

Extraction de métriques et utilisation d'algorithmes prédictifs pour la gestion de projet

IA pour le GL - 8 Novembre 2021

Quentin Perez, Christelle Urtado et Sylvain Vauttier

EuroMov Digital Health in Motion, Univ Montpellier, IMT Mines Ales, Ales, France

Génie Logiciel Empirique ? C'est quoi... Ça se mange ?

Analyse des pratiques logicielles par observations et expérimentations dans le but d'en améliorer la qualité via l'adoption de nouvelles pratiques.



Génie Logiciel Empirique ? C'est quoi... Ça se mange ?

Analyse des pratiques logicielles par observations et expérimentations dans le but d'en améliorer la qualité via l'adoption de nouvelles pratiques.



- Pléthore de pratiques de développement

Génie Logiciel Empirique ? C'est quoi... Ça se mange ?

Analyse des pratiques logicielles par observations et expérimentations dans le but d'en améliorer la qualité via l'adoption de nouvelles pratiques.



- Pléthore de pratiques de développement
- Beaucoup de littérature autour du développement logiciel (parfois contradictoire)

Génie Logiciel Empirique ? C'est quoi... Ça se mange ?

Analyse des pratiques logicielles par observations et expérimentations dans le but d'en améliorer la qualité via l'adoption de nouvelles pratiques.



- Pléthore de pratiques de développement
- Beaucoup de littérature autour du développement logiciel (parfois contradictoire)
- Écart en les bonnes pratiques et la “vrai vie”
- Trouver des nouvelles méthodes de mesures de la qualité

Utilité de l'IA dans le GL empirique

- Regrouper ou associer des données (apprentissage non-supervisé). Regrouper des sous-ensembles de projets par qualité

Exemple

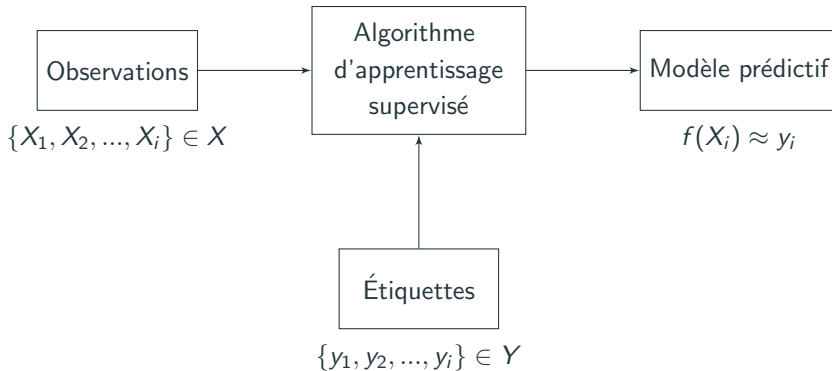
Regrouper des sous-ensembles de projets par qualité ou type de projet.

- **Classifier ou prédire des données** (apprentissage supervisé).

Exemple

Exemple catégoriser automatiquement un ticket logiciel ou déterminer le nombre de développeurs expérimentés nécessaires en fonction de la taille du projet.

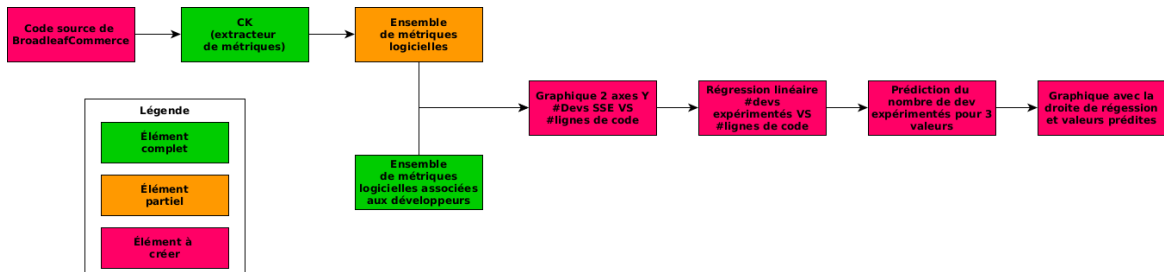
Apprentissage supervisé



Objectifs du TP

1. Extraire des métriques logicielles sur plusieurs versions d'un projet et en utiliser une spécifiquement (nombre de lignes de code)
2. Utiliser un classifieur (Random Forest) déjà entraîné permettant de catégoriser les développeurs avec 23 métriques sur plusieurs versions d'un projet
 - Développeurs expérimentés (*Senior Software Engineers*)
 - Développeurs non-expérimentés (*Non-Senior Software Engineers*)
3. Tracer différents graphiques
4. Créer un prédicteur du nombre de développeurs expérimentés en fonction de la taille des projets.

Objectifs du TP



- Langage : Python
- Extraction de données Git : GitPython
- Apprentissage machine : Scikit-Learn
- Graphiques : Matplotlib
- Extraction de métriques : logiciel Java (jar exécuté via Python)
- IDE : Visual Studio Code

Structuration du TP

TP sur GitHub : <https://github.com/qperez/TP-Master-MTP-GL-IA4GL>

Cas d'étude du TP : BroadleafCommerce

▼ COURS-IAIL-CLASSIF-DEV

> ck_metrics

> metrics_by_dev

> venv

≡ classifier_rf.pkl

📄 dataset_dev_anonymized.csv

<> developers_classification.html

📄 developers_classification.ipynb

🖼️ plot_sse_loc_prediction.png

🖼️ plot_sse_vs_loc_by_version.png

≡ requirements.txt

- ck_metrics : dossier contenant les métriques de BroadleafCommerce par version majeure et mineure
- metrics_by_dev : dossier contenant 23 métriques associées aux développeurs de BroadleafCommerce pour chaque version majeure et mineure
- venv : environnement virtuel Python
- classifier_rf.pkl : classifieur Random Forest à utiliser pour catégoriser les dévs
- dataset_dev_anonymized.csv : fichier CSV contenant les développeurs utilisé pour entraîner le classifieur
- developers_classification.ipynb : *Notebook* Jupyter