

Université Gaston Berger de Saint Louis

UFR : SAT/SEG

Département : Informatique/Gestion

Filière : Méthode Informatiques Appliques a la Gestion (MIAGE)

Année : 2022/2023

Etudiant : Omar Abd Al Wahab DIASSE

Sous la direction de : Pr Jean Marie DEMBELE

**Implémentation des modeles**

Sujet : Développement de modèles de Machine Learning pour faire une analyse prédictive des finances d’une entreprise

Contents

[Introduction de chapitre 3](#_Toc168923415)

[1 Spécifications des besoins 3](#_Toc168923416)

[1.1 Contexte de l’application 4](#_Toc168923417)

[1.2 L’expression des besoins 4](#_Toc168923418)

[1.3 Les fonctionnalités de l’application 5](#_Toc168923419)

[2 Présentation des outils 7](#_Toc168923420)

[2.1 Les langages de programmation 7](#_Toc168923421)

[2.1.1 C++ 7](#_Toc168923422)

[2.1.2 Python 8](#_Toc168923423)

[2.1.3 SQL 9](#_Toc168923424)

[2.2 Les Framework 9](#_Toc168923425)

[2.2.1 Scikt-learn 10](#_Toc168923426)

[2.2.2 Spacy 10](#_Toc168923427)

[2.2.3 Gensim 11](#_Toc168923428)

[2.2.4 Tensorflow 11](#_Toc168923429)

[2.2.5 Pandas 12](#_Toc168923430)

[2.2.6 Numpy 12](#_Toc168923431)

[2.2.7 Matplotlib 13](#_Toc168923432)

[2.2.8 Qt 13](#_Toc168923433)

[3 La collecte de données 13](#_Toc168923434)

[3.1 Les données d’états financiers 14](#_Toc168923435)

[3.2 Les données texte 15](#_Toc168923436)

[4 Prédiction des valeurs 15](#_Toc168923437)

[5 Le développement du Chatbot 19](#_Toc168923438)

[6 L’interface graphique 22](#_Toc168923439)

[6.1 Développement de l’interface graphique 22](#_Toc168923440)

[6.1.1 Le frontend 22](#_Toc168923441)

[6.1.2 Le Backend 24](#_Toc168923442)

[6.1.3 Le Web server 24](#_Toc168923443)

[6.2 Présentation de l’application 24](#_Toc168923444)

[Conclusion partielle 26](#_Toc168923445)

[Webographie 27](#_Toc168923446)

# Introduction de chapitre

Nous avons décidé de nous lancer dans un trajet très ambitieux, à savoir développer des modèles intelligents qui pourront non seulement prédire des valeurs futures mais aussi nous aider dans le développement d’un chatbot que les utilisateurs vont pouvoir utiliser pour communiquer facilement avec les états financiers de leurs entreprises respectives.

Il a été constaté qu’il n’est pas pertinent de parler de la théorie seulement sans pour autant réaliser une production issue de ce travail de mémoire. De là, nous allons parler de fond en comble dans le chapitre précèdent des théories derrières les algorithmes de l’IA, qui a été plein d’enseignement. Maintenant ici, nous allons mettre en pratique tout cela en développant un logiciel complet d’analyse financière renforcé par l’IA.

Avant tout projet informatique, il y a derrière le cahier des charges qui est un document retraçant les fonctionnalités du produit. C’est pour cela que nous allons commencer par-là pour qu’un non-informaticien puisse être en mesure de comprendre la chose qui va être faite. Ceci étant dit, nous allons continuer avec la présentation des outils que nous allons utiliser pour réaliser cette application, ensuite nous allons montrer les étapes que nous avons suivi pour faire la collecte de données. Apres c’est les trois parties les plus importantes, il s’agira de la prédiction des modèles, comprendre comment nous avons fait nos prédictions, le développement du chatbot, expliquer de manière claire le processus qui a abouti à ce chatbot et enfin nous allons parler de l’interface graphique.

# Spécifications des besoins

Premièrement avant le démarrage de projet informatique, il y a le cahier des charges. C’est un document qui retrace les fonctionnalités de l’application qui doivent être implémentées. Pour cette partie, nous allons nous mettre dans la peau du maitre d’ouvrage et rédiger les spécifications de tout le travail que nous allons faire. Il n’y aura rien de complexe, juste une rédaction simplifiée des étapes auxquelles nous devons passer pour réussir à créer un modèle intelligent qui pourra prédire une analyse financière, réussir à développer un chatbot et présenter tout cela dans une interface graphique.

## Contexte de l’application

Le domaine de la finance est très vaste, il y a beaucoup de professionnels, d’ingénieurs, de chercheurs qui s’y illustrent. La finance est une science qui vise à une meilleure gestion du portefeuille des entreprises, que cela soit l’investissement, le financement…

C’est ainsi que nous introduisons l’analyse financière qui nage dans cette vaste mer du domaine de la finance. L’analyse financière ou le diagnostic financier nous permet de comprendre le processus de création de richesse, de voir si l’entreprise est dans une santé financière et j’en passe.

Aujourd’hui cette analyse financière se fait par des logiciels généralistes, ou même des fois sur feuille ce qui constitue un travail fastidieux. Malgré le fait que ces logiciels peuvent être d’une grande importance, ils souffrent souvent du trop-plein de fonctionnalités. Il y a toujours des fonctionnalités dont nous n’aurions jamais besoin si nous les utilisons pour faire une analyse financière. Autre chose, ces logiciels sont souvent très couteux et nécessitent un réinvestissement annuel, chose qui n’est pas à la portée de monsieur tout le monde.

Ceci étant posé, imaginons une application unique à vocation financière, dans cette application, il n’y aura pas de fonctionnalité qui ne sera pas relatif à la finance. Dans cette application, les utilisateurs pourront être en mesure de gérer les finances de l’entreprise, de faire une analyse financière, et aussi faire des prédictions. C’est cela le désir manifesté par les personnes qui travaillent dans la finance. En plus de cela nous en tant que futur ingénieur on IA, nous nous sommes dit qu’un petit chatbot qui permettrait de faire gagner du temps aux usagers ne serait que bénéfique pour tout le monde.

Finalement, après toutes ces observations nous avons conclu que développer ce genre d’application peut avoir un réel succès autant pour le concepteur que pour les utilisateurs. Cependant, avant de s’y aventurer il faut bien savoir les besoins fonctionnels du logiciel.

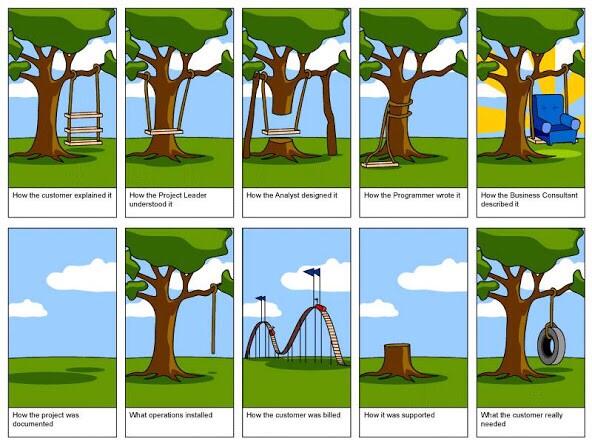
## L’expression des besoins

L’expression des besoins est une partie indéboulonnable dans la rédaction d’un cahier des charges. Dans sa forme la plus simple (ce que nous allons faire ici), l’expression des besoins se fait en listant les éléments qui entrent dans sa composition. Nous avons deux types de besoins, les fonctionnels et les non fonctionnels :

* Les besoins fonctionnels
  + Création de nouveaux fichiers
  + Sauvegarde d’un fichier
  + Ouverture d’un fichier
  + Consulter le bilan
  + Consulter le compte de résultat
  + Consulter le tableau des flux de trésorerie
  + Modifier le bilan
  + Modifier le compte de résultat
  + Modifier le tableau des flux de trésorerie
  + Prédire le bilan
  + Prédire le compte de résultat
  + Prédire le tableau des flux de trésorerie
  + Faire une analyse financière complète
  + Faire une analyse prédictive financière complète
  + Avoir un chatbot
  + Faire toutes les actions citées avec le chatbot
  + Imprimer des documents
  + Rédiger un rapport d’analyse
* Les besoins non fonctionnels
  + Un temps réduit de traitement
  + Un prix abordable
  + Facile à utiliser
  + Sécurité des données
  + Assurer la maintenance
  + Déploiement seulement dans un réseau local

## Les fonctionnalités de l’application

Pour la plupart des projets informatiques, il y a un déphasage entre les besoins du client et ce qu’a compris l’ingénieur, il peut même y avoir un décalage entre ce qu’a compris l’ingénieur et ce qui est vraiment réalisable comme l’illustre cette image.



C’est ce qui explique peut-être le taux de succès des projets informatiques mais cela, c’est un autre débat. Bref ! il va falloir toujours être en communication avec le client pour ne pas faire du travail dans le vent. Pour ce qui nous concerne bien que l’application finale qui aura pour vocation d’être vendu dans les entreprises doit répondre à tous les besoins cités, celle que nous allons développer ici qui sera une version bêta va répondre seulement aux besoins indispensables. De ce fait, les fonctionnalités que nous allons développer dans le logiciel pour ce travail de mémoire sont :

* Consulter le bilan
* Consulter le compte de résultat
* Consulter le tableau des flux de trésorerie
* Modifier le bilan
* Modifier le compte de résultat
* Modifier le tableau des flux de trésorerie
* Prédire le bilan
* Prédire le compte de résultat
* Prédire le tableau des flux de trésorerie
* Faire une analyse financière complète
* Faire une analyse prédictive financière complète
* Avoir un chatbot
* Faire toutes les actions citées avec le chatbot
* Rédiger un rapport d’analyse (limitée)

Les besoins non fonctionnels vont être plus pris au sérieux dans les versions qui vont entrer en production. Maintenant, nous savons exactement ce que nous voulons faire, il va falloir choisir les outils informatiques avec lesquelles nous allons travailler.

# Présentation des outils

Choisir un outil informatique pour implémenter un algorithme peut sembler être une chose d’évident mais nous allons voir qu’il y a des subtilités. Différents outils informatiques règles différents problèmes. Ce qui veut dire que ce n’est pas l’outil qui détermine le problème mais c’est le problème qui détermine l’outil. Le langage C est un des langages les plus rapide, mais ce n’est pas pour autant que l’on va l’utiliser pour faire du web, il y a des Framework pour cela. De ce fait comment choisis un langage ? Nous devons d’abord répondre à ces questions. Est-ce un langage utilisé pour nos problèmes posés ? Peut-il résoudre ces problèmes de manière optimiser ? Y a-t-il d’autres langages qui peuvent faire mieux le travail … Pour nos besoins cités préalablement, les langages de programmation et Framework qui répondent à ses questions sont les suivants.

## Les langages de programmation

Comme nous l’avons déjà dit dans ce document les langages de programmation sont les syntaxes qui traduisent les algorithmes d’une manière compréhensible à l’ordinateur. Apres recherche, nous avons descellé les deux meilleurs qui vont nous accompagner dans ce trajet. Que ce soit pour le développement des modèles mais aussi pour le développement une interface graphique.

### C++

Le C++ est un langage de programmation crée en 1985 par l’informaticien danois Bjarne Stroustrup pour pallier aux manquements du langage C qui n’est pas orienté objet. Le C++ est un langage de programmation très utilisé par les développeurs, notamment en ce qui concerne les applications. Il permet d'aborder le développement sous plusieurs paradigmes : programmation générique, procédurale et orientée objet. C'est un langage compilé, ce qui signifie que le code source est traduit en code objet ou binaire pour que la machine puisse l'exécuter. (C++ : présentation du langage de programmation, 2024)

Ce langage de programmation est un langage orienté objet ce qui veut dire il permet de créer des classes. Il est si populaire, on peut donner l’exemple de Google qui l’utilise pour son moteur de recherche, Microsoft qui l’utilise pour World, Excel ou PowerPoint et aussi Autodesk qui l’utilise pour Maya. Pour ce qui est de l’IA, avant l’avènement de Python, les ingénieurs l’utilisaient pour écrire les codes mais son impact est toujours présent. Car derrière presque tous les Framework de Python, qui nous aident dans l’IA, il y a le C++ ou C, le cas de Numpy, Pandas ou Matplotlib.

Les avantages de C++ :

* La performance, rapidité
* La popularité
* La portabilité dans les OS
* L’abondance de bibliothèques
* La programmation orienté objet

Les inconvénients :

* Syntaxe compliquée
* Langage pas du tout pour les débutants

### Python

Python est un langage de programmation créé par Guido Van Rossum. La première version publique du langage est sortie en 1991. Son nom provient de la troupe de comiques anglais les Monty Python.

Python est un langage de programmation dit de “très haut niveau”. Cela signifie qu’il possède un haut niveau d’abstraction par rapport au langage machine. Pour le dire très simplement : plus un langage de programmation est de “haut niveau”, plus sa syntaxe se rapproche de notre langage (l’anglais) plutôt que du langage machine. Un langage de haut niveau est donc plus facile à comprendre et à utiliser qu’un langage de plus bas niveau.

Certains langages (comme Python) utilisent un interpréteur comme traducteur tandis que d’autres utilisent un compilateur.

Un interpréteur se distingue d’un compilateur par le fait que, pour exécuter un programme, les opérations d’analyse et de traductions sont réalisées à chaque exécution du programme (par un interprète) plutôt qu’une fois pour toutes (par un compilateur). (Introduction à Python, 2024)

Les avantages de Python :

* Facile à utiliser
* Sécuriser
* Très populaire
* Compatibilité avec d’autres langages
* Possède beaucoup de bibliothèque pour le Machine Learning

Les limites de Python

* Temps d’interprétation très lent
* Mauvaise présentation des erreurs

### SQL

Structured Query Language (SQL) est un langage de gestion de données sous forme de base de données. Il est utilisé pour gérer des bases de données relationnelles avec ces quatre (4) actions principales que sont le CRUD (CREATE, RETREIVE, UPDATE ET DELETE).

De manière simple SQL va nous permettre de créer des bases de données en utilisant un système de gestion de base de données comme PostgreSQL, Oracle, Maria DB mais nous allons utiliser MySQL. Il va être créer une base de données locale pour stocker les états financiers avec lesquels nous allons travailler.

## Les Framework

Un Framework est en ensemble de fonction prédéfinie dans un langage de programmation nous permettant de faire une action bien précise. Pour une tache bien définie, si nous avons un Framework, il n’est pas nécessaire de commencer de zéro puisque certaines fonctionnalités sont déjà implémentées. Les Framework ont été créer pour les tâches complexes qui nécessite beaucoup de compétence, ainsi même les développeurs de niveau moyen peuvent créer des programmes avancés, ce qui va servir à la productivité.

Il y a différents Framework pour différents domaines informatiques (développement web, mobile, logiciel…), mais nous allons présenter les Framework qui vont nous aider dans le développement de modèle intelligent et le développement d’interface graphique, ils sont tous liés soit à Python ou à C++.

### Scikt-learn

Scikit-learn est une bibliothèque de Python qui a commencé en 2007 avec le Google Summer of Code Project par David Carpaneau.



Ce Framework s’est spécialisé dans le Machine Learning (supervisé et non supervisé) et nous donne des fonctions pratiques pour le développement de modèles. Scikit-learn supporte parfaitement des domaines comme la classification, la régression, le clustering …

Pour ce qui est de nos modèles, nous allons l’utiliser pour faire la prédiction des éléments des états financiers, la représentation de texte et bien d’autres.

### Spacy

Spacy est un Framework de python qui est utilisée dans le NLP, il nous permet tout simplement de raffiner du texte avant sa modélisation. La raison est que du texte écrit par un humain, avant de faire sa représentation en chiffre, va contenir du bruit. On entend par bruit tous les éléments, mots, vocabulaire qui ne nous sert pas. Exemple dans le langage français, il y a une conjugaison très lourde, un verbe comme « faire » peut donner fait, faite, furent etc. Spacy peut nous aider à trouver la racine de ses mots. Il peut aussi nous aider à tout mettre minuscule, supprimer les ponctuation…



### Gensim

Gensim est un Framework Python avec lequel, il est possible de faire directement de la classification de texte. Dans son fonctionnement, ce Framework utilise les modèle Word2vec et aussi Fasttext. Ces derniers sont des techniques qui sont appelé Self Supervided Learning ou l’on utilise un texte pour générer les inputs et les outputs.



Avec ce Framework on peut calculer la similarité entre deux mots, voir faire des calculs sur des mots, oui des calculs des sur des mots.

### Tensorflow

Si Scikit-Learn est une bibliothèque de Machine Learning, Tensorflow en est une spécialisée sur le Deep Learning. Cette bibliothèque a été développée par Google dans le but de permettre aux experts mais aussi de débutant d’avoir un environnement pour travailler dans le Machine Learning en général.



De tous les Framework que nous avons présentée, Tensorflow est très probablement le plus puissant car nous permettant de faire ce que tous les autres font. C’est un outil tout en un avec ses forces et ses faiblesses.

En plus de tout cela, il nous permet de faire du Computer Vision avec les CNN, du NLP avancée avec la RNN, les LSTM, de créer des API pour le déploiement et bien d’autres.

### Pandas

Pandas est un Framework Python très pratique dans le développement de modèle. Il est utilisé dans le travail a priori, le Feature Engineering. Avec pandas, nous pouvons importer des fichiers CSV, vérifier les données manquantes, les outliers…



Avant chaque développement de modèle, Pandas va certainement intervenir, ce Framework supporte les statistiques qui pourront nous permettre de mettre les données dans un format acceptable par l’ordinateur.

### Numpy

Nativement, les structure de type tableau n’existe pas en Python, il y a des listes en Python pour le remplacer. La différence entre ces deux c’est que les tableaux acceptent un seul type de donnée et taille ne varie pas or, les listes acceptent plusieurs types et sa taille peut varier.

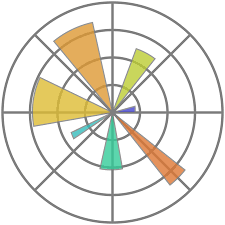


C’est là qu’intervient Numpy pour permettre d’utiliser des tableaux des Python, qui sont bien plus rapide à exécuter. En plus de cela, Numpy a un excellent support de l’algèbre linéaire, les matrices, les vecteurs et autres domaines mathématiques.

Il est nécessaire d’ajouter que Numpy, bien qu’utilisé en Python est écrit en langage C qui est plus puissant et plus rapide que le Python.

### Matplotlib

MatplotLib est un Framework de visualisation avec Python, il sert à tracer des courbes en utilisant Numpy ou Pandas. La visualisation peut intervenir avant et après le modèle, soit pour les comprendre les données brutes, soit pour vérifier les résultats.



La visualisation est en train de devenir une science à part entière, donc Matplotlib est utilisé dans des domaines autres que le Machine Learning, notamment dans le développement d’interface graphique que nous allons voir.

### Qt

Qt est une bibliothèque de C++ cross plateforme lancée en 1995, et qui est complètement gratuit. Qt nous permet de créer des interfaces très avancées et dans un IDE et simple à comprendre et à utiliser. Avec cette bibliothèque, il n’est pas nécessaire de savoir coder pour créer des interfaces graphiques car il y a la possibilité de créer des widgets avec du glisser-déposer.



Pour ce qui nous concerne nous allons bien évidemment l’utiliser pour l’interface qui va accueillir les clients. Mais un logiciel mais pas du web.

Puisque cette application n’a pas pour vocation d’être déployé dans le cloud, le web n’est pas nécessaire. De plus, développer l’interface graphique de cette manière nous donne une certaine sécurité car il n’y aura pas de brèche que des personnes extérieures à l’organisation peuvent utiliser pour accéder aux données sensibles.

# La collecte de données

S’il y a bien une chose sur laquelle nous avons insisté dans ce document, c’est que l’IA travaille sur des données, beaucoup de données : pour apprendre, il lui faut le matériel nécessaire. Les données sont pour l’IA ce que les livres, articles, les enseignements sont pour nous humains.

Cependant, après avoir compris l’importance des données, il faut comprendre qu’il n’est pas chose aisée de les recueillir ses données là. Très souvent c’est la partie la plus chronophage, et des fois demande beaucoup de ressources humaines, financières et matériels.

Pour nos modèles, nous avons la chance que les données soient déjà disponibles gratuitement. La raison est que nous allons travailler avec les données financières des entreprises, et pour les entreprises cotées en bourse c’est une obligation pour eux de les publier par souci de transparence. C’est pour cela que nous pouvons nous servir sur des données des entreprises sans risque. De l’autre part, nous aurons en outre besoin de données texte pour notre chatbot, et là, c’est encore plus facile car du texte, sous n’importe quelle forme, ce n’est pas ce qui manque sur internet. Nous allons être amener à voir comment il a été fait la collecte de nos données.

## Les données d’états financiers

Le travail que nous faisons va porter sur des entreprises cotées à la Bourse Régionale des Valeurs Mobilières (BRVM). La BRVM est le marché financier de l’Union Economique et Monétaire Ouest-Africain (UEMOA). C’est ici que l’on peut échanger des actions et obligation pour le compte des entreprises et Etats de la zone UEMOA. La BRVM regroupe les 46 entreprises et banque les plus performantes de la zone.

Eu égard de cela, tout entreprise cotée à la BRVM a le devoir chaque année de publiée ces états financiers pour le compte des actionnaires, des Etats, des investisseurs, de potentiels investisseurs ou n’importe quelles personnes physiques ou morales. Leurs états financiers sont aussi publiés dans le [site de la BRVM](https://www.brvm.org/) ainsi, il est possible de leur télécharger et de faire notre travail, on choisit une entreprise et c’est bon.

Une fois téléchargés, les états financiers d’une seule année se présentent comme suit :

* Le rapport d’activité
* Le résultat financier
* La synthèse des rapports de gestion

Mais ses données ne sont pas pour le moment exploitables, il va falloir faire un certain nombre de transformations c’est-à-dire extraire les données qui nous intéressent, les mettre sous format CSV, JSON ou autres avant de pouvoir passer au Feature Engineering. Avec des états financiers on peut faire une analyse financière, vérifier la rentabilité de l’entreprise, prédire des valeurs etc.

## Les données texte

De tous les types de données que l’on va utiliser pour des modèles intelligents, bien qu’étant les plus difficiles à exploiter, les données textes sont les plus faciles à collecter. Nous aurons besoin des données texte pour le développement du chatbot. Notre principal objectif ici c’est de comprendre les questions que l’utilisateur du chatbot peut poser. Bien sûr, le chatbot que nous allons développer va être spécialisé dans nos états financiers, donc nous aurons besoin de texte qui traite de la finance des entreprises. Il y a plusieurs manières de collecter ce genre de texte :

* Les sites de finance
* Les sites d’informations
* Les bases de données
* Les sites de génération de textes
* Le Web Scraping (des programmes qui vont aller extraire des informations automatiquement sur les sites web)

Nous avons utilisé un condensé de tout cela pour avoir les données texte dont nous aurons besoin. Il y aura d’abord les textes qui vont nous permettre de faire le Intent classification, ensuite il va utiliser des techniques pour reconnaitre les entités d’une question posée, pour pouvoir répondre avec la plus grande des précisions. Si vous n’avez pas compris cette dernière phrase, ne vous inquiétez pas car nous allons revenir en détail sur tout cela dans la partie où l’on parle du développement du chatbot.

# Prédiction des valeurs

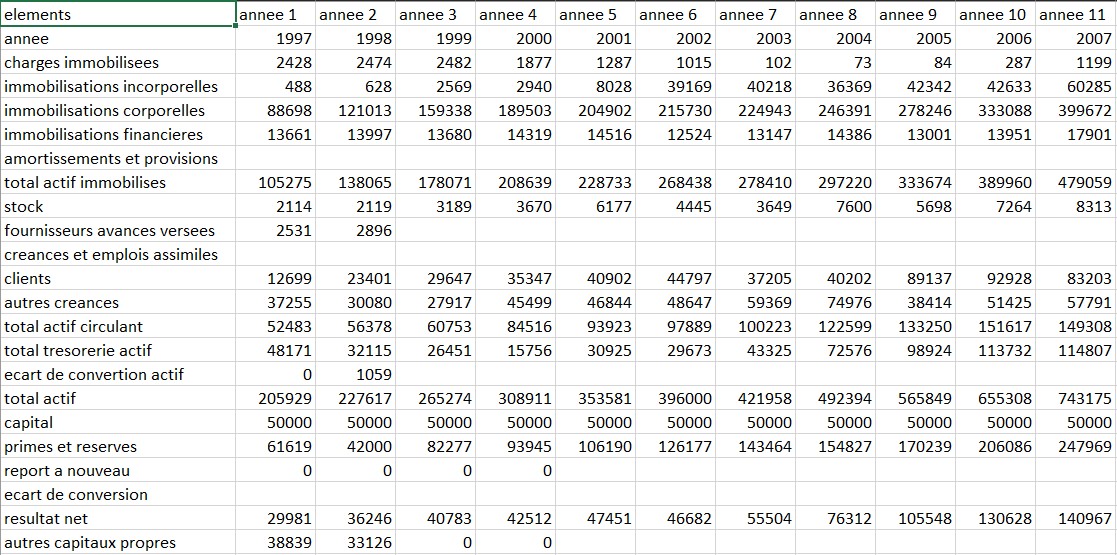
La prédiction est un vaste domaine qui ne relève pas seulement de l’IA. En effet beaucoup de domaines scientifiques essaient de faire des prédictions sur les valeurs avec lesquelles ils travaillent, c’est notamment le cas de l’économie, la météo, la bourse etc.

Les intérêts de faire des prédictions dépendent du domaine où elles sont faites, pour ce qui nous concerne qui est le domaine de la finance d’entreprise, cela nous permet d’approximer la future santé financière de l’entreprise en se basant sur les valeurs actuelles et passées.

Dans le chapitre passé, nous avons montré les techniques intelligentes qui permettent de prédire des valeurs ici nous allons voir comment cela fonctionne en pratique.

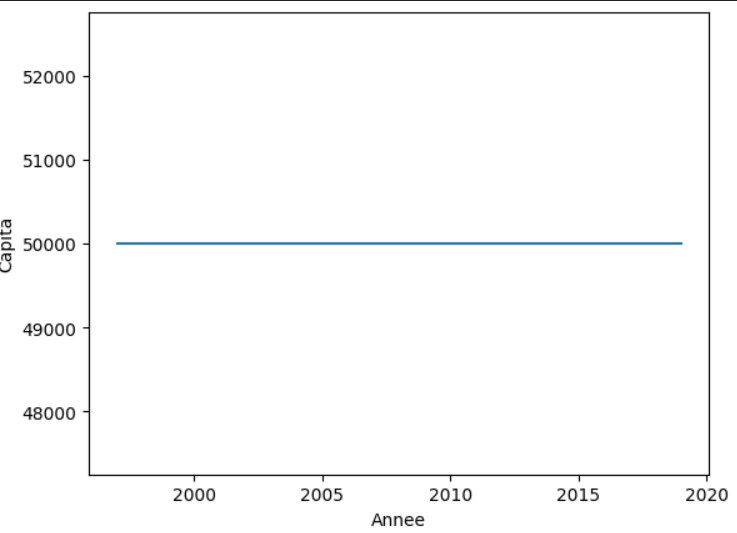
D’abord il nous faut des données, elles sont collectées, et nous avons vu comment dans la précédente partie. Nous allons faire la prédiction de chaque élément de chaque état financier. Ce qui nous fait une centaine de prédiction à faire.

Ce que nous allons faire par la suite c’est mettre toutes les valeurs dans un fichier CSV, il faut rappeler que les documents téléchargés dans le site de la BRVM, sont sous format PDF et donc non exploitables. Une fois sous le format CSV, nous aurons trois (3) fichiers à savoir les bilans, les comptes de résultats et les tableaux des flux de trésorerie.



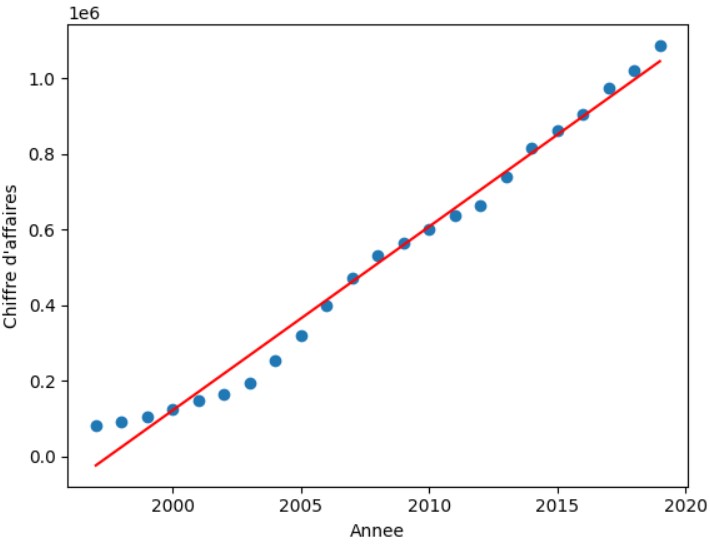
Du fait que nous avons plusieurs prédictions à faire, nous allons seulement en présenter quatre (4) du plus simple au plus intéressant.

* Le capital



Pour ce qui est du capital de cette société, nous voyons qu’il n’a pas évolué de 1997 à 2019, donc ici il n’y a pas de prédiction à faire, puisque nous savons que le capital de cette de société ne varie pas, sur toute cette période il reste à 50 000 000 000 de F CFA. Pour les valeurs futures et pour les calculs futurs qui vont faire intervenir le capital nous allons choisir cette même valeur.

* Le chiffre d’affaires

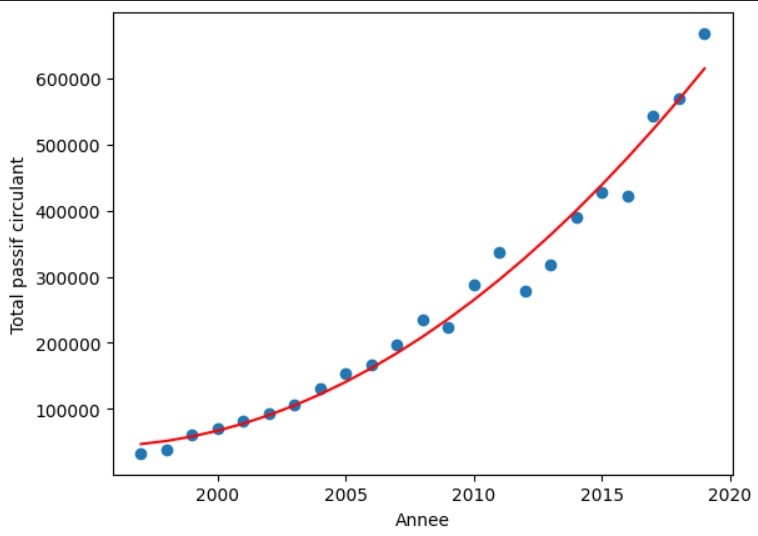


Il y a deux choses à voir ici, les points en bleu qui représentent les valeurs réelles du chiffre d’affaires en fonction des années et la droite linéaire en rouge qui est la droite de régression. La régression linéaire a bien fonctionné ici puisqu’elle épouse à la presque perfection les données. Cette droite va renvoyer le coefficient et la constante qui représentent nos w0 et w1 que nous avons déjà expliqué pour faire des prédictions pour les années à venir.

Le tableau ci-dessous nous donne pour la droite de régression du chiffre d’affaires son coefficient, sa constante et la prédiction de quelques valeurs futures.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Coefficient | Constante | 2025 | 2030 | 2035 |
| 48578.23616 | -97034585.22134 | 1336343.0148221 | 1579234.1956521 | 1822125.376482 |

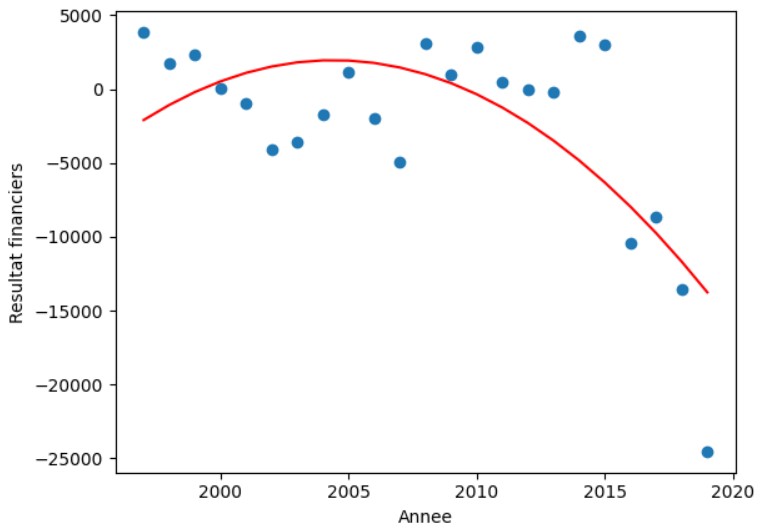
* Le passif circulant



Ce cas est un tantinet plus intéressant que le précédant les données ne suivent pas une forme linéaire, alors il nous faut autre chose. Cette autre chose c’est évidemment la régression polynomiale que nous avons aussi déjà vue. Pour ce cas de figure nous n’aurons pas seulement w0 et w1 mais aussi w2 et w3 (le biais), puisque le degré de notre polynôme est égal à 2.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| W0 | W1 | W2 | W3 | Année A |
| 0 | -4.00030285e+06 | 1.00251237e+03 | 3.99062354e+09 | W0+W1\*A + W2A2+W3 |

* Le résultat financier



Le modèle des résultats financiers est l’un des plus intéressants pour trois (3) raisons : d’abord les valeurs sont très dispersées par rapport aux autres, ensuite les valeurs décroisent de manière exponentielle et enfin il y a des valeurs négatives. C’est normal qu’il y ait des valeurs négatives pour le résultat financier dans la mesure où les charges financières sont supérieures aux produits financiers. Nous pourrions être tentés de penser que ce modèle va être plus compliquée que les autres, mais il n’en est rien. Nous allons simplement le faire passer dans une régression polynomiale et la magie va opérée. Le résultat pour ce modèle est la suivante.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| W0 | W1 | W2 | W3 | Année A |
| 0 | 2.95998776e+05 | -7.3836900e+01 | -2.9664942e+08 | W0+W1\*A + W2A2+W3 |

Il faut noter que pour tous les modèles, nous allons suivre exactement les mêmes procédures que ces quatre, tous les autres vont se classer dans l’un de ces cas de figure.

Maintenant que nous avons prédit nos modèles, il est possible de les déployer pour utiliser dans une interface graphique. Pour ce modèle, le déploiement peut être simple du fait qu’il y a seulement des nombres. On peut les stocker dans un fichier et l’appeler depuis le frontend.

# Le développement du Chatbot

L’année 1964 à 1967, fut le temps nécessaire pour développer le premier Chatbot nommé ELIZA par un scientifique de MIT. Ce Chatbot était capable de tenir une conversation en utilisant un algorithme de reconnaissance de mot, ce qui implique qu’il ne comprenait pas vraiment le texte mais rechercher des mots clés qu’il utilisait pour répondre.

Mais c’est quoi un Chatbot ? C’est un programme informatique comme les autres mais qui a des aspects bien particulier. Il sait mener une conversation qui se rapproche de celle des humains. Au début les Chatbots étaient créés pour reproduire le comportement humain chez les machines mais aussi faire avancer le NLP mais aujourd’hui les Chatbots sont là pour nous aider dans nos vies.

Comment marche les Chatbots ? Le fonctionnement des Chatbot est simple, c’est comme une conversation entre deux personnes, l’un parle l’autre écoute puis répond. Cependant dans le cas des Chatbot, très souvent c’est, je pose une question et le Chatbot réponds, mais il est possible d’avoir des banalités avec les Chatbot. Tout dépend de comment les scientifiques ont décidé de développer le Chatbot. Il faut préciser que même si le fonctionnement est simple, le développement ne l’est pas et nous allons voir pourquoi.

Mettons-nous à la place d’un Chatbot qui reçoit une question, comment ferions-nous pour répondre, oui c’est évident pour nous humains, mais pour un programme informatique c’est très compliqué. C’est là que l’on va introduire les deux (2) concepts de base de tous les Chatbot : les ***intents***et le ***entity detection.***

* *Les Intents*

Ils représentent l’objet de la question, de la phrase ou tout ce que l’utilisateur donne comme input au Chatbot, c’est la phase de compréhension de la question. Ci-après quelques phrases et leurs objets :

Comment vas-tu ? **Salutation**

Quel temps va-t-il faire demain ? **Prédiction**

Pourquoi le gouvernement ne peut-il pas imprimer plus de monnaie ? **Explication**

Nous avons trois questions et trois objets différents, c’est-à-dire que chaque question s’attend à une réponse différente. Encore une fois pour nous humains cela paraît évident mais ce n’est pas le cas pour les machines. Maintenant comment les machines font pour comprendre l’objet d’une question, il faut appeler quelqu’un que nous connaissons déjà : le Machine Learning. Tout à l’heure nous avons parlé de la collecte de données texte, maintenant nous allons les utiliser en faisant ce que l’on appelle un *intent classification.*

Nous allons faire un apprentissage supervisé, avec les données texte qui vont être étiquetées de leur objet. Les trois phases avec leur objet que nous avons donné représentent aussi des exemples d’à quoi va ressembler le *dataset.* Ce qui suit c’est la représentation de texte, transformer le texte en un format compréhensible par la machine. Apres avoir fait cela, nous le faisons passer dans un algorithme de Machine Learning, si nous avons une bonne *accuracy*, donc le modèle est bon, notre programme est capable maintenant de comprendre le texte. L’*intent classification* est un moyen très puissantqui nous permet de comprendre l’objet d’une question posée par un utilisateur.

* Le *entity detection*

Maintenant que l’on comprend le sens des questions il faut bien répondre, mais pour répondre il nous faut des arguments, c’est qu’y a-t-il dans la question qui peut nous permettre de répondre, les deux cas de figure se présentent. Il y a des questions où nous pouvons répondre directement sans recherche d’arguments comme quand l’utilisateur salue ou remercie le Chatbot. S’il dit « Salut » ou « merci beaucoup », si le Chatbot parvient à bien détecter que l’un a pour objet une salutation et l’autre un remerciement, alors nous pouvons renvoyer la réponse associée. Cependant si l’utilisateur pose la question suivante : « Quel sera notre chiffre d’affaires de 2026 », et que le Chatbot a bien compris que l’objet de cette question est une prédiction, c’est là, que va intervenir l’*entity detection.*

« Quel sera notre chiffre d’affaires de 2026 », quelle sont les éléments de cette question que nous allons utiliser pour y répondre. D’abord il y a le chiffre d’affaires, c’est l’élément que l’on veut prédire et aussi l’année 2026, c’est la période sur lequel on veut faire une prédiction. C’est cela l’*entity detection*, maintenant une fois que nous avons relevé chiffre d’affaires et 2026 de la question, on appelle la fonction qui permet de faire les prédictions en lui passant les arguments, et après on envoie la réponse à l’utilisateur.

Il peut y arriver que l’utilisateur pose une question ou il n’y a pas tous les arguments mais le modèle parvient quand même à comprendre l’objet, exemple : « quelle est la prédiction de l’actif total », c’est une prédiction mais in n’y a pas de période. Dans ce cas, le Chatbot va lui envoyer un message pour préciser la période ou l’élément sur lequel va être faite la prédiction si c’est ce dernier qui manque.

Parlons peu des réponses, ce n’est vraiment pas la partie la plus compliquée dans le processus de développement. Une fois que l’on a passé avec brio les étapes d’***intents***et le ***entity detection*** et que nous avons aussi appelé les fonctions appropriées, on renvoie des réponses fixes préalablement définies. Maintenant il est possible d’avoir plusieurs réponses et renvoyer une au hasard pour avoir cet aspect moins robotique. Les réponses fixes sont appropriées pour notre programme sinon il faut utiliser des algorithmes générateurs de textes qui va rendre la chose beaucoup plus compliquée.

Nous pouvons conclure cette partie sur le Chatbot en disant pour son développement la question est plus importante que la réponse, si nous parvenons à comprendre parfaitement la question, nous parviendrons à donner la bonne réponse.

# L’interface graphique

A ce moment de notre travail, tout a déjà été fait, ou presque. Oui les prédictions ont été faites, oui le Chatbot a été développé mais il subsiste un grand problème : à moins être un ingénieur en IA, vous ne pouvez pas interagir ou utiliser les modèles. C’est ce problème que nous allons essayer de résoudre ici en développant une interface graphique que n’importe quel utilisateur peut utiliser pour interagir avec les modèles. Pour ce faire il y aura deux phases : faire de Graphical User Interface (GUI) et lier les modèles avec le GUI à travers un web service.

## Développement de l’interface graphique

L’interface graphique va être développée avec la bibliothèque Qt/C++, qui est un Framework qui nous facilite cette tache en utilisant le langage C++. Nous avons utilisé cette bibliothèque car l’application n’a pas pour vocation d’être déployé sur internet, c’est une organisation qui va utiliser une instance de celle-ci.

### Le frontend

Qt nous donne un certain nombre de fonctionnalités pour réaliser des logiciels robustes et très avancés mais nous allons en prendre quelques-uns, qui nous ont été utiles dans notre travail, pour les expliquer et faire comprendre comment nous avons utilisé dans nos modèles.

* Widget

Les Widgets font partie des éléments de base sur Qt, au fait tout est un widget et il y a des Widgets pour tout faire. Pour mieux comprendre ce concept, un Widget est un élément visible que l’utilisateur peut interagir avec. Nous pouvons donner l’exemple d’un bouton, une zone de texte, un label ou voir même une page entière.

L’un des Widget les plus importante c’est ce que Qt appelle QMainWidget qui est le Widget principale sur lequel tous des autres vont s’accoler. Il faut préciser chaque Widget a un parent sauf le Widget principale. Ce qui nous donne une structure sous forme d’arbre.

Dans notre logiciel nous allons utiliser toute sorte de Widget commençant bien évidemment par le QMainWidget en passant par les autres qui permettent de présenter le résultat de nos modèles d’une manière simple.

* Layout

Le deuxième élément sur lequel nous allons un peu nous attarder ce sont les Layout. Ils représentent tous les manières de disposer nos Widgets. Par défaut, un Widget va empiler tous ses enfants sur son point (0, 0) en haut en gauche dans un ordinateur.

Avec l’aide des Layouts, nous pouvons avoir plusieurs manières de présenter les Widgets, ceci aura pour effet d’avoir un logiciel plus sérieux. Il y a trois sortes de Layout sur Qt : QVBoxLayout (disposition verticale), QHBoxLayout (disposition horizontale) et QGridLayout (disposition en grille). Ses trois héritent tous de la classe mère QLayout.

Bien que nous ayons plus utilisé les dispositions verticales et horizontales dans notre logiciel, la disposition en grille peut être très pertinente par moments. C’est impressionnant de se dire quand même que tous les frontends que nous voyons dans les logiciels ne sortent pas de ces trois Layout.

* Signal and Slot

Les Signal and Slot sont les deux concepts de la programmation événementielle que Qt met à notre disposition. Un Signal c’est un signal c’est-à-dire un déclencheur, cela peut être une clique, un focus, un survol un changement quelconque. Il y a des Signal prédéfinis comme ceux que nous venons de citer, mais nous avons la possibilité d’en créer des nouveaux. Néanmoins, les prédéfinis vont nous suffire dans la plupart des cas.

Les Slot ce sont les fonctions que nous allons utiliser pour réagir par rapport au Signal. Mais pourquoi ce ne sont pas des méthodes comme les autres ? C’est que les Signal n’acceptent que les Slot comme fonction, une méthode qui ne portent pas la mention Slot ne va pas être acceptée dans la programmation évènementielle de Qt.

Nous allons illustrer cela en donnant l’exemple de l’utilisateur qui veut ouvrir les paramètres. S’il clique sur le bouton paramètre (Signal), l’application va ouvrir la boite de dialogue paramètre (Slot). Ce qui intéressant avec Qt, c’est qu’il nous permet de faire cela en une seule ligne de code avec la fonction « connect » de la classe QObject.

### Le Backend

Le backend d’une application c’est tout ce que l’utilisateur ne voit pas, et qui fait toute la puissance d’une application. Tout ce qui est base de données, classes, appelle de fonctions sont classer dans le backend. Pour ce qu’il s’agit de nos applications nous aurons une base de données MySQL, des classes qui vont interagir avec la base de données.

* La base de données

Nous allons avoir une base de données locale qui va nous servir à stocker les valeurs des états financiers. Il y aura trois tables dans la base de données à savoir le bilan, le compte de résultat et aussi le tableau des flux de trésorerie et chaque enregistrement de chaque table va représenter une année. C’est une base de données simple, nous allons rarement y insérer des données ou même modifier. Dans la plupart des cas il sera seulement fait des requêtes SELECT pour obtenir des données.

* Le contrôleur

Même si nous ne sommes pas vraiment dans un modèle MVC (Model, View, Controller), car nous ne pouvons nous permettre de laisser les requêtes directement accéder à notre base de données, cela nous permet aussi une bonne gestion des données. Ce modèle va se présenter sous forme de classe, ici aussi nous aurons trois (3) classes pour nos trois états financiers. Pour chaque classe, nous aurons les méthodes qui renvoient les états financiers, les analyses verticales, les getters et les setters pour chaque élément des états financiers.

* Serveur

Avant de pouvoir utiliser une base de données, il nous faut un SGBD (système de gestion de base de données) qui va nous permettre d’interagir avec la base, nous l’avons déjà dit c’est MySQL que nous allons utiliser avec XAMPP, qui est un serveur qui nous permet la connexion avec une base de données MySQL, Apache etc.

### Le Web server

Un web server est un programme informatique qui nous permet une encapsulation de données avec les protocoles http ou https. Avec un web server on peut déployer une fonction qui peut être invoquée depuis plusieurs Endpoint.

De là, on peut voir comment cela peut nous être utile, nous avons déjà développé nos modèles intelligents et ce qui suit c’est de les lier avec l’interface graphique. Les modèles ont été développés en langage Python et l’interface graphique avec C++, il nous faut les lier. C’est là qu’intervient la puissance des web server. Il est possible d’écrire un Application Programmation Interface (API) dans un langage A et de l’invoque dans un langage B avec l’aide d’un navigateur.

Il y a deux types de web server : étendu avec SOAP et REST, nous allons utiliser REST avec la bibliothèque de Python FastAPI.

Nous pouvons donner l’exemple d’un cas d’utilisateur du Chatbot. L’utilisateur pose sa question depuis l’application Qt, une requête REST va être invoquée avec la question comme paramètre, le web server va le prendre le traiter et renvoyer la réponse sous format JSON ou XML.

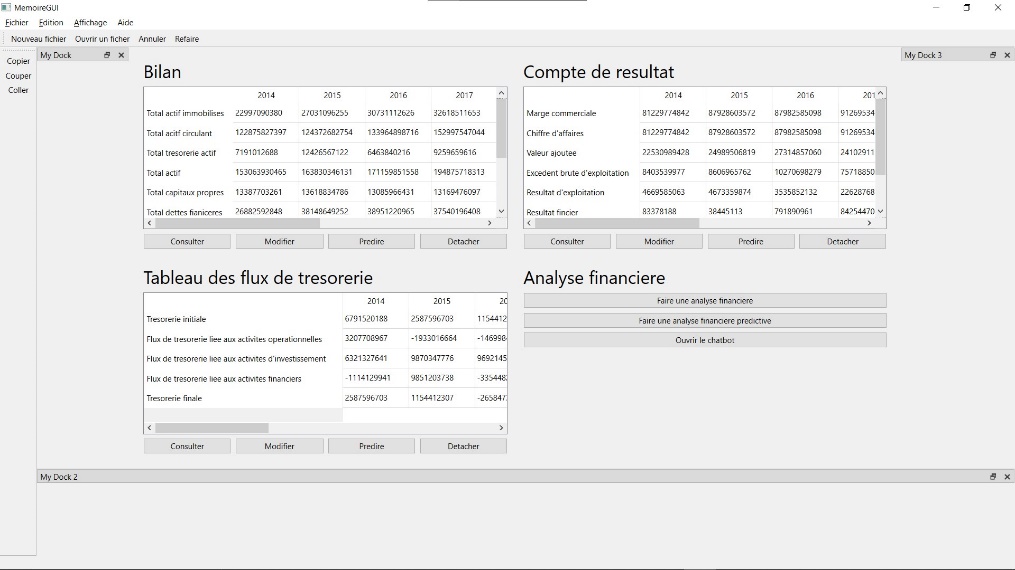
Cette méthode est aussi appelée le développement multi-tiers il y a deux programmes différents (deux tiers) mais ils peuvent communiquer tout en étant indépendant l’un de l’autre. Ci-après une image qui illustre l’architecture du processus global.

(Image illustrant l’architecture)

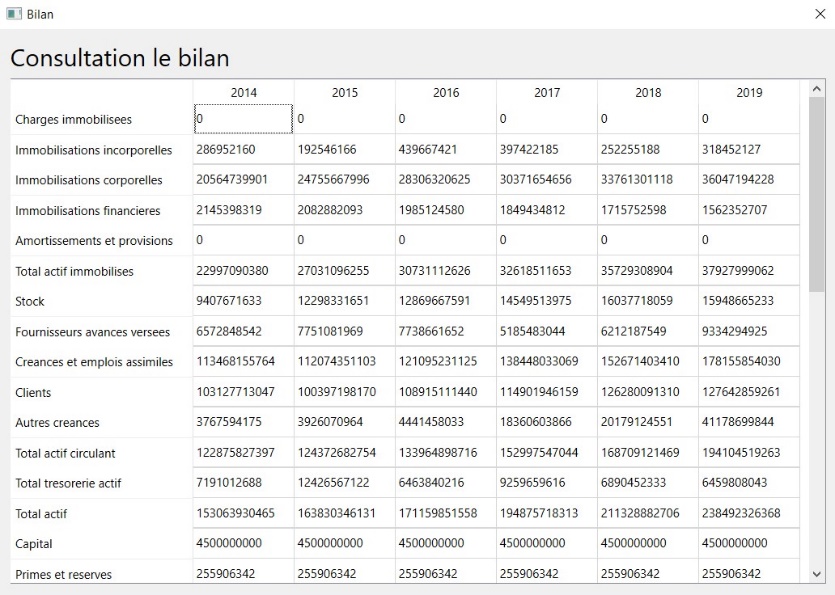
## Présentation de l’application

Toute l’explication a été faites donc ici nous allons juste montrer à quoi ressemble l’application, l’interface graphique de l’utilisateur. Toutes les pages ne seront pas montrées ici mais seulement les plus pertinentes, celle que l’utilisateur va ouvrir très souvent.

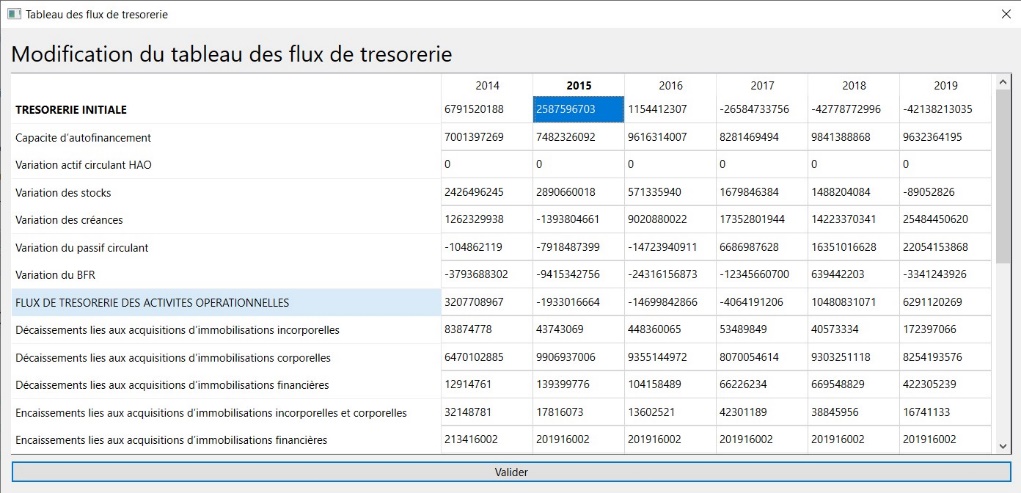
* La page d’accueil



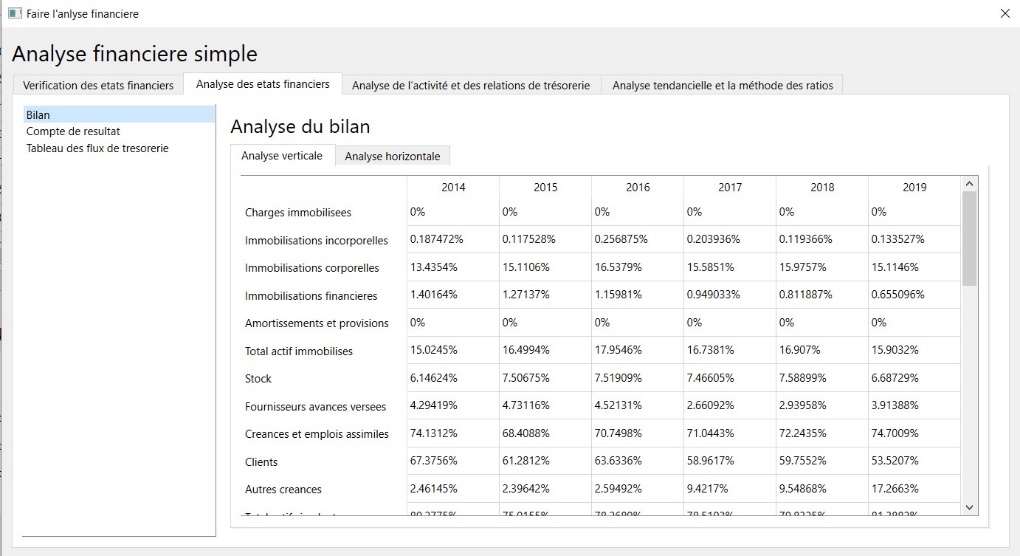
* La page de consultation du bilan



* La page de prédiction du compte de résultat sur les 3 prochaines années
* La page de modification du tableau des flux de trésorerie



* La partie analyse des états financiers



* La partie analyse prédictive de la rentabilité sur les 3 prochaines années
* Le chatbot

# Conclusion partielle

Nous laissons à l’appréciation du lecteur de juger si le travail de l’implémentation a été bien fait ou pas. Ce qui est sûr c’est nous avons utilisé toutes les armes que nous avons en main pour mener à bien cette partie. Bien que n’étant pas la partie qui nous a posé le plus de problèmes mais c’est définitivement d’une des plus intéressantes, car nous savions ce que nous faisions et là il n’y a rien de mieux que de coder.

Dans cette partie nous avons développé des modèles intelligent qui nous ont permis de prédire les états financiers avec un bon degré de confiance mais surtout le chatbot qui était le grand challenge de ce travail de mémoire. Il est intéressant de préciser qu’il n’est pas obligatoire pour un data scientist de développer des GUI, car très souvent le travail de ce dernier s’arrête au développement des modèles mais aussi le web server. Néanmoins, c’est toujours intéressant de maitriser ces genres de chose.

Ce travail de mémoire a été nécessaire car il nous a appris tellement de choses, nous nous sommes confrontés à la réalité du métier chose qui nous a forgés. Nous avons aussi découvert le trajet complet que les scientifiques prennent dans leur travail, à savoir de la collecte de données jusqu’au premier test de validité.

Ce chapitre est le terminus ce travail de mémoire …

# Webographie

*C++ : présentation du langage de programmation*. (2024, 5 20). Retrieved from Journal du net: https://www.journaldunet.fr/web-tech/dictionnaire-du-webmastering/1445316-c-definition-et-presentation-de-ce-langage-de-programmation/

*Introduction à Python*. (2024, 5 20). Récupéré sur pierre-giraud: https://www.pierre-giraud.com/python-apprendre-programmer-cours/introduction/