Processing Census Data

The census data was downloaded from https://factfinder.census.gov/faces/nav/jsf/pages/index.xhtml (https://factfinder.census.gov/faces/nav/jsf/pages/index.xhtml). The following data was downloaded per msa:

1. Age Group Demographics
2. Gender Demographics
3. Marital Status
4. Race Demographics
5. Education Levels
6. Income Statistics
7. Real Estate Vacancy Rates

Importing Libraries and Defining functions

In [467]:
```
import numpy as np
import pandas as pd
import matplotlib
import matplotlib.pyplot as plt
import sklearn.metrics as metrics
import seaborn as sns
import random
from sklearn.model_selection import cross_val_score
from sklearn import cross_validation
from sklearn import tree
from sklearn.tree import DecisionTreeClassifier
from sklearn.ensemble import RandomForestClassifier
from sklearn.ensemble import VotingClassifier
from sklearn.ensemble import AdaBoostClassifier
from sklearn.linear_model import LogisticRegressionCV
%matplotlib inline

import csv
from sklearn import ensemble
import math
from sklearn.metrics import confusion_matrix

from sklearn.base import BaseEstimator
from sklearn.base import ClassifierMixin

sns.set_context('notebook')
sns.set_style("darkgrid")

import requests
from bs4 import BeautifulSoup
from IPython.display import IFrame, HTML

import warnings
warnings.filterwarnings('ignore')
```

In [2]:
```
# Shorthand for easyy graphing
def get_axs(rows, columns, fig_size_width, fig_size_height):
    dims = (fig_size_width, fig_size_height)
    fig, axs = plt.subplots(rows, columns, figsize=dims)
    if(rows*columns>1):
         axs = axs.ravel()

# Converts numeric string into int
def get_int(s):
    return_value = np.nan
    if s!=None:
        try:
            return_value = int(s)
            return return_value
        except ValueError:
            return return_value
    else:
        return return_value

# Converts entire row into ints
def convert_to_int(row):
    return_value = []
    return_value = [get_int(i) for i in row]
    return return_value
```

Read in the downloaded datasets

In [618]:
```
age_gender_dict ={}
race_dict = {}
marital_dict = {}
vacancy_dict = {}
income_dict = {}
edu_dict = {}

for i in range(2006,2017):
    age_gender_dict[i] = pd.DataFrame(pd.read_csv('data/agegender_'+str(i)+'.c
sv', encoding='latin-1'))
    race_dict[i] = pd.DataFrame(pd.read_csv('data/race_'+str(i)+'.csv', encodi
ng='latin-1'))
    marital_dict[i] = pd.DataFrame(pd.read_csv('data/marital_'+str(i)+'.csv',
encoding='latin-1'))
    vacancy_dict[i] = pd.DataFrame(pd.read_csv('data/vacancy_'+str(i)+'.csv',
encoding='latin-1'))
    income_dict[i] = pd.DataFrame(pd.read_csv('data/income_'+str(i)+'.csv', en
coding='latin-1'))
    edu_dict[i] = pd.DataFrame(pd.read_csv('data/edu_'+str(i)+'.csv', encoding
='latin-1'))
```

Merging all years

In [620]:
```
##### MERGE DATA

merged_2 = {}
merged_3 = {}
merged_4 = {}
merged_5 = {}
merged_all = {}

for i in range(2006,2017):
    merged_2[i] = pd.merge(age_gender_dict[i], race_dict[i], left_on=('Id2'),
right_on=str(i)+'Id2', how='left')
    merged_3[i] = pd.merge(merged_2[i], marital_dict[i], left_on=(str(i)+'Id2'
), right_on=str(i)+'Id2', how='left')
    merged_4[i] = pd.merge(merged_3[i], vacancy_dict[i], left_on=(str(i)+'Id2'
), right_on=str(i)+'Id2', how='left')
    merged_5[i] = pd.merge(merged_4[i], income_dict[i], left_on=(str(i)+'Id2'
), right_on=str(i)+'Id2', how='left')
    merged_all[i] = pd.merge(merged_5[i], edu_dict[i], left_on=(str(i)+'Id2'),
 right_on=str(i)+'Id2', how='left')
```

In [622]:
```
merged_all[2006].head()
```

Out[622]:

	Id	Id2	Geography	total	male_total	male_under5	male_5to9	m
0	3100000US10180	10180	Abilene, TX Metro Area	158548	78912	5642	4648	6
1	3100000US10380	10380	Aguadilla-Isabela-San Sebastián, PR Metro Area	336502	166686	11601	11870	1
2	3100000US10420	10420	Akron, OH Metro Area	700943	337619	21106	22114	2
3	3100000US10500	10500	Albany, GA Metro Area	165062	78572	6720	6167	6
4	3100000US10580	10580	Albany-Schenectady-Troy, NY Metro Area	850957	413205	23804	24299	2

5 rows × 95 columns

Drop Redundant Columns

In [623]:
```
merged_all_clean = {}

for i in range(2006,2017):
    merged_all_clean[i] = merged_all[i].drop(['Unnamed: 0_x', str(i)+'Geograph
y_x', 'Unnamed: 0_y', str(i)+'Id_y', str(i)+'Id2', str(i)+'Geography_y', 'Unna
med: 0_x',str(i)+'Id_x', str(i)+'Geography_x', str(i)+'Geography_y', 'Unnamed:
 0_x', str(i)+'Unnamed: 0', str(i)+'Id_x', str(i)+'Geography_x', 'Unnamed: 0_
y', str(i)+'Id_y', str(i)+'Geography_y'], axis =1)
```

In [625]:
```
merged_all_clean[2016].head()
```

Out[625]:

	Id	Id2	Geography	total	male_total	male_under5	male_5to9	ma
0	310M300US10180	10180	Abilene, TX Metro Area	170860	87459	5571	6315	53
1	310M300US10380	10380	Aguadilla-Isabela, PR Metro Area	309764	153695	7976	9909	94
2	310M300US10420	10420	Akron, OH Metro Area	702221	341200	19415	19402	21
3	310M300US10500	10500	Albany, GA Metro Area	152506	72073	4555	4994	65
4	310M300US10540	10540	Albany, OR Metro Area	122849	61175	4247	4910	34

5 rows × 81 columns

In [626]:
```
years = range(2006, 2017)
for year in years:
    merged_all_clean[year].columns = merged_all_clean[year].columns.str.replac
e(str(year), '')
```

In [627]:
```
merged_all_clean[2016].head()
```

Out[627]:

	Id	Id2	Geography	total	male_total	male_under5	male_5to9	ma
0	310M300US10180	10180	Abilene, TX Metro Area	170860	87459	5571	6315	53
1	310M300US10380	10380	Aguadilla-Isabela, PR Metro Area	309764	153695	7976	9909	94
2	310M300US10420	10420	Akron, OH Metro Area	702221	341200	19415	19402	21
3	310M300US10500	10500	Albany, GA Metro Area	152506	72073	4555	4994	65
4	310M300US10540	10540	Albany, OR Metro Area	122849	61175	4247	4910	34

5 rows × 81 columns

Cleaning Data - Using percentages instead of actual values

In [628]:
```
for i in range(2006, 2017):
    print(i, end='\r')
    merged_all_clean[i]['age15to19'] = merged_all_clean[i]['male_15-17'] + mer
ged_all_clean[i]['male_18-19'] + merged_all_clean[i]['female_15-17'] + merged_
all_clean[i]['female_18-19']
    merged_all_clean[i]['age15to19'] = merged_all_clean[i]['age15to19']/merged
_all_clean[i]['total']

    merged_all_clean[i]['age20to24'] = merged_all_clean[i]['male_20'] + merged
_all_clean[i]['male_21'] +  merged_all_clean[i]['male_22to24'] + merged_all_cl
ean[i]['female_20'] + merged_all_clean[i]['female_21'] +  merged_all_clean[i][
'female_22to24']
    merged_all_clean[i]['age20to24'] = merged_all_clean[i]['age20to24']/merged
_all_clean[i]['total']

    merged_all_clean[i]['age25to29'] = merged_all_clean[i]['male_25to29'] + me
rged_all_clean[i]['female_25to29']
    merged_all_clean[i]['age25to29'] = merged_all_clean[i]['age25to29']/merged
_all_clean[i]['total']

    merged_all_clean[i]['age30to34'] = merged_all_clean[i]['male_30to34'] + me
rged_all_clean[i]['female_30to34']
    merged_all_clean[i]['age30to34'] = merged_all_clean[i]['age30to34']/merged
_all_clean[i]['total']

    merged_all_clean[i]['age35to44'] = merged_all_clean[i]['male_35to39'] + me
rged_all_clean[i]['male_40to44'] + merged_all_clean[i]['female_35to39'] + merg
ed_all_clean[i]['female_40to44']
    merged_all_clean[i]['age35to44'] = merged_all_clean[i]['age35to44']/merged
_all_clean[i]['total']


    merged_all_clean[i]['age45to59'] = merged_all_clean[i]['male_45to49'] + me
rged_all_clean[i]['male_50to54'] + merged_all_clean[i]['male_55to59'] + merged
_all_clean[i]['female_45to49'] + merged_all_clean[i]['female_50to54'] + merged
_all_clean[i]['female_55to59']
    merged_all_clean[i]['age45to59'] = merged_all_clean[i]['age45to59']/merged
_all_clean[i]['total']

    merged_all_clean[i]['age60plus'] = merged_all_clean[i]['male_60to61'] + me
rged_all_clean[i]['male_62to64'] + merged_all_clean[i]['male_65to66'] + merged
_all_clean[i]['male_67to69'] + merged_all_clean[i]['male_70to74'] + merged_all
_clean[i]['male_75to79'] + merged_all_clean[i]['male_80to84'] + merged_all_cle
an[i]['male_85plus'] + merged_all_clean[i]['female_60to61'] + merged_all_clean
[i]['female_62to64'] + merged_all_clean[i]['female_65to66'] + merged_all_clean
[i]['female_67to69'] + merged_all_clean[i]['female_70to74'] + merged_all_clean
[i]['female_75to79'] + merged_all_clean[i]['female_80to84'] + merged_all_clean
[i]['female_85plus']
    merged_all_clean[i]['age60plus'] = merged_all_clean[i]['age60plus']/merged
_all_clean[i]['total']
```

```
2016
```

In [631]:
```
for i in range(2006, 2017):
    print(i, end='\r')
    merged_all_clean[i]['total'] = convert_to_int(merged_all_clean[i]['total'
])
    merged_all_clean[i]['Now married (except separated)'] = convert_to_int(mer
ged_all_clean[i]['Now married (except separated)'])
    merged_all_clean[i]['Widowed'] = convert_to_int(merged_all_clean[i]['Widow
ed'])
    merged_all_clean[i]['Divorced'] = convert_to_int(merged_all_clean[i]['Divo
rced'])
    merged_all_clean[i]['Separated'] = convert_to_int(merged_all_clean[i]['Sep
arated'])
    merged_all_clean[i]['Never married'] = convert_to_int(merged_all_clean[i][
'Never married'])

    merged_all_clean[i]['Now married (except separated)'] = merged_all_clean[i
]['Now married (except separated)']/merged_all_clean[i]['total']
    merged_all_clean[i]['Widowed'] = merged_all_clean[i]['Widowed']/merged_all
_clean[i]['total']
    merged_all_clean[i]['Divorced'] = merged_all_clean[i]['Divorced']/merged_a
ll_clean[i]['total']
    merged_all_clean[i]['Separated'] = merged_all_clean[i]['Separated']/merged
_all_clean[i]['total']
    merged_all_clean[i]['Never married'] = merged_all_clean[i]['Never married'
]/merged_all_clean[i]['total']
```

```
2016
```

In [632]:
```
for i in range(2006, 2017):
    print(i, end='\r')
    merged_all_clean[i]['White'] = merged_all_clean[i]['White']/merged_all_cle
an[i]['total']
    merged_all_clean[i]['Black or African American'] = merged_all_clean[i]['Bl
ack or African American']/merged_all_clean[i]['total']
    merged_all_clean[i]['American Indian and Alaska Native'] = merged_all_clea
n[i]['American Indian and Alaska Native']/merged_all_clean[i]['total']
    merged_all_clean[i]['Asian'] = merged_all_clean[i]['Asian']/merged_all_cle
an[i]['total']
    merged_all_clean[i]['Native Hawaiian and Other Pacific Islander'] = merged
_all_clean[i]['Native Hawaiian and Other Pacific Islander']/merged_all_clean[i
]['total']
    merged_all_clean[i]['Other'] = merged_all_clean[i]['Other']/merged_all_cle
an[i]['total']
    merged_all_clean[i]['Two or More'] = merged_all_clean[i]['Two or More']/me
rged_all_clean[i]['total']
```

```
2016
```

In [633]:
```
for i in range(2006, 2017):
    print(i, end='\r')
    merged_all_clean[i]['Less than 9th grade'] = merged_all_clean[i]['Less tha
n 9th grade']/merged_all_clean[i]['Total25plus']
    merged_all_clean[i][' 9th to 12th grade, no diploma'] = merged_all_clean[i
][' 9th to 12th grade, no diploma']/merged_all_clean[i]['Total25plus']
    merged_all_clean[i]['High school graduate (includes equivalency)'] = merge
d_all_clean[i]['High school graduate (includes equivalency)']/merged_all_clean
[i]['Total25plus']
    merged_all_clean[i]['Some college, no degree'] = merged_all_clean[i]['Some
 college, no degree']/merged_all_clean[i]['Total25plus']
    merged_all_clean[i]["Associate's degree"] = merged_all_clean[i]["Associat
e's degree"]/merged_all_clean[i]['Total25plus']
    merged_all_clean[i]["Bachelor's degree"] = merged_all_clean[i]["Bachelor's
 degree"]/merged_all_clean[i]['Total25plus']
    merged_all_clean[i]['Graduate or professional degree'] = merged_all_clean[
i]['Graduate or professional degree']/merged_all_clean[i]['Total25plus']
```

```
2016
```

In [634]:
```
for i in range(2006, 2017):
    print(i, end='\r')
    merged_all_clean[i] = merged_all_clean[i].drop(['male_under5', 'Total', 'm
ale_5to9','female_5to9', 'male_10-14', 'male_15-17', 'male_18-19', 'male_20',
'male_21','male_22to24', 'male_25to29',  'male_30to34', 'male_35to39', 'male_4
0to44', 'male_45to49', 'male_50to54', 'male_55to59','male_60to61', 'male_62to6
4','male_65to66', 'male_67to69', 'male_70to74', 'male_75to79', 'male_80to84',
'male_85plus','female_under5', 'female_10-14', 'female_15-17', 'female_18-19',
 'female_20', 'female_21',  'female_22to24',  'female_25to29',  'female_30to3
4', 'female_35to39', 'female_40to44', 'female_45to49', 'female_50to54', 'femal
e_55to59', 'female_60to61', 'female_62to64', 'female_65to66', 'female_67to69',
 'female_70to74', 'female_75to79', 'female_80to84', 'female_85plus'], axis=1)
```

```
2016
```

In [635]:
```
for i in range(2006, 2017):
    print(i, end='\r')
    merged_all_clean[i]['Some College'] = merged_all_clean[i]['Some college, n
o degree'] + merged_all_clean[i]["Associate's degree"]
    merged_all_clean[i] = merged_all_clean[i].drop(['Some college, no degree',
 "Associate's degree", 'Total25plus'], axis = 1)
```

```
2016
```

In [636]:
```
for i in range(2006, 2017):
    print(i, end='\r')
    merged_all_clean[i]['Vacancy Rate'] = merged_all_clean[i]['Bldg_Vacant'] /
 merged_all_clean[i]['Bldg_Total']
    merged_all_clean[i]['MtoF'] = merged_all_clean[i]['male_total'] / merged_a
ll_clean[i]['female_total']
```

```
2016
```

In [637]:
```
for i in range(2006, 2017):
    print(i, end='\r')
    merged_all_clean[i] = merged_all_clean[i].drop(['Bldg_Vacant', 'Bldg_Occup
ied', 'Bldg_Total'], axis = 1)
    merged_all_clean[i] = merged_all_clean[i].drop(['male_total', 'female_tota
l'], axis = 1)
```

```
2016
```

In [638]:
```
merged_all_clean[2016].head()
```

Out[638]:

	Id	Id2	Geography	total	White	Black or African American	American Indian and Alaska Native	Asian
0	310M300US10180	10180	Abilene, TX Metro Area	170860	0.781371	0.082559	0.007433	0.020701
1	310M300US10380	10380	Aguadilla-Isabela, PR Metro Area	309764	0.704982	0.034152	0.000588	0.000000
2	310M300US10420	10420	Akron, OH Metro Area	702221	0.819179	0.120844	0.000964	0.028414
3	310M300US10500	10500	Albany, GA Metro Area	152506	0.421472	0.539284	0.001692	0.009882
4	310M300US10540	10540	Albany, OR Metro Area	122849	0.901383	0.004843	0.010159	0.010574

5 rows × 36 columns

In [639]:
```
for i in range (2006,2017):
    merged_all_clean[i] = merged_all_clean[i].drop(['Id', 'Geography', 'Unname
d: 0', 'Id', 'Geography'], axis= 1)
```

In [641]:
```
merged_all_clean[2011].shape
```

Out[641]:
```
(374, 31)
```

Renaming columns for easier readability

In [571]:
```
# 'Id2' - 'msa'
#  'total' - 'pop'
#  'White' - 'r1'
#  'Black or African American' - 'r2'
#  'American Indian and Alaska Native' - 'r3'
#  'Asian' - 'r4'
#  'Native Hawaiian and Other Pacific Islander' - 'r5'
#  'Other' - 'r6'
#  'Two or More' - 'r7'
#  'Now married (except separated)' - 'm1'
#  'Widowed' - 'm2'
#  'Divorced' - 'm3'
#  'Separated' - 'm4'
#  'Never married' - 'm5'
#  'IncomeHousehold' - 'i1'
#  'IncomeCapita' - 'i2'
#  'Less than 9th grade' - 'e1'
#  ' 9th to 12th grade no diploma' - 'e2'
#  'High school graduate (includes equivalency)' - 'e3'
#  "Bachelor's degree" - 'e4'
#  'Graduate or professional degree' - 'e5'
#  'age15to19' - 'a1'
#  'age20to24' - 'a2'
#  'age25to29' - 'a3'
#  'age30to34' - 'a4'
#  'age35to44' - 'a5'
#  'age45to59' - 'a6'
#  'age60plus' - 'a7'
#  'Some College' - 'e6'
#  'Vacancy Rate' - 'vr'
#  'MtoF' - 'mtof'
```

In [642]:
```
new_columns = ['msa', 'pop', 'r1', 'r2', 'r3', 'r4', 'r5', 'r6', 'r7', 'm1',
'm2', 'm3', 'm4', 'm5', 'i1', 'i2', 'e1', 'e2', 'e3', 'e4', 'e5', 'a1', 'a2',
'a3', 'a4', 'a5', 'a6', 'a7', 'e6', 'vr', 'mtof']
```

In [643]:
```
len(new_columns)
```

Out[643]: 31

In [644]:
```
for i in range(2006, 2017):
     merged_all_clean[i].columns = new_columns
```

In [645]:
```
for i in range(2006, 2017):
     merged_all_clean[i]['year'] = [i for j in range(0, len(merged_all_clean[i
].msa))]
```

Export Dataset

In [646]:
```
dfs = []
for i in range(2006, 2017):
    dfs.append(merged_all_clean[i])
df = pd.concat(dfs)
df.index = list(range(0, df.shape[0]))
df.to_csv("census_data_stacked.csv")
```

In [148]:
```
# MERGE EDUCATION ATTAINMENT

for i in range(2006, 2017):
    merged_result[str(i)+'Some College'] = merged_result[str(i)+'Some college,
 no degree'] + merged_result[str(i)+"Associate's degree"]

    merged_result = merged_result.drop([str(i)+'Some college, no degree', str(
i)+"Associate's degree"], axis = 1)
```

In [150]:
```
merged_result.to_csv('data/merged_census.csv')
```