

|  |
| --- |
| Classificazione di oggetti celesti tramite machine learning |
|  |
| Autore: Saladino Alessio |

# Sommario

1. Introduzione
2. Descrizione dei dataset
   1. Sloan Digital Sky Survey
   2. Star Dataset
3. Dettagli di implementazione
   1. Rielaborazione del dataset
   2. Addestramento
   3. Testing
4. Risultati
5. Introduzione

Lo scopo di questo caso di studio è la realizzazione di un sistema in grado di classificare oggetti celesti mediante l’utilizzo di diversi algoritmi di machine learning.

Sono stati presi in considerazione due dataset differenti:

* Sloan Digital Sky Survey Dataset
* Star Dataset

1. Descrizione dei dataset

In questa sezione si andranno ad analizzare le caratteristiche dei dataset utilizzati.

* 1. Sloan Digital Sky Survey

La *Sloan Digital Sky Survey* o *SDSS* è un'indagine conoscitiva del cielo effettuata con il telescopio ottico dedicato a grandangolo di 2,5 metri dell'osservatorio di Apache Point, New Mexico, Stati Uniti. La ricerca è effettuata in spettroscopia multi-filtro e rilevamento degli spostamenti verso il rosso. Al progetto è stato dato tale nome dopo il significativo contributo della fondazione Alfred P. Sloan.

Fonte: [Sloan Digital Sky Survey](https://it.wikipedia.org/wiki/Sloan_Digital_Sky_Survey)

Da questa indagine è stato realizzato un omonimo dataset pubblico, l’obiettivo del sistema realizzato è quello di classificare i vari oggetti celesti osservati all’interno di tre classi possibili: galassia, quasar e stella.

Il dataset presenta un totale di 17 features, di cui una dedicata all’id dell’oggetto e una feature di target.

Link al dataset: <https://www.kaggle.com/muhakabartay/sloan-digital-sky-survey-dr16>

* 1. Star Dataset

Dataset meno complesso del precedente, realizzato da un utente di Kaggle.

Lo Star Dataset consente la classificazione di diversi tipi di stelle. In particolare, la classificazione avviene all’interno di 6 classi possibili: nana marrone, nana rossa, nana bianca, sequenza principale, supergigante e ipergigante.

Il dataset presenta un totale di 7 feature, di cui una sola feature target rappresentante la classe della stella.

Link al dataset: <https://www.kaggle.com/deepu1109/star-dataset>

1. Dettagli di implementazione

Per la realizzazione del sistema si è proceduto nel seguente modo per entrambe le classificazioni svolte:

* 1. Rielaborazione del dataset

Come prima operazione è stato necessario eseguire delle operazioni di preprocessing sui dataset. Le operazioni svolte sono state la codifica dei valori non numerici, l’eliminazione dal dataframe di colonne superfue in fase di addestramento (es. l’id dell’oggetto) e la separazione delle feature target dalle feature di input.

Al termine di questa fase si sono ottenuti due array di feature X e y, l’array X conterrà le feature di input e l’array y le feature di output.

* 1. Addestramento

Si è scelto di eseguire l’addestramento di diversi modelli di machine learning, in modo da confrontare i diversi risultati e le diverse accuratezze calcolate su ciascun modello.

I modelli utilizzati sono:

* Albero di decisione
* Random Forest
* Rete neurale
* Naive Bayes

I dati calcolati gli array X e y sono stati suddivisi in 4 array:

1. X\_train: contiene le feature di input usate durante l’addestramento
2. y\_train: contiene le feature target usate durante l’addestramento
3. X\_test: contiene le feature di input usate durante il testing
4. y\_test: contiene le feature target usate durante il testing

Sono stati utilizzati anche due array aggiuntivi chiamati X\_trainNorm e X\_testNorm, contenenti i valori di X\_train e X\_test normalizzati in un valore compreso fra 0 e 1, questi array vengono utilizzati per l’addestramento e il testing della rete neurale.

Al termine di questa fase si sono ottenuti i modelli addestrati e pronti per essere testati.

* 1. Iperparametri

Sono stati utilizzati degli iperparametri per l’addestramento dell’albero di decisione e del random forest.

Gli iperparametri usati sono:

* Splitter: best,random
* Criterion: gini,entropy
* N\_estimators: 8,16,32 (solo per il modello random forest)

Usando il modulo GridSearchCV di python è stato possibile testare le varie combinazioni di iperparamentri, usando la cross validation dividendo in 5 parti il training set.

* 1. Testing

In questa fase è stata eseguita la classificazione degli elementi contenuti nell’array X\_test, mentre l’array y\_test è stato usato per determinare il valore di verità delle classificazioni.

Come risultato della classificazione si è ottenuto un array di predizioni calcolato da ogni modello.

Tale array è stato confrontato con i valori contenuti in y\_test; da questo confronto è stato possibile generare la matrice di confusione ed eseguire il calcolo dell’accuratezza di ciascun modello.

1. Risultati

Sono state eseguite un totale di 5 iterazioni del sistema che prevedevano un addestramento completo di ciascun modello ed un successivo calcolo dell’accuratezza; si è tenuto traccia dei risultati ottenuti i quali sono mostrati nelle seguenti tabelle (una per ogni dataset):

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Sloan Digital Sky Survey | | | | | | |
| Modello | Test 1 | Test 2 | Test 3 | Test 4 | Test 5 | Media |
| Albero Decisionale | 0.9869 | 0.9857 | 0.9861 | 0.9861 | 0.9859 | 0.9861 |
| Random Forest | 0.9904 | 0.9903 | 0.9897 | 0.9886 | 0.9907 | 0.9899 |
| Naive Bayes | 0.8978 | 0.9713 | 0.8175 | 0.9695 | 0.8760 | 0.9064 |
| Rete Neurale | 0.9774 | 0.9726 | 0.9607 | 0.9817 | 0.9775 | 0.9739 |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Star Dataset | | | | | | |
| Modello | Test 1 | Test 2 | Test 3 | Test 4 | Test 5 | Media |
| Albero Decisionale | 1 | 1 | 1 | 1 | 1 | 1 |
| Random Forest | 1 | 1 | 1 | 1 | 0.9791 | 0.9958 |
| Naive Bayes | 0.6041 | 0.5833 | 0.6250 | 0.6666 | 0.9791 | 0.6916 |
| Rete Neurale | 0.9583 | 0.8958 | 0.9791 | 0.8958 | 0.9375 | 0.9333 |