Michał Romańczyk s20422, Kacper Lis s20713

PRO1D 2021/2022

Raport

Predykcja skali przestępczości

Spis treści

[Spis ilustracji II](#_Toc93861519)

[Cel 1](#_Toc93861520)

[Opis zbioru danych 1](#_Toc93861521)

[Opis atrybutów 1](#_Toc93861522)

[Dystrybucja atrybutu decyzyjnego 4](#_Toc93861523)

[Opis bibliotek 4](#_Toc93861524)

[Scikit-learn 4](#_Toc93861525)

[Moduły 5](#_Toc93861526)

[Numpy 5](#_Toc93861527)

[Pandas 5](#_Toc93861528)

[Metodologia 5](#_Toc93861529)

[Przetworzenie danych 5](#_Toc93861530)

[Porządkowanie danych 5](#_Toc93861531)

[Normalizacja 6](#_Toc93861532)

[Wybór atrybutów (Feature selection) 6](#_Toc93861533)

[Podział na zbiory treningowe i testowe 6](#_Toc93861534)

[Opis modelu 7](#_Toc93861535)

[Sieć neuronowa 7](#_Toc93861536)

[Drzewo regresyjne 7](#_Toc93861537)

[Regresja liniowa 8](#_Toc93861538)

[Regresja nie-liniowa 8](#_Toc93861539)

[Losowy las regresyjny 8](#_Toc93861540)

[Ocena jakości modelu 9](#_Toc93861541)

[Cross validation 9](#_Toc93861542)

[Rezultat 10](#_Toc93861543)

[Sieć neuronowa 10](#_Toc93861544)

[Drzewo regresyjne 11](#_Toc93861545)

[Regresja liniowa 12](#_Toc93861546)

[Regresja nieliniowa 13](#_Toc93861547)

[Losowy las regresyjny 14](#_Toc93861548)

[Porównanie 16](#_Toc93861549)

[Wnioski 17](#_Toc93861550)

[Źródła 17](#_Toc93861551)

# Spis ilustracji

[Wykres 1: Rezultat MSE Sieć neuronowa. 10-krotna walidacja krzyżowa. 10](#_Toc92715398)

[Wykres 2: Rezultat R2 Sieć neuronowa. 10-krotna walidacja krzyżowa. 11](#_Toc92715399)

[Wykres 3: Rezultat MSE Drzewo regresyjne. 10-krotna walidacja krzyżowa. 11](#_Toc92715400)

[Wykres 4: Rezultat R2 Drzewo regresyjne. 10-krotna walidacja krzyżowa. 12](#_Toc92715401)

[Wykres 5: Rezultat MSE Regresja liniowa. 10-krotna walidacja krzyżowa. 12](#_Toc92715402)

[Wykres 6: Rezultat R2 Regresja liniowa. 10-krotna walidacja krzyżowa. 13](#_Toc92715403)

[Wykres 7: Rezultat MSE Regresja nieliniowa. 10-krotna walidacja krzyżowa. 13](#_Toc92715404)

[Wykres 8: Rezultat R2 Regresja nieliniowa. 10-krotna walidacja krzyżowa. 14](#_Toc92715405)

[Wykres 9: Rezultat MSE Losowy las regresyjny. 10-krotna walidacja krzyżowa. 14](#_Toc92715406)

[Wykres 10: Rezultat R2 Losowy las regresyjny. 10-krotna walidacja krzyżowa. 15](#_Toc92715407)

[Wykres 11: Porównanie wyników miarą MSE 15](#_Toc92715408)

[Wykres 12: Porównanie wyników miarą R2 16](#_Toc92715409)

# Cel

Celem badania jest stworzenie i zaprezentowanie modeli regresyjnych, które będą w stanie przewidzieć stopień przestępczości w miastach USA. Predykcje będą podejmowane na podstawie wcześniej przygotowanego zbioru danych. Model powinien poprawnie wskazać stopień przestępczości nowego regionu znając jego dane demograficzne.

# Opis zbioru danych

Zbiór danych zawiera wszystkie potrzebne dane do wyuczenia modelu i przetestowania go. Jednak dane, które nie miały związku z przestępczością zostały odfiltrowane przez twórców zbioru danych. Zbiór danych oryginalnie składał się ze 122 atrybutów wykorzystywanych do predykcji, 5 atrybutów dodatkowych, które nie będą wykorzystywane do predykcji, oraz dodatkowo atrybut decyzyjny ***Per Capita Violent Crimes (ViolentCrimesPerPop).*** Ilość rekordów znajdujących się w zbiorze to 1994.Dane zawarte w zbiorze posiadają między innymi informację o tym jaka jest mediana zarobków rodzinnych, ilość interwencji policji, ile jest policjantów w stosunku do sumy populacji.

Atrybut ***Per Capita Violent Crimes*** został obliczony na podstawie populacji i sumy ciężkich przestępstw w Stanach Zjednoczonych: morderstwo, gwałt, kradzież, napaść. Przy zbieraniu danych pojawiały się pewne nieścisłości w związku z różnicami w prawie Stanowym.

Wszystkie dane przygotowane przez twórców zbioru zostały już znormalizowane w przedziale dyskretnym w granicach **0.00 – 1.00**. Została utrzymana poprawna dystrybucja danych. Niemniej, dla wartości skrajnych (wszystkie wartości trzykrotności standardowej dewiacji powyżej średniej są normalizowane do 1.00; wszystkie wartości trzykrotności standardowej dewiacji poniżej średniej są normalizowane do 0.00).

Warto również zwrócić uwagę na to, że dane zostały zbierane wśród jednostek policji, których wielkość przekraczała 100 funkcjonariuszy, dodatkowo kilka mniejszych jednostek.

## Opis atrybutów

Informacje: (122 predykcyjne, 5 nie predykcyjne, 1 atrybut decyzyjny)

* state: Stan (region) (liczbowo) – nie predykcyjny (nominalny)
* county: kod numeryczny dla powiatu – nie predykcyjny (numeryczny)
* community: kod numeryczny dla wspólnoty – nie predykcyjny (numeryczny)
* communityname: nazwa wspólnoty – nie predykcyjny (słowny)
* fold: liczba fałd dla nielosowej 10-krotnej walidacji krzyżowej, potencjalnie przydatna do debugowania, testy sparowane – nie predykcyjny (numeryczny)
* population: ludność dla wspólnoty: (numeryczny)
* householdsize: średnia liczba osób w gospodarstwie domowym (numeryczny)
* racepctblack: odsetek ludności pochodzenia afroamerykańskiego (numeryczny)
* racePctWhite: procent populacji, która jest rasy kaukaskiej (numeryczny)
* racePctAsian: procent populacji o pochodzeniu azjatyckim (numeryczny)
* racePctHisp: odsetek ludności pochodzenia hispanoamerykańskiego (numeryczny)
* agePct12t21: odsetek ludności w wieku 12-21 lat (numeryczny)
* agePct12t29: odsetek ludności w wieku 12-29 lat (numeryczny)
* agePct16t24: odsetek ludności w wieku 16-24 lata (numeryczny)
* agePct65up: odsetek ludności w wieku 65 lat i więcej (numeryczny)
* numbUrban: liczba osób mieszkających na obszarach sklasyfikowanych jako miejskie (numeryczny)
* pctUrban: odsetek osób mieszkających na obszarach sklasyfikowanych jako miejskie (numeryczny)
* medIncome: mediana dochodu gospodarstwa domowego (numeryczny)
* pctWWage: odsetek gospodarstw domowych o dochodach z wynagrodzeń w 1989 r. (numeryczny)
* pctWFarmSelf: odsetek gospodarstw domowych z dochodami z gospodarstwa rolnego lub z pracy na własny rachunek w 1989 r. (numeryczny)
* pctWInvInc: odsetek gospodarstw domowych o dochodach z inwestycji / wynajmu w 1989 r. (numeryczny)
* pctWSocSec: odsetek gospodarstw domowych z dochodami z ubezpieczeń społecznych w 1989 r. (numeryczny)
* pctWPubAsst: odsetek gospodarstw domowych o dochodach z pomocy publicznej w 1989 r. (numeryczny)
* pctWRetire: odsetek gospodarstw domowych o dochodach z pomocy publicznej w 1989 r. (numeryczny)
* medFamInc: mediana dochodu rodziny (różni się od dochodu gospodarstwa domowego dla gospodarstw nierodzinnych) (numeryczny)
* perCapInc: dochód na mieszkańca (numeryczny)
* whitePerCap: dochód na głowę mieszkańca dla rasy kaukaskiej (numeryczny)
* blackPerCap: dochód na głowę mieszkańca ameryki afrykańskiej (numeryczny)
* indianPerCap: dochód na jednego mieszkańca dla rdzennych amerykanów (numeryczny)
* AsianPerCap: dochód na głowę mieszkańca dla osób o azjatyckim pochodzeniu (numeryczny)
* OtherPerCap: dochód per capita dla osób o "innym" dziedzictwie (numeryczny)
* HispPerCap: dochód na mieszkańca dla osób pochodzenia hispanoamerykańskiego (numeryczny)
* NumUnderPov: liczba osób poniżej poziomu ubóstwa (numeryczny)
* PctPopUnderPov: odsetek osób znajdujących się poniżej poziomu ubóstwa (numeryczny)
* PctLess9thGrade: odsetek osób w wieku 25 lat i więcej z wykształceniem niższym niż 9-cio klasowe (numeryczny)
* PctNotHSGrad: odsetek osób w wieku 25 lat i więcej, które nie ukończyły szkoły średniej (numeryczny)
* PctBSorMore: odsetek osób w wieku 25 lat i więcej posiadających dyplom licencjata lub wykształcenie wyższe (numeryczny)
* PctUnemployed: odsetek osób w wieku 16 lat i więcej, aktywnych zawodowo i bezrobotnych (numeryczny)
* PctEmploy: odsetek osób w wieku 16 lat i więcej, które są zatrudnione (numeryczny)
* PctEmplManu: odsetek osób w wieku 16 lat i więcej zatrudnionych w przemyśle (numeryczny)
* PctEmplProfServ: odsetek osób w wieku 16 lat i więcej zatrudnionych w usługach zawodowych (numeryczny)
* PctOccupManu: odsetek osób w wieku 16 lat i więcej zatrudnionych w przemyśle (numeryczny)
* PctOccupMgmtProf: odsetek osób w wieku 16 lat i więcej zatrudnionych w zawodach związanych z zarządzaniem lub zawodem (numeryczny)
* MalePctDivorce: odsetek mężczyzn rozwiedzionych (numeryczny)
* MalePctNevMarr: odsetek mężczyzn, którzy nigdy nie zawarli związku małżeńskiego (numeryczny)
* FemalePctDiv: odsetek kobiet, które się rozwiodły (numeryczny)
* TotalPctDiv: procent populacji, która jest rozwiedziona (numeryczny)
* PersPerFam: średnia liczba osób w rodzinie (numeryczny)
* PctFam2Par: odsetek rodzin (z dziećmi), które są prowadzone przez dwoje rodziców (numeryczny)
* PctKids2Par: odsetek dzieci w domach rodzinnych z dwojgiem rodziców (numeryczny)
* PctYoungKids2Par: odsetek dzieci w wieku 4 lat i poniżej w gospodarstwach domowych z dwoma rodzicami (numeryczny)
* PctTeen2Par: odsetek dzieci w wieku 12-17 lat w gospodarstwach domowych z dwoma rodzicami (numeryczny)
* PctWorkMomYoungKids: odsetek matek dzieci w wieku 6 lat i młodszych aktywnych zawodowo (numeryczny)
* PctWorkMom: odsetek matek dzieci poniżej 18 roku życia aktywnych zawodowo (numeryczny)
* NumIlleg: liczba dzieci urodzonych przez osoby nigdy nie będące w związku małżeńskim (numeryczny)
* PctIlleg: odsetek dzieci urodzonych przez osoby nigdy nie będące w związku małżeńskim (numeryczny)
* NumImmig: całkowita liczba osób, o których wiadomo, że są urodzone za granicą (numeryczny)
* PctImmigRecent: procent imigrantów, którzy imigrowali w ciągu ostatnich 3 lat (numeryczny)
* PctImmigRec5: procent imigrantów, którzy imigrowali w ciągu ostatnich 5 lat (numeryczny)
* PctImmigRec8: procent imigrantów, którzy imigrowali w ciągu ostatnich 8 lat (numeryczny)
* PctImmigRec10: procent imigrantów, którzy imigrowali w ciągu ostatnich 10 lat (numeryczny)
* PctRecentImmig: procent ludności, która wyemigrowała w ciągu ostatnich 3 lat (numeryczny)
* PctRecImmig5: procent ludności, która wyemigrowała w ciągu ostatnich 5 lat (numeryczny)
* PctRecImmig8: procent ludności, która wyemigrowała w ciągu ostatnich 8 lat (numeryczny)
* PctRecImmig10: procent ludności, która wyemigrowała w ciągu ostatnich 10 lat (numeryczny)
* PctSpeakEnglOnly: procent osób posługujących się wyłącznie językiem angielskim (numeryczny)
* PctNotSpeakEnglWell: procent osób, które nie znają dobrze języka angielskiego (numeryczny)
* PctLargHouseFam: procent rodzinnych gospodarstw domowych, które są duże (6 lub więcej) (numeryczny)
* PctLargHouseOccup: procent wszystkich zamieszkanych gospodarstw domowych, które są duże (6 lub więcej) (numeryczny)
* PersPerOccupHous: średnia osób na gospodarstwo domowe (numeryczny)
* PersPerOwnOccHous: średnia osób na gospodarstwo domowe zajmowane przez właściciela (numeryczny)
* PersPerRentOccHous: średnia liczba osób na jedno wynajmowane gospodarstwo domowe (numeryczny)
* PctPersOwnOccup: odsetek osób w gospodarstwach domowych zamieszkałych przez właścicieli (numeryczny)
* PctPersDenseHous: procent osób w gęstym budownictwie mieszkaniowym (więcej niż 1 osoba na pokój) (numeryczny)
* PctHousLess3BR: procent mieszkań z mniej niż 3 sypialniami (numeryczny)
* MedNumBR: mediana liczby sypialni (numeryczny)
* HousVacant: liczba wolnych gospodarstw domowych (numeryczny)
* PctHousOccup: procent mieszkań zamieszkanych (numeryczny)
* PctHousOwnOcc: procent gospodarstw domowych zamieszkałych przez właścicieli (numeryczny)
* PctVacantBoarded: procent pustych mieszkań, które są zabite deskami (numeryczny)
* PctVacMore6Mos: procent pustostanów, które były niewykorzystane przez ponad 6 miesięcy (numeryczny)
* MedYrHousBuilt: mediana roku wybudowania mieszkań (numeryczny)
* PctHousNoPhone: procent zamieszkanych mieszkań bez telefonu (numeryczny)
* PctWOFullPlumb: procent mieszkań bez pełnej instalacji wodno-kanalizacyjnej (numeryczny)
* OwnOccLowQuart: mieszkania zajmowane przez właścicieli - wartość dolnego kwartyla (numeryczny)
* OwnOccMedVal: mieszkania zajmowane przez właścicieli - wartość mediany (numeryczny)
* OwnOccHiQuart: mieszkania zajmowane przez właścicieli - wartość górnego kwartyla (numeryczny)
* RentLowQ: wynajem mieszkań - czynsz z dolnego kwartyla (numeryczny)
* RentMedian: mieszkania czynszowe - mediana czynszu (zmienna ze spisu powszechnego H32B z pliku STF1A) (numeryczny)
* RentHighQ: mieszkania czynszowe - górny kwartyl czynszu (numeryczny)
* MedRent: mediana czynszu brutto (zmienna spisowa H43A z pliku STF3A - obejmuje media) (numeryczny)
* MedRentPctHousInc: mediana czynszu brutto jako procent dochodu gospodarstwa domowego (numeryczny)
* MedOwnCostPctInc: mediana kosztu właściciela jako procent dochodu gospodarstwa domowego - dla właścicieli z kredytem hipotecznym (numeryczny)
* MedOwnCostPctIncNoMtg: mediana kosztu właściciela jako procent dochodu gospodarstwa domowego - dla właścicieli bez kredytu hipotecznego (numeryczny)
* NumInShelters: liczba osób w schroniskach dla bezdomnych (numeryczny)
* NumStreet: liczba osób bezdomnych policzonych na ulicy (numeryczny)
* PctForeignBorn: procent osób urodzonych za granicą (numeryczny)
* PctBornSameState: procent osób urodzonych w tym samym stanie co obecnie mieszkający (numeryczny)
* PctSameHouse85: odsetek osób mieszkających w tym samym domu co w 1985 r. (5 lat wcześniej) (numeryczny)
* PctSameCity85: procent osób mieszkających w tym samym mieście co w 1985 r. (5 lat wcześniej) (numeryczny)
* PctSameState85: procent osób mieszkających w tym samym stanie co w 1985 r. (5 lat wcześniej) (numeryczny)
* LemasSwornFT: liczba zaprzysiężonych funkcjonariuszy policji w pełnym wymiarze godzin (numeryczny)
* LemasSwFTPerPop: zaprzysiężeni policjanci w pełnym wymiarze godzin na 100 tys. mieszkańców (numeryczny)
* LemasSwFTFieldOps: liczba zaprzysiężonych funkcjonariuszy policji zatrudnionych w pełnym wymiarze czasu pracy w terenie (na ulicy, a nie w administracji itp.) (numeryczny)
* LemasSwFTFieldPerPop: zaprzysiężeni funkcjonariusze policji w pełnym wymiarze godzin w operacjach terenowych (na ulicy, w przeciwieństwie do administracji itp.) na 100 tys. mieszkańców (numeryczny)
* LemasTotalReq: wnioski ogółem dla policji (numeryczny)
* LemasTotReqPerPop: całkowita liczba wniosków o policję na 100 tys. mieszkańców (numeryczny)
* PolicReqPerOffic: całkowita liczba wniosków o policję na jednego funkcjonariusza policji (numeryczny)
* PolicPerPop: policjanci na 100 tys. mieszkańców (numeryczny)
* RacialMatchCommPol: miara dopasowania rasowego pomiędzy społecznością a siłami policyjnymi. Wysokie wartości wskazują, że proporcje w społeczności i siłach policyjnych są podobne. (numeryczny)
* PctPolicWhite: odsetek policjantów rasy białej (numeryczny)
* PctPolicBlack: odsetek policjantów będących amerykanami afrykańskiego pochodzenia (numeryczny)
* PctPolicHisp: procent policjantów pochodzenia hispanoamerykańskiego (numeryczny)
* PctPolicAsian: procent policjantów, którzy są Azjatami (numeryczny)
* PctPolicMinor: procent policjantów należących do jakiejkolwiek mniejszości (numeryczny)
* OfficAssgnDrugUnits: liczba funkcjonariuszy przydzielonych do specjalnych jednostek antynarkotykowych (numeryczny)
* NumKindsDrugsSeiz: liczba przechwyconych różnych rodzajów narkotyków (numeryczny)
* PolicAveOTWorked: średnia liczba godzin nadliczbowych przepracowanych przez policję (numeryczny)
* LandArea: obszar lądowy w milach kwadratowych (numeryczny)
* PopDens: gęstość zaludnienia w osobach na milę kwadratową (numeryczny)
* PctUsePubTrans: odsetek osób korzystających z transportu publicznego w dojazdach do pracy (numeryczny)
* PolicCars: liczba samochodów policyjnych (numeryczny)
* PolicOperBudg: budżet operacyjny policji (numeryczny)
* LemasPctPolicOnPatr: procent zaprzysiężonych pełnoetatowych funkcjonariuszy policji w patrolu (numeryczny)
* LemasGangUnitDeploy: rozmieszczona jednostka gangowa (numeryczny - ale tak naprawdę porządkowy - 0 oznacza NIE, 1 oznacza TAK, 0,5 oznacza niepełny etat)
* LemasPctOfficDrugUn: odsetek funkcjonariuszy przydzielonych do jednostek antynarkotykowych (numeryczny)
* PolicBudgPerPop: budżet operacyjny policji na ludność (numeryczny)
* ViolentCrimesPerPop: całkowita liczba przestępstw z użyciem przemocy na 100 tys. mieszkańców (numeryczny) atrybut decyzyjny

## Dystrybucja atrybutu decyzyjnego

**Range Frequency**  
 *0.000-0.067 484  
 0.067-0.133 420  
 0.133-0.200 284  
 0.200-0.267 177  
 0.267-0.333 142  
 0.333-0.400 113  
0.400-0.467 59  
0.467-0.533 76  
0.533-0.600 57  
0.600-0.667 38  
0.667-0.733 37  
0.733-0.800 20  
0.800-0.867 23  
0.867-0.933 14  
0.933-1.000 50*

# Opis bibliotek

## Scikit-learn

Scikit-learn jest darmową biblioteką uczenia maszynowego dla języka programowania Python. Zawiera różne algorytmy klasyfikacji, regresji i klastrowania, w tym maszyny wektorów nośnych, lasy losowe, gradient boosting, k-means i DBSCAN, i została zaprojektowana do współpracy z numerycznymi i naukowymi bibliotekami Pythona - NumPy i SciPy. [1]

W naszym projekcie skorzystaliśmy z wielu modułów dostarczonych przez tą bibliotekę. Głównie były to implementacje modeli i algorytmów sztucznej inteligencji, ale też posłużył nam przy porządkowaniu danych i walidacji naszych modeli.

### Moduły

1. sklearn.neural\_network.MLPRegressor [2] – Model wielowarstwowego perceptronu regresyjnego.
2. sklearn.tree.DecisionTreeRegressor [3] – Model regresyjnego drzewa decyzyjnego
3. sklearn.linear\_model.LinearRegression [4] – Model regresji liniowej.
4. sklearn.svm.SVR [5] – Model regresji nieliniowej.
5. sklearn.model\_selection.cross\_val\_score [6] – Moduł odpowiedzialny za przeprowadzenie walidacji krzyżowej i przechowanie wyników.
6. sklearn.ensemble.RandomForestRegressor [7] – Model losowego lasu regresyjnego.
7. sklearn.model\_selection.RandomizedSearchCV [8] – Randomizowane wyszukiwanie hiper parametrów. Parametry estymatora używanego do zastosowania tych metod są optymalizowane poprzez krzyżowe przeszukiwanie ustawień parametrów.

## Numpy

NumPy jest biblioteką dla języka programowania Python, dodającą obsługę dużych, wielowymiarowych tablic i macierzy, wraz z dużą kolekcją wysokopoziomowych funkcji matematycznych do operowania na tych tablicach. [9]

Biblioteka NumPy posłużyła nam jako narzędzie do zarządzania strukturami danych. Przy jej użyciu porządkowaliśmy dane i przekształcaliśmy tablice danych.

## Pandas

Pandas jest biblioteką programistyczną napisaną dla języka programowania Python, służącą do manipulacji i analizy danych. W szczególności oferuje struktury danych i operacje do manipulowania tablicami numerycznymi i szeregami czasowymi. Jest to oprogramowanie darmowe. [10]

# Metodologia

## Przetworzenie danych

### Porządkowanie danych

W pierwszej kolejności musimy oczyścić dane z atrybutów, które nie są potrzebne podczas podejmowania decyzji. W pliku z danymi pierwsze pięć kolumn jest informacyjnych, więc w pierwszym kroku się ich pozbyliśmy.

Następnym krokiem jest uzupełnienie wartości brakujących. Po analizie zbioru danych zauważyliśmy, że w 22 parametrach znajdują się brakujące informacje. W każdej kolumnie (poza jedną), w której występowały brakujące informacje, było to aż 1675 brakujących rekordów. Przy łącznej liczbie 1994 rekordów, brakujące rekordy stanowią aż 84% całości. Dodatkowo w jednej z kolumn znajdowała się tylko jedna brakująca informacja, którą postanowiliśmy uzupełnić wartością średnią ze wszystkich rekordów w tej kolumnie. Do problemu podeszliśmy na trzy różne sposoby:

1. Pozbycie się kolumn, w których znajdowały się brakujące informacje. Podjęliśmy tą decyzję ze względu na to, że odsetek brakujących informacji, był na tyle duży, że wprowadzenie danych mogło powodować pewne zakłamanie. (df\_cleaned)
2. Uzupełnienie brakujących danych średnią wartością istniejących danych odpowiadającego argumentu. (df\_mean)
3. Uzupełnienie brakujących danych medianą istniejących danych odpowiadającego argumentu.

Dzięki tej decyzji możemy sprawdzić, które z podejść będzie miało najlepszy rezultat. (df\_med)

### Normalizacja

Z racji, że dane zostały dostarczone już w postaci znormalizowanej nie podjęliśmy żadnych działań w tym kierunku.

### Wybór atrybutów (Feature selection)

Z racji na znaczną ilość argumentów, na podstawie których będzie podejmowana decyzja, oraz pomimo informacji o tym, że dane które nie są ze sobą powiązane, zostały wyrzucone ze zbioru, zdecydowaliśmy się zastosować mechanizm *Feature selection*. Zastosowanie tej techniki pozwoli wyeliminować przypadek, w którym jednak pomiędzy niektórymi danymi nie zachodzi odpowiednio duża korelacja. Dodatkowo uzyskamy mniej złożony model, skrócimy czas obliczeń oraz zwiększymy dokładność.

Mechanizm *Feature selection* polega na zmniejszeniu liczby wymiarów zbioru danych, do tych, pomiędzy którymi występuje silna korelacja. W naszym podejściu skorzystaliśmy z metody, która polega na tym, że każdy algorytm oddaje swój głos na odpowiednią liczbę atrybutów (parametr *k*). Następnie z atrybutów, które dostały przynajmniej jeden głos, został stworzony czwarty zestaw danych, na którym stworzymy model.

Do wykonania tego zadania posłużyliśmy się modułem z biblioteki [scikit-learn](#_Scikit-learn), SelectKBest. W celu uzyskania głosów wykorzystaliśmy dwie funkcje (scoring\_func) dostępne dla danych regresyjnych z tej biblioteki:

1. f\_regression [11] - Szybki model liniowy do testowania efektu pojedynczego regresora, sekwencyjnie dla wielu regresorów.
2. mutual\_info\_regression [12] - Funkcja opiera się na metodach nieparametrycznych opartych na estymacji entropii z odległości k-najbliższych sąsiadów.

Parametr k, czyli liczbę głosów które ma dostępny jeden algorytm, ustawiony jest na wartość 25. W wyniku głosowania tych dwóch algorytmów zostaje wybrane od 25 do 30 atrybutów, co skutkuje zmniejszeniem przestrzeni danych o ponad 70%.

### Podział na zbiory treningowe i testowe

Każdy zestaw danych, który przygotowaliśmy, został podzielony na dane testowe i dane treningowe, które następnie zostały użyte do wstępnego szkolenia i testowania modeli. Proporcje w jakich został podzielony każdy zestaw to 3:1, czyli 75% to dane treningowe, a 25% to dane testowe.

## Opis modelu

W naszym podejściu postanowiliśmy skorzystać z pięciu klasyfikatorów regresyjnych. W trakcie badania przetestowaliśmy ich skuteczność na każdym z czterech wcześniej przygotowanych zbiorach danych. Wszystkie modele wykorzystane pochodziły z biblioteki [scikit-learn](#_Scikit-learn).

Parametry dla **sieci neuronowej** i dla **losowego lasu regresyjnego** zostały dobrane na podstawie metody brute-force, która z wcześniej określonych przedziałów wybierała takie zestawy parametrów, dla których wyniki były najlepsze. Parametry, które zostały określone tą metodą zostały oznaczone kolorem czerwonym. w sekcjach o odpowiednim modelu.

### Sieć neuronowa

Perceptrony wielowarstwowe, w skrócie MLP, są klasycznym typem sieci neuronowej. Składają się one z jednej lub więcej warstw neuronów. Dane są podawane do warstwy wejściowej, może istnieć jedna lub więcej warstw ukrytych zapewniających poziomy abstrakcji, a przewidywania są dokonywane na warstwie wyjściowej, zwanej również warstwą widoczną.

Uczenie odbywa się w perceptronie poprzez zmianę wag połączeń po przetworzeniu każdej części danych, w oparciu o wielkość błędu na wyjściu w porównaniu z oczekiwanym wynikiem. Jest to przykład uczenia nadzorowanego , który jest realizowany przez propagację wsteczną , uogólnienie algorytmu najmniejszych średnich kwadratów w perceptronie liniowym.

Model, który wykorzystaliśmy optymalizuje kwadrat błędy przy użyciu stochastycznego optymalizatora opartego na gradiencie zaproponowanego przez Kingmę, Diederika i Jimmy’ego Ba.

Do stworzenia modelu sieci neuronowej posłużyliśmy się wielowarstwowym perceptronem regresyjnym. Parametry dobieraliśmy eksperymentalnie oraz metodą brute-force, tak aby wyniki były jak najlepsze.

1. hidden\_layer\_sizes – I-ty element reprezentuje liczbę neuronów w i-tej warstwie ukrytej. **Wartość=**(90,40,10)
2. activation – Funkcja aktywacji dla ukrytej warstwy. **Wartość=**’logistic’
3. alpha – Parametr kary L2 (okres regulacji). **Wartość=**0.001
4. learning\_rate\_init – Zastosowana początkowa szybkość uczenia się. Kontroluje wielkość kroku w aktualizacji wag. **Wartość=**0.001
5. learning\_rate – Rodzaj nauki dla aktualizacji wagi. **Wartość=**’adaptive’
6. solver – Solver do optymalizacji wagi. **Wartość=**’adam’
7. max\_itter – Maksymalna liczba iteracji. **Wartość=**10000
8. early\_stopping – Czy używać wczesnego zatrzymywania do zakończenia trenowania, gdy wynik walidacji nie poprawia się. **Wartość=**True

Moduł: sklearn.neural\_network.MLPRegressor

### Drzewo regresyjne

Drzewo regresji jest budowane poprzez proces znany jako binarne rekurencyjne partycjonowanie, które jest procesem iteracyjnym, który dzieli dane na partycje lub gałęzie, a następnie kontynuuje podział każdej partycji na mniejsze grupy w miarę przesuwania się metody w górę każdej gałęzi.

Takie drzewo może być traktowane jako wariant drzewa decyzyjnego, zaprojektowanego do przybliżenia funkcji o wartościach rzeczywistych, zamiast do metod klasyfikacji.

Różnica pomiędzy drzewem klasyfikacyjnym, a drzewem regresyjnym wynika z wartości, które są w liściach. Drzewo klasyfikacyjne przydzieli nam próbkę testową do odpowiedniej klasy, dziedziny. Zaś z kolei, drzewo regresyjne będzie potrafiło wskazać wartość ciągłą jako odpowiedź. W naszym problemie musimy wskazać ilość przestępstw na 100k mieszkańców. Do tego zadania o wiele lepiej sprawdzi się drzewo regresyjne, ponieważ przewidywane wartości znajdują się w przedziale ciągłym [0..1].

Skorzystaliśmy z podstawowej wersji drzewa regresyjnego dostarczonego przez bibliotekę [scikit-learn](#_Scikit-learn). Nie zmienialiśmy żadnych domyślnych parametrów.

Moduł: sklearn.tree.DecisionTreeRegressor

### Regresja liniowa

Regresja liniowa jest próbą modelowania związku między dwiema zmiennymi poprzez dopasowanie równania liniowego do zaobserwowanych danych. Jedna zmienna jest uważana za zmienną objaśniającą, a druga za zmienną zależną.

Skorzystaliśmy z podstawowej wersji regresji liniowej dostarczonego przez bibliotekę [scikit-learn](#_Scikit-learn). Nie zmienialiśmy żadnych domyślnych parametrów.

Moduł: sklearn.linear\_model.LinearRegression

### Regresja nie-liniowa

W statystyce regresja nieliniowa jest formą analizy regresji, w której dane obserwacyjne są modelowane za pomocą funkcji, która jest nieliniową kombinacją parametrów modelu i zależy od jednej lub więcej zmiennych niezależnych. Dane są dopasowywane za pomocą metody kolejnych przybliżeń.

Skorzystaliśmy z podstawowej wersji regresji nieliniowej dostarczonego przez bibliotekę [scikit-learn](#_Scikit-learn). Nie zmienialiśmy żadnych domyślnych parametrów.

Moduł: sklearn.svm.SVR

### Losowy las regresyjny

Las losowy jest meta estymatorem, który dopasowuje pewną liczbę klasyfikujących drzew decyzyjnych na różnych podpróbkach zbioru danych i wykorzystuje uśrednianie w celu poprawy dokładności predykcyjnej i kontroli nadmiernego dopasowania.

Parametry metodą brute-force, tak aby wyniki były jak najlepsze.

1. n\_estimators – Liczba drzew w lesie. **Wartość=**2000
2. min\_samples\_split – Minimalna liczba próbek wymagana do rozdzielenia węzła wewnętrznego. **Wartość=**2
3. min\_samples\_leaf – Minimalna liczba próbek wymagana, aby znaleźć się w węźle liścia. **Wartość=**2
4. max\_features – Liczba cech, które należy wziąć pod uwagę przy poszukiwaniu najlepszego rozdzielenia. "sqrt", wtedy max\_features=sqrt(n\_features). **Wartość=**’sqrt’
5. max\_depth – Maksymalna głębokość drzewa. **Wartość=**20
6. bootstrap – Czy próbki bootstrapowe są używane podczas budowania drzew. Jeśli False, do budowy każdego drzewa używany jest cały zbiór danych. **Wartość=**True

Moduł: sklearn.ensemble.RandomForestRegressor

## Ocena jakości modelu

Do oceny jakości modeli, z których korzystaliśmy w trakcie badania, posłużyliśmy się następującymi technikami.

### Cross validation

W walidacji krzyżowej, każdy rekord jest używany taką samą liczbę razy do szkolenia i dokładnie raz do testowania. Na przykład, w metodzie 2-krotnej walidacji krzyżowej wybierz jeden z podzbiorów do szkolenia, a drugi do testowania. Następnie należy zamienić role podzbiorów tak, aby poprzedni zbiór treningowy stał się zbiorem testowym i odwrotnie. W przypadku naszej analizy zastosowaliśmy metodę 10-krotnej walidacji krzyżowej, co powinno wykluczyć możliwość nadmiernego dopasowania się modelu do danych, przetrenowania.

W naszym podejściu postanowiliśmy wykorzystać moduł dostarczony przez [scikit-learn](#_Scikit-learn), cross\_val\_score. Metoda ta pozwala na przetestowanie różnych sposobów oceniania sprawności naszego modelu [13].

#### R2-scoring [14]

Funkcja r2\_score oblicza współczynnik determinacji, zwykle oznaczany jako R².

Jeśli jest przewidywaną wartością **i**-tej próbki, a jest odpowiadającą jej wartością prawdziwą, wówczas błąd średniokwadratowy (MSE) oszacowany na próbek definiuje się jako

Gdzie i .

#### Mean squared error (MSE) [15]

Funkcja mean\_squared\_error oblicza błąd średniokwadratowy, metrykę ryzyka odpowiadającą wartości oczekiwanej kwadratu (kwadratowego) błędu lub straty.

Jeśli jest przewidywaną wartością **i**-tej próbki, a jest odpowiadającą jej wartością prawdziwą, wówczas błąd średniokwadratowy (MSE) oszacowany na definiuje się jako

# Rezultat

W trakcie trwania naszego eksperymentu, za pomocą algorytmu *Future selection*, wybraliśmy 31 atrybutów, które posłużyły jako zbiór danych **df\_cleaned\_filtered**.

'FemalePctDiv' 'MalePctDivorce' 'NumIlleg' 'NumUnderPov' 'PctFam2Par'

'PctHousLess3BR' 'PctHousNoPhone' 'PctHousOwnOcc' 'PctIlleg'

'PctKids2Par' 'PctLess9thGrade' 'PctNotHSGrad' 'PctPersDenseHous'

'PctPersOwnOccup' 'PctPopUnderPov' 'PctTeen2Par' 'PctUnemployed'

'PctVacantBoarded' 'PctYoungKids2Par' 'TotalPctDiv' 'blackPerCap'

'medFamInc' 'medIncome' 'pctWInvInc' 'pctWPubAsst' 'racePctWhite'

'racepctblack'

## Sieć neuronowa

Wykres 1: Rezultat MSE Sieć neuronowa. 10-krotna walidacja krzyżowa.

Wyniki naszego eksperymentu przy użyciu modelu sieci neuronowej wskazują na to, że najlepszym zbiorem danych do predykcji argumentu decyzyjnego jest tutaj zbiór **df\_mean**. Niemniej, warto zwrócić również uwagę na to, że zbiór **df\_cleaned\_filtered** osiągnął bardzo podobne wyniki, a posiada znacznie mniejszy wymiar, przez co czas tworzenia modelu i predykcji jest znacząco mniejszy.

Wykres 2: Rezultat R2 Sieć neuronowa. 10-krotna walidacja krzyżowa.

Współczynnik determinacji jednoznacznie wskazuje na przewagę pozostałych zbiorów nad zbiorem **df\_cleaned\_filtered**. Niemniej metoda R2 jest uznawana tutaj za tą miarę pomocniczą.

## Drzewo regresyjne

Wykres 3: Rezultat MSE Drzewo regresyjne. 10-krotna walidacja krzyżowa.

Wynik zbioru **df\_cleaned\_filtered** posiada wyraźną przewagę jednocześnie będąc znacząco prostszym modelem. Natomiast pozostałe zbiory mają bardzo zbliżone wyniki, co indykuje, że metoda feature selection w drzewach regresyjnych może przynosić zauważalne korzyści.

Wykres 4: Rezultat R2 Drzewo regresyjne. 10-krotna walidacja krzyżowa.

Na powyższym wykresie znów możemy zauważyć znaczącą przewagę zbioru **df\_cleaned\_filtered**. Ciekawy jest również wynik **df\_med**, który posiada lepszy wynik niż **df\_mean**. Może być to kierować ku stwierdzeniu iż zastąpienie danych medianą jest lepsze niż średnią, lecz wymagało by to głębszej analizy.

## Regresja liniowa

Wykres 5: Rezultat MSE Regresja liniowa. 10-krotna walidacja krzyżowa.

Powyższy wykres nie pozwala nam stwierdzić istotnych różnic między zbiorami**. Df\_cleaned** posiada marginalnie niższy błąd. Mimo to wynik **df\_cleaned\_filtered** jest zadowalający biorąc pod uwagę ponad 3-krotne zmniejszenie liczby potrzebnych atrybutów.

Wykres 6: Rezultat R2 Regresja liniowa. 10-krotna walidacja krzyżowa.

Współczynnik determinacji pokrywa się z wynikami błędu.

## Regresja nieliniowa

Wykres 7: Rezultat MSE Regresja nieliniowa. 10-krotna walidacja krzyżowa.

Przy skorzystaniu z regresji nie-liniowej **df\_cleaned\_filtered** uzyskał najniższy błąd, jednak różnica pomiędzy najgorszym a najlepszym wynikiem to zaledwie 0.0005.

Wykres 8: Rezultat R2 Regresja nieliniowa. 10-krotna walidacja krzyżowa.

Współczynnik determinacji również pokazuje przewagę metody feature selection, jednakże różnice w wynikach są niewielkie.

## Losowy las regresyjny

Wykres 9: Rezultat MSE Losowy las regresyjny. 10-krotna walidacja krzyżowa.

Przy użyciu losowego lasu regresyjnego możemy zauważyć, że **df\_cleaned\_filtered** poradził sobie najgorzej. Przy reszcie zbiorów nie widać znacznej statystycznie różnicy.

Wykres 10: Rezultat R2 Losowy las regresyjny. 10-krotna walidacja krzyżowa.

Współczynnik determinacji nie wskazuje na żadne nowe wnioski.

## Porównanie

Wykres 11: Porównanie wyników miarą MSE

Wykres 12: Porównanie wyników miarą R2

Przy porównaniu błędu oraz współczynnika determinacji pomiędzy algorytmami regresyjnymi możemy zauważyć, że drzewa regresyjne najgorzej poradziły sobie z problemem. Sieci neuronowe były najbardziej czułe na sposób przygotowania danych. Regresja liniowa, nie-liniowa oraz las drzew regresyjnych osiągnęły podobne wyniki.

# Wnioski

Badanie miało na celu podjęcie próby znalezienia algorytmu, który najlepiej sprawdzi się przy opracowywaniu modelu do predykcji skali przestępczości, znając jego dane demograficzne. Po dokonaniu analizy miar R2 i MSE każdego algorytmu z wykorzystaniem różnych sposobów przygotowania danych doszliśmy do wniosku, iż Losowy Las Regresyjny z użyciem danych **df\_cleaned** czyli, po usunięciu danych z dużą ilością brakujących wartości. Dobre wyniki dały również algorytmy regresji linowej oraz nie-liniowej przy dużo szybszym czasie wykonania.

# Źródła

|  |  |
| --- | --- |
| [1] | [Online]. Available: https://en.wikipedia.org/wiki/Scikit-learn. |
| [2] | [Online]. Available: https://scikit-learn.org/stable/modules/generated/sklearn.neural\_network.MLPRegressor.html#sklearn.neural\_network.MLPRegressor. |
| [3] | [Online]. Available: https://scikit-learn.org/stable/modules/generated/sklearn.tree.DecisionTreeRegressor.html. |
| [4] | [Online]. Available: https://scikit-learn.org/stable/modules/generated/sklearn.linear\_model.LinearRegression.html. |
| [5] | [Online]. Available: https://scikit-learn.org/stable/modules/generated/sklearn.svm.SVR.html?highlight=svr#sklearn.svm.SVR. |
| [6] | [Online]. Available: https://scikit-learn.org/stable/modules/generated/sklearn.model\_selection.cross\_val\_score.html. |
| [7] | [Online]. Available: https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.RandomForestRegressor.html?highlight=randomforestregressor#sklearn.ensemble.RandomForestRegressor. |
| [8] | [Online]. Available: https://scikit-learn.org/stable/modules/generated/sklearn.model\_selection.RandomizedSearchCV.html?highlight=randomizedsearchcv#sklearn.model\_selection.RandomizedSearchCV. |
| [9] | [Online]. Available: https://en.wikipedia.org/wiki/NumPy. |
| [10] | [Online]. Available: https://en.wikipedia.org/wiki/Pandas\_(software). |
| [11] | [Online]. Available: https://scikit-learn.org/stable/modules/generated/sklearn.feature\_selection.f\_regression.html#sklearn.feature\_selection.f\_regression. |
| [12] | [Online]. Available: https://scikit-learn.org/stable/modules/generated/sklearn.feature\_selection.mutual\_info\_regression.html. |
| [13] | [Online]. Available: https://scikit-learn.org/stable/modules/model\_evaluation.html. |
| [14] | [Online]. Available: https://scikit-learn.org/stable/modules/generated/sklearn.metrics.r2\_score.html#sklearn.metrics.r2\_score. |
| [15] | [Online]. Available: https://scikit-learn.org/stable/modules/generated/sklearn.metrics.mean\_squared\_error.html#sklearn.metrics.mean\_squared\_error. |