ANÁLISE DE CURVAS DE LUZ OBTIDAS COM OS TELESCÓPIOS COROT E KEPLER COM TÉCNICAS DE MACHINE LEARNING

Leonardo B. Rodrigues[1], Roberto B. Menezes[2]

[1] Aluno de Iniciação Científica da Escola de Engenharia Mauá (EEM/CEUN-IMT);
[2] Professor da Escola de Engenharia Mauá (EEM/CEUN-IMT).

Resumo. *Este trabalho explora um fluxo produtivo para a análise de dados de curva de luz obtidos pelo satélite CoRoT, levando em consideração os requerimentos computacionais para obter-se melhor performance e produtividade na aplicação de modelos de aprendizado de máquina. Alguns modelos são analisados quanto sua taxa de assertividade para as bases de dados analisadas. Conclui-se que para pequenas quantidades de dados não é necessário um trabalho estrutural especializado de Big Data.*

Introdução

O Satélite CoRoT[i] (**Co**nvection, **Ro**tation and planetary **T**ransits – Auvergne et al. 2009) operou durante o período de 2006 a 2013, com o objetivo duplo de procurar planetas fora do sistema solar, referidos como e*xoplanetas,* com características similares à Terra, e realizar estudos a respeito da oscilação de estrelas. O satélite fez uso de câmeras cientificas[ii] CCD de ângulo amplo, similares às implementadas no telescópio Hubble, para coletar a luz de grandes quantidades de estrelas, ao longo de certos períodos, a fim de detectar variações de brilho que pudessem estar associadas à presença de exoplanetas.

As variações de brilho detectadas pelas câmeras podem ser causadas por processos internos ou pela ocorrência de fenômenos externos, tal qual um eclipse, que se dá quando outro corpo celeste em órbita da estrela se posiciona entre esta e o observador. Esse corpo orbitante causador do eclipse pode ser outra estrela, o que caracteriza o que é chamado de uma *binária eclipsante*, ou um exoplaneta.

Para acomodar os programas de astrosismologia e de busca de exoplanetas, a missão realizou uma série de observações, em execuções (*observing runs*) com durações[iii] de 20 dias (curto prazo) e 150 dias (longo prazo). Cada execução, com seus dados coletados, é referenciada por um código único (IRa01, LRa01 etc.) disponível nos diretórios da ESA.

Os sinais coletados foram organizados na forma de curvas de luz, que correspondem a gráficos do brilho observado das estrelas em função do tempo. As curvas de luz são capazes de revelar a presença dos eclipses, que, após uma análise cuidadosa, podem ser classificadas como: curvas com eclipses causados por exoplanetas ou curvas com eclipses causados por binárias eclipsantes. Também é possível identificar as curvas sem a presença de eclipses.

Esse tipo de estudo é dificultado pelo grande volume de dados obtidos pelo telescópio, tornando indispensáveis procedimentos automáticos de análise para diferenciar e classificar curvas de luz. Tais procedimentos podem ser referidos, de forma genérica, como aprendizado de máquina, um apanhado de métodos estatísticos e algorítmicos que melhoram em performance mediante o uso de dados.

Esse trabalho tem como principais objetivos:

I. Justificar requerimentos computacionais para otimizar o trabalho do pesquisador com alto volume de dados.
II. Validar e implementar filtros e modelos de classificação de curvas de luz.
III. Implementar um procedimento automático de análise de curvas de luz.

Fazendo uso dos dados abertos disponibilizados pela missão do CoRoT[iv], uma série de estudos foram realizados para a implementação destes objetivos.

Runs	AN2_FULLIMAGE	AN2_POINTING	AN2_STAR and AN2_WINDESCRIPTOR	EN2_FULLIMAGE	EN2_STAR, EN2_WINDESCRIPTOR and EN2_STAR_IMAG
IRa01***	here (Size : 17M)	!! NEW VERSION** !! here (Size : 826M)		here (Size : 186M)	!! NEW VERSION** !! here (Size : 12G)
SRc01***	here (Size : 46M)	!! NEW VERSION** !! here (Size : 345M)		here (Size : 370M)	!! NEW VERSION** !! here (Size : 3.7G)
LRc01	here (Size : 19M)	!! NEW VERSION* !! here (Size : 1.7G)	here (Size : 264M)	here (Size : 184M)	here (Size : 36G)

Figura 1 – Exemplo dos dados de observações de cada período, no site oficial da missão.

Materiais e Métodos

A base técnica deste trabalho foi desenvolvida com linguagem de programação Python e bibliotecas de ferramentas desenvolvidas para este ambiente, tal qual o pacote Pandas, que oferece estruturas de dados e ferramentas próprias para a utilização e manipulação de séries temporais, tabelas numéricas e grandes quantidades de dados. A estrutura principal do Pandas, o *Data Frame*, pode ser pensado como uma matriz, com a ressalva de que suas colunas podem ter nomes próprios. Cada linha corresponde a uma observação e cada coluna representa uma variável discriminada.

O ambiente interativo *Jupyter* foi utilizado para testes e desenvolvimento de análises, por oferecer flexibilidade e uma vasta gama de ferramentas próprias para ciências de dados, computação cientifica e aprendizado de máquina, além de ser um ambiente modular e personalizável.

Os arquivos contendo os dados de curvas de luz foram obtidos do site oficial da missão CoRoT. Cada passada de observação (*observing run*) realizada pelo satélite é referenciada por um código e contém 5 bancos de dados, conforme a tabela 1. Para os propósitos deste trabalho, utilizaram-se apenas os bancos de dados referentes a curvas de luz, com atenção especial às bases da categoria EN2_STAR, a fim de testar a eficácia de detecção dos modelos aplicados.

Os dados foram divididos e discriminados conforme o Catálogo de Trânsito do CoRoT[v] (Deleuil et. al - 2018).

Tabela 1 – Bancos de dados de observações do satélite CoRoT

Canal de estrelas brilhantes (AN2_*)	Canal de estrelas de luz fraca (EN2_*)
Imagem Completa - AN2_FULLIMAGE	Imagem Completa – EN2_FULLIMAGE
Arquivo de Apontamento – AN2_POINTING	-
Curvas de Luz – AN2_STAR	Curvas de Luz – EN2_STAR

Os arquivos de curvas de luz são disponibilizados em formato FITS (*Flexible Image Transport System*), elaborado especificamente para o armazenamento, transmissão e processamento de dados, sendo o formato de arquivo de uso comum na comunidade cientifica internacional de astronomia. Apesar de bibliotecas especializadas para uso e processamento de arquivos FITS estarem disponíveis para a linguagem Python, optou-se por transformar os arquivos para formatos mais eficientes e de uso comum em comunidades de ciência de dados. A seção a seguir detalha a metodologia e fundamentos para a escolha do arquivo de armazenamento.

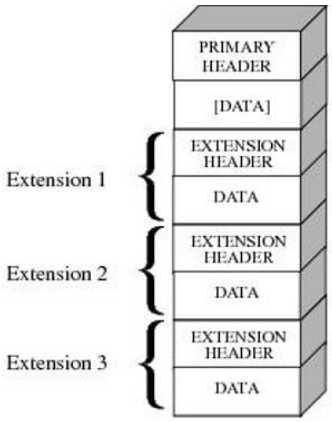


Figura 2 – Arquitetura do formato FITS

Armazenamento de dados para utilização em Pandas

Para elaboração de uma comparativa de desempenho foram identificados padrões de armazenamento comumente utilizados pela comunidade de cientistas de dados.

I. **CSV** – Formatação de dados em texto puro
II. **Pickle** – Formato nativo da linguagem Python para serialização de objetos e dados.
III. **HDF5** – Hierarchical Data Format 5, formatação para grandes quantidades de dados com hierarquias bem definidas, estruturados de forma similar a diretórios de um sistema operacional.
IV. **Parquet** – Formatação de dados em estruturas tabulares, desenvolvido para uso no ecossistema de análise de dados Hadoop, da Apache.
V. **MessagePack** – Serialização binária implementada de forma similar ao formato JSON

A fim de comparar os formatos em sua performance para análise das bases do CoRoT é útil estabelecer uma série de métricas para análise.

- Tempo Salvar (TS) – o tempo necessário para salvar o Data Frame com os dados, em segundos.
- Tempo Carrega (TC) – o tempo necessário para carregar o Data Frame com os dados, em segundos
- Salvar Consumo Memória (SCM) – a memória utilizada para salvar o Data Frame, em Mb.
- Carregar Consumo Memória (CCM) – a memória utilizada para carregar o Data Frame, em Mb.

Para cada um dos formatos estudados, foi gerado um número arbitrário de tabelas formatadas com a dimensionalidade dos dados baixados das bases do site oficial do CoRoT.

Em seguida, os arquivos foram carregados, abertos e salvos sucessivas vezes no formato objetivo por meio de um processo automatizado. O tempo gasto por cada operação, além da taxa de crescimento do uso de memória RAM, são anotados e colocados em gráficos para comparação, apresentados na última sessão deste artigo. O formato CSV foi utilizado como padrão de comparação, por representar a maneira mais rudimentar de se representar dados, sem qualquer otimização ou organização adicional.

Aplicação de aprendizado de máquina (ML)

Independentemente do modelo aplicado, o fluxo de trabalho manteve o padrão descrito na figura abaixo. Os dados foram baixados, pré-processados, passaram pela seleção de features e em seguida foram aplicados ao modelo escolhido.

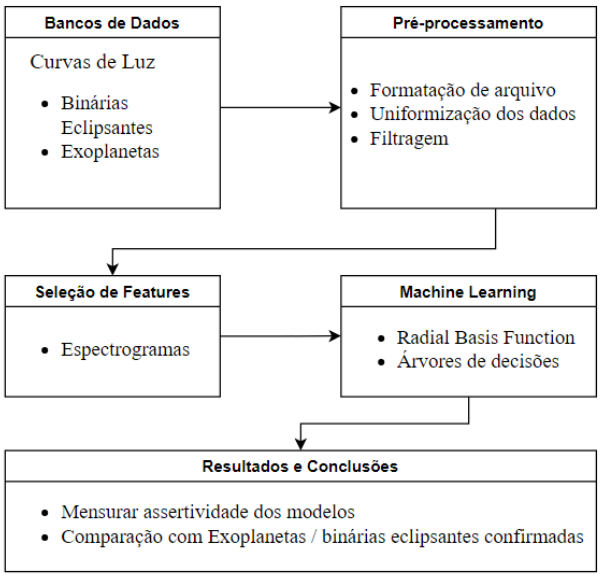


Figura 3 – Fluxograma de implementação do trabalho

I. Pré-processamento

Após os dados serem baixados, o trabalho passou a ser realizado no ambiente Jupyter, no qual os arquivos FITS foram abertos (com ajuda da biblioteca *astropy*) e passaram pelas etapas de pré-processamento. Fez-se necessário remover ruídos oriundos da instrumentação para que os dados fossem analisados de forma fidedigna

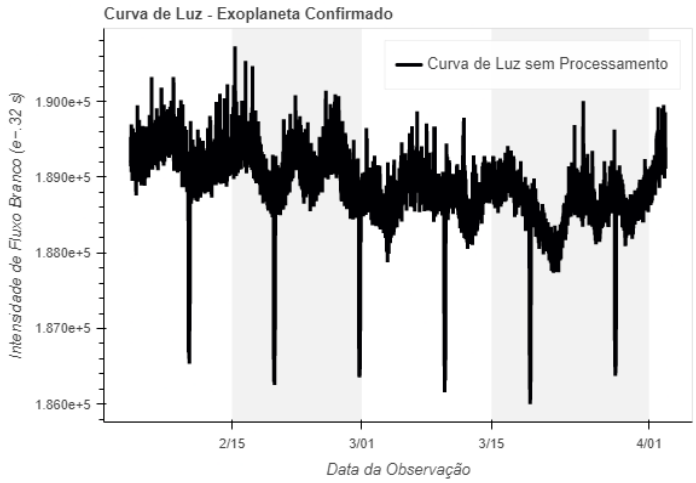


Figura 4 – Visualização de uma Curva de Luz

i. Amostragem temporal

Foi necessário garantir que o conjunto de curvas analisadas por observação apresentem uniformidade de amostragem em seus dados. É realizada uma análise na distribuição diferença

$$t[k] - t[k-1]$$

de todas as curvas de luz do banco de dados, no qual k representa cada pedaço da amostra e t o tempo da amostra. Curvas com grande diferença do padrão médio aferido (*Outliers*) foram removidas da base. Em seguida, os dados foram reamostrados a fim de uniformizar o formato de todas as curvas da base.

ii. Filtragem

Uma vez que as curvas tiveram suas amostragens temporais normalizadas, um filtro foi aplicado a fim de remover ruídos aleatórios que estão compondo o sinal. Para os fins deste trabalho o filtro de Butterworth foi escolhido. Maiores detalhes a respeito de características e aplicações do mesmo vão além do escopo aqui contemplado, bastando dizer que se trata de um filtro passa-baixa e que este apresenta uma resposta de frequência muito plana em sua banda passante, aproximando-se do ideal.

O filtro é representado matematicamente por:

$$G(\omega) = \frac{1}{\sqrt{1 + \ \omega^{2n}}}$$

no qual ω é a frequência angular em rad.s^{-1} e n representa a ordem de filtragem.

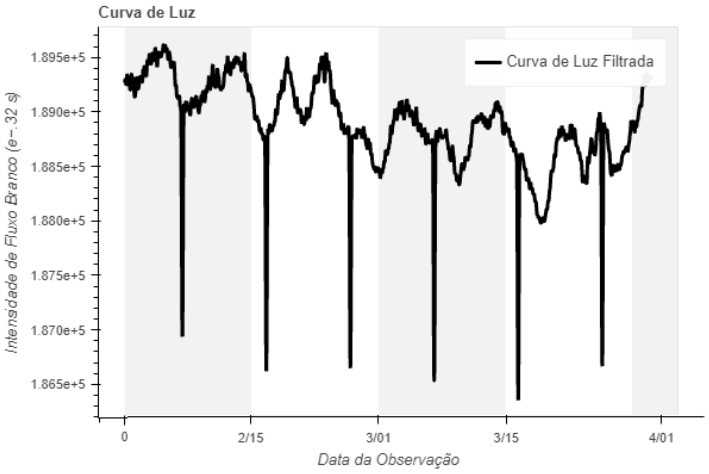


Figura 5 – Visualização de uma curva de luz filtrada

iii. Salvar arquivos

Feito o pré-processamento, os dados foram salvos em formato Pickle como "processados.plk", estando prontos para processamento com os modelos.

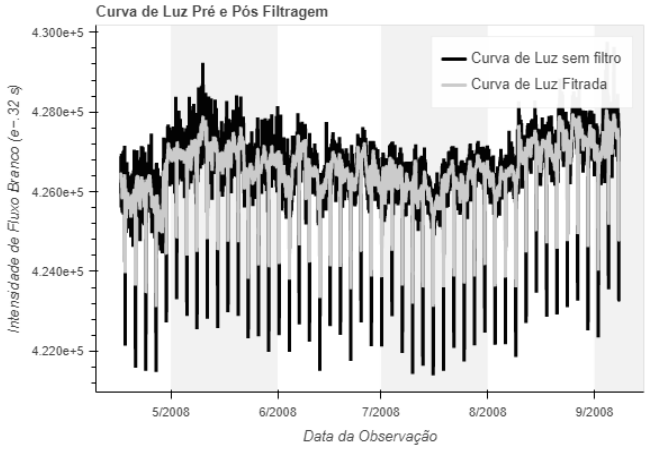


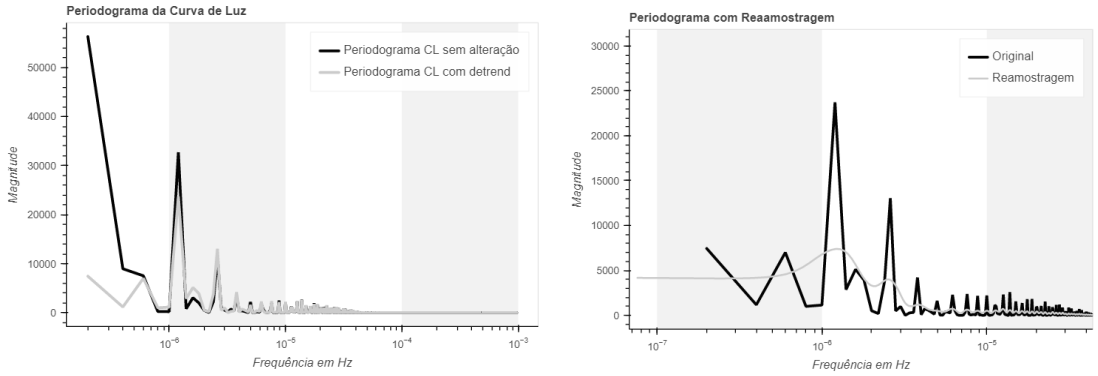
Figura 6 – Exemplo de uma curva de luz pós e pré filtragem.

II. Seleção de features

Para a aplicação de modelos de aprendizado supervisionados, é necessário escolher uma característica mensurável ("*feature*") do fenômeno estudado. Para curvas de luz, é proveitoso fazer uso de periodogramas, que estimam a densidade espectral do sinal e podem ser utilizados como método para extrair informações.

Utilizando as ferramentas de processamento de sinais da biblioteca *scipy* do Python, é possível criar periodogramas das séries filtradas das curvas de luz. Primeiramente é proveitoso retirar as faixas que não contém informação útil no periodograma, por um processo de "*detrend*" do sinal das curvas de luz, que representa a retirada do valor 'tendência' do sinal (seu valor médio).

Por fim, deve-se notar que cada curva de luz apresenta um periodograma com resoluções diferentes, por motivo de cada observação possuir taxas e janelas de amostragem diferentes. É necessário então realizar uma reamostragem dos periodogramas a fim de definir um padrão único a todas as observações.



Figuras 7 e 8 – Periodograma da Curva de luz após *detrend* e após reamostragem

III. Aplicação dos modelos

Após o pré-processamento dos dados, estes podem ser utilizados para aplicação de modelos distintos. Dois modelos foram testados neste trabalho, Árvores de decisão e o XGBoost. O primeiro modelo foi escolhido por sua simplicidade operacional e como base de comparação para performance do XGboost, que pode ser descrito como uma composição de outras técnicas construídas em cima da base de uma árvore de decisão. A comparação permite melhor ilustrar o impacto de modelos em aferir resultados partindo-se de uma mesma base de dados.

A aplicação de ambos foi realizada por meio da biblioteca *scikit-learn* e da biblioteca *xgboost*, e ambos são descritos em maiores detalhes a seguir.

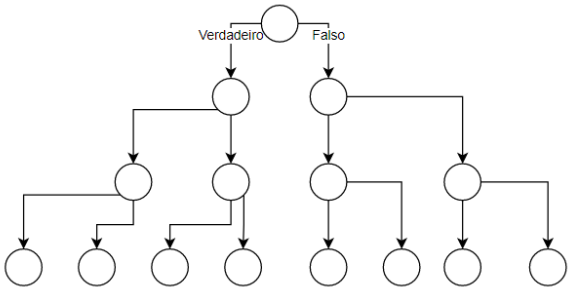


Figura 9 – Estrutura de uma Árvore de Decisão

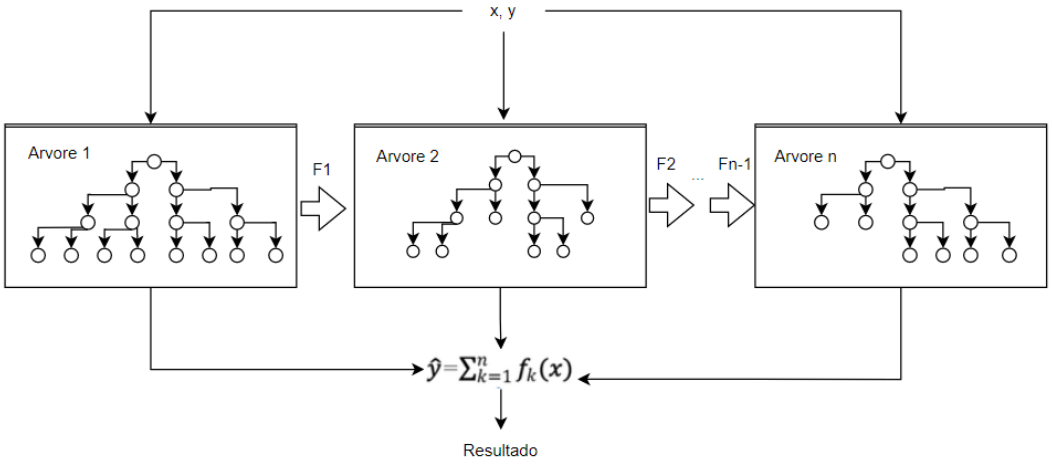


Figura 10 – Estrutura do XGBoost

O primeiro modelo utiliza árvores de decisão (*Decision Trees* ou DTs) preditivas para classificar as curvas em binárias eclipsantes ou exoplanetas. DTs são uma maneira dinâmica de aplicar diferentes métodos estatísticos para separar dados por suas semelhanças ou diferenças. Para este trabalho, o modelo DT utilizava medição de entropia entre dados para discriminar categorias. Em outras palavras, o modelo busca discriminar o quão homogêneos os dados são, associando um valor de 0 a 1 para representar o grau de "desordem", no qual 0 representa baixa desordem e 1 representa desordem máxima. Além disso o modelo faz uso do cálculo de ganho de informação, no qual a entropia de dois estados é comparada para medir a redução de incerteza da análise de um item do banco de dados para outro. Matematicamente isso é representado por:

(I) $E(S) = \sum_{i=1}^{c} -p_i log_2 p_i$ $\qquad$ $(II)\ IG(X,Y) = E(Y) - E(Y|X)$

Medida de entropia (I) e Ganho de informação (II)

O segundo modelo é conhecido por XGBoost (e**X**treme **G**radient **B**oosting), também um modelo de árvore de decisão. Em machine learning, o termo 'Boosting' se refere à combinação de múltiplos modelos para melhoria de precisão, no qual um modelo é aplicado e, subsequentemente, outro é usado para melhorar o anterior. O termo gradiente se refere à aplicação do modelo, no qual mudanças na predição do modelo inicial são observadas a fim de averiguar como afetam o erro. Mudanças que causam grandes reduções no erro recebem valores altos, mudanças que não afetam o erro recebem zero. O nome gradiente então se refere ao gradiente de erros gerado pelas medidas de mudança de erro das predições.

Primeiramente aplica-se um modelo de regressão (tal qual MMQ) aos dados. Se necessário, pode haver uma redução de dimensionalidade do resultado utilizando uma análise PCA. Em seguida, os dados são divididos em duas partes, para treino do modelo e teste. O modelo é treinado na primeira parte e subsequentemente validado com os dados de teste. O grau de assertividade é então aferido.

Resultados e Discussão

O formato Pickle apresenta boa performance relativa aos outros analisados, apesar de não apresentar a melhor do grupo. Entretanto, sua disponibilidade como pacote nativo da linguagem Python o coloca em uma posição vantajosa de uso, uma vez que não requer dependências externas para aplicação do fluxo de trabalho, simplificando o processo e diminuindo os requerimentos do sistema. Por tais motivos, foi escolhido como a forma de armazenamento para os dados deste estudo.

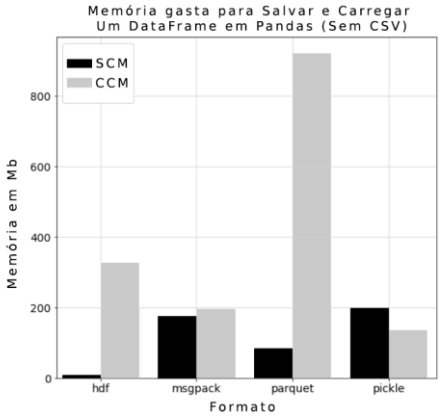
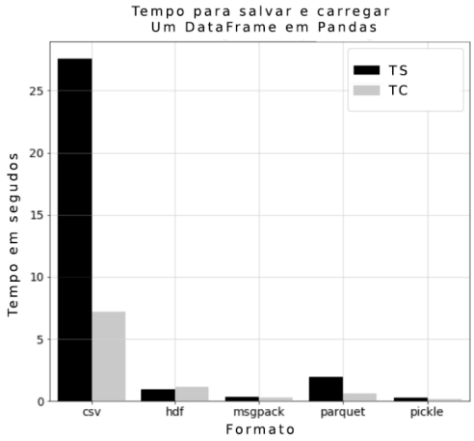
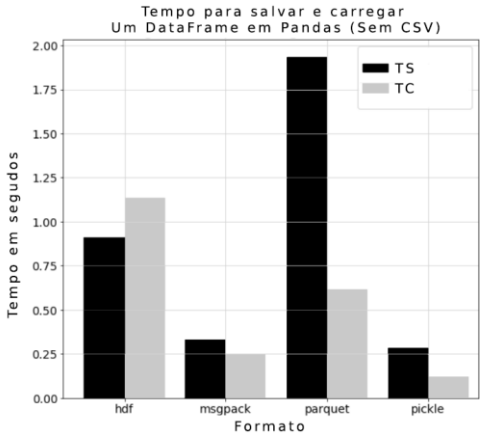


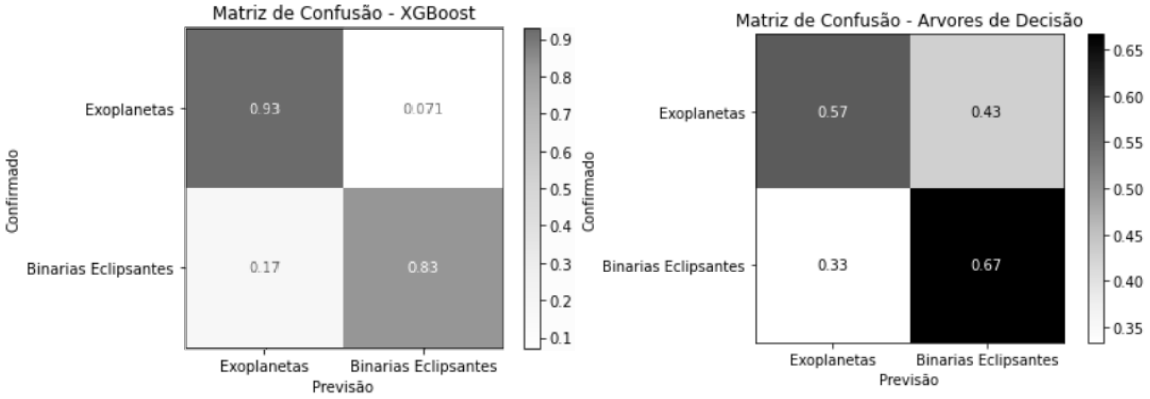
Figura 11 – Comparação de alocação de memória em diferentes formatos





Figuras 12 e 13 – Tempo gasto para carregar e salvar em diferentes formatos

Os modelos apresentam graus de assertividade em seus resultados com notáveis diferenças. O modelo de DTs possui baixa assertividade para identificação de exoplanetas e resultados um pouco melhores para validar binárias eclipsantes. Nota-se o fenômeno inverso na aplicação do XGBoost, no qual a assertividade é superior na identificação de exoplanetas, embora também muito elevada na identificação de binárias eclipsantes. É possível notar ainda como a composição de modelos, apesar de partir de uma base de implementação similar, apresenta ganhos significativos na assertividade de ambas as categorias nas bases treinadas. Melhorias posteriores poderiam ser encontradas com treinamento em mais dados, ajustes em hiper parâmetros de implementação e remoção de mais *outliers* no pré processamento dos dados.



Figuras 14 e 15 - Matriz de confusão com graus de assertividade dos modelos estudados

Conclusões

Observou-se que modelos com abordagens compostas como o *XGBoost* são mais efetivos para a classificação de dados relativamente uniformes do que métodos simplificados como DTs pautadas em métodos tradicionais como análise de entropia.

Para bancos de dados oriundos de um único satélite, a estrutura e ferramentas locais de uso comum para análise de dados foram adequadas para a aplicação de modelos e observação de resultados. Nota-se, entretanto, que o consumo de capacidade computacional demandaria acesso a ambientes especializados para *Big Data* caso o fluxo de trabalho aqui proposto fosse generalizado para outros satélites de outras missões com objetivos similares (como por exemplo o satélite Kepler) ou bancos de dados com quantidade muito elevada de dados.

Referências bibliográficas

Auvergne, M., Bodin, P., Boisnard, L., Buey, J. T., Chaintreuil, S., Epstein, G., ... & Zanatta, P. (2009). The CoRoT satellite in flight: description and performance. *Astronomy & Astrophysics*, *506*(1), 411-424.

Barge, P., Baglin, A., Auvergne, M., Rauer, H., Léger, A., Schneider, J., ... & Wuchterl, G. (2008). Transiting exoplanets from the CoRoT space mission-I. CoRoT-Exo-1b: a low-density short-period planet around a G0V star. *Astronomy & Astrophysics*, *482*(3), L17-L20.

Chaintreuil, S., Deru, A., Baudin, F., Ferrigno, A., Grolleau, E., & Romagnan, R. (2021). II. 4 The "ready to use" CoRoT data. In The CoRoT Legacy Book (pp. 61-108). EDP Sciences.

Deleuil, M., Aigrain, S., Moutou, C., Cabrera, J., Bouchy, F., Deeg, H. J., ... & Weingrill, J. (2018). Planets, candidates, and binaries from the CoRoT/Exoplanet programme-The CoRoT transit catalogue. *Astronomy & Astrophysics*, *619*, A97.

[i] Disponível em: https://corot.cnes.fr/en/COROT/index.htm/ Acesso em 26/11/2021.
[ii] Disponível em: https://sci.esa.int/web/hubble/-/34007-hubble-instruments?fbodylongid=1926 Acesso em 28/11/2021.

[iii] Disponível em: https://exoplanetarchive.ipac.caltech.edu/docs/datasethelp/ETSS_CoRoT.html Acesso em 27/11/2021.
[iv] Disponível em: http://idoc-corot.ias.u-psud.fr/sitools/client-user/COROT_N2_PUBLIC_DATA/project-index.html Acesso em 26/11/2021.
[v] Disponível em: http://cdsarc.u-strasbg.fr/viz-bin/qcat?J/A+A/619/A97 Acesso em 28/11/2021.