Avaliação de Técnicas de Aprendizado de Máquina Indutivo Supervisionado para Classificação Automática de Textos em Fluxo de Notícias

Renan Prates Alves[1], **Rafael Geraldeli Rossi**[1]

[1]Sistemas de Informação – Universidade Federal de Mato Grosso do Sul (UFMS)
Caixa Postal 210 – 79603-011 – Três Lagoas – MS – Brazil

renanidc@gmail.com, rafael.g.rossi@ufms.br

1. Resumo

Nos dias atuais há uma grande quantidade de dados textuais, como e-mails, relatórios, páginas web, e postagens em redes sociais, sendo produzidos diariamente. Parte desses dados textuais está na forma de notícias, as quais contém informações interessantes, relevantes e são acessadas por grande parte da população. Processar, organizar, gerenciar e extrair conhecimento dessa grande quantidade de notícias manualmente exige um grande esforço humano, sendo muitas vezes impossível de ser realizado manualmente. Com isso, técnicas computacionais que requerem pouca intervenção humana e que permitem a organização, gerenciamento e extração de conhecimento têm ganhado destaque nos últimos anos. Dentre as técnicas, destaca-se a classificação automática de textos, cujo objetivo é atribuir rótulos (identificadores de categorias pré-definidos) à documentos textuais ou porções de texto. Uma forma viável de realizar a classificação automática de textos é por meio de algoritmos de aprendizado de máquina, que são capazes de "aprender", generalizar, ou ainda extrair padrões das classes das coleções com base no conteúdo e rótulos de documentos textuais. Atualmente, uma grande quantidade de notícias é gerada em tempo real, e a classificação automática desse fluxo de notícias é um desafio real, visto que um fluxo contínuo de notícias é um ambiente em constante transformação, com novos temas e conceitos surgindo a todo momento. Dessa forma, fica evidente a importância da avaliação de técnicas de aprendizado de máquina para classificação automática em fluxo de notícias, a fim de avaliar a performance de classificação ao longo do tempo, e detectar se a performance de classificação se degrada ao decorrer do tempo. Assim, este trabalho tem como objetivo avaliar as técnicas de aprendizado de máquina indutivo supervisionado para classificação automática automática de textos em fluxo de notícias, e servir como base para pesquisas futuras, ao fornecer bases de bases de *benchmarking* e resultados *baseline* para a comunidade científica.

2. Introdução

A quantidade de dados disponível em formato digital na rede mundial de computadores tem aumentado incessantemente. De acordo com estimativas realizadas em 2014, de 2013 a 2020 o universo digital irá aumentar de 4,4 trilhões de gigabytes para 44 trilhões de gigabytes [Turner et al. 2014]. Parte dos dados no universo digital está no formato textual[1] , como e-mails, relatórios, boletins, artigos, registros de pacientes e conteúdo de páginas web.

[1]De acordo com [Ur-Rahman and Harding 2012] e [Kuechler 2007], aproximadamente 80% das informações corporativas são compostas por dados textuais.

Dentre essa grande quantidade de dados textuais encontram-se as notícias, as quais são importantes para manter a população informada acerca dos mais diversos assuntos. As notícias apresentam informações sobre um acontecimento novo, ou que por vezes divulga uma novidade sobre uma situação já existente, com o intuito de publicar informações atuais e relevantes. Porém, com o fluxo cada vez maior de geração de notícias, é necessário a definição de métodos computacionais para o seu gerenciamento, visto que é humanamente impossível realizar o gerenciamento manual do crescente fluxo de notícias gerado diariamente. Além disso, fica evidente a importância da organização e gerenciamento de notícias, tanto para os veículos de comunicação, para manter sua coleção de notícias organizadas, quanto para os usuários dos portais, os quais podem acessar rapidamente notícias de categorias de seu interesse. A organização e gerenciamento de notícias também é de grande utilidade para monitorar a incidência de determinados temas de notícias ao longo do tempo [Marcacini et al. 2013], para a recomendação de notícias aos usuários [Weiss et al. 2010, Billsus and Pazzani 1999], e para a extração de conhecimento de dados textuais presentes na web [Aggarwal and Zhai 2012]. Para viabilizar tal atividade computacionalmente, são necessárias técnicas para classificar automaticamente os textos em categorias, o que possibilita a realização automática das tarefas mencionadas acima [Rossi 2016].

Tradicionalmente, aplicam-se algoritmos de aprendizado indutivo supervisionado para induzir um modelo de classificação, o qual é utilizado para classificar novos textos automaticamente [Aggarwal et al. 2014, Rossi 2016]. No entanto, em um fluxo de notícias, novos conceitos e temas podem surgir ao longo do tempo, o que pode causar a degradação de performance do modelo de classificação. Dessa forma, o objetivo deste trabalho é avaliar técnicas de aprendizado de máquinas supervisionado para a classificação automática em fluxo de notícias e analisar se, de fato, a degradação da performance de classificação ocorre ao longo do tempo. Para viabilizar essa proposta foram realizadas as seguintes etapas:

(i) Desenvolvimento de *crawlers* para coletar notícias, com seus respectivos textos e categorias (rótulos), de diferentes portais de notícias.
(ii) Coleta de bases de notícias presentes na literatura, e adaptação dessas bases para um formato adequado para serem utilizadas nas ferramentas consideradas neste trabalho.
(iii) Aplicação de algoritmos de aprendizado de máquina para induzir modelos de classificação e avaliar a performance de classificação ao longo do tempo.

Para o processo de análise da performance de classificação ao longo do tempo, foram realizadas as seguintes etapas:

(i) As técnicas foram aplicadas em diferentes coleções de notícias com diferentes problemas de classificação.
(ii) Também foram avaliadas as gerações de diferentes modelos de classificação, considerando diferentes algoritmos indutivos supervisionados, diferentes quantidades de exemplos rotulados e diferentes períodos de tempo para análise do custo-benefício entre tempo gasto com rotulação e performance de classificação.

Além dos objetivos já citados, este trabalho tem o intuito de servir como base para outros projetos que venham a ser desenvolvidos sobre a temática de

classificação automática de textos em fluxo de notícias, por meio da coleta, estruturação e disponibilização de bases de notícias rotuladas e que possuam informação temporal, bem como resultados comparativos.

O restante deste artigo está dividido da seguinte forma: Na seção 3 são apresentados conceitos sobre a classificação em fluxo de dados, com uma visão geral sobre o estado-da-arte dos trabalhos relacionados a classificação automática de textos em fluxo de notícias. Na seção 4 é apresentado o fluxo de desenvolvimento deste trabalho, onde será exibido as técnicas utilizadas para coleta de notícias, os métodos de aprendizado de máquina supervisionado utilizados, bem como a avaliação experimental contendo os resultados gerados neste trabalho. Na seção 5 são apresentadas as considerações finais e trabalhos futuros.

3. Conceitos Sobre Classificação em Fluxo de Dados

Em geral, as seguintes abordagens podem ser consideradas para lidar com a classificação automática em fluxo de dados [Krawczyk and Woźniak 2015]:

(i) Treinar novos classificadores cada vez que novos dados estão disponíveis: tal abordagem é pouco prática e custosa do ponto de vista computacional, especialmente se a perca de performance ocorre muito rápido.
(ii) Detectar a mudança em novos dados: se essas mudanças são significativas o suficiente, então treinar o classificador com base nos novos dados coletados.
(iii) A adoção de um algoritmo incremental para o modelo de classificação, que possua uma fácil adaptação à mudança de natureza dos objetos de entrada.

Os primeiros algoritmos conhecidos para lidar com a perda de performance de classificadores, considerando fluxo de dados, foram STAGGER [Schlimmer and Granger 1986], IB3 [Aha 1991], e o conjunto de algoritmos FLORA [Widmer and Kubat 1996]. Já nos dias atuais há uma grande quantidade de algoritmos para lidar com a classificação em fluxo de dados. Basicamente, pode-se organizá-los nos seguinte grupos [Krawczyk and Woźniak 2015]:

(i) Algoritmos de alteração de conceito.
(ii) Algoritmos de aprendizado online.
(iii) Algoritmos de janela deslizante.
(iv) Abordagem de conjuntos.

Os algoritmos de alteração de conceito têm o objetivo de alertar o sistema ao detectar mudanças no conjunto de dados [Sobolewski and Wozniak 2013]. Alguns sistemas evoluem continuamente para ajustar o modelo para os dados de entrada [Žliobaitė 2010], o que é chamado de detecção de desvio implícito [Kuncheva 2008] em oposição a métodos de detecção de desvio explícitos que enviam um sinal para indicar mudanças no conjunto de dados. A detecção de alteração de conceito pode ser baseada em mudanças na distribuição probabilística das instâncias [Gaber and Yu 2006, Markou and Singh 2003] ou na performance de classificação [Klinkenberg and Joachims 2000, Baena-Garcıa et al. 2006]. Muitos algoritmos de detecção de alteração de conceito são baseados no conhecimento dos rótulos dos textos após a classificação para detectar a presença de um desvio (alteração de conceito). No entanto, como salientado por [Žliobaitė 2010], tal abordagem não é útil a partir de

um ponto de vista prático, pois em situações reais, os documentos advindos do fluxo não possuem rótulos.

Já o aprendizado online refere-se a algoritmos de classificação que continuamente atualizam seus modelos de classificação durante o processamento de dados recebidos. De acordo com [Domingos and Hulten 2003], esses métodos devem atender a alguns requisitos básicos:

- Cada objeto deve ser processado apenas uma vez no decorrer do treinamento;
- A memória e tempo de computação são limitadas;
- O treinamento do classificador pode ser interrompido várias vezes e a sua qualidade não deve ser menor do que a dos classificadores treinados em lote *(batch mode)*.

Os algoritmos de aprendizado de máquina popularmente aplicados no aprendizado online são: Naïve Bayes, Redes Neurais, *k*-vizinhos mais próximos, e Concept-Adaptin Very Fast Decision Tree (CVFDT) [Hulten et al. 2001].

O terceiro grupo de algoritmos utiliza o conceito de janelas deslizantes que incorporam a técnica de mecanismo de esquecimento *(forgetting mechanism)*. Esta abordagem é baseada no pressuposto de que os dados mais recentes têm maior relevância, porque eles contém características do cenário atual. Normalmente, os algoritmos de janela deslizante, utilizam três estratégias:

(i) Seleção dos textos por meio de uma janela deslizante que corta instâncias antigas [Widmer and Kubat 1996];
(ii) Ponderação dos dados de acordo com o tempo, onde documentos mais recentes recebem um peso maior;
(iii) Aplicação de *bagging and boosting* de algoritmos que classificam instâncias erroneamente no período de tempo mais recente. [Bifet et al. 2009, Chu and Zaniolo 2004].

Ao lidar com a janela deslizante, a questão principal é como ajustar o tamanho da janela. Se por um lado, uma menor janela permite focar no contexto atual, seus dados podem não ser representativos para um contexto mais duradouro. Por outro lado, uma janela maior pode resultar na mistura das instâncias que representam diferentes contextos. Por conseguinte, certos algoritmos avançados ajustam o tamanho da janela dinamicamente, conforme a entrada de novos documentos no conjunto, como é o caso dos algoritmos FLORA2 [Widmer and Kubat 1996] e ADWIN2 [Bifet and Gavalda 2007], ou pode ser utilizado o conceito de múltiplas janelas [Lazarescu et al. 2004], onde são utilizadas diferentes janelas de diferentes tamanhos.

Já o último grupo de algoritmos para lidar com a classificação em fluxo de dados adota a abordagem de conjuntos, e é composto de algoritmos que incorporam um comitê de classificadores [Wang et al. 2003, Stanley 2003, Tsymbal et al. 2008]. Uma decisão coletiva pode aumentar a performance de classificação porque o conhecimento que é distribuído entre os classificadores pode ser mais abrangente, visto que quando um classificador erra a classificação de um determinado documento do fluxo, os outros classificadores do comitê de classificadores podem acertar a classificação. Esta premissa é verdadeira se o comitê de classificadores é composto por classificadores com diferentes bias de aprendizado [Domingos and Hulten 2003, Shipp and Kuncheva 2002]. Pode-se distinguir três

principais abordagens relacionadas ao conceito de comitê de classificadores automáticos de texto:

- Combinadores dinâmicos, onde os classificadores individuais são treinados com antecedência e sua relevância para o contexto atual é avaliada de forma dinâmica durante o fluxo de dados [Jacobs et al. 1991, Haussler et al. 1994].
- Atualização de membros do comitê, onde cada comitê consiste em um conjunto de classificadores online que são atualizados incrementalmente com base no fluxo de entrada [Fern and Givan 2003, Kolter and Maloof 2007, Bifet et al. 2011, Rodríguez and Kuncheva 2008].
- Mudanças dinâmicas do arranjo do comitê de classificadores, ou seja, classificadores individuais são avaliados de forma dinâmica e o pior é substituído por um novo classificador treinado no conjunto de dados mais recente [Jackowski 2014, Kolter and Maloof 2003].

O *Streaming Ensemble Algorithm (SEA)* [Street and Kim 2001] ou o *Accuracy Weighted Ensemble (AWE)* [Wang et al. 2003] mantém um comitê de classificadores de tamanho fixo. Os dados são coletados em blocos de dados, que são usados para treinar novos classificadores. A SEA utiliza uma votação por maioria, enquanto que o AWE faz a decisão com base na votação ponderada, baseada na acurácia do conjunto de treinamento. O algoritmo *Dynamic Weighted Majority (DWM)* [Kolter and Maloof 2003] reduz o peso quando o classificador faz uma decisão incorreta. Eventualmente, o classificador é removido do conjunto quando seu peso cai abaixo de um determinado limiar. Independentemente, um novo classificador é adicionado ao conjunto quando o conjunto atual faz uma decisão errada.

Em [Jackowski 2014] é proposto um método de treinamento para conjuntos de classificadores chamado de *Recurring Context*. Esse método de treinamento consiste na seleção de um conjunto de classificadores para o modelo atual com base em programação evolucionária. [Sobolewski and Wozniak 2013] propõe o modelo de conjunto dinâmico chamado *Weighted Aging Ensemble (WAE)*. Ele pode modificar o arranjo de um comitê de classificadores com base em três fatores: medida de diversidade, precisão do conjunto geral, e o tempo que um classificador passou como membro de um conjunto.

4. Projeto: Avaliação de Técnicas de Aprendizado de Máquina Indutivo Supervisionado para a Classificação de Textos em Fluxo de Notícias

Nessa seção será apresentado o fluxo de desenvolvimento deste trabalho, onde será demonstrado quais bases de notícias foram consideradas, bem como as técnicas para coleta de notícias da *Web* e o processamento realizado em tais bases. Também serão apresentados os modelos de representação de textos, os métodos de aprendizado de máquina supervisionado utilizados neste trabalho, e os resultados obtidos com as avaliações experimentais.

4.1. Coleta

Para a execução deste trabalho foram utilizadas as seguintes bases de textos:

(i) **Reuters 21578:** base de textos[2] contendo notícias da Reuters [Lewis 1997]. Essa é uma base de *benchmark* encontrada na literatura.

[2]http://www.daviddlewis.com/resources/testcollections/reuters21578/

(ii) **Eventos do milho:** base de textos contendo informações sobre eventos referentes ao mercado ao milho (base cedida pelo Prof. Dr. Ricardo Marcondes Marcacini, referente ao projeto Websensor [Marcacini et al. 2013]).
(iii) **Notícias do Brasil:** base de textos extraídos de sites brasileiros de notícias (Essa base foi construída durante o projeto).

Todas as bases de texto foram organizadas da seguinte forma, para assim facilitar o posterior processamento e serem utilizadas na ferramenta de pré-processamento, aplicação e avaliação de algoritmos de aprendizado de máquina indutivo supervisionado utilizada neste trabalho[3]:

(i) Uma notícia por arquivo no formato de texto plano (*.txt*).
(ii) O nome de cada arquivo texto contendo a data de publicação da notícia, e um identificador, normalmente composto pelo título e a fonte de publicação.
(iii) Todas as notícias separadas por diretórios, sendo que cada diretório possui o nome correspondente a classe da notícia.

A seguir, são apresentados os procedimentos realizados para obtenção e adequação ao formato apresentado anteriormente, para cada uma das bases.

4.1.1. Eventos do Milho

Nessa base as notícias se encontravam em um aquivos no formato CSV[4], contendo a categoria, a data de publicação, e um identificador com o endereço da notícia. Para adequar a base ao formato utilizado neste trabalho foi necessário desenvolver um *script* em Python para realizar a leitura e a extração das notícias do arquivo CSV para o padrão utilizado neste trabalho. O script desenvolvido, juntamente com as demais bases coletadas está no endereço ao final dessa seção.

4.1.2. Notícias do Brasil

Para a construção dessa base de notícias foram utilizadas algumas tecnologias, para auxilar a construção dos *web crawlers*[5] desenvolvidos para este trabalho. A primeira delas é a biblioteca `Requests`[6], que é uma biblioteca *HTTP (Hypertext Transfer Protocol)* licenciada sob Apache2[7] e escrita em Python. A biblioteca possui módulos que permitem realizar requisições *HTTP* para extração do conteúdo *HTML (Hyper Text Markup Language)* de sites na web. Assim, com o auxílio da biblioteca *Requests* foi possível iniciar o desenvolvimento de *web crawlers*. O processo que um *web crawler* executa é chamado de *web crawling* ou *spidering* [Castillo 2005]. Os *web crawlers* são principalmente utilizados para criar uma cópia de todas as páginas visitadas. Em geral, o *crawler* começa com uma lista de *URLs (Uniform Resource Locator)* para visitar. À medida que o *crawler*

[3]Text Categorization Tool - `http://sites.labic.icmc.usp.br/ragero/thesis/text_categorization_tool/`
[4]`https://pt.wikipedia.org/wiki/Comma-separated_values`
[5]programa de computador que navega pela *WEB* de forma metódica e automatizada
[6]`http://docs.python-requests.org/pt\_BR/latest/`
[7]`http://www.apache.org/licenses/`

visita essas *URLs*, ele identifica todos os *links* na página e os adiciona na lista de *URLs* para visitar posteriormente. Tais *URLs* são visitadas recursivamente de acordo com um conjunto de regras. Muitos sites, em particular os motores de busca, usam *crawlers* para manter uma base de dados atualizada. *Crawlers* também podem ser usados para tarefas de manutenção automatizadas em um *website*, como checar suas *URLs* ou validar um código *HTML*.

Após a construção dos *crawlers* e coleta dos arquivos *HTML*, foi necessário realizar a limpeza dos dados coletados pelo *crawler*, para extração do texto puro, sem *tags* de marcação. Para a realização da limpeza dos documentos foi utilizada a biblioteca `BeautifulSoup`[8], que é uma biblioteca em Python que permite realizar a análise sintática (também conhecida pelo termo em inglês *parsing*) de documentos *HTML*. A análise sintática transforma um texto de entrada em uma estrutura de dados, em geral uma árvore, que possui uma hierarquia implícita e que seja conveniente para processamento posterior. Assim, com o auxílio da biblioteca `BeautifulSoup`, foi possível possível criar uma ferramenta para extração do conteúdo textual relevante para este trabalho, isto é, a data de publicação da notícia, o título, a fonte de publicação e o corpo da notícia.

A biblioteca `BeautifulSoup` possui alguns métodos e expressões para navegar e modificar uma árvore de análise de documentos *HTML*. Um exemplo do funcionamento destes métodos e expressões fornecidos pela biblioteca é demonstrado a seguir.

- Para representar um documento *HTML* foi atribuído um trecho *HTML* a uma variável, com pode ser visto na Figura 1.

```
html_doc = """
<html><head><title>The Dormouse's story</title></head>
<body>
<p class="title"><b>The Dormouse's story</b></p>

<p class="story">Once upon a time there were three little sisters;
and their names were <a href="http://example.com/elsie">Elsie</a>,
<a href="http://example.com/lacie">Lacie</a> and
<a href="http://example.com/tillie">Tillie</a>;
and they lived at the bottom of a well.</p>

<p class="story">...</p>"""
```

Figura 1. Variável com conteúdo *HTML* [Richardson 2015].

- A criação de um objeto `BeautifulSoup` que representa a estrutura aninhada do documento é exibido na Figura 2.

```
from bs4 import BeautifulSoup
soup = BeautifulSoup(html_doc, 'html.parser')
```

Figura 2. Criação de objeto [Richardson 2015].

- Algumas maneiras de como navegar no objeto `BeautifulSoup` são exibidas na Figura 3, no exemplo é demonstrado como capturar todos os elementos *HTML* que possuem a classe title.

[8]https://pypi.python.org/pypi/beautifulsoup4

```
soup.title
# <title>The Dormouse's story</title>

soup.title.name
# u'title'

soup.title.string
# u'The Dormouse's story'
```

Figura 3. Navegar no objeto `BeautifulSoup` [Richardson 2015].

- Como realizar a captura de todos os links do documento é exibido na Figura 4.

```
for link in soup.find_all('a'):
  print(link.get('href'))
# http://example.com/elsie
# http://example.com/lacie
# http://example.com/tillie
```

Figura 4. Captura de *links* [Richardson 2015].

- Como realizar a extração dos textos do documento é exibido na Figura 5

```
print(soup.get_text())
# The Dormouse's story
#
# The Dormouse's story
#
# Once upon a time there were three little sisters;
# and their names were
# Elsie,
# Lacie and
# Tillie;
# and they lived at the bottom of a well.
#
# ...
```

Figura 5. Extração de textos [Richardson 2015].

Para a construção da base **Notícias do Brasil** as fontes selecionadas para a extração de notícias foram:

- G1: `http://g1.globo.com/`
- Globo Esporte: `http://globoesporte.globo.com/`
- Band: `http://noticias.band.uol.com.br/economia/`
- Correio 24 horas: `http://www.correio24horas.com.br/`
- IG - Último Segundo: `http://ultimosegundo.ig.com.br/`
- IG – Tecnologia: `http://tecnologia.ig.com.br/`

Essa fontes foram escolhidas por possuírem uma grande quantidade de notícias publicadas de diversas categorias e pela facilidade de coletar notícias dessas fontes, visto que, durante o período de coleta, todas essas fontes possuíam uma página inicial com os endereços de todas as notícias publicadas e todas as notícias possuíam a categoria na própria *URL*. Os *scripts* desenvolvidos, para coleta de notícias e limpeza dos textos encontram-se no endereço: `https://github.com/renanidc/WebCrawlers`. As bases já pré-processadas, isto é, com todas as notícias separadas por data e categoria, encontram-se no endereço: `https://goo.gl/iqhrTv`.

4.2. Representação Estruturada para a Classificação de Textos

A representação estruturada dos textos é a base para o processamento de textos e consequentemente para a aplicação de algoritmos de aprendizado de máquina. A qualidade dos resultados dos algoritmos de aprendizado de máquina é diretamente proporcional a qualidade da representação da coleção de textos. Tipicamente o modelo espaço-vetorial tem sido utilizado para representar coleções de textos. No entanto há também a representação em redes, que tem se destacado nos últimos anos. Nas próximas seções, são descritos esses dois modelos para representação estruturada de uma coleção de textos.

4.2.1. Modelo Espaço-Vetorial

Representações baseadas no modelo espaço-vetorial são as mais comuns na área de aprendizado de máquina e na classificação automática de textos [Aggarwal et al. 2014, Shalev-Shwartz and Ben-David 2014]. Portanto, uma grande gama de algoritmos foi desenvolvida considerando este tipo de representação. Neste modelo, cada exemplo ou instância do conjunto de dados é constituído por um vetor, sendo que cada dimensão do vetor corresponde à uma característica ou um atributo do conjunto de dados. Normalmente, os atributos são os termos presentes na coleção de textos. Comumente usa-se a palavra "termo" para denotar as dimensões geradas com base nas palavras de um texto, como palavras simples *(bag-of-words)*, sequências ou conjuntos de palavras [Rossi and Rezende 2011].

A união de todos os vetores que representam os documentos forma uma matriz denominada matriz **documento-termo**. Nessa matriz, as linhas representam os documentos e as colunas representam os termos, sendo que na última coluna é representada, caso haja essa informação, a classe do documento. A coluna correspondente a classe contém valores nominais, que são denominados rótulos. Dessa forma, os documentos que possuem rótulos são denominados documentos rotulados, enquanto os que não possuem são denominados documentos não rotulados. Na Figura 6 é ilustrada a representação de uma coleção de documentos rotulados em uma matriz documento-termo.

	$\mathbf{t_1}$	$\mathbf{t_2}$	$\mathbf{t_3}$	$\dots$	$\mathbf{t_{M-2}}$	$\mathbf{t_{M-2}}$	$\mathbf{t_M}$	**Classe**
$\mathbf{d_1}$	w_{d_1,t_1}	w_{d_1,t_2}		$\dots$	$w_{d_1,t_{M-2}}$	$w_{d_1,t_{M-1}}$	w_{d_1,t_M}	c_{d_1}
$\mathbf{d_2}$	w_{d_2,t_1}	w_{d_2,t_2}		$\dots$	$w_{d_2,t_{M-2}}$	$w_{d_2,t_{M-1}}$	w_{d_2,t_M}	c_{d_2}
$\mathbf{d_3}$	w_{d_3,t_1}	w_{d_3,t_2}		$\dots$	$w_{d_3,t_{M-2}}$	$w_{d_3,t_{M-1}}$	w_{d_3,t_M}	c_{d_3}
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\ddots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$
$\mathbf{d_{N-2}}$	w_{d_{N-2},t_1}	w_{d_{N-2},t_2}		$\dots$	$w_{d_{N-2},t_{M-2}}$	$w_{d_{N-2},t_{M-1}}$	w_{d_{N-2},t_M}	$c_{d_{N-2}}$
$\mathbf{d_{N-1}}$	w_{d_{N-1},t_1}	w_{d_{N-1},t_2}		$\dots$	$w_{d_{N-1},t_{M-2}}$	$w_{d_{N-1},t_{M-1}}$	w_{d_{N-1},t_M}	$c_{d_{N-1}}$
$\mathbf{d_N}$	w_{d_N,t_1}	w_{d_N,t_2}		$\dots$	$w_{d_N,t_{M-2}}$	$w_{d_N,t_{M-1}}$	w_{d_N,t_M}	c_{d_N}

Figura 6. Ilustração de uma matriz documento-termo representando uma coleção com N documentos e M termos [Rossi 2016].

Os valores das células da matriz **documento-termo**, exceto os da coluna **Classe**, são numéricos. Esses valores representam o peso de um termo ou atributo em um documento. Os pesos dos termos para os documentos são medidas quantitativas baseadas na

frequência de um termo. Em geral, essas medidas são obtidas de maneira não supervisionada, isto é, não é considerado o rótulo dos documentos para gerar os pesos dos termos. Métodos não supervisionados mais comuns para definir os pesos do termos dos documentos são [Manning et al. 2008, Feldman and Sanger 2007]: (i) binário, no qual w_{d_i,t_j} é igual a 1 se t_j ocorre em d_i e 0 caso contrário; (ii) frequência do termo (do inglês *term frequency - tf*), no qual w_{d_i,t_j} corresponde à frequência (número de repetições) de t_j em d_i; e (iii) frequência do termo ponderada pelo inverso da frequência de documento (do inglês *term frequency - inverse document frequency - tf-idf*), que pondera a frequência do termo pelo inverso do número de documentos da coleção em que o termo ocorre.

4.2.2. Representação em Redes

Muitos problemas do mundo real podem ser modelados utilizando redes. Algumas definições para redes encontradas na literatura são:

- *"Uma rede, em sua forma mais simples, é uma coleção de pontos, nos quais pares de pontos são conectados por uma linha"* [Newman 2010].
- *"Uma rede é uma representação simplificada que reduz um sistema à uma representação abstrata"* [Newman 2010].

Independente do tipo de rede, todas elas podem ser formalmente definidas como uma tripla $\mathcal{N} = \langle \mathcal{O}, \mathcal{R}, \mathcal{W} \rangle$, na qual $\mathcal{O}$ representa o conjunto de objetos da rede, $\mathcal{R}$ representa o conjunto das relações entre os objetos e $\mathcal{W}$ representa o conjunto de pesos das relações entre os objetos. Basicamente, as redes se organizam em dois tipos: redes homogêneas, na qual os objetos se conectam a objetos do mesmo tipo, e redes heterogêneas, onde os objetos se conectam a objetos de tipos diferentes.

Dentre as inúmeras possibilidades para modelagem de textos em redes, observam-se dois tipos predominantes de redes para a tarefa de classificação: redes de documentos e redes de termos. Em uma rede documentos, os objetos correspondem aos documentos da coleção e as relações podem ser explícitas ou implícitas. Informações explícitas referem-se a relações naturais ou relações informadas explicitamente no conjunto de dados. Relações entre autores e artigos, entre documentos e termos, entre páginas *web* dada por *hyperlinks*, entre artigos científicos ou patentes dada por citações, são exemplo de relações geradas por meio de informações explícitas extraídas dos próprios conjuntos de dados [Rossi 2016]. Entretanto, alguns tipos de relações podem ser gerados por meio de informações implícitas, ou relações mineradas do conjunto de dados.

Normalmente, as relações implícitas são extraídas por meio do cálculo da similaridade entre os objetos de uma rede. Para tanto, cada objeto da rede deve conter um vetor de atributos ou alguma estrutura computacional que permita o cálculo da similaridade entre os objetos. No caso de textos, por exemplo, pode-se considerar cada documento como um objeto de rede e calcular a similaridade entre os objetos da rede considerando o vetor de termos dos documentos [Rossi 2016].

As primeiras pesquisas envolvendo a modelagem de textos utilizando redes de documentos consideraram relações explícitas para gerar a rede, como hiperlinks e citações [Lu and Getoor 2003, Oh et al. 2000, Chakrabarti et al. 1998]. Porém, pesquisas posteriores demonstraram que considerar relações implícitas como a similaridade ao invés

das relações explícitas para gerar uma rede de documentos provê melhores resultados [Angelova and Weikum 2006]. Além disso, com o advento da área de aprendizado semi-supervisionado, redes de documentos baseadas em similaridade têm sido mais utilizadas [Subramanya and Bilmes 2008, Belkin et al. 2006, Zhou et al. 2003, Zhu et al. 2003].

Uma outra forma de representar coleções de textos para a tarefa de classificação é considerar apenas relações entre documentos e termos, a esse tipo de representação dâ-se o nome de rede bipartida. Esse tipo de rede vem obtendo resultados promissores para a classificação automática de textos [Rossi 2016]. No caso de uma rede bipartida para representar coleções de textos, um documento d_i é conectado a um termo t_j se t_j ocorre em d_i. O peso da relação entre um documento d_i e um termo t_j, denotado por w_{d_i,t_j}, corresponde à frequência de t_j em d_i. Esse tipo de rede é rapidamente gerada, na Figura 7 é apresentada uma ilustração de uma rede bipartida para representar coleções de textos.

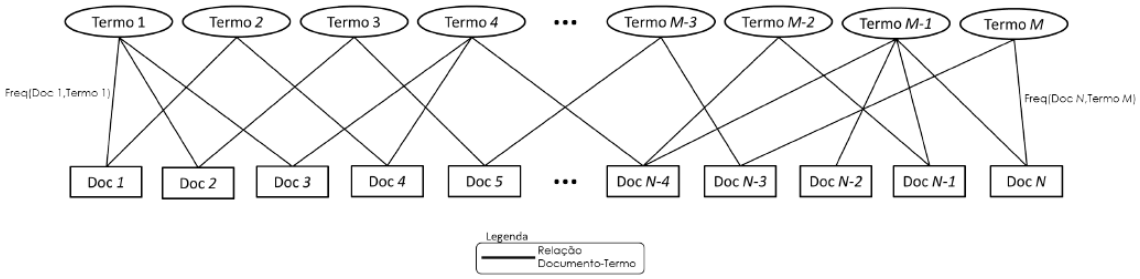

Figura 7. Ilustração de uma rede bipartida para representar coleções de textos [Rossi 2016].

4.3. Algoritmos de Aprendizado de Máquina

O aprendizado de máquina é um sub-campo da ciência da computação que evoluiu do estudo de reconhecimento de padrões e da teoria da aprendizagem computacional em inteligência artificial [Russell and Norvig 2002]. De acordo com [Samuel 1959] *"O aprendizado de máquina é campo de estudo que dá aos computadores a habilidade de aprender sem serem explicitamente programados"*. Na inteligência artificial existe o aprendizado indutivo supervisionado, na qual o algoritmo de aprendizado (indutor) recebe um conjunto de dados, e partir desse conjunto, extrai regras e padrões [Russell and Norvig 2002].

Na classificação automática de textos o processo de indução consiste em gerar um modelo de classificação, para assim, poder classificar novos textos. Neste trabalho, para a execução das análises de desempenho realizadas foram utilizados os algoritmos tradicionais de aprendizado indutivo supervisionado no modelo espaço-vetorial, sendo eles: MNB (*Multinomial Naive Bayes*), *k*-NN (*k-Nearest Neighbors*), SVM (***Support Vector Machine***) [Aggarwal et al. 2014], C4.5 e um algoritmo baseado em redes bipartidas chamado IMBHN (*Inductive Model Based on Bipartite Heterogeneous*). Mais detalhes sobre esses algoritmos são apresentados a seguir.

4.3.1. *Multinomial Naive Bayes (MNB)*

O algoritmo MNB [Aggarwal and Zhai 2012] é uma técnica de classificação probabilística baseada no teorema de Bayes, porém, possui uma suposição de independência

entre os atributos. Dessa forma, a probabilidade de ocorrer uma atributo para uma classe é independente das probabilidades dos demais atributos. O modelo MNB é fácil de construir e particularmente útil para grandes conjuntos de dados. Além disso, o algoritmo MNB é conhecido por superar a performance de classificação de métodos de classificação altamente sofisticados. Este algoritmo é amplamente utilizado em classificação de textos.

4.3.2. *k-Nearest Neighbors (kNN)*

O algoritmo *k*NN pertence a um grupo de técnicas denominada de aprendizado baseado em instâncias (*instance basead learning*). O algoritmo de classificação *k*NN [Tan et al. 2006] é uma técnica amplamente empregada para reconhecer padrões. O *k*NN classifica um dado elemento considerando as respectivas classes dos *k* ($k >= 1$) vizinhos mais próximos pertencentes a uma dada base de treinamento. O algoritmo calcula a distância do elemento dado, para cada elemento da base de treinamento, baseado na distância entre o elemento mais próximo e o menos próximo da base de treinamento. Dos elementos ordenados selecionam-se apenas os *k* primeiros que servem de parâmetro para a regra de classificação.

Dois pontos importantes no *k*NN são: a regra de classificação e a função que calcula a proximidade entre duas instâncias. A regra de classificação diz como o algoritmo vai tratar a importância de cada um dos *k* elementos selecionados: os elementos podem ser selecionados por votação majoritária ou votação ponderada pela proximidade [Tan 2006]. A função de proximidade é responsável por mensurar a disparidade entre dois elementos de forma a poder identificar quais são os *k* vizinhos mais próximos. Neste trabalho, foi utilizada a medida cosseno como medida de proximidade entre os documentos de uma coleção [Tan 2006].

4.3.3. *Support Vector Machine (SVM)*

O SVM tenta encontrar um hiperplano de separação entre duas classes para predizer a qual das classes um determinado exemplo faz parte. Esse hiperplano busca maximizar a distância entre os pontos mais próximos em relação a cada uma das classes e assim o SVM visa obter um hiperplano de margem máxima. Essa distância entre o hiperplano e o primeiro ponto de cada classe costuma ser chamada de margem.

O SVM é considerado um classificador binário, pois induz um hiperplano para separar duas classes. Classificadores com tais características podem ser aplicados na classificação multiclasse por meio de duas estratégias: (i) um-contra-todos e (ii) um-contra-um. Na estratégia **um-contra-todos**, são induzidos classificadores binários para cada classe $c_j \in \mathcal{C}$, na qual um documento d_i pertence a classe c_j será tratado como positivo (y_{di} = +1) e um documento d_k pertencente à classe c_l , $c_l \neq c_j$ será tratado como negativo (y_{dk} = -1). A classificação se dá pelo classificador que disparar a classe positiva. Em caso de empates pode-se utilizar a confiança de classificação como desempate [Tan et al. 2005].

Na estratégia **um-contra-um** são construídos classificadores binários para cada par de classes (c_i , c_j) $\in \mathcal{C}$. Instâncias que não pertencem à c_i ou c_j são ignoradas na

construção do classificador binário para as classes c_i e c_j . É realizada uma votação e a classe com maior número de votos é utilizada para definir a classe de um novo documento [Tan et al. 2005].

4.3.4. C4.5

O C4.5 é um algoritmo utilizado para criar uma árvore de decisão [Quinlan 1993]. Para gerar a árvore de decisão, em cada nó o algoritmo C4.5 escolhe o atributo dos dados que mais efetivamente particiona o seu conjunto de exemplos em subconjuntos, tendendo a uma categoria ou a outra. O critério de particionamento é o ganho de informação normalizado [Hall et al. 2009]. O atributo com maior ganho de informação normalizado é escolhido para tomar a decisão e cada decisão gera uma participação no conjunto de dados. O algoritmo C4.5 então repete a etapa anterior nas partições menores até atingir uma condição de parada. O algoritmo gera partições até que:

(i) Uma partição pura seja encontrada ou até que seja encontrado uma partição impura com um determinado grau de confiança.
(ii) Não há mais partições possíveis de serem geradas.

Ao final da condição de parada, cada partição (nós folhas) é rotulada com a classe majoritária dos exemplos contidos na mesma. Neste trabalho foi utilizado o algoritmo J48, que é uma implementação de código aberto em Java do algoritmo C4.5 no aplicativo de mineração de dados Weka [Hall et al. 2009].

4.3.5. IMBHN (*Inductive Model Based on Bipartite Heterogeneous*)

O algoritmo IMBHN [Rossi et al. 2014] induz um modelo de classificação a partir de uma rede heterogênea bipartida. O algoritmo calcula a influência de cada termo dos documentos do conjunto para cada classe, e assim induz um vetor de pesos dos termos para cada classe. O algoritmo é capaz de atribuir pesos negativos ao atributos, fazendo com que alguns termos diminuam o peso de um documento em uma classe. O algoritmo IMBHN tem três etapas principais: (i) definição dos pesos iniciais do vetor de pesos, (ii) cálculo dos erros (pesos negativos), e (iii) cálculo do pesos ajustados.

Na primeira etapa, inicialização dos pesos iniciais do vetor de peso, o peso para os termos são definidos. Os valores de peso podem ser 0, escolhidos aleatoriamente, ou é considerada a probabilidade de cada termo pertencer a determinada classe. Na segunda etapa, cálculo de erro, um vetor de saída para cada documento é calculado, onde cada posição do vetor é obtida pela soma dos valores dos pesos das relações documento-termo, multiplicas pelo peso de cada termo para cada classe. Na terceira etapa, passo de ajuste de pesos, o erro de cada documento para cada classe é usado para atualizar o peso dos vetores de termos. Esses processos são repetidos iterativamente até que todos os documentos de treinamento sejam rotulados corretamente ou até atingir um máximo de iterações.

4.4. Avaliação experimental

Nessa seção serão apresentados detalhes sobre a coleção de notícias e características das representações das mesmas, a configuração experimental e os resultados obtidos neste trabalho.

4.4.1. Coleção de notícias e características das representações

Para a realização dos experimentos deste trabalho foram utilizadas as coleções de notícias:

- Eventos do Milho.
- Notícias do Brasil.
- Reuters 21578.

A coleção notícias do Brasil foi dividida em duas coleções, sendo uma coleção contendo notícias com os seguinte temas: auto esporte, educação, gastronomia, tecnologia, beleza, empregos, meio ambiente, variedades, carnaval, entretenimento, mundo, cinema, esportes, política, economia, famosos e saúde. Essa coleção de notícias foi denominada Notícias do Brasil (Variedades). A segunda coleção de notícias obtida através da base Notícias do Brasil foi denominada Notícias do Brasil (Estados) e contém notícias com temas relacionados as unidades federativas do Brasil. Para todas as coleções de notícias utilizadas neste trabalho houve um pré-processamento, afim de:

(i) Padronizar as caixas.
(ii) Remover as *stopwords*.
(iii) Simplificar os termos.

Ao final da realização de todos os pré-processamentos foram geradas representações *bag-of-words* e redes bipartidas. As características das coleções Eventos do Milho, Notícias do Brasil (Variedades), Notícias do Brasil(Estados) e Reuters 21578 são apresentadas nas Tabelas 1, 2, 3 e 4 respectivamente.

Tabela 1. Base Eventos do Milho: 527 exemplos e 358 atributos.

Nº	Rótulo	Número de Documentos
1	EUA	55
2	alta_preco_milho	67
3	aumento_producao	52
4	comercializacao	52
5	expectativa_mercado	73
6	exportacao	28
7	intervencao_governo	55
8	investimento_milho	30
9	queda_preco_milho	34
10	reducao_producao	36
11	transgenico	45

4.4.2. Configuração Experimental

Para a realização dos experimentos deste trabalho foram considerados três cenários, sendo um com 10 exemplos rotulados de cada classe, outro com 20 exemplos rotulados e por último um conjunto com 50 documentos rotulados em cada classe, sendo que o restante dos documento de cada coleção foi utilizado para os testes de avaliação. Essa divisão

Tabela 2. Base Reuters 21578: 9075 exemplos e 10400 atributos.

Nº	Rótulo	Número de Documentos
1	acq	2259
2	alum	45
3	cocoa	54
4	coffee	112
5	copper	51
6	cpi	74
7	crude	359
8	crudenat-gas	50
9	crudeship	43
10	earn	3814
11	gnp	79
12	gold	92
13	grain	43
14	graincorn	73
15	grainwheat	132
16	interest	246
17	ipi	44
18	iron-steel	43
19	jobs	51
20	money-fx	290
21	money-fxdlr	84
22	money-fxinterest	137
23	money-supply	148
24	nat-gas	43
25	reserves	52
26	rubber	39
27	ship	151
28	sugar	134
29	trade	333

da coleção de notícias foi realizada com o objetivo de avaliar se o número de exemplos rotulados influi na performance dos classificadores ao longo do tempo.

A análise de performance dos modelos de classificação foi realizada nos seguintes períodos de tempo: anual, semestral, trimestral, bimestral e mensal. As medidas de avaliação de performance de classificação utilizadas foram: F1, Micro F1 e Macro F1 [Rossi 2016]. Os algoritmos de aprendizado de máquina indutivo supervisionado, juntamente com os valores dos respectivos parâmetros utilizados nesta avaliação experimental, foram:

- MNB: Sem parâmetros.
- C4.5: Taxa de confiança em 0.25.
- KNN: K=7 com peso ponderado pela distância.
- SVM: Kernel Linear e C=1.

Tabela 3. Base Noticias do Brasil (Variedades): 118844 exemplos e 81708 atributos.

Nº	Rótulo	Número de Documentos
1	Auto_Esporte	515
2	Beleza	75
3	Carnaval	2092
4	Cinema	494
5	Economia	9311
6	Educação	3574
7	Empregos	3371
8	Entretenimento	14257
9	Esportes	23628
10	Famosos	7906
11	Gastronomia	133
12	Meio_Ambiente	303
13	Mundo	24667
14	Política	5552
15	Saúde	1475
16	Tecnologia	5949
17	Variedades	15542

- IMBHN: Erro mínimo=0.01 e taxa de correção de erro=0.01 [Rossi 2016].

4.4.3. Resultados

Nas próximas seções são apresentados os resultados obtidos neste trabalho. Os resultados são exibidos em gráficos e foram agrupados pela base de notícias utilizada, tipo de algoritmo de aprendizado indutivo supervisionado e número de exemplos rotulados de cada conjunto de testes. No entanto, nesse momento, é importante definir alguns comportamentos que ocorreram nos gráficos afim de facilitar a leitura e compreensão dos resultados. Em conjuntos com 50 exemplos rotulados ocorreu uma sobreposição da performance de classificação de determinadas classes ao longo do tempo. Isso ocorreu devido a similaridade de comportamento de performance entre algumas classes da coleção. Devido a este fenômeno de sobreposição, parece haver menos dados plotados nos gráficos referentes aos conjuntos com 50 exemplos rotulados. Há também algumas classes que aparecem apenas em alguns períodos isolados, sendo representadas como pontos sem conexão no gráfico. No entanto, esses comportamentos em questão não interferiram na avaliação dos resultados.

4.4.3.1 Resultado: Base Eventos do Milho

Nessa seção são apresentados os resultados obtidos através dos testes executados na base de notícias Eventos do Milho. Os experimentos foram executados considerando a configuração experimental apresentada anteriormente. Os resultados são apresentados

Tabela 4. Base Noticias do Brasil (Estados): 72837 exemplos e 53423 atributos.

Nº	Rótulo	Número de Documentos
1	Acre	74
2	Amapá	57
3	Distrito_Federal	650
4	Espírito_Santo	199
5	Goiás	387
6	Maranhão	84
7	Mato_Grosso_do_Sul	142
8	Mato_Grosso	231
9	Minas_Gerais	307
10	Mundo	24668
11	Paraná	537
12	Paraíba	83
13	Pernambuco	251
14	Piauí	126
15	Rio_Grande_do_Sul	622
16	Rio_de_Janeiro	2372
17	Salvador	39386
18	Santa_Catarina	304
19	São_Paulo	2296
20	Tocantins	61

na sequência de figuras a seguir no intervalo da Figura 8 até a Figura 36.

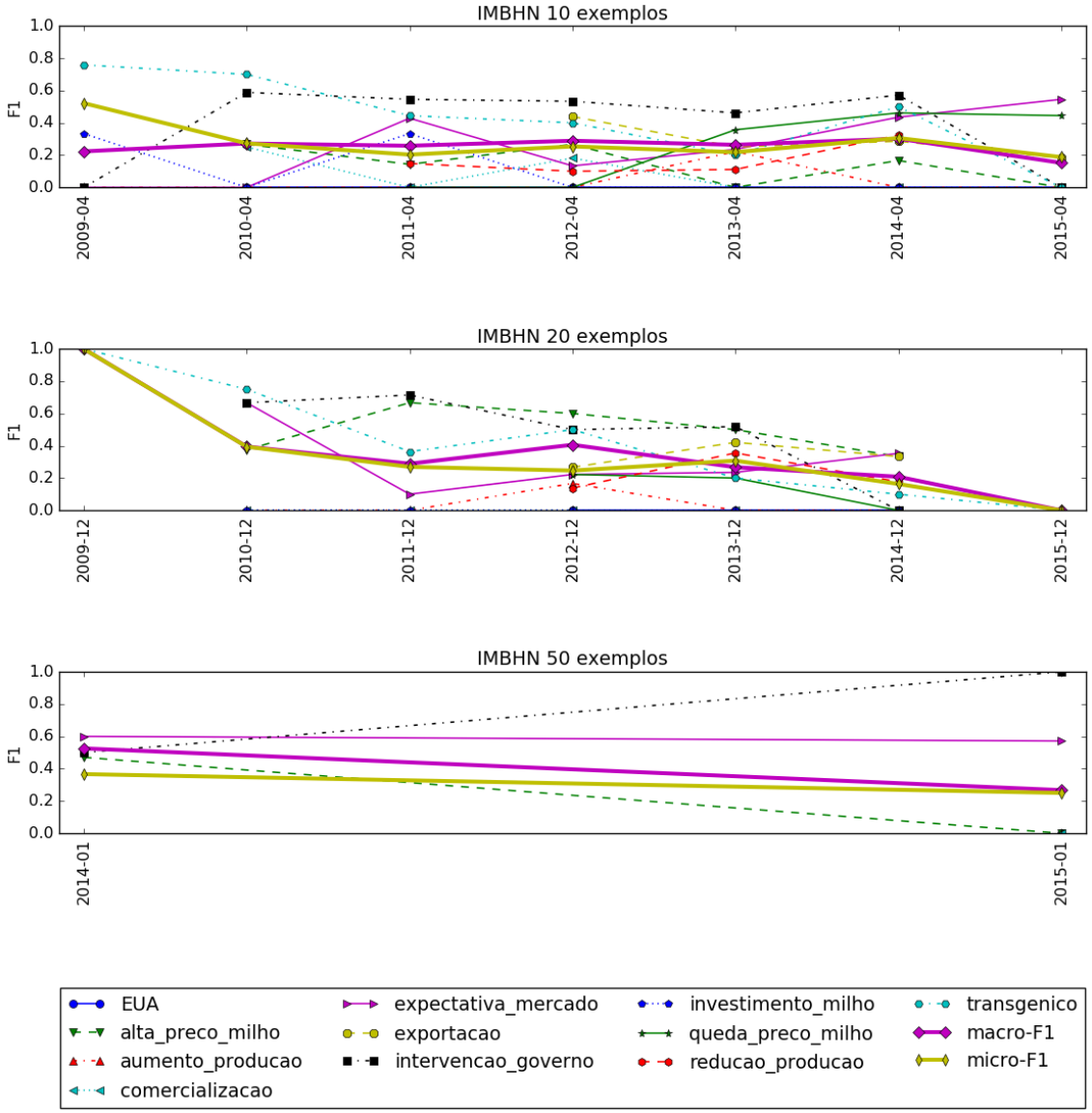

Figura 8. Resultado do algoritmo IMBHN em um conjunto com 10, 20 e 50 exemplos rotulados. Base: Eventos do Milho, período: Anual.

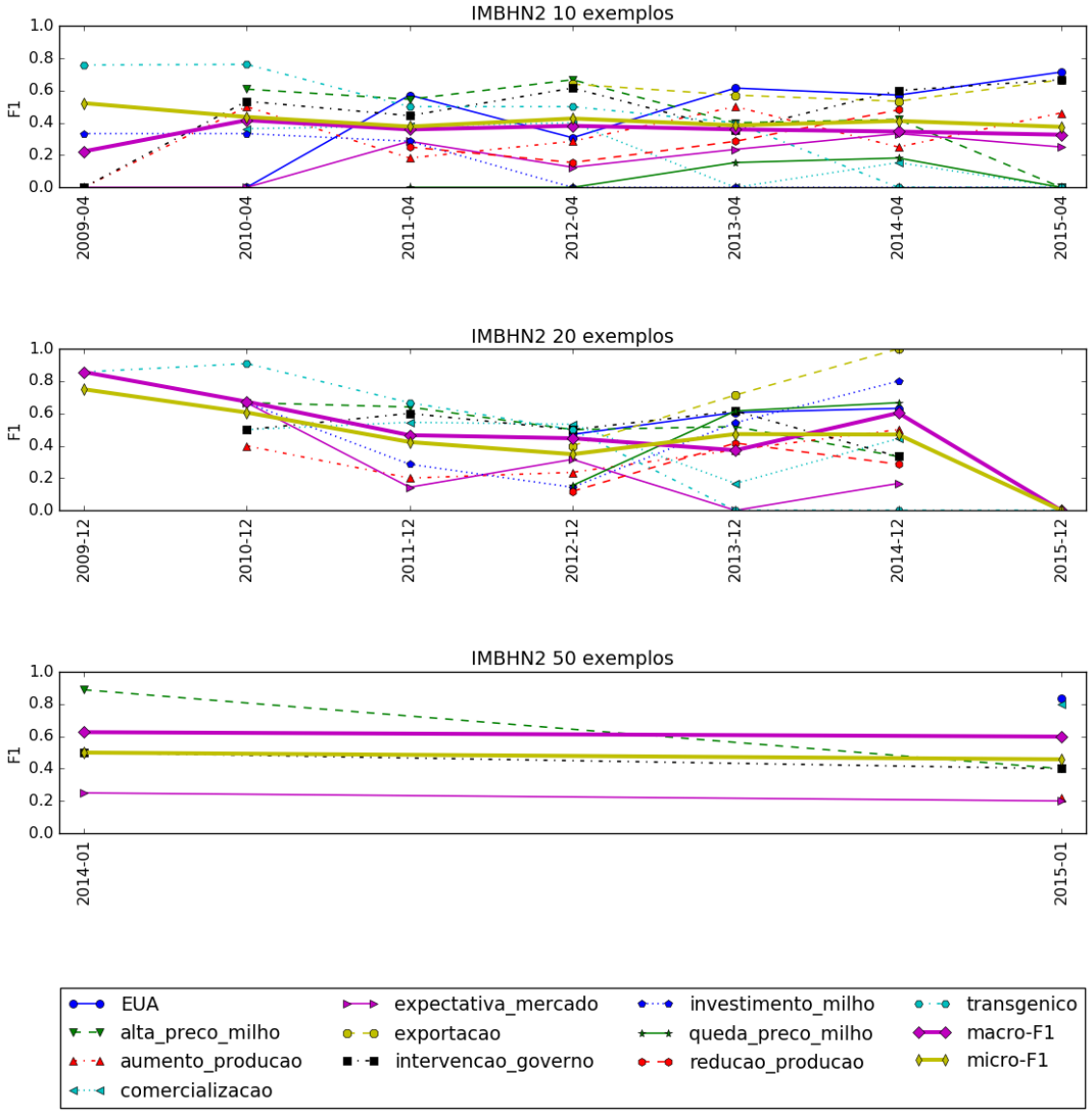

Figura 9. Resultado do algoritmo IMBHN2 em um conjunto com 10, 20 e 50 exemplos rotulados. Base: Eventos do Milho, período: Anual.

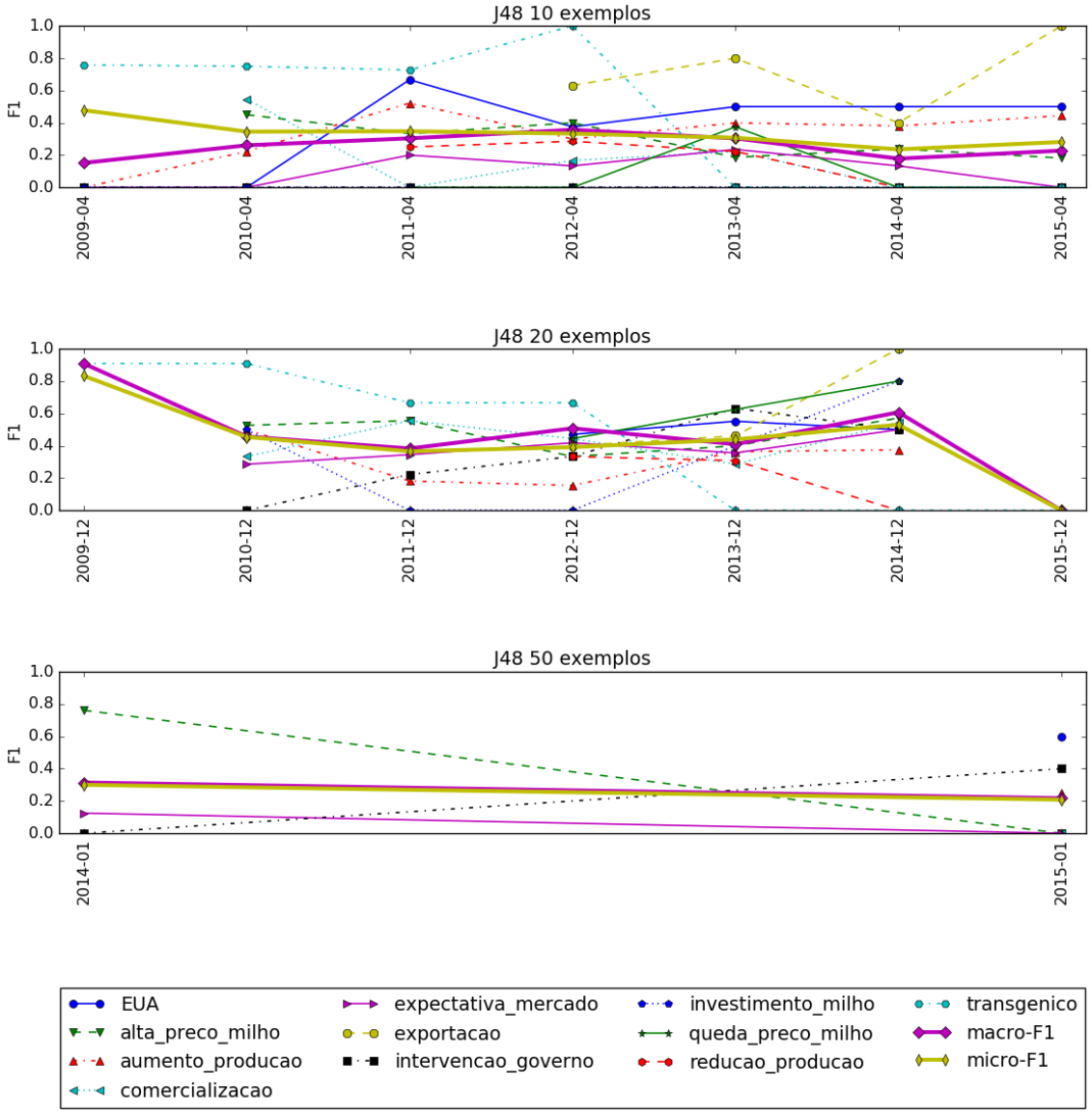

Figura 10. Resultado do algoritmo J48 em um conjunto com 10, 20 e 50 exemplos rotulados. Base: Eventos do Milho, período: Anual.

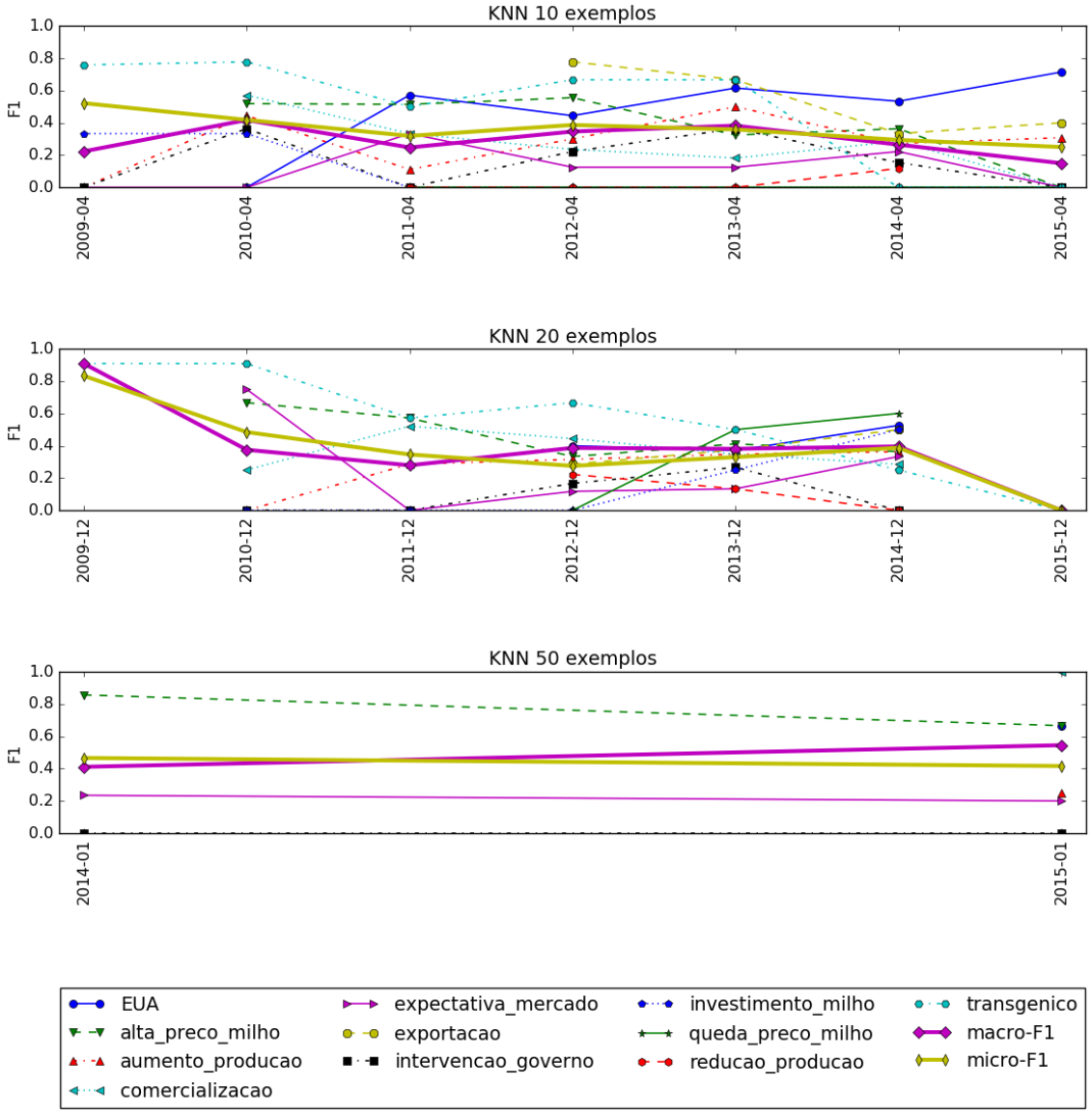

Figura 11. Resultado do algoritmo KNN em um conjunto com 10, 20 e 50 exemplos rotulados. Base: Eventos do Milho, período: Anual.

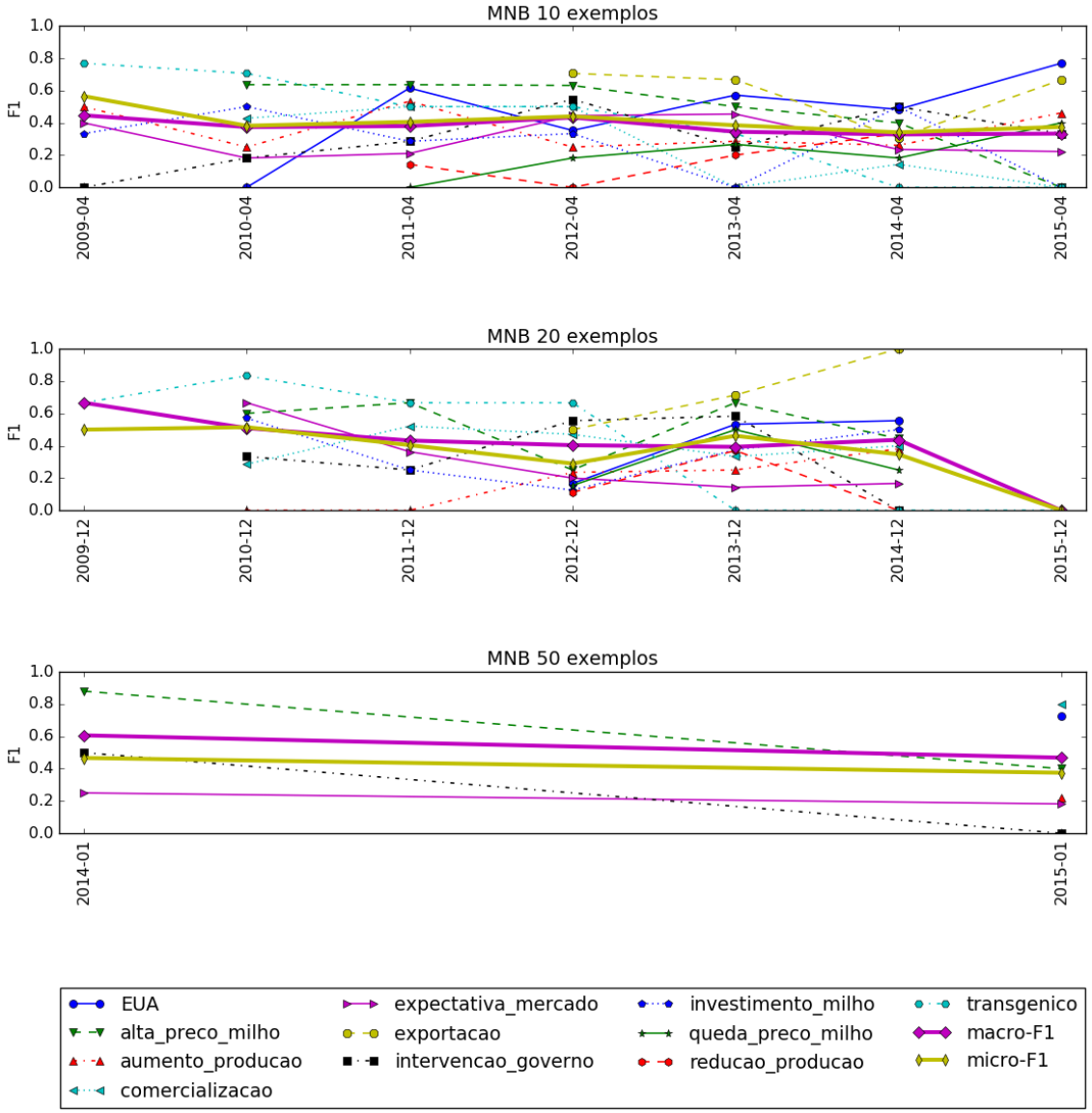

Figura 12. Resultado do algoritmo MNB em um conjunto com 10, 20 e 50 exemplos rotulados. Base: Eventos do Milho, período: Anual.

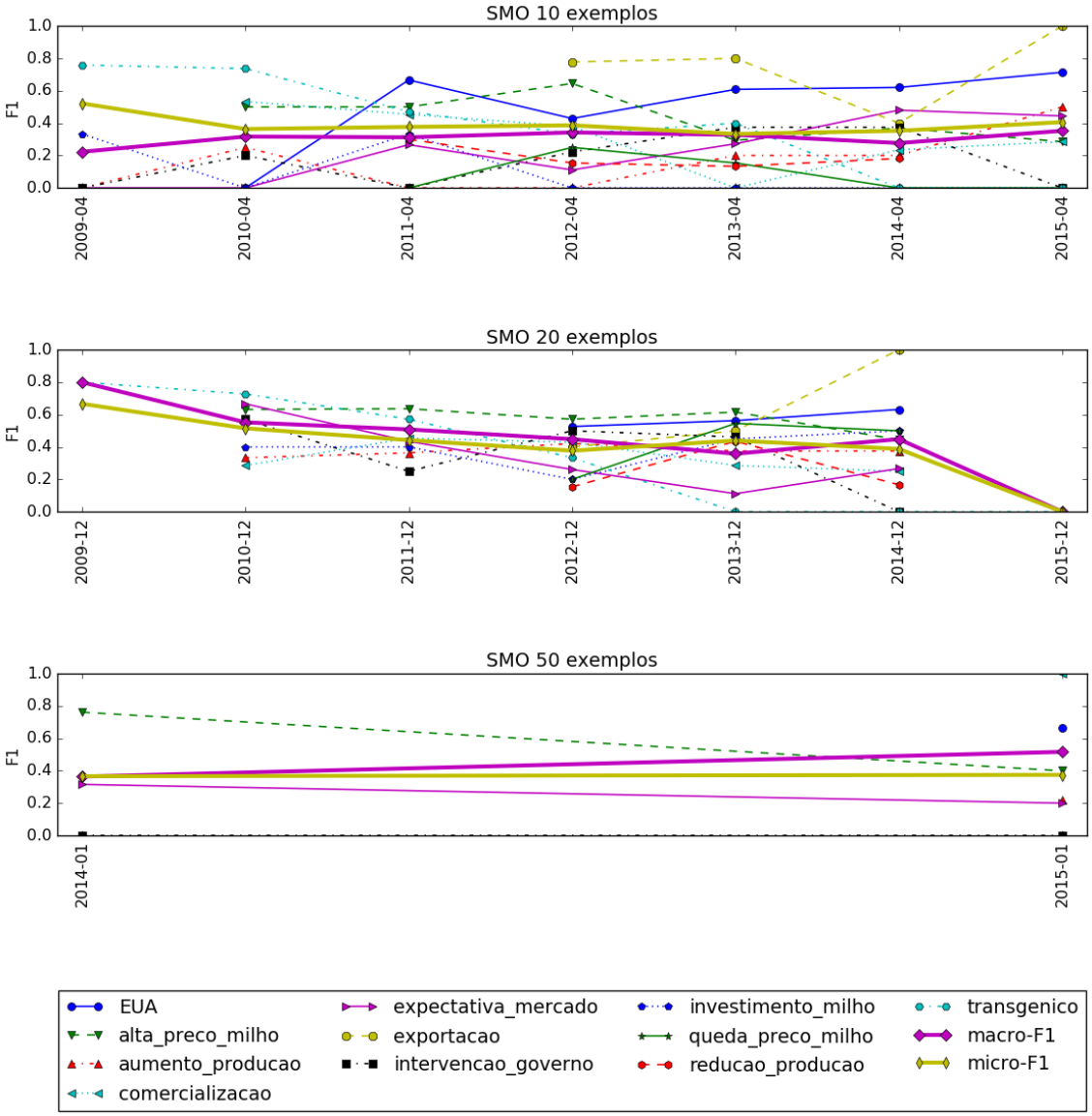

Figura 13. Resultado do algoritmo SMO em um conjunto com 10, 20 e 50 exemplos rotulados. Base: Eventos do Milho, período: Anual.

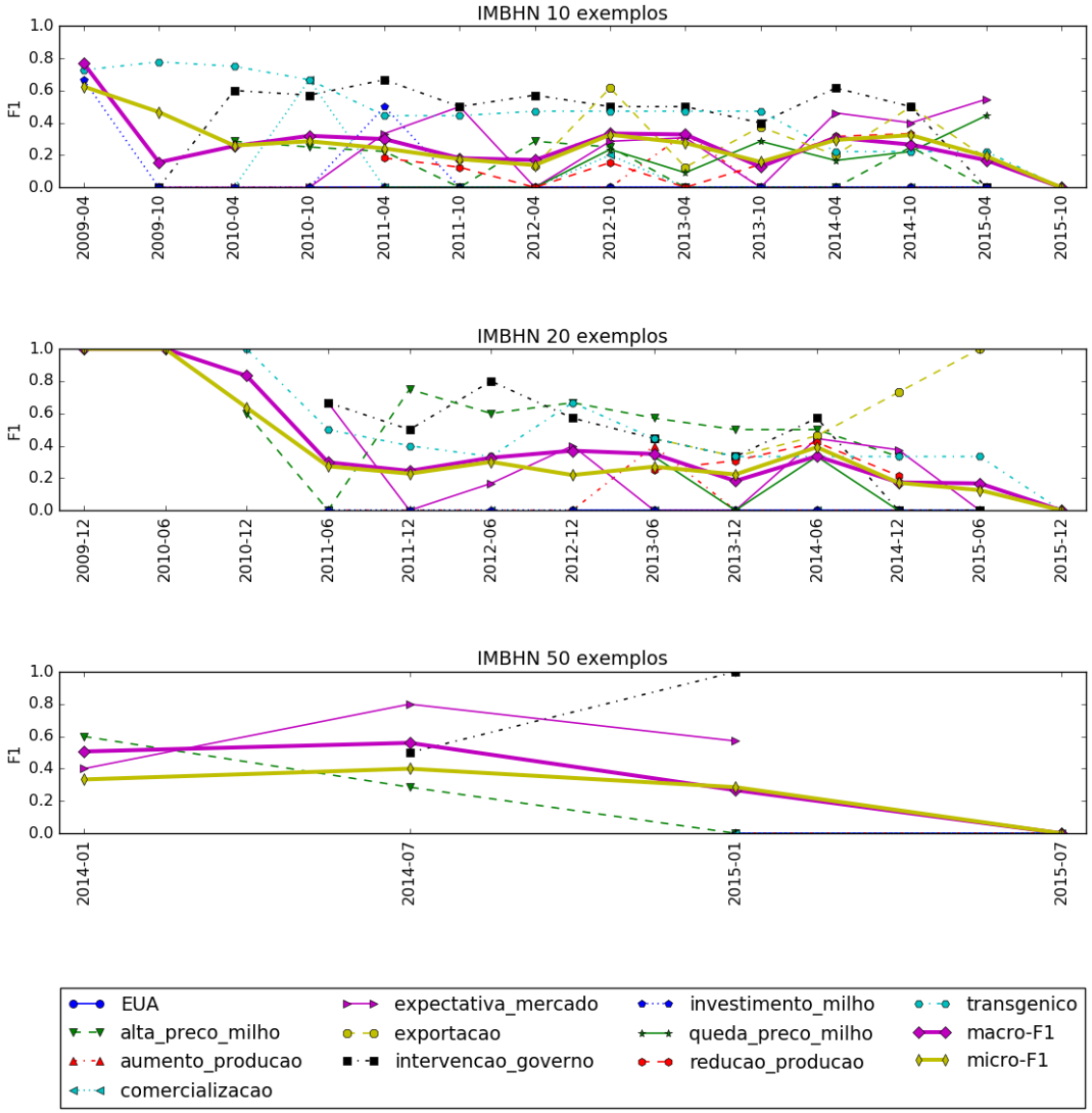

Figura 14. Resultado do algoritmo IMBHN em um conjunto com 10, 20 e 50 exemplos rotulados. Base: Eventos do Milho, período: Semestral.

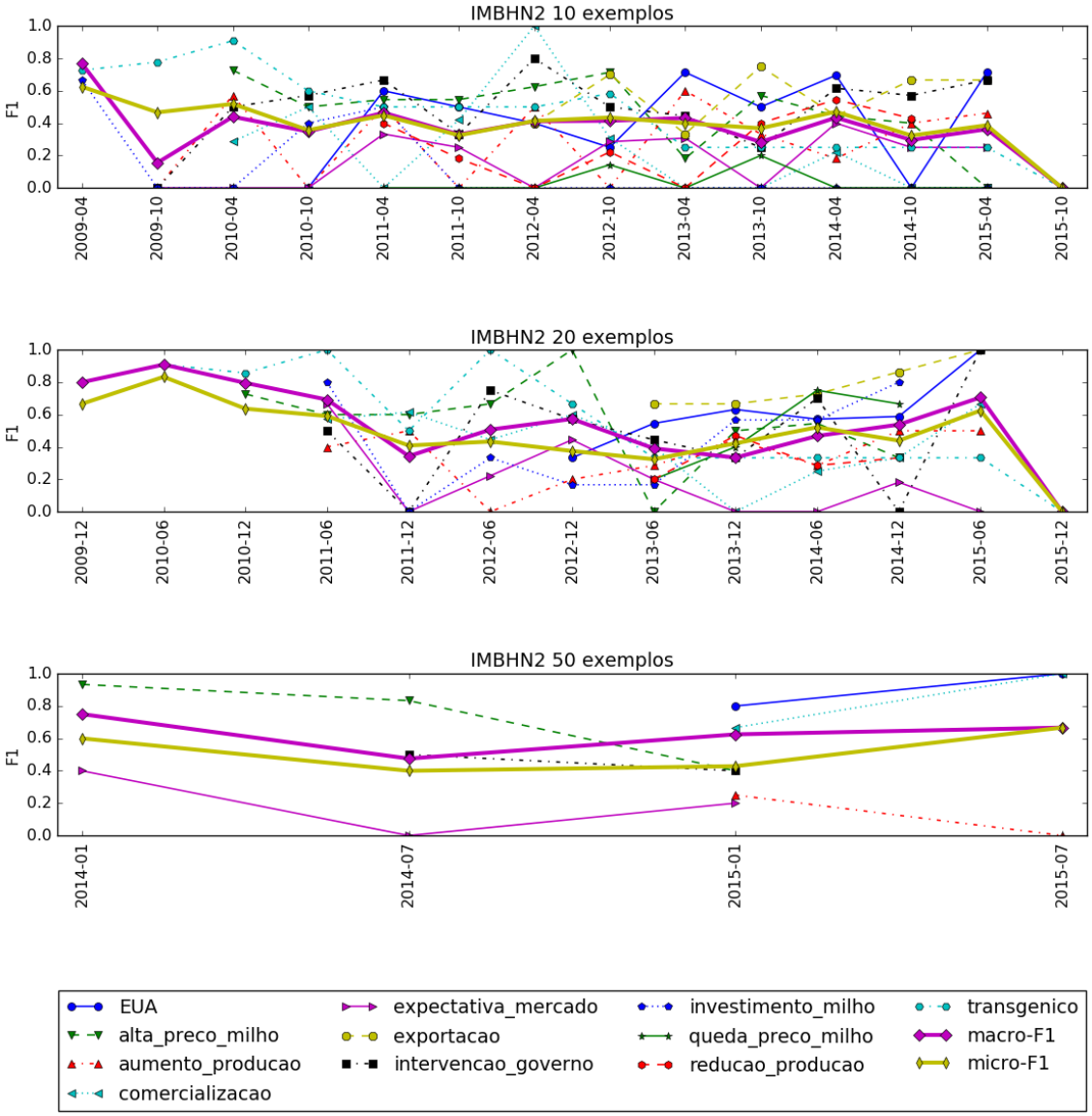

Figura 15. Resultado do algoritmo IMBHN2 em um conjunto com 10, 20 e 50 exemplos rotulados. Base: Eventos do Milho, período: Semestral.

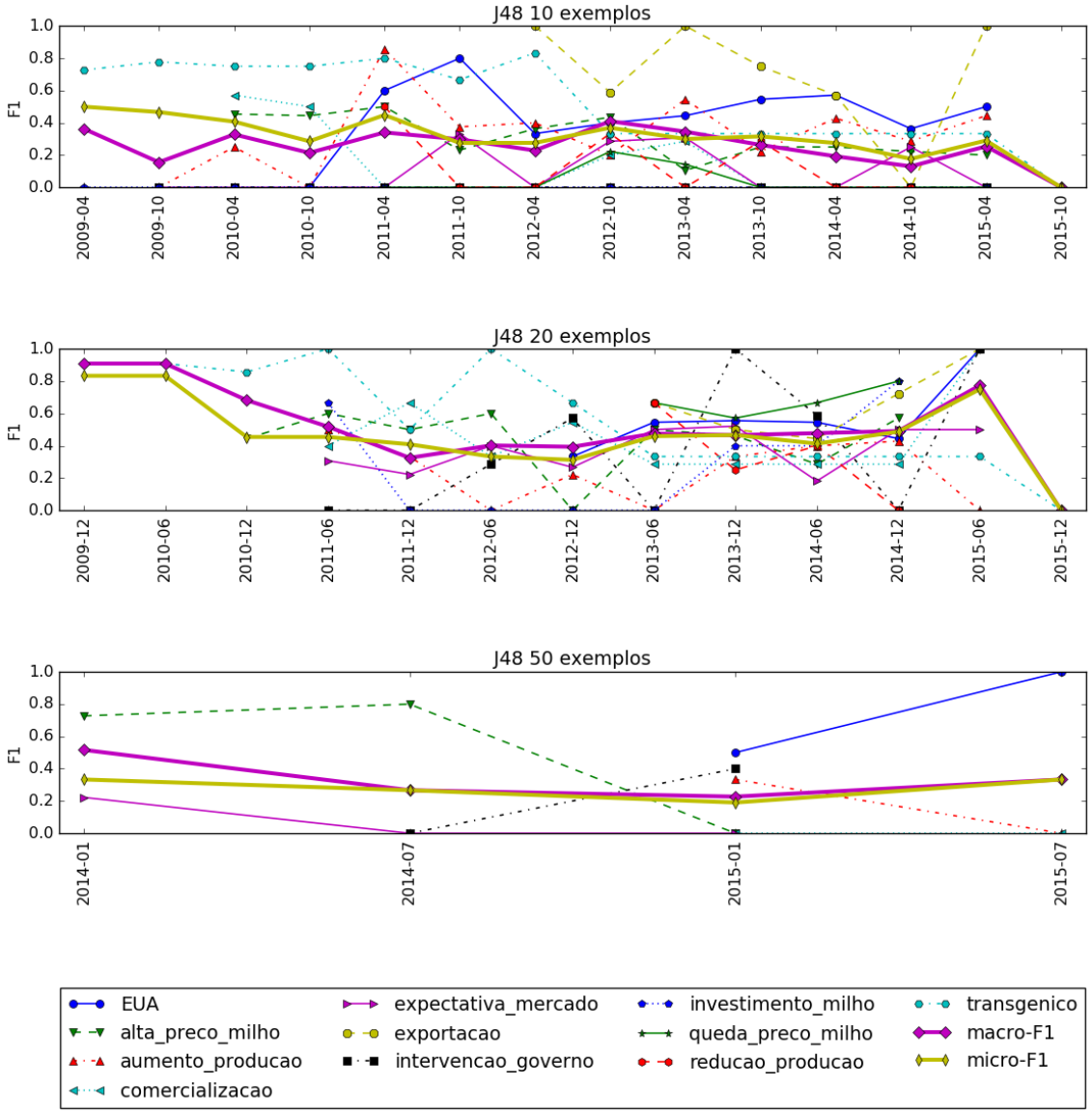

Figura 16. Resultado do algoritmo J48 em um conjunto com 10, 20 e 50 exemplos rotulados. Base: Eventos do Milho, período: Semestral.

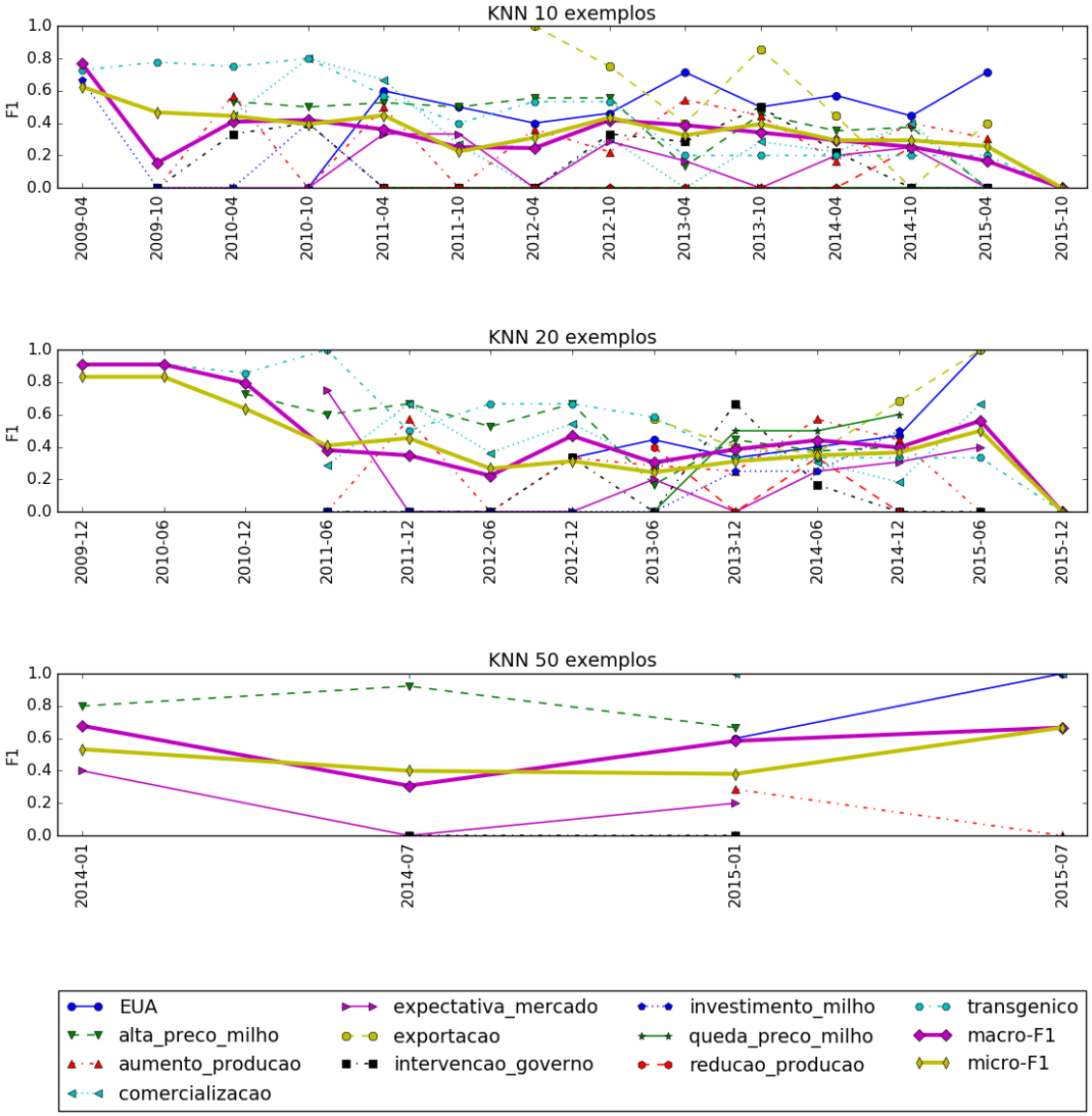

Figura 17. Resultado do algoritmo KNN em um conjunto com 10, 20 e 50 exemplos rotulados. Base: Eventos do Milho, período: Semestral.

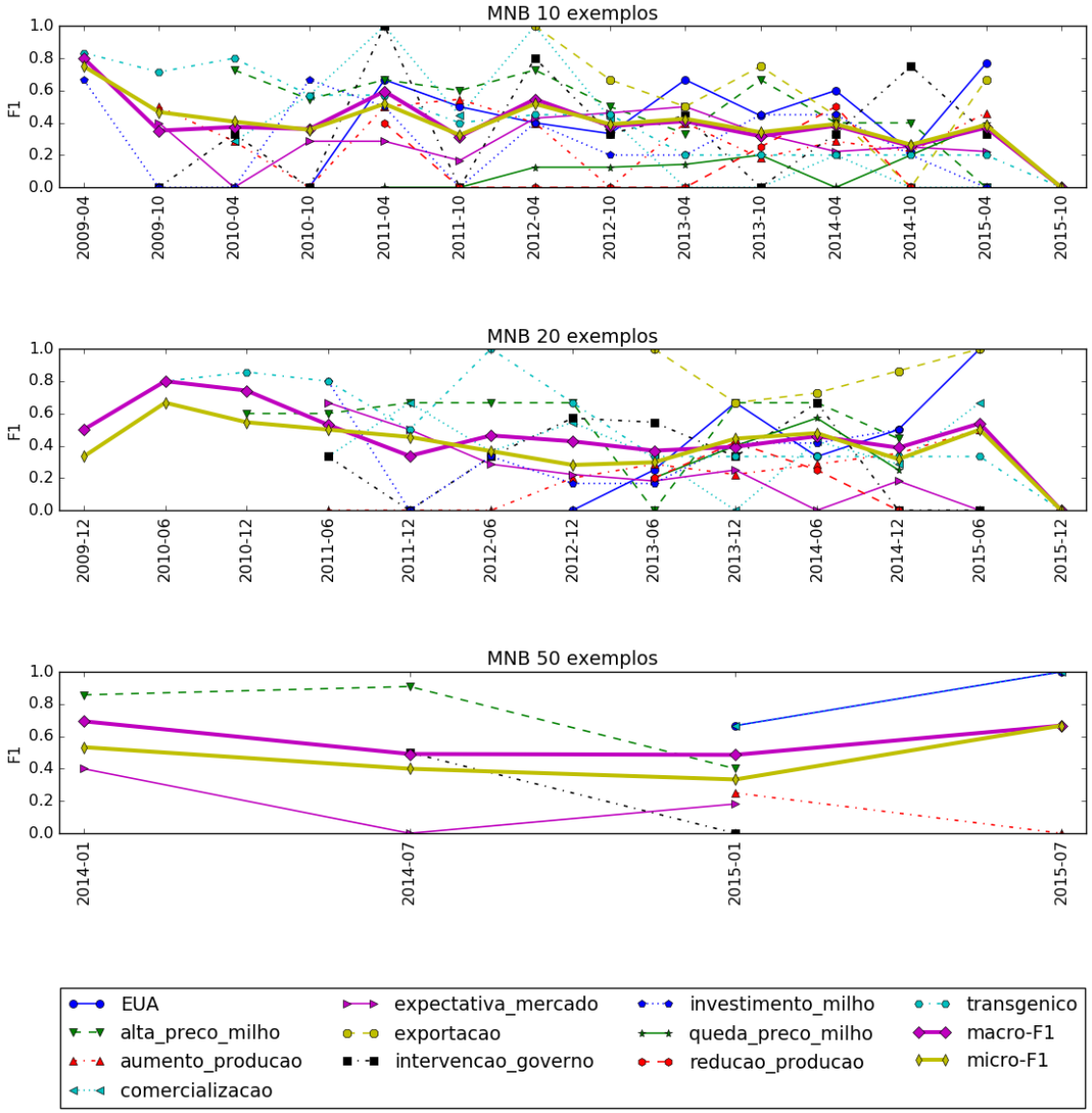

Figura 18. Resultado do algoritmo MNB em um conjunto com 10, 20 e 50 exemplos rotulados. Base: Eventos do Milho, período: Semestral.

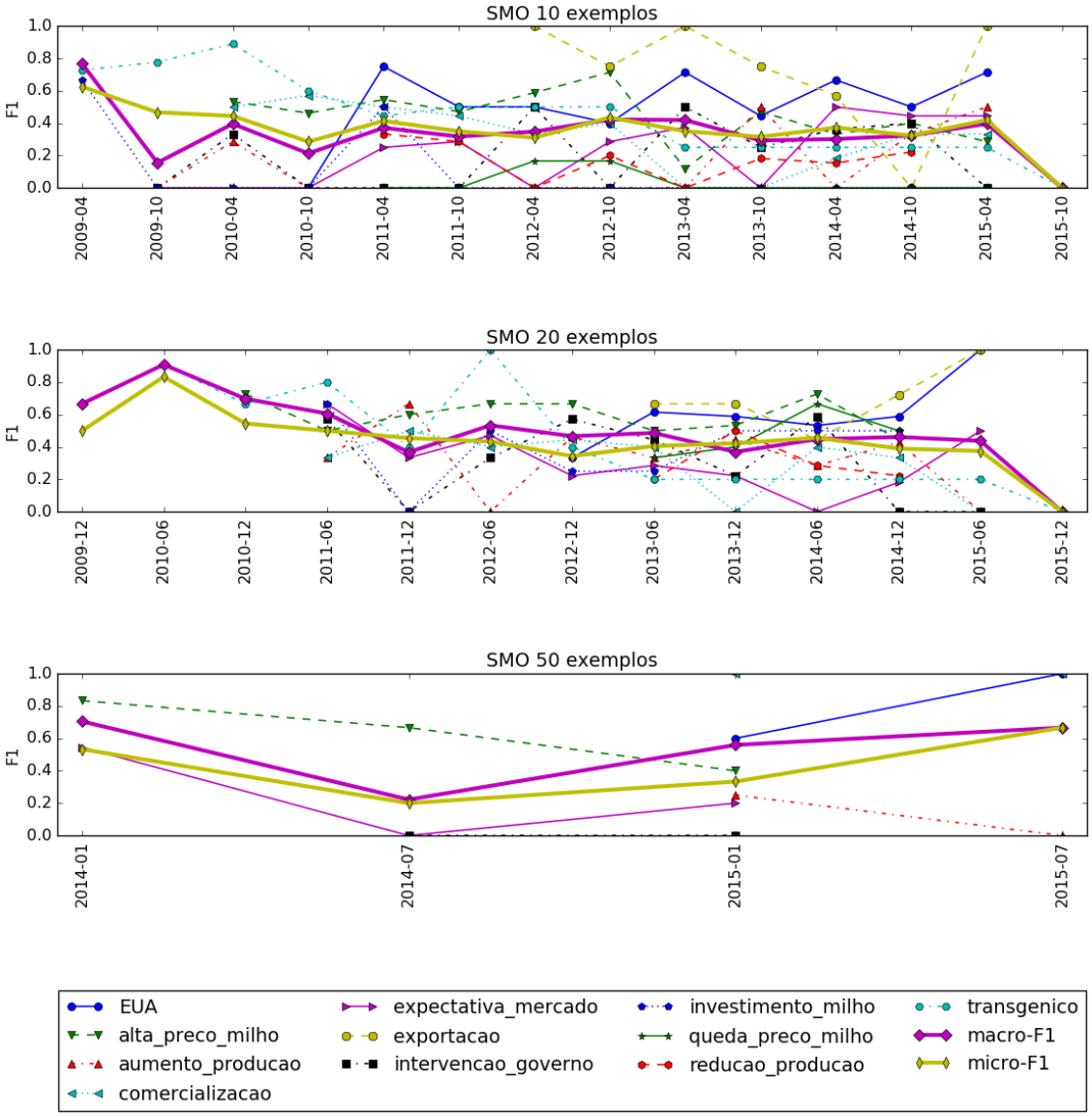

Figura 19. Resultado do algoritmo SMO em um conjunto com 10, 20 e 50 exemplos rotulados. Base: Eventos do Milho, período: Semestral.

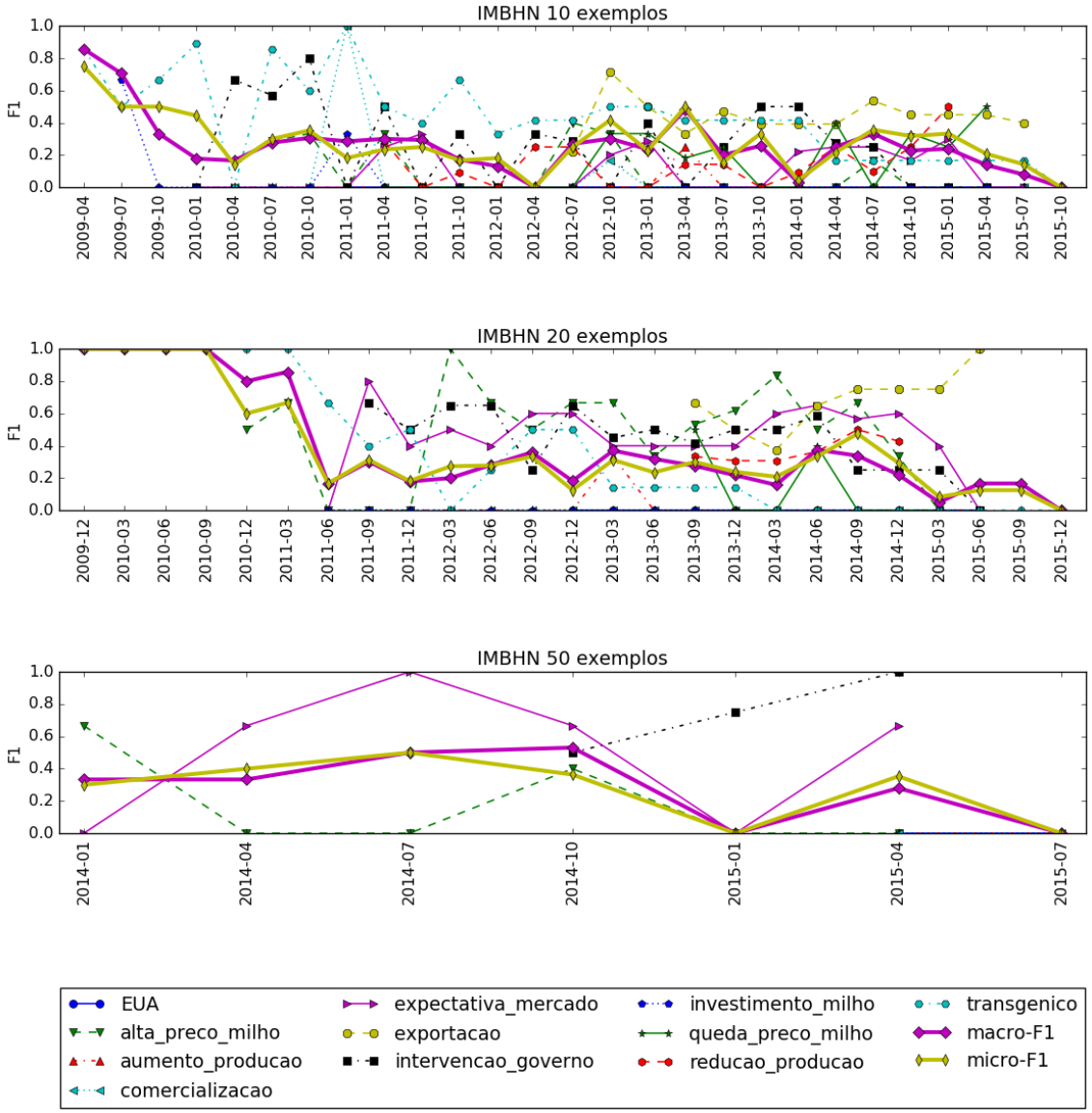

Figura 20. Resultado do algoritmo IMBHN em um conjunto com 10, 20 e 50 exemplos rotulados. Base: Eventos do Milho, período: Trimestral.

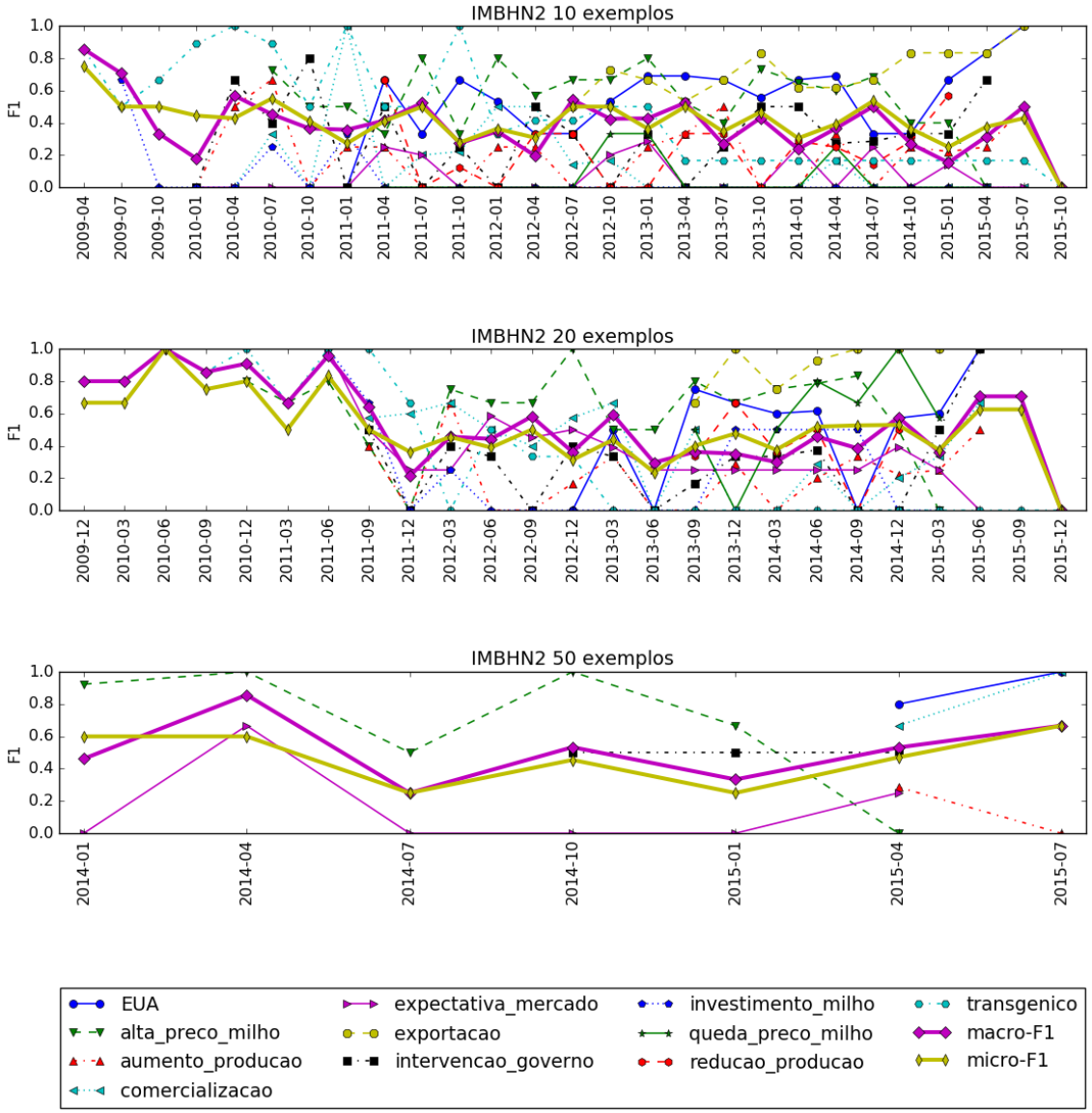

Figura 21. Resultado do algoritmo IMBHN2 em um conjunto com 10, 20 e 50 exemplos rotulados. Base: Eventos do Milho, período: Trimestral.

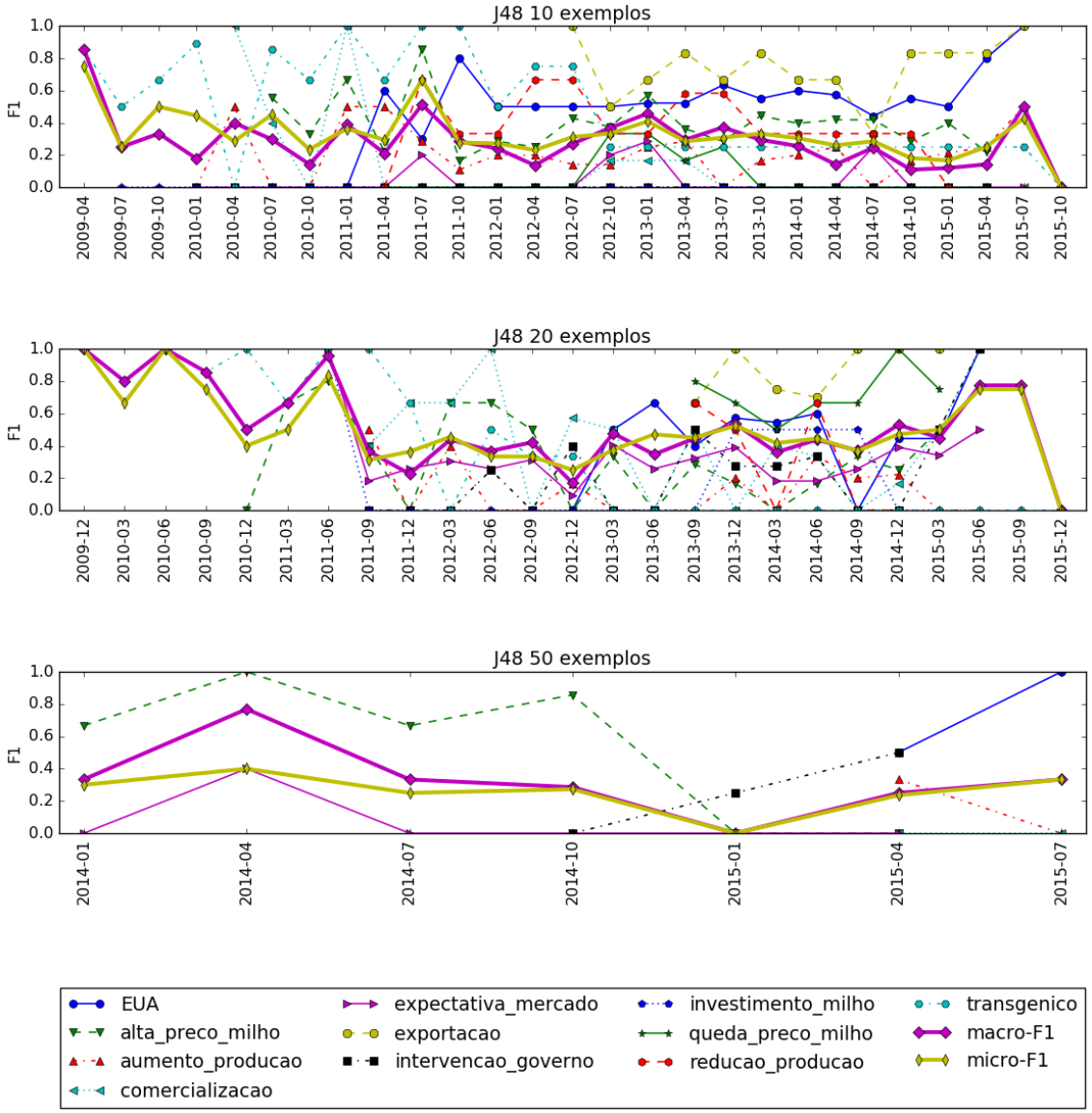

Figura 22. Resultado do algoritmo J48 em um conjunto com 10, 20 e 50 exemplos rotulados. Base: Eventos do Milho, período: Trimestral.

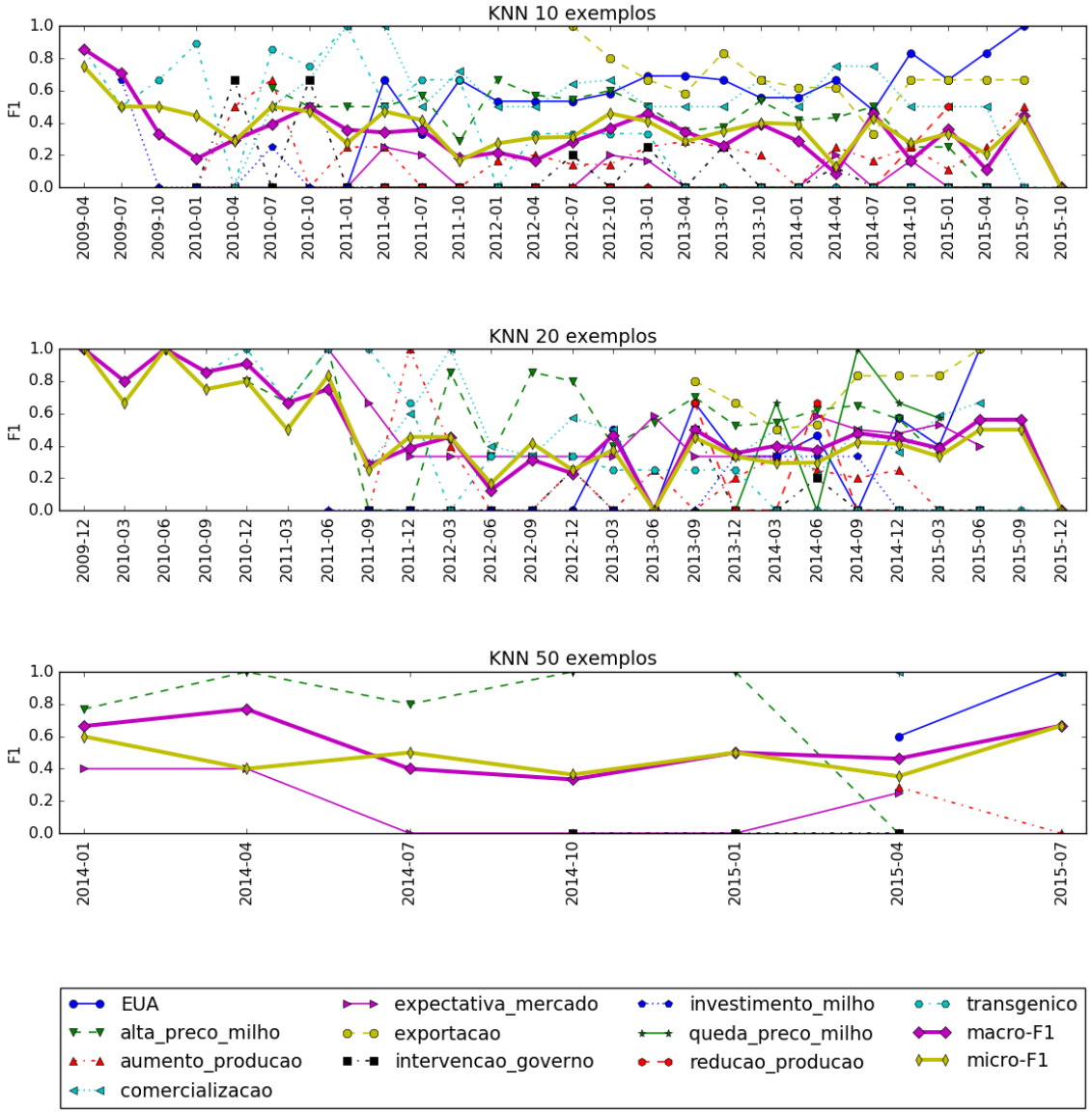

Figura 23. Resultado do algoritmo KNN em um conjunto com 10, 20 e 50 exemplos rotulados. Base: Eventos do Milho, período: Trimestral.

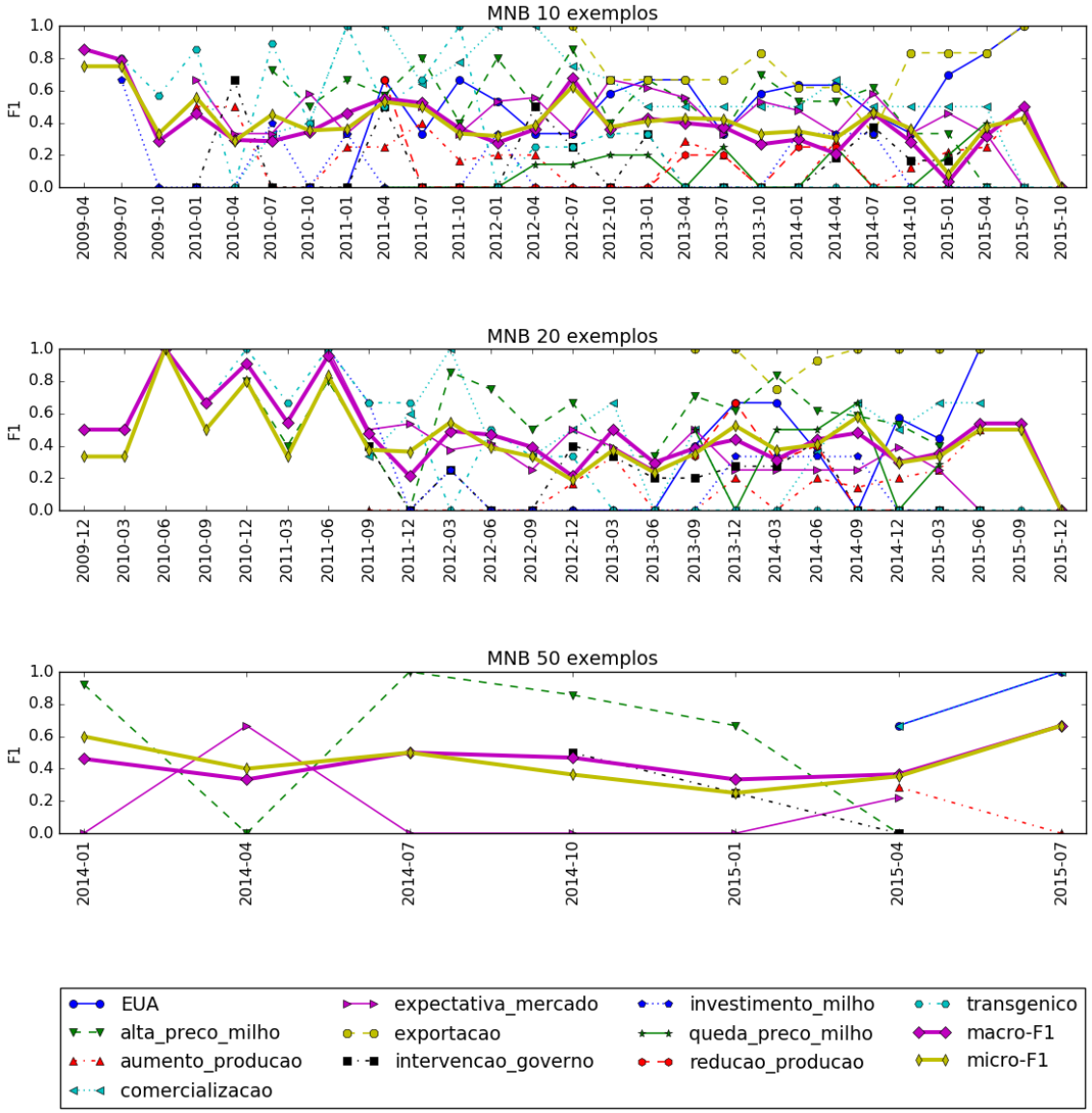

Figura 24. Resultado do algoritmo MNB em um conjunto com 10, 20 e 50 exemplos rotulados. Base: Eventos do Milho, período: Trimestral.

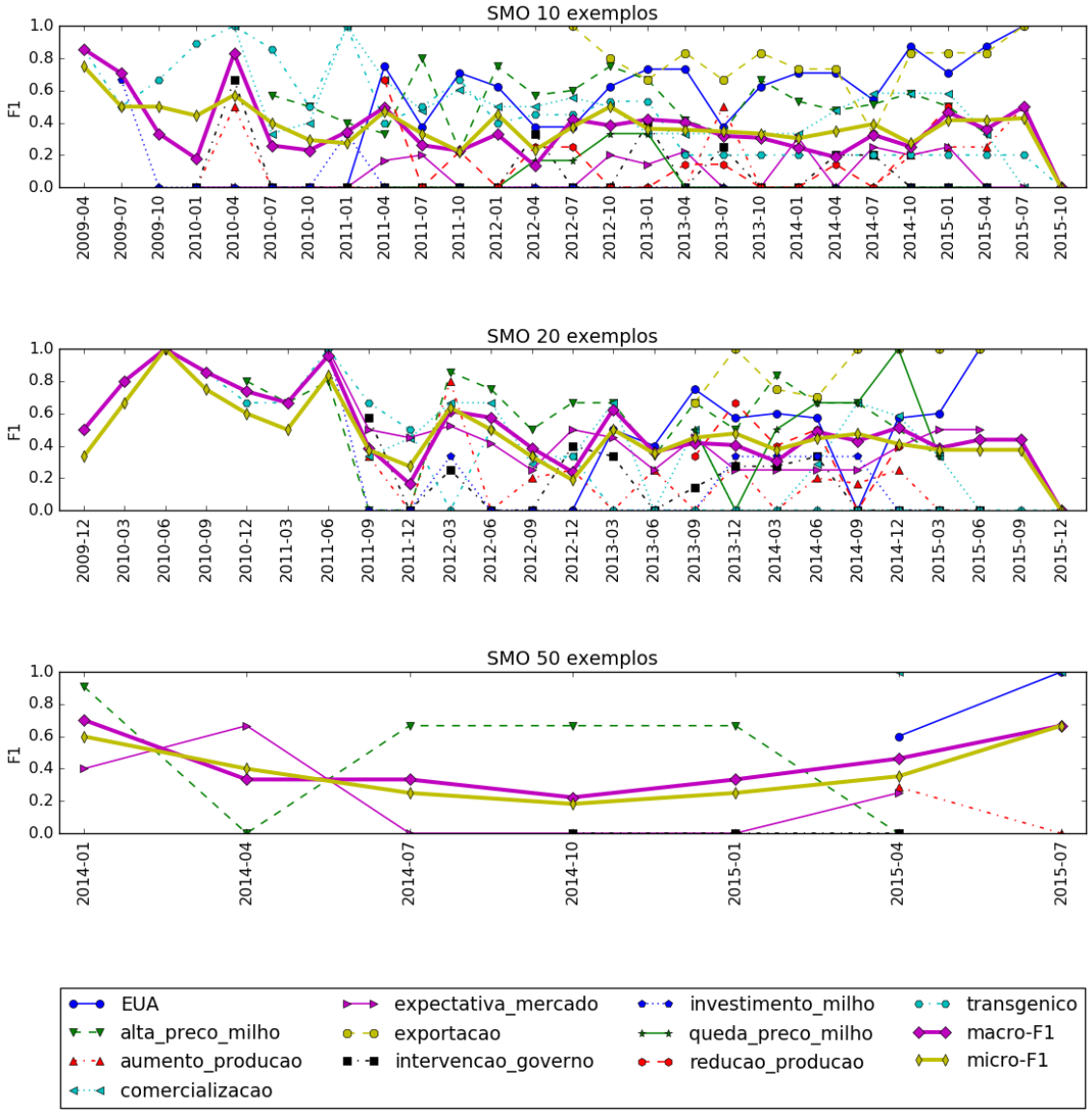

Figura 25. Resultado do algoritmo SMO em um conjunto com 10, 20 e 50 exemplos rotulados. Base: Eventos do Milho, período: Trimestral.

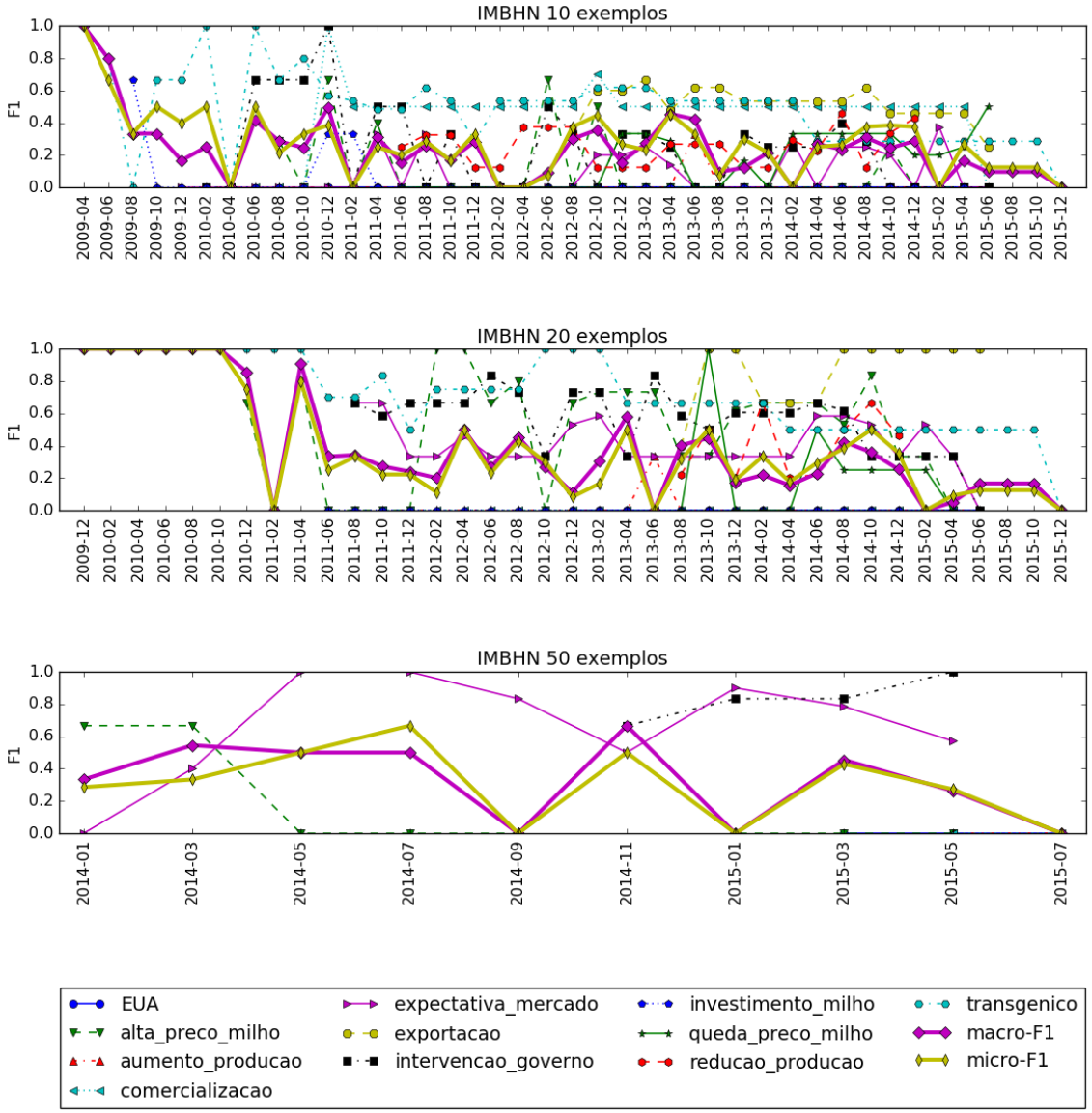

Figura 26. Resultado do algoritmo IMBHN em um conjunto com 10, 20 e 50 exemplos rotulados. Base: Eventos do Milho, período: Bimestral.

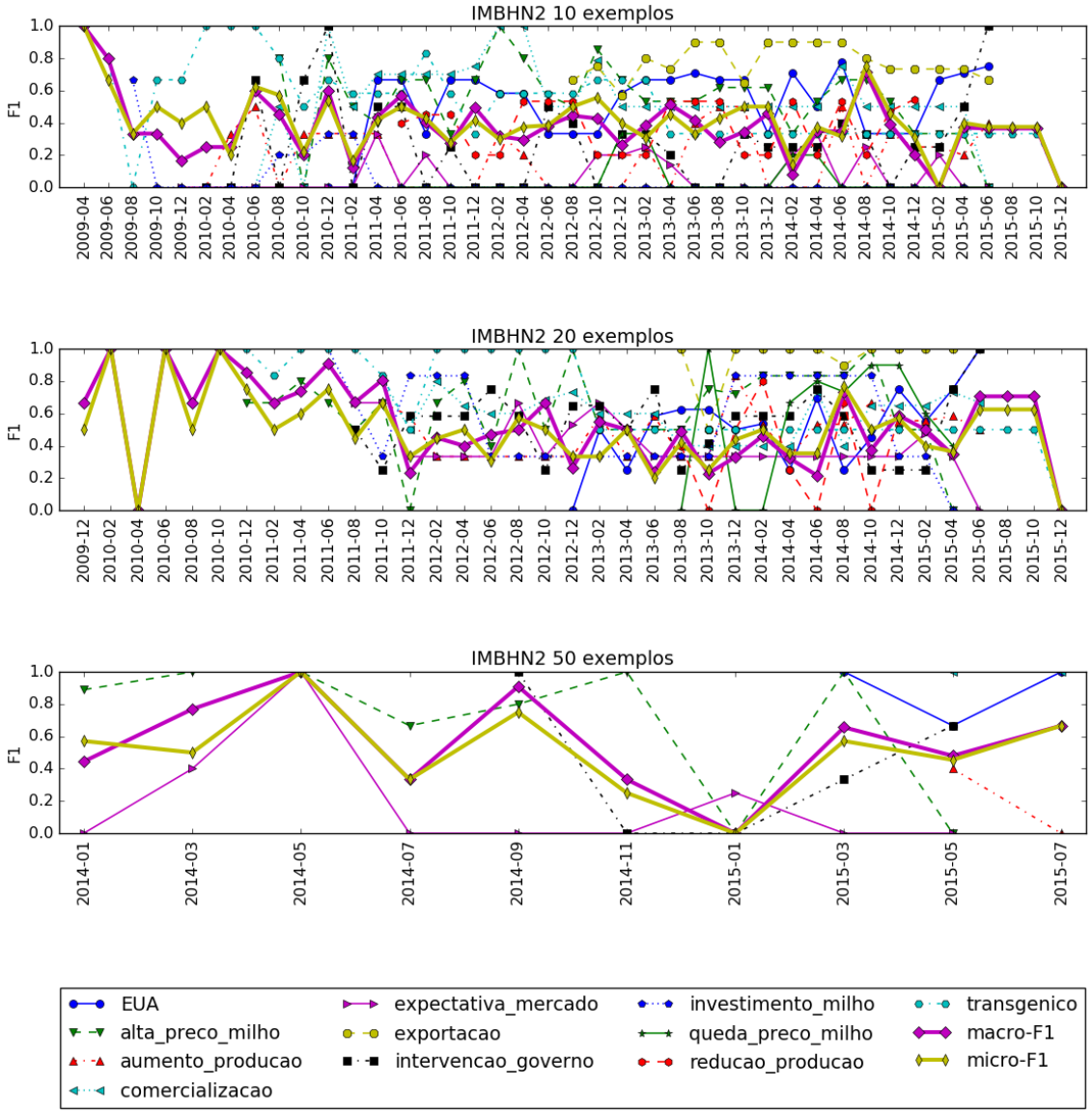

Figura 27. Resultado do algoritmo IMBHN2 em um conjunto com 10, 20 e 50 exemplos rotulados. Base: Eventos do Milho, período: Bimestral.

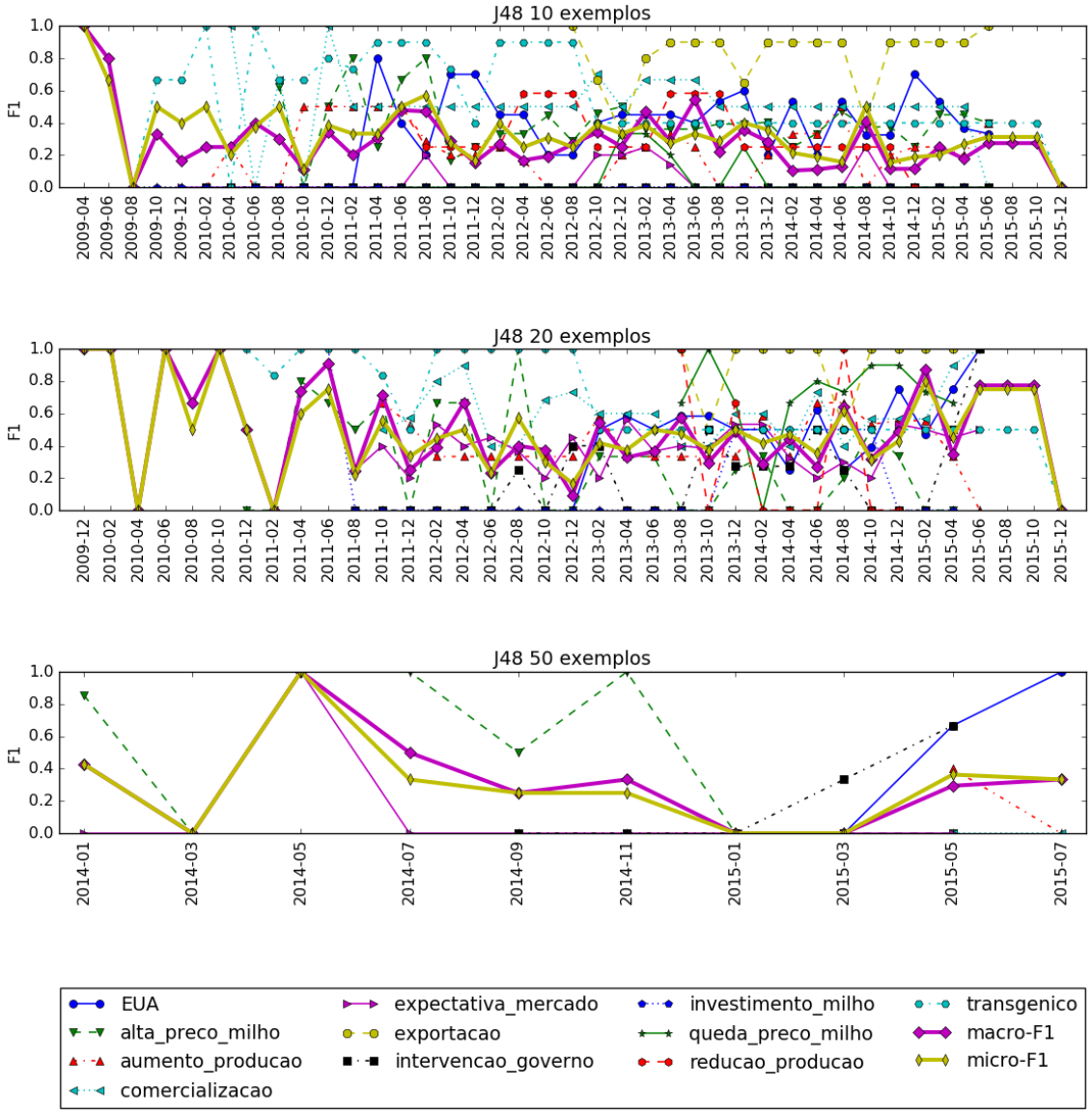

Figura 28. Resultado do algoritmo J48 em um conjunto com 10, 20 e 50 exemplos rotulados. Base: Eventos do Milho, período: Bimestral.

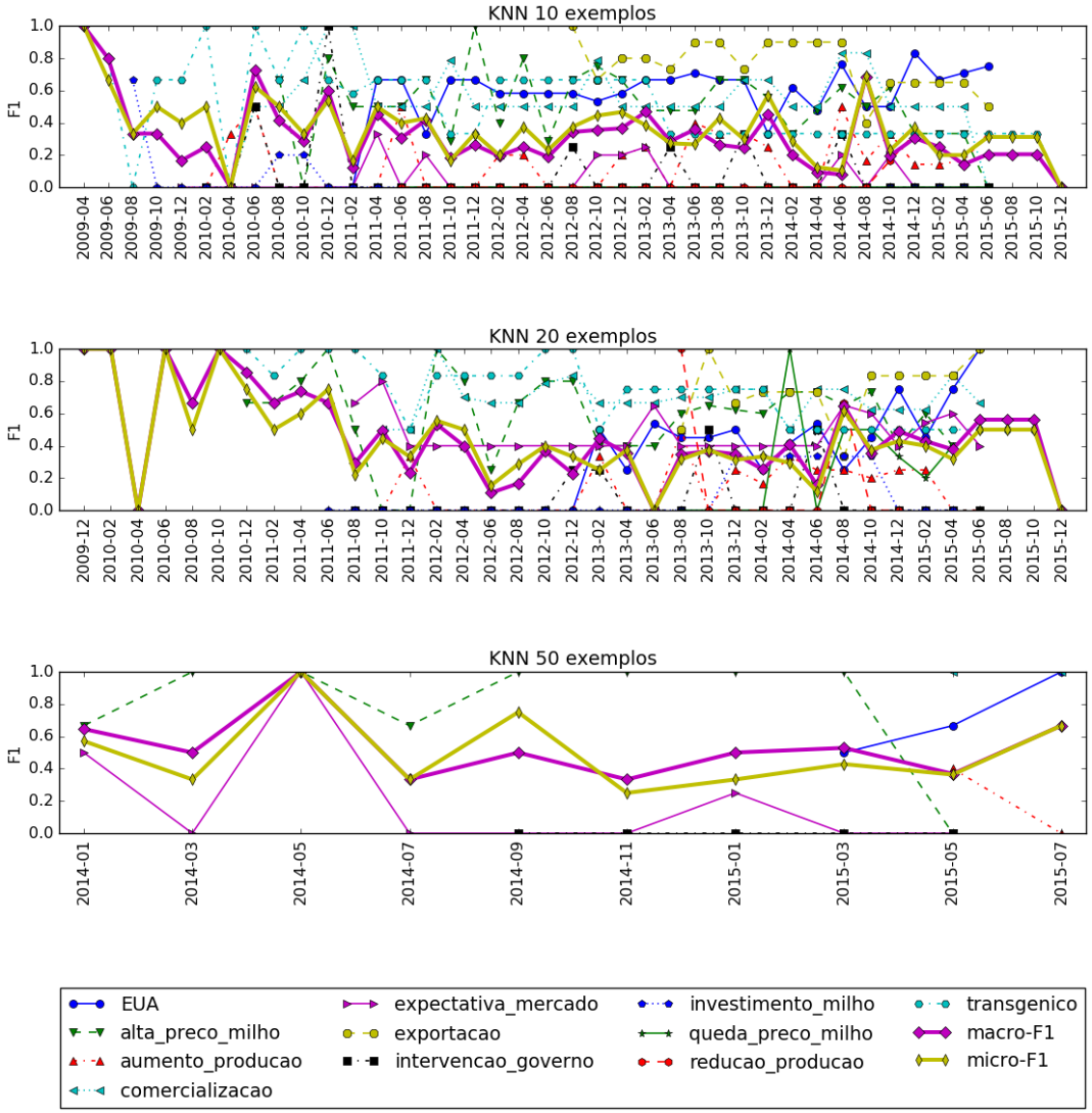

Figura 29. Resultado do algoritmo KNN em um conjunto com 10, 20 e 50 exemplos rotulados. Base: Eventos do Milho, período: Bimestral.

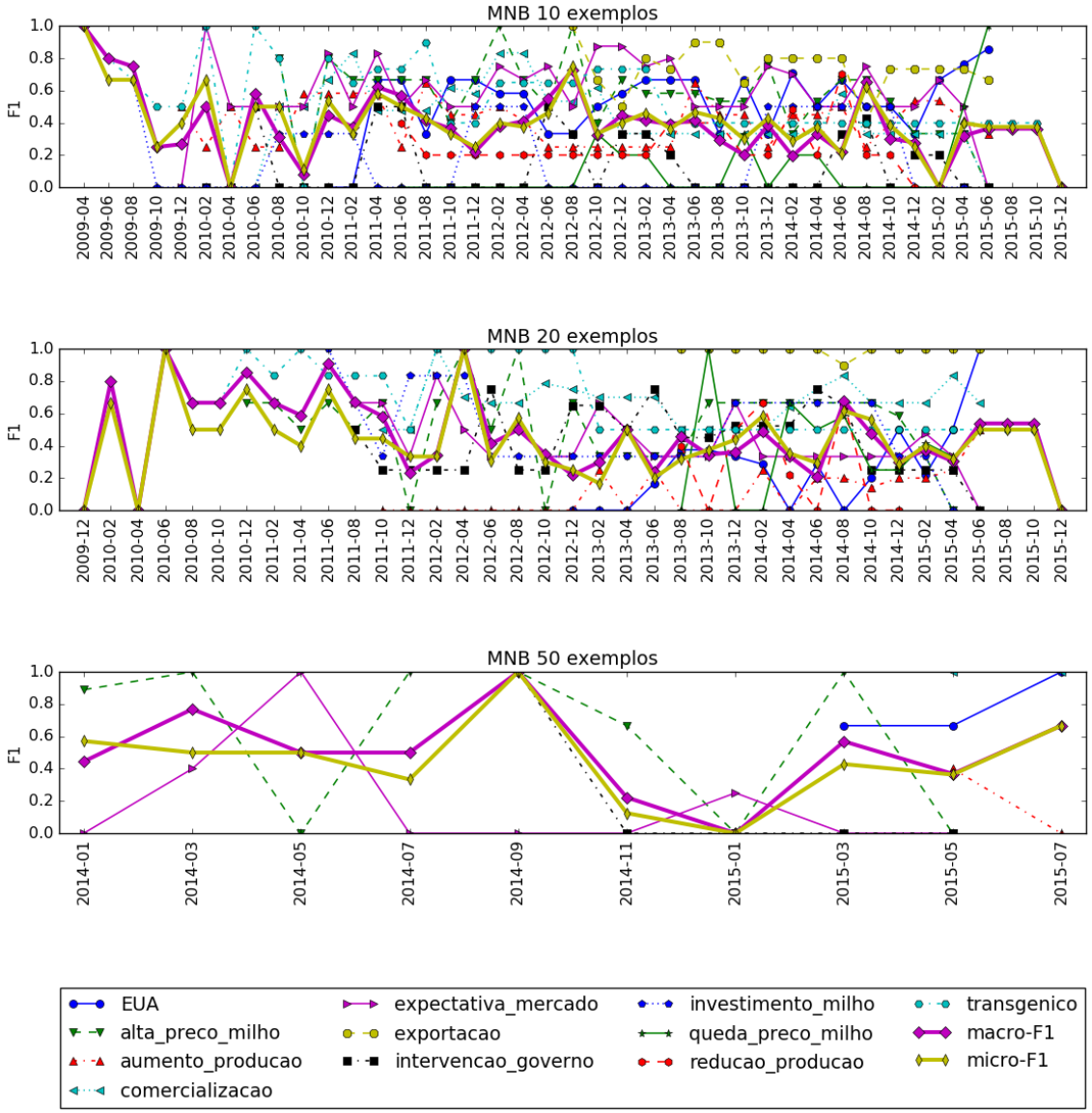

Figura 30. Resultado do algoritmo MNB em um conjunto com 10, 20 e 50 exemplos rotulados. Base: Eventos do Milho, período: Bimestral.

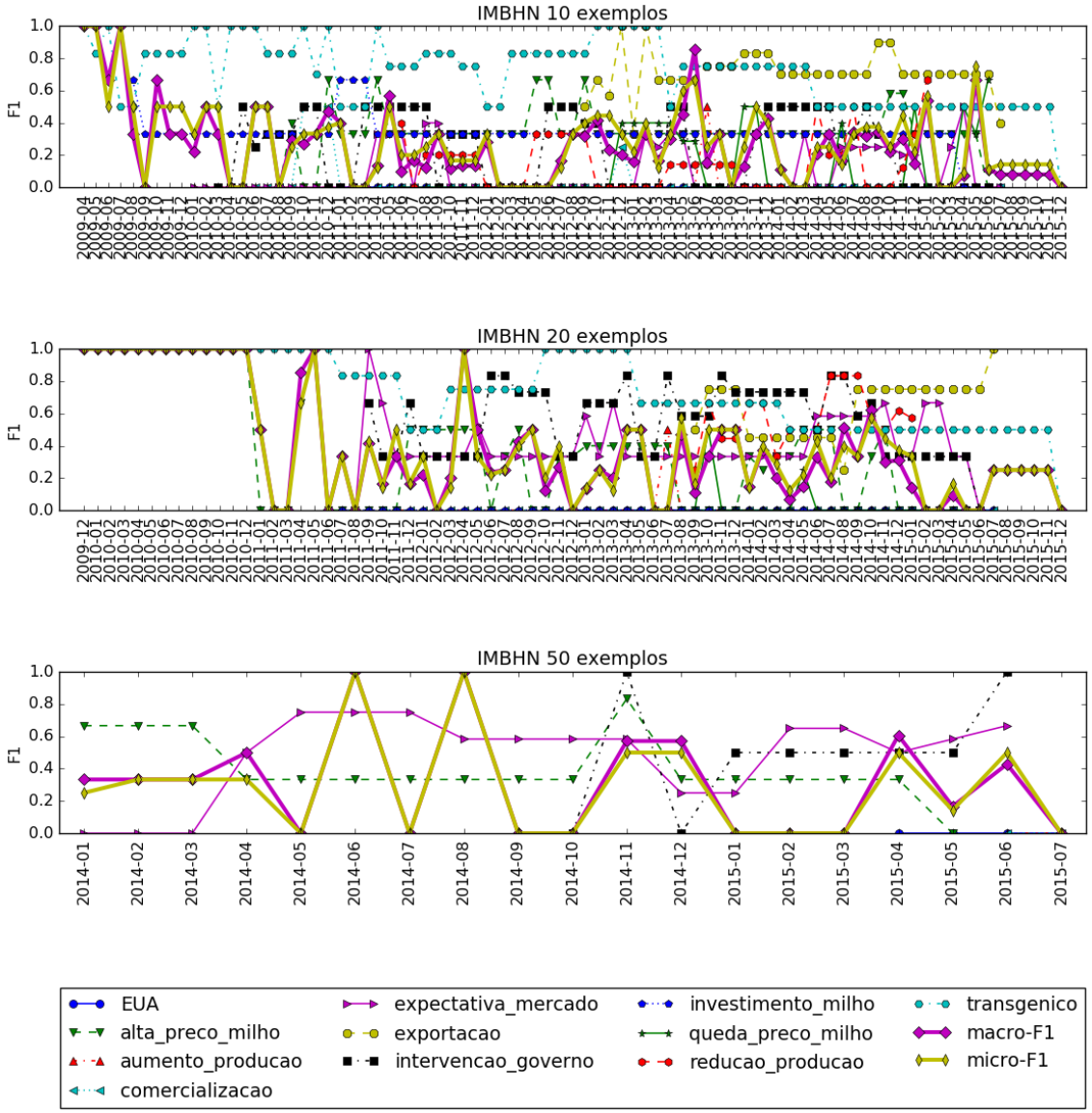

Figura 31. Resultado do algoritmo IMBHN em um conjunto com 10, 20 e 50 exemplos rotulados. Base: Eventos do Milho, período: Mensal.

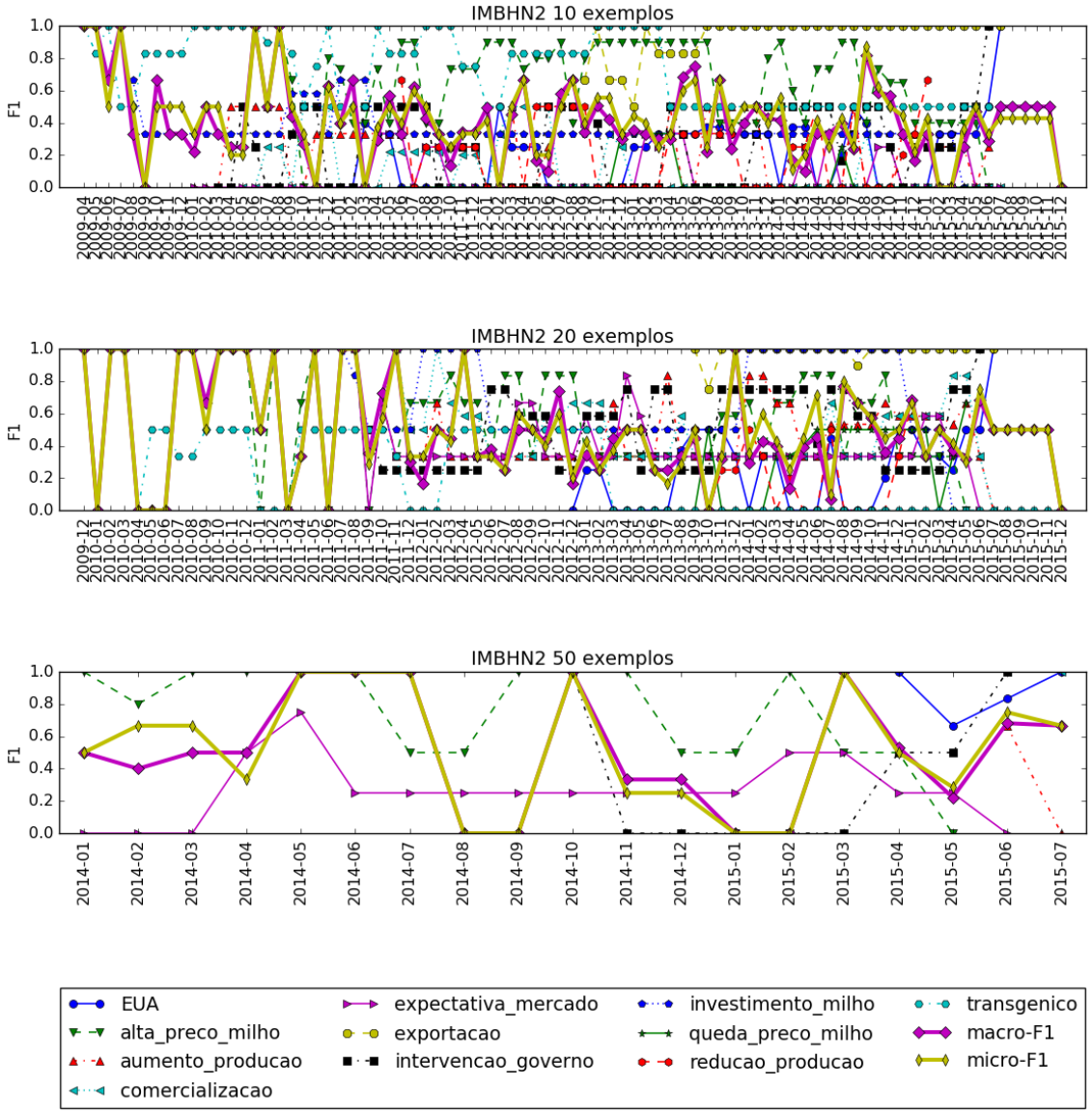

Figura 32. Resultado do algoritmo IMBHN2 em um conjunto com 10, 20 e 50 exemplos rotulados. Base: Eventos do Milho, período: Mensal.

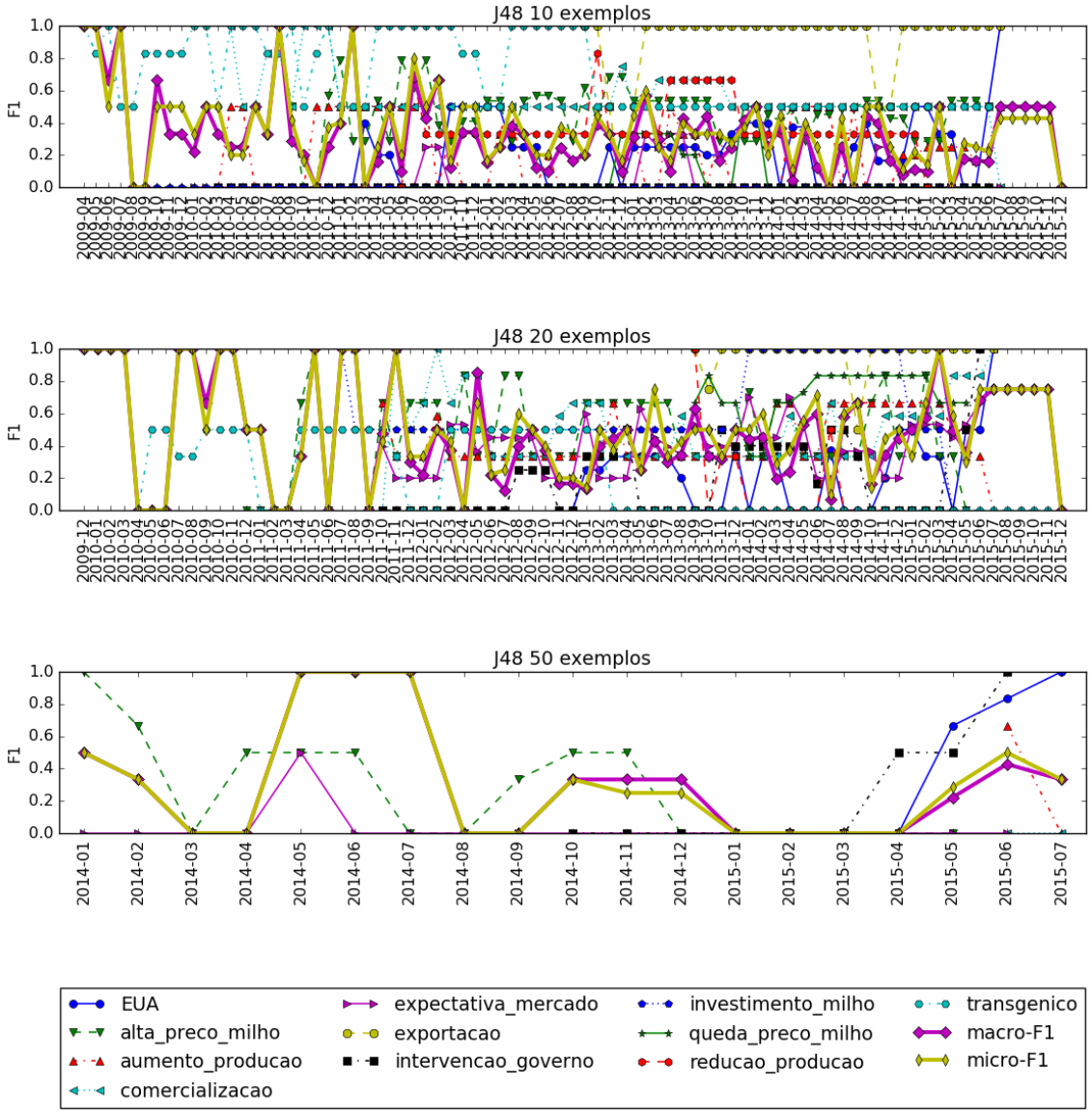

Figura 33. Resultado do algoritmo J48 em um conjunto com 10, 20 e 50 exemplos rotulados. Base: Eventos do Milho, período: Mensal.

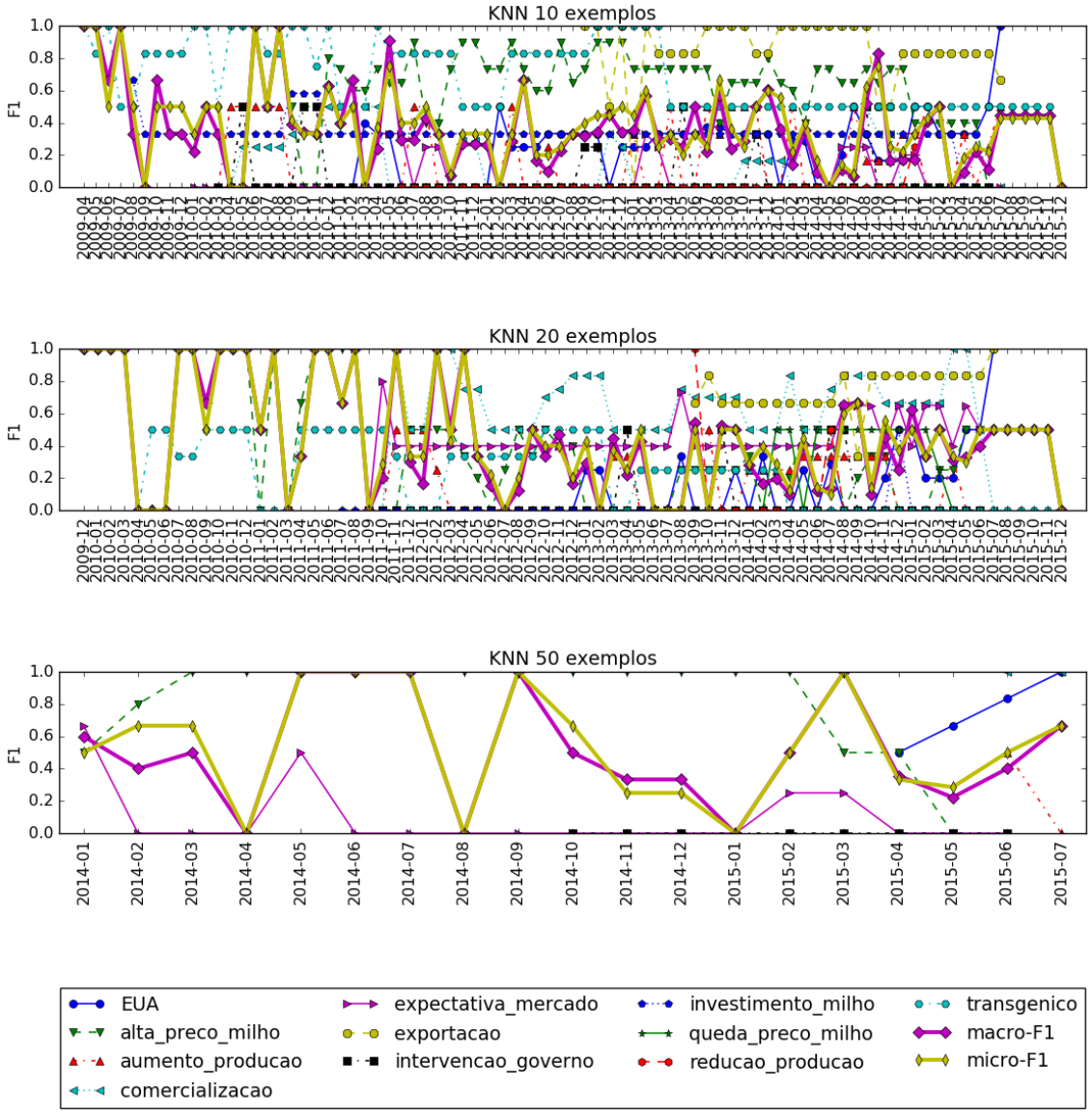

Figura 34. Resultado do algoritmo KNN em um conjunto com 10, 20 e 50 exemplos rotulados. Base: Eventos do Milho, período: Mensal.

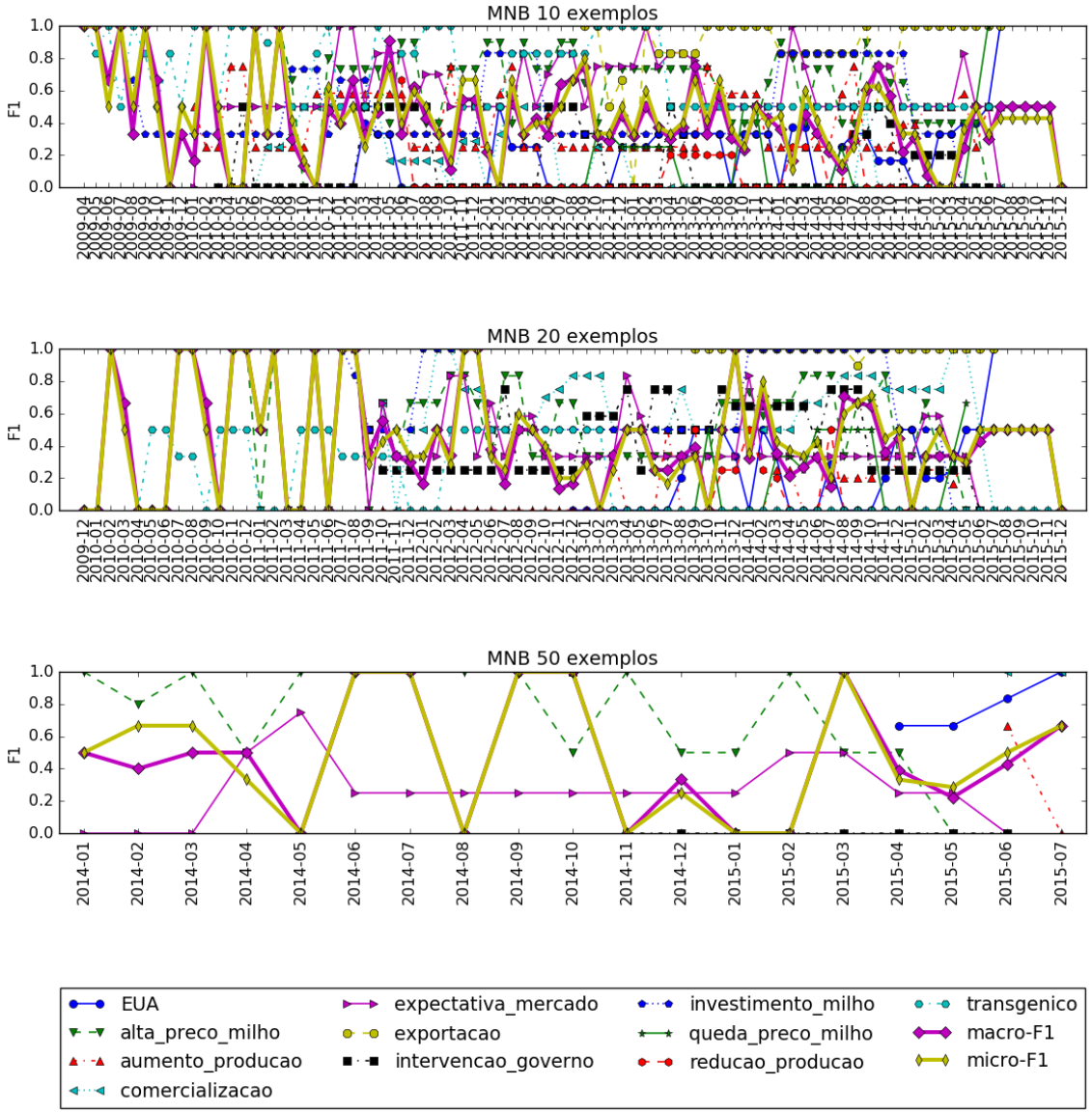

Figura 35. Resultado do algoritmo MNB em um conjunto com 10, 20 e 50 exemplos rotulados. Base: Eventos do Milho, período: Mensal.

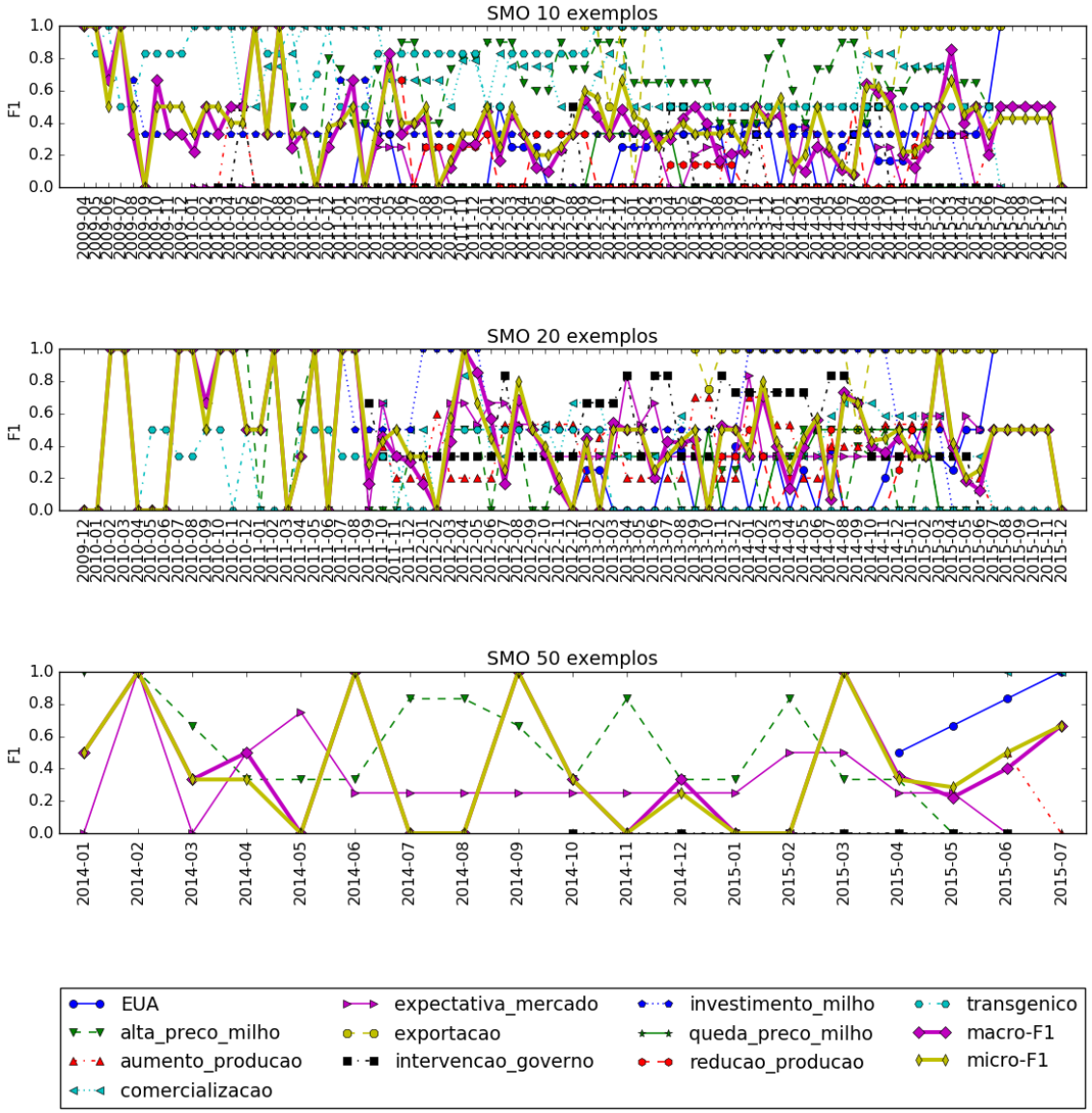

Figura 36. Resultado do algoritmo SMO em um conjunto com 10, 20 e 50 exemplos rotulados. Base: Eventos do Milho, período: Mensal.

4.4.3.2 Resultado: Reuters 21578

Nessa seção são apresentados os resultados obtidos através dos testes executados na base de notícias Reuters 21578. Os experimentos foram executados considerando a configuração experimental apresentada anteriormente. Os resultados são apresentados na sequência de figuras a seguir no intervalo da Figura 37 até a Figura 59. A base Reuters 21578 possui notícias no período de 1987 a 1988, portanto não foi possível realizar experimentos em períodos anuais. No entanto os testes nos períodos: semestral, trimestral, bimestral e mensal puderam ser executados normalmente.

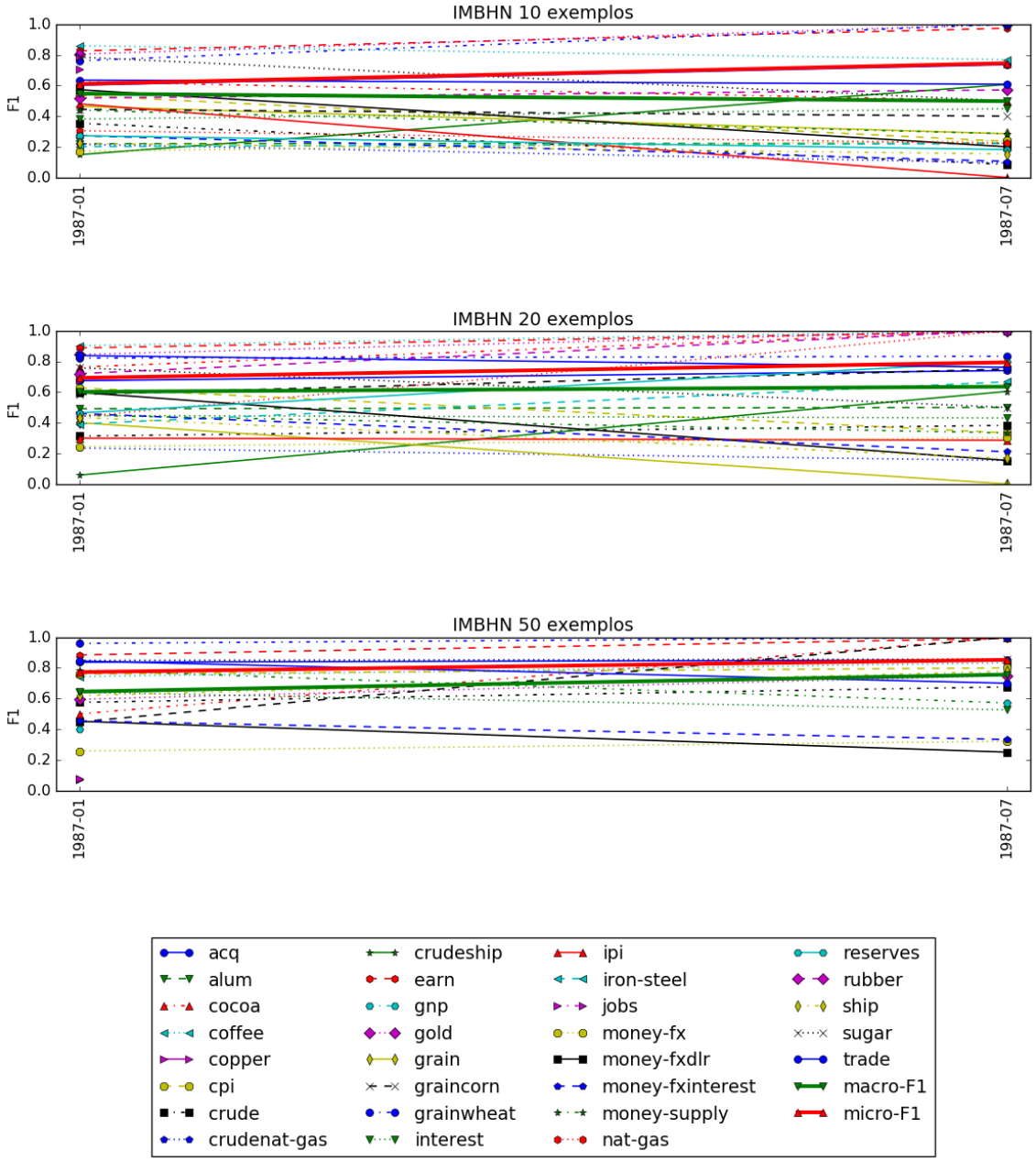

Figura 37. Resultado do algoritmo IMBHN em um conjunto com 10, 20 e 50 exemplos rotulados. Base: Reuters 21578, período: Semestral.

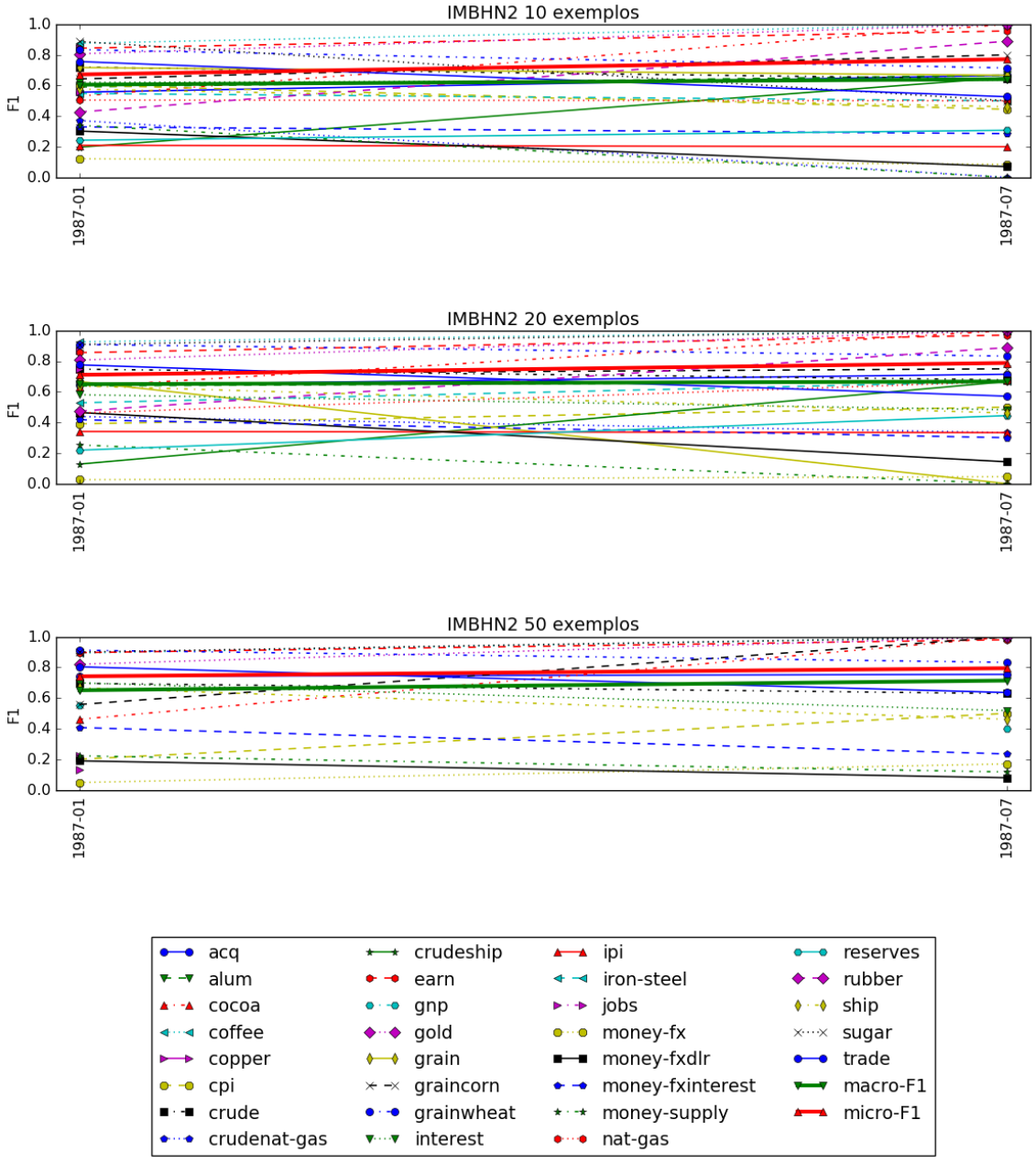

Figura 38. Resultado do algoritmo IMBHN2 em um conjunto com 10, 20 e 50 exemplos rotulados. Base: Reuters 21578, período: Semestral.

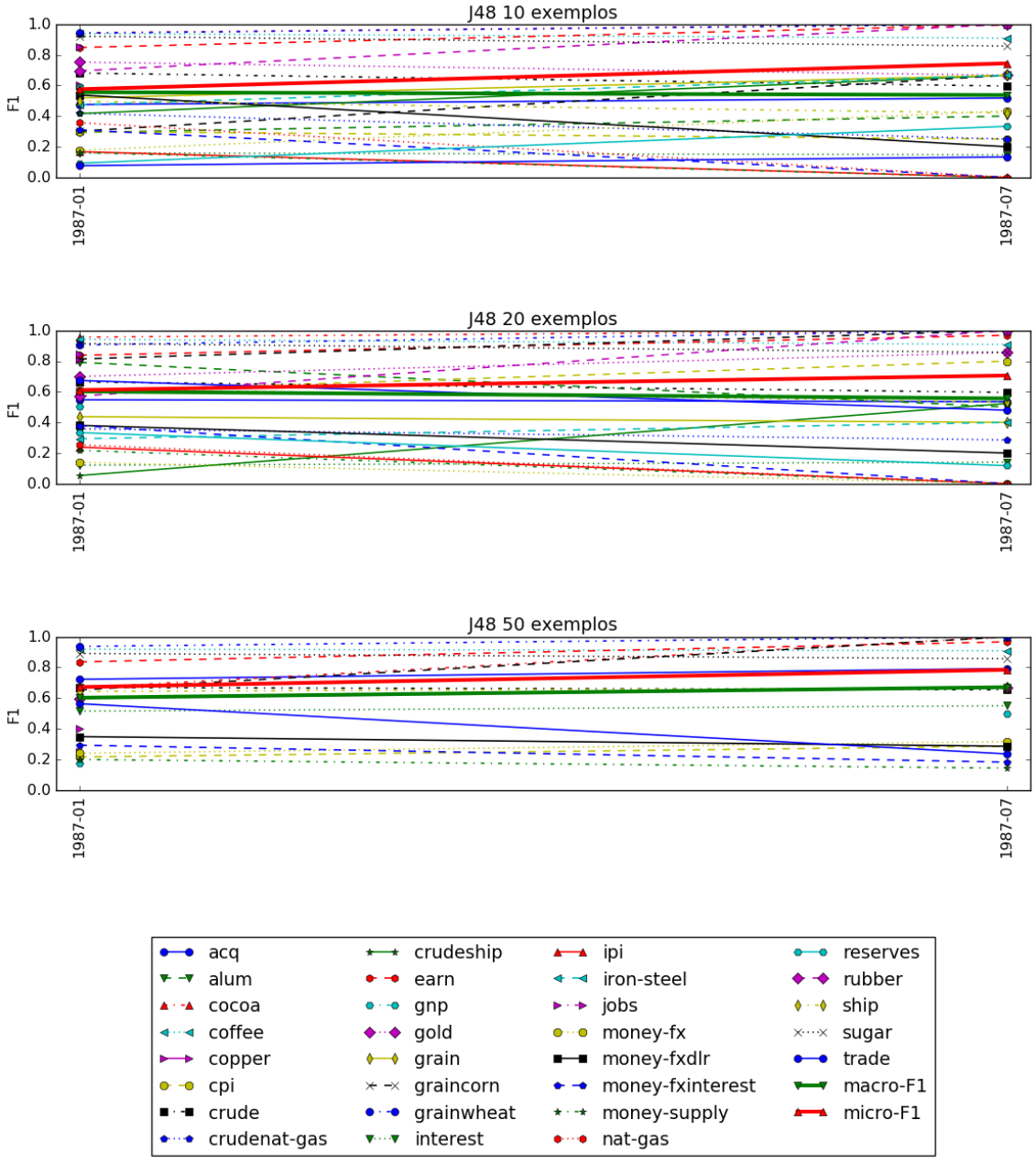

Figura 39. Resultado do algoritmo J48 em um conjunto com 10, 20 e 50 exemplos rotulados. Base: Reuters 21578, período: Semestral.

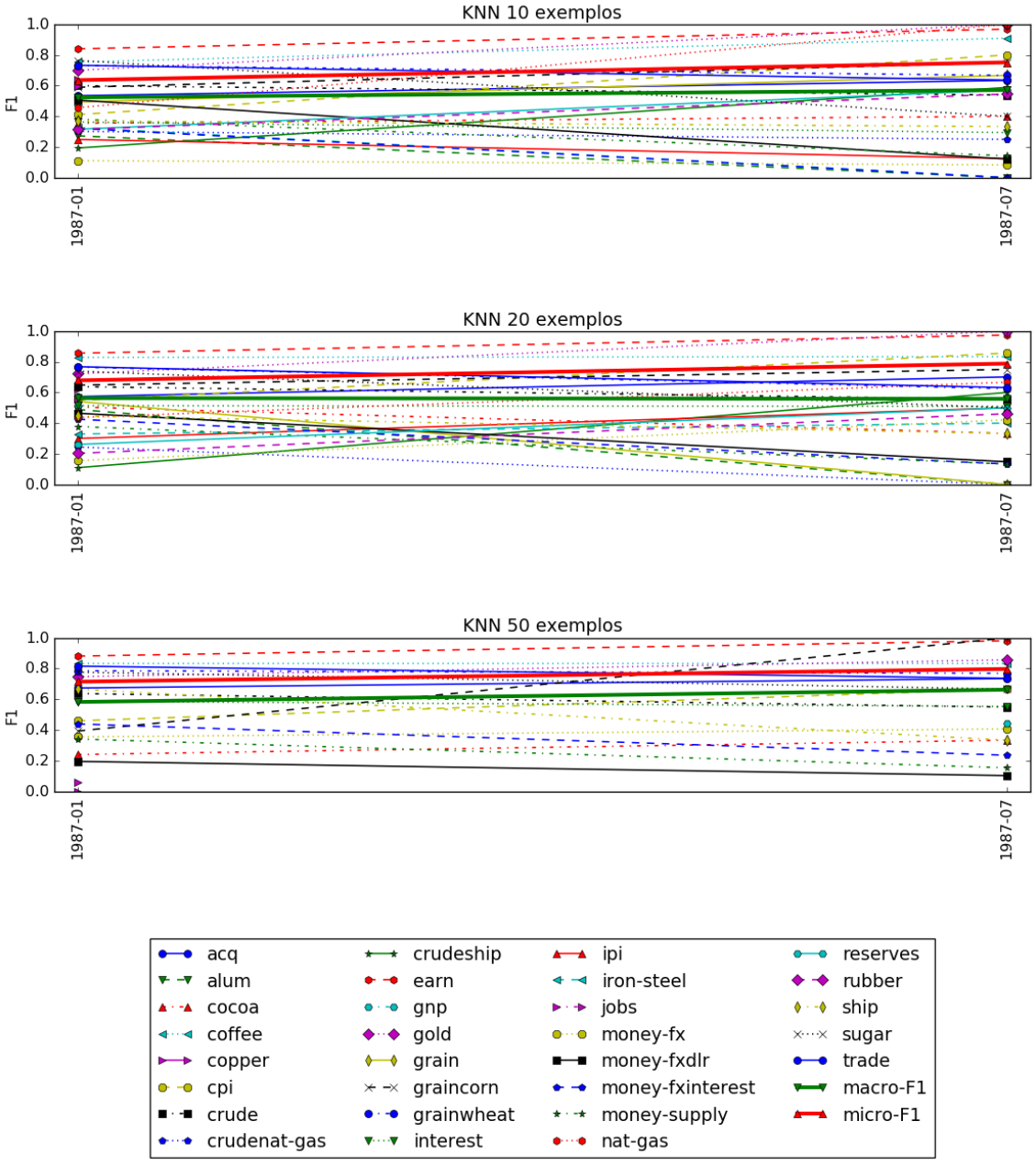

Figura 40. Resultado do algoritmo KNN em um conjunto com 10, 20 e 50 exemplos rotulados. Base: Reuters 21578, período: Semestral.

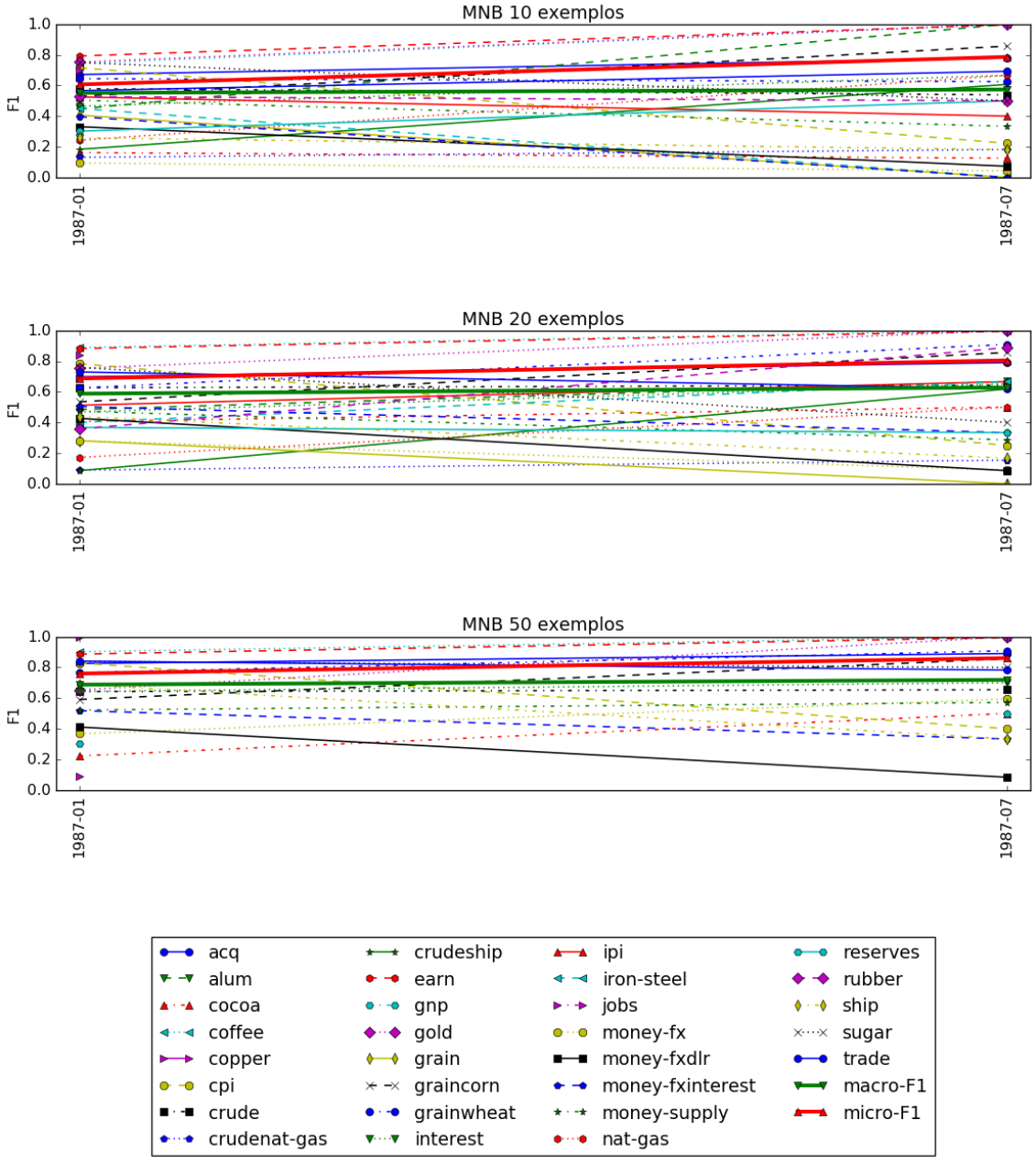

Figura 41. Resultado do algoritmo MNB em um conjunto com 10, 20 e 50 exemplos rotulados. Base: Reuters 21578, período: Semestral.

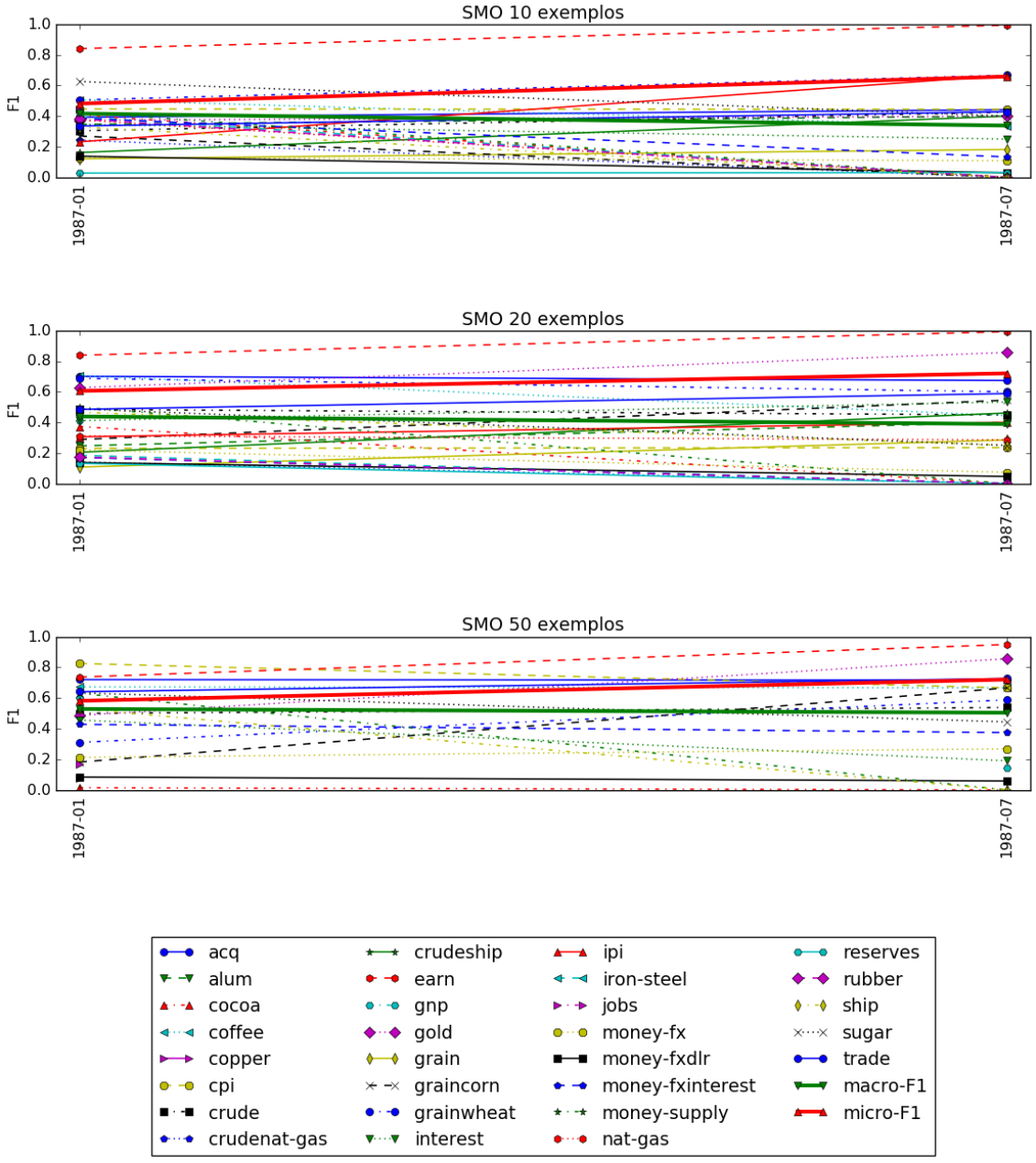

Figura 42. Resultado do algoritmo SMO em um conjunto com 10, 20 e 50 exemplos rotulados. Base: Reuters 21578, período: Semestral.

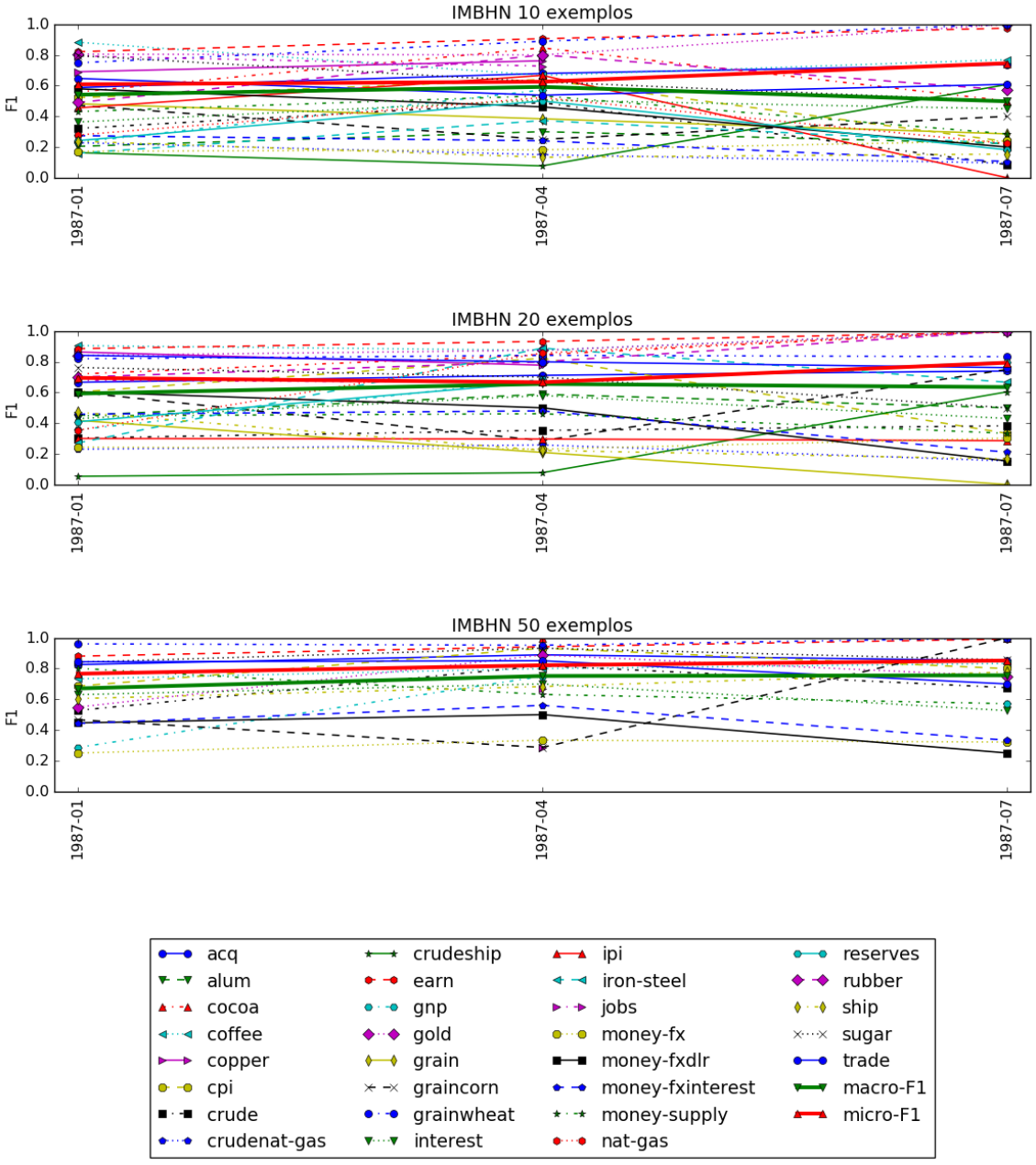

Figura 43. Resultado do algoritmo IMBHN em um conjunto com 10, 20 e 50 exemplos rotulados. Base: Reuters 21578, período: Trimestral.

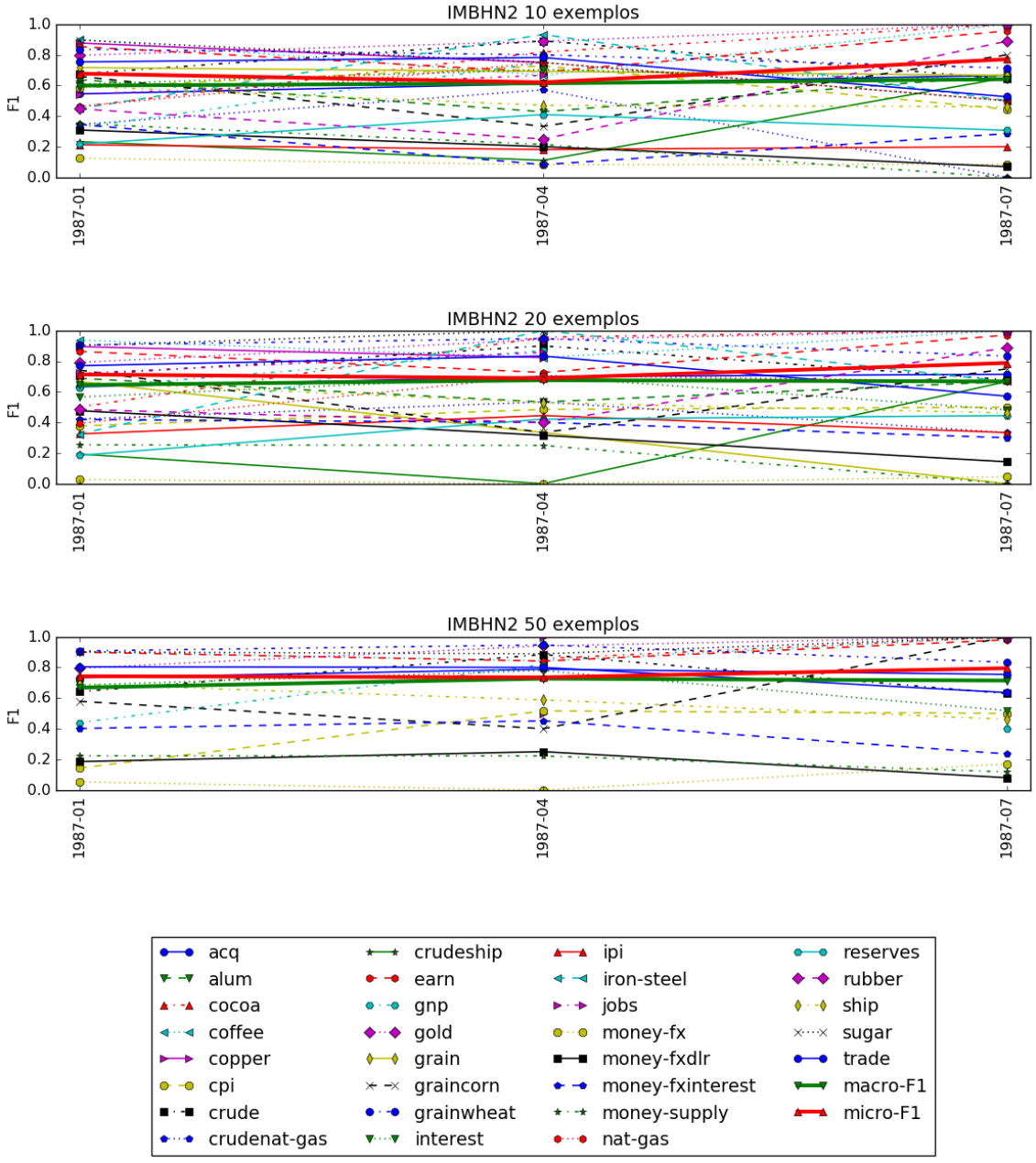

Figura 44. Resultado do algoritmo IMBHN2 em um conjunto com 10, 20 e 50 exemplos rotulados. Base: Reuters 21578, período: Trimestral.

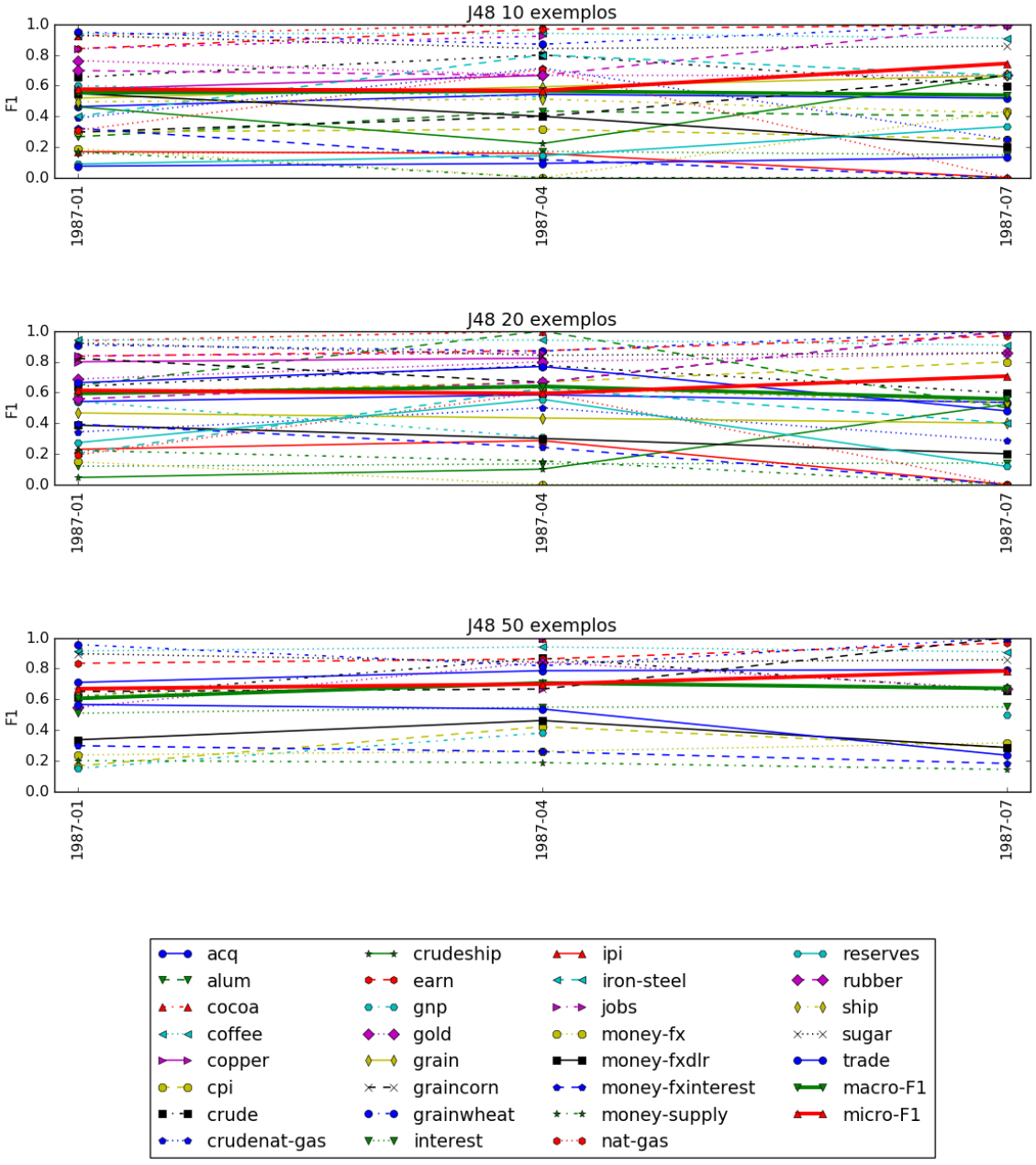

Figura 45. Resultado do algoritmo J48 em um conjunto com 10, 20 e 50 exemplos rotulados. Base: Reuters 21578, período: Trimestral.

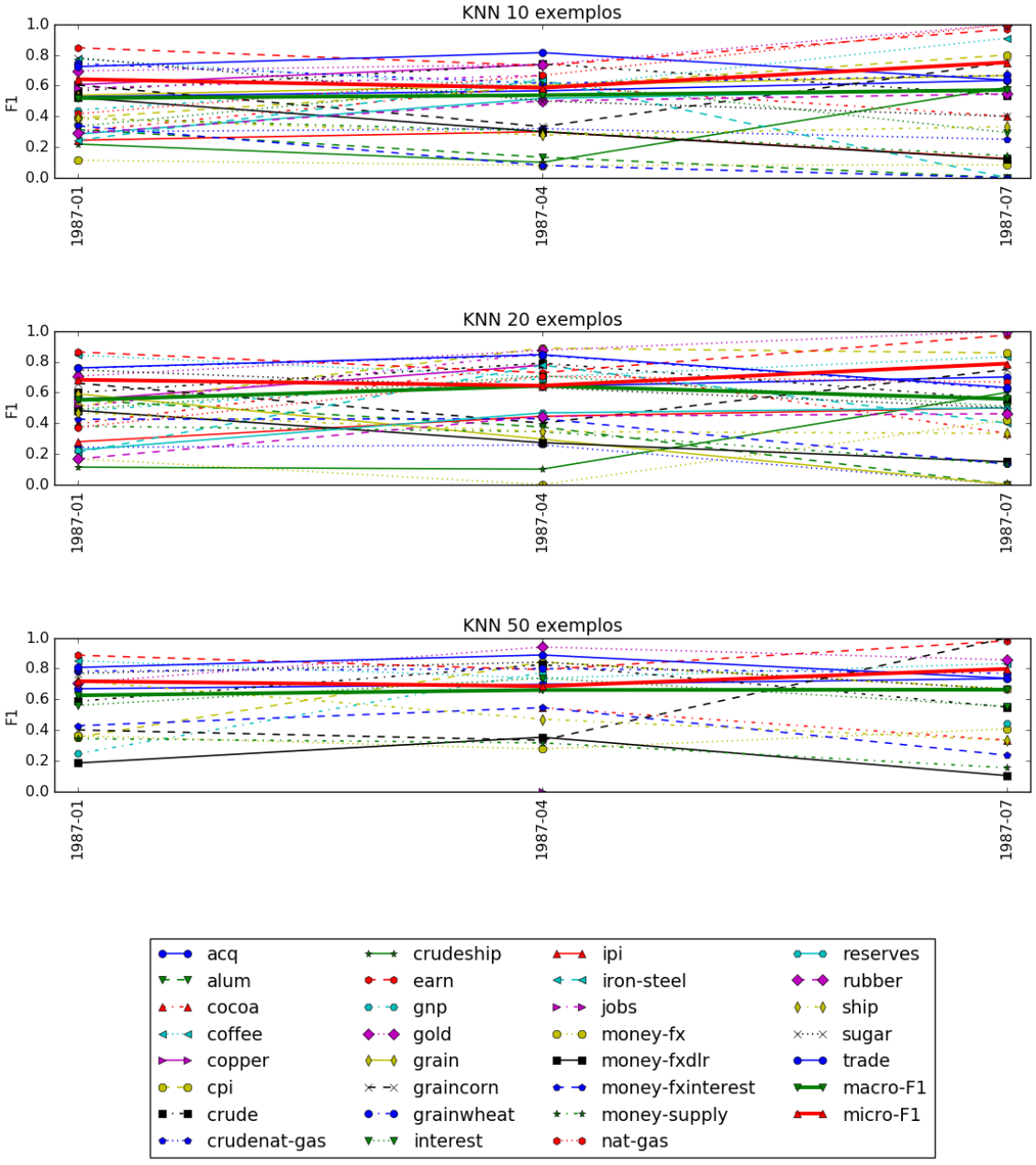

Figura 46. Resultado do algoritmo KNN em um conjunto com 10, 20 e 50 exemplos rotulados. Base: Reuters 21578, período: Trimestral.

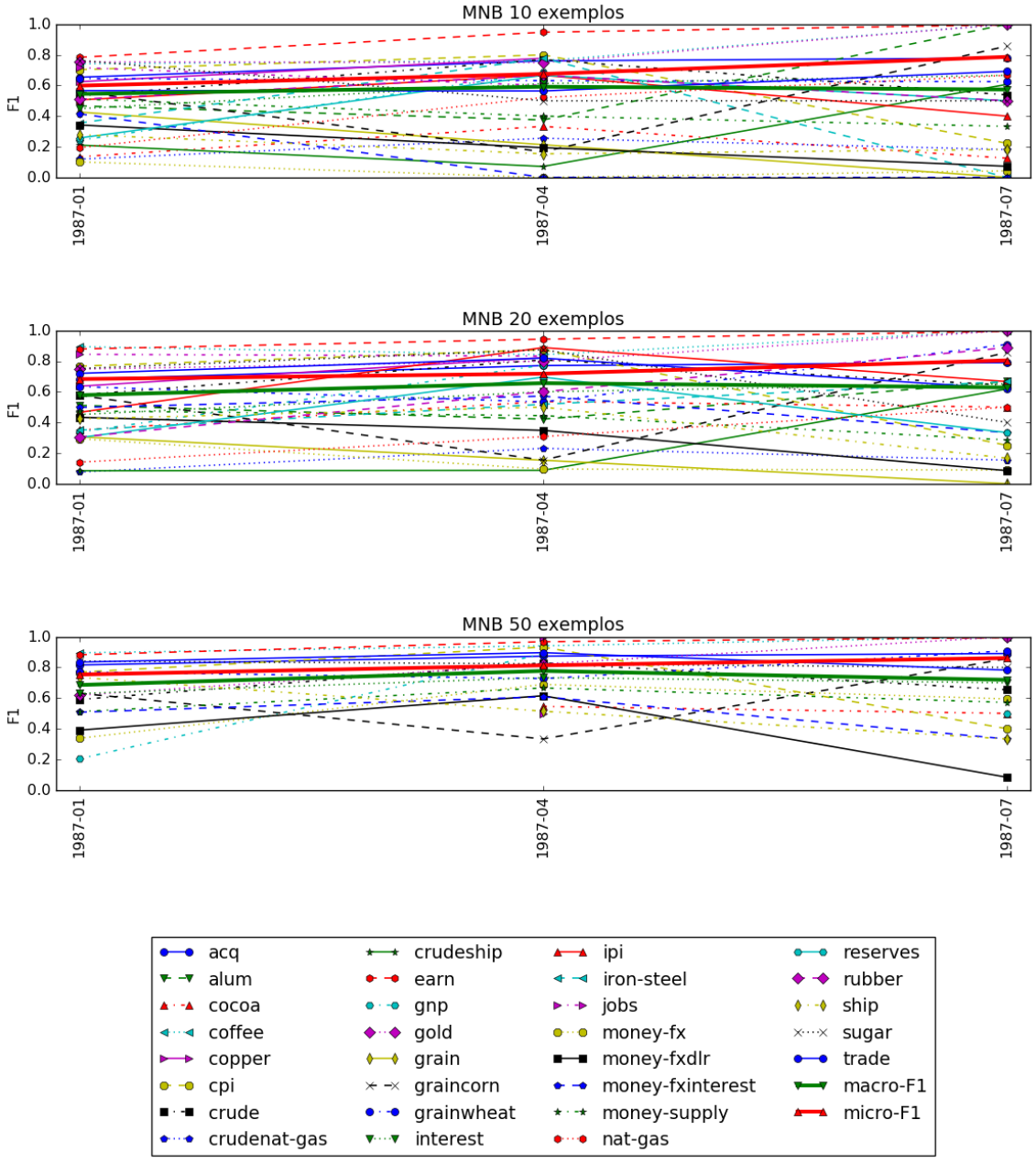

Figura 47. Resultado do algoritmo MNB em um conjunto com 10, 20 e 50 exemplos rotulados. Base: Reuters 21578, período: Trimestral.

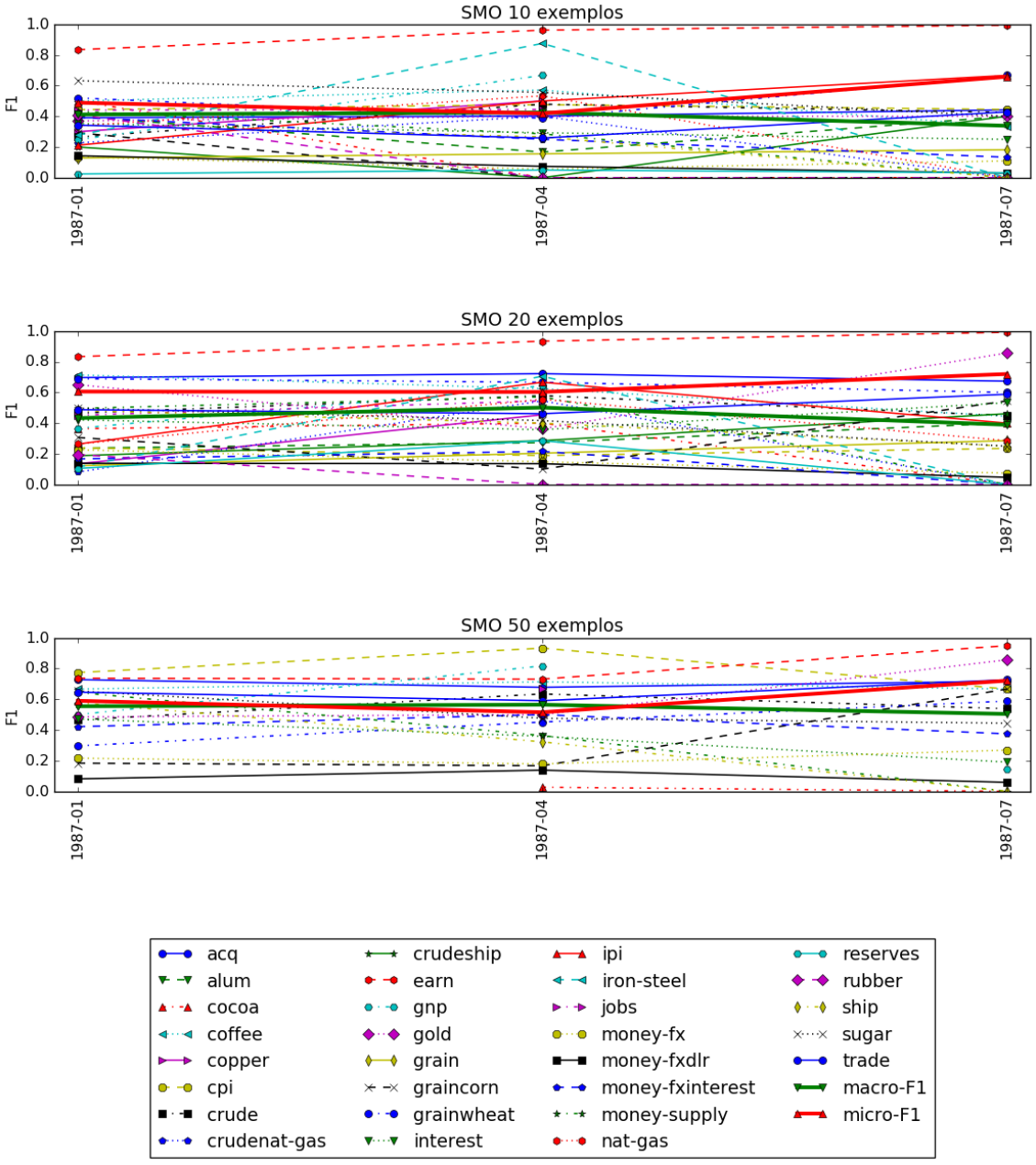

Figura 48. Resultado do algoritmo SMO em um conjunto com 10, 20 e 50 exemplos rotulados. Base: Reuters 21578, período: Trimestral.

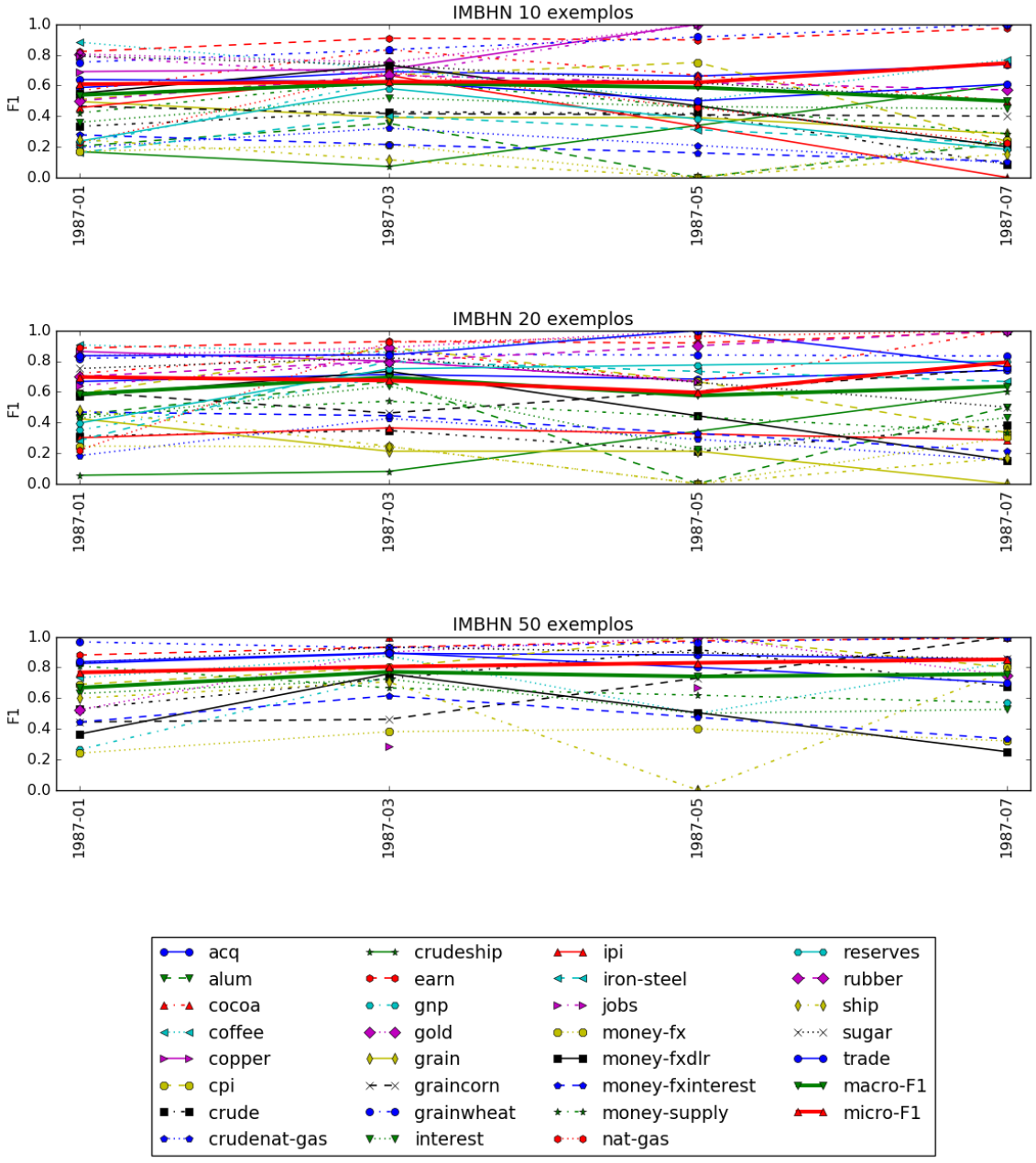

Figura 49. Resultado do algoritmo IMBHN em um conjunto com 10, 20 e 50 exemplos rotulados. Base: Reuters 21578, período: Bimestral.

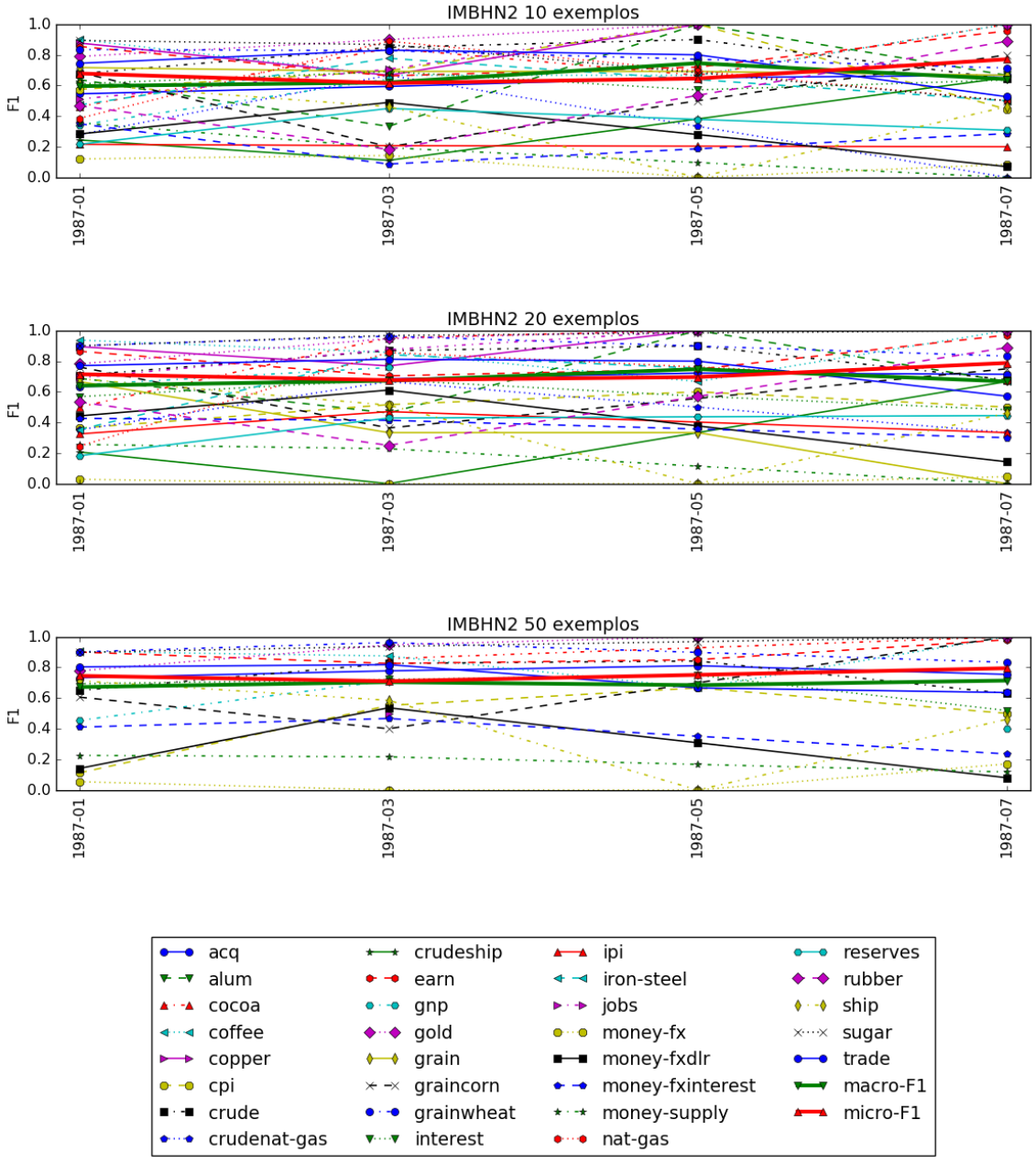

Figura 50. Resultado do algoritmo IMBHN2 em um conjunto com 10, 20 e 50 exemplos rotulados. Base: Reuters 21578, período: Bimestral.

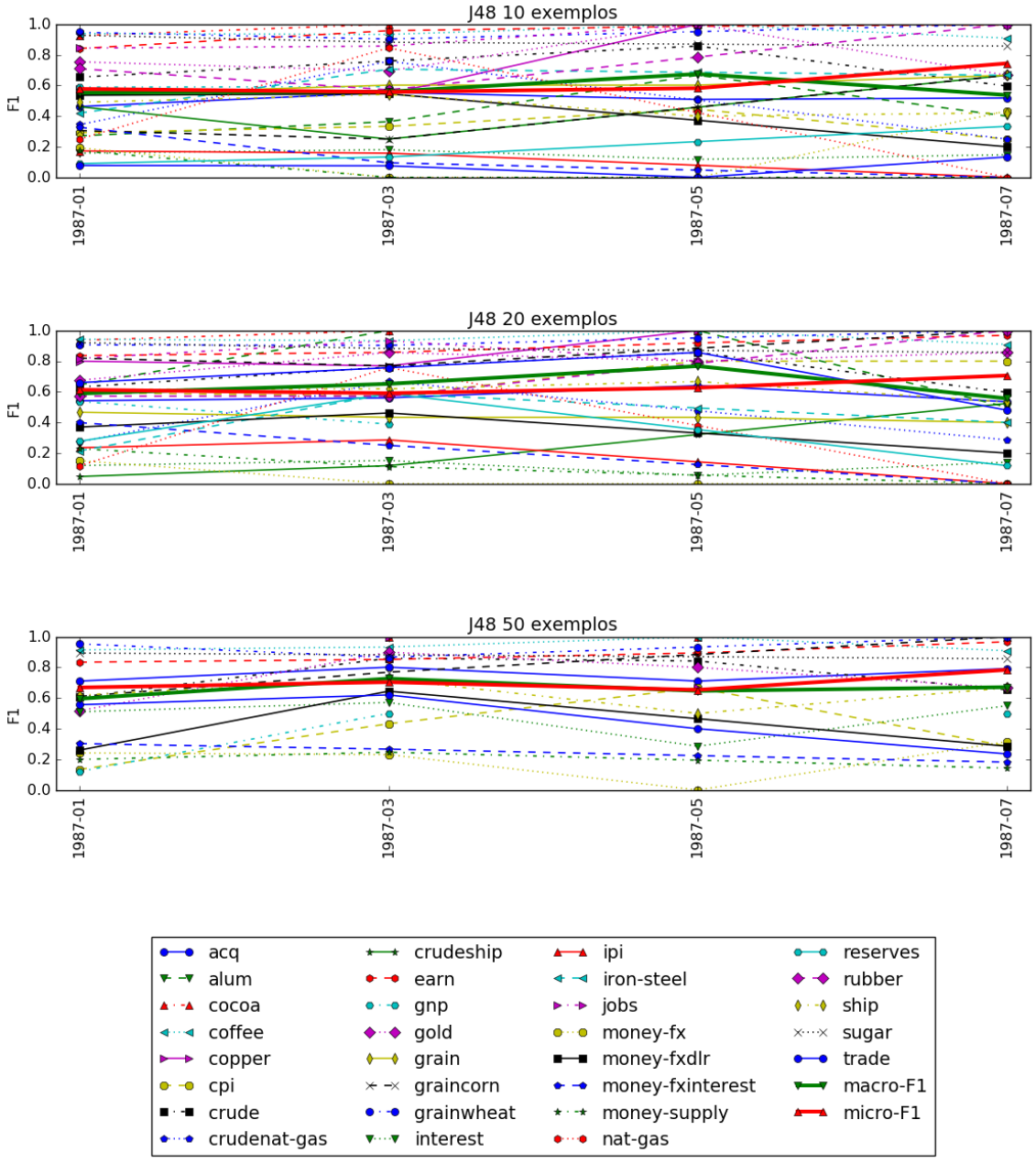

Figura 51. Resultado do algoritmo J48 em um conjunto com 10, 20 e 50 exemplos rotulados. Base: Reuters 21578, período: Bimestral.

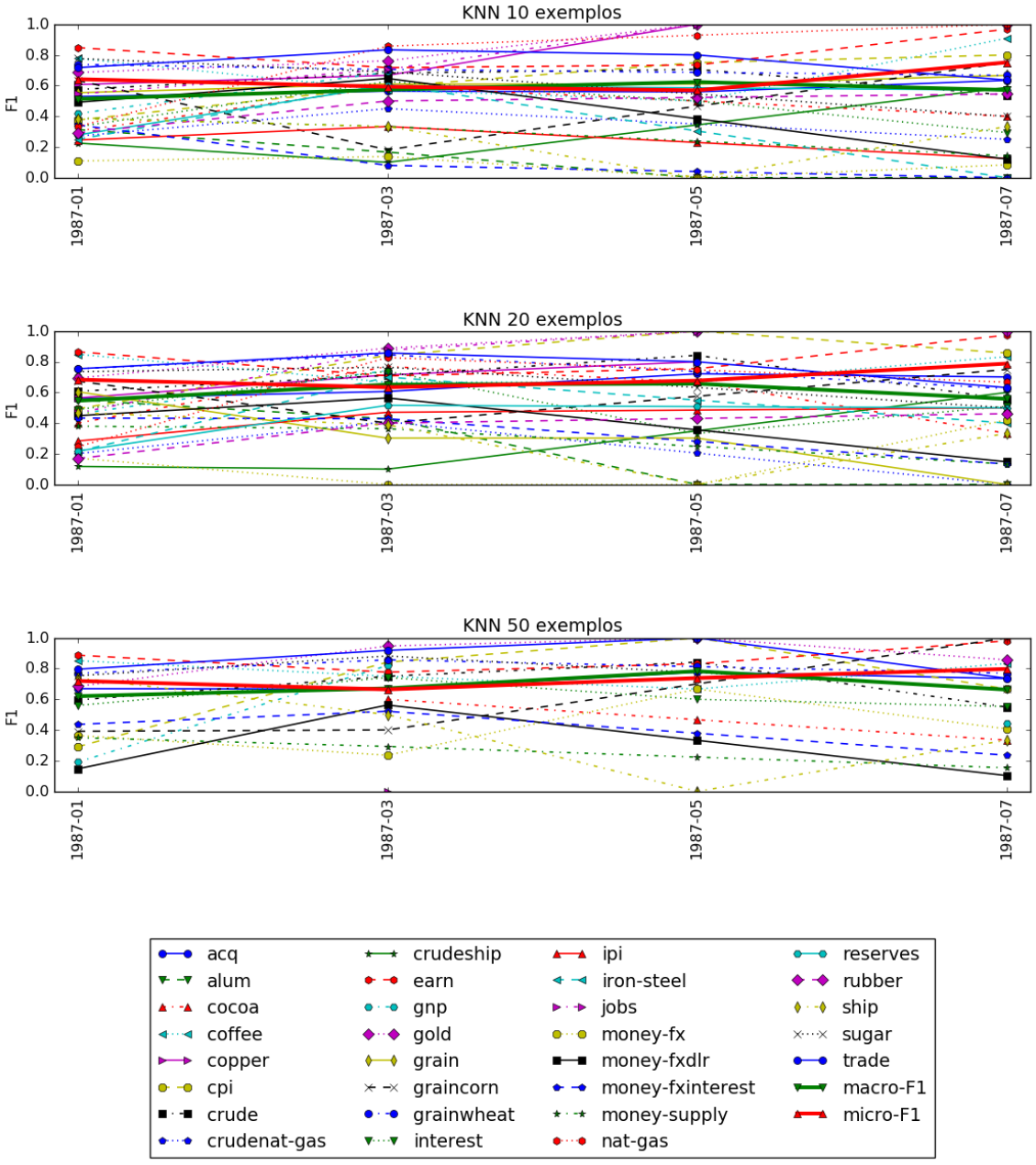

Figura 52. Resultado do algoritmo KNN em um conjunto com 10, 20 e 50 exemplos rotulados. Base: Reuters 21578, período: Bimestral.

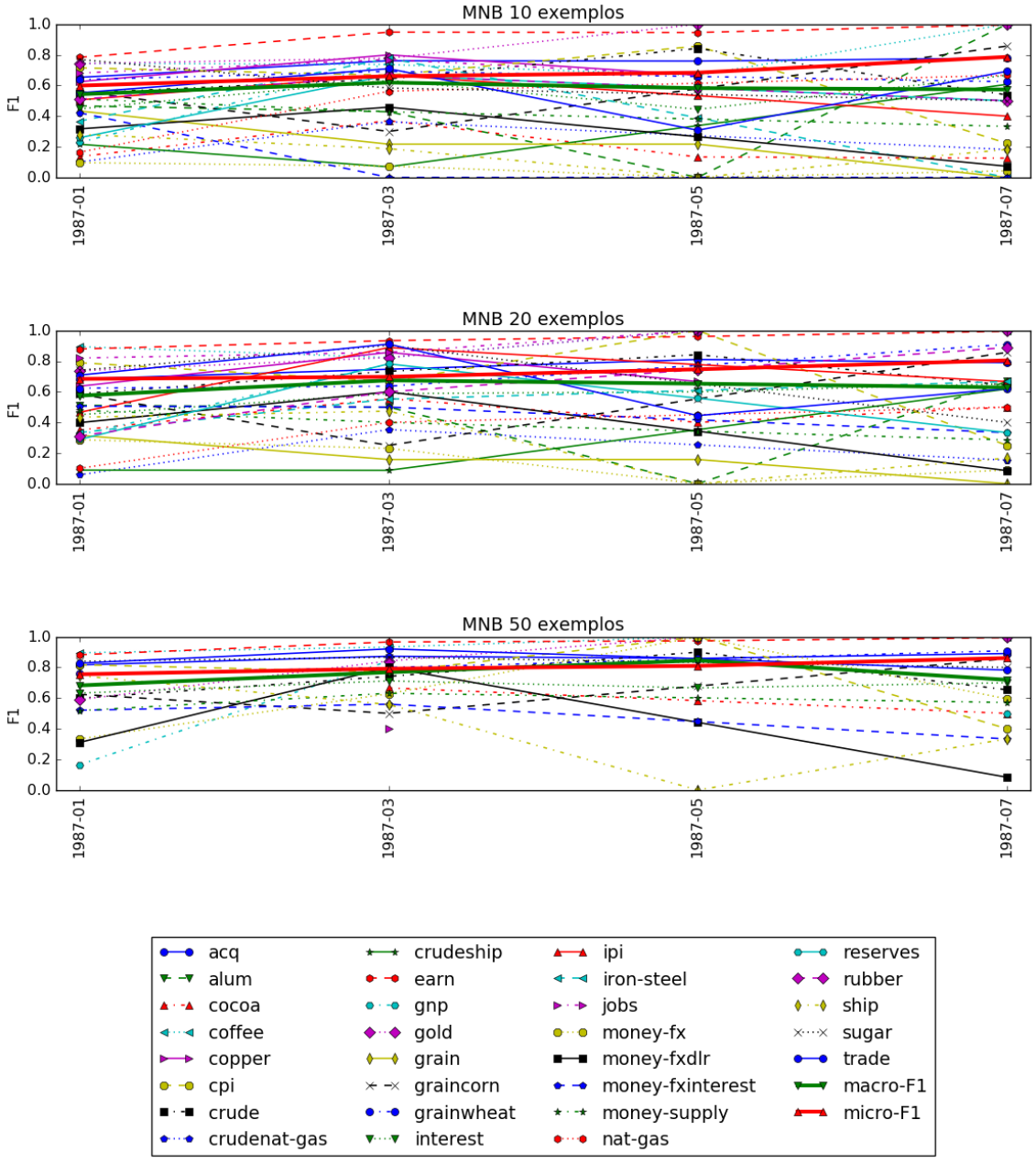

Figura 53. Resultado do algoritmo MNB em um conjunto com 10, 20 e 50 exemplos rotulados. Base: Reuters 21578, período: Bimestral.

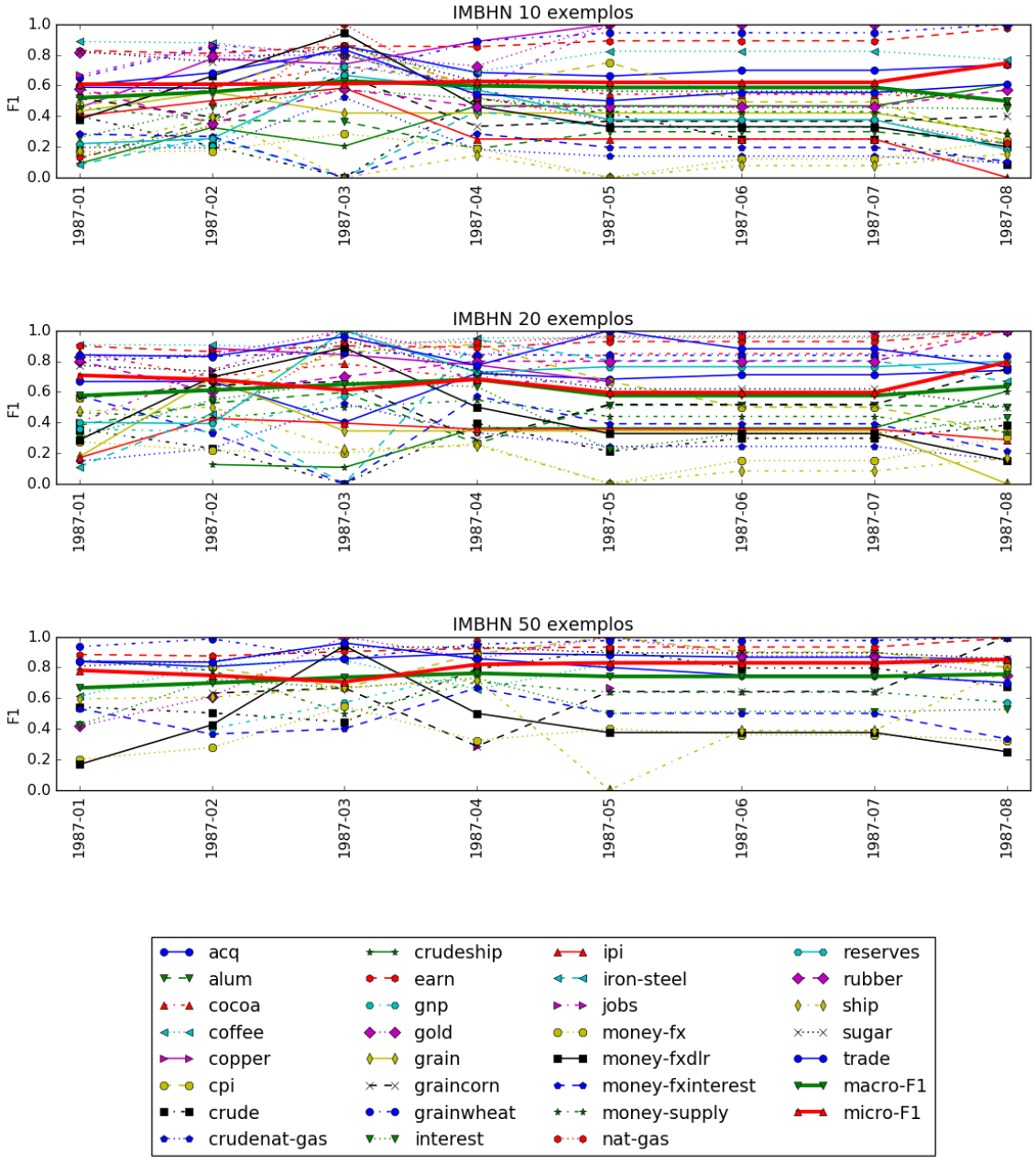

Figura 54. Resultado do algoritmo IMBHN em um conjunto com 10, 20 e 50 exemplos rotulados. Base: Reuters 21578, período: Mensal.

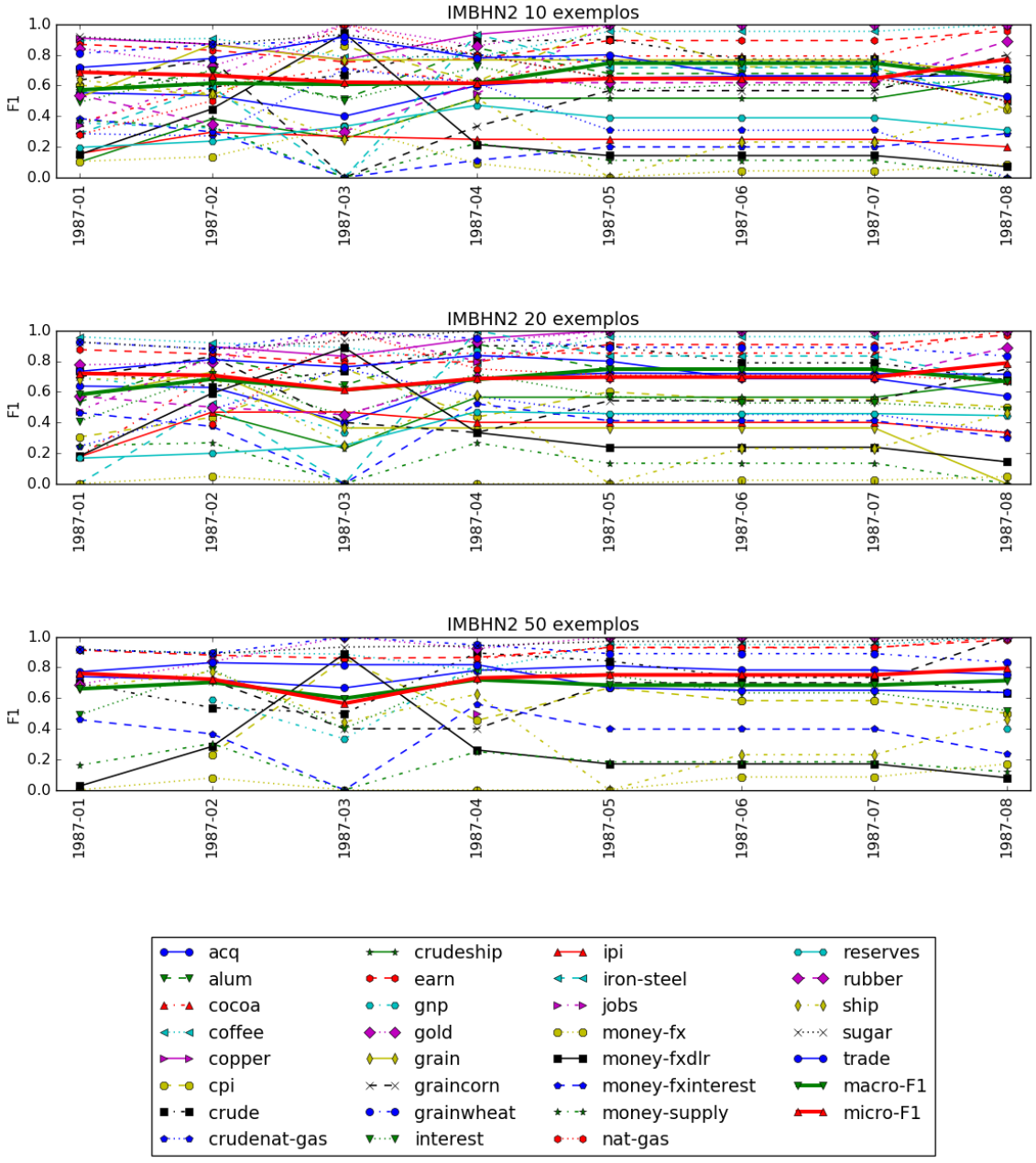

Figura 55. Resultado do algoritmo IMBHN2 em um conjunto com 10, 20 e 50 exemplos rotulados. Base: Reuters 21578, período: Mensal.

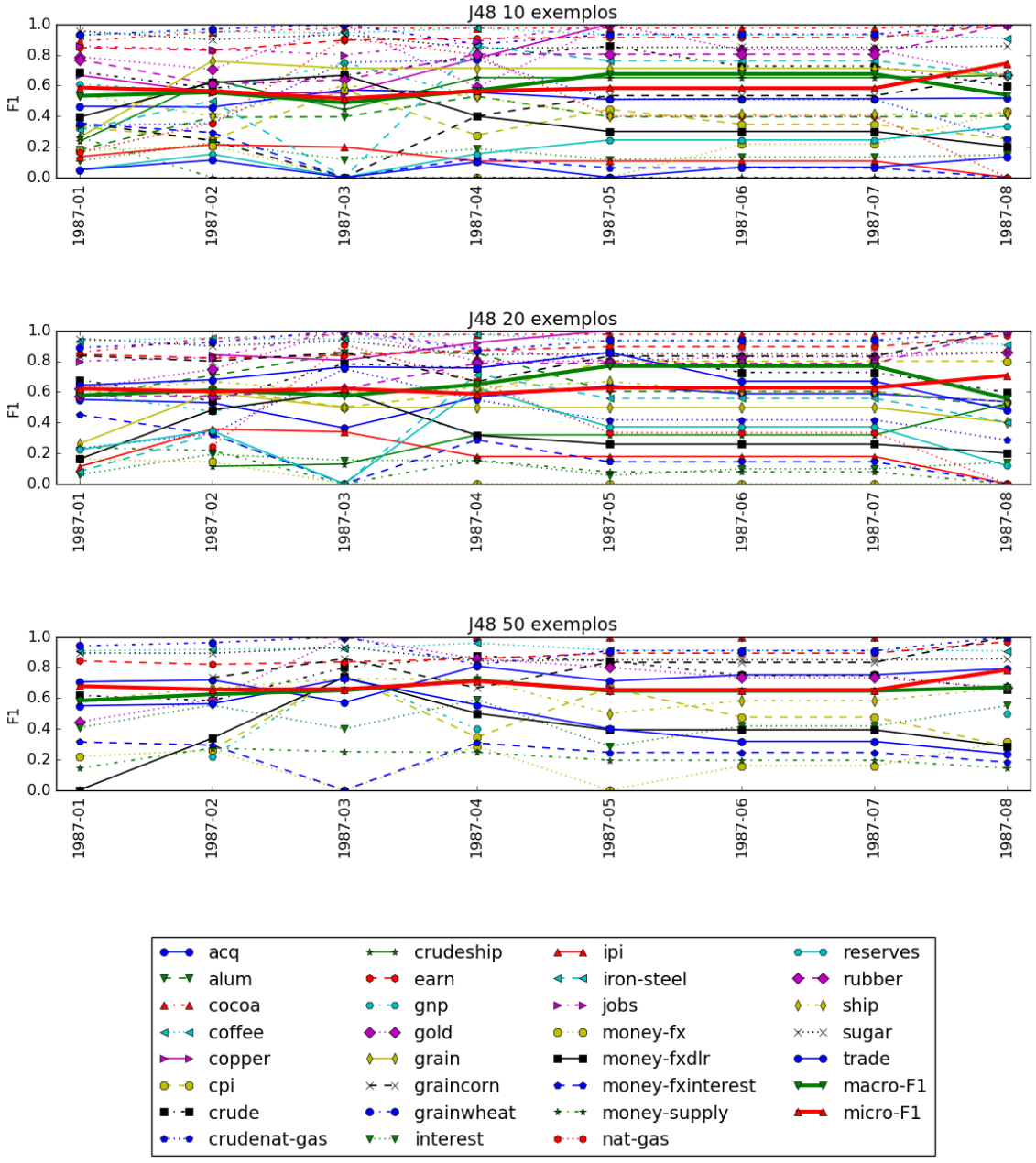

Figura 56. Resultado do algoritmo J48 em um conjunto com 10, 20 e 50 exemplos rotulados. Base: Reuters 21578, período: Mensal.

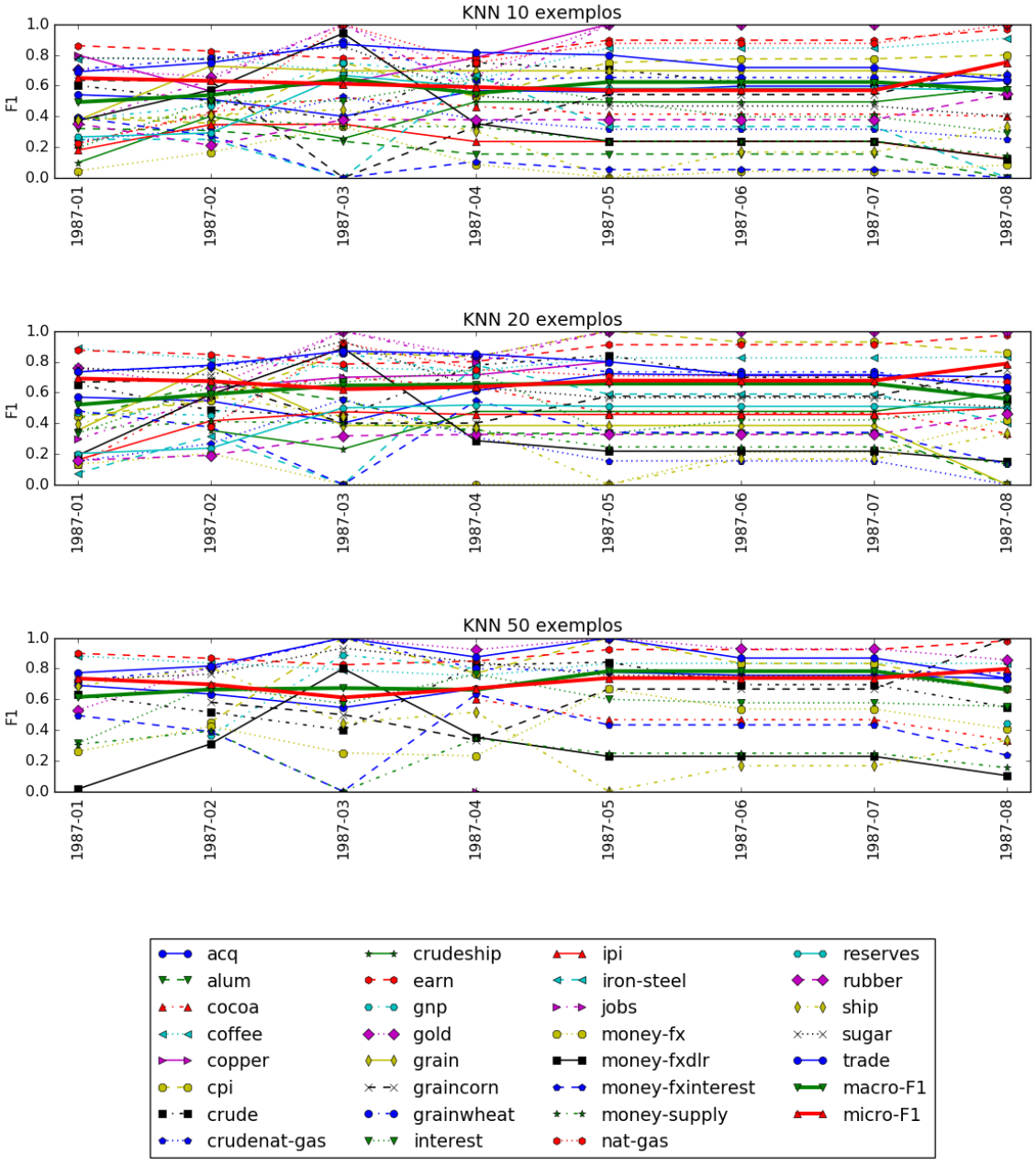

Figura 57. Resultado do algoritmo KNN em um conjunto com 10, 20 e 50 exemplos rotulados. Base: Reuters 21578, período: Mensal.

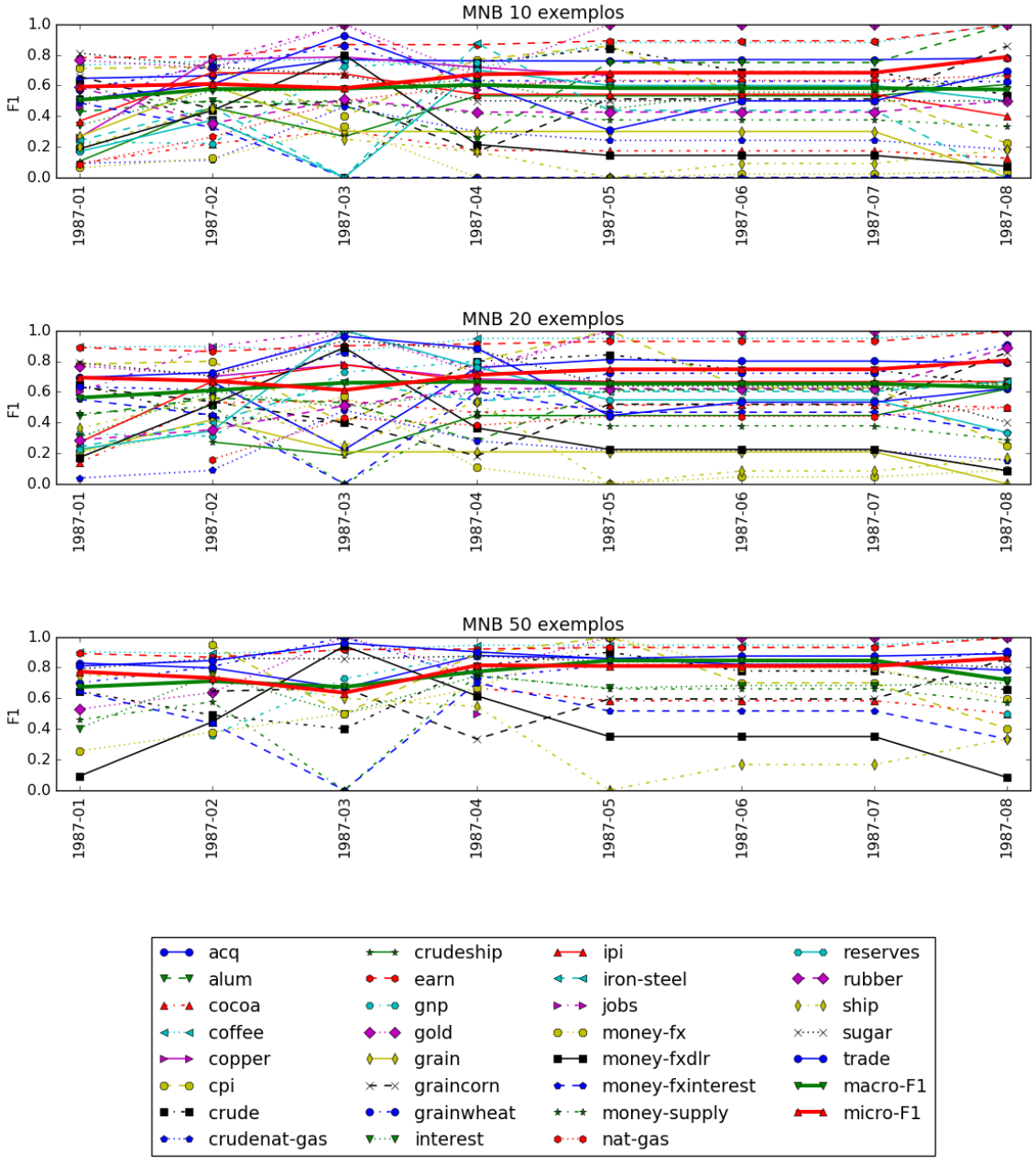

Figura 58. Resultado do algoritmo MNB em um conjunto com 10, 20 e 50 exemplos rotulados. Base: Reuters 21578, período: Mensal.

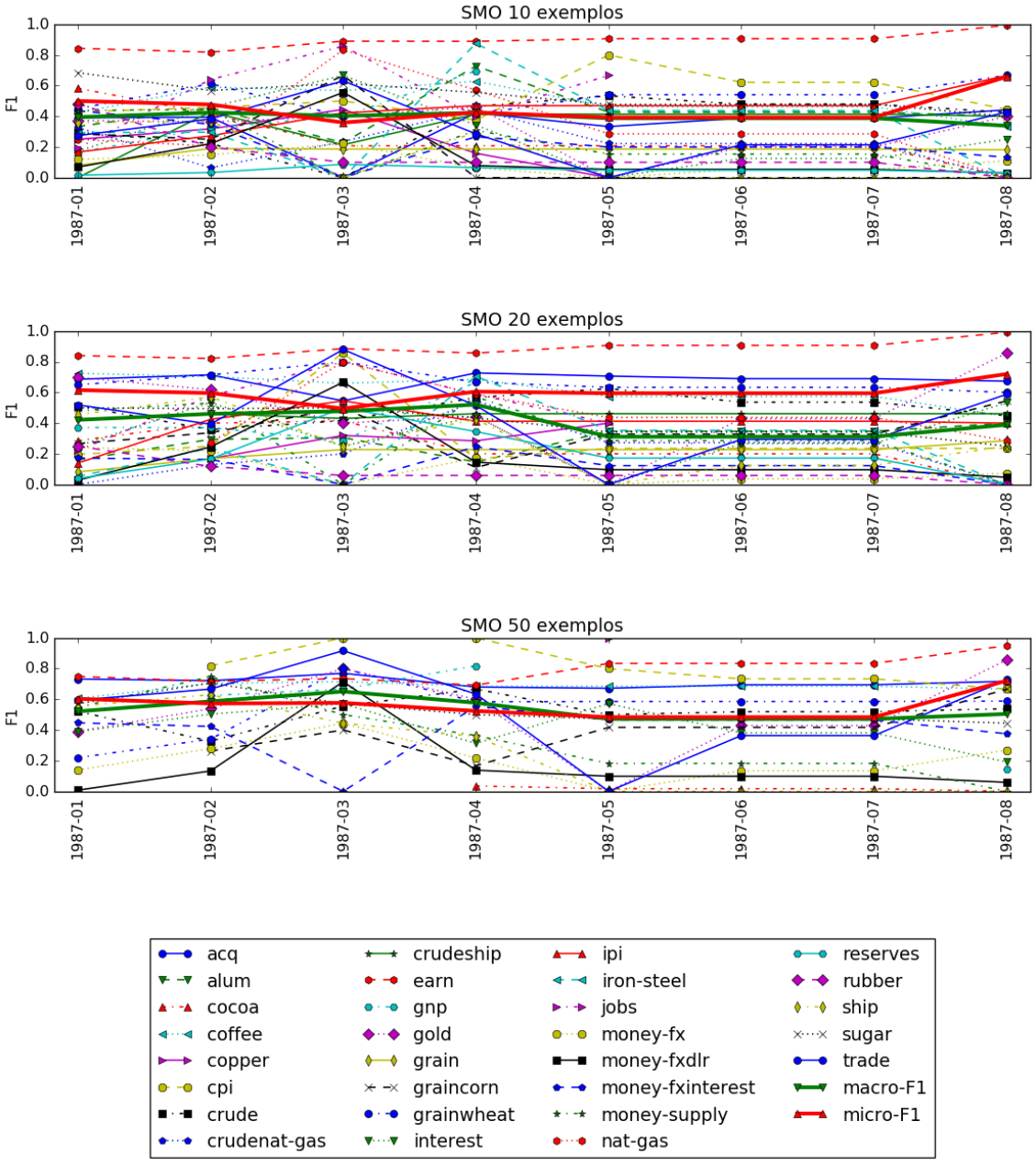

Figura 59. Resultado do algoritmo SMO em um conjunto com 10, 20 e 50 exemplos rotulados. Base: Reuters 21578, período: Mensal.

4.4.3.3 Resultado: Base Notícias do Brasil (Variedades)

Nessa seção são apresentados os resultados obtidos através dos testes executados na base de Notícias do Brasil (Variedades). Os experimentos foram executados considerando a configuração experimental apresentada anteriormente. Os resultados são apresentados na sequência de figuras a seguir no intervalo da Figura 60 até a Figura 88.

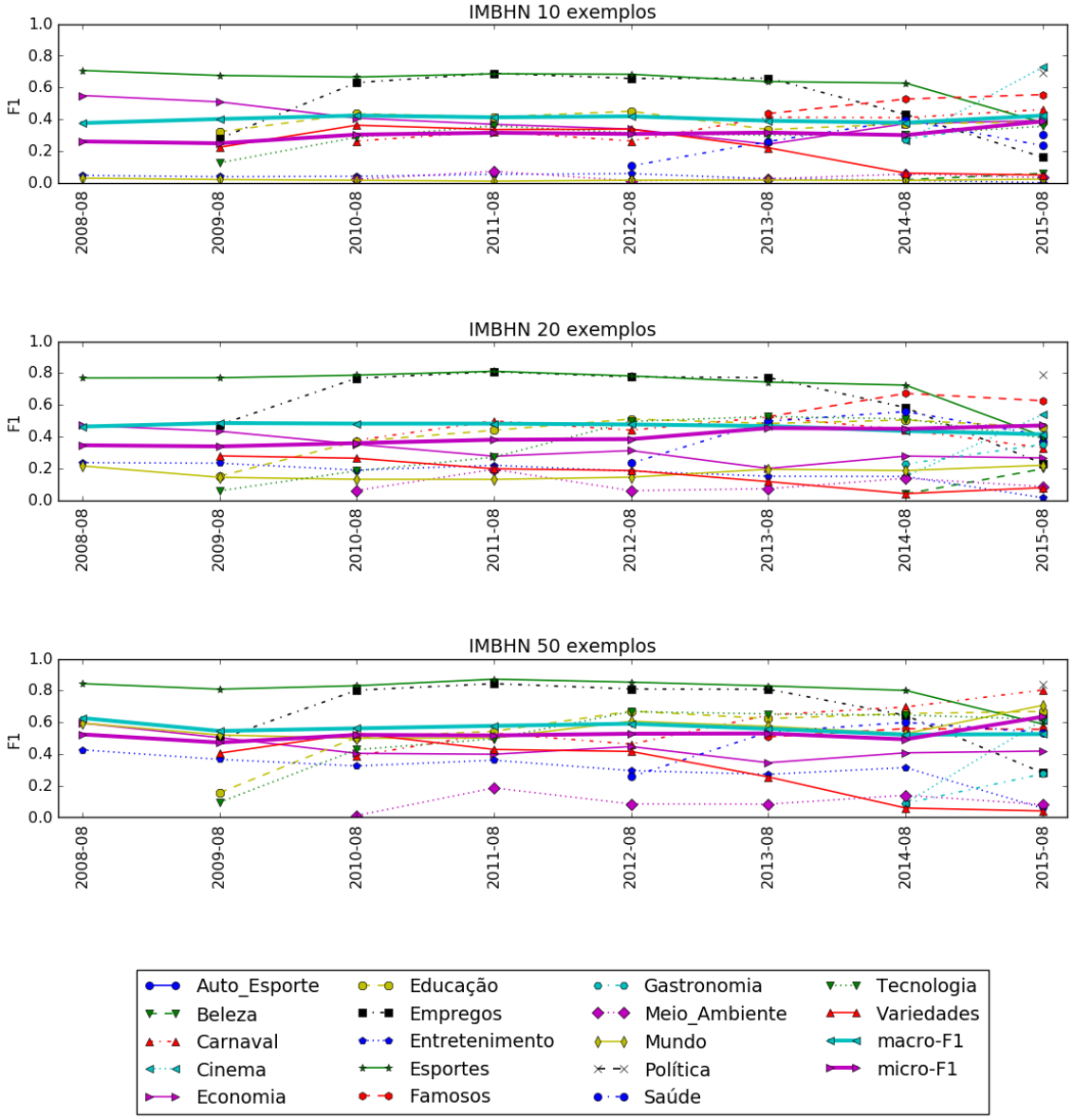

Figura 60. Resultado do algoritmo IMBHN em um conjunto com 10, 20 e 50 exemplos rotulados. Base: Notícias do Brasil (Variedades), período: Anual.

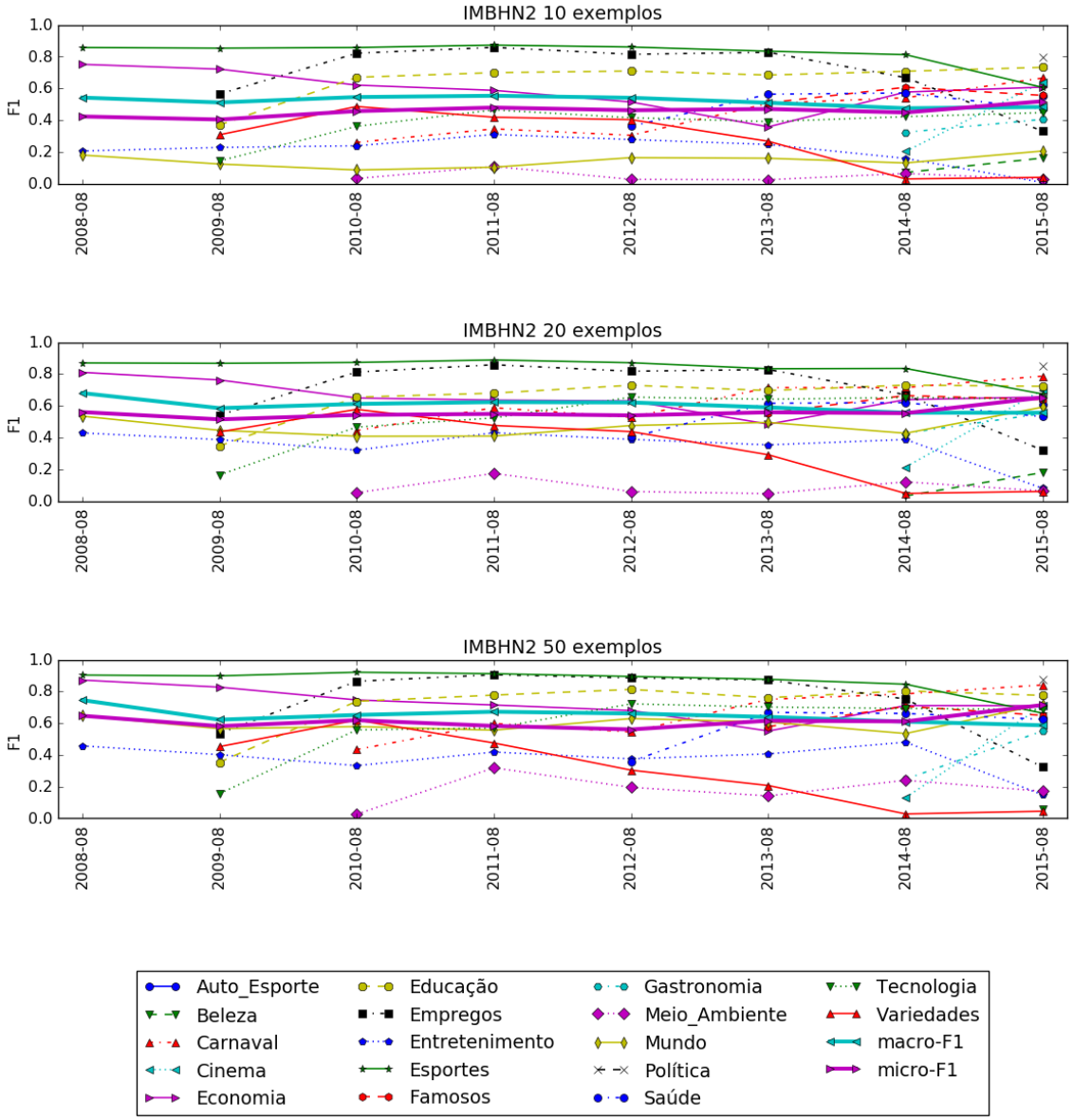

Figura 61. Resultado do algoritmo IMBHN2 em um conjunto com 10, 20 e 50 exemplos rotulados. Base: Notícias do Brasil (Variedades), período: Anual.

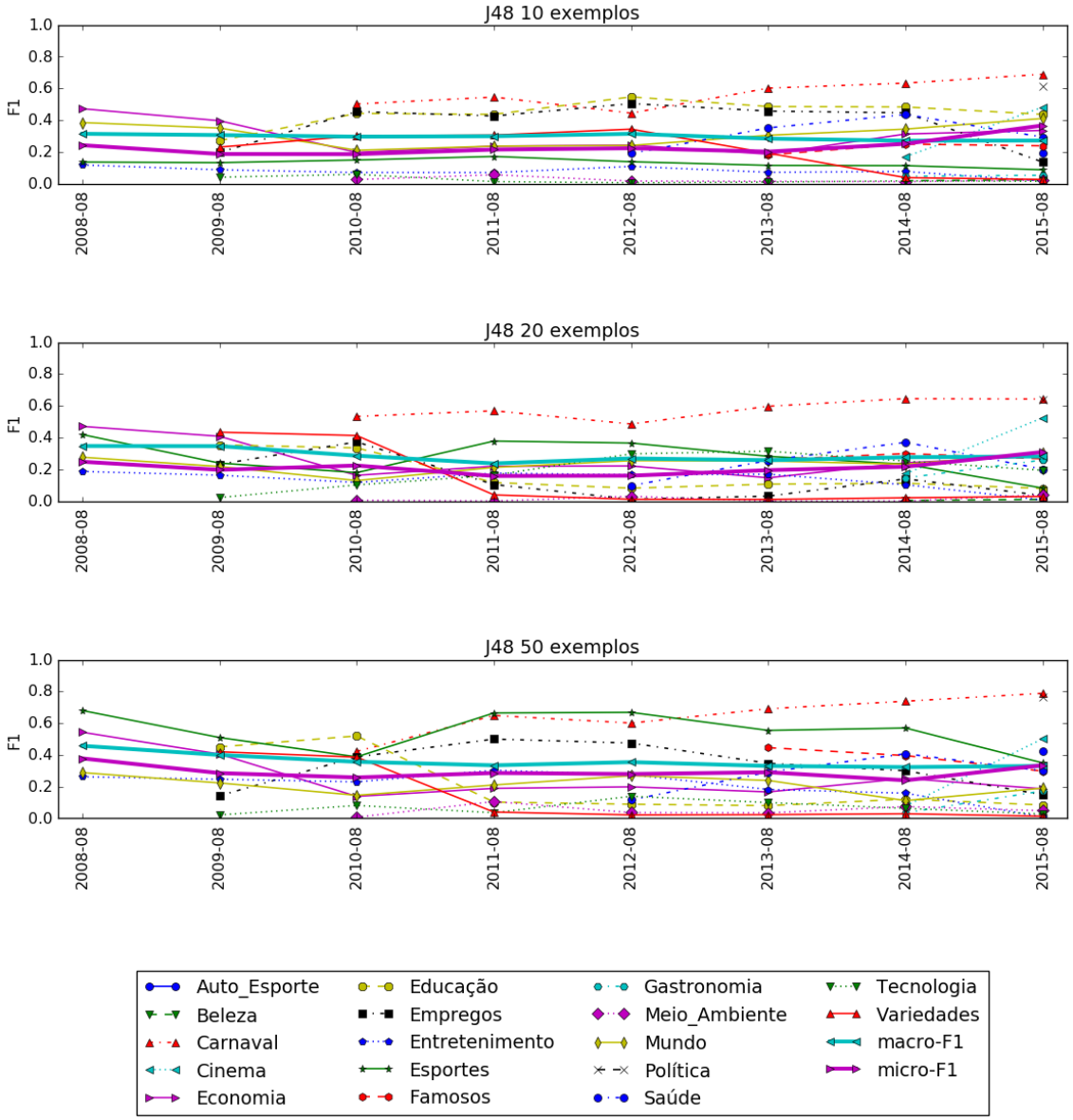

Figura 62. Resultado do algoritmo J48 em um conjunto com 10, 20 e 50 exemplos rotulados. Base: Notícias do Brasil (Variedades), período: Anual.

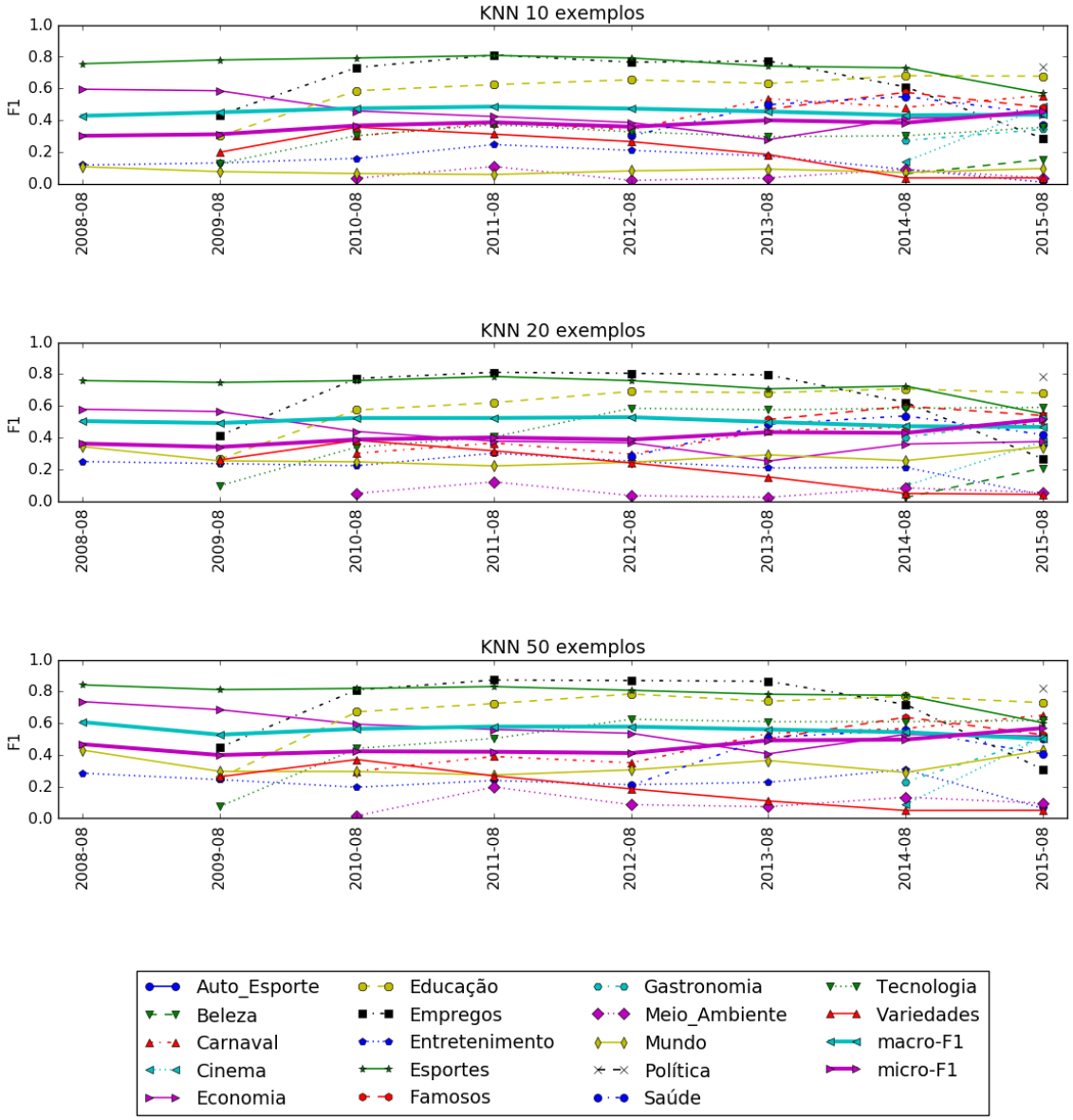

Figura 63. Resultado do algoritmo KNN em um conjunto com 10, 20 e 50 exemplos rotulados. Base: Notícias do Brasil (Variedades), período: Anual.

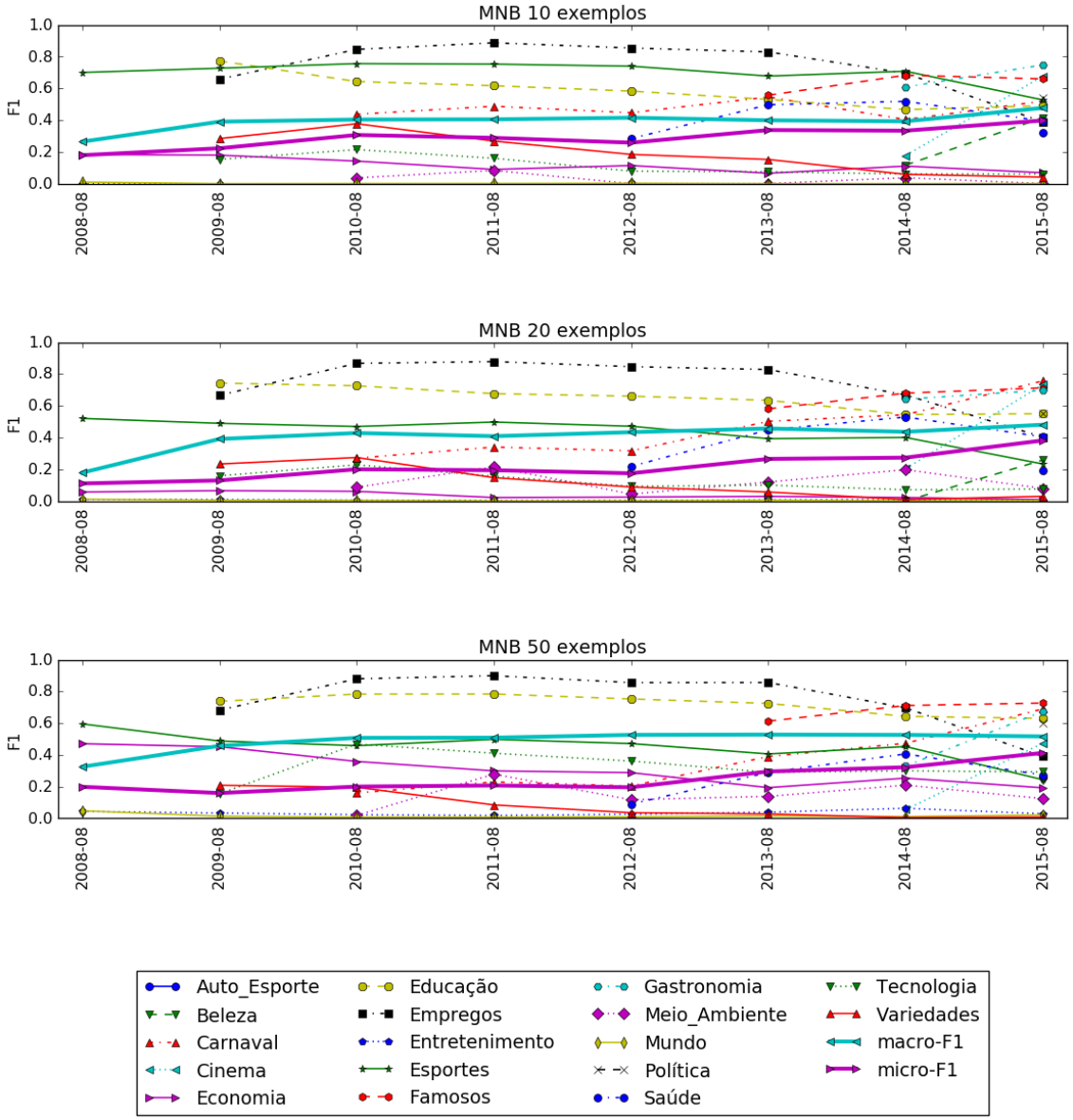

Figura 64. Resultado do algoritmo MNB em um conjunto com 10, 20 e 50 exemplos rotulados. Base: Notícias do Brasil (Variedades), período: Anual.

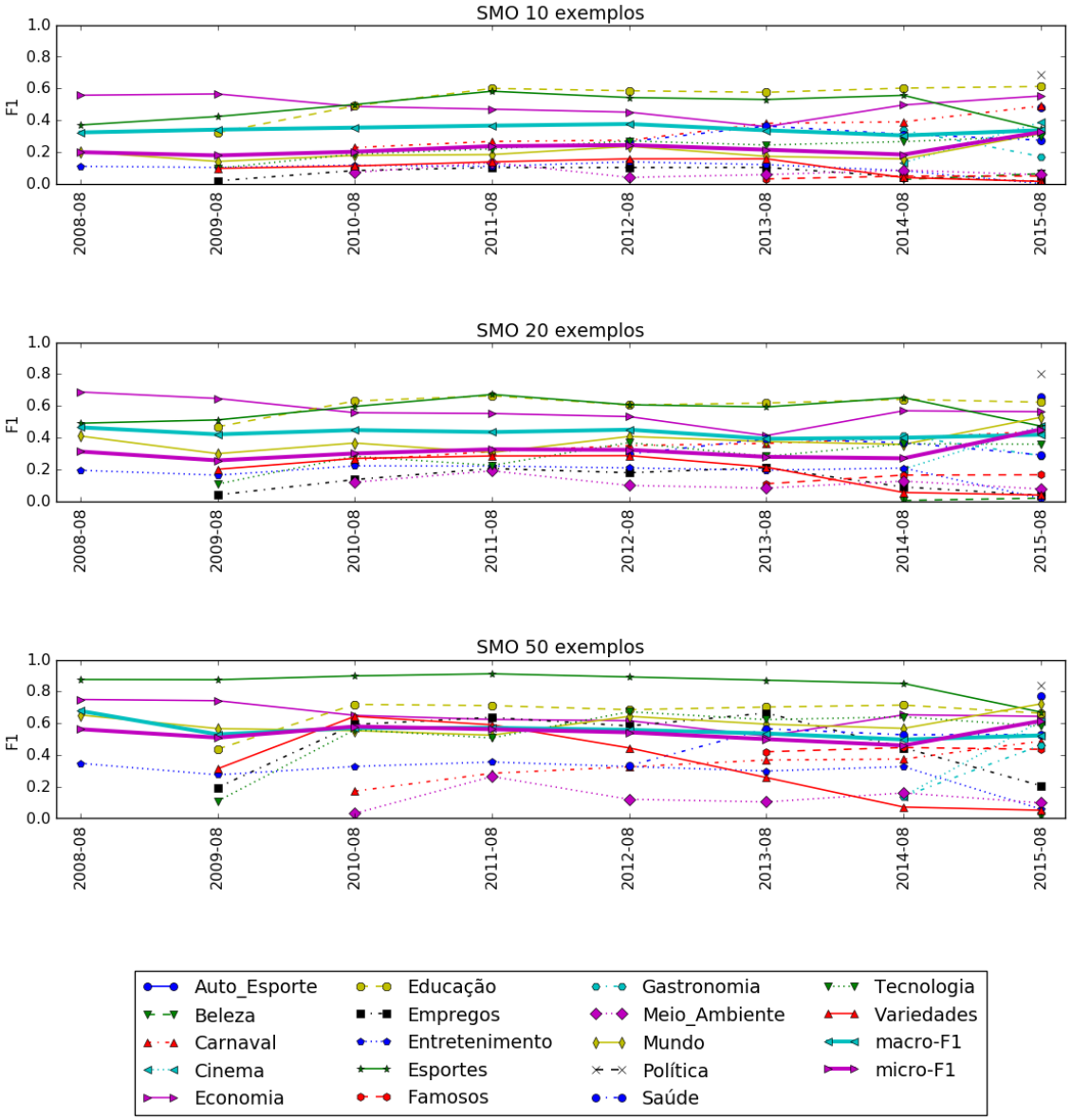

Figura 65. Resultado do algoritmo SMO em um conjunto com 10, 20 e 50 exemplos rotulados. Base: Notícias do Brasil (Variedades), período: Anual.

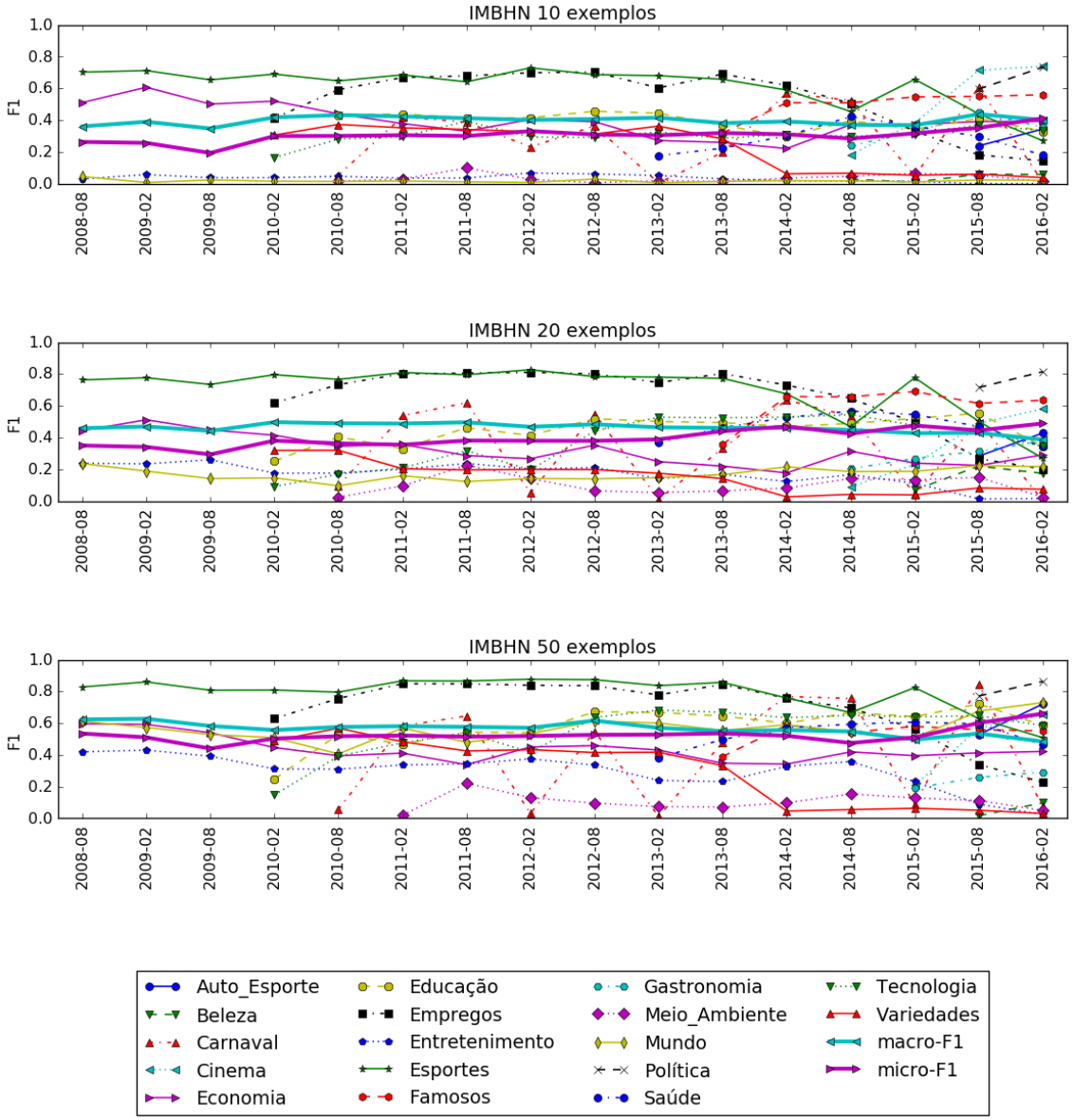

Figura 66. Resultado do algoritmo IMBHN em um conjunto com 10, 20 e 50 exemplos rotulados. Base: Notícias do Brasil (Variedades), período: Semestral.

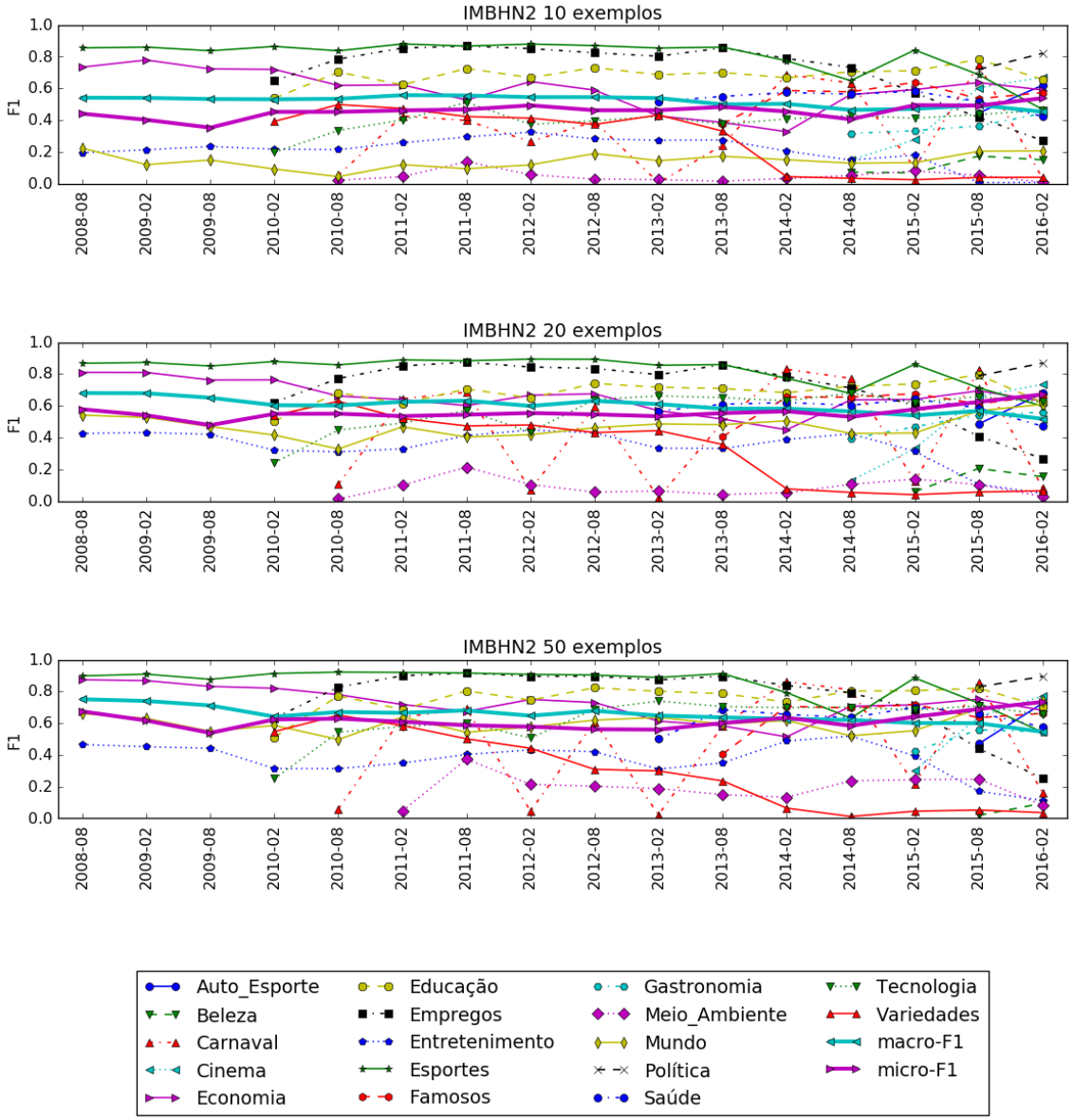

Figura 67. Resultado do algoritmo IMBHN2 em um conjunto com 10, 20 e 50 exemplos rotulados. Base: Notícias do Brasil (Variedades), período: Semestral.

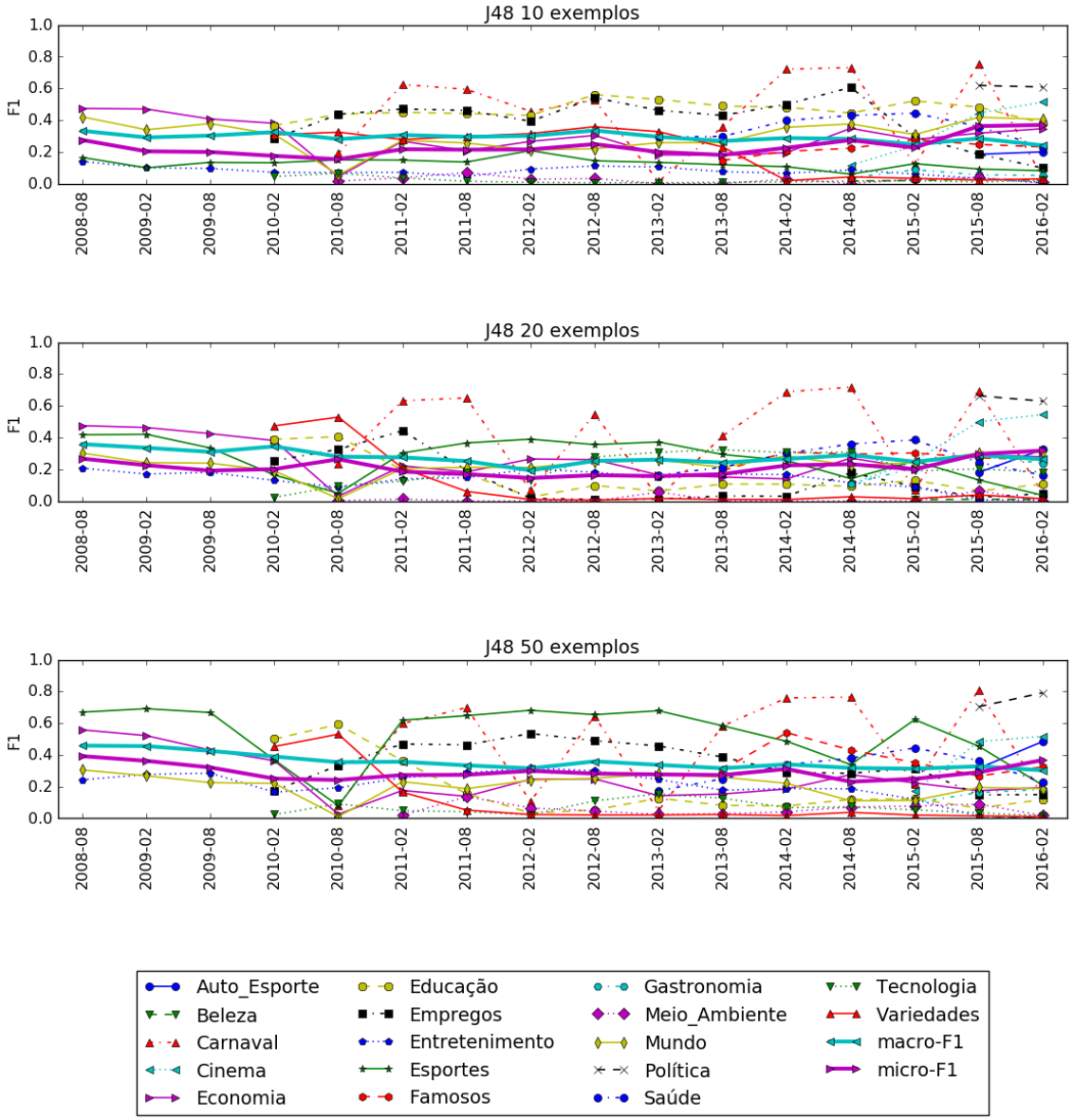

Figura 68. Resultado do algoritmo J48 em um conjunto com 10, 20 e 50 exemplos rotulados. Base: Notícias do Brasil (Variedades), período: Semestral.

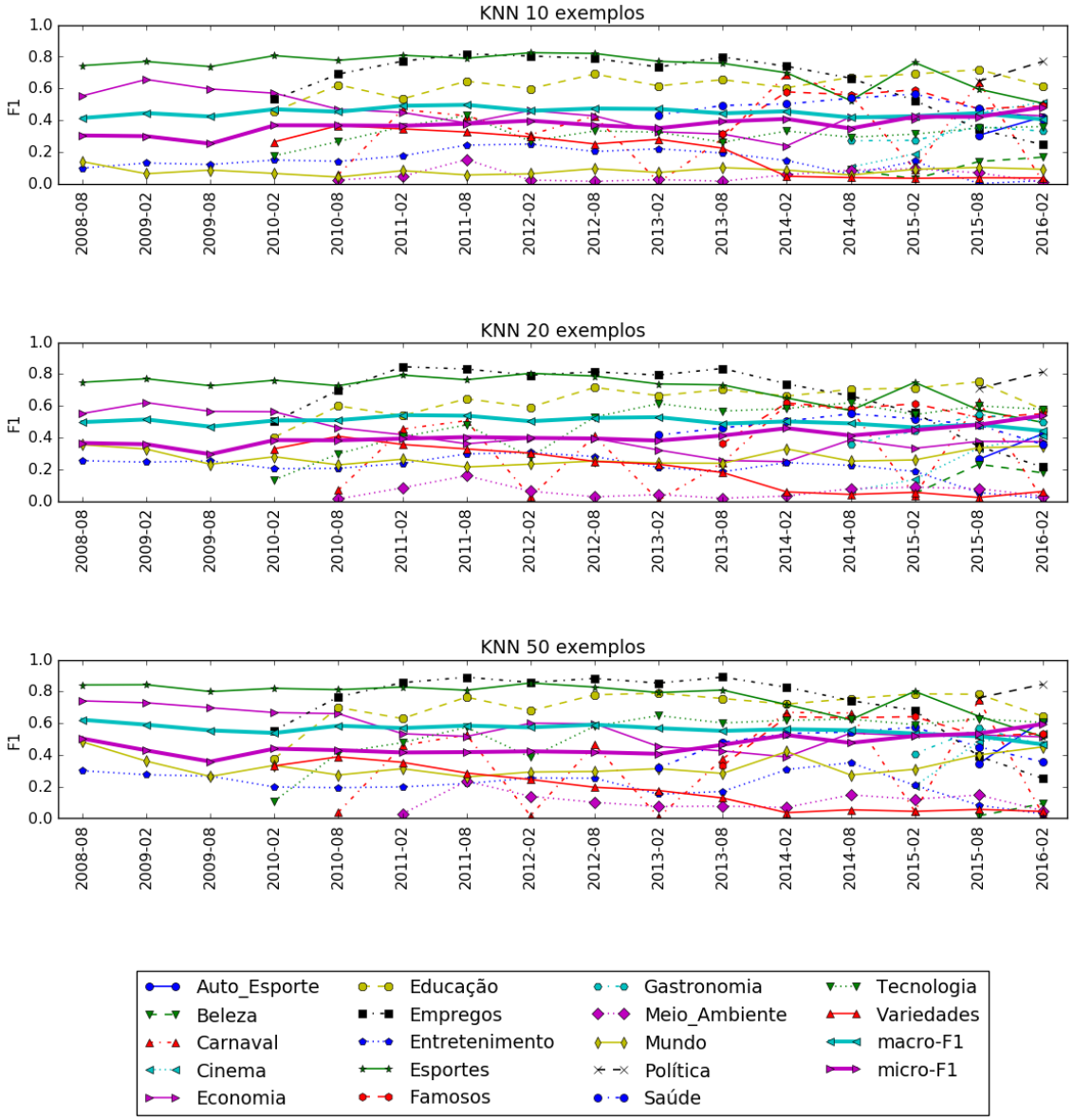

Figura 69. Resultado do algoritmo KNN em um conjunto com 10, 20 e 50 exemplos rotulados. Base: Notícias do Brasil (Variedades), período: Semestral.

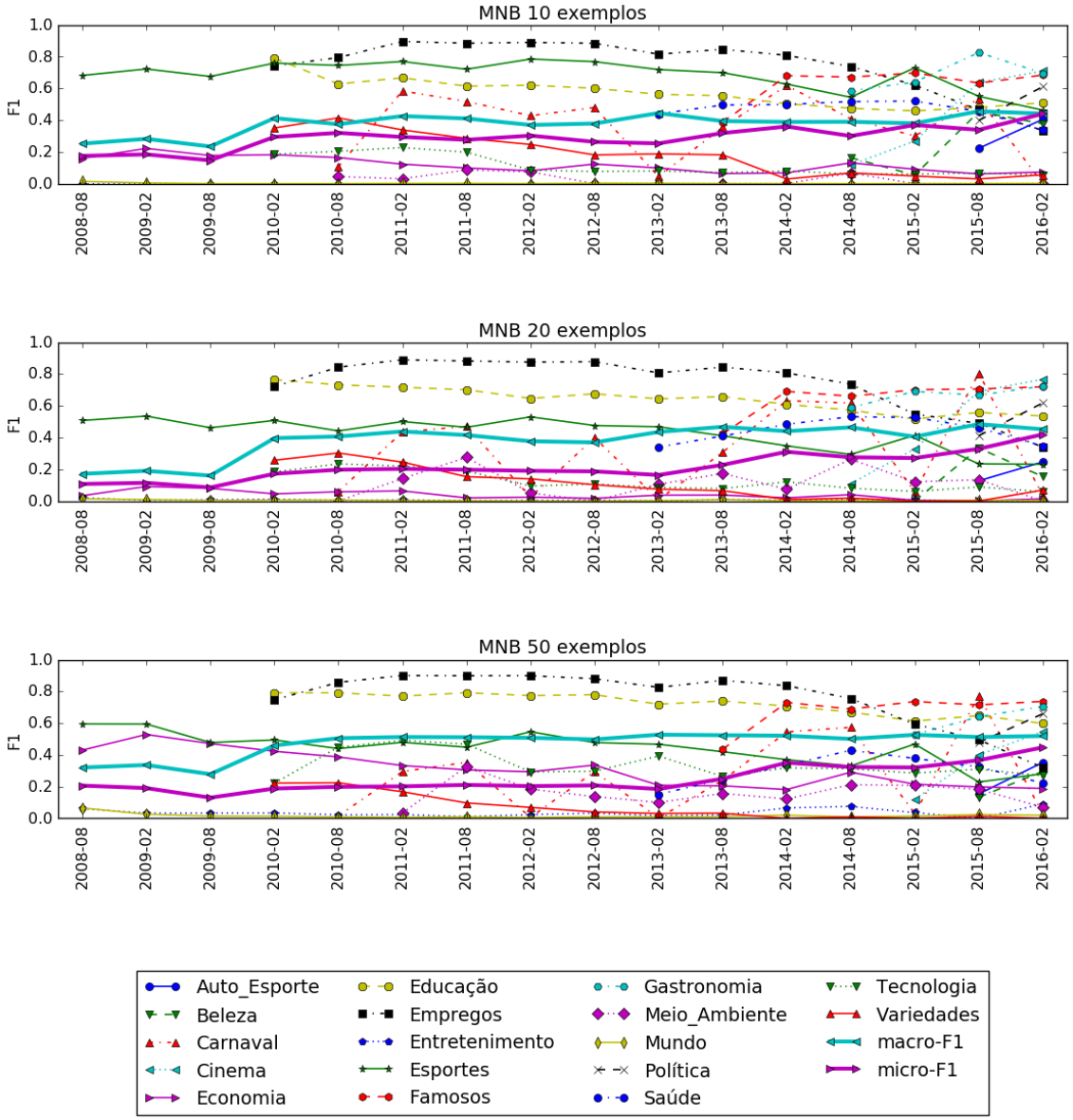

Figura 70. Resultado do algoritmo MNB em um conjunto com 10, 20 e 50 exemplos rotulados. Base: Notícias do Brasil (Variedades), período: Semestral.

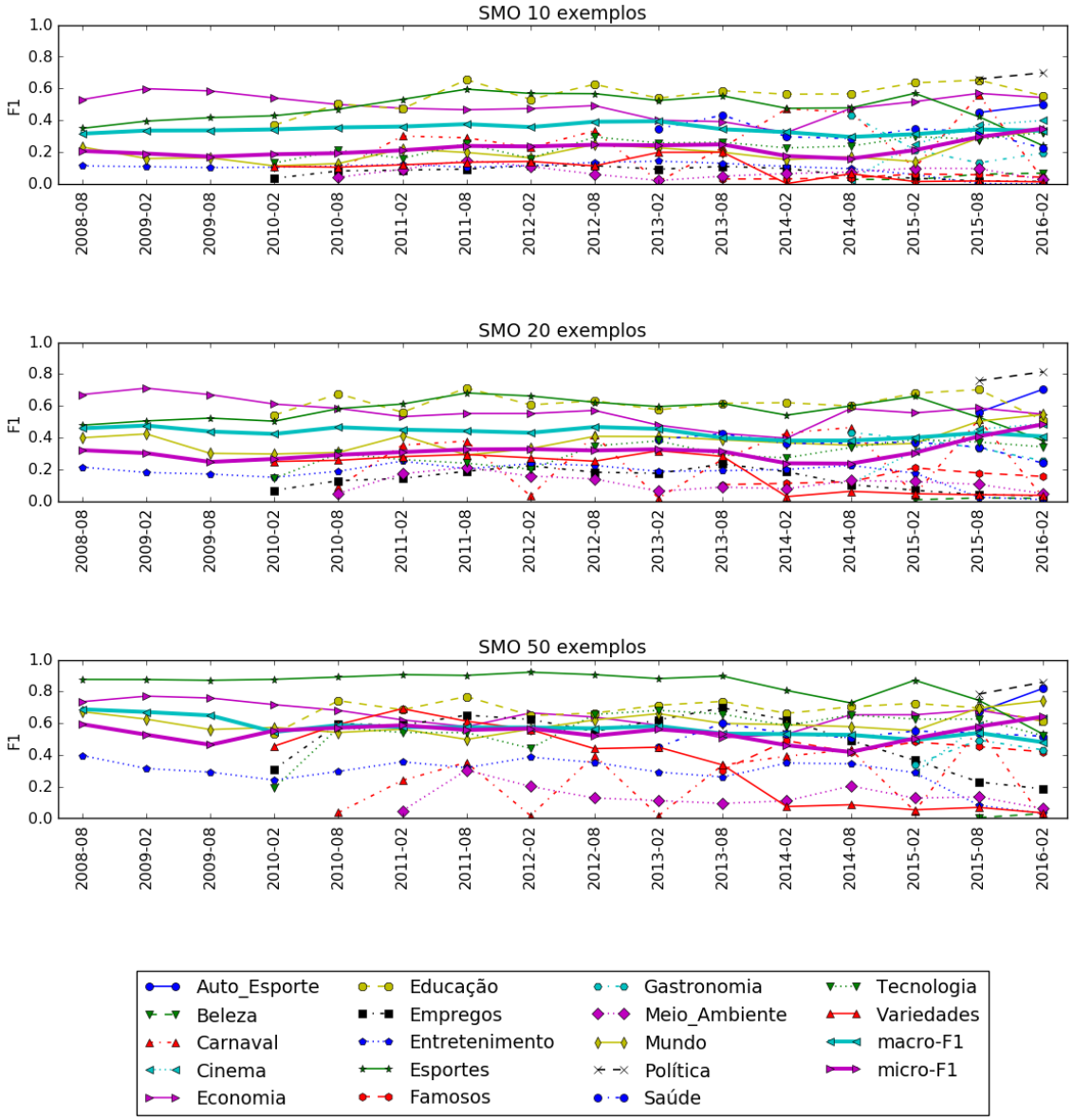

Figura 71. Resultado do algoritmo SMO em um conjunto com 10, 20 e 50 exemplos rotulados. Base: Notícias do Brasil (Variedades), período: Semestral.

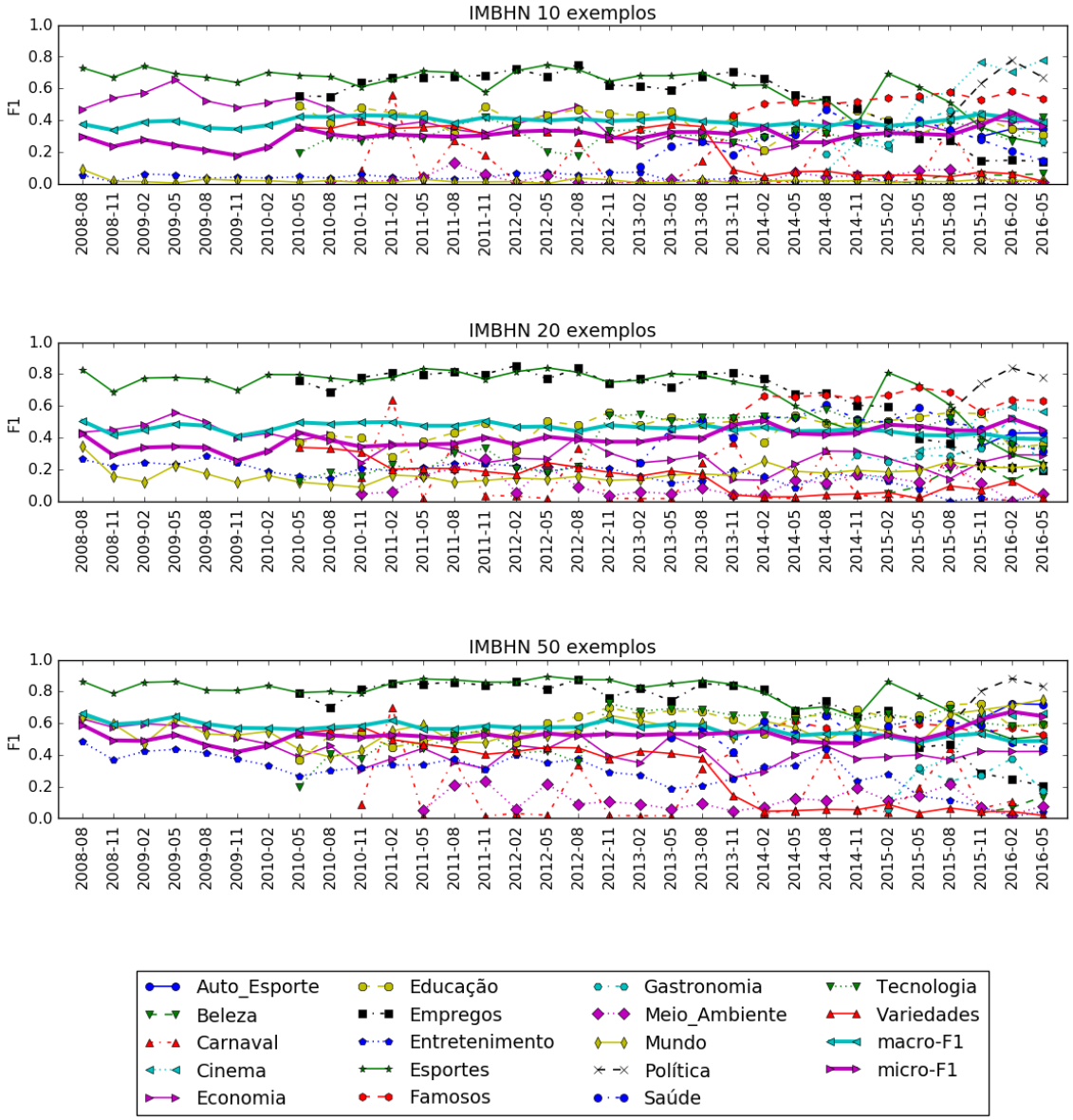

Figura 72. Resultado do algoritmo IMBHN em um conjunto com 10, 20 e 50 exemplos rotulados. Base: Notícias do Brasil (Variedades), período: Trimestral.

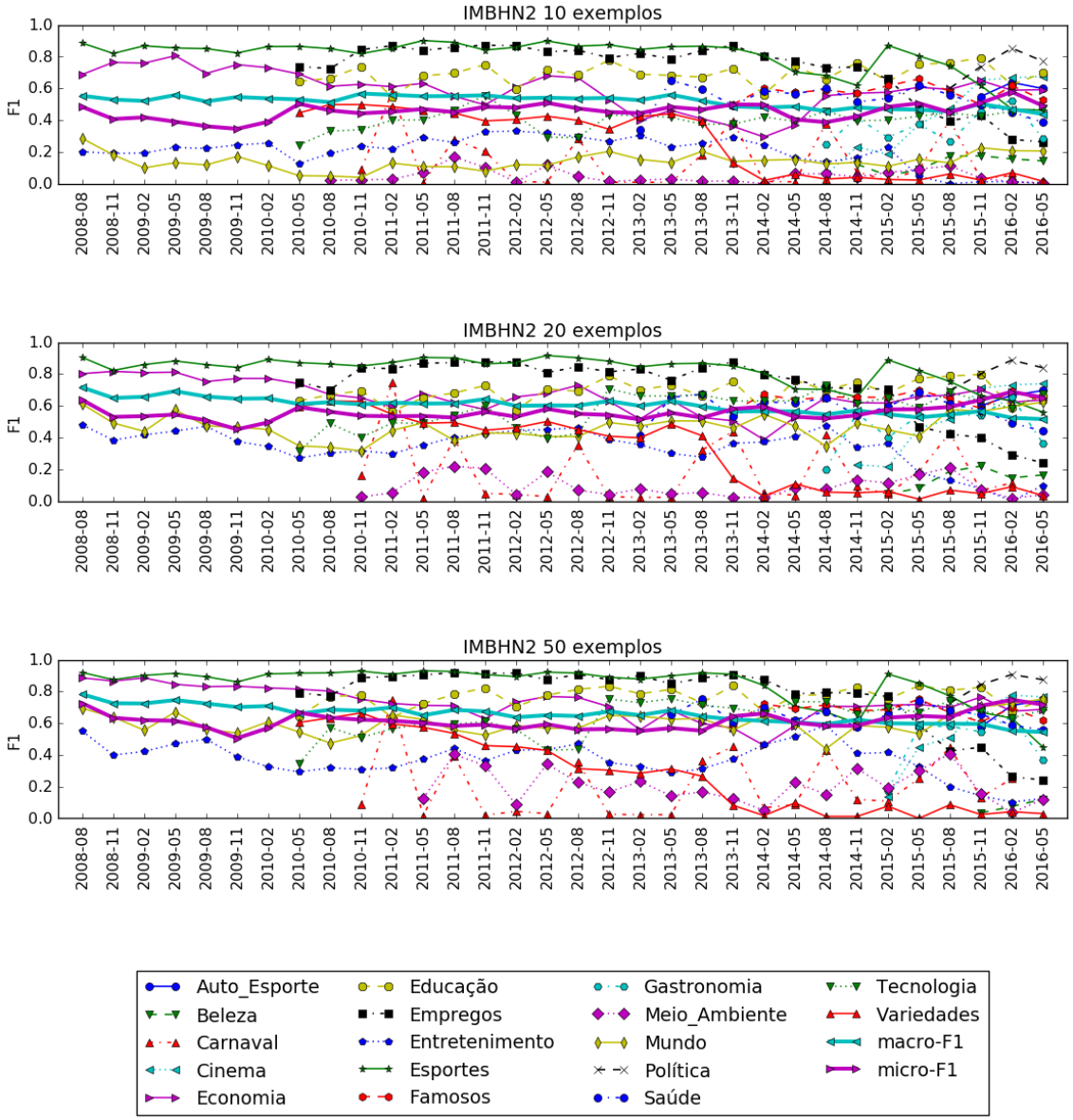

Figura 73. Resultado do algoritmo IMBHN2 em um conjunto com 10, 20 e 50 exemplos rotulados. Base: Notícias do Brasil (Variedades), período: Trimestral.

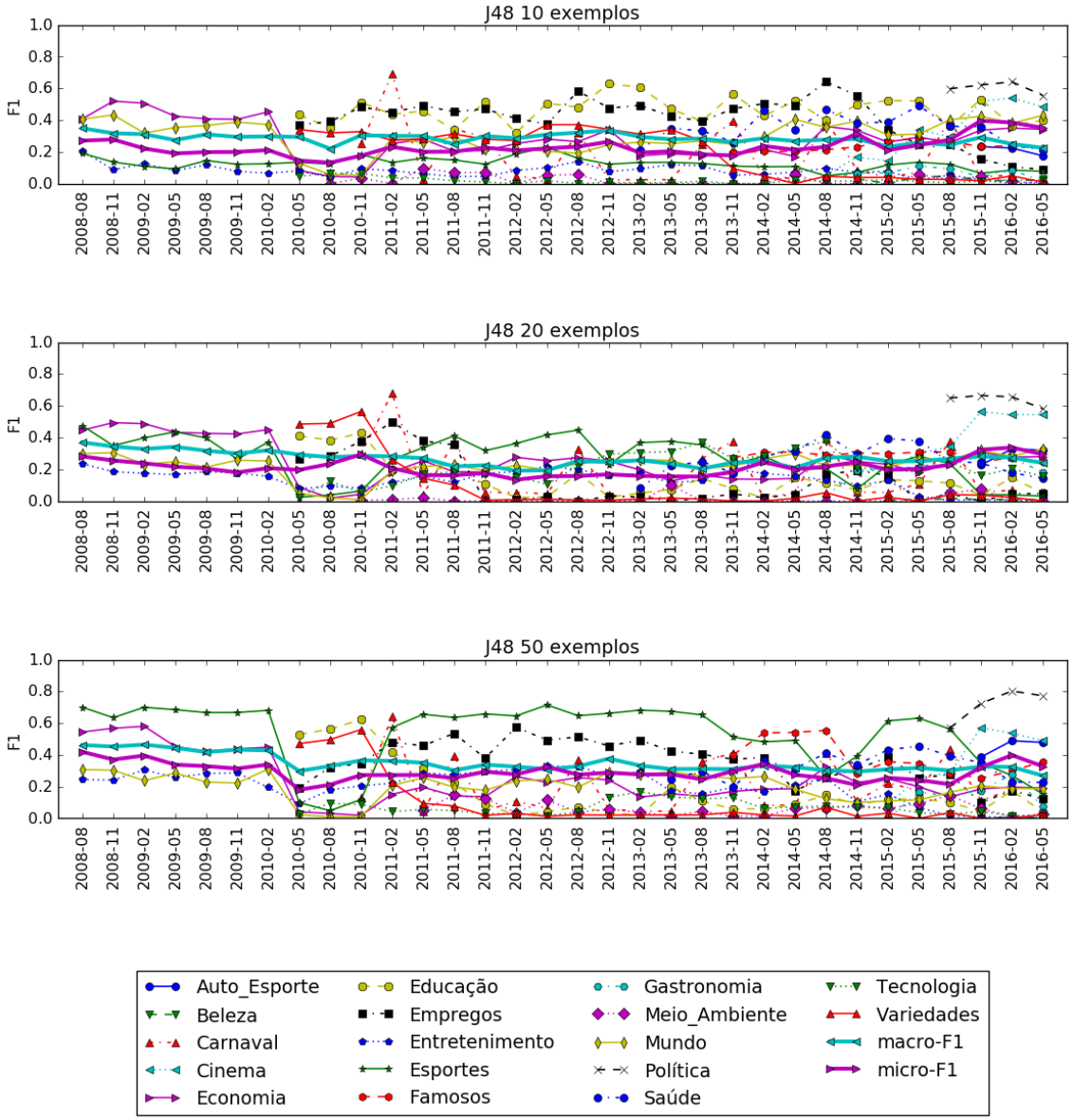

Figura 74. Resultado do algoritmo J48 em um conjunto com 10, 20 e 50 exemplos rotulados. Base: Notícias do Brasil (Variedades), período: Trimestral.

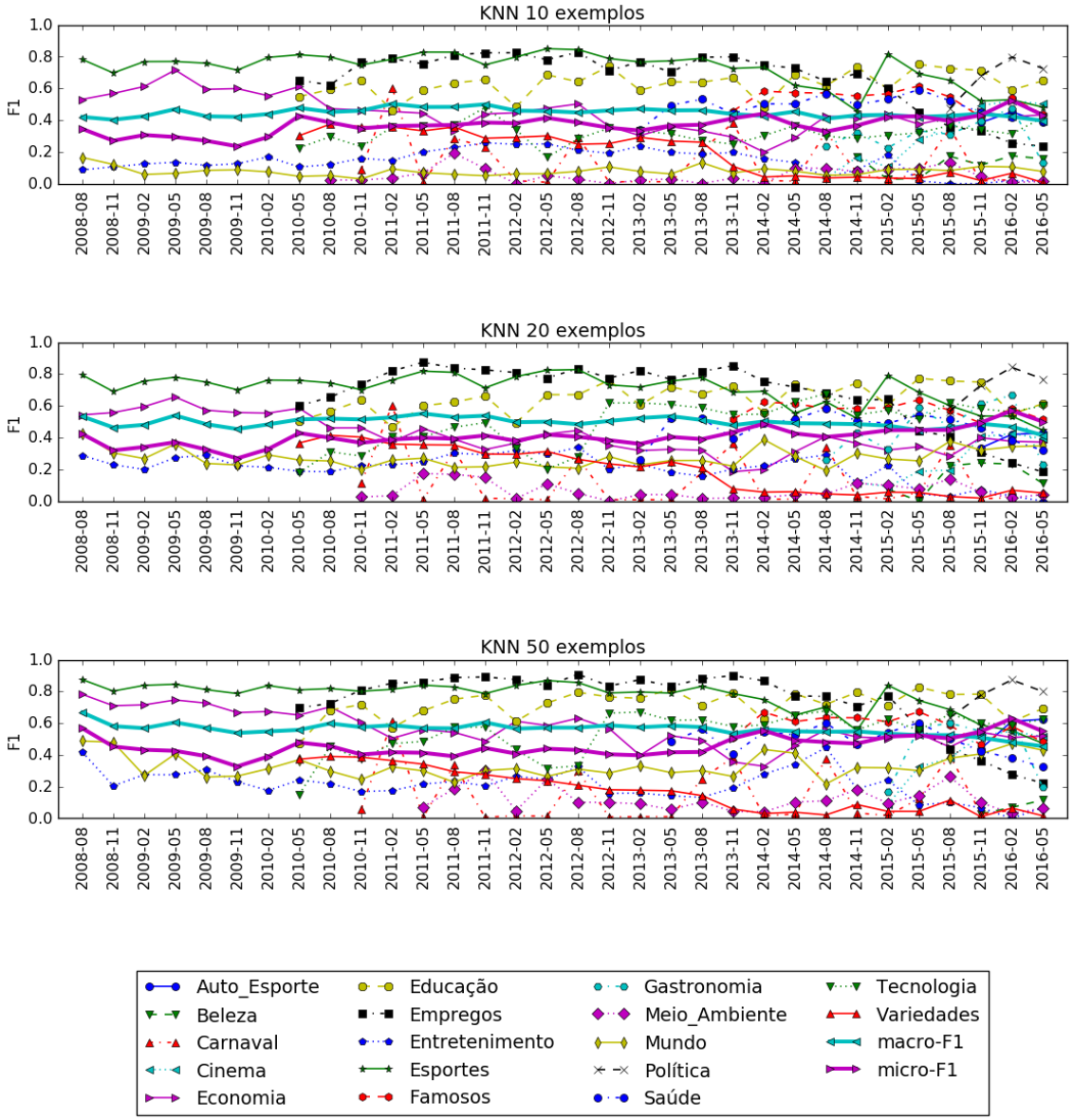

Figura 75. Resultado do algoritmo KNN em um conjunto com 10, 20 e 50 exemplos rotulados. Base: Notícias do Brasil (Variedades), período: Trimestral.

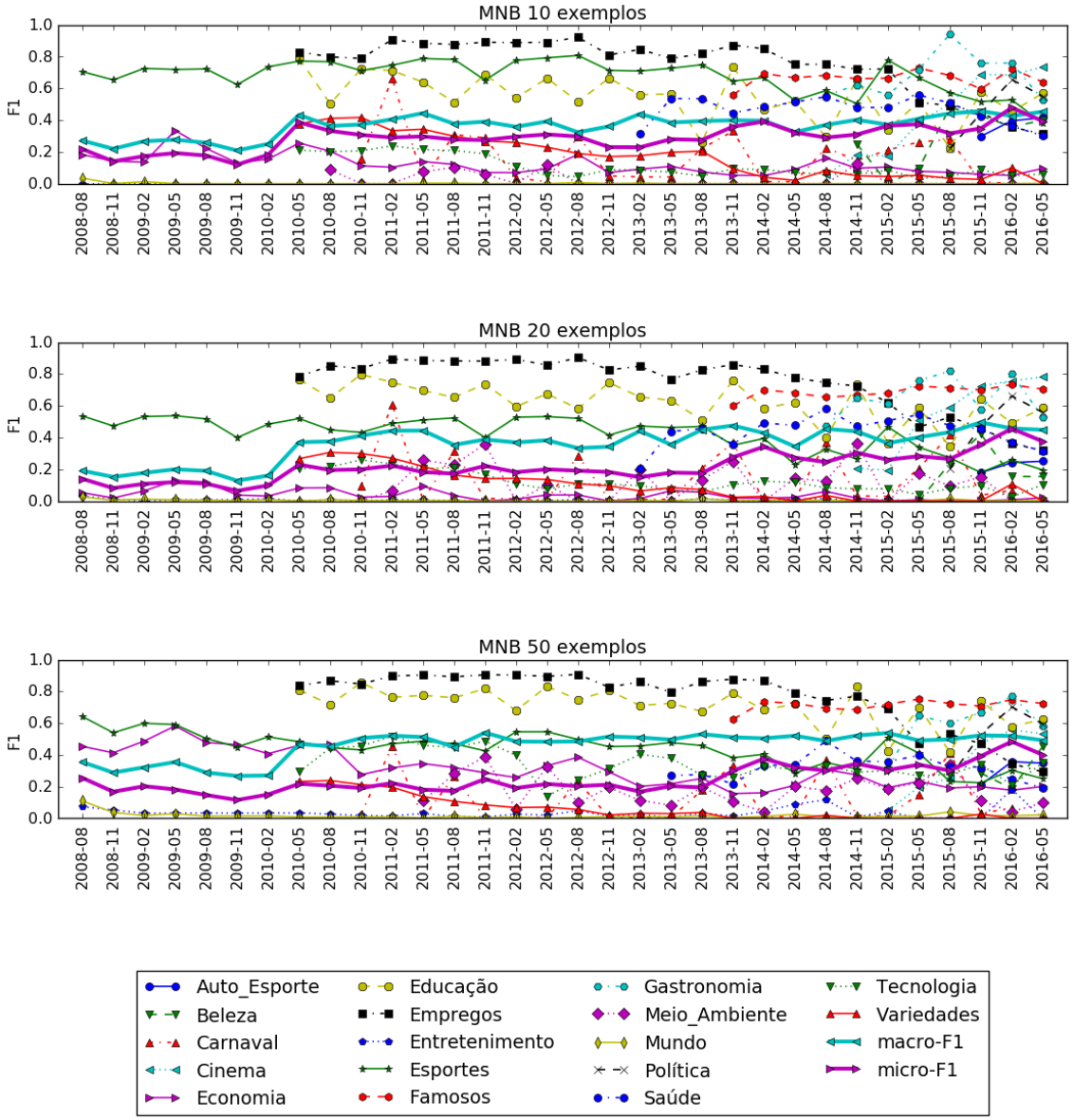

Figura 76. Resultado do algoritmo MNB em um conjunto com 10, 20 e 50 exemplos rotulados. Base: Notícias do Brasil (Variedades), período: Trimestral.

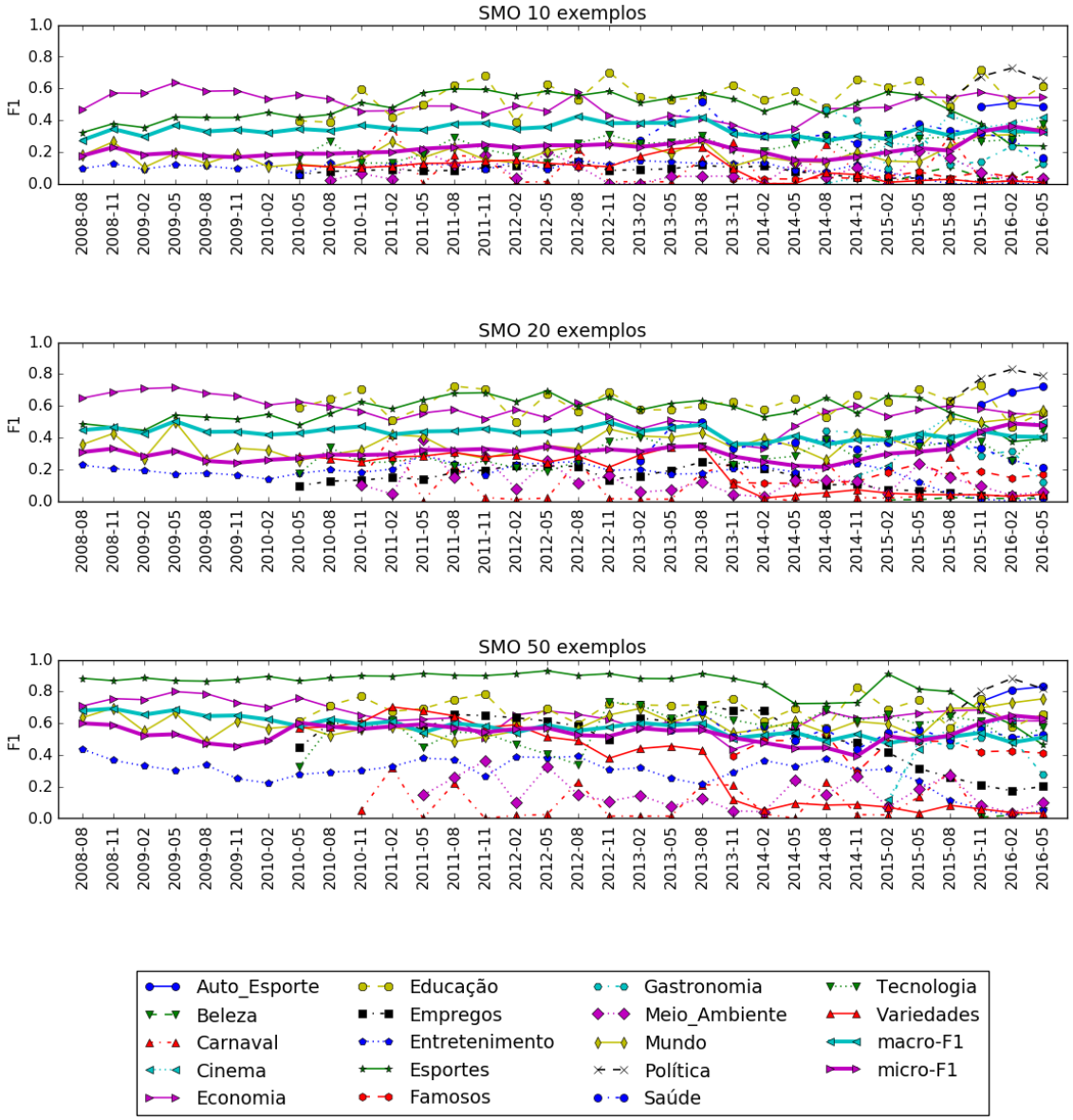

Figura 77. Resultado do algoritmo SMO em um conjunto com 10, 20 e 50 exemplos rotulados. Base: Notícias do Brasil (Variedades), período: Trimestral.

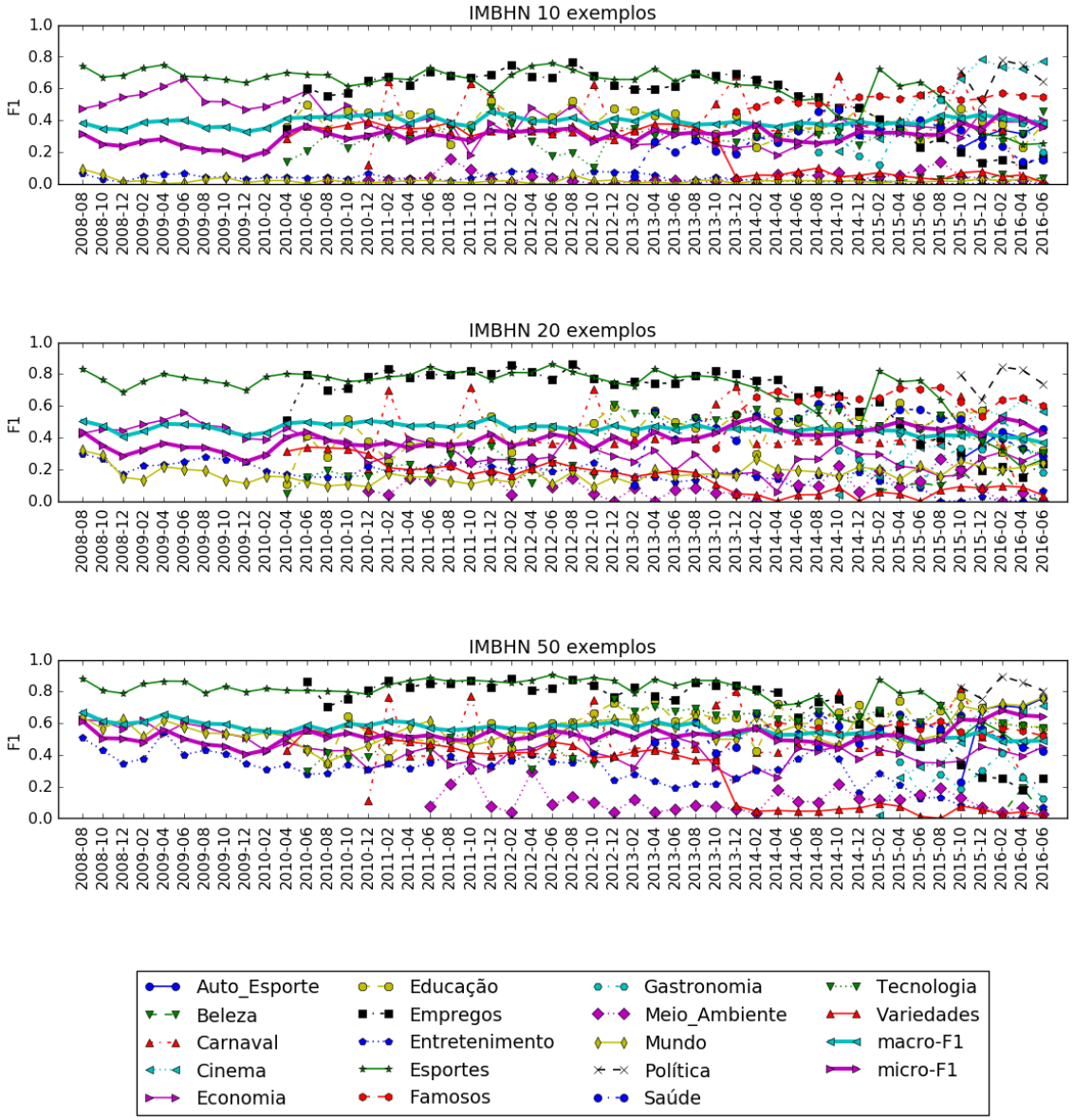

Figura 78. Resultado do algoritmo IMBHN em um conjunto com 10, 20 e 50 exemplos rotulados. Base: Notícias do Brasil (Variedades), período: Bimestral.

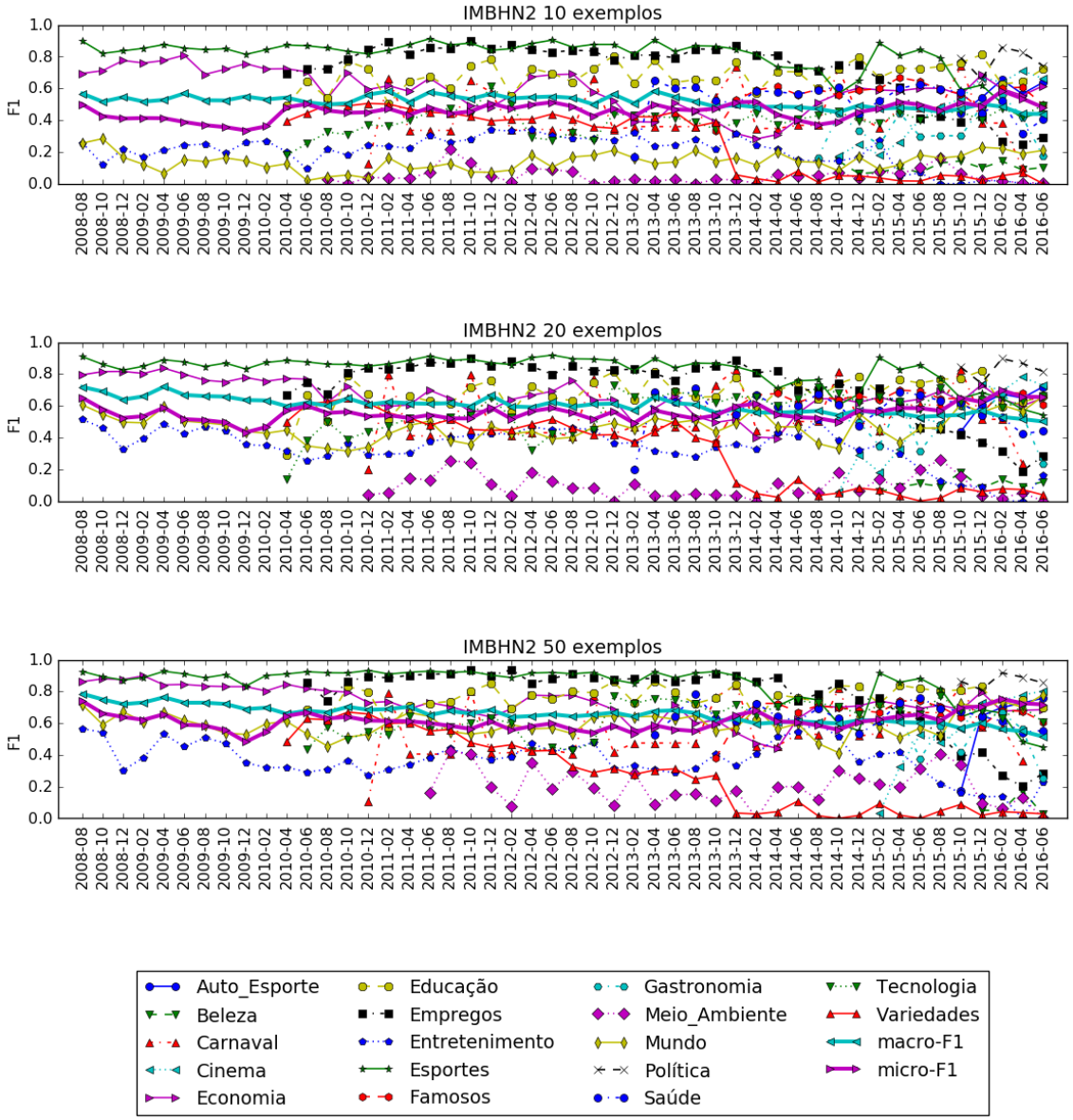

Figura 79. Resultado do algoritmo IMBHN2 em um conjunto com 10, 20 e 50 exemplos rotulados. Base: Notícias do Brasil (Variedades), período: Bimestral.

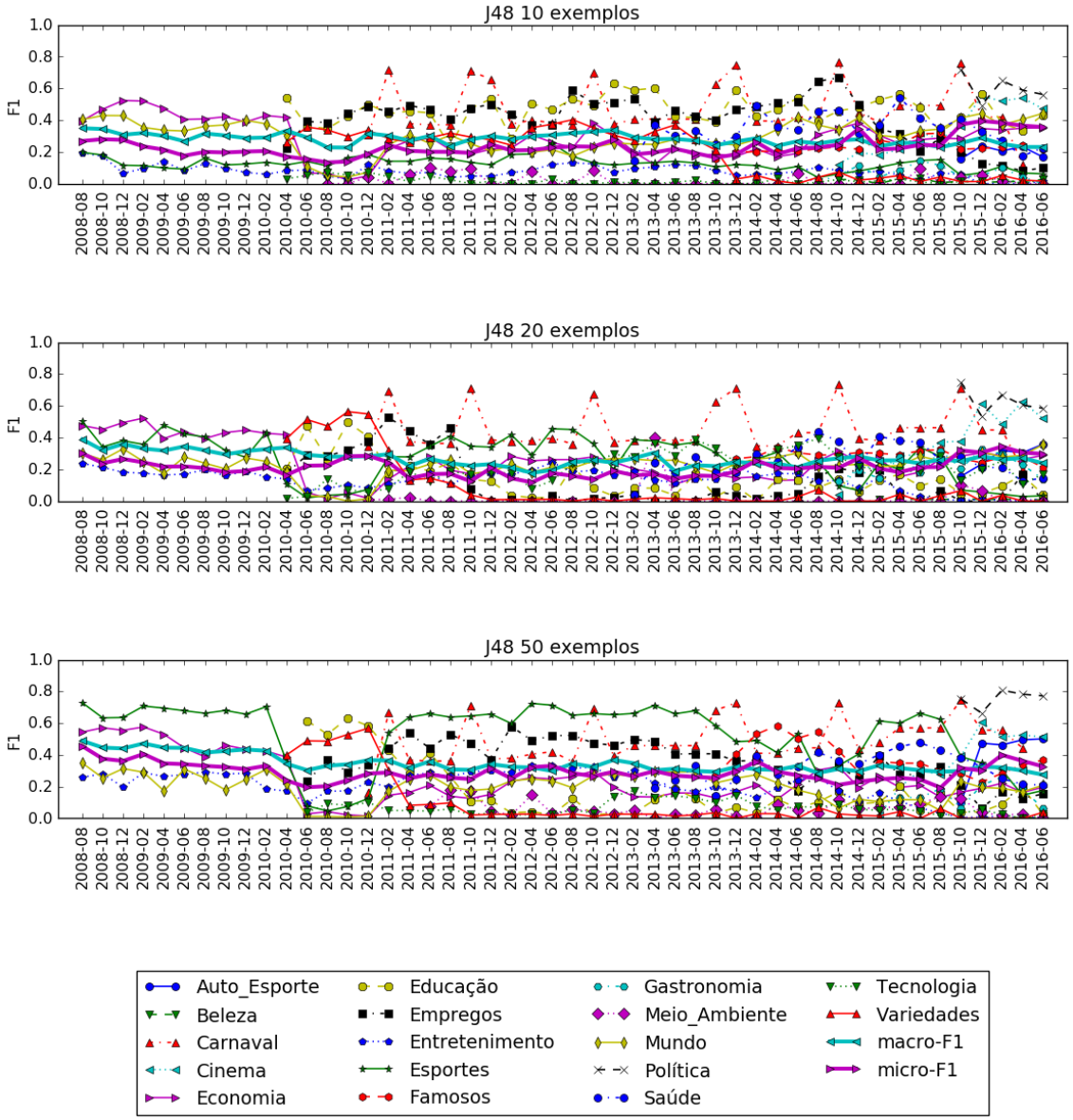

Figura 80. Resultado do algoritmo J48 em um conjunto com 10, 20 e 50 exemplos rotulados. Base: Notícias do Brasil (Variedades), período: Bimestral.

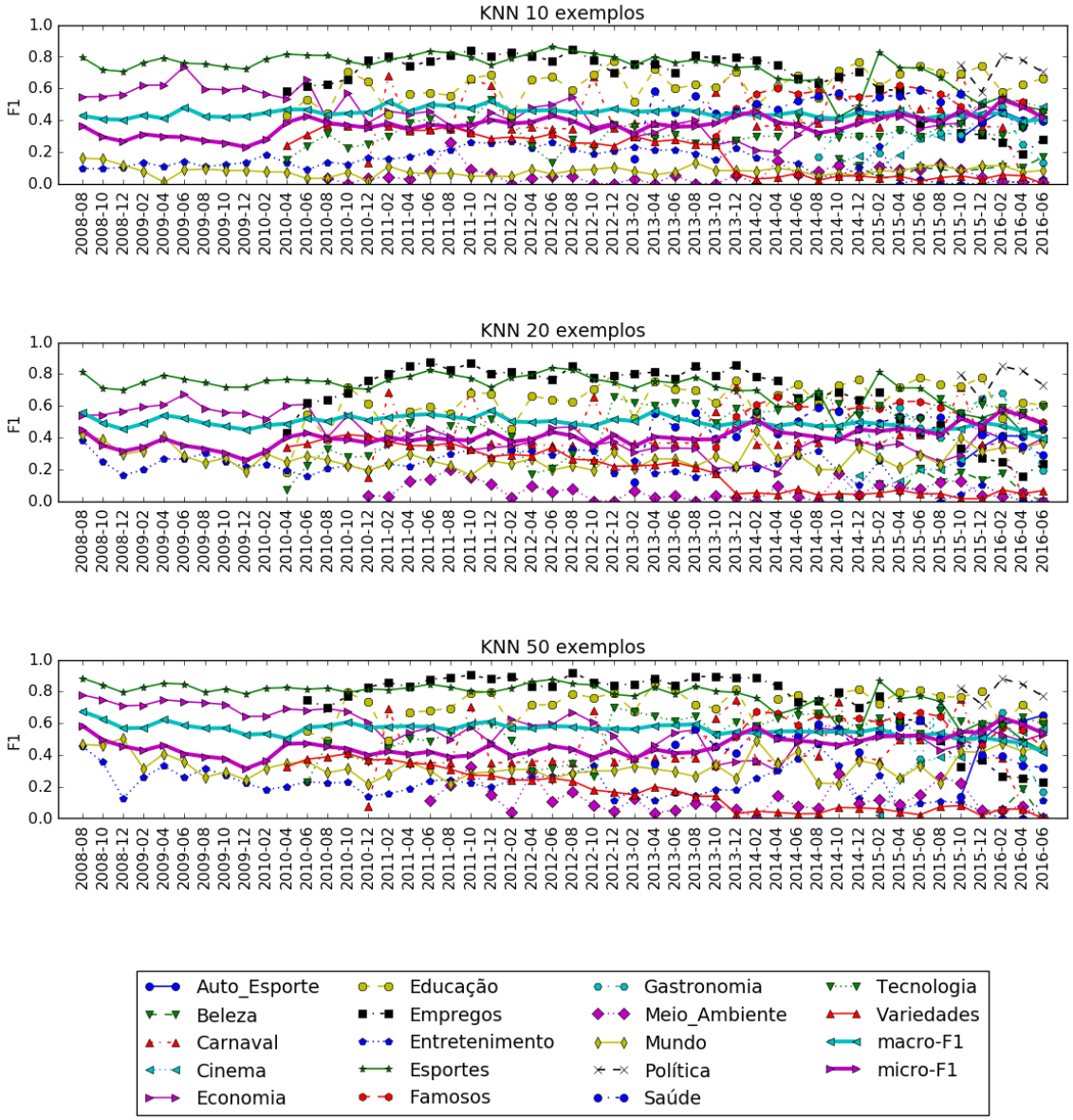

Figura 81. Resultado do algoritmo KNN em um conjunto com 10, 20 e 50 exemplos rotulados. Base: Notícias do Brasil (Variedades), período: Bimestral.

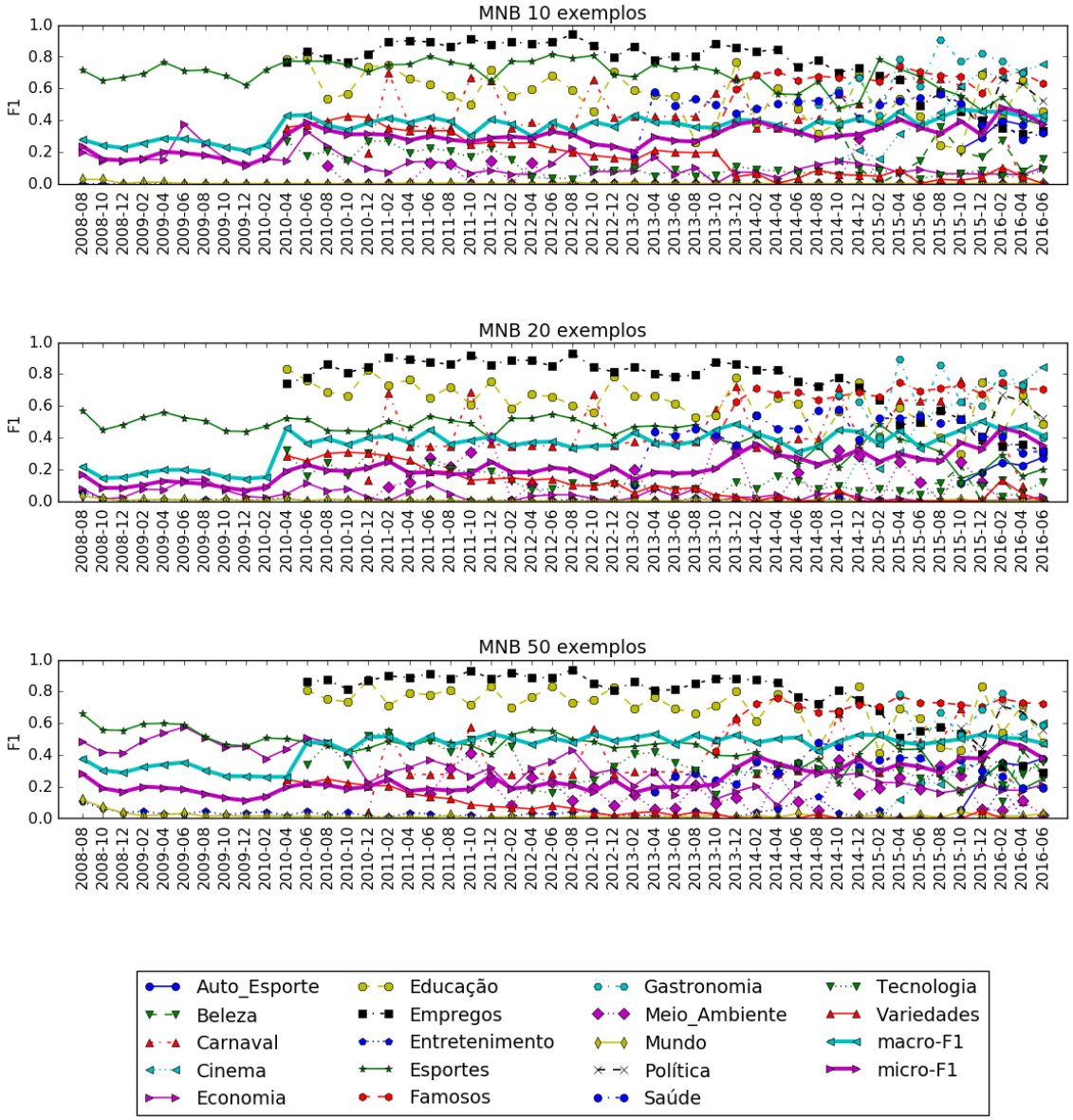

Figura 82. Resultado do algoritmo MNB em um conjunto com 10, 20 e 50 exemplos rotulados. Base: Notícias do Brasil (Variedades), período: Bimestral.

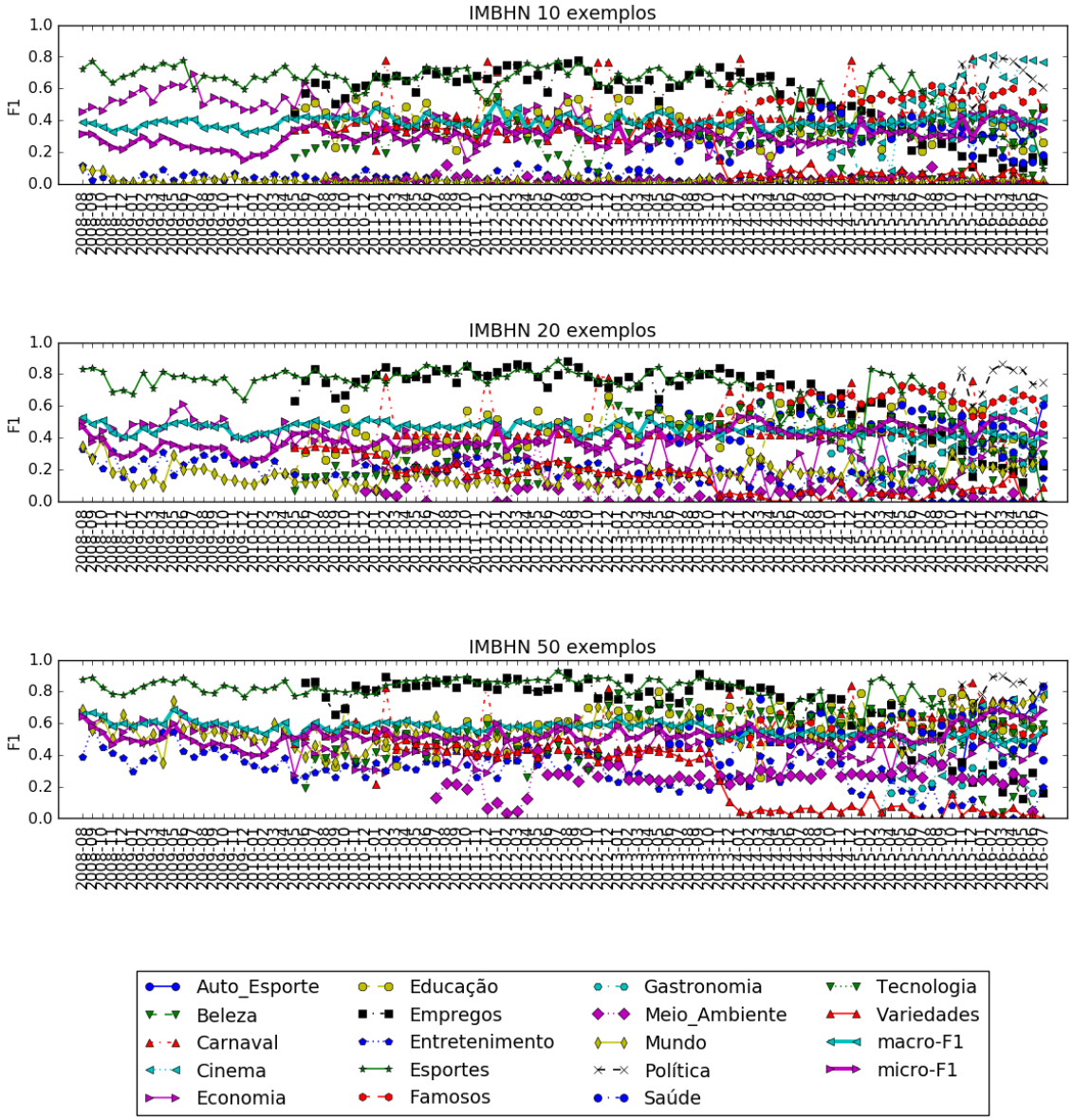

Figura 83. Resultado do algoritmo IMBHN em um conjunto com 10, 20 e 50 exemplos rotulados. Base: Notícias do Brasil (Variedades), período: Mensal.

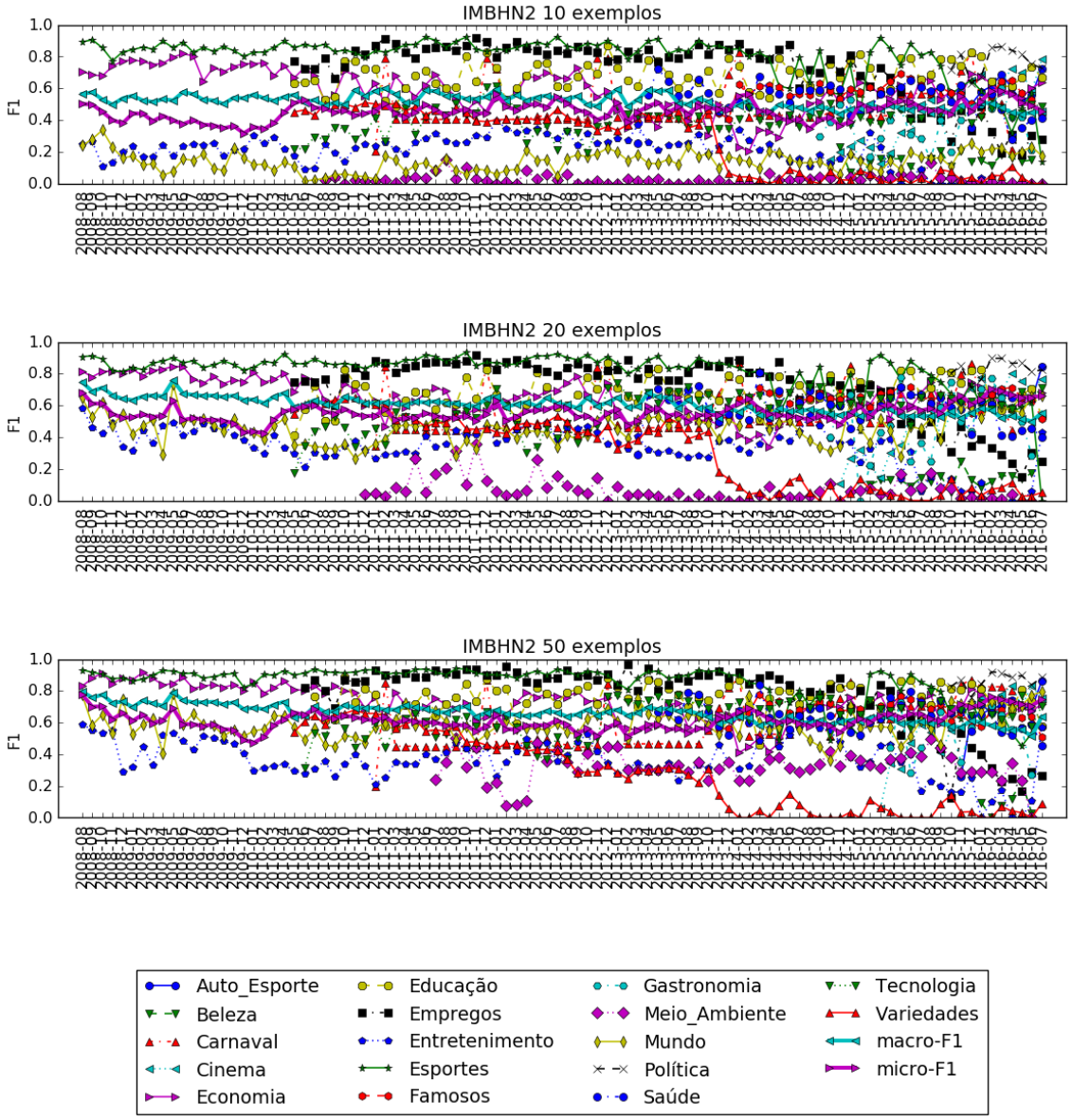

Figura 84. Resultado do algoritmo IMBHN2 em um conjunto com 10, 20 e 50 exemplos rotulados. Base: Notícias do Brasil (Variedades), período: Mensal.

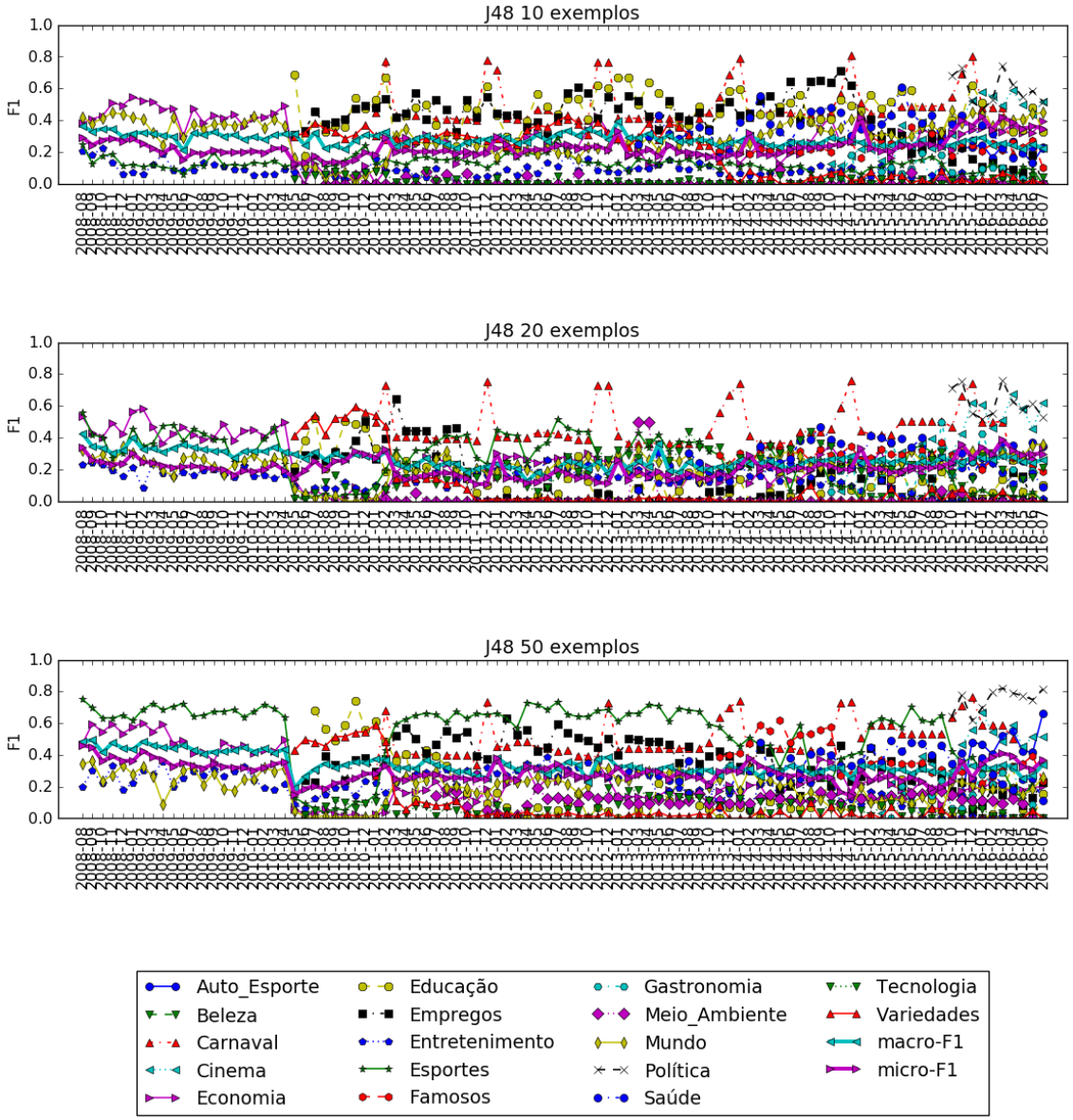

Figura 85. Resultado do algoritmo J48 em um conjunto com 10, 20 e 50 exemplos rotulados. Base: Notícias do Brasil (Variedades), período: Mensal.

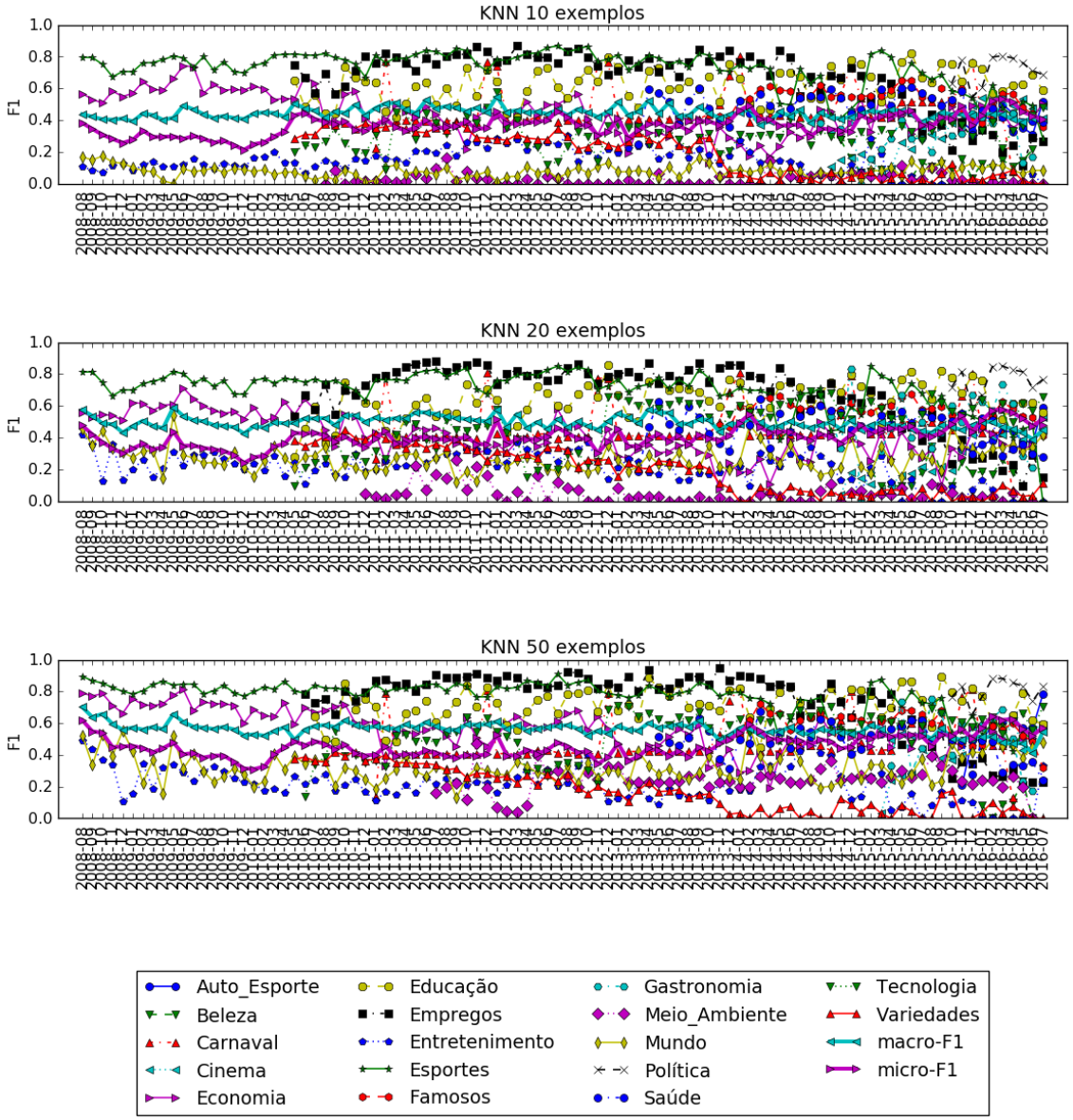

Figura 86. Resultado do algoritmo KNN em um conjunto com 10, 20 e 50 exemplos rotulados. Base: Notícias do Brasil (Variedades), período: Mensal.

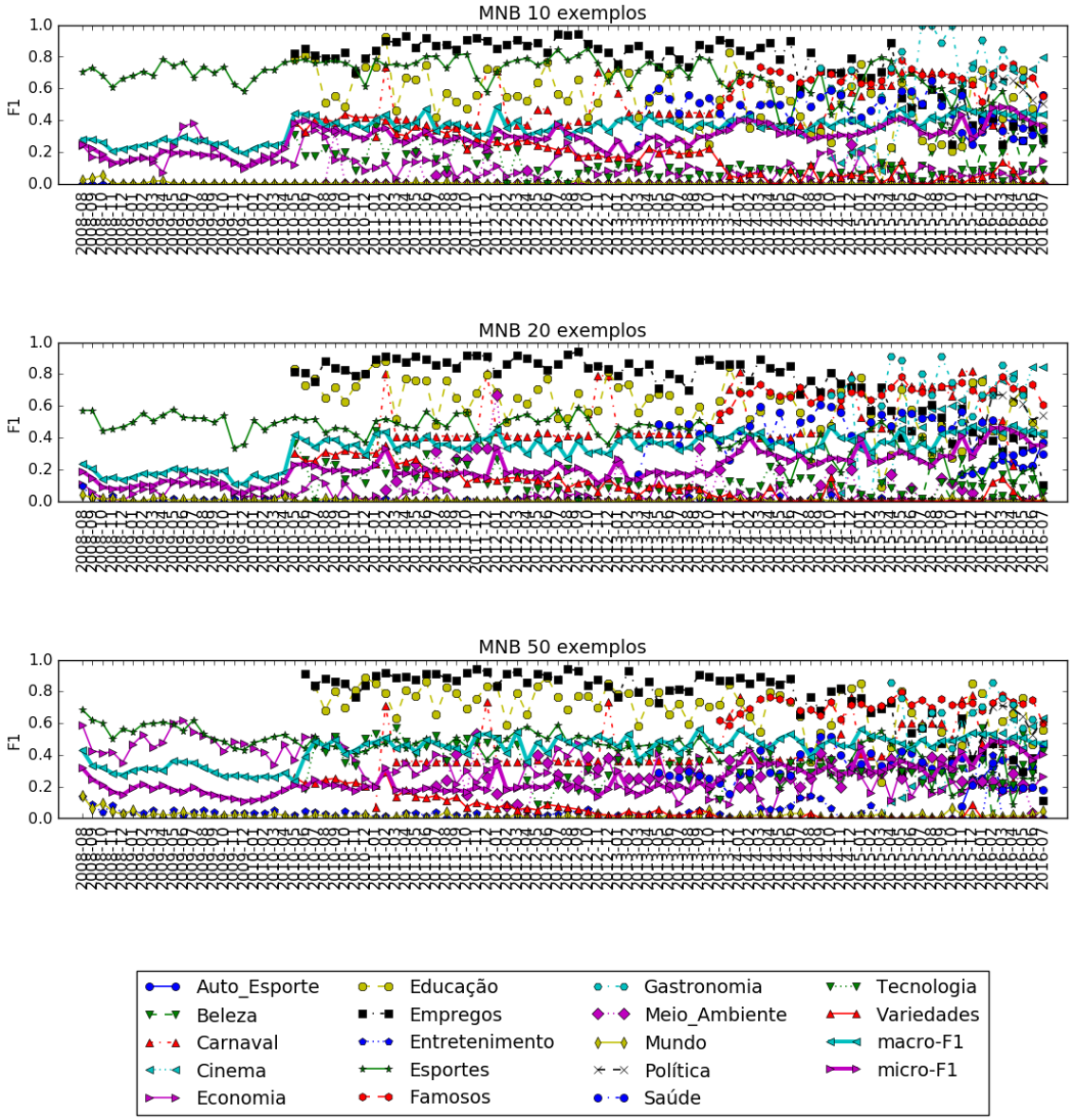

Figura 87. Resultado do algoritmo MNB em um conjunto com 10, 20 e 50 exemplos rotulados. Base: Notícias do Brasil (Variedades), período: Mensal.

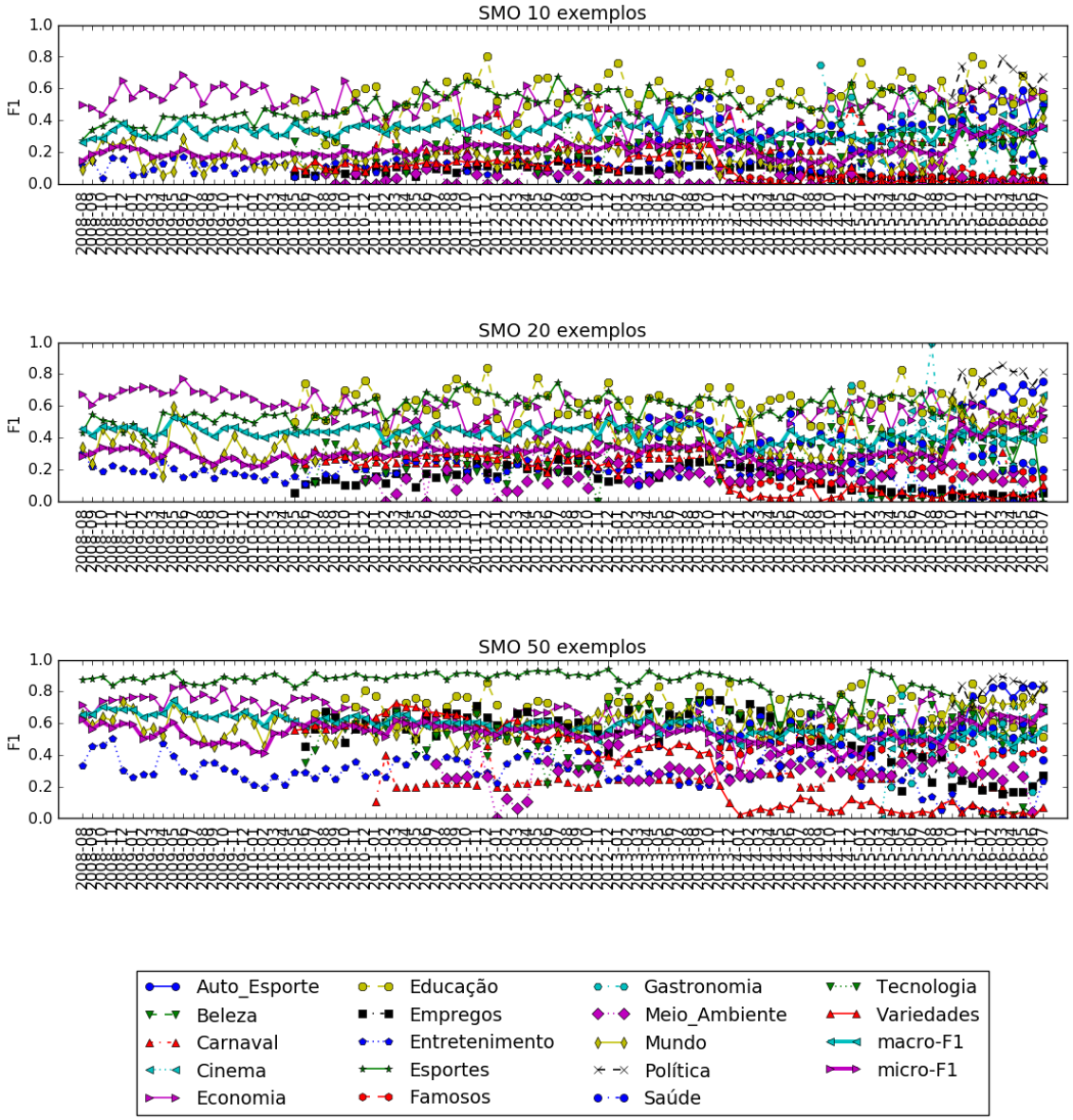

Figura 88. Resultado do algoritmo SMO em um conjunto com 10, 20 e 50 exemplos rotulados. Base: Notícias do Brasil (Variedades), período: Mensal.

4.4.3.4 Resultado: Base Notícias do Brasil (Estados)

Nessa seção são apresentados os resultados obtidos através dos testes executados na base Notícias do Brasil (Estados). Os experimentos foram executados considerando a configuração experimental apresentada anteriormente. Os resultados são apresentados na sequência de figuras a seguir no intervalo da Figura 89 até a Figura 117.

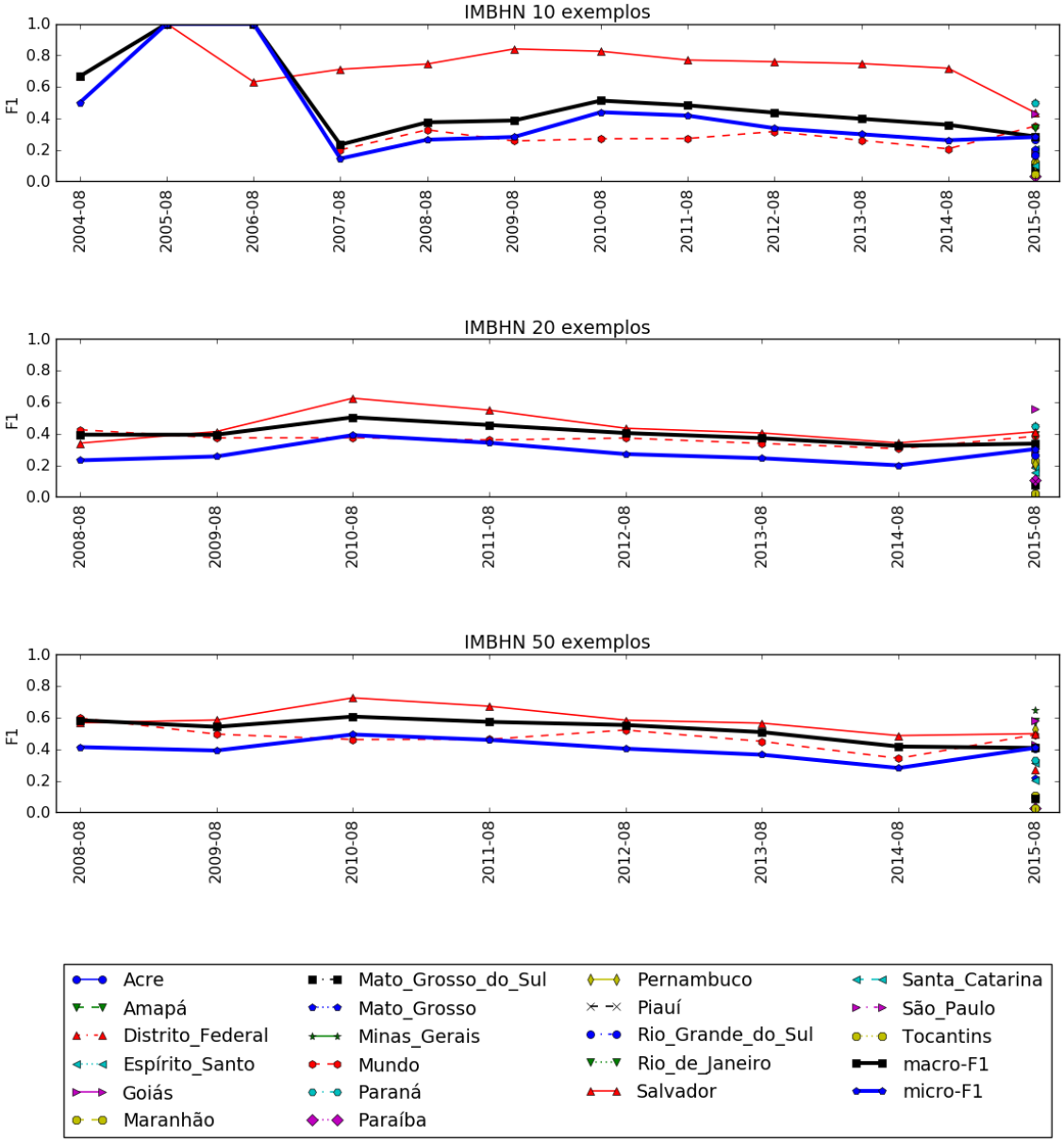

Figura 89. Resultado do algoritmo IMBHN em um conjunto com 10, 20 e 50 exemplos rotulados. Base: Notícias do Brasil (Estados), período: Anual.

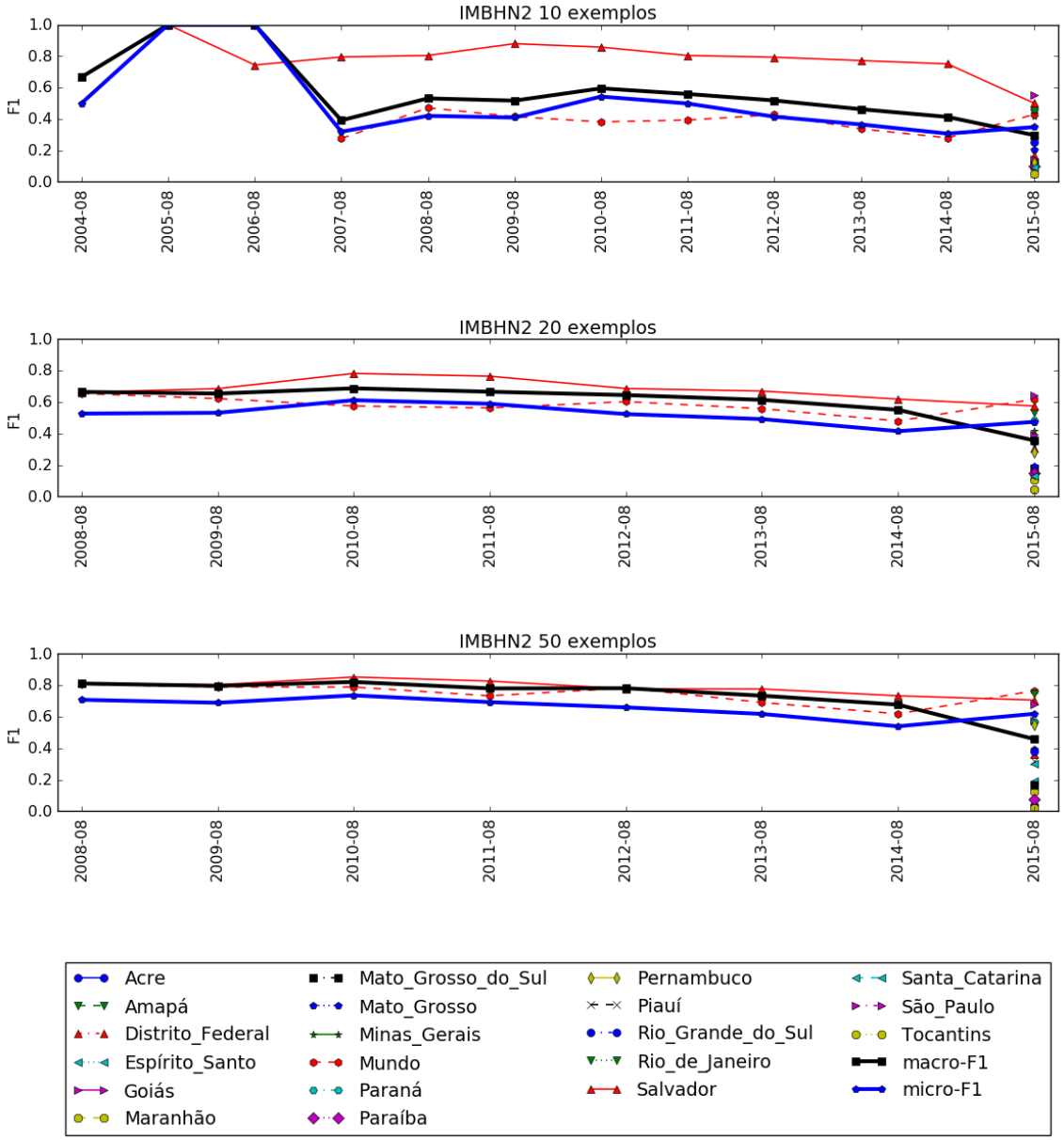

Figura 90. Resultado do algoritmo IMBHN2 em um conjunto com 10, 20 e 50 exemplos rotulados. Base: Notícias do Brasil (Estados), período: Anual.

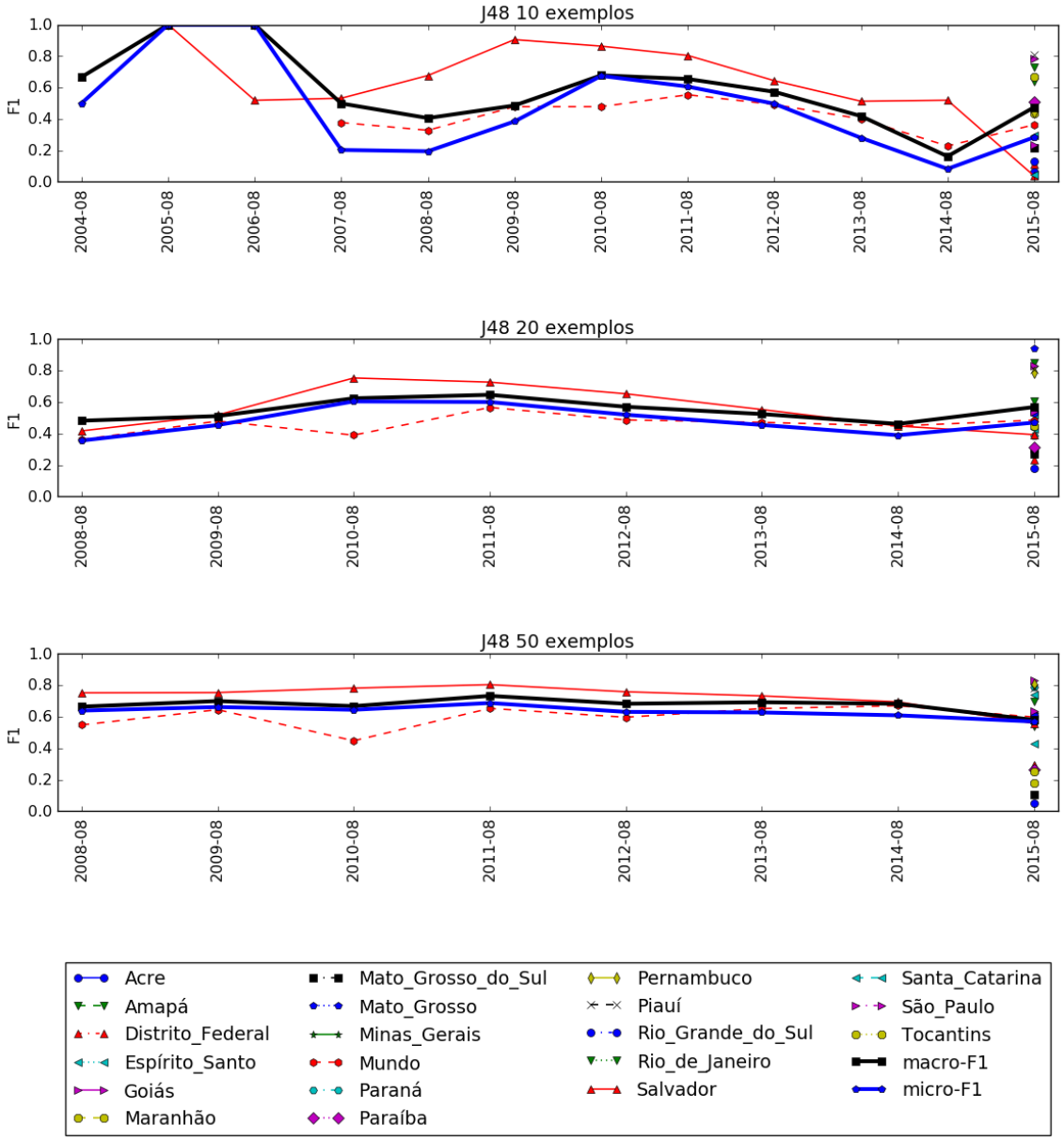

Figura 91. Resultado do algoritmo J48 em um conjunto com 10, 20 e 50 exemplos rotulados. Base: Notícias do Brasil (Estados), período: Anual.

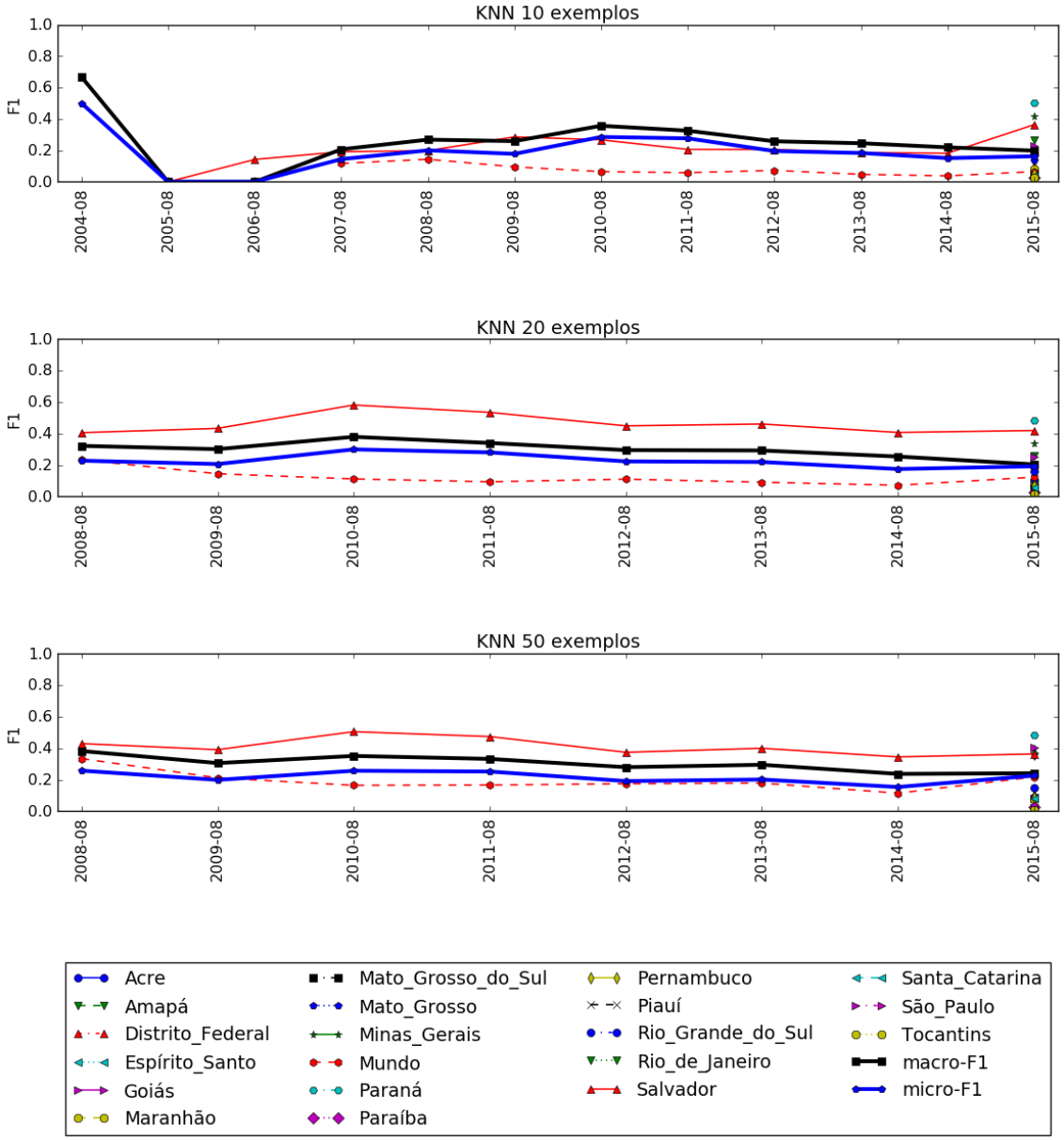

Figura 92. Resultado do algoritmo KNN em um conjunto com 10, 20 e 50 exemplos rotulados. Base: Notícias do Brasil (Estados), período: Anual.

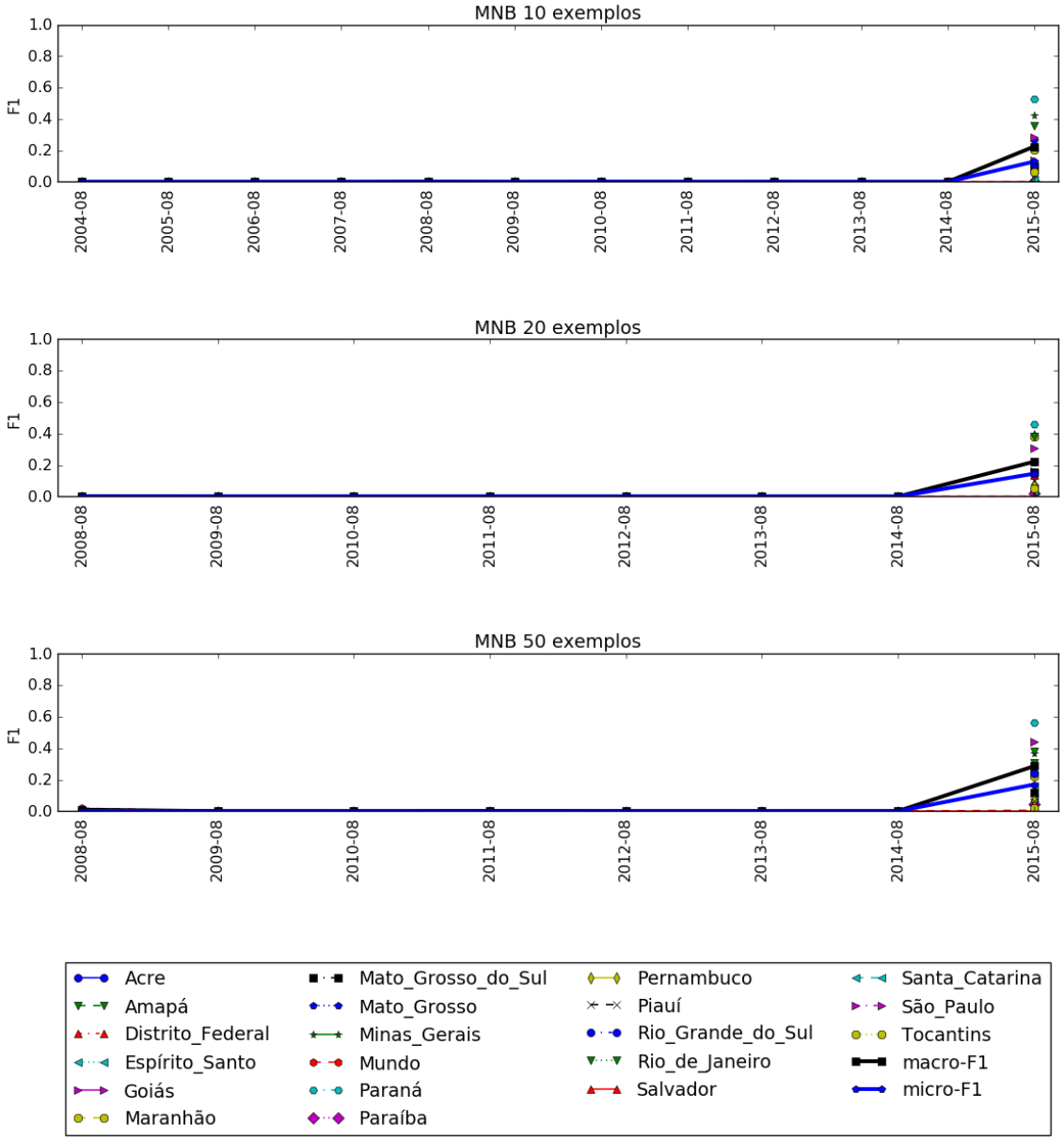

Figura 93. Resultado do algoritmo MNB em um conjunto com 10, 20 e 50 exemplos rotulados. Base: Notícias do Brasil (Estados), período: Anual.

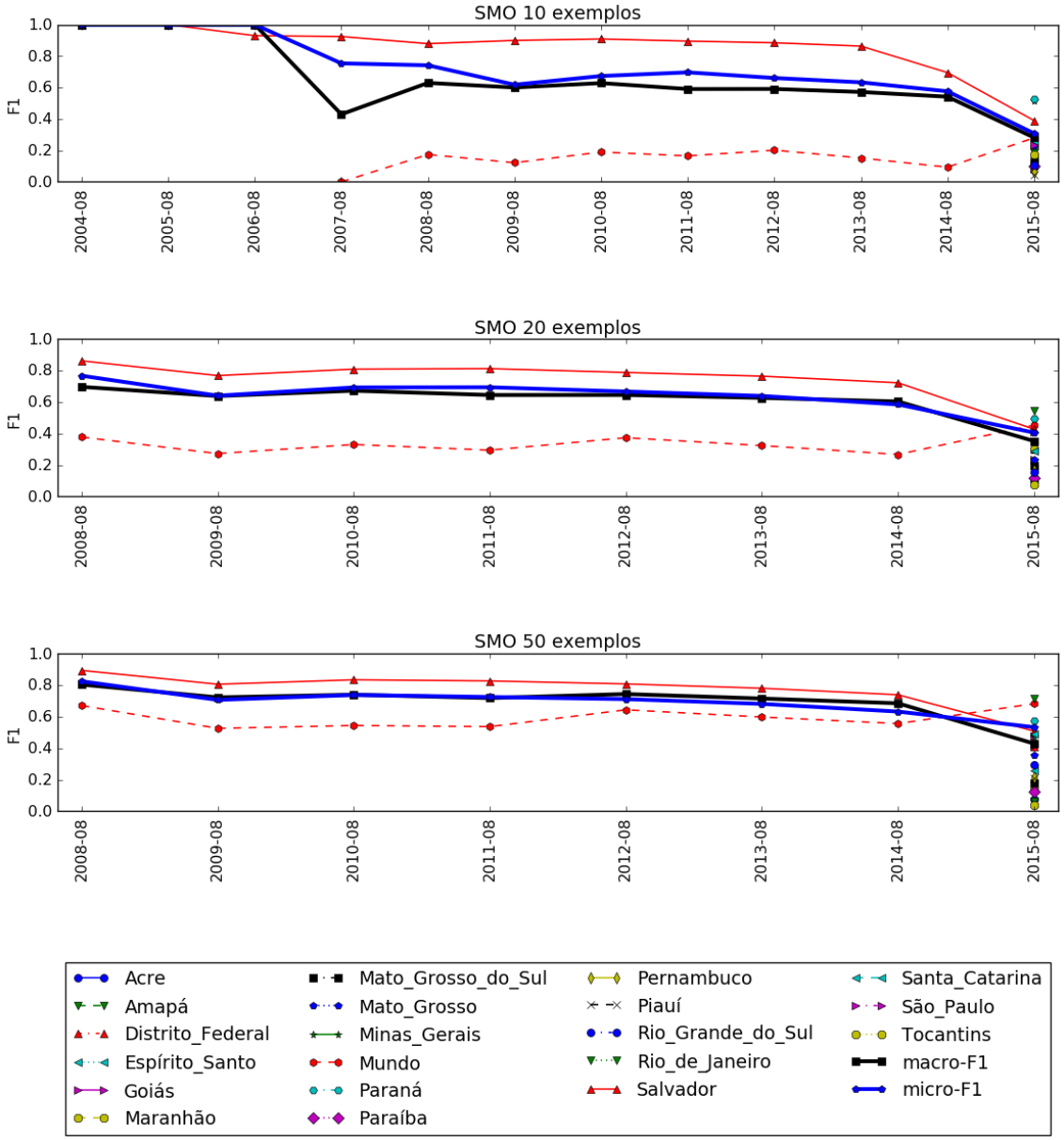

Figura 94. Resultado do algoritmo SMO em um conjunto com 10, 20 e 50 exemplos rotulados. Base: Notícias do Brasil (Estados), período: Anual.

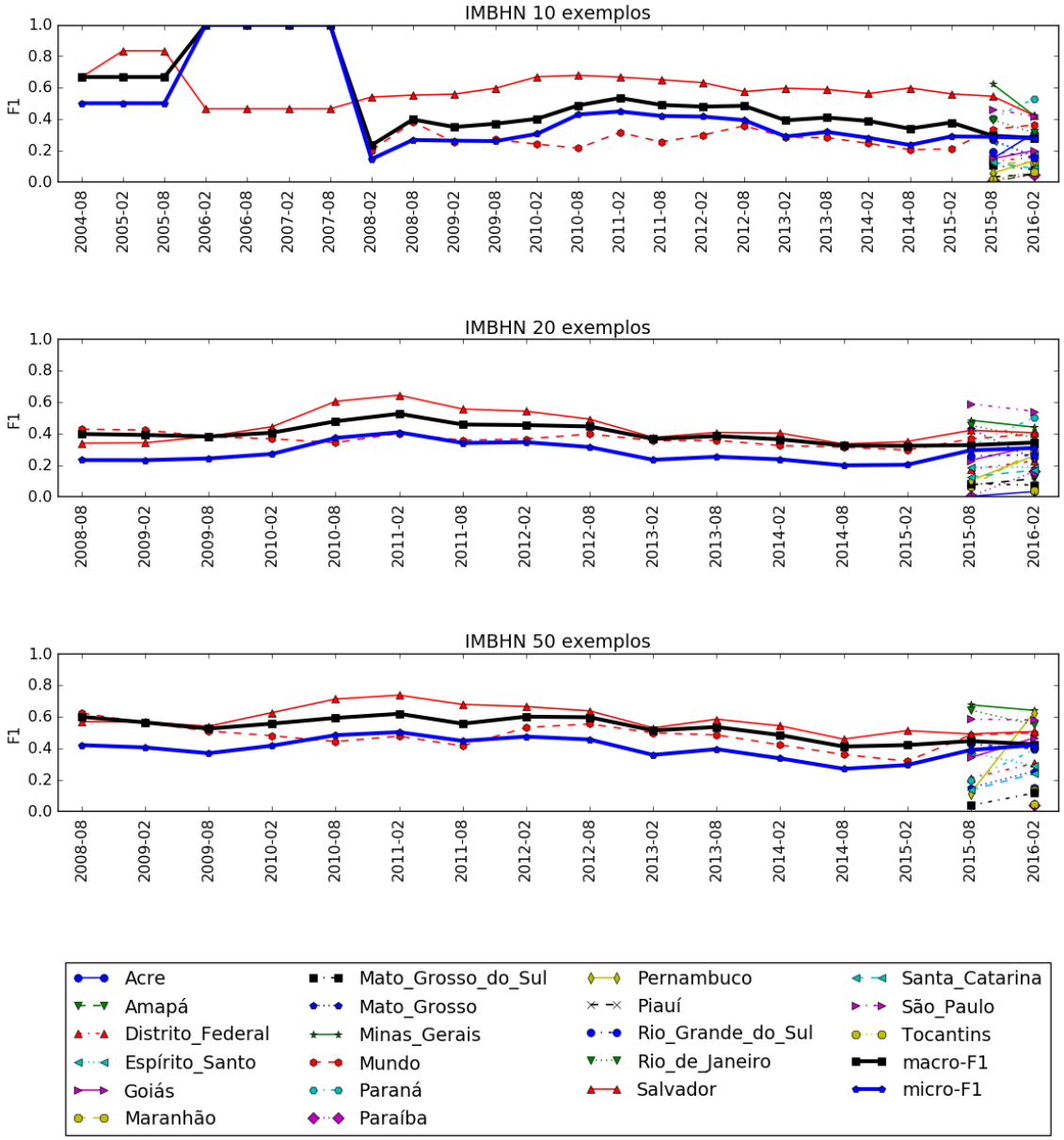

Figura 95. Resultado do algoritmo IMBHN em um conjunto com 10, 20 e 50 exemplos rotulados. Base: Notícias do Brasil (Estados), período: Semestral.

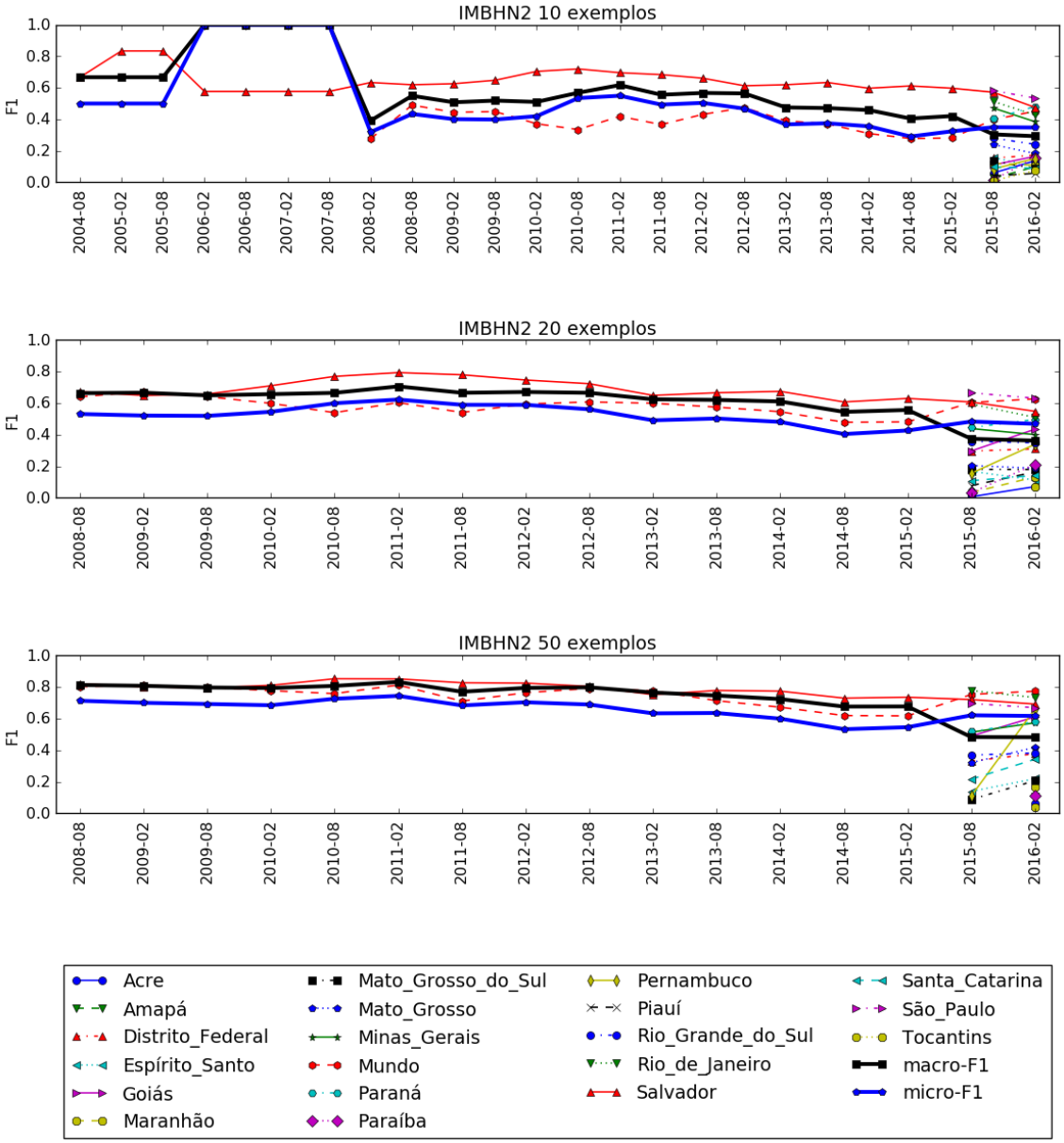

Figura 96. Resultado do algoritmo IMBHN2 em um conjunto com 10, 20 e 50 exemplos rotulados. Base: Notícias do Brasil (Estados), período: Semestral.

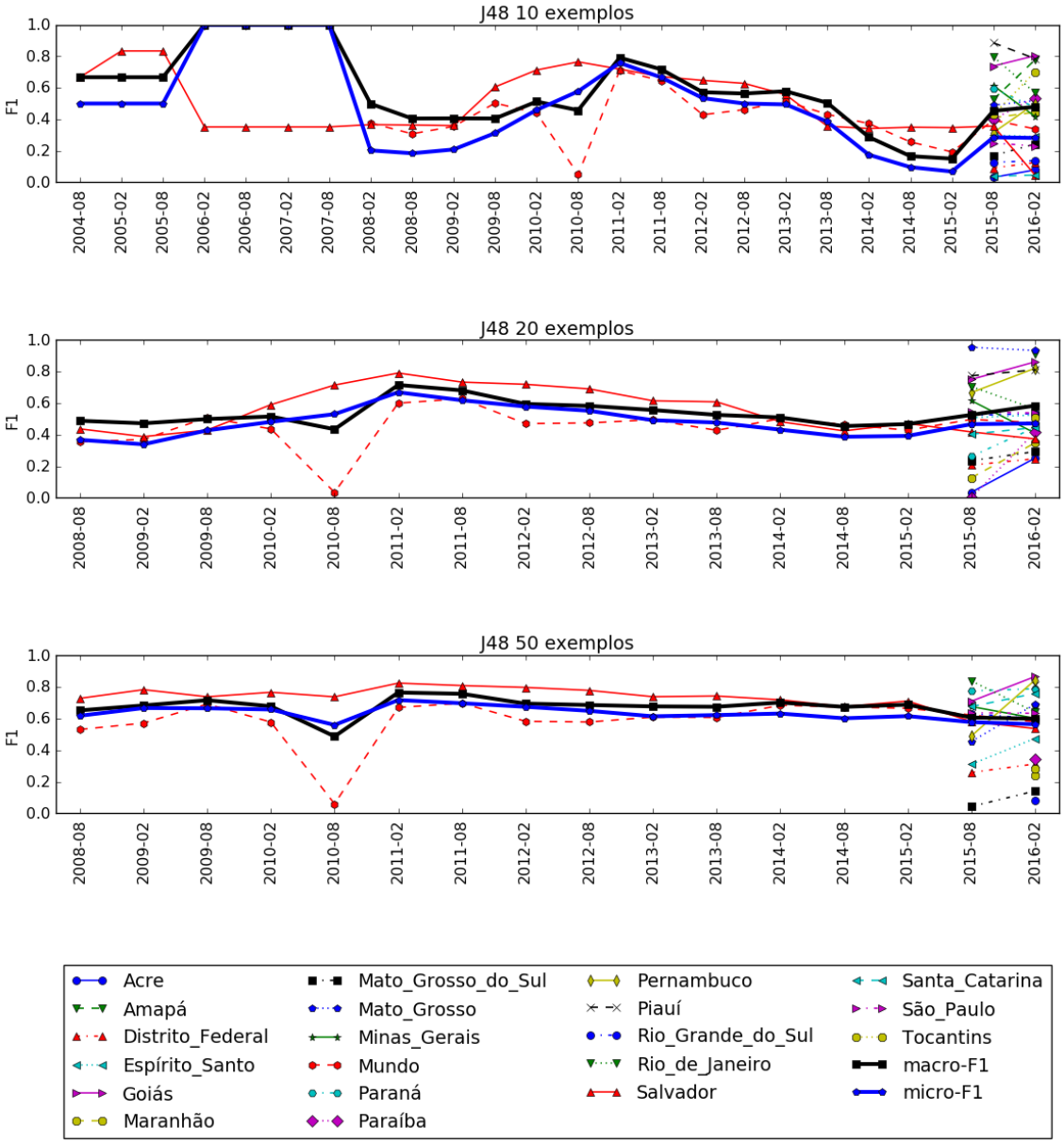

Figura 97. Resultado do algoritmo J48 em um conjunto com 10, 20 e 50 exemplos rotulados. Base: Notícias do Brasil (Estados), período: Semestral.

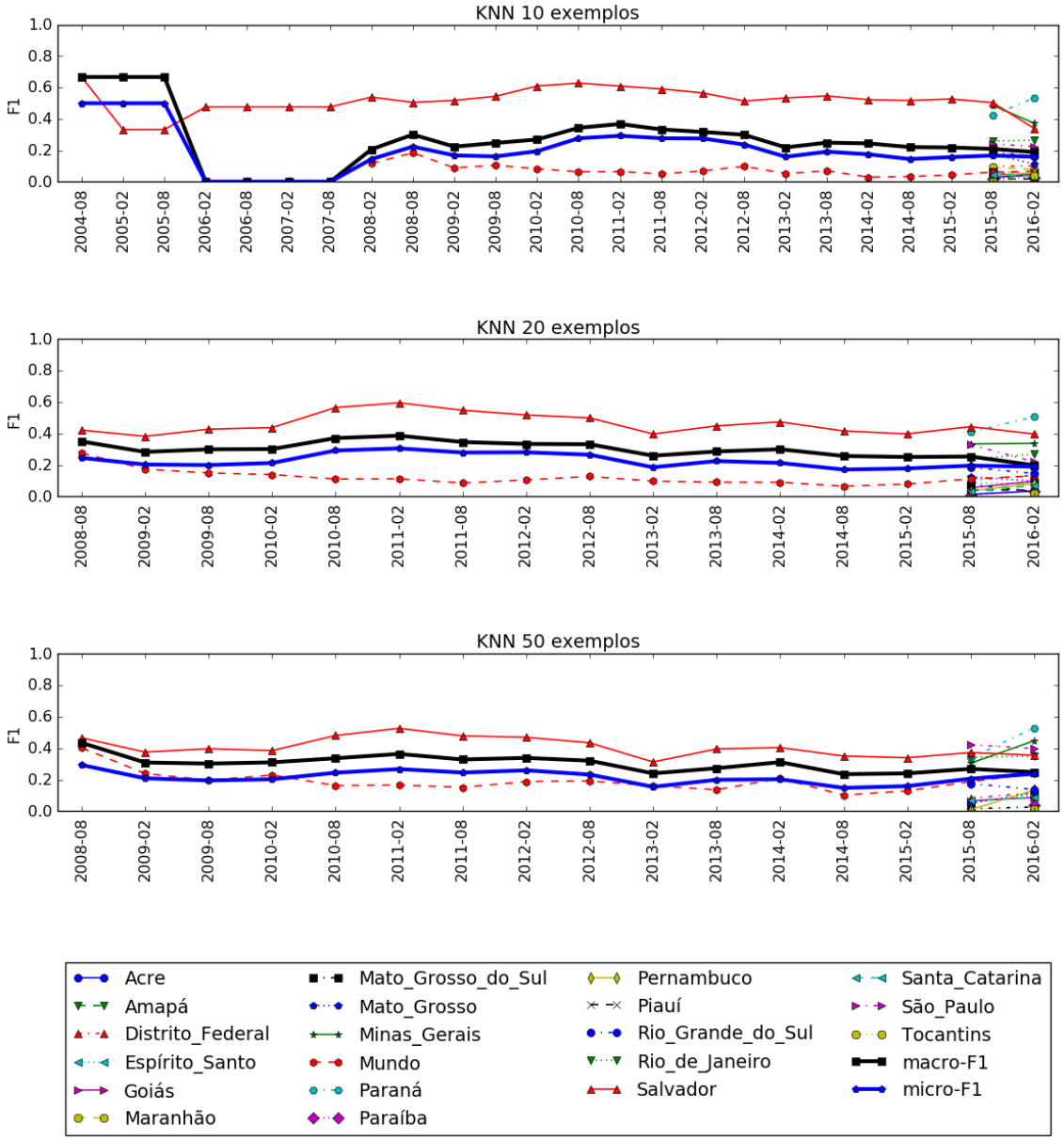

Figura 98. Resultado do algoritmo KNN em um conjunto com 10, 20 e 50 exemplos rotulados. Base: Notícias do Brasil (Estados), período: Semestral.

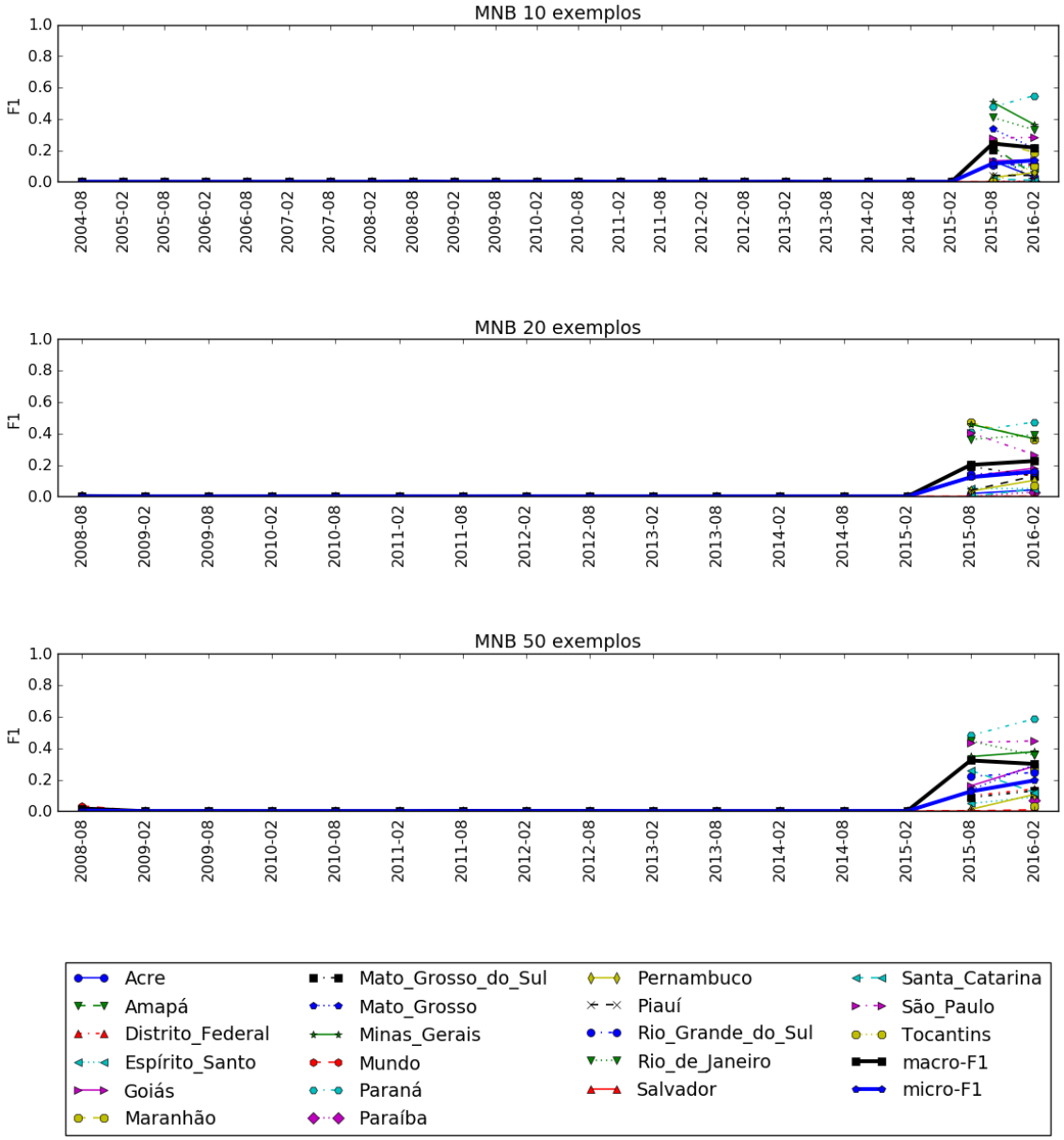

Figura 99. Resultado do algoritmo MNB em um conjunto com 10, 20 e 50 exemplos rotulados. Base: Notícias do Brasil (Estados), período: Semestral.

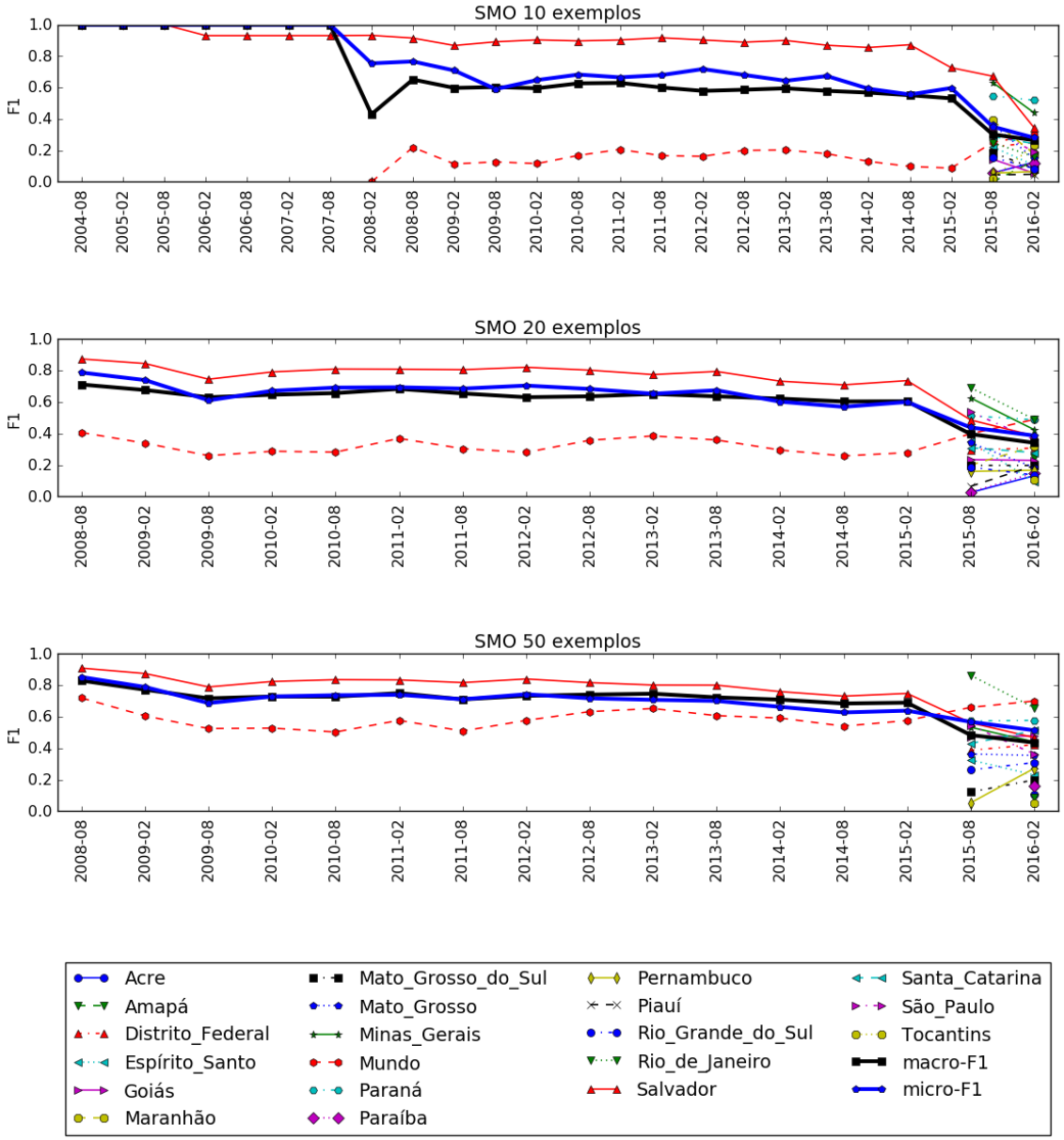

Figura 100. Resultado do algoritmo SMO em um conjunto com 10, 20 e 50 exemplos rotulados. Base: Notícias do Brasil (Estados), período: Semestral.

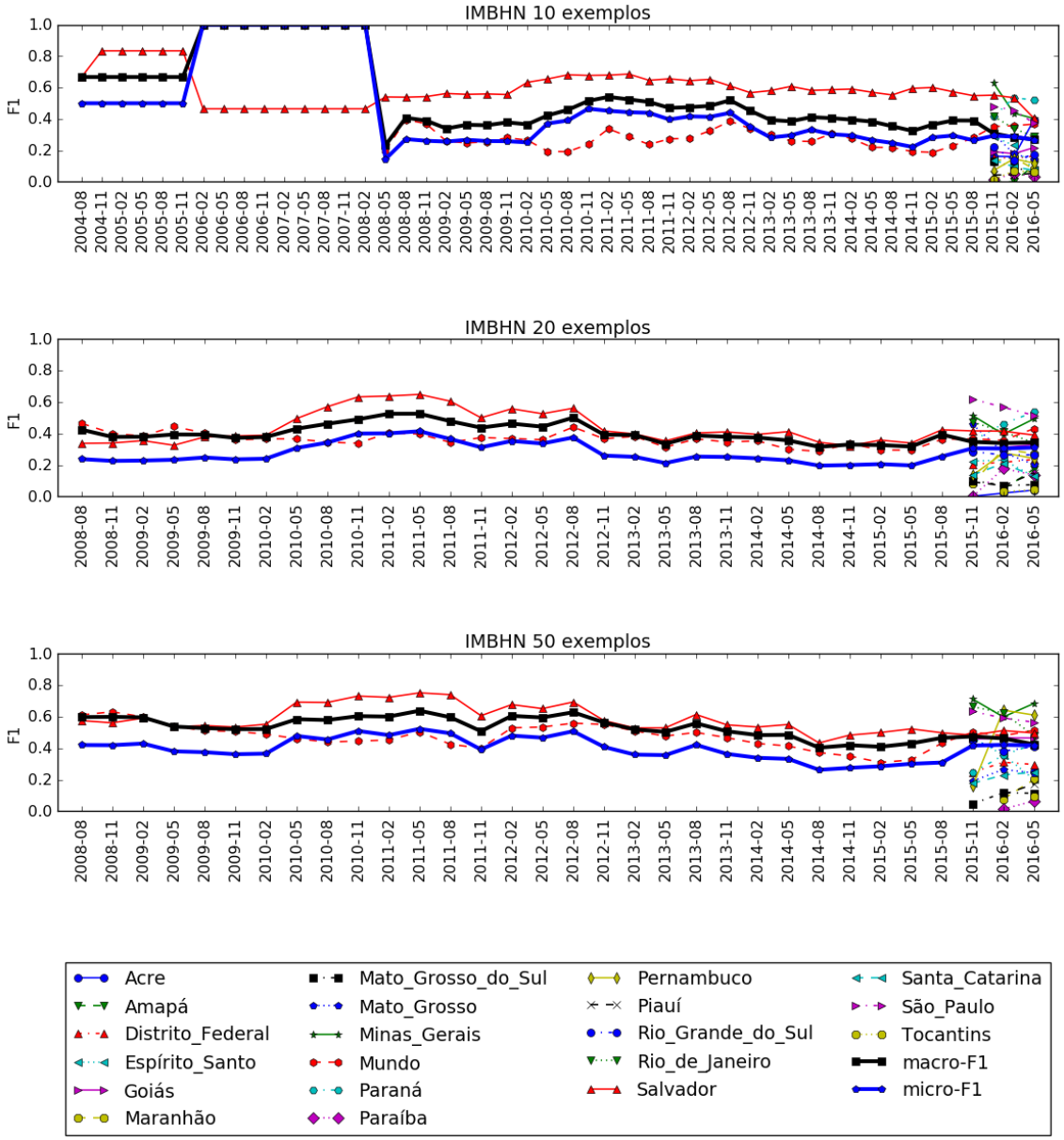

Figura 101. Resultado do algoritmo IMBHN em um conjunto com 10, 20 e 50 exemplos rotulados. Base: Notícias do Brasil (Estados), período: Trimestral.

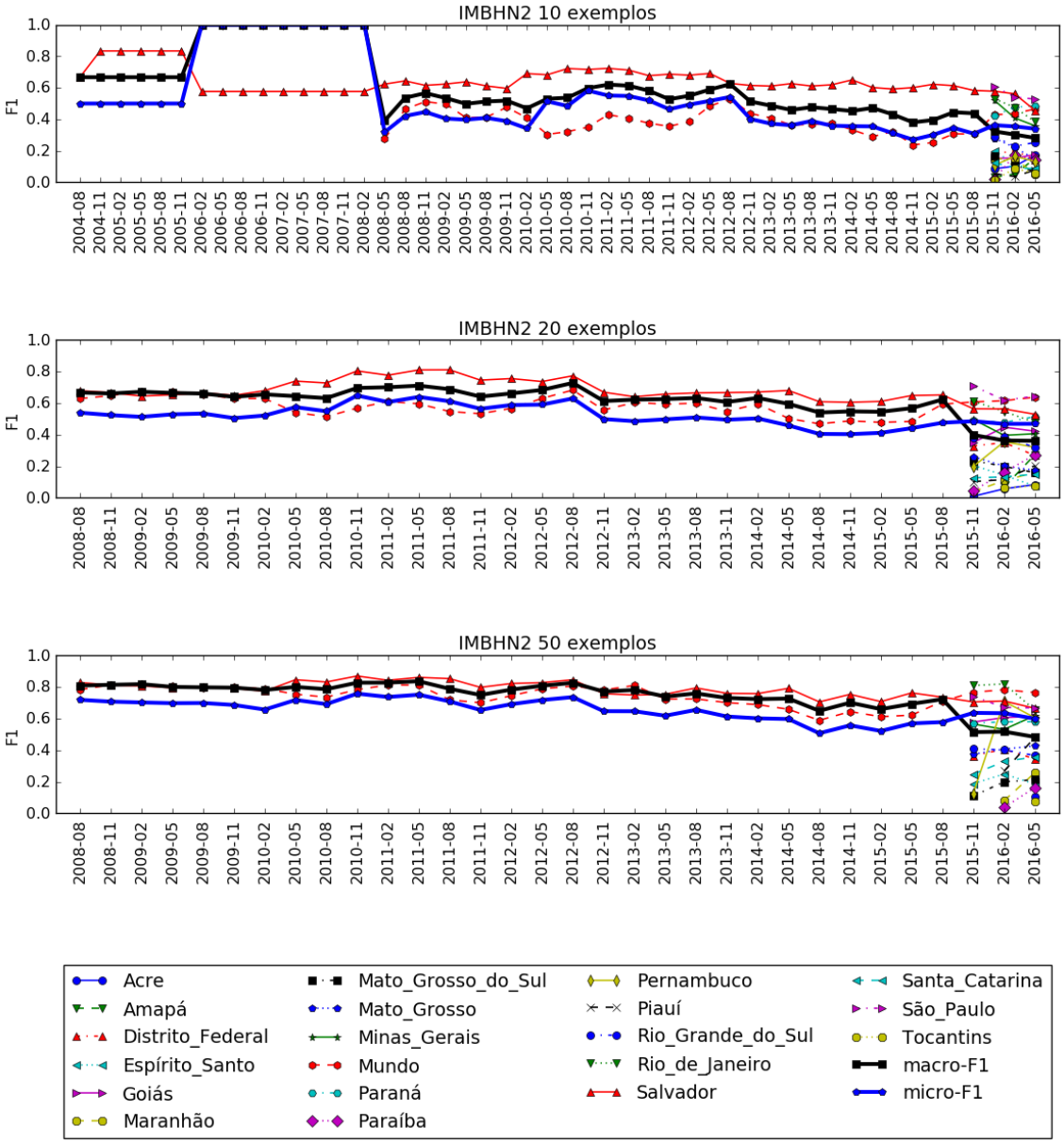

Figura 102. Resultado do algoritmo IMBHN2 em um conjunto com 10, 20 e 50 exemplos rotulados. Base: Notícias do Brasil (Estados), período: Trimestral.

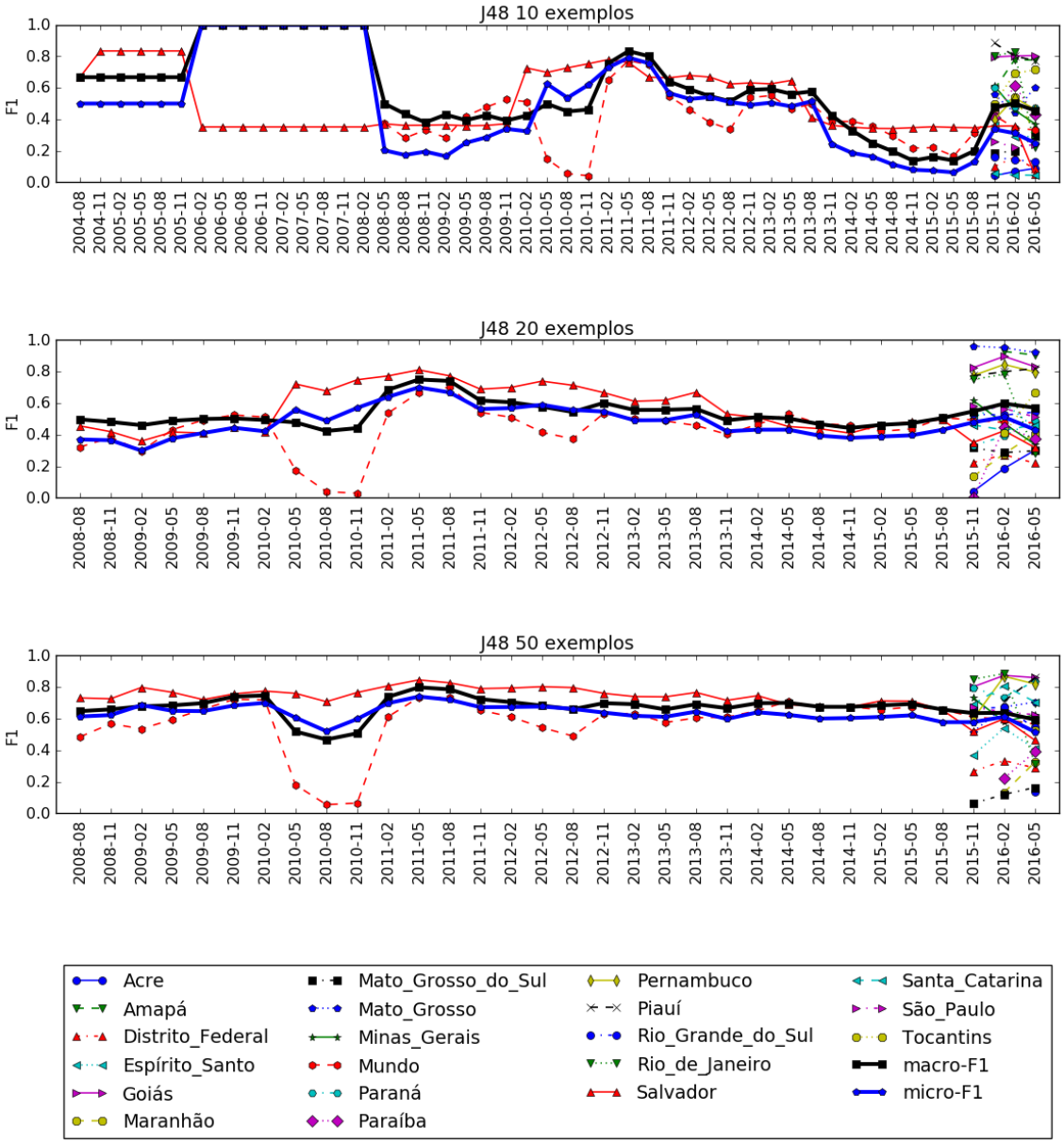

Figura 103. Resultado do algoritmo J48 em um conjunto com 10, 20 e 50 exemplos rotulados. Base: Notícias do Brasil (Estados), período: Trimestral.

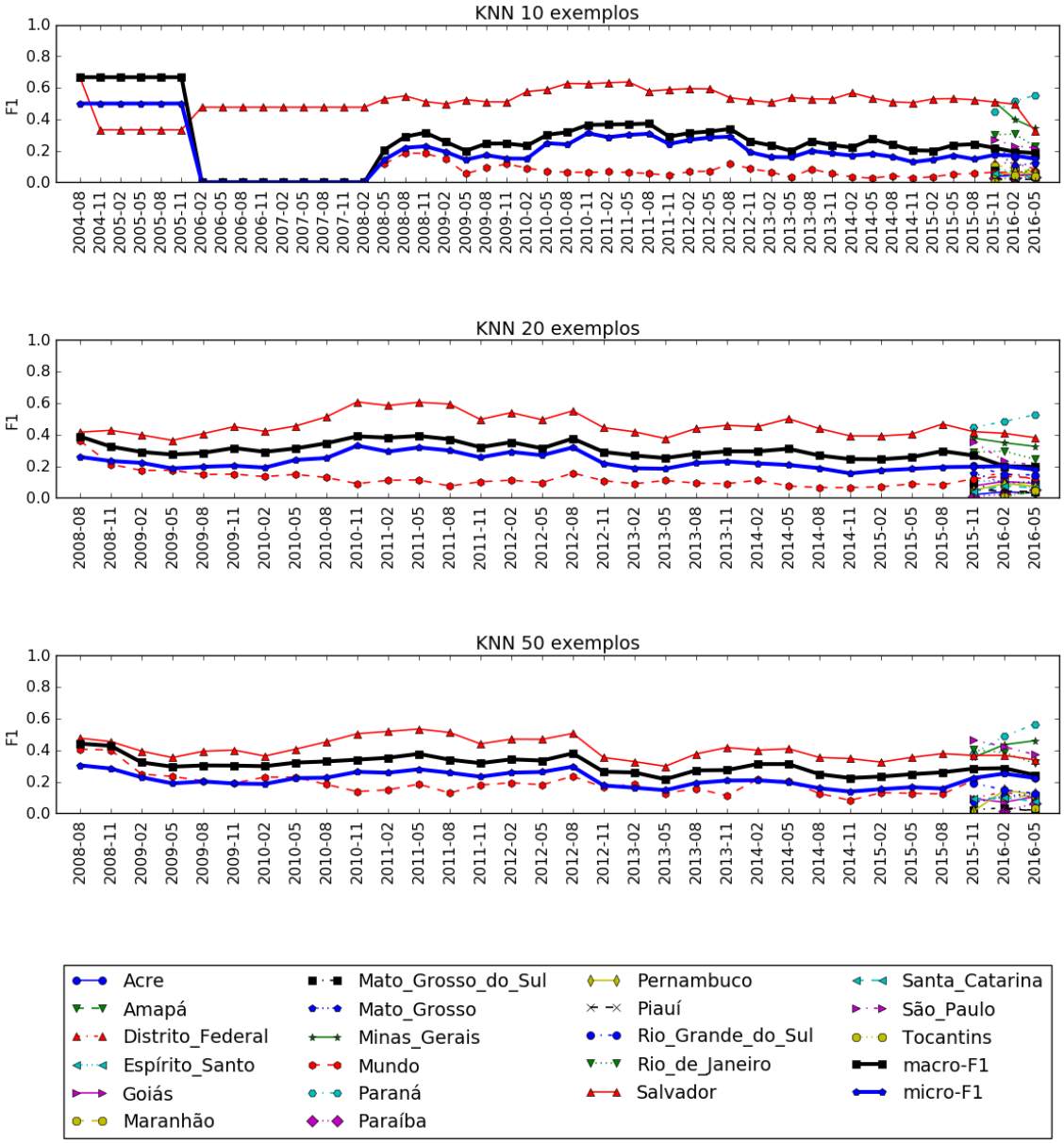

Figura 104. Resultado do algoritmo KNN em um conjunto com 10, 20 e 50 exemplos rotulados. Base: Notícias do Brasil (Estados), período: Trimestral.

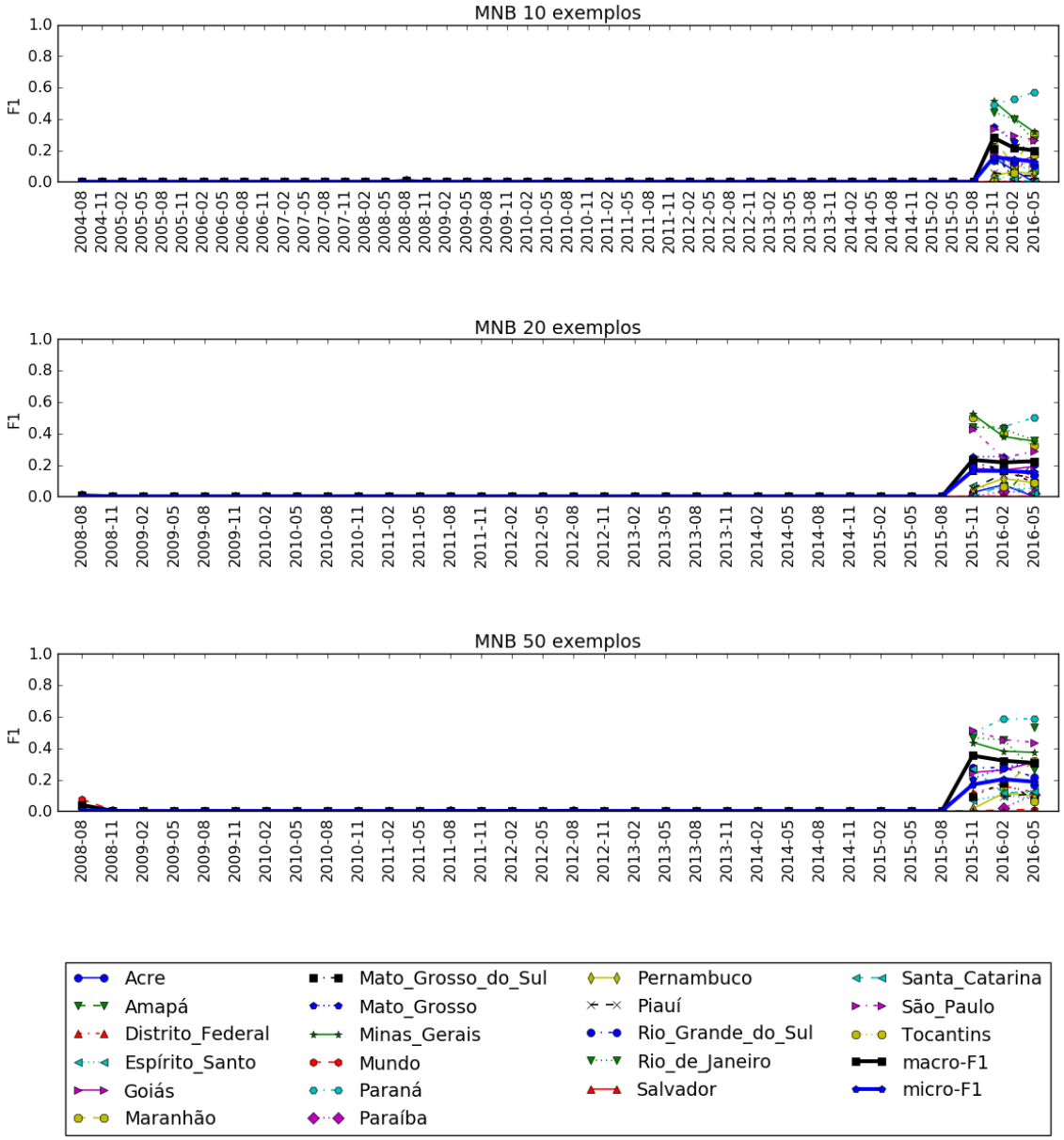

Figura 105. Resultado do algoritmo MNB em um conjunto com 10, 20 e 50 exemplos rotulados. Base: Notícias do Brasil (Estados), período: Trimestral.

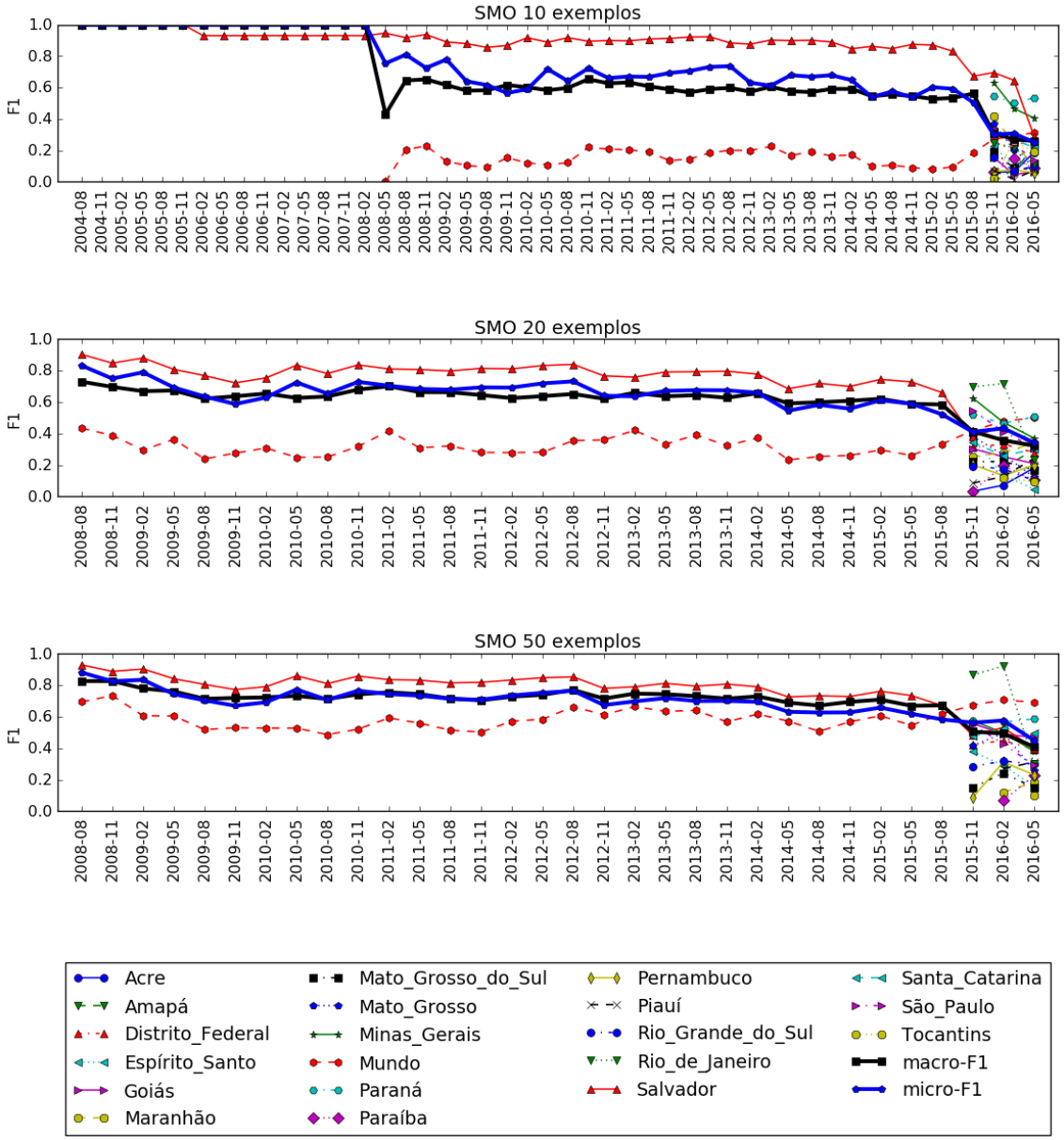

Figura 106. Resultado do algoritmo SMO em um conjunto com 10, 20 e 50 exemplos rotulados. Base: Notícias do Brasil (Estados), período: Trimestral.

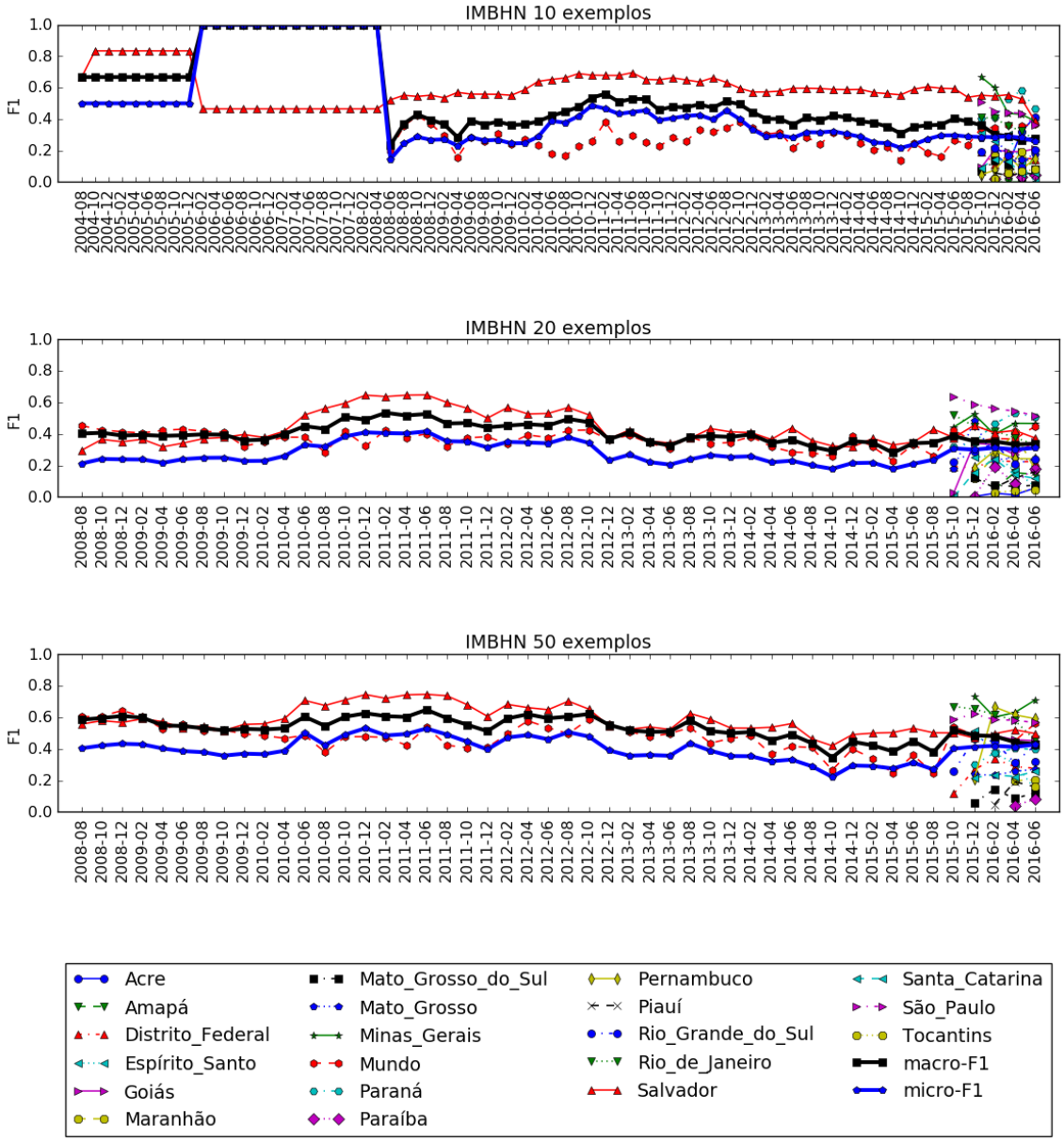

Figura 107. Resultado do algoritmo IMBHN em um conjunto com 10, 20 e 50 exemplos rotulados. Base: Notícias do Brasil (Estados), período: Bimestral.

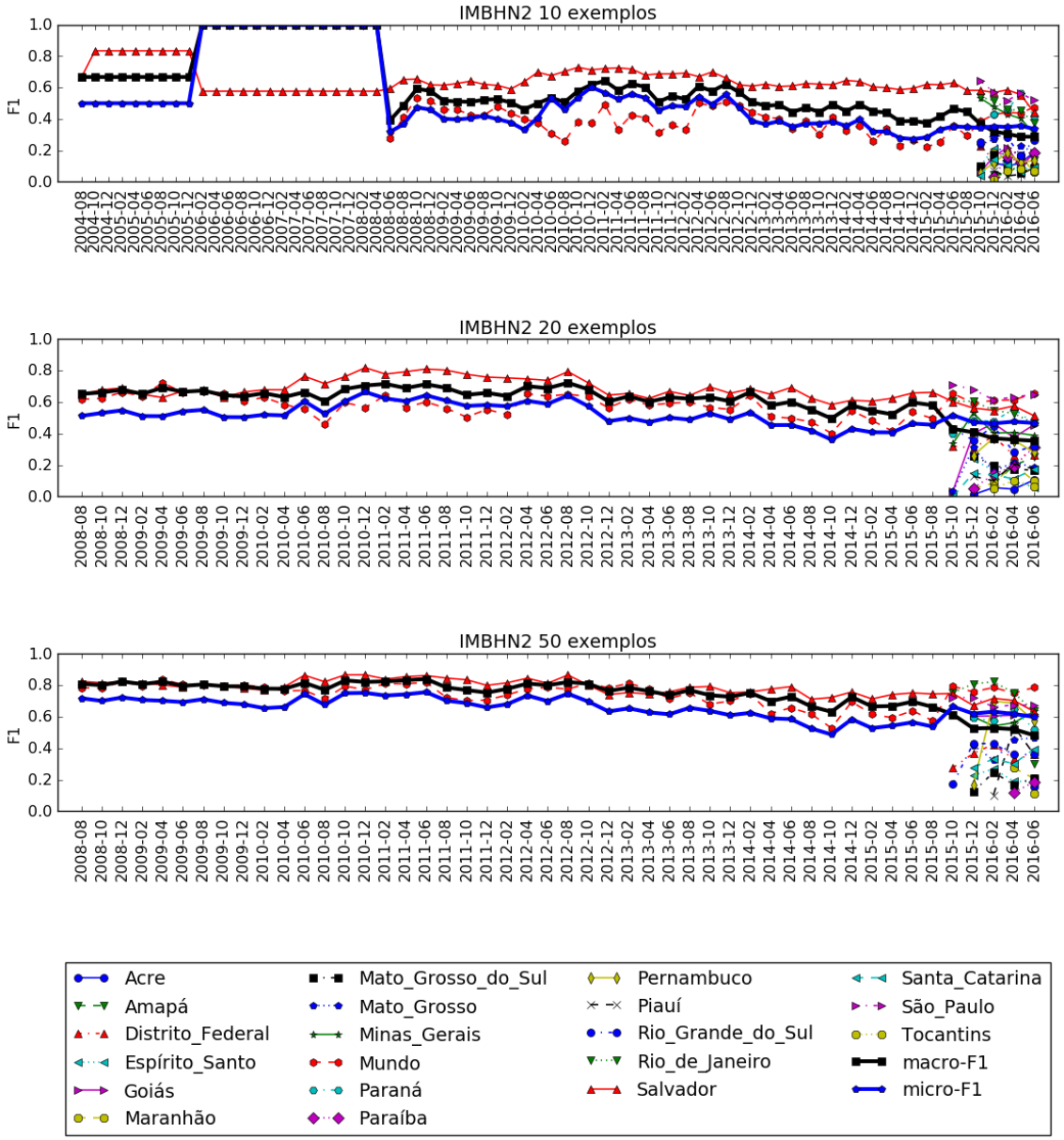

Figura 108. Resultado do algoritmo IMBHN2 em um conjunto com 10, 20 e 50 exemplos rotulados. Base: Notícias do Brasil (Estados), período: Bimestral.

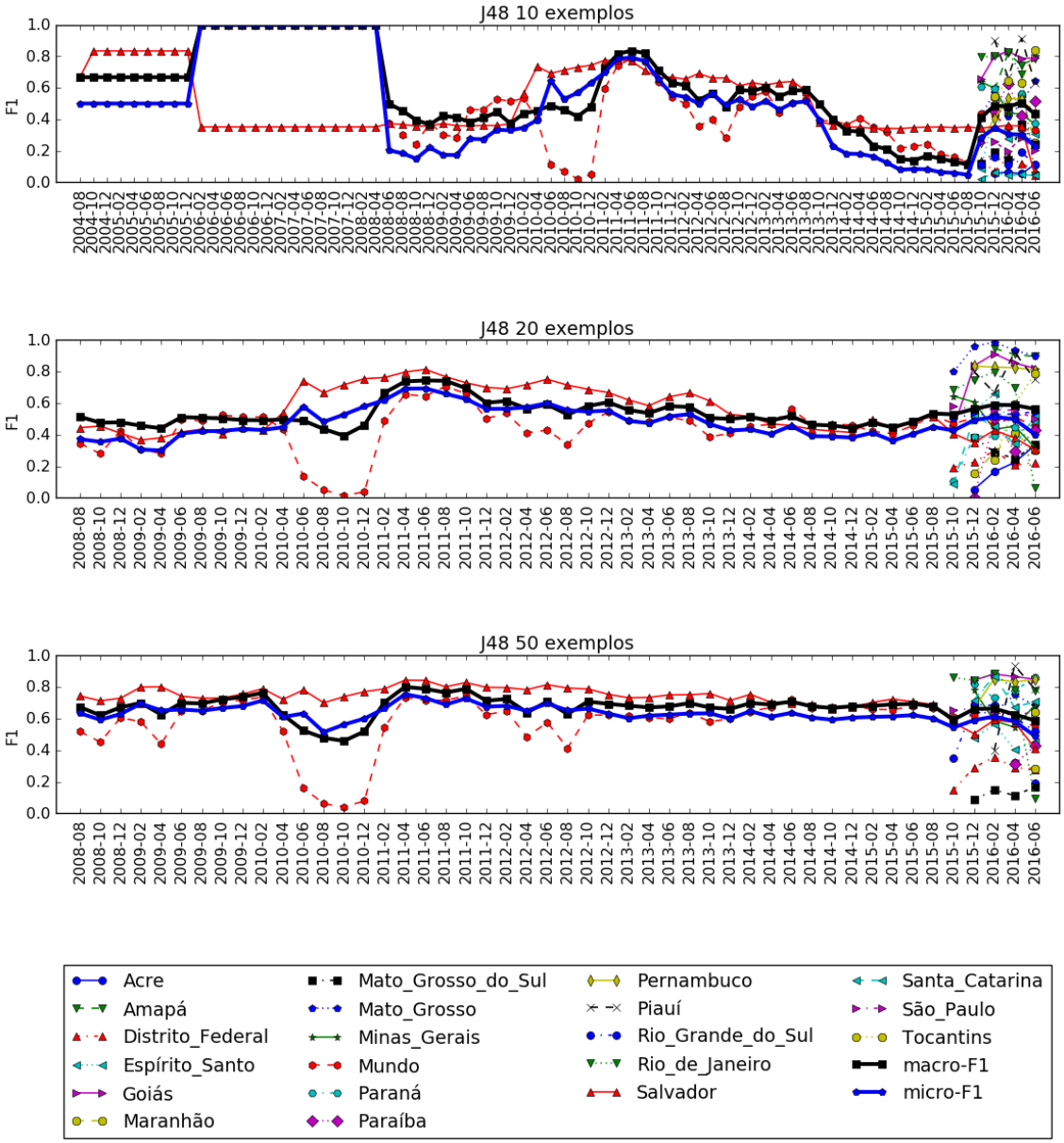

Figura 109. Resultado do algoritmo J48 em um conjunto com 10, 20 e 50 exemplos rotulados. Base: Notícias do Brasil (Estados), período: Bimestral.

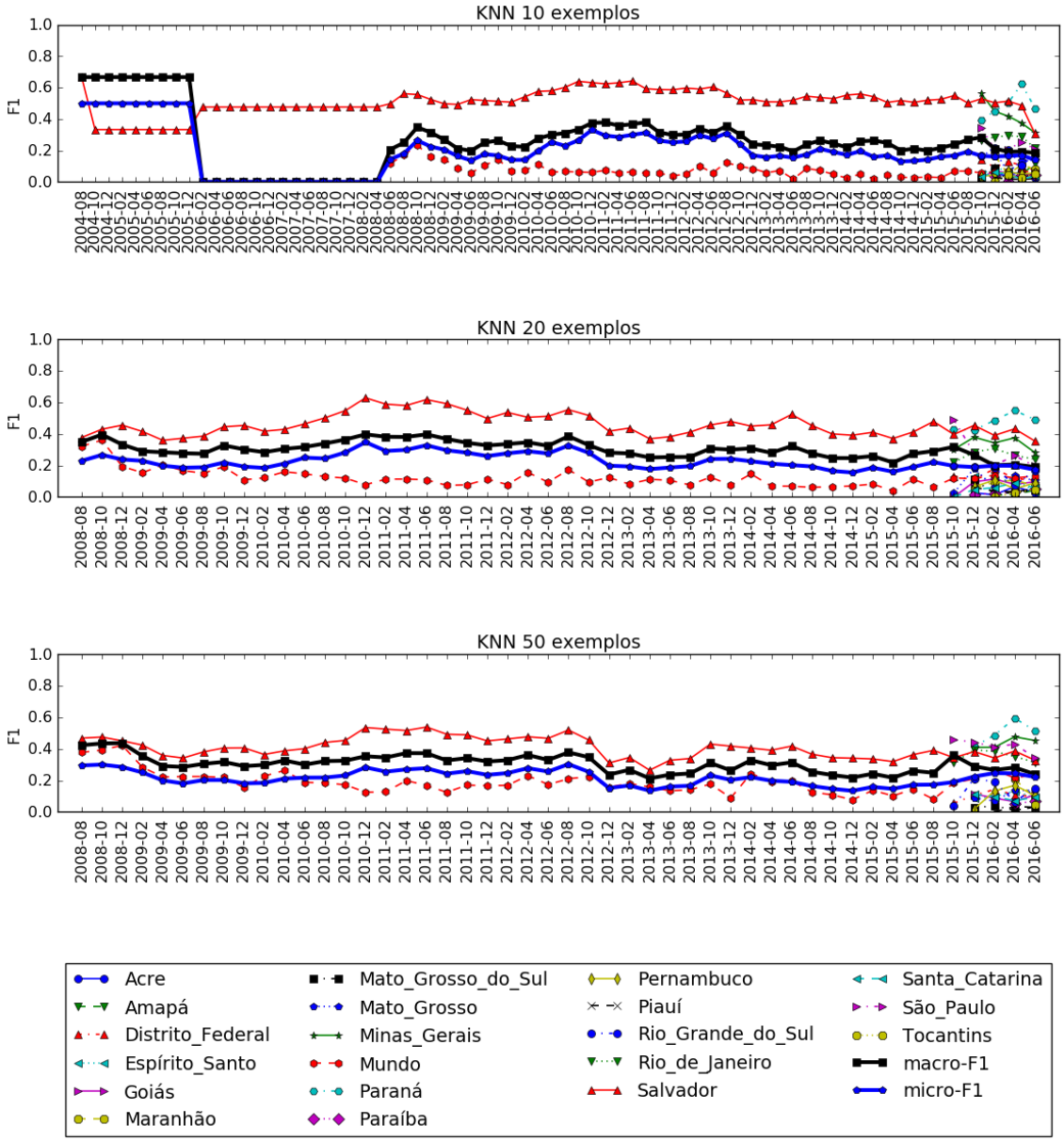

Figura 110. Resultado do algoritmo KNN em um conjunto com 10, 20 e 50 exemplos rotulados. Base: Notícias do Brasil (Estados), período: Bimestral.

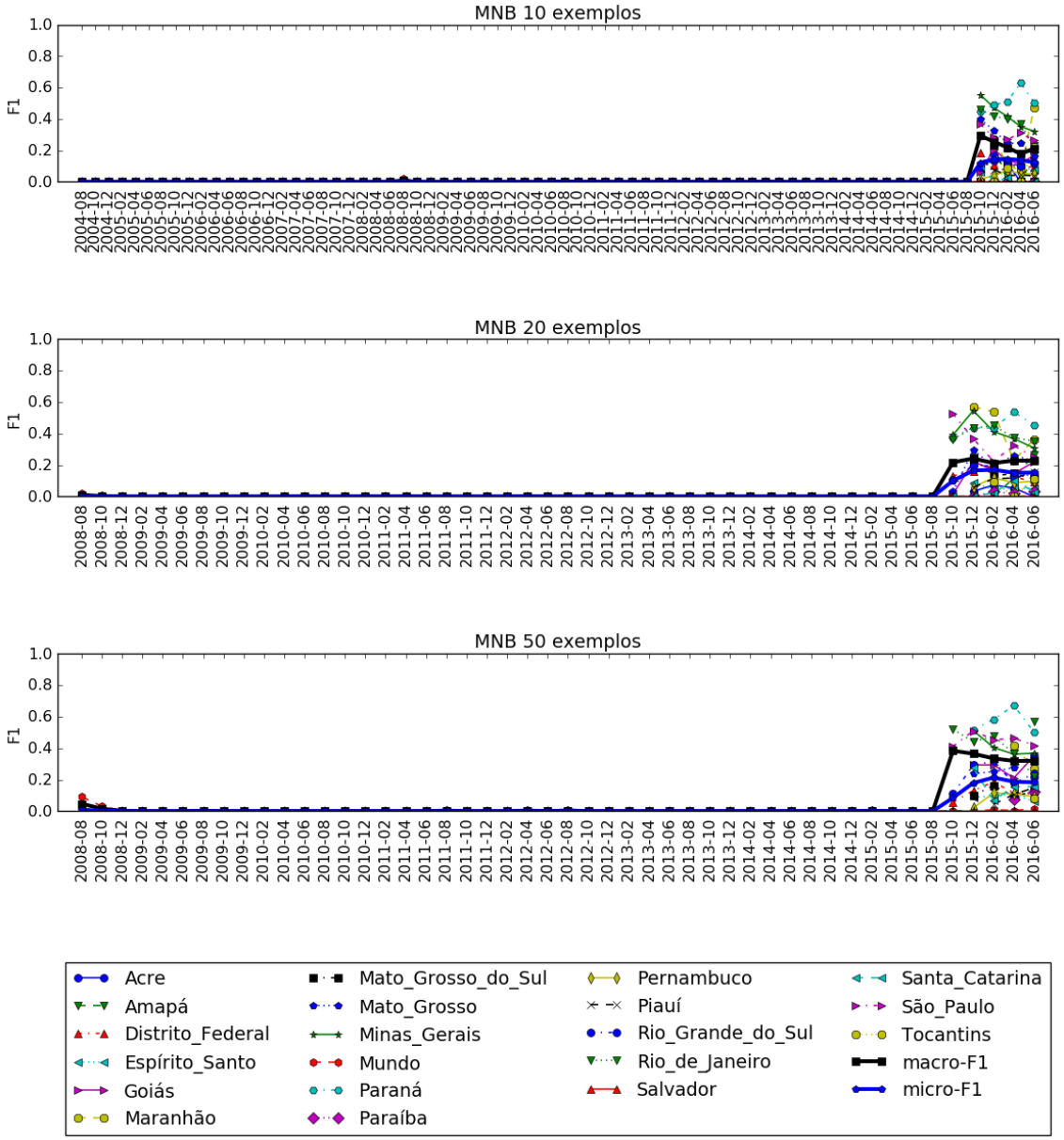

Figura 111. Resultado do algoritmo MNB em um conjunto com 10, 20 e 50 exemplos rotulados. Base: Notícias do Brasil (Estados), período: Bimestral.

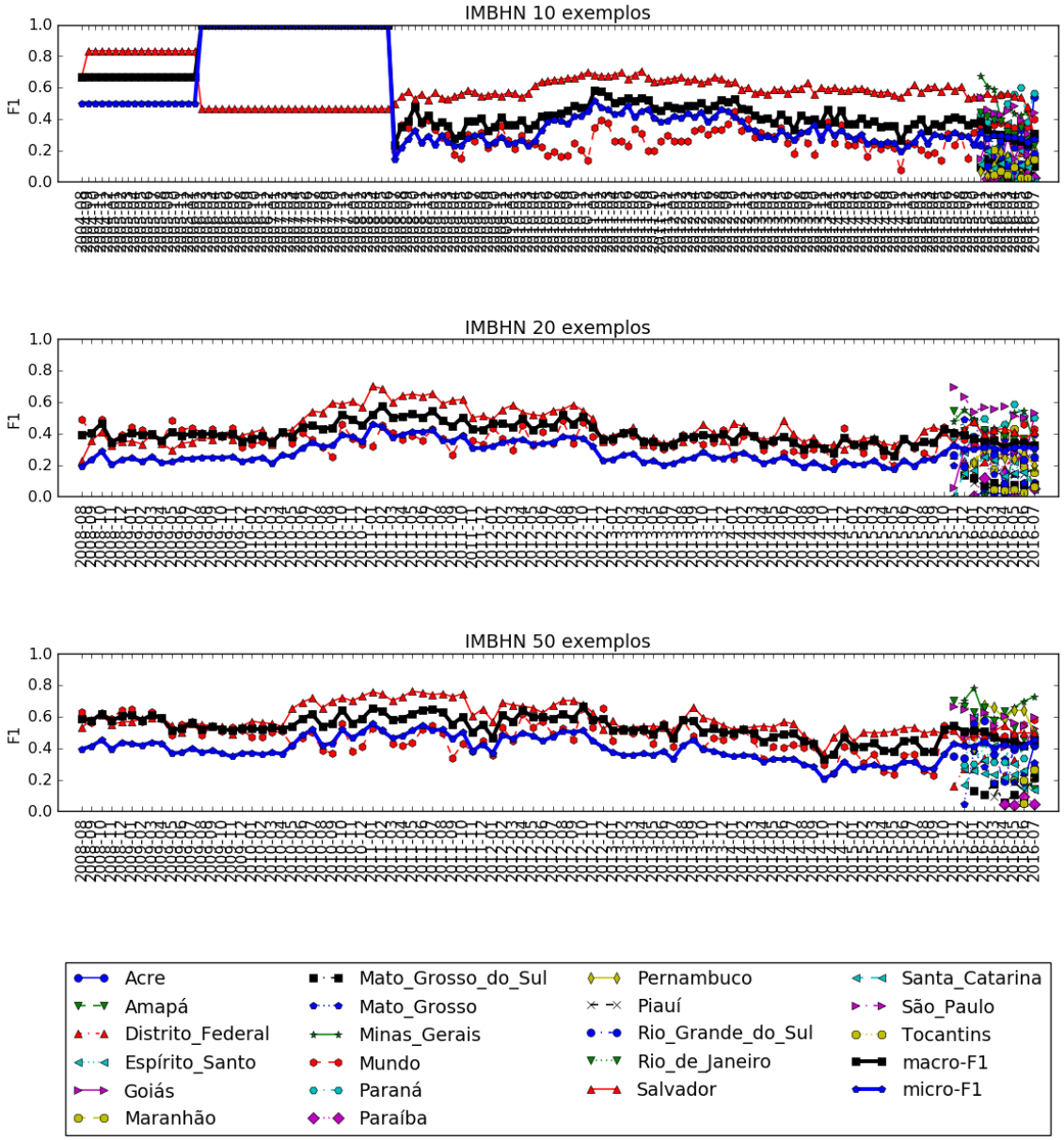

Figura 112. Resultado do algoritmo IMBHN em um conjunto com 10, 20 e 50 exemplos rotulados. Base: Notícias do Brasil (Estados), período: Mensal.

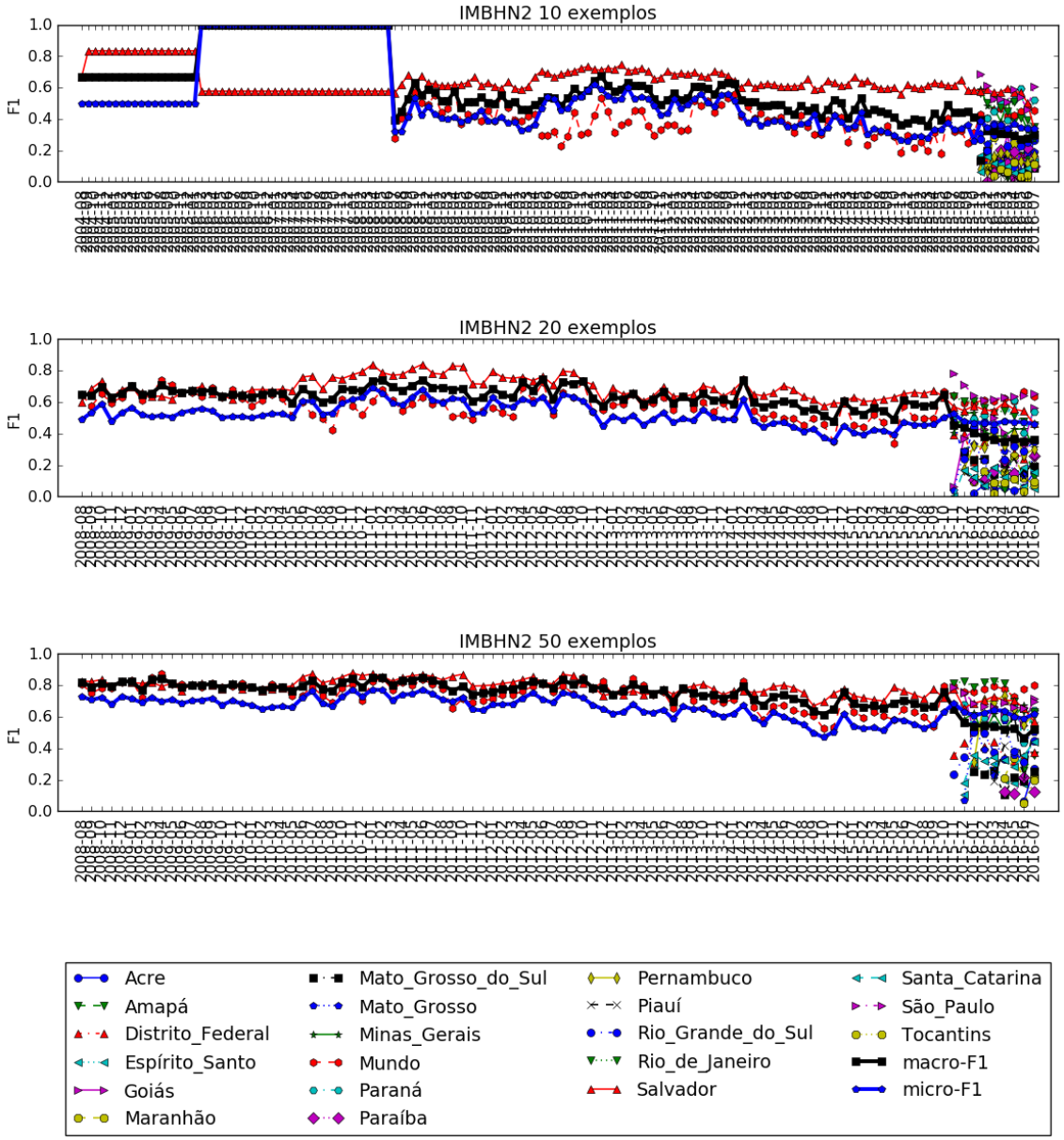

Figura 113. Resultado do algoritmo IMBHN2 em um conjunto com 10, 20 e 50 exemplos rotulados. Base: Notícias do Brasil (Estados), período: Mensal.

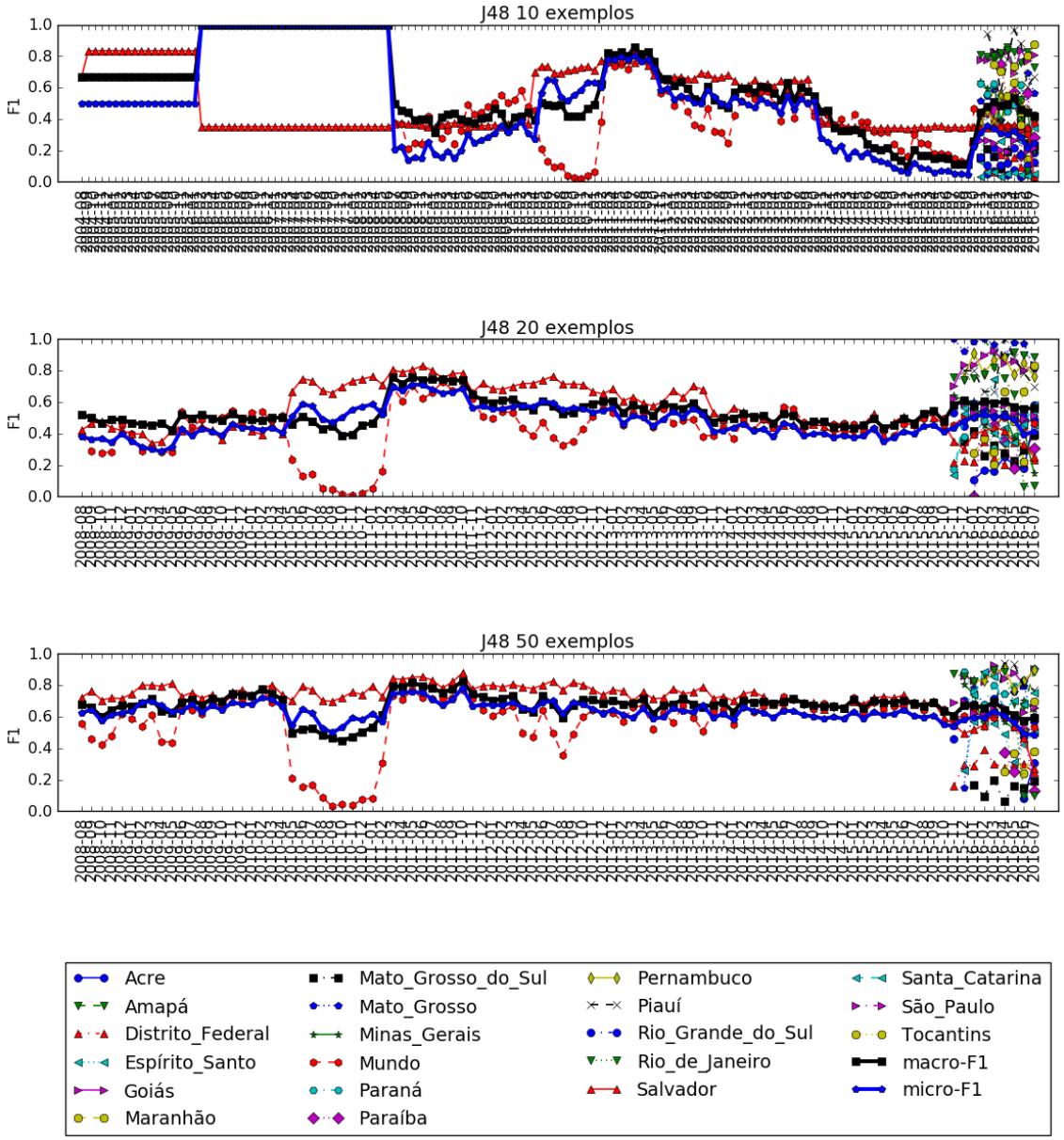

Figura 114. Resultado do algoritmo J48 em um conjunto com 10, 20 e 50 exemplos rotulados. Base: Notícias do Brasil (Estados), período: Mensal.

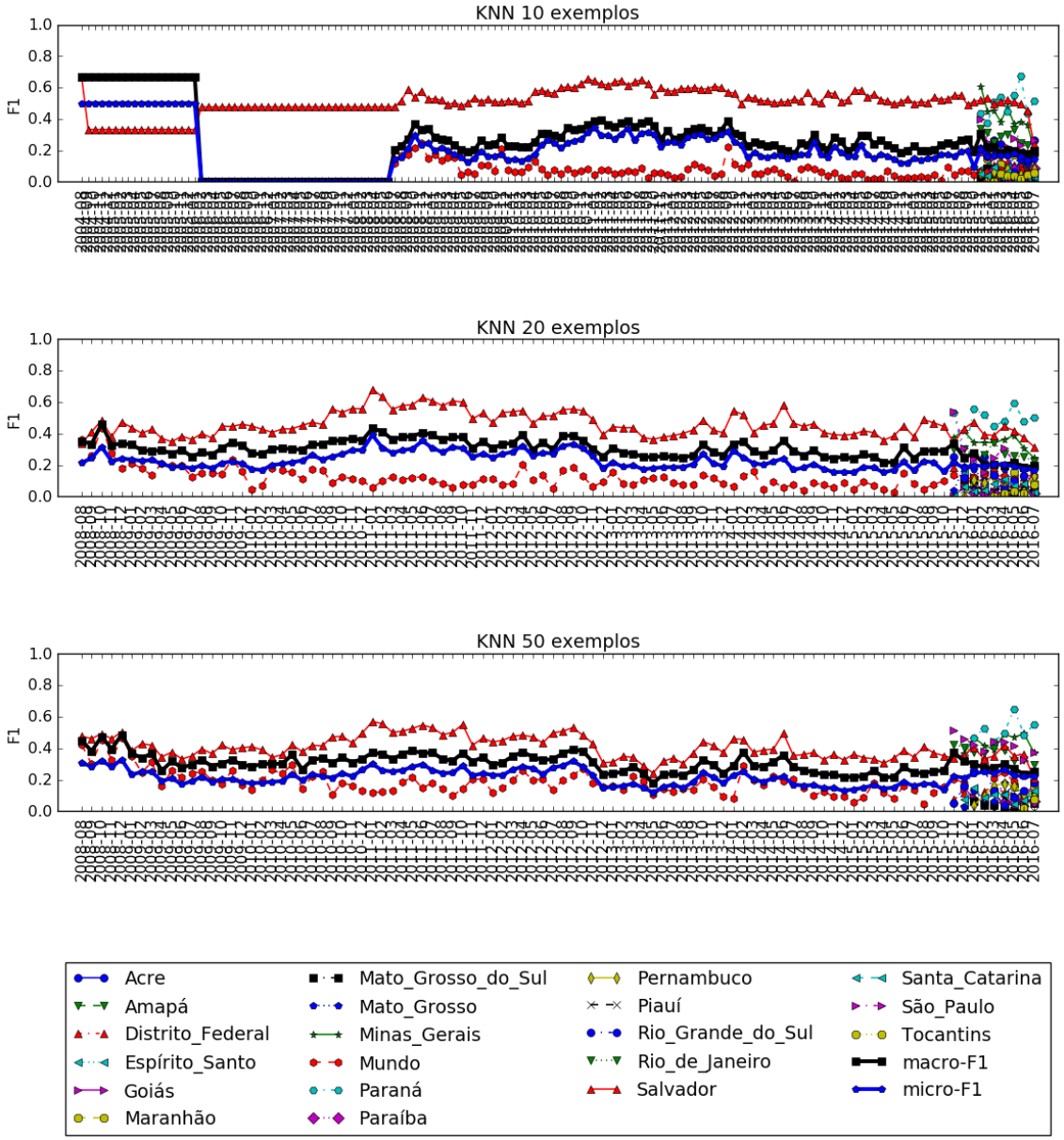

Figura 115. Resultado do algoritmo KNN em um conjunto com 10, 20 e 50 exemplos rotulados. Base: Notícias do Brasil (Estados), período: Mensal.

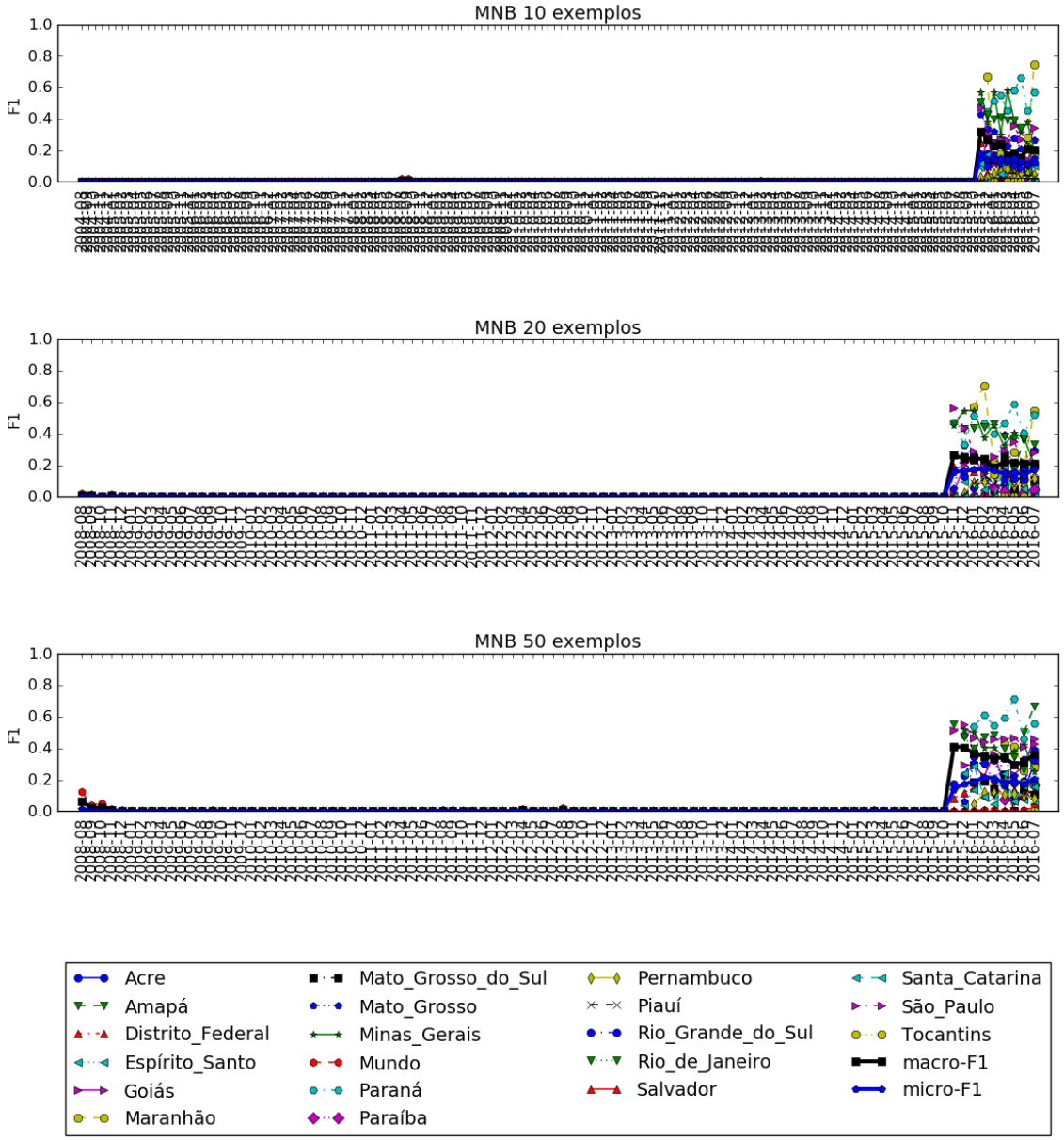

Figura 116. Resultado do algoritmo MNB em um conjunto com 10, 20 e 50 exemplos rotulados. Base: Notícias do Brasil (Estados), período: Mensal.

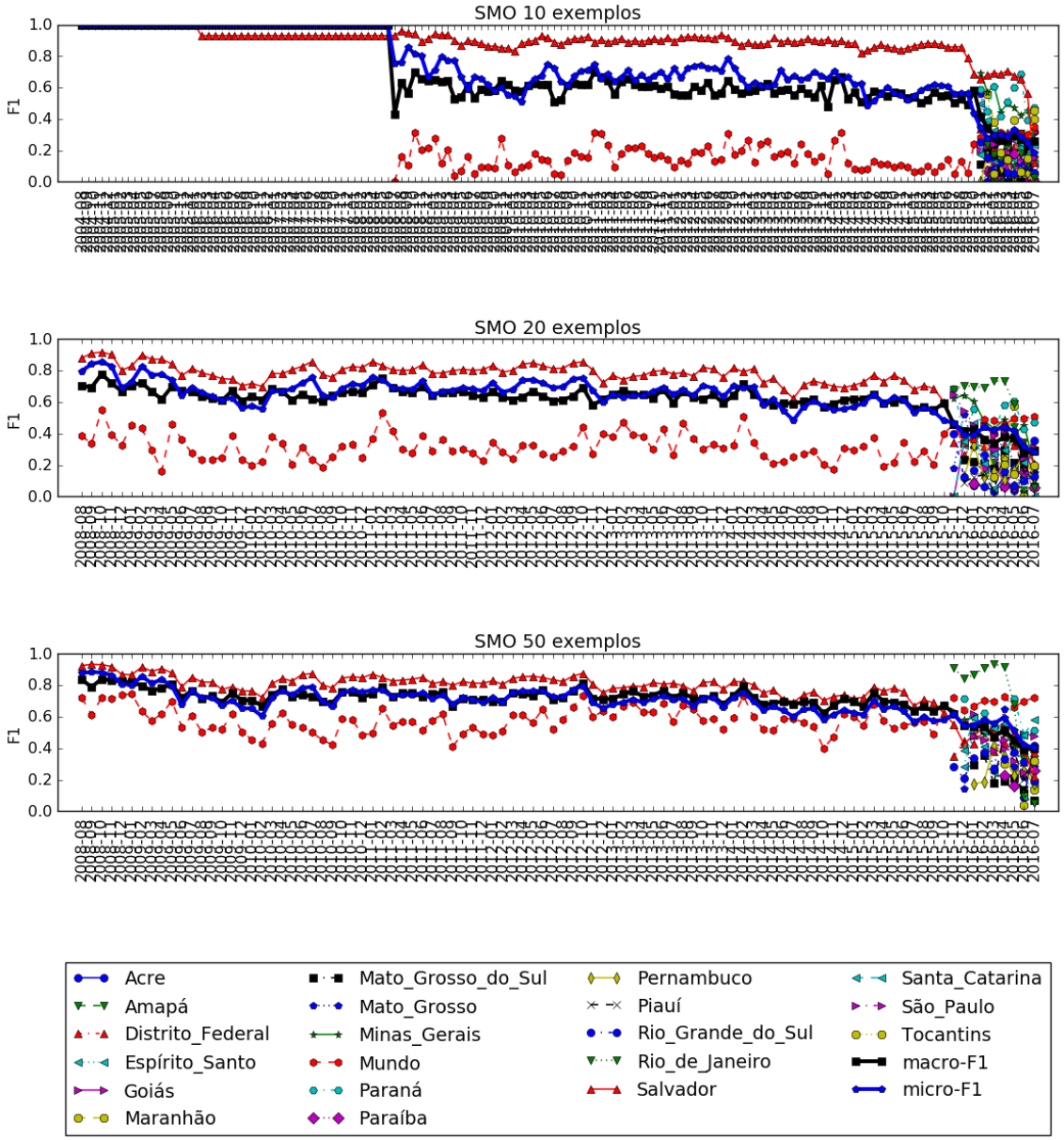

Figura 117. Resultado do algoritmo SMO em um conjunto com 10, 20 e 50 exemplos rotulados. Base: Notícias do Brasil (Estados), período: Mensal.

4.4.3.5 Considerações Gerais Sobre os Resultados

Os resultados obtidos através dos experimentos executados neste trabalho foram agrupados em tabelas, afim de possibilitar uma avaliação objetiva. Foram utilizados dois critérios para a realização da avaliação objetiva, sendo eles: o valor de degradação dos valores Macro F1 e Micro F1, que consiste na diferença entre o primeiro valor e o último valor onde as medidas Macro F1 e Micro F1 ocorrem. Já o segundo critério utilizado na avaliação objetiva foi a utilização do valor final da Macro F1 e Micro F1, sendo que o valor final da Macro F1 e Micro F1 se referem ao valor do último período onde ocorreu a medida.

Os resultados da análise objetiva foram agrupados em tabelas, sendo que cada tabela contém a base na qual foram executados os experimentos, o nome do algoritmo avaliado e a quantidade de exemplos rotulados. Para a execução da análise objetiva foi selecionado o período semestral, pois esse é o maior período de tempo onde há uma maior ocorrência de classes em mais de um intervalo de tempo. Nas Tabelas 5, 6, 7 e 8 são apresentados os valores de degradação da Macro F1 para as coleções Eventos do Milho, Notícias do Brasil (Variedades), Notícias do Brasil (Estados) e Reuters 21578 respectivamente.

Tabela 5. Degradação Macro F1. Coleção: Eventos do Milho.

Algoritmo	10 Exs	20 Exs	50 Exs
IMBHN2	0.4049	0.0941	0.0833
IMBHN	0.5989	0.8333	0.2395
J48	**0.1095**	0.1353	0.1832
KNN	0.6012	0.3474	**0.0115**
MNB	0.4297	**-0.0385**	0.0270
SMO	0.3708	0.2292	0.0385

Tabela 6. Degradação Macro F1. Coleção: Notícias do Brasil (Variedades).

Algoritmo	10 Exs	20 Exs	50 Exs
IMBHN2	0.0886	0.1627	0.2058
IMBHN	-0.0404	0.0720	0.1392
J48	0.0919	0.0955	0.1555
KNN	0.0084	0.0550	0.1551
MNB	**-0.1893**	**-0.2808**	**-0.1986**
SMO	-0.0189	0.0506	0.2071

Tabela 7. Degradação Macro F1. Coleção: Notícias do Brasil (Estados).

IMBHN2	0.3737	0.3012	0.3312
IMBHN	0.3872	0.0542	0.1742
J48	0.1882	-0.0956	0.0530
KNN	0.4771	0.1513	0.1861
MNB	**-0.2180**	**-0.2226**	**-0.2830**
SMO	0.7346	0.3703	0.3931

Tabela 8. Degradação Macro F1. Coleção: Reuters 21578.

Algoritmo	**10 Exs**	**20 Exs**	**50 Exs**
IMBHN2	-0.0360	-0.0179	-0.0632
IMBHN	0.0493	-0.0337	**-0.1111**
J48	0.0186	0.0457	-0.0688
KNN	**-0.0482**	0.0048	-0.0803
MNB	-0.0224	**-0.0407**	-0.0313
SMO	0.0782	0.0492	0.0246

Nas Tabelas 9, 10, 11 e 12 são apresentados os valores de degradação da Micro F1 para as coleções Eventos do Milho, Notícias do Brasil (Variedades), Notícias do Brasil (Estados) e Reuters 21578 respectivamente.

Tabela 9. Degradação Micro F1. Coleção: Eventos do Milho.

Algoritmo	**10 Exs**	**20 Exs**	**50 Exs**
IMBHN2	0.2379	0.0417	-0.0667
IMBHN	0.4315	0.8750	0.0476
J48	0.2097	0.0833	0.0000
KNN	0.3669	0.3333	**-0.1333**
MNB	0.3629	**-0.1667**	**-0.1333**
SMO	**0.2056**	0.1250	**-0.1333**

Tabela 10. Degradação Micro F1. Coleção: Notícias do Brasil (Variedades).

Algoritmo	**10 Exs**	**20 Exs**	**50 Exs**
IMBHN2	-0.0973	-0.0941	-0.0597
IMBHN	-0.1486	-0.1390	-0.1252
J48	-0.0947	-0.0510	0.0244
KNN	-0.1788	-0.1727	-0.0910
MNB	**-0.2651**	**-0.3124**	**-0.2409**
SMO	-0.1415	-0.1627	-0.0474

Tabela 11. Degradação Micro F1. Coleção: Notícias do Brasil (Estados).

Algoritmo	10 Exs	20 Exs	50 Exs
IMBHN2	0.1523	0.0621	0.0964
IMBHN	0.2220	-0.0765	0.0000
J48	0.2182	-0.1044	0.0539
KNN	0.3401	0.0555	0.0557
MNB	**-0.1370**	**-0.1575**	**-0.1911**
SMO	0.7192	0.4009	0.3386

Tabela 12. Degradação Micro F1. Coleção: Reuters 21578.

Algoritmo	10 Exs	20 Exs	50 Exs
IMBHN2	-0.0993	-0.0776	-0.0530
IMBHN	-0.1353	-0.1016	-0.0800
J48	-0.1688	-0.0951	-0.1127
KNN	-0.1154	-0.1094	-0.0832
MNB	**-0.1801**	**-0.1183**	-0.1012
SMO	-0.1763	-0.1147	**-0.1390**

Nas Tabelas 13, 14, 15 e 16 são apresentados os valores finais da Macro F1 para as coleções Eventos do Milho, Notícias do Brasil (Variedades), Notícias do Brasil (Estados) e Reuters 21578 respectivamente.

Tabela 13. Valor final da Macro F1. Coleção: Eventos do Milho.

Algoritmo	10 Exs	20 Exs	50 Exs
IMBHN2	0,3625	0,7059	**0,6667**
IMBHN	0,1685	0,1667	0,2671
J48	0,2541	**0,7738**	0,3333
KNN	0,1663	0,5617	**0,6667**
MNB	0,3703	0,5385	**0,6667**
SMO	**0,3966**	0,4375	**0,6667**

Tabela 14. Valor final da Macro F1. Coleção: Notícias do Brasil(Variedades).

Algoritmo	10 Exs	20 Exs	50 Exs
IMBHN2	**0,4526**	**0,5174**	**0,5450**
IMBHN	0,4008	0,3865	0,4845
J48	0,2408	0,2630	0,3029
KNN	0,4047	0,4419	0,4663
MNB	0,4411	0,4517	0,5191
SMO	0,3335	0,4071	0,4802

Tabela 15. Valor final da Macro F1. Coleção: Notícias do Brasil (Estados).

Algoritmo	10 Exs	20 Exs	50 Exs
IMBHN2	0,2930	0,3626	0,4824
IMBHN	0,2795	0,3430	0,4254
J48	**0,4785**	**0,5845**	**0,5997**
KNN	0,1896	0,1988	0,2487
MNB	0,2180	0,2261	0,3003
SMO	0,2654	0,3415	0,4365

Tabela 16. Valor final da Macro F1. Coleção: Reuters 21578.

Algoritmo	10 Exs	20 Exs	50 Exs
IMBHN2	**0,6413**	**0,6674**	0,7144
IMBHN	0,4978	0,6350	**0,7564**
J48	0,5375	0,5566	0,6711
KNN	0,5714	0,5584	0,6629
MNB	0,5738	0,6284	0,7192
SMO	0,3383	0,3895	0,5045

Nas Tabelas 17, 18, 19 e 20 são apresentados os valores finais da Micro F1 para as coleções Eventos do Milho, Notícias do Brasil (Variedades), Notícias do Brasil (Estados) e Reuters 21578 respectivamente.

Tabela 17. Valor final da Micro F1. Coleção: Eventos do Milho.

Algoritmo	10 Exs	20 Exs	50 Exs
IMBHN2	0,3871	0,6250	**0,6667**
IMBHN	0,1935	0,1250	0,2857
J48	0,2903	**0,7500**	0,3333
KNN	0,2581	0,5000	**0,6667**
MNB	0,3871	0,5000	**0,6667**
SMO	0,4194	0,3750	**0,6667**

Tabela 18. Valor final da Micro F1. Coleção: Notícias do Brasil (Variedades).

Algoritmo	10 Exs	20 Exs	50 Exs
IMBHN2	**0,5381**	**0,6721**	**0,7339**
IMBHN	0,4113	0,4886	0,6597
J48	0,3695	0,3178	0,3678
KNN	0,4821	0,5379	0,5945
MNB	0,4397	0,4198	0,4474
SMO	0,3470	0,4830	0,6408

Tabela 19. Valor final da Micro F1. Coleção: Notícias do Brasil (Estados).

Algoritmo	10 Exs	20 Exs	50 Exs
IMBHN2	**0,3477**	0,4693	**0,6172**
IMBHN	0,2780	0,3084	0,4199
J48	0,2818	**0,4720**	0,5646
KNN	0,1599	0,1901	0,2378
MNB	0,1370	0,1584	0,1953
SMO	0,2808	0,3874	0,5139

Tabela 20. Valor final da Micro F1. Coleção: Reuters 21578.

Algoritmo	10 Exs	20 Exs	50 Exs
IMBHN2	0,7712	0,7881	0,7950
IMBHN	0,7458	0,7938	0,8525
J48	0,7444	0,7062	0,7847
KNN	0,7514	0,7881	0,7979
MNB	**0,7867**	**0,8051**	**0,8614**
SMO	0,6582	0,7203	0,7212

Com base na análise objetiva, o algoritmo que obteve a menor degradação de performance nas medidas Macro F1 e Micro F1 foi o algoritmo MNB. No valor final da Macro F1, os algoritmos com melhor desempenho em ordem decrescente foram o IMBHN2 e o J48, e na medida Micro F1 os algoritmos que tiveram a maior performance, em ordem decrescente, foram o IMBHN2 e o MNB. Assim, com base na análise da avaliação experimental executada neste trabalho, foi possível concluir que:

(i) Em geral, a performance dos classificadores diminui ao longo do tempo independentemente do período avaliado.
(ii) A base de textos utilizada influi diretamente na performance dos classificadores.
(iii) Um número maior de exemplos rotulados no conjunto de teste garante uma menor degradação de performance e um maior valor final de performance de classificação.

5. Considerações Finais e Trabalho Futuros

Nos dias atuais há uma grande quantidade de dados textuais sendo produzida, e grande parte desses dados textuais está na forma de notícias. Processar, organizar, gerenciar e extrair conhecimento dessa grande quantidade de notícias manualmente exige um grande esforço humano, sendo muitas vezes impossível de ser realizado manualmente. Dessa forma, fica evidente a necessidade de definir métodos computacionais para a classificação automática de textos, sendo que na literatura há poucas pesquisa retratando como a performance de classificadores automáticas de texto se comporta ao longo do tempo, considerando diferentes algoritmos, periodicidades e número de exemplos rotulados.

O resultados obtidos neste trabalho demostram que a base de textos utilizada influi diretamente na performance dos classificadores. Esse comportamento pode ser observado

na base da Reuters 21578, onde houve uma menor degradação de performance. Os resultados obtidos também demonstraram que, em geral, a performance dos classificadores diminui ao longo do tempo e um conjunto com um maior número de exemplos rotulados mantém a retenção de performance mais linear. Assim, têm-se como objetivo em trabalhos futuros:

(i) Analisar os algoritmos de classificação em novas bases contendo textos de domínios diferentes.
(ii) Analisar bases com uma maior diversidade de exemplos rotulados. Afim de encontrar um número suficiente de exemplos rotulados para manter uma maior performance de classificação e uma menor degradação ao longo do tempo.
(iii) Analisar as técnicas apresentadas na Seção II e compará-las com os resultados obtidos neste trabalho.

Referências

Aggarwal, C. C. and Zhai, C. (2012). *Mining text data*. Springer Science & Business Media.

Aggarwal, C. C., Zhao, Y., and Philip, S. Y. (2014). On the use of side information for mining text data. *IEEE transactions on knowledge and data engineering*, 26(6):1415–1429.

Aha, D. W. (1991). Incremental constructive induction: An instance-based approach. In *Proceedings of the Eighth International Workshop on Machine Learning*, pages 117–121.

Angelova, R. and Weikum, G. (2006). Graph-based text classification: learn from your neighbors. In *Proceedings of the 29th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 485–492. ACM.

Baena-Garcıa, M., del Campo-Ávila, J., Fidalgo, R., Bifet, A., Gavalda, R., and Morales-Bueno, R. (2006). Early drift detection method. In *Fourth international workshop on knowledge discovery from data streams*, volume 6, pages 77–86.

Belkin, M., Niyogi, P., and Sindhwani, V. (2006). Manifold regularization: A geometric framework for learning from labeled and unlabeled examples. *Journal of machine learning research*, 7(Nov):2399–2434.

Bifet, A. and Gavalda, R. (2007). Learning from time-changing data with adaptive windowing. In *Proceedings of the 2007 SIAM International Conference on Data Mining*, pages 443–448. SIAM.

Bifet, A., Holmes, G., Pfahringer, B., Kirkby, R., and Gavaldà, R. (2009). New ensemble methods for evolving data streams. In *Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 139–148. ACM.

Bifet, A., Holmes, G., Pfahringer, B., Read, J., Kranen, P., Kremer, H., Jansen, T., and Seidl, T. (2011). Moa: a real-time analytics open source framework. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, pages 617–620. Springer.

Billsus, D. and Pazzani, M. J. (1999). A hybrid user model for news story classification. In *UM99 User Modeling*, pages 99–108. Springer.

Castillo, C. (2005). Effective web crawling. In *Acm sigir forum*, volume 39, pages 55–56. Acm.

Chakrabarti, S., Dom, B., and Indyk, P. (1998). Enhanced hypertext categorization using hyperlinks. In *ACM SIGMOD Record*, volume 27, pages 307–318. ACM.

Chu, F. and Zaniolo, C. (2004). Fast and light boosting for adaptive mining of data streams. In *Pacific-Asia Conference on Knowledge Discovery and Data Mining*, pages 282–292. Springer.

Domingos, P. and Hulten, G. (2003). A general framework for mining massive data streams. *Journal of Computational and Graphical Statistics*, 12(4):945–949.

Feldman, R. and Sanger, J. (2007). *The text mining handbook: advanced approaches in analyzing unstructured data*. Cambridge university press.

Fern, A. and Givan, R. (2003). Online ensemble learning: An empirical study. *Machine Learning*, 53(1-2):71–109.

Gaber, M. M. and Yu, P. S. (2006). Detection and classification of changes in evolving data streams. *International Journal of Information Technology & Decision Making*, 5(04):659–670.

Hall, M., Frank, E., Holmes, G., Pfahringer, B., Reutemann, P., and Witten, I. H. (2009). The weka data mining software: an update. *ACM SIGKDD explorations newsletter*, 11(1):10–18.

Haussler, D., Littlestone, N., and Warmuth, M. K. (1994). Predicting $\{0, 1\}$-functions on randomly drawn points. *Information and Computation*, 115(2):248–292.

Hulten, G., Spencer, L., and Domingos, P. (2001). Mining time-changing data streams. In *Proceedings of the seventh ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 97–106. ACM.

Jackowski, K. (2014). Fixed-size ensemble classifier system evolutionarily adapted to a recurring context with an unlimited pool of classifiers. *Pattern Analysis and Applications*, 17(4):709–724.

Jacobs, R. A., Jordan, M. I., Nowlan, S. J., and Hinton, G. E. (1991). Adaptive mixtures of local experts. *Neural computation*, 3(1):79–87.

Klinkenberg, R. and Joachims, T. (2000). Detecting concept drift with support vector machines. In *ICML*, pages 487–494.

Kolter, J. Z. and Maloof, M. A. (2003). Dynamic weighted majority: A new ensemble method for tracking concept drift. In *Data Mining, 2003. ICDM 2003. Third IEEE International Conference on*, pages 123–130. IEEE.

Kolter, J. Z. and Maloof, M. A. (2007). Dynamic weighted majority: An ensemble method for drifting concepts. *Journal of Machine Learning Research*, 8(Dec):2755–2790.

Krawczyk, B. and Woźniak, M. (2015). One-class classifiers with incremental learning and forgetting for data streams with concept drift. *Soft Computing*, 19(12):3387–3400.

Kuechler, W. L. (2007). Business applications of unstructured text. *Communications of the ACM*, 50(10):86–93.

Kuncheva, L. I. (2008). Classifier ensembles for detecting concept change in streaming data: Overview and perspectives. In *2nd Workshop SUEMA*, volume 2008, pages 5–10.

Lazarescu, M. M., Venkatesh, S., and Bui, H. H. (2004). Using multiple windows to track concept drift. *Intelligent data analysis*, 8(1):29–59.

Lewis, D. D. (1997). Reuters-21578 text categorization test collection, distribution 1.0. *http://www. research. att. com/˜ lewis/reuters21578. html*.

Lu, Q. and Getoor, L. (2003). Link-based classification. In *ICML*, volume 3, pages 496–503.

Manning, C. D., Raghavan, P., Schütze, H., et al. (2008). *Introduction to information retrieval*, volume 1. Cambridge university press Cambridge.

Marcacini, R. M., Sinoara, R. A., Matsuno, I. P., and Rezende, S. O. (2013). Aprendizado não supervisionado de websensors. *Mining and Learning*.

Markou, M. and Singh, S. (2003). Novelty detection: a review—part 1: statistical approaches. *Signal processing*, 83(12):2481–2497.

Newman, M. (2010). Networks: an introduction. 2010. *United Slates: Oxford University Press Inc., New York*, pages 1–2.

Oh, H.-J., Myaeng, S. H., and Lee, M.-H. (2000). A practical hypertext catergorization method using links and incrementally available class information. In *Proceedings of the 23rd annual international ACM SIGIR conference on Research and development in information retrieval*, pages 264–271. ACM.

Quinlan, J. (1993). C4. 5: Programs for machine learning. c4. 5-programs for machine learning/j. ross quinlan.

Richardson, L. (2015). Beautiful Soup Documentation.

Rodríguez, J. J. and Kuncheva, L. I. (2008). Combining online classification approaches for changing environments. In *Joint IAPR International Workshops on Statistical Techniques in Pattern Recognition (SPR) and Structural and Syntactic Pattern Recognition (SSPR)*, pages 520–529. Springer.

Rossi, R. G. (2016). *Classificação automática de textos por meio de aprendizado de máquina baseado em redes*. PhD thesis, Universidade de São Paulo.

Rossi, R. G., de Andrade Lopes, A., de Paulo Faleiros, T., and Rezende, S. O. (2014). Inductive model generation for text classification using a bipartite heterogeneous network. *Journal of Computer Science and Technology*, 29(3):361–375.

Rossi, R. G. and Rezende, S. O. (2011). Building a topic hierarchy using the bag-of-related-words representation. In *Proceedings of the 11th ACM symposium on Document engineering*, pages 195–204. ACM.

Russell, S. J. and Norvig, P. (2002). Artificial intelligence: a modern approach (international edition).

Samuel, A. L. (1959). Some studies in machine learning using the game of checkers. *IBM Journal of research and development*, 3(3):210–229.

Schlimmer, J. C. and Granger, R. H. (1986). Incremental learning from noisy data. *Machine learning*, 1(3):317–354.

Shalev-Shwartz, S. and Ben-David, S. (2014). *Understanding machine learning: From theory to algorithms*. Cambridge university press.

Shipp, C. A. and Kuncheva, L. I. (2002). Relationships between combination methods and measures of diversity in combining classifiers. *Information fusion*, 3(2):135–148.

Sobolewski, P. and Wozniak, M. (2013). Concept drift detection and model selection with simulated recurrence and ensembles of statistical detectors. *J. UCS*, 19(4):462–483.

Stanley, K. O. (2003). Learning concept drift with a committee of decision trees. *Informe técnico: UT-AI-TR-03-302, Department of Computer Sciences, University of Texas at Austin, USA*.

Street, W. N. and Kim, Y. (2001). A streaming ensemble algorithm (sea) for large-scale classification. In *Proceedings of the seventh ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 377–382. ACM.

Subramanya, A. and Bilmes, J. (2008). Soft-supervised learning for text classification. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pages 1090–1099. Association for Computational Linguistics.

Tan, P.-N. et al. (2006). *Introduction to data mining*. Pearson Education India.

Tan, P.-N., Steinbach, M., and Kumar, V. (2005). Association analysis: basic concepts and algorithms. *Introduction to data mining*, pages 327–414.

Tan, S. (2006). An effective refinement strategy for knn text classifier. *Expert Systems with Applications*, 30(2):290–298.

Tsymbal, A., Pechenizkiy, M., Cunningham, P., and Puuronen, S. (2008). Dynamic integration of classifiers for handling concept drift. *Information fusion*, 9(1):56–68.

Turner, V., Gantz, J. F., Reinsel, D., and Minton, S. (2014). The digital universe of opportunities: Rich data and the increasing value of the internet of things. *IDC Analyze the Future*.

Ur-Rahman, N. and Harding, J. A. (2012). Textual data mining for industrial knowledge management and text classification: A business oriented approach. *Expert Systems with Applications*, 39(5):4729–4739.

Wang, H., Fan, W., Yu, P. S., and Han, J. (2003). Mining concept-drifting data streams using ensemble classifiers. In *Proceedings of the ninth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 226–235. AcM.

Weiss, S. M., Indurkhya, N., and Zhang, T. (2010). *Fundamentals of predictive text mining*, volume 41. Springer.

Widmer, G. and Kubat, M. (1996). Learning in the presence of concept drift and hidden contexts. *Machine learning*, 23(1):69–101.

Zhou, D., Bousquet, O., Lal, T. N., Weston, J., and Schölkopf, B. (2003). Learning with local and global consistency. In *NIPS*, volume 16, pages 321–328.

Zhu, X., Ghahramani, Z., Lafferty, J., et al. (2003). Semi-supervised learning using gaussian fields and harmonic functions. In *ICML*, volume 3, pages 912–919.

Žliobaitė, I. (2010). Learning under concept drift: an overview. *arXiv preprint arXiv:1010.4784*.