ISNN: 2358-6052

PROPOSTA DE UM ALGORITMO DE CLASSIFICAÇÃO INCREMENTAL E ON-LINE UTILIZANDO CONCEITOS DO K-MEANS E DO KNN

Bruno Alberto Soares Oliveira [1]; Servílio Souza de Assis [2]; Breno Costa Dolabela Dias [3]; Thais Macela de Lira Menegaldi [5]; Frederico Gadelha Guimarães [5];

1 Programa de Pós-Graduação em Engenharia Elétrica, UFMG, Belo Horizonte – MG; brunoalbertobambui@ufmg.br
2 Programa de Pós-Graduação em Engenharia Elétrica, UFMG, Belo Horizonte – MG
3 Programa de Pós-Graduação em Engenharia Elétrica, UFMG, Belo Horizonte – MG
4 Engenharia de Sistemas, UFMG, Belo Horizonte – MG
5 Machine Intelligence and Data Science (MINDS) Laboratory, UFMG, Belo Horizonte – MG; fredericoguimaraes@ufmg.br

RESUMO

Acredita-se que aprendizado de máquina é a ciência dos computadores que são programados com o objetivo de que eles possam aprender com os dados. A aprendizagem incremental é um método no qual os dados de entrada são continuamente usados para ampliar o conhecimento do modelo existente, ou seja, para treinar mais o modelo. O aprendizado de máquina on-line é um método em que os dados ficam disponíveis em uma ordem sequencial e é usado para atualizar o melhor preditor de dados futuros em cada etapa. No aprendizado supervisionado, os algoritmos aprendem com dados rotulados. Depois de entender os dados, o algoritmo determina qual rótulo deve ser aplicado aos novos dados com base no padrão e associa os padrões aos novos dados não rotulados. Dois algoritmos comumente usados na área do aprendizado de máquina são o K-means e o K-Nearest Neighbors. O k-Means Clustering é um algoritmo de aprendizado não supervisionado que é usado para agrupamento, enquanto o KNN é um algoritmo de aprendizado supervisionado usado para classificação. Baseado neste contexto, o objetivo deste trabalho é implementar um código que utiliza os conceitos dos algoritmos de aprendizado de máquina citados para classificar dados. Toda implementação do algoritmo foi feita em nuvem, no ambiente de desenvolvimento Jupyter Notebook da Google, utilizando a linguagem de programação Python. Foram usadas algumas bibliotecas como a Numpy, Pandas, Sklearn, entre outras. A implementação proposta foi avaliada em datasets sintéticos e reais, como forma de validação do método proposto. As bases de dados sintéticas escolhidas são utilizadas como estado de arte nesse tipo de problema, sendo algumas bastante conhecidas, como a da espiral e a que forma duas meias-luas. Após os experimentos realizados, pôde-se verificar que o método proposto obteve resultados semelhantes ao se comparar com os resultados do estado da arte sob esses conjuntos de dados.

INTRODUÇÃO:

A pesquisa em aprendizado de máquina procura desenvolver sistemas de computadores que melhorem automaticamente seu desempenho por meio da experiência. Esse estudo é capaz de produzir sistemas computacionais, como robôs, que aprendem a operar em novos ambientes, sistemas de compreensão de fala que se adaptam automaticamente a novos falantes e novas condições ambientais, sistemas de consultoria baseados em conhecimento que colaboram com especialistas humanos para resolver problemas difíceis e adquirir novas táticas, observando a contribuição do humano para a solução do problema, ou até mesmo programas de computador que possam adquirir a capacidade de resolver problemas de física ou cálculo.

O objetivo da pesquisa em aprendizado de máquina é produzir uma tecnologia de capacitação independente de domínio para uma ampla gama de aplicativos de computador. Um avanço nessa área pode ter um impacto significativo em um amplo espectro de aplicativos tão diversos como na robótica, design assistido por computador, bancos de dados inteligentes e sistemas de consultoria baseados em conhecimento. Essas aplicações são cada vez mais baseadas em conhecimento, isto é, dependem de um grande número de fatos específicos sobre o domínio da tarefa. O aprendizado de máquina oferece o potencial para remover o gargalo da aquisição de conhecimento que limita o desempenho e aumenta os custos de desenvolvimento para esses sistemas (BISHOP, 2006).

Os algoritmos de aprendizado de máquina, que têm como principal objetivo resolver problemas de classificação baseia-se no processo de prever a classe de determinados pontos de dados. Às vezes, as classes são chamadas de destinos, rótulos ou categorias. A modelagem preditiva de classificação é a tarefa de aproximar uma função de mapeamento (f) de variáveis de entrada (X) para variáveis de saídas discretas (y). A classificação pertence à categoria de aprendizado supervisionado, em que os alvos também

ISNN: 2358-6052

forneceram os dados de entrada. Existem muitas aplicações na classificação em muitos domínios, como aprovação de crédito, diagnóstico médico, marketing de destino etc (PEDREGOSA, 2011).

Há muitos algoritmos de classificação disponíveis, mas não é possível concluir qual deles é superior ao outro. Depende da aplicação e natureza do conjunto de dados disponível. QUINLAN (2014) deu como exemplo o seguinte caso: se as classes são linearmente separáveis, os classificadores lineares como a regressão logística e o discriminante linear de Fisher podem superar modelos sofisticados e vice-versa.

A aprendizagem incremental é um método de aprendizado de máquina em que os dados de entrada são continuamente usados para ampliar o conhecimento do modelo existente, ou seja, para treinar mais o modelo. Ele representa uma técnica dinâmica de aprendizado supervisionado e não supervisionado que pode ser aplicada quando os dados de treinamento se tornam disponíveis gradualmente ao longo do tempo ou seu tamanho está fora dos limites de memória do sistema. Algoritmos que podem facilitar o aprendizado incremental são conhecidos como algoritmos incrementais de aprendizado de máquina (ROSS et al., 2008).

O objetivo da aprendizagem incremental é que o modelo a ser aprendido se adapte aos novos dados sem esquecer o conhecimento existente, não reciclando o modelo. Alguns algoritmos incrementais incorporaram algum parâmetro ou suposição que controla a relevância de dados antigos, enquanto outros, chamados de algoritmos incremental estáveis, aprendem representações dos dados de treinamento que nem sequer são parcialmente esquecidos ao longo do tempo (ROSS et al., 2008).

Algoritmos incrementais são frequentemente aplicados a fluxos de dados ou big data, abordando questões de disponibilidade de dados e escassez de recursos, respectivamente. Previsão de tendência de estoque e perfis de usuário são alguns exemplos de fluxos de dados em que novos dados se tornam continuamente disponíveis. A aplicação do aprendizado incremental ao Big Data tem como objetivo produzir tempos mais rápidos de classificação ou previsão (CAUWENBERGHS e POGGIO, 2001).

De acordo com HAZAN et al. (2016), o aprendizado de máquina on-line é um método em que os dados ficam disponíveis em uma ordem sequencial e é usado para atualizar o melhor preditor de dados futuros em cada etapa, em oposição a técnicas de aprendizado em lote que geram o melhor preditor em todo o conjunto de dados de treinamento de uma só vez. O aprendizado on-line é uma técnica comum usada em áreas de aprendizado de máquina, que é computacionalmente inviável treinar todo o conjunto de dados, exigindo a necessidade de algoritmos fora do núcleo.

O aprendizado on-line é usado em situações em que é necessário que o algoritmo se adapte dinamicamente a novos padrões nos dados, e quando os dados em si são gerados em função do tempo, por exemplo, predição do preço das ações. Os algoritmos de aprendizado on-line podem ser propensos a interferência catastrófica, um problema que pode ser resolvido por abordagens de aprendizado incrementais (VIJAYAKUMAR et al., 2005).

O algoritmo k-means é um algoritmo de clustering não supervisionado. É preciso um número considerável de amostras não rotuladas para que ele tente agrupá-las em um número “k” de clusters. Não é supervisionado porque os dados não têm classificação externa. O “k” significa o número de clusters que você deseja ter no final. Se k = 5, você terá 5 clusters no conjunto de dados (JAIN, 2010).

O algoritmo dos k vizinhos mais próximos é um algoritmo de classificação supervisionado. É necessário um considerável número de amostras rotuladas para que ele possa aprender a prever a classe de outras. Para rotular uma nova amostra, ele tem como base as amostras rotuladas mais próximos daquele nova, que são seus k vizinhos mais próximos, e faz com que esses vizinhos votem. Então, qualquer que seja o rótulo, a maioria dos vizinhos tem o rótulo para a nova amostra. O “k” em K-Nearest Neighbors é o número de vizinhos que ele verifica. É supervisionado porque o algoritmo está tentando classificar uma amostra com base na classificação conhecida de outras (BEYER et al., 1999).

Dado esse contexto, o objetivo desse trabalho foi implementar um algoritmo de classificação que utiliza conceitos do algoritmo KNN e do algoritmo não supervisionado K-means. O método proposto foi desenvolvido para que o mesmo atue de forma incremental e on-line.

METODOLOGIA:

Para o desenvolvimento do algoritmo proposto foi utilizado o Collaboratory, que é um ambiente de notebook gratuito da Jupyter que não requer nenhum tipo de configuração e é executado inteiramente na nuvem. Com o Collaboratory é possível implementar e executar o código desenvolvido, além de ter a opção de salvar e compartilhar as análises e ainda acessar poderosos recursos de computação, tudo gratuitamente, necessitando apenas de um navegador de internet.

A linguagem de programação escolhida foi a linguagem Python. Inicialmente são importadas as bibliotecas que serão utilizadas no decorrer da implementação e alguns exemplos são a Numpy, Pandas e

ISNN: 2358-6052

Sklearn. Essa última é uma biblioteca de aprendizado de máquina de código aberto para a linguagem de programação Python que inclui vários algoritmos de classificação, regressão e agrupamento como as máquinas de vetores de suporte, florestas aleatórias, gradient boosting, k-means e DBSCAN, e que é projetada para interagir com as bibliotecas do Python numéricas e científicas como a Numpy e SciPy.

Realizada as importações de cada biblioteca a ser utilizada nesse trabalho, é feita a leitura da base de dados. Nesse trabalho, para a realização dos experimentos, foi utilizado seis datasets com diferentes configurações. Os quatro primeiros datasets sintéticos possuem duas classes e duas dimensões, conforme ilustrados na Figura 1.

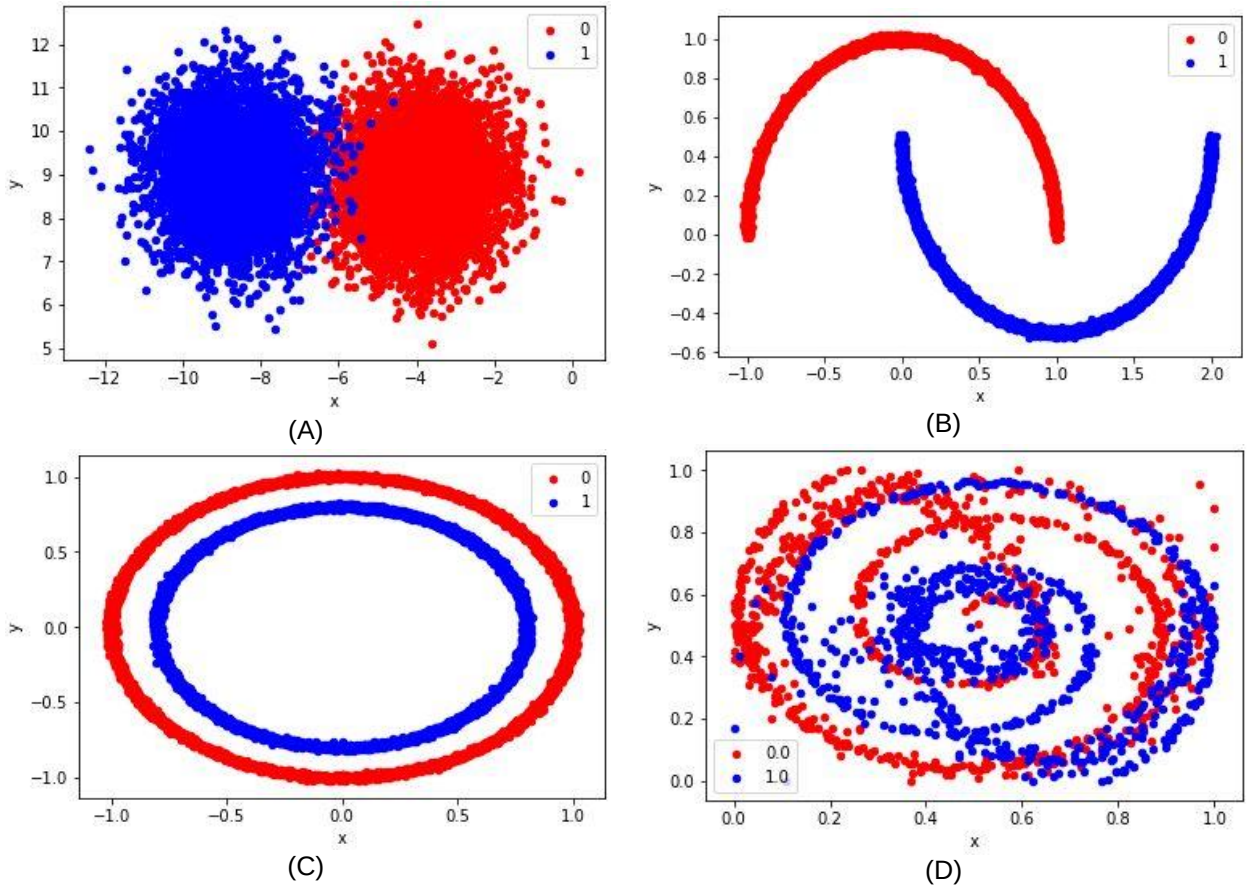


Figura 1: Datasets de duas dimensões.

O quinto conjunto de dados se trata de um dataset real, o Breast Cancer. Este conjunto de dados possui um total de 569 amostras, cada uma possuindo 30 características e pertencente a classe 0 ou a classe 1. O sexto dataset se trata do SEA Concepts, uma base de dados com 60 mil exemplos com três atributos. A cada 15 mil amostras é adicionado um ruído nos dados de forma que se altere seu comportamento.

Levando-se em conta um hipotético cenário on-line, em que se tem milhões de dados e que são considerados os fluxos de dados de maneira contínua, foi realizado na implementação um método que visa a economia do recurso computacional que, para este caso, é a memória disponível do computador. Para isto, após o carregamento da base de dados, essa foi dividida sequencialmente em 20 bases, as quais chegarão em lote para o aprendizado incremental do modelo.

Iniciando-se o processo, para cada lote, é realizada a divisão do dataset atual em um conjunto para dados de treinamento e em um conjunto de base de teste. O conjunto de dados do treinamento possui aproximadamente 80% do tamanho total que é a quantidade de elementos contida no lote atual. Essa divisão é feita de maneira aleatória para que posteriormente se avalie o modelo de forma mais justa e não enviesada.

Na primeira iteração do algoritmo, uma vez que não há nenhum modelo existente, é calculada a média entre os valores de cada característica entre as duas primeiras amostras do lote de treinamento e gerada uma nova amostra para cada classe, que é chamada de centroide. Posteriormente, esse centroide é

ISNN: 2358-6052

atualizado com cada nova amostra que está no conjunto de treinamento e chega incrementalmente, um por um, atualizando o valor de cada centroide de cada classe e consecutivamente o modelo.

Após ser gerado o primeiro modelo com o conjunto atual de treinamento é feita a validação do modelo utilizando o conjunto de teste do lote atual. É feita a classificação das amostras e calculada a acurácia, a área sob a curva ROC das pontuações de predição e também a matriz da confusão.

Para que o algoritmo possa decidir a qual classe tal nova amostra não rotulada pertence, é utilizada uma metodologia parecida com a do algoritmo KNN, para o caso em que o "N" possua o valor um. É calculada a distância euclidiana da nova amostra para todos os centroides de cada classe e, em seguida, com essa matriz de distâncias, define-se a qual classe tal amostra analisada pertence, baseando-se na menor delas.

Posteriormente, simulando um fluxo contínuo de dados, o algoritmo recebe um novo lote de amostras em sua segunda iteração e precisa tomar a decisão entre utilizar estes novos dados que chegam continuamente, para que incrementalmente vá melhorando o modelo ou, esquecer o modelo atual e retreiná-lo do zero. Existem algumas soluções para essa tomada de decisão e uma delas, a implementada neste trabalho, é a técnica do esquecimento abrupto. Em um primeiro momento, decidiu-se criar um limiar que especificava se o modelo atual estava bom ou ruim, ou seja, caso a acurácia do modelo fosse superior a um valor de limiar, o modelo existente seria melhorado gradativamente, caso essa acurácia tivesse um valor inferior ao limiar previamente escolhido, o modelo já treinado era completamente esquecido e apagado da memória.

Após alguns testes foi verificado que uma melhor metodologia para essa técnica poderia ser implementada então, substituiu-se esse limiar pré-especificado por um valor que é uma razão de 90% do valor da acurácia do atual modelo. Exemplificando: se o modelo da iteração atual possui um valor de acurácia de 80%, é calculado se esse valor é maior do que 90% do valor da acurácia do modelo da iteração anterior. Caso o cálculo conclua que o valor atual é inferior a essa razão, acredita-se que exista grande possibilidade de ter ocorrido um concept drift e por isso, o modelo será esquecido e retreinado do zero, haja vista que o modelo atual não consegue classificar bem os novos dados que ele ainda não conhece.

Outra importante decisão a ser tomada durante a execução do algoritmo é: quando acontece o concept drift de forma altamente brusca nos dados, apenas a técnica do esquecimento abrupto baseada por uma razão pode não ser o suficiente. Baseado nisso, foi implementada uma condição que cria uma terceira dimensão caso a acurácia do modelo não esteja melhorando com o passar das iterações. Neste momento, são atualizados alguns parâmetros do modelo de forma que ele consiga se adequar a essa nova estrutura. A Figura 2 ilustra os novos dados após ter ocorrido esse processo de transformação.

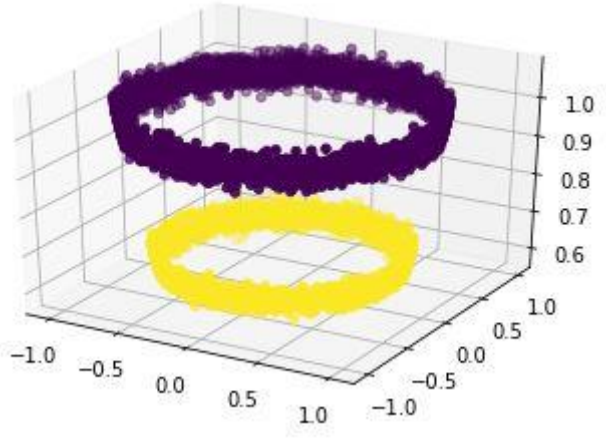


Figura 2: Transformação dos dados.

A partir deste ponto são realizadas as iterações dos demais lotes do conjunto de dados e as etapas descritas anteriormente são repetidas.

RESULTADOS E DISCUSSÕES:

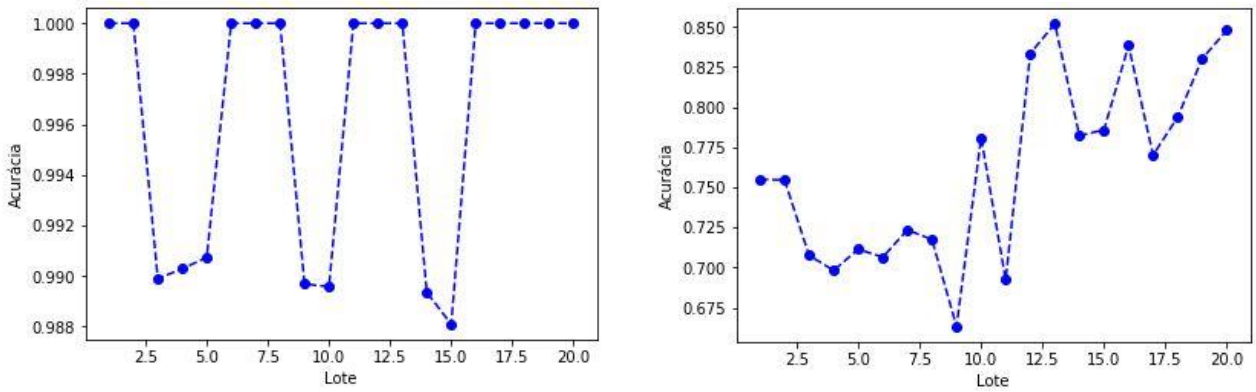
A Tabela 1 apresenta os valores com os resultados obtidos dos experimentos realizados. Os quatro primeiros datasets são conjuntos de dados sintéticos, sendo que o quarto tem sua distribuição de geração dos dados alterada algumas vezes durante o decorrer do conjunto.

ISNN: 2358-6052

Tabela 1: Resultados dos experimentos realizados.

Dataset	ACC	AUC	Matriz de confusão
1	1.0	1.0	44 0 0 46
2	0.84	0.84	34 9 5 44
3	1.0	1.0	50 0 0 44
4	0.62	0.66	7 5 1 3
Breast Cancer	1.0	1.0	4 0 0 13
SEA Concepts	0.74	0.74	96 38 39 130

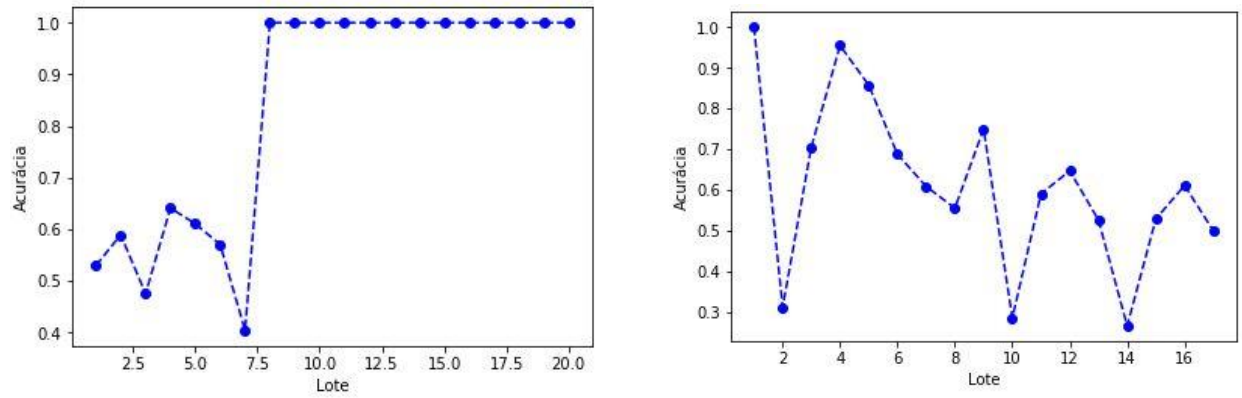
As Figuras 3, 4 e 6 apresentam o gráfico para cada dataset testado, que relaciona a acurácia avaliada sob o conjunto de teste em cada lote, de forma que as novas amostras eram adicionadas ao modelo de maneira incremental.



(A) (B)

Figura 3: Em (A) os resultados obtidos no dataset 1 e em (B) os resultados obtidos no dataset 2.

Por se tratar de um problema cuja característica é praticamente linearmente separável, o método proposto se comportou muito bem. Observou-se uma pequena quantidade de erros em amostras bem específicas que podem ser consideradas outliers. Para o segundo dataset, pôde-se observar resultados bastante similares com os encontrados pela literatura, que giram em torno de 85% de acurácia.



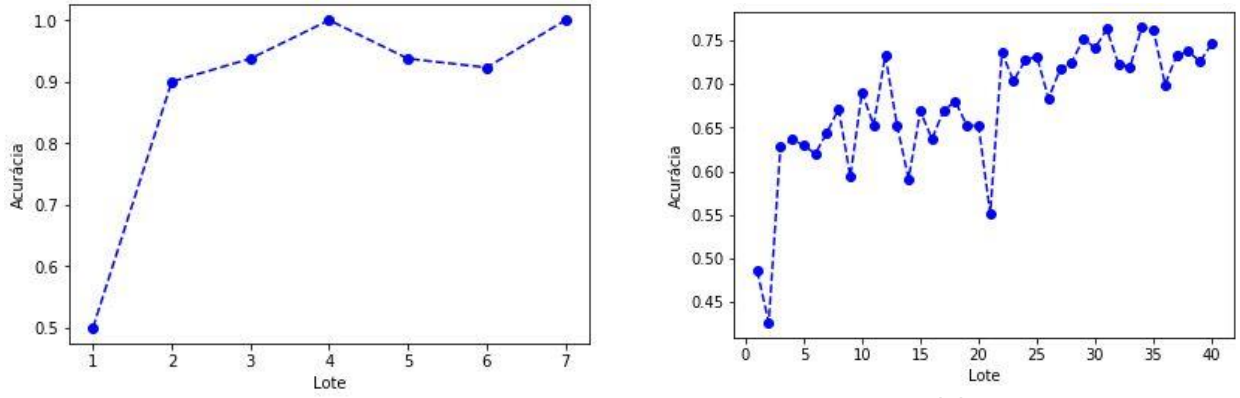
(A) (B)

Figura 4: Em (A) os resultados obtidos no dataset 3 e em (B) os resultados obtidos no dataset 4.

O terceiro dataset tem como característica amostras espalhadas em forma de círculo. Como são duas classes, a disposição das amostras representa um círculo dentro do outro. O método proposto conseguiu resolver bem esse problema devido a condição imposta durante as suas iterações, uma vez que, caso a acurácia esteja baixa e não possua previsão de melhora, cria-se uma terceira dimensão nos dados (Figura 2) e a partir do momento que ocorre essa transformação é possível observar que acontece um aumento significativo e estável da acurácia, conforme é apresentado na 4(A).

O gráfico representado pela Figura 4(B) ilustra bem a mudança de distribuição dos dados. As 100 primeiras amostras possuem separação linear, com o método proposto tendo obtido excelentes resultados. Com a mudança de distribuição, nota-se uma baixa acurácia do modelo para o segundo lote, o que caracteriza o esquecimento abrupto do modelo que será retreinado do 0. Com o seu retreino, novamente têm-se uma ótima acurácia para a nova distribuição.

Os picos inferiores representam as mudanças de distribuição e as quedas bruscas do valor da acurácia para cada lote dos dados. O método proposto conseguiu se comportar bem nas distribuições onde as amostras eram linearmente separáveis, no problema das duas meias-luas e no dataset em que as amostras formavam dois círculos. Para o problema da espiral, é notório que a implementação necessita de melhorias em trabalhos futuros.



(A) (B)

Figura 5: Em (A) os resultados obtidos no dataset Breast Cancer e em (B) os resultados obtidos no dataset SEA Concepts.

Para a base de dados real Breast Cancer é possível perceber na Figura 5(A) o quão bem se comportou o método para classificar corretamente as amostras do conjunto de teste, tendo obtido uma acurácia de 100% até o fim do último lote. Caso não haja uma mudança de distribuição para esses dados, acredita-se que o algoritmo irá conseguir manter esses valores de acerto, considerando um fluxo contínuo de dados.

Considerando o conjunto de dados SEA Concepts, ao se avaliar os resultados obtidos através da Figura 5(B), percebe-se que o modelo demorou algumas iterações até atingir uma acurácia aceitável e a partir desse momento o modelo obteve pequenas variações em seus resultados, exceto quando ocorria o caso do concept drift, que aconteceu no lote 10, 20 e 30, conforme esperado e descrito na documentação da base de dados.

CONCLUSÕES:

Neste trabalho foi proposto um método que utilizou o conceito do algoritmo k-means, para encontrar um centroide para cada classe em um problema supervisionado, e o conceito do algoritmo KNN de classificar as amostras levando em consideração a menor distância entre tal amostra e o centróide mais próximo. A implementação desenvolvida trabalha de maneira incremental, melhorando o modelo já existente sequencialmente e também de maneira on-line, considerando uma melhor manipulação da memória disponível e um possível fluxo contínuo de dados.

Para os datasets avaliados, mostrou-se uma eficiência do algoritmo nas distribuições onde as amostras eram linearmente separáveis, no problema das duas meias-luas, no dataset que as amostram formavam dois círculos, na base de dados reais Breast Cancer e SEA Concepts. Para o problema da espiral, é notório que a implementação necessita de melhorias em trabalhos futuros.

ISNN: 2358-6052

Pôde-se perceber que o algoritmo se mostrou eficaz em alguns casos quando ocorria uma mudança brusca de distribuição, tendo esse sido implementado utilizando a técnica do esquecimento abrupto por meio de uma razão. Existe casos que é muito mais eficiente esquecer totalmente o modelo e retreiná-lo do zero ao invés de tentar melhorá-lo incrementalmente.

Assim, esse estudo mostrou um método incremental capaz de resolver problemas de classificação, trazendo mais uma ferramenta para o vasto estudo desse tipo de algoritmo.

REFERÊNCIAS BIBLIOGRÁFICAS:

BEYER, Kevin et al. When is “nearest neighbor” meaningful?. In: **International conference on database theory**. Springer, Berlin, Heidelberg, 1999. p. 217-235.

BISHOP, Christopher M. **Pattern recognition and machine learning**. springer, 2006.

CAUWENBERGHS, Gert; POGGIO, Tomaso. Incremental and decremental support vector machine learning. In: **Advances in neural information processing systems**. 2001. p. 409-415.

HAZAN, Elad et al. Introduction to online convex optimization. **Foundations and Trends® in Optimization**, v. 2, n. 3-4, p. 157-325, 2016.

JAIN, Anil K. Data clustering: 50 years beyond K-means. **Pattern recognition letters**, v. 31, n. 8, p. 651-666, 2010.

PEDREGOSA, Fabian et al. Scikit-learn: Machine learning in Python. **Journal of machine learning research**, v. 12, n. Oct, p. 2825-2830, 2011.

QUINLAN, J. Ross. **C4. 5: programs for machine learning**. Elsevier, 2014.

ROSS, David A. et al. Incremental learning for robust visual tracking. **International journal of computer vision**, v. 77, n. 1-3, p. 125-141, 2008.

VIJAYAKUMAR, Sethu; D'SOUZA, Aaron; SCHAAL, Stefan. Incremental online learning in high dimensions. **Neural computation**, v. 17, n. 12, p. 2602-2634, 2005.