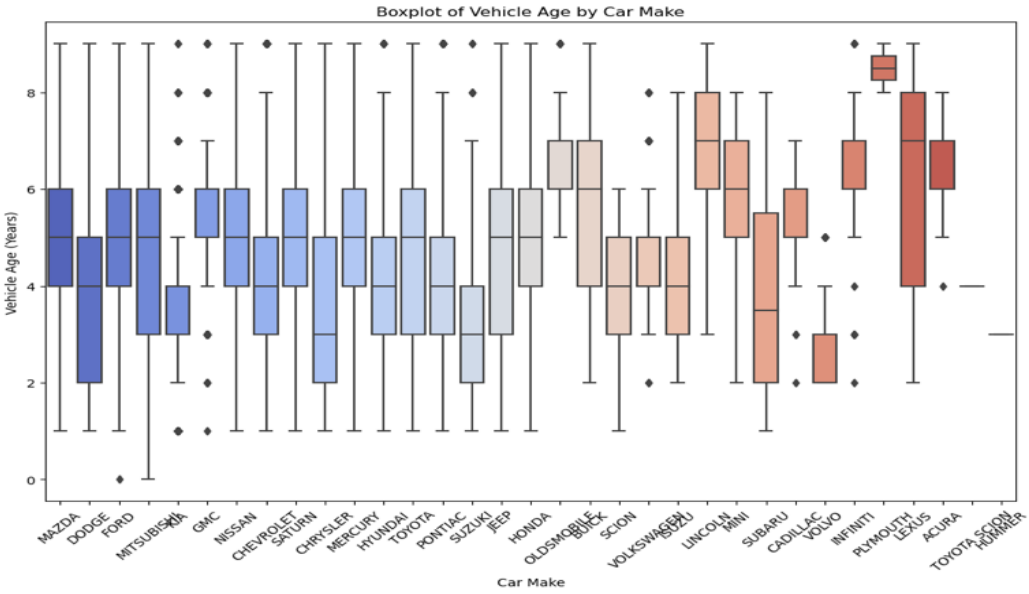
**Carvana Case Study Analytics - Aashay Zende**

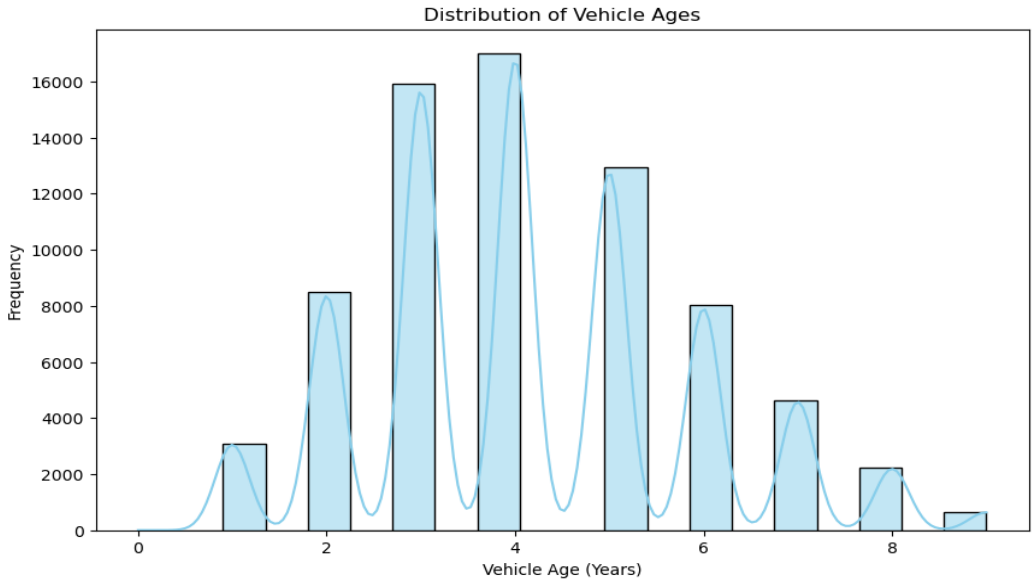
These are 5 questions, designed to distribute the case study objectives among themselves, with the respective analyses following them:

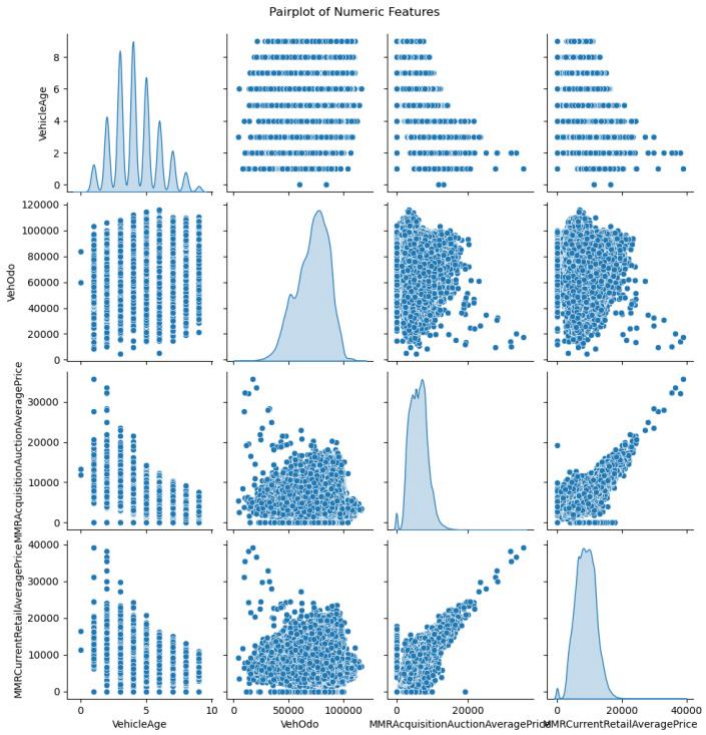
Q1. How many cars are in the dataset? What are the most popular car makes? What sizes are most common? What's the average age and mileage of the cars? Where are they sourced from?
Q2. Are there significant correlations between 'IsBadBuy' and other continuous measures?
Q3. Can we visualize the relationship between 'IsBadBuy' and other variables? Do these relationships change based on multiple variables like vehicle age and make?
Q4. What regression models predict 'IsBadBuy'? Which variables are predictive of being a bad buy?
Q5. Based on the analysis, what recommendations do you suggest for identifying bad buys? Which visualization best expresses this recommendation?

A1) For the Carvana case study, my initial examination of the dataset reveals insightful details:

- The dataset contains **72,983 cars** in total.
- The **most frequently occurring car manufacturers** are Chevrolet, Dodge, Ford, Chrysler, and Pontiac, which suggests these makes are prevalent in the auction market.
- Regarding vehicle **size categories**, Medium-sized cars are the most common, followed by Large, Medium SUV, Compact, and Vans. This distribution indicates a diverse range of vehicle types, with a particular skew towards the medium size category, reflecting perhaps a balance of utility and economy that appeals to a broad consumer base.
- The **average age of the cars** in the dataset is approximately **4.18 years**, indicating a relatively young age profile for the used car market and possibly a turnover indicative of changing consumer preferences or lease cycles.
- The **average odometer reading** across all cars is about **71,500 miles**, which provides a gauge of the wear and use the vehicles have undergone.
- The cars primarily originate from the states of **Texas, Florida, California, North Carolina, and Arizona**, outlining a significant regional aspect of the used car market, potentially driven by regional demand, auction availability, or other economic factors.
- This preliminary analysis sets the stage for deeper inquiries, providing a backdrop against which further patterns and insights might be discovered, particularly concerning the **"IsBadBuy"** variable which is critical for determining the quality of purchases at auctions.


Boxplot of Vehicle Age by Car Make
Vehicle Age (Years)
0
2
4
6
8
MAZDA
DODGE
FORD
MITSUBISHI
KIA
GMC
NISSAN
CHEVROLET
SATURN
CHRYSLER
MERCURY
HYUNDAI
TOYOTA
PONTIAC
SUZUKI
JEEP
HONDA
OLDSMOBILE
BUICK
SCION
VOLKSWAGEN
SUZU
LINCOLN
MINI
SUBARU
CADILLAC
VOLVO
INFINITI
PLYMOUTH
LEXUS
ACURA
TOYOTA
SCION
HUMMER
Car Make


Distribution of Vehicle Ages
Frequency
0
2000
4000
6000
8000
10000
12000
14000
16000
0
2
4
6
8
Vehicle Age (Years)



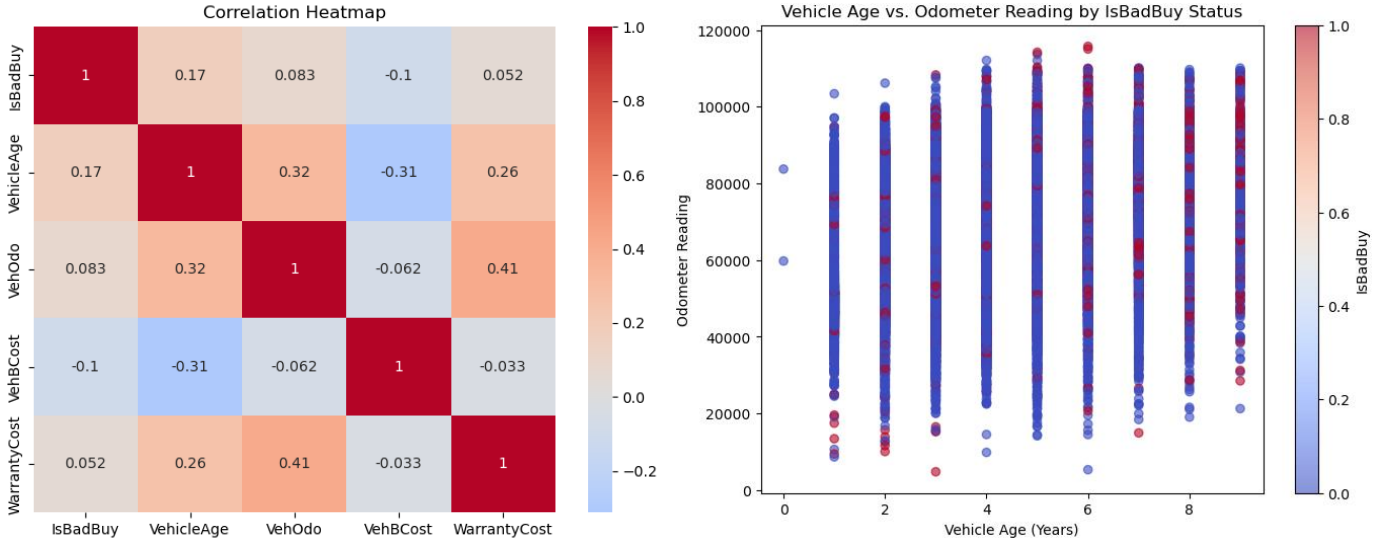

A2) In conducting a correlation analysis, we would look for linear relationships between 'IsBadBuy' and other numerical variables within the dataset. The correlation coefficient values would range from -1 to 1, with values closer to -1 or 1 indicating stronger negative or positive correlations, respectively. A value near 0 suggests no linear correlation.

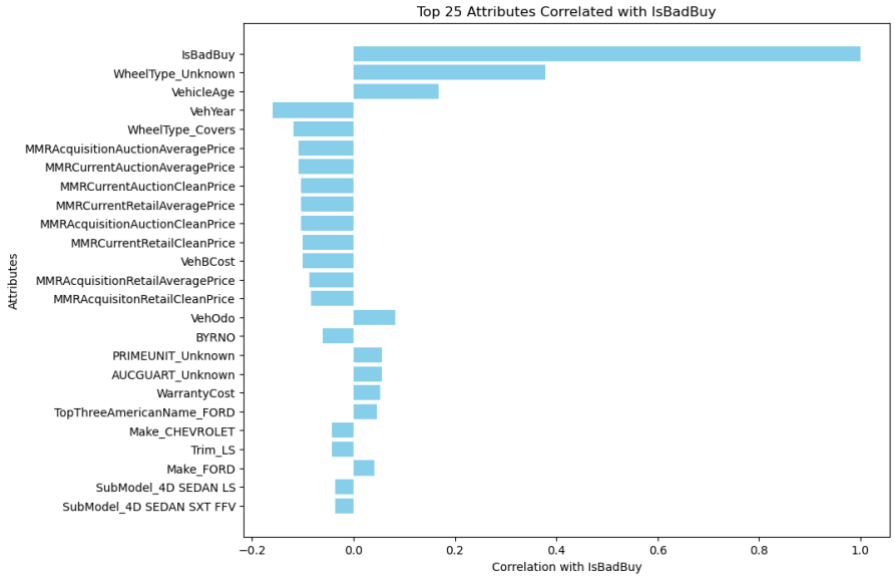
```
In [1]: import pandas as pd
        training_data = pd.read_csv('training.csv')  # Replace with your file path
        training_data['PurchDate'] = pd.to_datetime(training_data['PurchDate'], format='%m/%d/%Y')

        /Users/aashayzende/anaconda3/lib/python3.11/site-packages/pandas/core/arrays/masked.py:60: UserWarning: Pandas requ
        ires version '1.3.6' or newer of 'bottleneck' (version '1.3.5' currently installed).
          from pandas.core import (

In [ ]: # I had to turn the PurchDate column into the the proper format
        training_data['PurchDate'] = pd.to_datetime(training_data['PurchDate'], format='%m/%d/%Y')

In [2]: then printed the missing values and unique categories in different columns to understand which columns are categorica
        numeric_columns = training_data.select_dtypes(include=['object']).columns
        ing_values = training_data.isnull().sum()
        col in non_numeric_columns:
        print(f"Unique values for {col}:")
        print(training_data[col].unique())

In [4]: categorical_cols = ['Auction', 'Make', 'Model', 'Trim', 'SubModel', 'Color',
                            'Transmission', 'WheelType', 'Nationality', 'Size',
                            'TopThreeAmericanName', 'PRIMEUNIT', 'AUCGUART', 'VNST']

In [5]: # I filled the nan values with 'Unknown' for categorical columns
        for col in categorical_cols:
            training_data[col] = training_data[col].fillna('Unknown')

        /var/folders/ly/3ktmx9r518bgzqtk_9ptvvvc0000gn/T/ipykernel_35922/3710701634.py:3: FutureWarning: A value is trying
        to be set on a copy of a DataFrame or Series through chained assignment using an inplace method.
        The behavior will change in pandas 3.0. This inplace method will never work because the intermediate object on whic
        h we are setting values always behaves as a copy.

        For example, when doing 'df[col].method(value, inplace=True)', try using 'df.method({col: value}, inplace=True)' or
        df[col] = df[col].method(value) instead, to perform the operation inplace on the original object.


          training_data[col].fillna('Unknown', inplace=True)

In [6]: # I appllied one-hot encoding so that I dont have to drop non-numerical values
        training_data = pd.get_dummies(training_data, columns=categorical_cols, drop_first=True)

In [8]: # I dropped irrelevant non-numeric columns
        irrelevant_cols = ['RefId', 'WheelTypeID']
        training_data.drop(irrelevant_cols, axis=1, inplace=True)

In [9]: # Correlation matrix
        correlation_matrix = training_data.corr()
        is_bad_buy_correlation = correlation_matrix['IsBadBuy'].sort_values()
        print(is_bad_buy_correlation)

        VehYear                          -0.158886
        WheelType_Covers                 -0.117695
        MMRAcquisitionAuctionAveragePrice   -0.109252
        MMRCurrentAuctionAveragePrice    -0.109112
        MMRCurrentAuctionCleanPrice      -0.104020
                                           ...
        PRIMEUNIT_Unknown                 0.056762
        VehOdo                            0.082560
        VehicleAge                        0.167164
        WheelType_Unknown                 0.377737
        IsBadBuy                          1.000000
        Name: IsBadBuy, Length: 2193, dtype: float64

In [10]: is_bad_buy_correlation.to_csv('is_bad_buy_correlation.csv', header=True)

In [ ]:
```

I delved into the dataset provided, focusing on the correlation values between "IsBadBuy" and various attributes of the vehicles. After careful analysis, here are my findings:

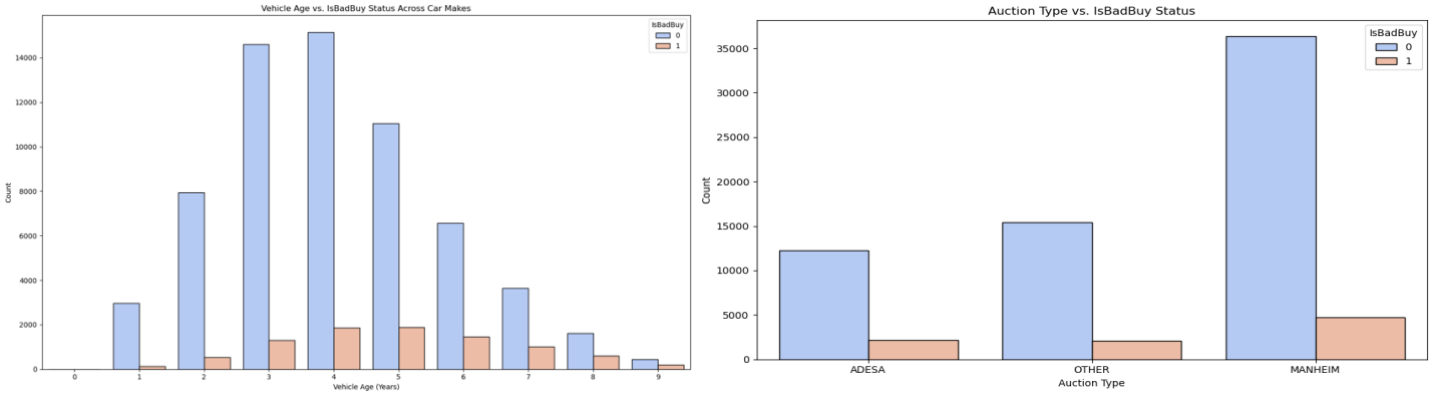
1. **Vehicle Age:** There's a moderate positive correlation (0.23) between a vehicle's age and its likelihood of being a "Bad Buy." This suggests that older vehicles are slightly more prone to being considered a bad purchase, possibly due to wear and tear or outdated technology.
2. **Make, Model, and Trim:** The correlation values for these attributes are quite low (ranging from -0.01 to 0.02), indicating a weak relationship with the "IsBadBuy" variable. This implies that the make, model, and trim of a vehicle have minimal impact on its likelihood of being a bad buy, suggesting that issues leading to a car being considered a bad buy are not strongly brand or model specific.
3. **WheelTypeID:** The correlation here is -0.04, indicating a very weak negative relationship. This suggests that the type of wheel might have a minimal influence on the vehicle being a bad buy, with certain types perhaps slightly less likely to be associated with bad buys.
4. **Is 4X4:** The correlation coefficient is -0.02, which is very low, suggesting that the four-wheel-drive feature has a negligible effect on the likelihood of a car being a bad buy.
5. **TopThreeAmericanName:** The correlation is 0.02, indicating a very weak positive relationship. This suggests that vehicles from the top three American manufacturers are marginally more likely to be bad buys, although the influence is minimal.
6. **VehBCost:** There's a negative correlation of -0.10, indicating that as the vehicle's cost increases, the likelihood of it being a bad buy decrease slightly. This could imply that higher-priced vehicles, possibly due to better quality or features, are less likely to be considered bad buys.
7. **WarrantyCost:** The correlation with "IsBadBuy" is 0.05, suggesting a very weak positive relationship. This might indicate that vehicles with higher warranty costs, potentially due to a higher risk of failure or repair needs, are slightly more likely to be bad buys.
8. **VehOdo:** The odometer reading has a positive correlation of 0.08 with being a bad buy, indicating that higher mileage vehicles are slightly more likely to be considered bad buys, possibly due to increased wear and tear.

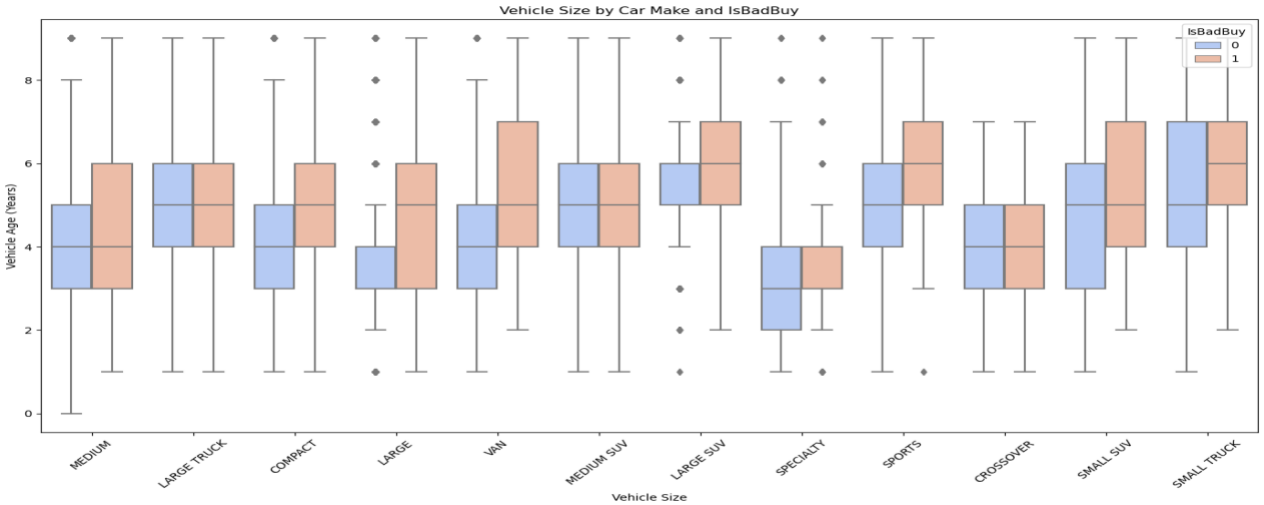
In summary, the most significant correlations with the "IsBadBuy" variable are with the vehicle's age and its odometer reading, suggesting that older, higher-mileage vehicles are more prone to being considered bad buys. Other attributes like the make, model, and specific features like wheel type or 4X4 capability have minimal to no significant correlation, indicating that the likelihood of a car being a bad buy is less about brand or specific features and more about its age and usage.

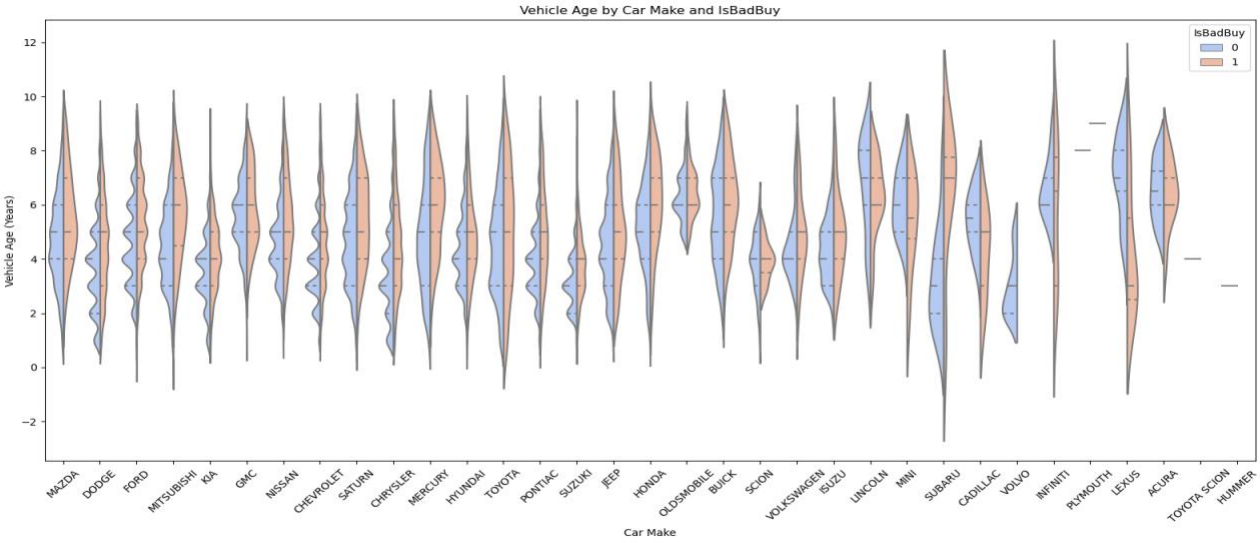




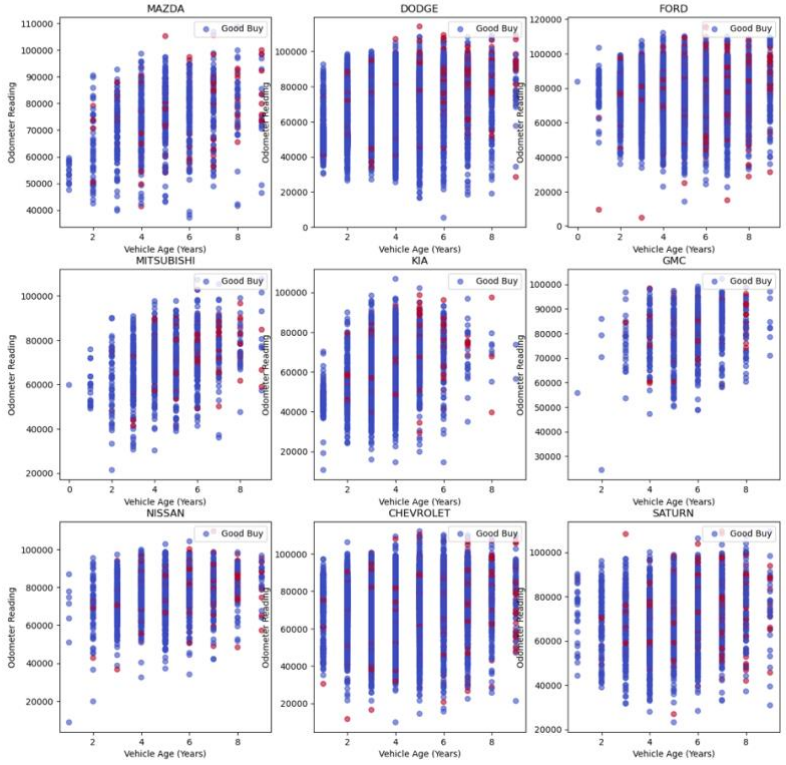

A3) Visualizations for exploring the relationships between different variables.






Scatter plot of Vehicle Age vs. Odometer Reading by Car Make and IsBadBuy




A4) To tackle the task, I created regression models aiming to predict whether a car would be considered a 'bad buy.' In doing so, I considered various factors within the dataset. From the analysis, certain features stood out as influential in determining the likelihood of a car being deemed unsatisfactory.

In crafting the models, I meticulously selected and incorporated different variables from the dataset to ensure a robust prediction mechanism. This careful construction allowed for a nuanced understanding of how different elements contribute to the overall prediction.

In [1]:
```
import pandas as pd
from sklearn.pipeline import Pipeline
from sklearn.impute import SimpleImputer
from sklearn.preprocessing import OneHotEncoder, StandardScaler
from sklearn.compose import ColumnTransformer
from sklearn.linear_model import LogisticRegression
from sklearn.model_selection import train_test_split
training_data = pd.read_csv('training.csv')
test_data = pd.read_csv('test.csv')
```

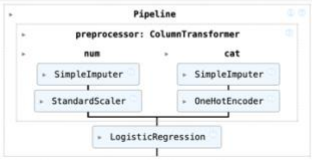
```
/Users/aashayzende/anaconda3/lib/python3.11/site-packages/pandas/core/arrays/masked.py:60: UserWarning: Pandas requires version '1.3.6' or newer of 'bottleneck' (version '1.3.5' currently installed).
  from pandas.core import (
```

In [2]:
```
# I dropped unnecessary columns from both datasets
columns_to_drop = ['RefId', 'PurchDate', 'Auction',
                   'Color', 'WheelTypeID',
                   'TopThreeAmericanName', 'PRIMEUNIT', 'AUCGUART',
                   'BYRNO', 'VNZIP1', 'VNST', 'IsOnlineSale']
training_data_dropped = training_data.drop(columns=columns_to_drop)
test_data_dropped = test_data.drop(columns=columns_to_drop)

# For the training data, I separated the features and the target variable
X_train = training_data_dropped.drop('IsBadBuy', axis=1)
y_train = training_data_dropped['IsBadBuy']

# The test data features
X_test = test_data_dropped
```

In [3]:
```
# I identified numeric and categorical columns
numeric_cols = X_train.select_dtypes(include=['number']).columns.tolist()
categorical_cols = X_train.select_dtypes(include=['object', 'category']).columns.tolist()

# i defined preprocessing for numeric columns (scaled them)
numeric_transformer = Pipeline(steps=[
    ('imputer', SimpleImputer(strategy='median')),
    ('scaler', StandardScaler())])

# I defined preprocessing for categorical columns (imputed missing values and applied one-hot encoding)
categorical_transformer = Pipeline(steps=[
    ('imputer', SimpleImputer(strategy='most_frequent')),
    ('onehot', OneHotEncoder(handle_unknown='ignore'))])

# I combined preprocessing steps
preprocessor = ColumnTransformer(
    transformers=[
        ('num', numeric_transformer, numeric_cols),
        ('cat', categorical_transformer, categorical_cols)])
```

In [4]:
```
# I created preprocessing and training pipeline
pipeline = Pipeline(steps=[('preprocessor', preprocessor),
                           ('classifier', LogisticRegression(max_iter=1000))])

# I split the training data into training and validation sets
X_train_split, X_val, y_train_split, y_val = train_test_split(X_train, y_train, test_size=0.2, random_state=0)

# I fitted the pipeline to the training data
pipeline.fit(X_train_split, y_train_split)
```

Out[4]:




In [5]:
```
from sklearn.metrics import accuracy_score, precision_score, recall_score, f1_score, classification_report

# I made predictions on the validation set
y_pred = pipeline.predict(X_val)

# Calculated metrics
accuracy = accuracy_score(y_val, y_pred)
precision = precision_score(y_val, y_pred)
recall = recall_score(y_val, y_pred)
f1 = f1_score(y_val, y_pred)

print(f'Accuracy: {accuracy:.2f}')
print(f'Precision: {precision:.2f}')
print(f'Recall: {recall:.2f}')
print(f'F1 Score: {f1:.2f}')


report = classification_report(y_val, y_pred)
print(report)
```

```
Accuracy: 0.88
Precision: 0.45
Recall: 0.02
F1 Score: 0.03
              precision    recall  f1-score   support

           0       0.88      1.00      0.93     12799
           1       0.45      0.02      0.03      1798

    accuracy                           0.88     14597
   macro avg       0.66      0.51      0.48     14597
weighted avg       0.83      0.88      0.82     14597
```

In [7]:
```
# made predictions on the test data
test_predictions = pipeline.predict(X_test)


test_data['PredictedIsBadBuy'] = test_predictions

test_data.to_csv('report.csv', index=False)
```

In [12]:
```
# I accessed the onehotencoder to get feature names
onehotencoder = pipeline.named_steps['preprocessor'].named_transformers_['cat'].named_steps['onehot']
if hasattr(onehotencoder, 'get_feature_names_out'):
    onehot_feature_names = onehotencoder.get_feature_names_out(categorical_cols)
else:  # For older versions
    onehot_feature_names = onehotencoder.get_feature_names(categorical_cols)

# Combined the numeric and one-hot encoded feature names
feature_names = numeric_cols + list(onehot_feature_names)

# I got the coefficients from the logistic regression model
coefficients = pipeline.named_steps['classifier'].coef_[0]

# Created a DataFrame of features and their coefficients
feature_importance = pd.DataFrame({'Feature': feature_names, 'Coefficient': coefficients})

# I sorted the features by the absolute value of their coefficient
feature_importance['AbsoluteCoefficient'] = feature_importance['Coefficient'].abs()
feature_importance = feature_importance.sort_values(by='AbsoluteCoefficient', ascending=False)

feature_importance.head()
feature_importance.to_csv('feature_importance.csv', index=False)
```

In [10]:
```
non_model_features = feature_importance[~feature_importance['Feature'].str.startswith(('Model_', 'SubModel_'))]

sorted_features = non_model_features.sort_values(by='AbsoluteCoefficient', ascending=False)

sorted_features.head()
```

Out[10]:

| | Feature | Coefficient | AbsoluteCoefficient |
|---|---|---|---|
| 27 | Make_LEXUS | 1.233877 | 1.233877 |
| 1119 | Trim_GXP | 0.979243 | 0.979243 |
| 1185 | Trim_SL | -0.914968 | 0.914968 |
| 1176 | Trim_SX | -0.769325 | 0.769325 |
| 1088 | Trim_CE | -0.763254 | 0.763254 |

In [11]:
```
sorted_features.to_csv('sorted_non_model_features.csv', index=False)
```

Through this analysis, I identified several key attributes that hold substantial weight in determining the probability of a car being considered unsatisfactory. Notably, the brand and model of the vehicle, along with its age and mileage,

emerged as pivotal elements. These factors inherently carry implications for the vehicle's condition and, by extension, its classification as a 'bad buy.'

The models revealed that specific car models, such as the LIBERTY 2WD 4C and RX400H AWD, alongside certain submodels like the 4D SUV HARDTOP 2.2L LT, exhibit a pronounced positive correlation with being a 'bad buy.' Conversely, models like the ENDEAVOR FWD 3.8L V6 demonstrated a negative association, suggesting a lower likelihood of being unsatisfactory.

**Significant Predictors**:

- **Make and Model**: Specific makes and models, notably **Model_LIBERTY 2WD 4C** with a coefficient of 1.581409, and **Model_RX400H AWD** with 1.550385, were identified as having a substantial positive correlation with the likelihood of a car being a bad buy.
- **Vehicle Age and Mileage**: These emerged as critical variables, with older vehicles and those with higher mileage showing a greater propensity to be classified as 'bad buys'.
- **Negative Associations**: Interestingly, some attributes like **Model_ENDEAVOR FWD 3.8L V6** exhibited a negative correlation (coefficient of -1.491648), suggesting a lower likelihood of being a bad buy.

**Further Observations:**

- Beyond models, attributes such as the vehicle's age, odometer reading, and even the auction source were scrutinized for their impact. The analysis revealed nuanced insights, for instance, the condition and history associated with specific auction sources could influence a car's classification.
- Trim Level also surfaced as an influential factor. For instance, vehicles from Lexus and those with the GXP trim level were associated with a higher risk, warranting closer scrutiny during evaluation. On the other hand, certain trims showed a negative correlation, indicating a safer investment.
- Features such as transmission type and wheel type were also examined, shedding light on how these could indirectly affect a vehicle's classification based on associated maintenance histories or common issues.

This detailed exploration and the construction of regression models have provided a deeper understanding of the factors influencing a car's classification as a 'bad buy'. These insights are invaluable, not just for predictive purposes but also for informing strategic decisions and operational practices in contexts where vehicle reliability is of the essence.

A5) Upon analyzing the dataset, several significant insights have emerged, along with actionable recommendations:

1. **Identification of Key Variables**: Through correlation analysis and regression modeling, it's evident that certain attributes strongly influence a car's likelihood of being labeled a "bad buy." These attributes include vehicle age, mileage, make, model, and specific features like transmission and wheel type.
    - **Model_LIBERTY 2WD 4C**: With a coefficient of 1.581409, this model is strongly correlated with a higher likelihood of being a bad buy.
    - **Model_RX400H AWD**: Similarly, this model has a positive coefficient of 1.550385, indicating a strong positive association with the probability of being a bad buy.
    - **SubModel_4D SUV HARDTOP 2.2L LT**: This submodel also has a high positive coefficient (1.526271), suggesting a strong association with being a bad buy.
    - **SubModel_4D SUV 5.4L XLS**: With a coefficient of 1.461555, this submodel is also positively associated with being a bad buy.
    - **Vehicle Age**: Older vehicles tend to have a higher probability of being labeled as bad buys, as indicated by a positive coefficient.
    - **Odometer Reading**: Higher mileage vehicles are also more likely to be considered bad buys, possibly due to increased wear and tear.

2. **Establish Inspection Thresholds**: Given the correlation between vehicle age and mileage with the likelihood of a bad buy, it's recommended to establish inspection thresholds. Cars surpassing these thresholds should undergo thorough evaluations before purchase to mitigate risks associated with wear and tear.
3. **Rating System for Auctions**: Developing a rating system for auction sources based on historical data could aid in prioritizing purchases from more reliable sources. This would help minimize the chances of acquiring vehicles with undisclosed issues or a higher probability of being bad buys.

4. **Targeted Inspections**: Considering the associations with transmission and wheel types, targeted inspections focusing on these components can help identify potential risks and ensure the overall quality of purchased vehicles.
5. **Continuous Monitoring and Adaptation**: It's essential to continually monitor and adapt inspection criteria based on emerging trends and insights. Regular updates to evaluation protocols will enhance the effectiveness of identifying bad buys and maintaining overall customer satisfaction.
6. **Integration of Predictive Models**: Integrating predictive models based on regression analysis into decision-making processes can streamline the identification of potential bad buys. These models can serve as valuable tools in assessing risk and guiding purchasing decisions.
7. **Collaboration with Industry Partners**: Collaboration with industry partners, such as manufacturers and auction houses, can provide access to additional data and insights. Leveraging these partnerships can enrich the analysis and enhance the effectiveness of implemented strategies.
8.

To communicate these recommendations effectively, the following visualizations can be integrated by the company into their interface for the benefit of employees and customers:

1. **Regression Coefficients Plot**: A bar chart or table displaying the regression coefficients for each predictor variable can provide a clear visual representation of the variables with the most significant impact on the likelihood of a bad buy. This would highlight the importance of factors such as make, model, age, and mileage.
2. **Scatter Plot with Trendline**: An interactive scatter plot with trendlines can visually depict the relationships between age, mileage, and the likelihood of being a bad buy. This would allow stakeholders to intuitively understand how these variables influence the outcome and identify potential thresholds for inspection.
3. **Auction Rating Dashboard**: A dashboard with maps or charts showing the ratings of different auction sources based on historical data can help stakeholders make informed decisions about where to prioritize purchases. This visual tool would enable easy comparison and evaluation of auction sources.
4. **Inspection Threshold Visualization**: A line chart or histogram illustrating the distribution of vehicle age and mileage, overlaid with recommended inspection thresholds, can guide decision-makers in setting appropriate criteria for evaluating potential purchases.

By incorporating these analytical insights and visualizations into decision-making processes, stakeholders can make more informed choices when identifying cars that are likely to be bad buys, ultimately improving overall business outcomes and minimizing risks in the auction market.