Open in Colab

Custo de seguro saúde

Contexto

Prever corretamente os custos para um paciente em um seguro de saúde é uma forma de realizar cobranças mais justas e mesmo de prever os futuros negócios da empresa.

Conteúdo

O seguinte dicionário de dados pode ser usado para melhor entendimento dos atributos.

Variável	Definição
age	Idade do beneficiário principal
sex	Sexo do contratante de seguro feminino, masculino
bmi	Índice de massa corporal
children	Número de filhos cobertos pelo seguro de saúde / Número de dependentes
smoker	Se o contratante é fumante
region	A área residencial do beneficiário nos EUA, nordeste, sudeste, sudoeste, noroeste
charges	Custos médicos individuais cobrados pelo seguro de saúde

In [ ]:
```
import pandas as pd
import numpy  as np
import seaborn as sea
import matplotlib.pyplot as plt
from matplotlib import animation
from mpl_toolkits.mplot3d import Axes3D
```

In [ ]:
```
data = pd.read_csv(r'dados/insurance.csv',sep=',')
data.head(4)
```

Out[ ]:

	age	sex	bmi	children	smoker	region	charges
0	19	female	27.900	0	yes	southwest	16884.92400
1	18	male	33.770	1	no	southeast	1725.55230
2	28	male	33.000	3	no	southeast	4449.46200
3	33	male	22.705	0	no	northwest	21984.47061

In [ ]:
```
from sklearn.preprocessing import LabelBinarizer,LabelEncoder
```

In [ ]:
```
dummies = pd.get_dummies(data['region'],prefix='region',dtype=int)
data    = pd.concat([
    data.drop(columns='region'),
    dummies
],axis=1)
```

In [ ]:
```
data['sex']    = LabelBinarizer().fit_transform(data['sex'])
data['smoker'] = LabelBinarizer().fit_transform(data['smoker'])
#data['region'] = LabelEncoder().fit_transform(data['region'])
```

In [ ]:
```
data.head(3)
```

Out[ ]:

	age	sex	bmi	children	smoker	charges	region_northeast	region_northwest	region_southeast	region_southwest
0	19	0	27.90	0	1	16884.9240	0	0	0	1
1	18	1	33.77	1	0	1725.5523	0	0	1	0
2	28	1	33.00	3	0	4449.4620	0	0	1	0

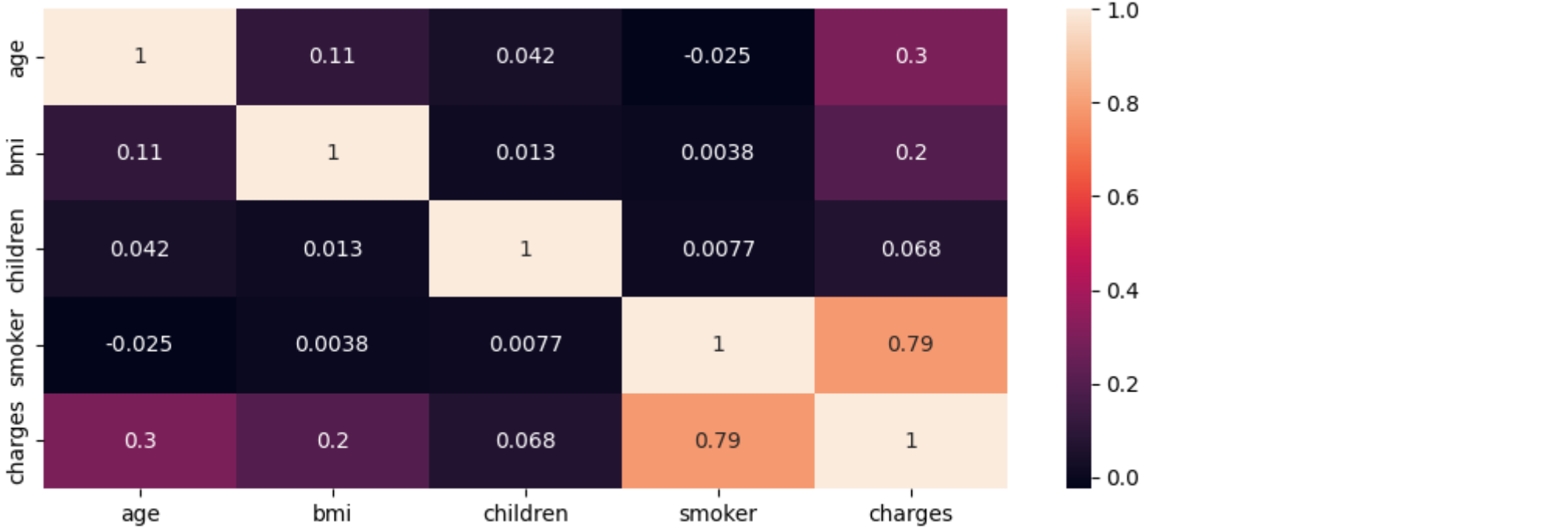
In [ ]:
```
data.describe()
```

Out[ ]:

	age	sex	bmi	children	smoker	charges	region_northeast	region_northwest	region_southeast	region_
count	1338.000000	1338.000000	1338.000000	1338.000000	1338.000000	1338.000000	1338.000000	1338.000000	1338.000000	1
mean	39.207025	0.505232	30.663397	1.094918	0.204783	13270.422265	0.242152	0.242900	0.272048	
std	14.049960	0.500160	6.098187	1.205493	0.403694	12110.011237	0.428546	0.428995	0.445181	
min	18.000000	0.000000	15.960000	0.000000	0.000000	1121.873900	0.000000	0.000000	0.000000	
25%	27.000000	0.000000	26.296250	0.000000	0.000000	4740.287150	0.000000	0.000000	0.000000	
50%	39.000000	1.000000	30.400000	1.000000	0.000000	9382.033000	0.000000	0.000000	0.000000	
75%	51.000000	1.000000	34.693750	2.000000	0.000000	16639.912515	0.000000	0.000000	1.000000	
max	64.000000	1.000000	53.130000	5.000000	1.000000	63770.428010	1.000000	1.000000	1.000000	

In [ ]:
```
plt.figure(figsize=(10,4))
sea.heatmap(data.drop(columns=['sex','region_northeast','region_northwest','region_southeast','region_southwest']).corr(),annot
```

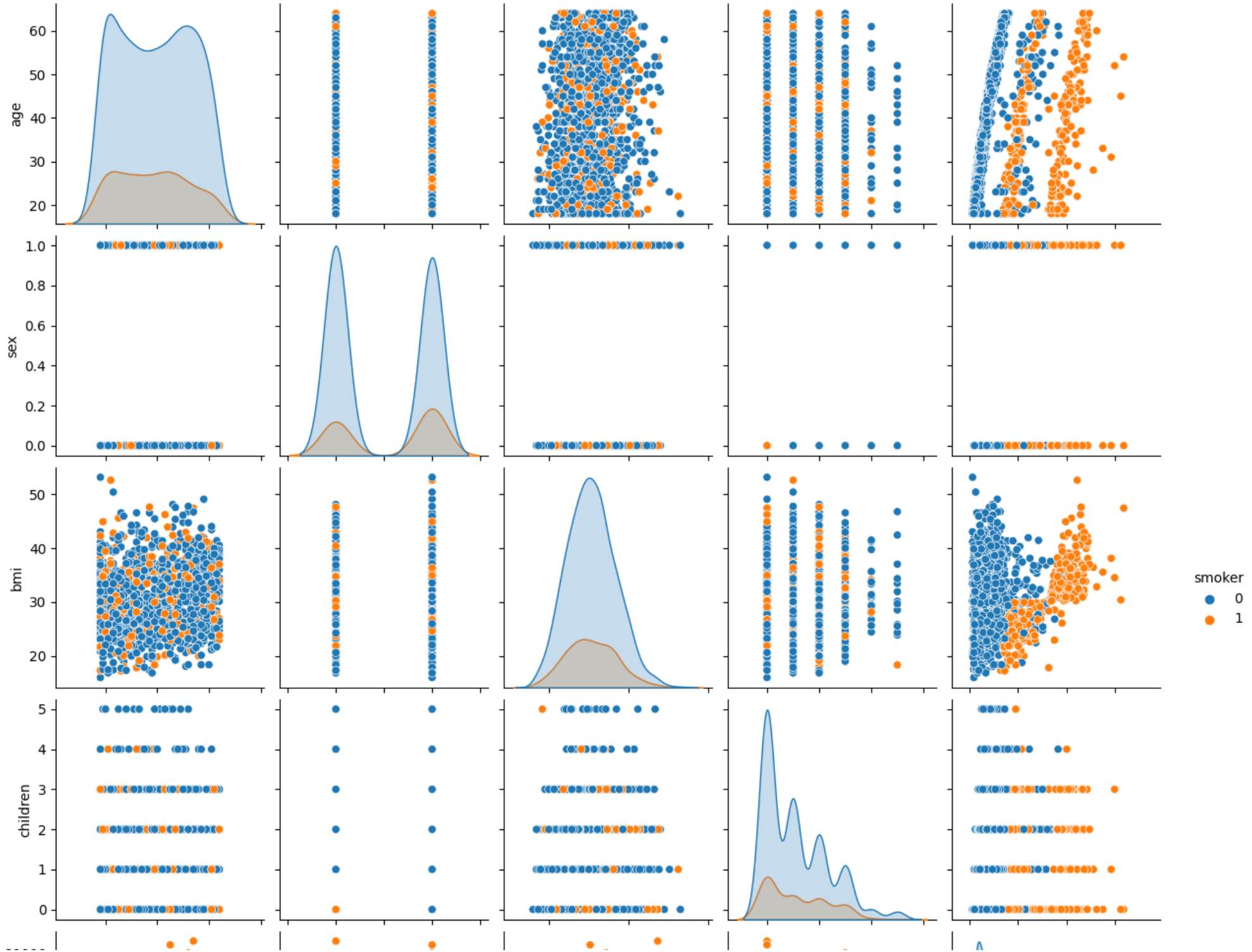
Out[ ]: <Axes: >

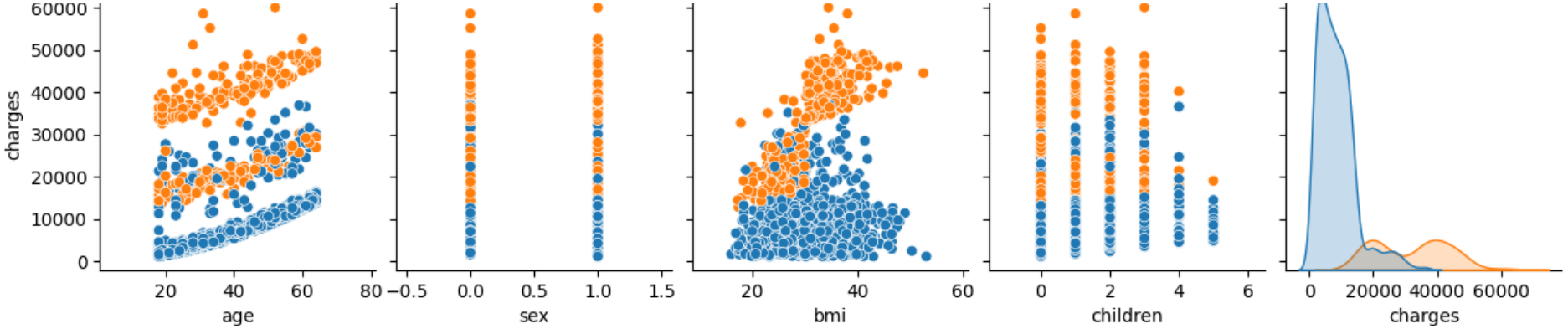


In [ ]:
```
plt.figure(figsize=(40,25))
sea.pairplot(data.drop(columns=data.columns.to_list()[6:]),hue='smoker')
```

Out[ ]: <seaborn.axisgrid.PairGrid at 0x7f5f49b5f8d0>

<Figure size 4000x2500 with 0 Axes>


age
sex
bmi
children
smoker
0
1



In [ ]:
```
from sklearn.model_selection import train_test_split
from sklearn.linear_model    import LinearRegression
from sklearn.metrics         import accuracy_score,r2_score,recall_score,roc_auc_score
```

In [ ]:
```
Y = data['charges'].values
X = data.drop(columns=['charges','sex','children']).values
```

In [ ]:
```
x_train , x_test , y_train , y_test = train_test_split(X,Y,test_size=0.3)
```

In [ ]:
```
regressao =  LinearRegression()
fits   = regressao.fit(x_train,y_train)
y_pred = fits.predict(x_test)
```

In [ ]:
```
fits.coef_.tolist(),fits.intercept_
```

Out[ ]:
```
([244.75902115583563,
  335.53988025347536,
  24292.334368188305,
  622.3389889566472,
  424.84878829858536,
  -525.4015857028794,
  -521.7861915523426],
 -11462.920773909129)
```

In [ ]:
```
for i in range(np.size(y_pred)):
    print(y_test[i],' <=>',y_pred[i],end='\n')
    if (i==10):break
```

```
27000.98473  <=> 13742.821584824138
43578.9394  <=> 37660.94925715869
1725.5523  <=> 3748.5217773528966
10269.46  <=> 11793.058183212655
2689.4954  <=> 7635.578075198664
42124.5153  <=> 34898.01941099474
27533.9129  <=> 36257.720044815025
2902.9065  <=> 4156.399699828524
34779.615  <=> 28634.836637508655
4349.462  <=> 5937.746281116077
11987.1682  <=> 12841.5149469467
```

In [ ]:
```
r2_score(y_test,y_pred)
```

Out[ ]:
```
0.7640857127574718
```

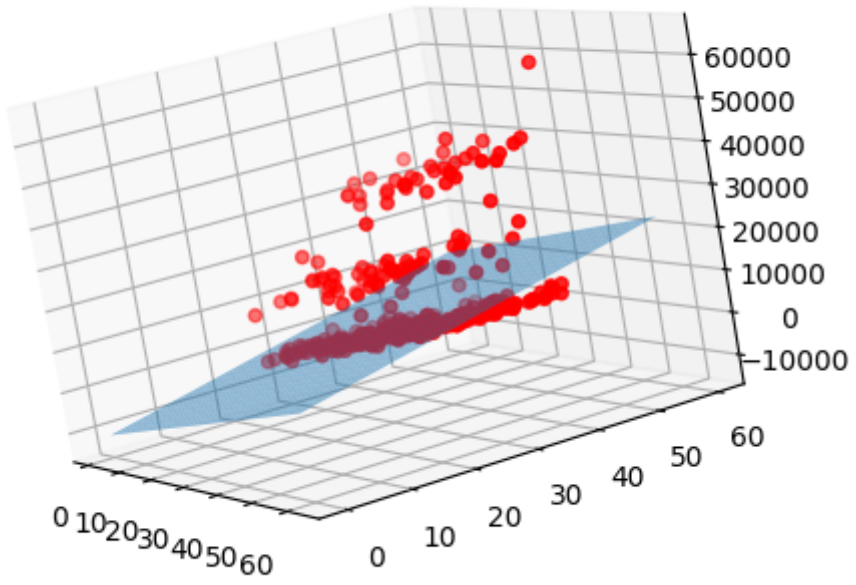
In [ ]:
```
coefs = regressao.coef_
intercept = regressao.intercept_

xs = np.tile(np.arange(61), (61,1))
ys = np.tile(np.arange(61), (61,1)).T
zs = xs*coefs[0]+ys*coefs[1]+intercept

fig = plt.figure(figsize=(10, 5))
ax = fig.add_subplot(projection='3d')

x = x_test[:,0]
y = x_test[:,1]
z = y_test
z_pred = y_pred

ax.scatter(x,y,z,c='r')
ax.plot_surface(xs,ys,zs,alpha=0.5)
ax.view_init(10, -30,10)
```


60000
50000
40000
30000
20000
10000
0
−10000
0 10 20 30 40 50 60
0 10 20 30 40 50 60