Raport

Soccer Players Clustering

Michał Iwaniuk, Anna Ostrowska

1. Wstęp

Celem projektu było zapoznanie się z pojęciem klasteryzacji danych. Projekt był podzielony na 3 kamienie milowe. KM1 – EDA, zapoznanie się ze zbiorem danych oraz zrozumienie jego cech. KM2 – feature engineering, odpowiednie przetworzenie zbioru danych, wyczyszczenie oraz wybór cech. KM3 – stworzenie finalnych modeli klasteryzujących nasze dane i ocenienie ich jakości, wyciągnięcie wniosków.

Przez cały okres tworzenia projektu walidowaliśmy projekt naszych kolegów oraz byliśmy przez kolejnych walidowani. Zadaniem walidatorów było ciągłe sprawdzanie czy np. nie został popełniony gdzieś błąd logiczny w trakcie procesu oraz ewentualna pomoc w całym zagadnieniu

1. EDA

Nasze dane pochodzą z meczów piłki nożnej i zawierają szczegółowe statystyki meczowe dla każdego gracza. Dataset składa się z 76185 obserwacji oraz 127 kolumn.

Przykładowe znaczenia kolumn: awayScore (wynik drużyny gracza w tym meczu) homeScore (wynik drużyny przeciwnej) homeTeamName (nazwa drużyny przeciwnej) home/awayTeamTacticalSchema (schemat taktyczny drużyn) matchWeek (numer kolejki) awayTeamName (nazwa drużyny gracza) matchDate (data meczu) playerName (imię i nazwisko gracza) championshipName (nazwa rozgrywek) age (wiek gracza) assists (asysty w tym meczu) xg (oczekiwane bramki) xa (oczekiwane asysty) touches in different areas (dotknięcia piłki w różnych obszarach)

Już na tym etapie usunelismy kolumnę playerName ponieważ uznaliśmy ze to jak się nazywa dany gracz jest nieistotne.

Wiekszosc zmiennych jest numerycznych, zaledwie 8 kategorycznych. Dane zawierają dużo wartości NA bo pojawiają się w az 100 kolumnach.

W danych mamy bardzo dużo kolumn mocno skorelowanych (tj. abs(corr) >= 0.5) co widać nizej. Jest to uzyteczna wiedza w kontekście przyszłego usuwania niepotrzebnych kolumn.

Obraz zawierający tekst, zrzut ekranu, linia, numer

Opis wygenerowany automatycznie

Przebadalismy zmienne kategoryczne i okazało się ze jest dosyc sporo wartości unikalnych. Prezentowaly się one tak:

Obraz zawierający tekst, Czcionka, zrzut ekranu, biały

Opis wygenerowany automatycznie

Okazalo się ze dane pochodzą z 5 najlepszych lig europejskich oraz ze statystyki były prowadzone w latach 2019-2021. Na tym etapie postanowiliśmy usunąć kolumnę team\_name ponieważ zawiera dokładnie te same dane co kolumna homeTeamName

Nastepnie przebadaliśmy zmienne numeryczne. Oto przykladowe rozklady zmiennych:

Obraz zawierający diagram, zrzut ekranu, Prostokąt, Plan

Opis wygenerowany automatycznieObraz zawierający diagram, linia, tekst, Wykres

Opis wygenerowany automatycznie

Pomimo ze po boxplotach można by się spodziewac ze w danych mamy duzo outlierow to jednak po zrobieniu histogramow okazuje się ze wiekszosc cech ma rozklad zblizony do geometrycznego i de facto nie ma tu outlierow, po prostu dane sa „rozciagniete”.

1. Feature engineering

Zabralismy się za obrobke danych najpierw usuwając wartości NA.

Sprawdzilismy ile wartości NA zawierają poszczególne kolumny i wygladalo to tak:

Obraz zawierający tekst, zrzut ekranu, Czcionka

Opis wygenerowany automatycznie

Tutaj postanowiliśmy usunąć wszystkie kolumny zawierające „pct”, ponieważ jest to skrót od "percentage" (procent). Przykładowo, kolumna „dribbles\_completed\_pct” oznacza procent udanych dryblingów. Informacje zawarte w tych kolumnach są już obecne w innych kolumnach, takich jak „dribbles” i „dribbles\_completed”. Usunięcie tych kolumn jest uzasadnione, ponieważ pozwala nam także pozbyć się wartości NA.

Następnie zdecydowaliśmy się usunąć kolumny z dużą korelacją (tj. korelacja >= 0.7) oraz te, których informacje są zawarte w innych kolumnach. Na przykład zostawiliśmy kolumnę „passes”, ale usunęliśmy „passes\_total\_distance”, ponieważ jest ona silnie skorelowana z liczbą podań. Wiele kolumn usuwaliśmy wybiórczo: najpierw wypisywaliśmy kolumny silnie skorelowane z kolumną, którą chcieliśmy zachować, a następnie wybieraliśmy spośród nich te, które uznaliśmy za nieistotne.

Następnie usunęliśmy 50 wierszy (meczów), w których nie były prowadzone szczegółowe statystyki i zawierały one tylko kilka ogólnych informacji o meczu, co pozwoliło nam również pozbyć się wartości NA z 44 kolumn. Pozostałe wartości NA, które występowały już tylko w 5 kolumnach numerycznych i 1 kategorycznej, uzupełniliśmy medianą (dla kolumn numerycznych) oraz wartością „unknown” (dla kolumny kategorycznej).

Pozbyliśmy się już niepotrzebnych kolumn i wartości NA, więc pozostało zakodować zmienne. Przypomnijmy, jak wygląda rozkład zmiennych kategorycznych: wygląda rozklad zmiennych kategorycznych:

Obraz zawierający tekst, Czcionka, zrzut ekranu

Opis wygenerowany automatycznie

Formacje obu drużyn (awayTeamTacticalSchema, homeTeamTacticalSchema) postanowiliśmy zakodować w taki sposób, aby zapisać liczbę obrońców, pomocników i napastników w trzech osobnych kolumnach. Na przykład, w formacji (4-4-2) mamy 4 obrońców, 4 pomocników i 2 napastników, natomiast w formacji (4-2-3-1) mamy 4 obrońców, 5 pomocników (suma środkowych wartości) oraz 1 napastnika. Kolumny awayTeamName oraz homeTeamName uznaliśmy za nieistotne, więc je usunęliśmy. Kolumnę matchDate podzieliliśmy na dwie kolumny: „year” oraz „month” danego meczu, przy czym wartości „month” przekształciliśmy na wartości cykliczne. Zastosowaliśmy OneHotEncoding dla kolumny „championship\_name”.

Następnie zastosowaliśmy automatyczną detekcję outlierow przy użyciu metody kNN z biblioteki pyod, na wszelki wypadek, gdyby w zbiorze danych pojawiły się jakiekolwiek niespodziewane odstępstwa. Usnęliśmy 4% obserwacji. Jak się okazało nie zaburzyło to nam zbioru i nie usunęło rzadkich aczkolwiek wartościowych informacji takich jak np. duża liczba goli w meczu.

Następnie zastosowaliśmy skalowanie MinMax na naszych danych, ponieważ dane w ogóle nie były zbliżone do rozkładu normalnego.

W procesie wstępnego modelowania okazało się, że najbardziej optymalna liczba klastrów dla metody KMeans wynosi 5. Okazało się, że ta liczba wynika z faktu, że klastry zostały ustalone według 5 lig, w których prowadzone są rozgrywki. Można uznać to za wstępny sukces.

1. Final models