



UNIVERSITÀ DEGLI STUDI DI MILANO - BICOCCA
Scuola di Scienze
Dipartimento di Informatica, Sistemistica e Comunicazione
Corso di laurea in Informatica

Progetto

Corso: Machine Learning

Relazione di:
Davide Finati 817508
Fabio Beltramelli 816912
Alessandro Capelli 816302

Anno Accademico 2019-2020

1 Introduzione

L'obiettivo di questo elaborato consiste nel presentare gli aspetti fondamentali rilevati durante lo svolgimento del progetto. Tale progetto consiste nella scelta di un dataset distribuito in forma gratuita da Kaggle (raggiungibile al seguente indirizzo: <https://www.kaggle.com/jsphyg/weather-dataset-rattle-package>), in una analisi dei dati presenti, nello sviluppo di diversi modelli di Machine Learning e nella valutazione degli stessi. La scelta del dataset è ricaduta su "Rain in Australia" che contiene i dati giornalieri sul meteo in Australia rilevato da diverse stazioni meteorologiche. In particolare, il dominio applicativo del dataset consiste in una serie di dati inerenti alle condizioni meteorologiche rilevate da molteplici sistemi automatizzati, i quali monitorano quotidianamente, in diverse zone dell'Australia, la situazione in termini di presenza o meno di piogge. L'obiettivo del progetto è stato quello di definire modelli di Machine Learning in grado di predire, a partire da una istanza iniziale diversa da quelle contenute nel dataset, la presenza o meno di piogge sul territorio Australiano.

2 Dataset

Il dataset considerato è scaricabile al seguente indirizzo: <https://www.kaggle.com/jsphyg/weather-dataset-rattle-package>

Il dataset è formato da 142193 osservazioni. Le features presenti nel dataset completo sono 24 e hanno i seguenti significati:

- Date: data in cui è stata raccolta la rilevazione. Il valore dell'attributo varia da 1 Novembre 2007 a 25 Giugno 2017 (circa 10 anni di osservazioni)
- Location: città in cui è stata raccolta la rilevazione (esistono diverse stazioni meteorologiche). Il valore dell'attributo assume i seguenti valori: Canberra 2%, Sydney 2%, altri valori (47) 95%
- MinTemp: temperatura minima rilevata nella giornata. L'unità di misura di riferimento è il grado Celsius. Il valore dell'attributo assume 390 valori univoci
- MaxTemp: temperatura massima rilevata nella giornata. L'unità di misura di riferimento è il grado Celsius. Il valore dell'attributo assume 506 valori univoci
- Rainfall: quantità di pioggia rilevata nella giornata. L'unità di misura di riferimento è il millimetro. Il valore dell'attributo assume i seguenti valori: 0 (63%), 0.2 (6%), altri valori (678, 30%)
- Evaporation: quantità di acqua evaporata nelle precedenti 24 ore. L'unità di misura di riferimento è il millimetro
- Sunshine: numero di ore di luce nella giornata
- WindGustDir: direzione del vento più forte delle precedenti 24 ore
- WindGustSpeed: velocità del vento più forte delle precedenti 24 ore. L'unità di misura di riferimento è il km/h

- WindDir9am: direzione del vento alle 9 del mattino
- WindDir3pm: direzione del vento alle 3 del pomeriggio
- WindSpeed9am: velocità del vento alle 9 del mattino
- WindSpeed3pm: velocità del vento alle 3 del pomeriggio
- Humidity9am: percentuale di umidità rilevata alle 9 del mattino
- Humidity3pm: percentuale di umidità rilevata alle 3 del pomeriggio
- Pressure9am: pressione atmosferica rilevata alle 9 del mattino. L'unità di misura di riferimento è hPa (centinaia di Pascal)
- Pressure3pm: pressione atmosferica rilevata alle 3 del pomeriggio. L'unità di misura di riferimento è hPa (centinaia di Pascal)
- Cloud9am: frazione di cielo oscurato dalle nuvole alle 9 del mattino. L'unità di misura di riferimento è Okta
- Cloud3pm: frazione di cielo oscurato dalle nuvole alle 3 del pomeriggio. L'unità di misura di riferimento è Okta
- Temp9am: temperatura rilevata alle ore 9 del mattino. L'unità di misura di riferimento è il grado Celsius
- Temp3pm: temperatura rilevata alle ore 3 del pomeriggio. L'unità di misura di riferimento è il grado Celsius
- RainToday: indica la presenza o meno di pioggia per il giorno attuale. Il valore dell'attributo è rappresentato da "Yes" e "No"
- RISK_MM: è il target di un ipotetico modello di regressione; questo attributo è stato rimosso come richiesto dal proprietario del dataset
- RainTomorrow: indica la presenza o meno di pioggia per il giorno successivo. Il valore dell'attributo è rappresentato da "Yes" e "No"

Il target è rappresentato dall'attributo "RainTomorrow". Gli attributi "Rain Today" e "Rain Tomorrow" sono booleani, quindi come prima trasformazione, è stato deciso di convertirli da stringhe "Yes" e "No" in 0 e 1 rispettivamente; successivamente sono stati convertiti a fattori così da evitare che vengano trattati come dati numerici. Durante una prima esplorazione del dataset, è stato possibile rilevare la presenza di alcuni attributi contententi valori "NA" e ciò ha portato ad un'analisi più approfondita:

```
"Date | Number of NA: 0 | % of NA: 0"
"Location | Number of NA: 0 | % of NA: 0"
"MinTemp | Number of NA: 637 | % of NA: 0.447982671439522"
"MaxTemp | Number of NA: 322 | % of NA: 0.226452778969429"
"Rainfall | Number of NA: 1406 | % of NA: 0.988796916866512"
"Evaporation | Number of NA: 60843 | % of NA: 42.7890261827235"
"Sunshine | Number of NA: 67816 | % of NA: 47.6929244055615"
```

```

"WindGustDir | Number of NA: 9330 | % of NA: 6.5615044341142"
"WindGustSpeed | Number of NA: 9270 | % of NA: 6.51930826411989"
"WindDir9am | Number of NA: 10013 | % of NA: 7.04183750254935"
"WindDir3pm | Number of NA: 3778 | % of NA: 2.65695217064131"
"WindSpeed9am | Number of NA: 1348 | % of NA: 0.948007285872019"
"WindSpeed3pm | Number of NA: 2630 | % of NA: 1.8495987847503"
"Humidity9am | Number of NA: 1774 | % of NA: 1.24760009283157"
"Humidity3pm | Number of NA: 3610 | % of NA: 2.53880289465726"
"Pressure9am | Number of NA: 14014 | % of NA: 9.85561877166949"
"Pressure3pm | Number of NA: 13981 | % of NA: 9.83241087817262"
"Cloud9am | Number of NA: 53657 | % of NA: 37.7353315564057"
"Cloud3pm | Number of NA: 57094 | % of NA: 40.1524688275794"
"Temp9am | Number of NA: 904 | % of NA: 0.635755627914173"
"Temp3pm | Number of NA: 2726 | % of NA: 1.91711265674119"
"RainToday | Number of NA: 1406 | % of NA: 0.988796916866512"
"RISK_MM | Number of NA: 0 | % of NA: 0"
"RainTomorrow | Number of NA: 0 | % of NA: 0"

```

Questa analisi ha portato alla rimozione delle variabili con un'elevata percentuale di valori nulli, quali "Evaporation" (42.78%), "Sunshine" (47.68%), "Cloud9am" (37.73%) e "Cloud3pm" (40.15%). Tali variabili sono state rimosse in quanto, avendo un numero così elevato di valori nulli è stato preferito evitare di introdurre troppi valori artificiali, preferendo perdere tali informazioni. Sono state poi rimosse tutte le osservazioni contenenti valori nulli anche per tutte le altre variabili sempre per evitare di introdurre dati artificiali nel dataset. Alla fine di questa operazione è stato ottenuto il dataset "pulito" sul quale sono state eseguite analisi di vario tipo e successivamente è stato diviso per poter allenare (70%) e testare (30%) i modelli.

3 Analisi delle covariate e PCA

La prima fase di analisi è consistita nella valutazione delle distribuzioni a cui sono soggette le feature presenti nel dataset "pulito"; in particolare sono state attribuite delle possibili distribuzioni tramite l'utilizzo di test basati su asimmetria e curtosi che hanno aiutato nella scelta e hanno contribuito a identificare situazioni di particolare interesse. Successivamente sono state svolte le analisi sulla matrice di correlazione con successiva fase di feature selection. Infine è stata applicata PCA per confermare i risultati ottenuti precedentemente e per verificare l'assenza di ulteriori anomalie negli attributi.

3.1 Distribuzioni

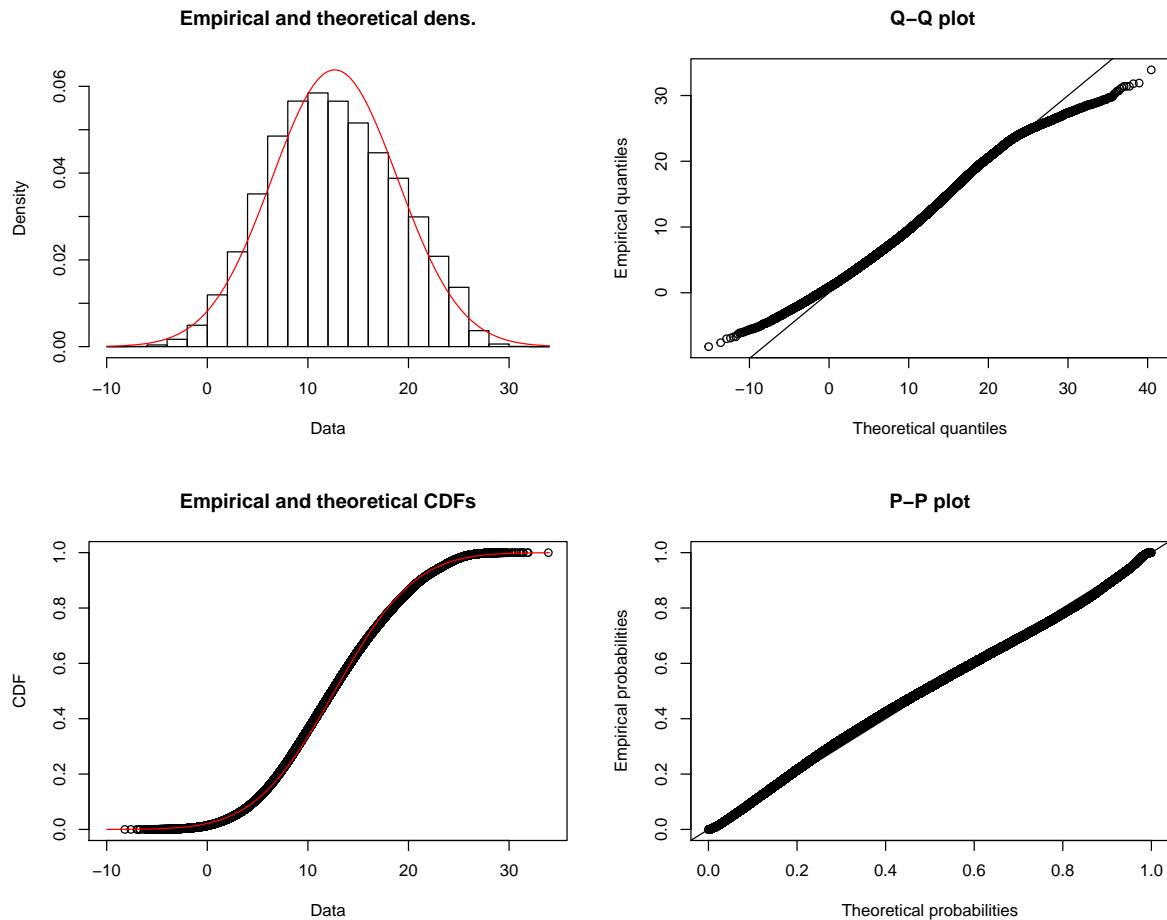
3.1.1 MinTemp

MinTemp assume valori nel seguente modo:

- min: -8.2
- max: 33.9
- mediana: 12.4

- media: 12.66
- deviazione standard: 6.25

Sembra adeguato assumere che una distribuzione valida sia la gaussiana.



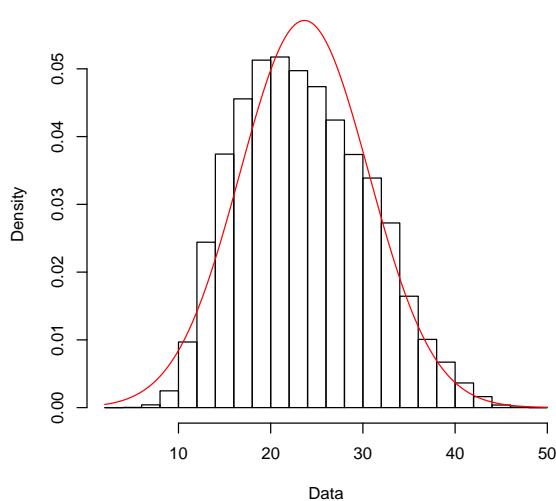
3.1.2 MaxTemp

MaxTemp assume valori nel seguente modo:

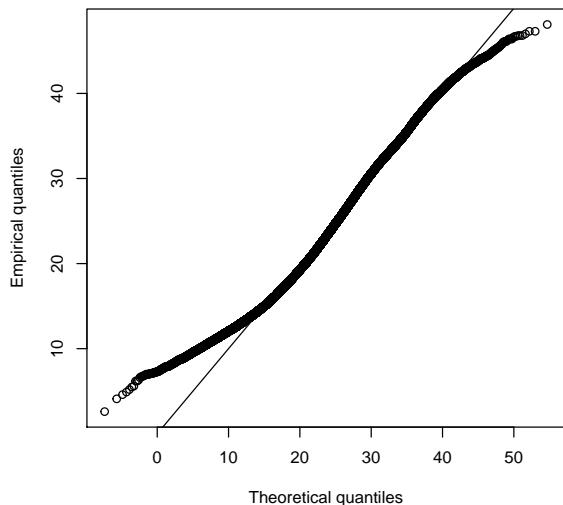
- min: 2.6
- max: 48.1
- mediana: 23.1
- media: 23.66
- deviazione standard: 6.98

Sembra adeguato assumere che una distribuzione valida sia la gaussiana.

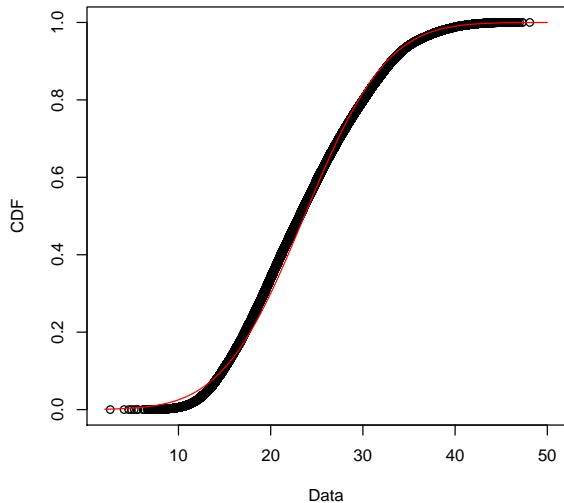
Empirical and theoretical dens.



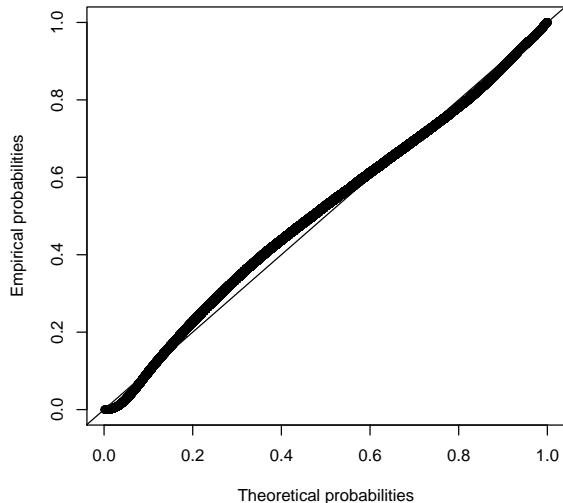
Q–Q plot



Empirical and theoretical CDFs



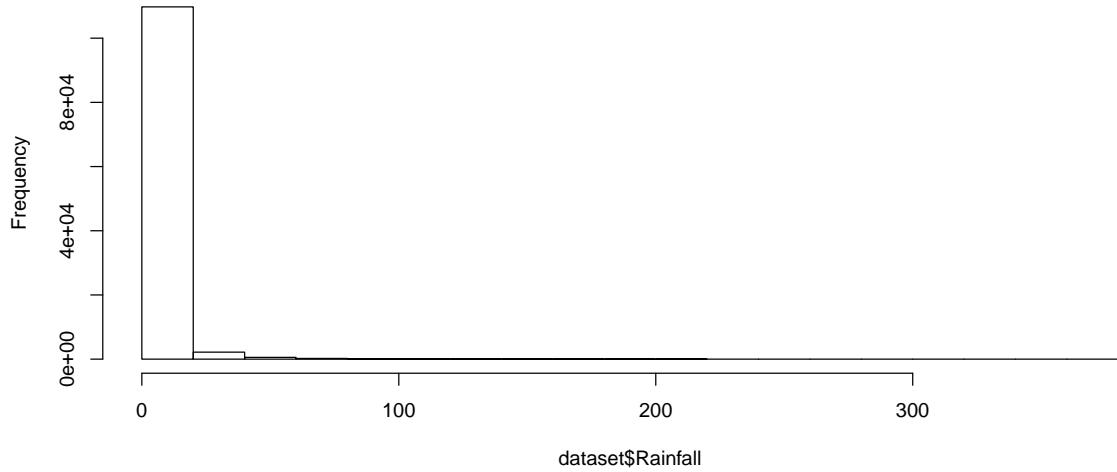
P–P plot



3.1.3 Rainfall

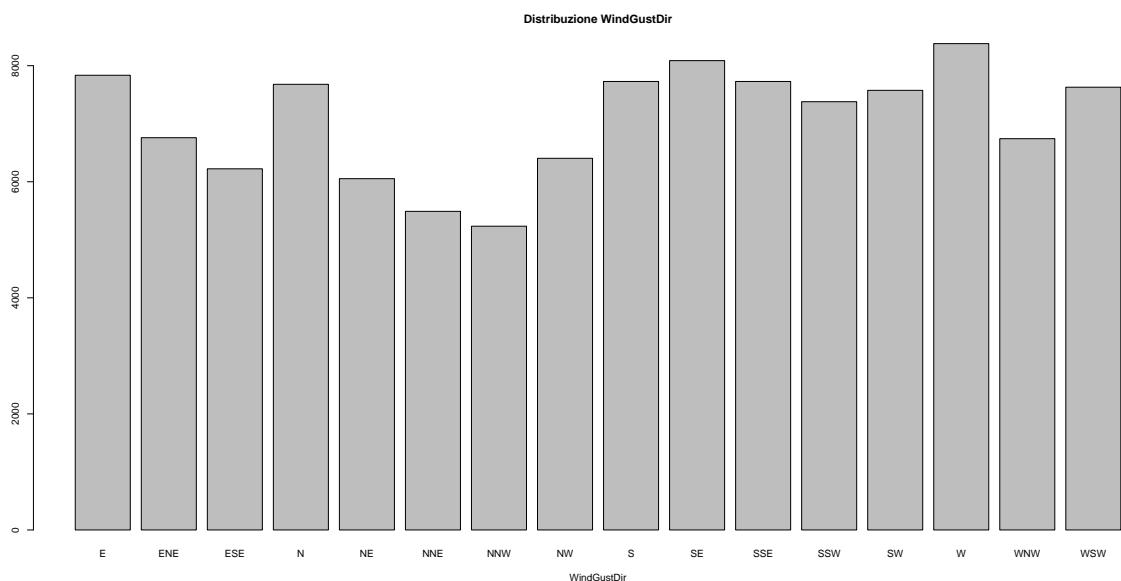
Considerando l'attributo Rainfall, si nota che i valori sono quasi tutti a zero (64.5%).

Histogram of dataset\$Rainfall



3.1.4 WindGustDir

WindGustDir assume valori di direzione del vento e per tale ragione si è scelto di sviluppare un grafico relativo alle occorrenze di ogni direzione assunta.



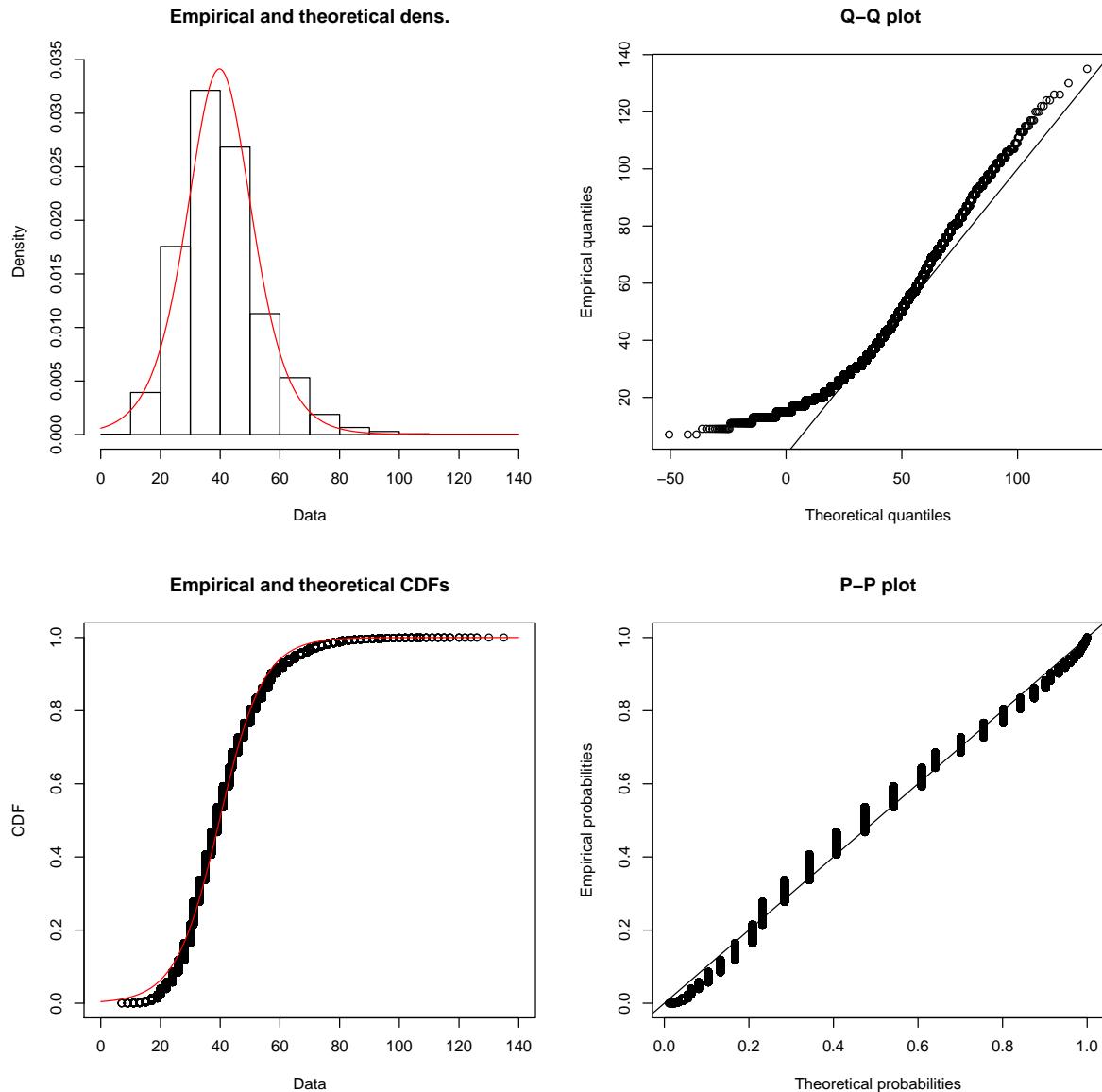
3.1.5 WindGustSpeed

WindGustSpeed assume valori nel seguente modo:

- min: 7
- max: 135
- mediana: 39
- media: 40.79

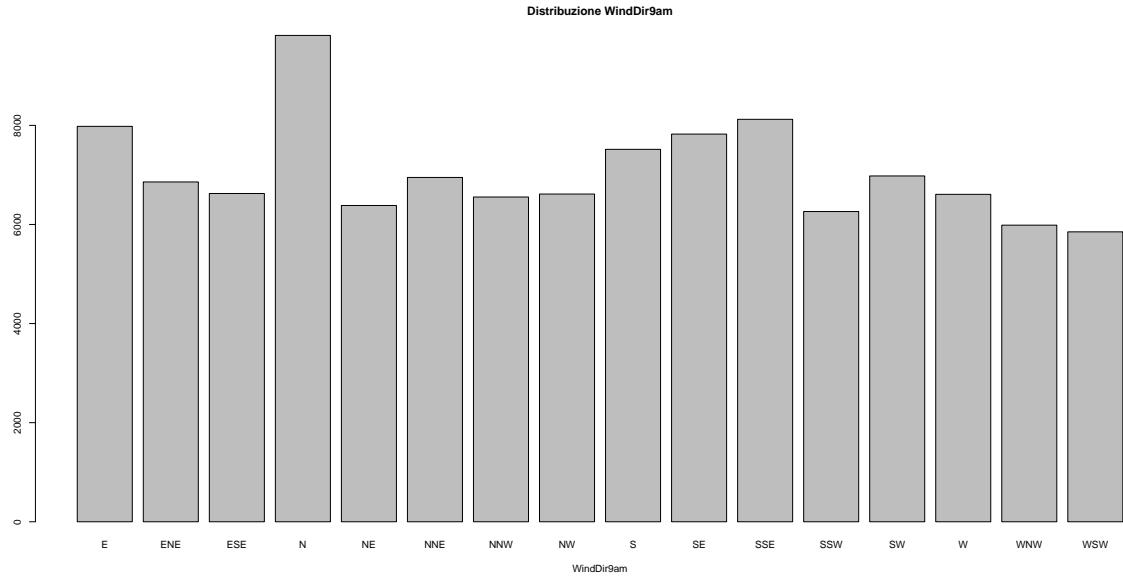
- deviazione standard: 13.32

Sembra adeguato assumere che una distribuzione valida sia la logistica.



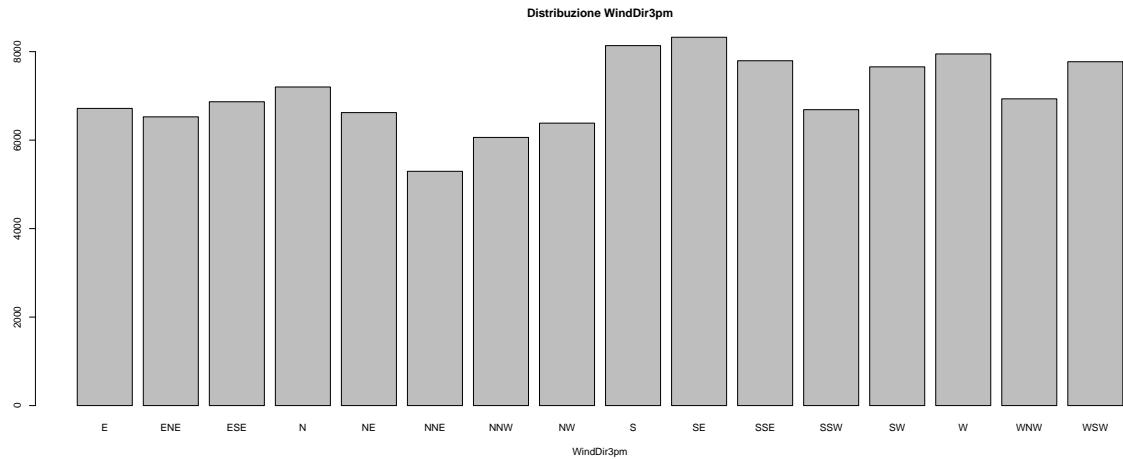
3.1.6 WindDir9am

WindDir9am assume valori di direzione del vento e per tale ragione si è scelto di sviluppare un grafico relativo alle occorrenze di ogni direzione assunta.



3.1.7 WindDir3pm

WindDir3pm assume valori di direzione del vento e per tale ragione si è scelto di sviluppare un grafico relativo alle occorrenze di ogni direzione assunta.

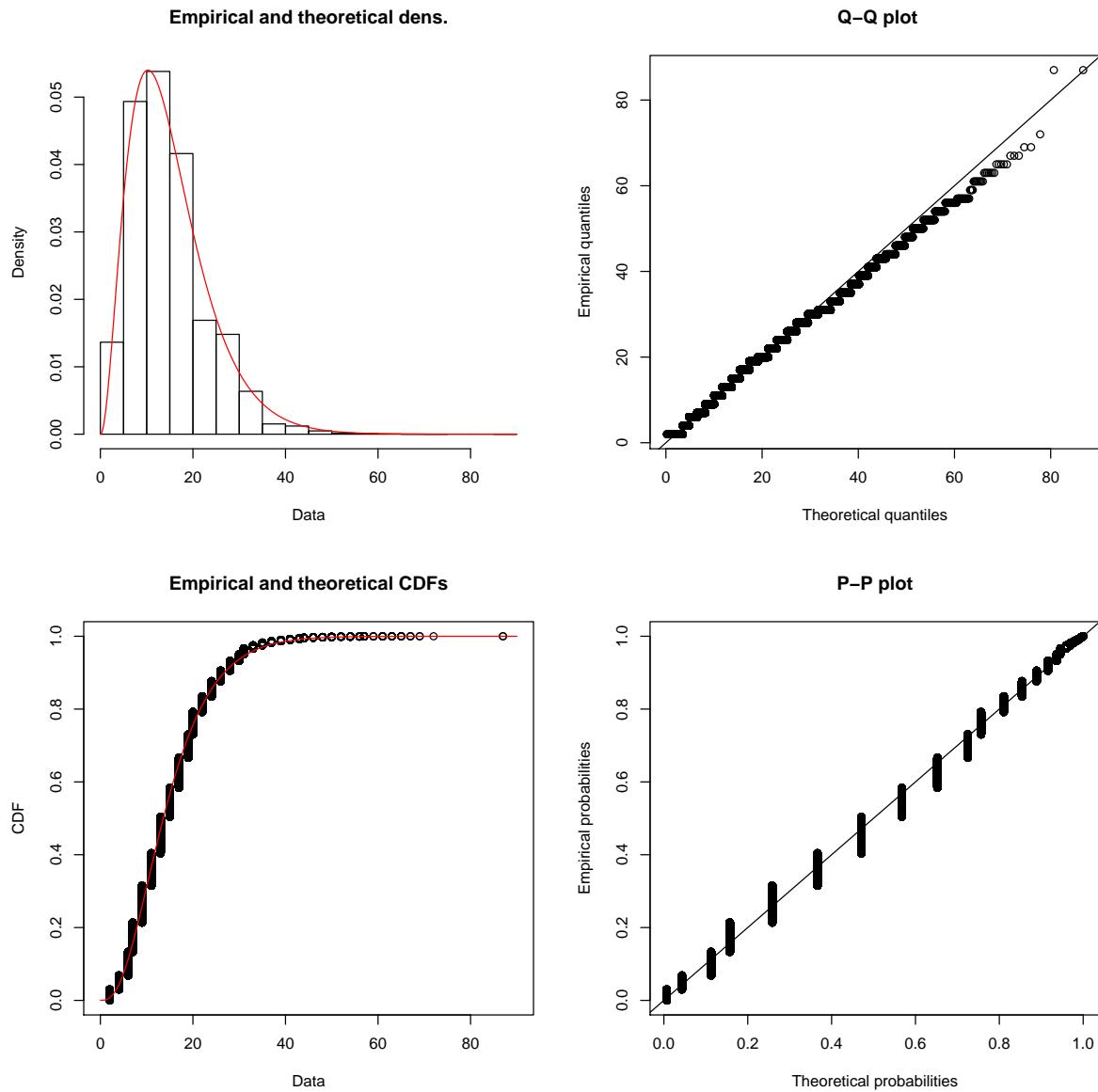


3.1.8 WindSpeed9am

WindSpeed9am assume valori nel seguente modo:

- min: 2
- max: 87
- mediana: 13
- media: 15.18
- deviazione standard: 8.34

Sembra adeguato assumere che una distribuzione adeguata sia la gamma.

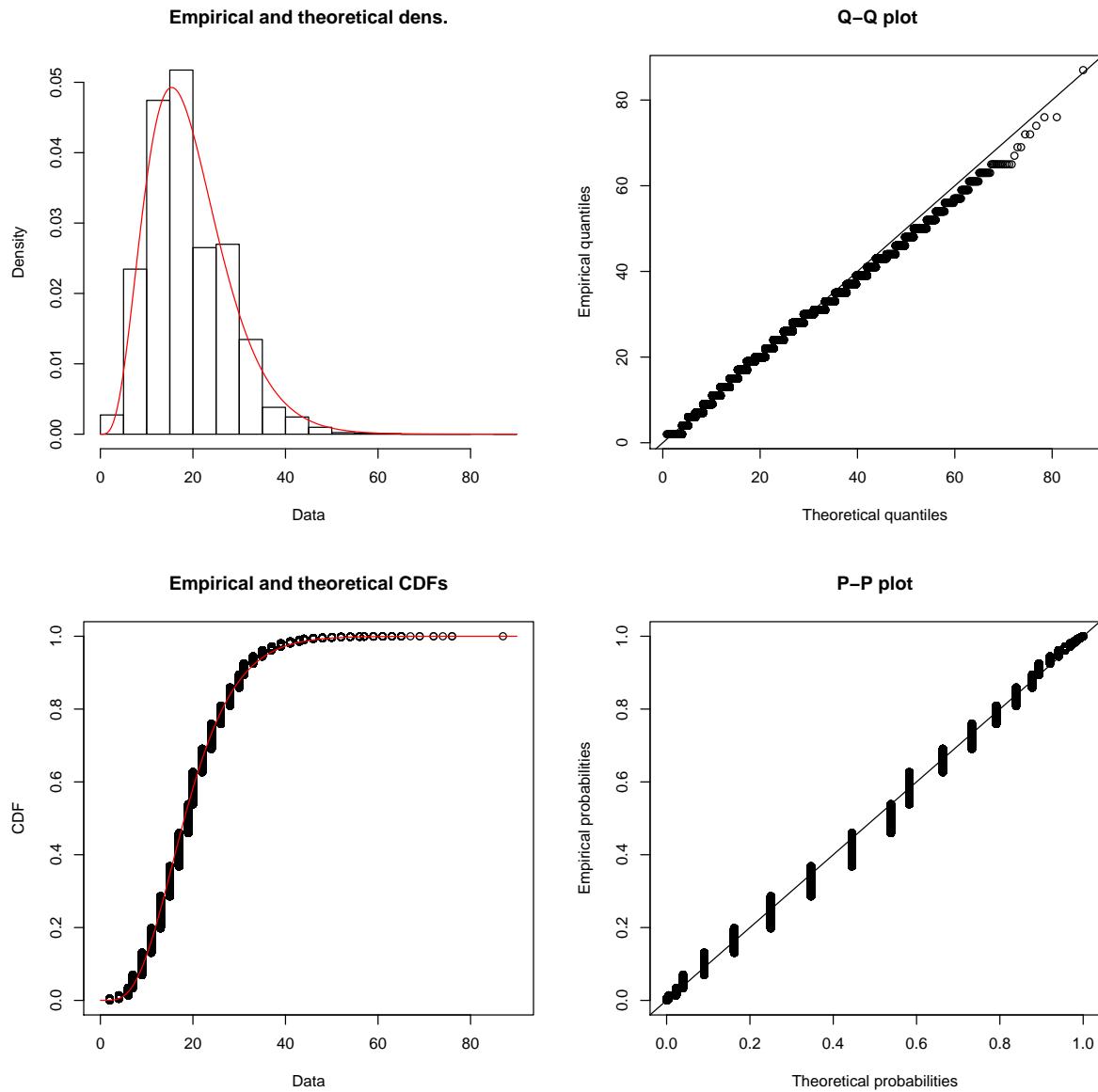


3.1.9 WindSpeed3pm

WindSpeed3pm assume valori nel seguente modo:

- min: 2
- max: 87
- mediana: 19
- media: 19.5
- deviazione standard: 8.58

Sembra adeguato assumere che una distribuzione adeguata sia la gamma.

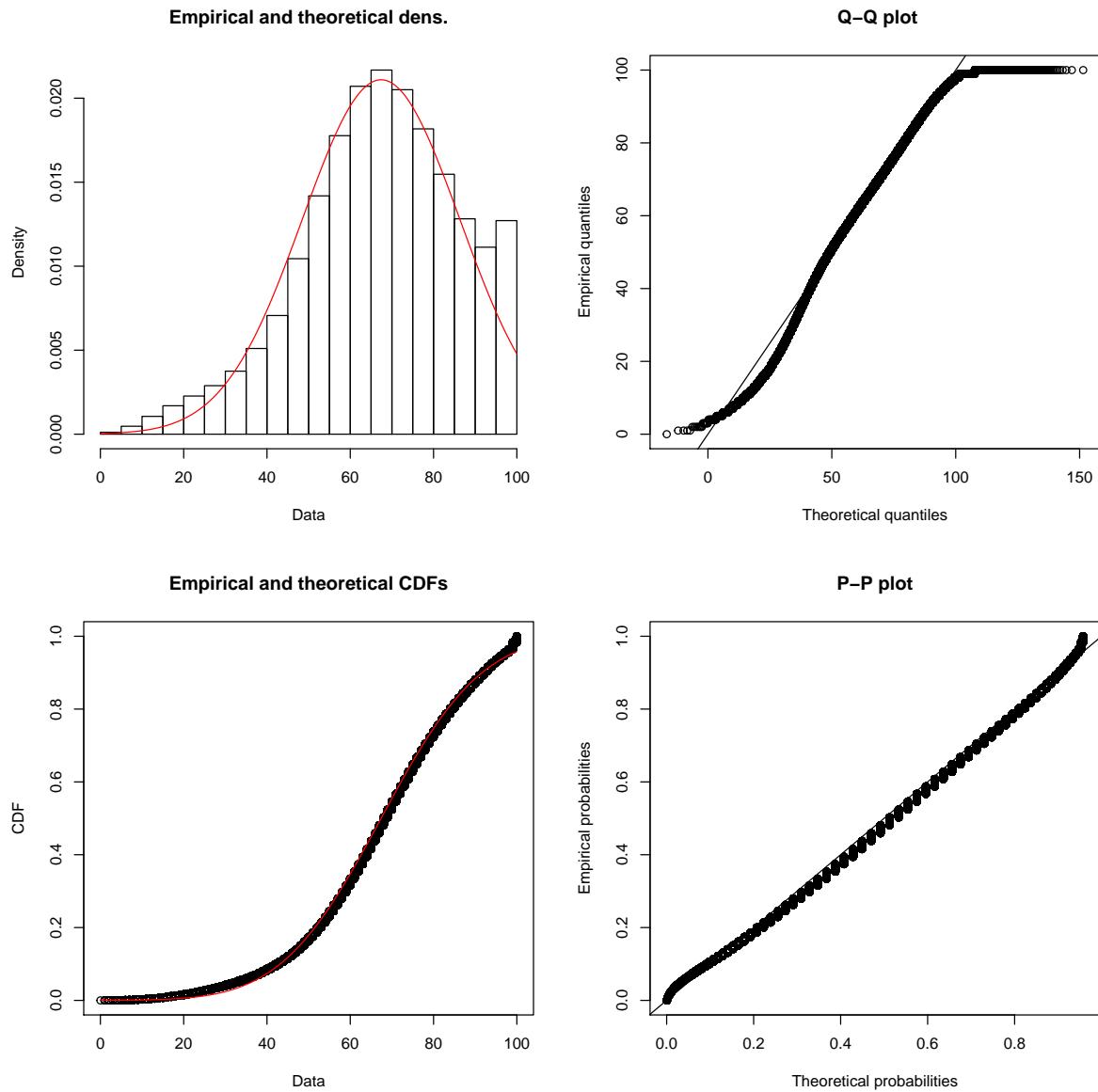


3.1.10 Humidity9am

Humidity9am assume valori nel seguente modo:

- min: 0
- max: 100
- mediana: 68
- media: 67.4
- deviazione standard: 18.91

Sembra adeguato assumere che una distribuzione valida sia la normale.



3.1.11 Humidity3pm

Humidity3pm assume valori nel seguente modo:

- min: 0
- max: 100
- mediana: 51
- media: 50.67
- deviazione standard: 20.77

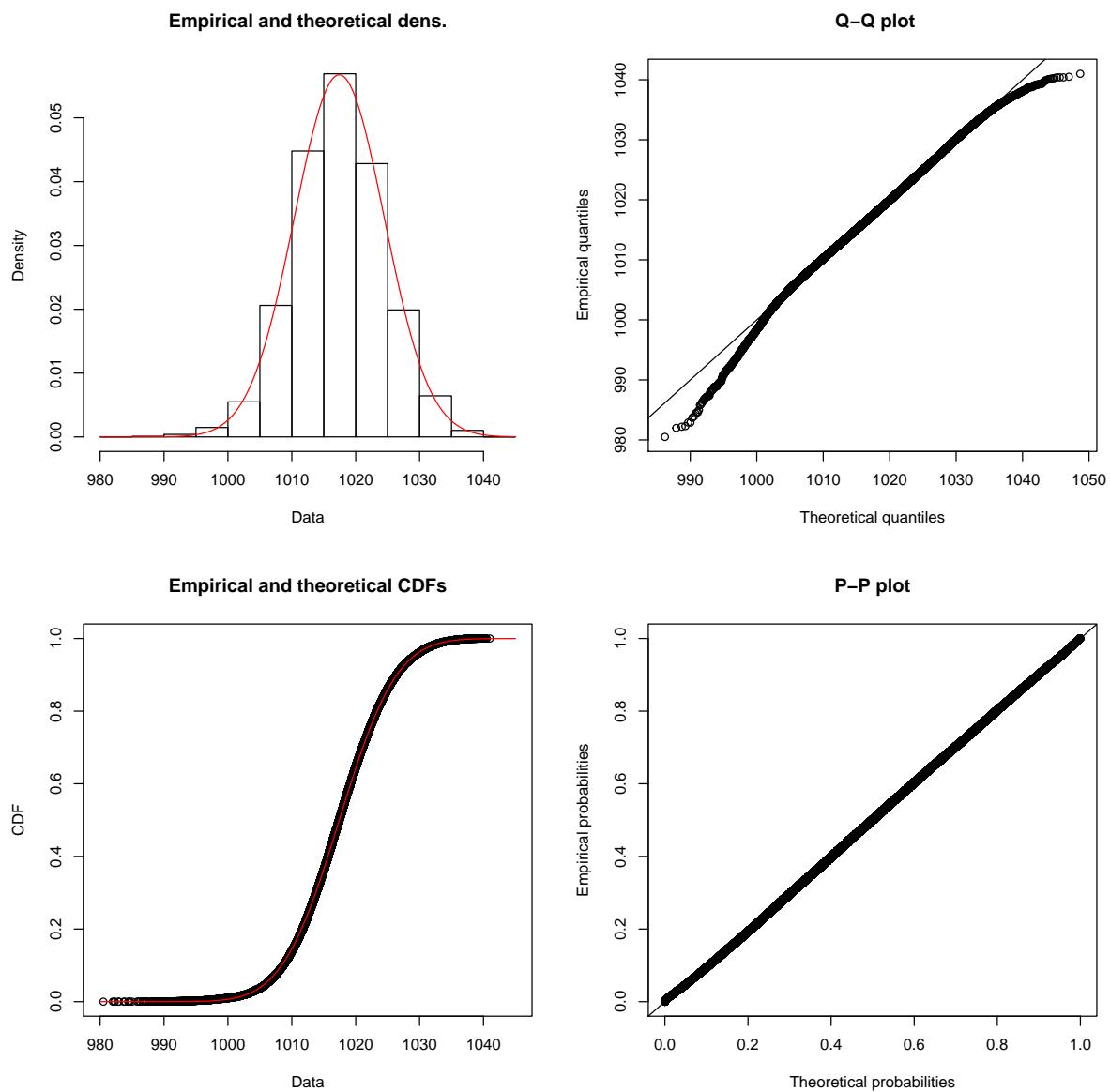
Sembra adeguato assumere che una distribuzione valida sia la normale.

3.1.12 Pressure9am

Pressure9am assume valori nel seguente modo:

- min: 980.5
- max: 1041
- mediana: 1017.4
- media: 1017.43
- deviazione standard: 7.03

Sembra adeguato assumere che una distribuzione valida sia la normale.

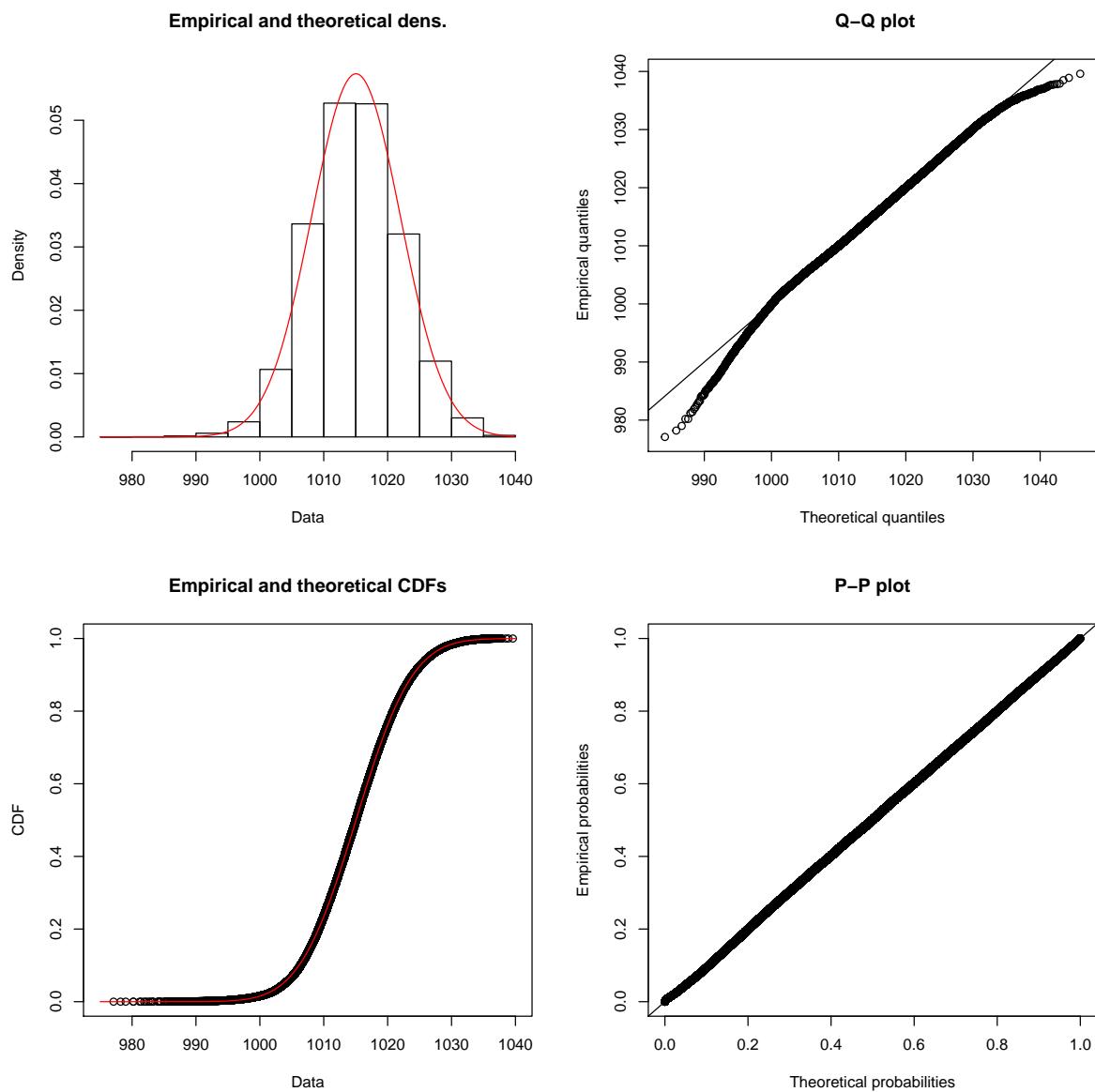


3.1.13 Pressure3pm

Pressure3pm assume valori nel seguente modo:

- min: 977.1
- max: 1039.6
- mediana: 1015
- media: 1015.05
- deviazione standard: 6.96

Sembra adeguato assumere che una distribuzione valida sia la normale.

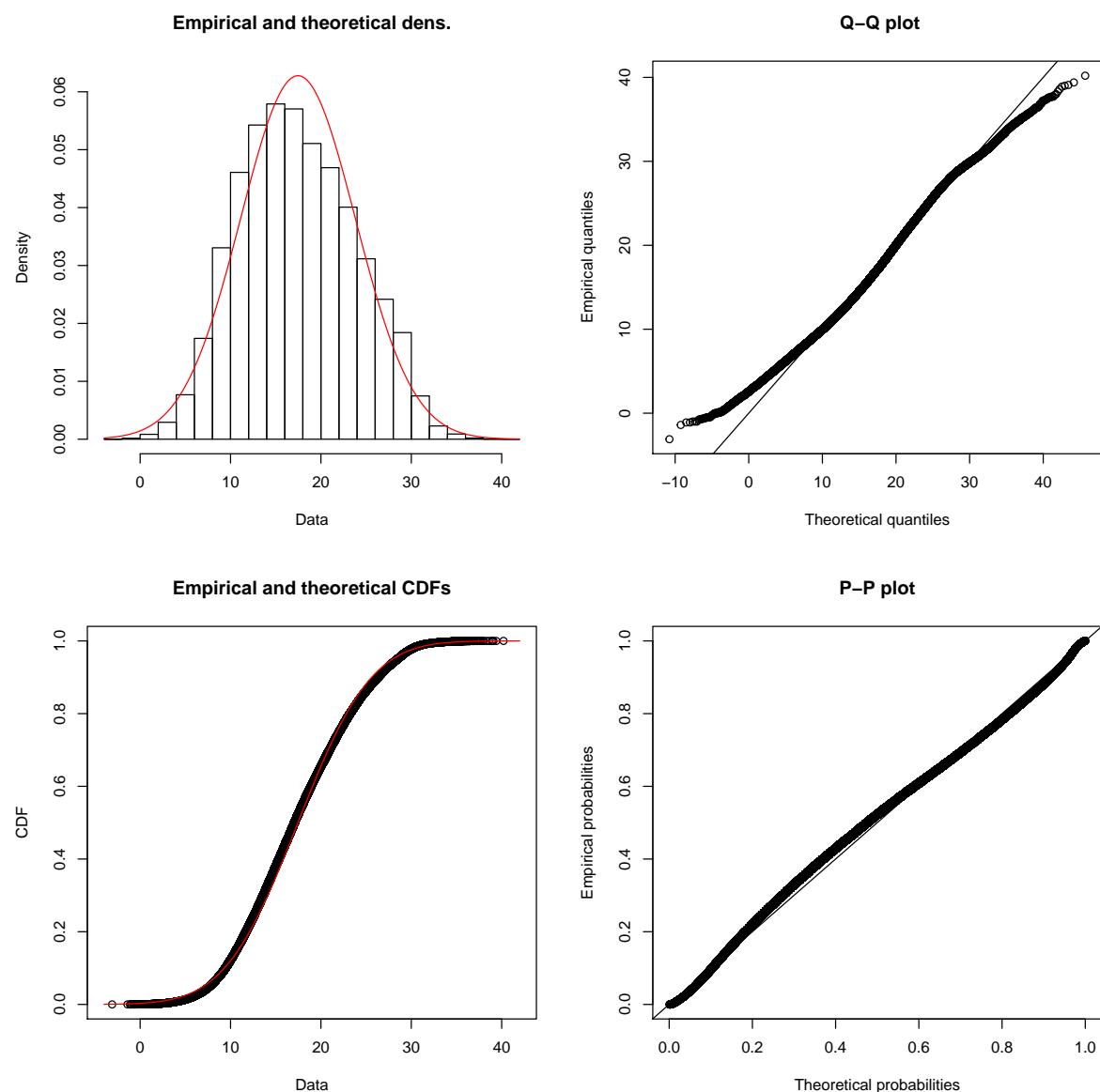


3.1.14 Temp9am

Temp9am assume valori nel seguente modo:

- min: -3.1
- max: 40.2
- mediana: 17.1
- media: 17.46
- deviazione standard: 6.36

Sembra adeguato assumere che una distribuzione valida sia la normale.

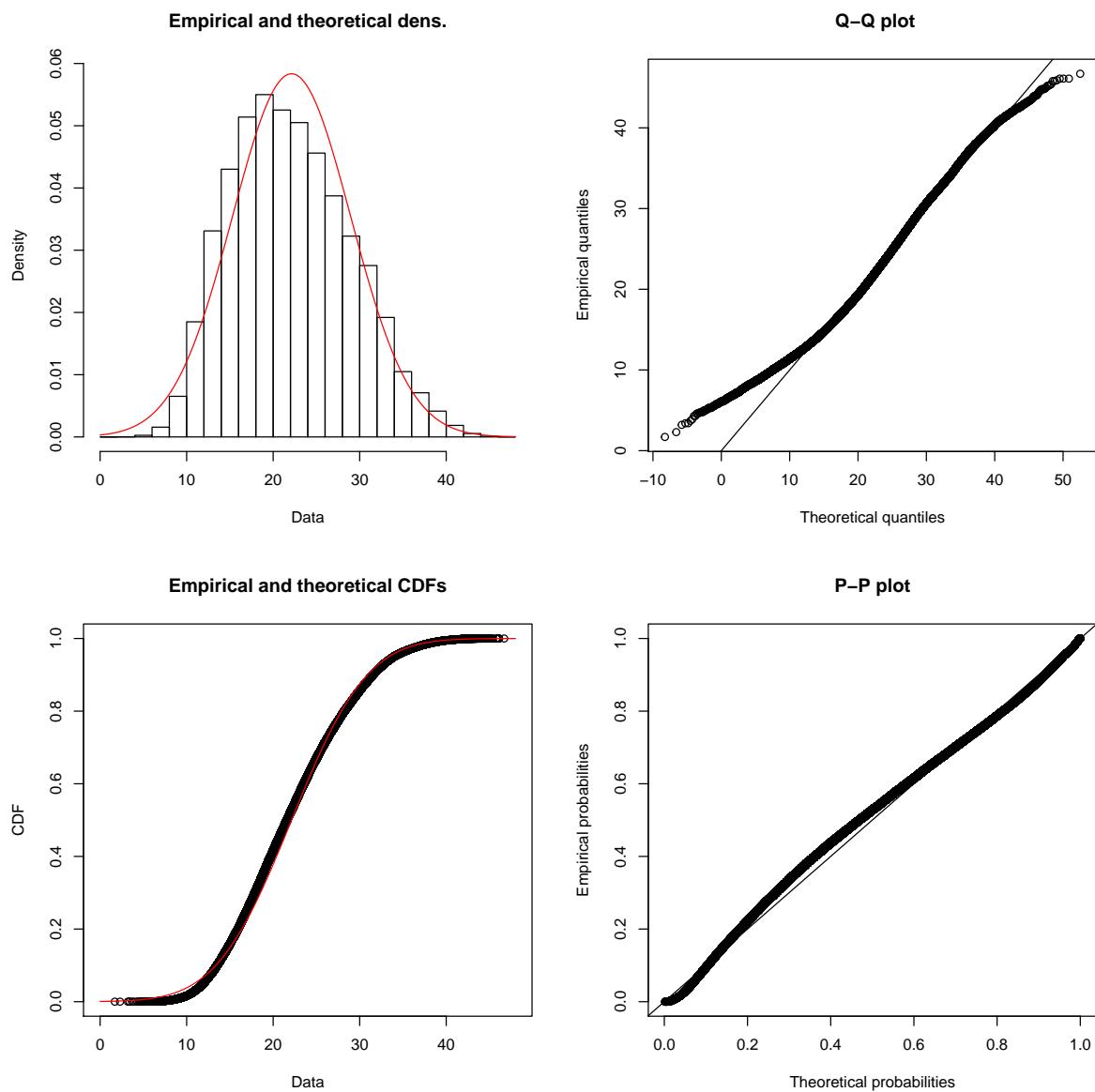


3.1.15 Temp3pm

Temp3pm assume valori nel seguente modo:

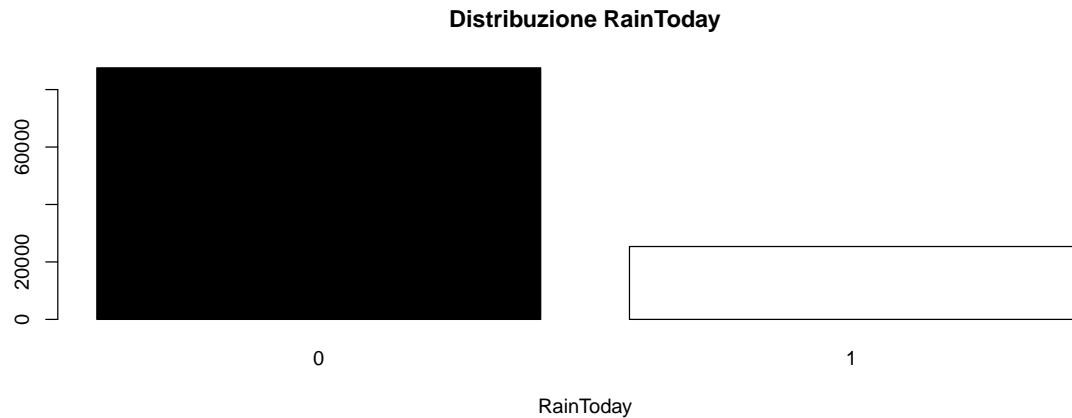
- min: 1.7
- max: 46.7
- mediana: 21.6
- media: 22.13
- deviazione standard: 6.84

Sembra adeguato assumere che una distribuzione valida sia la normale.



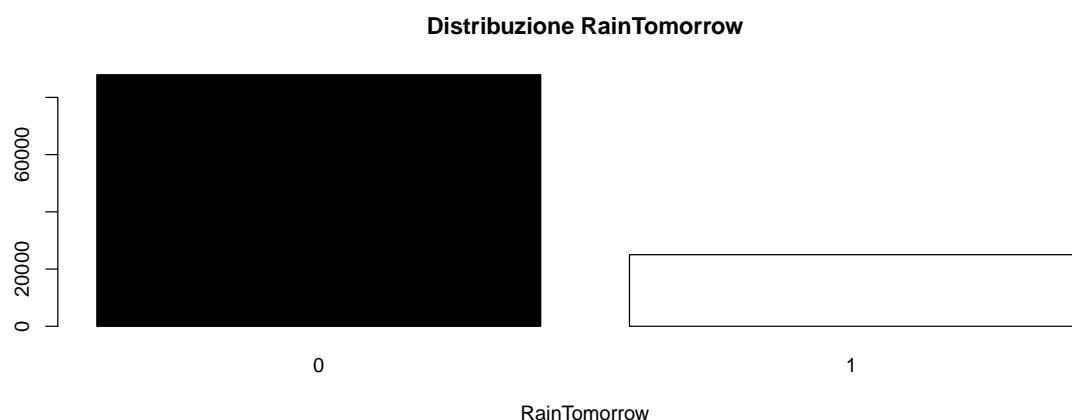
3.1.16 RainToday

Considerando l'attributo RainToday, si nota che i valori sono sbilanciati: 0 (77.5%) E 1 (22.5%)



3.1.17 RainTomorrow

Considerando l'attributo RainTomorrow, si nota che i valori sono sbilanciati: 0 (77.8%) E 1 (22.2%)

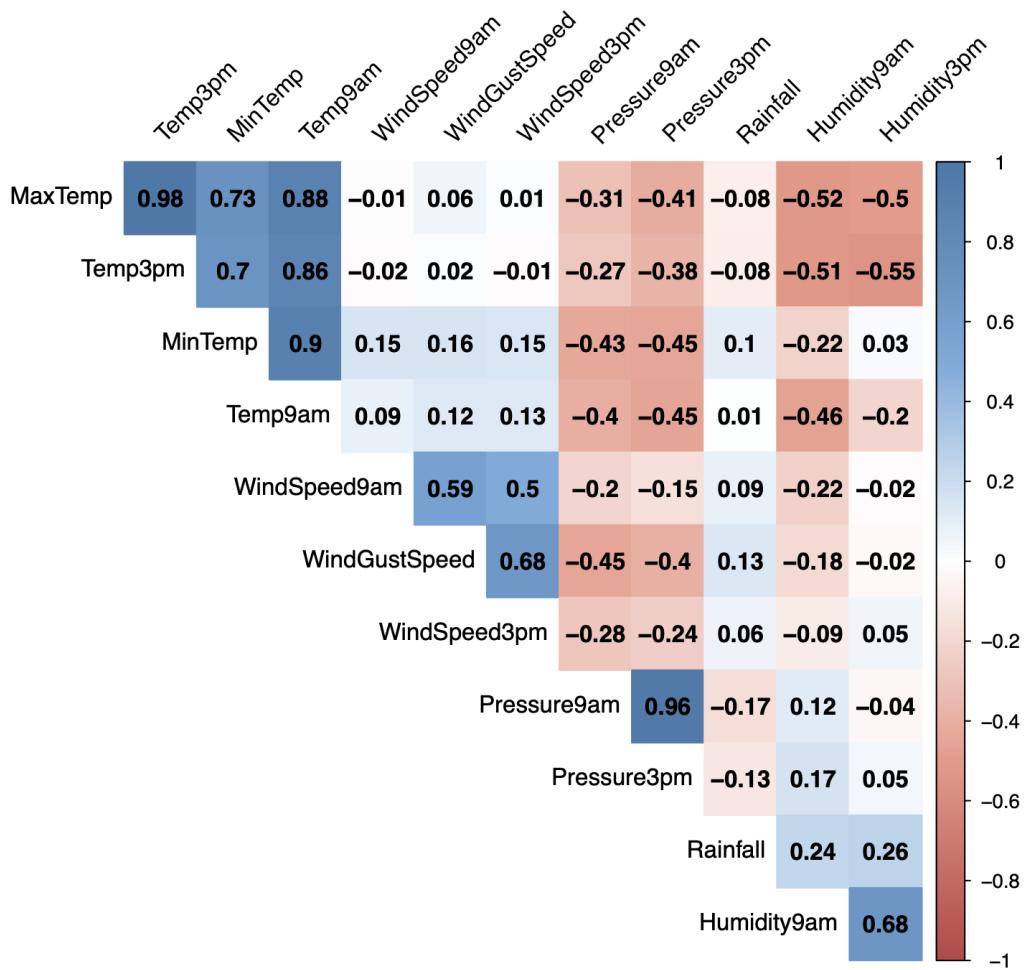


3.2 Correlazioni

Durante l'analisi delle correlazioni è emerso che molti degli attributi presenti nel dataset presentano alta correlazione tra loro e ciò ha portato ad eseguire approfondite analisi del fenomeno, così da poter gestire al meglio i dati da utilizzare durante il training ed il testing dei modelli che verranno sviluppati. In particolare, gli attributi anomali emersi per l'alta correlazione sono i seguenti:

- MaxTemp
- MinTemp
- Temp9am

- Temp3pm
- WindSpeed9am
- WindSpeed3pm
- WindGustSpeed
- Pressure9am
- Pressure3pm
- Humidity9am
- Humidity3pm



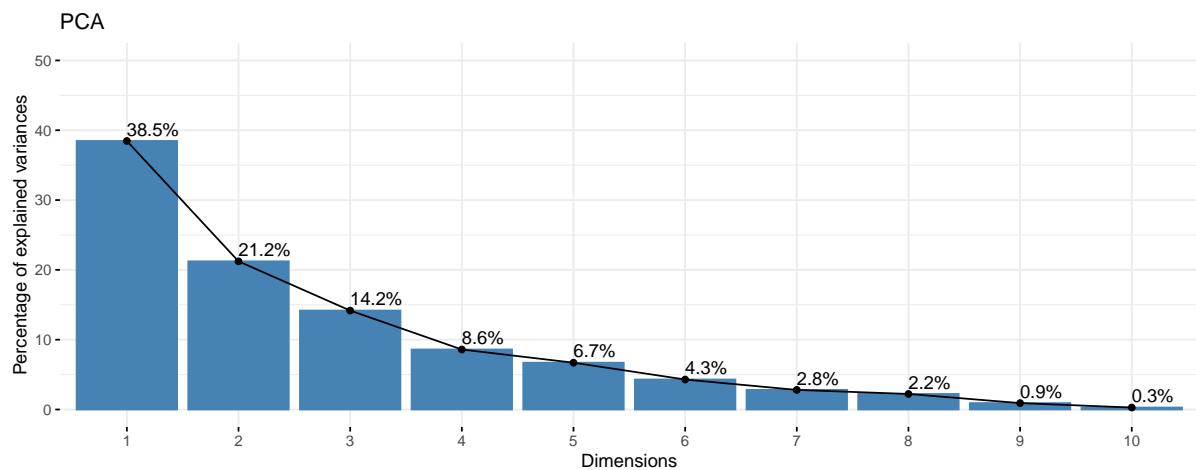
Successivamente è stata svolta l'operazione di "feature selection" che ha permesso di diminuire gli attributi considerati durante la creazione dei modelli: tramite la funzione findCorrelation sono stati selezionati gli attributi da rimuovere in base alla loro correlazione (se due attributi hanno correlazione molto alta portano informazioni ridondanti e ha senso rimuoverne uno dei due). Si è deciso di considerare le correlazioni maggiori di 0.67. Gli attributi scelti per l'eliminazione sono i seguenti:

- MaxTemp

- Temp9am
- Temp3pm
- WindGustSpeed
- Pressure3pm
- Humidity9am

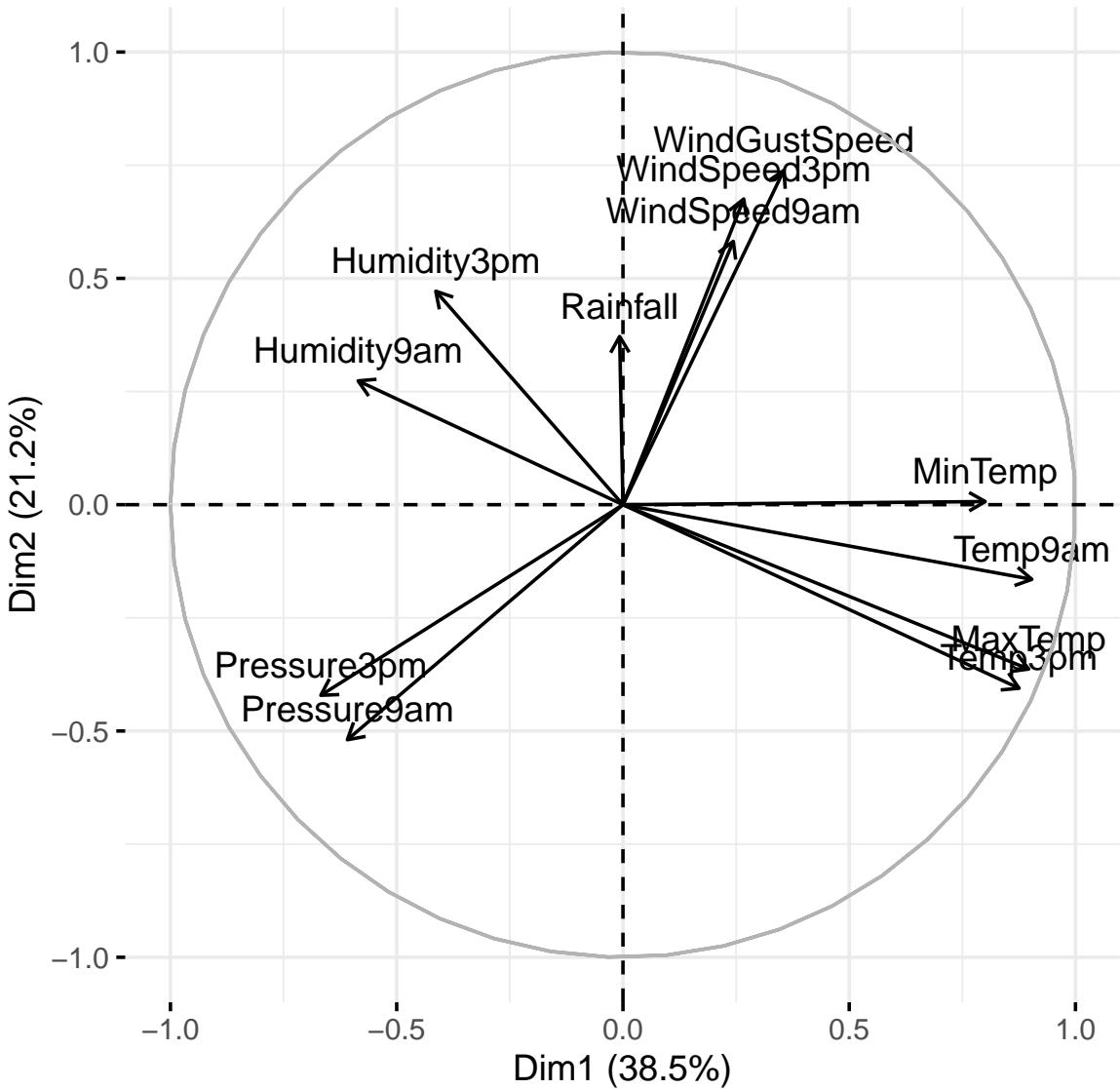
3.3 PCA

L'analisi tramite PCA ha permesso di verificare le componenti principali (misurate in termini di varianza); in particolare, prendendo gli autovalori maggiori di 1 (ovvero quelli associati alle prime 4 dimensioni) è possibile esprimere una varianza pari all'82.45% di quella totale.



Considerando il grafico, è facile avere una conferma di ciò che è emerso durante l'analisi della matrice di correlazione. L'alta correlazione tra attributi indica la presenza di ridondanza nell'informazione e porta alla necessità di effettuare una riduzione di dimensionalità.

Variables – PCA



4 Modelli

4.1 Holdout

In questa sezione verranno presentati e commentati diversi modelli allenati nella modalità holdout (considerando perciò una solo suddivisione del dataset in trainset e testset) e di cui sono state calcolate successivamente le relative performance per avere una traccia generale su quali esperimenti portare nella modalità 10-fold cross validation.

Una prima parte riguarda delle prove preliminari su modelli definiti dummy in quanto utili solo ad una introduzione di modelli basati su assunzioni e per tale motivo non allenati con tecniche di machine learning ma che possono comunque fornire informazioni utili nella prosecuzione del progetto.

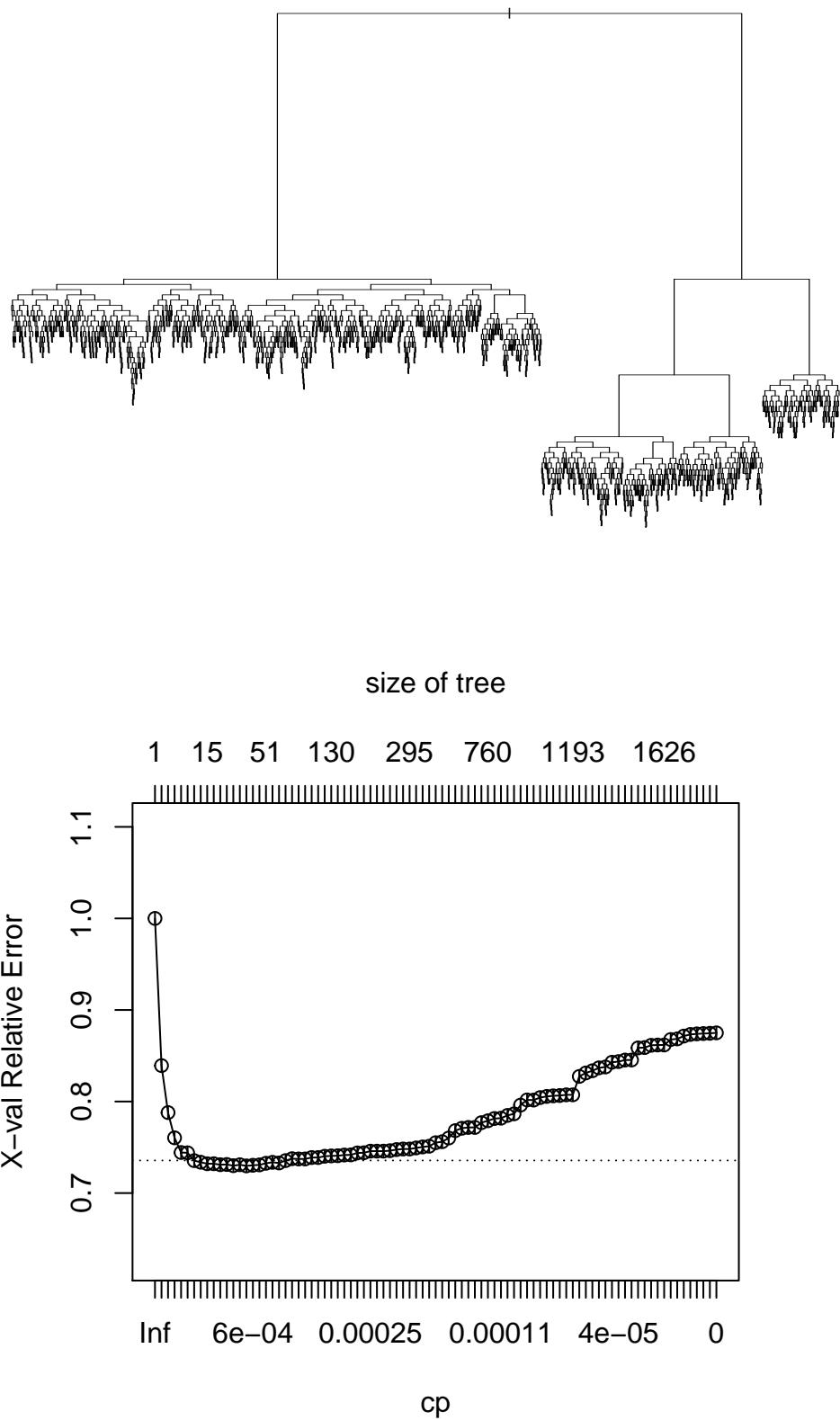
1. **Dummy model 1:** In prima istanza è stata considerata una predizione nella quale l'unico output è 0 (quindi il fatto che il giorno successivo non ci sarà pioggia); tale decisione è stata presa dato l'esito dell'analisi effettuata sul dataset nella quale si

nota che la variabile target è sbilanciata proprio sul valore "no" o 0. Sono state poi calcolate le performance su tale approccio e queste hanno portato un risultato che in termini generali è tutto sommato buono (accuracy del 77.6%), frutto però dello sbilanciamento di tale variabile. Lo scopo di tale primo approccio è stato quello di fornire un benchmark di performance che i modelli allenati dovranno superare in quanto questo primo approccio non comporta alcun costo in termini computazionali ed ottiene comunque una buona base da cui iniziare; non avrebbe perciò senso utilizzare del tempo aggiuntivo se non al fine di migliorarne le performance.

2. **Dummy model 2:** Una seconda analisi è stata quella di valutare un modello sulla base di tale assunzione: "se in un giorno ci sarà pioggia allora anche in quello successivo è probabile che pioverà". A questo punto è stato assegnato un valore "yes" o 1 solo a quelle osservazioni in cui la variabile "RainToday" è uguale a "yes" e un valore di "no" o 0 alle rimanenti. Sono state poi calcolate le performance di tale modello con le quali è possibile scoprire che tale assunzione non è fondata in quanto le performance ottenute sono minori del caso precedente (accuracy di 76.4% contro 77.6% del precedente approccio).

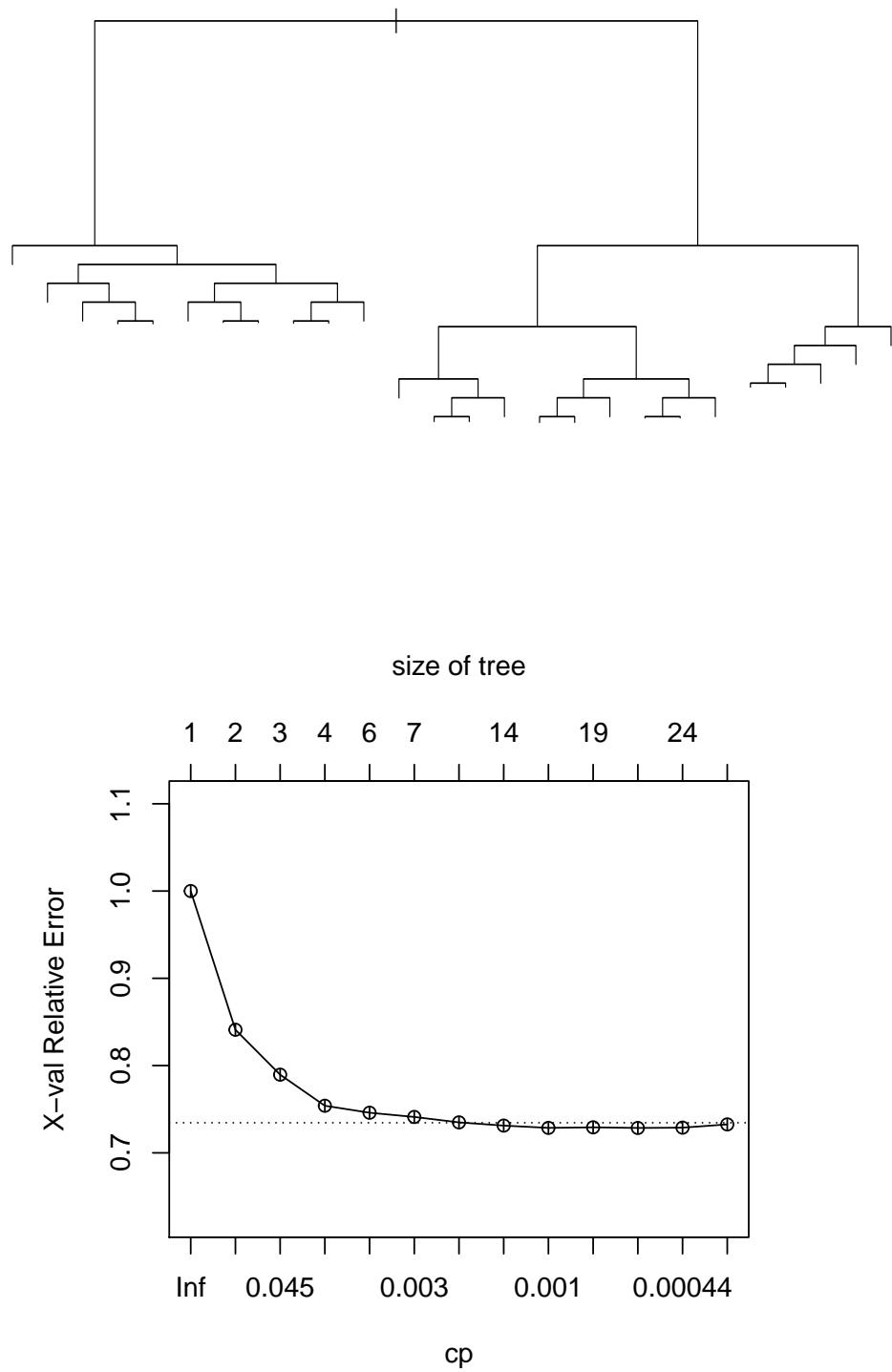
Verranno ora presi in considerazione modelli di machine learning veri e propri che verranno allenati sempre con la modalità holdout e di cui verranno valutate le performance. È importante sottolineare che avendo una classe target leggermente sbilanciata è importante valutare oltre che accuracy anche le altre metriche, quali precision, recall e F1 meausure.

1. **Decision tree - Random forest:** Come primo modello è stato scelto decision tree, utilizzando la libreria rpart. Tale modello viene allenato andando a creare un albero di decisione via via sempre più profondo prendendo per ogni nodo la variabile più "esplicativa" in termini di predizione del target. Il primo approccio è stato quello di allenare un albero senza limiti di complessità e profondità per valutarne i risultati. Come visibile dal plot dell'albero e dai parametri di complessità questo approccio non porta a risultati ottimi, in quanto dopo una certa profondità le performance iniziano a diminuire invece che aumentare e al contrario la complessità computazionale aumenta sempre di più.

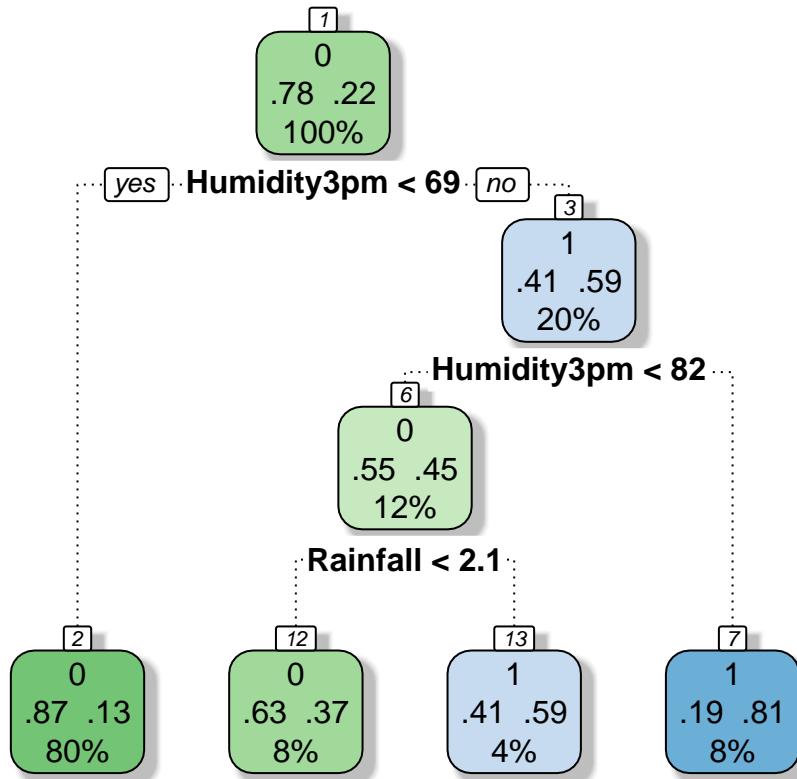


Questo primo approccio di decision tree ha una accuracy di 81% circa che risulta già migliore dei primi modelli dummy, ma a cui si possono apportare ulteriori migliorie. La strategia attuata è stata quella di ridurre la profondità massima che l'albero può

raggiungere basandoci sul grafico delle complessità e il plot appena prodotti, in particolare è stato scelto di limitare la crescita dell'albero ad una profondità massima di sei livelli. È stato così allenato un nuovo albero di decisione che rispettasse tali caratteristiche e ne sono state confrontate le performance con quello precedente.



Già dal grafico della complessità e dal plot è visibile una netta semplificazione dell'albero di decisione, sono state poi valutate le performance su tale modello e queste hanno mostrato che tale modello è preferibile rispetto a quello precedente poiché oltre ad essere migliorato in termini di complessità computazionale (l'albero è stato molto ridotto) è migliore in termini di accuracy (84% circa contro l'81% del precedente). Un'ultima analisi è stata valutare le performance dello stesso albero ma potato, quindi ridotto ulteriormente di complessità. È stato deciso di effettuare la potatura a un livello di complessità pari a 0.017 ($cp = 0.017$).



Come si nota dal plot di tale modello è stato possibile effettuare un render molto più comprensibile dell'albero (attraverso la libreria "fancyRpartPlot"). Sono state successivamente valutate le sue performance con le quali si è potuto stabilire che tale modello è il migliore in quanto abbassa la complessità generale con una riduzione di accuracy accettabile (da 84% circa a 83.5% di questo modello semplificato).

Tempistiche modello:

- Tempo creazione modello: 2 secondi
- Tempo calcolo prediction: 0.05 secondi

Metriche performance Modello:

- Accuracy: 0.8356
- Precision: 0.7511

- Recall: 0.3973
- F1 measure: 0.5197

In ultima istanza è stato allenato un modello random forest che lavora costruendo molti alberi di decisione per poi considerarne il migliore. Questo modello rispetto ai precedenti ha richiesto più tempo per il suo allenamento (in quanto allena diversi alberi di decisione, in particolare ntree = 500). Le sue performance sono però le più alte di quelle ottenute da tutti gli altri modelli in quanto raggiunge una accuracy di quasi 85%.

Tempistiche modello:

- Tempo creazione modello: 1.5 minuti
- Tempo calcolo prediction: 3.5 secondi

Metriche performance Modello:

- Accuracy: 0.8481
- Precision: 0.7140
- Recall: 0.5353
- F1 measure: 0.6119

2. **Naive Bayes:** Il secondo modello valutato è stato Naive Bayes. Questo modello basa la sua logica sul teorema di Bayes, fornisce perciò una classificazione in base alla probabilità che il target assuma un valore piuttosto che un altro, in particolare però è stato adottato un approccio naive di questo teorema in quanto vi è l'assunzione abbastanza forte che ogni variabile è indipendente da tutte le altre (indipendenza tra le features). Il modello allenato con questo approccio è molto veloce nel costruire il modello (il più veloce tra i modelli valutati) ma impiega più tempo degli altri (restando comunque in tempi molto contenuti, meno di 10 secondi) nella fase di predizione dei risultati. Tale modello ottiene buoni risultati (accuracy 82% circa e f1 score leggermente minore di decision tree)

Tempistiche modello:

- Tempo creazione modello: 0.1 secondi
- Tempo calcolo prediction: 8 secondi

Metriche performance Modello:

- Accuracy: 0.8198
- Precision: 0.6505
- Recall: 0.4213
- F1 measure: 0.5114

3. **Neural Network:** Il terzo modello scelto è stata una Neural network tramite package "nnet". Neural network è sicuramente il modello più complesso da allenare in quanto sono presenti molti parametri da ottimizzare. Le performance di tale approccio sono in linea con gli altri modelli e in aggiunta questo si crea in tempo decisamente minore ad altri modelli (per esempio a SVM)

Tempistiche modello:

- Tempo creazione modello: 1 minuto
- Tempo calcolo prediction: 0.7 secondi

Metriche performance Modello:

- Accuracy: 0.835
- Precision: 0.6688
- Recall: 0.5205
- F1 measure: 0.5854

4. **SVM**: Un ultimo approccio utilizzato è stato quello delle Support Vector Machine, un modello che cerca attraverso delle funzioni kernel (Lineare, Radiale, Polinomiale o Sigmoide) di trovare l'iperpiano separatore migliore, che abbia perciò il margine maggiore tra le due classi. Questo è stato il modello più complesso in termini di complessità computazionale, impiegando circa 15 minuti per allenare il modello e 2 minuti per calcolare le predizioni del target. Sono state effettuate diverse prove utilizzando tutti i kernel disponibili e cercando di ottimizzare gli iperparametri, quali Costo e Gamma ma tali prove non hanno portato benefici ed hanno solamente complicato ulteriormente il modello. Nonostante i tempi maggiori, le performance sono in linea con gli altri modelli (sia in termini di accuracy pari a 84% circa sia in termini di precision, recall e f1 score).

Tempistiche modello:

- Tempo creazione modello: 17 minuti
- Tempo calcolo prediction: 2 minuti

Metriche performance Modello:

- Accuracy: 0.8413
- Precision: 0.7916
- Recall: 0.3948
- F1 measure: 0.5268

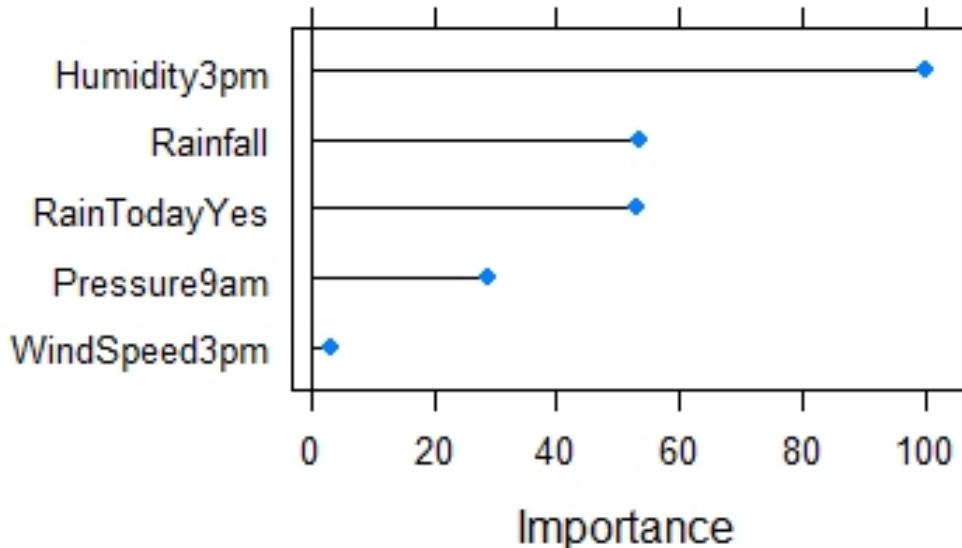
Il fine di tale analisi è stata capire quali modelli hanno i migliori risultati con il dataset utilizzato per capire con quali modelli ha senso effettuare esperimenti in 10-fold cross validation, per confermare o approfondire le conclusioni dell'approccio holdout e quali modelli non si ha convenienza ad analizzare. In particolare è stato scelto di approfondire i seguenti modelli: Decision tree, Random Forest, Naive Bayes e Neural Network. Il modello SVM invece non verrà preso in considerazione nell'approccio 10-fold cross validation in quanto troppo oneroso da allenare ed ottimizzare e avendo notato che non fornisce performance migliori degli altri modelli.

4.2 10-fold cross validation

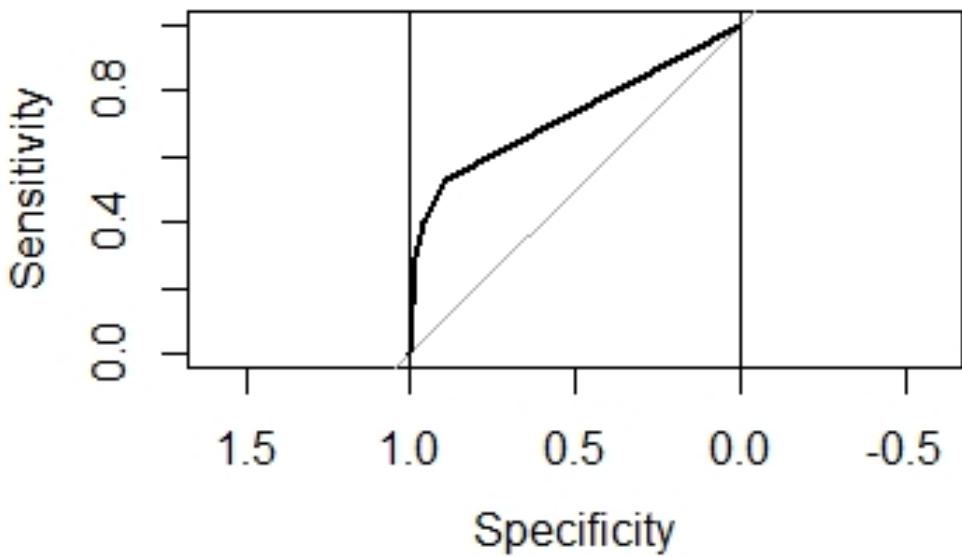
Con questo approccio gli elementi fondamentali sono l'impostazione di un controllo che assicuri che i modelli vengano eseguiti con una 10-fold cross validation, viene cioè diviso il dataset in 10 fold e calcola le performance come media delle performance di questi

sottoinsiemi, questo evita di ottenere dati che hanno performance sovrastimate o sottostimate a causa di divisioni "particolari" del trainset introducendo dei bias. È stata poi tracciata la curva ROC (Receiver Operating Characteristic) che illustra come performano i vari modelli di classificazione binaria utilizzando il true positive rate contro il false positive rate per diversi punti. Attraverso la ROC si può calcolare l'AUC (Area Under Curve) che costituisce un'ulteriore misura di performance dei modelli. Tutti i modelli seguenti sono stati allenati e valutati utilizzando il package "caret" che fornisce un numero molto alto di modelli da allenare ed ottimizzare.

- Decision Tree:** Come primo esperimento 10-fold cross validation è stato allenato un decision tree (method "rpart2"), il modello è stato ottimizzato per il parametro "maxdepth" cioè la profondità massima che l'albero può raggiungere. Su tale modello sono state calcolate le performance e si può notare come queste siano esattamente le stesse del modello decision tree in modalità holdout. Anche i tempi di creazione del modello e delle predizioni sono accettabili e questo conferma decision tree come uno dei modelli migliori tra quelli analizzati come compromesso tra complessità computazionale e prestazioni. Vengono ora presentati i grafici riguardanti le variabili più importanti per questo modello e il grafico raffigurante la curva ROC.



Questo modello pone la variabile "Humidity3pm" come la più importante per la classificazione, seguita da "Rainfall" e "Raintoday" con valore "Yes" (1). Si nota inoltre che utilizza un insieme ristretto di variabili per la creazione dell'albero.



Sono ora presentate le tempistiche e le performance di tale modello.

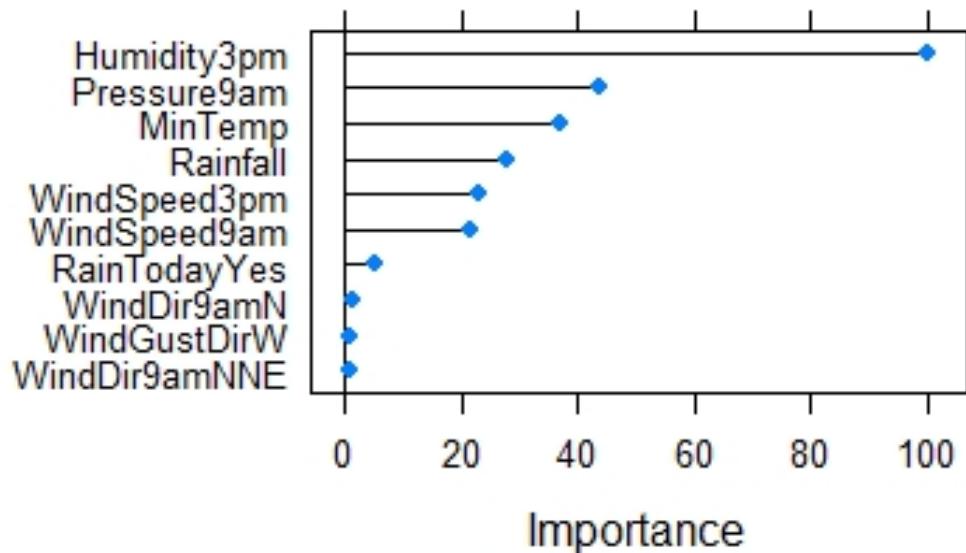
Tempistiche modello:

- Tempo creazione modello: 19 secondi
- Tempo calcolo prediction: 1 secondo

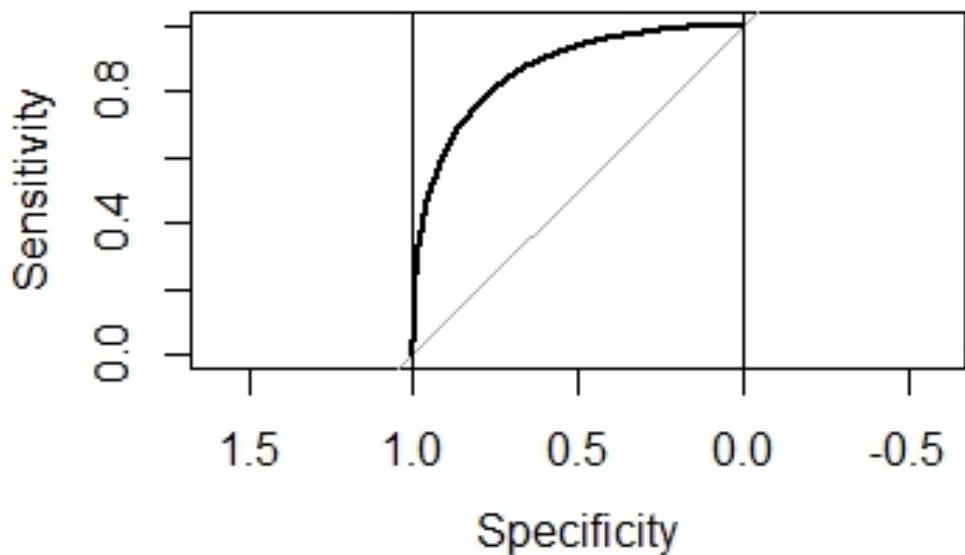
Metriche performance Modello:

- Accuracy: 0.8356
- Precision: 0.7511
- Recall: 0.3973
- F1 measure: 0.5197
- AUC: 0.7236

2. **Random Forest:** Il secondo modello allenato con approccio 10-fold cross validation è stato random forest (method "rf"). Questo modello è stato sicuramente il più complesso da allenare in quanto il package "caret" ricerca una serie di ottimizzazioni ai parametri di random forest, in particolare il parametro "mtry", esso infatti prova diverse configurazioni di tale parametro per poi allenare il dataset con quello migliore. Questo porta ad un'esplosione della complessità del modello portando la sua costruzione ad un tempo di 4 ore, una volta scoperto però il parametro utilizzato si può riallenare il modello utilizzando solo uno specifico valore del parametro riuscendo così a ridurre il tempo di costruzione. Il tempo impiegato permette però di ottenere i risultati migliori sia in termini di accuracy sia in termini di AUC, bisogna però ponderare la complessità molto maggiore di tale modello rispetto ad esempio a decision tree che risulta molto meno complesso e con performance solo di poco minori. Vengono ora presentati i grafici riguardanti le variabili più importanti per questo modello e il grafico raffigurante la curva ROC.



Questo modello pone la variabile "Humidity3pm" come la più importante per la classificazione seguita da "Pressure9am", "Mintemp" e "Rainfall".

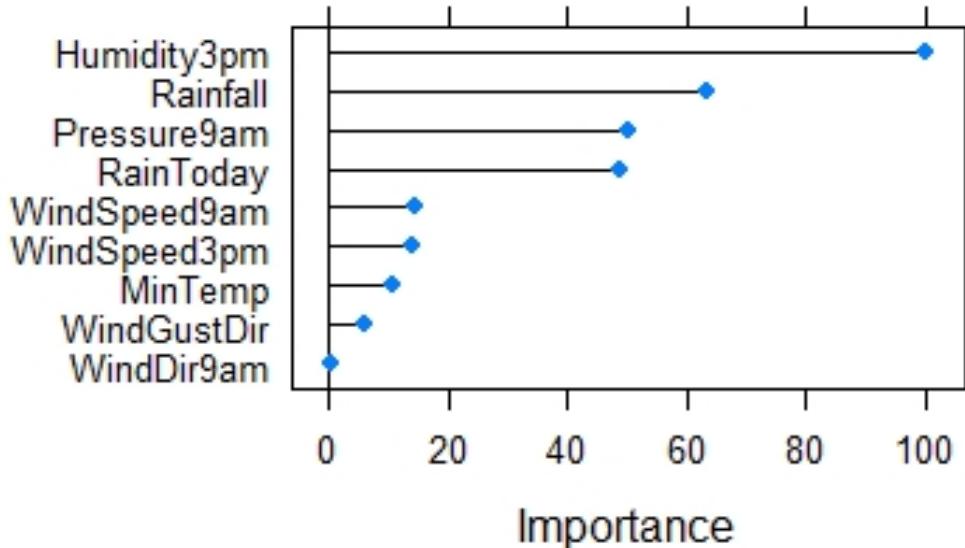


Sono ora presentate le tempistiche e le performance di tale modello.
Tempistiche modello:

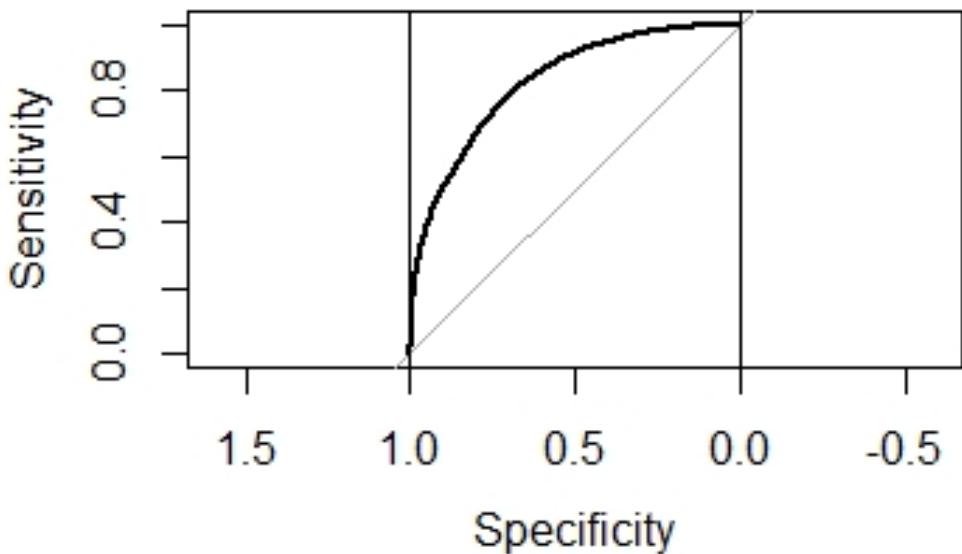
- Tempo creazione modello: 4 ore
- Tempo calcolo prediction: 4 secondi

Metriche performance Modello:

- Accuracy: 0.8463
 - Precision: 0.7370
 - Recall: 0.4874
 - F1 measure: 0.5868
 - AUC: 0.8646
3. **Naive Bayes:** Il terzo modello considerato è stato naive Bayes (method "naive_Bayes"). Tale modello è stato costruito ottimizzando i paramentri di "usekernel", "laplace" e "adjust". Il modello viene costruito in tempi molto brevi ma si nota dalle performance che è l'unico modello che subisce una dimunuzione di prestazioni in confronto allo stesso modello nella modalità holdout, questo può essere dato dalla suddivisione presa in esame nella modalità holdout che performa in modo migliore della media dell'esperimento in 10-fold cross validation. Vengono ora presentati i grafici riguardanti le variabili più importanti per questo modello e il grafico raffigurante la curva ROC.



Questo modello pone la variabile "Humidity3pm" come la più importante per la classificazione seguita da "Rainfall", "Pressure9am" e "RainToday".



Sono ora presentate le tempistiche e le performance di tale modello.

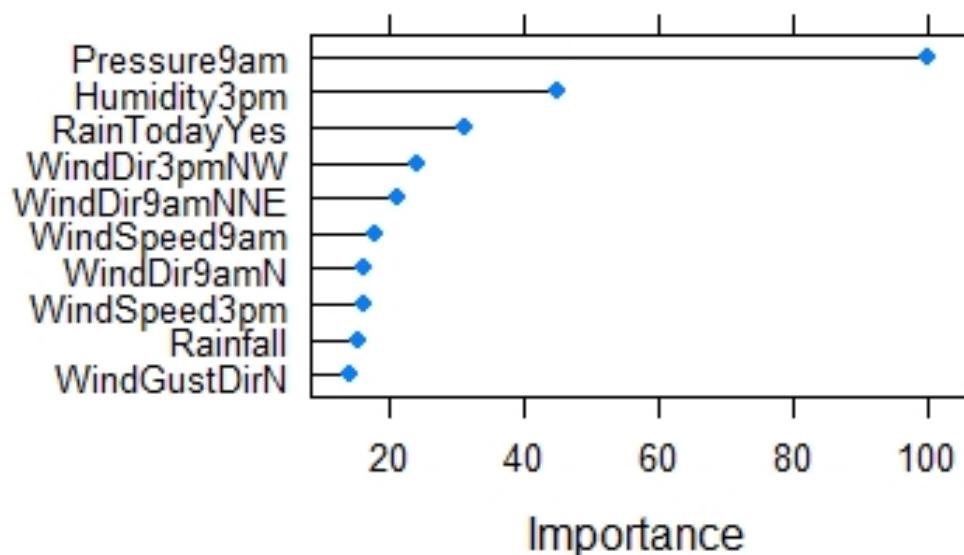
Tempistiche modello:

- Tempo creazione modello: 22 secondi
- Tempo calcolo prediction: 1 secondo

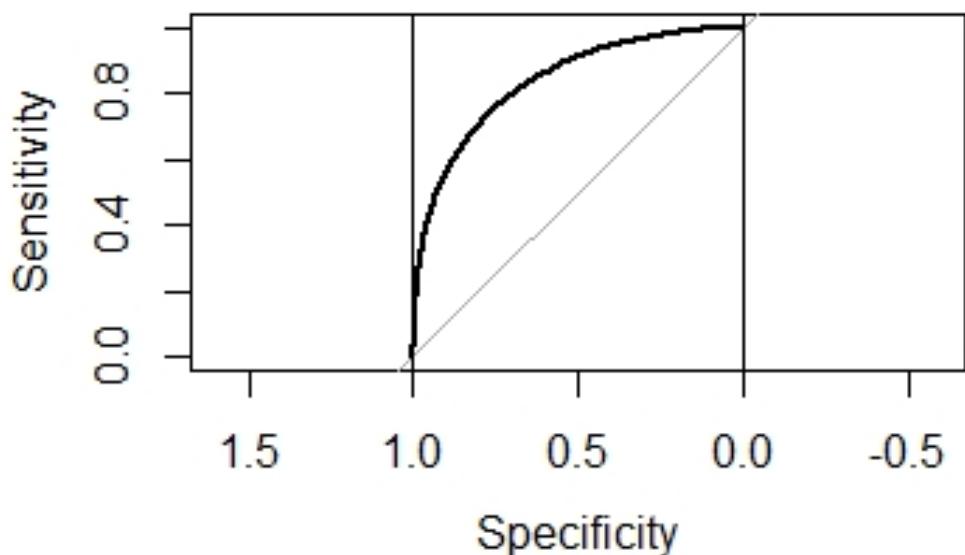
Metriche performance Modello:

- Accuracy: 0.7763
- Precision: 0.600
- Recall: 0.0023
- F1 measure: 0.0047
- AUC: 0.8246

4. **Neural Network:** L'ultimo esperimento effettuato con 10-fold cross validation è stata una neural network (method "nnet"). Tale modello ha mostrato una complessità maggiore rispetto alla media (in particolare maggiore rispetto a naive Bayes e decision tree e minore rispetto a random forest), poiché essa cerca di ottimizzare i parametri possibili, "size" e "decay". Il tempo per la costruzione del modello è quindi da tenere in considerazione insieme alle performance in linea con gli altri modelli. Il package "caret" permette di utilizzare svariate librerie per implementare una neural network quindi potrebbe essere uno sviluppo futuro considerare altre implementazioni e relative ottimizzazioni dei parametri per considerare poi l'alternativa migliore. Vengono ora presentati i grafici riguardanti le variabili più importanti per questo modello e il grafico raffigurante la curva ROC.



Questo modello pone la variabile "Pressure9am" come la più importante per la classificazione seguita da "Humidity3pm", "Rainfall" con valore "Yes" e diverse direzione del vento sia alle 9 di mattina sia alle 3 di pomeriggio.



Sono ora presentate le tempistiche e le performance di tale modello.
Tempistiche modello:

- Tempo creazione modello: 13 minuti
- Tempo calcolo prediction: 1 secondo

Metriche performance Modello:

- Accuracy: 0.8363
- Precision: 0.7065
- Recall: 0.4599
- F1 measure: 0.5571
- AUC: 0.8400

5 Conclusioni

La prima conclusione da evidenziare è sulle performance dei modelli: in generale tutti i modelli hanno mostrato dei valori di precision buoni, segno del fatto che essi riescono in modo positivo a classificare correttamente le osservazioni con target negativo (e questo aiutato dal fatto di avere un dataset sbilanciato) mentre presentano dei valori peggiori per quanto riguarda la metrica di recall, in quanto il compito di classificare correttamente le osservazioni positive ("RainTomorrow" = "Yes") è la vera sfida di questo dataset. Il modello che più soffre sotto questo aspetto è naive Bayes che ha un valore di recall molto basso (sotto l'1%) in quanto assegna solamente a 30 osservazioni un target positivo. Una seconda osservazione conclusiva riguarda l'importanza delle variabili attribuite da ciascun modello, in generale si può osservare come benché ogni modello lavori con tecniche e strategie diverse per capire quali siano le variabili più significative, alla fine le variabili considerate più importanti sono simili, per 3 modelli su 4 la più importante è "Humidity3pm", invece le altre variabili più considerate sono state "Pressure9am", "Rainfall", "MinTemp" e "Raintoday".

Di seguito vengono riportate le tabelle con performance e tempistiche per i modelli sia in modalità holdout sia 10-fold cross validation.

Modelli Holdout						
Nome modello	Timing		Performance			
	Tempo creazione modello	Tempo calcolo prediction	Accuracy	Precision	Recall	F1 measure
Decision tree (pruned)	2 secondi	0.05 secondi	0.8356	0.7511	0.3973	0.5197
Random Forest	1.5 minuti	3.5 secondi	0.8481	0.7140	0.5353	0.6119
Naive Bayes	0.1 secondi	8 secondi	0.8198	0.6505	0.4213	0.5114
Neural Network	1 minuto	0.73 secondi	0.835	0.6688	0.5205	0.5854
SVM (radial kernel)	17 minuti	2 minuti	0.8413	0.7916	0.3948	0.5268

Esperimenti 10 fold cross validation						
Nome Modello	Timing		Performance			
	Tempo creazione modello	Tempo calcolo prediction (probs)	Accuracy	Precision	Recall	F1 measure
Decision tree	19 secondi	1 secondo	0.8356	0.7511	0.3973	0.5197
Random Forest	4 ore	4 secondi	0.8463	0.7370	0.4874	0.5868
Naive Bayes	22 secondi	1 secondo	0.7763	0.6000	0.0023	0.0047
Neural Network	13 minuti	1 secondo	0.8363	0.7065	0.4599	0.5571

L'obiettivo del progetto è stato quello di analizzare e valutare diversi modelli di machine learning applicati al dataset e al contenuto scelto. In linea generale non è semplice dire quale sia il modello migliore e per farlo bisogna tenere in considerazione molti aspetti, tra cui sicuramente complessità computazionale dei modelli e le loro performance cercando il giusto trade-off tra questi elementi. Si può affermare che se l'obiettivo è quello di ottenere un modello semplice e computazionalmente veloce, sacrificando un po' le performance, allora la scelta ricadrebbe su decision tree, che ha il vantaggio di essere anche il modello più semplice da interpretare. Se invece l'obiettivo mira principalmente alle performance, si può optare per modelli più complessi come Random Forest e Neural Network. Come

detto in precedenza, Naive Bayes viene scartato in quanto ha performance inferiori dovute alla difficoltà di predizione della classe positiva. Il modello SVM è stato invece scartato nella fase holdout in quanto troppo oneroso da allenare ed ottimizzare.