# Detecção de Explosões Solares

Ana Cunha, Luan Pires, Jeiel Santana, David Silva, Ícaro Souza, Centro de Informática - Universidade Federal de Pernambuco

## I. OBJETIVOS

Este trabalho tem por objetivo utilizar de diversos parâmetros desses eventos, como sua atividade, evolução e área para predizer a quantidade de cada uma das classes de explosões solares que irão ocorrer nas próximas 24 horas em uma dada região do sol, existindo um total de 3 classes, divididas em: classe C (comum), classe M (moderado) e classe X (severo), que variam de acordo com a produção de chamas da região em análise.

## II. JUSTIFICATIVA

As explosões solares são grandes explosões que ocorrem na superfície do Sol em razão das variações de seu campo magnético. Essas explosões liberam grande quantidade de energia na forma de radiação eletromagnética que podem danificar equipamentos eletrônicos, afetando dispositivos de comunicação na Terra e acarretando grandes danos econômicos.

Diante disso, faz-se necessário prever quando essas explosões solares podem ocorrer para evitar que avarias sejam causadas aos equipamentos de transmissão da informação. Assim, este trabalho visa utilizar métodos estatísticos para alcançar o objetivo proposto.

## III. METODOLOGIA

Para o projeto em questão, será necessário que façamos uma análise exploratória de uma série de dados para que a classificação das explosões solares e suas quantidades possam ser feitas corretamente.

Dessa maneira, levando em conta os parâmetros coletados, que dizem respeito às características dessas explosões e da sua complexidade ao longo do tempo, serão aplicados técnicas de aprendizagem de máquina para que a classificação seja, por fim, realizada. Portanto, a fim que tais técnicas sejam aplicadas, faremos uso do algoritmo classificador Naive Bayes por meio da biblioteca Scikit-learn. Além do algoritmo, toda a modelagem e análise dos dados se dará através do Google Collaboratory, na linguagem Python, em que serão utilizadas algumas das suas principais bibliotecas de modelagem, como: Pandas, Matplotlib, Seaborn, Numpy, a já citada Scikit-learn e outras.

### *A. Dataset*

Fazendo uso do UCI Machine Learning Repository, repositório no qual nos embasamos, obtivemos dados sobre os parâmetros utilizados e os possíveis outputs (classes) de explosões, totalizando um total de 10 desses parâmetros e 3 classes. O Dataset utilizado possui um total de 1389 instâncias, onde cada instância se referencia a uma região ativa do sol além dos 13 atributos já mencionados (classes incluídas), sendo essas:

1) Código para classe (classe de Zurique modificada), que se divide em: A, B, C, D, E, F e G
2) Código do tamanho do maior ponto, dividido em X, R, S, A, H e K
3) Código de distribuição dos pontos, dividido em X, O, I e C
4) Atividade (1 = Reduzida, 2 = Inalterada)
5) Evolução (1 = Decaimento, 2 = Sem Crescimento, 3 = Crescimento)
6) Código da atividade de explosões nas últimas 24 horas (1 = não tão grande quanto um M1, 2 = é um M1, 3 = mais ativo que um M1), em que um M1 é uma classe de explosão solar moderadamente intensa na faixa de raios-X
7) Historicamente complexo (1 = Sim, 2 = Não)
8) A região tornou-se historicamente complexa nessa passagem atual pelo disco solar? (1 = Sim, 2 = Não)
9) Área (1 = pequena, 2 = grande)
10) Área do maior ponto (1 se <= 5, 2 se > 5)
11) Número de explosões solares de classe C nas próximas 24 horas nesta região
12) Número de explosões solares de classe M nas próximas 24 horas nesta região
13) Número de explosões solares de classe X nas próximas 24 horas nesta região

### *B. Processamento do dataset*

Para que seja realizado o processamento do nosso dataset, precisamos que o conjunto de dados em questão passe por 3 etapas ou procedimentos, que se dividem em Filtro de Instâncias, Seleção de Atributos e, por último, Engenharia de Atributos.

Na etapa de Filtro de Instâncias, é feita uma verdadeira limpeza dos dados, removendo ou preenchendo dados incompletos, retirando instâncias repetidas e/ou irrelevantes na análise em questão.

Para a etapa de Seleção de Atributos, como o nome propõe, faremos uma análise dos atributos que, de fato, serão relevantes para a pesquisa, de tal forma que, juntos, formem uma dimensão razoável para explorarmos, servindo dessa forma como input para o uso concreto das técnicas de aprendizagem de máquina.

Finalmente, durante a etapa de Engenharia de Atributos, trabalhamos em cima dos atributos selecionados na etapa anterior, adaptando-os propriamente para uso nas entradas do algoritmo. Como o dataset escolhido já nos forneceu atributos tratados e prontos para uso, não será necessária a realização desta etapa.

### *C. Teorema de Bayes*

O Teorema de Bayes recebe esse nome por ter sido elaborado pelo matemático inglês Thomas Bayes (1702 - 1761). É um teorema fundamental e de uso frequente na Probabilidade e na Estatística, que utiliza o conceito de partições de um conjunto para determinar a probabilidade de

um dado evento ocorrer a partir de informações já conhecidas relacionadas a esse evento.

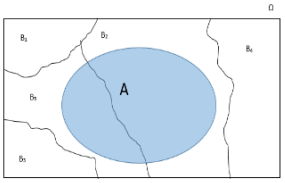


Fig. 1. Esquema utilizado pelo Teorema de Bayes, onde Bi é uma partição do espaço de probabilidade

A fórmula associada a esse teorema, conhecida como fórmula da probabilidade das "causas", é utilizada por outros ramos do conhecimento além da Estatística (como na Medicina e na Engenharia) e é dada a seguir:

$$P(B) = \frac{P(B|A)P(A)}{P(B)} \quad [1]$$

onde temos:

- $P(B)$ sendo a probabilidade do evento A ocorrer sabendo que o evento B já ocorreu
- $P(A)$ sendo a probabilidade de B ocorrer sabendo que A já ocorreu
- $P(A)$ sendo a probabilidade de A ocorrer isoladamente
- $P(B)$ sendo a probabilidade de B ocorrer isoladamente

Ou seja, o Teorema de Bayes permite que você atualize suas deduções iniciais sobre um evento A a partir de deduções já conhecidas sobre o evento B, tornando o cálculo de probabilidades desconhecidas muito mais eficiente. Como se nota, o Teorema de Bayes trata de probabilidades condicionais, P(A | B), utilizando probabilidades marginais, P(A) e P(B).

O Teorema de Bayes é a ferramenta primordial para a modelagem de um classificador probabilístico conhecido como Classificador Ingênuo de Bayes, em que uma certa instância é categorizada em um conjunto de classes a partir de suas características.

*D. Classificador de Bayes*

O Classificador Naive Bayes é um algoritmo probabilístico muito utilizado em machine learning, baseado no "Teorema de Bayes". O método pode ser usado quando os atributos que descrevem as instâncias forem condicionalmente independentes, ou seja, desconsidera a correlação entre as variáveis. Por ser muito simples e rápido, possui um desempenho relativamente maior do que outros classificadores. Além disso, o Naive Bayes só precisa de um pequeno número de dados de teste para concluir classificações com uma boa precisão.

O seu funcionamento pode ser facilmente descrito em termos estatísticos: para calcular a predição, o algoritmo define, primeiramente, uma tabela de probabilidades, em que consta a frequência dos preditores com relação às variáveis de saída. Então, o cálculo final leva em conta a probabilidade maior para oferecer uma solução.

Pontos Positivos:

- É fácil e rápido para prever o conjunto de dados da classe de teste. Também tem um bom desempenho na previsão de classes múltiplas.
- Quando a suposição de independência prevalece, um classificador Naive Bayes tem melhor desempenho em comparação com outros modelos como regressão logística, e você precisa de menos dados de treinamento.
- O desempenho é bom em caso de variáveis categóricas de entrada comparada com a variáveis numéricas. Para variáveis numéricas, assume-se a distribuição normal.

Pontos Negativos:

- Se a variável categórica tem uma categoria que não foi observada no conjunto de dados de treinamento, então o modelo irá atribuir uma probabilidade de 0 e não será capaz de fazer uma previsão. Isso é muitas vezes conhecido como "Zero Frequency". Para resolver isso, podemos usar a técnica de alisamento. Uma das técnicas mais simples de alisamento é a chamada estimativa de Laplace.
- Por outro lado, Naive Bayes é também conhecido como um mau estimador, por isso, as probabilidades calculadas não devem ser levadas muito a sério.
- Outra limitação do Naive Bayes é a suposição de preditores independentes. Na vida real, é quase impossível que ter um conjunto de indicadores que sejam completamente independentes.

*E. Aplicação*

O classificador ingênuo de Bayes será implementado na linguagem de programação *Python* - com o auxílio da ferramenta *Google Colab*, que permite um ambiente de desenvolvimento colaborativo baseado em nuvem. Assim, possibilitando o uso de bibliotecas como a *Numpy, Scipy, Pandas* e *Scikit-learn*. A última contém uma implementação do algoritmo *Naive Bayes,* além de outros recursos de aprendizagem de máquina.

Já a biblioteca *Numpy* é utilizada para a realização de operações matemáticas, integrando-se com a *Scipy* para complementar as funcionalidades de cálculos científicos. Enquanto a *Pandas* é aplicada para manipulação e análise de dados, o que possibilitará a organização de um *dataset* para uso do classificador. Com esses recursos, será possível uma execução eficiente do classificador ingênuo de Bayes.

Outrossim, o conjunto de dados que será utilizado precisará suprir a necessidade do treinamento e do teste do classificador. Assim, sendo separado em dois subconjuntos para melhor controle da aprendizagem de padrões. O treinamento tem como objetivo criar uma base de padrões para o classificador. Enquanto o teste servirá para determinar a sua eficiência.

## IV. ANÁLISE EXPLORATÓRIA

A análise exploratória é parte indispensável da elaboração de um algoritmo preditivo para compreender em sua totalidade os dados do dataset utilizado. A falta dessa análise prejudica a confiabilidade do resultado obtido. O foco principal desta análise reside em examinar detalhadamente a distribuição dos dados utilizando gráficos plotados no Google Colab para entender de que forma cada parâmetro pode afetar as variáveis que são objeto de estudo. No nosso caso, optou-se por uma análise univariada, em que cada variável é examinada uma por uma, reduzindo nossa análise -por questão de praticidade- às variáveis de maior relevância do conjunto de dados. Além da análise univariada, existem diversas outras abordagens que podem ser utilizadas em uma análise exploratória.

Na presente análise exploratória do conjunto de dados escolhido, são observadas apenas variáveis categóricas, isto é, que divide os dados em classes qualitativas, trazendo informações bem estabelecidas sobre as características do conjunto.

Como dito anteriormente, para uma análise mais fluida e prática, foi plotado um gráfico para cada variável de maior relevância do dataset escolhido, contando o número de ocorrências de cada classe da variável. Com isso, podemos identificar como cada variável influencia no target, a variável de interesse (no nosso caso, identificada como sum-class que descreve a quantidade de explosões solares ocorridas no período de 24h dadas as características da área observada no Sol). Nos gráficos exibidos abaixo, conta-se para cada classe da variável duas categorias, 0 e 1, onde 0 indica a não ocorrência e 1 indica a ocorrência de uma explosão solar, dada a característica descrita pela variável.

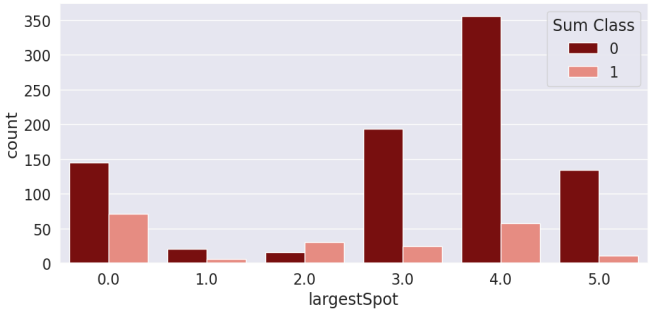


Fig. 2. Distribuição de *largestSpot*

Aqui, observamos o código do tamanho do maior ponto observado do Sol (X = 0, R = 1, S = 2, A = 3, H = 4 ou K = 5), substituída por valores numéricos para que pudesse ser elaborado o algoritmo. Como é evidenciado no gráfico, pontos de código X indicam maior probabilidade da ocorrência de uma explosão solar na área observada.

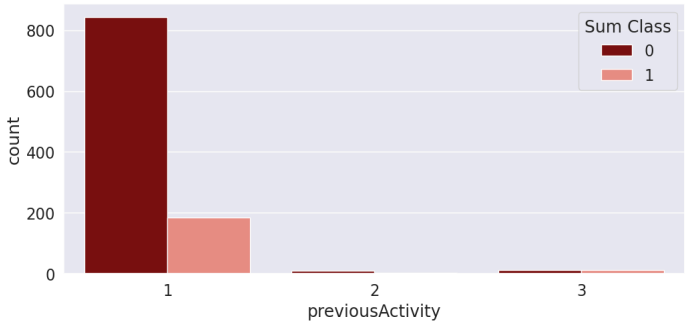


Fig. 3. Distribuição de *previousActivity*

Nesse caso, observamos o tipo da intensidade de atividades anteriores na área observada no Sol: 1 = não tanto quanto em um M1, 2 = um M1, 3 = mais atividade que em um M1. Aqui, vê-se que o dataset se restringiu principalmente a áreas com atividades não tão intensas quanto em um M1, por isso em geral a ocorrência ou não de uma explosão solar são limitadas a essa categoria.

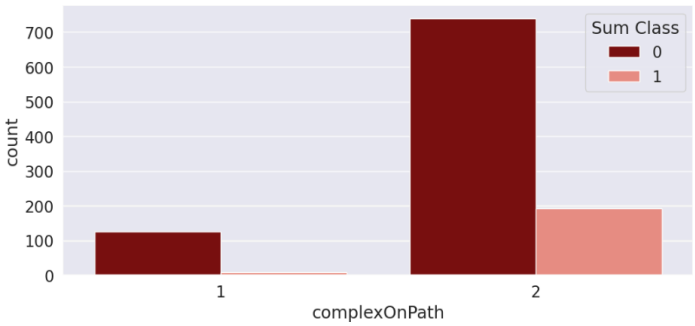


Fig. 4. Distribuição de *complexOnPath*

Dessa vez, observa-se se a área observada no Sol é historicamente complexa, em que 1 = sim e 2 = não. Mais uma vez, nota-se que o banco de dados tem, em sua maioria, áreas não historicamente complexas, o que faz com que a ocorrência ou não de uma explosão solar se limite, nesse caso, a essas áreas.

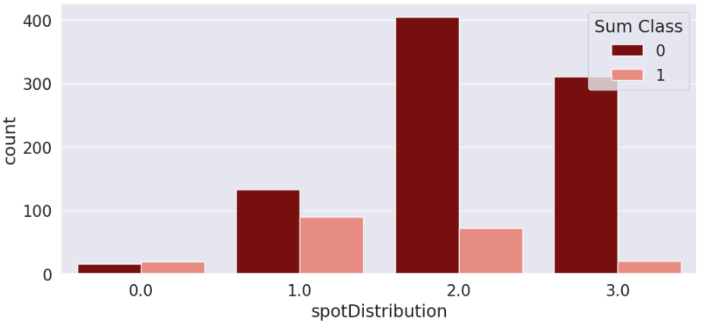


Fig. 5. Distribuição de *spotDistribution*

Agora, analisa-se a distribuição dos pontos na área observada no Sol, sendo essa distribuição categorizada em X = 0, O = 1, I = 2 ou C = 3. Nesse caso, a maior parte das explosões solares ocorrem com uma distribuição do tipo O, enquanto uma distribuição do tipo I geralmente indica não ocorrência de explosão solar.

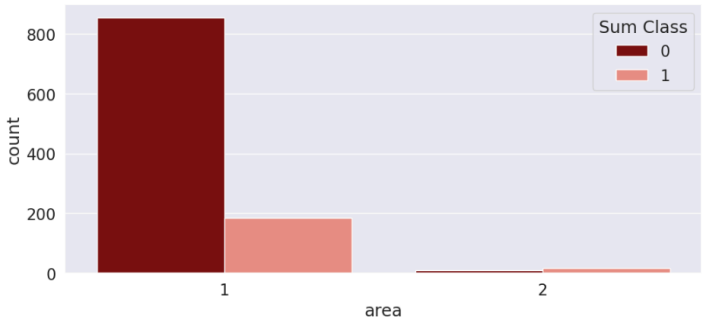


Fig. 5. Distribuição de *area*

Por fim, aqui vemos a análise do tamanho da área observada no Sol, com 1 = pequena e 2 = grande. Novamente, o banco de dados registra principalmente regiões do Sol com áreas pequenas, o que faz a ocorrência e não ocorrência de explosões solares se limitar a tais áreas.

## V. RESULTADOS, EXPERIMENTOS E DISCUSSÃO

Após a análise exploratória, visualizamos os resultados do algoritmo utilizando a função classification_report, da biblioteca Scikit-Learn, a fim principalmente de se obter a precisão dos resultados.

Inicialmente, utilizando sempre as seis variáveis mais relevantes do dataset -class, largestSpot, previousActivity, complexOnPath, spotDistribution e area-, para 70% dos dados reservados para treinamento do modelo, obtemos uma acurácia média de 0.79; com 80% dos dados no treinamento, obtemos 0.82 de precisão média; e, por fim, com 90% como treinamento do modelo, tivemos cerca de 0.85 de precisão -no entanto, os valores de precisão para encontrar a média tiveram um desvio maior. Para este último caso, temos abaixo o resultado do experimento projetado pela biblioteca (lembrando que a precisão dada acima foi uma média das precisões de vários experimentos sob as mesmas condições):

| | precision | recall | f1-score | support |
|---|---|---|---|---|
| 0 | 0.93 | 0.92 | 0.93 | 93 |
| 1 | 0.53 | 0.57 | 0.55 | 14 |
| accuracy | | | 0.88 | 107 |
| macro avg | 0.73 | 0.75 | 0.74 | 107 |
| weighted avg | 0.88 | 0.88 | 0.88 | 107 |

Accuracy: 87.85%

Fig. 6. Avaliação do modelo

Para melhor compreensão dos resultados obtidos e para visualizar de maneira clara os erros cometidos pelo algoritmo, projetamos uma Matriz de Confusão. Nela, pode-se identificar a porcentagem de falsos positivos, falsos negativos, verdadeiros positivos e verdadeiros negativos que o modelo obteve.

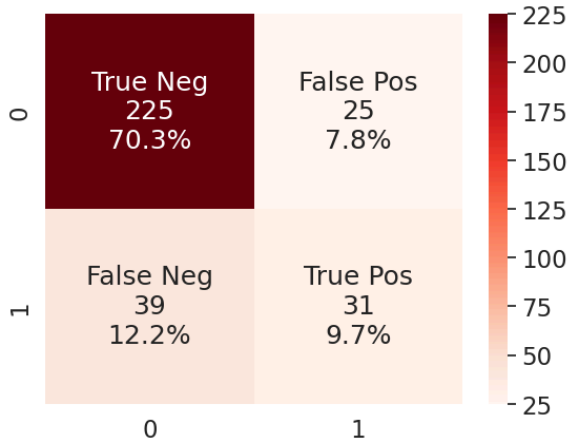


Fig. 6. Matriz de Confusão para avaliação do modelo

Nesse caso, foi projetada uma matriz para 30% dos dados reservados para o teste do modelo. Nota-se que mais de 70% dos dados de teste foram corretamente identificados com a não ocorrência de uma explosão solar como resultado (verdadeiro negativo), enquanto cerca de 10% foi corretamente identificado com a ocorrência de uma explosão solar nas próximas 24h (verdadeiro positivo).

O código da análise exploratória e dos resultados pode ser encontrado no link: https://github.com/LuanThiers/Solar-Flare-Dataset.

## VI. CONCLUSÃO

Realizada a análise exploratória e observando os resultados, nota-se que o modelo utilizado no algoritmo apresentou um resultado com alta precisão. Após uma discussão do grupo, chegamos a algumas conclusões acerca dessa acurácia. Primeiramente, como o nosso output é binário (se tem ou se não tem explosão solar nas próximas 24h dadas as características da região), há menos possibilidades de erro por parte do algoritmo, que não precisa "decidir" entre muitas possíveis respostas.

Além disso, é possível que o dataset escolhido seja tendencioso, no sentido de que muitas variáveis (mesmo as mais relevantes do conjunto de dados) tenham instâncias restritas a poucas de suas categorias. Por exemplo, como já falado na análise exploratória, a maior parte das regiões do Sol observadas no dataset foram de áreas não historicamente complexas. Isso limita a diversidade de combinações entre as várias variáveis e torna "mais fácil" para o algoritmo acertar sua resposta: ora, se há poucas combinações para cada output possível, o resultado entregue pode se "chutar" como sendo aquele que mais se repete entre essas poucas combinações.

## REFERÊNCIAS

[1] Meyer, Paul, "Probabilidade Aplicações à Estatística", 2a ed.

[2] Explosão Solar além do apagão da internet: quais são os outros efeitos para a Terra? Terra, 2023. Disponível em: <https://www.terra.com.br/byte/explosao-solar-alem-do-apagao-da-internet-quais-sao-os-outros-efeitos-para-a-terra,8c118ce1fd525024a0a277ecbac1bf2byadi1uo3.html?utm_source=clipboard>. Acesso em: 24 jan. 2024

[3] As violentas tempestades solares que ameaçam a Terra. BBC News, 2022. Disponível em: <https://www.bbc.com/portuguese/geral-59942719>. Acesso em: 24 jan. 2024

[4] Naive Bayes classifier. Wikipedia. Disponível em: <https://en.wikipedia.org/wiki/Naive_Bayes_classifier>. Acesso em: 24 jan. 2024

[5] Naive Bayes: como funciona esse algoritmo de classificação. Tera Blog. Disponível em: <https://blog.somostera.com/data-science/naive-bayes>. Acesso em: 24 jan. 2024

[6] Teorema de Bayes. Wikipedia. Disponível em: <https://pt.wikipedia.org/wiki/Teorema_de_Bayes>. Acesso em: 24 jan. 2