's decision-making process.
- **Visualizing Feature Importance:** The feature importance scores can be visualized using a bar plot or presented in a table. The features are ranked based on their importance scores, with higher scores indicating greater importance in predicting car prices.

Interpreting Insights: The feature importance analysis provides valuable insights into the factors that strongly influence car prices. By examining the important features, one can gain a better understanding of the key drivers affecting the pricing decisions in the used car market. Here are some possible insights:

a. **Odometer Reading:** If the feature importance analysis reveals that the odometer reading has a high importance score, it suggests that the mileage is a crucial factor affecting car prices. Lower mileage generally correlates with higher prices, as cars with lower mileage are often perceived as having less wear and tear and greater longevity.
b. **Year of the Car:** If the year of the car has a high importance score, it indicates that the age of the vehicle significantly impacts its price. Newer cars tend to command higher prices compared to older ones, assuming other factors remain constant.
c. **Make and Model:** If specific makes and models score high in terms of feature importance, it highlights the influence of brand reputation, market demand, and perceived value associated with certain car manufacturers and models.
d. **Trim Level and Features:** If features related to the trim level or specific features of the car (e.g., sunroof, leather seats, advanced safety features) have high importance scores, it suggests that these factors contribute significantly to the pricing decisions. Higher trim levels and additional features often lead to higher prices.
e. **Other Features:** The analysis may also reveal the importance of other features, such as market mirror ratio (MMR), interior and exterior colors, transmission type, and body type. The importance of these features suggests their influence on the pricing decisions made by buyers and sellers in the used car market.

Understanding the feature importance helps stakeholders in the automotive industry, including buyers, sellers, and manufacturers, to make informed decisions regarding pricing, marketing, and product development. It enables them to focus on the most influential factors affecting car prices, align their strategies accordingly, and meet the demands of the market effectively.

Limitations and Future Work

Data Limitations: The current car price prediction model may have certain limitations related to data availability. The model's performance heavily relies on the quality, completeness, and representativeness of the dataset used for training. If the dataset is limited in size or lacks diversity in terms of car brands, models, or geographical regions, the model's predictions may not generalize well to unseen data. Future work should focus on obtaining larger and more diverse datasets to improve the model's accuracy and robustness.

Model Assumptions: The car price prediction model is based on certain assumptions about the relationships between the input features and the target variable. These assumptions may include linearity, independence, and absence of multicollinearity among the features. However, in real-world scenarios, these assumptions may not always hold true. Future work could involve exploring more flexible modeling techniques, such as non-linear regression models or deep learning approaches, to capture complex relationships and interactions among the features and improve prediction accuracy.

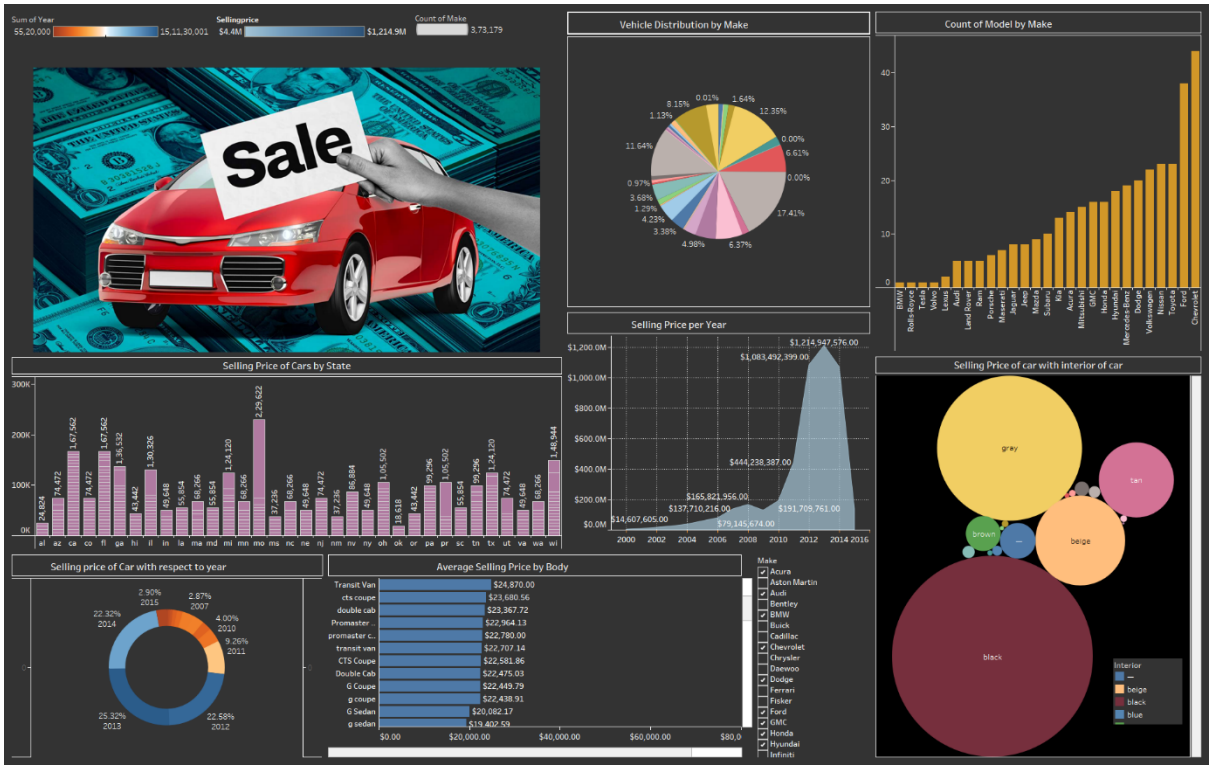
Potential Bias: The car price prediction model may be susceptible to bias if the training data is biased or reflects existing societal biases. For example, if the dataset predominantly includes certain car brands or models, the model's predictions may be skewed towards those brands or models. It is crucial to address potential biases and ensure fairness in the model's predictions. Future work should focus on collecting diverse and representative data and employing techniques to detect and mitigate bias in the model's predictions.

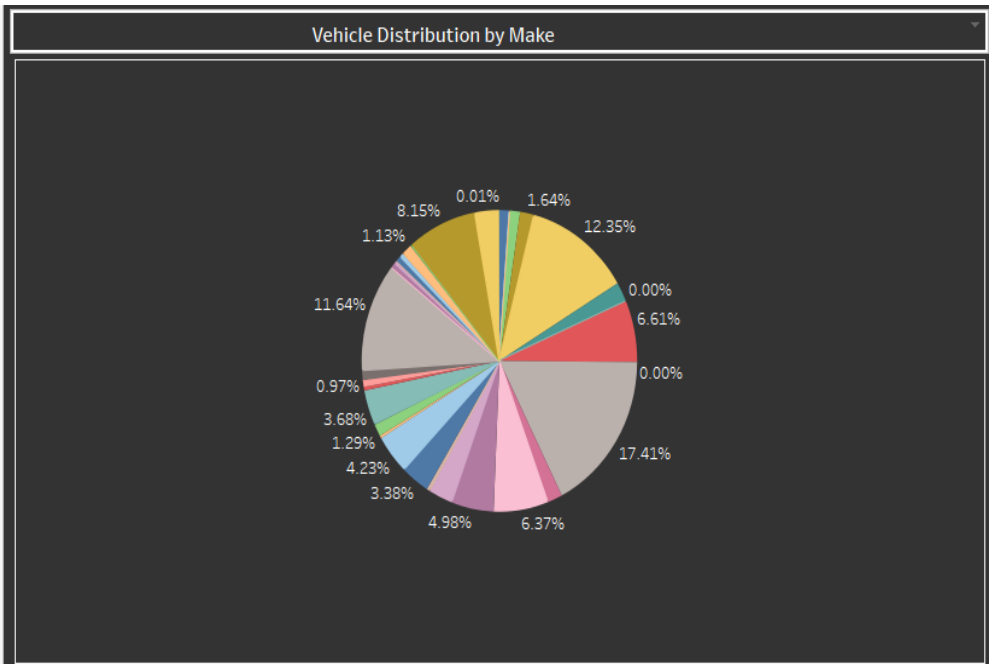
Future Directions and Improvements:

- **Improved Feature Engineering:** Future work can involve more extensive feature engineering to capture additional relevant information. This may include creating derived features such as car age since the manufacturing year, depreciation rate, or incorporating sentiment analysis of user reviews and ratings associated with specific car models. These enriched features can provide a more comprehensive representation of the factors influencing car prices, leading to improved predictive accuracy.
- **Ensemble Modeling:** Ensemble modeling techniques, such as combining predictions from multiple models or using model stacking, can be explored to enhance the accuracy and robustness of car price predictions. By leveraging the strengths of different models, ensemble techniques can potentially improve performance and reduce model bias.
- **Incorporating External Data Sources:** Integrating external data sources can enhance the predictive power of the model. For example, incorporating macroeconomic indicators, fuel prices, interest rates, or regional market trends can provide additional context and improve the model's ability to capture market dynamics and fluctuations in car prices.
- **Dynamic Pricing Models:** Future work can focus on developing dynamic pricing models that consider real-time market information, such as demand-supply dynamics, competitor prices, and seasonal variations. This would enable the model to adapt and provide more accurate and timely predictions in dynamic market conditions.
- **User-Specific Preferences:** Incorporating user-specific preferences and customization options in the model can further enhance its applicability. For instance, allowing users to input their desired features or preferences and tailoring the predictions accordingly can better align the model's outputs with individual buyer requirements.

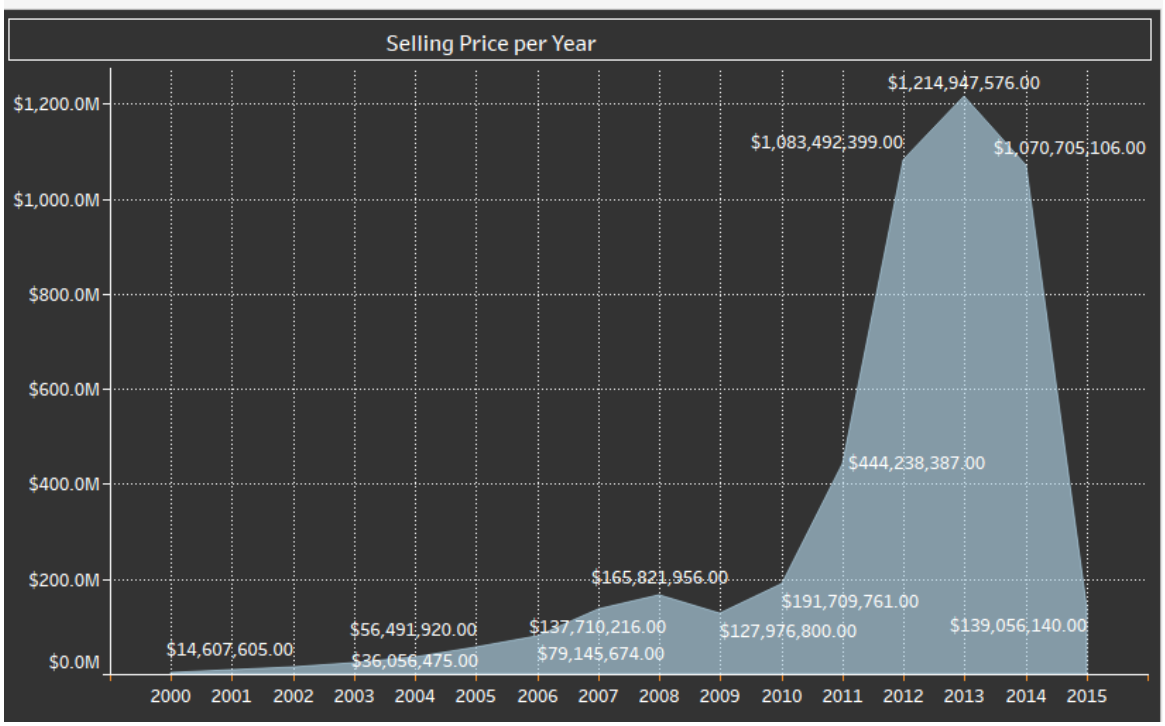
By addressing the limitations, improving feature engineering, considering ensemble techniques, leveraging external data sources, and developing dynamic pricing models, the accuracy and applicability of the car price prediction model can be significantly enhanced, benefiting both buyers and sellers in the used car market.

Data Visualizations:

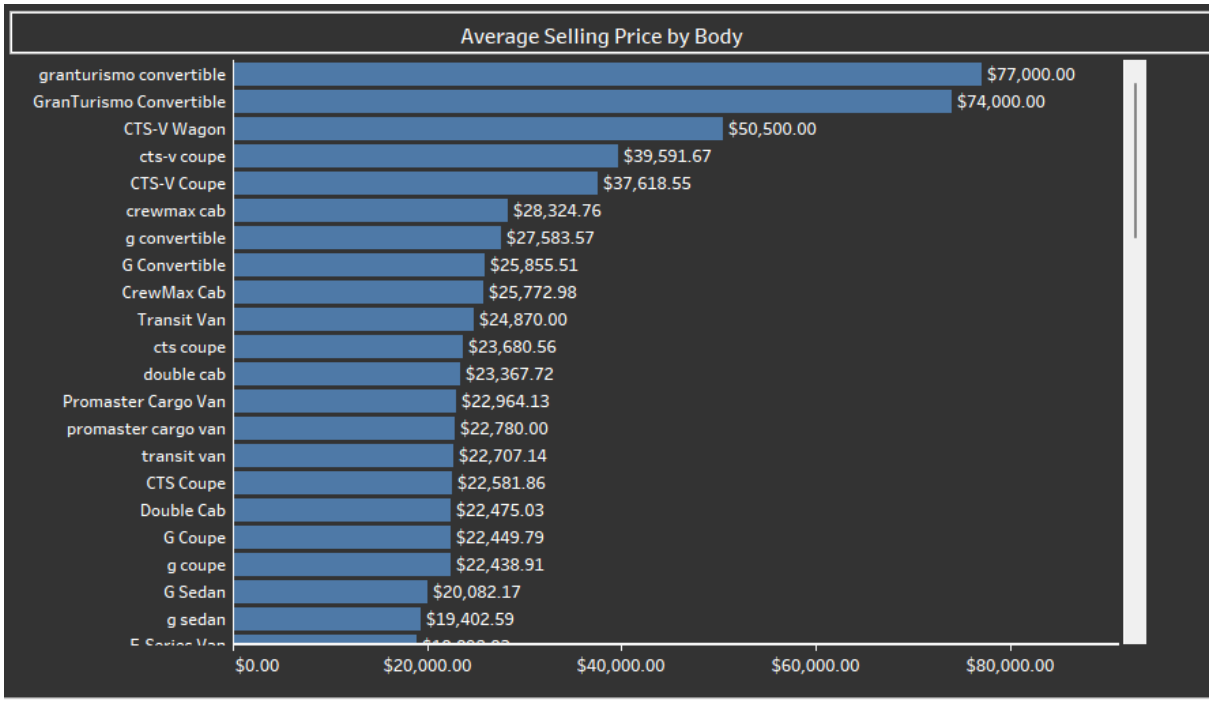




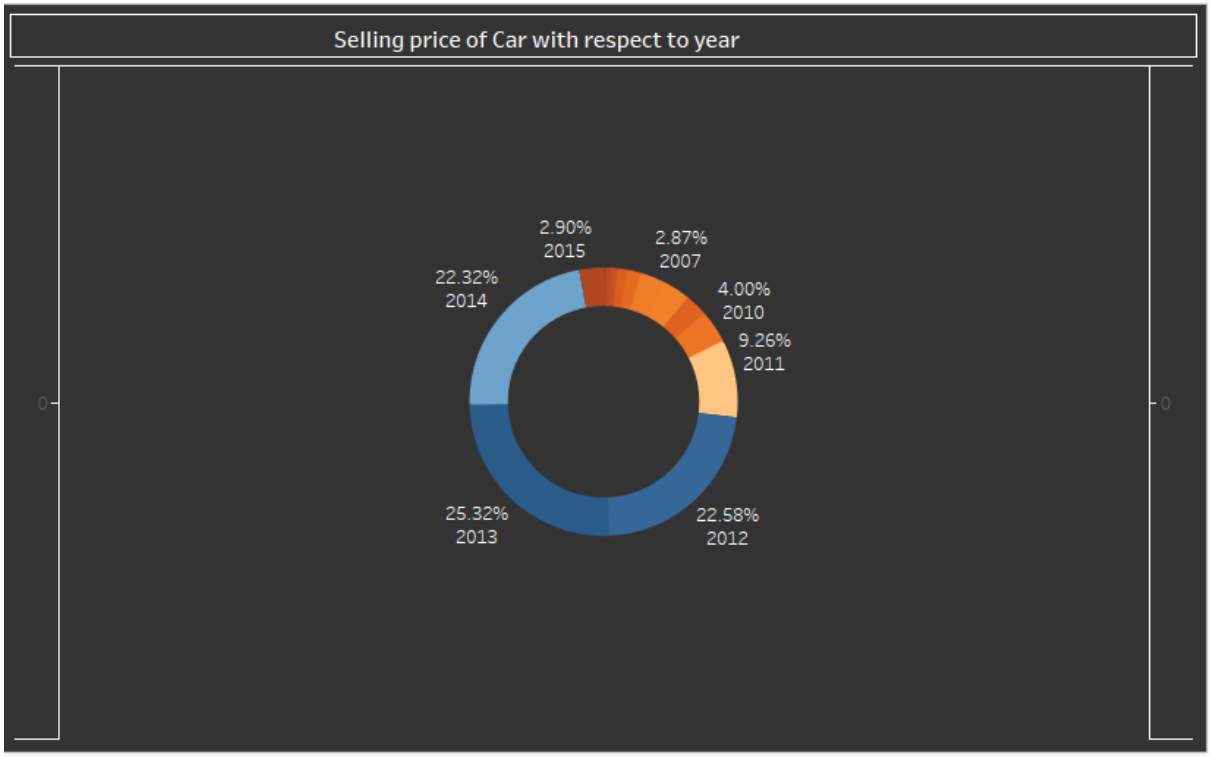
This pie chart provides information on the distribution of car makes, showcasing the count and percentage of each make. Chevrolet, Ford, and Toyota are the most common mainstream car makes, while luxury brands like Mercedes-Benz, BMW, and Audi also have a presence. This pie chart exhibits a diverse range of car makes, including lesser-known or niche brands. Further analysis goals and correlations with other variables can provide deeper insights into the data.



This area chart spanning from 2000 to 2015, reveals that selling prices of cars generally exhibit an increasing trend with occasional fluctuations. Prices experienced a steep rise from 2006 to 2008, followed by a decline in 2009. The peak selling prices were recorded in 2011, with subsequent years showing some fluctuations. Th area chart encompasses a wide price range, from millions to billions, indicating the presence of cars from different market segments. The increasing prices may reflect changes in market dynamics, including demand, inflation, and other economic factors.

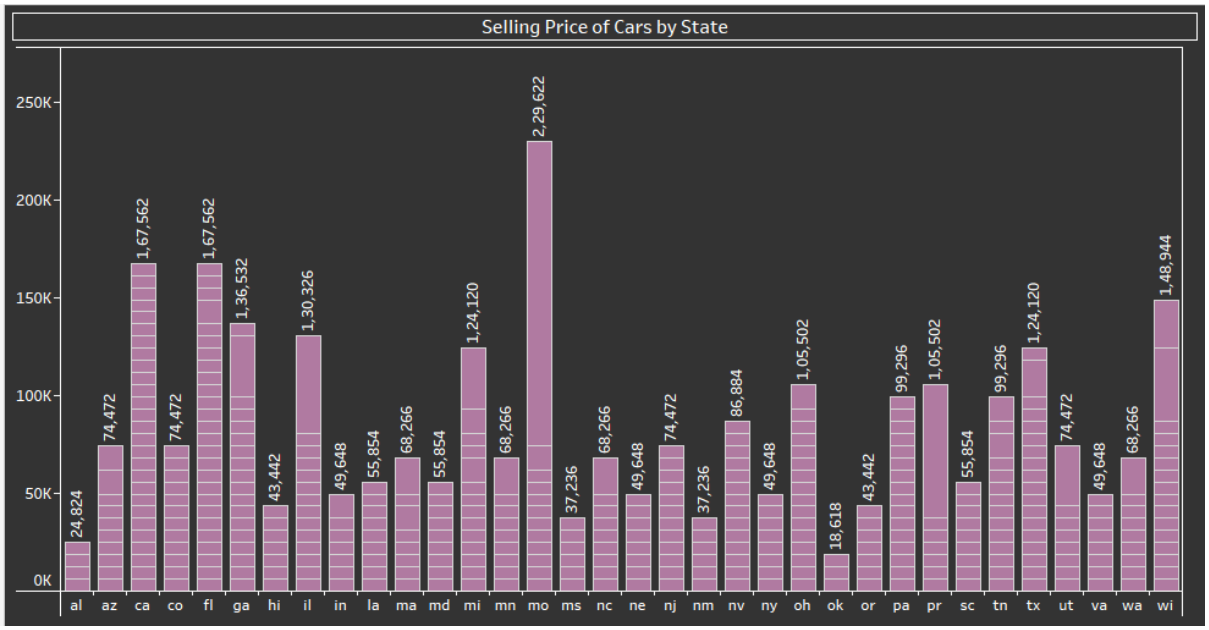


This bar chart shows that vehicle selling prices vary by body type, with Sedan, SUV, Coupe, Minivan, and Convertible having higher average prices. Consistent prices are observed for certain body types despite minor naming variations. Luxury and performance models command significantly higher prices. The bar graph offers a wide price range, necessitating further analysis for specific insights.



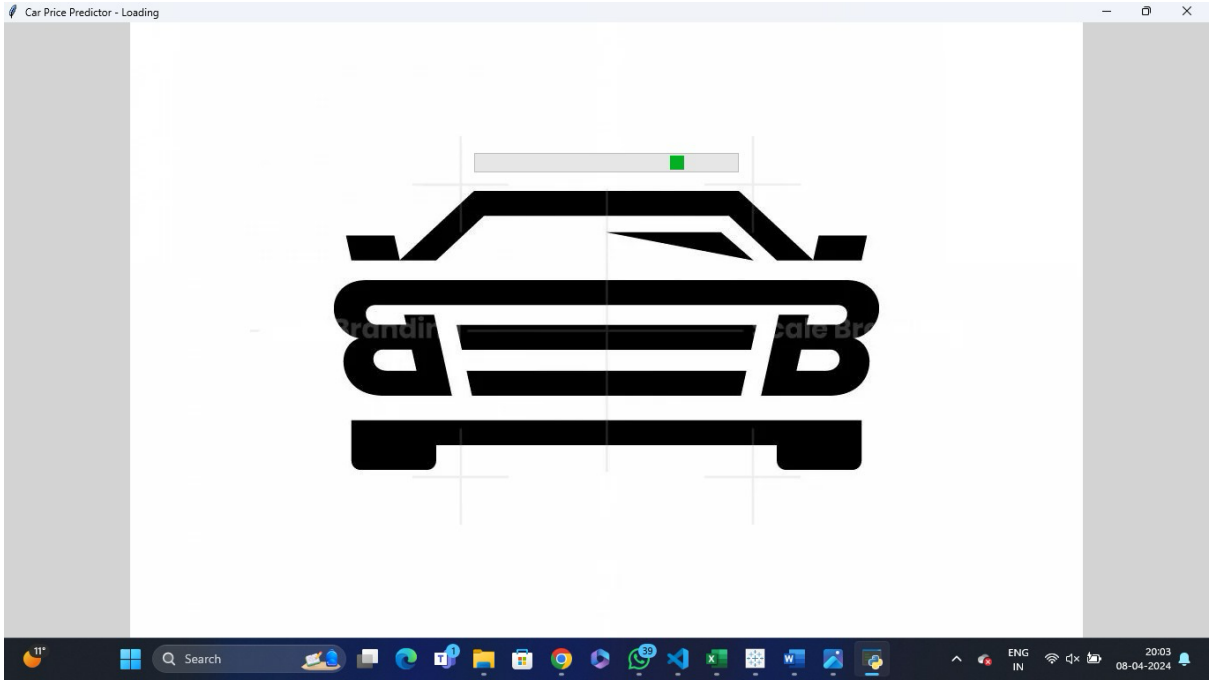
The above donut chart that selling prices of vehicles generally increased from 2000 to 2012, with a peak in 2011, but decreased in subsequent years. The year 2013 stood out with the highest percentage

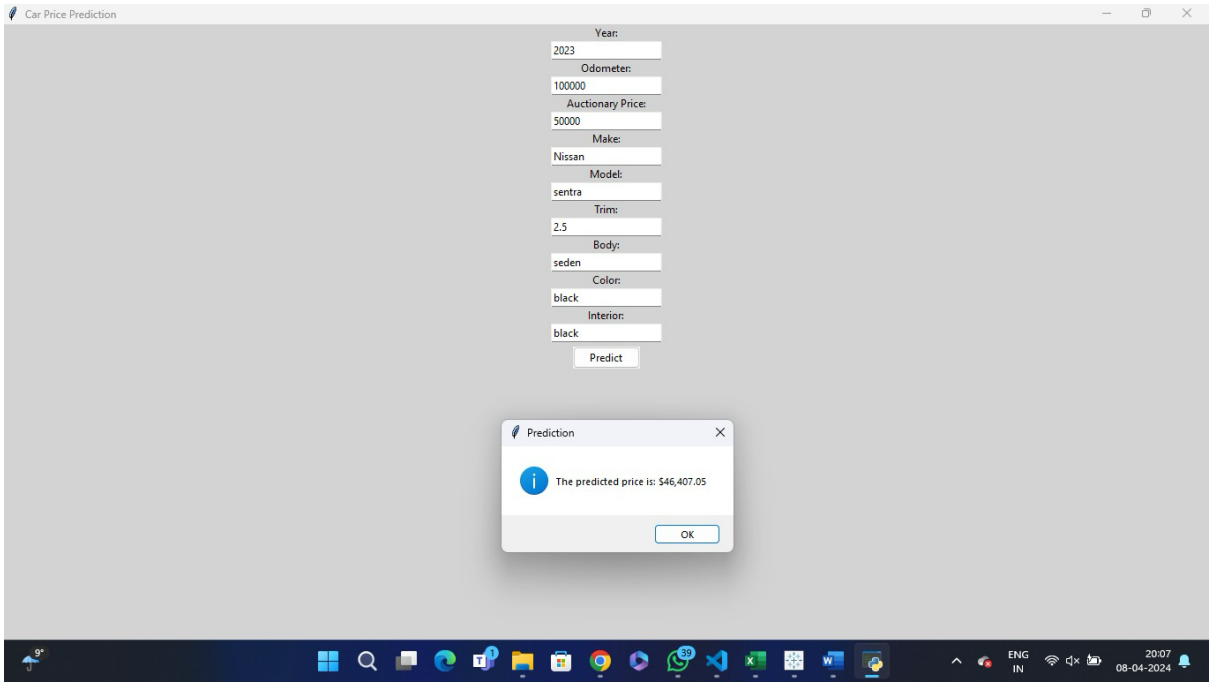
of total selling price. Fluctuations in selling prices indicate market volatility. Incomplete data is present in the last row.



The dataset shows the selling prices of vehicles categorized into price ranges along with the corresponding states. Prices vary across states, with higher-priced sales concentrated in California, Florida, and Texas, while lower-priced sales are prominent in Wisconsin, Washington, and Virginia. Further analysis is needed for comprehensive insights.

GUI (Graphical User Interface):





This GUI is a system for predicting car prices. There are several input fields, buttons, and a graph on it. This GUI's objective is to enable users—such as car sellers, car buyers, and industry professionals such as dealerships, appraisers, and insurance companies to enter information, make choices, and view the outcomes of the predicted car prices. Users can maximize their profit margins, set competitive car prices, and make well-informed decisions based on precise price predictions by utilizing this GUI.

Conclusion:

In conclusion, developing a car price prediction system using XGBoost represents a significant advancement in the automotive market. By accurately estimating used car prices based on various features, the system empowers buyers, sellers, and industry professionals to make informed decisions, set competitive prices, and optimize profits. Through meticulous data collection, preprocessing, and feature engineering, the model achieves high predictive accuracy, as evidenced by evaluation metrics such as MSE, RMSE, and R-squared. Feature importance analysis reveals key factors influencing car prices, providing valuable insights for stakeholders. Despite challenges such as data quality issues and model assumptions, future work holds promise for improving the system's accuracy through enhanced feature engineering, ensemble modeling, and dynamic pricing approaches. Overall, the car price prediction system facilitates transparency, fairness, and efficiency in the automotive market, contributing to better decision-making and customer satisfaction.

References:

1. Kaggle: https://www.kaggle.com/datasets/sujay1844/used-car-prices
2. XGBoost Documentation: https://xgboost.readthedocs.io/en/stable/
3. XGBoost Documentation: https://medium.com/sfu-cspmp/xgboost-a-deep-dive-into-boosting-f06c9c41349
4. Geeks for Geeks: https://www.geeksforgeeks.org/xgboost/
5. W3school: https://www.w3schools.com/python/