Università degli studi di
Milano-Bicocca

Advanced Machine Learning

Final Project

Prevedere le precipitazioni piovose mediante Deep Learning: il caso australiano.

Authors:
Carlone Alberto - 726894
Miori Davide - 813692
Ondei Alice - 826399

Sommario

L'elaborato presenta la costruzione di un modello di apprendimento profondo atto alla previsione delle condizioni meteorologiche australiane giorno su giorno. A tal fine è stata implementata una rete neurale *fully connected*, senza considerare il fattore temporale presente nel dato proposto. L'ottimizzazione del modello costruito ha portato a risultati soddisfacenti per quanto riguarda i giorni con assenza di pioggia e mediocri per i giorni di precipitazioni piovose.

1 Introduzione

Il fine dell'analisi qui proposta è stato quello di costruire un modello di apprendimento profondo in grado di prevedere, dai dati meteorologici raccolti sul territorio australiano, se il giorno successivo avrebbe piovuto oppure no, date le condizioni del giorno corrente.

In termini pratici si è trattato di risolvere un problema di classificazione binaria, avente un forte sbilanciamento delle classi a favore dei giorni senza precipitazioni piovose. Il dataset utilizzato, inoltre, presentava un'elevata numerosità di valori mancanti. A tutto questo si è aggiunto un esiguo numero di osservazioni per singola località di provenienza del dato; per questo motivo i record non sono stati considerati come cronologicamente ordinati ma come indipendenti.

Nella prima parte dell'elaborato si esplicita la strategia adottata al fine di risolvere il problema dei *missing value*, prioritario data la necessità di avere osservazioni complete per la costruzione della rete. La problematica dello sbilanciamento, invece, è stata affrontata applicando le usuali tecniche di ricampionamento, esposte nella seconda parte dell'elaborato. Risolte le due complicazioni, sono stati costruiti i modelli di apprendimento profondo *fully connected*, poi ottimizzati.

2 Dati

Il dataset utilizzato è fornito dalla piattaforma online Kaggle [1]. Esso contiene dati giornalieri relativi al meteo di alcune località australiane, in un arco di tempo che copre circa 10 anni; si compone di 145460 righe e 23 colonne.

Tabella 1: Descrizione delle variabili presenti nel dataset.

Variabile	**Descrizione**
Date	Espressa nel formato: AAAA - MM - GG.
Location	Luogo di rilevazione.
MinTemp	Temperatura minima registrata, espressa in °C.
MaxTemp	Temperatura massima registrata, espressa in °C.
Rainfall	Precipitazioni (in mm), nell'arco delle 24 ore.
Evaporation	Evaporazione (in mm)
Sunshine	Ore di luce.
WindGustDir	Direzione della raffica di vento più forte registrata.
WindGustSpeed	Velocità della raffica di vento più forte registrata, in km/h.
WindDir9am	Direzione del vento alle 9 del mattino.
WindDir3pm	Direzione del vento alle 3 di pomeriggio.
WindSpeed9am	Velocità del vento alle 9 del mattino, in km/h.
WindSpeed3pm	Velocità del vento alle 3 di pomeriggio, in km/h.
Humidity9am	Umidità rilevata alle 9 del mattino, in percentuale.
Humidity3pm	Umidità rilevata alle 3 del pomeriggio, in percentuale.
Pressure9am	Pressione atmosferica s.l.m. alle 9 del mattino, in hPa.
Pressure3pm	Pressione atmosferica s.l.m. alle 3 del pomeriggio, in hPa.
Cloud9am	Porzione di cielo coperta da nuvole alle 9 del mattino, su una scala da 0 a 8.
Cloud3pm	Porzione di cielo coperta da nuvole alle 3 del pomeriggio, su una scala da 0 a 8.
Temp9am	Temperatura alle 9 del mattino, in °C.
Temp3pm	Temperatura alle 3 del pomeriggio, in °C.
RainToday	Variabile binaria che indica se il giorno corrente le precipitazioni sono state almeno di 1 mm.
RainTomorrow	Variabile binaria che indica se il giorno seguente le precipitazioni sono state almeno di 1 mm.

La variabile target "RainTomorrow" è un attributo binario (Yes/No) che certifica se il giorno successivo alle rilevazioni riportate le piogge sono state superiori ad $1mm$. Le due classi che la costituiscono sono sbilanciate, dal momento che presentano una percentuale del 76% per la classe 'No' e del 22% circa per la classe 'Yes'. Il restante 2% sono valori mancanti.

Le osservazioni giornaliere sono state tratte dal sito del Bureau of Meteo-

rology australiano [2].

Come anticipato dalla Tabella 1 il dataset presenta una suddivisione per luogo di rilevazione, identificata dalla variabile "Location". Ognuna delle espressioni di questo attributo assume la forma di una serie storica a lunghezza variabile. Inoltre, non tutte le rilevazioni per località sono avvenute nello stesso arco temporale.

Al fine di comprendere meglio quanto la posizione geografica delle diverse località influisse sulle condizioni meteo, si è deciso di calcolare due probabilità per ciascuna location, partendo da una tabella di contingenza che incrociasse le variabili "RainToday" e "RainTomorrow". I due valori sono stati calcolati come segue:

$$P(RainTomorrow = Yes | RainToday = No) \tag{1}$$

$$P(RainTomorrow = Yes | RainToday = Yes) \tag{2}$$

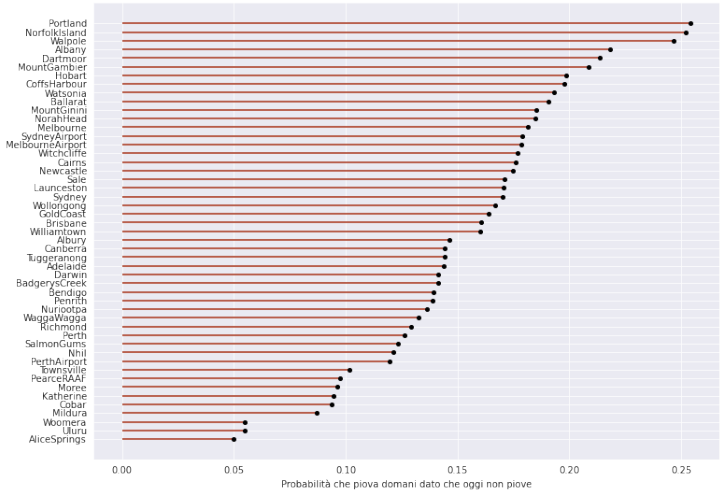

Figura 1: "Location" ordinate secondo la probabilità 1.

Le due probabilità calcolate sono state rappresentate nella Figura 1 e nella Figura 2. Dal risultato ottenuto è stato possibile constatare quanto il fattore geografico sia cruciale. Alice Springs, ad esempio, essendo una città situata in un territorio desertico, presenta una probabilità molto bassa di

pioggia per il giorno seguente quando nel giorno corrente non vi sono precipitazioni. Inversamente, quando ne sono presenti durante il giorno corrente, la probabilità che queste proseguano nel giorno successivo è superiore al 40%.

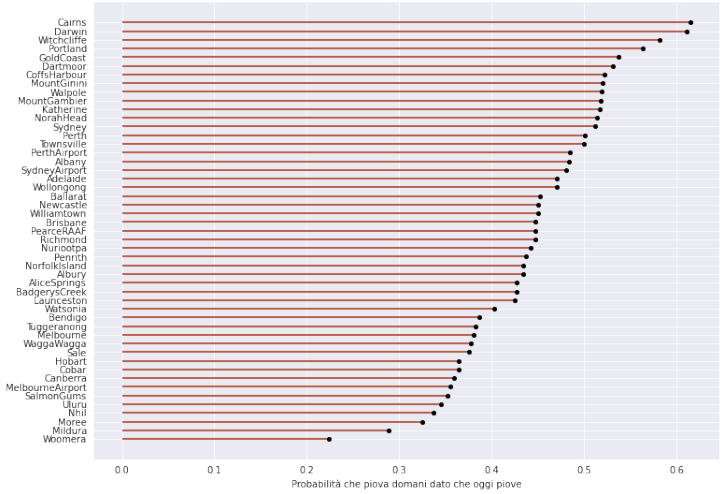

Figura 2: "Location" ordinate secondo la probabilità 2.

I dati analizzati avevano una forte presenza di valori mancanti, essi infatti rappresentavano circa il $10,26\%$ dell'intero dataset. La stessa fonte da cui provengono i dati, in una nota, ha dichiarato: *"Some cloud observations are from automated equipment; these are somewhat different to those made by a human observer and may not appear every day."* [3] e inoltre *"These observations have been taken from the Bureau of Meteorology's "real time" system. Most of the data are generated and handled automatically. Some quality checking has been performed, but it is still possible for erroneous values to appear. From time to time, observations will not be available, for a variety of reasons. Sometimes when the daily maximum and minimum temperatures, rainfall or evaporation are missing, the next value given has been accumulated over several days rather than the normal one day. It is very difficult for an automatic system to detect this reliably, so caution is advised"* [4].

Per questa ragione, la maggior parte del pre - processing effettuato è stato volto alla gestione della mancanza di informazione.

In prima battuta sono state rimosse le poche righe in cui risultavano mancanti le variabili "RainToday" e "RainTomorrow", fondamentali ai fini dell'analisi e difficilmente imputabili.

Prima di procedere alla gestione dei restanti valori mancanti si è indagata la natura degli outlier. Le due situazioni più degne di nota sono identificabili nelle variabili "Rainfall" ed "Evaporation", per cui è stato costruito un boxplot discriminando per la variabile di interesse al fine di comprendere se questi valori anomali fossero comunque informativi.

Come visibile dalla Figura 3, in entrambe le situazioni si è notato che ol-

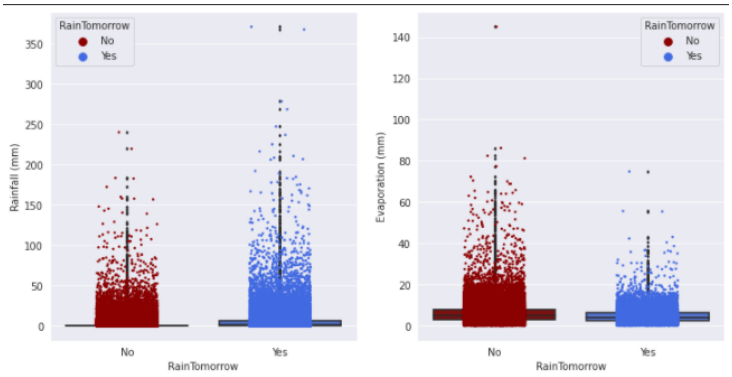

Figura 3: Boxplot per variabili "Rainfall" ed "Evaporation" in scala originale.

tre una certa soglia i punti appartengono ad una sola classe, perciò, poiché discriminanti, si è deciso di mantenere questi valori.

Tornando invece alla problematica dei missing, questa è stata affrontata seguendo tre diverse possibili strategie: la prima prevede di considerare esclusivamente le osservazioni che presentano dati completi, escludendo quindi tutti quei record aventi almeno un valore mancante. In questo modo si passa da un totale di 145460 righe a 56420. La perdita di informazione è quindi consistente.

Le altre due strategie tengono conto del fatto che la maggior parte dati mancanti appartiene alle colonne "Evaporation", "Sunshine", "Cloud3pm" e "Cloud9am". Tra queste la via più drastica ha previsto la rimozione di tutti gli attributi citati e la successiva rimozione di tutte le righe che ancora presentano almeno un missing, mantenendo un totale di 112925 righe.
L'ultima strategia invece ha comportato uno studio più approfondito circa l'origine del dato mancante ed una conseguente gestione a più fasi.

Sono state considerate nuovamente le colonne sopra citate, decidendo di selezionare le città aventi valori mancanti al 100% per tali variabili; questo perchè sarebbe stato pressoché impossibile procedere con un qualunque tipo

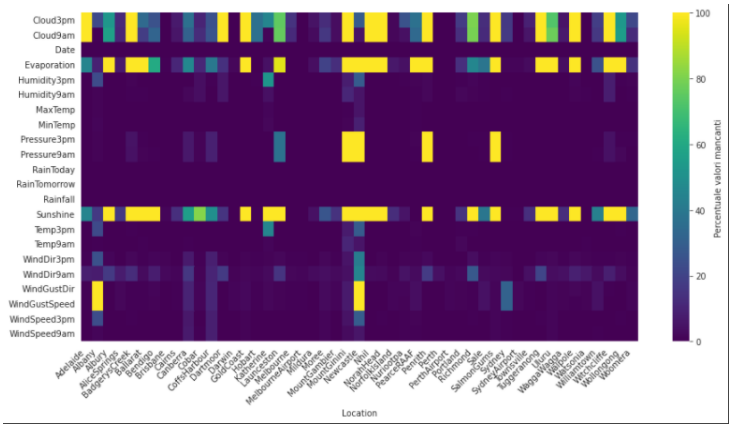

Figura 4: Percentuale missing value per la variabile "Location" rispetto le altre.

di imputazione dei dati. La Figura 4 mostra la presenza di valori mancanti incrociando le località alle variabili esplicative a disposizione, escludendo ovviamente la location e la data; si può notare come le quattro variabili di interesse si distinguano particolarmente.

Sono state così selezionate 22 location candidate alla rimozione su 49 totali. Dopo un'attenta analisi geografica, ci si è accorti che alcune località non rappresentavano vere e proprie città ma sobborghi o quartieri del medesimo luogo; altre invece erano altipiani, basi militari, monti o altre strutture. La maggior parte dei luoghi individuati come da rimuovere rientrano nella seconda casistica.

L'approfondimento svolto ha inoltre portato alla luce per Badgerys Creek e Newcastle, inizialmente inserite nella lista delle città da rimuovere, location a loro affini, e per questo motivo vengono mantenute. In definitiva, le località eliminate sono state 20. Infine, si sono decise di eliminare tutte quelle righe che presentavano un valore nullo almeno in 10 dei 23 attributi, considerando che tra essi vi erano anche "Location", data di rilevazione e l'indicazione di avvenute precipitazioni per la giornata corrente o successiva, per i quali si ha la certezza di avere l'informazione completa. Il dataset risultante si compone di 85108 righe, è stato quindi mantenuto circa il 60% delle osservazioni iniziali.

Si è proceduto dunque alla normalizzazione delle location nel dataset incorporando, come citato in precedenza, le località affini in una singola. Dopo queste analisi preliminari la percentuale di valori mancanti risulta pari

al 4.07% del nuovo dato. Questi dati sono stati poi imputati.

É stato effettuato uno splitting dei dati iniziale tra train e test set con proporzione $90\% - 10\%$, seguito da un ulteriore splitting sulla porzione di training per generare un set di validation sempre con proporzione $90\% - 10\%$.

A partire dal subset di training finale ottenuto è stata effettuata l'imputazione delle variabili numeriche utilizzando la funzione *IterativeImputer* che permette, attraverso l'uso di un regressore e partendo da un'inizializzazione dei valori mancanti (media, mediana, moda o un valore costante), di generarli iterativamente. La scelta, nel caso del dataset utilizzato, è ricaduta su un'inizializzazione effettuata sulla mediana e un regressore basato su alberi di decisione chiamato *ExtraTrees*.

É stato deciso di rimuovere direttamente le righe presentanti un valore mancante in una variabile categorica, questo sia per il numero esiguo sia per la difficoltà nell'imputare determinati valori con la moda: ad esempio le categoriche relative alla direzione del vento subirebbero una distorsione notevole se il valore imputato fosse semplicemente il più frequente. Chiaramente, un'imputazione che tenesse conto della sequenza temporale avrebbe portato all'individuazione di valori più veritieri di quelli utilizzati. Tuttavia, come anticipato, la grande quantità di valori mancanti e la scarsa numerosità delle singole serie ha spinto verso l'approccio sopra discusso.

Per evitare di imputare i dati di una location utilizzando quelli di una differente, l'assegnazione dei valori è avvenuta all'interno di un ciclo in cui veniva effettuato un subsetting del train set basato sulla location, per poi unificare nuovamente i dati. All'interno del medesimo ciclo viene applicata la trasformazione allenata sul training anche su validation e test set. In seguito sono state aggiunte le variabili categoriche non considerate durante l'imputazione ed è stato randomizzato l'ordine delle righe.

3 Approccio Metodologico

Per ciascuna delle 3 strategie di gestione dei missing, introdotte nel paragrafo precedente, si è proceduto alla costruzione di un modello da ottimizzare, seguendo uno schema comune. In primo luogo le variabili numeriche sono state standardizzate (per test e validation utilizzando le statistiche calcolate sul training set), successivamente è stato svolto il one - hot encoding per le variabili categoriche. Si ricorda che il problema di classificazione trattato è fortemente sbilanciato, per questo è stato applicato sia l'*oversampling*, uti-

lizzando l'algoritmo *SMOTE-NC* quindi sovracampionando le osservazioni della classe minoritaria (giorni in cui ha piovuto), sia l'*undersampling*, sottocampionando la classe maggioritaria (giorni in cui non ha piovuto). Come sarà discusso successivamente, l'*oversampling* ha mostrato alcune criticità e pertanto si è proseguito solamente con l'*undersampling*. Le nuove proporzioni sono riportate nella Tabella 2.

Tabella 2: Proporzione tra le classi

Metodo	No	Yes	No - undersampling	Yes - undersampling
Metodo 1	0,780	0,220	0,588	0,412
Metodo 2	0,778	0,222	0,588	0,412
Metodo 3	0,785	0,215	0,588	0.412

Si ricorda che il Metodo 1 è quello che ha previsto la rimozione di tutte le osservazioni aventi almeno un missing, il Metodo 2 è quello che ha previsto la rimozione di 4 colonne ed il Metodo 3 è quello in cui sono stati imputati gli NAs.

La rete sequenziale che è stata ottimizzata ha la seguente struttura comune:
- Input layer, con numero di neuroni pari a 64 o 128, dimensione dell'input pari al numero di attributi e ReLu come funzione di attivazione;
- Layer di dropout, con valori di probabilità compresi tra 0.1 e 0.7 e step di 0.1;
- Layer denso hidden con numero di neuroni pari a 32 o 64 e ReLu come funzione di attivazione;
- Layer di dropout, con valori di probabilità compresi tra 0.1 e 0.7 e step di 0.1;
- Ulteriore layer denso con numero di neuroni pari a 16 o 32 e ReLu come funzione di attivazione;
- Layer di dropout, con valori di probabilità compresi tra 0.1 e 0.7 e step di 0.1;
- Layer di output con funzione di attivazione sigmoide.

L'ottimizzatore utilizzato è Adam specificando il learning rate pari a 0.0001. Si utilizza come funzione di loss la binary crossentropy.

Per ogni modello quindi sono stati ottimizzati il numero di neuroni dei layer densi e la percentuale di neuroni esclusi dai layer di dropout.

Per effettuare il tuning di questi iperparametri è stata utilizzata la libreria keras_tuner [5] che ha permesso, attraverso la definizione di un *hypermodel*, di individuare i valori degli iperparametri migliori rispetto ad una specifica funzione obiettivo. È stato scelto *Hyperband* [6] come *hypermodel* e la statistica F_1 del validation set come funzione obiettivo. *Hyperband* allena molti modelli con diverse combinazioni di iperparametri per poche epoche, confrontandoli a coppie e mantenendone la metà per il round successivo, alla stregua di una competizione sportiva ad eliminazione. Gli iperparametri così ottimizzati sono stati utilizzati per allenare il modello finale. Per i tre modelli finali il batch size è stato di 32 e la rete è stata allenata per 100 epoche. È stato inoltre aggiunto anche un *early stop* che tenesse conto del valore della funzione di loss sul validation set.

4 Risultati

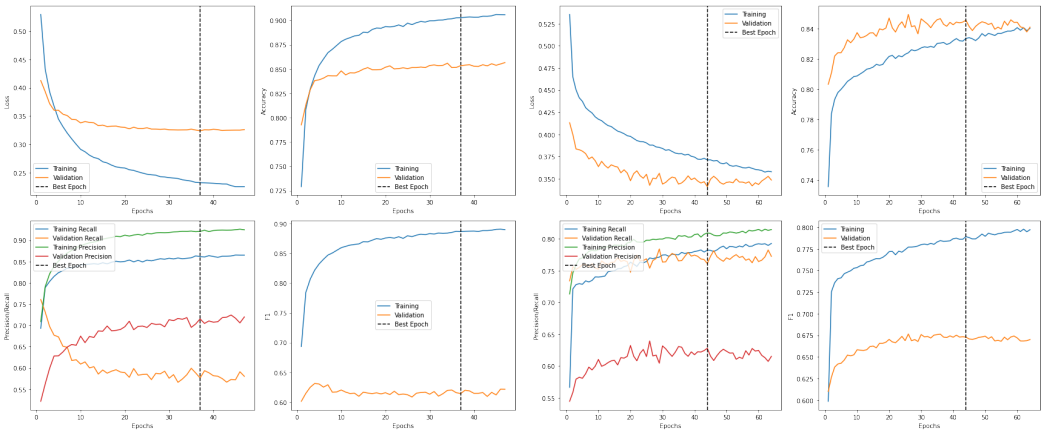

Figura 5: Sulla sinistra: andamento delle metriche di interesse per training e validation utilizzando un dataset oggetto di *oversampling*. Sulla destra: andamento delle metriche di interesse per training e validation utilizzando un dataset oggetto di *undersampling*. Entrambi i grafici fanno riferimento alla strategia 2.

Tabella 3: Model Evaluation. Si ricorda che Yes = 1 e No = 0

Metodo	Rec 1	Rec 0	Prec 1	Prec 0	F_1 1	F_1 0	F_1w	AUC
Model 1	0,787	0,852	0,600	0,934	0,681	0,891	0,845	0,819
Model 2	0,760	0,867	0,619	0,927	0,682	0,896	0,849	0,814
Model 3	0,754	0,870	0,613	0,928	0,677	0,898	0,850	0.812

5 Discussione

La Figura 5 mostra, da sinistra verso destra, le curve relative alle metriche ottenute mediante allenamento con *oversampling* e *undersampling*, sia per il training sia per il validation, con l'unica differenza che i modelli con *undersampling* sono stati ottimizzati. La ragione per cui l'ottimizzazione non è stata eseguita anche per i modelli dopo *oversampling* risiede nel fatto che, in seguito ai primi tentativi di costruzione di una rete neurale, si è notato come le curve della precision e della recall nel validation set avessero un comportamento inusuale. La curva della seconda metrica aveva un andamento decrescente già nelle prime epoche mentre la prima presentava un andamento opposto portando l'F_1 score a mantenere costante la sua andatura. Si è ritenuto che tale comportamento fosse dovuto alla generazione fittizia dei dati per la classe minoritaria propria degli algoritmi di *oversampling*. Per questo, nonostante la strategia di *undersampling* implichi una riduzione del numero di osservazioni nella classe maggioritaria - portando quindi ad avere un numero totale di record particolarmente esiguo rispetto all'*oversampling* - il sottocampionamento è stato preferito in quanto in grado di evitare la generazione di dati considerati fallaci per l'addestramento corretto della rete.

Il grafico relativo a tali modelli mostra un andamento comune delle curve di precision e recall che crescono leggermente al passare delle epoche. Inoltre, l'ottimizzazione è stata realizzata impostando come metrica obiettivo F_1 score, dunque un andamento opposto della precision e della recall avrebbe portato tale procedura ad ottimizzare una metrica che resta costante.

La Tabella 3 mostra i valori ottenuti nel test set dai modelli ottimizzati per le tre strategie proposte. Il modello 2 e il modello 3 presentano le stesse performance per F_1 score ed AUC mentre il modello 1 si distingue dai precedenti per una recall più alta e una precision inferiore. La scelta del modello migliore potrebbe essere presa sulla base della necessità di prevedere

con una certa preferenza se il giorno seguente pioverà o meno. Nel caso in cui si volesse adottare un criterio oggettivo, a parità di risultati ottenuti per le metriche target, varrebbe il criterio di parsimonia e pertanto il modello 2, utilizzando meno variabili esplicative, sarebbe preferibile ai restanti. Tale modello è riuscito a classificare, con una buona confidenza, le giornate in cui non pioverà al contrario della previsione delle giornate di pioggia che hanno mostrato risultati mediocri.

Al fine di migliorare le performance e i risultati ottenuti è possibile valutare diversi aspetti: in primo luogo, i dati meteorologici utilizzati hanno delle grandi carenze in termini di variabili a disposizione. Lo stato dell'arte delle previsioni meteorologiche prevede l'utilizzo combinato di dati rappresentanti le condizioni al suolo, in quota e dati satellitari. Nel dataset qui proposto sono presenti soltanto alcune delle più importanti variabili relative alle condizioni atmosferiche al suolo. In aggiunta a ciò, tali rilevazioni sono riportate, nel migliore dei casi, in riferimento a soli due momenti della giornata. Come riportato nella Tabella 1, si hanno a disposizione unicamente i dati relativi alle 9 del mattino o alle 3 del pomeriggio. Ne consegue che vengono ignorati i valori assunti dalle variabili nelle 18 ore precedenti al momento della previsione, a fronte del fatto che il momento di rilevazione delle precipitazioni è proprio alle 9 del mattino.

Si specifica inoltre che, anche nei modelli classici di previsione meteorologica (*Numerical Weather Prediction - NWP*), la variabile "Cloud" è tra quelle con maggior potere esplicativo. Sfortunatamente la documentazione relativa ai dati utilizzati riporta in modo esplicito la presenza di un bias, dato dal fatto che alcune osservazioni provengono da apparecchiature automatizzate altre da un osservatore umano, specificando che le prime vengono svolte in maniera differente e che possono non apparire tutti i giorni.

Oltre a queste problematiche, avere a disposizione più dati per singola location permetterebbe di allenare una rete per singola località, avendo informazioni più specifiche e adeguate in riferimento al tipo di clima caratteristico. Effettuare uno split in questo senso, quindi raggruppando location affini, non è stato possibile per via della quantità di dati necessaria al fine di allenare una rete neurale e per il problema dello sbilanciamento delle classi, con la conseguente necessità di un sottocampionamento.

Avere dati a granularità più elevata potrebbe inoltre introdurre la possibilità di considerare i dati come una serie storica, di conseguenza far riferimento ad un'altra architettura di rete neurale in grado di prendere in considerazione anche la componente temporale del dato, ovvero le *recurrent neural network*.

6 Conclusioni

La natura dei risultati ottenuti ha portato per i modelli sotto campionati a preferire quello più parsimonioso, il quale di fatto garantisce la raccolta di un numero minore di attributi per una futura previsione. I valori delle performance ottenute per questo modello mostrano una differenza sostanziale nella previsione dei giorni di pioggia rispetto ai giorni in cui non piove poiché per riconoscere le giornate piovose sono fondamentali i dati rilevati in quota, come ad esempio gli spostamenti delle masse d'aria calda e fredda. Per concludere, il modello selezionato mostra delle performance accettabili vista la natura del dato raccolto, che presentava oltre a numerosi valori mancanti anche inaccortezze nella rilevazione dell'informazione.

Riferimenti bibliografici

[1] https://www.kaggle.com/jsphyg/weather-dataset-rattle-package.

[2] http://www.bom.gov.au/climate/data. Copyright Commonwealth of Australia 2010, Bureau of Meteorology.

[3] http://www.bom.gov.au/climate/dwo/idcjdw2801.latest.shtml.

[4] http://www.bom.gov.au/climate/dwo/idcjdw0000.shtml.

[5] T. O'Malley, E. Bursztein, J. Long, F. Chollet, H. Jin, L. Invernizzi *et al.*, "Kerastuner," https://github.com/keras-team/keras-tuner, 2019.

[6] L. Li, K. Jamieson, G. DeSalvo, A. Rostamizadeh, and A. Talwalkar, "Hyperband: A novel bandit-based approach to hyperparameter optimization," *Journal of Machine Learning Research*, vol. 18, no. 185, pp. 1–52, 2018. [Online]. Available: http://jmlr.org/papers/v18/16-558.html

[7] M. Schultz, C. Betancourt, B. Gong, F. Kleinert, M. Langguth, L. Leufen, A. Mozaffari, and S. Stadtler, "Can deep learning beat numerical weather prediction?" *Phil. Trans. R. Soc.*, no. 379, 2021. [Online]. Available: https://doi.org/10.1098/rsta.2020.0097