Compte Rendu

Sonny Klotz - Jean-Didier Pailleux - Malek Zemni

Interface de chargement, de contrôle
et d'analyse statistique des données
pour la constitution d'un graphe de flux

22/05/2017

Module *Projet*

Table des matières

Introduction

Motivations du projet : un mix de - Description Sujet (notre premier oral) - Intro et partie 1 cahier charges

Plan : - archi appli (cahier charges) + données transitent + fonctionnalités - langages de programmation : découlant des contraintes de l'appli - bilan technique - organisation interne

Documents fournis en plus du code : - Manuel - Documentation

-> 1 page (Sonny)

1 Architecture de l'application

1.1 Organigramme et données échangées

Phrase de présentation -> diagramme -> légende (non modifiée, on la met en accord avec le dernier produit quand on aura finit)

1.2 Format du fichier CSV

une demi-page

1.3 Fonctionnalités des modules

Les fonctionnalités de notre produit final (pas ceux du cahier des charges)

-> entre 3 pages (pas plus) pour la partie (Sonny)

2 Contraintes et Langages de programmation

2.1 Contraintes

Le développement de l'application s'accompagne de contraintes imposées. Celles-ci établie par DCbrain au commencement du projet. On peut en énumérer trois, qui sont les suivantes :

Contrainte	Fiche

1. Le produit doit fournir une application web	**Description :** notre produit sera une application fonctionnant sur un navigateur web, appelée ***applet***. **Justification :** assure une très grande portabilité et fournit à l'utilisateur une interface interactive. **Critère de satisfaction :** on peut lancer l'application sur un navigateur web.
2. Le produit doit être développé avec un langage de programmation compatible avec l'analyse de données	**Description :** le langage de programmation choisi doit inclure des bibliothèques qui permettent d'analyser les données (taches de data mining et de machine Learning). **Justification :** permettre une analyse de données efficace. **Critère de satisfaction :** on peut effectuer une analyse descriptive de données.
3. Le produit doit fournir une API d'analyse de données en sortie	**Description :** l'application doit intégrer des API d'analyse descriptive de données qui pourront être livrées en sortie au client. **Justification :** permettre une réutilisabilité des fonctionnalités majeures du produit. **Critère de satisfaction :** on peut exporter une API d'analyse de données en sortie.

2.2 Choix langages et frameworks

Les contraintes appliquées au projet ainsi que les exigences définies nous ont permis de fixer notre choix sur un langage de programmation : ***Python***.
Ce choix est justifié par le fait que ***Python*** soit compatible pour l'Analyse Descriptive de Données, puisqu'il est doté de nombreux modules permettant d'effectuer du calcul scientifique sur des données, donc l'analyse de données. De même, il est compatible avec le développement d'applications web, grâce aux nombreux frameworks disponibles.
Pour ce dernier, notre choix s'est porté sur ***Flask***, qui s'efforce d'être le plus simple possible pour une prise en main et le plus petit possible. À l'opposé de Django ou autres, il est donc plus abordable pour une première approche sur le développement d'applications web.

L'application devant être une applet intégrée dans une page web pour s'exécuter, le recours à des langages de balisage comme ***HTML*** pour la présentation et ***CSS*** pour la mise en forme de ces pages web étaient nécessaires.

Ensuite, pour permettre à l'application de faire communiquer les pages web avec un serveur web sans occasionner le rechargement de la page, nous avons eu recours à ***Ajax***. Il s'agit d'un concept de programmation web reposant sur plusieurs technologies comme le ***Javascript***, qui lui se charge d'établir la connexion entre la page web et serveur. ***Javascript*** est également employé pour rendre les pages web interactives. ***Jquery*** est quant à lui utilisé pour faire le lien entre ***HTML*** et ***Ajax***.

Enfin, pour la génération de la documentation des fonctionnalités du programme, on utilisera le logiciel ***Sphinx*** qui se charge de généré la docstring à partir des commentaires présents dans le code python.

3 Partie technique du projet

Quoi marche (comment ca marche - à l'aide de la partie avant) QUoi marche pas (pb et et resolution si trouvee)

1 page, plus tard

4 Points délicats

Justification des points qui divergent par rapport aux specs et organigramme de départ

1 page, plus tard à moins que t'ai déjà des idées - (Peut-être rajouté les points délicats de flask/application web sur l'approche et tout ce qui va avec ?

5 Organisation interne du groupe

Module	Malek	Sonny	Jean-Didier	Total
Gestion des flux			x	1
Fenêtre choix fenêtre			x	1
Fenêtre rôle et choix colonne	x			1
Fenêtre résultats ADD		x		1
ADD qualitatives			x	1

ADD quantitatives discrètes		x		1
ADD quantitative continues		x		1
Vérification format fichier	x			1
Analyse contenu fichier	x			1

1 page On expliquera plus tard comment ça s'est déroulé concrètement par rapport à ce qui était prévu.

6 Coût

Module	**Nombre de lignes**	**Justification**	**Coût final**
Gestion des flux	15	Mise en forme du main et appel de l'application	
Fenêtre choix fichier	10 + 20	Fonctions pour : Drag& Drop + Système de fichiers	
Fenêtre rôle et choix colonne	5 + 20 + 10 + 10 + 20	Communication avec le module application + Affichage de l'interface+ lecteurs des valeurs + affichage des valeurs	
Fenêtre résultats ADD	10 + 3* 30	Envoie d'informations au module application + Construction des graphe pour l'ADD pour les 3 types d'analyse	
ADD qualitatives	20 + 20*3	Application des formules pour les calculs de fréquences et d'effectifs + calcul des valeurs pour la construction de 3 graphes	
ADD quantitatives discrètes	60 + 20*2	Application des formules attaché à l'analyse quantitative discret + calcul des valeurs pour la construction de 2 graphes	
ADD quantitatives continu	20 + 10 + 10 + 5 + 20*2	Parcours + choix précision classe d'intervalle + écriture + communication avec les modules+ calcul des valeurs pour la construction de 2 graphes	

Vérification format fichier	30	Ouverture fichier + vérification si ouverture en lecture + présence de texte formaté ou non	
Analyse contenu fichier	20 + 5 + 25 + 10	Recopie et vérification + initialisation de la structure+ Parcours du fichiers avec condition + Fonction pour donner nom et type de colonne	
Coût Total	**565**	**Estimation totale du coût**	

Conclusion

1 demi-page, plus tard