1 Objectif

Découverte du logiciel Data Science Studio (DSS) de Dataiku.

« Big data » et « data science » sont des termes particulièrement en vogue aujourd'hui. Il suffit de regarder les tendances sur Google trends pour s'en rendre compte (1,2). Tout le monde s'accorde à dire que les statistiques et l'informatique tiennent une place primordiale dans cette reconfiguration de la valorisation des données. Mais qu'y a-t-il de vraiment nouveau aujourd'hui par rapport aux statistiques exploratoires et à la modélisation statistique qui sont des thématiques anciennes ? Qu'y a-t-il de nouveau par rapport au data mining, autre domaine relativement ancien ? Je note avec un certain amusement d'ailleurs que la vague « data mining » avait aussi suscité un certain scepticisme à son époque (milieu des années 90)[1].

Je doute que l'on puisse apporter une réponse définitive à ces questions. Je ne suis pas sûr que ce soit réellement intéressant en réalité. A mon sens, nous (enseignants-chercheurs) devrions plutôt s'attacher à identifier les compétences et savoir-faire nouveaux que nous devons transmettre à nos étudiants, atouts qu'ils peuvent valoriser lorsqu'ils arrivent sur le marché du travail. Et, à ce titre, la question des outils informatiques tient une place importante.

En effet, l'évolution du métier de statisticien s'accompagne de l'arrivée de logiciels de nouvelle génération. Mon attention a été attirée récemment par le logiciel Data Science Studio (DSS) de la société Dataiku. Un logiciel de plus me direz-vous. Oui et non. Certes, la trame du processus d'analyse reste la même : accéder aux données, les préparer, créer des modèles statistiques, valider ces derniers. Mettre au point des méthodes et des implémentations performantes reste d'actualité. C'est le mode opératoire proposé qui est nouveau. L'outil fonctionne de manière comparable à Azure Machine Learning Studio de Microsoft que j'avais présenté sur ce blog bien que, fondamentalement, il soit différent parce que ne repose pas *exclusivement* sur le paradigme SaaS (logiciel en tant que service)[2].

[1] Google trends, sans pitié, nous annonce que la vague data mining est en train de retomber doucement ces dernières années. Comme quoi il faut toujours se méfier des effets de mode.

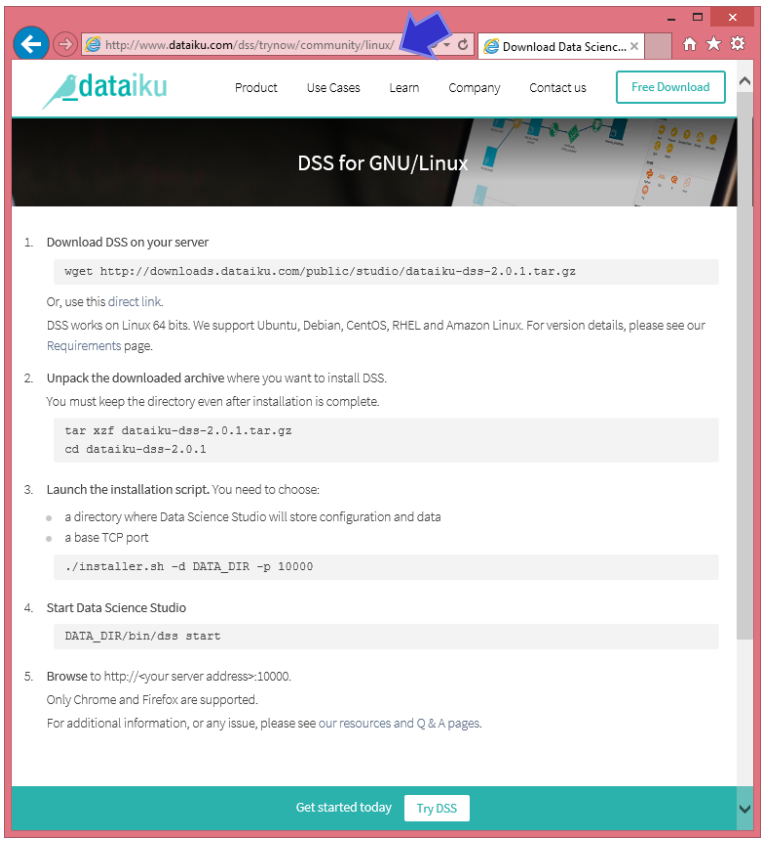
[2] « Dataiku veut conquérir le marché américain avec ses analyses prédictives », Interview de Clément Stenac, cofondateur de Dataiku par FrenchWeb.fr (Olivier Harmant), 15 juin 2015.

Les traits communs de ces outils de nouvelle génération peuvent se résumer de la manière suivante (de manière non exhaustive) : architecture client-serveur, travail en ligne et pilotage via une interface web, possibilité de mettre en place un travail collaboratif, simplification à l'extrême des process, centré sur les aspects opérationnels.

Dans ce tutoriel, je présente la version **Community Edition** du logiciel DSS. Je me centre sur un processus « Machine Learning » d'analyse prédictive c.-à-d. développer un modèle statistique de scoring. J'explore de manière relativement sommaire les possibilités de l'outil. D'une part, parce que dans une première approche, il convient de rester schématique pour bien discerner ses principales caractéristiques. D'autre part, parce qu'il serait vain de vouloir tout résumer dans un document de quelques pages toutes ses fonctionnalités. Le lecteur curieux pourra se référer au site de documentation de l'éditeur (http://learn.dataiku.com/) ou aux tutoriels accessibles sur youtube.

2 Installation et démarrage de DSS

J'ai choisi d'expérimenter la version de DSS pour Linux. J'utilise la distribution **Ubuntu 15.04 - 64 bits**. La procédure d'installation est décrite en ligne.

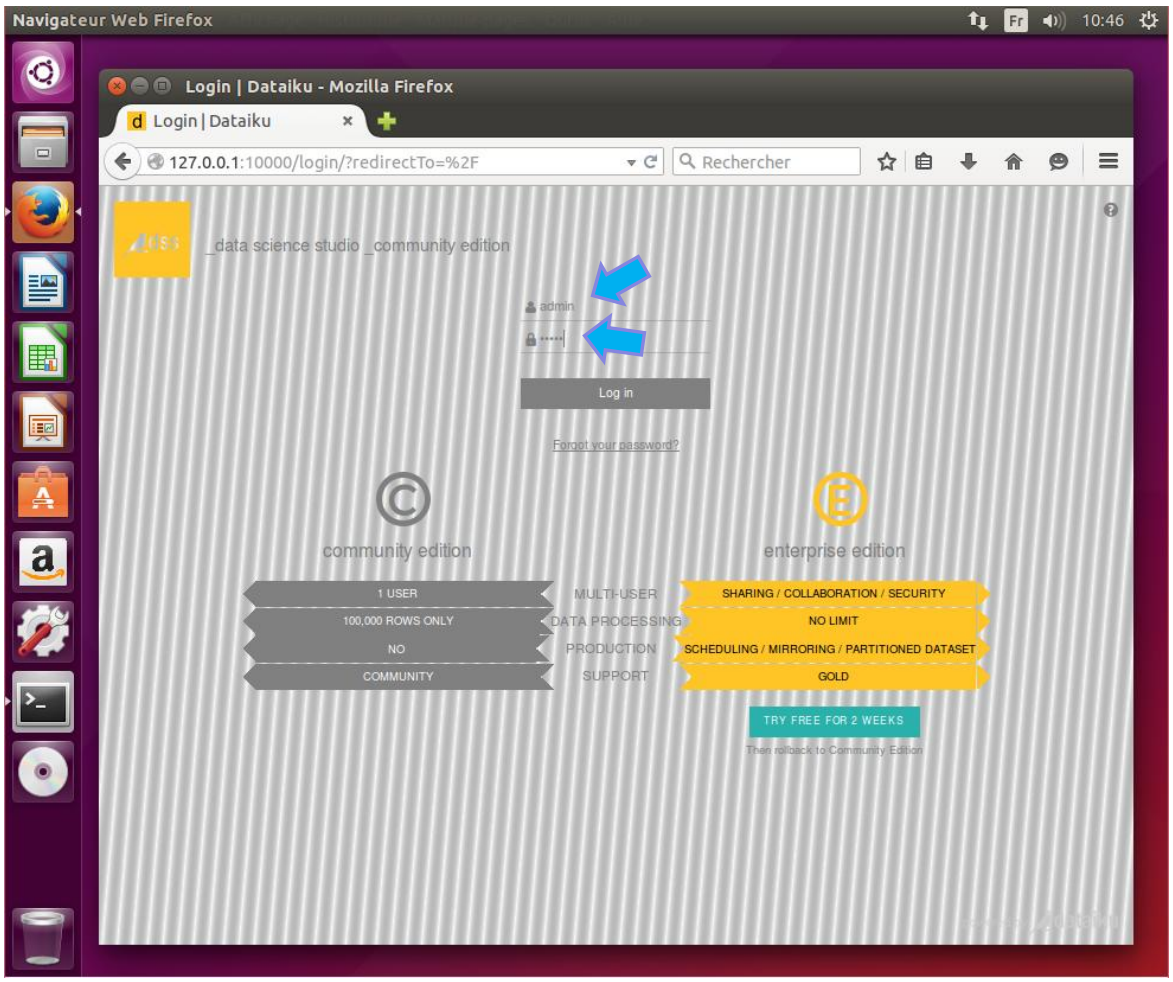


Je l'ai reproduite telle quelle et elle a fonctionné. Il faudra simplement songer à installer au préalable les dépendances. Un message d'avertissement est affiché. L'instruction adéquate est indiquée. Pour ma part, je n'ai rencontré aucune difficulté.

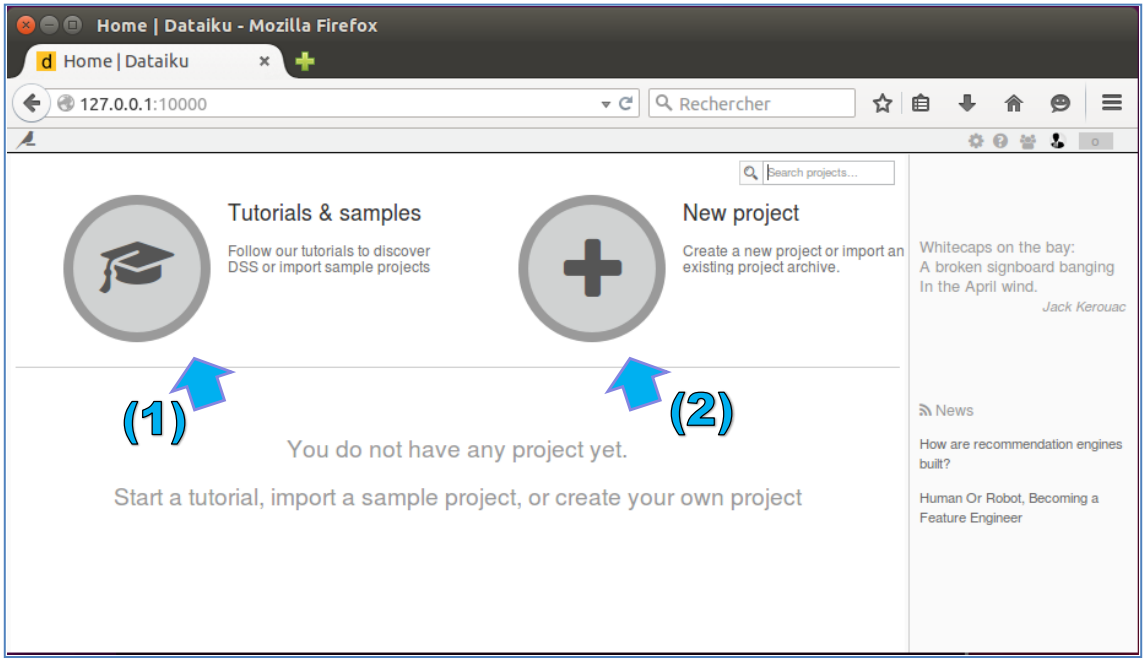
La commande…

```
$ dss start
```

…permet de démarrer le serveur. Il faudra introduire cette commande à chaque redémarrage de votre machine. Par la suite, pour accéder au logiciel, nous démarrons un navigateur web, Firefox en ce qui me concerne. J'introduis l'URL : `http://127.0.0.1:10000` ; « 127.0.0.1 » parce que le serveur est localement sur ma machine, « 10000 » parce que c'est le port que j'ai spécifié lors de l'installation. Nous aboutissons à la page de garde, nous utilisons le login standard « `admin/admin` ».



Nous aboutissons à notre espace de travail. Nous avons le choix entre nous laisser guider par un tutoriel interactif (1) ou créer un nouveau projet (2).

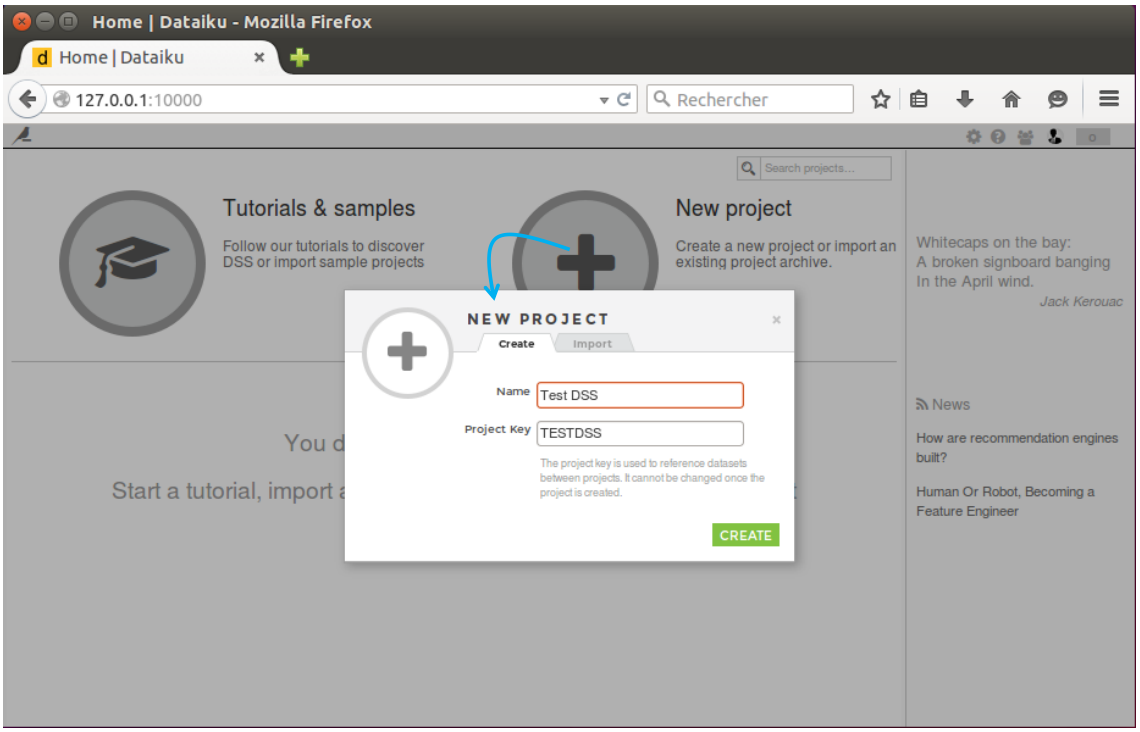


3 Processus « machine learning »

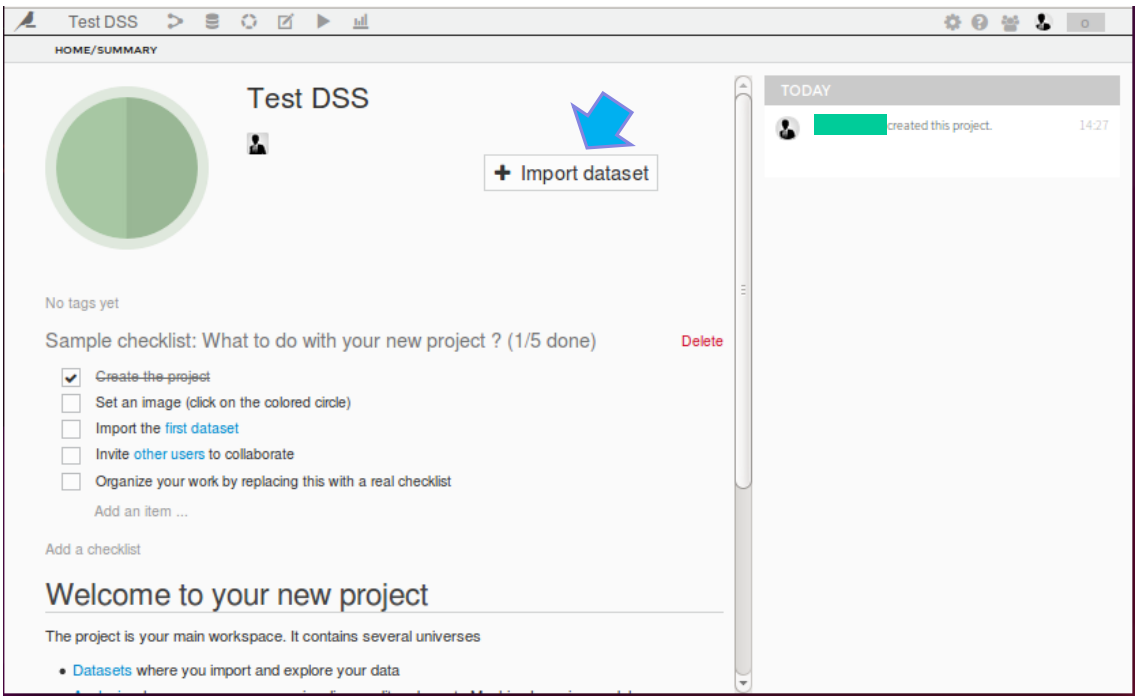
Cette section est fortement inspirée du DSS 103, à la différence qu'il est en français et que j'utilise mes propres données.

3.1 Création d'un projet et importation des données

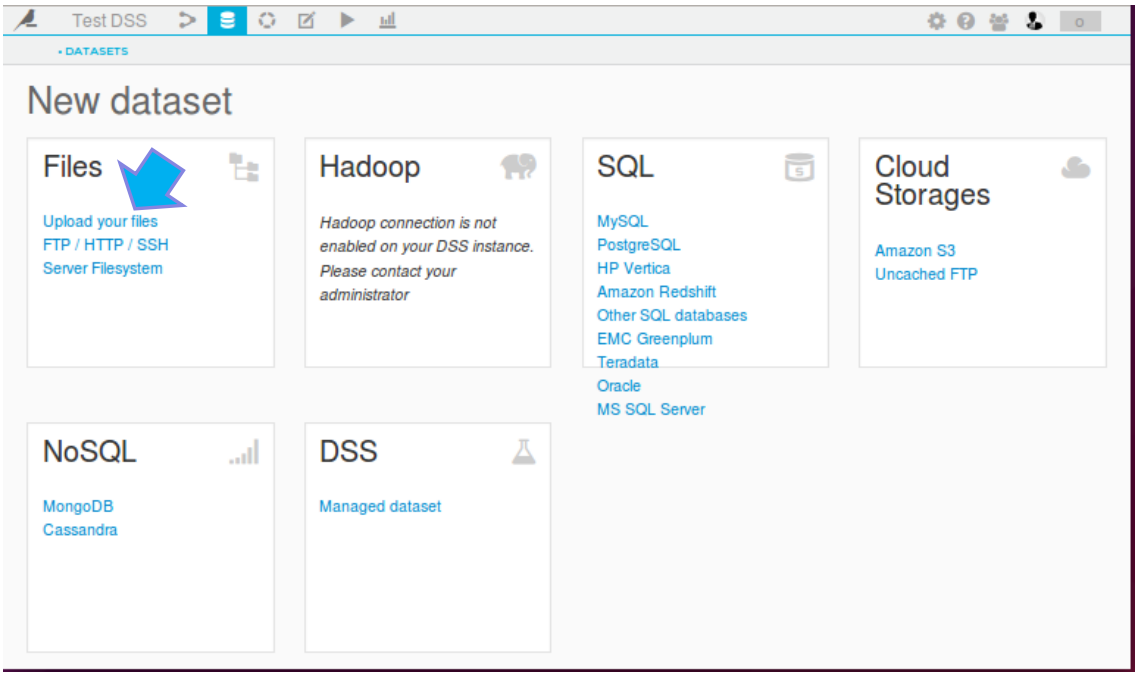
Nous cliquons sur le bouton « NEW PROJECT ». Une boîte de dialogue apparaît nous demandant le nom du projet. Nous l'appelons très prosaïquement « Test DSS ».



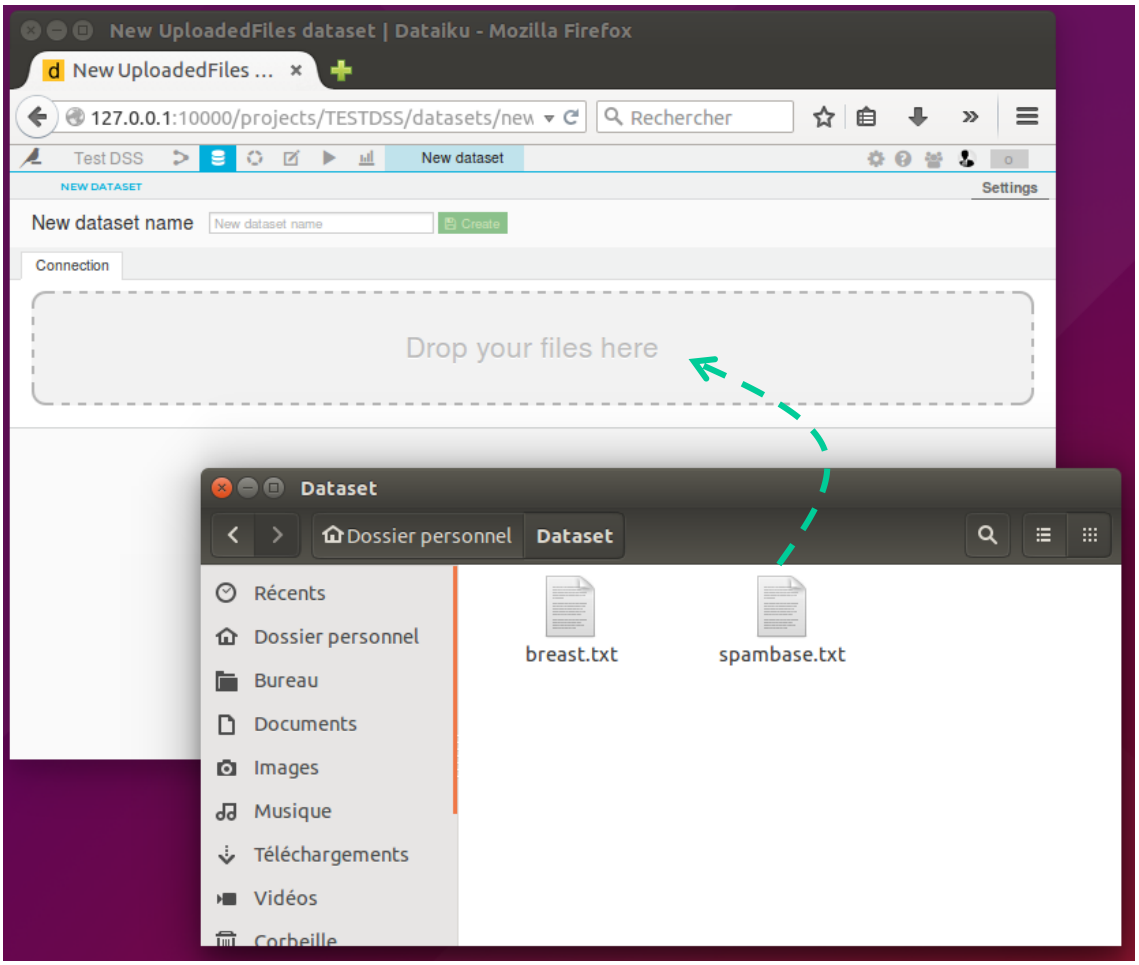
Nous cliquons sur CREATE pour valider. Une nouvelle page est affichée, nous invitant à importer les données.



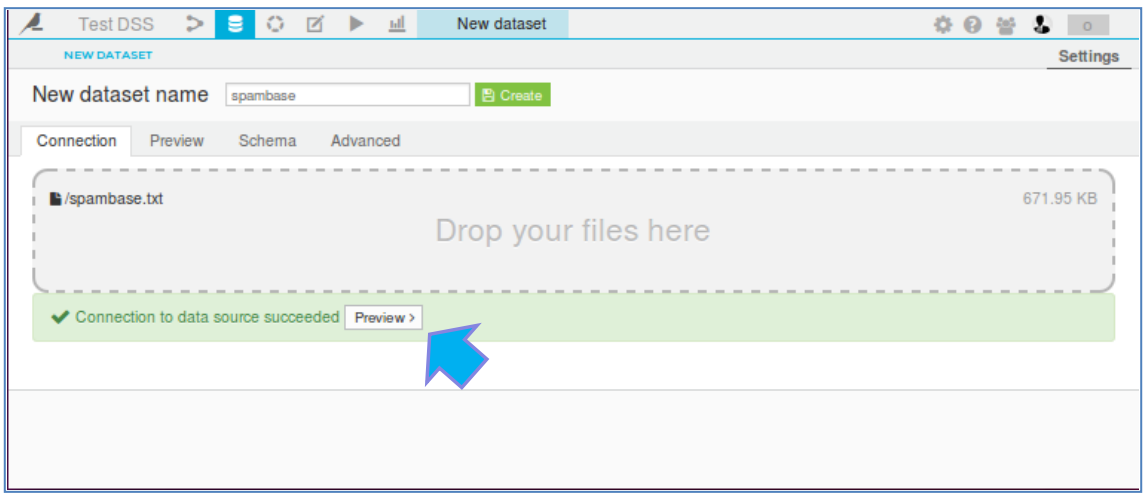
On aurait tort de s'en priver. Nous cliquons sur le bouton idoine. La page suivante s'affiche nous indiquant les différentes sources de données accessibles. Nous notons la richesse fonctionnelle de l'outil. Nous nous en tiendrons au fichier texte avec séparateur tabulation en ce qui nous concerne. Nous sélectionnons l'option UPLOAD YOUR FILES.



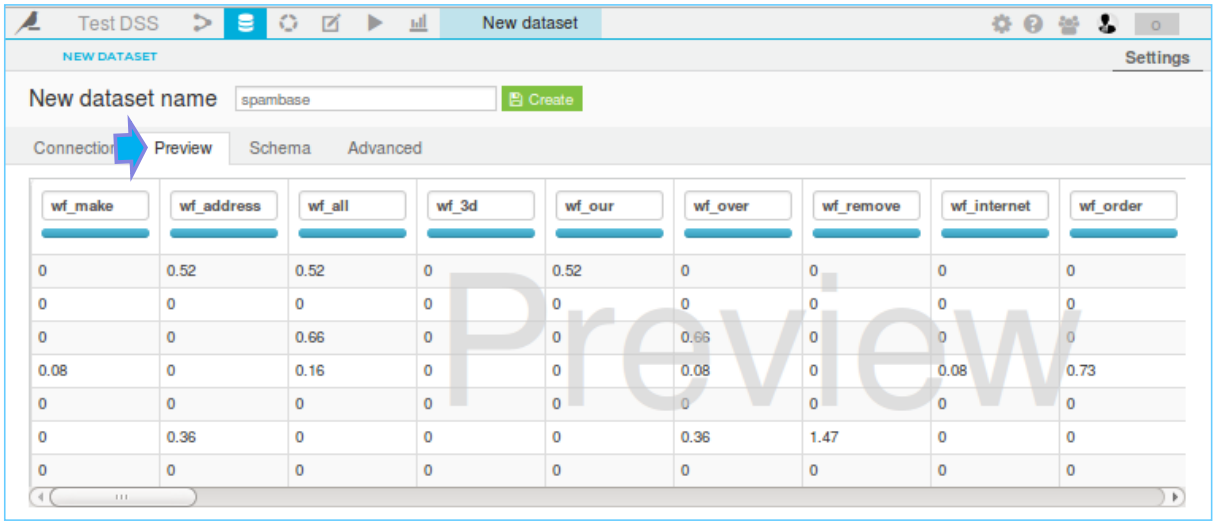
L'interface d'importation s'affiche. Nous glissons la base « spambase.txt » que nous avions préparé au préalable. L'objectif est de détecter le caractère frauduleux (spam : yes ou no) d'un message à partir de ses caractéristiques (fréquence des mots, fréquence de certains caractères, etc. ; voir « Spambase Data Set » - UCI Machine Learning Repository).



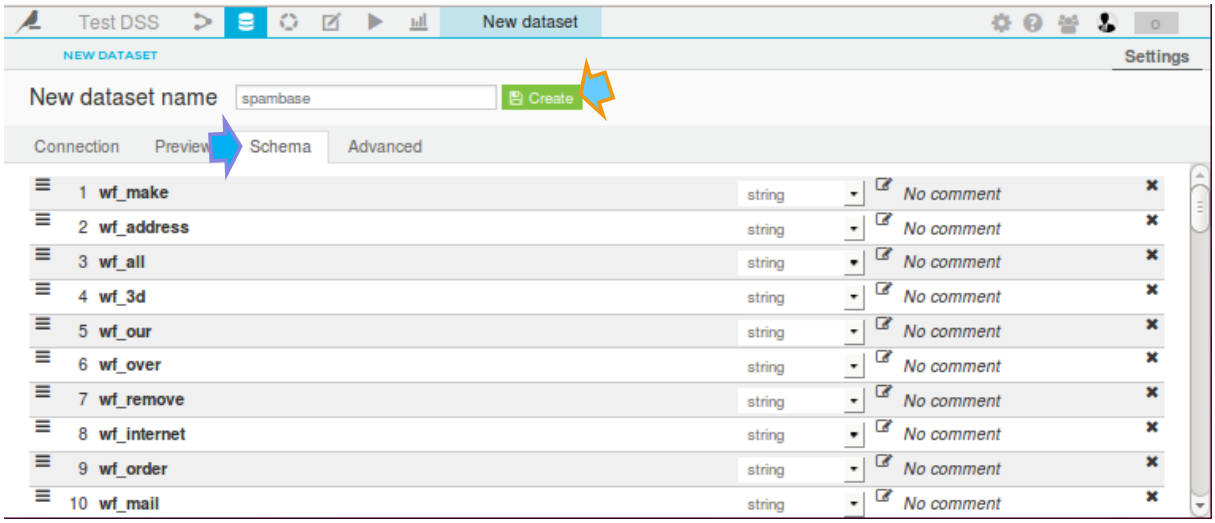
Les données sont importées, nous pouvons les pré-visualiser.



Elles sont affichées dans une grille de l'onglet « Preview ». La première ligne correspond aux noms des variables.



Dans l'onglet « Schema », nous disposons de la liste des variables avec le typage initial. Pour l'instant, toutes les colonnes sont considérées comme « chaîne de caractères » (string). Nous reviendrons sur ce point lorsqu'il faudra lancer réellement l'analyse.

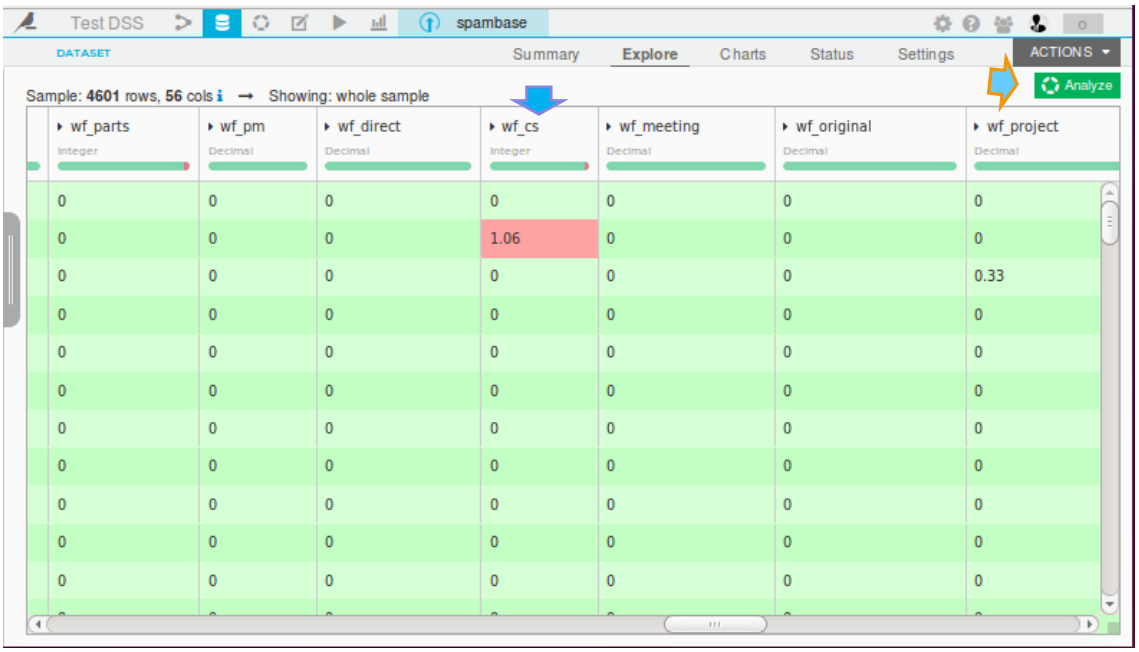


Nous cliquons sur le bouton vert CREATE pour rendre effectif l'importation. C'est durant cette étape que le typage des variables s'effectue.

3.2 Typage explicite des variables - Préparation des données

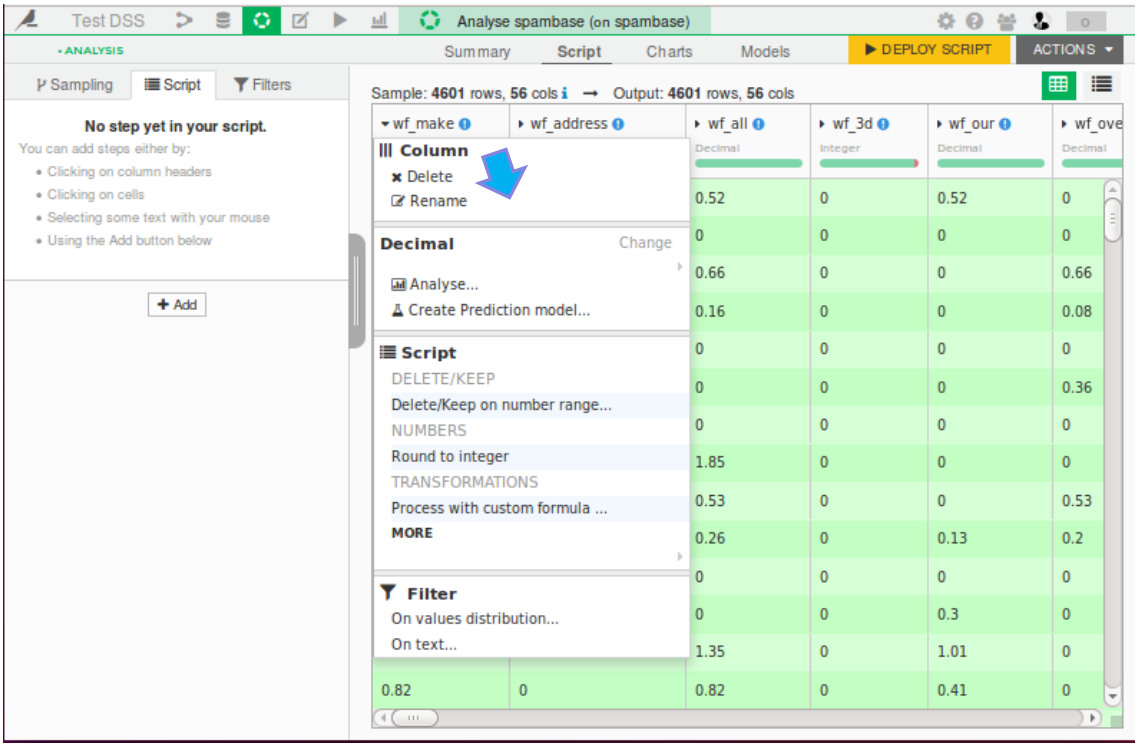
DSS s'appuie sur les premières lignes des valeurs pour détecter le type des variables. Lorsqu'il parcourt par la suite le reste de la colonne pour chaque variable, il signale les incohérences dans l'en-tête. Si la barre verte est complète, toute la colonne est validée. Si une partie rouge

apparaît, des incohérences ont été détectées et les cellules correspondantes sont mises en surbrillance. Dans la copie d'écran ci-dessous, WF_CS est identifiée comme une colonne de valeurs entières, et le nombre 1.06 (entre autres) pose problème.

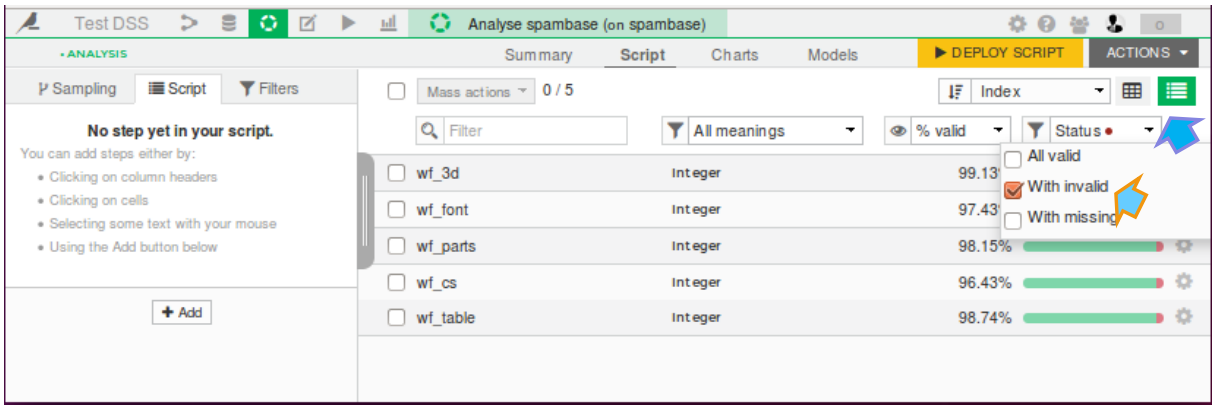


Pour démarrer la manipulation des données, nous actionnons le bouton vert ANALYZE.

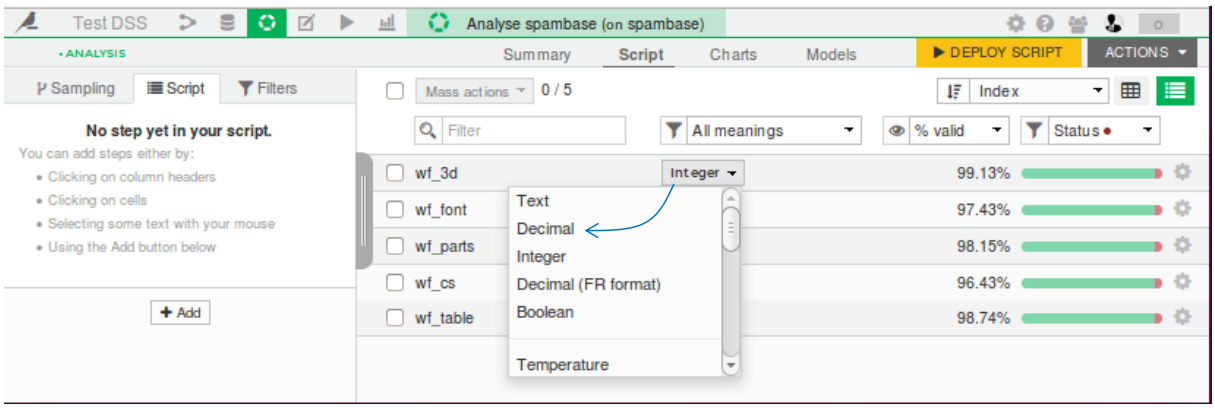
Dans la nouvelle interface, diverses opérations (filtrage, transformation, analyse) sont possibles. Lorsque nous cliquons sur l'en-tête de la première colonne, le menu suivant apparaît.



Nous nous attachons dans un premier temps à corriger les problèmes de typage automatique relevés ci-dessus. Nous activons la visualisation par colonnes (COLUMNS VIEW, le bouton à droite), nous filtrons la liste afin de ne faire apparaître que les variables à problème.

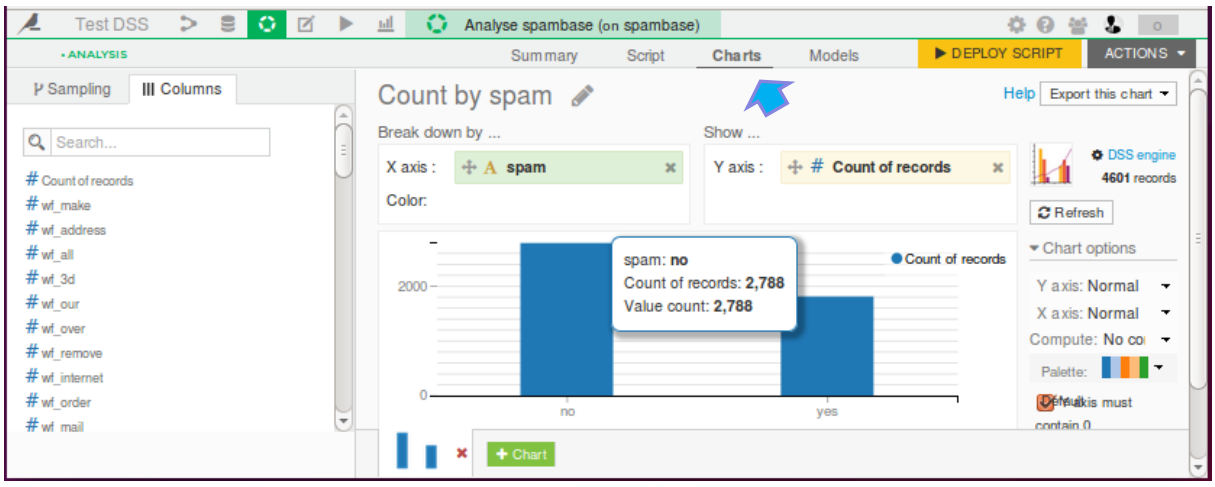


Nous modifions alors les types manuellement en passant les INTEGER en DECIMAL.



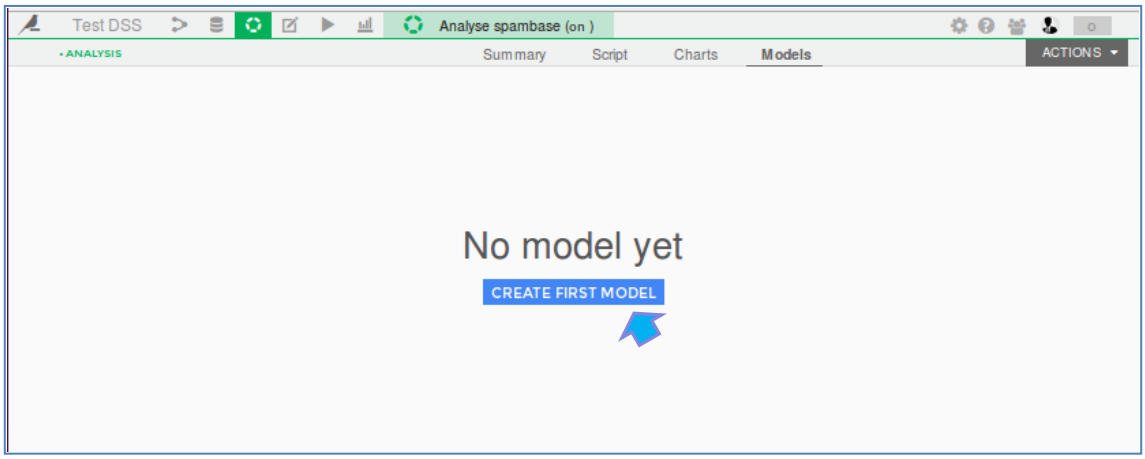
3.3 Graphiques

L'onglet CHARTS permet de définir des graphiques simples. Pour calculer la distribution de fréquence des classes par exemple, nous paramétrons comme suit (X axis et Y axis).

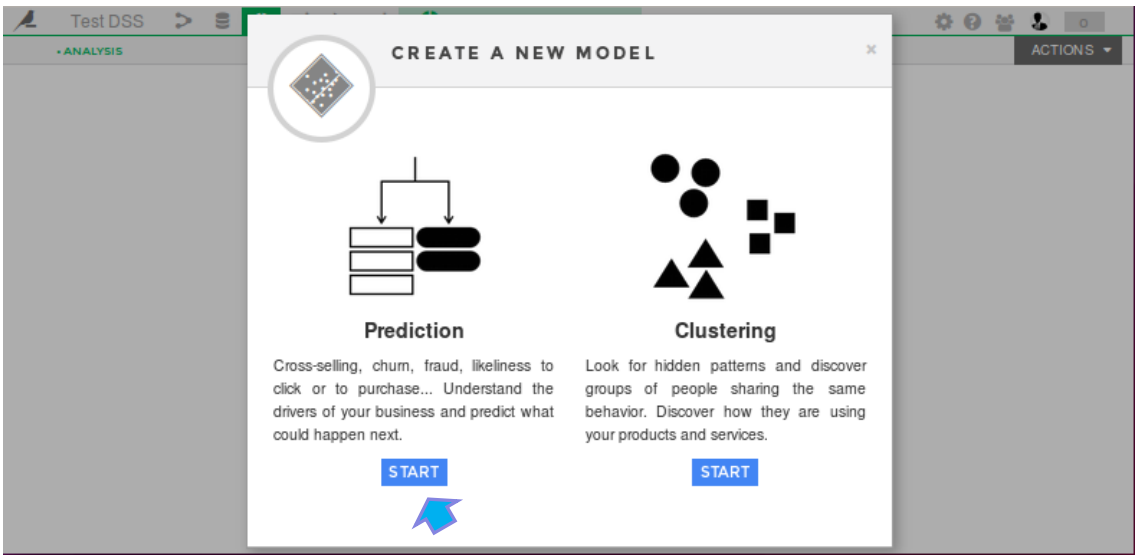


3.4 Analyse prédictive avec DSS

Entrons maintenant dans le cœur de notre propos : la modélisation statistique. Nous activons l'onglet MODELS.

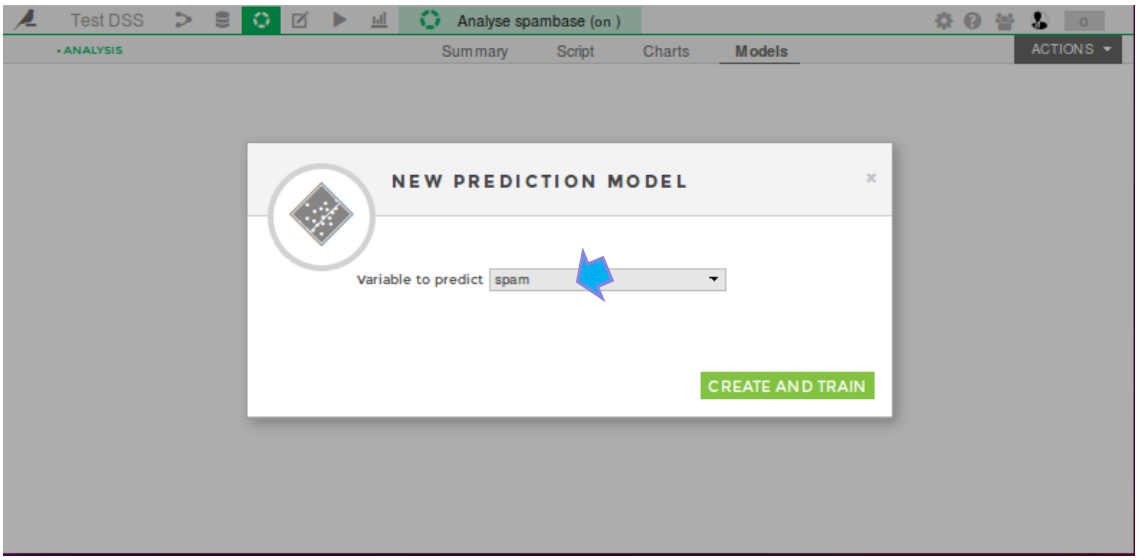


Nous souhaitons créer notre premier modèle.

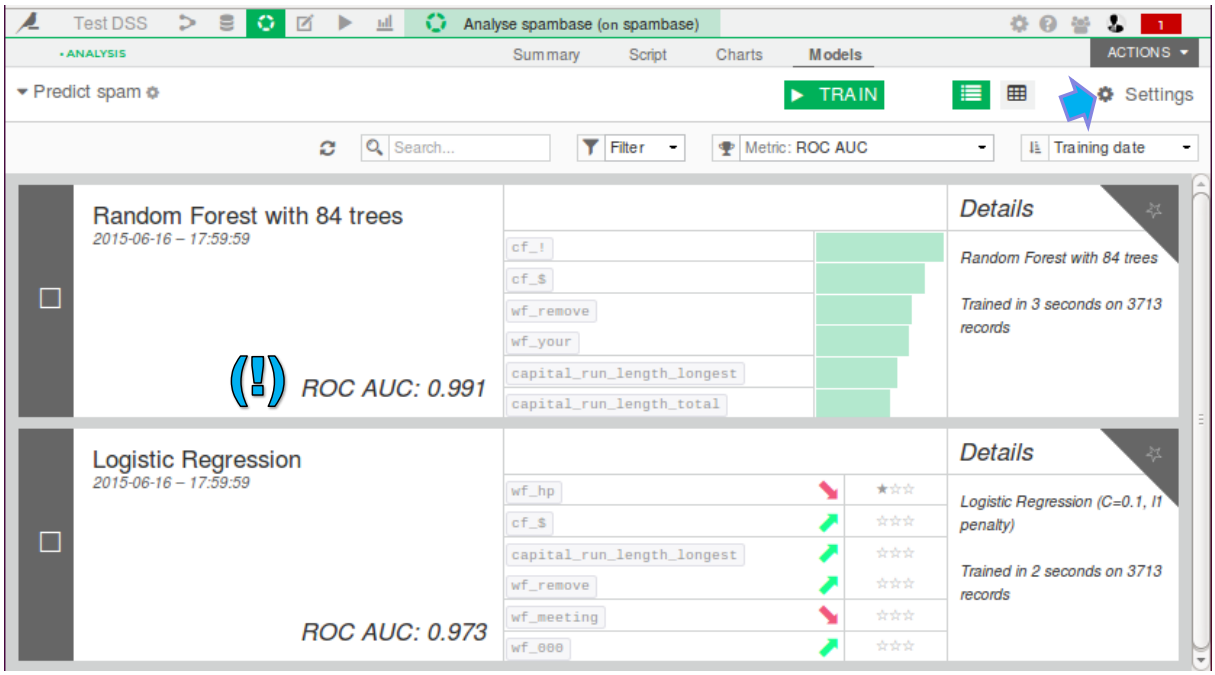


Nous avons le choix entre l'apprentissage supervisé (PREDICTION) ou non supervisé (CLUSTERING). Nous souhaitons identifier le caractère frauduleux des messages électroniques (SPAM : yes ou no) à partir de leurs propriétés. SPAM est la variable à prédire, nous sommes bien dans le cadre de l'apprentissage supervisé que l'on désigne communément « analyse prédictive » dans le jargon du data scientist.

Nous cliquons sur START de la partie PREDICTION. Le logiciel nous demande désigner la variable cible, en l'occurrence la colonne SPAM. Nous validons avec CREATE AND TRAIN.



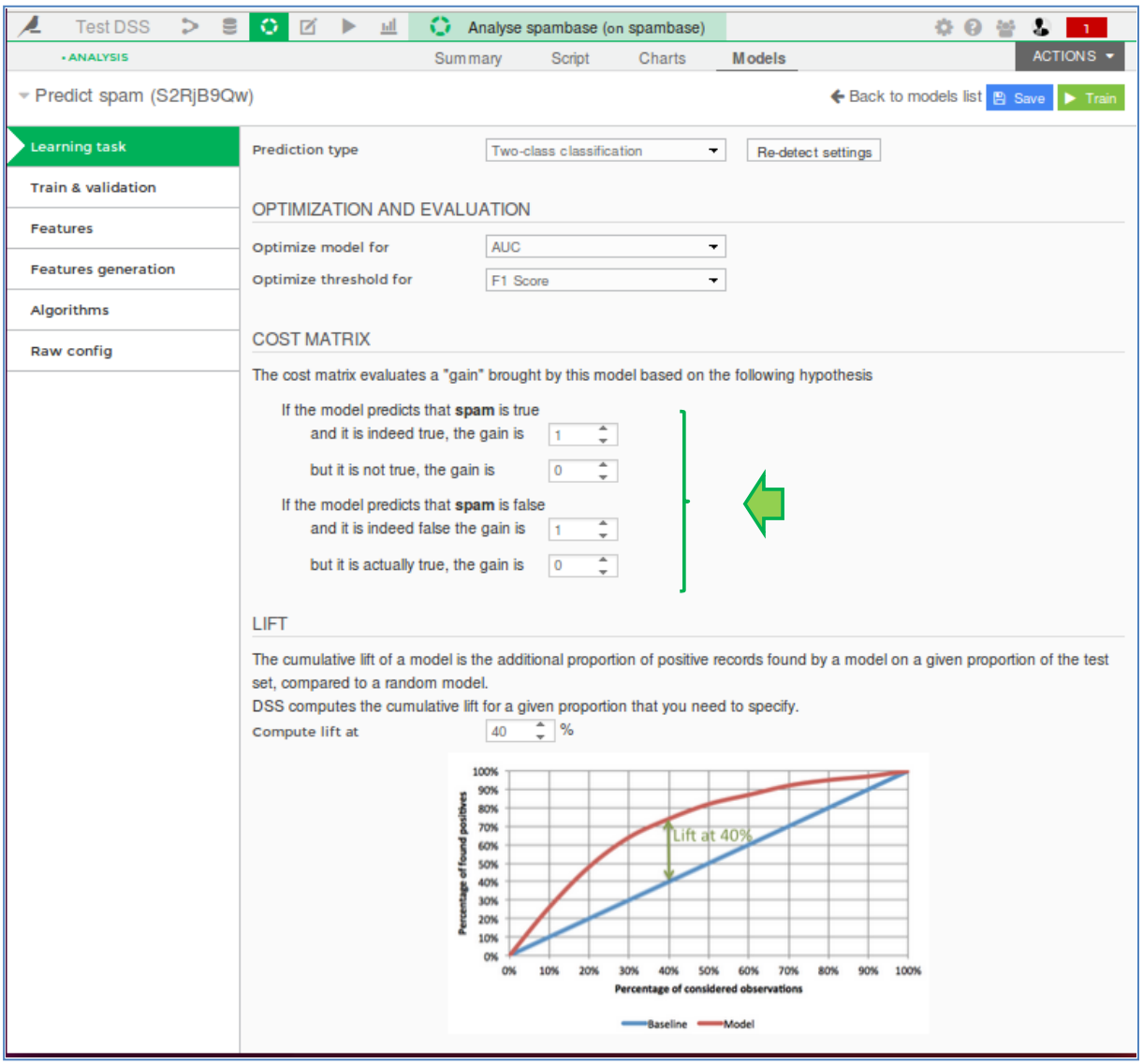
Deux modèles scoring sont automatiquement générés : une forêt aléatoire (Random Forest) avec 84 arbres, et une régression logistique (Logistic Regression). Le critère AUC (Area Under Curve) de la courbe ROC permet de les comparer. Sur nos données, Random Forest semble meilleur avec une AUC de 0.992 (contre 0.973). Les variables prédictives sont affichées par ordre de pertinence, les 6 premières sont directement visibles.



Tout ça paraît très facile. La réalité est un peu plus complexe. Déjà, les performances des algorithmes d'apprentissage sont dépendant de paramètres. On ne nous a rien demandé. Il nous revient de nous enquérir à quel endroit ils sont accessibles et modifiables. Il faut actionner le bouton **SETTINGS** en l'occurrence.

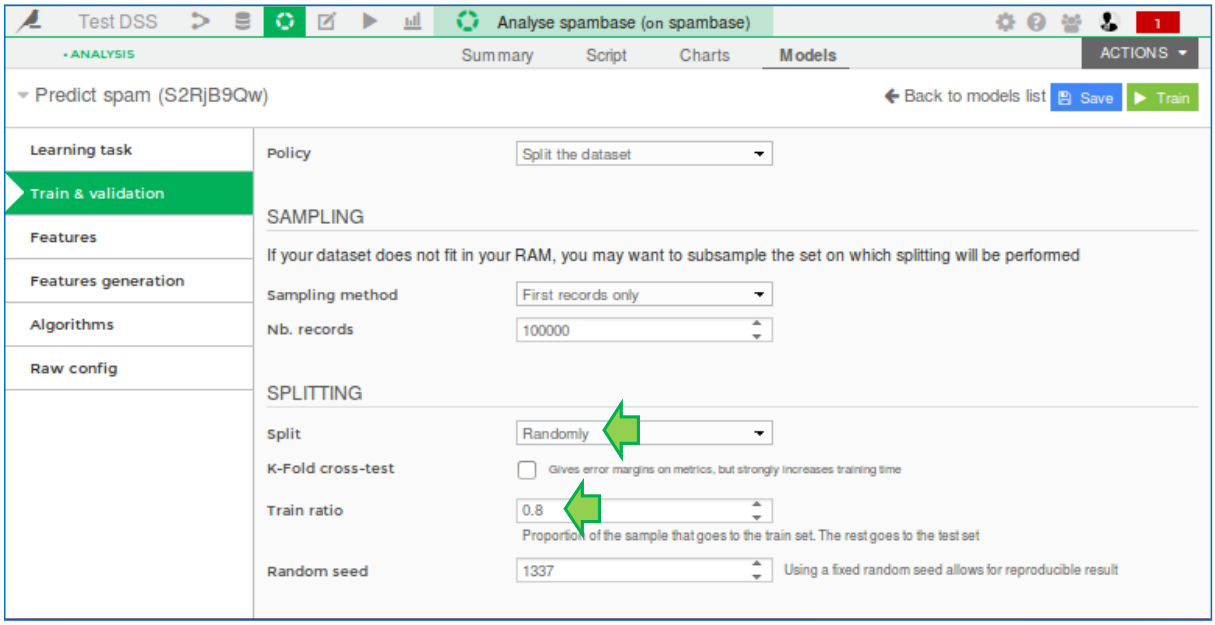
3.4.1 Paramétrage des algorithmes

L'item **LEARNING TASK** indique la nature de la modélisation, les critères à optimiser, la matrice de coûts de mauvais classement (matrice « gain » ici en réalité), et la définition du seuil de ciblage dans la courbe LIFT. Nous modifions uniquement la matrice coûts.

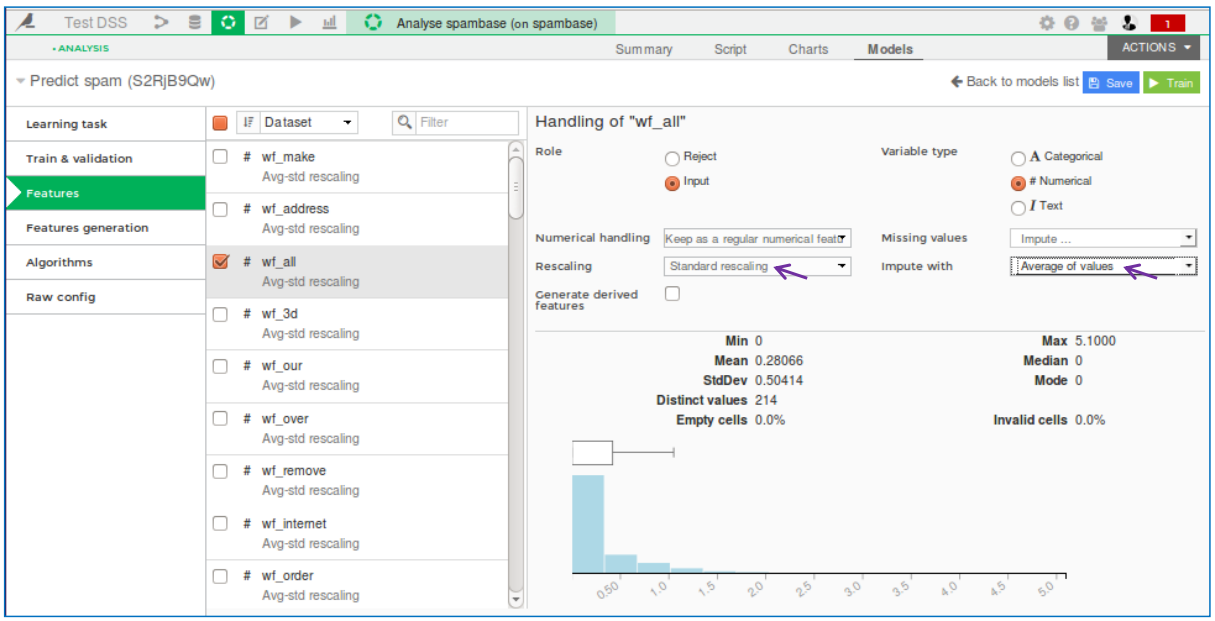


L'item **TRAIN & VALIDATION** indique la partition aléatoire des données en échantillons d'apprentissage (pour élaborer les modèles) et de test (pour en évaluer les performances). Par défaut, la méthode « holdout »[3] est utilisée avec une subdivision 80% (TRAIN RATIO) / 20%. Nous pouvons également passer par une validation croisée.

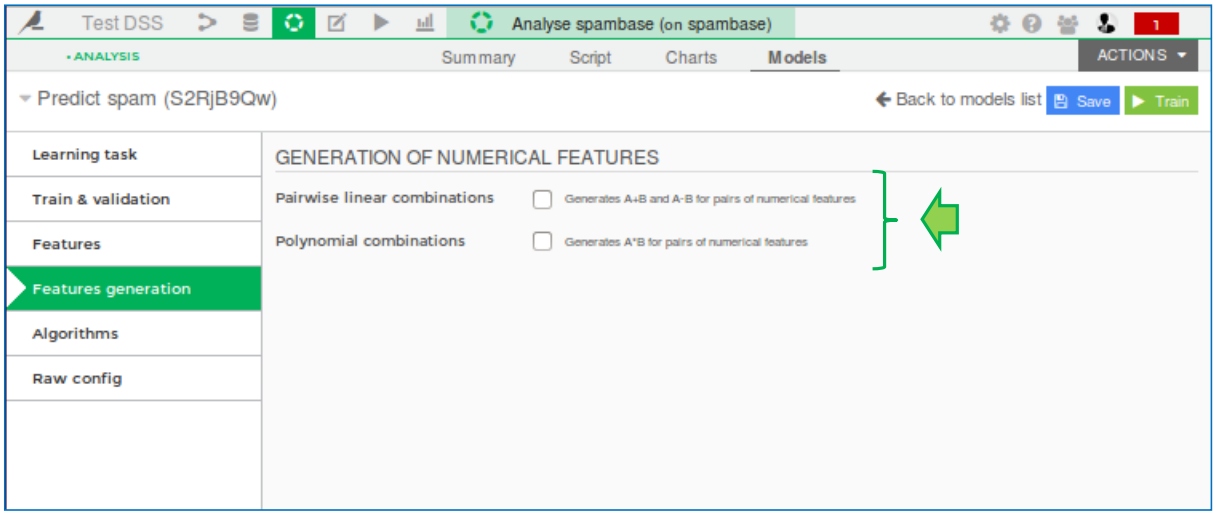
[3] Tutoriel Tanagra, « Validation croisée, bootstrap – Diapos », février 2015.



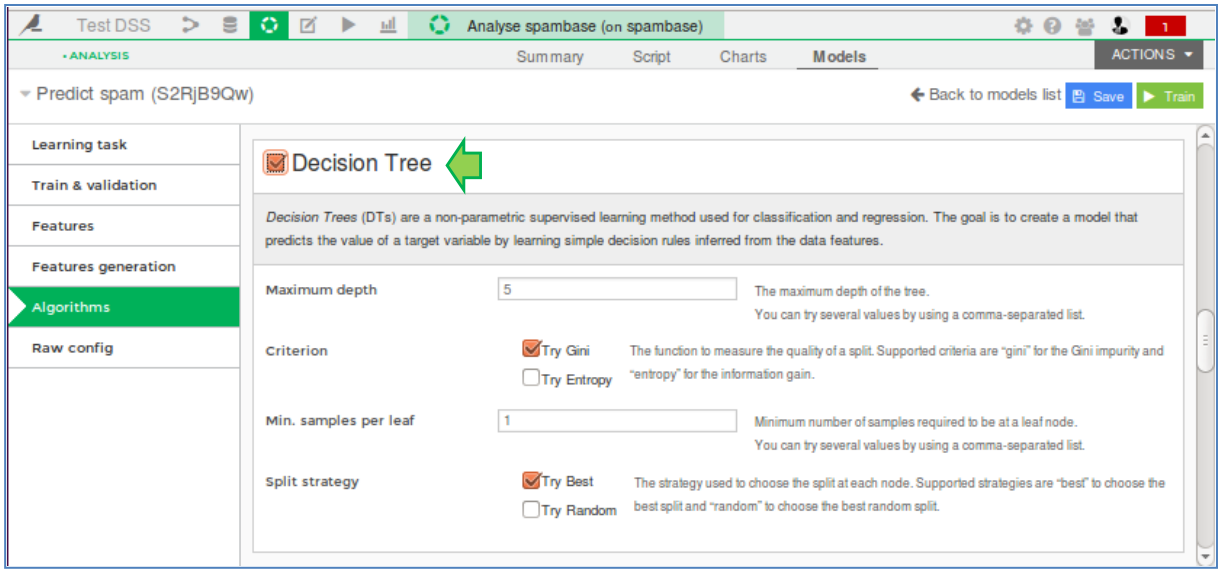
FEATURES énumère les variables disponibles, leurs caractéristiques et leurs rôles. On constate que les variables numériques (ex. WF_ALL) sont automatiquement centrées et réduites. En cas de données manquantes, l'imputation par la moyenne est utilisée.



FEATURES GENERATION permet de définir automatiquement des combinaisons de variables pour améliorer la qualité de la représentation. C'est aussi une grande porte ouverte au sur-apprentissage ceci étant dit. On notera que les options sont désactivés par défaut.



ALGORITHMS liste les méthodes disponibles et leur paramétrage. Il faut un minimum de connaissances pour pouvoir modifier à bon escient les paramètres des algorithmes. Ouf ! Faire des études de statistiques et de data mining (et accessoirement de les enseigner) sert encore un peu à quelque chose. L'apparente convivialité de l'outil ne doit pas masquer cette difficulté.

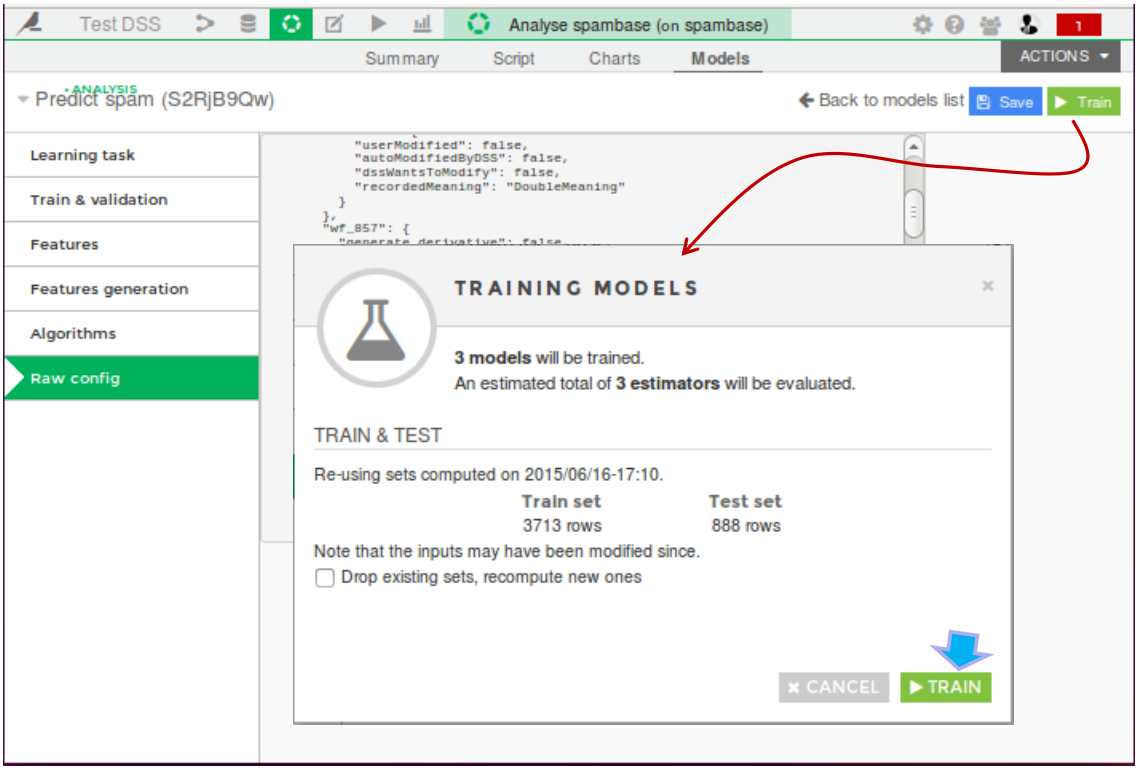


Outre les forêts aléatoires et la régression logistique, nous décidons d'activer l'induction d'un arbre de décision. La profondeur de l'arbre sera au maximum de 5 niveau (MAXIMUM DEPTH).

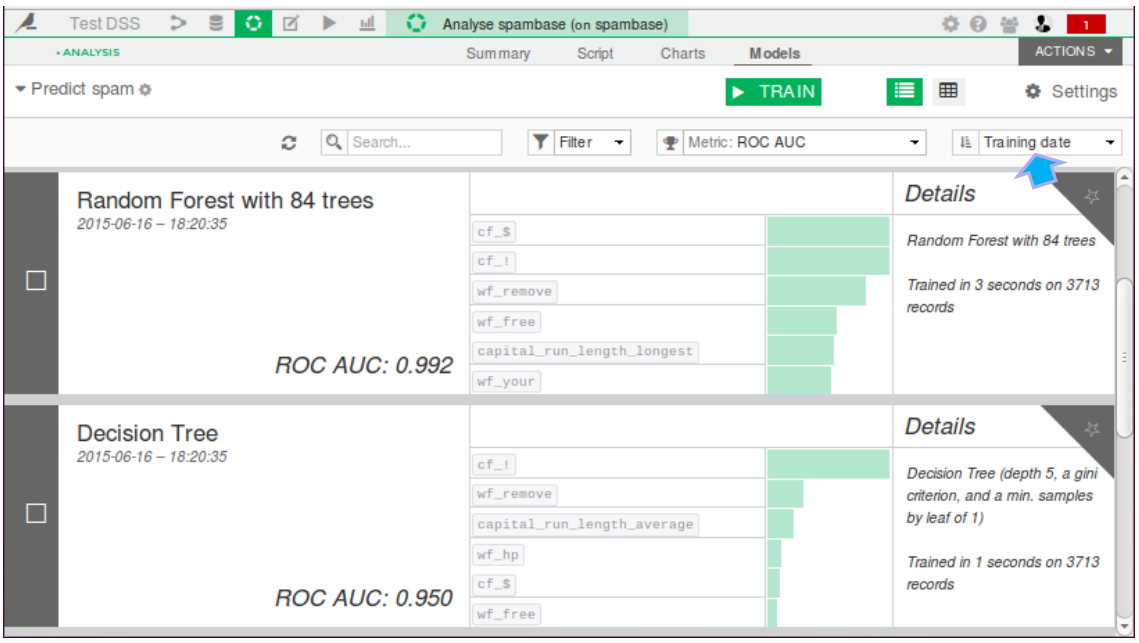
Enfin, **RAW CONFIG** donne accès au détail du paramétrage du processus au format JSON. Nous distinguons plus ou moins les paramètres précités. Cette option est vraisemblablement destinée à la sauvegarde. L'utilisateur initié saura aussi en tirer parti puisque nous disposons de tous les détails.

3.4.2 Construction des modèles (2)

Maintenant que nous avons mis la main sur le paramétrage, nous pouvons construire les modèles incluant l'arbre de décision. Nous cliquons sur le bouton TRAIN.



Une boîte de dialogue annonçant que les 3 nouveaux modèles seront construits sur les mêmes ensembles d'apprentissage et de test définis précédemment apparaît. Nous validons en cliquant sur l'autre bouton TRAIN. Ils apparaissent dans notre espace de travail.

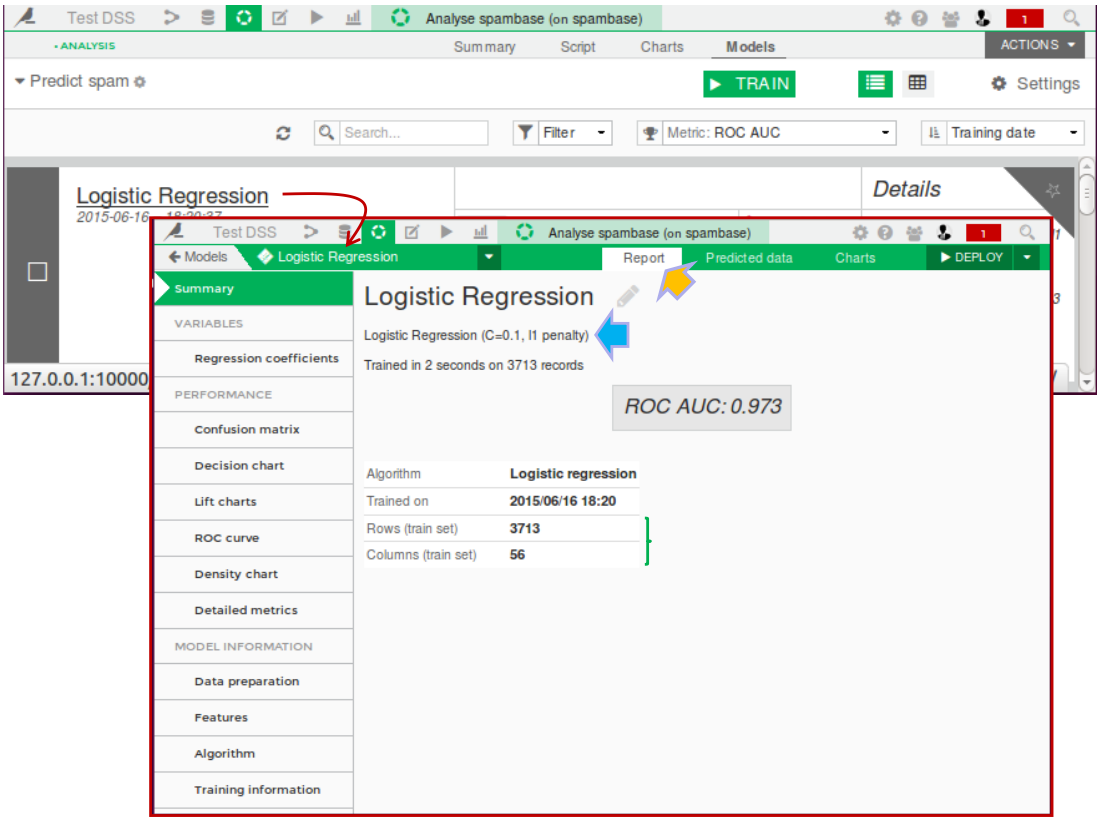


3.4.3 Inspection des modèles - Régression logistique

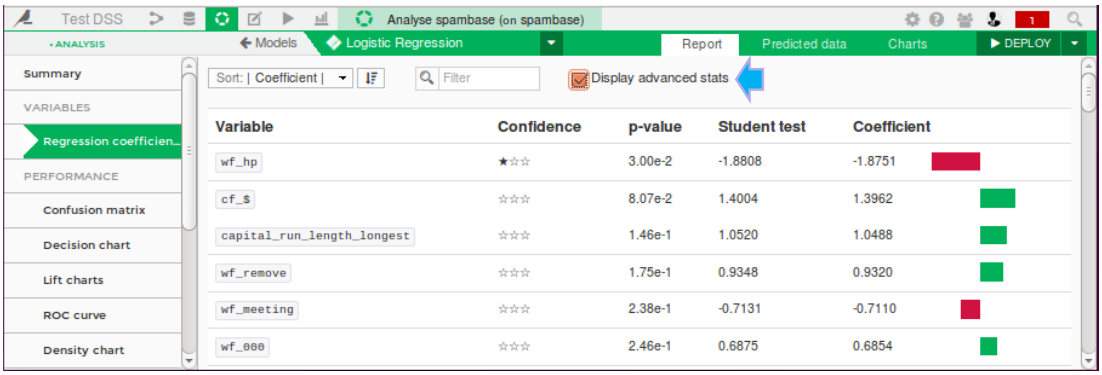
Les fonctionnalités sont nombreuses. Allons à l'essentiel (si on peut dire) dans cette section.

3.4.3.1 Description du modèle et de ses performances

Summary. Nous cliquons sur la régression logistique dans la liste des modèles. La page des résultats apparaît, l'onglet REPORT est automatiquement activé. Nous avons un résumé dans un premier temps : la méthode et ses paramètres, les caractéristiques des données utilisées.



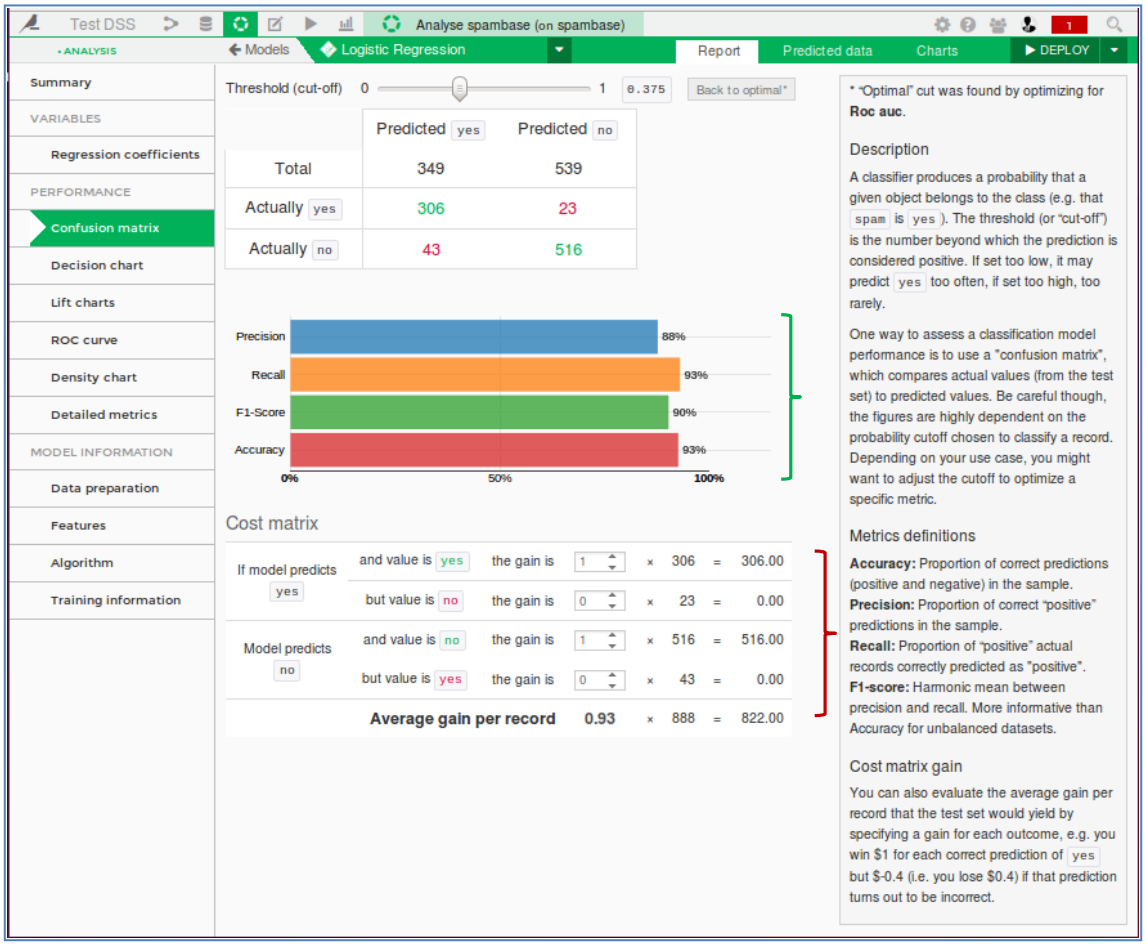
Regression Coefficients. Cet item nous affiche les coefficients de la régression.



Comme les variables sont automatiquement centrées et réduites, ils permettent (en valeur absolue) de situer leur importance dans la régression. Pour les férus de statistique, nous

pouvons afficher les statistiques avancées avec DISPLAY ADVANCED STATS. Nous constatons ici que la fréquence du mot « hp » (wf_hp : word frequency hp) est la plus déterminante. Plus elle est fréquence, moins le message est suspect (le coefficient est négatif). C'est le caractère « $ » (cf_$: character frequency $) qui joue en suite, positivement c.-à-d. plus il est fréquent dans un message, plus ce dernier est suspect. Vient ensuite la longueur maximale de séquence de caractères en majuscule (capital run length longest). Etc.

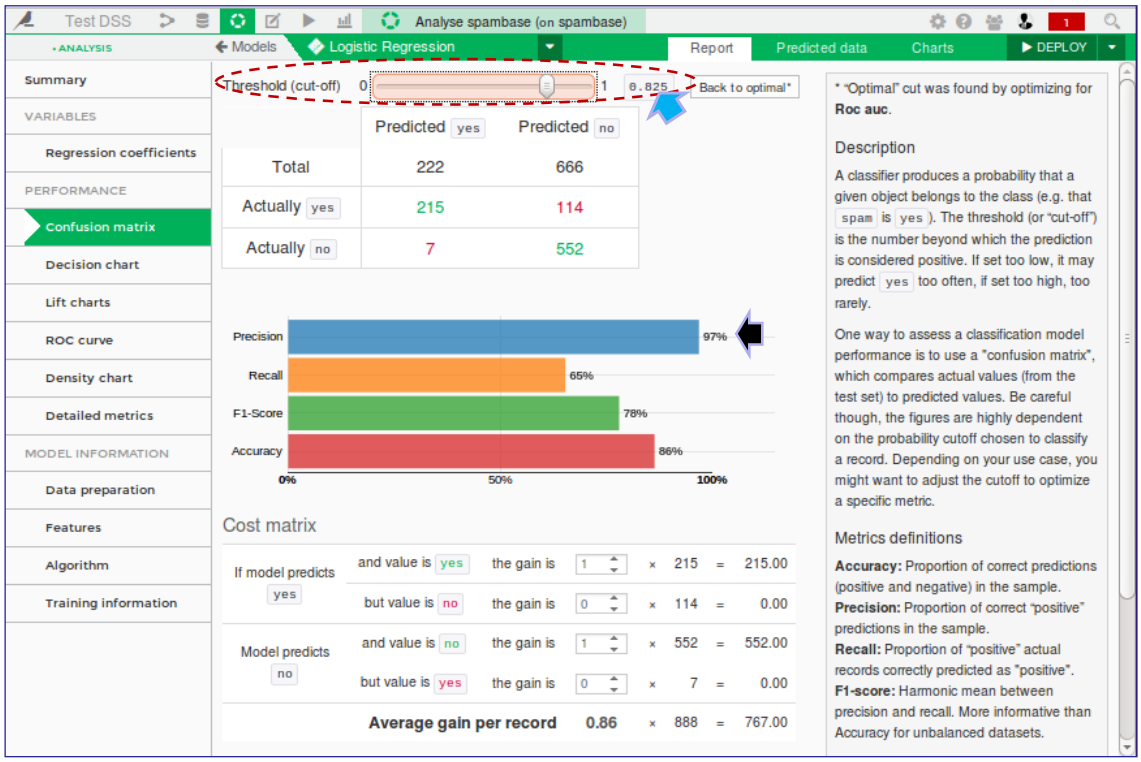
Confusion matrix. DSS nous présente la matrice de confusion **calculée sur l'échantillon test** et les indicateurs associés : précision, rappel, F1-score et taux de succès.



Nous pouvons modifier interactivement la matrice de coûts de mauvais classement pour mesurer l'impact des prédictions selon différents scénarios. Supprimer automatiquement un e-mail sain n'équivaut pas à laisser passer un spam. Si l'on sait chiffrer les coûts, nous pourrons tout de suite en évaluer les conséquences sur le gain moyen.

Confusion matrix (suite). L'outil propose un dispositif intéressant. Il est possible de **moduler le seuil d'affectation** de manière à favoriser le taux de vrais positifs (spam positif = spam yes dans le cas présent) ou, inversement, le taux de faux positifs. La valeur « optimale » fournie (0.375 dans notre exemple) est fixée en fonction de l'AUC de la courbe ROC[4] c.à.d. si le score est supérieur à 0.375, le message est prédit « spam », il est désigné « non spam » sinon.

Mettons que nous souhaitons réduire la suppression des messages sains pour éviter d'avoir des problèmes avec les usagers de la messagerie, quitte à laisser passer des spams[5]. Nous devons donc augmenter le seuil d'affectation c.-à-d. on ne prédit spam qu'à coup sûr, lorsque le score est très élevé. Nous le fixons à 0.825 pour obtenir une précision de 97% (qui pourrait être demandée dans le cahier des charge par ex.). Il suffit pour ce faire de déplacer le curseur THRESHOLD (CUT-OFF). Nous obtenons une nouvelle matrice de confusion.

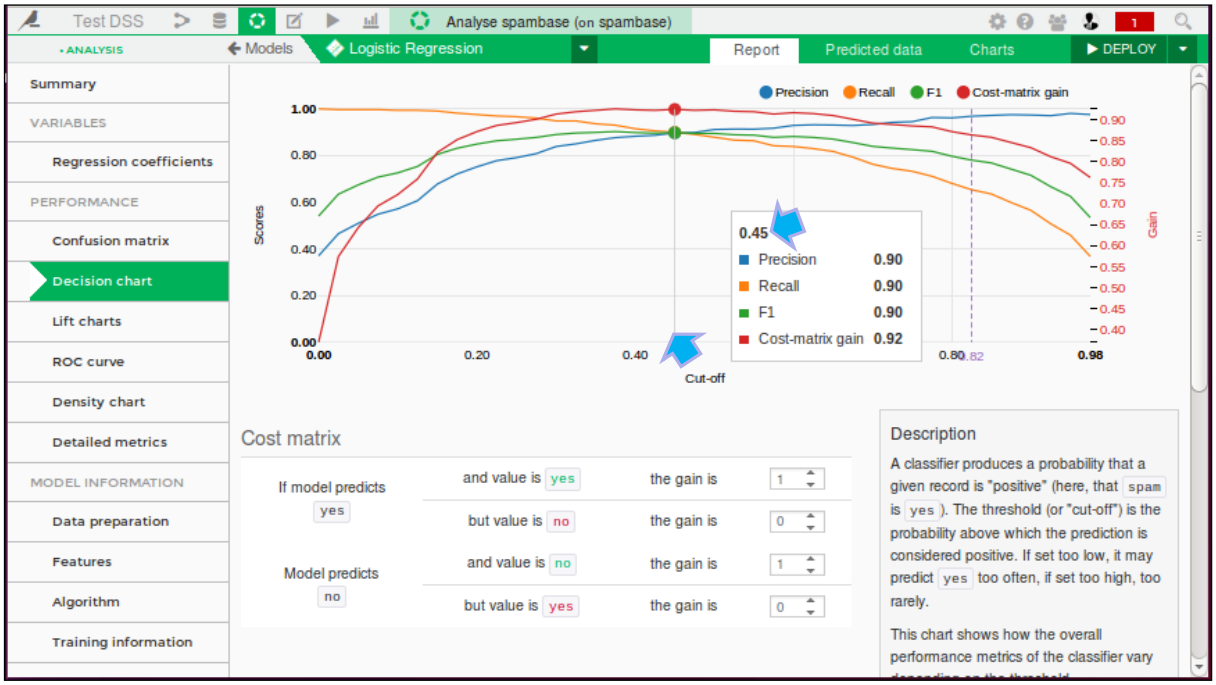


Bien sûr, le rappel est dégradé dans ce cas.

[4] J'avoue ne pas avoir saisi l'idée. L'AUC n'est pas dépendante du seuil d'affectation. En tous les cas, une valeur est fournie, différente du seuil usuel de 0.5 pour ce qui concerne la régression logistique.

[5] J'en parle de manière un peu plus détaillée dans : Tutoriel Tanagra, « Apprentissage – test avec Sipina », mars 2008.

Decision chart. Cet item fournit un graphique mettant en relation le seuil d'affectation (CUT-OFF) et les différents indicateurs d'évaluation des modèles. Cette vision globale nous permet de mieux mesurer les conséquences de nos décisions. Mettons que cette fois-ci, nous désirons optimiser le critère F1-SCORE[6]. En déplaçant le curseur dans le graphique, nous observons qu'un seuil à 0.45 permet d'obtenir un F1 de 0.90 et, dans ce cas, le rappel est égal à 0.90, tout comme la précision.

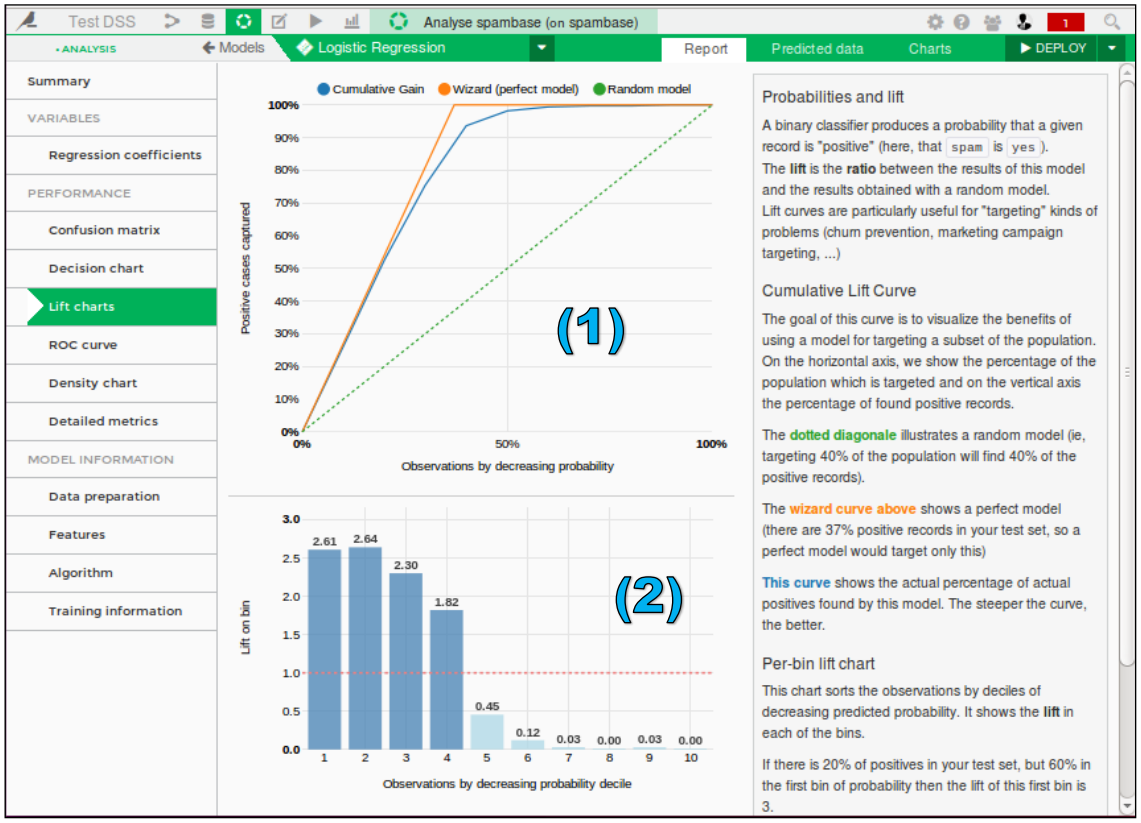


Lift charts présente la courbe de gain (courbe lift cumulé) (**1**). L'outil est précieux en ciblage marketing[7]. Sur la diagonale, nous avons le ciblage au hasard (nous attribuons un score aléatoire aux individus) ; nous avons également la courbe maximale théorique (tous les positifs présentent un score plus élevé que n'importe quel individu négatif). Plus on s'en rapproche, meilleur est le ciblage. Dans notre exemple, nous avons un excellent résultat.

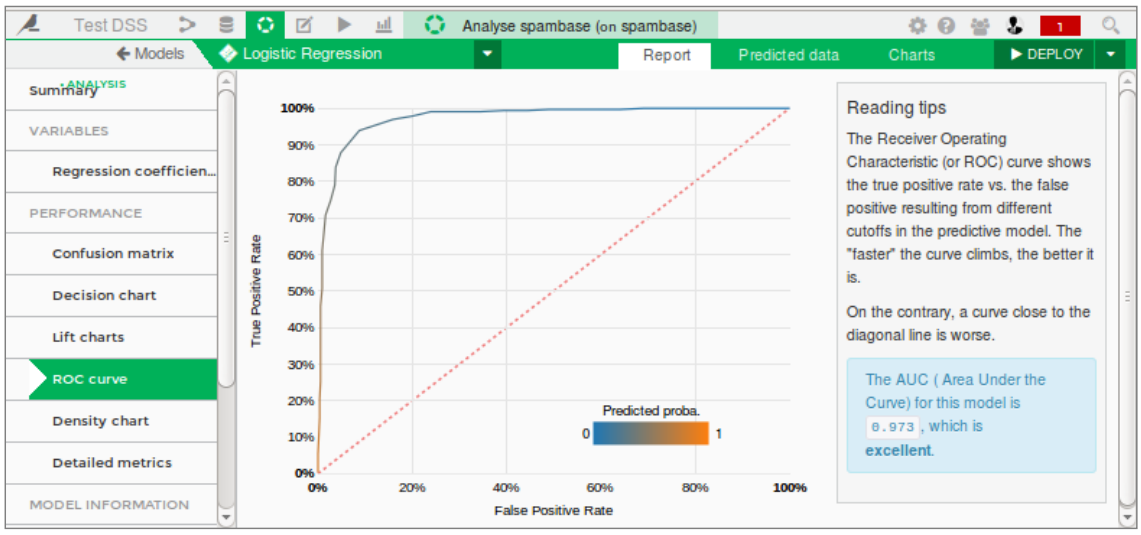
Une autre présentation moins usitée est disponible (**2**) : le lift pour chaque décile du score décroissant c.-à-d. la surreprésentation des positifs dans chaque intervalle de score. La valeur de référence est 1, matérialisée par la ligne rouge en pointillée.

[6] https://en.wikipedia.org/wiki/F1_score

[7] Ricco Rakotomalala, « Ciblage marketing - Construire la courbe lift » ; Tutoriel Tanagra, « Evaluation des classifieurs - Quelques courbes », octobre 2009.



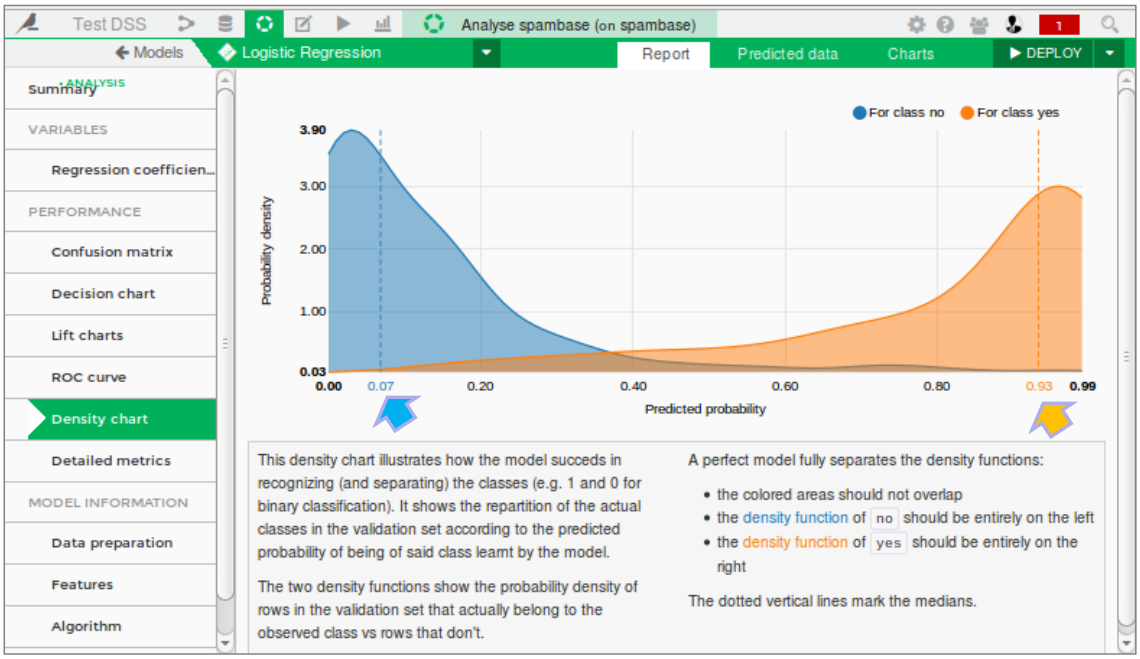
ROC Curve affiche la courbe ROC. L'outil est plus riche que la matrice de confusion (une matrice de confusion est un cas particulier de la courbe ROC pour un seuil d'affectation donné)[8]. Plus la courbe s'éloigne de la diagonale, meilleur est le modèle. Ce qui est le cas ici.



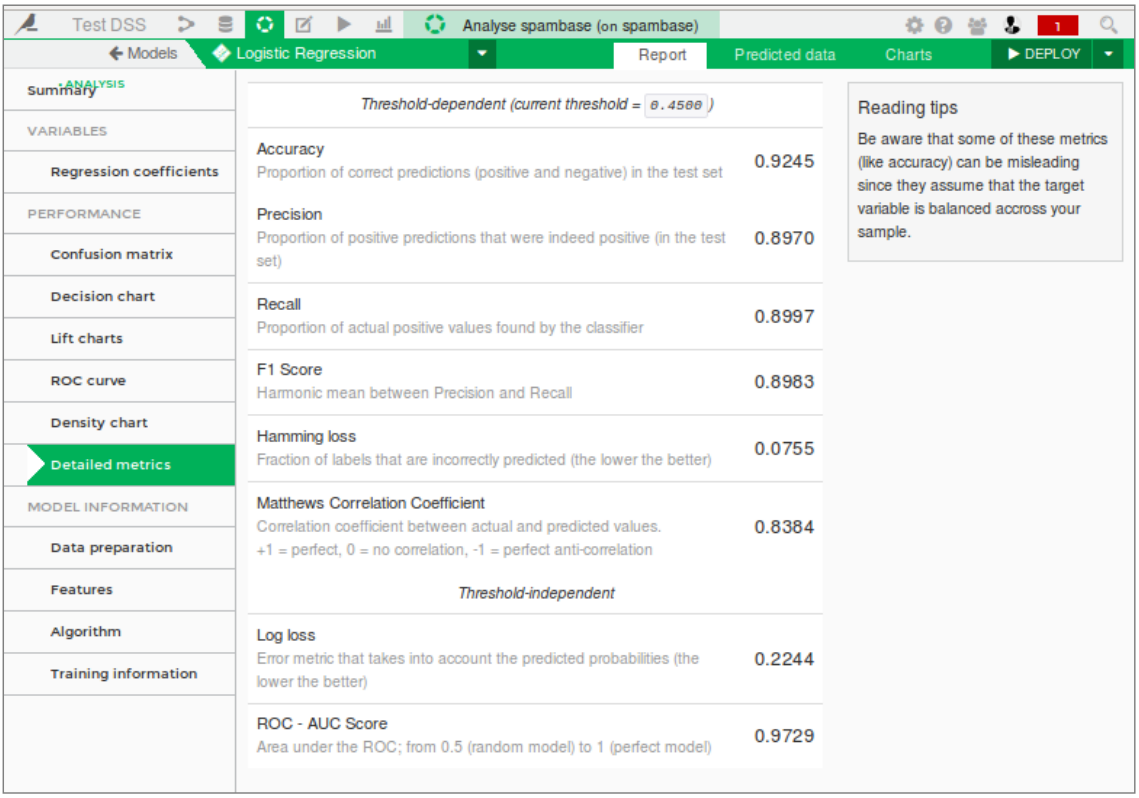
Bien qu'il y ait une certaine similitude avec la courbe de gain, rappelons que la courbe ROC répond à une finalité différente. La lecture n'est pas du tout la même.

[8] Ricco Rakotomalala, « Courbe ROC ».

Density chart indique la distribution lissée des scores conditionnellement à la classe d'appartenance (spam = yes ou no). Les médianes conditionnelles sont affichées. On voit à peu près l'idée qui guide ce graphique. Mais je préfère de loin le « decision chart » pour déterminer interactivement le seuil d'affectation par exemple.



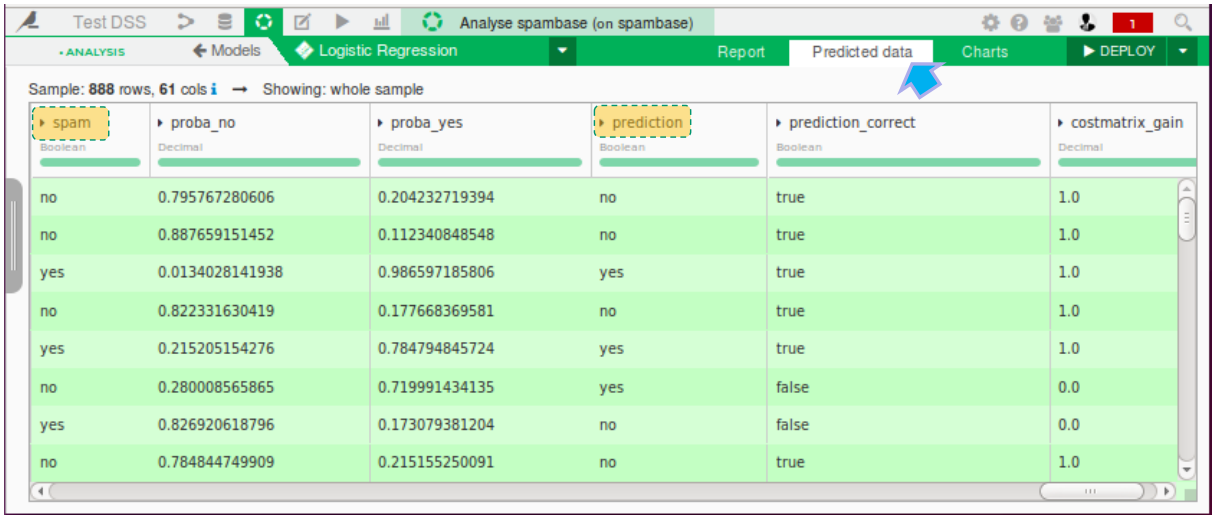
Detailled metrics récapitule les indicateurs d'évaluation du modèle sur l'échantillon test.



Les autres indications regroupées dans **MODEL INFORMATION** sont relatives à la préparation des données (la transformation des entier en décimales par ex.), le mode de subdivision des données en échantillons d'apprentissage (3713 obs.) et de test (888 obs.), et la description de l'algorithme et de ses paramètres (à ce propos, il ne semble pas qu'un procédé de sélection automatique de variables soit prévu pour la régression logistique).

3.4.3.2 Prédiction sur l'échantillon test

DSS affiche le détaille des prédictions sur l'échantillon test dans l'onglet **PREDICTED DATA**. Nous y observons pour les 888 observations, la valeur de la variable cible, les probabilités d'affectation aux classes, la classe prédite (qui tient compte du seuil d'affectation).

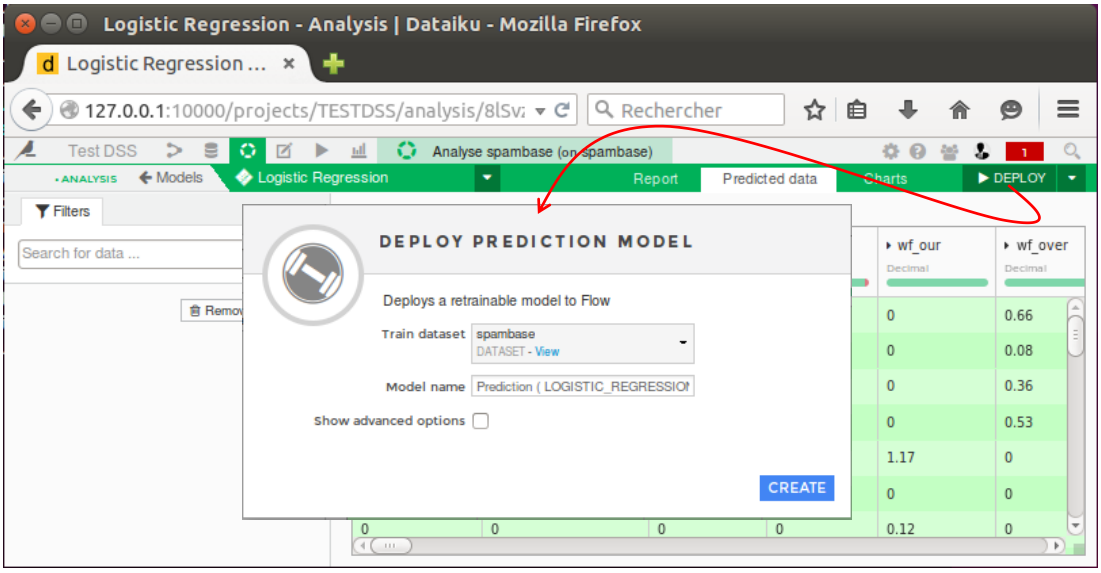


spam	proba_no	proba_yes	prediction	prediction_correct	costmatrix_gain
no	0.795767280606	0.204232719394	no	true	1.0
no	0.887659151452	0.112340848548	no	true	1.0
yes	0.0134028141938	0.986597185806	yes	true	1.0
no	0.822331630419	0.177668369581	no	true	1.0
yes	0.215205154276	0.784794845724	yes	true	1.0
no	0.280008565865	0.719991434135	yes	false	0.0
yes	0.826920618796	0.173079381204	no	false	0.0
no	0.784844749909	0.215155250091	no	true	1.0

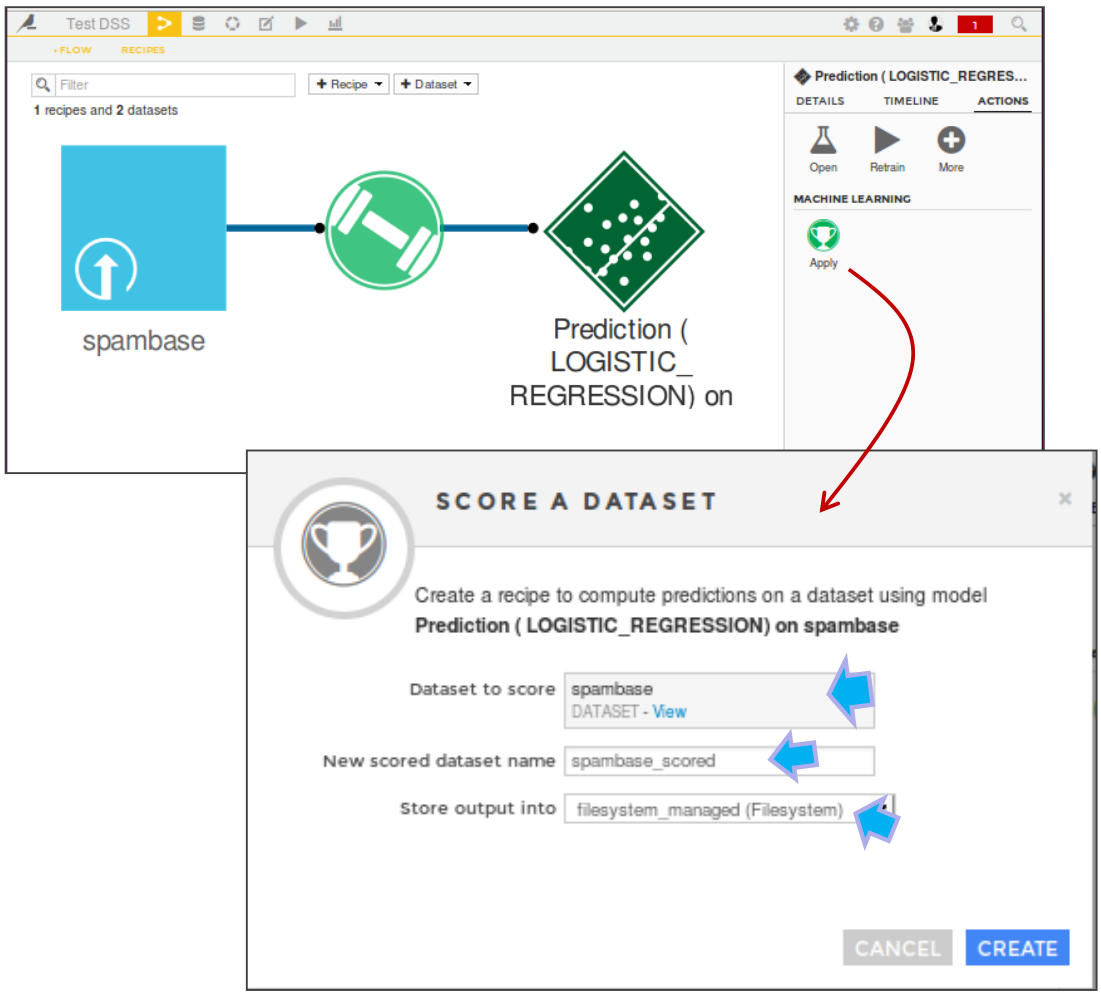
3.4.3.3 Déploiement

Une des principales finalités de l'analyse prédictive est le déploiement c.-à-d. l'application du modèle sur de nouveaux individus (une nouvelle base) pour lesquelles nous ignorons les valeurs de la variables cible. Par exemple, nous disposons d'une nouvelle liste d'e-mails. Nous en déduisons les propriétés (fréquence des mots, etc.). Nous aimerions identifier les messages délictueux en utilisant notre modèle prédictif. Le logiciel devrait donc affecter pour chaque nouvel individu la probabilité d'appartenance aux classes, et l'étiquette attribuée en appliquant le seuil optimisé durant la phase de modélisation.

Nous cliquons sur le bouton DEPLOY. Une boîte de dialogue apparaît permettant de spécifier les données d'apprentissage et le modèle à utiliser. Nous cliquons sur CREATE.

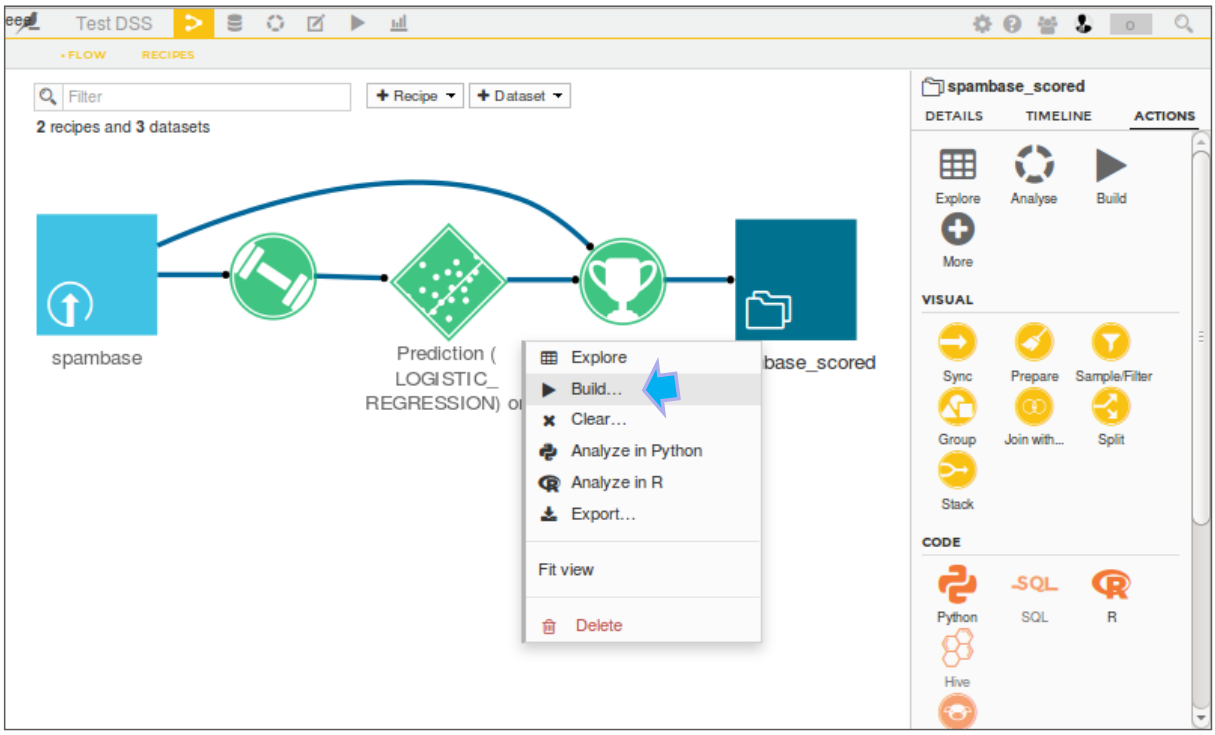


Le diagramme de traitements apparaît. Pour réaliser le déploiement (la prédiction) proprement dit, il faut cliquer sur l'icône APPLY. Une boîte de dialogue demandant la base à traiter (contenant les nouvelles observations) et le nom de la nouvelle colonne générée apparaît. La sortie sera stockée dans notre système de fichier interne.

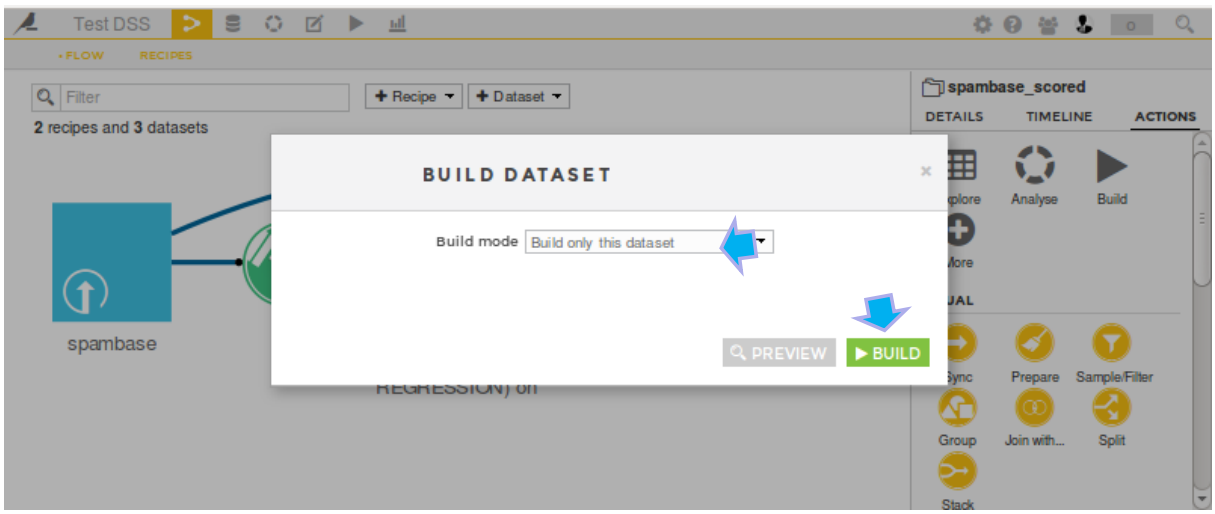


Nous utilisons les mêmes données que pour l'apprentissage dans ce tutoriel. Bien sûr, pour que la procédure prenne tout son sens, il aurait fallu importer la base contenant la description des nouveaux individus et l'indiquer dans DATASET TO SCORE.

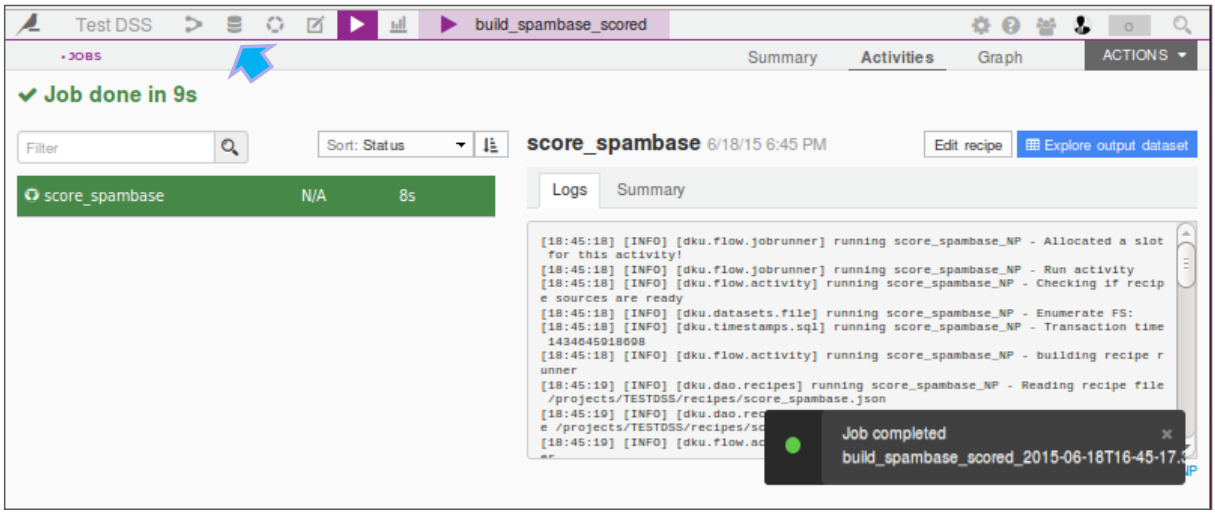
Nous validons avec CREATE. Le flux est mis à jour. Le fait de n'utiliser qu'une seule base est parfaitement identifiée ici. Pour finaliser la création de la base scorée, nous affichons le menu contextuel (clic bouton droit) de SPAMBASE_SCORED et nous sélection BUILD.



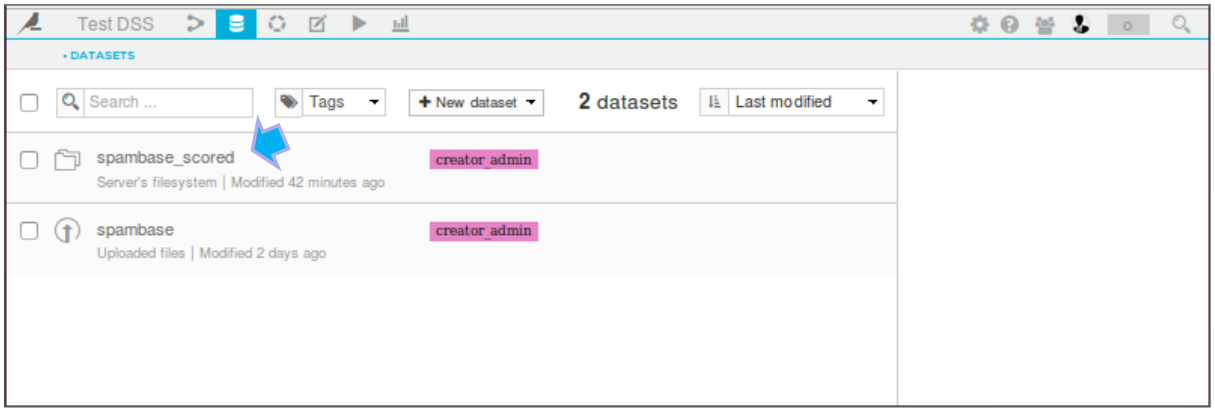
Dans la boîte de dialogue qui suit, nous confirmons ne vouloir construire que cette base. Nous validons avec le bouton BUILD.



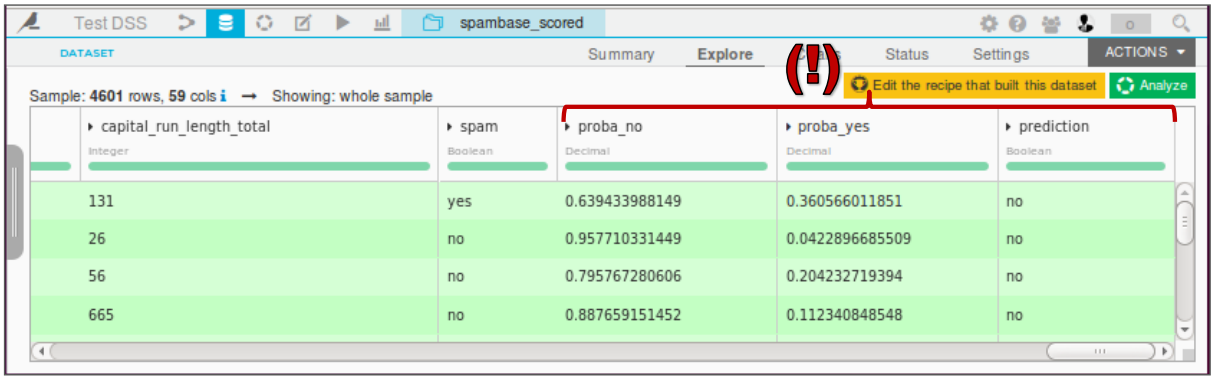
Un état d'avancement de la tâche est affichée jusqu'à sa finalisation.



Pour afficher les bases disponibles, nous cliquons sur l'icône dédiée (les disques empilés). La base issue du scoring (SPAMBASE_SCORED) peut être consultée.

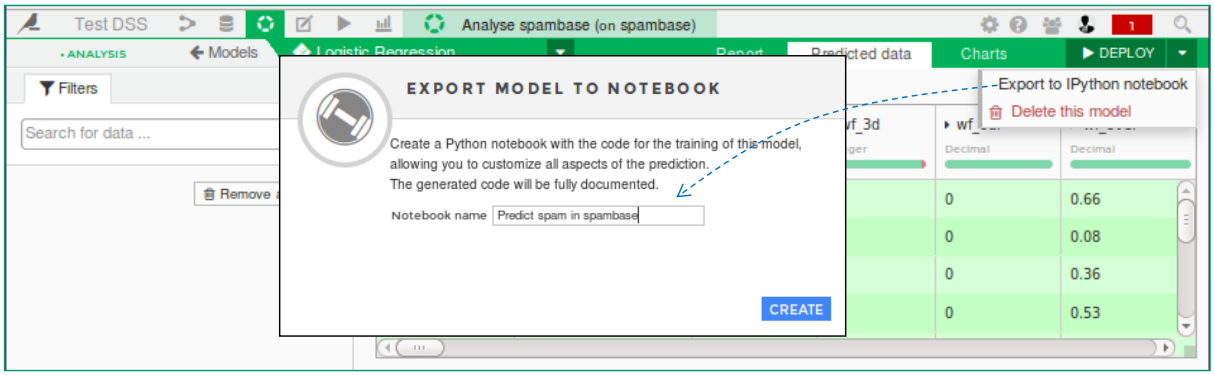


Nous cliquons sur l'item, les données apparaissent dans une grille avec comme nouvelles colonnes : les probabilités d'affectation et la prédiction. La colonne SPAM n'est pas censée être présente dans cette base, c'est pour cela que la colonne de vérification n'est pas affichée, contrairement à ce qui avait été réalisé sur l'échantillon test.

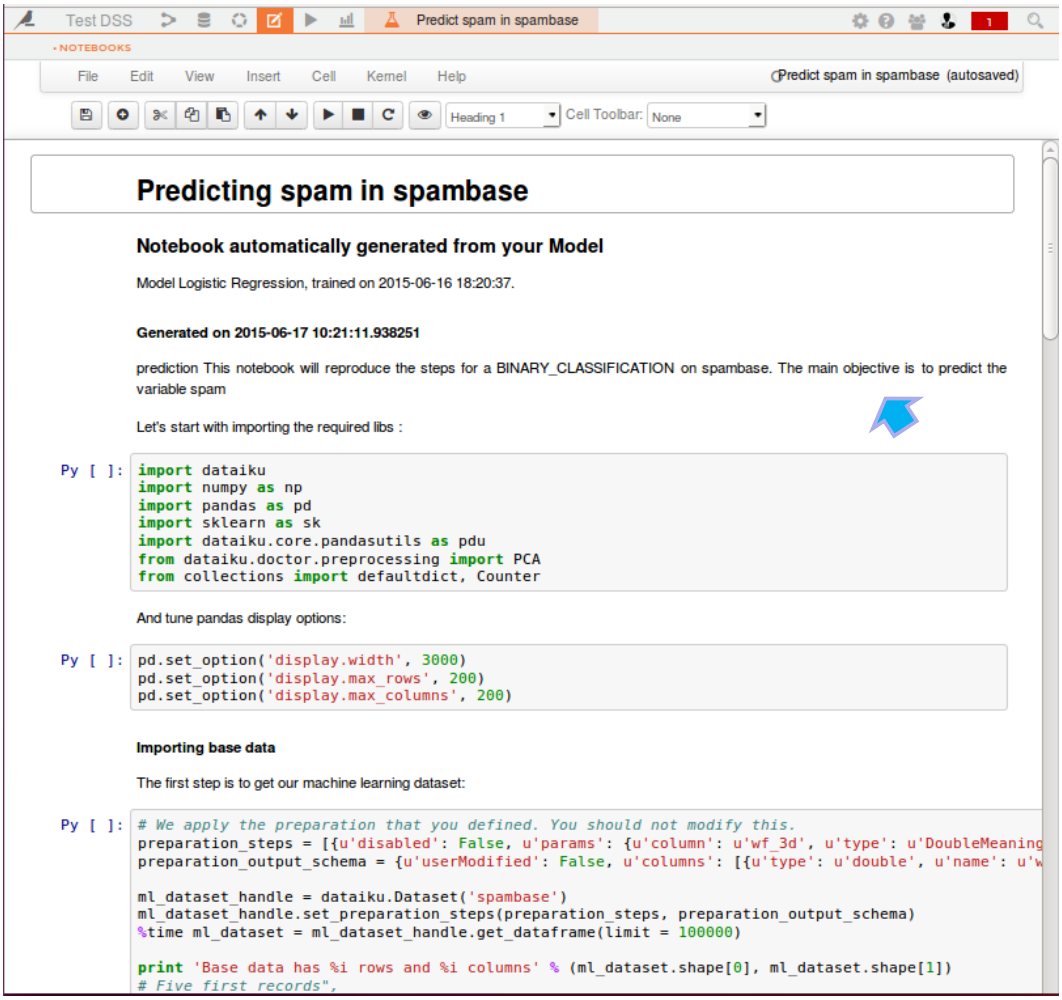


3.4.3.4 Exportation du modèle

La démarche d'analyse peut être retracée en langage Python. Cela peut être intéressant pour son automatisation et sa reproductibilité. Nous cliquons sur le bouton à côté de DEPLOY et nous activons EXPORT TO IPYTHON NOTEBOOK. DSS nous demande le nom du fichier.



Nous validons avec le bouton CREATE. Le projet s'affiche dans un éditeur où toutes ses étapes sont traduites en programme Python.

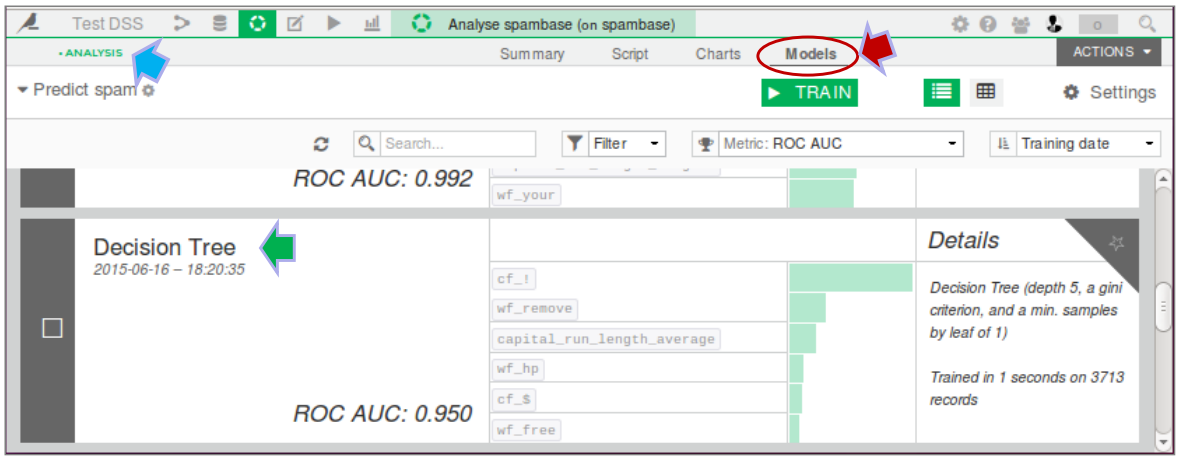


Le programmeur et les spécialistes peuvent s'en donner à cœur de joie pour affiner au mieux les opérations à chaque stade du processus.

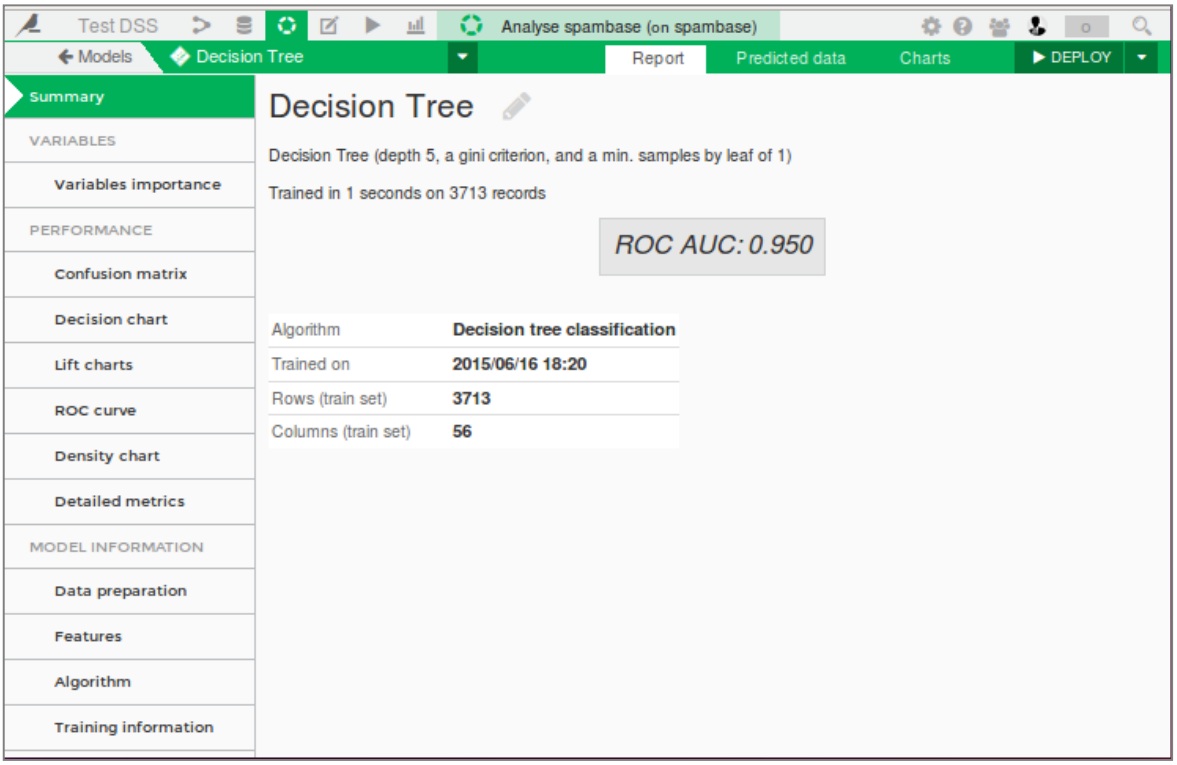
3.4.4 Inspection des modèles - Arbre de décision

Nous nous attardons exclusivement sur les sorties spécifiques aux arbres dans cette section. Les résultats relatifs aux performances et les prédictions sont de la même teneur quel que soit l'algorithme de machine learning utilisé.

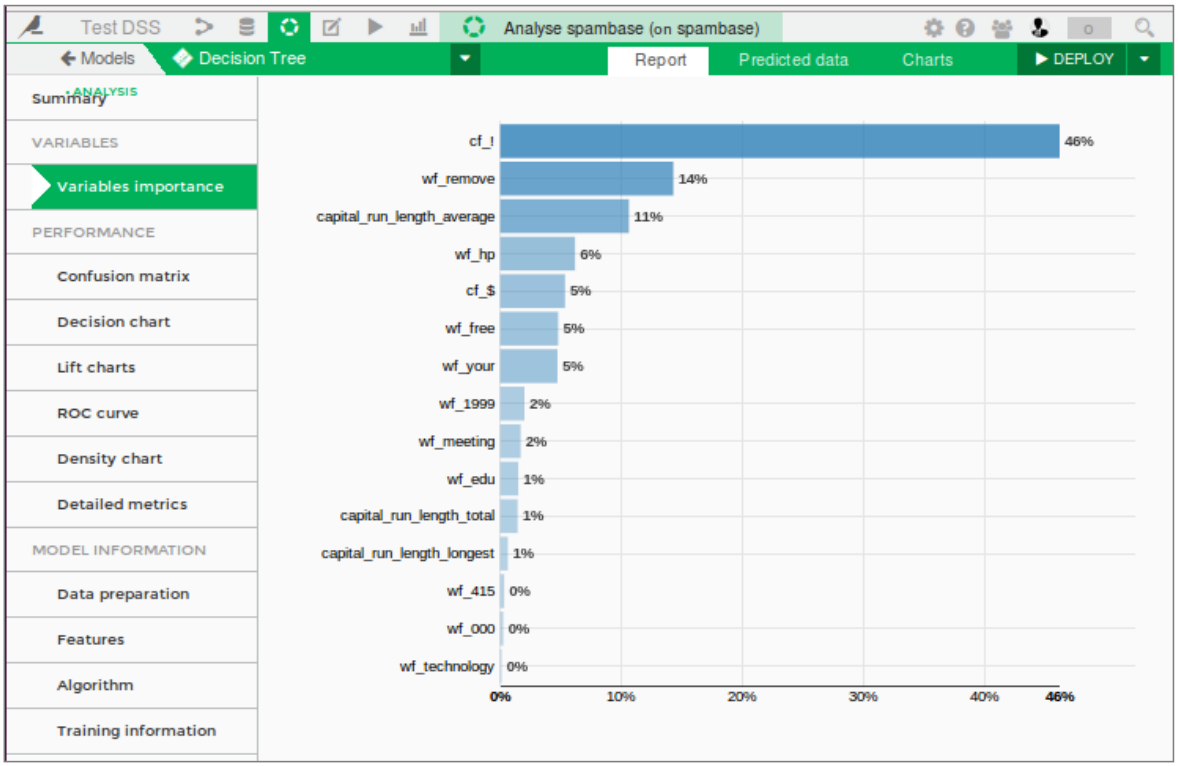
Nous revenons dans la liste des modèles et nous sélectionnons DECISION TREE.



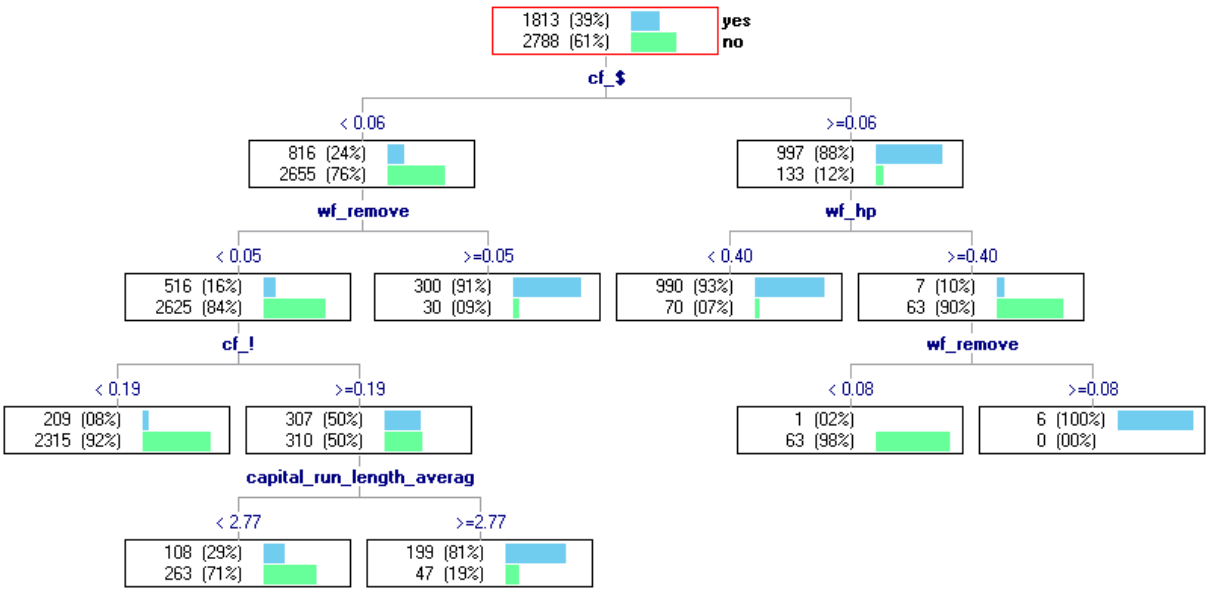
Nous retrouvons la présentation standard des modèles (cf. Régression logistique, section 3.4.3).



Avec VARIABLES IMPORTANCE, nous visualisons l'impact des variables dans la prédiction.



Je n'ai pas su trouver dans le logiciel ni l'arbre de décision ni les règles qui en découlent. Pour me faire une idée, j'ai lancé la construction de l'arbre sous **SIPINA**, voici une solution possible calculée sur l'ensemble de la base, non partitionnée en apprentissage et test.



On comprend mieux l'énumération dans VARIABLES IMPORTANCE de DSS. Le cas de « capital run length average » est assez particulier. La variable n'apparaît que dans la partie

basse de l'arbre, pour une seule segmentation. En réalité, lorsque nous inspectons chaque sommet, on se rend compte qu'elle est souvent en bonne position (au sens de la mesure de qualité de segmentation) dans la liste des variables candidates. C'est la raison pour laquelle une importance élevée lui est attribuée.

4 Autres fonctionnalités de DSS

Bien sûr, le logiciel propose d'autres fonctionnalités. Il est difficile de tous les énumérer ici. Pour le lecteur désireux d'en savoir plus, un guide de référence est accessible en ligne : http://doc.dataiku.com/dss/latest/

5 Conclusion

Data Science Studio montre bien que nous sommes à une époque charnière. Le fond est ultra-classique. Il ne peut pas en être autrement. Il s'agit bien de mettre en œuvre des méthodes de modélisation statistique ou de machine learning (d'obédience informatique) sur des données. C'est le mode opératoire qui change. Les usages évoluent avec les besoins et les progrès technologiques (accès aux bases NoSQL, accès à des bases stockées sur des architectures distribuées, etc.). On notera ainsi que le sondage annuel du portail mondial KDnuggets indique la percée de nouveaux logiciels dans le Top 10[9]. Le panorama est peut-être en train de se redessiner. De nouveaux acteurs et de nouvelles caractéristiques se font jour. Cela ne veut pas dire pour autant que les outils historiques soient à ranger dans les placards. Je ne le pense pas. De nouveaux paradigmes d'utilisation émergent simplement. Il importe à nous, utilisateurs, de cerner suffisamment nos attentes pour choisir à bon escient. Il ne faut pas céder à des effets de mode et se retrouver avec des outils qui ne nous correspondent pas.

Je constate en tous les cas que le thème des outils suscite plus que jamais de nombreuses interrogations sur le site KDnuggets : « Which Big Data, Data Mining, and Data Science Tools go together ? », « R vs Python for Data Science: The Winner is… », etc. Tout ça est passionnant.

[9] KDnuggets Polls, « Analytics, Data Mining, Data Science software/tools used in the past 12 months », May 2015.