projet BI et ML

réalisé par : Eya Bouajila; Nadine Hamada, onan Mermoz Effi, Ayer Med Amine

2022-2023

Rapport

Table des matières

Liste des figures

Liste des tables

1 Introduction générale

Le domaine de la technologie de l'informatique joue un rôle important dans la numérisation de gestion d'un nombre important de données. C'est une solution optimale pour faciliter le travail et le rendre plus précis en s'appuyant sur les points d'amélioration d'organisation du travail.

Les experts en football de nos jours tentent de faire le plus d'effort possible pour gagner plus de temps et numériser les systèmes de prédictions des matchs. Dans le domaine du foot, et plus particulièrement dans le football britannique, l'informatique décisionnelle (Data Science) occupe une grande place. En effet le processus du prédictions des résultats des matchs consiste généralement en un nombre important des phases parmi lesquelles on cite : comprendre la logique métier, obtenir tout l'historique des résultats,choisir le meilleur modèle de Machine Learning possible pour la prédiction et en fin tester pour obtenir les meilleurs résultats.

En effet la prédiction des scores des matchs et des équipes gagnantes joue un rôle important dans les places de loterie donc ils cherchent toujours à améliorer leurs moyens de prédiction pour assurer leurs gains. Notre projet se situe dans ce thème, qui consiste à réaliser une application pour prédire les gagnants dans les matchs du football britanniques matchs et relativement à l'historique des dernières 30 ans. L'objectif de ce projet est de trouver le meilleur modèle de prédiction en minimisant les erreurs.

Le présent rapport décrit les différentes étapes de notre travail.

2 Cadre général du projet

2.1 Introduction

Pour la réalisation d'une application, une étude préliminaire est une phase primordiale qui permet l'analyse, l'évaluation et la critique des applications similaires pour arriver enfin à proposer une solution adéquate. On propose, tout d'abord, de présenter le système actuel et son fonctionnement, ainsi que quelques applications similaires et leurs critiques afin de concevoir une idée générale pour proposer une meilleure solution.

2.2 Etat de l'art

2.2.1 Etude de l'existant

La prédiction des parties gagnantes dans les matchs du foot est très complexe et divisée en plusieurs parties : la compréhension du logique métier (tirs cadrés, score à la mi-temps ..), la collecte des données historiques du foot, la création d'un modèle de prédiction du machine learning, et tester pour trouver les meilleurs résultats en minimisant les erreurs. meme s'il y a d'autres solutions sur le marché, cette prédiction demeure limitée avec des modèles classiques.

2.2.2 Revue de quelques applications similaires

Il existe des solutions de prédiction des scores des matchs du football. On cite dans ce qui suit :

Nom	Description
	Matchguess est une application de paris sportifs qui permet de prédire le résultat d'un match de football.
	Stats24 : Football Stats, Odds, Betting Predictions est une application analytique permettant de prédire le résultat des matchs de football. L'algorithme de l'application est basé sur l'analyse des indicateurs des équipes, des joueurs et des entraîneurs. La base de données de l'application comprend des informations statistiques détaillées sur les compétitions de football.

Table 1: Applications similaires.

2.2.3 Critique de l'existant

Plusieurs applications offrent des fonctionnalités et des solutions pour la bonne prédiction des scores des matchs et des équipes gagnantes, mais le mieux ce n'est pas celui qui propose le plus de fonctionnalités, mais de disposer de toutes les fonctionnalités essentielles dont les personnes qui jouent en loto ont besoin pour atteindre les objectifs de leurs stratégie. C'est dans ce contexte que nous cherchons à faciliter la prédiction qui va répondre aux besoins de ces individus ou entreprises.

2.2.4 Solution envisagée

on propose alors un système de prédiction et visualisation graphique des scores des matchs britanniques. La gestion de ces fonctionnalités à travers notre application assurera les points suivants :

1. Une meilleure prédiction : permet d'ameliorer le processus de prédiction et maximiser le gain en éliminant toute source d'erreurs possible.
2. Visibilité totale des scores : le suivi et la sauvegarde des scores dans l'historique des données.
3. Une meilleure réponse aux attentes des individus /entreprises de loto: Gérer les demandes et leurs mises à jour en temps réel lors de la confirmation et la mise en place des activités.

2.3 Méthodologie de travail

Le choix d'une méthodologie agile est le choix le plus répandu pour les différents types de projets. La méthode agile se base sur un cycle de développement qui porte sur l'interaction avec le client.
L'implication du client dans le projet au cours de la réalisation donne toujours une vision plus claire et améliore le travail au fur et à mesure.
On a choisi d'utiliser la méthode CRISP-DM comme méthode de travail, en raison de sa pertinence par rapport à nos exigences de BI.

2.3.1 Gestion du projet avec CRISP-DM

Travailler avec CRISP-DM signifie passer par 6 étapes dans la réalisation du projet :

1. Compréhension du problème métier : Cette étape consiste à comprendre le contexte commercial et les objectifs du projet.
2. Compréhension des données : Cette étape implique la collecte des données et leur exploration pour comprendre leur qualité et leur adéquation pour le projet.
3. Préparation des données : Cette étape consiste à nettoyer et à transformer les données en vue de les préparer pour l'analyse.
4. Modélisation : Cette étape consiste à créer des modèles d'analyse de données pour répondre aux objectifs du projet.
5. Évaluation : Cette étape consiste à évaluer les modèles créés pour déterminer leur qualité et leur adéquation pour les objectifs du projet.
6. Déploiement : Cette étape consiste à mettre en œuvre les modèles dans un environnement opérationnel.
7. Surveillance : Cette étape consiste à surveiller le système en production pour s'assurer que les résultats sont conformes aux attentes.

La planification sert à réaliser un projet bien organisé pour atteindre les objectifs souhaités,on commence tout d'abord par l'affectation des rôles CRISP-DM, ensuite la planification des réunions, le Backlog du produit et enfin le plan des livrables et des sprints.

La figure suivante présente le cycle de vie du CRISP-DM :

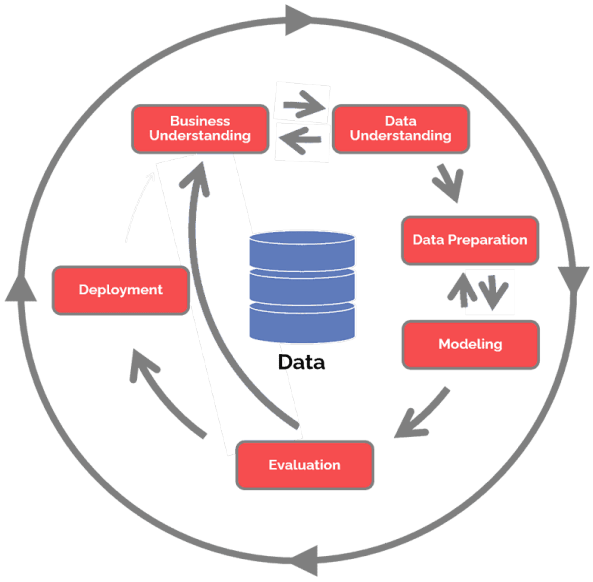


Figure 1: Cycle de vie de CRISP-DM

2.4 Conclusion

On a présenté dans ce chapitre le cadre général et le contexte du sujet. On a cité aussi la critique de l'existant et l'étude des applications similaires qui nous ont guidé vers une solution adéquate qui répond aux besoins du projet.

3 Processus décisionnel

3.1 Introduction

pour permettre un pilotage efficace et faire face à la concurrence, se doter d'un systeme d'information décisionnel est primordial afin d'aider les décideurs à prendre les bonnes décisions.
Compte tenue de ces évolutions, il est nécessaire de bien définir l'informatique décisionnelle.

3.2 Définition

L'informatique décisionnelle, également connue sous le nom de Business Intelligence (BI), est un ensemble de techniques et d'outils qui permettent de collecter, d'analyser et de présenter des données pour aider les entreprises à prendre des décisions éclairées. L'objectif de l'informatique décisionnelle est de fournir des informations précieuses et des perspectives pour aider les décideurs à comprendre les tendances, les opportunités et les défis de leur entreprise. Les systèmes d'informatique décisionnelle peuvent inclure des

technologies telles que la modélisation des données, l'analyse de données, l'exploration de données, la visualisation de données, les tableaux de bord et les rapports. Ces systèmes peuvent être utilisés pour suivre les performances de l'entreprise, identifier les tendances et les opportunités, et prendre des décisions basées sur des données concrètes plutôt que sur des conjectures ou des suppositions. En résumé,

l'informatique décisionnelle est un domaine de l'informatique qui vise à aider les entreprises à prendre des décisions plus informées et plus efficaces en utilisant des données et des analyses.

3.3 Objectifs

Les objectifs de l'informatique décisionnelle sont multiples, mais ils ont tous en commun l'objectif général de fournir des informations utiles pour aider les entreprises à prendre des décisions éclairées. Voici quelques-uns des objectifs les plus courants de l'informatique décisionnelle :

- Collecter et intégrer des données provenant de différentes sources : L'un des principaux objectifs de l'informatique décisionnelle est de collecter et d'intégrer des données provenant de différentes sources au sein d'une entreprise, telles que les données de ventes, les données de production, les données de marketing, les données financières, etc. Les données sont souvent stockées dans des entrepôts de données ou des data marts pour faciliter l'accès et la manipulation des données.
- Analyser les données pour identifier des tendances et des opportunités : L'informatique décisionnelle utilise des techniques d'analyse de données pour identifier les tendances et les opportunités cachées dans les données. Ces analyses peuvent inclure des techniques telles que l'analyse de corrélation, l'analyse de régression, l'analyse de clustering, l'analyse de segmentation et bien d'autres encore.
- Fournir des informations en temps réel : L'informatique décisionnelle permet souvent de fournir des informations en temps réel aux décideurs de l'entreprise. Cela peut inclure des tableaux de bord en temps réel qui présentent les données clés et les indicateurs de performance de l'entreprise.
- Améliorer la prise de décision : Enfin, l'un des principaux objectifs de l'informatique décisionnelle est d'améliorer la prise de décision en fournissant des informations factuelles et des analyses approfondies aux décideurs de l'entreprise. Cela peut aider les entreprises à prendre des décisions plus éclairées et plus stratégiques, ce qui peut conduire à une meilleure performance et à une croissance plus rapide.

3.4 Etapes du projet

L'informatique décisionnelle implique plusieurs étapes pour permettre aux entreprises de collecter, analyser et présenter des données de manière à faciliter la prise de décisions éclairées. Voici les étapes typiques de l'informatique décisionnelle :

- Collecte de données : La première étape de l'informatique décisionnelle consiste à collecter des données provenant de différentes sources, telles que des bases de données, des feuilles de calcul, des fichiers texte, des sources externes, etc. Les données sont souvent stockées dans un entrepôt de données centralisé pour faciliter leur accès.

Figure 2: site de récupération des données

- Nettoyage et transformation de données : Les données collectées peuvent contenir des erreurs, des doublons ou des incohérences. La deuxième étape consiste donc à nettoyer et à transformer les données pour garantir leur qualité et leur cohérence. Cela peut inclure la normalisation de données, la suppression des doublons, la conversion de formats, etc.

```
[ ] # Deleting non fruitful features
    df_18= df_18[['Div', 'Date', 'HomeTeam', 'AwayTeam', 'FTHG', 'FTAG', 'FTR', 'HTHG',
           'HTAG', 'HTR', 'Referee', 'HS', 'AS', 'HST', 'AST', 'HF', 'AF', 'HC',
           'AC', 'HY', 'AY', 'HR', 'AR']]
    df_19= df_19[['Div', 'Date', 'HomeTeam', 'AwayTeam', 'FTHG', 'FTAG', 'FTR', 'HTHG',
           'HTAG', 'HTR', 'Referee', 'HS', 'AS', 'HST', 'AST', 'HF', 'AF', 'HC',
           'AC', 'HY', 'AY', 'HR', 'AR']]
    df_20= df_20[['Div', 'Date', 'HomeTeam', 'AwayTeam', 'FTHG', 'FTAG', 'FTR', 'HTHG',
           'HTAG', 'HTR', 'Referee', 'HS', 'AS', 'HST', 'AST', 'HF', 'AF', 'HC',
           'AC', 'HY', 'AY', 'HR', 'AR']]
    df_21= df_21[['Div', 'Date', 'HomeTeam', 'AwayTeam', 'FTHG', 'FTAG', 'FTR', 'HTHG',
           'HTAG', 'HTR', 'Referee', 'HS', 'AS', 'HST', 'AST', 'HF', 'AF', 'HC',
           'AC', 'HY', 'AY', 'HR', 'AR']]
    df_22= df_22[['Div', 'Date', 'HomeTeam', 'AwayTeam', 'FTHG', 'FTAG', 'FTR', 'HTHG',
           'HTAG', 'HTR', 'Referee', 'HS', 'AS', 'HST', 'AST', 'HF', 'AF', 'HC',
           'AC', 'HY', 'AY', 'HR', 'AR']]
    df= df[['Div', 'Date', 'HomeTeam', 'AwayTeam', 'FTHG', 'FTAG', 'FTR', 'HTHG',
           'HTAG', 'HTR', 'Referee', 'HS', 'AS', 'HST', 'AST', 'HF', 'AF', 'HC',
           'AC', 'HY', 'AY', 'HR', 'AR']]
```

Figure 3: Nettoyage des données sur Jupyter

- Analyse de données : La troisième étape de l'informatique décisionnelle consiste à analyser les données pour en tirer des informations utiles. Cela peut inclure l'utilisation d'outils de visualisation de données, d'outils de modélisation statistique, d'algorithmes d'apprentissage automatique, etc.

```
[ ] # Perform the chi-square test
    chi2, p, dof, expected = chi2_contingency(cross_tab_RED)

    # Print the results
    print("Chi-square statistic:", chi2)
    print("p-value:", p)
    print("Degrees of freedom:", dof)
    print("Expected values:", expected)
```

```
Chi-square statistic: 28.525266510186118
p-value: 9.759926534095438e-06
Degrees of freedom: 4
Expected values: [[585.68958743  36.16502947   2.1453831 ]
 [492.76768173  30.42730845   1.80500982]
 [832.54273084  51.40766208   3.04960707]]
```

the calculated test statistic is less than or equal to the critical value, fail to reject the null hypothesis. it means that there is not a significant association between the two variables.

Figure 4: Analyse préliminaire avec des Tests Statistiques

- Présentation de données : La quatrième étape consiste à présenter les données analysées de manière à faciliter la prise de décisions éclairées. Cela peut inclure la création de tableaux de bord interactifs, de rapports personnalisés, de visualisations de données, etc.

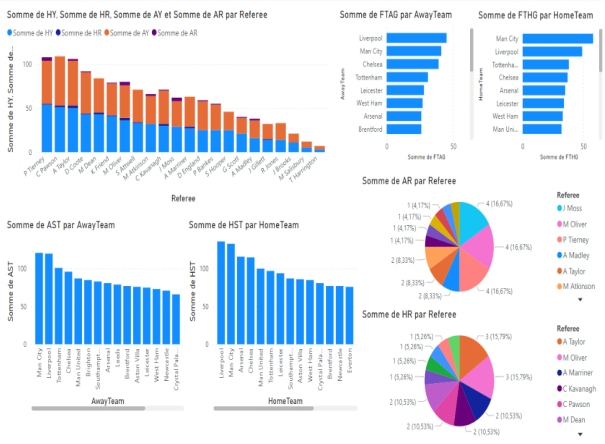
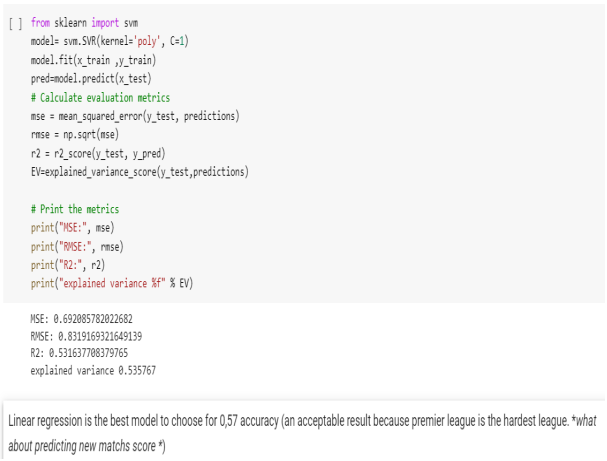


Figure 5: Visualisation sur PowerBi

- Prise de décision : La dernière étape de l'informatique décisionnelle consiste à utiliser les informations collectées et analysées pour prendre des décisions éclairées. Cette étape peut inclure la définition d'objectifs clairs, la création de plans d'action, la mise en œuvre de stratégies, etc.



```
[ ] from sklearn import svm
    model= svm.SVR(kernel='poly', C=1)
    model.fit(x_train ,y_train)
    pred=model.predict(x_test)
    # Calculate evaluation metrics
    mse = mean_squared_error(y_test, predictions)
    rmse = np.sqrt(mse)
    r2 = r2_score(y_test, y_pred)
    EV=explained_variance_score(y_test,predictions)

    # Print the metrics
    print("MSE:", mse)
    print("RMSE:", rmse)
    print("R2:", r2)
    print("explained variance %f" % EV)

    MSE: 0.692085782022682
    RMSE: 0.8319169321649139
    R2: 0.531637708379765
    explained variance 0.535767
```

Linear regression is the best model to choose for 0,57 accuracy (an acceptable result because premier league is the hardest league. *what about predicting new matchs score *)

Figure 6: Prédiction par le modèle de Machine Learning

3.5 Outils

Il existe plusieurs outils de BI (Business Intelligence) qui peuvent aider les entreprises à collecter, analyser et présenter des données pour faciliter la prise de décisions éclairées. On a choisi de travailler avec Talend

Talend est une plateforme logicielle d'intégration de données open-source. Elle est conçue pour aider les entreprises à intégrer, transformer, nettoyer et gérer des données provenant de différentes sources et destinées à différentes destinations.
La plateforme Talend comprend une variété d'outils et de fonctionnalités pour aider les entreprises à gérer leurs données, notamment :

- L'intégration de données : Talend offre une suite complète d'outils pour l'intégration de données, y compris la création de connexions à des sources de données, la conception de flux de données, la transformation de données, la gestion de la qualité des données et la gestion de la conformité réglementaire.
- L'intégration d'applications : Talend facilite l'intégration d'applications en utilisant des connecteurs prêts à l'emploi pour les applications courantes telles que Salesforce, SAP, Oracle, etc.
- Le Big Data : La plateforme Talend supporte également le traitement de Big Data, permettant aux entreprises d'ingérer, de transformer et de traiter des données massives provenant de sources telles que Hadoop, Spark, NoSQL et Cloud.
- La gestion de la qualité des données : Talend offre des outils pour la gestion de la qualité des données, permettant aux entreprises de nettoyer et d'enrichir leurs données pour garantir leur exactitude et leur cohérence.
- L'orchestration des données : Talend permet également de planifier et d'orchestrer des tâches de traitement de données pour automatiser les flux de travail de traitement de données. Talend est

  utilisé par de nombreuses entreprises pour améliorer la qualité et la gestion de leurs données, faciliter l'intégration de leurs applications, et améliorer la prise de décisions basées sur des données précises et actualisées.

Figure 7: Logo Talend

3.6 Schéma dimensionnel

Un schéma dimensionnel est un modèle de conception de base de données pour un entrepôt de données (datawarehouse) qui organise les données en fonction de leur signification et de leur pertinence pour les utilisateurs finaux.
Le schéma dimensionnel se compose de deux types de tables: les tables de faits (fact tables) et les tables de dimension (dimension tables).

- Les tables de faits contiennent les mesures numériques, telles que les ventes, les quantités, les marges bénéficiaires, etc., qui représentent l'activité de l'entreprise. Elles sont généralement grandes et contiennent beaucoup de données.
- Les tables de dimension contiennent des informations textuelles qui décrivent les données dans les tables de faits. Les dimensions sont souvent des listes hiérarchiques, telles que le temps, le produit, le client, le lieu, etc. Les tables de dimension sont généralement plus petites que les tables de faits.

Dans un schéma dimensionnel, les tables de dimension sont reliées aux tables de faits par des clés étrangères. Cette structure permet aux utilisateurs de naviguer facilement dans les données en utilisant les dimensions pour filtrer, trier et agréger les mesures numériques.
Le schéma dimensionnel est un choix courant pour les entrepôts de données car il est facile à comprendre et à utiliser pour les utilisateurs finaux, et il peut également améliorer les performances de requête en permettant des agrégations rapides et efficaces des données.

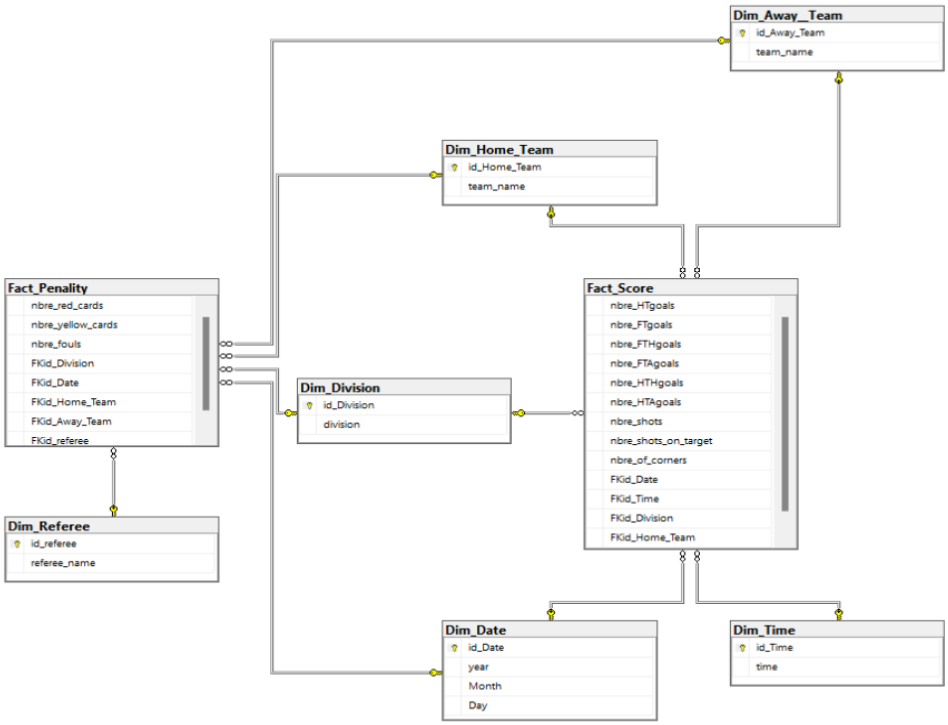


Figure 8: Schéma conceptuel du DataWarehouse

4 Machine Learning

4.1 Introduction

L'apprentissage automatique (ou machine learning en anglais) est une branche de l'intelligence artificielle qui permet à des systèmes informatiques d'apprendre à partir de données, sans avoir été explicitement programmés pour accomplir une tâche spécifique. Cette approche permet d'automatiser des tâches complexes qui seraient difficiles ou impossibles à réaliser avec des algorithmes classiques.

4.2 Bibliothèques utilisés

Python est l'un des langages de programmation les plus populaires pour les projets de machine learning. Il est connu pour sa simplicité, sa polyvalence et sa grande communauté de développeurs qui ont créé de nombreuses bibliothèques pour faciliter les tâches courantes de la science des données et de l'apprentissage automatique.
Dans ce rapport, nous présenterons quelques-unes des bibliothèques Python les plus couramment utilisées dans notre projet de machine learning. Ces bibliothèques ont été sélectionnées en fonction de leur popularité, de leur efficacité et de leur facilité d'utilisation.

- NumPy : une bibliothèque pour le calcul scientifique qui fournit des structures de données de base pour le traitement des tableaux multidimensionnels.

- Pandas : une bibliothèque pour la manipulation de données en tables, offrant des outils pour lire et écrire des fichiers de données, manipuler des données en tableaux, effectuer des opérations de traitement de données complexes, etc.

- Matplotlib : une bibliothèque pour la création de graphiques et de visualisations de données en 2D.

- Seaborn : une bibliothèque pour la visualisation de données statistiques, offrant des fonctionnalités pour la création de graphiques en 2D plus avancés.

- Scikit-learn : une bibliothèque pour l'apprentissage automatique, proposant une gamme d'algorithmes pour la classification, la régression, le clustering, etc.

- SciPy : SciPy est une bibliothèque Python utilisée pour le calcul scientifique et technique.

4.3 Prétraitement des données

Le prétraitement des données (ou data preprocessing en anglais) est une étape cruciale dans le processus de l'apprentissage automatique. Il s'agit de nettoyer, de transformer et de préparer les données brutes afin de les rendre utilisables pour les algorithmes d'apprentissage automatique.
Le prétraitement des données vise à améliorer la qualité des données d'entrée pour les algorithmes d'apprentissage automatique, en éliminant les problèmes potentiels tels que les biais, les valeurs extrêmes ou les données manquantes, qui peuvent affecter négativement les performances des modèles de prédiction.
En somme, le prétraitement des données est une étape clé de l'apprentissage automatique qui permet d'optimiser les performances des modèles en améliorant la qualité et la pertinence des données d'entrée.

```
[ ] # Deleting non fruitful features
    df_18= df_18[['Div', 'Date', 'HomeTeam', 'AwayTeam', 'FTHG', 'FTAG', 'FTR', 'HTHG',
          'HTAG', 'HTR', 'Referee', 'HS', 'AS', 'HST', 'AST', 'HF', 'AF', 'HC',
          'AC', 'HY', 'AY', 'HR', 'AR']]
    df_19= df_19[['Div', 'Date', 'HomeTeam', 'AwayTeam', 'FTHG', 'FTAG', 'FTR', 'HTHG',
          'HTAG', 'HTR', 'Referee', 'HS', 'AS', 'HST', 'AST', 'HF', 'AF', 'HC',
          'AC', 'HY', 'AY', 'HR', 'AR']]
    df_20= df_20[['Div', 'Date', 'HomeTeam', 'AwayTeam', 'FTHG', 'FTAG', 'FTR', 'HTHG',
          'HTAG', 'HTR', 'Referee', 'HS', 'AS', 'HST', 'AST', 'HF', 'AF', 'HC',
          'AC', 'HY', 'AY', 'HR', 'AR']]
    df_21= df_21[['Div', 'Date', 'HomeTeam', 'AwayTeam', 'FTHG', 'FTAG', 'FTR', 'HTHG',
          'HTAG', 'HTR', 'Referee', 'HS', 'AS', 'HST', 'AST', 'HF', 'AF', 'HC',
          'AC', 'HY', 'AY', 'HR', 'AR']]
    df_22= df_22[['Div', 'Date', 'HomeTeam', 'AwayTeam', 'FTHG', 'FTAG', 'FTR', 'HTHG',
          'HTAG', 'HTR', 'Referee', 'HS', 'AS', 'HST', 'AST', 'HF', 'AF', 'HC',
          'AC', 'HY', 'AY', 'HR', 'AR']]
    df= df[['Div', 'Date', 'HomeTeam', 'AwayTeam', 'FTHG', 'FTAG', 'FTR', 'HTHG',
          'HTAG', 'HTR', 'Referee', 'HS', 'AS', 'HST', 'AST', 'HF', 'AF', 'HC',
          'AC', 'HY', 'AY', 'HR', 'AR']]
```

Figure 9: exemple de prétraitement

```
[ ] df.dropna()

[ ] num_of_nan = df.isnull().sum()
    num_of_nan

    Div          0
    Date         0
    HomeTeam     0
    AwayTeam     0
    FTHG         0
    FTAG         0
    FTR          0
    HTHG         1
    HTAG         1
    HTR          1
    Referee      1
    HS           1
    AS           1
    HST          1
    AST          1
    HF           1
    AF           1
    HC           1
    AC           1
    HY           1
    AY           1
    HR           1
    AR           1
    dtype: int64
```

Figure 10: exemple de prétraitement

```
[ ] mean_values = df.mean()

    df = df.fillna(mean_values)
```

Figure 11: exemple de prétraitement

4.4 Modélisation

La modélisation est une étape centrale dans le processus de l'apprentissage automatique. Elle consiste à choisir et à entraîner un modèle statistique à partir des données prétraitées pour accomplir une tâche spécifique, telle que la classification, la régression, ou la prédiction.
Le choix du modèle dépend de la nature de la tâche à accomplir et des caractéristiques des données d'entrée. Il existe une variété de modèles d'apprentissage automatique, allant des modèles simples tels que la régression linéaire, aux modèles plus complexes tels que les réseaux de neurones.

```
from sklearn.linear_model import LinearRegression
# Create a linear regression model
model = LinearRegression()

# Fit the model to the data
model.fit(x_train , y_train)
```

Figure 12: entrainement du model 1

```
model = RandomForestRegressor(n_estimators=100,criterion='squared_error')
model.fit(x_train,y_train)
```

Figure 13: entrainement du model 2

```
import xgboost as xgb
from xgboost import XGBRegressor

my_model = XGBRegressor()
# Add silent=True to avoid printing out updates with each cycle
my_model.fit(x_train, y_train, verbose=False)
model.fit(x_train,y_train)
```

Figure 14: entrainement du model 3

```
from sklearn import svm
from sklearn.metrics import accuracy_score
model= svm.SVR(kernel='linear', C=1)
model.fit(x_train ,y_train )
```

Figure 15: entrainement du model 4

4.5 Prédiction

La prédiction est l'une des applications principales de l'apprentissage automatique. Elle consiste à utiliser un modèle entraîné sur des données pour prédire une sortie ou une valeur future à partir de données d'entrée.
Une fois que le modèle a été entraîné et évalué, il peut être utilisé pour faire des prédictions sur de nouvelles données. Les données d'entrée sont fournies au modèle, qui utilise les relations apprises pendant l'entraînement pour prédire la sortie souhaitée.

```
y_pred=model.predict(x_test)
```

Figure 16: Code de la prédiction

4.6 Résultats et analyse

La section des résultats est l'une des parties les plus importantes d'un rapport de machine learning. Elle présente les résultats obtenus après l'entraînement et l'évaluation du modèle sur des données de test. Cette section comprend généralement une analyse détaillée des performances du modèle, y compris des mesures telles que la précision, le rappel, la F-mesure, l'aire sous la courbe ROC, l'erreur quadratique

moyenne, etc. Ces mesures permettent d'évaluer la capacité du modèle à généraliser à de nouvelles données et à faire des prédictions précises.

Model	Evaluation
model = LinearRegression()	MSE: 0.6310802148283544 RMSE: 0.7944055732611361 R2: 0.5765797372848642 explained variance 0.576747
model = RandomForestRegressor(n_estimators=100,criterion='squared_error')	MSE: 0.6980633702266905 RMSE: 0.8355018672789969 R2: 0.531637708379765 explained variance 0.531780
model = XGBRegressor()	MSE: 0.692085782022682 RMSE: 0.8319169321649139 R2: 0.531637708379765 explained variance 0.535767
model= svm.SVR(kernel='rbf', C=1)	MSE: 0.692085782022682 RMSE: 0.8319169321649139 R2: 0.531637708379765 explained variance 0.535767

Figure 17: Evaluation des modèles

4.7 Conclusion

Dans cette partie on a présenté la partie Machine Learning de notre projet, qui servira come "input" pour la partie déploiement.

5 Déploiement

5.1 Introduction

La partie Déploiement est une étape importante dans la mise en place d'un système d'Intelligence Artificielle (IA) et de Business Intelligence (BI). Elle concerne la mise en production des modèles de Machine Learning et des tableaux de bord de BI pour permettre aux utilisateurs finaux d'interagir avec les données et de prendre des décisions informées.

5.2 Préparation de l'environnement de production

La première étape du déploiement consiste à préparer l'environnement de production. Cette étape comprend l'installation et la configuration des bibliothèques Python, la création de la base de données et la configuration des paramètres de l'application Django.
Framework :

Django : Django est un framework web gratuit et open-source, basé sur Python, qui suit le modèle architectural modèle-vues. Il est maintenu par la Django Software Foundation

5.3 Développement de l'application Django

La deuxième étape du déploiement consiste à développer l'application Django pour mettre en production les modèles de Machine Learning et les tableaux de bord de BI. Cette étape comprend la création des vues

pour les tableaux de bord, la création des modèles pour la base de données et la configuration des URLs pour l'application.

5.4 Avantages de l'utilisation de Django

L'utilisation de Django pour le déploiement des modèles de Machine Learning et des tableaux de bord de BI présente de nombreux avantages, notamment une réduction du temps de développement, une simplicité d'utilisation et une flexibilité pour la création d'applications Web.

5.5 Conclusion

En conclusion, la partie Déploiement de ce rapport a montré que la mise en production des modèles de Machine Learning et des tableaux de bord de BI est une étape critique dans la mise en place d'un système d'IA et de BI. Le déploiement nécessite une préparation minutieuse de l'environnement de production, ainsi qu'une gestion et une surveillance régulières pour assurer un fonctionnement optimal et une maintenance efficace.

6 Conclusion Générale

La partie Machine Learning de ce rapport et la partie de BI on montré l'importance de l'analyse de données pour aider les entreprises à prendre des décisions informées. Les techniques de Machine Learning ont été appliquées à un ensemble de données de Foot pour prédire les variables cibles (Score/winner)et identifier les facteurs les plus importants qui influencent ces variables.