

Analisi dei test effettuati 220119

Test di Stefano Biddau su MNIST con SVM

Cenni teorici SVM

DEFINIZIONE:

La Support Vector Machines (SVM) è un insieme di metodi di apprendimento supervisionato che possono essere utilizzati sia per fare classificazione sia per fare regressione.

FUNZIONAMENTO

L'obiettivo dell'algoritmo della macchina vettoriale di supporto è trovare un iperpiano in uno spazio N-dimensionale ($N \rightarrow$ il numero di caratteristiche) che classifichi distintamente i punti dati. Per separare le N classi di punti dati ci sono molti possibili iper-piani che potrebbero essere scelti. Lo scopo è trovare un piano che abbia il margine massimo, ovvero la distanza tra i punti dati delle seguenti classi. Massimizzare la distanza del margine fornisce un certo rinforzo in modo che i punti dati futuri possano essere classificati con maggiore sicurezza. I vettori di supporto, invece, sono punti dati più vicini all'iper-piano e influenzano la sua posizione e il suo orientamento. Utilizzando questi vettori di supporto si massimizza il margine del classificatore. L'eliminazione dei vettori di supporto cambierà la posizione dell'iper-piano. Questi sono i punti che aiutano la costruzione dell'SVM.

PRINCIPALI IPER-PARAMETRI DEL MODELLO

Anche per le SVM ci sono diversi iper- parametri che ne influenzano le prestazioni e l'accuratezza nella classificazione.

Principalmente abbiamo la **funzione del kernel** che è un metodo utilizzato per prendere i dati come input e trasformarli nella forma richiesta di elaborazione dei dati. Il "Kernel" viene utilizzato a causa di una serie di funzioni matematiche utilizzate in SVM che forniscono la finestra per manipolare i dati. Quindi la funzione del kernel generalmente trasforma l'insieme dei dati di addestramento in modo che una superficie decisionale non lineare sia in grado di trasformarsi in un'equazione lineare in un numero maggiore di spazi dimensionali. **In altre parole, restituisce il prodotto interno tra due punti in una dimensione caratteristica standard.**

Possono essere di diverso tipo:

- **Funzione di base radiale del kernel gaussiana (RBF):** viene utilizzata per eseguire la trasformazione quando non si hanno conoscenze preliminari sui dati, aggiungendo il metodo della base radiale per migliorare la trasformazione;
- **Sigmoid Kernel:** questa funzione equivale ad un modello perceptron a due strati della rete neurale, che viene utilizzato come funzione di attivazione per i neuroni artificiali;
- **Kernel Polinomiale:** rappresenta la somiglianza dei vettori nell'insieme dei dati di addestramento in uno spazio di funzionalità rispetto ai polinomi delle variabili originali utilizzate nel kernel;
- **Kernel lineare:** utilizzato quando i dati sono separabili linearmente;

Altro iper parametro importante è il **valore C** che compensa la corretta classificazione degli esempi di addestramento con la massimizzazione del margine della funzione di decisione. Per i valori **maggiori di C** sarà accettato un margine inferiore se la funzione di decisione è migliore nel classificare correttamente tutti i punti di allenamento. Un valore **più basso di C** incoraggerà un

marginale più ampio, quindi una funzione decisionale più semplice, a scapito dell'accuratezza dell'allenamento. **In altre parole C si comporta come un parametro di regolarizzazione nella SVM.**

Infine un altro parametro molto importante da tenere in considerazione è **gamma**. Esso definisce fino a che punto arriva l'influenza di un singolo esempio di addestramento, con valori bassi che significano "lontano" e valori alti che significano "vicino". I parametri gamma possono essere visti come l'inverso del raggio di influenza dei campioni selezionati dal modello come vettori di supporto. Questo parametro è associato a tutte le diverse funzioni del kernel **tranne quella lineare**.

PRO E CONTRO DELLA RANDOM FOREST

PRO

- Efficace in spazi ad alta dimensione;
- Ancora efficace nei casi in cui il numero di dimensioni è maggiore del numero di campioni;
- Utilizza un sottoinsieme di punti di addestramento nella funzione decisionale, quindi è anche efficiente in termini di memoria;
- É versatile, in quanto è possibile specificare diverse funzioni del kernel per la funzione di decisione;

CONTRO

- Le SVM non forniscono direttamente stime di probabilità, queste sono calcolate utilizzando 5 costose convalide incrociate;

TEST EFFETTUATI

MNIST

Dai grafici di accuratezza generati dall'esecuzione dello script per l'ottimizzazione degli Iper parametri notiamo che

1° Blocco di test (Aumento profondità, c: Gini, mf: None, msl: 1)

Indice test	Profondità massima	Criterio	Max features	Min samples leaf	Acc Train	Acc Test	Differenza MSE	Over-fit
1	11	Gini	None	1	92,76%	87,49%	0,84	NO
2	13	Gini	None	1	96,50%	88,03%	1,43	NO
3	15	Gini	None	1	98,44%	88,20%	1,71	NO
4	17	Gini	None	1	99,01%	88,21%	1,83	NO
5	21	Gini	None	1	99,57%	88,02%	1,94	SI
6	25	Gini	None	1	99,73%	88,01%	1,99	SI

Da questi primi 6 test si evince che all'aumentare della profondità massima dell'albero decisionale aumenta la percentuale di accuratezza sul set di train, mentre aumenta più lentamente quella su set di test che mantiene sempre una differenza di circa 10 punti percentuali, a partire dalla profondità di 21 l'accuratezza sul set di test inizia addirittura a calare avviandosi verso l'overfitting. Il miglior risultato ottenuto è il **test 4** in termini di percentuale di accuratezza più alta sul set di test, tuttavia è da considerare ottimo anche il **test 1** in cui la differenza tra l'errore sul set di train e l'errore sul set di test è davvero minima.

2° Blocco di test (Aumento profondità, c: Gini, mf: None, msl: 25)

Indice test	Profondità massima	Criterio	Max features	Min samples leaf	Acc Train	Acc Test	Differenza MSE	Over-fit
1	11	Gini	None	25	87,99%	85,79%	0,23	NO
2	13	Gini	None	25	88,53%	86,04%	0,32	NO
3	15	Gini	None	25	88,61%	86,13%	0,32	NO
4	17	Gini	None	25	88,61%	86,13%	0,32	NO
5	21	Gini	None	25	88,63%	86,15%	0,33	NO
6	25	Gini	None	25	88,63%	86,16%	0,32	NO
7	30	Gini	None	25	88,63%	86,15%	0,32	NO
8	35	Gini	None	25	88,63%	86,15%	0,32	NO

Per questi 8 test effettuati raccogliamo degli ottimi risultati. Aumentando a 25 il numero di esemplari minimi che finiscono nelle foglie, abbiamo trovato una combinazione di parametri che stabilizza dopo una certa profondità i valori di accuratezza sia sul set di train che sul set di test, che oltretutto mostrano avere dei livelli di accuratezza in termini di percentuale molto vicini.

Questa cosa ci garantisce l'assenza di overfitting. Il miglior risultato ottenuto è il **test 6** in termini di percentuale di accuratezza più alta sul set di test ma anche per differenza tra l'errore sul set di train e di test che è minima. Unica pecca data l'estrema semplicità del dataset le percentuali di accuratezza sono leggermente basse per gli standard.

3° Blocco di test (aumento profondità, c: Gini, mf: None, msl: 50)

Indice test	Profondità massima	Criterio	Max features	Min samples leaf	Acc Train	Acc Test	Differenza MSE	Over-fit
1	11	Gini	None	50	85,64%	84,53%	0,05	NO
2	13	Gini	None	50	85,87%	84,66%	0,08	NO
3	15	Gini	None	50	85,87%	84,65%	0,08	NO
4	17	Gini	None	50	85,87%	84,66%	0,08	NO

5	21	Gini	None	50	85,87%	84,66%	0,08	NO
6	25	Gini	None	50	85,87%	84,66%	0,08	NO

Per questi 6 test effettuati notiamo un'analogia col 2° blocco di test effettuati, in cui dopo aver raggiunto al **test 2** il miglior risultato sia in termini di percentuali sull'accuratezza del set di test, questo si stabilizza. Unica pecca data l'estrema semplicità del dataset le percentuali di accuratezza sono leggermente basse per gli standard.

*Ripetiamo adesso i 3 blocchi di test effettuati cambiando il parametro **max_features** da **None** (standard) a **0,5** e vediamo come si comporta il classificatore.*

4° Blocco di test (Aumento profondità, c: Gini, mf: 0,5, msl: 1)

Indice test	Profondità massima	Criterio	Max features	Min samples leaf	Acc Train	Acc Test	Differenza MSE	Over-fit
1	11	Gini	0,5	1	92,59%	86,94%	0,86	NO
2	13	Gini	0,5	1	95,80%	87,46%	1,35	NO
3	15	Gini	0,5	1	98,22%	87,88%	1,62	NO
4	17	Gini	0,5	1	99,01%	88,51%	1,68	NO
5	21	Gini	0,5	1	99,51%	87,34%	1,97	SI
6	25	Gini	0,5	1	99,68%	87,57%	1,96	SI
7	30	Gini	0,5	1	99,87%	87,36%	2.1	SI

I risultati ottenuto da questi 7 test si mostrano in linea con l'andamento del 1° blocco di test effettuati. Il miglior risultato ottenuto è il **test 4** in termini di percentuale di accuratezza più alta sul set di test, tuttavia è da considerare ottimo anche il **test 1** in cui la differenza tra l'errore sul set di train e l'errore sul set di test è davvero minima. I valori ottenuti sono più alti anche se di poco rispetto a quelli ottenuti dal 1° blocco di test di sotto mostrata una comparazione:

- 4° blocco test 4: **88,51%**
- 1° blocco test 4: **88,21 %**

Come per il primo blocco anche per questi test dopo la soglia di profondità 17 l'accuratezza sul set di test inizia a calare in termini di percentuale avvicinandosi verso l'overfitting.

5° Blocco di test (Aumento profondità, c: Gini, mf: 0,5, msl: 25)

Indice test	Profondità massima	Criterio	Max features	Min samples leaf	Acc Train	Acc Test	Differenza MSE	Over-fit
1	11	Gini	0,5	25	87,12%	85,30%	0,25	NO
2	13	Gini	0,5	25	87,67%	85,26%	0,30	NO
3	15	Gini	0,5	25	88,00%	85,60%	0,31	NO
4	17	Gini	0,5	25	88,18%	85,95%	0,36	NO
5	21	Gini	0,5	25	87,95%	85,37%	0,35	NO
6	25	Gini	0,5	25	88,27%	85,83%	0,37	NO
7	30	Gini	0,5	25	88,19%	86,00%	0,38	NO
8	35	Gini	0,5	25	88,19%	86,00%	0,38	NO

I risultati ottenuti da questo blocco di 8 test evidenziano un andamento altalenante delle percentuali di accuratezza sia sul set di test che sul set di train. All'aumentare della profondità dell'albero l'accuratezza sul set di train cresce e decresce fino ad arrivare alla profondità di 30 oltre il quale il tasso di accuratezza sui set si stabilizza. Infatti i **test 7,8** sono i migliori ottenuti in termini di punti percentuali, da considerare anche il **test 1** che mostra una differenza di errore tra i due set di dati veramente minima con dei valori percentuali di accuratezza rispettabili.

Rispetto al 2° blocco di test i risultati sono lievemente più bassi in termini di percentuali di accuratezza, la cosa che li accomuna è che entrambe i set di parametri portano dopo una certa profondità ad una stabilizzazione dei valori.

6° Blocco di test (Aumento profondità, c: Gini, mf: 0,5, msl: 50)

Indice test	Profondità massima	Criterio	Max features	Min samples leaf	Acc Train	Acc Test	Differenza MSE	Over-fit
1	11	Gini	0,5	50	84,45%	83,44%	0,20	NO
2	13	Gini	0,5	50	84,62%	83,63%	0,20	NO

3	15	Gini	0,5	50	84,86%	82,99%	0,33	NO
4	17	Gini	0,5	50	84,98%	84,09%	0,012	NO
5	21	Gini	0,5	50	84,82%	83,40%	0,20	SI
6	25	Gini	0,5	50	84,83%	82,78%	0,33	SI

Da questi 6 test notiamo un comportamento estremamente peggiore rispetto al 3° blocco di test con max_features: None in cui addirittura si va in overfitting a partire da una profondità di 21 dell'albero, **il test 4** risulta il migliore sia in termini di percentuale di accuratezza sia in termini di differenza di errore tra il set di train e quello di test che risulta essere il più piccolo fin ora ottenuto tuttavia, le percentuali di accuratezza sono molto basse per il dataset MNIST.

*Ora ripetiamo questi 6 blocchi di test cambiando il **criterio** da **Gini** -> **Entropy** e vediamo se otteniamo migliori risultati*

7° Blocco di test (Aumento profondità, c: Entropy, mf: None, msl: 1)

Indice test	Profondità massima	Criterio	Max features	Min samples leaf	Acc Train	Acc Test	Differenza MSE	Over-fit
1	11	Entropy	None	1	94,16%	88,08%	0,95	NO
2	13	Entropy	None	1	98,19%	89,09%	1,50	NO
3	15	Entropy	None	1	99,52%	88,75%	1,74	SI
4	17	Entropy	None	1	99,88%	88,61%	1,92	SI
5	21	Entropy	None	1	100%	88,63%	1,91	SI
6	25	Entropy	None	1	100%	88,64%	1,92	SI

Dopo questo gruppo di 6 test possiamo evincere che il criterio Entropy rispetto al Gini fa lievitare di molto sia l'accuratezza sul set di train che sul set di test. Registriamo infatti con il **test 2** i migliori risultati fin ora ottenuti con un'accuratezza sul set di test che sfiora il 90%. Tuttavia notiamo anche in negativo che il criterio entropy accelera il processo di raggiungimento dell'overfitting infatti come si può vedere con il **test 5** e il **test 6** la percentuale di accuratezza sul set di train è pari al **100%** che è sinonimo di un **modello inutile**.

8° Blocco di test (Aumento profondità, c: Entropy, mf: None, msl: 25)

Indice test	Profondità massima	Criterio	Max features	Min samples leaf	Acc Train	Acc Test	Differenza MSE	Over-fit
1	11	Gini	None	25	88,81%	86,52%	0,30	NO
2	13	Gini	None	25	89,32%	86,86%	0,36	NO
3	15	Gini	None	25	89,34%	86,86%	0,36	NO
4	17	Gini	None	25	89,34%	86,86%	0,36	NO
5	21	Gini	None	25	89,34%	86,85%	0,37	NO
6	25	Gini	None	25	89,34%	86,85%	0,37	NO

Da questi 6 test si evince che dopo aver raggiunto un picco massimo (**test 2**) sia in termini di accuratezza sul set di test e di train, sia per la differenza di errore tra i due set, i valori di accuratezza si stabilizzano all'aumentare della profondità evitando di andare in overfitting. Il criterio **entropy** a differenza di **Gini** ci ha permesso di stabilizzare i valori di accuratezza sui set di dati a partire da una profondità più piccola.

9° Blocco di test (Aumento profondità, c: Entropy, mf: None, msl: 50)

Indice test	Profondità massima	Criterio	Max features	Min samples leaf	Acc Train	Acc Test	Differenza MSE	Over-fit
1	11	Entropy	None	50	86,19%	84,76%	0.17	NO
2	13	Entropy	None	50	86,31%	84,72%	0,21	NO
3	15	Entropy	None	50	86,31%	84,72%	0,21	NO
4	17	Entropy	None	50	86,31%	84,72%	0,21	NO
5	21	Entropy	None	50	86,31%	84,72%	0,21	NO

6	25	Entropy	None	50	86,31%	84,72%	0,21	NO
---	----	---------	------	----	--------	--------	------	----

Da questi 6 test effettuati si evince come per il medesimo blocco (n°6) di come il parametro **min_samples_leaf** settato a **50** stabilizzi i valori di accuratezza sia sul set di train che sul set di test dopo una certa profondità, il criterio entropy anche in questo caso svolge un ruolo sia di aumento del valore in percentuale su tutti e due i set sia stabilizzando prima il modello dopo i miglior risultato ottenuto nel **test 1**.

10° Blocco di test (Aumento profondità, c: Entropy, mf: 0,5, msl: 1)

Indice test	Profondità massima	Criterio	Max features	Min samples leaf	Acc Train	Acc Test	Differenza MSE	Over-fit
1	11	Entropy	0,5	1	93,63%	87,50%	1,07	NO
2	13	Entropy	0,5	1	97,76%	87,95%	1,61	NO
3	15	Entropy	0,5	1	99,41%	88,39%	1,82	NO
4	17	Entropy	0,5	1	99,81%	88,27%	1,97	NO
5	21	Entropy	0,5	1	99,96%	88,35%	1,99	SI
6	25	Entropy	0,5	1	100%	88,12%	2,01	SI

Da questo gruppo di 6 test effettuati si evince un comportamento analogo al blocco i test effettuato con criterio **Gini**, con l'unica differenza che abbiamo un aumento dei valori di accuratezza su entrambe i set di dati, ma anche un avvicinamento più veloce all'overfitting all'aumentare della profondità dell'albero, infatti dopo aver raggiunto i risultati migliori con il **test 3** il modello diventa inutilizzabile.

11° Blocco di test (Aumento profondità, c: Entropy, mf: 0,5, msl: 25)

Indice test	Profondità massima	Criterio	Max features	Min samples leaf	Acc Train	Acc Test	Differenza MSE	Over-fit
1	11	Entropy	0,5	25	88,26%	85,95%	0,38	NO
2	13	Entropy	0,5	25	89,05%	86,10%	0,52	NO
3	15	Entropy	0,5	25	88,56%	86,42%	0,35	NO
4	17	Entropy	0,5	25	88,56%	85,97%	0,36	NO
5	21	Entropy	0,5	25	88,85%	86,38%	0,41	SI

Indice test	Profondità massima	Criterio	Max features	Min samples leaf	Acc Train	Acc Test	Differenza MSE	Over-fit
6	25	Entropy	0,5	25	88,85%	86,38%	0,41	SI

Questo gruppo di 6 test mostra delle analogie con il blocco di test effettuati con gli stessi parametri e criterio **Gini**, ormai sembra di aver compreso che il criterio **Entropy** fa lievitare il tasso di accuratezza sui set sia di train che di test. Per questo blocco il miglior risultato è stato ottenuto dal **test 3** gli altri ci mostrano un andamento per quanto riguarda la percentuale di accuratezza altalenante fino a stabilizzarsi a partire da una profondità di 21 dell'albero.

12° Blocco di test (Aumento profondità, c: Entropy, mf: 0,5, msl: 50)

Indice test	Profondità massima	Criterio	Max features	Min samples leaf	Acc Train	Acc Test	Differenza MSE	Over-fit
1	11	Entropy	0,5	50	85,20%	83,76%	0,34	NO
2	13	Entropy	0,5	50	86,16%	84,74%	0,27	NO
3	15	Entropy	0,5	50	85,52%	84,73%	0,23	NO
4	17	Entropy	0,5	50	85,57%	84,57%	0,06	NO
5	21	Entropy	0,5	50	85,73%	84,77%	0,13	SI
6	25	Entropy	0,5	50	85,73%	84,77%	0,13	SI

Questo gruppo di 6 test mostra un comportamento molto diverso da quello analogo (6° Blocco), qui notiamo di come ci sia una costante crescita dei valori di accuratezza sul set di test, i migliori risultati si ottengono con il **test 5** dopodiché i valori di ottimali si stabilizzano.

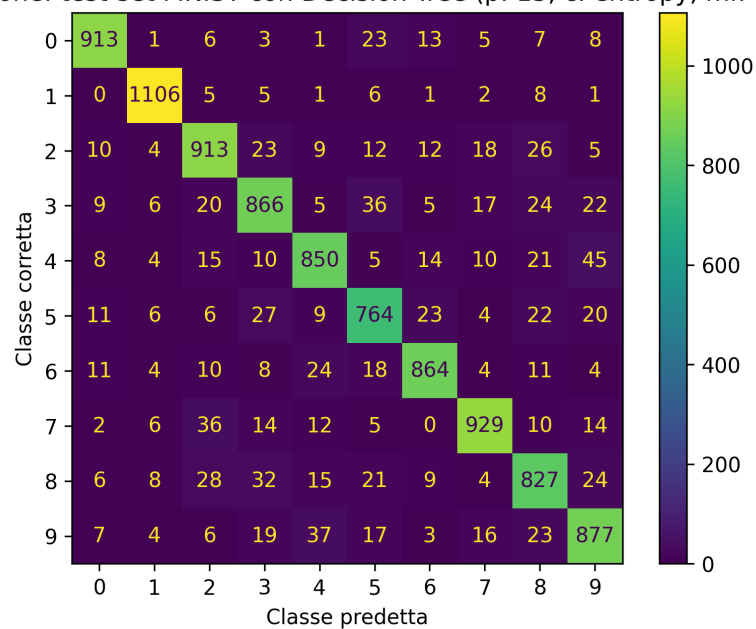
Possiamo stilare una lista di 3 test, quelli con i migliori risultati ottenuti in termini di percentuale di accuratezza sul set di test:

- 1) Blocco 7 - Test 2: Test del 2022-10-31 12:46:39
- 2) Blocco 4 - Test 4: Test del 2022-10-30 17:34:43
- 3) Blocco 10 - Test 3: Test del 2022-11-01 17:24:20

1) Test del 2022-10-31 12:46:39 MNIST con Decision Tree (profondità: 13, c: entropy, mf: None, msl:1)

Report di classificazione					
	precision	recall	f1-score	support	
0	0.93	0.93	0.93	980	
1	0.96	0.97	0.97	1135	
2	0.87	0.88	0.88	1032	
3	0.86	0.86	0.86	1010	
4	0.88	0.87	0.87	982	
5	0.84	0.86	0.85	892	
6	0.92	0.90	0.91	958	
7	0.92	0.90	0.91	1028	
8	0.84	0.85	0.85	974	
9	0.86	0.87	0.86	1009	
accuracy			0.89	10000	
macro avg	0.89	0.89	0.89	10000	
weighted avg	0.89	0.89	0.89	10000	

Matrice di Confusione: test-set MNIST con Decision Tree (p: 13, c: entropy, mf: None, msl: 1)



Dal report di classificazione notiamo che il nostro classificatore ottiene i migliori risultati classificando la **classe 1 -> cifra 1** e i peggiori risultati di classificazione con la **classe 8 -> cifra 8**

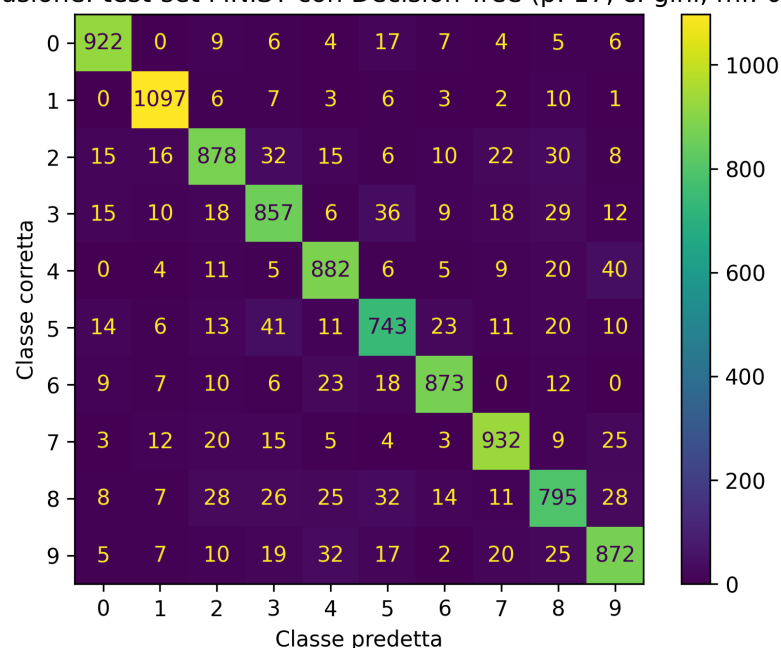
La matrice di confusione conferma ciò che viene mostrato nel report di classificazione la classe che corrisponde alla **cifra 1** mostra su circa **1135** campioni di **cifra 1** **1106** vengono classificati correttamente, analogamente la classe che corrisponde alla **cifra 8** è quella che ottiene i peggiori risultati infatti su **974** campioni disponibili solo **827** vengono classificati correttamente. Risaltano all'occhio **32** campioni di **cifra 8** classificati come **cifra 3** e **28** campioni di **cifra 8** classificata come **cifra 2**

2) Test del 2022-10-30 17:34:43 MNIST con Decision Tree (profondità: 17, c: gini, mf: 0.5, msl: 1)

Report di classificazione				
	precision	recall	f1-score	support
0	0.93	0.94	0.94	980
1	0.94	0.97	0.95	1135
2	0.88	0.85	0.86	1032
3	0.85	0.85	0.85	1010
4	0.88	0.90	0.89	982
5	0.84	0.83	0.84	892
6	0.92	0.91	0.92	958
7	0.91	0.91	0.91	1028
8	0.83	0.82	0.82	974
9	0.87	0.86	0.87	1009
accuracy			0.89	10000
macro avg	0.88	0.88	0.88	10000
weighted avg	0.88	0.89	0.88	10000

Dal report di classificazione notiamo che il nostro classificatore ottiene i migliori risultati classificando la **classe 1 -> cifra 1** e i peggiori risultati di classificazione con la **classe 8 -> cifra 8**

Matrice di Confusione: test-set MNIST con Decision Tree (p: 17, c: gini, mf: 0.5, msl: 1)



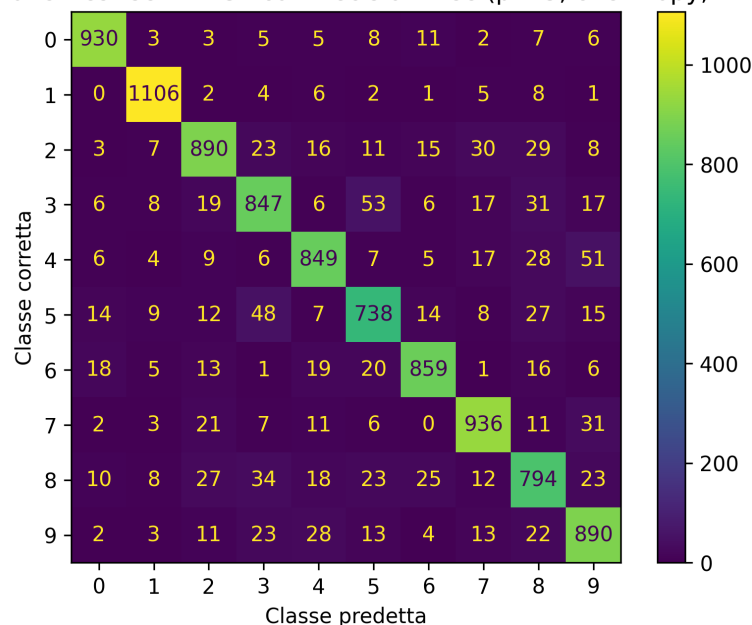
La matrice di confusione conferma ciò che viene mostrato nel report di classificazione la classe che corrisponde alla **cifra 1** mostra su circa **1135** campioni di **cifra 1** **1097** vengono classificati correttamente, analogamente la classe che corrisponde alla **cifra 8** è quella che ottiene i peggiori risultati infatti su **974** campioni disponibili solo **795** vengono classificati correttamente. Risaltano all'occhio **32** campioni di **cifra 8** classificati come **cifra 5**, **28** campioni di **cifra 8** classificata come **cifra 2** e **26** campioni di **cifra 8** classificati come **cifra 3**

3) Test del 2022-11-01 17:24:20 MNIST con Decision Tree (profondità: 15, c: entropy, mf: 0.5, msl: 1)

Report di classificazione				
	precision	recall	f1-score	support
0	0.94	0.95	0.94	980
1	0.96	0.97	0.97	1135
2	0.88	0.86	0.87	1032
3	0.85	0.84	0.84	1010
4	0.88	0.86	0.87	982
5	0.84	0.83	0.83	892
6	0.91	0.90	0.91	958
7	0.90	0.91	0.90	1028
8	0.82	0.82	0.82	974
9	0.85	0.88	0.87	1009
accuracy			0.88	10000
macro avg	0.88	0.88	0.88	10000
weighted avg	0.88	0.88	0.88	10000

Dal report di classificazione notiamo che il nostro classificatore ottiene i migliori risultati classificando la **classe 1 -> cifra 1** e i peggiori risultati di classificazione con la **classe 8 -> cifra 8**

Matrice di Confusione: test-set MNIST con Decision Tree (p: 15, c: entropy, mf: 0.5, msl: 1)



La matrice di confusione conferma ciò che viene mostrato nel report di classificazione la classe che corrisponde alla **cifra 1** mostra su circa **1135** campioni di **cifra 1** **1106** vengono classificati correttamente, analogamente la classe che corrisponde alla **cifra 8** è quella che ottiene i peggiori risultati infatti su **974** campioni disponibili solo **794** vengono classificati correttamente. Risaltano all'occhio **34** campioni di **cifra 8** classificati come **cifra 3**, **27** campioni di **cifra 8** classificata come **cifra 2** e **25** campioni di **cifra 8** classificati come **cifra 5**