

# RELATÓRIO CDD

## Vetor de características e Distribuição do conjunto de dados

Bruno Farias - GRR20186715

Vetor de características: ['Time', 'V1', 'V2', 'V3', 'V4', 'V5', 'V6', 'V7', 'V8', 'V9', 'V10', 'V11', 'V12', 'V13', 'V14', 'V15', 'V16', 'V17', 'V18', 'V19', 'V20', 'V21', 'V22', 'V23', 'V24', 'V25', 'V26', 'V27', 'V28', 'Amount', 'Class']

É um problema binário: fraude ou não fraude

Exemplos de amostras:

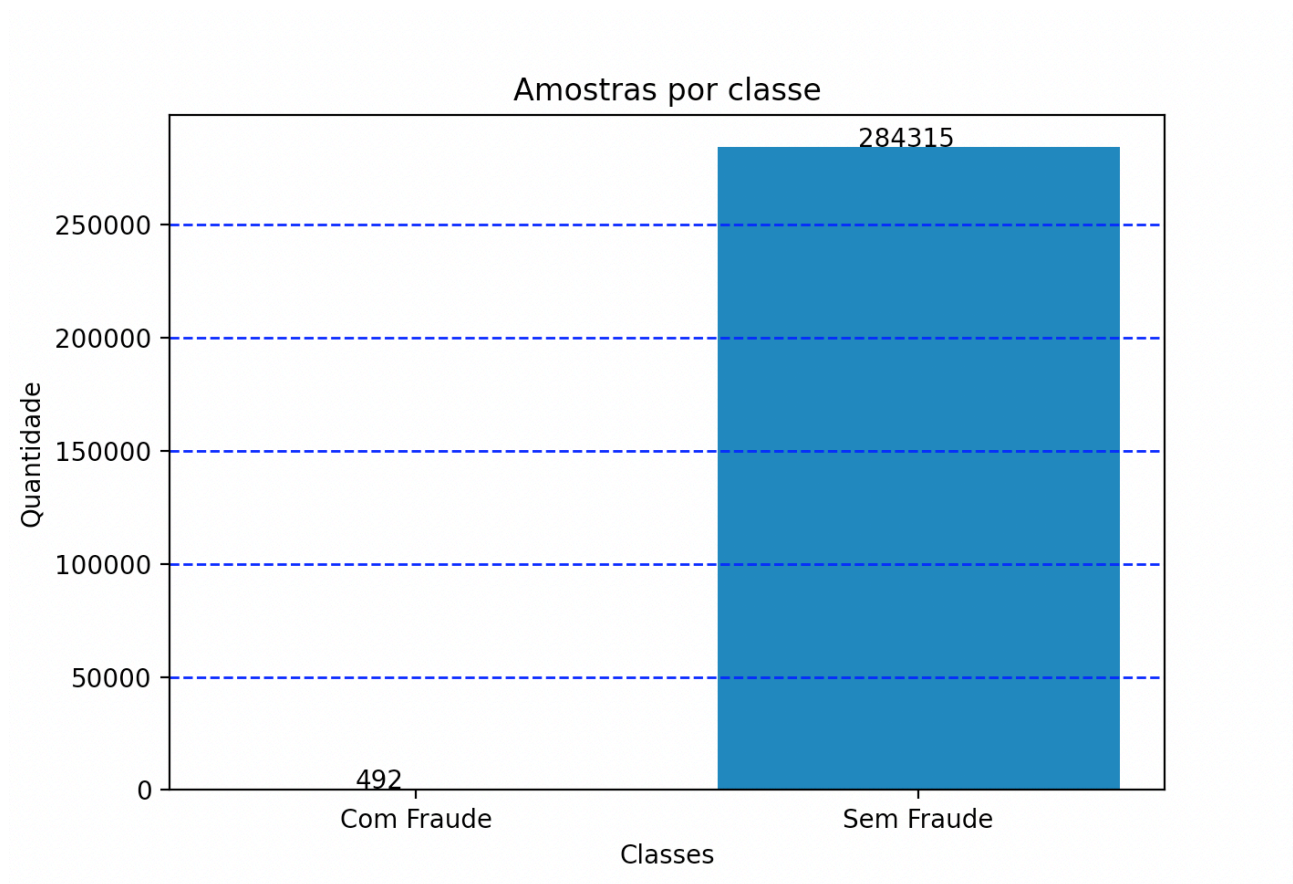
Sem fraude:

```
['172786', '-11.8811178854323', '10.0717849710003', '-9.83478345739033',  
'-2.06665568459413', '-5.36447278097585', '-2.6068373309456',  
'-4.91821543115252', '7.305334020798', '1.9144282734458',  
'4.35617041320691', '-1.59310526246153', '2.71194079105717',  
'-0.68925560917964', '4.62694202525016', '-0.92445871482528',  
'1.10764060095394', '1.99169110706243', '0.510632329130648',  
'-0.682919680356916', '1.4758291346555', '0.213454108437359',  
'0.111863735978609', '1.01447989719391', '-0.509348453168509',  
'1.4368069070214', '0.250034279569581', '0.943651171507532',  
'0.823730961486502', '0.77', '0']  
['172787', '-0.732788670658956', '-0.0550804899173326',  
'2.03502974528243', '-0.73858858438744', '0.868229398914682',  
'1.05841527222565', '0.0243296959175797', '0.294868698501783',  
'0.584800017281683', '-0.975926063322508', '-0.150188847103276',  
'0.915801914403555', '1.21475584849424', '-0.675142955809576',  
'1.16493090944588', '-0.711757349978881', '-0.025692855268572',  
'-1.22117885840624', '-1.54555608554509', '0.0596158998872689',  
'0.214205341747019', '0.924383584903381', '0.0124630383316793',  
'-1.01622566867336', '-0.606623985854728', '-0.395255065710324',  
'0.0684724700405593', '-0.0535273892010011', '24.79', '0']
```

Com fraude:

```
['406', '-2.3122265423263', '1.95199201064158', '-1.60985073229769',  
'3.9979055875468', '-0.522187864667764', '-1.42654531920595',  
'-2.53738730624579', '1.39165724829804', '-2.77008927719433',  
'-2.77227214465915', '3.20203320709635', '-2.89990738849473',  
'-0.595221881324605', '-4.28925378244217', '0.389724120274487',  
'-1.14074717980657', '-2.83005567450437', '-0.0168224681808257',  
'0.416955705037907', '0.126910559061474', '0.517232370861764',  
'-0.0350493686052974', '-0.465211076182388', '0.320198198514526',  
'0.0445191674731724', '0.177839798284401', '0.261145002567677',  
'-0.143275874698919', '0', '1']
```

Gráfico de barras para amostras por classe:



492 amostras com fraude e 284315 amostras "sem fraude"

Código para obtenção do vetor e geração de gráfico:

```

try:
    #abre arquivo
    with open(nome_arquivo, 'r') as arquivo:
        try:
            #le as linhas do csv
            reader = csv.reader(arquivo)

            quantidadeDeLinhas = sum(1 for linha in reader)
            quantidadeDeLinhas -= 1 #retira cabeçalho = quantidade
de amostras

            print('Quantidade de amostras: %d' %
(quantidadeDeLinhas))

            #volta pro início do arquivo
            arquivo.seek(0,0)

            print('Cabecalho:')
            rowCabecalho = next(reader)
            print(rowCabecalho)

            contFraudes = 0
            for linha in reader:
                print(linha)
                if linha[30] == '1':
                    #somatório para total de fraudes
                    contFraudes += 1

            #calcula quantidade de amostras sem fraudes
            qtdAmostrasSemFraude = quantidadeDeLinhas - contFraudes

            print('Quantidade de Amostras COM Fraudes: %d' %
(contFraudes))
            print('Quantidade de Amostras SEM Fraudes: %d' %
(qtdAmostrasSemFraude))

            #gera gráfico de barras
            nomes = ['Com Fraude', 'Sem Fraude']
            valores = [contFraudes, qtdAmostrasSemFraude]

            bars = plt.bar(nomes, valores)
            for bar in bars:
                yval = bar.get_height()
                plt.text(bar.get_x() + 0.28, yval + .005, yval)
            plt.rcParams.update({'figure.figsize':(10,8),
'figure.dpi':100})
            plt.title('Amostras por classe')
            plt.xlabel('Classes')
            plt.ylabel('Quantidade')
            plt.grid(color='b', linestyle='--', linewidth=1,
axis='y')

            plt.show()

        except csv.Error as e:
            sys.exit('arquivo %s, linha %d: %s' %
(nome_arquivo, reader.line_num, e))

```

```
except FileNotFoundError as e:  
    sys.exit('Arquivo \'%s\' não encontrado!' % (nome_arquivo))
```