

Programa de Pós-graduação em Sistemas de Informação

SIN5007 - Reconhecimento de Padrões (2023)

Censo da Educação Superior - Cursos 2021

MSc. Leonardo Cunha dos Santos
Gabriel Francisco dos Santos Silva

São Paulo / 2023




Agenda

- 1 Descrição do dataset e análise exploratória
- 2 Pré-processamento e PCA
- 3 Seleção de características
- 4 Naive Bayes Classifier
- 5 Estimação de desempenho

Descrição do dataset e análise exploratória

JORNAL DA USP

 PORTAL DA USP —  FALE CONOSCO —  WHATSAPP —  ENVIE UMA PAUTA —  NEWSLETTER —  PODCASTS —  RÁDIO USP


ATUALIDADES ▾ CIÊNCIAS ▾ CULTURA ▾ DIVERSIDADE ▾ EDUCAÇÃO INSTITUCIONAL ▾ RÁDIO USP ▾ TECNOLOGIA UNIVERSIDADE ▾  BUSCA

Início > Institucional > Evasão na graduação da USP é de 17%, mas número varia por curso



Evasão na graduação da USP é de 17%, mas número varia muito por curso

Pesquisa investigou fatores relacionados à evasão na turma de 2018, a primeira com cotas sociais e raciais

 Publicado: 15/08/2023

Texto: Silvana Salles

Arte: Carolina Borin*

<https://jornal.usp.br/diversidade/evasao-na-graduacao-da-usp-e-de-17-mas-numero-varia-muito-por-curso/>

- Os microdados do Inep reúnem informações detalhadas sobre pesquisas do INEP;
- As estatísticas produzidas pelo Inep visam fornecer os subsídios para a formulação e implementação de políticas voltadas para a melhoria contínua da educação no país;
- Os formatos de apresentação foram reestruturados de acordo com a Lei Geral de Proteção de Dados Pessoais (LGPD).

<https://www.gov.br/inep/pt-br/aceso-a-informacao/dados-abertos/microdados/censo-da-educacao-superior>

Conjunto de dados

Microdados do Censo da Educação Superior

Cadastro de IES

- Localização
- Quadro de funcionários
- Infraestrutura
- Estatísticas sobre professores

Cadastro de Cursos

- Grau acadêmico
- Modalidade
- Estatísticas sobre estudantes

Pré-processamento de dados

Dicionário de dados

	Alteração de nomenclatura
	Variável nova
	Descontinuidade

Cadastro_IES

N	Nome da Variável	Descrição da Variável	Tipo	Tam.	Categoria
1	NU_ANO_CENSO	Ano de referência do Censo da Educação Superior	Num	4	
DADOS DA INSTITUIÇÃO DE ENSINO SUPERIOR (IES) - SEDE ADMINISTRATIVA/REITORIA					
2	NO_REGIAO_IES	Nome da região geográfica da sede administrativa ou reitoria da IES	Char	20	
3	CO_REGIAO_IES	Código da região geográfica da sede administrativa ou reitoria da IES	Num	2	
4	NO_UF_IES	Nome da Unidade da Federação da sede administrativa ou reitoria da IES	Char	50	
5	SG_UF_IES	Sigla da Unidade da Federação da sede administrativa ou reitoria da IES	Char	2	
6	CO_UF_IES	Código da Unidade da Federação da sede administrativa ou reitoria da IES	Num	2	
7	NO_MUNICIPIO_IES	Nome do Município da sede administrativa ou reitoria da IES	Char	150	
8	CO_MUNICIPIO_IES	Código do Município da sede administrativa ou reitoria da IES	Num	7	
9	IN_CAPITAL_IES	Informa se a sede administrativa ou reitoria da IES está localizada na capital da Unidade da Federação	Num	2	0. Não 1. Sim
10	NO_MESORREGIAO_IES	Nome da Mesorregião da sede administrativa ou reitoria da IES	Char	100	
11	CO_MESORREGIAO_IES	Código da Mesorregião da sede administrativa ou reitoria da IES	Num	4	
12	NO_MICRORREGIAO_IES	Nome da Microrregião da sede administrativa ou reitoria da IES	Char	100	
13	CO_MICRORREGIAO_IES	Código da Microrregião da sede administrativa ou reitoria da IES	Num	5	
14	TP_ORGANIZACAO_ACADEMICA	Tipo de Organização Acadêmica da IES	Num	1	1. Universidade 2. Centro Universitário 3. Faculdade 4. Instituto Federal de Educação, Ciência e Tecnologia 5. Centro Federal de Educação Tecnológica

Pré-processamento de dados

Microdados 2021 - Variáveis

Cadastro de IES

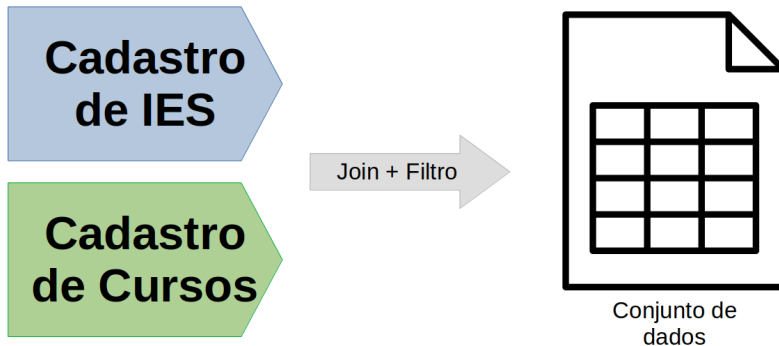
- Categóricas: 32 + 1 (ID:ANO)
- Numéricas: 48
- Total: 81

Cadastro de CURSOS

- Categóricas: 27 + 1 (ID:ANO)
- Numéricas: 172
- Total: 200

Pré-processamento de dados

Cursos de tecnologia



Variáveis numéricas: 05 | Variáveis categóricas: 22 + 1 (ID:ANO)
34 cursos | Instâncias: 19158 | 558 cursos distintos

Pré-processamento de dados

Depara de cursos



EACH | campus capital
USP
Escola de Artes, Ciências e Humanidades
Universidade de São Paulo

NO CURSO DEPARA	NO CURSO
Agrocomputação	Agrocomputação
Análise e Desenvolvimento de Sistemas	Administração Em Sistemas E Serviços De Saúde Análise De Infraestrutura De Redes E Sistemas Computacionais Análise De Sistemas Análise E Desenvolvimento De Sistemas Desenvolvimento De Sistemas Sistemas Para Internet
Ciências da Computação	Abi - Ciência Da Computação Ciência Da Computação Ciências Da Computação Ciências De Computação Computação Computação E Informática Computação E Robótica Educativa Computação Em Nuvem Computação Gráfica Internet Das Coisas E Computação Em Nuvem
Engenharia da Computação	Engenharia Da Computação Engenharia De Computação Engenharia De Computação - Ênfase Sistemas Corporativos Engenharia De Computação E Informação
Engenharia de Sistemas	Engenharia De Automação E Sistemas Engenharia De Produção E Sistemas Engenharia De Sistemas Engenharia De Sistemas Ciber Físicos
Engenharia Elétrica - Ênfase Em Computação	Engenharia Elétrica - Ênfase Em Computação Engenharia Elétrica - Ênfase Em Eletrônica E Sistemas Computacionais Engenharia Eletrônica E De Computação
Matemática Aplicada e Computação Científica	Interdisciplinar Em Matemática E Computação E Suas Tecnologias Matemática Aplicada Com Habilitação Em Sistemas E Controle Matemática Aplicada E Computação Científica Matemática Aplicada E Computacional Com Habilitação Em Sistemas E Controle
Sistemas de Computação	Sistemas De Computação
Sistemas de Informação	Sistemas De Informação

Análise exploratória

Método info da biblioteca Pandas para variáveis numéricas

```
RangeIndex: 19158 entries, 0 to 19157
```

```
Data columns (total 28 columns):
```

#	Column	Non-Null Count	Dtype
0	NU_ANO_CENSO	19158 non-null	object
1	NO_REGIAO	19158 non-null	object
2	NO_UF	19158 non-null	object
3	SG_UF	19158 non-null	object
4	NO_MUNICIPIO	19158 non-null	object
5	CO_MUNICIPIO	19158 non-null	object
6	IN_CAPITAL_DEPARA	19158 non-null	object
7	CO_IES	19158 non-null	object
8	SG_IES	19158 non-null	object
9	NO_IES	19158 non-null	object
10	CO_MANTENEDORA	19158 non-null	object
11	NO_MANTENEDORA	19158 non-null	object
12	NO_CURSO	19158 non-null	object

13	NO_CURSO_DEPARA	19158 non-null	object
14	CO_CURSO	19158 non-null	object
15	TP_GRAU_ACADEMICO_DEPARA	19155 non-null	object
16	IN_GRATUITO_DEPARA	19158 non-null	object
17	TP_MODALIDADE_ENSINO_DEPARA	19158 non-null	object
18	TP_NIVEL_ACADEMICO_DEPARA	19158 non-null	object
19	TP_DIMENSAO_DEPARA	19158 non-null	object
20	TP_ORGANIZACAO_ACADEMICA_DEPARA	19158 non-null	object
21	TP_CATEGORIA_ADMINISTRATIVA_DEPARA	19158 non-null	object
22	TP_REDE_DEPARA	19158 non-null	object
23	QT_V6_TOTAL	19158 non-null	int64
24	QT_INSCRITO_TOTAL	19158 non-null	int64
25	QT_ING	19158 non-null	int64
26	QT_MAT	19158 non-null	int64
27	QT_CONC	19158 non-null	int64

```
dtypes: int64(5), object(23)
```

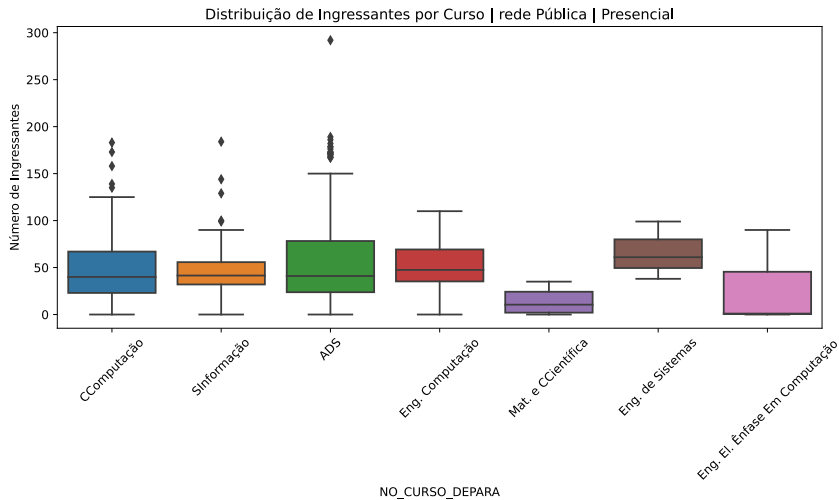
```
memory usage: 4.1+ MB
```

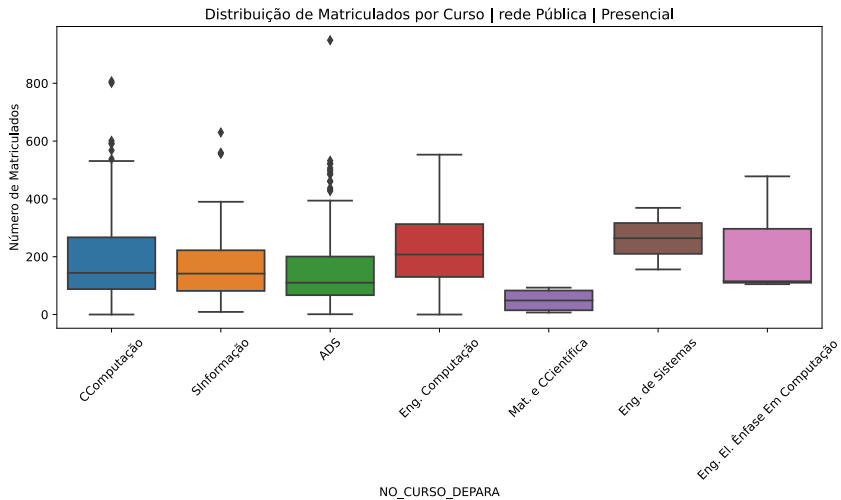
Análise exploratória

Método describe para variáveis numéricas (base total)

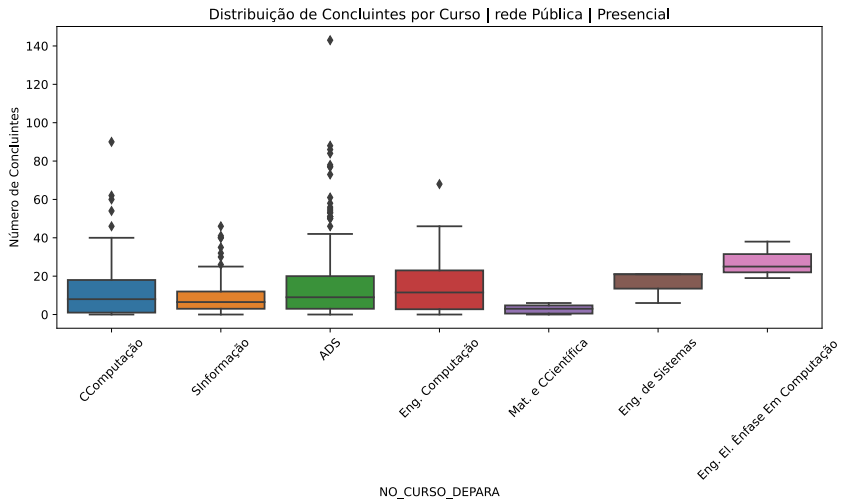
	QT_VG_TOTAL	QT_INSCRITO_TOTAL	QT_ING	QT_MAT	QT_CONC
count	19158	19158	19158	19158	19158
mean	42,84	36,20	10,22	19,61	2,22
std	754,03	468,54	37,86	68,31	10,61
min	0	0	0	0	0
25%	0	0	1	1	0
50%	0	0	2	2	0
75%	0	0	6	8	1
max	73280	32024	1794	3028	608

Análise exploratória



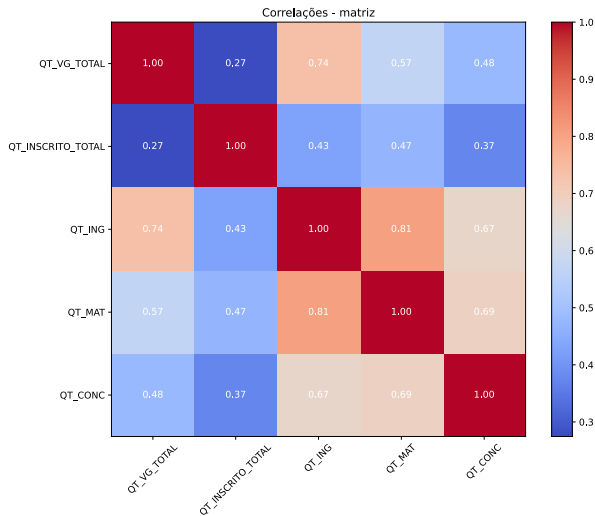


Análise exploratória



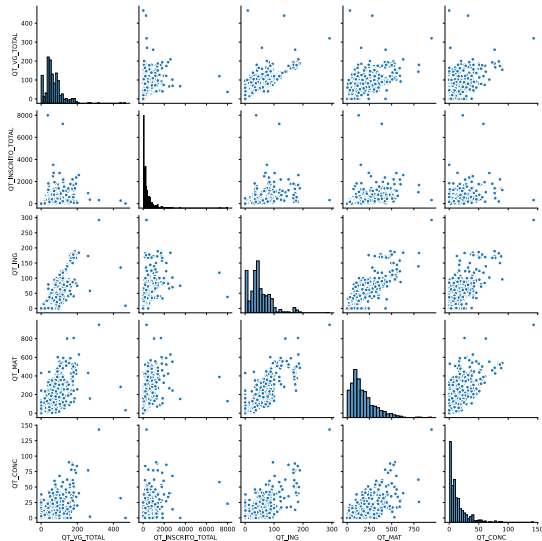
Análise exploratória

Correlações - matriz | rede Pública | Presencial



Análise exploratória

Correlações - Pairplot | rede Pública | Presencial



Considerações finais

Com este conjunto de dados, podemos:

- regionalizar (geograficamente) as análises;
- analisar cursos, modalidade, grau acadêmico e rede de ensino separadamente;
- ampliar o conjunto de dados com informações não utilizadas na primeira versão ou complementar com dados de anos anteriores.

Pré-processamento e PCA

Conjunto de dados

Versão depura

```
RangeIndex: 19158 entries, 0 to 19157
```

```
Data columns (total 17 columns):
```

#	Column	Non-Null Count	Dtype
0	NO_REGIAO	19158 non-null	object
1	NO_UF	19158 non-null	object
2	IN_CAPITAL_DEPARA	19158 non-null	object
3	NO_CURSO_DEPARA	19158 non-null	object
4	TP_GRAU_ACADEMICO_DEPARA	19155 non-null	object
5	IN_GRATUITO_DEPARA	19158 non-null	object
6	TP_MODALIDADE_ENSINO_DEPARA	19158 non-null	object
7	TP_NIVEL_ACADEMICO_DEPARA	19158 non-null	object

8	TP_DIMENSAO_DEPARA	19158 non-null	object
9	TP_ORGANIZACAO_ACADEMICA_DEPARA	19158 non-null	object
10	TP_CATEGORIA_ADMINISTRATIVA_DEPARA	19158 non-null	object
11	TP_REDE_DEPARA	19158 non-null	object
12	QT_VG_TOTAL	19158 non-null	int64
13	QT_INSCRITO_TOTAL	19158 non-null	int64
14	QT_ING	19158 non-null	int64
15	QT_MAT	19158 non-null	int64
16	QT_CONC	19158 non-null	int64

```
dtypes: int64(5), object(12)
```

```
memory usage: 2.5+ MB
```

One-hot encoding

Utilização do pacote Pandas: `get_dummies`

```
categories = ['NO_REGIAO', 'NO_UF', 'IN_CAPITAL_DEPARA', 'NO_CURSO_DEPARA',  
             'TP_GRAU_ACADEMICO_DEPARA', 'IN_GRATUITO_DEPARA', 'TP_MODALIDADE_ENSINO_DEPARA',  
             'TP_NIVEL_ACADEMICO_DEPARA', 'TP_DIMENSAO_DEPARA', 'TP_ORGANIZACAO_ACADEMICA_DEPARA',  
             'TP_CATEGORIA_ADMINISTRATIVA_DEPARA']  
  
encoded_data = pd.get_dummies(df[categories], prefix=categories, prefix_sep='_')  
encoded_data = encoded_data.apply(lambda x: x.astype(bool).astype(int))  
  
encoded_data = encoded_data.merge(df[['QT_VG_TOTAL', 'QT_INSCRITO_TOTAL', 'QT_ING',  
                                     'QT_MAT', 'QT_CONC', 'TP_REDE_DEPARA']],  
                                left_index=True, right_index=True, how='left')
```

One-hot encoding

Result set

```
##### One hot encode #####  
      NO_REGIAO_Centro-Oeste  NO_REGIAO_Nordeste  ...  QT_CONC  TP_REDE_DEPARA  
0                0                0  ...      0      Pública  
1                0                0  ...      0      Pública  
2                0                0  ...      0      Pública  
3                0                0  ...      0      Pública  
4                0                0  ...      0      Privada  
...                ...                ...  ...      ...      ...  
19153            0                0  ...      0      Privada  
19154            0                0  ...      0      Privada  
19155            0                0  ...      1      Privada  
19156            0                0  ...      0      Privada  
19157            0                0  ...      0      Privada  
  
[19158 rows x 75 columns]
```

StandardScaler

Utilização do pacote StandardScaler do Sklearn

```
#####  
print(f' Normalização de variáveis '.center( __width: 80 __fillchar: '#'))  
  
scaler = StandardScaler()  
  
columns=['QT_VG_TOTAL', 'QT_INSCRITO_TOTAL', 'QT_ING', 'QT_MAT', 'QT_CONC']  
  
normalized_data = scaler.fit_transform(encoded_data[columns])  
normalized_data = pd.DataFrame(normalized_data, columns=columns)  
print(normalized_data)
```

StandardScaler

Result set

Normalização de variáveis

	QT_VG_TOTAL	QT_INSCRITO_TOTAL	QT_ING	QT_MAT	QT_CONC
0	0.724341	3.745327	-0.269880	-0.287079	-0.20961
1	-0.056820	-0.077259	-0.269880	-0.287079	-0.20961
2	-0.056820	-0.077259	-0.269880	-0.287079	-0.20961
3	-0.056820	-0.077259	-0.269880	-0.287079	-0.20961
4	12.420536	12.534500	-0.269880	-0.287079	-0.20961
...
19153	-0.056820	-0.077259	-0.243467	-0.272439	-0.20961
19154	-0.056820	-0.077259	-0.164229	-0.169958	-0.20961
19155	-0.056820	-0.077259	-0.111404	-0.228518	-0.11534
19156	-0.056820	-0.077259	-0.243467	-0.272439	-0.20961
19157	-0.056820	-0.077259	-0.243467	-0.272439	-0.20961

Dimensões do conjunto de dados

Após a padronização

- Variáveis qualitativas (binárias): 69
- Variáveis quantitativas (padronizadas): 5
- Alvo: TP_REDE_DEPARA
- Total de variáveis no result set: 75
- Instâncias: 19158

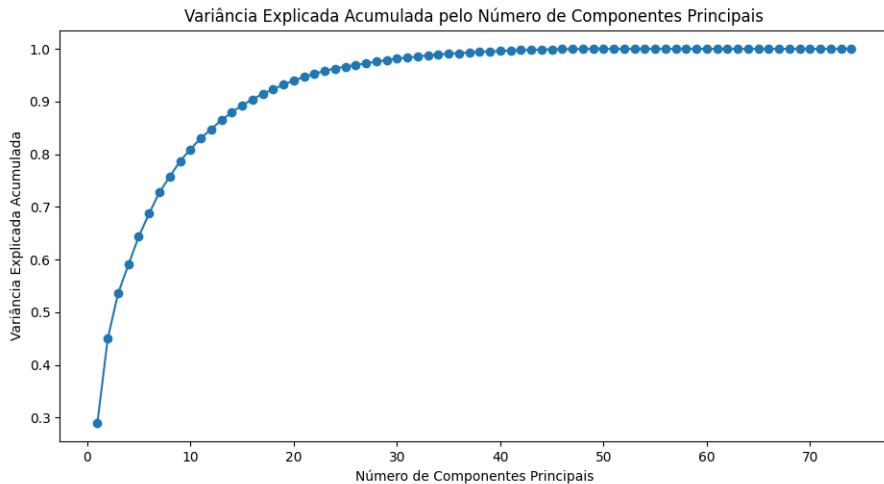
PCA

Principal Component Analysis

```
from sklearn.decomposition import PCA
import matplotlib.pyplot as plt

data_X = normalized_data_vf.drop('TP_REDE_DEPARA', axis=1)

pca = PCA()
pca.fit(data_X)
explained_variance_ratio = pca.explained_variance_ratio_
```



Número de Componentes Principais para 95% de Variância: 22

PCA com número de componentes esperado

	PC1	PC2	PC3	...	PC20	PC21	PC22
0	0.190351	3.345125	-0.140522	...	-1.051111	-1.318289	-0.026662
1	-0.271789	0.161091	-0.924383	...	-1.180369	-1.477429	-0.046666
2	-0.309983	0.141671	0.358834	...	-1.032111	-1.547045	-0.003615
3	-0.309983	0.141671	0.358834	...	-1.032111	-1.547045	-0.003615
4	1.720391	17.651678	-0.612151	...	-0.292294	-0.570287	-0.030889
...
19153	-0.535000	-0.050070	0.392255	...	-0.039530	-0.041791	0.950386
19154	-0.439304	-0.069639	0.916337	...	0.018221	-0.060235	0.949860
19155	-0.392884	-0.077730	0.929753	...	-0.017430	-0.033971	0.951916
19156	-0.447089	-0.046038	0.713141	...	0.101151	0.011569	0.959502
19157	-0.535000	-0.050070	0.392255	...	-0.039530	-0.041791	0.950386

[19158 rows x 22 columns]

Considerações finais

Com esta atividade:

- Efetuamos o processo one-hot encoding, que transformou as variáveis categóricas em numéricas;
- Reduzimos a quantidade de variáveis para 22 componentes principais.

Seleção de características

RELIEF (Kira and Rendell, 1992)

Tipo filtro: seleção de características independente do classificador

```
def relief(X, y):  
    X = X.to_numpy()  
    y = y.to_numpy()  
    print(f'Dimensões da entrada X: {X.shape}')  
    print(f'Dimensões da entrada y: {y.shape}')  
  
    num_samples, num_features = X.shape  
    weights = np.zeros(num_features)  
  
    for i in range(num_samples):  
        current_instance = X[i, :]  
  
        nearest_hit = None  
        nearest_miss = None  
        min_hit_distance = float("inf")  
        min_miss_distance = float("inf")
```

```
        for j in range(num_samples):  
            if i != j:  
                distance = np.linalg.norm(current_instance - X[j, :])  
                if y[i] == y[j]:  
                    if distance < min_hit_distance:  
                        min_hit_distance = distance  
                        nearest_hit = X[j, :]  
                else:  
                    if distance < min_miss_distance:  
                        min_miss_distance = distance  
                        nearest_miss = X[j, :]  
  
        weights += np.abs(current_instance - nearest_hit) - \  
            np.abs(current_instance - nearest_miss)  
  
    return weights / num_samples
```

RELIEF (Kira and Rendell, 1992)

Tipo filtro: seleção de características independente do classificador



EACH

Escola de Artes, Ciências e Humanidades
Universidade de São Paulo



```
##### Relief #####  
Dimensões da entrada X: (19158, 74)  
Dimensões da entrada y: (19158,)  
Selected Feature Names:  
NO_CURSO_DEPARA_Agrocomputação  
NO_UF_Paraíba  
NO_UF_Tocantins  
NO_UF_Rio Grande do Norte  
NO_UF_Amazonas  
NO_UF_Sergipe  
NO_UF_Distrito Federal  
NO_UF_Acre  
NO_UF_Amapá  
NO_UF_Roraima  
TP_DIMENSAO_DEPARA_Cursos a distância com dimensão de dados somente a nível Brasil  
NO_CURSO_DEPARA_Eng. El. Ênfase Em Computação  
TP_ORGANIZACAO_ACADEMICA_DEPARA_Centro Federal de Educação Tecnológica  
TP_DIMENSAO_DEPARA_Cursos a distância ofertados por instituições brasileiras no exterior  
NO_UF_Sem_uf  
NO_REGIAO_Sem_regiao  
NO_CURSO_DEPARA_Mat. e CCientifica  
NO_CURSO_DEPARA_Eng. de Sistemas  
TP_NIVEL_ACADEMICO_DEPARA_Sequencial de Formação Específica  
TP_NIVEL_ACADEMICO_DEPARA_Graduação
```


Naive Bayes Classifier

Naive Bayes Classifier

Bibliotecas

```
#####  
print(f' Carga de bibliotecas '.center(80 , '#'))  
  
import pandas as pd  
import numpy as np  
from sklearn.model_selection import train_test_split  
from sklearn.naive_bayes import GaussianNB  
from sklearn.metrics import accuracy_score  
  
# Defina a semente  
seed_value = 45  
np.random.seed(seed_value)
```

Naive Bayes Classifier

Separação em conjuntos de treinamento e teste

```
#####
```

```
print(f' Carga de dados '.center(80 , '#'))
```

```
df = pd.read_csv('arquivos/dados/tema02_cursos_2021_ti_03_normalized.csv',  
                 index_col=0, sep='|')
```

```
print(df.head())
```

```
print(df.shape)
```

```
X = df.drop('TP_REDE_DEPARA', axis=1)
```

```
y = df['TP_REDE_DEPARA']
```

```
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2,  
                                                    stratify=y, random_state=seed_value)
```

Naive Bayes Classifier

Treinamento, teste e cálculo da acurácia

```
#####
```

```
print(f' Naive Bayes Classifier '.center(80 , '#'))
```

```
# Inicialize o classificador Naive Bayes Gaussiano
```

```
naive_bayes_classifier = GaussianNB()
```

```
# Treine o classificador com os dados de treinamento
```

```
naive_bayes_classifier.fit(X_train, y_train)
```

```
# Faça previsões nos dados de teste
```

```
y_pred = naive_bayes_classifier.predict(X_test)
```

```
# Avalie a precisão do modelo
```

```
accuracy = accuracy_score(y_test, y_pred)
```

```
print("Acurácia do modelo Naive Bayes:", accuracy)
```

Naive Bayes Classifier

Conjunto de dados Total

```
##### Carga de dados #####
  NO_REGIAO_Centro-Oeste  NO_REGIAO_Nordeste  ...  QT_CONC  TP_REDE_DEPARA
0                0                0  ... -0.20961      Pública
1                0                0  ... -0.20961      Pública
2                0                0  ... -0.20961      Pública
3                0                0  ... -0.20961      Pública
4                0                0  ... -0.20961      Pública

[5 rows x 75 columns]
(19158, 75)
##### Naive Bayes Classifier #####
Acurácia do modelo Naive Bayes: 1.0
```

Naive Bayes Classifier

Conjunto de dados PCA

```
##### Carga de dados #####  
      PC1      PC2      PC3  ...      PC21      PC22  TP_REDE_DEPARA  
0 -0.271789  0.161091 -0.924383  ... -1.477637 -0.047432      Pública  
1  0.190351  3.345125 -0.140522  ... -1.318910 -0.022477      Pública  
2 -0.245303  0.359979 -0.920680  ... -1.464840 -0.046929      Pública  
3  0.237584  3.408979 -0.922878  ... -1.264806 -0.039676      Pública  
4 -0.284295  0.157172 -0.929598  ... -1.490612 -0.019619      Pública  
  
[5 rows x 23 columns]  
(19158, 23)  
  
##### Naive Bayes Classifier #####  
Acurácia do modelo Naive Bayes: 0.9835594989561587
```

Naive Bayes Classifier

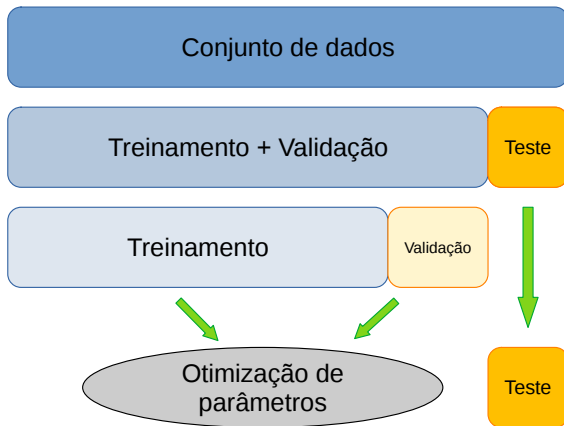
Conjunto de dados Selecionado - 22 atributos do Relief

```
##### Carga de dados #####  
  NO_UF_Rondônia  ...  TP_REDE_DEPARA  
0                0  ...      Pública  
1                0  ...      Pública  
2                0  ...      Pública  
3                0  ...      Pública  
4                0  ...      Pública  
  
[5 rows x 23 columns]  
(19158, 23)  
  
##### Naive Bayes Classifier #####  
Acurácia do modelo Naive Bayes: 0.9191022964509394
```

Estimação de desempenho

Separação de conjuntos de dados

Treinamento, validação e teste



Planejamento do experimento

Validação cruzada estratificada

- Estratégia: grid-search
- Maximização: F1-score
- Valor de k: 5
- Modelo: Naive Bayes
- Valores de hiperparâmetros testados:
'var_ssmoothing' : *np.logspace*(0, -15, num = 100)

* *var_ssmoothing*float, default=1e-9

Portion of the largest variance of all features that is added to variances for calculation stability.

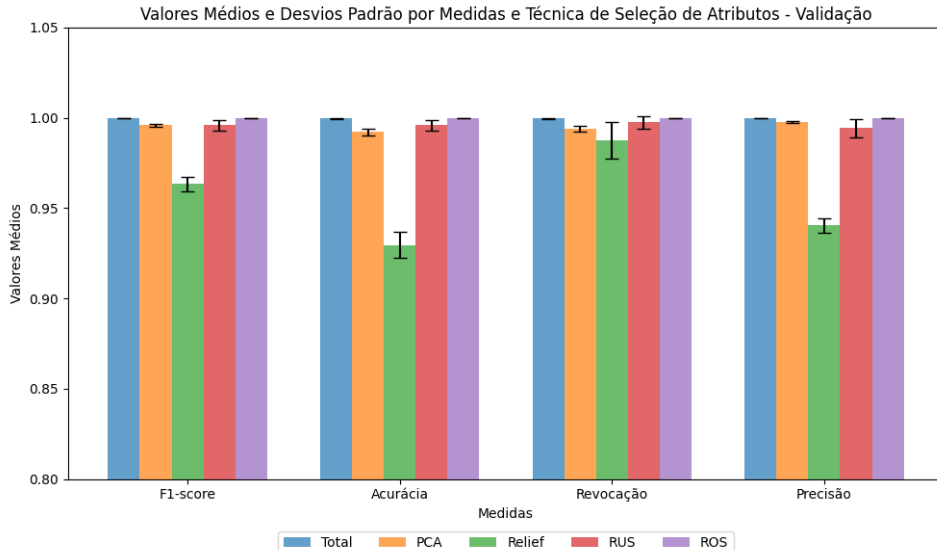
Planejamento do experimento

Validação cruzada estratificada

Médias	F1-score	Acurácia	Revocação	Precisão	var_smoothing(avg)
Total	0.9998	0.9997	0.9997	0.9999	5.3e-04
PCA	0.9958	0.9921	0.9939	0.9977	5.1e-01
Relief	0.9634	0.9296	0.9876	0.9404	8.0e-01
RUS	0.9958	0.9958	0.9975	0.9942	1.1e-02
ROS	1.0000	1.0000	1.0000	1.0000	4.7e-04

Planejamento do experimento

Resultados



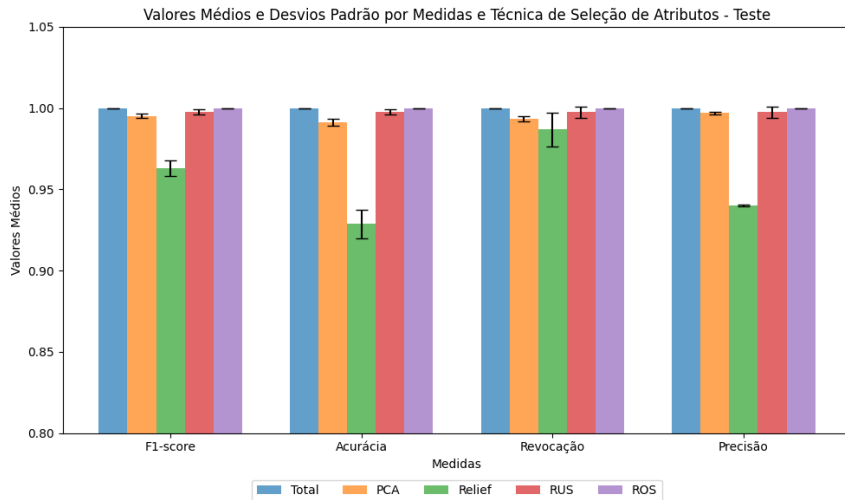
Planejamento do experimento

Valores obtidos com os conjuntos de teste

Médias	F1-score	Acurácia	Revocação	Precisão
Total	0.9999	0.9998	0.9999	0.9999
PCA	0.9952	0.9911	0.9934	0.9971
Relief	0.9629	0.9287	0.9868	0.9402
RUS	0.9975	0.9975	0.9975	0.9975
ROS	1.0000	1.0000	1.0000	1.0000

Planejamento do experimento

Resultados utilizando o conjunto de teste



Obrigado!

Thanks! / ¡Gracias!

Leonardo Cunha dos Santos

`lattes.cnpq.br/5620610314140397`

`leonardo.cunha.santos@usp.br`

Gabriel Francisco dos Santos Silva

`gabfssilva@gmail.com`