RESUMO

O Online Learning é um modelo de Machine Learning que nasce da necessidade de sistemas inteligentes que dispensem a atualização do código de aplicações clientes para o aperfeiçoamento do resultado que essas proporcionam.  
  
Neste trabalho foi demonstrado o desenvolvimento de um software de Marketing Online Inteligente que utiliza de ferramentas de I.A, como o Machine Learning e o Online Learning, de forma a aperfeiçoar o resultado de features como o apontamento de produtos com melhor possibilidade de aquisição sem a necessidade da atualização das aplicações utilizadas pelos usuários.

Nota-se que o software desenvolvido assegura bom funcionamento, de forma a prover o incremento dinâmico da base de dados utilizada bem como a implementação de modelos computacionais aperfeiçoados baseados na mesma. Nota-se também a ausência de travamentos mesmo sob múltiplas conexões.

Deve-se enfatizar, porém, que o software apresentado neste trabalho tem cunho acadêmico, não se mostrando competitivo ou robusto o suficiente para o uso comercial. Todavia, as tecnologias utilizadas bem como o conhecimento adquirido pode ser utilizado para esse fim.

1 INTRODUÇÃO

Em meio a expansão tecnológica a sociedade é imersa na era digital e, em pouco tempo, a rede interligada de computadores é difundida à comercialização mundial. Assim, aos poucos somos introduzidos ao mundo inteligente: computadores, dispositivos móveis, eletrodomésticos e páginas na web embebidos em inteligência artificial que auxiliam no manuseio diário, proporcionam maior praticidade em tarefas corriqueiras e trazem maior eficácia a tarefas comuns e a tarefas complexas.

A Inteligência Artificial, então, torna-se parte do cotidiano social de tal forma a se fazer absolutamente necessária à comodidade e, ainda assim, imperceptível ao usuário; seja por propostas de correção de digitação ao utilizar do teclado virtual, seja na melhor eficiência das pesquisas de um buscador online ou, ainda, na eficácia de propagandas dispostas em determinado website.

O aprendizado online surge, então, como uma vertente da Inteligência Artificial, contido nas definições de Machine Learning. Deve-se enfatizar, assim, que ML pauta-se em um sistema Inteligente que possibilita a um computador executar tarefas comumente associadas a seres vivos dotados de inteligência: o aprendizado.

Dessa maneira, pode-se definir o aprendizado Online por um sistema de Machine Learning utilizado por diversos dispositivos digitais que se comunicam por uma rede, de forma a possibilitar o aprendizado com base na modelagem de problemas que tem por característica o fluxo massivo e contínuo de dados. De forma sucinta, Aprendizado Online pauta-se em softwares centralizados em uma rede capazes de processar dados recém chegados, aprende-los e, com isso, gerar melhores respostas baseadas nas informações recebidas ou, ainda, aprendizado online pauta-se em sistemas e/ou máquinas online que fornecem serviços a diversos clientes e são capazes de aprender com os dados de entrada.

Neste trabalho, o tema Aprendizado Online é abordado através da implementação de um sistema computacional voltado ao Marketing Digital Inteligente, sendo capaz de computar entradas, processa-las, agregar dados à base de conhecimento, reestruturar modelos computacionais e sugerir produtos baseados no histórico de interesses de um indivíduo.

* 1. OBJETIVO

O objetivo deste trabalho pauta-se na implementação de um software voltado ao marketing digita.

O software em questão deverá ser capaz de oferecer sugestões de produtos baseado na sequência de interesses de consumo de um usuário que navega pela rede interligada de computadores.

Para isso, faz-se necessário a existência de uma base de dados munida de produtos e consumidores. Então, a base em questão deverá ser processada de forma a dar origem a modelos computacionais baseados em regra de associação. Desta forma, o computador que executa o software aqui citado tornar-se-á capaz de “compreender” padrões de interesse e, então, sugerir produtos, de forma mais eficaz, ao indivíduo conectado ao sistema.

* 1. JUSTIFICATIVA

O intuito que deu origem a implementação do trabalho aqui realizado parte da oportunidade de tornar mais eficaz o marketing digital online e, assim, incrementar a probabilidade, em relação ao sistema randômico convencional, de um usuário aderir a uma propaganda sugerida por determinado website ou sistema.

* 1. Estrutura do trabalho

Este documento encontra-se organizado da seguinte maneira: No capítulo 1 é apresentada a introdução do mesmo, bem como os motivos de implementação e justificativas.

No capítulo 2 são apresentadas as tecnologias embebidas no projeto em questão. Além disso, são explicados conceitos como Inteligência Artificial, Machine Learning e Online Learning.

No capítulo 3 pauta-se no desenvolvimento da aplicação proposta. Para tal, é apresentado o guia de instalação das ferramentas necessárias, fluxogramas do projeto e a implementação dos mesmos, munidos das tecnologias citadas previamente.

Testes e resultados são apresentados ao longo do capítulo 4 e, considerações finais são dispostas no capítulo 5.

No capítulo 6 encontram-se as referências utilizadas ao longo do desenvolvimento do trabalho aqui documentado.

* 1. PROPOSTA

A proposta o presente trabalho pauta-se no desenvolvimento de um software que possua como principal característica o aprendizado online. Além disso, o programa em questão deverá suportar conexões simultâneas, bem como executar sugestões, inteligentes, de produtos às mesmas.

Para isso, a base de dados deverá conter os devidos produtos a serem ofertados, de forma a possibilitar ao software a criação de modelos computacionais que serão utilizados, posteriormente, na feature de decisões.

O programa em questão será implementado na linguagem Python e a base de dados utilizada refere-se a uma loja inglesa no período de 2010 a 2011 e pode ser encontrada nos arquivos da UCI.

Deve-se enfatizar que para a simulação de funcionamento do serviço criado, o software implementado nesta documentação contará com um módulo “cliente” responsável por simular o computador de um usuário, conectado a mesma rede de computadores do módulo “servidor”, que será responsável pelo manuseio da base de dados citada, bem como processamento dos dados extraídos da mesma.

1. TECNOLOGIAS E FERRAMENTAS

Neste capítulo serão descritas as ferramentas utilizadas para a implementação do software descrito no presente documento, bem como motivações para o uso das mesmas.

Além disso, será apresentado o fluxo de dados do projeto em questão, bem como explanação dos conceitos utilizados.

* 1. WINDOWS O.S

Historicamente, a primeira versão, extremamente simplificada, do Microsoft Windows nasce em 1985, batizado de MS-DOS e parcialmente baseado nos conceitos de funcionamento do Apple Macintosh System.

As versões subsequentes introduziram grandes novidades funcionais, incluindo a interface gráfica, cujo funcionamento se dava através de janelas (Windows), explorador de arquivos nativo, gerenciador de programas e, futuramente, interface mais dinâmica.

Com o lançamento do Windows XP, em 2001, a gigante americana abandona o uso do antigo Kernel do Windows 95, de forma a fornecer avanços significativos na usabilidade, performance e gerenciamento de memória.

Em 2009 é lançado o Windows 7, com ganhos notáveis de performance e requisitos de sistema modestos, de forma a abocanhar parte considerável do mercado de computadores pessoais mundial.

Em 2015 surge o Windows 10, utilizado para a implementação, bem como testes, do software aqui apresentado; Selecionado por compor, segundo a base de dados do site NetMarketShare, mais de 30% dos computadores pessoais ao redor do globo no ano de 2017.

* 1. REDES

Rede de computadores pauta-se no conjunto de dois ou mais dispositivos digitais interligados por um link de dados guiado por um conjunto de regras (protocolo de rede) para troca de informações.

Os dispositivos existentes em uma rede de computadores responsáveis pelo roteamento e término de dados denominam-se “nós de redes” ou pontos de conexão.

Os pontos de conexão do software implementado pautam-se no módulo servidor e, pelos diversos módulos clientes que poderão ser conectados ao módulo servidor.

* 1. BASE DE DADOS

A base de dados utilizada é fornecida pelo UCI - Machine Learning Repository e refere-se a dados de uma loja online britânica no período de 2010 a 2011, fornecida no formato XLSX.

De forma a evitar possíveis complicações na conversão do arquivo XLSX para a extensão CSV e, então, na importação para um banco de dados convencional (como Postgres, Oracle ou MySql), a base foi mantida em seu padrão XSLX.

O XLSX em questão é composto por 8 colunas representadas por:

* InvoiceNumber
* StockCode
* Description
* Quantity
* InvoiceDate
* UnitPrice
* CustomerID
* Country

Cada linha na base é estruturada por um exclusivo produto adquirido por determinado consumidor.

Enfatiza-se, ainda, que a única coluna que interessa ao projeto em questão é ‘description’ que executará papel essencial na diferenciação de produtos bem como execução da regra de associação (ver 2.5.3.1)

* 1. PYTHON 3.8

A Linguagem Python surge em 1989, implementada pelo Holandes Guido Van Rossun, como sucessora a linguagem ABC.

O Python é uma linguagem de programação interpretada cuja filosofia enfatiza a legibilidade do código; Deve-se dizer que, atualmente, a linguagem Python é gerenciada pela Python Software Foundation e possui licença de código aberto: A Python Software License, compatível com a Generic Public License (GNU 2.1.1)

O python foi escolhido por se tratar de uma linguagem leve, quando comparada a outras linguagens interpretadas no mercado atual e amplamente difundida em empresas e projetos pessoais.

Além disso, trata-se de uma linguagem multi-paradigma, de forma a possibilitar a implementação no modelo orientado a objeto ou, em menor escala, no modelo funcional; o que possibilita ao programador melhor organização e legibilidade de código.

* + 1. NUMPY

O Numpy é uma framework para a linguagem Python de programação, nascida em 1995 como Numeric e, em 2006 como o conhecido Numpy.

A biblioteca open-source mencionada é implementada por Jim Hungunin e Travis Oliphant nas linguagens Python e C, de forma a adicionar suporte a Arrays e Matrizes multi-dimencionais de tamanho elevado.

* + 1. PANDAS

Pandas é uma biblioteca para a linguagem python destinada a manipulação e analise de dados. Oferece estruturas de dados úteis e ferramentas para a manipulação das mesmas; bem como tabelas numéricas e séries temporais: sequência de “data-points” indexados organizados com base no tempo discreto, ou seja, não contínuo.

Trata-se de uma framework gratuita, que atua sob a nova licença BSD (BSD-new), cujo nome deriva-se da aglutinação dos termos “Panel” e “Data” ou, “Painel” e “Dados”; termos que remetem a observação de dados sob diferentes períodos de tempo.

A biblioteca aqui citada é utilizada para, a partir da base de dados, criar-se estruturas de dados, denominadas DataFrames, responsáveis pela primeira etapa da modelagem de dados que virá a ocorrer durante a implementação do software documentado.

* + - 1. DATAFRAMES

Pandas é comumente utilizado para a execução de algoritimos de Machine Learning estruturados em dataframes, de forma a possibilitar a importação de dados em formatos recorrentes como o Microsoft xls, xlsx Comma Separated Value (CSV). Possibilita, ainda, a manipulação de dados através de operadores como groupby e join.

* + 1. XLRD

XLRD pauta-se em uma biblioteca destinada a leitura de dados e formatação de informações extraídas de arquivos Microsoft Excel (XLS ou XLSX) e é utilizado como dependência da framework Pandas.

* + 1. MLXTEND

MLXTEND é uma biblioteca destinada a linguagem python que fornece ferramentas importantes ao desenvolvimento de aplicações voltadas ao Machine Learning. Atua como módulo ajudante para a analise de dados na linguagem python e fornece módulos de algoritimos essenciais a proposta apresentada, como o Apriori e a regra de associação.

* + 1. \_Thread

Threads, de forma análoga, pautam-se em pequenos programas computacionais que atuam como um subsistema independente ao software principal, de forma a possibilitar a distribuição de tarefas.

Essas tarefas podem, então, ser executadas simultaneamente de forma síncrona ou assíncrona; podendo trocar informações e dados entre si, bem como compartilhar recursos do sistema operacional em que se encontram.

Possibilita, também, a criação de “módulos” assíncronos, que tem papel vital no software em questão. As threads, no projeto em questão, são baseadas na biblioteca \_Thred da linguagem python3 e são responsáveis por possibilitar o handle de múltiplas conexões por parte do servidor, bem como segregar a classe “server” em módulos destinados a rede: conexões, buffer de dados e respostas; e módulos destinados ao processamento de dados: gerar modelos computacionais, buscar dados em tabelas e, sugerir produtos.

Além disso, as threads são responsáveis por evitar o congelamento da U.I, no módulo cliente, quando um comando é enviado pela rede.

* + 1. SOCKET

Um soquete de rede trata-se do ponto final de determinado fluxo de comunicação entre sistemas digitais sob uma rede de computadores.

A biblioteca sockets trata-se de um módulo da linguagem python que faz uso da API (Application Programming Interface) de redes computacionais fornecida pelo sistema operacional que, por sua vez, permite que o script codificado tenha acesso bem como usufrua dos soquetes de internet fornecidos pelo hardware da máquina utilizada.

O protocolo de comunicação utilizado é o TCP/IP utilizando da família de endereços AF\_INET.

* + - 1. AF\_INET

AF\_INET trata-se de uma família de endereços lógicos utilizados de forma a designar o tipo de endereço que um socket irá se comunicar. Ao utilizar da família AF\_INET, o socket aberto receberá conexões baseadas no protocolo de redes IPv4.

Existem inúmeras opções de protocolos como o aqui citado, bem como AF\_UNIX, AF\_IPX , AF\_IRDA e AF\_BLUETOOTH.

A opção pelo AF\_INET se deu pela segurança apresentada pelo protocolo, bem como ampla gama de documentações. Todavia, o protocolo IPv6 (AF\_INET6) também poderia ser utilizado.

* + - 1. TCP/IP

O protocolo de dados de rede utilizado para a implementação da comunicação entre o módulo cliente e o servidor é o TCP/IP. Trata-se do principal protocolo de envio e recebimento de dados através da internet.

TCP define-se por Transmission Control Protocol, ou seja, o TCP IP é um protocolo de controle de transmissão de dados baseado em endereços IP.

* + 1. OPENPYXL

Openpyxl é uma biblioteca nascida do PHPExcel que atua como opção simplificada do módulo de leitura de planilhas da framework pandas. Todavia, trata-se de uma opção simples e veloz, destinada ao uso exclusivo de escrita de dados no documento excel utilizado como base de dados.

* + 1. PYQT5

O Qt é uma framework multiplataforma criada originalmente para a linguagem C++. Nascido na empresa TrollTech, o Qt é mantido pelo QtProject, de forma a ser utilizado por empresas como a Google, NASA, Disney, volvo, etc.

No Python, Qt existe sob a frameworks PyQt e PySide; de modo a possibilitar o uso de licenças comerciais, como o QtEnteprise; Licença LGPL e licença GPL v3.

O Qt é utilizado em sua quinta versão no módulo cliente, de forma a responsabilizar-se pela interface gráfica que simulará o ambiente de marketing de um site ou aplicação genéricos.

* 1. INTELIGÊNCIA ARTIFICIAL

Oficialmente, a inteligência artificial nasce em 1956, no colégio inglês de Dartmouth, alguns anos depois de Alan Turin, matemático criptoanalista britânico, propor, pela primeira vez, a ideia de ‘máquina pensante’. Todavia, o desenvolvimento lento da área, agregado a escassez de resultados, acarretou no “Inverno da Inteligência Artificial”, na década de 60.

Então, em 1993, o MIT Cog Project apresenta o Dynamic Analysis and Replanning Tool (DART): robô humanoide responsável por reestabelecer a credibilidade da inteligência artificial perante o mundo, de forma a criar os alicerces em que, em 1997, o IBM Deep Blue, ao derrotar campeão mundial de xadrez Kasparov, se encarregaria de cravar a bandeira da inteligência artificial

Todavia, apesar dos constantes avanços tecnológicos ocorridos ao longo das últimas décadas de pesquisa acadêmica, a inteligência artificial vem a ser reconhecida como mudança de paradigma no quarto trimestre do ano de 2012, como demonstra o gráfico criado por Francesco Corea, líder de pesquisas na UCLA (University of California – Los Angeles).

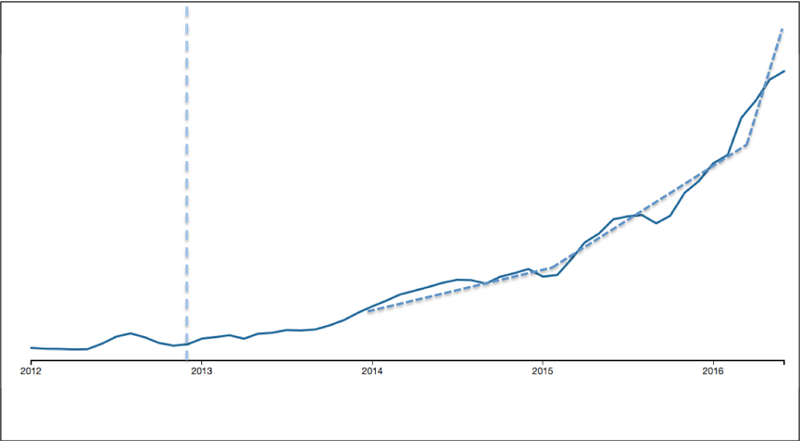


Figura 1 – Tendência da inteligência artificial no período de 2012 a 2016

Atualmente, a inteligência artificial pode ser segregada em diferentes níveis de habilidade cognitiva. Uma I.A é considerada ‘estreita’ quando performa atividades, ou funções específicas, que resultam em soluções cujo resultado de uma ação se faz superior, quando em comparação, as soluções empregadas pelo ser humano até então.

Além disso, uma I.A é considerada ‘geral’ quando se faz capaz de performar uma atividade qualquer coma mesma média de acertos, ou eficácia, de um ser humano.

Por fim, considera-se ‘ativa’ quando as respostas geradas pelo algorítimo inteligente ultrapassa o nível performático humano em diversas atividades.

Enfatiza-se, então, que o algorítimo contido no trabalho em questão trata-se de um algorítimo de Machine Learning enquadrado como Inteligência Artificial Estreita, de forma a performar soluções ótimas em atividade específica: sugerir produtos.

* + 1. Machine Learning

Machine Learning trata-se de um algorítimo, destinado a computadores, capaz de coletar dados, processa-los, ‘aprende-los’ e, então, tomar decisões ou realizar determinações.

Pode-se dizer, então, que o trabalho aqui documentado utiliza de Machine Learning, de forma a coletar dados da base UCI, processa-los, gerar modelos relacionais baseados no ‘aprendizado’ e, então, gerar uma saída.

* + 1. Deep Learning

Apesar de não implementado neste trabalho, deve-se enfatizar as diferenças existentes entre o Machine Learning e o Deep Learning. Como demonstra a figura 2, Deep Learning trata-se de uma técnica de implementação para o aprendizado de máquina.

De maneira simplificada, algorítimos de ML não trabalham com dados básicos, como inteiros, pontos flutuantes e booleanos.

Em contrapartida, algorítimos de DL, geralmente, trabalham com dados RAW, ou seja, utilizam dados que melhor representam o natural: Imagens, vídeos e áudio.

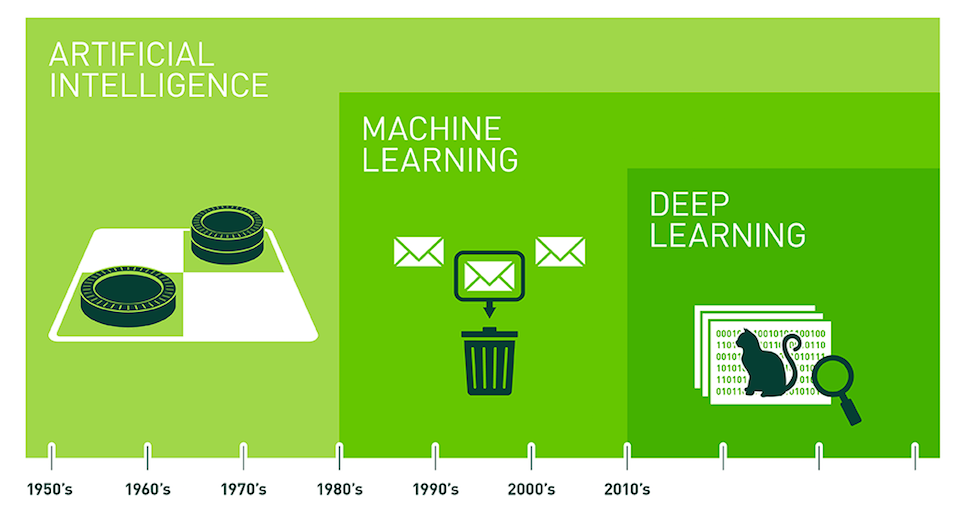


Figura 2 – Inteligência artificial, Machine Learning e Deep Learning

Enfatiza-se, também, que o aprendizado profundo caracteriza-se pela capacidade de aprendizado baseado nos momentos de falha e sucesso do próprio algorítimo.

* + 1. Data-mining

A mineração de dados trata-se do processo de explorar grandes quantidades de dados à procura de padrões consistentes. Pode-se dizer que cria-se novos conjuntos de dados baseados em regra de associação ou sequencia temporal.

Enfatiza-se que o data mining forma-se pelo conjunto de algorítimos de aprendizado de máquina, ou classificação de dados, e estatística.

O Data Mining é utilizado na implementação do software abordado no presente documento junto ao algoritimo Apriori e a Regra de Associação.

* + - 1. Regra de Associação

Trata-se de um padrão lógico que engloba algorítimos como o Apriori, Partition, Eclat e FP-Growth.

Pode-se dizer que as regras de associação tem como premissa a busca por associações de elementos em uma tabela de dados que, tenha por resultado, um ou mais elementos dentro de uma mesma base. De forma a transpor o conhecimento de forma iterativa e ágil, considere a tabela 1, contemplada por dados fictícios de uma padaria fictícia.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Transação | Leite | Pão | Manteiga | Cerveja |
| 1 | 1 | 1 | 0 | 0 |
| 2 | 0 | 1 | 1 | 0 |
| 3 | 1 | 1 | 1 | 0 |

Tabela 1 – Dados para regra de associação

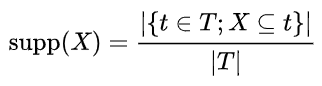
Pode-se perceber, através da tabela disponibilizada anteriormente, que o cliente que adquire leite sempre irá adquirir pão. Todavia, um cliente que adquire pão, pode ou não adquirir leite e manteiga. Além disso, percebe-se que cerveja nunca é consumida.

De forma a possibilitar um sistema digital a execução de análises facilmente executadas por humanos, métricas se fazem necessárias à criação de limiares mínimos de confiança.

* + - * 1. Suporte

Trata-se da indicação da frequência que um item aparece em uma base de dados.

Dessa maneira, o suporte de X em relação a T pode ser definido como a proporção de transações t no conjunto de dados que contém o conjunto de itens X.



De acordo com a tabela exemplo fornecida, o conjunto X composto por {pão, manteiga} tem o suporte definido por 2/3 = 0,66. Ou seja, ocorre em 66% das transações encontradas na tabela de dados.

* + - * 1. Confiança

Confiança é a indicação do quão frenquente uma regra encontrou-se verdadeira e pode ser definida pela função explicita na figura 4.

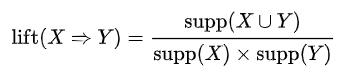


Dessa maneira, de acordo com a tabela exemplo fornecida, a regra {pão, manteiga} -> {leite} tem índice de confiança 0,66/0,66 = 1

* + - * 1. LIFT

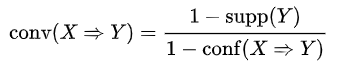
Trata-se de uma medida de desempenho quanto a um modelo de dados na previsão ou classificação de elementos de determinada base de dados. Pode-se dizer que um modelo de dados é considerado bom quando a resposta apresentada pelo cálculo de LIFT é superior a média apresentada pela base de dados.

O LIFT pode ser calculado pela função explicitada na imagem 7.



* + - * 1. Convicção

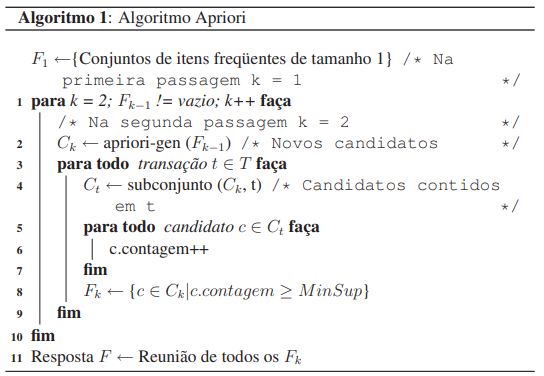
A convicção de um modelo pode ser definido como a frequência média esperada para que um elemento de uma tabela ocorra na mesma e pode ser explicito pela função descrita na figura 8.



* + - 1. APRIORI

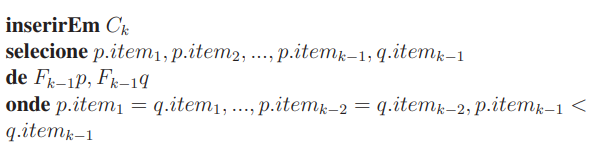
Apriori é um algorítimo, representado pela figura abaixo, destinado a mineração de dados baseado em regras de associação. O algorítimo emprega busca em profundidade e gera conjuntos de itens candidatos de K elementos a partir de conjuntos de itens de K – 1 elementos.

Então, os padrões menos frequentes ou, de frequência nula, são eliminados e, então, os itens mais frequentes são obtidos por meio do conjunto de itens candidatos.

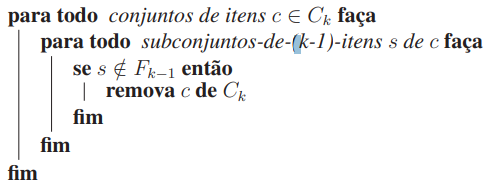


De forma a explanar o algoritimo acima inferido, Fk representa o conjunto de itens frequentes de tamanho k que atende ao valor de suporte mínimo estabelecido. Além disso, Ck representa o conjunto de itens candidatos de tamanho k.

Deve-se enfatizar que o algoritimo em questão faz uso de sub-rotinas, bem como a apriori-gen, responsável por gerar o conjunto de itens candidatos. A função em questão tem por argumentos o conjunto de k-1 itens e retorna o conjunto de todos os conjuntos de k itens, como demonstra a figura abaixo.



E, enfim, alimina-se todo o conjunto de itens c que percença a Ck caso um conjunto de k-1 itens de c não se encontre em Fk, como demonstra o algorítimo a seguir.



* + - 1. Online Learning

O Online Learning é uma técnica frequentemente empregada em projetos baseados em Machine Learning em que treinar um algoritimo sob uma base de dados é computacionalmente inviável. O OL também pode ser utilizado em situações em que a adaptação dinâmica a novos padrões em uma base de dados se faça necessária.

No projeto em questão, o Online Learning é implementado de maneira que os modelos computacionais treinados se encontram centralizados em uma máquina que pode ser acessada por N módulos clientes.

Dessa maneira, possibilita-se o ganho de conhecimento (dados) constante ao algorítimo implementado, que atualiza o modelo relacional de forma frequente.

Além disso, os clientes conectados tem a possibilidade de acessar a base de conhecimento atualizada simultaneamente, o que isenta o projeto de problemas de assincronia de conhecimento no módulo cliente.

1. DESENVOLVIMENTO

O presente capitulo descreve, de forma detalhada, todo o procedimento executado enquanto implementava-se o software aqui documentado. Inicia-se pela instalação das dependências necessárias ao correto funcionamento da aplicação, transcorre pelo fluxo de dados e, por fim, destrincha o funcionamento dos métodos implementados em seus respectivos módulos.

Enfatiza-se que os algoritimos apriori e regra de associação foram utilizados através a biblioteca mlxtend, da linguagem python. Além disso, o modelo computacional devidamente relacionado da base de dados OnlineRetail.xlsx pode ser encontrado ao executar a classe isolated\_apriori.py, cujos dados são aplicados a estrutura DataFrame da biblioteca Pandas.

* 1. Instalação de Ferramentas

O primeiro requisito a ser atendido para execução do código aqui documentado é a instalação da linguagem Python 3. Para isso, deve-se acessar o domínio da Python Software Foundation (python.org) e efetuar o download do instalador o python3. Enfatiza-se que qualquer build do python3 pode ser utilizada.

Assim que o download for realizado, basta seguir as etapas explicitadas pelo guia Wizard de instalação.

* + 1. PIP

PIP é o gerenciador de pacotes para a linguagem python e, pode ou não ser instalado de forma automática junto a instalação do interpretador Python.

Para verificar se o PIP se encontra instalado, basta abrir o Windows CMD ou Windows Power Shell e digitar:

Pip –version

Dessa forma, o comando de requisição de versão do gerenciador de pacotes será executado pelo sistema operacional de forma a informar a numeração da build instalada ou, então, informar que o comando é desconhecido par o sistema.

Caso o comando não seja encontrado, deve-se descarregar o arquivo ‘get-pip.py’ do domínio Python Foundation ou, caso o gerenciador de pacotes do MS-Windows se encontre instalado, executar a seguinte linha de comando no terminal:

Curl <https://bootstrap.pypa.io/get-pip.py> -o get-pip.py

Após o download do arquivo get-pip, basta executar a linha de comando:

Python get-pip.py

Assim que instalado, verifique novamente o versionamento através do comando citado anteriormente. Caso o sistema ainda não reconheça a instalação, basta adicionar a pasta PIP presente no diretório de instalação do python às variáveis de ambiente do MS Windows.

* + 1. INSTALAÇÃO DE FERRAMENTAS PYTHON

Satisfeitas as condições anteriores, todas as outras dependências exigidas pelo projeto podem ser satisfeitas através do uso do gerenciador de pacotes da linguagem utilizada. Para isso, basta executar os seguintes comandos no terminal:

* Pip install xlrd
* Pip install colorama
* Pip install openpyxl
* Pip install pandas
* Pip install numpy
* Pip install mlxtend
* Pip install pyqt5

Além disso, pode-se utilizar do pip freeze para instalação automatizada. Para isso, cria-se um arquivo nomeado requirements.txt e adiciona-se o nome das bibliotecas desejadas separadas por quebra de linha. Então, executa-se a linha de comando:

Pip freeze requirement.txt

Enfatiza-se, ainda, que os módulos \_thread e sockets são nativos a linguagem python3.

* 1. MÓDULOS

Como mencionado anteriormente, o projeto em questão é contemplado por dois módulos, de forma a simular a comunicação entre um sistema cliente e um sistema servidor. Deve-se enfatizar, porém, que cada módulo é compreendido por múltiplas classes de dados, detalhados posteriormente ao longo deste documento, desenvolvidas sob a linguagem python de programação, como mostra o fluxograma abaixo.

* + 1. MÓDULO SERVIDOR

O módulo servidor é o coração do projeto em questão; responsável por estabelecer conexões com o cliente, distribuir threads, manipular a base de dados e, sobretudo, criar modelos computacionais baseados em regra de associação que serão utilizados posteriormente.

É dividido em três classes (arquivos) principais segregadas de acordo com sua função principal. Desta maneira, tem-se:

* Server.py
* Data\_process.py
* Xls\_utils.py
  + - 1. Server.py

A classe servidor é responsável por estabelecer conexões, criar threads, decodificar os dados recebidos, realizar controle de buffer e, então, processar os dados recebidos e já decodificados, como mostra o fluxograma a seguir.

Como mencionado no tópico Database, a única informação necessária ao software de machine learning é a descrição. Todavia, para não escrever campos nulos na base de dados durante a rotina de atualização, o módulo cliente encarrega-se de agregar valores fixos aos campos não utilizados.

Dessa forma, quando uma conexão é estabelecida com o servidor, a porta gerenciada pela biblioteca socket recebe um código binário que, quando convertido pela função .decode() revela uma string estruturada de forma que dados irrelevantes são separados pelo dígito # e, o array de elementos a serem cadastrados são segregados pelo digito $, como demonstra o exemplo:

123#321#1#2020#1.20#99#france$sapato preto$cinto branco

Dessa maneira, quando extraídos os produtos da string recebida, de forma a evitar problemas com o padrão de extensão excel, abre-se conexões para escrita em instantes diferentes de conexões que são abertas destinadas a leitura. Para isso, armazena-se os produtos extraídos da string recebida em um buffer de escrita que contempla arrays de elementos que serão escritos juntos em uma única conexão.

Dessa maneira, quando escrever elementos se faz necessário, a classe servidor cria uma nova thread, através da biblioteca \_treads, destinada a escrita na base de dados.

Assim que o buffer de dados é escrito e esvaziado, a base de dados utiliza da mesma thread aberta para executar uma atualização dos modelos de dados já existentes.

Além disso, enfatiza-se que as threads são criadas de forma a possibilitar ao servidor realizar conexões enquanto se encontra sobre processamentos que consomem tempo elevado.

Caso o servidor não se encontre em rotina de escrita, após a decodificação da string recebida, no método apriori\_país, é realizada uma busca pela sequencia de elementos recebidos no modelo gerado pela classe data\_process. Caso o padrão de antecedentes seja reconhecido, o método codifica a lista de consequentes e envia ao módulo cliente correspondente. Do contrário, um comando de reset é enviado.

Enfatiza-se ainda que, por inconsistências contínuas na comparação de objetos realizada pela linguagem python, a solução aplicada foi executar a comparação de hashs. Para isso, cria-se um hash que recebe como argumento o array de antecedentes que será comparado a um hash que recebe como argumento o array de recebido via conexão socket.

* + - 1. Data\_process.py

A classe em questão é instanciada pela classe servidor e utilizada para gerar modelos preditivos baseados na base de dados XLSX Online\_retails.

Para isso, utiliza-se da biblioteca Pandas para carregar a base de dados em uma variável chamada data. Então, utiliza-se da rotina “clean\_data” criada para efetuar a remoção de caracteres indesejados, bem como excluir linhas munidas de cadastros nulos ou cadastros falhos.

Então, segrega-se os dados carregados de acordo com o país do consumidor e, por fim, os codifica de forma a padronizar os dados carregados pela base de forma a possibilitar seu processamento.

Por fim, a classe em questão utiliza da biblioteca MLXTEND para aplicar os algorítimos apriori e association rule.

* + - 1. XLS\_UTILS.py

A Classe XLS\_UTILS, bem como o nome explicita, fornece utilidades para a manipulação da base de dados xlsx. Em principal, a rotina de escrita.

* + 1. CLIENTE

O módulo cliente é responsável por simular o campo de propagandas de determinado site ou aplicação. Baseia-se em uma GUI simples que contém um botão munido da string que representa um produto qualquer na base de dados.

É construído por duas classes básicas: main.py e data.py.

* + - 1. MAIN

A classe main é responsável por implementar a GUI utilizada pelo usuário, bem como realizar conexões com o servidor, codificar os dados a serem enviados e recebe-los.

De forma a construir a interface gráfica de maneira dinâmica, que possibilita a alteração de elementos presentes na tela quando necessário, utiliza-se da biblioteca PyQT em paralelo a biblioteca threads.

Ao iniciar, a classe main realiza o carregamento de toda a interface gráfica junto ao produto de maior interesse na região do usuário. Então, um botão é criado de forma a representar uma aba de propaganda em um site ou sistema.

O botão em questão, quando clicado, aciona um slot que, por sua vez, inicia uma thead responsável pelo serviço de comunicação do módulo cliente com o módulo servidor, como demonstra o trecho de código a seguir.

Enquanto o módulo servidor necessita de threads para que possa realizar os devidos processamentos necessários enquanto se mantém apto a receber novas conexões, o módulo cliente necessita de threads para que a GUI se mantenha funcional enquanto aguarda uma resposta do servidor.

Enfatiza-se, também, que conexões são abertas exclusivamente quando necessárias e, então, são fechadas de forma a prevenir a ‘quebra’ de portas lógicas do sistema operacional.

Além disso, a classe main é responsável por codificar as informações de usuário que serão enviadas posteriormente ao servidor, de forma a concatenar o dígito ‘$’ a dados necessários ao projeto e ‘#’ aos demais., como demonstra o trecho de código a seguir.

Existe, ainda, uma thread responsável pelo ‘timeout’ de produtos quando o mesmo não se faz interessante o usuário. Dessa maneira, o timeout é atingido, utiliza-se a rotina ‘get\_product’ da classe data.py para a busca de um produto ainda não ofertado.

* + - 1. DATA

A classe data é utilizada como suporte para a classe main. Contém informações como o endereço do servidor, porta de acesso, endereçamentos importantes à GUI, valores estáticos utilizados para cadastro na base e, uma lista de produtos mais utilizados.

A classe exemplificada utiliza-se, ainda, de um sistema de buffer implementado em uma estrutura do tipo set(), que coíbe a repetição de elementos. Dessa forma, a classe passa a ter controle do elementos que foram, ou não, ofertados ao possível consumidor.

Então, no momento em que a classe main executa a função de timeout, a classe data é acionada e um novo produto é retornado conforme demonstra a implementação abaixo.

De forma a descrever o funcionamento do código explicitado, a cada requisição por um novo produto, a lista de produtos é percorrida até que o loop encontre um elemento que ainda não se encontre no buffer de dados representado pela variável \*\_PRODUCTS.

Então, adiciona-se o novo elemento ao buffer e, retorna-se o ‘produto’ à classe main.

Caso não seja encontrado nenhum produto que não exista no buffer, ou seja, ainda não tenha sido ofertado, o buffer é zerado e retorna-se o primeiro elemento para a classe main.

1. RESULTADOS

De forma a facilitar a manipulação de entradas, gerou-se um script implementado na linguagem python que se encarrega de criar modelos computacionais embasados na mesma base de dados utilizada pelo projeto em questão.

Enfatiza-se, ainda, que o algoritimo utilizado para gerar os modelos é o mesmo utilizado no código fonte do projeto.

1. Condiserações Finais

A área de atuação de sistemas inteligentes tem se expandido e, ao longo dos anos, se mostrado cada vez mais importante à evolução de sistemas digitais; sejam esses sistemas assistentes pessoais, teclados digitais, predições meteorológicas, análise de imagens, sistemas de busca inteligentes ou marketing eficiente.

Neste trabalho, buscou-se demonstrar, de forma suscinta, o uso da inteligência artificial em um sistema de aprendizado online. Assim, apresentou-se o conceito por de trás do funcionamento do sistema em questão, bem como ferramentas gratuitas que tem papel crucial em facilitar o desenvolvimento desse e de outros tipos de sistema.

Considera-se então que os objetivos estabelecidos no inicio deste trabalho foram atingidos, vez que o software criado se mostrou eficiente nas sugestões efetuadas. Enfatiza-se, ainda, que o uso da linguagem Python foi um grande acerto, vez que possibilitou o uso de excelentes ferramentas como a biblioteca MLXTEND e PANDAS.

Uma boa sugestão para trabalhos futuros é o uso de databases relacionais no lugar de de um arquivo XLSX fixo. O uso de uma base de dados relacional seria responsável por granhos de desemprenho gerais; em especial, durante rotinas de carregamento de dados.

1. Referencias