

Machine Learning in Practice

Prof. Ivan Donatello

Final Project Report – Fashion MNIST

2022/2023

Irene Avezzù, Ivana Nworah Bortot

17.06.2023

Indice

Introduction	3
Context	3
Obiettivo	3
Choice of dataset.....	3
Overview of techniques used	4
Dataset	5
Labels.....	5
Data collection.....	5
Cleaning and preprocessing.....	5
Methods	7
KNN	7
MLP	7
PREC-REC Curve.....	7
ROC Curve.....	8
Accuracy	8
F1.....	8
Motivation behind the choice of methods.....	8
Experiments and results.....	9
Experiments	9
Results.....	10
Conclusion	14

Introduction

Context

Fashion-MNIST is a dataset of images representing items sold on the online shopping site Zalando.

It is a dataset consisting of 60,000 training instances and 10,000 test instances. Each of these instances has 785 features of which 784 represent the image pixels (28x28) and one to represent the label, chosen from 10 options.

The Fashion-MNIST dataset is intended to be a direct replacement of the original MNIST dataset so that it can be used for benchmarking machine learning algorithms.

Benchmarking in economics refers to a methodology based on the comparison of products, services or business processes that allows companies using it to compare themselves with the best companies in order to learn from them and improve.

The original MNIST dataset contains many handwritten figures. Members of the AI/ML/Data Science community favour this dataset and use it as a benchmark to validate their algorithms. In fact, MNIST is often the first dataset that researchers test by claiming that "if it doesn't work on MNIST, it won't work at all. ... and if it does work on MNIST, it might still fail on others'.

You can find more information on the dataset at the following link with the possibility of downloading the data from csv files.

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

Obiettivo

The aim of this project was to implement and consolidate the various techniques and methodologies acquired during the *Machine Learning in Practice* course. Using the principles learnt, our aim was to gain practical experience and deepen our understanding of the subject.

Choice of dataset

The choice of dataset fell on Fashion-MNIST in particular for three reasons.

The first was the size of the dataset. In fact, the analysed dataset contains a 'limited' number of elements (70,000 elements), which allows for more efficient computing times for classifications and data comparison.

The second reason is the fact that the data is already vectorized as each feature represents the value of an image pixel.

The third reason falls on an aspect of pure personal preference among the options provided by the professor.

Overview of techniques used

Data pre-processing

- recognition and replacement of null-values
- detection of outliers
- removal of irrelevant columns

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

The Fashion MNIST dataset consists of 70,000 images, each of which has a height of 28 pixels and a width of 28 pixels and represents 10 different types of clothing.

The dataset is divided into two subsets, one for training and one for testing the data. The training dataset contains 60,000 images, while the testing dataset contains 10,000 images. This division makes it possible to evaluate the performance of machine learning models on previously unseen data.

The training dataset and the testing dataset each have 785 columns. The first column consists of the labels, which are T-shirts/tops, Trousers, Pullovers, Dresses, Coats, Sandals, Shirts, Sneakers, Bags, and Ankle boots. Each label represents the type of garment. The other columns contain the pixel values of the associated image. Each pixel is associated with a single value, between 0 and 255, indicating the brightness or darkness of that pixel. A value near zero indicates brighter brightness, values near 255 indicate that the pixels will be darker.

Labels

Each training and test example is assigned one of the following labels:

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

Data collection

The data in the dataset was collected using images of clothes for sale on the Zalando website. The data was collected with the permission of Zalando, which is the owner and distributor of the dataset, as specified in Kaggle's site licence. Given this data source, the dataset realistically represents the types of garments on the site.

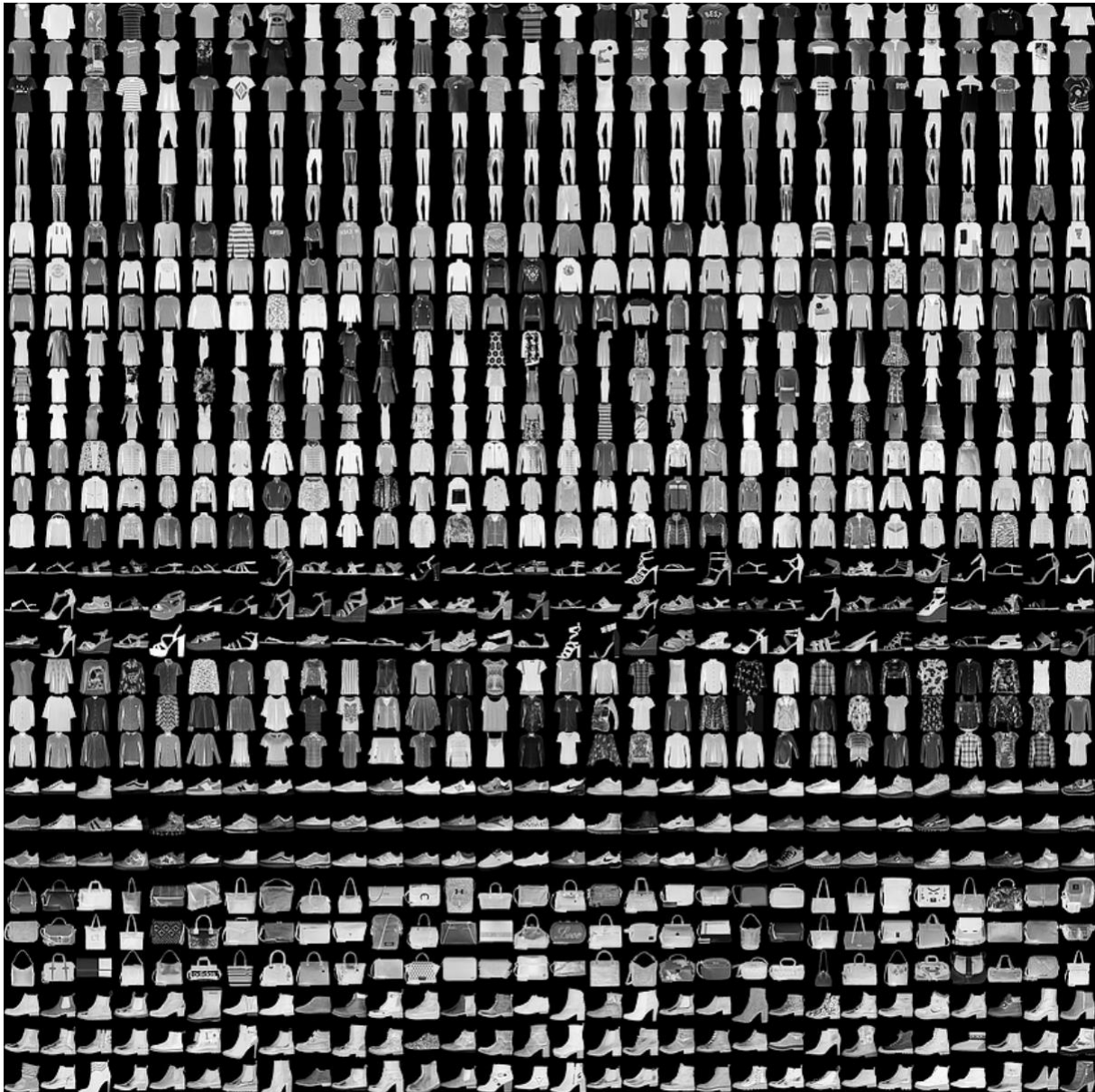
Cleaning and preprocessing

In our process of analysing the dataset, we did not apply many cleaning and preprocessing operations as the dataset was already quite structured. No null values or irrelevant columns requiring specific actions were found.

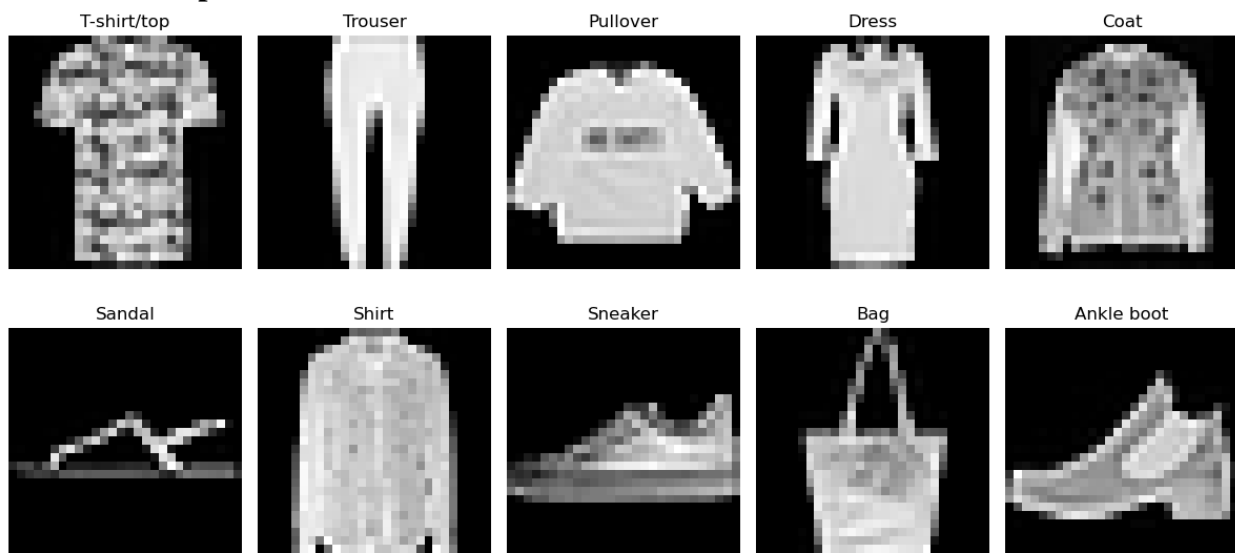
In addition, no outliers were found in the dataset as the pixel values of the images are between 0 and 255, representing the pixel brightness scale.

Both the training dataset and the testing dataset have balanced classes, with each class representing approximately 10 per cent of the total elements. This ensures that each class is adequately represented during the training and validation process of the machine learning models.

Example of dataset images:



Labelled example:



Methods

KNN

KNN (K-Nearest Neighbours) is a classification algorithm based on the idea that similar objects tend to be in the same class. The algorithm assigns a new test instance to the most frequent class among its K nearest neighbours in the training set.

The main hyperparameter is the number of K neighbours to be considered. Optimising the hyperparameter K means finding the optimal value that maximises the performance of the model.

MLP

MLP (Multi-Layer Perceptron) is a deep learning algorithm that relies on a multi-layer artificial neural network to classify data. It uses multiple layers of neurons to learn the relationships between inputs and class labels.

During the training process, the MLPClassifier optimises the weights of the neurons based on the training data and the corresponding class labels. Subsequently, the model can be used to make predictions on new data, returning the predicted class labels.

The flexibility of the MLPClassifier allows it to adapt to a wide range of classification problems, learn complex models and handle non-linear data.

PREC-REC Curve

The precision-recall curve is a graph showing the relationship between the precision and recall of a classification model as the decision threshold changes.

Precision measures the proportion of instances classified as positive in the total number of instances predicted as positive. It indicates the ability of the model to correctly classify positive instances.

Recall (also known as true positive rate, sensitivity) represents the proportion of positive instances correctly identified by the model as a proportion of the total number of instances that are actually positive. It indicates the ability of the model to distinguish all positive instances.

The prec-rec curve shows how precision and recall influence each other when the model's decision threshold is changed. In general, as the decision threshold increases, precision tends to increase while recall decreases and vice versa. The goal is to find an equilibrium point where precision and recall are both high.

ROC Curve

The ROC (Receiver Operating Characteristic) curve is a graph representing the performance of a binary classification model as the decision threshold changes. The ROC curve shows the true positive rate as the false positive rate varies. The true positive rate represents the proportion of positive instances correctly identified by the model, while the false positive rate represents the proportion of negative instances incorrectly classified as positive.

In an optimal model, the true positive rate is high, and the false positive rate is low. This results in a high Area Under the Curve (AUC). An AUC of 1 indicates a perfect performance.

Accuracy

Accuracy is a measure of how often the model correctly predicts class labels. It is calculated by dividing the number of correct predictions by the total number of predictions. Accuracy provides an overview of the model's performance, but may not be suitable in cases where the distribution of classes is unbalanced.

F1

F1 is a harmonic mean between precision and recall and is calculated by assigning equal weight to both measures. The F1 score varies between 0 and 1, where 1 represents the best possible score.

The F1 score is particularly useful when the dataset has unbalanced class distributions or when both precision and recall are important for the problem at hand.

Motivation behind the choice of methods

We chose the KNN classifier because it is a non-parametric method, which means that it makes no assumptions about the distribution of the data or the shape of the decision boundary. This characteristic can make it flexible and suitable for complex or non-linear datasets.

In this project, we used the Grid Search technique to optimise the hyperparameter K. Grid Search examined different values of K (1, 3, 5, 7, 9) and evaluated the performance of the model using the value of F1 on a cross-validation set. The best F1 score and the best hyperparameter were chosen as a result of the optimisation process. After performing Grid Search, hyperparameter 7 was selected as the best value based on accuracy and F1 score.

MLP tends to have better computational performances. This stems from the fact that the training of a neural network can be parallelised on specialised hardware and can make direct predictions on the basis of the learnt weights, whereas KNN requires neighbour search, which takes longer the larger the data mode.

Experiments and results

Experiments

During the analysis of this dataset, we started by analysing the state of the original data in order to pre-process it by removing null-values, outliers and irrelevant features where necessary.

The objective of our analysis was to recognize which classifier was the most suitable and performing in the analysed dataset. Of the possible classifiers seen during the course, KNN was preferred, as it was suitable for the type of data collected in the dataset. In addition, the MLP classifier, based on Deep Learning, was introduced and used as a comparison.

Several metrics were considered for each analysed classifier in order to obtain a detailed view of the performance of the classification algorithm on the dataset. Among the metrics considered, the accuracy, F1 score, prec-rec curve and ROC curve were observed.

Accuracy is used to gain an insight into how often the model correctly predicts the label.

The F1 score combining with the harmonic mean precision and recall provides a weighted value between these two. It is useful when the data set has unbalanced class distributions or when both precision and recall are important for the problem at hand.

Precision indicates the model's ability to correctly classify positive instances.

Recall, or sensitivity, indicates the model's ability to capture all positive instances.

The Prec-Rec curve visualises the trade-off between precision and recall when adjusting the model's decision threshold.

The ROC curve provides an overview of the model's performance at different classification thresholds. It compares the trade-off between TPR and TFR. It allows one to recognise the model's ability to separate classes.

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

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

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

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

Another metric that was taken into account is the AUC, i.e. the area under the ROC curve, which provides a summary of the observed metrics. The higher it is, the better the chances of discriminating between classes.

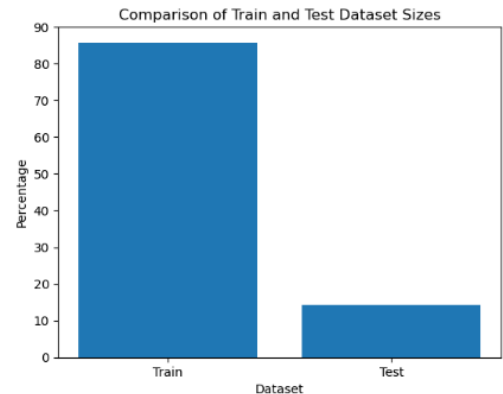
All these metrics were analysed on the two selected classifiers (KNN, MLP).

Finally, the K-means method was used to perform image clustering, evaluating its effectiveness by measuring the F1 score. To determine the optimal number of clusters to be used, the Elbow Method was used.

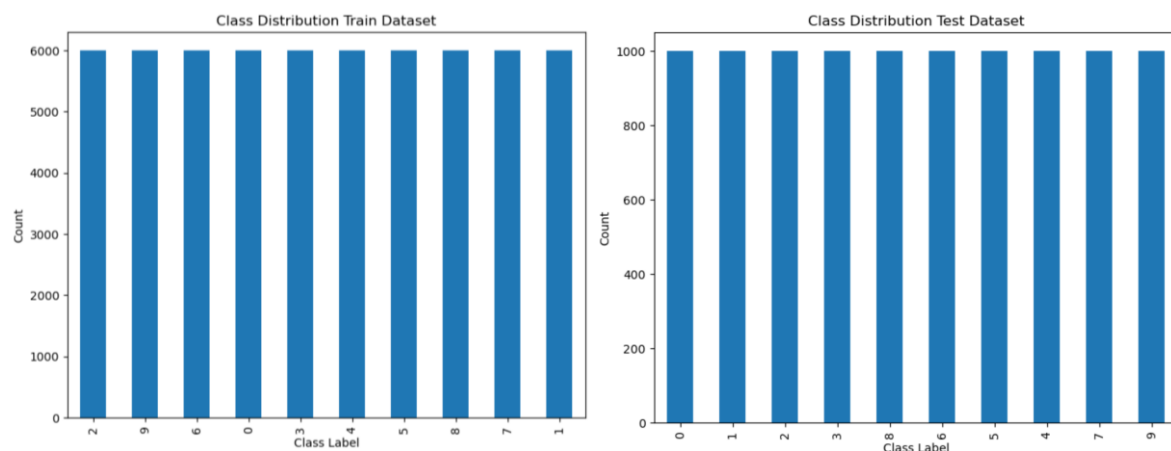
Results

This section will analyse the results obtained in sequential order with respect to the analysis described in the section on experiments.

During the preliminary analysis of the data, it emerged that the split between training and testing data is 85.71% for training data and 14.29% for testing data. These proportions are slightly different from what we normally work with in the other datasets (75% for training and 25% for testing) but still maintain a good ratio.



It was observed that the distribution of classes is perfectly balanced in both the training dataset and the testing dataset. In both sets, there are 10 classes representing 10% of the data (6000 in the training dataset and 1000 in the testing dataset).



The analysis of null and outlier values led to the realisation that all instances have relevant values for each feature and that there are no outliers because the value of each pixel must be understood to be in the range 0-255.

Also for similar reasons, it was not necessary to drop some columns as each pixel is necessary to represent the image.

Before starting with the actual analysis of the classifiers, it was necessary to perform the KNN hyperparameter optimisation with which we obtained the following tables.

The first table provides the detailed results of the grid search. We observed, for each possible value of K (1, 3, 5, 7, 9), the mean time and standard deviation of the fitting and test scores for all possible subdivisions of the data.

	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 summary, the output includes the best hyperparameters found using Grid Search, their associated scores and a table showing the results of the different hyperparameter configurations tested during the search. Successivamente, sono stati osservati gli split score, che rappresentano i punteggi ottenuti durante la fase di cross-validation.

Machine Learning in Practice - Report sul Progetto Finale

During cross-validation, the dataset is split into several parts called folds and the model is trained and evaluated on each fold sequentially.

In our case, we used 5 folds (split0, split1, split2, split3, split4) to train and test the model. Subsequently, we evaluated the performance of the model using this split.

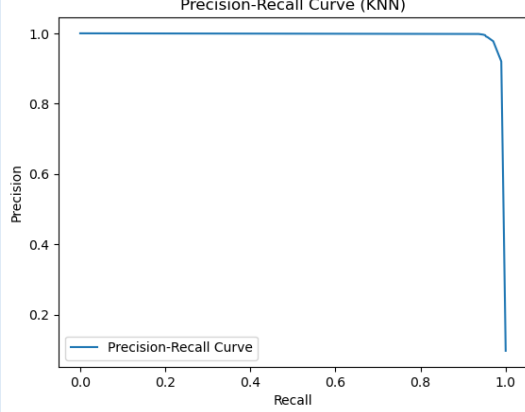
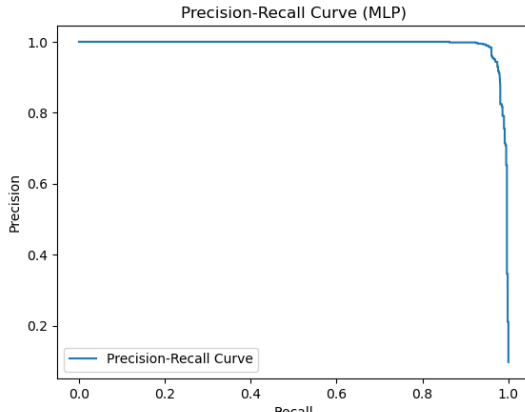
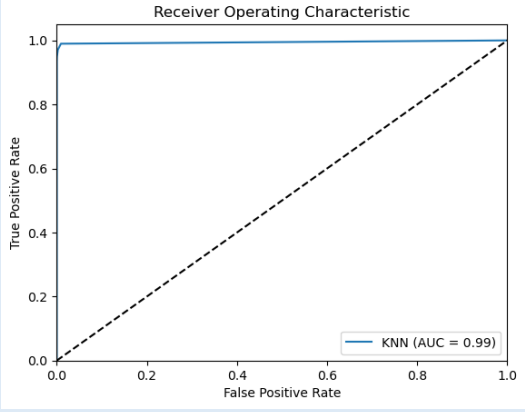
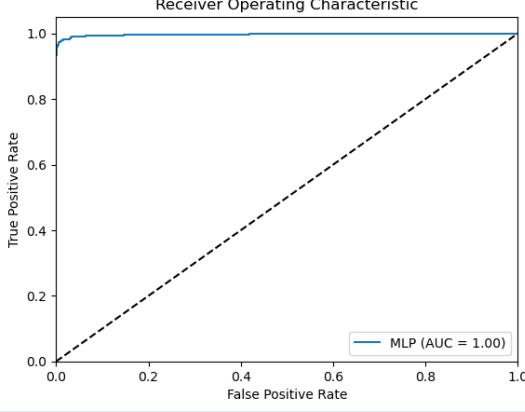
	param_n_neighbors	params	split0_test_score	split1_test_score	\
0	1	{'n_neighbors': 1}	0.845521	0.848854	
1	3	{'n_neighbors': 3}	0.853229	0.853437	
2	5	{'n_neighbors': 5}	0.852292	0.857292	
3	7	{'n_neighbors': 7}	0.854167	0.857917	
4	9	{'n_neighbors': 9}	0.852812	0.853958	
	split2_test_score	split3_test_score	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 conclusion, it is observed that the best hyper-parameter turns out to be $K = 7$ which allows for a best score of 0.850 and the F1 Score of 0.856.

Following the delivery instructions, no analysis was performed on the possible optimisation of the MLP hyper-parameter.

Machine Learning in Practice - Report sul Progetto Finale

Next, an analysis of the classifiers was conducted, as described in the methods section, including KNN and MLP. Below, the performance metrics are summarised in the table.

	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

It can be seen that the two classifiers perform similarly. Both classifiers show high F1 scores, indicating a good overall level of precision and recall. However, the MLP classifier shows a slightly higher F1 score, suggesting a slight advantage in striking a balance between precision and recall compared to the KNN classifier.

Both the precision-recall and ROC curves of the KNN and MLP classifiers are similar and perform well. Both classifiers are effective in distinguishing different classes.

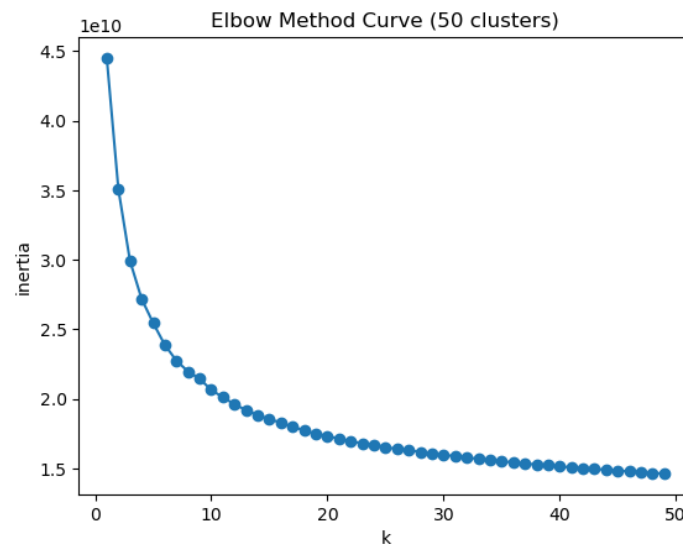
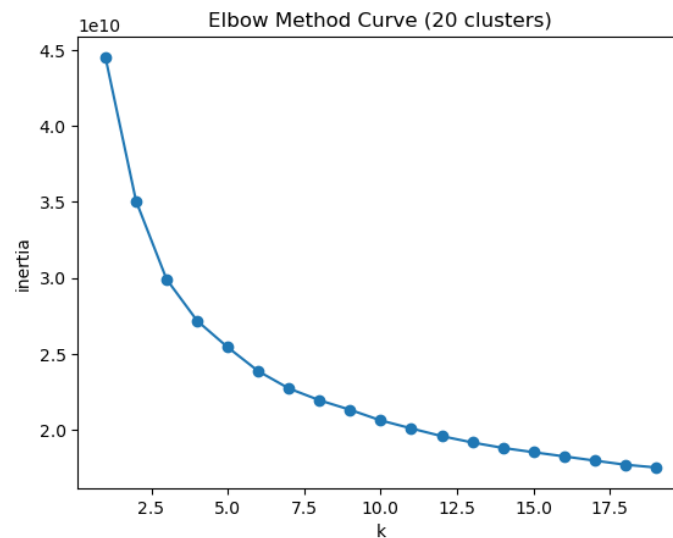
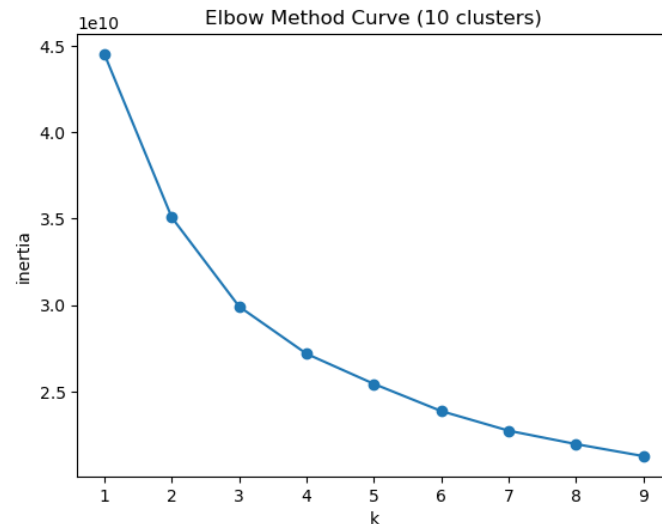
It can also be seen that the classes are slightly easier to separate using the MLP classifier as the value of the area under the ROC curve is more. For both classifiers, the value of the AUC is very high (almost = 1), so it can be said that the model has achieved a high degree of separation between the classes.

Another important factor to consider when analysing the two classifiers is the computational performance. In particular, the KNN classifier requires significantly more time to perform the model fitting than the MLP classifier.

Machine Learning in Practice - Report sul Progetto Finale

With regard to K-Means clustering, the Elbow Method did not provide a clear indication of the optimal number of clusters to use, even with high values of K.

When using a number of clusters equal to the number of classes, the F1 score obtained was 0.010, a very low value. This score suggests that the clustering algorithm fails to accurately assign the data points to the respective classes. This indicates that the K-Means clustering approach may not be suitable for this specific classification task.



Conclusion

In this project, we analysed the Fashion MNIST dataset and implemented and consolidated the techniques learnt during the 'Machine Learning in Practice' course. The main goal was to gain practical experience and deepen our understanding of machine learning.

During the analysis, we performed limited pre-processing operations as the dataset was already well structured and had no null values or irrelevant columns.

For the classification of the data, we used the KNN and MLP classifiers. KNN is a non-parametric algorithm based on similarity between objects, whereas MLP is a deep learning algorithm based on artificial neural networks. We optimised the K-parameter of KNN using the Grid Search technique.

We evaluated several metrics to assess the performance of the classifiers, including accuracy, F1 score, precision-recall curve and ROC curve. The accuracy provides an overview of the model's performance, while the F1 score combines precision and recall to handle unbalanced data. The precision-recall and ROC curves allowed us to evaluate the performance of the model at different classification thresholds.

The results show that both classifiers performed well, but MLP showed better computational performance than KNN. The accuracy and F1 score were high for both classifiers, indicating a good prediction capability of the class labels.