

Tópicos Especiais em Sistemas de Informação

**Pré-processamento: tratando valores
inconsistentes e outliers**

Ely – elydasilvamiranda@gmail.com

Datasets necessários

- Titanic;
- Credit-data.

Pré-processamento

- Os datasets quase sempre possuem inconsistências prejudicam os algoritmos de ML;
- Antes de tudo, precisam ser analisados e pré-processados. Algumas tarefas bem comuns:
 - Limpar os dados;
 - Tratar valores inconsistentes;
 - Normalizar valores;
 - Tratar valores que sejam fora do padrão;
 - Combinar ou calcular colunas;
 - Etc.

Conhecendo os dados

- Olhar pro dataset apenas como um conjunto de registros é muito limitado;
- Em um dataset grande o suficiente, muitos valores inconsistentes passam despercebidos;
- Algumas abordagens para conhecer melhor os datasets:
 - Descrever o dataset;
 - Verificar valores nulos, negativos...;
 - Filtrar colunas;
 - Exibir gráficos;
 - Etc.

Função head()

- Descreve as “n” primeira linhas:

```
ds_credit.head()
```

	clientid	income	age	loan	default
0	1	66155.925095	59.017015	8106.532131	0
1	2	34415.153966	48.117153	6564.745018	0
2	3	57317.170063	63.108049	8020.953296	0
3	4	42709.534201	45.751972	6103.642260	0
4	5	66952.688845	18.584336	8770.099235	1

```
ds_titanic.head(3)
```

	PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare	Cabin	Embarked
0	1	0	3	Braund, Mr. Owen Harris	male	22.0	1	0	A/5 21171	7.2500	NaN	S
1	2	1	1	Cumings, Mrs. John Bradley (Florence Briggs Th...	female	38.0	1	0	PC 17599	71.2833	C85	C
2	3	1	3	Heikkinen, Miss. Laina	female	26.0	0	0	STON/O2. 3101282	7.9250	NaN	S

Função describe()

- Descreve valores mínimos, máximos e médias.



```
ds_credit.describe()
```

	clientid	income	age	loan	default
count	2000.000000	2000.000000	1997.000000	2000.000000	2000.000000
mean	1000.500000	45331.600018	40.807559	4444.369695	0.141500
std	577.494589	14326.327119	13.624469	3045.410024	0.348624
min	1.000000	20014.489470	-52.423280	1.377630	0.000000
25%	500.750000	32796.459717	28.990415	1939.708847	0.000000
50%	1000.500000	45789.117313	41.317159	3974.719419	0.000000
75%	1500.250000	57791.281668	52.587040	6432.410625	0.000000
max	2000.000000	69995.685578	63.971796	13766.051239	1.000000

Procurar por valores nulos


```
: ds_titanic.isnull().sum()
```

```
: PassengerId      0
   Survived        0
   Pclass          0
   Name            0
   Sex             0
   Age            177
   SibSp           0
   Parch           0
   Ticket          0
   Fare           0
   Cabin          687
   Embarked        2
dtype: int64
```



```
ds_credit.isnull().sum()
```

```
clientid      0
income        0
age           3
loan          0
default       0
dtype: int64
```



Outliers

- Infelizmente, descobrir valores inconsistentes pode não ser uma tarefa simples;
- Os valores que fogem à normalidade são chamados de outliers;
- São valores que podem não aparecer “a olho nu” ou simplesmente não serem de fácil verificação;
- Uma alternativa é o uso de gráficos que ajude a revelar os valores que fogem à normalidade;
- Algumas gráficas são: histogramas, gráficos de dispersão e gráficos.

Histogramas

- Fornecem uma distribuição de dados. No exemplo abaixo, percebem-se idades negativas

```
import matplotlib.pyplot as plt
import numpy

ds_credit['age'].hist()
plt.show()
```

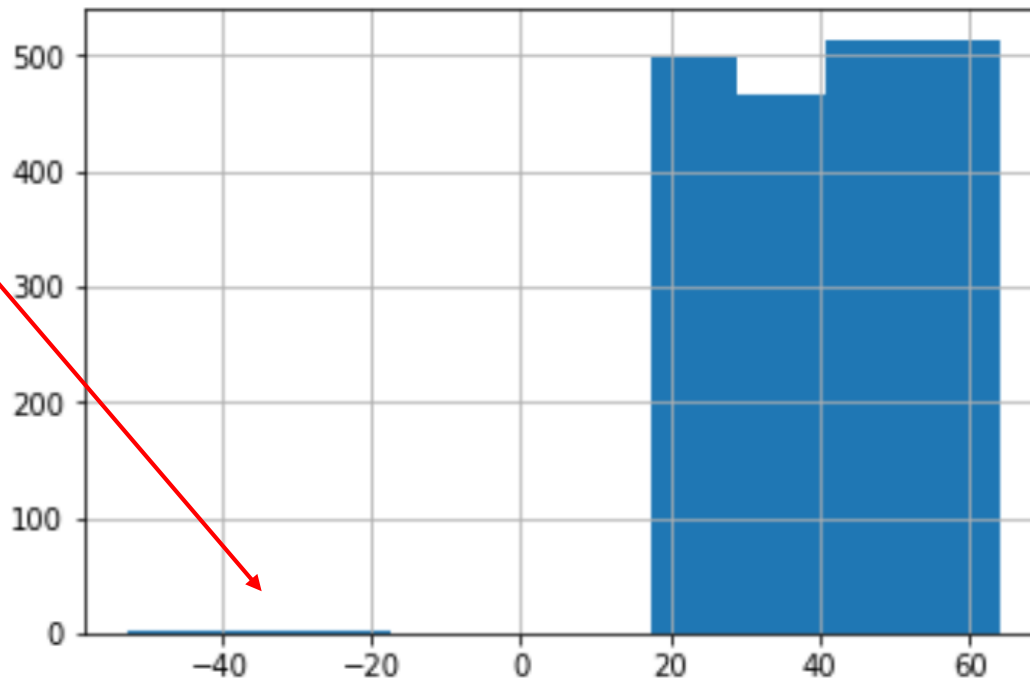
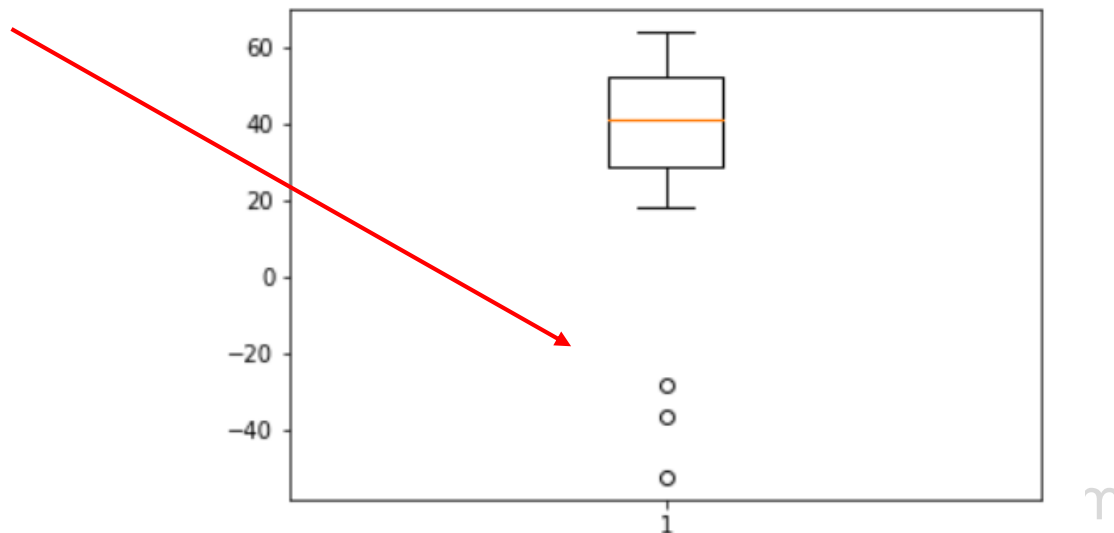


Diagrama de caixa

- Agrupa os dados em caixa evidenciando o primeiro, segundo e terceiro quartis;
- Além disso, exibe os limites inferiores e superiores e os outliers;

```
# outliers idade
import matplotlib.pyplot as plt
ds_credit_age = ds_credit['age'].dropna()
plt.boxplot(ds_credit_age, showfliers = True)
```



Filtragens

- Após verificar outliers, é interessante verificar 'in loco' os registros com problemas;
- Para isso, pode-se usar uma filtragem simples:

```
ds_credit[ds_credit['age'].isnull()]
```

	clientid	income	age	loan	default
28	29	59417.805406	NaN	2082.625938	0
30	31	48528.852796	NaN	6155.784670	0
31	32	23526.302555	NaN	2862.010139	0

```
ds_credit[ds_credit['age'] < 0]
```

	clientid	income	age	loan	default
15	16	50501.726689	-28.218361	3977.287432	0
21	22	32197.620701	-52.423280	4244.057136	0
26	27	63287.038908	-36.496976	9595.286289	0

Tratando dados

- Sobre valores inconsistentes e outliers:
 - Se possível, repassar os registros para averiguação de possíveis fraudes;
 - Não é desejável aplicar algoritmos sem antes tratar esses casos;
 - Algumas alternativas para tratar essas ocorrências antes de aplicar um algoritmo:
 - Corrigir com os valores reais (nem sempre é possível);
 - Remover colunas ou linhas;
 - Preencher ou substituir valores com médias, valores padrão etc.

Remover colunas ou linhas

- Essas opções devem ser cuidadosamente analisadas;
- Remover itens a serem avaliados podem influenciar muito as avaliações;
- Remover algumas linhas é menos grave que remover colunas inteiras.

https://pandas.pydata.org/pandas-docs/stable/missing_data.html

Remover valores nulos

- Remover valores nulos com dropna():

```
>>> df = pd.DataFrame({"name": ['Alfred', 'Batman', 'Catwoman'],  
...                    "toy": [np.nan, 'Batmobile', 'Bullwhip'],  
...                    "born": [pd.NaT, pd.Timestamp("1940-04-25"),  
...                             pd.NaT]})  
>>> df
```

	name	toy	born
0	Alfred	NaN	NaT
1	Batman	Batmobile	1940-04-25
2	Catwoman	Bullwhip	NaT

```
>>> df.dropna()
```

	name	toy	born
1	Batman	Batmobile	1940-04-25

```
>>> df.dropna(axis='columns')
```

	name
0	Alfred
1	Batman
2	Catwoman

Preencher ou substituir valores

- Há várias estratégias para preencher valores nulos ou inconsistentes;
- A mais comum, quando o valor é numérico, é preencher com a média dos demais;

```
# preencher os valores com a média
print(ds_credit['age'].mean())
media = ds_credit['age'][ds_credit['age'] > 0].mean()
print(media)
dataset.loc[ds_credit.age < 0, 'age'] = media
```

```
40.80755937840458
40.92770044906149
```

- Preencher valores nulos em mais de uma coluna:

```
# preencher os valores nulos com a média
ds_credit.loc[ds_credit['age'].isnull(), 'age'] = media
```

Para aprofundar

- Há bibliotecas especializadas em detecção de outliers, uma bem conhecida é a PyOD:

<https://pyod.readthedocs.io/en/latest/>

Tópicos Especiais em Sistemas de Informação

**Pré-processamento: tratando valores
inconsistentes e outliers**

Ely – elydasilvamiranda@gmail.com