辨別某近地天體是否為潛在威脅天體

[2021 Fall] Intro. to ML: Final Project

作者：0816066 官澔恩
關鍵字：機器學習、二元分類、潛在威脅天體

I. Introduction

根據定義，潛在威脅天體指的是與地球的最小軌道相交距離少於 0.05 天文單位，並且絕對星等亮達 22 等或更亮的近地天體。[1] 不過，通常判斷一個未知天體是否有機會造成威脅需要綜合多種研究方法與觀察數據才可以斷定的。

因此我希望能利用 NASA 提供的開放資料，訓練出三種二元分類模型，試圖預測某個未知天體是否為潛在威脅天體。並分析與比較各模型在準確率、精準率及召回率等表現上的差異。

II. Data Collection

首先，我到 NASA 的 API 入口網站[2] 註冊以取得 API key。並且在公開的 APIs 中找到「Asteroids - NeoWs」這支 API。於是，我寫了一個腳本來協助我從 NASA 的 Open API 取得資料。為了避免戳爆 API，每戳一次我會休息一秒。

此外，為了彈性運用取得的資料，我把它們全部存下來。之後就能夠在我的電腦自行運用，不需要再透過 API 去取得了。

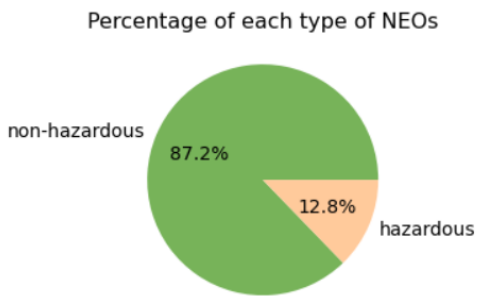
接著，我去分析每個近地天體的欄位有哪些，並決定只使用以下的欄位來作為訓練模型的資料：

API 欄位名稱	說明	CSV 標題
id	近地天體的編號	id
name	近地天體的名稱	name
absolute_magnitude_h	天體的絕對星等	abs_mag
estimated_diameter_min	估計直徑的最小值	est_dia_min
estimated_diameter_max	估計直徑的最大值	est_dia_max
epoch_osculation	描述該天體的時刻（採用 J2000.0 歷元）	epoch
minimum_orbit_intersection	和地球的最小軌道相交距離	min_orb_inter
jupiter_tisserand_invariant	相對於木星的蒂塞朗參數	jup_tisser
eccentricity	軌道離心率	e
semi_major_axis	軌道半長軸	a
inclination	天體傾角	i
ascending_node_longitude	升交點黃經	asc_node_long
perihelion_argument	近心點幅角	peri_arg
orbital_period	繞行週期	orb_period
perihelion_distance	近日距	peri_dist
aphelion_distance	遠日距	ap_dist
mean_anomaly	平近點角	mean_ano
mean_motion	平均運動	mean_mot
is_potentially_hazardous_asteroid	是否為潛在威脅天體	hazard

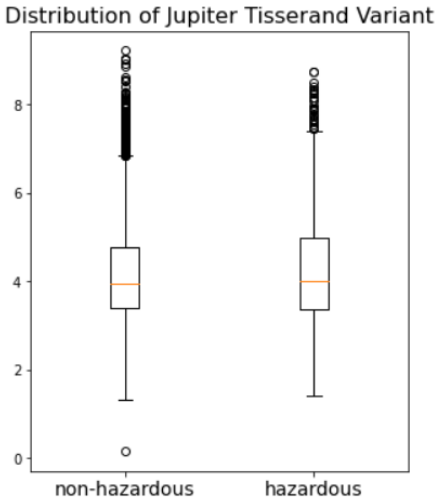
於是我再寫了另外一個腳本將需要的資料讀出來並存成 csv 檔。總共有 12000 筆資料，其中有 7 筆在某些欄位有缺漏，因此我將它們直接捨棄掉。

接著在前處理前我將資料視覺化，嘗試從中看出特別的地方。

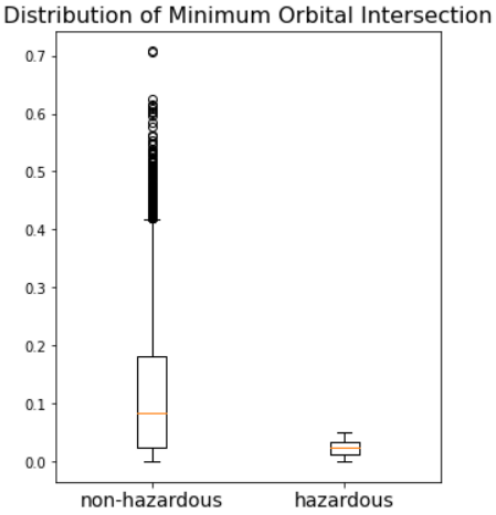
我發現絕大多數的近地天體是沒有潛在威脅的，不過這也代表資料的分佈不平均，因此在資料前處理時需要讓它們的比例可以接近 1:1。

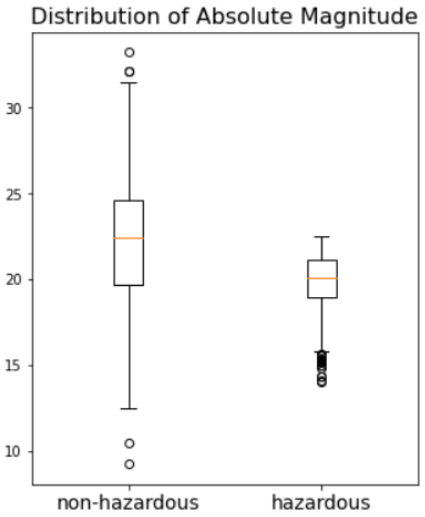


對土星的蒂塞朗參數可以用來區分受擾動的天體屬於小行星或是木星族的彗星[3]。我認為多少可以用來協助分辨是否對地球有危害，因為如果主要受木星的引力影響的話，就不太可能危害地球。然而不管有沒有潛在危險性，蒂塞朗參數在兩者的分佈上幾乎沒有差異。



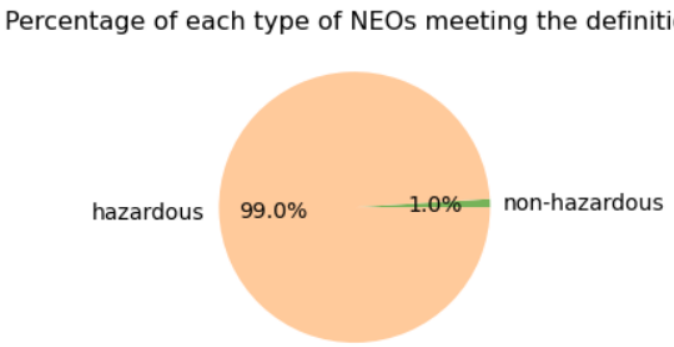
另外，由於最小軌道相交距離及絕對星等和潛在威脅天體的定義直接相關，因此我也去觀察他們的分佈情形。畫出來後，發現果然能在兩者間看出差異。

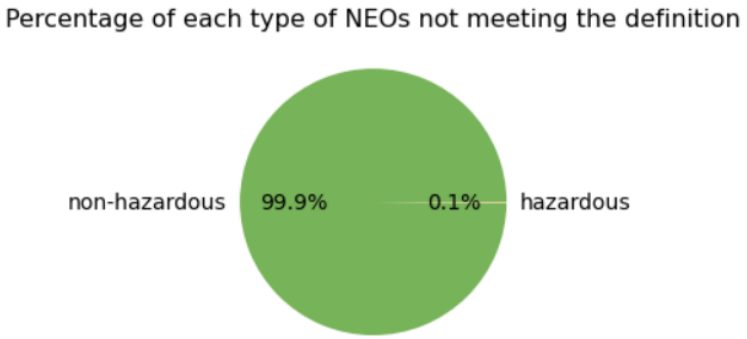




接著我將資料分成符合定義與不符合定義的部份，去看兩類的比例關係。總共 11993 個近地天體，其中 1533 個滿足潛在威脅天體的定義， 10460 個不滿足。可以看到有天體雖然滿足定義但是卻並非潛在威脅天體，也有天體不滿足定義卻是潛在威脅天體。

不過不滿足定義的比例實在是太低了，如果我直接使用定義去判斷的話，輕輕鬆鬆就可以達到 99% 以上的準確率，因此我需要在訓練時將這兩個資訊移除。





III. Preprocessing

1. 首先，我將蒐集到的資料最後面的 30% 分出來作為最後評估表現之用。剩下的七成作為訓練使用，並且會先隨機打亂。
2. 接著我使用 SMOTE 來增加 training set 中潛在威脅天體的數量，使得讓兩種類別的資料平衡。
3. 為了要讓每種 numerical feature 的權重一致，因此我將資料標準化，讓資料的最小最大值 scale 到 [0,1]。其中，testing set 不參與 scalor 的 fitting，但是仍會套用 scaling，才能在輸入模型時使用相同的形式。

IV. Models

我分別選擇使用 random forest、SVM 和 logistic regression 作為我要拿來訓練和預測的模型。以下是我使用的參數：

Random Forest

- number of decision tree: 100
- criterion: gini
- max_depth: (待找出)
- min_samples_leaf: (待找出)

內部 decision tree 的數量我就直接使用預設的 100 棵，至於 criterion 我使用 gini index 來計算 information gain 是因為它在計算上比 entropy 快。

max_depth 和 min_sample_leaf 是我打算在接下來使用 grid search 去找出最佳組合的參數，其中搜尋範圍為：

$$\text{max\_depth} \in \{20,25,30,35,40\}, \text{min\_samples\_leaf} \in \{1,2,4,8,16\}$$

SVM

- kernel: rbf
- C: (待找出)
- gamma: (待找出)

我選擇使用 radial basis function 作為 kernel 的原因是因為，之前在作業三使用它得到的表現最好。而 C 和 gamma 為我接下來要使用 grid search 找出最佳組合的參數，其中搜尋範圍為：
$C \in \{200,210,220,230,240\}, \gamma \in \{5,6,7,8,9\}$

Logistic Regression

- penalty: l2
- C: (待找出)

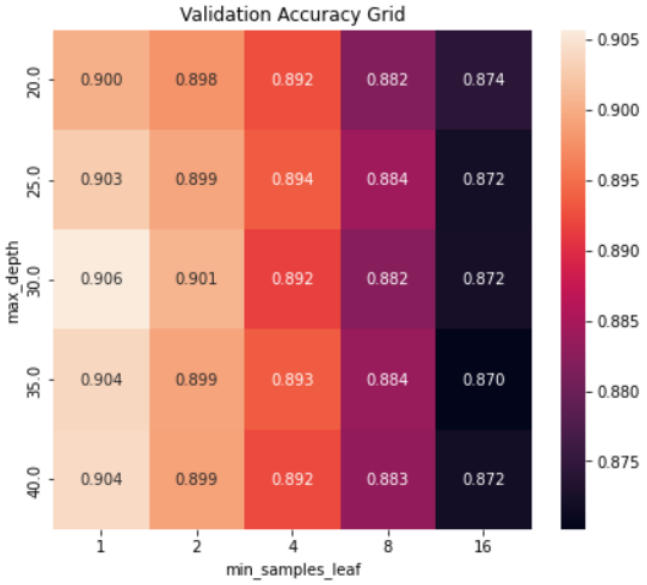
loss 的計算我使用預設的 L2 norm，然後 C 是我接下來要去搜尋的參數，搜尋範圍為：
$C \in \{600,650,700,750,800\}$

V. Results

我針對每個模型去做 5-fold cross validation，取得表現最好的模型後，再用原先預留的那三成資料去測試模型的好壞。

Random forest

以下是經由 grid search 找出來每種參數組合下執行 cross validation 的平均準確率：



其中表現最好的參數組合為 {max_depth: 30.0,min_samples_leaf: 1}。以下是該參數組合對應的模型在 validation 時的平均表現和最終測試時的表現：

{'max_depth': 30.0, 'min_samples_leaf': 1}

truth\pred	0	1
0	5942	1151
1	187	6906

	value
Accuracy	0.905682
Precision	0.857143
Recall	0.973636

Testing Performance

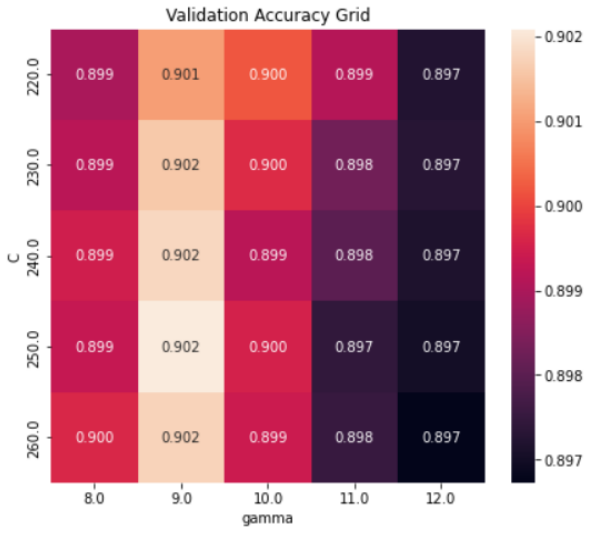
truth\pred	0	1
0	3177	192
1	46	183

	value
Accuracy	0.933852
Precision	0.488000
Recall	0.799127

最後，使用一開始預留的資料去測試這個模型時，準確率能達到 93% 左右。不過可以發現 positive 的 precision 並不高，我認為原因是用來測試的資料中，**negative 的資料數高出 positive 的資料數許多**，導致它把一部分的 negative 預測成 positive 時，會佔預測為 positive 的資料數中很高的比例，使得 positive precision 不到五成。如果看 negative precision 的話，其實就沒有那麼低。

SVM

以下是經由 grid search 找出來每種參數組合下執行 cross validation 的平均準確率：



其中表現最好的參數組合為 {C: 250.0,gamma: 9.0}。以下是該參數組合對應的模型在 validation 時的平均表現和最終測試時的表現：

{'C': 250.0, 'gamma': 9.0}

truth\pred	0	1
0	6060	1033
1	356	6737

	value
Accuracy	0.902087
Precision	0.867053
Recall	0.949810

Testing Performance

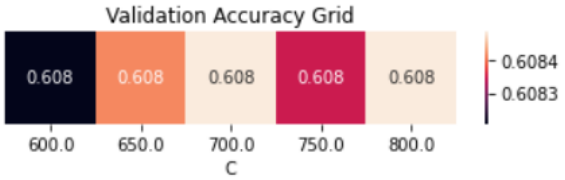
truth\pred	0	1
0	2954	415
1	123	106

	value
Accuracy	0.850472
Precision	0.203455
Recall	0.462882

一樣使用預留的資料去測試模型的表現，準確率來到 85%。positive precision 和前者有一樣的狀況，不過理由是相同的。值得注意的是準確率和 positive recall 比訓練時的還要低，尤其是後者。因此可以推測模型可能有 overfitting 的跡象。

Logistic Regression

以下是經由 grid search 找出來每種參數組合下執行 cross validation 的平均準確率：



其中表現最好的參數為 {C: 700.0} 和 {C: 800.0}，不過我取前者。以下是該參數對應的模型在 validation 時的平均表現和最終測試時的表現：

{'C': 700.0}

truth\pred	0	1
0	4542	2551
1	3003	4090

	value
Accuracy	0.608487
Precision	0.615871
Recall	0.576625

Testing Performance

truth\pred	0	1
0	2894	475
1	129	100

	value
Accuracy	0.832129
Precision	0.173913
Recall	0.436681

一樣使用一開始預留的三成資料去測試這個模型，最後得到的準確率只有 83% 左右。不過比起訓練時的準確率，logistic regression 在測試時的準確率明顯較高。positive precision 不高的理由相同，然而 positive recall 卻不高。其實這點從訓練時就可以看出，它在訓練時的表現本來就很平庸了。

VI. Conclusion

1. performance 排名：random forest > SVM > logistic regression

    - 其中 random forest 的準確率能在不使用定義所用到的「最小軌道相交距離」和「絕對星等」時達到 93%，因此我認為它有潛力協助判斷近地天體是否有潛在危害性。
    - SVM 在訓練時有不錯的表現，然而最終有 overfitting 的現象，我認為它還有機會再去微調，達到更好的表現。
    - 而 logistic regression 在訓練時的表現就已經不好了，可能是因為 feature 的數量過少。加上它已經沒有能夠調整的參數了，因此我認為它不適合用在這個題目上。

2. 訓練所要花的時間：SVM > random forest > logistic regression

    - random forest 訓練所花的時間算適中，然而可以在這三者中得到最好的表現，我認為是值得投入時間調整參數的模型。

總結來說，random forest 是我發現最適合用來預測一個近地天體是否具有潛在危害性的模型。或許是因為科學家在分類時，本來就是依靠一個閾值是否超過來分類的。不過我已經排除定義所使用的 feature 了，所以也有可能是因為其他的 feature 和定義所使用的 feature 之間有潛在的關聯。

VII. References

1. 潛在威脅天體。維基百科。https://zh.wikipedia.org/zh-tw/潛在威脅天體 (https://zh.wikipedia.org/zh-tw/%E6%BD%9B%E5%9C%A8%E5%A8%81%E8%84%85%E5%A4%A9%E9%AB%94) ↩

2. { NASA APIs }. NASA Open Innovation Team. https://api.nasa.gov/ (https://api.nasa.gov/) ↩

3. 蒂塞朗參數。維基百科。https://zh.wikipedia.org/wiki/蒂塞朗参数 (https://zh.wikipedia.org/wiki/%E8%92%82%E5%A1%9E%E6%9C%97%E5%8F%82%E6%95%B0) ↩