

Projet A3 IA

DIGUER Louison : louison.diguer@isen-ouest.yncrea.fr

RADIN Alexandre : alexandre.radin@isen-ouest.yncrea.fr

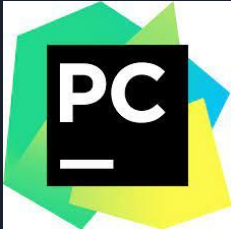
SOYDEMIR Antoine : antoine.soydemir@isen-ouest.yncrea.fr



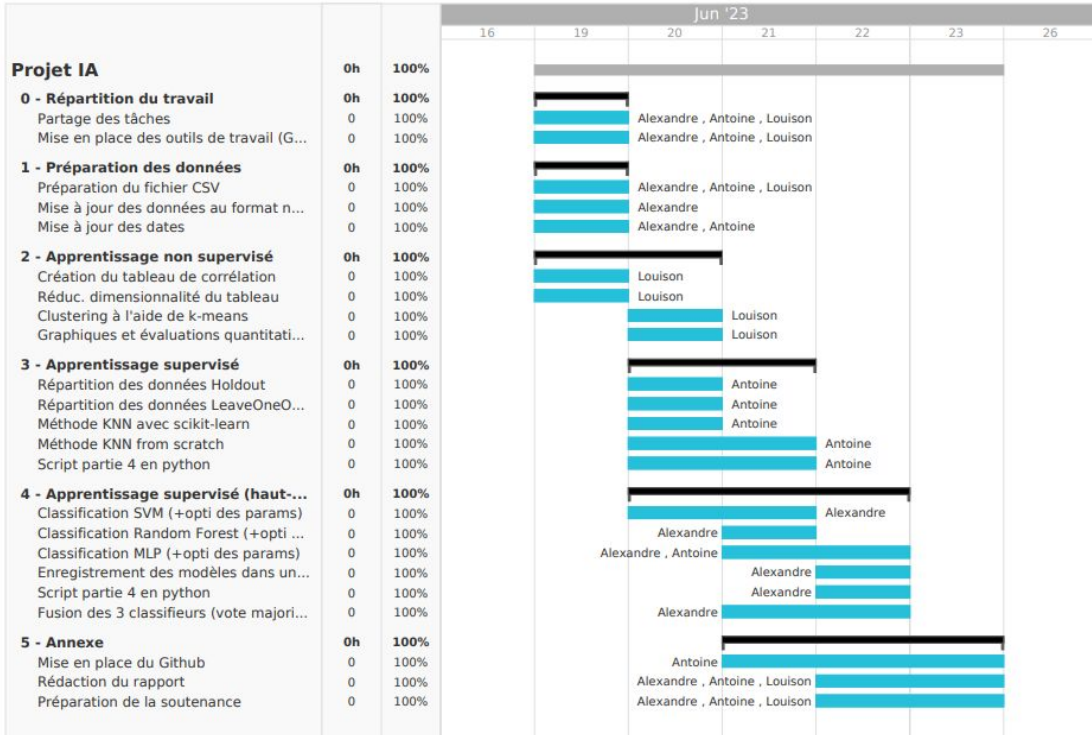
Sommaire

- 1) Environnement et organisation du travail
- 2) Découverte et préparation des données
- 3) Apprentissage non supervisé
- 4) Apprentissage supervisé

1) Environnement et organisation du travail



teamgantt
Created with Free Edition



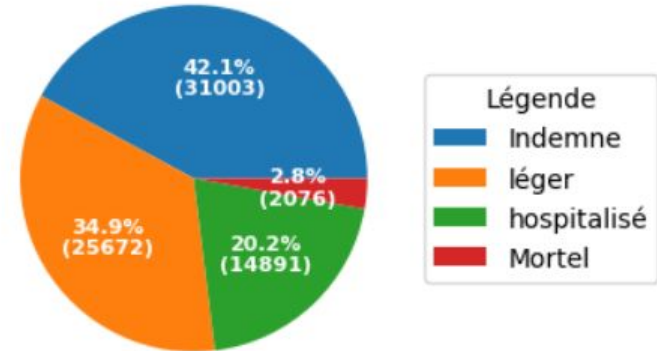
2) Découverte et préparation des données

Le nombre d'instances dans la base de données est de : 73643
Le nombre de features est de : 22

Le nombre d'accident indemne est de : 31004
Le nombre d'accident blessé léger est de : 25672
Le nombre d'accident blessé hospitalisé est de : 14891
Le nombre d'accident mortel est de : 2076

type date to date_string

Taux de gravité par accident



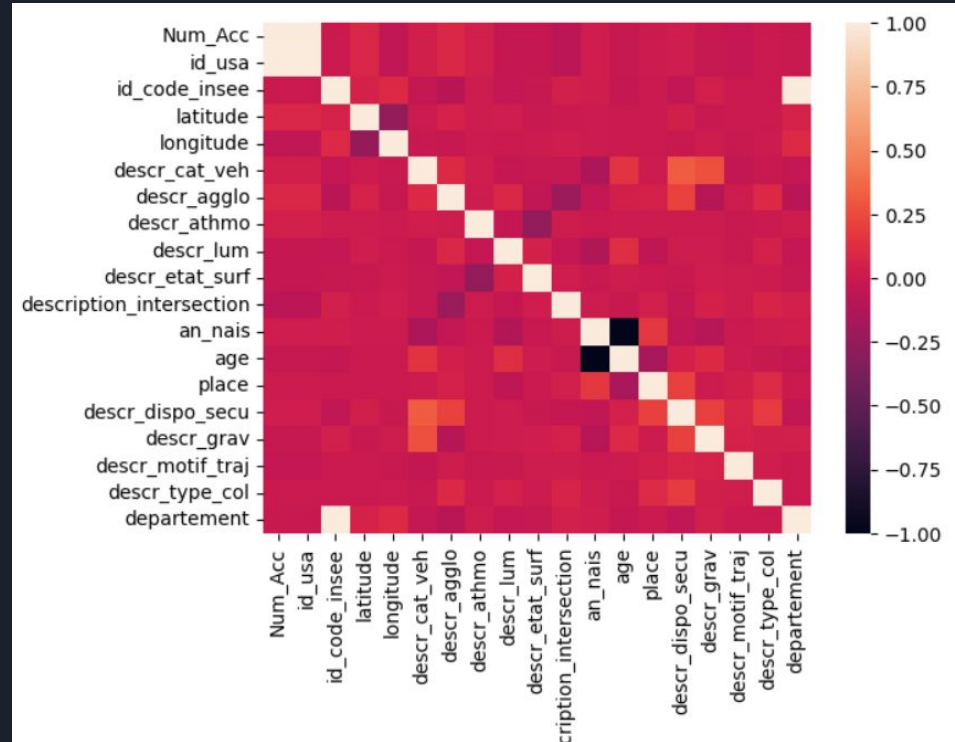
3) Apprentissage non-supervisé (réduction de données)

```
### TEST CORRELATION > 0.75 ###
```

```
Num_Acc 0.9999959213788927  
id_usa 0.9999959213788927  
id_code_insee 0.9999754324787202  
an_nais -1.0000000000000004  
age -1.0000000000000004  
departement 0.9999754324787202
```

```
### TEST CORRELATION > 0.5 ###
```

```
Num_Acc 0.9999959213788927  
id_usa 0.9999959213788927  
id_code_insee 0.9999754324787202  
an_nais -1.0000000000000004  
age -1.0000000000000004  
departement 0.9999754324787202
```



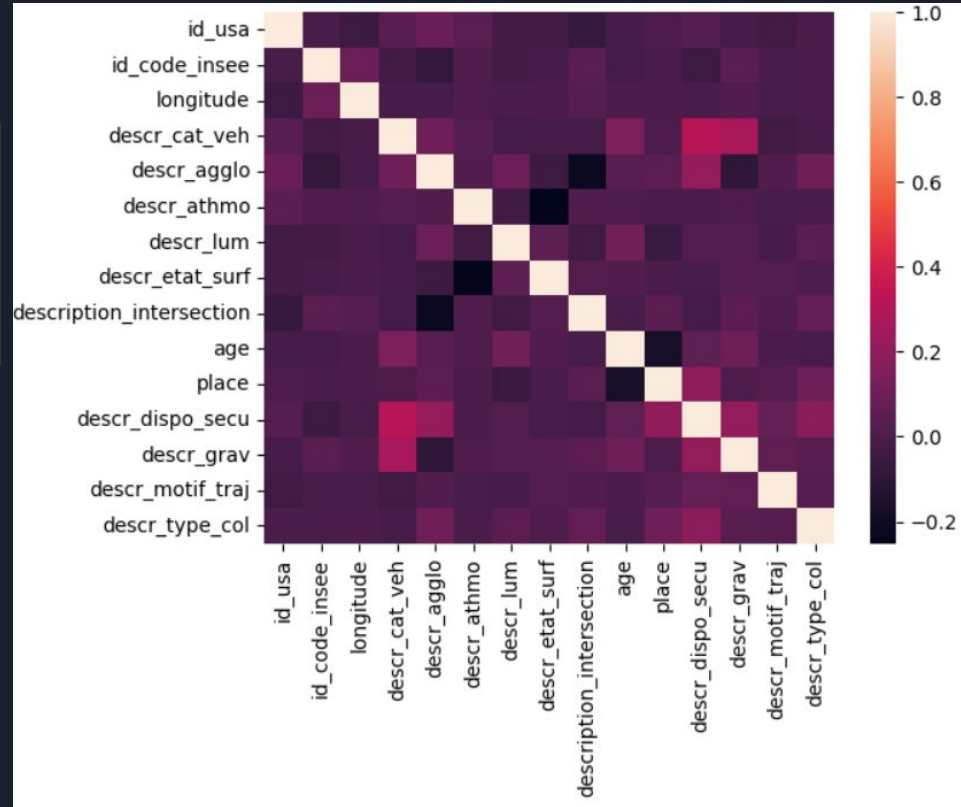
3) Apprentissage non-supervisé (réduction de données)

Colonne Num_Acc supprimée
Colonne departement supprimée
Colonne latitude supprimée
Colonne an_nais supprimée
Le pourcentage de réduction est de : 18.18 %

```
### TEST CORRELATION > 0.75 ###
```

```
### TEST CORRELATION > 0.5 ###
```

Nous n'avons finalement plus de corrélation



3) Apprentissage non-supervisé (clustering)

POO \rightarrow KMeans_from_scratch(n_clusters, max_iter, distance)
attributs : n_clusters, max_iter, distance, centroides, labels
méthodes : constructeur, fit, fit_predict, assignation_labels,
initialisation_centroides, update_centroides, calculs de distance




FS euclidean



FS haversine



sklearn




3) Apprentissage non-supervisé (Évaluation quantitative)

Tableau des silhouette score en fonction de l'algorithme utilisé

KMeans\cluster	5	10	15	20	25	30	35	40	45	50
FS euclidian	0.524034	0.496133	0.648167	0.585722	0.570483	0.608618	0.658289	0.63923	0.673448	0.638631
FS manhattan	0.567909	0.616585	0.583598	0.616976	0.579025	0.633244	0.621491	0.714025	0.69473	0.683205
FS haversine	0.409283	0.493031	0.47644	0.493608	0.531936	0.527755	0.540506	0.576528	0.529105	0.609056
sklearn	0.680573	0.72884	0.731803	0.746002	0.771416	0.78369	0.799725	0.81814	0.826161	0.842601

silhouette score = Moyenne(cohésion-séparation)

Domaine de variation : [-1 ; 1]



3) Apprentissage non-supervisé (Évaluation quantitative)

Tableau des calinski_harabasz_score en fonction de l'algorithme utilisé

KMeans\cluster	5	10	15	20	25	30	35	40	45	50
FS euclidian	43746.5	32491.5	34727.7	81319.1	57098.6	22536	68888.6	25083.2	69264.3	66530.8
FS manhattan	45750.1	42321.3	26013.7	64759.4	51345.9	22029.7	17418.1	60286.9	53748.7	17534.3
FS haversine	59245.6	35985.4	20060.4	27752.5	23227.3	19948.9	22771.1	18061.2	16023.8	49990.2
sklearn	101116	182980	236614	281621	350178	390982	444331	504735	562744	623050

Calinski-Harabasz index = variance intergroupe/variance intragroupe

Domaine de variation : $[0 ; +\infty[$



3) Apprentissage non-supervisé (Évaluation quantitative)

Tableau des davies_bouldin_score en fonction de l'algorithme utilisé

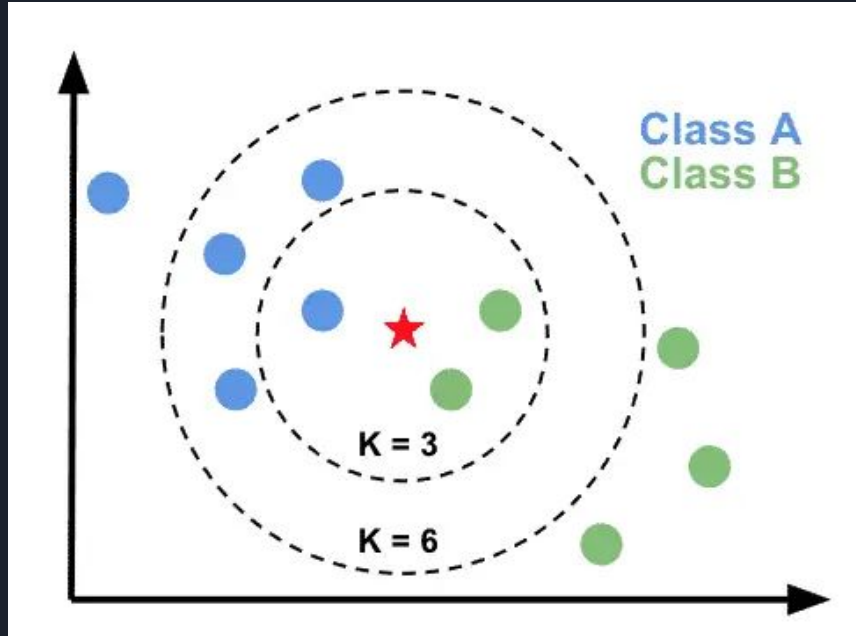
KMeans\cluster	5	10	15	20	25	30	35	40	45	50
FS euclidian	0.774011	0.811766	0.743962	0.953988	1.00938	0.977539	0.831136	0.821504	1.0768	1.31658
FS manhattan	0.92844	0.990488	1.27506	1.0693	0.835342	1.01699	0.651721	1.02586	0.900541	0.913095
FS haversine	0.772077	1.02349	0.871202	0.882489	0.888187	1.00182	1.19488	1.30921	0.947916	0.896864
sklearn	0.545981	0.47464	0.581558	0.555695	0.55238	0.579249	0.52346	0.549306	0.452548	0.461527

Davies-Bouldin index= $\text{moyenne}(\text{distance_point_centre} / \text{distance_centre_centre})$

Domaine de variation : $[0 ; +\infty[$

4) Apprentissage supervisé

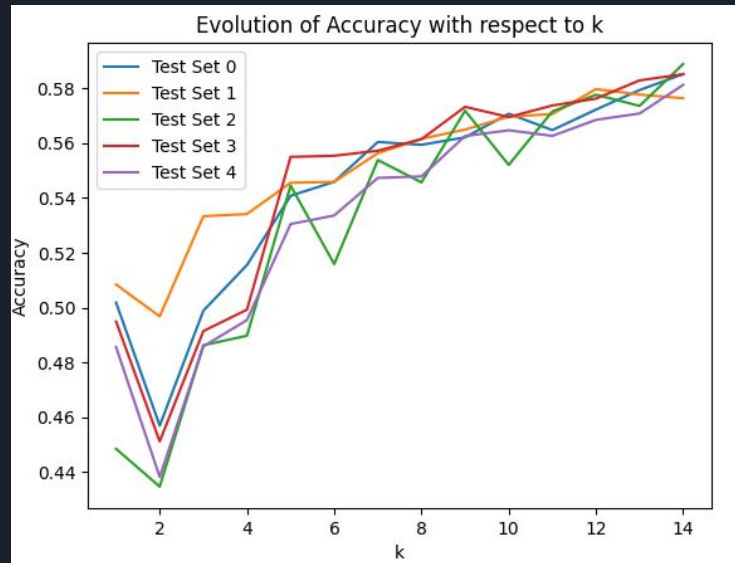
Méthode Knn (K-nearest neighbors) :



- étoile rouge : échantillon de test
- Le choix de la classe est effectué en fonction du plus grand nombre de voisins.

4) Résultats à partir de Sklearn

Répartition Holdout



Moyenne des performances : 0.5412761995286266

Répartition Leave-one-out

- Échantillon de 10% de la base de données.
- Consomme beaucoup plus de ressources.

Leave-One-Out Cross Validation



Moyenne des performances : 0.5178207640519222



4) Résultats from “scratch”

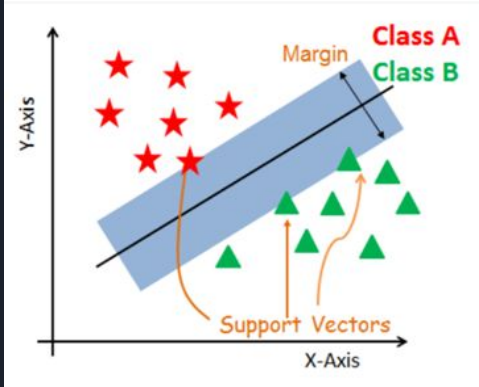
Étapes de la méthode :

- Création d'une classe Knn (avec les différents paramètres : nombre de voisins, type de distance..)
- Calcul des distances (entre 2 vecteurs de données)
- Entraînement du modèle (méthode .fit())
- Prédiction sur l'ensemble de test

```
Metric: euclidean, accuracy: 53.768 %  
Metric: manhattan, accuracy: 53.734 %  
Metric: minkowski, accuracy: 53.768 %
```

- Précision inférieure aux résultats de la bibliothèque Sklearn.
- Temps d'exécution beaucoup plus important.

4) Apprentissage supervisé (SVM)



[Scikit-learn SVM Tutorial with Python \(Support Vector Machines\) | DataCamp](#)

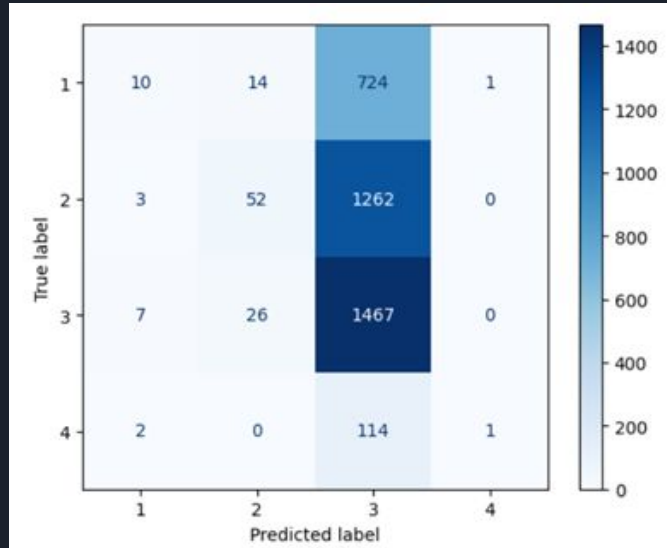
3 paramètres à régler :

- kernels
- C
- Gamma

```
SVC : Best parameters : {'C': 1, 'gamma': 0.01}  
SVC : Best estimator : SVC(C=1, gamma=0.01)
```

```
SVC accuracy score (hyper parameter tuning) : 0.41542221015476516
```

4) Apprentissage supervisé (SVM)



	precision	recall	f1-score	support
1	0.45	0.01	0.03	749
2	0.57	0.04	0.07	1317
3	0.41	0.98	0.58	1500
4	0.50	0.01	0.02	117
accuracy			0.42	3683
macro avg	0.48	0.26	0.17	3683
weighted avg	0.48	0.42	0.27	3683



4) Apprentissage supervisé (Random Forest)

2 paramètres à régler :

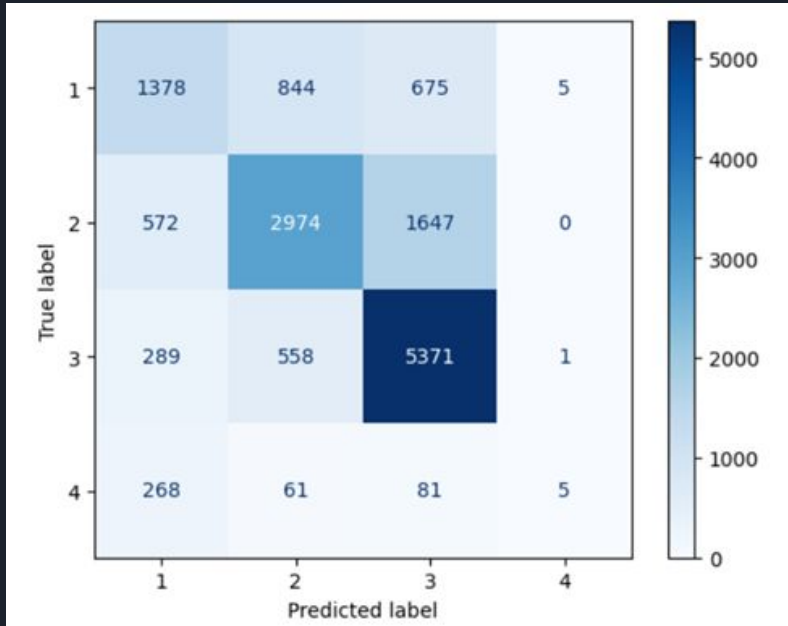
- `n_estimators`
- `max_depth`

```
Position : 1 - Score : 0.649302 (+/-0.002055) for {'max_depth': 20, 'n_estimators': 500}
```

```
Random Forest : Best parameters : {'max_depth': 20, 'n_estimators': 500}  
Random Forest : Best estimator : RandomForestClassifier(max_depth=20, n_estimators=500)
```

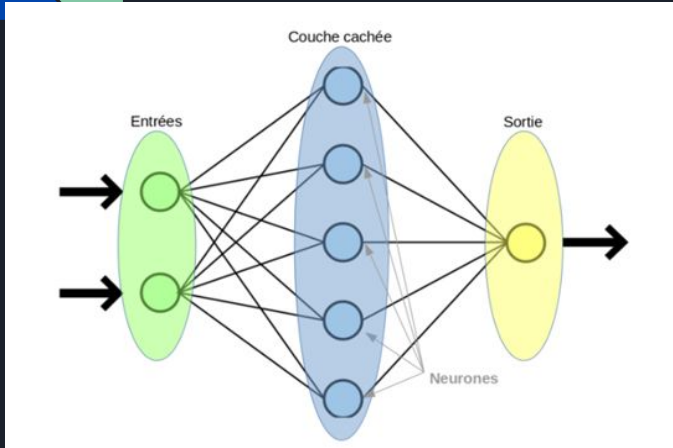
```
Random Forest accuracy score (hyper parameter tuning) : 0.6604657478443886
```

4) Apprentissage supervisé (Random Forest)



	precision	recall	f1-score	support
1	0.55	0.47	0.51	2902
2	0.67	0.57	0.62	5193
3	0.69	0.86	0.77	6219
4	0.45	0.01	0.02	415
accuracy			0.66	14729
macro avg	0.59	0.48	0.48	14729
weighted avg	0.65	0.66	0.64	14729

4) Apprentissage supervisé (MLP)



Fonctionnement du perceptron multicouche – Bloom Magazine
(home.blog)

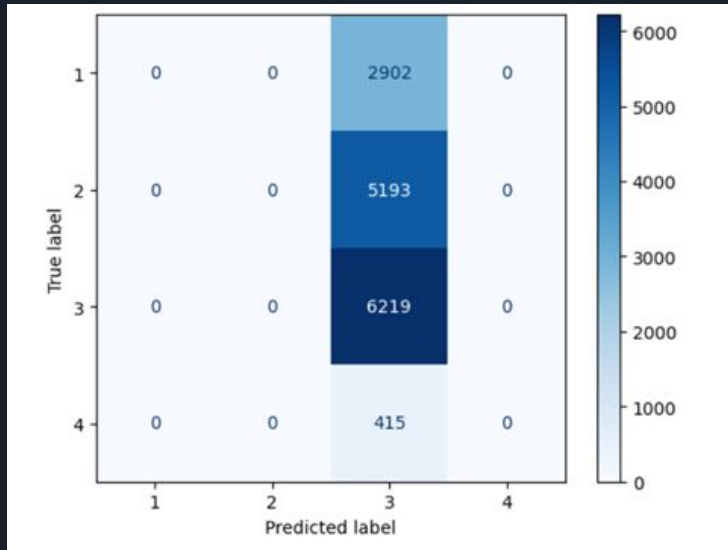
5 paramètres à régler :

- hidden_layer_sizes
- activation
- solver
- alpha
- learning rate

```
MLP : Best parameters : {'activation': 'tanh', 'alpha': 0.05, 'hidden_layer_sizes': (100,), 'learning_rate': 'constant', 'solver': 'adam'}  
MLP : Best estimator : MLPClassifier(activation='tanh', alpha=0.05)
```

MLP accuracy score (without hyper parameter tuning) : 0.4222282571797135

4) Apprentissage supervisé (MLP)



	precision	recall	f1-score	support
1	0.00	0.00	0.00	2902
2	0.00	0.00	0.00	5193
3	0.42	1.00	0.59	6219
4	0.00	0.00	0.00	415
accuracy			0.42	14729
macro avg	0.11	0.25	0.15	14729
weighted avg	0.18	0.42	0.25	14729



4) Apprentissage supervisé (Vote majoritaire)

2 types de vote :

- hard voting
- soft voting

Voting Classifier ne marche pas sur les modèles déjà entraînés => on passe les modèles avec les paramètres optimaux pour l'entraînement

```
Hard Voting Score : 0.44354674451761833
```

```
Soft Voting Score : 0.6318826804263697
```



Perspectives

Bonus (PCA, leave-one-out from scratch....)

KMeans avec conditions d'autres conditions d'arrêt:

- Convergence
- Calcul de score

Plus de test pour paramètres optimaux pour les algorithmes de hauts niveaux



Avez-vous des questions?