In [1773]:
```
# -*- coding: utf-8 -*-
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
import re
from collections import defaultdict
from sklearn.feature_extraction.text import TfidfVectorizer

from sklearn.neural_network import MLPRegressor
import statsmodels.api as sm

from sklearn.preprocessing import (LabelEncoder, OneHotEncoder,
                                   PolynomialFeatures, StandardScaler,
                                   label_binarize,MultiLabelBinarizer)

from scipy.sparse import csr_matrix

#from tensorflow.keras.models import Sequential
#from tensorflow.keras.layers import Dense
#from tensorflow.keras.wrappers.scikit_learn import KerasClassifier
from skopt import gp_minimize
from skopt import BayesSearchCV
from skopt.space import Real, Integer, Categorical
from sklearn.base import BaseEstimator, RegressorMixin

from importlib import reload

import trainers as tr
import price_calculations as pr
import neural_models as nm
import xgboost as xgb

import matplotlib.pyplot as plt
```

In [1774]:
```
# Reload Trainers due to possibility of local changes
reload(tr)
reload(pr)
reload(nm)
```

Out[1774]:
```
<module 'neural_models' from 'C:\\Users\\aflyn\\repos\\Boeing_ML_Pred\\vehicl
e_predictions\\neural_models.py'>
```

In [1775]:

```
def engineerTestData(df,log_cols,encoded_cols,freq_cols,
                     mask_cols,token_cols,orig_cols, feats_to_drop,
                     coder,token_1,token_2):
    df.columns =  df.columns.str.lower()
    df.columns = [col.strip() for col in df.columns]
    df.set_index('listingid',inplace=True)
    df = df.apply(lambda col: col.fillna('Unknown') if col.dtype == 'O' else c
ol.fillna(0))

    [df.__setitem__(col, np.log(np.ceil(df[col]))) for col in log_cols]
    [df.__setitem__(col, df[col].map(df[col].value_counts())) for col in freq_
cols]
    [df.__setitem__(col, df[col].astype(int)) for col in mask_cols]

    handle_encode = orig_cols
    test_encode = df[encoded_cols]
    for col in handle_encode:
        col = col.strip()
        func_name = 'handle_' + col  # Prepare function name
        if func_name in globals() and callable(globals()[func_name]):
            func = globals()[func_name]
            if col == 'vehdrivetrain':
                temp_df = func(df[col].copy())  # Call the function dynamicall
y
                test_encode[col] = temp_df
            elif col == 'vehhistory':
                df.loc[df[col] == 'Unknown', col] = 0
                temp_df = df[col].copy().str.split(',',n=1,expand=True)
                temp_df.columns = ['Owners', 'History']
                temp_df['Owners'] = temp_df['Owners'].str.extract(r'^(\d+)')
                encoded_hist = func(temp_df['History'])  # Call the function d
ynamically
                df['owners'] = temp_df['Owners']
                df[encoded_hist.columns] = encoded_hist
            elif col == 'vehengine':
                temp_df = func(df[col].copy())  # Call the function dynamicall
y
                temp_df.columns = temp_df.columns.str.lower()
                df[temp_df.columns] = temp_df
            elif col == 'vehcolorext':
                col_temp = func(df[col].copy())  # Call the function dynamical
ly
                col_temp.columns = col_temp.columns.str.lower()
            elif col == 'vehcolorint':
                col_tmp = func(df[col].copy())  # Call the function dynamicall
y
                col_tmp.columns = col_tmp.columns.str.lower()
        else:
            print(f"Function '{func_name}' does not exist or is not callabl
e.")

    colors = pd.merge(col_temp,col_tmp,left_index=True, right_index=True)
    df = pd.merge(df,colors,left_index=True, right_index=True)
    temp_encoded = oHotEncode(test_encode,coder)
    df.drop(columns=encoded_cols,inplace=True)
    df = pd.merge(df,temp_encoded,left_index=True, right_index=True)
```

```
    df.columns = df.columns.astype(str)

    tf1 = tf_idfTokenizer(df[token_cols[0]].copy(),token_1)
    tf2 = tf_idfTokenizer(df[token_cols[1]].copy(),token_2)
    #tfs = pd.concat([tf1, tf2])
    #tfs = combined_tf.loc[:,~combined_tf.columns.duplicated()]
    tfs = pd.merge(tf1, tf2,left_index=True,right_index=True)
    df = pd.merge(df,tfs,left_index=True,right_index=True)
    df.drop(columns=feats_to_drop,inplace=True)

    return df

def oHotEncode(df_,coder):
    encoded_mat = coder.transform(df_)
    return pd.DataFrame(encoded_mat.todense(),
                        columns=[cat for columns in coder.categories_ for cat
in columns],
                        index=df_.index)


def zScoreTransform(col):
    return np.divide(np.subtract(col,col.mean()),col.std())


def tf_idfTokenizer(df_,tfidf):
    tf_mat = tfidf.transform(df_)
    return pd.DataFrame(tf_mat.toarray(),
                          columns=tfidf.get_feature_names_out(['feature']),
                          index=df_.index)


def setFeatPtr(data,index):
    return data.iloc[:,index],data.columns[index]


def plotDist(data,title):
    # Plotting a histogram of frequencies
    fig, ax = plt.subplots()
    sns.histplot(data, kde=True, ax=ax)
    ax.set_xlabel('Values')
    ax.set_ylabel('Frequency')
    ax.set_title(title)
    plt.show()

def categorize_train(phrase, awd_pattern, fwd_pattern, wd_pattern):
    if re.search(awd_pattern, phrase) and re.search(wd_pattern, phrase):
        return 'hybrid'
    elif re.search(awd_pattern, phrase):
        return 'awd'
    elif re.search(fwd_pattern, phrase):
        return 'fwd'
    elif re.search(wd_pattern, phrase):
        return '_4_wd'
    else:
        return 'Unknown'

def handle_vehdrivetrain(df):
```

```
    df = df.str.lower()
    awd_pattern = re.compile(r'awd|all', flags=re.IGNORECASE)
    fwd_pattern = re.compile(r'fwd|front', flags=re.IGNORECASE)
    wd_pattern = re.compile(r'4x4|4wd|four\s?WHEEL\s?DRIVE\b', flags=re.IGNORE
CASE)
    return df.apply(categorize_train, args=(awd_pattern, fwd_pattern, wd_patte
rn))

# Function extracts engine size and configuration
def categorize_engine(phrase):
    engine_size_match = re.search(r'\b\d+(\.\d+)?\s*L\b', phrase)  # Matches p
attern with number (with or without decimal) followed by L
    config_match = re.search(r'V[-]?6|V[-]?8|\b\d\s*cylinder|\b6\s*cylinde', p
hrase, re.IGNORECASE)  # Matches V6, V-6, V8, V-8, or a number followed by cyl
inders

    if engine_size_match:
        engine_size = float(re.search(r'\d+(\.\d+)?', engine_size_match.group
()).group())  # Extracts engine size
        size_category = engine_size  # Assigning the engine size directly as t
he size category
    else:
        size_category = 0

    if config_match:
        config_str = config_match.group().upper()
        config = 6 if '6' in config_str else 8  # Assign 6 or 8 based on the p
resence of 'Vx' or 'Cyclinders'
    else:
        config = 0

    return size_category, config

def handle_vehengine(df):
    extracted_info = df.apply(categorize_engine)
    # Convert the extracted information into a DataFrame
    df = pd.DataFrame(extracted_info.tolist(), columns=['EngineSize', 'Cylinde
rs'], index=df.index)
    return df

def handle_vehhistory(df):
    print("HISTORY")
    # List of unique phrases
    unique_phrases = [
        'Accident(s) Reported',
        'Buyback Protection Eligible',
        'Non-Personal Use Reported',
        'Title Issue(s) Reported'
    ]

    # Strip whitespace from each string in the Series
    df = df.astype(str).str.strip()

    # Initialize a DataFrame to store the encoded values
    encoded_df = pd.DataFrame(index=df.index)

    # Iterate over each unique phrase
```

```
    for phrase in unique_phrases:
        # Check if the phrase exists in each row and create a binary indicator
        encoded_df[phrase] = df.apply(lambda x: 1 if phrase in x else 0)

    # Create a 'None of the above' column to indicate if none of the phrases w
ere found
    encoded_df['None of the above'] = (encoded_df.sum(axis=1) == 0).astype(in
t)

    return encoded_df

def handle_vehcolorext(df_):
    print("COLOR")
    common_colors = ['Black', 'Blue', 'Brown', 'Gray', 'Green', 'Steel', 'Meta
llic','Pearlcoat', 'Clearcoat',
                     'Charcoal','Granite', 'Red', 'Silver', 'White']
    silver_colors = ['Gray', 'Steel', 'Charcoal', 'Silver']

    temp = pd.DataFrame(index=df_.index)
    for color in common_colors:
        temp[f'{color}'] = df_.str.contains(color, case=False).astype(int)

    # Grouping similar silver colors into a single 'Silver' category
    temp['Silver'] = df_.str.contains('|'.join(silver_colors), case=False).ast
ype(int)
    temp.drop([col for col in silver_colors if col != 'Silver'], axis=1, inpla
ce=True)

    # Populates a 'None' category if none of the common colors are present
    temp['None'] = 1 - temp[[f'{color}' for color in temp.columns]].max(axis=
1)

    return temp

def handle_vehcolorint(df_):
    print("COLOR2")
    common_colors = ['Black', 'Blue', 'Brown', 'Gray', 'Steel', 'Beige','tri
m',
                     'Charcoal','Red', 'Silver', 'Frost','Maple','Tan','Cirru
s','carbon','plum']
    silver_colors = ['Gray', 'Steel', 'Charcoal', 'Silver']
    temp = pd.DataFrame(index=df_.index)
    for color in common_colors:
        temp[f'{color}'] = df_.str.contains(color, case=False).astype(int)

    # Grouping similar silver colors into a single 'Silver' category
    temp['Silver'] = df_.str.contains('|'.join(silver_colors), case=False).ast
ype(int)
    temp.drop([col for col in silver_colors if col != 'Silver'], axis=1, inpla
ce=True)

    # Populates a 'None' category if none of the common colors are present
    temp['None'] = 1 - temp[[f'{color}' for color in temp.columns]].max(axis=
1)

    return temp
```

```
def calculate_age(df_):
    age = 2024 - df_
    return age
```

In [1776]:

```
#Initialize training and test dataframes
orig_train = pd.read_csv('Training_DataSet.csv')
df_test = pd.read_csv('Test_Dataset.csv')


#Drop blank cells from training set to clean up data (contemplated using mean,
median, or mode imputation,
#but will explore without corrupting the data and due to the large size of the
dataset eliminating
#some rows should suffice
orig_train.dropna(axis=0,how='any',inplace=True) #EXPLICIT CALL TO DROP ROWS W
ITH A SINGLE MISSING VALUE
                                                 #(DEFAULT CALL DOES SAME)
```

In [1777]:
```
orig_train.columns = orig_train.columns.str.lower()
orig_train.set_index('listingid',inplace=True)

df_train = orig_train.copy()
df_train.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Index: 5045 entries, 4777 to 8620012
Data columns (total 28 columns):
 #   Column                Non-Null Count  Dtype
---  ------                --------------  -----
 0   sellercity            5045 non-null   object
 1   sellerispriv          5045 non-null   bool
 2   sellerlistsrc         5045 non-null   object
 3   sellername            5045 non-null   object
 4   sellerrating          5045 non-null   float64
 5   sellerrevcnt          5045 non-null   int64
 6   sellerstate           5045 non-null   object
 7   sellerzip             5045 non-null   float64
 8   vehbodystyle          5045 non-null   object
 9   vehcertified          5045 non-null   bool
 10  vehcolorext           5045 non-null   object
 11  vehcolorint           5045 non-null   object
 12  vehdrivetrain         5045 non-null   object
 13  vehengine             5045 non-null   object
 14  vehfeats              5045 non-null   object
 15  vehfuel               5045 non-null   object
 16  vehhistory            5045 non-null   object
 17  vehlistdays           5045 non-null   float64
 18  vehmake               5045 non-null   object
 19  vehmileage            5045 non-null   float64
 20  vehmodel              5045 non-null   object
 21  vehpricelabel         5045 non-null   object
 22  vehsellernotes        5045 non-null   object
 23  vehtype               5045 non-null   object
 24  vehtransmission       5045 non-null   object
 25  vehyear               5045 non-null   int64
 26  vehicle_trim          5045 non-null   object
 27  dealer_listing_price  5045 non-null   float64
dtypes: bool(2), float64(5), int64(2), object(19)
memory usage: 1.0+ MB
```

In [1778]:
```
#NOTICE THERES ONLY JEEPS AND CADILLACS IN DATA SET BRAKE THEM UP FURTHER TO S
EE
#THE TRIMS SINCE TRIMS ARE USUALLY EXCLUSIVE TO MANUFACTURER LINE
jeeps = df_train[df_train['vehmake'].str.lower() == 'jeep'].copy()
caddy = df_train[df_train['vehmake'].str.lower() == 'cadillac'].copy()

print(jeeps['vehicle_trim'].value_counts())
print(caddy['vehicle_trim'].value_counts())
```

```
vehicle_trim
Limited                              1616
Laredo                                616
Overland                              342
Altitude                              257
Summit                                203
Trailhawk                             149
High Altitude                          74
SRT                                    62
Laredo E                               42
Trackhawk                              27
Sterling Edition                       25
Limited 75th Anniversary Edition        5
75th Anniversary                        4
Upland                                  3
SRT Night                               2
75th Anniversary Edition                1
Name: count, dtype: int64
vehicle_trim
Premium Luxury        658
Luxury                535
Base                  137
Platinum              116
Luxury FWD             49
FWD                    47
Premium Luxury FWD     35
Luxury AWD             19
Platinum AWD           12
Premium Luxury AWD      9
Name: count, dtype: int64
```

In [1779]:
```
#MASSIVE CLASS IMBALANCE WILL NEED TO CONDENSE THIS AND IGNORE LOW FREQUENCY C
LASSES
#BECAUSE THEY ADD NOISE AND CLASSIFIER WILL NOT BE ABLE TO ARBITRATE
conditions = [
    caddy['vehicle_trim'].str.lower().str.contains('premium'),
    caddy['vehicle_trim'].str.lower().str.contains('luxury'),
    caddy['vehicle_trim'].str.lower().str.contains('base'),
    caddy['vehicle_trim'].str.lower().str.contains('platinum')
]

choices = ['Premium Luxury', 'Luxury', 'Base', 'Platinum']

# Use np.select() to relabel based on conditions
caddy['vehicle_trim'] = np.select(conditions, choices, default='other')

# Filter the DataFrame to keep only rows labeled as 'premium', 'luxury', 'bas
e', or 'platinum'
valid_labels = ['Premium Luxury', 'Luxury', 'Base', 'Platinum']
caddy = caddy[caddy['vehicle_trim'].isin(valid_labels)]
caddy["vehicle_trim"]
```

Out[1779]:
```
listingid
7108               Luxury
21448      Premium Luxury
21807              Luxury
30524                Base
34061      Premium Luxury
                 ...
8599564    Premium Luxury
8601212            Luxury
8604205    Premium Luxury
8616294            Luxury
8617378            Luxury
Name: vehicle_trim, Length: 1570, dtype: object
```

In [1780]:
```
valid_labels_jeep = ['limited', 'laredo',  'summit',
                     'overland', 'altitude','trailhawk',
                     'trackhawk','srt','sterling']

conditions_jeep = [
    jeeps['vehicle_trim'].str.lower().str.contains(label) for label in valid_l
abels_jeep
]

choices_jeep = ['Limited', 'Laredo',  'Summit',
                     'Overland', 'Altitude','Trailhawk', 'Trackhawk',
                        'SRT','Sterling Edition']

# Use np.select() to classify based on conditions
jeeps['vehicle_trim'] = np.select(conditions_jeep, choices_jeep, default='othe
r')
print(jeeps['vehicle_trim'].value_counts())
# Filter the DataFrame to keep only rows labeled with valid labels
jeeps = jeeps[jeeps['vehicle_trim'].isin(choices_jeep)]
jeeps["vehicle_trim"]
```

```
vehicle_trim
Limited             1621
Laredo               658
Overland             342
Altitude             331
Summit               203
Trailhawk            149
SRT                   64
Trackhawk             27
Sterling Edition      25
other                  8
Name: count, dtype: int64
```

Out[1780]:
```
listingid
4777          Laredo
6242         Limited
10882        Limited
12013         Laredo
12334        Limited
             ...
8610847      Limited
8612731     Altitude
8614177      Limited
8615510      Limited
8620012    Trailhawk
Name: vehicle_trim, Length: 3420, dtype: object
```

In [1781]:
```
print("CADDY")
print(caddy["vehicle_trim"].value_counts())
print("JEEP")
print(jeeps["vehicle_trim"].value_counts())
```

```
CADDY
vehicle_trim
Premium Luxury    702
Luxury            603
Base              137
Platinum          128
Name: count, dtype: int64
JEEP
vehicle_trim
Limited             1621
Laredo               658
Overland             342
Altitude             331
Summit               203
Trailhawk            149
SRT                   64
Trackhawk             27
Sterling Edition      25
Name: count, dtype: int64
```

In [1782]:
```
print(jeeps.index)
print(caddy.index)
df_train.update(jeeps[['vehicle_trim']])
df_train.update(caddy[['vehicle_trim']])
df_train["vehicle_trim"].value_counts()
```

```
Index([   4777,    6242,   10882,   12013,   12334,   13173,   17626,   1798
2,
         20984,   25753,
       ...
       8588930, 8589426, 8595032, 8604328, 8605237, 8610847, 8612731, 861417
7,
       8615510, 8620012],
      dtype='int64', name='listingid', length=3420)
Index([   7108,   21448,   21807,   30524,   34061,   34857,   48367,   5252
8,
         56883,   60529,
       ...
       8567162, 8567842, 8577448, 8584119, 8597526, 8599564, 8601212, 860420
5,
       8616294, 8617378],
      dtype='int64', name='listingid', length=1570)
```

Out[1782]:

vehicle_trim	
Limited	1621
Premium Luxury	702
Laredo	658
Luxury	603
Overland	342
Altitude	331
Summit	203
Trailhawk	149
Base	137
Platinum	128
SRT	64
FWD	47
Trackhawk	27
Sterling Edition	25
75th Anniversary	4
Upland	3
75th Anniversary Edition	1

Name: count, dtype: int64

In [1783]:
```
options = choices + choices_jeep
df_train = df_train[df_train['vehicle_trim'].isin(options)]

print(df_train["vehicle_trim"].value_counts())
df_train.head()
```

```
vehicle_trim
Limited             1621
Premium Luxury       702
Laredo               658
Luxury               603
Overland             342
Altitude             331
Summit               203
Trailhawk            149
Base                 137
Platinum             128
SRT                   64
Trackhawk             27
Sterling Edition      25
Name: count, dtype: int64
```

Out[1783]:

listingid	sellercity	sellerispriv	sellerlistsrc	sellername	sellerrating	sellerrevcnt	sellerstate	se
4777	Waukesha	False	Jeep Certified Program	Wilde Chrysler Jeep Dodge Ram & Subaru	4.8	1405	WI	5
6242	Wentzville	False	Inventory Command Center	Century Dodge Chrysler Jeep RAM	4.4	21	MO	6
7108	Fayetteville	False	HomeNet Automotive	Superior Buick GMC of Fayetteville	3.7	74	AR	7
10882	Olean	False	Digital Motorworks (DMi)	Paul Brown Chrysler Dodge Jeep RAM Kia	3.0	51	NY	1
12013	Ottawa	False	Digital Motorworks (DMi)	Sierra Motor Mall	3.5	17	IL	6

5 rows × 28 columns

In [1784]:
```
feats_to_drop = []
encoded_cols = []
freq_cols = []
same_cols = []
mask_cols = []
log_cols = []
orig_cols = []


jeeps = df_train[orig_train["vehmake"]=="Jeep"].copy()
caddys = df_train[df_train["vehmake"]=="Cadillac"]

input_jeeps = jeeps.copy()
input_jeeps = input_jeeps.iloc[:,:-2]

input_caddys = caddys.copy()
input_caddys = input_caddys.iloc[:,:-2]

col = 0
feat_ptrj,column = setFeatPtr(input_jeeps,col)
feat_ptrc,column = setFeatPtr(input_caddys,col)
```

```
C:\Users\aflyn\AppData\Local\Temp\ipykernel_19256\180486138.py:10: UserWarnin
g: Boolean Series key will be reindexed to match DataFrame index.
  jeeps = df_train[orig_train["vehmake"]=="Jeep"].copy()
```

In [1785]:
```
feat_ptrj.head()
```

Out[1785]:
```
listingid
4777       Waukesha
6242     Wentzville
10882         Olean
12013        Ottawa
12334      Elmhurst
Name: sellercity, dtype: object
```

In [1786]:
```
feat_ptrc.head()
```

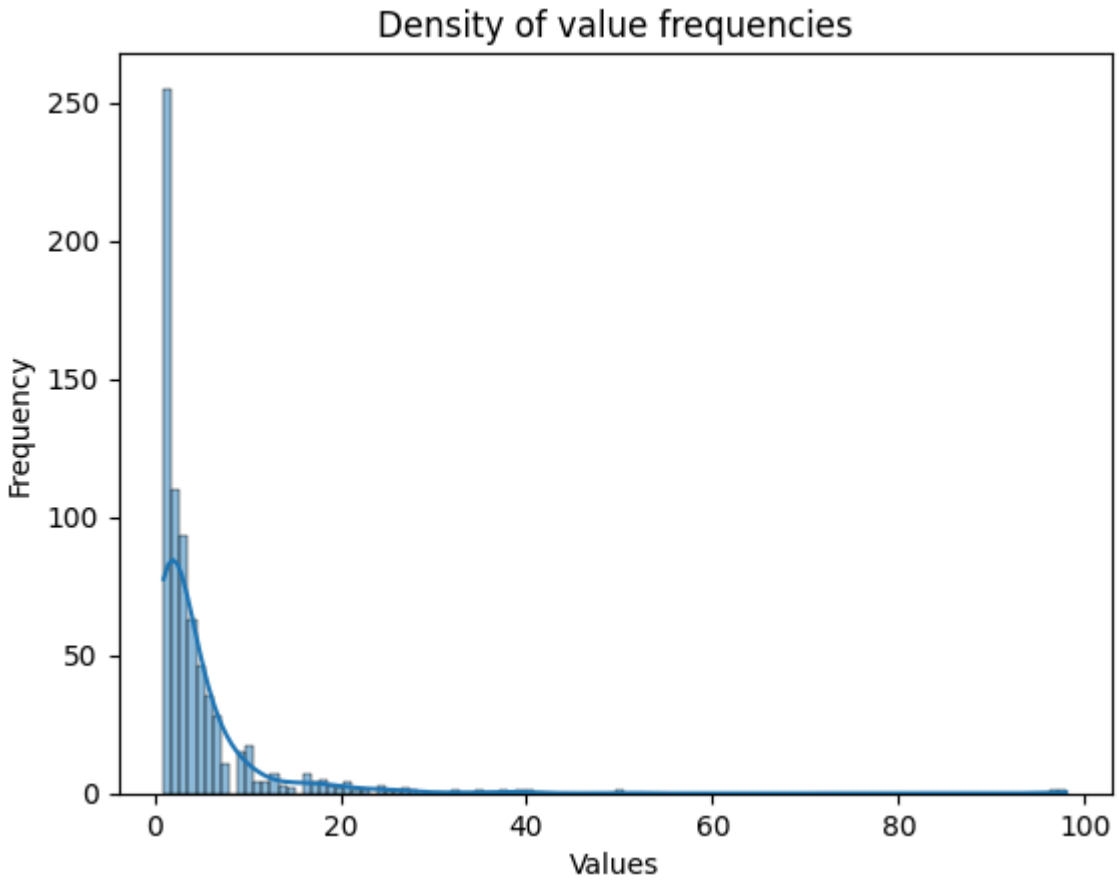
Out[1786]:
```
listingid
7108     Fayetteville
21448     New Orleans
21807       Bradenton
30524      Fort Worth
34061         Baytown
Name: sellercity, dtype: object
```

In [1787]:
```
#PERCENTAGE MODE APPEARS
count = (feat_ptrj==feat_ptrj.mode()[0]).sum()
print(count/len(feat_ptrj))
print(feat_ptrj.nunique())
count = (feat_ptrc==feat_ptrc.mode()[0]).sum()
print(count/len(feat_ptrc))
print(feat_ptrc.nunique())
```

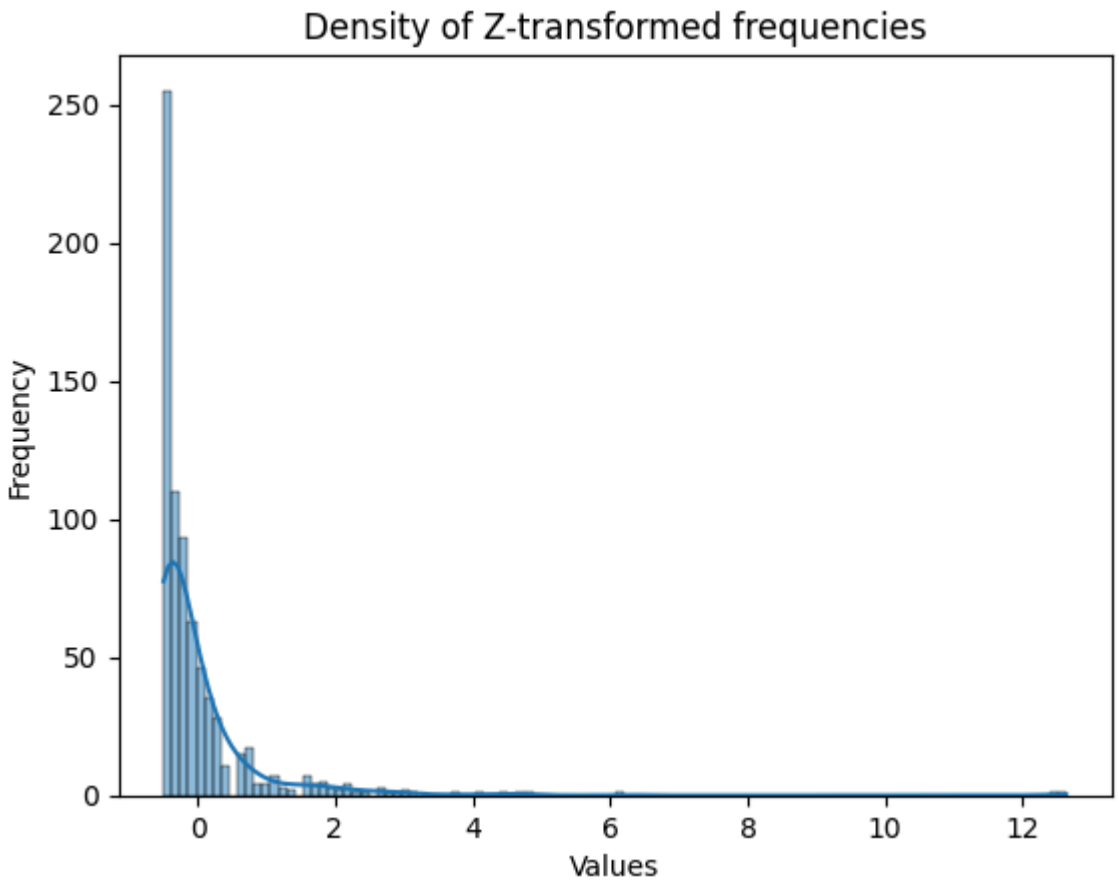
```
0.02865497076023392
736
0.02356687898089172
621
```

In [1788]:

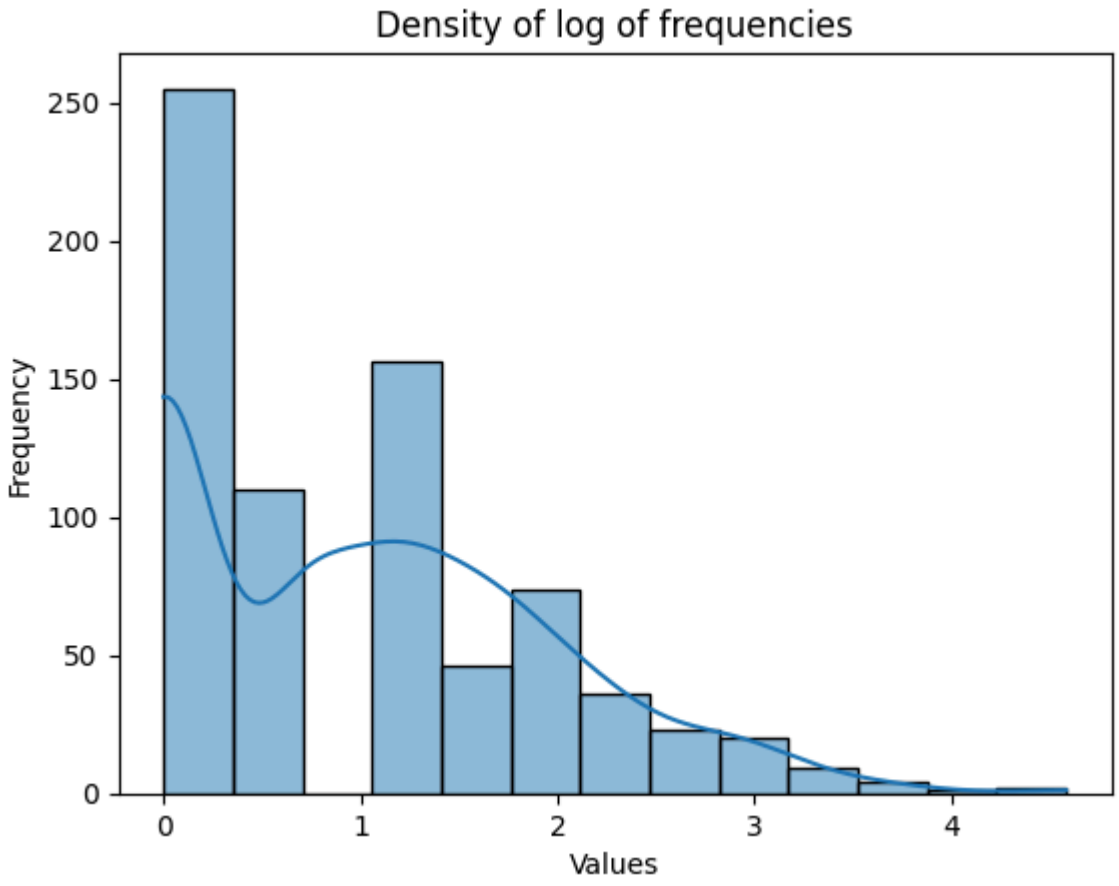
```
value_counts = feat_ptrj.value_counts()
# Plotting a histogram of frequencies (Frequencies of Frequencies)
plotDist(value_counts,"Density of value frequencies")
#FREQUENCY ENCODE THESE VALUES AND THEN TAKE Z SCORE OR THE FREQUENCIES
zvalues = zScoreTransform(value_counts)
print(zvalues)
plotDist(zvalues,"Density of Z-transformed frequencies")
# Assuming 'value_counts' contains the frequencies
log_frequencies = np.log(value_counts)
# Plotting the density plot of the log of frequencies
plt.figure(figsize=(8, 6))
plotDist(log_frequencies,"Density of log of frequencies")
# Plotting the density plot of the log of frequencies
zlog = zScoreTransform(log_frequencies)
plotDist(zlog,"Density of Z-transformed log of frequencies")
freq = feat_ptrj.value_counts().to_dict()
feat_ptrj = feat_ptrj.map(freq)
feat_ptrj.head()
```

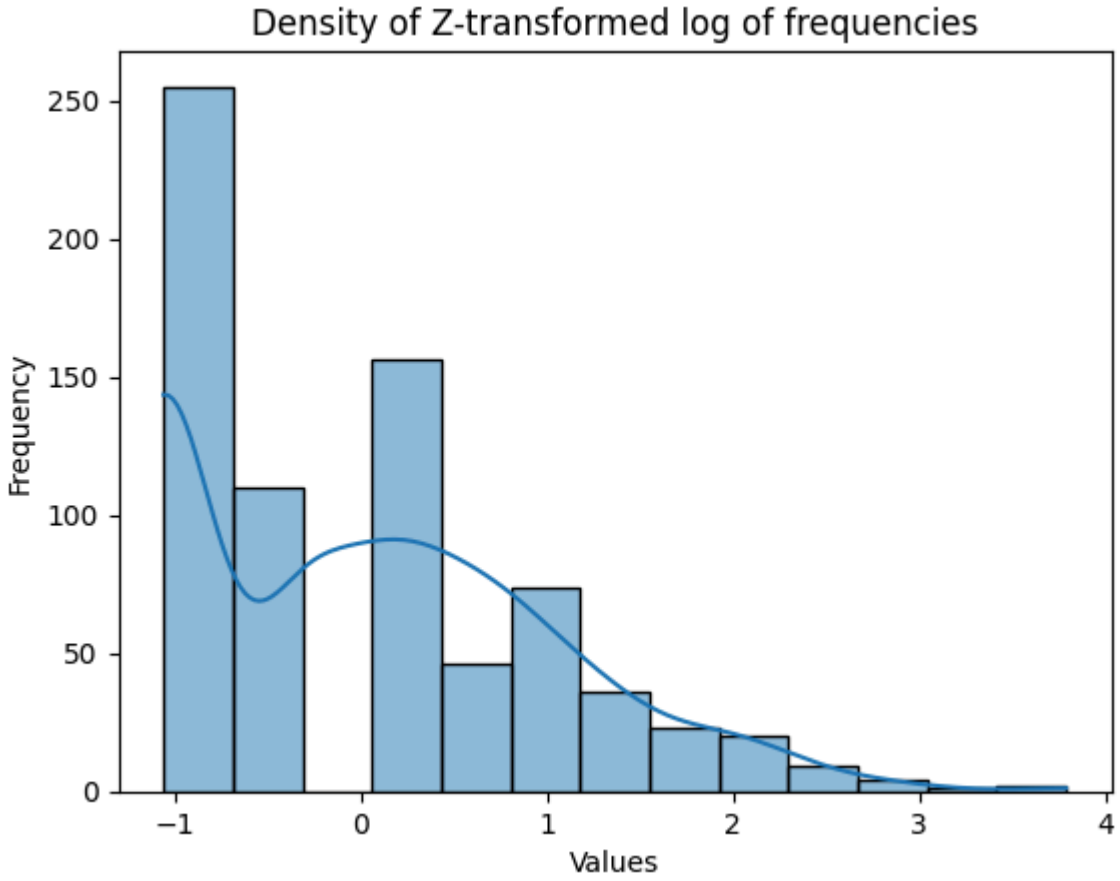



```
sellercity
Battle Creek    12.629819
Chicago         12.494528
Louisville       6.135870
Atlanta          4.782964
Richmond         4.647673
                  ...
Florissant      -0.493370
Mansfield       -0.493370
Bridgeville     -0.493370
West Chicago    -0.493370
Vincennes       -0.493370
Name: count, Length: 736, dtype: float64
```



<Figure size 800x600 with 0 Axes>

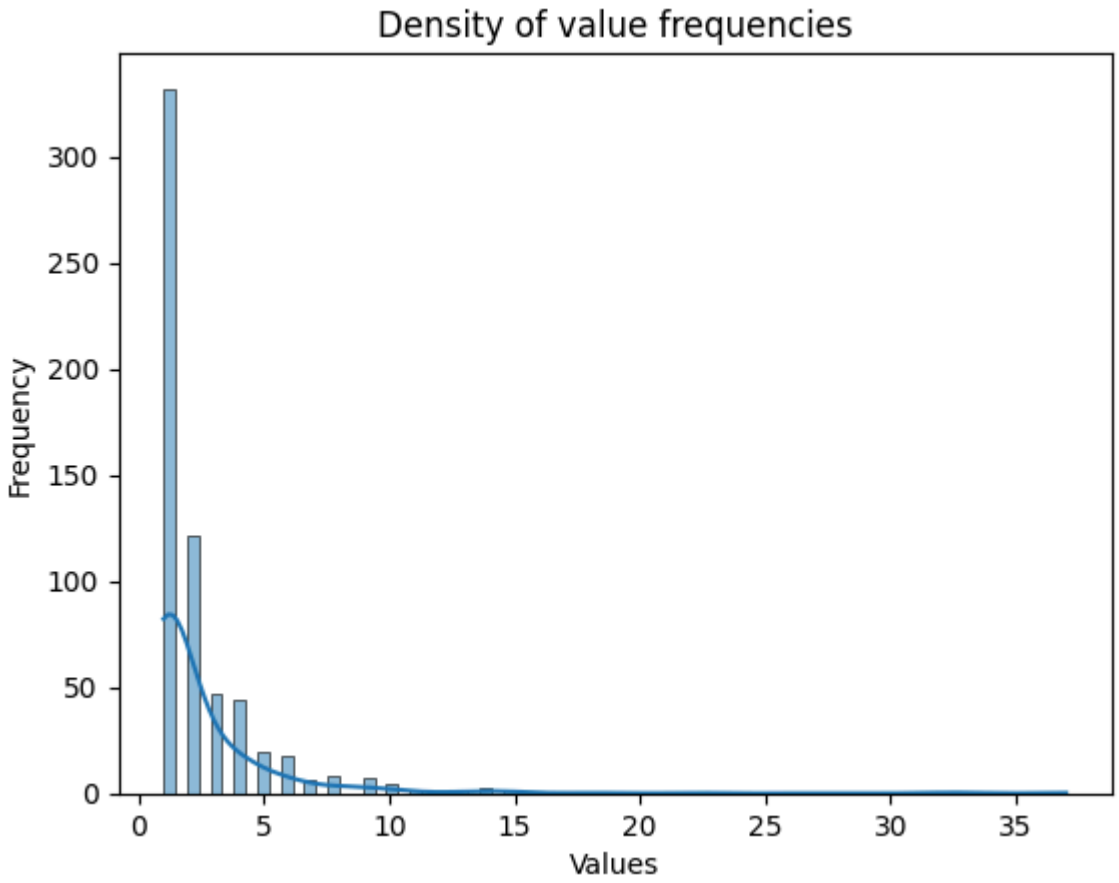




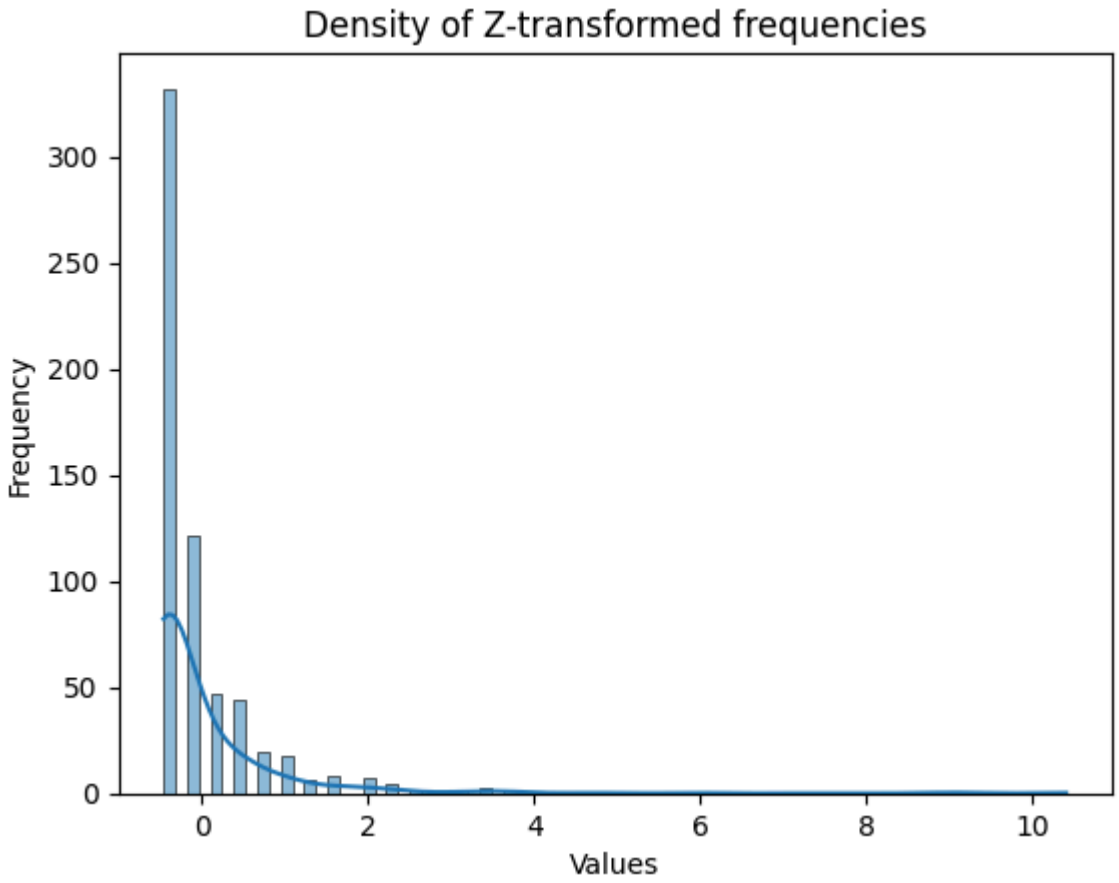
Out[1788]:
```
listingid
4777     17
6242      4
10882     2
12013     3
12334    25
Name: sellercity, dtype: int64
```

In [1789]:

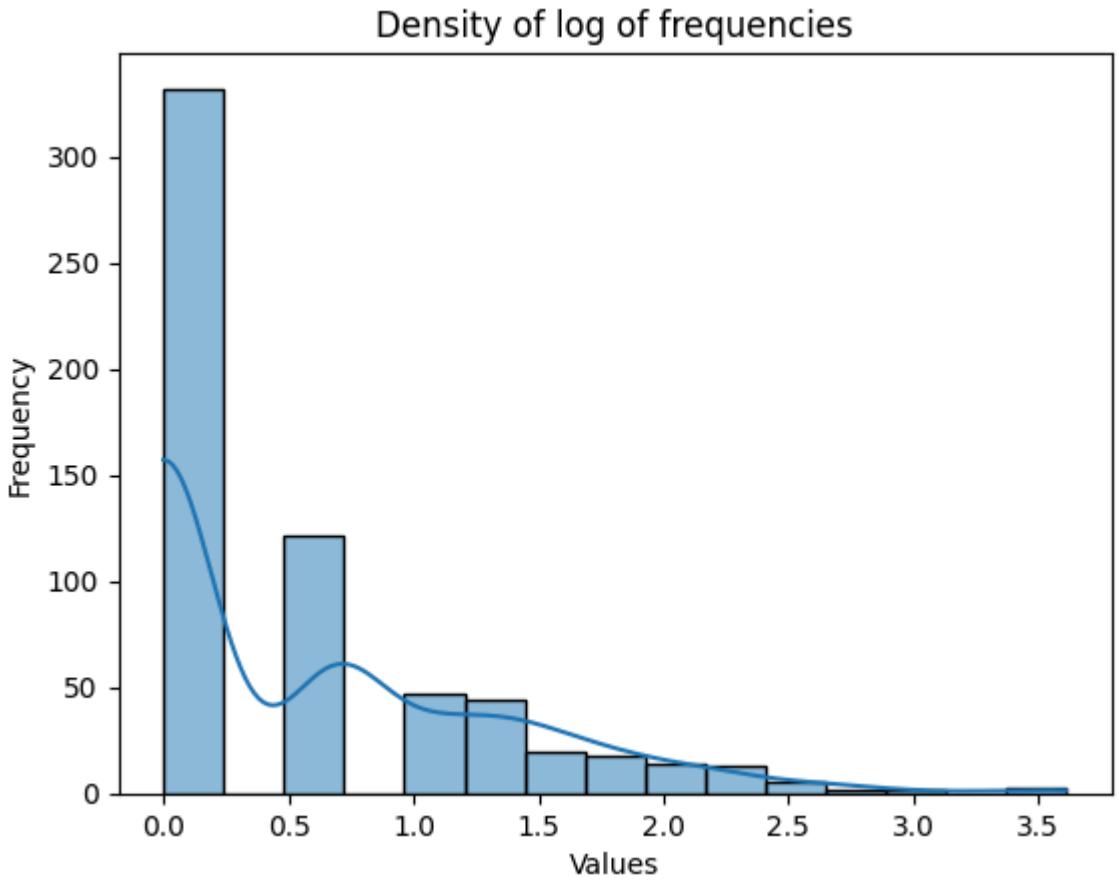
```
value_counts = feat_ptrc.value_counts()
# Plotting a histogram of frequencies (Frequencies of Frequencies)
plotDist(value_counts,"Density of value frequencies")
#FREQUENCY ENCODE THESE VALUES AND THEN TAKE Z SCORE OR THE FREQUENCIES
zvalues = zScoreTransform(value_counts)
print(zvalues)
plotDist(zvalues,"Density of Z-transformed frequencies")
# Assuming 'value_counts' contains the frequencies
log_frequencies = np.log(value_counts)
# Plotting the density plot of the log of frequencies
plt.figure(figsize=(8, 6))
plotDist(log_frequencies,"Density of log of frequencies")
# Plotting the density plot of the log of frequencies
zlog = zScoreTransform(log_frequencies)
plotDist(zlog,"Density of Z-transformed log of frequencies")
freq = feat_ptrc.value_counts().to_dict()
feat_ptrc = feat_ptrc.map(freq)
feat_ptrc.head()
```

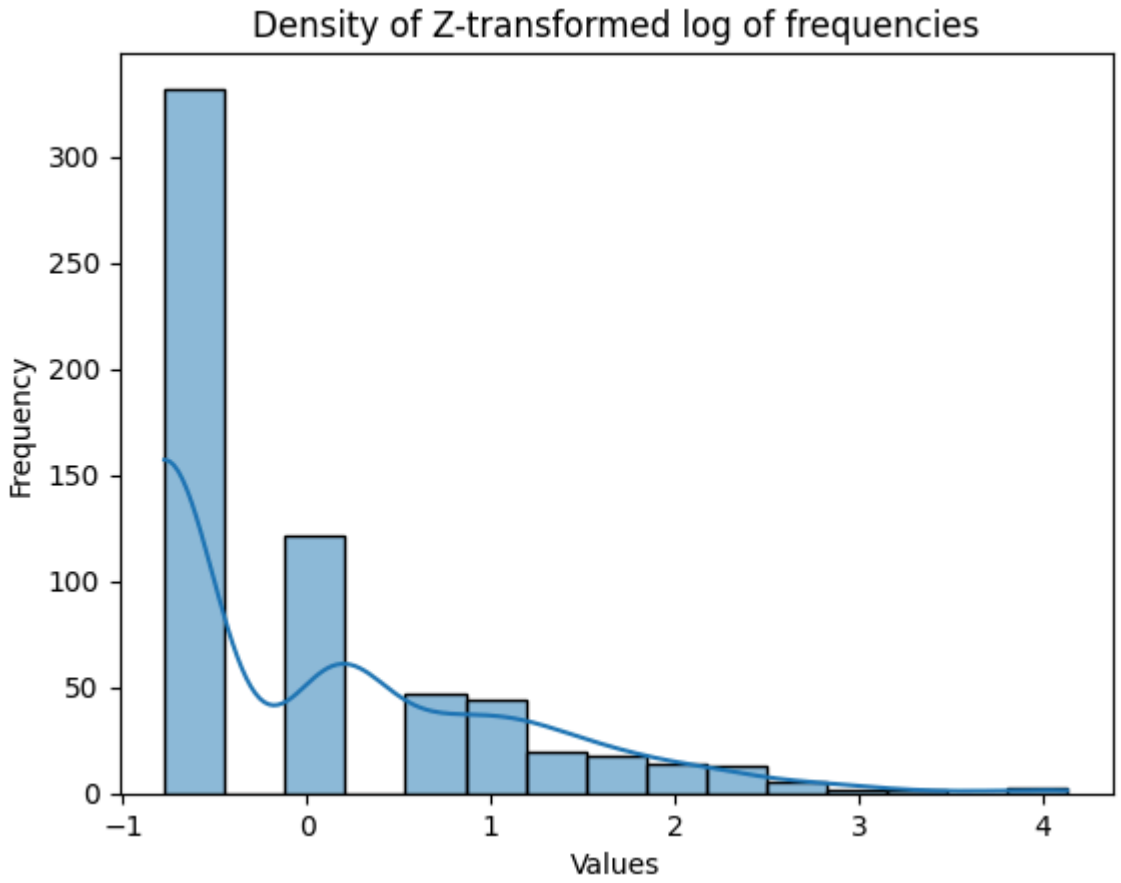



```
sellercity
Dallas         10.406809
Houston         9.199236
Palmyra         8.897343
Columbus        5.878410
Friendswood     4.670838
                 ...
Sylvania       -0.461347
Alto           -0.461347
Riverdale      -0.461347
Cicero         -0.461347
Sandusky       -0.461347
Name: count, Length: 621, dtype: float64
```



<Figure size 800x600 with 0 Axes>





Out[1789]:
```
listingid
7108     2
21448    3
21807    4
30524    3
34061    2
Name: sellercity, dtype: int64
```

In [1790]:
```
input_jeeps[column] = feat_ptrj
input_jeeps.head()
```

Out[1790]:

listingid	sellercity	sellerispriv	sellerlistsrc	sellername	sellerrating	sellerrevcnt	sellerstate	sell
4777	17	False	Jeep Certified Program	Wilde Chrysler Jeep Dodge Ram & Subaru	4.8	1405	WI	531
6242	4	False	Inventory Command Center	Century Dodge Chrysler Jeep RAM	4.4	21	MO	633
10882	2	False	Digital Motorworks (DMi)	Paul Brown Chrysler Dodge Jeep RAM Kia	3.0	51	NY	147
12013	3	False	Digital Motorworks (DMi)	Sierra Motor Mall	3.5	17	IL	613
12334	25	False	Digital Motorworks (DMi)	Larry Roesch Dodge Chrysler Jeep RAM	4.6	240	IL	601

5 rows × 26 columns

In [1791]:
```
input_caddys[column] = feat_ptrc
input_caddys.head()
```

Out[1791]:

listingid	sellercity	sellerispriv	sellerlistsrc	sellername	sellerrating	sellerrevcnt	sellerstate	sell
7108	2	False	HomeNet Automotive	Superior Buick GMC of Fayetteville	3.7	74	AR	727
21448	3	False	HomeNet Automotive	Vroom (Online Dealer - Nationwide Delivery)	3.7	629	LA	701
21807	4	False	HomeNet Automotive	Sunset Cadillac of Bradenton	4.9	360	FL	342
30524	3	False	Digital Motorworks (DMi)	CarMax Ft. Worth-Arlington	5.0	4	TX	761
34061	2	False	Digital Motorworks (DMi)	Ron Craft Chevrolet Cadillac	4.3	312	TX	775

5 rows × 26 columns

In [1792]:
```
freq_cols.append(column)
col+=1
feat_ptrj,column = setFeatPtr(input_jeeps,col)
feat_ptrc,column = setFeatPtr(input_caddys,col)
print(feat_ptrj)
print(feat_ptrc)
```

```
listingid
4777       False
6242       False
10882      False
12013      False
12334      False
           ...
8610847    False
8612731    False
8614177    False
8615510    False
8620012    False
Name: sellerispriv, Length: 3420, dtype: bool
listingid
7108       False
21448      False
21807      False
30524      False
34061      False
           ...
8599564    False
8601212    False
8604205    False
8616294    False
8617378    False
Name: sellerispriv, Length: 1570, dtype: bool
```

In [1793]:
```
print(feat_ptrj.value_counts())
print(feat_ptrc.value_counts())
```

```
sellerispriv
False    3420
Name: count, dtype: int64
sellerispriv
False    1570
Name: count, dtype: int64
```

In [1794]:
```
feats_to_drop.append(column)

col+=1
feat_ptrj,column = setFeatPtr(input_jeeps,col)
print(feat_ptrj.nunique())
print(feat_ptrj.unique())

feat_ptrc,column = setFeatPtr(input_caddys,col)
print(feat_ptrc.nunique())
print(feat_ptrc.unique())
```

```
5
['Jeep Certified Program' 'Inventory Command Center'
 'Digital Motorworks (DMi)' 'HomeNet Automotive' 'My Dealer Center']
4
['HomeNet Automotive' 'Digital Motorworks (DMi)'
 'Inventory Command Center' 'My Dealer Center']
```

In [1795]:

```
#A CATEGORY COLUMN EASY TO ONE HOT ENCODE WITH A SMALL ENUMERATION AMOUNT (ONL
Y REQUIRES
# 5 COLUMNS TO ENCODE)
encoded_cols.append(column)
print(feat_ptrj)
print(feat_ptrc)

col+=1
feat_ptrj,column = setFeatPtr(input_jeeps,col)
print(feat_ptrj)
feat_ptrc,column = setFeatPtr(input_caddys,col)
print(feat_ptrc)
```

```
listingid
4777         Jeep Certified Program
6242       Inventory Command Center
10882      Digital Motorworks (DMi)
12013      Digital Motorworks (DMi)
12334      Digital Motorworks (DMi)
                   ...
8610847          HomeNet Automotive
8612731      Jeep Certified Program
8614177    Digital Motorworks (DMi)
8615510    Digital Motorworks (DMi)
8620012          HomeNet Automotive
Name: sellerlistsrc, Length: 3420, dtype: object
listingid
7108             HomeNet Automotive
21448            HomeNet Automotive
21807            HomeNet Automotive
30524      Digital Motorworks (DMi)
34061      Digital Motorworks (DMi)
                   ...
8599564    Digital Motorworks (DMi)
8601212    Digital Motorworks (DMi)
8604205          HomeNet Automotive
8616294    Digital Motorworks (DMi)
8617378    Digital Motorworks (DMi)
Name: sellerlistsrc, Length: 1570, dtype: object
listingid
4777        Wilde Chrysler Jeep Dodge Ram &amp;amp; Subaru
6242                  Century Dodge Chrysler Jeep RAM
10882           Paul Brown Chrysler Dodge Jeep RAM Kia
12013                              Sierra Motor Mall
12334             Larry Roesch Dodge Chrysler Jeep RAM
                             ...
8610847           Woody&amp;#x27;s Dodge Jeep Chrysler RAM
8612731             Bud&amp;#x27;s Chrysler Dodge Jeep RAM
8614177                          CarMax Columbus Sawmill
8615510                              CarMax Indianapolis
8620012    Vroom (Online Dealer - Nationwide Delivery)
Name: sellername, Length: 3420, dtype: object
listingid
7108                      Superior Buick GMC of Fayetteville
21448            Vroom (Online Dealer - Nationwide Delivery)
21807                             Sunset Cadillac of Bradenton
30524                               CarMax Ft. Worth-Arlington
34061                             Ron Craft Chevrolet Cadillac
                                   ...
8599564                               Cadillac of New Orleans
8601212    Easterns Automotive Group of Sterling / Direct...
8604205                                    Paul Conte Cadillac
8616294                                     Cadillac of Dublin
8617378                                Foster Chevrolet Cadillac
Name: sellername, Length: 1570, dtype: object
```

In [1796]:
```
print(feat_ptrj.value_counts()[feat_ptrj.mode()]/len(feat_ptrj))
feat_ptrj.value_counts().head(30)
```

```
sellername
Carvana    0.069006
Name: count, dtype: float64
```

Out[1796]:
```
sellername
Carvana                                                    236
Vroom (Online Dealer - Nationwide Delivery)                183
Henkel Chrysler Dodge Jeep Ram                              98
Marino Chrysler Jeep Dodge RAM                              34
Barnett Chrysler Jeep Kia                                   33
Blue Knob Auto Sales                                        27
CarMax White Marsh                                          24
Larry Roesch Dodge Chrysler Jeep RAM                        23
Cross Chrysler Jeep Fiat                                    23
Park Chrysler Jeep                                          21
Sherman Dodge Chrysler Jeep RAM                             19
Westgate Chrysler Jeep Dodge RAM                            19
Woody&#x27;s Dodge Jeep Chrysler RAM                        18
Don White&#x27;s Timonium Chrysler Jeep Dodge and RAM       17
Hawk Chrysler Jeep Dodge RAM                                17
Mancari&#x27;s Chrysler Jeep Dodge Ram of Oak Lawn          17
Brad Deery Motors                                           16
Grogan&#x27;s Towne Chrysler Jeep Dodge RAM                 16
Dan Deery Superstore                                        16
Wholesale Inc.                                              15
Tom Ahl Family of Dealerships                               13
Heller Motors                                               13
Uftring Chrysler Dodge Jeep RAM                             13
OffLeaseOnly.com The Nation&#x27;s Used Car Destination     13
Basil Resale Center                                         12
Oxmoor Chrysler Dodge Jeep RAM                              11
Doan Dodge Chrysler Jeep RAM Fiat                           11
Chrysler World                                              11
Hyman Bros Automobiles                                      11
Zeigler Chrysler Jeep Dodge RAM of Schaumburg               11
Name: count, dtype: int64
```

In [1797]:
```
#NOT CATEGORICAL OR CONTAINS DOMINATE VALUES, WILL NOT SIGNFICANTLY IMPACT MOD
EL PREDICTION EFFICIENCY
feats_to_drop.append(column)
```

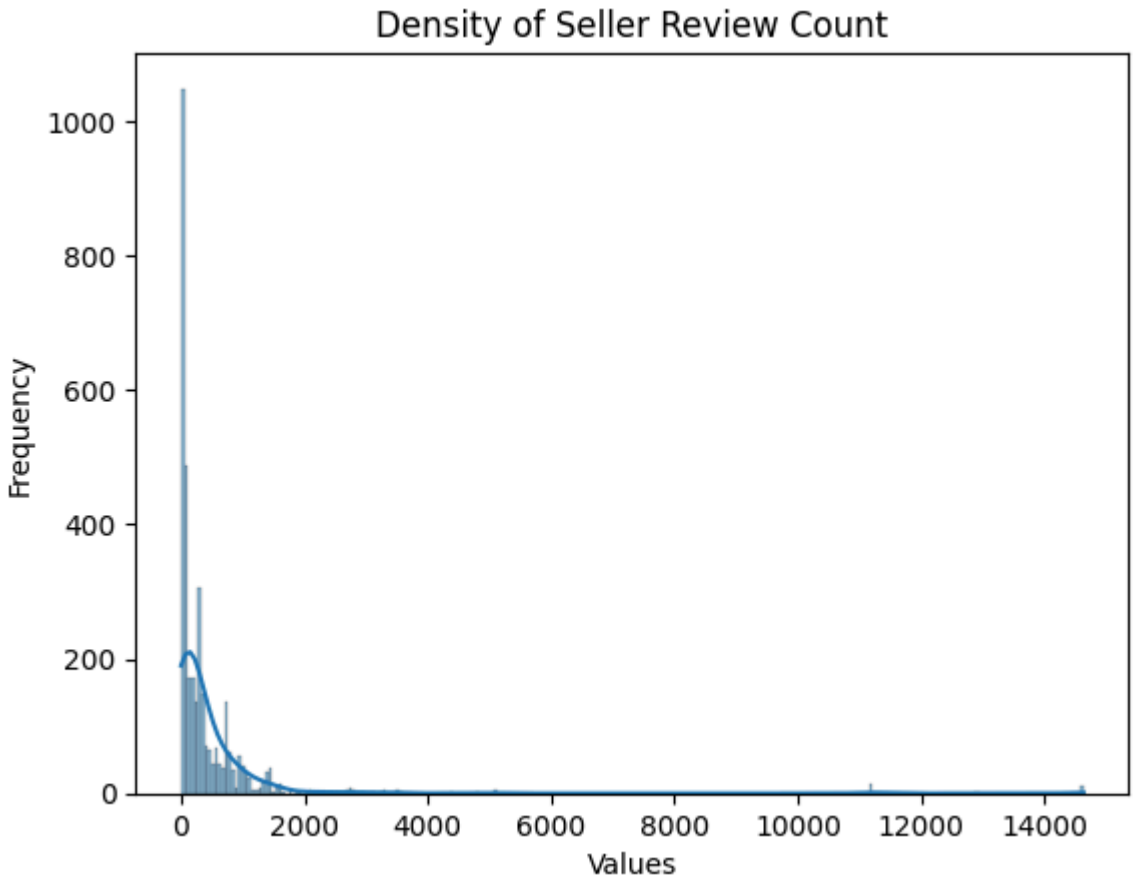
In [1798]:
```
col +=1
feat_ptrj,column = setFeatPtr(input_jeeps,col)
print(feat_ptrj)
feat_ptrc,column = setFeatPtr(input_caddys,col)
print(feat_ptrc)
```

```
listingid
4777       4.8
6242       4.4
10882      3.0
12013      3.5
12334      4.6
           ...
8610847    4.8
8612731    4.9
8614177    1.5
8615510    3.3
8620012    3.8
Name: sellerrating, Length: 3420, dtype: float64
listingid
7108       3.7
21448      3.7
21807      4.9
30524      5.0
34061      4.3
           ...
8599564    4.4
8601212    4.8
8604205    4.7
8616294    4.1
8617378    4.9
Name: sellerrating, Length: 1570, dtype: float64
```

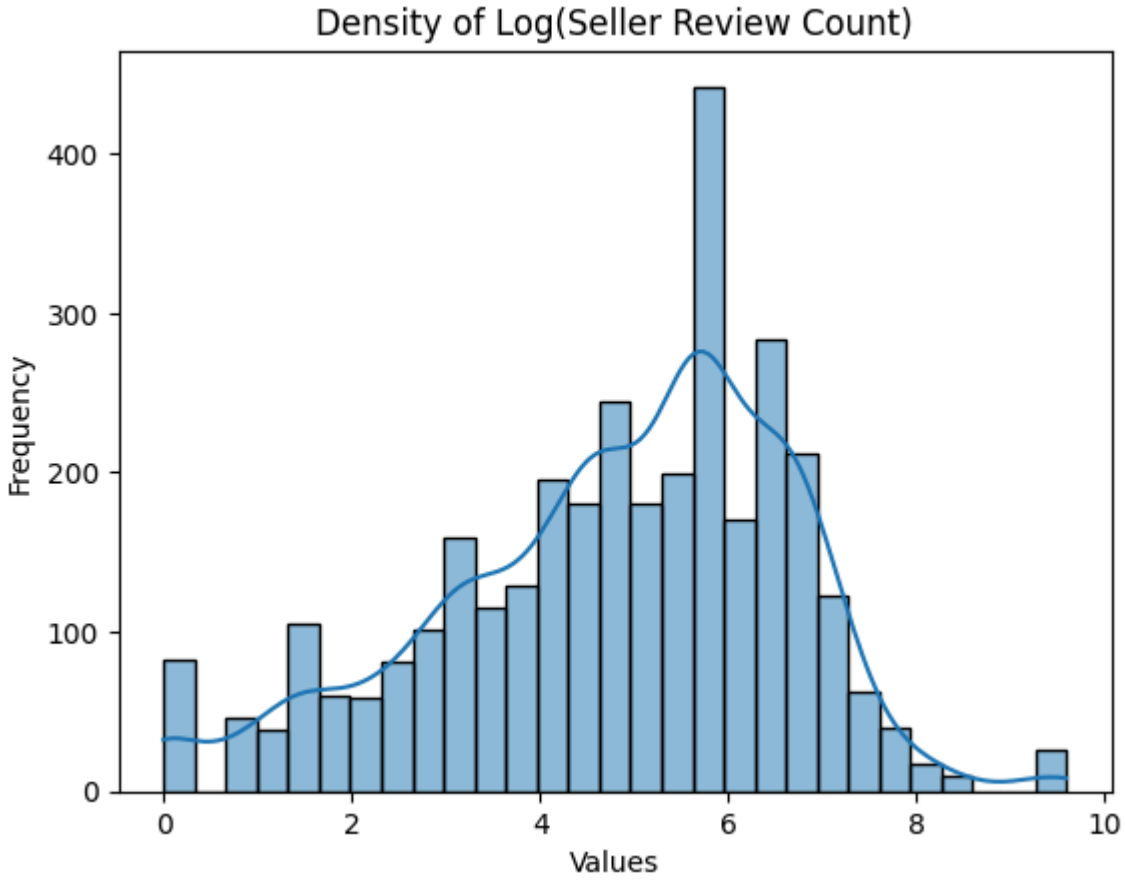
In [1799]:
```
#POSSIBLY NORMALIZE (Z-TRANSFORM) FOR NOW KEEP IT INTACT
same_cols.append(column)
col+=1
feat_ptrj,column = setFeatPtr(input_jeeps,col)
print(feat_ptrj)
feat_ptrc,column = setFeatPtr(input_caddys,col)
print(feat_ptrc)
```

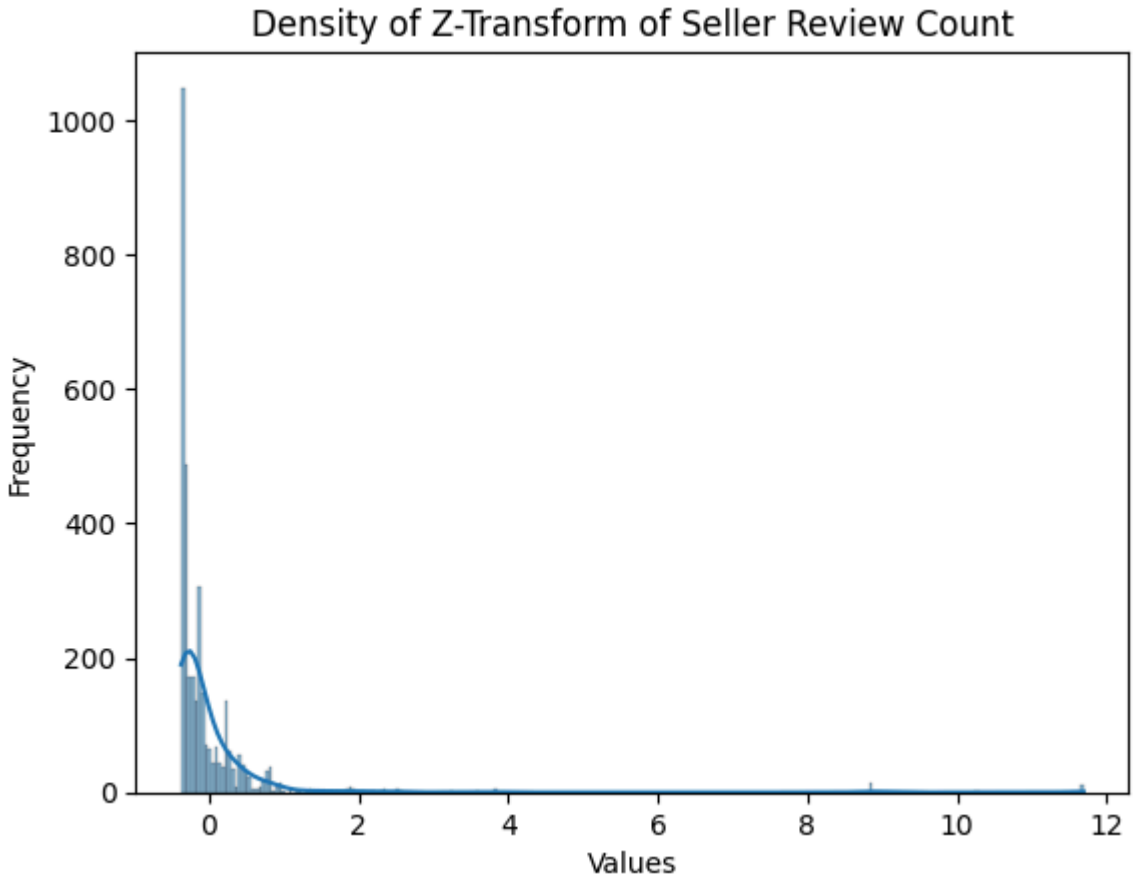
```
listingid
4777       1405
6242         21
10882        51
12013        17
12334       240
           ...
8610847    1016
8612731     121
8614177       6
8615510      16
8620012     727
Name: sellerrevcnt, Length: 3420, dtype: int64
listingid
7108         74
21448       629
21807       360
30524         4
34061       312
           ...
8599564      15
8601212     461
8604205      58
8616294      20
8617378     278
Name: sellerrevcnt, Length: 1570, dtype: int64
```

In [1800]:
```
plotDist(feat_ptrj,'Density of Seller Review Count')
plotDist(np.log(feat_ptrj),'Density of Log(Seller Review Count)')
plotDist(zScoreTransform(feat_ptrj),'Density of Z-Transform of Seller Review C
ount')
plotDist(zScoreTransform(np.log(feat_ptrj)),'Density of Z-Transform of Log of
Seller Review Count')
```

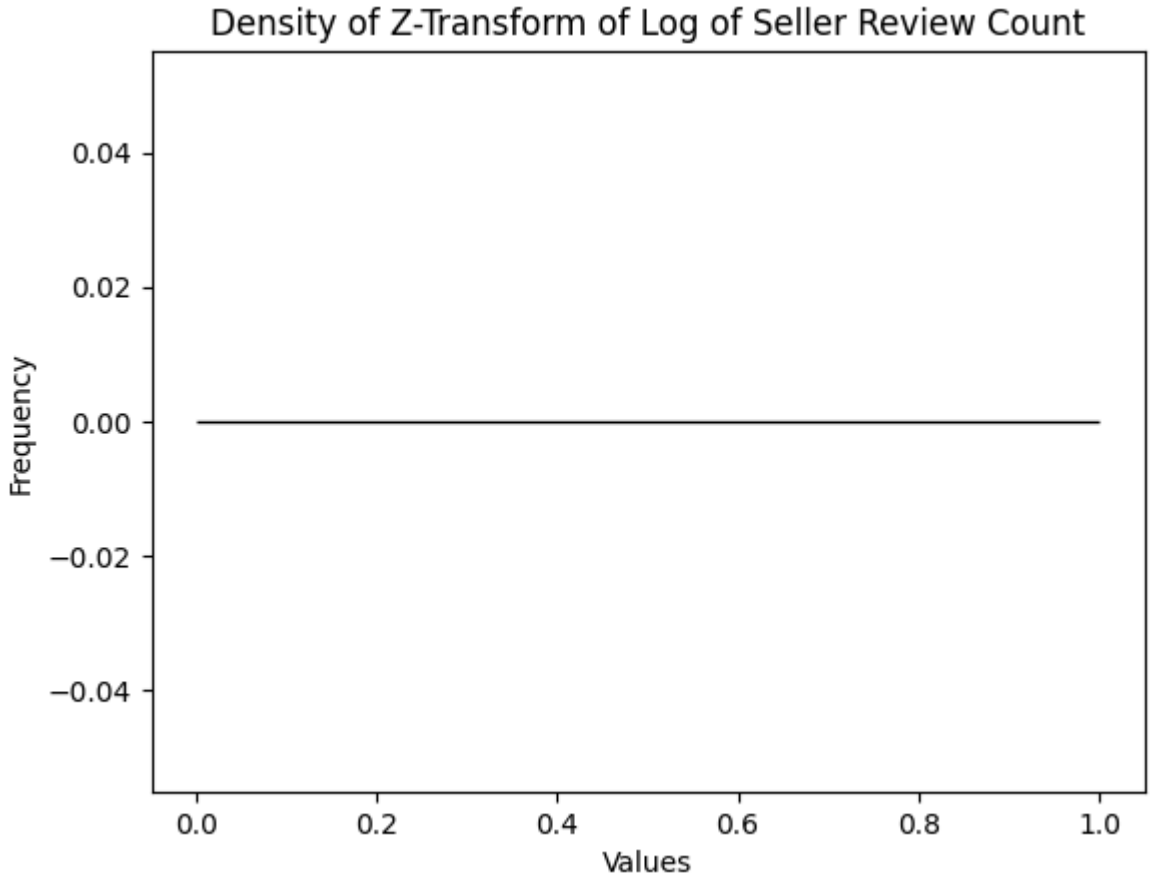



```
C:\Users\aflyn\AppData\Local\Programs\Python\Python312\Lib\site-packages\pand
as\core\arraylike.py:396: RuntimeWarning: divide by zero encountered in log
  result = getattr(ufunc, method)(*inputs, **kwargs)
```


Density of Log(Seller Review Count)
Frequency
Values
0
100
200
300
400
0
2
4
6
8
10


Density of Z-Transform of Seller Review Count
Frequency
Values
0
200
400
600
800
1000
0
2
4
6
8
10
12

```
C:\Users\aflyn\AppData\Local\Programs\Python\Python312\Lib\site-packages\pand
as\core\arraylike.py:396: RuntimeWarning: divide by zero encountered in log
  result = getattr(ufunc, method)(*inputs, **kwargs)
C:\Users\aflyn\AppData\Local\Programs\Python\Python312\Lib\site-packages\pand
as\core\nanops.py:1010: RuntimeWarning: invalid value encountered in subtract
  sqr = _ensure_numeric((avg - values) ** 2)
```



In [1801]:
```
#KEEP REVIEW COUNT AS IS FOR NOW
same_cols.append(column)

col+=1
feat_ptrj,column = setFeatPtr(input_jeeps,col)
feat_ptrj
feat_ptrc,column = setFeatPtr(input_caddys,col)
feat_ptrc
```

Out[1801]:
```
listingid
7108       AR
21448      LA
21807      FL
30524      TX
34061      TX
           ..
8599564    LA
8601212    VA
8604205    NY
8616294    OH
8617378    OH
Name: sellerstate, Length: 1570, dtype: object
```

In [1802]:
```
#STATES -> CATEGORICAL
encoded_cols.append(column)

col+=1
feat_ptrj,column = setFeatPtr(input_jeeps,col)
print(feat_ptrj.nunique())
feat_ptrc,column = setFeatPtr(input_caddys,col)
print(feat_ptrc.nunique())
```

```
970
775
```

In [1803]:
```
#ZIP SEEMS REDUNDANT WITH CITY/STATE INFO ALREADY EXISTING
#PLUS THE AMOUNT OF VARYING ZIPS PROVIDES NOISY DATA
feats_to_drop.append(column)
print(feats_to_drop)

col+=1
feat_ptrj,column = setFeatPtr(input_jeeps,col)
print(feat_ptrj)
feat_ptrc,column = setFeatPtr(input_caddys,col)
print(feat_ptrc)
```

```
['sellerispriv', 'sellername', 'sellerzip']
listingid
4777       SUV
6242       SUV
10882      SUV
12013      SUV
12334      SUV
           ...
8610847    SUV
8612731    SUV
8614177    SUV
8615510    SUV
8620012    SUV
Name: vehbodystyle, Length: 3420, dtype: object
listingid
7108       SUV
21448      SUV
21807      SUV
30524      SUV
34061      SUV
           ...
8599564    SUV
8601212    SUV
8604205    SUV
8616294    SUV
8617378    SUV
Name: vehbodystyle, Length: 1570, dtype: object
```

In [1804]:
```
print(feat_ptrj.nunique())
print(feat_ptrc.nunique())
```

```
1
1
```

In [1805]:
```
#ALL SUV, MEANINGLESS DATA
feats_to_drop.append(column)
col+=1
feat_ptrj,column = setFeatPtr(input_jeeps,col)
print(feat_ptrj)
feat_ptrc,column = setFeatPtr(input_caddys,col)
print(feat_ptrc)
```

```
listingid
4777        True
6242       False
10882      False
12013      False
12334      False
           ...
8610847    False
8612731     True
8614177    False
8615510    False
8620012    False
Name: vehcertified, Length: 3420, dtype: bool
listingid
7108       False
21448      False
21807       True
30524      False
34061      False
           ...
8599564     True
8601212    False
8604205     True
8616294     True
8617378    False
Name: vehcertified, Length: 1570, dtype: bool
```

In [1806]:
```
#MASK BOOLEANS AS 1 AND 0's
feat_ptrj = (feat_ptrj).astype(int)
input_jeeps[column] = feat_ptrj
feat_ptrc = (feat_ptrc).astype(int)
input_caddys[column] = feat_ptrc
mask_cols.append(column)

col+=1
feat_ptrj,column = setFeatPtr(input_jeeps,col)
print(feat_ptrj)
feat_ptrc,column = setFeatPtr(input_caddys,col)
print(feat_ptrc)
```

```
listingid
4777       Brilliant Black Crystal Pearlcoat
6242         Diamond Black Crystal Pearlcoat
10882       Billet Silver Metallic Clearcoat
12013                    True Blue Pearlcoat
12334                                    Red
                          ...
8610847      Walnut Brown Metallic Clearcoat
8612731    Brilliant Black Crystal Pearlcoat
8614177                                Black
8615510                                 Gray
8620012      Diamond Black Crystal Pearlcoat
Name: vehcolorext, Length: 3420, dtype: object
listingid
7108       Radiant Silver Metallic
21448        Dark Granite Metallic
21807        Crystal White Tricoat
30524                        White
34061        Dark Granite Metallic
                   ...
8599564     Stellar Black Metallic
8601212                      White
8604205     Stellar Black Metallic
8616294                      Black
8617378                      Black
Name: vehcolorext, Length: 1570, dtype: object
```

In [1807]:
```
print(feat_ptrj.value_counts())
print(feat_ptrc.value_counts())
```

```
vehcolorext
Bright White Clearcoat                 529
Brilliant Black Crystal Pearlcoat      336
Billet Silver Metallic Clearcoat       336
Granite Crystal Metallic Clearcoat     215
Diamond Black                          189
                                       ...
True Blue Pearl                          1
Diamond Black Crystal                    1
Certified Lthr Pano Roof Nav Camera      1
Db Black                                 1
Bright Sil                               1
Name: count, Length: 106, dtype: int64
vehcolorext
Stellar Black Metallic         289
Radiant Silver Metallic        245
Dark Granite Metallic          193
Crystal White Tricoat          117
Crystal White Tri-Coat          96
Red Passion Tintcoat            88
Black                           61
Black Metallic                  59
Silver                          56
Crystal White                   50
Red                             36
Silver Coast Metallic           34
Dark Adriatic Blue Metallic     33
Bronze Dune Metallic            32
White                           30
Harbor Blue Metallic            29
Gray                            21
Deep Amethyst Metallic          16
Blue                            10
Blue Metallic                    8
Radiant Silver                   8
Stellar Black                    6
Midnight Sky                     6
Silver Moonlight Metallic        5
Dark Granite                     4
Brown                            3
Gold                             3
Bronze                           3
Dark Mocha Metallic              3
Purple                           3
Maroon                           3
Pink                             2
Granite                          2
Dark Adriatic Blue               2
Red Horizon Tintcoat             2
Dark Blue                        1
Red Passion Tc                   1
Midnight Sky Metallic            1
Silver Black                     1
Shadow Metallic                  1
Silver Awd Pano Roof Loaded      1
Silver Coast Me                  1
Pearl White                      1
```

```
Silver Coast                    1
Silver Metallic                 1
Beige                           1
Stellar Black M                 1
Name: count, dtype: int64
```

In [1808]:

```
temp_dfj = handle_vehcolorext(feat_ptrj)
temp_dfj.columns = temp_dfj.columns.str.lower()
temp_dfc = handle_vehcolorext(feat_ptrc)
temp_dfc.columns = temp_dfc.columns.str.lower()

#Encoded with hand-written function rather than the encoder
self_encodej = pd.DataFrame(temp_dfj, index=temp_dfj.index,columns=temp_dfj.co
lumns)
self_encodec = pd.DataFrame(temp_dfc, index=temp_dfc.index,columns=temp_dfc.co
lumns)

orig_cols.append(column)
#Want to drop original
feats_to_drop.append(column)

print(temp_dfj.sum())
print(temp_dfj[temp_dfj["none"]==1].index)

print(temp_dfc.sum())
print(temp_dfc[temp_dfc["none"]==1].index)
```

```
COLOR
COLOR
black         827
blue          160
brown          52
green          14
metallic      868
pearlcoat     818
clearcoat    1338
granite       439
red           407
silver        679
white         719
none           93
dtype: int64
Index([ 287172,  528899,  792826,  822663,  827114,  860178,  989049,  99340
3,
       1006148, 1134881, 1311053, 1433145, 1468297, 1594834, 1782190, 186343
8,
       1886831, 1920561, 1976588, 2639728, 2647801, 2834957, 2852632, 296893
7,
       2986112, 3251931, 3259360, 3318075, 3323591, 3512146, 3576115, 376807
7,
       3825286, 3834986, 3866247, 3891373, 3967275, 4039043, 4518085, 454537
7,
       4651188, 4718768, 5021539, 5055233, 5111906, 5375784, 5535172, 554554
2,
       5578181, 5587030, 5631096, 5644968, 5708516, 5992176, 6162356, 624669
4,
       6273905, 6515958, 6523263, 6577912, 6585299, 6606369, 6651125, 665629
4,
       6675697, 6725167, 6832160, 6908201, 6958924, 6996435, 7089792, 718717
1,
       7255713, 7274472, 7307601, 7395341, 7423652, 7433381, 7565989, 763512
2,
       7700050, 7715629, 7889715, 7957123, 7977183, 8001660, 8043488, 814782
7,
       8164007, 8182771, 8255035, 8257649, 8400784],
      dtype='int64', name='listingid')
black        417
blue          83
brown          3
green          0
metallic     949
pearlcoat      0
clearcoat      0
granite      199
red          127
silver       374
white        294
none          21
dtype: int64
Index([ 736178,  816014, 1083437, 1741814, 1918647, 2808512, 2910575, 386273
3,
       4142609, 4743987, 5466930, 5662854, 5932615, 6022159, 6562720, 666179
6,
```

```
 6784340, 6815471, 7415386, 7807696, 7999952],
dtype='int64', name='listingid')
```

In [1809]:

```
col+=1
feat_ptrj,column = setFeatPtr(input_jeeps,col)
print(feat_ptrj.value_counts())
feat_ptrc,column = setFeatPtr(input_caddys,col)
print(feat_ptrc.value_counts())
```

```
vehcolorint
Black                          2620
Black Leather                    95
black                            87
Light Frost                      83
Brown                            54
                                ...
Black / Light Frost Beige         1
CHARCOAL                          1
Ruby Red/Black Leather            1
Black/Ruby Red                    1
Black, cloth                      1
Name: count, Length: 65, dtype: int64
vehcolorint
Jet Black                                              747
Shara Beige                                            236
Sahara Beige                                           143
Cirrus                                                 131
Black                                                   89
Maple Sugar                                             67
Tan                                                     38
jet black                                               29
Carbon Plum                                             15
sahara beige                                            12
JET BLACK LUNAR BRUSHED ALUMINIUM TRIM                   8
Beige                                                    6
SAHARA BEIGE/JET BLACK ACCENTS NATURAL SAPELE HIGH       4
Other                                                    4
Sahara Beige Leather                                     3
BLACK                                                    3
GREY,LEATHER                                             2
Cream                                                    2
Jet Black w/Full Leather Seats w/Mini Perforated I       2
SAHARA BEIGE/JET BLACK ACCENTS OKAPI STRIPE DESIGN       2
Jet Black Leather                                        2
Gray                                                     2
Jet Black Lunar Brushed Aluminium Trim                   2
JET BLACK                                                2
Sahara Beige W/ Jet Black Accent                         2
Jet Black w/Leather Seating Surfaces w/Mini Perfor       1
Black Leather                                            1
Jet Black W/Leather Seating Surfaces W/Mini Perfor       1
CIRRUS                                                   1
Bronze                                                   1
Cirrus w/Dark Titanium Accents w/Full Leather Seat       1
CIRRUS W/ DARK TITANIUM ACCENTS DIAMOND CUT ALUMIN       1
JET BLACK BRONZE CARBON FIBER TRIM                       1
Dark Granite                                             1
Jet Black, premium leather                               1
Sahara Beige w/ Jet Black Accent                         1
Cirrus w/dark atmosphere                                 1
SUGAR MAPLE LEATHER                                      1
TAN                                                      1
Cirrus w/Dark Titanium Accents w/Leather Seating S       1
Carbon                                                   1
```

```
JET BLACK, LEATHER SEATING SURFACES WITH MINI-PERF      1
Name: count, dtype: int64
```

In [1810]:

```
temp_dfj = handle_vehcolorint(feat_ptrj)
temp_dfj.columns = temp_dfj.columns.str.lower()
temp_dfc = handle_vehcolorint(feat_ptrc)
temp_dfc.columns = temp_dfc.columns.str.lower()

#Merge two handwritten encoded columns
self_encodej = pd.merge(self_encodej, temp_dfj, left_index=True,right_index=Tr
ue)
self_encodec = pd.merge(self_encodec, temp_dfc, left_index=True,right_index=Tr
ue)

orig_cols.append(column)
#Want to drop original
feats_to_drop.append(column)

print(temp_dfj.sum())
print(temp_dfj[temp_dfj["none"]==1].index)

print(temp_dfc.sum())
print(temp_dfc[temp_dfc["none"]==1].index)
```

```
COLOR2
COLOR2
black     3050
blue        18
brown      168
beige      148
trim         0
red        110
silver      39
frost      282
maple        0
tan         16
cirrus       0
carbon       0
plum         0
none        13
dtype: int64
Index([ 903203, 1491106, 2000241, 2028388, 2102637, 2585204, 3198504, 340954
9,
       4761924, 5102938, 5252166, 6400857, 6475088],
      dtype='int64', name='listingid')
black     899
blue        0
brown       0
beige     409
trim       11
red         0
silver      2
frost       0
maple      68
tan        42
cirrus    136
carbon     17
plum       15
none       10
dtype: int64
Index([2729520, 2839309, 2842810, 4833074, 4904298, 5013738, 5538849, 602215
9,
       7504666, 8084993],
      dtype='int64', name='listingid')
```

In [1811]:
```
col+=1
feat_ptrj,column = setFeatPtr(input_jeeps,col)
print(feat_ptrj)
feat_ptrc,column = setFeatPtr(input_caddys,col)
print(feat_ptrc)
```

```
listingid
4777       4x4/4WD
6242           4WD
10882          4WD
12013          4WD
12334          4WD
             ...
8610847        4WD
8612731        4WD
8614177        4WD
8615510        4WD
8620012        4WD
Name: vehdrivetrain, Length: 3420, dtype: object
listingid
7108       FWD
21448      FWD
21807      FWD
30524      FWD
34061      FWD
           ...
8599564    FWD
8601212    FWD
8604205    AWD
8616294    FWD
8617378    FWD
Name: vehdrivetrain, Length: 1570, dtype: object
```

In [1812]:
```
print(feat_ptrj.value_counts())
print(feat_ptrc.value_counts())
```

```
vehdrivetrain
4WD                 3335
4X4                   32
Four Wheel Drive      29
4x4/4WD                9
4x4                    6
AWD or 4x4             5
AWD                    3
4WD/AWD                1
Name: count, dtype: int64
vehdrivetrain
FWD                                                          903
AWD                                                          627
All Wheel Drive                                               18
Front Wheel Drive                                             16
All-wheel Drive                                                3
ALL-WHEEL DRIVE WITH LOCKING AND LIMITED-SLIP DIFFERENTIAL     1
ALL WHEEL                                                      1
AllWheelDrive                                                  1
Name: count, dtype: int64
```

In [1813]:
```
#BASED OFF UNIQUE VALUES SEPERATE INTO 4WD,FWD,or AWD
temp_dfj = handle_vehdrivetrain(feat_ptrj)
print(temp_dfj.value_counts())
temp_dfc = handle_vehdrivetrain(feat_ptrc)
print(temp_dfc.value_counts())
```

```
vehdrivetrain
_4_wd     3411
hybrid       6
awd          3
Name: count, dtype: int64
vehdrivetrain
fwd    919
awd    651
Name: count, dtype: int64
```

In [1814]:
```
input_jeeps[column] = temp_dfj
input_caddys[column] = temp_dfc
encoded_cols.append(column)
orig_cols.append(column)
col+=1
print(encoded_cols)
print(input_jeeps[column])
print(input_caddys[column])
```

```
['sellerlistsrc', 'sellerstate', 'vehdrivetrain']
listingid
4777       _4_wd
6242       _4_wd
10882      _4_wd
12013      _4_wd
12334      _4_wd
           ...
8610847    _4_wd
8612731    _4_wd
8614177    _4_wd
8615510    _4_wd
8620012    _4_wd
Name: vehdrivetrain, Length: 3420, dtype: object
listingid
7108       fwd
21448      fwd
21807      fwd
30524      fwd
34061      fwd
          ...
8599564    fwd
8601212    fwd
8604205    awd
8616294    fwd
8617378    fwd
Name: vehdrivetrain, Length: 1570, dtype: object
```

In [1815]:
```
feat_ptrj,column = setFeatPtr(input_jeeps,col)
feat_ptrc,column = setFeatPtr(input_caddys,col)
print(feat_ptrj.value_counts())
print(feat_ptrc.value_counts())
```

```
vehengine
3.6L V6 24V MPFI DOHC                    1566
Regular Unleaded V-6 3.6 L/220            552
3.6L V6 24V MPFI DOHC Flexible Fuel       301
3.6L V6 24V VVT                           254
3.6L V6                                   169
                                         ...
6.4L HEMI V8                                1
3.0L V6                                     1
5.7L V8 OHV 16V                             1
V6 3.6L Natural Aspiration                  1
6.2L 8 Cyl.                                 1
Name: count, Length: 66, dtype: int64
vehengine
3.6L V6 24V GDI DOHC                                    920
Gas V6 3.6L/222.6                                       317
3.6L V6 DI VVT                                          133
6 Cylinder                                               51
3.6L                                                     28
V6 Cylinder Engine 3.6L                                  25
3.6L V6 CYLINDER                                         18
3.6L V6                                                  18
Gas V6 3.6L/222                                          11
3.6L V6 DI VVT Engine                                     9
V6 Cylinder Engine                                        8
3.6L V6 Cylinder Engine                                   7
3.6L 6 cyl Fuel Injected                                  3
3.6L 6 cyl                                                3
V-6 cyl                                                   3
3.6L V6 DOHC 24V                                          3
3.6L V6 Cylinder                                          2
V6 3.6 Liter                                              2
Gas V6                                                    2
3.6L V6 310HP                                             1
3.6L 6 Cylinders                                          1
Gas V6 3.6L/222.6 CU.IN.                                  1
3.6L 6 CYL. GAS                                           1
3.6L V6 DI VVT with Automatic Stop/Start (310 hp          1
3.6L V6, DI, VVT, WITH AUTOMATIC STOP/START               1
V6, 3.6 Liter                                             1
Name: count, dtype: int64
```

In [1816]:
```
#handle_vehengine takes the vehEngine column and turns it into a
#2 column data frame by splitting the phrases into engine size
#and cyclinder configuration
temp_dfj = handle_vehengine(feat_ptrj)
temp_dfc = handle_vehengine(feat_ptrc)

print(temp_dfj["EngineSize"].value_counts())
print(temp_dfj["Cylinders"].value_counts())
print(temp_dfc["EngineSize"].value_counts())
print(temp_dfc["Cylinders"].value_counts())

# '0' represents unknown for either columns
```

```
EngineSize
3.6    3021
5.7     198
0.0      78
6.4      63
3.0      37
6.2      23
Name: count, dtype: int64
Cylinders
6    3063
8     279
0      78
Name: count, dtype: int64
EngineSize
3.6    1503
0.0      67
Name: count, dtype: int64
Cylinders
6    1535
0      35
Name: count, dtype: int64
```

In [1817]:
```
input_jeeps[temp_dfj.columns] = temp_dfj
input_caddys[temp_dfc.columns] = temp_dfc

orig_cols.append(column)
feats_to_drop.append(column)

col+=1
print(encoded_cols)
print(temp_dfj)
print(temp_dfc)
```

```
['sellerlistsrc', 'sellerstate', 'vehdrivetrain']
           EngineSize  Cylinders
listingid
4777              3.6          6
6242              3.6          6
10882             3.6          6
12013             3.6          6
12334             3.6          6
...               ...        ...
8610847           3.6          6
8612731           3.6          6
8614177           3.6          6
8615510           3.0          6
8620012           3.6          6

[3420 rows x 2 columns]
           EngineSize  Cylinders
listingid
7108              3.6          6
21448             3.6          6
21807             3.6          6
30524             3.6          6
34061             3.6          6
...               ...        ...
8599564           3.6          6
8601212           3.6          6
8604205           3.6          6
8616294           3.6          6
8617378           3.6          6

[1570 rows x 2 columns]
```

In [1818]:
```
feat_ptrj,column = setFeatPtr(input_jeeps,col)
print(feat_ptrj)
feat_ptrc,column = setFeatPtr(input_caddys,col)
print(feat_ptrc)
```

```
listingid
4777       ['18 WHEEL &amp;amp; 8.4 RADIO GROUP-inc: Nav-Capa...
6242       ['Android Auto', 'Antilock Brakes', 'Apple Car...
10882      ['1st and 2nd row curtain head airbags', '4-wh...
12013      ['1st and 2nd row curtain head airbags', '4-wh...
12334      ['1st and 2nd row curtain head airbags', '4-wh...
                                  ...
8610847    ['3.45 REAR AXLE RATIO(STD)', '4-Wheel Disc Br...
8612731    ['1st and 2nd row curtain head airbags', '4-wh...
8614177    ['1st and 2nd row curtain head airbags', '4-wh...
8615510    ['1st and 2nd row curtain head airbags', '4-wh...
8620012    ['Airbag Occupancy Sensor', 'Curtain 1st And 2...
Name: vehfeats, Length: 3420, dtype: object
listingid
7108       ['4-Wheel Disc Brakes', 'ABS', 'Adjustable Ste...
21448      ['20 Inch Alloy Wheels', '3.6L V6 Engine', 'An...
21807      ['ABS', 'Aluminum Wheels', 'AUDIO SYSTEM FEATU...
30524      ['1st and 2nd row curtain head airbags', '4-wh...
34061      ['1st and 2nd row curtain head airbags', '4-wh...
                                  ...
8599564    ['1st and 2nd row curtain head airbags', '4-wh...
8601212    ['1st and 2nd row curtain head airbags', '4-wh...
8604205    ['4-Wheel Disc Brakes', 'ABS', 'Active Suspens...
8616294    ['1st and 2nd row curtain head airbags', '4-wh...
8617378    ['1st and 2nd row curtain head airbags', '4-wh...
Name: vehfeats, Length: 1570, dtype: object
```

In [1819]:
```
#ELIMINATE WORDS THAT APPEAR IN MORE THAN max_doc_freq OF DOCUMENTS (DOCUMENT
~ ROW)
#WILL GET RID OF COMMON WORDS SUCH AS "THE", "A", etc.
#LIMIT VOCABULARY TO max_feats COLUMNS (ONE FOR EACH WORD)
tf_featsj = TfidfVectorizer(max_df=0.50,max_features=30)
temp_dfj = feat_ptrj.copy()
tf_featsj = tf_featsj.fit(temp_dfj)
vocab1j = tf_idfTokenizer(temp_dfj,tf_featsj)
#THOUGHT: TUNE THE HYPERPARAMETERS TO OPTIMIZE THE TOKENIZER?
vocab1j.head()
```

Out[1819]:

listingid	air	alarm	alloy	aluminum	am	anti	antilock	auto	automatic
4777	0.000000	0.000000	0.0	0.362589	0.260283	0.000000	0.000000	0.359060	0.261243
6242	0.000000	0.000000	0.0	0.000000	0.000000	0.000000	0.397475	0.426209	0.000000
10882	0.000000	0.303521	0.0	0.000000	0.000000	0.292306	0.000000	0.000000	0.000000
12013	0.289613	0.000000	0.0	0.000000	0.256202	0.000000	0.000000	0.000000	0.257148
12334	0.000000	0.303521	0.0	0.000000	0.000000	0.292306	0.000000	0.000000	0.000000

5 rows × 30 columns

In [1820]:
```
#ELIMINATE WORDS THAT APPEAR IN MORE THAN max_doc_freq OF DOCUMENTS (DOCUMENT
~ ROW)
#WILL GET RID OF COMMON WORDS SUCH AS "THE", "A", etc.
#LIMIT VOCABULARY TO max_feats COLUMNS (ONE FOR EACH WORD)
tf_featsc = TfidfVectorizer(max_df=0.50,max_features=30)
temp_dfc = feat_ptrc.copy()
tf_featsc = tf_featsc.fit(temp_dfc)
vocab1c = tf_idfTokenizer(temp_dfc,tf_featsc)
#THOUGHT: TUNE THE HYPERPARAMETERS TO OPTIMIZE THE TOKENIZER?
vocab1c.head()
```

Out[1820]:

listingid	all	aluminum	android	apple	assist	auto	automatic	auxiliary	beverage
7108	0.0	0.273007	0.000000	0.000000	0.000000	0.305368	0.318800	0.000000	0.000000
21448	0.0	0.000000	0.308276	0.315974	0.000000	0.274410	0.572961	0.325261	0.000000
21807	0.0	0.169650	0.213179	0.218502	0.186359	0.189760	0.198107	0.449849	0.000000
30524	0.0	0.000000	0.000000	0.000000	0.332808	0.000000	0.353788	0.000000	0.000000
34061	0.0	0.279240	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.317566

5 rows × 30 columns

In [1821]:
```
#DROP ORIGINAL STATE COLUMN AND LATER REPLACE WITH ENCODED MATRIX COLUMNS
feats_to_drop.append(column)
tokenize_cols = [column]
input_jeeps.head()
```

Out[1821]:

listingid	sellercity	sellerispriv	sellerlistsrc	sellername	sellerrating	sellerrevcnt	sellerstate	sell
4777	17	False	Jeep Certified Program	Wilde Chrysler Jeep Dodge Ram & Subaru	4.8	1405	WI	531
6242	4	False	Inventory Command Center	Century Dodge Chrysler Jeep RAM	4.4	21	MO	633
10882	2	False	Digital Motorworks (DMi)	Paul Brown Chrysler Dodge Jeep RAM Kia	3.0	51	NY	147
12013	3	False	Digital Motorworks (DMi)	Sierra Motor Mall	3.5	17	IL	613
12334	25	False	Digital Motorworks (DMi)	Larry Roesch Dodge Chrysler Jeep RAM	4.6	240	IL	601

5 rows × 28 columns

In [1822]:
```
input_caddys.head()
```

Out[1822]:

listingid	sellercity	sellerispriv	sellerlistsrc	sellername	sellerrating	sellerrevcnt	sellerstate	sell
7108	2	False	HomeNet Automotive	Superior Buick GMC of Fayetteville	3.7	74	AR	727
21448	3	False	HomeNet Automotive	Vroom (Online Dealer - Nationwide Delivery)	3.7	629	LA	701
21807	4	False	HomeNet Automotive	Sunset Cadillac of Bradenton	4.9	360	FL	342
30524	3	False	Digital Motorworks (DMi)	CarMax Ft. Worth-Arlington	5.0	4	TX	761
34061	2	False	Digital Motorworks (DMi)	Ron Craft Chevrolet Cadillac	4.3	312	TX	775

5 rows × 28 columns

In [1823]:
```
col+=1
feat_ptrj,column = setFeatPtr(input_jeeps,col)
print(feat_ptrj.value_counts())
feat_ptrc,column = setFeatPtr(input_caddys,col)
print(feat_ptrc.value_counts())
```

```
vehfuel
Gasoline         3070
E85 Flex Fuel     310
Diesel             35
Unknown             5
Name: count, dtype: int64
vehfuel
Gasoline    1570
Name: count, dtype: int64
```

In [1824]:
```
encoded_cols.append(column)

col+=1
```

In [1825]:
```
feat_ptrj,column = setFeatPtr(input_jeeps,col)
print(feat_ptrj)
feat_ptrc,column = setFeatPtr(input_caddys,col)
print(feat_ptrc)

temp_dfj = feat_ptrj.str.split(',',n=1,expand=True)
temp_dfj.columns = ['Owners', 'History']
temp_dfj["History"].unique()
```

```
listingid
4777                    1 Owner, Buyback Protection Eligible
6242       1 Owner, Non-Personal Use Reported, Buyback Pr...
10882      1 Owner, Non-Personal Use Reported, Buyback Pr...
12013      1 Owner, Accident(s) Reported, Non-Personal Us...
12334      1 Owner, Non-Personal Use Reported, Buyback Pr...
                                ...
8610847                 1 Owner, Buyback Protection Eligible
8612731                 1 Owner, Buyback Protection Eligible
8614177                 1 Owner, Buyback Protection Eligible
8615510    1 Owner, Accident(s) Reported, Non-Personal Us...
8620012    1 Owner, Non-Personal Use Reported, Buyback Pr...
Name: vehhistory, Length: 3420, dtype: object
listingid
7108       1 Owner, Non-Personal Use Reported, Buyback Pr...
21448      1 Owner, Non-Personal Use Reported, Buyback Pr...
21807                   1 Owner, Buyback Protection Eligible
30524      1 Owner, Non-Personal Use Reported, Buyback Pr...
34061      1 Owner, Non-Personal Use Reported, Buyback Pr...
                                ...
8599564    1 Owner, Non-Personal Use Reported, Buyback Pr...
8601212    1 Owner, Non-Personal Use Reported, Buyback Pr...
8604205    1 Owner, Non-Personal Use Reported, Buyback Pr...
8616294                  0 Owners, Buyback Protection Eligible
8617378    2 Owners, Non-Personal Use Reported, Buyback P...
Name: vehhistory, Length: 1570, dtype: object
```

Out[1825]:
```
array([' Buyback Protection Eligible',
       ' Non-Personal Use Reported, Buyback Protection Eligible',
       ' Accident(s) Reported, Non-Personal Use Reported, Buyback Protection
Eligible',
       ' Accident(s) Reported, Buyback Protection Eligible',
       ' Title Issue(s) Reported',
       ' Accident(s) Reported, Non-Personal Use Reported, Title Issue(s) Repo
rted',
       ' Title Issue(s) Reported, Buyback Protection Eligible',
       ' Accident(s) Reported, Non-Personal Use Reported, Title Issue(s) Repo
rted, Buyback Protection Eligible',
       ' Non-Personal Use Reported, Title Issue(s) Reported, Buyback Protecti
on Eligible',
       ' Accident(s) Reported, Title Issue(s) Reported',
       ' Accident(s) Reported, Title Issue(s) Reported, Buyback Protection El
igible',
       None], dtype=object)
```

In [1826]:
```
temp_dfc = feat_ptrc.str.split(',',n=1,expand=True)
temp_dfc.columns = ['Owners', 'History']
temp_dfc["History"].unique()
```

Out[1826]:
```
array([' Non-Personal Use Reported, Buyback Protection Eligible',
       ' Buyback Protection Eligible',
       ' Accident(s) Reported, Non-Personal Use Reported, Title Issue(s) Repo
rted',
       ' Accident(s) Reported, Buyback Protection Eligible',
       ' Accident(s) Reported, Non-Personal Use Reported, Buyback Protection
Eligible',
       ' Accident(s) Reported, Title Issue(s) Reported',
       ' Title Issue(s) Reported',
       ' Non-Personal Use Reported, Title Issue(s) Reported',
       ' Non-Personal Use Reported, Title Issue(s) Reported, Buyback Protecti
on Eligible',
       ' Accident(s) Reported, Non-Personal Use Reported, Title Issue(s) Repo
rted, Buyback Protection Eligible',
       ' Accident(s) Reported, Title Issue(s) Reported, Buyback Protection El
igible'],
      dtype=object)
```

In [1827]:
```
temp_dfj['Owners'] = temp_dfj['Owners'].str.extract(r'^(\d+)')
temp_dfc['Owners'] = temp_dfc['Owners'].str.extract(r'^(\d+)')

temp_dfj['Owners'].head()
```

Out[1827]:
```
listingid
4777     1
6242     1
10882    1
12013    1
12334    1
Name: Owners, dtype: object
```

In [1828]:
```
input_jeeps['Owners'] = temp_dfj['Owners']
input_caddys['Owners'] = temp_dfc['Owners']

print(input_jeeps['Owners'])
print(input_caddys['Owners'])
```

```
listingid
4777       1
6242       1
10882      1
12013      1
12334      1
          ..
8610847    1
8612731    1
8614177    1
8615510    1
8620012    1
Name: Owners, Length: 3420, dtype: object
listingid
7108       1
21448      1
21807      1
30524      1
34061      1
          ..
8599564    1
8601212    1
8604205    1
8616294    0
8617378    2
Name: Owners, Length: 1570, dtype: object
```

In [1829]:
```
temp_dfj["History"].value_counts()
```

Out[1829]:
```
History
 Buyback Protection Eligible
1901
 Non-Personal Use Reported, Buyback Protection Eligible
1116
 Accident(s) Reported, Buyback Protection Eligible
196
 Accident(s) Reported, Non-Personal Use Reported, Buyback Protection Eligible
117
 Accident(s) Reported, Non-Personal Use Reported, Title Issue(s) Reported
43
 Title Issue(s) Reported
20
 Accident(s) Reported, Title Issue(s) Reported, Buyback Protection Eligible
5
 Accident(s) Reported, Non-Personal Use Reported, Title Issue(s) Reported, Bu
yback Protection Eligible       4
 Non-Personal Use Reported, Title Issue(s) Reported, Buyback Protection Eligi
ble                             4
 Accident(s) Reported, Title Issue(s) Reported
4
 Title Issue(s) Reported, Buyback Protection Eligible
3
Name: count, dtype: int64
```

In [1830]:
```
#TURNS OUT THAT THESE PHRASES CAN ACTUALLY BE TURNED INTO CATEGORICAL COLUMNS
#EACH ELEMENT IS A COMBINATION OF VARYING SIZE OF THE 4 POSSIBLE UNIQUE PHRASE
S
#ONE HOT ENCODE WITH A COLUMN FOR EACH PHRASE
encoded_histj = handle_vehhistory(temp_dfj["History"])
encoded_histj.head()
```

```
HISTORY
```

Out[1830]:

listingid	Accident(s) Reported	Buyback Protection Eligible	Non-Personal Use Reported	Title Issue(s) Reported	None of the above
4777	0	1	0	0	0
6242	0	1	1	0	0
10882	0	1	1	0	0
12013	1	1	1	0	0
12334	0	1	1	0	0

In [1831]:
```
encoded_histc = handle_vehhistory(temp_dfc["History"])
encoded_histc.head()
```

```
HISTORY
```

Out[1831]:

listingid	Accident(s) Reported	Buyback Protection Eligible	Non-Personal Use Reported	Title Issue(s) Reported	None of the above
7108	0	1	1	0	0
21448	0	1	1	0	0
21807	0	1	0	0	0
30524	0	1	1	0	0
34061	0	1	1	0	0

In [1832]:
```
#DROP ORIGINAL COLUMN AND LATER REPLACE WITH ENCODED MATRIX COLUMNS
feats_to_drop.append(column)
self_encodej = pd.merge(self_encodej, encoded_histj, left_index=True, right_in
dex=True)
self_encodec = pd.merge(self_encodec, encoded_histc, left_index=True, right_in
dex=True)

orig_cols.append(column)
self_encodej.head()
```

Out[1832]:

listingid	black_x	blue_x	brown_x	green	metallic	pearlcoat	clearcoat	granite	red_x	silver_x
4777	1	0	0	0	0	1	0	0	0	0
6242	1	0	0	0	0	1	0	0	0	0
10882	0	0	0	0	1	0	1	0	0	1
12013	0	1	0	0	0	1	0	0	0	0
12334	0	0	0	0	0	0	0	0	1	0

5 rows × 31 columns

In [1833]:
```
col+=1
feat_ptrj,column = setFeatPtr(input_jeeps,col)
print(feat_ptrj.value_counts())
feat_ptrc,column = setFeatPtr(input_caddys,col)
print(feat_ptrc.value_counts())
feat_ptrj.head()
```

```
vehlistdays
95.771331     12
6.913530       6
95.771343      5
106.541817     5
109.752431     5
              ..
145.458866     1
7.672801       1
202.458657     1
26.955544      1
20.678600      1
Name: count, Length: 3243, dtype: int64
vehlistdays
42.228171     6
11.784618     6
29.929606     5
11.093495     5
3.422338      4
             ..
12.586238     1
146.129201    1
265.647998    1
4.786713      1
73.868426     1
Name: count, Length: 1501, dtype: int64
```

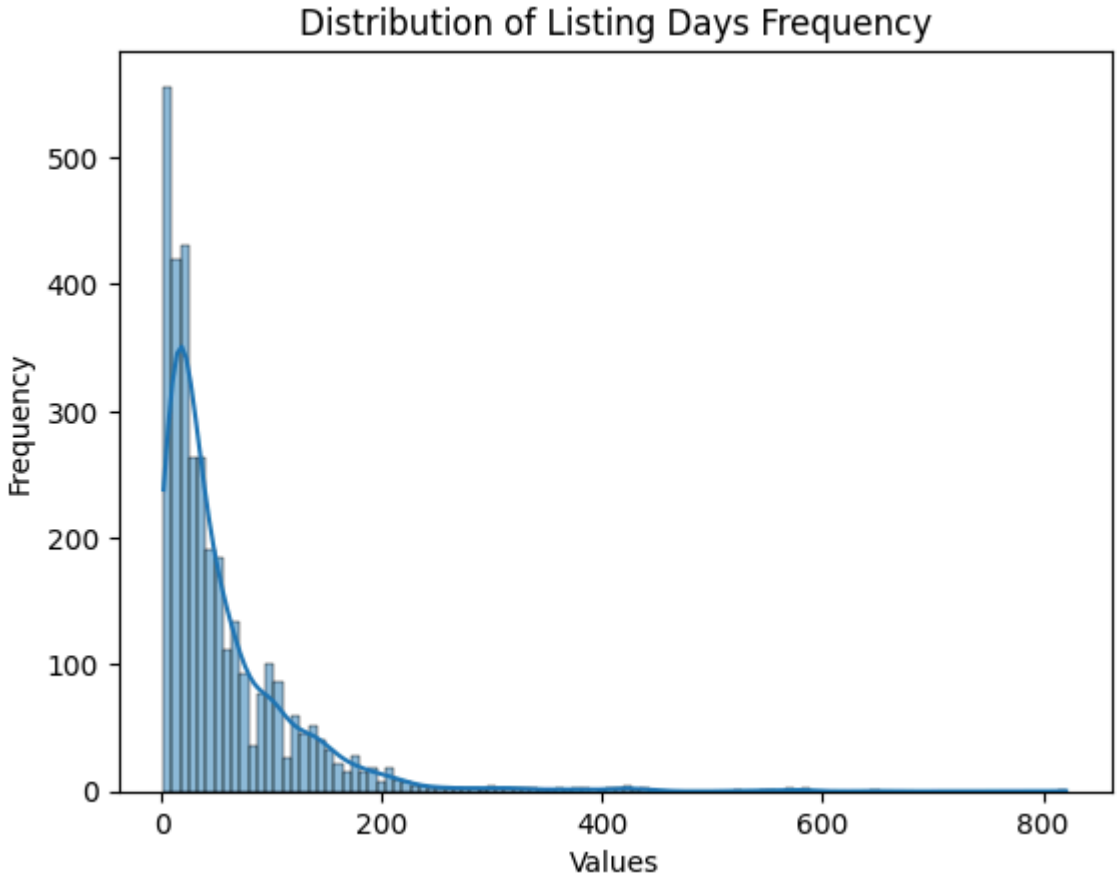
Out[1833]:
```
listingid
4777       28.107014
6242       59.816875
10882      30.967500
12013     194.482338
12334      28.849537
Name: vehlistdays, dtype: float64
```

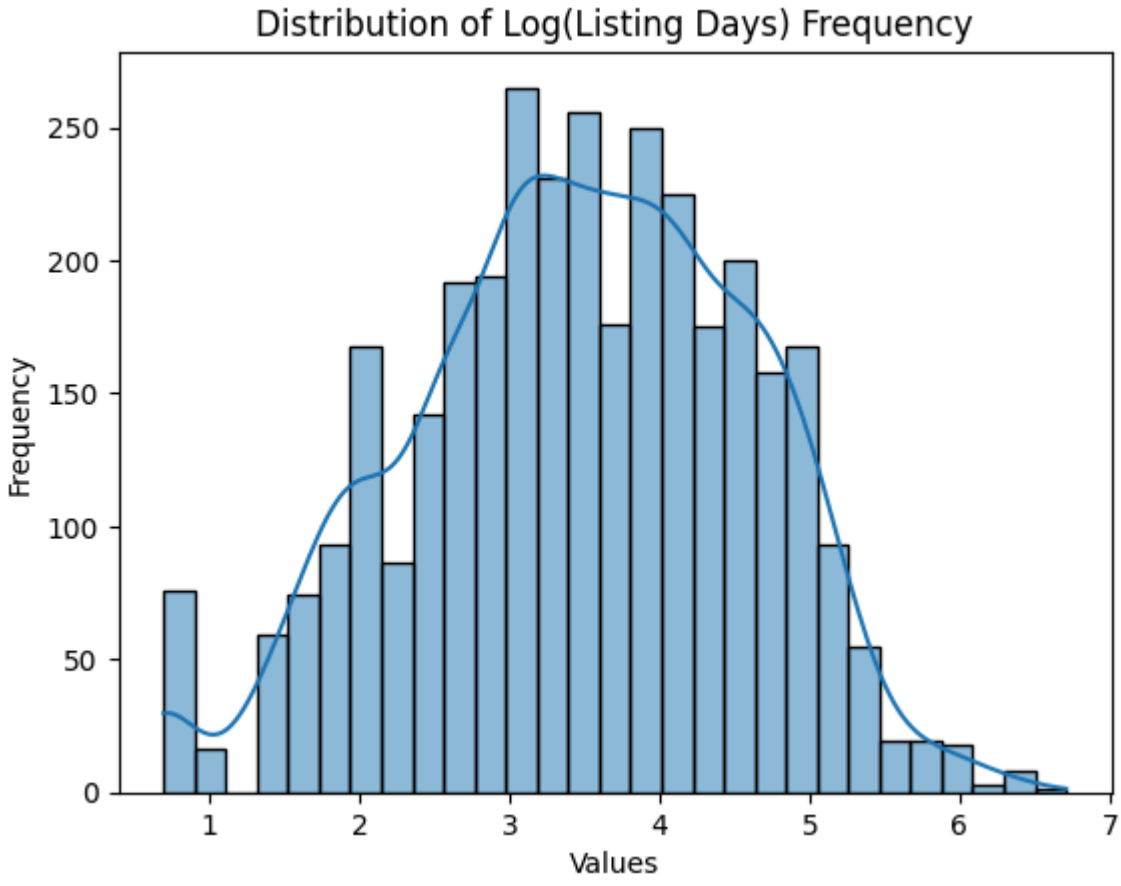
In [1834]:
```
#Use ceiling in order to round to whole days and start the listings
#on day 1 rather than day 0
feat_ptrj = pd.Series(np.ceil(feat_ptrj),index=feat_ptrj.index)
feat_ptrc = pd.Series(np.ceil(feat_ptrc),index=feat_ptrc.index)

feat_ptrj.head()
```

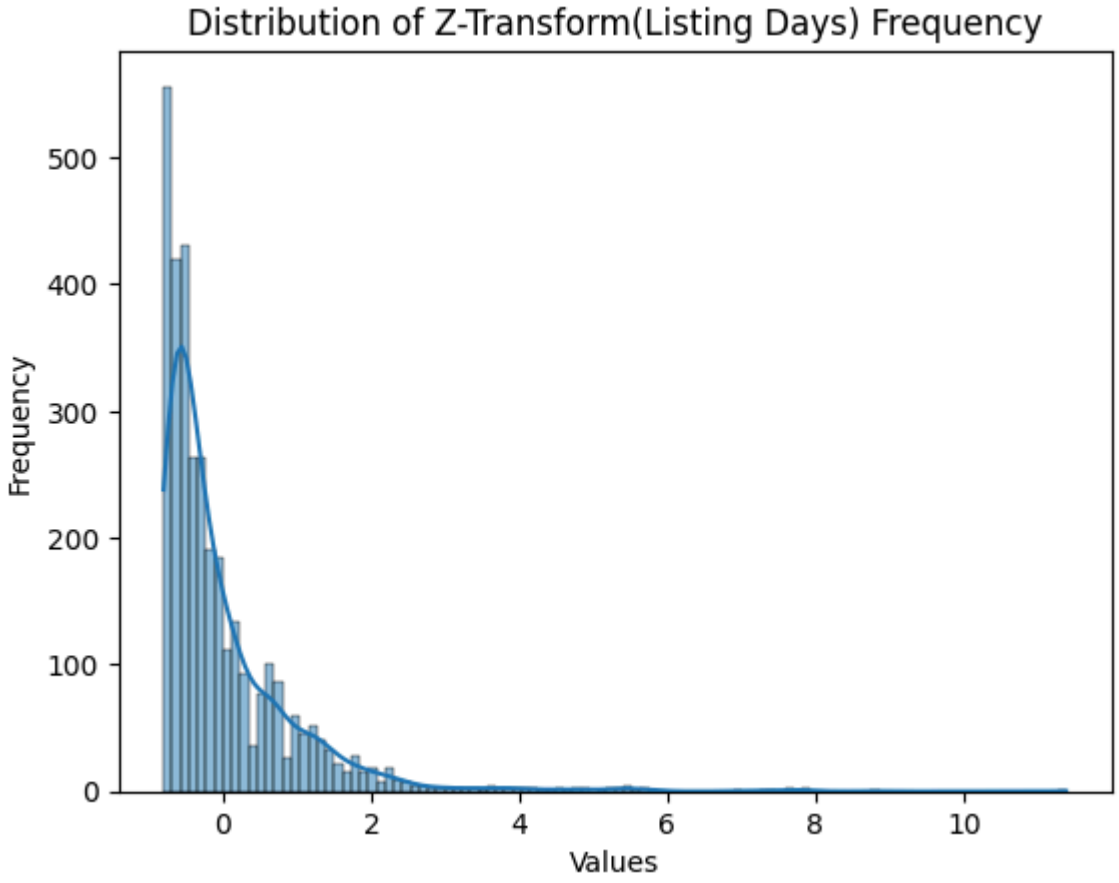
Out[1834]:
```
listingid
4777       29.0
6242       60.0
10882      31.0
12013     195.0
12334      29.0
Name: vehlistdays, dtype: float64
```

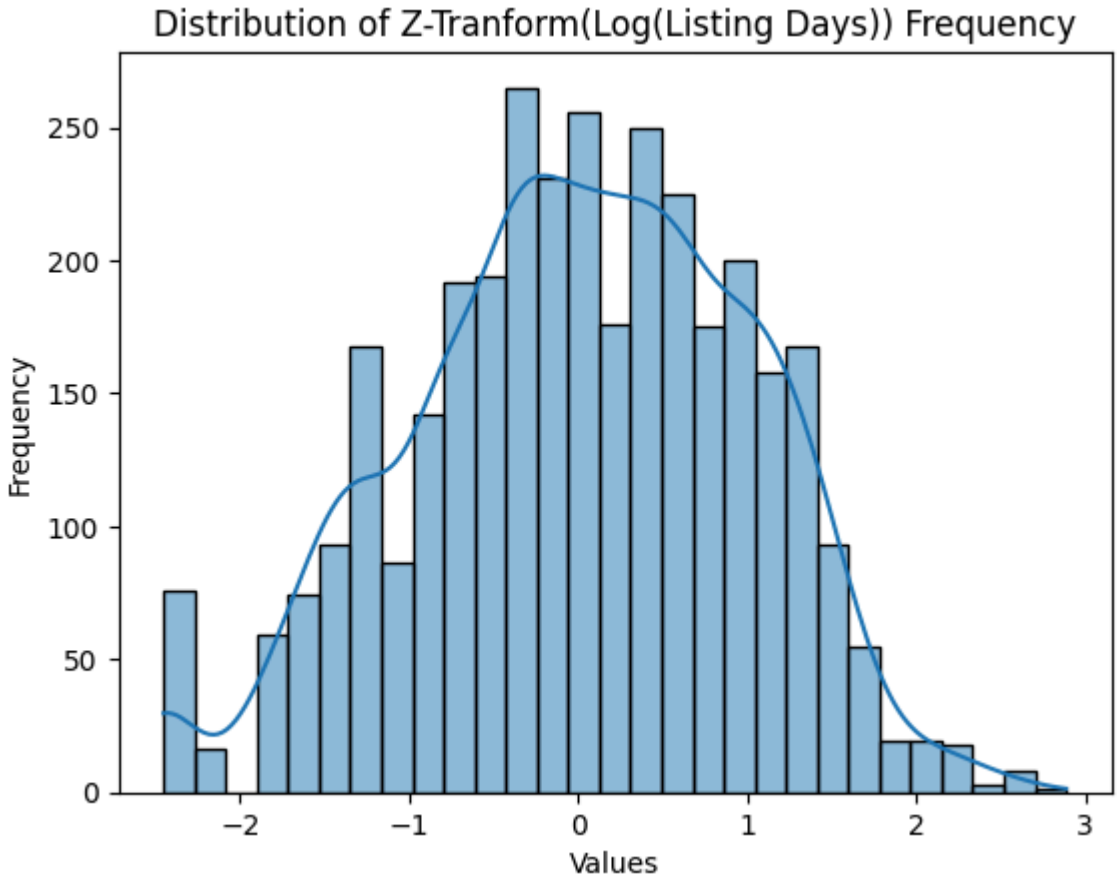
In [1835]:

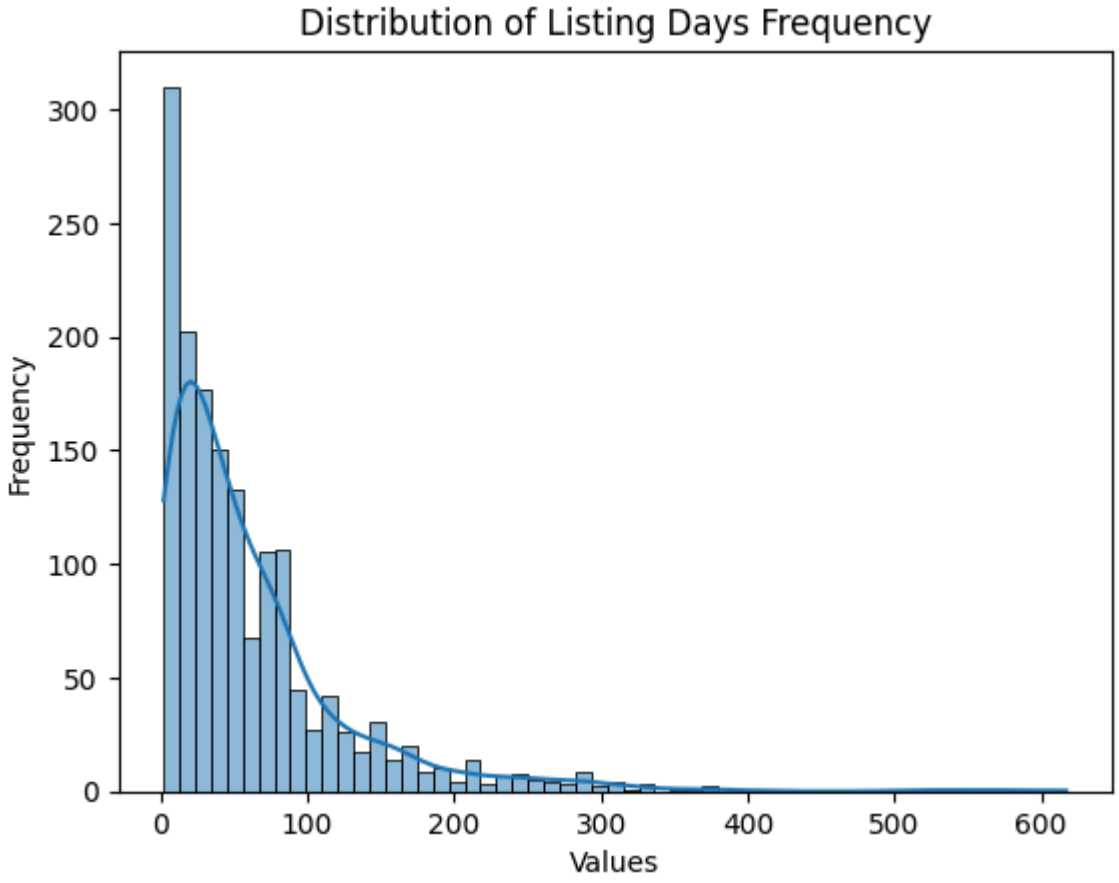
```
plotDist(feat_ptrj,"Distribution of Listing Days Frequency")
plotDist(np.log(feat_ptrj),"Distribution of Log(Listing Days) Frequency")
plotDist(zScoreTransform(feat_ptrj),"Distribution of Z-Transform(Listing Days)
Frequency")
plotDist(zScoreTransform(np.log(feat_ptrj)),"Distribution of Z-Tranform(Log(Li
sting Days)) Frequency")

plotDist(feat_ptrc,"Distribution of Listing Days Frequency")
plotDist(np.log(feat_ptrc),"Distribution of Log(Listing Days) Frequency")
plotDist(zScoreTransform(feat_ptrc),"Distribution of Z-Transform(Listing Days)
Frequency")
plotDist(zScoreTransform(np.log(feat_ptrc)),"Distribution of Z-Tranform(Log(Li
sting Days)) Frequency")
```

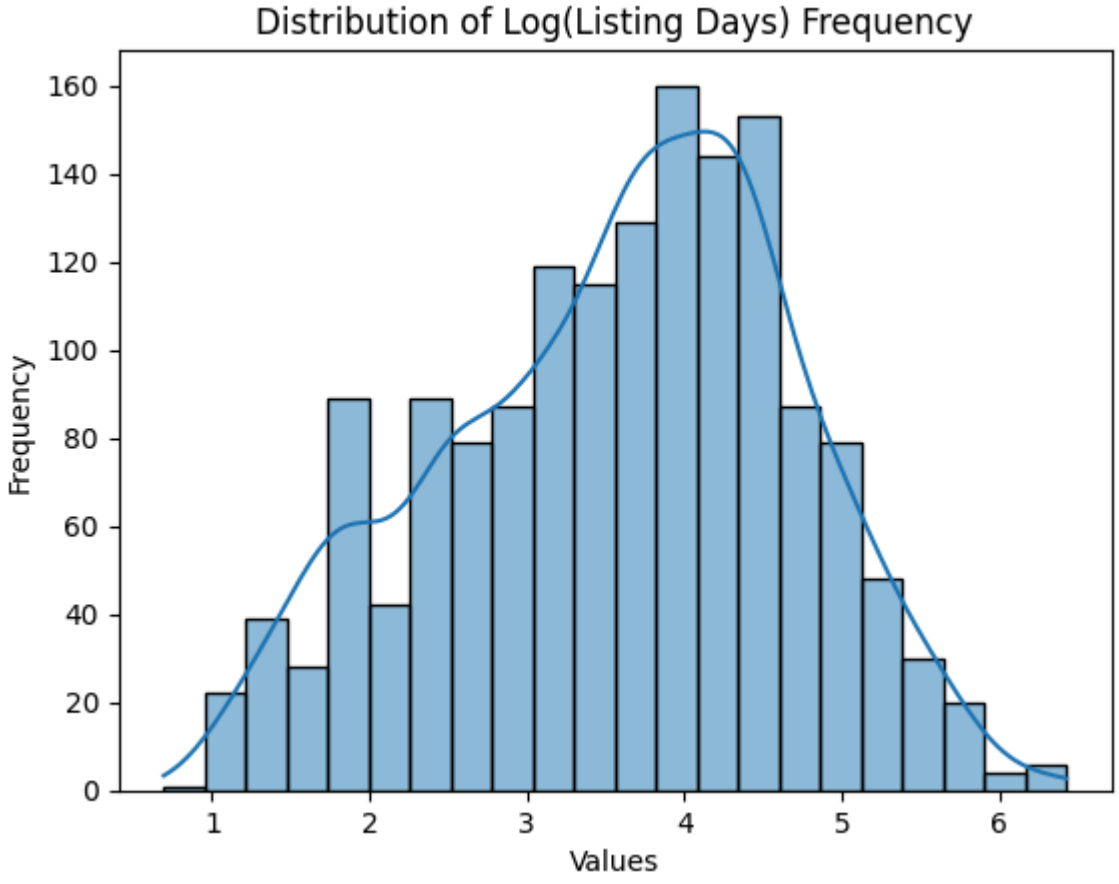

Distribution of Listing Days Frequency
Frequency
Values
0
100
200
300
400
500
0
200
400
600
800

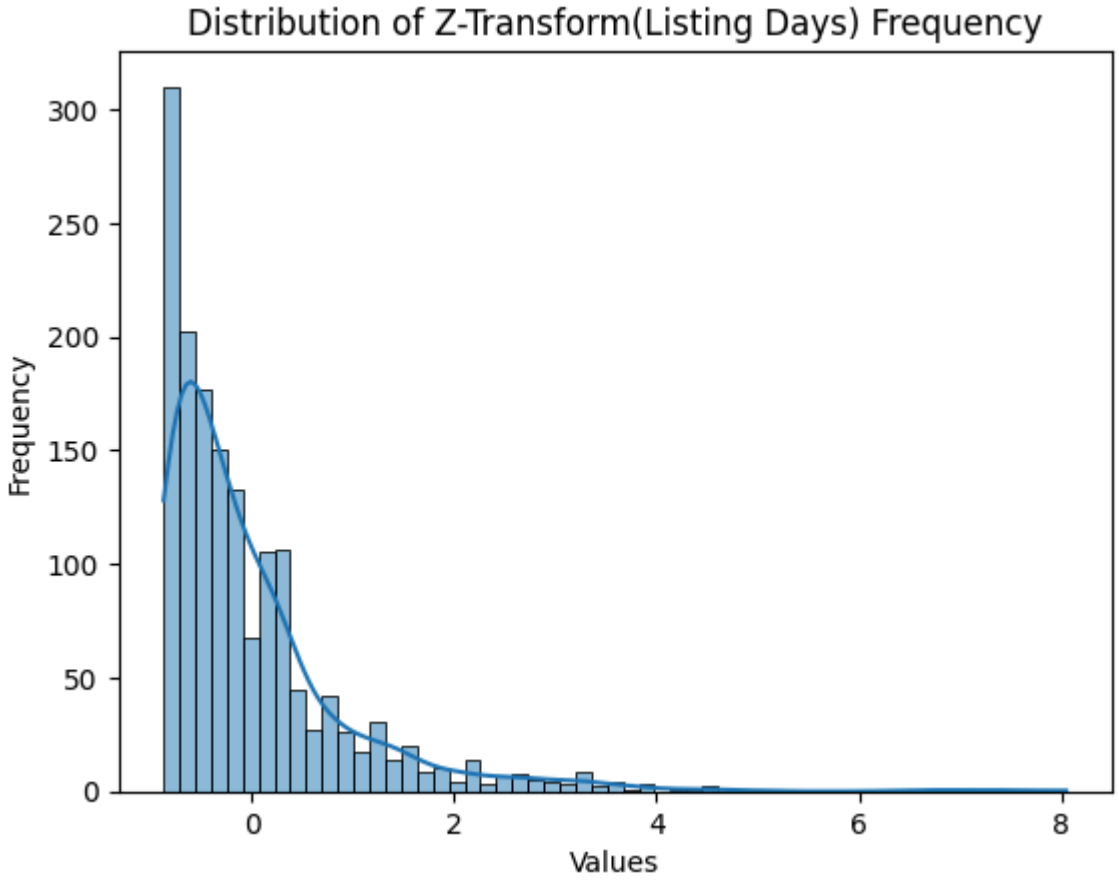

Distribution of Log(Listing Days) Frequency
Frequency
Values
0
50
100
150
200
250
1
2
3
4
5
6
7

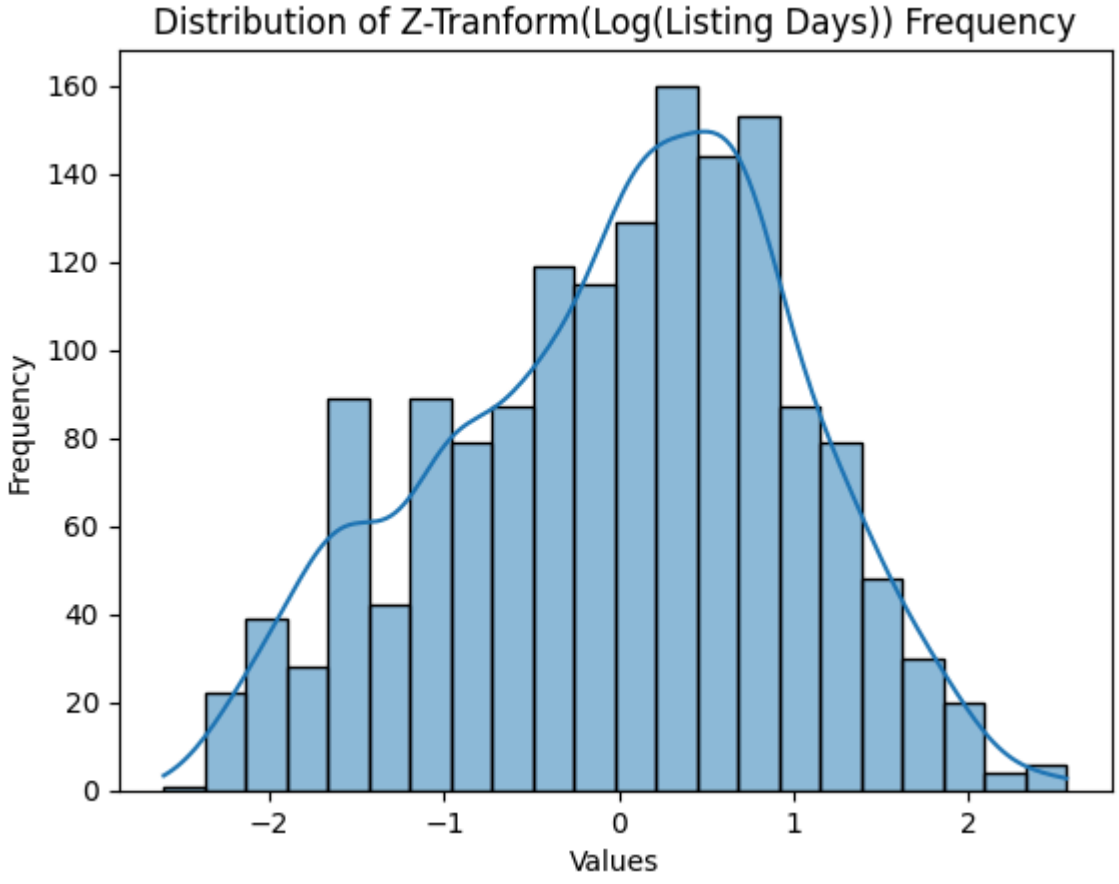

Distribution of Z-Transform(Listing Days) Frequency
Frequency
Values
0
100
200
300
400
500
0
2
4
6
8
10


Distribution of Z-Tranform(Log(Listing Days)) Frequency
Frequency
Values
0
50
100
150
200
250
−2
−1
0
1
2
3


Distribution of Listing Days Frequency
Frequency
Values
0
50
100
150
200
250
300
0
100
200
300
400
500
600


Distribution of Log(Listing Days) Frequency
Frequency
Values
0
20
40
60
80
100
120
140
160
1
2
3
4
5
6


Distribution of Z-Transform(Listing Days) Frequency
Frequency
Values
0
50
100
150
200
250
300
0
2
4
6
8


Distribution of Z-Tranform(Log(Listing Days)) Frequency
Frequency
Values
0
20
40
60
80
100
120
140
160
−2
−1
0
1
2

In [1836]:
```
#CHOOSE LOG VALUE
#feat_ptrj = np.log(feat_ptrj)
#feat_ptrc = np.log(feat_ptrc)

print(feat_ptrj)
print(feat_ptrc)
```

```
listingid
4777        29.0
6242        60.0
10882       31.0
12013      195.0
12334       29.0
           ...
8610847    103.0
8612731     39.0
8614177     40.0
8615510      5.0
8620012     21.0
Name: vehlistdays, Length: 3420, dtype: float64
listingid
7108        99.0
21448        6.0
21807        7.0
30524       12.0
34061       26.0
           ...
8599564     35.0
8601212     19.0
8604205     20.0
8616294    185.0
8617378     74.0
Name: vehlistdays, Length: 1570, dtype: float64
```

In [1837]:
```
input_jeeps[column] = feat_ptrj
input_caddys[column] = feat_ptrc
log_cols.append(column)

col+=1
input_jeeps.head()
```

Out[1837]:

listingid	sellercity	sellerispriv	sellerlistsrc	sellername	sellerrating	sellerrevcnt	sellerstate	sell
4777	17	False	Jeep Certified Program	Wilde Chrysler Jeep Dodge Ram & Subaru	4.8	1405	WI	531
6242	4	False	Inventory Command Center	Century Dodge Chrysler Jeep RAM	4.4	21	MO	633
10882	2	False	Digital Motorworks (DMi)	Paul Brown Chrysler Dodge Jeep RAM Kia	3.0	51	NY	147
12013	3	False	Digital Motorworks (DMi)	Sierra Motor Mall	3.5	17	IL	613
12334	25	False	Digital Motorworks (DMi)	Larry Roesch Dodge Chrysler Jeep RAM	4.6	240	IL	601

5 rows × 29 columns

In [1838]:
```
feat_ptrj,column = setFeatPtr(input_jeeps,col)
print(feat_ptrj.value_counts())
feat_ptrc,column = setFeatPtr(input_caddys,col)
print(feat_ptrc.value_counts())
feat_ptrj.head()
```

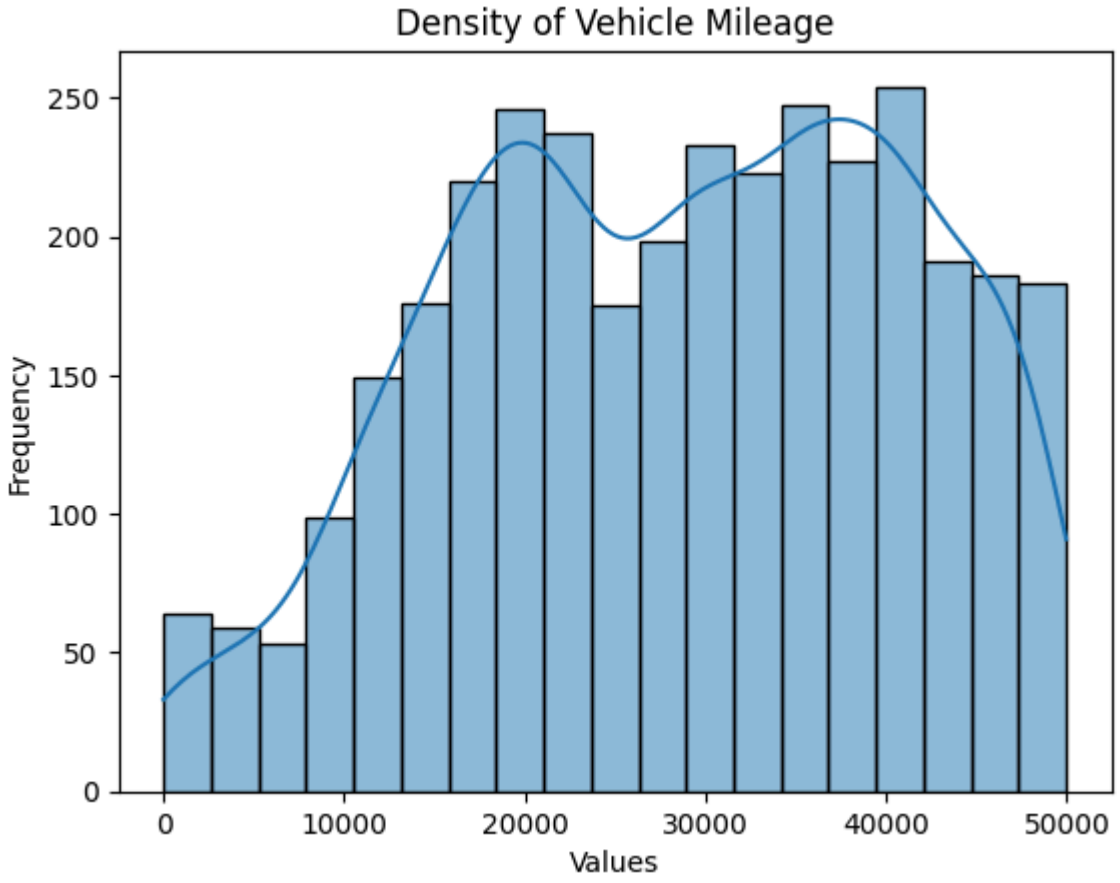
```
vehmake
Jeep    3420
Name: count, dtype: int64
vehmake
Cadillac    1570
Name: count, dtype: int64
```

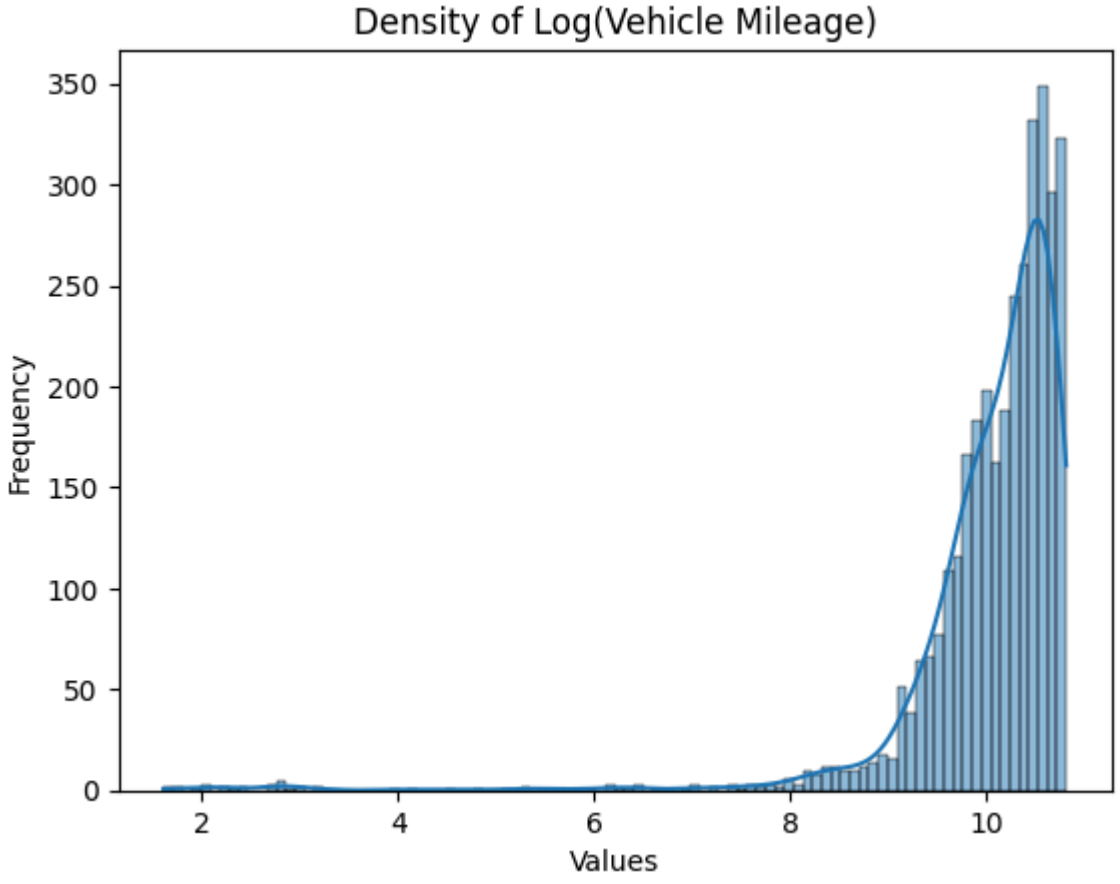
Out[1838]:
```
listingid
4777     Jeep
6242     Jeep
10882    Jeep
12013    Jeep
12334    Jeep
Name: vehmake, dtype: object
```

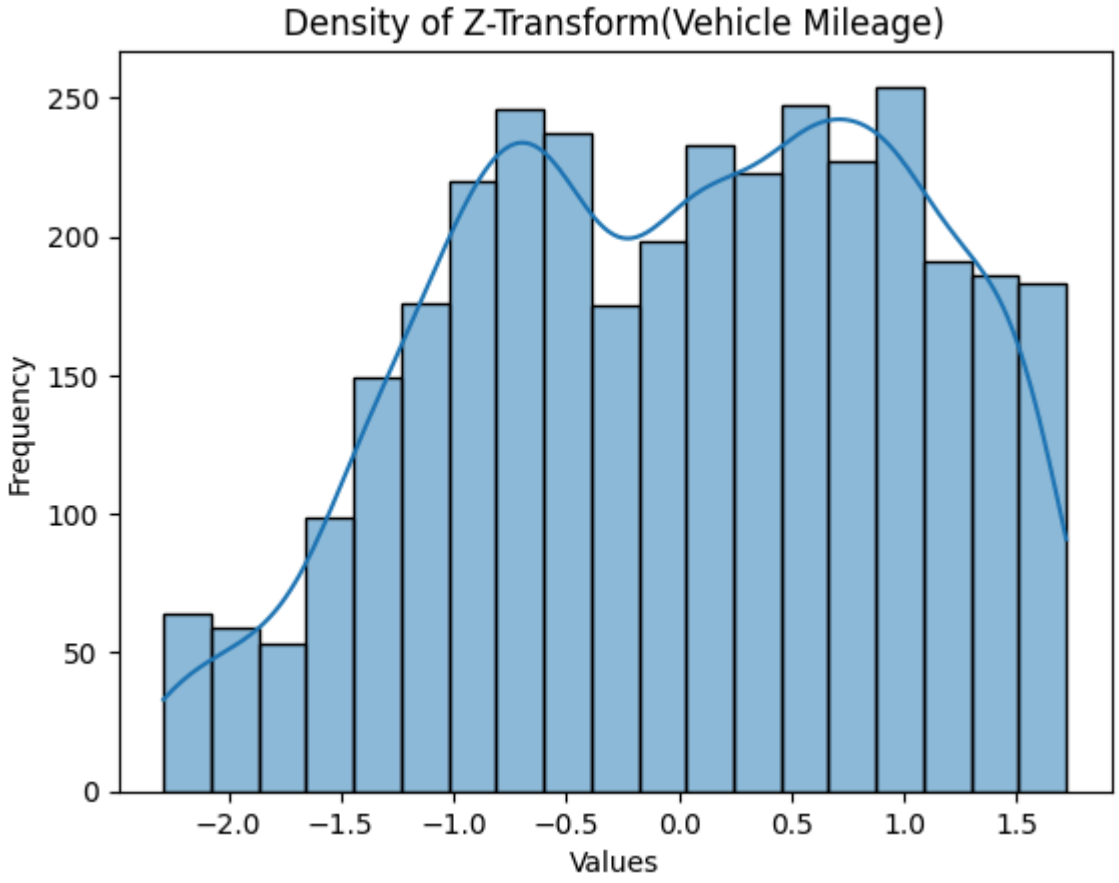
In [1839]:
```
#The defining attribute of each list, going to keep the same for now in case t
he handler functions become "make" specific
same_cols.append(column)

col+=1
feat_ptrj,column = setFeatPtr(input_jeeps,col)
print(feat_ptrj.value_counts())
feat_ptrc,column = setFeatPtr(input_caddys,col)
print(feat_ptrc.value_counts())
```

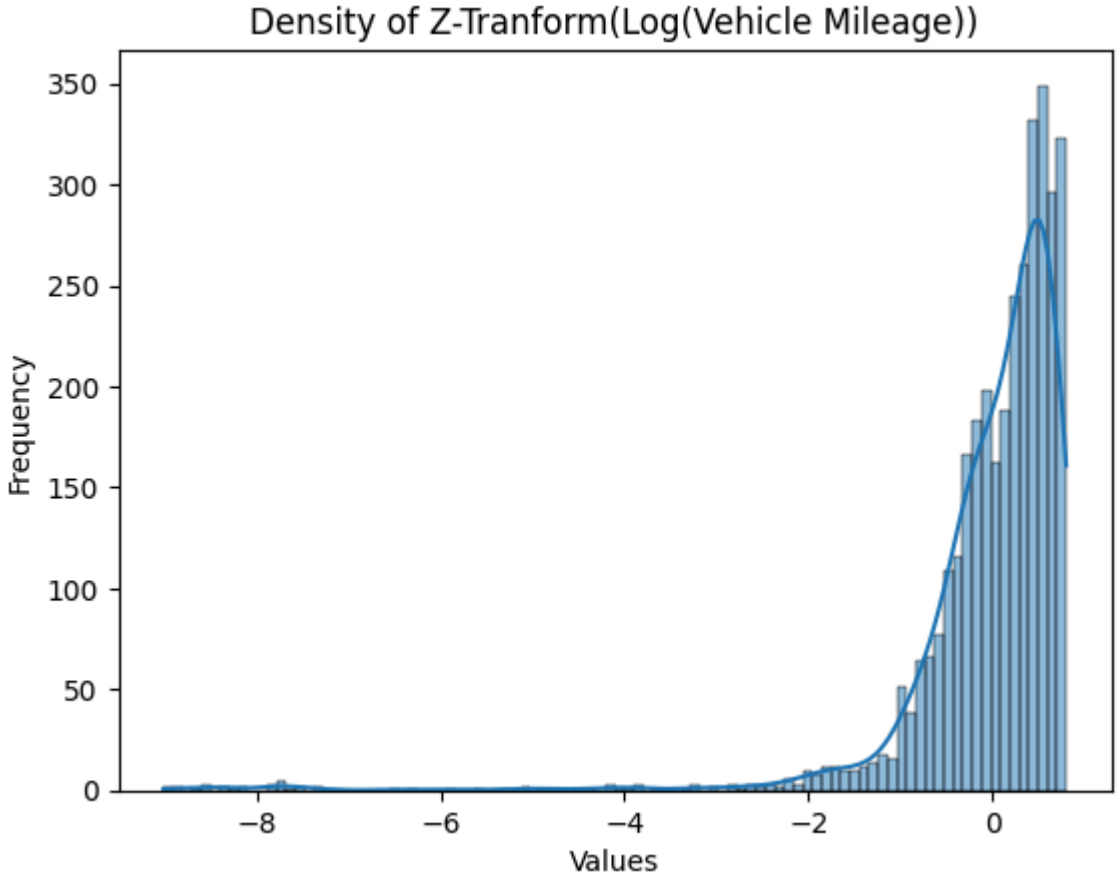
```
vehmileage
40277.0    24
37151.0    20
36334.0    17
20207.0    12
40592.0    12
           ..
26449.0     1
41859.0     1
31281.0     1
17332.0     1
20039.0     1
Name: count, Length: 2940, dtype: int64
vehmileage
17536.0    27
26181.0    24
21098.0    23
5343.0     22
11310.0    21
           ..
24553.0     1
13727.0     1
10774.0     1
17201.0     1
38146.0     1
Name: count, Length: 1395, dtype: int64
```

In [1840]:
```
plotDist(feat_ptrj,"Density of Vehicle Mileage")
plotDist(np.log(feat_ptrj),"Density of Log(Vehicle Mileage)")
plotDist(zScoreTransform(feat_ptrj),"Density of Z-Transform(Vehicle Mileage)")
plotDist(zScoreTransform(np.log(feat_ptrj)),"Density of Z-Tranform(Log(Vehicle
Mileage))")
```


Density of Vehicle Mileage
Frequency
Values
0
50
100
150
200
250
0
10000
20000
30000
40000
50000


Density of Log(Vehicle Mileage)
Frequency
Values
0
50
100
150
200
250
300
350
2
4
6
8
10


Density of Z-Transform(Vehicle Mileage)
Frequency
Values
0
50
100
150
200
250
−2.0
−1.5
−1.0
−0.5
0.0
0.5
1.0
1.5


Density of Z-Tranform(Log(Vehicle Mileage))
Frequency
Values
0
50
100
150
200
250
300
350
−8
−6
−4
−2
0

In [1841]:
```
#ORIGINAL DATA LOOKS ~NORMAL~

same_cols.append(column)
col+=1
feat_ptrj,column = setFeatPtr(input_jeeps,col)
print(feat_ptrj.value_counts())
feat_ptrc,column = setFeatPtr(input_caddys,col)
print(feat_ptrc.value_counts())
```

```
vehmodel
Grand Cherokee    3420
Name: count, dtype: int64
vehmodel
XT5    1570
Name: count, dtype: int64
```

In [1842]:
```
#ALREADY HAVE JEEP/CADILLAC ENCODED COLUMNS WHICH HAVE A DIRECT CORRELATION TO
THIS
#WILL REMOVE THIS EXTRANEOUS COLUMN
feats_to_drop.append(column)

col+=1
feat_ptrj,column = setFeatPtr(input_jeeps,col)
print(feat_ptrj.value_counts())
feat_ptrc,column = setFeatPtr(input_caddys,col)
print(feat_ptrc.value_counts())
```

```
vehpricelabel
Good Deal     2484
Great Deal     508
Fair Price     428
Name: count, dtype: int64
vehpricelabel
Good Deal     1250
Great Deal     186
Fair Price     134
Name: count, dtype: int64
```

In [1843]:
```
encoded_cols.append(column)

col+=1
feat_ptrj,column = setFeatPtr(input_jeeps,col)
feat_ptrc,column = setFeatPtr(input_caddys,col)
feat_ptrj.head()
```

Out[1843]:
```
listingid
4777     Backed by a rigorous 125-point inspection by f...
6242     Drop by to see us and you will quickly see how...
10882    Priced below KBB Fair Purchase Price! Clean CA...
12013    2017 Jeep Grand Cherokee Laredo True Blue Pear...
12334    BLIND SPOT/CROSS PATH DETECTION, APPLE CARPLAY...
Name: vehsellernotes, dtype: object
```

In [1844]:
```
#ELIMINATE WORDS THAT APPEAR IN MORE THAN max_doc_freq OF DOCUMENTS (DOCUMENT
~ ROW)
#WILL GET RID OF COMMON WORDS SUCH AS "THE", "A", etc.
#LIMIT VOCABULARY TO max_feats COLUMNS (ONE FOR EACH WORD)
tfidfj = TfidfVectorizer(max_df=.50,max_features=60)
tf_revj = tfidfj.fit(feat_ptrj.copy())
vocab2j = tf_idfTokenizer(feat_ptrj,tf_revj)
#THOUGHT: TUNE THE HYPERPARAMETERS TO OPTIMIZE THE TOKENIZER?
vocab2j.head()
```

Out[1844]:

listingid	4wd	6l	all	amp	as	auto	automatic	black	bluetooth
4777	0.000000	0.000000	0.159555	0.000000	0.000000	0.000000	0.000000	0.0	0.000000
6242	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.0	0.000000
10882	0.148115	0.160205	0.000000	0.000000	0.000000	0.000000	0.151473	0.0	0.000000
12013	0.108562	0.000000	0.352540	0.148565	0.440374	0.000000	0.111023	0.0	0.111715
12334	0.124005	0.000000	0.134230	0.000000	0.000000	0.157404	0.000000	0.0	0.000000

5 rows × 60 columns

In [1845]:
```
#ELIMINATE WORDS THAT APPEAR IN MORE THAN max_doc_freq OF DOCUMENTS (DOCUMENT
~ ROW)
#WILL GET RID OF COMMON WORDS SUCH AS "THE", "A", etc.
#LIMIT VOCABULARY TO max_feats COLUMNS (ONE FOR EACH WORD)
tfidfc = TfidfVectorizer(max_df=.50,max_features=60)
tf_revc = tfidfc.fit(feat_ptrc.copy())
vocab2c = tf_idfTokenizer(feat_ptrc,tf_revc)
#THOUGHT: TUNE THE HYPERPARAMETERS TO OPTIMIZE THE TOKENIZER?
vocab2c.head()
```

Out[1845]:

listingid	2017	6l	alert	all	amp	are	assist	at	be	call	.
7108	0.0	0.0	0.0	0.000000	0.000000	0.195364	0.121573	0.095904	0.000000	0.000000	.
21448	0.0	0.0	0.0	0.000000	0.000000	0.259235	0.000000	0.000000	0.000000	0.000000	.
21807	0.0	0.0	0.0	0.000000	0.194211	0.000000	0.000000	0.147205	0.000000	0.000000	.
30524	0.0	0.0	0.0	0.170694	0.000000	0.331159	0.000000	0.000000	0.214261	0.000000	.
34061	0.0	0.0	0.0	0.000000	0.000000	0.000000	0.224589	0.177168	0.000000	0.321063	.

5 rows × 60 columns

In [1846]:
```
#DROP ORIGINAL STATE COLUMN AND LATER REPLACE WITH ENCODED MATRIX COLUMNS
feats_to_drop.append(column)
tokenize_cols.append(column)
input_jeeps.head()
```

Out[1846]:

listingid	sellercity	sellerispriv	sellerlistsrc	sellername	sellerrating	sellerrevcnt	sellerstate	sell
4777	17	False	Jeep Certified Program	Wilde Chrysler Jeep Dodge Ram & Subaru	4.8	1405	WI	531
6242	4	False	Inventory Command Center	Century Dodge Chrysler Jeep RAM	4.4	21	MO	633
10882	2	False	Digital Motorworks (DMi)	Paul Brown Chrysler Dodge Jeep RAM Kia	3.0	51	NY	147
12013	3	False	Digital Motorworks (DMi)	Sierra Motor Mall	3.5	17	IL	613
12334	25	False	Digital Motorworks (DMi)	Larry Roesch Dodge Chrysler Jeep RAM	4.6	240	IL	601

5 rows × 29 columns

In [1847]:
```
vocabjs = pd.merge(vocab1j,vocab2j,left_index=True,right_index=True)
vocabcs = pd.merge(vocab1c,vocab2c,left_index=True,right_index=True)
vocabjs.head()
```

Out[1847]:

listingid	air	alarm	alloy	aluminum	am	anti	antilock	auto_x	automatic_
4777	0.000000	0.000000	0.0	0.362589	0.260283	0.000000	0.000000	0.359060	0.26124
6242	0.000000	0.000000	0.0	0.000000	0.000000	0.000000	0.397475	0.426209	0.00000
10882	0.000000	0.303521	0.0	0.000000	0.000000	0.292306	0.000000	0.000000	0.00000
12013	0.289613	0.000000	0.0	0.000000	0.256202	0.000000	0.000000	0.000000	0.25714
12334	0.000000	0.303521	0.0	0.000000	0.000000	0.292306	0.000000	0.000000	0.00000

5 rows × 90 columns

In [1848]:
```
vocabcs.head()
```

Out[1848]:

listingid	all_x	aluminum	android	apple	assist_x	auto	automatic	auxiliary	beverage
7108	0.0	0.273007	0.000000	0.000000	0.000000	0.305368	0.318800	0.000000	0.000000
21448	0.0	0.000000	0.308276	0.315974	0.000000	0.274410	0.572961	0.325261	0.000000
21807	0.0	0.169650	0.213179	0.218502	0.186359	0.189760	0.198107	0.449849	0.000000
30524	0.0	0.000000	0.000000	0.000000	0.332808	0.000000	0.353788	0.000000	0.000000
34061	0.0	0.279240	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.317566

5 rows × 90 columns

In [1849]:
```
col+=1
feat_ptrj,column = setFeatPtr(input_jeeps,col)
print(feat_ptrj.value_counts())
feat_ptrc,column = setFeatPtr(input_caddys,col)
print(feat_ptrc.value_counts())
```

```
vehtype
Used    3420
Name: count, dtype: int64
vehtype
Used    1570
Name: count, dtype: int64
```

In [1850]:
```
#ENTIRE COLUMN HAS VALUE "USED".....  DROPPING....
feats_to_drop.append(column)

col+=1
feat_ptrj,column = setFeatPtr(input_jeeps,col)
print(feat_ptrj.value_counts())
feat_ptrc,column = setFeatPtr(input_caddys,col)
print(feat_ptrc.value_counts())
```

```
vehtransmission
8-Speed Automatic                2377
8-Speed Automatic w/OD            596
Automatic                         189
Automatic 8-Speed                  54
8-Speed Automatic (845RE)          48
AUTOMATIC                          41
8-Speed Shiftable Automatic        36
8-Speed A/T                        33
8-Speed Automatic (850RE)          11
8-Spd TorqueFlite Automatic         7
8 Speed Automatic                   6
8-Speed Automatic (8HP70)           5
8-Speed                             4
Automatic, 8-Spd                    2
8 speed automatic                   2
8-Spd Auto 850RE Trans (Make        1
8-SPEED AUTOMATIC                   1
AUTO                                1
8-Speed TorqueFlite Automatic       1
Automatic w/OD                      1
aujtomatic                          1
A                                   1
Not Specified                       1
8-Spd Auto 850RE Trans (Make)       1
Name: count, dtype: int64
vehtransmission
8-Speed Automatic              1396
Automatic                        83
AUTOMATIC                        28
8-Speed Shiftable Automatic      26
8-Speed A/T                      11
Automatic 8-Speed                 9
8-Speed                           7
6-Speed Automatic                 2
Automatic, 8-Spd                  2
8-SPEED AUTOMATIC                 2
8 Speed Automatic                 1
Not Specified                     1
automatic                         1
Shiftable Automatic               1
Name: count, dtype: int64
```

In [1851]:
```
#BASICALLY ALL 8-SPEED SO IT GETS DROPPED
feats_to_drop.append(column)

col+=1
feat_ptrj,column = setFeatPtr(input_jeeps,col)
print(feat_ptrj.value_counts())
feat_ptrc,column = setFeatPtr(input_caddys,col)
print(feat_ptrc.value_counts())
```

```
vehyear
2015    1278
2017     863
2018     861
2016     380
2019      38
Name: count, dtype: int64
vehyear
2018    798
2017    698
2019     74
Name: count, dtype: int64
```

In [1852]:
```
#ONLY 5 UNIQUES IN OUR DATASET SO WE WILL ONE HOT ENCODE THE CATEGORIES
agej = calculate_age(feat_ptrj)
agec = calculate_age(feat_ptrc)

input_jeeps[column] =  agej
input_caddys[column] = agec

same_cols.append(column)

print(encoded_cols)
print(self_encodej)
print(self_encodec)
print(tokenize_cols)
```

```
['sellerlistsrc', 'sellerstate', 'vehdrivetrain', 'vehfuel', 'vehpricelabel']
           black_x  blue_x  brown_x  green  metallic  pearlcoat  clearcoat  \
listingid
4777             1       0        0      0         0          1          0
6242             1       0        0      0         0          1          0
10882            0       0        0      0         1          0          1
12013            0       1        0      0         0          1          0
12334            0       0        0      0         0          0          0
...            ...     ...      ...    ...       ...        ...        ...
8610847          0       0        1      0         1          0          1
8612731          1       0        0      0         0          1          0
8614177          1       0        0      0         0          0          0
8615510          0       0        0      0         0          0          0
8620012          1       0        0      0         0          1          0

           granite  red_x  silver_x  ...  tan  cirrus  carbon  plum  none_y
\
listingid                            ...
4777             0      0         0  ...    0       0       0     0       0
6242             0      0         0  ...    0       0       0     0       0
10882            0      0         1  ...    0       0       0     0       0
12013            0      0         0  ...    0       0       0     0       0
12334            0      1         0  ...    0       0       0     0       0
...            ...    ...       ...  ...  ...     ...     ...   ...     ...
8610847          0      0         0  ...    0       0       0     0       0
8612731          0      0         0  ...    0       0       0     0       0
8614177          0      0         0  ...    0       0       0     0       0
8615510          0      0         1  ...    0       0       0     0       0
8620012          0      0         0  ...    0       0       0     0       0

           Accident(s) Reported  Buyback Protection Eligible  \
listingid
4777                          0                           1
6242                          0                           1
10882                         0                           1
12013                         1                           1
12334                         0                           1
...                         ...                         ...
8610847                       0                           1
8612731                       0                           1
8614177                       0                           1
8615510                       1                           1
8620012                       0                           1

           Non-Personal Use Reported  Title Issue(s) Reported  \
listingid
4777                               0                        0
6242                               1                        0
10882                              1                        0
12013                              1                        0
12334                              1                        0
...                              ...                      ...
8610847                            0                        0
8612731                            0                        0
8614177                            0                        0
8615510                            1                        0
8620012                            1                        0
```

```
           None of the above
listingid
4777                       0
6242                       0
10882                      0
12013                      0
12334                      0
...                      ...
8610847                    0
8612731                    0
8614177                    0
8615510                    0
8620012                    0

[3420 rows x 31 columns]
           black_x  blue_x  brown_x  green  metallic  pearlcoat  clearcoat  \
listingid
7108             0       0        0      0         1          0          0
21448            0       0        0      0         1          0          0
21807            0       0        0      0         0          0          0
30524            0       0        0      0         0          0          0
34061            0       0        0      0         1          0          0
...            ...     ...      ...    ...       ...        ...        ...
8599564          1       0        0      0         1          0          0
8601212          0       0        0      0         0          0          0
8604205          1       0        0      0         1          0          0
8616294          1       0        0      0         0          0          0
8617378          1       0        0      0         0          0          0

           granite  red_x  silver_x  ...  tan  cirrus  carbon  plum  none_y
\
listingid                             ...
7108             0      0         1  ...    0       1       0     0       0
21448            1      0         0  ...    0       0       0     0       0
21807            0      0         0  ...    0       1       0     0       0
30524            0      0         0  ...    1       0       0     0       0
34061            1      0         0  ...    0       1       0     0       0
...            ...    ...       ...  ...  ...     ...     ...   ...     ...
8599564          0      0         0  ...    0       0       0     0       0
8601212          0      0         0  ...    0       0       0     0       0
8604205          0      0         0  ...    0       0       0     0       0
8616294          0      0         0  ...    0       0       0     0       0
8617378          0      0         0  ...    0       0       0     0       0

           Accident(s) Reported  Buyback Protection Eligible  \
listingid
7108                          0                           1
21448                         0                           1
21807                         0                           1
30524                         0                           1
34061                         0                           1
...                         ...                         ...
8599564                       0                           1
8601212                       0                           1
8604205                       0                           1
8616294                       0                           1
```

```
8617378                       0                            1

           Non-Personal Use Reported  Title Issue(s) Reported  \
listingid
7108                               1                        0
21448                              1                        0
21807                              0                        0
30524                              1                        0
34061                              1                        0
...                              ...                      ...
8599564                            1                        0
8601212                            1                        0
8604205                            1                        0
8616294                            0                        0
8617378                            1                        0

           None of the above
listingid
7108                       0
21448                      0
21807                      0
30524                      0
34061                      0
...                      ...
8599564                    0
8601212                    0
8604205                    0
8616294                    0
8617378                    0

[1570 rows x 31 columns]
['vehfeats', 'vehsellernotes']
```

In [1853]:

```
feats_handled = (log_cols+encoded_cols+freq_cols+same_cols+mask_cols+tokenize_
cols+orig_cols)
#print(orig_cols)
print("HANDLED FEATS:",feats_handled)
print("FEATS TO DROP:",feats_to_drop)


overlap = list(set(feats_handled) & set(feats_to_drop))

print("Overlapping elements:", overlap)
print("SIZE IS 26: ", len(feats_handled+feats_to_drop)-len(overlap)==26)
```

```
HANDLED FEATS: ['vehlistdays', 'sellerlistsrc', 'sellerstate', 'vehdrivetrai
n', 'vehfuel', 'vehpricelabel', 'sellercity', 'sellerrating', 'sellerrevcnt',
'vehmake', 'vehmileage', 'vehyear', 'vehcertified', 'vehfeats', 'vehsellernot
es', 'vehcolorext', 'vehcolorint', 'vehdrivetrain', 'vehengine', 'vehhistor
y']
FEATS TO DROP: ['sellerispriv', 'sellername', 'sellerzip', 'vehbodystyle', 'v
ehcolorext', 'vehcolorint', 'vehengine', 'vehfeats', 'vehhistory', 'vehmode
l', 'vehsellernotes', 'vehtype', 'vehtransmission']
Overlapping elements: ['vehengine', 'vehhistory', 'vehcolorint', 'vehsellerno
tes', 'vehfeats', 'vehcolorext']
SIZE IS 26:  False
```

In [1854]:
```
feats_to_drop = [col.strip().lower() for col in feats_to_drop]
input_jeeps.columns = [col.strip().lower() for col in input_jeeps.columns]
input_jeeps.drop(columns=feats_to_drop,inplace=True)
input_jeeps.head()
```

Out[1854]:

listingid	sellercity	sellerlistsrc	sellerrating	sellerrevcnt	sellerstate	vehcertified	vehdrivetrain	
4777	17	Jeep Certified Program	4.8	1405	WI	1	_4_wd	C
6242	4	Inventory Command Center	4.4	21	MO	0	_4_wd	C
10882	2	Digital Motorworks (DMi)	3.0	51	NY	0	_4_wd	C
12013	3	Digital Motorworks (DMi)	3.5	17	IL	0	_4_wd	C
12334	25	Digital Motorworks (DMi)	4.6	240	IL	0	_4_wd	C

In [1855]:
```
input_caddys.columns = [col.strip().lower() for col in input_caddys.columns]
input_caddys.drop(columns=feats_to_drop,inplace=True)
input_caddys.head()
```

Out[1855]:

listingid	sellercity	sellerlistsrc	sellerrating	sellerrevcnt	sellerstate	vehcertified	vehdrivetrain	
7108	2	HomeNet Automotive	3.7	74	AR	0	fwd	C
21448	3	HomeNet Automotive	3.7	629	LA	0	fwd	C
21807	4	HomeNet Automotive	4.9	360	FL	1	fwd	C
30524	3	Digital Motorworks (DMi)	5.0	4	TX	0	fwd	C
34061	2	Digital Motorworks (DMi)	4.3	312	TX	0	fwd	C

In [1856]:
```
input_jeeps = pd.merge(input_jeeps,self_encodej,left_index=True,right_index=True)
print(self_encodej.columns)
input_jeeps.info()
```

```
Index(['black_x', 'blue_x', 'brown_x', 'green', 'metallic', 'pearlcoat',
       'clearcoat', 'granite', 'red_x', 'silver_x', 'white', 'none_x',
       'black_y', 'blue_y', 'brown_y', 'beige', 'trim', 'red_y', 'silver_y',
       'frost', 'maple', 'tan', 'cirrus', 'carbon', 'plum', 'none_y',
       'Accident(s) Reported', 'Buyback Protection Eligible',
       'Non-Personal Use Reported', 'Title Issue(s) Reported',
       'None of the above'],
      dtype='object')
<class 'pandas.core.frame.DataFrame'>
Index: 3420 entries, 4777 to 8620012
Data columns (total 47 columns):
 #   Column                         Non-Null Count  Dtype
---  ------                         --------------  -----
 0   sellercity                     3420 non-null   int64
 1   sellerlistsrc                  3420 non-null   object
 2   sellerrating                   3420 non-null   float64
 3   sellerrevcnt                   3420 non-null   int64
 4   sellerstate                    3420 non-null   object
 5   vehcertified                   3420 non-null   int32
 6   vehdrivetrain                  3420 non-null   object
 7   vehfuel                        3420 non-null   object
 8   vehlistdays                    3420 non-null   float64
 9   vehmake                        3420 non-null   object
 10  vehmileage                     3420 non-null   float64
 11  vehpricelabel                  3420 non-null   object
 12  vehyear                        3420 non-null   int64
 13  enginesize                     3420 non-null   float64
 14  cylinders                      3420 non-null   int64
 15  owners                         3420 non-null   object
 16  black_x                        3420 non-null   int32
 17  blue_x                         3420 non-null   int32
 18  brown_x                        3420 non-null   int32
 19  green                          3420 non-null   int32
 20  metallic                       3420 non-null   int32
 21  pearlcoat                      3420 non-null   int32
 22  clearcoat                      3420 non-null   int32
 23  granite                        3420 non-null   int32
 24  red_x                          3420 non-null   int32
 25  silver_x                       3420 non-null   int32
 26  white                          3420 non-null   int32
 27  none_x                         3420 non-null   int32
 28  black_y                        3420 non-null   int32
 29  blue_y                         3420 non-null   int32
 30  brown_y                        3420 non-null   int32
 31  beige                          3420 non-null   int32
 32  trim                           3420 non-null   int32
 33  red_y                          3420 non-null   int32
 34  silver_y                       3420 non-null   int32
 35  frost                          3420 non-null   int32
 36  maple                          3420 non-null   int32
 37  tan                            3420 non-null   int32
 38  cirrus                         3420 non-null   int32
 39  carbon                         3420 non-null   int32
 40  plum                           3420 non-null   int32
 41  none_y                         3420 non-null   int32
 42  Accident(s) Reported           3420 non-null   int64
 43  Buyback Protection Eligible    3420 non-null   int64
```

```
 44  Non-Personal Use Reported    3420 non-null   int64
 45  Title Issue(s) Reported      3420 non-null   int64
 46  None of the above            3420 non-null   int32
dtypes: float64(4), int32(28), int64(8), object(7)
memory usage: 1.0+ MB
```

In [1857]:
```
input_caddys = pd.merge(input_caddys,self_encodec,left_index=True,right_index=
True)
print(self_encodec.columns)
input_caddys.info()
```

```
Index(['black_x', 'blue_x', 'brown_x', 'green', 'metallic', 'pearlcoat',
       'clearcoat', 'granite', 'red_x', 'silver_x', 'white', 'none_x',
       'black_y', 'blue_y', 'brown_y', 'beige', 'trim', 'red_y', 'silver_y',
       'frost', 'maple', 'tan', 'cirrus', 'carbon', 'plum', 'none_y',
       'Accident(s) Reported', 'Buyback Protection Eligible',
       'Non-Personal Use Reported', 'Title Issue(s) Reported',
       'None of the above'],
      dtype='object')
<class 'pandas.core.frame.DataFrame'>
Index: 1570 entries, 7108 to 8617378
Data columns (total 47 columns):
 #   Column                         Non-Null Count  Dtype
---  ------                         --------------  -----
 0   sellercity                     1570 non-null   int64
 1   sellerlistsrc                  1570 non-null   object
 2   sellerrating                   1570 non-null   float64
 3   sellerrevcnt                   1570 non-null   int64
 4   sellerstate                    1570 non-null   object
 5   vehcertified                   1570 non-null   int32
 6   vehdrivetrain                  1570 non-null   object
 7   vehfuel                        1570 non-null   object
 8   vehlistdays                    1570 non-null   float64
 9   vehmake                        1570 non-null   object
 10  vehmileage                     1570 non-null   float64
 11  vehpricelabel                  1570 non-null   object
 12  vehyear                        1570 non-null   int64
 13  enginesize                     1570 non-null   float64
 14  cylinders                      1570 non-null   int64
 15  owners                         1570 non-null   object
 16  black_x                        1570 non-null   int32
 17  blue_x                         1570 non-null   int32
 18  brown_x                        1570 non-null   int32
 19  green                          1570 non-null   int32
 20  metallic                       1570 non-null   int32
 21  pearlcoat                      1570 non-null   int32
 22  clearcoat                      1570 non-null   int32
 23  granite                        1570 non-null   int32
 24  red_x                          1570 non-null   int32
 25  silver_x                       1570 non-null   int32
 26  white                          1570 non-null   int32
 27  none_x                         1570 non-null   int32
 28  black_y                        1570 non-null   int32
 29  blue_y                         1570 non-null   int32
 30  brown_y                        1570 non-null   int32
 31  beige                          1570 non-null   int32
 32  trim                           1570 non-null   int32
 33  red_y                          1570 non-null   int32
 34  silver_y                       1570 non-null   int32
 35  frost                          1570 non-null   int32
 36  maple                          1570 non-null   int32
 37  tan                            1570 non-null   int32
 38  cirrus                         1570 non-null   int32
 39  carbon                         1570 non-null   int32
 40  plum                           1570 non-null   int32
 41  none_y                         1570 non-null   int32
 42  Accident(s) Reported           1570 non-null   int64
 43  Buyback Protection Eligible    1570 non-null   int64
```

```
 44  Non-Personal Use Reported    1570 non-null   int64
 45  Title Issue(s) Reported      1570 non-null   int64
 46  None of the above            1570 non-null   int32
dtypes: float64(4), int32(28), int64(8), object(7)
memory usage: 449.3+ KB
```

In [1858]:
```
temp_encodedj = input_jeeps[encoded_cols]
print(encoded_cols)
print(temp_encodedj.columns)
```

```
['sellerlistsrc', 'sellerstate', 'vehdrivetrain', 'vehfuel', 'vehpricelabel']
Index(['sellerlistsrc', 'sellerstate', 'vehdrivetrain', 'vehfuel',
       'vehpricelabel'],
      dtype='object')
```

In [1859]:
```
encoderj = OneHotEncoder(handle_unknown='ignore')
coderj = encoderj.fit(temp_encodedj)
temp_encodedj.columns = temp_encodedj.columns.astype(str)
temp_encodedj = oHotEncode(temp_encodedj,coderj)
temp_encodedj.head()
```

Out[1859]:

listingid	Digital Motorworks (DMi)	HomeNet Automotive	Inventory Command Center	Jeep Certified Program	My Dealer Center	AL	AR	GA	IA	IL	...	_4_wc
4777	0.0	0.0	0.0	1.0	0.0	0.0	0.0	0.0	0.0	0.0	...	1.0
6242	0.0	0.0	1.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	1.0
10882	1.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	1.0
12013	1.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	1.0	...	1.0
12334	1.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	1.0	...	1.0

5 rows × 36 columns

In [1860]:
```
temp_encodedc = input_caddys[encoded_cols]
print(temp_encodedc.columns)
```

```
Index(['sellerlistsrc', 'sellerstate', 'vehdrivetrain', 'vehfuel',
       'vehpricelabel'],
      dtype='object')
```

In [1861]:
```
encoderc = OneHotEncoder(handle_unknown='ignore')
coderc = encoderc.fit(temp_encodedc)
temp_encodedc.columns = temp_encodedc.columns.astype(str)
temp_encodedc = oHotEncode(temp_encodedc,coderc)
temp_encodedc.head()
```

Out[1861]:

listingid	Digital Motorworks (DMi)	HomeNet Automotive	Inventory Command Center	My Dealer Center	AK	AL	AR	AZ	CA	CO	...	VT	WA	\
7108	0.0	1.0	0.0	0.0	0.0	0.0	1.0	0.0	0.0	0.0	...	0.0	0.0	0
21448	0.0	1.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0
21807	0.0	1.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0
30524	1.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0
34061	1.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0

5 rows × 58 columns

In [1862]:
```
input_jeeps.drop(columns=encoded_cols,inplace=True)
post_feat_engj = pd.merge(input_jeeps,temp_encodedj,left_index=True, right_ind
ex=True)
post_feat_engj.head()
```

Out[1862]:

listingid	sellercity	sellerrating	sellerrevcnt	vehcertified	vehlistdays	vehmake	vehmileage	vehy
4777	17	4.8	1405	1	29.0	Jeep	38957.0	
6242	4	4.4	21	0	60.0	Jeep	20404.0	
10882	2	3.0	51	0	31.0	Jeep	34649.0	
12013	3	3.5	17	0	195.0	Jeep	48814.0	
12334	25	4.6	240	0	29.0	Jeep	29095.0	

5 rows × 78 columns

In [1863]:
```
input_caddys.drop(columns=encoded_cols,inplace=True)
post_feat_engc = pd.merge(input_caddys,temp_encodedc,left_index=True, right_in
dex=True)
post_feat_engc.head()
```

Out[1863]:

listingid	sellercity	sellerrating	sellerrevcnt	vehcertified	vehlistdays	vehmake	vehmileage	vehy
7108	2	3.7	74	0	99.0	Cadillac	19788.0	
21448	3	3.7	629	0	6.0	Cadillac	21098.0	
21807	4	4.9	360	1	7.0	Cadillac	3547.0	
30524	3	5.0	4	0	12.0	Cadillac	12146.0	
34061	2	4.3	312	0	26.0	Cadillac	28293.0	

5 rows × 100 columns

In [1864]:
```
post_feat_engj = pd.merge(post_feat_engj,vocabjs,left_index=True, right_index=
True)
post_feat_engj.head()
```

Out[1864]:

listingid	sellercity	sellerrating	sellerrevcnt	vehcertified	vehlistdays	vehmake	vehmileage	vehy
4777	17	4.8	1405	1	29.0	Jeep	38957.0	
6242	4	4.4	21	0	60.0	Jeep	20404.0	
10882	2	3.0	51	0	31.0	Jeep	34649.0	
12013	3	3.5	17	0	195.0	Jeep	48814.0	
12334	25	4.6	240	0	29.0	Jeep	29095.0	

5 rows × 168 columns

In [1865]:
```
post_feat_engc = pd.merge(post_feat_engc,vocabcs,left_index=True, right_index=True)
post_feat_engc.head()
```

Out[1865]:

listingid	sellercity	sellerrating	sellerrevcnt	vehcertified	vehlistdays	vehmake	vehmileage	vehy
7108	2	3.7	74	0	99.0	Cadillac	19788.0	
21448	3	3.7	629	0	6.0	Cadillac	21098.0	
21807	4	4.9	360	1	7.0	Cadillac	3547.0	
30524	3	5.0	4	0	12.0	Cadillac	12146.0	
34061	2	4.3	312	0	26.0	Cadillac	28293.0	

5 rows × 190 columns

In [1866]:
```
types = post_feat_engj.select_dtypes(include=['object'])

# Display the object-type columns
print(types)
```

```
          vehmake owners
listingid
4777         Jeep      1
6242         Jeep      1
10882        Jeep      1
12013        Jeep      1
12334        Jeep      1
...           ...    ...
8610847      Jeep      1
8612731      Jeep      1
8614177      Jeep      1
8615510      Jeep      1
8620012      Jeep      1

[3420 rows x 2 columns]
```

In [1867]:
```
post_feat_engj["owners"] = pd.to_numeric(post_feat_engj["owners"], errors='coe
rce').fillna(0).astype(int)
print(post_feat_engj["owners"].value_counts())
post_feat_engj.columns = post_feat_engj.columns.astype(str)
post_feat_engj.info()
```

```
owners
1    3057
2     219
0     129
3      12
4       3
Name: count, dtype: int64
<class 'pandas.core.frame.DataFrame'>
Index: 3420 entries, 4777 to 8620012
Columns: 168 entries, sellercity to x3d
dtypes: float64(130), int32(29), int64(8), object(1)
memory usage: 4.2+ MB
```

In [1868]:
```
post_feat_engc["owners"] = pd.to_numeric(post_feat_engc["owners"], errors='coe
rce').fillna(0).astype(int)
print(post_feat_engc["owners"].value_counts())
post_feat_engc.columns = post_feat_engc.columns.astype(str)
post_feat_engc.info()
```

```
owners
1    1368
0     118
2      79
3       4
4       1
Name: count, dtype: int64
<class 'pandas.core.frame.DataFrame'>
Index: 1570 entries, 7108 to 8617378
Columns: 190 entries, sellercity to your
dtypes: float64(152), int32(29), int64(8), object(1)
memory usage: 2.1+ MB
```

In [1869]:
```
columnsj_missing = post_feat_engj.columns[post_feat_engj.isna().any()].tolist
()

# Display columns with missing values
print("Columns with missing values:", columnsj_missing)
```

```
Columns with missing values: []
```

In [1870]:
```
columnsc_missing = post_feat_engc.columns[post_feat_engc.isna().any()].tolist
()

# Display columns with missing values
print("Columns with missing values:", columnsc_missing)
```

```
Columns with missing values: []
```

In [1871]:
```
print(post_feat_engj.isna().sum().sum())
post_feat_engj.head()
```

```
0
```

Out[1871]:

listingid	sellercity	sellerrating	sellerrevcnt	vehcertified	vehlistdays	vehmake	vehmileage	vehy
4777	17	4.8	1405	1	29.0	Jeep	38957.0	
6242	4	4.4	21	0	60.0	Jeep	20404.0	
10882	2	3.0	51	0	31.0	Jeep	34649.0	
12013	3	3.5	17	0	195.0	Jeep	48814.0	
12334	25	4.6	240	0	29.0	Jeep	29095.0	

5 rows × 168 columns

In [1872]:
```
print(post_feat_engc.isna().sum().sum())
post_feat_engc.head()
```

```
0
```

Out[1872]:

listingid	sellercity	sellerrating	sellerrevcnt	vehcertified	vehlistdays	vehmake	vehmileage	vehy
7108	2	3.7	74	0	99.0	Cadillac	19788.0	
21448	3	3.7	629	0	6.0	Cadillac	21098.0	
21807	4	4.9	360	1	7.0	Cadillac	3547.0	
30524	3	5.0	4	0	12.0	Cadillac	12146.0	
34061	2	4.3	312	0	26.0	Cadillac	28293.0	

5 rows × 190 columns

In [1873]:
```
output_data = pd.DataFrame(df_train.iloc[:,-2:].copy())
output_jeeps = output_data[df_train["vehmake"] == "Jeep"].copy()
output_caddys = output_data[df_train["vehmake"] == "Cadillac"].copy()
print(output_jeeps["vehicle_trim"].value_counts())
print(output_caddys["vehicle_trim"].value_counts())
```

```
vehicle_trim
Limited             1621
Laredo               658
Overland             342
Altitude             331
Summit               203
Trailhawk            149
SRT                   64
Trackhawk             27
Sterling Edition      25
Name: count, dtype: int64
vehicle_trim
Premium Luxury    702
Luxury            603
Base              137
Platinum          128
Name: count, dtype: int64
```

In [1874]:
```
df_test.isna().sum()
test_df = df_test.copy()
test_jeeps = pd.DataFrame(test_df[test_df["VehMake"]=="Jeep"])
test_caddys = pd.DataFrame(test_df[test_df["VehMake"]=="Cadillac"])
```

In [1875]:
```
#NOW APPLY THE SAME ENCODING AND TRANSFORMATIONS TO THE TEST DATASET
test_data_jeeps = engineerTestData(test_jeeps,log_cols,encoded_cols,freq_cols,
                                   mask_cols,tokenize_cols,orig_cols,feats_to_drop,
                                   coderj,tf_featsj,tf_revj)
```

```
COLOR
COLOR2
HISTORY

C:\Users\aflyn\AppData\Local\Temp\ipykernel_19256\3378287000.py:22: SettingWi
thCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/s
table/user_guide/indexing.html#returning-a-view-versus-a-copy
  test_encode[col] = temp_df
```

In [1876]:
```
#NOW APPLY THE SAME ENCODING AND TRANSFORMATIONS TO THE TEST DATASET
test_data_caddys = engineerTestData(test_caddys,log_cols,encoded_cols,freq_col
s,
                                    mask_cols,tokenize_cols,orig_cols,feats_to_drop,
                                    coderc,tf_featsc,tf_revc)
```

```
COLOR

C:\Users\aflyn\AppData\Local\Temp\ipykernel_19256\3378287000.py:22: SettingWi
thCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/s
table/user_guide/indexing.html#returning-a-view-versus-a-copy
  test_encode[col] = temp_df

COLOR2
HISTORY
```

In [1877]:
```
print(test_data_jeeps.isna().sum().sum())
print(test_data_jeeps.info())
test_data_jeeps.columns
```

```
21
<class 'pandas.core.frame.DataFrame'>
Index: 658 entries, 8625693 to 9997199
Columns: 168 entries, sellercity to x3d
dtypes: float64(130), int32(28), int64(8), object(2)
memory usage: 813.0+ KB
None
```

Out[1877]:
```
Index(['sellercity', 'sellerrating', 'sellerrevcnt', 'vehcertified',
       'vehlistdays', 'vehmake', 'vehmileage', 'vehyear', 'enginesize',
       'cylinders',
       ...
       'up', 'us', 'v6', 'vehicles', 'warranty', 'wheel', 'wheels_y', 'will',
       'x27', 'x3d'],
      dtype='object', length=168)
```

In [1878]:
```
print(test_data_caddys.isna().sum().sum())
print(test_data_caddys.info())
test_data_caddys.head()
```

```
6
<class 'pandas.core.frame.DataFrame'>
Index: 342 entries, 8622015 to 9999562
Columns: 190 entries, sellercity to your
dtypes: float64(152), int32(28), int64(8), object(2)
memory usage: 481.0+ KB
None
```

Out[1878]:

listingid	sellercity	sellerrating	sellerrevcnt	vehcertified	vehlistdays	vehmake	vehmileage	vehy
8622015	1	2.5	59	0	4.969813	Cadillac	13625.0	2
8627430	7	4.6	162	0	3.218876	Cadillac	22269.0	2
8630863	1	4.7	55	0	3.871201	Cadillac	31828.0	2
8636188	5	4.8	59	1	3.091042	Cadillac	25500.0	2
8639022	1	2.8	18	0	5.099866	Cadillac	21088.0	2

5 rows × 190 columns

In [1879]:
```
columns_with_missing_values = test_data_jeeps.columns[test_data_jeeps.isna().a
ny()].tolist()
print(test_data_jeeps.index)
# Display columns with missing values
print("Columns with missing values:", columns_with_missing_values)
```

```
Index([8625693, 8625750, 8626885, 8629427, 8629962, 8630638, 8630762, 863228
3,
       8649823, 8649882,
       ...
       9983383, 9983736, 9986645, 9986711, 9987122, 9987266, 9989599, 999217
0,
       9992442, 9997199],
      dtype='int64', name='listingid', length=658)
Columns with missing values: ['owners']
```

In [1880]:
```
columns_with_missing_values = test_data_caddys.columns[test_data_caddys.isna
().any()].tolist()
print(test_data_caddys.index)
# Display columns with missing values
print("Columns with missing values:", columns_with_missing_values)
```

```
Index([8622015, 8627430, 8630863, 8636188, 8639022, 8639992, 8642278, 864500
5,
       8647894, 8649794,
       ...
       9948073, 9953049, 9957979, 9976374, 9981584, 9987374, 9991651, 999356
2,
       9994646, 9999562],
      dtype='int64', name='listingid', length=342)
Columns with missing values: ['owners']
```

In [1881]:
```
test_data_jeeps["owners"] = pd.to_numeric(test_data_jeeps["owners"], errors='c
oerce').fillna(0).astype(int)
test_data_caddys["owners"] = pd.to_numeric(test_data_caddys["owners"], errors
='coerce').fillna(0).astype(int)
```

In [1882]:
```
print(test_data_jeeps.shape)
print(test_data_caddys.shape)
#test_data_jeeps = test_data_jeeps[post_feat_engj.columns]

post_feat_engj.to_csv("postfeatengj.csv")
test_data_jeeps.to_csv("testtransj.csv")
columns_unique_to_df1 = set(post_feat_engj.columns) - set(test_data_jeeps.colu
mns)
columns_unique_to_df2 = set(test_data_jeeps.columns) - set(post_feat_engj.colu
mns)
common_columns = post_feat_engj.columns.intersection(test_data_jeeps.columns)

print("Columns unique to DataFrame 1:", columns_unique_to_df1)
print("Columns unique to DataFrame 2:", columns_unique_to_df2)
print("Common columns:", common_columns)
```

```
(658, 168)
(342, 190)
Columns unique to DataFrame 1: set()
Columns unique to DataFrame 2: set()
Common columns: Index(['sellercity', 'sellerrating', 'sellerrevcnt', 'vehcert
ified',
       'vehlistdays', 'vehmake', 'vehmileage', 'vehyear', 'enginesize',
       'cylinders',
       ...
       'up', 'us', 'v6', 'vehicles', 'warranty', 'wheel', 'wheels_y', 'will',
       'x27', 'x3d'],
      dtype='object', length=168)
```

In [1883]:
```
jeep_encoder = LabelEncoder()
caddy_encoder = LabelEncoder()

pre_encoded_jeeps = output_jeeps["vehicle_trim"]
pre_encoded_caddys = output_caddys["vehicle_trim"]

print(pre_encoded_jeeps.value_counts())
jeep_veh_trim = pd.Series(jeep_encoder.fit_transform(pre_encoded_jeeps),
                      index=post_feat_engj.index,
                      name=pre_encoded_jeeps.name)
caddy_veh_trim = pd.Series(caddy_encoder.fit_transform(pre_encoded_caddys),
                      index=pre_encoded_caddys.index,
                      name=pre_encoded_caddys.name)
print(np.unique(jeep_encoder.inverse_transform(jeep_veh_trim),return_counts=Tr
ue))

list_pricej = output_jeeps["dealer_listing_price"]
list_pricec = output_caddys["dealer_listing_price"]
```

```
vehicle_trim
Limited             1621
Laredo               658
Overland             342
Altitude             331
Summit               203
Trailhawk            149
SRT                   64
Trackhawk             27
Sterling Edition      25
Name: count, dtype: int64
(array(['Altitude', 'Laredo', 'Limited', 'Overland', 'SRT',
       'Sterling Edition', 'Summit', 'Trackhawk', 'Trailhawk'],
      dtype=object), array([ 331,  658, 1621,  342,   64,   25,  203,   27,
149], dtype=int64))
```

In [1884]:
```
post_feat_engj.drop("vehmake", axis=1, inplace=True)
print(jeep_veh_trim.value_counts())

post_feat_engc.drop("vehmake", axis=1, inplace=True)
print(caddy_veh_trim.value_counts())
```

```
vehicle_trim
2    1621
1     658
3     342
0     331
6     203
8     149
4      64
7      27
5      25
Name: count, dtype: int64
vehicle_trim
3    702
1    603
0    137
2    128
Name: count, dtype: int64
```

In [1885]:
```
clfj = tr.XGB_Classifier(post_feat_engj,jeep_veh_trim,False,jeep_encoder)
clfc = tr.XGB_Classifier(post_feat_engc,caddy_veh_trim,False,caddy_encoder)
```

```
Best Parameters:  {'learning_rate': 0.05, 'max_depth': 5, 'n_estimators': 20
0}
Best Score:  0.9837414357642424
Best Parameters:  {'learning_rate': 0.05, 'max_depth': 7, 'n_estimators': 20
0}
Best Score:  0.9965302977883251
```

In [1886]:
```
regj = tr.XGB_Regressor(post_feat_engj,list_pricej,jeep_veh_trim,jeep_encoder,
False,True)
regc = tr.XGB_Regressor(post_feat_engc,list_pricec,caddy_veh_trim,caddy_encode
r,False,True)
```

```
DUMMIES:             Altitude  Laredo  Limited  Overland    SRT  Sterling Edi
tion  \
listingid
4777          False    True    False     False  False            False
6242          False   False     True     False  False            False
10882         False   False     True     False  False            False
12013         False    True    False     False  False            False
12334         False   False     True     False  False            False
...             ...     ...      ...       ...    ...              ...
8610847       False   False     True     False  False            False
8612731        True   False    False     False  False            False
8614177       False   False     True     False  False            False
8615510       False   False     True     False  False            False
8620012       False   False    False     False  False            False

           Summit  Trackhawk  Trailhawk
listingid
4777        False      False      False
6242        False      False      False
10882       False      False      False
12013       False      False      False
12334       False      False      False
...           ...        ...        ...
8610847     False      False      False
8612731     False      False      False
8614177     False      False      False
8615510     False      False      False
8620012     False      False       True

[3420 rows x 9 columns]
SECOND :             sellercity  sellerrating  sellerrevcnt  vehcertified  ve
hlistdays  \
listingid
4777                 17           4.8          1405             1         29.0
6242                  4           4.4            21             0         60.0
10882                 2           3.0            51             0         31.0
12013                 3           3.5            17             0        195.0
12334                25           4.6           240             0         29.0
...                 ...           ...           ...           ...          ...
8610847              18           4.8          1016             0        103.0
8612731               5           4.9           121             1         39.0
8614177              35           1.5             6             0         40.0
8615510              27           3.3            16             0          5.0
8620012              21           3.8           727             0         21.0

           vehmileage  vehyear  enginesize  cylinders  owners  ...        up
\
listingid                                                       ...
4777          38957.0        9         3.6          6       1  ...  0.171019
6242          20404.0        6         3.6          6       1  ...  0.000000
10882         34649.0        6         3.6          6       1  ...  0.000000
12013         48814.0        7         3.6          6       1  ...  0.125957
12334         29095.0        6         3.6          6       1  ...  0.287750
...               ...      ...         ...        ...     ...  ...       ...
8610847        1644.0        6         3.6          6       1  ...  0.058345
8612731       31369.0        9         3.6          6       1  ...  0.000000
8614177       35773.0        9         3.6          6       1  ...  0.000000
```

```
8615510       20039.0        9         3.0          6       1  ...  0.000000
8620012       17806.0        6         3.6          6       1  ...  0.213997

                us        v6  vehicles  warranty     wheel  wheels_y  \
listingid
4777       0.308908  0.000000  0.000000  0.165515  0.000000  0.000000
6242       0.343764  0.000000  0.000000  0.000000  0.000000  0.000000
10882      0.000000  0.150977  0.000000  0.166317  0.000000  0.166582
12013      0.000000  0.000000  0.000000  0.121903  0.111063  0.122097
12334      0.000000  0.000000  0.000000  0.139244  0.000000  0.000000
...             ...       ...       ...       ...       ...       ...
8610847    0.105387  0.102517  0.053794  0.000000  0.051446  0.056557
8612731    0.060796  0.059140  0.000000  0.325748  0.000000  0.065253
8614177    0.000000  0.000000  0.231065  0.242547  0.000000  0.000000
8615510    0.000000  0.000000  0.239290  0.251180  0.000000  0.000000
8620012    0.000000  0.000000  0.197305  0.000000  0.000000  0.000000

              will       x27  x3d
listingid
4777       0.000000  0.156198  0.0
6242       0.777900  0.347645  0.0
10882      0.175603  0.000000  0.0
12013      0.000000  0.230082  0.0
12334      0.000000  0.262812  0.0
...             ...       ...  ...
8610847    0.059620  0.213154  0.0
8612731    0.000000  0.061482  0.0
8614177    0.256090  0.000000  0.0
8615510    0.265205  0.000000  0.0
8620012    0.437347  0.000000  0.0

[3420 rows x 167 columns]
           sellercity  sellerrating  sellerrevcnt  vehcertified  vehlistdays
\
listingid
4777               17           4.8          1405             1         29.0
6242                4           4.4            21             0         60.0
10882               2           3.0            51             0         31.0
12013               3           3.5            17             0        195.0
12334              25           4.6           240             0         29.0
...               ...           ...           ...           ...          ...
8610847            18           4.8          1016             0        103.0
8612731             5           4.9           121             1         39.0
8614177            35           1.5             6             0         40.0
8615510            27           3.3            16             0          5.0
8620012            21           3.8           727             0         21.0

           vehmileage  vehyear  enginesize  cylinders  owners  ...  x3d  \
listingid                                                      ...
4777          38957.0        9         3.6          6       1  ...  0.0
6242          20404.0        6         3.6          6       1  ...  0.0
10882         34649.0        6         3.6          6       1  ...  0.0
12013         48814.0        7         3.6          6       1  ...  0.0
12334         29095.0        6         3.6          6       1  ...  0.0
...               ...      ...         ...        ...     ...  ...  ...
8610847        1644.0        6         3.6          6       1  ...  0.0
8612731       31369.0        9         3.6          6       1  ...  0.0
```

```
8614177       35773.0        9         3.6          6       1  ...  0.0
8615510       20039.0        9         3.0          6       1  ...  0.0
8620012       17806.0        6         3.6          6       1  ...  0.0

           Altitude  Laredo  Limited  Overland    SRT  Sterling Edition  \
listingid
4777          False    True    False     False  False             False
6242          False   False     True     False  False             False
10882         False   False     True     False  False             False
12013         False    True    False     False  False             False
12334         False   False     True     False  False             False
...             ...     ...      ...       ...    ...               ...
8610847       False   False     True     False  False             False
8612731        True   False    False     False  False             False
8614177       False   False     True     False  False             False
8615510       False   False     True     False  False             False
8620012       False   False    False     False  False             False

           Summit  Trackhawk  Trailhawk
listingid
4777        False      False      False
6242        False      False      False
10882       False      False      False
12013       False      False      False
12334       False      False      False
...           ...        ...        ...
8610847     False      False      False
8612731     False      False      False
8614177     False      False      False
8615510     False      False      False
8620012     False      False       True

[3420 rows x 176 columns]
Best Regression Parameters:  {'learning_rate': 0.1, 'max_depth': 5, 'n_estima
tors': 200}
Best Regression Score R2:  0.9547829303419022
DUMMIES:              Base  Luxury  Platinum  Premium Luxury
listingid
7108       False    True     False           False
21448      False   False     False            True
21807      False    True     False           False
30524       True   False     False           False
34061      False   False     False            True
...          ...     ...       ...             ...
8599564    False   False     False            True
8601212    False    True     False           False
8604205    False   False     False            True
8616294    False    True     False           False
8617378    False    True     False           False

[1570 rows x 4 columns]
SECOND :             sellercity  sellerrating  sellerrevcnt  vehcertified  ve
hlistdays  \
listingid
7108                2           3.7           74             0        99.0
21448               3           3.7          629             0         6.0
21807               4           4.9          360             1         7.0
```

```
30524               3          5.0            4             0        12.0
34061               2          4.3          312             0        26.0
...               ...          ...          ...           ...         ...
8599564             4          4.4           15             1        35.0
8601212             2          4.8          461             0        19.0
8604205             6          4.7           58             1        20.0
8616294             3          4.1           20             1       185.0
8617378             1          4.9          278             0        74.0

           vehmileage  vehyear  enginesize  cylinders  owners  ...      this
\
listingid                                                       ...
7108          19788.0        6         3.6          6       1  ...  0.098747
21448         21098.0        6         3.6          6       1  ...  0.000000
21807          3547.0        5         3.6          6       1  ...  0.000000
30524         12146.0        7         3.6          6       1  ...  0.334768
34061         28293.0        6         3.6          6       1  ...  0.182420
...               ...      ...         ...        ...     ...  ...       ...
8599564       41886.0        6         3.6          6       1  ...  0.024555
8601212       39882.0        7         3.6          6       1  ...  0.356044
8604205       17314.0        6         3.6          6       1  ...  0.000000
8616294       16278.0        6         3.6          6       0  ...  0.000000
8617378       38146.0        7         3.6          6       2  ...  0.000000

                 us        v6  vehicles  warranty     wheel      will  \
listingid
7108       0.231728  0.000000  0.101986  0.000000  0.110024  0.119530
21448      0.000000  0.000000  0.135329  0.000000  0.000000  0.317216
21807      0.177843  0.000000  0.000000  0.517238  0.000000  0.183471
30524      0.000000  0.000000  0.172875  0.190402  0.000000  0.202614
34061      0.107021  0.000000  0.188405  0.000000  0.101626  0.000000
...             ...       ...       ...       ...       ...       ...
8599564    0.000000  0.024942  0.000000  0.000000  0.054718  0.000000
8601212    0.000000  0.000000  0.000000  0.000000  0.000000  0.215491
8604205    0.000000  0.000000  0.182802  0.000000  0.000000  0.000000
8616294    0.000000  0.140945  0.000000  0.473517  0.000000  0.167962
8617378    0.000000  0.164172  0.166927  0.000000  0.000000  0.000000

                x27  x3d      your
listingid
7108       0.216169  0.0  0.308417
21448      0.000000  0.0  0.545664
21807      0.000000  0.0  0.000000
30524      0.000000  0.0  0.000000
34061      0.299505  0.0  0.000000
...             ...  ...       ...
8599564    0.000000  0.0  0.000000
8601212    0.194856  0.0  0.185340
8604205    0.000000  0.0  0.184271
8616294    0.151879  0.0  0.433384
8617378    0.000000  0.0  0.168268

[1570 rows x 189 columns]
           sellercity  sellerrating  sellerrevcnt  vehcertified  vehlistdays
\
listingid
7108                2           3.7            74             0        99.0
```

```
21448              3          3.7          629             0         6.0
21807              4          4.9          360             1         7.0
30524              3          5.0            4             0        12.0
34061              2          4.3          312             0        26.0
...              ...          ...          ...           ...         ...
8599564            4          4.4           15             1        35.0
8601212            2          4.8          461             0        19.0
8604205            6          4.7           58             1        20.0
8616294            3          4.1           20             1       185.0
8617378            1          4.9          278             0        74.0

           vehmileage  vehyear  enginesize  cylinders  owners  ...  warranty
\
listingid                                                       ...
7108          19788.0        6         3.6          6       1  ...  0.000000
21448         21098.0        6         3.6          6       1  ...  0.000000
21807          3547.0        5         3.6          6       1  ...  0.517238
30524         12146.0        7         3.6          6       1  ...  0.190402
34061         28293.0        6         3.6          6       1  ...  0.000000
...               ...      ...         ...        ...     ...  ...       ...
8599564       41886.0        6         3.6          6       1  ...  0.000000
8601212       39882.0        7         3.6          6       1  ...  0.000000
8604205       17314.0        6         3.6          6       1  ...  0.000000
8616294       16278.0        6         3.6          6       0  ...  0.473517
8617378       38146.0        7         3.6          6       2  ...  0.000000

              wheel      will       x27  x3d      your   Base  Luxury  \
listingid
7108       0.110024  0.119530  0.216169  0.0  0.308417  False    True
21448      0.000000  0.317216  0.000000  0.0  0.545664  False   False
21807      0.000000  0.183471  0.000000  0.0  0.000000  False    True
30524      0.000000  0.202614  0.000000  0.0  0.000000   True   False
34061      0.101626  0.000000  0.299505  0.0  0.000000  False   False
...             ...       ...       ...  ...       ...    ...     ...
8599564    0.054718  0.000000  0.000000  0.0  0.000000  False   False
8601212    0.000000  0.215491  0.194856  0.0  0.185340  False    True
8604205    0.000000  0.000000  0.000000  0.0  0.184271  False   False
8616294    0.000000  0.167962  0.151879  0.0  0.433384  False    True
8617378    0.000000  0.000000  0.000000  0.0  0.168268  False    True

           Platinum  Premium Luxury
listingid
7108          False           False
21448         False            True
21807         False           False
30524         False           False
34061         False            True
...             ...             ...
8599564       False            True
8601212       False           False
8604205       False            True
8616294       False           False
8617378       False           False

[1570 rows x 193 columns]
Best Regression Parameters:  {'learning_rate': 0.1, 'max_depth': 3, 'n_estima
```

```
tors': 200}
Best Regression Score R2:  0.8658582954494091
```

In [1887]:
```python
#TEST VEHICLE TRIM PREDICTIONS
test_data_jeeps.drop("vehmake",axis=1,inplace=True)
clfj.prediction(test_data_jeeps)
print(clfj.preds.isna().sum())
print(clfj.preds.value_counts())

test_data_caddys.drop("vehmake",axis=1,inplace=True)
clfc.prediction(test_data_caddys)
print(clfc.preds.isna().sum())
print(clfc.preds.value_counts())
```

```
0
vehicle_trim
Limited      302
Laredo       134
Trailhawk    107
Overland      50
Summit        29
Altitude      16
SRT           12
Trackhawk      8
Name: count, dtype: int64
0
vehicle_trim
Luxury            157
Premium Luxury    139
Base               28
Platinum           18
Name: count, dtype: int64
```

In [1888]:
```
percentages = pd.DataFrame({"TEST PREDS" : clfj.preds.value_counts(normalize=True),
                            "TRAINING LABELS" : pre_encoded_jeeps.value_counts(normalize=True),
                           "Delta": clfj.preds.value_counts(normalize=True)-pre_encoded_jeeps.value_counts(normalize=True)})
percentages
```

Out[1888]:

vehicle_trim	TEST PREDS	TRAINING LABELS	Delta
Altitude	0.024316	0.096784	-0.072468
Laredo	0.203647	0.192398	0.011250
Limited	0.458967	0.473977	-0.015010
Overland	0.075988	0.100000	-0.024012
SRT	0.018237	0.018713	-0.000476
Sterling Edition	NaN	0.007310	NaN
Summit	0.044073	0.059357	-0.015284
Trackhawk	0.012158	0.007895	0.004263
Trailhawk	0.162614	0.043567	0.119047

In [1889]:
```
exp_pricesj = pr.calc_exp_prices(test_data_jeeps,pre_encoded_jeeps,regj)
exp_pricesc =  pr.calc_exp_prices(test_data_caddys,pre_encoded_caddys,regc)
print(exp_pricesj)
```

```
               Altitude        Laredo       Limited      Overland  \
listingid
8625693    26305.958984  22511.492188  24416.382812  26650.417969
8625750    26110.199219  22075.974609  24407.529297  26284.402344
8626885    26965.820312  22791.937500  24911.263672  26866.148438
8629427    28336.867188  23484.220703  27234.685547  29382.015625
8629962    27863.626953  23005.833984  25557.113281  27602.482422
...                 ...           ...           ...           ...
9987266    27125.679688  23125.240234  25254.228516  27236.339844
9989599    49921.476562  49098.902344  47680.218750  50203.109375
9992170    27023.886719  22395.115234  25338.798828  27421.513672
9992442    28509.769531  24144.958984  26720.724609  28895.845703
9997199    39859.289062  35374.953125  38411.335938  39344.082031

                    SRT  Sterling Edition        Summit     Trackhawk  \
listingid
8625693    26612.169922      25998.539062  28463.914062  27542.953125
8625750    26214.865234      24958.871094  27939.746094  27420.683594
8626885    27067.583984      25770.371094  28093.382812  28224.376953
8629427    28391.119141      27880.527344  30893.474609  32368.236328
8629962    27408.531250      27165.111328  29683.357422  28676.951172
...                 ...               ...           ...           ...
9987266    26953.228516      26768.818359  29129.783203  28297.748047
9989599    50043.984375      49599.730469  50301.023438  53258.480469
9992170    27183.914062      26179.144531  28368.599609  27642.472656
9992442    28792.849609      27712.750000  30112.353516  29694.699219
9997199    39152.003906      39064.460938  42755.484375  48953.597656

              Trailhawk
listingid
8625693    26137.488281
8625750    25783.185547
8626885    26425.335938
8629427    28134.537109
8629962    27111.201172
...                 ...
9987266    26697.958984
9989599    49541.722656
9992170    26748.267578
9992442    28341.298828
9997199    38963.285156

[658 rows x 9 columns]
```

In [1890]:
```
test_preds_pricej = pr.calc_test_prices(test_data_jeeps,clfj,regj,exp_pricesj)
test_preds_pricec = pr.calc_test_prices(test_data_caddys,clfc,regc,exp_prices
c)
```

In [1891]:
```
#JEEPS WITHOUT POSTERIOR PROBABILITY
model_w_trims_preds = regj.prediction(pd.concat([test_data_jeeps,clfj.preds_pr
oba],axis=1))
price_predsj_wtrims = pd.Series(model_w_trims_preds,index=test_data_jeeps.inde
x,name=list_pricej.name)
#CADDYS WITHOUT POSTERIOR PROBABILITY
model_w_trims_predsc = regc.prediction(pd.concat([test_data_caddys,clfc.preds_
proba],axis=1))
price_predsc_wtrims = pd.Series(model_w_trims_predsc,index=test_data_caddys.in
dex,name=list_pricec.name)
```

In [1892]:
```
#Train a new model without the trims involved
regj_no_trim = tr.XGB_Regressor(post_feat_engj,list_pricej,jeep_veh_trim,jeep_
encoder,False,False)
regj_no_trim.prediction(test_data_jeeps)
no_trims = pd.Series(regj_no_trim.preds,index=test_data_jeeps.index,name=list_
pricej.name)
```

listingid	sellercity	sellerrating	sellerrevcnt	vehcertified	vehlistdays
4777	17	4.8	1405	1	29.0
6242	4	4.4	21	0	60.0
10882	2	3.0	51	0	31.0
12013	3	3.5	17	0	195.0
12334	25	4.6	240	0	29.0
...	...	...	...	...	...
8610847	18	4.8	1016	0	103.0
8612731	5	4.9	121	1	39.0
8614177	35	1.5	6	0	40.0
8615510	27	3.3	16	0	5.0
8620012	21	3.8	727	0	21.0

listingid	vehmileage	vehyear	enginesize	cylinders	owners	...	up
4777	38957.0	9	3.6	6	1	...	0.171019
6242	20404.0	6	3.6	6	1	...	0.000000
10882	34649.0	6	3.6	6	1	...	0.000000
12013	48814.0	7	3.6	6	1	...	0.125957
12334	29095.0	6	3.6	6	1	...	0.287750
...	...	...	...	...	...	...	...
8610847	1644.0	6	3.6	6	1	...	0.058345
8612731	31369.0	9	3.6	6	1	...	0.000000
8614177	35773.0	9	3.6	6	1	...	0.000000
8615510	20039.0	9	3.0	6	1	...	0.000000
8620012	17806.0	6	3.6	6	1	...	0.213997

listingid	us	v6	vehicles	warranty	wheel	wheels_y
4777	0.308908	0.000000	0.000000	0.165515	0.000000	0.000000
6242	0.343764	0.000000	0.000000	0.000000	0.000000	0.000000
10882	0.000000	0.150977	0.000000	0.166317	0.000000	0.166582
12013	0.000000	0.000000	0.000000	0.121903	0.111063	0.122097
12334	0.000000	0.000000	0.000000	0.139244	0.000000	0.000000
...	...	...	...	...	...	...
8610847	0.105387	0.102517	0.053794	0.000000	0.051446	0.056557
8612731	0.060796	0.059140	0.000000	0.325748	0.000000	0.065253
8614177	0.000000	0.000000	0.231065	0.242547	0.000000	0.000000
8615510	0.000000	0.000000	0.239290	0.251180	0.000000	0.000000
8620012	0.000000	0.000000	0.197305	0.000000	0.000000	0.000000

listingid	will	x27	x3d
4777	0.000000	0.156198	0.0
6242	0.777900	0.347645	0.0
10882	0.175603	0.000000	0.0
12013	0.000000	0.230082	0.0
12334	0.000000	0.262812	0.0
...	...	...	...
8610847	0.059620	0.213154	0.0
8612731	0.000000	0.061482	0.0
8614177	0.256090	0.000000	0.0
8615510	0.265205	0.000000	0.0
8620012	0.437347	0.000000	0.0

```
[3420 rows x 167 columns]
Best Regression Parameters:  {'learning_rate': 0.1, 'max_depth': 5, 'n_estima
tors': 200}
Best Regression Score R2:  0.9122842851604821
```

In [1893]:
```
desc_stats = pd.DataFrame({'Training Prices': list_pricej.describe(),
                           'Expected Price Prob Calculation': test_preds_price
j.describe(),
                          'Model Prediction with trim probs': price_predsj_wtr
ims.describe(),
                          'Trim Agnostic Preds': no_trims.describe()})

desc_stats
```

Out[1893]:

	Training Prices	Expected Price Prob Calculation	Model Prediction with trim probs	Trim Agnostic Preds
count	3420.000000	658.000000	658.000000	658.000000
mean	30113.606725	28108.866128	29229.689453	27678.312500
std	7525.329608	5512.387086	4381.339355	5402.897461
min	18289.000000	20857.853516	24607.675781	19886.496094
25%	25500.000000	25208.970703	27100.701172	24978.932129
50%	28700.000000	27061.351562	28293.933594	26606.495117
75%	32992.000000	29378.322754	29926.392578	28815.147949
max	89500.000000	63451.113281	54994.359375	61349.957031

In [1894]:
```
#Test Exp price calc on training data
```

In [1895]:
```
'''import optuna

# Define the objective function for Optuna
def objective_trim(trial,X_train,y_train):
    # Sample hyperparameters from the search space
    learning_rate = trial.suggest_loguniform('learning_rate', 1e-4, 1e-2)
    num_units = trial.suggest_int('num_units', 64, 256)
    dropout_rate = trial.suggest_uniform('dropout_rate', 0.1, 0.5)
    num_layers = trial.suggest_int('num_layers', 1, 3)
    batch_size = trial.suggest_int('batch_size', 32, 128)
    hidden_activation = trial.suggest_categorical('hidden_activation', ['rel
u', 'tanh', 'sigmoid'])
    output_activation = trial.suggest_categorical('output_activation', ['softm
ax'])

    # Pack hyperparameters into a list
    params = [learning_rate, num_units, dropout_rate, num_layers, batch_size,
hidden_activation, output_activation]

    # Call the objective function with the sampled hyperparameters
    return nm.objective_function(params, X_train, y_train, 'trim')

# Run Optuna optimization
study_trim = optuna.create_study(direction='minimize')
study_trim.optimize(lambda trial: objective_trim(trial, post_feat_engj, jeep_v
eh_trim), n_trials=50,n_jobs=-1)  # You can adjust the number of trials

# Get the best parameters from the optimization
best_params_trim = study_trim.best_params'''
```

Out[1895]:
```
"import optuna\n\n# Define the objective function for Optuna\ndef objective_t
rim(trial,X_train,y_train):\n    # Sample hyperparameters from the search spa
ce\n    learning_rate = trial.suggest_loguniform('learning_rate', 1e-4, 1e-2)
\n    num_units = trial.suggest_int('num_units', 64, 256)\n    dropout_rate =
trial.suggest_uniform('dropout_rate', 0.1, 0.5)\n    num_layers = trial.sugge
st_int('num_layers', 1, 3)\n    batch_size = trial.suggest_int('batch_size',
32, 128)\n    hidden_activation = trial.suggest_categorical('hidden_activatio
n', ['relu', 'tanh', 'sigmoid'])\n    output_activation = trial.suggest_categ
orical('output_activation', ['softmax'])\n\n    # Pack hyperparameters into a
list\n    params = [learning_rate, num_units, dropout_rate, num_layers, batch
_size, hidden_activation, output_activation]\n\n    # Call the objective func
tion with the sampled hyperparameters\n    return nm.objective_function(param
s, X_train, y_train, 'trim')\n\n# Run Optuna optimization\nstudy_trim = optun
a.create_study(direction='minimize')\nstudy_trim.optimize(lambda trial: objec
tive_trim(trial, post_feat_engj, jeep_veh_trim), n_trials=50,n_jobs=-1)  # Yo
u can adjust the number of trials\n\n# Get the best parameters from the optim
ization\nbest_params_trim = study_trim.best_params"
```

In [1896]:
```
'''best_score = study_trim.best_value

# Print the best score
print(f"Best Score: {best_score}")'''
```

Out[1896]:
```
'best_score = study_trim.best_value\n\n# Print the best score\nprint(f"Best S
core: {best_score}")'
```

In [1897]:
```
# Concatenating predictions for Jeep
final_jeep_outputs = pd.concat([clfj.preds, test_preds_pricej], axis=1)

# Concatenating predictions for Caddy
final_caddy_outputs = pd.concat([clfc.preds, test_preds_pricec], axis=1)

# Concatenating final outputs and preserving the index order of test_df
final_outputs = pd.concat([final_jeep_outputs, final_caddy_outputs], axis=0)
final_outputs = final_outputs.sort_index()

# Naming columns appropriately
final_outputs.columns = output_data.columns

print(final_outputs)
```

```
             vehicle_trim  dealer_listing_price
listingid
8622015    Premium Luxury          36669.597656
8625693           Limited          24815.503906
8625750            Laredo          23464.814453
8626885           Limited          24937.939453
8627430    Premium Luxury          39133.730469
...                   ...                   ...
9992442           Limited          27154.781250
9993562    Premium Luxury          37285.820312
9994646    Premium Luxury          38078.312500
9997199           Limited          38978.777344
9999562            Luxury          35750.925781

[1000 rows x 2 columns]
```

In [1898]:
```
final_outputs['Index'] = final_outputs.index
final_outputs = final_outputs[['Index',output_data.columns[0],output_data.colu
mns[1]]]
final_outputs.head()
```

Out[1898]:

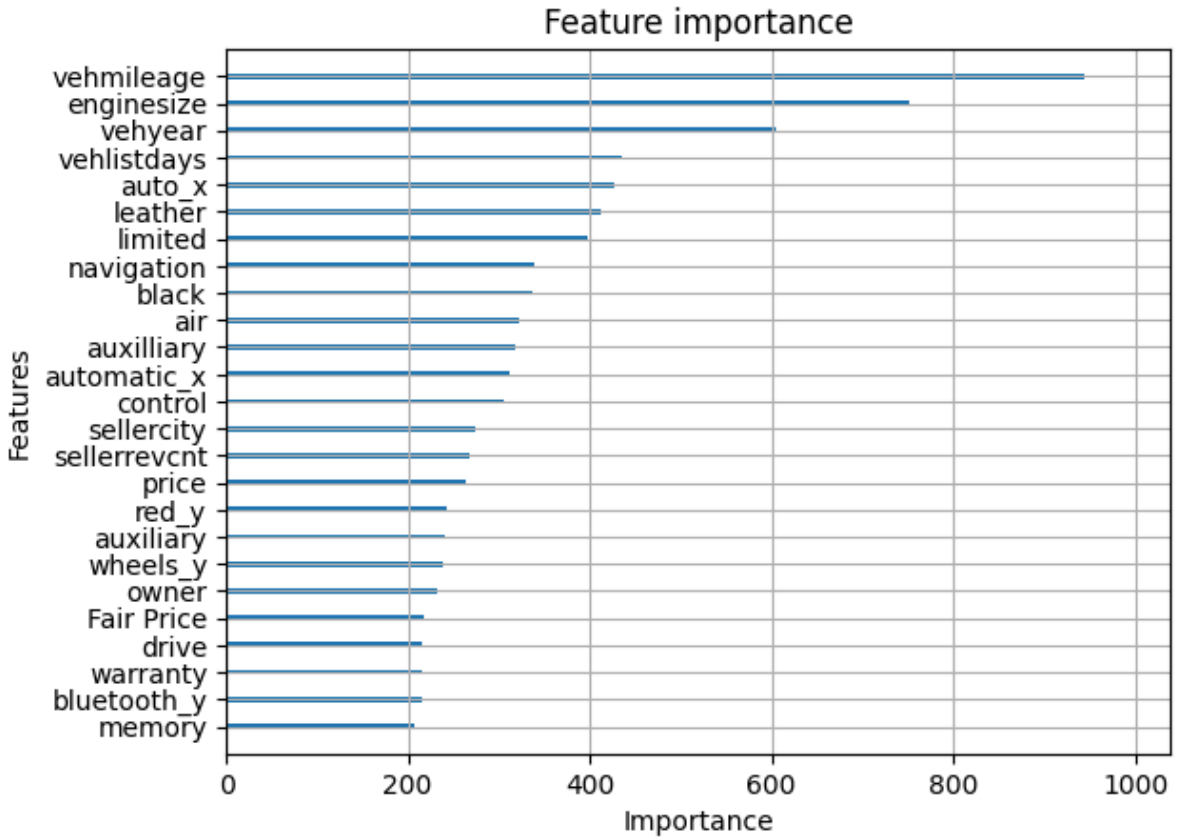
	Index	vehicle_trim	dealer_listing_price
listingid			
8622015	8622015	Premium Luxury	36669.597656
8625693	8625693	Limited	24815.503906
8625750	8625750	Laredo	23464.814453
8626885	8626885	Limited	24937.939453
8627430	8627430	Premium Luxury	39133.730469

In [1899]:
```
print(final_outputs.isna().sum())
```

```
Index                 0
vehicle_trim          0
dealer_listing_price  0
dtype: int64
```

In [1900]:
```
final_outputs.to_csv('submission.csv', index=False, header=False)
```

In [1901]:
```
xgb.plot_importance(clfj.model,max_num_features=25,importance_type='weight', s
how_values=False, xlabel='Importance', ylabel='Features', orientation='horizon
tal')
plt.show()
```



In [1902]:
```
#Gathering stats using expected price calculation on training data splits
test_exp_j = tr.XGB_Regressor(post_feat_engj,list_pricej,jeep_veh_trim,jeep_en
coder,True,True)

test_exp_c = tr.XGB_Regressor(post_feat_engc,list_pricec,caddy_veh_trim,caddy_
encoder,True,True)
```

```
DUMMIES:             Altitude  Laredo  Limited  Overland    SRT  Sterling Edi
tion  \
listingid
5168348       False    True    False     False  False             False
2467433       False   False     True     False  False             False
5750036       False   False    False     False  False             False
6878110       False    True    False     False  False             False
712545        False   False     True     False  False             False
...             ...     ...      ...       ...    ...               ...
6479897       False   False     True     False  False             False
1143545       False    True    False     False  False             False
4126695        True   False    False     False  False             False
3692251       False   False    False     False   True             False
8037873       False   False     True     False  False             False

          Summit  Trackhawk  Trailhawk
listingid
5168348     False      False      False
2467433     False      False      False
5750036      True      False      False
6878110     False      False      False
712545      False      False      False
...           ...        ...        ...
6479897     False      False      False
1143545     False      False      False
4126695     False      False      False
3692251     False      False      False
8037873     False      False      False

[2736 rows x 9 columns]
SECOND :             sellercity  sellerrating  sellerrevcnt  vehcertified  ve
hlistdays  \
listingid
4777                 17           4.8          1405             1         29.0
6242                  4           4.4            21             0         60.0
10882                 2           3.0            51             0         31.0
12013                 3           3.5            17             0        195.0
12334                25           4.6           240             0         29.0
...                 ...           ...           ...           ...          ...
8610847              18           4.8          1016             0        103.0
8612731               5           4.9           121             1         39.0
8614177              35           1.5             6             0         40.0
8615510              27           3.3            16             0          5.0
8620012              21           3.8           727             0         21.0

          vehmileage  vehyear  enginesize  cylinders  owners  ...        up
\
listingid                                                      ...
4777         38957.0        9         3.6          6       1  ...  0.171019
6242         20404.0        6         3.6          6       1  ...  0.000000
10882        34649.0        6         3.6          6       1  ...  0.000000
12013        48814.0        7         3.6          6       1  ...  0.125957
12334        29095.0        6         3.6          6       1  ...  0.287750
...              ...      ...         ...        ...     ...  ...       ...
8610847       1644.0        6         3.6          6       1  ...  0.058345
8612731      31369.0        9         3.6          6       1  ...  0.000000
8614177      35773.0        9         3.6          6       1  ...  0.000000
```

```
8615510       20039.0        9         3.0          6       1  ...  0.000000
8620012       17806.0        6         3.6          6       1  ...  0.213997

                us        v6  vehicles  warranty     wheel  wheels_y  \
listingid
4777       0.308908  0.000000  0.000000  0.165515  0.000000  0.000000
6242       0.343764  0.000000  0.000000  0.000000  0.000000  0.000000
10882      0.000000  0.150977  0.000000  0.166317  0.000000  0.166582
12013      0.000000  0.000000  0.000000  0.121903  0.111063  0.122097
12334      0.000000  0.000000  0.000000  0.139244  0.000000  0.000000
...             ...       ...       ...       ...       ...       ...
8610847    0.105387  0.102517  0.053794  0.000000  0.051446  0.056557
8612731    0.060796  0.059140  0.000000  0.325748  0.000000  0.065253
8614177    0.000000  0.000000  0.231065  0.242547  0.000000  0.000000
8615510    0.000000  0.000000  0.239290  0.251180  0.000000  0.000000
8620012    0.000000  0.000000  0.197305  0.000000  0.000000  0.000000

              will       x27  x3d
listingid
4777       0.000000  0.156198  0.0
6242       0.777900  0.347645  0.0
10882      0.175603  0.000000  0.0
12013      0.000000  0.230082  0.0
12334      0.000000  0.262812  0.0
...             ...       ...  ...
8610847    0.059620  0.213154  0.0
8612731    0.000000  0.061482  0.0
8614177    0.256090  0.000000  0.0
8615510    0.265205  0.000000  0.0
8620012    0.437347  0.000000  0.0

[3420 rows x 167 columns]
           sellercity  sellerrating  sellerrevcnt  vehcertified  vehlistdays
\
listingid
5168348             6           4.6           287             0          2.0
2467433            26           3.8           726             0         30.0
5750036            16           4.8           260             0         77.0
6878110            13           4.9         14628             0         23.0
712545              1           5.0             5             0        567.0
...               ...           ...           ...           ...          ...
6479897             3           4.9           420             0         20.0
1143545            13           5.0            15             0         80.0
4126695            97           4.9          1448             1         31.0
3692251             7           4.8          1076             0         69.0
8037873            19           3.8           726             0         23.0

           vehmileage  vehyear  enginesize  cylinders  owners  ...  x3d  \
listingid                                                      ...
5168348       38944.0        7         3.6          6       1  ...  0.0
2467433       17633.0        7         3.6          6       1  ...  0.0
5750036       24795.0        8         3.6          6       1  ...  0.0
6878110       37662.0        6         3.6          6       1  ...  0.0
712545        31827.0        9         3.6          6       1  ...  0.0
...               ...      ...         ...        ...     ...  ...  ...
6479897       28246.0        9         3.6          6       1  ...  0.0
1143545       17580.0        7         3.6          6       1  ...  0.0
```

```
4126695       29146.0        6         5.7          8       1  ...  0.0
3692251       28114.0        9         6.4          8       1  ...  0.0
8037873       46388.0        9         3.6          6       1  ...  0.0

           Altitude  Laredo  Limited  Overland    SRT  Sterling Edition  \
listingid
5168348       False    True    False     False  False             False
2467433       False   False     True     False  False             False
5750036       False   False    False     False  False             False
6878110       False    True    False     False  False             False
712545        False   False     True     False  False             False
...             ...     ...      ...       ...    ...               ...
6479897       False   False     True     False  False             False
1143545       False    True    False     False  False             False
4126695        True   False    False     False  False             False
3692251       False   False    False     False   True             False
8037873       False   False     True     False  False             False

           Summit  Trackhawk  Trailhawk
listingid
5168348     False      False      False
2467433     False      False      False
5750036      True      False      False
6878110     False      False      False
712545      False      False      False
...           ...        ...        ...
6479897     False      False      False
1143545     False      False      False
4126695     False      False      False
3692251     False      False      False
8037873     False      False      False

[2736 rows x 176 columns]
Best Regression Parameters:  {'learning_rate': 0.05, 'max_depth': 7, 'n_estim
ators': 200}
Best Regression Score R2:  0.9514298867968307
```

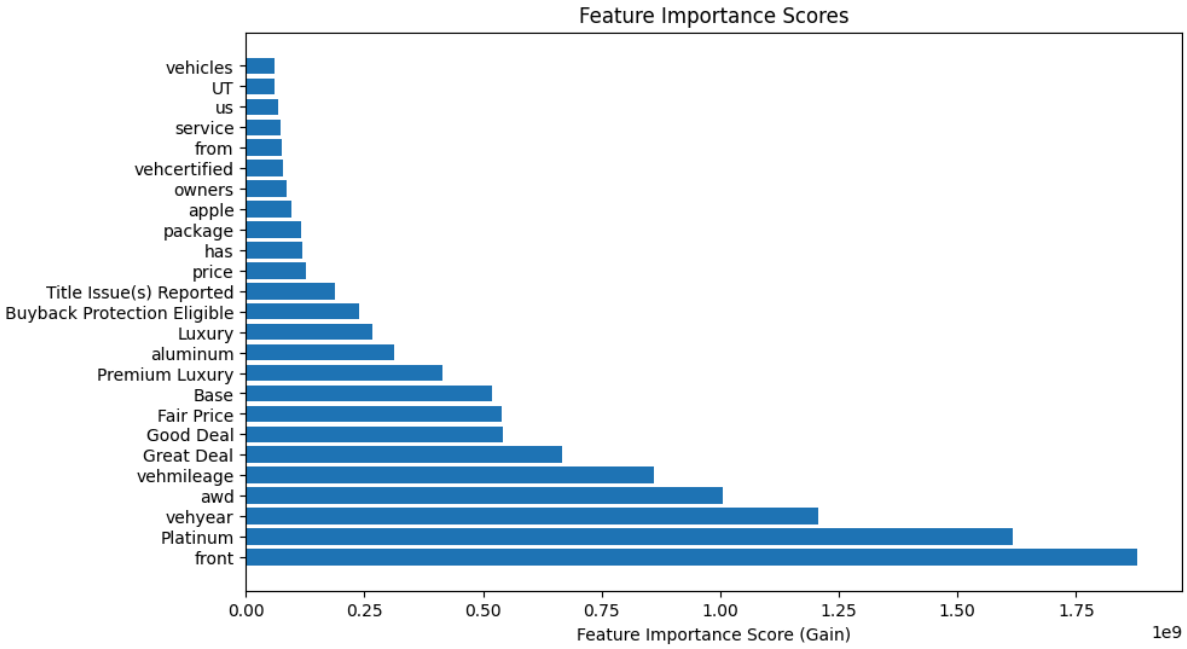
```
C:\Users\aflyn\AppData\Local\Programs\Python\Python312\Lib\site-packages\skle
arn\metrics\_classification.py:1471: UndefinedMetricWarning: Precision is ill
-defined and being set to 0.0 in labels with no predicted samples. Use `zero_
division` parameter to control this behavior.
  _warn_prf(average, modifier, msg_start, len(result))
C:\Users\aflyn\AppData\Local\Programs\Python\Python312\Lib\site-packages\skle
arn\metrics\_classification.py:1471: UndefinedMetricWarning: Precision is ill
-defined and being set to 0.0 in labels with no predicted samples. Use `zero_
division` parameter to control this behavior.
  _warn_prf(average, modifier, msg_start, len(result))
C:\Users\aflyn\AppData\Local\Programs\Python\Python312\Lib\site-packages\skle
arn\metrics\_classification.py:1471: UndefinedMetricWarning: Precision is ill
-defined and being set to 0.0 in labels with no predicted samples. Use `zero_
division` parameter to control this behavior.
  _warn_prf(average, modifier, msg_start, len(result))
C:\Users\aflyn\AppData\Local\Programs\Python\Python312\Lib\site-packages\skle
arn\metrics\_classification.py:1471: UndefinedMetricWarning: Precision is ill
-defined and being set to 0.0 in labels with no predicted samples. Use `zero_
division` parameter to control this behavior.
  _warn_prf(average, modifier, msg_start, len(result))
C:\Users\aflyn\AppData\Local\Programs\Python\Python312\Lib\site-packages\skle
arn\metrics\_classification.py:1471: UndefinedMetricWarning: Precision is ill
-defined and being set to 0.0 in labels with no predicted samples. Use `zero_
division` parameter to control this behavior.
  _warn_prf(average, modifier, msg_start, len(result))
C:\Users\aflyn\AppData\Local\Programs\Python\Python312\Lib\site-packages\skle
arn\metrics\_classification.py:1471: UndefinedMetricWarning: Precision is ill
-defined and being set to 0.0 in labels with no predicted samples. Use `zero_
division` parameter to control this behavior.
  _warn_prf(average, modifier, msg_start, len(result))
C:\Users\aflyn\AppData\Local\Programs\Python\Python312\Lib\site-packages\skle
arn\metrics\_classification.py:1471: UndefinedMetricWarning: Precision is ill
-defined and being set to 0.0 in labels with no predicted samples. Use `zero_
division` parameter to control this behavior.
  _warn_prf(average, modifier, msg_start, len(result))
C:\Users\aflyn\AppData\Local\Programs\Python\Python312\Lib\site-packages\skle
arn\metrics\_classification.py:1471: UndefinedMetricWarning: Precision is ill
-defined and being set to 0.0 in labels with no predicted samples. Use `zero_
division` parameter to control this behavior.
  _warn_prf(average, modifier, msg_start, len(result))
C:\Users\aflyn\AppData\Local\Programs\Python\Python312\Lib\site-packages\skle
arn\metrics\_classification.py:1471: UndefinedMetricWarning: Precision is ill
-defined and being set to 0.0 in labels with no predicted samples. Use `zero_
division` parameter to control this behavior.
  _warn_prf(average, modifier, msg_start, len(result))
C:\Users\aflyn\AppData\Local\Programs\Python\Python312\Lib\site-packages\skle
arn\metrics\_classification.py:1471: UndefinedMetricWarning: Precision is ill
-defined and being set to 0.0 in labels with no predicted samples. Use `zero_
division` parameter to control this behavior.
  _warn_prf(average, modifier, msg_start, len(result))
```

```
Best Parameters:  {'learning_rate': 0.1, 'max_depth': 5, 'n_estimators': 200}
Best Score:  0.9428785876524717
R2 Score for straight model predict: 0.9483743317477421
R2 Score for Exp price calc: 0.9481416481654419
EXPLAINED VARIANCE  0.9482183320576489
rmse:  1664.477929063799
DUMMIES:              Base  Luxury  Platinum  Premium Luxury
listingid
1284766    False   False     False            True
1578334    False   False     False            True
3629973    False    True     False           False
5133024    False    True     False           False
3762156    False   False     False            True
...          ...     ...       ...             ...
6871295     True   False     False           False
1956592    False    True     False           False
4397435    False   False     False            True
3754507    False   False     False            True
610777     False    True     False           False

[1256 rows x 4 columns]
SECOND :             sellercity  sellerrating  sellerrevcnt  vehcertified  ve
hlistdays  \
listingid
7108                2           3.7            74             0         99.0
21448               3           3.7           629             0          6.0
21807               4           4.9           360             1          7.0
30524               3           5.0             4             0         12.0
34061               2           4.3           312             0         26.0
...               ...           ...           ...           ...          ...
8599564             4           4.4            15             1         35.0
8601212             2           4.8           461             0         19.0
8604205             6           4.7            58             1         20.0
8616294             3           4.1            20             1        185.0
8617378             1           4.9           278             0         74.0

           vehmileage  vehyear  enginesize  cylinders  owners  ...      this
\
listingid                                                       ...
7108          19788.0        6         3.6          6       1  ...  0.098747
21448         21098.0        6         3.6          6       1  ...  0.000000
21807          3547.0        5         3.6          6       1  ...  0.000000
30524         12146.0        7         3.6          6       1  ...  0.334768
34061         28293.0        6         3.6          6       1  ...  0.182420
...               ...      ...         ...        ...     ...  ...       ...
8599564       41886.0        6         3.6          6       1  ...  0.024555
8601212       39882.0        7         3.6          6       1  ...  0.356044
8604205       17314.0        6         3.6          6       1  ...  0.000000
8616294       16278.0        6         3.6          6       0  ...  0.000000
8617378       38146.0        7         3.6          6       2  ...  0.000000

                 us        v6  vehicles  warranty     wheel      will  \
listingid
7108       0.231728  0.000000  0.101986  0.000000  0.110024  0.119530
21448      0.000000  0.000000  0.135329  0.000000  0.000000  0.317216
21807      0.177843  0.000000  0.000000  0.517238  0.000000  0.183471
30524      0.000000  0.000000  0.172875  0.190402  0.000000  0.202614
```

```
34061      0.107021  0.000000  0.188405  0.000000  0.101626  0.000000
...             ...       ...       ...       ...       ...       ...
8599564    0.000000  0.024942  0.000000  0.000000  0.054718  0.000000
8601212    0.000000  0.000000  0.000000  0.000000  0.000000  0.215491
8604205    0.000000  0.000000  0.182802  0.000000  0.000000  0.000000
8616294    0.000000  0.140945  0.000000  0.473517  0.000000  0.167962
8617378    0.000000  0.164172  0.166927  0.000000  0.000000  0.000000

               x27  x3d      your
listingid
7108       0.216169  0.0  0.308417
21448      0.000000  0.0  0.545664
21807      0.000000  0.0  0.000000
30524      0.000000  0.0  0.000000
34061      0.299505  0.0  0.000000
...             ...  ...       ...
8599564    0.000000  0.0  0.000000
8601212    0.194856  0.0  0.185340
8604205    0.000000  0.0  0.184271
8616294    0.151879  0.0  0.433384
8617378    0.000000  0.0  0.168268

[1570 rows x 189 columns]
           sellercity  sellerrating  sellerrevcnt  vehcertified  vehlistdays
\
listingid
1284766             7           4.2           934             0        273.0
1578334             4           3.7           629             0         83.0
3629973             5           5.0           136             0         15.0
5133024             1           4.6           155             0         59.0
3762156             8           4.7           105             1        289.0
...               ...           ...           ...           ...          ...
6871295            18           4.6          7094             1        236.0
1956592            10           3.5           450             0         28.0
4397435             2           3.7           629             0         83.0
3754507             1           5.0            17             0         49.0
610777             37           4.8            59             1          5.0

           vehmileage  vehyear  enginesize  cylinders  owners  ...  warranty
\
listingid                                                       ...
1284766       18204.0        6         3.6          6       1  ...  0.000000
1578334       11310.0        7         3.6          6       1  ...  0.000000
3629973        5731.0        6         3.6          6       1  ...  0.000000
5133024        5931.0        6         3.6          6       1  ...  0.000000
3762156       12995.0        6         3.6          6       1  ...  0.000000
...               ...      ...         ...        ...     ...  ...       ...
6871295        6460.0        6         3.6          6       0  ...  0.190547
1956592        3881.0        5         3.6          6       1  ...  0.063010
4397435       11310.0        7         3.6          6       1  ...  0.000000
3754507       33643.0        6         3.6          6       1  ...  0.000000
610777         9548.0        6         3.6          6       0  ...  0.196659

              wheel      will       x27  x3d      your   Base  Luxury  \
listingid
1284766    0.390669  0.000000  0.127927  0.0  0.000000  False   False
1578334    0.000000  0.317216  0.000000  0.0  0.545664  False   False
```

```
3629973    0.000000  0.000000  0.579071  0.0  0.220316  False    True
5133024    0.000000  0.000000  0.181483  0.0  0.000000  False    True
3762156    0.234219  0.000000  0.230091  0.0  0.109427  False   False
...             ...       ...       ...  ...       ...    ...     ...
6871295    0.000000  0.101384  0.000000  0.0  0.523192   True   False
1956592    0.123437  0.000000  0.121261  0.0  0.000000  False    True
4397435    0.000000  0.317216  0.000000  0.0  0.545664  False   False
3754507    0.000000  0.000000  0.000000  0.0  0.000000  False   False
610777     0.096314  0.104636  0.094616  0.0  0.000000  False    True

           Platinum  Premium Luxury
listingid
1284766       False           True
1578334       False           True
3629973       False          False
5133024       False          False
3762156       False           True
...             ...            ...
6871295       False          False
1956592       False          False
4397435       False           True
3754507       False           True
610777        False          False

[1256 rows x 193 columns]
Best Regression Parameters:  {'learning_rate': 0.1, 'max_depth': 3, 'n_estima
tors': 200}
Best Regression Score R2:  0.8468536588262524
Best Parameters:  {'learning_rate': 0.1, 'max_depth': 5, 'n_estimators': 200}
Best Score:  0.9773581904393623
R2 Score for straight model predict: 0.8580872344571056
R2 Score for Exp price calc: 0.856391085977314
EXPLAINED VARIANCE  0.8564651618521013
rmse:  2283.8107204034545
```

In [1903]:
```
importance_scores = regc.model.get_booster().get_score(importance_type='gain')
sorted_importance = sorted(importance_scores.items(), key=lambda x: x[1], reve
rse=True)[:25]

# Separate feature names and scores for plotting
feature_names, scores = zip(*sorted_importance)

# Create a bar plot
plt.figure(figsize=(10, 6))
plt.barh(range(len(feature_names)), scores, align='center')
plt.yticks(range(len(feature_names)), feature_names)
plt.xlabel('Feature Importance Score (Gain)')
plt.title('Feature Importance Scores')
plt.show()
```



In [ ]: