

PROGRAMA COMPUTACIONAL PARA A IDENTIFICAÇÃO AUTOMÁTICA DE EXOPLANETAS

COMPUTATIONAL PROGRAM FOR AUTOMATIC IDENTIFICATION OF EXOPLANETS

PROGRAMA COMPUTACIONAL PARA LA IDENTIFICACIÓN AUTOMÁTICA DE EXOPLANETAS

Patricia Oliveira Montanger¹
Willian Zalewski²

Resumo: A coleta de informações ao longo do tempo se aplica em inúmeras situações, sendo de grande interesse a análise desses dados de maneira rápida e eficaz. Dados coletados ao longo do tempo podem ser representados por uma série temporal, como acontece com as curvas de luz de exoplanetas. Neste trabalho buscamos desenvolver métodos para a análise dessas séries a partir da aplicação de gráficos de recorrência, uma ferramenta de visualização de séries temporais baseada na exploração do seu comportamento característico. Assim identificamos exoplanetas por meio de algoritmos de aprendizagem de máquina e junto a validação cruzada avaliamos seus desempenhos.

Palavras-chave: Séries temporais. Curvas de luz. Gráficos de recorrência. Aprendizado de máquina.

Abstract: The collection of information over time applies in numerous situations, being of great interest the analysis of these data quickly and effectively. Data collected over time can be represented through [a](#) time series, is what happens to the light curves of exoplanets. In this work we seek the development of methods for analysis of temporal series from the application of recurrence plots, which are time series visualization tool based on the exploration of the recurring behavior characteristic. Thus, we identify exoplanets through machine learning algorithms and analyze the results along [cross](#)-validation that evaluates the performance of the classification models.

Keywords: Time series. Light curves. Recurrence plots. Machine Learning.

Resumen: La recopilación de información a lo largo del tiempo se aplica en numerosas situaciones, siendo de gran interés analizar estos datos de manera rápida y efectiva. Datos recopilados a lo largo del tiempo se pueden representar mediante una serie temporal, como las curvas de luz de exoplanetas. En este trabajo buscamos desarrollar métodos para el análisis de estas series a partir de la aplicación de gráficos de recurrencia, una herramienta de visualización de series basada en la exploración de su comportamiento característico. Así, identificamos exoplanetas utilizando algoritmos de aprendizaje automático y junto con la validación cruzada, evaluamos su rendimiento.

Palabras-clave: Series temporales. Curvas de luz. Gráficos de recurrencia. Aprendizaje de máquina.

Envio 20/01/2020

Revisão 30/01/2020

Aceite 15/03/2020

¹Graduanda em Engenharia Física. Universidade Federal da Integração Latino-Americana. E-mail: patricia.montanger@aluno.unila.edu.br.

²Doutor. Universidade Federal da Integração Latino-Americana. E-mail: willian.zalewski@unila.edu.br.

Introdução

Em diversas áreas do conhecimento estão presentes informações que estão sujeitas a variações temporais, como na economia com os preços diários de ações, na medicina em eletrocardiogramas, na meteorologia e na astrofísica com a identificação de objetos celestes. Estes exemplos tão importantes e comuns no dia a dia demonstram como o desenvolvimento tecnológico referente ao armazenamento e ao processamento de dados temporais tem se tornado cada vez mais relevante. O tipo de dado temporal mais comum é chamado de série temporal, a qual pode ser entendida como um conjunto ordenado de observações registradas cronologicamente. Neste trabalho, a série temporal de interesse é a variação da intensidade luminosa de corpos celestes coletada pelo telescópio Kepler da NASA, que foi lançado a órbita no ano de dois mil e nove com a missão de procurar por planetas fora do Sistema Solar, os exoplanetas, especialmente aqueles que possuísem características habitáveis. Para isso foram observadas mais de cem mil das estrelas mais brilhantes do céu, resultando assim em milhares de diferentes curvas de luz. A missão acabou em dois mil e dezoito porém durante os anos de utilização do telescópio foram descobertos mais de dois mil e seiscentos exoplanetas e muitos ainda podem ser descobertos visto a grande banco de dados que foi coletada durante as observações. Para continuar a busca por exoplanetas, foi lançado o telescópio TESS que também será capaz de coletar informações referentes a asteróides e cometas.

A base do Kepler que escolhemos para realizar esta pesquisa nos fornece uma base de dados com cadência máxima de 30 minutos entre cada registro do objeto de interesse, denominadas curvas de luz. Esse tipo de dado possibilita obter diversas informações sobre os valores de massa e raio de estrelas, supernovas, sistemas binários e, em especial, pode ser utilizado para identificação de exoplanetas por meio da análise do trânsito planetário.

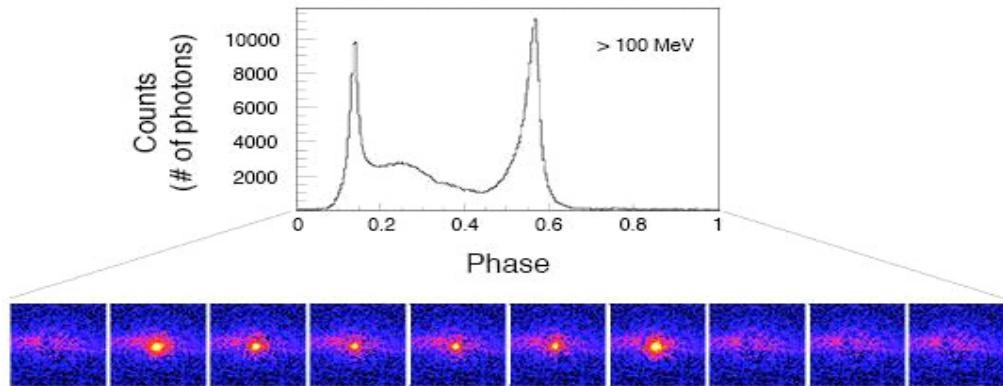
As abordagens tradicionais para a análise de séries temporais são baseadas em métodos estatísticos, os quais, em geral, não se mostram eficientes em domínios de dados não lineares. Estes métodos analisam cada dado da série independentemente, sem considerar o fato de que existe uma relação temporal entre cada uma das observações realizadas, ou seja, a medida no tempo t_2 tem influência das medidas t_1 e t_3 . Mediante esta restrição das abordagens estatísticas, muitos estudos propuseram a utilização de técnicas de aprendizado de máquina (Willian Zalewski, 2015). Essas técnicas são baseadas na inferência indutiva, a qual

possibilita derivar novos conhecimentos automaticamente a partir de outros previamente conhecidos (Mitchell, 1997). Nesse contexto, neste trabalho estudamos as curvas de luz provenientes do trânsito planetário, por meio da técnica de gráficos de recorrência em combinação com algoritmos de aprendizado de máquina. Nosso objetivo é contribuir de modo significativo para o processo de classificação automática de exoplanetas, agregando informações para auxiliar no processo de tomada de decisões de astrônomos e tornar mais rápido o processo de análise da grande quantidade de dados temporais que são tomados todos os dias.

Fundamentação Teórica

Nosso objeto de estudo são as curvas de luz que identificam exoplanetas, as quais podem ser entendidas como séries temporais constituídas pela variação do brilho de um objeto celeste no tempo. Essa variação no brilho das estrelas também pode ser denominada variação da amplitude, onde menor é o valor da amplitude quanto maior for o brilho, e ocorre devido a um fenômeno chamado trânsito planetário. Este fenômeno ocorre quando um exoplaneta realiza a órbita em torno de sua estrela e visto que o telescópio aponta sempre numa mesma direção, em algum momento o exoplaneta passa entre a estrela e o telescópio, o que tem como consequência a diminuição do brilho capturado pelas imagens feitas pelo telescópio. Reunindo as imagens capturadas com o passar do tempo, de pelo menos uma órbita completa, começamos a observar a formação de uma curva de luz e o padrão que existe nela quando se trata de uma curva de uma estrela que possui exoplanetas. A Figura 1 contém uma representação esquemática do conceito de trânsito planetário. Nas primeiras imagens em que a estrela apresenta maior brilho, a curva está num ponto mais alto e que com o passar do tempo o brilho da estrela diminui, nas últimas imagens, assim como na curva que ao fim do gráfico vai se aproximando de pontos mais baixos. Lembrando que quando trabalhamos com magnitudes de estrelas, quanto menor o valor a magnitude, incluindo valores negativos, mais visível é seu brilho, e quanto maior o valor da magnitude mais difícil é visualizar o brilho da estrela.

Figura 1 - Formação de uma curva de luz através de imagens capturadas por telescópio

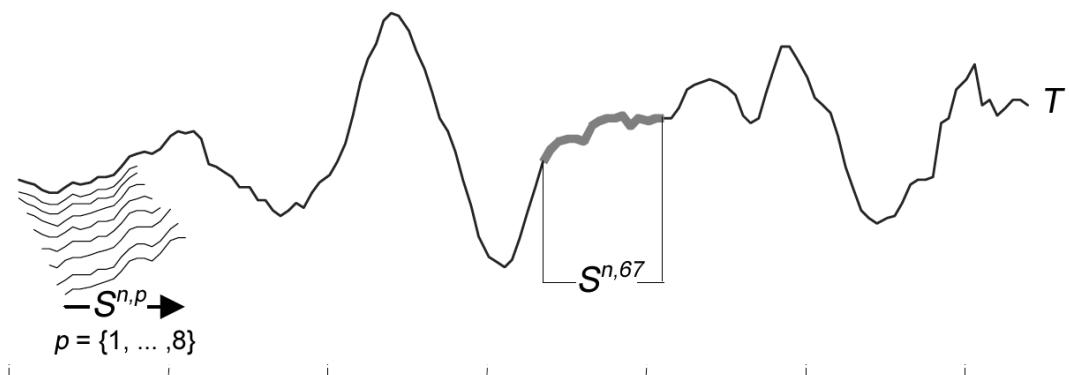


Fonte: Kepler and K2 Science Center, 2018.

Para compreender este trabalho é necessário formalizar alguns conceitos, como séries temporais, subsequências, aprendizado de máquina e gráficos de recorrências. Uma série temporal é um conjunto de m variáveis dadas por $T = t_1, t_2, \dots, t_m$ que são organizados por ordem temporal e espaçadas em intervalos de tempo iguais. Uma subsequência pode ser definida em uma série T de tamanho m como uma subsequência S de comprimento $l \leq m$ de posições contíguas de T , isto é, $S = t_p, \dots, t_{p+l-1}$, para $1 \leq p \leq m - l + 1$.

Um dos processos para se obter uma subsequência de tamanho l em determinada posição p de uma série temporal T é a aplicação de uma janela deslizante, como está ilustrado na Figura 1, onde observamos oito subsequências extraídas das posições $p = \{1, 2, \dots, 8\}$ e ainda uma subsequência extraída da posição $p = 67$.

Figura 2 – Processo de aplicação de uma janela deslizante



Fonte: Keogh, 2003.

Para identificar os dados desejados precisamos buscar os padrões dessas curvas T e a partir desses padrões reconhecer curvas, que representam exoplanetas para qualquer base de dados existente. Para cumprir com esse objetivo existem métodos como a técnica de Doppler, que é baseada na observação do trânsito do planeta em torno de sua estrela, a técnica de análise da velocidade radial, a do tempo de eclipses, entre outros. Algumas dessas técnicas geram curvas de fácil reconhecimento visual, porém em nossos dados não temos essa facilidade devido a grande quantidade de informações e de ruídos, é por isso que utilizamos os métodos baseados em aprendizado de máquina. O aprendizado de máquina consiste na construção de um modelo a partir de uma base de dados previamente conhecida que possibilite a identificação/classificação de novos dados automaticamente, sem necessidade de interferência humana durante o processo. Utilizamos alguns dos principais algoritmos de aprendizado de máquina para realizar nossos experimentos, tal como o Decision Trees, o Support Vector Machines, o Naive Bayes, o Nearest Neighbors e o Neural Network. Sendo o mais conhecido deles o Decision Trees, que é um método supervisionado (quando é utilizado um agente externo que indica à rede a resposta desejada para o padrão de entrada) que tem como objetivo criar um modelo que prevê o valor de uma variável de destino, aprendendo regras de decisão simples inferidas a partir de dados de treino, quanto mais profunda for a árvore, mais complexa será a decisão. Apresenta vantagens como sua simplicidade e fácil interpretação, afinal as árvores podem ser visualizadas esquematicamente. Diferente de outros algoritmos, não é obrigatório que seja feita a normalização dos dados e é um modelo de caixa branca, ou seja, a explicação para as classificações realizadas pode ser facilmente compreendida pela lógica booleana. Também utilizamos o Support Vector Machines, o qual tem por objetivo encontrar uma linha de separação, também denominada de hiperplano, entre dados de duas ou mais classes. Essa linha busca maximizar a distância entre os pontos mais próximos em relação a cada uma das classes, essa distância entre o hiperplano e o primeiro ponto de cada classe costuma ser chamada de margem, definindo assim cada ponto pertencente a cada uma das classes, e em seguida maximiza a margem. Ou seja ela primeiro classifica as classes corretamente e depois em função dessa restrição define a distância entre as margens. Portanto, esse algoritmo é melhor aplicável em espaços dimensionais elevados e apresenta possibilidades para a adição de parâmetros que auxiliam na classificação. Já o

algoritmo Naive Bayes é baseado na aplicação do teorema de Bayes. Apresenta vantagens pois exige uma pequena quantidade de dados de treinamento para estimar os parâmetros necessários e porque pode ser extremamente rápido em comparação com métodos mais sofisticados.

O Nearest Neighbors oferece métodos supervisionados e não supervisionados, o princípio por trás deste algoritmo é encontrar um número de amostras de treinamento mais próximas da distância de um novo ponto e prever a classe a partir delas. O número de amostras pode ser uma constante definida pelo usuário ou variar com base na densidade local de pontos, que é o aprendizado baseado no raio. A distância pode, em geral, ser qualquer medida métrica, onde a distância euclidiana é a escolha mais comum. Apesar de sua simplicidade, o Nearest Neighbors teve sucesso em um grande número de problemas de classificação e regressão, como em imagens de satélites e por muitas vezes foi bem sucedido em situações de classificação onde o limite de decisão é muito irregular.

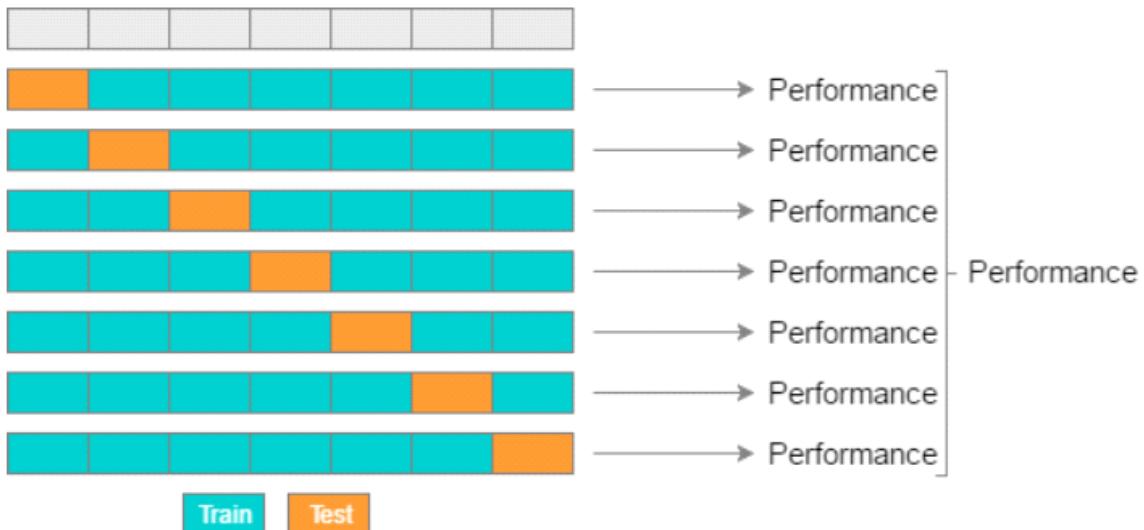
E por fim, o algoritmo Neural Network, que representa [modelos](#) computacionais inspirados pelo [sistema nervoso central](#) biológico, os quais são capazes de realizar o [aprendizado de máquina](#) bem como o [reconhecimento de padrões](#). Seus elementos de processamento são neurônios que produzem a soma ponderada das entradas e aplicam o resultado a uma função de transferência não linear, para gerar uma saída. Este modelo têm se mostrado adequado para o reconhecimento de padrões em reconhecimento de voz, biometria e sensoriamento remoto, no entanto ele não faz isto de forma explícita. As redes neurais são o melhor exemplo de um sistema sub simbólico, onde mesmo que cada parte de um padrão apresentado a uma rede tenha um significado explícito, associável a um símbolo de nosso modelo do mundo real, a representação interna dos dados no processador (a rede) não é explícita e não possui significado inteligível. Um método sub simbólico tipicamente é incapaz de explicar porque chegou a uma determinada conclusão, uma vez que um mapeamento explícito de causa-e-efeito não existe (Aldo von Wangenheim, 2015, p. 1).

Neste trabalho utilizamos todos esses algoritmos em combinação com os dados gerados pelo processamento de recorrências. Esse processamento consiste da utilização de um gráfico projetado para localizar padrões recorrentes que sejam aparentemente ocultos para o observador e pode ser explicado a partir do teorema de Takens. Pelo teorema, podemos recriar

uma imagem topologicamente equivalente do comportamento do sistema multidimensional original usando a série temporal de uma única variável observável, ou seja para a série x_i , construímos vetores do tipo $x_i^m = (x_i, x_{i+d}, x_{i+2d}, \dots, x_{i+(m-1)d})$, onde m é a dimensão de incorporação e d é o atraso de tempo. Em seguida, uma matriz simétrica de distâncias pode ser construída calculando distâncias entre todos os pares de vetores embutidos; o gráfico de recorrência relaciona cada distância de tal matriz a uma cor, assim o gráfico de recorrência é um gráfico retangular sólido que consiste em pixels cujas cores correspondem à magnitude dos valores dos dados em uma matriz bidimensional e cujas coordenadas correspondem às localizações dos valores de dados na matriz (Belaire-franch, 2002).

Outros conceitos que merecem atenção são os de normalização e o de validação cruzada, os quais são fundamentais para cumprir com os objetivos deste trabalho. A normalização é um processo onde os dados são ajustados de modo que todos os seus valores pertençam a um determinado intervalo, o que permite deixar todos os dados em uma mesma escala e garantir que podemos realizar uma comparação direta entre seus valores. Não existe apenas uma técnica de normalização, mas para a realização deste trabalho escolhemos a normalização-Z, nessa técnica os valores das séries temporais são ajustados de modo que a média de seus valores seja nula e que o desvio padrão seja unitário. O formato da série temporal original é conservado e para o cálculo dos valores normalizados é necessário conhecer o valor médio e o desvio padrão dos dados originais. A validação cruzada é utilizada para criar uma série de combinações durante o experimento, ou seja, a tabela de dados é dividida em blocos, que tem sua quantidade predeterminada de acordo com a necessidade de cada algoritmo. Com isso, os blocos são classificados como sendo dados de teste e como dados de treino, feito isso são realizados os experimentos e é gerado um resultado percentual de acertos na classificação dos dados para cada uma das combinações possíveis. Desse modo após todas as possibilidades terem sido esgotadas pode ser feita uma média geral e uma comparação entre todos os resultados apresentados. O principal objetivo da avaliação por meio da validação cruzada consiste na tentativa de atenuar algum eventual viés que possa existir amostra de dados analisada. Na Figura 3 é apresentada uma representação esquemática do processo de validação cruzada.

Figura 3 – Processo de aplicação da validação cruzada



Fonte: Eduard Bonada i Cruells, Cross-Validation Strategies.

202

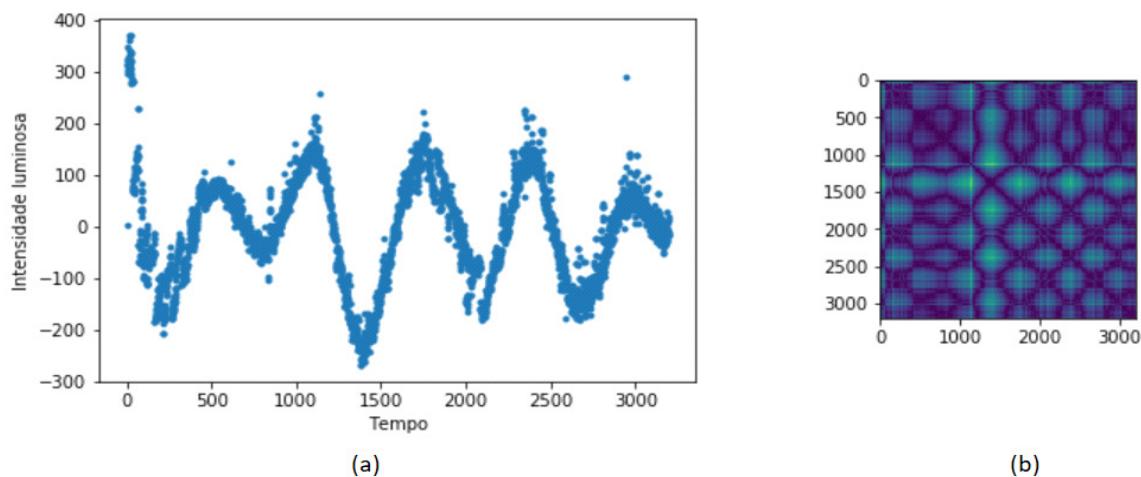
Método

Neste trabalho analisamos as curvas de luz provenientes da base de dados do Kaggle (www.kaggle.com), que consiste em uma plataforma para hospedar competições de Data Science públicas, privadas e acadêmicas. Também existem competições de aprendizado disponibilizadas pelo próprio Kaggle ou por empresas para treinamento de habilidades. A plataforma também armazena e disponibiliza dados sobre assuntos diversos, os chamados datasets; também possui fóruns para troca de conhecimentos entre seus usuários. Por este meio obtemos as curvas de luz de estrelas do telescópio Kepler, as quais foram fornecidas em formato CSV. No conjunto de dados com o qual escolhemos trabalhar cada estrela tem um rótulo binário de 0 ou 1, o 0 indica que a estrela está confirmada para ter pelo menos um exoplaneta em sua órbita e o 1 indica que aquela estrela não possui exoplanetas. Nossos dados são divididos em duas categorias, a de treino do algoritmo de aprendizagem de máquina e a de teste do algoritmo. O conjunto de treino é composto por trinta e sete estrelas confirmadas com exoplanetas e cinco mil e cinquenta estrelas sem exoplanetas, enquanto o conjunto de teste possui cinco estrelas com exoplanetas confirmadas e quinhentas e sessenta e cinco estrelas

sem exoplanetas. Unificando os conjuntos de treino e de teste do Kaggle temos uma base de dados composta por 42 estrelas confirmadas com exoplanetas e 5615 estrelas sem exoplanetas, sendo que cada uma destas curvas possui um total de 3197 registros. No entanto, devido à alta dimensionalidade dos dados, para realizar os experimentos utilizamos as 42 curvas de luz de estrelas com exoplanetas e apenas 158 curvas de não exoplanetas.

Os dados das curvas de luz foram analisados considerando duas abordagens: na primeira pela utilização direta dos dados brutos; e na segunda pela aplicação da estratégia de gráficos de recorrências. Para o desenvolvimento da segunda abordagem tomamos os dados originais e aplicamos a função de análise de recorrência. Esse processo nos fornece uma matriz de recorrência para cada curva de luz, as quais podemos visualizar por meio dos exemplos representados na Figura 4.

Figura 4 – (a) exemplo de curva de luz; (b) gráfico de recorrência gerado



Fonte: autoria própria, 2019.

O processo descrito acima foi realizado mediante a utilização da biblioteca sklearn (Scikit-Learn para Python), a qual reúne vários métodos de algoritmos de [classificação](#), [regressão](#) e [agrupamento](#) de dados.

A biblioteca Pandas foi utilizada para a coleta e preparação de dados, pois possui ferramentas para ler e gravar dados na memória em diferentes formatos (arquivos CSV e de texto); permite que colunas possam ser inseridas e excluídas de estruturas de dados para

mutação de tamanho; e possibilita operações de combinação e divisão em conjuntos de dados. Essas funções são extremamente úteis na manipulação de dados como os das curvas de luz, onde a representação é feita em matrizes e a quantidade de informações é muito grande. O pacote NumPy é fundamental para programação em Python, dado sua sofisticada técnica para lidar com dados N-dimensionais; suas ferramentas para integrar código C/C++ e Fortran; suas funções em álgebra linear, transformada de Fourier e capacidade de gerar números aleatórios. Além disso, o NumPy também pode ser usado como um contêiner multidimensional eficiente de dados genéricos, isso permite que o NumPy integre-se de forma fácil e rápida a uma ampla variedade de bancos de dados.

Para que os experimentos com os gráficos de recorrências fossem realizados, precisamos contar com o desenvolvimento de códigos em Python através do Jupyter Notebook (meio para desenvolver software de código aberto, padrões abertos e serviços para computação interativa em dezenas de linguagens de programação). Desenvolvemos também um programa computacional com os algoritmos de aprendizagem de máquina, nos quais estão incluídos algoritmos como o Decision Trees, Support Vector Machines, Naive Bayes, Nearest Neighbors e Neural Network.

204

Como citado anteriormente, a análise do desempenho dos algoritmos de classificação foi realizada por meio da biblioteca de validação cruzada. Como medida de desempenho utilizamos o escore F1, que pode ser interpretado como uma média ponderada da *precision* e da *recall* (métricas de precisão), em que um escore F1 alcança seu melhor valor em 1 e o pior escore em 0. A equação para a pontuação de F1 é:

(1)

Para a execução desses códigos contamos com a utilização do Cluster C3HPC (DINF-UFPR), que possui seis nodos de processamento, cada um com quatro sockets 3.30GHz (oito núcleos por socket) e 256 GB de RAM executamos o código desenvolvido. Escolhemos executar os experimentos no Cluster C3HPC pois este é um sistema capaz de combinar vários computadores para trabalharem em conjunto, onde os computadores ficam integrados e atuam conjuntamente no processamento de dados e execução de tarefas complexas, que exigem

muitos processadores. Com os resultados obtidos para cada algoritmo, os comparamos e definimos qual apresentou o resultado mais satisfatório na classificação de estrelas com exoplanetas e sem exoplanetas.

Resultados

Os algoritmos de aprendizado de máquina foram aplicados (1) para os dados originais das séries temporais e (2) para as matrizes obtidas a partir das análises de recorrência e seus resultados estão apresentados na Tabela 1, por meio da métrica F1. Observamos melhores resultados nos dados classificados com a aplicação da análise de recorrência, com exceção do algoritmo nearest neighbors, para o qual em ambos os casos apresentou médias muito similares. Por outro lado, podemos observar significativa melhora nos resultados obtidos com os algoritmos de decision tree e naive bayes.

Tabela 1: Resultados com F1 score.

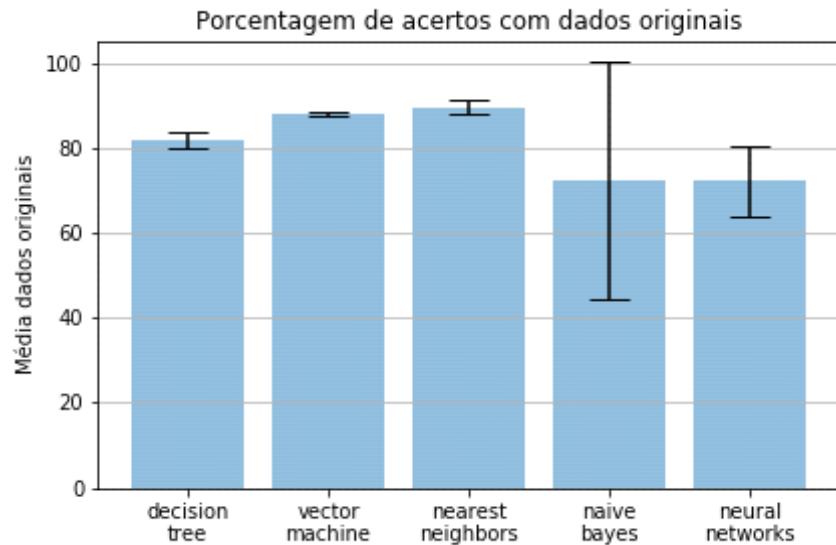
205

Algoritmos	Médias dados originais (1)	Médias análise de recorrência(2)
decision tree	$81,78\% \pm 1,85\%$	$91,67\% \pm 2,32\%$
vector machine	$88,27\% \pm 0,50\%$	$89,02\% \pm 0,89\%$
nearest neighbors	$89,73\% \pm 1,58\%$	$89,67\% \pm 3,60\%$
naive bayes	$72,37\% \pm 27,93\%$	$98,14\% \pm 1,51\%$
neural networks	$72,26\% \pm 8,30\%$	$88,29\% \pm 1,23\%$

Fonte: autoria própria, 2019.

Na Figura 5 e na Figura 6 podemos visualizar os mesmos resultados acima porém de uma maneira mais intuitiva, ficando mais clara a performance de cada um dos algoritmos de aprendizado de máquina em cada um dos dois casos apresentados.

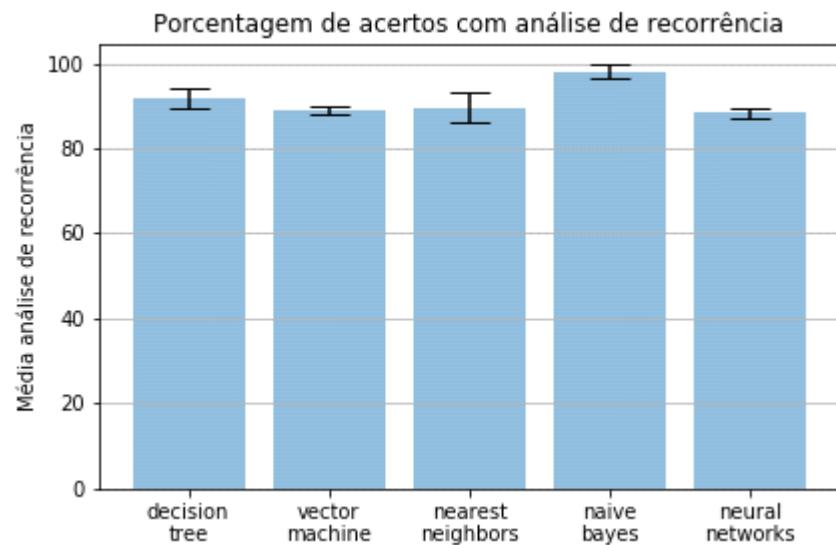
Figura 5 – Gráfico de barras com barra de desvio padrão para dados originais.



Fonte: autoria própria, 2019.

206

Figura 6 – Gráfico de barras com barra de desvio padrão para dados com análise de recorrência.



Fonte: autoria própria, 2019.

Conclusões

Com base nos resultados apresentados neste trabalho percebemos em alguns algoritmos uma proporção maior de acertos para os dados com análise de recorrência, o que nos permite concluir, pela análise da métrica F1, que essa técnica possibilitou melhoria no processo de classificação.

Quando analisamos a tabela obtida como resultado deste trabalho percebemos que o algoritmo nearest neighbors apresentou a maior porcentagem de acertos e o menor desvio padrão na análise realizada com os dados originais, enquanto que o algoritmo naive bayes e neural networks apresentaram médias semelhantes porém com desvios padrões bem diferentes, onde o neural networks se apresentou mais preciso e o naive bayes apresentou o maior desvio padrão entre todos os cinco algoritmos utilizados. Já nos resultados obtidos a partir da aplicação da análise de recorrência podemos observar que a melhor média foi do algoritmo naive bayes e a pior do support vector machine, entretanto a diferença entre ambas é pequena, menos de dez por cento, mostrando assim como a aplicação da técnica de gráficos de recorrência de fato acrescentou positivamente no processo de classificação dos algoritmos de aprendizado de máquina.

Porém, acreditamos que estes resultados poderiam ser ainda melhores, o que não foi possível por motivos como a baixa amostragem de objetos que eram estrelas com exoplanetas. Outro fator a ser considerado é que devido a limitação computacional não foi possível utilizar cem por cento da base de dados existentes nos experimentos realizados, o que com certeza trouxe consequências para o resultado final.

Em projetos futuros pretendemos utilizar bases de dados com uma menor divergência na quantidade de itens e ainda dados em que tenhamos a certeza da existência de apenas duas classes, que é o tipo de análise que determinamos para os algoritmos de aprendizado de máquina que utilizamos neste trabalho. Sabemos que algumas das estrelas possuem exoplanetas, mas isso não pode nos dar a certeza de que as estrelas classificadas como as que não possuem exoplanetas apresentam todas as mesmas características em suas curvas de luz. Essa reflexão baseia-se no fato de que apesar de ser uma única classe de estrelas sem exoplanetas, ainda podem existir várias subclasses que desconhecemos dentro desta classe de

estrelas, contendo vários outros tipos de objetos astronômicos que não conhecemos ou que não conseguimos detectar em suas órbitas.

Referências

ZALEWSKI, Willian. **Modelagem Simbólica de Padrões Morfológicos para a Classificação de Séries Temporais**. Curitiba, PR, p. 55-58, 2015.

MITCHELL, T. M. **Machine Learning**. Boston, USA: McGraw-Hill, 1997.

The UCR Matrix Profile Page. Disponível em: <http://www.cs.ucr.edu/~eamonn/MatrixProfile.html>. Acesso em: 2018.

CASTRILLÓN, J. P. B. **Análise de Curvas de Luz do Corot usando diferentes processos comparativos:** estimando períodos de rotação estelar. UFRN, 2010.

AMARAL, Fernando. **Introdução à Ciência de Dados:** mineração de dados e big data. Alta Books Editora, 2016.

REZENDE, Solange Oliveira. **Sistemas Inteligentes:** fundamentos e aplicações. Editora Manole Ltda, 2003.

BELAIRE-FRANCH, Jorge. CONTRERAS, Dulce. **Recurrence Plots in Nonlinear Time Series Analysis:** Free Software. Dept. of Economic Analysis University of Valencia, 2002.