Team 80 - There Goes the Neighborhood: Predicting Gentrification in the United States

Omar Pineda Jr., Melynda Schreiber, Filsan Yousuf, Dominique Reynolds, Naomi Rankin

One of the biggest challenges with gentrification is that people are not aware that it is happening in a community until it is too late to address it and native residents have already been displaced. If local governments are aware that their communities are likely to become gentrified, they can allocate resources, gather community input and enact policies to avoid widespread displacement of native residents. In this project, we develop a model to predict which U.S. counties are at risk of becoming gentrified based on several predictors.

Our main dataset is sourced from the “2020 Healthiest Communities rankings”. The 2020 Healthiest Communities rankings were created in collaboration with the Aetna Foundation, an independent affiliate of CVS Health. The University of Missouri Extension Crnter for Applied Research and Engagement Systems performed data collection and analysis. This data is publicly available through the following website which has a profile for each U.S. county and shows their respective performance across 84 metrics in 10 categories (Population Health, Equity, Education, Economy, Housing, Food & Nutrition, Environment, Public Safety, Community Vitality, and Infrastructure): https://www.usnews.com/news/healthiest-communities/district-of-columbia/district-of-columbia (https://www.usnews.com/news/healthiest-communities/district-of-columbia/district-of-columbia)

First, we load in our data after we have webscraped it and notice that it is tall.

```
df <- read.csv("https://raw.githubusercontent.com/omarp120/Predicting-Gentrification/main/us_news_factors.csv")
head(df)
```

```
##   X       County_Name State_Name
## 1 0 los-alamos-county new-mexico
## 2 1 los-alamos-county new-mexico
## 3 2 los-alamos-county new-mexico
## 4 3 los-alamos-county new-mexico
## 5 4 los-alamos-county new-mexico
## 6 5 los-alamos-county new-mexico
##                                          Variables County    US Peer_Group
## 1                    Hospital Bed Availability /1k    2.8   2.0        2.1
## 2                Population With No Health Insurance   3.0% 10.6%       8.1%
## 3               Primary Care Doctor Availability /100k    2.5   1.1        1.5
## 4         Adults With No Leisure-Time Physical Activity  12.5% 25.9%      22.9%
## 5 Medicare Beneficiaries With Recent Primary Care Visit  31.0% 21.0%      27.0%
## 6                                       Smoking Rate  10.0% 17.3%      15.6%
##   State
## 1   3.0
## 2 11.0%
## 3   1.6
## 4 19.3%
## 5 14.6%
## 6 15.5%
```

We also drop any columns that we will not use at this time, like the state and peer group comparisons for each county.

```
df2 <- subset(df, select = c(County_Name, State_Name, Variables, County))
head(df2)
```

```
##         County_Name State_Name
## 1 los-alamos-county new-mexico
## 2 los-alamos-county new-mexico
## 3 los-alamos-county new-mexico
## 4 los-alamos-county new-mexico
## 5 los-alamos-county new-mexico
## 6 los-alamos-county new-mexico
##                                           Variables County
## 1                       Hospital Bed Availability /1k    2.8
## 2                  Population With No Health Insurance   3.0%
## 3               Primary Care Doctor Availability /100k    2.5
## 4         Adults With No Leisure-Time Physical Activity  12.5%
## 5 Medicare Beneficiaries With Recent Primary Care Visit  31.0%
## 6                                       Smoking Rate  10.0%
```

Here we transform our data from tall to wide so that each column represents a variable. We have a total of 500 counties and 84 different variables that we can use as inputs for our model.

```
df_wide <- df2 %>% spread(Variables, County)
#head(df_wide)
```

In this section, we export our file to Excel so that we can convert our variables to the correct numerical formats and ratios (double or integer), and then we re-import our file into R. This will allow us to properly impute missing values.

```
#write.csv(df_wide, 'all-us-counties-transformed.csv')
df_trans <- read.csv("all-us-counties-transformed.csv")
#head(df_trans)
```

Next, we use KNN imputation to fill in any missing values, using the nearest 2 neighbors to each missing datapoint to approximate the values to impute them with.

```
df_imputed <- knnImputation(df_trans[,3:86], k=2)
```

Here we calculate some summary statistics for all of our variables.

```
summary(df_imputed)
```

```
##  AccidentalDeathRate AdultsinPoororFairGeneralHealth
##  Min.   :0.0001910   Min.   :0.0830
##  1st Qu.:0.0003658   1st Qu.:0.1130
##  Median :0.0004395   Median :0.1225
##  Mean   :0.0004448   Mean   :0.1241
##  3rd Qu.:0.0005144   3rd Qu.:0.1330
##  Max.   :0.0009590   Max.   :0.2690
##  AdultsWithNoLeisure.TimePhysicalActivity AffordableHousingShortfall
##  Min.   :0.0940                           Min.   :-96.00
##  1st Qu.:0.1790                           1st Qu.:-67.90
##  Median :0.2075                           Median :-54.35
##  Mean   :0.2100                           Mean   :-50.73
##  3rd Qu.:0.2400                           3rd Qu.:-38.10
##  Max.   :0.3370                           Max.   : 52.80
##  Air.QualityHazard AirToxicsExposureDisparityIndexScore AirborneCancerRisk
##  Min.   :0.1000    Min.   : 0.000                       Min.   :  9.35
##  1st Qu.:0.2200    1st Qu.: 1.038                       1st Qu.: 17.46
##  Median :0.2700    Median : 2.272                       Median : 21.20
##  Mean   :0.3173    Mean   : 2.780                       Mean   : 25.07
##  3rd Qu.:0.4000    3rd Qu.: 3.842                       3rd Qu.: 28.53
##  Max.   :1.5000    Max.   :21.900                       Max.   :525.56
##  AreaWithTreeCanopy AverageWeeklyWage BabiesBornWithLowBirthWeight
##  Min.   :0.0000     Min.   : 570.0    Min.   :0.02600
##  1st Qu.:0.0290     1st Qu.: 744.8    1st Qu.:0.05900
##  Median :0.0945     Median : 834.0    Median :0.06500
##  Mean   :0.1685     Mean   : 897.4    Mean   :0.06646
##  3rd Qu.:0.2592     3rd Qu.: 970.5    3rd Qu.:0.07300
##  Max.   :0.7920     Max.   :2590.0    Max.   :0.12600
##  BusinessGrowthRate CancerIncidenceRate CensusSelf.ResponseRate
##  Min.   :0.00900    Min.   :0.002410    Min.   :0.3540
##  1st Qu.:0.06600    1st Qu.:0.004123    1st Qu.:0.6980
##  Median :0.08500    Median :0.004491    Median :0.7270
##  Mean   :0.08321    Mean   :0.004420    Mean   :0.7255
##  3rd Qu.:0.10400    3rd Qu.:0.004745    3rd Qu.:0.7610
##  Max.   :0.18700    Max.   :0.005932    Max.   :1.0000
##  ChangeinHousingValue ChildCareFacilitiesRate
##  Min.   :-0.04200     Min.   :0.000e+00
##  1st Qu.: 0.06275     1st Qu.:2.400e-05
##  Median : 0.11700     Median :5.600e-05
##  Mean   : 0.13795     Mean   :9.309e-05
##  3rd Qu.: 0.17800     3rd Qu.:1.363e-04
##  Max.   : 0.58900     Max.   :1.493e-03
##  ChildrenMeetingStandardsinGrade4ELARate
##  Min.   :0.2190
##  1st Qu.:0.5030
##  Median :0.5795
##  Mean   :0.6033
##  3rd Qu.:0.7200
##  Max.   :0.9200
##  ContinuingEducationTaxCreditsasShareofTotalTaxFilings DeathsofDespairRate
##  Min.   :0.0410                                        Min.   :0.0001050
##  1st Qu.:0.0900                                        1st Qu.:0.0002840
##  Median :0.1040                                        Median :0.0003505
```

```
##  Mean   :0.1031                                          Mean   :0.0003532
##  3rd Qu.:0.1160                                          3rd Qu.:0.0004002
##  Max.   :0.2330                                          Max.   :0.0006800
##  DiabetesPrevalence DisabilityEmploymentGap  EvictionRate
##  Min.   :0.01500    Min.   :0.2200          Min.   :0.00000
##  1st Qu.:0.06200    1st Qu.:0.7100          1st Qu.:0.00300
##  Median :0.07300    Median :0.7700          Median :0.00800
##  Mean   :0.07478    Mean   :0.7754          Mean   :0.01095
##  3rd Qu.:0.08800    3rd Qu.:0.8300          3rd Qu.:0.01600
##  Max.   :0.19300    Max.   :1.2200          Max.   :0.07900
##  ExtremeHeatDaysperYeardays FoodEnvironmentIndexScore GiniIndexScore
##  Min.   : 2.300             Min.   : 0.000            Min.   :0.3600
##  1st Qu.: 7.000             1st Qu.: 8.935            1st Qu.:0.4100
##  Median : 8.700             Median :10.895            Median :0.4300
##  Mean   : 8.916             Mean   :12.719            Mean   :0.4299
##  3rd Qu.:10.300             3rd Qu.:15.220            3rd Qu.:0.4500
##  Max.   :26.000             Max.   :58.360            Max.   :0.6000
##  HeartDiseasePrevalenceAmongMedicareBeneficiaries HighSchoolGraduationRate
##  Min.   :0.1400                                   Min.   :0.6540
##  1st Qu.:0.2000                                   1st Qu.:0.8700
##  Median :0.2300                                   Median :0.9100
##  Mean   :0.2276                                   Mean   :0.8998
##  3rd Qu.:0.2500                                   3rd Qu.:0.9350
##  Max.   :0.3400                                   Max.   :0.9750
##  HomeownershipRate HospitalBedAvailabilityRate HouseholdsinFloodHazardZone
##  Min.   :0.2410    Min.   :0.000000            Min.   :0.00000
##  1st Qu.:0.6953    1st Qu.:0.001100            1st Qu.:0.02100
##  Median :0.7445    Median :0.001900            Median :0.03400
##  Mean   :0.7340    Mean   :0.002542            Mean   :0.04743
##  3rd Qu.:0.7850    3rd Qu.:0.003000            3rd Qu.:0.05700
##  Max.   :0.8970    Max.   :0.042900            Max.   :0.72500
##  HouseholdsReceivingPublicAssistanceIncome
##  Min.   :0.00000
##  1st Qu.:0.01200
##  Median :0.01700
##  Mean   :0.01763
##  3rd Qu.:0.02200
##  Max.   :0.05600
##  HouseholdsSpendingatLeast30.ofIncomeonHousing
##  Min.   :0.1070
##  1st Qu.:0.2000
##  Median :0.2375
##  Mean   :0.2493
##  3rd Qu.:0.2963
##  Max.   :0.4440
##  HouseholdsWithIncompletePlumbingFacilities HouseholdsWithInternetAccess
##  Min.   :0.0000                             Min.   :0.2200
##  1st Qu.:0.0010                             1st Qu.:0.8480
##  Median :0.0030                             Median :0.9495
##  Mean   :0.0038                             Mean   :0.8970
##  3rd Qu.:0.0050                             3rd Qu.:0.9842
##  Max.   :0.0280                             Max.   :1.0000
##  HouseholdsWithNoVehicle IdleYouth.NotWorkingorEnrolled. JobDiversityIndexScore
##  Min.   :0.00500         Min.   :0.00000                 Min.   :0.1200
```

```
##  1st Qu.:0.03300         1st Qu.:0.00700                 1st Qu.:0.5700
##  Median :0.04400         Median :0.01400                 Median :0.7200
##  Mean   :0.04721         Mean   :0.01822                 Mean   :0.6802
##  3rd Qu.:0.05500         3rd Qu.:0.02400                 3rd Qu.:0.8300
##  Max.   :0.77000         Max.   :0.15900                 Max.   :0.9200
##  JobsWithi45.MinuteCommute LaborForceParticipation LifeExpectancy
##  Min.   :    289           Min.   :0.3840          Min.   :77.30
##  1st Qu.:   2630           1st Qu.:0.6310          1st Qu.:79.70
##  Median :   8052           Median :0.6600          Median :80.40
##  Mean   :  33923           Mean   :0.6565          Mean   :80.48
##  3rd Qu.:  34160           3rd Qu.:0.6870          3rd Qu.:81.20
##  Max.   :1059922           Max.   :0.7970          Max.   :86.80
##  LocalFoodOutletsRate MedianHouseholdIncome MedicalDebtinCollections
##  Min.   :0.000e+00    Min.   : 31468        Min.   :0.0000
##  1st Qu.:3.175e-05    1st Qu.: 55878        1st Qu.:0.0700
##  Median :7.500e-05    Median : 62977        Median :0.1200
##  Mean   :1.040e-04    Mean   : 67930        Mean   :0.1156
##  3rd Qu.:1.433e-04    3rd Qu.: 77788        3rd Qu.:0.1500
##  Max.   :5.810e-04    Max.   :136268        Max.   :0.3100
##  MedicareBeneficiariesWithDepression
##  Min.   :0.1000
##  1st Qu.:0.1600
##  Median :0.1700
##  Mean   :0.1727
##  3rd Qu.:0.1900
##  Max.   :0.2400
##  MedicareBeneficiariesWithRecentPrimaryCareVisit NaturalAmenitiesIndexScore
##  Min.   :0.0200                                  Min.   :-1.9700
##  1st Qu.:0.1800                                  1st Qu.:-1.3100
##  Median :0.2600                                  Median :-0.3748
##  Mean   :0.2579                                  Mean   : 0.5583
##  3rd Qu.:0.3225                                  3rd Qu.: 1.6275
##  Max.   :0.5600                                  Max.   :11.1700
##  NeighborhoodDisparityinEducatiolAttainment NeighborhoodDisparityinPoverty
##  Min.   : 2.690                             Min.   : 0.590
##  1st Qu.: 7.835                             1st Qu.: 2.993
##  Median :11.570                             Median : 4.220
##  Mean   :11.877                             Mean   : 5.060
##  3rd Qu.:15.418                             3rd Qu.: 5.968
##  Max.   :27.940                             Max.   :26.340
##  NetMigrationRate   NonprofitsRate      ObesityPrevalence OvercrowdedHouseholds
##  Min.   :-0.12500   Min.   :0.0001570   Min.   :0.1230    Min.   :0.00000
##  1st Qu.:-0.02000   1st Qu.:0.0004337   1st Qu.:0.2490    1st Qu.:0.00900
##  Median : 0.00650   Median :0.0005485   Median :0.2930    Median :0.01350
##  Mean   : 0.02802   Mean   :0.0006040   Mean   :0.2882    Mean   :0.01733
##  3rd Qu.: 0.05125   3rd Qu.:0.0006985   3rd Qu.:0.3270    3rd Qu.:0.02000
##  Max.   : 0.76400   Max.   :0.0020290   Max.   :0.4680    Max.   :0.10500
##  Per.PupilExpenditures PerCapitaSpendingonHealthandEmergencyServices
##  Min.   : 6198         Min.   :   0.0
##  1st Qu.:11986         1st Qu.: 330.0
##  Median :13717         Median : 445.0
##  Mean   :14937         Mean   : 549.4
##  3rd Qu.:17218         3rd Qu.: 628.8
##  Max.   :45688         Max.   :8862.0
```

```
##  PoorMentalHealthDaysperMonth PopulationLivingClosetoEmergencyFacilities
##  Min.   :2.500                Min.   :0.0390
##  1st Qu.:3.100                1st Qu.:0.3040
##  Median :3.300                Median :0.4160
##  Mean   :3.327                Mean   :0.4176
##  3rd Qu.:3.600                3rd Qu.:0.5350
##  Max.   :4.500                Max.   :1.0000
##  PopulationWithAdvancedDegree PopulationWithNoHealthInsurance
##  Min.   :0.1390               Min.   :0.02300
##  1st Qu.:0.3460               1st Qu.:0.05300
##  Median :0.4105               Median :0.06900
##  Mean   :0.4252               Mean   :0.07785
##  3rd Qu.:0.4858               3rd Qu.:0.09600
##  Max.   :0.8230               Max.   :0.23800
##  PopulationWithin0.5MileofaPark PopulationWithin0.5MileofWalkableDestitions
##  Min.   :0.0000                   Min.   :0.0000
##  1st Qu.:0.1638                   1st Qu.:0.0640
##  Median :0.2960                   Median :0.1305
##  Mean   :0.3283                   Mean   :0.1806
##  3rd Qu.:0.4600                   3rd Qu.:0.2532
##  Max.   :1.0000                   Max.   :0.9680
##  PopulationWithoutAccesstoLargeGroceryStore  PovertyRate
##  Min.   :0.0000                             Min.   :0.02700
##  1st Qu.:0.1258                             1st Qu.:0.07100
##  Median :0.2140                             Median :0.09000
##  Mean   :0.2287                             Mean   :0.09257
##  3rd Qu.:0.3160                             3rd Qu.:0.10900
##  Max.   :0.7550                             Max.   :0.24700
##  PrematureDeathDisparityIndexScore PreschoolEnrollment
##  Min.   :0.00000                   Min.   :0.1430
##  1st Qu.:0.02000                   1st Qu.:0.4050
##  Median :0.03000                   Median :0.4880
##  Mean   :0.03260                   Mean   :0.4882
##  3rd Qu.:0.03668                   3rd Qu.:0.5645
##  Max.   :0.17000                   Max.   :0.9640
##  PreventableHospitalAdmissionsAmongMedicareBeneficiariesRate
##  Min.   :0.00860
##  1st Qu.:0.03008
##  Median :0.03685
##  Mean   :0.03825
##  3rd Qu.:0.04442
##  Max.   :0.10020
##  PrimaryCareDoctorAvailabilityRate PropertyCrimeRate
##  Min.   :0.000e+00                 Min.   :0.00050
##  1st Qu.:1.100e-05                 1st Qu.:0.00880
##  Median :1.400e-05                 Median :0.01310
##  Mean   :1.502e-05                 Mean   :0.01381
##  3rd Qu.:1.900e-05                 3rd Qu.:0.01730
##  Max.   :1.000e-04                 Max.   :0.05480
##  PublicSafetyProfessiolsinPopulation RacialDisparityinEducatiolAttainment
##  Min.   :0.000000                      Min.   :0.0100
##  1st Qu.:0.004000                      1st Qu.:0.1372
##  Median :0.006500                      Median :0.1771
##  Mean   :0.007142                      Mean   :0.2100
```

```
##  3rd Qu.:0.009000                   3rd Qu.:0.2903
##  Max.   :0.050000                   Max.   :0.6500
##  RacialDisparityinPoverty SegregationIndexScore
##  Min.   :0.0100           Min.   :0.1200
##  1st Qu.:0.0900           1st Qu.:0.2700
##  Median :0.1138           Median :0.3400
##  Mean   :0.1265           Mean   :0.3639
##  3rd Qu.:0.1600           3rd Qu.:0.4425
##  Max.   :0.3300           Max.   :0.7500
##  ShareofAt.HomeFoodExpendituresonFruit.Veg  SmokingRate     TeenBirthRate
##  Min.   :0.1120                            Min.   :0.0670   Min.   :0.00290
##  1st Qu.:0.1180                            1st Qu.:0.1340   1st Qu.:0.01030
##  Median :0.1210                            Median :0.1440   Median :0.01340
##  Mean   :0.1231                            Mean   :0.1434   Mean   :0.01438
##  3rd Qu.:0.1290                            3rd Qu.:0.1550   3rd Qu.:0.01773
##  Max.   :0.1500                            Max.   :0.2010   Max.   :0.04470
##  ToxicReleaseIndexScore UnemploymentRate  UnsafeDrinkingWater  VacantHouses
##  Min.   :       0       Min.   :0.01600   Min.   :0.00000     Min.   :0.00000
##  1st Qu.:       0       1st Qu.:0.02600   1st Qu.:0.00000     1st Qu.:0.00700
##  Median :     301       Median :0.03000   Median :0.00000     Median :0.01400
##  Mean   :  115337       Mean   :0.03097   Mean   :0.03708     Mean   :0.02229
##  3rd Qu.:    8423       3rd Qu.:0.03500   3rd Qu.:0.01425     3rd Qu.:0.02925
##  Max.   :16326582       Max.   :0.06900   Max.   :1.00000     Max.   :0.16600
##  VehicleCrashFatalityRate ViolentCrimeRate.100k VoterParticipationRate
##  Min.   :0.000e+00        Min.   :0.000000      Min.   :0.4270
##  1st Qu.:8.175e-05        1st Qu.:0.000837      1st Qu.:0.6280
##  Median :1.210e-04        Median :0.001264      Median :0.6715
##  Mean   :1.523e-04        Mean   :0.001531      Mean   :0.6717
##  3rd Qu.:1.960e-04        3rd Qu.:0.001911      3rd Qu.:0.7200
##  Max.   :6.540e-04        Max.   :0.007605      Max.   :0.8790
##  WalkabilityIndexScore WorkHoursNeededtoPayforAffordableHousing
##  Min.   : 3.200        Min.   : 18.00
##  1st Qu.: 6.200        1st Qu.: 37.00
##  Median : 6.700        Median : 43.00
##  Mean   : 7.247        Mean   : 45.44
##  3rd Qu.: 7.700        3rd Qu.: 51.00
##  Max.   :16.200        Max.   :112.00
##  WorkersCommuting60MinutesorMore YouthWithin5MilesofaPublicSchool
##  Min.   :0.00700                 Min.   :0.3670
##  1st Qu.:0.03700                 1st Qu.:0.7268
##  Median :0.05150                 Median :0.8475
##  Mean   :0.07044                 Mean   :0.8297
##  3rd Qu.:0.08425                 3rd Qu.:0.9650
##  Max.   :0.34800                 Max.   :1.0000
```

There are too many different correlation pairs to attempt observing as we have 84 different variables, so we instead look at which variables have the strongest relationships in our dataset.

Below are the top 20 strongest correlation pairs in our dataset. It appears that the two variables with the strongest correlation are NeighborhoodDisparityinEducationAttainment and PopulationWithAdvancedDegree.

```
#devtools::install_github("laresbernardo/lares")
library(lares)

corr_cross(df_imputed, # name of dataset
  max_pvalue = 0.05, # display only significant correlations (at 5% level)
  top = 20
)
```

```
## Returning only the top 20. You may override with the 'top' argument
```

```
## Warning in theme_lares2(legend = "top"): Font Arial Narrow is not installed, has
## other name, or can't be found
```

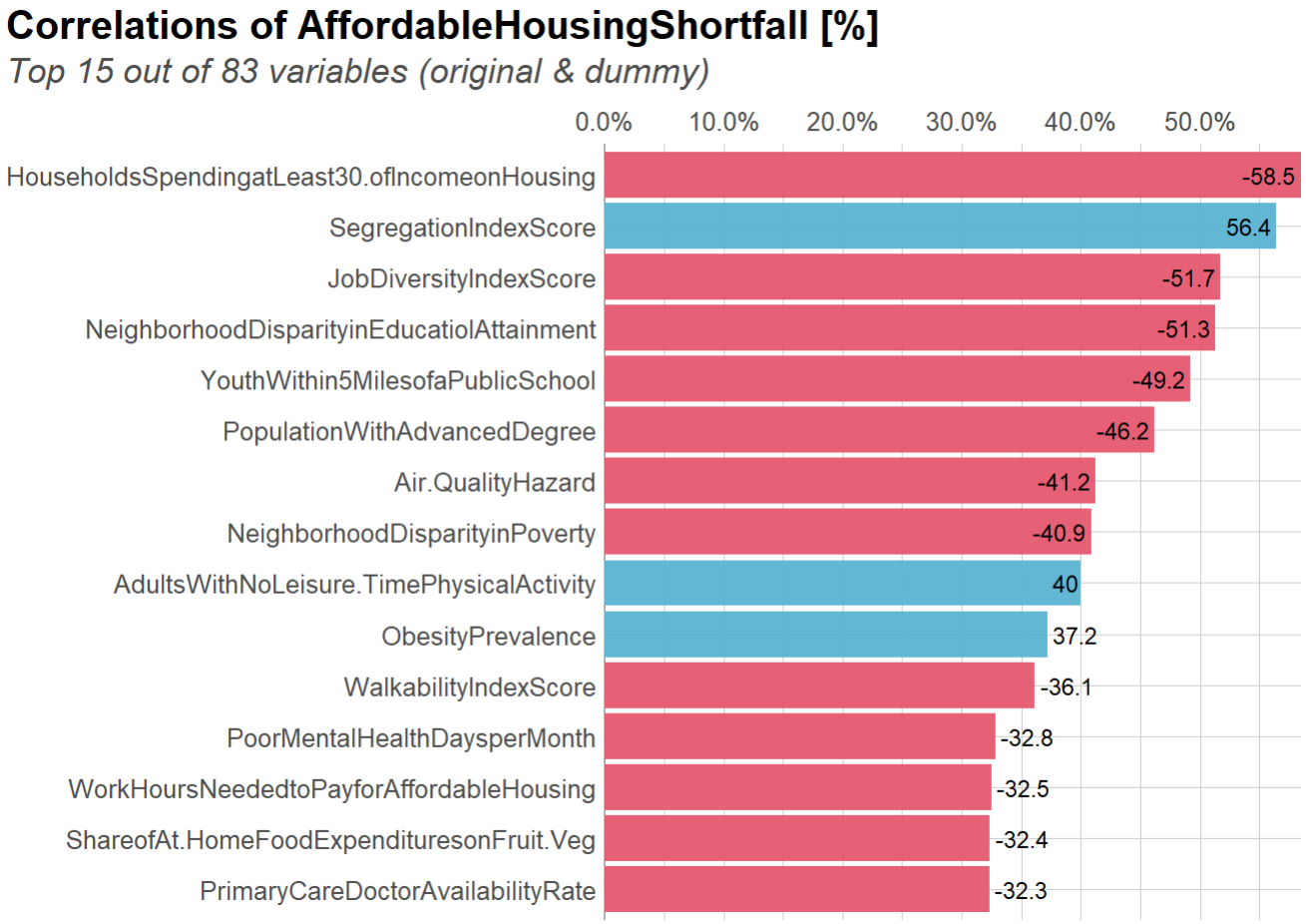
Ranked Cross-Correlations

20 most relevant

Next, we look at which variables are most highly correlated with our 'AffordableHousingShortfall' variable which is our proxy variable for predicting gentrification. The data source defines this variable as the "Availability of affordable housing relative to a community's low-income population. Negative numbers indicate a shortfall."

It appears that "Households Spending at least 30% of income on Housing" is the strongest predictor of an affordable housing shortfall, or gentrification.

```
corr_var(df_imputed, # name of dataset
  AffordableHousingShortfall, # name of variable to focus on
  top = 15 # display top 5 correlations
)
```



Next, we work on developing 7 different models for predicting gentrification using our proxy AffordableHousingShortfall variable. We will select the “best” model by evaluating them in terms of their RMSE.

We start by splitting our data into a training and test set, using 80% of our data to train our models and holding out 20% to test them.

```
#data splitting
set.seed(101)

sample = sample.split(df_imputed$AffordableHousingShortfall, SplitRatio = .8)

train = subset(df_imputed, sample == TRUE)
test  = subset(df_imputed, sample == FALSE)

train_X = subset(train, select = -AffordableHousingShortfall)
train_y = train[,'AffordableHousingShortfall']

test_X = subset(test, select = -AffordableHousingShortfall)
test_y = test[,'AffordableHousingShortfall']
```

Linear Regression Model

Linear regression is an attractive model because representation is simply done. The representation is a linear equation that combines a specific set of input values (x) the solution to which is the predicted output for that set of input values (y). In this section, we will be predicting AffordableHousingShortfall values through three different

linear regression techniques.

GLM

The generalized linear model (GLM) is a flexible generalization of ordinary linear regression that allows for response variables that have error distribution models other than a normal distribution. The GLM generalizes linear regression by allowing the linear model to be related to the response variable via a link function and by allowing the magnitude of the variance of each measurement to be a function of its predicted value.

```
control = trainControl(method = 'cv', number = 5,
  verboseIter = FALSE, savePredictions = TRUE,allowParallel = T)
```

```
set.seed(17)
GLM_train = train(AffordableHousingShortfall ~ ., data = train, metric = 'RMSE', method = 'glm',
preProcess = c('center', 'scale'), trControl = control)
GLM_reg_pred <- predict(GLM_train, test_X)

GLM_train
```

```
## Generalized Linear Model
##
## 400 samples
##  83 predictor
##
## Pre-processing: centered (83), scaled (83)
## Resampling: Cross-Validated (5 fold)
## Summary of sample sizes: 320, 320, 320, 320, 320
## Resampling results:
##
##   RMSE      Rsquared   MAE
##   19.19956  0.4346039  13.68569
```

Using RMSE as a benchmark for linear regression modelling, we determined that our GLM model performed quite well at 19.2. We will attempt two other iterations with different linear techniques to determine if our baseline can be improved.

glmnet

glmnet is an extremely efficient procedure for fitting the entire lasso or elastic-net regularization path for linear regression, logistic and multinomial regression models, Poisson regression and the Cox model. Two recent additions are the multiple-response Gaussian, and the grouped multinomial regression. The algorithm uses cyclical coordinate descent in a path-wise fashion to determine the best linear fit.

```
set.seed(17)
glmnet_train = train(AffordableHousingShortfall ~ ., data = train , metric = 'RMSE', method = 'g
lmnet',preProcess = c('center', 'scale'), trControl = control)
glmnet_reg_pred <- predict(glmnet_train, test_X)

glmnet_train
```

```
## glmnet
##
## 400 samples
##  83 predictor
##
## Pre-processing: centered (83), scaled (83)
## Resampling: Cross-Validated (5 fold)
## Summary of sample sizes: 320, 320, 320, 320, 320
## Resampling results across tuning parameters:
##
##   alpha  lambda      RMSE      Rsquared   MAE
##   0.10   0.02846147  19.12225  0.4370419  13.63454
##   0.10   0.28461473  18.73773  0.4488048  13.37479
##   0.10   2.84614733  17.73569  0.4756701  12.57925
##   0.55   0.02846147  19.04562  0.4395182  13.57765
##   0.55   0.28461473  18.30699  0.4588184  13.05158
##   0.55   2.84614733  17.60362  0.4802538  12.55867
##   1.00   0.02846147  18.96178  0.4421959  13.51636
##   1.00   0.28461473  18.03679  0.4643275  12.84684
##   1.00   2.84614733  18.11644  0.4569902  13.00166
##
## RMSE was used to select the optimal model using the smallest value.
## The final values used for the model were alpha = 0.55 and lambda = 2.846147.
```

The final values used for the GLM Net model were alpha = 0.55 and lambda = 2.85, which produced an RMSE of 17.6. This is a slight improvement over the GLM model. The close results can be explained by low lambda value, where a zero lambda is in effect a standard glm model.

Partial Least Squares

Partial least squares (PLS) is a method for constructing predictive models when the factors are many and highly collinear. Partial least squares is a popular method for soft modeling in industrial applications. We believed that it would be an effective model to demonstrate.

```
set.seed(17)
pls_train = train(AffordableHousingShortfall ~ ., data = train , metric = 'RMSE', method = 'pls'
,preProcess = c('center', 'scale'), trControl = control)
pls_reg_pred <- predict(pls_train, test_X)

pls_train
```

```
## Partial Least Squares
##
## 400 samples
##  83 predictor
##
## Pre-processing: centered (83), scaled (83)
## Resampling: Cross-Validated (5 fold)
## Summary of sample sizes: 320, 320, 320, 320, 320
## Resampling results across tuning parameters:
##
##   ncomp  RMSE      Rsquared   MAE
##   1      18.99295  0.3939885  13.54142
##   2      18.11396  0.4505461  12.81498
##   3      18.22649  0.4573401  12.98211
##
## RMSE was used to select the optimal model using the smallest value.
## The final value used for the model was ncomp = 2.
```

The final value used for the model was ncomp = 2, and it produced an RMSE of 18.11. The GLM Net model outperformed this with a RMSE of 17.6.

Non-linear Regression Models

In this section, we are going to fit a simple neural network using the neuralnet package.

Here we confirm that there are no more empty data:

```
#apply(df_imputed,2,function(x) sum(is.na(x)))
```

Preparing to fit the neural network

Before fitting a neural network, some preparation needs to be done.

As a first step, we are going to address data preprocessing. We will be normalizing the data before training a neural network. We chose to use the min-max method and scale the data in the interval [0,1]. Usually scaling in the intervals [0,1] or [-1,1] tends to give better results.

We therefore scale and split the data before moving forward:

```
maxs <- apply(df_imputed, 2, max)
mins <- apply(df_imputed, 2, min)
```

Scaled returns a matrix that needs to be coerced into a data.frame.

```
scaled <- as.data.frame(scale(df_imputed, center = mins, scale = maxs - mins))
#scaled
index <- sample(1:nrow(df_imputed),round(0.75*nrow(df_imputed)))
train_ <- scaled[index,]
test_ <- scaled[-index,]
```

Parameters

In this dataset, we are going to use 2 hidden layers with this configuration: 83:5:3:1. The input layer has 83 inputs, the two hidden layers have 5 and 3 neurons and the output layer has, of course, a single output since we are doing regression.

Let's fit the net: Setting the linear.output = True does regression instead of classification.

```
n <- names(train_)
f <- as.formula(paste("AffordableHousingShortfall ~", paste(n[!n %in% "AffordableHousingShortfal
l"], collapse = " + ")))
nn <- neuralnet(f,data=train_,hidden=c(5,3),linear.output=T)
```

Plot the Neural Network

```
plot(nn)
```

The black lines show the connections between each layer and the weights on each connection while the blue lines show the bias term added in each step. The bias can be thought as the intercept of a linear model. The net is essentially a black box so we cannot say that much about the fitting, the weights and the model. Suffice to say that the training algorithm has converged and therefore the model is ready to be used.

Predicting AffordableHousingShortfall using the neural network

Now we can try to predict the values for the test set and calculate the RMSE. The net will output a normalized prediction, so we need to scale it back in order to make a meaningful comparison (or just a simple prediction).

```
pr.nn <- compute(nn,test_[,1:84])
pr.nn_ <- pr.nn$net.result*(max(df_imputed$AffordableHousingShortfall)-min(df_imputed$Affordable
HousingShortfall))+min(df_imputed$AffordableHousingShortfall)
test.r <- (test_$AffordableHousingShortfall)*(max(df_imputed$AffordableHousingShortfall)-min(df_
imputed$AffordableHousingShortfall))+min(df_imputed$AffordableHousingShortfall)
MSE.nn <- sum((test.r - pr.nn_)^2)/nrow(test_)
```

The RMSE for our Neural Network is 31.55 which does not perform as well as our GLM net model.

```
nn.rmse <- sqrt(MSE.nn)
print(nn.rmse)
```

```
## [1] 31.54765
```

Tree Models

In this next section, we consider various tree models to predict the AffordableHousingShortfall, or gentrification, of a U.S. county given the 83 other indicators.

Basic Regression Tree

Classification and regression trees can be generated througn rpart to create simple tree models. Tree-based models consist of one or more nested if-then statements for the predictors that partition the data. A model is used to predict the outcome within these partitions.

```
treeb <- train(x = train_X, y = train_y, method = "rpart", preProcess = c('center', 'scale'))
```

```
## Warning in nominalTrainWorkflow(x = x, y = y, wts = weights, info = trainInfo, :
## There were missing values in resampled performance measures.
```

```
treeb
```

```
## CART
##
## 400 samples
##  83 predictor
##
## Pre-processing: centered (83), scaled (83)
## Resampling: Bootstrapped (25 reps)
## Summary of sample sizes: 400, 400, 400, 400, 400, 400, ...
## Resampling results across tuning parameters:
##
##   cp          RMSE      Rsquared   MAE
##   0.06071757  21.18997  0.2847308  15.21984
##   0.07429463  21.34260  0.2719466  15.34983
##   0.32829427  21.59173  0.2794404  15.95130
##
## RMSE was used to select the optimal model using the smallest value.
## The final value used for the model was cp = 0.06071757.
```

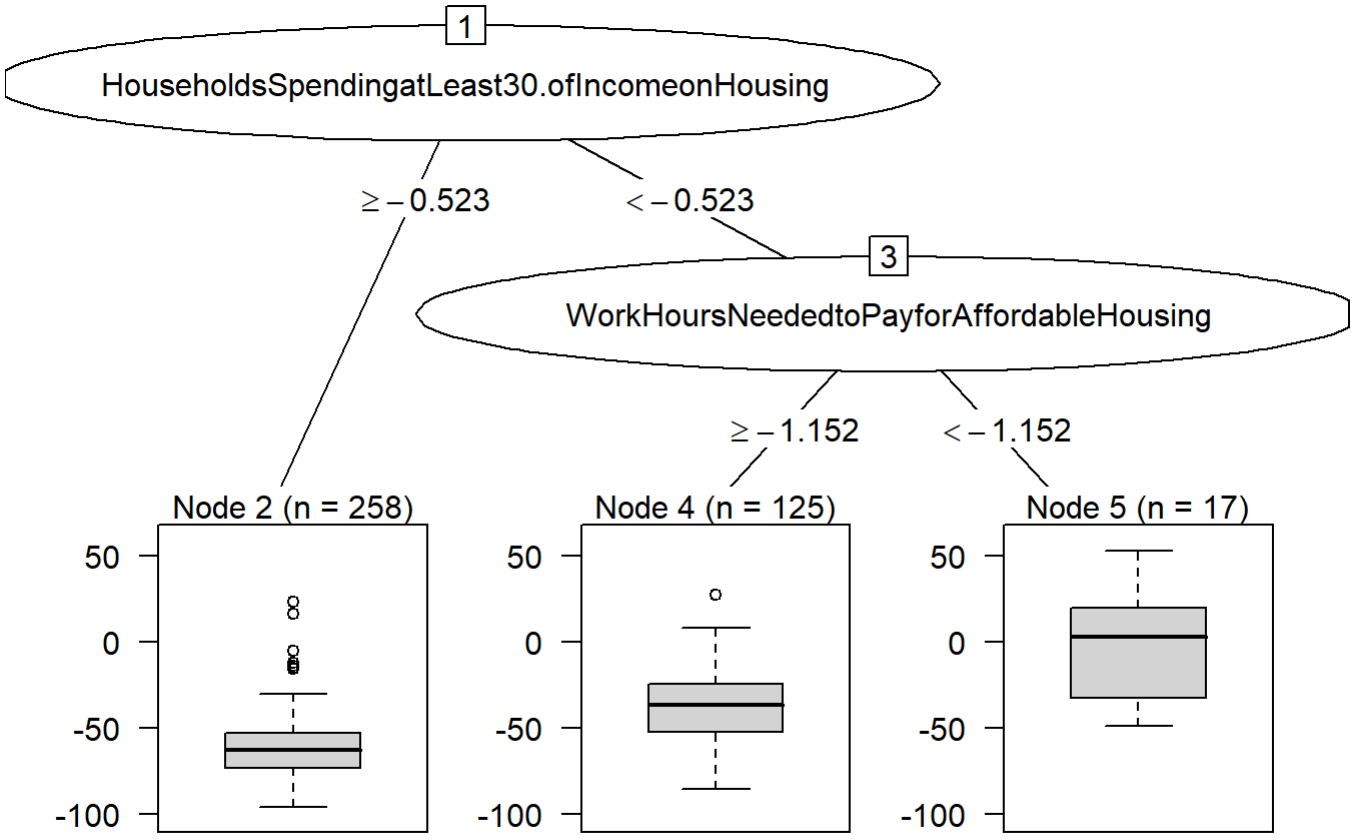
```
treebPred <- predict(treeb, newdata = test_X)
treeb.results <- postResample(pred = treebPred, obs = test_y)
treeb.results
```

```
##       RMSE   Rsquared        MAE
## 24.0467330  0.1727561 17.6478419
```

The RMSE for this basic regression tree model is 24.05 with an R^2 of 0.17.

We also plot this specific tree below:

```
plot(as.party(treeb$finalModel))
```



Next, we will try out a Random Forest model to see if we can improve upon this model.

Random Forest

Random Forest is an ensemble model where each tree splits out a class prediction and the class with the most contributions becomes the model's prediction value. Random Forest creates as many trees on the subset of the data and combines the output of all the trees. This thus reduces problems in overfitting and reduces the variance.

```
rf <- train(x = train_X, y = train_y, method = "rf", preProcess = c('center', 'scale'))
rf
```

```
## Random Forest
##
## 400 samples
##  83 predictor
##
## Pre-processing: centered (83), scaled (83)
## Resampling: Bootstrapped (25 reps)
## Summary of sample sizes: 400, 400, 400, 400, 400, 400, ...
## Resampling results across tuning parameters:
##
##   mtry  RMSE      Rsquared   MAE
##    2    18.49052  0.4460399  12.84827
##   42    18.01823  0.4521409  12.43415
##   83    18.27414  0.4352865  12.62710
##
## RMSE was used to select the optimal model using the smallest value.
## The final value used for the model was mtry = 42.
```

```
rfPred <- predict(rf, newdata = test_X)
rf.results <- postResample(pred = rfPred, obs = test_y)
rf.results
```

```
##       RMSE   Rsquared        MAE
## 18.8211486  0.4164222 13.3131592
```

The RMSE for this model is 18.82 with a RM^2 of 0.416. This Random Forest model performes better than our Basic Regression Tree model. We will also consider an XGBoost model to see if we can find any improvements.

XGBoost

XGBoost is another ensemble model, this time using the gradient boosting framework which is a special case of boosting where errors are minimized by gradient descent algorithm. XGBoost only manages numeric vectors, and luckily all of our variables are numeric.

```
xgbPred <- predict(xgb, newdata = test_X)
xgb.results <- postResample(pred = xgbPred, obs = test_y)
xgb.results
```

```
##       RMSE   Rsquared        MAE
## 20.0079475  0.3574677 14.1354080
```

The RMSE for this model is 20.01 with a R^2 of 0.357.

We considered Basic Regression, Random Forest and XGBoost tree models, and Random Forest performed the best out of the three in predicting AffordableHousingShortfall as it had the smallest RMSE value at 18.82 with a R^2 of 0.416.

Conclusion

Finally, we will choose our best model for predicting AffordableHousingShortfall among the chosen Linear (GLM Net), Non-Linear (Neural Network) and Tree Models (Random Forest) that we have gone over in this analysis. Their RMSE metrics are summarized here:

```
lin_model_perf <- getTrainPerf(glmnet_train)
print(lin_model_perf)
```

```
##   TrainRMSE TrainRsquared TrainMAE method
## 1  17.60362     0.4802538 12.55867 glmnet
```

```
print(nn.rmse)
```

```
## [1] 31.54765
```

```
rf_perf <- as.data.frame(as.list(rf.results))
print(rf_perf)
```

```
##       RMSE  Rsquared      MAE
## 1 18.82115 0.4164222 13.31316
```

The RMSE for the chosen Linear Model (GLM Net) was 17.604. The RMSE for the chosen Non-Linear Model (Neural Net) was 31.54. And lastly, the RMSE for the chosen Tree model (Random Forest) was 18.82. So, we chose the GLM Net Model as our final model to predict the AffordableHousingShortfall of our counties given the predictors in our dataset.

In this final chosen GLM Net model, the top 5 predictors that we found to influence AffordableHousingShortfall, or gentrification, for counties are HouseholdsSpendingatLeast30.ofIncomeonHousing, SegregationIndexScore, JobDiversityIndexScore, PopulationWithAdvancedDegree, and UnemploymentRate.

```
varImp(glmnet_train)
```

```
## glmnet variable importance
##
##   only 20 most important variables shown (out of 83)
##
##                                              Overall
## HouseholdsSpendingatLeast30.ofIncomeonHousing   100.0000
## SegregationIndexScore                            70.5762
## JobDiversityIndexScore                           49.8749
## PopulationWithAdvancedDegree                     39.3378
## UnemploymentRate                                 27.1655
## PovertyRate                                      25.2956
## CancerIncidenceRate                              16.5135
## Air.QualityHazard                                16.2184
## YouthWithin5MilesofaPublicSchool                 14.9089
## NeighborhoodDisparityinPoverty                   14.3494
## MedicareBeneficiariesWithRecentPrimaryCareVisit  14.2138
## HouseholdsWithNoVehicle                          11.2255
## PopulationWithoutAccesstoLargeGroceryStore        9.3253
## MedicalDebtinCollections                          8.5635
## NonprofitsRate                                    8.3418
## PopulationWithNoHealthInsurance                   8.1551
## PoorMentalHealthDaysperMonth                      5.4384
## PrimaryCareDoctorAvailabilityRate                 4.0731
## EvictionRate                                      1.3699
## ChildCareFacilitiesRate                           0.4047
```

We should pay close attention to these predictors when trying to learn more about and control for the gentrification of counties in the U.S. For example, efforts should be done so that households are not spending more than 30% of their income on housing. We should also desegregate neighborhoods, promote diversity in jobs, and address higher education disparities in order to prevent the gentrificaiton of U.S. counties.