Sprawozdanie

Imię i nazwisko: Bartłomiej Ruszaj

Data wykonania ćwiczenia: 06.06.2023

1. Wprowadzenie

- Celem projektu jest znalezienie najlepszego klasyfikatora pozycji piłkarzy na podstawie ich statystyk meczowych z sezonu 2021-2022. W tym celu wykorzystano zbiór danych „2021-2022 Football Player Stats" zawierający ponad 2500 rekordów oraz 143 kolumny (cechy). Analizie poddano średnie statystki piłkarzy na mecz z pięciu najlepszych lig (Premier League, Ligue 1, Bundesliga, Serie A, La Liga).

- Spodziewane wyniki z przeprowadzonych badań nad klasyfikacją pozycji piłkarzy powinny wskazać klasyfikator na tyle dobry, że jego dokładność będzie wynosić powyżej 90%. Posiadana duża ilość danych oraz wiele różnych cech do analizy pozwala na zastosowanie różnorodnych algorytmów klasyfikacji. Dzięki różnorodności cech, jesteśmy w stanie wybrać te które wykazują największą korelację z wynikiem, lub wręcz przeciwnie, naiwnie przyjąć, że bierzemy pod uwagę wszystkie cechy.

- Podczas wykonywania ćwiczenia powinienem nauczyć się wielu ważnych umiejętności pozwalających na sprawne przetwarzanie danych „Big Data". Przede wszystkim, nauczę się jak przygotować dane i jak odpowiednio je przetworzyć, co umożliwi mi ich dalsze użytkowanie. Nauczę się, jak wybierać odpowiednie cechy do klasyfikacji, jak interpretować wyniki klasyfikacji i jak dokonywać optymalizacji modelu. Ostatecznie będę potrafił wybrać najlepszy klasyfikator.

2. Użyte narzędzia i ich zastosowanie

- Do przeprowadzenia klasyfikacji piłkarzy zostały wykorzystane następujące narzędzia i biblioteki dostępne w języku programowania python:

- *Pandas*: Do wczytania danych z pliku CSV i manipulacji nimi. Wstępna analiza i przetworzenie danych.
- *Seaborn i Matplotlib*: Do wizualizacji danych i generowania wykresów.
- *Scikit-learn*: Do standaryzacji danych, redukcji wymiarów za pomocą PCA oraz klasyfikacji za pomocą *RandomForestClassifier, KNeighborsClassifier, SVC.*
- *StandardScaler*: Do standaryzacji danych przed zastosowaniem algorytmu klasyfikacji.
- *PCA* (Principal Component Analysis): Do redukcji wymiarów danych.
- *RandomForestClassifier.feature_importances_:* Do uzyskania posortowanych cech, ze względu na wpływ na zdolność do klasyfikacji zmiennej docelowej.
- *Cross_val_score*: Do oceny jakości klasyfikatorów, na podstawie F1-Score.
- *Classification_report*: Do szczegółowej oceny najlepszego klasyfikatora.
- *Confusion_matrix:* Do oceny dokładności przeprowadzonej klasyfikacji.
- *Train_test_split*: Do podziału zbioru na treningowy i testowy.

3. Wyniki

- Schemat wykonywanych badań:

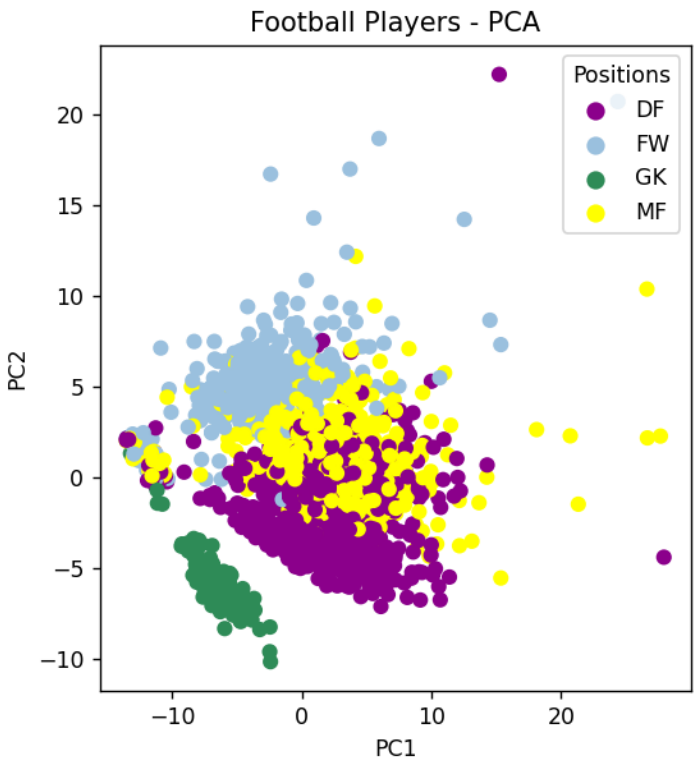
1. Wczytanie i przetworzenie zbioru danych
2. Wizualizacja problemu poprzez PCA(n=2)
3. Naiwna klasyfikacja na podstawie wszystkich cech
4. Klasyfikacja przy analizie PCA
5. Klasyfikacja przy użyciu RandomForestClassifier.feature_importances_
6. Wybór najlepszego klasyfikatora
7. Klasyfikacja pozycji piłkarzy, szczegółowa ocena klasyfikatora, macierz pomyłek

1. Wczytanie i przetworzenie zbioru danych:

```
Unikalne pozycje w danych:  ['DF' 'MF' 'FW' 'MFFW' 'FWMF' 'GK' 'DFMF' 'FWDF' 'MFDF' 'DFFW' 'GKMF']
Liczba rekordów:  2921
Wybrane pozycje:  ['GK', 'DF', 'MF', 'FW']
Liczba rekordów wybraniu pozycji: 2175
Czy w dataset są NaN:  False
Liczba wszystkich cech: 143
Nazwy usuniętych kolumn: {'Player', 'Comp', 'Nation', 'Pos', 'Squad'}
Liczba numerycznych cech: 138
```

Rysunek 1. Wczytanie i przetworzenie danych

2. Wizualizacja problemu poprzez PCA(n=2):



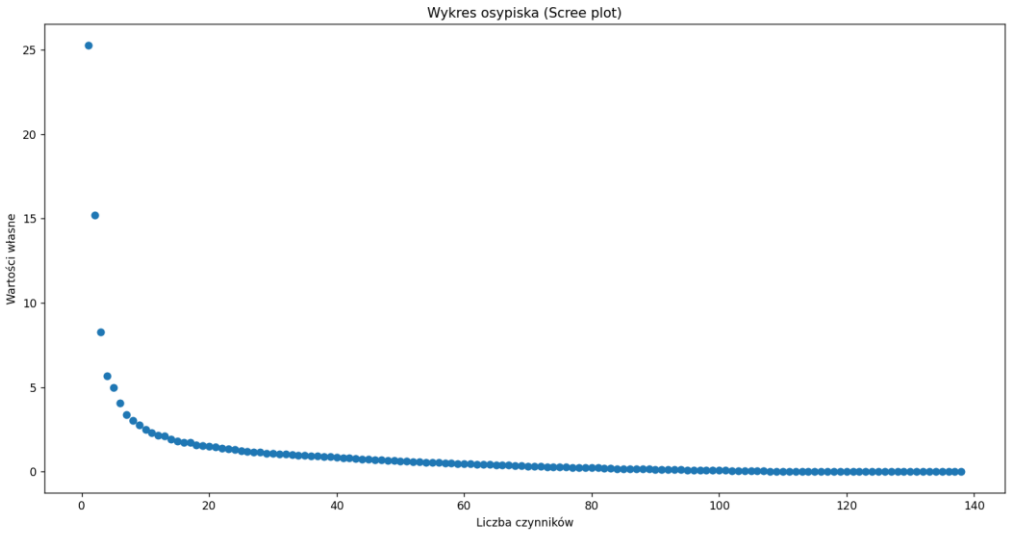
Rysunek 2. Wizualizacja PCA(n=2)

3. Naiwna klasyfikacja na podstawie wszystkich cech:

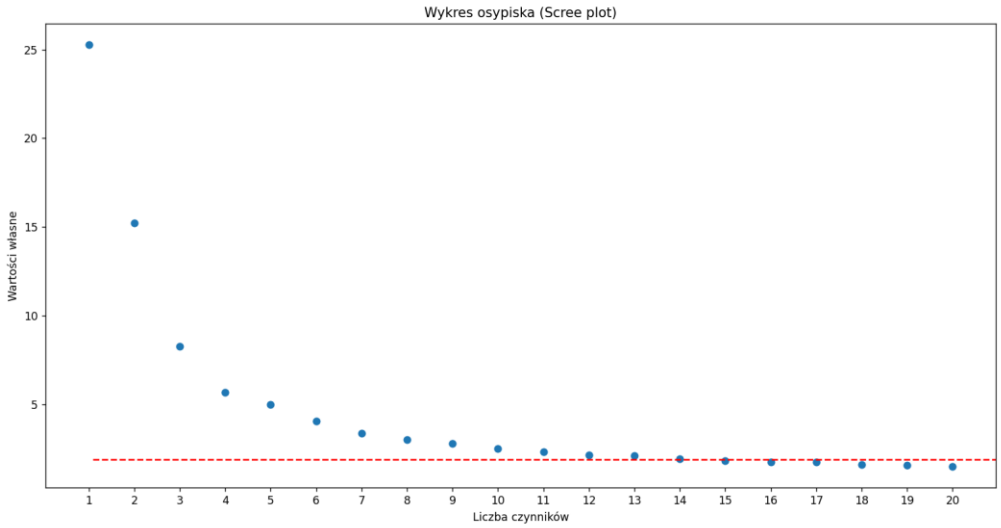
Klasyfikator	Cross_value_score(F1)	Czas
Random Forest	0.942	5.93 s
K Neighbors Classifier	0.924	0.31 s
Support Vector Classifier	0.930	0.64 s

Tabela 1. Ocena jakości klasyfikacji z użyciem wszystkich cech

4. Klasyfikacja przy analizie PCA:



Rysunek 3. Wykres Osypiska dla wszystkich cech



Rysunek 4. Wykres osypiska dla n=20 cech

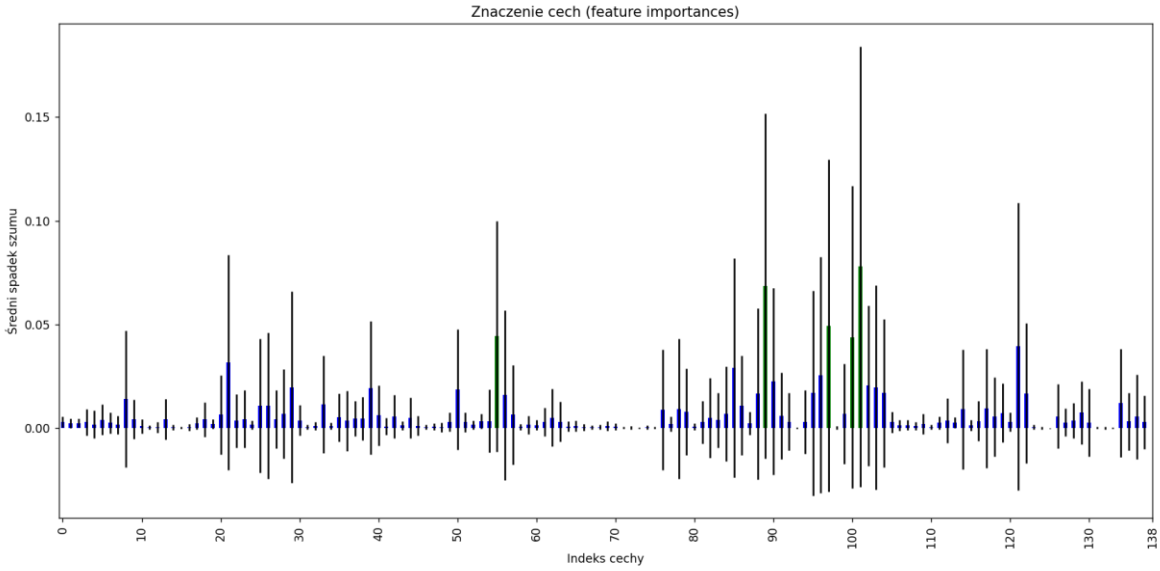
Klasyfikator	Cross_value_score(F1)	Czas
Random Forest	0.921	3.76 s
K Neighbors Classifier	0.926	0.20 s
Support Vector Classifier	0.931	0.38 s

Tabela 2. Ocena jakości klasyfikacji z użyciem PCA(n=13)

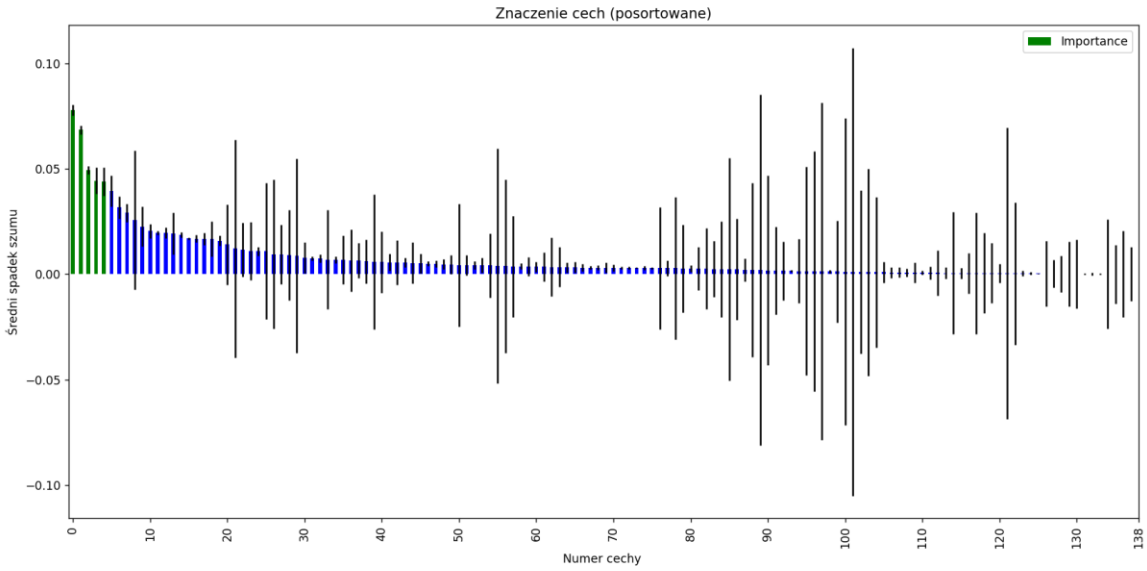
5. Klasyfikacja przy użyciu RandomForestClassifier.feature_importances_:

```
5 najważniejszych cech:
89     PresMid3rd
101     TouDef3rd
100     TouDefPen
97            Clr
55             TI
```

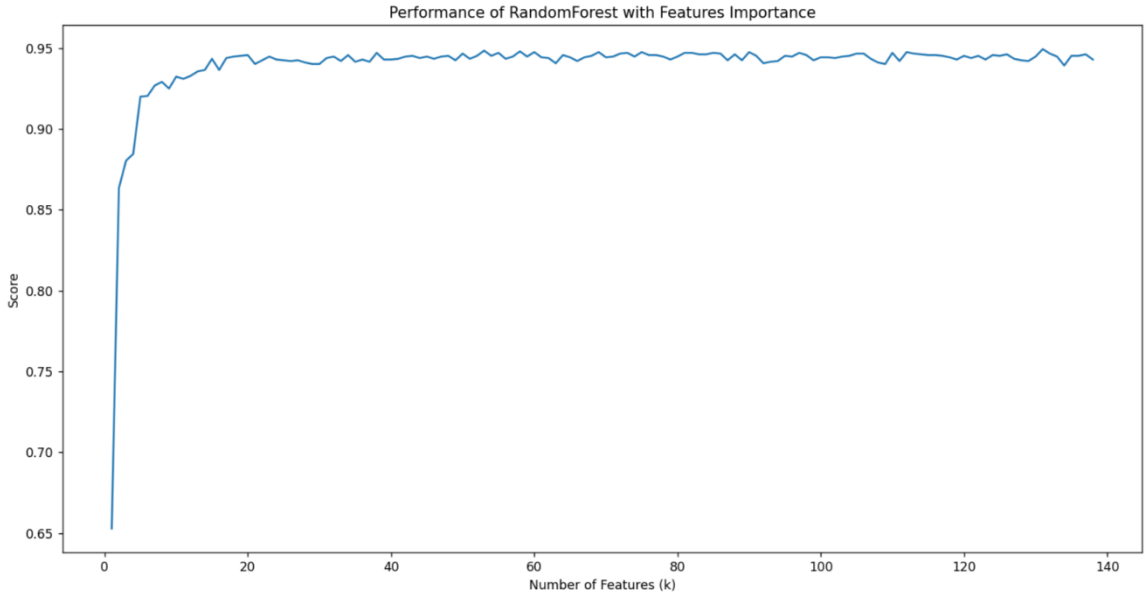
Rysunek 5. Pięć najważniejszych cech



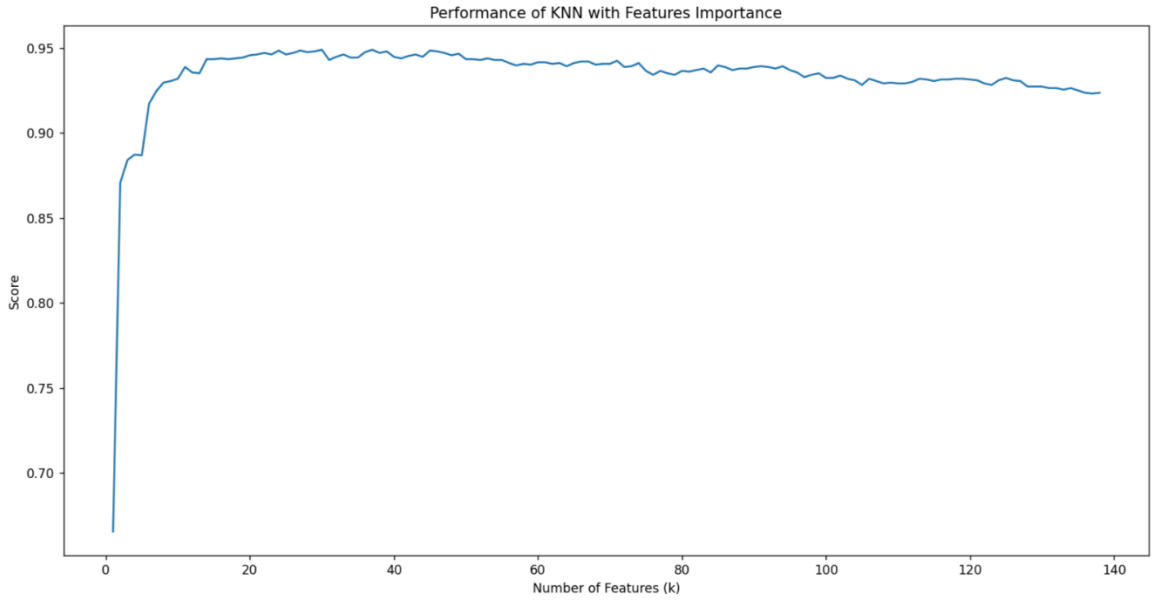
Rysunek 6. Wykres znaczenia cech



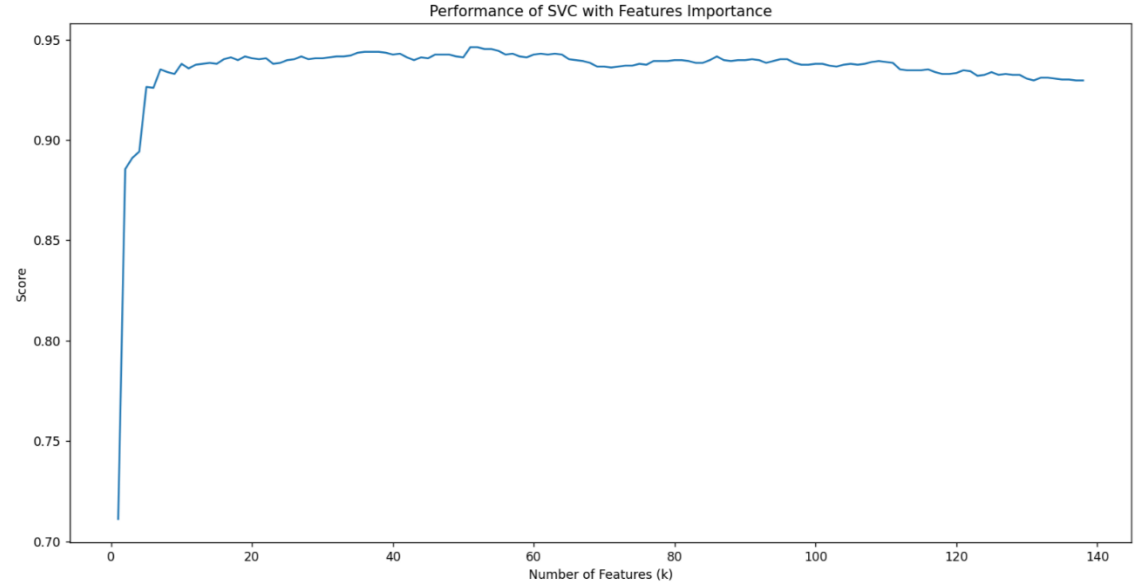
Rysunek 7. Znaczenie cech (posortowane od najbardziej znaczących)



Rysunek 8. Wpływ ilości cech na cross_value_score dla RandomForest



Rysunek 9. Wpływ ilości cech na cross_value_score dla KNN



Rysunek 10. Wpływ ilości cech na cross_value_score dla SVC

Klasyfikator	Cross_value_score(F1)	N_cech	Czas
Random Forest	0.951	80	401.70 s
K Neighbors Classifier	0.949	30	20.88 s
Support Vector Classifier	0.946	51	42.82 s

Tabela 3. Ocena jakości klasyfikacji dla najważniejszych cech

6. Wybór najlepszego klasyfikatora:

Klasyfikator	Cross_value_score(F1)	Metoda selekcji cech	Czas
Random Forest	0.951	Znaczenie cech	401.70 s
K Neighbors Classifier	0.949	Znaczenie cech	20.88 s
Support Vector Classification	0.946	Znaczenie cech	42.82 s
Random Forest	0.942	brak	5.93 s
Support Vector Classification	0.931	PCA	0.38 s
Support Vector Classification	0.930	brak	0.64 s
K Neighbors Classifier	0.926	PCA	0.20 s
K Neighbors Classifier	0.924	brak	0.31 s
Random Forest	0.921	PCA	3.76 s

Tabela 4. Statystyki klasyfikatorów posortowane według cross_value_score

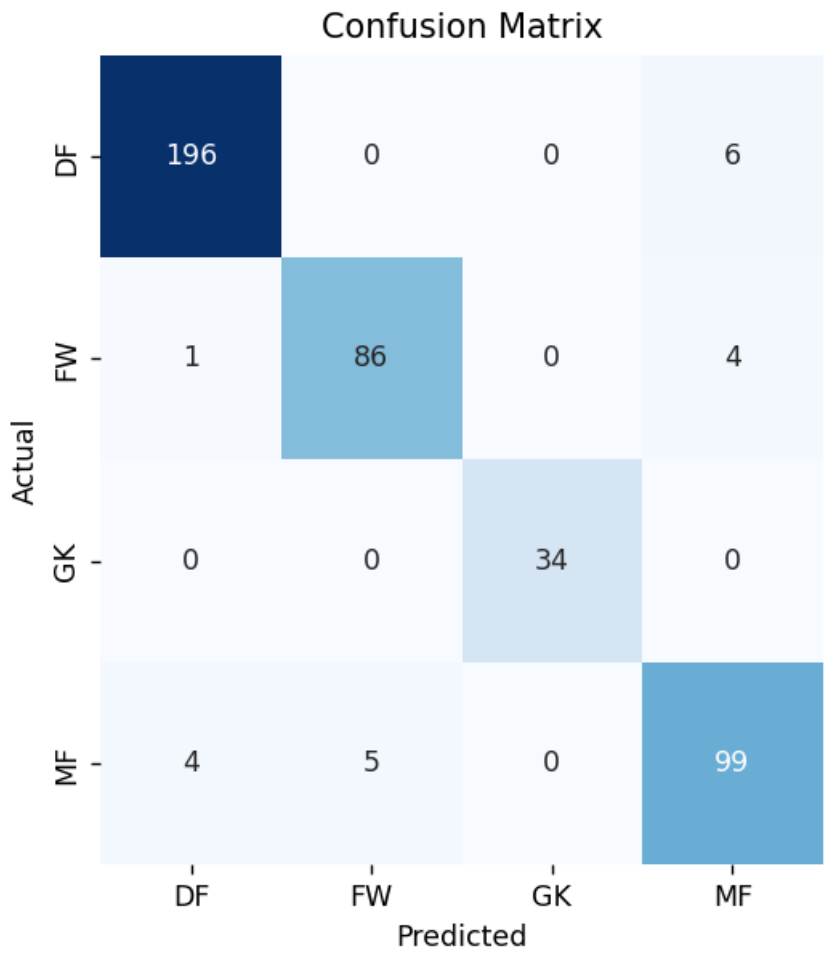
Klasyfikator	Cross_value_score(F1)	Metoda selekcji cech	Czas
K Neighbors Classifier	0.926	PCA	0.20
K Neighbors Classifier	0.924	brak	0.31
Support Vector Classification	0.931	PCA	0.38
Support Vector Classification	0.930	brak	0.64
Random Forest	0.921	PCA	3.76
Random Forest	0.942	brak	5.93
K Neighbors Classifier	0.949	Znaczenie cech	20.88
Support Vector Classification	0.946	Znaczenie cech	42.82
Random Forest	0.951	Znaczenie cech	401.70

Tabela 5. Statystyki klasyfikatorów posortowane według czasu

7. Klasyfikacja pozycji piłkarzy, szczegółowa ocena klasyfikatora, macierz pomyłek:

	precyzja	czułość	f1-score	wsparcie
DF	0.98	0.97	0.97	202
FW	0.95	0.95	0.95	91
GK	1.00	1.00	1.00	34
MF	0.91	0.92	0.91	108
dokładność			0.95	435
średnia macro	0.96	0.96	0.96	435
średnia ważona	0.95	0.95	0.95	435

Tabela 6. Raport klasyfikacji



Rysunek 11. Macierz pomyłek

4. Analiza uzyskanych wyników

1. Wczytanie i przetworzenie zbioru danych:

- Zbiór danych jest wczytywany z pliku CSV. Wymaga to użycia biblioteki *pandas*, oraz wyboru odpowiedniego kodowania pliku.

- W danych znajduje się 11 unikalnych pozycji, jednakże nie są to bezpośrednio pozycje na jakich piłkarze grali. Po analizie unikalnych pozycji możemy dojść do wniosku, że do naszego problemu klasyfikacji nie potrzebujemy pozycji takich jak „MFFW” (pomocnik-napastnik), ponieważ oznaczałoby to, że piłkarz może zostać zaklasyfikowany jednocześnie do dwóch pozycji.

- Docelowe pozycje wybrane do klasyfikacji to:

GK – bramkarz
DF – obrońca

MF – pomocnik
FW – napastnik

- Po wybraniu tych pozycji w zbiorze danych zostało 2175 rekordów, z czego wszystkie komórki mają jakąś wartość (nie ma NaN).

- Dodatkowo, porzucamy 5 kolumn (cech), ponieważ interesują nas tylko statystyki, czyli wartości numeryczne. Kolumny zaniechane: Nazwa gracza, narodowość gracza, pozycja, nazwa zespołu, liga. Przed porzuceniem pozycji zapisujemy ją jako pojedyncza kolumna w osobnej zmiennej, co przyda się w późniejszej analizie danych (Rysunek 1).

2. Wizualizacja problemu poprzez PCA(n=2):

-Poprzez redukcję wymiarowości do dwóch wymiarów jesteśmy w stanie zwizualizować problem tak, aby wyobraźnia była w stanie go przetworzyć.

-Na wykresie widać, że najłatwiej separowalną pozycją są bramkarze, natomiast najtrudniej odróżnić obrońców od pomocników oraz pomocników od napastników (Rysunek 2).

3. Naiwna klasyfikacja na podstawie wszystkich cech:

- Dalsze rozważania porównują działanie trzech klasyfikatorów:

  - Random Forest
  - K Neighbors Classifier
  - Support Vertor Classificator

- Do oceny jakości używany jest *cross_value_score*, czyli funkcja, która przeprowadza walidację krzyżową na naszych danych. Dzieli ona dane na zbiór treningowy i testowy, a następnie wielokrotnie przeprowadza procesu uczenia i testowania modelu na różnych kombinacjach tych zbiorów. Dzięki temu można uniknąć przetrenowania modelu na jednym konkretnym zbiorze danych i uzyskać bardziej realistyczne wyniki jego skuteczności. Walidacja krzyżowa zapobiega nadmiernemu dopasowaniu modelu do danych treningowych i zwiększa jego zdolność do generalizacji na nowe dane.

- Pierwsza próba wybranych klasyfikatorów została przeprowadzona na wszystkich dostępnych cechach.

- Najlepszy wynik z wykorzystaniem wszystkich cech osiągnął *Random Forest*, a najszybciej problem klasyfikacji rozwiązał *K Neighbors* (Tabela 1).

4. Klasyfikacja przy analizie PCA:

- Do wyboru pożądanej liczby czynników PCA wykorzystany jest wykres osypiska (Scree plot, Rysunek 3).

- Z wykresu zawierającego wszystkie 138 czynników trudno jest odczytać jakiekolwiek załamanie, więc wyświetlony zostaje wykres z liczbą 20 czynników.

- Metodą łokcia, odczytane załamanie następuje po *n*=13, więc PCA przyjmuje parametr *n_components* = 13 (Rysunek 4).

- Z wyników przeprowadzonej walidacji krzyżowej, przy użyciu danych zredukowanych do 13 cech, widać, że tym razem najlepszy klasyfikator to *SVC*, a najszybszy nadal *K Neigbors*. *Random Forest* pogorszył swój wynik, natomiast *K Neighbors* i *SVC* nieznacznie polepszyły swoje wyniki (Tabela 2).

5. Klasyfikacja przy użyciu RandomForestClassifier.feature_importances_:

-Wykorzystanie algorytmu *Random Forest* umożliwia odczytanie informacji o przydatności każdej cechy z osobna. Atrybut *feature_importances* jest obliczany jako średnia i odchylenie standardowe akumulacji spadku szumu w każdym drzewie.

-Dla lepszego zrozumienia zbioru danych, wyświetlone zostało 5 cech najbardziej wpływających na klasyfikację (Rysunek 5):

- PresMid3rd – liczba nałożonych pressingów na zawodnika z piłką w środku pola
-TouDef3rd – kontakty z piłką w okolicach defensywnych
-TouDefPen – kontakty z piłką we własnym polu karnym
-Clr – wykonane odbiory
-Ti – wykonane auty

- Znaczenie cech przedstawione jest na wykresie (Rysunek 6). Wraz ze spadkiem średniego szumu rośnie znaczenie cech w klasyfikacji. Cechy z dużym odchyleniem standardowym możemy traktować jako te najbardziej zaszumione, co widać na posortowanym wykresie (Rysunek 7). Od około n=80, cechy posiadają bardzo mało informacji, a bardzo dużo szumu.

- Tym razem do znalezienia najlepszej ilości cech, która pozwoli uzyskać najlepsze wyniki zastosujemy brute force. Dla każdej możliwej ilości cech, zaczynając od tych najbardziej istotnych, sprawdzimy jaki wynik *cross_value_score* otrzymuje każdy klasyfikator. Zależność wyniku score od liczby cech branych pod uwagę w klasyfikacji przedstawiają wykresy odpowiednio na rysunkach Rysunek 8, Rysunek 9, Rysunek 10.

- Taki sposób doboru cech polepszył wyniki wszystkich klasyfikatorów. Niestety czas potrzebny do takich obliczeń wzrósł wykładniczo. *Random Forest* najlepiej poradził sobie przy klasyfikacji z danymi o 80 cechach, *KNN* z 30 cechami, a *SVC* z 51 (Tabela 3).

6. Wybór najlepszego klasyfikatora:

- Na podstawie zgromadzonych danych wybieramy najprecyzyjniejszy klasyfikator. Najwyższy wynik otrzymał *Random Forest* przy uczeniu z danymi o 80 najważniejszych cechach. Uzyskał on *cross_value_score* = 0.951 (Tabela 4). Najszybszym klasyfikatorem okazał się *K Neighbors,* przy użyciu redukcji wymiarowości PCA (Tabela 5).

7. Klasyfikacja pozycji piłkarzy, szczegółowa ocena klasyfikatora, macierz pomyłek:

- Klasyfikacja została przeprowadzona przy użyciu funkcji *test_train_split*. Wszystkie oceny jakości są bardzo wysokie, przyjmują wartości powyżej 95%. Nauczony klasyfikator został przetestowany na 202 obrońcach, 91 napastnikach, 34 bramkarzach, 108 pomocnikach, co w sumie dało 435 piłkarzy. Dokładność wyniosła 95%, więc 415 piłkarzy zostało poprawnie zaklasyfikowanych. Średnie ważone precyzji, czułości oraz f1-score też wynosiły 95% co oznacza, że 95% pozytywnych jest rzeczywiście pozytywne, 95% pozytywnych przypadków zostało pozytywnie zaklasyfikowanych, oraz że średnia harmoniczna tych dwóch wskaźników jest dokładnie taka sama (Tabela 6).

- Macierz pomyłek pokazuje, ile dokładnie przypadków zostało zaklasyfikowanych do każdej z grup. Żaden bramkarz nie został źle zaklasyfikowany. Jeden napastnik został zaklasyfikowany jako obrońca. W reszcie przypadków pojawiło się więcej negatywnych klasyfikacji (Rysunek 11).

5. Wnioski:

- Klasyfikacja pozycji piłkarzy na podstawie ich statystyk meczowych jest możliwa z użyciem różnych klasyfikatorów, oraz jest możliwe osiągnięcie dokładności każdego klasyfikatora z {*Random Forest, K Neighbors, Suport Vector Classifier*} na poziomie powyżej 94%.

- Najdokładniejszy klasyfikator to *Random Forest*, który przy uczeniu wykorzystuje tylko 80 najważniejszych cech w zbiorze danych. Jego dokładność wynosi powyżej 95%. Model uzyskał tak dużą dokładność kosztem czasu, w którym uczył się danych. Potrzebował on 401.70s (ponad 6 min) na znalezienie najlepszej liczby cech, która pozwoliła osiągnąć tak wysoki wynik.

- Najszybszy klasyfikator to *K Neighbors*, który w zaledwie 0.20s osiągnął *cross_val_score* na poziomie 92,6%.

-Najoptymalniejszy klasyfikator to *Suport Vector Classifier*, ponieważ osiągnął wysoki wynik 93,1% w krótkim czasie 0,38s.

-Modele są podatne na optymalizacje poprzez skrupulatną i przemyślaną selekcję cech.

- Uczenie modelu na wszystkich cechach prowadzi do przetrenowania, co oznacza, że model uczy się niepotrzebnych szumów ze zbioru danych. W efekcie model ma trudności z poprawną klasyfikacją nowych próbek. Warto zatem ograniczyć liczbę cech, aby poprawić zdolność modelu do generalizacji.

-Zastosowanie PCA pozwala na pewien kompromis pomiędzy dokładnością klasyfikatora a czasem jego uczenia. Dzięki redukcji wymiarowości dane stają się mniej skomplikowane a także pozbywamy się nadmiaru informacji, które mogą zmniejszyć precyzję modelu.

-Najłatwiej separowalną klasą są bramkarze. Najtrudniej odróżnić obrońców od napastników i pomocników od napastników.

-Cechy, które zawierają najwięcej informacji o pozycji piłkarza, mają największy wpływ na klasyfikację to:

- liczba nałożonych pressingów na zawodnika z piłką w środku pola
- kontakty z piłką w okolicach defensywnych
- kontakty z piłką we własnym polu karnym
- wykonane odbiory
- wykonane auty

-Podczas tego projektu nauczyłem się wiele na temat analizy danych i klasyfikacji za pomocą metod uczenia maszynowego. Napotkane problemy utwierdziły mnie w przekonaniu, jak ważne jest przygotowanie danych, selekcja cech i optymalizacja modelu.

- Nauczyłem się, jak PCA poprawia wydajność modelu a także jak niepożądane może być przetrenowanie modelu.

- Istotne jest, że aby wytrenować jak najlepszy model, potrzeba czynnika ludzkiego do analizy i oceny wyników.