Ciência de dados - Dados abertos da UFRN: ITP pré e pós pandemia

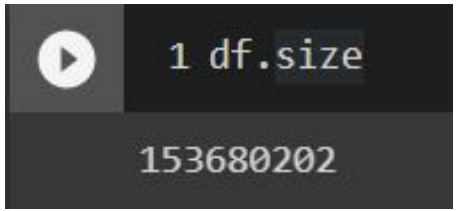
Abmael Dantas Gomes, Addan Felipe Neri Andrade
e Jeová Henrique Linhares

recap?

O tema proposto escolhido por nós foi a análise pré e pós pandemia da disciplina de ITP do curso BTI da UFRN.

153.680.202

linhas na base antes de qualquer modificação



Relembrando a limpeza e filtragem dos dados

```
notas_imd_2018_02_merge[notas_imd_2018_02_merge["nivel_ensino"] == "GRADUAÇÃO"]
notas_imd_2018_01_merge[notas_imd_2018_01_merge["nivel_ensino"] == "GRADUAÇÃO"]
notas_imd_2020_01_merge[notas_imd_2020_01_merge["nivel_ensino"] == "GRADUAÇÃO"]
notas_imd_2019_02_merge[notas_imd_2019_02_merge["nivel_ensino"] == "GRADUAÇÃO"]
notas_imd_2019_01_merge[notas_imd_2019_01_merge["nivel_ensino"] == "GRADUAÇÃO"]
```

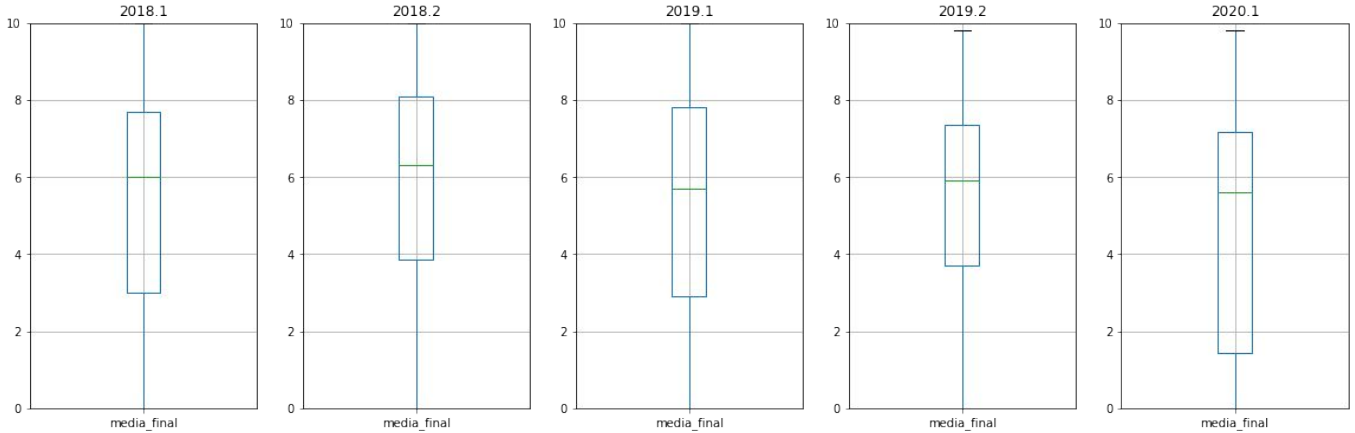
```
.loc[graduacao_imd_2018_02['nome_x'] == 'INTRODUÇÃO ÀS TÉCNICAS DE PROGRAMAÇÃO']
.loc[graduacao_imd_2018_01['nome_x'] == 'INTRODUÇÃO ÀS TÉCNICAS DE PROGRAMAÇÃO']
.loc[graduacao_imd_2020_01['nome_x'] == 'INTRODUÇÃO ÀS TÉCNICAS DE PROGRAMAÇÃO']
.loc[graduacao_imd_2019_02['nome_x'] == 'INTRODUÇÃO ÀS TÉCNICAS DE PROGRAMAÇÃO']
.loc[graduacao_imd_2019_01['nome_x'] == 'INTRODUÇÃO ÀS TÉCNICAS DE PROGRAMAÇÃO']
```

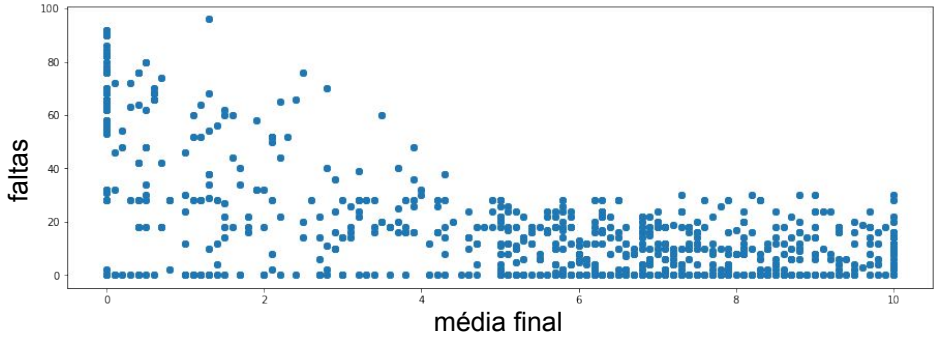
De 2019.1 até o atual: DIM0133 e IMD1012
Em 2018 : DIM0118.0 e IMD0012.0
Antes de 2018: IMD0012.0

Um pouco mais do tratamento dos dados (mudança de tipo e ajustes)

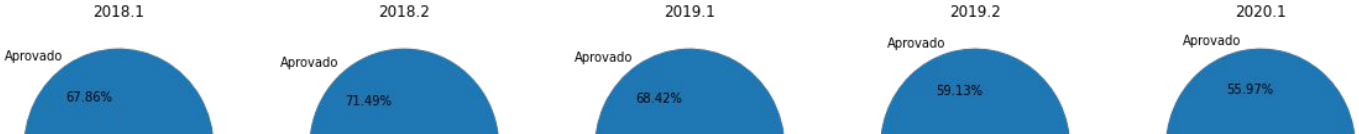
```
graduacao_imd_2018_02 = graduacao_imd_2018_02.replace({',': '.'}, regex=True)
graduacao_imd_2018_01 = graduacao_imd_2018_01.replace({',': '.'}, regex=True)
graduacao_imd_2020_01 = graduacao_imd_2020_01.replace({',': '.'}, regex=True)
graduacao_imd_2019_02 = graduacao_imd_2019_02.replace({',': '.'}, regex=True)
graduacao_imd_2019_01 = graduacao_imd_2019_01.replace({',': '.'}, regex=True)

graduacao_imd_2018_02.nota = graduacao_imd_2018_02.nota.astype(np.float64)
graduacao_imd_2018_01.nota = graduacao_imd_2018_01.nota.astype(np.float64)
graduacao_imd_2020_01.nota = graduacao_imd_2020_01.nota.astype(np.float64)
graduacao_imd_2019_02.nota = graduacao_imd_2019_02.nota.astype(np.float64)
graduacao_imd_2019_01.nota = graduacao_imd_2019_01.nota.astype(np.float64)

graduacao_imd_2018_02.media_final = graduacao_imd_2018_02.media_final.astype(np.float64)
graduacao_imd_2018_01.media_final = graduacao_imd_2018_01.media_final.astype(np.float64)
graduacao_imd_2020_01.media_final = graduacao_imd_2020_01.media_final.astype(np.float64)
graduacao_imd_2019_02.media_final = graduacao_imd_2019_02.media_final.astype(np.float64)
graduacao_imd_2019_01.media_final = graduacao_imd_2019_01.media_final.astype(np.float64)
```

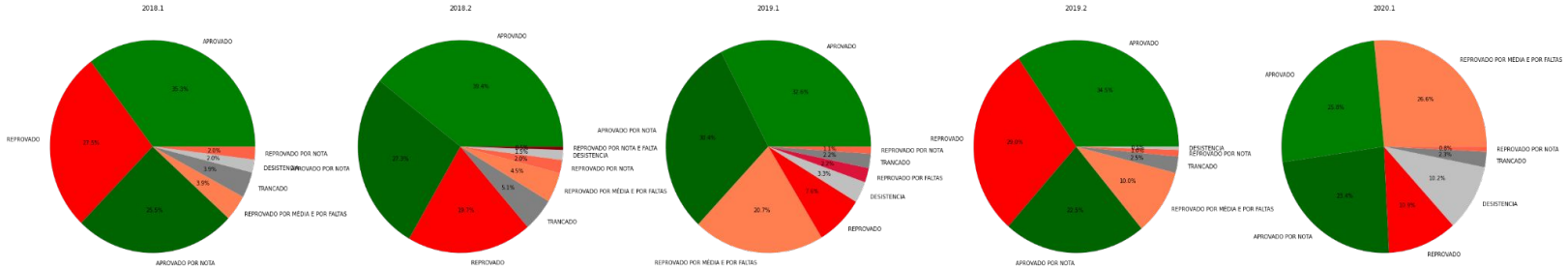
Análise exploratória (já apresentada)

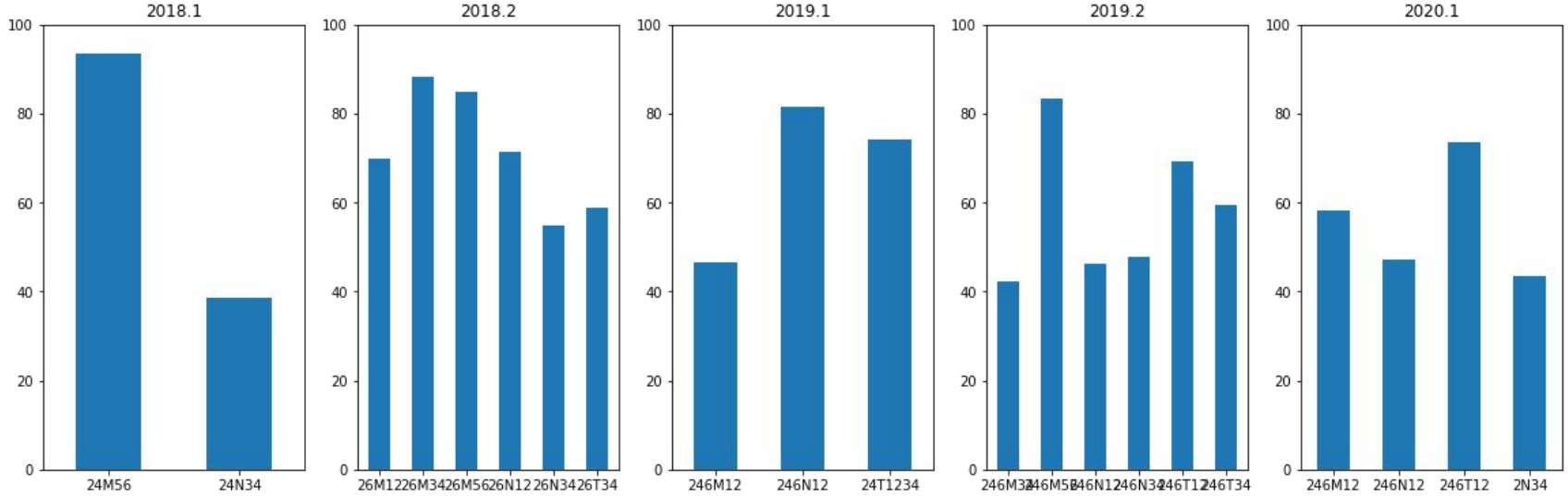




Análise exploratória (já apresentada)






2018.1
100
80
60
40
20
0
24M56
24N34
2018.2
26M12
26M34
26M56
26N12
26N34
26T34
2019.1
246M12
246N12
24T1234
2019.2
246M34
246M56
246N12
246N34
246T12
246T34
2020.1
246M12
246N12
246T12
2N34

Seleção de características

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 8537789 entries, 0 to 8537788
Data columns (total 18 columns):
 #   Column               Dtype
---  ------               -----
 0   id_turma             int64
 1   discente             object
 2   id_curso             float64
 3   unidade              float64
 4   nota                 object
 5   reposicao            object
 6   faltas_unidade       float64
 7   media_final          object
 8   numero_total_faltas  float64
 9   descricao            object
 10  nome_y               object
 11  codigo               object
 12  nivel_ensino         object
 13  nome_x               object
 14  capacidade_aluno     int64
 15  descricao_horario    object
 16  aprovado             int64
 17  semestre             object
dtypes: float64(4), int64(3), object(11)
memory usage: 1.1+ GB
```

```
dfc = df[['unidade', 'nota',
'reposicao', 'faltas_unidade'
'media_final', 'numero_total_
'descricao', 'nome_x', 'aprova
'aprovado', 'semestre']]
```

essas foram as características principais escolhidas para a versão final da base de dados para as análises desse projeto

Recortes:

```
dfc = df[['discente', 'unidade', 'nota', 'reposicao', 'faltas_unidade',
          'media_final', 'numero_total_faltas', 'descricao', 'nome_x', 'aprovado', 'semestre']]
dfc.head()
```

	discente	unidade	nota	reposicao	faltas_unidade	media_final	numero_total_faltas	descricao	nome_x	aprovado	semestre
0	fdfbc41af495cd2ed1ebfc0edac0a8b9	1.0	7,0	False	0.0	8,3	18.0	APROVADO	HISTORIA MODERNA II	1	2018.2
1	fdfbc41af495cd2ed1ebfc0edac0a8b9	2.0	9,0	False	0.0	8,3	18.0	APROVADO	HISTORIA MODERNA II	1	2018.2
2	fdfbc41af495cd2ed1ebfc0edac0a8b9	3.0	9,0	False	0.0	8,3	18.0	APROVADO	HISTORIA MODERNA II	1	2018.2
3	9fa6e0da8cb50d663b767a23a72ffb31	1.0	3,4	False	0.0	6,5	0.0	APROVADO POR NOTA	HISTORIA MODERNA II	1	2018.2
4	9fa6e0da8cb50d663b767a23a72ffb31	2.0	8,0	False	0.0	6,5	0.0	APROVADO POR NOTA	HISTORIA MODERNA II	1	2018.2

aparecem registros das disciplinas de todos os cursos, pois os recortes por curso, disciplina, nível de ensino são feitos depois (para termos o potencial de explorar e analisar outros cursos e/ou disciplinas)

Agrupamento e flags:

Algumas estratégias foram adotadas para fazer o agrupamento (“clusterizar” de certa forma) os alunos. Primeiramente era indispensável a noção de aprovação de uma forma mais direta, já que a nos dados abertos o status da matrícula é um objeto (string). Definimos uma forma de ter essa informação como dado booleano:

```
def aprovado(row):
    if row["descricao"] == "APROVADO" or row["descricao"] == "APROVADO POR NOTA":
        return 1
    else:
        return 0
```

Refinamento e melhora da base:

Outro desafio encontrado durante o desenvolvimento do projeto foi o de tornar a base mais fiel à realidade do curso/disciplina fazendo a remoção exclusivamente de alunos matematicamente incapazes de serem APRN na disciplina de acordo com o regulamento.

Art. 108. O estudante que realiza avaliação de reposição é considerado aprovado, quanto à avaliação de aprendizagem, se satisfaz um dos seguintes critérios:

I – tem média final igual ou superior a 7,0 (sete); ou

II – tem média final igual ou superior a 5,0 (cinco), com rendimento acadêmico igual ou superior a 3,0 (três) na avaliação de reposição.

Parágrafo único. O estudante que realiza avaliação de reposição e não atinge os critérios de aprovação definidos neste artigo é considerado reprovado.

Refinamento e melhora da base:

```
11 def matematicamentereprovado(row):
12   if row["aprovado"] == 0:
13     if row["media_final"] < 5:
14       if row["reposicao"] == True:
15         return 1;
16         # ele tem que sair
17       else:
18         return 0;
19     else:
20       return 1;
21       # ele tem que sair
22   else:
23     return 0;
```

Após os rotular as entradas na base com a coluna ‘reprovadomatematicamente’ foi feita e exclusão de 168 linhas:

```
1 filtrados.loc[filtrados['reprovadomatematic
```

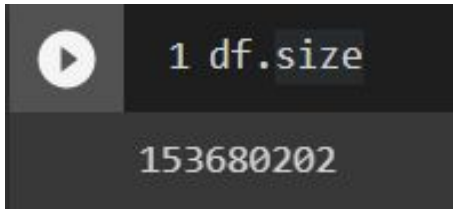
```
168
```

Esses seriam estudantes que, matematicamente, não conseguiriam alcançar APRN. Como não existe a coluna “nota_reposição” fiz a verificação se:

- Reprovado
- Média < 5
- Se fez reposição e foi reprovado, não obteve 3
- Então matematicamente não conseguiria APRN

153.680.202

linhas na base antes de qualquer modificação



2148

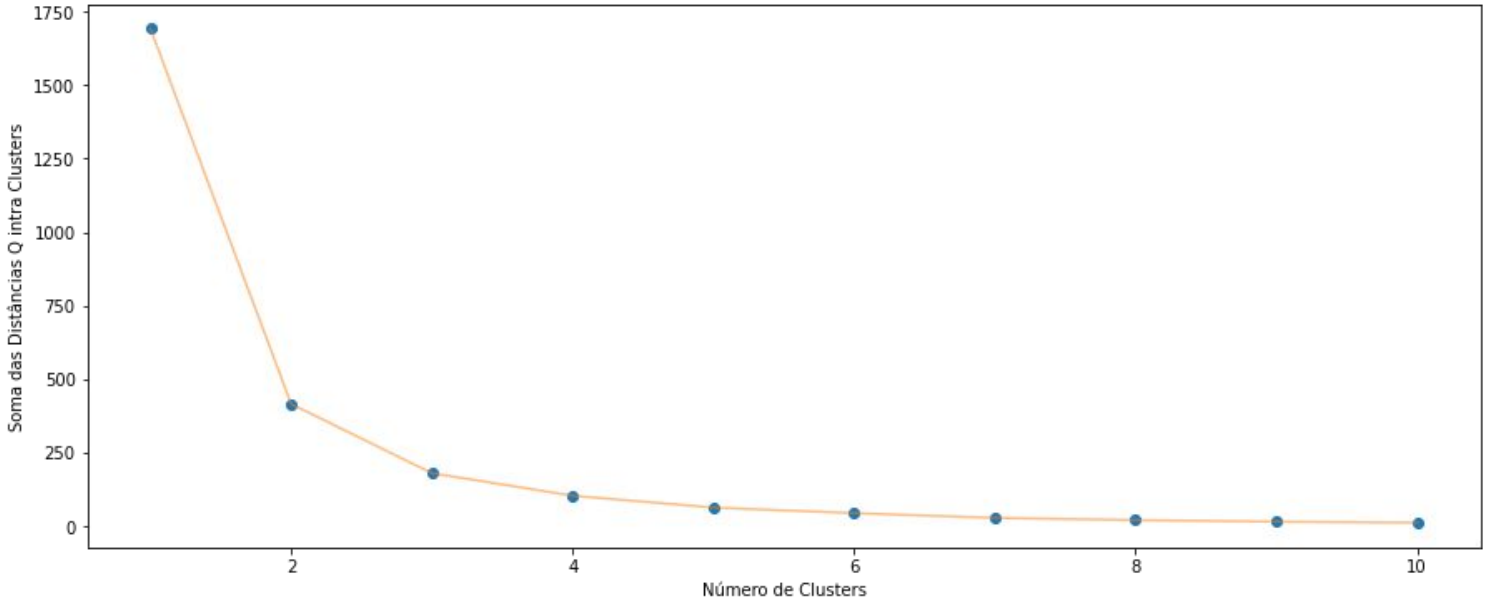
linhas na base depois de filtros e limpezas

```
1 print(base_final.size)
```

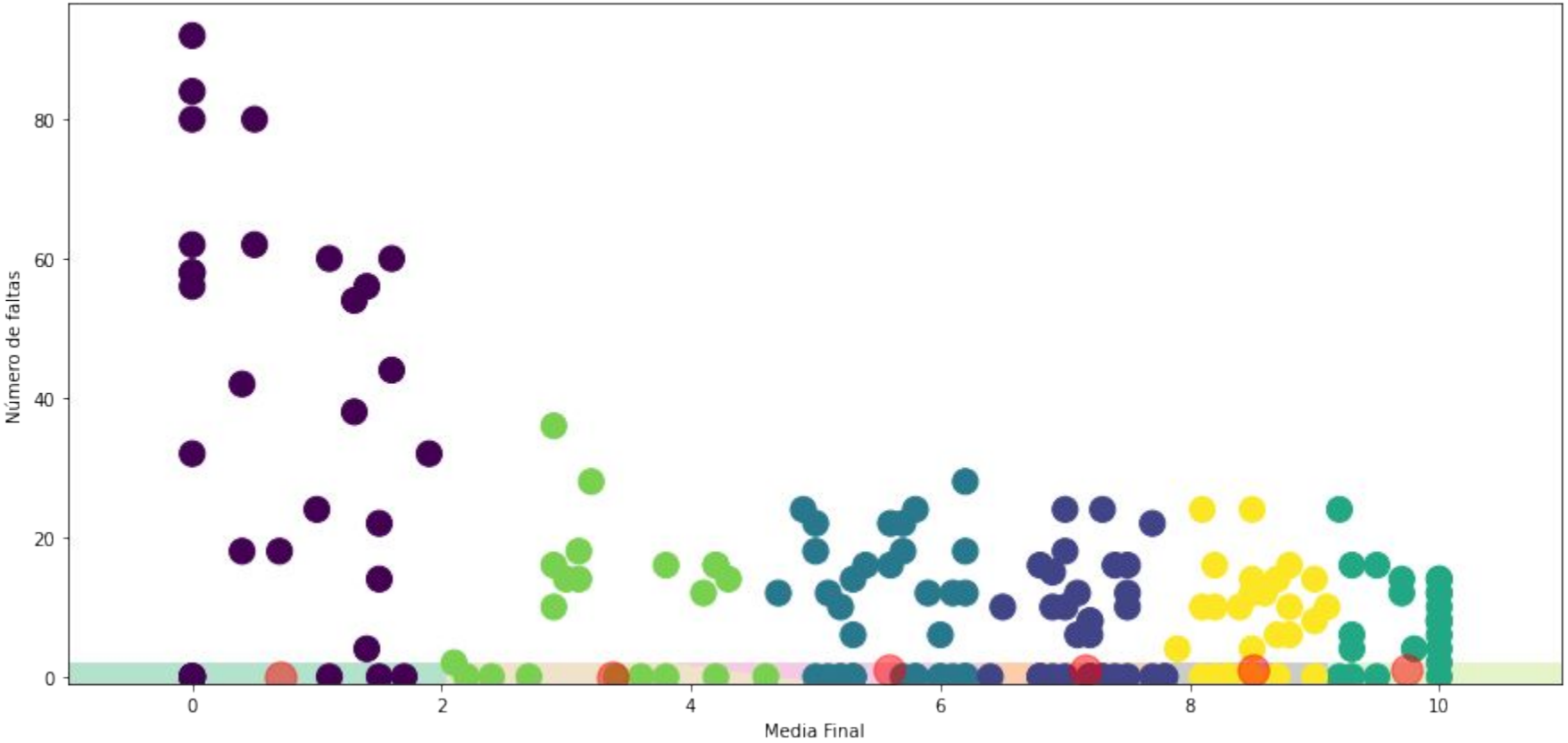
```
2148
```

K-means:

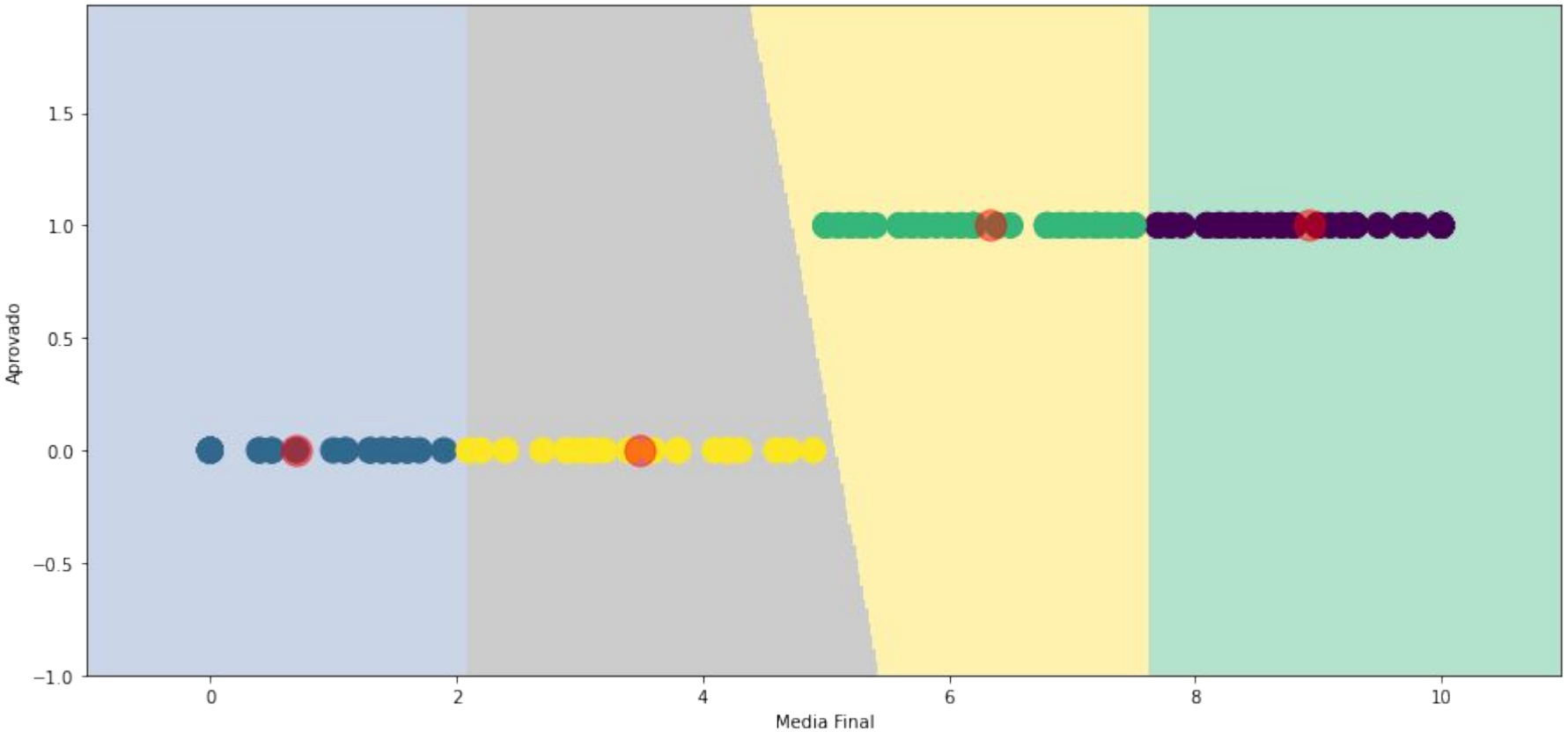
O K-means é um algoritmo do tipo não supervisionado, ou seja, que não trabalha com dados rotulados O objetivo desse algoritmo é encontrar similaridades entre os dados e agrupá-los conforme o número de cluster passado pelo argumento K.



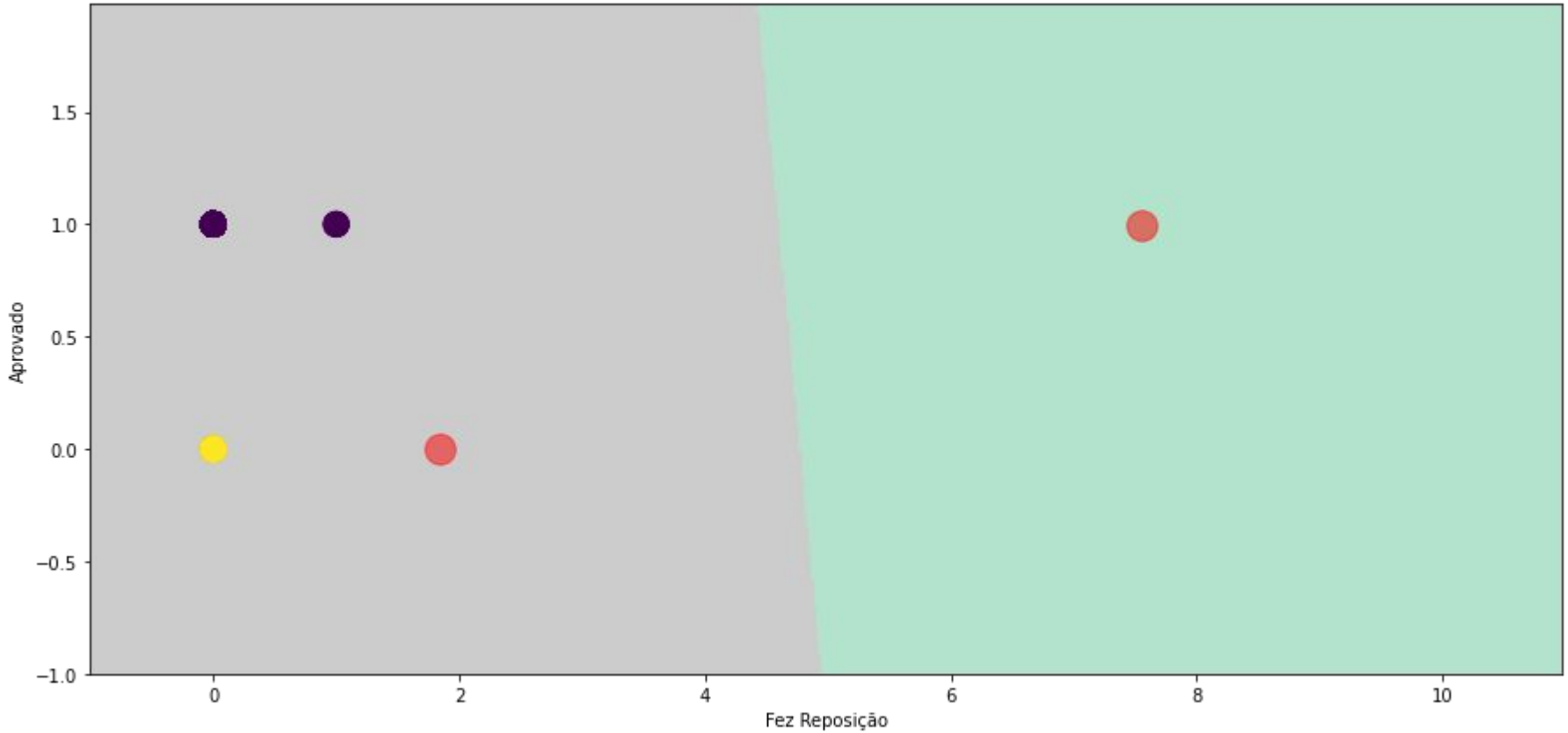
Número de faltas x média final



Aprovado x média final



Aprovado x fez reposição



Obrigado pela atenção

abmaeld/data-science-itp-analisys: Projeto de ciência de dados: Análise de dados abertos da UFRN usando aprendizado não supervisionado (github.com)