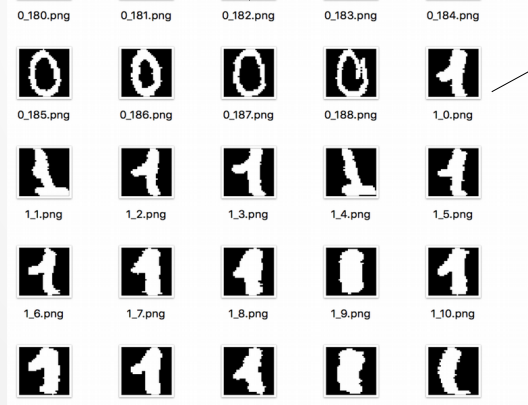


Chiffre Reconnnaissance



0000000000001111100000000000000000
0000000000001111110000000000000000
0000000000111111111000000000000000
0000000001111111111100000000000000
0000000011111111111110000000000000
0000000111111101111111000000000000
0000000111111000111111100000000000
0000000111111000011111100000000000
0000001111110000011111100000000000
0000001111110000001111100000000000
0000001111110000000111110000000000
0000001111110000000011111000000000
0000001111110000000001111100000000
0000001111110000000000111110000000
0000001111110000000000011111000000
0000001111110000000000001111100000
00000011111100000000000001111100000
000000111111000000000000001111100000
0000001111110000000000000001111100000
0000000011111000000111111100000000
0000000001111100000011111110000000
0000000000111110000011111110000000
0000000000111111000111111100000000
0000000000111111111111111000000000
0000000000111111111111111000000000
0000000000111111111111111000000000

Chiffre Reconnaissance

- Gagner des données ([kaggle.com](https://www.kaggle.com))
- Analyser des données et préproccession (scaling et PCA)
- Générer modèle(KNN et SVM)
- Evaluation

Dataset

- Source : Kaggle.com
- 70MB+ dataset, parmi lesquels on a extrait 42000 échantillons, et 785 features par échantillon.
- Each image is 28 pixels in height and 28 pixels in width, for a total of $28 \times 28 = 784$ pixels in total. Each pixel-value is an integer between 0 and 255.

Pre-processing

- Normalisation de données(scaling)
- $X = \frac{X - \mu}{\sigma}$
- Réduction de dimension
- PCA (Principal Component Analysis)

PCA(Analyse en composantes principales)

- Analyser des données
- Calcule le variance pour chaque dimension
- Mise en ordre (max \rightarrow min)
- Faire l'addition jusqu'au somme de variances représente plus 95%
- Sauvegarder les premiers 320 dimensions , Couper les restes dimensions

```
wp@wp-MI: ~/Documents/Fouille de donnees/12.12
analyse data....
main: 1 ,      sqrs: 10.04 % ,    sum : 10.04 %
main: 2 ,      sqrs: 6.45 % ,    sum : 16.49 %
main: 3 ,      sqrs: 4.86 % ,    sum : 21.34 %
main: 4 ,      sqrs: 3.64 % ,    sum : 24.98 %
main: 5 ,      sqrs: 2.51 % ,    sum : 27.49 %
main: 6 ,      sqrs: 2.32 % ,    sum : 29.81 %
main: 7 ,      sqrs: 1.98 % ,    sum : 31.78 %
main: 8 ,      sqrs: 1.84 % ,    sum : 33.62 %
main: 9 ,      sqrs: 1.70 % ,    sum : 35.33 %
main: 10 ,     sqrs: 1.55 % ,    sum : 36.88 %
main: 11 ,     sqrs: 1.54 % ,    sum : 38.43 %
main: 12 ,     sqrs: 1.41 % ,    sum : 39.83 %
main: 13 ,     sqrs: 1.39 % ,    sum : 41.22 %
main: 14 ,     sqrs: 1.27 % ,    sum : 42.49 %
main: 15 ,     sqrs: 1.23 % ,    sum : 43.71 %
main: 16 ,     sqrs: 1.13 % ,    sum : 44.84 %
main: 17 ,     sqrs: 1.09 % ,    sum : 45.93 %
main: 18 ,     sqrs: 0.99 % ,    sum : 46.92 %
main: 19 ,     sqrs: 0.98 % ,    sum : 47.90 %
main: 20 ,     sqrs: 0.95 % ,    sum : 48.85 %
main: 21 ,     sqrs: 0.91 % ,    sum : 49.76 %
main: 22 ,     sqrs: 0.88 % ,    sum : 50.64 %
main: 23 ,     sqrs: 0.83 % ,    sum : 51.46 %
main: 24 ,     sqrs: 0.81 % ,    sum : 52.28 %
main: 25 ,     sqrs: 0.75 % ,    sum : 53.03 %
main: 26 ,     sqrs: 0.74 % ,    sum : 53.77 %
main: 27 ,     sqrs: 0.72 % ,    sum : 54.49 %
```

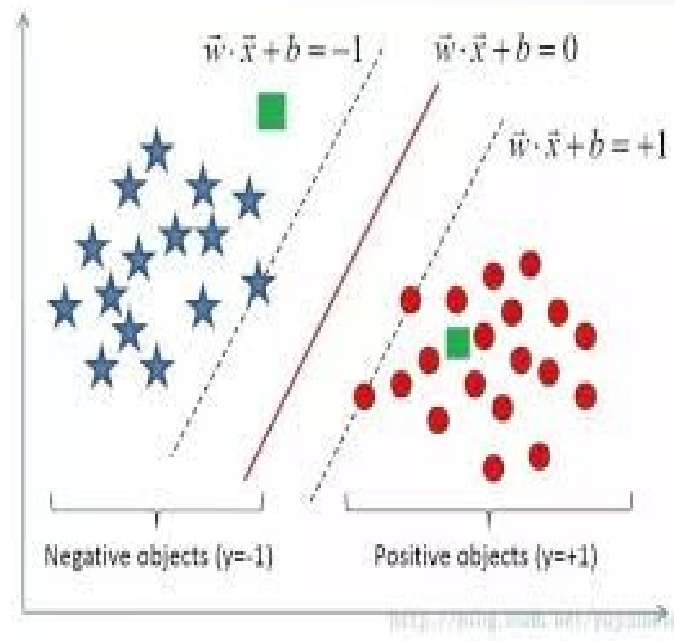
Principe de PCA

- Le tableau X (M lignes N columns)
- Covariance matrix $C = \frac{1}{m} XX^T$
- Eigenvector et eigenvalue par SVD (singular value decomposition)
 $A = UDV^T$,
- Mise en ordre pour eigenvalue, sauvegarder les premiers k lignes comme un nouveau matrix P.
- $Y = PX$ est le resultat , et qui a eu k dimensions

$$\begin{aligned} D &= \frac{1}{m} Y^T Y \\ &= \frac{1}{m} (PX)^T (PX) \\ &= \frac{1}{m} X^T P^T P X \\ &= X^T \left(\frac{1}{m} P^T P \right) X \\ &= X^T C X \end{aligned}$$

SVM (Support Vector Machine)

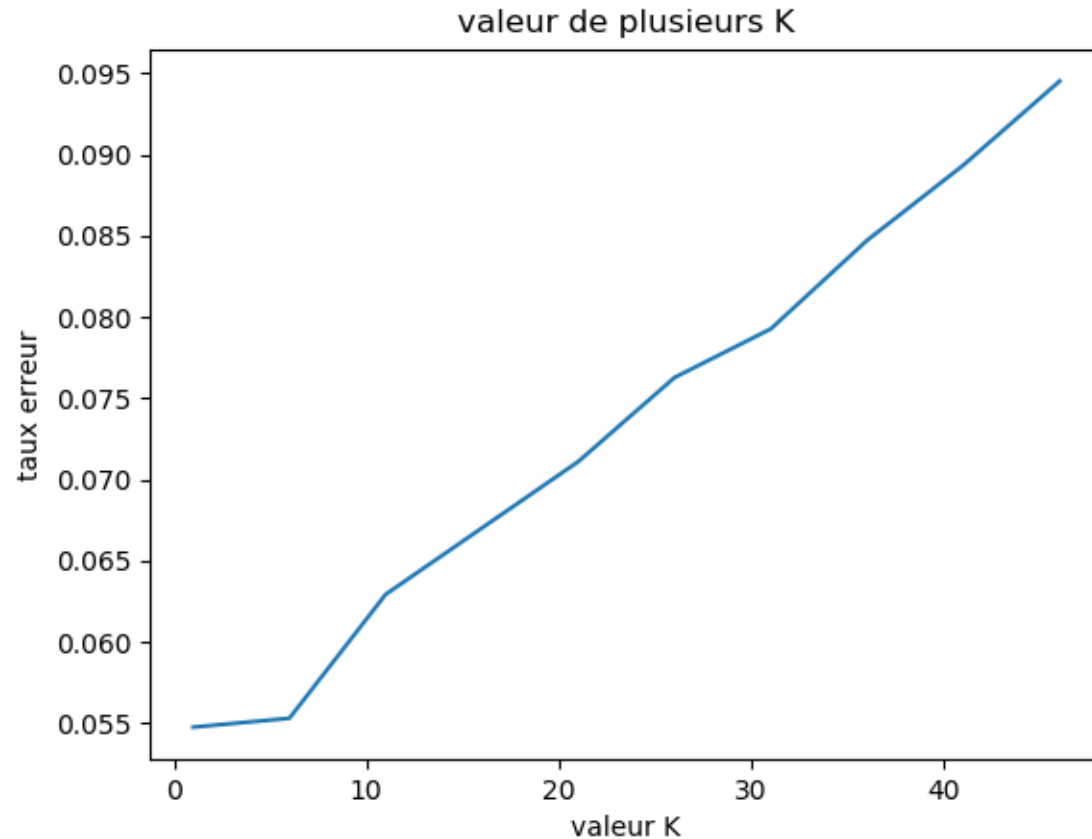
- Kernel : Linear
- Linear separability
- Haute-dimensions
- Le tableau est clairsemé
- Économique le temps , et obtenir de bons résultats

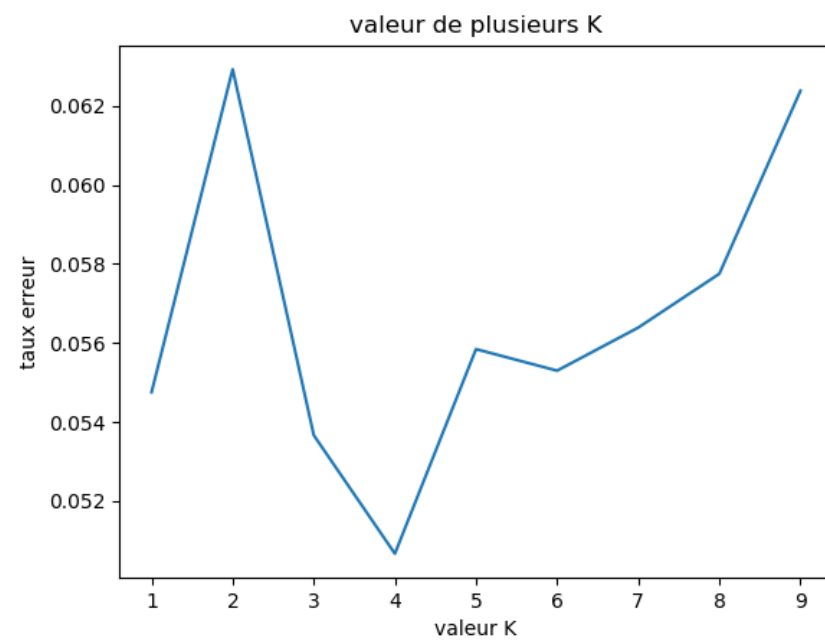


Entrainement / Validation

- 70% pour l'entrainement
- 30% pour les tests

K nearest neighbors





Précision de modèle

- KNN : 0.9493326069190956
- SVM : 0.9588667937891583

Retropection

- Séparation de Entrainement / Validation
 - Mal séparer les 2 ensembles avec la taille espéré
 - On aurait du utiliser `train_test_split` de sklearn

- Pour KNN
 - Manque validation croisé
 - Améliorer la fonction de distance
 - Avec pondération peut-être