2_Engineer_Training_Dataset

May 3, 2021

[1]:
```
import modin.pandas as pd
import numpy as np

import pickle

from sklearn.preprocessing import QuantileTransformer

import seaborn as sns
```

[2]:
```
training_dat = pd.read_pickle('./PROCESSED/training_dat.pkl')
training_dat.head()
```

```
UserWarning: Ray execution environment not yet initialized. Initializing…
To remove this warning, run the following python code before doing dataframe
operations:

    import ray
    ray.init()

UserWarning: `read_pickle` defaulting to pandas implementation.
To request implementation, send an email to feature_requests@modin.org.
```

[2]:

	Loan Sequence Number	Loan Deliquency Within Year	Credit Score	First Payment Date	First Time Homebuyer Flag	Maturity Date	Metropolitan Statistical Area (MSA) Or Metropolitan Division
0	F110Q1000008	False	804	201004	N	204003	17860.0
1	F110Q1000064	False	816	201003	N	204002	
2	F110Q1000072	False	783	201003	N	204002	
3	F110Q1000080	False	667	201003	N	204002	
4	F110Q1000096	False	799	201003	N	204002	

```
1                                          44220.0
2                                          45820.0
3                                          17900.0
4                                              NaN

   Mortgage Insurance Percentage (MI %)  Number of Units Occupancy Status  \
0                                    0               1                P
1                                    0               1                P
2                                    0               1                P
3                                    0               1                P
4                                    0               1                P

   …  Original Loan Term Number of Borrowers    Seller Name  \
0  …                 360                   2  Other sellers
1  …                 360                   1  Other sellers
2  …                 360                   2  Other sellers
3  …                 360                   1  Other sellers
4  …                 360                   1  Other sellers

     Servicer Name Super Conforming Flag Pre-HARP Loan Sequence Number  \
0   U.S. BANK N.A.                   NaN                          NaN
1  Other servicers                   NaN                          NaN
2   U.S. BANK N.A.                   NaN                          NaN
3  Other servicers                   NaN                          NaN
4   U.S. BANK N.A.                   NaN                          NaN

  Program Indicator HARP Indicator Property Valuation Method  \
0                 9            NaN                         9
1                 9            NaN                         9
2                 9            NaN                         9
3                 9            NaN                         9
4                 9            NaN                         9

  Interest Only (I/O) Indicator
0                           NaN
1                           NaN
2                           NaN
3                           NaN
4                           NaN

[5 rows x 32 columns]
```

[3]:
```
training_dat.dtypes
```

[3]:
```
Loan Sequence Number                                    object
Loan Deliquency Within Year                               bool
Credit Score                                             int64
```

```
First Payment Date                                                  int64
First Time Homebuyer Flag                                          object
Maturity Date                                                       int64
Metropolitan Statistical Area (MSA) Or Metropolitan Division      float64
Mortgage Insurance Percentage (MI %)                                int64
Number of Units                                                     int64
Occupancy Status                                                   object
Original Combined Loan-to-Value (CLTV)                              int64
Original Debt-to-Income (DTI) Ratio                                 int64
Original UPB                                                        int64
Original Loan-to-Value (LTV)                                        int64
Original Interest Rate                                            float64
Channel                                                            object
Prepayment Penalty Mortgage (PPM) Flag                             object
Amortization Type (Formerly Product Type)                          object
Property State                                                     object
Property Type                                                      object
Postal Code                                                         int64
Loan Purpose                                                       object
Original Loan Term                                                  int64
Number of Borrowers                                                 int64
Seller Name                                                        object
Servicer Name                                                      object
Super Conforming Flag                                              object
Pre-HARP Loan Sequence Number                                      object
Program Indicator                                                  object
HARP Indicator                                                     object
Property Valuation Method                                           int64
Interest Only (I/O) Indicator                                     float64
dtype: object
```

[4]:
```
# Get rid of HARP record
training_dat = training_dat[training_dat['HARP Indicator']!='Y'].copy()
```

[5]:
```
# The improved training dataset dataframe. Starting with initial columns, and then adding predictors
training_dat2 = training_dat[['Loan Sequence Number','Loan Deliquency Within Year']].copy()
training_dat2.head()
```

[5]:

	Loan Sequence Number	Loan Deliquency Within Year
0	F110Q1000008	False
1	F110Q1000064	False
2	F110Q1000072	False
3	F110Q1000080	False
4	F110Q1000096	False

[6]:
```
# Storing scalers
scaler_dct = {}
```

[7]:
```
# Cleaning Credit Score
training_dat['Credit Score'].describe()
```

[7]:
```
count    396337.000000
mean        755.107507
std          81.524064
min         508.000000
25%         725.000000
50%         764.000000
75%         790.000000
max        9999.000000
Name: Credit Score, dtype: float64
```

[8]:
```
tf = training_dat['Credit Score'] == 9999
training_dat.loc[tf, 'Credit Score'] = np.nan
training_dat['Credit Score'].describe()
```

[8]:
```
count    396315.000000
mean        754.594366
std          43.621586
min         508.000000
25%         725.000000
50%         764.000000
75%         790.000000
max         850.000000
Name: Credit Score, dtype: float64
```

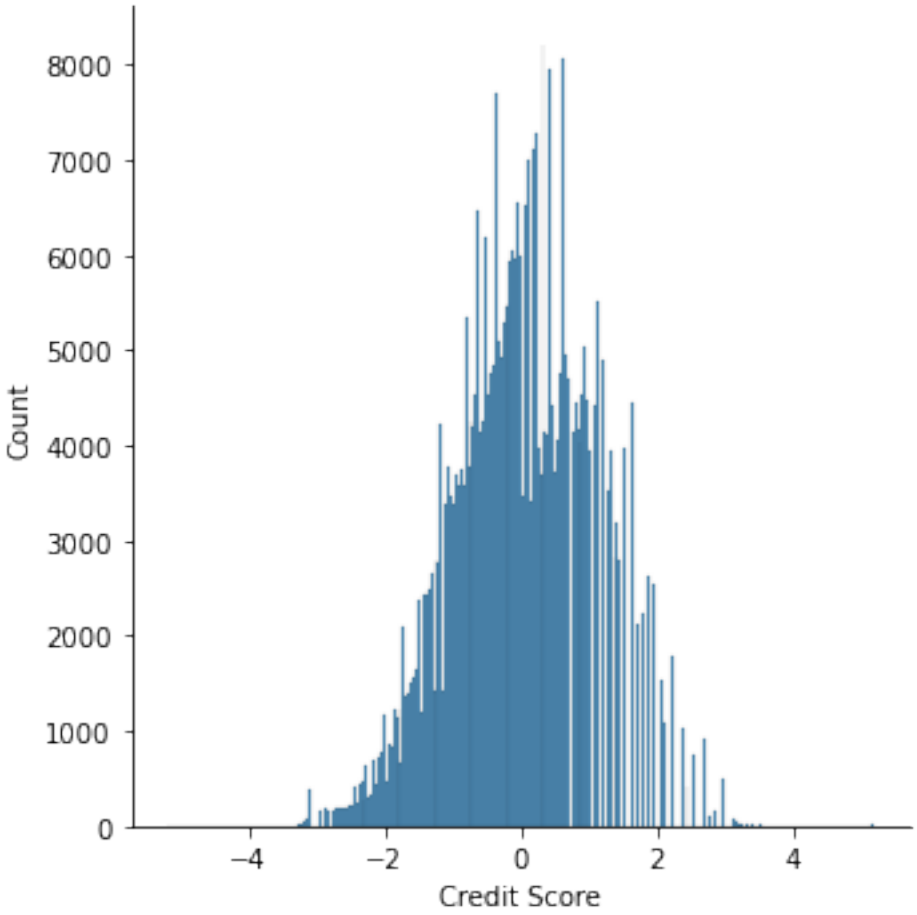
[9]:
```
scaler_0_1_creditscore = QuantileTransformer(output_distribution='normal')
training_dat2['Credit Score'] = scaler_0_1_creditscore.
 ↪fit_transform(training_dat[['Credit Score']])
training_dat2.head()
```

[9]:

	Loan Sequence Number	Loan Deliquency Within Year	Credit Score
0	F110Q1000008	False	1.315958
1	F110Q1000064	False	2.225823
2	F110Q1000072	False	0.451469
3	F110Q1000080	False	-1.733071
4	F110Q1000096	False	1.033647

[10]:
```
sns.displot(x='Credit Score',data=training_dat2)
```

[10]:
```
<seaborn.axisgrid.FacetGrid at 0x212d511c080>
```



[11]:
```
training_dat['Credit Score'].head()
```

[11]:
```
0    804
1    816
2    783
3    667
4    799
Name: Credit Score, dtype: int64
```

[12]:
```
scaler_dct['Credit Score'] = scaler_0_1_creditscore
```

[13]:
```
# Cleaning First Time Homebuyer Flag
training_dat['First Time Homebuyer Flag'].value_counts()
```

```
UserWarning: value_counts defaulting to pandas implementation.
```

[13]:
```
9    215276
N    117864
```

```
Y     63197
Name: First Time Homebuyer Flag, dtype: int64
```

[14]:
```
training_dat2['First Time Homebuyer Flag'] = np.nan
tf1 = training_dat['First Time Homebuyer Flag'] == 'Y'
training_dat2.loc[tf1, 'First Time Homebuyer Flag'] = True
tf2 = training_dat['First Time Homebuyer Flag'] == 'N'
training_dat2.loc[tf2, 'First Time Homebuyer Flag'] = False
training_dat2.head()
```

[14]:

	Loan Sequence Number	Loan Deliquency Within Year	Credit Score	First Time Homebuyer Flag
0	F110Q1000008	False	1.315958	False
1	F110Q1000064	False	2.225823	False
2	F110Q1000072	False	0.451469	False
3	F110Q1000080	False	-1.733071	False
4	F110Q1000096	False	1.033647	False

[15]:
```
training_dat2['First Time Homebuyer Flag'].unique()
```

[15]:
```
array([False, True, nan], dtype=object)
```

[16]:
```
training_dat[training_dat['First Time Homebuyer Flag']=='Y'][['Loan Sequence␣
↪Number','First Time Homebuyer Flag']].head()
```

[16]:

	Loan Sequence Number	First Time Homebuyer Flag
7	F110Q1000168	Y
18	F110Q1000443	Y
37	F110Q1000922	Y
38	F110Q1000963	Y
43	F110Q1001060	Y

[17]:
```
training_dat2[training_dat2['First Time Homebuyer Flag']==True].head()
```

[17]:

	Loan Sequence Number	Loan Deliquency Within Year	Credit Score	First Time Homebuyer Flag
7	F110Q1000168	False	-0.465405	True
18	F110Q1000443	False	0.396560	True
37	F110Q1000922	False	1.959536	True
38	F110Q1000963	False	-1.895259	True
43	F110Q1001060	False	0.670557	True

```
   First Time Homebuyer Flag
7                        True
18                       True
37                       True
38                       True
43                       True
```

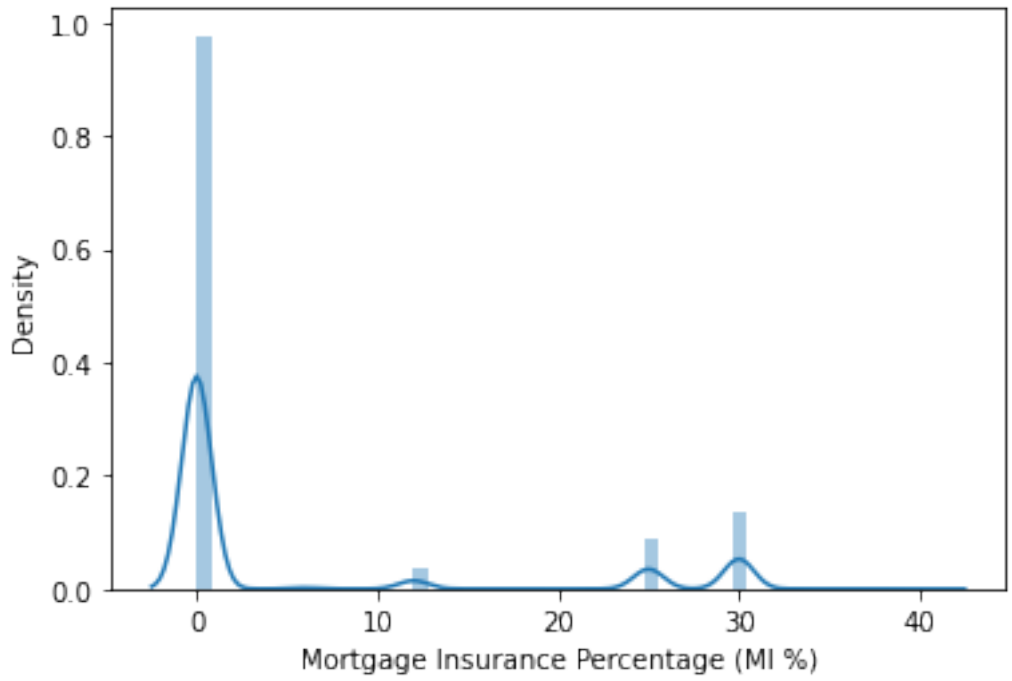
[18]:
```
# Cleaning Mortgage Insurance Percentage (MI %)
training_dat['Mortgage Insurance Percentage (MI %)'].describe()
```

[18]:
```
count    396337.000000
mean          5.540459
std          10.918153
min           0.000000
25%           0.000000
50%           0.000000
75%           0.000000
max          40.000000
Name: Mortgage Insurance Percentage (MI %), dtype: float64
```

[19]:
```
sns.distplot(training_dat['Mortgage Insurance Percentage (MI %)'])
```

```
FutureWarning: `distplot` is a deprecated function and will be removed in a
future version. Please adapt your code to use either `displot` (a figure-level
function with similar flexibility) or `histplot` (an axes-level function for
histograms).
```

[19]:
```
<AxesSubplot:xlabel='Mortgage Insurance Percentage (MI %)', ylabel='Density'>
```



[20]:
```
training_dat['Mortgage Insurance Percentage (MI %)'].unique()
```

[20]:
```
array([ 0, 30, 25, 12, 17, 20,  6, 35, 14, 18, 16, 13, 15, 36, 40,  7, 37,
       27, 22, 32, 26], dtype=int64)
```

[21]:
```
# scaler_0_1_mortpct = QuantileTransformer(output_distribution='normal')
# training_dat2['Mortgage Insurance Percentage (MI %)'] = scaler_0_1_mortpct.
 ↪fit_transform(training_dat[['Mortgage Insurance Percentage (MI %)']])
# training_dat2.head()
```

[26]:
```
# scaler_dct['Mortgage Insurance Percentage (MI %)'] = scaler_0_1_mortpct
```

[28]:
```
training_dat2['MortgageInsuranceFlag'] = 1

tf = training_dat['Mortgage Insurance Percentage (MI %)'] == 0
training_dat2.loc[tf, 'MortgageInsuranceFlag'] = 0
```

[29]:
```
# Transform Number of Units
training_dat['Number of Units'].unique()
```

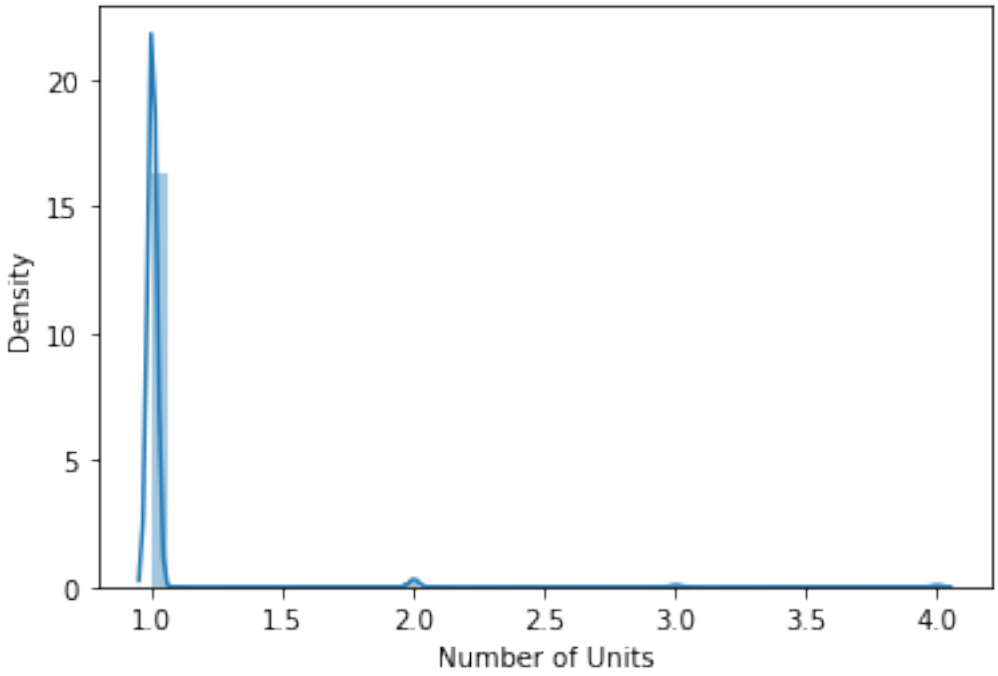
[29]:
```
array([1, 4, 2, 3], dtype=int64)
```

[30]:
```
sns.distplot(training_dat['Number of Units'])
```

```
FutureWarning: `distplot` is a deprecated function and will be removed in a
```

```
future version. Please adapt your code to use either `displot` (a figure-level
function with similar flexibility) or `histplot` (an axes-level function for
histograms).
```

[30]:
```
<AxesSubplot:xlabel='Number of Units', ylabel='Density'>
```



[31]:
```
dum_df = pd.get_dummies(training_dat['Number of Units'],prefix='Units')
for col in list(dum_df.columns):
    training_dat2[col] = dum_df[col]
training_dat2.head()
```

```
UserWarning: `get_dummies` on non-DataFrame defaulting to pandas implementation.
UserWarning: Distributing <class 'pandas.core.frame.DataFrame'> object. This may
take some time.
```

[31]:

	Loan Sequence Number	Loan Deliquency Within Year	Credit Score	\
0	F110Q1000008	False	1.315958	
1	F110Q1000064	False	2.225823	
2	F110Q1000072	False	0.451469	
3	F110Q1000080	False	-1.733071	
4	F110Q1000096	False	1.033647	

	First Time Homebuyer Flag	MortgageInsuranceFlag	Units_1	Units_2	Units_3	\
0	False	0	1	0	0	
1	False	0	1	0	0	

```
2                    False                         0        1        0        0
3                    False                         0        1        0        0
4                    False                         0        1        0        0

   Units_4
0        0
1        0
2        0
3        0
4        0
```

[32]:
```
# Clean/Transform Occupancy Status
training_dat['Occupancy Status'].unique()
```

[32]:
```
array(['P', 'I', 'S'], dtype=object)
```

[33]:
```
training_dat['Occupancy Status'].value_counts()
```

```
UserWarning: value_counts defaulting to pandas implementation.
```

[33]:
```
P    351163
I     28686
S     16488
Name: Occupancy Status, dtype: int64
```

[34]:
```
dum_df = pd.get_dummies(training_dat['Occupancy␣
 ↪Status'],prefix='OccupancyStatus')
for col in list(dum_df.columns):
    training_dat2[col] = dum_df[col]
training_dat2.head()
```

```
UserWarning: `get_dummies` on non-DataFrame defaulting to pandas implementation.
UserWarning: Distributing <class 'pandas.core.frame.DataFrame'> object. This may
take some time.
```

[34]:
```
  Loan Sequence Number  Loan Deliquency Within Year  Credit Score  \
0         F110Q1000008                        False      1.315958
1         F110Q1000064                        False      2.225823
2         F110Q1000072                        False      0.451469
3         F110Q1000080                        False     -1.733071
4         F110Q1000096                        False      1.033647

  First Time Homebuyer Flag  MortgageInsuranceFlag  Units_1  Units_2  Units_3  \
0                     False                      0        1        0        0
1                     False                      0        1        0        0
2                     False                      0        1        0        0
3                     False                      0        1        0        0
```

```
4                    False                        0        1        0        0

   Units_4  OccupancyStatus_I  OccupancyStatus_P  OccupancyStatus_S
0        0                  0                  1                  0
1        0                  0                  1                  0
2        0                  0                  1                  0
3        0                  0                  1                  0
4        0                  0                  1                  0
```

[35]:
```
# Clean Original Combined Loan-to-Value (CLTV)
training_dat['Original Combined Loan-to-Value (CLTV)'].describe()
```

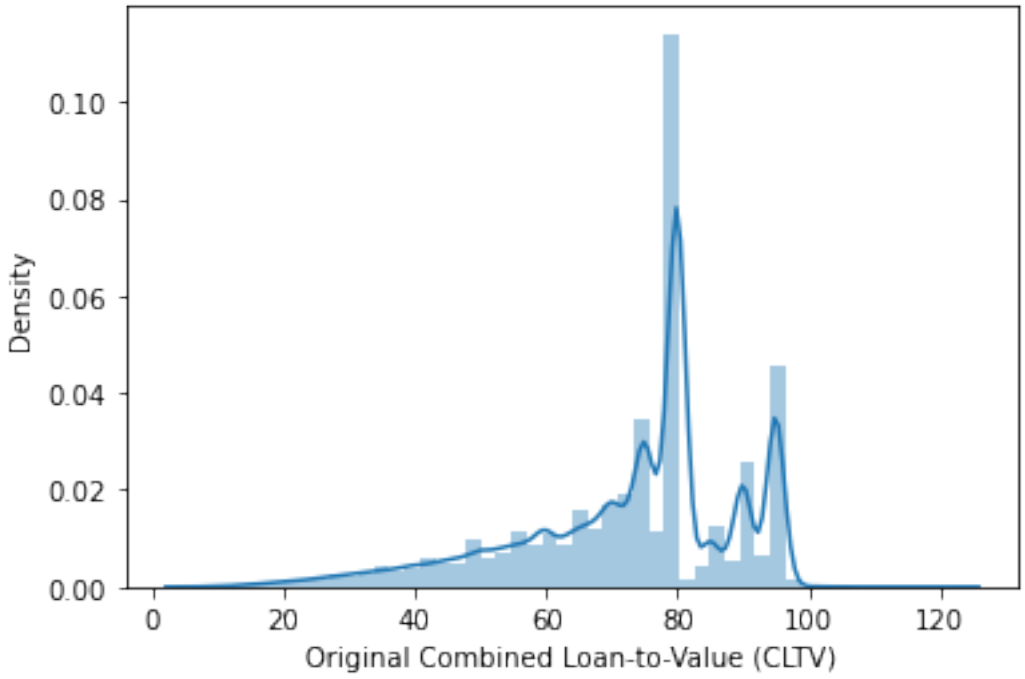
[35]:
```
count    396337.000000
mean         72.872349
std          17.604025
min           6.000000
25%          64.000000
50%          78.000000
75%          80.000000
max         999.000000
Name: Original Combined Loan-to-Value (CLTV), dtype: float64
```

[36]:
```
tf = training_dat['Original Combined Loan-to-Value (CLTV)'] == 999
training_dat.loc[tf, 'Original Combined Loan-to-Value (CLTV)'] = np.nan
```

[37]:
```
sns.distplot(training_dat['Original Combined Loan-to-Value (CLTV)'])
```

```
FutureWarning: `distplot` is a deprecated function and will be removed in a
future version. Please adapt your code to use either `displot` (a figure-level
function with similar flexibility) or `histplot` (an axes-level function for
histograms).
```

[37]:
```
<AxesSubplot:xlabel='Original Combined Loan-to-Value (CLTV)', ylabel='Density'>
```



[42]:
```
scaler_0_1_cltv = QuantileTransformer(output_distribution='normal')
training_dat2['Original Combined Loan-to-Value (CLTV)'] = scaler_0_1_cltv.
 ↪fit_transform(training_dat[['Original Combined Loan-to-Value (CLTV)']])
training_dat2.head()
```

[42]:

	Loan Sequence Number	Loan Deliquency Within Year	Credit Score	\
0	F110Q1000008	False	1.315958	
1	F110Q1000064	False	2.225823	
2	F110Q1000072	False	0.451469	
3	F110Q1000080	False	-1.733071	
4	F110Q1000096	False	1.033647	

	First Time Homebuyer Flag	MortgageInsuranceFlag	Units_1	Units_2	Units_3	\
0	False	0	1	0	0	
1	False	0	1	0	0	
2	False	0	1	0	0	
3	False	0	1	0	0	
4	False	0	1	0	0	

	Units_4	OccupancyStatus_I	OccupancyStatus_P	OccupancyStatus_S	\
0	0	0	1	0	
1	0	0	1	0	
2	0	0	1	0	
3	0	0	1	0	
4	0	0	1	0	

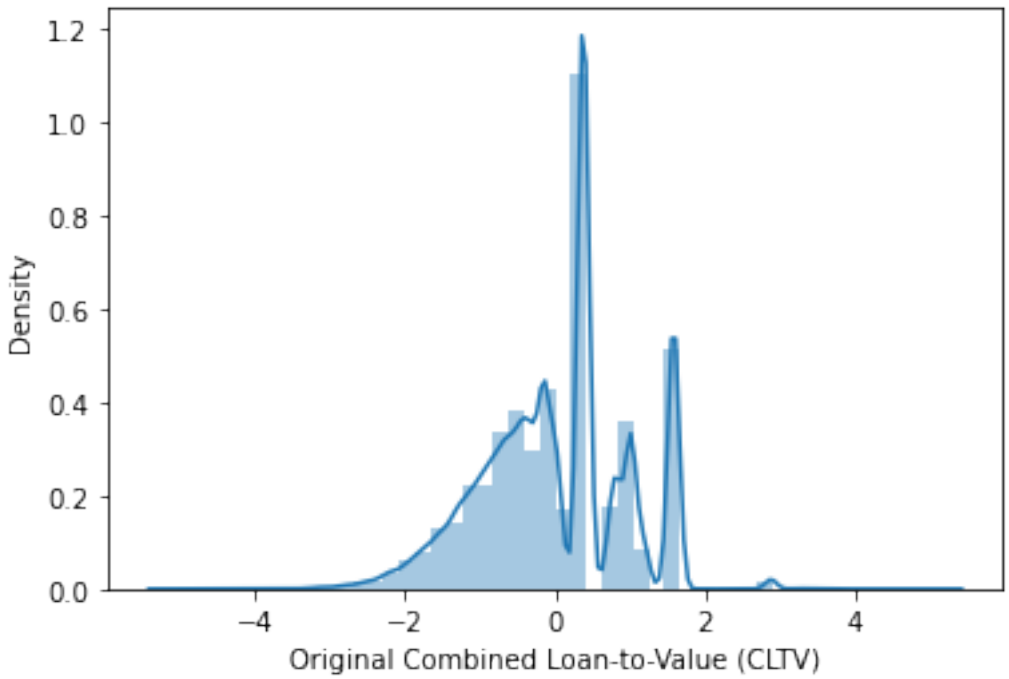
```
   Original Combined Loan-to-Value (CLTV)
0                               -1.606755
1                               -0.979511
2                                0.376283
3                                0.376283
4                                0.056486
```

[43]:
```
scaler_dct['Original Combined Loan-to-Value (CLTV)'] = scaler_0_1_cltv
```

[44]:
```
sns.distplot(training_dat2['Original Combined Loan-to-Value (CLTV)'])
```

```
FutureWarning: `distplot` is a deprecated function and will be removed in a
future version. Please adapt your code to use either `displot` (a figure-level
function with similar flexibility) or `histplot` (an axes-level function for
histograms).
```

[44]: <AxesSubplot:xlabel='Original Combined Loan-to-Value (CLTV)', ylabel='Density'>



[45]:
```
# Clean Original Debt-to-Income (DTI) Ratio
training_dat['Original Debt-to-Income (DTI) Ratio'].describe()
```

[45]:
```
count    396337.000000
mean         33.481499
std          13.532473
```

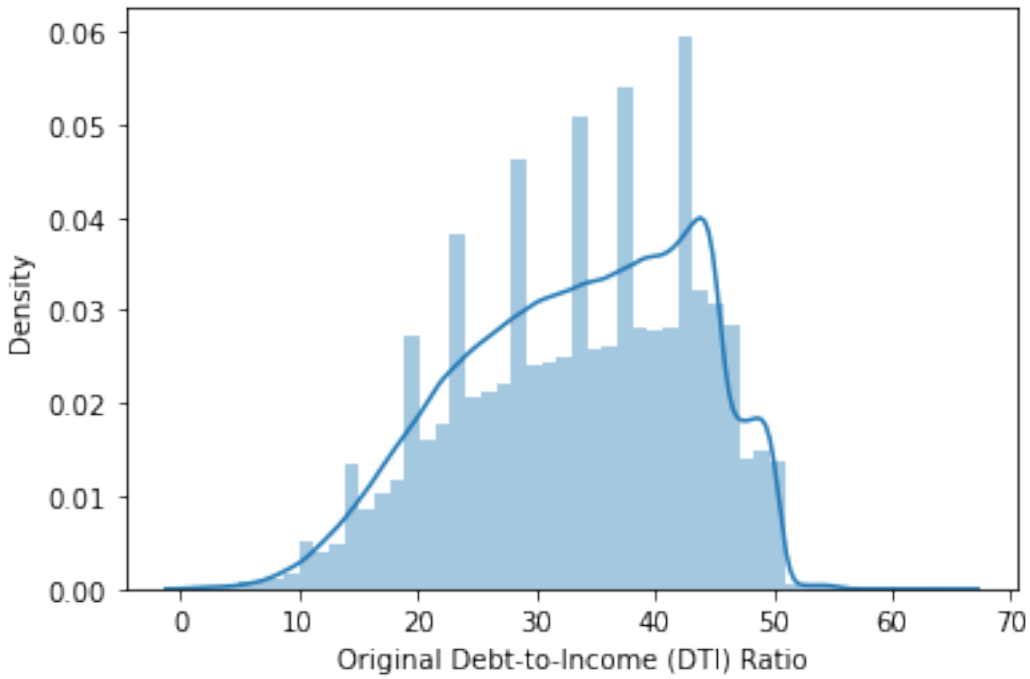
```
min            1.000000
25%           26.000000
50%           34.000000
75%           41.000000
max          999.000000
Name: Original Debt-to-Income (DTI) Ratio, dtype: float64
```

[46]:
```
tf = training_dat['Original Debt-to-Income (DTI) Ratio'] == 999
training_dat.loc[tf, 'Original Debt-to-Income (DTI) Ratio'] = np.nan
```

[47]:
```
sns.distplot(training_dat['Original Debt-to-Income (DTI) Ratio'])
```

```
FutureWarning: `distplot` is a deprecated function and will be removed in a
future version. Please adapt your code to use either `displot` (a figure-level
function with similar flexibility) or `histplot` (an axes-level function for
histograms).
```

[47]:
```
<AxesSubplot:xlabel='Original Debt-to-Income (DTI) Ratio', ylabel='Density'>
```



[48]:
```
scaler_0_1_dti = QuantileTransformer(output_distribution='normal')
training_dat2['Original Debt-to-Income (DTI) Ratio'] = scaler_0_1_dti.
 ↪fit_transform(training_dat[['Original Debt-to-Income (DTI) Ratio']])
training_dat2.head()
```

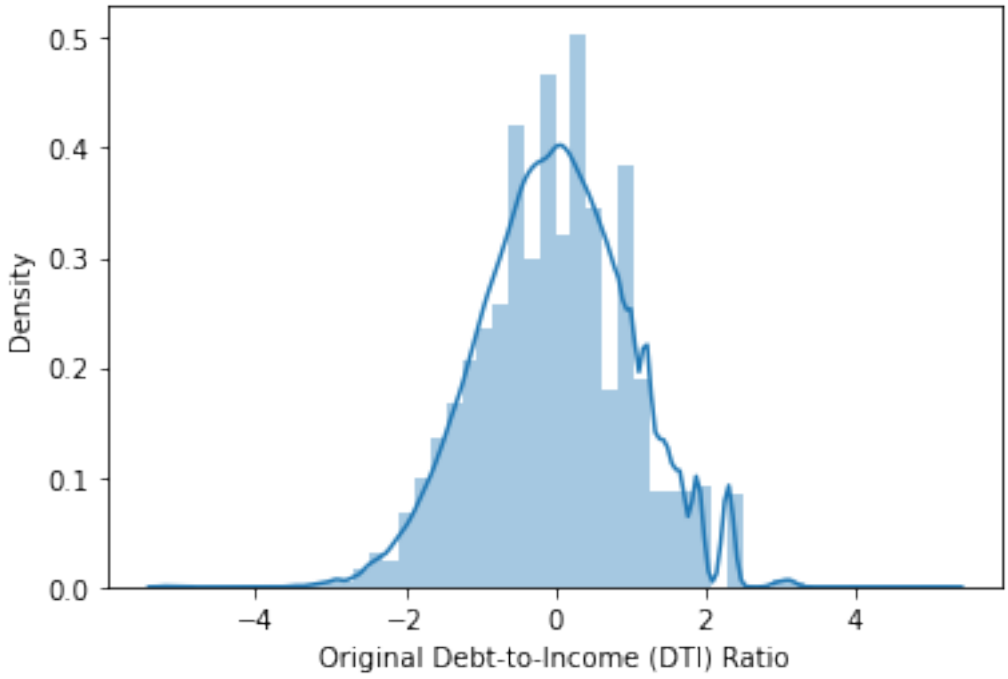
[48]:
	Loan Sequence Number	Loan Deliquency Within Year	Credit Score	First Time Homebuyer Flag	MortgageInsuranceFlag	Units_1	Units_2	Units_3	Units_4	OccupancyStatus_I	OccupancyStatus_P	OccupancyStatus_S	Original Combined Loan-to-Value (CLTV)	Original Debt-to-Income (DTI) Ratio
0	F110Q1000008	False	1.315958	False	0	1	0	0	0	0	1	0	-1.606755	-1.252988
1	F110Q1000064	False	2.225823	False	0	1	0	0	0	0	1	0	-0.979511	-0.599937
2	F110Q1000072	False	0.451469	False	0	1	0	0	0	0	1	0	0.376283	0.139710
3	F110Q1000080	False	-1.733071	False	0	1	0	0	0	0	1	0	0.376283	-1.252988
4	F110Q1000096	False	1.033647	False	0	1	0	0	0	0	1	0	0.056486	-1.684464

[49]:
```
scaler_dct['Original Debt-to-Income (DTI) Ratio'] = scaler_0_1_dti
```

[50]:
```
sns.distplot(training_dat2['Original Debt-to-Income (DTI) Ratio'])
```

```
FutureWarning: `distplot` is a deprecated function and will be removed in a
future version. Please adapt your code to use either `displot` (a figure-level
function with similar flexibility) or `histplot` (an axes-level function for
histograms).
```

[50]:
```
<AxesSubplot:xlabel='Original Debt-to-Income (DTI) Ratio', ylabel='Density'>
```



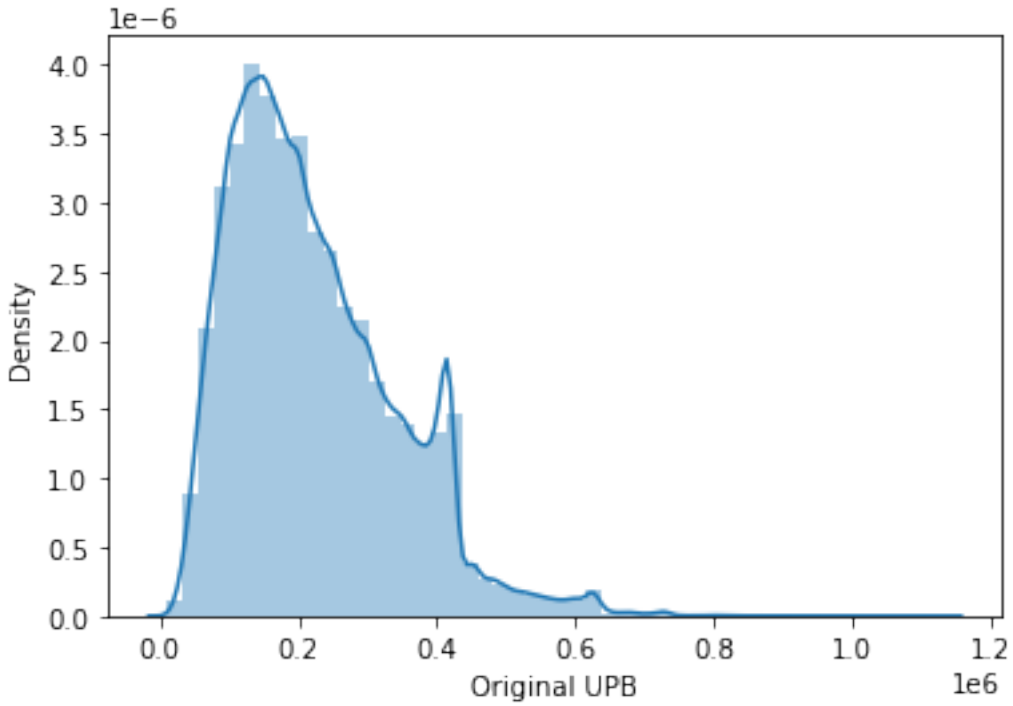
[51]:
```
# Clean Original UPB
training_dat['Original UPB'].describe()
```

[51]:
```
count    3.963370e+05
mean     2.225630e+05
std      1.189133e+05
min      1.100000e+04
25%      1.310000e+05
50%      2.000000e+05
75%      2.960000e+05
max      1.129000e+06
Name: Original UPB, dtype: float64
```

[52]:
```
sns.distplot(training_dat['Original UPB'])
```

```
FutureWarning: `distplot` is a deprecated function and will be removed in a
future version. Please adapt your code to use either `displot` (a figure-level
function with similar flexibility) or `histplot` (an axes-level function for
histograms).
```

[52]:
```
<AxesSubplot:xlabel='Original UPB', ylabel='Density'>
```



[53]:
```
scaler_0_1_upb = QuantileTransformer(output_distribution='normal')
training_dat2['Original UPB'] = scaler_0_1_upb.
 ↪fit_transform(training_dat[['Original UPB']])
training_dat2.head()
```

[53]:

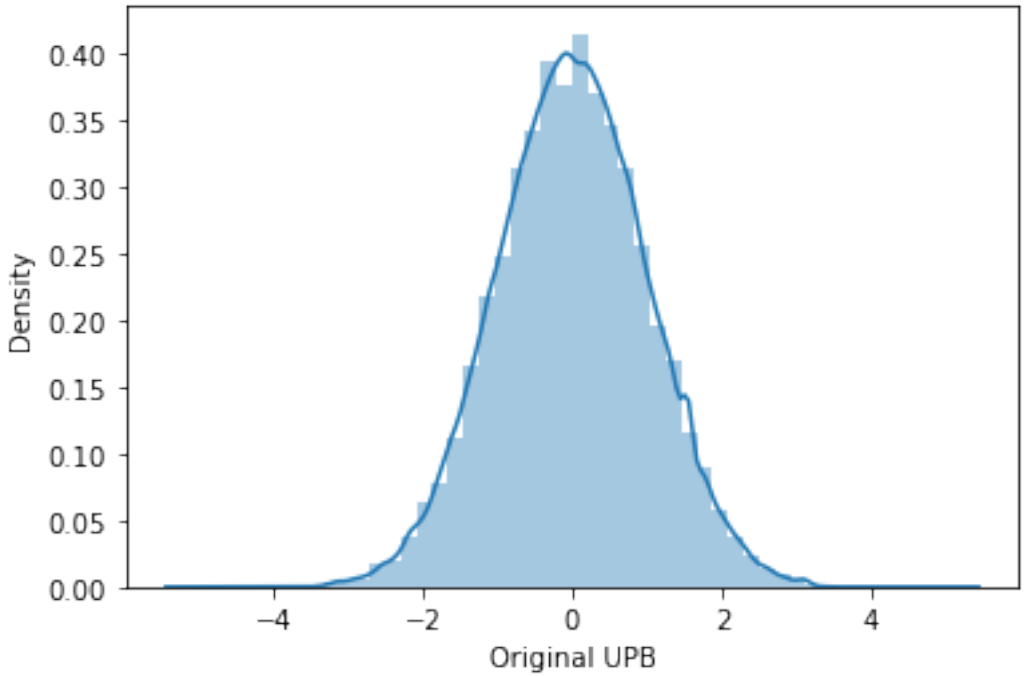
	Loan Sequence Number	Loan Deliquency Within Year	Credit Score	First Time Homebuyer Flag	MortgageInsuranceFlag	Units_1	Units_2	Units_3	Units_4	OccupancyStatus_I	OccupancyStatus_P	OccupancyStatus_S
0	F110Q1000008	False	1.315958	False	0	1	0	0	0	0	1	0
1	F110Q1000064	False	2.225823	False	0	1	0	0	0	0	1	0
2	F110Q1000072	False	0.451469	False	0	1	0	0	0	0	1	0
3	F110Q1000080	False	-1.733071	False	0	1	0	0	0	0	1	0
4	F110Q1000096	False	1.033647	False	0	1	0	0	0	0	1	0

```
4        0                   0                   1                   0

   Original Combined Loan-to-Value (CLTV)  \
0                               -1.606755
1                               -0.979511
2                                0.376283
3                                0.376283
4                                0.056486

   Original Debt-to-Income (DTI) Ratio  Original UPB
0                            -1.252988     -0.605955
1                            -0.599937     -1.446104
2                             0.139710     -0.350817
3                            -1.252988      1.545927
4                            -1.684464     -1.550085
```

[56]:
```
sns.distplot(training_dat2['Original UPB'])
```

```
FutureWarning: `distplot` is a deprecated function and will be removed in a
future version. Please adapt your code to use either `displot` (a figure-level
function with similar flexibility) or `histplot` (an axes-level function for
histograms).
```

[56]:
```
<AxesSubplot:xlabel='Original UPB', ylabel='Density'>
```



[57]:
```
scaler_dct['Original UPB'] = scaler_0_1_upb
```

[58]:
```
# Clean Original Loan-to-Value (LTV)
training_dat['Original Loan-to-Value (LTV)'].describe()
```

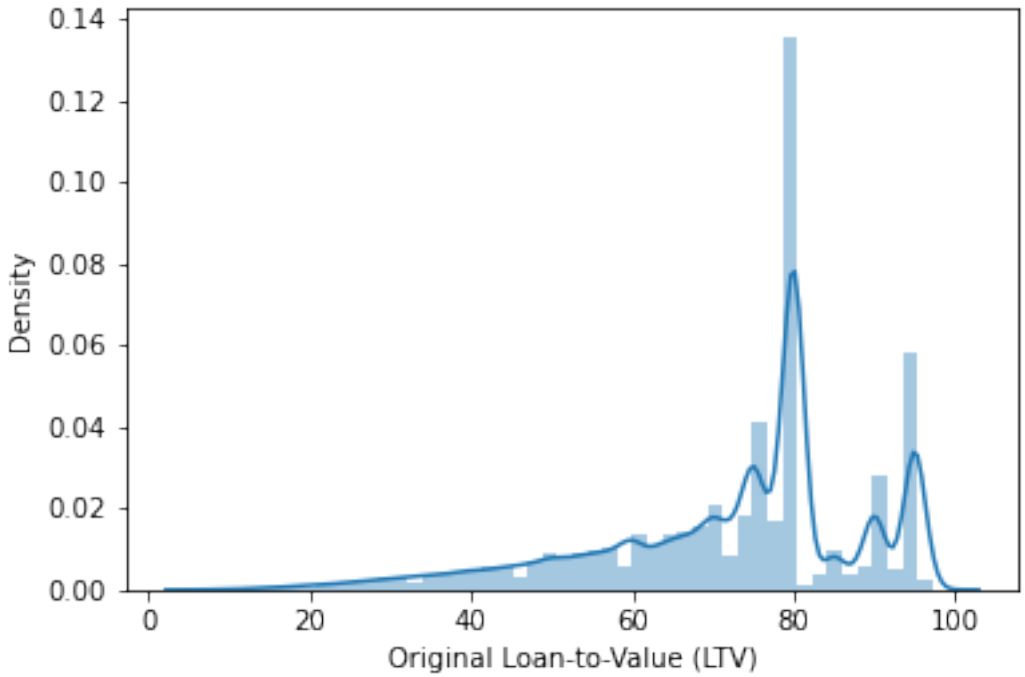
[58]:
```
count    396337.000000
mean         72.258210
std          17.734249
min           6.000000
25%          63.000000
50%          77.000000
75%          80.000000
max         999.000000
Name: Original Loan-to-Value (LTV), dtype: float64
```

[59]:
```
tf = training_dat['Original Loan-to-Value (LTV)'] == 999
training_dat.loc[tf, 'Original Loan-to-Value (LTV)'] = np.nan
```

[60]:
```
sns.distplot(training_dat['Original Loan-to-Value (LTV)'])
```

```
FutureWarning: `distplot` is a deprecated function and will be removed in a
future version. Please adapt your code to use either `displot` (a figure-level
function with similar flexibility) or `histplot` (an axes-level function for
histograms).
```

[60]:
```
<AxesSubplot:xlabel='Original Loan-to-Value (LTV)', ylabel='Density'>
```



[61]:
```
scaler_0_1_ltv = QuantileTransformer(output_distribution='normal')
training_dat2['Original Loan-to-Value (LTV)'] = scaler_0_1_ltv.
 ↪fit_transform(training_dat[['Original Loan-to-Value (LTV)']])
training_dat2.head()
```

[61]:

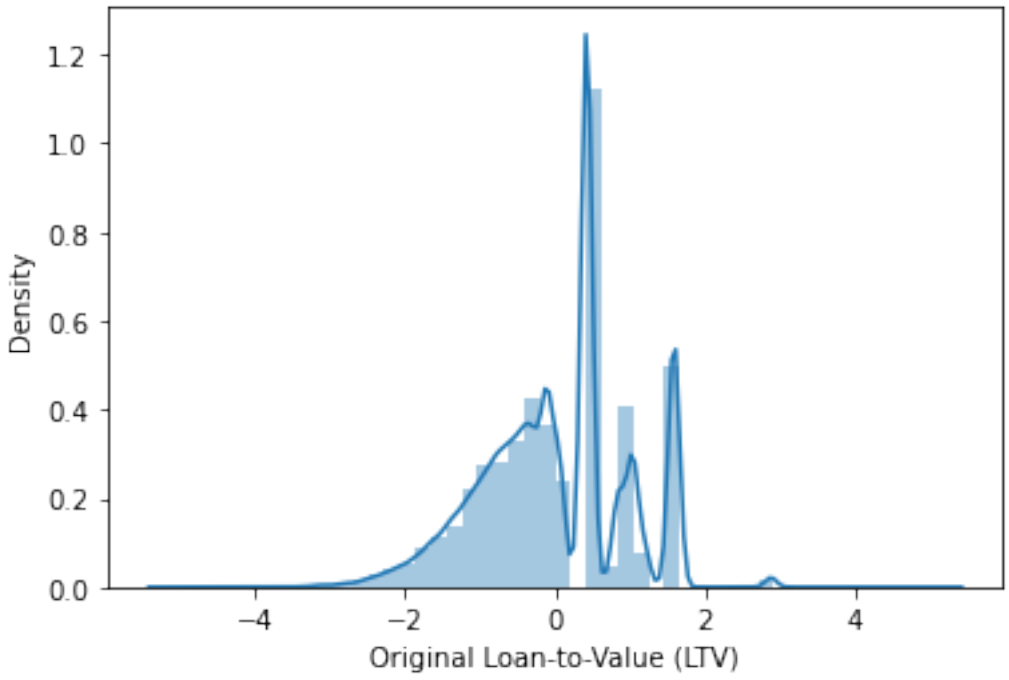
	Loan Sequence Number	Loan Deliquency Within Year	Credit Score	First Time Homebuyer Flag	MortgageInsuranceFlag	Units_1	Units_2	Units_3	Units_4	OccupancyStatus_I	OccupancyStatus_P	OccupancyStatus_S	Original Combined Loan-to-Value (CLTV)	Original Debt-to-Income (DTI) Ratio	Original UPB	Original Loan-to-Value (LTV)
0	F110Q1000008	False	1.315958	False	0	1	0	0	0	0	1	0	-1.606755	-1.252988	-0.605955	-1.588771
1	F110Q1000064	False	2.225823	False	0	1	0	0	0	0	1	0	-0.979511	-0.599937	-1.446104	-0.949547
2	F110Q1000072	False	0.451469	False	0	1	0	0	0	0	1	0	0.376283	0.139710	-0.350817	0.421111
3	F110Q1000080	False	-1.733071	False	0	1	0	0	0	0	1	0	0.376283	-1.252988	1.545927	-1.057485
4	F110Q1000096	False	1.033647	False	0	1	0	0	0	0	1	0	0.056486	-1.684464	-1.550085	0.095492

[62]:
```
scaler_dct['Original Loan-to-Value (LTV)'] = scaler_0_1_ltv
```

[63]:
```
sns.distplot(training_dat2['Original Loan-to-Value (LTV)'])
```

```
FutureWarning: `distplot` is a deprecated function and will be removed in a
future version. Please adapt your code to use either `displot` (a figure-level
function with similar flexibility) or `histplot` (an axes-level function for
histograms).
```

[63]:
```
<AxesSubplot:xlabel='Original Loan-to-Value (LTV)', ylabel='Density'>
```



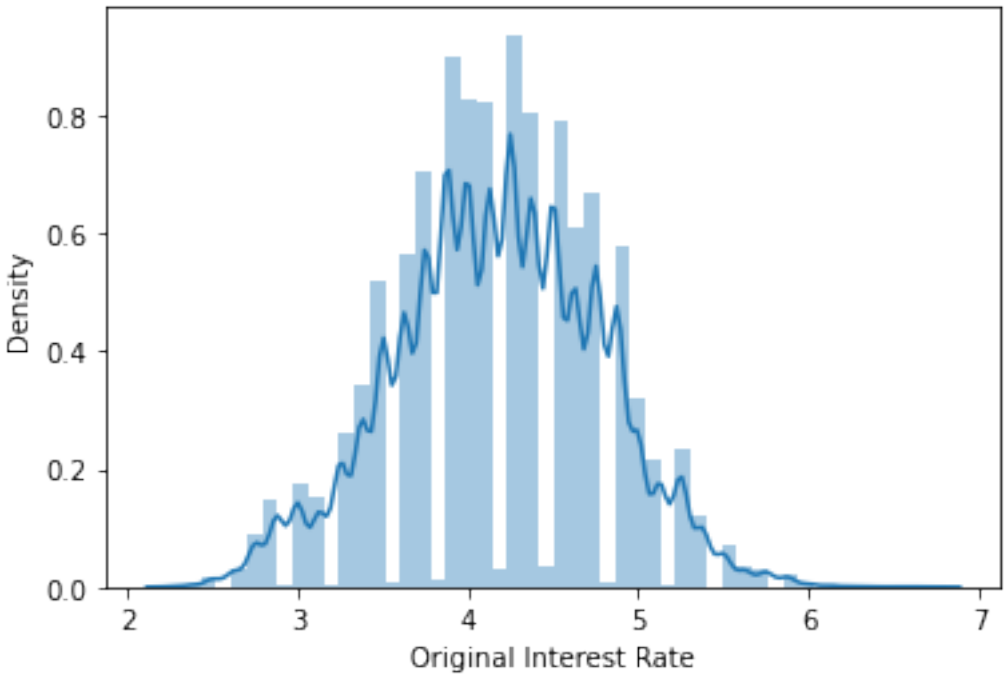
[64]:
```
# Clean Original Interest Rate
training_dat['Original Interest Rate'].describe()
```

[64]:
```
count    396337.000000
mean          4.172227
std           0.607312
min           2.250000
25%           3.750000
50%           4.125000
75%           4.625000
max           6.750000
Name: Original Interest Rate, dtype: float64
```

[65]:
```
sns.distplot(training_dat['Original Interest Rate'])
```

```
FutureWarning: `distplot` is a deprecated function and will be removed in a
future version. Please adapt your code to use either `displot` (a figure-level
function with similar flexibility) or `histplot` (an axes-level function for
histograms).
```

[65]: `<AxesSubplot:xlabel='Original Interest Rate', ylabel='Density'>`



[66]:
```
scaler_0_1_intrestrate = QuantileTransformer(output_distribution='normal')
training_dat2['Original Interest Rate'] = scaler_0_1_intrestrate.
 ↪fit_transform(training_dat[['Original Interest Rate']])
training_dat2.head()
```

[66]:

	Loan Sequence Number	Loan Deliquency Within Year	Credit Score	\
0	F110Q1000008	False	1.315958	
1	F110Q1000064	False	2.225823	
2	F110Q1000072	False	0.451469	
3	F110Q1000080	False	-1.733071	
4	F110Q1000096	False	1.033647	

	First Time Homebuyer Flag	MortgageInsuranceFlag	Units_1	Units_2	Units_3	\
0	False	0	1	0	0	
1	False	0	1	0	0	
2	False	0	1	0	0	
3	False	0	1	0	0	
4	False	0	1	0	0	

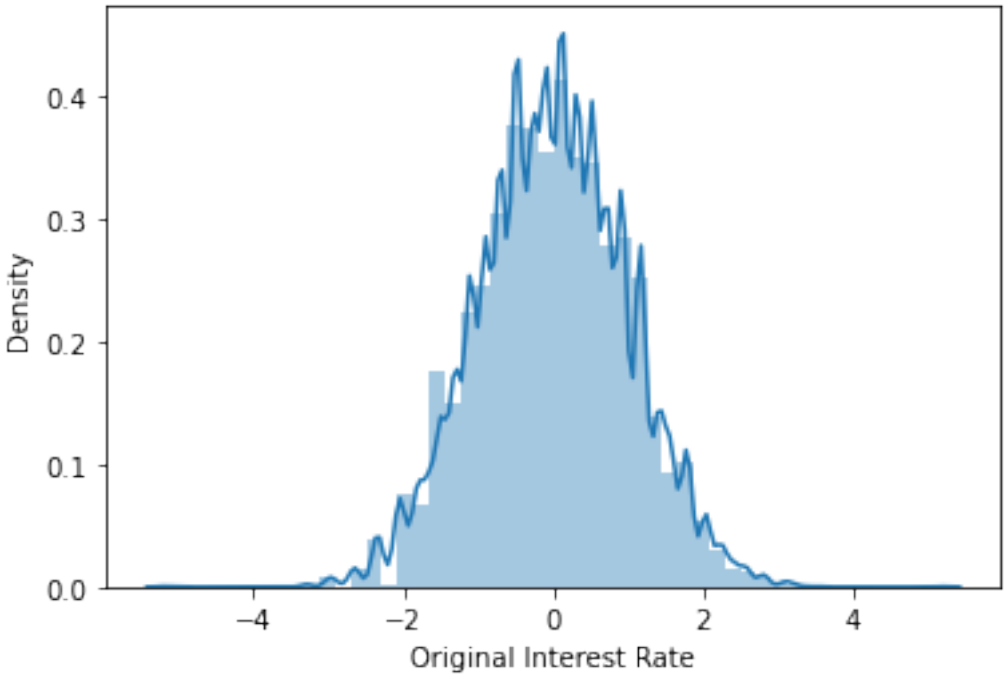
```
   Units_4  OccupancyStatus_I  OccupancyStatus_P  OccupancyStatus_S  \
0        0                  0                  1                  0
1        0                  0                  1                  0
2        0                  0                  1                  0
3        0                  0                  1                  0
4        0                  0                  1                  0

   Original Combined Loan-to-Value (CLTV)  \
0                               -1.606755
1                               -0.979511
2                                0.376283
3                                0.376283
4                                0.056486

   Original Debt-to-Income (DTI) Ratio  Original UPB  \
0                            -1.252988     -0.605955
1                            -0.599937     -1.446104
2                             0.139710     -0.350817
3                            -1.252988      1.545927
4                            -1.684464     -1.550085

   Original Loan-to-Value (LTV)  Original Interest Rate
0                     -1.588771                1.161976
1                     -0.949547                1.786156
2                      0.421111                1.418119
3                     -1.057485                1.786156
4                      0.095492                1.418119
```

[67]:
```
scaler_dct['Original Interest Rate'] = scaler_0_1_intrestrate
```

[68]:
```
sns.distplot(training_dat2['Original Interest Rate'])
```

```
FutureWarning: `distplot` is a deprecated function and will be removed in a
future version. Please adapt your code to use either `displot` (a figure-level
function with similar flexibility) or `histplot` (an axes-level function for
histograms).
```

[68]:
```
<AxesSubplot:xlabel='Original Interest Rate', ylabel='Density'>
```



[69]:
```
# Transform Channel
training_dat['Channel'].value_counts()
```

```
UserWarning: value_counts defaulting to pandas implementation.
```

[69]:
```
R    227989
C    125527
B     42821
Name: Channel, dtype: int64
```

[70]:
```
dum_df = pd.get_dummies(training_dat['Channel'],prefix='Channel')
for col in list(dum_df.columns):
    training_dat2[col] = dum_df[col]
training_dat2.head()
```

```
UserWarning: `get_dummies` on non-DataFrame defaulting to pandas implementation.
UserWarning: Distributing <class 'pandas.core.frame.DataFrame'> object. This may
take some time.
```

[70]:

	Loan Sequence Number	Loan Deliquency Within Year	Credit Score	\
0	F110Q1000008	False	1.315958	
1	F110Q1000064	False	2.225823	
2	F110Q1000072	False	0.451469	
3	F110Q1000080	False	-1.733071	
4	F110Q1000096	False	1.033647	

```
  First Time Homebuyer Flag  MortgageInsuranceFlag  Units_1  Units_2  Units_3  \
0                     False                      0        1        0        0
1                     False                      0        1        0        0
2                     False                      0        1        0        0
3                     False                      0        1        0        0
4                     False                      0        1        0        0

   Units_4  OccupancyStatus_I  OccupancyStatus_P  OccupancyStatus_S  \
0        0                  0                  1                  0
1        0                  0                  1                  0
2        0                  0                  1                  0
3        0                  0                  1                  0
4        0                  0                  1                  0

   Original Combined Loan-to-Value (CLTV)  \
0                               -1.606755
1                               -0.979511
2                                0.376283
3                                0.376283
4                                0.056486

   Original Debt-to-Income (DTI) Ratio  Original UPB  \
0                            -1.252988     -0.605955
1                            -0.599937     -1.446104
2                             0.139710     -0.350817
3                            -1.252988      1.545927
4                            -1.684464     -1.550085

   Original Loan-to-Value (LTV)  Original Interest Rate  Channel_B  Channel_C  \
0                     -1.588771                1.161976          0          0
1                     -0.949547                1.786156          0          0
2                      0.421111                1.418119          0          0
3                     -1.057485                1.786156          0          0
4                      0.095492                1.418119          0          0

   Channel_R
0          1
1          1
2          1
3          1
4          1
```

[71]:
```
training_dat['Prepayment Penalty Mortgage (PPM) Flag'].value_counts()
```

```
UserWarning: value_counts defaulting to pandas implementation.
```

[71]:
```
N    395950
Name: Prepayment Penalty Mortgage (PPM) Flag, dtype: int64
```

[72]:
```
# Only ns, so exclude this
training_dat['Prepayment Penalty Mortgage (PPM) Flag'].unique()
```

[72]:
```
array(['N', nan], dtype=object)
```

[73]:
```
# Almost no non FRMs, so exclude
training_dat['Amortization Type (Formerly Product Type)'].value_counts()
```

[73]:
```
FRM    396280
_          57
Name: Amortization Type (Formerly Product Type), dtype: int64
```

[74]:
```
# Tranform Property Type
training_dat['Property Type'].value_counts()
```

[74]:
```
SF    266922
PU    100295
CO     27220
MH      1369
CP       531
Name: Property Type, dtype: int64
```

[75]:
```
dum_df = pd.get_dummies(training_dat['Property Type'],prefix='PropertyType')
for col in list(dum_df.columns):
    training_dat2[col] = dum_df[col]
training_dat2.head()
```

```
UserWarning: `get_dummies` on non-DataFrame defaulting to pandas implementation.
UserWarning: Distributing <class 'pandas.core.frame.DataFrame'> object. This may
take some time.
```

[75]:

	Loan Sequence Number	Loan Deliquency Within Year	Credit Score	\
0	F110Q1000008	False	1.315958	
1	F110Q1000064	False	2.225823	
2	F110Q1000072	False	0.451469	
3	F110Q1000080	False	-1.733071	
4	F110Q1000096	False	1.033647	

	First Time Homebuyer Flag	MortgageInsuranceFlag	Units_1	Units_2	Units_3	\
0	False	0	1	0	0	
1	False	0	1	0	0	
2	False	0	1	0	0	
3	False	0	1	0	0	
4	False	0	1	0	0	

```
   Units_4  OccupancyStatus_I  …  Original Loan-to-Value (LTV)  \
0        0                  0  …                     -1.588771
1        0                  0  …                     -0.949547
2        0                  0  …                      0.421111
3        0                  0  …                     -1.057485
4        0                  0  …                      0.095492

   Original Interest Rate  Channel_B  Channel_C  Channel_R  PropertyType_CO  \
0                1.161976          0          0          1                0
1                1.786156          0          0          1                0
2                1.418119          0          0          1                0
3                1.786156          0          0          1                0
4                1.418119          0          0          1                0

   PropertyType_CP  PropertyType_MH  PropertyType_PU  PropertyType_SF
0                0                0                0                1
1                0                0                0                1
2                0                0                0                1
3                0                0                0                1
4                0                0                0                1

[5 rows x 25 columns]
```

[76]:
```
# Transform Loan Purpose
training_dat['Loan Purpose'].value_counts()
```

```
UserWarning: value_counts defaulting to pandas implementation.
```

[76]:
```
P    186799
N    114318
C     95220
Name: Loan Purpose, dtype: int64
```

[77]:
```
dum_df = pd.get_dummies(training_dat['Loan Purpose'],prefix='LoanPurpose')
for col in list(dum_df.columns):
    training_dat2[col] = dum_df[col]
training_dat2.head()
```

```
UserWarning: `get_dummies` on non-DataFrame defaulting to pandas implementation.
UserWarning: Distributing <class 'pandas.core.frame.DataFrame'> object. This may
take some time.
```

[77]:
```
  Loan Sequence Number  Loan Deliquency Within Year  Credit Score  \
0         F110Q1000008                        False      1.315958
1         F110Q1000064                        False      2.225823
2         F110Q1000072                        False      0.451469
```

```
3         F110Q1000080                        False     -1.733071
4         F110Q1000096                        False      1.033647

  First Time Homebuyer Flag  MortgageInsuranceFlag  Units_1  Units_2  Units_3  \
0                     False                      0        1        0        0
1                     False                      0        1        0        0
2                     False                      0        1        0        0
3                     False                      0        1        0        0
4                     False                      0        1        0        0

   Units_4  OccupancyStatus_I  …  Channel_C  Channel_R  PropertyType_CO  \
0        0                  0  …          0          1                0
1        0                  0  …          0          1                0
2        0                  0  …          0          1                0
3        0                  0  …          0          1                0
4        0                  0  …          0          1                0

   PropertyType_CP  PropertyType_MH  PropertyType_PU  PropertyType_SF  \
0                0                0                0                1
1                0                0                0                1
2                0                0                0                1
3                0                0                0                1
4                0                0                0                1

   LoanPurpose_C  LoanPurpose_N  LoanPurpose_P
0              0              1              0
1              1              0              0
2              0              1              0
3              0              1              0
4              0              1              0

[5 rows x 28 columns]
```

[78]:
```
# Clean Original Loan Term
# Going to go with one hot encoding, since most loans are 30,25,20,15 etc years
training_dat['Original Loan Term'].value_counts().head(10)
```

```
UserWarning: value_counts defaulting to pandas implementation.
```

[78]:
```
360    292292
180     74346
240     18955
120      6234
300      2428
144       282
156       225
168       153
```

```
324       134
336       126
Name: Original Loan Term, dtype: int64
```

[79]:
```
term_flags = [360,180,240,120,300]
for term in term_flags:
    col_name = 'LoanTerm_' + str(term)
    training_dat2[col_name] = 0
    tf = training_dat['Original Loan Term'] == term
    training_dat2.loc[tf, col_name] = 1
training_dat2.head()
```

[79]:

	Loan Sequence Number	Loan Deliquency Within Year	Credit Score	First Time Homebuyer Flag	MortgageInsuranceFlag	Units_1	Units_2	Units_3	Units_4	OccupancyStatus_I	…	PropertyType_PU	PropertyType_SF	LoanPurpose_C	LoanPurpose_N	LoanPurpose_P	LoanTerm_360	LoanTerm_180	LoanTerm_240	LoanTerm_120	LoanTerm_300
0	F110Q1000008	False	1.315958	False	0	1	0	0	0	0	…	0	1	0	1	0	1	0	0	0	0
1	F110Q1000064	False	2.225823	False	0	1	0	0	0	0	…	0	1	1	0	0	1	0	0	0	0
2	F110Q1000072	False	0.451469	False	0	1	0	0	0	0	…	0	1	0	1	0	1	0	0	0	0
3	F110Q1000080	False	-1.733071	False	0	1	0	0	0	0	…	0	1	0	1	0	1	0	0	0	0
4	F110Q1000096	False	1.033647	False	0	1	0	0	0	0	…	0	1	0	1	0	1	0	0	0	0

```
[5 rows x 33 columns]
```

[80]:
```
training_dat2['LoanTerm_300'].sum()
```

[80]:
```
2428
```

[81]:
```
# Will also try continuous feature
scaler_0_1_loanterm = QuantileTransformer(output_distribution='normal')
training_dat2['Original Loan Term'] = scaler_0_1_loanterm.
 ↪fit_transform(training_dat[['Original Loan Term']])
training_dat2.head()
```

[81]:

	Loan Sequence Number	Loan Deliquency Within Year	Credit Score	First Time Homebuyer Flag	MortgageInsuranceFlag	Units_1	Units_2	Units_3	Units_4	OccupancyStatus_I	…	PropertyType_SF	LoanPurpose_C	LoanPurpose_N	LoanPurpose_P	LoanTerm_360	LoanTerm_180	LoanTerm_240	LoanTerm_120	LoanTerm_300	Original Loan Term
0	F110Q1000008	False	1.315958	False	0	1	0	0	0	0	…	1	0	1	0	1	0	0	0	0	0.334851
1	F110Q1000064	False	2.225823	False	0	1	0	0	0	0	…	1	1	0	0	1	0	0	0	0	0.334851
2	F110Q1000072	False	0.451469	False	0	1	0	0	0	0	…	1	0	1	0	1	0	0	0	0	0.334851
3	F110Q1000080	False	-1.733071	False	0	1	0	0	0	0	…	1	0	1	0	1	0	0	0	0	0.334851
4	F110Q1000096	False	1.033647	False	0	1	0	0	0	0	…	1	0	1	0	1	0	0	0	0	0.334851

```
[5 rows x 34 columns]
```

[82]:
```
scaler_dct['Original Loan Term'] = scaler_0_1_loanterm
```

[83]:
```
# Transform Number of Borrowers
training_dat['Number of Borrowers'].value_counts()
```

```
UserWarning: value_counts defaulting to pandas implementation.
```

[83]:
```
2    213369
1    182968
Name: Number of Borrowers, dtype: int64
```

[84]:
```
training_dat2['OneBorrower'] = 0
tf = training_dat['Number of Borrowers'] == 1
training_dat2.loc[tf, 'OneBorrower'] = 1
```

[85]:
```
training_dat2.head()
```

[85]:

	Loan Sequence Number	Loan Deliquency Within Year	Credit Score	First Time Homebuyer Flag	MortgageInsuranceFlag	Units_1	Units_2	Units_3	Units_4	OccupancyStatus_I	…	LoanPurpose_C	LoanPurpose_N	LoanPurpose_P	LoanTerm_360	LoanTerm_180	LoanTerm_240	LoanTerm_120
0	F110Q1000008	False	1.315958	False	0	1	0	0	0	0	…	0	1	0	1	0	0	0
1	F110Q1000064	False	2.225823	False	0	1	0	0	0	0	…	1	0	0	1	0	0	0
2	F110Q1000072	False	0.451469	False	0	1	0	0	0	0	…	0	1	0	1	0	0	0
3	F110Q1000080	False	-1.733071	False	0	1	0	0	0	0	…	0	1	0	1	0	0	0
4	F110Q1000096	False	1.033647	False	0	1	0	0	0	0	…	0	1	0	1	0	0	0

```
   LoanTerm_300  Original Loan Term  OneBorrower
0             0            0.334851            0
1             0            0.334851            1
2             0            0.334851            0
3             0            0.334851            1
4             0            0.334851            1

[5 rows x 35 columns]
```

[86]:
```
# Transform Program Indicator
training_dat['Program Indicator'].value_counts()
```

```
UserWarning: value_counts defaulting to pandas implementation.
```

[86]:
```
9    393058
H      3279
Name: Program Indicator, dtype: int64
```

[87]:
```
training_dat2['AffordableProgramFlag'] = 0
tf = training_dat['Program Indicator'] == 'H'
training_dat2.loc[tf,'AffordableProgramFlag'] = 1
training_dat2.head()
```

[87]:
```
  Loan Sequence Number  Loan Deliquency Within Year  Credit Score  \
0         F110Q1000008                        False      1.315958
1         F110Q1000064                        False      2.225823
2         F110Q1000072                        False      0.451469
3         F110Q1000080                        False     -1.733071
4         F110Q1000096                        False      1.033647

  First Time Homebuyer Flag  MortgageInsuranceFlag  Units_1  Units_2  Units_3  \
0                     False                      0        1        0        0
1                     False                      0        1        0        0
2                     False                      0        1        0        0
3                     False                      0        1        0        0
4                     False                      0        1        0        0

   Units_4  OccupancyStatus_I  …  LoanPurpose_N  LoanPurpose_P  \
0        0                  0  …              1              0
1        0                  0  …              0              0
2        0                  0  …              1              0
3        0                  0  …              1              0
4        0                  0  …              1              0

   LoanTerm_360  LoanTerm_180  LoanTerm_240  LoanTerm_120  LoanTerm_300  \
0             1             0             0             0             0
1             1             0             0             0             0
```

```
2              1              0              0              0              0
3              1              0              0              0              0
4              1              0              0              0              0

   Original Loan Term  OneBorrower  AffordableProgramFlag
0            0.334851            0                      0
1            0.334851            1                      0
2            0.334851            0                      0
3            0.334851            1                      0
4            0.334851            1                      0

[5 rows x 36 columns]
```

[88]:
```
# Exclude Property Valuation Method (2017 onwards only)
# Exclude Interest Only (I/O) Indicator nans only
training_dat['Interest Only (I/O) Indicator']
```

[88]:
```
0        NaN
1        NaN
2        NaN
3        NaN
4        NaN
          ..
396333   NaN
396334   NaN
396335   NaN
396336   NaN
396337   NaN
Name: Interest Only (I/O) Indicator, Length: 396337, dtype: float64
```

[89]:
```
training_dat2.columns
```

[89]:
```
Index(['Loan Sequence Number', 'Loan Deliquency Within Year', 'Credit Score',
       'First Time Homebuyer Flag', 'MortgageInsuranceFlag', 'Units_1',
       'Units_2', 'Units_3', 'Units_4', 'OccupancyStatus_I',
       'OccupancyStatus_P', 'OccupancyStatus_S',
       'Original Combined Loan-to-Value (CLTV)',
       'Original Debt-to-Income (DTI) Ratio', 'Original UPB',
       'Original Loan-to-Value (LTV)', 'Original Interest Rate', 'Channel_B',
       'Channel_C', 'Channel_R', 'PropertyType_CO', 'PropertyType_CP',
       'PropertyType_MH', 'PropertyType_PU', 'PropertyType_SF',
       'LoanPurpose_C', 'LoanPurpose_N', 'LoanPurpose_P', 'LoanTerm_360',
       'LoanTerm_180', 'LoanTerm_240', 'LoanTerm_120', 'LoanTerm_300',
       'Original Loan Term', 'OneBorrower', 'AffordableProgramFlag'],
      dtype='object')
```

[90]:
```
len(training_dat2.columns)
```

[90]: 36

[91]:
```
tf1 = training_dat2['Loan Deliquency Within Year'] == True
tf2 = training_dat2['Loan Deliquency Within Year'] == False
training_dat2.loc[tf1, 'Loan Deliquency Within Year'] = 1
training_dat2.loc[tf2, 'Loan Deliquency Within Year'] = 0
```

[92]:
```
tf1 = training_dat2['First Time Homebuyer Flag'] == True
tf2 = training_dat2['First Time Homebuyer Flag'] == False
training_dat2.loc[tf1, 'First Time Homebuyer Flag'] = 1
training_dat2.loc[tf2, 'First Time Homebuyer Flag'] = 0
```

[93]:
```
training_dat2.isnull().sum()
```

[93]:
```
Loan Sequence Number                        0
Loan Deliquency Within Year                 0
Credit Score                               22
First Time Homebuyer Flag                  215276
MortgageInsuranceFlag                       0
Units_1                                     0
Units_2                                     0
Units_3                                     0
Units_4                                     0
OccupancyStatus_I                           0
OccupancyStatus_P                           0
OccupancyStatus_S                           0
Original Combined Loan-to-Value (CLTV)     11
Original Debt-to-Income (DTI) Ratio        37
Original UPB                                0
Original Loan-to-Value (LTV)               11
Original Interest Rate                      0
Channel_B                                   0
Channel_C                                   0
Channel_R                                   0
PropertyType_CO                             0
PropertyType_CP                             0
PropertyType_MH                             0
PropertyType_PU                             0
PropertyType_SF                             0
LoanPurpose_C                               0
LoanPurpose_N                               0
LoanPurpose_P                               0
LoanTerm_360                                0
LoanTerm_180                                0
LoanTerm_240                                0
LoanTerm_120                                0
LoanTerm_300                                0
```

```
Original Loan Term                    0
OneBorrower                           0
AffordableProgramFlag                 0
dtype: int64
```

[94]:
```
# Get rid of first time homebuyer flag, and then drop all nulls
del training_dat2['First Time Homebuyer Flag']
training_dat2.dropna(inplace=True)
```

[95]:
```
len(training_dat2)
```

[95]:
```
396267
```

[96]:
```
training_dat2.to_pickle('./PROCESSED/training_dat2.pkl')
```

```
UserWarning: `DataFrame.to_pickle` defaulting to pandas implementation.
UserWarning: Distributing <class 'pandas.core.frame.DataFrame'> object. This may
take some time.
```

[97]:
```
with open("./PROCESSED/scalers.pkl","wb") as f:
    pickle.dump(scaler_dct, f)
```

[ ]:
```

```