The Effect of Pro Sports teams on a City's Economy

April 22, 2020

1 The Effect of Pro Sports teams on a City's Economy

DS 5559
Ben Rogers, br9wr
Jordan Bales, jcb4dt
Jae Hyun Lee, jl3fp

2 Questions:

1. Does a city having a sports team affect its GDP or unemployment?
2. Does market size matter to economic impact?
3. The year the team joins a league or when ownership changes?

3 Data Import and Preprocessing

[84]:
```
import os
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
from pandas.api.types import CategoricalDtype
from plotnine import *
from plotnine.data import mpg
%matplotlib inline
```

3.1 Webscrapper

The codes below are used to create one compiled version of csv file with Excel.

[15]:
```
import pandas as pd
url = 'https://www.pro-football-reference.com/years/2001/attendance.htm'
url2 = 'https://www.pro-football-reference.com/years/2002/attendance.htm'
url3 = 'https://www.pro-football-reference.com/years/2003/attendance.htm'
url4 = 'https://www.pro-football-reference.com/years/2004/attendance.htm'
```

```
url5 = 'https://www.pro-football-reference.com/years/2005/attendance.htm'
url6 = 'https://www.pro-football-reference.com/years/2006/attendance.htm'
url7 = 'https://www.pro-football-reference.com/years/2007/attendance.htm'
url8 = 'https://www.pro-football-reference.com/years/2008/attendance.htm'
url9 = 'https://www.pro-football-reference.com/years/2009/attendance.htm'
url10 = 'https://www.pro-football-reference.com/years/2010/attendance.htm'
url11 = 'https://www.pro-football-reference.com/years/2011/attendance.htm'
url12 = 'https://www.pro-football-reference.com/years/2012/attendance.htm'
url13 = 'https://www.pro-football-reference.com/years/2013/attendance.htm'
url14 = 'https://www.pro-football-reference.com/years/2014/attendance.htm'
url15 = 'https://www.pro-football-reference.com/years/2015/attendance.htm'
url16 = 'https://www.pro-football-reference.com/years/2016/attendance.htm'
url17 = 'https://www.pro-football-reference.com/years/2017/attendance.htm'
url18 = 'https://www.pro-football-reference.com/years/2018/attendance.htm'

df = pd.read_html(url)[0]
df2 = pd.read_html(url2)[0]
df3 = pd.read_html(url3)[0]
df4 = pd.read_html(url4)[0]
df5 = pd.read_html(url5)[0]
df6 = pd.read_html(url6)[0]
df7 = pd.read_html(url7)[0]
df8 = pd.read_html(url8)[0]
df9 = pd.read_html(url9)[0]
df10 = pd.read_html(url10)[0]
df11 = pd.read_html(url11)[0]
df12 = pd.read_html(url12)[0]
df13 = pd.read_html(url13)[0]
df14 = pd.read_html(url14)[0]
df15 = pd.read_html(url15)[0]
df16 = pd.read_html(url16)[0]
df17 = pd.read_html(url17)[0]
df18 = pd.read_html(url18)[0]

df.to_csv('nflattendance2001.csv')
df2.to_csv('nflattendance2002.csv')
df3.to_csv('nflattendance2003.csv')
df4.to_csv('nflattendance2004.csv')
df5.to_csv('nflattendance2005.csv')
df6.to_csv('nflattendance2006.csv')
df7.to_csv('nflattendance2007.csv')
df8.to_csv('nflattendance2008.csv')
df9.to_csv('nflattendance2009.csv')
df10.to_csv('nflattendance2010.csv')
df11.to_csv('nflattendance2011.csv')
df12.to_csv('nflattendance2012.csv')
df13.to_csv('nflattendance2013.csv')
```

```
df14.to_csv('nflattendance2014.csv')
df15.to_csv('nflattendance2015.csv')
df16.to_csv('nflattendance2016.csv')
df17.to_csv('nflattendance2017.csv')
df18.to_csv('nflattendance2018.csv')
```

[39]:
```
url1 = 'https://www.pro-football-reference.com/years/2001/#all_AFC'
url2 = 'https://www.pro-football-reference.com/years/2002/#all_AFC'
url3 = 'https://www.pro-football-reference.com/years/2003/#all_AFC'
url4 = 'https://www.pro-football-reference.com/years/2004/#all_AFC'
url5 = 'https://www.pro-football-reference.com/years/2005/#all_AFC'
url6 = 'https://www.pro-football-reference.com/years/2006/#all_AFC'
url7 = 'https://www.pro-football-reference.com/years/2007/#all_AFC'
url8 = 'https://www.pro-football-reference.com/years/2008/#all_AFC'
url9 = 'https://www.pro-football-reference.com/years/2009/#all_AFC'
url10 = 'https://www.pro-football-reference.com/years/2010/#all_AFC'
url11 = 'https://www.pro-football-reference.com/years/2011/#all_AFC'
url12 = 'https://www.pro-football-reference.com/years/2012/#all_AFC'
url13 = 'https://www.pro-football-reference.com/years/2013/#all_AFC'
url14 = 'https://www.pro-football-reference.com/years/2014/#all_AFC'
url15 = 'https://www.pro-football-reference.com/years/2015/#all_AFC'
url16 = 'https://www.pro-football-reference.com/years/2016/#all_AFC'
url17 = 'https://www.pro-football-reference.com/years/2017/#all_AFC'
url18 = 'https://www.pro-football-reference.com/years/2018/#all_AFC'

df1 = pd.read_html(url1)[0]
df2 = pd.read_html(url2)[0]
df3 = pd.read_html(url3)[0]
df4 = pd.read_html(url4)[0]
df5 = pd.read_html(url5)[0]
df6 = pd.read_html(url6)[0]
df7 = pd.read_html(url7)[0]
df8 = pd.read_html(url8)[0]
df9 = pd.read_html(url9)[0]
df10 = pd.read_html(url10)[0]
df11 = pd.read_html(url11)[0]
df12 = pd.read_html(url12)[0]
df13 = pd.read_html(url13)[0]
df14 = pd.read_html(url14)[0]
df15 = pd.read_html(url15)[0]
df16 = pd.read_html(url16)[0]
df17 = pd.read_html(url17)[0]
df18 = pd.read_html(url18)[0]

df1.to_csv('afcrecord2001.csv')
df2.to_csv('afcrecord2002.csv')
df3.to_csv('afcrecord2003.csv')
```

```
df4.to_csv('afcrecord2004.csv')
df5.to_csv('afcrecord2005.csv')
df6.to_csv('afcrecord2006.csv')
df7.to_csv('afcrecord2007.csv')
df8.to_csv('afcrecord2008.csv')
df9.to_csv('afcrecord2009.csv')
df10.to_csv('afcrecord2010.csv')
df11.to_csv('afcrecord2011.csv')
df12.to_csv('afcrecord2012.csv')
df13.to_csv('afcrecord2013.csv')
df14.to_csv('afcrecord2014.csv')
df15.to_csv('afcrecord2015.csv')
df16.to_csv('afcrecord2016.csv')
df17.to_csv('afcrecord2017.csv')
df18.to_csv('afcrecord2018.csv')
```

[2]:
```
url1 = 'https://www.baseball-reference.com/leagues/MLB/2001-misc.shtml'
url2 = 'https://www.baseball-reference.com/leagues/MLB/2002-misc.shtml'
url3 = 'https://www.baseball-reference.com/leagues/MLB/2003-misc.shtml'
url4 = 'https://www.baseball-reference.com/leagues/MLB/2004-misc.shtml'
url5 = 'https://www.baseball-reference.com/leagues/MLB/2005-misc.shtml'
url6 = 'https://www.baseball-reference.com/leagues/MLB/2006-misc.shtml'
url7 = 'https://www.baseball-reference.com/leagues/MLB/2007-misc.shtml'
url8 = 'https://www.baseball-reference.com/leagues/MLB/2008-misc.shtml'
url9 = 'https://www.baseball-reference.com/leagues/MLB/2009-misc.shtml'
url10 = 'https://www.baseball-reference.com/leagues/MLB/2010-misc.shtml'
url11 = 'https://www.baseball-reference.com/leagues/MLB/2011-misc.shtml'
url12 = 'https://www.baseball-reference.com/leagues/MLB/2012-misc.shtml'
url13 = 'https://www.baseball-reference.com/leagues/MLB/2013-misc.shtml'
url14 = 'https://www.baseball-reference.com/leagues/MLB/2014-misc.shtml'
url15 = 'https://www.baseball-reference.com/leagues/MLB/2015-misc.shtml'
url16 = 'https://www.baseball-reference.com/leagues/MLB/2016-misc.shtml'
url17 = 'https://www.baseball-reference.com/leagues/MLB/2017-misc.shtml'
url18 = 'https://www.baseball-reference.com/leagues/MLB/2018-misc.shtml'

df1 = pd.read_html(url1)[0]
df2 = pd.read_html(url2)[0]
df3 = pd.read_html(url3)[0]
df4 = pd.read_html(url4)[0]
df5 = pd.read_html(url5)[0]
df6 = pd.read_html(url6)[0]
df7 = pd.read_html(url7)[0]
df8 = pd.read_html(url8)[0]
df9 = pd.read_html(url9)[0]
df10 = pd.read_html(url10)[0]
df11 = pd.read_html(url11)[0]
df12 = pd.read_html(url12)[0]
```

```
df13 = pd.read_html(url13)[0]
df14 = pd.read_html(url14)[0]
df15 = pd.read_html(url15)[0]
df16 = pd.read_html(url16)[0]
df17 = pd.read_html(url17)[0]
df18 = pd.read_html(url18)[0]

df1.to_csv('mlbattendance2001.csv')
df2.to_csv('mlbattendance2002.csv')
df3.to_csv('mlbattendance2003.csv')
df4.to_csv('mlbattendance2004.csv')
df5.to_csv('mlbattendance2005.csv')
df6.to_csv('mlbattendance2006.csv')
df7.to_csv('mlbattendance2007.csv')
df8.to_csv('mlbattendance2008.csv')
df9.to_csv('mlbattendance2009.csv')
df10.to_csv('mlbattendance2010.csv')
df11.to_csv('mlbattendance2011.csv')
df12.to_csv('mlbattendance2012.csv')
df13.to_csv('mlbattendance2013.csv')
df14.to_csv('mlbattendance2014.csv')
df15.to_csv('mlbattendance2015.csv')
df16.to_csv('mlbattendance2016.csv')
df17.to_csv('mlbattendance2017.csv')
df18.to_csv('mlbattendance2018.csv')
```

[3]:
```
url1 = 'https://www.basketball-reference.com/leagues/NBA_2001_ratings.html'
url2 = 'https://www.basketball-reference.com/leagues/NBA_2002_ratings.html'
url3 = 'https://www.basketball-reference.com/leagues/NBA_2003_ratings.html'
url4 = 'https://www.basketball-reference.com/leagues/NBA_2004_ratings.html'
url5 = 'https://www.basketball-reference.com/leagues/NBA_2005_ratings.html'
url6 = 'https://www.basketball-reference.com/leagues/NBA_2006_ratings.html'
url7 = 'https://www.basketball-reference.com/leagues/NBA_2007_ratings.html'
url8 = 'https://www.basketball-reference.com/leagues/NBA_2008_ratings.html'
url9 = 'https://www.basketball-reference.com/leagues/NBA_2009_ratings.html'
url10 = 'https://www.basketball-reference.com/leagues/NBA_2010_ratings.html'
url11 = 'https://www.basketball-reference.com/leagues/NBA_2011_ratings.html'
url12 = 'https://www.basketball-reference.com/leagues/NBA_2012_ratings.html'
url13 = 'https://www.basketball-reference.com/leagues/NBA_2013_ratings.html'
url14 = 'https://www.basketball-reference.com/leagues/NBA_2014_ratings.html'
url15 = 'https://www.basketball-reference.com/leagues/NBA_2015_ratings.html'
url16 = 'https://www.basketball-reference.com/leagues/NBA_2016_ratings.html'
url17 = 'https://www.basketball-reference.com/leagues/NBA_2017_ratings.html'
url18 = 'https://www.basketball-reference.com/leagues/NBA_2018_ratings.html'

df1 = pd.read_html(url1)[0]
df2 = pd.read_html(url2)[0]
```

```
df3 = pd.read_html(url3)[0]
df4 = pd.read_html(url4)[0]
df5 = pd.read_html(url5)[0]
df6 = pd.read_html(url6)[0]
df7 = pd.read_html(url7)[0]
df8 = pd.read_html(url8)[0]
df9 = pd.read_html(url9)[0]
df10 = pd.read_html(url10)[0]
df11 = pd.read_html(url11)[0]
df12 = pd.read_html(url12)[0]
df13 = pd.read_html(url13)[0]
df14 = pd.read_html(url14)[0]
df15 = pd.read_html(url15)[0]
df16 = pd.read_html(url16)[0]
df17 = pd.read_html(url17)[0]
df18 = pd.read_html(url18)[0]

df1.to_csv('nbarecord01.csv')
df2.to_csv('nbarecord02.csv')
df3.to_csv('nbarecord03.csv')
df4.to_csv('nbarecord04.csv')
df5.to_csv('nbarecord05.csv')
df6.to_csv('nbarecord06.csv')
df7.to_csv('nbarecord07.csv')
df8.to_csv('nbarecord08.csv')
df9.to_csv('nbarecord09.csv')
df10.to_csv('nbarecord10.csv')
df11.to_csv('nbarecord11.csv')
df12.to_csv('nbarecord12.csv')
df13.to_csv('nbarecord13.csv')
df14.to_csv('nbarecord14.csv')
df15.to_csv('nbarecord15.csv')
df16.to_csv('nbarecord16.csv')
df17.to_csv('nbarecord17.csv')
df18.to_csv('nbarecord18.csv')
```

3.2 Import Data

[2]:
```
df = pd.read_csv("City_combined_v4.csv")
```

3.3 Quick Summary

[3]:
```
df.head()
```

[3]:
```
        city        gdp  private_industries agriculture  mining utilities  \
0    Atlanta  209741698           191749470      366262  331505   2100718
1     Austin   55307638            47708348       21993  205947    381802
2      Boise   16435618            14448912         (D)   13511     57379
3     Boston  236051306           213403308         (D)   87120   2666446
4  Charlotte   76027214            69766921         (D)  105937   1805777

  construction manufacturing durable_goods_manufacturing  \
0          (D)           (D)                         (D)
1      4324915       7829516                         (D)
2      1174578       3196453                         (D)
3      9560212      26949729                         (D)
4      3895799      14555666                         (D)

  nondurable_goods_manufacturing  … super_bowl_winner nfl_division_champion  \
0                            (D)  …                 0                     0
1                            (D)  …                 0                     0
2                            (D)  …                 0                     0
3                            (D)  …                 1                     1
4                            (D)  …                 0                     0

  nfl_playoff_teams nfl_win_percentage  nfl_wins nfl_home_attendance  \
0                 0             0.4375         7              425717
1                 0             0.0000         0                   0
2                 0             0.0000         0                   0
3                 1             0.6875        11              482336
4                 0             0.0625         1              579080

  nfl_attendance_per_game  year team_relocated team_purchased
0               53214.625  2001              0              0
1                   0.000  2001              0              0
2                   0.000  2001              0              0
3               60292.000  2001              0              0
4               72385.000  2001              0              0

[5 rows x 97 columns]
```

[4]:
```
df.describe()
```

[4]:
```
                gdp  private_industries  finance_total  \
count  4.320000e+02        4.320000e+02   4.320000e+02
mean   2.452478e+08        2.186487e+08   5.642425e+07
std    2.928506e+08        2.652310e+08   8.719089e+07
min    1.091764e+07        9.892020e+06   1.855698e+06
25%    7.513007e+07        6.255351e+07   1.369553e+07
50%    1.227686e+08        1.106198e+08   2.684385e+07
75%    3.188245e+08        2.803013e+08   6.003646e+07
```

```
max    1.772320e+09       1.611478e+09   5.751021e+08

       government_and_government_enterprises  personal_income_total  \
count                           4.320000e+02           4.320000e+02
mean                            2.659914e+07           1.998965e+08
std                             3.019768e+07           2.384918e+08
min                             1.025624e+06           8.653562e+06
25%                             8.118968e+06           6.317938e+07
50%                             1.432709e+07           1.036139e+08
75%                             3.140421e+07           2.529817e+08
max                             1.608420e+08           1.480233e+09

       net_earnings_by_place_of_residence  personal_current_transfer_receipts  \
count                        4.320000e+02                        4.320000e+02
mean                         1.348661e+08                        2.614331e+07
std                          1.566991e+08                        3.434414e+07
min                          6.047825e+06                        9.235270e+05
25%                          4.232933e+07                        8.789657e+06
50%                          6.841517e+07                        1.485404e+07
75%                          1.735379e+08                        2.797278e+07
max                          9.413716e+08                        2.100819e+08

       income_maintenance_benefits  unemployment_insurance_compensation  \
count                 4.320000e+02                         4.320000e+02
mean                  2.844495e+06                         8.148527e+05
std                   3.910961e+06                         1.333127e+06
min                   5.328500e+04                         2.191400e+04
25%                   8.716368e+05                         1.612758e+05
50%                   1.507994e+06                         3.537850e+05
75%                   2.749312e+06                         8.761160e+05
max                   2.113625e+07                         1.143793e+07

       retirement_and_other  …  super_bowl_winner  nfl_division_champion  \
count          4.320000e+02  …         432.000000             432.000000
mean           2.248396e+07  …           0.030093               0.175926
std            2.948231e+07  …           0.171040               0.381199
min            8.273510e+05  …           0.000000               0.000000
25%            7.559103e+06  …           0.000000               0.000000
50%            1.280419e+07  …           0.000000               0.000000
75%            2.447055e+07  …           0.000000               0.000000
max            1.865861e+08  …           1.000000               1.000000

       nfl_playoff_teams  nfl_win_percentage    nfl_wins  nfl_home_attendance  \
count         432.000000          432.000000  432.000000         4.320000e+02
mean            0.256944            0.317708    5.423611         3.790125e+05
std             0.447937            0.288044    5.115256         3.246718e+05
min             0.000000            0.000000    0.000000         0.000000e+00
```

```
25%              0.000000             0.000000     0.000000          0.000000e+00
50%              0.000000             0.312500     5.000000          5.200060e+05
75%              1.000000             0.562500     9.000000          5.655115e+05
max              2.000000             1.000000    25.000000          1.276668e+06

       nfl_attendance_per_game         year  team_relocated  team_purchased
count               432.000000   432.000000      432.000000      432.000000
mean              44067.875723  2009.500000        0.011574        0.055556
std               34502.934266     5.194143        0.107082        0.229327
min                   0.000000  2001.000000        0.000000        0.000000
25%                   0.000000  2005.000000        0.000000        0.000000
50%               64589.625000  2009.500000        0.000000        0.000000
75%               70640.937500  2014.000000        0.000000        0.000000
max              157161.375000  2018.000000        1.000000        1.000000

[8 rows x 64 columns]
```

[5]:

```python
numRow = len(df)
numCol = len(df.columns)
print("Total number of rows: {}".format(numRow))
print("Total number of columns: {}".format(numCol))
```

```
Total number of rows: 432
Total number of columns: 97
```

[6]:

```python
pd.set_option('display.max_rows', None)
df.dtypes
```

[6]:

```
city
object
gdp
int64
private_industries
int64
agriculture
object
mining
object
utilities
object
construction
object
manufacturing
object
durable_goods_manufacturing
object
nondurable_goods_manufacturing
```

```
object
wholesale_trade
object
retail_trade
object
transportation_and_warehousing
object
information
object
finance_total
int64
finance
object
real_estate
object
professional_and_business_services
object
professional_scientific_and_technical_services
object
management_of_companies_and_enterprises
object
administrative_and_support_and_waste_management_and_remediation services
object
educational_services_health_care_and_social_assistance
object
educational_services
object
health_care_and_social_assistance
object
arts_total
object
arts_entertainment_and_recreation
object
accommodation_and_food_services
object
other_services
object
government_and_government_enterprises
int64
natural_resources_and_mining
object
trade
object
transportation_and_utilities
object
manufacturing_and_information
object
```

```
private_goods_producing_industries
object
private_services_providing_industries
object
personal_income_total
int64
net_earnings_by_place_of_residence
int64
personal_current_transfer_receipts
int64
income_maintenance_benefits
int64
unemployment_insurance_compensation
int64
retirement_and_other
int64
dividends_interest_and_rent
int64
population
int64
per_capita_personal_income
int64
per_capita_net_earnings
int64
per_capita_personal_current_transfer_receipts
int64
per_capita_income_maintenance_benefits
int64
per_capita_unemployment_insurance_compensation
int64
per_capita_retirement_and_other
int64
per_capita_dividends_interest_and_rent
int64
earnings_by_place_of_work
int64
wages_and_salaries
int64
supplements_to_wages_and_salaries
int64
employer_contributions_for_employee_pension_and_insurance_funds
int64
employer_contributions_for_government_social_insurance
int64
proprietors_income
int64
farm_proprietors_income
```

```
object
nonfarm_proprietors_income
int64
total_employment
int64
wage_and_salary_employment
int64
proprietors_employment
int64
farm_proprietors_employment
int64
nonfarm_proprietors_employment
int64
average_earnings_per_job
int64
average_wages_and_salaries
int64
average_nonfarm_proprietors_income
int64
total_mlb_teams
int64
total_mlb_team_value
float64
total _team _revenue_mlb
object
home_games_mlb
int64
reg_season_wins_mlb
int64
home_wins_mlb
int64
world_series_title
int64
division _title_mlb
int64
attendance_mlb
int64
attendance_per_game_mlb
float64
payroll_mlb
int64
total_nfl_teams
int64
nfl_team_values
int64
nfl_team_revenue
int64
```

```
total_nba_team
int64
nba_team_value
float64
nba_team_revenue
float64
large_market
int64
medium_market
int64
small_market
int64
no_teams
int64
super_bowl_winner
int64
nfl_division_champion
int64
nfl_playoff_teams
int64
nfl_win_percentage
float64
nfl_wins
int64
nfl_home_attendance
int64
nfl_attendance_per_game
float64
year
int64
team_relocated
int64
team_purchased
int64
dtype: object
```

3.4 Preprocessing

[7]:
```python
# Replace all missing data with np.NaN
df = df.replace(to_replace='(D)', value=np.NaN)
df = df.replace(to_replace='(L)', value=np.NaN)
df = df.replace(to_replace='na', value=np.NaN)
```

[8]:
```python
# Find percentage of np.NaN per column
x = df.isnull().sum().sort_values(ascending=False) / len(df)
```

[9]:
```
# Drop any column with 25% or more missing data
df = df.loc[:, x < .25]
```

[10]:
```
# Expected to drop 20 columns
numColNew = len(df.columns)
print("{} column(s) have been dropped; {} columns remain.".format(numCol -␣
↪numColNew, numColNew))
```

```
20 column(s) have been dropped; 77 columns remain.
```

[11]:
```
# Check dtypes of columns
df.dtypes
```

[11]:
```
city                                                           object
gdp                                                             int64
private_industries                                              int64
mining                                                         object
construction                                                   object
manufacturing                                                  object
retail_trade                                                   object
finance_total                                                   int64
finance                                                        object
real_estate                                                    object
professional_and_business_services                             object
educational_services_health_care_and_social_assistance         object
arts_total                                                     object
other_services                                                 object
government_and_government_enterprises                           int64
personal_income_total                                           int64
net_earnings_by_place_of_residence                              int64
personal_current_transfer_receipts                              int64
income_maintenance_benefits                                     int64
unemployment_insurance_compensation                             int64
retirement_and_other                                            int64
dividends_interest_and_rent                                     int64
population                                                      int64
per_capita_personal_income                                      int64
per_capita_net_earnings                                         int64
per_capita_personal_current_transfer_receipts                   int64
per_capita_income_maintenance_benefits                          int64
per_capita_unemployment_insurance_compensation                  int64
per_capita_retirement_and_other                                 int64
per_capita_dividends_interest_and_rent                          int64
earnings_by_place_of_work                                       int64
wages_and_salaries                                              int64
supplements_to_wages_and_salaries                               int64
employer_contributions_for_employee_pension_and_insurance_funds       int64
```

```
employer_contributions_for_government_social_insurance              int64
proprietors_income                                                  int64
farm_proprietors_income                                            object
nonfarm_proprietors_income                                          int64
total_employment                                                    int64
wage_and_salary_employment                                          int64
proprietors_employment                                              int64
farm_proprietors_employment                                         int64
nonfarm_proprietors_employment                                      int64
average_earnings_per_job                                            int64
average_wages_and_salaries                                          int64
average_nonfarm_proprietors_income                                  int64
total_mlb_teams                                                     int64
total_mlb_team_value                                              float64
total _team _revenue_mlb                                           object
home_games_mlb                                                      int64
reg_season_wins_mlb                                                 int64
home_wins_mlb                                                       int64
world_series_title                                                  int64
division _title_mlb                                                 int64
attendance_mlb                                                      int64
attendance_per_game_mlb                                           float64
payroll_mlb                                                         int64
total_nfl_teams                                                     int64
nfl_team_values                                                     int64
nfl_team_revenue                                                    int64
total_nba_team                                                      int64
nba_team_value                                                    float64
nba_team_revenue                                                  float64
large_market                                                        int64
medium_market                                                       int64
small_market                                                        int64
no_teams                                                            int64
super_bowl_winner                                                   int64
nfl_division_champion                                               int64
nfl_playoff_teams                                                   int64
nfl_win_percentage                                                float64
nfl_wins                                                            int64
nfl_home_attendance                                                 int64
nfl_attendance_per_game                                           float64
year                                                                int64
team_relocated                                                      int64
team_purchased                                                      int64
dtype: object
```

[12]:

```
# Replace all np.NaN with 0
df = df.replace(to_replace=np.NaN, value=int(0))
```

[13]:
```
# Change dtype of all columns except the first (city) to numeric. In order to␣
↪avoid muddling regression results
df.iloc[:,1:] = df.iloc[:,1:].apply(pd.to_numeric)
df.dtypes
```

[13]:
```
city                                                         object
gdp                                                           int64
private_industries                                            int64
mining                                                        int64
construction                                                  int64
manufacturing                                                 int64
retail_trade                                                  int64
finance_total                                                 int64
finance                                                       int64
real_estate                                                   int64
professional_and_business_services                            int64
educational_services_health_care_and_social_assistance        int64
arts_total                                                    int64
other_services                                                int64
government_and_government_enterprises                         int64
personal_income_total                                         int64
net_earnings_by_place_of_residence                            int64
personal_current_transfer_receipts                            int64
income_maintenance_benefits                                   int64
unemployment_insurance_compensation                           int64
retirement_and_other                                          int64
dividends_interest_and_rent                                   int64
population                                                    int64
per_capita_personal_income                                    int64
per_capita_net_earnings                                       int64
per_capita_personal_current_transfer_receipts                 int64
per_capita_income_maintenance_benefits                        int64
per_capita_unemployment_insurance_compensation                int64
per_capita_retirement_and_other                               int64
per_capita_dividends_interest_and_rent                        int64
earnings_by_place_of_work                                     int64
wages_and_salaries                                            int64
supplements_to_wages_and_salaries                             int64
employer_contributions_for_employee_pension_and_insurance_funds    int64
employer_contributions_for_government_social_insurance        int64
proprietors_income                                            int64
farm_proprietors_income                                       int64
nonfarm_proprietors_income                                    int64
total_employment                                              int64
wage_and_salary_employment                                    int64
proprietors_employment                                        int64
farm_proprietors_employment                                   int64
```

```
nonfarm_proprietors_employment                                          int64
average_earnings_per_job                                                int64
average_wages_and_salaries                                              int64
average_nonfarm_proprietors_income                                      int64
total_mlb_teams                                                         int64
total_mlb_team_value                                                  float64
total _team _revenue_mlb                                                int64
home_games_mlb                                                          int64
reg_season_wins_mlb                                                     int64
home_wins_mlb                                                           int64
world_series_title                                                      int64
division _title_mlb                                                     int64
attendance_mlb                                                          int64
attendance_per_game_mlb                                               float64
payroll_mlb                                                             int64
total_nfl_teams                                                         int64
nfl_team_values                                                         int64
nfl_team_revenue                                                        int64
total_nba_team                                                          int64
nba_team_value                                                        float64
nba_team_revenue                                                      float64
large_market                                                            int64
medium_market                                                           int64
small_market                                                            int64
no_teams                                                                int64
super_bowl_winner                                                       int64
nfl_division_champion                                                   int64
nfl_playoff_teams                                                       int64
nfl_win_percentage                                                    float64
nfl_wins                                                                int64
nfl_home_attendance                                                     int64
nfl_attendance_per_game                                               float64
year                                                                    int64
team_relocated                                                          int64
team_purchased                                                          int64
dtype: object
```

[14]:
```
# Correct column names
df = df.rename(columns={'total _team _revenue_mlb': 'total_team_revenue_mlb',␣
↪'division _title_mlb': 'division_title_mlb'})
```

[15]:
```
# Convert categorical columns to type category to make our model functions␣
↪easier to deal with
df[["large_market",␣
↪"medium_market","small_market","year","city","no_teams","super_bowl_winner",␣
↪\
```

```
␣
↪"nfl_division_champion","nfl_playoff_teams","world_series_title","division_title_mlb"]]␣
↪\
= df[["large_market",␣
↪"medium_market","small_market","year","city","no_teams","super_bowl_winner",␣
↪\
␣
↪"nfl_division_champion","nfl_playoff_teams","world_series_title","division_title_mlb"]].
↪astype('category')

df.dtypes
```

[15]:

```
city                                                            category
gdp                                                                int64
private_industries                                                 int64
mining                                                             int64
construction                                                       int64
manufacturing                                                      int64
retail_trade                                                       int64
finance_total                                                      int64
finance                                                            int64
real_estate                                                        int64
professional_and_business_services                                 int64
educational_services_health_care_and_social_assistance             int64
arts_total                                                         int64
other_services                                                     int64
government_and_government_enterprises                              int64
personal_income_total                                              int64
net_earnings_by_place_of_residence                                 int64
personal_current_transfer_receipts                                 int64
income_maintenance_benefits                                        int64
unemployment_insurance_compensation                                int64
retirement_and_other                                               int64
dividends_interest_and_rent                                        int64
population                                                         int64
per_capita_personal_income                                         int64
per_capita_net_earnings                                            int64
per_capita_personal_current_transfer_receipts                      int64
per_capita_income_maintenance_benefits                             int64
per_capita_unemployment_insurance_compensation                     int64
per_capita_retirement_and_other                                    int64
per_capita_dividends_interest_and_rent                             int64
earnings_by_place_of_work                                          int64
wages_and_salaries                                                 int64
supplements_to_wages_and_salaries                                  int64
employer_contributions_for_employee_pension_and_insurance_funds    int64
employer_contributions_for_government_social_insurance             int64
```

```
proprietors_income                                    int64
farm_proprietors_income                               int64
nonfarm_proprietors_income                            int64
total_employment                                      int64
wage_and_salary_employment                            int64
proprietors_employment                                int64
farm_proprietors_employment                           int64
nonfarm_proprietors_employment                        int64
average_earnings_per_job                              int64
average_wages_and_salaries                            int64
average_nonfarm_proprietors_income                    int64
total_mlb_teams                                       int64
total_mlb_team_value                                float64
total_team_revenue_mlb                                int64
home_games_mlb                                        int64
reg_season_wins_mlb                                   int64
home_wins_mlb                                         int64
world_series_title                                 category
division_title_mlb                                 category
attendance_mlb                                        int64
attendance_per_game_mlb                             float64
payroll_mlb                                           int64
total_nfl_teams                                       int64
nfl_team_values                                       int64
nfl_team_revenue                                      int64
total_nba_team                                        int64
nba_team_value                                      float64
nba_team_revenue                                    float64
large_market                                       category
medium_market                                      category
small_market                                       category
no_teams                                           category
super_bowl_winner                                  category
nfl_division_champion                              category
nfl_playoff_teams                                  category
nfl_win_percentage                                  float64
nfl_wins                                              int64
nfl_home_attendance                                   int64
nfl_attendance_per_game                             float64
year                                               category
team_relocated                                        int64
team_purchased                                        int64
dtype: object
```

END PREPROCESSING

4 Exploratory Data Analysis

We don't have a sudo privilege to install 'plotnine' package on DS 5559 Module.
The codes below were executed locally and resulting image (graphs) will be attached separately.

The image of scatter matrix is also attached separately to reduce the file size of this notebook.

4.1 Graphs (ggplot)

[ ]:
```
(ggplot(df)
 + aes(x='year', y='nba_team_value', color='city')
 + geom_jitter()
 + theme(axis_text_x = element_text(angle = 45,vjust = 1, hjust = 1))
 + labs(title='NBA Team Values over the years', x='Years', y='NBA Team Value in␣
 ↪Thousands of Dollars')
```

[ ]:
```
(ggplot(df)
 + aes(x='year', y='nba_team_revenue', color='city')
 + geom_jitter()
 + theme(axis_text_x = element_text(angle = 45,vjust = 1, hjust = 1))
 + labs(title='NBA Team Revenues over the years', x='Years', y='NBA Team␣
 ↪Revenue in Thousands of Dollars')
)
```

[ ]:
```
(ggplot(df)
 + aes(x='year', y='nfl_team_values', color='city')
 + geom_jitter()
 + theme(axis_text_x = element_text(angle = 45,vjust = 1, hjust = 1))
 + labs(title='NFL Team Values over the years', x='Years', y='NFL Team Value in␣
 ↪Thousands of Dollars')
)
```

[ ]:
```
(ggplot(df)
 + aes(x='year', y='nfl_team_revenue', color='city')
 + geom_jitter()
 + theme(axis_text_x = element_text(angle = 45,vjust = 1, hjust = 1))
 + labs(title='NFL Team Revenue over the years', x='Years', y='NFL Team Revenue␣
 ↪in Thousands of Dollars')
)
```

[ ]:
```
(ggplot(df)
 + aes(x='year', y='total_mlb_team_value', color='city')
 + geom_jitter()
 + theme(axis_text_x = element_text(angle = 45,vjust = 1, hjust = 1))
 + labs(title='MLB Team Values over the years', x='Years', y='MLB Team Value in␣
 ↪Thousands of Dollars')
```

```
)
```

[ ]:
```
(ggplot(df)
 + aes(x='year', y='total_team _revenue_mlb', color='city')
 + geom_jitter()
 + theme(axis_text_x = element_text(angle = 45,vjust = 1, hjust = 1))
 + labs(title='MLB Team Revenues over the years', x='Years', y='MLB Team␣
↪Revenue in Thousands of Dollars')
)
```

[ ]:
```
(ggplot(df)
 + aes(x='year', y='gdp', color='city')
 + geom_jitter()
 + theme(axis_text_x = element_text(angle = 45,vjust = 1, hjust = 1))
 + labs(title='City GDP over the years', x='Years', y='GDP in Thousands of␣
↪Dollars')
)
```

[ ]:
```
(ggplot(df)
 + aes(x='year', y='private_industries', color='city')
 + geom_jitter()
 + theme(axis_text_x = element_text(angle = 45,vjust = 1, hjust = 1))
 + labs(title='Private Industry over the years', x='Years', y='Private Industry␣
↪in Thousands of Dollars')
)
```

[ ]:
```
(ggplot(df)
 + aes(x='year', y='total_employment', color='city')
 + geom_jitter()
 + theme(axis_text_x = element_text(angle = 45,vjust = 1, hjust = 1))
 + labs(title='Employment over the years', x='Years', y='Employment')
)
```

4.2 Spark

[16]:
```
from pyspark.sql import SparkSession
from pyspark.mllib.regression import LabeledPoint, LinearRegressionWithSGD
from pyspark.ml.regression import LinearRegression

spark = SparkSession.builder \
        .master("local") \
        .appName("mllib_classifier") \
        .getOrCreate()
sc = spark.sparkContext
```

[17]:
```
# Going to export our formatted file into csv so we can use a fresh copy for␣
↪each question and algorithm iteration
df.to_csv('FormattedData.csv',index=False)
```

[18]:
```
# Create spark dataframe
from pyspark import SparkConf, SparkContext
from pyspark.sql import SQLContext

sqlContext = SQLContext(sc)

SparkDF = sqlContext.read.format('com.databricks.spark.csv').
↪options(header='true', inferschema='true').load('FormattedData.csv')
SparkDF.take(1)
```

[18]:
```
[Row(city='Atlanta', gdp=209741698, private_industries=191749470, mining=331505,
construction=0, manufacturing=0, retail_trade=13217659, finance_total=41922449,
finance=0, real_estate=0, professional_and_business_services=26620247,
educational_services_health_care_and_social_assistance=10466645, arts_total=0,
other_services=4145753, government_and_government_enterprises=17992228,
personal_income_total=153691997, net_earnings_by_place_of_residence=117319662,
personal_current_transfer_receipts=12145861,
income_maintenance_benefits=1204893, unemployment_insurance_compensation=316462,
retirement_and_other=10624506, dividends_interest_and_rent=24226474,
population=4402455, per_capita_personal_income=34911,
per_capita_net_earnings=26649,
per_capita_personal_current_transfer_receipts=2759,
per_capita_income_maintenance_benefits=274,
per_capita_unemployment_insurance_compensation=72,
per_capita_retirement_and_other=2413,
per_capita_dividends_interest_and_rent=5503,
earnings_by_place_of_work=132211508, wages_and_salaries=97270037,
supplements_to_wages_and_salaries=17607619,
employer_contributions_for_employee_pension_and_insurance_funds=11274319,
employer_contributions_for_government_social_insurance=6333300,
proprietors_income=17333852, farm_proprietors_income=188311,
nonfarm_proprietors_income=17145541, total_employment=2809261,
wage_and_salary_employment=2381096, proprietors_employment=428165,
farm_proprietors_employment=9902, nonfarm_proprietors_employment=418263,
average_earnings_per_job=47063, average_wages_and_salaries=40851,
average_nonfarm_proprietors_income=40992, total_mlb_teams=1,
total_mlb_team_value=407000.0, total_team_revenue_mlb=145500, home_games_mlb=81,
reg_season_wins_mlb=88, home_wins_mlb=40, world_series_title=0,
division_title_mlb=1, attendance_mlb=2823530,
attendance_per_game_mlb=34858.39506, payroll_mlb=91936166, total_nfl_teams=1,
nfl_team_values=338000, nfl_team_revenue=113000, total_nba_team=1,
nba_team_value=199000.0, nba_team_revenue=76000.0, large_market=1,
medium_market=0, small_market=0, no_teams=0, super_bowl_winner=0,
```

```
nfl_division_champion=0, nfl_playoff_teams=0, nfl_win_percentage=0.4375,
nfl_wins=7, nfl_home_attendance=425717, nfl_attendance_per_game=53214.625,
year=2001, team_relocated=0, team_purchased=0)]
```

[19]:
```
# Check schema
SparkDF.cache()
SparkDF.printSchema()
```

```
root
 |-- city: string (nullable = true)
 |-- gdp: integer (nullable = true)
 |-- private_industries: integer (nullable = true)
 |-- mining: integer (nullable = true)
 |-- construction: integer (nullable = true)
 |-- manufacturing: integer (nullable = true)
 |-- retail_trade: integer (nullable = true)
 |-- finance_total: integer (nullable = true)
 |-- finance: integer (nullable = true)
 |-- real_estate: integer (nullable = true)
 |-- professional_and_business_services: integer (nullable = true)
 |-- educational_services_health_care_and_social_assistance: integer (nullable =
true)
 |-- arts_total: integer (nullable = true)
 |-- other_services: integer (nullable = true)
 |-- government_and_government_enterprises: integer (nullable = true)
 |-- personal_income_total: integer (nullable = true)
 |-- net_earnings_by_place_of_residence: integer (nullable = true)
 |-- personal_current_transfer_receipts: integer (nullable = true)
 |-- income_maintenance_benefits: integer (nullable = true)
 |-- unemployment_insurance_compensation: integer (nullable = true)
 |-- retirement_and_other: integer (nullable = true)
 |-- dividends_interest_and_rent: integer (nullable = true)
 |-- population: integer (nullable = true)
 |-- per_capita_personal_income: integer (nullable = true)
 |-- per_capita_net_earnings: integer (nullable = true)
 |-- per_capita_personal_current_transfer_receipts: integer (nullable = true)
 |-- per_capita_income_maintenance_benefits: integer (nullable = true)
 |-- per_capita_unemployment_insurance_compensation: integer (nullable = true)
 |-- per_capita_retirement_and_other: integer (nullable = true)
 |-- per_capita_dividends_interest_and_rent: integer (nullable = true)
 |-- earnings_by_place_of_work: integer (nullable = true)
 |-- wages_and_salaries: integer (nullable = true)
 |-- supplements_to_wages_and_salaries: integer (nullable = true)
 |-- employer_contributions_for_employee_pension_and_insurance_funds: integer
(nullable = true)
 |-- employer_contributions_for_government_social_insurance: integer (nullable =
true)
```

```
 |-- proprietors_income: integer (nullable = true)
 |-- farm_proprietors_income: integer (nullable = true)
 |-- nonfarm_proprietors_income: integer (nullable = true)
 |-- total_employment: integer (nullable = true)
 |-- wage_and_salary_employment: integer (nullable = true)
 |-- proprietors_employment: integer (nullable = true)
 |-- farm_proprietors_employment: integer (nullable = true)
 |-- nonfarm_proprietors_employment: integer (nullable = true)
 |-- average_earnings_per_job: integer (nullable = true)
 |-- average_wages_and_salaries: integer (nullable = true)
 |-- average_nonfarm_proprietors_income: integer (nullable = true)
 |-- total_mlb_teams: integer (nullable = true)
 |-- total_mlb_team_value: double (nullable = true)
 |-- total_team_revenue_mlb: integer (nullable = true)
 |-- home_games_mlb: integer (nullable = true)
 |-- reg_season_wins_mlb: integer (nullable = true)
 |-- home_wins_mlb: integer (nullable = true)
 |-- world_series_title: integer (nullable = true)
 |-- division_title_mlb: integer (nullable = true)
 |-- attendance_mlb: integer (nullable = true)
 |-- attendance_per_game_mlb: double (nullable = true)
 |-- payroll_mlb: integer (nullable = true)
 |-- total_nfl_teams: integer (nullable = true)
 |-- nfl_team_values: integer (nullable = true)
 |-- nfl_team_revenue: integer (nullable = true)
 |-- total_nba_team: integer (nullable = true)
 |-- nba_team_value: double (nullable = true)
 |-- nba_team_revenue: double (nullable = true)
 |-- large_market: integer (nullable = true)
 |-- medium_market: integer (nullable = true)
 |-- small_market: integer (nullable = true)
 |-- no_teams: integer (nullable = true)
 |-- super_bowl_winner: integer (nullable = true)
 |-- nfl_division_champion: integer (nullable = true)
 |-- nfl_playoff_teams: integer (nullable = true)
 |-- nfl_win_percentage: double (nullable = true)
 |-- nfl_wins: integer (nullable = true)
 |-- nfl_home_attendance: integer (nullable = true)
 |-- nfl_attendance_per_game: double (nullable = true)
 |-- year: integer (nullable = true)
 |-- team_relocated: integer (nullable = true)
 |-- team_purchased: integer (nullable = true)
```

[20]:

```
SparkDF.describe().toPandas().transpose()
```

[20]:

```
                                                         0  \
summary                                              count
city                                                   432
gdp                                                    432
private_industries                                     432
mining                                                 432
construction                                           432
manufacturing                                          432
retail_trade                                           432
finance_total                                          432
finance                                                432
real_estate                                            432
professional_and_business_services                     432
educational_services_health_care_and_social_ass…     432
arts_total                                             432
other_services                                         432
government_and_government_enterprises                  432
personal_income_total                                  432
net_earnings_by_place_of_residence                     432
personal_current_transfer_receipts                     432
income_maintenance_benefits                            432
unemployment_insurance_compensation                    432
retirement_and_other                                   432
dividends_interest_and_rent                            432
population                                             432
per_capita_personal_income                             432
per_capita_net_earnings                                432
per_capita_personal_current_transfer_receipts          432
per_capita_income_maintenance_benefits                 432
per_capita_unemployment_insurance_compensation         432
per_capita_retirement_and_other                        432
per_capita_dividends_interest_and_rent                 432
earnings_by_place_of_work                              432
wages_and_salaries                                     432
supplements_to_wages_and_salaries                      432
employer_contributions_for_employee_pension_and…     432
employer_contributions_for_government_social_in…     432
proprietors_income                                     432
farm_proprietors_income                                432
nonfarm_proprietors_income                             432
total_employment                                       432
wage_and_salary_employment                             432
proprietors_employment                                 432
farm_proprietors_employment                            432
nonfarm_proprietors_employment                         432
average_earnings_per_job                               432
average_wages_and_salaries                             432
```

```
average_nonfarm_proprietors_income                     432
total_mlb_teams                                        432
total_mlb_team_value                                   432
total_team_revenue_mlb                                 432
home_games_mlb                                         432
reg_season_wins_mlb                                    432
home_wins_mlb                                          432
world_series_title                                     432
division_title_mlb                                     432
attendance_mlb                                         432
attendance_per_game_mlb                                432
payroll_mlb                                            432
total_nfl_teams                                        432
nfl_team_values                                        432
nfl_team_revenue                                       432
total_nba_team                                         432
nba_team_value                                         432
nba_team_revenue                                       432
large_market                                           432
medium_market                                          432
small_market                                           432
no_teams                                               432
super_bowl_winner                                      432
nfl_division_champion                                  432
nfl_playoff_teams                                      432
nfl_win_percentage                                     432
nfl_wins                                               432
nfl_home_attendance                                    432
nfl_attendance_per_game                                432
year                                                   432
team_relocated                                         432
team_purchased                                         432

                                                                     1  \
summary                                                           mean
city                                                              None
gdp                                                2.4524781537962964E8
private_industries                                 2.1864867616203704E8
mining                                                     1267060.625
construction                                         7558340.398148148
manufacturing                                      1.9947657002314813E7
retail_trade                                       1.2508954724537037E7
finance_total                                      5.6424249083333336E7
finance                                             2.090223631712963E7
real_estate                                         2.942935178935185E7
professional_and_business_services                 2.6076587840277776E7
educational_services_health_care_and_social_ass…  1.7408494321759257E7
```

arts_total	9100840.372685185
other_services	4707902.766203703
government_and_government_enterprises	2.659913920601852E7
personal_income_total	1.998965341597222E8
net_earnings_by_place_of_residence	1.3486612967592594E8
personal_current_transfer_receipts	2.6143312696759257E7
income_maintenance_benefits	2844495.0833333335
unemployment_insurance_compensation	814852.7175925926
retirement_and_other	2.2483964895833332E7
dividends_interest_and_rent	3.888709178703704E7
population	4021506.997685185
per_capita_personal_income	45473.270833333336
per_capita_net_earnings	30633.01851851852
per_capita_personal_current_transfer_receipts	6135.189814814815
per_capita_income_maintenance_benefits	625.6180555555555
per_capita_unemployment_insurance_compensation	185.51388888888889
per_capita_retirement_and_other	5323.99537037037
per_capita_dividends_interest_and_rent	8705.104166666666
earnings_by_place_of_work	1.5443057660416666E8
wages_and_salaries	1.105815576712963E8
supplements_to_wages_and_salaries	2.4100582578703705E7
employer_contributions_for_employee_pension_and…	1.6313874368055556E7
employer_contributions_for_government_social_in…	7786708.210648148
proprietors_income	1.9748436354166668E7
farm_proprietors_income	60900.36342592593
nonfarm_proprietors_income	1.9687535840277776E7
total_employment	2499206.5810185187
wage_and_salary_employment	1981788.9837962964
proprietors_employment	517417.59722222225
farm_proprietors_employment	4974.395833333333
nonfarm_proprietors_employment	512443.2013888889
average_earnings_per_job	55523.6712962963
average_wages_and_salaries	49996.62037037037
average_nonfarm_proprietors_income	33689.45138888889
total_mlb_teams	0.7013888888888888
total_mlb_team_value	549779.6653196759
total_team_revenue_mlb	149599.7685185185
home_games_mlb	56.085648148148145
reg_season_wins_mlb	57.613425925925924
home_wins_mlb	31.01388888888889
world_series_title	0.027777777777777776
division_title_mlb	0.16435185185185186
attendance_mlb	1975673.9560185184
attendance_per_game_mlb	19982.61024363426
payroll_mlb	7.157765349074075E7
total_nfl_teams	0.6712962962962963
nfl_team_values	895157.4074074074

```
nfl_team_revenue                                          176266.2037037037
total_nba_team                                           0.6319444444444444
nba_team_value                                                     463875.0
nba_team_revenue                                          91664.51851851853
large_market                                                           0.25
medium_market                                                          0.25
small_market                                                           0.25
no_teams                                                               0.25
super_bowl_winner                                       0.03009259259259259
nfl_division_champion                                   0.17592592592592593
nfl_playoff_teams                                        0.2569444444444444
nfl_win_percentage                                       0.3177083333333333
nfl_wins                                                  5.423611111111111
nfl_home_attendance                                      379012.50694444444
nfl_attendance_per_game                                   44067.87572337963
year                                                                 2009.5
team_relocated                                         0.011574074074074073
team_purchased                                          0.05555555555555555

                                                                          2  \
summary                                                              stddev
city                                                                   None
gdp                                                   2.9285056394180304E8
private_industries                                     2.652309847794597E8
mining                                                    4488817.401528948
construction                                              9308089.068500169
manufacturing                                         2.3264202645681888E7
retail_trade                                          1.4182950630311672E7
finance_total                                          8.719088511636335E7
finance                                                4.454914393066328E7
real_estate                                            4.689620853421105E7
professional_and_business_services                     3.942271354493554E7
educational_services_health_care_and_social_ass…   2.451222136616712E7
arts_total                                             1.206079294849609E7
other_services                                             6286300.79037286
government_and_government_enterprises                 3.0197679797612283E7
personal_income_total                                 2.3849175413430935E8
net_earnings_by_place_of_residence                     1.566990613733322E8
personal_current_transfer_receipts                     3.434413936949845E7
income_maintenance_benefits                              3910960.5169997723
unemployment_insurance_compensation                      1333126.5614502167
retirement_and_other                                  2.9482307442539826E7
dividends_interest_and_rent                            4.875490035154773E7
population                                                4218882.190463905
per_capita_personal_income                               11012.662021984966
per_capita_net_earnings                                   7616.010534027159
per_capita_personal_current_transfer_receipts            1861.9341653980507
```

per_capita_income_maintenance_benefits	213.2753738772361
per_capita_unemployment_insurance_compensation	146.13004662255034
per_capita_retirement_and_other	1668.2439327040986
per_capita_dividends_interest_and_rent	2910.2854789814924
earnings_by_place_of_work	1.8092549093509924E8
wages_and_salaries	1.282162655984165E8
supplements_to_wages_and_salaries	2.78412987072954E7
employer_contributions_for_employee_pension_and…	1.905221683503727E7
employer_contributions_for_government_social_in…	8817769.20073897
proprietors_income	2.5844273353188615E7
farm_proprietors_income	88646.28777151584
nonfarm_proprietors_income	2.5834960523421932E7
total_employment	2541126.576045151
wage_and_salary_employment	1980726.1266527828
proprietors_employment	574019.0276180834
farm_proprietors_employment	3802.4900686658984
nonfarm_proprietors_employment	573867.462160621
average_earnings_per_job	11232.147004995046
average_wages_and_salaries	11062.72271879733
average_nonfarm_proprietors_income	11966.706218665588
total_mlb_teams	0.67860777135784
total_mlb_team_value	901924.1737218464
total_team_revenue_mlb	197306.9783080198
home_games_mlb	54.37985823337057
reg_season_wins_mlb	57.60225331797121
home_wins_mlb	30.858704889221485
world_series_title	0.16452608356500964
division_title_mlb	0.4068183907583431
attendance_mlb	3621216.75030037
attendance_per_game_mlb	39936.1124559457
payroll_mlb	8.912562962797254E7
total_nfl_teams	0.560336479277811
nfl_team_values	1044801.6467090223
nfl_team_revenue	173127.84050170673
total_nba_team	0.6326007282332002
nba_team_value	866664.5896890457
nba_team_revenue	115614.3595369789
large_market	0.4335147457731791
medium_market	0.4335147457731792
small_market	0.43351474577317944
no_teams	0.43351474577317906
super_bowl_winner	0.17104019348454436
nfl_division_champion	0.38119859093391084
nfl_playoff_teams	0.4479371814039837
nfl_win_percentage	0.28804390948864644
nfl_wins	5.115256096644763
nfl_home_attendance	324671.7622615185

nfl_attendance_per_game	34502.93426581142
year	5.194142694368531
team_relocated	0.10708248257015061
team_purchased	0.22932700219682278

	3	4
summary	min	max
city	Atlanta	Virginia Beach
gdp	10917643	1772319824
private_industries	9892020	1611477854
mining	0	33066973
construction	0	55829143
manufacturing	0	93939097
retail_trade	0	74526288
finance_total	1855698	575102095
finance	0	312484712
real_estate	0	262617382
professional_and_business_services	0	262520531
educational_services_health_care_and_social_ass…	0	156446504
arts_total	0	74418929
other_services	0	33801689
government_and_government_enterprises	1025624	160841970
personal_income_total	8653562	1480232981
net_earnings_by_place_of_residence	6047825	941371633
personal_current_transfer_receipts	923527	210081879
income_maintenance_benefits	53285	21136252
unemployment_insurance_compensation	21914	11437933
retirement_and_other	827351	186586136
dividends_interest_and_rent	1617407	329359178
population	285783	19345820
per_capita_personal_income	26546	99424
per_capita_net_earnings	16209	66294
per_capita_personal_current_transfer_receipts	2396	11065
per_capita_income_maintenance_benefits	186	1126
per_capita_unemployment_insurance_compensation	22	775
per_capita_retirement_and_other	2084	9838
per_capita_dividends_interest_and_rent	4335	24904
earnings_by_place_of_work	7263306	1088320101
wages_and_salaries	5398964	764408075
supplements_to_wages_and_salaries	1346532	158451649
employer_contributions_for_employee_pension_and…	938239	107207667
employer_contributions_for_government_social_in…	408293	51243982
proprietors_income	517810	165460377
farm_proprietors_income	-109700	596857
nonfarm_proprietors_income	235626	165401953
total_employment	195881	12917191
wage_and_salary_employment	168607	9736816

```
proprietors_employment                               26735          3180375
farm_proprietors_employment                            164            18617
nonfarm_proprietors_employment                       23433          3175504
average_earnings_per_job                             35672           100347
average_wages_and_salaries                           31014            97212
average_nonfarm_proprietors_income                    1512            90786
total_mlb_teams                                          0                2
total_mlb_team_value                                   0.0        6100000.0
total_team_revenue_mlb                                   0          1675000
home_games_mlb                                           0              163
reg_season_wins_mlb                                      0              194
home_wins_mlb                                            0              109
world_series_title                                       0                1
division_title_mlb                                       0                2
attendance_mlb                                           0         64571233
attendance_per_game_mlb                                0.0      768705.1548
payroll_mlb                                              0        396662929
total_nfl_teams                                          0                2
nfl_team_values                                          0          6150000
nfl_team_revenue                                         0           936000
total_nba_team                                           0                2
nba_team_value                                         0.0        5900000.0
nba_team_revenue                                       0.0         699000.0
large_market                                             0                1
medium_market                                            0                1
small_market                                             0                1
no_teams                                                 0                1
super_bowl_winner                                        0                1
nfl_division_champion                                    0                1
nfl_playoff_teams                                        0                2
nfl_win_percentage                                     0.0              1.0
nfl_wins                                                 0               25
nfl_home_attendance                                      0          1276668
nfl_attendance_per_game                                0.0       157161.375
year                                                  2001             2018
team_relocated                                           0                1
team_purchased                                           0                1
```

4.3 Scatter Matrix

[ ]:
```
# Scatter Matrix
numeric_features = [t[0] for t in SparkDF.dtypes if t[1] == 'int' or t[1] ==␣
↪'float']
sampled_data = SparkDF.select(numeric_features).sample(False, 0.8).toPandas()
axs = pd.plotting.scatter_matrix(sampled_data, figsize=(100, 100))
n = len(sampled_data.columns)
```

```
for i in range(n):
    v = axs[i, 0]
    v.yaxis.label.set_rotation(0)
    v.yaxis.label.set_ha('right')
    v.set_yticks(())
    h = axs[n-1, i]
    h.xaxis.label.set_rotation(90)
    h.set_xticks(())
```

4.4 Variable Correlation

[21]:
```
# Explore variable correlation
import six
for i in SparkDF.columns:
    if not( isinstance(SparkDF.select(i).take(1)[0][0], six.string_types)):
        print( "GDP for ", i, SparkDF.stat.corr('gdp',i))
```

```
GDP for  gdp 1.0
GDP for  private_industries 0.9990404733420539
GDP for  mining 0.1150729440967743
GDP for  construction 0.7514387855785662
GDP for  manufacturing 0.46107496971927375
GDP for  retail_trade 0.9561629708376779
GDP for  finance_total 0.9762518593797699
GDP for  finance 0.8947362589491431
GDP for  real_estate 0.9471521873365634
GDP for  professional_and_business_services 0.7529513307902886
GDP for  educational_services_health_care_and_social_assistance
0.9377869678321251
GDP for  arts_total 0.8496316631288121
GDP for  other_services 0.8945359056247975
GDP for  government_and_government_enterprises 0.923053544270096
GDP for  personal_income_total 0.9989813064962608
GDP for  net_earnings_by_place_of_residence 0.9984149213329097
GDP for  personal_current_transfer_receipts 0.9774870639814784
GDP for  income_maintenance_benefits 0.965363388732667
GDP for  unemployment_insurance_compensation 0.7315864750439102
GDP for  retirement_and_other 0.9775407353600813
GDP for  dividends_interest_and_rent 0.9891758734953132
GDP for  population 0.9682150952291451
GDP for  per_capita_personal_income 0.5169426852139307
GDP for  per_capita_net_earnings 0.501291222954431
GDP for  per_capita_personal_current_transfer_receipts 0.2761941748396393
GDP for  per_capita_income_maintenance_benefits 0.4031884361561063
GDP for  per_capita_unemployment_insurance_compensation 0.09914192977915737
GDP for  per_capita_retirement_and_other 0.24804122053057684
```

```
GDP for  per_capita_dividends_interest_and_rent 0.46759916300949883
GDP for  earnings_by_place_of_work 0.9987207477726697
GDP for  wages_and_salaries 0.9973454383887348
GDP for  supplements_to_wages_and_salaries 0.9958413595275908
GDP for  employer_contributions_for_employee_pension_and_insurance_funds
0.9942645603165186
GDP for  employer_contributions_for_government_social_insurance
0.9960084644092829
GDP for  proprietors_income 0.970915950215245
GDP for  farm_proprietors_income 0.15064231622810131
GDP for  nonfarm_proprietors_income 0.970749051467064
GDP for  total_employment 0.9810784627683331
GDP for  wage_and_salary_employment 0.9730976968656444
GDP for  proprietors_employment 0.9853410701483265
GDP for  farm_proprietors_employment 0.027761868259293207
GDP for  nonfarm_proprietors_employment 0.9854173585678606
GDP for  average_earnings_per_job 0.6249155133387088
GDP for  average_wages_and_salaries 0.619572814063881
GDP for  average_nonfarm_proprietors_income 0.3713524169510003
GDP for  total_mlb_teams 0.7639096357850249
GDP for  total_mlb_team_value 0.8533391220395156
GDP for  total_team_revenue_mlb 0.8289881121048082
GDP for  home_games_mlb 0.7496767633158015
GDP for  reg_season_wins_mlb 0.7718437739400922
GDP for  home_wins_mlb 0.7604786704470279
GDP for  world_series_title 0.13890047672749486
GDP for  division_title_mlb 0.3940155978813875
GDP for  attendance_mlb 0.43730479108883746
GDP for  attendance_per_game_mlb 0.25144820461079037
GDP for  payroll_mlb 0.8765608598725453
GDP for  total_nfl_teams 0.5595205385803989
GDP for  nfl_team_values 0.6320326310782421
GDP for  nfl_team_revenue 0.6125531693064749
GDP for  total_nba_team 0.7580095628371489
GDP for  nba_team_value 0.716561077109635
GDP for  nba_team_revenue 0.8354385903196401
GDP for  large_market 0.7115784223602855
GDP for  medium_market -0.02591899952478598
GDP for  small_market -0.3193502309249145
GDP for  no_teams -0.36630919191058464
GDP for  super_bowl_winner 0.10084380810456242
GDP for  nfl_division_champion 0.12348318295717965
GDP for  nfl_playoff_teams 0.214317586261741
GDP for  nfl_win_percentage 0.25749161191372316
GDP for  nfl_wins 0.446287769826114
GDP for  nfl_home_attendance 0.5870259087858878
GDP for  nfl_attendance_per_game 0.35472208778761805
GDP for  year 0.16435755389826923
```

```
GDP for  team_relocated 0.09245128029228405
GDP for  team_purchased 0.10189908935712476
```

5 Data Spliting/Sampling

[58]:
```
# Select columns to keep
col_names = (list(df.columns)[-31:])

#col_names.extend(['total_employment', 'per_capita_personal_income'])
col_names.extend(['per_capita_personal_income'])  # Removed for second run␣
 ↪through of regression.

remove_cols = ['attendance_mlb',␣
 ↪'nfl_home_attendance','attendance_per_game_mlb','home_games_mlb','total_nba_team','total_ml
              ␣
 ↪'home_wins_mlb','world_series_title','division_title_mlb','super_bowl_winner','nfl_division
               'nfl_attendance_per_game','payroll_mlb']
col_names = [col for col in col_names if col not in remove_cols]

# Different set of columns we used in training/evaluating models:
#remove_cols = ['attendance_mlb', 'nfl_home_attendance',␣
 ↪'nfl_wins','total_mlb_teams','home_wins_mlb','reg_season_wins_mlb']
#remove_cols = ['attendance_mlb',␣
 ↪'nfl_home_attendance','attendance_per_game_mlb','home_games_mlb','total_nba_team','total_ml
              ␣
 ↪#'home_wins_mlb','world_series_title','division_title_mlb','super_bowl_winner','nfl_divisio
               #'nfl_attendance_per_game','payroll_mlb']
#remove_cols = ['attendance_mlb', 'nfl_home_attendance',␣
 ↪'nfl_wins','total_mlb_teams','home_wins_mlb','reg_season_wins_mlb','nfl_win_percentage','nf
               #'home_games_mlb']
#remove_cols = ['attendance_mlb', 'nfl_home_attendance',␣
 ↪'nfl_wins','total_mlb_teams','home_wins_mlb','reg_season_wins_mlb','nfl_win_percentage','nf
               #'home_games_mlb', 'large_market','medium_market','small_market',]
```

[59]:
```
# Transform into vector features
from pyspark.ml.feature import VectorAssembler
vectorAssembler = VectorAssembler(inputCols = col_names, outputCol = 'features')
vSparkDF = vectorAssembler.transform(SparkDF)
vSparkDF = vSparkDF.select(['features', 'gdp'])
vSparkDF.show(3)
```

```
+--------------------+---------+
|            features|      gdp|
+--------------------+---------+
|[407000.0,145500…|209741698|
```

```
|(14,[9,10,13],[1…| 55307638|
|(14,[9,10,13],[1…| 16435618|
+--------------------+---------+
only showing top 3 rows
```

[60]:
```
# Split data into train and test set
splits = vSparkDF.randomSplit([0.7, 0.3])
train_df = splits[0]
test_df = splits[1]
```

6 Model Construction and Evaluation

1. Linear Regression
2. Decision Tree Regression
3. Gradient-boosted Tree Regression

We used the same logic repeatedly to train and analyze different models (features) (see code block 58).

6.1 Linear Regression

[61]:
```
# Linear Regression
from pyspark.ml.regression import LinearRegression
lr = LinearRegression(featuresCol = 'features', labelCol='gdp', maxIter=10,␣
 ↪regParam=0.3, elasticNetParam=0.8)
lr_model = lr.fit(train_df)
print("Coefficients: " + str(lr_model.coefficients))
print("Intercept: " + str(lr_model.intercept))
```

```
Coefficients: [97.97768857870597,268.24153858837775,13.180338120907132,153.78697
575257442,-0.0,683.73012986315,138133408.13347384,-57408676.30342582,-94051875.8
7989247,-433397.6031829992,-7616017.523846555,-122000519.75638849,25315267.21949
936,-504.09989097273296]
Intercept: 15381737250.28748
```

6.2 Linear Regression (Train) Evaluation

[63]:
```
trainingSummary = lr_model.summary
print("RMSE: %f" % trainingSummary.rootMeanSquaredError)
print("r2: %f" % trainingSummary.r2)
print("Adjusted r2: %f" % trainingSummary.r2adj)
```

```
RMSE: 124594860.634852
r2: 0.845292
Adjusted r2: 0.837875
```

[64]:
```
train_df.describe().show()
```

```
+-------+--------------------+
|summary|                 gdp|
+-------+--------------------+
|  count|                 307|
|   mean|2.6218122320846906E8|
| stddev| 3.172870083581394E8|
|    min|            10917643|
|    max|          1772319824|
+-------+--------------------+
```

[65]:
```
# analyze coefficients of features
c = lr_model.coefficients
d = {'features': col_names, 'coefficients': c}
features = pd.DataFrame(data=d)
features
```

[65]:
```
                      features  coefficients
0          total_mlb_team_value  9.797769e+01
1        total_team_revenue_mlb  2.682415e+02
2               nfl_team_values  1.318034e+01
3              nfl_team_revenue  1.537870e+02
4                nba_team_value -0.000000e+00
5              nba_team_revenue  6.837301e+02
6                  large_market  1.381334e+08
7                 medium_market -5.740868e+07
8                  small_market -9.405188e+07
9                      no_teams -4.333976e+05
10                         year -7.616018e+06
11               team_relocated -1.220005e+08
12               team_purchased  2.531527e+07
13   per_capita_personal_income -5.040999e+02
```

6.3 Linear Regression (Test) Evaluation

[66]:
```
lr_predictions = lr_model.transform(test_df)
lr_predictions.select("prediction","gdp","features").show(5)
from pyspark.ml.evaluation import RegressionEvaluator
lr_evaluator = RegressionEvaluator(predictionCol="prediction", \
                 labelCol="gdp",metricName="r2")
```

```
print("R Squared (R2) on test data = %g" % lr_evaluator.
 ↪evaluate(lr_predictions))
```

```
+--------------------+---------+--------------------+
|          prediction|      gdp|            features|
+--------------------+---------+--------------------+
|1.2359962999925423E8|112796544|(14,[0,1,2,3,7,10…|
| 1.261586847247982E8|114389181|(14,[0,1,2,3,7,10…|
|1.0880535902512741E8|120334797|(14,[0,1,2,3,7,10…|
|1.4053713604518318E8|233226865|(14,[0,1,2,3,7,10…|
|1.1918326277298927E8|129677300|(14,[0,1,2,3,7,10…|
+--------------------+---------+--------------------+
only showing top 5 rows

R Squared (R2) on test data = 0.801138
```

[67]:

```
test_result = lr_model.evaluate(test_df)
lr_rmse = test_result.rootMeanSquaredError
print("Linear Regression: Root Mean Squared Error (RMSE) on test data = %g" %␣
 ↪lr_rmse)
```

```
Linear Regression: Root Mean Squared Error (RMSE) on test data = 9.65013e+07
```

[68]:

```
print("numIterations: %d" % trainingSummary.totalIterations)
print("objectiveHistory: %s" % str(trainingSummary.objectiveHistory))
trainingSummary.residuals.show()
```

```
numIterations: 11
objectiveHistory: [0.5, 0.4000603758181479, 0.1888294343316928,
0.12766569691331303, 0.1067952248516049, 0.09681861964038191,
0.09023375397947249, 0.08534957240760684, 0.08189970188292792,
0.07793815609225353, 0.07735397530143193]
+--------------------+
|           residuals|
+--------------------+
| 2.087052418919735E8|
|-5.23719960187282…|
| 9.792833400663185E7|
|-3.71654276974449…|
| -5.63367761282177E7|
|-1.929798369767952E7|
|-1.00577646830959…|
|  -6717820.027656555|
|-1.81838107777652…|
|    445348.274225235|
|1.0434324251304626E8|
|1.1862082525510788E8|
|   7505935.088083267|
```

```
|1.2956148664688683E8|
| 1.429576796466694E8|
|   4249138.746261597|
|1.5914974679213715E8|
|  -4574678.638895035|
|1.2552667636040306E8|
| 2.217305166693306E7|
+--------------------+
only showing top 20 rows
```

[69]:
```python
predictions = lr_model.transform(test_df)
predictions.select("prediction","gdp","features").show()
```

```
+--------------------+---------+--------------------+
|          prediction|      gdp|            features|
+--------------------+---------+--------------------+
|1.2359962999925423E8|112796544|(14,[0,1,2,3,7,10…|
| 1.261586847247982E8|114389181|(14,[0,1,2,3,7,10…|
|1.0880535902512741E8|120334797|(14,[0,1,2,3,7,10…|
|1.4053713604518318E8|233226865|(14,[0,1,2,3,7,10…|
|1.1918326277298927E8|129677300|(14,[0,1,2,3,7,10…|
|1.3587489805522537E8|138118159|(14,[0,1,2,3,7,10…|
| 1.582710464039402E8|150163428|(14,[0,1,2,3,7,10…|
|2.3559198463868332E8|360940192|(14,[0,1,2,3,7,10…|
|2.3442978188765144E8|392036945|(14,[0,1,2,3,7,10…|
|   5.4751426684062E8|665296297|(14,[0,1,4,5,6,10…|
| 6.332436414239159E8|739857938|(14,[0,1,4,5,6,10…|
| 6.542940890929108E8|756470973|(14,[0,1,4,5,6,10…|
| 8.110207117820358E8|820353615|(14,[0,1,4,5,6,10…|
|1.0252974920246525E9|912384865|(14,[0,1,4,5,6,10…|
| 1.867524293989296E8|213096390|(14,[0,1,4,5,7,10…|
| 1.328072569291668E8| 90850236|(14,[0,1,4,5,8,10…|
|1.0335512075218582E8| 70579101|(14,[0,1,4,5,8,10…|
|1.0651210590215683E8| 77618804|(14,[0,1,4,5,8,10…|
| 3.559821363691788E8|240353006|(14,[0,2,3,4,5,6,…|
| 1.069674357549324E8| 84628076|(14,[0,2,3,7,10,1…|
+--------------------+---------+--------------------+
only showing top 20 rows
```

6.4 Decision Tree Regression

[ ]:
```python
# Decision Tree Regression
from pyspark.ml.regression import DecisionTreeRegressor
dt = DecisionTreeRegressor(featuresCol ='features', labelCol = 'gdp')
```

```
dt_model = dt.fit(train_df)
dt_predictions = dt_model.transform(test_df)
```

6.5 Decision Tree Regression Evaluation

[70]:
```
dt_evaluator = RegressionEvaluator(
    labelCol="gdp", predictionCol="prediction", metricName="rmse")
dt_rmse = dt_evaluator.evaluate(dt_predictions)
print("Decision Tree: Root Mean Squared Error (RMSE) on test data = %g" %␣
 ↪dt_rmse)
```

```
Decision Tree: Root Mean Squared Error (RMSE) on test data = 6.90014e+07
```

[71]:
```
# Feature Importance
a = dt_model.featureImportances.toArray()
d = {'features': col_names, 'importance': a}
fi = pd.DataFrame(data=d)
fi.sort_values(by='importance', ascending=False)
```

[71]:

	features	importance
1	total_team_revenue_mlb	0.645348
0	total_mlb_team_value	0.133827
3	nfl_team_revenue	0.047209
5	nba_team_revenue	0.040221
6	large_market	0.037229
2	nfl_team_values	0.024212
10	year	0.021677
4	nba_team_value	0.019270
7	medium_market	0.013007
13	per_capita_personal_income	0.012476
11	team_relocated	0.002765
12	team_purchased	0.001814
8	small_market	0.000946
9	no_teams	0.000000

6.6 Gradient-boosted Tree Regression

[72]:
```
# Gradient-boosted Tree Regression
from pyspark.ml.regression import GBTRegressor
gbt = GBTRegressor(featuresCol = 'features', labelCol = 'gdp', maxIter=10)
gbt_model = gbt.fit(train_df)
gbt_predictions = gbt_model.transform(test_df)
gbt_predictions.select('prediction', 'gdp', 'features').show(5)
```

```
+--------------------+---------+--------------------+
|          prediction|      gdp|            features|
+--------------------+---------+--------------------+
|1.0580206106817895E8|112796544|(14,[0,1,2,3,7,10…|
| 9.958060905157915E7|114389181|(14,[0,1,2,3,7,10…|
| 9.283220574138588E7|120334797|(14,[0,1,2,3,7,10…|
|2.4881729941875306E8|233226865|(14,[0,1,2,3,7,10…|
| 1.750199646928488E8|129677300|(14,[0,1,2,3,7,10…|
+--------------------+---------+--------------------+
only showing top 5 rows
```

6.7 Gradient-boosted Tree Regression Evaluation

[73]:
```
gbt_evaluator = RegressionEvaluator(
    labelCol="gdp", predictionCol="prediction", metricName="rmse")
gbt_rmse = gbt_evaluator.evaluate(gbt_predictions)
print("GBT: Root Mean Squared Error (RMSE) on test data = %g" % gbt_rmse)
```

```
GBT: Root Mean Squared Error (RMSE) on test data = 5.87912e+07
```

7 Function to Compare Models

For a quick comparison of models in terms of R-squared and RMSE.

[74]:
```
def compareModels(col_names):
    from pyspark.ml.feature import VectorAssembler
    vectorAssembler = VectorAssembler(inputCols = col_names, outputCol =␣
 ↪'features')
    vSparkDF = vectorAssembler.transform(SparkDF)
    vSparkDF = vSparkDF.select(['features', 'gdp'])
    # vSparkDF.show(3)

    splits = vSparkDF.randomSplit([0.7, 0.3])
    train_df = splits[0]
    test_df = splits[1]

    # Linear Regression
    from pyspark.ml.regression import LinearRegression
    lr = LinearRegression(featuresCol = 'features', labelCol='gdp', maxIter=10,␣
 ↪regParam=0.3, elasticNetParam=0.8)
    lr_model = lr.fit(train_df)
    # print("Coefficients: " + str(lr_model.coefficients))
    # print("Intercept: " + str(lr_model.intercept))
```

```
    # trainingSummary = lr_model.summary
    # print("RMSE: %f" % trainingSummary.rootMeanSquaredError)
    # print("r2: %f" % trainingSummary.r2)

    lr_predictions = lr_model.transform(test_df)
    # lr_predictions.select("prediction","gdp","features").show(5)
    from pyspark.ml.evaluation import RegressionEvaluator
    lr_evaluator = RegressionEvaluator(predictionCol="prediction", \
                     labelCol="gdp",metricName="r2")
    print("Linear Regression: R Squared (R2) on test data = %g" % lr_evaluator.
↪evaluate(lr_predictions))

    test_result = lr_model.evaluate(test_df)
    lr_rmse = test_result.rootMeanSquaredError
    print("Linear Regression: Root Mean Squared Error (RMSE) on test data = %g"␣
↪% lr_rmse)

    from pyspark.ml.regression import DecisionTreeRegressor
    dt = DecisionTreeRegressor(featuresCol ='features', labelCol = 'gdp')
    dt_model = dt.fit(train_df)
    dt_predictions = dt_model.transform(test_df)
    dt_evaluator = RegressionEvaluator(
        labelCol="gdp", predictionCol="prediction", metricName="rmse")
    dt_rmse = dt_evaluator.evaluate(dt_predictions)

    dt_evaluator_r2 = RegressionEvaluator(
        labelCol="gdp", predictionCol="prediction", metricName="r2")
    dt_r2 = dt_evaluator_r2.evaluate(dt_predictions)
    print("Decision Tree: R Squared (R2) on test data = %g" % dt_r2)
    print("Decision Tree: Root Mean Squared Error (RMSE) on test data = %g" %␣
↪dt_rmse)

    from pyspark.ml.regression import GBTRegressor
    gbt = GBTRegressor(featuresCol = 'features', labelCol = 'gdp', maxIter=10)
    gbt_model = gbt.fit(train_df)
    gbt_predictions = gbt_model.transform(test_df)
    # gbt_predictions.select('prediction', 'gdp', 'features').show(5)

    gbt_evaluator = RegressionEvaluator(
        labelCol="gdp", predictionCol="prediction", metricName="rmse")
    gbt_rmse = gbt_evaluator.evaluate(gbt_predictions)

    gbt_evaluator_r2 = RegressionEvaluator(
        labelCol="gdp", predictionCol="prediction", metricName="r2")
    gbt_r2 = gbt_evaluator_r2.evaluate(gbt_predictions)
    print("GBT: R Squared (R2) on test data = %g" % gbt_r2)
    print("GBT: Root Mean Squared Error (RMSE) on test data = %g" % gbt_rmse)
```

[75]:
```python
def featuresCoeff(col_names):
    from pyspark.ml.feature import VectorAssembler
    vectorAssembler = VectorAssembler(inputCols = col_names, outputCol =␣
↪'features')
    vSparkDF = vectorAssembler.transform(SparkDF)
    vSparkDF = vSparkDF.select(['features', 'gdp'])
    # vSparkDF.show(3)

    splits = vSparkDF.randomSplit([0.7, 0.3])
    train_df = splits[0]
    test_df = splits[1]

    # Linear Regression
    from pyspark.ml.regression import LinearRegression
    lr = LinearRegression(featuresCol = 'features', labelCol='gdp', maxIter=10,␣
↪regParam=0.3, elasticNetParam=0.8)
    lr_model = lr.fit(train_df)
    # print("Coefficients: " + str(lr_model.coefficients))
    # print("Intercept: " + str(lr_model.intercept))

    # trainingSummary = lr_model.summary
    # print("RMSE: %f" % trainingSummary.rootMeanSquaredError)
    # print("r2: %f" % trainingSummary.r2)

    c = lr_model.coefficients
    d = {'features': col_names, 'coefficients': c}
    features = pd.DataFrame(data=d)

    return features
```

8 Uses

[76]:
```python
col_names2 = (list(df.columns)[-31:])
col_names2.extend(['per_capita_personal_income'])

remove_cols = ['attendance_mlb', 'nfl_home_attendance',␣
↪'nfl_wins','total_mlb_teams','home_wins_mlb',\
                    ␣
↪'reg_season_wins_mlb','nfl_win_percentage','nfl_home_attendance','home_games_mlb']
col_names2 = [col for col in col_names2 if col not in remove_cols]
```

[77]:
```python
compareModels(col_names2)
```

```
Linear Regression: R Squared (R2) on test data = 0.823725
Linear Regression: Root Mean Squared Error (RMSE) on test data = 1.42175e+08
```

```
Decision Tree: R Squared (R2) on test data = 0.959344
Decision Tree: Root Mean Squared Error (RMSE) on test data = 6.82793e+07
GBT: R Squared (R2) on test data = 0.966251
GBT: Root Mean Squared Error (RMSE) on test data = 6.22097e+07
```

[78]:
```
col_names3 = (list(df.columns)[-31:])
col_names3.extend(['per_capita_personal_income'])

remove_cols = ['attendance_mlb', 'nfl_home_attendance',␣
↪'nfl_wins','total_mlb_teams','home_wins_mlb', \
              ␣
↪'reg_season_wins_mlb','nfl_win_percentage','nfl_home_attendance', \
               'home_games_mlb', 'large_market','medium_market','small_market']
col_names3 = [col for col in col_names3 if col not in remove_cols]
```

[79]:
```
compareModels(col_names3)
```

```
Linear Regression: R Squared (R2) on test data = 0.806799
Linear Regression: Root Mean Squared Error (RMSE) on test data = 1.04353e+08
Decision Tree: R Squared (R2) on test data = 0.919527
Decision Tree: Root Mean Squared Error (RMSE) on test data = 6.73484e+07
GBT: R Squared (R2) on test data = 0.930767
GBT: Root Mean Squared Error (RMSE) on test data = 6.24679e+07
```

[80]:
```
featuresCoeff(col_names3)
```

[80]:

	features	coefficients
0	total_mlb_team_value	9.917152e+01
1	total_team_revenue_mlb	1.566667e+02
2	world_series_title	-3.993886e+07
3	division_title_mlb	-0.000000e+00
4	attendance_per_game_mlb	-6.693744e+01
5	payroll_mlb	1.024234e+00
6	total_nfl_teams	7.835476e+07
7	nfl_team_values	8.007400e+00
8	nfl_team_revenue	1.020734e+02
9	total_nba_team	6.647545e+07
10	nba_team_value	-0.000000e+00
11	nba_team_revenue	3.000807e+02
12	no_teams	9.946526e+07
13	super_bowl_winner	5.833481e+06
14	nfl_division_champion	-2.978073e+07
15	nfl_playoff_teams	3.408585e+07
16	nfl_attendance_per_game	-9.135435e+02
17	year	-6.724380e+06
18	team_relocated	-1.197504e+08
19	team_purchased	2.827948e+07

```
20  per_capita_personal_income  1.093796e+03
```

[2]:

```
!jupyter nbconvert --to pdf `pwd`/*.ipynb
```

```
[NbConvertApp] WARNING | pattern 'Folder/*.ipynb' matched no files
[NbConvertApp] Converting notebook /sfs/qumulo/qhome/jl3fp/Untitled.ipynb to pdf
[NbConvertApp] Writing 20652 bytes to ./notebook.tex
[NbConvertApp] Building PDF
[NbConvertApp] Running xelatex 3 times: ['xelatex', './notebook.tex', '-quiet']
[NbConvertApp] Running bibtex 1 time: ['bibtex', './notebook']
[NbConvertApp] WARNING | bibtex had problems, most likely because there were no
citations
[NbConvertApp] PDF successfully created
[NbConvertApp] Writing 7599 bytes to /sfs/qumulo/qhome/jl3fp/Untitled.pdf
```