Should I Stay or Should I Joe: Predicting Trader Joe’s Presence in a Zip Code

Lauren Sun
Fall 2024

Background

Trader Joe’s, Starbucks, and Gentrification

- Much beloved grocery chain
    - Quirky branding
    - Unique business model
- Caters to educated middle class
    - “overeducated and underpaid” - Joe Coulombe

Starbucks effect?

- Starbucks location is an indicator of home price increases

Gentrification

- Increase in home prices
- influx of educated, wealthier residents
- Demographic shift

Problem Formulation and Data

Predict whether or not there is a Trader Joe's store in a given zip code using demographic and economic US Census data.

521 Trader Joe's locations (2021), 505 zip codes

33,122 zip codes US Census Bureau American Community Survey

8971 cities in Zillow Rent Zestimates (Price per square foot)

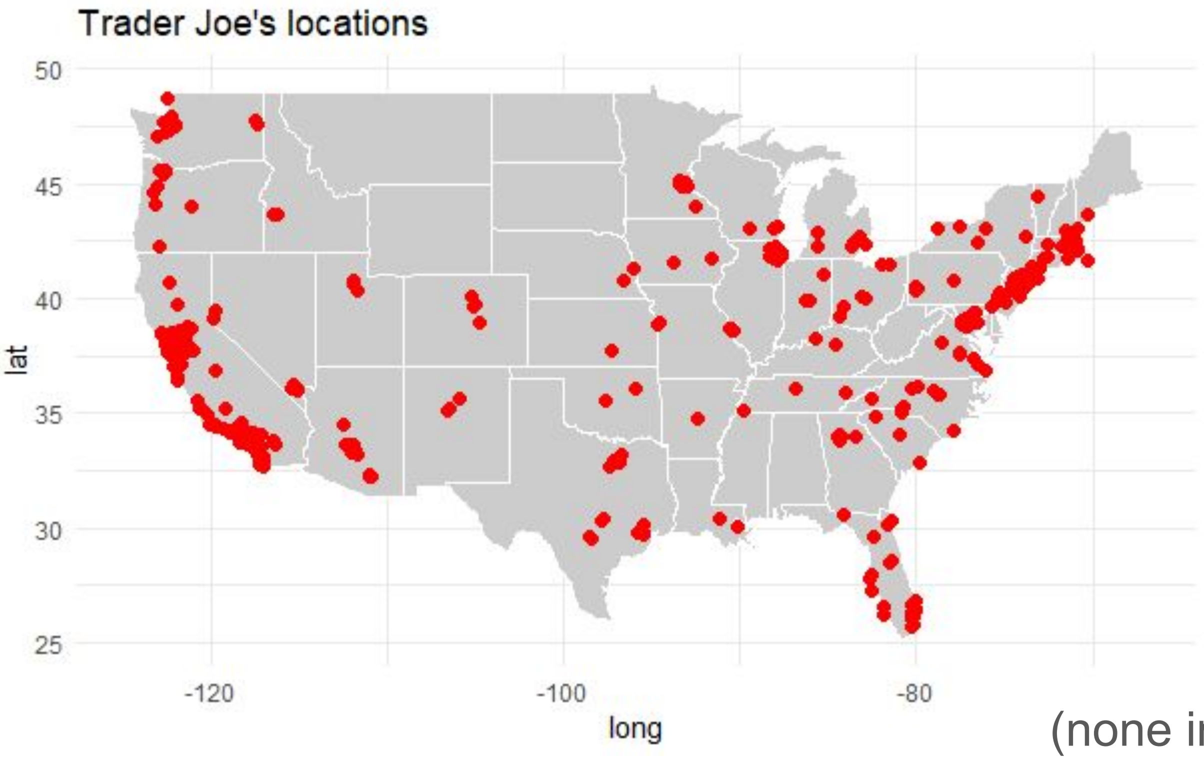
499 ZIP codes with Trader Joe's

28817 ZIP codes without

75/25 Train/Test split

Baseline = 0.983

Locations



(none in AK or HI)

Predictors Shortlist

21 predictors:

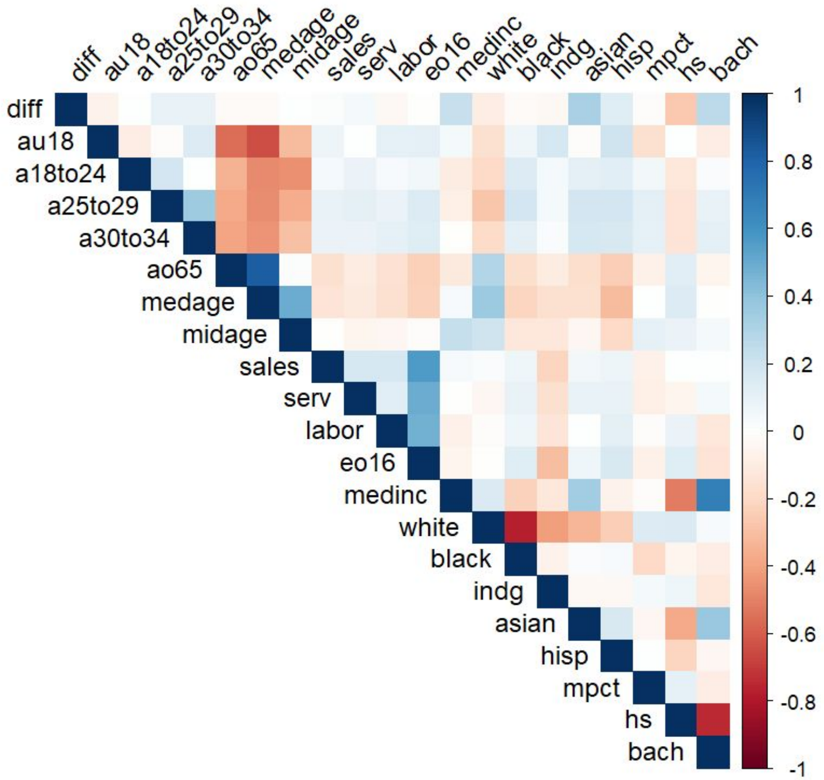
- % HS diploma
- % 4-year degree
- % households with children
- % White households
- % Black households
- % Indigenous households
- % Asian households
- % Hispanic / Latino households

- %M
- median age
- % aged < 18
- % aged 18-24
- % aged 25-29
- % aged 30-34
- % aged 35-64
- % aged > 65

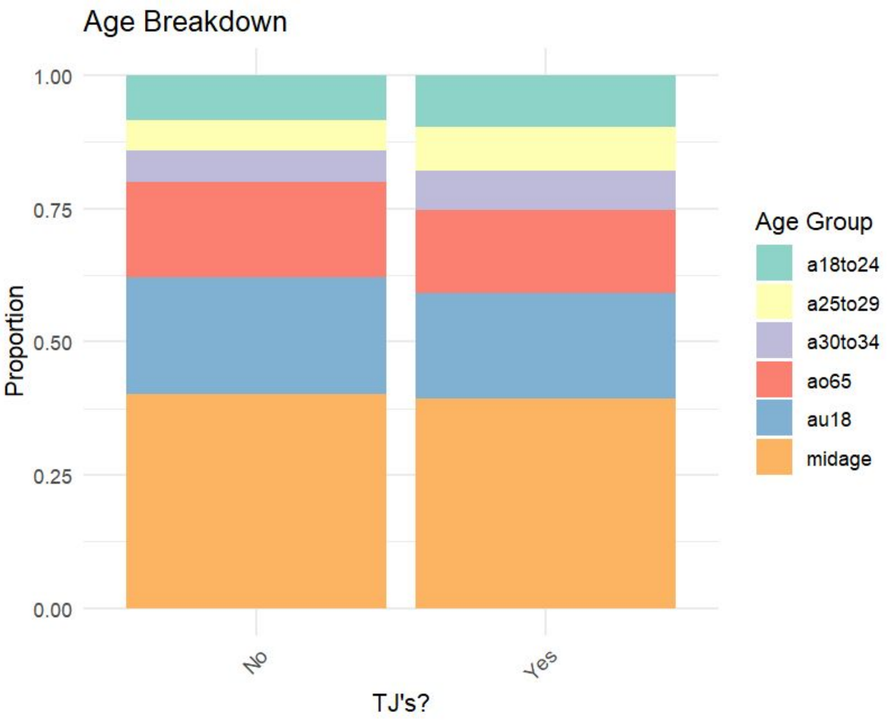
- median income
- % employed (>16 yo)
- % service jobs
- % sales/office jobs
- % labor jobs

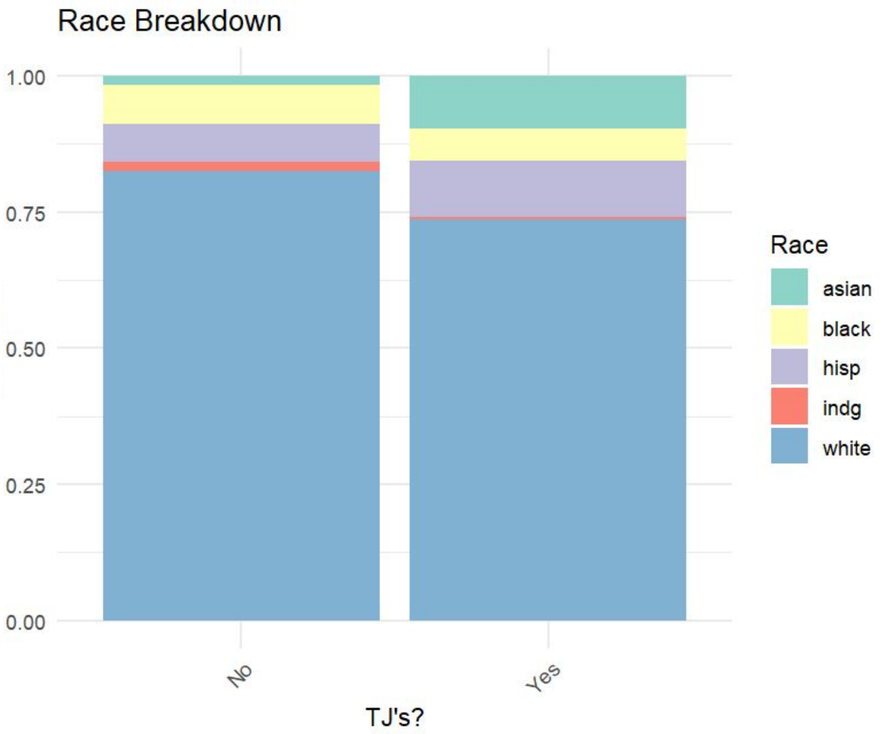
- rent price/ft^2 overall 4-year increase

Predictor Correlation Matrix

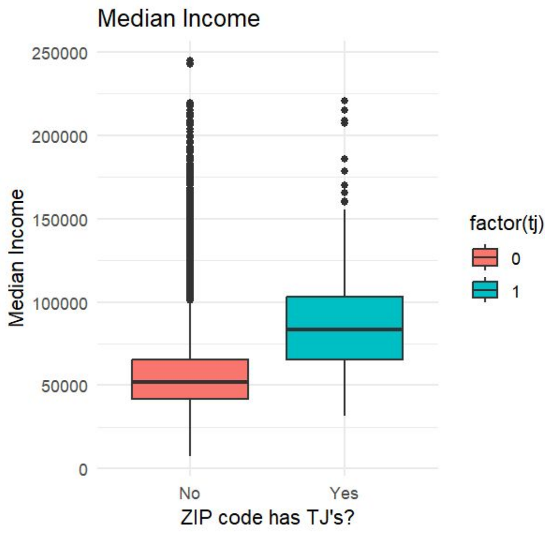


Preliminary Data Analysis

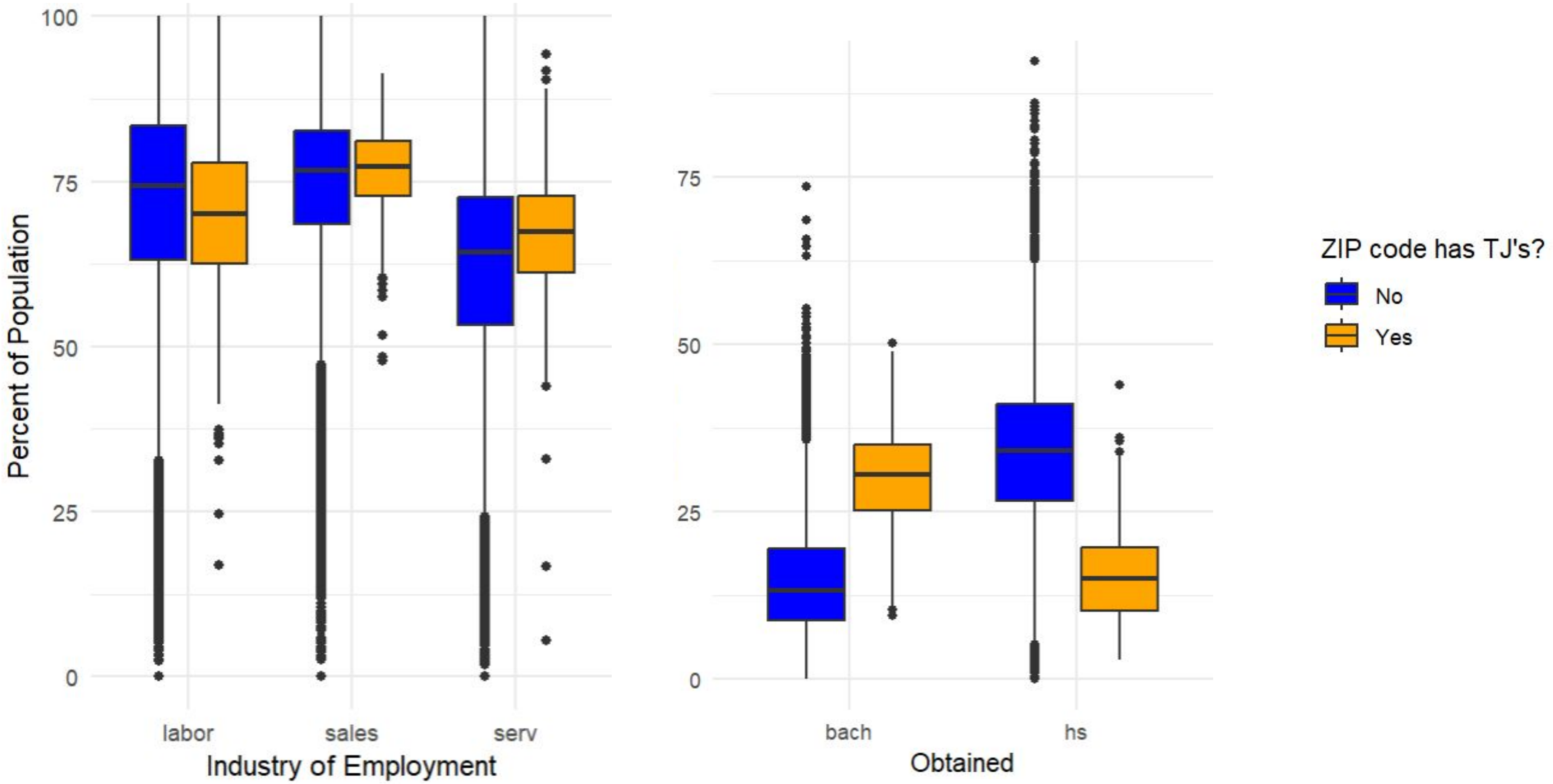




Preliminary Data Analysis



Preliminary Data Analysis



Models

- Simple linear model
  - All 21 predictors
  - Best subset with 5 predictors
- Decision tree
  - 12 predictors
- Neural network
  - All predictors

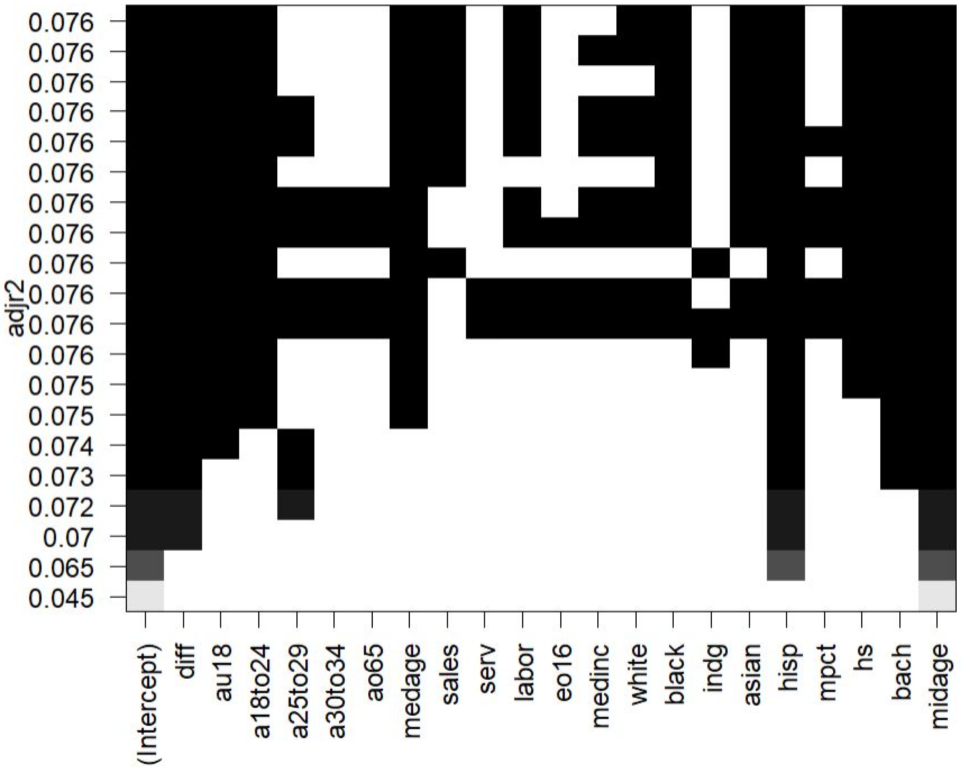
Best Subset Selection

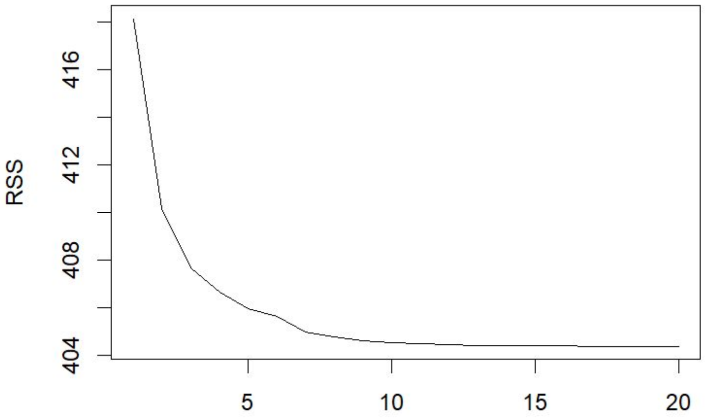
```
1 subsets of each size up to 20
Selection Algorithm: exhaustive
          diff au18 a18to24 a25to29 a30to34 ao65 medage midage sales serv labor eo16 medinc white black indg asian hisp mpct hs  bach
1  ( 1 )  " "  " "  " "     " "     " "     " "  " "    " "    " "   " "  " "   " "  " "    " "   " "   " "  " "   " "  " "  " " "*" 
2  ( 1 )  " "  " "  " "     " "     " "     " "  " "    " "    " "   " "  " "   " "  " "    " "   " "   " "  "*"   " "  " "  " " "*" 
3  ( 1 )  "*"  " "  " "     " "     " "     " "  " "    " "    " "   " "  " "   " "  " "    " "   " "   " "  "*"   " "  " "  " " "*" 
4  ( 1 )  "*"  " "  " "     "*"     " "     " "  " "    " "    " "   " "  " "   " "  " "    " "   " "   " "  "*"   " "  " "  " " "*" 
5  ( 1 )  "*"  " "  " "     "*"     " "     " "  " "    " "    " "   " "  " "   " "  " "    " "   " "   " "  "*"   " "  " "  "*" "*" 
6  ( 1 )  "*"  "*"  " "     "*"     " "     " "  " "    " "    " "   " "  " "   " "  " "    " "   " "   " "  "*"   " "  " "  "*" "*" 
7  ( 1 )  "*"  "*"  "*"     " "     " "     " "  "*"    " "    " "   " "  " "   " "  " "    " "   " "   " "  "*"   " "  " "  "*" "*" 
8  ( 1 )  "*"  "*"  "*"     " "     " "     " "  "*"    " "    " "   " "  " "   " "  " "    " "   " "   " "  "*"   " "  "*"  "*" "*" 
9  ( 1 )  "*"  "*"  "*"     " "     " "     " "  "*"    " "    " "   " "  " "   " "  " "    " "   "*"   " "  "*"   " "  "*"  "*" "*" 
10  ( 1 ) "*"  "*"  "*"     " "     " "     " "  "*"    "*"    " "   " "  " "   " "  " "    " "   "*"   " "  "*"   " "  "*"  "*" "*" 
11  ( 1 ) "*"  "*"  "*"     " "     " "     " "  "*"    "*"    " "   " "  " "   " "  "*"    " "   "*"   " "  "*"   " "  "*"  "*" "*" 
12  ( 1 ) "*"  "*"  "*"     " "     " "     " "  "*"    "*"    " "   " "  " "   "*"  " "    "*"   " "   "*"  "*"   " "  "*"  "*" "*" 
13  ( 1 ) "*"  "*"  "*"     " "     " "     " "  "*"    "*"    " "   " "  " "   "*"  "*"    "*"   " "   "*"  "*"   " "  "*"  "*" "*" 
14  ( 1 ) "*"  "*"  "*"     " "     " "     " "  "*"    "*"    " "   "*"  " "   "*"  "*"    "*"   " "   "*"  "*"   " "  "*"  "*" "*" 
15  ( 1 ) "*"  "*"  "*"     " "     " "     " "  "*"    "*"    " "   "*"  " "   "*"  "*"    " "   "*"   "*"  "*"   "*"  "*"  "*" "*" 
16  ( 1 ) "*"  "*"  "*"     " "     " "     " "  "*"    "*"    " "   "*"  " "   "*"  "*"    "*"   "*"   "*"  "*"   "*"  "*"  "*" "*" 
17  ( 1 ) "*"  "*"  "*"     "*"     " "     " "  "*"    "*"    " "   "*"  " "   "*"  "*"    "*"   "*"   "*"  "*"   "*"  "*"  "*" "*" 
18  ( 1 ) "*"  "*"  "*"     "*"     " "     " "  "*"    "*"    " "   "*"  "*"   "*"  "*"    "*"   "*"   "*"  "*"   "*"  "*"  "*" "*" 
```

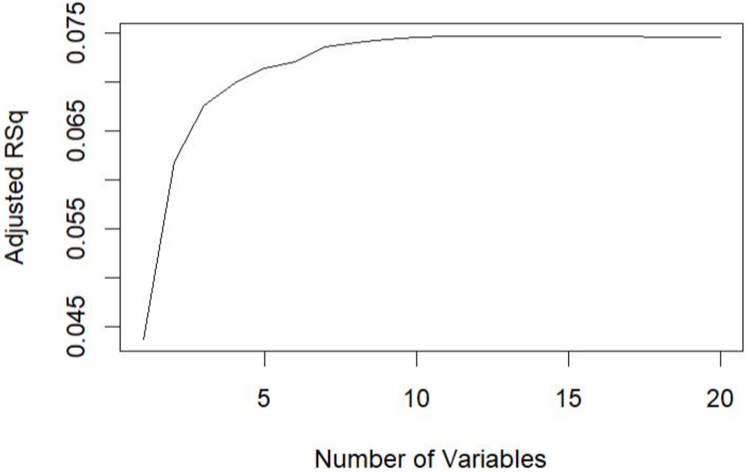
Most useful predictors:

1. % Bachelor's degree
2. % Asian
3. rent/ft^2 increase
4. % Age 25 to 29
5. % High school diploma
6. % Age 18-24
7. Median age
8. Male
9. Black

Best Subset Selection







Logistic Regression

Choose best subset of size 5.

```
Coefficients:
              Estimate Std. Error t value Pr(>|t|)    
(Intercept) -5.266e-02  6.217e-03  -8.471  < 2e-16 ***
diff         9.210e-02  6.070e-03  15.172  < 2e-16 ***
a25to29      2.404e-03  2.677e-04   8.979  < 2e-16 ***
hisp         2.040e-04  5.912e-05   3.451 0.000559 ***
bach         2.509e-03  9.232e-05  27.175  < 2e-16 ***
midage       1.039e-04  1.340e-04   0.775 0.438193    
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

For reference: using all predictors got 4 more wrong

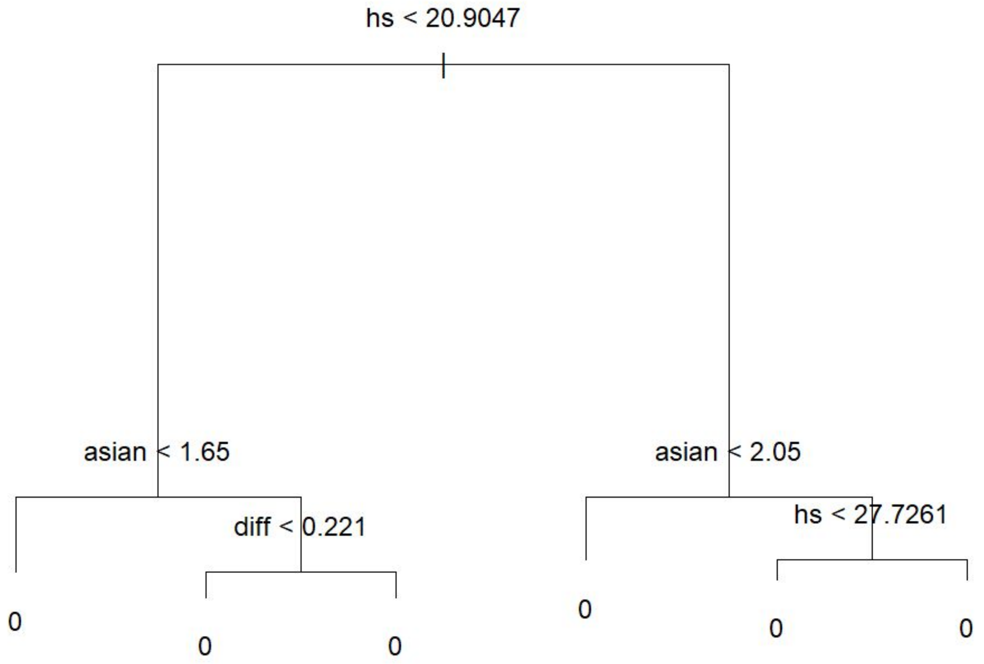
Baseline = 0.9828

```
          Reference
Prediction    0    1
         0 7210  106
         1    8    5

               Accuracy : 0.9844
                 95% CI : (0.9813, 0.9872)
    No Information Rate : 0.9849
    P-Value [Acc > NIR] : 0.6362

                  Kappa : 0.0777

 Mcnemar's Test P-Value : <2e-16

            Sensitivity : 0.99889
            Specificity : 0.04505
         Pos Pred Value : 0.98551
         Neg Pred Value : 0.38462
             Prevalence : 0.98485
         Detection Rate : 0.98376
   Detection Prevalence : 0.99823
      Balanced Accuracy : 0.52197

       'Positive' Class : 0
```

Decision Tree

```
tree = tree(formula, data=df[-test_indices,])
```

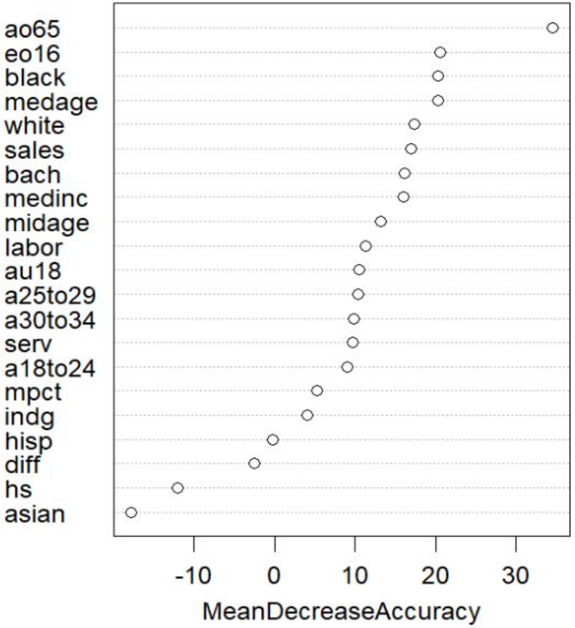
Just predicts all 0’s.
Accuracy = baseline = 0.9828

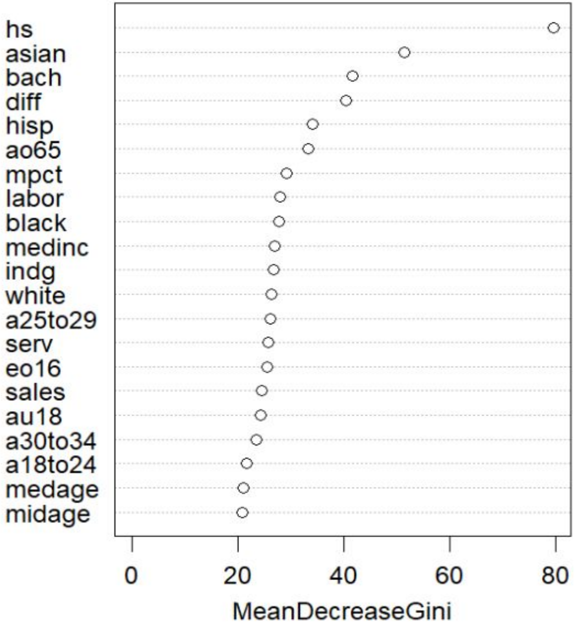


Random Forest

500 decision trees, 12 predictors each

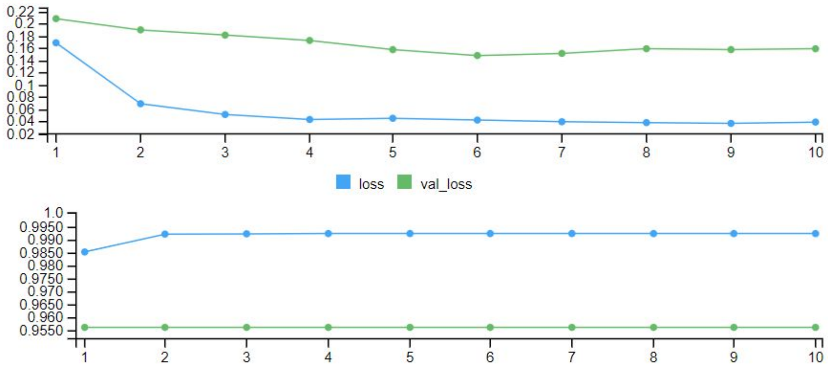
```
          Reference
Prediction    0    1
         0 7217  108
         1    1    3

               Accuracy : 0.9851
                 95% CI : (0.9821, 0.9878)
    No Information Rate : 0.9849
    P-Value [Acc > NIR] : 0.449

                  Kappa : 0.0512

 Mcnemar's Test P-Value : <2e-16

            Sensitivity : 0.99986
            Specificity : 0.02703
         Pos Pred Value : 0.98526
         Neg Pred Value : 0.75000
             Prevalence : 0.98485
         Detection Rate : 0.98472
   Detection Prevalence : 0.99945
      Balanced Accuracy : 0.51344

       'Positive' Class : 0
```





Neural Network

```
input <- layer_input(shape = c(21))
output <- input %>%
  layer_dense(units = 16, activation = "relu") %>%
  layer_dropout(rate = 0.4) %>%
  layer_dense(units = 8, activation = "relu") %>%
  layer_dropout(rate = 0.3) %>%
  layer_dense(units = 8, activation = "relu") %>%
  layer_dropout(rate = 0.3) %>%
  layer_dense(units = 2, activation = "softmax")
```




```
          Reference
Prediction    0    1
         0 7218  111
         1    0    0

Accuracy : 0.9849
```

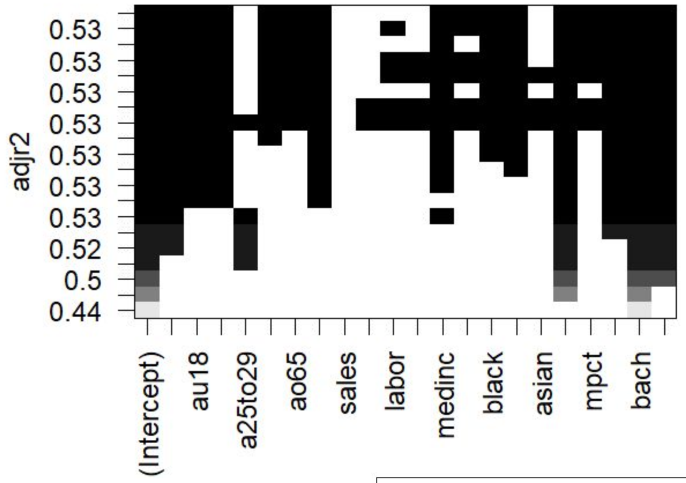
Fixing Class Imbalance

Keep **1000** negative observations and all 499 positive ones

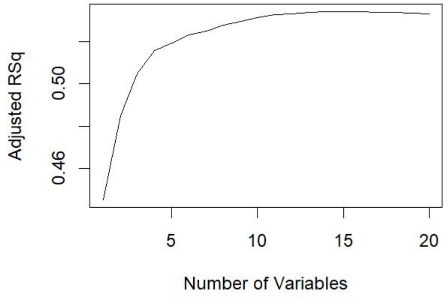
	Positive	Negative
1125 training instances	375	750
374 test	124	250

1499 data points
New baseline: 0.667

Best Subset - Balanced




```
Confusion Matrix and Statistics

          Reference
Prediction   0   1
         0 227  17
         1  23 107

               Accuracy : 0.893
```

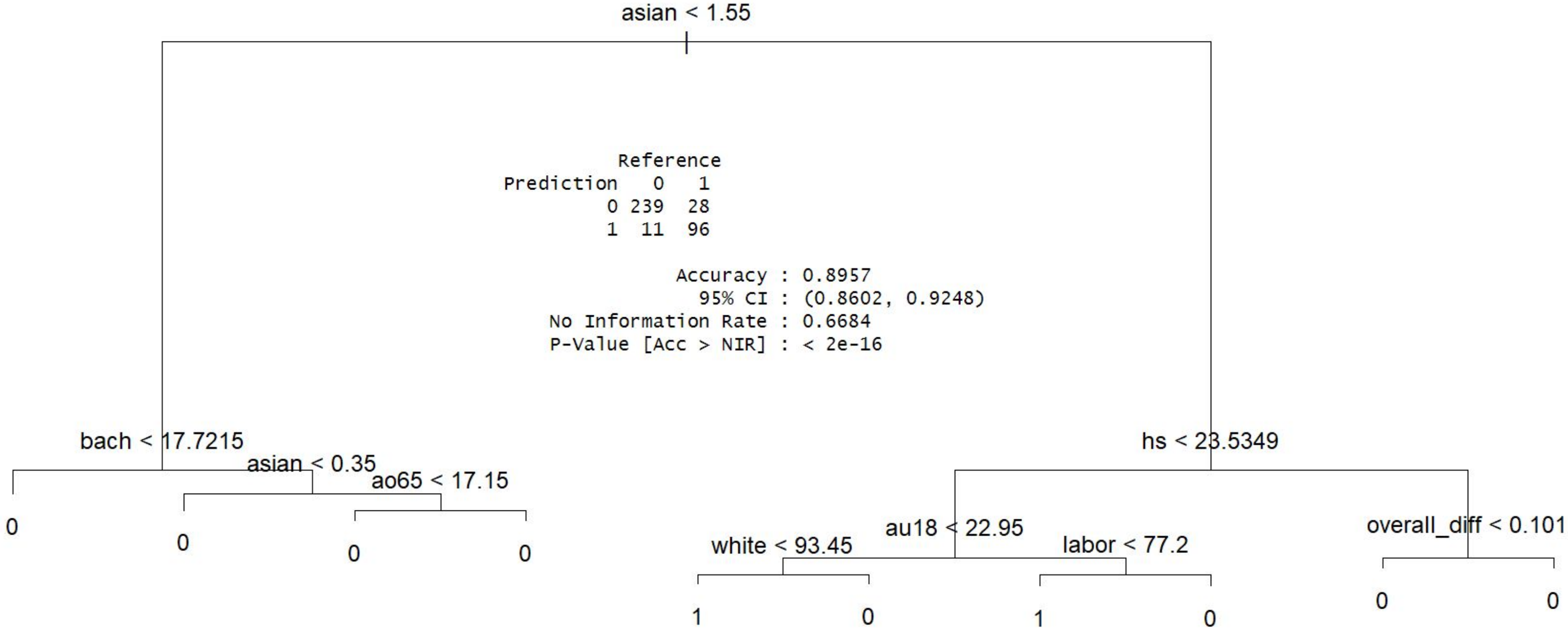



```
Coefficients:
(Intercept)      a25to29         hisp
 -0.6811283    0.1681260    0.0257066

         hs         bach       midage
 -0.1288011    0.0717491    0.0002936
```

Model with all predictors (instead of 5) got one more correct

Decision Tree

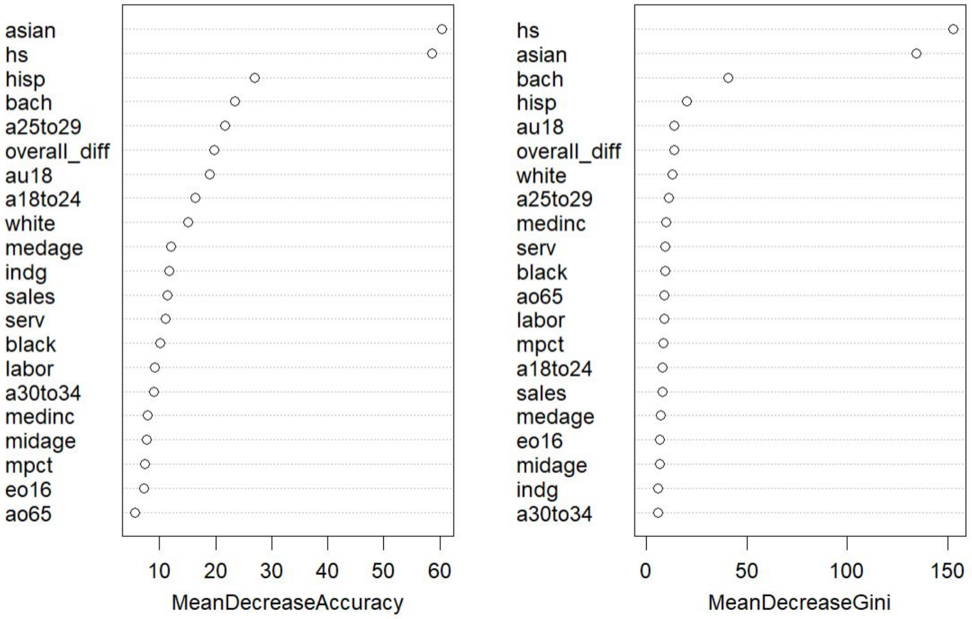
```
Classification tree:
tree(formula = formula, data = df[train_indices, ])
Variables actually used in tree construction:
[1] "hs"             "asian"         "overall_diff"
Number of terminal nodes:  5
```




```
          Reference
Prediction   0   1
         0 239  28
         1  11  96

               Accuracy : 0.8957
                 95% CI : (0.8602, 0.9248)
    No Information Rate : 0.6684
    P-Value [Acc > NIR] : < 2e-16
```

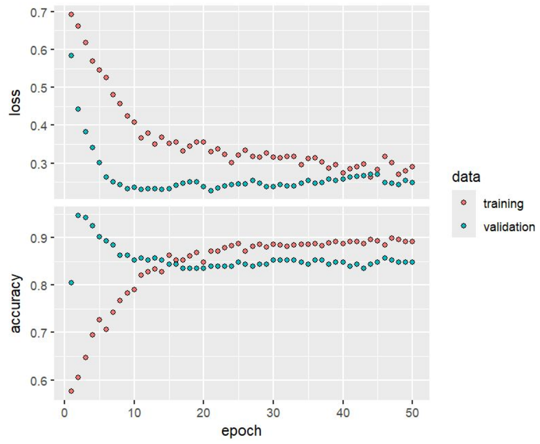
Random Forest

```
          Reference
Prediction   0   1
         0 233  18
         1  17 106

               Accuracy : 0.9064
                 95% CI : (0.8723, 0.9339)
    No Information Rate : 0.6684
    P-Value [Acc > NIR] : <2e-16
```



Neural Network

```
input <- layer_input(shape = c(21))
output <- input %>%
  layer_dense(units = 16, activation = "relu") %>%
  layer_dropout(rate = 0.4) %>%
  layer_dense(units = 8, activation = "relu") %>%
  layer_dropout(rate = 0.3) %>%
  layer_dense(units = 8, activation = "relu") %>%
  layer_dropout(rate = 0.3) %>%
  layer_dense(units = 2, activation = "softmax")
```



Same structure, more epochs

```
            Reference
Prediction    0   1
         0  219   5
         1   31 119

Accuracy : 0.9037
```

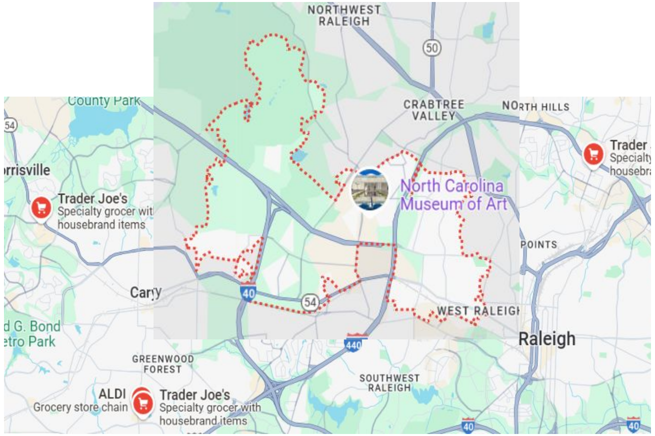
Discussion

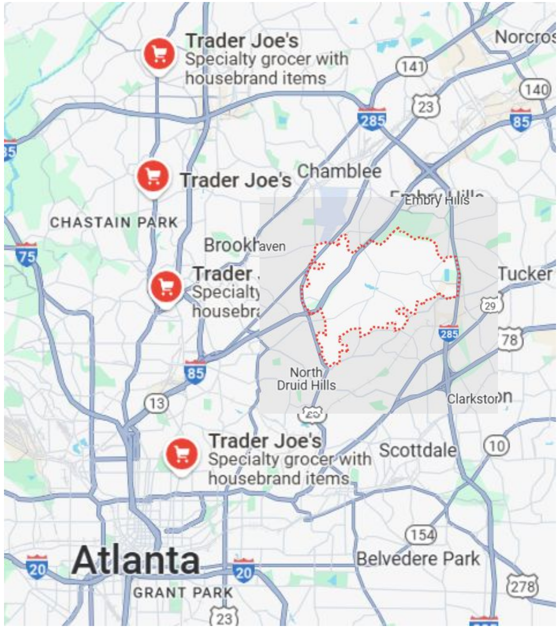
- Improved performance with smaller, better-balanced datasets:
    - Including more observations can skew training data and reduce effectiveness.
    - Balanced datasets allow simpler models (logistic regression, decision tree) to perform as well as complex models (random forest, neural network).
- Predicting Trader Joe’s locations is not a difficult problem:
    - Simple models suffice under Occam’s razor and for better interpretability.
- Best predictors identified:
    - Higher income and education levels.
    - Recent home price increases.
    - Higher Asian population.

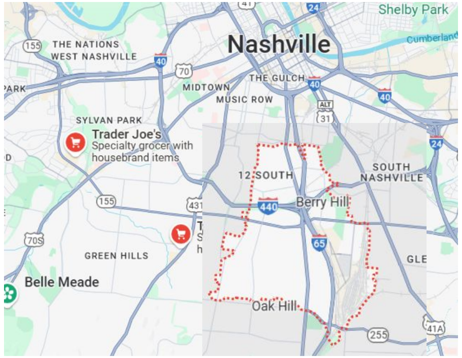
Note: Since the models were given many age-related variables and many more race variables, it's almost trivial to observe that age or race emerge as a predictor—it reflects the limited scope of the input features rather than any deeper insight.

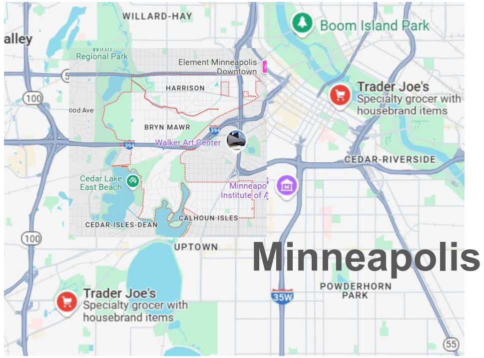
New Trader Joe's Locations?

zip	city	overall_diff	medage	medinc	asian	white	black	tj
<chr>	<chr>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<fct>
37204	Nashville	0.256	30.1	76250	2	81.8	14	0
27607	Raleigh	0.1	24	74786	4.9	81.9	9.7	0
55405	Minneapolis	0.198	31.9	56365	3.8	78.4	11.8	0
79119	Amarillo	0.096	36.8	87565	2.2	93.4	1.1	0
19301	Paoli	0.148	44.4	93211	6	89.3	4.2	0
30345	Atlanta	0.416	34.1	71914	6.2	64.1	25.3	0









New Locations?

zip	city	diff	a25to29	hisp	bach	midage
<dbl>	<chr>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>
94123	San Francisco	1.10	18.5	6.2	52.5	35.8
98243	Deer Harbor	0.29	2.3	3.2	55.4	36.3
90049	Los Angeles	1.06	9.2	4.7	40.5	38.7
10006	New York	0.366	21.6	11.1	51.3	31.4
94129	San Francisco	1.10	12.3	8.2	43.1	36.7