Compte Rendu Projet Eigenface

1. Introduction

Le but de ce projet est de permettre de faire une reconnaissance faciale en employant des photos de référence issues d'un extrait du dataset de visages LFW (Labeled Faces in the Wild) de l'Université du Massachusetts. Pour chaque visage, il faut lui attribuer la classe qui a la probabilité la plus élevée de correspondre.

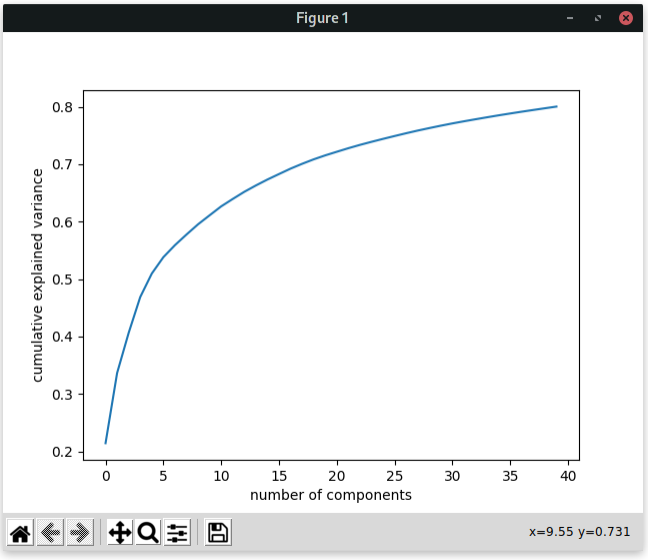
Pour cela, la méthode de reconnaissance faciale Eigenface a recours à l'Analyse en Composantes Principales pour simplifier le traitement, ainsi que différents classifieurs permettant de définir la classe d'un visage, c'est-à-dire le nom de la personne. C'est une méthode globale puisque l'ensemble du visage est analysé et non seulement certaines caractéristiques.

2. Principes fondamentaux

A. Analyse en composantes principales (ACP)

L'analyse en composantes principales (ou Principal Component Analysis) est une méthode qui consiste à transformer un ensemble de variables liées (= corrélées) entre elles en nouvelles variables décorrélées les unes des autres et qui sont des combinaisons linéaires des variables d'origine. Ces nouvelles variables sont appelées **composantes principales** ou **axes principaux**. Elle permet de réduire le nombre de variables/données et de rendre l'information moins redondante.

Les axes étant indépendants permettent de mieux expliquer la variance des données comme on peut parfaitement le voir sur ce graphique de la somme cumulée des variances expliquées par chacune des composantes :



Sur ce graphe, on comprend que les composantes principales permettent d'expliquer environ 80% de la variance des données.

Dans notre projet, ACP va servir à calculer les approximations des visages de notre dataset pour l'entraînement et le test de prédiction du classifieur. La qualité des estimations auxquelles conduit l'ACP dépend du choix du nombre de composantes retenues pour reconstituer les données.

On applique en général une ACP sur un ensemble de variables aléatoires (X1, X2, …, Xn)

B. Classifieurs

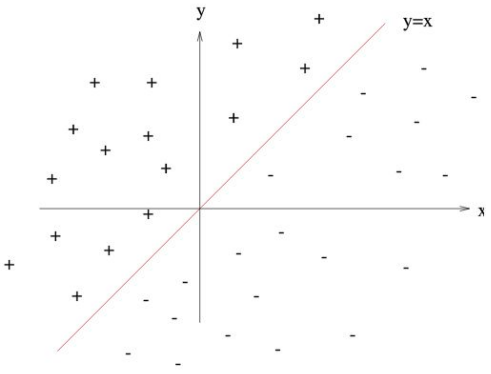
SVM

Les **machines à vecteurs de support** (SVM) sont un ensemble de techniques d'apprentissage supervisé destinées à résoudre des problèmes de **discrimination**, c'est-à-dire décider à quelle classe appartient un échantillon, et de **régression**, c'est-à-dire prédire la valeur numérique d'une variable. Les SVM sont une généralisation des classifieurs linéaires.

Pour résoudre ces deux problèmes (discrimination et régression), on construit une fonction h, qui fait correspondre à un vecteur d'entrée x, une sortie y : y = h(x).
Pour réaliser une discrimination à deux classes (discrimination binaire, par ex: vrai/faux), on aurait $y \in \{-1, 1\}$ et x est le vecteur d'entrée situé dans un espace vectoriel $X = \mathbb{R}^N$ de dimension N.

La fonction discriminante est obtenue par la combinaison linéaire du vecteur d'entrée x avec un vecteur de poids w. L'équation est de la même forme que celle d'une fonction affine, **y = h(x) = wx + w_0**. La frontière de décision est l'équation h(x) = 0. Donc, x est de classe 1 si h(x) >= 0, -1 sinon. Ceci est un **classifieur linéaire**.

La frontière de décision est aussi appelée **séparatrice** et on peut la représenter graphiquement :



Pour des problèmes plus complexes, il est fort possible que cela ne soit pas séparable par une droite. On dit que le problème n'est pas linéairement séparable.

Afin de remédier au problème de l'absence de séparateur linéaire, l'idée de l'**astuce du noyau** *(en anglais kernel trick)* est de reconsidérer le problème dans un espace de dimension supérieure. Dans ce nouvel espace, il est alors probable qu'il existe une séparation linéaire. Avec cette astuce, le calcul de l'équation de l'hyperplan séparateur dépend d'une fonction noyau. Cette fonction doit correspondre à un produit scalaire dans un espace de grande dimension. Il existe des noyaux usuels : linéaire, polynomial, gaussien, etc...

K Nearest Neighbors (KNN)

La méthode des k plus proches voisins est une méthode d'apprentissage supervisée comme la précédente. On dispose d'un jeu de données d'apprentissage composé de N couples entrée-sortie. Pour estimer la sortie y associée à une nouvelle entrée x, la méthode des k plus proches voisins consiste à prendre en compte (de façon identique) les k échantillons d'apprentissage dont l'entrée est la plus proche de la nouvelle entrée x, selon une distance à définir.

Étant donné que cette méthode repose sur un calcul de distance euclidienne, la normalisation des données (soustraire la moyenne et diviser par l'écart-type, ainsi l'espérance vaut 0 et l'écart-type vaut 1) peut améliorer la précision. Comme SVM, il peut servir pour résoudre des problèmes de classification et de régression.

Dans un problème de classification, on retiendra la classe la plus représentée parmi les k sorties associées aux k entrées les plus proches de la nouvelle entrée x alors que dans un problème de régression, le résultat est la valeur pour cette entrée. Cette valeur est la moyenne des valeurs des k plus proches voisins.

Que ce soit pour la classification ou la régression, une technique efficace peut être utilisée pour pondérer l'influence contributive des voisinages, ainsi les plus proches voisins ont une influence plus forte sur le calcul de la moyenne (pondérée) que les voisins plus éloignés. Par exemple, il est fréquent de donner à chaque voisin une pondération de 1/d, où d est la distance de l'élément, à classer ou à pondérer, de ce voisin.

L'algorithme k-NN est parmi les plus simples des algorithmes de machine learning.

Random Forest (Forêt d'arbres décisionnels)

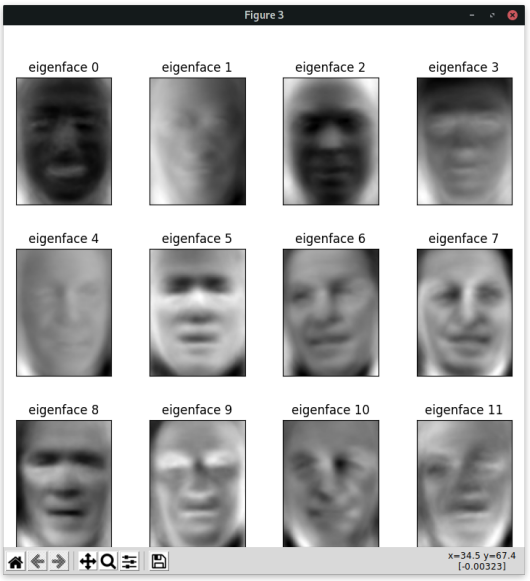
L'**algorithme des forêts d'arbres décisionnels** effectue un apprentissage sur de multiples arbres de décision entraînés sur des sous-ensembles de données légèrement différents. Cet algorithme corrige plusieurs inconvénients connus des arbres de décisions. On peut notamment citer le fait que cela peut mener à des arbres de décision très complexes qui généralisent mal l'ensemble d'apprentissage, que certains concepts sont difficiles à exprimer comme XOR ou la parité ce qui nous renvoie au premier problème, la taille des arbres mais aussi la sensibilité à l'ordre des prédicteurs.

Le principal inconvénient des forêts d'arbres décisionnels est que l'on perd l'aspect visuel des arbres de décision uniques.

3. Mise en pratique & Déroulement du projet

Tout d'abord, au niveau du dataset, nous ne gardons que les personnes qui ont au minimum 25 images de leur visage. Nous avons donc 1941 visages de personnes différentes. Grâce à l'analyse en composantes principales, nous allons pouvoir diminuer ce nombre de visages. Le nombre de composantes est le nombre de visages les plus représentatifs du dataset que nous souhaitons garder au maximum. C'est aussi le nombre de dimensions du nouvel espace vectoriel dont les eigenfaces en constituent la base. Les eigenfaces sont les visages les plus représentatifs.

Récapitulons. Les eigenfaces sont un ensemble de vecteurs propres qui permettent de définir la base d'un nouvel espace vectoriel. Chaque vecteur propre a la même dimension (ou nombre de pixels) que les images d'origine (112x84 soit 9408 pixels) et peuvent être affichés comme des images. Si nous souhaitons travailler avec 40 eigenfaces, nous aurons un espace vectoriel à 40 dimensions ce qui est nettement mieux que 9408 dimensions… Cela produit donc une réduction de dimension en permettant à un ensemble restreint d'images de représenter les images d'origine.



Pour résumé, voici la signification des paramètres utilisés :
Composantes principales : Ce sont les vecteurs propres (non-corrélés entre eux) calculés par ACP. Ce sont les "axes" d'un espace vectoriel. Ils peuvent être affichés comme des images. Il s'agit de visages représentant et approximant l'ensemble du dataset. Ici, nous les appelons eigenfaces ou eigenimages.

Classes : Les "catégories" dans lesquelles on classe les images. Ici, cela correspond à l'identité des personnes. Pour nos prédictions avec l'ensemble de test, nous allons seulement utiliser 42 classes (soit 42 personnes)

Résolution des images : C'est le nombre de pixels en largeur multiplié par le nombre de pixels en hauteur. C'est aussi le nombre de dimensions des vecteurs propres (eigenfaces). Plus les images ont une résolution élevée, plus la matrice de covariance est grande (pour

une image de 112x84, chaque image est un point dans un espace de dimension 9408, la taille de la matrice de covariance est donc de 9408x9408).

Taille de l'ensemble d'apprentissage : C'est le nombre d'images sélectionnées pour entraîner et créer un modèle pour un classifieur. Au début, l'ensemble d'apprentissage constitue 75% du dataset et l'ensemble de test en constitue 25% mais c'est modifiable.

Étapes du programme

1. On **prépare un ensemble d'images pour l'entraînement et un ensemble d'images pour le test de prédiction** à partir du dataset LFW. Toutes les images doivent avoir la même résolution. Chaque image est considérée comme un vecteur. Nous avons donc un tableau numpy qui contient sur chaque ligne 1 image, soit une shape **(1941, 9408)**
2. On **effectue une Analyse en Composantes Principales** : On soustrait l'image moyenne à chaque image de l'ensemble d'entraînement. Puis, les vecteurs propres et les valeurs propres de la matrice de covariance sont calculés. Chaque vecteur propre a la même dimension *(ou nombre de composantes)* que les images d'origine et peut être affiché comme une image. C'est pour cela qu'on appelle ça des eigenfaces. Ils représentent les variations des images par rapport à la moyenne.
   Remarque : Calculer les vecteurs propres à partir de la matrice de covariance est très coûteux en terme de calcul, voire infaisable. C'est pour ça que nous appliquons la **Décomposition en Valeurs Singulières** (SVD) sur l'ensemble d'entraînement pour obtenir les vecteurs et valeurs propres sans devoir la calculer.
3. On **projette les données d'entraînement et de test sur la base constituée par les vecteurs propres** (ou eigenfaces) ce qui permet d'effectuer une réduction de dimension, aussi appelée une compression. Nous avons alors un nouveau tableau numpy de forme **(nbre d'images, nbre de composantes principales)**
4. *(J'effectue ma reconstruction faciale)*
5. On entraîne le classifieur choisi (SVM, KNN, etc…) avec l'ensemble d'images d'entraînement. => *Méthode fit()*
6. On effectue un test de prédiction sur l'ensemble d'images de test => *Méthode predict()*

Reconstruction Faciale

Après avoir réalisé l'Analyse en Composantes Principales, j'ai écrit un morceau de code Python pour faire la reconstruction faciale d'un visage du dataset à partir de de la moyenne des visages, et du calcul du produit scalaire entre l'image d'origine, à laquelle on a soustrait l'image moyenne, et les vecteurs propres calculés par ACP. Ce produit scalaire nous permet d'avoir des coefficients/poids qu'il faut ensuite multiplier avec les eigenfaces. Il y a autant d'eigenfaces que de composantes principales.

Dans la partie Résultats, je montre l'évolution de la qualité de la reconstruction en changeant le nombre de composantes principales.

```
#### Face Reconstruction
faceIndex = 24 # 25th face out of 1941 faces in the dataset

face = X_train[faceIndex] # Original face image
faceVector = face.flatten() - mean_face # Face substracted with mean face
output = mean_face.copy() # Output image
eigenvectors = X_train_pca[faceIndex] # eigenvectors returned by PCA

# For each principal component/eigenface
for componentIdx in range(n_components):
    weights = faceVector.dot(eigenvectors[componentIdx]) # Dot product between original face and eigenvectors
    output = output + eigenfaces[componentIdx].flatten() * weights # Sum of the mean face with the eigenfaces multiplied by weights
output = output.reshape(112, 84)

# Display reconstruction result
n_col = 2
n_row = 1
plt.figure(figsize=(1.8 * n_col, 2.4 * n_row))
plt.subplots_adjust(bottom=0, left=.01, right=.99, top=.90, hspace=.35)

plt.subplot(n_row, n_col, 1)
plt.imshow(face.reshape((h, w)), cmap=plt.cm.gray)
plt.title("Original", size=12)

plt.subplot(n_row, n_col, 2)
plt.imshow(output.reshape((h, w)), cmap=plt.cm.Greys)
plt.title("Face n°" + str(faceIndex) + " Reconstruction", size=12)

plt.xticks(())
plt.yticks(())
plt.show()
```

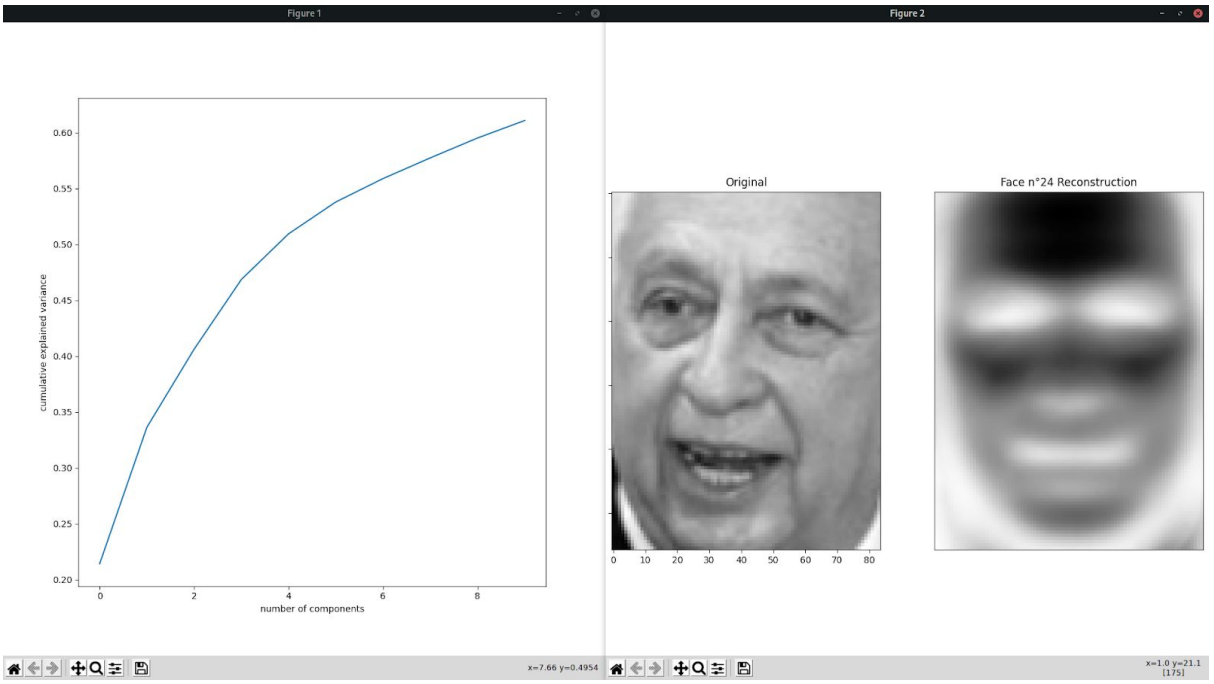
Classifieurs

Ensuite, nous effectuons une classification avec SVM mais d'autres classifieurs tels que ceux cités plus tôt dans ce rapport peuvent être utilisés. Nous allons d'abord comparer les différents noyaux de SVM ainsi que l'efficacité d'un autre classifieur.
Cf partie Résultats.

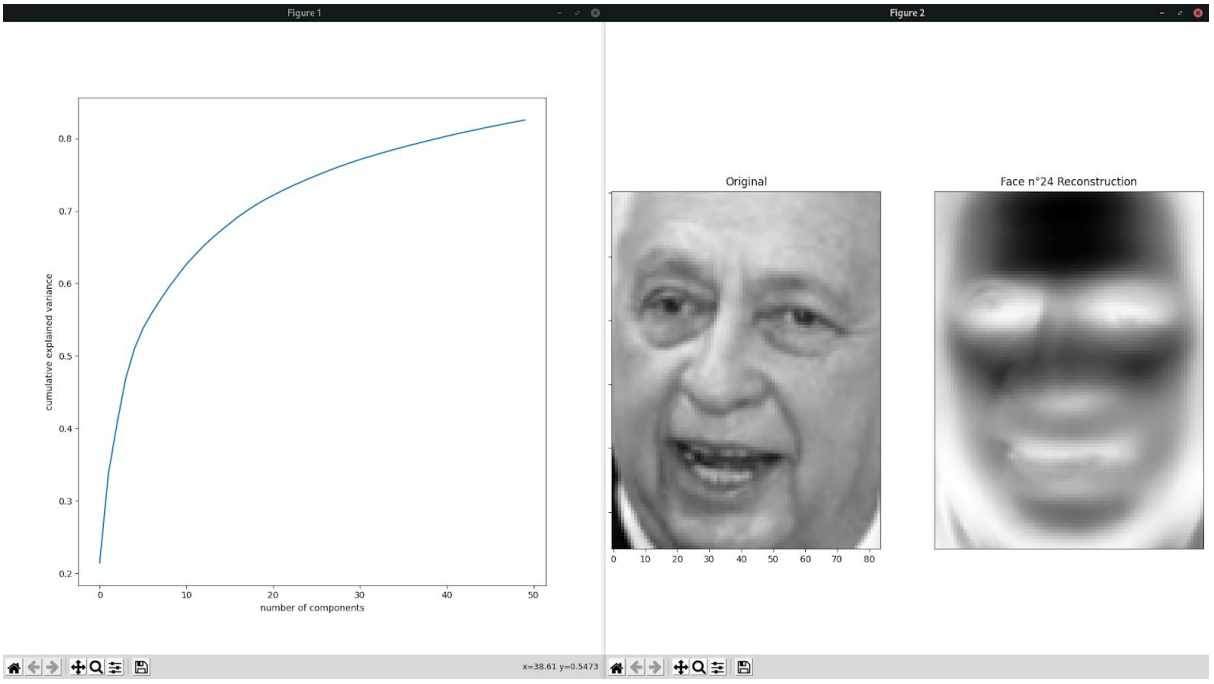
4. Résultats

Reconstruction Faciale

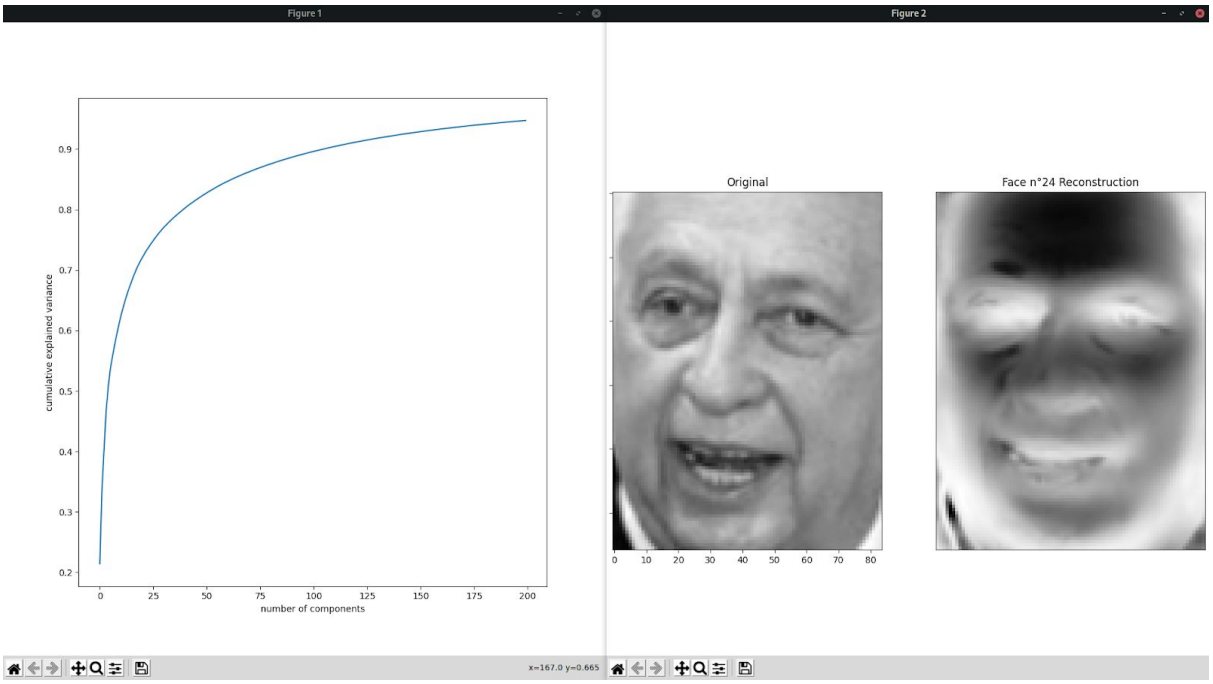
10 composantes principales



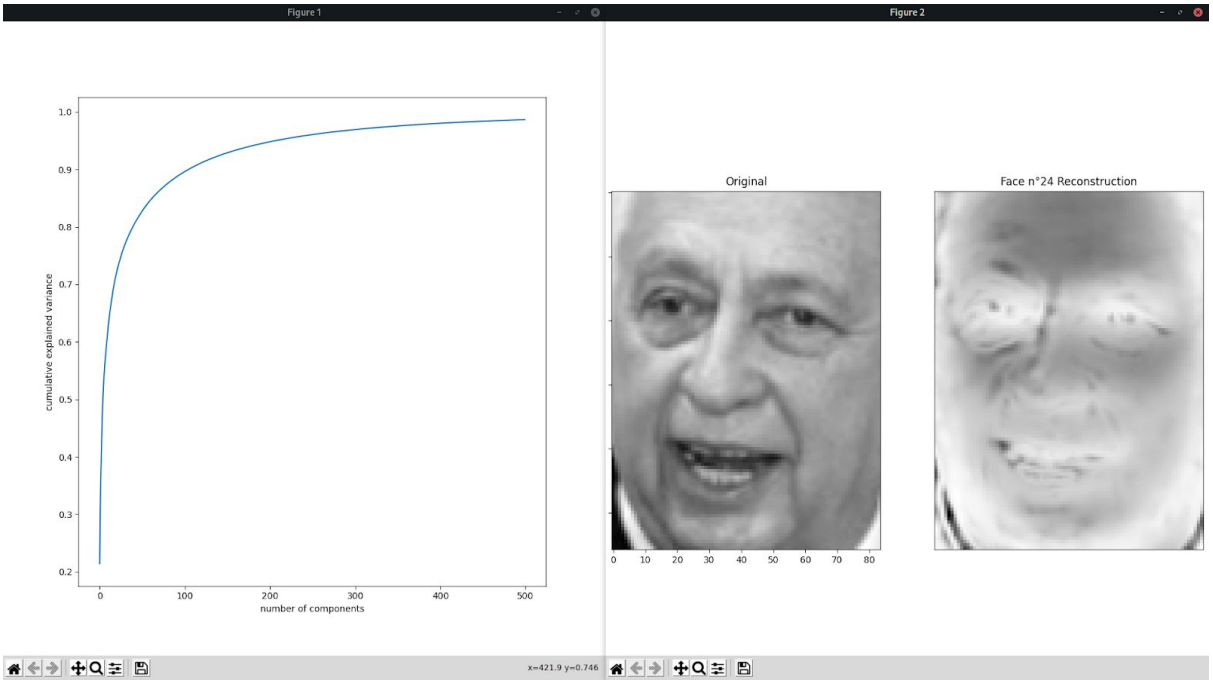
50 composantes principales



200 composantes principales



500 composantes principales



On remarque qu'à mesure que l'on augmente le nombre de composantes principales, le visage reconstruit est de plus en plus détaillé et éloigné du visage moyen, calculé sur l'ensemble du dataset. 500 composantes principales semblent suffir pour reconstruire le visage dans son intégralité puisque nous atteignons une variance expliquée cumulée égale à 1, soit 100% ce qui signifie que les 500 eigenfaces représentent suffisamment la variance des visages du dataset par rapport à la moyenne.

Métriques d'évaluation

Recall : Le rappel est un indicateur qui mesure la **capacité du modèle à prédire l'ensemble des résultats attendus**. Ainsi, un modèle peut avoir une très bonne précision mais avoir un mauvais rappel. En effet, un modèle peut-être exact lorsqu'il prédit la présence d'un objet, mais ne pas détecter des objets qui auraient dû l'être. Il se calcule par ce ratio : *Vrai Positifs / (Vrai Positifs + Faux Négatifs)*

Precision : La précision est le pourcentage de détections correctes. Il met en évidence l'**exactitude des prédictions**. Il se calcule par ce ratio : *Vrai Positifs / (Vrai Positifs + Faux Positifs)*

Accuracy : Taux de reconnaissance : *(Vrai positifs + Vrai Négatifs) / (Vrai Positifs + Vrai Négatifs + Faux Positifs + Faux Négatifs)*

F1-Score : Il combine la précision et le rappel.
Il se calcule par ce ratio : *2*Precision*Rappel/(Precision + Rappel)*

ROC (Receiver Operating Characteristic) : Cette mesure donne le taux de Vrais Positifs en fonction du taux de Faux Positifs.

DET (Detection Error Tradeoff) : Le taux de Faux Négatifs en fonction des Faux Positifs
N'existe pas dans la version stable de scikit-learn (0.23.2) mais sera ajouté dans la future version (0.24)

Classifieurs

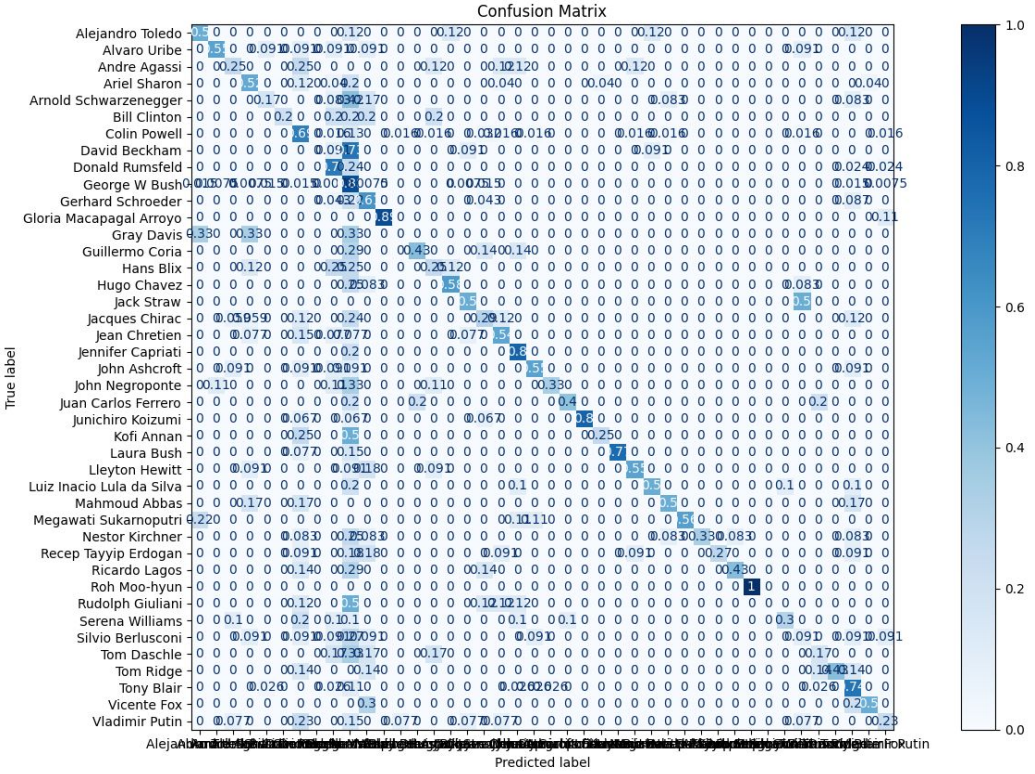
SVM

Sur ce classifieur, on change la fonction noyau (kernel) avec le paramètre "kernel" de sklearn.svm.SVC().

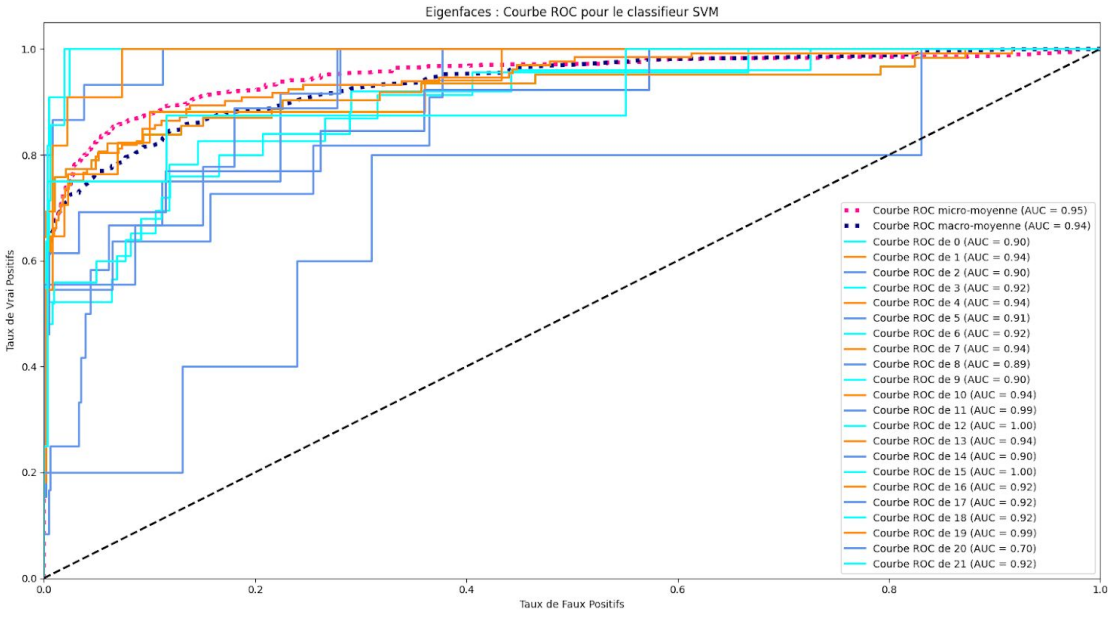
Noyau RBF (Radial Basis Function)

Tout d'abord, la matrice de confusion (dont les valeurs ont été normalisées). Elle nous permet aussi d'évaluer la qualité de la sortie d'un classifieur. Les éléments en diagonal dans cette matrice représentent la proportion de visages dont la classe prédite est égale à celle connue dans l'ensemble de test. Ce qui est en diagonal correspond donc à des prédictions réussies. Plus les proportions dans la diagonale sont élevées, plus les prédictions sont correctes.

Nous pouvons distinguer de manière assez distincte une diagonale avec des cases plus foncées, indiquant des proportions plus élevées que dans les autres cases. On peut tout de même remarquer encore pas mal d'erreurs de prédiction. Les résultats sont corrects mais perfectibles.



Passons à la courbe ROC :



Tout d'abord, on peut remarquer 2 courbes pour la micro-moyenne et la macro-moyenne. On établit la courbe pour la micro-moyenne en se ramenant à une classification binaire individuellement pour chaque classe. Tandis que pour la macro-moyenne, chaque classe a une pondération identique.

Ensuite, j'ai également tracé les courbes ROC des 21 premières classes. Dans la partie supérieure gauche du graphique, on peut trouver le point "idéal" puisque cela signifie que nous n'avons que des Vrai Positifs et aucun Faux Positif. Ce n'est évidemment pas réaliste. L'objectif est donc d'avoir la courbe avec la pente la plus abrupte possible et d'avoir la surface en dessous de la courbe (AUC, Area under Curve) la plus grande possible.

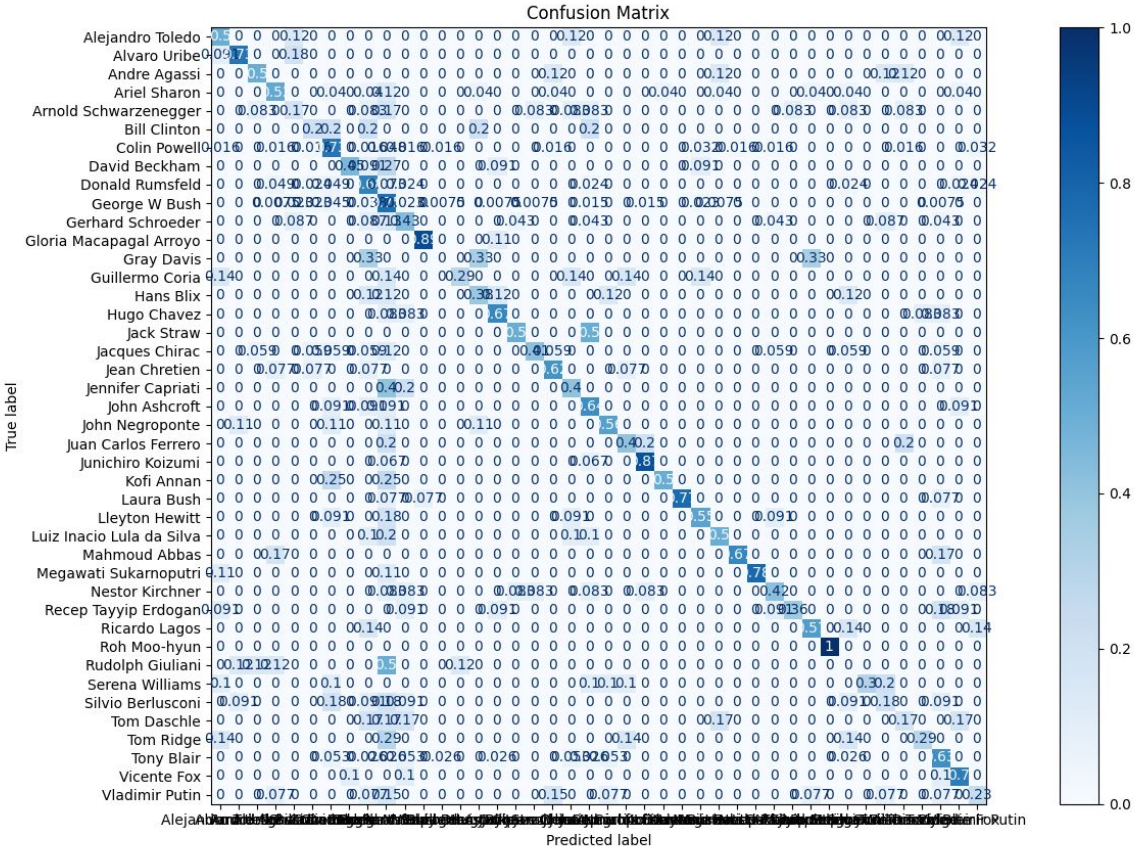
Malgré certaines classes avec une AUC plus faible que les autres, la macro-moyenne et la micro-moyenne sont plus que correctes (~0.95). Il aurait été intéressant d'étudier la courbe DET qui permet de visualiser la relation entre les Faux Positifs et les Faux Négatifs. De plus, que vaut-il mieux privilégier ? Vaut-il mieux avoir beaucoup de Faux Positifs ou beaucoup de Faux Négatifs ? Tout dépend du contexte mais pour un dispositif de sécurité, il vaudra mieux privilégier les Faux Négatifs… C'est frustrant pour l'utilisateur mais la sécurité n'est pas en péril. L'idéal est évidemment d'en avoir peu des deux.

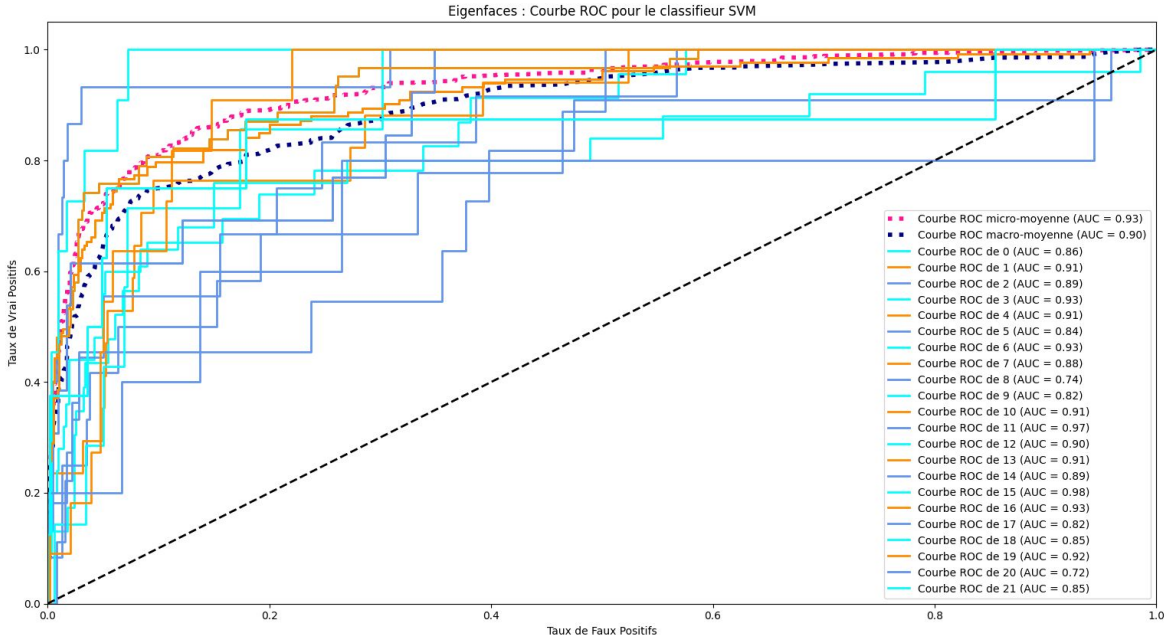
	precision	recall	f1-score	support
Alejandro Toledo	0.50	0.50	0.50	8
Alvaro Uribe	0.75	0.55	0.63	11
Andre Agassi	0.25	0.25	0.25	8
Ariel Sharon	0.56	0.56	0.56	25
Arnold Schwarzenegger	0.50	0.25	0.33	12
Bill Clinton	0.25	0.20	0.22	5
Colin Powell	0.70	0.77	0.73	62
David Beckham	0.80	0.36	0.50	11
Donald Rumsfeld	0.64	0.68	0.66	41
George W Bush	0.63	0.90	0.74	133
Gerhard Schroeder	0.52	0.70	0.59	23
Gloria Macapagal Arroyo	1.00	0.78	0.88	9
Gray Davis	0.00	0.00	0.00	3
Guillermo Coria	0.75	0.43	0.55	7
Hans Blix	0.50	0.38	0.43	8
Hugo Chavez	0.70	0.58	0.64	12
Jack Straw	0.00	0.00	0.00	2
Jacques Chirac	0.85	0.65	0.73	17
Jean Chretien	0.62	0.62	0.62	13
Jennifer Capriati	0.31	0.80	0.44	5
John Ashcroft	0.62	0.73	0.67	11
John Negroponte	1.00	0.44	0.62	9
Juan Carlos Ferrero	0.25	0.20	0.22	5
Junichiro Koizumi	0.93	0.87	0.90	15
Kofi Annan	0.50	0.25	0.33	4
Laura Bush	1.00	0.77	0.87	13
Lleyton Hewitt	0.60	0.55	0.57	11
Luiz Inacio Lula da Silva	0.71	0.50	0.59	10
Mahmoud Abbas	0.83	0.83	0.83	6
Megawati Sukarnoputri	1.00	0.78	0.88	9
Nestor Kirchner	1.00	0.58	0.74	12
Recep Tayyip Erdogan	0.80	0.36	0.50	11
Ricardo Lagos	0.80	0.57	0.67	7
Roh Moo-hyun	1.00	1.00	1.00	6
Rudolph Giuliani	0.33	0.12	0.18	8
Serena Williams	1.00	0.30	0.46	10
Silvio Berlusconi	0.67	0.18	0.29	11
Tom Daschle	0.50	0.17	0.25	6
Tom Ridge	1.00	0.43	0.60	7
Tony Blair	0.62	0.79	0.70	38
Vicente Fox	1.00	0.70	0.82	10
Vladimir Putin	0.38	0.23	0.29	13
accuracy			0.65	647
macro avg	0.65	0.51	0.55	647
weighted avg	0.67	0.65	0.64	647

L'accuracy ou le taux de reconnaissance est un peu faiblard mais reste correct *(0.65)*. Quant à la précision/l'exactitude des prédictions sont correctes. Toutefois, on constate des chutes

comprises entre 0.25 et 0.38. Au niveau du rappel, sa capacité à prédire ce qui était attendu est plutôt mauvaise. Beaucoup de choses n'ont pas pu être prédites.

Noyau Linéaire

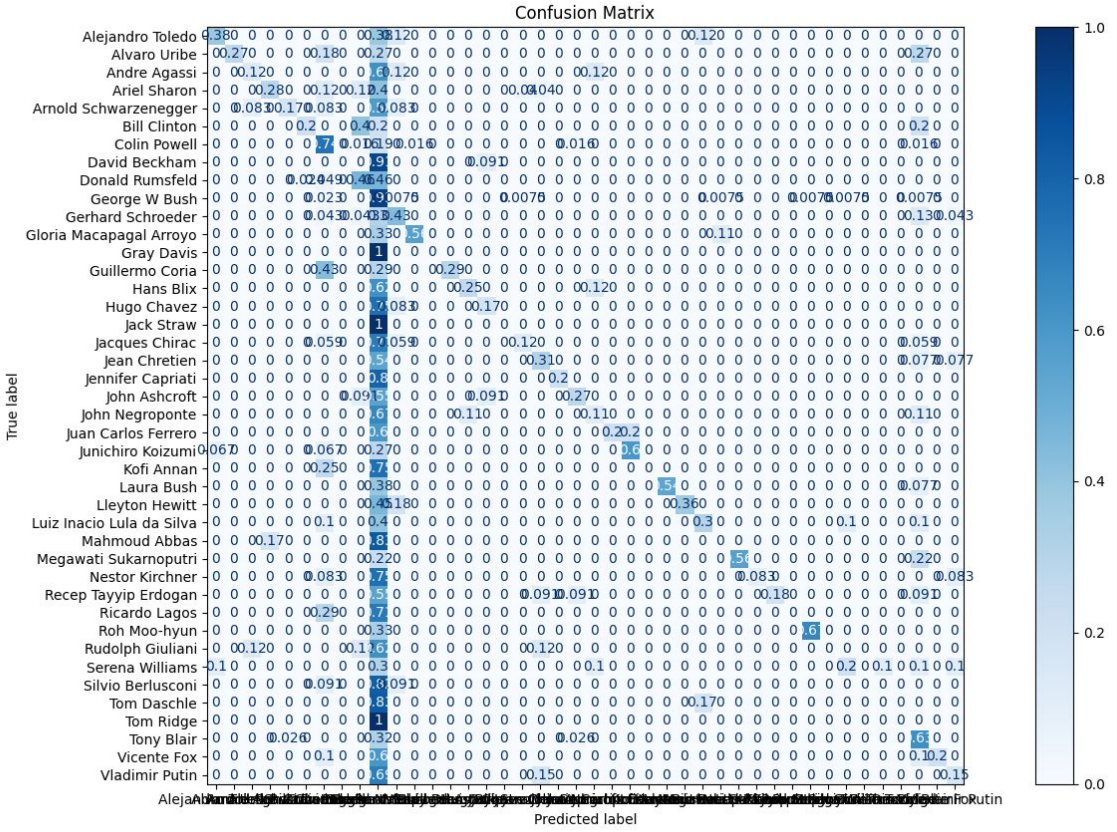




La matrice de confusion est encore plus brouillonne. Il y a davantage d'erreurs de prédiction. Toutefois, en observant les courbes ROC, nous arrivons à une AUC moyenne entre 0.90 et 0.93 ce qui n'est pas trop mal mais moins bien qu'avec le noyau RBF.

	precision	recall	f1-score	support
Alejandro Toledo	0.36	0.50	0.42	8
Alvaro Uribe	0.73	0.73	0.73	11
Andre Agassi	0.57	0.50	0.53	8
Ariel Sharon	0.62	0.52	0.57	25
Arnold Schwarzenegger	0.20	0.17	0.18	12
Bill Clinton	0.12	0.20	0.15	5
Colin Powell	0.69	0.73	0.71	62
David Beckham	0.83	0.45	0.59	11
Donald Rumsfeld	0.55	0.68	0.61	41
George W Bush	0.68	0.75	0.71	133
Gerhard Schroeder	0.42	0.43	0.43	23
Gloria Macapagal Arroyo	0.89	0.89	0.89	9
Gray Davis	0.00	0.00	0.00	3
Guillermo Coria	0.67	0.29	0.40	7
Hans Blix	0.43	0.38	0.40	8
Hugo Chavez	0.57	0.67	0.62	12
Jack Straw	0.33	0.50	0.40	2
Jacques Chirac	0.70	0.41	0.52	17
Jean Chretien	0.57	0.62	0.59	13
Jennifer Capriati	0.22	0.40	0.29	5
John Ashcroft	0.37	0.64	0.47	11
John Negroponte	0.50	0.56	0.53	9
Juan Carlos Ferrero	0.33	0.40	0.36	5
Junichiro Koizumi	0.76	0.87	0.81	15
Kofi Annan	0.67	0.50	0.57	4
Laura Bush	1.00	0.77	0.87	13
Lleyton Hewitt	0.46	0.55	0.50	11
Luiz Inacio Lula da Silva	0.50	0.50	0.50	10
Mahmoud Abbas	0.80	0.67	0.73	6
Megawati Sukarnoputri	1.00	0.78	0.88	9
Nestor Kirchner	0.50	0.42	0.45	12
Recep Tayyip Erdogan	0.80	0.36	0.50	11
Ricardo Lagos	0.57	0.57	0.57	7
Roh Moo-hyun	1.00	1.00	1.00	6
Rudolph Giuliani	0.00	0.00	0.00	8
Serena Williams	1.00	0.30	0.46	10
Silvio Berlusconi	0.25	0.18	0.21	11
Tom Daschle	0.20	0.17	0.18	6
Tom Ridge	0.67	0.29	0.40	7
Tony Blair	0.67	0.63	0.65	38
Vicente Fox	0.54	0.70	0.61	10
Vladimir Putin	0.38	0.23	0.29	13
accuracy			0.59	647
macro avg	0.55	0.50	0.51	647
weighted avg	0.61	0.59	0.59	647

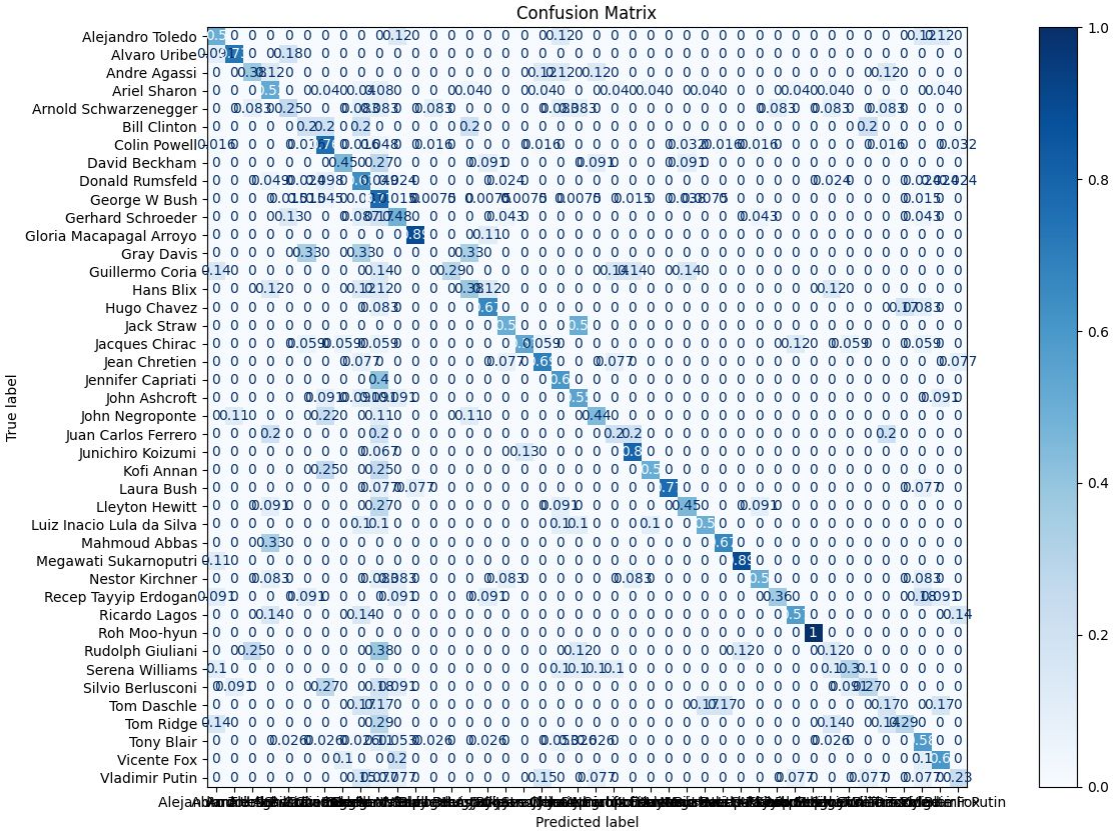
Noyau Polynomial (Degré 2)

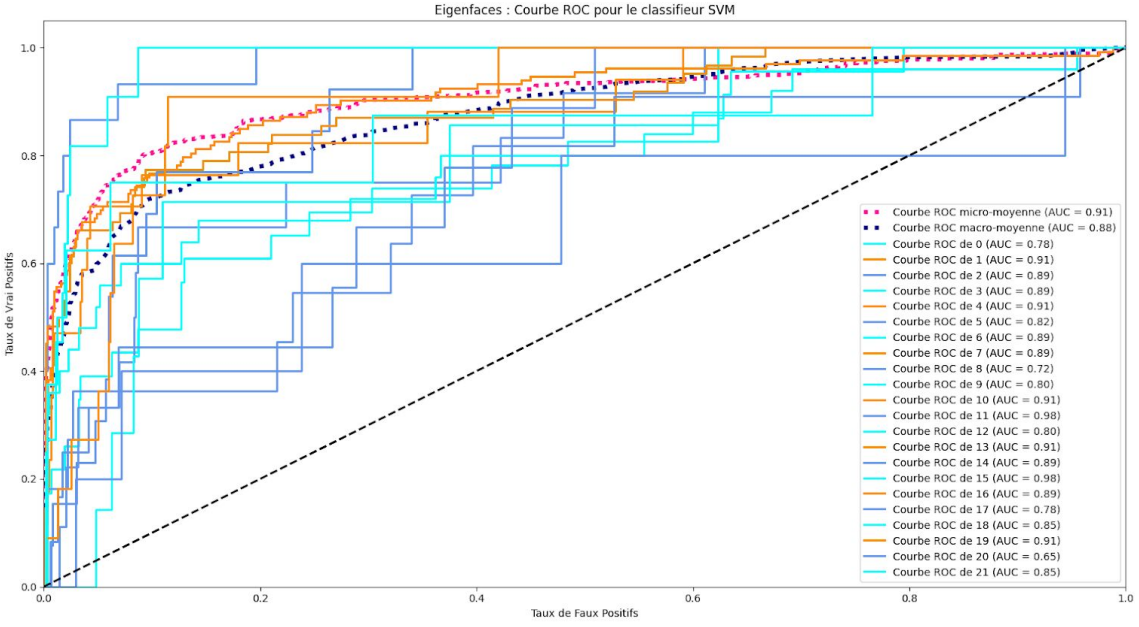


Ce n'est vraiment pas un résultat encourageant. Il est possible qu'en changeant le degré du polynôme, nous pourrions atteindre une meilleure mais avec un coût en terme de temps de calcul. Je n'ai malheureusement pas pu vérifier car cela prenait trop de temps. Et en allouant jusqu'à 7 threads pour accélérer les choses, le programme plantait.

	precision	recall	f1-score	support
Alejandro Toledo	0.60	0.38	0.46	8
Alvaro Uribe	1.00	0.27	0.43	11
Andre Agassi	0.33	0.12	0.18	8
Ariel Sharon	0.88	0.28	0.42	25
Arnold Schwarzenegger	0.67	0.17	0.27	12
Bill Clinton	0.50	0.20	0.29	5
Colin Powell	0.66	0.74	0.70	62
David Beckham	0.00	0.00	0.00	11
Donald Rumsfeld	0.68	0.46	0.55	41
George W Bush	0.34	0.93	0.49	133
Gerhard Schroeder	0.53	0.43	0.48	23
Gloria Macapagal Arroyo	0.83	0.56	0.67	9
Gray Davis	0.00	0.00	0.00	3
Guillermo Coria	1.00	0.29	0.44	7
Hans Blix	0.67	0.25	0.36	8
Hugo Chavez	0.50	0.17	0.25	12
Jack Straw	0.00	0.00	0.00	2
Jacques Chirac	0.50	0.12	0.19	17
Jean Chretien	0.44	0.31	0.36	13
Jennifer Capriati	1.00	0.20	0.33	5
John Ashcroft	0.50	0.27	0.35	11
John Negroponte	0.25	0.11	0.15	9
Juan Carlos Ferrero	1.00	0.20	0.33	5
Junichiro Koizumi	0.90	0.60	0.72	15
Kofi Annan	0.00	0.00	0.00	4
Laura Bush	1.00	0.54	0.70	13
Lleyton Hewitt	1.00	0.36	0.53	11
Luiz Inacio Lula da Silva	0.60	0.30	0.40	10
Mahmoud Abbas	0.00	0.00	0.00	6
Megawati Sukarnoputri	1.00	0.56	0.71	9
Nestor Kirchner	1.00	0.08	0.15	12
Recep Tayyip Erdogan	1.00	0.18	0.31	11
Ricardo Lagos	0.00	0.00	0.00	7
Roh Moo-hyun	0.80	0.67	0.73	6
Rudolph Giuliani	0.00	0.00	0.00	8
Serena Williams	0.50	0.20	0.29	10
Silvio Berlusconi	0.00	0.00	0.00	11
Tom Daschle	0.00	0.00	0.00	6
Tom Ridge	0.00	0.00	0.00	7
Tony Blair	0.56	0.63	0.59	38
Vicente Fox	1.00	0.20	0.33	10
Vladimir Putin	0.33	0.15	0.21	13
accuracy			0.47	647
macro avg	0.54	0.26	0.32	647
weighted avg	0.55	0.47	0.43	647

Noyau Sigmoïde



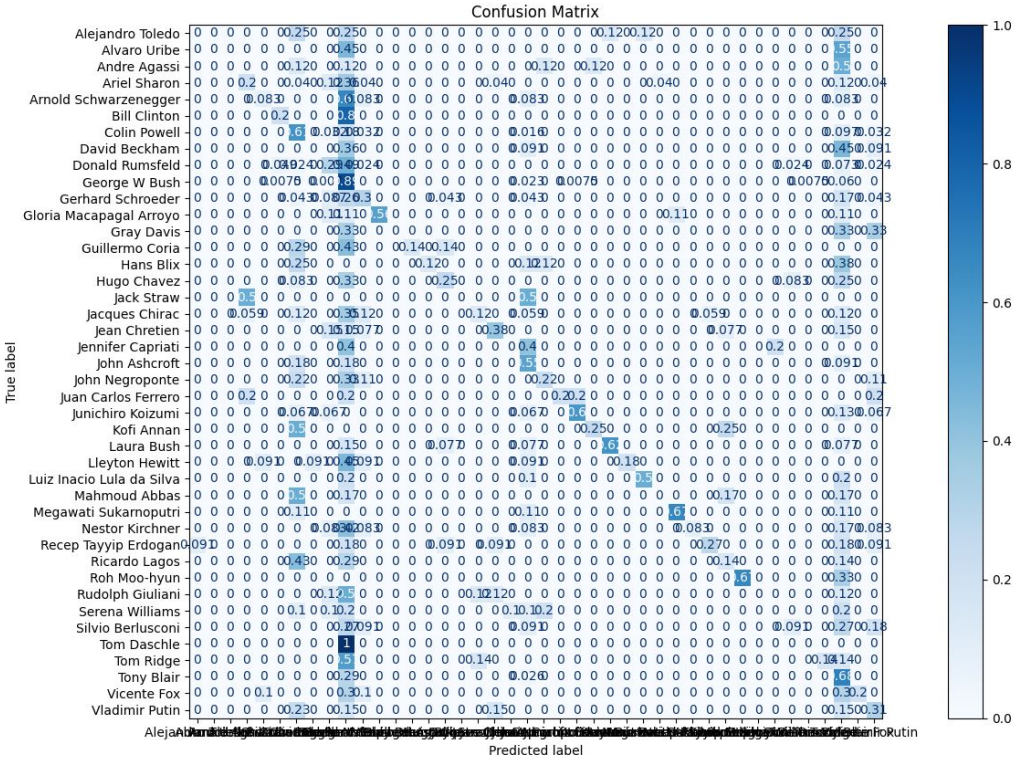


C’est très similaire aux résultats avec le noyau linéaire et cela reste moins bien que le noyau RBF.

	precision	recall	f1-score	support
Alejandro Toledo	0.36	0.50	0.42	8
Alvaro Uribe	0.80	0.73	0.76	11
Andre Agassi	0.50	0.38	0.43	8
Ariel Sharon	0.57	0.52	0.54	25
Arnold Schwarzenegger	0.27	0.25	0.26	12
Bill Clinton	0.12	0.20	0.15	5
Colin Powell	0.70	0.76	0.73	62
David Beckham	0.71	0.45	0.56	11
Donald Rumsfeld	0.57	0.63	0.60	41
George W Bush	0.70	0.77	0.73	133
Gerhard Schroeder	0.46	0.48	0.47	23
Gloria Macapagal Arroyo	0.89	0.89	0.89	9
Gray Davis	0.00	0.00	0.00	3
Guillermo Coria	1.00	0.29	0.44	7
Hans Blix	0.43	0.38	0.40	8
Hugo Chavez	0.57	0.67	0.62	12
Jack Straw	0.20	0.50	0.29	2
Jacques Chirac	0.75	0.53	0.62	17
Jean Chretien	0.60	0.69	0.64	13
Jennifer Capriati	0.27	0.60	0.37	5
John Ashcroft	0.46	0.55	0.50	11
John Negroponte	0.44	0.44	0.44	9
Juan Carlos Ferrero	0.20	0.20	0.20	5
Junichiro Koizumi	0.71	0.80	0.75	15
Kofi Annan	0.50	0.50	0.50	4
Laura Bush	1.00	0.77	0.87	13
Lleyton Hewitt	0.36	0.45	0.40	11
Luiz Inacio Lula da Silva	0.62	0.50	0.56	10
Mahmoud Abbas	0.67	0.67	0.67	6
Megawati Sukarnoputri	0.89	0.89	0.89	9
Nestor Kirchner	0.67	0.50	0.57	12
Recep Tayyip Erdogan	0.80	0.36	0.50	11
Ricardo Lagos	0.50	0.57	0.53	7
Roh Moo-hyun	1.00	1.00	1.00	6
Rudolph Giuliani	0.12	0.12	0.12	8
Serena Williams	0.60	0.30	0.40	10
Silvio Berlusconi	0.50	0.27	0.35	11
Tom Daschle	0.17	0.17	0.17	6
Tom Ridge	0.50	0.29	0.36	7
Tony Blair	0.63	0.58	0.60	38
Vicente Fox	0.50	0.60	0.55	10
Vladimir Putin	0.38	0.23	0.29	13
accuracy			0.60	647
macro avg	0.54	0.50	0.50	647
weighted avg	0.61	0.60	0.59	647

K Nearest Neighbors

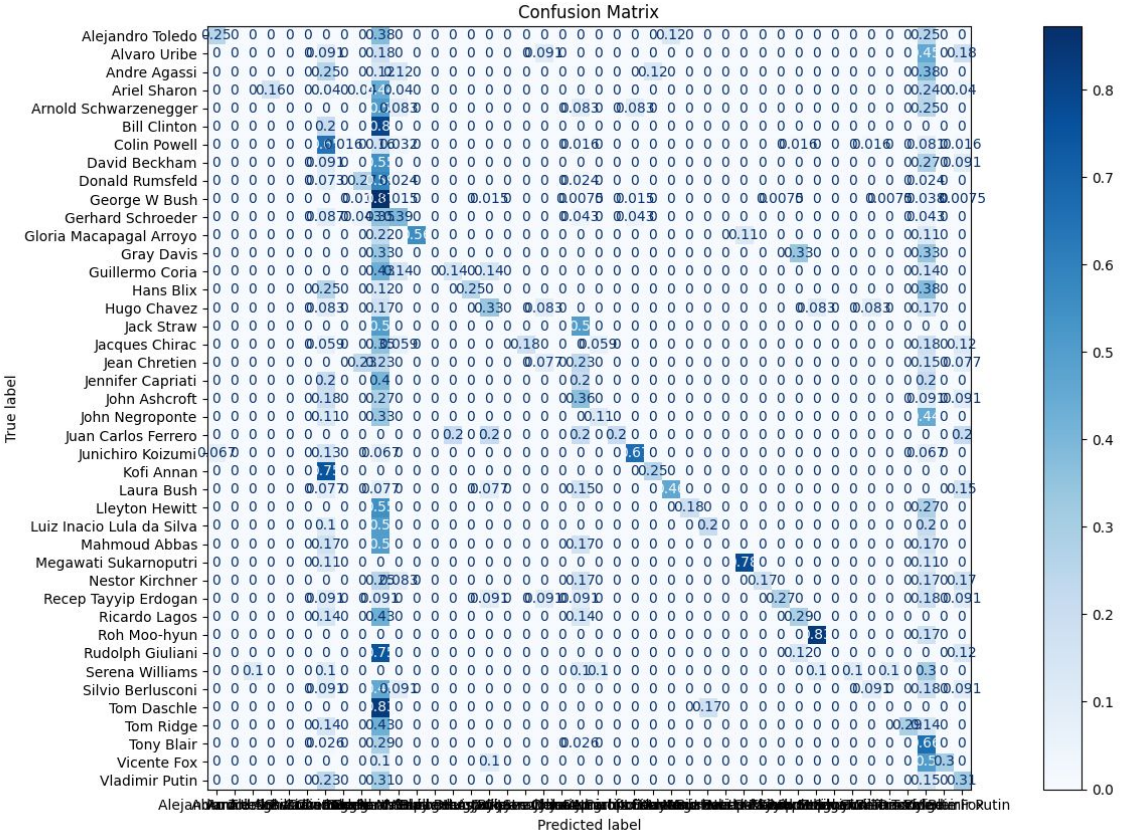
Avec Distance Euclidienne



L'allure de la matrice de confusion donne déjà un certain avant-goût. Ce n'est vraiment pas terrible.

	precision	recall	f1-score	support
Alejandro Toledo	0.00	0.00	0.00	8
Alvaro Uribe	0.00	0.00	0.00	11
Andre Agassi	0.00	0.00	0.00	8
Ariel Sharon	0.62	0.20	0.30	25
Arnold Schwarzenegger	0.33	0.08	0.13	12
Bill Clinton	0.25	0.20	0.22	5
Colin Powell	0.55	0.61	0.58	62
David Beckham	0.00	0.00	0.00	11
Donald Rumsfeld	0.44	0.29	0.35	41
George W Bush	0.44	0.89	0.59	133
Gerhard Schroeder	0.35	0.30	0.33	23
Gloria Macapagal Arroyo	1.00	0.56	0.71	9
Gray Davis	0.00	0.00	0.00	3
Guillermo Coria	1.00	0.14	0.25	7
Hans Blix	1.00	0.12	0.22	8
Hugo Chavez	0.43	0.25	0.32	12
Jack Straw	0.00	0.00	0.00	2
Jacques Chirac	0.50	0.12	0.19	17
Jean Chretien	0.50	0.38	0.43	13
Jennifer Capriati	0.00	0.00	0.00	5
John Ashcroft	0.22	0.55	0.32	11
John Negroponte	0.33	0.22	0.27	9
Juan Carlos Ferrero	1.00	0.20	0.33	5
Junichiro Koizumi	0.82	0.60	0.69	15
Kofi Annan	0.50	0.25	0.33	4
Laura Bush	0.89	0.62	0.73	13
Lleyton Hewitt	1.00	0.18	0.31	11
Luiz Inacio Lula da Silva	0.83	0.50	0.62	10
Mahmoud Abbas	0.00	0.00	0.00	6
Megawati Sukarnoputri	0.86	0.67	0.75	9
Nestor Kirchner	1.00	0.08	0.15	12
Recep Tayyip Erdogan	0.75	0.27	0.40	11
Ricardo Lagos	0.25	0.14	0.18	7
Roh Moo-hyun	1.00	0.67	0.80	6
Rudolph Giuliani	0.00	0.00	0.00	8
Serena Williams	0.00	0.00	0.00	10
Silvio Berlusconi	0.33	0.09	0.14	11
Tom Daschle	0.00	0.00	0.00	6
Tom Ridge	1.00	0.14	0.25	7
Tony Blair	0.24	0.68	0.36	38
Vicente Fox	1.00	0.20	0.33	10
Vladimir Putin	0.22	0.31	0.26	13
accuracy			0.44	647
macro avg	0.47	0.25	0.28	647
weighted avg	0.48	0.44	0.39	647

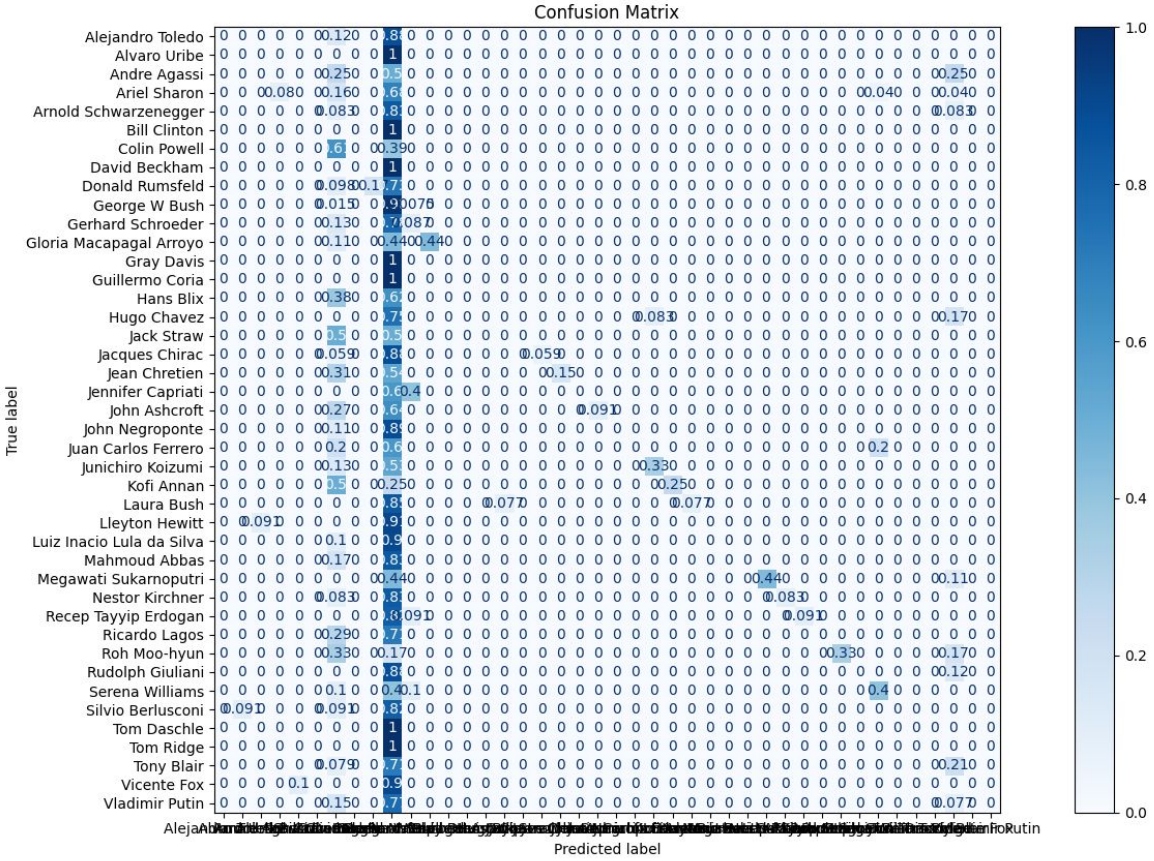
Avec Distance de Manhattan



Le résultat est similaire au précédent : mauvais. Il n'y a pas de réelle diagonale donc les prédictions sont erronées.

	precision	recall	f1-score	support
Alejandro Toledo	0.67	0.25	0.36	8
Alvaro Uribe	0.00	0.00	0.00	11
Andre Agassi	0.00	0.00	0.00	8
Ariel Sharon	1.00	0.16	0.28	25
Arnold Schwarzenegger	0.00	0.00	0.00	12
Bill Clinton	0.00	0.00	0.00	5
Colin Powell	0.52	0.65	0.58	62
David Beckham	0.00	0.00	0.00	11
Donald Rumsfeld	0.61	0.27	0.37	41
George W Bush	0.42	0.87	0.57	133
Gerhard Schroeder	0.43	0.39	0.41	23
Gloria Macapagal Arroyo	1.00	0.56	0.71	9
Gray Davis	0.00	0.00	0.00	3
Guillermo Coria	0.50	0.14	0.22	7
Hans Blix	1.00	0.25	0.40	8
Hugo Chavez	0.36	0.33	0.35	12
Jack Straw	0.00	0.00	0.00	2
Jacques Chirac	1.00	0.18	0.30	17
Jean Chretien	0.25	0.08	0.12	13
Jennifer Capriati	0.00	0.00	0.00	5
John Ashcroft	0.17	0.36	0.23	11
John Negroponte	0.33	0.11	0.17	9
Juan Carlos Ferrero	1.00	0.20	0.33	5
Junichiro Koizumi	0.71	0.67	0.69	15
Kofi Annan	0.50	0.25	0.33	4
Laura Bush	0.86	0.46	0.60	13
Lleyton Hewitt	1.00	0.18	0.31	11
Luiz Inacio Lula da Silva	0.67	0.20	0.31	10
Mahmoud Abbas	0.00	0.00	0.00	6
Megawati Sukarnoputri	0.88	0.78	0.82	9
Nestor Kirchner	1.00	0.17	0.29	12
Recep Tayyip Erdogan	0.75	0.27	0.40	11
Ricardo Lagos	0.40	0.29	0.33	7
Roh Moo-hyun	0.71	0.83	0.77	6
Rudolph Giuliani	0.00	0.00	0.00	8
Serena Williams	1.00	0.10	0.18	10
Silvio Berlusconi	0.33	0.09	0.14	11
Tom Daschle	0.00	0.00	0.00	6
Tom Ridge	1.00	0.29	0.44	7
Tony Blair	0.24	0.66	0.35	38
Vicente Fox	1.00	0.30	0.46	10
Vladimir Putin	0.18	0.31	0.23	13
accuracy			0.43	647
macro avg	0.49	0.25	0.29	647
weighted avg	0.51	0.43	0.38	647

Random Forest



Il n'y a aucune diagonale dans cette matrice de confusion, ce qui signifie qu'il y a énormément d'erreurs de prédiction. Bref, c'est inexploitable malgré le fait que le classifieur Random Forest soit meilleur que le Decision Trees.

	precision	recall	f1-score	support
Alejandro Toledo	0.00	0.00	0.00	8
Alvaro Uribe	0.00	0.00	0.00	11
Andre Agassi	0.00	0.00	0.00	8
Ariel Sharon	1.00	0.08	0.15	25
Arnold Schwarzenegger	0.00	0.00	0.00	12
Bill Clinton	0.00	0.00	0.00	5
Colin Powell	0.43	0.61	0.51	62
David Beckham	0.00	0.00	0.00	11
Donald Rumsfeld	1.00	0.17	0.29	41
George W Bush	0.26	0.98	0.42	133
Gerhard Schroeder	0.29	0.09	0.13	23
Gloria Macapagal Arroyo	1.00	0.44	0.62	9
Gray Davis	0.00	0.00	0.00	3
Guillermo Coria	0.00	0.00	0.00	7
Hans Blix	0.00	0.00	0.00	8
Hugo Chavez	0.00	0.00	0.00	12
Jack Straw	0.00	0.00	0.00	2
Jacques Chirac	1.00	0.06	0.11	17
Jean Chretien	1.00	0.15	0.27	13
Jennifer Capriati	0.00	0.00	0.00	5
John Ashcroft	1.00	0.09	0.17	11
John Negroponte	0.00	0.00	0.00	9
Juan Carlos Ferrero	0.00	0.00	0.00	5
Junichiro Koizumi	0.83	0.33	0.48	15
Kofi Annan	1.00	0.25	0.40	4
Laura Bush	1.00	0.08	0.14	13
Lleyton Hewitt	0.00	0.00	0.00	11
Luiz Inacio Lula da Silva	0.00	0.00	0.00	10
Mahmoud Abbas	0.00	0.00	0.00	6
Megawati Sukarnoputri	1.00	0.44	0.62	9
Nestor Kirchner	1.00	0.08	0.15	12
Recep Tayyip Erdogan	1.00	0.09	0.17	11
Ricardo Lagos	0.00	0.00	0.00	7
Roh Moo-hyun	1.00	0.33	0.50	6
Rudolph Giuliani	0.00	0.00	0.00	8
Serena Williams	0.67	0.40	0.50	10
Silvio Berlusconi	0.00	0.00	0.00	11
Tom Daschle	0.00	0.00	0.00	6
Tom Ridge	0.00	0.00	0.00	7
Tony Blair	0.44	0.21	0.29	38
Vicente Fox	0.00	0.00	0.00	10
Vladimir Putin	0.00	0.00	0.00	13
accuracy			0.33	647
macro avg	0.36	0.12	0.14	647
weighted avg	0.43	0.33	0.24	647

5. Conclusion

En conclusion, nous pouvons très clairement affirmer que SVM a les meilleurs résultats au niveau des tests de prédiction parmi les classifieurs testés. Toutefois, le choix du noyau peut permettre d'améliorer les résultats : parmi ceux testés, le noyau RBF (par défaut) est le meilleur.