Challenge 1: The banknote-authentication data set problem

Introduzione

I dati utilizzati nell'analisi provengono da:
https://archive.ics.uci.edu/dataset/267/banknote+authentication

Descrizione del set di dati

I dati sono stati estratti da immagini prese da esemplari di banconote autentiche e contraffatte. Per la digitalizzazione è stata utilizzata una fotocamera industriale solitamente impiegata per l'ispezione delle stampe. Le immagini finali hanno una risoluzione di 400x400 pixel. A causa della lente dell'oggetto e della distanza dall'oggetto in esame, sono state ottenute immagini in scala di grigi con una risoluzione di circa 660 dpi. Per estrarre le caratteristiche dalle immagini è stato utilizzato lo strumento della trasformata Wavelet.
Queste caratteristiche sono:

- varianza dell'immagine trasformata Wavelet (continua)
- skewness dell'immagine trasformata Wavelet (continua)
- curtosi dell'immagine trasformata Wavelet (continua)
- entropia dell'immagine (continua)
- classe (intero)

Data pretreatment

Dopo aver caricato il dataset e aver visualizzato le colonne e i tipi delle variabili, ci assicuriamo che non ci siano valori mancanti e visualizziamo le statistiche di base per vedere se ci sono valori anomali, appuriamo che non ci sono e quindi procediamo nell'analisi.

I dati sono ordinati per categoria, per gli algoritmi di Unsupervised learning che utilizzeremo questo non influisce sui risultati, mentre per quelli di Supervised learning applicheremo una randomizzazione nella selezione dei dati per il train set e per il test set al fine di avere una performance del modello unbiased, robusta e generalizzabile.

Unsupervised Learning

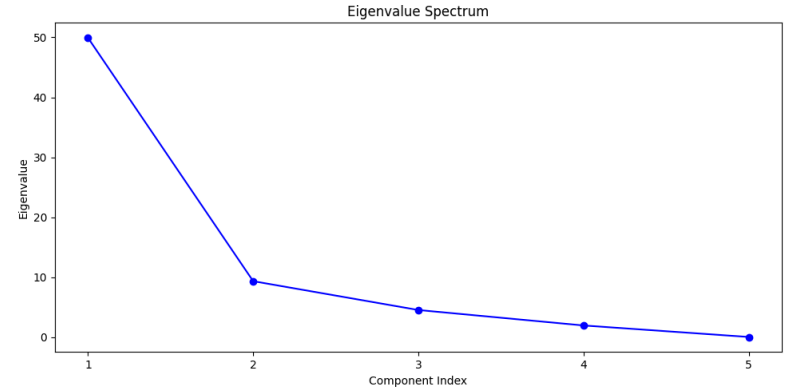
PCA

Come prima cosa calcoliamo gli autovalori della matrice di covarianza dei nostri dati, il risultato ottenuto è il seguente:

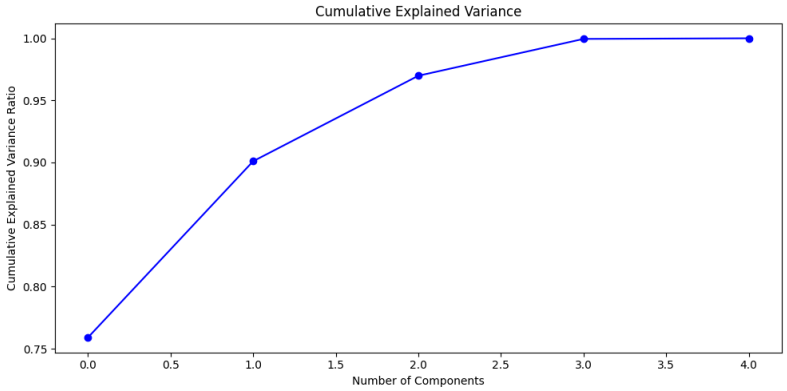
[4.99182748e+01 9.33596699e+00 4.52722782e+00 1.95106291e+00 3.22035275e-02]

Maggiore è l'autovalore, maggiore sarà la varianza spiegata dalla componente principale corrispondente.
Calcoliamo la varianza spiegata di ogni componente e vediamo che le prime due componenti sono già abbastanza per spiegare il 90% della varianza dei dati.

Cumulative Explained Variance Ratio:
[0.75904319 0.90100326 0.96984301 0.99951032 1.]

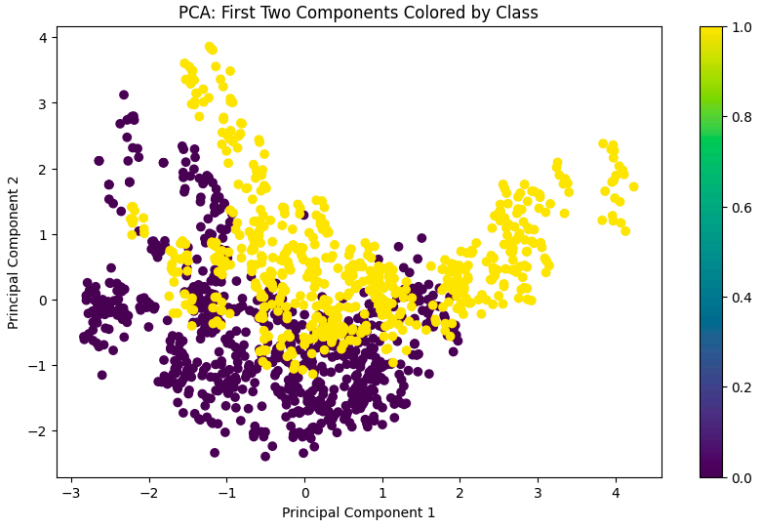


Dal grafico a sinistra (scree plot) vediamo che al valore due c'è un gomito, questo conferma quanto detto prima, il valore di componenti principali da utilizzare è due.



Il grafico a sinistra è una visualizzazione della varianza spiegata calcolata in precedenza.

Applichiamo quindi la PCA con numero di componenti principali pari a due, il grafico ottenuto è il seguente.

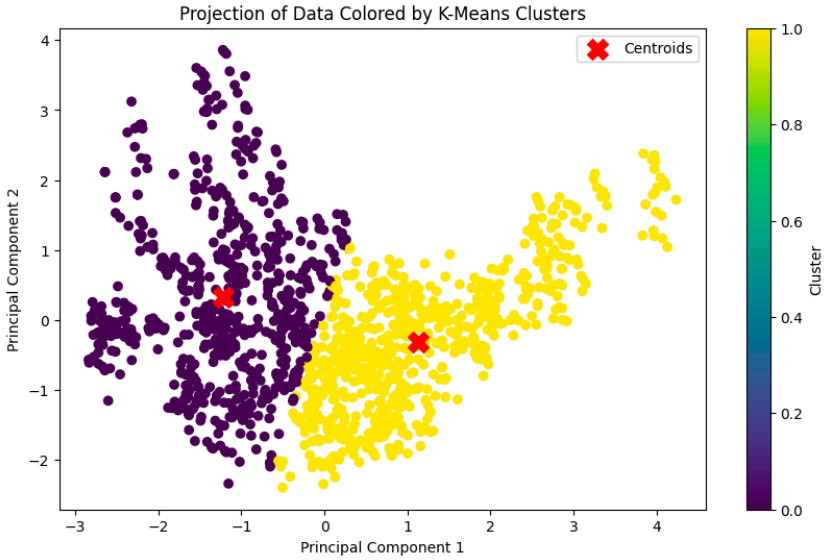


Le classi sembrano essere linearmente separabili nella proiezione basata sul grafico PCA.

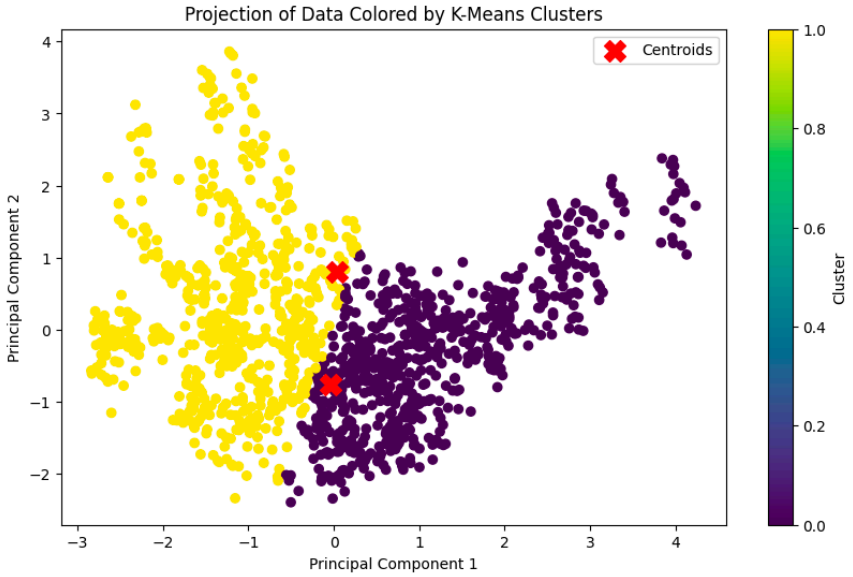
Sembra esserci un raggruppamento e separazione tra le diverse classi nello spazio bidimensionale ridotto. Tuttavia, la separazione potrebbe non essere perfettamente lineare, indicando che potrebbe esserci una sovrapposizione tra le classi.

K-means

Applicando K-means solo con le prime due componenti principali otteniamo il seguente grafico:



I dati sembrano dividersi in due cluster in modo netto.



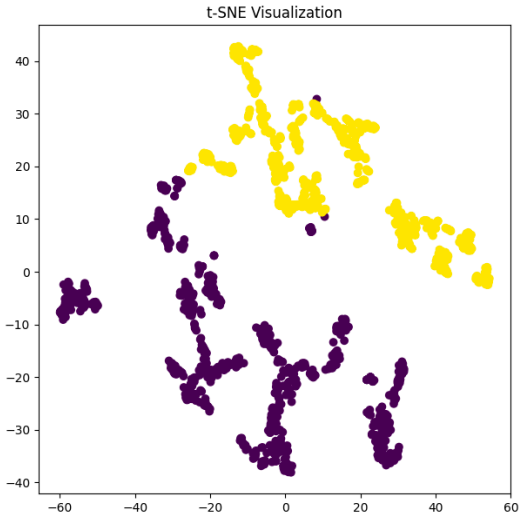
Applicando k-means con tutte le coordinate otteniamo il seguente grafico:

Anche in questo caso i dati sono divisi in maniera abbastanza netta nei due cluster, la posizione dei centroidi è differente rispetto a prima, più vicina ai confini della divisione tra i due cluster, possiamo dire che sia sufficiente applicare l'algoritmo alla due componenti principali, viste le minime differenze tra i due grafici.

t-SNE

Applicando t-SNE otteniamo il seguente grafico:

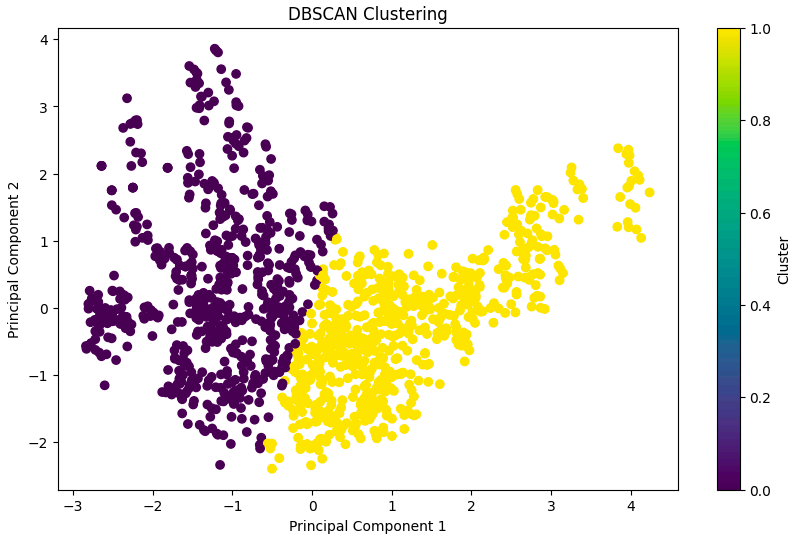
I dati sono colorati in base alle ground truth labels e sembrano essere divisi in due cluster ben definiti, lo evinciamo dal fatto che i punti di colori diversi non si mescolano e sono abbastanza



distanti nel grafico.
Rispetto alla riduzione dimensionale fatta con PCA otteniamo una divisione più evidente dei dati nei due cluster.

DBSCAN

Applicando l'algoritmo DBSCAN otteniamo i seguenti risultati:



Anche da questa analisi emerge una divisione netta dei dati in due cluster.

Supervised Learning

Logistic regression

Applicando la Logistic regression si ottengono i seguenti risultati per accuracy, precision, recall e F1-score:
- Accuracy: 0.9865591397849462
- Precision: 0.9817073170731707
- Recall: 0.9877300613496932
- F1-score: 0.9847094801223242

Hanno tutte valori molto vicini a uno, indicando una prestazione molto buona dell'algoritmo.

Decision tree

Applicando l'algoritmo Decision tree si ottengono i seguenti risultati per accuracy, precision, recall e F1-score:
- Accuracy: 0.9865591397849462
- Precision: 0.99375
- Recall: 0.9754601226993865
- F1-score: 0.9845201238390093

Anche in questo caso hanno tutte valori molto vicini a uno, indicando una prestazione molto buona dell'algoritmo.

Naive Bayes

Applicando l'algoritmo Naive Bayes si ottengono i seguenti risultati per accuracy, precision, recall e F1-score:
- Accuracy: 0.8360215053763441
- Precision: 0.8642857142857143
- Recall: 0.7423312883435583
- F1-score: 0.7986798679867987

In questo caso i valori ottenuti sono leggermente peggiori rispetto a quelli generati dagli algoritmi precedenti, tuttavia possono comunque essere considerati valori buoni.

Ridge Regression

Con la Ridge Regression valutiamo l'effetto della regolarizzazione sulla Logistic Regression, il parametro di regolarizzazione è stato calcolato con la cross-validation.
I risultati ottenuti sono i seguenti:
- Accuracy with Best Regularization Parameter (C=1): 0.99
- Precision with Best Regularization Parameter (C=1): 0.98
- Recall with Best Regularization Parameter (C=1): 0.99
- F1-score with Best Regularization Parameter (C=1): 0.98

Anche in questo caso la performance del modello è vicina all'ottimo, la regolarizzazione non sembra avere effetti significativi sui risultati.

LASSO Regression

Con la LASSO Regression valutiamo l'effetto della regolarizzazione sulla Logistic Regression, il parametro di regolarizzazione è stato calcolato con la cross-validation.
I risultati ottenuti sono i seguenti:
- Accuracy with Best Regularization Parameter (C=100): 0.99
- Precision with Best Regularization Parameter (C=100): 0.99
- Recall with Best Regularization Parameter (C=100): 0.99
- F1-score with Best Regularization Parameter (C=100): 0.99

Come osservato anche con la precedente regolarizzazione, non sembrano esserci significativi cambiamenti nella performance del nostro modello.

Elastic Nets

Con l'algoritmo Elastic Nets valutiamo l'effetto della regolarizzazione sulla Logistic Regression, il parametro di regolarizzazione è stato calcolato con la cross-validation.
I risultati ottenuti sono i seguenti:
- Accuracy with Best Regularization Parameters (C=1, l1_ratio=0.1): 0.99
- Precision with Best Regularization Parameters (C=1, l1_ratio=0.1): 0.98
- Recall with Best Regularization Parameters (C=1, l1_ratio=0.1): 0.99
- F1-score with Best Regularization Parameters (C=1, l1_ratio=0.1): 0.98

Anche in questo caso la performance rimane molto buona, con statistiche molto vicine a uno, non cambiando molto dai primi risultati ottenuti.

Conclusione

In conclusione possiamo dire che tutti gli algoritmi applicati, sia Unsupervised che Supervised, sembrano performare in maniera eccellente, i risultati di entrambe le analisi sembrano essere coerenti tra loro. Detto ciò, risultati così vicini alla perfezione potrebbero significare che stiamo facendo overfitting, per questo potrebbe essere utile verificare i risultati procedendo nell'analisi con la raccolta di nuovi dati (sempre se possibile) o magari usando tecniche di validazione incrociata come la k-fold cross-validation, visto che abbiamo già utilizzato tecniche per la riduzione della complessità del modello e tecniche di regolarizzazione.