Master 1 Informatique - Techniques d'Apprentissage Artificiel
Projet : Prédictions de matchs de ligue 1 - Machine Learning
Rendu : 04 décembre 2025
Auteurs : RONDEAU Arthur & NACEUR Hella
Dépôt GitHub : https://github.com/Arthur94R/LIGUE1-PREDICTIONS.git

---

1. INTRODUCTION

1.1 Contexte et Problématique

Le football est l'un des sports les plus populaires au monde, et la prédiction des résultats de matchs représente un défi majeur tant pour les passionnés que pour les professionnels du secteur des paris sportifs. L'objectif de ce projet est de développer un modèle de machine learning capable de prédire les résultats des matchs de Ligue 1 pour la saison 2025-2026.

1.2 Objectifs

- Construire un modèle de classification pour prédire trois classes : **Victoire à domicile**, **Match nul**, **Victoire à l'extérieur**
- Explorer différents algorithmes de machine learning (RandomForest, XGBoost)
- Gérer le déséquilibre des classes inhérent aux données de football
- Analyser les limites de la prédiction dans un contexte aussi imprévisible que le football

---

2. DONNÉES

2.1 Source des données

Les données proviennent de 8 saisons consécutives de Ligue 1 :

- **Période :** 2017-2018 à 2024-2025
- **Nombre total de matchs :** 2791
- **Format :** Fichiers CSV contenant les statistiques de chaque match

2.2 Variables collectées

Variables de résultat :

- `FTHG` / `FTAG` : Buts marqués à domicile / extérieur
- `FTR` : Résultat final (H/D/A) - **variable cible**

Statistiques de match :

- Tirs (HS, AS), Tirs cadrés (HST, AST)
- Fautes (HF, AF), Corners (HC, AC)
- Cartons jaunes (HY, AY), Cartons rouges (HR, AR)

2.3 Preprocessing

Étapes de nettoyage

1. Combinaison des 8 fichiers CSV
2. Sélection des colonnes nécessaires
3. Gestion des valeurs manquantes (suppression des lignes incomplètes)
4. Conversion des dates au format datetime
5. Création d'une colonne "Saison" pour identifier chaque saison

Résultat : Dataset final de 2791 matchs avec 20 colonnes pertinentes.

---

3. FEATURE ENGINEERING

3.1 Principe

Pour prédire un match futur, le modèle ne peut pas utiliser les statistiques du match lui-même (qui n'existent pas encore). Il faut donc créer des **features basées sur l'historique** des équipes.

3.2 Calcul de la forme récente

Pour chaque match, nous calculons la **forme des 5 derniers matchs** de chaque équipe dans la saison en cours :

Pour l'équipe à domicile (7 features) :

- `Home_goals_scored` : Moyenne de buts marqués
- `Home_goals_conceded` : Moyenne de buts encaissés
- `Home_wins`, `Home_draws`, `Home_losses` : Nombre de victoires/nuls/défaites
- `Home_shots`, `Home_shots_on_target` : Moyennes de tirs et tirs cadrés

Pour l'équipe à l'extérieur (7 features) :

- Mêmes statistiques (`Away_goals_scored`, `Away_goals_conceded`, etc.)

Total : 14 features prédictives

3.3 Gestion des cas particuliers

Début de saison :

- Les premiers matchs n'ont pas d'historique
  - Solution : Attribution de valeurs par défaut (0)
  - Impact : ~200 matchs sur 2791 concernés

Équipes promues/reléguées :

- Chaque saison est traitée indépendamment
  - Une équipe promue démarre sans historique en Ligue 1

---

4. MODÉLISATION

4.1 Préparation des données

Encodage de la variable cible :

'A' (Away win) → 0

'D' (Draw) → 1

'H' (Home win) → 2

Division train/test :

- Train : 80% matchs

- Test : 20% matchs

- Stratification : aléatoire (random_state=42)

4.2 Déséquilibre des classes

Distribution observée :

Home win (H) : ~ 45% des matchs

Away win (A) : ~ 30% des matchs

Draw (D) : ~ 25% des matchs

Problème : Les modèles tendent à sur-prédire les victoires à domicile (classe majoritaire).

Solution : Utilisation du paramètre `class_weight='balanced'` pour pondérer les classes minoritaires.

4.3 Algorithmes testés

Afin d'entraîner notre modèle, nous avons décidé de retenir deux algorithmes : RandomForest Classifier et XGBoost Classier. Pour chaque algorithme, nous avons réalisé un deuxième essai en pondérant les classes minoritaires avec le paramètre `class_weight='balanced'`.

RandomForest Classifier

Hyperparamètres :

- n_estimators = 100

- max_depth = 10

- random_state = 42

Avantages :

- Robuste au surapprentissage
- Gère bien les features hétérogènes
- Fournit l'importance des features

XGBoost Classifier

Hyperparamètres :

- n_estimators = 100

- max_depth = 6

- learning_rate = 0.1

Avantages :

- Apprentissage séquentiel (boosting)
- Généralement plus performant que RandomForest
- Optimisé pour la vitesse

5. RÉSULTATS

5.1 Comparaison des modèles

Modèle	Accuracy	Precision Draw	Recall Draw	F1-score Draw
RandomForest	47.76%	30%	2%	0.04
RandomForest balanced	44.19%	17%	25%	0.20
XGBoost	42.04%	32%	18%	0.23
XGBoost balanced	37.75%	22%	22%	0.22

On compare ici precision, recall et f1-score sur le 'Draw' car c'est la difficulté rencontrée sur ce projet, donc la plus intéressante à comparer. C'est suite à cette difficulté que nous utilisons plusieurs modèles afin d'obtenir des résultats de plus en plus intéressants.

5.2 Analyse du meilleur modèle (RF balanced)

Matrice de confusion :

	Prédiction			
Réalité	**Away win**	**Draw**	**Home win**	**Total de matchs**
Away win	74	33	60	← 167 matchs
Draw	58	25	63	← 146 matchs
Home win	57	41	148	← 246 matchs

Exemple d'interprétation :

Sur les 167 victoires de l'équipe extérieure (Away), le modèle RandomForest Classifier balanced a trouvé en majorité des bons résultats avec la victoire de l'équipe extérieure (74) mais a environ 50% d'échec en prédisant 33 nuls et 60 victoires de l'équipe domicile.

Rapport de classification :

	Precision	**Recall**	**F1-score**
Away win	0.44	0.39	0.42
Draw	0.17	0.25	0.20

Home win	0.60	0.55	0.57

Accuracy globale : **44.19%**

Interprétation :

- **Home win :** Classe la mieux prédite (F1 = 0.57)
- **Away win :** Performance moyenne (F1 = 0.42)
- **Draw :** Classe la plus difficile (F1 = 0.20)

6. DISCUSSION

6.1 Interprétation des résultats

Accuracy de 44.19% : Est-ce acceptable ?

OUI, pour plusieurs raisons :

1. **Le football est intrinsèquement imprévisible**
    - Nombreux facteurs incontrôlables (arbitrage, météo, chance)
    - Les experts humains (bookmakers) dépassent rarement 50-55% d'accuracy
2. **Les matchs nuls sont quasi-impossibles à prédire**
    - Résultat le plus aléatoire
    - Même les bookmakers professionnels échouent majoritairement
3. **Contexte académique**
    - L'objectif est de démontrer la méthodologie, pas d'atteindre 90% d'accuracy
    - Un modèle à 44% qui prédit équitablement les 3 classes est meilleur qu'un modèle à 50% biaisé

6.2 Limites du projet

1. Features limitées

- Pas de confrontations directes entre équipes
- Pas de classement au moment du match
- Absence de données contextuelles (blessures, météo, enjeux)

2. Historique court

- Seulement 5 derniers matchs
- Pas d'utilisation des données de la saison précédente

3. Déséquilibre des classes

- Malgré le rééquilibrage, les nuls restent sous-prédits

Notre modèle (43.73%) est donc cohérent avec un projet académique disposant de features basiques.

7. PISTES D'AMÉLIORATION

7.1 Enrichissement des features

Features calculables à partir des données existantes :

- Ratio buts pour/contre : `goals_scored / goals_conceded`
- Forme domicile/extérieur séparée :
    - 5 derniers matchs À DOMICILE pour l'équipe home
    - 5 derniers matchs À L'EXTÉRIEUR pour l'équipe away
- Confrontations directes : historique des matchs entre les deux équipes
- Classement approximatif au moment du match
- Prendre en compte les résultats des saisons passées

Features nécessitant des données externes :

- Blessures et suspensions
- Valeur marchande des effectifs
- Météo (température, pluie)
- Contexte (match décisif, période de trêve)

7.2 Autres algorithmes

Modèles à tester :

- **LightGBM** : Variante de XGBoost, souvent plus rapide
- **CatBoost** : Gère bien les variables catégorielles
- **Réseaux de neurones** : Pour capturer des interactions complexes
- **Ensembles de modèles** : Voting / Stacking de plusieurs modèles

8. CONCLUSION

8.1 Bilan du projet

Ce projet a permis d’appliquer une méthodologie complète de **machine learning**, en partant de la collecte de données à l'évaluation de modèles.

Nous avons pu également comprendre les défis spécifiques du football : déséquilibre des classes, imprévisibilité intrinsèque.

Enfin, la comparaison de plusieurs algorithmes : RandomForest vs XGBoost, avec/sans équilibrage nous a permis d'obtenir plusieurs résultats cohérents dont **notre résultat final** : 44.19% d'accuracy avec un modèle équilibré.

8.2 Perspectives

Court terme :

- Implémenter les features d'amélioration (classement, domicile/extérieur, etc…)

- Tester des modèles ensemblistes

Long terme :

- Intégrer des données externes (transferts, blessures)

- Étendre à d'autres championnats (Premier League, La Liga)