



Fakultät für Informatik
Facoltà di Scienze e Tecnologie informatiche
Faculty of Computer Science

Machine Learning in Practice

Prof. Ivan Donatello

Report sul Progetto Finale – Fashion MNIST

2022/2023

Irene Avezzù, Ivana Nworah Bortot

17.06.2023

Indice

| | |
|---|----|
| Introduzione | 3 |
| Contesto | 3 |
| Obiettivo | 3 |
| Scelta del dataset | 3 |
| Panoramica sulle tecniche utilizzate | 4 |
| Dataset | 5 |
| Labels | 5 |
| Raccolta dei dati | 5 |
| Cleaning e preprocessing | 5 |
| Metodi | 7 |
| KNN | 7 |
| MLP | 7 |
| PREC-REC Curve | 7 |
| ROC Curve | 8 |
| Accuracy | 8 |
| F1 | 8 |
| Motivazione dietro la scelta dei metodi adottati | 8 |
| Esperimenti e risultati | 9 |
| Esperimenti | 9 |
| Risultati | 10 |
| Conclusione | 14 |

Introduzione

Contesto

Fashion-MNIST è un dataset di immagini che rappresentano articoli venduti sul sito di shopping online Zalando.

È un dataset formato da 60.000 istanze di training e 10.000 di test. Ognuna di queste istanze è caratterizzata da 785 features di cui 784 rappresentanti i pixel dell'immagine (28x28) e una per rappresentare l'etichetta, scelta tra 10 opzioni.

Il dataset Fashion-MNIST vuole essere una sostituzione diretta del dataset MNIST originale in modo da poterlo usare per il benchmarking degli algoritmi di apprendimento automatico.

Per benchmarking in economia s'intende una metodologia basata sul confronto di prodotti, servizi o processi aziendali che permette alle aziende che lo utilizzano di confrontarsi con le migliori imprese per poter apprendere da queste e migliorare.

Il dataset MNIST originale contiene molte cifre scritte a mano. I membri della comunità AI/ML/Data Science prediligono questo set di dati e lo usano come benchmark per convalidare i loro algoritmi. MNIST è infatti spesso il primo set di dati che i ricercatori provano affermando che "se non funziona su MNIST, non funzionerà affatto. ... e se funzionasse su MNIST, potrebbe comunque fallire su altri".

È possibile trovare ulteriori informazioni sul dataset presso il link seguente con la possibilità di poter scaricare i dati da file csv.

<https://www.kaggle.com/datasets/zalando-research/fashionmnist>

Obiettivo

L'obiettivo di questo progetto era quello di implementare e consolidare le varie tecniche e metodologie acquisite durante il corso *Machine Learning in Practice*. Utilizzando i principi appresi, il nostro obiettivo era quello di acquisire esperienza pratica e approfondire la nostra comprensione della materia.

Scelta del dataset

La scelta del dataset è ricaduta su Fashion-MNIST in particolar modo per 3 ragioni.

La prima è stata la dimensione del dataset. Il dataset analizzato infatti contiene un numero "limitato" di elementi (70.000 elementi), il che consente di avere tempi computazionali più efficienti per le classificazioni e il confronto tra dati.

La seconda ragione è il fatto che i dati risultano essere già vettorizzati in quanto ogni feature rappresenta il valore di un pixel dell'immagine.

La terza ragione ricade su un aspetto di pura preferenza personale tra le opzioni fornite dal professore.

Panoramica sulle tecniche utilizzate

Pre-processing dati:

- riconoscimento e sostituzione dei null-values
- rilevamento di outliers
- rimozione colonne non rilevanti

Classificazione:

- KNN classifier
 - Accuracy
 - F1-score
 - Prec-rec curve
 - Roc curve
- MLP classifier
 - Accuracy
 - F1-score
 - Prec-rec curve
 - Roc curve

Clustering:

- K-means
- Majority vote

Dataset

Il dataset Fashion MNIST è composto da 70.000 immagini, ognuna delle quali ha un'altezza di 28 pixel e una larghezza di 28 pixel e rappresentano 10 diverse tipologie di capi d'abbigliamento.

Il dataset è suddiviso in due subset, uno per il train e uno per il test dei dati. Il training dataset contiene 60.000 immagini, mentre il testing dataset contiene 10.000 immagini. Questa divisione consente di valutare le prestazioni dei modelli di machine learning su dati non visti in precedenza.

Il training dataset e il testing dataset hanno 785 colonne ciascuno. La prima colonna è costituita dalle label, che sono T-shirts/tops, Trousers, Pullovers, Dresses, Coats, Sandals, Shirts, Sneakers, Bags, e Ankle boots. Ciascuna label rappresenta la tipologia di capo d'abbigliamento. Le altre colonne contengono i valori dei pixel dell'immagine associata. A ogni pixel è associato un singolo valore, compreso tra 0 e 255, che indica la luminosità o l'oscurità di quel pixel. Un valore vicino a zero indica luminosità più intensa, valori vicino a 255 indicano che i pixel saranno più scuri.

Labels

A ogni esempio di training e test viene assegnata una delle seguenti labels:

| | |
|---|-------------|
| 0 | T-shirt/top |
| 1 | Trouser |
| 2 | Pullover |
| 3 | Dress |
| 4 | Coat |
| 5 | Sandal |
| 6 | Shirt |
| 7 | Sneaker |
| 8 | Bag |
| 9 | Ankle boot |

Raccolta dei dati

I dati nel dataset sono stati raccolti utilizzando immagini di capi d'abbigliamento in vendita sul sito Zalando. I dati sono stati raccolti con l'autorizzazione di Zalando, che è il proprietario e distributore del dataset, come specificato nella licenza del sito di Kaggle. Data questa origine dei dati, il dataset rappresenta realisticamente le tipologie di capi d'abbigliamento presenti sul sito.

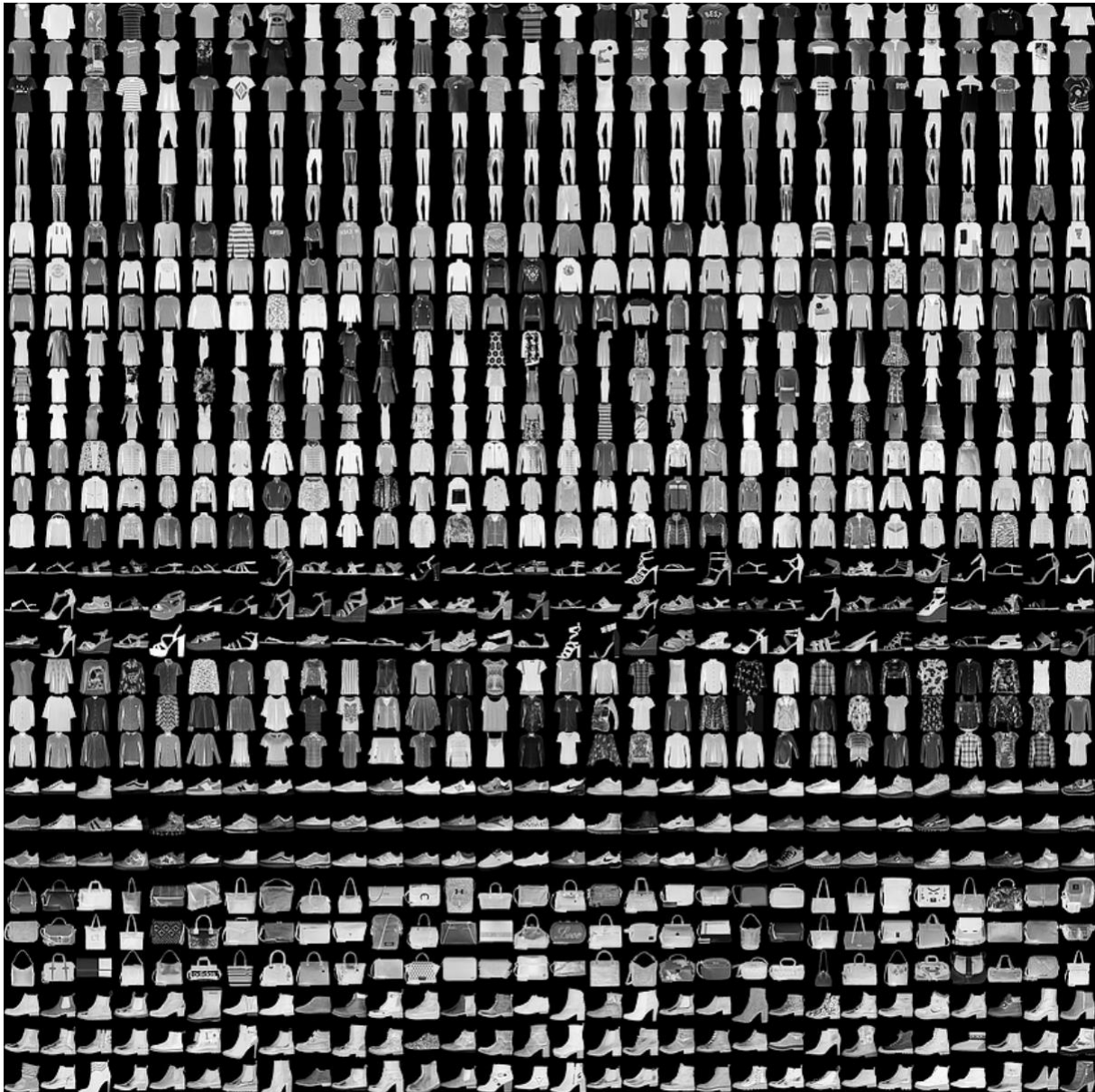
Cleaning e preprocessing

Nel nostro processo di analisi del dataset, non abbiamo applicato molte operazioni di cleaning e preprocessing poiché il dataset era già abbastanza strutturato. Non sono state riscontrate presenza di valori nulli o colonne non rilevanti che richiedessero azioni specifiche.

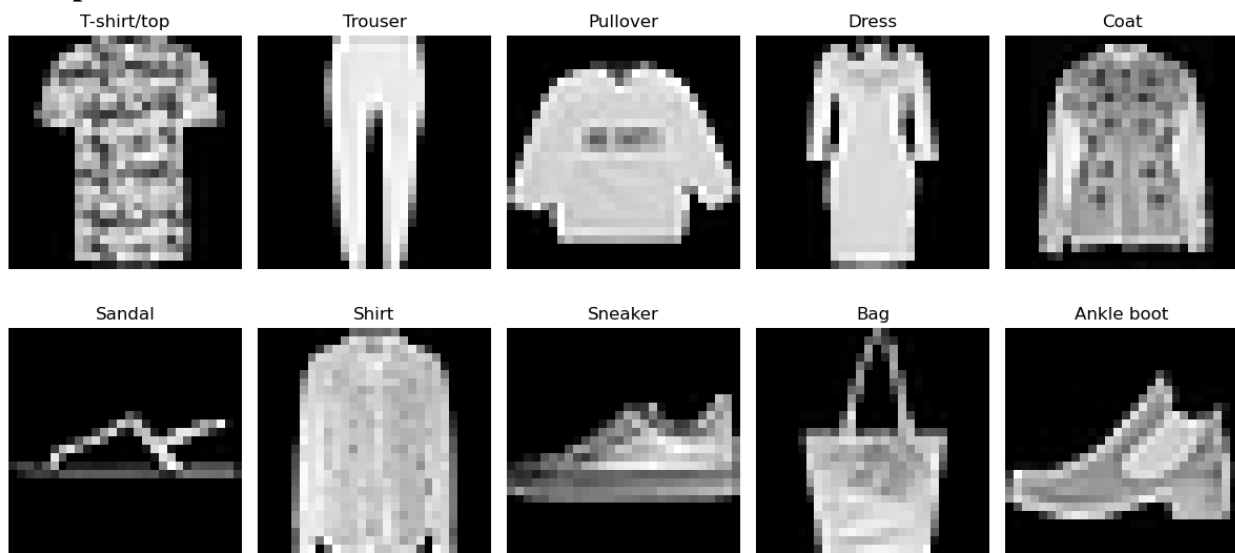
Inoltre, non sono stati individuati outliers nel dataset poiché i valori dei pixel delle immagini sono compresi tra 0 e 255, che rappresentano la scala di luminosità dei pixel.

Sia il training dataset che il testing dataset hanno classi bilanciate, con ogni classe che rappresenta approssimativamente il 10% del totale degli elementi. Ciò assicura che ogni classe sia adeguatamente rappresentata durante il processo di training e validation dei modelli di machine learning.

Esempio di immagini del dataset:



Esempio labellato:



Metodi

KNN

KNN (K-Nearest Neighbors) è un algoritmo di classificazione basato sull'idea che gli oggetti simili tendono ad essere nella stessa classe. L'algoritmo assegna una nuova istanza di test alla classe più frequente tra i suoi K vicini più prossimi nel set di training.

L'iperparametro principale è il numero di vicini K da considerare. Ottimizzare l'iperparametro K significa trovare il valore ottimale che massimizza le prestazioni del modello.

MLP

MLP (Multi-Layer Perceptron) è un algoritmo di deep learning che si basa su una rete neurale artificiale multi-layer per la classificazione dei dati. Utilizza molteplici layers di neuroni per apprendere le relazioni tra gli input e le labels delle classi.

Durante il processo di training, l'MLPClassifier ottimizza i pesi dei neuroni in base ai dati di training e alle labels delle classi corrispondenti. In seguito il modello può essere utilizzato per effettuare predizioni su nuovi dati, restituendo le labels di classe previste.

La flessibilità del MLPClassifier gli consente adattarsi a una vasta gamma di problemi di classificazione, apprendere modelli complessi e gestire dati non lineari.

PREC-REC Curve

La precision-recall curve è un grafico che mostra la relazione tra la precisione e il recall di un modello di classificazione al variare della soglia di decisione.

La precision misura la proporzione di istanze classificate come positive nel totale delle istanze predette come positive. Indica l'abilità del modello di classificare correttamente le istanze positive.

Il recall (anche conosciuto come true positive rate, sensitivity) rappresenta la proporzione di istanze positive correttamente identificate dal modello in proporzione al totale delle istanze effettivamente positive. Indica l'abilità del modello di distinguere tutte le istanze positive.

La prec-rec curve mostra come la precision e il recall si influenzano reciprocamente quando viene modificata la soglia di decisione del modello. In generale, all'aumentare della soglia di decisione, la precision tende ad aumentare mentre il richiamo diminuisce e viceversa. L'obiettivo è trovare un punto di equilibrio in cui la precision e il richiamo siano entrambi elevati.

ROC Curve

La curva ROC (Receiver Operating Characteristic) è un grafico che rappresenta la performance di un modello di classificazione binaria al variare della soglia di decisione. La curva ROC mostra il true positive rate al variare del false positive rate. Il true positive rate rappresenta la proporzione di istanze positive correttamente identificate dal modello, mentre il false positive rate rappresenta la proporzione di istanze negative erroneamente classificate come positive.

In un modello ottimale, il true positive rate è alto, e il false positive rate è basso. Questo si traduce in un'area sotto la curva ROC (Area Under the Curve, AUC) elevata. Un'AUC pari a 1 indica una performance perfetta.

Accuracy

L'accuracy è una misura della frequenza in cui il modello predice correttamente le labels delle classi. Viene calcolata dividendo il numero di predizioni corrette per il numero totale di predizioni. L'accuracy fornisce una visione generale della performance del modello, ma potrebbe non essere adatta nei casi in cui la distribuzione delle classi è sbilanciata.

F1

L'F1 è una media armonica tra precision e recall e viene calcolata assegnando uguale peso a entrambe le misure. L'F1 score varia tra 0 e 1, dove 1 rappresenta il punteggio migliore possibile.

L'F1 score è particolarmente utile quando il dataset ha distribuzioni di classe sbilanciate o quando sia la precision che il recall sono importanti per il problema in questione.

Motivazione dietro la scelta dei metodi adottati

Abbiamo scelto il classifier KNN poiché è un metodo non parametrico, il che significa che non fa alcuna ipotesi riguardo alla distribuzione dei dati o alla forma della decision boundary. Questa caratteristica può renderlo flessibile e adatto a dataset complessi o non lineari.

In questo progetto abbiamo utilizzato la tecnica del Grid Search per ottimizzare l'iperparametro K. La Grid Search ha esaminato diversi valori di K (1, 3, 5, 7, 9) e ha valutato le prestazioni del modello utilizzando il valore di F1 su un set di cross-validation. Il miglior punteggio F1 e il miglior iperparametro sono stati scelti come risultato del processo di ottimizzazione. Dopo aver eseguito la Grid Search, è stato selezionato l'iperparametro 7 come il valore migliore in base dell'accuracy e dell'F1 score.

MLP tende ad avere prestazioni computazionali migliori in fase di training rispetto a KNN, specialmente su grandi dati. Questo deriva dal fatto che il training di una rete neurale può essere parallelizzato su un hardware specializzato e può effettuare predizioni dirette sulla base dei pesi appresi, mentre KNN richiede la ricerca dei vicini, che richiede più tempo più grande è la mode di dati.

Esperimenti e risultati

Esperimenti

Durante l'analisi su questo dataset siamo partite dall'analizzare lo stato dei dati originali per poterne eseguire il pre-processing andando a rimuovere i null-values, gli outliers e le features non rilevanti ove necessario.

L'obiettivo della nostra analisi era quello di riconoscere quale classificatore fosse il più adatto e prestante in nel dataset analizzato. Tra i possibili classificatori visti durante il corso, è stato preferito utilizzare KNN, in quanto adatto al tipo di dati raccolti nel dataset. Inoltre è stato introdotto e utilizzato come confronto, il classificatore MLP, basato sul Deep Learning.

Sono state considerate diverse metriche per ogni classificatore analizzato al fine di ottenere una visione dettagliata delle performance dell'algoritmo di classificazione sul dataset. Tra le metriche considerate, sono state osservate l'accuracy, il punteggio F1, la curve prec-rec e la curva ROC.

L'Accuracy serve per poter avere una visione per quanto riguarda la frequenza con cui il modello predice correttamente la label.

Il punteggio F1 combinando con la media armonica precision e recall fornisce un valore pesato tra questi due. Risulta essere utile quando il set di dati ha distribuzioni di classi sbilanciate o quando sia la precisione che il richiamo sono importanti per il problema in questione.

La Precision indica la capacità del modello di classificare correttamente le istanze positive.

Il Recall, o sensibilità, indica la capacità del modello di catturare tutte le istanze positive.

La Prec-Rec curve consente di visualizzare il trade-off tra precision e recall quando si regola la soglia decisionale del modello.

La ROC curve fornisce una panoramica delle prestazioni del modello a diverse soglie di classificazione. Confronta il trade-off tra TPR e TFR. Permette di riconoscere la capacità del modello di separare le classi.

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN}$$

$$Precision = \frac{TP}{TP + FP}$$

$$Recall = \frac{TP}{TP + FN}$$

$$F1-score = \frac{2 \times Precision \times Recall}{Precision + Recall}$$

Un'altra metrica che è stata presa in considerazione è la AUC, ovvero l'area sotto la ROC curve, che permette di ottenere una sintesi delle metriche osservate. Più è elevata migliori saranno le possibilità di discriminare le classi.

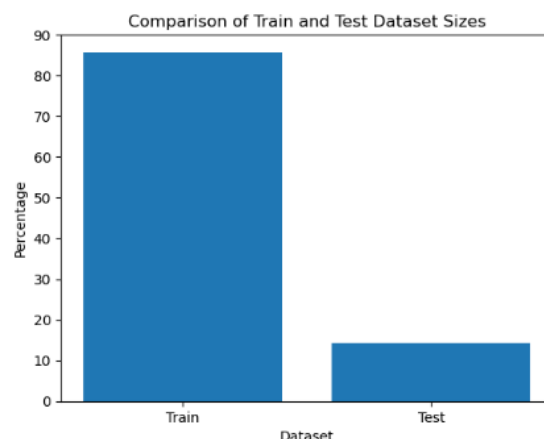
Tutte queste metriche sono state analizzate sui due classificatori selezionati (KNN, MLP).

Infine, è stato utilizzato il metodo del K-means per eseguire il clustering delle immagini, valutandone l'efficacia attraverso la misurazione del punteggio F1. Per determinare il numero ottimale di cluster da utilizzare, è stato utilizzato l'Elbow Method.

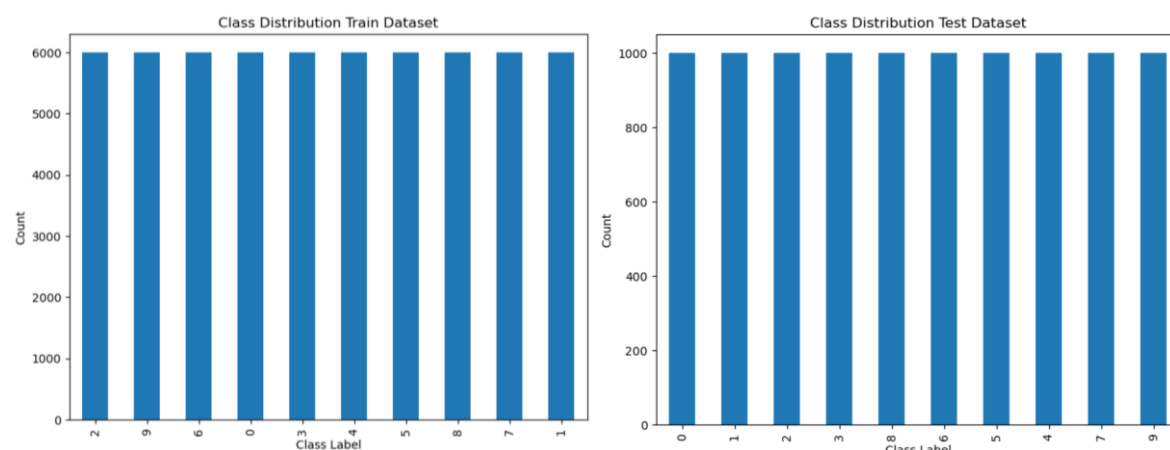
Risultati

In questa sezione daranno analizzati i risultati ottenuti in ordine sequenziale rispetto all'analisi descritta nel paragrafo degli esperimenti.

Durante l'analisi preliminare dei dati, è emerso che la suddivisione tra dati di training e di testing è del 85,71% per i dati di addestramento e del 14,29% per i dati di test. Queste proporzioni sono leggermente differenti da quello con cui abbiamo normalmente operato negli altri dataset (75% per train e 25% per test) ma mantengono comunque un buon rapporto.



È stato osservato che la distribuzione delle classi è perfettamente bilanciata sia nel training dataset che nel testing dataset. In entrambi i set, ci sono 10 classi che rappresentano il 10% dei dati (6000 nel training dataset e 1000 nel testing dataset).



L'analisi dei valori nulli e anomali ha portato alla realizzazione che tutte le istanze hanno valori rilevanti per ogni features e che non esistono outliers perché il valore di ogni pixel deve essere inteso nel range 0-255.

Sempre per analoghe ragioni non è stato necessario eseguire il drop di alcune colonne in quanto ogni pixel è necessario a rappresentare l'immagine.

Prima di iniziare con la vera e propria analisi dei classificatori è stato necessario eseguire l'ottimizzazione dell'iper-parametro di KNN con la quale abbiamo ottenuto le seguenti tabelle.

La prima tabella fornisce i risultati dettagliati della ricerca a griglia. Abbiamo osservato, per ogni possibile valore di K (1, 3, 5, 7, 9), il tempo medio e la standard deviation del fitting e dei punteggi dei test per tutte le possibili suddivisioni dei dati.

| | mean_fit_time | std_fit_time | mean_score_time | std_score_time | \ |
|---|---------------|--------------|-----------------|----------------|---|
| 0 | 0.078753 | 0.017589 | 9.429504 | 0.338285 | |
| 1 | 0.097185 | 0.018200 | 12.348335 | 1.174082 | |
| 2 | 0.089473 | 0.015392 | 18.859344 | 1.851536 | |
| 3 | 0.110058 | 0.019863 | 20.443566 | 0.898599 | |
| 4 | 0.115402 | 0.021074 | 21.354599 | 1.214340 | |

In sintesi, l'output include i migliori iperparametri trovati utilizzando la Grid Search, i relativi punteggi associati e una tabella che mostra i risultati delle diverse configurazioni di iperparametri testate durante la ricerca.

Machine Learning in Practice - Report sul Progetto Finale

Successivamente, sono stati osservati gli split score, che rappresentano i punteggi ottenuti durante la fase di cross-validation.

Durante la cross-validation, il dataset viene diviso in diverse parti chiamate "fold" e il modello viene addestrato e valutato su ciascuna fold in modo sequenziale.

Nel nostro caso abbiamo usato 5 fold (split0, split1, split2, split3, split4) per fare il training e il testing del modello. Successivamente, abbiamo valutato le prestazioni del modello utilizzando questa divisione dei dati.

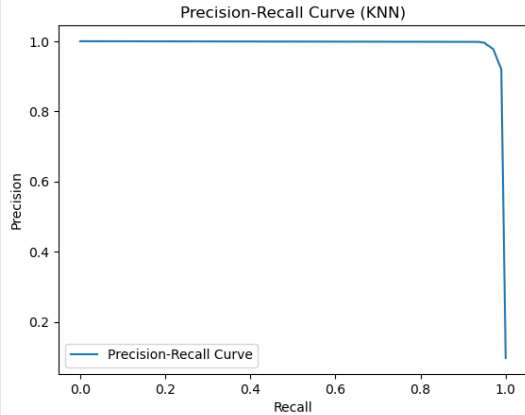
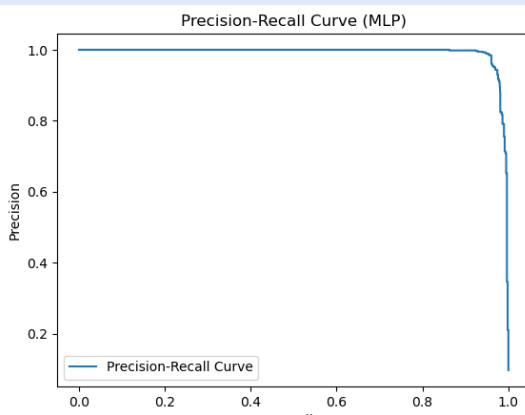
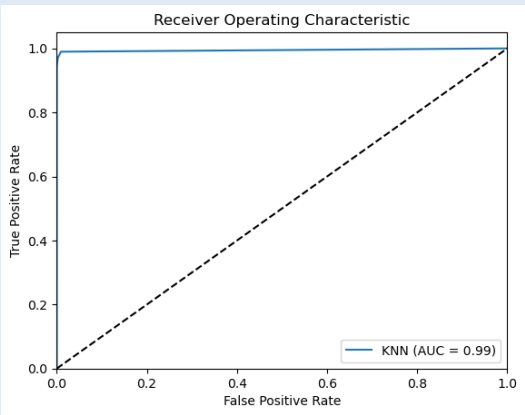
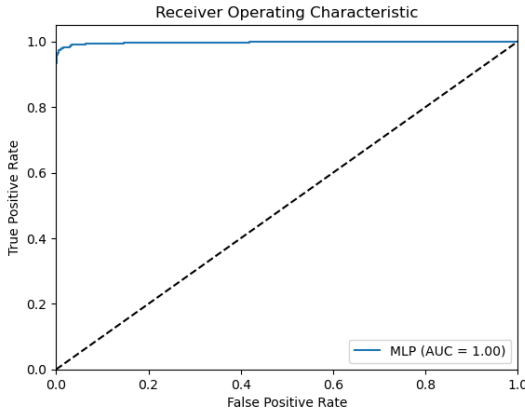
| | param_n_neighbors | params | split0_test_score | split1_test_score | \ |
|---|-------------------|--------------------|-------------------|-------------------|---|
| 0 | 1 | {'n_neighbors': 1} | 0.845521 | 0.848854 | |
| 1 | 3 | {'n_neighbors': 3} | 0.853229 | 0.853437 | |
| 2 | 5 | {'n_neighbors': 5} | 0.852292 | 0.857292 | |
| 3 | 7 | {'n_neighbors': 7} | 0.854167 | 0.857917 | |
| 4 | 9 | {'n_neighbors': 9} | 0.852812 | 0.853958 | |
| | split2_test_score | split3_test_score | split4_test_score | mean_test_score | \ |
| 0 | 0.845938 | 0.836771 | 0.839375 | 0.843292 | |
| 1 | 0.851979 | 0.843333 | 0.842500 | 0.848896 | |
| 2 | 0.852083 | 0.843750 | 0.843958 | 0.849875 | |
| 3 | 0.852708 | 0.842604 | 0.842812 | 0.850042 | |
| 4 | 0.848229 | 0.843021 | 0.841979 | 0.848000 | |
| | std_test_score | rank_test_score | | | |
| 0 | 0.004489 | 5 | | | |
| 1 | 0.004914 | 3 | | | |
| 2 | 0.005258 | 2 | | | |
| 3 | 0.006224 | 1 | | | |
| 4 | 0.004894 | 4 | | | |

In conclusione, si osserva che il miglior iper-parametro risulti essere $K = 7$ che permette di avere un best score di 0.850 e l'F1 Score di 0.856.

Seguendo le istruzioni della consegna non è stata eseguita un'analisi sulla possibile ottimizzazione dell'iper-parametro di MLP.

Machine Learning in Practice - Report sul Progetto Finale

Successivamente, è stata condotta un'analisi dei classificatori, come descritto nella sezione dedicata ai metodi, includendo KNN e MLP. Di seguito, sono riassunte le metriche di performance attraverso la tabella.

| | KNN (K = 7) | MLP |
|----------------|--|---|
| Accuracy | 0.849 | 0.858 |
| F1 score | 0.849 | 0.858 |
| Prec-rec curve |  |  |
| ROC curve |  |  |
| AUC | 0.973 | 0.985 |

Si osserva come i due classificatori abbiano prestazioni analoghe. Entrambi i classificatori mostrano punteggi F1 elevati, indicando un buon livello complessivo di precision e recall. Tuttavia, il classificatore MLP presenta un punteggio F1 leggermente superiore, suggerendo un leggero vantaggio nel trovare un equilibrio tra precision e recall rispetto al classificatore KNN.

Sia le curve precision-recall che le curve ROC dei classificatori KNN e MLP sono simili e hanno prestazioni ottimali. Entrambi i classificatori sono efficaci nel distinguere le diverse classi.

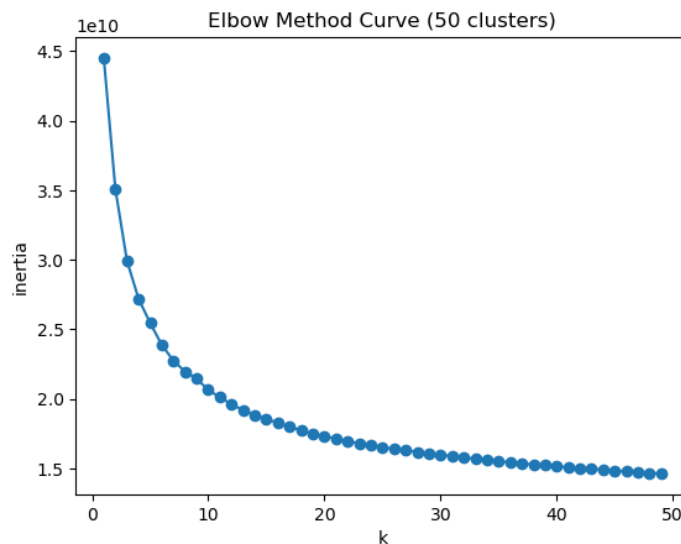
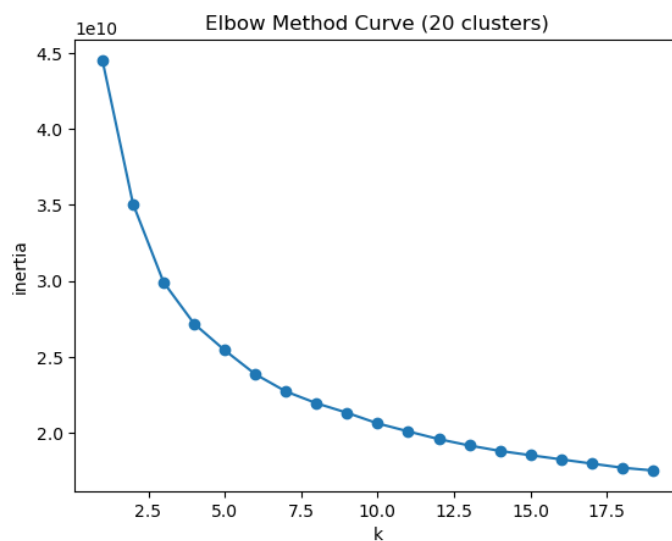
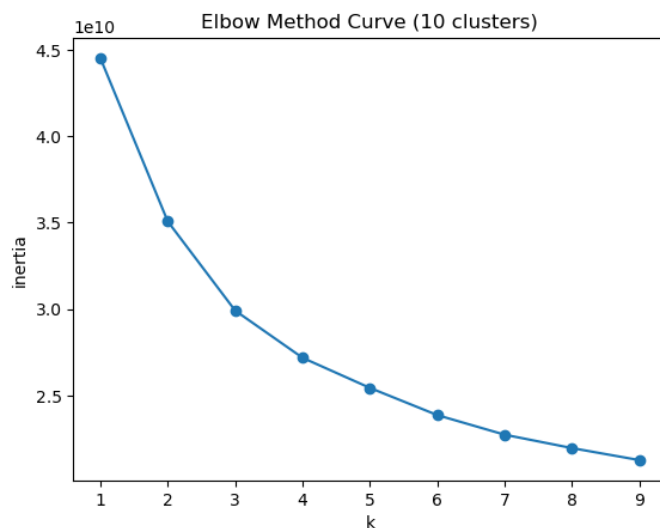
Si può inoltre notare che le classi sono leggermente più facilmente separabili utilizzando il classificatore MLP in quanto il valore dell'area al di sotto della curva ROC è più. Per entrambi i classificatori il valore di AUC è molto elevato (quasi = 1) e per questo si può dire che il modello ha raggiunto un alto grado di separazione tra le classi.

Un altro fattore importante da considerare nell'analisi dei due classificatori è la prestazione computazionale. In particolare, il classificatore KNN richiede significativamente più tempo per eseguire il fitting del modello rispetto al classificatore MLP.

Machine Learning in Practice - Report sul Progetto Finale

Per quanto concerne il K-Means clustering, l'Elbow Method non ha fornito una chiara indicazione del numero ottimale di cluster da utilizzare, nemmeno con valori elevati di K.

Nel caso in cui utilizziamo un numero di cluster pari al numero di classi, il punteggio F1 ottenuto risulta essere di 0.010, un valore molto basso. Tale punteggio suggerisce che l'algoritmo di clustering non riesce ad assegnare accuratamente i punti dati alle rispettive classi. Questo indica che l'approccio di clustering K-Means potrebbe non essere adatto per questo specifico compito di classificazione.



Conclusione

In questo progetto abbiamo analizzato il dataset Fashion MNIST e abbiamo implementato e consolidato le tecniche apprese durante il corso "Machine Learning in Practice". L'obiettivo principale era acquisire esperienza pratica e approfondire la comprensione del machine learning.

Durante l'analisi, abbiamo eseguito operazioni di pre-processing limitate poiché il dataset era già ben strutturato e non presentava valori nulli o colonne non rilevanti.

Per la classificazione dei dati, abbiamo utilizzato i classificatori KNN e MLP. KNN è un algoritmo non parametrico che si basa sulla similarità tra gli oggetti, mentre MLP è un algoritmo di deep learning basato su reti neurali artificiali. Abbiamo ottimizzato l'iperparametro K di KNN utilizzando la tecnica del Grid Search.

Abbiamo valutato diverse metriche per valutare le performance dei classificatori, tra cui accuracy, F1 score, precision-recall curve e ROC curve. L'accuracy fornisce una visione generale delle prestazioni del modello, mentre l'F1 score combina precision e recall per gestire dati sbilanciati. Le curve di precision-recall e ROC ci hanno permesso di valutare le prestazioni del modello a diverse soglie di classificazione.

I risultati mostrano che entrambi i classificatori hanno ottenuto buone prestazioni, ma MLP ha mostrato prestazioni computazionali migliori rispetto a KNN. L'accuratezza e l'F1 score sono risultate elevate per entrambi i classificatori, indicando una buona capacità di predizione delle labels delle classi.