Rapport: Analyse de performance des modèles de classification supervisée

ANDRIATSIFERANA No Kanto Lorida

June 10, 2025

Abstract

Ce rapport présente une comparaison globale et une interprétation des performances de cinq modèles de classification appliqués à un problème binaire : K-Nearest Neighbors (KNN), Arbre de décision, Régression logistique, Support Vector Machine (SVM) et Regession linéaire.

1 Régression linéaire

1.1 Rappel

La **régression linéaire** modélise une relation linéaire entre une ou plusieurs variables explicatives (features) X et une variable cible y. Elle cherche une droite (ou un hyperplan) qui minimise la somme des carrés des erreurs entre les prédictions et les vraies valeurs. Formule :

$$\hat{y} = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \ldots + \beta_n X_n$$

1.2 Résultat et interprétation

Métrique	**Valeur**
MAE	0.2563
MSE	0.1157
RMSE	0.4271

- MAE (Mean Absolute Error) = 0.2563
  En moyenne, la prédiction du modèle est à 0.2563 unités de la vraie valeur.C'est une erreur modérée, mais elle doit être interprétée en fonction de l'échelle de la variable cible.

- MSE (Mean Squared Error) = 0.1157
  Les erreurs quadratiques sont légèrement plus élevées que l'erreur absolue. Cela signifie que certaines erreurs importantes (outliers) sont présentes, car le MSE pénalise davantage les grandes erreurs.

- RMSE (Root Mean Squared Error) = 0.4271
  Environ 42.71 % de la variance de la variable cible est expliquée par le modèle. Ce score est relativement faible, ce qui indique que le modèle n'explique pas bien la variabilité des données — probablement parce que la régression linéaire est **mal adaptée à une tâche de classification binaire comme "RainTomorrow".

2 KNN (K-Nearest Neighbors)

2.1 Rappel

Le **KNN** est un algorithme de classification non paramétrique. Il prédit la classe d'un échantillon en regardant les k plus proches voisins (selon une distance, souvent euclidienne) et en prenant la classe majoritaire parmi eux.

Accuracy: proportion d'exemples bien classés sur le total.

Indice de Jaccard: intersection sur union entre les vraies classes et les classes prédites (pour la classe positive).

$$Jaccard = \frac{TP}{TP + FP + FN}$$

F1-Score: moyenne harmonique entre précision (precision) et rappel (recall).

$$F1 = 2 \cdot \frac{Precision \cdot Recall}{Precision + Recall}$$

2.2 Résultat et interprétation

Métrique	**Valeur**
Accuracy	0.8183
Indice de Jaccard	0.4251
F1-Score	0.5966

- Accuracy = 0.8183
  Le modèle prédit correctement la variable "RainTomorrow" dans 81.83% des cas. C'est un bon score global, mais il peut être trompeur si les classes sont déséquilibrées (ex. : beaucoup de "No" et peu de "Yes").

- Indice de Jaccard = 0.4251
  Cet indice mesure le recouvrement entre les classes prédites et les classes réelles. Un score de 0.4251 indique un recouvrement modéré, ce qui est acceptable, mais montre qu'il y a encore des erreurs importantes.

- F1-Score = 0.5966
  Le F1-score combine précision et rappel : ici, le score de 0.5966 suggère un équilibre moyen entre la capacité du modèle à identifier correctement les cas de pluie et à éviter les faux positifs. Ce score montre que le modèle se débrouille, mais reste loin d'être optimal.

3 Decision Tree

3.1 Rappel

Un **arbre de décision** est un modèle prédictif qui divise les données en sous-groupes successifs à partir de règles logiques sur les variables. À chaque nœud, il choisit une variable et un seuil pour maximiser la séparation des classes (souvent via l'entropie ou l'indice de Gini).

3.2 Résultat et interprétation

Métrique	**Valeur**
Accuracy	0.8183
Indice de Jaccard	0.4803
F1-Score	0.6490

- Accuracy = 0.8183
  L'arbre de décision prédit correctement 81.83% des cas. Ce score est équivalent à celui du KNN, ce qui signifie que le modèle est globalement bon pour distinguer pluie / pas de pluie.

- Indice de Jaccard = 0.4803
  Le score de 0.4803 indique un meilleur recouvrement entre les prédictions et les vraies classes qu'avec KNN (0.4251). Cela suggère que l'arbre génère moins de faux positifs ou faux négatifs.

- F1-Score = 0.6490
  Ce score mesure l'équilibre entre précision et rappel, et il est meilleur que celui du KNN (0.5966). Cela montre que l'arbre est plus efficace pour détecter les cas de pluie tout en limitant les erreurs.

4 Régression logistique

4.1 Rappel

La régression logistique est un modèle linéaire utilisé pour la classification binaire. Elle estime la probabilité qu'une observation appartienne à une classe, via la fonction sigmoïde :

$$P(y = 1|X) = \frac{1}{1 + e^{-(\beta_0 + \beta_1 x_1 + ... + \beta_n x_n)}}$$

Le modèle prédit la classe 1 si P $>$ 0.5, sinon 0.

4.2 Résultat et interprétation

Métrique	**Valeur**
Accuracy	0.8382
Indice de Jaccard	0.5204
F1-Score	0.6845
LogLoss	0.3571

- Accuracy = 0.8382
  Le modèle prédit correctement la pluie ou non dans 83.82% des cas. C'est le meilleur score de précision globale jusqu'ici (mieux que KNN et l'arbre de décision).

- Indice de Jaccard = 0.5204
  Ce score reflète un recouvrement significatif entre les classes prédites et les vraies classes. À 0.5204, c'est un score solide, meilleur que les modèles précédents (KNN : 0.425, Arbre : 0.480).

- F1-Score = 0.6845
  Ce score combine la précision et le rappel. Il montre que le modèle est capable de bien détecter les cas de pluie sans trop d'erreurs. Ce F1-Score est supérieur à tous les modèles précédents, ce qui montre un bon équilibre entre sensibilité et exactitude.

- LogLoss = 0.3571
  Le LogLoss mesure la qualité des probabilités prédites. Plus ce score est faible, plus le modèle est confiant et précis dans ses prédictions probabilistes.
  Un LogLoss de 0.3571 est très raisonnable : les prédictions ne sont pas seulement bonnes, elles sont aussi bien calibrées.

5 SVM (Support Vector Machine)

5.1 Résultat et interprétation

Métrique	Valeur
Accuracy	0.7191
Indice de Jaccard	0.0000
F1-Score	0.0000
LogLoss	0.3928

- Accuracy = 0.7191
  Le modèle obtient une précision globale de 71.91%, ce qui est nettement inférieur à la régression logistique, au KNN ou à l'arbre de décision.

- Indice de Jaccard = 0.0000
  Un score nul signifie que le modèle ne détecte pas du tout la classe positive (RainTomorrow = Yes). En d'autres termes, le modèle prédit uniquement la classe majoritaire (No), sans jamais reconnaître les jours de pluie.

- F1-Score = 0.0000
  Comme le F1-score combine précision et rappel, un score de 0 indique que le modèle ne reconnaît jamais correctement la classe "Yes", ce qui le rend inutile pour détecter la pluie.

- LogLoss = 0.3928
  Le LogLoss est légèrement plus élevé que celui de la régression logistique (0.357), ce qui indique que même les probabilités prédites sont moins bien calibrées.

5.2 Causes possibles de mauvais résultat

- Déséquilibre des classes
  Le jeu de données contient probablement beaucoup plus d'exemples de la classe 0 que de la classe 1. Le SVM a "joué la sécurité" en prédisant uniquement la classe majoritaire.

- SVM mal paramétré
  Par défaut, "SVC()" n'utilise pas de pondération entre les classes. Et peut être très sensible à la mauvaise échelle des variables (il est nécessaire de standardiser les données).

6 Interprétation globale

6.1 Récapitulatif des performances

Modèle	Accuracy	Jaccard	F1-score	Remarque principale
Régression logistique	**0.8382**	**0.5204**	**0.6845**	Le plus équilibré et performant globalement
Decision Tree	0.8183	0.4803	0.6490	Bon compromis, interprétable
KNN	0.8183	0.4251	0.5966	Correct mais un peu moins précis
SVM	0.7191	0.0000	0.0000	Échec total : ne détecte pas la classe 1
Régression linéaire	–	–	–	Modèle de régression, non comparable ici

6.2 Interprétation comparative

- Régression logistique :
  Le plus performant dans l'ensemble : meilleur équilibre entre précision, rappel, et exactitude. Stabilité élevée, interprétable. Excellent choix pour un premier modèle de base. Recommandé comme modèle de référence.

- Decision Tree :
  Bonne précision, bon F1-score. Moins bon que la régression logistique, mais plus interprétable (peut être visualisé). Peut sur-apprendre si non régularisé, mais utile en cas de non-linéarité.

- KNN :
  Performances correctes, mais un peu en retrait. Sensible à la distribution des données et à la distance. Efficace sur petits jeux de données bien préparés, mais moins robuste.

- SVM :
  Échec total ici : ne détecte aucune instance de la classe positive. Résultat probable d'un déséquilibre de classes + mauvais prétraitement (non-standardisation). Nécessite une correction du code pour exploiter son potentiel.

- Régression linéaire :
  Donne de bonnes performances sur une tâche de régression (MAE : 0.2563, RMSE : 0.4271). Utile si ta variable cible est continue, pas binaire. Peut être utilisé pour la sélection de variables grâce à la pénalisation L1.

7 Conclusion

Après avoir appliqué plusieurs algorithmes d'apprentissage supervisé sur le dataset météorologique de Sydney, nous avons obtenu des performances variées selon le type de modèle.

La régression linéaire, bien qu'initialement testée, ne s'adapte pas parfaitement à cette tâche de classification binaire. Elle est plus appropriée pour des prédictions continues. Les scores R^2, MAE et MSE confirment qu'elle ne capture pas correctement la logique binaire de "RainTomorrow".

Le modèle KNN (k=4) offre une performance correcte, notamment avec une bonne précision (accuracy) et un indice de Jaccard équilibré. Ce modèle est simple mais sensible au choix de 'k' et à l'échelle des variables.

L'arbre de décision, avec une profondeur maximale de 4, s'avère être un compromis intéressant. Il offre une interprétabilité naturelle grâce à sa structure arborescente, mais ses scores sont légèrement en dessous de ceux de la régression logistique et du SVM.

La régression logistique se distingue par son bon équilibre entre les scores de classification (Accuracy, F1, Jaccard) et un LogLoss très raisonnable, montrant que les probabilités prédites sont cohérentes.

Le SVM, avec un noyau RBF, affiche les meilleurs scores globaux, surpassant même la régression logistique en F1-Score. C'est donc un excellent candidat pour ce problème, bien qu'il soit plus coûteux en calcul et moins interprétable.