Fundusze Europejskie dla Rozwoju Społecznego | Rzeczpospolita Polska | Dofinansowane przez Unię Europejską

	Nr projektu	FERS.01.05-IP.08-0335/23
	Tytuł projektu	„STUDENCI HIPOKRATESA- kompleksowy program utworzenia i wdrożenia kierunku lekarskiego na Politechnice Bydgoskiej”
	Beneficjent:	Politechnika Bydgoska im. Jana i Jędrzeja Śniadeckich

Projekt pt.: „STUDENCI HIPOKRATESA - kompleksowy program utworzenia i wdrożenia kierunku lekarskiego na Politechnice Bydgoskiej” w ramach programu Fundusze Europejskie dla Rozwoju Społecznego 2021-2027 współfinansowanego ze środków Europejskiego Funduszu Społecznego Plus, nr umowy: FERS.01.05-IP.08-0335/23-00

Przedmiot: Analiza danych medycznych

Forma zajęć: Ćwiczenia

Instrukcja

POLITECHNIKA BYDGOSKA
im. Jana i Jędrzeja Śniadeckich

Fundusze Europejskie dla Rozwoju Społecznego
Rzeczpospolita Polska
Dofinansowane przez Unię Europejską

Ćwiczenie 1

Podstawy statystyki medycznej

Cel:

Celem tego ćwiczenia jest przypomnienie kluczowych pojęć statystycznych, nabycie umiejętności obliczania podstawowych miar statystycznych oraz wizualizacji wyników. Praktyczna analiza danych medycznych.

1. Przygotowanie danych pacjentów:
    - Pobierz udostępniony przez prowadzącego zestaw danych dotyczących pacjentów (plik CSV zawierający dane demograficzne pacjentów, m.in.: wiek, płeć, masa ciała, wzrost, ciśnienie tętnicze, BMI).
    - Wczytaj dane do wybranego programu analitycznego:
        - Excel: Otwórz plik i korzystaj z wbudowanych funkcji statystycznych.
        - R lub Python (opcjonalnie): Zaimportuj dane za pomocą odpowiednich bibliotek (np. pandas dla Pythona, read.csv() dla R).
2. Obliczanie miar statystycznych. Wykonaj następujące obliczenia:
    - Średnia arytmetyczna i mediana dla wybranych zmiennych (np. wiek, masa ciała). Zinterpretuj różnice między nimi.
    - Odchylenie standardowe i wariancja. Miary pozycyjne (kwartyle i percentyle) dla rozkładu wieku. Zakres (min, max) dla każdej zmiennej. Jakie wnioski można wyciągnąć na temat zróżnicowania tej grupy?
3. Analiza rozkładu danych:
    - Stwórz histogramy, wykresy słupkowe oraz wykresy pudełkowe (boxploty) dla zmiennych ilościowych. Jakie są różnice w rozkładzie BMI między mężczyznami a kobietami?
    - Analizują kształt rozkładu (symetryczny, skośny, wielomodalny). Czy wiek pacjentów ma rozkład normalny? Czy rozkład zmiennych różni się między grupami (np. płci)?
    - Instrukcja techniczna (w zależności od narzędzia):
        - Excel: Użyj funkcji "Wstaw wykres" i wybierz odpowiedni typ wykresu.
        - R: Skorzystaj z funkcji hist(), boxplot() i barplot().
        - Python: Użyj bibliotek takich jak matplotlib lub seaborn (np. sns.boxplot()).
4. Interpretacja wyników. Przygotuj krótkie odpowiedzi na pytania:
    - Jakie są kluczowe cechy statystyczne analizowanych danych?
    - Jakie różnice można zauważyć między grupami (np. różnice w średnim BMI między płciami)?
    - Jakie potencjalne błędy mogą wystąpić podczas analizy takich danych?
    - Jakie potencjalne problemy mogą pojawić się podczas analizy realnych zestawów danych (real-world datasets)?

Ćwiczenie 2

Analiza epidemiologiczna

Cel:

Celem tego ćwiczenia jest zrozumienie i zastosowanie wskaźników epidemiologicznych (zachorowalności, umieralności, przeżywalności) do analizy występowania chorób w populacji. Nauka interpretacji wskaźników i wyciągania wniosków w kontekście zdrowia publicznego.

1. Przygotowanie danych pacjentów:
   - Pobierz udostępniony przez prowadzącego zestaw danych dotyczących pacjentów (plik CSV zawierający dane demograficzne pacjentów, m.in.: wiek, płeć, liczba nowych przypadków choroby w określonym czasie, liczba zgonów związanych z chorobą, wielkość populacji badanej).
   - Wczytaj dane do wybranego programu analitycznego:
     - Excel: Otwórz plik i korzystaj z wbudowanych funkcji statystycznych.
     - R lub Python (opcjonalnie): Zaimportuj dane za pomocą odpowiednich bibliotek (np. pandas dla Pythona, read.csv() dla R).
2. Obliczanie wskaźników epidemiologicznych. Oblicz wskaźniki:
   - Wskaźnik zachorowalności (ang. *incidence rate*):

$$IR = \frac{Liczba\ nowych\ przypadków\ choroby}{Wielkość\ populacji * Czas\ obserwacji}$$

   - Wskaźnik umieralności (ang. *mortality rate*):

$$MR = \frac{Liczba\ zgonów\ z\ powodu\ choroby}{Wielkość\ populacji * Czas\ obserwacji}$$

   - Wskaźnik przeżywalności (ang. *survival rate*):

$$SR = \frac{Liczba\ osób, które\ przeżyły}{Wielkość\ populacji * Czas\ obserwacji}$$

   - Oblicz wartości tych wskaźników dla różnych podgrup (np. wiekowych, płciowych).
   - Porównaj wskaźniki zachorowalności i umieralności w populacjach miejskich i wiejskich.
3. Przygotuj wykresy:
   - Histogramy lub wykresy słupkowe pokazujące różnice w wskaźnikach między grupami (np. płeć, wiek).
   - Wykresy liniowe przedstawiające trendy czasowe w zachorowalności i umieralności.
4. Zinterpretuj uzyskane wyniki analiz. W tym celu możesz odpowiedzieć na następujące pytania:
   - Jakie są kluczowe różnice w wskaźnikach między grupami?
   - Czy widać jakieś niepokojące trendy w analizowanych danych (np. wzrost zachorowalności)?
   - Jakie mogą być potencjalne przyczyny tych różnic?

Fundusze Europejskie dla Rozwoju Społecznego | Rzeczpospolita Polska | Dofinansowane przez Unię Europejską

Ćwiczenie 3

Analiza przeżywalności

Cel:

Zrozumienie i zastosowanie metod analizy przeżycia, takich jak krzywe Kaplan-Meiera, oraz interpretacja czasu do zdarzenia w kontekście medycznym.

1. Przygotuj dane. Wczytaj dostarczony przez prowadzącego zestaw danych dotyczących pacjentów, zawierający zmienne, m.in.: identyfikator pacjenta, czas obserwacji (w miesiącach lub latach), status (0 = brak zdarzenia, 1 = zdarzenie, np. zgon), zmienna prognostyczna (np. wiek, płeć, rodzaj leczenia). W tym celu zaimportuj dane do wybranego narzędzia analitycznego:
    - Excel: Użyj odpowiednich funkcji tabelarycznych.
    - R: Użyj pakietu survival (np. funkcja Surv()).
    - Python: Wczytaj dane za pomocą pandas i użyj pakietu lifelines.
2. Konstrukcja krzywych przeżycia Kaplan-Meiera:
    - Podziel pacjentów na grupy według wybranej zmiennej prognostycznej (np. grupy wiekowe, rodzaj leczenia).
    - Skonstruuj krzywe przeżycia Kaplan-Meiera dla każdej grupy:
    - Excel: Użyj narzędzi do obliczania prawdopodobieństwa przeżycia w kolejnych przedziałach czasowych.
    - R: Wykorzystaj funkcje survfit() i ggsurvplot().
    - Python: Użyj funkcji KaplanMeierFitter() z pakietu lifelines.
3. Wizualizacja wyników. Stwórz wykresy krzywych przeżycia dla poszczególnych grup. Oznacz na wykresie osie: czas (oś X) oraz prawdopodobieństwo przeżycia (oś Y). Uwzględnij różne kolory lub style linii dla grup, aby ułatwić ich porównanie. Dodaj istotne elementy wykresu, np. legendę wyjaśniającą podział na grupy. Przy generowaniu wykresów uwzględnij aspekty estetyczne, takie jak czytelne opisy osi i legendy.
4. Porównaj krzywe Kaplan-Meiera między grupami za pomocą testu log-rank:
    - R: Użyj funkcji survdiff().
    - Python: Skorzystaj z logrank_test() w pakiecie lifelines.
    - Excel: Oblicz różnice ręcznie, korzystając z odpowiednich danych tabelarycznych.
    - Określ, czy istnieje istotna statystycznie różnica między krzywymi.
5. Zinterpretuj różnice w krzywych przeżycia:
    - Która grupa ma najwyższe prawdopodobieństwo przeżycia po określonym czasie?
    - Czy zmienna prognostyczna istotnie wpływa na przeżycie pacjentów?
    - Omów potencjalne przyczyny zaobserwowanych różnic.
    - Zidentyfikuj ograniczenia przeprowadzonej analizy.

Fundusze Europejskie dla Rozwoju Społecznego
Rzeczpospolita Polska
Dofinansowane przez Unię Europejską

Ćwiczenie 4

Analiza zmiennych kategorycznych

Cel:

Celem tego ćwiczenia jest nauczenie się analizy danych kategorycznych z wykorzystaniem testów statystycznych (test chi-kwadrat i test Fishera) oraz interpretacja wyników w kontekście medycznym.

1. Przygotuj dane. Wczytaj dostarczony przez prowadzącego zestaw danych dotyczących pacjentów, zawierający zmienne, m.in.: płeć, występowanie określonego objawu (np. obecność/nieobecność), rodzaj leczenia, wynik leczenia (np. sukces/niepowodzenie). W tym celu zaimportuj dane do wybranego narzędzia analitycznego:
    - Excel: Wczytaj dane do arkusza kalkulacyjnego.
    - R: Użyj funkcji read.csv() lub read.table().
    - Python: Skorzystaj z biblioteki pandas (pd.read_csv()).
2. Budowa tabeli kontyngencji:
    - Stwórz tabelę kontyngencji, która przedstawia liczbę obserwacji dla kombinacji zmiennych kategorycznych, np.: rząd: Rodzaj leczenia (np. lek A, lek B), kolumna: Wynik leczenia (np. sukces, niepowodzenie).
    - Upewnij się, że tabela jest poprawnie skonstruowana i obejmuje wszystkie możliwe kombinacje zmiennych.
3. Zastosuj test chi-kwadrat, aby zbadać, czy istnieje zależność między wybranymi zmiennymi:
    - Excel: Użyj funkcji CHI.TEST do obliczenia statystyki testu.
    - R: Użyj funkcji chisq.test().
    - Python: Użyj funkcji chi2_contingency() z pakietu scipy.
    - Zwróć uwagę na: statystykę chi-kwadrat, stopnie swobody, wartość p (p-value). Zapisz wynik i zinterpretuj go. Jeśli wartość $p < 0,05$, istnieje statystycznie istotna zależność między zmiennymi.
4. W przypadku małych próbek zastosuj test Fishera:
    - Excel: Użyj zewnętrznego kalkulatora online (jeśli nie masz odpowiedniego dodatku).
    - R: Użyj funkcji fisher.test().
    - Python: Użyj funkcji fisher_exact() z pakietu scipy.
    - Zapisz wartość p i zinterpretuj wynik w kontekście medycznym.
5. Przedstaw wyniki w formie graficznej:
    - Wykres słupkowy przedstawiający proporcje sukcesów i niepowodzeń dla każdego rodzaju leczenia.
    - Dodaj opisy osi, tytuł wykresu i legendę, aby wykres był czytelny.
6. Zinterpretuj wyniki. Odpowiedz na pytania:
    - Czy rodzaj leczenia ma wpływ na wynik (np. sukces/niepowodzenie)?
    - Jakie są ograniczenia testów statystycznych w analizie zmiennych kategorycznych?
    - Porównaj wyniki testu chi-kwadrat i testu Fishera (jeśli oba były wykonane).

Ćwiczenie 5

Analiza regresji

Cel:

Celem tego ćwiczenia jest poznanie i zastosowanie metod regresji liniowej i logistycznej w analizie danych medycznych, interpretacja wyników w kontekście klinicznym.

1. Wczytaj dostarczony zestaw danych zawierający zmienne predykcyjne i zmienne zależne:
   - Zmienne predykcyjne: wiek, płeć, BMI, wynik badania laboratoryjnego.
   - Zmienna zależna dla regresji liniowej: poziom glukozy we krwi.
   - Zmienna zależna dla regresji logistycznej: obecność/nieobecność choroby (0 = brak, 1 = obecność).
2. Importuj dane do wybranego narzędzia (Excel, R, Python).
3. Stwórz model regresji liniowej, w którym zmienną zależną jest poziom glukozy, a predyktorami są wiek, BMI i płeć:
   - Excel: Skorzystaj z dodatku „Analiza danych” i opcji regresji.
   - R: Użyj funkcji lm() (np. lm(glucose ~ age + BMI + sex)).
   - Python: Użyj biblioteki statsmodels (np. ols() z modułu statsmodels.formula.api).
4. Odczytaj kluczowe wyniki modelu: współczynniki regresji dla każdego predyktora, wartość R^2, wartość p dla każdego predyktora.
   - Zidentyfikuj, które predyktory są istotne statystycznie ($p < 0,05$),
   - Zinterpretuj kierunek wpływu (dodatni/ujemny) każdego predyktora na zmienną zależną.
5. Stwórz model regresji logistycznej, w którym zmienną zależną jest obecność choroby (0/1), a predyktorami są wiek, BMI i płeć:
   - Excel: Skorzystaj z zaawansowanych dodatków lub zewnętrznych kalkulatorów.
   - R: Użyj funkcji glm() (np. glm(disease ~ age + BMI + sex, family = binomial)).
   - Python: Użyj LogisticRegression z pakietu sklearn lub logit() z statsmodels.
6. Odczytaj kluczowe wyniki modelu: współczynniki regresji (log-odds), wartość p dla każdego predyktora, miary dopasowania modelu (np. AUC, pseudo R^2).
   - Oceń, które zmienne predykcyjne są istotne statystycznie,
   - Przekształć współczynniki log-odds na ryzyko względne (odds ratio).
7. W celu walidacji przygotowanych modeli sprawdź, czy model jest dobrze dopasowany do danych:
   - W przypadku regresji liniowej: zweryfikuj rozkład reszt (np. histogram reszt, wykres reszt w zależności od przewidywanych wartości).
   - W przypadku regresji logistycznej: wygeneruj krzywą ROC i oblicz pole pod krzywą (AUC).
   - Oceń, czy model wymaga ulepszenia (np. dodania nowych predyktorów lub zmiany sposobu transformacji danych).
8. Przygotuj wykresy:
   - Wykres punktowy z linią regresji (dla regresji liniowej),
   - Wykresy ilustrujące przewidywane prawdopodobieństwa dla różnych grup (dla regresji logistycznej).
   - Dodaj opisy osi, tytuły i legendy, aby wykresy były czytelne.
9. Zastanów się i zinterpretuj otrzymane wyniki. Poniższe pytania mogą ułatwić ci interpretację wyników:
   - Które zmienne mają największy wpływ na zmienną zależną?
   - Jak można wykorzystać wyniki analizy w praktyce klinicznej?

- Jakie są ograniczenia zastosowanych modeli regresyjnych?
- Omów znaczenie uzyskanych wyników w kontekście problemu medycznego,
- Zaproponuj możliwości ulepszenia modelu oraz potencjalne zastosowania wyników.

Fundusze Europejskie dla Rozwoju Społecznego
Rzeczpospolita Polska
Dofinansowane przez Unię Europejską

Ćwiczenie 6

Analiza danych wielowymiarowych

Cel:

Celem tego ćwiczenia jest poznanie i zastosowanie metod analizy wielowymiarowej, takich jak analiza skupień i analiza głównych składowych (PCA), do grupowania i redukcji wymiarowości danych medycznych.

1. Wczytaj dostarczony zestaw danych zawierający wiele zmiennych dla pacjentów, np.: wyniki badań laboratoryjnych (np. poziomy glukozy, cholesterolu, kreatyniny), wiek, płeć, wskaźnik BMI, informacje o stanie zdrowia (np. ciśnienie krwi, obecność chorób przewlekłych).
2. Wczytaj dane do wybranego narzędzia analitycznego (Excel, R, Python).
3. Wykonaj analizę skupień (klasyczna i hierarchiczna). Przygotuj dane:
    - Znormalizuj dane (np. sprowadź zmienne do skali [0, 1] lub standaryzuj z użyciem średniej i odchylenia standardowego).
    - W R: Użyj funkcji scale().
    - W Python: Skorzystaj z StandardScaler() z pakietu sklearn.
4. Klasyczna analiza skupień:
    - Zastosuj algorytm k-średnich (k-means) w celu podziału pacjentów na grupy.
    - W R: Użyj funkcji kmeans().
    - W Python: Użyj KMeans z sklearn.
    - W Excelu: Wykorzystaj narzędzia analizy danych zewnętrznych, jeśli dostępne.
    - Eksperymentuj z różną liczbą skupień (k) i wybierz optymalne k, korzystając z metody „łokcia” (elbow method).
5. Hierarchiczna analiza skupień:
    - Stwórz dendrogram, aby wizualizować hierarchię grupowania.
        - W R: Użyj funkcji hclust() i dendrogram().
        - W Python: Skorzystaj z scipy.cluster.hierarchy.
6. Analiza głównych składowych (PCA)
    - Przygotowanie danych:
    - Znormalizuj dane (jeśli nie zrobiłeś tego wcześniej).
    - Wykonanie PCA:
    - Oblicz główne składowe, które wyjaśniają największą część wariancji w danych.
        - W R: Użyj funkcji prcomp().
        - W Python: Użyj PCA z sklearn.
    - Zidentyfikuj liczbę komponentów, które wyjaśniają większość wariancji (np. >80%).
7. Wizualizacja wyników:
    - Utwórz wykres biplot, aby przedstawić pierwsze dwie główne składowe i ich relacje do zmiennych oryginalnych.
        - W R: Użyj funkcji fviz_pca_biplot() z pakietu factoextra.
        - W Python: Narysuj wykres z użyciem matplotlib lub seaborn.
8. Interpretacja wyników dla analizy skupień:
    - Opisz charakterystykę grup pacjentów w poszczególnych skupieniach (np. grupy pacjentów z podobnymi profilami zdrowotnymi).
    - Oceń, jakie cechy różnicują grupy (np. poziom cholesterolu, wiek).
9. Interpretacja wyników dla PCA: Wyjaśnij, jakie zmienne miały największy wpływ na pierwsze dwie główne składowe. Oceń, jak zredukowana liczba zmiennych może uprościć analizę danych bez utraty kluczowych informacji. Zinterpretuj wyniki w kontekście danych

medycznych, wskaż potencjalne zastosowania wyników w praktyce klinicznej (np. grupowanie pacjentów z podobnymi potrzebami terapeutycznymi).

Fundusze Europejskie dla Rozwoju Społecznego | Rzeczpospolita Polska | Dofinansowane przez Unię Europejską

Ćwiczenie 7

Analiza danych genetycznych w medycynie

Cel:

Celem tego ćwiczenia jest poznanie metod analizy danych genetycznych, takich jak analiza SNP (ang. *single nucleotide polymorphism*), analiza ekspresji genów oraz interpretacja wyników w kontekście ryzyka wystąpienia chorób dziedzicznych.

1. Wczytaj dostarczony zestaw danych genetycznych zawierający: genotypy SNP (np. AA, AG, GG), ekspresję genów (wartości RPKM lub TPM dla różnych genów), fenotypy pacjentów (np. obecność choroby, wiek, płeć).
2. W celu analizy SNP:
   - Zidentyfikuj zmienne odpowiadające genotypom SNP.
   - Podziel dane na grupy według genotypu (np. homozygoty dominujące, heterozygoty, homozygoty recesywne).
3. Oblicz częstości występowania każdego allelu w populacji zgodnie z poniższym przykładem:

$$Częstość\ allelu\ A = \frac{2 * liczba\ AA + liczba\ AG}{Liczba\ wszystkich\ próbek}$$

   - W Excelu: Wykorzystaj funkcje COUNTIF() do zliczania genotypów.
   - W R lub Python: Użyj funkcji agregujących.
4. Analiza asocjacyjna:
   - Wykonaj test statystyczny (np. test chi-kwadrat lub regresję logistyczną) w celu zbadania związku między SNP a fenotypem (np. obecność choroby):
   - R: Użyj funkcji chisq.test() lub glm().
   - Python: Użyj funkcji chi2_contingency() z scipy lub LogisticRegression z sklearn.
5. Analiza ekspresji genów. Przygotowanie danych:
   - Wybierz zmienne odpowiadające ekspresji wybranych genów.
   - Znormalizuj dane (np. logarytmuj wartości, aby zredukować wpływ dużych odchyleń).
6. Analiza różnic w ekspresji. Porównaj poziomy ekspresji genów między grupami (np. osoby zdrowe vs chore):
   - W Excelu: Wykorzystaj test t-Studenta lub ANOVA (jeśli więcej grup).
   - W R: Użyj t.test() lub aov().
   - W Python: Skorzystaj z funkcji ttest_ind() z scipy lub anova_lm() z statsmodels.
7. Utwórz wykresy pudełkowe (boxplot) lub wykresy słupkowe, aby porównać ekspresję genów między grupami.
8. Identyfikacja genów ryzyka:
   - Zastosuj analizę wielowymiarową (np. regresję logistyczną lub PCA), aby zidentyfikować geny najbardziej związane z ryzykiem choroby.
   - Oceń, które geny są statystycznie istotne i jakie mają znaczenie biologiczne.
9. Zinterpretuj uzyskane wyniki analiz i odpowiedz na pytania:
   - Czy istnieje związek między analizowanymi SNP a fenotypem?
   - Które geny wykazują różnice w ekspresji między grupami?
   - Jakie wyniki mogą sugerować potencjalne biomarkery dla danej choroby?
   - Omów, które SNP i geny mogą być związane z ryzykiem choroby,
   - Zidentyfikuj ograniczenia analizy i zaproponuj dalsze kroki badawcze.

Fundusze Europejskie dla Rozwoju Społecznego
Rzeczpospolita Polska
Dofinansowane przez Unię Europejską

Ćwiczenie 8

Sztuczna inteligencja w medycynie

Cel:

Celem tego ćwiczenia jest wprowadzenie do zastosowania sztucznej inteligencji (SI) w analizie danych medycznych, w tym klasyfikacji danych oraz predykcji wyników klinicznych, z wykorzystaniem prostych modeli uczenia maszynowego.

1. Wczytaj dostarczony zestaw danych zawierający: dane demograficzne pacjentów (np. wiek, płeć), wyniki badań laboratoryjnych (np. poziom glukozy, kreatyniny, ciśnienie krwi), informację o stanie zdrowia (np. obecność/nieobecność choroby, wynik leczenia).
2. Przygotuj zmienne. Upewnij się, że dane są kompletne (brakujące wartości uzupełnij średnią, medianą lub usuń rekordy, jeśli to konieczne).
3. Podziel zmienne na:
    - Zmienną docelową (np. obecność choroby: 0 = brak, 1 = obecność),
    - Zmiennie predykcyjne (np. wiek, BMI, wyniki badań).
4. Podziel dane na zbiór treningowy (70–80%) i testowy (20–30%).
    - W Python: Użyj train_test_split() z sklearn.
    - W R: Użyj funkcji sample().
5. Przygotuj modele klasyfikacyjne:
6. Algorytm drzewa decyzyjnego:
    - W Python:
        - Importuj DecisionTreeClassifier z sklearn.tree.
        - Dopasuj model do zbioru treningowego (np. model.fit(X_train, y_train)).
        - Oceń model na zbiorze testowym (np. model.score(X_test, y_test)).
    - W R:
        - Użyj funkcji rpart() z pakietu rpart do budowy drzewa decyzyjnego.
7. Oblicz metryki ewaluacyjne modelu: dokładność, precyzję, czułość, miara F1.
8. Wygeneruj macierz pomyłek (ang. c*onfusion matrix*).
    - Python: confusion_matrix() z sklearn.
    - R: confusionMatrix() z pakietu caret.
9. Zwizualizuj drzewo decyzyjne:
    - W Python: Skorzystaj z plot_tree() z sklearn.
    - W R: Użyj rpart.plot().
10. Model regresji logistycznej:
    - Trenowanie modelu:
    - W Python:
        - Importuj LogisticRegression z sklearn.linear_model.
    - Dopasuj model do danych treningowych i oceń na danych testowych.
    - W R:
        - Użyj funkcji glm() z opcją family = binomial.
    - Ocena modelu:
        - Wygeneruj współczynniki regresji i oceń istotność zmiennych:
        - Python: Użyj model.coef_ i model.intercept_.
        - R: Skorzystaj z funkcji summary().
11. Oblicz miary dopasowania (np. AUC, accuracy).
12. Wizualizacja wyników:
    - Wykonaj wykres ROC i oblicz pole pod krzywą (AUC):
        - Python: roc_curve() i auc() z sklearn.metrics.

- R: roc() z pakietu pROC.

13. Walidacja na nowych danych:
    - Wczytaj nowy zestaw danych testowych i użyj wytrenowanego modelu do przewidywania wyników (np. prawdopodobieństwa obecności choroby).
    - Oceń, jak dobrze model działa na nowych danych.
14. Zinterpretuj uzyskane wyniki.
    - Jak dokładny jest Twój model? Który zbudowany model (drzewo decyzyjne czy regresja logistyczna) lepiej przewiduje wyniki i dlaczego?
    - Jakie zmienne predykcyjne mają największy wpływ na wynik?
    - Jakie są ograniczenia zastosowanych modeli?
    - Wskaż potencjalne zastosowania wyników w praktyce klinicznej.