

Scoring d’entreprises

Table des matières

[1. Anaconda Installation 2](#_Toc515443528)

[2. Les données 2](#_Toc515443529)

[3. Les notebooks (ipynb) 3](#_Toc515443530)

[3.1. Stats\_5\_ans 3](#_Toc515443531)

[3.2. Analyse 3](#_Toc515443532)

[3.3. Scores\_predictions\_CJ 4](#_Toc515443533)

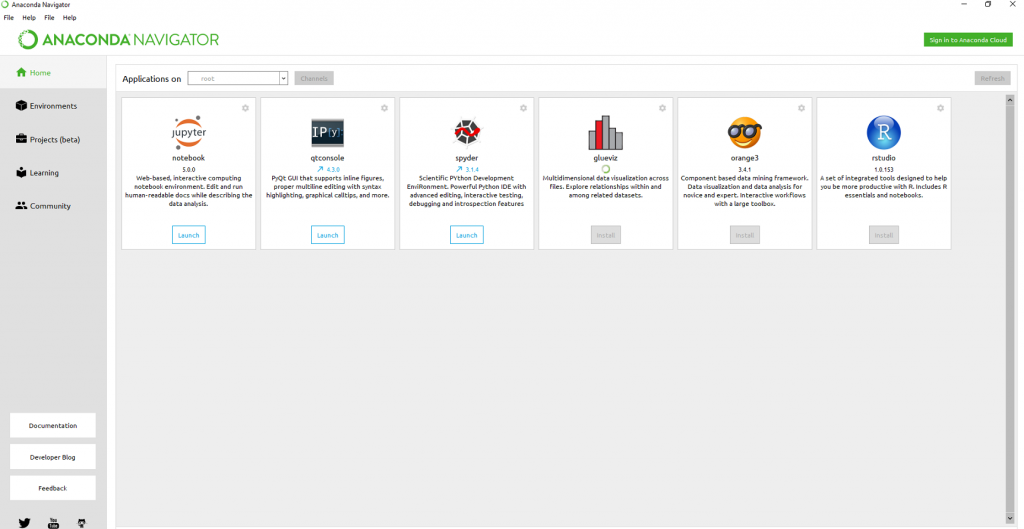
[3.4. CalculPrevision 4](#_Toc515443534)

# Anaconda Installation

Anaconda + Python 3.6 :

<https://www.anaconda.com/download/>

A l’issue de l’installation, vous aurez Anaconda Navigator. Ce dernier est une interface graphique permettant de lancer Jupyter (pour écrire du python au format notebook), ou encore Spyder (IDE pour éditer du code python).



# Les données

Les données proviennent de plusieurs tables.

Il y a deux fichiers de scripts permettant de les récupérer afin de remplir deux tables situées dans un schéma temporaire tmp.

* alim\_stats.sql
* alim\_scores\_predictions.sql

Ces fichiers sont stockés ici :

[\\GESTIONSAGE1\Fichiers\Services S&D\S&D\6\_ProjetsInterne\Morgan\scripts\](file:///\\GESTIONSAGE1\Fichiers\Services%20S&D\S&D\6_ProjetsInterne\Morgan\scripts\)

Paramétrage :

* @dtref : Date de référence
* @exoref : Exercice de référence

Avant de lancer les requêtes de ces fichiers il faut modifier les paramètres en fonction de la date courante. Par exemple, on est en avril 2018, on met @dtref au 1er du mois, ce qui nous permet de récupérer des infos valides au 1er avril 2018 et de faire une prédiction à 12 mois pour le 1er avril 2019.

Du coup @exoref doit être à 20171231 pour récupérer les chiffres du dernier bilan, @exoref1 et @exoref2 étant respectivement les dates des bilans à 1 an et à 2 ans.

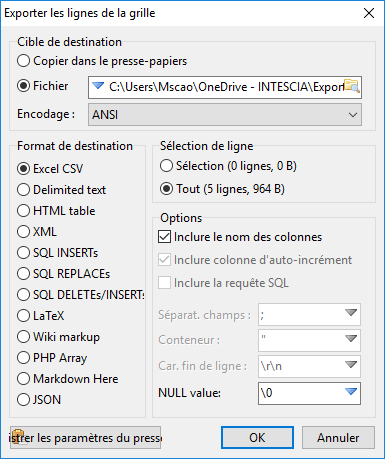
Ces deux tables sont ensuite exportées vers 2 fichiers csv :

Select \* from tmp.scores\_stats where substr(CJ,1,1)=5 ;

* stats\_5\_ans\_PROD\_CJ5.csv

Select \* from tmp.scores\_stats where substr(CJ,1,1)=5 ;

* scores\_predictions\_PROD\_CJ5.csv



Le répertoire de stockage de ces 2 fichiers doit être renseigné dans le fichier python score\_predictions.py dans la constante :

CT\_DIR\_DATA

# Le code

## Le python (.py)

Les méthodes communes se trouvent dans **score\_functions.py**. Ce fichier ne sert qu’à la factorisation du code. Il y a deux variables à paramétrer :

* CT\_DIR\_DATA : Répertoire où se trouvent les fichiers csv
* CT\_DIR\_SAVE : Répertoire pour sauvegarder les résultats

## Les notebooks (.ipynb)

## Stats\_5\_ans.ipynb

Ce notebook permet de faire des statistiques sur l’historique des indiScores :

* Répartition des SIREN par note sur 20
* Répartition des SIREN tombés en défaut par rapport à leur indiScore 12 mois avant
* Pourcentage d’erreur à 12 mois, 24 mois et 36 mois

Les SIREN en défaut sont ceux avec une procédure collective (procol) en cours, sauf cas d’absorption (procol=A).

Paramétrage possible :

* CT\_CJ : Choix de la catégorie juridique (0 = Toutes les catégories juridiques)

## Analyse.ipynb

Analyses univariées et multivariées des données avant recherche de modélisation. Complémentaire de stats\_5\_ans.

Paramétrage possible :

* CT\_CJ : Choix de la catégorie juridique (0 = Toutes les catégories juridiques)

## Scores\_predictions\_CJ.ipynb

Ce fichier entraine différents modèles et les enregistre les meilleurs dans CT\_DIR\_SAVE.

Paramétrage possible :

* CT\_CJ : Choix de la catégorie juridique (0 = Toutes les catégories juridiques)
* CT\_SAMPLE : Utilisation d’un échantillon des données
* nSample : Nombre de données (si CT\_SAMPLE=True)
* CT\_BILAN : Prise en compte des bilans financiers
* CT\_SAVE : Sauvegarde du modèle et des outils de préparation des données
* CT\_REDUC : Réduction dimensionnelle avant recherche du modèle
* Dropcols : Colonnes à ne pas utiliser pour les calculs, utile pour enlever les historiques de l’indiScore, l’encours et procol

Pour utiliser la sauvegarde (CT\_SAVE=True), il faut que toutes les données soient chargées, c’est-à-dire que CT\_SAMPLE soit à False.

Une fois le meilleur modèle entrainé, il faudra renommer les fichiers correspondants pour les utiliser avec le notebook suivant (CalculPrevision), par ex :

* CJ5\_RandomForestClassifier(1143846, 39)816.pkl 🡪 CJ5\_RandomForestClassifier\_ok.pkl
* CJ5\_model\_scaler(1143846, 322).pkl 🡪 CJ5\_model\_scaler\_ok.pkl

## CalculPrevision.ipynb

Ce fichier permet de calculer la prévision finale pour un dataset complet

Paramétrage possible :

* CT\_CJ : Choix de la catégorie juridique (0 = Toutes les catégories juridiques)
* CT\_NEW\_SCORE :
  + True pour calculer le score correspondant à @dtref
  + False pour calculer le score correspondant à @dtref moins 1 an

Ce calcul final a donc besoin des fichiers généré précédemment par Scores\_predictions\_CJ.ipynb :

* CJ5\_RandomForestClassifier\_ok.pkl
* CJ5\_model\_scaler\_ok.pkl

Mais aussi des fichiers contenant les listes des différents choix possibles pour les features catégorielles :

* CJ5\_column\_xxx.pkl, où xxx correspond au nom de la feature (généré aussi par Scores\_predictions\_CJ.ipynb)