CENTRO UNIVERSITÁRIO ENIAC

AUTORES:

José Antonio Dias de Carvalho

Mateus Barbosa Duarte

UTILIZANDO MACHINE LEARNING NA IDENTIFICAÇÃO DE EXOPLANETAS ATRAVÉS DE ANÁLISE DE FLUXO DE LUZ

GUARULHOS

2023

UTILIZANDO MACHINE LEARNING NA IDENTIFICAÇÃO DE EXOPLANETAS ATRAVÉS DE ANÁLISE DE FLUXO DE LUZ

RESUMO

A busca por planetas fora do sistema solar é um dos campos mais promissores da astronomia moderna. Nos últimos anos, o telescópio Kepler da NASA foi fundamental na identificação de exoplanetas, fornecendo uma grande quantidade de dados de fluxo de luz para análise. No entanto, o processo de detecção de exoplanetas é complexo e pode ser difícil para os astrônomos identificarem padrões. O presente artigo elabora o desenvolvimento de um modelo de machine learning em Python para detectar exoplanetas a partir dos gráficos de fluxo de luz do telescópio Kepler, utilizando-se de duas abordagens, Árvores de Decisões e Regressão Logística. Os modelos foram treinados usando um conjunto de dados extenso e rotulado, contendo exemplos de exoplanetas e objetos não planetários, sendo realizada a comparação da precisão de ambos. Os resultados mostraram que a Regressão Logística apresentou uma precisão superior na detecção de exoplanetas. Tais pesquisas fazem-se necessárias para o estudo de técnicas de detecção e análise de dados astronômicos por meio de inteligência artificial.

Palavras-chave: Exoplanetas. Inteligência Artificial. Machine Learning. Árvores de Decisões. Regressão Logística.

ABSTRACT

The search for planets outside the solar system is one of the most promising fields in modern astronomy. In recent years, NASA's Kepler telescope has been instrumental in identifying exoplanets, providing a vast amount of light curve data for analysis. However, the process of exoplanet detection is complex and can be challenging for astronomers to identify patterns. This article outlines the development of a Python machine learning model for exoplanet detection using light curve graphs from the Kepler telescope, employing two approaches, Decision Trees, and Logistic Regression. The models were trained using an extensive labeled dataset containing examples of exoplanets and non-planetary objects, and a comparison of their accuracy was conducted. The results demonstrated that Logistic Regression exhibited higher accuracy in exoplanet detection. Such research is essential for the study of techniques for the detection and analysis of astronomical data through artificial intelligence.

1. INTRODUÇÃO

A busca por planetas fora do nosso sistema solar, os exoplanetas, tem sido uma área de interesse crescente na astronomia (Wolszczan & Frail 1992). O telescópio Kepler, lançado em 2009 pela NASA, foi projetado para identificar exoplanetas através da detecção de trânsitos planetários, quando um planeta passa em frente à sua estrela hospedeira, causando uma pequena queda no fluxo de luz observado (Borucki et al. 2010). No entanto, analisar e interpretar os dados brutos dos gráficos de fluxo de luz gerados por telescópios espaciais, pode ser um desafio para os astrônomos (Howell et al. 2014). Neste trabalho, foi desenvolvido um modelo de inteligência artificial (IA) em Python para ajudar na detecção de exoplanetas a partir dos gráficos de fluxo de luz do telescópios Kepler.

2. OBJETIVOS

O objetivo geral deste trabalho é desenvolver um modelo de inteligência artificial que possa detectar exoplanetas com base nos dados de fluxo de luz fornecidos pelo telescópio Kepler.

Objetivos Específicos:

Obter dados gráficos de fluxo de luz das estrelas observadas pelo telescópio Kepler através das instituições responsáveis.

Programar em Python utilizando aprendizado de máquina através do método de árvores de decisões, bem como avaliar o método de acordo com sua precisão.

Tratar os dados e programar em Python utilizando aprendizado de máquina através do preceito de regressão logística e avaliá-lo a partir de sua precisão.

Comparar a performance de Árvores de Decisões com Regressão Logística em ordem de estimar o método mais eficaz.

3. METODOLOGIA

A pesquisa possui caráter quantitativo, e por tal ordem, utiliza-se da ferramenta de programação Python, para construir modelos de aprendizado de máquina bem como avaliar sua performance. A biblioteca em Python utilizada para tal fim foi *Scikit-Learn* enquanto para a análise dos resultados foi utilizada a biblioteca *Seaborn* e *Matplotlib.*

Os dados utilizados para o treinamento de máquina são disponibilizados pelo *Mikulski Archive for Space Telescopes* (MAST 2022), e são compostos por amostras coletadas de forma regular em intervalos iguais, abrangendo 80 dias no total. Essa coleta de mais de 3000 intervalos é realizada levando em consideração o uso do Tempo Bariocêntrico, uma escala de tempo que leva em conta as variações gravitacionais causadas pela posição relativa dos corpos celestes no sistema solar. Essa abordagem permite uma maior precisão nas medições e é particularmente

relevante em estudos astronômicos e cálculos de órbita.

O aprendizado de máquina é uma subárea da inteligência artificial que se baseia em algoritmos que podem aprender a partir de dados e fazer previsões ou tomar decisões com base nesses dados (Malik et al. 2022). Com os gráficos de fluxo de luz (medindo a variação do brilho da estrela ao longo do tempo), o objetivo é que o modelo aprenda a identificar os padrões que indicam a presença de um exoplaneta.

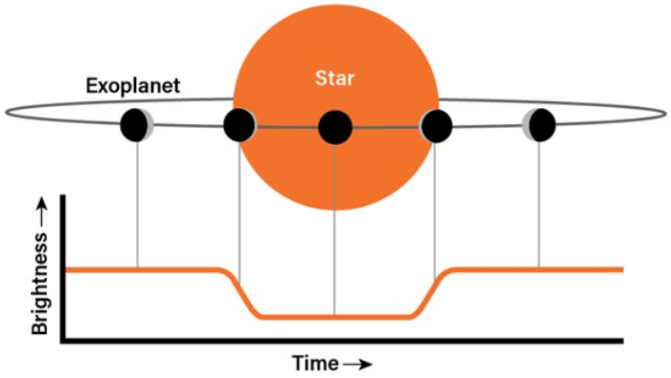
Os métodos de aprendizado de máquina utilizados neste trabalho foram Árvores de decisões e Regressão logística. A Árvore de decisão é um método que se baseia em criar uma árvore de perguntas para classificar os dados (Navada et al. 2022). A partir do conjunto de treinamento, o modelo de Árvore de decisão aprende a melhor sequência de perguntas para classificar corretamente os dados. Na classificação de exoplanetas, o modelo pode fazer perguntas como "o gráfico de fluxo de luz tem um padrão de queda regular?" ou "o tempo entre as quedas é consistente?". O resultado é uma árvore de decisão que pode ser usada para classificar novos dados.

A Regressão logística é uma técnica de classificação que usa uma função matemática para modelar a relação entre as variáveis de entrada (no caso, as informações do gráfico de fluxo de luz) e a saída (a presença ou ausência de exoplanetas). A ideia é que a função matemática possa ser ajustada aos dados de treinamento para encontrar a melhor forma de classificar novos dados. A regressão logística é comumente usada para problemas de classificação binária (Gladence et al. 2015), como o caso de exoplanetas, onde a saída é ou "exoplaneta" ou "não exoplaneta".

4. DESENVOLVIMENTO

A curva de luz presente nos dados utilizados neste estudo para treinamento de máquina é ilustrada de forma mais representativa na Figura 1. Nessa figura, é possível observar a diminuição gradual da luminosidade (*brightness*) ao longo do tempo (*time)* à medida que um exoplaneta (*exoplanet*) transita ao redor de sua estrela (*star*). Essa característica distintiva nos gráficos de fluxo de luz acompanhada de um comportamento periódico, fornece indícios para identificação de exoplanetas (GILLILAND et al. 2010) por meio da análise das curvas de luz.

Figura 1 – Luminosidade Observada e Posição do Planeta em Relação a Estrela.

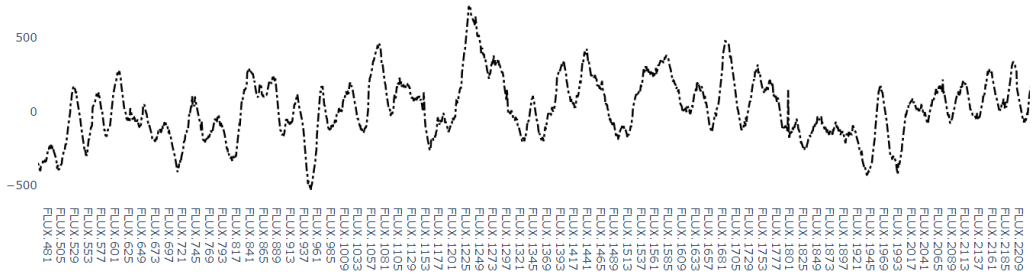


Fonte: Astronomy/Roen Kelly, 2023

Para que seja possível trabalhar com duas classes: exoplanetas e não exoplanetas, temos os dados rotulados com "1" são representados pela presença de um exoplaneta, enquanto os rotulados com "0" são representados pela ausência de um exoplaneta. As Figuras 2 e 3 ilustram os gráficos obtidos pelas medições.

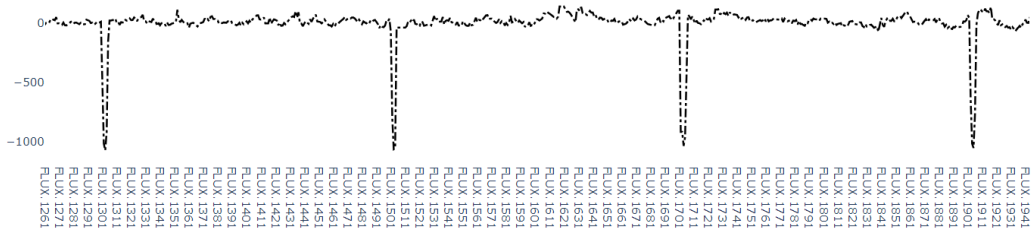
Nota-se o comportamento na Figura 2, característico de ruídos comuns em estrelas onde estima-se não haver exoplanetas, enquanto a Figura 3, apresenta um gráfico com certa periodicidade em sua queda de fluxo, onde a teoria aponta existência de exoplanetas (GILLILAND et al. 2010).

Figura 2 - Fluxo de Luz Rotulado Como “0”: Ausência de Exoplaneta



Fonte: Elaborada pelo Autor

Figura 3 - Fluxo de Luz Rotulado como “1”: Presença de Exoplaneta



Fonte: Elaborada pelo Autor

Como há um desequilíbrio entre as classes, é utilizado o SMOTE (técnica de oversampling

da classe minoritária sintetizando amostras) para equilibrar as classes.

Após o balanceamento das classes, os conjuntos de dados são divididos em subconjuntos de treinamento e teste e normalizados. Em seguida, são criados três modelos de classificação: Árvore de Decisões, e Regressão Logística. Os modelos são ajustados aos dados de treinamento e testados nos dados de teste.

As figuras 4 e 5 mostram os códigos utilizados no programa para ambos os modelos de teste.

Figura 4 – Código Python: Arvore de Decisões

```
ds_model = DecisionTreeClassifier(max_depth=3.8, random_state=1)
ds_model.fit(x_train_res, y_train_res)
prediction = ds_model.predict(x_test)
print("\nParâmetros de Desempenho:\n", (classification_report(y_test, prediction)))
# Calcular a matriz de confusão
conf_mat = confusion_matrix(y_test, prediction)
cmap = "Blues"
plt.figure(figsize=(8, 6))
sns.heatmap(conf_mat, annot=True, cmap=cmap, fmt="d", linecolor="k", linewidths=3)
plt.title("Matriz de Confusão Árvore de Decisões", fontsize=16)
plt.xlabel("Classe Prevista")
plt.ylabel("Classe Verdadeira")
plt.show()
```

Fonte: Elaborada pelo Autor

Figura 5 – Código Python: Regressão Logística

```
#Regressão Logística
lr_model = LogisticRegression(C=0.002)
lr_model.fit(x_train_res,y_train_res)
prediction=lr_model.predict(x_test)
print ("\nParâmetros de Desempenho :\n",(classification_report(y_test,prediction)))
# Calcular a matriz de confusão
conf_mat = confusion_matrix(y_test, prediction)
cmap = "Blues"
plt.figure(figsize=(8, 6))
sns.heatmap(conf_mat, annot=True, cmap=cmap, fmt="d", linecolor="k", linewidths=3)
plt.title("Matriz de Confusão Regressão Logística", fontsize=16)
plt.xlabel("Classe Prevista")
plt.ylabel("Classe Verdadeira")
plt.show()
```

Fonte: Elaborada pelo Autor

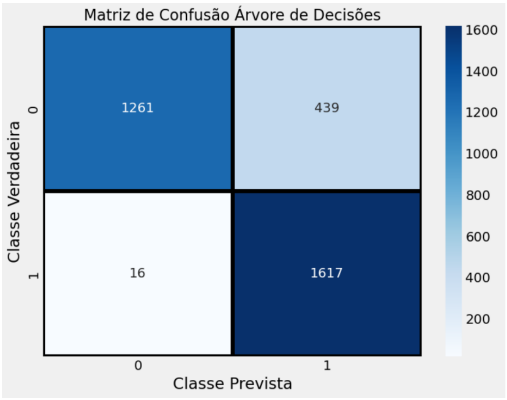
Para cada modelo, foi ajustado os dados de treinamento, fazendo uma previsão com os dados de teste. Foi plotado um relatório de classificação usando a função *classification_report()* e exibindo uma matriz de confusão usando a biblioteca *Seaborn.*

5. RESULTADOS

Os modelos de aprendizado de máquina foram treinados e testados utilizando a base de dados disponível. As Tabelas 1 e 2, apresentam os resultados da matriz de confusão das quais

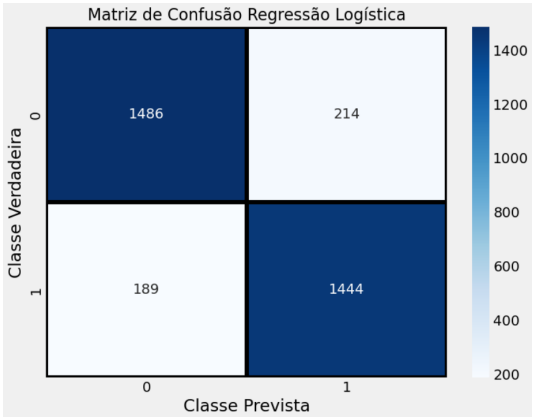
foram criadas como demonstrado nas Figuras 4 e 5

Tabela 1 – Matriz de Confusão Árvore de Decisões



Fonte: Elaborada pelo Autor

Tabela 2 – Matriz de Confusão Regressão Logística



Fonte: Elaborada pelo Autor

A matriz de confusão é composta por quatro valores: verdadeiro positivo (canto inferior direito), verdadeiro negativo (canto superior esquerdo), falso positivo (canto superior direito) e falso negativo (canto inferior esquerdo). O verdadeiro positivo representa a quantidade de instâncias que foram classificadas corretamente como positivas, o verdadeiro negativo representa a quantidade de instâncias que foram classificadas corretamente como negativas, o falso positivo representa a quantidade de instâncias que foram classificadas incorretamente como positivas, e o falso negativo representa a quantidade de instâncias que foram classificadas incorretamente como negativas.

A matriz de confusão da Árvore de Decisões (Tabela 1) mostrou um total de 1617 verdadeiros positivos, 1261 verdadeiros negativos, 439 falsos positivos e 16 falso negativo. Enquanto a matriz de confusão da Regressão Logística (Tabela 2) apresentou um total de 1444

verdadeiros positivos, 1486 verdadeiros negativos, 214 falsos positivos e 189 falsos negativos.

Nas Tabelas 3 e 4 apresenta-se precisão, *recall, f1-score* e acurácia, índices de desempenho dos quais são possíveis obter através dos valores da matriz de confusão, bem como pela função *classification_report()*, apresentado nas Figuras 2 e 3. Este possui o intuito de coerentemente analisar cada modelo aqui trabalhado.

Obtido o quociente do verdadeiro positivo pela soma do verdadeiro positivo com o falso positivo, isto é, a proporção de instâncias identificadas como positivas e que realmente o eram, denomina-se precisão:

$$precisão = \frac{verdadeiro\ positivo}{verdadeiro\ positivo + falso\ positivo}$$

Tem-se, portanto, que a Árvore de Decisões apresentou uma precisão de 79% em encontrar gráficos de fluxos de luz característicos de estrelas das quais possuem exoplanetas em sua orbita, enquanto a precisão da Regressão Logística foi de 87%. Nota-se maior capacidade da Regressão Logística em evitar falsos positivos.

Com os dados de verdadeiro positivo, dividido pela soma de verdadeiros positivos e falsos negativos, obtém-se o *recall*:

$$precisão = \frac{verdadeiro\ positivo}{verdadeiro\ positivo + falso\ negativo}$$

Indicativo da capacidade do modelo de prever resultados positivos, ou no caso da avaliação dos negativos, a capacidade do modelo de prever resultados negativos (nesta situação o cálculo seria obtido pelo quociente de verdadeiros negativos dividido pela soma de verdadeiros negativos e falsos positivos).

Apesar de a Regressão Logística possuir maior precisão para identificar casos com exoplanetas, como a Tabela 3 e 4 indicam, houve um nível maior de *recall* na árvore de decisões 99%, do que na Regressão Logística 88%, onde constata-se uma maior capacidade do primeiro modelo, de evitar falsos negativos.

Desta forma o índice de f1-score faz-se necessários. Este, indica o balanço entre o *recall* e a precisão, a partir do cálculo:

$$f1\_score = 2\ x\ \frac{precisão\ x\ recall}{precisão + recall}$$

Levando em conta os dois métodos, é notável o maior rendimento apresentado na Tabela

4 marcando 88% de *f1-score* em casos de positivos e negativos, em comparação ao *f1-score* de 85% em casos negativos na Tabela 3.

Por fim, a acurácia consolida o atestado pela métrica de *recall*, através das soluções possíveis, calculando o quociente de verdadeiro positivo pela soma de todos os índices:

$$acurácia = \frac{verdadeiro\ positivo}{verdadeiro\ positivo + verdadeiro\ negativo + falso\ positivo + falso\ negativo}$$

A Regressão Logística apresenta um índice superior (88%) em comparação a Árvore de Decisões (86%)

Tabela 3 – Parâmetros de Desempenho Árvore de Decisões

	Existência de Exoplanetas	**Não Existência de Exoplanetas**
Precisão	79%	99%
Recall	99%	74%
f1-Score	88%	85%
Acurácia	86%	

Fonte: Elaborada pelo Autor

Tabela 4 – Parâmetros de Desempenho Regressão Logística

	Existência de Exoplanetas	**Não Existência de Exoplanetas**
Precisão	87%	89%
Recall	88%	87%
f1-Score	88%	88%
Acurácia	88%	

Fonte: Elaborada pelo Autor

Com base nos resultados obtidos, nota-se uma maior capacidade do modelo de Regressão Logística em encontrar padrões de existência de exoplanetas em gráficos de fluxo de luz, embora a Árvore de Decisões tenha obtido maior eficácia em encontrar gráficos ausentes de exoplanetas, obteve maiores índices de falsos positivos, atribuindo casos positivos, como não existência de exoplanetas, consequentemente, não obteve um valor de acurácia e *f1-score* superiores ao segundo modelo.

6. CONSIDERAÇÕES FINAIS

Foi desenvolvido um modelo de aprendizado de máquina para detectar exoplanetas a partir dos dados de fluxo de luz fornecidos pelo telescópio Kepler. Utilizando os métodos de Árvore de Decisões e Regressão Logística, comparando a precisão de ambos os modelos. Os resultados mostraram que o modelo de Regressão Logística teve uma precisão melhor que o modelo de Árvore de Decisões.

Conclui-se que a utilização de técnicas de aprendizado de máquina pode ser muito útil para auxiliar na identificação de exoplanetas a partir dos dados brutos do telescópio Kepler. Além disso, o modelo de Regressão Logística pode ser mais adequado para essa tarefa específica, embora a escolha final dependa do conjunto de dados e do contexto específico.

Embora haja evidências da eficácia da aplicação de aprendizado de máquina na detecção de exoplanetas, é importante ressaltar que quanto maior a quantidade de dados melhor este tipo de aplicação será posto à prova, sendo necessário prevenir o *overfitting* (Dietterich 2022), ainda que (pela natureza do trabalho) os dados catalogados como positivos, isto é, onde há a presença de exoplanetas, sejam mais difíceis de encontrar. Também cabe aqui constatar que a interpretação dos resultados ainda requer a intervenção humana. Portanto, essa tecnologia pode ser usada para aumentar a eficiência dos astrônomos, mas nunca os substituir.

Este modelo pode continuar sendo aperfeiçoado conforme novos dados sejam disponibilizados pelos órgãos vigentes das missões espaciais cujo o escopo produza dados semelhantes, como *Tess, Hubble e Webb*. Em suma, o desenvolvimento de modelos de aprendizado de máquina para a detecção de exoplanetas é uma área promissora de pesquisa que pode levar a descobertas ainda mais significativas no futuro.

7. FONTES CONSULTADAS

WOLSZCZAN, Aleksander; FRAIL, Dail A. A planetary system around the millisecond pulsar PSR1257+ 12. Nature, v. 355, n. 6356, p. 145-147, 1992.

GILLILAND, Ronald L. et al. Kepler asteroseismology program: introduction and first results. Publications of the Astronomical Society of the Pacific, v. 122, n. 888, p. 131, 2010.

HOWELL, Steve B. et al. The K2 mission: characterization and early results. Publications of the Astronomical Society of the Pacific, v. 126, n. 938, p. 398, 2014.

MALIK, Abhishek; MOSTER, Benjamin P.; OBERMEIER, Christian. Exoplanet detection using machine learning. Monthly Notices of the Royal Astronomical Society, v. 513, n. 4, p. 5505-5516, 2022.

NAVADA, Arundhati et al. Overview of use of decision tree algorithms in machine learning.

In: 2011 IEEE control and system graduate research colloquium. IEEE, 2011. p. 37-42.
GLADENCE, L. Mary; KARTHI, M.; ANU, V. Maria. A statistical comparison of logistic regression and different Bayes classification methods for machine learning. ARPN Journal of Engineering and Applied Sciences, v. 10, n. 14, p. 5947-5953, 2015.
Mikulski Archive for Space Telescopes (MAST) Portal. Disponível em: <https://mast.stsci.edu/portal/Mashup/Clients/Mast/Portal.html>.
DIETTERICH, Tom. Overfitting and undercomputing in machine learning. ACM computing surveys (CSUR), v. 27, n. 3, p. 326-327, 1995.