Dokumentacja projektu

1. Cel i założenia projektu

Celem projektu było stworzenie systemu rozmytego, który będzie przewidywać wynik meczu piłki nożnej w oparciu o statystyki z kilku ostatnich meczów. Docelowo system ma przewidywać jedną z trzech klas: wygrana, przegrana lub remis dla drużyny grającej u siebie. Główną metryką służącą do oceny modelu jest accuracy - im większy wynik na zbiorze testowym tym system lepiej przewiduje wynik spotkania. Ponadto zakłada się stworzenie kilku modeli rozmytych różnymi metodami i na różnych zbiorach danych oraz porównanie ich skuteczności.

1. Zbiory danych

Dane pochodzą z sofascore oraz fbref. Zawierają różnorodne statystyki dotyczące każdego spotkania np. ilość akcji zakończonych golem, celność podań czy skuteczność obron bramkarza. Dane zostały stworzone zarówno dla drużyny grającej u siebie, jak i wyjazdowej. Zliczone zostały ich sumy oraz średnie z kilku ostatnich meczów (maksymalnie 15) a także wyliczono dodatkowe statystyki tj. momentum. Końcowo do stworzenia modeli rozmytych wykorzystano dane z Premier League oraz LaLiga dotyczące sezonów od 2019 do 2023. Przygotowane dane nie są dostępne publicznie.

1. Algorytmy i metody

Pierwszym krokiem do stworzenia modeli rozmytych było przygotowanie danych oraz dokonanie analizy korelacji w celu znalezienia najbardziej istotnych cech. Użyto do tego macierzy korelacji oraz feature importance z klasyfikatora XGBoost. Dla modelu wykorzystującego dane bukmacherskie wybrano łącznie 28 cech, a dla modeli opartych tylko na danych statystycznych wybrano 23 cechy. Mając gotowe dane można przystąpić do budowania i testowania modeli. W ramach projektu stworzono 5 modeli:

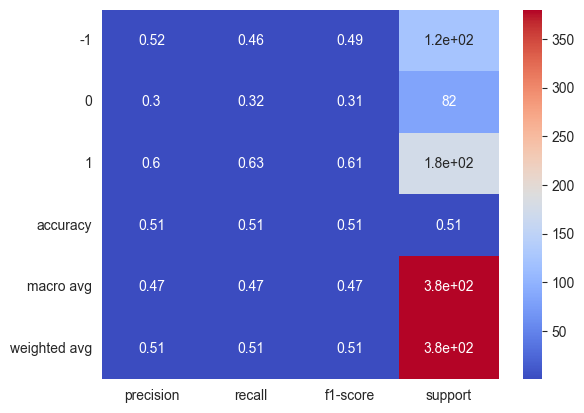
* Model Premier League oparty na statystykach oraz wiedzy eksperckiej
* Model LaLiga oparty na statystykach oraz wiedzy eksperckiej
* Model Premier League oparty na statystykach i danych bukmacherskich oraz wiedzy eksperckiej
* Model LaLiga oparty na statystykach oraz bibliotece PyFume
* Model Premier League oparty na statystykach oraz bibliotece PyFume

Modele oparte na wiedzy eksperckiej zostały zbudowane przy użyciu biblioteki Simpful. W ramach tych modeli stworzono funkcje: definiujące zmienne lingwistyczne na podstawie kwantyli, tworzące klasyfikatory oraz ustalające reguły eksperckie.

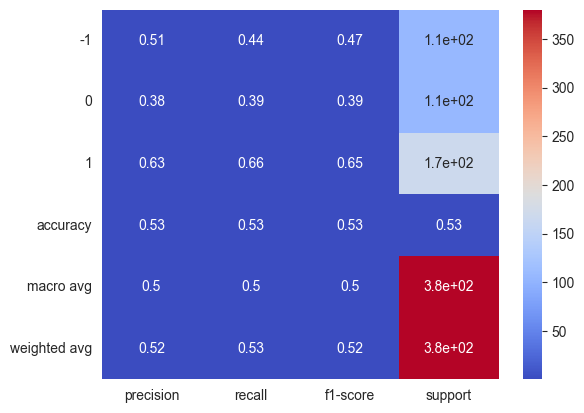
Modele oparte na bibliotece PyFume zostały automatycznie skonstruowane przy użyciu dostępnych funkcji. Biblioteka utworzyła zmienne lingwistyczne oraz reguły na podstawie danych uczących. Wszystkie modele zostały następnie przetestowane na odpowiednich zbiorach testowych oraz porównano ich wyniki.

1. Analiza wyników

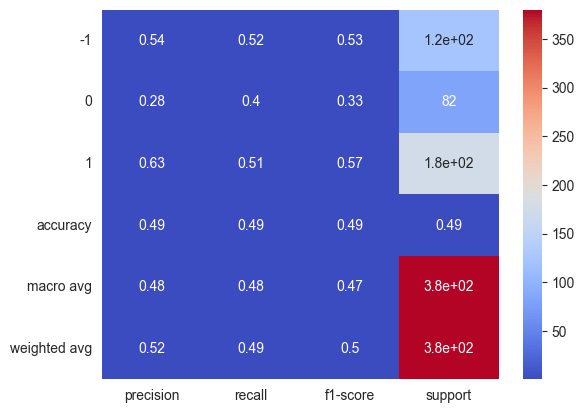
Poniższe tabele przedstawiają wyniki ewaluacji każdego z pięciu modeli



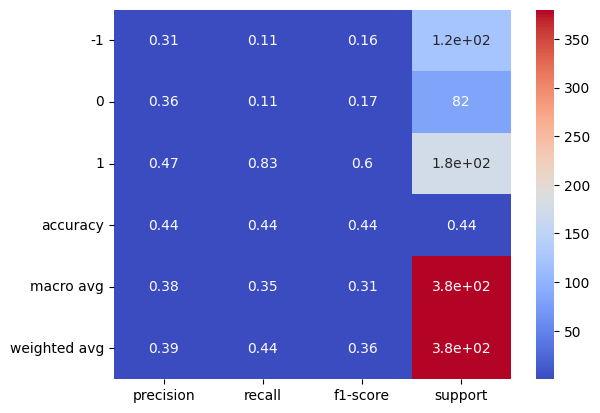
Model Premier League oparty na statystykach oraz wiedzy eksperckiej



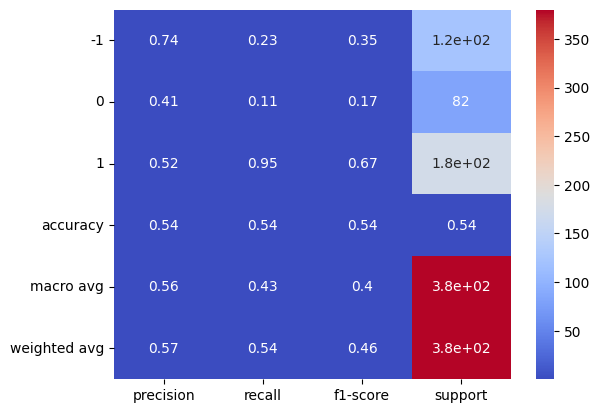
Model LaLiga oparty na statystykach oraz wiedzy eksperckiej



Model Premier League oparty na statystykach i danych bukmacherskich oraz wiedzy eksperckiej



Model LaLiga oparty na statystykach oraz bibliotece PyFume



Model Premier League oparty na statystykach oraz bibliotece PyFume

Wyniki ewaluacji wskazują, że średni wynik accuracy wszystkich modeli wynosi ok. 50% dla problemu klasyfikacji z trzema klasami. Najlepiej wypadł model dla Premier League z biblioteką PyFume z wynikiem 54% a najgorzej model LaLiga z biblioteką PyFume z wynikiem 44%. Warto zauważyć, że modele eksperckie oparte tylko na statystykach wypadły lepiej niż model z danymi bukmacherskimi o odpowiednio 2 i 4 punkty procentowe. Dla Premier League najlepszy model osiągnął 54% a dla LaLiga 53% accuracy.

Wyniki ewaluacji nie są z pewnością w pełni satysfakcjonujące, ponieważ są dosyć niskie. Wynika to przede wszystkim ze specyfiki sportu, jakim jest piłka nożna. Dane statystyczne nie zawsze dobrze oddają to co się wydarzy na boisku. Służą one głównie do przewidywania szansy danej drużyny na zwycięstwo co nierzadko nie ma pokrycia z faktycznym wynikiem spotkania. Często można dostrzec, że któraś drużyna jest lepsza od drugiej, ale np. nie potrafią zakończyć akcji golem i przegrywają mecz po jedynej skutecznej akcji przeciwnika. Dodatkowym problemem jest również fakt, że na ogół trudno jest przewidywać remisy. Modele zazwyczaj przewidują wynik na korzyść jednej bądź drugiej drużyny, co można zauważyć również w powyższych modelach, gdzie wyniki precision, recall oraz f1-score są znacznie niższe dla remisów niż dla pozostałych klas.

1. Opis repozytorium

Repozytorium zawiera następujące pliki i foldery:

* fduk\_data\_description.md – zawiera opisy statystyk, głównie bukmacherskich
* fduk\_data\_description.txt – zawiera opisy statystyk, głównie bukmacherskich
* player\_statistics\_fbref.md – zawiera opisy statystyk ze strony fbef
* player\_statistics\_sofascore.md – zawiera opisy statystyk ze strony sofacore
* Przygotowanie danych.ipynb – zawiera kod dokonujący wstępnego przetwarzania danych
* Przykładowy system rozmyty – biblioteka simpful.ipynb – zawiera model rozmyty stworzony na potrzeby zaznajomienia się z biblioteką simpful
* requirements.txt – zawiera nazwy bibliotek potrzebnych do uruchomienia programów
* Simpful\_code\_la\_liga.py – kod wygenerowany przez bibliotekę pyFume tworzący model rozmyty na podstawie danych uczących z LaLiga
* Simpful\_code\_premier\_league.py – kod wygenerowany przez bibliotekę pyFume tworzący model rozmyty na podstawie danych uczących z Premier League
* System rozmyty – bez danych bukmacherskich.ipynb – zawiera program, który tworzy oraz ewaluuje model rozmyty w oparciu o statystyki meczowe oraz wiedzę ekspercką
* System rozmyty – dane bukmacherskie.ipynb - zawiera program, który tworzy oraz ewaluuje model rozmyty w oparciu o statystyki meczowe, dane bukmacherskie oraz wiedzę ekspercką
* System rozmyty – PyFume – La Liga.ipynb - zawiera program, który tworzy oraz ewaluuje model rozmyty w oparciu o statystyki meczowe z LaLiga oraz bibliotekę PyFume
* System rozmyty – PyFume – Premier League.ipynb - zawiera program, który tworzy oraz ewaluuje model rozmyty w oparciu o statystyki meczowe z Premier League oraz bibliotekę PyFume