```python
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
import os
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import StandardScaler, OneHotEncoder
from sklearn.compose import ColumnTransformer
from sklearn.pipeline import Pipeline
from sklearn.impute import SimpleImputer, KNNImputer
from sklearn.metrics import accuracy_score, classification_report, confusion_matrix, r2_score, roc_auc_score
from sklearn.linear_model import LogisticRegression
from sklearn.tree import DecisionTreeClassifier
from sklearn.ensemble import RandomForestClassifier
import statsmodels.api as sm
from scipy.stats import chi2_contingency
from sklearn.preprocessing import OneHotEncoder

%matplotlib inline
sns.set_theme()
```

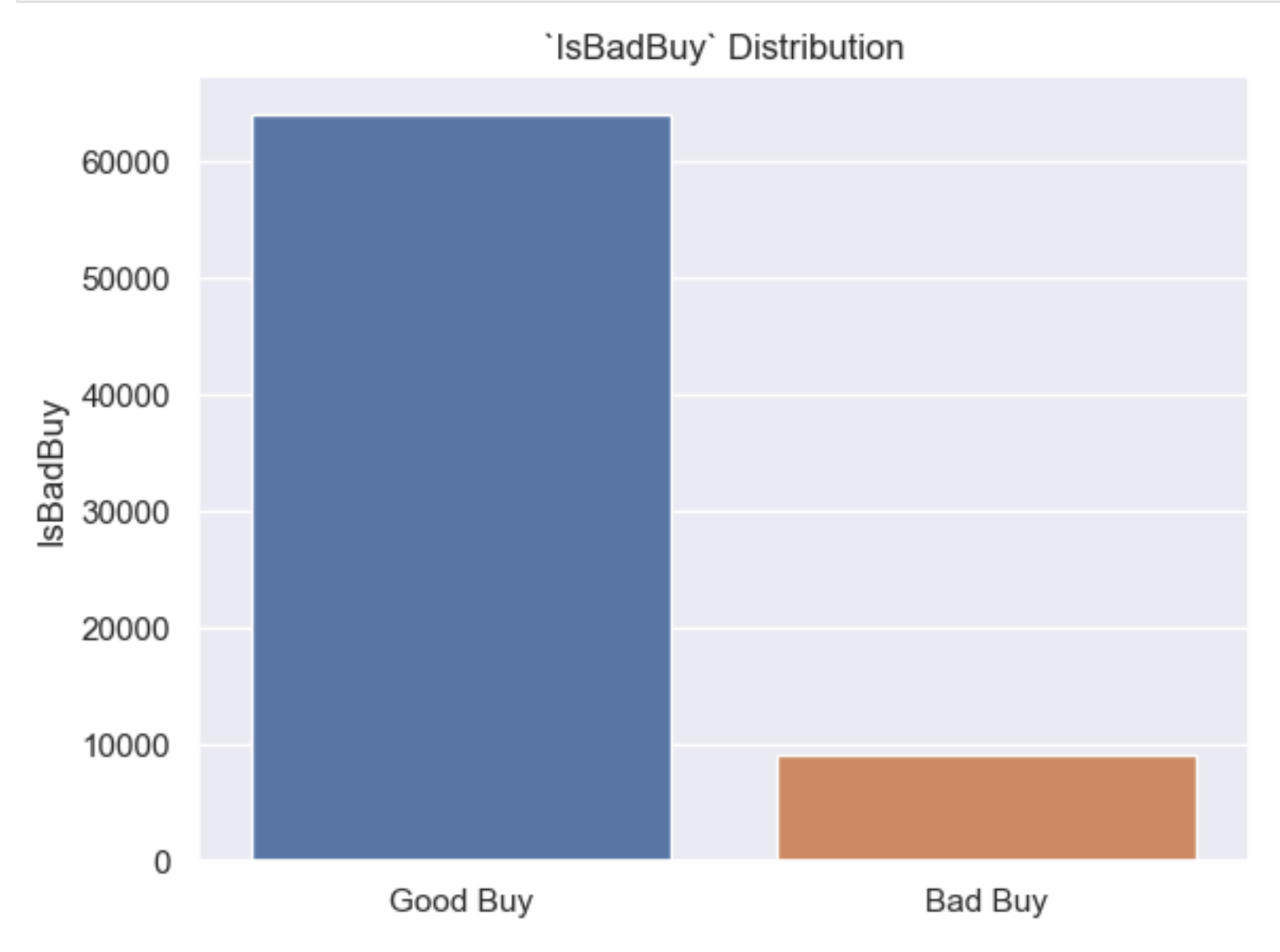
```python
# import data set, check column data types

def get_file_path():
    if os.name == 'nt':  # Windows
        return r'C:\Users\joshu\OneDrive\Desktop Files\Textbooks and Syllabi\CSUN Semester 6\MRKT 656\Case2\case2\training.csv'
    else:  # macOS or other Unix-like OS
        return '/Users/josh/Library/CloudStorage/OneDrive-Personal/Desktop Files/Textbooks and Syllabi/CSUN Semester 6/MRKT 656/Case2/case2/training.csv'

filepath = get_file_path()

df = pd.read_csv(filepath)
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 72983 entries, 0 to 72982
Data columns (total 34 columns):
 #   Column                             Non-Null Count  Dtype
---  ------                             --------------  -----
 0   RefId                              72983 non-null  int64
 1   IsBadBuy                           72983 non-null  int64
 2   PurchDate                          72983 non-null  object
 3   Auction                            72983 non-null  object
 4   VehYear                            72983 non-null  int64
 5   VehicleAge                         72983 non-null  int64
 6   Make                               72983 non-null  object
 7   Model                              72983 non-null  object
 8   Trim                               70623 non-null  object
 9   SubModel                           72975 non-null  object
 10  Color                              72975 non-null  object
 11  Transmission                       72974 non-null  object
 12  WheelTypeID                        69814 non-null  float64
 13  WheelType                          69809 non-null  object
 14  VehOdo                             72983 non-null  int64
 15  Nationality                        72978 non-null  object
 16  Size                               72978 non-null  object
 17  TopThreeAmericanName               72978 non-null  object
 18  MMRAcquisitionAuctionAveragePrice  72965 non-null  float64
 19  MMRAcquisitionAuctionCleanPrice    72965 non-null  float64
 20  MMRAcquisitionRetailAveragePrice   72965 non-null  float64
 21  MMRAcquisitonRetailCleanPrice      72965 non-null  float64
 22  MMRCurrentAuctionAveragePrice      72668 non-null  float64
 23  MMRCurrentAuctionCleanPrice        72668 non-null  float64
 24  MMRCurrentRetailAveragePrice       72668 non-null  float64
 25  MMRCurrentRetailCleanPrice         72668 non-null  float64
 26  PRIMEUNIT                          3419 non-null   object
 27  AUCGUART                           3419 non-null   object
 28  BYRNO                              72983 non-null  int64
 29  VNZIP1                             72983 non-null  int64
 30  VNST                               72983 non-null  object
 31  VehBCost                           72983 non-null  float64
 32  IsOnlineSale                       72983 non-null  int64
 33  WarrantyCost                       72983 non-null  int64
dtypes: float64(10), int64(9), object(15)
memory usage: 18.9+ MB
```

EDA

`IsBadBuy` Profiling

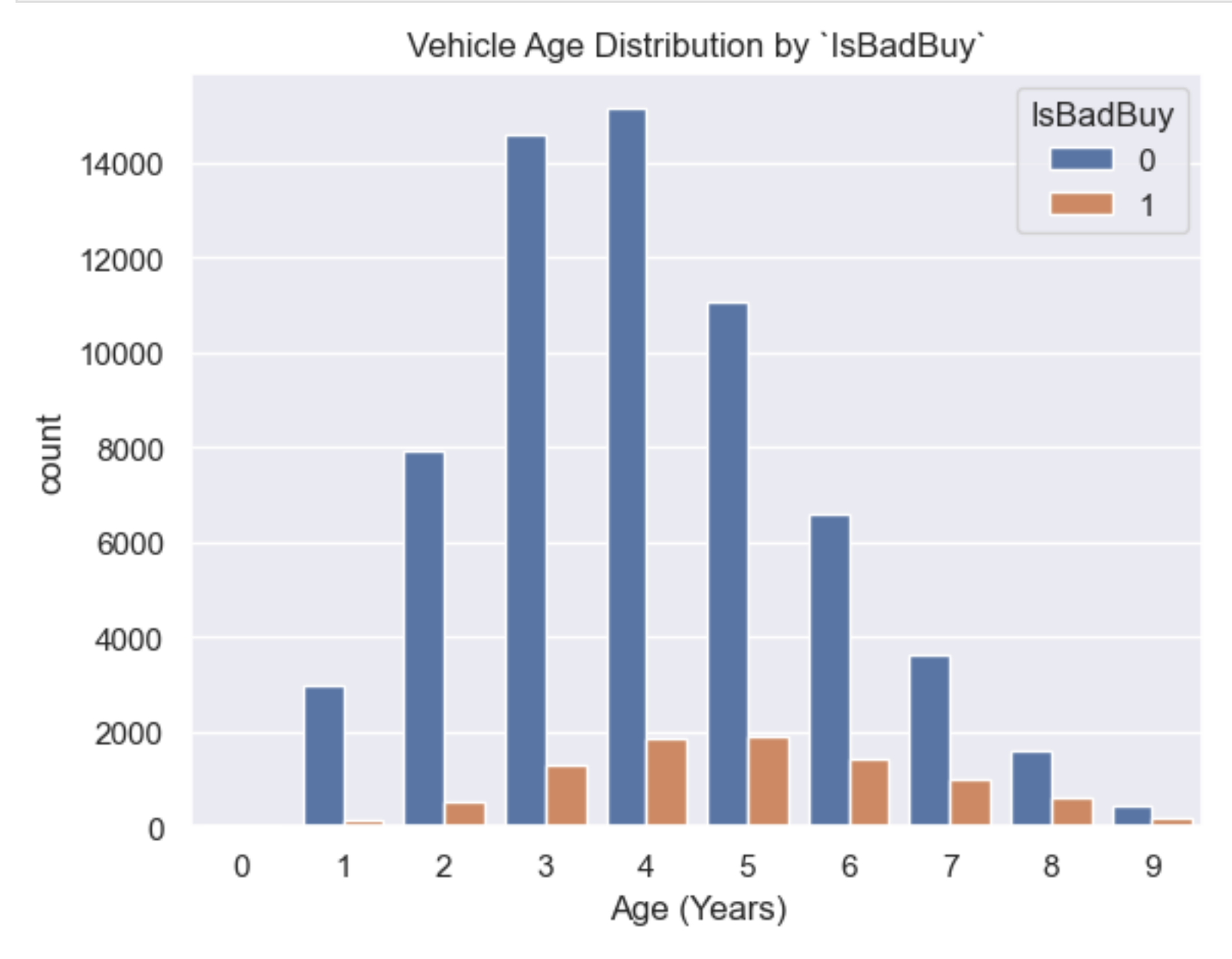
```python
val = df['IsBadBuy'].value_counts()
labels = ['Good Buy', 'Bad Buy']

sns.barplot(x=labels, y=val)
plt.title('`IsBadBuy` Distribution')
plt.show()
```



Dataset is quite unbalanced. Picking the most relevant features for determining bad buys will be important for model performance and accuracy will be good even for `null`/dumb models.

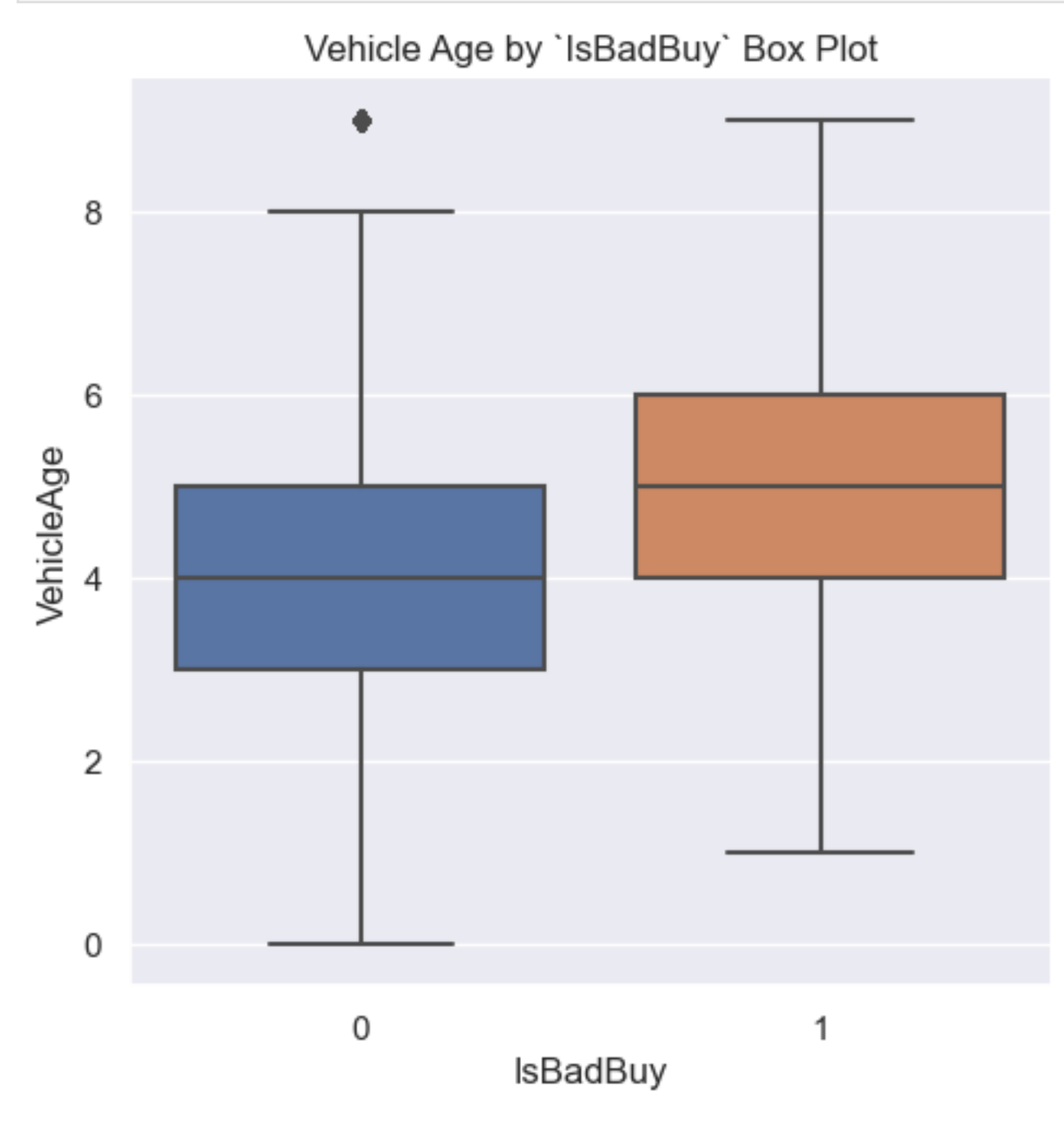
`VehicleAge` Profiling

```python
sns.countplot(data = df, x='VehicleAge', hue='IsBadBuy')

plt.title('Vehicle Age Distribution by `IsBadBuy`')
plt.xlabel('Age (Years)')
plt.show()
```



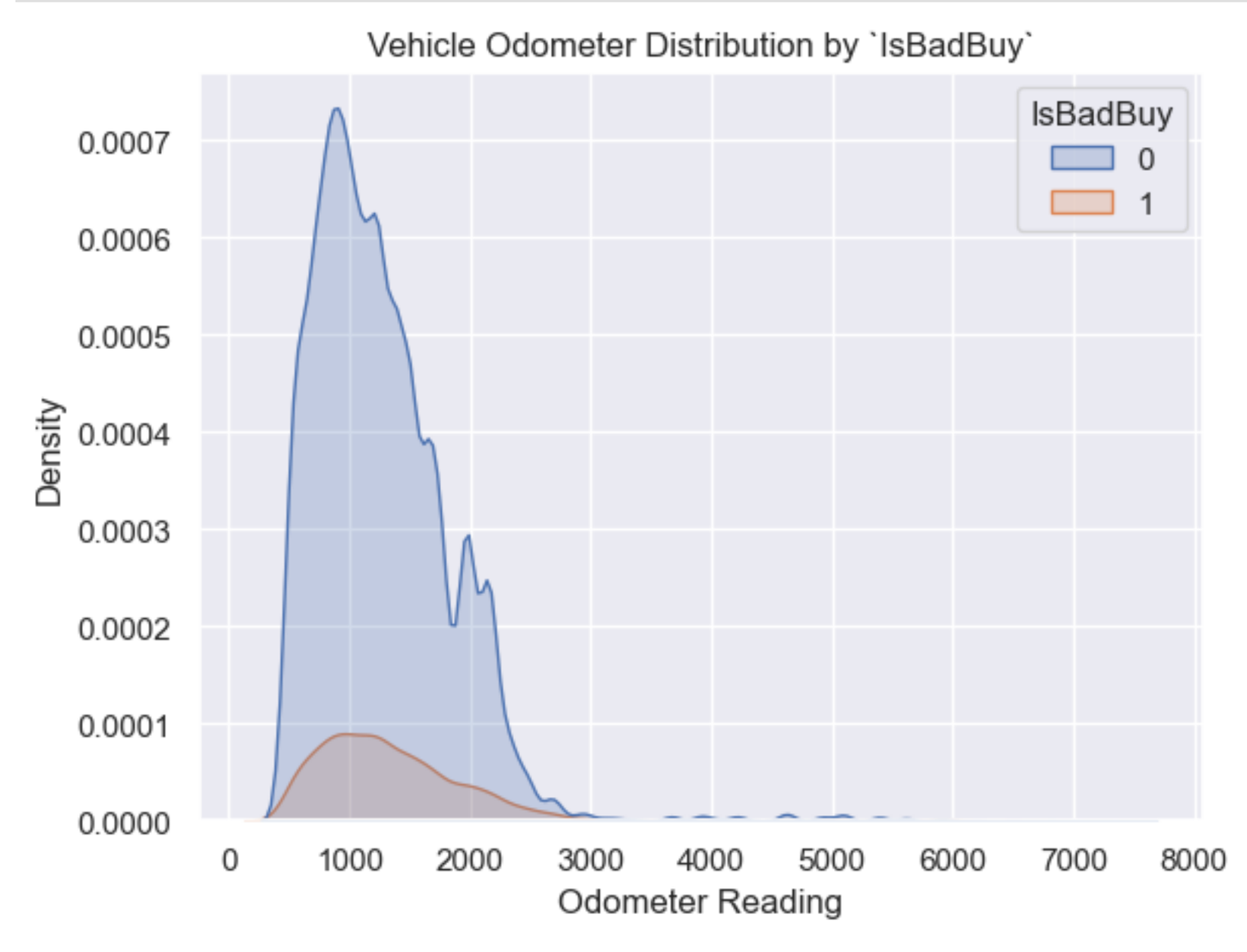
As expected, vehicle age will be an important feature for determing the target.

```python
sns.catplot(data=df, x='IsBadBuy', y='VehicleAge', kind = 'box')

plt.title('Vehicle Age by `IsBadBuy` Box Plot')
plt.show()
```



`VehOdo` Profiling

```python
sns.kdeplot(data = df, x='WarrantyCost', hue='IsBadBuy', fill=True)

plt.title('Vehicle Odometer Distribution by `IsBadBuy`')
plt.xlabel('Odometer Reading')
plt.show()
```

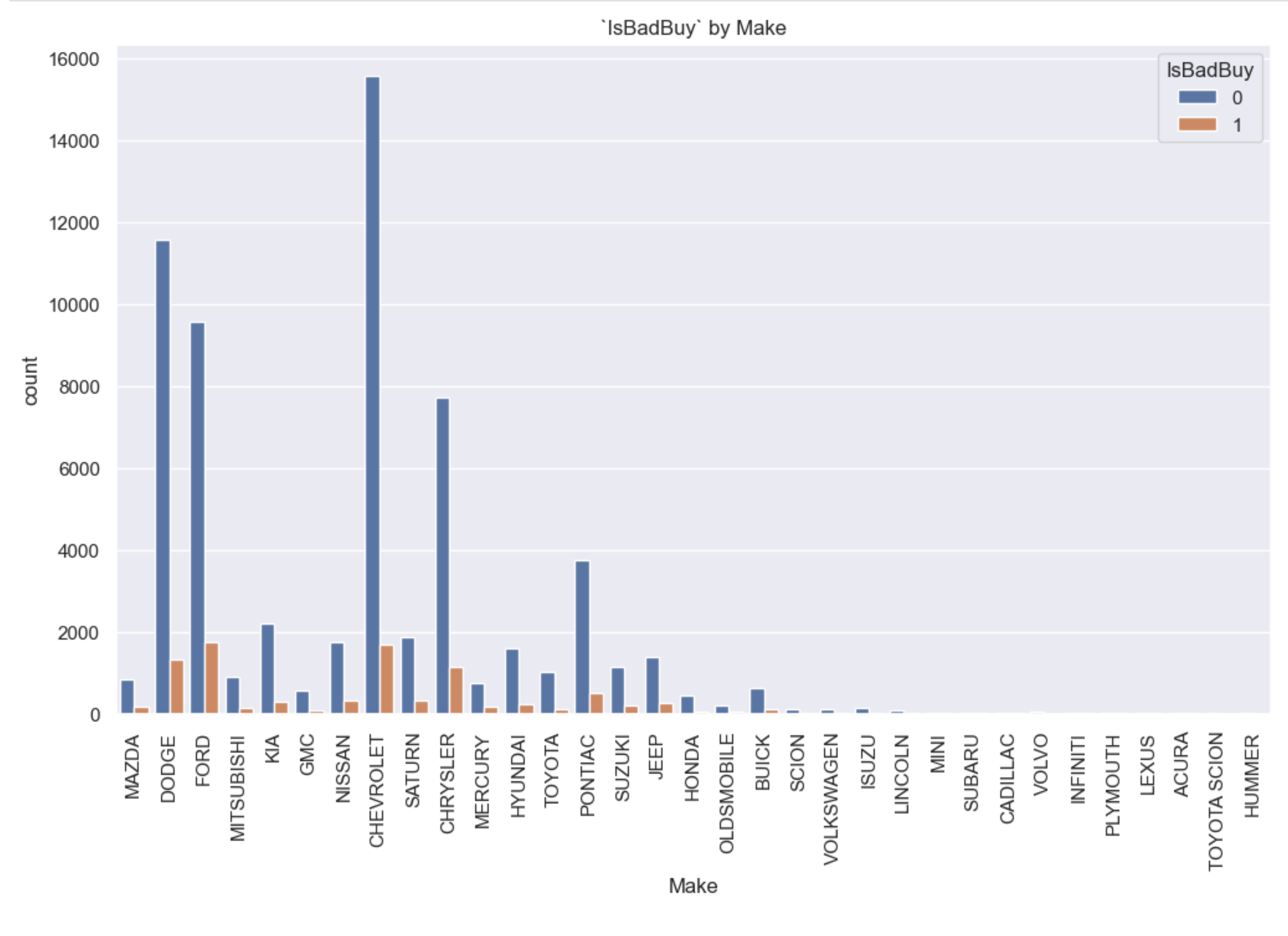



```python
dfBadBuy0 = df[df['IsBadBuy'] == 0]
dfBadBuy1 = df[df['IsBadBuy'] == 1]

odoCompare = pd.concat([dfBadBuy0['WarrantyCost'].describe().round(2),
                         dfBadBuy1['WarrantyCost'].describe().round(2)],
                        axis=1)

odoCompare.columns =['IsBadBuy0_WarrantyCost', 'IsBadBuy1_WarrantyCost']

odoCompare
```

	IsBadBuy0_WarrantyCost	IsBadBuy1_WarrantyCost
count	64007.00	8976.00
mean	1264.85	1360.25
std	585.64	679.89
min	462.00	462.00
25%	834.00	909.00
50%	1155.00	1243.00
75%	1613.00	1647.25
max	7498.00	6492.00

`Make` Profiling

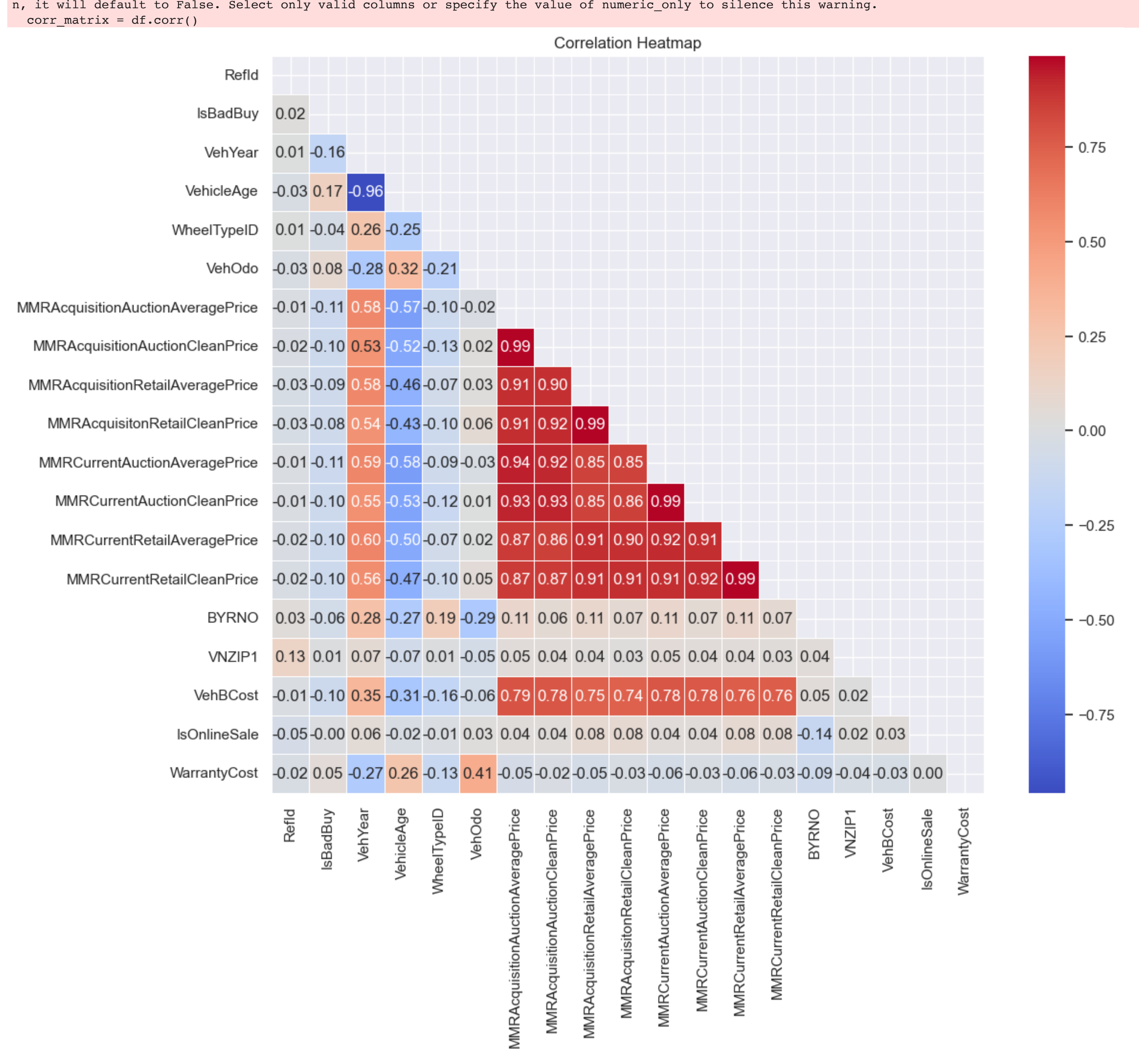
```python
plt.figure(figsize=(12,7))
sns.countplot(data = df, x='Make', hue = 'IsBadBuy')
plt.xticks(rotation=90)

plt.title('`IsBadBuy` by Make')
plt.show()
```



Correlation

```python
corr_matrix = df.corr()

mask = np.triu(np.ones_like(corr_matrix, dtype=bool))

plt.figure(figsize=(12, 10))
sns.heatmap(corr_matrix, annot=True, fmt=".2f", cmap="coolwarm",
            mask=mask, cbar=True, linewidths=.5)
plt.title("Correlation Heatmap")
plt.show()
```

```
C:\Users\joshu\AppData\Local\Temp\ipykernel_18096\3834104031.py:1: FutureWarning: The default value of numeric_only in DataFrame.corr is deprecated. In a future versio
n, it will default to False. Select only valid columns or specify the value of numeric_only to silence this warning.
  corr_matrix = df.corr()
```



Data Preprocessing

Data Definitions

Field Name	Definition
RefID	Unique (sequential) number assigned to vehicles
IsBadBuy	Identifies if the kicked vehicle was an avoidable purchase
PurchDate	The Date the vehicle was Purchased at Auction
Auction	Auction provider at which the vehicle was purchased
VehYear	The manufacturer's year of the vehicle
VehicleAge	The Years elapsed since the manufacturer's year
Make	Vehicle Manufacturer
Model	Vehicle Model
Trim	Vehicle Trim Level
SubModel	Vehicle Submodel
Color	Vehicle Color
Transmission	Vehicles transmission type (Automatic, Manual)
WheelTypeID	The type id of the vehicle wheel
WheelType	The vehicle wheel type description (Alloy, Covers)
VehOdo	The vehicles odometer reading
Nationality	The Manufacturer's country
Size	The size category of the vehicle (Compact, SUV, etc.)
TopThreeAmericanName	Identifies if the manufacturer is one of the top three American manufacturers
MMRAcquisitionAuctionAveragePrice	Acquisition price for this vehicle in average condition at time of purchase
MMRAcquisitionAuctionCleanPrice	Acquisition price for this vehicle in the above Average condition at time of purchase
MMRAcquisitionRetailAveragePrice	Acquisition price for this vehicle in the retail market in average condition at time of purchase
MMRAcquisitonRetailCleanPrice	Acquisition price for this vehicle in the retail market in above average condition at time of purchase
MMRCurrentAuctionAveragePrice	Acquisition price for this vehicle in average condition as of current day
MMRCurrentAuctionCleanPrice	Acquisition price for this vehicle in the above condition as of current day
MMRCurrentRetailAveragePrice	Acquisition price for this vehicle in the retail market in average condition as of current day
MMRCurrentRetailCleanPrice	Acquisition price for this vehicle in the retail market in above average condition as of current day
PRIMEUNIT	Identifies if the vehicle would have a higher demand than a standard purchase
AUCGUART	The level guarantee provided by auction for the vehicle (Green light - Guaranteed/arbitratable, Yellow Light - caution/issue, red light - sold as is)
KickDate	Date the vehicle was kicked back to the auction
BYRNO	Unique number assigned to the buyer that purchased the vehicle
VNZIP1	Zipcode where the car was purchased
VNST	State where the the car was purchased
VehBCost	Acquisition cost paid for the vehicle at time of purchase
IsOnlineSale	Identifies if the vehicle was originally purchased online
WarrantyCost	Warranty price (term=36month and mileage=36K)

Series to Drop

- `RefId`: UID, not relevant to determine `IsBadBuy`
- `BYRNO`: UID, not relevant to determine `IsBadBuy`
- `VNZIP1`: UID, not relevant to determine `IsBadBuy`
- `VNST`: UID, not relevant to determine `IsBadBuy`
- `VehYear`: Feature already captured by `VehAge` (drop to remove collinearity)