Bruno Henrique Joia

Gustavo dos Santos Rocha

Igor Yasuo Gushiken

Bom Cupom – Sistema Inteligente para Geração de Promoções Personalizadas em Supermercados

Monografia apresentada à Escola Politécnica da Universidade de São Paulo para a conclusão do curso de graduação em Engenharia de Computação.

São Paulo
2012

Bruno Henrique Joia

Gustavo dos Santos Rocha

Igor Yasuo Gushiken

Bom Cupom – Sistema Inteligente para Geração de Promoções Personalizadas em Supermercados

Monografia apresentada à Escola Politécnica da Universidade de São Paulo para a conclusão do curso de graduação em Engenharia de Computação.

Área de concentração:
Engenharia de Computação

Orientador:
Prof. Dr. Antonio Mauro Saraiva

Co-orientador:
Eng. MSc. Etienne Américo Cartolano Júnior

São Paulo
2012

Agradecimentos

O trabalho aqui apresentado é fruto da colaboração de mais do que três graduandos de Engenharia da Computação da Escola Politécnica da USP. Somos gratos às inúmeras pessoas que de alguma forma contribuiram para sua realização, seja acompanhando-nos no nosso dia-a-dia de trabalho, seja na manutenção de uma estrutura acadêmica propícia para que este pudesse se concretizar.

Gostaríamos de agradecer especialmente ao nosso orientador, Prof. Dr. Antonio Saraiva, e ao nosso co-orientador, Eng. MSc. Etienne Cartolano Junior, por terem nos acompanhado e orientado durante todo o projeto de formatura, ajudando-nos a manter um bom equilíbrio entre trabalho focado e visão abrangente ao longo de todo o projeto, e a encontrar a boa direção quando este parecia oferecer várias possibilidades obscuras.

Agradecemos também ao Prof. Dr. Paulo Cugnasca e ao Prof. Dr. João Batista, por estarem sempre à disposição dos alunos no que diz respeito ao projeto de formatura, orientando-nos pontualmente a cada apresentação e relatório entregue.

Somos igualmente gratos à Profa. Dra. Solange Rezende, do ICMC - USP, por sua orientação específica, porém fundamental, no que tange os algoritmos de regras de associação, e ao Prof. Dr. Jaime Sichman, do PCS - USP, quem voluntariosamente nos apresentou à sua colega.

Por fim, agradecemos aos colegas de sala que acompanharam a evolução de nosso trabalho, criticando e fazendo sugestões ao longo do ano, e aos colegas egressos que se interessaram por nosso projeto, comentando nas mídias sociais e fazendo sugestões para torná-lo mais completo. Todos foram fundamentais para revermos nossos métodos e tornarmos este trabalho fruto de uma inteligência coletiva.

Resumo

Existe atualmente no mundo empresarial um consenso de que a satisfação do cliente é um dos fatores chave para o sucesso de um negócio. Uma das formas de satisfazer os clientes é conhecer suas preferências para lhes oferecer um serviço ou produto mais próximo do que eles esperam. Durante a operação diária de uma empresa, dados valiosos sobre a interação dos clientes com a empresa são gerados continuamente. A análise destes dados pode revelar o comportamento e as preferências dos clientes, gerando conhecimento e valor para o negócio.

Com isso em mente, o trabalho apresentado nesta monografia consiste no desenvolvimento de um sistema que analisa os hábitos de consumo dos clientes de um supermercado para gerar cupons de desconto promocionais personalizados, utilizando técnicas de *data mining*. Neste projeto temos acesso a uma pequena parte do banco de dados de um supermercado de médio porte e com esses dados testamos uma série de modelos sugeridos para a geração dos cupons de desconto. Uma classificação dos produtos é feita para desconsiderar características muito específicas e assim aumentar a relevância dos resultados. Essa classificação é feita com o auxílio de APIs de sites de busca de preços e técnicas de comparação de strings. Em seguida construímos modelos dos padrões de consumo globais e individuais dos clientes a partir de técnicas de *data mining* e técnicas puramente estatísticas, respectivamente. Finalmente a geração dos cupons é feita para cada cliente individualmente com uma integração dos modelos gerados anteriormente.

Apresentamos neste documento a descrição detalhada de cada uma das etapas e técnicas utilizadas, bem como uma análise dos resultados obtidos.

Abstract

There is a consensus in the entrepreunerial world that customer satisfaction is one of the key factors for the success of a business nowadays. One way to satisfy customers is to know their preferences, in order to offer a service or product aligned with their expectations. During the daily operation of a business, valuable data on customer interaction with the company are generated continuously. Analysis of these data may reveal the behavior and preferences of customers, generating knowledge and adding value to the business.

With that in mind, the work presented in this thesis consists in the development of a system that analyzes the spending habits of customers in a supermarket to generate custom promotional discount coupons, using data mining techniques. In this project we have access to a little portion of a midrange supermarket database, with which we tested a series of models suggested for the generation of discount coupons. A classification of products is made to disregard very specific characteristics and thus improve the relevance of results. This classification is done with the aid of online product search engine's APIs and string comparison techniques. Then we build models of the global and individual consumption patterns using data mining and purely statistical techniques, respectively. Finally the generation of coupons is made for each customer individually by using the previously generated models together.

We present herein a detailed description of each step and techniques used during this work, as well as an analysis of the results obtained.

Sumário

Lista de Figuras

Lista de Tabelas

1 Introdução

1.1 Objetivo

O objetivo deste trabalho é desenvolver um sistema que utiliza técnicas de *data mining* para identificar e modelar padrões de consumo em supermercados a fim de gerar cupons de desconto promocionais personalizados, em produtos potencialmente interessantes ao cliente.

Esse sistema, chamado Bom Cupom, deve ser capaz de emitir um cupom de desconto (figura 1.1) gerado de forma inteligente - pois utiliza técnicas de *data mining* para extrair conhecimento da base de dados de compras de seus clientes - e criteriosa - pois a escolha dos produtos baseia-se em critérios bem definidos pelas políticas de desconto dos supermercados.

1.2 Motivação

É cada vez maior o consenso de que a satisfação do cliente é um dos fatores chave para o sucesso de um negócio atualmente. Clientes satisfeitos tendem a retornar ao estabalecimento, tornam-se fiéis e em muitos casos defensores da marca. Uma das formas de satisfazer os clientes é conhecer o seu comportamento, hábitos e preferências, para assim lhes oferecer um serviço ou produto mais próximo do que ele espera.

Durante a operação diária de uma empresa, novas informações e dados sobre a interação dos clientes são gerados continuamente. Estes dados valiosos, que podem justamente revelar o comportamento e preferências dos clientes, são acumulados em bases de dados de grande porte, mas muitas vezes não são utilizados para a geração de conhecimento. Além disso, sem um tratamento adequado este conjunto de dados não poderá ter o seu potencial de geração de conhecimento aproveitado ao máximo. A possibilidade de utilizar técnicas bem estabelecidas de análise de grandes quantidades de dados brutos (como as técnicas de *data mining*)

Figura 1.1: Exemplo de cupom gerado pelo Bom Cupom

permite a uma empresa extrair conhecimento sobre os hábitos e preferências de seus clientes, a fim de adaptar seu negócio para satisfazer às expectativas dos mesmos (RYGIELSKI; WANG; YEN, 2002). Hoje este tema ganhou notoriedade no meio acadêmico e corporativo, sob a denominação de Big Data (TAYLOR, 2011).

Neste trabalho, parte-se da premissa de que o conhecimento do comportamento de cada cliente permite a uma empresa tomar ações a fim de fornecer uma melhor experiência de compra ao consumidor, não só para adaptar o seu negócio ao cliente, mas também para atrair e incentivar mudanças de hábitos destes, sempre de forma vantajosa para ambos. A possibilidade de agir neste sentido com a ajuda das ferramentas de *data mining* é a principal motivação deste trabalho.

As técnicas de *data mining* podem ser aplicadas em um vasto conjunto de áreas da engenharia. Portanto, o estudo aprofundado dessas técnicas gera um conhecimento muito importante não somente para a realização deste projeto de formatura, mas também para projetos futuros, o que consiste em outra grande

motivação para a escolha deste tema.

1.3 Justificativa

Os donos de estabelecimentos varejistas deparam-se frequentemente com questões como as seguintes:

- Como fidelizar os clientes do meu estabelecimento?
- Como vender mais produtos que me retornam maior margem de lucro?
- Como atrair novos clientes?
- Como aumentar o volume de compras de clientes que compram pouco?

Por outro lado, os clientes são desejosos de um tratamento individualizado, que lhes permita encontrar os produtos que eles mais precisam com os melhores preços no mesmo lugar, sem necessidade de procurar em vários estabelecimentos. O resultado deste trabalho visa fornecer aos varejistas e consumidores uma alternativa interessante a este problema complexo, atendendo simultaneamente os desejos de ambas as partes. De forma geral, o sistema Bom Cupom justifica-se pela possibilidade de:

- Fidelizar o cliente por meio de descontos em produtos que ele mais consome;
- Permitir aos clientes experimentarem produtos de qualidade superior por um preço razoável;
- Introduzir produtos de maior margem de lucro potencialmente interessantes ao cliente;
- Concentrar as compras do cliente em um único lugar, consequentemente aumentando o volume de compras do cliente para o supermercado;
- Atrair novos clientes interessados pelos descontos personalizados.

A ideia de oferecer descontos personalizados aos clientes de fato não é nova, e algumas delas inclusive já foram implantadas. Uma rede de drogarias em São Paulo adotou um sistema onde o cliente com o cartão da drogaria pode solicitar

e receber descontos personalizados no balcão, ao apresentar o seu cartão de fidelidade [1]. A existência prévia de um sistema como este mostra que existe um real interesse de mercado.

No entanto, uma distinção é necessária, visto que os hábitos de consumo em um supermercado são ligeiramente diferentes dos de uma drogaria. Os supermercados possuem uma oferta de produtos muito maior e mais diversificada que as drogarias. Por isso, e aliado a hábitos impulsivos, um cliente no supermercado está muito mais suscetível a conhecer novos produtos. Este comportamento abre um leque maior de possibilidades para sugerir aos clientes produtos potencialmente interessantes, porém, inexplorados. Além disso, supermercados vendem em volume superior, o que lhes confere uma grande flexibilidade para negociar descontos com seus fornecedores ou admitir perdas temporárias. Por isso os requisitos do sistema, e consequentemente as técnicas e critérios adotados não são necessariamente os mesmos nos dois casos, sendo necessário um estudo específico para o caso dos supermercados.

1.4 Organização do Trabalho

A primeira parte deste trabalho introduz conceitos gerais relativos às técnicas de *data mining* relevantes neste projeto, bem como uma breve descrição do estado da arte neste domínio.

Na segunda parte do trabalho apresentamos o modelo de funcionamento de um supermercado comum sem o sistema de cupons personalizados e a especificação do sistema proposto, detalhando seu escopo, requisitos funcionais e não funcionais, modelo do processo, alguns casos de uso e políticas de desconto possíveis.

Na quarta parte apresentamos a estrutura do banco de dados utilizado neste projeto, a divisão do Bom Cupom em módulos funcionais e os preparativos para o desenvolvimento dos mesmos.

Na quinta parte, que envolve os capítulos 5 a 8, apresentamos os conceitos mais intimamente ligados a cada um dos módulos do Bom Cupom, bem como a metodologia adotada no seu desenvolvimento e os resultados obtidos.

Finalmente, apresentamos nossas conclusões e considerações sobre possíveis trabalhos futuros.

[1] `http://www.revistafator.com.br/ver_noticia.php?not=177625`, acessado em 20 de novembro de 2012

2 Aspectos Conceituais

2.1 Data Mining

O conceito de *data mining* é relativamente amplo e permite várias interpretações. Todas as definições do conceito, porém, concordam sobre o fato de que *data mining* trata da extração de conhecimento a partir de dados.

Segundo (WITTEN; FRANK, 2005), *data mining* é a tecnologia que permite a extração de informação implícita, previamente desconhecida e potencialmente útil a partir de conjuntos de dados. A idéia é construir programas de computador que investiguem bancos de dados automaticamente, em busca de regularidades ou padrões nestes dados. Muitas pessoas tratam a mineração de dados como sinônimo de outro termo popularmente utilizado, Descoberta de Conhecimento a partir de Dados ou KDD (*Knowledge Discovery from Data*) (HAN; KAMBER, 2006).

De maneira geral, a tecnologia de *data mining* é formada por um conjunto de ferramentas e técnicas que são capazes de explorar um conjunto de dados, extraindo ou ajudando a evidenciar padrões e auxiliando na descoberta de conhecimento, através do uso de algoritmos de aprendizagem ou classificação baseados em redes neurais e estatística. Esse conhecimento pode ser apresentado por essas ferramentas de diversas formas: agrupamentos, hipóteses, regras, árvores de decisão, grafos ou dendrogramas.

Nas análises de *data mining*, o conjunto de dados a ser garimpado normalmente consiste de uma tabela desnormalizada, preparada para este fim a partir do banco de dados original. Esta tabela normalmente é um conjunto de exemplos, com seus atributos e eventualmente a classe de cada exemplo. Na teoria de *data mining* estes são alguns termos comumente usados:

Exemplo: instância a ser classificada numa tarefa de aprendizagem. Neste projeto, dependendo do contexto, pode ser referente a produtos, clientes ou uma sacola de compras;

Classe (categoria): classificação atribuída a um conjunto de exemplos. No caso de clientes, possíveis classes são alto e baixo poder de compra, e para os produtos, possíveis classes são definidas pelos avatares (capítulo 5);

Atributo: características que definem um exemplo. Cada exemplo tem um ou mais atributos e um valor relacionado a cada atributo. No caso dos produtos, possíveis atributos são nome e preço.

As técnicas de *data mining* mais relevantes neste projeto podem ser classificadas em três grandes grupos: aprendizado supervisionado, aprendizado não supervisionado e regras de associação (um tipo particular de aprendizado não supervisionado).

2.1.1 Aprendizado Supervisionado

É a tarefa de classificar os exemplos segundo um conjunto de classes previamente definido. É chamado de "supervisionado" pois utiliza exemplos previamente e corretamente classificados, de modo a permitir uma orientação do aprendizado. Uma regra de classificação é inferida a partir da análise das relações entre atributos e classes dos exemplos já classificados.

Uma possível aplicação dessa técnica dentro deste projeto é a classificação de cada produto em um avatar. A partir de alguns métodos é possível classificar alguns produtos, porém não todos. Os outros podem ser classificados com um classificador gerado com aprendizado supervisionado a partir dos exemplos já classificados.

2.1.2 Aprendizado Não Supervisionado

No caso da ausência de exemplos corretamente classificados, uma alternativa é a análise dos dados em busca de padrões e regularidades relacionados aos atributos. Tais regularidades podem caracterizar exemplos que seriam classificados dentro de uma mesma categoria.

Clustering é uma técnica de aprendizado não supervisionado, pois realiza a classificação sem qualquer informação previamente fornecida, somente a partir da análise de características e semelhanças. A partir das semelhanças entre os atributos de um conjunto de exemplos, os algoritmos de *clustering* podem criar grupos de exemplos com características em comum.

Para o projeto em questão, esta técnica pode ser útil para a classificação de avatares na ausência de uma classificação prévia, assim como para a classificação de clientes segundo seus hábitos de consumo e poder de compra.

2.1.3 Regras de Associação

Regras de associação são uma técnica de *data mining* que busca identificar padrões em grandes quantidades de dados para inferir implicações e relações entre atributos dentro um banco de dados. Essa técnica é aplicada a fim de inferir os valores de atributos que são mais prováveis na presença ou ausência de outros atributos.

Neste projeto, as técnicas de regras de associação podem ser aplicadas na identificação de produtos comumente comprados juntos, a fim de prever em que tipo de produto um cliente pode estar interessado caso já se tenha conhecimento de quais produtos costumam interessá-lo.

2.2 Estado da Arte

A aplicação de técnicas de *data mining* em supermercados e outros estabelecimentos varejistas é prática comum e muito explorada[1]. O objetivo mais comum dentre essas aplicações é a análise de carrinho de compras dos consumidores, ou seja, identificar quais são as relações de interdependência entre os produtos e categorias de produtos comprados por um cliente. O entendimento destas relações é muito importante para o desenvolvimento de estratégias de marketing, que visam atrair um número maior de compradores e assim aumentar o lucro do estabelecimento. Este tipo de problema é endereçado no meio acadêmico como análise de carrinho de compras (ou *Market Basket Analysis* em inglês).

Segundo (REUTTERER et al., 2006) existem duas principais linhas de pesquisa na análise de carrinho de compras. Pesquisas do tipo exploratória visam identificar inter-relações entre produtos e categorias de produtos, com base em padrões de compra observados (MILD; REUTTERER, 2003). Em literaturas de marketing, esta técnica é também chamada de "Análise de Afinidade". Já pesquisas do tipo explicativa (ou preditiva) buscam principalmente estimar quais serão os efeitos de uma determinada ação de marketing (CHIB; SEETHARAMAN; STRIJNEV, 2002; SEETHARAMAN et al., 2005).

A utilização de técnicas de *data mining* para a geração de promoções em

[1] `http://www.nytimes.com/2012/08/10/business/supermarkets-try-customizing-prices-for-shoppers.html`, acessado em 30 de novembro de 2012

estabelecimentos comerciais também já foi estudada em (YANG; HAO, 2011), e buscam identificar quais são os produtos cujas promoções darão um retorno financeiro máximo ao estabelecimento.

3 Especificação do Sistema Bom Cupom

3.1 Entendendo o Funcionamento de um Supermercado

O primeiro passo para construir um sistema de descontos inteligentes para supermercados é entender o seu funcionamento sem esse sistema. Nos interessamos aqui sobretudo pelos aspectos de abastecimento do supermercado e relacionamento com as marcas, pois constatamos que esses são os que mais influenciam na forma como os descontos são oferecidos atualmente.

Para esclarecer melhor esses aspectos, entrevistamos uma administradora de um supermercado de pequeno porte na região do bairo do Butantã, em São Paulo. Segundo ela, todos as promoções que o supermercado faz são fruto de um acordo com as marcas. Nesse acordo, o supermercado compra um determinado produto em quantidades maiores e recebe uma redução no valor unitário, permitindo que ele repasse o desconto ao consumidor.

Mesmo quando o desconto é negociado com o fornecedor, há uma perda na margem de lucro do produto que é inerente às promoções. Isto porque as promoções são publicadas no jornal de ofertas do supermercado, quinzenalmente, e abrangem todas as marcas de um mesmo produto, ao passo que a negociação é feita com apenas uma ou outra marca. No entanto a diminuição da margem unitária geralmente não excede 5%.

São raros os casos em que o supermercado diminui sua margem de venda unitária para oferecer promoções sem haver negociado com algum fornecedor. Levando em conta o fato de se tratar de um supermercado de pequeno porte, faz sentido imaginar que não se admitam grandes reduções nas margens, pois o volume de vendas pequeno não viabiliza financeiramente este tipo de ação.

Indagamos também se, do ponto de vista legal, descontos personalizados re-

presentariam um problema. Não seria um problema pois estes descontos funcionariam da mesma forma que os contra-vales que hoje são distribuídos em supermercados. Por outro lado, existe uma preocupação com possíveis reclamações de clientes que venham a se sentir prejudicados por receber menos desconto que outros. Outro aspecto relevante revelado durante a entrevista é que campanhas de promoção devem ser resgistradas legalmente, ao passo que concursos culturais não precisam.

Do ponto de vista do sistema de informação do supermercado, a administradora afirma que hoje um supermercado, ainda que de pequeno porte, não consegue operar sem o auxílio de um sistema de gerenciamento que proveja no mínimo um módulo de cadastro das vendas no caixa. Junta-se a isto um módulo de controle de estoque e um módulo financeiro.

Uma provável dificuldade de implantação do sistema Bom Cupom é a de que todos os descontos oferecidos no supermercado devem estar pré-cadastrados no sistema de gerenciamento pré-existente, não havendo flexibilidade para oferecer o desconto no momento da passagem pelo caixa. Por outro lado, esta dificuldade representa uma oportunidade de diferenciação, caso deseje-se desenvolver um sistema que faça o gerenciamento do supermercado integrado com o Bom Cupom.

As informações levantadas nessa fase nos ajudaram a entender melhor a dinâmica de um supermercado para a definição da especificação do Bom Cupom, além de revelar possíveis restrições a serem consideradas durante uma eventual implementação.

3.2 Definições e Termos

Antes de partirmos de fato para a especificação do sistema, definimos nesta seção alguns termos que serão usados frequentemente durante a descrição do projeto. O leitor pode se referir a esta seção quando tiver alguma dúvida sobre um dos termos usados no texto.

Avatar: um avatar (ou classe de produto) é um agrupamento funcional de produtos de mesmo tipo, que se diferenciam apenas pela marca, modelo ou tamanho da embalagem. Por exemplo, Coca-Cola 2L, Coca-Cola Lata, Fanta 2L e Guaraná Antarctica Lata pertencem ao avatar *Refrigerante*. Este agrupamento diminui a granularidade dos produtos. Ele é necessário por diversas razões:

i Melhora o desempenho dos algoritmos de *data mining*, diminuindo drasticamente as chances de que seu tempo de execução os torne proibitivos;

ii Permite a análise de hábitos de consumo independente de marcas ou modelos, correlacionando classes de produtos em vez de produtos individualmente;

iii Em muitas aplicações é difícil encontrar fortes associações entre itens em baixo nível de abstração, devido à dispersão dos dados neste nível. Com o agrupamento em avatares obtém-se resultados mais consistentes nas análises do que se for analisado produto por produto. Por outro lado, associações fortes encontradas em níveis de abstração muito altos podem representar nada mais do que o senso comum (TUNG et al., 2003).

Gama: denota o padrão da marca do produto, baseado no seu preço comparado aos outros produtos de mesmo avatar. Permite distinguir produtos comprados por clientes de maior ou menor poder aquisitivo. Produtos de gama alta são mais caros (e normalmente geram mais lucro) que os produtos de gama baixa, e atendem os clientes de maior poder aquisitivo. Importante notar que a definição de *margem de lucro* em economia é exatamente o valor de venda de um produto menos o valor de compra (a rigor deveríamos descontar outros custos de venda do produto). Naturalmente não temos acesso ao valor de compra dos produtos, e portanto, usaremos essa aproximação de que quanto mais caro um produto, maior será sua margem de lucro.

Nota fiscal: indica uma compra realizada por um cliente em uma determinada data. Cada vez que cada cliente finaliza uma compra, uma nota fiscal é gerada. Cada nota fiscal pode contar um número indeterminado de produtos, incluindo repetições.

Cesta básica de um cliente: é o conjunto de avatares já comprados pelo cliente no supermercado, ou seja, o conjunto de avatares para os quais existe pelo menos um produto que está presente em pelo menos uma nota fiscal do cliente.

Índice de recompra: denota a importância de um dado avatar dentro da cesta básica do cliente, baseado em uma série de parâmetros calculados sobre as notas fiscais. Para esse cálculo, consideram-se as várias notas fiscais das compras, anotando-se em quantas destas o produto está presente. O

cálculo deste índice esta descrito com mais detalhes no capítulo 7. Permite ordenar a importância dos produtos dentro da cesta básica do cliente e a probabilidade de que o produto venha a ser comprado novamente.

Grafo de regras de associação: é o conjunto de regras que denota a correlação entre dois avatares diferentes, no sentido em que quanto mais frequentemente eles são comprados juntos, maior é a correlação. Sua função é responder à seguinte pergunta: dado que o cliente compra um avatar A, quais são os avatares que o cliente irá provavelmente comprar junto, e com qual probabilidade? Para ilustrar a ideia, pode-se imaginar um grafo ordenado cujos nós são os avatares do sistema e cujas arestas do tipo $A \rightarrow B$ quantificam a intensidade da correlação entre os dois avatares, ou seja, as chances de que o avatar B seja comprado caso o avatar A tenha sido comprado (figura 3.1). Existem várias métricas para descrever esta intensidade, e estão descritas com mais detalhes no capítulo 6. Estas regras são obtidas através da análise de padrões nas notas fiscais no banco de dados do supermercado. É considerado o conjunto de todas as notas fiscais armazenadas no banco de dados do supermercado, sem distinção de clientes.

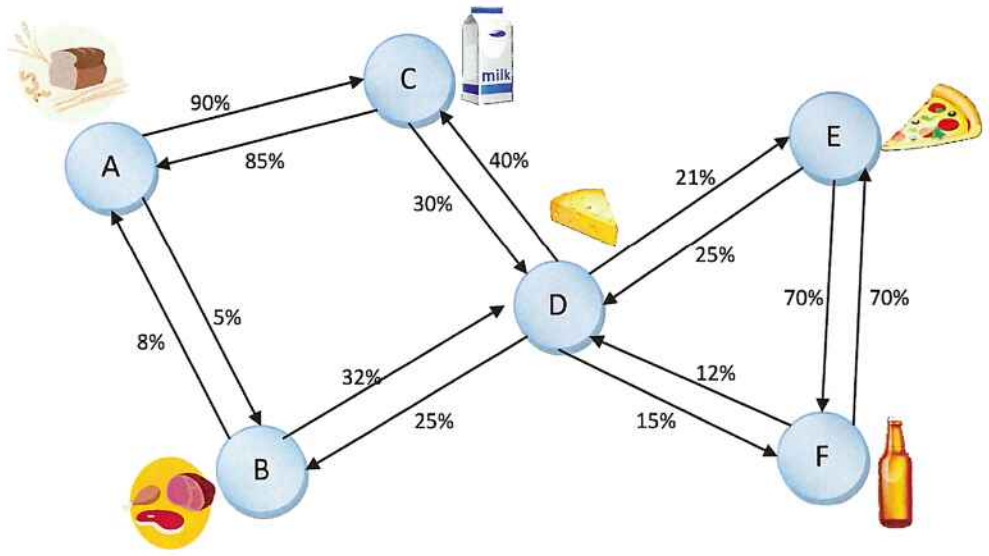

Figura 3.1: Grafo de regras de associação.
Cada aresta reta uma regra de associação.

3.3 Escopo do Projeto

A utilização das técnicas de *data mining* pode gerar conhecimento sobre os hábitos de consumo dos clientes, mas isto não é suficiente para viabilizar comercialmente

um sistema inteligente para de geração de cupons promocionais individualizados. Para que ele seja viável comercialmente, é desejável que o sistema possa responder a algumas questões:

- Conhecendo os hábitos de consumo dos clientes, em qual combinação de produtos devo oferecer-lhe desconto para que um produto que ele não compra tenha maior probabilidade de ser aceito ?
- Quanto desconto oferecer em cada produto?
- Qual é o ponto ótimo entre valor do desconto oferecido e retorno financeiro gerado pelo aumento do volume de compras?
- Como avaliar os efeitos dos descontos nos hábitos do consumidor? E como adaptar o modelo aos resultados destes efeitos (feedback) para gerar descontos mais adaptados continuamente?

O principal objetivo acadêmico deste trabalho é entender como os conceitos e técnicas de *data mining* podem ser aplicados na criação de um sistema de inteligência para supermercados. Os aspectos financeiros, embora sejam a principal motivação comercial de um tal sistema, não serão objeto de análise neste projeto. Para respondermos à útlima questão seria necessário que implantássemos o Bom Cupom de fato em um supermercado piloto e que coletássemos dados durante algum tempo, o que é inviável dentro do tempo de duração do projeto. Por isso concentramos nosso foco em responder somente a primeira questão.

Tendo isso em vista, um critério importante que adotamos na definição do escopo do projeto foi o de priorizar os módulos de um sistema de inteligência para supermercados que mais intimamente estão ligados à aplicação de conceitos de *data mining*, muito embora tenhamos analisado mais adiante também alguns aspectos ligados à implantação do mesmo em um supermercado real. Definimos o escopo do projeto limitando-se aos seguintes itens:

Itens dentro do escopo:

- Limpeza, qualificação e estruturação das informações do banco de dados de um supermercado;
- Classificação dos dados: Identificar e gerar grupos de produtos de um memso avatar;
- Desenvolvimento de um modelo para identificar os tipos de produtos comprados geralmente juntos;

- Desenvolvimento de um modelo para identificar os tipos de produtos mais frequentemente comprados por cada cliente;
- Desenvolvimento de um módulo que integre os dois modelos anteriores, capaz de gerar cupons de descontos individualizados com o auxílio de políticas de desconto pré-definidas.

Levantamos ainda outros itens que seriam necessários em um sistema implantado em um supermercado real. Decidiu-se, no entanto, que estes itens deveriam ficar fora do escopo deste projeto acadêmico. São eles:

Itens fora do escopo:

- Feedback: análise da reação do consumidor aos descontos oferecidos, em termos de aumento ou diminuição do volume de compras, para reajustar os parâmetros dos modelos;
- Cálculo da porcentagem de desconto ótima, considerando margem de venda vs. feedback do consumidor;
- Contabilização dos descontos oferecidos no fluxo de caixa do supermercado;
- Política de acúmulo de créditos para utilização do desconto;
- Aspectos ligados à implantação física do Bom Cupom em um supermercado real (totens, comunicação entre servidores, etc).

Dessa forma, ficou claro que o Bom Cupom não deve substituir o sistema de gestão de supermercado (SGS) que já estiver instalado, mas sim complementá-lo, trabalhando paralelamente. Essa limitação de escopo, embora necessária, nos leva a uma dificuldade de ordem prática: o Bom Cupom deve poder de alguma forma sincronizar-se com o sistema de gestão do supermercado (qualquer que ele seja), para ler periodicamente as vendas realizadas, os novos produtos cadastrados, entre outros dados.

Felizmente a legislação que institui e regulamenta o funcionamento da Nota Fiscal Paulista no estado de São Paulo obriga os estabelecimentos comerciais a informarem à Secretaria da Fazenda as transações realizadas com CPF na nota fiscal [1]. Por isso, a maioria dos sistemas de gerenciamento de supermercado feitos para supermercados de São Paulo já dispõem de uma funcionalidade que permite

[1] Mais detalhes em http://www.nfp.fazenda.sp.gov.br/legislacao.shtm

exportar os dados relativos às transações realizadas em formato de arquivo texto, especificado pela Secretaria da Fazenda.

Para eliminar esta necessidade de sincronização (e consequente dependência da legislação), o Bom Cupom deveria realizar toda a gestão financeira e operacional do supermercado, o que está fora do escopo deste projeto.

3.4 Requisitos Funcionais

Os requisitos funcionais do Sistema de Geração de Descontos Personalizados foram levantados observando-se as limitações do escopo do projeto. São eles:

- Sincronizar-se com o Sistema de Gestão do Supermercado;
- Classificar em Avatares os produtos que não estiverem classificados;
- Atualizar a cesta básica de cada cliente após cada sincronização;
- Atualizar o grafo de regras de associação periodicamente (período escolhido pelo supermercado);
- Emitir automaticamente um cupom de descontos personalizado a partir do fornecimento da identificação do cliente.

Os requisitos descritos anteriormente são uma referência para a primeira versão do Bom Cupom a ser implementada.

3.5 Requisitos Não Funcionais

A seguir é apresentada uma lista não exaustiva dos requisitos não funcionais do Bom Cupom. Esta lista é apresentada apenas como referência para uma eventual implementação em um supermercado real. No contexto acadêmico deste projeto, nos preocupamos em cumprir com os requisitos funcionais, mostrando as metodologias, ferramentas e desafios impostos por este projeto.

- A emissão de um cupom de desconto deve ser feita em tempo razoável após a solicitação do cliente (valor de referência: 5 segundos);
- O montante total de descontos oferecidos para um produto ou avatar não pode exceder um limite mensal (parametrizável, em dinheiro) estabelecido pelo supermercado;

- Os descontos oferecidos devem estar de acordo com as políticas de desconto do supermercado.

3.6 Modelo do Processo de Vendas com Desconto

Para complementar a especificação dos requisitos do Bom Cupom, imaginamos qual será o cenário de utilização do mesmo e com isso modelamos o principal processo no qual estará inserido quando estiver implantado em um supermercado. Este modelo é apresentado no pêndice A.

Embora o escopo do sistema neste projeto limite-se à emissão do cupom de desconto, buscamos modelar o processo completo. No apêndice A o escopo do projeto é indicado pelo retângulo vermelho na figura A.1.

Um aspecto importante a notar-se neste modelo é que, segundo nossa visão, o cliente do supermercado recebe o cupom de desconto na entrada, antes de começar suas compras (clientes que nunca fizeram compras no supermercado receberiam um cupom genérico, baseado no padrão de compras global do supermercado). Uma outra alternativa, a qual não exploramos em maiores detalhes neste projeto, seria imaginar um sistema onde o cupom é oferecido ao cliente logo após sua passagem pelo caixa.

3.7 Políticas de Desconto

As políticas de desconto são diretrizes utilizadas pelo Bom Cupom que devem ser seguidas para definir em quais produtos serão oferecidos descontos. Elas foram criadas para que seja possível oferecer descontos em produtos específicos, em função do efeito que o supermercado espera ter.

Um cupom pode ser gerado inteiramente segundo uma mesma política de desconto, ou combinando descontos gerados por políticas diferentes. No Bom Cupom, o mix de políticas de desconto utilizadas em um cupom a ser gerado deve ser personalizável pelo usuário (no caso, o analista do supermercado).

3.7.1 Política de Fidelização

Esta política consiste em oferecer desconto apenas em produtos que o cliente já tem o hábito de comprar, com o único propósito de fidelizá-lo. Para isto, o

Bom Cupom deve observar apenas a cesta básica do cliente, montar uma lista de produtos candidatos a desconto e, a partir da lista, escolher um ou mais produtos para compor o cupom de descontos.

3.7.2 Política de Upgrade de Marca

Esta política consiste em oferecer desconto em produtos de um avatar que o consumidor já tem o hábito de comprar, porém de uma gama superior à que ele normalmente compra.

Para isto, o Bom Cupom deve escolher um avatar da cesta básica de avatares do cliente (levando em conta os índices de recompra) e determinar qual é a gama de produtos que o cliente consome dentro deste avatar. O desconto será então oferecido em um produto da gama imediatamente superior, se existir. Se não houver uma gama imediatamente superior para o avatar escolhido, escolher o avatar seguinte e repetir o processo.

Exemplo:

1. Cliente compra muito o avatar pão.
2. Sistema verifica que a marca intermediária é a favorita deste cliente.
3. Sistema sugere um desconto na marca de pão premium.

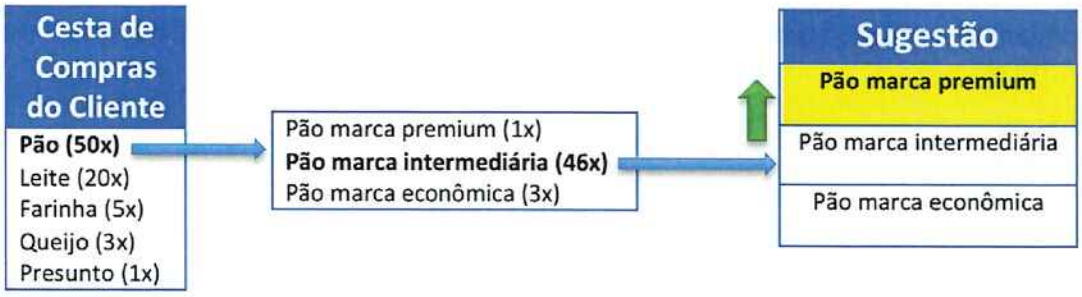

Figura 3.2: Processo de geração de cupom segundo a política de upgrade de marca

3.7.3 Política de Introdução de Produto

Esta política consiste em oferecer desconto em um produto de um avatar que o cliente não tem hábito de comprar, porém com grande probabilidade de interessá-lo se um desconto for oferecido.

Para isso, o Bom Cupom deve escolher inicialmente um dos avatares que o cliente mais consome (avatar original). Em seguida, observar a lista de avatares relacionados (baseado no grafo de propensão de compras), excluindo dessa lista aqueles avatares que ja estão na cesta básica do cliente. Dentre os avatares restantes, selecionar um e, dentro desse avatar, um produto cuja gama seja a mesma do avatar original.

Exemplo:

1. Cliente compra muito o avatar pão.
2. Sistema verifica que a marca intermediária é a favorita deste cliente.
3. Sistema procura produtos correlacionados com pão.
4. Sistema elimina produtos que o cliente já compra.
5. Sistema seleciona o avatar Manteiga.
6. Sistema sugere desconto no produto Manteiga de marca intermediária.

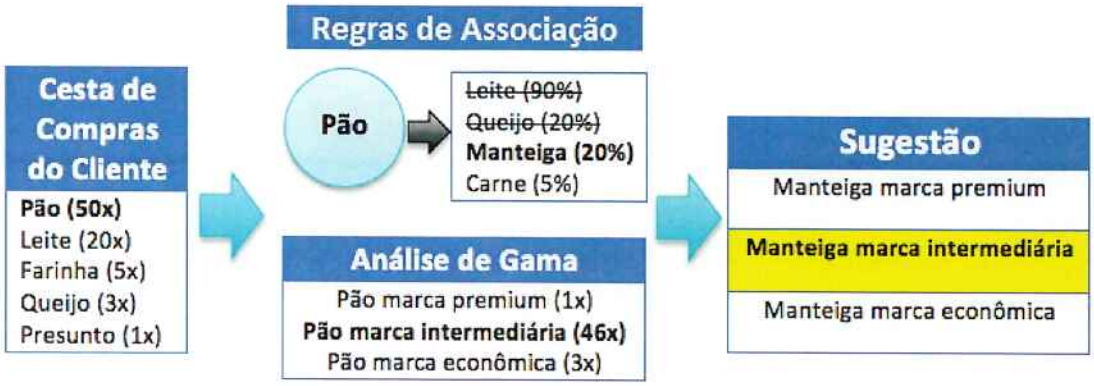

Figura 3.3: Processo de geração de cupom segundo a política de introdução de produto

3.8 Processo de Geração dos Cupons

3.8.1 Composição de um Cupom

A composição padrão de um cupom de desconto é uma sugestão do Bom Cupom ao responsável pelos descontos do supermercado. Ela contém:

- Desconto em um produto escolhido com a Política de Fidelização.
- Desconto em um produto escolhido com a Política de Upgrade de Marcas.

- Desconto em um produto escolhido com a Política de Introdução de Novos Produtos
- Desconto em um produto de um avatar com grande probabilidade de ser comprado com algum dos avatares que já compõem o cupom

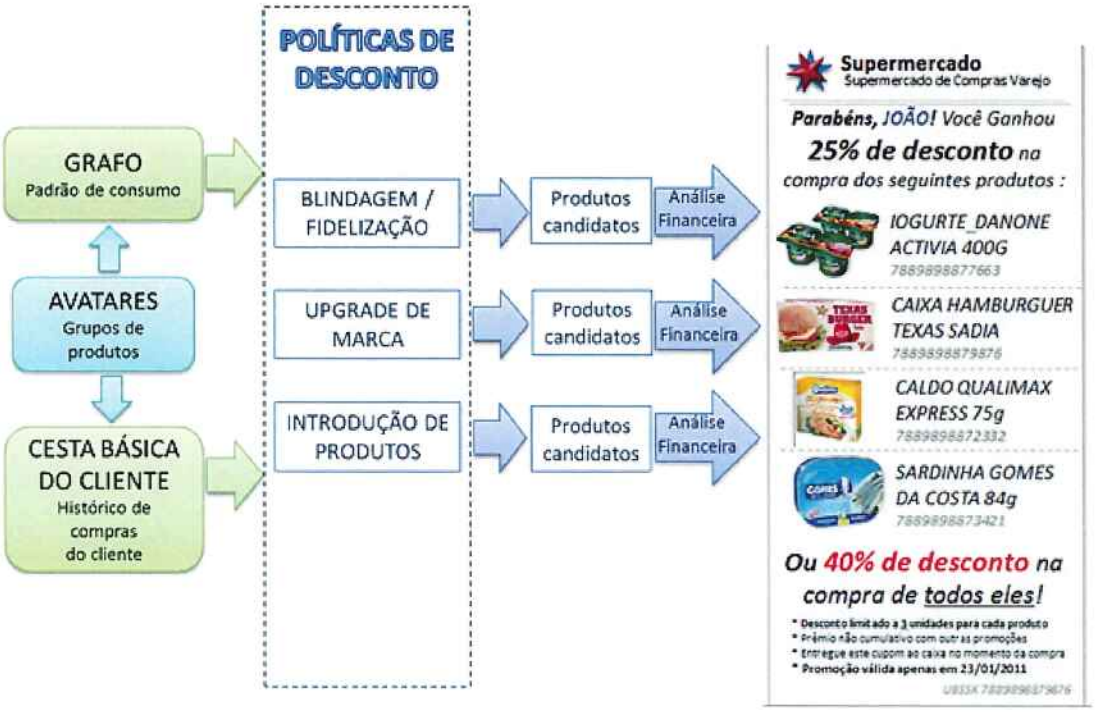

Figura 3.4: Processo geral de geração de um cupom

Esta composição não é obrigatória, e o responsável pelos descontos do supermercado pode personalizá-la segundo seus critérios. Cada produto individualmente terá uma porcentagem de desconto definida pela análise financeira. E caso o cliente compre todos os produtos do cupom, um desconto adicional é oferecido à todos os produtos do cupom. Como a análise financeira está fora do escopo deste projeto, para efeitos práticos serão adotados arbitrariamente descontos de 15% em cada produto individualmente, e um adicional de 10% em cada produto caso todos sejam comprados.

3.8.2 Emissão dos Cupons

Definem-se:

Cupom gerado: É todo cupom de descontos que tenha sido criado utilizando o processo de geração dos cupons e armazenado na base de dados. Um cupom gerado não necessariamente foi emitido. Ele apenas está preparado para ser emitido assim que solicitado.

Cupom emitido: É todo cupom previamente gerado que tenha sido impresso no totem e oferecido ao cliente.

Cupom utilizado: É todo cupom previamente emitido que tenha sido utilizado pelo cliente no ato do pagamento de sua compra.

Uma das questões que foram discutidas é em que momento os cupons de descontos devem ser gerados e emitidos. Duas possibilidades foram analisadas: 1) Gerar os cupons previamente e emiti-los apenas quando solicitado pelo cliente; 2) Gerar os cupons quando um cliente solicita um novo cupom, emitindo-os assim que gerados. Apesar de discutida, a resposta a esta questão está fora do escopo do projeto. A discussão é feita no intuito de compreender a complexidade do sistema como um todo.

As vantagens e desvantagens das duas abordagens foram analisadas na tabela 3.1:

	Vantagens	Desvantagens
Geração de cupons sob demanda	• Não há necessidade de armazenar cupons na base dados; • Não há necessidade de gerar cupons para todos os clientes; • Garante que os cupons são gerados sempre de acordo com os preços atuais e produtos existentes.	• Tempo de espera do cliente pelo processamento do cliente pode ser longo. • A estação deve permanecer online no ato da emissão.
Geração de cupons offline	• Emissao do cupom praticamente instantânea, sem necessidade de processamento; • A estação pode permanecer offline durante a emissão.	• Necessidade de verificação se o cupom gerado não está obsoleto ao emitir; • Exige a geração prévia de cupons para todos os clientes, o que dilui os descontos entre clientes que não vão necessariamente realizar compras no dia seguinte.

Tabela 3.1: Vantagens e desvantagens em função do momento de geração dos cupons

O ideal seria gerar cupons sob demanda, desde que o tempo para a geração e emissão não ultrapasse o valor de referência definido nos requisitos não funcionais. Esta verificação só é possível por meio de testes do tempo de execução. Outro

fator a se analisar é a disponibilidade de acesso aos servidores do banco de dados 24 horas por dia.

3.9 Casos de Uso

A modelagem do processo de utilização do Bom Cupom e a definição de algumas políticas de descontos permitiram identificar alguns casos de uso do sistema, que são descritos a seguir.

3.9.1 UC01 - Identificar Usuário

Ator Iniciante: Usuário no terminal do supermercado.

Evento Iniciante: Usuário solicita identificação.

Pré-Condições: Sistema e terminal devem estar operacionais.

Pós-Condições: Usuário identificado pelo sistema.

Regras de Negócio: -

Descrição:

1. Usuário solicita identificação.
2. Sistema solicita CPF.
3. Usuário entra com seu número de CPF.
4. Sistema busca na base de dados pelo CPF inserido.
5. Sistema exibe mensagem de sucesso e outras opções para o usuário.

Eventos alternativos:

1. No passo 4, se CPF ainda não constar na base de dados, sistema gera cupom para usuário não cadastrado (UC03)

3.9.2 UC02 - Emitir Cupom para Usuário Cadastrado

Ator Iniciante: Usuário no terminal do supermercado.

Evento Iniciante: Usuário solicita cupom via terminal.

Pré-Condições:

- Sistema e terminal devem estar operacionais.

- Usuário identificado pelo sistema.
- Todos os avatares do supermercado estão definidos.

Pós-Condições:

- Cupom emitido.
- Dados do cupom gravados no banco de dados.

Regras de Negócio:

- Quantidade máxima de descontos por produto / por mês, se houver.
- Quantidade máxima de descontos por dia.

Descrição:

1. Usuário envia solicitação de geração de um novo cupom.
2. Sistema gera a cesta básica do cliente identificado.
3. Sistema seleciona produto segundo a política de fidelização.
4. Sistema seleciona produto segundo a política de upgrade de marca.
5. Sistema seleciona produto segundo a política de introdução de novo produto.
6. Sistema imprime o cupom com as promoções geradas, segundo as diretrizes definidas pelo mercado.

3.9.3 UC03 - Emitir Cupom para Usuário Não Cadastrado

Ator Iniciante: Usuário no terminal do supermercado.

Evento Iniciante: Usuário solicita cupom via terminal.

Pré-Condições:

- Sistema e terminal devem estar operacionais.
- Todos os avatares do supermercado estão definidos.

Pós-Condições:

- Cupom emitido.
- Dados do cupom gravados no banco de dados.

Regras de Negócio:

- Quantidade máxima de descontos por produto / por mês, se houver.
- Quantidade máxima de descontos por dia.

Descrição:

1. Usuário envia solicitação de identificação.
2. Sistema constata que o usuário não existe no sistema.
3. Sistema cadastra o CPF no banco de dados.
4. Sistema gera e imprime um cupom de desconto padrão segundo as diretrizes definidas pelo mercado.

4 Desenvolvimento e Implementação

4.1 O Banco de Dados

Nesta seção descreveremos o banco de dados com o qual trabalhamos durante o projeto.

O banco de dados foi obtido de um supermercado regional de médio porte (faturamento anual entre R$ 1 MM e R$ 10 MM) que demonstrou interesse na aplicação do sistema e aceitou colaborar com o projeto. Tivemos acesso aos dados das vendas realizadas em 26 dias diferentes, escolhidos ao acaso, durante o período entre abril de 2009 e maio de 2010. Por uma questão de privacidade, o CPF dos clientes foi removido da base de dados, e no lugar foi utilizado um campo que identifica unicamente cada cliente (idcliente), sem prejuízo à capacidade de personalização dos cupons pelo Bom Cupom.

O banco de dados foi inicialmente fornecido no formato de arquivos texto, que foram exportados pelo sistema de gerenciamento do supermercado. Estes arquivos texto tiveram que ser analisados e estruturados em tabelas relacionais. Nossa equipe projeto teve acesso às tabelas relacionais já estruturadas em um banco de dados MySQL. Coube à nossa equipe projeto verificar a integridade dos dados presentes nestas tabelas e fazer as correções necessárias para garantir a integridade dos mesmos (remover os registros duplicados, remover chaves estrangeiras sem uma chave primária correspondente, etc). O modelo lógico do banco de dados que nos foi fornecido é mostrado na figura 4.1.

A extração dos dados e estruturação dos mesmos em tabelas relacionais, conforme o modelo anterior, cumpre o requisito funcional de sincronização com o sistema de gerenciamento do supermercado.

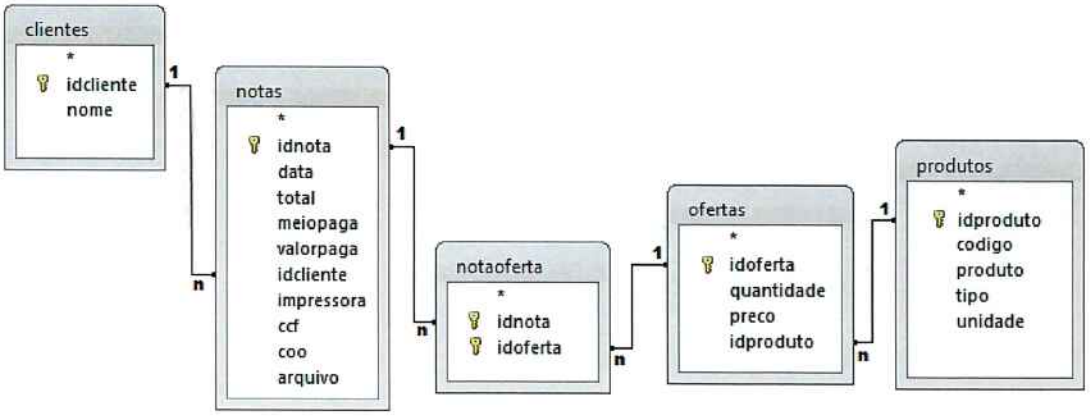

Figura 4.1: Modelo lógico do banco de dados original

4.2 Módulos do Sistema

A partir dos requisitos funcionais do Bom Cupom, o desenvolvimento do projeto foi dividido em quatro módulos, cada um referindo-se a um requisito funcional, com um subproduto bem definido [1]. A figura 4.2 ilustra a divisão do Bom Cupom nos seus respectivos módulos.

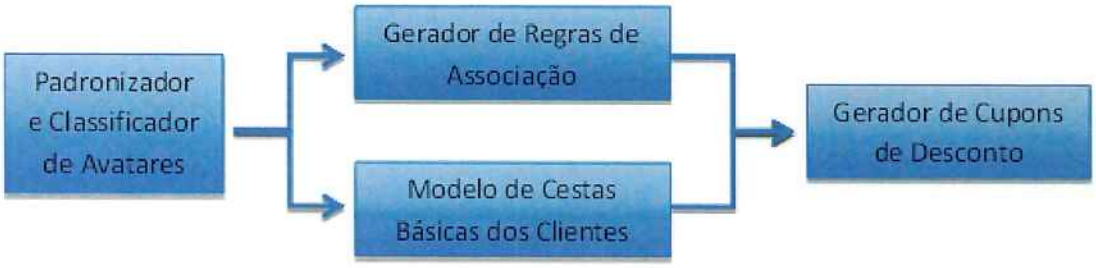

Figura 4.2: Diagrama de blocos do Bom Cupom

Para cada módulo, definimos de antemão suas entradas e saídas (resultados esperados), tratando-os como unidades independentes. Desta forma as interfaces entre os módulos ficaram bem definidas, o que facilitou sua posterior integração. Ficou definido também que cada membro do grupo seria responsável por um módulo (vide seção 4.2.1), e que os módulos seriam desenvolvidos paralelamente, observando-se a interface entre eles.

4.2.1 Interfaces dos Módulos do Sistema

Classificador de Avatares:

Entrada: um conjunto não-uniforme de produtos com nome (abreviado e truncado) e seu respectivo código de barras (que podem seguir diferentes ou até

[1] O requisito de sincronização com o sistema de gerenciamento do supermercado fora tratado quando da extração e estruturação dos dados em tabelas relacionais, na seção 4.1

mesmo nenhum padrão).

Saída: banco de dados padronizado segundo a especificação do diagrama entidade-relacionamento presente no documento de especificações finais, com produtos agrupados em categorias denominadas avatares.

Responsável: Igor Gushiken

Gerador de Regras de Associação:

Entrada: Tabelas do banco de dados padronizadas, segundo especificado na interface de saída do Padronizador e Classificador de Avatares.

Saída: Um conjunto de regras de associação entre os avatares, com pelo menos as seguintes métricas: suporte, confiança, melhoria e convicção.

Responsável: Bruno Joia

Identificador de Cestas Básicas de Clientes:

Entrada: Tabelas do banco de dados padronizadas, segundo especificado na interface de saída do Classificador de Avatares.

Saída: Um conjunto de cestas básicas de clientes, onde cada cesta básica deve explicitar a probabilidade de que um dado avatar já comprado pelo cliente seja adquirido novamente na sua próxima compra.

Responsável: Gustavo Rocha

Gerador de Cupons de Desconto:

Entrada: Regras de Associação entre Avatares e Cestas Básicas dos Clientes e identificação de um cliente.

Saída: Um produto específico para cada política de desconto para o cliente que solicitou o cupom.

Responsável: Todos participam

4.2.2 Preparação para o Desenvolvimento dos Módulos

Um passo importante que viabilizou a paralelização do desenvolvimento do sistema foi a classificação manual que fizemos de cada produto do banco de dados em um avatar e um superavatar. Esta classificação manual foi necessária para permitir que o Gerador de Regras de Associação e o Identificador de Cestas Básicas pudessem ser desenvolvidos e testados com avatares antes que o Classificador de Avatares estivesse pronto.

Nesta etapa de classificação manual, foram adicionadas duas tabelas ao banco de dados original: avatares e superavatares. Um superavatar é um agrupamento de avatares similares. Por exemplo, os avatares LARANJA, MAÇÃ e PÊRA são todos associados ao superavatar frutas. A classificação em superavatares, num nível de abstração superior, mostrou-se necessária para para permitir testar o Gerador de Regras de Associação e validar a regras geradas em dois níveis de abstração.

A figura 4.3 mostra como ficou o modelo lógico do banco de dados após a classificação manual dos produtos em avatares e superavatares.

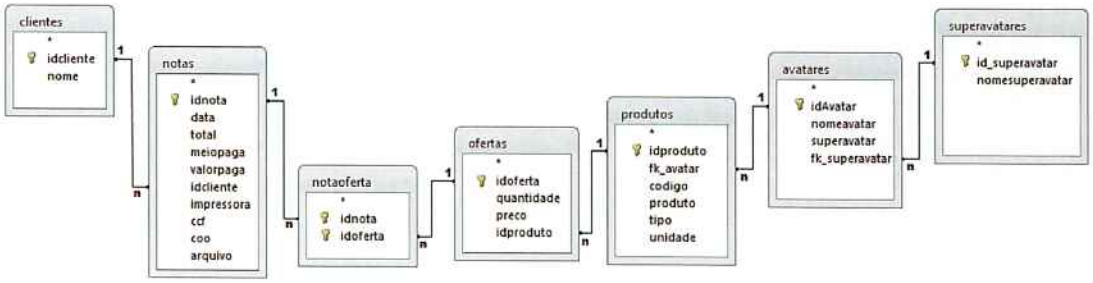

Figura 4.3: Modelo lógico do banco de dados após a classificação manual dos produtos

Os capítulos seguintes entram nos detalhes do desenvolvimento de cada um dos módulos do sistema, provendo os conceitos mais específicos a cada um e os resultados obtidos em cada módulo.

5 Classificador de Avatares

5.1 Contexto

Um supermercado possui uma grande quantidade de produtos, com alta rotatividade. Cada variação do produto como tamanho, embalagem ou sabor recebe um código de barras distinto, e é tratado pelo sistema do supermercado como um produto distinto.

Código de Barras	Produto
509546028019	CD.COLGATE 100G TOTAL
7891024111130	CD.COLGATE 180G TOT.CLE
7891024132906	CD.COLGATE 50G MAXIMA P
7891024135358	CD.COLGATE 70G TOTAL 12
7891024145302	CD.COLGATE 90G BICARB.S

Tabela 5.1: Exemplos de cremes dentais presentes no banco de dados de um supermercado

Devido ao grande número de produtos existentes na base de dados, uma análise da correlação de compra no nível dos produtos gerará resultados dispersos e pouco significativos. Este problema pode ser resolvido através do agrupamento dos produtos em categorias. Desta forma, a análise da correlação de compra entre categorias mais generalistas trará resultados mais significativos.

Para agrupar os produtos em categorias, métodos de classificação são apresentados na seções 5.4 e 5.5. A seção 5.7 contém detalhes da implementação de um programa classificador para os produtos de um supermercado de pequeno porte. A seção 5.8 traz os resultados dessa implementação e a sua análise.

5.2 Objetivo

O objetivo deste módulo é desenvolver um classificador dos produtos do banco de dados de trabalho, a fim de permitir que as análises realizadas sobre os produtos possam gerar resultados mais significativos. Após a classificação cada produto

deve ser associado a um avatar). Cada avatar deve conter um conjunto de produtos similares, que se diferenciem apenas pela marca, preço e outras características mais específicas dos produtos. Por exemplo, o avatar REFRIGERANTE agrupará todos os tipos de refrigerantes, independente de marca, preço ou volume da embalagem.

Figura 5.1: Entradas e saídas do classificador

Todo banco de dados de supermercados possui pelo menos duas informações sobre seus produtos: código de barras e nome. Essas são informações primordiais para o sistema do caixa e para a geração de recibos e notas fiscais. No entanto, como demonstrado na tabela 5.1, os códigos de barras podem seguir diferentes padrões, e os nomes podem estar abreviados ou truncados.

O classificador aqui introduzido deve ser capaz de atribuir uma categoria a cada um dos produtos do supermercado com base no nome e no código de barras de cada produto, armazenados em um Banco de Dados ou em um arquivo, salvando seus resultados em um banco de dados estruturado.

Nome do Produto	Categoria
SAB FRANCIS 90G AVEIA T SAB PROTEX 90G CREAM PE SAB PHEBO 90G ODOR ROSA	Sabonete
BISC TOSTINES 200G AGUA BISC TRIUNFO 200G AGUA	Biscoito de Água e Sal

Tabela 5.2: Exemplos de classificação de produtos

5.3 Funcionamento do Código de Barras

A fim de dar ao leitor uma visão mais ampla do assunto, e de deixar mais claro qual é o desafio enfrentado neste módulo, antes de entrarmos em mais detalhes de como estruturamos nosso classificador, fazemos nesta seção a introdução de alguns conceitos que julgamos importantes sobre os códigos de barra e alguns padrões existentes.

Criados nos anos 70, os códigos de barra são basicamente uma sequência variável de barras paralelas brancas e pretas que representam uma determinada

informação. São comumente utilizados para representar uma numeração (identificação) que é atribuída a produtos, documentos e ativos em geral. Por permitir a automatização do processo de identificação dos produtos nos caixas, a tecnologia de código de barras é amplamente utilizada nos supermercados. Os códigos de barra funcionam a partir do princípio de refletividade da luz. O leitor de códigos de barra emite sobre o código a ser lido um feixe de luz, que é refletido de volta ao leitor apenas pelas barras brancas. A luz refletida é então transformada em sinais elétricos, que é posteriormente digitalizada.

5.3.1 Simbologias

Simbologia é o termo utilizado para descrever as regras que especificam a maneira na qual os dados devem ser convertidos em barras e espaços de variados tamanhos que compõem um código de barras. Assim como em uma linguagem, para que a comunicação entre a entidade que imprimiu o código de barras e o leitor de código de barras ocorra corretamente, é preciso que ambos utilizem a mesma simbologia. Existem atualmente dezenas de simbologias de código de barras. Apresentamos a seguir as principais utilizadas no varejo.

5.3.2 UPC

O *Universal Product Code* (UPC) é uma simbologia de código de barras amplamente utilizado na América do Norte, Reino Unido, Austrália, Nova Zelândia e outros países para o rastreamento de produtos em lojas. O UPC é uma simbologia contínua (ou seja, não há espaços entre os caracteres codificados), numérica, de comprimento fixo, com suporte a quatro espessuras diferentes de barra ou espaço. Dois tipos comuns de UPC são o UPC tipo A, que contém 12 dígitos, e o UPC tipo E, que contém 6 dígitos.

5.3.3 UPC-A

UPC-A é o tipo mais comum da simbologia UPC e possui 12 dígitos. Um código UPC-A é dividido em quatro partes:

- Sistema de numeração (um dígito)
- Código do fornecedor (cinco dígitos)
- Código do produto (cinco dígitos)

- Dígito de controle (um dígito)

O dígito de sistema de numeração indica o tipo do produto, de acordo com a numeração abaixo:

0 = códigos normais

1 = reservado

2 = produtos de peso variável (marcados na loja)

3 = produto farmacêutico

4 = uso livre dentro da loja

5 = cupons

6 = reservado

7 = códigos normais

8 = reservado

9 = reservado

Os cinco dígitos do código do fornecedor são estabelecidos por uma entidade reguladora, para que não haja colisão entre códigos de barra. Já os cinco dígitos relativos ao código do produto são escolhidos pelo fornecedor, de acordo com suas próprias políticas.

O dígito de controle é calculado da seguinte forma:

1. Somam-se os dígitos das posições ímpares (primeira, terceira, quinta, etc) e o resultado é multiplicado por três.

2. Soma-se os dígitos das posições pares (segunda, quarta, sexta, etc.) ao resultado de (1).

3. Calcula-se o módulo 10 do resultado de (2).

4. Se o resultado de (3) for diferente de zero, subtraí-lo de 10. Este será o dígito de controle.

5.3.4 EAN

O padrão EAN (originalmente *European Article Numbering*, hoje *International Article Number* - a sigla foi mantida) foi criado para ser um padrão europeu, porém hoje é utilizado mundialmente. O EAN é um superconjunto do padrão americano UPC, o que faz com que leitores de códigos de barra EAN sejam capazes de decodificar código de barras UPC – embora o inverso não seja necessariamente verdadeiro. O EAN possui duas versões: EAN-13 e EAN-8, que possuem 13 e 8 dígitos, respectivamente.

5.3.5 EAN-13

Assim como o UPC-A, o EAN-13 é composto por quatro partes:

- Prefixo GS1 (três dígitos)
- Código do fornecedor (três a oito dígitos)
- Código do produto (dois a seis dígitos)
- Dígito de controle (um dígito)

O prefixo GS1 indica qual é a filial da GS1 na qual o código de barras foi cadastrado. Em termos práticos, o prefixo GS1 geralmente indica o país onde o produto foi fabricado. O dígito de controle é calculado da mesma forma que na simbologia UPC.

5.3.6 EAN-8

O EAN-8 é uma versão reduzida do EAN-13, e é utilizado em pequenas mercadorias, como cigarros e doces. Ele é composto de apenas três partes:

- Prefixo GS1 (2 ou 3 dígitos)
- Código da mercadoria (4 ou 5 dígitos)
- Dígito de controle

O dígito de controle é calculado da mesma forma que no EAN-13.

5.4 Classificação por Código de Barras

A escolha pelo atributo código de barras como o primeiro critério para classificação dos produtos de nossa base de dados é natural, pois ele identifica unicamente o produto no meio comercial. A nossa primeira idéia de abordagem seria utilizar técnicas de aprendizado não supervisionado, como clustering, para agrupar os produtos segundo seus códigos de barras. No entanto, descartamos esta abordagem porque existem diversas codificações diferentes que podem ser usadas nos códigos de barras de produtos, conforme expusemos na seção 5.3. Além disso, para aplicarmos esta técnica, deveria ser possível calcular de alguma forma a similaridade dos códigos de barra. No entanto, os código de barras de cada produto são definidos pelos fabricantes segundo critérios que não necessariamente levam em conta a similaridade dos produtos, isto é, produtos similares não necessariamente terão códigos de barras similares. Por estas razões, optamos por uma abordagem diferente, baseada na busca de informação disponibilizada em bases de dados de terceiros na internet, conforme explicaremos a seguir.

5.4.1 Classificação através da API Buscapé

A API do Buscapé [1] é um conjunto de serviços oferecido pela Buscapé Company, empresa especializada em pesquisa de preços de produtos. Através dessa API, é possível obter acesso aos produtos, ofertas e serviços oferecidos pelo Buscapé. A API utiliza tecnologia REST[2], sendo facilmente acessada por navegador, linha de comando ou código por meio de uma URL.

Uma das grandes vantagens deste sistema é que o Buscapé agrega muitas lojas brasileiras, possuindo um grande número de produtos nacionais já cadastrados em sua base de dados. Um de seus serviços disponíveis é o de busca de ofertas por código de barras, que pode ser usado de forma a se obter a categoria de um produto. A partir do código de barras de um produto, é possível obter uma lista de ofertas, que contém, além da categoria do produto, seu nome, preço, entre outras informações.

No apêndice B está apresentado um exemplo de requisição e do arquivo XML recebido como resposta. Em seguida pode-se ver o trecho que indica a categoria do produto buscado como exemplo no apêndice. A tag XML ¡name¿ dentro da tag ¡category¿ indica exatamente a informação que estamos procurando:

[1] `http://developer.buscape.com/api/`
[2] `http://en.wikipedia.org/wiki/Representational_state_transfer`

```
...
<category hasOffer="true" isFinal="true"
parentCategoryId="517" id="3261">
  ...
  <name>Bebida Achocolatada</name>
</category>
...
```

5.5 Classificação por Semelhança do Nome

Conforme exposto na seção 5.8 nem todos os produtos podem ser classificados usando a API Buscapé com o código de barras, seja porque os códigos de barras no banco de dados com o qual trabalhamos está impreciso em alguns produtos, seja porque nem todos os produtos procurados estão disponíveis na base de dados do Buscapé. Para os produtos que não puderam ser classificados usando seu código de barras, foi tomada uma abordagem de comparação dos nomes dos produtos.

Mais especificamente, essa abordagem visa classificar os produtos restantes através da comparação de seus nomes com os nomes daqueles que já foram classificados usando seus códigos de barras. Ou seja, trata-se da realização de uma análise de semelhança entre strings de produtos classificados e não classificados. Esta abordagem é um caso específico de aprendizado supervisionado pois já dispomos de um conjunto de exemplos corretamente classificados por hipótese e desejamos extender essa classificação a exemplos novos ainda não classificados. Esse método especificamente trata-se de um aprendizado baseado em instância, pois a cada exemplo novo (não classificado) será atribuída uma classe de acordo com a proximidade e similaridade dele em relação a todos os exemplos já classificados.

Nesta seção vamos descrever o processo de análise de semelhança léxica entre os nomes dos produtos, enquanto na seção 5.6 exploramos a possibilidade de análise de semelhança semântica, detalhada no apêndice C.

Para ilustrar melhor a abordagem, observe a tabela 5.3, que contém um produto classificado e um não classificado:

Nota-se que ambos os produtos são Desodorantes Rexona, o que pode ser verificado pela semelhança de seus nomes, embora apenas o primeiro produto já esteja classificado. A partir da comparação dos nomes, utilizando técnicas que

Nome do Produto	Categoria
DES.REXONA 50ML ROLL-ON	Desodorante
DES.REXONA 90ML SPRAY A	SEM CATEGORIA

Tabela 5.3: Exemplo de classificação por análise de semelhança ortográfica

descreveremos a seguir, podemos categorizar o produto dois como "Desodorante". As seções à seguir apresentam métodos para calcular a semelhança entre duas strings (dois nomes de produto).

5.5.1 Longest Common Subsequence (LCS)

Este algoritmo consiste em achar a maior sequência comum entre duas strings, comparando-as caractere por caractere. Por exemplo, dadas as strings:

SOP. BABY NESTLE 115
PAPINHA BABY NESTLE 120

A maior sequência comum é "BABY NESTLE 1" .

5.5.2 Distância de Levenshtein

Este algoritmo consiste em calcular a diferença entre duas strings considerando-se o número mínimo de edições (inserção, remoção ou substituição de um único caractere) necessário para que as duas strings se tornem iguais. Por exemplo, dadas as strings:

SOP. BABY NESTLE 115
PAPINHA BABY NESTLE 120

São necessárias apenas 8 edições para que a primeira se iguale à segunda:

1) Substitui-se o S por P [POP. BABY NESTLE 115]
2) Substitui-se o O por A [PAP. BABY NESTLE 115]
3) Substitui-se o . por I [PAPI BABY NESTLE 115]
4) Acrescenta-se N após o I [PAPIN BABY NESTLE 115]
5) Acrescenta-se H após o N [PAPINH BABY NESTLE 115]
6) Acrescenta-se A após o H [PAPINHA BABY NESTLE 115]
7) Substitui-se o 1 por 2. [PAPINHA BABY NESTLE 125]
8) Substitui-se o 5 por 0. [PAPINHA BABY NESTLE 120]

Este método é muito útil para detectarem-se erros de digitação. Uma de suas propriedades é que uma diferença em qualquer posição da String tem o

mesmo peso. No entanto, no caso de produtos de um supermercado, normalmente encontramos em primeiro lugar o nome do produto, por exemplo:

SAL CISNE 500G TRADICIO
SAL LEBRE REFINADO 1K
SAL IODADO JASMINE 1000

Nota-se que a única palavra em comum entre esses produtos é a sequência SAL, de apenas 3 caracteres. Desta forma, sua distância de Levenshtein será muito grande, assim como sua maior sequência comum será muito pequena.

No entanto, fica claro que esses produtos correspondem a uma mesma categoria (Sal). Conclui-se que os caracteres do início da string devem ter maior relevância para o cálculo da distância entre duas strings. Deve-se, portanto, utilizar um algoritmo que leve esta condição em consideração. A seção 5.7 descreve em detalhes como esse problema foi resolvido.

5.6 Classificação por Semelhança Semântica

Uma outra abordagem para a classificação dos produtos que não puderam ser classificados é a comparação por semelhança semântica. Para tal são necessárias estruturas que associam termos através de relações semânticas e a principal estrutura deste tipo é a WordNet.

Um estudo aprofundado foi feito com relação à viabilidade da WordNet neste projeto, porém, devido a algumas limitações não foi possivel o emprego dos métodos de classificação por semelhança semântica. Mais detalhes sobre WordNet e suas limitações estão descritos no apêndice C.

5.7 Implementação

Com base nas técnicas de classificação apresentadas, foi realizada a implementação de um classificador. Esse classificador é composto por duas grandes partes, como ilustrado na figura 5.2:

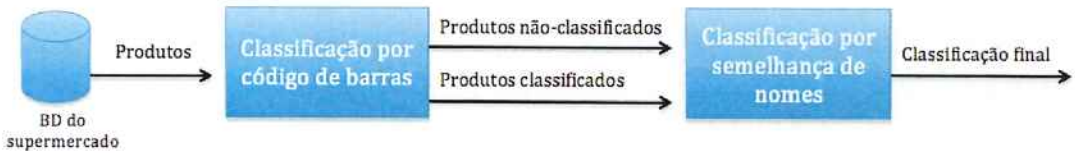

Figura 5.2: Fluxo de informação do classificador proposto

Todos os produtos da base passam primeiramente por um módulo de classificação por código de barras. Esta técnica é a primeira a ser aplicada por ser mais assertiva, pois os códigos de barra são únicos. Ela também permite que classifiquemos produtos de nomes diferentes, porém de mesmo valor semântico, na mesma categoria (ex: Bolacha e Biscoito). Em um segundo momento, os produtos não classificados pelo módulo de classificação por código de barras são classificados pelo método de inferência por semelhança de strings.

5.7.1 Implementação da Classificação por código de barras

O primeiro passo da implementação do classificador por código de barras foi a homogenização do banco de dados do supermercado. Isso foi necessário pois o banco de dados utilizado possuía produtos contendo diversos padrões diferentes de códigos de barra. Um fator agravante foi o fato de que o banco de dados utilizado pelo supermercado só suportava 12 dígitos de código de barras, o que fez com que códigos de barra do padrão EAN-13 (13 dígitos) tivessem o seu primeiro dígito truncado. Este fato foi facilmente minimizado pois todos códigos de barra de produtos fabricados no Brasil começam com os dígitos 789 ou 790. Um segundo fator agravante foi que o banco de dados excluía zeros à esquerda, o que fazia com que códigos de barra do tipo UPC-A (12 dígitos) que começassem com o dígito zero perdessem o primeiro dígito, resultado em códigos com 11 dígitos ou menos. A tabela 5.4 ilustra a estratégia adotada para a homogenização dos códigos de barra:

N° de dígitos	Dígitos iniciais	Estratégia Adotada
12	89 ou 90	Anexar o dígito 7 no início no código e tratá-lo como EAN-13
12	indiferente	Tratar o código como UPC-A
9 a 11	indiferente	Anexar zeros à esquerda e tratar código como UPC-A
9 a 11	indiferente	Anexar zeros à esquerda e tratar código como UPC-A
8	indiferente	Tratar código como EAN-8
6	indiferente	Tratar código como UPC-E
menos que 6	indiferente	Códigos de barra criado no supermercado. Não usar o classificador por códigos de barra.

Tabela 5.4: Estratégias de homogenização dos códigos de barras

Após a homogenização dos códigos de barras, foi desenvolvido um programa em linguagem Ruby que recebe como entrada um arquivo CSV com todos os pro-

dutos do supermercado. Este programa faz, para cada produto, uma requisição HTTP GET à API do Buscapé. A resposta, em formato XML, é analisada para verificar se o Buscapé possui este produto (categoria e nome completo) em sua base de dados.

5.7.2 Implementação da Classificação por inferência por semelhança de nomes

Para calcular a semelhança entre os nomes de dois produtos, foram testados os seguintes métodos de comparação de string:

Distância de Levenshtein entre os nomes dos produtos

Neste método, o algoritmo de cálculo de distância de Levenshtein foi aplicado entre todos os pares de nomes de produtos não-classificados com nomes de produtos já classificados. Apesar de ser muito útil para detecção de erros de digitação, este método não foi muito eficiente neste caso pois, além da base de produtos possuir um número significante de erros de digitação, os caracteres do final do nome do produto são pouco significativos.

Longest Common Subsequence entre os nomes dos produtos

Neste método, o algoritmo de cálculo de *Longest Common Subsequence* foi aplicado entre todos os pares de nomes de produtos não classificados com nomes de produtos já classificados. Apesar desse algoritmo classificar corretamente um bom número de produtos, o fato dos últimos caracteres do nome do produto serem menos significativo faz com que alguns produtos sejam classificados de forma errada. Por exemplo:

LEITE SHEFA 1 LT.
IOG SHEFA 1 LT.

São dois produtos que possuem uma grande sequência comum ("SHEFA 1 LT."), trata-se, porém, de dois produtos de categorias diferentes (LEITE e IOGURTE, respectivamente). Nota-se, novamente, que deve-se dar maior importância à similaridade presente no início do nome do produto.

LCS entre os primeiros caracteres dos nomes dos produtos

Neste método, o algoritmo de cálculo de *Longest Common Subsequence* foi aplicado da mesma forma que no método de Longest Common Subsequence entre os nomes dos produtos, porém apenas os primeiros 8 caracteres dos nomes dos produtos foram considerados na comparação. Este foi o método com melhor taxa de acerto para os produtos do supermercado estudado.

pois as palavras que descrevem o produto estão sempre no começo do seu nome, enquanto detalhes como marca ou tamanho da embalagem estão sempre situadas no final do nome.

5.8 Resultados

A classificação por código de barras (Buscapé API) classificou com sucesso 4353 dos 10754 produtos da base (aproximadamente 40%) em 28/09/2012 e 3056 dos 10754 produtos (aproximadamente 28%) em 22/10/2012. Com esta diferença notou-se que os produtos da base de dados do supermercado tornam-se obsoletos com o passar do tempo. Isso ocorre porque a API do Buscapé retorna apenas os produtos que possuem uma oferta ativa em algum de seus sites parceiros, ou seja, apenas os produtos que estão sendo vendidos em um determinado momento. Como os produtos de um supermercado são atualizados com frequência (novo formato, nova embalagem, etc), alguns produtos acabam sendo substituídos por outros similares, porém com outro código de barras. Portanto, a classificação de produtos de um supermercado deve ser feita periodicamente. A tabela 5.6 apresenta algumas classificações que foram feitas apenas com base na classificação por código de barras:

Código de Barras	Nome do Produto	Avatar
7894321722016	ACHOC LIQ TODDYNHO 200M	Bebida Achocolatada
7894321619033	AZEIT.RAIOLA PRETA 200G	Azeitona em Conserva
7892840176815	BAT ELMA CHIPS 100G PAL	Salgadinho
7897600306006	OLEO GERG.HONG 100ML NA	Outros Temperos

Tabela 5.5: Produtos classificados pelo código de barras

O método de *Longest Common Subsequence*, aplicado somente aos primeiros caracteres dos nomes dos produtos, foi o método utilizado no módulo de classificação por semelhança entre nomes. Com esse algoritmo, foram inferidas as categorias de 5569 produtos, totalizando 8625 produtos classificados (aproximadamente 80% da base). Este resultado nos permite aplicar regras de associação entre os produtos do supermercado, como demonstrado no capítulo 6. A tabela 5.6 apresenta exemplos de classificações feitas a partir do método de classificação por semelhança de nomes:

Os produtos que não puderam ser classificados com esta metodologia não poderão ser utilizados nos demais módulos do Bom Cupom e, portanto, não poderão estar presentes em cupons de desconto. Para garantir que 100% dos produtos sejam candidatos a aparecerem em cupons promocionais, pode-se utilizar

Código de Barras	Nome do Produto	Nome Semelhante	Avatar
7891000379707	ACHOC LIQ NESCAU 1LT	ACHOC LIQ TODDYNHO 200M	Bebida Achocolatada
7892999161007	ACHOC LIQ LECO 1L	ACHOC LIQ TODDYNHO 200M	Bebida Achocolatada
70404005444	AZEIT.MUSA 500G VERDE	AZEIT.RAIOLA PRETA 200G	Azeitona em Conserva
20991	AZEIT.PRETA PORTUGUESA	AZEIT.RAIOLA PRETA 200G	Azeitona em Conserva

Tabela 5.6: Produtos classificados por semelhança dos nomes

a classificação manual que foi feita na preparação para o desenvolvimento, como explicado na seção 4.1.

6 Gerador de Regras de Associação

6.1 Objetivo

O objetivo deste módulo é gerar um conjunto de regras de associação, onde cada regra representa a força de associação entre dois avatares presentes na base de dados, isto é, uma medida da frequência com que dois avatares são comprados juntos na mesma nota fiscal. Para isto, o gerador de regras de associação deve analisar especificamente a tabela de vendas do supermercado, onde constam todas as notas fiscais emitidas, seus respectivos produtos e preços.

Portanto, o problema que este módulo deve resolver é o da mineração de padrões frequentes. Este tipo de problema é bastante explorado na literatura e existem atualmente diversos algoritmos de *data mining* destinados a resolver esse problema. Dito isto, o desafio técnico deste módulo pode ser dividido em duas partes: (a) identificar na literatura e programar um sistema capaz de utilizar os algoritmos mais indicados para realização desta tarefa, dadas as especificidades de nossa base de dados e os requisitos não funcionais do Bom Cupom, e (b) escolher as métricas de qualidade que devemos usar para fazer a correta interpretação das regras geradas.

6.2 Técnicas Existentes

Como dito anteriormente, uma regra de associação é uma regra que avalia o quão forte é a probabilidade de dois itens serem encontrados simultaneamente numa mesma transação, evidenciando um padrão. Entendemos a emissão de uma nota fiscal como uma transação, e os produtos vendidos como os itens dentro da transação. Uma regra de associação é denotada $A \Rightarrow B$ (lê-se "A implica B"), onde A e B, denominados premissa (ou antecedente) e consequência, respectivamente, são conjuntos de itens frequentes (*frequent itemsets*, em inglês) em

uma base de dados transacional. Por exemplo, na base de dados de vendas de um supermercado, 'A' poderia ser {pão} e 'B' poderia ser {margarina}. Ou ainda, 'A' poderia ser {cereal, leite} e 'B' poderia ser {pão,margarina}. Neste caso, no contexto de um supermercado a regra $A \Rightarrow B$ deve ser entendida como "Comprar cereal e leite implica em comprar pão e margarina". Por simplicidade, neste projeto limitamos a premissa e a consequência a conjuntos de apenas um item cada.

A cada regra podem-se associar métricas de qualidade, que medem, segundo certos critérios, quão interessante é a regra. As métricas mais simples e mais comumente utilizadas são suporte (*Support*) e confiança (*Confidence*). No entanto essas duas métricas por si só não são suficientes para medir a qualidade de uma regra, e por isso devem ser utilizadas em conjunto com outras métricas: melhoria (*Lift*), e convicção (*Conviction*). [1]

6.2.1 Suporte (*Support*) [2]

Pode-se falar em suporte de um conjunto de itens e em suporte de uma regra. O suporte de um conjunto de itens é a frequência com a qual os itens do conjunto aparecem juntos na mesma transação dentro da base de dados. Recursivamente, o suporte de uma regra é o suporte do conjunto de itens da regra (na premissa e na consequência). Por exemplo, pão e margarina podem aparecer juntos em 80% das transações. Se assim for, as regras $\{pão\} \Rightarrow \{margarina\}$ e $\{margarina\} \Rightarrow \{pão\}$ têm suporte de 80% Matematicamente, suporte de uma regra é a razão entre o número de transações que incluem todos os itens da regra (da premissa e da consequência) pelo número total de transações existentes na base de dados. O suporte de uma regra pode ser expresso como segue:

$$suporte(A \Rightarrow B) = P(A \cap B)^3 \tag{6.1}$$

[1] Os nomes das métricas de qualidade já estão bem consolidados em inglês, mas existem diferentes traduções dos nomes das métricas para o portugués, o que pode levar a interpretações ambíguas. Para eliminar a ambiguidade, adotaremos nesta monografia os nomes em português aqui mencionados.

[2] O significado das métricas apresentadas nesta seção foi extraído e adaptado de `http://docs.oracle.com/cd/B28359_01/datamine.111/b28129/algo_apriori.htm` e `http://michael.hahsler.net/research/association_rules/measures.htmlem20/09/2012`

[3] Nesta monografia tratamos a ocorrência conjunta de dois ou mais conjuntos de itens como a intersecção entre eles. Assim, $\{pão\} \cap \{margarina\} = \{pão, margarina\}$. É importante notar que cada item é representado por uma variável de valores binários (presença e ausência). Se o conjunto {pão,margarina} denota a presença de pão e de margarina em uma transação, então ele representa todas as transações onde há a coocorrência de pão e margarina, sendo mais restritivo do que {pão} e {margarina} separadamente. Ou seja, as transações que satisfazem {pão,margarina} são a intersecção das transações que satisfazem {pão} com as que satisfazem

6.2.2 Confiança (*Confidence*)

A confiança é a probabilidade condicional de ocorrer o itemset consequente dado que ocorre o antecedente. Por exemplo, em uma base de dados com 100 transações, os cereais podem aparecer em 50 transações, e 40 destas 50 também podem incluir leite. A confiança da regra "cereais implica leite"seria de 80%, e o suporte desta regra seria de 40% .

Matematicamente, a confiança é a razão entre o suporte da regra e o suporte do antecedente. A confiança pode ser expressa em notação de probabilidade como se segue.

$$confiança(A \Rightarrow B) = P(B/A) = (A \cap B)/P(A) \tag{6.2}$$

6.2.3 Melhoria (*Lift*)

Tanto o suporte quanto a confiança devem ser utilizados para determinar se uma regra é válida. No entanto, há casos em que ambas as medidas podem ser altas, e ainda assim produzir uma regra que não é útil. Por exemplo: clientes de lojas de conveniência que compram suco de laranja também compram leite com uma confiança de 75%. A combinação de leite e suco de laranja tem um suporte de 30%.

Aparentemente esta é uma excelente regra, e na maioria dos casos, seria. Mas e se os clientes de lojas de conveniência em geral compram leite 90% das vezes? Nesse caso, os clientes de suco de laranja são na verdade menos propensos a comprar leite do que os clientes em geral.

Uma terceira medida é, portanto, necessária para avaliar a qualidade da regra. A melhoria indica a força de uma regra em relação à co-ocorrência aleatória do antecedente e do consequente. Ela fornece informações sobre a melhoria, o aumento da probabilidade de o consequente dado o antecedente. Melhoria é definida como segue:

$$melhoria(A \Rightarrow B) = P(A \cap B)/(P(A) * P(B)) \tag{6.3}$$

Portanto, a melhoria é uma métrica que evidencia quantas vezes a confiança real da regra é maior do que se era de se esperar caso a ocorrência dos itens

{margarina}.

envolvidos fosse independente, isto é, caso não houvesse nenhuma correlação entre a premissa e a consequência.

Assim, no nosso exemplo, assumindo que 40% dos clientes compram suco de laranja, a melhoria seria 30% / (40% * 90%) = 0,83. Uma melhoria menor do que 1.

Qualquer regra com uma melhoria menor que 1 não indica uma oportunidade real de *cross-selling*, não importa o quão alto sejam o suporte e a confiança, pois, na verdade, oferece menos capacidade de prever uma compra conjunta do que se elas fossem feitas ao acaso.

6.2.4 Convicção (*Conviction*)

A convicção é uma métrica desenvolvida como alternativa à confiança, que segundo (BRIN et al., 1997), não é capaz de capturar a direção das associações corretamente. Em uma regra $\{A\} \Rightarrow \{B\}$, a convicção compara a probabilidade de que A ocorra sem B se eles forem dependentes com a real frequência de ocorrência de A e não B.

$$convicção(A \Rightarrow B) = P(A) * P(\neg B)/P(A \cap \neg B) \tag{6.4}$$

É importante notar que, no sentido em que é empregado, $P(\neg B)$ significa a probabilidade de que o conjunto de itens B esteja ausente em uma transação, em contraposição ao significado de P(B), que é a probabilidade de que o conjunto de itens B esteja presente em uma transação.

Diferente da confiança, a convicção é normalizada baseada tanto na premissa quanto na consequência da regra, assim como a noção estatística de correlação. Ainda, diferente da melhoria, ela é direcional (assimétrica), e mede uma real implicação em oposição à simples coocorrência (BRIN et al., 1997). Outro ponto interessante é que a convicção não é tão fortemente afetada por valores baixos de suporte quanto a melhoria.

Esta propriedade assimétrica muda a forma de interpretá-la em relação à melhoria. Tomemos como exemplo uma regra $\{A\} \Rightarrow \{B\}$ com convicção de 3 e a regra homóloga $\{B\} \Rightarrow \{A\}$ com convicção 1,5. Isto significa que a compra do item A implica na compra do item B duas vezes mais frequentemente que a regra inversa. Ou seja, existe uma relação de precedência entre os itens da regra.

De acordo com as descrições aqui apresentadas, a convicção nos parece ser

a mais indicada para identificar oportunidades de *cross-selling*, onde a intenção é oferecer ao consumidor um produto fortemente relacionado com outro produto que ele comprou recentemente.

Um bom exemplo é o das vendas de eletrônicos em supermercados. Suponha que os itens A e B do exemplo anterior sejam câmera digital e acessórios para câmera digital, respectivamente. Embora câmeras digitais não sejam os itens mais frequentemente vendidos em um supermercado, um consumidor que compra uma determinada câmera digital está fortemente propenso a comprar acessórios para sua câmera recém-adquirida. No entanto, clientes que compram acessórios para câmera digital não necessariamente têm interesse em comprar uma câmera digital, provavelmente porque já possuem uma.

Em posse deste tipo de informação é possível oferecer, a um cliente que comprou uma câmera digital, descontos em acessórios de câmeras na próxima vez que ele vier ao supermercado, antecipando assim sua compra e até mesmo evitando que ele compre os acessórios no concorrente.

6.3 Algoritmos

6.3.1 O Princípio Apriori e o Algoritmo Apriori

Uma vez que há geralmente um grande número de itens individuais diferentes em uma base de dados transacional típica, e suas combinações podem formar um número muito grande de conjuntos de itens, desenvolver métodos escaláveis para mineração de conjuntos de itens frequentes em um banco de dados transacional de grande porte é um desafio considerável. No entanto, existe uma propriedade de fechamento interessante (*Downward Closure Property*), chamada Apriori, entre *k-itemsets* frequentes (*k-itemsets* são conjuntos contendo k itens). Segundo esta propriedade, um *k-itemset* é frequente apenas se todos os seus subconjuntos de itens são frequentes (HAN et al., 2007).

Isto implica que os conjuntos de itens frequentes podem ser minerados pela primeira varredura do banco de dados para encontrar os 1-itemsets frequentes. Em seguida, usa-se os 1-itemsets frequentes para gerar candidatos a 2-itemsets frequentes, e verifica-se no banco de dados para obter a os 2-itemsets frequentes. Este processo se repete até que não haja mais k-itemsets frequentes que possam ser gerados por algum k. Esta é a essência do algoritmo Apriori. (HAN et al., 2007)

6.3.2 FP-Growth: Mineração de conjuntos de itens frequentes sem geração de candidatos

Em muitos casos, o algoritmo Apriori reduz significativamente o tamanho dos conjuntos candidatos utilizando o princípio Apriori. No entanto, pode sofrer de dois custos não triviais: (1) geração de um grande número de conjuntos de candidatos, e (2) varrer a base de dados repetidamente e fazer o controle dos candidatos por correspondência de padrão. (HAN; PEI; YIN, 2000) desenvolveram um método chamado *FP-Growth* que minera o conjunto completo de conjuntos de itens frequentes sem geração de candidatos.

O *FP-Growth* trabalha seguindo o princípio de dividir para conquistar. O núcleo deste método é o uso de uma estrutura de dados especial chamada de árvore de padrões frequentes (FP-Árvore), que retém as informações de associação entre conjunto de itens.

Em palavras simples, esse algoritmo funciona da seguinte forma: primeiramente ele comprime o banco de dados de entrada através da criação de uma instância da FP-árvore para representar itens frequentes em ordem descendente de frequência. Após este primeiro passo, ele divide o banco de dados comprimido em um conjunto de subbases de dados condicionais, cada uma associada a um padrão frequente. Por fim, cada subbase de dados condicional é minerada separadamente, iniciando pelos padrões mais curtos (com menos itens). Usando essa estratégia, o *FP-Growth* reduz os custos de busca procurando por padrões curtos recursivamente, concatenando-os em seguida aos padrões longos frequentes. Ou seja, o algoritmo *FP-Growth* transforma o problema de encontrar longos padrões frequentes em uma busca recursiva por padrões mais curtos, e e em seguida concatena o sufixo.

Estudos demonstram que o desempenho do método reduz substancialmente o tempo de busca em relação ao Apriori (HAN et al., 2007).

6.3.3 Outros Algoritmos

Além dos algoritmos anteriormente descritos, existem ainda algoritmos como o Eclat, que usa um formato de dados vertical para realizar a mineração, e algoritmos mais sofisticados para mineração multidimensional, multinível e de padrões ordenados cronologicamente (HAN et al., 2007).

Muito embora estes tipos de algoritmos tenham grande potencial de aplicação em análises de carrinho de compras, infelizmente estes algoritmos não estão dis-

poníveis no acervo de algoritmos da versão atual do Weka, a ferramenta escolhida para este projeto (vide seção 6.4). No entanto, a utilização do algoritmo *FP-Growth* mostrou-se extremamente satisfatória na geração de regras de associação a partir de nossa base de dados (vide seção 6.6).

6.4 A Escolha da Ferramenta

Antes de partir para a implementação efetiva do sistema planejado, fizemos uma breve pesquisa de ferramentas de *data mining* disponíveis que nos auxiliassem durante o desenvolvimento. Os critérios utilizados, ferramentas pesquisadas e escolha são detalhados na sequência.

Algumas caracteristicas desejaveis na ferramenta são as seguintes:

- Ser gratuita e open source;
- Dispor de uma API bem documentada, facilmente reutilizável e integrável em outros projetos;
- Dispor de uma gama variada de algoritmos de classificação, clusterização e regras de associação implementados;
- O desempenho dos algoritmos dos quais dispõe não pode ser um empecilho para testar a metodologia;
- Desejável que permita a visualização gráfica dos resultados das análises.

6.4.1 Elki [4]

Elki é um projeto de pesquisa da Universidade Ludwig Maximilian de Munique com capacidades avançadas de clusterização e métodos de detecção de outlier escritos na linguagem Java. Um de seus diferenciais é a possibilidade de usar árvores de índices na execução dos algoritmos, o que acelera a execução dos mesmos, segundo o benchmarking divulgado em seu website. Outra particularidade é sua alta modularidade, que permite a combinação arbitrária de algoritmos, tipos de dados, índices e funções. No entanto, o projeto foi desenvolvido para fins de pesquisa e aprendizado, e por esta razão, seu código foi escrito focando a legibilidade, reusabilidade e extensibilidade, em vez de otimização para desempenho. Portanto o ganho de desempenho com o uso de árvores de índices só é comparável

[4] `http://elki.dbs.ifi.lmu.de/`

com outros algoritmos implementados no próprio projeto Elki. Além disso, o Elki oferece poucas opções de algoritmos de classificação e regras de associação.

6.4.2 Orange [5]

Orange é uma ferramenta de *data mining* de código aberto desenvolvida em Python. Seu diferencial é a possibilidade de programar as análises visualmente ou via script em Python. Sua interface permite visualizar os resultados graficamente, dispondo de diversos *widgets* para este fim. Suas bibliotecas podem ser importadas em qualquer script Python, sendo, portanto possível criar novas soluções a partir delas. Oferece uma boa gama de algoritmos para resolver diferentes tipos de problemas (classificação, clusterização, regras de associação, amostragem de dados, mineração de texto, regressões, etc), porém dispõe de poucas opções de algoritmos em cada categoria de problema (o que é desvantajoso para efeitos de comparação de resultados e de desempenho).

6.4.3 Weka [6]

O Weka é uma plataforma gratuita e de código aberto, desenvolvida na Universidade de Waikato, Nova Zelândia, em constante desenvolvimento por uma equipe que disponibiliza o código e suas bibliotecas em formato .jar (para desenvolvimento em Java), sendo possível utilizá-lo como um arcabouço para novas soluções. Esta ferramenta dispõe de uma variada gama de algoritmos de pré-processamento, classificação, regressão, clusterização e regras de associação, além de prover acesso a bancos de dados via JDBC.

O framework do Weka foi desenvolvido de forma que suas classes sejam facilmente reutilizáveis caso se deseje utilizar algum algoritmo específico que porventura ainda não esteja implementado no Weka. Dentre as soluções open source pesquisadas, é a que possui a comunidade mais ativa e a única cujos desenvolvedores escreveram um livro específico e é a mais citada em estudos de benchmarking de desempenho.

[5] `http://orange.biolab.si/`
[6] `http://www.cs.waikato.ac.nz/ml/weka/`

6.4.4 RapidMiner [7]

RapidMiner é uma solução open source com versões comerciais pagas e fechadas (modelo freemium) baseado no framework do Weka. Com um apelo fortemente empresarial, carrega consigo todas as funcionalidades do Weka, além de outras funcionalidades que visam facilitar e agilizar o trabalho do analista, como interface de usuário mais amigável, e visualização de gráficos avançada. Sua versão gratuita também permite a reutilização do seu código via API.

6.4.5 Pentaho Analytics [8]

Pertencente à Pentaho Corporation, que em 2006 adquiriu uma licença exclusiva para usar o Weka para sua solução de business intelligence. É portanto mais uma ferramenta open source com versões comerciais pagas baseada em Weka, que lhe confere funcionalidades extras.

6.4.6 Ferramenta Escolhida: O Framework Weka

O Weka foi escolhido por atender todas as características procuradas (embora não seja o melhor em todas elas) e por manter-se numa camada arquitetural baixa o suficiente para não contrariar os interesses acadêmicos deste projeto.

Dentre as ferramentas pesquisadas, as mais completas são RapidMiner e Pentaho (ambas se baseiam no framework Weka para análises de data mining). No entanto, por estarem em uma camada de software superior à do Weka (do ponto de vista de arquitetura), estas ferramentas visam tornar as análises de data mining transparentes ao usuário e por isso se distanciam muito da essência do conceito de data mining, o que foge do interesse acadêmico deste projeto. Num contexto empresarial, RapidMiner e Pentaho seriam preferíveis ao Weka pela sua agilidade em gerar resultados e facilidade de uso.

Além de RapidMiner e Pentaho, o Weka também é o arcabouço principal de outras ferramentas de data mining, como o KNIME. A recorrência do uso do Weka como arcabouço de outros projetos nos chamou atenção, pois lhe credencia como um framework respeitado e difundido na comunidade de data mining. A possibilidade de integrar o Weka dentro de nosso projeto, embora não seja exclusiva desta ferramenta, permite focar-nos no desenvolvimento da solução para supermercados sem termos que nos preocupar com o desenvolvimento dos algoritmos

[7] http://rapid-i.com/
[8] http://www.pentaho.com/

que já estão consolidados no meio científico.

Por último, e não menos importante, o Weka dispõe da maior variedade de algoritmos de regras de associação (análise de data mining mais extensivamente usada neste projeto), se comparado ao Orange e Elki

Por estas razões o Weka é um ótimo ponto de partida para um projeto como o que foi desenvolvido por nós.

6.5 Desenvolvimento

O desenvolvimento foi feito em linguagem Java utilizando o framework WEKA 3.6 (última versão estável). O código desenvolvido permite a realização de consultas diretamente ao banco de dados (MySQL 5.5), extraindo-os e manipulando-os para que fiquem organizados na estrutura requerida pela interface de entrada do framework do Weka.

Importante notar que o Weka especifica um formato de arquivo próprio, cuja extensão é .arff (*Attribute Relationship File Format*). Este formato de arquivo de texto permite descrever claramente a estrutura de uma relação de banco de dados utilizada para uma determinada análise, suas instâncias (observações) e seus atributos (características). Mais detalhes sobre a estrutura deste formato de arquivo podem ser consultados no Wiki do Weka, na página `http://weka.wikispaces.com/ARFF+%28stable+version%29`. Entender a estrutura deste formato de arquivo foi essencial para realizarmos os primeiros testes de execução dos algoritmos diretamente a partir da interface de usuário do Weka, e a partir disso nos familiarizarmos com seu funcionamento. Num segundo momento, o entendimento do formato do arquivo nos ajudou a entender rapidamente a API de entrada e saída do framework.

O Bom Cupom não precisa gerar e salvar um arquivo .arff em disco para que os dados possam ser informados aos algoritmos do Weka. Graças a métodos disponíveis no próprio framework, nossa implementação carrega os dados do banco de dados e os organiza na estrutura do formato .arff diretamente na memória do programa, o que torna o processo de extração e organização dos dados mais eficiente do que se utilizássemos o disco rígido como intermediário.

A estrutura dos dados de entrada para os algoritmos geradores de regras de associação é uma coleção de instâncias, todas possuindo os mesmos atributos, porém obviamente os valores dos atributos são específicos a cada instância. Em função da abstração que se escolhe para as instâncias ou atributos, é possível obter

regras diferentes, com interpretações diferentes. Por exemplo, se os dados são organizados e agrupados de forma que cada instância seja uma venda específica (um cupom fiscal), os atributos sejam os superavatares, e os valores dos atributos de cada instância indiquem presença ou ausência de cada superavatar naquela instância, então a interpretação de uma regra $A \Rightarrow B$ será "A compra de produtos do superavatar 'A' implica na compra de produtos do superavatar 'B' no mesmo cupom fiscal". Se reagruparmos os dados de forma que cada instância seja um cliente específico e os valores dos atributos indiquem se este cliente já comprou ou não determinado superavatar (considerando todas as compras que ele já fez), a interpretação da regra passa a ser "clientes que já compraram algum produto do superavatar A também já compraram algum produto do superavatar B (não necessariamente no mesmo cupom fiscal)". Portanto fica claro que a escolha das instâncias influencia na interpretação das regras geradas.

Poderíamos também escolher abstrair os atributos no nível dos avatares, ao invés dos superavatares. Como avatares são subclasses dos superavatares (com grau de granularidade maior), a consequência imediata é que cada instância passará a ter muito mais atributos do que tinha antes. Com isso as regras passarão a ser muito mais específicas, revelando relações entre avatares menos evidentes de se imaginar pelo senso comum e, portanto, mais interessantes. Por outro lado, se o grau de granularidade dos atributos for muito alto, as regras serão tão específicas, com um suporte tão baixo, que terão pouca ou nenhuma relevância estatística (lembrando ainda que valores de suporte muito baixos costumam distorcer as métricas confiança e melhoria).

O algoritmo escolhido para gerar as regras de associação foi o *FP-Growth*, devido à sua melhor performance em relação a outros algoritmos. (HAN et al., 2007), em sua revisão sobre os algoritmos de regras de associação existentes, explica que o *FP-Growth* tem desempenho melhor que o algoritmo Apriori, baseando-se em estudos desenvolvidos.

Além de escolher qual o nível de granularidade das instâncias e dos atributos dos dados de entrada, os algoritmos precisam ser parametrizados. A implementação do *FP-Growth* no Weka 3.6 solicita 12 parâmetros, dos quais 4 são condicionalmente obrigatórios (são usados pelo algoritmo apenas em função de outros parâmetros). Os principais parâmetros são:

lowerBoundMinSupport: Limite inferior para o suporte mínimo que as regras geradas devem ter;

upperBoundMinSupport: Limite superior para o suporte mínimo que as re-

gras devem ter. O algoritmo procurar regras diminuindo o suporte mínimo iterativamente a partir deste valor;

numRulesToFind: O número máximo de regras para exibir na saída;

delta: O algoritmo vai iterativamente diminuir o suporte das regras procuradas por este fator até que o limite inferior do suporte (lowerBoundMinSupport) seja alcançado ou o número necessário de regras (numRulesToFind) tenha sido gerado;

metricType: métrica usada para ordenar as regras na saída (do maior valor para o menor). As opções são confiança, melhoria, convicção e alavancagem;

minMetric: determina um valor para a métrica definida em metricType abaixo do qual as as regras geradas devem ser ignoradas;

maxNumberOfItems: determina o máximo número de itens que devem ser incluídos nos conjuntos de itens frequentes. Útil para impor que as regras geradas tenham apenas um item na premissa e um item na consequência (para isto, definir este parâmetro com o valor '2');

findAllRulesForSupportLevel: verdadeiro ou falso. Determina se o algoritmo deve ou não encontrar todas as regras que atendam as restrições do lowerBoundMinSupport e do minMetric. Ativar esse modo desativa o processo iterativo de redução do suporte para encontrar o número especificado de regras.

Na versão atual do sistema, é possível escolher o nível de abstração das instâncias e atributos em tempo de execução, assim como a parametrização do algoritmo. Como saída, as regras geradas são introduzidas em uma tabela do banco de dados, para que possam ser consultadas pelos demais módulos quando forem executadas as rotinas de geração do cupom de desconto. As regras geradas pelo algoritmo também são salvas em um arquivo out.csv, para que possam ser facilmente consultadas e lidas por pessoas.

6.6 Resultados

Durante nossos testes, o Gerador de Regras de Associação foi executado diversas vezes, testando diferentes parametrizações. Observou-se que a parametrização influencia consideravelmente quais regras são geradas e na performance do algoritmo. Por exemplo, quanto menor for o parametro *delta*, mais regras são geradas.

com um tempo de execução maior. Tem-se o mesmo efeito quanto maior for a diferença entre os parâmetros *upperBoundMinSupport* e *lowerBoundMinSupport* e quanto maior for o parâmetro *numRulesToFind*. A ativação do parâmetro *findAllRulesForSupportLevel* permite controlar indiretamente o número de regras geradas ajustando-se apenas os parâmetros *upperBoundMinSupport* e *lowerBoundMinSupport*. Adotamos essa abordagem para gerar as regras de associação.

Se a parametrização for tal que o número de regras geradas for muito pequeno, pode-se perder regras interessantes. Contrariamente, se o número de regras geradas for muito grande, pode-se gerar muitas regras de pouco interesse, que não representam uma real correlação entre dois tipos de produto (por exemplo, com confiança muito baixa, ou melhoria menor que 1), o que dificulta a identificação das regras interessante dentro do todo.

Como os requisitos do Bom Cupom (seção 3.4) não exigem que a atualização das regras de associação entre avatares seja feita constantemente (apenas semanalmente), admitimos um tempo de processamento para geração das regras da ordem de algumas horas (de fato com os dados dos quais dispomos o tempo para geração das regras é da ordem de 10 minutos). Por isso, nos preocupamos apenas que o sistema não deixe de gerar as regras mais interessantes, mesmo que isso implique na geração de muitas regras de pouco interesse. A identificação das regras de maior interesse pode ser feita ordenando-as e filtrando-as segundo o critério desejado.

A tabela 6.1 mostra os resultados obtidos da aplicação do algoritmo *FP-Growth*, utilizando a classificação em avatares que foi feita manualmente, com os seguintes parâmetros:

lowerBoundMinSupport = 0.001
upperBoundMinSupport =1.0
numRulesToFind = 10
delta: 0.05
metricType: confiança
minMetric: 0.001
maxNumberOfItems: 2
findAllRulesForSupportLevel: verdadeiro

Notar que os valores de *delta* e *numRulesToFind* são indiferentes para o resultado, visto que *findAllRulesForSupportLevel* é verdadeiro.

Premissa	Consequencia	confidence	lift	conviction	support of rule (%)
[POLENTA=1]	[PAO=1]	0,82	2,13	2,26	0,1%
[ESPECIARIA=1]	[LEGUME=1]	0,79	3,28	3,54	3,6%
[PATE=1]	[PAO=1]	0,79	2,06	2,83	1,2%
[FRIOS=1]	[PAO=1]	0,78	2,04	2,84	9,0%
[GRAOS=1]	[FRUTA=1]	0,78	2,21	2,81	1,0%
[DROGARIA=1]	[FRUTA=1]	0,78	2,23	2,67	0,3%
[EMBALAGEM=1]	[PAO=1]	0,78	2,03	2,22	0,2%
[REQUEIJAO=1]	[PAO=1]	0,77	2	2,64	4,4%
[CARNE PEIXE=1]	[FRUTA=1]	0,77	2,21	2,65	0,5%
[ELETRODOMESTICO=1]	[FRUTA=1]	0,76	2,17	2,32	0,2%
[COMIDA BEBE=1]	[PAO=1]	0,76	1,98	2,16	0,2%
[SABAO=1]	[PRODUTO LIMPEZA=1]	0,75	7,77	3,5	1,3%
[COMIDA JAPONESA=1]	[FRUTA=1]	0,75	2,14	2,34	0,3%
[COMIDA JAPONESA=1]	[LEGUME=1]	0,75	3,13	2,74	0,3%
[UTENSILIO BANHEIRO=1]	[HIGIENE PESSOAL=1]	0,75	4,76	2,89	0,2%
[SOBREMESA CONGELADA=1]	[PAO=1]	0,75	1,95	1,97	0,1%
[PAO DE QUEIJO=1]	[FRUTA=1]	0,74	2,1	2,24	0,3%
[VERDURA=1]	[FRUTA=1]	0,73	2,08	2,4	11,5%
[FEIJAO=1]	[LEGUME=1]	0,73	3,04	2,78	2,4%
[QUEIJO=1]	[PAO=1]	0,72	1,88	2,22	14,1%
[PANOS=1]	[PRODUTO LIMPEZA=1]	0,72	7,51	3,09	0,5%
[AZEITONA=1]	[LEGUME=1]	0,72	3,02	2,55	0,4%
[EMBALAGEM=1]	[CAFÉ DA MANHA=1]	0,72	5	2,57	0,1%
[VERDURA=1]	[LEGUME=1]	0,71	2,95	2,58	11,1%
[FRIOS=1]	[QUEIJO=1]	0,71	3,64	2,78	8,2%
[OVO=1]	[FRUTA=1]	0,71	2,03	2,25	5,7%
[GELATINA=1]	[FRUTA=1]	0,71	2,02	2,18	1,1%
[FAROFA=1]	[LEGUME=1]	0,71	2,95	2,48	0,6%
[HAMBURGER=1]	[PAO=1]	0,71	1,85	2,03	0,5%
[FARINHA LACTEA=1]	[LEITE=1]	0,71	4,41	2,38	0,1%
[ARROZ=1]	[LEGUME=1]	0,7	2,9	2,48	2,9%
[AZEITONA=1]	[FRUTA=1]	0,7	2	2,03	0,4%
[TEMPERO=1]	[LEGUME=1]	0,69	2,86	2,42	7,7%
[OLEO COZINHA=1]	[LEGUME=1]	0,69	2,86	2,4	2,8%
[GRAOS=1]	[LEGUME=1]	0,69	2,89	2,42	0,9%
[CURATIVO=1]	[QUEIJO=1]	0,69	3,55	2,09	0,1%
[CURATIVO=1]	[HIGIENE PESSOAL=1]	0,69	4,39	2,19	0,1%
[AZEITE=1]	[FRUTA=1]	0,68	1,93	1,99	3,4%
[AGUA COCO=1]	[FRUTA=1]	0,68	1,95	2,02	1,1%
[GELATINA=1]	[PAO=1]	0,68	1,77	1,88	1,0%
[ALCOOL=1]	[PRODUTO LIMPEZA=1]	0,68	7,11	2,79	0,8%
[CARNE PEIXE=1]	[LEGUME=1]	0,68	2,83	2,24	0,4%
[PAO DE QUEIJO=1]	[PAO=1]	0,68	1,78	1,8	0,3%
[DROGARIA=1]	[HIGIENE PESSOAL=1]	0,68	4,29	2,4	0,3%
[ELETRODOMESTICO=1]	[PAO=1]	0,68	1,77	1,71	0,2%
[LEGUME=1]	[FRUTA=1]	0,67	1,9	1,94	16,0%
[ESPECIARIA=1]	[FRUTA=1]	0,67	1,91	1,96	3,0%
[SAL=1]	[LEGUME=1]	0,67	2,78	2,23	1,0%
[VINAGRE=1]	[FRUTA=1]	0,67	1,9	1,9	0,8%
[FERMENTO=1]	[FRUTA=1]	0,67	1,92	1,93	0,7%
[EMBALAGEM=1]	[FRUTA=1]	0,67	1,9	1,67	0,1%
[OVO=1]	[LEGUME=1]	0,66	2,77	2,26	5,3%
[AZEITE=1]	[LEGUME=1]	0,66	2,74	2,2	3,3%
[UTENSILIO LIMPEZA=1]	[PRODUTO LIMPEZA=1]	0,66	6,81	2,6	2,6%

Tabela 6.1: Regras obtidas após a execução do algortimo *FP-Growth*

6.7 Homologação das Regras

6.7.1 Generalidade

Uma das questões que observamos durante o planejamento do modelo de regras de associação é a necessidade de avaliar o quanto as regras geradas podem ser generalizadas para qualquer situação. Em outras palavras, sentimos a necessidade

de medir a capacidade de que o conjunto de dados utilizado possa gerar regras que façam sentido no contexto de outros supermercados (outros conjuntos de dados), e não somente no supermercado de onde os dados foram extraídos.

Numa primeira análise, é fácil observar que a capacidade de generalização das regras obtidas com um conjunto de dados depende muito da quantidade de observações que o conjunto possui. A esse fator, adiciona-se o perfil do consumidor que fez as compras presentes no conjunto de dados. Este perfil pode variar de um supermercado pra outro, em função de seu porte, sua localização, promoções pontuais, orientação comercial (atacado ou varejo), etc. Estatisticamente falando, consumidores com características diferentes não fazem parte da mesma população.

Embora não disponhamos de dados de outros supermercados, pelos critérios acima mencionados, o conjunto de dados do qual dispomos já pode ser classificado como não generalizável. Isso porque contempla apenas as vendas realizadas em 26 dias durante um período de 13 meses em um supermercado de médio porte específico, isto é, com perfil específico. Portanto, é natural de se esperar que as regras obtidas com este conjunto de dados sejam diferentes de um conjunto de dados de outro supermercado, ou do mesmo supermercado em um período diferente. Desta forma, a menos de regras de senso comum, não podemos assumir que as associações obtidas com o nosso conjunto de dados modelem o comportamento dos consumidores de um supermercado qualquer, sendo necessário repetir a geração das regras para cada supermercado.

6.7.2 Plausibilidade

Mesmo que não sejam generalizáveis, as regras de associação geradas por nosso modelo devem ser plausíveis, ou seja, devem corresponder na sua maior parte ao senso comum. Uma forma empírica de homologar a validade do modelo é verificar se as regras geradas correspondem com as compras que fazemos no nosso cotidiano. O senso comum normalmente se manifesta nas regras que possuem maior suporte (as mais recorrentes no dia-a-dia).

Seguindo o senso comum, é de se esperar, por exemplo, que a regra $Leite \Rightarrow Pão$ esteja entre as mais recorrentes e de maior confiança. O mesmo vale para regras como $Frios \Rightarrow Pão$, $Café\,da\,Manhã \Rightarrow Fruta$, $Verdura \Rightarrow Legume$, etc.

Ordenando as regras obtidas com a parametrização apresentada na seção 6.6 em ordem decrescente de suporte, observamos que de fato estas regras, e outras naturalmente esperadas, estão no topo da lista e com níveis de confiança elevados.

o que esta demonstrado na tabela 6.2:

Premissa	Consequencia	confidence	lift	conviction	support of rule (%)
[LEGUME=1]	[FRUTA=1]	0,67	1,9	1,94	16,0%
[FRUTA=1]	[LEGUME=1]	0,45	1,9	1,39	16,0%
[QUEIJO=1]	[PAO=1]	0,72	1,88	2,22	14,1%
[PAO=1]	[QUEIJO=1]	0,37	1,88	1,27	14,1%
[VERDURA=1]	[FRUTA=1]	0,73	2,08	2,4	11,5%
[FRUTA=1]	[VERDURA=1]	0,33	2,08	1,25	11,5%
[VERDURA=1]	[LEGUME=1]	0,71	2,95	2,58	11,1%
[QUEIJO=1]	[FRUTA=1]	0,57	1,62	1,5	11,1%
[LEGUME=1]	[VERDURA=1]	0,46	2,95	1,57	11,1%
[FRUTA=1]	[QUEIJO=1]	0,32	1,62	1,18	11,1%
[LEITE=1]	[PAO=1]	0,63	1,64	1,66	10,1%
[PAO=1]	[LEITE=1]	0,26	1,64	1,14	10,1%
[CAFÉ DA MANHA=1]	[PAO=1]	0,65	1,7	1,78	9,4%
[PAO=1]	[CAFÉ DA MANHA=1]	0,25	1,7	1,13	9,4%
[QUEIJO=1]	[LEGUME=1]	0,47	1,96	1,44	9,2%
[LEGUME=1]	[QUEIJO=1]	0,38	1,96	1,3	9,2%
[LEITE=1]	[FRUTA=1]	0,56	1,61	1,49	9,0%
[FRUTA=1]	[LEITE=1]	0,26	1,61	1,13	9,0%
[FRIOS=1]	[PAO=1]	0,78	2,04	2,84	9,0%
[PAO=1]	[FRIOS=1]	0,23	2,04	1,16	9,0%
[HIGIENE PESSOAL=1]	[FRUTA=1]	0,54	1,53	1,4	8,5%
[FRUTA=1]	[HIGIENE PESSOAL=1]	0,24	1,53	1,11	8,5%
[CAFÉ DA MANHA=1]	[FRUTA=1]	0,57	1,63	1,52	8,3%
[FRUTA=1]	[CAFÉ DA MANHA=1]	0,24	1,63	1,12	8,3%

Tabela 6.2: Regras mais relevantes

Também observamos algumas regras que nos surpreendem, como $HigienePessoal \Rightarrow Fruta$. Num primeiro momento não parece ser tão óbvio que haja correlação entre comida e itens de higiene, mas pensando com mais cuidado, esta regra reflete o hábito de consumo de pessoas que se preocupam com sua saúde e bem estar. Esta é uma das grandes utilidades das regras de associação: trazem à tona realidades que ao nos serem apresentadas parecem óbvias, porém que dificilmente seriam pensadas por alguém sem o auxílio das mesmas. Portanto, de forma empírica, validamos a plausibilidade das regras de associação obtidas com o conjunto de dados que dispomos.

6.7.3 Consistência Interna

Outra questão que nos fizemos é se o conjunto de dados do qual dispomos representa o perfil de consumo de uma só população estatística.

Para ilustrar o problema, suponha que um supermercado tenha 100 clientes, e que num período de um ano nenhum deles deixa de comprar regularmente e nenhum novo cliente surge. Durante um ano, o comportamento de compras destes mesmos 100 clientes pode mudar sazonalmente (em épocas de datas festivas como páscoa e natal). Se o comportamento de compras dos clientes mudar num período de duas semanas, então os conjuntos de dados de compras obtidos

antes e depois dessa mudança de comportamento não representam a mesma população (estatisticamente falando), muito embora estejamos falando dos mesmos 100 clientes.

Para avaliar se o nosso conjunto de dados representa uma só população, precisamos avaliar se ele é internamente consistente, ou seja, se dois ou mais subconjuntos de dados avaliados separadamente levam à geração de regras de associação similares.

Um conjunto de dados pode ser inconsistente internamente por duas razões:

- O conjunto de dados é muito pequeno, a ponto de que os padrões identificados nas regras de associação não representem significativamente nenhuma população específica;
- O conjunto de dados abrange períodos de tempo em que houve mudança no comportamento dos clientes, o que faz com que as regras geradas em um subperíodo sejam muito diferentes das regras geradas em outro subperíodo.

Conforme descrito na seção 4.1, o banco de dados com o qual trabalhamos consiste das vendas realizadas em 26 dias diferentes num período de um ano. Por isto, é provável que mudanças sazonais de comportamento tenham afetado a consistência interna destes dados.

Para avaliar se de fato os dados dos quais dispomos são internamente consistentes, pode-se utilizar a técnica de validação cruzada adaptada para regras de associação.

No seu uso mais geral, a validação cruzada é uma técnica de homologação de análises estatísticas cujo propósito é avaliar como os resultados de uma dada análise estatística podem ser generalizados para um conjunto de dados independente. É usada em situações em que o propósito é prever resultados em situações nunca observadas até o presente momento, e deseja-se estimar quão preciso o modelo preditivo vai ser na prática. Uma particularidade da validação cruzada é que ela é feita quando não se dispõe de um conjunto de dados independente para testar o modelo. Desta forma, o modelo é validado contra ele mesmo, tomando-se um subconjunto de dados aleatórios como conjunto de treinamento e o subconjunto complementar como conjunto de validação. A cada iteração da validação cruzada, um subconjunto de dados diferente do anterior é tomado como conjunto de validação, e é feita uma medida da validade do conjunto de treinamento contra o conjunto de validação. Para todos os efeitos, o conjunto de validação é tido como a verdade absoluta naquela iteração.

No caso da validação de regras de associação, a cada iteração medir-se-ia o quanto os resultados das regras geradas com o conjunto de teste são correlacionados com as regras geradas a partir do conjunto de validação. Após várias iterações com conjuntos de validação diferentes, far-se-ia uma média das medidas de correlação de cada iteração. Se na média as regras geradas por cada subconjunto fossem pouco correlacionadas, concluir-se-ia que o conjunto de dados mais geral não é internamente consistente, ou seja, não representam uma única população.

Embora seja desejável fazer a validação cruzada das regras de associação tal como descrito acima, um obstáculo para isto é: como medir a correlação entre conjuntos de regras de associação? O framework do Weka dispõe de algoritmos ou métodos prontos para realizar a validação cruzada de algoritmos de classificação e de clusterização, mas não faz o mesmo para regras de associação. Uma razão para isto é que a correlação entre regras de associação não é um conceito bem definido. Portanto, a viabilidade de realizar-se a validação cruzada das regras de associação depende da possibilidade de definir-se uma medida de correlação entre regras de associação. Não pretendemos nos aprofundar nesta discussão, tampouco propor um método para esta medida de correlação. Para todos os efeitos, confiaremos na assunção feita anteriormente, que nos parece bastante razoável.

Para todos os efeitos, consideramos que o conjunto de dados dos quais dispomos neste trabalho não é internamente consistente. Num caso real de implementação em supermercado, isto deve ser contornado pela atualização das regras de associação a cada semana ou mês, utilizando apenas os dados referentes a este período para geração das regras.

7 Identificador de Cestas Básicas

7.1 Objetivo

No processo de geração de ofertas personalizadas uma etapa importante é a modelagem dos hábitos de consumo dos clientes. Este módulo tem como objetivo identificar e modelar os hábitos de consumo de cada cliente individualmente baseado somente em suas compras anteriores. O resultado deste módulo será então cruzado com as regras de associação geradas pelo Gerador de Regras de Associação a fim de implementar as políticas de desconto individualizadas.

O modelo a ser gerado deve identificar todos os produtos (representados pelos seus respectivos avatares) que já foram comprados pelo menos uma vez pelo cliente. Além disso algumas informações adicionais são necessárias, como o preço médio ou gama dos produtos comprados com mais frequência.

O primeiro passo é comparar o preço dos produtos dentro de um mesmo avatar, para identificar em qual categoria de preço cada produto se encontra: premium (gama alta), intermediário (gama média) e econômico (gama baixa). Este passo é feito com o modelo de valuation dos produtos.

O segundo passo é gerar a cesta básica de cada cliente, determinando os avatares mais frequentemente comprados por cada um, traduzindo-se em um índice de recompra.

Uma cesta básica será composta, portanto, de itens que são definidos por três atributos:

- Nome do avatar
- Índice de recompra
- Gama predominante

Com esses atributos, aliado às regras de associação geradas pelo módulo Gerador de Regras de Associação, será possível analisar e determinar em quais produtos o supermercado deve oferecer desconto de acordo com as políticas descritas na seção 3.7.

7.2 Modelo de Valuation

Para auxiliar a geração final dos cupons de desconto um modelo deve ser criado para representar a variação de preço dentro de um mesmo tipo de produto e diferentes marcas. O objetivo deste modelo é identificar, dentro de cada avatar, quais produtos se encaixam em cada uma das três categorias de gama. Essa classificação será denominada Modelo de Valuation dos Produtos, e deve produzir a classificação de cada produto em faixas de preço.

7.2.1 Preço Real

Os produtos não possuem um preço único, mas o preço varia de acordo com diversos fatores e pode ser diferente em cada venda. Chamamos de preço real o preço que será considerado para cada produto em todas as análises de preço.

Para determinar o preço real, foram considerados todos os preços pelos quais esse produto já foi vendido. Algumas técnicas propostas para, a partir desse conjunto de valores, determinar o preço real estão descritas a seguir:

Média: o preço real é definido como a média dos preços pelos quais ele já foi vendido. Com essa técnica consideramos que todos os valores pelos quais o produto já foi vendido são relevantes no cálculo do preço real, porém levamos em consideração que quanto mais vezes um preço foi utilizado, maior a sua influência da determinação do valor final.

Moda: o preço real é definido como a moda dos preços pelos quais ele já foi vendido. A moda é definida como o resultado mais comum encontrado num conjunto de amostras. Com essa análise consideramos que o preço real de venda do produto é sempre aquele pelo qual é vendido mais frequentemente e todos os outros valores são ignorados. Essa análise parte do pressuposto que um produto é vendido a maior parte do tempo pelo seu preço real, e esse somente deve ser considerado para fins de análise.

Apesar da média ser mais fácil de calcular, alguns erros podem ser introdu-

zidos com dados ruidosos e preços muito fora do padrão, o que pode acontecer em casos de queima de estoque ou ofertas. Portanto, para o cálculo do preço real neste projeto utilizaremos a moda, deixando o modelo mais robusto a ruídos e considerando somente o preço mais comum para cada produto.

7.2.2 Expressões Regulares

Expressões regulares são uma importante ferramenta na análise de linguagem natural. São definidas por padrões que correspondem a determinadas seções de um texto. Seu nome é derivado seu poder expressivo. pois expressões regulares (comumente abreviadas por regexp ou simplesmente regex) podem representar qualquer linguagem regular.

Expressões regulares são comumente usadas para buscar padrões previamente definidos em textos ou strings. Para este projeto, como cada produto vem representado por uma string descritiva, o uso de expressões regulares pode ser muito eficaz para certos tipos de análise.

Uma explicação mais detalhada sobre os operadores e como expressões regulares representam padrões de texto pode ser encontrada no website Regular Expressions Info.[1]

7.2.3 Preço Normal

Há ainda uma outra questão a ser considerada na determinação dos preços dos produtos. Existem muitos produtos que diferem somente pelo tamanho da embalagem (peso ou volume). Por exemplo, considerando uma marca específica de refrigerante, é comum que um mesmo supermercado venda garrafas de 600mL e também garrafas de 2L desse mesmo produto. E naturalmente esses produtos não devem ter o mesmo preço real.

Devido à desestruturação das informações no banco de dados, essa informação não está a disposição. Porém uma simplificação que pode ser feita novamente considera a análise das strings que representam os produtos. Aqueles que apresentam características importantes de peso e volume normalmente vem com essa descrição no seu nome. Podemos ver como exemplo os seguintes produtos tirados da nossa base de dados:

```
REF COCA COLA 350ML LAT
```

[1] urlhttp://www.regular-expressions.info

```
REF COCA COLA 2L NORMAL
REF COCA COLA 600ML
```

Podemos ver claramente que esses três produtos são essencialmente os mesmos e diferem somente pelo tamanho da embalagem. Portanto uma simples análise de strings com o auxílio de Expressões Regulares será utilizada para definir o tamanho do produto.

Para determinar então o tamanho de cada produto para fins de comparação de preços, busca-se no banco de dados por todas as strings que representam os produtos. Neste passo devem ser considerados também strings recuperadas de outros sistemas a partir dos métodos descritos na seção 5.4.

Na definição do tamanho da embalagem temos, porém, dois casos distintos: Quando o produto é determinado por um volume e quando ele é determinado por um peso. Com uma análise dos produtos em nossa base de dados, os seguintes padrões foram determinados:

Para peso: uma descrição que define o peso de um produto apresenta sempre um número, contendo eventualmente o ponto (ou vírgula) decimal, seguido pela letra G (indicador de grama) podendo vir precedida pela letra K (quilogramas). Portanto a seguinte expressão regular deve encontrar todos os produtos desta categoria:

```
(\d+[\.,])?\d+[Kk]?[Gg]
```

Para posterior cálculo do tamanho, devemos, porém, introduzir alguns grupos de captura na expressão, resultando no seguinte:

```
(?:(\d+)[\.,])?(\d+)([Kk])?[Gg]
```

Da mesma maneira, observamos que produtos para os quais o volume é relevante, é encontrado na descrição do produto um número eventualmente contendo o separados decimal seguido pela letra L (litros), podendo ser precedida da letra M (mililitros). Analogamente, a expressão regular para determinar esses produtos, fica, portanto (já incluindo os grupos de captura):

```
(?:(\d+)[\.,])?(\d+)([Mm])?[Ll]
```

7.2.3.1 Cálculo do Tamanho da Embalagem

Tendo identificado a descrição de volume e peso, a seguinte metodologia foi utilizada no cálculo do tamanho da embalagem:

- Define-se um parâmetro "tamanho" para cada produto, que por padrão deve ser inicializado com o valor 1.0.
- Faz-se a busca por descrições de peso e volume para todos os produtos.
- Caso uma descrição de peso tenha sido encontrada, o parâmetro tamanho indicará a quantidade deste produto em quilogramas. Portanto, o valor encontrado na string deve ser atribuído ao parâmetro, e caso estivermos tratando com grama, em vez de quilogramas, este valor deve ser dividido por mil.
- Caso uma descrição de volume tenha sido encontrada, o parâmetro indicará a quantidade deste produto em litros. Analogamente, se se trata de uma descrição em mililitros, o valor deve ser dividido por mil.

Desta maneira produtos que não tenham descrição nenhuma de tamanho, ficam com o valor 1.0 nesse parâmetro e os outros estão definidos comparativamente com outros produtos dentro da mesma categoria.

7.2.4 Identificação da Gama do Produto

Definido um parâmetro justo para a comparação dos preços e tendo definido também o preço real normalizado de cada produto, podemos agora classificá-los em categorias de preço.

A categoria de preço é um indicador indireto de quanto de retorno (margem) o produto oferece ao supermercado (uma vez que a margem financeira de venda de cada produto não está disponível no banco de dados). Para o projeto serão definidos três tipos de categoria: premium (gama alta), intermediária (gama média) e econômica (gama baixa).

Primeiramente um passo em comum a qualquer método de definição de margens de lucro é a obtenções de todos os preços dentro de uma mesma categoria, que neste projeto são definidas com os avatares. De acordo com a distribuição de preços dentro desta amostra, alguns métodos estatísticos podem ser usados para definir as margens.

Método dos Tercis: Utilizando a definição estatística de quantis (HYNDMAN; FAN, 1996), divide-se a amostra em três grupos com o mesmo número de experimentos. Com esse método garante-se que, dentro de um mesmo avatar, o número de produtos classificados em cada gama será igual ou no máximo uma unidade diferente.

Os tercis de uma amostra estatística são números que dividem a amostra em porções equivalente quando dividida entre maiores e menores que este número. Na prática é a divisão da amostra em três sub-amostras de mesmo tamanho compostas pelos menores valores, maiores valores e valores intermediários.

É importante notar que a divisão da amostra só é possível se existirem pelo menos três exemplos. No caso de amostras com menos de três elementos, com este método, todos eles foram classificados à gama intermediária.

Método da Variação: Podemos ainda levar em consideração os valores na divisão em três sub-amostras. Pelo método da variação, analisamos o maior variação possível dentro da amostra (valor máximo menos valor mínimo) e dividimos esse intervalo em três intervalos equivalentes, classificando cada preço de acordo com seu valor e em qual intervalo está compreendido.

Nota-se que essa divisão não implica numa igualdade das sub-partes, visto que grande parte das amostras pode estar concentrada em um lado específico do intervalo.

Neste projeto utilizamos o método dos tercis, que nos garante um número mínimo de produtos em cada gama para cada avatar, de maneira a facilitar a implementação da Política de Upgrade de Marca.

7.3 Hábitos Individuais de Consumo

Para modelar os hábitos indviduais de consumo, levamos em conta o Modelo de Valuation definido na seção anterior e uma tabela constando todas as compras realizadas pelos clientes. Deve ser gerada uma cesta básica para os clientes, que contém itens indicando a probabilidade de que um produto que já tenha sido comprado venha a ser comprado novamente numa visita futura.

Uma primeira etapa para a geração do modelo de cestas básicas é, para cada cliente, isolar todas as suas compras. Com esses dados, podemos então definir

algumas métricas que serão utilizadas, potencialmente combinadas para gerar um modelo probabilístico:

Presença

A presença indica a proporção das vezes em que um determinado avatar está presente nas notas fiscais do cliente. É definida como o número de compras que contém o determinado avatar dividido pelo total de compras do cliente.

Densidade

A densidade indica quanto este avatar é comprado pelo mesmo cliente, ou seja, dentre todos os produtos comprados quantos representam este determinado avatar. A densidade é definida como a soma da quantidade presente de um avatar em todas as compras do cliente dividido pelo número total de produtos já comprados por esse cliente.

As métricas definidas a seguir dependem de dados temporais relacionados às compras. O banco de dados deve fornecer a data em que cada compra foi realizada.

Frequência

A frequência indica quantas vezes determinado avatar foi comprado desde que ele apareceu pela primeira vez na compra do cliente. A frequência é definida como o número de notas em que o avatar está presente dividido pelo intervalo de tempo entre a data da primeira compra deste avatar e a data atual.

Intervalo Médio

O intervalo médio está relacionado ao tempo entre duas compras em que o avatar está presente. Tem como objetivo indicar quanto tempo o cliente demora em média para comprar novamente um avatar. O intervalo médio é definido como a média dos intervalos de tempo de compras consecutivas em que o avatar esteja presente. Importante observar que o intervalo médio só se define se o cliente já tiver comprado pelo menos duas vezes o mesmo avatar em compras diferentes.

Atraso

O atraso indica há quanto tempo o cliente comprou pela última vez determinado avatar. O atraso é definido pelo intervalo de tempo entre a data da última compra em que o avatar está presente e a data atual.

Tais parâmetros foram definidos arbitrariamente de acordo com uma noção do que deve ser relevante para a definição dos padrões de compra de um cliente. Eles foram combinados de maneira a permitir o cálculo que indique a probabilidade do produto ser comprado novamente. Esse cálculo está descrito em mais detalhes na próxima seção.

7.4 Índice de Recompra

O índice de recompra é um valor atribuído a cada um dos itens em uma cesta básica. Esse valor é diretamente proporcional à probabilidade de que o cliente venha a comprar o produto novamente.

Para definir o índice de recompra foram utilizados os parâmetros definidos na seção anterior em conjunto. O cálculo do índice de recompra foi definido de maneira empírica utilizando métodos que refletem o entendimento do padrão de consumo de um cliente.

7.4.1 Parâmetros Temporais

Como descrito na seção anterior alguns parâmetros definidos para os itens da cesta básica dependem de fatores temporais, ou seja, da data na qual as compras foram realizadas.

Devido a limitações impostas pelo banco de dados à nossa disposição (seção 4.1), a utilização desses parâmetros poderia introduzir anomalias e causar distorções inesperadas no resultado, especialmente pelo fato das compras não terem sido registradas num período contínuo.

Isso nos impede de calcular a frequência, o intervalo médio e o atraso dos produtos, e portanto esses parâmetros serão desconsiderados para o cálculo do índice de recompra. Apesar das limitações dos dados, é possível ainda calcular a presença e a densidade de maneira satisfatória. Portanto o índice de recompra foi calculado a partir desses dois parâmetros.

7.4.2 Cálculo do Índice de Recompra

Para o cálculo do Índice de Recompra, levamos em consideração os parâmetro presença e densidade. Para esses dois parâmetro, tal como definidos nessa seção, é intuitivo imaginar que quanto maior eles forem maior deve ser também o índice

de recompra. Podemos definir então que é diretamente proporcional a esses dois parâmetros.

Porém ainda é preciso definir como exatamente eles influenciam no valor final do índice de recompra. Uma primeira abordagem sugere que simplesmente multipliquemos os valores entre si para gerar o valor final. Porém, com uma análise mais minuciosa dos dados, observamos que, devido à escassez de datas diferentes para as compras. Existem muitos clientes que possuem uma, duas ou três compras somente. Nesse caso, observamos a existência de cestas básicas com uma variação muito baixa de valores de presença em seus itens.

Por exemplo, para um cliente que apresente somente duas notas fiscais registradas no banco de dados, todos os itens de sua cesta basica só podem ter valores de preseça iguais a 1,0 ou 0,5 (ou o item está presente nas duas notas ou somente em uma, respectivamente). Já a densidade apresenta valores mais variados e pode estar mais diretamente ligada ao valor final do índice de recompra.

Para contornar esse problema, devemos definir o quanto de informação esses parâmetros nos trazem e isso é definido seguindo a ideia de entropia (FAYYAD; IRANI, 1992).

A entropia mede a quantidade de informação presente em um conjunto considerando quão diferente são seus valores. No nosso caso, como temos com muita frequência somente os valores 0,5 e 1,0 para presença, esse conjunto teria uma entropia baixa, enquanto a densidade, cujos valores variam muito mais, teria uma entropia mais alta.

Para definir um peso para os parâmetro portanto, vamos utilizar uma medida análoga à entropia, porém não exatamente igual. Definimos então o peso do parâmetro como sendo:

$$peso(parâmetro) = \frac{\#valores\ diferentes(parâmetro)}{\#total\ de\ valores(parâmetro)} \tag{7.1}$$

Definimos assim o índice de recompra segundo a seguinte equação:

$$IR = 10 * (presença * peso(presença) + densidade * peso(densidade))^2 \tag{7.2}$$

[2]utilizamos o fator multiplicativo 10 para obter valores um pouco maiores e facilitar a análise uma vez que todos os índices na equação são menores que 1 e portanto o resultado provavalmente não passaria de 1.

7.5 Resultados

Os resultados do Identificador de Cestas Básicas podem ser analisados e validados a partir de uma observação das notas de um cliente e da cesta básica gerada. Tomamos um cliente como exemplo do nosso banco de dados. Para esse cliente, observamos o registro de 4 notas fiscais, apresentadas na figura 7.1:

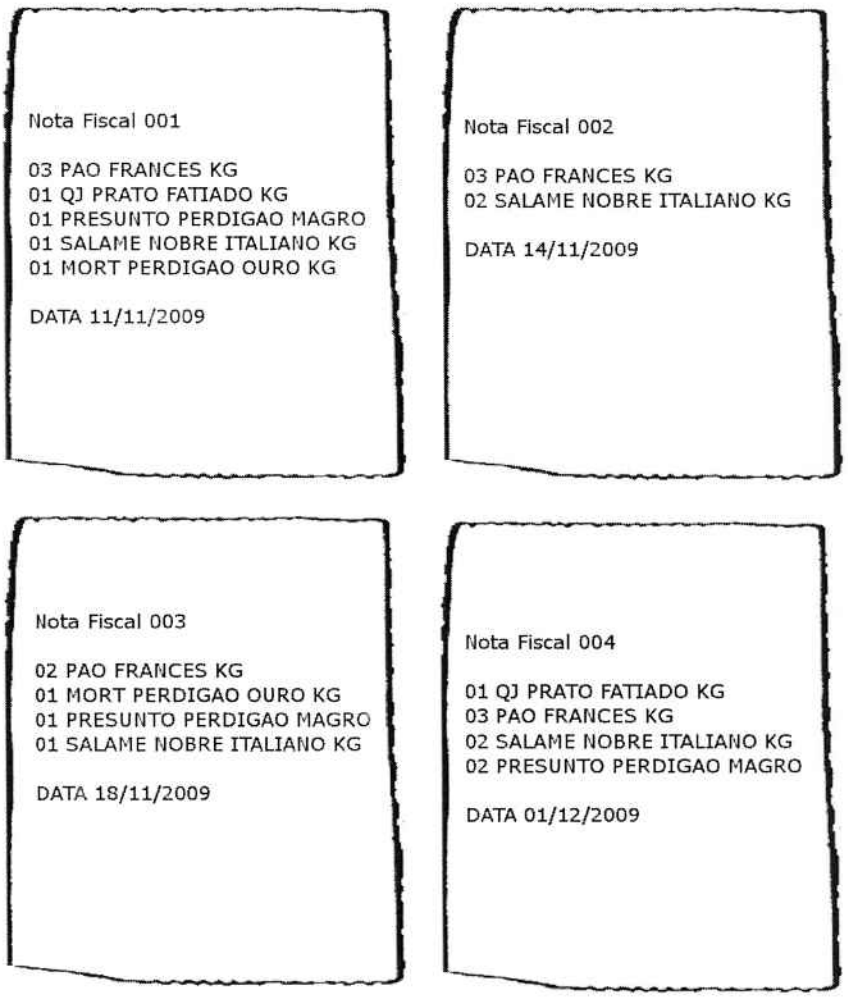

Figura 7.1: Notas fiscais para um cliente da base de dados

Pela classificação de avatares feita manualmente, que foi usada nos testes do identificador de cestas básicas, podemos ver que esse cliente comprou somente três avatares diferentes. A cesta básica gerada pelo Bom Cupom foi:

Avatar	Gama Predominante	Índice de Recompra
FRIOS	ECONÔMICA	11,467
PAO	INTERMEDIARIA	11,067
QUEIJO	ECONÔMICA	4,133

Tabela 7.1: Cesta básica gerada pelo Bom Cupom

Importante notar que o índice de recompra somente é proporcional à pro-

babilidade do produto voltar a ser comprado e não corresponde exatamente à probabilidade. Podemos observar esse fato simplesmente pelos valores serem todos maiores que 1.

A cesta básica apresentada na tabela 7.1, gerada para o cliente representado pelas notas fiscais da figura 7.1 está de acordo com o esperado e podemos também perceber que os índices de recompra refletem de fato o padrão de compras do cliente, dizendo que a probabilidade maior é que ele compre FRIOS novamente, seguido bem próximo da probabilidade de que ele compre PAO (a probabilidade de FRIOS é maior devido à maior quantidade de frios no total). Por último vemos que a probabilidade de que ele compre QUEIJO é um pouco menor que a metade das outras duas. Isso se deve ao fato de QUEIJO não estar presente em todas as notas fiscais.

8 Gerador de Cupons de Desconto

8.1 Objetivo

O objetivo deste módulo é integrar os modelos gerados pelo Gerador de Regras de Associação e pelo Identificador de Cestas Básicas para decidir quais produtos devem ser oferecidos com desconto no cupom promocional. Essa decisão deve ainda levar em conta a política de descontos escolhida pelo supermercado. As políticas de desconto foram definidas na especificação do Bom Cupom, na seção 3.7.

Uma função importante deste modulo também é realizar a tradução de avatares para produtos. Todo o processamento com os modelos propostos foi feito baseado na classificação em avatares, mas o desconto final vai ser oferecido em um produto. Portanto, uma vez decidido o avatar no qual o desconto deve ser oferecido, faz-se necessária uma correspondência com o produto final, e isso é realizado por este módulo.

Para que a seleção de um produto segundo qualquer uma das políticas não seja sempre o mesmo, é necessário adicionar um fator de indeterminismo nas funções de cálculo dos produtos. Para que isso fosse possível, sempre que é feita a seleção de um produto em alguma das políticas é definido um fator para cada produto denominado de *viés*, que é proporcional à probabilidade de que o produto seja selecionado nessa determinada etapa.

As próximas seções descrevem em datalhes a implementação dos algoritmos de seleção de produtos segundo cada uma das políticas e quando necessário é definido também exatamente como foi utilizado o viés.

8.2 Implementação da Política de Fidelização

Pela Política de Fidelização, será oferecido desconto em um produto que o cliente já costuma comprar frequentemente. Dado um cliente identificado pelo seu ID

do banco de dados o Bom Cupom segue os seguintes passos para selecionar o produto:

1. Com a identificação do cliente, o sistema recupera a cesta básica correspondente;
2. Pela cesta básica, o sistema identifica os cinco avatares com os maiores índices de recompra[1];
3. Dentre esses cinco avatares, é selecionado um probabilisticamente com o viés igual ao índice de recompra (isso significa que quanto maior o índice de recompra de um desses avatares, maior e a probabilidade de ele ser selecionado);
4. Com o avatar selecionado, o sistema busca qual é o produto desse avatar mais comprado pelo cliente;
5. O desconto é oferecido nesse produto.

Por exemplo, digamos que o avatar CAFÉ tenha sido selecionado no passo 3 para um determinado cliente. Existem vários produtos que são classificados como CAFÉ, e precisamos traduzir o avatar em um produto para oferecer o desconto. Considerando que esse cliente tenha comprado muito do produto CAFE MELITTA 250G FORTE, esse será o produto selecionado para oferecer desconto.

O algortimo da Política de Fidelização pode ser representada pelo seguinte pseudocódigo:

```
POLITICA_DE_FIDELIZACAO(clienteID):
    cesta = cestaBasica(clienteID);
    avataresCandidatos = cesta.ordenar(indiceRecompra)[0:5];
    avatar = avataresCandidatos.selecionarComVies(indiceRecompra);

    produtosCandidatos = clienteID.produtosJaComprados();
    produtosCandidatos.filtrarPorAvatar(avatar);

    RETURN produtosCandidatos.selecionarMaximo(NumeroDeVezesComprado);
END
```

[1]Aqui o valor cinco foi definido arbitrariamente. É uma constante que pode ser ajustada de acordo com a performance do sistema. Esse valor é o mesmo utilizado para selecionar os avatares para as três políticas de desconto.

8.3 Implementação da Política de Upgrade de Marca

Para a Política de Upgrade de Marca será oferecido desconto em um avatar que o cliente já costuma comprar, porém em um produto de uma gama superior. Novamente com o ID de um cliente, o Bom Cupom segue os seguintes passos para definir o produto selecionado seguindo essa política:

1. O sistema recupera a cesta básica do cliente com o ID;
2. Da cesta básica são eliminados todos os itens cuja gama predominantes seja premium (pois não é possível realizar upgrade de marca caso o cliente já costume comprar a marca de maior gama), deixando somente produtos com gama predominante econômica e intermediária;
3. O sistema identifica os cinco avatares com os maiores índices de recompra, ja desconsiderando aqueles com a margem premium;
4. Dentre esses cinco avatres é selecionado um com o mesmo método da Política de Fidelização. A seleção é feita probabilisticamente, levando em conta o viés como o índice de recompra dos avatares;
5. É identificada a margem predominante do avatar selecionado, e definida a margem do produto final como sendo uma acima dessa;
6. O sistema busca a última nota fiscal do cliente que contenha um produto do avatar selecionado;
7. Se o produto em questão for da gama selecionada, o desconto é oferecido neste produto;
8. Caso não haja nenhum produto com a gama selecionada na nota fiscal, o sistema seleciona um produto do avatar e gama selecionados aleatoriamente, excluindo aqueles que o cliente já tenha eventualmente comprado.

Com esses passos pretendemos entender as intenções do cliente da seguinte forma:

Uma vez selecionada a gama e o avatar do produto em que será oferecido desconto, precisamos definir se o cliente já comprou algum produto que atenda a esses critérios. [2]

[2] A gama selecionada está sempre um nível acima da gama predominante desse avatar na cesta básica do cliente, porém ainda assim é possível que o cliente tenha comprado produtos

Caso o cliente já tenha comprado e o produto está presente na última compra de um produto desse avatar, consideramos que o cliente já está experimentando um produto de uma marca superior. Porém como a gama predominante ainda está um nível abaixo, esse produto é selecionado para incentivar a fidelização a este produto com a gama mais alta.

Caso o produto que o atenda a esses critérios já tenha sido comprado, mas não na última compra que contém um produto desse avatar, consideramos que o cliente já experimentou esse produto no passado, mas não ficou satisfeito e voltou ao produto da gama inferior. Portanto, não é oferecido o desconto nesse determinado produto, mas sim em um outro que também atenda os critérios.

A algoritmo da Política de Upgrade de Marca pode ser representado pelo seguinte pseudocódigo:

```
POLITICA_DE_UPGRADE_DE_MARCA(clienteID):
    cesta = cestaBasica(clienteID);
    cesta.filtrarPorGama([Economica, Intermediaria]);
    avataresCandidatos = cesta.ordenar(IndiceRecompra)[0:5];
    avatar = avataresCandidatos.selecionarComVies(indiceRecompra);

    gamaPredominante = avatarSelecionado.getGamaPredominante();
    gama = gamaPredominante.proximoNivel();

    notasFiscais = clienteID.notasFiscais.contendo(avatar);
    ultimaNotaFiscal = notasFiscais.selecionarMaximo(data);
    IF (ultimaNotaFiscal.contem(avatar, gama)):
        RETURN ultimaNotaFiscal.getProduto(avatar, gama);

    ELSE:
        produtosExcluidos = clienteID.produtosJaComprados();
        produtosExcluidos.filtrarPorAvatar&Gama(avatar, gama);

        produtosCandidatos = produtos.getTodos(avatar, gama);
        produtosCandidatos.deletar(produtosExcluidos);

        RETURN produtosCandidatos.selecionarAleatoriamente();
END
```

que atendam esse critério, pois a gama predominante só indica aquela comprada na maior parte das vezes.

8.4 Implementação da Política de Introdução de Novo Produto

Para a Política de Introdução de Novo Produto será oferecido desconto em um avatar que o cliente ainda não costuma comprar. Como esta política envolve o modelo de cestas básicas e o de regras de associação, o algoritmo seguido é um pouco mais complexo:

1. Um avatar é selecionado utilizando o mesmo procedimento da Política de Fidelização (Passos de 1 a 3):
2. São recuperados os avatares que estejam na consequência de alguma Regra de Associação que tenha o avatar selecionado como premissa;
3. São eliminados os avatares que ja estão presentes na cesta básica do cliente;
4. Caso o conjunto de avatares não tenha ficado vazio, o viés é selecionado como sendo a convicção da Regra de Associação[3];
5. O produto final é selecionado aleatoriamente dentro do conjunto de produtos do avatar selecionado com uma gama igual à do item da cesta básica do cliente;
6. Se o conjunto de avatares ficou vazio após a eliminação daqueles que já ocorrem na cesta básica, todos os avatares são colocados de volta no conjunto e é selecionado um com viés igual à convicção da Regra de Associação;
7. Recuperar todos os produtos que sejam do avatar selecionado e da gama igual ao item da cesta básica primeiramente selecionado;
8. Eliminar desse conjunto os produtos ja comprados pelo cliente;
9. O produto final e selecionado aleatoriamente dentro do conjunto de produtos restantes;
10. Caso não haja mais nenhum produto no conjunto, colocar todos os produtos de volta e selecionar aquele que foi comprado o menos numero de vezes.

[3]A convicção foi selecionada como métrica mais adequada para definir a seleção do avatar pelos motivos apresentados na seção 6.2.4

Com essa metodologia pretende-se oferecer desconto em um avatar que o cliente nunca tenha comprado, levando em consideração as Regras de Associação para gerar sugestões relevantes ao cliente. Caso ele já tenha comprado todos os avatares sugeridos pelas Regras de Associação, é selecionado um produto de um desses avatares que ele ainda nao tenha comprado. Caso não haja nenhum produto que atenda a esses critérios, é selecionado o produto que ele comprou o menor número de vezes.

O algoritmo da Política de Upgrade de Marca pode ser representado pelo seguinte pseudocódigo:

```
POLITICA_DE_INTRODUCAO_DE_NOVO_PRODUTO(clienteID):
    cesta = cestaBasica(clienteID);
    avataresPremissa = cesta.ordenar(IndiceRecompra)[0:5];
    avataresPremissa.setVies(IndiceRecompra);
    avatarPremissa = avataresPremissa.selecionar();

    gama = avatarPremissa.getGamaPredominante();

    regras = RegrasDeAssociacao.filtrarPorPremissa(avatarPremissa);
    avataresCandidatos = regras.getAvataresConsequencia();

    avataresJaComprados = clienteID.avataresJaComprados();
    avataresCandidatos.deletar(avataresJaComprados);

    IF (avataresCandidatos is not vazio):
        avatar = avataresCandidatos.selecionarComVies(conviccao);
        produtosCandidatos = produtos.getTodos(avatar, gama);

        RETURN produtosCandidatos.selecionarAleatoriamente();

    ELSE:
        avataresCandidatos = regras.getAvataresConsequencia();
        avatar = avataresCandidatos.selecionarComVies(conviccao);

        produtosCandidatos = produtos.getTodos(avatar, gama);
        produtosJaComprados = clienteID.produtosJaComprados();
        produtosCandidatos.deletar(produtosJaComprados);
```

```
        IF (produtosCandidatos is not vazio):
            RETURN produtosCandidatos.selecionarAleatoriamente();

        ELSE:
            produtosCandidatos = produtos.getTodos(avatar, gama);
            RETURN produtosCandidatos.selecionarMinimo(compras);
END
```

8.5 Resultados

Para analisar os resultados consideramos a cesta básica representada na tabela 7.1, replicada na tabela 8.1 para facilitar a análise.

Avatar	Gama Predominante	Índice de Recompra
FRIOS	ECONÔMICA	11,467
PAO	INTERMEDIARIA	11,067
QUEIJO	ECONÔMICA	4,133

Tabela 8.1: Cesta básica gerada pelo Bom Cupom

Na tabela 8.2 podemos ver os produtos comprados por esse cliente.

Com a execução do gerador de cupons, os produtos sugeridos para desconto foram os seguintes:

Política de Fidelização: PAO FRANCES KG

Política de Upgrade de Marca: PRESUNTO SADIA MAGRO

Política de Introdução de Novo Produto: ASA FRANGO RESF KG

Os resultados observados acima estão de acordo com as expectativas:

Para a Política de Fidelização, foi escolhido de fato um produto que o cliente já costuma comprar, como pode ser visto na tabela de produtos do cliente.

Para a Política de Upgrade de Marca, podemos ver que o produto selecionado é um que o cliente não costuma comprar, mas existem produtos do mesmo avatar nas suas notas fiscais. Uma análise mais minuciosa do banco de dados revela que o produto selecionado tem a gama intermediária, o que condiz com o modelo proposto, pois a gama predominante para FRIOS na cesta básica do cliente é econômica.

Nota Fiscal	Produto	Avatar
001	PAO FRANCES KG	PAO
001	PAO FRANCES KG	PAO
001	PAO FRANCES KG	PAO
001	QJ PRATO FATIADO KG	QUEIJO
001	PRESUNTO PERDIGAO MAGRO	FRIOS
001	SALAME NOBRE ITALIANO K	FRIOS
001	MORT PERDIGAO OURO KG	FRIOS
002	PAO FRANCES KG	PAO
002	PAO FRANCES KG	PAO
002	PAO FRANCES KG	PAO
002	SALAME NOBRE ITALIANO K	FRIOS
002	SALAME NOBRE ITALIANO K	FRIOS
003	PAO FRANCES KG	PAO
003	PAO FRANCES KG	PAO
003	MORT PERDIGAO OURO KG	FRIOS
003	PRESUNTO PERDIGAO MAGRO	FRIOS
003	SALAME NOBRE ITALIANO K	FRIOS
004	QJ PRATO FATIADO KG	QUEIJO
004	PAO FRANCES KG	PAO
004	PAO FRANCES KG	PAO
004	PAO FRANCES KG	PAO
004	SALAME NOBRE ITALIANO K	FRIOS
004	SALAME NOBRE ITALIANO K	FRIOS
004	PRESUNTO PERDIGAO MAGRO	FRIOS
004	PRESUNTO PERDIGAO MAGRO	FRIOS

Tabela 8.2: Produtos comprados por um cliente

Para a política de Introdução de Novo Produto analisamos também a tabela 8.3, que apresenta as regras de associação mais relevantes para essa análise. Podemos ver que o produto sugerido com essa política é do avatar CARNE AVES, e de fato esse avatar está presente na consequência de duas regras nessa tabela, ou seja, o produto pode ter sido gerado pelo avatar PAO ou pelo avatar FRIOS da cesta básica do cliente, aquele que tiver a gama predominante correspondente à do produto selecionado.

Confirmamos assim as expectativas segundo as sugestões de produtos para as três políticas propostas.

Esse módulo tem como objetivo a seleção dos produtos para a geração dos cupons de desconto. Como descrito na seção 3.8.1, a composição final do cupom depende de uma análise financeira e das operações do supermercado.

Para que os cupons sejam gerados de fato, seria preciso complementar este módulo com um processo de decisão que considere essas regras do negócio (que dependem de cada supermercado) e que possa montar e imprimir os cupons para

Premissa	Consequência	Conviccção
FRIOS	QUEIJO	2,82
FRIOS	PAO	2,57
PAO	QUEIJO	1,33
FRIOS	CARNE AVES	1,26
FRIOS	CARNE BOVINA	1,23
FRIOS	FRUTA	1,23
FRIOS	LEGUME	1,22
PAO	FRUTA	1,21
FRIOS	VERDURA	1,2
PAO	LEGUME	1,2
FRIOS	REFRIGERANTE	1,19
FRIOS	CAFÉ DA MANHA	1,18
PAO	CAFÉ DA MANHA	1,15
PAO	CARNE AVES	1,14
PAO	FRIOS	1,14
PAO	LEITE	1,14
FRIOS	LEITE	1,14
FRIOS	TEMPERO	1,13
FRIOS	MASSA	1,12
FRIOS	BISCOITO	1,11

Tabela 8.3: Regras de associação mais relevantes para esse exemplo

os clientes a partir dos produtos escolhidos.

9 Considerações Finais

Nesta seção, apresentamos as principais conclusões deste trabalho, assim como sugestões para possíveis trabalhos futuros.

9.1 Conclusão

Capazes de identificar padrões em uma grande quantidade de dados de forma eficaz, as técnicas de *data mining* apresentadas neste trabalho podem ser aproveitadas em um universo muito grande de aplicações. Devido a sua generalidade, estas técnicas têm um valor enorme, tanto cientificamente quanto industrialmente, independente da área de aplicação. Desse ponto de vista, este trabalho cumpriu uma de suas missões acadêmicas: nos expôs a um desafio em uma promissora área do conhecimento, cujo aprendizado poderá ser utilizado em projetos futuros.

Ao longo deste projeto nos preocupamos não somente com os aspectos técnicos da utilização de *data mining* para identificação de padrões. O exercício de imaginar como seria a utilização do Bom Cupom em um supermercado real, dadas suas restrições operacionais e de negócio, nos ajudou a vislumbrar mais facilmente as possibilidades de aplicação dessas técnicas e os obstáculos à sua implantação.

Durante a execução do projeto constatamos a dificuldade de se trabalhar eficientemente com uma grande quantidade de dados, ainda que a massa crítica de dados da qual dispúnhamos fosse relativamente pequena em relação ao que se encontra na prática. Outra grande dificuldade foi a de trabalhar com informações pouco estruturadas e incompletas, o que exigiu um trabalho de pré-processamento de dados não negligenciável, senão o mais importante.

Notou-se também uma dificuldade muito grande em se obter mais dados reais de vendas de um supermercado, pois esta é uma informação de alto valor estratégico de um negócio, mantida sob sete chaves por seus detentores. O acesso a uma maior quantidade de dados, de diversas fontes, seria de alta valia para a aprimoração da metodologia de geração de descontos personalizados proposta

neste trabalho.

Do ponto de vista técnico, a metodologia proposta neste trabalho se mostrou eficiente na identificação de padrões de consumo de um supermercado. Os resultados obtidos foram satisfatórios, pois as regras de associação foram geradas de forma automatizada e os resultados (produtos presentes nos cupons de descontos gerados) se mostraram plausíveis quando analisados individualmente.

Esperamos que os conceitos, a metodologia e as referências apresentados neste trabalho sejam úteis ao leitor interessado em se aprofundar no assunto, seja ele específico ao mundo dos supermercados ou não.

9.2 Extensões e Trabalhos Futuros

Como possíveis extensoões e trabalhos futuros propomos:

- A utilização de um banco de dados mais completo que o deste trabalho, de forma a poder utilizar parâmetros como a frequência, o intervalo médio e o atraso de compra, especificados na seção 7.3.
- A contribuição para o desenvolvimento de uma WordNet brasileira, o que tornaria possível a classificação dos produtos usando a técnica de comparação semântica.
- A aplicação de técnicas de *data mining* para a análise de padrões de consumo de diferentes supermercados, de forma a criar um modelo de consumo que não seja restrito aos clientes de apenas um supermercado.
- A elaboração de uma análise financeira para determinar, dentre os produtos candidatos a desconto segundo uma das políticas do supermercado, qual destes trará um maior retorno financeiro ao estabelecimento caso o desconto seja concedido.
- A elaboração de uma interface de acompanhamento de promoções para o supermercado, para que os seus gestores possam avaliar em tempo real o desempenho de suas promoções.

Referências

AGIRRE, E. et al. A study on similarity and relatedness using distributional and wordnet-based approaches. In: *Proceedings of Human Language Technologies: The 2009 Annual Conference of the North American Chapter of the Association for Computational Linguistics*. Stroudsburg, PA, USA: Association for Computational Linguistics, 2009. (NAACL '09), p. 19–27. ISBN 978-1-932432-41-1. Disponível em: <http://dl.acm.org/citation.cfm?id=1620754.1620758>.

BRIN, S. et al. Dynamic itemset counting and implication rules for market basket data. *SIGMOD Rec.*, ACM, New York, NY, USA, v. 26, n. 2, p. 255–264, jun. 1997. ISSN 0163-5808. Disponível em: <http://doi.acm.org/10.1145-/253262.253325>.

BUDANITSKY, A.; HIRST, G. Evaluating wordnet-based measures of lexical semantic relatedness. *Computational Linguistics*, MIT Press, v. 32, n. 1, p. 13–47, 2006.

CHIB, S.; SEETHARAMAN, P.; STRIJNEV, A. Analysis of multi-category purchase incidence decisions using iri market basket data. *Advances in Econometrics*, Emerald Group Publishing Limited, v. 16, p. 57–92, 2002.

DEPAIVA, V.; RADEMAKER, A. Revisiting a brazilian wordnet. In: *Proceedings of Global Wordnet Conference*. Matsue: Global Wordnet Association, 2012. Http://www.globalwordnet.org/gwa/gwa_conferences.html.

FAYYAD, U. M.; IRANI, K. B. The attribute selection problem in decision tree generation. In: *Proceedings of the tenth national conference on Artificial intelligence*. AAAI Press, 1992. (AAAI'92), p. 104–110. ISBN 0-262-51063-4. Disponível em: <http://dl.acm.org/citation.cfm?id=1867135.1867151>.

FELLBAUM, C. Wordnet. In: POLI, R.; HEALY, M.; KAMEAS, A. (Ed.). *Theory and Applications of Ontology: Computer Applications*. Springer Netherlands, 2010. p. 231–243. ISBN 978-90-481-8847-5. 10.1007/978-90-481-8847-5_10. Disponível em: <http://dx.doi.org/10.1007/978-90-481-8847-5_10>.

HAN, J. et al. Frequent pattern mining: Current status and future directions. *Data Mining and Knowledge Discovery*, Springer, v. 15, n. 1, p. 55–86, 2007.

HAN, J.; KAMBER, M. *Data Mining: Concepts and Techniques*. Elsevier Science & Tech, 2006. (The Morgan Kaufmann Series in Data Management Systems Series). ISBN 9781558609013. Disponível em: <http://books.google-.com.br/books?id=AfL0t-YzOrEC>.

HAN, J.; PEI, J.; YIN, Y. Mining frequent patterns without candidate generation. *SIGMOD Rec.*, ACM, New York, NY, USA, v. 29, n. 2, p. 1–12, maio 2000. ISSN 0163-5808. Disponível em: <http://doi.acm.org/10.1145/335191-.335372>.

HYNDMAN, R. J.; FAN, Y. Sample quantiles in statistical packages. *The American Statistician*, v. 50, n. 4, p. 361–365, 1996. Disponível em: <http://www.tandfonline.com/doi/abs/10.1080/00031305.1996.10473566>.

MARRAFA, P. Portuguese WordNet: general architecture and internal semantic relations. *DELTA: Documentação de Estudos em Lingüística Teórica e Aplicada*, scielo, v. 18, p. 131 – 146, 00 2002. ISSN 0102-4450.

MILD, A.; REUTTERER, T. An improved collaborative filtering approach for predicting cross-category purchases based on binary market basket data. *Journal of Retailing and Consumer Services*, v. 10, n. 3, p. 123 – 133, 2003. ISSN 0969-6989. Model Building in Retailing and Consumer Services. Disponível em: <http://www.sciencedirect.com/science/article/pii/S0969698903000031>.

MILLER, G. A. et al. Introduction to wordnet: An on-line lexical database*. *International Journal of Lexicography*, v. 3, n. 4, p. 235–244, 1990. Disponível em: <http://ijl.oxfordjournals.org/content/3/4/235.abstract>.

NEDERSTIGT, L.; VANDIC, D.; FRASINCAR, F. An automated approach to product taxonomy mapping in e-commerce. In: CASILLAS, J.; MARTíNEZ-LóPEZ, F. J.; CORCHADO, J. M. (Ed.). *Management Intelligent Systems*. Springer Berlin Heidelberg, 2012, (Advances in Intelligent Systems and Computing, v. 171). p. 111–120. ISBN 978-3-642-30864-2. 10.1007/978-3-642-30864-2_11. Disponível em: <http://dx.doi.org/10.1007/978-3-642-30864-2_11>.

REUTTERER, T. et al. A dynamic segmentation approach for targeting and customizing direct marketing campaigns. *Journal of Interactive Marketing*, Wiley Online Library, v. 20, n. 3-4, p. 43–57, 2006.

RYGIELSKI, C.; WANG, J.-C.; YEN, D. C. Data mining techniques for customer relationship management. *Technology in Society*, v. 24, n. 4, p. 483 – 502, 2002. ISSN 0160-791X. Disponível em: <http://www.sciencedirect.com-/science/article/pii/S0160791X0200038>.

SEETHARAMAN, P. et al. Models of multi-category choice behavior. *Marketing Letters*. Springer, v. 16, n. 3, p. 239–254, 2005.

SILVA, B. Dias-da; FELIPPO, A. D.; NUNES, M. The automatic mapping of princeton wordnet lexical-conceptual relations onto the brazilian portuguese wordnet database. *Proceedings of the 6th LREC*, 2008.

TAYLOR, M. *It's Time for Big Data to Improve Customer Experience*. 2011. Disponível em: <http://razorfishoutlook.razorfish.com/articles/bigdata.aspx>.

TUNG, A. et al. Efficient mining of intertransaction association rules. *Knowledge and Data Engineering, IEEE Transactions on*, IEEE, v. 15, n. 1, p. 43–56, 2003.

VOSSEN, P. Introduction to eurowordnet. *Computers and the Humanities*. Springer Netherlands, v. 32, p. 73–89, 1998. ISSN 0010-4817. 10.1023/A:1001175424222. Disponível em: <http://dx.doi.org/10.1023-/A:1001175424222>.

WITTEN, I.; FRANK, E. *Data Mining: Practical Machine Learning Tools and Techniques*. [S.l.]: Morgan Kaufmann, 2005.

YANG, Y.; HAO, C. Product selection for promotion planning. *Knowledge and information systems*. Springer, v. 29, n. 1, p. 223–236, 2011.

Apêndice A – Processo de Utilização dos Cupons

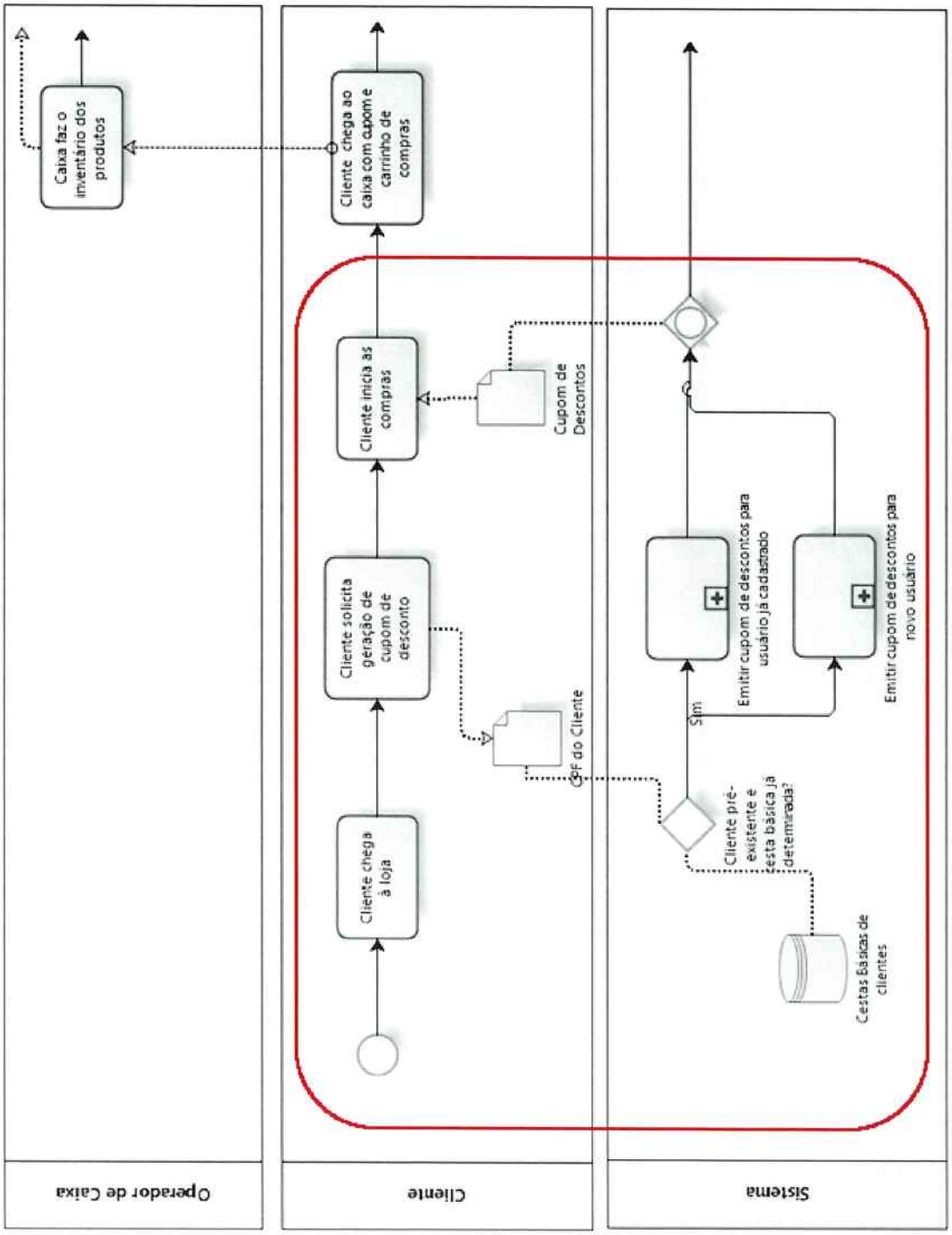

Figura A.1: Processo geral de utilização do sistema (Parte 1)

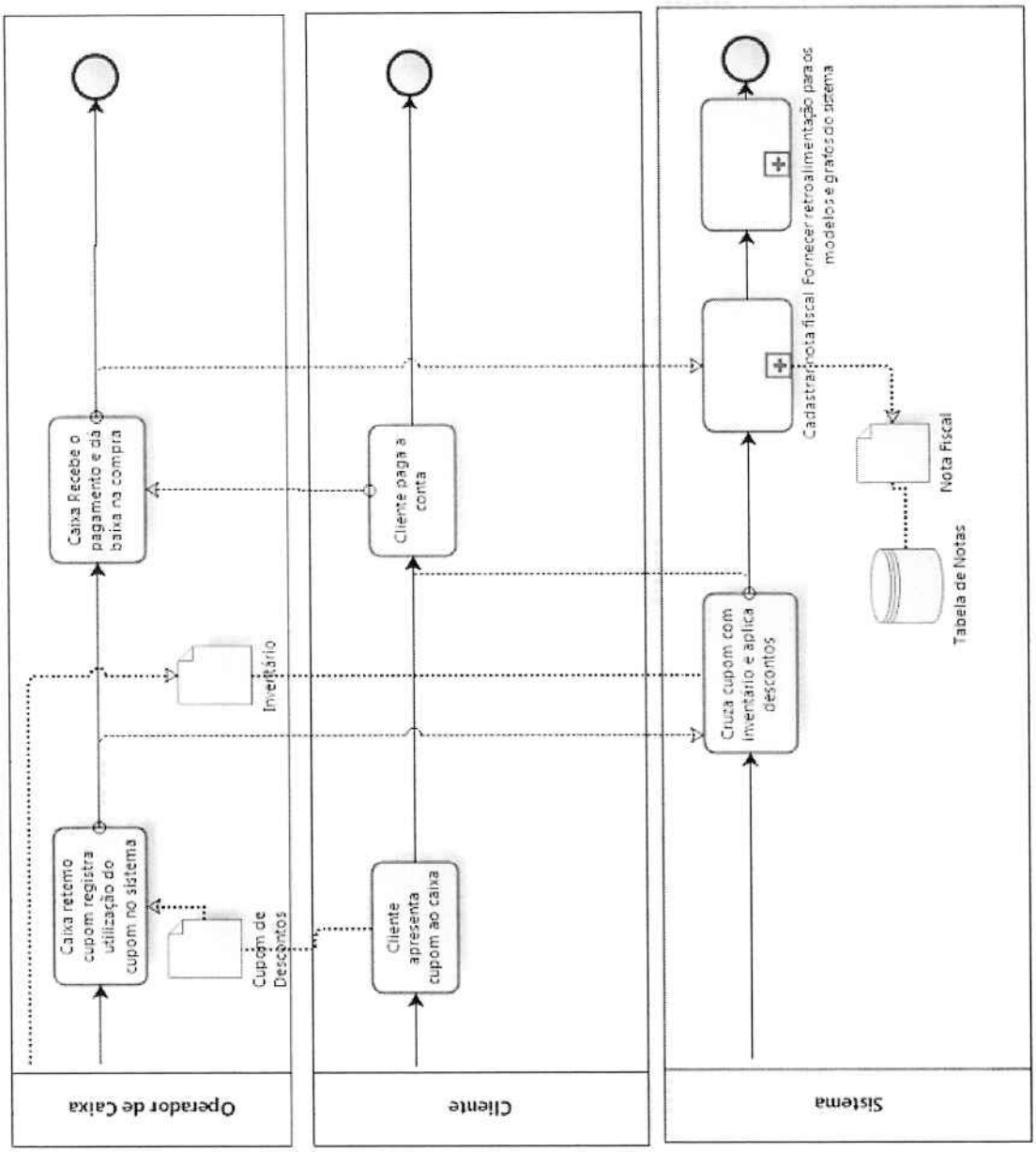

Figura A.2: Processo geral de utilização do sistema (Parte 2)

Apêndice B – Requisiçao à API do Buscapé

B.1 Exemplo de URL de Requisição

URL de consulta de ofertas do produto cujo código de barras é 7894321722016:

`http://sandbox.buscape.com/service/findOfferList/564771466d477a4458664d3d/?barcode=7894321722016`

B.2 Resposta à Requisição

```
<Result xmlns="urn:buscape" xmlns:xsi="http://www.w3.org
/2001/XMLSchema-instance" totalLooseOffers="0" schk="true"
 page="1" totalPages="1" totalResultsSellers="8"
totalResultsReturned="11" totalResultsAvailable="11" xsi:
schemaLocation="http://developer.buscape.com/admin/buscape
.xsd">
  <details>
    <applicationID>564771466d477a4458664d3d</applicationID
    >
    <applicationVersion>1.0.0.0</applicationVersion>
    <applicationPath>http://bws-apps.buscape.com/mobile/
    update</applicationPath>
    <date>2012-09-20T15:44:37.193-03:00</date>
    <elapsedTime>20</elapsedTime>
    <status>success</status>
    <code>0</code>
    <message>success</message>
  </details>
  <category hasOffer="true" isFinal="true"
  parentCategoryId="517" id="3261">
```

```
    <thumbnail url="http://imagem.buscape.com.br/bp5/
    categorias/3261.jpg"/>
    <links>
      <link type="category" url="http://compare.buscape.
      com.br/bebida-achocolatada.html?mdapp=100&mddtn
      =69672797"/>
      <link type="xml" url="http://sandbox.buscape.com/
      service/findOfferList/564771466d477a4458664d3d/br/?
      categoryId=3261"/>
    </links>
    <name>Bebida Achocolatada</name>
  </category>
  <offer id="114020907" categoryId="3261">
    <offerName>ACHOCOLATADO TODDYNHO 200ML</offerName>
    <links>
      <link type="offer" url="http://tracker.lomadee.com/
      tr/rd?b=
      bGUtBx8GZTodEjdvF2oHbwY4bglrFBgfKSgQEg9zc2VnbG--"/>
    </links>
    <thumbnail url="http://thumbs.buscape.com.br/T100x100/
    __2.1115285-6cbd22b.jpg"/>
    <price>
      <currency abbreviation="BRL"/>
      <value>0.99</value>
    </price>
    <seller oneClickBuyValue="0" oneClickBuy="false"
    advertiserId="0" pagamentoDigital="false"
    isTrustedStore="false" id="1115285">
      <sellerName>Primo Supermercado</sellerName>
      <links>
        <link type="seller" url="http://www.
        primosupermercado.com.br/"/>
      </links>
      <contacts/>
      <rating>
        <userAverageRating>
          <numComments>0</numComments>
          <rating>0.0</rating>
```

```
            </userAverageRating>
        </rating>
    </seller>
  </offer>
</Result>
```

Apêndice C – As Limitações da WordNet

Para a classificação de produtos poder-se-ia analisar o sentido semântico da string que representa o produto no banco de dados, ou daquela que o representa em qualquer um dos outros sistemas, caso o produto tenha sido encontrado através de seu código de barras.

Uma lista que associa a cada palavra a descrição do seu significado é chamada de dicionário. Já uma lista que associa a cada palavra uma lista de sinônimos recebe o nome thesaurus.

Porém, com um dicionário ou um thesaurus não é possível avaliar palavras de acordo com relações semânticas. Para poder avaliar se uma palavra está relacionada a outra semanticamente, essas estruturas não são suficientes, e faz-se necessário o uso de uma WordNet.

Definido por (MILLER et al., 1990), WordNet é uma estrutura em forma de grafo, onde cada nó representa um conjunto de sinônimos (chamado daqui para a frente de synset), e cada aresta representa uma relação semântica dentro de uma lista pré-definida. Alguns exemplos de relações semânticas descritos pela WordNet em inglês da Universidade de Princeton são:

Hipônimo: um synset é considerado hipônimo de um outro quando ambos se encontram dentro do mesmo campo semântico, porém o primeiro apresenta um sentido mais restrito, ou seja, é mais específico. Por exemplo, o synset que representa o conceito de bananeira é um hipônimo daquele que representa o conceito de árvore, que por sua vez é um hipônimo daquele que representa o conceito de planta;

Hiperônimo: é a relação inversa à hiponímia. Um hiperônimo é um termo dentro do mesmo campo semântico, porém com sentido mais abrangente, mais geral;

Merônimo: um synset é considerado merônimo de um outro quando descreve um conceito que pode ser classificado como parte do outro conceito. Por exemplo, o synset que representa o conceito de dedo é um merônimo daquele que representa o conceito de mão, que por sua vez é um merônimo daquele que representa o conceito de braço;

Holônimo: é a relação inversa à meronímia. Um holônimo descreve um conceito que engloba um outro conceito como parte de um todo.

Além disso a WordNet pode apresentar relações lexicais entre as palavras, como por exemplo:

Sinônimo: uma palavra é considerada sinônimo de outra quando ambas podem expressar o conceito de um mesmo synset. Sinonímia é uma relação simétrica (se A é sinônimo de B, então B é sinônimo de A) e não transitiva (se A é sinônimo de B, e B é sinônimo de C, não necessariamente A é sinônimo de C).

Antônimo: é a relação inversa à sinonímia. Uma palavra é considerada antônimo de uma outra quando a substituição da segunda pela primeira em uma sentença pode alterar seu valor lógico dentro de um contexto específico. Apesar de ser relativamente fácil para seres humanos reconhecer antônimos, é uma tarefa complexa definir formalmente o que são antônimos.

C.1 WordNet na Desambiguação de Strings

Como mostrado em estudos já feitos (AGIRRE et al., 2009), as relações representadas por uma WordNet podem ser utilizadas para determinar o quão próximo semanticamente estão dois synsets. Essa definição pode ser de certa maneira estendida para definir a proximidade semântica (daqui em diante chamada de similaridade semântica) entre duas palavras. Ao tratarmos com palavras, é comum as considerar como nós dentro do grafo da WordNet que se conectam aos nós dos synsets através de relações de inclusão (AGIRRE et al., 2009). Desta maneira podemos calcular a similaridade semântica entre duas palavras. Existem vários métodos de análise para determinar esse valor, levando em consideração as relações entre os nós da WordNet, como descritos em (BUDANITSKY; HIRST, 2006).

C.2 Similaridade Semântica na Classificação dos Avatares

O nome dos produtos disponíveis no sistema do supermercado não necessariamente devem seguir qualquer tipo de padronização, já que tem o simples propósito de tornar o sistema mais legível ao usuário. Porém, essa legibilidade proporcionada ao usuário carrega um aspecto semântico que pode ser explorado para determinar também a classificação em avatares de um produto.

Um problema que deve ser levado em consideração nessa abordagem é que muitas vezes essa string representando o nome do produto pode vir abreviada ou truncada, não representando necessariamente uma palavra do português. Porém, métodos de integração de plataformas e uma classificação inicial de avatares podem gerar também uma string que seja um pouco mais descritiva para o produto em questão (descrição obtida através da API do Buscapé, seção 5.4).

Considerando essas limitações, após a classificação prévia de alguns produtos através de outros procedimentos, uma sequência de passos pode ser seguida para tentar aprimorar essa classificação e expandi-la aos demais produtos que eventualmente não tenham sido classificados.

1. Busca-se uma string mais descritiva para cada produto, de maneira a aumentar o poder expressivo sobre o mesmo.

2. Identificam-se palavras que podem expressar mais adequadamente a descrição do produto em questão.

3. Definem-se os avatares como palavras que igualmente expressam significado sobre o avatar.

4. A partir da avaliação da similaridade semântica entre as palavras encontradas, pode-se determinar a qual classe de avatares um produto mais provavelmente pertence.

De fato, tal abordagem já foi utilizada por (NEDERSTIGT; VANDIC; FRASINCAR, 2012) se mostrando eficiente quando tratando strings que de fato descrevem o produto para um ser humano. Nesse projeto, nomes de produtos extraídos da taxonomia apresentada em sites de e-commerce, foram testadas contra as extraídas de outros sites para tentar fazer o mapeamento de produtos expostos em diferentes plataformas.

Para o nosso projeto, além de testar as strings contra os nomes dos avatares em si, podemos também testar contra strings de outros produtos que já tenham sido classificados por métodos anteriores. Nesse caso, porém, mantém-se o desafio de definir quais são as palavras que expressam com maior exatidão a descrição do produto.

C.3 WordNet Brasileira

É importante ressaltar, que assim como dicionários e thesauri, WordNets estão intrinsicamente ligadas ao idioma em que estão representadas. As relações nela expressas só são relevantes para análises de textos escritos nessa língua.

A ideia do desenvolvimento de uma WordNet que integre várias línguas, incluindo relações novas de equivalência entre termos e funcionalidades de tradução, foi apresentada por (VOSSEN, 1998). Porém esse projeto requer muito mais trabalho e esforço do que uma WordNet simples, considerando-se que o mapeamento entre termos semelhantes através de idiomas diferentes não é trivial. Atualmente ainda não atingiu proporções suficientes para ser utilizado numa análise válida.

O conceito de WordNet não é exatamente novo e uma versão em língua inglesa já está relativamente bem avançada com o projeto desenvolvido em Princeton a partir de 1986, como descrito por (FELLBAUM, 2010). Esse mesmo avanço, porém, não é observado quando se trata de WordNet em outras línguas, incluindo português brasileiro.

Iniciativas para o desenvolvimento de uma WordNet brasileira não faltam. Há um projeto de adaptação da WordNet de Princeton a partir da tradução de termos, descrito em (SILVA; FELIPPO; NUNES, 2008) e ainda um projeto pela PUC do Rio de janeiro muito recente de uma WordNet puramente brasileira (DEPAIVA; RADEMAKER, 2012). Porém tais projetos ainda não estão completamente funcionais e não se adequam ainda às análises de similaridade semânticas necessárias no contexto deste projeto.

Há também projetos de WordNet para português europeu pela Universidade de Lisboa que estão bem mais avançadas que os projetos brasileiros (MARRAFA, 2002), porém além de termos em jogo as diferenças linguísticas entre português europeu e brasileiro, esses projetos só estão disponíveis online, para consultas individuais, ou seja, não há como integrar num processo automatizado para explorar a função de cálculo de similaridade semântica entre palavras.

O uso da similaridade semântica é uma ferramenta poderosa para a desambi-

guação de termos que pode melhorar muito a classificação dos produtos em avatares neste projeto. Porém devido às limitações aqui apresentadas não é possível o uso dessas técnicas, simplesmente pela falta de uma WordNet totalmente operante em português brasileiro.