Machine Learning and Artificial Intelligence 2023/24

Computer Engineering for Robotics and Smart Industry

Fabio Forlin, VR508909

Contents

1 Introduction

The goal of the exam is a project structured as follows: choose a dataset in the net, reduce the dimensionality, extract features and evaluate the results with some metrics. I will list some techniques in this report.

1.1 Dataset choice

One of the most crucial parts was the choice of the dataset. My idea was to find a dataset with defined clusters, without too many images and with training and validation samples. The choice fell on a dataset of playing card images[1], found in kaggle, a data science platform that enables users to find and publish datasets. All images are 224 X 224 in jpg format. There are *7624 training images, 265 validation images and 265 test images.* The train, test and validation directories are partitioned into 53 sub directories, one for each of the 53 types of cards.

Figure 1: An example of a training image (joker)

2 Dimensionality Reduction and extracting features

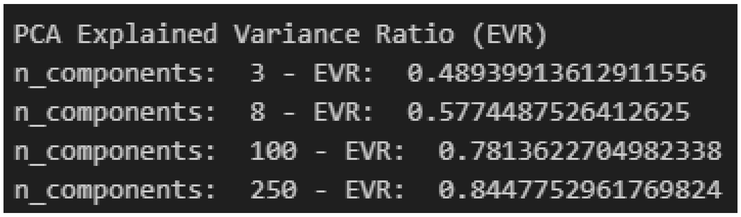
Once the dataset has been loaded, normalize and standardize data with a Scaler and apply a method to extract features and reduce feature dimensionality. It

[1] https://www.kaggle.com/datasets/gpiosenka/cards-image-datasetclassification?resource=download

is very important because it transforms raw data into more meaningful representations of the model and reduces the dimension, maintaining the essence of the data. There are many techniques, I will show 3: *PCA, LDA and t-SNE*. It extracts the most informative features from datasets, preserving the most relevant information. This reduces the complexity of the model

2.1 PCA (Principal Component Analysis)

PCA transforms potentially correlated variables into a smaller set of variables, called principal components. It extracts the most informative features from large datasets, preserving the most relevant information from the initial dataset. This reduces the complexity of the model. I evaluated the results using the explained variance ratio (EVR) metric, the percentage of variance explained by each of the selected components[2].

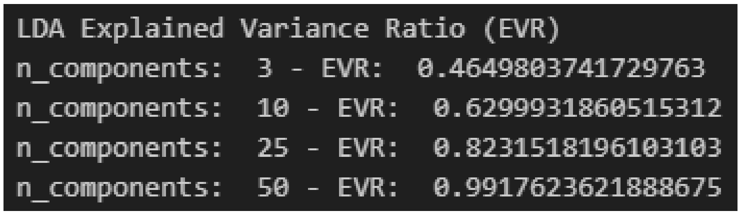


```
PCA Explained Variance Ratio (EVR)
n_components:  3 - EVR:  0.48939913612911556
n_components:  8 - EVR:  0.5774487526412625
n_components:  100 - EVR:  0.7813622704982338
n_components:  250 - EVR:  0.8447752961769824
```

Figure 2: PCA EVR results with n. components

2.2 LDA (Linear Discriminant Analysis)

The main idea of LDA is finding a projection to a line such that data samples of different classes are well separated and dimensionally reduced. The maximum number of components are *(k-1)*, where *k* is the number of classes.



```
LDA Explained Variance Ratio (EVR)
n_components:  3 - EVR:  0.4649803741729763
n_components:  10 - EVR:  0.6299931860515312
n_components:  25 - EVR:  0.8231518196103103
n_components:  50 - EVR:  0.9917623621888675
```

Figure 3: LDA EVR results with n. components until 50

2.3 t-SNE (t-distributed Stochastic Neighbor Embedding)

Last method is t-SNE, very useful for visualizing high-dimensional data in lower dimensions like 2D or 3D. It reduces dimensionality by maintaining the closeness

[2] `https://scikit-learn.org/stable/modules/generated/sklearn.decomposition.PCA.html`

relationship between points, emphasizing local clusters.

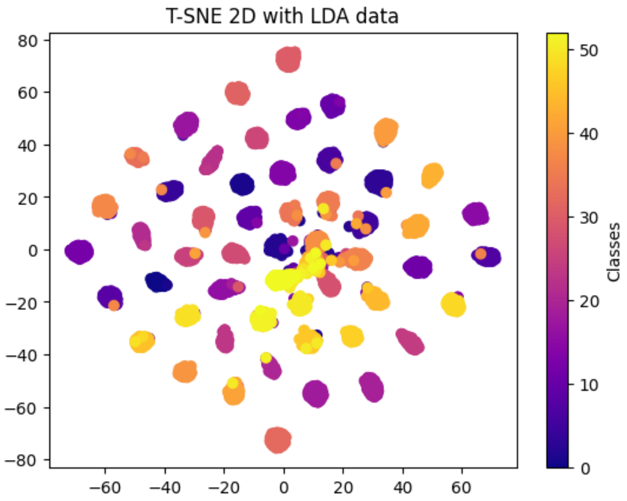


Figure 4: t-SNE results with n. components=2

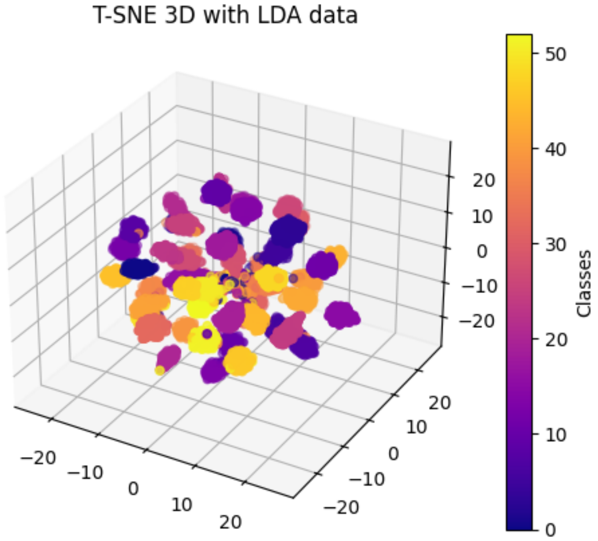


Figure 5: t-SNE results with n. components=3

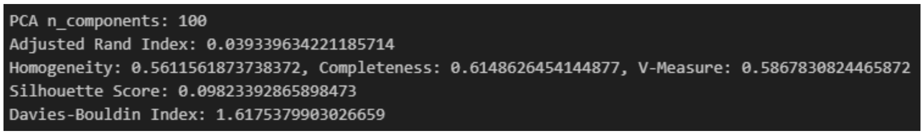
3 Classification and Clustering

Classification is a supervised learning task in which a model is trained to predict a category for new data, using a dataset already labeled. Clustering is an unsupervised learning task in which data is grouped into clusters based on similar characteristics. For each method I used PCA and LDA data, to see the difference between them.

3.1 K-Means

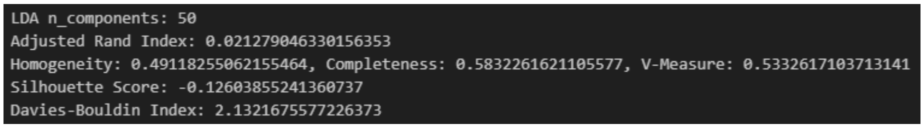
We have to know a priori how many clusters we need (k). The K-Means algorithm may not perform well when clusters are not spherical or if clusters have different densities. I evaluated the results using:

- Adjusted Rand Index (ARI) checks the similarity between two partitions of a data set, for example between a clustering produced by an algorithm and a true partition. From 0 to 1.
- Silhouette score that goes from -1 (points are in the wrong cluster) to 1 (cluster is well defined)
- Davies-Bouldin Index that measures the compactness and separation between clusters. Lower values indicate better clusters.



```
PCA n_components: 100
Adjusted Rand Index: 0.039339634221185714
Homogeneity: 0.5611561873738372, Completeness: 0.6148626454144877, V-Measure: 0.5867830824465872
Silhouette Score: 0.09823392865898473
Davies-Bouldin Index: 1.6175379903026659
```

Figure 6: Best values with PCA (n. components = 100)

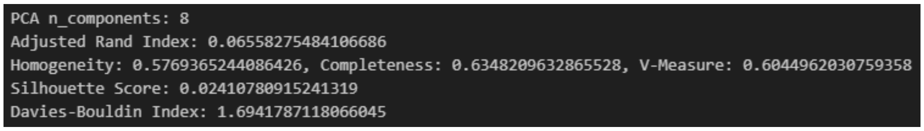


```
LDA n_components: 50
Adjusted Rand Index: 0.021279046330156353
Homogeneity: 0.49118255062155464, Completeness: 0.5832261621105577, V-Measure: 0.5332617103713141
Silhouette Score: -0.12603855241360737
Davies-Bouldin Index: 2.1321675577226373
```

Figure 7: Best values with LDA (n. components = 50)

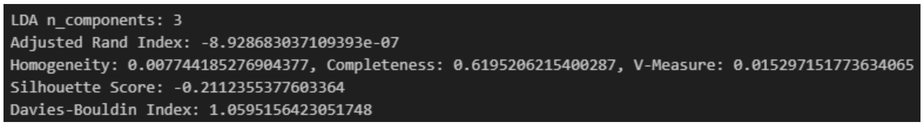
3.2 GMM (Gaussian Mixture Model)

A Gaussian mixture model is a probabilistic model that assumes all the data points are generated from a mixture of a finite number of Gaussian distributions with unknown parameters.



```
PCA n_components: 8
Adjusted Rand Index: 0.06558275484106686
Homogeneity: 0.5769365244086426, Completeness: 0.6348209632865528, V-Measure: 0.6044962030759358
Silhouette Score: 0.02410780915241319
Davies-Bouldin Index: 1.6941787118066045
```

Figure 8: Best values with PCA (n. components = 8)

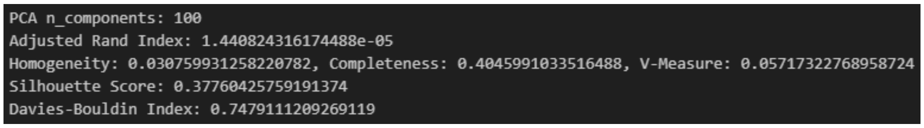


```
LDA n_components: 3
Adjusted Rand Index: -8.928683037109393e-07
Homogeneity: 0.007744185276904377, Completeness: 0.6195206215400287, V-Measure: 0.015297151773634065
Silhouette Score: -0.2112355377603364
Davies-Bouldin Index: 1.0595156423051748
```

Figure 9: Best values with LDA (n. components = 3)

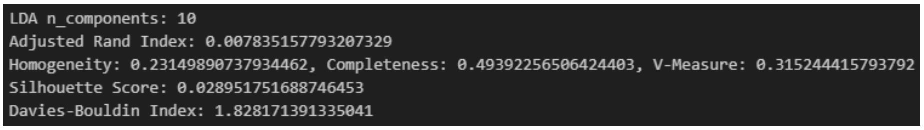
3.3 Mean-Shift

It is a centroid-based algorithm, which works by updating candidates for centroids to be the mean of the points within a given region. Doesn't require prior knowledge of the number of clusters. Uses a kernel function (e.g., Gaussian) to estimate the density of data points.



```
PCA n_components: 100
Adjusted Rand Index: 1.440824316174488e-05
Homogeneity: 0.030759931258220782, Completeness: 0.4045991033516488, V-Measure: 0.05717322768958724
Silhouette Score: 0.37760425759191374
Davies-Bouldin Index: 0.7479111209269119
```

Figure 10: Best values with PCA (n. components = 100)

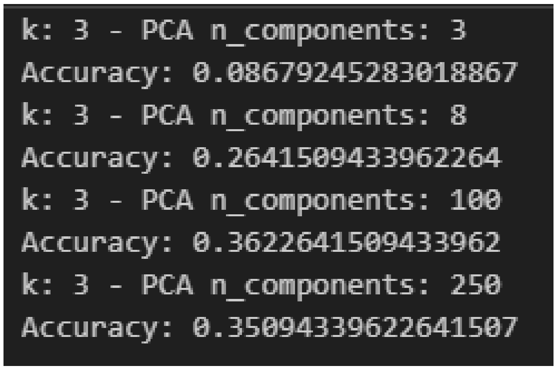


```
LDA n_components: 10
Adjusted Rand Index: 0.007835157793207329
Homogeneity: 0.23149890737934462, Completeness: 0.49392256506424403, V-Measure: 0.315244415793792
Silhouette Score: 0.028951751688746453
Davies-Bouldin Index: 1.828171391335041
```

Figure 11: Best values with LDA (n. components = 10)

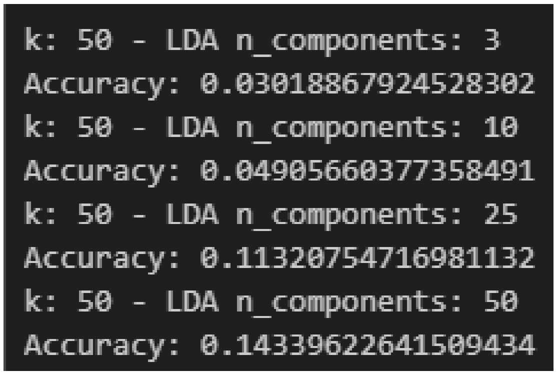
3.4 KNN (K-Nearest Neighbors)

KNN calculates the distance between a point and all points in the training dataset. The k closest points are identified and the most common label among the neighbors is assigned. It is important to evaluate k, the number of neighbors to consider.



```
k: 3 - PCA n_components: 3
Accuracy: 0.08679245283018867
k: 3 - PCA n_components: 8
Accuracy: 0.2641509433962264
k: 3 - PCA n_components: 100
Accuracy: 0.3622641509433962
k: 3 - PCA n_components: 250
Accuracy: 0.35094339622641507
```

Figure 12: Best values with PCA (k = 3)

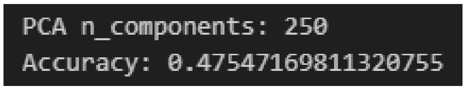


```
k: 50 - LDA n_components: 3
Accuracy: 0.03018867924528302
k: 50 - LDA n_components: 10
Accuracy: 0.04905660377358491
k: 50 - LDA n_components: 25
Accuracy: 0.11320754716981132
k: 50 - LDA n_components: 50
Accuracy: 0.14339622641509434
```

Figure 13: Best values with LDA (k = 50)

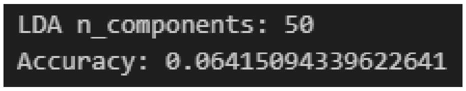
3.5 SVM (Support Vector Machines)

The goal of SVM is to find an optimal hyperplane that separates the data into different classes.



```
PCA n_components: 250
Accuracy: 0.47547169811320755
```

Figure 14: Best values with PCA (n. components = 250)

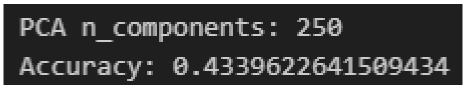


```
LDA n_components: 50
Accuracy: 0.06415094339622641
```

Figure 15: Best values with LDA (n. components = 50)

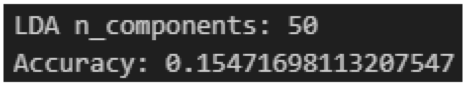
3.6 SGD (Stochastic Gradient Descent)

The goal of SGD is to reduce the error (or loss function) of a model by iteratively updating its parameters.



```
PCA n_components: 250
Accuracy: 0.4339622641509434
```

Figure 16: Best values with PCA (n. components = 250)



```
LDA n_components: 50
Accuracy: 0.15471698113207547
```

Figure 17: Best values with LDA (n. components = 50)

3.7 Overview

The general results obtained are very poor. For example in K-Means I have an ARI of only 4% with PCA, while LDA does even worse with only 2%. Same thing with GMM and Mean-Shift. The situation instead improves with KNN, SVM and SGD, which bring good results with PCA, but not with LDA. This is why I decided to try extracting features with ResNet, a convolutional neural network, to see if ARI improves.

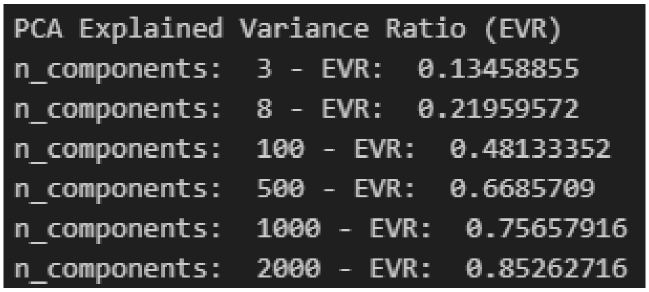
4 Feature Extraction with ResNet

A residual neural network (ResNet) is a deep learning architecture in which the layers learn residual functions with reference to the layer inputs[3].

4.1 Results with PCA and LDA

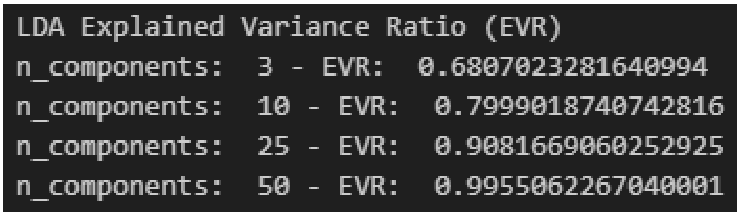
After creating the model with ResNet, I reduced the dimensionality again with PCA and LDA. You can notice a different accuracy: in PCA I have to increase the number of components to have the same accuracy of before, while in LDA I can also decrease the components.

[3] https://en.wikipedia.org/wiki/Residual_neural_network



```
PCA Explained Variance Ratio (EVR)
n_components:  3 - EVR:  0.13458855
n_components:  8 - EVR:  0.21959572
n_components:  100 - EVR:  0.48133352
n_components:  500 - EVR:  0.6685709
n_components:  1000 - EVR:  0.75657916
n_components:  2000 - EVR:  0.85262716
```

Figure 18: PCA EVR results with n. components with ResNet

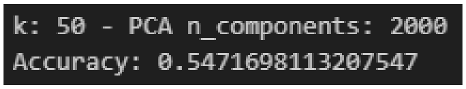


```
LDA Explained Variance Ratio (EVR)
n_components:  3 - EVR:  0.6807023281640994
n_components:  10 - EVR:  0.7999018740742816
n_components:  25 - EVR:  0.9081669060252925
n_components:  50 - EVR:  0.9955062267040001
```

Figure 19: LDA EVR results with n. components with ResNet

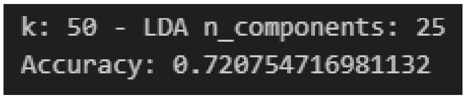
4.2 KNN with ResNet

I tried to make a prediction only with KNN and the results are largely better: with LDA I achieved an accuracy of 73%, while with PCA it was 54%. Note that in general I got the best values with the highest number of components.



```
k: 50 - PCA n_components: 2000
Accuracy: 0.5471698113207547
```

Figure 20: Best values with PCA and ResNet



```
k: 50 - LDA n_components: 25
Accuracy: 0.720754716981132
```

Figure 21: Best values with PCA and ResNet

Given the high accuracy, it seemed interesting to include the confusion matrix. It is a square matrix of the size of the number of clusters where the rows represent the actual clusters, while the columns represent the clusters predicted by the model. The values on the diagonal represent the samples that are classified correctly, while off the diagonal you can see where the incorrect samples are classified. As can be seen in the figure, the diagonal has the most values.

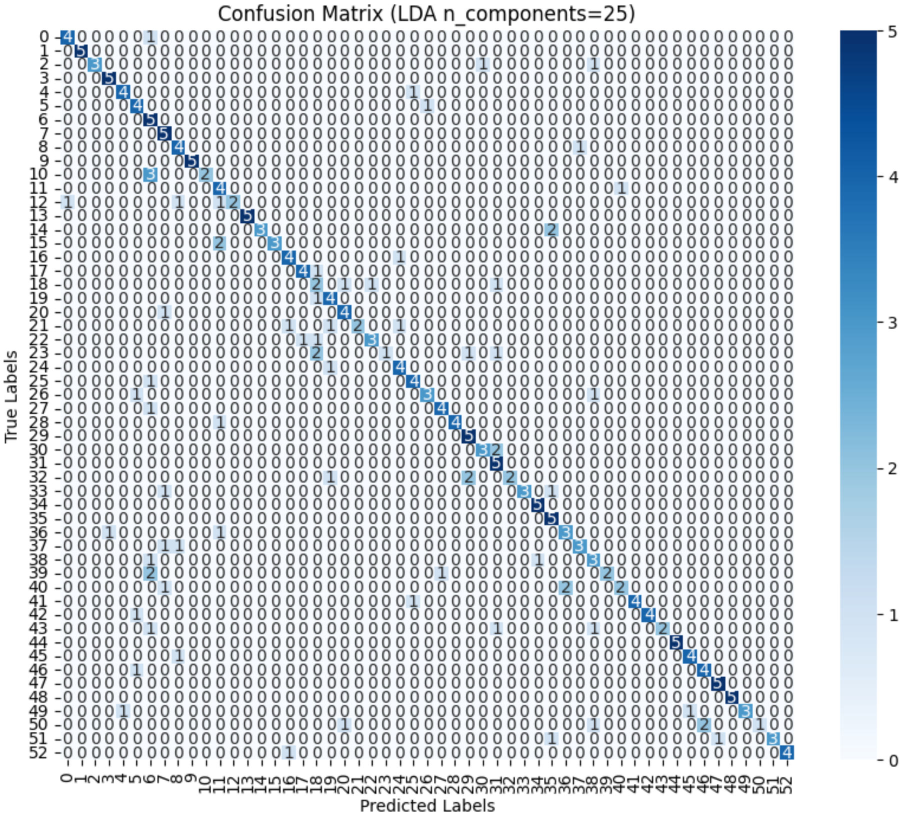


Figure 22: Confusion Matrix with 72% of accuracy

5 Conclusions

In this project, various dimensionality reduction methods, classification and clustering techniques were tested and applied to a dataset of playing card images. The analysis showed that traditional methods such as PCA and LDA brought different results: while some some techniques (e.g. KNN, SVM, SGD) achieved good levels of accuracy with PCA, other methods (K-Means, GMM, Mean-Shift) provided bad results.

The ResNet neural network has instead significantly improved the situation, obtaining superior performance, in particular with KNN, reaching an accuracy of 72% with LDA. This demonstrates how deep learning techniques allow an improvement compared to traditional techniques.