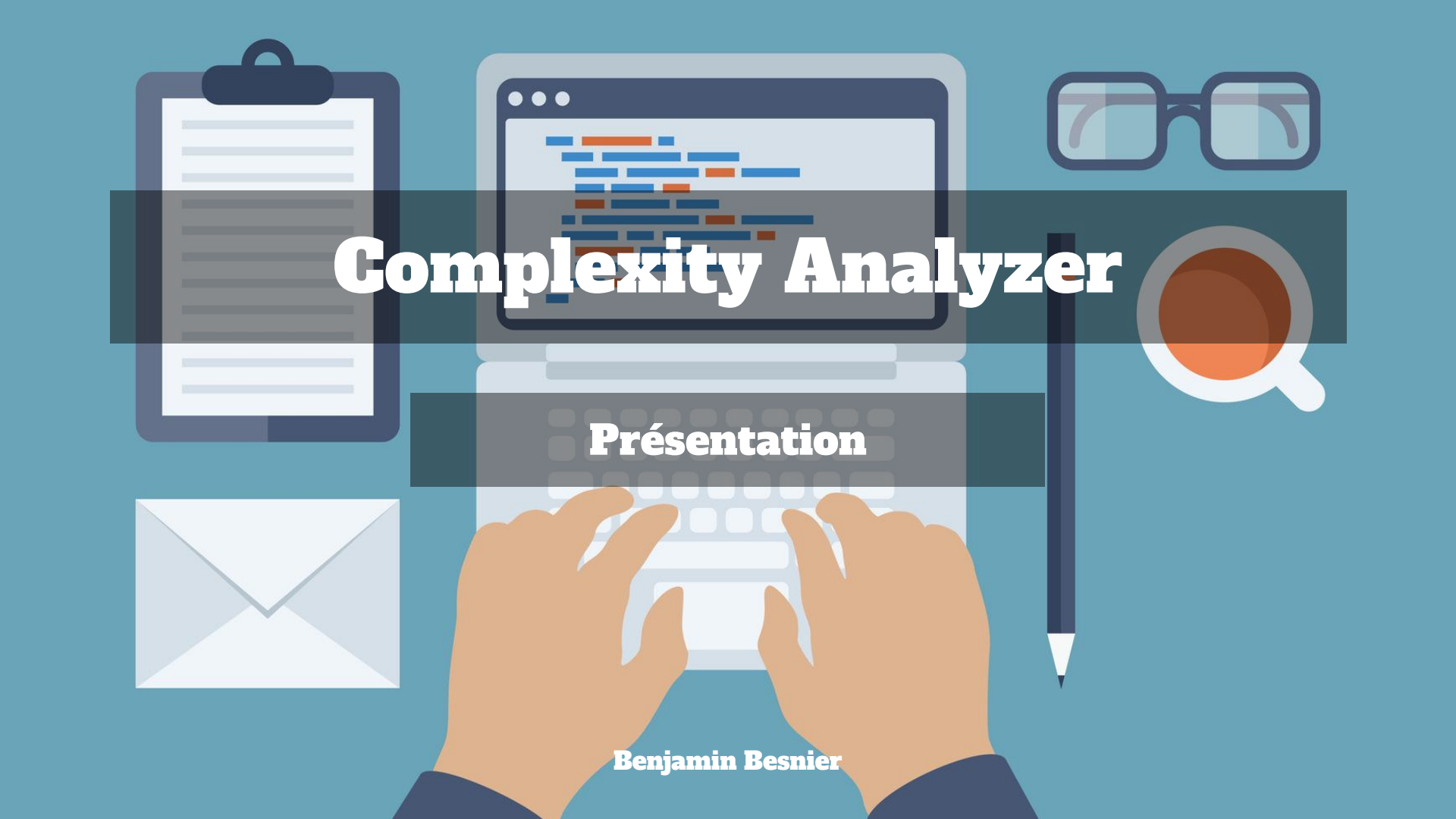


Complexity Analyzer

Présentation

Benjamin Besnier

Présentation 1

Déterminer la complexité temporel d’un algorithme en disposant de son code source

Par régression linéaire

Dans un temps convenable

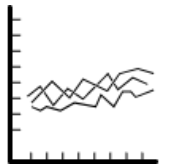
Fonctionnement 2

Plusieurs exécutions de l'algorithme pour des tailles d'entrées différentes

- ➜ Collecte du temps d'exécution pour chacune des entrées
- ➜ Construction du tableau permettant la régression

Réalisation d'une première régression linéaire

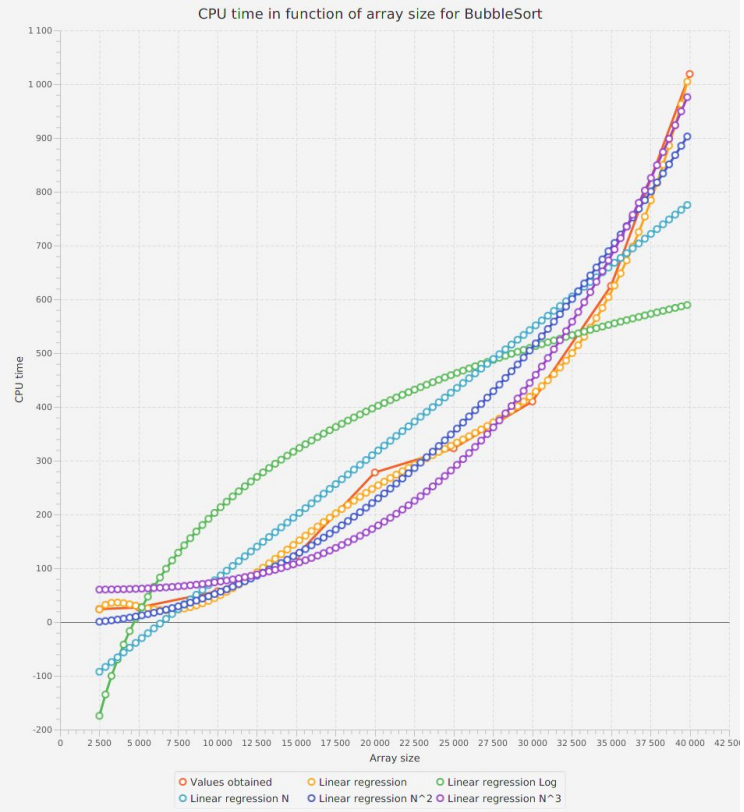
- ➜ Méthode des moindres carrés
- ➜ Si impossibilité de conclure, application de la seconde approche



Réalisation d'une régression par classe de complexité

- ➜ Calcul de l'erreur (la somme des carrés des écarts aux points connus)
- ➜ Détermination de la classe de complexité

Résultats



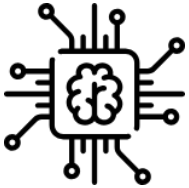
Résultats obtenus pour le tri à bulle

Régression	O(Log)	O(N)	**O(N^2)**	O(N^3)
Erreur	29280.9	9225.85	**146.96**	1772.99
Indicateur	19824%	6177%	**0%**	1106%

Poursuite 4

Étude de codes étudiants posté sur caséine

➔ En fonction des codes soumis, déterminer la complexité de nouvelles soumissions



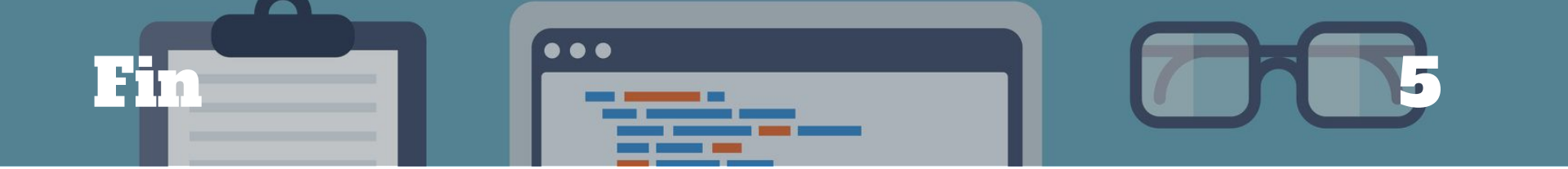
Approche par machine learning

➔ Permettre de distinguer les classes en log(n)

Intégrer le code sur caséine

➔ Déterminer la complexité des codes soumis par les étudiants, ce qui pourrait amener un nouveau critère d’ évaluation

Fin



Merci de votre attention, si vous avez des questions n'hésitez pas