



Algoritmos Fuzzy aplicados a seleção de locais propícios para instalação de sensores.

Fuzzy algorithms applied to the selection of suitable locations for installing sensors.

Lucas da Silva Lima *

Dr^o Claudio Leones Bazzi †

RESUMO

O projeto possui como base o uso do algoritmo Fuzzy C-means para tomada de decisão na instalação de sensores em árvores nas zonas de manejo, em ambientes de agricultura. O uso deste método computacional de Aprendizado de Máquina prova-se satisfatório, de acordo com os resultados obtidos no texto original (BAZZI et al., 2019). Este projeto publica uma versão para uso pela comunidade de agricultores e também pela comunidade científica. E este projeto é parte do AGRIDATABOX. O projeto tem início com a tradução do código proposto em (BAZZI et al., 2019). Inicialmente escrito em PL SQL, agora traduzido para Python o método apresenta os mesmos resultados, com uma melhora na performance da execução do método. Por fim existe uma API disponibilizando o método para ser utilizado, de modo a melhorar o entendimento há também uma documentação que possui exemplos e estrutura de dados que necessitam ser passadas para o funcionamento do método Fuzzy C-means.

Palavras-chave: Agricultura de Precisão. Aprendizado de Máquina. Zonas de Manejos.

ABSTRACT

The project is based on the use of the Fuzzy C-means algorithm for decision making in the installation of sensors in trees in management zones, in agricultural environments. The use of this computational method of Machine Learning proves to be satisfactory, according to the results obtained in the original text (BAZZI et al., 2019). This project publishes a version for use by the farming community and also by the scientific community. And this project is part of AGDATABOX. The project starts with the translation of the proposed code into (BAZZI et al., 2019). Initially written in PL SQL, now translated to Python the method presents the same results, with an improvement in method execution performance. Finally there is an API providing the method to be used, in order to improve the understanding there is also a documentation that has examples and data structure that need to be passed for the Fuzzy C-means method to work.

Keywords: Precision Agriculture. Machine Learning. Management Zones.

1 INTRODUÇÃO

O agronegócio brasileiro tem tido grande influência no contexto econômico do país e merece que pesquisas sejam desenvolvidas com o intuito de promover e melhorar os procedimentos de produção, reduzindo custos e otimizando o uso dos recursos naturais. Neste sentido, este projeto teve como objetivo: 1) Transferir a funcionalidade do procedimento desenvolvido, referente ao método Fuzzy C-means para seleção de locais propícios para instalação de sensores à campo, proposto em (BAZZI et al., 2019), escrito inicialmente em

* Departamento de computação, Aluno, Curso de Ciência da Computação; ✉ lima@alunos.utfpr.edu.br.

† Departamento de computação, Coordenação, Curso de Ciência da Computação; ✉ bazzi@utfpr.edu.br.



linguagem Postgresql (PL-SQL), para a linguagem Python, visando otimizar o desempenho do algoritmo em função da nova linguagem utilizada; 2) o desenvolvimento de uma Application Programming Interface (API), visando disponibilizar o método para uso ao público através do protocolo Hypertext Transfer Protocol Secure (HTTPS).

A publicação deste método produz acesso aos produtores do campo a um método computacional que indicará os melhores pontos, dado um conjunto de dados, para instalação de sensores em locais propícios. Tais sensores coletam dados para tomada de decisão acerca de parâmetros de clima e desenvolvimento vegetativa das plantas. Com isto promove-se acesso a um recurso importantíssimo que irá melhorar e auxiliar os produtos de um modo geral.

2 MÉTODO (OU PROCEDIMENTOS OPERACIONAIS DA PESQUISA)

A primeira etapa do projeto é a tradução do código escrito inicialmente em linguagem PLPG/SQL, para linguagem Python. Ambas as linguagens possuem paradigmas e propósitos distintos, entretanto as estruturas de dados, estruturas de repetição e condicionais são relativamente semelhantes, apesar do desempenho da linguagem Python ser considerada bastante superior. Enquanto na linguagem PL SQL estava-se utilizando de tabelas no banco de dados Postgresql para armazenar os dados, na nova versão, em linguagem Python, utilizou-se da biblioteca Pandas (MCKINNEY, 2010). A escolha de pandas foi pelos vastos recursos disponíveis por essa biblioteca para a construção de métodos de Aprendizado de Máquinas, como o Fuzzy C-means.

Esta biblioteca Pandas possui os dados, denominados como Data Frame, ele é basicamente uma matriz de dados que possuem colunas com os atributos para cada conjunto de dados. Esta estrutura é fácil de manipular e as operações de leituras, escritas e outras operações, como operações matemáticas ou booleanas, são relativamente simples de realizar. Através desta biblioteca, ainda é possível realizar leitura direto de banco de dados ou de arquivos de textos devidamente formatados, como os CSV.

Para a construção da API, foi utilizada a biblioteca Flasgger. Esta biblioteca é muito comum no mercado e extremamente consolidada, possui uma interface de usuário amigável para a documentação dos *endpoints*, baseado no Swagger. Os dois principais motivos para a escolha desta biblioteca foram dois. Primeiro é que ela é construída em Python e, deste modo, fornecerá fácil integração com a solução traduzida do método Fuzzy C-means. Segundo é que possui uma interface com o usuário baseado no Swagger, o que fornece uma documentação completa para os usuários desta API.

Flasgger corresponde a uma extensão de Flask para extração da Open API-Specification a partir de views do Flask registrada na API que está sendo construída. Com um simples arquivo de estruturação de dados, como YAML, Flask consegue construir quais parâmetros cada endpoint suporte e deve receber, como o que será retornado em cada ocasião. Ficando assim a cargo de quem codifica construir as regras de negócio.

3 RESULTADOS

O primeiro passo realizado foi a construção do algoritmo Fuzzy C-means, como este algoritmo foi traduzido da linguagem PL-SQL alguns problemas foram encontrados, como o uso de *array* bidimensionais de *string* que indicam tabelas de dados dentro do banco de dados. Isso ocorria pois a bordagem no código PL SQL primeiro organizava os dados e os atributos, para só então iniciar, de fato, o algoritmo Fuzzy. Ao tentar fazer isso em Python houve diversos problemas, como a listagem das tabelas disponíveis e tratamento dos erros caso estas não



existissem. Logo a solução foi iniciar de fato o algoritmo Fuzzy, considerando já haver todos os dados e atributos devidamente organizados.

Uma vez que os dados já estão organizados, em estruturas de *Data Frame*, da biblioteca Pandas, as traduções seguintes foram simples, apenas entender os cálculos sobre os dados e validar cada retorno das operações feitas sobre o conjunto de dados. Na Figura 1 é possível entender o modelo implementado e seu funcionamento.

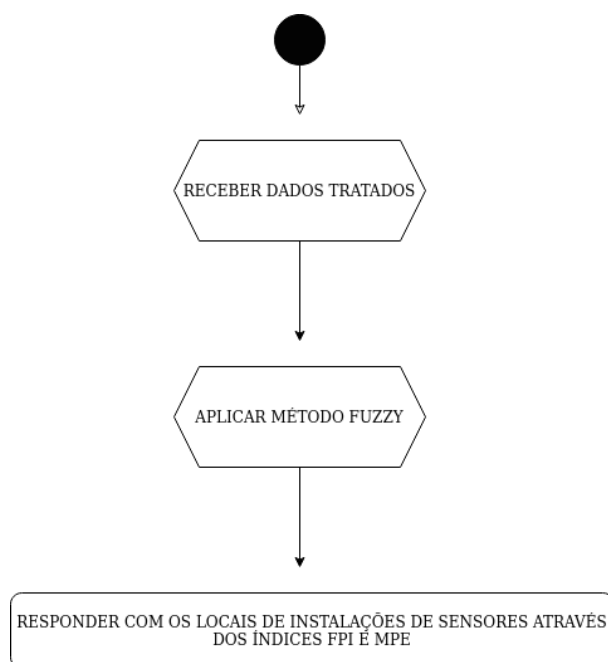


Figura 1 – Flowchart simplificado de fluxo de execução na API

Fonte: Autoria Própria

O algoritmo Fuzzy C-means possui um cálculo de pertinência, este cálculo irá “guiar” cada elemento, do conjunto de dados, a receber um grau de pertinência ao *cluster* no qual está sendo calculado. Os pontos com melhores valores serão escolhidos pelo algoritmo como candidatos a instalação dos sensores. Na Figura 2 é possível ver como a tradução foi feita e como este cálculo funciona.

```
def fuzzy_cmeans(data, clusters):  
    ...  
    pertinence = pd.DataFrame()  
    for columns in dists.columns:  
        pertinenes = pd.DataFrame()  
        for columns_aux in dists.columns:  
            pertinenes[columns_aux] = np.power(  
                (dists[columns] / dists[columns_aux]), (1.0/3.0))  
        pertinence[columns] = pertinenes.sum(axis=1)  
        pertinence[columns] = np.power(pertinence[columns], -1.0)  
    ...
```

Figura 2 – Trecho do código que é calculado a pertinência no Fuzzy C-means

Fonte: Autoria Própria

A produção de uma API, como segunda etapa do projeto seguiu de maneira mais autônoma, uma vez



que não foi uma tradução de algo já pronto. A interface de usuário para utilização da API ajuda os usuários entender o que necessitam passar como requisição para a API e recebem um retorno. Esta API está integrada ao AGRIDATABOX e a autenticação é feita nela. A requisição para usar o método Fuzzy segue dois tipos. O primeiro no qual o usuário passa todos os dados e o Fuzzy retorna os melhores pontos, ou árvores, onde se deve instalar a quantidade de sensores desejada. Este resultado é dado através de dois valores, obtidos pelas estatísticas dos índices Fuzzy performance index (FPI) (FRIDGEN et al., 2004) e Modified partition entropy index (MPE) (BOYDELL; MCBRATNEY, 2002).

4 CONCLUSÕES

Com este trabalho é possível que pesquisadores e agricultores utilizem de uma excelente solução computacional para a instalação de sensores em zonas de manejo. A API está disposta nos mesmos endereços do AGRIDATABOX e prontas a serem utilizadas.

Os métodos de Aprendizado de Máquina, como Fuzzy C-means apresentam bons resultados para tomada de decisões que não é necessário ser 100% preciso, mas uma boa aproximação já é suficiente.

AGRADECIMENTOS

Agradecimento ao Ministério de Agricultura, Pecuária e Abastecimento (MAPA), Conselho Nacional de Desenvolvimento Científico e Tecnológico (CNPq) por fomentar tal pesquisa e agradecimento a Universidade Tecnológica Federal do Paraná por promover o ensino e extensão para a comunidade de forma eficiente, ética e responsiva.

REFERÊNCIAS

- BAZZI, Claudio Leones et al. Optimal placement of proximal sensors for precision irrigation in tree crops. **Precision Agriculture**, Springer, v. 20, n. 4, p. 663–674, 2019.
- BOYDELL, B; MCBRATNEY, AB. Identifying potential within-field management zones from cotton-yield estimates. **Precision agriculture**, Springer, v. 3, n. 1, p. 9–23, 2002.
- FRIDGEN, Jon J et al. Management Zone Analyst (MZA) Software for Subfield Management Zone Delineation. **Agronomy Journal**, Wiley Online Library, v. 96, n. 1, p. 100–108, 2004.
- MCKINNEY, Wes. Data Structures for Statistical Computing in Python. In: WALT, Stéfan van der; MILLMAN, Jarrod (Ed.). **Proceedings of the 9th Python in Science Conference**. [S.l.: s.n.], 2010. P. 56–61. DOI: [10.25080/Majora-92bf1922-00a](https://doi.org/10.25080/Majora-92bf1922-00a).