EIGSI
ÉCOLE D'INGÉNIEURS
LA ROCHELLE • CASABLANCA

PROJET PYTHON

Rapport d'Analyse de Données

Membre du groupe : Mathis Ehkirch – Simon Allam
Date : 27/04/2024

StarCraft 2 Player Prediction Challenge 2019

Can you predict who is playing a game?

1 Objectif

Le projet vise à développer un modèle de prédiction des joueurs de StarCraft 2 en se basant sur leurs inputs lors de différentes parties. L'objectif est de créer un système capable d'identifier le joueur à partir de ses actions en jeu.

2 Base de données

Nous avons utilisé le dataset fourni par Kaggle, comprenant des datas comportementales étiquetées pour l'entraînement (TRAIN) et des datas non étiquetées pour les tests (TEST). Les données comportent diverses informations sur les actions des joueurs pendant les parties de StarCraft 2 (la race choisi et les touches rentrées).
Lien: https://www.kaggle.com/competitions/starcraft-2-player-prediction-challenge-2019/data

3 Prétraitement des données

Les données ont nécessité un prétraitement. En effet, lorsque nous avons ajouté nos données, cela nous les à charger sous la forme d'un dataframe ou chaque ligne contenait une liste. Ainsi, nous avons dû créer les colonnes nécessaires (joueur, race, input). De plus nous avons utilisé une variante du OneHotEncoder pour pouvoir créer une colonne par type de touche entré avec un compte du nombre de touche. Suite à cela, nous avons dû encoder manuellement la colonne race mais nous avons dû également supprimer les quelques valeurs nul de notre dataset. Enfin, nous avons égalisé le nombre de colonne entre notre X_train et notre X_test pour pouvoir faire tourner nos models.

4 Choix et entraînement des modèles

Une fois tous les problèmes résolus concernant les données, nous avons choisi d'entraîner différents modèles pour prédire les joueurs de StarCraft 2. Nous avons commencé par exécuter un modèle de KMeans sur nos données dans le but de faire une pré-sélection des joueurs. Le résultat de cette classification a été ajouté dans une nouvelle colonne pour améliorer la performance de nos modèles. Suite à cela, nous avons comparé différents modèles tels que le KNN et le RandomForest pour sélectionner les plus adaptés à notre tâche. Nous avons choisi les modèles KNN et RandomForest, car le modèle KNN est efficace pour détecter des structures complexes dans les données, tandis que le modèle RandomForest est robuste aux valeurs aberrantes et peut gérer des ensembles de données de grande dimension.

Nous avons ensuite pu entraîner nos modèles grâce aux prétraitements effectués sur les données. Nous avons défini la métrique d'évaluation pour choisir le meilleur modèle, en l'occurrence le F1-score. Le F1-score, couramment utilisé en recherche d'information, mesure l'exactitude en utilisant les statistiques de précision (p) et de rappel (r). La précision est le rapport des vrais positifs (tp) à tous les positifs prédits (tp + fp). Le rappel est le rapport des vrais positifs à tous les positifs réels (tp + fn). Le F1-score est donné par :
$F1=2p\cdot rp+r$ où $p=tptp+fp$ et $r=tptp+fn$.
La métrique F1 équilibre le rappel et la précision de manière égale, et un bon algorithme de recherche maximisera à la fois la précision et le rappel simultanément. Ainsi, une performance modérément bonne sur les deux sera favorisée par rapport à une performance extrêmement bonne sur l'un et une mauvaise performance sur l'autre.

5 Optimisation

Nous avons pu optimiser notre modèle grâce à l'utilisation du randomsearch. Au début du projet nous voulions utiliser le gridsearch pour améliorer les paramètres de notre modèle mais cela était extrêmement long, ainsi nous avons opter pour un randomsearch qui grâce à l'amélioration de certains paramètres, nous a permis de passer d'un score de 0.55 à un score de 0.7 les hyperparamètres optimisés sont le nombre d'estimateurs (n_estimators), la profondeur maximale de l'arbre (max_depth), le nombre minimal d'échantillons requis pour diviser un nœud (min_samples_split), et le nombre minimal d'échantillons requis pour être à un nœud terminal (min_samples_leaf). En utilisant une métrique F1-Score pour évaluer les performances du modèle, nous effectuons une recherche aléatoire sur l'espace des hyperparamètres avec 25 itérations et une validation croisée à 5 plis.

6 Conclusion et Perspectives

Pour conclure, notre modèle c'est révélé efficace pour automatiser la reconnaissance des joueurs dans le jeu StarCraft 2. Grâce à notre travail, nous avons réussi à développer un modèle capable de prédire les joueurs avec une précision satisfaisante, comme en témoigne notre score de 0.7 sur l'échantillon de test fourni par Kaggle. L'utilisation de différentes techniques de prétraitement des données et de modélisation a permis d'obtenir ces résultats prometteurs.

Cependant, il est important de noter que notre modèle peut encore être amélioré. Des axes d'améliorations pourraient inclure l'utilisation de feature engineering plus avancées pour capturer des informations plus subtiles sur le comportement des joueurs. De plus, l'utilisation de modèles d'apprentissage profond tels que les réseaux neuronaux pourraient également être envisagée.
En outre, une analyse plus approfondie des erreurs de prédiction du modèle pourrait fournir des informations précieuses sur les cas difficiles. Cela pourrait conduire à des ajustements supplémentaires des hyperparamètres ou à l'ajout de nouvelles fonctionnalités pour améliorer la performance globale du modèle.