

Comment créer un réseau de neurones?

Recherche



Projet UML



ClassiPy V1.0.0 – Ilyas El Amrani & Mohamed El Jaouhari
Encadré par : Pr. Pr. Lamghari Nidal

Sommaire :

Le bon enchainement se traduit en une excellente communication.

- Introduction

- 1ère Version
- 2ème Version
- 3ème Version

- Maquette
- Implémentation
- Démo

- Bibliographie
- Remerciement

ÉQUIPE DE PROJET



✕ □ — Ilyas El Amrani

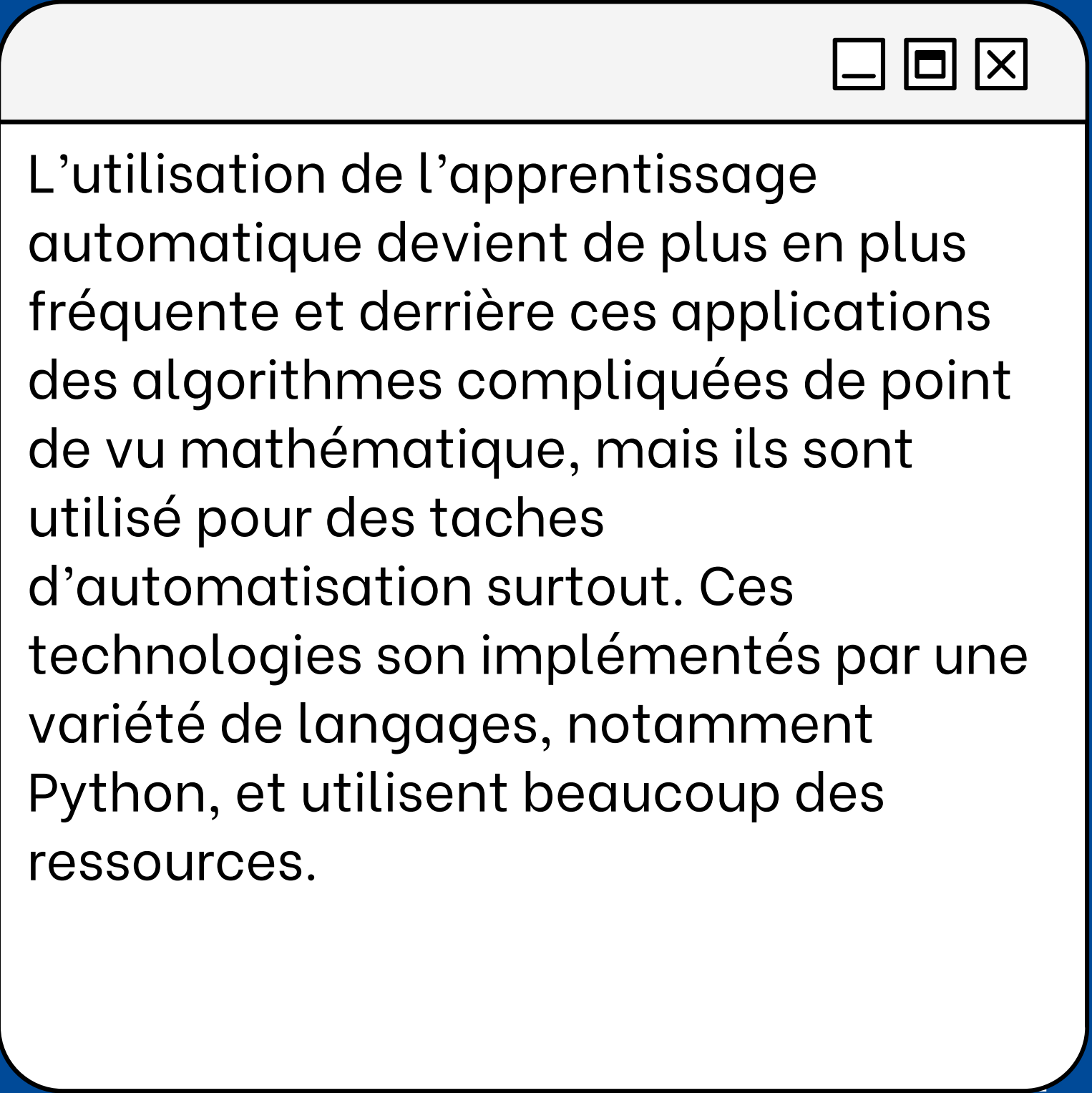


✕ □ — Mohamed El Jaouhari





Introduction



L'utilisation de l'apprentissage automatique devient de plus en plus fréquente et derrière ces applications des algorithmes compliquées de point de vu mathématique, mais ils sont utilisé pour des taches d'automatisation surtout. Ces technologies son implémentés par une variété de langages, notamment Python, et utilisent beaucoup des ressources.



ClassiPy

Dans notre application **ClassiPy**, nous essaierons de faciliter aux utilisateurs, surtout ceux qui n'aime pas coder mais sont passionnés par ce domaine la création, l'entraînement, le test, l'utilisation et la gestion de deux types de réseaux de neurones, tous on donnant l'utilisateur à chaque fois le choix des paramètres, de jeu de données et l'avertir en cas d'erreur ou de mauvaise utilisation de notre application simple.



II. 1ère Version

Que ClassiPy peut vous offrir et comment ?





III. 2ème Version

Classify? un peut plus clairement svp !





IV. 3ème Version

Et encore ? Que spécifier encore ?





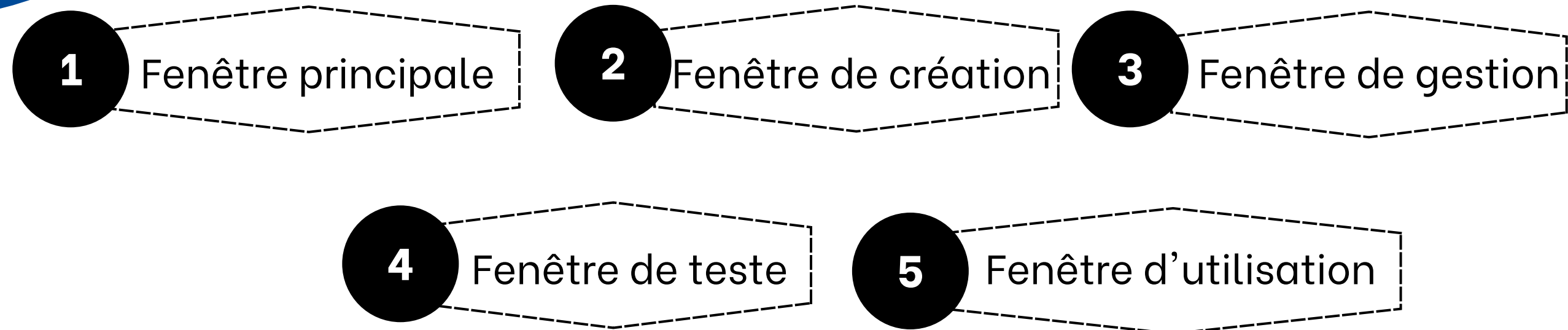
V. Maquette

Là commence le fun ? Huh



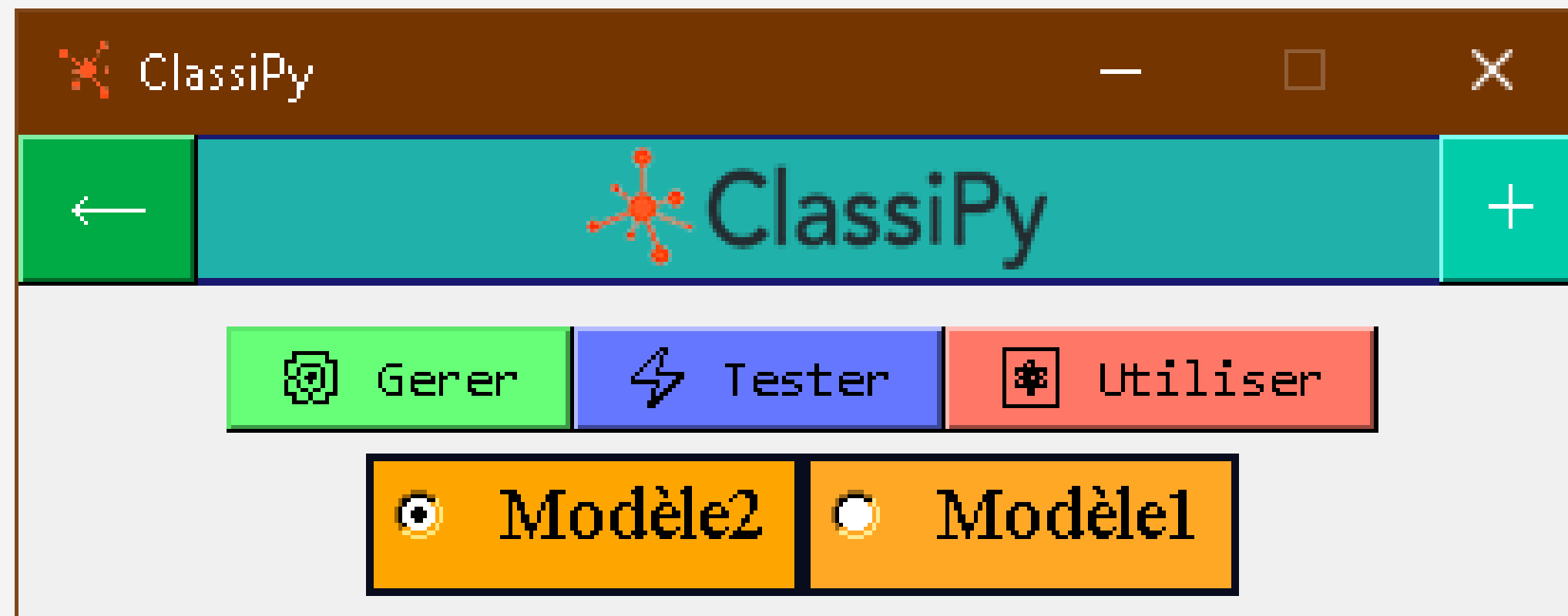
La Maquette

Simplement réalisé avec Python Tkinter, peut vous faire mal aux yeux, mais ça vous aidera pour créer vos réseaux de neurones pratiquement et facilement !



Maquette

Fenêtre Principale



Affiche les modèles disponibles et permet d'accéder aux différentes fonctionnalités de programme

Maquette

Fenêtre de création

Création du modèle et entraînement

← ClassiPy →

Nom du Réseau :

Description du réseau :

Nom du Créateur :

Type de réseau :

fonction d'activation :

fonction d'apprentissage :

Fichier d'entraînement : Selectionner

Colonne Cible :

Créer et trainer le modèle

En cliquant sur le button '+' vous pouvez facilement créer et entraîner votre modèle

Maquette

Fenêtre de gestion

Gestion du modèle : Modèle2

← ClassiPy +

Nom du Réseau :	Modèle2	Date de création :	06-12-2022 16:56:09
Description du réseau :	600epochnoscali ng	Date de modification :	18-12-2022 00:19:02
Nom du Créateur :	ilyas	Fonction d'activation :	sigmoid
Type de réseau :	P.M.C.	Fonction d'activation :	Adam
Nombre de couches :	5	Colonne de décision :	species
		Nombre de tests :	0

✓ Confirmer ⓧ Supprimer 📄 Exporter

Permet de modifier et afficher les information sur le modèle, même de l'exporter ou le supprimer.

Maquette

Fenêtre de teste

The screenshot shows a software window titled "Tester le modèle : Modèle2". The window has a brown title bar with standard minimize, maximize, and close buttons. Below the title bar is a teal header bar containing a back arrow on the left, the "ClassiPy" logo in the center, and a plus sign on the right. The main content area has a light gray background and contains the following elements:

- The title "Tester le modèle : Modèle2" in a large, bold, black monospace font.
- A label "Data du teste :" followed by a white text input field and a gray "Importer" button.
- A green "Tester" button with a circular arrow icon.
- A blue "Exporter" button with a document icon.
- A blue "Afficher l'historique" button with a list icon.
- A red "Supprimer l'historique" button with a trash can icon.

Permet de tester et afficher l'historique des testes du modèle, même de les exporter ou les supprimer.

Maquette

Fenêtre d'utilisation

Utilisation du modèle : Modèle2

← ClassiPy +

Vous utiliser maintenant le modèle : Modèle2

☐ fichier .csv

☒ 1 enregistrement

enregistrement :

Évaluer

Utilisation du modèle : Modèle2

← ClassiPy +

Vous utiliser maintenant le modèle : Modèle2

☒ fichier .csv

☐ 1 enregistrement

Data du testing : Importer

Évaluer

Extraire les données

Permet de générer des prédictions sur des enregistrements de données ou un seul.



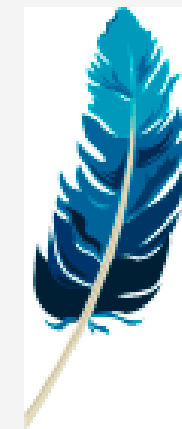
VI. Implémentation

Coder, coder, copier, coder...

1.Module et Packages utilisés



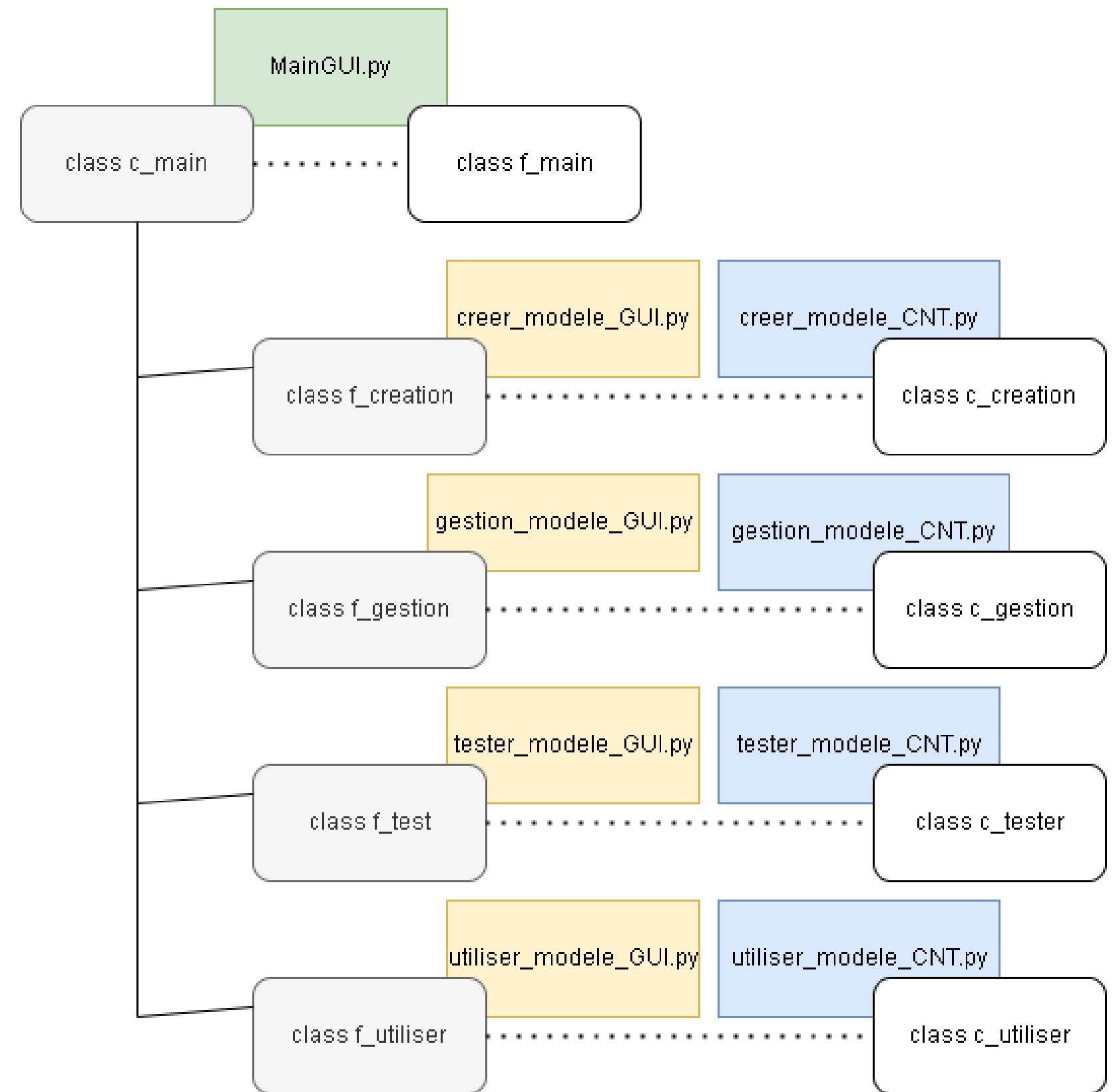
Pour implémenter cette solution, nous avons opté à utiliser le langage Python et ses fameux package 'Tkinter' pour l'interface graphique, 'TensorFlow' pour la création, l'entraînement, l'utilisation et l'exportation des modèles, 'Sqlite3' pour des manipulation sur une base de données qui stock les données sur les différents modèles et les tests qui étaient effectués sur eux; et le package 'OS' pour l'interaction avec la machine et d'autre pour des opérations élémentaires.

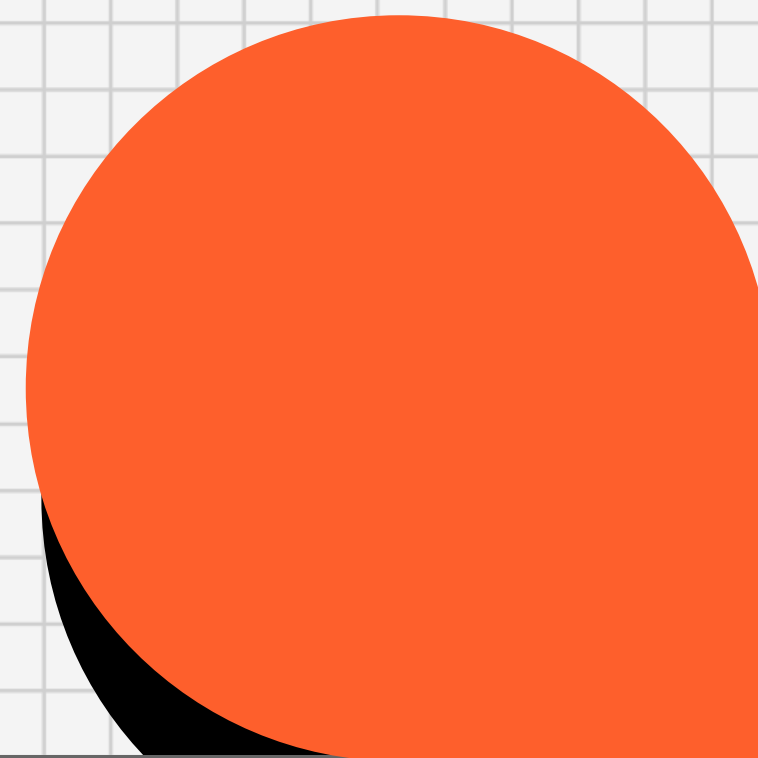
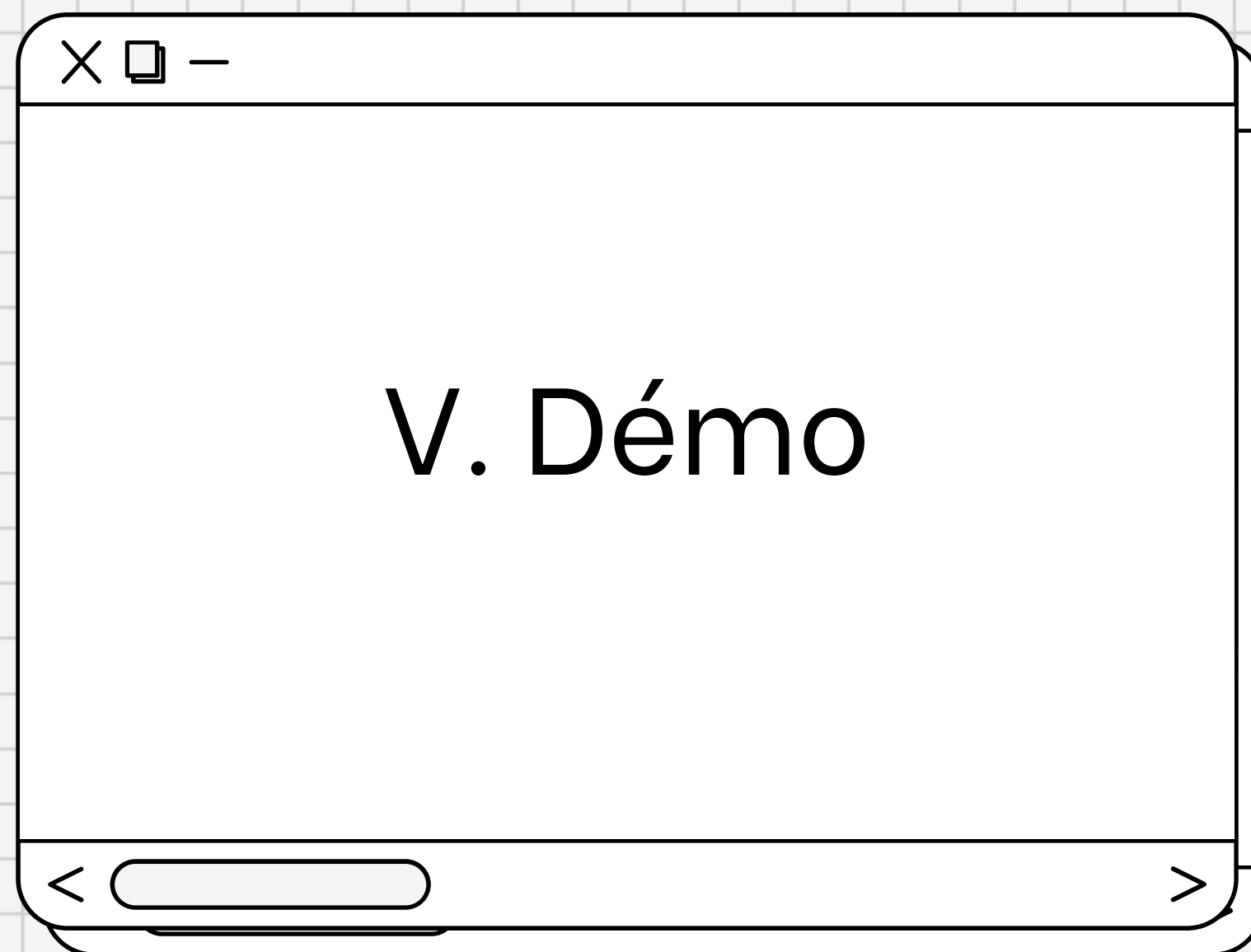
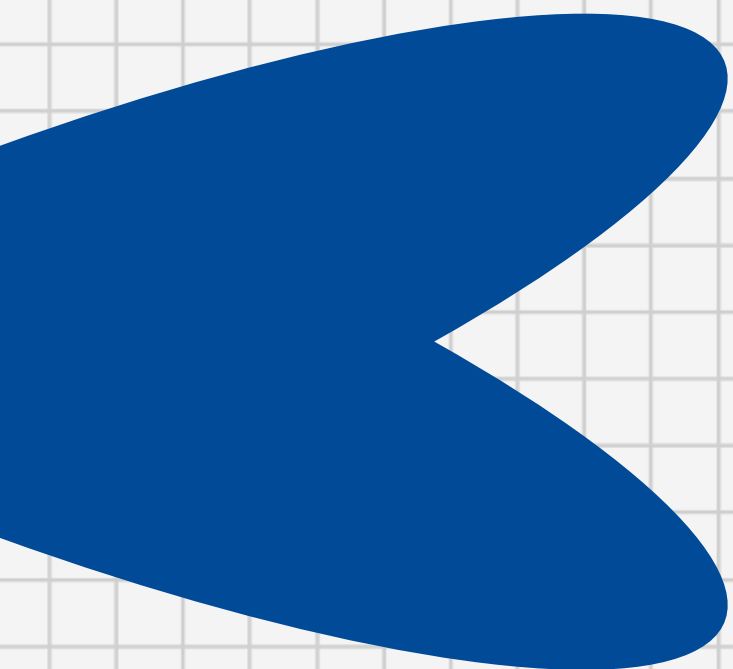


TKINTER



2. Structure en termes de
fichiers (modules) :
Le graphe ci-dessous présente
la structure du logiciel
ClassiPy V1.0.0







VI. Bibliographie

Documentation TensorFlow

https://www.tensorflow.org/api_docs

Documentation Tkinter

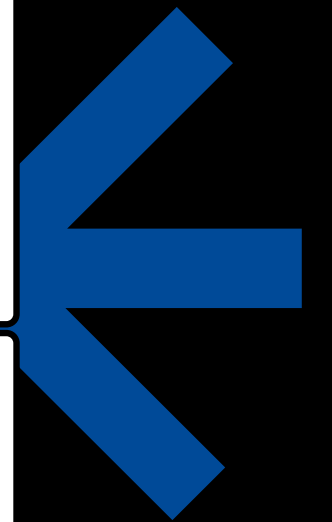
<https://docs.python.org/3/library/tk.html>

TensorFlow for Deep Learning

by Bharath Ramsundar, Reza Bosagh Zadeh - Publisher(s): O'Reilly Media, Inc.

Méthode Minimale en conception

<https://masterrssi.wordpress.com/2015/01/01/methode-minimale/>





**Merci de votre
attention !**

