P. Enthusiasts

Predicting Future Potential Starbucks Locations

University of Colorado, Denver

Anamika Singh, Rubina Shaik, Shivali Arora and Shruthi Prasad

April,2020

Finding Future Potential Starbucks locations

Abstract:

Planning of outlet locations is undoubtedly the most important decision for a business. A well-designed location strategy is integral to the overall corporate strategy of any business. Use of data analytics in location intelligence gives an important insight to foster potential business opportunities. For this project, we chose one such scenario to evaluate and recommend the best possible solution. We aimed to determine potential business opportunities for Starbucks by finding locations where Starbucks could open a store. To accomplish this task, we used four data sets: demographics dataset (Demographics - Kaggle) for 2017 and Starbucks location data from Kaggle (www.kaggle.com); and county data from United States census bureau .These data sets were used to arrive at a model that describe Starbucks count in association with Total population, Income per capita, Number of professionals and Asians in that county. Potential locations were suggested using k-nearest algorithm. Further, we used multiple linear regression to determine the factors affecting Starbucks location; the factors that significantly affected our model include total population, income per capita, Asian ethnicity and professional status. Data visualization was performed using various Python packages and tools. In conclusion, our analysis provides relevant factors that can be considered for opening a future Starbucks store. Based on the available data and our analysis, we suggested 62 potential locations for Starbucks stores across the US.

Keywords: Starbucks, Location Intelligence, Multiple Linear Regression, K-Nearest Neighbor, Prediction Model

Datasets mining:

We brainstormed on various data sets using sites such as Kaggle, Google Data sets, WHO data sets and UN data sets. For this project, we found relevant datasets on Kaggle. For latest data on counties and their respective ZIP codes, we used the latest data from the United States census bureau official website.

Datasets used:

We merged and used four data sets for our analysis:

a) ***Census Demographics data*** from Kaggle for 2017 (includes information on population, employment, unemployment rate, type of jobs, Income per cap, Professionals, and Voting Age Citizens) (1)
b) ***Starbucks locations Worldwide Data*** from Kaggle for 2017 (includes store number, store name, ownership type, street address, city, state, country, postcode, phone number, time zone, longitude and latitude) (2)
c) ***ZIP-COUNTY-FIPS data*** from Kaggle for 2017 (includes ZIPcode, County names, US State, ST County FP) (3)
d) ***2019_Gaz_counties_national*** from United States census bureau (includes counties and respective zip codes) (4)

Steps to Clean and Format Dataset (s):

1. Handling missing data: We used Python "isna.sum()" to look for any missing data. The missing data rows were removed from analysis.
2. Merging the data sets: Since, we used four data sets proper merging was crucial for our analysis. We used 'outer' merge in Pandas to merge our data sets.
3. Formatting data: Slicing the post codes to 5 digits was performed using "str.slice()" as zip codes had different formats in our data sets.
4. Checking the assumptions of Regression: Before proceeding with multiple linear regression, regression assumptions such as Linearity, Independence of Errors, Normality and Equal variance for the considered data columns were checked. We observed that plotting Starbucks count versus Total population violated these assumptions. To overcome this issue, we transformed the data by taking cube root on both Starbucks count and total population.
5. Summary statistics like count(), summary(), mean() etc were used to conduct an overall inspection of various columns in the data set. Converting absolute numbers to percentages for some columns such as Employed, Men, Women and VotingAge Citizen was performed for uniformity in the data set.

Finding Future Potential Starbucks locations

Objective:

Predicting locations in the United States where Starbucks could open future stores by analyzing present store locations data and its mapping to Census demographics.

Methods:

We used Numpy, Pandas, SkiLearn, StatsModel and Seaborn for our analysis. The analysis comprised three major data analytic techniques; K nearest neighbors (KNN) algorithm, multiple linear regression and testing and training our data set and prediction based on the model. Various data visualization tools were used to present data.

Analysis and Results:

The Starbucks datasets consisted of worldwide Starbucks location. The dataset included data for four of its brands: Coffee House Holdings, Tevana, Evolution Fresh and Starbucks. For our analysis, we only considered the locations across the United States. We inspected the data for missing values and also visualized it for any other discrepancy. We included all the brands of Starbucks Corp. for our analysis. The postcode entry in Starbucks data was non-uniform with either 9- or 5-digit entry. We sliced the postcode to retrieve the first five digits and merged this dataset with Census Demographic data, County data and ZIP code county data. For merging, we used the identifier field as ZIP code and used the “outer” merge option. Thus, we could retrieve the ZIP Codes that do not have Starbucks. We used this information for KNN algorithm. Applying “Groupby” to this resulting data, we calculated total Starbucks count in a county and then merged the result with counties data to get the respective latitude and longitude coordinates. Further, this data was merged with another county data file consisting of latitude and longitude of each county. This information was used to find the five nearest counties for the KNN algorithm. The algorithm suggested 377 counties which currently do not have any Starbucks. This data was then filtered based on the outcome of multiple linear regression model.

The data was visually inspected using pairplots (**Figure 1**).

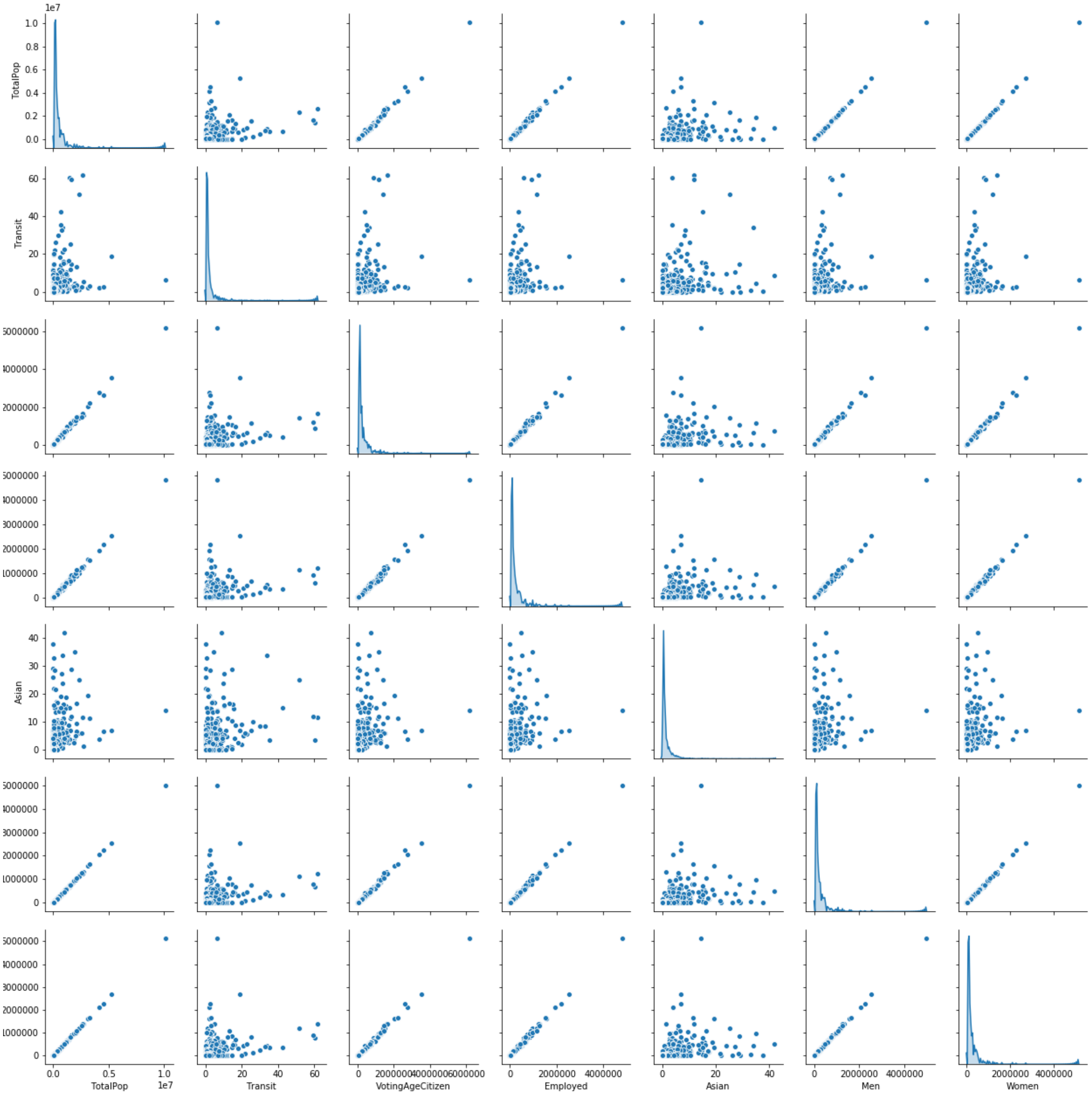


Figure 1: Pair plots of Various Variables of the Dataset

As a general assumption, the number of Starbucks (or any other retail outlet) in a county depends on the demographics of the population in that area. To find the relevant factors that could affect potential Starbucks location, we used the county demographics data ("acs_2017_county" data from Kaggle) that consisted of 3221 rows and 37 columns. Each row describes demographics of a county.

Utilizing multiple linear regression, we built a model to establish the relationship between Starbucks count (Y; dependent variable) and Total population, Asian (ethnicity), Professional and Income per capita as independent variables (X). We chose the variables that had correlation coefficient > ~0.3 (**Figure 2**). These include Total population, men, women, Asian, Voting Age citizen, Income per cap, Professional, const, transit and employed. We further improved the model by considering total population as a factor influencing the number of Starbucks and then adding more parameters while checking for most optimum adjusted R squared values. While optimizing, we also checked whether the regression assumptions were met.

MLR Equation:

CubicRootStarbucksCount = -1.5171+ 3.93753743e-02(CubicRootTotalPop)+ 2.27014241e-02(Asian)+

1.06304548e-02(Professional) + 2.22515756e-05(IncomePerCap)

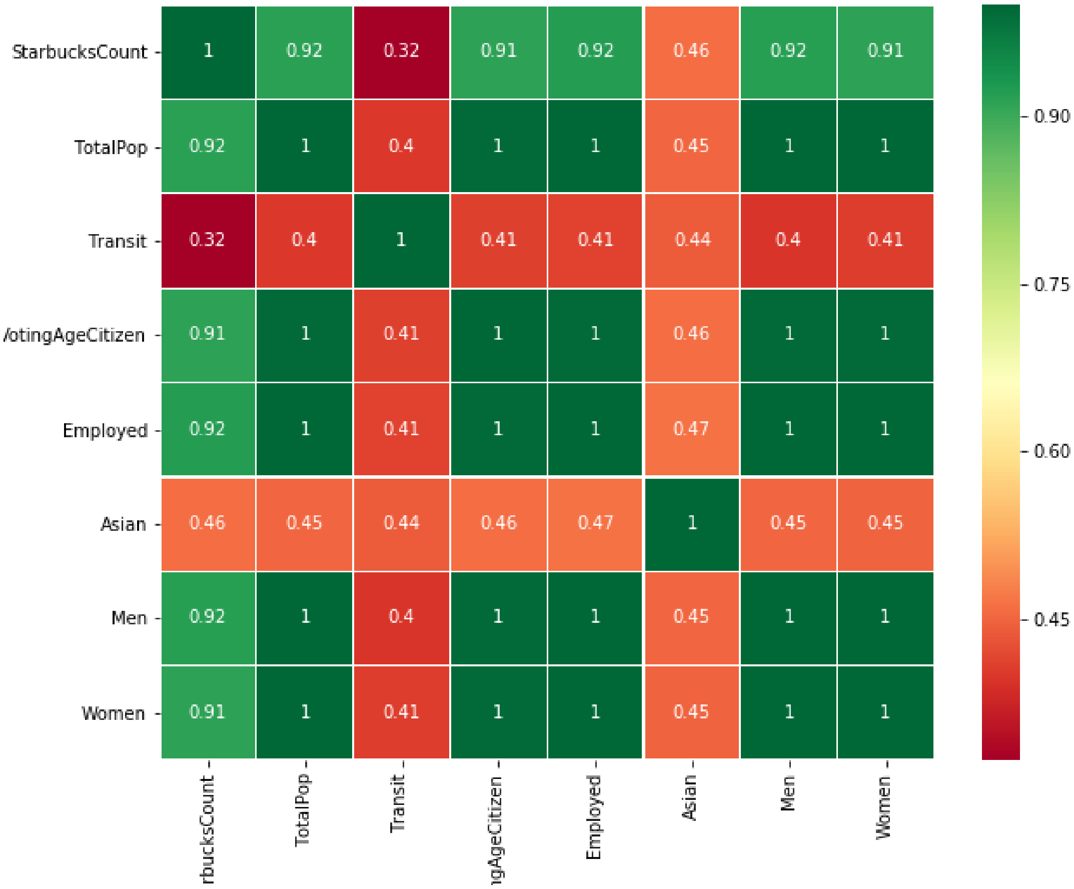
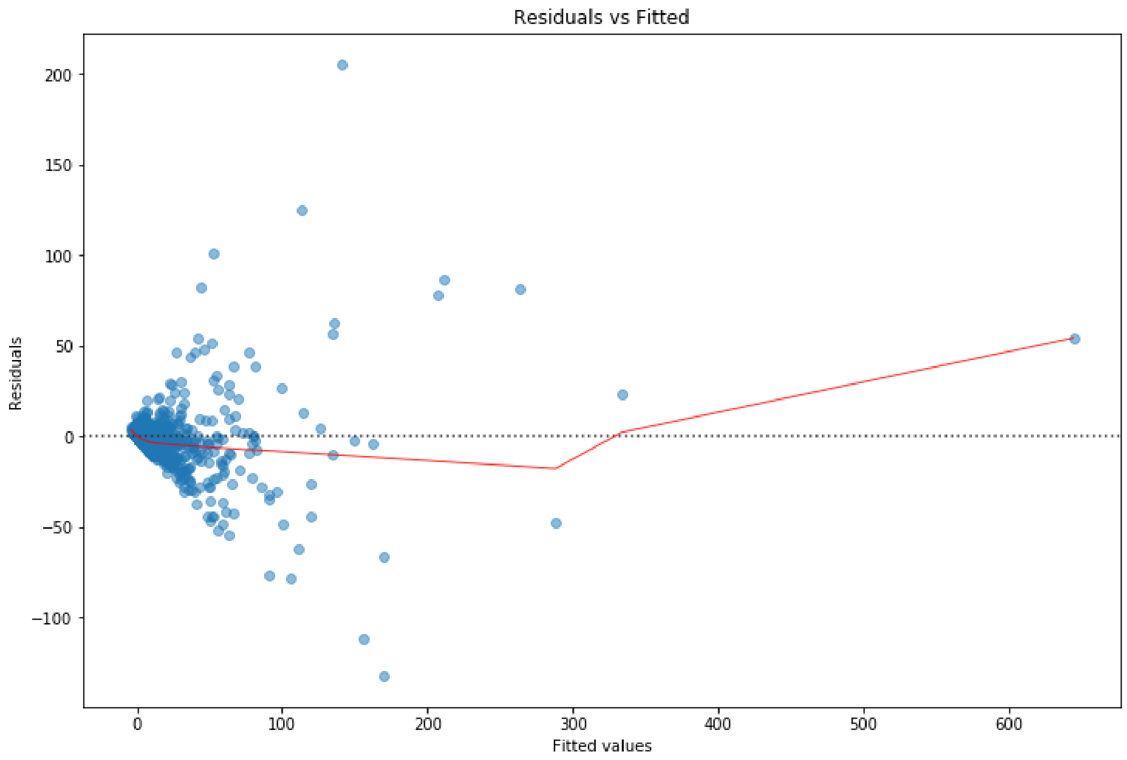


Figure 2: Correlation matrix for Variables in the Dataset (correlation coefficient>0.3 presented here)

Starbucks count and Total population were transformed to their cube roots to comply with the regression assumptions (5). The assumptions were checked using residual and QQ plots (**Figures 3a and 3b, Figures 4a and 4b**).



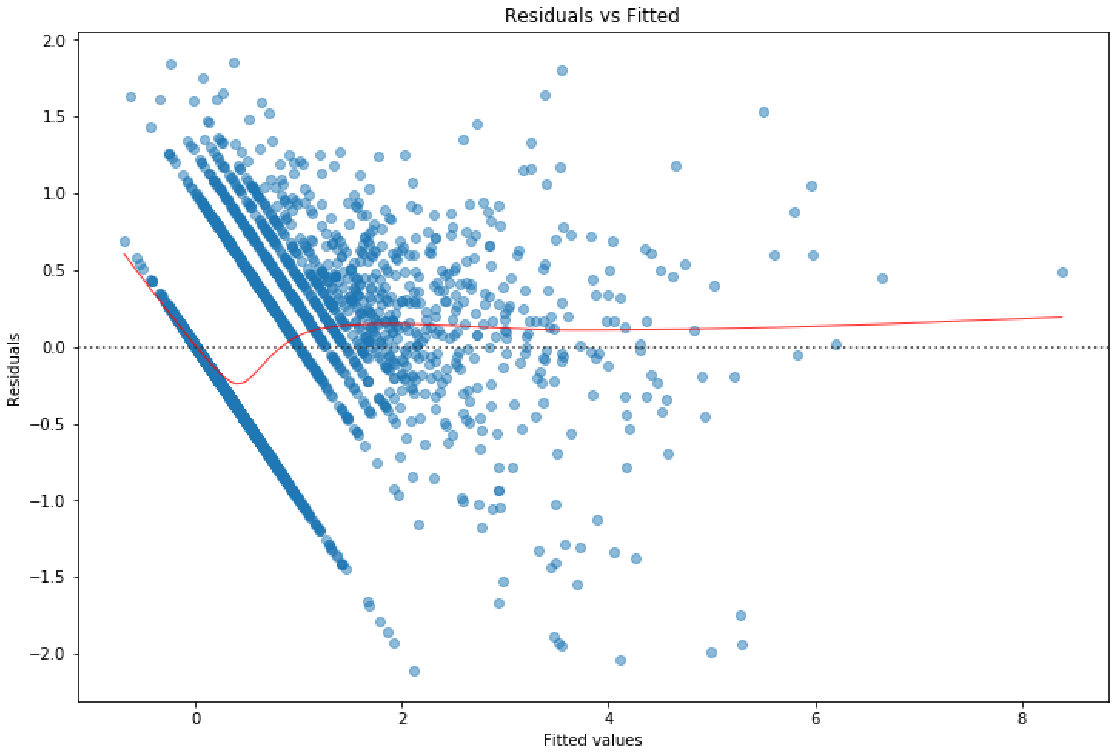
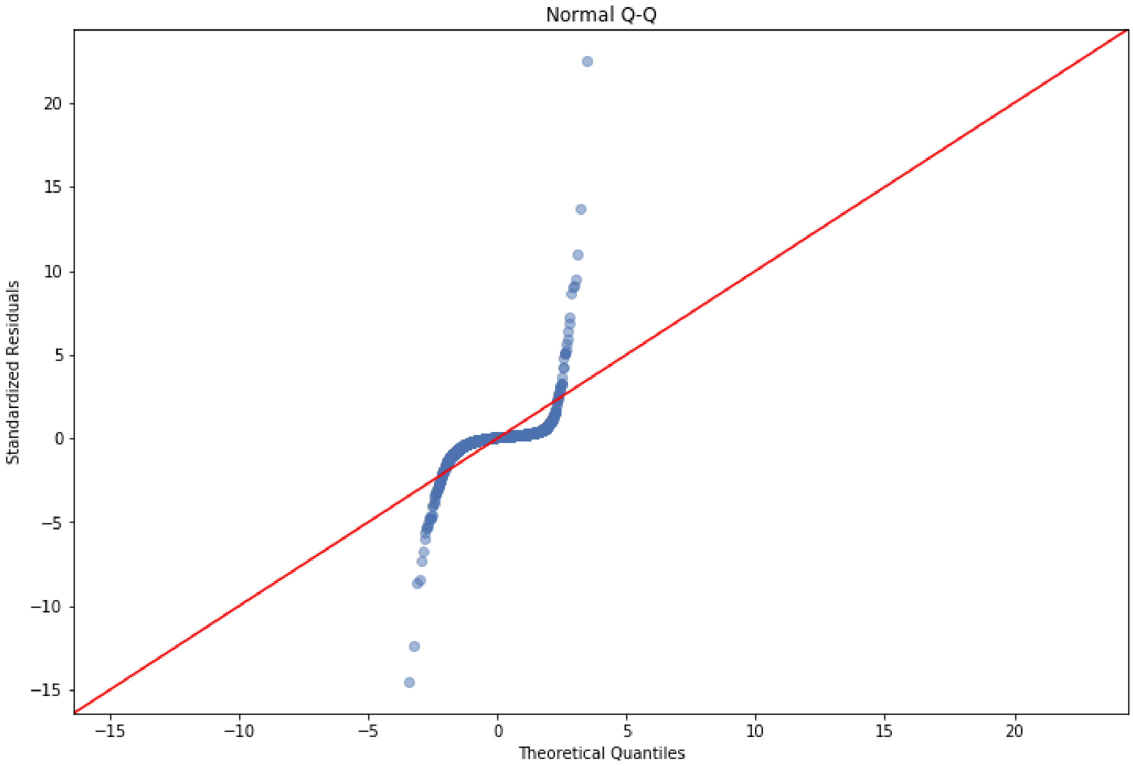


Figure 3: Residual Plots of Total Population Before (a) and After (b) Cubic Transformation



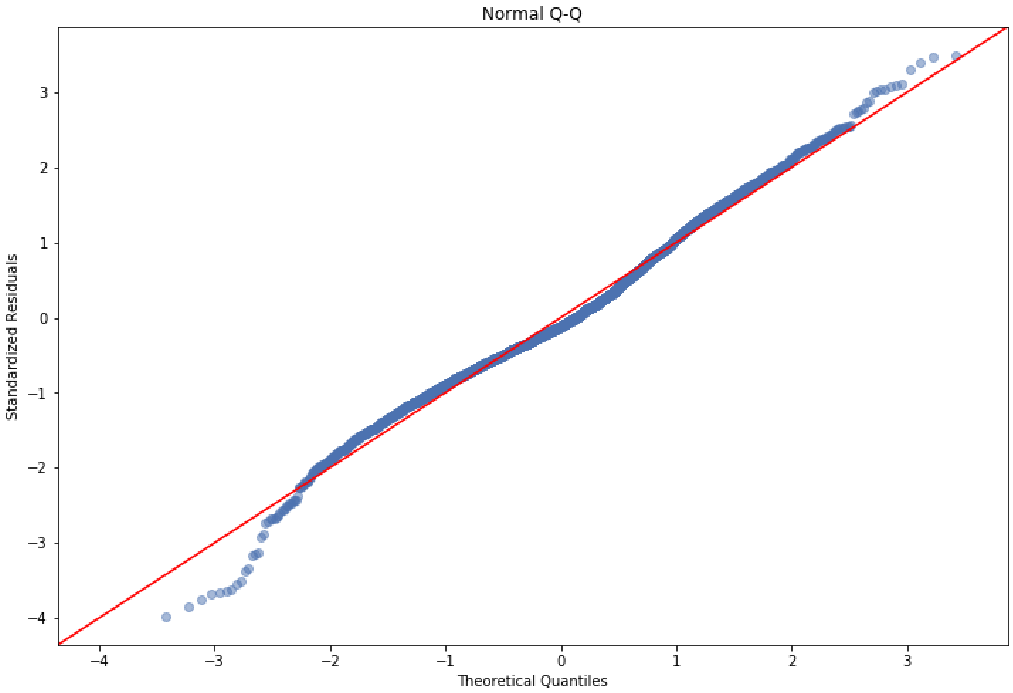
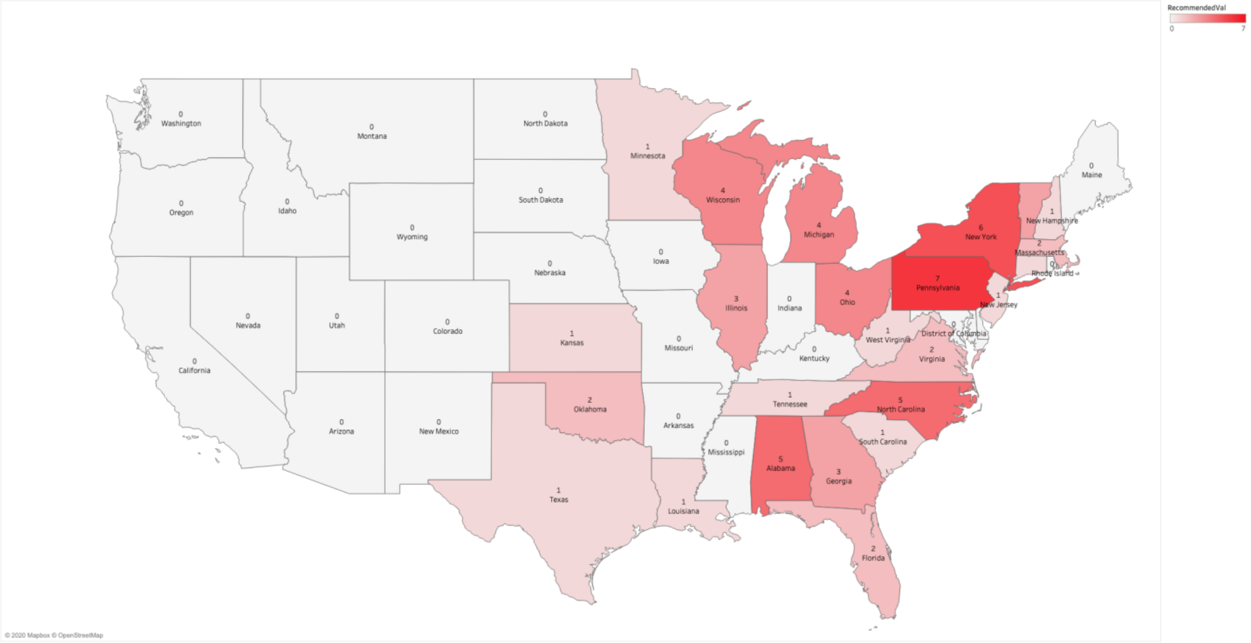


Figure 4: QQ Plot Before (a) and After (b) Cubic Transformation

The KKN algorithm resulted in 377 suggested locations where Starbucks could potentially open a new store. Applying MLR to these suggestions resulted in predictions for 62 counties. Computing Group by based on States, we got ~20 suggestions. This is plotted as a heat map in Tableau (**Figure 5**).

Finding Future Potential Starbucks locations

Sheet 1



Map based on Longitude (generated) and Latitude (generated). Color shows sum of RecommendedVal. The marks are labeled by sum of RecommendedVal and State. Details are shown for State.

Figure 5: Heat map showing predicted Starbucks store across US

Key Insights:

Based on KNN, we came up with top 377 potential locations where Starbucks could open future stores (**Appendix IA**). Multiple linear regression showed that total population, income per capita, professional status and Asian (ethnicity) as important variables in predicting a Starbucks location (**Appendix IB**). Applying this model to the KNN algorithm resulted in final recommendations of 62 new Starbucks store locations. To avoid overfitting, we did not include overlapping variables in our analysis.

With KNN algorithm, we were trying to answer the question whether a county should have a Starbucks store or not. We assumed that greater than or equal to three surrounding counties with Starbucks would lead to higher brand recognition and hence higher sales. The initial results from the KNN algorithm provided a list of counties which sometimes had lower population, income per capita, Asians and professionals. Our model was well fitted with an adjusted r2 of about 72.5% and mse of 25.9.

We cross verified the KNN algorithm results with multiple linear regression model and chose those counties where prediction was greater than or equal to 1. We compared the predicted values against the current Starbucks count in a county. Positive difference indicated an opportunity to increase the number of Starbucks count in that county and a negative difference indicated an opportunity to reduce the footprint of Starbucks. **Figure 5** indicates that there is a higher density of Starbucks in West coast and an opportunity to have more Starbucks in East coast.

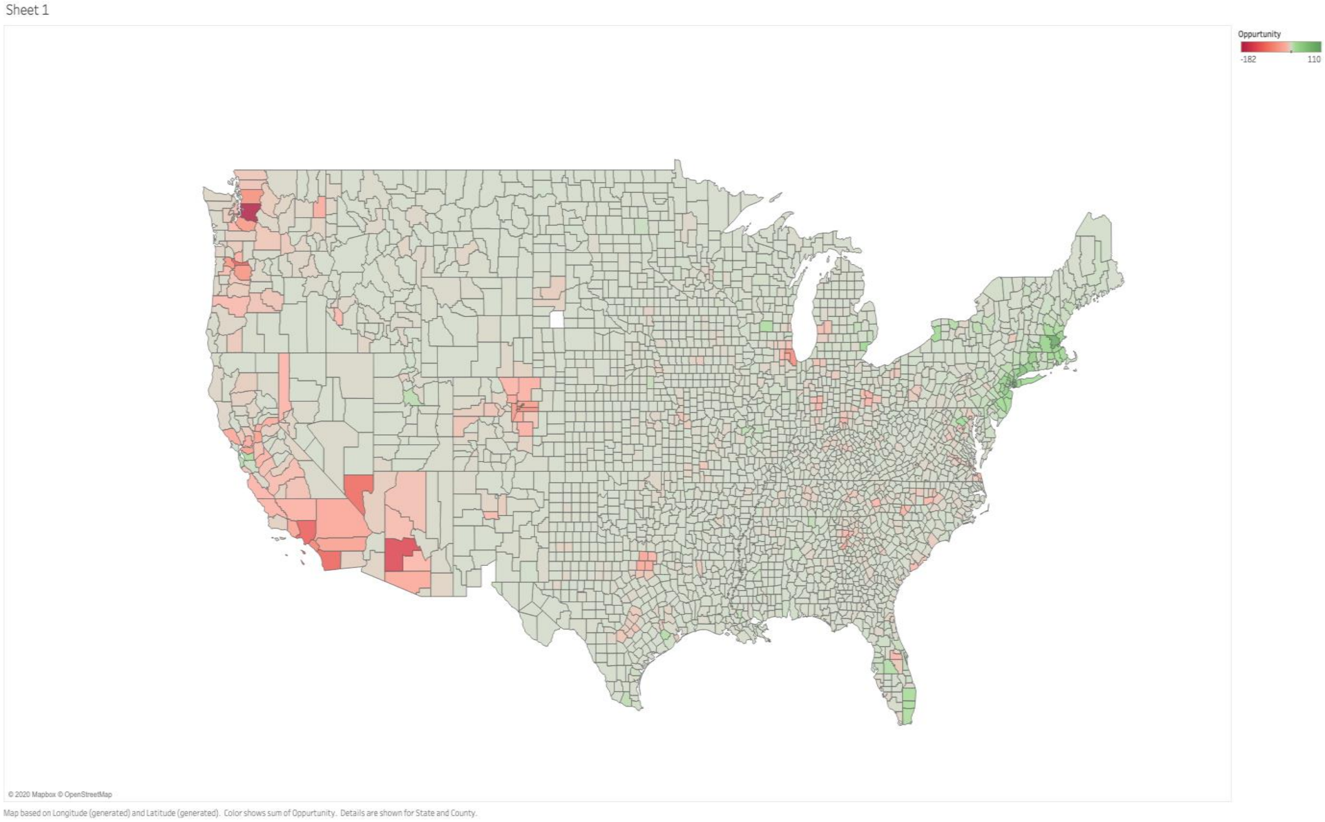
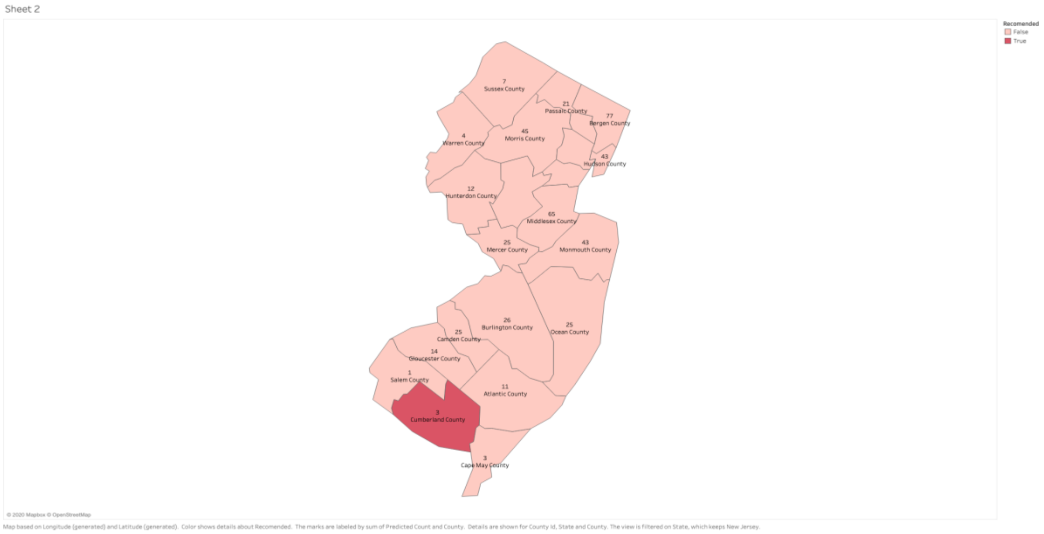

Sheet 1
Oppurtunity
-182
110
© 2020 Mapbox © OpenStreetMap
Map based on Longitude (generated) and Latitude (generated). Color shows sum of Oppurtunity. Details are shown for State and County.

Figure 5: Red Regions Highlight Opportunities to Reduce Starbucks Footprint and Green Regions Highlight Opportunities to Open more Starbucks

Overall, we observed higher predictions for opening Starbucks stores in the East coast; specifically Pennsylvania, New York, Alabama and North-Carolina with highest predictions of 7, 6, 5 and 5 store openings, respectively (**Figure 6**).

We used Starbucks location data of 2017 for our analysis. We noticed that some of the suggestions as per our model now have a Starbucks store. **Figure 7** presents one such suggestion; Cumberland county in New Jersey where our model suggested 3 Starbucks stores.

Finding Future Potential Starbucks locations



Map based on Longitude (generated) and Latitude (generated). Color shows details about Recomended. The marks are labeled by sum of Predicted Count and County. Details are shown for County Id, State and County. The view is filtered on State, which keeps New Jersey.

Figure 6: Suggested Starbucks in Cumberland County (New Jersey) as per Multiple Linear Regression Model (Starbucks 2017 Dataset)

We noticed that Starbucks has opened 2 stores in Cumberland county (**Figure 7; Source: Google maps**).

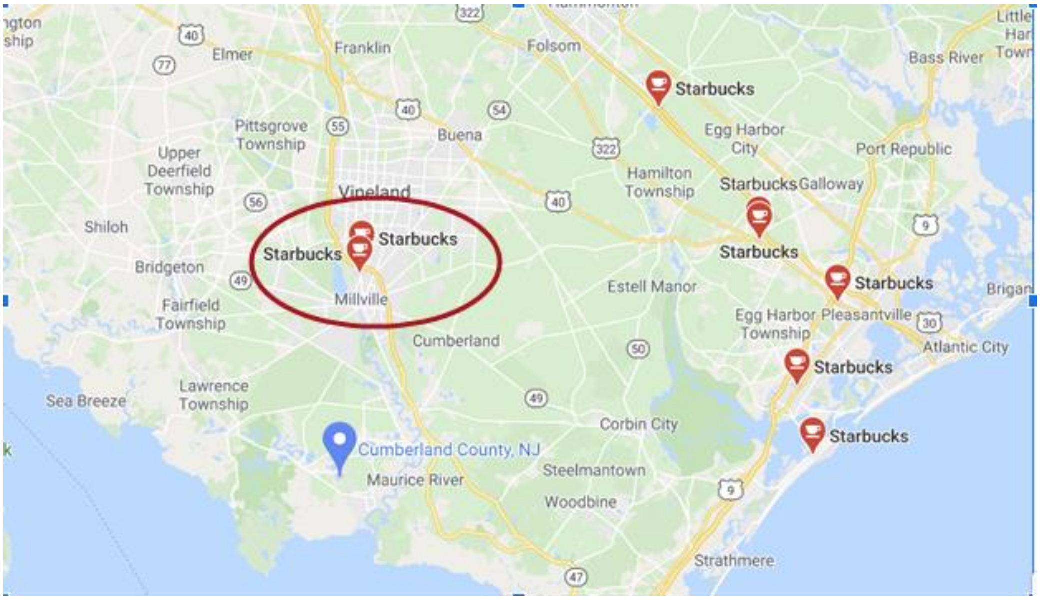


Figure 7: Two Newly Opened Starbucks Stores in Cumberland County (Prediction as per our Model: 3)

Scope of our model:

We did not consider factors like market saturation, real estate and labor costs, management strategy and growth goals of the company. This may also partially explain the difference in our prediction and actual Starbucks count in some counties.

Other factors:

We used Tableau to depict maps showing recommended Starbucks location.

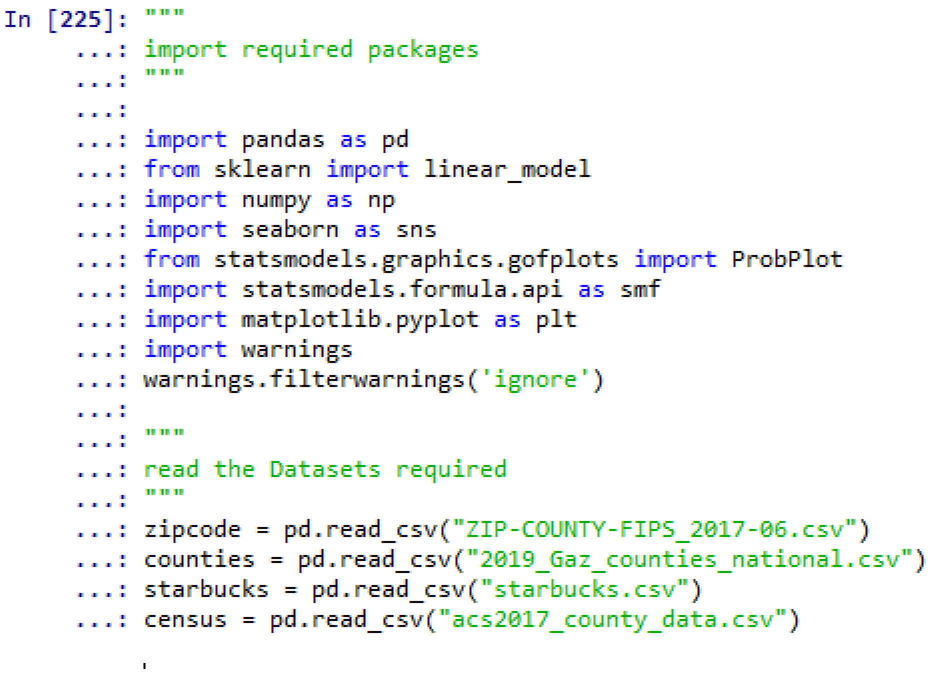
Sources/References:

1. Census dataset available at: https://www.kaggle.com/muonneutrino/us-census-demographic-data)
2. Starbucks Location Worldwide dataset available at: https://www.kaggle.com/starbucks/store-locations
3. ZIP county data available at: https://www.kaggle.com/danofer/zipcodes-county-fips-crosswalk
4. National county data available at; Index of /geo/docs/maps-data/data/gazetteer/2019_Gazetteer
5. Cubic Transformation: https://medium.com/@emredjan/emulating-r-regression-plots-in-python-43741952c034

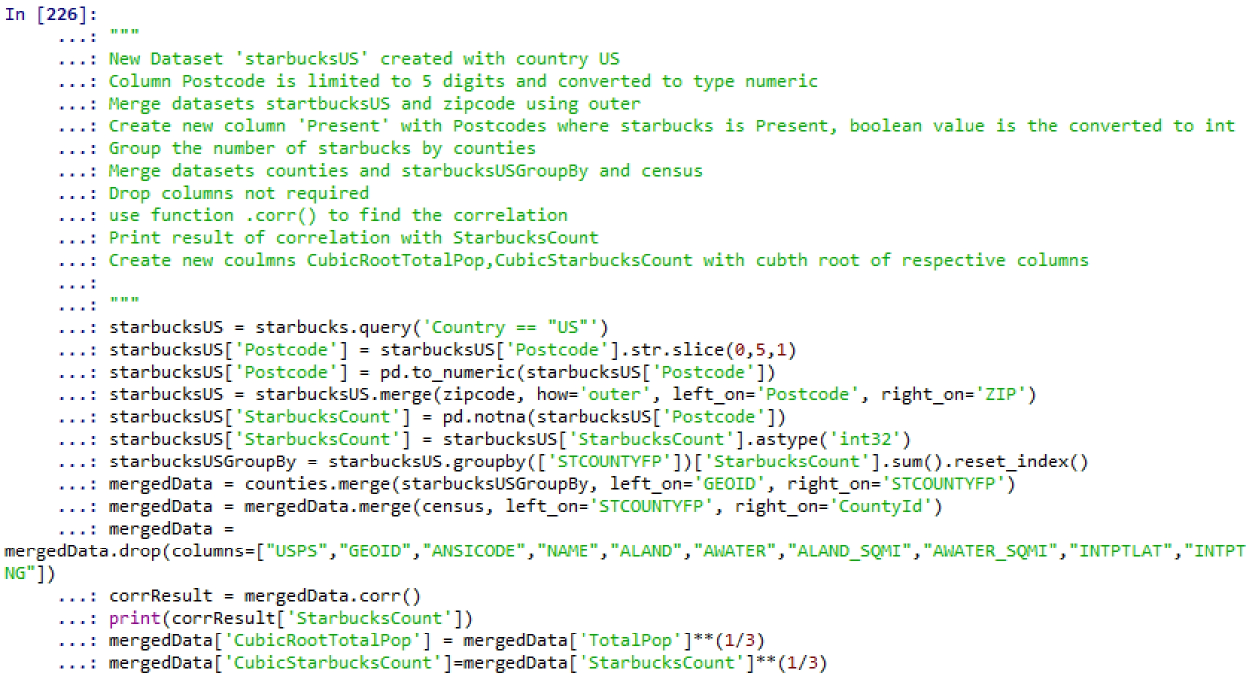
Appendix I:

Python Code and Outcome:

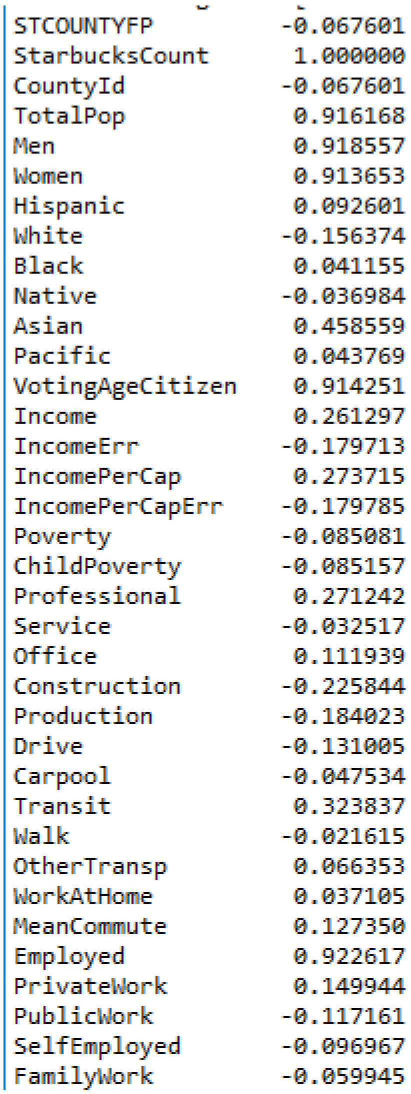
I A: Multiple Linear Regression: Model Building



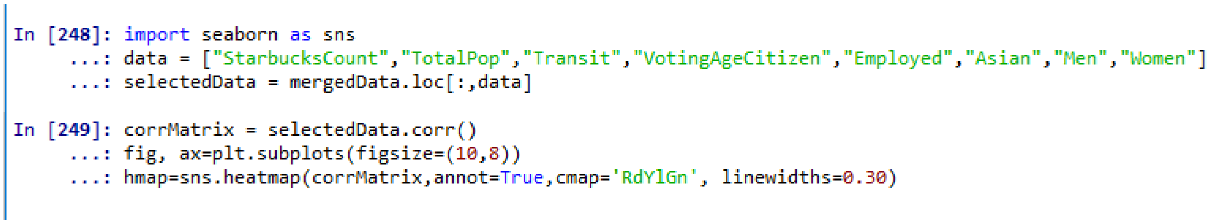
```
In [225]: """
     ...: import required packages
     ...: """
     ...:
     ...: import pandas as pd
     ...: from sklearn import linear_model
     ...: import numpy as np
     ...: import seaborn as sns
     ...: from statsmodels.graphics.gofplots import ProbPlot
     ...: import statsmodels.formula.api as smf
     ...: import matplotlib.pyplot as plt
     ...: import warnings
     ...: warnings.filterwarnings('ignore')
     ...:
     ...: """
     ...: read the Datasets required
     ...: """
     ...: zipcode = pd.read_csv("ZIP-COUNTY-FIPS_2017-06.csv")
     ...: counties = pd.read_csv("2019_Gaz_counties_national.csv")
     ...: starbucks = pd.read_csv("starbucks.csv")
     ...: census = pd.read_csv("acs2017_county_data.csv")
```



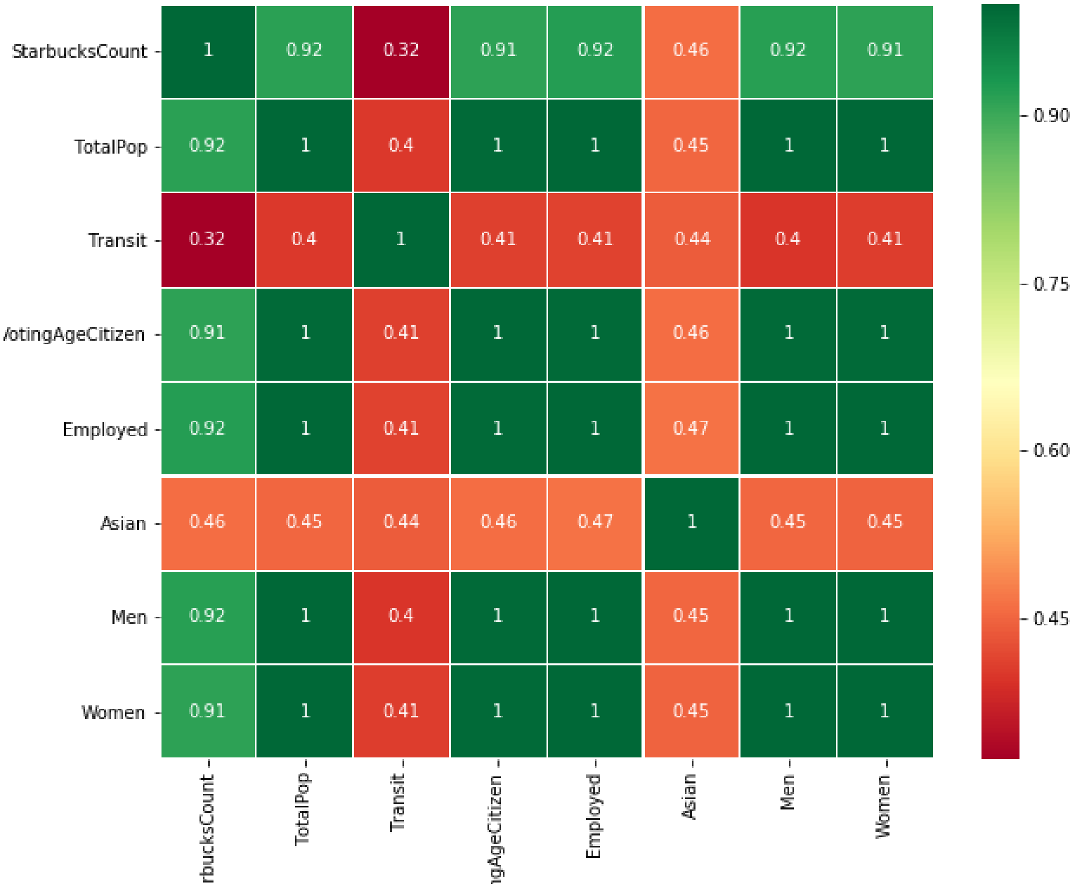
```
In [226]:
     ...: """
     ...: New Dataset 'starbucksUS' created with country US
     ...: Column Postcode is limited to 5 digits and converted to type numeric
     ...: Merge datasets startbucksUS and zipcode using outer
     ...: Create new column 'Present' with Postcodes where starbucks is Present, boolean value is the converted to int
     ...: Group the number of starbucks by counties
     ...: Merge datasets counties and starbucksUSGroupBy and census
     ...: Drop columns not required
     ...: use function .corr() to find the correlation
     ...: Print result of correlation with StarbucksCount
     ...: Create new coulmns CubicRootTotalPop,CubicStarbucksCount with cubth root of respective columns
     ...:
     ...: """
     ...: starbucksUS = starbucks.query('Country == "US"')
     ...: starbucksUS['Postcode'] = starbucksUS['Postcode'].str.slice(0,5,1)
     ...: starbucksUS['Postcode'] = pd.to_numeric(starbucksUS['Postcode'])
     ...: starbucksUS = starbucksUS.merge(zipcode, how='outer', left_on='Postcode', right_on='ZIP')
     ...: starbucksUS['StarbucksCount'] = pd.notna(starbucksUS['Postcode'])
     ...: starbucksUS['StarbucksCount'] = starbucksUS['StarbucksCount'].astype('int32')
     ...: starbucksUSGroupBy = starbucksUS.groupby(['STCOUNTYFP'])['StarbucksCount'].sum().reset_index()
     ...: mergedData = counties.merge(starbucksUSGroupBy, left_on='GEOID', right_on='STCOUNTYFP')
     ...: mergedData = mergedData.merge(census, left_on='STCOUNTYFP', right_on='CountyId')
     ...: mergedData =
mergedData.drop(columns=["USPS","GEOID","ANSICODE","NAME","ALAND","AWATER","ALAND_SQMI","AWATER_SQMI","INTPTLAT","INTPT
NG"])
     ...: corrResult = mergedData.corr()
     ...: print(corrResult['StarbucksCount'])
     ...: mergedData['CubicRootTotalPop'] = mergedData['TotalPop']**(1/3)
     ...: mergedData['CubicStarbucksCount']=mergedData['StarbucksCount']**(1/3)
```



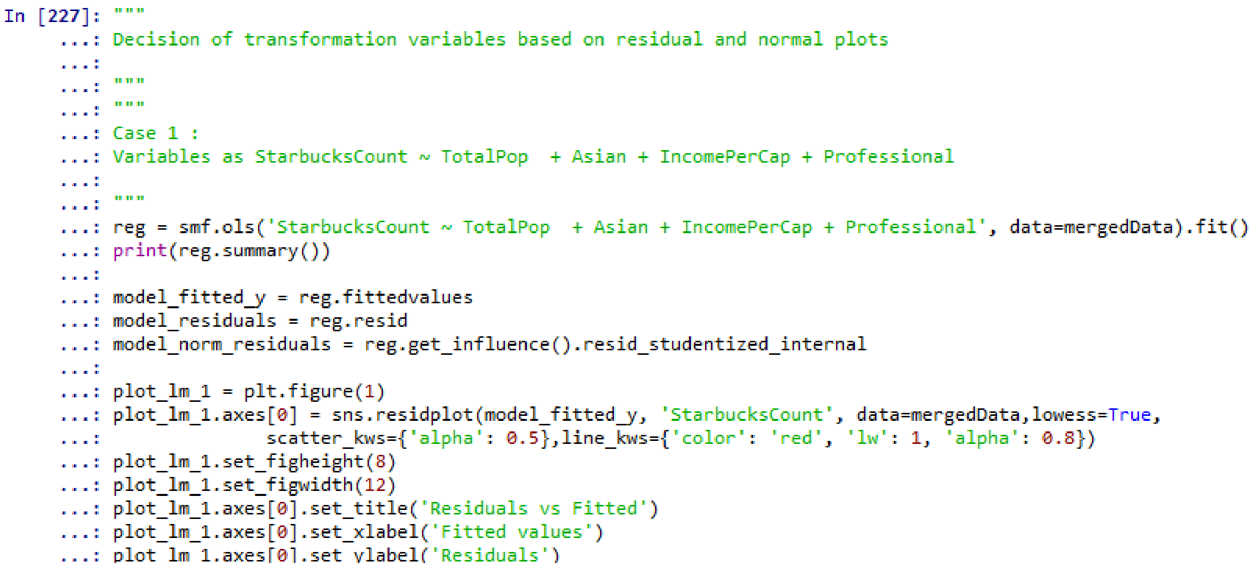
```
STCOUNTYFP          -0.067601
StarbucksCount       1.000000
CountyId            -0.067601
TotalPop             0.916168
Men                  0.918557
Women                0.913653
Hispanic             0.092601
White               -0.156374
Black                0.041155
Native              -0.036984
Asian                0.458559
Pacific              0.043769
VotingAgeCitizen     0.914251
Income               0.261297
IncomeErr           -0.179713
IncomePerCap         0.273715
IncomePerCapErr     -0.179785
Poverty             -0.085081
ChildPoverty        -0.085157
Professional         0.271242
Service             -0.032517
Office               0.111939
Construction        -0.225844
Production          -0.184023
Drive               -0.131005
Carpool             -0.047534
Transit              0.323837
Walk                -0.021615
OtherTransp          0.066353
WorkAtHome           0.037105
MeanCommute          0.127350
Employed             0.922617
PrivateWork          0.149944
PublicWork          -0.117161
SelfEmployed        -0.096967
FamilyWork          -0.059945
```



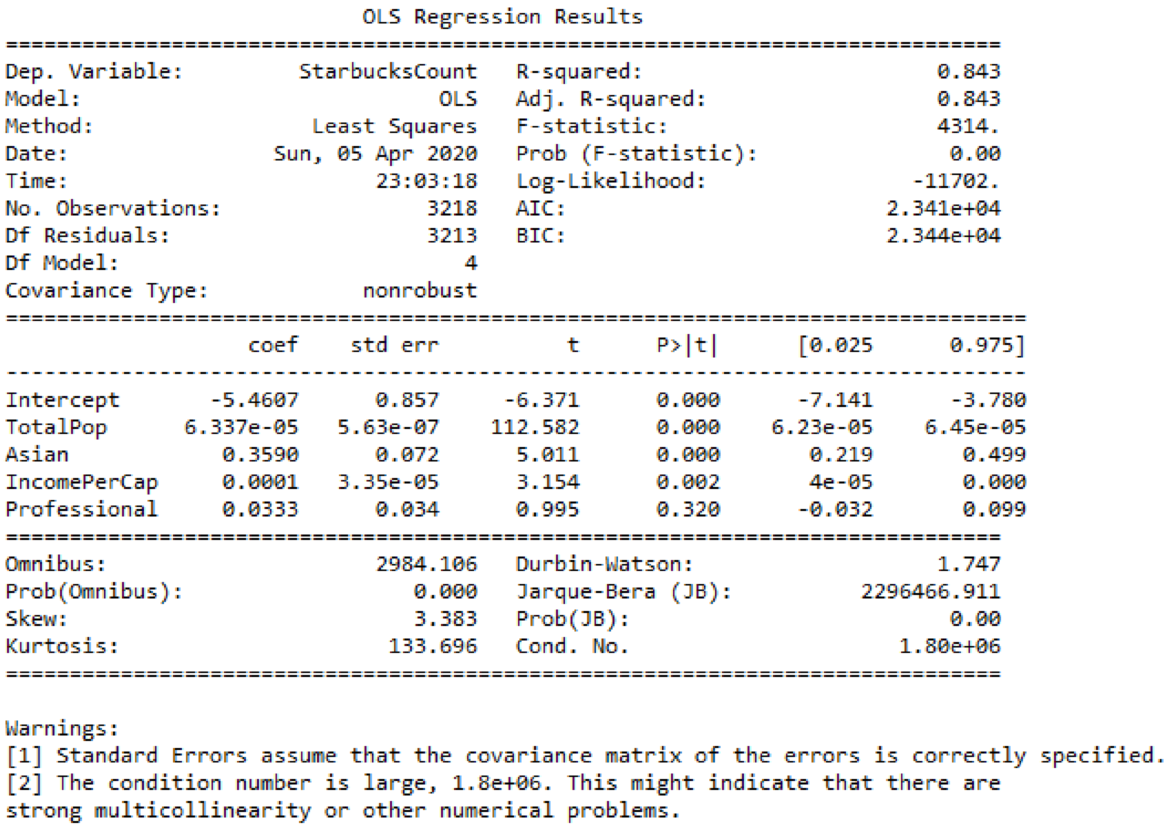
```
In [248]: import seaborn as sns
     ...: data = ["StarbucksCount","TotalPop","Transit","VotingAgeCitizen","Employed","Asian","Men","Women"]
     ...: selectedData = mergedData.loc[:,data]

In [249]: corrMatrix = selectedData.corr()
     ...: fig, ax=plt.subplots(figsize=(10,8))
     ...: hmap=sns.heatmap(corrMatrix,annot=True,cmap='RdYlGn', linewidths=0.30)
```



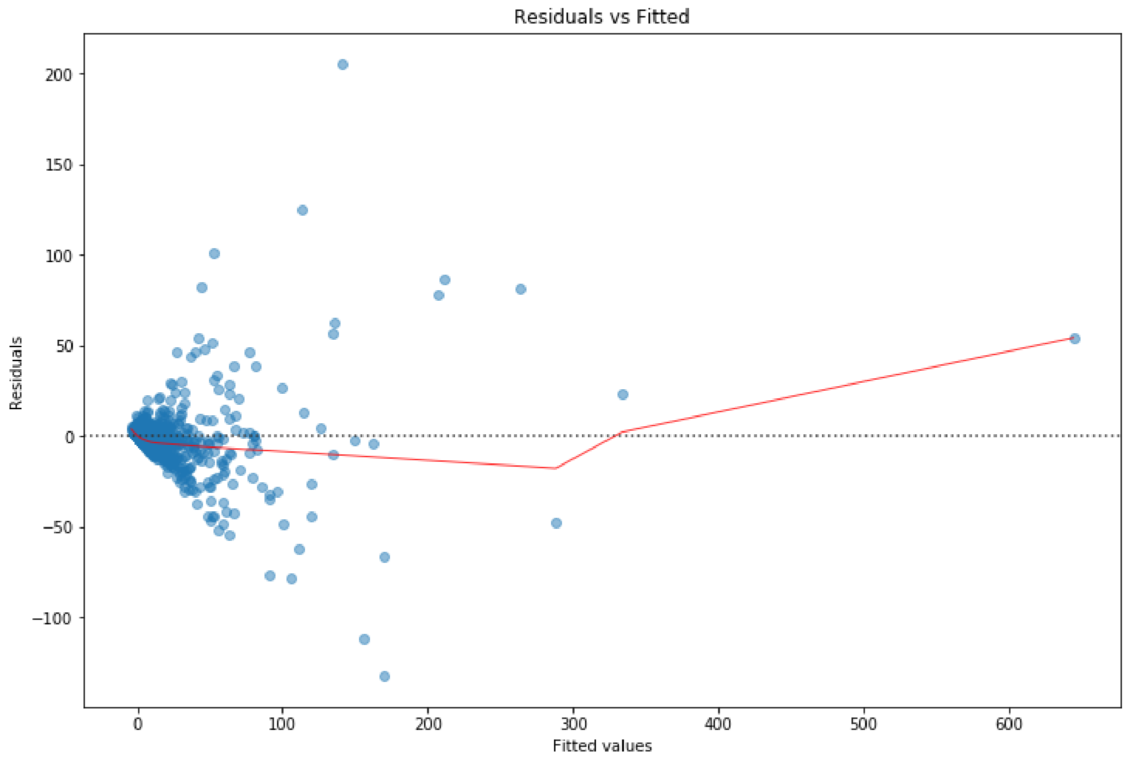
	rbucksCount	TotalPop	Transit	ıgAgeCitizen	Employed	Asian	Men	Women
StarbucksCount	1	0.92	0.32	0.91	0.92	0.46	0.92	0.91
TotalPop	0.92	1	0.4	1	1	0.45	1	1
Transit	0.32	0.4	1	0.41	0.41	0.44	0.4	0.41
/otingAgeCitizen	0.91	1	0.41	1	1	0.46	1	1
Employed	0.92	1	0.41	1	1	0.47	1	1
Asian	0.46	0.45	0.44	0.46	0.47	1	0.45	0.45
Men	0.92	1	0.4	1	1	0.45	1	1
Women	0.91	1	0.41	1	1	0.45	1	1

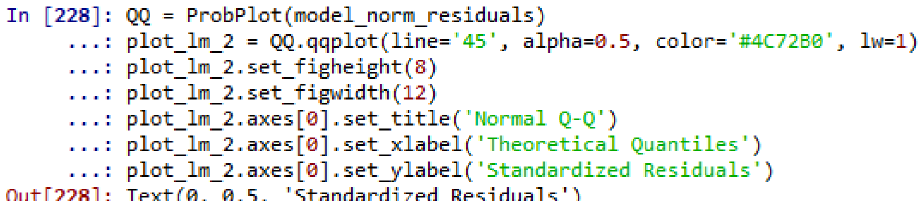


```
In [227]: """
     ...: Decision of transformation variables based on residual and normal plots
     ...:
     ...: """
     ...: """
     ...: Case 1 :
     ...: Variables as StarbucksCount ~ TotalPop  + Asian + IncomePerCap + Professional
     ...:
     ...: """
     ...: reg = smf.ols('StarbucksCount ~ TotalPop  + Asian + IncomePerCap + Professional', data=mergedData).fit()
     ...: print(reg.summary())
     ...:
     ...: model_fitted_y = reg.fittedvalues
     ...: model_residuals = reg.resid
     ...: model_norm_residuals = reg.get_influence().resid_studentized_internal
     ...:
     ...: plot_lm_1 = plt.figure(1)
     ...: plot_lm_1.axes[0] = sns.residplot(model_fitted_y, 'StarbucksCount', data=mergedData,lowess=True,
     ...:               scatter_kws={'alpha': 0.5},line_kws={'color': 'red', 'lw': 1, 'alpha': 0.8})
     ...: plot_lm_1.set_figheight(8)
     ...: plot_lm_1.set_figwidth(12)
     ...: plot_lm_1.axes[0].set_title('Residuals vs Fitted')
     ...: plot_lm_1.axes[0].set_xlabel('Fitted values')
     ...: plot_lm_1.axes[0].set_ylabel('Residuals')
```

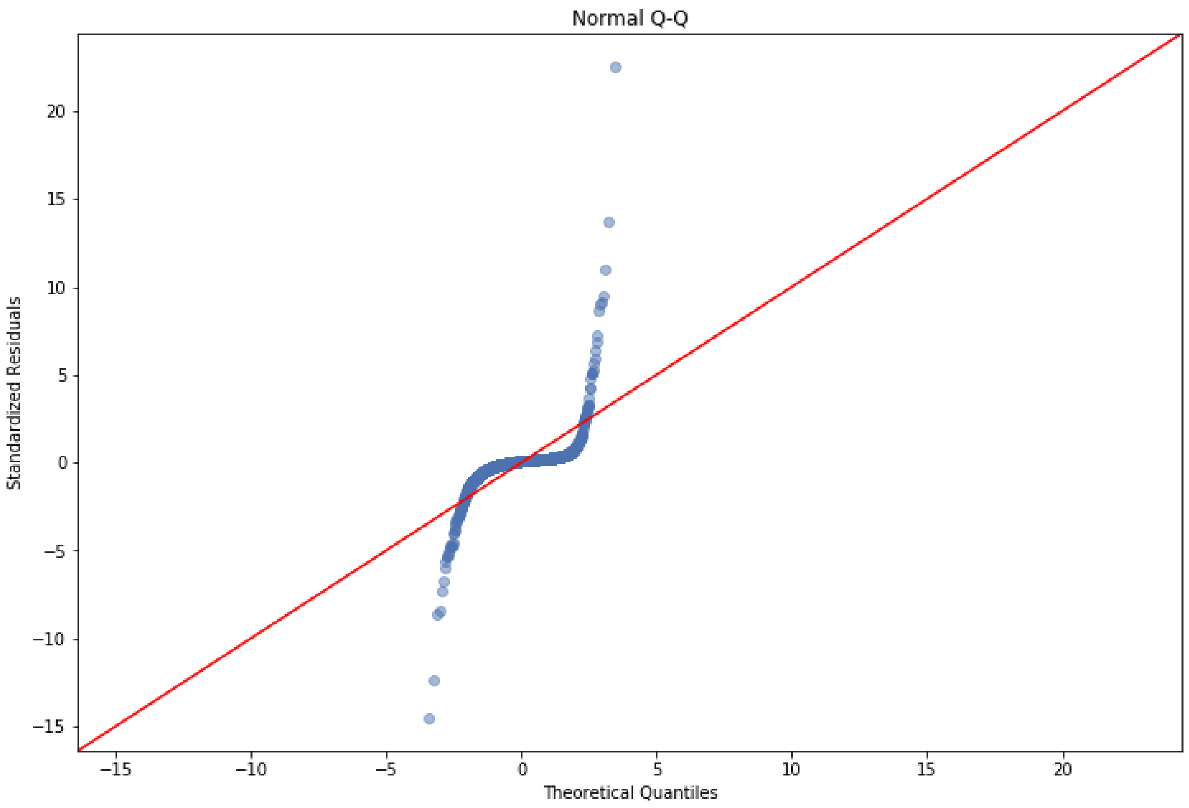


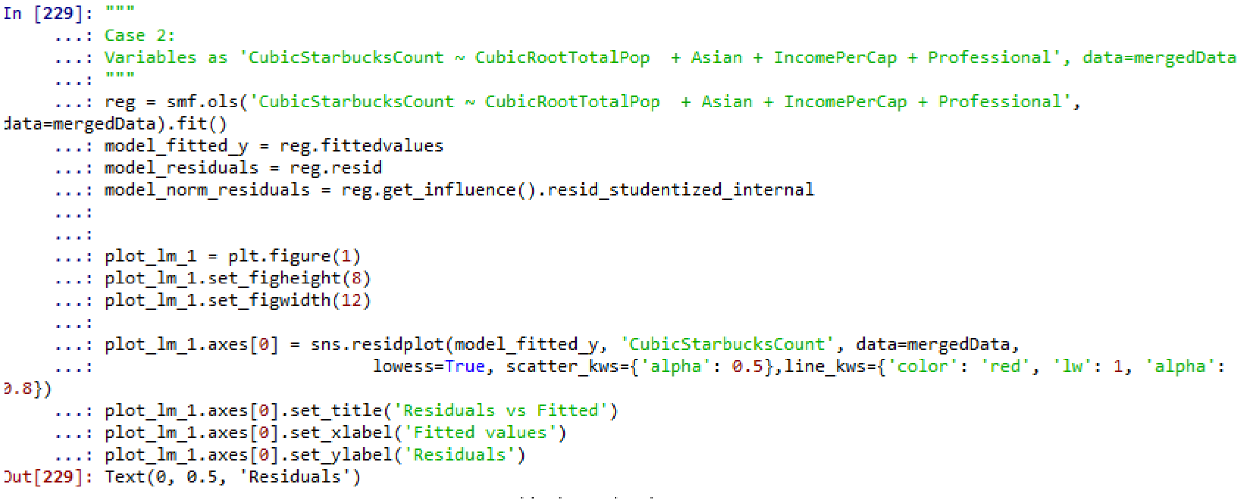
```
                            OLS Regression Results                            
==============================================================================
Dep. Variable:         StarbucksCount   R-squared:                       0.843
Model:                            OLS   Adj. R-squared:                  0.843
Method:                 Least Squares   F-statistic:                     4314.
Date:                Sun, 05 Apr 2020   Prob (F-statistic):               0.00
Time:                        23:03:18   Log-Likelihood:                -11702.
No. Observations:                3218   AIC:                         2.341e+04
Df Residuals:                    3213   BIC:                         2.344e+04
Df Model:                           4                                         
Covariance Type:            nonrobust                                         
================================================================================
                   coef    std err          t      P>|t|      [0.025      0.975]
--------------------------------------------------------------------------------
Intercept       -5.4607      0.857     -6.371      0.000      -7.141      -3.780
TotalPop      6.337e-05   5.63e-07    112.582      0.000    6.23e-05    6.45e-05
Asian            0.3590      0.072      5.011      0.000       0.219       0.499
IncomePerCap     0.0001   3.35e-05      3.154      0.002       4e-05       0.000
Professional     0.0333      0.034      0.995      0.320      -0.032       0.099
==============================================================================
Omnibus:                     2984.106   Durbin-Watson:                   1.747
Prob(Omnibus):                  0.000   Jarque-Bera (JB):          2296466.911
Skew:                           3.383   Prob(JB):                         0.00
Kurtosis:                     133.696   Cond. No.                     1.80e+06
==============================================================================

Warnings:
[1] Standard Errors assume that the covariance matrix of the errors is correctly specified.
[2] The condition number is large, 1.8e+06. This might indicate that there are
strong multicollinearity or other numerical problems.
```



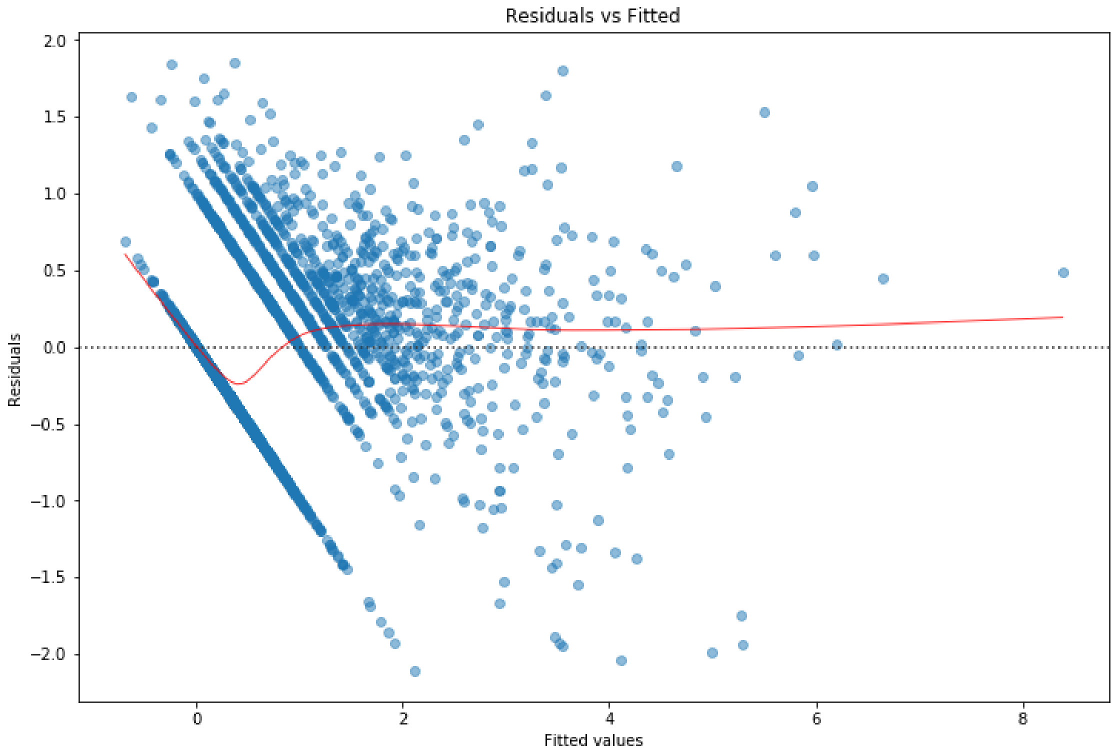


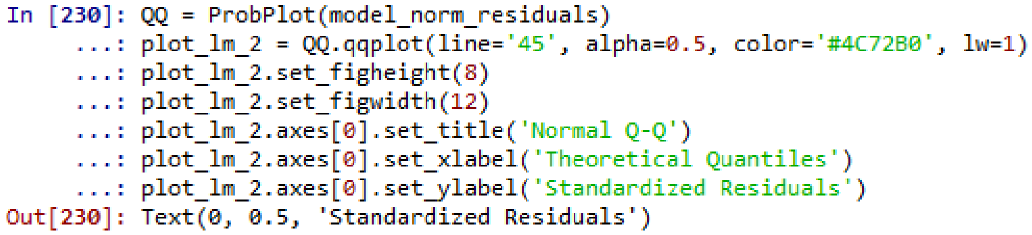
```
In [228]: QQ = ProbPlot(model_norm_residuals)
     ...: plot_lm_2 = QQ.qqplot(line='45', alpha=0.5, color='#4C72B0', lw=1)
     ...: plot_lm_2.set_figheight(8)
     ...: plot_lm_2.set_figwidth(12)
     ...: plot_lm_2.axes[0].set_title('Normal Q-Q')
     ...: plot_lm_2.axes[0].set_xlabel('Theoretical Quantiles')
     ...: plot_lm_2.axes[0].set_ylabel('Standardized Residuals')
Out[228]: Text(0, 0.5, 'Standardized Residuals')
```

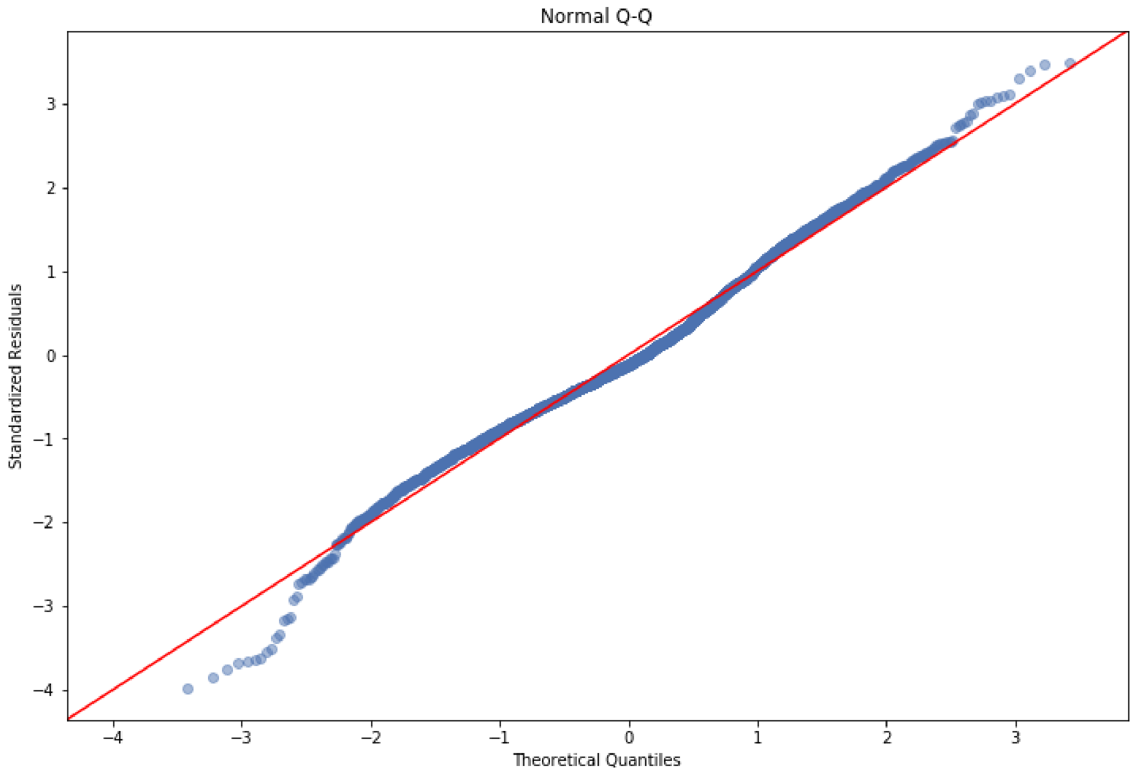


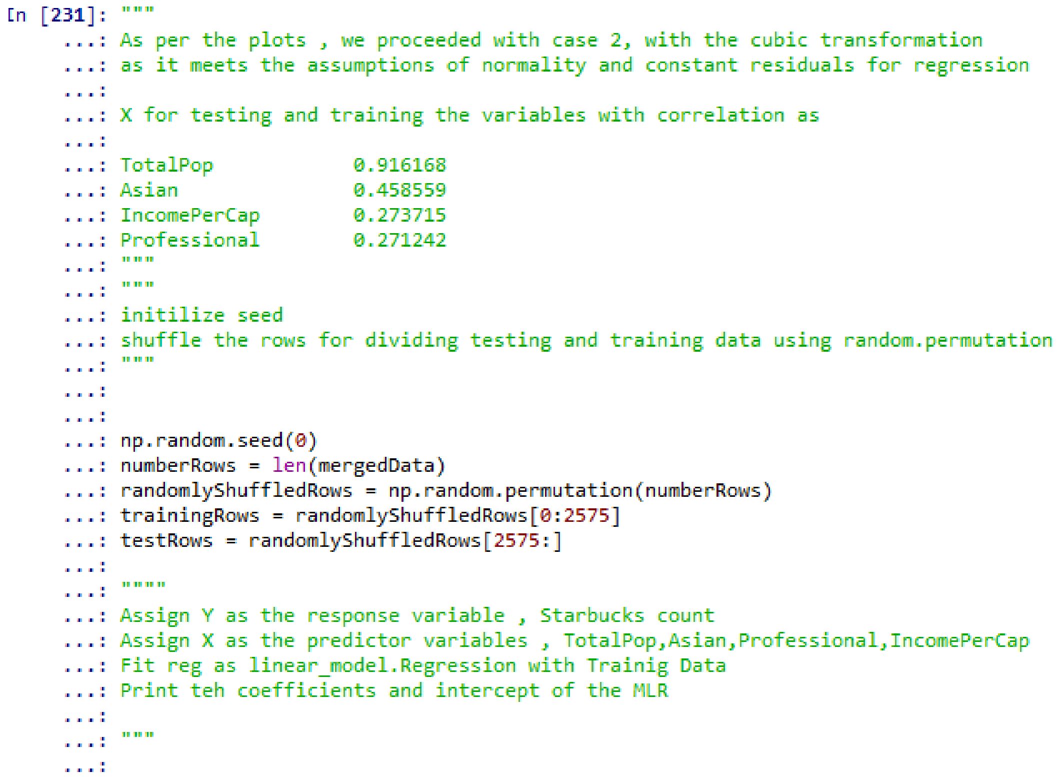


```
In [229]: """
     ...: Case 2:
     ...: Variables as 'CubicStarbucksCount ~ CubicRootTotalPop  + Asian + IncomePerCap + Professional', data=mergedData
     ...: """
     ...: reg = smf.ols('CubicStarbucksCount ~ CubicRootTotalPop  + Asian + IncomePerCap + Professional',
data=mergedData).fit()
     ...: model_fitted_y = reg.fittedvalues
     ...: model_residuals = reg.resid
     ...: model_norm_residuals = reg.get_influence().resid_studentized_internal
     ...:
     ...:
     ...: plot_lm_1 = plt.figure(1)
     ...: plot_lm_1.set_figheight(8)
     ...: plot_lm_1.set_figwidth(12)
     ...:
     ...: plot_lm_1.axes[0] = sns.residplot(model_fitted_y, 'CubicStarbucksCount', data=mergedData,
     ...:                           lowess=True, scatter_kws={'alpha': 0.5},line_kws={'color': 'red', 'lw': 1, 'alpha':
0.8})
     ...: plot_lm_1.axes[0].set_title('Residuals vs Fitted')
     ...: plot_lm_1.axes[0].set_xlabel('Fitted values')
     ...: plot_lm_1.axes[0].set_ylabel('Residuals')
Out[229]: Text(0, 0.5, 'Residuals')
```

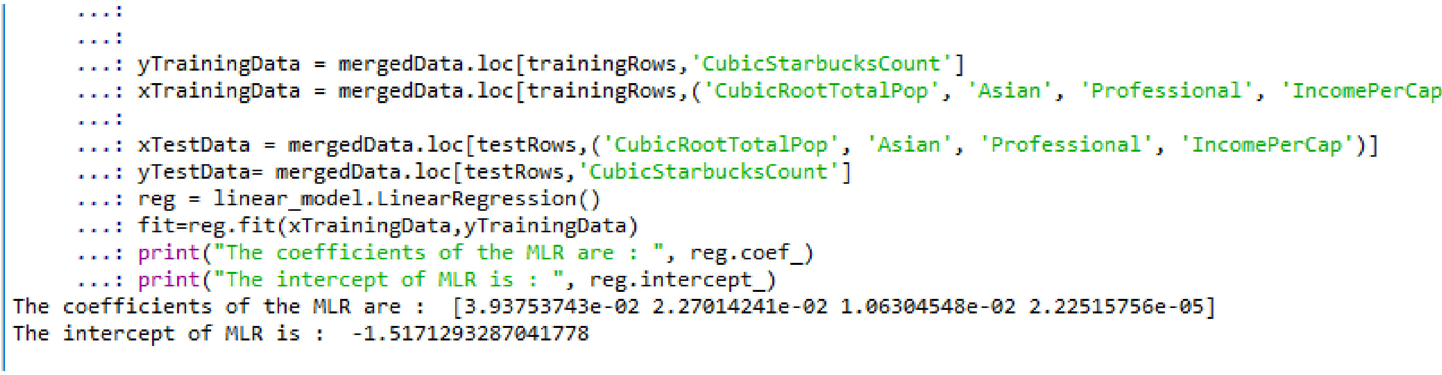



```
In [230]: QQ = ProbPlot(model_norm_residuals)
     ...: plot_lm_2 = QQ.qqplot(line='45', alpha=0.5, color='#4C72B0', lw=1)
     ...: plot_lm_2.set_figheight(8)
     ...: plot_lm_2.set_figwidth(12)
     ...: plot_lm_2.axes[0].set_title('Normal Q-Q')
     ...: plot_lm_2.axes[0].set_xlabel('Theoretical Quantiles')
     ...: plot_lm_2.axes[0].set_ylabel('Standardized Residuals')
Out[230]: Text(0, 0.5, 'Standardized Residuals')
```

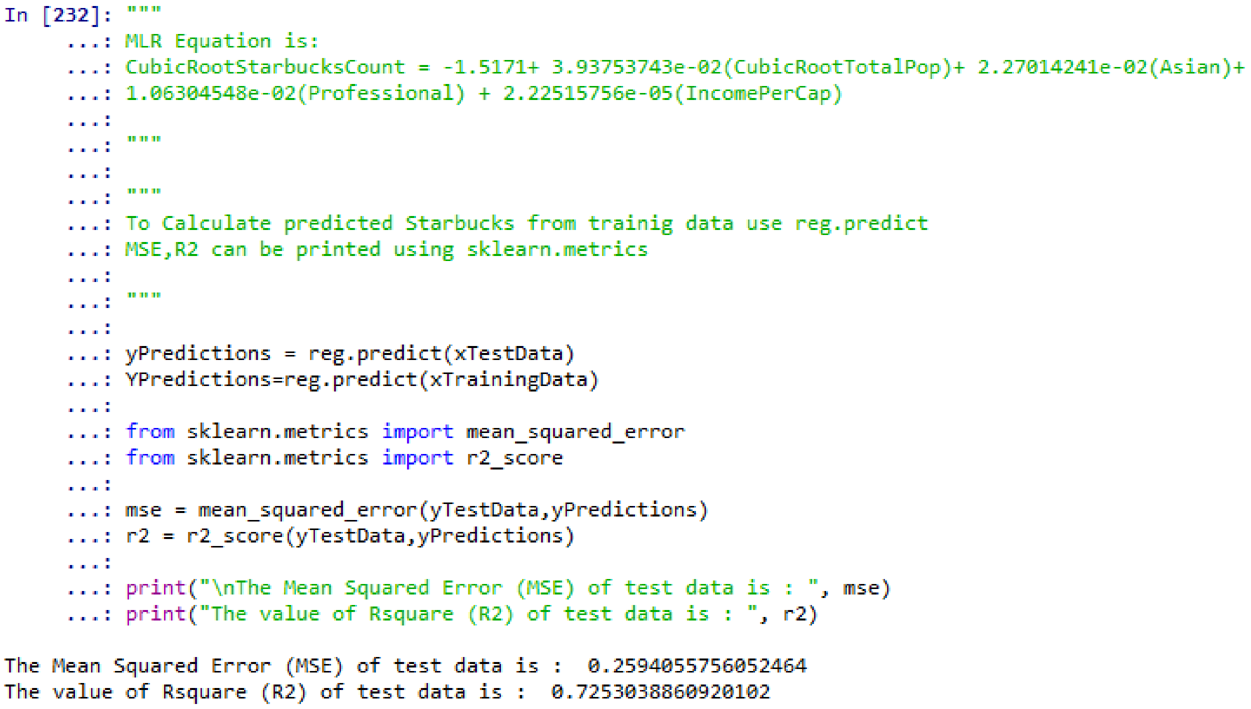




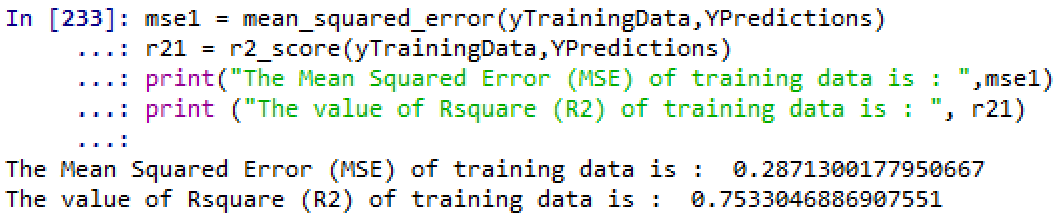
```
In [231]: """
     ...: As per the plots , we proceeded with case 2, with the cubic transformation
     ...: as it meets the assumptions of normality and constant residuals for regression
     ...:
     ...: X for testing and training the variables with correlation as
     ...:
     ...: TotalPop            0.916168
     ...: Asian               0.458559
     ...: IncomePerCap        0.273715
     ...: Professional        0.271242
     ...: """
     ...: """
     ...: initilize seed
     ...: shuffle the rows for dividing testing and training data using random.permutation
     ...: """
     ...:
     ...:
     ...: np.random.seed(0)
     ...: numberRows = len(mergedData)
     ...: randomlyShuffledRows = np.random.permutation(numberRows)
     ...: trainingRows = randomlyShuffledRows[0:2575]
     ...: testRows = randomlyShuffledRows[2575:]
     ...:
     ...: """"
     ...: Assign Y as the response variable , Starbucks count
     ...: Assign X as the predictor variables , TotalPop,Asian,Professional,IncomePerCap
     ...: Fit reg as linear_model.Regression with Trainig Data
     ...: Print teh coefficients and intercept of the MLR
     ...:
     ...: """
     ...:
```



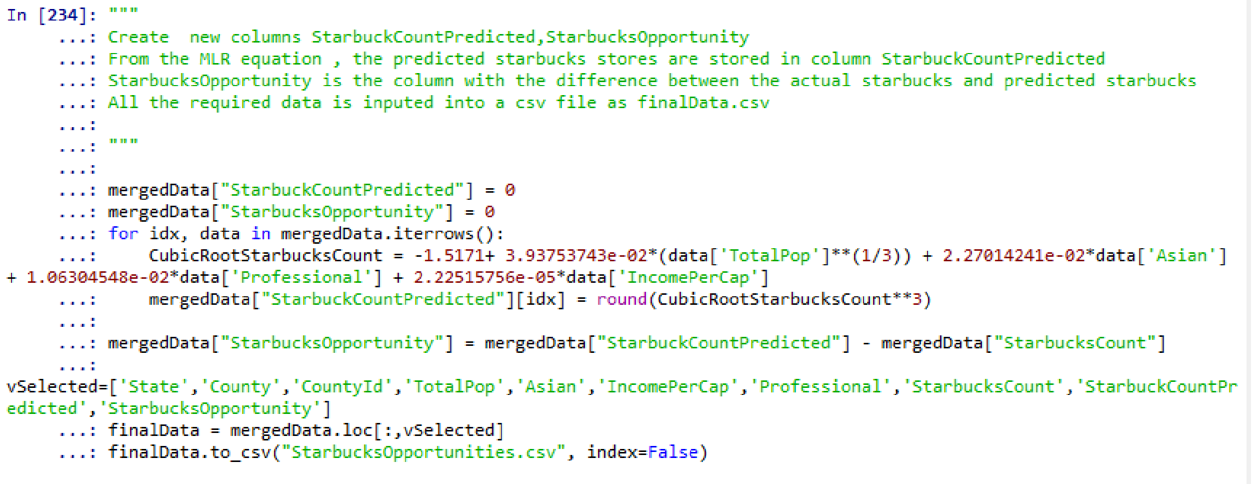
```
     ...:
     ...:
     ...: yTrainingData = mergedData.loc[trainingRows,'CubicStarbucksCount']
     ...: xTrainingData = mergedData.loc[trainingRows,('CubicRootTotalPop', 'Asian', 'Professional', 'IncomePerCap
     ...:
     ...: xTestData = mergedData.loc[testRows,('CubicRootTotalPop', 'Asian', 'Professional', 'IncomePerCap')]
     ...: yTestData= mergedData.loc[testRows,'CubicStarbucksCount']
     ...: reg = linear_model.LinearRegression()
     ...: fit=reg.fit(xTrainingData,yTrainingData)
     ...: print("The coefficients of the MLR are : ", reg.coef_)
     ...: print("The intercept of MLR is : ", reg.intercept_)
The coefficients of the MLR are :  [3.93753743e-02 2.27014241e-02 1.06304548e-02 2.22515756e-05]
The intercept of MLR is :  -1.5171293287041778
```



```
In [232]: """
     ...: MLR Equation is:
     ...: CubicRootStarbucksCount = -1.5171+ 3.93753743e-02(CubicRootTotalPop)+ 2.27014241e-02(Asian)+
     ...: 1.06304548e-02(Professional) + 2.22515756e-05(IncomePerCap)
     ...:
     ...: """
     ...:
     ...: """
     ...: To Calculate predicted Starbucks from trainig data use reg.predict
     ...: MSE,R2 can be printed using sklearn.metrics
     ...:
     ...: """
     ...:
     ...: yPredictions = reg.predict(xTestData)
     ...: YPredictions=reg.predict(xTrainingData)
     ...:
     ...: from sklearn.metrics import mean_squared_error
     ...: from sklearn.metrics import r2_score
     ...:
     ...: mse = mean_squared_error(yTestData,yPredictions)
     ...: r2 = r2_score(yTestData,yPredictions)
     ...:
     ...: print("\nThe Mean Squared Error (MSE) of test data is : ", mse)
     ...: print("The value of Rsquare (R2) of test data is : ", r2)

The Mean Squared Error (MSE) of test data is :  0.2594055756052464
The value of Rsquare (R2) of test data is :  0.7253038860920102
```

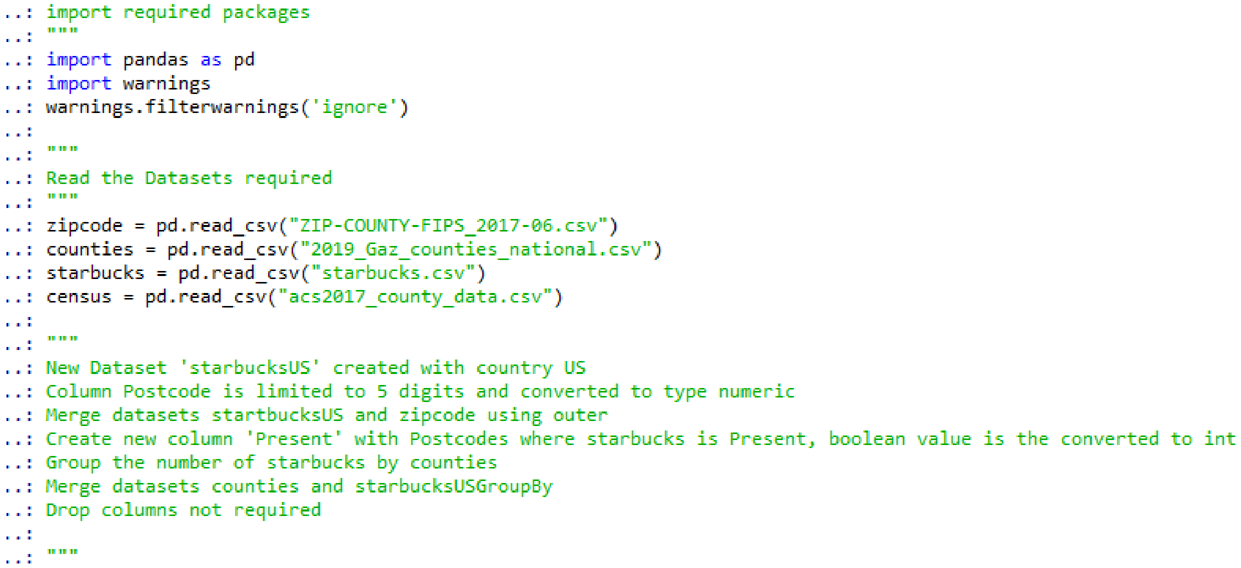


```
In [233]: mse1 = mean_squared_error(yTrainingData,YPredictions)
     ...: r21 = r2_score(yTrainingData,YPredictions)
     ...: print("The Mean Squared Error (MSE) of training data is : ",mse1)
     ...: print ("The value of Rsquare (R2) of training data is : ", r21)
     ...:
The Mean Squared Error (MSE) of training data is :  0.2871300177950667
The value of Rsquare (R2) of training data is :  0.7533046886907551
```

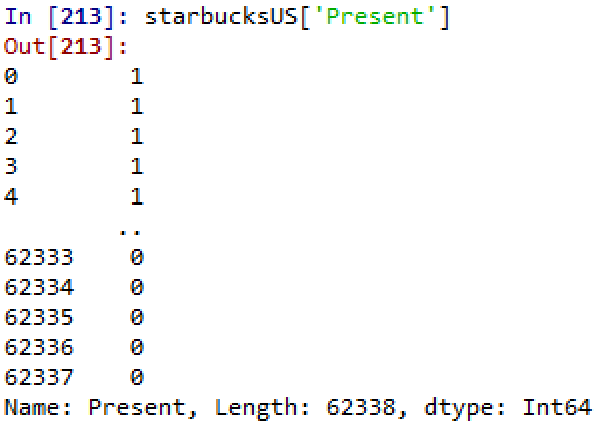


```
In [234]: """
     ...: Create  new columns StarbuckCountPredicted,StarbucksOpportunity
     ...: From the MLR equation , the predicted starbucks stores are stored in column StarbuckCountPredicted
     ...: StarbucksOpportunity is the column with the difference between the actual starbucks and predicted starbucks
     ...: All the required data is inputed into a csv file as finalData.csv
     ...:
     ...: """
     ...:
     ...: mergedData["StarbuckCountPredicted"] = 0
     ...: mergedData["StarbucksOpportunity"] = 0
     ...: for idx, data in mergedData.iterrows():
     ...:     CubicRootStarbucksCount = -1.5171+ 3.93753743e-02*(data['TotalPop']**(1/3)) + 2.27014241e-02*data['Asian']
+ 1.06304548e-02*data['Professional'] + 2.22515756e-05*data['IncomePerCap']
     ...:     mergedData["StarbuckCountPredicted"][idx] = round(CubicRootStarbucksCount**3)
     ...:
     ...: mergedData["StarbucksOpportunity"] = mergedData["StarbuckCountPredicted"] - mergedData["StarbucksCount"]
     ...:
vSelected=['State','County','CountyId','TotalPop','Asian','IncomePerCap','Professional','StarbucksCount','StarbuckCountPr
edicted','StarbucksOpportunity']
     ...: finalData = mergedData.loc[:,vSelected]
     ...: finalData.to_csv("StarbucksOpportunities.csv", index=False)
```

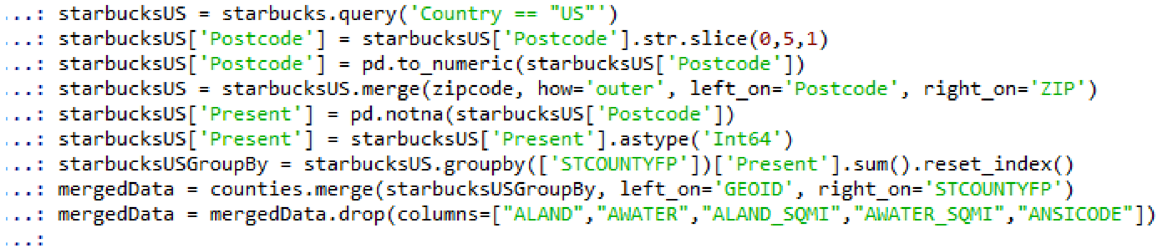
I B: K Nearest Neighbor (KNN) Algorithm:



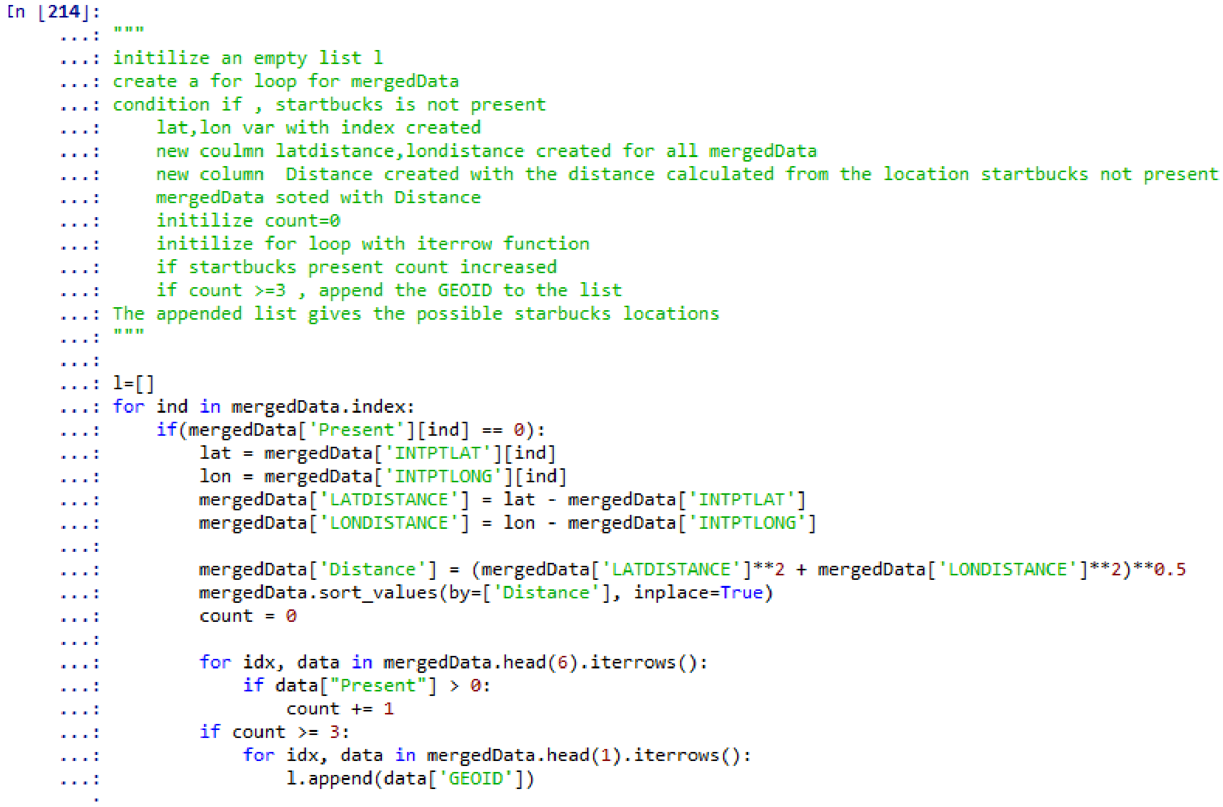
```
..: import required packages
..: """
..: import pandas as pd
..: import warnings
..: warnings.filterwarnings('ignore')
..:
..: """
..: Read the Datasets required
..: """
..: zipcode = pd.read_csv("ZIP-COUNTY-FIPS_2017-06.csv")
..: counties = pd.read_csv("2019_Gaz_counties_national.csv")
..: starbucks = pd.read_csv("starbucks.csv")
..: census = pd.read_csv("acs2017_county_data.csv")
..:
..: """
..: New Dataset 'starbucksUS' created with country US
..: Column Postcode is limited to 5 digits and converted to type numeric
..: Merge datasets startbucksUS and zipcode using outer
..: Create new column 'Present' with Postcodes where starbucks is Present, boolean value is the converted to int
..: Group the number of starbucks by counties
..: Merge datasets counties and starbucksUSGroupBy
..: Drop columns not required
..:
..: """
```



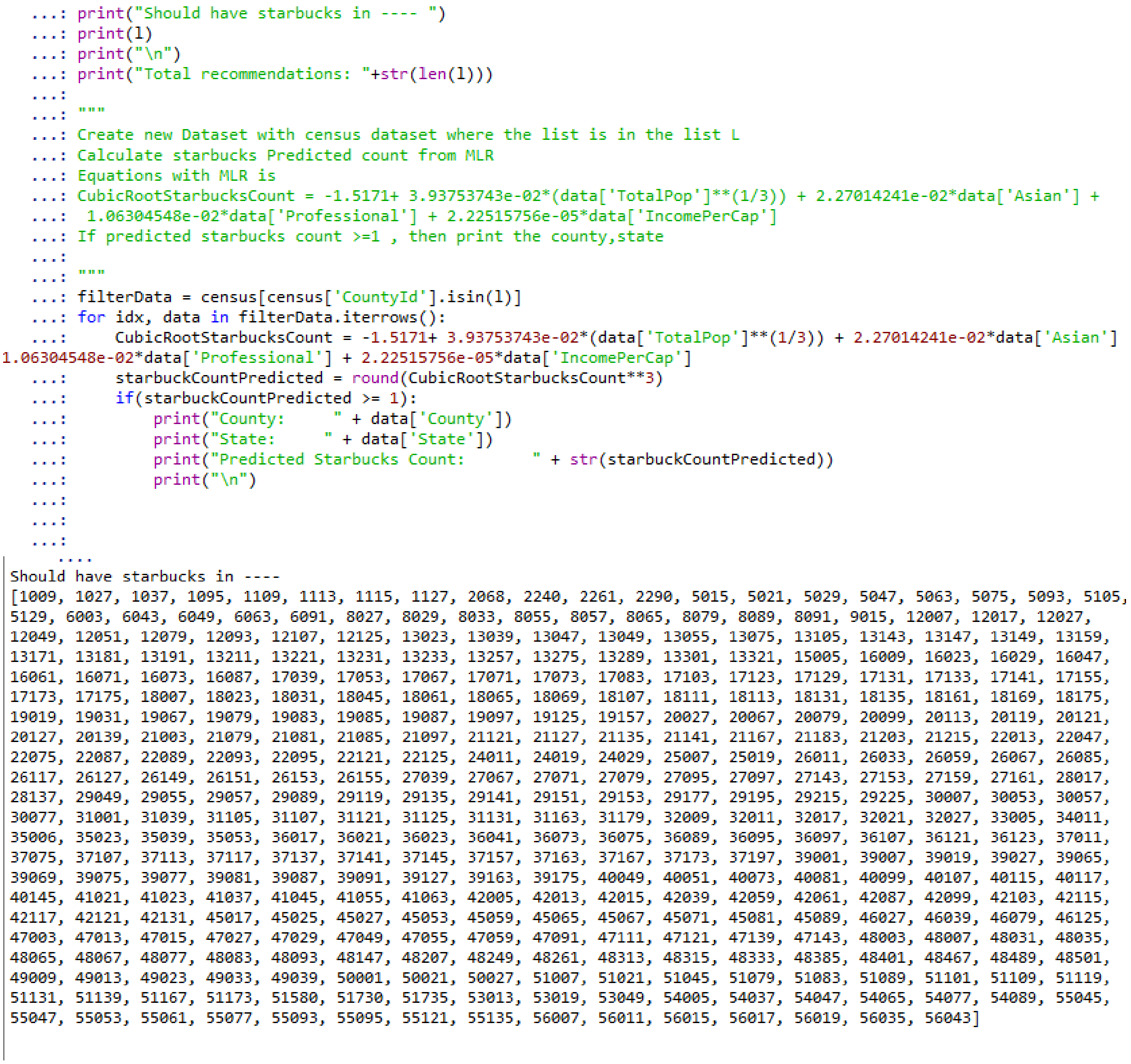
```
In [213]: starbucksUS['Present']
Out[213]:
0        1
1        1
2        1
3        1
4        1
        ..
62333    0
62334    0
62335    0
62336    0
62337    0
Name: Present, Length: 62338, dtype: Int64
```



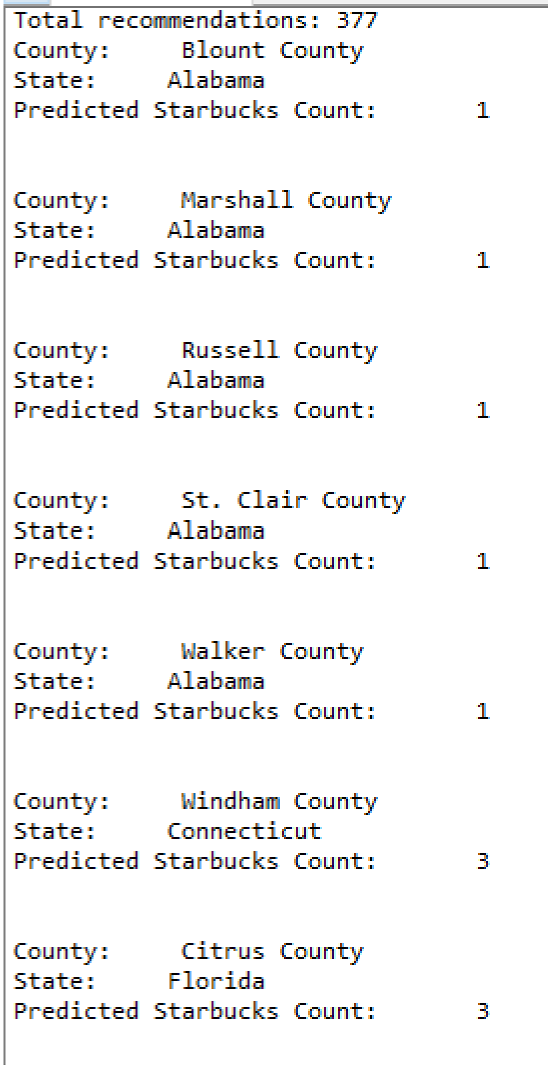
```
...: starbucksUS = starbucks.query('Country == "US"')
...: starbucksUS['Postcode'] = starbucksUS['Postcode'].str.slice(0,5,1)
...: starbucksUS['Postcode'] = pd.to_numeric(starbucksUS['Postcode'])
...: starbucksUS = starbucksUS.merge(zipcode, how='outer', left_on='Postcode', right_on='ZIP')
...: starbucksUS['Present'] = pd.notna(starbucksUS['Postcode'])
...: starbucksUS['Present'] = starbucksUS['Present'].astype('Int64')
...: starbucksUSGroupBy = starbucksUS.groupby(['STCOUNTYFP'])['Present'].sum().reset_index()
...: mergedData = counties.merge(starbucksUSGroupBy, left_on='GEOID', right_on='STCOUNTYFP')
...: mergedData = mergedData.drop(columns=["ALAND","AWATER","ALAND_SQMI","AWATER_SQMI","ANSICODE"])
...:
```



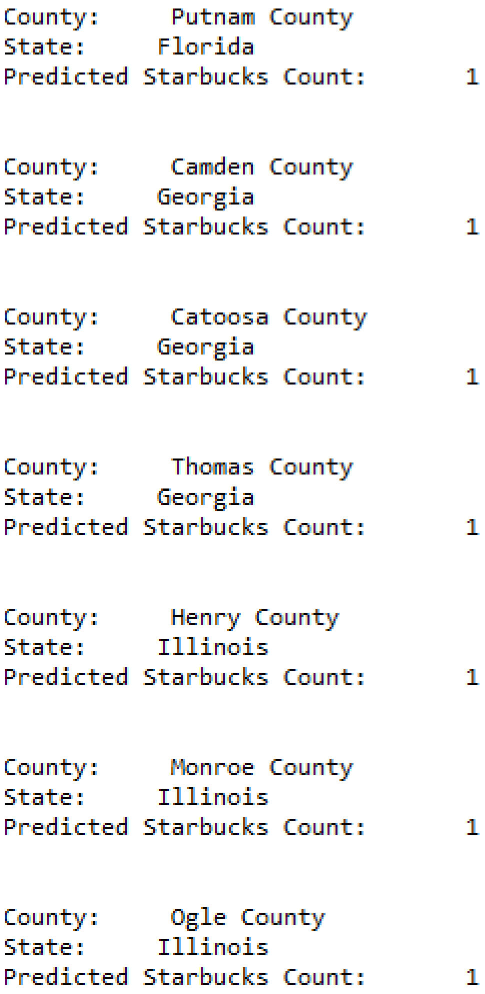
```
In [214]:
    ...: """
    ...: initilize an empty list l
    ...: create a for loop for mergedData
    ...: condition if , startbucks is not present
    ...:     lat,lon var with index created
    ...:     new coulmn latdistance,londistance created for all mergedData
    ...:     new column  Distance created with the distance calculated from the location startbucks not present
    ...:     mergedData soted with Distance
    ...:     initilize count=0
    ...:     initilize for loop with iterrow function
    ...:     if startbucks present count increased
    ...:     if count >=3 , append the GEOID to the list
    ...: The appended list gives the possible starbucks locations
    ...: """
    ...:
    ...: l=[]
    ...: for ind in mergedData.index:
    ...:     if(mergedData['Present'][ind] == 0):
    ...:         lat = mergedData['INTPTLAT'][ind]
    ...:         lon = mergedData['INTPTLONG'][ind]
    ...:         mergedData['LATDISTANCE'] = lat - mergedData['INTPTLAT']
    ...:         mergedData['LONDISTANCE'] = lon - mergedData['INTPTLONG']
    ...:
    ...:         mergedData['Distance'] = (mergedData['LATDISTANCE']**2 + mergedData['LONDISTANCE']**2)**0.5
    ...:         mergedData.sort_values(by=['Distance'], inplace=True)
    ...:         count = 0
    ...:
    ...:         for idx, data in mergedData.head(6).iterrows():
    ...:             if data["Present"] > 0:
    ...:                 count += 1
    ...:         if count >= 3:
    ...:             for idx, data in mergedData.head(1).iterrows():
    ...:                 l.append(data['GEOID'])
```



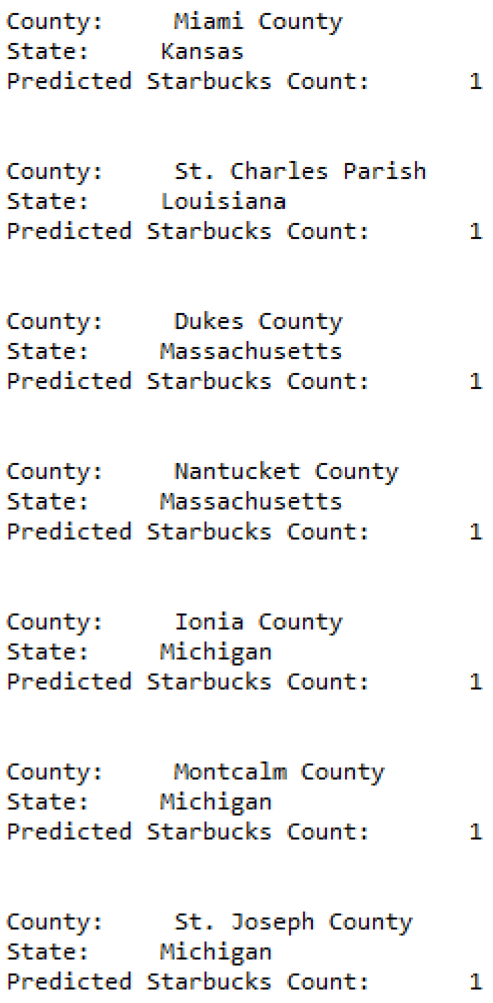
```
   ...: print("Should have starbucks in ---- ")
   ...: print(l)
   ...: print("\n")
   ...: print("Total recommendations: "+str(len(l)))
   ...:
   ...: """
   ...: Create new Dataset with census dataset where the list is in the list L
   ...: Calculate starbucks Predicted count from MLR
   ...: Equations with MLR is
   ...: CubicRootStarbucksCount = -1.5171+ 3.93753743e-02*(data['TotalPop']**(1/3)) + 2.27014241e-02*data['Asian'] +
   ...:  1.06304548e-02*data['Professional'] + 2.22515756e-05*data['IncomePerCap']
   ...: If predicted starbucks count >=1 , then print the county,state
   ...:
   ...: """
   ...: filterData = census[census['CountyId'].isin(l)]
   ...: for idx, data in filterData.iterrows():
   ...:     CubicRootStarbucksCount = -1.5171+ 3.93753743e-02*(data['TotalPop']**(1/3)) + 2.27014241e-02*data['Asian']
1.06304548e-02*data['Professional'] + 2.22515756e-05*data['IncomePerCap']
   ...:     starbuckCountPredicted = round(CubicRootStarbucksCount**3)
   ...:     if(starbuckCountPredicted >= 1):
   ...:         print("County:     " + data['County'])
   ...:         print("State:     " + data['State'])
   ...:         print("Predicted Starbucks Count:       " + str(starbuckCountPredicted))
   ...:         print("\n")
   ...:
   ...:
   ...:
     ....
Should have starbucks in ----
[1009, 1027, 1037, 1095, 1109, 1113, 1115, 1127, 2068, 2240, 2261, 2290, 5015, 5021, 5029, 5047, 5063, 5075, 5093, 5105,
5129, 6003, 6043, 6049, 6063, 6091, 8027, 8029, 8033, 8055, 8057, 8065, 8079, 8089, 8091, 9015, 12007, 12017, 12027,
12049, 12051, 12079, 12093, 12107, 12125, 13023, 13039, 13047, 13049, 13055, 13075, 13105, 13143, 13147, 13149, 13159,
13171, 13181, 13191, 13211, 13221, 13231, 13233, 13257, 13275, 13289, 13301, 13321, 15005, 16009, 16023, 16029, 16047,
16061, 16071, 16073, 16087, 17039, 17053, 17067, 17071, 17073, 17083, 17103, 17123, 17129, 17131, 17133, 17141, 17155,
17173, 17175, 18007, 18023, 18031, 18045, 18061, 18065, 18069, 18107, 18111, 18113, 18131, 18135, 18161, 18169, 18175,
19019, 19031, 19067, 19079, 19083, 19085, 19087, 19097, 19125, 19157, 20027, 20067, 20079, 20099, 20113, 20119, 20121,
20127, 20139, 21003, 21079, 21081, 21085, 21097, 21121, 21127, 21135, 21141, 21167, 21183, 21203, 21215, 22013, 22047,
22075, 22087, 22089, 22093, 22095, 22121, 22125, 24011, 24019, 24029, 25007, 25019, 26011, 26033, 26059, 26067, 26085,
26117, 26127, 26149, 26151, 26153, 26155, 27039, 27067, 27071, 27079, 27095, 27097, 27143, 27153, 27159, 27161, 28017,
28137, 29049, 29055, 29057, 29089, 29119, 29135, 29141, 29151, 29153, 29177, 29195, 29215, 29225, 30007, 30053, 30057,
30077, 31001, 31039, 31105, 31107, 31121, 31125, 31131, 31163, 31179, 32009, 32011, 32017, 32021, 32027, 33005, 34011,
35006, 35023, 35039, 35053, 36017, 36021, 36023, 36041, 36073, 36075, 36089, 36095, 36097, 36107, 36121, 36123, 37011,
37075, 37107, 37113, 37117, 37137, 37141, 37145, 37157, 37163, 37167, 37173, 37197, 39001, 39007, 39019, 39027, 39065,
39069, 39075, 39077, 39081, 39087, 39091, 39127, 39163, 39175, 40049, 40051, 40073, 40081, 40099, 40107, 40115, 40117,
40145, 41021, 41023, 41037, 41045, 41055, 41063, 42005, 42013, 42015, 42039, 42059, 42061, 42087, 42099, 42103, 42115,
42117, 42121, 42131, 45017, 45025, 45027, 45053, 45059, 45065, 45067, 45071, 45081, 45089, 46027, 46039, 46079, 46125,
47003, 47013, 47015, 47027, 47029, 47049, 47055, 47059, 47091, 47111, 47121, 47139, 47143, 48003, 48007, 48031, 48035,
48065, 48067, 48077, 48083, 48093, 48147, 48207, 48249, 48261, 48313, 48315, 48333, 48385, 48401, 48467, 48489, 48501,
49009, 49013, 49023, 49033, 49039, 50001, 50021, 50027, 51007, 51021, 51045, 51079, 51083, 51089, 51101, 51109, 51119,
51131, 51139, 51167, 51173, 51580, 51730, 51735, 53013, 53019, 53049, 54005, 54037, 54047, 54065, 54077, 54089, 55045,
55047, 55053, 55061, 55077, 55093, 55095, 55121, 55135, 56007, 56011, 56015, 56017, 56019, 56035, 56043]
```



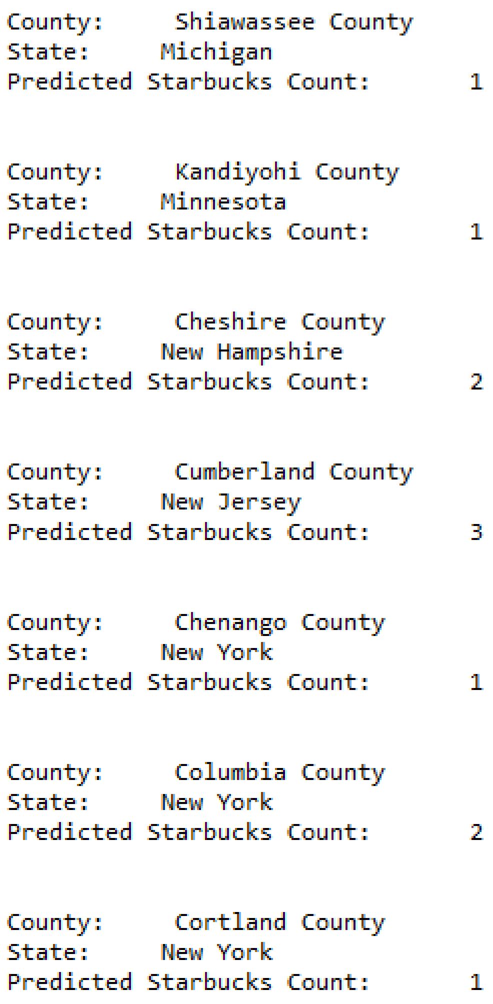
```
Total recommendations: 377
County:     Blount County
State:     Alabama
Predicted Starbucks Count:       1

County:     Marshall County
State:     Alabama
Predicted Starbucks Count:       1

County:     Russell County
State:     Alabama
Predicted Starbucks Count:       1

County:     St. Clair County
State:     Alabama
Predicted Starbucks Count:       1

County:     Walker County
State:     Alabama
Predicted Starbucks Count:       1

County:     Windham County
State:     Connecticut
Predicted Starbucks Count:       3

County:     Citrus County
State:     Florida
Predicted Starbucks Count:       3
```



```
County:     Putnam County
State:     Florida
Predicted Starbucks Count:       1

County:     Camden County
State:     Georgia
Predicted Starbucks Count:       1

County:     Catoosa County
State:     Georgia
Predicted Starbucks Count:       1

County:     Thomas County
State:     Georgia
Predicted Starbucks Count:       1

County:     Henry County
State:     Illinois
Predicted Starbucks Count:       1

County:     Monroe County
State:     Illinois
Predicted Starbucks Count:       1

County:     Ogle County
State:     Illinois
Predicted Starbucks Count:       1
```

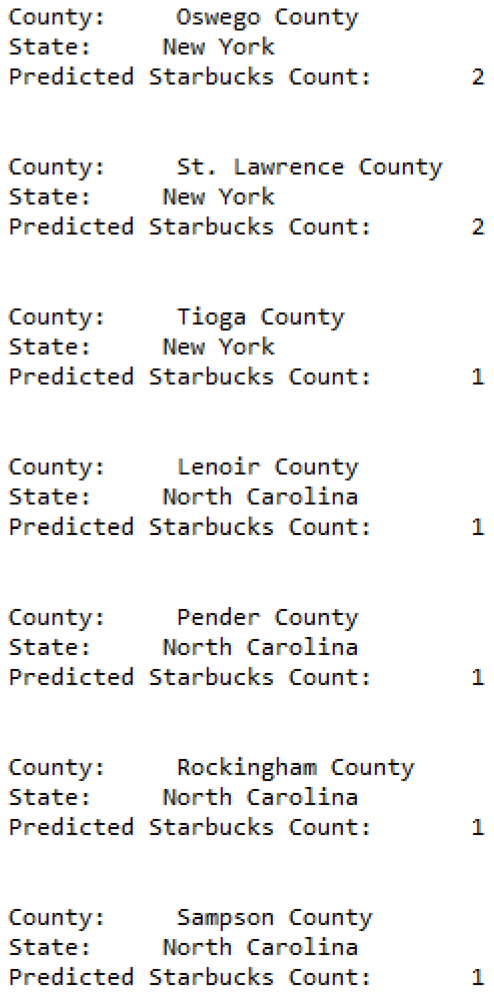


```
County:     Miami County
State:     Kansas
Predicted Starbucks Count:        1


County:     St. Charles Parish
State:     Louisiana
Predicted Starbucks Count:        1


County:     Dukes County
State:     Massachusetts
Predicted Starbucks Count:        1


County:     Nantucket County
State:     Massachusetts
Predicted Starbucks Count:        1


County:     Ionia County
State:     Michigan
Predicted Starbucks Count:        1


County:     Montcalm County
State:     Michigan
Predicted Starbucks Count:        1


County:     St. Joseph County
State:     Michigan
Predicted Starbucks Count:        1
```

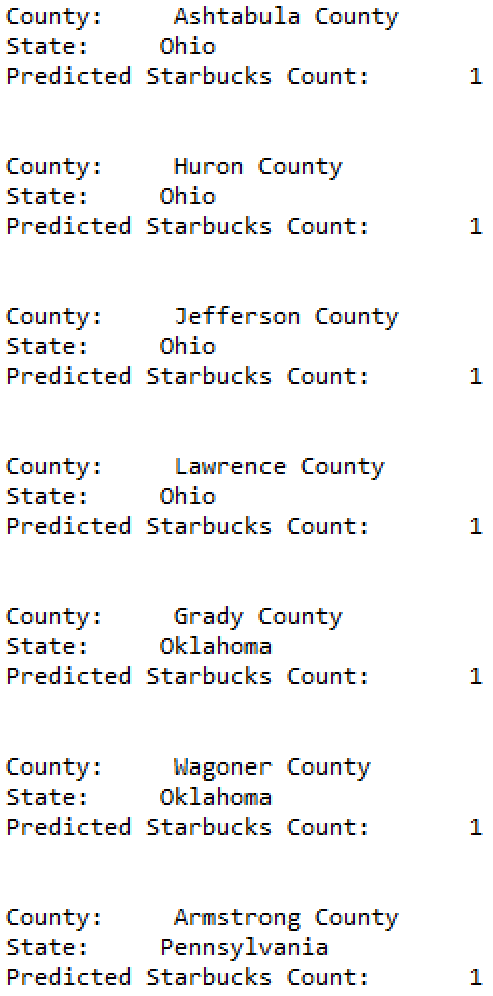


```
County:     Shiawassee County
State:     Michigan
Predicted Starbucks Count:        1


County:     Kandiyohi County
State:     Minnesota
Predicted Starbucks Count:        1


County:     Cheshire County
State:     New Hampshire
Predicted Starbucks Count:        2


County:     Cumberland County
State:     New Jersey
Predicted Starbucks Count:        3


County:     Chenango County
State:     New York
Predicted Starbucks Count:        1


County:     Columbia County
State:     New York
Predicted Starbucks Count:        2


County:     Cortland County
State:     New York
Predicted Starbucks Count:        1
```

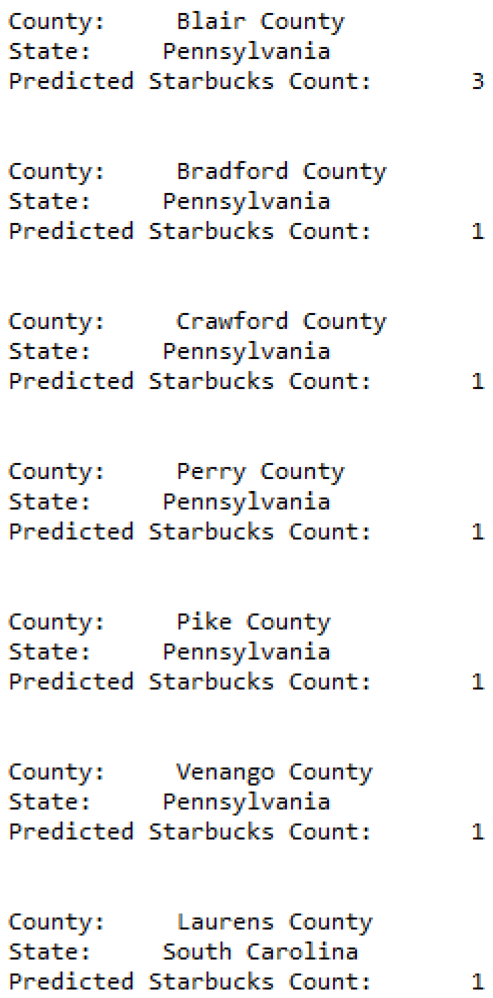
Finding Future Potential Starbucks locations



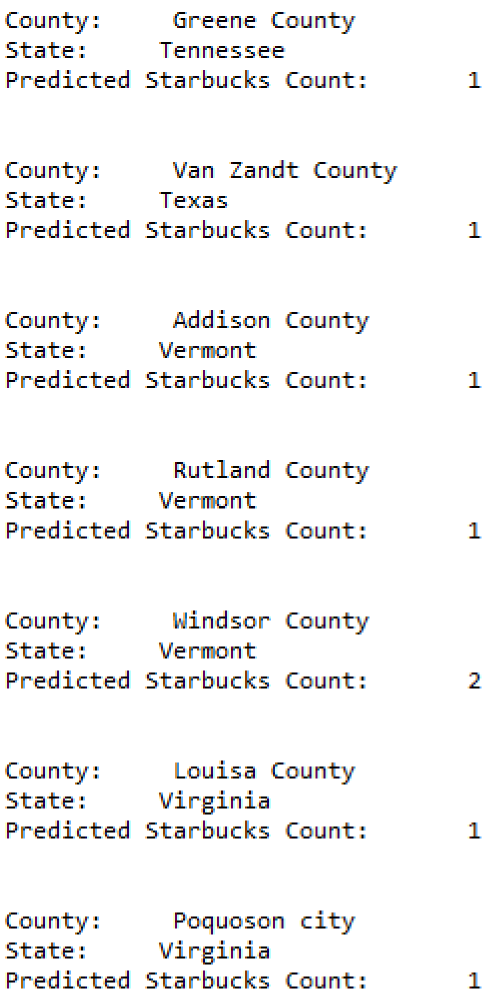
```
County:     Oswego County
State:     New York
Predicted Starbucks Count:       2


County:     St. Lawrence County
State:     New York
Predicted Starbucks Count:       2


County:     Tioga County
State:     New York
Predicted Starbucks Count:       1


County:     Lenoir County
State:     North Carolina
Predicted Starbucks Count:       1


County:     Pender County
State:     North Carolina
Predicted Starbucks Count:       1


County:     Rockingham County
State:     North Carolina
Predicted Starbucks Count:       1


County:     Sampson County
State:     North Carolina
Predicted Starbucks Count:       1
```



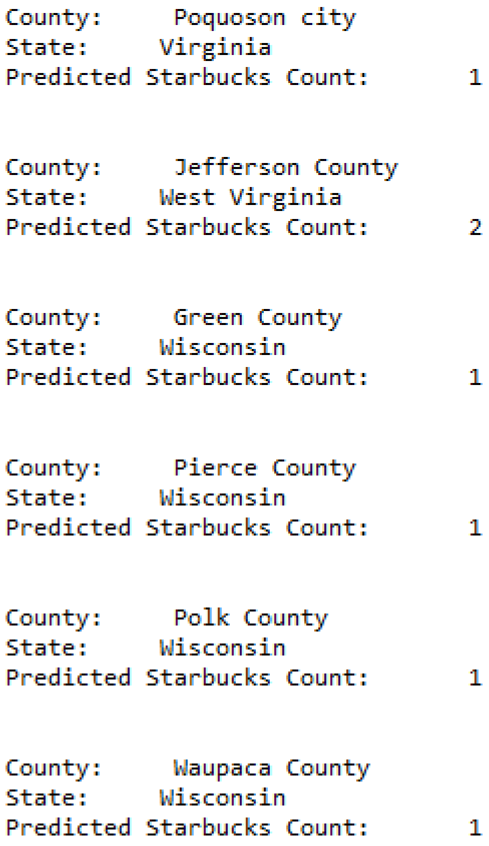
```
County:     Ashtabula County
State:     Ohio
Predicted Starbucks Count:       1


County:     Huron County
State:     Ohio
Predicted Starbucks Count:       1


County:     Jefferson County
State:     Ohio
Predicted Starbucks Count:       1


County:     Lawrence County
State:     Ohio
Predicted Starbucks Count:       1


County:     Grady County
State:     Oklahoma
Predicted Starbucks Count:       1


County:     Wagoner County
State:     Oklahoma
Predicted Starbucks Count:       1


County:     Armstrong County
State:     Pennsylvania
Predicted Starbucks Count:       1
```



```
County:     Blair County
State:     Pennsylvania
Predicted Starbucks Count:       3


County:     Bradford County
State:     Pennsylvania
Predicted Starbucks Count:       1


County:     Crawford County
State:     Pennsylvania
Predicted Starbucks Count:       1


County:     Perry County
State:     Pennsylvania
Predicted Starbucks Count:       1


County:     Pike County
State:     Pennsylvania
Predicted Starbucks Count:       1


County:     Venango County
State:     Pennsylvania
Predicted Starbucks Count:       1


County:     Laurens County
State:     South Carolina
Predicted Starbucks Count:       1
```



```
County:     Greene County
State:     Tennessee
Predicted Starbucks Count:       1


County:     Van Zandt County
State:     Texas
Predicted Starbucks Count:       1


County:     Addison County
State:     Vermont
Predicted Starbucks Count:       1


County:     Rutland County
State:     Vermont
Predicted Starbucks Count:       1


County:     Windsor County
State:     Vermont
Predicted Starbucks Count:       2


County:     Louisa County
State:     Virginia
Predicted Starbucks Count:       1


County:     Poquoson city
State:     Virginia
Predicted Starbucks Count:       1
```



```
County:     Poquoson city
State:     Virginia
Predicted Starbucks Count:      1


County:     Jefferson County
State:     West Virginia
Predicted Starbucks Count:      2


County:     Green County
State:     Wisconsin
Predicted Starbucks Count:      1


County:     Pierce County
State:     Wisconsin
Predicted Starbucks Count:      1


County:     Polk County
State:     Wisconsin
Predicted Starbucks Count:      1


County:     Waupaca County
State:     Wisconsin
Predicted Starbucks Count:      1
```