Statistiques et Data Mining
TD 4 (Régression)

Youna Froger, Alice Gyde et Mathis Quais

10 décembre 2024

1 Réponses au questionnaire

1.1 What is regression in machine learning ?

C'est une technique d'apprentissage supervisé où les résultats en sortie sont comparés à une droite continue suivant une forme $ax + b$. Le résultat attendu est donc la répartition de nos données sous forme de points autour d'une droite continue (ou d'une autre forme en fonction de l'algorithme de régression choisi).

1.2 Why and when should regression be applied ?

Cela permet d'avoir une vue d'ensemble sur les données, de voir les fluctuations et de comprendre l'interdépendance des variables. On l'utilise lorsque le résultat est linéaire.

1.3 What are commonly used algorithms for regression ?

Les cinq algorithmes principaux sont :
- **Régression linéaire** : établit une relation linéaire entre variables dépendantes et indépendantes.
- **Régression logistique** : utilise une classification binaire et calcule la probabilité qu'une variable appartienne à une classe.
- **Régression polynomiale** : utilise une équation polynomiale pour repérer la relation (linéaire ou non).
- **Régression des séries temporelles** : se base sur des observations passées pour prédire le futur.
- **SVR (Support Vector Regression)** : se base sur les SVM et cherche à minimiser l'erreur tout en maximisant la marge.

2 Exercices du TD

Voir le lien GitHub : https://github.com/younafroger/TAL$_H NM2$.

Les données utilisées sont issues du *California Housing Dataset*. Le code sépare les données en jeux de test et d'entraînement pour appliquer différents modèles : régression linéaire, régression Ridge, régression Lasso, Random Forest Regressor et Gradient Boosting Regressor.

En premier lieu, analysons les calculs des métriques pour nos trois tests :

1. Application des modèles de régression sur notre jeu de données de base.
2. Application avec modification de certains hyperparamètres.
3. Réduction de la matrice avec la méthode de réduction de dimensions LSA.

Pour le premier test, les métriques telles que la MSE et la MAE indiquent une précision modérée des prédictions, avec un score relativement bas. Cela indiquerait que les modèles, dans leur configuration standard (sans modification des hyperparamètres), montrent des limites.

Pour le deuxième test, après modification des hyperparamètres, les métriques (surtout la MSE) montrent une amélioration. Cependant, le R^2 score a diminué. Tous ces éléments laissent penser que les changements d'hyperparamètres ont eu un effet négatif et n'ont pas permis au modèle de mieux capturer la structure des données. Cela est probablement dû à nos choix de modifications, qui n'étaient pas les plus judicieux.

Enfin, dans le troisième test, nous avons réduit notre matrice initiale à quatre dimensions (au lieu de huit initialement), ce qui nous donne des métriques dont les scores sont dégradés par rapport au deuxième test. Nous avons donc une perte d'information importante après une réduction à quatre dimensions. Nous essaierons de changer le paramètre `n_components` pour d'autres calculs et graphiques.

Modèles initiaux :

```
Model Performance:
                             Model       MSE       MAE  R2 Score
0                Linear Regression  0.555892  0.533200  0.575788
1                 Ridge Regression  0.555803  0.533204  0.575855
2                 Lasso Regression  0.612990  0.581291  0.532214
3          Random Forest Regressor  0.255368  0.327543  0.805123
4  Gradient Boosting Regressor      0.293997  0.371643  0.775645
Plotting performance metrics...
```

Modèles avec changement des hyperparamètres :

```
Model Performance:
                             Model       MSE       MAE  R2 Score
0                Linear Regression  0.662396  0.604625  0.494512
1                 Ridge Regression  0.555803  0.533204  0.575855
2                 Lasso Regression  0.613512  0.581607  0.531817
3          Random Forest Regressor  0.255368  0.327543  0.805123
4  Gradient Boosting Regressor      0.344685  0.385128  0.736964
Plotting performance metrics...
```

Modèles après réduction LSA :

```
Model Performance:
                             Model       MSE       MAE  R2 Score
0                Linear Regression  1.293401  0.898471  0.012979
1                 Ridge Regression  1.293401  0.898471  0.012979
2                 Lasso Regression  1.294546  0.900179  0.012106
3          Random Forest Regressor  1.215758  0.856727  0.072230
4  Gradient Boosting Regressor      1.163658  0.834234  0.111989
```

Après avoir calculé les métriques MAE, MSE, et R^2 score, nous avons généré des graphiques de régression pour nos cinq modèles. De manière générale, les modèles Random Forest et Gradient Boosting semblent les plus efficaces, car les données (points) suivent correctement la ligne de régression proposée par ces modèles. Cependant, après modification des hyperparamètres, une plus large dispersion des points est visible. Cela confirme que cette configuration a un effet négatif.

Lors de la réduction de la matrice, les graphiques montrent une dispersion importante des points, qui ne suivent plus la ligne. En changeant le paramètre `n_components`, nous avons constaté qu'à partir d'une réduction en dessous de six dimensions, cette déformation apparaît.

Voir annexe `illustration_regression.pdf` page 1.

Les graphiques de résidus montrent les erreurs de prédiction (différences entre valeurs réelles et prédites). Pour le deuxième test, les résidus sont très éparpillés, alors qu'ils sont davantage centrés autour de zéro dans le premier test.

Voir annexe `illustration_regression.pdf` page 2.

3 Conclusion

Lors de ce TP, nous avons étudié la régression appliquée à un jeu de données sur les prix des maisons en fonction de huit paramètres. Nous avons remarqué que la modification des hyperparamètres de nos modèles réduit la précision des prédictions et augmente les erreurs (comme le montrent les graphiques des résidus). Cela est probablement dû à des choix inadaptés dans les modifications. Par ailleurs, une réduction excessive des dimensions de la matrice (moins de six) engendre une perte d'information importante, ce qui dégrade les résultats et rend les graphiques de régression inutilisables.