

Realtime weather forecast attraverso il modello ARIMA

Università di Bologna
Laurea Magistrale in Informatica
Corso di Scalable and Cloud Programming
A.A. 2018-2019

Santilli Paolo
842785
paolo.santilli2@atudio.unibo.it

Prof. Zavattaro Gianluigi

Problema del weather forecast

Il problema della previsione del tempo si divide in tre fasi:

1. Osservazione e analisi -> si prendono i dati storici e si analizzano
2. Estrazione -> dai dati si estraggono determinati parametri, come temperatura minima e massima, ore di luce solare ecc.
3. Previsione -> i parametri estratti vengono dati in pasto ad un modello di previsione

ARIMA (Autoregressive integrated moving average)

Per modelli ARIMA si intende una particolare tipologia di modelli atti ad indagare serie temporali storiche che presentano caratteristiche particolari.

I modelli ARIMA sono generalmente denotati come $ARIMA(p,d,q)$ dove:

- p è l'ordine del modello autoregressivo (**AR**)
- d è il grado di differenziazione (il numero di volte in cui ai dati sono stati sottratti valori passati) (**I**)
- q è l'ordine del modello a media mobile (**MA**)

Perchè ARIMA?

1. Perchè è un modello adatto per risolvere il problema della previsione del tempo, in quanto si presta perfettamente alla fase di estrazione e di predizione discussi sopra.
2. Perchè esiste una libreria Spark che permette l'utilizzo di questo modello.

Realtime weather forecast

1) Osservazione e analisi

Le città su cui è stata svolta la previsione sono le principali città Australiane

Sydney	Wollongong
Canberra	Brisbane
Adelaide	Gold Coast
Hobart	Cairns
Darwin	Perth
Melbourne	



Realtime weather forecast

- I dati storici meteorologici sono stati scaricati dal sito meteorologico Australiano:
 - BOM(Bureau Of Meteorology) : <http://www.bom.gov.au/climate/dwo/>
- Su questo sito sono presenti i dati di tutte le città Australiane e sono scaricabili senza nessun tipo di limitazione.
- I dati si possono scaricare sia in formato .pdf che in formato .csv. In questo caso il formato scaricato è il .csv .
- I file .csv contengono le informazioni mensili del tempo divise giorno per giorno.

Realtime weather forecast

2) Estrazione

- I file scaricati contengono informazioni sul meteo giornaliero del mese di interesse, quindi per ogni giorno sono presenti tante informazioni tra cui quelle che ci servono, quindi in questa fase si è proceduto estraendo solo i campi di cui abbiamo bisogno che sono :
 - Data
 - Temperatura massima
 - Temperatura minima
 - Piovosità
 - Luce Solare
 - Umidità relativa
 - Copertura nuvolosa
 - Pressione

Realtime weather forecast

3) Predizione

- Per ogni file i dati quindi vengono estratti e messi insieme, per poi essere dati in pasto al modello ARIMA.
- Quindi al modello ARIMA vengono passati i dati divisi per tipologia (luce solare, temperatura media, ecc.) sotto forma di serie temporale.
- Il modello darà in output la predizione di un nuovo valore.

Implementazione in Scala

Realtime weather forecast è stato implementato nel linguaggio Scala utilizzando 4 oggetti:

- CommonData -> oggetto che contiene alcune delle strutture di dati comuni e costanti utilizzate in vari punti dell'applicazione
- FileOperations -> oggetto contenente metodi per il caricamento dei dati da file ed altre operazioni su questi
- PredictFunction -> oggetto contenente metodi che organizzano le funzioni utilizzate nell'esecuzione della previsione
- PredictWeather -> punto di accesso all'applicativo

Parallelizzazione con Spark

- La libreria con Spark è stata utilizzata per caricare i dati dei file scaricati nel cluster.
- Nel cluster di n unità di elaborazione vengono distribuiti i dati degli n giorni, in modo che ad ogni unità venga assegnata un giorno.
- Inoltre Spark è stato utilizzato per recuperare i dati in output dal modello di previsione, organizzando il cluster come precedentemente fatto.
- Infine si è proceduto con il salvare i risultati in un file di testo

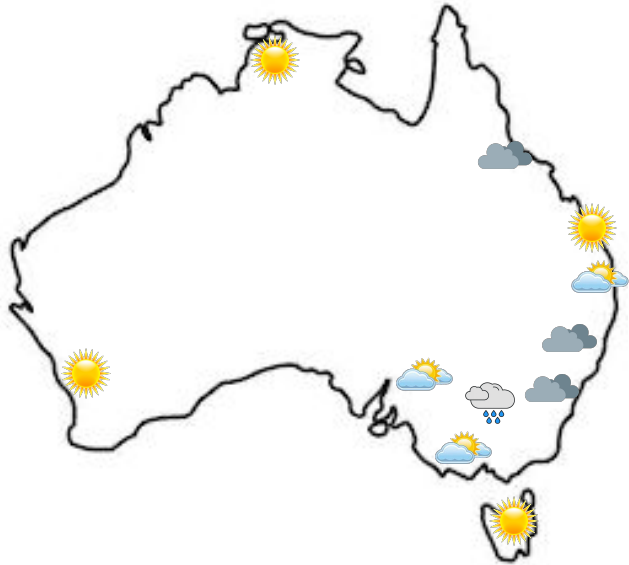
Amazon Web Services (AWS)

- Il progetto è stato esportato nel formato JAR con l'utility sbt-assembly
- Il file JAR è caricato su AWS attraverso il servizio S3
- Il cluster è avviato utilizzando EMR Sono stati utilizzati nodi del tipo x5.large e sono stati creati cluster con 1 unità di calcolo.
- Attraverso l'applicazione Putty e utilizzando una chiave è stata fatta la connessione da terminale al cluster
- Il JAR è copiato nel cluster

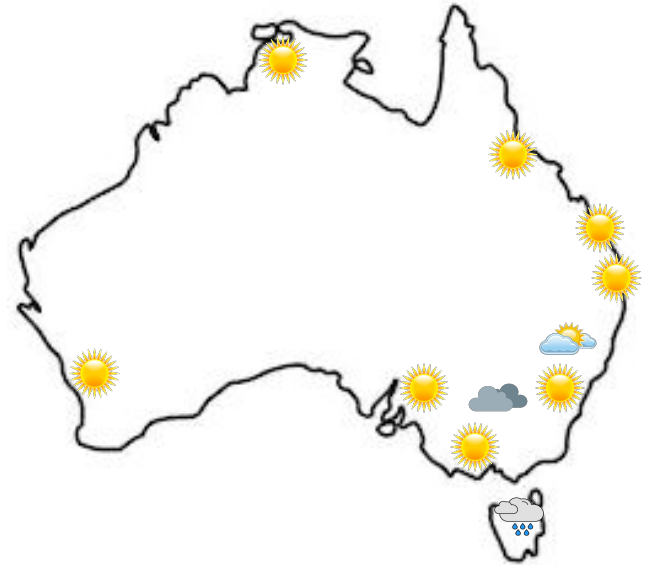
Risultati

Questi sono i risultati del 13 Aprile 2020 per quanto riguarda il tempo

Previsione



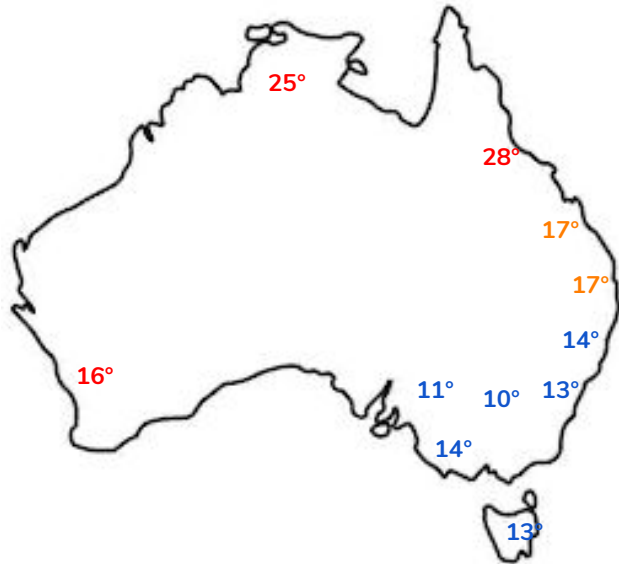
Realtà



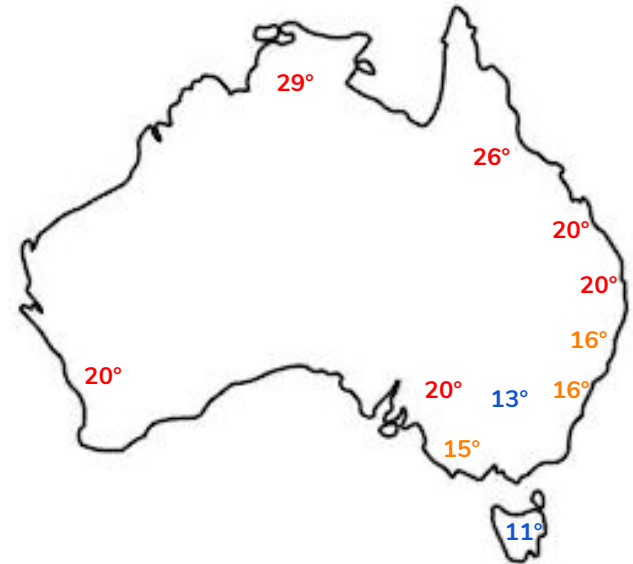
Risultati

Questi sono i risultati del 13 Aprile 2020 per quanto riguarda la temperatura media

Previsione



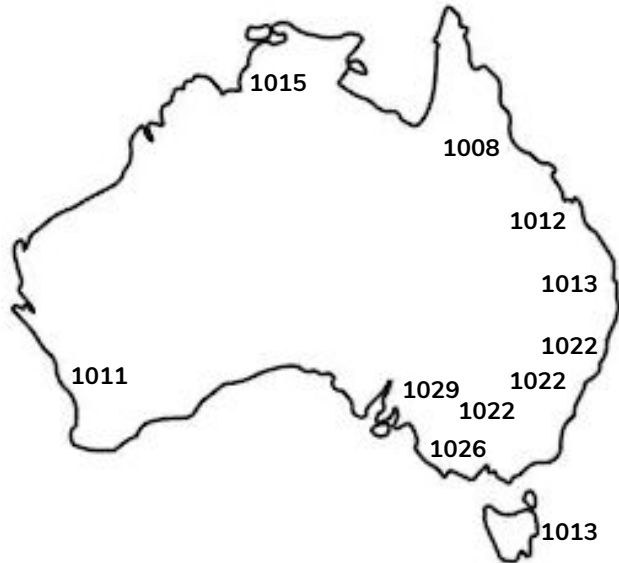
Realtà



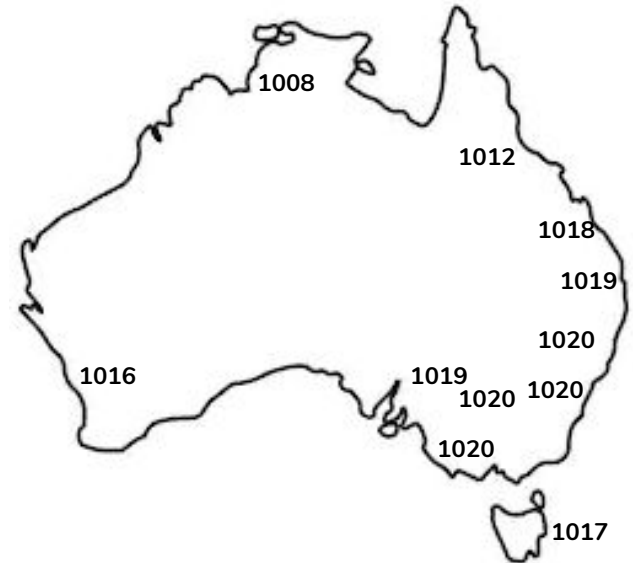
Risultati

Questi sono i risultati del 13 Aprile 2020 per quanto riguarda la pressione

Previsione



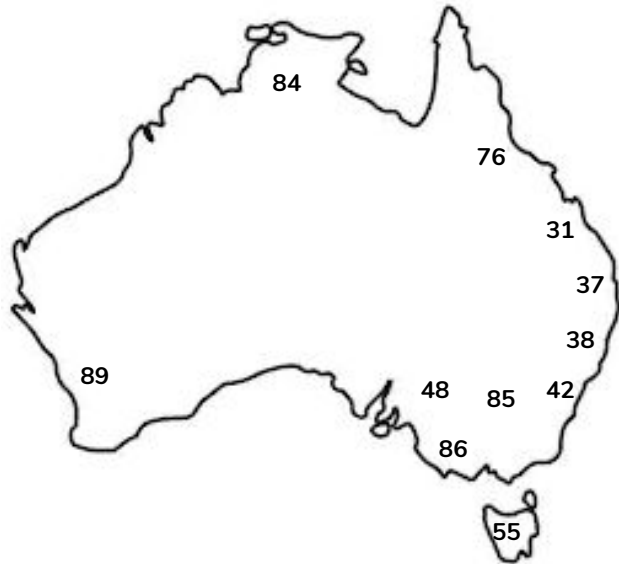
Realtà



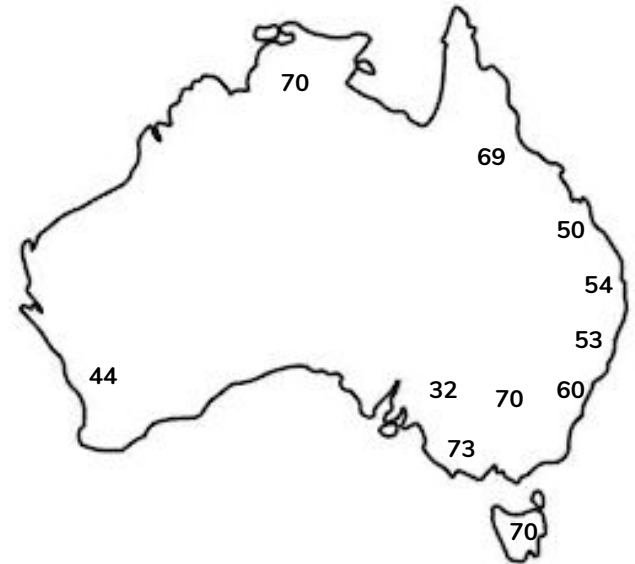
Risultati

Questi sono i risultati del 13 Aprile 2020 per quanto riguarda l'umidità relativa

Previsione



Realtà



Conclusioni

I risultati mostrano un evidente imprecisione tra i dati reali e quelli previsti. Questo può essere dato dalla poca quantità data in input al modello di previsione in quanto sono stati forniti i dati solamente del mese di marzo 2020 e dei primi dodici giorni di Aprile.

Per concludere, in questo progetto si è proceduto nella realizzazione di un applicativo che ha come obiettivo quello di sfruttare il modello di previsione ARIMA per la previsione del meteo. Si è proceduto con l'implementazione tramite linguaggio Scala e l'ausilio della libreria Spark. Dopodichè si sono effettuati test sulla piattaforma AWS.

Come sviluppo futuro si potrebbe sicuramente aumentare il volume di dati da dare al modello così da rendere la previsione certamente più accurata.