Universidade Federal do Agreste de Pernambuco
Bacharelado em Ciências da Computação
Prof. Tiago Buarque A. de Carvalho
Alunos: David Brito e Laisy Cristina

Aprendizagem de Máquina:
Pré-processamento de Dados

1. Nesta questão você deve utilizar a base Student Performance , archive.ics.uci.edu/ml/datasets/Student+Performance (ver arquivo student-mat.csv no student.zip).

   (a) (5 pontos) Explique qual a forma mais adequada para converter todos os atributos da base para numéricos.

   Usar a função df.map para converter os atributos de String para numéricos. No caso dos binários nominais, os atributos permanecem os mesmos, mas os valores são substituídos. Já no caso de binários não nominais, é necessário criar novos atributos e eliminar os antigos.

   (b) (10 pontos) Converta todos os atributos da base para numéricos (exceto a classe).

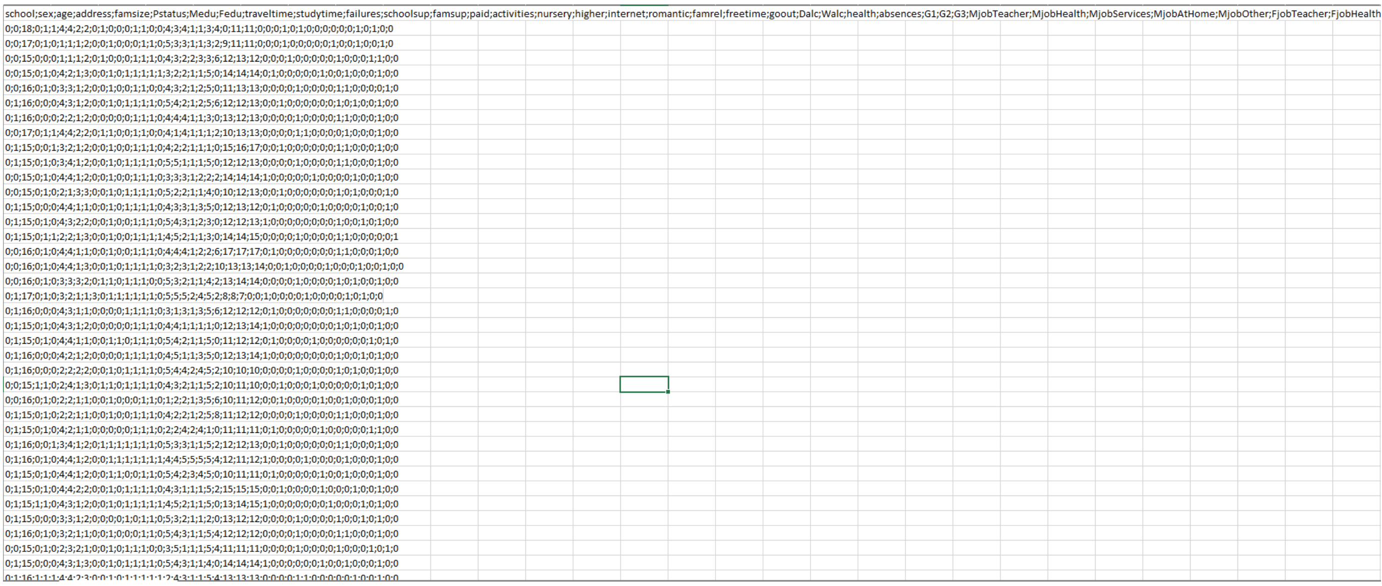
```
import pandas as pd
import collections

df = pd.read_csv('./student-mat.csv', sep=';')

#binario nominal

df['school'] = df['school'].map({'GP': 0, 'MS': 1 })
df['sex'] = df['sex'].map({'F': 0, 'M': 1 })
df['address'] = df['address'].map({'U': 0, 'R': 1 })
df['famsize'] = df['famsize'].map({'LE3': 0, 'GT3': 1 })
df['Pstatus'] = df['Pstatus'].map({'T': 0, 'A': 1 })
df['schoolsup'] = df['schoolsup'].map({'yes': 1, 'no': 0})
df['famsup'] = df['famsup'].map({'yes': 1, 'no': 0})
df['paid'] = df['paid'].map({'yes': 1, 'no': 0})
df['activities'] = df['activities'].map({'yes': 1, 'no': 0})
df['nursery'] = df['nursery'].map({'yes': 1, 'no': 0})
df['higher'] = df['higher'].map({'yes': 1, 'no': 0})
df['internet'] = df['internet'].map({'yes': 1, 'no': 0})
df['romantic'] = df['romantic'].map({'yes': 1, 'no': 0})
```

```python
## não binário nominal - cria novos atrbutos e elimina o antigo
df['MjobTeacher'] = df['Mjob'].map(collections.defaultdict(lambda: 0, { 'teacher': 1}))
df['MjobHealth'] = df['Mjob'].map(collections.defaultdict(lambda: 0, { 'health': 1}))
df['MjobServices'] = df['Mjob'].map(collections.defaultdict(lambda: 0, { 'services': 1}))
df['MjobAtHome'] = df['Mjob'].map(collections.defaultdict(lambda: 0, { 'at_home': 1}))
df['MjobOther'] = df['Mjob'].map(collections.defaultdict(lambda: 0, { 'other': 1}))
del df['Mjob']

df['FjobTeacher'] = df['Fjob'].map(collections.defaultdict(lambda: 0, { 'teacher': 1}))
df['FjobHealth'] = df['Fjob'].map(collections.defaultdict(lambda: 0, { 'health': 1}))
df['FjobServices'] = df['Fjob'].map(collections.defaultdict(lambda: 0, { 'services': 1}))
df['FjobAtHome'] = df['Fjob'].map(collections.defaultdict(lambda: 0, { 'at_home': 1}))
df['FjobOther'] = df['Fjob'].map(collections.defaultdict(lambda: 0, { 'other': 1}))
del df['Fjob']

df['reasonClose'] = df['reason'].map(collections.defaultdict(lambda: 0, { 'home': 1}))
df['reasonSchool'] = df['reason'].map(collections.defaultdict(lambda: 0, { 'reputation': 1}))
df['reasonCourse'] = df['reason'].map(collections.defaultdict(lambda: 0, { 'course': 1}))
df['reasonOther'] = df['reason'].map(collections.defaultdict(lambda: 0, { 'other': 1}))
del df['reason']

df['guardianMother'] = df['guardian'].map(collections.defaultdict(lambda: 0, { 'mother': 1}))
df['guardianFather'] = df['guardian'].map(collections.defaultdict(lambda: 0, { 'father': 1}))
df['guardianOther'] = df['guardian'].map(collections.defaultdict(lambda: 0, { 'other': 1}))
del df['guardian']
```

Base pré-conversão:


```
school;sex;age;address;famsize;Pstatus;Medu;Fedu;Mjob;Fjob;reason;guardian;traveltime;studytime;failures;schoolsup;famsup;paid;activities;nursery;higher;internet;romantic;famrel;freetime;goout;Dalc;Walc;health;absences;G1;G2;G3
GP;"F";18;"U";"GT3";"A";4;4;"at_home";"teacher";"course";"mother";2;2;0;"yes";"no";"no";"no";"yes";"yes";"no";"no";4;3;4;1;1;3;6;"5";"6";6
GP;"F";17;"U";"GT3";"T";1;1;"at_home";"other";"course";"father";1;2;0;"no";"yes";"no";"no";"no";"yes";"yes";"no";5;3;3;1;1;3;4;"5";"5";6
GP;"F";15;"U";"LE3";"T";1;1;"at_home";"other";"other";"mother";1;2;3;"yes";"no";"yes";"no";"yes";"yes";"yes";"no";4;3;2;2;3;3;10;"7";"8";10
GP;"F";15;"U";"GT3";"T";4;2;"health";"services";"home";"mother";1;3;0;"no";"yes";"yes";"yes";"yes";"yes";"yes";"yes";3;2;2;1;1;5;2;"15";"14";15
GP;"F";16;"U";"GT3";"T";3;3;"other";"other";"home";"father";1;2;0;"no";"yes";"yes";"no";"yes";"yes";"no";"no";4;3;2;1;2;5;4;"6";"10";10
GP;"M";16;"U";"LE3";"T";4;3;"services";"other";"reputation";"mother";1;2;0;"no";"yes";"yes";"yes";"yes";"yes";"yes";"no";5;4;2;1;2;5;10;"15";"15";15
GP;"M";16;"U";"LE3";"T";2;2;"other";"other";"home";"mother";1;2;0;"no";"no";"no";"no";"yes";"yes";"yes";"no";4;4;4;1;1;3;0;"12";"12";11
GP;"F";17;"U";"GT3";"A";4;4;"other";"teacher";"home";"mother";2;2;0;"yes";"yes";"no";"no";"yes";"yes";"no";"no";4;1;4;1;1;1;6;"6";"5";6
GP;"M";15;"U";"LE3";"A";3;2;"services";"other";"home";"mother";1;2;0;"no";"yes";"yes";"no";"yes";"yes";"yes";"no";4;2;2;1;1;1;0;"16";"18";19
GP;"M";15;"U";"GT3";"T";3;4;"other";"other";"home";"mother";1;2;0;"no";"yes";"yes";"yes";"yes";"yes";"yes";"no";5;5;1;1;1;5;0;"14";"15";15
GP;"F";15;"U";"GT3";"T";4;4;"teacher";"health";"reputation";"mother";1;2;0;"no";"yes";"yes";"no";"yes";"yes";"yes";"no";3;3;3;1;2;2;0;"10";"8";9
GP;"F";15;"U";"GT3";"T";2;1;"services";"other";"reputation";"father";3;3;0;"no";"yes";"no";"yes";"yes";"yes";"yes";"no";5;2;2;1;1;4;4;"10";"12";12
GP;"M";15;"U";"LE3";"T";4;4;"health";"services";"course";"father";1;1;0;"no";"yes";"yes";"yes";"yes";"yes";"yes";"no";4;3;3;1;3;5;2;"14";"14";14
GP;"M";15;"U";"GT3";"T";4;3;"teacher";"other";"course";"mother";2;2;0;"no";"yes";"yes";"no";"yes";"yes";"yes";"no";5;4;3;1;2;3;2;"10";"10";11
GP;"M";15;"U";"GT3";"A";2;2;"other";"other";"home";"other";1;3;0;"no";"yes";"no";"no";"yes";"yes";"yes";"yes";4;5;2;1;1;3;0;"14";"16";16
GP;"F";16;"U";"GT3";"T";4;4;"health";"other";"home";"mother";1;1;0;"no";"yes";"no";"no";"yes";"yes";"yes";"no";4;4;4;1;2;2;4;"14";"14";14
GP;"F";16;"U";"GT3";"T";4;4;"services";"services";"reputation";"mother";1;3;0;"no";"yes";"yes";"yes";"yes";"yes";"yes";"no";3;2;3;1;2;2;6;"13";"14";14
GP;"F";16;"U";"GT3";"T";3;3;"other";"other";"reputation";"mother";3;2;0;"yes";"yes";"no";"yes";"yes";"yes";"no";"no";5;3;2;1;1;4;4;"8";"10";10
GP;"M";17;"U";"GT3";"T";3;2;"services";"services";"course";"mother";1;1;3;"no";"yes";"no";"yes";"yes";"yes";"yes";"no";5;5;5;2;4;5;16;"6";"5";5
GP;"M";16;"U";"LE3";"T";4;3;"health";"other";"home";"father";1;1;0;"no";"no";"yes";"yes";"yes";"yes";"yes";"no";3;1;3;1;3;5;4;"8";"10";10
GP;"M";15;"U";"GT3";"T";4;3;"teacher";"other";"reputation";"mother";1;2;0;"no";"no";"no";"no";"yes";"yes";"yes";"no";4;4;1;1;1;1;0;"13";"14";15
GP;"M";15;"U";"GT3";"T";4;4;"health";"health";"other";"father";1;1;0;"no";"yes";"yes";"no";"yes";"yes";"yes";"no";5;4;2;1;1;5;0;"12";"15";15
GP;"M";16;"U";"LE3";"T";4;2;"teacher";"other";"course";"mother";1;2;0;"no";"no";"no";"yes";"yes";"yes";"yes";"no";4;5;1;1;3;5;2;"15";"15";16
GP;"M";16;"U";"LE3";"T";2;2;"other";"other";"reputation";"mother";2;2;0;"no";"yes";"no";"yes";"yes";"yes";"yes";"no";5;4;4;2;4;5;0;"13";"13";12
GP;"F";15;"R";"GT3";"T";2;4;"services";"health";"course";"mother";1;3;0;"yes";"yes";"yes";"yes";"yes";"yes";"yes";"no";4;3;2;1;1;5;2;"10";"9";8
GP;"F";16;"U";"GT3";"T";2;2;"services";"services";"home";"mother";1;1;2;"no";"yes";"yes";"no";"no";"yes";"yes";"no";1;2;2;1;3;5;14;"6";"9";8
GP;"M";15;"U";"GT3";"T";2;2;"other";"other";"home";"mother";1;1;0;"no";"yes";"yes";"no";"yes";"yes";"yes";"no";4;2;2;1;2;5;2;"12";"12";11
GP;"M";15;"U";"GT3";"T";4;2;"health";"services";"other";"mother";1;1;0;"no";"no";"yes";"no";"yes";"yes";"yes";"no";2;2;4;2;4;1;4;"15";"16";15
GP;"M";16;"U";"LE3";"A";3;4;"services";"other";"home";"mother";1;2;0;"yes";"yes";"no";"yes";"yes";"yes";"yes";"no";5;3;3;1;1;5;4;"11";"11";11
GP;"M";16;"U";"GT3";"T";4;4;"teacher";"teacher";"home";"mother";1;2;0;"no";"yes";"yes";"yes";"yes";"yes";"yes";"yes";4;4;5;5;5;5;16;"10";"12";11
GP;"M";15;"U";"GT3";"T";4;4;"health";"services";"home";"mother";1;2;0;"no";"yes";"yes";"no";"no";"yes";"yes";"no";5;4;2;3;4;5;0;"9";"11";12
GP;"M";15;"U";"GT3";"T";4;4;"services";"services";"reputation";"mother";2;2;0;"no";"yes";"no";"yes";"yes";"yes";"yes";"no";4;3;1;1;1;5;0;"17";"16";17
GP;"M";15;"R";"GT3";"T";4;3;"teacher";"at_home";"course";"mother";1;2;0;"no";"yes";"no";"yes";"yes";"yes";"yes";"yes";4;5;2;1;1;5;0;"17";"16";16
GP;"M";15;"U";"LE3";"T";3;3;"other";"other";"course";"mother";1;2;0;"no";"no";"no";"yes";"no";"yes";"yes";"no";5;3;2;1;1;2;0;"8";"10";12
GP;"M";16;"U";"GT3";"T";3;2;"other";"other";"home";"mother";1;1;0;"no";"yes";"yes";"no";"no";"yes";"yes";"no";5;4;3;1;1;5;0;"12";"14";15
GP;"F";15;"U";"GT3";"T";2;3;"other";"other";"other";"father";2;1;0;"no";"yes";"no";"yes";"yes";"yes";"no";"no";3;5;1;1;1;5;0;"8";"7";6
GP;"M";15;"U";"LE3";"T";4;3;"teacher";"services";"home";"mother";1;3;0;"no";"yes";"no";"yes";"yes";"yes";"yes";"no";5;4;3;1;1;4;2;"15";"16";18
GP;"M";16;"R";"GT3";"A";4;4;"other";"teacher";"reputation";"mother";2;3;0;"no";"yes";"no";"yes";"yes";"yes";"yes";"yes";2;4;3;1;1;5;7;"15";"16";15
```


Base pós conversão:



```
school;sex;age;address;famsize;Pstatus;Medu;Fedu;traveltime;studytime;failures;schoolsup;famsup;paid;activities;nursery;higher;internet;romantic;famrel;freetime;goout;Dalc;Walc;health;absences;G1;G2;G3;MjobTeacher;MjobHealth;MjobServices;MjobAtHome;MjobOther;FjobTeacher;FjobHealth
0;0;18;0;1;1;4;4;2;2;0;1;0;0;0;1;1;0;0;4;3;4;1;1;3;4;0;11;11;0;0;0;1;0;1;0;0;0;0;0;0;1;0;1;0;0
0;0;17;0;1;0;1;1;1;2;0;0;1;0;0;0;1;1;0;5;3;3;1;1;3;2;9;11;11;0;0;0;1;0;0;0;0;0;1;0;0;1;0;0;1;0
0;0;15;0;0;0;1;1;1;2;0;1;0;0;0;1;1;1;0;4;3;2;2;3;3;6;12;13;12;0;0;0;1;0;0;0;0;0;1;0;0;0;1;1;0;0
0;0;15;0;1;0;4;2;1;3;0;0;1;0;1;1;1;1;1;3;2;2;1;1;5;0;14;14;14;0;1;0;0;0;0;0;1;0;0;1;0;0;0;1;0;0
0;0;16;0;1;0;3;3;1;2;0;0;1;0;0;1;1;0;0;4;3;2;1;2;5;0;11;13;13;0;0;0;0;1;0;0;0;0;1;1;0;0;0;0;1;0
0;1;16;0;0;0;4;3;1;2;0;0;1;0;1;1;1;1;0;5;4;2;1;2;5;6;12;12;13;0;0;1;0;0;0;0;0;0;1;0;1;0;0;1;0;0
0;1;16;0;0;0;2;2;1;2;0;0;0;0;0;1;1;1;0;4;4;4;1;1;3;0;13;12;13;0;0;0;0;1;0;0;0;0;1;1;0;0;0;1;0;0
0;0;17;0;1;1;4;4;2;2;0;1;1;0;0;1;1;0;0;4;1;4;1;1;1;2;10;13;13;0;0;0;0;1;1;0;0;0;0;1;0;0;0;1;0;0
0;1;15;0;0;1;3;2;1;2;0;0;1;0;0;1;1;1;0;4;2;2;1;1;1;0;15;16;17;0;0;1;0;0;0;0;0;0;1;1;0;0;0;1;0;0
0;1;15;0;1;0;3;4;1;2;0;0;1;0;1;1;1;1;0;5;5;1;1;1;5;0;12;12;13;0;0;0;0;1;0;0;0;0;1;1;0;0;0;1;0;0
0;0;15;0;1;0;4;4;1;2;0;0;1;0;0;1;1;1;0;3;3;3;1;2;2;2;14;14;14;1;0;0;0;0;0;1;0;0;0;0;1;0;0;1;0;0
0;0;15;0;1;0;2;1;3;3;0;0;1;0;1;1;1;1;0;5;2;2;1;1;4;0;10;12;13;0;0;1;0;0;0;0;0;0;1;0;1;0;0;0;1;0
0;1;15;0;0;0;4;4;1;1;0;0;1;0;1;1;1;1;0;4;3;3;1;3;5;0;12;13;12;0;1;0;0;0;0;0;1;0;0;0;0;1;0;0;1;0
0;1;15;0;1;0;4;3;2;2;0;0;1;0;0;1;1;1;0;5;4;3;1;2;3;0;12;12;13;1;0;0;0;0;0;0;0;0;1;0;0;1;0;1;0;0
0;1;15;0;1;1;2;2;1;3;0;0;1;0;0;1;1;1;1;4;5;2;1;1;3;0;14;14;15;0;0;0;0;1;0;0;0;0;1;1;0;0;0;0;0;1
0;0;16;0;1;0;4;4;1;1;0;0;1;0;0;1;1;1;0;4;4;4;1;2;2;6;17;17;17;0;1;0;0;0;0;0;0;0;1;1;0;0;0;1;0;0
0;0;16;0;1;0;4;4;1;3;0;0;1;0;1;1;1;1;0;3;2;3;1;2;2;10;13;13;14;0;0;1;0;0;0;0;1;0;0;0;1;0;0;1;0;0
0;0;16;0;1;0;3;3;3;2;0;1;1;0;1;1;1;0;0;5;3;2;1;1;4;2;13;14;14;0;0;0;0;1;0;0;0;0;1;0;1;0;0;1;0;0
0;1;17;0;1;0;3;2;1;1;3;0;1;1;1;1;1;1;0;5;5;5;2;4;5;2;8;8;7;0;0;1;0;0;0;0;1;0;0;0;0;1;0;1;0;0
0;1;16;0;0;0;4;3;1;1;0;0;0;0;1;1;1;1;0;3;1;3;1;3;5;6;12;12;12;0;1;0;0;0;0;0;0;0;1;1;0;0;0;0;1;0
0;1;15;0;1;0;4;3;1;2;0;0;0;0;0;1;1;1;0;4;4;1;1;1;1;0;12;13;14;1;0;0;0;0;0;0;0;0;1;0;1;0;0;1;0;0
0;1;15;0;1;0;4;4;1;1;0;0;1;1;0;1;1;1;0;5;4;2;1;1;5;0;11;12;12;0;1;0;0;0;0;1;0;0;0;0;0;0;1;0;1;0
0;1;16;0;0;0;4;2;1;2;0;0;0;0;1;1;1;1;0;4;5;1;1;3;5;0;12;13;14;1;0;0;0;0;0;0;0;0;1;0;0;1;0;1;0;0
0;1;16;0;0;0;2;2;2;2;0;0;1;0;1;1;1;1;0;5;4;4;2;4;5;2;10;10;10;0;0;0;0;1;0;0;0;0;1;0;1;0;0;1;0;0
0;0;15;1;1;0;2;4;1;3;0;1;1;0;1;1;1;1;0;4;3;2;1;1;5;2;10;11;10;0;0;1;0;0;0;1;0;0;0;0;0;1;0;1;0;0
0;0;16;0;1;0;2;2;1;1;0;0;1;0;0;0;1;1;0;1;2;2;1;3;5;6;10;11;12;0;0;1;0;0;0;0;1;0;0;1;0;0;0;1;0;0
0;1;15;0;1;0;2;2;1;1;0;0;1;0;0;1;1;1;0;4;2;2;1;2;5;8;11;12;12;0;0;0;0;1;0;0;0;0;1;1;0;0;0;1;0;0
0;1;15;0;1;0;4;2;1;1;0;0;0;0;0;1;1;1;0;2;2;4;2;4;1;0;11;11;11;0;1;0;0;0;0;0;1;0;0;0;0;0;1;1;0;0
0;1;16;0;0;1;3;4;1;2;0;1;1;1;1;1;1;1;0;5;3;3;1;1;5;2;12;12;13;0;0;1;0;0;0;0;0;0;1;1;0;0;0;1;0;0
0;1;16;0;1;0;4;4;1;2;0;0;1;1;1;1;1;1;1;4;4;5;5;5;5;4;12;11;12;1;0;0;0;0;1;0;0;0;0;1;0;0;0;1;0;0
0;1;15;0;1;0;4;4;1;2;0;0;1;1;0;0;1;1;0;5;4;2;3;4;5;0;10;11;11;0;1;0;0;0;0;0;1;0;0;1;0;0;0;1;0;0
0;1;15;0;1;0;4;4;2;2;0;0;1;0;1;1;1;1;0;4;3;1;1;1;5;2;15;15;15;0;0;1;0;0;0;0;1;0;0;0;1;0;0;1;0;0
0;1;15;1;1;0;4;3;1;2;0;0;1;0;1;1;1;1;1;4;5;2;1;1;5;0;13;14;15;1;0;0;0;0;0;0;0;1;0;0;0;1;0;1;0;0
0;1;15;0;0;0;3;3;1;2;0;0;0;0;1;0;1;1;0;5;3;2;1;1;2;0;13;12;12;0;0;0;0;1;0;0;0;0;1;0;0;1;0;1;0;0
0;1;16;0;1;0;3;2;1;1;0;0;1;0;0;0;1;1;0;5;4;3;1;1;5;4;12;12;12;0;0;0;0;1;0;0;0;0;1;1;0;0;0;1;0;0
0;0;15;0;1;0;2;3;2;1;0;0;1;0;1;1;1;0;0;3;5;1;1;1;5;4;11;11;11;0;0;0;0;1;0;0;0;0;1;0;0;0;1;0;1;0
0;1;15;0;0;0;4;3;1;3;0;0;1;0;1;1;1;1;0;5;4;3;1;1;4;0;14;14;14;1;0;0;0;0;0;0;1;0;0;1;0;0;0;1;0;0
0;1;16;1;1;1;4;4;2;3;0;0;1;0;1;1;1;1;2;4;3;1;1;5;4;13;13;13;0;0;0;0;1;1;0;0;0;0;0;1;0;0;1;0;0
```

(c) (10 pontos) Assuma a última coluna (G3, que representa a nota nal de cada estudante) como classe. Converta esta coluna (atributo numérico) para uma variável categórica bi nária. Após esta conversão é possível realizar a tarefa a seguir.

Convertendo os valores entre 0~10 para a 0 e os valores 11~20 para 1, temos:

```python
import pandas as pd
import numpy as np
from sklearn.model_selection import train_test_split
from sklearn.neighbors import  KNeighborsRegressor

df = pd.read_csv(r'c:\Users\David\Desktop\RP-2022.1\semana 5\student-mat.csv', sep=';')


df['G3'] = df['G3'].map({0: 0, 1: 0, 2: 0, 3: 0, 4: 0, 5: 0,
                        6: 0, 7: 0, 8: 0, 9: 0, 10: 0, 11: 1,
                        12: 1, 13: 1, 14: 1, 15: 1, 16: 1, 17: 1,
                        18: 1, 19: 1, 20: 1})



df.to_csv(r'c:\Users\David\Desktop\RP-2022.1\semana 5\student-matResultado2.csv', sep=';', index=False)
```

Base pré-conversão:

```
school;sex;age;address;famsize;Pstatus;Medu;Fedu;Mjob;Fjob;reason;guardian;traveltime;studytime;failures;schoolsup;famsup;paid;activities;nursery;higher;internet;romantic;famrel;freetime;goout;Dalc;Walc;health;absences;G1;G2;G3
GP;"F";18;"U";"GT3";"A";4;4;"at_home";"teacher";"course";"mother";2;2;0;"yes";"no";"no";"no";"yes";"yes";"no";"no";4;3;4;1;1;3;6;"5";"6";6
GP;"F";17;"U";"GT3";"T";1;1;"at_home";"other";"course";"father";1;2;0;"no";"yes";"no";"no";"no";"yes";"yes";"no";5;3;3;1;1;3;4;"5";"5";6
GP;"F";15;"U";"LE3";"T";1;1;"at_home";"other";"other";"mother";1;2;3;"yes";"no";"yes";"no";"yes";"yes";"yes";"no";4;3;2;2;3;3;10;"7";"8";10
GP;"F";15;"U";"GT3";"T";4;2;"health";"services";"home";"mother";1;3;0;"no";"yes";"yes";"yes";"yes";"yes";"yes";"yes";3;2;2;1;1;5;2;"15";"14";15
GP;"F";16;"U";"GT3";"T";3;3;"other";"other";"home";"father";1;2;0;"no";"yes";"yes";"no";"yes";"yes";"no";"no";4;3;2;1;2;5;4;"6";"10";10
GP;"M";16;"U";"LE3";"T";4;3;"services";"other";"reputation";"mother";1;2;0;"no";"yes";"yes";"yes";"yes";"yes";"yes";"no";5;4;2;1;2;5;10;"15";"15";15
GP;"M";16;"U";"LE3";"T";2;2;"other";"other";"home";"mother";1;2;0;"no";"no";"no";"no";"yes";"yes";"yes";"no";4;4;4;1;1;3;0;"12";"12";11
GP;"F";17;"U";"GT3";"A";4;4;"other";"teacher";"home";"mother";2;2;0;"yes";"yes";"no";"no";"yes";"yes";"no";"no";4;1;4;1;1;1;6;"6";"5";6
GP;"M";15;"U";"LE3";"A";3;2;"services";"other";"home";"mother";1;2;0;"no";"yes";"yes";"no";"yes";"yes";"yes";"no";4;2;2;1;1;1;0;"16";"18";19
GP;"M";15;"U";"GT3";"T";3;4;"other";"other";"home";"mother";1;2;0;"no";"yes";"yes";"yes";"yes";"yes";"yes";"no";5;5;1;1;1;5;0;"14";"15";15
GP;"F";15;"U";"GT3";"T";4;4;"teacher";"health";"reputation";"mother";1;2;0;"no";"yes";"yes";"no";"yes";"yes";"yes";"no";3;3;3;1;2;2;0;"10";"8";9
GP;"F";15;"U";"GT3";"T";2;1;"services";"other";"reputation";"father";3;3;0;"no";"yes";"no";"yes";"yes";"yes";"yes";"no";5;2;2;1;1;4;4;"10";"12";12
GP;"M";15;"U";"LE3";"T";4;4;"health";"services";"course";"father";1;1;0;"no";"yes";"yes";"yes";"yes";"yes";"yes";"no";4;3;3;1;3;5;2;"14";"14";14
GP;"M";15;"U";"GT3";"T";4;3;"teacher";"other";"course";"mother";2;2;0;"no";"yes";"yes";"no";"yes";"yes";"yes";"no";5;4;3;1;2;3;2;"10";"10";11
GP;"M";15;"U";"GT3";"A";2;2;"other";"other";"home";"other";1;3;0;"no";"yes";"no";"no";"yes";"yes";"yes";"yes";4;5;2;1;1;3;0;"14";"16";16
GP;"F";16;"U";"GT3";"T";4;4;"health";"other";"home";"mother";1;1;0;"no";"yes";"no";"no";"yes";"yes";"yes";"no";4;4;4;1;2;2;4;"14";"14";14
GP;"F";16;"U";"GT3";"T";4;4;"services";"services";"reputation";"mother";1;3;0;"no";"yes";"yes";"yes";"yes";"yes";"yes";"no";3;2;3;1;2;2;6;"13";"14";14
GP;"F";16;"U";"GT3";"T";3;3;"other";"other";"reputation";"mother";3;2;0;"yes";"yes";"no";"yes";"yes";"yes";"no";"no";5;3;2;1;1;4;4;"8";"10";10
GP;"M";17;"U";"GT3";"T";3;2;"services";"services";"course";"mother";1;1;3;"no";"yes";"no";"yes";"yes";"yes";"yes";"no";5;5;5;2;4;5;16;"6";"5";5
GP;"M";16;"U";"LE3";"T";4;3;"health";"other";"home";"father";1;1;0;"no";"no";"yes";"yes";"yes";"yes";"yes";"no";3;1;3;1;3;5;4;"8";"10";10
GP;"M";15;"U";"GT3";"T";4;3;"teacher";"other";"reputation";"mother";1;2;0;"no";"no";"no";"no";"yes";"yes";"yes";"no";4;4;1;1;1;1;0;"13";"14";15
GP;"M";15;"U";"GT3";"T";4;4;"health";"health";"other";"father";1;1;0;"no";"yes";"yes";"no";"yes";"yes";"yes";"no";5;4;2;1;1;5;0;"12";"15";15
GP;"M";16;"U";"LE3";"T";4;2;"teacher";"other";"course";"mother";1;2;0;"no";"no";"no";"yes";"yes";"yes";"yes";"no";4;5;1;1;3;5;2;"15";"15";16
GP;"M";16;"U";"LE3";"T";2;2;"other";"other";"reputation";"mother";2;2;0;"no";"yes";"no";"yes";"yes";"yes";"yes";"no";5;4;4;2;4;5;0;"13";"13";12
GP;"F";15;"R";"GT3";"T";2;4;"services";"health";"course";"mother";1;3;0;"yes";"yes";"yes";"yes";"yes";"yes";"yes";"no";4;3;2;1;1;5;2;"10";"9";8
GP;"F";16;"U";"GT3";"T";2;2;"services";"services";"home";"mother";1;1;2;"no";"yes";"yes";"no";"no";"yes";"yes";"no";1;2;2;1;3;5;14;"6";"9";8
GP;"M";15;"U";"GT3";"T";2;2;"other";"other";"home";"mother";1;1;0;"no";"yes";"yes";"no";"yes";"yes";"yes";"no";4;2;2;1;2;5;2;"12";"12";11
GP;"M";15;"U";"GT3";"T";4;2;"health";"services";"other";"mother";1;1;0;"no";"no";"yes";"no";"yes";"yes";"yes";"no";2;2;4;2;4;1;4;"15";"16";15
GP;"M";16;"U";"LE3";"A";3;4;"services";"other";"home";"mother";1;2;0;"yes";"yes";"no";"yes";"yes";"yes";"yes";"no";5;3;3;1;1;5;4;"11";"11";11
GP;"M";16;"U";"GT3";"T";4;4;"teacher";"teacher";"home";"mother";1;2;0;"no";"yes";"yes";"yes";"yes";"yes";"yes";"yes";4;4;5;5;5;5;16;"10";"12";11
GP;"M";15;"U";"GT3";"T";4;4;"health";"services";"home";"mother";1;2;0;"no";"yes";"yes";"no";"no";"yes";"yes";"no";5;4;2;3;4;5;0;"9";"11";12
GP;"M";15;"U";"GT3";"T";4;4;"services";"services";"reputation";"mother";2;2;0;"no";"yes";"no";"yes";"yes";"yes";"yes";"no";4;3;1;1;1;5;0;"17";"16";17
GP;"M";15;"R";"GT3";"T";4;3;"teacher";"at_home";"course";"mother";1;2;0;"no";"yes";"no";"yes";"yes";"yes";"yes";"yes";4;5;2;1;1;5;0;"17";"16";16
GP;"M";15;"U";"LE3";"T";3;3;"other";"other";"course";"mother";1;2;0;"no";"no";"no";"yes";"no";"yes";"yes";"no";5;3;2;1;1;2;0;"8";"10";12
GP;"M";16;"U";"GT3";"T";3;2;"other";"other";"home";"mother";1;1;0;"no";"yes";"yes";"no";"no";"yes";"yes";"no";5;4;3;1;1;5;0;"12";"14";15
GP;"F";15;"U";"GT3";"T";2;3;"other";"other";"other";"father";2;1;0;"no";"yes";"no";"yes";"yes";"yes";"no";"no";3;5;1;1;1;5;0;"8";"7";6
GP;"M";15;"U";"LE3";"T";4;3;"teacher";"services";"home";"mother";1;3;0;"no";"yes";"no";"yes";"yes";"yes";"yes";"no";5;4;3;1;1;4;2;"15";"16";18
GP;"M";16;"R";"GT3";"A";4;4;"other";"teacher";"reputation";"mother";2;3;0;"no";"yes";"no";"yes";"yes";"yes";"yes";"yes";2;4;3;1;1;5;7;"15";"16";15
```

Base pós-conversão:

```
school;sex;age;address;famsize;Pstatus;Medu;Fedu;Mjob;Fjob;reason;guardian;traveltime;studytime;failures;schoolsup;famsup;paid;activities;nursery;higher;internet;romantic;famrel;freetime;goout;Dalc;Walc;health;absences;G1;G2;G3
GP;F;18;U;GT3;A;4;4;at_home;teacher;course;mother;2;2;0;yes;no;no;no;yes;yes;no;no;4;3;4;1;1;3;6;5;6;0
GP;F;17;U;GT3;T;1;1;at_home;other;course;father;1;2;0;no;yes;no;no;no;yes;yes;no;5;3;3;1;1;3;4;5;5;0
GP;F;15;U;LE3;T;1;1;at_home;other;other;mother;1;2;3;yes;no;yes;no;yes;yes;yes;no;4;3;2;2;3;3;10;7;8;0
GP;F;15;U;GT3;T;4;2;health;services;home;mother;1;3;0;no;yes;yes;yes;yes;yes;yes;yes;3;2;2;1;1;5;2;15;14;1
GP;F;16;U;GT3;T;3;3;other;other;home;father;1;2;0;no;yes;yes;no;yes;yes;no;no;4;3;2;1;2;5;4;6;10;0
GP;M;16;U;LE3;T;4;3;services;other;reputation;mother;1;2;0;no;yes;yes;yes;yes;yes;yes;no;5;4;2;1;2;5;10;15;15;1
GP;M;16;U;LE3;T;2;2;other;other;home;mother;1;2;0;no;no;no;no;yes;yes;yes;no;4;4;4;1;1;3;0;12;12;1
GP;F;17;U;GT3;A;4;4;other;teacher;home;mother;2;2;0;yes;yes;no;no;yes;yes;no;no;4;1;4;1;1;1;6;6;5;0
GP;M;15;U;LE3;A;3;2;services;other;home;mother;1;2;0;no;yes;yes;no;yes;yes;yes;no;4;2;2;1;1;1;0;16;18;1
GP;M;15;U;GT3;T;3;4;other;other;home;mother;1;2;0;no;yes;yes;yes;yes;yes;yes;no;5;5;1;1;1;5;0;14;15;1
GP;F;15;U;GT3;T;4;4;teacher;health;reputation;mother;1;2;0;no;yes;yes;no;yes;yes;yes;no;3;3;3;1;2;2;0;10;8;0
GP;F;15;U;GT3;T;2;1;services;other;reputation;father;3;3;0;no;yes;no;yes;yes;yes;yes;no;5;2;2;1;1;4;4;10;12;1
GP;M;15;U;LE3;T;4;4;health;services;course;father;1;1;0;no;yes;yes;yes;yes;yes;yes;no;4;3;3;1;3;5;2;14;14;1
GP;M;15;U;GT3;T;4;3;teacher;other;course;mother;2;2;0;no;yes;yes;no;yes;yes;yes;no;5;4;3;1;2;3;2;10;10;1
GP;M;15;U;GT3;A;2;2;other;other;home;other;1;3;0;no;yes;no;no;yes;yes;yes;yes;4;5;2;1;1;3;0;14;16;1
GP;F;16;U;GT3;T;4;4;health;other;home;mother;1;1;0;no;yes;no;no;yes;yes;yes;no;4;4;4;1;2;2;4;14;14;1
GP;F;16;U;GT3;T;4;4;services;services;reputation;mother;1;3;0;no;yes;yes;yes;yes;yes;yes;no;3;2;3;1;2;2;6;13;14;1
GP;F;16;U;GT3;T;3;3;other;other;reputation;mother;3;2;0;yes;yes;no;yes;yes;yes;no;no;5;3;2;1;1;4;4;8;10;0
GP;M;17;U;GT3;T;3;2;services;services;course;mother;1;1;3;no;yes;no;yes;yes;yes;yes;no;5;5;5;2;4;5;16;6;5;0
GP;M;16;U;LE3;T;4;3;health;other;home;father;1;1;0;no;no;yes;yes;yes;yes;yes;no;3;1;3;1;3;5;4;8;10;0
GP;M;15;U;GT3;T;4;3;teacher;other;reputation;mother;1;2;0;no;no;no;no;yes;yes;yes;no;4;4;1;1;1;1;0;13;14;1
GP;M;15;U;GT3;T;4;4;health;health;other;father;1;1;0;no;yes;yes;no;yes;yes;yes;no;5;4;2;1;1;5;0;12;15;1
GP;M;16;U;LE3;T;4;2;teacher;other;course;mother;1;2;0;no;no;no;yes;yes;yes;yes;no;4;5;1;1;3;5;2;15;15;1
GP;M;16;U;LE3;T;2;2;other;other;reputation;mother;2;2;0;no;yes;no;yes;yes;yes;yes;no;5;4;4;2;4;5;0;13;13;1
GP;F;15;R;GT3;T;2;4;services;health;course;mother;1;3;0;yes;yes;yes;yes;yes;yes;yes;no;4;3;2;1;1;5;2;10;9;0
GP;F;16;U;GT3;T;2;2;services;services;home;mother;1;1;2;no;yes;yes;no;no;yes;yes;no;1;2;2;1;3;5;14;6;9;0
GP;M;15;U;GT3;T;2;2;other;other;home;mother;1;1;0;no;yes;yes;no;yes;yes;yes;no;4;2;2;1;2;5;2;12;12;1
GP;M;15;U;GT3;T;4;2;health;services;other;mother;1;1;0;no;no;yes;no;yes;yes;yes;no;2;2;4;2;4;1;4;15;16;1
GP;M;16;U;LE3;A;3;4;services;other;home;mother;1;2;0;yes;yes;no;yes;yes;yes;yes;no;5;3;3;1;1;5;4;11;11;1
GP;M;16;U;GT3;T;4;4;teacher;teacher;home;mother;1;2;0;no;yes;yes;yes;yes;yes;yes;yes;4;4;5;5;5;5;16;10;12;1
GP;M;15;U;GT3;T;4;4;health;services;home;mother;1;2;0;no;yes;yes;no;no;yes;yes;no;5;4;2;3;4;5;0;9;11;1
GP;M;15;U;GT3;T;4;4;services;services;reputation;mother;2;2;0;no;yes;no;yes;yes;yes;yes;no;4;3;1;1;1;5;0;17;16;1
GP;M;15;R;GT3;T;4;3;teacher;at_home;course;mother;1;2;0;no;yes;no;yes;yes;yes;yes;yes;4;5;2;1;1;5;0;17;16;1
GP;M;15;U;LE3;T;3;3;other;other;course;mother;1;2;0;no;no;no;yes;no;yes;yes;no;5;3;2;1;1;2;0;8;10;1
GP;M;16;U;GT3;T;3;2;other;other;home;mother;1;1;0;no;yes;yes;no;no;yes;yes;no;5;4;3;1;1;5;0;12;14;1
GP;F;15;U;GT3;T;2;3;other;other;other;father;2;1;0;no;yes;no;yes;yes;yes;no;no;3;5;1;1;1;5;0;8;7;0
GP;M;15;U;LE3;T;4;3;teacher;services;home;mother;1;3;0;no;yes;no;yes;yes;yes;yes;no;5;4;3;1;1;4;2;15;16;1
GP;M;16;R;GT3;A;4;4;other;teacher;reputation;mother;2;3;0;no;yes;no;yes;yes;yes;yes;yes;2;4;3;1;1;5;7;15;16;1
```

student-matResultado2

(d) (5 pontos) Calcule o intervalo de confiança da acurácia para o 100 repetições de holdout 50/50 utilizando o classi cador 1-NN com distância Euclidiana.

```
accuracy = []

for i in range(100):
    treinoX, testeX, treinoY, testeY = train_test_split(x,y,test_size=0.50)
    knn = KNeighborsClassifier(n_neighbors=1, weights= "distance",metric="euclidean")
    knn.fit(treinoX,treinoY)
    y_pred = knn.predict(testeX)
    y_true = testeY
    accuracy.append(accuracy_score(y_true,y_pred))
```
1.8s

TAXAS DE ACERTO

```
[0.8333333333333334, 0.8383838383838383, 0.8080808080808081, 0.8131313131313131, 0.7777777777777778, 0.8383838383838383, 0.7929292929292929, 0.7929292929292929,
0.7727272727272727, 0.8080808080808081, 0.8131313131313131, 0.7828282828282829, 0.797979797979798, 0.7777777777777778, 0.8131313131313131, 0.8282828282828283,
0.8232323232323232, 0.7727272727272727, 0.8232323232323232, 0.8181818181818182, 0.8333333333333334, 0.8181818181818182, 0.803030303030303, 0.8181818181818182,
0.7929292929292929, 0.803030303030303, 0.8232323232323232, 0.8131313131313131, 0.8131313131313131, 0.7727272727272727, 0.7878787878787878, 0.8333333333333334,
0.8787878787878788, 0.7626262626262627, 0.8131313131313131, 0.7676767676767676, 0.8535353535353535, 0.7777777777777778, 0.7727272727272727, 0.8080808080808081,
0.797979797979798, 0.8232323232323232, 0.7929292929292929, 0.8333333333333334, 0.8181818181818182, 0.7929292929292929, 0.803030303030303, 0.7878787878787878,
0.803030303030303, 0.7777777777777778, 0.8080808080808081, 0.8131313131313131, 0.8131313131313131, 0.7676767676767676, 0.8080808080808081, 0.7878787878787878,
0.8232323232323232, 0.8080808080808081, 0.797979797979798, 0.8131313131313131, 0.8232323232323232, 0.8232323232323232, 0.7828282828282829, 0.8181818181818182,
0.7878787878787878, 0.7727272727272727, 0.7878787878787878, 0.8080808080808081, 0.7777777777777778, 0.803030303030303, 0.7525252525252525, 0.8080808080808081,
0.8383838383838383, 0.7777777777777778, 0.8232323232323232, 0.7929292929292929, 0.7878787878787878, 0.7878787878787878, 0.7929292929292929, 0.8282828282828283,
0.7727272727272727, 0.7828282828282829, 0.7878787878787878, 0.797979797979798, 0.8232323232323232, 0.7525252525252525, 0.803030303030303, 0.7777777777777778,
0.8232323232323232, 0.7626262626262627, 0.7777777777777778, 0.7575757575757576, 0.8080808080808081, 0.8383838383838383, 0.797979797979798, 0.797979797979798,
0.8181818181818182, 0.7676767676767676, 0.7777777777777778, 0.8080808080808081]
```

INTERVALO DE CONFIANÇA:

```
INTERVALO CONFIANÇA [0.76;0.85]
```

2. Utilizando a base Forest Fires . archive.ics.uci.edu/ml/datasets/Forest+Fires

   (a) (5 pontos) Indique a forma mais adequada de converter para numéricos cada um dos atributos da base.

   Usando a função df.map para converter os atributos de String para numéricos.

   (b) (10 pontos) Realize a conversão da base conforme a resposta indicada.

```python
import pandas as pd
import numpy as np

df = pd.read_csv('forestfires.csv', sep=',')

df['month'] = df['month'].map({
    'jan': 1, 'feb': 2, 'mar': 3, 'apr': 4, 'may': 5, 'jun': 6,
    'jul': 7, 'aug': 8, 'sep': 9, 'oct': 10, 'nov': 11, 'dec': 12
})


df['day'] = df['day'].map({
    'mon': 1, 'tue': 2, 'wed': 3, 'thu': 4,
    'fri': 5, 'sat': 6, 'sun': 7
})
```

Base pré-conversão:

X	Y	month	day	FFMC	DMC	DC	ISI	temp	RH	wind	rain	area
7	5	mar	fri	86.2	26.2	94.3	5.1	8.2	51	6.7	0	0
7	4	oct	tue	90.6	35.4	669.1	6.7	18	33	0.9	0	0
7	4	oct	sat	90.6	43.7	686.9	6.7	14.6	33	1.3	0	0
8	6	mar	fri	91.7	33.3	77.5	9	8.3	97	4	0.2	0
8	6	mar	sun	89.3	51.3	102.2	9.6	11.4	99	1.8	0	0
8	6	aug	sun	92.3	85.3	488	14.7	22.2	29	5.4	0	0
8	6	aug	mon	92.3	88.9	495.6	8.5	24.1	27	3.1	0	0
8	6	aug	mon	91.5	145.4	608.2	10.7	8	86	2.2	0	0
8	6	sep	tue	91	129.5	692.6	7	13.1	63	5.4	0	0
7	5	sep	sat	92.5	88	698.6	7.1	22.8	40	4	0	0
7	5	sep	sat	92.5	88	698.6	7.1	17.8	51	7.2	0	0
7	5	sep	sat	92.8	73.2	713	22.6	19.3	38	4	0	0
6	5	aug	fri	63.5	70.8	665.3	0.8	17	72	6.7	0	0
6	5	sep	mon	90.9	126.5	686.5	7	21.3	42	2.2	0	0
6	5	sep	wed	92.9	133.3	699.6	9.2	26.4	21	4.5	0	0
6	5	sep	fri	93.3	141.2	713.9	13.9	22.9	44	5.4	0	0
5	5	mar	sat	91.7	35.8	80.8	7.8	15.1	27	5.4	0	0
8	5	oct	mon	84.9	32.8	664.2	3	16.7	47	4.9	0	0
6	4	mar	wed	89.2	27.9	70.8	6.3	15.9	35	4	0	0
6	4	apr	sat	86.3	27.4	97.1	5.1	9.3	44	4.5	0	0
6	4	sep	tue	91	129.5	692.6	7	18.3	40	2.7	0	0
5	4	sep	mon	91.8	78.5	724.3	9.2	19.1	38	2.7	0	0
7	4	jun	sun	94.3	96.3	200	56.1	21	44	4.5	0	0
7	4	aug	sat	90.2	110.9	537.4	6.2	19.5	43	5.8	0	0
7	4	aug	sat	93.5	139.4	594.2	20.3	23.7	32	5.8	0	0
7	4	aug	sun	91.4	142.4	601.4	10.6	16.3	60	5.4	0	0
7	4	sep	fri	92.4	117.9	668	12.2	19	34	5.8	0	0
7	4	sep	mon	90.9	126.5	686.5	7	19.4	48	1.3	0	0
6	3	sep	sat	93.4	145.4	721.4	8.1	30.2	24	2.7	0	0
6	3	sep	sun	93.5	149.3	728.6	8.1	22.8	39	3.6	0	0
6	3	sep	fri	94.3	85.1	692.3	15.9	25.4	24	3.6	0	0
6	3	sep	mon	88.6	91.8	709.9	7.1	11.2	78	7.6	0	0
6	3	sep	fri	88.6	69.7	706.8	5.8	20.6	37	1.8	0	0
6	3	sep	sun	91.7	75.6	718.3	7.8	17.7	39	3.6	0	0
6	3	sep	mon	91.8	78.5	724.3	9.2	21.2	32	2.7	0	0
6	3	sep	tue	90.3	80.7	730.2	6.3	18.2	62	4.5	0	0
6	3	oct	tue	90.6	35.4	669.1	6.7	21.7	24	4.5	0	0
7	4	oct	fri	90	41.5	682.6	8.7	11.3	60	5.4	0	0

forestfires

Base pós-conversão:

```
X;Y;month;day;FFMC;DMC;DC;ISI;temp;RH;wind;rain;area
7;5;3;5;86.2;26.2;94.3;5.1;8.2;51;6.7;0.0;0.0
7;4;10;2;90.6;35.4;669.1;6.7;18.0;33;0.9;0.0;0.0
7;4;10;6;90.6;43.7;686.9;6.7;14.6;33;1.3;0.0;0.0
8;6;3;5;91.7;33.3;77.5;9.0;8.3;97;4.0;0.2;0.0
8;6;3;7;89.3;51.3;102.2;9.6;11.4;99;1.8;0.0;0.0
8;6;8;7;92.3;85.3;488.0;14.7;22.2;29;5.4;0.0;0.0
8;6;8;1;92.3;88.9;495.6;8.5;24.1;27;3.1;0.0;0.0
8;6;8;1;91.5;145.4;608.2;10.7;8.0;86;2.2;0.0;0.0
8;6;9;2;91.0;129.5;692.6;7.0;13.1;63;5.4;0.0;0.0
7;5;9;6;92.5;88.0;698.6;7.1;22.8;40;4.0;0.0;0.0
7;5;9;6;92.5;88.0;698.6;7.1;17.8;51;7.2;0.0;0.0
7;5;9;6;92.8;73.2;713.0;22.6;19.3;38;4.0;0.0;0.0
6;5;8;5;63.5;70.8;665.3;0.8;17.0;72;6.7;0.0;0.0
6;5;9;1;90.9;126.5;686.5;7.0;21.3;42;2.2;0.0;0.0
6;5;9;3;92.9;133.3;699.6;9.2;26.4;21;4.5;0.0;0.0
6;5;9;5;93.3;141.2;713.9;13.9;22.9;44;5.4;0.0;0.0
5;5;3;6;91.7;35.8;80.8;7.8;15.1;27;5.4;0.0;0.0
8;5;10;1;84.9;32.8;664.2;3.0;16.7;47;4.9;0.0;0.0
6;4;3;3;89.2;27.9;70.8;6.3;15.9;35;4.0;0.0;0.0
6;4;4;6;86.3;27.4;97.1;5.1;9.3;44;4.5;0.0;0.0
6;4;9;2;91.0;129.5;692.6;7.0;18.3;40;2.7;0.0;0.0
5;4;9;1;91.8;78.5;724.3;9.2;19.1;38;2.7;0.0;0.0
7;4;6;7;94.3;96.3;200.0;56.1;21.0;44;4.5;0.0;0.0
7;4;8;6;90.2;110.9;537.4;6.2;19.5;43;5.8;0.0;0.0
7;4;8;6;93.5;139.4;594.2;20.3;23.7;32;5.8;0.0;0.0
7;4;8;7;91.4;142.4;601.4;10.6;16.3;60;5.4;0.0;0.0
7;4;9;5;92.4;117.9;668.0;12.2;19.0;34;5.8;0.0;0.0
7;4;9;1;90.9;126.5;686.5;7.0;19.4;48;1.3;0.0;0.0
6;3;9;6;93.4;145.4;721.4;8.1;30.2;24;2.7;0.0;0.0
6;3;9;7;93.5;149.3;728.6;8.1;22.8;39;3.6;0.0;0.0
6;3;9;5;94.3;85.1;692.3;15.9;25.4;24;3.6;0.0;0.0
6;3;9;1;88.6;91.8;709.9;7.1;11.2;78;7.6;0.0;0.0
6;3;9;5;88.6;69.7;706.8;5.8;20.6;37;1.8;0.0;0.0
6;3;9;7;91.7;75.6;718.3;7.8;17.7;39;3.6;0.0;0.0
6;3;9;1;91.8;78.5;724.3;9.2;21.2;32;2.7;0.0;0.0
6;3;9;2;90.3;80.7;730.2;6.3;18.2;62;4.5;0.0;0.0
6;3;10;2;90.6;35.4;669.1;6.7;21.7;24;4.5;0.0;0.0
7;4;10;5;90.0;41.5;682.6;8.7;11.3;60;5.4;0.0;0.0
```

forestfiresResultado

3. Utilizando a base Car Evaluation . archive.ics.uci.edu/ml/datasets/Car+Evaluation

(a) (5 pontos) Indique a forma mais adequada de converter para numéricos cada um dos atributos da base.

Utilizando a função df.map para converter cada atributo string dessa base para numéricos. Porém, antes de começar a conversão, foi preciso criar uma linha com os nomes de cada coluna durante a leitura do arquivo.

(b) (10 pontos) Realize a conversão da base conforme a resposta indicada.

Para os atributos “5more” e “more”, o valor numérico usado foi 55.

```python
import pandas as pd
import collections


df = pd.read_csv(r'c:\Users\David\Desktop\RP-2022.1\semana 5\car.data.csv', header=None,
                 names=['price','maint','doors','persons','lug_boot','safety','value'])

df['price'] = df['price'].map({'low': 0, 'mid': 1, 'high': 2, 'vhigh': 3})
df['maint'] = df['maint'].map({'low': 0, 'mid': 1, 'high': 2, 'vhigh': 3})
df['doors'] = df['doors'].map({'2': 2, '3': 3, '4': 4, '5more': 55})
df['persons'] = df['persons'].map({'2': 2, '4': 4, 'more': 55})
df['lug_boot'] = df['lug_boot'].map({'small': 0, 'med': 1, 'big': 2,})
df['safety'] = df['safety'].map({'low': 0, 'mid': 1, 'high': 2,})
df['value'] = df['value'].map({'unacc': 0, 'acc': 1, 'good': 2, 'vgood': 3})

df.to_csv(r'c:\Users\David\Desktop\RP-2022.1\semana 5\car.dataResultado.csv', sep=';', index=False)
```

Base pré conversão:

vhigh	vhigh	2	2	small	low	unacc
vhigh	vhigh	2	2	small	med	unacc
vhigh	vhigh	2	2	small	high	unacc
vhigh	vhigh	2	2	med	low	unacc
vhigh	vhigh	2	2	med	med	unacc
vhigh	vhigh	2	2	med	high	unacc
vhigh	vhigh	2	2	big	low	unacc
vhigh	vhigh	2	2	big	med	unacc
vhigh	vhigh	2	2	big	high	unacc
vhigh	vhigh	2	4	small	low	unacc
vhigh	vhigh	2	4	small	med	unacc
vhigh	vhigh	2	4	small	high	unacc
vhigh	vhigh	2	4	med	low	unacc
vhigh	vhigh	2	4	med	med	unacc
vhigh	vhigh	2	4	med	high	unacc
vhigh	vhigh	2	4	big	low	unacc
vhigh	vhigh	2	4	big	med	unacc
vhigh	vhigh	2	4	big	high	unacc
vhigh	vhigh	2	more	small	low	unacc
vhigh	vhigh	2	more	small	med	unacc
vhigh	vhigh	2	more	small	high	unacc
vhigh	vhigh	2	more	med	low	unacc
vhigh	vhigh	2	more	med	med	unacc
vhigh	vhigh	2	more	med	high	unacc
vhigh	vhigh	2	more	big	low	unacc
vhigh	vhigh	2	more	big	med	unacc
vhigh	vhigh	2	more	big	high	unacc
vhigh	vhigh	3	2	small	low	unacc
vhigh	vhigh	3	2	small	med	unacc
vhigh	vhigh	3	2	small	high	unacc
vhigh	vhigh	3	2	med	low	unacc
vhigh	vhigh	3	2	med	med	unacc
vhigh	vhigh	3	2	med	high	unacc
vhigh	vhigh	3	2	big	low	unacc
vhigh	vhigh	3	2	big	med	unacc
vhigh	vhigh	3	2	big	high	unacc
vhigh	vhigh	3	4	small	low	unacc
vhigh	vhigh	3	4	small	med	unacc
vhigh	vhigh	3	4	small	high	unacc

Base pós-conversão:

price;maint;doors;persons;lug_boot;safety;value			
3.0;3.0;2;2;0;0.0;0			
3.0;3.0;2;2;0;;0			
3.0;3.0;2;2;0;2.0;0			
3.0;3.0;2;2;1;0.0;0			
3.0;3.0;2;2;1;;0			
3.0;3.0;2;2;1;2.0;0			
3.0;3.0;2;2;2;0.0;0			
3.0;3.0;2;2;2;;0			
3.0;3.0;2;2;2;2.0;0			
3.0;3.0;2;4;0;0.0;0			
3.0;3.0;2;4;0;;0			
3.0;3.0;2;4;0;2.0;0			
3.0;3.0;2;4;1;0.0;0			
3.0;3.0;2;4;1;;0			
3.0;3.0;2;4;1;2.0;0			
3.0;3.0;2;4;2;0.0;0			
3.0;3.0;2;4;2;;0			
3.0;3.0;2;4;2;2.0;0			
3.0;3.0;2;55;0;0.0;0			
3.0;3.0;2;55;0;;0			
3.0;3.0;2;55;0;2.0;0			
3.0;3.0;2;55;1;0.0;0			
3.0;3.0;2;55;1;;0			
3.0;3.0;2;55;1;2.0;0			
3.0;3.0;2;55;2;0.0;0			
3.0;3.0;2;55;2;;0			
3.0;3.0;2;55;2;2.0;0			
3.0;3.0;3;2;0;0.0;0			
3.0;3.0;3;2;0;;0			
3.0;3.0;3;2;0;2.0;0			
3.0;3.0;3;2;1;0.0;0			
3.0;3.0;3;2;1;;0			
3.0;3.0;3;2;1;2.0;0			
3.0;3.0;3;2;2;0.0;0			
3.0;3.0;3;2;2;;0			
3.0;3.0;3;2;2;2.0;0			
3.0;3.0;3;4;0;0.0;0			
3.0;3.0;3;4;0;;0			

4. A base Heart Disease (hungarian) possui alguns valores de atributos omissos. Realize o experimento descrito abaixo utilizando o classi cador 1-NN. Divida a base em treino (90%) e teste (10%) de forma estrati cada. Calcule o intervalo de confiança para a taxa de acerto do classi cador utilizando 100 repetições deste experimento.

   https://archive.ics.uci.edu/ml/machine-learning-databases/heart-disease/processed.hungarian.data

   https://archive.ics.uci.edu/ml/datasets/Heart+Disease

   A primeira etapa foi transformar os dados que estavam com “?” para “null”

```
df.isnull().sum()
```

✓ 0.3s

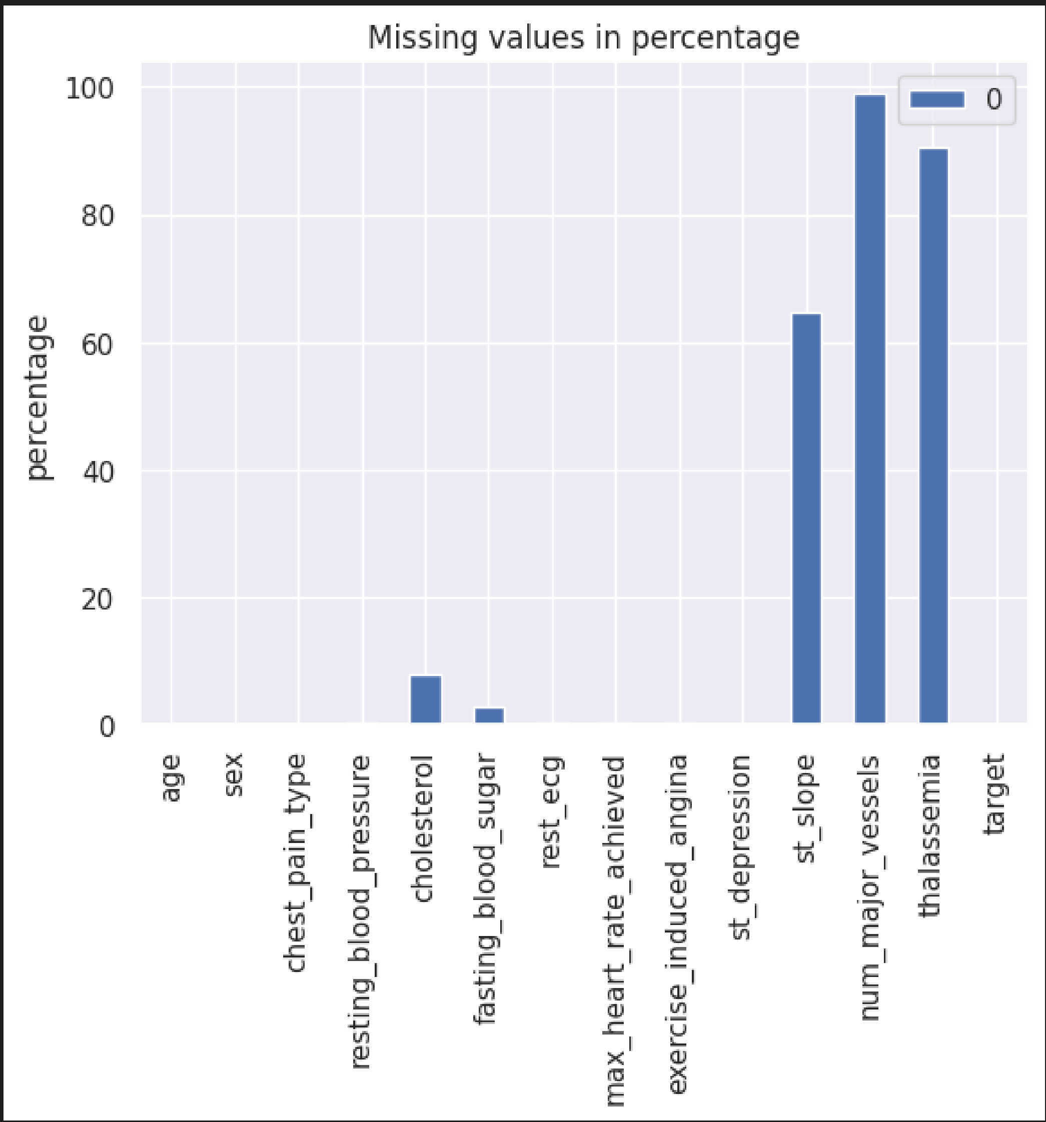
```
age                          0
sex                          0
chest_pain_type              0
resting_blood_pressure       1
cholesterol                 23
fasting_blood_sugar          8
rest_ecg                     1
max_heart_rate_achieved      1
exercise_induced_angina      1
st_depression                0
st_slope                   190
num_major_vessels          291
thalassemia                266
target                       0
dtype: int64
```

Porcentagem de valores faltantes:

```python
sns.set()
miss_vals = pd.DataFrame(df.isnull().sum() / len(df) * 100)
miss_vals.plot(kind='bar',title='Missing values in percentage',ylabel='percentage')
```

✓ 1.9s

```
<AxesSubplot: title={'center': 'Missing values in percentage'}, ylabel='percentage'>
```



```python
print(f'Number of null values before: {df.resting_blood_pressure.isnull().sum()}')

# A classe SimpleImputer funciona com o método fit_transform que executa os métodos fit() e transform() em uma única linha.
imp = SimpleImputer(strategy='mean')

# Preenche os valores ausentes com a média da coluna
df['resting_blood_pressure'] = imp.fit_transform(df[['resting_blood_pressure']])

print(f'Number of null values after: {df.resting_blood_pressure.isnull().sum()}')
```

0.2s

```
Number of null values before: 1
Number of null values after: 0
```

```python
# função para definir os parâmetros a serem usados na classe SimpleImputer().
def get_parameters(df):
    parameters = {}
    for col in df.columns[df.isnull().any()]:

        if df[col].dtype == 'float64' or df[col].dtype == 'int64' or df[col].dtype =='int32':
            strategy = 'mean'
        else:
            strategy = 'most_frequent'
        missing_values = df[col][df[col].isnull()].values[0]
        parameters[col] = {'missing_values':missing_values, 'strategy':strategy}
    return parameters

parameters = get_parameters(df)
```

0.3s

```python
# percorra cada coluna para transformá-la.
for col, param in parameters.items():
    missing_values = param['missing_values']
    strategy = param['strategy']
    imp = SimpleImputer(missing_values=missing_values,strategy=strategy)
    df[col] = imp.fit_transform(df[[col]])

df.isnull().sum()
```

0.3s

```
age                        0
sex                        0
chest_pain_type            0
resting_blood_pressure     0
cholesterol                0
fasting_blood_sugar        0
rest_ecg                   0
max_heart_rate_achieved    0
exercise_induced_angina    0
st_depression              0
st_slope                   0
num_major_vessels          0
thalassemia                0
target                     0
dtype: int64
```

100 taxas de acerto:

```python
x = np.array(df[df.columns[0:3:]])
y = np.array(df[df.columns[-1::]]).flatten()

skf = StratifiedKFold(n_splits = 100)

classificador = KNeighborsClassifier(n_neighbors=1)

array_taxas = []

for train_index, test_index in skf.split(x,y):
    x_train, x_test, y_train, y_test = train_test_split (x, y, test_size=0.1)

    classificador.fit(x_train, y_train)
    taxas = classificador.score(x_test, y_test)
    array_taxas.append(taxas)
    y_pred = classificador.predict(x_test)

print(array_taxas)
```

```
0.6s
[0.5666666666666667, 0.7666666666666667, 0.7, 0.8333333333333334, 0.6, 0.6666666666666666, 0.5666666666666667, 0.6, 0.4666666666666667, 0.6333333333333333, 0.7333333333333333,
0.8666666666666667, 0.9, 0.6333333333333333, 0.6333333333333333, 0.7, 0.6333333333333333, 0.6666666666666666, 0.7, 0.7, 0.5666666666666667, 0.7, 0.7666666666666667, 0.5333333333333333,
0.6666666666666666, 0.8, 0.8, 0.6, 0.6666666666666666, 0.5333333333333333, 0.8, 0.7, 0.6, 0.6333333333333333, 0.6, 0.7333333333333333, 0.6, 0.7, 0.6333333333333333, 0.6333333333333333,
0.7333333333333333, 0.7333333333333333, 0.8, 0.7333333333333333, 0.7666666666666667, 0.7666666666666667, 0.6666666666666666, 0.6666666666666666, 0.5, 0.6333333333333333,
0.7333333333333333, 0.7, 0.7, 0.6, 0.5666666666666667, 0.7333333333333333, 0.7, 0.7333333333333333, 0.7, 0.9, 0.6333333333333333, 0.7333333333333333, 0.5666666666666667, 0.7,
0.43333333333333335, 0.6666666666666666, 0.8333333333333334, 0.7666666666666667, 0.5666666666666667, 0.6, 0.6333333333333333, 0.7333333333333333, 0.6666666666666666, 0.7,
0.5666666666666667, 0.6333333333333333, 0.7, 0.7, 0.5, 0.5666666666666667, 0.6666666666666666, 0.7333333333333333, 0.6, 0.7333333333333333, 0.8333333333333334, 0.7, 0.6666666666666666,
0.6666666666666666, 0.7666666666666667, 0.7, 0.8666666666666667, 0.5666666666666667, 0.7, 0.7, 0.6666666666666666, 0.7666666666666667, 0.8, 0.6666666666666666, 0.5666666666666667,
0.6333333333333333]
```

```
intervalo de confiança:
(0.4893494813419292, 0.8673171853247377)
```

(a) (10 pontos) Preencha os valores omissos no conjunto de treino.

(b) (10 pontos) Preencha os valores omissos no conjunto de teste utilizando o método e os valores de nidos para o conjunto de treino.

5. Utilizando a base de dados Wine https://archive.ics.uci.edu/ml/datasets/wine, para cada um dos casos abaixo, realize 100 repetições de Holdout 50/50 e calcule o intervalo de confiança da acurácia utilizando o classificador 1-NN com distância Euclidiana. Realize testes de hipótese por sobreposição dos intervalos de confiança comparando os pré-processamentos de cada um dos casos abaixo com a base de dados original:

(a) (10 pontos) Com todas as características ajustadas para o intervalo [0,1].

VALORES ANTES:

```
.. Output exceeds the size limit. Open the full output data in a text editor
     Class  Alcohol  Malic_acid   Ash  Alcalinity_of_ash  Magnesium  \
0        1    14.23        1.71  2.43               15.6        127
1        1    13.20        1.78  2.14               11.2        100
2        1    13.16        2.36  2.67               18.6        101
3        1    14.37        1.95  2.50               16.8        113
4        1    13.24        2.59  2.87               21.0        118
..     ...      ...         ...   ...                ...        ...
173      3    13.71        5.65  2.45               20.5         95
174      3    13.40        3.91  2.48               23.0        102
175      3    13.27        4.28  2.26               20.0        120
176      3    13.17        2.59  2.37               20.0        120
177      3    14.13        4.10  2.74               24.5         96

     Total_phenols  Flavanoids  Nonflavanoid_phenols  Proanthocyanins  \
0             2.80        3.06                  0.28             2.29
1             2.65        2.76                  0.26             1.28
2             2.80        3.24                  0.30             2.81
3             3.85        3.49                  0.24             2.18
4             2.80        2.69                  0.39             1.82
..             ...         ...                   ...              ...
173           1.68        0.61                  0.52             1.06
174           1.80        0.75                  0.43             1.41
175           1.59        0.69                  0.43             1.35
176           1.65        0.68                  0.53             1.46
177           2.05        0.76                  0.56             1.35
...
176             9.30  0.60                           1.62      840
177             9.20  0.61                           1.60      560

[178 rows x 14 columns]
```

MinMaxScaler() coloca todos os valores numéricos em uma escala de 0 a 1.

```python
# Selecionar colunas numéricas
num_cols = df.select_dtypes(include=['int64', 'float64', 'int32']).columns
print(num_cols)

# Valores ausentes
for col in num_cols:
    fill_value = df[col].mean()
    df[col].fillna(fill_value, inplace=True)

minmax = MinMaxScaler()
df[num_cols] = minmax.fit_transform(df[num_cols])
df[num_cols]
```

OU TAMBÉM Reescala para [0,1]

```python
cont= 0
for i in df.columns:
    if(cont > 0):
        for j in range(178):
            #Vnovo = (Vatual - menor) / maior - menor
            result = (df[i][j] - df[i].min())/(df[i].max() - df[i].min())
            df[i][j] = result
    cont +=1
df
```

VALORES DEPOIS:

	Class	Alcohol	Malic_acid	Ash	Alcalinity_of_ash	Magnesium	Total_phenols	Flavanoids	Nonflavanoid_phenols	Proanthocyanins	Color_intensity	Hue	AOD2
0	0.0	0.842105	0.191700	0.572193	0.257732	0.619565	0.627586	0.573840	0.283019	0.593060	0.372014	0.455285	
1	0.0	0.571053	0.205534	0.417112	0.030928	0.326087	0.575862	0.510549	0.245283	0.274448	0.264505	0.463415	
2	0.0	0.560526	0.320158	0.700535	0.412371	0.336957	0.627586	0.611814	0.320755	0.757098	0.375427	0.447154	
3	0.0	0.878947	0.239130	0.609626	0.319588	0.467391	0.989655	0.664557	0.207547	0.558360	0.556314	0.308943	
4	0.0	0.581579	0.365613	0.807487	0.536082	0.521739	0.627586	0.495781	0.490566	0.444795	0.259386	0.455285	
...	...	...	...	...	...	...	...	...	...	...	...	...	
173	1.0	0.705263	0.970356	0.582888	0.510309	0.271739	0.241379	0.056962	0.735849	0.205047	0.547782	0.130081	
174	1.0	0.623684	0.626482	0.598930	0.639175	0.347826	0.282759	0.086498	0.566038	0.315457	0.513652	0.178862	
175	1.0	0.589474	0.699605	0.481283	0.484536	0.543478	0.210345	0.073840	0.566038	0.296530	0.761092	0.089431	
176	1.0	0.563158	0.365613	0.540107	0.484536	0.543478	0.231034	0.071730	0.754717	0.331230	0.684300	0.097561	
177	1.0	0.815789	0.664032	0.737968	0.716495	0.282609	0.368966	0.088608	0.811321	0.296530	0.675768	0.105691	

178 rows × 14 columns

INTERVALO DE CONFIANÇA

```
[0.83;0.95]
```

(b) (10 pontos) Com todas as características ajustadas para ter média zero e desvio padrão igual a um.

StandardScaler() coloca todos os valores numéricos em uma escala onde a média é igual a 0 e o desvio padrão é igual a 1.

```python
ss = StandardScaler()
df[num_cols] = ss.fit_transform(df[num_cols])
df[num_cols].describe()
```

✓ 0.2s

Pytho

	0	1	2	3	4	5	6	7	8	9	10	11
count	1.780000e+02	1.780000e+02	1.780000e+02	1.780000e+02	1.780000e+02	1.780000e+02	1.780000e+02	1.780000e+02	1.780000e+02	1.780000e+02	1.780000e+02	1.780000e+02
mean	1.596725e-16	-4.390994e-16	-3.991813e-17	1.921060e-16	-3.991813e-17	-5.987720e-17	-7.983626e-17	1.596725e-16	-3.193450e-16	3.991813e-17	1.496930e-16	2.395088e-16
std	1.002821e+00	1.002821e+00	1.002821e+00	1.002821e+00	1.002821e+00	1.002821e+00	1.002821e+00	1.002821e+00	1.002821e+00	1.002821e+00	1.002821e+00	1.002821e+00
min	-1.213944e+00	-2.434235e+00	-1.432983e+00	-3.679162e+00	-2.671018e+00	-2.088255e+00	-2.107246e+00	-1.695971e+00	-1.868234e+00	-2.069034e+00	-1.634288e+00	-2.094732e+00
25%	-1.213944e+00	-7.882448e-01	-6.587486e-01	-5.721225e-01	-6.891372e-01	-8.244151e-01	-8.854682e-01	-8.275393e-01	-7.401412e-01	-5.972835e-01	-7.951025e-01	-7.675624e-01
50%	7.996036e-02	6.099988e-02	-4.231120e-01	-2.382132e-02	1.518295e-03	-1.222817e-01	9.595986e-02	1.061497e-01	-1.760948e-01	-6.289785e-02	-1.592246e-01	3.312687e-02
75%	1.373864e+00	8.361286e-01	6.697929e-01	6.981085e-01	6.020883e-01	5.096384e-01	8.089974e-01	8.490851e-01	6.095413e-01	6.291754e-01	4.939560e-01	7.131644e-01
max	1.373864e+00	2.259772e+00	3.109192e+00	3.156325e+00	3.154511e+00	4.371372e+00	2.539515e+00	3.062832e+00	2.402403e+00	3.485073e+00	3.435432e+00	3.301694e+00

count	1.780000e+02	1.780000e+02	1.780000e+02	1.780000e+02	1.780000e+02	1.780000e+02	1.780000e+02	1.780000e+02	1.780000e+02	1.780000e+02	1.780000e+02	1.780000e+02
mean	1.596725e-16	-4.390994e-16	-3.991813e-17	1.921060e-16	-3.991813e-17	-5.987720e-17	-7.983626e-17	1.596725e-16	-3.193450e-16	3.991813e-17	1.496930e-16	2.395088e-16
std	1.002821e+00	1.002821e+00	1.002821e+00	1.002821e+00	1.002821e+00	1.002821e+00	1.002821e+00	1.002821e+00	1.002821e+00	1.002821e+00	1.002821e+00	1.002821e+00

A média não parece igual a 0, porém, **1.596725e-16** é igual a **0,0000000000000001596725**. Isso é tão próximo de 0 que pode ser considerado igual a 0. O mesmo acontece com o desvio padrão que é tão próximo de 1 que pode ser considerado igual a 1.