Submission date: 13/Abr/2017
Resubmission date: 14/Jul/2017
Resubmission date: 29/Oct/2017
Camera ready submission: 30/May/2018

Section: regular article

1st round notification: 14/Jun/2017
2nd round notification: 29/Set/2017
3rd round notification: 24/May/2018
Available online: 05/Jun/2018
Publication date: 30/Jun/2018

Classificação Automática de Códigos NCM Utilizando o Algoritmo Naïve Bayes

Title: Automatic Classification of NCM Codes Using the Naïve Bayes Algorithm

Rodrigo de Abreu Batista[1,3], Daniela D. S. Bagatini[1,4,5], Rejane Frozza[1,2]

[1]Departamento de Computação e [2]Programa de Pós-Graduação em Sistemas e Processos Industriais – Universidade de Santa Cruz do Sul (UNISC)
Av. Independências, 2293 – 96.815-900 – Santa Cruz do Sul – RS – Brazil

[3]Companhia de Processamento de Dados do Estado do RS – PROCERGS
Praça dos Açorianos S/N – 90010-340 – Porto Alegre – RS – Brazil

[4]Programa de Pós-Graduação em Informática na Educação – Universidade Federal do Rio Grande do Sul (UFRGS)
Av. Paulo Gama, 110 – 90040-060 – Porto Alegre – RS – Brazil

[5]Centro Universitário FADERGS
R. General Vitorino, 25 – 90.020-171 – Porto Alegre – RS – Brazil

rabatista@inf.ufrgs.br, {bagatini, frozza}@unisc.br

Abstract. *This paper consists of the development of a classifier for the automatic categorization of product item descriptions into their appropriate Common Mercosul Nomenclature (NCM) codes. This classifier was developed using the Naïve Bayes supervised learning algorithm. For training, data from items of consumer invoices belonging to chapters 22 and 90 of the NCM were used. The results evidenced the capacity of the model to correctly classify the instances. For the simpler and easier data set, based on chapter 22, an accuracy of 98% was obtained, while for the medium and difficult sets, based on chapters 22 and 90, the accuracy obtained was 90% and 83%, respectively.*

Keywords*. Machine Learning; Consumer Product Classification; NCM; Text Classification; Naïve Bayes Algorithm.*

Resumo. *Esse artigo consiste no desenvolvimento de um classificador para a categorização automática de descrições de itens de produto em seus códigos de Nomenclatura Comum do Mercosul (NCM). O desenvolvimento desse classificador foi realizado utilizando aprendizado supervisionado com o algoritmo Naïve Bayes. Para o treinamento, foram utilizados dados de itens de notas fiscais ao consumidor pertencentes aos capítulos 22 e 90 do NCM. Os resultados evidenciaram a capacidade do modelo em classificar corretamente as instâncias. Para o conjunto de dados mais simples, baseado*

Cite as: Batista, R. A., Bagatini, D. D. S. & Frozza, R. (2018). Automatic Classification of NCM Codes Using the Naïve Bayes Algorithm (Classificação Automática de Códigos NCM Utilizando o Algoritmo Naïve Bayes). iSys: Revista Brasileira de Sistemas de Informação (Brazilian Journal of Information Systems), 11(2), 4-29.

no capítulo 22, obteve-se uma acurácia de aproximadamente 98%, enquanto para os conjuntos médio e difícil, baseados nos capítulos 22 e 90, as acurácias obtidas foram de 90% e 83%, respectivamente.

Palavras-Chave. *Aprendizagem de máquina; Classificação de Produto; NCM; Classificação de Texto; Algoritmo Naïve Bayes.*

1. Introdução

A criação da versão eletrônica dos documentos fiscais tornou possível a automatização de processos, proporcionando maior agilidade e confiabilidade das atividades contábeis e fiscais. A Figura 1 exibe um documento de Nota Fiscal Eletrônica ao Consumidor (NFC-e). Esse documento é a versão impressa do documento digital que está disponível no site da Receita Estadual (Rio Grande do Sul, Brasil) através do site http://www.sefaz.rs.gov.br/NFCE, mediante a utilização da chave de acesso informada no corpo do documento.

A versão digital desse documento apresenta informações detalhadas da nota e itens de produtos, que são omitidas na versão impressa, como o código NCM (Nomenclatura Comum do Mercosul) dos itens e as alíquotas de ICMS aplicadas. O NCM trata-se de um código hierárquico de categorias que classifica os produtos. A cada código de NCM estão atreladas descrições e alíquotas de ICMS. Por exemplo, a Tabela 1 exibe as informações NCM, Descrição e TEC (Tarifa Externa Comum) e corresponde à parte do Capítulo 22, Posição 04 da tabela de NCM, correspondente a vinhos de uvas frescas, vinhos enriquecidos com álcool e mostos de uvas, extraída do site da Receita Estadual.

DIMED S/A DISTRIBUIDORA DE MEDICAMENTOS F. 101 POA

CNPJ: 92.665.611/0134-06 Inscrição Estadual: 0962203505

PRAIA DE BELAS, 1181, PRAIA DE BELAS, PORTO ALEGRE, RS

DANFE NFC-e - Documento Auxiliar da Nota Fiscal Eletrônica para Consumidor Final - Via Consumidor

NFC-e não permite aproveitamento de crédito de ICMS

Emissão normal

NFC-e nº: 17690 Série: 1 Data de Emissão: 02/07/2015 12:10:29

Consulte pela Chave de Acesso em https://www.sefaz.rs.gov.br/NFCE

CHAVE DE ACESSO

Protocolo de Autorização: 143150042457601

CONSUMIDOR

Código	Descrição	Qtde	Un	Vl Unit	Vl Total
7081	TANDRILAX 30 CP	1	CX	24	24,00

Valor total R$	22,50
Valor descontos R$	1,50
FORMA PAGAMENTO	VALOR PAGO R$
Cartão de Débito	22,50

Versão XSLT: 1.10

Figura 1. Exemplo de documento de Nota Fiscal Eletrônica ao Consumidor (Fonte: SEFAZ RS - https://www.sefaz.rs.gov.br/NFE/NFE-NFC.aspx)

Fica a critério de cada estabelecimento comercial informar a descrição dos produtos que aparecerão na Nota Fiscal do Consumidor (NFC-e) e vincular ao produto descrito o correspondente código NCM. Cada NCM possui um valor de alíquota

atrelado, correspondente ao imposto que incidirá sobre a venda do produto. Alguns contribuintes, agindo por falta de conhecimento ou intencionalmente, associam códigos NCM que não correspondem, de fato, aos itens de produtos descritos, de modo que alíquotas distintas incidem sobre os mesmos. Dessa forma, o Estado deixa de arrecadar o referido imposto e não consegue autuar o contribuinte envolvido.

As divergências existentes entre descrições de produtos e códigos NCM consistem no problema que originou o propósito desta pesquisa. O presente artigo busca responder a questão relacionada à como classificar automaticamente os itens de NFC de modo a tornar possível a identificação dessas divergências. Desta forma, procura contribuir em duas frentes: (1) no campo tecnológico, ao abordar o tema classificação (uma das atividades de aprendizagem de máquina) aplicado ao setor público, onde no Brasil recém surgem às primeiras aplicações dessa tecnologia; (2) no campo socioeconômico, uma vez que ao munir o governo com mecanismos eficientes no combate à corrupção, também auxilia no aumento da arrecadação e, consequentemente, no aumento dos investimentos e benefícios para a população.

Portanto, a solução proposta foi classificar os itens de NFC-e com base nas descrições textuais dos produtos, atribuindo os códigos NCM correspondentes. Para isso, desenvolveu-se um classificador, utilizando aprendizado supervisionado com o algoritmo Naïve Bayes, que com base na descrição do produto, informa o código NCM ao qual o mesmo deve pertencer. Esse classificador foi testado sobre registros de transações obtidos a partir do banco de dados da Secretaria da Fazenda do Rio Grande do Sul (SEFAZ-RS).

Tabela 1 - Trecho do Capítulo 22, Posição 04 da tabela de NCM, destinada a vinhos de uvas frescas, vinhos enriquecidos com álcool e mostos de uvas.

NCM	DESCRIÇÃO	TEC (%)
22.04	**Vinhos de uvas frescas, incluindo os vinhos enriquecidos com álcool; mostos de uvas, excluindo os da posição 20.09.**	
2204.10	- Vinhos espumantes e vinhos espumosos	
2204.10.10	Tipo champanha (*champagne*)	20
2204.10.90	Outros	20
2204.2	- Outros vinhos; mostos de uvas cuja fermentação tenha sido impedida ou interrompida por adição de álcool:	
2204.21.00	-- Em recipientes de capacidade não superior a 2 litros	20
2204.29	-- Outros	
2204.29.1	Vinhos	
2204.29.11	Em recipientes de capacidade não superior a 5 litros	20
2204.29.19	Outros	20
2204.29.20	Mostos	20
2204.30.00	- Outros mostos de uvas	20

Fonte: http://www.mdic.gov.br/comercio-exterior/estatisticas-de-comercio-exterior-9/

O artigo está organizado nas seguintes seções: A seção 2 realiza a fundamentação necessária para o estudo; a seção 3 discute a abordagem proposta, o método, a amostra e a avaliação do modelo; a seção 4 apresenta o experimento e os resultados; por fim, as conclusões e trabalhos futuros na seção 5.

2. Definições

Esta seção discute conceitos e técnicas utilizados na pesquisa, assim a Subseção 2.1 apresenta a Nomenclatura Comum do Mercosul (NCM), histórico e importância. A Subseção 2.2 descreve e formaliza a atividade de classificação de dados, o teorema de Bayes e o classificador Naïve Bayes. A Subseção 2.3 introduz os conceitos e técnicas envolvidas na atividade de processamento de texto que são utilizados no pré-processamento dos documentos pelo classificador. A Subseção 2.4 aborda as métricas utilizadas na avaliação de modelos, juntamente da metodologia de validação cruzada.

2.1. NCM – Nomenclatura Comum do Mercosul

A Nomenclatura Comum do Mercosul, mais comumente conhecida pelo acrônimo NCM, é uma convenção de categorização de mercadorias adotada desde 1995 pelos países que compõem o bloco econômico do Mercosul, sendo esses Uruguai, Paraguai, Brasil e Argentina. O NCM toma como base e estende o Sistema Harmonizado (SH), criado em 1983 pela Organização Mundial das Alfândegas (OMA) e utilizado internacionalmente para padronizar e classificar produtos de importação e exportação (RECEITA FEDERAL, 2015).

Os códigos do NCM são compostos por oito dígitos, sendo os seis primeiros dígitos herdados do Sistema Harmonizado, acrescido de dois dígitos, que são específicos do âmbito do Mercosul. A Figura 1 ilustra como é composta essa codificação. Os seis primeiros dígitos correspondem a três pares de dígitos que especificam o Capítulo, a Posição e a Subposição, todos esses originados do Sistema Harmonizado. O sétimo e oitavo dígitos existem exclusivamente na codificação NCM, e referem-se ao Item e Subitem, respectivamente. O NCM está estruturado em 99 Capítulos, que dizem respeito a categorias gerais, e que estão agrupados em 21 seções. Os Capítulos estão organizados de acordo com um ordenamento lógico, em ordem crescente de sofisticação ou participação humana na produção do bem. Dessa forma, o primeiro capítulo refere-se à categoria Animais Vivos, enquanto que o último se refere a Obras de Arte. O Capítulo 77 permanece em branco e está destinado a uma eventual utilização futura pelo Sistema Harmonizado. Os Capítulos 98 e 99 constam como reservados para usos especiais pelas partes contratantes. No Brasil, o Capítulo 99 é utilizado para registrar operações especiais na atividade de exportação (Ministério da Indústria, Comércio Exterior e Serviços, 2016).

Figura 1 – Composição do código NCM como extensão do Sistema Harmonizado SH (Fonte: https://www.significados.com.br/ncm/)

2.2. Classificação de Dados

A classificação é uma das atividades de mineração de dados que consiste em associar rótulos de classe a um conjunto de instâncias (casos) não classificadas. Essa classificação pode ser de dois tipos, podendo ser supervisionada e não supervisionada. Na classificação supervisionada, o conjunto de possíveis classes é conhecido previamente, enquanto que na classificação não supervisionada (*clustering*), o conjunto de classes possíveis não é conhecido (Russell e Norvig, 2003; Korde e Mahender, 2012; Ko e Seo, 2000; McCallum, 1999). Como solução de classificação utilizou-se a abordagem de classificação supervisionada de documentos. O problema de classificação pode ser formalizado como segue (Manning *et al.*, 2008), dados:

- espaço de documentos X: os documentos são representados nesse espaço – tipicamente, algum espaço com grande número de dimensões.
- conjunto finito de classes $C = \{c_1, c_2, \dots, c_m\}$: as classes são definidas manualmente, de acordo com as necessidades da aplicação (ex. relevante vs. não-relevante).
- conjunto de treinamento D de documentos classificados, com cada documento classificado $\langle d, c \rangle \in X \times C$.

Usando o Algoritmo Naïve Bayes, foi construído um classificador Υ capaz de mapear documentos em classes $\Upsilon: X \to C$. A aplicação do modelo treinado Υ a uma descrição $d \in X$ de um documento ocorre através da determinação de $\Upsilon(d) \in C$, que representa a classe mais apropriada para d (Manning *et al.*, 2008).

2.2.1. Teorema de Bayes

O conceito de probabilidade condicional, introduzido pela estatística elementar, define que a probabilidade condicional de um evento é a probabilidade obtida pela informação adicional de algum outro evento que já ocorreu. A Equação 1 pode ser utilizada para encontrar $\boldsymbol{P(D|h)}$, ou seja, a probabilidade de $\boldsymbol{D}$ dada a ocorrência de $\boldsymbol{h}$ (Triola, 2008). Por exemplo, a probabilidade de a descrição do produto possuir o termo "vinho", dado que o item pertence ao NCM 2204.29.11.

$$P(D|h) = \frac{P(h\ e\ D)}{P(h)} \tag{1}$$

Na teoria da probabilidade, o Teorema de Bayes mostra a relação entre uma probabilidade condicional e sua inversa, isso é, a probabilidade de uma hipótese $\boldsymbol{h}$ ser verdadeira dada observação $\boldsymbol{D}$ de uma evidência, e a probabilidade da evidência $\boldsymbol{D}$ dada pela hipótese $\boldsymbol{h}$ (Triola, 2008). Por exemplo, a probabilidade de um produto pertencer ao NCM 2204.29.11, dado que sua descrição possui o termo "vinho". Dessa forma, a probabilidade de ocorrência de um evento $\boldsymbol{h}$, dada a observação de $\boldsymbol{D}$, pode ser obtida pelo cálculo de probabilidade representado na Equação 2, onde $\boldsymbol{P(h)}$ e $\boldsymbol{P(D)}$ são as probabilidades *a priori* de $\boldsymbol{h}$ e $\boldsymbol{D}$, e $\boldsymbol{P(D|h)}$ e $\boldsymbol{P(h|D)}$ são as probabilidades *a posteriori* de $\boldsymbol{D}$ condicionada a $\boldsymbol{h}$ e de $\boldsymbol{h}$ condicionada a $\boldsymbol{D}$, respectivamente (Mitchell, 1997).

$$P(h|D) = \frac{P(D|h).P(h)}{P(D)} \tag{2}$$

2.2.2. Classificador Naïve Bayes

Em aprendizagem de máquina, classificadores Naïve Bayes são uma família de classificadores probabilísticos baseados na aplicação do Teorema de Bayes. O classificador leva em seu nome o adjetivo *naïve* (ingênuo), por apresentar a forte premissa de existir independência entre as características (*features*) das instâncias a serem classificadas. Classificadores Naïve Bayes são altamente escaláveis e, por isso, adequados para serem utilizados em aplicações on-line e de tempo real. O treinamento de um modelo é feito através da avaliação de uma expressão fechada, que possui tempo linear, em vez de aproximações iterativas e custosas utilizadas por outros tipos de classificadores (Russell e Norvig, 2003).

Classificadores, portanto, são modelos que associam rótulos a instâncias-problema, representadas por vetores de características. Um vetor de características é um vetor $\boldsymbol{x} = (\boldsymbol{x_1}, \dots, \boldsymbol{x_n})$ que representa as $\boldsymbol{n}$ características (variáveis independentes) de cada uma das instâncias do conjunto a ser classificado. Não existe um único algoritmo para treinar esses classificadores, mas uma família de algoritmos que podem ser utilizados, dos quais para esse trabalho foi selecionado o Naïve Bayes. O algoritmo Naïve Bayes assume como premissa o fato de todas as variáveis serem independentes (Manning *et al.*, 2008). Por exemplo, uma fruta pode ser considerada uma maçã se ela é vermelha, redonda e possui cerca de 10 cm de diâmetro. Um classificador Naïve Bayes considera que cada uma dessas características contribui independentemente para a probabilidade dessa fruta ser uma maçã, desconsiderando qualquer possível correlação existente entre as características cor, formato e diâmetro.

Naïve Bayes tem sido estudado extensivamente desde a década de 50, tendo sido introduzido sob um nome diferente na comunidade de recuperação de informação no início dos anos 60, permanecendo um método popular e usado como *baseline* para a classificação de textos (Russell e Norvig, 2003). O problema da classificação de texto consiste em categorizar documentos como pertencendo a uma categoria ou outra, utilizando a frequência de ocorrência das palavras como características. A frequência de ocorrência das palavras é realizada em uma etapa prévia à classificação, e deve ser feita para cada texto a ser categorizado. A probabilidade de um documento $\boldsymbol{d}$ pertencer à classe $\boldsymbol{c}$ pode ser computada conforme a Equação 3, onde (Manning *et al.*, 2008):

- n_d: refere-se à quantidade de termos únicos (*tokens*) em $\boldsymbol{d}$;
- $P(t_k|c)$: refere-se à probabilidade condicional do termo t_k ocorrer em um documento da classe c. Também pode ser visto como uma medida de quanto a presença de t_k contribui para determinar que c é a classe correta do documento;
- $P(c)$: refere-se à probabilidade *a priori* de c.

$$P(c|d) \propto P(c) \prod_{1 \le k \le n_d} P(t_k|c) \tag{3}$$

Além disso, caso os termos de um documento não forneçam evidências suficientes para determinar a qual classe o documento pertence, atribui-se a classe $\boldsymbol{c}$ com maior $\boldsymbol{P(c)}$. Como o objetivo com o algoritmo Naïve Bayes é encontrar a melhor

classe para o documento, muito provavelmente deseja-se obter a classe máxima *a posteriori* $\boldsymbol{c_{map}}$, ou seja, a classe com maior probabilidade quando as probabilidades condicionais são consideradas. Dessa forma, a obtenção de $\boldsymbol{c_{map}}$ pode ser obtida conforme a Equação 4 (Manning *et al.*, 2008).

$$c_{map} = arg \max_{c \in C} \hat{P}(c|d) = arg \max_{c \in C} \hat{P}(c) \prod_{1 \leq k \leq n_d} \hat{P}(t_k|c) \tag{4}$$

A multiplicação de um número grande de probabilidades pode resultar em um número muito pequeno, ocasionando em um possível *underflow* de ponto flutuante. Uma maneira de contornar isso é através do uso de logaritmos. Utilizando-se da propriedade de logaritmos expressa na Equação 5 e, sendo $\boldsymbol{log}$ uma função monotônica, pode-se utilizar desse artifício para a identificação da classe com maior escore, uma vez que a classe com maior probabilidade não mudará após a aplicação do logaritmo. A equação modificada para cálculo de $\boldsymbol{c_{map}}$ após a aplicação do logaritmo está descrita na Equação 6 (Manning *et al.*, 2008).

$$\log(xy) = log(x) + \log(y) \tag{5}$$

$$c_{map} = arg \max_{c \in C} \left[\log \hat{P}(c) + \sum_{1 \leq k \leq n_d} \log \hat{P}(t_k|c) \right] \tag{6}$$

Na Equação 6, a contribuição de cada parâmetro condicional, representado por $\boldsymbol{\log \hat{P}(t_k|c)}$, reflete o quão bom é para o indicador a presença do *token* $\boldsymbol{t_k}$ para a classe $\boldsymbol{c}$. Já a probabilidade *a priori*, representada na equação pela componente $\boldsymbol{\log \hat{P}(c)}$, é uma medida que reflete exclusivamente a frequência relativa de $\boldsymbol{c}$. Dessa forma, a soma da probabilidade *a priori* com os pesos dos termos é uma medida de quanta evidência existe para afirmar que um documento pertence a uma determinada classe $\boldsymbol{c}$. Ao final do processo, $\boldsymbol{arg \max_{c \in C}}$ refere-se à classe com maior evidência $\boldsymbol{c_{map}}$. Os parâmetros $\boldsymbol{\hat{P}(c)}$ e $\boldsymbol{\hat{P}(t|c)}$ podem ser estimados a partir das Equações 7 e 8, respectivamente, onde (Manning *et al.*, 2008):

- N: refere-se ao número total de documentos;
- N_c: refere-se ao número de documentos na classe c;
- $T_{c,t}$: refere-se ao número de ocorrências de *tokens* t nos documentos de treinamento da classe c.

$$\hat{P}(c) = \frac{N_c}{N} \tag{7}$$

$$\hat{P}(t|c) = \frac{T_{c,t}}{\sum_{t' \in V} T_{c,t'}} \tag{8}$$

2.3. Processamento de Texto

A atividade de processamento de texto consiste na tarefa de converter texto puro armazenado em um arquivo, essencialmente uma sequência digital de bits, em uma sequência bem definida de unidades linguisticamente significativas: no nível mais baixo, caracteres representando os morfemas do sistema de escrita de uma linguagem; palavras, formadas consistindo de um ou mais caracteres; e sentenças, formadas por uma ou mais palavras (Indurkhya e Damerau, 2010). O processamento de texto é uma parte fundamental de um sistema de processamento de linguagem natural (PLN), visto que todo o tipo de análise realizada sobre um texto depende da identificação dos caracteres, palavras e sentenças realizadas nessa etapa.

A Figura 3 apresenta as etapas que consistem no pré-processamento realizado sobre as descrições de produtos de itens de NFC-e, utilizadas como entrada para a criação do modelo. O fluxo do pré-processamento inicia com a tokenização, que se refere ao processo de converter um texto em sentenças e palavras. A segmentação por palavra é realizada pela quebra da sequência de caracteres através da localização dos limites de uma palavra, ou seja, os pontos de início e fim da palavra. Para fins linguísticos e computacionais, as palavras identificadas desse modo são chamadas de *tokens*, e o processo de segmentação de palavras é chamado de tokenização (Indurkhya e Damerau, 2010).

O passo seguinte nesse processo, representado pela etapa 2, é a remoção de *stop words*[1], ou palavras de parada, e referem-se às palavras mais comuns de uma linguagem. Tais palavras são frequentemente removidas antes ou depois do processamento de texto, por pouco contribuírem, por exemplo, em análises ou classificações. Frequentemente as listas de *stop words* são criadas tendo como base análises estatísticas sobre os textos que pertencem ao conjunto de textos a serem analisados (corpus) (Leskovec *et al.*, 2014).

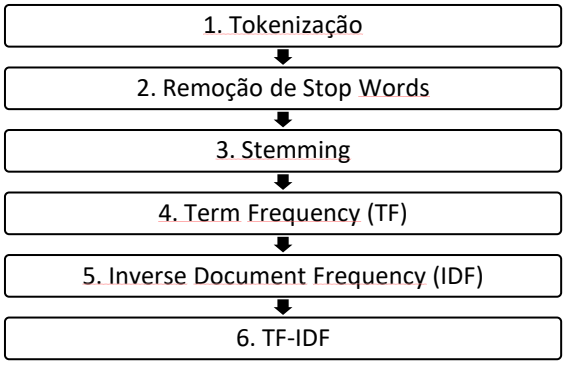

Figura 3. Etapas do pré-processamento de texto realizado sobre descrições de produtos pertencentes aos itens de NFC-e.

Na etapa 3 do pré-processamento dos documentos de entrada é aplicada a técnica de *stemming* sobre os termos que compõem as descrições dos produtos. *Stemming* é o nome dado ao processo de juntar as formas variantes de uma palavra em uma expressão comum, o radical. Radical é o elemento que contém o significado básico da palavra. A técnica de *stemming* consiste basicamente em remover os sufixos das

[1] Lista de *stopwords* disponível em http://goo.gl/i7erBE

palavras. Esse processo é largamente usado em recuperação de informações com o objetivo de aumentar a revocação, ou seja, a quantidade de documentos relevantes retornada a partir de uma palavra chave utilizada como busca (Orengo e Huyck, 2001a). Por exemplo, as palavras pedra, pedreiro, pedregulho, pedrada, apedrejar e pedreira possuem o mesmo radical, que é "ped-", possibilitando que uma consulta realizada utilizando um desses termos retorne documentos que possuam outras palavras derivadas do mesmo radical.

Nas etapas 4 e 5, são calculadas as medidas TF (*Term Frequency*) e IDF (*Inverse Document Frequency*), ambas utilizadas na etapa 6 para o cálculo da métrica TF-IDF (*Term Frequency-Inverse Document Frequency*). A medida TF corresponde à frequência com que um termo ocorre em um documento. A frequência de um termo t em um documento d, denotada por $TF(t,d)$, corresponde ao número de vezes que t ocorre em d. A medida TF normalmente é representada em número absoluto, podendo ser usada uma variação booleana, que atribui $TF(t,d) = 1$, caso t ocorra em d e $TF(t,d) = 0$, caso contrário (Luhn, 1957).

A medida IDF, corresponde ao inverso da frequência com que um dado termo ocorre nos documentos. Essa medida reflete quanto de informação cada termo fornece, isso é, se o termo é comum ou raro considerando-se a totalidade do conjunto de documentos. A Equação 9 representa o cálculo da medida IDF, onde N refere-se ao número total de documentos no corpus ($N = |D|$) e $|\{d \in D: t \in d\}|$ ao número de documentos onde o termo t aparece (Sparck Jones, 1972).

$$IDF(t,D) = \log \frac{N}{|\{d \in D: t \in d\}|} \tag{9}$$

A métrica TF-IDF, é uma medida estatística que tem como objetivo refletir o grau de importância de uma palavra em um documento, levando em consideração toda a coleção de documentos (*corpus*) sob análise (Leskovec *et al.*, 2014). O cálculo de TF-IDF se dá através do produto entre as métricas TF e IDF, conforme expresso pela Equação 10. Dessa forma, altos valores de TF-IDF serão obtidos para termos com alta frequência em um dado documento e com baixa frequência sobre a coleção de documentos. Por outro lado, TF-IDF produzirá baixos valores para termos presentes em muitos documentos. Essas características permitem que essa métrica seja adequada na identificação de termos significativos e na filtragem de termos relativamente comuns (Manning *et al.*, 2008).

$$TFIDF(t,d,D) = TF(t,d).IDF(t,D) \tag{10}$$

2.4. Avaliação do Modelo

O treinamento de um modelo e teste utilizando os mesmos dados é um erro comum de metodologia, pois ainda que um modelo consiga reproduzir os rótulos das instâncias recém treinadas com bom desempenho, ele pode falhar ao classificar dados úteis ainda não vistos. Essa situação é chamada *overfitting*, e ocorre quando um modelo se torna tão adaptado aos dados que falha ao classificar novas instâncias. No aprendizado supervisionado, uma prática comum para evitar o *overfitting* é separar os dados

disponíveis em conjuntos disjuntos para serem utilizados nas etapas de treinamento e teste.

Após o particionamento, o treinamento do modelo é realizado e os dados de teste são apresentados para avaliação do desempenho do classificador. Esta abordagem é indicada quando se tem à disposição uma grande quantidade de dados. Caso o conjunto total de dados seja pequeno, o erro gerado na rotulação pode sofrer muita variação. Quando se dispõe de um conjunto de dados reduzido, uma maneira apropriada de estimar o desempenho preditivo de um modelo é através de validação cruzada.

A validação cruzada é uma técnica para avaliar a capacidade de generalização de um modelo a partir de um conjunto de dados reduzido (Kohavi, 1995). Esse método se baseia na divisão do conjunto total de dados em outros dois subconjuntos mutuamente exclusivos, sendo o primeiro desses destinado ao treinamento e o segundo destinado ao teste.

Com o objetivo de reduzir a variabilidade, esse processo é repetido iterativamente por diversas vezes, selecionando-se a cada repetição diferentes instâncias para compor os conjuntos de treinamento e teste. Outro fator motivador para a utilização dessa técnica é a insuficiência de dados rotulados disponíveis para o particionamento do conjunto principal sem a perda da capacidade de modelagem ou teste. Para isso, a validação cruzada promove a rotatividade dos dados entre os conjuntos, permitindo que todas as instâncias participem da fase de treinamento e teste.

O método de validação cruzada denominado *K-fold* consiste em dividir o conjunto total de dados em k subconjuntos mutuamente exclusivos e de mesmo tamanho (*fold*). O primeiro desses subconjuntos é então separado para teste, enquanto que os demais $k-1$ subconjuntos são utilizados para treinamento do modelo. Ao final de cada etapa, a taxa de erro do classificador treinado pode ser calculada através da Equação 11.

O processo de treinamento e validação do modelo é realizado por k vezes, alternando-se de forma circular o subconjunto de teste, conforme representado na Figura 4. Ao final das k iterações calcula-se a acurácia média sobre os resultados encontrados, através da Equação 12, obtendo assim uma medida mais confiável sobre a capacidade do modelo de classificar corretamente os dados (Kohavi, 1995).

$$Err(\Upsilon) = \frac{1}{n}\sum_{1}^{n} \|y_i \neq \Upsilon(x_i)\| \tag{11}$$

$$Acc(\Upsilon) = 1 - Err(\Upsilon) \tag{12}$$

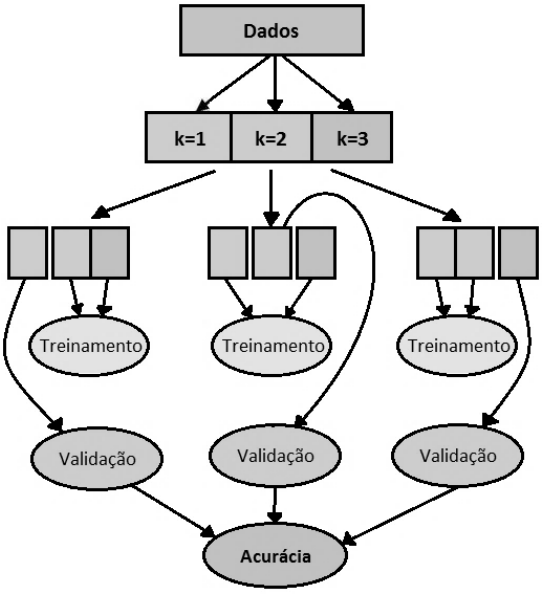

Figura 4 - Exemplo de particionamento de dados do método K-*fold* com k=3.

3. Abordagem Proposta

Esta seção tem como objetivo apresentar o método utilizado para classificação de descrições de itens de NFC em códigos de NCM correspondentes. O método consiste em treinar um classificador Naïve Bayes utilizando-se de instâncias previamente rotuladas, e avaliar o seu desempenho através da validação sobre um conjunto de instâncias de teste.

O estudo de caso foi desenvolvido sobre um subconjunto de registros do banco de dados de Notas Fiscais Eletrônicas do Consumidor da Secretaria da Fazenda do Estado do Rio Grande do Sul, sobre os quais foram aplicadas rotinas de tokenização, remoção de *stop words* e *stemming*. Após o treinamento do classificador foram utilizadas, para análise dos resultados, métricas específicas e adequadas para as técnicas escolhidas, como acurácia, erro e metodologia de validação cruzada K-*fold*, com $k = 10$ *folds*.

Essa pesquisa se deu sob a luz dos trabalhos de Manning *et al.* (2008), Leskovec *et al.* (2014) e Ding *et al*. (2015). No trabalho de Manning *et al*. (2008) foi justificado que o desempenho do Naïve Bayes é adequado para aplicações *on-line* e de tempo real por sua classificação ser realizada através do cálculo de uma fórmula fechada, em vez de métodos iterativos e aproximações de maior complexidade realizadas por outros algoritmos. Além disso, o espaço da estrutura de dados demandada pelo modelo é relativamente simples, consistindo nas probabilidades *a priori* e nas contagens de ocorrências de palavras em cada classe. No trabalho de Leskovec *et al.* (2014), intitulado *Mining of Massive Datasets*, os autores abordam especificamente a mineração de dados em bancos de dados muito grandes, e confirmam que TF-IDF é uma métrica utilizada na atividade de classificação de textos quando se lida com grandes volumes de dados. No trabalho de Ding *et al*. (2015), os autores apresentam uma solução para a classificação automática de códigos pertencentes ao Sistema Harmonizado (SH), do qual o NCM deriva, utilizando *Background Nets* como a técnica base para seu sistema

de classificação. As Background Nets consistem em redes de palavras estruturadas como grafos que capturam o domínio da informação através da associação contextual entre os termos. Nessas redes, os vértices representam as palavras, e as arestas, as relações entre essas.

3.1. Método

O método utilizado para a construção de um classificador de descrições de itens de produtos em códigos NCM foi dividido em seis etapas, sendo essas: (1) extração dos documentos NFC de sua base de origem; (2) definição dos experimentos; (3) pré-processamento dos documentos de entrada; (4) treinamento do classificador; (5) avaliação do modelo, que corresponde à classificação propriamente dita; e (6) análise dos resultados e comparação com o *baseline*. A Figura 5 ilustra as etapas e atividades que foram desenvolvidas durante o treinamento do classificador.

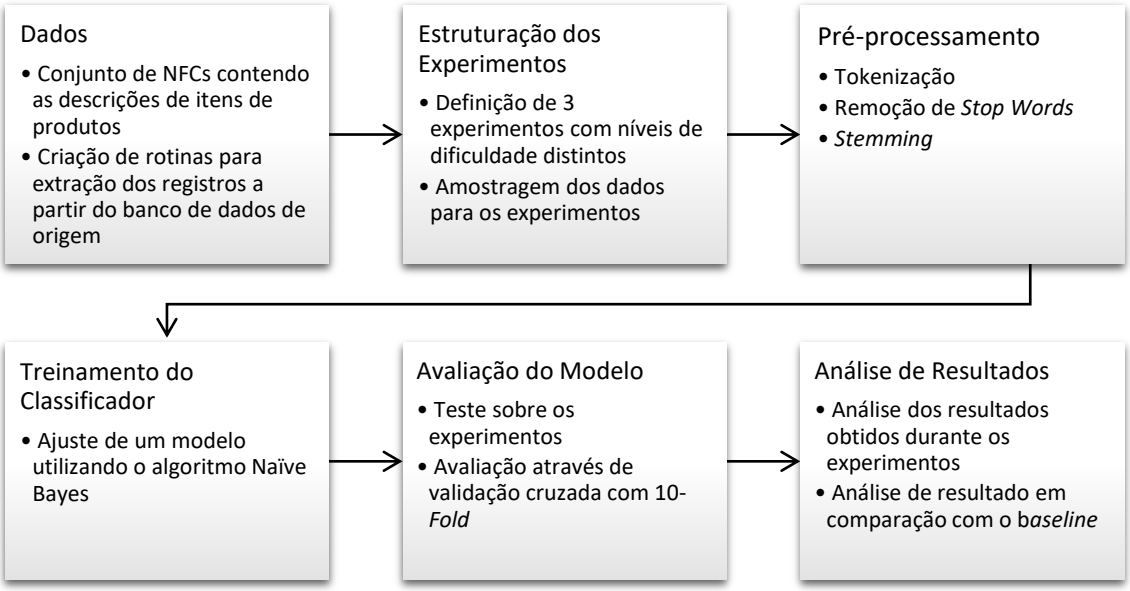

Figura 5 - Etapas para treinamento e avaliação do classificador.

O Algoritmo 1 descreve as etapas presentes no pré-processamento realizado sobre os documentos, e tem como objetivo extrair o vetor de características de cada um dos documentos de entrada. A rotina recebe um conjunto de documentos *D* como parâmetro, e inicia com a tokenização dos documentos, quebrando o texto e extraindo vetor de *tokens* correspondente. O passo seguinte do algoritmo é remoção de *stop words*. A sequência do pré-processamento se dá pela aplicação de *stemming* sobre os termos restantes nos documentos, conforme mencionado na Subseção 2.3.

Algoritmo 1 - Algoritmo para Pré-processamento dos documentos

Entrada: Conjunto de Documentos (D)
Saída: Matriz (W) de pesos correspondente aos vetores de características extraídos de D

```
1.  ROTINA ExtrairVetorDeCaracteristicas(D)
```

```
2.    Realizar tokenização nos documentos d_i ∈ D
3.    Remover Stop Words de D
4.    Realizar Stemming dos termos em D
5.    Montar uma matriz H contendo o número de ocorrências dos termos em D
6.    Para cada d_i ∈ D:
7.      Criar um vetor t contendo a frequência dos termos t_k ∈ d_i
8.      Calcular TF-IDF dos termos t_k ∈ d_i em um vetor x usando t e H
9.      Incluir x em W
10.   RETORNA W
```

Os passos seguintes do algoritmo de pré-processamento, representados nas linhas 5-9, referem-se à contabilização dos termos para calcular as métricas TF e IDF, que servirão de base para o cálculo de TF-IDF, conforme explicado na Subseção 2.3, cujos resultados produzirão os pesos dos vetores de características a serem utilizados para o treinamento do classificador.

Algoritmo 2 – Algoritmo para treinamento do classificador Naïve Bayes.

Entrada: Conjunto de Descrições (X)
Conjunto de Rótulos (Y)
Saída: Matriz contendo a contagem de instâncias pertencentes a cada uma das Classes (H)
Matriz com a Frequência de Ocorrências dos *Tokens* x Classe (F)

```
1.  ROTINA Treinar(X,Y)
2.    Criar uma matriz H com a frequência das instâncias x classes c_i ∈ Y
3.    Criar uma matriz F (tokens × c) de ocorrências de tokens x classes
    utilizando a matriz W contendo os pesos TF-IDF
4.    RETORNA H, F
```

A matriz de documentos pré-processados, composta pelos vetores de características, preenchidos com os pesos originados pelo TF-IDF, são passadas para o algoritmo de treinamento do classificador Naïve Bayes. O Algoritmo 2 descreve os passos necessários para o treinamento do classificador. O que se faz ao treinar um classificador Naïve Bayes é pré-computar as frequências de ocorrências de classes (linha 2), que serão utilizadas para cálculo das probabilidades *a priori*, e a matriz de ocorrência de *tokens* em classes (linha 3), que será utilizada no cálculo das probabilidades *a posteriori*, conforme descrito na Equação 6 da Subseção 2.2. O vetor H e a matriz F ficam disponíveis para o cálculo de probabilidades a ser realizado durante a etapa de classificação.

O Algoritmo 3 descreve os passos necessários para classificação de uma instância não rotulada x, utilizando as frequências pré-computadas armazenadas em H e F. Nas linhas 2-3 são calculadas as probabilidades de x pertencer a cada uma das classes $c_i \in C$ possíveis. Após, em um segundo passo representado na linha 4, é selecionada a classe com maior probabilidade $P(x, c_i)$ para que a classe correspondente seja retornada como rótulo pelo algoritmo. As linhas 2-4 do Algoritmo 3, portanto,

representam a aplicação do algoritmo Naïve Bayes para a classificação de uma instância não rotulada.

Algoritmo 3 – Algoritmo para classificação de instâncias não rotuladas.

Entrada: Instância não rotulada (x)
Matriz contendo a contagem de instâncias pertencentes a cada uma das Classes (H)
Matriz com a Frequência de Ocorrências dos *Tokens* x Classe (F)
Saída: Rótulo de classe (y)

```
1. ROTINA Classificar(x)
2.   Para cada c_i ∈ H:
3.     Calcular a probabilidade de x pertencer à classe c_i P(x,c_i), usando os valores computados em H e F
4.   y := max_{c_i ∈ C} P(x, c_i)
5.   RETORNA y
```

3.2. Seleção da Amostra

Os registros de transações que deram origem aos conjuntos de dados utilizados nesse trabalho foram extraídos da base de Notas Fiscais do Consumidor da SEFAZ-RS e estavam armazenados em um servidor com tecnologia Microsoft SQL Server versão 2014. Para isso, utilizou-se o recurso TABLESAMPLE[2], uma função nativa da linguagem de consulta T-SQL[3], que seleciona registros aleatoriamente. O Algoritmo 4 apresenta a consulta para a seleção das transações contendo descrições de produtos correspondentes aos Capítulos 22 e 90 do NCM, que foram usadas para treinamento e validação do modelo. Como resultado, o Algoritmo 4 retorna para cada um dos itens amostrados pertencentes aos Capítulos 22 e 90, o código do NCM (COD_NCM), a descrição do item (TEX_DESC_PSERV), a quantidade tributada (QTD_TRIB) e a unidade tributada (UNID_TRIB).

Segundo Ding et al. (2015), os itens pertencentes aos capítulos 22 e 90 do NCM são os que mais apresentam erros de classificação na prática e, portanto, são os que possuem maior probabilidade de erro. Essa possibilidade de erro pode ocorrer basicamente devido a dois fatores, sendo esses a quantidade de categorias pertencentes ao capítulo e a quantidade de tipos de produtos compreendidos. Assim, esses fatores contribuíram para que os capítulos 22 e 90 fossem escolhidos e utilizados para testar o desempenho do classificador.

Algoritmo 4 - Consulta para seleção dos itens pertencentes aos Capítulos 22 e 90 do NCM.

```
1. SELECT COD_NCM, TEX_DESC_PSERV, QTD_TRIB, UNID_TRIB
2.   FROM ADT_NFE_DF_ITEM A TABLESAMPLE (2330000 ROWS)
3.   WHERE LEFT(COD_NCM, 2) = '22'
4.
```

[2] https://technet.microsoft.com/pt-br/library/ms189108.aspx
[3] https://technet.microsoft.com/en-us/library/ms189826.aspx

```
5.  SELECT COD_NCM, TEX_DESC_PSERV, QTD_TRIB, UNID_TRIB
6.    FROM ADT_NFE_DF_ITEM A TABLESAMPLE (3400000 ROWS)
7.    WHERE LEFT(COD_NCM, 2) = '90'
```

Após a seleção da amostra, foi realizada uma análise com o objetivo de eliminar documentos que apresentassem descrições de itens duplicadas, assim como a contagem das classes existentes em cada conjunto de dados. A Tabela 3 descreve o total de registros presente em cada uma das amostras de NFC extraídas antes e depois da remoção de duplicatas, assim como a quantidade de categorias existentes em cada um. Nesse contexto, uma duplicata refere-se a itens de produtos que contêm descrições repetidas. A versão do conjunto de dados com duplicatas, refere-se, portanto, ao conjunto original amostrado no banco de dados de origem no momento da extração, enquanto a versão sem duplicatas, refere-se ao conjunto pré-processado contemplando a remoção de itens repetidos, de modo que fosse mantida apenas um registro representante por grupo.

A opção por manter nos experimentos as versões com e sem duplicatas se deu pelo seguinte motivo: a versão sem duplicatas permite uma comparação mais próxima com a utilizada como *baseline*, uma vez que esse também utiliza conjuntos de dados sem duplicatas; já a versão com duplicatas foi selecionada, pois imagina-se que, ao realizar uma seleção amostral puramente aleatória no banco de dados original de NFC, estão sendo mantidas na amostra as características do conjunto original, sem fazer, portanto, sentido algum a remoção das duplicatas por distorcer as probabilidades *a priori*.

As Figuras 6 e 7 apresentam o total de instâncias por classe para o Datasets (1) e (2) relacionados ao Capítulo 22. O gráfico referente ao Capítulo 90 foi omitido devido à quantidade elevada de classes a serem plotadas. Analisando a Figura 6 é possível notar que no *dataset* referente ao Capítulo 22 (1), apesar de aparentar a inexistência de dados para os códigos de NCM 22029001 e 22042919, existe uma instância para cada uma dessas classes. O mesmo se aplica para as classes aparentemente sem dados do *dataset* referente ao Capítulo 22 (2), ilustrado na Figura 7.

Tabela 3 - Número de registros e categorias antes e depois do pré-processamento.

Dataset, Registros, Categorias	Antes (com duplicatas)	Depois (sem duplicatas)
A - Dataset Capítulo 22 (1)		
Registros	210.836	20.259
Categorias	29	29
B - Dataset Capítulo 22 (2)		
Registros	207.310	64.269
Categorias	59	59
C - Dataset Capítulo 90		
Registros	412.969	262.244
Categorias	819	819

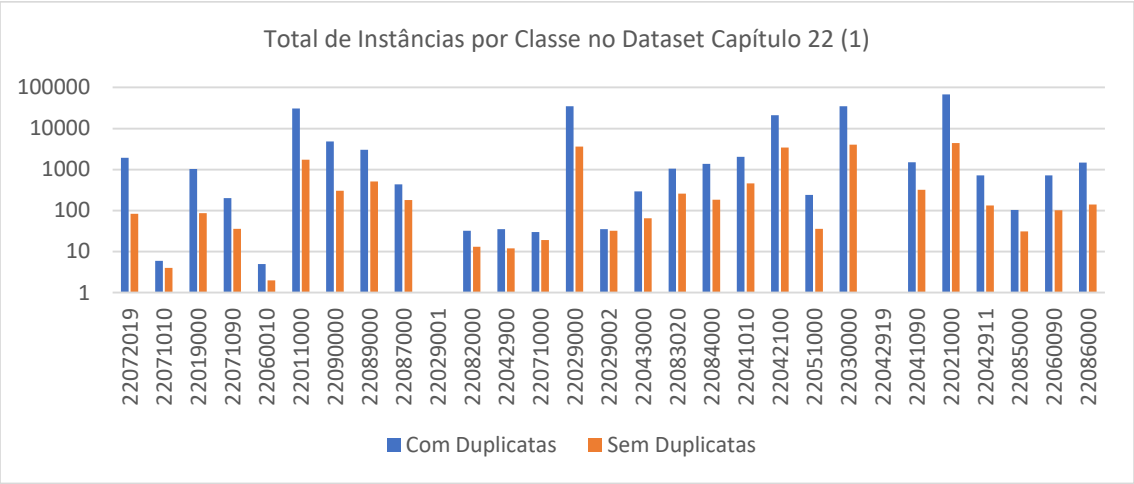

Figura 6 – Total de instâncias por classe para o DataSet Capítulo 22 (1) antes e após a remoção de duplicatas. O gráfico está em escala logarítmica.

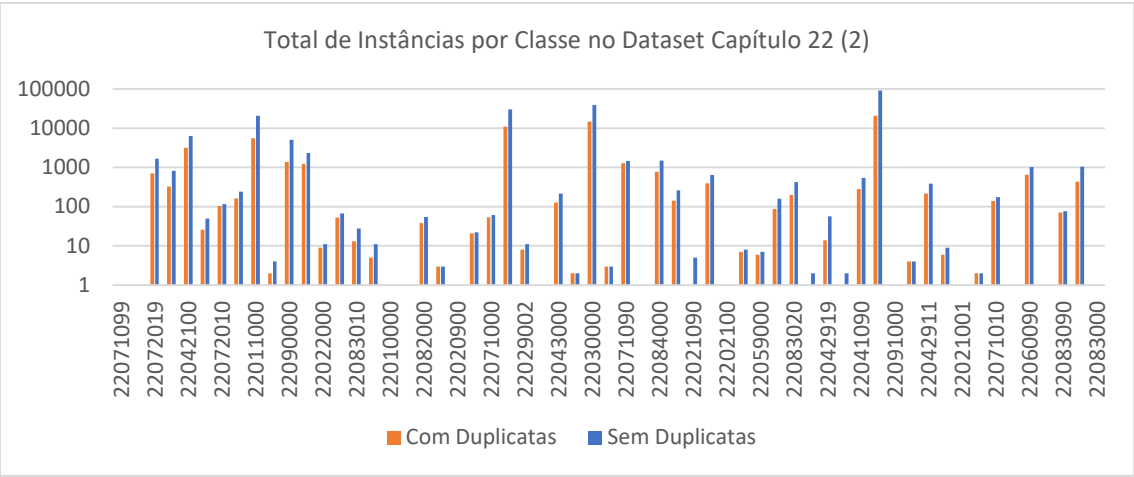

Figura 7 – Total de instâncias por classe para o DataSet Capítulo 22 (2) antes e após a remoção de duplicatas. O gráfico está em escala logarítmica.

3.3. Avaliação do Modelo

Para a realização da avaliação do modelo utilizou-se a acurácia média calculada sobre os testes parciais do método de validação cruzada K-*fold* com $k = 10$ *folds*, conforme tratado na Subseção 2.4. Apesar de existirem outras métricas de avaliação de desempenho utilizadas nas áreas de classificação de texto e recuperação de informações, tais como precisão, revocação e a média harmônica entre essas, denominada medida-F1, a métrica mais relevante para avaliar os resultados de modelos de classificação de NCM mostrou-se a acurácia (Ding *et al.*, 2015). A acurácia apresenta a proporção entre as instâncias que o modelo classificou corretamente e o total de instâncias classificadas pelo modelo.

A adoção do método de avaliação cruzada com 10-*folds*, diferente do método adotado no estudo utilizado como *baseline*, que realiza a divisão dos dados em conjuntos de 60% dos dados para treinamento e 40% para teste, auxilia em eliminar

algum viés realizado por uma eventual escolha tendenciosa dos dados. Apesar de ser mais custoso computacionalmente, o método de avaliação cruzada com 10-*folds* auxilia na redução da variância (Kohavi, 1995), além de permitir uma avaliação mais precisa da acurácia.

3.4. Implementação

O classificador Naïve Bayes desenvolvido ao longo desse estudo foi implementado na linguagem de programação Python 2.7[4]. Para as tarefas relacionadas ao processamento de texto, utilizou-se a biblioteca NLTK (Bird *et al.*, 2009). A biblioteca NLTK é uma suíte para processamento de linguagem natural que possui mais de 50 corpora de exemplo e recursos para classificação de texto, tokenização, *stemming*, rotulação, *parsing* e associação semântica. O pacote NLTK pode ser utilizado também para tradução de máquina, rotulação semântica, análise de sentimento, modelagem de tópicos e possui funções para download e processamento de páginas HTML e mensagens do Twitter.

O treinamento do classificador bayesiano, assim como a metodologia de avaliação, baseada em validação cruzada, foram implementados com o suporte das rotinas presentes na biblioteca Scikit-learn (Pedregosa *et al*., 2011). A etapa de classificação de instâncias não rotuladas, descrita no Algoritmo 3, também foi implementada utilizando funções dessa biblioteca. Scikit-learn é uma biblioteca escrita em Python e que recentemente tornou-se difundida entre os entusiastas e cientistas de dados sendo até mesmo utilizada em sistemas em produção. Essa biblioteca é utilizada para desenvolvimento de classificadores, análise de regressão, agrupamento, redução de dimensionalidade, seleção de modelos e pré-processamento de dados de entrada.

Dos recursos mencionados, o pré-processamento da entrada, tokenização, remoção de *stop words*, treinamento do modelo de classificação, estudo dos parâmetros e avaliação de desempenho utilizando validação cruzada, foram realizadas com o auxílio da biblioteca Scikit-learn. Entre os recursos oferecidos pela biblioteca NLTK, apenas o recurso de *stemming* foi utilizado. O algoritmo de *stemming* para língua portuguesa disponibilizada na biblioteca NLTK é uma implementação do proposto por Orengo e Huyck (2001b).

4. Experimentos e Resultados

Com o objetivo de testar o classificador foram desenvolvidos três experimentos que ocorreram sobre conjuntos de dados distintos compreendidos sobre transações que continham itens pertencentes a códigos de NCM dos capítulos 22 (Bebidas, líquidos alcoólicos e vinagres) e 90 (Instrumentos e aparelhos de óptica, de fotografia, de cinematografia, de medida, de controle ou de precisão; instrumentos e aparelhos médico-cirúrgicos; suas partes e acessórios).

Tendo como objetivo a avaliação do classificador com métricas e metodologia de validação aceitas pela comunidade científica e, com o objetivo de se ter um *baseline* para comparação da efetividade do método aplicado nesse estudo, optou-se por estruturar os experimentos dos conjuntos de dados utilizados no trabalho de Ding *et al.*

[4] http://www.python.org/

(2015). No trabalho mencionado, os autores construíram um classificador baseado em *Background Nets* para o Sistema Harmonizado (SH), e realizaram testes sobre descrições de itens que estavam no idioma inglês. Salvo as diferenças de idioma e o fato de o sistema NCM possuir hierarquia mais profunda do que o SH sendo, portanto, seu nível de especificidade ainda maior, tentou-se tornar tão próximas quanto possível a quantidade de registros, os códigos dos capítulos e métrica para avaliação utilizados.

O resultado dos experimentos sobre os conjuntos de dados A, B e C, estruturados na Subseção 3.2, estão descritos e discutidos individualmente nas Subseções 4.1, 4.2 e 4.3 a seguir. A Subseção 4.4 apresenta a análise geral sobre os resultados e comparação com o *baseline*.

4.1. Experimento sobre o conjunto de dados A

O primeiro experimento foi realizado sobre o conjunto de dados baseado em 210.836 transações ocorridas no ano de 2016, pertencentes a 29 NCMs do Capítulo 22 (Bebidas, líquidos alcoólicos e vinagres), selecionadas de modo aleatório, de uma grande rede varejista do Rio Grande do Sul. Por se tratar de registros de uma mesma rede de varejo, pressupõe-se que as descrições de itens sigam certa uniformidade, o que motivou o julgamento da classificação desse conjunto como fácil. Sobre esse conjunto de treinamento foi realizado o processamento de texto, a fim de se obter os vetores de características para cada instância e o pré-processamento dessas através do cálculo do TF-IDF. O conjunto de instâncias foi apresentado para o treinamento do classificador Naïve Bayes Multinomial.

Uma vez treinado o modelo do classificador, prosseguiu-se então com a avaliação de desempenho do modelo, em relação a sua capacidade em fazer classificações corretas, utilizando-se a técnica de validação cruzada com 10-*folds*. A Tabela 4 exibe o resultado do modelo para o conjunto de dados em questão. Nessa tabela é possível observar que o resultado da acurácia do classificador para o conjunto com duplicatas é igual a 99,55% de acerto, enquanto que para o conjunto sem duplicatas, a acurácia ficou em 98,68%. Esse resultado superior para o conjunto de dados com duplicatas sugere que a maior parte das duplicatas se refere a instâncias para as quais o classificador correlacionou corretamente, pois contribuíram positivamente para a métrica. No entanto, através da comparação entre as Figuras 8 e 9, também é possível notar que a versão do *dataset* sem duplicatas se saiu melhor na identificação de algumas classes, como por exemplo as que possuem códigos NCM 22029002, 22060010 e 22071000.

Tabela 4 - Acurácia média do classificador sobre o conjunto de dados A para os casos com e sem duplicatas.

Conjunto de Dados	Configuração	Média
A - Dataset Capítulo 22 (1)	com Duplicatas	0,9955
	sem Duplicatas	0,9868

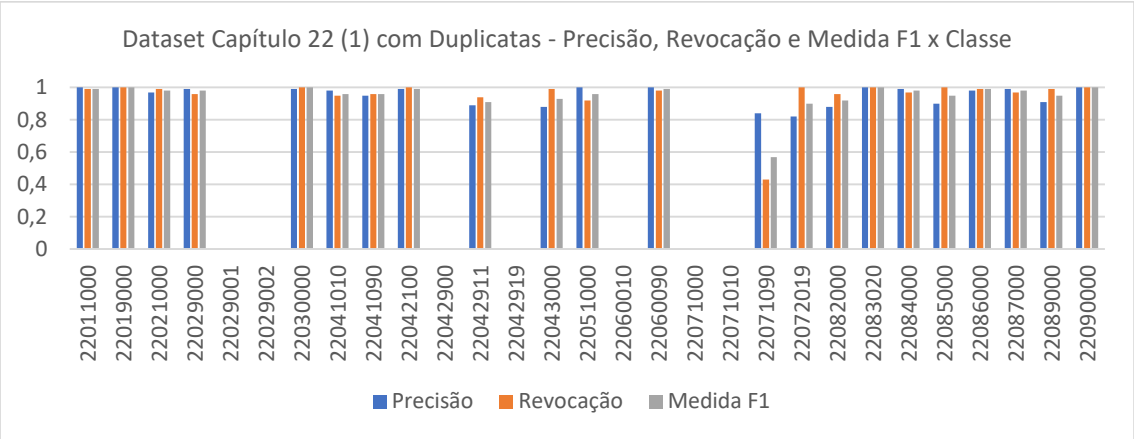

Figura 8 – Gráfico das métricas precisão, revocação e medida-F1 calculadas para o Dataset Capítulo 22 com Duplicatas.

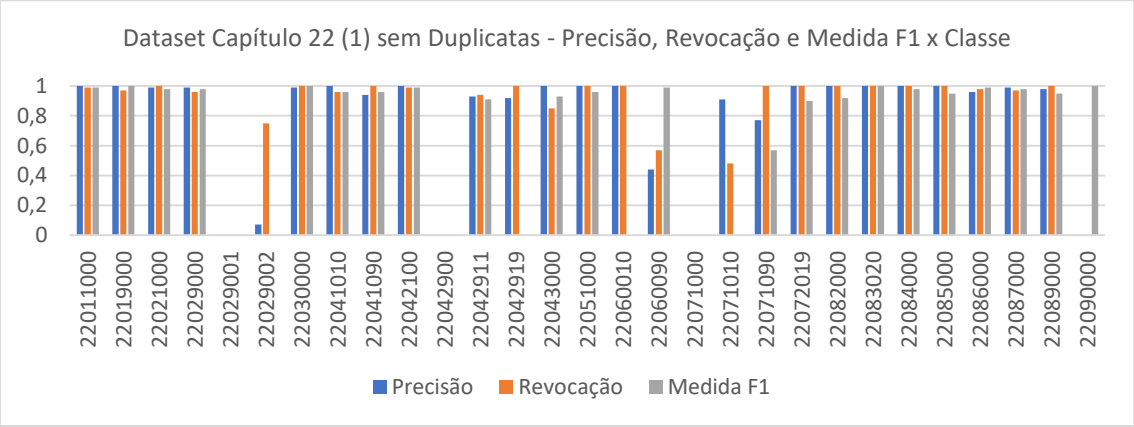

Figura 9 – Gráfico das métricas precisão, revocação e medida-F1 calculadas para o Dataset Capítulo 22 sem duplicatas.

O cálculo da acurácia foi realizado pela média das acurácias obtidas pelas avaliações dos modelos obtidos em cada uma das dez etapas da validação cruzada com 10 *folds*. O valor da acurácia para cada *fold* foi calculado durante a etapa de avaliação do experimento sobre o Conjunto de Dados A, versão sem duplicatas, e encontra-se listado na Tabela 5. O cálculo da acurácia foi realizado através das Equações 11 e 12. As Figuras 8 e 9 apresentam gráficos contendo as métricas de precisão, revocação e medida-F1 para o dataset 22 (1) para o caso com e sem duplicatas, respectivamente.

Como a Equação 11 refere-se ao cálculo do erro de classificação do modelo em relação ao rótulo original das classes informadas sobre o conjunto de teste, foi necessária a realização da contagem sobre o mesmo. A Tabela 6 exibe exemplos de instâncias correta e incorretamente classificadas.

Após o treinamento do modelo, onde são pré-calculadas as probabilidades *a priori* das classes $\hat{P}(c)$ e *a posteriori* $\hat{P}(t_k|c)$, a classificação foi realizada pela aplicação da Equação 6 sobre as instâncias do conjunto de testes. Por exemplo, para identificar a classe que pertence a instância "Guaraná Charrua 2L", deve-se calcular a probabilidade dessa instância pertencer a cada uma das classes do conjunto de dados e, por fim, tomar a classe com a maior probabilidade *a posteriori* c_{map}. Abaixo segue o

exemplo do cálculo da probabilidade da instância pertencer aos NCMs 22019000 e 22021000, representado por $\hat{P}(c|d)$.

Tabela 5 - Cálculo da acurácia por *fold* no experimento sobre o conjunto de dados A, na versão sem duplicatas.

Nro. *Fold*	Acurácia	Nro. *Fold*	Acurácia
1	0,9785	6	0,9866
2	0,9848	7	0,9871
3	0,9892	8	0,9812
4	0,9852	9	0,9916
5	0,9916	10	0,9926

Tabela 6 - Exemplos de classificações realizadas pelo modelo treinado sobre o conjunto de dados A, versão sem duplicatas.

Instância	Classe Real	Classe Prevista	Resultado
guarana antarctica zero 2l GF	22021000	22021000	Correto
bebida pessego suvalan 1l UN	22021000	22029000	Incorreto
vh marcus james cab sauvig 750ml GF	22042100	22042100	Correto
coca zero 600ml c/6 leve+ pague- CJ	22021000	22021000	Correto
esp casa perini demi-sec 750ml GF	22041010	22042100	Incorreto
filt perlage rose 660ml GF	22043000	22041010	Incorreto

NCM 22019000

$\log \hat{P}(22019000) = -5{,}5863$

$\log \hat{P}(Guaran\acute{a} \mid 22019000) = -8{,}7454$

$\log \hat{P}(Charrua \mid 22019000) = -9{,}0524$

$\log \hat{P}(2L \mid 22019000) = -8{,}6732$

$$\hat{P}(c|d) = \log \hat{P}(c) + \sum_{1 \le k \le n_d} \log \hat{P}(t_k|c)$$

$$\hat{P}(22019000 \mid Guaran\acute{a}\ Charrua\ 2L)$$
$$= \log \hat{P}(22019000) + \log \hat{P}(Guaran\acute{a} \mid 22019000) + \log \hat{P}(Charrua \mid 22019000)$$
$$+ \log \hat{P}(2L \mid 22019000)$$
$$\hat{P}(22019000 \mid Guaran\acute{a}\ Charrua\ 2L) = -5{,}5863 - 8{,}7454 - 9{,}0524 - 8{,}6732$$
$$\hat{P}(22019000 \mid Guaran\acute{a}\ Charrua\ 2L) = -32{,}0574$$

NCM 22021000

$\log \hat{P}(22021000) = -0{,}8420$

$\log \hat{P}(Guaran\acute{a} \mid 22021000) = -4{,}3790$

$\log \hat{P}(Charrua \mid 22021000) = -8{,}2765$

$\log \hat{P}(2L \mid 22021000) = -2{,}9343$

$$\hat{P}(c|d) = \log \hat{P}(c) + \sum_{1 \le k \le n_d} \log \hat{P}(t_k|c)$$

$$\begin{aligned}
&\hat{P}(22021000|\ Guaran\acute{a}\ Charrua\ 2L) \\
&\quad = \log \hat{P}(22021000) + \log \hat{P}(Guaran\acute{a} \mid 22021000) + \log \hat{P}(Charrua \mid 22021000) \\
&\quad + \log \hat{P}(2L \mid 22021000) \\
&\hat{P}(22021000|\ Guaran\acute{a}\ Charrua\ 2L) = -0{,}8420 - 4{,}3790 - 8{,}2765 - 2{,}9343 \\
&\hat{P}(22021000|\ Guaran\acute{a}\ Charrua\ 2L) = -16{,}4319
\end{aligned}$$

No caso exemplificado, o item "Guaraná Charrua 2L" pertenceria então ao NCM 22021000, que corresponde à classe com maior probabilidade calculada. De fato, a classe 22021000 é a classe correta para esse item. No exemplo foram exibidos os cálculos para apenas duas classes, mas no caso real devem ser calculadas as probabilidades *a posteriori* para todas as classes de modo a certificar-se que não existe nenhuma outra classe com probabilidade de ocorrência maior. Ainda no exemplo, optou-se por utilizar as formas completas dos termos "Guaraná", "Charrua" e "2L" para fins didáticos. Dentro do classificador, os cálculos são feitos utilizando as versões dos termos que foram reduzidas pela etapa de *stemming* descrita na Subseção 2.3.

4.2. Experimento sobre o conjunto de dados B

O segundo experimento foi realizado sobre um conjunto de dados baseado em 207.310 registros de transações ocorridas no ano de 2016, distribuídas em 29 códigos distintos de NCMs e pertencentes, assim como o conjunto de dados anterior, ao Capítulo 22. No entanto, diferente do conjunto de dados do experimento anterior, as transações selecionadas nesse caso foram de estabelecimentos comerciais aleatoriamente selecionados. Com isso, espera-se que as descrições de itens, apesar de pertencerem todas ao mesmo Capítulo, possuam maiores divergências e variações em suas descrições, motivando a classificação desse conjunto de dados como dificuldade média.

Ao treinar o classificador e prosseguir com a avaliação de desempenho do modelo, obteve-se os resultados apresentados na Tabela . Confirmando a suspeita de tratar-se de um conjunto de dados mais difícil do que o utilizado no experimento anterior, observou-se uma queda no desempenho do classificador. Mais uma vez o conjunto de dados com duplicatas apresentou um desempenho melhor, atingindo uma acurácia média de 95,60%. Para o conjunto de dados sem duplicatas, o desempenho do classificador atingiu 90,30%.

Tabela 7 - Acurácia média do classificador sobre o conjunto de dados B para os casos com e sem duplicatas.

Conjunto de Dados	Configuração	Média
B - Dataset Capítulo 22 (2)	com Duplicatas	0,9560
	sem Duplicatas	0,9030

4.3. Experimento sobre o conjunto de dados C

O terceiro e último experimento foi realizado sobre um conjunto de dados baseado em 412.969 registros de transações ocorridas no ano de 2016 e distribuídas em 819 códigos distintos de NCMs, pertencentes ao Capítulo 90. As transações selecionadas para esse experimento pertenciam a estabelecimentos comerciais do Rio Grande do Sul, aleatoriamente selecionados. Além disso, existe um aumento na complexidade desse conjunto de dados, devido ao fato da quantidade elevada de classes (códigos NCM). Com isso, espera-se que as descrições de itens, apesar de ainda pertencerem todas a um mesmo capítulo, sejam mais difíceis de serem classificadas, devido à especificidade da hierarquia de classes, fazendo com que esse conjunto de dados fosse classificado como difícil.

A Tabela exibe os resultados do desempenho do classificador para o terceiro experimento. Conforme o esperado, por se tratar de um conjunto de dados mais difícil, observou-se uma queda nos valores de acurácia, tanto para o caso com duplicatas (88,22%), quanto para o caso sem duplicatas (83,38%), permanecendo o caso com duplicatas apresentando desempenho superior.

Tabela 8 - Acurácia média do classificador sobre o conjunto de dados C para os casos com e sem duplicatas.

Conjunto de Dados	Configuração	Média
C - Dataset Capítulo 90	com Duplicatas	0,8822
	sem Duplicatas	0,8338

4.4. Resultados

Ao confrontar os resultados obtidos durante a avaliação do classificador com os resultados do estudo definido como *baseline*, observou-se que os resultados ficaram ligeiramente abaixo ao comparar o conjunto de dados B, sem duplicatas, da Tabela 9 (90,30%), com o conjunto de dados X, da Tabela 10 (98,56%), que correspondem aos experimentos sobre o Capítulo 22. No entanto, quando se comparou os experimentos relacionados ao Capítulo 90, que corresponde ao conjunto de dados difícil, notou-se um desempenho equivalente. Os resultados do classificador desenvolvido nesse estudo estão exibidos no conjunto de dados C, sem duplicatas, da Tabela 10 (83,38%), enquanto que os dados do *baseline* estão dispostos no conjunto de dados Y da Tabela 11 (83,30%).

Entre os fatores que podem justificar o desempenho inferior para o primeiro caso (conjunto de dados B x conjunto de dados X), está o fato de se estar comparando dados do sistema NCM com dados do sistema harmonizado. Enquanto o sistema harmonizado possui 6 dígitos, o sistema NCM possui 8 dígitos, sendo uma hierarquia mais profunda e, portanto, mais específica, fazendo com que a mesma tenha mais classes e, portanto,

seja mais difícil de classificar. Isso pode ser constatado, pois no conjunto de dados desse experimento, o Capítulo 22 possui 59 classes, versus 52 classes existentes no *baseline*, enquanto que para o Capítulo 90, o conjunto de dados desse experimento possui 819 classes, versus 204 classes do conjunto de dados utilizado como *baseline*, conforme evidenciado na Tabela 11.

Tabela 9 - Acurácia média do classificador para os conjuntos de dados A, B e C para os casos com e sem duplicatas.

Conjunto de Dados	Configuração	Média
A - Dataset Capítulo 22 (1)	com Duplicatas	0,9955
	sem Duplicatas	0,9868
B - Dataset Capítulo 22 (2)	com Duplicatas	0,9560
	sem Duplicatas	0,9030
C - Dataset Capítulo 90	com Duplicatas	0,8822
	sem Duplicatas	0,8338

Tabela 2 - Acurácia Média dos Capítulos 22 e 90 do SH para o classificador desenvolvido por Ding et al. (2015).

Conjunto de Dados	Configuração	Média
X – Dataset Capítulo 22 do SH	sem Duplicatas	0,9856
Y – Dataset Capítulo 90 do SH	sem Duplicatas	0,8330

Tabela 3 - Comparação entre a quantidade de classes nos conjuntos de dados utilizados nos experimentos, em relação à quantidade de classes utilizada no *baseline*.

Capítulo \ Quantidade de Classes	Experimentos (NCM)	*Baseline* (SH)
Capítulo 22	59	52
Capítulo 90	819	204

Outro fator que pode explicar a diferença de desempenho entre os classificadores é o fato do experimento desenvolvido nesse estudo estar utilizando descrições de produtos escritas no idioma Português, enquanto que os conjuntos de dados do *baseline* possuíam descrições em inglês. Esse fator é muito importante e não pode ser ignorado, uma vez que a língua portuguesa é mais complexa e mais rica em variações linguísticas sendo, certamente, mais difícil de classificar.

Além disso, outro fator a ser questionado é se está correta a remoção das descrições de produtos duplicadas, uma vez que se está utilizando um classificador com base estatística, que armazena as probabilidades *a priori* de cada classe e encontrará essa mesma distribuição de valores quando tornar a olhar os registros de transações do mundo real. Como o método de seleção dos dados é aleatório, o fato de existir uma distribuição desigual de itens entre as classes está evidenciando o fato de que aquela classe é mais provável de ocorrer, que é uma característica do conjunto de dados original que está sendo representada na amostra.

5. Conclusão

A motivação para este estudo está em apresentar a aplicabilidade e os resultados da classificação automática de descrições textuais de itens de produtos em seus respectivos códigos NCM. A divergência de classificação de itens de produtos nas NFC-e informadas pelos estabelecimentos comerciais em relação aos códigos reais aos quais os produtos pertencem é um dilema real enfrentado pela Secretaria da Fazenda do Rio Grande do Sul (SEFAZ-RS). Desta forma, o estudo envolveu pesquisa relacionada às técnicas de processamento de texto, treinamento de classificadores utilizando aprendizado supervisionado e métodos de avaliação.

O classificador desenvolvido utilizou o algoritmo Naïve Bayes para a construção de seu modelo de classificação. O método proposto foi testado em três conjuntos de dados experimentais: o primeiro deles, considerado fácil, composto por transações pertencentes a NCMs do Capítulo 22 de uma grande rede varejista; o segundo conjunto de dados, considerado de dificuldade intermediária, composto por descrições de itens pertencentes a NCMs do Capítulo 22, pertencentes a transações de estabelecimentos comerciais selecionados aleatoriamente; e, por fim, o terceiro conjunto, considerado difícil por conter descrições de itens de NCMs do Capítulo 90, pertencentes a estabelecimentos comerciais amostrados aleatoriamente.

Os conjuntos de dados foram pré-processados com o objetivo de extrair os vetores de características utilizados como entrada para o algoritmo de treinamento do classificador Naïve Bayes. Esse pré-processamento envolveu tokenização, remoção de *stop words*, *stemming* e cálculo dos pesos dos vetores de características utilizando a métrica TF-IDF. Em seguida, foi realizado o treinamento do classificador, teste e avaliação do desempenho do mesmo utilizando a metodologia de validação cruzada.

Os resultados foram analisados e evidenciaram as características de dificuldade dos conjuntos de dados selecionados como experimento. Quando os resultados dos experimentos foram comparados com o caso utilizado como *baseline*, discutido no trabalho de Ding *et al.* (2015), observou-se que para os itens do Capítulo 22, o classificador desenvolvido nesse estudo mostrou desempenho inferior ao apresentado pelo *baseline*. No entanto, ao se considerar o conjunto de dados difícil, composto por transações de 819 classes (códigos de NCMs distintos), observaram-se resultados equivalentes. Do ponto de vista prático, considera-se que os resultados obtidos são significativos, uma vez que o sistema NCM, por possuir uma hierarquia mais profunda e, portanto, específica, é mais difícil de classificar do que o sistema harmonizado (SH), que possui uma hierarquia mais rasa. Outro fator a ser considerado, e que poderia contribuir para essa diferença de desempenho, é o fato desse estudo utilizar descrições de produtos baseadas na língua portuguesa, enquanto os experimentos do *baseline* foram baseados na língua inglesa, que possui construções mais simples e regulares sendo, portanto, mais fácil de classificar.

Em relação aos objetivos iniciais que motivaram o desenvolvimento desse trabalho, acredita-se que tenham sido satisfatoriamente atingidos, uma vez que se conseguiu estudar os conceitos e métodos relacionados à aprendizagem de máquina e processamento de texto, aplicando-os com sucesso no desenvolvimento de um classificador de NCM. Além disso, o classificador foi validado e comparado com outro trabalho da literatura, reforçando os resultados aqui descritos. Dessa forma, o

classificador especificado ao longo desse texto pode ser utilizado na classificação de descrições de itens de produtos em códigos NCM, além de contribuir para a identificação de divergências existentes em notas fiscais do consumidor emitidas pelos estabelecimentos comerciais, auxiliando assim na atividade de auditoria eletrônica. Assim, baseando-se nos resultados satisfatórios obtidos pela aplicação das técnicas abordadas ao longo desse estudo na classificação de documentos eletrônicos, espera-se que esta pesquisa mobilize a criação do protótipo funcional de um classificador que possa, futuramente, ser utilizado para contribuir na atividade de auditoria eletrônica ligada à SEFAZ-RS.

5.1. Trabalhos Futuros

Apesar dos modelos desenvolvidos demonstrarem desempenho satisfatório sobre Capítulos do NCM, relativamente difíceis de classificar, o desenvolvimento da solução completa depende do treinamento de classificadores para os demais tipos de documentos. Para que isso seja possível, também há a necessidade de dispor de conjuntos de dados rotulados ainda maiores, de modo que possam ser utilizados para o treinamento dos demais modelos.

Algoritmos baseados em aprendizagem de máquina normalmente utilizam conjuntos de dados selecionados aleatoriamente para o treinamento dos modelos. Em muitos desses casos, existe a possibilidade de utilizar aprendizado ativo, que permite ao algoritmo sinalizar ao especialista as instâncias que, se rotuladas, aumentam o desempenho do classificador. Esse tipo de técnica permite que após o treinamento inicial do modelo, esse seja aperfeiçoado ao longo do seu uso, através do fornecimento de instâncias rotuladas que garantam o aumento de sua acurácia. Mais detalhes sobre a utilização de aprendizado ativo em classificação de texto estão disponíveis no trabalho de Tong e Koller (2001).

Além disso, apesar da construção de um classificador de NCM auxiliar na tarefa de categorizar produtos com base em suas descrições, trata-se de apenas uma atividade inserida em um contexto maior, que é a aplicação desse classificador em larga escala sobre bancos de dados massivos. Assim, outra sugestão de trabalho futuro é o desenvolvimento de tal arquitetura, que além de permitir a classificação de grandes quantidades de documentos em lote, também auxiliaria na atividade de auditoria eletrônica, através da identificação de divergências entre os códigos NCM informados pelos estabelecimentos comerciais e os códigos inferidos pelo classificador treinado.

6. Referências

Bird, S., Klein, E., Loper, E. (2009) “Natural language processing with Python: analyzing text with the natural language toolkit”, O'Reilly Media, Inc.

Ding, L., Fan, Z., Chen, D. (2015) “Auto-Categorization of HS Code Using Background Net Approach”, Procedia Computer Science, v. 60, p. 1462-1471.

Flick, U. (2012) “Introdução à metodologia de pesquisa: um guia para iniciantes”, Penso Editora.

Indurkhya, N., Damerau, F. J. (2010) “Handbook of natural language processing”, CRC Press.

Ko, Y; Seo, J. (2000) “Automatic text categorization by unsupervised learning” Proceedings of the 18th conference on Computational linguistics - Volume 1. Association for Computational Linguistics. p. 453-459.

Kohavi, R. (1995) “A study of cross-validation and bootstrap for accuracy estimation and model selection”, International joint Conference on artificial intelligence. (S.l.: s.n.). v. 14, p. 1137–1145.

Korde, V. and Mahender, C. N. (2012) “Text classification and classifiers: A survey”, International Journal of Artificial Intelligence & Applications, v. 3, n. 2, p. 85.

Leskovec, J., Rajaraman, A. and Ullman, J. D. (2014) “Mining of massive datasets”, Cambridge University Press.

Luhn, H. P. (1957) “A statistical approach to mechanized encoding and searching of literary information”, IBM Journal of research and development, v. 1, n. 4, p. 309-317.

McCallum, A. (1999) “Multi-label text classification with a mixture model trained by EM”, AAAI workshop on Text Learning. p. 1-7.

Manning, C. D., Raghavan, P. and Schütze, H. (2008) “Introduction to information retrieval”, v. 1, n. 1. Cambridge: Cambridge University Press.

Ministério da Indústria, Comércio Exterior e Serviços. (2016) “TEC em Excel Completa”, http://www.mdic.gov.br/comercio-exterior/estatisticas-de-comercio-exterior-9/arquivos-atuais, Abril.

Mitchell, T. M. (1997) “Machine learning”, Burr Ridge, IL: McGraw Hill, v. 45, p. 37.

Orengo, V. M. and Huyck, C. R. (2001) “A Stemming Algorithm for the Portuguese Language”, Spire. p. 186-193.

Orengo, V. M. and Huyck, C. R. (2001) “RSLP Stemmer (Removedor de Sufixos da Língua Portuguesa)”, http://www.inf.ufrgs.br/~viviane/rslp/, Abril.

Pedregosa, F., Buitinck, L., Louppe, G., Blondel, M., Mueller, A., Grisel, O., Niculae, V., Prettenhofer, P., Gramfort, A., Grobler, J., Layton, R., Vanderplas, J., Joly, A., Holt, B. and Varoquaux, G. (2011) “Scikit-learn: Machine Learning in Python”, JMLR 12, pp. 2825-2830.

Receita Federal. (2015) “Sistema harmonizado de designação e de codificação de mercadorias”, http://idg.receita.fazenda.gov.br/acesso-rapido/legislacao/legislacao-por-assunto/sistema-harmonizado, Abril.

Russell, S. J. and Norvig, P. (2003) “Artificial intelligence: a modern approach”, Upper Saddle River: Prentice hall.

Sparck Jones, K. (1972) “A statistical interpretation of term specificity and its application in retrieval”, Journal of documentation, v. 28, n. 1, p. 11-21.

Tong, S., Koller, D. (2001) “Support vector machine active learning with applications to text classification”, Journal of machine learning research, v. 2, n. Nov, p. 45-66.

Triola, M. F. (2008) “Bayes’ Theorem”, http://faculty.washington.edu/tamre/BayesTheorem.pdf, Abril.