Machine Learning come supporto alla classificazione KOI

Marzio Della Bosca[1]

[1]m.dellabosca@campus.uniurb.it

Riassunto

Il progetto è stato realizzato utilizzando i dati raccolti dal telescopio Kepler nel corso degli anni. Esistono diverse versioni dei dati raccolti, e nelle versioni più recenti il parametro relativo all'età della stella del sistema a cui appartiene il KOI (Kepler Object of Interest) è stato rimosso. Dopo aver eseguito il preprocessing sulla release più recente e aver ottenuto un nuovo set di dati mediante un operazione di merge basata sull'identificativo dei KOI, di cui avevo l'età stellare, ho confrontato le accuratezze ottenute addestrando gli stessi modelli con gli stessi parametri su entrambi i dataset. Ho scelto di realizzare il progetto su questa tematica perché mi è sembrata una valida applicazione di classificazione supervisionata: determinare se un oggetto è o non è un esopianeta, poiché nei dataset era specificato per ciascun oggetto se la sua natura fosse confermata o meno. Per gli oggetti classificati come candidati ho optato per la loro rimozione.

1 Introduzione

Il problema che si cerca di affrontare attraverso l'uso di modelli di machine learning sui dati di Kepler è la classificazione dei KOI come esopianeti o meno, e si mira a quantificare l'importanza del parametro relativo all'età stellare della stella associata al KOI. Intuitivamente, l'età stellare è un parametro cruciale da considerare, poiché più una stella è "anziana", più tempo ha avuto il materiale del sistema per precipitare (agglomerarsi) sotto l'influenza della sua gravità.

La rilevanza di questo problema è strettamente connessa agli obiettivi principali della missione Kepler1. Progettata per rilevare esopianeti tramite il metodo del transito, che consiste nell'osservare le variazioni periodiche nella luminosità di una stella quando un pianeta transita di fronte ad essa. [2][4]

L'applicazione del machine learning a questo problema di classificazione è fondamentale poiché, considerando che Kepler rileva un gran numero di segnali candidati che potrebbero essere transiti di esopianeti, la classificazione consente innanzitutto di identificare i candidati più interessanti. Ciò ottimizza l'allocazione delle risorse, evitando di dedicare tempo ed energie a segnali che potrebbero essere falsi positivi. Inoltre, una classificazione accurata è essenziale per garantire la validità dei risultati scientifici derivanti dall'analisi dei dati di Kepler. Ciò assicura che le scoperte riportate siano affidabili e contribuisce alla costruzione di una solida base di conoscenza. [1][3][5]

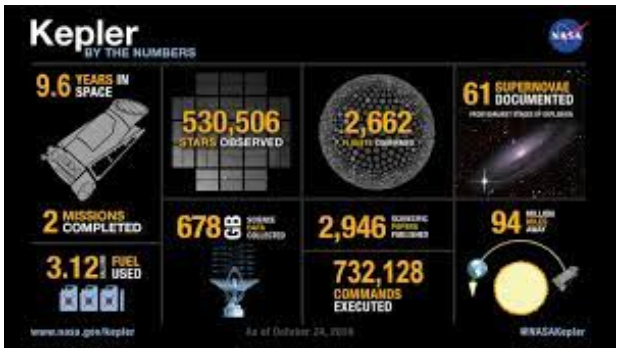

Figura 1: Progetto Kepler, in sintesi.

Il progetto è stato implementato tramite un notebook jupyter il cui codice è stato interamente scritto in linguaggio interpretato python con il supporto di diverse librerie gratuite. (e.g. `scikit-learn`, `Pandas`, `Numpy`, `Matplotlib`, `Seaborn`, `Os`, `imblearn`).

2 Metodi

Dopo aver scaricato la release più recente del dataset, in formato csv, dal sito `https://exoplanetarchive.ipac.caltech.edu/docs/intro.html` ho implementato il preprocessing dei dati attraverso diversi passaggi:

- **Riduzione della dimensionalità**: Ho eliminato le feature che mi sembravano meno rilevanti per incrementare l'accuratezza del modello e anche per ridurre i tempi di addestramento, causa risorse limitate. Passando così da 83 a 24 feature.
- **Gestione dati mancanti**: Ho eliminato poi le colonne che avevano valori nulli per più del 70% delle istanze (arrivando a 23 feature) e successivamente ho eliminato le istanze in cui comparivano valori nulli nelle feature rimanenti passando da 9564 a 8305 elementi.
- **Dataset con età stellare**: Dopo aver ottenuto il dataset, parzialmente pulito, della release più recente ho prodotto un nuovo dataset attraverso un operazione di merge sul parametro kepid aggiungendo l'età stellare ad ogni istanza il cui kepid era presente in entrambe le release mantenendo tutti i dati della release più recente in quanto, teoricamente, più aggiornati.
- **Pulizia finale**: Data la natura della mia classificazione, ovvero supervisionata, ho poi eliminato da entrambi i dataset le istanze dei KOI la cui natura non era verificata e, per sicurezza, eventuali istanze con kepid uguale. Passando così a 5951 istanze per il dataset senza la feature dell'età stellare e a 2336 istanze per il dataset con la feature relativa all'età stellare (24 feature).
- **Gestione delle etichette**: Ho utilizzato prima la tecnica di encoding binario, ponendo a 1 le etichette quando erano confermati come esopianeti e a 0 quando non lo erano e poi la "bipolar labeling convention" ponendo le etichette rispettivamente a 1 e -1.

Ho poi tolto da entrambi i dataset le colonne "kepid", "rowid" e "kepoi_name" non necessarie al fine dell'addestramento. I set sono stati suddivisi in training-set e test-set con una suddivisione 80% per il training-set e 20% per il test-set. Di seguito sono riportate le feature mantenute al fine dell'addestramento:

- **koi_disposition**: Disposizione dell'Archivio degli Esopianeti, se 1 è un pianeta confermato, se 0 è un falso positivo.
- **koi_period**: Il tempo necessario al KOI per orbitare attorno alla sua stella (in giorni).
- **koi_impact**: Impatto, che è la frazione del raggio del pianeta che attraversa il disco della stella durante il transito.
- **koi_duration**: Durata del transito, che è la durata di ciascun transito (ore).
- **koi_depth**: Profondità del transito in parti per milione (ppm), che è una misura di quanto il pianeta blocca la luce della sua stella.
- **koi_ror**: Rapporto tra il raggio del KOI e il raggio della stella.
- **koi_prad**: Raggio del KOI in raggi terrestri, che è la dimensione del KOI rispetto alla Terra.
- **koi_sma**: Semiasse maggiore dell'orbita in unità astronomiche (UA), che è la distanza media tra il KOI e la sua stella.
- **koi_teq**: Temperatura di equilibrio (K), che è la temperatura che il KOI avrebbe se fosse un corpo nero perfetto e assorbisse tutta la luce dalla sua stella.
- **koi_insol**: Flusso di insolazione in flussi terrestri, che è una misura di quanta luce solare il pianeta riceve dalla sua stella.
- **koi_dor**: Distanza tra il pianeta e la stella rispetto al raggio della stella.
- **koi_count**: Numero di pianeti che sono stati rilevati orbitare attorno alla stessa stella.
- **koi_num_transits**: Numero di volte che il KOI è passato davanti alla sua stella.
- **koi_steff**: Temperatura della superficie della stella a cui il KOI è associato (K).
- **koi_slogg**: Gravità superficiale della stella in $log(cm/s^2)$, che è una misura della gravità.
- **koi_smet**: Metallicità della stella, una misura dell'abbondanza dei metalli nella stella (dex).
- **koi_srad**: Raggio della stella in raggi solari. 2
- **koi_smass**: Massa della stella in masse solari.
- **koi_sage**: Età della stella in miliardi di anni, presente in solo uno dei due dataset (Gyr).

I modelli utilizzati sono stati inizialmente il modello SVC, un'istanza di Support Vector Classification, che rappresenta un tipo di Support Vector Machine (SVM) con kernel lineare. Successivamente, ho optato per un'altra classe di SVM, il modello SGDClassifier (linear support vector machine), principalmente a causa delle limitazioni delle risorse computazionali.

Le SVM sono un tipo di algoritmo di apprendimento automatico utilizzato sia per problemi di classificazione che di regressione, specialmente per classificazio-

ni lineari. L'idea fondamentale è trovare l'iperpiano ottimale che divide lo spazio delle feature in modo da massimizzare il margine (distanza tra i punti più vicini di classi diverse) tra le diverse classi.

Il primo modello utilizzato è stato SVC con kernel lineare, mantenendo tutti gli altri parametri con i valori predefiniti. Ciò implica l'utilizzo di una SVM con soft-margin, come descritto nell'equazione 1. In questo modo, il modello consente alcuni errori di classificazione per ottenere una separazione più flessibile, specialmente quando si trattano dati che potrebbero non essere linearmente separabili o che contengono rumore.

$$L = \frac{1}{2}||\omega||^2 + \Sigma_i \lambda_i (y_i(\omega * x_i + b) - 1) \tag{1}$$

Il secondo modello utilizzato è stato l'SGDClassifier3, che implementa un classificatore lineare attraverso la discesa del gradiente stocastico (Stochastic Gradient Descent, SGD). Il termine "hinge" nell'ambito della funzione di perdita (loss2) indica che questo classificatore è configurato per l'addestramento utilizzando la perdita di cerniera (hinge loss), comunemente associata alle Support Vector Machines (SVM) per la classificazione binaria.

$$L = (y_i, f(x_i)) = max(0, 1 - y_i f(x_i)) \tag{2}$$

$$E(\omega, b) = \frac{1}{n} \sum_{i=1}^{n} L(y_i, f(x_i)) + \alpha R(\omega) \tag{3}$$

Nell'equazione 3 sono descritti i dettagli matematici della procedura SDG per una classificazione binaria. Dato un set di addestramento $(x_1, y_1) \dots (x_n, y_n)$ dove $x_i \in \mathbf{R}^m$ e $y_i \in \mathbf{R}$ ($y_i \in [1, -1]$), lo scopo è quello di apprendere una funzione di score lineare $f(x) = \omega^T x + b$ con i parametri del modello $\omega \in \mathbf{R}^m$ e intercetta $b \in \mathbf{R}$. Per effettuare previsioni per la classificazione binaria si guarda al segno della funzione di score $f(x)$. Per trovare i parametri del modello si va a minimizzare l'errore di addestramento regolarizzato. Nell'equazione R è il termine di regolarizzazione (penalità). $\alpha > 0$ è un iperparametro positivo che gestisce la forza della regolarizzazione.

Sono state implementate diverse funzioni al fine di non appesantire eccessivamente il codice. Le batterie di test sono state progettate in modo che per ogni iterazione venissero effettuati 5 addestramenti su diverse combinazioni di set di test e set di addestramento. Ogni volta veniva definito il parametro "random_state" del metodo "train_test_split" di scikit-learn.

Inizialmente è stata effettuata una prova con il modello SVC, ma è stata successivamente abbandonata a causa della lunga durata dell'addestramento. Si è quindi optato per procedere con gli esperimenti utilizzando SGDClassifier, che si è dimostrato molto più veloce. Ad ogni passo, è stata eseguita una batteria di test (addestramento, test e raccolta delle metriche) con un modello sul dataset che non conteneva i parametri relativi all'età della stella e un altro modello sul dataset che li includeva. Ciò ha consentito di confrontare l'andamento delle metriche applicando le stesse correzioni su entrambi i modelli.

La prima batteria di test è stata condotta sui dataset subito dopo la fase iniziale di preprocessing. La seconda ha applicato una normalizzazione dei dati attraverso la funzione "StandardScaler()"4.

$$X_{new} = \frac{X_i - X_{mean}}{StandardDeviation} \tag{4}$$

La terza ha modificato la tecnica di etichettatura, passando da un encoding binario alla convenzione di etichettatura bipolare. Infine, la quarta e la quinta batteria di test hanno esplorato tecniche di bilanciamento: la quarta ha eseguito un oversampling5 attraverso la tecnica SMOTE, fornita da imblearn, che genera nuovi esempi della classe minoritaria; la quinta e ultima batteria di test è stata implementata attraverso un undersampling utilizzando la tecnica di Random-under-sampler, anch'essa fornita da imblearn.

$$x_i' = x_i + \lambda(x_j - x_i) \tag{5}$$

La nuova istanza (x_i') è sintetizzata a partire dall'esempio (x_i) e dall'esempio (x_j) selezionato casualmente tra i vicini più prossimi di (x_i), dove (λ) è un valore casuale compreso nell'intervallo $[0, 1]$.

Infine, è stata implementata una funzione di plotting per visualizzare l'andamento di tutte le metriche considerate durante gli esperimenti, tra cui accuratezza, precisione, f1-score e recall.

3 Risultati sperimentali

Le metriche di valutazione adottate al fine di valutare la bontà del modello sono:

- **Accuratezza**: Proporzione totale di istanze correttamente classificate rispetto al numero totale di istanze (vero positivo + vero negativo) / (vero positivo + falso positivo + vero negativo + falso negativo).

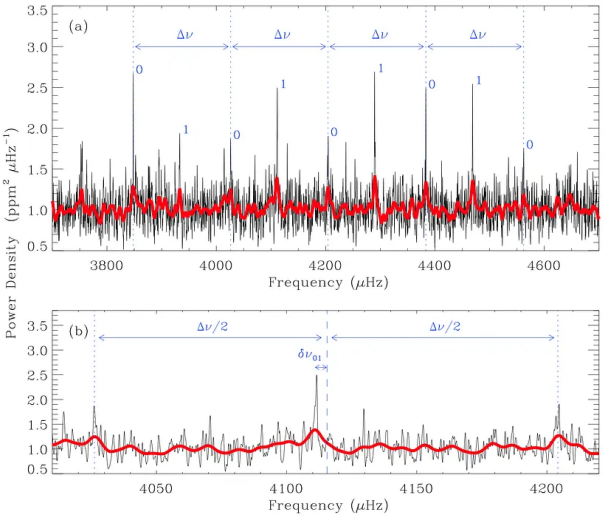

Figura 2: Oscillazione di Kepler-37, hanno consentito di determinare con precisione il raggio della stella.

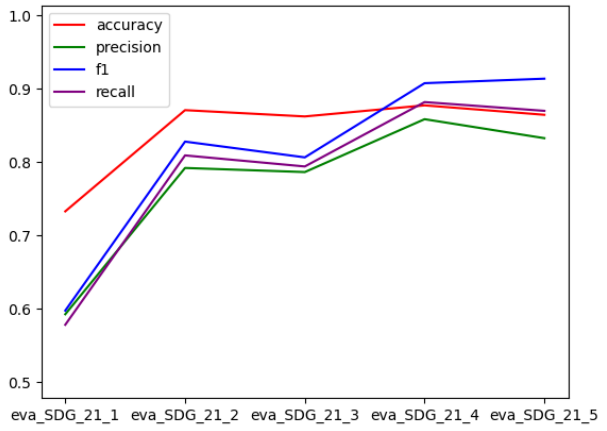

Figura 3: Andamento delle metriche per SGDClassifier per dataset senza età stellare.

- **Precisione**: Proporzione di istanze positive predette correttamente rispetto a tutte le istanze predette come positive (vero positivo / (vero positivo + falso positivo)).
- **Recall**: Proporzione di istanze positive predette correttamente rispetto a tutte le istanze effettivamente positive (vero positivo / (vero positivo + falso negativo)).
- **F1 score**: Media armonica tra precisione e recall (2 * (precisione * recall) / (precisione + recall)).

Entrambi i dataset erano limitati per quanto riguarda il numero di istanze, inoltre il dataset che include la feature relativa all'età stellare associata ai KOI era ancor più limitato rispetto all'altro. Entrambi i dataset erano fortemente sbilanciati: quello senza informazioni sull'età stellare contava 5951 elementi, con 1959 pianeti confermati e 3992 oggetti che non erano pianeti. D'altra parte, il dataset contenente le informazioni sull'età stellare era composto da 2336 istanze, di cui 1520 erano confermate come pianeti e 816 no. Quindi, oltre ad essere sbilanciati, i due dataset erano sbilanciati in maniera speculare l'uno rispetto all'altro.

Durante l'esperimento, il miglioramento più evidente è stato riscontrato tra la prima e la seconda batteria di esperimenti, grazie all'applicazione della normalizzazione dei dati. Con la modifica del sistema di etichettatura, l'accuratezza è rimasta più o meno stabile per entrambi i modelli. La differenza principale risiede nel fatto che, nel caso del modello a 21 feature (senza la variabile relativa all'età della stella), tutte le metriche hanno registrato un decremento, mentre per il modello a 22 feature si è osservato un lieve miglioramento per tutte le metriche considerate.

Per quanto riguarda gli esperimenti di oversampling e undersampling, l'uso dell'oversampling ha avuto un impatto positivo sul modello a 21 feature, migliorando leggermente anche l'accuratezza. Tuttavia, nel caso del modello a 22 feature, si è verificato un generale abbassamento di tutte le metriche, ad eccezione dell'accuratezza che ha registrato un leggero miglioramento.

L'ultimo esperimento, che ha coinvolto l'undersampling, ha avuto un impatto discretamente negativo su tutte le metriche del modello a 22 feature. D'altra parte, per il modello a 21 feature, l'abbassamento delle metriche è stato meno significativo, registrando addirittura un leggero miglioramento per l'F1-score.

4 Conclusioni

Lo scopo del progetto era evidenziare l'importanza della caratteristica relativa all'età della stella associata al KOI per la classificazione di quest'ultimo come esopianeta o meno. Sebbene gli esperimenti abbiano sofferto di una forte mancanza di istanze durante le fasi di addestramento, posso affermare che il dataset contenente il parametro in questione, nonostante avesse approssimativamente la metà delle istanze rispetto alla sua controparte a 21 feature, ha mostrato una buona performance.

È importante notare che l'accuratezza del modello a 21 feature è solamente circa il 4% superiore ri-

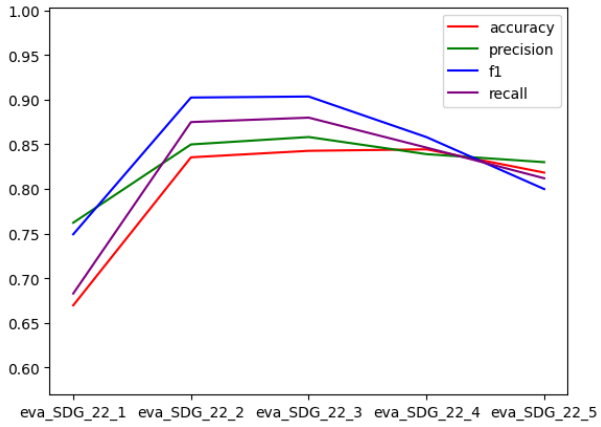

Figura 4: Andamento delle metriche per SGDClassifier per dataset con età stellare.

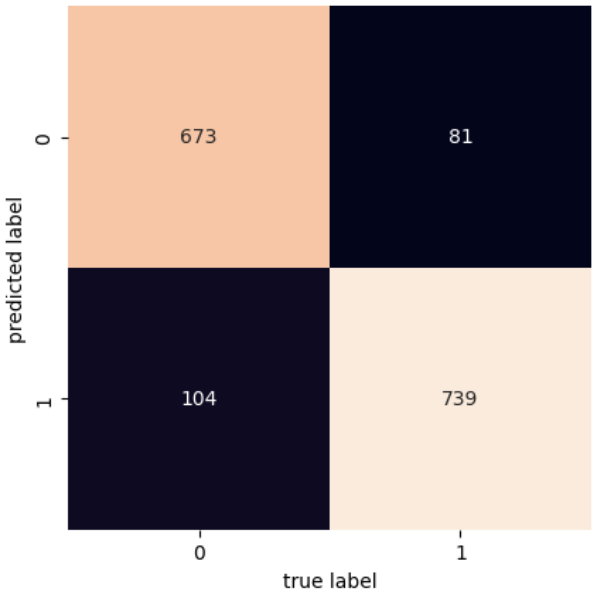

Figura 5: Matrice di confusione modello a 21 feature, esperimento 4.

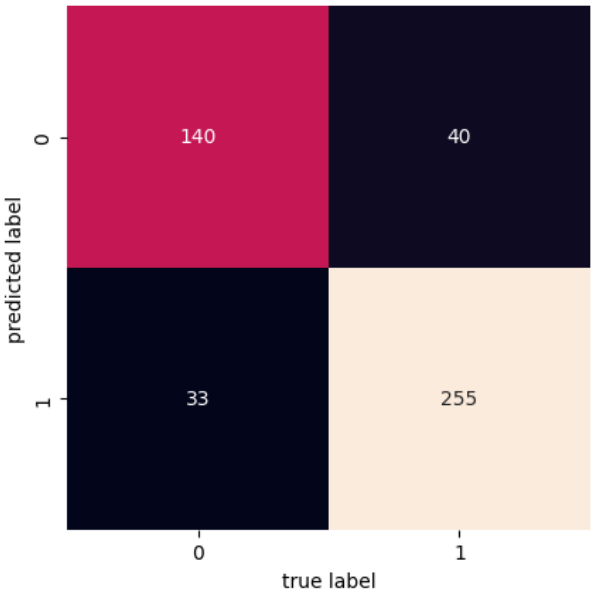

Figura 6: Matrice di confusione modello a 22 feature, esperimento 3.

spetto all'altro, il che è un incremento modesto considerando che ha quasi il doppio delle istanze per l'addestramento.

Un altro aspetto interessante riguarda gli esperimenti sul bilanciamento. Per quanto riguarda l'oversampling, il modello a 21 feature, a differenza di quello a 22, ha registrato un apprezzabile aumento delle metriche, probabilmente perché, avendo più dati, è stato in grado di generare istanze fittizie di migliore qualità. L'undersampling ha avuto un effetto negativo su entrambi i modelli, soprattutto sul modello a 22 feature, ma questo risultato non è sorprendente considerando la significativa carenza di istanze, anche nel modello a 21 che ne ha il doppio rispetto al modello a 22.

Intuitivamente, oltre al problema della mancanza di istanze, sembra esserci un'incertezza legata al rumore nei dataset. Al fine di ridurre la complessità dei calcoli e condurre più esperimenti possibili in tempi accettabili per apprezzare meglio le variazioni delle metriche di valutazione, ho eliminato 59 feature dal dataset. La scelta di mantenere o eliminare una caratteristica è stata basata su considerazioni personali. Riconosco che le mie competenze nel campo dell'astrofisica potrebbero non essere sufficienti per fare una selezione ottimale delle feature. Questo aspetto è intuitivamente collegato al rumore, poiché, nonostante la sicurezza nelle etichette corrette, non posso garantire di aver mantenuto feature importanti o che

si combinano bene per la classificazione. Di conseguenza, il modello potrebbe aver definito dei punti nell'iperspazio condizionato da feature non correlate tra loro, influenzando così la classificazione.

Come idee su come procedere sempre su questo tema, se avessi avuto più tempo e risorse computazionali, lavorerei dapprima operando una buona selezione delle feature. Partirei dal dataset originale e farei degli esperimenti mantenendo tutte le feature meno una in modo che a giro le provo tutte, poi manterrei quelle la cui mancanza ha avuto più effetto negativo sulle metriche di valutazione del modello. Ci sarebbe anche da lavorare su alcuni iperparametri, come, sempre per il SGDClassifier, il parametro alpha che governa la regolarizzazione, mettendolo a valori più bassi in modo da rendere il modello più soft-margin. Sarebbe poi interessante provare ad utilizzare modelli delle reti neurali in quanto possono imparare relazioni complesse tra le caratteristiche e la variabile target. Inoltre le reti neurali possono essere adattate a nuovi dati, il che le rende una buona scelta per la scoperta di esopianeti dato che man mano che vengono raccolte nuove osservazioni le reti neurali possono essere aggiornate.

5 Contributi

Ho lavorato al progetto in totale autonomia, occupandomi della ricerca dell'ambito a cui applicare le conoscenze che ho maturato durante il corso, della ricerca del set di dati, dell'implementazione del codice, della ricerca di articoli e studi pre-esistenti sulle argomentazioni che ho trattato durante il progetto e della stesura della relazione.

Riferimenti bibliografici

[1] David J Armstrong, Jevgenij Gamper e Theodoros Damoulas. "Exoplanet validation with machine learning: 50 new validated Kepler planets". In: *Monthly Notices of the Royal Astronomical Society* 504.4 (2021), pp. 5327–5344.

[2] William J Borucki. "Kepler: a brief discussion of the mission and exoplanet results". In: *Proceedings of the American Philosophical Society* 161.1 (2017), pp. 38–65.

[3] Adina Daniela Feinstein. "A Multi-Wavelength Investigation of Young Stellar and Planetary Systems". Tesi di dott. The University of Chicago, 2023.

[4] Veselin Kostov. "Discovery and characterization of transiting circumbinary planets from NASA's Kepler mission". Tesi di dott. Johns Hopkins University, 2014.

[5] Zuo-Lin Tu et al. "Convolutional Neural Networks for Searching Superflares from Pixel-level Data of the Transiting Exoplanet Survey Satellite". In: *The Astrophysical Journal* 935.2 (2022), p. 90.