Marcos Geraldo Braga Emiliano

19.1.4012

Caracterização de um conjunto de dados

A) Identifique os tipos de atributos:

(contínuos , discretos, binário (simétricos ou assimétricos), categóricos (nominais ou ordinais)).

1.PassengerId -> Discreto

2.Pclass -> Discreto

3.Name -> Categórico Nominal

4.Sex -> Binário Simetrico

5.Age -> Discreto

6.SibSp -> Discreto

7.Parch -> Discreto

8.Ticket -> Categórico Nominal

9.Fare -> Contínuo

10.Cabin -> Categórico Nominal

11.Embarked -> Categórico Nominal

```
!pip install kaggle
```

```
Looking in indexes: https://pypi.org/simple, https://us-python.pkg.dev/colab-wheels/p
Requirement already satisfied: kaggle in /usr/local/lib/python3.7/dist-packages (1.5
Requirement already satisfied: python-slugify in /usr/local/lib/python3.7/dist-packag
Requirement already satisfied: tqdm in /usr/local/lib/python3.7/dist-packages (from k
Requirement already satisfied: urllib3 in /usr/local/lib/python3.7/dist-packages (fro
Requirement already satisfied: certifi in /usr/local/lib/python3.7/dist-packages (fro
Requirement already satisfied: python-dateutil in /usr/local/lib/python3.7/dist-packa
Requirement already satisfied: six>=1.10 in /usr/local/lib/python3.7/dist-packages (f
Requirement already satisfied: requests in /usr/local/lib/python3.7/dist-packages (fr
Requirement already satisfied: text-unidecode>=1.3 in /usr/local/lib/python3.7/dist-p
Requirement already satisfied: idna<3,>=2.5 in /usr/local/lib/python3.7/dist-packages
Requirement already satisfied: chardet<4,>=3.0.2 in /usr/local/lib/python3.7/dist-pac
```

```
from google.colab import drive
drive.mount('/content/drive')
import pandas as pd
import numpy as np
import math

%matplotlib inline
```

```
Drive already mounted at /content/drive; to attempt to forcibly remount, call drive.m
```

```
train_data = pd.read_csv('/content/drive/My Drive/kaggle-titanic/train.csv')
test_data = pd.read_csv('/content/drive/My Drive/kaggle-titanic/test.csv')
p_id = test_data['PassengerId']
data = pd.concat([train_data, test_data])
data.shape
```

```
(1309, 12)
```

B) Atributos Numericos

```
#train_data['PassengerId'][i]

min = np.min(data['PassengerId'])
max = np.max(data['PassengerId'])
media = sum(data['PassengerId'])/1309
desv= math.sqrt(np.sum((data['PassengerId']-media)**2)/1309)
inter=max-min
out=[]

print("PassengerId:")
print("Minimo: ",min)
print("Maximo: ",max)
print("Media: ",media)
print("Desvio Padrao: ", desv)
print("Intervalo: ", inter)
print("Valores Aberrantes: ", out)
```

```
PassengerId:
Minimo:  1
Maximo:  1309
Media:  962.283950617284
Desvio Padrao:  590.3378163508897
Intervalo:  1308
Valores Aberrantes:  []
```

```
min = np.min(data['Pclass'])
max = np.max(data['Pclass'])
media = sum(data['Pclass'])/1309
desv= math.sqrt(np.sum((data['Pclass']-media)**2)/1309)
inter=max-min
out=[]

print("Pclass:")
print("Minimo: ",min)
print("Maximo: ",max)
print("Media: ",media)
print("Desvio Padrao: ", desv)
print("Intervalo: ", inter)
print("Valores Aberrantes: ", out)
```

```
     Pclass:
     Minimo:  1
     Maximo:  3
     Media:  2.308641975308642
     Desvio Padrao:  0.8356019334795166
     Intervalo:  2
     Valores Aberrantes:  []
```

```
#train_data['PassengerId'][i]

min = np.min(data['Age'])
max = np.max(data['Age'])
media = sum(data['Age'])/1309
desv= math.sqrt(np.sum((data['Age']-media)**2)/1309)
inter=max-min
out=[]

print("Age:")
print("Minimo: ",min)
print("Maximo: ",max)
print("Media: ",media)
print("Desvio Padrao: ", desv)
print("Intervalo: ", inter)
print("Valores Aberrantes: ", out)
```

```
     Age:
     Minimo:  0.42
     Maximo:  80.0
     Media:  nan
     Desvio Padrao:  0.0
     Intervalo:  79.58
     Valores Aberrantes:  []
```

```
#train_data['PassengerId'][i]

min = np.min(data['SibSp'])
max = np.max(data['SibSp'])
media = sum(data['SibSp'])/1309
desv= math.sqrt(np.sum((data['SibSp']-media)**2)/1309)
inter=max-min
out=[]

print("SibSp:")
print("Minimo: ",min)
print("Maximo: ",max)
print("Media: ",media)
print("Desvio Padrao: ", desv)
print("Intervalo: ", inter)
print("Valores Aberrantes: ", out)
```

```
     SibSp:
```

```
     Minimo:  0
     Maximo:  8
     Media:  0.5230078563411896
     Desvio Padrao:  1.1021244350892878
     Intervalo:  8
     Valores Aberrantes:  []
```

```
#train_data['PassengerId'][i]

min = np.min(data['Parch'])
max = np.max(data['Parch'])
media = sum(data['Parch'])/1309
desv= math.sqrt(np.sum((data['Parch']-media)**2)/1309)
inter=max-min
out=[]

print("Parch:")
print("Minimo: ",min)
print("Maximo: ",max)
print("Media: ",media)
print("Desvio Padrao: ", desv)
print("Intervalo: ", inter)
print("Valores Aberrantes: ", out)
```

```
     Parch:
     Minimo:  0
     Maximo:  6
     Media:  0.38159371492704824
     Desvio Padrao:  0.8056047612452208
     Intervalo:  6
     Valores Aberrantes:  []
```

```
#train_data['PassengerId'][i]

min = np.min(data['Fare'])
max = np.max(data['Fare'])
media = sum(data['Fare'])/1309
desv= math.sqrt(np.sum((data['Fare']-media)**2)/1309)
inter=max-min
out=[]

print("Fare:")
print("Minimo: ",min)
print("Maximo: ",max)
print("Media: ",media)
print("Desvio Padrao: ", desv)
print("Intervalo: ", inter)
print("Valores Aberrantes: ", out)
```

```
     Fare:
     Minimo:  0.0
     Maximo:  512.3292
     Media:  32.2042079685746
     Desvio Padrao:  49.66553444477411
```

```
Intervalo:  512.3292
Valores Aberrantes:  []
```

C)Atributos binários, nominais e ordinais

```
print("Name: ")
print(data['Name'])
```

```
Name: 
0                                Braund, Mr. Owen Harris
1      Cumings, Mrs. John Bradley (Florence Briggs Th...
2                                 Heikkinen, Miss. Laina
3           Futrelle, Mrs. Jacques Heath (Lily May Peel)
4                               Allen, Mr. William Henry
                             ...                        
886                                Montvila, Rev. Juozas
887                         Graham, Miss. Margaret Edith
888             Johnston, Miss. Catherine Helen "Carrie"
889                                Behr, Mr. Karl Howell
890                                  Dooley, Mr. Patrick
Name: Name, Length: 891, dtype: object
```

```
print("Sex: Female || Male")
#print(data['Sex'][30])

female=0
male=0
for i in range(891):
  if(((train_data['Sex'][i]) == 'female')):
    female += 1
  else:
   male += 1

print("Female =",female/891)
print("Male =", male/891)
```

```
Sex: Female || Male
Female = 0.35241301907968575
Male = 0.6475869809203143
```

```
print("ticket: ")
print(data['Ticket'])
```

```
ticket: 
0               A/5 21171
1                PC 17599
2        STON/O2. 3101282
3                  113803
4                  373450
```

```
                ...
    413             A.5. 3236
    414              PC 17758
    415    SOTON/O.Q. 3101262
    416                359309
    417                  2668
    Name: Ticket, Length: 1309, dtype: object
```

```
print("Cabin: ")
print(data['Cabin'])
nan=0
```

```
    Cabin:
    0       NaN
    1       C85
    2       NaN
    3      C123
    4       NaN
           ...
    413     NaN
    414    C105
    415     NaN
    416     NaN
    417     NaN
    Name: Cabin, Length: 1309, dtype: object
```

```
#print(data['Sex'][30])

S=0
Q=0
C=0
for i in range(891):
  if(((train_data['Embarked'][i]) == 'S')):
    S += 1
  if(((train_data['Embarked'][i]) == 'Q')):
    Q +=1
  if(((train_data['Embarked'][i]) == 'C')):
   C += 1

print("Embarked:")
print("S =", S/891)
print("Q =", Q/891)
print("C =", C/891)
```

```
    Embarked:
    S = 0.7227833894500562
    Q = 0.08641975308641975
    C = 0.18855218855218855
```

D)

E) Não sei de que forma seria possivel extrair correlação, porem com uma analise manual não encontrei correlação entre atributos

✓ 0s conclusão: 11:06 ● ×