Recuperação de Informação 2021.1

Centro de Informática
UFPE

Universidade Federal de Pernambuco

Centro de Informática
UFPE

Universidade Federal de Pernambuco

Vitor Sousa, 21
Ciência da computação
@vss2

Lucas Silva, 21
Ciência da computação
@lvjs

Agenda:

- Crawler:
  - Domínio e Sites escolhidos
  - Crawling:
    - Baseline
    - Heurística:
      - Classificador de links
  - Harvest ratio
  - Dificuldades
- Classificador:
  - Rotulação de exemplos
  - Conjunto de features
  - Treinamento:
    - Scikit-learn
  - Comparação de estratégias

Crawler:

Domínio = "Televisão"

Sites = ["Amazon", "Magazine Luiza", "Carrefour", "Colombo", "Havan", "Kabum", "Laser Eletro", "Mercado Livre", "Ricardo Eletro", "Gazin"]

Crawler:

Crawling:

Ferramentas:

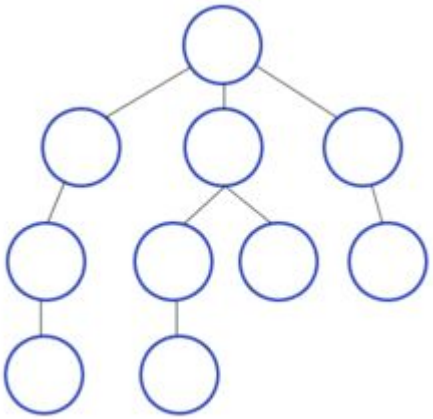
- Selenium
- BeautifulSoup

Filtros:

- Regular Expression
- Robots.txt

Crawling:

Baseline:



1	2	3	4	5	6	7	8	9	10

Crawling:

Heurística:

Classificador de links e/ou âncora:

#links extraídos com notas

1	2	2	4	8	16	32

#ordem dos links a serem visitados

32	16	8	4	2	2	1

Heurística:

```python
import re
from getRecDados import getBagOfWords

def classificador(texto: str = '') -> int:
    if texto == '':
        return 0
    texto = texto.casefold()

    for x in ['-', '_', ',', '/']:
        texto = texto.replace(x, ' ')

    texto_sp = texto.split(' ')
    texto_sp = [k.strip() for k in texto_sp]

    bow = getBagOfWords()

    score = 1

    for b in bow:
        if b in texto_sp:
            score *= 2
    try:
        # Removendo polegadas pela expressão regular
        valor = re.search('[0-9]{2}', texto)
        if(valor.regs):
            score *= 2
    except:
        pass

    return score

if __name__ == '__main__':
    print(classificador())
```

Uso:

```python
def filtroGuiado(soup):
    a_com_link = soup.findAll('a', href=True)
    sitesAvisitar = []

    for acl in a_com_link:
        try:
            k = acl.get('href')
            link_a_class = ' '.join(str(k).casefold().split())
            notalink = classificador(link_a_class)
            notatexto = classificador(acl.text)
            nota = notalink + notatexto
            nota = nota / 2
            if (notalink + notatexto) > 1:
                sitesAvisitar.insert(0, (k, nota))
        except Exception as e:
            # print(e)
            pass

    return sorted(sitesAvisitar, key=lambda x:x[1], reverse=True)
```

Harvest ratio (Amazon)

Baseline (Total Harvest ratio:)

	Naive Bayes	Random Forest	Multi-layer Perceptron	SVC	Logistic Regression
Relevantes	20	5	33	5	9
Visitadas	1000	1000	1000	1000	1000
Harvest ratio	0.02	0.005	0.33	0.005	0.009

Heurística (Total Harvest ratio:)

	Naive Bayes	Random Forest	Multi-layer Perceptron	SVC	Logistic Regression
Relevantes	247	130	185	131	134
Visitadas	1000	1000	1000	1000	1000
Harvest ratio	0.247	0.130	0.185	0.131	0.134

Harvest ratio (Mercado Livre)

Baseline (Total Harvest ratio:)

	Naive Bayes	Random Forest	Multi-layer Perceptron	SVC	Logistic Regression
Relevantes	11	6	6	6	6
Visitadas	1000	1000	1000	1000	1000
Harvest ratio	0.011	0.006	0.006	0.006	0.006

Heurística (Total Harvest ratio:)

	Naive Bayes	Random Forest	Multi-layer Perceptron	SVC	Logistic Regression
Relevantes	942	938	939	626	934
Visitadas	1000	1000	1000	1000	1000
Harvest ratio	0.942	0.938	0.939	0.626	0.934

Harvest ratio (Kabum)

Baseline (Total Harvest ratio:)

	Naive Bayes	Random Forest	Multi-layer Perceptron	SVC	Logistic Regression
Relevantes	76	13	35	15	25
Visitadas	1000	1000	1000	1000	1000
Harvest ratio	0.076	0.013	0.035	0.015	0.025

Heurística (Total Harvest ratio:)

	Naive Bayes	Random Forest	Multi-layer Perceptron	SVC	Logistic Regression
Relevantes	592	229	366	206	304
Visitadas	976	976	976	976	976
Harvest ratio	0.606	0.234	0.375	0.211	0.311

Harvest ratio (Magazine Luiza)

Baseline (Total Harvest ratio:)

	Naive Bayes	Random Forest	Multi-layer Perceptron	SVC	Logistic Regression
Relevantes	53	35	41	37	36
Visitadas	989	989	989	989	989
Harvest ratio	0.05358	0.03558	0.04145	0.03771	0.03640

Heurística (Total Harvest ratio:)

	Naive Bayes	Random Forest	Multi-layer Perceptron	SVC	Logistic Regression
Relevantes	884	462	460	471	469
Visitadas	1000	1000	1000	1000	1000
Harvest ratio	0.884	0.462	0.460	0.471	0.469

Harvest ratio (Havan)

Baseline (Total Harvest ratio:)

	Naive Bayes	Random Forest	Multi-layer Perceptron	SVC	Logistic Regression
Relevantes	1	1	4	1	1
Visitadas	998	998	998	998	998
Harvest ratio	0.0010	0.0010	0.0010	0.0010	0.0010

Heurística (Total Harvest ratio:)

	Naive Bayes	Random Forest	Multi-layer Perceptron	SVC	Logistic Regression
Relevantes	27	26	26	26	26
Visitadas	1000	1000	1000	1000	1000
Harvest ratio	0.027	0.026	0.026	0.026	0.026

Harvest ratio (Gazin)

Baseline (Total Harvest ratio:)

	Naive Bayes	Random Forest	Multi-layer Perceptron	SVC	Logistic Regression
Relevantes	61	39	55	41	41
Visitadas	1000	1000	1000	1000	1000
Harvest ratio	0.061	0.039	0.055	0.041	0.041

Heurística (Total Harvest ratio:)

	Naive Bayes	Random Forest	Multi-layer Perceptron	SVC	Logistic Regression
Relevantes	46	7	35	9	9
Visitadas	1000	1000	1000	1000	1000
Harvest ratio	0.046	0.007	0.035	0.009	0.009

Harvest ratio (Laser Eletro)

Baseline (Total Harvest ratio:)

	Naive Bayes	Random Forest	Multi-layer Perceptron	SVC	Logistic Regression
Relevantes	119	5	79	6	11
Visitadas	1000	1000	1000	1000	1000
Harvest ratio	0.119	0.005	0.079	0.006	0.011

Heurística (Total Harvest ratio:)

	Naive Bayes	Random Forest	Multi-layer Perceptron	SVC	Logistic Regression
Relevantes	277	91	172	76	77
Visitadas	935	935	935	935	935
Harvest ratio	0.296	0.097	0.183	0.081	0.082

Harvest ratio (Ricardo Eletro)

Baseline (Total Harvest ratio:)

	Naive Bayes	Random Forest	Multi-layer Perceptron	SVC	Logistic Regression
Relevantes	16	3	10	3	4
Visitadas	1000	1000	1000	1000	1000
Harvest ratio	0.016	0.003	0.010	0.003	0.004

Heurística (Total Harvest ratio:)

	Naive Bayes	Random Forest	Multi-layer Perceptron	SVC	Logistic Regression
Relevantes	532	456	482	480	479
Visitadas	1000	1000	1000	1000	1000
Harvest ratio	0.532	0.456	0.482	0.480	0.479

Harvest ratio (Carrefour)

Baseline (Total Harvest ratio:)

	Naive Bayes	Random Forest	Multi-layer Perceptron	SVC	Logistic Regression
Relevantes	23	14	24	13	13
Visitadas	1000	1000	1000	1000	1000
Harvest ratio	0.023	0.014	0.024	0.013	0.013

Heurística (Total Harvest ratio:)

	Naive Bayes	Random Forest	Multi-layer Perceptron	SVC	Logistic Regression
Relevantes	616	208	577	355	442
Visitadas	1000	1000	1000	1000	1000
Harvest ratio	0.616	0.208	0.577	0.355	0.442

Harvest ratio (Colombo)

Baseline (Total Harvest ratio:)

	Naive Bayes	Random Forest	Multi-layer Perceptron	SVC	Logistic Regression
Relevantes	32	6	14	7	7
Visitadas	1000	1000	1000	1000	1000
Harvest ratio	0.032	0.006	0.014	0.007	0.007

Heurística (Total Harvest ratio:)

	Naive Bayes	Random Forest	Multi-layer Perceptron	SVC	Logistic Regression
Relevantes	168	134	140	137	137
Visitadas	1000	1000	1000	1000	1000
Harvest ratio	0.168	0.134	0.140	0.137	0.137

Crawler:

Dificuldades:

- Links que levam para fora do site (aplicativo, redes sociais, cartão-fidelidade, etc)
- Sites com alta “indisposição” (verificação de robôs) a serem minerados
- Links com prefixo
- Links com redirecionadores
- Especificidades de links de determinadas páginas
- Parâmetros causando comportamento obsessivo na busca heurística

Tomada de decisões do classificador:

Usar de tabela composta com 9 features positivas e negativas mais observadas:

- Classificação booleana (tem ou não tem)

Usar score (sem produto, apenas soma de aparições) para título e score para body:

- Classificação por aparição

Motivação: quando fizemos um produto, alguns sites com catálogo ou lista lateral acabavam poluindo a busca, por isso um score mais simples foi usado.

Treino e teste de rótulos

- Usamos **10 links positivos e 10 negativos rotulados previamente de cada site** para avaliar os classificadores;
- Dos 200 HTMLs disponíveis usamos **50** para **teste**, e os outros **150** para **treino**;
- As **páginas selecionadas como exemplos positivos** são telas de produto que são **claramente uma TV**.
- As páginas selecionadas como **exemplos negativos são telas de produto não são uma TV mas estão correlacionados** (acessórios: suporte, controle, cabos; aparelhos: receptores/smart plug, etc)

Naive Bayes (scikit-learn: Gaussian NB)

Avaliação	Precision	F-Measure	Acurácia
Pontuação	0.8968115942028986	0.9361702127659574	0.94
Tempo	em média 200s		
Parâmetros	var_smoothing:	1e-9 (default)	

Random Forest

Avaliação	Precision	F-Measure	Acurácia
Pontuação	0.9115415019762846	0.9333333333333332	0.94
Tempo	em média 300s		
Parâmetros	n_estimators: 100	criterion: “gini” (deft.)	**n_jobs: -1**
random_state: len//2	min_samples_leaf: 1	max_features: sqrt (deft.)	

Multilayer Perceptron

Avaliação	Precision	F-Measure	Acurácia
Pontuação	0.9530434782608695	0.9545454545454545	0.96
Tempo	em média +300s		
Parâmetros	solver: “adam” (deft.)	learning_rate: “cnst”	max_iter: 200
random_state: None	activation: “relu” (deft.)	n_iter_no_change: 10	

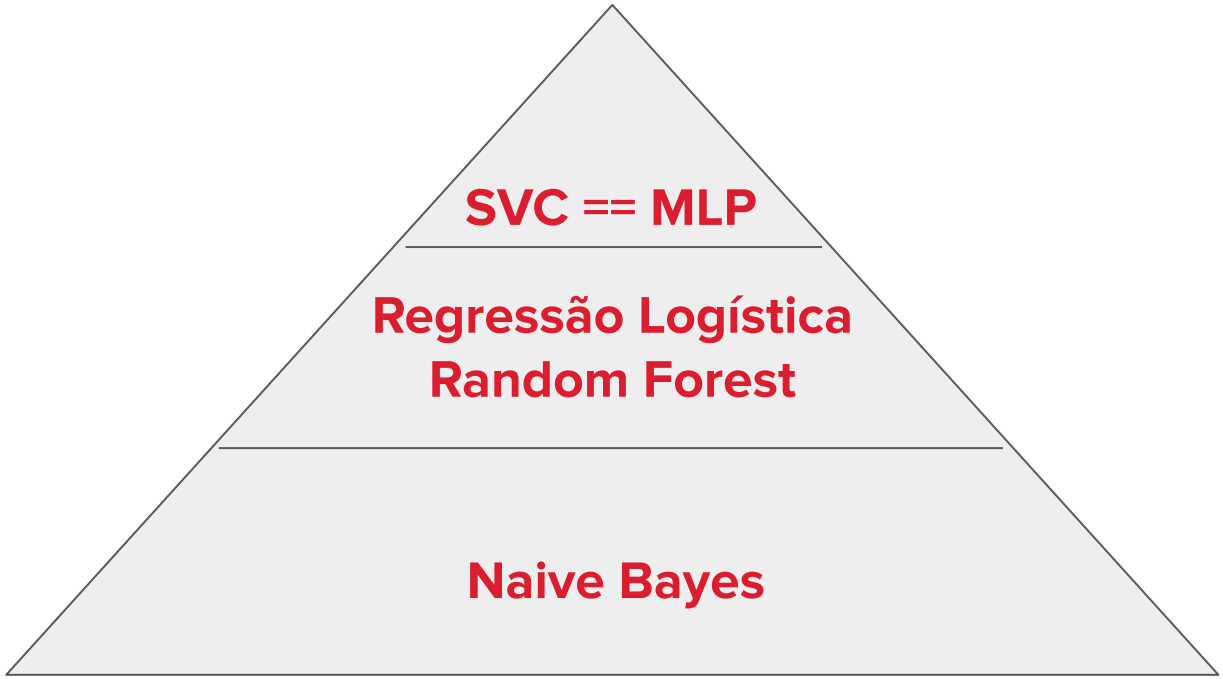
SVC

Avaliação	Precision	F-Measure	Acurácia
Pontuação	0.9530434782608695	0.9545454545454545	0.96
Tempo	em média 210s		
Parâmetros	random_state: None	max_iter: -1 (deft.)	

Regressão Logística

Avaliação	Precision	F-Measure	Acurácia
Pontuação	0.9115415019762846	0.9333333333333332	0.94
Tempo	em média 200-230s		
Parâmetros	max_iter: 100 (deft.)	tol: 1e-4 (deft.)	**n_jobs: -1**
fit_intercept: True (default)		intercept_scaling: 1 (default)	

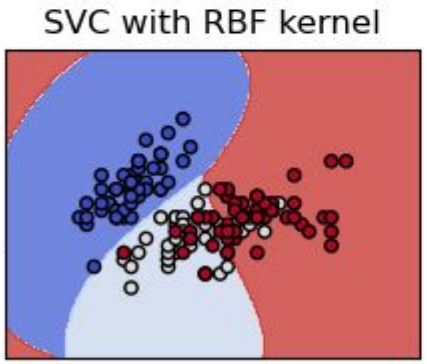
Classificação dos classificadores

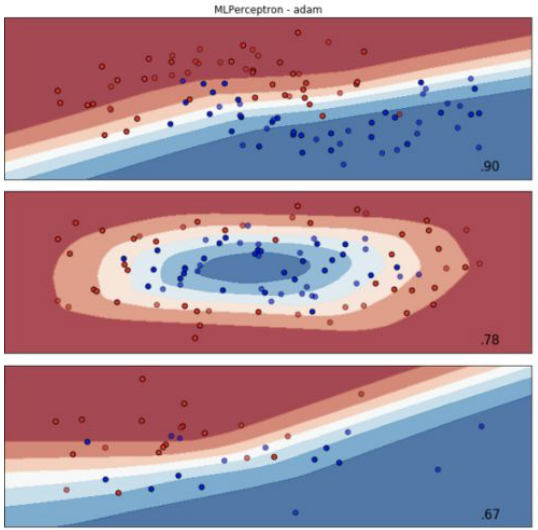


Classificação dos classificadores

Suspeitamos que o SVC usando “RBF” e o Multilayer Perceptron usando o “adam” (parâmetros padrão) se tornaram um pouco melhores por sua natural eficiência a se adequar aos modelos. Já **esperávamos que o Naive Bayes fosse o pior**.

Entretanto, dado que **a diferença entre as pontuações foi tão curta** (menos de 2%), assumimos que estamos lidando com caso de overfit. Vamos testar testar com outros %.





Imagens retiradas da bibliografia do SK-learn

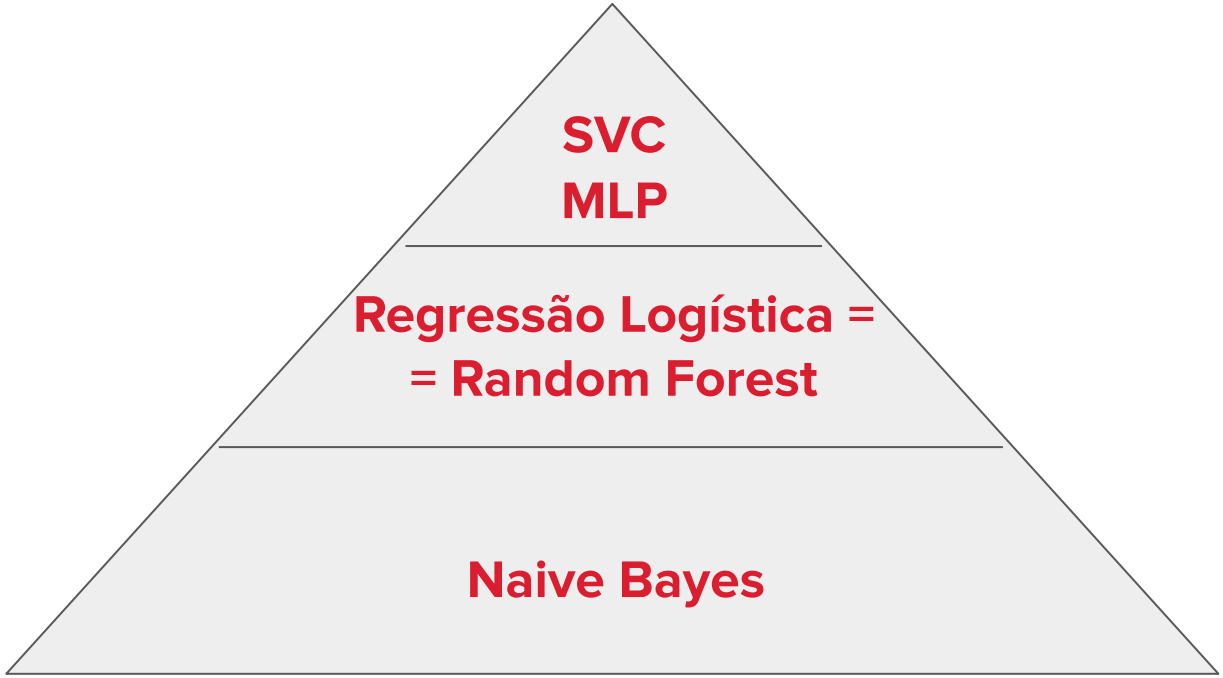
Revisitando classificadores (com 75% de teste)

Algoritmo	Precision	F-Measure	Acurácia
Naive Bayes	0.7075323737295568	0.8148148148148148	0.8
Random Forest	0.8406771143291676	0.8873239436619719	0.8933333333333333
MLP	0.8446499285568483	0.8857142857142857	0.8933333333333333
SVC	0.8781273556833962	0.9154929577464789	0.92
Regressão Log.	0.863498673198612	0.9	0.906666666666666

Revisitando classificadores (com 50% de teste)

Algoritmo	Precision	F-Measure	Acurácia
Naive Bayes	0.7784615384615384	0.8571428571428571	0.84
Random Forest	0.916571906354515	0.9183673469387755	0.92
MLP	0.9257937806873977	0.9292929292929293	0.93
SVC	0.9353846153846154	0.9278350515463918	0.93
Regressão Log.	0.916571906354515	0.9183673469387755	0.92

Nova classificação dos classificadores



Usando Optuna e MLflow para MLP e Random Forest (100 trials)

Algoritmo	Parâmetro 1	Parâmetro 2	Score
Random Forest	N estimators: 5 → 15	Max depth: 2 → 16	0.92
MLP	Power t: (0.5 → 0.9, 0.1)	Learning Rt: (1e-5 → 1e-3, log=True)	0.93

Resultados

Algoritmo	Parâmetro 1	Parâmetro 2	Trial n°	Resultado
Random Forest	10	10	11	0.96
MLP	0.9	0.0004620713702750547	1	0.96

Percebemos que os trials atingiram máximos rapidamente