Black_Friday_Notebook_2

September 16, 2019

0.1 Black Friday Notebook 2

[1]:
```
# IST 718 Big Data Analytics, Final Project: Black Friday
# Notebook Name: Black Friday Notebook #2
# Notebook Contents: EDA and Linear Regression on Purchase Amount
# Group Members: Arielle Dortch, Lauren Foltz, Stephen Clark Washington III,␣
 ↪Inga Fleishhacker
# Written for Jupyter Notebook using Python 3.5

# Data is from https://www.kaggle.com/mehdidag/black-friday
```

[2]:
```
# Import Libraries

import numpy as np #arrays and math functions
import scipy.stats
import pandas as pd #data frame operations

import matplotlib
import matplotlib.pyplot as pp #2D plotting (note: sometimes people import as␣
 ↪plt)
from IPython import display
from ipywidgets import interact, widgets

%matplotlib inline

import os
import re
import mailbox
import csv
import warnings
warnings.filterwarnings("ignore")

from scipy.stats import uniform  # for training-and-test split
import statsmodels.api as sm  #R-like model specification (note: sometimes␣
 ↪people import as smf)
import statsmodels.formula.api as smf

import seaborn as sns # Provides Trellis and Small Multiple Plotting
```

[3]:
```
# Optional: Get current working directory
#os.getcwd()
```

[4]:
```
# Read in Data, name it BF, and look at it

BF = pd.read_csv('BlackFriday.csv')

print("\nContents of Black Friday data frame ---------------")
BF
```

```
Contents of Black Friday data frame ---------------
```

[4]:

	User_ID	Product_ID	Gender	Age	Occupation	City_Category	\
0	1000001	P00069042	F	0-17	10	A	
1	1000001	P00248942	F	0-17	10	A	
2	1000001	P00087842	F	0-17	10	A	
3	1000001	P00085442	F	0-17	10	A	
4	1000002	P00285442	M	55+	16	C	
5	1000003	P00193542	M	26-35	15	A	
6	1000004	P00184942	M	46-50	7	B	
7	1000004	P00346142	M	46-50	7	B	
8	1000004	P0097242	M	46-50	7	B	
9	1000005	P00274942	M	26-35	20	A	
10	1000005	P00251242	M	26-35	20	A	
11	1000005	P00014542	M	26-35	20	A	
12	1000005	P00031342	M	26-35	20	A	
13	1000005	P00145042	M	26-35	20	A	
14	1000006	P00231342	F	51-55	9	A	
15	1000006	P00190242	F	51-55	9	A	
16	1000006	P0096642	F	51-55	9	A	
17	1000006	P00058442	F	51-55	9	A	
18	1000007	P00036842	M	36-45	1	B	
19	1000008	P00249542	M	26-35	12	C	
20	1000008	P00220442	M	26-35	12	C	
21	1000008	P00156442	M	26-35	12	C	
22	1000008	P00213742	M	26-35	12	C	
23	1000008	P00214442	M	26-35	12	C	
24	1000008	P00303442	M	26-35	12	C	
25	1000009	P00135742	M	26-35	17	C	
26	1000009	P00039942	M	26-35	17	C	
27	1000009	P00161442	M	26-35	17	C	
28	1000009	P00078742	M	26-35	17	C	
29	1000010	P00085942	F	36-45	1	B	
…	…	…	…	…	…	…	
537547	1004733	P00244042	M	18-25	18	C	
537548	1004734	P00111042	M	51-55	1	B	

537549	1004734	P00345842	M	51-55	1	B
537550	1004735	P00278242	M	46-50	3	C
537551	1004735	P00313442	M	46-50	3	C
537552	1004735	P0098642	M	46-50	3	C
537553	1004735	P00119342	M	46-50	3	C
537554	1004735	P00114042	M	46-50	3	C
537555	1004735	P00135142	M	46-50	3	C
537556	1004736	P00194542	M	18-25	20	A
537557	1004736	P00175242	M	18-25	20	A
537558	1004736	P00101942	M	18-25	20	A
537559	1004736	P00109142	M	18-25	20	A
537560	1004736	P00084842	M	18-25	20	A
537561	1004736	P00078142	M	18-25	20	A
537562	1004736	P00146742	M	18-25	20	A
537563	1004736	P00154642	M	18-25	20	A
537564	1004736	P00117442	M	18-25	20	A
537565	1004736	P00051142	M	18-25	20	A
537566	1004736	P00048742	M	18-25	20	A
537567	1004736	P00157542	M	18-25	20	A
537568	1004736	P00250642	M	18-25	20	A
537569	1004736	P00023142	M	18-25	20	A
537570	1004736	P00162442	M	18-25	20	A
537571	1004737	P00221442	M	36-45	16	C
537572	1004737	P00193542	M	36-45	16	C
537573	1004737	P00111142	M	36-45	16	C
537574	1004737	P00345942	M	36-45	16	C
537575	1004737	P00285842	M	36-45	16	C
537576	1004737	P00118242	M	36-45	16	C

	Stay_In_Current_City_Years	Marital_Status	Product_Category_1	\
0	2	0	3	
1	2	0	1	
2	2	0	12	
3	2	0	12	
4	4+	0	8	
5	3	0	1	
6	2	1	1	
7	2	1	1	
8	2	1	1	
9	1	1	8	
10	1	1	5	
11	1	1	8	
12	1	1	8	
13	1	1	1	
14	1	0	5	
15	1	0	4	
16	1	0	2	

17		1	0	5
18		1	1	1
19		4+	1	1
20		4+	1	5
21		4+	1	8
22		4+	1	8
23		4+	1	8
24		4+	1	1
25		0	0	6
26		0	0	8
27		0	0	5
28		0	0	5
29		4+	1	2
…	…	…	…	
537547		1	0	1
537548		1	1	15
537549		1	1	2
537550		3	0	1
537551		3	0	5
537552		3	0	6
537553		3	0	10
537554		3	0	5
537555		3	0	13
537556		1	1	8
537557		1	1	2
537558		1	1	8
537559		1	1	8
537560		1	1	8
537561		1	1	8
537562		1	1	1
537563		1	1	8
537564		1	1	5
537565		1	1	8
537566		1	1	5
537567		1	1	8
537568		1	1	11
537569		1	1	5
537570		1	1	1
537571		1	0	1
537572		1	0	1
537573		1	0	1
537574		1	0	8
537575		1	0	5
537576		1	0	5

	Product_Category_2	Product_Category_3	Purchase
0	NaN	NaN	8370

1	6.0	14.0	15200
2	NaN	NaN	1422
3	14.0	NaN	1057
4	NaN	NaN	7969
5	2.0	NaN	15227
6	8.0	17.0	19215
7	15.0	NaN	15854
8	16.0	NaN	15686
9	NaN	NaN	7871
10	11.0	NaN	5254
11	NaN	NaN	3957
12	NaN	NaN	6073
13	2.0	5.0	15665
14	8.0	14.0	5378
15	5.0	NaN	2079
16	3.0	4.0	13055
17	14.0	NaN	8851
18	14.0	16.0	11788
19	5.0	15.0	19614
20	14.0	NaN	8584
21	NaN	NaN	9872
22	NaN	NaN	9743
23	NaN	NaN	5982
24	8.0	14.0	11927
25	8.0	NaN	16662
26	NaN	NaN	5887
27	14.0	NaN	6973
28	8.0	14.0	5391
29	4.0	8.0	16352
…	…	…	…
537547	2.0	15.0	11543
537548	NaN	NaN	20924
537549	8.0	14.0	13082
537550	NaN	NaN	11658
537551	6.0	8.0	6863
537552	8.0	NaN	16415
537553	13.0	NaN	18526
537554	14.0	NaN	7099
537555	16.0	NaN	578
537556	14.0	NaN	2183
537557	NaN	NaN	12724
537558	17.0	NaN	7796
537559	17.0	NaN	7770
537560	16.0	NaN	5940
537561	16.0	NaN	7834
537562	13.0	14.0	11508
537563	NaN	NaN	6074

```
537564                14.0                    NaN      7084
537565                 NaN                    NaN      7934
537566                 NaN                    NaN      5350
537567                14.0                    NaN      1994
537568                 NaN                    NaN      5930
537569                14.0                    NaN      7042
537570                16.0                    NaN     15491
537571                 2.0                    5.0     11852
537572                 2.0                    NaN     11664
537573                15.0                   16.0     19196
537574                15.0                    NaN      8043
537575                 NaN                    NaN      7172
537576                 8.0                    NaN      6875

[537577 rows x 12 columns]
```

[5]:
```
# Look at columns
BF.columns
```

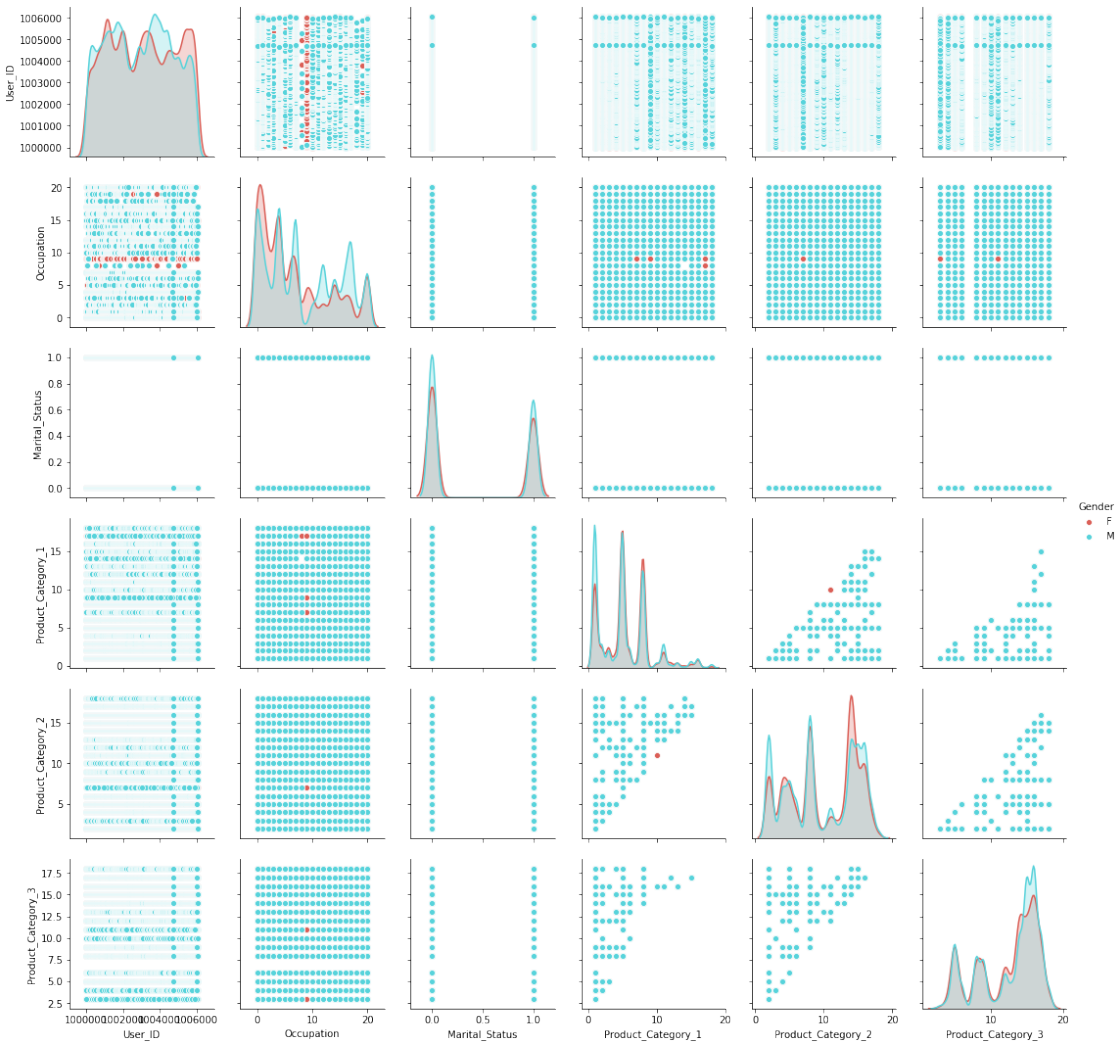
[5]:
```
Index(['User_ID', 'Product_ID', 'Gender', 'Age', 'Occupation', 'City_Category',
       'Stay_In_Current_City_Years', 'Marital_Status', 'Product_Category_1',
       'Product_Category_2', 'Product_Category_3', 'Purchase'],
      dtype='object')
```

[6]:
```
# Look for missing values
BF.isna().sum()

# missing some values under Product Category 2 and 3
```

[6]:
```
User_ID                           0
Product_ID                        0
Gender                            0
Age                               0
Occupation                        0
City_Category                     0
Stay_In_Current_City_Years        0
Marital_Status                    0
Product_Category_1                0
Product_Category_2           166986
Product_Category_3           373299
Purchase                          0
dtype: int64
```

[7]:
```
# Generate pair plots (takes a while)
sns.pairplot(BF[BF.columns[:11]], hue='Gender', palette='hls')
```

[7]:
```
<seaborn.axisgrid.PairGrid at 0x2444eaabd30>
```



[8]:
```
# Look at a portion of the dataframe
BF.loc[0:100:20]
```

[8]:

	User_ID	Product_ID	Gender	Age	Occupation	City_Category	\
0	1000001	P00069042	F	0-17	10	A	
20	1000008	P00220442	M	26-35	12	C	
40	1000010	P00087242	F	36-45	1	B	
60	1000015	P00333042	M	26-35	7	A	
80	1000018	P0094142	F	18-25	3	B	
100	1000022	P00195942	M	18-25	15	A	

	Stay_In_Current_City_Years	Marital_Status	Product_Category_1	\
0	2	0	3	
20	4+	1	5	
40	4+	1	14	

```
60                           1                0                   5
80                           3                0                   4
100                         4+                0                   3

     Product_Category_2  Product_Category_3  Purchase
0                   NaN                 NaN      8370
20                 14.0                 NaN      8584
40                  NaN                 NaN      7947
60                  8.0                 NaN      3594
80                  5.0                 NaN       697
100                 4.0                 NaN     10681
```

[9]:
```
# Import more Libraries

import math
from scipy import stats as st
from sklearn import linear_model
from sklearn.model_selection import train_test_split
from sklearn import metrics
from matplotlib import pyplot as plt

import sklearn
from sklearn.linear_model import LinearRegression
from sklearn.preprocessing import scale
from collections import Counter
from sklearn.model_selection import train_test_split
```

[10]:
```
# Visualize Customers
# https://www.kaggle.com/sriharshaatyam/
 ↪eda-and-predictions-using-various-ml-models

# Set up grid for how plots will show up:
fig, axes = plt.subplots(nrows=5, ncols=2,figsize=(15,15))
fig.tight_layout()
fig.subplots_adjust(wspace=0.4,hspace=0.4)

# Aggregate by User ID:
BF['User_ID'].groupby(BF['Gender']).nunique().plot(kind='bar',ax=axes[0,0])
BF['User_ID'].groupby(BF['Age']).nunique().plot(kind='bar',ax=axes[0,1])
BF['User_ID'].groupby(BF['Occupation']).nunique().plot(kind='bar',ax=axes[1,0])
BF['User_ID'].groupby(BF['City_Category']).nunique().
 ↪plot(kind='bar',ax=axes[1,1])
BF['User_ID'].groupby(BF['Stay_In_Current_City_Years']).nunique().
 ↪plot(kind='bar',ax=axes[2,0])
BF['User_ID'].groupby(BF['Marital_Status']).nunique().
 ↪plot(subplots=True,kind='bar',ax=axes[2,1])
```

```
BF['User_ID'].groupby(BF['Product_Category_1']).nunique().
 ↪plot(subplots=True,kind='bar',ax=axes[3,0])
BF['User_ID'].groupby(BF['Product_Category_2']).nunique().
 ↪plot(subplots=True,kind='bar',ax=axes[3,1])
BF['User_ID'].groupby(BF['Product_Category_3']).nunique().
 ↪plot(kind='bar',ax=axes[4,0])
```

[10]: <matplotlib.axes._subplots.AxesSubplot at 0x2445f97b748>

[11]:
```
# Set up Parameters for plots
# https://stackoverflow.com/questions/332289/
 ↪how-do-you-change-the-size-of-figures-drawn-with-matplotlib
```

```
%matplotlib inline
from pylab import rcParams
rcParams['figure.figsize'] = 5, 4
sns.set_style('whitegrid')
```

[12]:
```
# Generate Correlation Matrix
print("Correlation:")
BF.corr()
```

```
Correlation:
```

[12]:

	User_ID	Occupation	Marital_Status	Product_Category_1	\
User_ID	1.000000	-0.023024	0.018732	0.003687	
Occupation	-0.023024	1.000000	0.024691	-0.008114	
Marital_Status	0.018732	0.024691	1.000000	0.020546	
Product_Category_1	0.003687	-0.008114	0.020546	1.000000	
Product_Category_2	0.001471	-0.000031	0.015116	0.540423	
Product_Category_3	0.004045	0.013452	0.019452	0.229490	
Purchase	0.005389	0.021104	0.000129	-0.314125	

	Product_Category_2	Product_Category_3	Purchase
User_ID	0.001471	0.004045	0.005389
Occupation	-0.000031	0.013452	0.021104
Marital_Status	0.015116	0.019452	0.000129
Product_Category_1	0.540423	0.229490	-0.314125
Product_Category_2	1.000000	0.543544	-0.209973
Product_Category_3	0.543544	1.000000	-0.022257
Purchase	-0.209973	-0.022257	1.000000

0.2 Change Data Types

[13]:
```
# Check data types
print('Initial Data Types:')
print()
print(BF.dtypes)
```

```
Initial Data Types:

User_ID                       int64
Product_ID                   object
Gender                       object
Age                          object
Occupation                    int64
City_Category                object
Stay_In_Current_City_Years   object
Marital_Status                int64
```

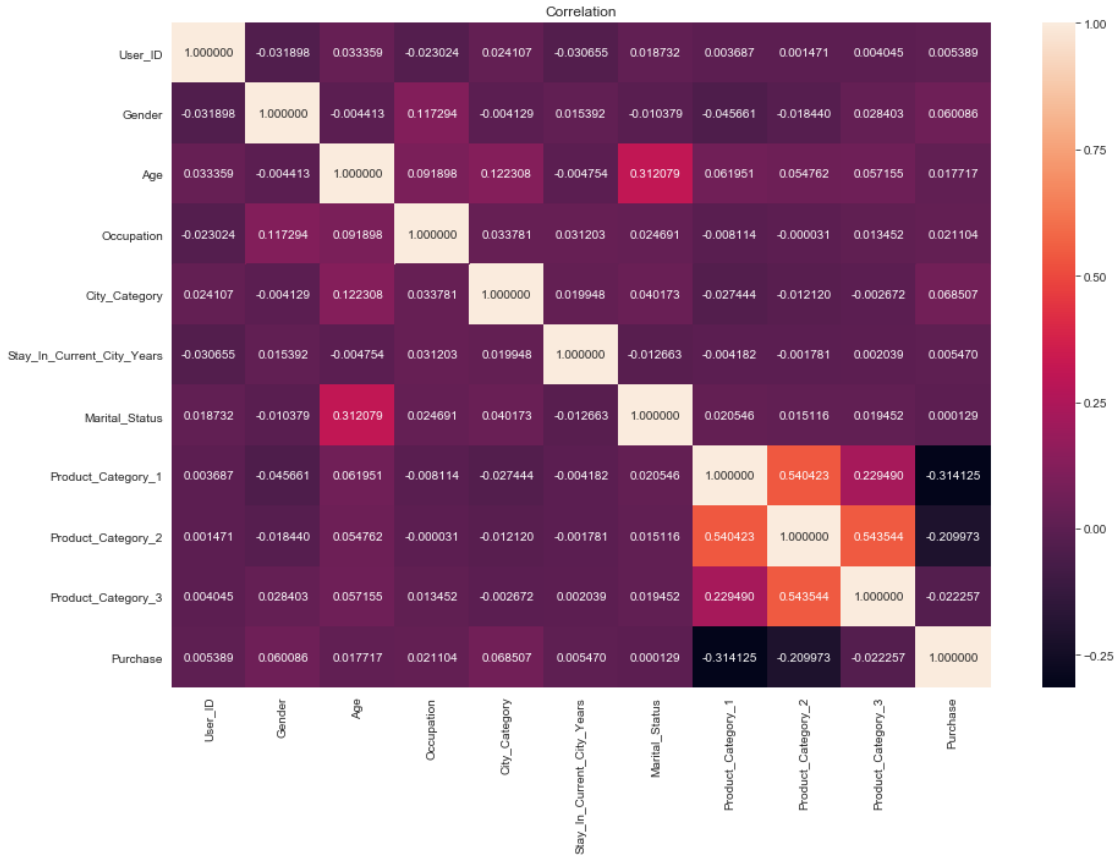
```
Product_Category_1              int64
Product_Category_2            float64
Product_Category_3            float64
Purchase                        int64
dtype: object
```

[14]:
```
# Change some data types from object to int
BF["Age"] = BF["Age"].astype("category").cat.codes
BF["Gender"] = BF["Gender"].astype("category").cat.codes
BF["Stay_In_Current_City_Years"] = BF["Stay_In_Current_City_Years"].
 ↪astype("category").cat.codes
BF["City_Category"] = BF["City_Category"].astype("category").cat.codes

#Check data types
print('New Data Types:')
print()
print(BF.dtypes)
```

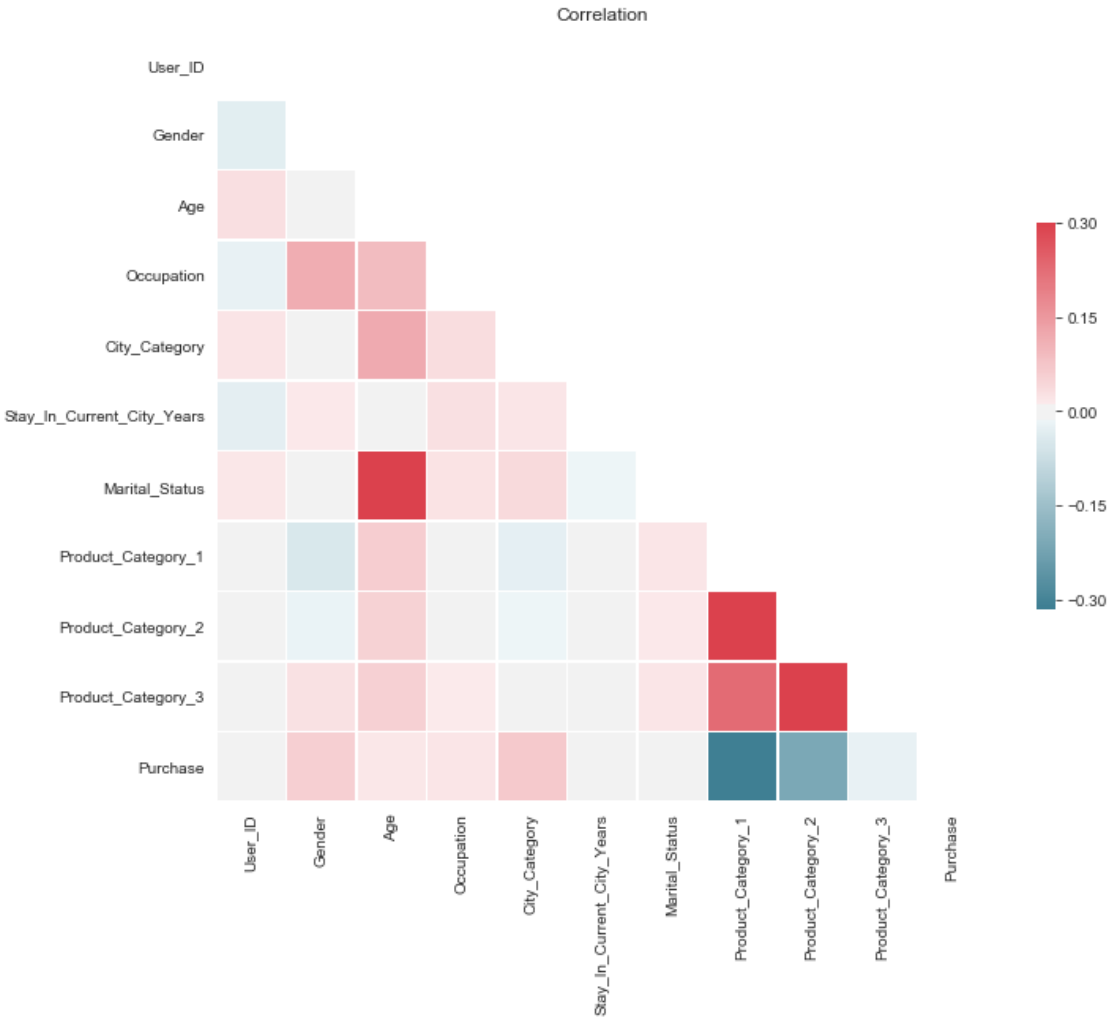
```
New Data Types:

User_ID                        int64
Product_ID                    object
Gender                          int8
Age                             int8
Occupation                     int64
City_Category                   int8
Stay_In_Current_City_Years      int8
Marital_Status                 int64
Product_Category_1             int64
Product_Category_2           float64
Product_Category_3           float64
Purchase                       int64
dtype: object
```

[15]:
```
# Visualize Correlation Matrix

#adust parameters
rcParams['figure.figsize'] = 15,10 #adjust figure size

#plot
sns.heatmap(BF.corr(), annot=True, fmt="f")
plt.title("Correlation")
plt.show()

# Looks like Marital Status correlates somewhat to Age, which makes sense
# Purchase and product category 1 also have a high correlation.
# City Category is .069
```

```
# Gender is .060
```



[16]:
```
# Another Correlation Matrix Visualization
corr = BF.corr()

# Generate a mask for the upper triangle
mask = np.zeros_like(corr, dtype=np.bool)
mask[np.triu_indices_from(mask)] = True

rcParams['figure.figsize'] = 11,9 # set up figure size

# Generate a custom diverging colormap
cmap = sns.diverging_palette(220, 10, as_cmap=True)

# Draw the heatmap with the mask and correct aspect ratio
sns.heatmap(corr, mask=mask, cmap=cmap, vmax=.3, center=0,
            square=True, linewidths=.5, cbar_kws={"shrink": .5})
plt.title("Correlation")
plt.show()
```



0.3 OLS Regression

[17]:
```
from sklearn.model_selection import train_test_split

train, test = train_test_split(BF, test_size=0.2)
```

0.4 Model 1

[18]:
```
# Model 1 (Train)
# Use all variables except Product ID and User ID

my_model = str('Purchase ~ Gender + Age + Occupation + City_Category +␣
 ↪Stay_In_Current_City_Years + Marital_Status + Product_Category_1 +␣
 ↪Product_Category_2 + Product_Category_3')
```

```
# fit the model to the training set
train_model_fit = smf.ols(my_model, data = train).fit()

# summary of model fit to the training set
print(train_model_fit.summary())

# R^2 is ~17%
# Marital_Status crosses zero and p-value is too high, so probably shouldn't␣
 ↪use it.
```

```
                            OLS Regression Results
==============================================================================
Dep. Variable:                Purchase   R-squared:                       0.169
Model:                            OLS   Adj. R-squared:                  0.169
Method:                 Least Squares   F-statistic:                     2978.
Date:                Fri, 14 Jun 2019   Prob (F-statistic):               0.00
Time:                        10:45:22   Log-Likelihood:            -1.2978e+06
No. Observations:              131624   AIC:                         2.596e+06
Df Residuals:                  131614   BIC:                         2.596e+06
Df Model:                           9
Covariance Type:            nonrobust
==============================================================================
==============
                                 coef    std err          t      P>|t|
[0.025      0.975]
------------------------------------------------------------------------------
--------------
Intercept                   1.172e+04     58.654    199.817      0.000
1.16e+04    1.18e+04
Gender                       331.5013     30.918     10.722      0.000
270.902     392.100
Age                          132.2998     10.176     13.002      0.000
112.356     152.244
Occupation                     5.8655      1.996      2.939      0.003
1.954       9.777
City_Category                423.2499     16.900     25.045      0.000
390.127     456.373
Stay_In_Current_City_Years    27.4571      9.930      2.765      0.006
7.994      46.920
Marital_Status               -44.6751     27.402     -1.630      0.103
-98.382       9.032
Product_Category_1          -828.7416      5.757   -143.951      0.000
-840.025    -817.458
Product_Category_2            26.3041      3.804      6.915      0.000
18.849      33.759
Product_Category_3            70.5053      3.697     19.069      0.000
63.258      77.752
```

```
==============================================================================
Omnibus:                     1561.102   Durbin-Watson:                   2.001
Prob(Omnibus):                  0.000   Jarque-Bera (JB):             1666.762
Skew:                           0.247   Prob(JB):                         0.00
Kurtosis:                       3.246   Cond. No.                         85.4
==============================================================================

Warnings:
[1] Standard Errors assume that the covariance matrix of the errors is correctly
specified.
```

[19]:
```
# Model 1 (Test)
# Use all variables except Product ID and User ID

my_model = str('Purchase ~ Gender + Age + Occupation + City_Category +␣
 ↪Stay_In_Current_City_Years + Marital_Status + Product_Category_1 +␣
 ↪Product_Category_2 + Product_Category_3')

# fit the model to the training set
test_model_fit = smf.ols(my_model, data = test).fit()

# summary of model fit to the training set
print(test_model_fit.summary())

# R^2 is ~16%
# Marital_Status crosses zero and p-value is too high, so probably shouldn't␣
 ↪use it.
# Also, p-value for Stay_In_Current_City_Years has a p-value too high
# Occupation P-value is a bit high
```

```
                            OLS Regression Results
==============================================================================
Dep. Variable:               Purchase   R-squared:                       0.170
Model:                            OLS   Adj. R-squared:                  0.170
Method:                 Least Squares   F-statistic:                     744.0
Date:                Fri, 14 Jun 2019   Prob (F-statistic):               0.00
Time:                        10:45:25   Log-Likelihood:            -3.2194e+05
No. Observations:               32654   AIC:                         6.439e+05
Df Residuals:                   32644   BIC:                         6.440e+05
Df Model:                           9
Covariance Type:            nonrobust
================================================================================
==============
                                 coef    std err          t      P>|t|
[0.025      0.975]
--------------------------------------------------------------------------------
--------------
Intercept                   1.166e+04    118.366     98.533      0.000
```

```
1.14e+04    1.19e+04
Gender                        308.6199     61.888      4.987      0.000
187.317     429.923
Age                           146.0326     20.366      7.170      0.000
106.114     185.952
Occupation                     10.6095      3.974      2.670      0.008
2.821      18.398
City_Category                 427.4889     33.995     12.575      0.000
360.858     494.120
Stay_In_Current_City_Years     -7.0979     20.042     -0.354      0.723
-46.380      32.184
Marital_Status                  0.5239     54.977      0.010      0.992
-107.234     108.282
Product_Category_1           -828.6637     11.544    -71.785      0.000
-851.290    -806.037
Product_Category_2             22.7280      7.620      2.983      0.003
7.792      37.664
Product_Category_3             81.8566      7.445     10.995      0.000
67.264      96.449
==============================================================================
Omnibus:                      441.989   Durbin-Watson:                   2.018
Prob(Omnibus):                  0.000   Jarque-Bera (JB):              472.025
Skew:                           0.267   Prob(JB):                    3.17e-103
Kurtosis:                       3.250   Cond. No.                         86.1
==============================================================================

Warnings:
[1] Standard Errors assume that the covariance matrix of the errors is correctly
specified.
```

0.5 Model 2

[20]:
```
# Model 2 (train)
# Removed Marital Status, Stay in City, and Occupation

my_model = str('Purchase ~ Gender + Age + City_Category + Product_Category_1 +␣
 ↪Product_Category_2 + Product_Category_3')

# fit the model to the training set
train_model_fit = smf.ols(my_model, data = train).fit()

# summary of model fit to the training set
print(train_model_fit.summary())

#R^2 is still low at ~ 17%
```

```
                            OLS Regression Results
==============================================================================
```

```
Dep. Variable:                Purchase   R-squared:                       0.169
Model:                             OLS   Adj. R-squared:                  0.169
Method:                  Least Squares   F-statistic:                     4463.
Date:                 Fri, 14 Jun 2019   Prob (F-statistic):               0.00
Time:                         10:45:28   Log-Likelihood:            -1.2978e+06
No. Observations:              131624   AIC:                         2.596e+06
Df Residuals:                  131617   BIC:                         2.596e+06
Df Model:                           6
Covariance Type:            nonrobust
================================================================================
======
                           coef    std err          t      P>|t|      [0.025
0.975]
--------------------------------------------------------------------------------
------
Intercept             1.18e+04     54.778    215.340      0.000    1.17e+04
1.19e+04
Gender                343.0753     30.721     11.167      0.000     282.862
403.289
Age                   129.6795      9.637     13.456      0.000     110.790
148.568
City_Category         425.3181     16.889     25.183      0.000     392.216
458.420
Product_Category_1   -828.9620      5.757   -143.989      0.000    -840.246
-817.678
Product_Category_2     26.3161      3.804      6.918      0.000      18.860
33.772
Product_Category_3     70.5926      3.698     19.091      0.000      63.345
77.840
==============================================================================
Omnibus:                     1561.121   Durbin-Watson:                   2.001
Prob(Omnibus):                  0.000   Jarque-Bera (JB):             1667.111
Skew:                           0.247   Prob(JB):                         0.00
Kurtosis:                       3.246   Cond. No.                         70.9
==============================================================================

Warnings:
[1] Standard Errors assume that the covariance matrix of the errors is correctly
specified.
```

[21]:
```
# Model 2 (test)
my_model = str('Purchase ~ Gender + Age + City_Category + Product_Category_1 +␣
 ↪Product_Category_2 + Product_Category_3')

# fit the model to the training set
test_model_fit = smf.ols(my_model, data = test).fit()
```

```
# summary of model fit to the training set
print(test_model_fit.summary())

#R^2 is still low at ~ 16%
```

```
                            OLS Regression Results
==============================================================================
Dep. Variable:               Purchase   R-squared:                       0.170
Model:                            OLS   Adj. R-squared:                  0.170
Method:                 Least Squares   F-statistic:                     1115.
Date:                Fri, 14 Jun 2019   Prob (F-statistic):               0.00
Time:                        10:45:32   Log-Likelihood:            -3.2195e+05
No. Observations:               32654   AIC:                         6.439e+05
Df Residuals:                   32647   BIC:                         6.440e+05
Df Model:                           6
Covariance Type:            nonrobust
==============================================================================
                 coef    std err          t      P>|t|      [0.025      0.975]
------------------------------------------------------------------------------
Intercept   1.171e+04    110.375    106.076      0.000    1.15e+04    1.19e+04
Gender       326.5359     61.514      5.308      0.000     205.966     447.105
Age          150.7469     19.316      7.804      0.000     112.887     188.607
City_Category  429.8931     33.975     12.653      0.000     363.302     496.485
Product_Category_1  -828.7782     11.544    -71.792      0.000    -851.405    -806.151
Product_Category_2    22.6979      7.621      2.979      0.003       7.761      37.634
Product_Category_3    81.9934      7.445     11.013      0.000      67.401      96.586
==============================================================================
Omnibus:                      441.122   Durbin-Watson:                   2.017
Prob(Omnibus):                  0.000   Jarque-Bera (JB):              471.237
Skew:                           0.266   Prob(JB):                    4.70e-103
Kurtosis:                       3.251   Cond. No.                         71.2
==============================================================================

Warnings:
[1] Standard Errors assume that the covariance matrix of the errors is correctly
specified.
```

0.6 Change Data Types

[22]:
```
# Change Product Category 1 to data type category
train['Product_Category_1'] = train['Product_Category_1'].astype('category')
test['Product_Category_1'] = test['Product_Category_1'].astype('category')

# Check if change went through okay
print('New Train Data Types:')
print(train.dtypes)
print()
print('New Test Data Types:')
print()
print(test.dtypes)
```

```
New Train Data Types:
User_ID                       int64
Product_ID                   object
Gender                         int8
Age                            int8
Occupation                    int64
City_Category                  int8
Stay_In_Current_City_Years     int8
Marital_Status                int64
Product_Category_1         category
Product_Category_2          float64
Product_Category_3          float64
Purchase                      int64
dtype: object

New Test Data Types:

User_ID                       int64
Product_ID                   object
Gender                         int8
Age                            int8
Occupation                    int64
City_Category                  int8
Stay_In_Current_City_Years     int8
Marital_Status                int64
Product_Category_1         category
Product_Category_2          float64
Product_Category_3          float64
Purchase                      int64
dtype: object
```

0.7 Model 3

[23]:
```
# Model 3 (train)
# product category 1 has changed data type. Also, removed product category 2␣
 ↪and 3

my_model = str('Purchase ~ Gender + Age + City_Category + Product_Category_1')

# fit the model to the training set
train_model_fit = smf.ols(my_model, data = train).fit()

# summary of model fit to the training set
print(train_model_fit.summary())

#R^2 is much higher at ~63%
# Where is product category #1? I only see 2-18.
```

```
                            OLS Regression Results
==============================================================================
Dep. Variable:               Purchase   R-squared:                       0.629
Model:                            OLS   Adj. R-squared:                  0.629
Method:                 Least Squares   F-statistic:                 3.653e+04
Date:                Fri, 14 Jun 2019   Prob (F-statistic):               0.00
Time:                        10:45:41   Log-Likelihood:            -4.0577e+06
No. Observations:              430061   AIC:                         8.115e+06
Df Residuals:                  430040   BIC:                         8.116e+06
Df Model:                          20
Covariance Type:            nonrobust
================================================================================
============
                                 coef    std err          t      P>|t|      [0.025
0.975]
--------------------------------------------------------------------------------
------------
Intercept                   1.311e+04     16.247    806.971      0.000    1.31e+04
1.31e+04
Product_Category_1[T.2]  -2350.5990     23.899    -98.357      0.000   -2397.440
-2303.758
Product_Category_1[T.3]  -3494.4331     25.696   -135.991      0.000   -3544.796
-3444.070
Product_Category_1[T.4]  -1.125e+04     32.720   -343.879      0.000   -1.13e+04
-1.12e+04
Product_Category_1[T.5]  -7344.6368     12.718   -577.497      0.000   -7369.564
-7319.710
Product_Category_1[T.6]   2224.9852     25.532     87.145      0.000    2174.943
2275.027
Product_Category_1[T.7]   2730.8131     56.758     48.113      0.000    2619.569
2842.057
```

```
Product_Category_1[T.8]  -6108.9138     13.707   -445.662      0.000   -6135.780
-6082.048
Product_Category_1[T.9]   2250.5914    169.612     13.269      0.000    1918.158
2583.025
Product_Category_1[T.10]  5995.6055     48.550    123.494      0.000    5900.449
6090.762
Product_Category_1[T.11] -8896.1839     23.719   -375.064      0.000   -8942.673
-8849.695
Product_Category_1[T.12] -1.229e+04     55.110   -223.051      0.000   -1.24e+04
-1.22e+04
Product_Category_1[T.13] -1.288e+04     46.888   -274.759      0.000    -1.3e+04
-1.28e+04
Product_Category_1[T.14]  -347.2231     88.971     -3.903      0.000    -521.603
-172.843
Product_Category_1[T.15]  1179.8051     43.853     26.903      0.000    1093.854
1265.756
Product_Category_1[T.16]  1162.4373     35.632     32.623      0.000    1092.599
1232.276
Product_Category_1[T.17] -3598.2274    142.062    -25.328      0.000   -3876.666
-3319.789
Product_Category_1[T.18] -1.068e+04     61.680   -173.080      0.000   -1.08e+04
-1.06e+04
Gender                     -37.5437     10.807     -3.474      0.001     -58.725
-16.362
Age                         83.8734      3.461     24.231      0.000      77.089
90.658
City_Category              291.5573      6.136     47.513      0.000     279.530
303.584
==============================================================================
Omnibus:                    34248.420   Durbin-Watson:                   2.006
Prob(Omnibus):                  0.000   Jarque-Bera (JB):            52555.104
Skew:                          -0.629   Prob(JB):                         0.00
Kurtosis:                       4.162   Cond. No.                         119.
==============================================================================

Warnings:
[1] Standard Errors assume that the covariance matrix of the errors is correctly
specified.
```

[24]:
```
# Model 3 (test)
my_model = str('Purchase ~ Gender + Age + City_Category + Product_Category_1')

# fit the model to the training set
test_model_fit = smf.ols(my_model, data = test).fit()

# summary of model fit to the training set
print(test_model_fit.summary())
```

```
#R^2 is much higher at ~63%
#Gender now crosses 0 and the p-value is too high
```

```
                            OLS Regression Results
==============================================================================
Dep. Variable:               Purchase   R-squared:                       0.633
Model:                            OLS   Adj. R-squared:                  0.633
Method:                 Least Squares   F-statistic:                     9289.
Date:                Fri, 14 Jun 2019   Prob (F-statistic):               0.00
Time:                        10:45:43   Log-Likelihood:            -1.0143e+06
No. Observations:              107516   AIC:                         2.029e+06
Df Residuals:                  107495   BIC:                         2.029e+06
Df Model:                          20
Covariance Type:            nonrobust
================================================================================
============
                                 coef    std err          t      P>|t|      [0.025
0.975]
--------------------------------------------------------------------------------
------------
Intercept                   1.313e+04     32.395    405.236      0.000    1.31e+04
1.32e+04
Product_Category_1[T.2]  -2350.6620     47.863    -49.113      0.000   -2444.472
-2256.852
Product_Category_1[T.3]  -3508.4904     51.895    -67.607      0.000   -3610.204
-3406.777
Product_Category_1[T.4]  -1.129e+04     66.360   -170.157      0.000   -1.14e+04
-1.12e+04
Product_Category_1[T.5]  -7397.2074     25.316   -292.198      0.000   -7446.826
-7347.589
Product_Category_1[T.6]   2264.7671     51.082     44.336      0.000    2164.647
2364.887
Product_Category_1[T.7]   3070.5240    112.642     27.259      0.000    2849.748
3291.300
Product_Category_1[T.8]  -6139.8095     27.342   -224.555      0.000   -6193.400
-6086.219
Product_Category_1[T.9]    753.3503    330.561      2.279      0.023     105.456
1401.245
Product_Category_1[T.10]  6197.5625     97.850     63.337      0.000    6005.778
6389.347
Product_Category_1[T.11] -8943.6575     47.239   -189.328      0.000   -9036.245
-8851.070
Product_Category_1[T.12] -1.233e+04    111.565   -110.556      0.000   -1.26e+04
-1.21e+04
Product_Category_1[T.13] -1.293e+04     93.201   -138.688      0.000   -1.31e+04
-1.27e+04
```

```
Product_Category_1[T.14]  -851.5865    168.399     -5.057      0.000   -1181.646
-521.527
Product_Category_1[T.15]  1130.8217     88.737     12.744      0.000     956.898
1304.745
Product_Category_1[T.16]  1185.5919     70.800     16.746      0.000    1046.826
1324.358
Product_Category_1[T.17] -3337.4211    289.122    -11.543      0.000   -3904.097
-2770.745
Product_Category_1[T.18] -1.069e+04    124.387    -85.963      0.000   -1.09e+04
-1.04e+04
Gender                     -31.1438     21.596     -1.442      0.149     -73.473
11.185
Age                         93.7569      6.900     13.587      0.000      80.232
107.281
City_Category              285.8958     12.248     23.342      0.000     261.890
309.902
==============================================================================
Omnibus:                     8340.931   Durbin-Watson:                   2.010
Prob(Omnibus):                  0.000   Jarque-Bera (JB):            12737.644
Skew:                          -0.618   Prob(JB):                         0.00
Kurtosis:                       4.146   Cond. No.                         116.
==============================================================================

Warnings:
[1] Standard Errors assume that the covariance matrix of the errors is correctly
specified.
```

0.8 Model 4

[25]:
```
# Model 4 (train)
# product category 1 has changed data type. Also, removed product category 2
 ↪and 3

my_model = str('Purchase ~ Age + City_Category + Product_Category_1')

# fit the model to the training set
train_model_fit = smf.ols(my_model, data = train).fit()

# summary of model fit to the training set
print(train_model_fit.summary())

#R^2 is still ~63%
```

```
                            OLS Regression Results
==============================================================================
Dep. Variable:                Purchase   R-squared:                       0.629
Model:                            OLS   Adj. R-squared:                  0.629
Method:                 Least Squares   F-statistic:                 3.845e+04
```

```
Date:                Fri, 14 Jun 2019   Prob (F-statistic):               0.00
Time:                        10:45:47   Log-Likelihood:            -4.0577e+06
No. Observations:              430061   AIC:                         8.115e+06
Df Residuals:                  430041   BIC:                         8.116e+06
Df Model:                          19
Covariance Type:            nonrobust
==============================================================================
============
                                coef    std err          t      P>|t|      [0.025
0.975]
------------------------------------------------------------------------------
------------
Intercept                 1.308e+04     13.558    964.771      0.000    1.31e+04
1.31e+04
Product_Category_1[T.2]  -2348.2037     23.889    -98.297      0.000   -2395.025
-2301.382
Product_Category_1[T.3]  -3490.0212     25.665   -135.984      0.000   -3540.324
-3439.719
Product_Category_1[T.4]  -1.125e+04     32.690   -344.048      0.000   -1.13e+04
-1.12e+04
Product_Category_1[T.5]  -7340.8188     12.671   -579.356      0.000   -7365.653
-7315.985
Product_Category_1[T.6]   2226.7009     25.528     87.227      0.000    2176.667
2276.734
Product_Category_1[T.7]   2733.7954     56.752     48.171      0.000    2622.563
2845.028
Product_Category_1[T.8]  -6104.4875     13.648   -447.270      0.000   -6131.238
-6077.737
Product_Category_1[T.9]   2250.5542    169.614     13.269      0.000    1918.116
2582.992
Product_Category_1[T.10]  5997.5216     48.547    123.540      0.000    5902.370
6092.673
Product_Category_1[T.11] -8895.4887     23.719   -375.043      0.000   -8941.976
-8849.001
Product_Category_1[T.12] -1.228e+04     55.066   -223.090      0.000   -1.24e+04
-1.22e+04
Product_Category_1[T.13] -1.288e+04     46.879   -274.741      0.000    -1.3e+04
-1.28e+04
Product_Category_1[T.14]  -338.8713     88.939     -3.810      0.000    -513.190
-164.553
Product_Category_1[T.15]  1179.5113     43.854     26.896      0.000    1093.559
1265.463
Product_Category_1[T.16]  1165.0247     35.625     32.702      0.000    1095.201
1234.849
Product_Category_1[T.17] -3600.9876    142.062    -25.348      0.000   -3879.425
-3322.550
Product_Category_1[T.18] -1.068e+04     61.678   -173.121      0.000   -1.08e+04
-1.06e+04
```

```
Age                          83.8780      3.461     24.232      0.000      77.094
90.662
City_Category               291.7072      6.136     47.538      0.000     279.680
303.734
==============================================================================
Omnibus:                    34225.836   Durbin-Watson:                   2.006
Prob(Omnibus):                  0.000   Jarque-Bera (JB):            52499.323
Skew:                          -0.629   Prob(JB):                         0.00
Kurtosis:                       4.161   Cond. No.                         116.
==============================================================================

Warnings:
[1] Standard Errors assume that the covariance matrix of the errors is correctly
specified.
```

[26]:
```
# Model 4 (test)
my_model = str('Purchase ~ Age + City_Category + Product_Category_1')

# fit the model to the training set
test_model_fit = smf.ols(my_model, data = test).fit()

# summary of model fit to the training set
print(test_model_fit.summary())

#R^2 is still ~63%
# This is a good model, but should Age and City_Category also have data types␣
 ↪changed?
```

```
                            OLS Regression Results
==============================================================================
Dep. Variable:               Purchase   R-squared:                       0.633
Model:                            OLS   Adj. R-squared:                  0.633
Method:                 Least Squares   F-statistic:                     9777.
Date:                Fri, 14 Jun 2019   Prob (F-statistic):               0.00
Time:                        10:45:50   Log-Likelihood:            -1.0143e+06
No. Observations:              107516   AIC:                         2.029e+06
Df Residuals:                  107496   BIC:                         2.029e+06
Df Model:                          19
Covariance Type:            nonrobust
==============================================================================
============
                               coef    std err          t      P>|t|      [0.025
0.975]
------------------------------------------------------------------------------
------------
Intercept                   1.31e+04     26.999    485.265      0.000     1.3e+04
1.32e+04
Product_Category_1[T.2]  -2349.1212     47.851    -49.092      0.000   -2442.909
```

```
-2255.334
Product_Category_1[T.3]  -3504.5037     51.822    -67.626      0.000   -3606.074
-3402.934
Product_Category_1[T.4]  -1.129e+04     66.294   -170.262      0.000   -1.14e+04
-1.12e+04
Product_Category_1[T.5]  -7394.1165     25.225   -293.127      0.000   -7443.557
-7344.676
Product_Category_1[T.6]   2266.1963     51.072     44.372      0.000    2166.095
2366.298
Product_Category_1[T.7]   3072.6855    112.632     27.281      0.000    2851.928
3293.443
Product_Category_1[T.8]  -6136.1612     27.225   -225.387      0.000   -6189.522
-6082.801
Product_Category_1[T.9]    753.0402    330.562      2.278      0.023     105.142
1400.938
Product_Category_1[T.10]  6198.9565     97.846     63.354      0.000    6007.180
6390.733
Product_Category_1[T.11] -8943.1044     47.238   -189.321      0.000   -9035.690
-8850.519
Product_Category_1[T.12] -1.233e+04    111.455   -110.601      0.000   -1.25e+04
-1.21e+04
Product_Category_1[T.13] -1.292e+04     93.184   -138.686      0.000   -1.31e+04
-1.27e+04
Product_Category_1[T.14]  -843.1684    168.298     -5.010      0.000   -1173.031
-513.306
Product_Category_1[T.15]  1130.0386     88.736     12.735      0.000     956.118
1303.959
Product_Category_1[T.16]  1187.5437     70.787     16.776      0.000    1048.802
1326.285
Product_Category_1[T.17] -3338.9757    289.122    -11.549      0.000   -3905.650
-2772.301
Product_Category_1[T.18] -1.069e+04    124.384    -85.976      0.000   -1.09e+04
-1.05e+04
Age                         93.7497      6.900     13.586      0.000      80.225
107.274
City_Category              286.0914     12.247     23.359      0.000     262.087
310.096
==============================================================================
Omnibus:                     8337.589   Durbin-Watson:                   2.010
Prob(Omnibus):                  0.000   Jarque-Bera (JB):            12729.640
Skew:                          -0.618   Prob(JB):                         0.00
Kurtosis:                       4.146   Cond. No.                         113.
==============================================================================

Warnings:
[1] Standard Errors assume that the covariance matrix of the errors is correctly
specified.
```

0.9 Change more Data Types

[27]:
```
# Change Age and City Category to data type category
train['Age'] = train['Age'].astype('category')
train['City_Category'] = train['City_Category'].astype('category')

test['Age'] = test['Age'].astype('category')
test['City_Category'] = test['City_Category'].astype('category')

# Check if change went through okay
print('New Train Data Types:')
print(train.dtypes)
print()
print('New Test Data Types:')
print()
print(test.dtypes)
```

```
New Train Data Types:
User_ID                        int64
Product_ID                    object
Gender                         int8
Age                         category
Occupation                     int64
City_Category               category
Stay_In_Current_City_Years      int8
Marital_Status                 int64
Product_Category_1          category
Product_Category_2           float64
Product_Category_3           float64
Purchase                       int64
dtype: object

New Test Data Types:

User_ID                        int64
Product_ID                    object
Gender                         int8
Age                         category
Occupation                     int64
City_Category               category
Stay_In_Current_City_Years      int8
Marital_Status                 int64
Product_Category_1          category
Product_Category_2           float64
Product_Category_3           float64
Purchase                       int64
dtype: object
```

0.10 Model 5

[28]:
```
# Model 5 (train)
# Age and City Category have changed Data Types

my_model = str('Purchase ~ Age + City_Category + Product_Category_1')

# fit the model to the training set
train_model_fit = smf.ols(my_model, data = train).fit()

# summary of model fit to the training set
print(train_model_fit.summary())

#R^2 is still ~63%
# Where is Age category 0? Where is City category 0?
# Age group #1 crosses 0 and has a p-value that is too high
```

```
                            OLS Regression Results
==============================================================================
Dep. Variable:               Purchase   R-squared:                       0.630
Model:                            OLS   Adj. R-squared:                  0.630
Method:                 Least Squares   F-statistic:                 2.926e+04
Date:                Fri, 14 Jun 2019   Prob (F-statistic):               0.00
Time:                        10:45:59   Log-Likelihood:            -4.0575e+06
No. Observations:              430061   AIC:                         8.115e+06
Df Residuals:                  430035   BIC:                         8.115e+06
Df Model:                          25
Covariance Type:            nonrobust
================================================================================
============
                                 coef    std err          t      P>|t|      [0.025
0.975]
--------------------------------------------------------------------------------
------------
Intercept                  1.326e+04     30.452    435.461      0.000    1.32e+04
1.33e+04
Age[T.1]                     -35.0861     29.996     -1.170      0.242     -93.877
23.705
Age[T.2]                      36.2003     28.969      1.250      0.211     -20.578
92.979
Age[T.3]                     165.7951     29.809      5.562      0.000     107.370
224.220
Age[T.4]                     171.2240     32.244      5.310      0.000     108.026
234.422
Age[T.5]                     412.2283     32.987     12.497      0.000     347.574
476.883
Age[T.6]                     291.7475     36.543      7.984      0.000     220.125
363.370
```

```
City_Category[T.1]          139.5805     11.463     12.177      0.000     117.113
162.048
City_Category[T.2]          571.0156     12.399     46.053      0.000     546.714
595.317
Product_Category_1[T.2]  -2346.4978     23.886    -98.237      0.000   -2393.314
-2299.682
Product_Category_1[T.3]  -3491.9230     25.670   -136.030      0.000   -3542.236
-3441.610
Product_Category_1[T.4]  -1.125e+04     32.700   -343.947      0.000   -1.13e+04
-1.12e+04
Product_Category_1[T.5]  -7339.4897     12.668   -579.377      0.000   -7364.318
-7314.661
Product_Category_1[T.6]   2228.3125     25.519     87.320      0.000    2178.296
2278.329
Product_Category_1[T.7]   2735.2284     56.734     48.212      0.000    2624.032
2846.425
Product_Category_1[T.8]  -6103.1581     13.646   -447.262      0.000   -6129.903
-6076.413
Product_Category_1[T.9]   2245.2509    169.554     13.242      0.000    1912.929
2577.572
Product_Category_1[T.10]  5995.6316     48.532    123.540      0.000    5900.510
6090.753
Product_Category_1[T.11] -8893.7588     23.713   -375.065      0.000   -8940.235
-8847.283
Product_Category_1[T.12] -1.229e+04     55.058   -223.132      0.000   -1.24e+04
-1.22e+04
Product_Category_1[T.13] -1.288e+04     46.863   -274.862      0.000    -1.3e+04
-1.28e+04
Product_Category_1[T.14]  -342.4732     88.908     -3.852      0.000    -516.729
-168.217
Product_Category_1[T.15]  1179.1209     43.839     26.897      0.000    1093.198
1265.044
Product_Category_1[T.16]  1165.3126     35.613     32.722      0.000    1095.512
1235.113
Product_Category_1[T.17] -3597.2119    142.015    -25.330      0.000   -3875.557
-3318.867
Product_Category_1[T.18] -1.068e+04     61.657   -173.149      0.000   -1.08e+04
-1.06e+04
==============================================================================
Omnibus:                    34298.026   Durbin-Watson:                   2.006
Prob(Omnibus):                  0.000   Jarque-Bera (JB):            52652.996
Skew:                          -0.630   Prob(JB):                         0.00
Kurtosis:                       4.163   Cond. No.                         47.7
==============================================================================

Warnings:
[1] Standard Errors assume that the covariance matrix of the errors is correctly
specified.
```

[29]:
```
# Model 5 (test)
# Age and City Category have changed Data Types

my_model = str('Purchase ~ Age + City_Category + Product_Category_1')

# fit the model to the training set
test_model_fit = smf.ols(my_model, data = test).fit()

# summary of model fit to the training set
print(test_model_fit.summary())

#R^2 is still ~63%
# Age group #1 crosses 0 and has a p-value that is too high
```

```
                            OLS Regression Results
==============================================================================
Dep. Variable:               Purchase   R-squared:                       0.634
Model:                            OLS   Adj. R-squared:                  0.634
Method:                 Least Squares   F-statistic:                     7439.
Date:                Fri, 14 Jun 2019   Prob (F-statistic):               0.00
Time:                        10:46:02   Log-Likelihood:            -1.0142e+06
No. Observations:              107516   AIC:                         2.028e+06
Df Residuals:                  107490   BIC:                         2.029e+06
Df Model:                          25
Covariance Type:            nonrobust
================================================================================
============
                                 coef    std err          t      P>|t|      [0.025
0.975]
--------------------------------------------------------------------------------
------------
Intercept                  1.319e+04     61.255    215.344      0.000    1.31e+04
1.33e+04
Age[T.1]                     36.0936     60.335      0.598      0.550     -82.161
154.349
Age[T.2]                    163.5652     58.312      2.805      0.005      49.275
277.856
Age[T.3]                    300.2422     59.991      5.005      0.000     182.660
417.824
Age[T.4]                    236.9752     64.692      3.663      0.000     110.180
363.771
Age[T.5]                    507.7630     66.213      7.669      0.000     377.987
637.539
Age[T.6]                    465.9148     73.190      6.366      0.000     322.464
609.366
City_Category[T.1]          143.9487     22.892      6.288      0.000      99.081
188.816
```

```
City_Category[T.2]          568.2717     24.748     22.962      0.000     519.766
616.777
Product_Category_1[T.2]  -2347.4357     47.842    -49.066      0.000   -2441.206
-2253.666
Product_Category_1[T.3]  -3500.5152     51.840    -67.526      0.000   -3602.121
-3398.910
Product_Category_1[T.4]  -1.128e+04     66.322   -170.138      0.000   -1.14e+04
-1.12e+04
Product_Category_1[T.5]  -7393.0156     25.220   -293.135      0.000   -7442.447
-7343.584
Product_Category_1[T.6]   2266.8694     51.057     44.399      0.000    2166.798
2366.940
Product_Category_1[T.7]   3076.6485    112.601     27.323      0.000    2855.951
3297.346
Product_Category_1[T.8]  -6135.6097     27.219   -225.413      0.000   -6188.959
-6082.260
Product_Category_1[T.9]    752.4516    330.456      2.277      0.023     104.762
1400.141
Product_Category_1[T.10]  6192.1345     97.817     63.303      0.000    6000.415
6383.854
Product_Category_1[T.11] -8941.0213     47.229   -189.313      0.000   -9033.589
-8848.454
Product_Category_1[T.12] -1.232e+04    111.437   -110.585      0.000   -1.25e+04
-1.21e+04
Product_Category_1[T.13] -1.293e+04     93.155   -138.750      0.000   -1.31e+04
-1.27e+04
Product_Category_1[T.14]  -841.0222    168.246     -4.999      0.000   -1170.782
-511.262
Product_Category_1[T.15]  1125.8941     88.712     12.692      0.000     952.019
1299.769
Product_Category_1[T.16]  1187.2560     70.765     16.777      0.000    1048.558
1325.954
Product_Category_1[T.17] -3332.0573    289.036    -11.528      0.000   -3898.565
-2765.550
Product_Category_1[T.18] -1.069e+04    124.344    -85.967      0.000   -1.09e+04
-1.04e+04
==============================================================================
Omnibus:                     8373.787   Durbin-Watson:                   2.010
Prob(Omnibus):                  0.000   Jarque-Bera (JB):            12804.220
Skew:                          -0.620   Prob(JB):                         0.00
Kurtosis:                       4.150   Cond. No.                         46.5
==============================================================================

Warnings:
[1] Standard Errors assume that the covariance matrix of the errors is correctly
specified.
```

0.11 Model 6

[30]:
```
# Model 6 (train)
# Removed Age

my_model = str('Purchase ~ City_Category + Product_Category_1')

# fit the model to the training set
train_model_fit = smf.ols(my_model, data = train).fit()

# summary of model fit to the training set
print(train_model_fit.summary())

#R^2 is still ~63%
```

```
                            OLS Regression Results
==============================================================================
Dep. Variable:               Purchase   R-squared:                       0.629
Model:                            OLS   Adj. R-squared:                  0.629
Method:                 Least Squares   F-statistic:                 3.840e+04
Date:                Fri, 14 Jun 2019   Prob (F-statistic):               0.00
Time:                        10:46:07   Log-Likelihood:            -4.0579e+06
No. Observations:              430061   AIC:                         8.116e+06
Df Residuals:                  430041   BIC:                         8.116e+06
Df Model:                          19
Covariance Type:            nonrobust
================================================================================
============
                               coef    std err          t      P>|t|      [0.025
0.975]
--------------------------------------------------------------------------------
------------
Intercept                 1.333e+04     12.016   1109.138      0.000    1.33e+04
1.34e+04
City_Category[T.1]          160.7359     11.410     14.088      0.000     138.373
183.098
City_Category[T.2]          611.9507     12.194     50.185      0.000     588.051
635.850
Product_Category_1[T.2]  -2336.7666     23.897    -97.784      0.000   -2383.604
-2289.929
Product_Category_1[T.3]  -3507.1680     25.664   -136.656      0.000   -3557.469
-3456.867
Product_Category_1[T.4]  -1.125e+04     32.703   -344.022      0.000   -1.13e+04
-1.12e+04
Product_Category_1[T.5]  -7336.6051     12.675   -578.805      0.000   -7361.449
-7311.762
Product_Category_1[T.6]   2234.5671     25.536     87.507      0.000    2184.517
2284.617
```

```
Product_Category_1[T.7]   2750.5623     56.771     48.450      0.000    2639.292
2861.833
Product_Category_1[T.8]  -6083.2313     13.626   -446.434      0.000   -6109.938
-6056.524
Product_Category_1[T.9]   2259.2467    169.682     13.315      0.000    1926.675
2591.819
Product_Category_1[T.10]  6032.2749     48.545    124.262      0.000    5937.129
6127.421
Product_Category_1[T.11] -8894.1683     23.729   -374.829      0.000   -8940.676
-8847.661
Product_Category_1[T.12] -1.223e+04     55.051   -222.232      0.000   -1.23e+04
-1.21e+04
Product_Category_1[T.13] -1.285e+04     46.885   -274.154      0.000   -1.29e+04
-1.28e+04
Product_Category_1[T.14]  -316.5860     88.970     -3.558      0.000    -490.964
-142.208
Product_Category_1[T.15]  1195.7966     43.867     27.260      0.000    1109.819
1281.774
Product_Category_1[T.16]  1174.4705     35.637     32.957      0.000    1104.623
1244.318
Product_Category_1[T.17] -3509.1647    142.077    -24.699      0.000   -3787.631
-3230.698
Product_Category_1[T.18] -1.062e+04     61.665   -172.238      0.000   -1.07e+04
-1.05e+04
==============================================================================
Omnibus:                    34739.977   Durbin-Watson:                   2.006
Prob(Omnibus):                  0.000   Jarque-Bera (JB):            53427.243
Skew:                          -0.635   Prob(JB):                         0.00
Kurtosis:                       4.169   Cond. No.                         44.0
==============================================================================

Warnings:
[1] Standard Errors assume that the covariance matrix of the errors is correctly
specified.
```

[31]:
```
# Model 6 (test)
# Removed Age

my_model = str('Purchase ~ City_Category + Product_Category_1')

# fit the model to the training set
test_model_fit = smf.ols(my_model, data = test).fit()

# summary of model fit to the training set
print(test_model_fit.summary())

#R^2 is still ~63%
```

OLS Regression Results

Dep. Variable:	Purchase	R-squared:	0.633
Model:	OLS	Adj. R-squared:	0.633
Method:	Least Squares	F-statistic:	9759.
Date:	Fri, 14 Jun 2019	Prob (F-statistic):	0.00
Time:	10:46:10	Log-Likelihood:	-1.0143e+06
No. Observations:	107516	AIC:	2.029e+06
Df Residuals:	107496	BIC:	2.029e+06
Df Model:	19		
Covariance Type:	nonrobust		

	coef	std err	t	P>|t|	[0.025	0.975]
Intercept	1.337e+04	23.946	558.344	0.000	1.33e+04	1.34e+04
City_Category[T.1]	160.9483	22.789	7.063	0.000	116.283	205.614
City_Category[T.2]	605.1018	24.345	24.855	0.000	557.385	652.818
Product_Category_1[T.2]	-2339.5837	47.877	-48.866	0.000	-2433.422	-2245.745
Product_Category_1[T.3]	-3520.5364	51.835	-67.918	0.000	-3622.132	-3418.941
Product_Category_1[T.4]	-1.13e+04	66.329	-170.303	0.000	-1.14e+04	-1.12e+04
Product_Category_1[T.5]	-7389.0672	25.238	-292.774	0.000	-7438.533	-7339.601
Product_Category_1[T.6]	2275.9067	51.098	44.540	0.000	2175.755	2376.059
Product_Category_1[T.7]	3104.7693	112.681	27.554	0.000	2883.915	3325.623
Product_Category_1[T.8]	-6112.7630	27.183	-224.872	0.000	-6166.042	-6059.484
Product_Category_1[T.9]	759.3907	330.758	2.296	0.022	111.110	1407.671
Product_Category_1[T.10]	6240.0813	97.847	63.774	0.000	6048.303	6431.859
Product_Category_1[T.11]	-8941.7901	47.266	-189.181	0.000	-9034.431	-8849.150
Product_Category_1[T.12]	-1.226e+04	111.427	-110.059	0.000	-1.25e+04	-1.2e+04
Product_Category_1[T.13]	-1.29e+04	93.215	-138.349	0.000	-1.31e+04	-1.27e+04
Product_Category_1[T.14]	-803.1675	168.377	-4.770	0.000	-1133.184	

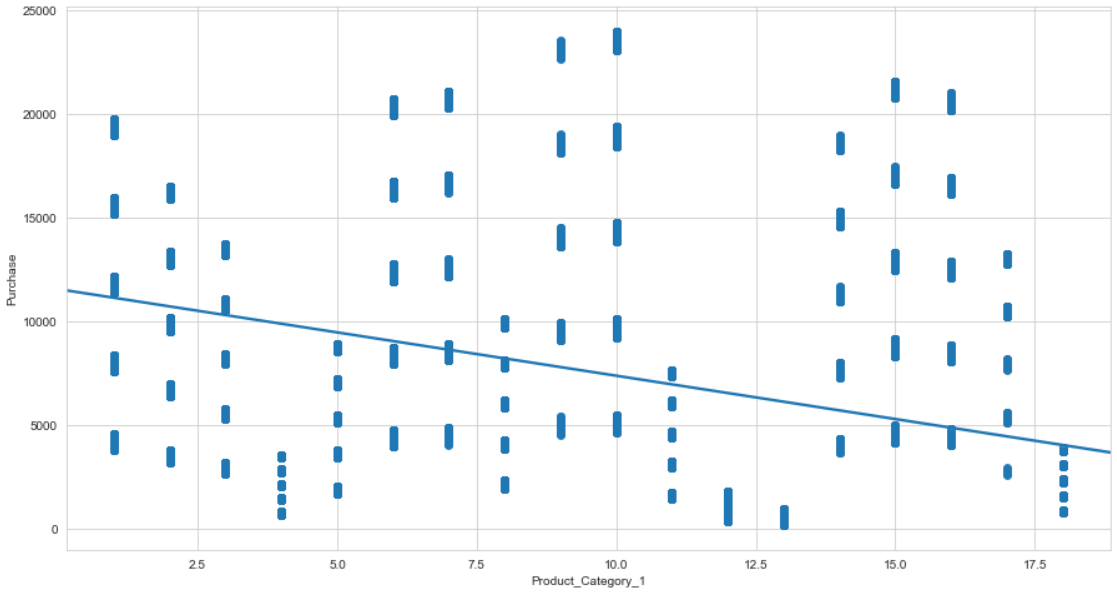
```
-473.151
Product_Category_1[T.15]  1141.9338     88.783     12.862      0.000     967.921
1315.946
Product_Category_1[T.16]  1197.7358     70.824     16.911      0.000    1058.921
1336.550
Product_Category_1[T.17] -3215.8342    289.171    -11.121      0.000   -3782.605
-2649.064
Product_Category_1[T.18] -1.063e+04    124.389    -85.475      0.000   -1.09e+04
-1.04e+04
==============================================================================
Omnibus:                     8522.658   Durbin-Watson:                   2.009
Prob(Omnibus):                  0.000   Jarque-Bera (JB):            13072.393
Skew:                          -0.627   Prob(JB):                         0.00
Kurtosis:                       4.160   Cond. No.                         42.9
==============================================================================

Warnings:
[1] Standard Errors assume that the covariance matrix of the errors is correctly
specified.
```

[32]:
```
# OLS Regression Notes

# In our final Model #6, we ended up keeping
    # Product Category 1 (but it only shows 2-18)
    # City Category (but it only shows 1-2)

# Best R^2 we got was 63%
```

0.12 Visualizations

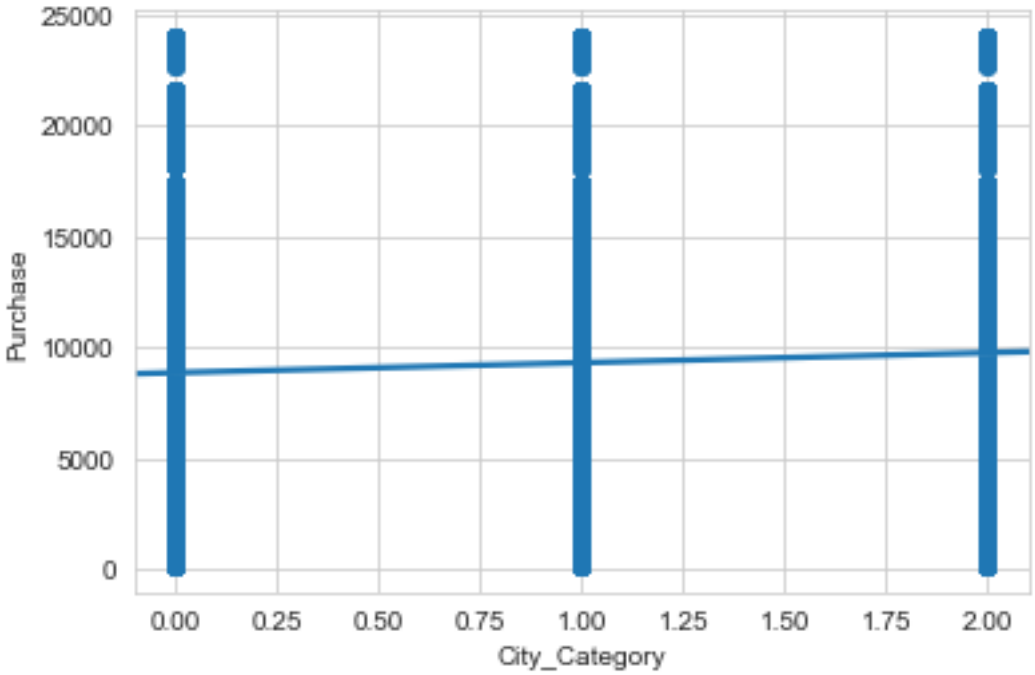
[35]:
```
# Generate Plot of Product Category 1 vs Purchase Amount
# (takes a while, ~ 1 min)

rcParams['figure.figsize'] = 15,8 # set up figure size
sns.regplot(x='Product_Category_1', y='Purchase', data=BF, scatter=True)
```

[35]:
```
<matplotlib.axes._subplots.AxesSubplot at 0x2446266cf98>
```



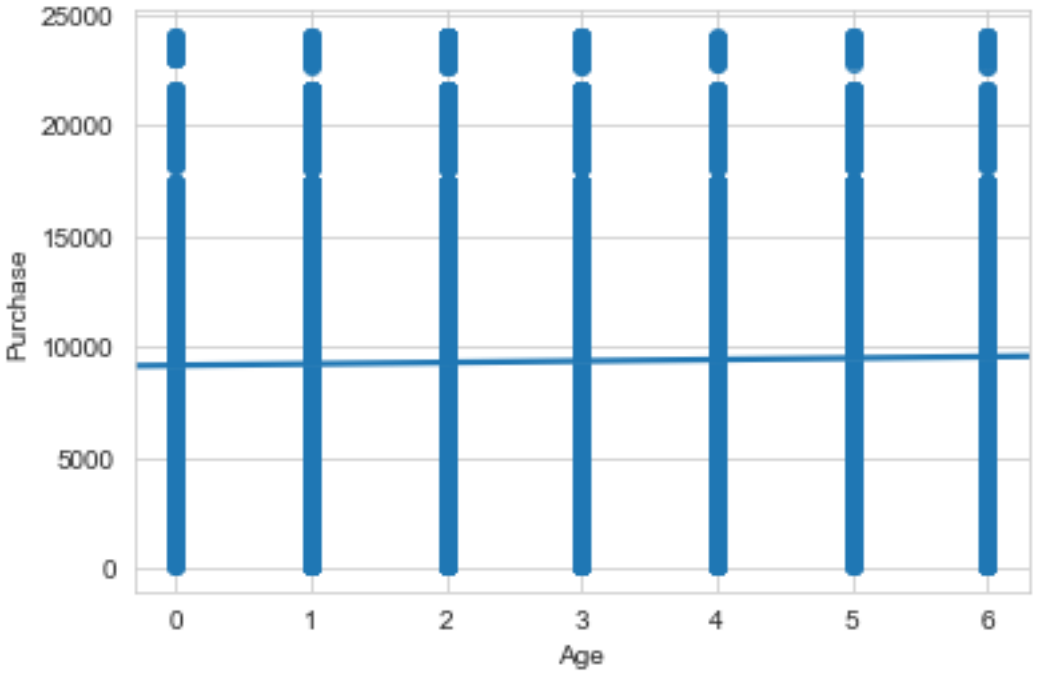
[34]:
```
# Generate Plot of City Category vs Purchase Amount
# (takes a while, ~ 1 min)

rcParams['figure.figsize'] = 6,4 # set up figure size
sns.regplot(x='City_Category', y='Purchase', data=BF, scatter=True)
```

[34]: <matplotlib.axes._subplots.AxesSubplot at 0x24463d8a1d0>



[36]:
```
# Generate Plot of Gender vs Purchase Amount
# (takes a while, ~ 1 min)

rcParams['figure.figsize'] = 6,4 # set up figure size
sns.regplot(x= 'Age', y='Purchase', data=BF, scatter=True)
```

[36]:
```
<matplotlib.axes._subplots.AxesSubplot at 0x244625d7c50>
```



[ ]:
```
# End of Black Friday Notebook 2
```