PROGETTO IA - ANOMALY DETECTION

Michele Dalla Chiara - VR464051
Davide Zampieri - VR458470

1 Introduzione agli HMM

Gli Hidden Markov Models (HMM) sono dei modelli statistici che hanno numerose applicazioni nel riconoscimento di pattern temporali. Prima di introdurre il funzionamento di un Hidden Markov Model, è necessario definire il concetto di *catena di Markov*.

Concettualmente, la si può vedere come una macchina a stati finiti in cui la transizione da uno stato all'altro avviene con una certa probabilità. La catena di Markov è un **processo stocastico**, ovvero una famiglia di variabili aleatorie nel tempo discreto, che gode della proprietà di Markov, cioè che la probabilità di transizione da uno stato all'altro dipende solamente dallo stato immediatamente precedente e non dall'intera storia degli stati passati (un processo markoviano non ha memoria).

Una catena di Markov a N stati può essere rappresentata tramite una matrice A = {a_i,j} di dimensione NxN. Ogni elemento della matrice a_i,j corrisponde alla probabilità di transizione dallo stato i allo stato j. Dato che i valori rappresentano probabilità, la somma di ogni riga della matrice dev'essere pari a 1. Ovviamente, uno stato può avere una certa probabilità di rimanere nello stesso stato in seguito alla valutazione delle transizioni.

Per completare il modello, è necessario definire anche la distribuzione di probabilità degli stati iniziali, ovvero le probabilità di trovarsi in uno stato all'inizializzazione della macchina a stati. Anche la somma di queste ultime probabilità deve essere pari a 1.

Un **Hidden Markov Model** è una catena di Markov in cui non è possibile osservare direttamente lo stato corrente, per via del fatto che gli stati sono nascosti. Tuttavia, è possibile ottenere un'osservazione legata ad un certo stato. Ad ogni valutazione della macchina a stati, lo stato corrente produrrà un simbolo in uscita, che dipenderà da una distribuzione di probabilità (discreta o continua) caratteristica dello stato in oggetto. Quindi ogni stato può avere una distribuzione di probabilità diversa per l'emissione dei diversi simboli.

Un *HMM continuo* lo descriviamo con i seguenti parametri:

- Un insieme di N stati.
- Una matrice di transizione A = {a_i,j}.
- Una distribuzione di probabilità continua per generare le osservazioni in uscita.
- Una funzione densità della distribuzione continua.

Solitamente, in assenza di assunzioni sui dati, si usa la distribuzione normale, quindi per ogni stato si definiscono i parametri μ e σ. Gli HMM continui sono ideali per modellare quei processi che per loro natura sono continui.

Inoltre, permettono di trattare anche **distribuzioni multivariate**, cioè ad ogni transizione di stato viene emesso, invece che un simbolo, un vettore di osservazioni. Nel caso delle distribuzioni normali multivariate, si rende necessario associare, ad ogni stato, un vettore μ (le medie di ogni dimensione) ed una matrice di covarianza Σ.
Gli *usi degli Hidden Markov Models* sono molteplici, ma nel contesto del machine learning vengono sfruttati per la classificazione di serie temporali. Data una sequenza di osservazioni O = (o_1, o_2, ..., o_n) ed un HMM λ noto, è possibile ottenere un valore che descrive la probabilità che tale sequenza sia stata generata dal modello in esame. Questo valore, denominato likelihood (verosimiglianza), è scritto come P(O | λ). Se si desidera conoscere la sequenza di stati più probabile (Viterbi path), è possibile utilizzare l'algoritmo di Viterbi, il quale restituisce la likelihood lungo la sequenza di stati più probabile.

2 Costruzione di un HMM

L'obiettivo è quello di costruire il modello partendo da un dataset, cioè da un insieme di sequenze di osservazioni. Nel nostro caso il *dataset* contiene dati di velivoli autonomi, inclusi anche quelli relativi a varie anomalie (vedi https://theairlab.org/alfa-dataset/). Per costruire il modello è quindi necessario decidere il numero dei suoi stati (K) e la sua topologia. Per quel che riguarda il numero di stati, non esiste una regola precisa: si può scegliere un valore che si crede ragionevole per l'applicazione; oppure si possono fare più tentativi per trovare il numero di stati che massimizza la qualità del sistema. In genere, sono sufficienti pochi stati (5-10). Per topologia, invece, si intende l'insieme delle transizioni ammesse dal modello (vedi Tutorial — hmmlearn 0.2.4 documentation).
Una volta stabilito il modello, si può procedere alla fase di training, il cui obiettivo è il seguente: dato un insieme di sequenze di osservazioni (train set) indipendenti fra loro si vogliono trovare i parametri di un Hidden Markov Model λ che massimizzino la likelihood di tutte le sequenze. Dal punto di vista funzionale, ciò equivale a trovare i parametri del modello che permettano di descrivere nel modo più fedele possibile le sequenze del train set. Purtroppo, non esistono algoritmi che siano in grado di trovare tale massimo analiticamente, ma sono state sviluppate alcune procedure iterative che permettono di approssimare il modello ottimo. Una di queste è l'algoritmo di Baum-Welch.

3 Algoritmo di Baum-Welch

Questo algoritmo non ha bisogno di conoscere gli stati associati alle osservazioni. Partendo da un modello iniziale (una stima), l'algoritmo esegue iterativamente una procedura chiamata forward-backward convergendo ad un massimo locale della likelihood. Dato che i parametri trovati corrispondono ad un massimo locale (e non globale) la stima iniziale del modello può influenzare fortemente il risultato. Tuttavia nella pratica, i parametri del modello vengono inizializzati casualmente o uniformemente, in quanto l'algoritmo trova una soluzione soddisfacente nella maggior parte dei casi. Negli HMM continui (specialmente se multivariati) è fondamentale scegliere dei parametri ragionevoli per evitare che la procedura converga ad una soluzione inaccettabile.

Fonti

- https://amslaurea.unibo.it/10999/1/GestureRecognitionPavllo.pdf
- Lawrence Rabiner. 1989. A tutorial on hidden Markov models and selected applications in speech recognition.

Analisi del dataset ALFA

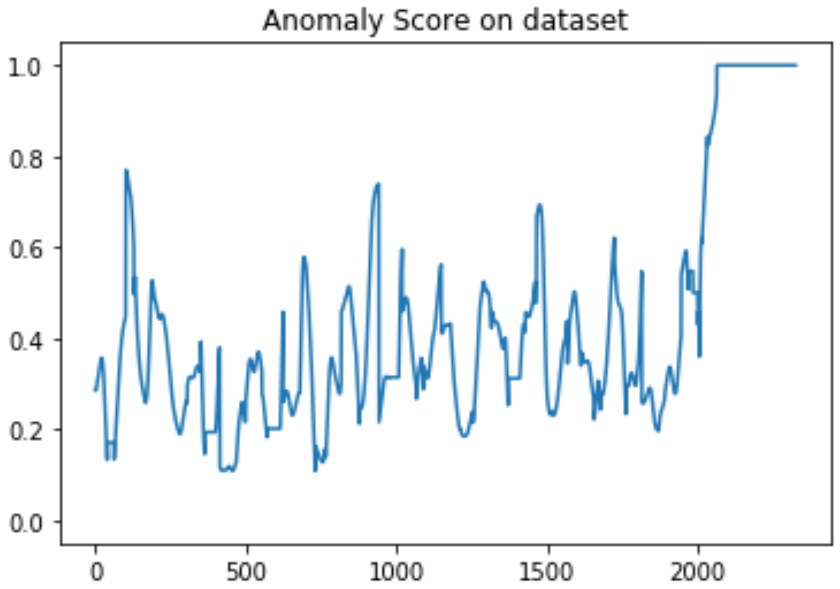
1 Feature selection

Siccome la misura usata per generare gli anomaly score (ovvero la Hellinger Distance) è molto sensibile al numero di feature che vengono usate, si può ricorrere alla feature selection. In particolare, si può utilizzare *RFE* (vedi sklearn.feature_selection.RFE — scikit-learn 0.24.0 documentation (scikit-learn.org)), ovvero un algoritmo che classifica le feature eliminando ricorsivamente le feature stesse. RFE funziona in questo modo: dato un estimatore esterno che assegna pesi alle feature, seleziona le feature considerando ricorsivamente insiemi di feature sempre più piccoli, eliminando di volta in volta le feature meno importanti; tale procedura viene ripetuta fino a raggiungere il numero desiderato di feature.

2 Anomaly detection

La procedura seguita per fare anomaly detection è stata:

1. *Standardizzare* le feature del dataset su scala unitaria (media = 0 e varianza = 1) per avere prestazioni ottimali nella feature selection.
2. Feature selection con *RFE* il quale, dopo varie prove, ci ha permesso di ridurre il numero di dimensioni da 23 a 3, ossia pitch_commanded, throttle e altitude (le più significative per individuare un'anomalia al motore).
3. Miglioramento dei dati con rolling mean e decomposizione per ridurre il *rumore*.
4. Procedura di *fit* (sul train set) tenendo le 3 componenti individuate.
5. Procedura di *evaluation* (su tutto il dataset).
6. Rappresentazione grafica degli *anomaly score* (figura sottostante).



Come si può vedere gli score (Hellinger Distance) rimangono sotto a 1 quando si riferiscono a valori nominali (non anomali), mentre quando cominciano i valori anomali (ultima parte del dataset) si stabilizzano su 1. Ciò accade perché calcoliamo la *Hellinger Distance* per valutare la somiglianza tra la distribuzione di probabilità delle osservazioni (intero dataset) e la distribuzione di probabilità dei valori che sappiamo essere nominali (fit sul train set).

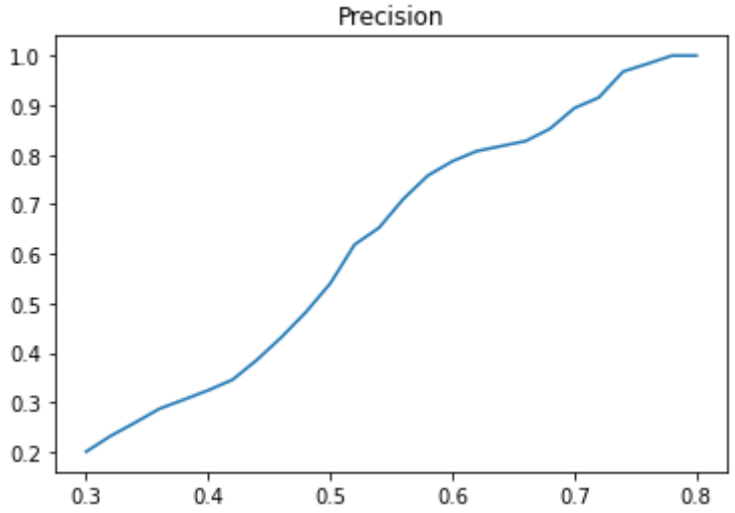
Come detto, i dati usati per fare il fit del modello sono stati migliorati riducendo il rumore attraverso l'applicazione di una lieve rolling mean simmetrica e della decomposizione delle serie temporali. Riguardo quest'ultima trasformazione, consideriamo che una serie temporale definita additivamente può essere scritta come Serie = T + S + R dove R è un processo aleatorio che aggiunge rumore come serie stazionaria in media. Decomponendo la serie ed eliminando la componente dei residui (R) abbiamo ottenuto dei buoni risultati.

Osservando i dati possiamo pensare che il threshold sia intorno a 0,6. I picchi compresi nell'intervallo 0-2000 hanno una durata molto breve e il fenomeno rientra subito al di sotto del threshold per cui potremmo immaginarli come semplici spike che non causano anomalie.

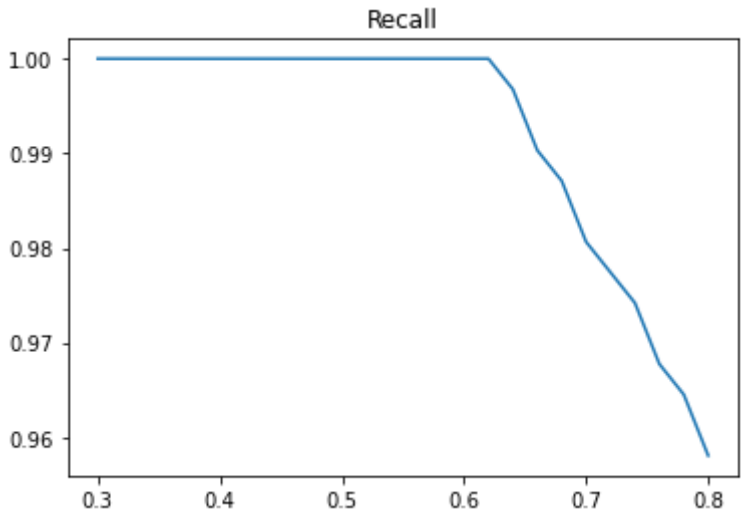
3 Precisione e altre metriche statistiche

Per calcolare la precisione e altre metriche statistiche si è deciso di far variare la *threshold* in un range di 26 valori compresi tra 0,3 e 0,8 (aumentando di 0,02 ad ogni passo). Di seguito diamo la definizione delle varie metriche statistiche utilizzate e presentiamo i grafici dei valori ottenuti al variare della threshold (l'ascissa rappresenta la threshold mentre l'ordinata rappresenta la metrica calcolata).

In un processo di classificazione la **precisione** è definita come $\frac{TP}{TP+FP}$ dove TP identifica i veri positivi cioè quegli oggetti appartenenti ad una *classe C* che vengono correttamente riconosciuti come membri di *C*, mentre FP sono i falsi positivi cioè quegli oggetti che vengono etichettati come appartenenti alla classe *C* ma che in realtà non vi appartengono. Se la precisione assume come valore 0 significa che nessun oggetto di *C* viene riconosciuto come tale, mentre se assume valore 1 significa che tutti gli oggetti etichettati con *C* sono davvero elementi che appartengono all'insieme.

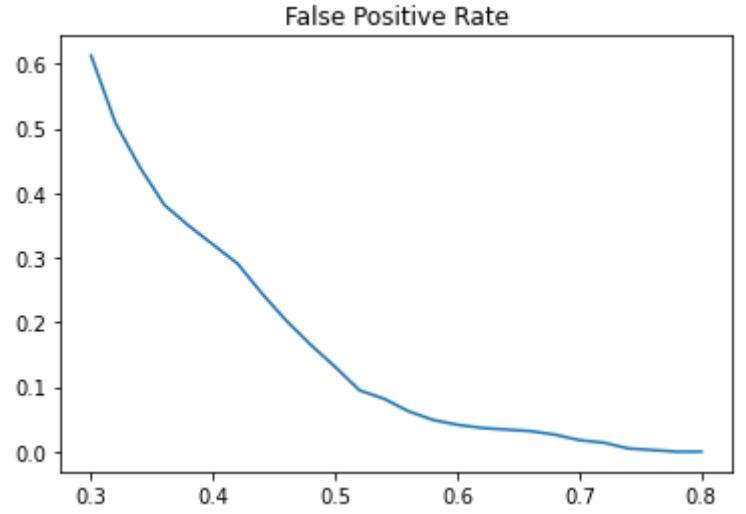


La **recall** (o TPR - True Positive Rate), invece, viene calcolata come $\frac{TP}{TP+FN}$ dove TP sono sempre i veri positivi, mentre FN identifica i falsi negativi cioè quegli oggetti che appartengono a *C* ma che non vengono riconosciuti come tali. La recall viene chiamata anche sensitività, cioè la probabilità che un oggetto di *C* venga etichettato correttamente.



Dal grafico si può notare che finché il threshold è inferiore 0.6 tutti gli stati anomali vengono riconosciuti come tali, tuttavia guardando la precisione (sempre a 0.6) abbiamo un valore intorno a 0.8, ovvero l'80% degli stati etichettati come anomali lo è davvero mentre il rimanente 20% sono stati che non sono anomali.

Il **False Positive Rate** (FPR), chiamato anche fall-out rate o false-alarm rate, indica la probabilità che un oggetto non appartenente all'insieme *C* venga classificato come membro di *C*. Viene calcolato come $\frac{FP}{FP+TN}$ dove TN sono i veri negativi.

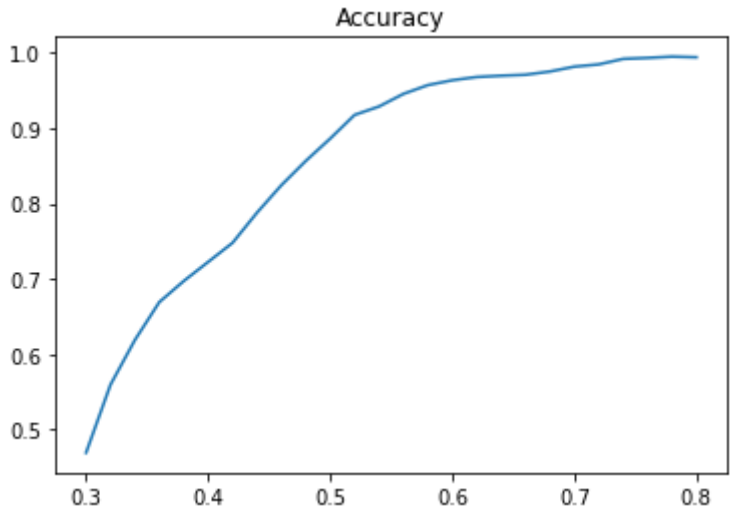


Dal grafico si può notare che all'aumentare del threshold è quasi impossibile (statisticamente è un evento con probabilità pari a 0) che uno stato regolare venga etichettato come anomalo.

L'**accuratezza** misura il grado di vicinanza fra i dati osservati e quelli attesi. È definita come:

$$accuracy = \frac{TP+TN}{P+N} = \frac{TP+TN}{(TP+FP)+(TN+FN)}$$

A partire dalla definizione possiamo definire l'accuracy, nel nostro classificatore binario, come "quanto correttamente riconosciamo gli stati come anomali oppure non anomali". La metrica assume valori alti quando i falsi positivi e i falsi negativi sono valori molto bassi, cioè il classificatore riesce a distinguere molto bene quando un oggetto è membro di *C* oppure non lo è.



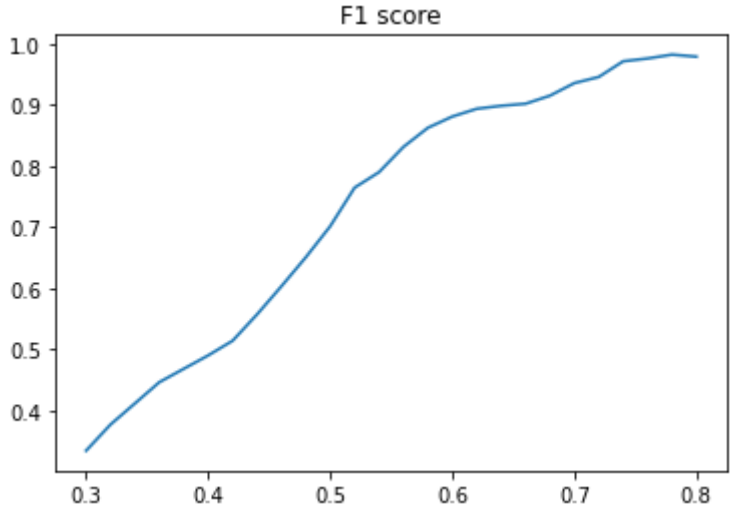
L'**F1 score** è una misura di accuratezza che viene calcolata tramite la media armonica di precision e recall, ovvero:

$$F1 = \frac{2*(precision * recall)}{precision + recall}$$

Ne esiste una versione più generale chiamata $F\beta$calcolata come:

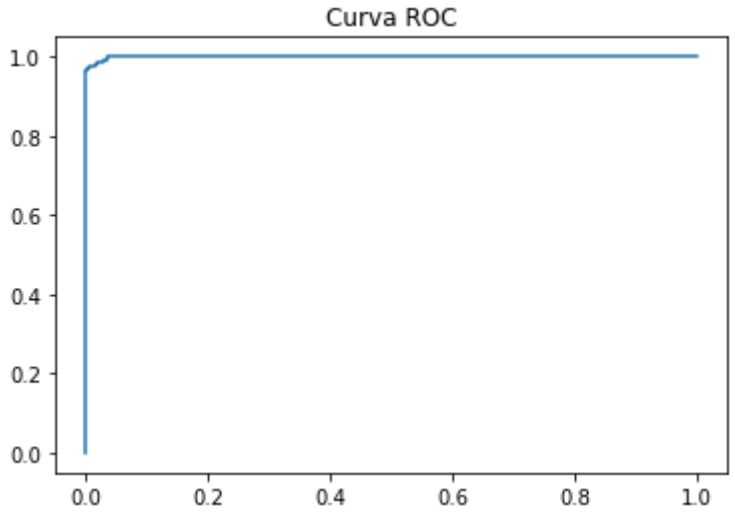
$$F_{\beta} = (1 + \beta^2)\frac{precision*recall}{(\beta^2*precision)+recall}$$

(con βreale positivo) in cui il peso di recall è βvolte maggiore rispetto a quello di precision.



4 Curva ROC

Come ultima cosa abbiamo realizzato la *curva ROC* (Receiver Operating Characteristic), ovvero uno schema grafico che rappresenta il tasso dei veri positivi (TPR) in funzione del tasso dei falsi positivi (FPR). La curva ROC viene creata tracciando il valore del TPR (probabilità di rilevazione) rispetto all'FPR (probabilità di falsi allarmi) per ogni possibile valore della threshold.



Attraverso l'analisi delle curve ROC si valuta la bontà del classificatore. Ad esempio, si può calcolare l'area sottesa alla curva ROC (*AUC* - Area Under Curve), da cui si riesce a determinare la probabilità che il risultato del classificatore applicato ad un individuo scelto casualmente dal gruppo degli appartenenti a C sia superiore al risultato di un individuo del gruppo $\neg C$. Nel caso del nostro classificatore, l'AUC (pari a 0.9993) è molto alto, per cui prendendo a caso uno stato appartenente al gruppo dei valori anomali è quasi certo che il valore del classificatore sia più alto del valore di uno stato che non appartiene ai valori anomali. Infatti, il valore di AUC è molto più vicino a 1 (classificatore perfetto) che a 0.5 (classificatore casuale).

Analisi del dataset dei voli aerei

1 Introduzione

L'ultima parte del progetto consiste nell'effettuare la stessa analisi su un dataset differente. Tale dataset contiene *rotte di aerei reali* divise ognuna in train set (traiettorie programmate) e test set (traiettorie effettuate nella realtà). Nel solo test set è presente una colonna chiamata GT (ground truth) che rappresenta il labeling. Tale colonna contiene 0 quando il valore è nominale e 1 quando il valore è anomalo, ossia quando la traiettoria effettiva presenta uno scostamento da quella programmata).

2 Anomaly detection

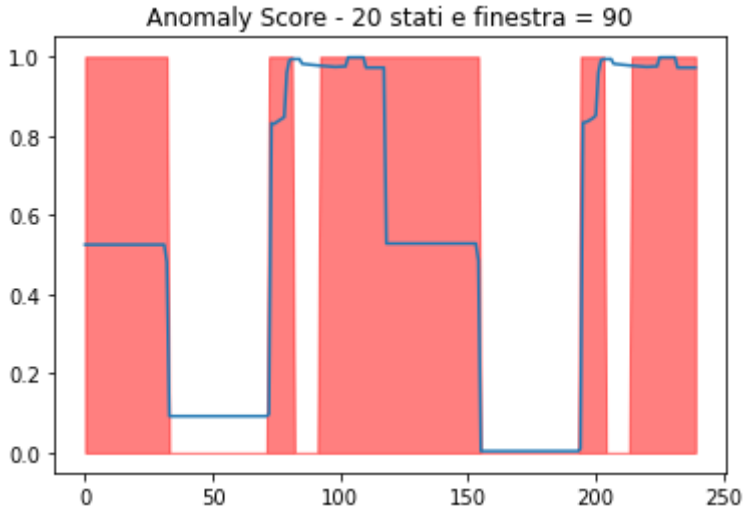
La procedura seguita per fare anomaly detection è stata:

1. Procedura di *fit* sul train set scegliendo il miglior modello basandosi sulla metrica BIC e facendo variare il numero di *stati* prima fino ad un massimo di 20 e poi fino ad un massimo di 30.
2. Procedura di *evaluation* sul test set utilizzando la Hellinger Distance e facendo variare la grandezza della *finestra* fino ad un massimo di 100 osservazioni.
3. Rappresentazione grafica degli *anomaly score* e delle varie *metriche statistiche*.

Rispetto a quanto osservato nell'analisi del dataset precedente, sia provando ad applicare una rolling mean che applicando una decomposizione, non si ottengono migliorie importanti. Questo è dovuto al fatto che i dati variano molto poco e quindi eseguendo una media mobile anche molto piccola si perdono eventuali stati che risultano anomali.

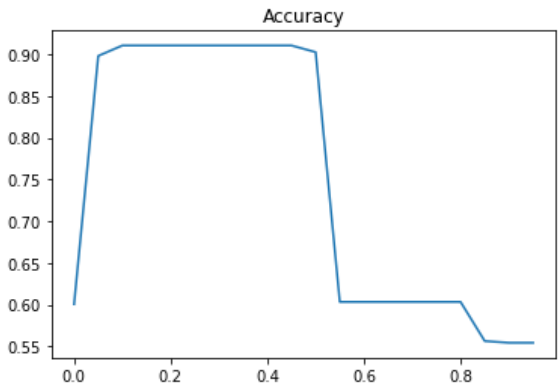
3 Grafici del modello con 20 stati

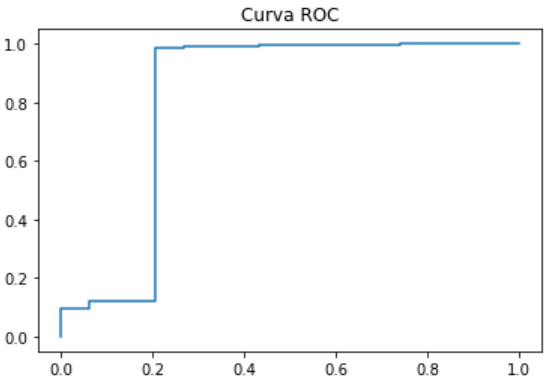
Osservando i valori dell'area sotto la curva ROC (AUC = 0.8156), abbiamo stabilito che la grandezza della finestra ottimale risulta essere pari a 90 osservazioni.



Siccome il pattern degli aerei in realtà è lungo solo un centinaio di osservazioni, esso è stato ripetuto un certo numero di volte per poter avere train e test set di grandezza accettabile. Quindi è normale che gli anomaly score presentino sempre lo stesso andamento; nel grafico sono infatti riportate solo le prime 240 osservazioni, con anche l'informazione se si tratta realmente di valori anomali o meno (l'area colorata di rosso indica le anomalie reali).

Tuttavia, guardando il grafico dell'*accuratezza* (grado di corrispondenza tra un risultato e un valore di riferimento) si può notare come questo non rispetti il trend che si era ottenuto nell'analisi del dataset precedente. Probabilmente ciò è dovuto al fatto che la soglia ottimale sembra essere compresa tra 0.2 e 0.5, in quanto per valori maggiori si comincia a classificare come nominali (sotto soglia) molti valori che in realtà sono anomali (ancora peggio per valori di soglia maggiori di 0.8).



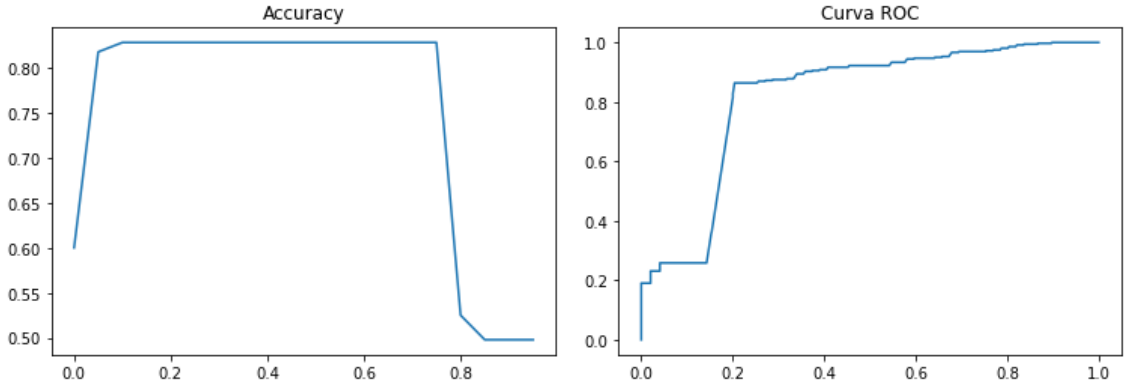


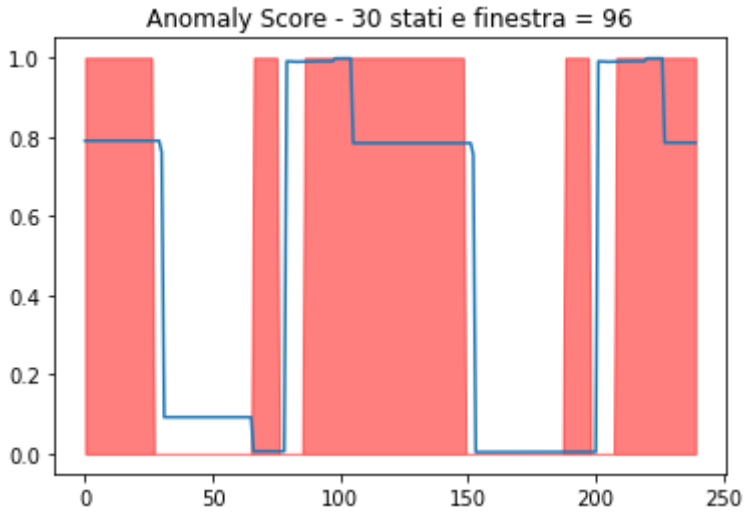
4 Grafici del modello con 30 stati

A questo punto si è deciso di aumentare il numero di stati per la ricerca del miglior modello fino ad un massimo di 30. Ancora una volta il modello che ha il massimo numero di stati risulta essere quello con BIC minore. Poi, abbiamo osservato i valori dell'area sotto la curva ROC al variare della grandezza della finestra, ottenendo il seguente andamento:

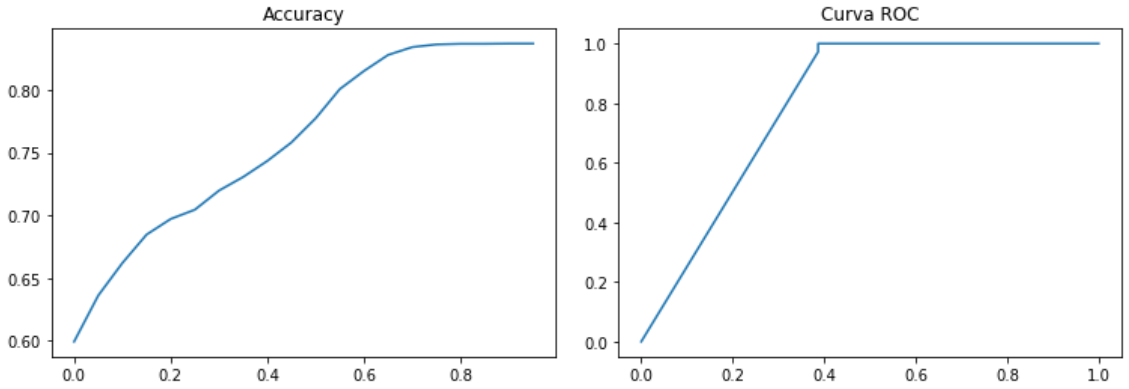
10	0.8013389976037217
22	0.6418619760090685
24	0.6164373401967591
26	0.5647323259426634
28	0.48222168827277406
30	0.26458148732834275
32	0.19909087305407533
34	0.17007203541127064
36	0.15860290989534587
...	...
86	0.7659242355899023
88	0.7707765728842219
90	0.7766116611153479
92	0.7991044012518804
94	0.810550316088947
96	0.8182805467200357
98	0.8049715427196839
100	0.7938397774075617

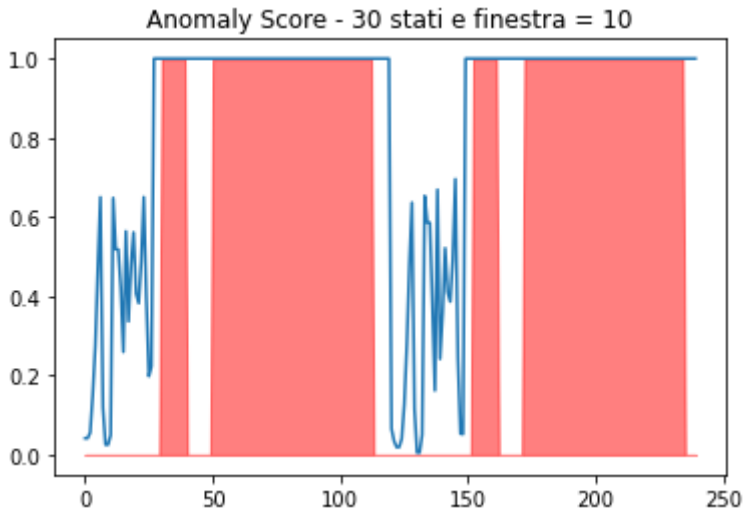
Si può quindi notare che sono presenti due punti di massimo (evidenziati in rosso), i quali ci hanno suggerito di provare a fare l'analisi di anomaly detection per entrambi i valori della finestra. I risultati dell'analisi con finestra pari a 96 sono in linea con i precedenti.





Però si riesce anche a notare che il problema potrebbe risiedere nel fatto che ci sono valori anomali seguiti da valori nominali in uno spazio molto piccolo (nell'intervallo 50-100 e così via). Quindi fare l'analisi con finestra pari a 10 sembrerebbe ragionevole.





Infatti, ora si può notare che il grafico dell'*accuratezza* rispetta il trend che si era ottenuto in precedenza. Analizzando anche il grafico degli *anomaly score* si può vedere che gli spazi relativi ai valori nominali e alle anomalie vengono identificati correttamente, a meno del piccolo intervallo di valori nominali che si trova tra i valori anomali. Proprio alla luce di questo possiamo constatare che per avere un'accuratezza dell'80% si deve scegliere una soglia pari almeno a 0,6.