Supervised Learning: Near Earth Objects Classification

Artificial Intelligence

Turma 10 - Grupo A2 91

Ntsay Zacarias up202008863

27 de Maio de 2024

Contents

1 Introdução **2**

2 Implementação/Processo **2**
2.1 Pré-processamento de Dados 2
2.2 Seleção de Características e Treino do Modelo: 2

3 Treino dos Modelos **4**
3.1 Preparação dos Dados 4
3.2 Support Vector Machine (SVM) 5
3.3 Neural Networks 6
3.4 Random Forest 6
3.5 K-Nearest Neighbors (K-NN) 6

4 Resultados e Discussão **6**
4.1 Curvas de Precisão-Recall 6
4.2 Curvas ROC 7
4.3 Tempo de Treino e Resultados 8

5 Conclusão **10**

6 Dataset **10**

1 Introdução

No âmbito dos estudos espaciais, a classificação de objetos celestes, como os asteroides, desempenha um papel crucial na compreensão das suas características e potenciais riscos. Este projeto tem como objetivo utilizar *Machine Learning* para prever se os asteróides são perigosos com base nas suas características físicas e orbitais. O objetivo deste projeto foi desenvolver um modelo que melhore a precisão da classificação de asteróides.

Nesta iteração, implementei e comparei vários algoritmos de ML, incluindo Random Forest, Decision Tree, K-Nearest Neighbors (K-NN), Support Vector Machines (SVM) com diferentes kernels (linear, polinomial, RBF), e Neural Networks.

2 Implementação/Processo

A implementação envolveu várias fases-chave:

2.1 Pré-processamento de Dados

A fase de pré-processamento dos dados é crucial para garantir a qualidade e a precisão das análises subsequentes. Este processo envolveu várias etapas estratégicas para refinar o conjunto de dados, detalhadas a seguir:

- **Limpeza de Atributos Redundantes ou Desnecessários:** Para simplificar o modelo e melhorar o desempenho computacional, atributos duplicados ou irrelevantes foram removidos. Isso incluiu a eliminação de medidas em diferentes unidades para o mesmo atributo (como diâmetro em metros, milhas e pés) mantendo apenas em quilômetros *(Km)* e unidades astronómicas *(AU)* que são padrão para análises astronômicas.

- **Unificação de Nomes e Eliminação de Variáveis Constantes:** O atributo '*Name*', que era redundante com '*Neo Reference ID*', foi removido, assim como '*Orbiting Body*' e '*Equinox*', que apresentavam valores constantes e, portanto, não contribuíam para a análise preditiva.

- **Valores Ausentes:** Após a limpeza inicial, a análise dos dados revelou a presença de valores ausentes em algumas colunas. Esses foram imputados utilizando métodos estatísticos apropriados para preservar a integridade do conjunto de dados sem introduzir viés significativo.

2.2 Seleção de Características e Treino do Modelo:

Um classificador Random Forest foi treinado, e as características foram selecionadas com base na sua importância.

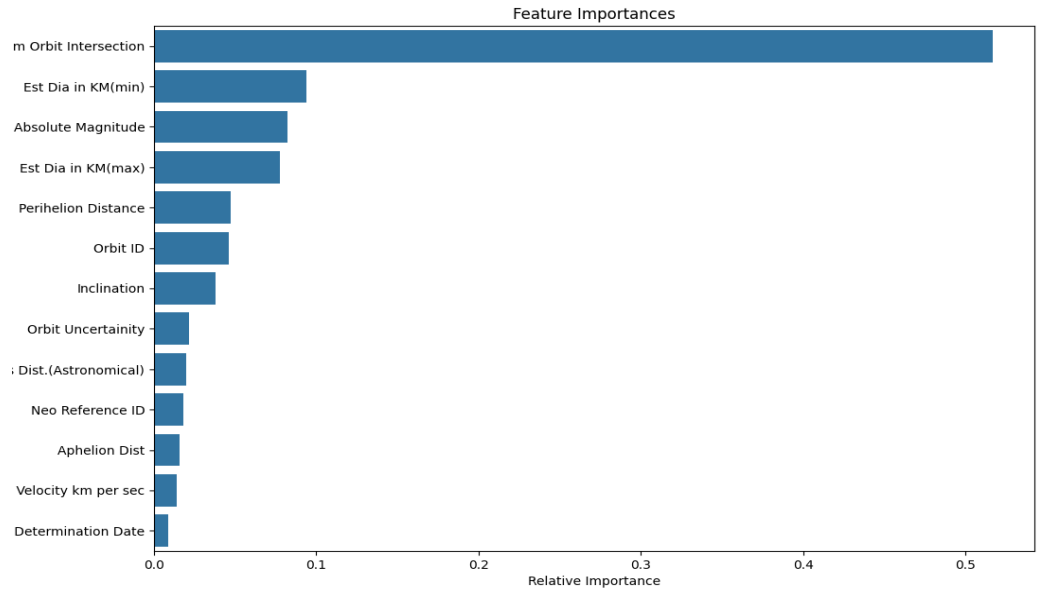

Figure 1: Features escolhidas (acima da mediana)

Inicialmente, utilizei as características selecionadas diretamente, mas notei que os resultados eram demasiadamente altos, indicando que algumas variáveis provavelmente estavam a informar o modelo da resposta correta. Para tentar remediar este problema, criei uma matriz de correlação das características, e os resultados foram os seguintes:

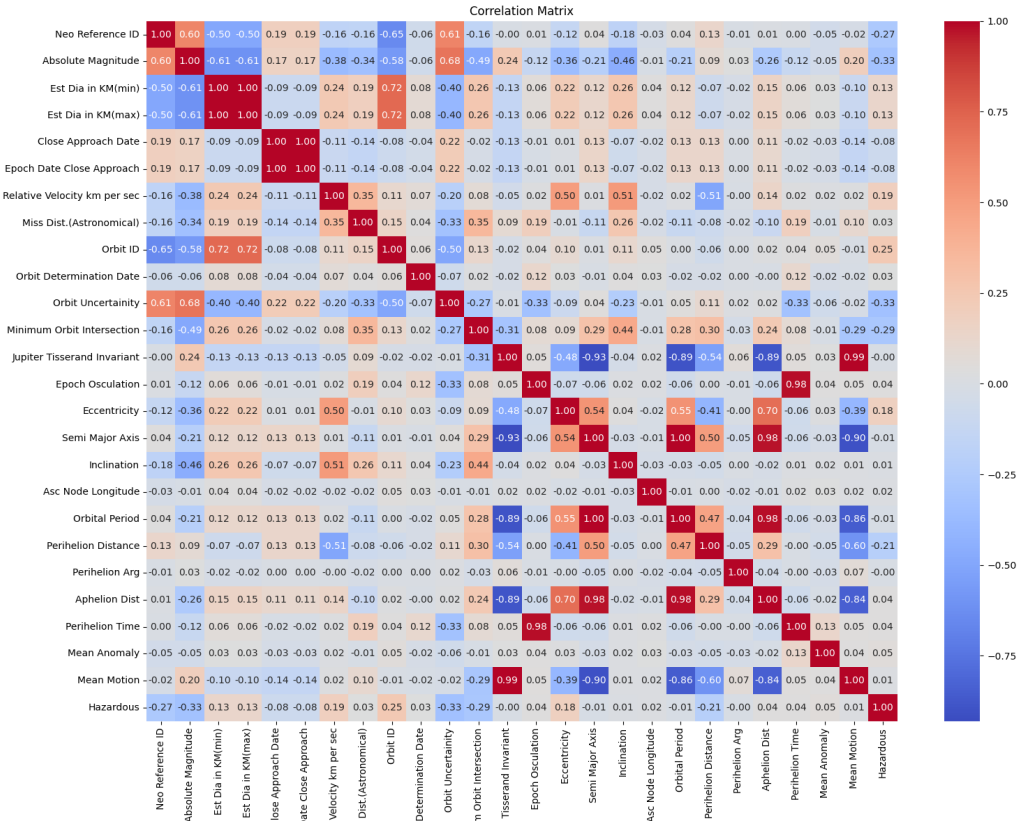

Figure 2: Matriz de Correlação das Características

As variáveis ***Absolute Magnitude***, ***Minimum Orbit Intersection Distance (MOID)*** e ***Perihelion Distance*** mostraram ter mais influência, embora mínima, na correlação com a variável '*Hazardous*'. Embora a influência seja pequena, decidi remover estas variáveis para tentar simular uma situação mais realista.

3 Treino dos Modelos

3.1 Preparação dos Dados

Antes de treinar os modelos, os dados foram preparados da seguinte forma:

- **Seleção de Características**: Características específicas, como 'Orbit Uncertainty', 'Absolute Magnitude', 'Minimum Orbit Intersection', e 'Perihelion Distance' foram removidas.
- **Escalonamento de Características**: Utilizou-se o *StandardScaler* para normalizar as características.
- **Redução de Dimensionalidade**: Aplicação de PCA para manter 95
- **Divisão dos Dados**: Os dados foram divididos em conjuntos de treino (60

3.2 Support Vector Machine (SVM)

Para escolher o melhor modelo SVM, testei diferentes kernels: linear, polinomial (poly) e radial (RBF). Cada kernel tem suas próprias características e é adequado para diferentes tipos de problemas:

- **Linear**: Utiliza uma função de decisão linear para separar as classes. É mais eficiente para problemas linearmente separáveis.
- **Polinomial (Poly)**: Utiliza uma função de decisão polinomial para criar limites de decisão mais complexos, adequado para dados com relações não lineares.
- **Radial Basis Function (RBF)**: Utiliza uma função de decisão baseada em distância, ideal para problemas onde as fronteiras de decisão são altamente não lineares.

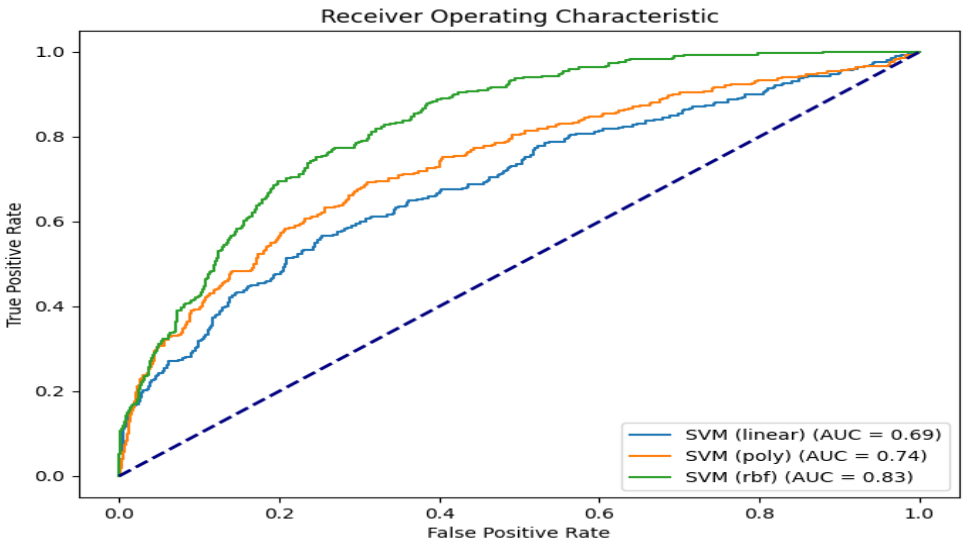

Figure 3: Comparação das Curvas ROC dos Diferentes Kernels do SVM

Os três kernels foram comparados usando as curvas ROC e PRC, e os resultados mostraram que o SVM com kernel RBF teve o melhor desempenho com AUC = 0.83, seguido pelo kernel polinomial com AUC = 0.74 e o kernel linear com AUC = 0.69. Com base nesses resultados, escolhi o SVM com kernel RBF para ser o modelo comparado com os demais algoritmos (NN, RF, etc.).

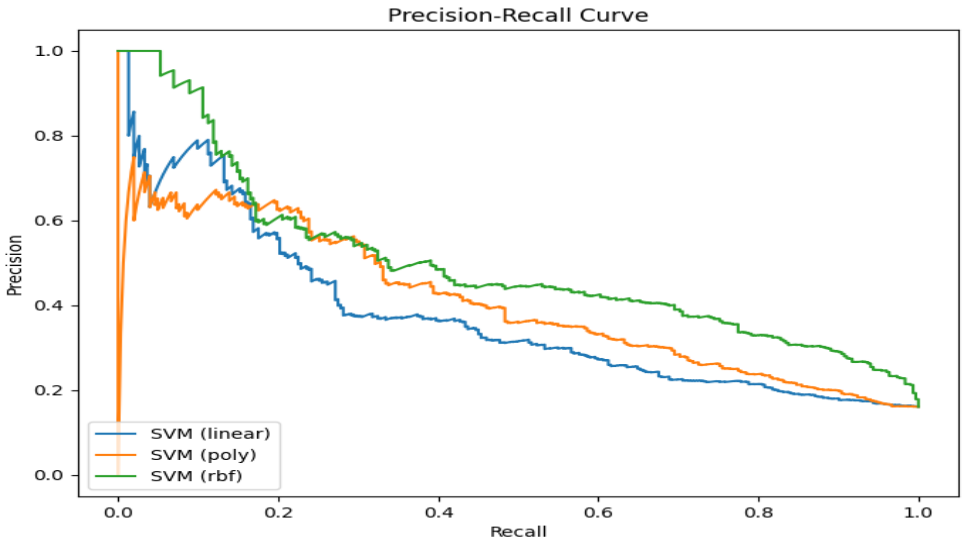

Figure 4: Comparação das Curvas PRC dos Diferentes Kernels do SVM

3.3 Neural Networks

Para Redes Neurais Artificiais, utilizei um classificador ***Multilayer Perceptron*** (***MLPClassifier***), com a configuração padrão do MLPClassifier, com um máximo de 1000 iterações.

3.4 Random Forest

Para este projeto, utilizei um Random Forest com 100 estimadores (árvores), o que proporcionou um bom equilíbrio entre desempenho e precisão.

3.5 K-Nearest Neighbors (K-NN)

Para este projeto, utilizei k=5, o que significa que cada instância é classificada com base nas 5 instâncias mais próximas no conjunto de treino.

4 Resultados e Discussão

4.1 Curvas de Precisão-Recall

A NN apresentou a melhor performance global, seguida pelo Random Forest e SVM (RBF).

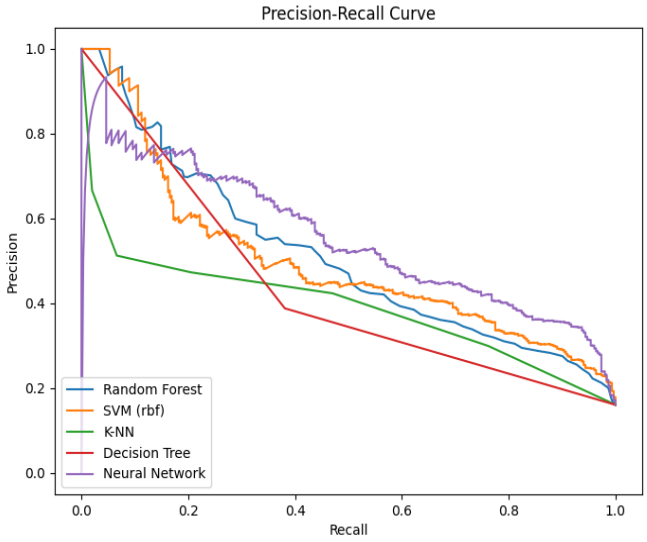

Figure 5: Curvas de Precisão-Recall dos Diferentes Modelos (Reduced Features)

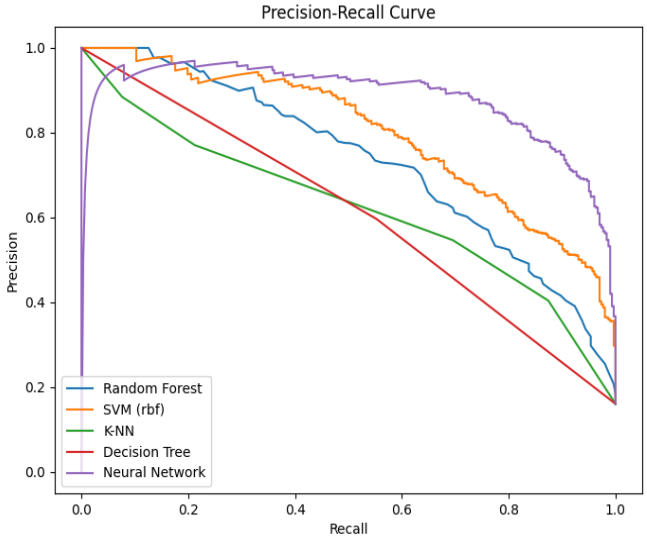

Figure 6: Curvas de Precisão-Recall dos Diferentes Modelos

4.2 Curvas ROC

A NN novamente apresentou a melhor performance com AUC = 0.87, seguida pelo SVM (RBF) com AUC = 0.83 e Random Forest com AUC = 0.81.

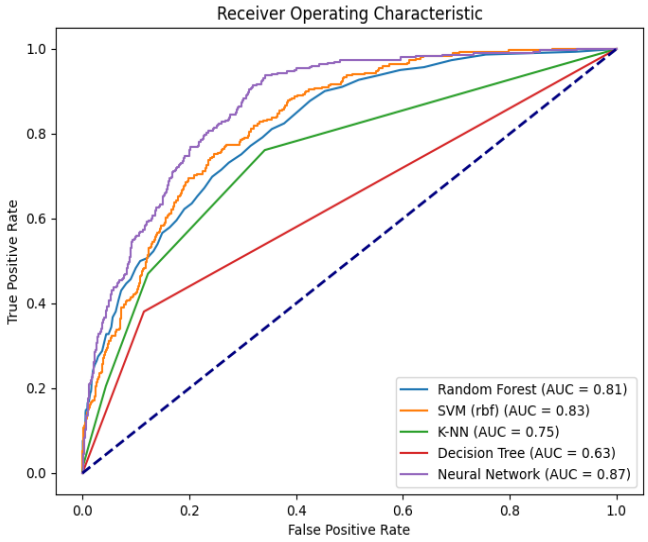

Figure 7: Curvas ROC dos Diferentes Modelos (Reduced Features)

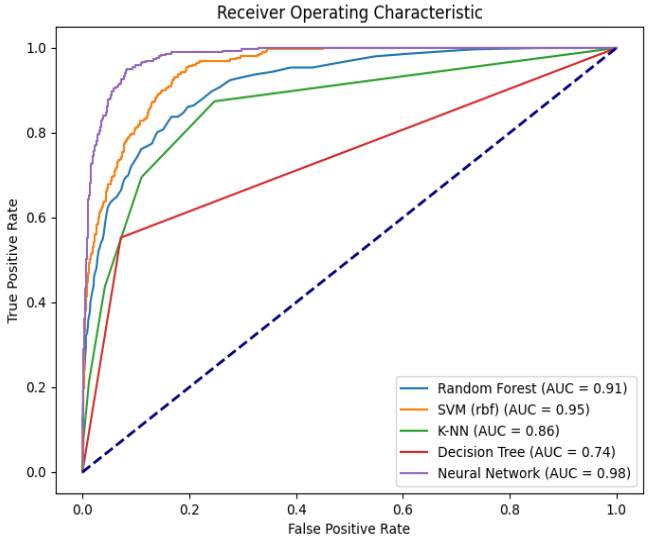

Figure 8: Curvas ROC dos Diferentes Modelos

4.3 Tempo de Treino e Resultados

Observa-se que a NN exigiu um tempo de treino significativamente maior em comparação com os outros modelos.

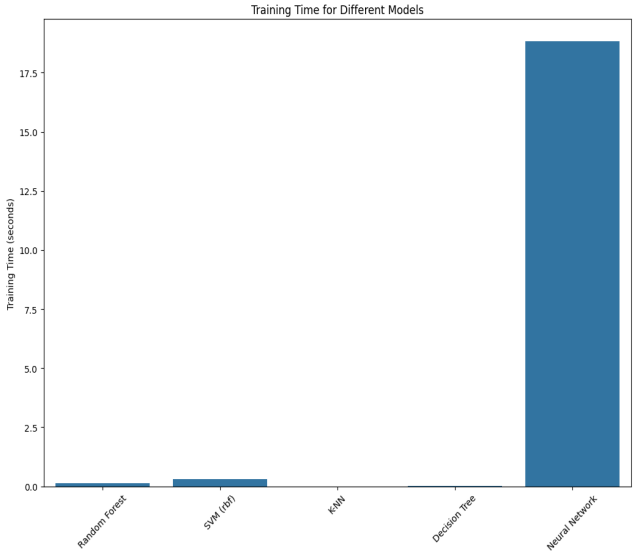

Figure 9: Tempo de Treino dos Diferentes Modelos

```
Displaying results...
Model: Random Forest
Accuracy: 0.8570666666666666
Precision: 0.6976744186046512
Recall: 0.1986754966887417
F1 Score: 0.30927835051546393
Training Time: 0.12523698806762695 seconds
Confusion Matrix:
 [[1547   26]
 [ 242   60]]
================================================================
Model: SVM (rbf)
Accuracy: 0.8538666666666667
Precision: 0.71875
Recall: 0.152317880794702
F1 Score: 0.25136612021857924
Training Time: 0.3016388416290283 seconds
Confusion Matrix:
 [[1555   18]
 [ 256   46]]
================================================================
Model: K-NN
Accuracy: 0.8352
Precision: 0.4732824427480916
Recall: 0.2052980132450331
F1 Score: 0.2863741339491917
Training Time: 0.0008039474487304688 seconds
Confusion Matrix:
 [[1504   69]
 [ 240   62]]
================================================================
Model: Decision Tree
Accuracy: 0.8037333333333333
Precision: 0.3885135135135135
Recall: 0.38079470198675497
F1 Score: 0.38461538461538464
Training Time: 0.022948026657104492 seconds
Confusion Matrix:
 [[1392  181]
 [ 187  115]]
================================================================
Model: Neural Network
Accuracy: 0.856
Precision: 0.5689655172413793
Recall: 0.4370860927152318
F1 Score: 0.4943820224719101
Training Time: 18.824346780776978 seconds
Confusion Matrix:
 [[1473  100]
 [ 170  132]]
================================================================
```

Figure 10: Resultados Detalhados dos Modelos

5 Conclusão

Os resultados deste projeto demonstram que a Neural Network foi o modelo mais eficaz para a classificação de asteroides perigosos, apresentando a melhor performance em termos de AUC nas curvas ROC e PRC, apesar do seu maior tempo de treino. O Random Forest e o SVM (RBF) também mostraram bons resultados, com tempos de treino mais reduzidos.

A análise sugere que, embora modelos complexos como Redes Neurais possam oferecer alta precisão, é importante considerar o tempo de treino e a complexidade do modelo ao escolher a melhor abordagem para a classificação de asteroides. Modelos mais simples, como Random Forest, podem oferecer um bom equilíbrio entre precisão e eficiência computacional.

Adicionalmente, foi feita uma comparação entre modelos com *feature engineering*, nos quais reduzi o número de características, e modelos sem essa redução. A correlação entre as características parecia baixa, e isso refletiu-se nos resultados. Embora os resultados sejam diferentes, a ordem de desempenho dos modelos manteve-se. Ao adicionar mais características, os modelos mostraram-se mais performativos em termos de precisão e accuracy. Isso indica que a inclusão de mais características relevantes pode melhorar o desempenho dos modelos, mas a simplicidade e a eficiência computacional também devem ser consideradas.

6 Dataset

O dados foram fornecidos pela NASA a partir um serviço web RESTful *NeoWs (Near Earth Object Web Service)*.

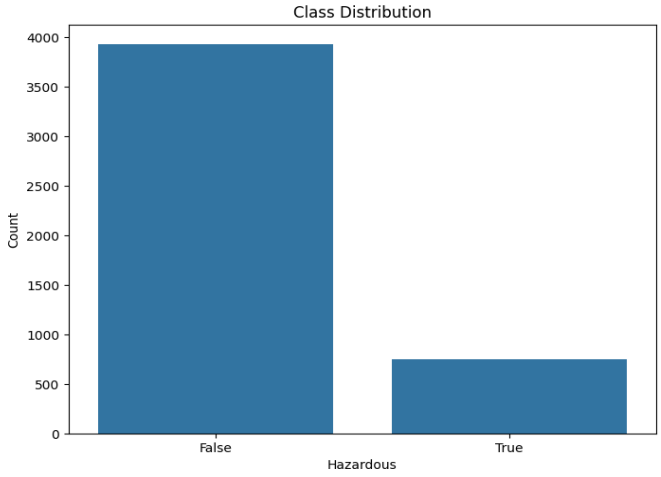

Figure 11: A distribuição das classes no conjunto de dados mostra um desequilíbrio significativo entre os asteroides classificados como perigosos e não perigosos.

Segue uma breve descrição de cada atributo dos dados fornecidos (com omissão de atributos redundantes):

- **Neo Reference ID**: Identificador único para cada asteroide na base de dados, usado como referência principal.
- **Name**: Nome oficial ou designação do asteroide.

- **Absolute Magnitude**: Medida do brilho do asteroide, que é uma indicação do seu tamanho e refletividade.
- **Est Dia in KM(min)** e **Est Dia in KM(max)**: Diâmetro estimado do asteroide, apresentado como um intervalo mínimo e máximo em quilômetros.
- **Close Approach Date**: Data em que o asteroide passa mais próximo da Terra.
- **Epoch Date Close Approach**: Representação em formato de época (UNIX timestamp) da data em que o asteroide esteve mais próximo da Terra.
- **Relative Velocity km per sec**: Velocidade do asteroide em relação à Terra, medida em quilômetros por segundo.
- **Miss Dist.(Astronomical)**: Distância por que o asteroide passa da Terra, medida em unidades astronômicas (AU).
- **Orbit ID**: Identificador da órbita do asteroide, usado para rastrear diferentes observações da mesma trajetória.
- **Orbit Determination Date**: Data em que a órbita do asteroide foi determinada com precisão.
- **Orbit Uncertainity**: Medida da incerteza da órbita do asteroide, indicando quão bem sua trajetória é conhecida.
- **Minimum Orbit Intersection**: Distância mínima entre a órbita do asteroide e a órbita da Terra, um indicador crítico do potencial de colisão.
- **Jupiter Tisserand Invariant**: Um valor numérico que ajuda a classificar a órbita do asteroide em relação à influência de Júpiter.
- **Epoch Osculation**: Data de referência para os elementos orbitais dados.
- **Eccentricity**: Medida da excentricidade da órbita do asteroide, onde 0 significa uma órbita perfeitamente circular.
- **Semi Major Axis**: O maior eixo da órbita elíptica do asteroide.
- **Inclination**: Ângulo de inclinação da órbita do asteroide em relação ao plano eclíptico da Terra.
- **Asc Node Longitude**: Longitude do nó ascendente da órbita do asteroide.
- **Orbital Period**: Tempo que o asteroide leva para completar uma órbita ao redor do Sol.
- **Perihelion Distance**: Distância mais próxima do Sol no ponto da órbita do asteroide.
- **Perihelion Arg**: Argumento do periélio, que é o ângulo da órbita do asteroide no ponto de maior aproximação do Sol.
- **Aphelion Dist**: Distância mais afastada do Sol no ponto da órbita do asteroide.

- **Perihelion Time**: Momento em que o asteroide está no periélio, o ponto mais próximo do Sol.
- **Mean Anomaly**: Medida do ponto em que o asteroide se encontra na sua órbita em relação ao periélio.
- **Mean Motion**: Velocidade média do asteroide ao longo da sua órbita.
- **Equinox**: Sistema de coordenadas usado para representar os elementos orbitais do asteroide.
- **Hazardous**: Indicador de se o asteroide é considerado potencialmente perigoso para a Terra.

References

[1] Scikit-learn developers. *Model evaluation: quantifying the quality of predictions*. Disponível em: `https://scikit-learn.org/stable/modules/model_evaluation.html`.

[2] Medium - Analytics Vidhya. *Evaluating a Random Forest Model*. Disponível em: `https://medium.com/analytics-vidhya/evaluating-a-random-forest-model-9d165595ad56`.

[3] Analytics Vidhya. *Metrics to Evaluate your Classification Model to take the Right Decisions*. Disponível em: `https://www.analyticsvidhya.com/blog/2021/07/metrics-to-evaluate-your-classification-model-to-take-the-right-decisions/`. Acesso em: [01-05-2024].

[4] Kaggle - Will Koehrsen. *Intro to Model Tuning: Grid and Random Search*. Disponível em: `https://www.kaggle.com/code/willkoehrsen/intro-to-model-tuning-grid-and-random-search`.

[5] Bansal, L. *NASA Asteroids Classification*. Recuperado de Kaggle: `https://www.kaggle.com/datasets/lovishbansal123/nasa-asteroids-classification`.