# Outil de Profilage de données

#### Resolution Compliance Consulting Inc.

Par Mouhameth Takha Faye

Le 03 Juin 2020

Table des matières

[Outil de Profilage de données 1](#_Toc42515487)

[1. Objectif : 3](#_Toc42515488)

[2. Plan de développement : 4](#_Toc42515489)

[3. Persona d’utilisateurs : 5](#_Toc42515490)

[4. Exigences fonctionnelles : 5](#_Toc42515491)

[5. Design : 7](#_Toc42515492)

[**Module 1 :** 8](#_Toc42515493)

[**Module 2 :** 8](#_Toc42515494)

[**Module 3 :** 9](#_Toc42515495)

[6. Autres Exigences : 9](#_Toc42515496)

# Objectif :

Le but de ce projet est de créer une application de profilage de données qui permettra d’automatiser certaines tâches redondantes dans le processus d’extraction et d’analyse descriptive. L’outil doit pouvoir collecter des informations de différentes sources (fichier CSV et base de données), calculer les principales caractéristiques de la table de données, et écrire les résultats statistiques sur un fichier texte et les graphiques sur des fichiers d’images.

L’objectif principal de ce document est de présenter la méthodologie de développement et les exigences fonctionnelles pour les faire valider. Les points suivants seront traités en détail :

* Le plan de développement et les différentes périodes de livraison.
* Les exigences fonctionnelles et les types d’utilisateurs.
* Le diagramme d’interaction pour une meilleure compréhension du fonctionnement de l’application.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **DESCRIPTION DES TÂCHES** | **ASSIGNÉ** | **DÉDUT** | **FIN** | **DURÉE**  **EN JOUR** | **STATUS** |
| * Générer un fichier texte contenant les résultats statistiques. * Générer fichier d’images JPEG séparément. | Mouhameth | **27 Mai** | **10 Juin** | 10 | Complété |
| **Réunion le 10 Juin** |  |  |  |  |  |
| * Ajouter une fonctionnalité pour contrôler la taille des bins d’histogrammes et peut être d’autres fonctionnalités au besoin pour donner plus de contrôle à l’utilisateur. | Mouhameth | **10 Juin** | **17 Juin** | 5 | Complété |
| **Réunion le 17 juin** |  |  |  |  |  |
| * Trouver une meilleure solution pour que l’utilisateur spécifie le chemin des fichiers sans toucher aux codes. | Mouhameth | **17 Juin** | **24 Juin** | 5 | Complété |
| **Réunion le 24 Juin** |  |  |  |  |  |
| * Ajouter le PSI et s’assurer que tout le système est stable avant livraison. | Mouhameth | **24 Juin** | **30 Juin** | 4 |  |
| **Réunion finale le 30 Juin : Livraison** |  |  |  |  |  |

# Plan de développement :

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Nom du Projet | Outil de profilage de données | | | Superviseur | Jean-François Isabelle |
| Date de début | 27 Mai 2020 | Date de fin | 30 Juin 2020 | Progression | 20% |

Les phases de développement sont illustrées sur le tableau suivant. Les tâches pour chaque phase ont été listées par ordre de priorité en débutant par la tâche la plus importante mentionnée sur la première ligne du tableau.

Tableau 1: Plan de développement

# 3. Persona d’utilisateurs :

Afin d’alléger le projet et de minimiser le temps de développement, il a été convenu de concevoir une première version de l’application qui ne nécessite pas une interface d’utilisateur graphique. Cela impliquerait donc que les principaux utilisateurs de cette application posséderont des habiletés techniques et devront être en mesure d’écrire des scripts de Python ou de commande Shell au besoin pour effectuer des tâches ou configurer certaines fonctionnalités de l’application.

# 4. Exigences fonctionnelles :

Pour chaque colonne de la table de données, l’application doit accomplir les fonctionnalités et fournir les informations statistiques mentionnées ci-dessous. Il faut néanmoins noter qu’au fur et à mesure que le développement progresse d’autres fonctionnalités pourront être ajoutées.

|  |  |
| --- | --- |
| **Fonctionnalités** | **Descriptions** |
| **Sommaire de la table de données** | * Nombre de variables * Nombre d’observations * Nombre de variables catégoriques * Nombre de variables numériques * Nombre de valeurs manquantes * % de valeurs manquantes |
| **Variables Catégoriques** | * Nombre de catégories * Valeurs distinctes par catégories * Valeurs manquantes par catégories * Valeurs maximales et minimales par catégories |
| **Variables Numériques** | * Moyennes * Maximales * Minimales * Valeurs manquantes * % de valeurs manquantes * Valeurs distinctes * Écart-Types * Médianes * 5-th et 95-th centiles |
| **Visualisation de données** | * Graphes d’histogrammes pour les variables numériques * Graphes en barres pour les variables catégoriques |
| **Exportation de fichiers** | * Fichiers textes pour les résultats statistiques * Fichiers images pour les graphes |
| **Contrôle** | * Taille des bins d’histogrammes * Annotations et légendes des graphiques * Contrôle des quantiles (centiles, quartiles, déciles) |
| **Calculer PSI** | * Mise à jour des données |

Tableau 2 : Liste des fonctionnalités

# 5. Design :

Python et Pandas offrent de nombreuses fonctions pour profiler des données mais la plupart des commandes doivent être émises individuellement et le processus manuel est très répétitif et fastidieux. Pour développer l’outil et lui permettre de fonctionner automatiquement et en si peu de scripts, il a fallu faire appel au Data Profiler qui est un moyen efficace d’accéder facilement à toutes les informations de profilage tout en combinant de nombreuses fonctionnalités du paquet et des fonctions génériques de Python.

Le design de l’application est relativement simple ; il y’a trois modules et un fichier de configuration qui communiquent ensemble pour exécuter l’application au complet.

Le schéma ci-dessous est une illustration simplifiée de son fonctionnement.

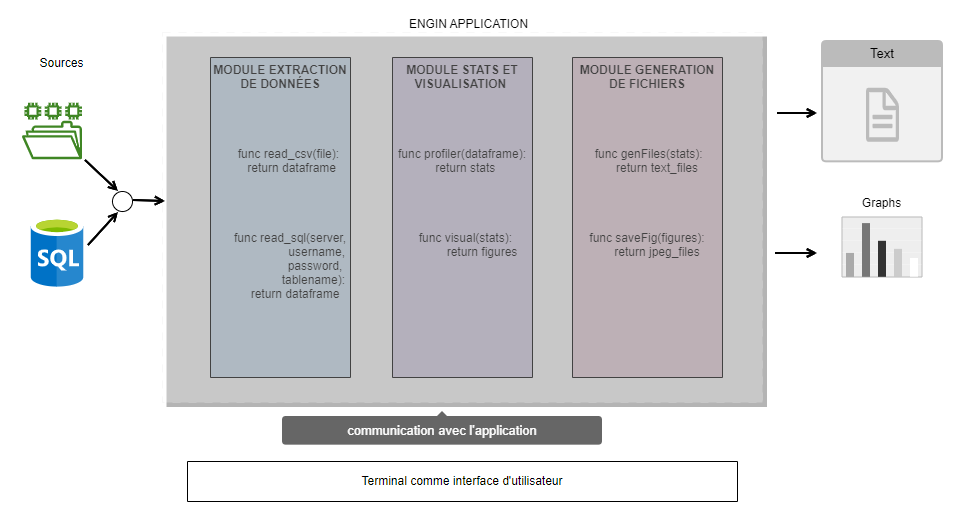


Figure 1: Schéma d’interaction

**Config.py : Fichier de configuration**

Le fichier Config.py contient tous les paramètres de configuration. Il permet de spécifier le chemin du fichier CSV et de sauvegarder les accès d’une base de données. Ce fichier ne contient pas de code et c’est le seul fichier que l’utilisateur aura accès pour configurer l’application.

**Module 1 : Processus d’extraction de données**

Le module extraction de données contient deux méthodes :

* La méthode **read\_csv** prend la localisation du fichier CSV comme argument et retourne un objet de pandas dataframe.
* La deuxième méthode **read\_sql** prend comme argument les accès de la base de données notamment le nom du serveur, le nom d’utilisateur de l’admin, le mot de passe et le nom de la table de données et retourne ainsi un objet pandas dataframe. Ces paramètres devront être insérés par l’utilisateur dans le fichier config.py.

**Module 2 : Calcules Statistiques**

* Ce module contient la méthode **profiler** qui effectue le sommaire des calcules statistiques pour chaque colonne de la table de donnée. Elle prend comme argument l’objet pandas dataframe retourné par l’une des méthodes du module 1 et génère ainsi à son tour une liste d’informations statistiques.
* Une deuxième méthode nommée **visual** prend le relais pour créer des graphes et les sauvegarder en fichier JPEG sur le disque. La méthode prend comme entrée la sortie de la méthode profiler.

**Module 3 : Exportation de fichiers**

Ce module génère des fichiers textes sur le disque en utilisant une fonction intégrée de Python nommée OS.

# 6. Autres Exigences :

Les fonctionnalités qui entrent en jeu pour assurer la fiabilité et la stabilité de l’outil sont nombreuses et elles devront toutes être implémentées éventuellement. Voici quelques-unes qu’on n’a pas pu couvrir sur ce document :

* **Gestion des exceptions :** Gérer des types de fichiers ou de base de données différentes, autres CSV et SQL.
* **Portabilité** : L’outil a été développé sur un système d’opération Windows, des erreurs peuvent apparaitre si on l’utilise sur Unix ou MacOs.
* **Unit-Testing**: Il faudra aussi avoir un module qui performe un test sur chaque méthode de l’application pour s’assurer que les résultats fournis sont exacts.
* **Sécurité des données** : Implémenter des exigences de rétention de données.