Introduction to Data Science

Hugo Cunha - 1610565
Pedro Felipe - 1611074
Renan Moreira - 1611061
Yan Cunha - 1610886

Tema escolhido (T1-1)

PROMOTE GENDER EQUALITY AND EMPOWER WOMEN

Dataset escolhido

Datasets de Gender Inequality Index e Gender Development Index foram escolhidos pois, combinados, apresentavam os principais parâmetros necessários para medir a desigualdade presente entre os gêneros em todo o mundo. Os dados neles contidos pertencem a diversos campos do desenvolvimento humano.

http://hdr.undp.org/en/composite/GII

http://hdr.undp.org/en/composite/GDI

Perguntas realizadas

- Qual a relação entre o HDI e GII?
- Qual a razão da renda per capita média feminina para a renda per capita média masculina?
- Qual a relação entre IDH de um país e participação feminina na força de trabalho?
- Qual a relação entre renda per capta por gênero e anos de educação por gênero?
- Existe relação entre representação política feminina e igualdade de renda entre os gêneros?
- Existe relação entre participação feminina na força de trabalho e IDH?
- Qual a razão média entre IDH feminino e masculino?
- Qual a relação entre a razão da participação na força de trabalho feminina e masculina e igualdade de renda entre os gêneros?
- Como as diferentes regiões do mundo se comportam em relação ao GII?
- Existe relação entre a participação política feminina e a expectativa de vida feminina?

Arrumando o Dataset

Dataset utilizado possuía, originalmente, diversas colunas em branco e eram dois arquivos excel separados.

Juntamos em um único arquivo .csv eliminando as colunas em branco

Problemas encontrados

Um dos atributos estavam com uma notação utilizando ‘ . ’

Ex:

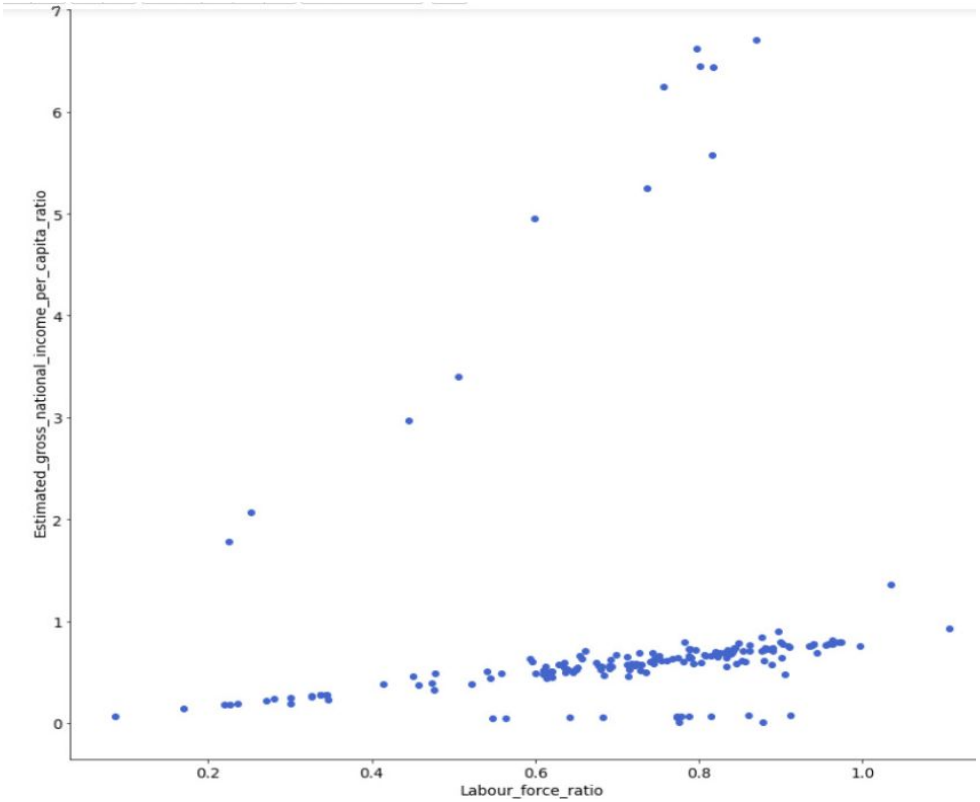
1.000 (mil)

100.000 (cem mil)

Isso era lido como não mil e cem mil e sim como um (1.0) e cem (100.0)

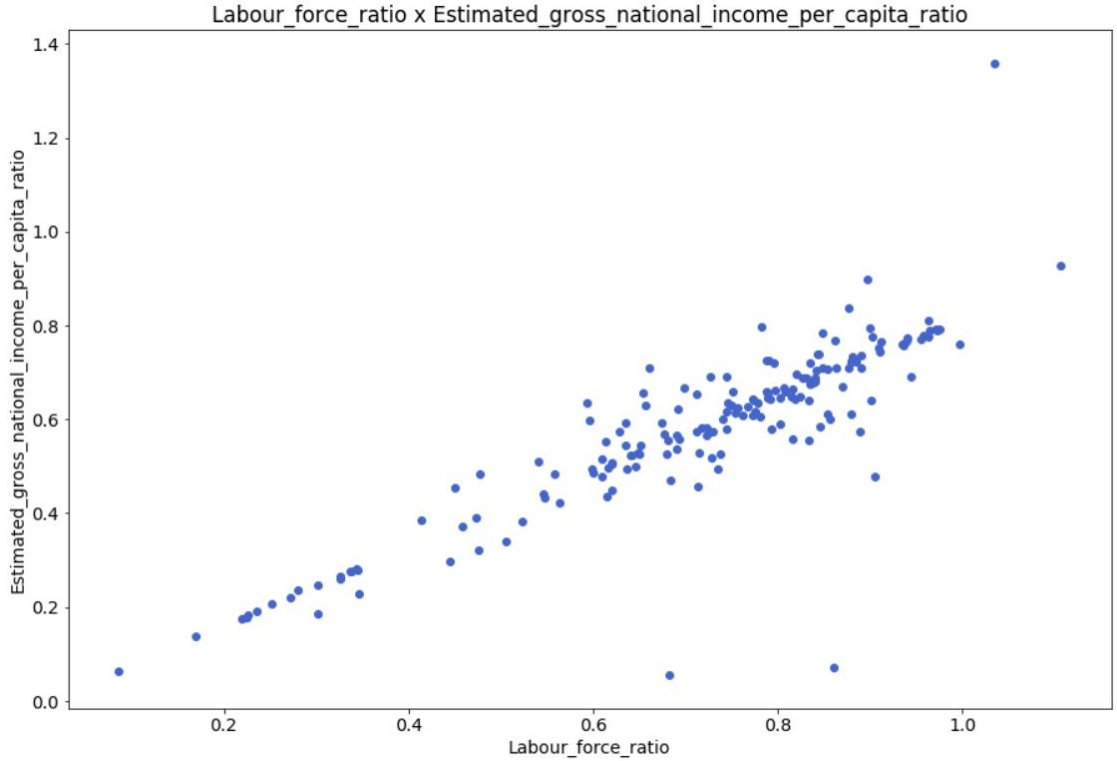
Problemas encontrados

Detecção:



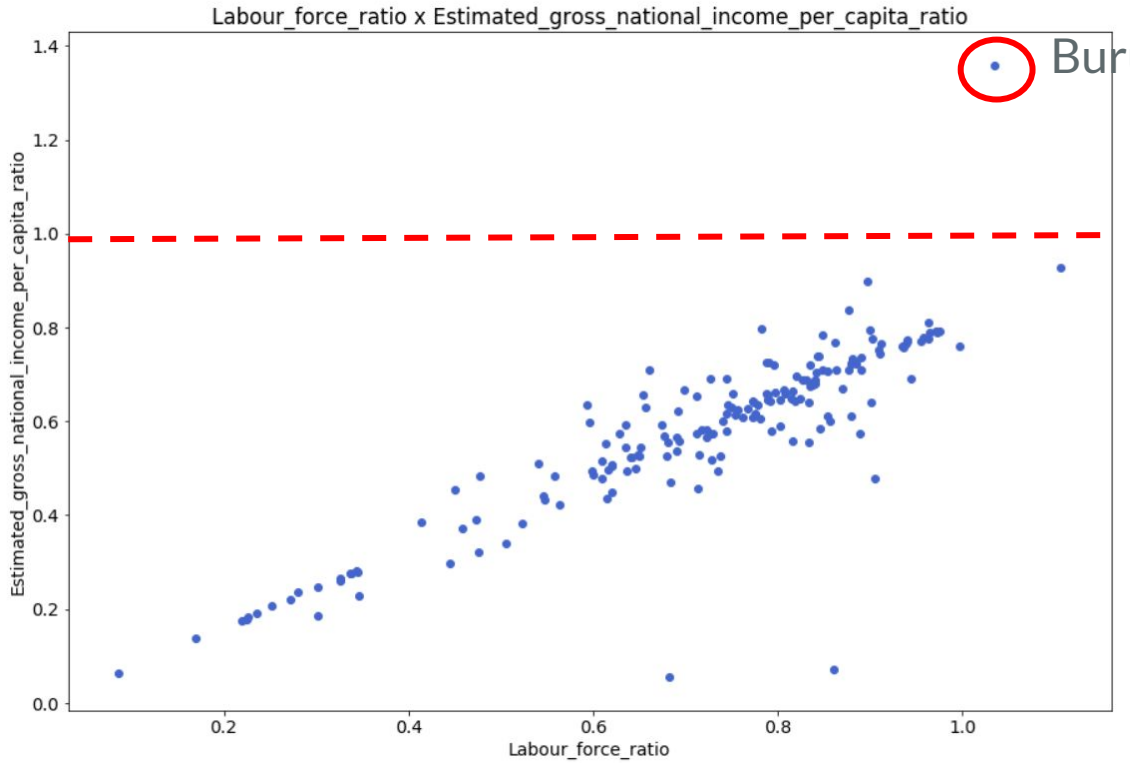
Problemas encontrados

Detecção:



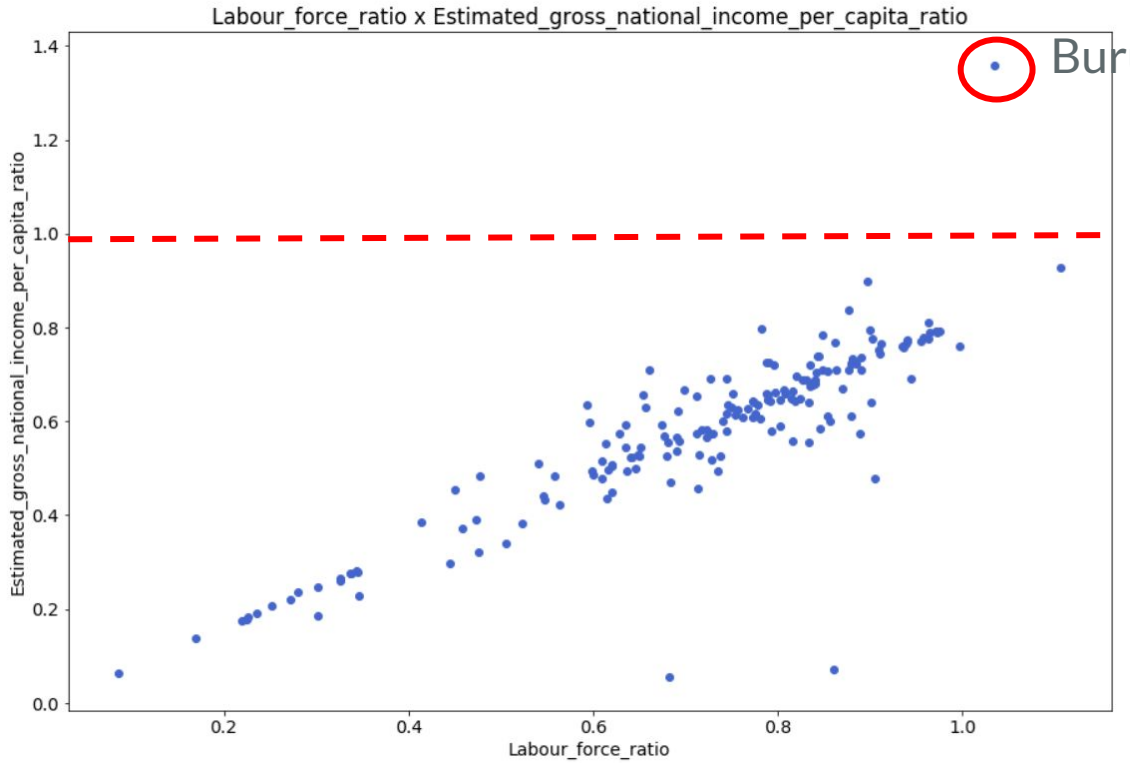
Problemas encontrados

Detecção:



Problemas encontrados

Detecção:



RperC_f = 807
RperC_m = 594
dolares por ano

Tratamento em código

Primeiro problema

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 189 entries, 0 to 188
Data columns (total 23 columns):
HDI_rank                                        189 non-null int64
Country                                         189 non-null object
GDI_Value                                       189 non-null object
GDI_Group                                       189 non-null object
HDI_f                                           189 non-null object
HDI_m                                           189 non-null object
life_expectancy_f                               189 non-null object
life_expectancy_m                               189 non-null object
Expected_years_of_schooling_f                   189 non-null object
Expected_years_of_schooling_m                   189 non-null object
Mean_years_of_schooling_f                       189 non-null object
Mean_years_of_schooling_m                       189 non-null object
Estimated_gross_national_income_per_capita_f    189 non-null object
Estimated_gross_national_income_per_capita_m    189 non-null object
GII_value                                       189 non-null object
GII_rank                                        189 non-null object
maternal_mortality_ratio                        189 non-null object
adolescent_birth_rate                           189 non-null object
share_of_seats_parliament                       189 non-null object
population_above_secondary_education_f          189 non-null object
population_above_secondary_education_m          189 non-null object
labour_force_participation_rate_f               189 non-null object
labour_force_participation_rate_m               189 non-null object
dtypes: int64(1), object(22)
memory usage: 34.0+ KB
```

Tratamento em código

Solução:

```
def str_to_float(c):
    if isinstance(c, float):
        return c
    else:
        try:
            return float(c)
        except:
            return None


# aplicamos a função a todas as colunas numéricas
df.HDI_rank = df.HDI_rank.apply(str_to_int)
df.GDI_Value = df.GDI_Value.apply(str_to_float)
df.GDI_Group = df.GDI_Group.apply(str_to_int)
df.HDI_f = df.HDI_f.apply(str_to_float)
df.HDI_m = df.HDI_m.apply(str_to_float)
df.life_expectancy_f = df.life_expectancy_f.apply(str_to_float)
df.life_expectancy_m = df.life_expectancy_m.apply(str_to_float)
df.Expected_years_of_schooling_f = df.Expected_years_of_schooling_f.apply(str_to_float)
df.Expected_years_of_schooling_m = df.Expected_years_of_schooling_m.apply(str_to_float)
df.Mean_years_of_schooling_f = df.Mean_years_of_schooling_f.apply(str_to_float)
df.Mean_years_of_schooling_m = df.Mean_years_of_schooling_m.apply(str_to_float)
df.Estimated_gross_national_income_per_capita_f = df.Estimated_gross_national_income_per_capita_f.apply(str_to_float)
df.Estimated_gross_national_income_per_capita_m = df.Estimated_gross_national_income_per_capita_m.apply(str_to_float)
df.GII_value = df.GII_value.apply(str_to_float)
df.GII_rank = df.GII_rank.apply(str_to_int)
df.maternal_mortality_ratio = df.maternal_mortality_ratio.apply(str_to_float)
df.adolescent_birth_rate = df.adolescent_birth_rate.apply(str_to_float)
df.share_of_seats_parliament = df.share_of_seats_parliament.apply(str_to_float)
df.population_above_secondary_education_f = df.population_above_secondary_education_f.apply(str_to_float)
df.population_above_secondary_education_m = df.population_above_secondary_education_m.apply(str_to_float)
df.labour_force_participation_rate_f = df.labour_force_participation_rate_f.apply(str_to_float)
df.labour_force_participation_rate_m = df.labour_force_participation_rate_m.apply(str_to_float)
```

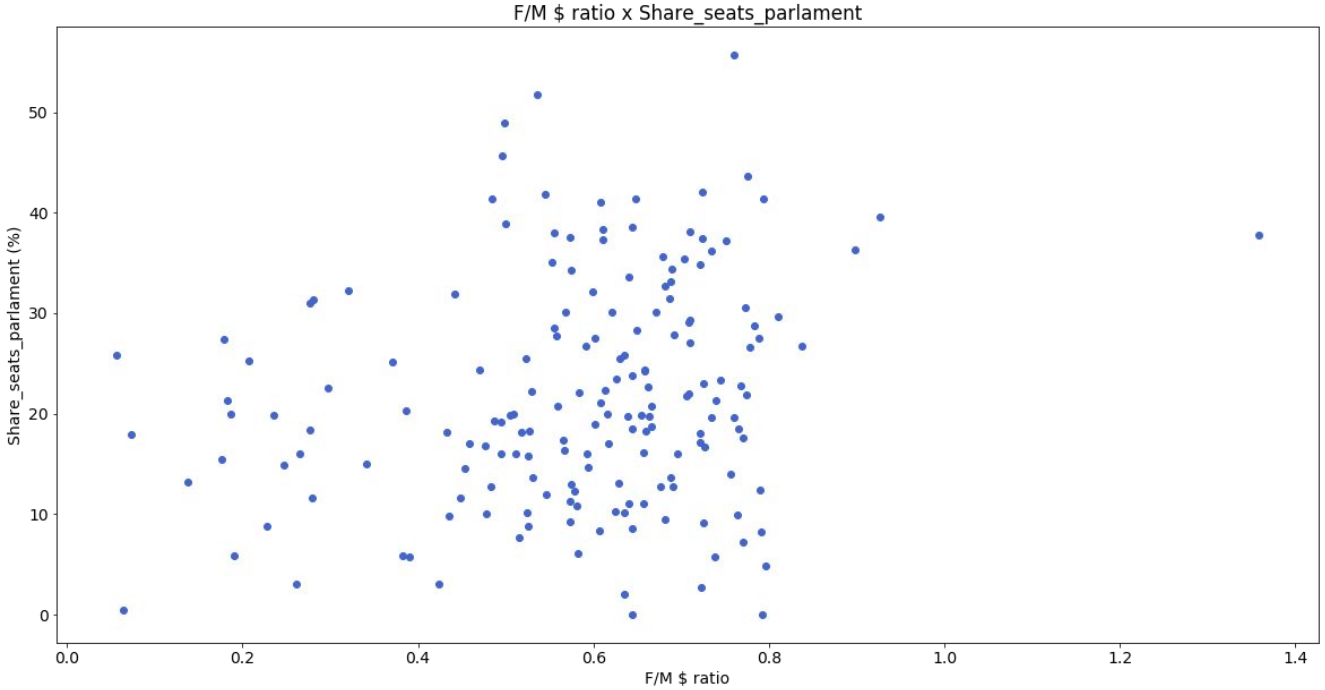
Tratamento em código

Solução:

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 189 entries, 0 to 188
Data columns (total 23 columns):
HDI_rank                                    189 non-null int64
Country                                     189 non-null object
GDI_Value                                   164 non-null float64
GDI_Group                                   164 non-null float64
HDI_f                                       164 non-null float64
HDI_m                                       164 non-null float64
life_expectancy_f                           183 non-null float64
life_expectancy_m                           183 non-null float64
Expected_years_of_schooling_f               179 non-null float64
Expected_years_of_schooling_m               179 non-null float64
Mean_years_of_schooling_f                   171 non-null float64
Mean_years_of_schooling_m                   171 non-null float64
Estimated_gross_national_income_per_capita_f    178 non-null float64
Estimated_gross_national_income_per_capita_m    178 non-null float64
GII_value                                   160 non-null float64
GII_rank                                    160 non-null float64
maternal_mortality_ratio                    180 non-null float64
adolescent_birth_rate                       183 non-null float64
share_of_seats_parliament                   187 non-null float64
population_above_secondary_education_f      165 non-null float64
population_above_secondary_education_m      165 non-null float64
labour_force_participation_rate_f           178 non-null float64
labour_force_participation_rate_m           178 non-null float64
dtypes: float64(21), int64(1), object(1)
memory usage: 34.0+ KB
```

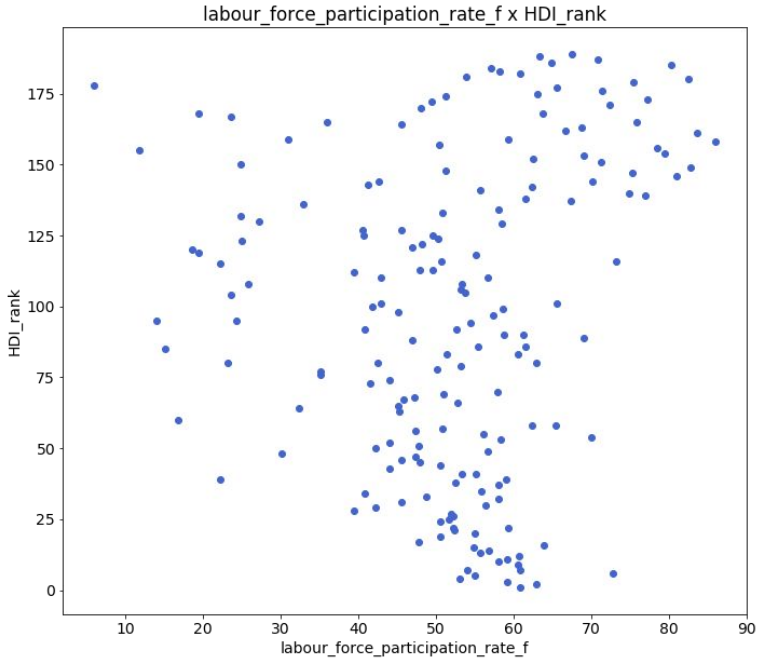
Perguntas interessantes respondidas

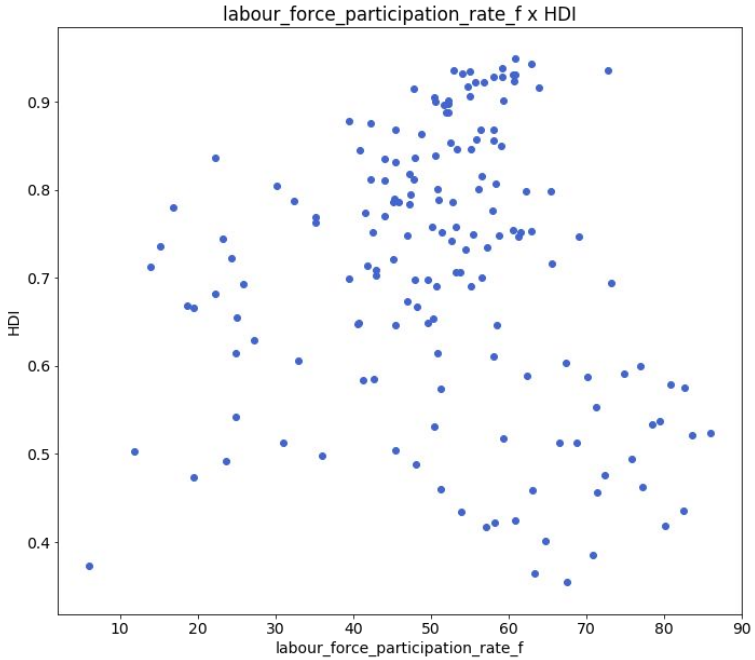
Existe relação entre representação política feminina e igualdade de renda entre os gêneros?



Perguntas interessantes respondidas

Existe relação entre participação feminina na força de trabalho e IDH?





T1-2 Aprendizado supervisionado

Nesta parte do trabalho, utilizamos algoritmos de aprendizado de máquina aprendidos em sala de aula, como o Support Vector Machine (SVM) e o K-Nearest Neighbors (KNN). Escolhemos instâncias encontradas no DataSet “Car Evaluation” do repositório da UCI. Esse DataSet possui 6 atributos para classificação de carros : Preço de compra, preço de manutenção, número de portas, número de pessoas, tamanho da mala e segurança. Todos os atributos têm valores discretos. Existem 4 classificações para os carros: Não aceitável, aceitável, bom e muito bom. No arquivo de DataSet existem 1728 instâncias e esse grupo de instâncias cobre todo o espaço de possibilidades de combinações de atributos.

https://pedro-magalhaes.github.io/Teste/t1_2/machine_learning.html

Atributos escolhidos para o treinamento

Inicialmente fizemos a avaliação de quanto de informação cada atributo provê para a classificação. Utilizamos o Weka para ver mais facilmente esse ranking de atributos.

Verificamos que o atributo número de portas contribuía muito pouco, no entanto, ao retirarmos esse atributos , os resultados foram piores. Assim, optamos por manter todos os atributos no nosso treinamento.

KNN

No algoritmo de KNN, alteramos o nosso K para ver qual opção dava o melhor resultado. Ao final, chegamos a conclusão de que o K = 7 dava o melhor resultado. Abaixo, os testes com cada K testado:

k 3 = 0.9050925925925926

k 5 = 0.9444444444444444

k 7 = 0.9606481481481481

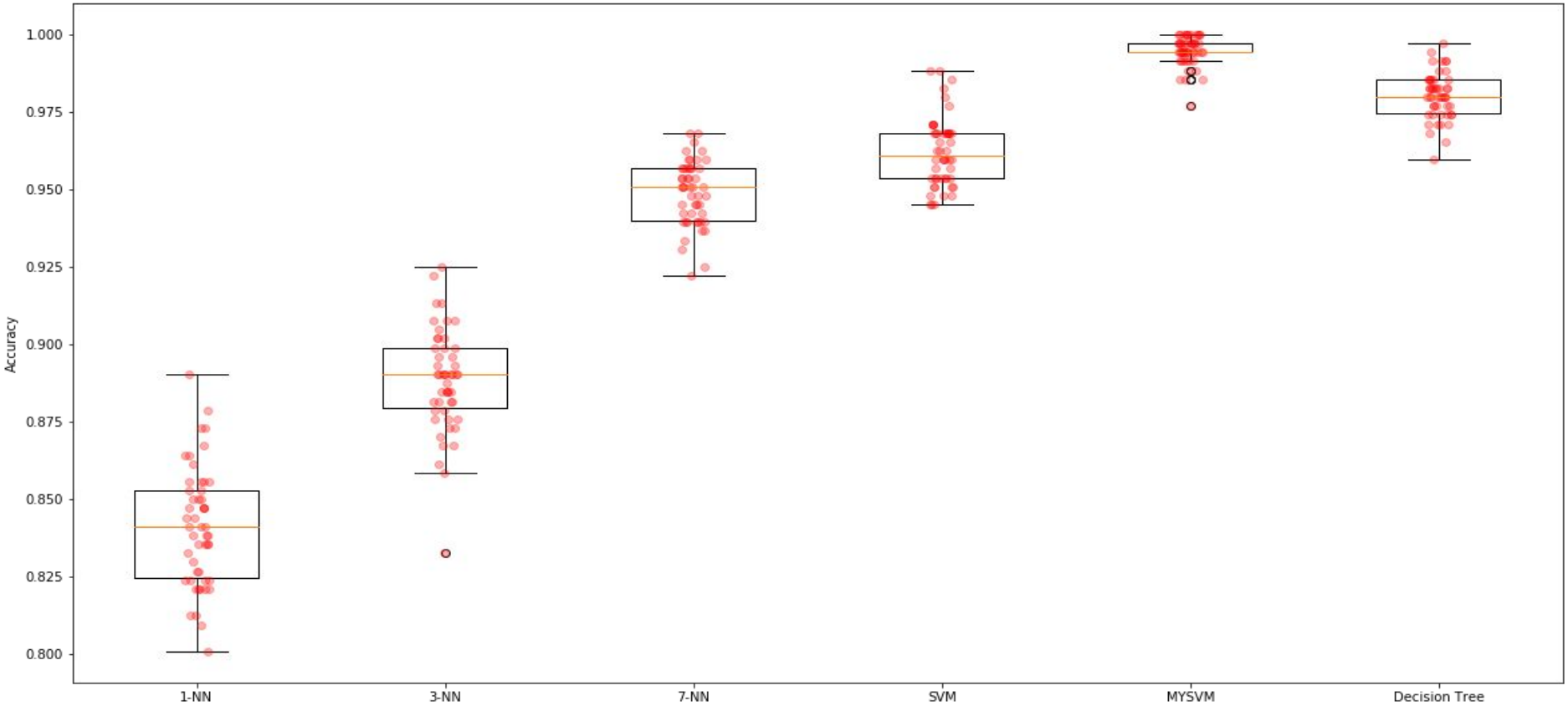
k 9 = 0.9467592592592593

k 11 = 0.9259259259259259

SVM

No SVM, tínhamos varias opções de kernel e o que deu os melhores resultados foi o kernel gaussiano (rbf), combinado com o “C” e o “gamma”. O kernel é a função que projeta o espaço de entrada no espaço de características. Nos nossos testes variando os parâmetros, usamos o Kernel linear primeiro, depois o polinomial e por último o gaussiano. Os melhores resultados foram obtidos com o gaussiano. Dentre todos os algoritmos testados, o SVM com o kernel rbf foi o que trouxe os melhores resultados. Um passo que melhorou nossos resultados foi a estandardização dos valores dos atributos em um número de 0 a 1

Resultados


Accuracy
1.000
0.975
0.950
0.925
0.900
0.875
0.850
0.825
0.800
1-NN
3-NN
7-NN
SVM
MYSVM
Decision Tree

T1-3 dataset utilizado:

	Year	State_Name	High_School_Graduation	Income_Inequality_Gini_Index	Violent_Crime
0	2015	Alabama	80.0	0.476	430.8
1	2015	Alaska	71.8	0.418	640.4
2	2015	Arizona	75.1	0.468	416.5
3	2015	Arkansas	84.9	0.471	460.3
4	2015	California	80.4	0.489	402.1
5	2015	Colorado	76.9	0.458	308.0
6	2015	Connecticut	85.5	0.501	262.5
7	2015	Delaware	80.4	0.449	491.4
8	2015	Florida	75.6	0.483	470.4
9	2015	Georgia	71.7	0.480	365.7

Dados:

- Ano (2015-2018)
- Estado dos EUA (50 estados)
- Percentual de pessoas que se graduaram no ensino médio
- Desigualdade econômica medida com o Gini index
- Número de crimes violentos por 100,000 pessoas

https://www.americashealthrankings.org

Análise de regressão linear

Predictors: variáveis usadas para a previsão.

Coefficients: o ‘peso’ que cada variável têm na equação da previsão.

Intercept: valor esperado de Y quando todas as variáveis são iguais a 0

Mean squared error: diferença média quadrática entre as variáveis e o que é estimado

Análise de regressão linear

Caso 1: Previsão baseada somente no ano

Predictors: ['Year']

Coefficients: [8.23804237]

Intercept: -16243.189012251141

Mean squared error: 15528.31

Análise de regressão linear

Caso 2: Previsão baseada no ano e educação

Predictors: ['Year', 'High_School_Graduation']

Coefficients: [15.77325404 -2.48217076]

Intercept: -31246.9428323701

Mean squared error: 17793.42

Análise de regressão linear

Caso 3: Previsão baseada no ano e pobreza

Predictors: ['Year', 'Income_Inequality_Gini_Index']

Coefficients: [1.57617042 1560.10282786]

Intercept: -3537.5920359750817

Mean squared error: 21890.04

Análise de regressão linear

Caso 4: Previsão baseada no ano, pobreza e educação

Predictors: ['High_School_Graduation', 'Income_Inequality_Gini_Index', 'Year']

Coefficients: [-2.484413 1395.62526004 12.84994129]

Intercept: -25982.52163079244

Mean squared error: 13259.21

Análise de regressão linear

Caso 5: Previsão baseada no ano, pobreza e educação usando regressão de Lasso

Predictors: ['High_School_Graduation', 'Income_Inequality_Gini_Index', 'Year']

Coefficients: [0. 847.73302618 0.]

Intercept: -22.743162697750563

Mean squared error: 13227.44

Conclusão

Comparando o Mean Squared Error:

Caso 1: 15528.31

Caso 2: 17793.42

Caso 3: 21890.04

Caso 4: 13259.21

Caso 5: 13227.44

Conclusão

Comparando o Mean Squared Error:

Caso 1: 15528.31

Caso 2: 17793.42

Caso 3: 21890.04

Caso 4: 13259.21

Caso 5: 13227.44