IV SIMGETEC: SIMPÓSIO DE GESTÃO E TECNOLOGIA

Centro Estadual de Educação Tecnológica Paula Souza
Faculdade de Tecnologia de Carapicuíba
13 a 16 de Junho de 2018
ISSN: 2526-6241

ANÁLISE DE ALGORITIMO DE RECOMENDAÇÃO DE VENDAS ONLINE

Carlos Henrique Pires Tanzi[1] – Fatec Carapicuíba
Mauro Ferreira de Lima[2] – Fatec Carapicuíba
Rodrigo Sipriano Pereira[3] – Fatec Carapicuíba
Paulo Rodrigues do Prado[4] – Fatec Carapicuíba
Prof. Dr. Luz Sérgio de Souza[5] – Fatec Carapicuíba

RESUMO

Sistemas de Recomendação (SR) constituem importantes ferramentas para portais web tais como sítios de comércio eletrônico. Para realizar recomendações, os SRs acessam fontes de dados variadas, as quais buscam capturar as características dos usuários, dos itens consumidos e de suas transações, Além da capacidade de predição, um outro fator que deve ser levado em consideração no desenvolvimento de SR é o tempo de resposta. Considerando este cenário, a presente pesquisa tem como objetivo verificar e analisar o desempenho em carga de um Sistema de Recomendação e, em particular, do algoritmo de Naive Bayes, bastante utilizado nos sistemas de vendas online, por ser de implementação relativamente simples se comparado com outras abordagens, e possuir um aumento de precisão proporcional à quantidade de dados usados para seu treinamento. Durante os testes foram avaliados parâmetros como vazão e tempo de resposta. Para isto um servidor TomCat foi implantado em um computador, onde um script CGI em Python estava rodando, então um script de teste foi criado com o Jmeter que foi usado para enviar as requisições http para o servidor e para coletar os dados de performance.

Palavras-chave: Sistemas Web. Naive Bayes. Big Data. Comércio Eletrônico. Java.

ABSTRACT

Recommendation Systems (SR) are important tools for web portals such as e-commerce sites. In order to make recommendations, SRs access a variety of data sources, which seek to capture the characteristics of users, items consumed and their transactions. In addition to predictability, another factor that must be taken into account in SR development is the response time. Considering this scenario, this research aims to verify and analyze the load performance of a Recommendation System, and in particular the Naive Bayes algorithm, widely used in online sales systems, because it is relatively simple to implement when compared to other approaches, and have a precision increase proportional to the amount of data used for your training. During the tests, parameters such as flow rate and response time were evaluated. For this a TomCat server was deployed to a computer where a CGI script in Python was running, then a test script was created with Jmeter that was used to send the http requests to the server and to collect the performance data.

Keywords: *Web Systems. Naive Bayes. Big data. E-commerce. Java.*

[1] - Análise e Desenvolvimento de Sistemas – *e-mail*: carlos.tanzi@fatec.gov.br
[2] - Análise e Desenvolvimento de Sistemas – *e-mail*: mauro.lima2@fatec.gov.br
[3] - Análise e Desenvolvimento de Sistemas – *e-mail*: rodrigo.sipriano@fatec.gov.br
[4] - Análise e Desenvolvimento de Sistemas – *e-mail*: paulo.rodrigues@fatec.gov.br
[5] - Análise e Desenvolvimento de Sistemas – *e-mail*: luiz.souza71@fatec.sp.gov.br – Orientador

Centro Estadual de Educação Tecnológica Paula Souza
Faculdade de Tecnologia de Carapicuíba
13 a 16 de Junho de 2018
ISSN: 2526-6241

1 INTRODUÇÃO

Com o avanço da tecnologia e principalmente com a expansão da internet, o homem passou a ter acesso a muitas coisas, mesmo que estas estejam muito longe de sua localização, pois realmente com poucos cliques é possível conhecer lugares jamais visitados, visualizar documentos com grande valor histórico, conhecer as tradições e costumes de outras civilizações, ou até mesmo comprar produtos, aproveitando-se da comodidade que a internet pode oferecer.

Caminhamos cada vez mais em direção a um mundo em que todas as coisas estarão conectadas à internet, a chamada internet das coisas, todos esses objetos serão meios de acesso em potencial, permitindo que pessoas comprem produtos da internet (Davis, 2008; W3C, 2010 apud Lacerda, 2012). Não obstante, muitas vezes um indivíduo possui pouca ou quase nenhuma experiência pessoal para realizar escolhas dentre alternativas que lhe são apresentadas.

Em geral, para dirimir dúvidas e necessidades frente à escolha, considera-se recomendações que são passadas por outras pessoas de forma direta ou indiretamente por meio de textos de recomendação, opiniões de revisores ou mecanismos de curadoria.

Neste senário, Sistemas de Recomendação visam o aumento da capacidade e eficácia no processo de tomada de decisão. Em uma típica configuração, recomendações são fornecidas como entradas que o sistema compõe e direciona para indivíduos potencialmente interessados.

Os modelos de recomendação têm importância na personalização de aplicações, principalmente na web, pois buscam identificar preferências e sugerir itens relevantes para cada usuário, de acordo com a análise de seu comportamento de navegação, consultas, compra anteriores, preferências, entre outros. O desafio é realizar a combinação adequada entre as expectativas e necessidades dos usuários e os produtos, serviços e pessoas a serem recomendados aos mesmos. Alguns pontos que tem grande importância nestes sistemas são o de desempenho à medida que novos usuários acessam a aplicação, pois é necessário que o sistema possua estabilidade, eficiência para propor estas recomendações.

Desta forma o objetivo deste trabalho é compreender e analisar o comportamento de um Sistema de Recomendação e, em particular, a utilização do algoritmo de Naive Bayes, que tem grande importância nos ambientes web, pois a velocidade com que o usuário é atraído através de recomendações pode ter uma influência positiva para que este se aprofunde na navegação.

Centro Estadual de Educação Tecnológica Paula Souza
Faculdade de Tecnologia de Carapicuíba
13 a 16 de Junho de 2018
ISSN: 2526-6241

2 FUNDAMENTAÇÃO TEÓRICA

2.1 Comércio Eletrônico

O comércio eletrônico permite que uma infinidade de negócios seja realizada através da rede, especialmente pela internet. Existem diferentes formas de comércio eletrônico, como negócios entre empresas e seus consumidores (Business to Consumer – B2C) e entre as próprias empresas (Business to Business – B2B) (KUROSE, Jim, 2013).

2.2 Os sistemas de recomendação.

Sistemas de recomendação têm como propósito ofertar itens aos usuários conforme suas preferências. Estas preferências são preditas a partir de um histórico, por meio de técnicas computacionais de Aprendizado de Máquina Por exemplo, se no passado o usuário tiver avaliado um filme de comédia, o sistema tenderá a recomendar outro filme do mesmo gênero (RICCI, Francesco, ROKACH, Lior e SHAPIRA, Bracha, 2011).

Para realizar uma recomendação baseada em conteúdo os sistemas sintetizam o conteúdo dos itens candidatos à recomendação e dos perfis dos usuários por meio da extração de atributos. Conhecendo os perfis, conteúdos, o sistema deve ser capaz de recomendar com maior assertividade novos itens aos usuários.

Na verdade, existem diversas razões para que esta tecnologia seja usada, entre estas razões podemos destacar:

a) O aumento do número de itens vendidos;

É o objetivo para todo negócio, os sistemas de recomendações nestes casos oferecem uma função importante para os negócios das lojas virtuais. Estes objetivos de vendas são alcançados em muitos dos casos, pois estas recomendações atendem as necessidades e desejos do usuário;

b) Vender itens mais diversos;

Outra função muito importante é apresentar ao usuário produtos que podem ser difíceis de encontrar sem uma recomendação precisa.

c) Aumentar a satisfação do usuário;

Uma recomendação bem direcionada permite melhorar a experiência do usuário com o site ou aplicativo. O usuário encontrará as recomendações interessantes, permitindo uma navegação rápida e agradável. A combinação de recomendações precisas e uma interface agradável farão

Centro Estadual de Educação Tecnológica Paula Souza
Faculdade de Tecnologia de Carapicuíba
13 a 16 de Junho de 2018
ISSN: 2526-6241

com que o usuário goste de utilizar o sistema, isso aumentará o uso do sistema e a probabilidade de que as recomendações sejam aceitas.

d) Aumentar a fidelidade do usuário:

O usuário é reconhecido pelo sistema que o trata como um cliente valioso e faz recomendações conforme seu histórico em acessos anteriores;

Quanto mais tempo o usuário interage com o sistema mais refinado se torna a recomendação de suas preferências, ou seja, toda interação se dá de forma personalizada.

e) Compreender melhor o que o usuário deseja.

Outra função dos sistemas de recomendação que podem ser utilizadas em diversas aplicações é a preferências dos usuários. Estas podem ser coletadas explicitamente ou previstas pelo sistema, podendo-se assim utilizar estes conhecimentos para, por exemplo, gerenciar estoques, determinar quantidades a serem produzidas de um item (RICCI, Francesco, ROKACH, Lior e SHAPIRA, Bracha, 2011).

O sistema de recomendação é um esforço multidisciplinar, este foi favorecido com diversos resultados obtido em vários campos de estudos da ciência da computação, tais como Aprendizado de Máquina, Mineração de Dados, interação humana-computador entre outras.

O Aprendizado de Máquina e a Mineração de Dados como subcampos da inteligência artificial permitem que o computador aprenda a executar de forma adequada certas tarefas, estando este munido de dados e/ou experiências (RICCI, Francesco, ROKACH, Lior e SHAPIRA, Bracha, 2011).

2.2.1 Identificação de perfis de usuários do Sistema

Para que seja construído um perfil de usuário com precisão, é necessário que os sistemas sejam capazes de representar com grande grau de acerto os interesses atuais dos destes, pois esta precisão é vital para que estes sistemas sejam bem sucedidos ao realizarem recomendações, sendo estas apropriadas para cada usuário. Muitas abordagens são utilizadas de forma a descrever o perfil de um usuário, entre elas podem ser mencionadas os históricos de compras, um classificador incluindo redes neurais, árvores de decisões, regras induzidas ou redes bayesianas (MONTANER, Miguel; LÓPEZ, Beatriz; LLUÍZ DE LA ROSA, Josep, 2003).

2.2.2 Modelos Classificadores

Estes modelos permitem aos sistemas que a atribuição de uma categoria a uma entrada. As construções de sistemas de recomendação baseadas nestes modelos significam a utilização de informações de um perfil como a entrada do sistema e como saída uma categoria para que sejam feitas recomendações para usuário (MONTANER, Miguel; LÓPEZ, Beatriz; LLUÍZ DE LA ROSA, Josep, 2003 apud Jennings e Higuchi, 1993).

2.3 O algoritmo de Naive Bayes

Naive Bayes é um recurso de aprendizagem de máquina, utilizado para classificação, este considera as variáveis como independentes, sendo por este motivo conhecido como Naive (ingênuo em Português). É um método simples de compreender e de fácil implementação, muito utilizado em linguagem natural (CHICARELLI, Ronaldo, 2017).

O algoritmo de Naive Bayes pode ser usado em diversas aplicações devido a sua eficácia, eficiência e simplicidade, como na detecção de doenças do coração (RAJKUMAR, Asha; REENA, G. Sophia, 2010). O algoritmo de Naive Bayes precisa de um conjunto de dados para ser treinado, sendo preferível a utilização de um único conjunto para diminuir a variância, esse conjunto de dados alimentarão uma tabela para o cálculo de probabilidades, que serão posteriormente utilizados nos cálculos do algoritmo (WEBB, Geoffrey; BOUGHTON, Janice; WANG, Zhihai, 2005). Para o entendimento do classificador de Naive Bayes é exigido o conhecimento de alguns conceitos.

> O treinamento do classificador Naive Bayes envolve o cálculo de uma distribuição geradora Pr(d|c) para cada classe c∈{-1,1}. Na fase de classificação, simplesmente é calculada qual distribuição tem a maior probabilidade de ter gerado cada documento. Nas máquinas Bayesianas, a criação de documentos é modelada como o seguinte processo:
>
> 1. Cada classe c possui uma probabilidade a priori associada Pr(c) tal que ∑c Pr(c) =1.
> 2. Dado que existe uma distribuição de documentos Pr(d|c) associada a classe c escolhida, esta distribuição é usada para gerar o documento. Sendo assim, a probabilidade de se gerar um documento da classe c é Pr(c)Pr(d|c). Finalmente, dado um documento d a probabilidade a Posteriori de que d foi gerado da classe c é:
>
> $$\Pr(c|d) = \frac{\Pr(c)\,\Pr(d|c)}{\sum_{\gamma}\Pr(\gamma)\,\Pr(d|\gamma)}$$
>
> Onde γ itera sobre ambas as classes de forma que Pr(c|d) torna-se uma medida de probabilidade apropriada. Pr(γ|c) é estimado através da modelagem das distribuição dos termos sobre as classes gerando um conjunto de parâmetros que chamaremos de Θ. A estimativa de Θ é baseada nos termos contidos nos documentos de treinamento. Após ter sido observado os dados de treinamento D, a distribuição a posteriori para Θ pode ser expressa por Pr(Θ|D).Dadas estas definições e dado um documento d, a probabilidade de d pertencer à classe c é dada por:
>
> $$\Pr(c|d) = \sum_{\theta}\Pr(c|d,\theta)\,\Pr(\theta|D)$$

Centro Estadual de Educação Tecnológica Paula Souza
Faculdade de Tecnologia de Carapicuíba
13 a 16 de Junho de 2018
ISSN: 2526-6241

$$= \sum_{\theta} \frac{\Pr(c|\theta)\Pr(d|c,\theta)}{\sum_{\gamma}\Pr(\gamma,\theta)\Pr(d|\gamma,\theta)} \Pr(\theta|D)$$

(CARRILHO JUNIOR, 2007, p.64)

2.4 Machine Learning

O Aprendizado de Máquina é um método que realiza a análise de dados automatizando a construção de modelos analíticos, ele é uma das ramificações da inteligência artificial no qual se baseia da ideia de que os sistemas podem aprender com dados, identificando padrões e tomando as decisões necessárias sem que haja a intervenção humana.

Seu desenvolvimento se deu a partir do momento em que pesquisadores interessados no assunto queriam saber se os computadores eram capazes de aprender com dados. Estes aspectos de independência são importantes, pois conforme os modelos são expostos a novos dados, estes conseguem se adaptar de forma autônoma (DAVENPORT, Thomas H, 2016).

O aprendizado se dá por meio de cálculos anteriores que produzem decisões e resultados confiáveis. Como exemplos de aplicações que utilizam Aprendizado de Máquina temos:

a) Os carros autônomos que estão sendo desenvolvidos pela Google;
b) Ofertas sugeridas pela loja virtual da Amazon;
c) Sugestão de filmes do Netflix etc.

O aumento do volume e da variedade de dados, a alta capacidade de processamento, o armazenamento de grande quantidade de informações fez com que o interesse por este método fosse renovado, possibilitando a produção de modelos capazes de analisar dados maiores e complexos de forma rápida e automática, devolvendo resultados mais rápidos e precisos.

Com a construção de dados precisos, as empresas podem identificar oportunidades ou evitar riscos que possam surgir. Para a criação de bons sistemas de Aprendizagem de Máquina é necessário conhecer como preparar os dados, fazer uso de algoritmos básicos e avançados, utilizar processos interativos e por fim permitir que estes sejam escaláveis (DAVENPORT, Thomas H, 2016).

Muitas empresas que trabalham com uma quantidade gigantesca de dados reconhecem a importância da tecnologia de Aprendizagem de Máquina. Ao extrair insights desses dados em muitos casos em tempo real as organizações são capazes de trabalhar com mais eficiência ou de ganhar uma vantagem competitiva sobre seus concorrentes. O Aprendizado de máquina pode ser usado em:

Centro Estadual de Educação Tecnológica Paula Souza
Faculdade de Tecnologia de Carapicuíba
13 a 16 de Junho de 2018
ISSN: 2526-6241

(a) Bancos e entidades financeiras para prevenção de fraudes e busca de insights importantes em seus dados;
(b) Governo de um modo geral para a mineração de dados de forma a buscar melhoria dos serviços e economia de dinheiro;
(c) Áreas médicas, onde auxiliam os médicos nas análises de dados que levem ao aperfeiçoamento de diagnósticos e tratamentos;
(d) Marketing e vendas, onde através da análise do histórico de compras dos clientes é efetuada a recomendação de produtos e serviços de forma personalizada.
(e) Transporte, na identificação de caminhos e rotas mais eficientes (DAVENPORT, Thomas H, 2016).

2.5 Big Data

Este termo faz referência a um grande conjunto de dados armazenados, sendo estas grandes quantidades de dados que serão processados e analisados. Estes dados podem ser encontrados na forma estruturada ou não estruturada. A ato de reunir e armazenar um grande volume de dados para posterior análise é bem antigo. Este conceito ganhou força nos anos 2000, quando Doug Laney, articulou a definição de big data como os 3V's (volume, velocidade e variedade). Atualmente são considerados 5V's (volume, velocidade, variedade, veracidade e valor).

a) Volume: relacionado com a grande quantidade de dados a serem processados.
b) Velocidade: referente à rapidez com que os dados devem ser processados para que dependendo de sua finalidade não se tornem obsoletos;
c) Variedade: representam as diversas fontes, como e-mails, redes sociais etc;
d) Veracidade: referente à confiança que se tem nos dados coletados;
e) Valor: o valor que a coleta e análise dos dados trarão para o negócio. A partir das informações coletadas e decodificadas, um analista de marketing pode entender seu público e oferecer produtos personalizados para cada nicho (SAS, 2015).

2.6 Python

Esta linguagem de programação foi desenvolvida em meados de 1980 por Guido van Rossum, ele trabalhava no CWI (*Centrum Wiskunde & Informatica*, em Amsterdã, Holanda). A linguagem recebeu esse nome por causa do seriado de comédia da BBC *Monty Python's Flying Circus* da qual Guido van Rossum é um fã. O Python tornou-se um projeto de

ISSN: 2526-6241

desenvolvimento de software de código-fonte aberto, assim como outros. Seu criador, Guido van Rossum tem um papel fundamental na decisão sobre a linha de evolução da linguagem. Para consolidar esse papel, ele recebeu da comunidade Python o título de "Benevolente Ditador para a Vida".

Python é uma linguagem de uso geral, sendo projetada com o intuito de tornar os programas bastante legíveis. Python também dispõem de uma farta biblioteca, o que torna possível a criação de aplicações bem avançadas usando código de aparência relativamente simples. Esses motivos fazem com que o Python se torne a cada dia mais popular e uma preferência como "primeira" linguagem de programação (PERKOVIC, Ljubomir, 2016).

2.7 A linguagem Java para a Web

A linguagem Java é muito utilizada no desenvolvimento de aplicações Web devido ao seu grande poder, pois é uma linguagem de programação interpretada e orientada a objetos, ela foi desenvolvida nos anos 90. Uma equipe de programadores da empresa Sun Microsystems e subordinados a James Gosling foram os responsáveis por esta criação. Outras linguagens de programação são compiladas para o código nativo, no entanto a linguagem Java é compilada para um *bytecode* que é interpretado por uma máquina virtual (*Java Virtual Machine*, mais conhecida pela sua abreviação JVM). Uma JVM isola totalmente a aplicação do sistema operacional. A plataforma java é dividida em 3 segmentos:

a) *Java Micro Edition*: Seguimento destinado aos dispositivos móveis, compostos de máquinas virtuais otimizadas e indicadas para ambientes mais específicos;
b) *Java Standard Edition*: Integra elementos padrão, permitindo o desenvolvimento de aplicações de pequeno e médio porte. Neste segmento estão inclusas todas as APIs de base, assim como a máquina virtual padrão.
c) *Java Enterprise Edition*: Voltada para o desenvolvimento de aplicações complexas. São adicionadas APIs específicas aos elementos padrão da plataforma (JUNIOR, PETER JANDL, 2014).

3 PROCEDIMENTOS METODOLÓGICOS

A presente pesquisa busca compreender e analisar o comportamento do algoritmo de Naive Bayes em sistemas de recomendação. Por ser de implementação relativamente simples, se comparado com outras abordagens, e por apresentar um aumento de precisão proporcional à quantidade de dados usados para seu treinamento, este algoritmo é muito utilizado nos

Centro Estadual de Educação Tecnológica Paula Souza
Faculdade de Tecnologia de Carapicuíba
13 a 16 de Junho de 2018
ISSN: 2526-6241

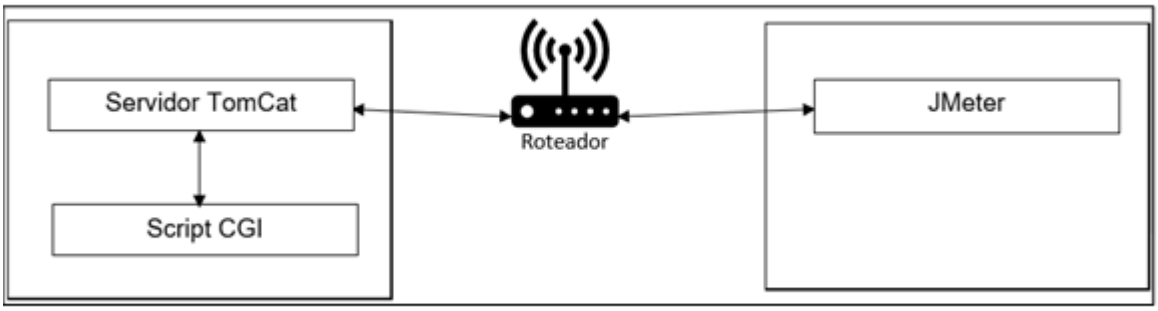
sistemas de vendas online (COMEIRO FERREIRA, Fernando, APARECIDA DE OLIVEIRA, Adicinéia, 2012). Neste trabalho para a implementação do algoritmo de Naive Bayes, foi usado a linguagem Python e um conjunto de dados consistindo dos hábitos de compra de 80 usuários em 2 categorias não nomeadas, pois as categorias teriam que ser convertidas para valores numéricos, então elas já foram criadas em formato numérico para poupar tempo. Essa base foi criada aleatoriamente no momento do treinamento do algoritmo e guardada em um arquivo em formato csv, tendo em vista a previsibilidade dos resultados esperados como saída do algoritmo, os dados presentes na base consistem de valores numéricos, o número 0 representa a primeira categoria e o número 1 a segunda categoria, seguido de um valor binário indicando a compra de produtos dessa categoria. Esse tipo de abordagem quanto à base de dados foi escolhido para facilitar o carregamento dos dados no algoritmo, evitando-se assim conversão de dados textuais para valores numéricos, pois o algoritmo apenas trabalha com valores numéricos, portanto essa base de dados foi criada tendo com referência uma base de dados de um sistema de recomendação de uma pizzaria, que foi obtido em um site de repositórios de aprendizado de máquina da Universidade da Califórnia, para ser usado no treinamento do algoritmo (BURKE, Robin, 2000). Para a realização do treinamento foi implementado um vetor multidimensional X com os dados das categorias de itens acessados pelos usuários e se o mesmo comprou um produto dessa categoria, obtidos a partir do conjunto de dados, um segundo vetor Y de tamanho igual ao primeiro também foi criado. Cada posição do vetor Y contém a informação da categoria que os dados da mesma posição no vetor X devem pertencer, para este trabalho foram determinadas duas categorias, uma categoria de recomendar e outra de não recomendar. Em seguida os dados foram distribuídos em uma distribuição Gaussiana, porque o algoritmo nesse tipo de implementação assume que os dados são contínuos e calcula sua variância de modo a permitir que o algoritmo faça uma predição com a inserção de dados, baseado na probabilidade desses dados pertencer às categorias, de acordo com a distância que o dado está da média da categoria, considerando que quanto mais distante da média está o dado menor a probabilidade de ele pertencer à média dos dados usados nos vetores para treinamento (MACHADO, Fábio P.;PAULA, Gilberto A., 2016) (CHAN, Tony F.; GOLUB, Gene H.; LEVEQUE, Randall J.,1979).

4 DESENVOLVIMENTO

Centro Estadual de Educação Tecnológica Paula Souza
Faculdade de Tecnologia de Carapicuíba
13 a 16 de Junho de 2018
ISSN: 2526-6241

Para realizar a análise proposta, a aplicação foi implementada sobre um servidor Tomcat, em um notebook rodando como um CGI (Common Gateway Interface), o notebook do servidor possui um processador i7-6500U com 2,5GH de frequência e 16GB de memória ram. Um robô para enviar as requisições HTTP, simulando 50 usuários virtuais, foi criado usando a ferramenta JMeter, o robô rodou em um segundo notebook com processador i7 e 8GB de memória ram, durante 1 hora. Foi determinado que o robô rodaria em uma segunda máquina, para evitar que o uso de processamento e memória RAM pelo JMeter afetasse o desempenho do script. A (Figura 1) mostra o diagrama de rede utilizado nos testes.

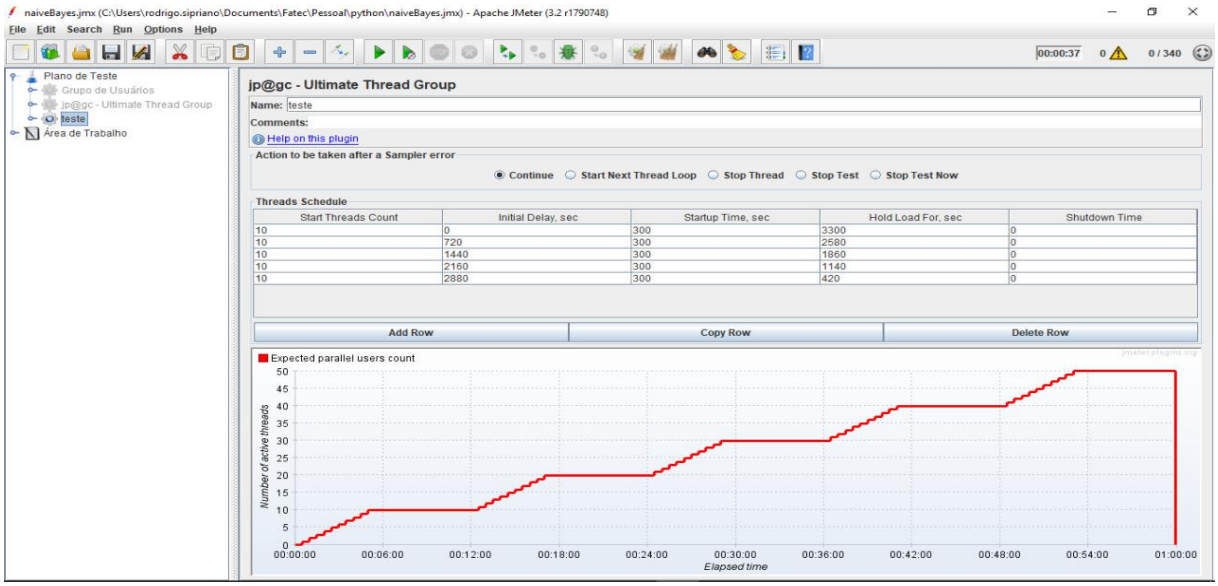
Figura 1 – Diagrama da rede local.



Fonte: Elaborada pelos Autores.

O JMeter também foi usado para coletar os dados de vazão e tempo de resposta, que foram enviados para um documento no formato csv, posteriormente usado na geração de gráficos dos dados obtidos, utilizando o framework Highcharts para este fim. Na (figura 2) é possível visualizar a interface do Software de teste.

Figura 2 – Interface do Jmeter.



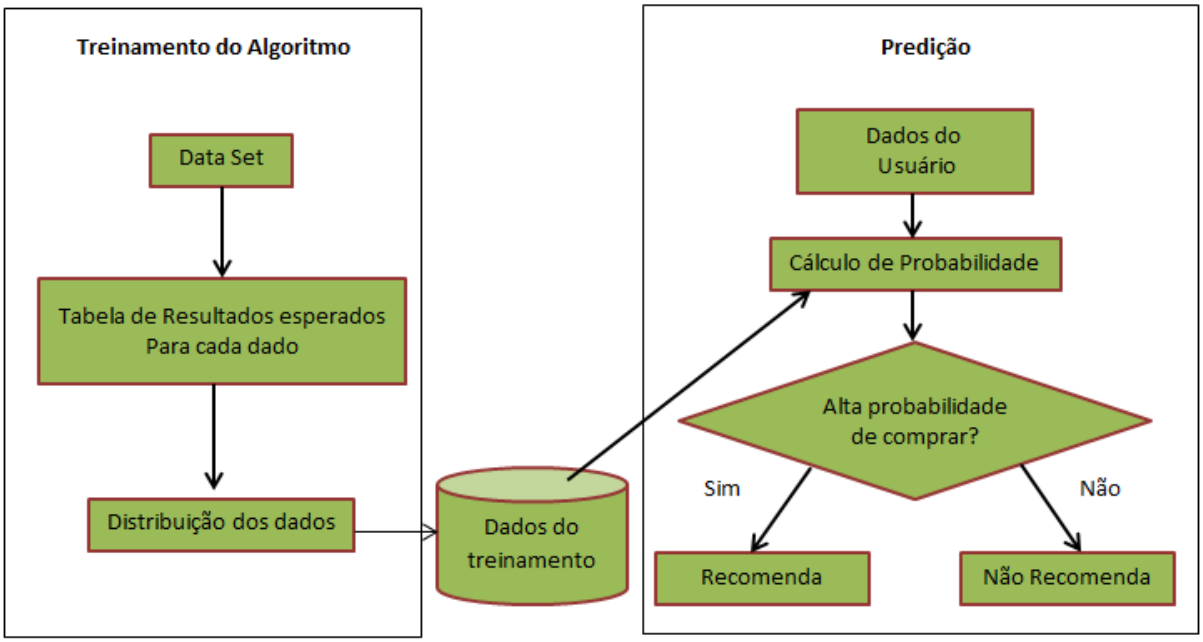
Fonte: Elaborada pelos Autores.

Centro Estadual de Educação Tecnológica Paula Souza
Faculdade de Tecnologia de Carapicuíba
13 a 16 de Junho de 2018
ISSN: 2526-6241

Para o monitoramento do uso de recursos do servidor, tais como uso de processador e de memória RAM, foi utilizado o gerenciador de tarefas do sistema operacional Windows. Uma rede local foi criada com um roteador, configurado para permitir acesso externo às máquinas da rede, permitindo assim o acesso do script pelo JMeter, utilizando o caminho deste na rede.

Na (figura 3) é possível visualizar um diagrama da arquitetura do sistema de recomendação.

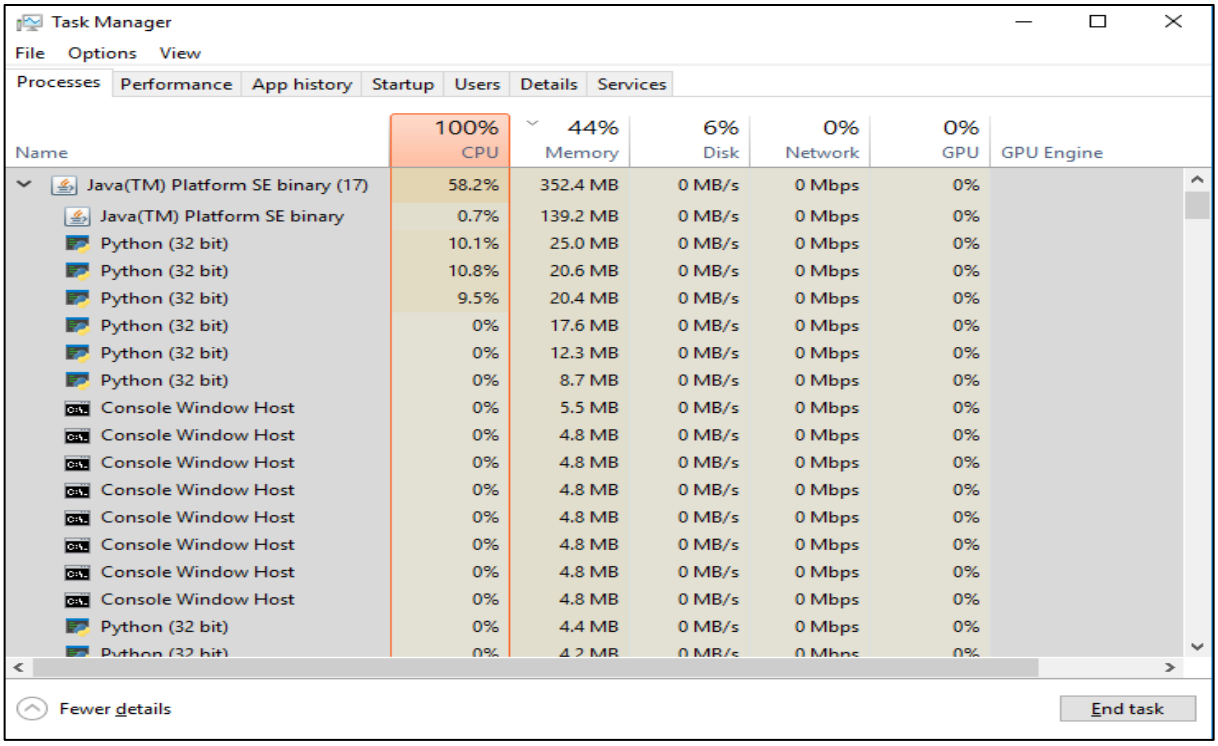
Figura 3 – Diagrama do Sistema de Recomendação.



Fonte: Elaborada pelos Autores.

O teste, após o fim de sua execução, reunião dos dados coletados e análise, mostrou que o mesmo do lado do servidor utilizou grande quantidade de processamento, atingindo 100% de utilização do processador (Figura 4), isso é devido ao fato do algoritmo realizar muitos cálculos para realizar a classificação e também devido ao script estar rodando em diversos processos concorrentes, fazendo o processador, mesmo após a troca de contexto para outro processo, continuar realizando cálculos quase ininterruptamente, a utilização de memória RAM, por sua vez, não apresentou nenhum comportamento que chamasse a atenção, o que pode ser explicado pelo fato de o algoritmo não precisar manter grandes quantidade de dados para realizar os cálculos.

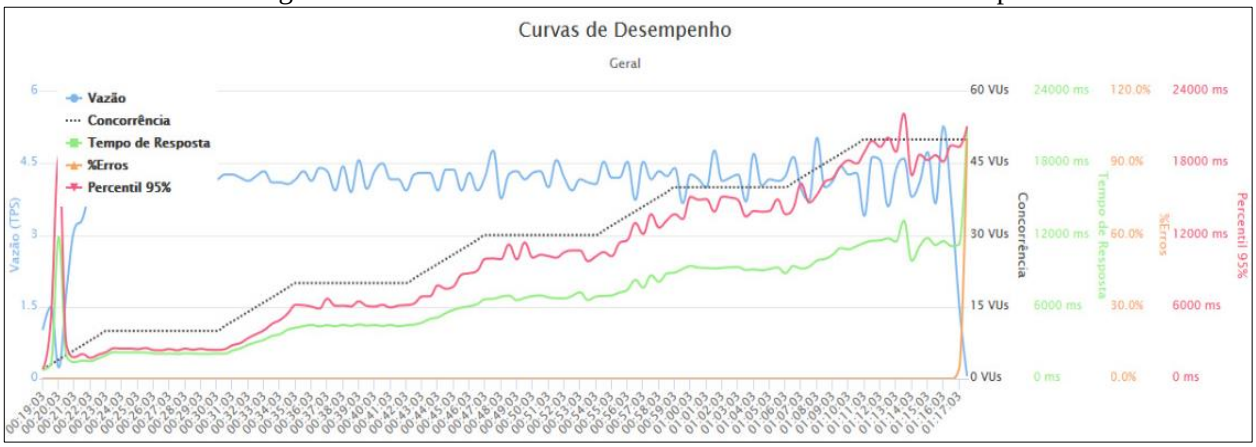
Figura 4 – Uso de recursos do servidor, pelos processos.



Fonte: Elaborada pelos Autores

Os dados obtidos pelo JMeter, ao se analisar as curvas de tempo de resposta e quantidade de usuários virtuais (Figura 5), demonstra que o aumento do tempo de resposta acompanha o aumento do número de usuários virtuais, fato esperado ao gerar maior concorrência do lado do servidor, causando enfileiramento de processos, para aguardar a liberação do processador e para serem atendidos e terem seus próprios cálculos realizados e enviar a resposta ao cliente que realizou a requisição. O tempo de resposta apresentou valores satisfatórios, no início do teste, com 10 usuários virtuais, o valor apresentado foi de 1,5s, valor que aumentou ao longo do teste e atingiu 12s com 50 usuários virtuais, em um momento que o servidor já tinha dificuldades para processar todas as requisições que chegavam, ao utilizar um percentil 95 no tempo de resposta, é possível verificar um tempo de resposta máximo nesse percentil de 22s ao final do teste. A vazão permaneceu estável ao longo do teste, sempre com valores em torno de 4 requisições por segundo e não apresentou erros, exceto ao final do teste devido ao encerramento dos usuários virtuais.

Figura 5 – Gráfico dos dados colhidos durante o teste de desempenho.



Fonte: Elaborada pelos Autores.

5 RESULTADOS E DISCUSSÕES

Os resultados encontrados no presente estudo confirmam que o algoritmo Naive Bayes apresenta um bom desempenho em ambiente web, com um tempo de resposta em torno de 1,5s com 10 usuários virtuais concorrentes atingindo algo em torno de 12s para 50 usuários virtuais concorrentes, sempre mantendo uma vazão em torno de 4 requisições por segundo. Tendo em vista que uma quantidade de 50 usuários virtuais é equivalente a uma quantidade até 100 vezes maior de usuários reais.

A confirmação do bom desempenho do Naive Bayes justifica-se: O tempo médio de resposta de um site de vendas é de 8,3s, 9,8s e 10,3s no Japão, Estados Unidos e Reino Unido respectivamente (MACHMETRICS, 2018).

Levando em consideração o carregamento de todos os recursos do site como imagens, scripts e html, se levado em consideração que um grande site de vendas, como a Amazon, teve 197 milhões de visitantes únicos em dezembro de 2017 (STATISTA, 2017) e convertendo esse valor para um número de usuários concorrentes da seguinte forma temos:

$\frac{U}{t} = Au\ (1)$ e $Vt.Ta.Tn\ (2)$, onde, Au representa acessos únicos, Vt representa, variação de tempo, Ta, representa tempo de acesso e Tn, representa tempo de navegação. Podemos converter primeiramente os valores mensais para valores diários, considerando os 30 dias do mês, que quase toda a carga do site ocorre durante 15 horas do dia e que o usuário permanece 15 minutos, temos:

$$x = \frac{197}{30.15.4} = 0{,}10944\ .10^{6}$$

Ou seja, um site como a Amazon tem cerca de 109 mil acessos diários, agora para obter um valor para acessos concorrentes, vamos converter para o número de usuários que acessam por

ISSN: 2526-6241

15 minutos, o tempo que estamos considerando como média de permanência do usuário. Temos o seguinte:

$$x = \frac{109444}{1.8.4} = 3420.125\ usuários.$$

Com esses números obtemos um valor de aproximadamente 3420 usuários simultâneos, um número até menor do que a carga de trabalho teórico dos usuários virtuais do teste, que representa aproximadamente 5000 usuários reais simultâneos. Acrescenta-se a isto a consideração de que um site como a Amazon trabalha em ambiente distribuído com a carga distribuída por um balanceador de carga e nunca estando totalmente concentrada em um único servidor, e se considerarmos que o script manterá um comportamento semelhante se distribuído em servidores iguais, podemos dividir o tempo de resposta do script pelo número de servidores e obter um valor para o mesmo em ambientes distribuídos, assim com 2 servidores o tempo máximo de resposta seria de 6 segundos e com 3 servidores, seria de 4 segundos, gerando pouco impacto no tempo de resposta médio dos sites de venda.

6 CONSIDERAÇÕES FINAIS

O algoritmo de Naive Bayes faz uso intensivo do processador, o que, devido às configurações da máquina utilizada, pode ter limitado o algoritmo de oferecer resultados ainda melhores. Por isso propõe-se que em testes futuros seja utilizado um servidor dedicado com maior poder de processamento e em um ambiente distribuído, com balanceador de carga.

REFERÊNCIAS

BURKE, Robin - ***Entree Chicago Recommendation Data Data Set*** -2000. https://archive.ics.uci.edu/ml/datasets/Entree+Chicago+Recommendation+Data - 24 de Maio. 2018.

CARRILHO Junior – ***Desenvolvimento de uma metodologia para mineração de textos*** – dissertação de mestrado – 2007. http://www.lambda.maxwell.ele.puc-rio.br/11675/11675_5.PDF - acesso em: 24 de Maio. 2018.

CHAN, Tony F.; Golub, Gene H.; LeVeque, Randall J. – ***Updating Formulae and a Pairwise Algorithm for Computing Sample Variances*** – 1979 http://i.stanford.edu/pub/cstr/reports/cs/tr/79/773/CS-TR-79-773.pdf - acesso em: 29 de Maio. 2018.

CHICARELLI, Ronaldo – ***Machine Learning – Naive Bayes*** - 2017 http://matera.com/br/2017/11/17/machine-learning-naive-bayes/ - acesso em: 26 de Abr. 2018.

COMEIRO FERREIRA, Fernando, APARECIDA DE OLIVEIRA, Adicinéia – Website ***Reccommende Systems As Prescriptive Determiners in the Decision Making Process*** - acesso em: 23 de Abr. 2018.
http://www.jistem.fea.usp.br/index.php/jistem/article/view/10.4301%252FS1807 17752012000200008/313

JUNIOR, Peter Jandl. ***Java Guia do programador.*** 3 ed. São Paulo: Novatec, 2015.

MONTANER, Miguel; LÓPEZ, Beatriz; LLUÍZ DE LA ROSA, Josep - ***A Taxonomy of Recommender Agents on the Internet*** – 2003.
http://ccc.inaoep.mx/~villasen/bib/agentesRecomendadores.pdf - acesso em: 24 de Maio. 2018.

PERKOVIC, Ljubomir. ***Introdução À Computação Usando Python - Um Foco No Desenvolvimento de Aplicações.*** 1 ed. São Paulo: LTC, 2016.

MACHADO, Fábio P.; PAULA, Gilberto A. – ***Distribuição Normal*** – 2016.
https://www.ime.usp.br/~giapaula/Aula%207-Distribui%E7%E3o%20Normal-MAE0219.pdf - 24 de Maio. 2018.

MACHMETRICS - ***MachMetrics Speed*** -2018
https://www.machmetrics.com/speed-blog/average-page-load-times-websites-2018/- acesso em: 29 de Maio. 2018.

RAJKUMAR, Asha, REENA, G. Sophia - ***Diagnosis Of Heart Disease Using Datamining Algorithm*** - 2010
https://computerresearch.org/index.php/computer/article/view/1028/1026 - acesso em: 23 de Abr. 2018.

RICCI, Francesco, ROCACH, Lior, SHAPIRA, Bracha - ***Introduction to Recommender Systems Handbook*** - 2011
http://www.inf.unibz.it/~ricci/papers/intro-rec-sys-handbook.pdf - acesso em: 26 de Abr. 2018.

SAS, - ***Qual A Importância Do Big Data*** - 2015
https://www.sas.com/pt_br/insights/big-data/what-is-big-data.html#importancia - acesso em: 23 de Abr. 2018.

STATISTA - ***The Statistics Portal*** - 2017
https://www.statista.com/statistics/271450/monthly-unique-visitors-to-us-retail-websites/ - acesso em: 29 de Maio. 2018.

WEBB, Geoffrey, BOUGHTON, Janice; WANG, Zhihai - ***Not So Naive Bayes: Aggregating One-Dependence Estimators*** - 2005
https://link.springer.com/content/pdf/10.1007%2Fs10994-005-4258-6.pdf - acesso em: 23 de Abr. 2018.