Conception d'un modèles prédictifs pour prédire la présence d'or.

Rapport du projet d'intelligence artificielle

Auteurs :
M. Khaled Hajjar

Encadrants :
M. LaurentBougrain

Table des matières

Introduction

Les arbres de décision sont des méthodes d'apprentissage non paramétriques utilisées pour des problèmes de classification. L'objectif est de créer un modèle qui prédit les valeurs de la variable cible, en se basant sur un ensemble de séquences de règles de décision déduites à partir des données d'apprentissage. L'arbre approxime donc la cible par une succession de règles if-then-else. Plus l'arbre généré est complexe, mieux le modèle « explique » les données d'apprentissage mais plus le risque de sur-apprentissage (over-fitting) est élevé. Les arbres de décision ont plusieurs avantages qui les rendent intéressants dans des contextesoù il est utile de comprendre la séquence de décisions prise par le modèle :

— Ils sont simples à comprendre et à visualiser.
— Ils nécessitent peu de préparation des données (normalisation, etc.).
— Le coût d'utilisation des arbres est logarithmique.
— Ils sont capables d'utiliser des données catégorielles et numériques.
— Ils sont capables de traiter des problèmes multi-classe.

Pour ce projet j'ai essayé deux methodes afin de classifer les données. En effet, j'ai tout d'abord tenté une approche par réseaux de neuronnes à l'aide de keras, l'API de TenserFlow. Malheureusement les resultats n'etait pas concluants et le projet prenait beaucoup trop de temps (6 h de travail pour arriver à des resultats). J'ai donc décidé d'utiliser le Decision Tree de Scikit-Learn.

Chapitre 1

Implémentation du projet

Pour ce projet j'ai utilisé les librairies pandas et scikit-learn afin de pouvoir analyser les fichiers txt fournis, au préalable transformés en CSV et classifer les fichier cibles.
Dans un premier temps, l'objectif était donc de récuperer les CSV afin de pouvoir les manipuler correctement. La fonction read_csv de pandas a permis de charger correctement les données.

```
[5 rows x 23 columns]
     X_COORD    Y_COORD  BOUGUER  ...          AGE         ROCK        OR
0  5666700.0 -3632981.5    -51.0  ...    PALEOZOIC  METAMORPHIC   STERILE
1  5244869.5 -3261162.5   -340.0  ...  PROTEROZOIC     PLUTONIC   STERILE
2  5666491.0 -3612491.2    -47.0  ...    PALEOZOIC  METAMORPHIC   STERILE
3  5528703.5 -2565500.0   -412.0  ...  PROTEROZOIC     PLUTONIC   STERILE
4  5528679.0 -3031326.0   -235.0  ...    PALEOZOIC     VOLCANIC  GISEMENT
```

FIGURE 1.1 – Capture d'écran des 5 premières lignes d'un fichier du projet.

Dans un second temps, il faut traiter les données. En effet, l'arbre de décision ne peux pas travailler à partir de String, il faut donc trouver un moyen de "traduire" chaque String en entier. La fonction encode_target effectue ce travail dans mon projet, elle effectue une boucle for sur les String de la colonne sélectionnée et remplace à chaque fois ce String par un entier. Nous obtenons un résultat comme ceci :

```
[5 rows x 23 columns]
     X_COORD    Y_COORD  BOUGUER  GRTOPISO  ...     DIST_90  ROCK2  AGE2  OR
0  5666700.0 -3632981.5    -51.0    0.0002  ...  480480.880      0     0   0
1  5244869.5 -3261162.5   -340.0   -0.0008  ...  105266.016      1     1   0
2  5666491.0 -3612491.2    -47.0    0.0004  ...  500917.160      0     0   0
3  5528703.5 -2565500.0   -412.0   -0.0005  ...   85373.600      1     1   0
4  5528679.0 -3031326.0   -235.0    0.0000  ...   89502.984      2     0   1
```

FIGURE 1.2 – Capture d'écran des 5 premières lignes d'un fichier transformé.

D'autre part j'ai décidé, avant d'entrainer le programme sur gisementLearn.CSV complet, de l'entrainer sur 1500 lignes et tester le resultat sur les 133 dernières. Cette phase me permet de tester la précision de mon programme. J'ai donc pu comparer le nombre minimal (min_samples_split) d'observations requises pour rechercher une dichotomie par rapport à la précision de la classification. Cela servira dans la prédiction future et renforcera la véracité de mes résultats. Ici la precision maximale vaut 0.88 et est atteinte en 16.

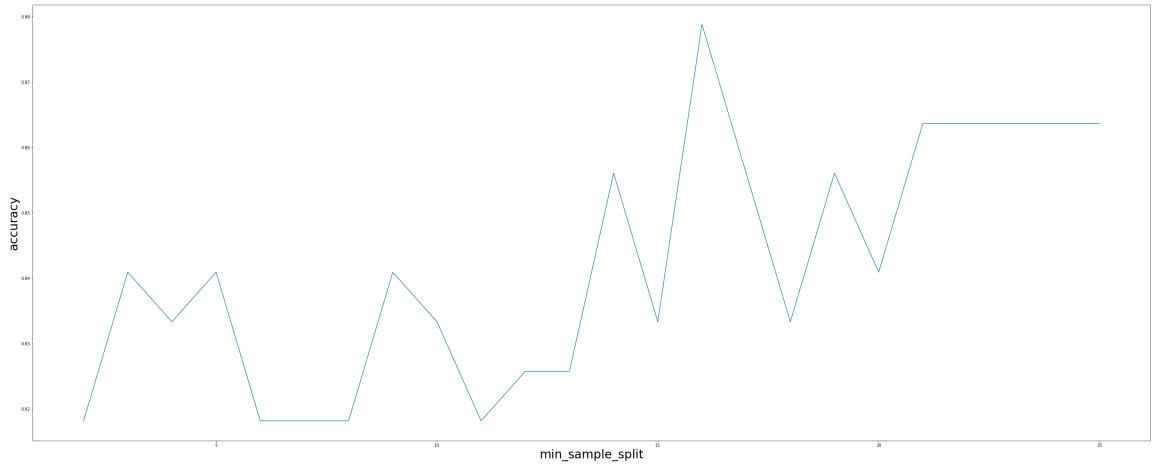

FIGURE 1.3 – Tableau représentant min_samples_split en fonction de l'accuracy.

Une fois ces étapes terminées j'ai finalement implémenté l'arbe de décision qui servira à la future classification.Cette implémentation à été possible grâce a la fonction DecisionTreeClassifier avec en argument le résultat trouvé précédemment, à savoir min_samples_split=0.88.

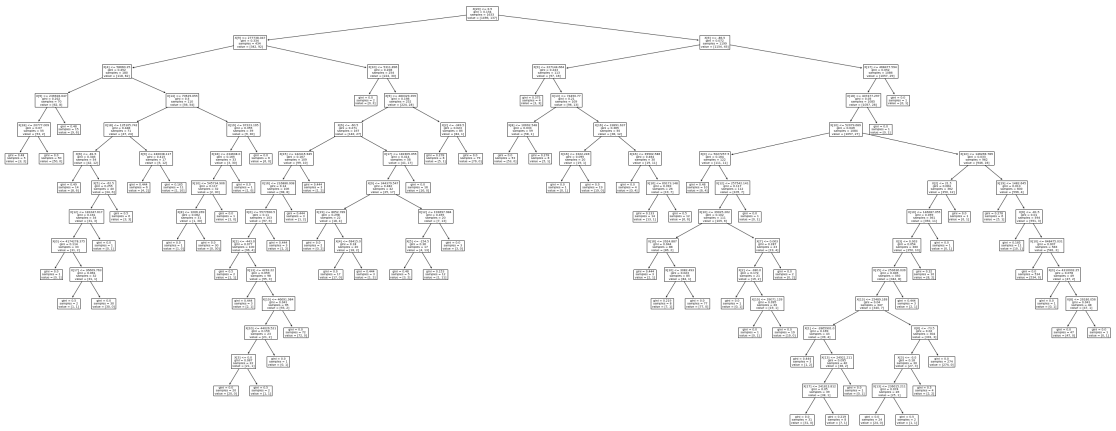

FIGURE 1.4 – Arbre de décision obtenu.

Finalement, j'ai utilisé la fonction predict appliquée au fichier gisementTestNolabel.CSV auquel j'ai également transformé les données en entiers. Le résultat de cette fonction est

alors une colonne de 0 et de 1. Il faut ensuite retransformer en String, ce qui est fait assez facilement avec un dictionnaire et la fonction map.
Nous avons enfin un fichier final appelé gisementTest.txt qui contient les résultats théoriques obtenus grâce à l'arbre de décision.

```
[5 rows x 23 columns]
     X_COORD    Y_COORD  BOUGUER  ...          AGE         ROCK        OR
0  5666700.0 -3632981.5    -51.0  ...    PALEOZOIC  METAMORPHIC   STERILE
1  5244869.5 -3261162.5   -340.0  ...  PROTEROZOIC     PLUTONIC   STERILE
2  5666491.0 -3612491.2    -47.0  ...    PALEOZOIC  METAMORPHIC   STERILE
3  5528703.5 -2565500.0   -412.0  ...  PROTEROZOIC     PLUTONIC   STERILE
4  5528679.0 -3031326.0   -235.0  ...    PALEOZOIC     VOLCANIC  GISEMENT
```

FIGURE 1.5 – Capture d'écran des 5 premières lignes du fichier final.

Conclusion

Pour conclure sur ce projet, je peux dire que ça à été un projet extrêmement intéressant qui change des autres projets grâce au machine-learning. Il est également à noter que ce projet est tout à fait réalisable et je remercie par la les encadrants pour ne pas avoir donné un projet trop volumineux qui pour un élève membre de l'approfondissement LE signifie simplement un long projet de plus parmis les six autres. Ce projet m'a donc permis d'approfondir mes connaissances en machine learning en employant une méthode qui pour moi était inconnue il y a encore une semaine.