**1. Opis zbioru danych**

Zbiór danych *Student Performance in Mathematics and Portuguese* (znany także jako *Student Alcohol Consumption*) został zebrany w dwóch portugalskich szkołach średnich. Dane dotyczą wyników uczniów z przedmiotów matematyka oraz język portugalski. Zestaw ten pozwala na analizę różnorodnych czynników (demograficznych, społeczno-ekonomicznych, behawioralnych) i ich potencjalnego wpływu na wyniki w nauce.

**1.1. Zakres informacji w zbiorze**

Zbiór danych zawiera szczegółowe informacje o uczniach, m.in.:

* **Dane demograficzne:**
  + Szkoła, płeć, wiek, adres (miejski/wiejski), liczba osób w rodzinie.
* **Warunki rodzinne:**
  + Status współzamieszkania rodziców, wykształcenie matki i ojca, zawód matki i ojca.
* **Wsparcie w nauce:**
  + Dodatkowe zajęcia w szkole (ang. *schoolsup*), wsparcie rodziców (ang. *famsup*), płatne korepetycje (ang. *paid*).
* **Aktywności pozalekcyjne:**
  + Udział w zajęciach dodatkowych (ang. *activities*), internet w domu, posiadanie romansu/stałego związku (ang. *romantic*).
* **Czas wolny i zachowania:**
  + Czas poświęcony na naukę (ang. *studytime*), czas dojazdu do szkoły (ang. *traveltime*), relacje rodzinne (ang. *famrel*), wolny czas (ang. *freetime*), wyjścia ze znajomymi (ang. *goout*), spożycie alkoholu w ciągu tygodnia (ang. *Dalc*) i w weekendy (ang. *Walc*).
* **Wyniki nauczania:**
  + Oceny z pierwszego, drugiego i trzeciego okresu (G1, G2, G3) oraz liczba nieobecności (ang. *absences*).

Dane zostały udostępnione w dwóch plikach CSV:

* **student-mat.csv** – wyniki z przedmiotu matematyka.
* **student-por.csv** – wyniki z przedmiotu język portugalski.

W przedstawionym kodzie skupiamy się na pliku *student-mat.csv* (matematyka).

**2. Potencjalne zastosowania**

1. **Analiza czynników wpływających na oceny uczniów**
   * Badanie zależności między wskaźnikami społeczno-ekonomicznymi (m.in. wykształcenie rodziców, zajęcia dodatkowe) a ocenami.
2. **Prognozowanie wyników**
   * Budowa modeli uczenia maszynowego, które przewidują ostateczną ocenę (G3) na podstawie wcześniejszych ocen (G1, G2) i innych czynników (np. ilość godzin nauki, aktywności dodatkowe, spożycie alkoholu).
3. **Ocena efektywności interwencji edukacyjnych**
   * Sprawdzenie, jak dodatkowe wsparcie (np. *schoolsup*, *famsup*) wpływa na postępy i ostateczne wyniki uczniów.
4. **Zrozumienie wpływu zachowań i środowiska rodzinnego**
   * Wpływ spożycia alkoholu, relacji rodzinnych, ilości czasu wolnego na osiągnięcia szkolne.

**3. Struktura i znaczenie najważniejszych kolumn**

Poniżej wymieniono część atrybutów z pliku *student-mat.csv*, wraz z interpretacją:

1. **school**
   * Szkoła ucznia (zmienna binarna: "GP" – Gabriel Pereira, "MS" – Mousinho da Silveira).
2. **sex**
   * Płeć ucznia (zmienna binarna: "F" – kobieta, "M" – mężczyzna).
3. **age**
   * Wiek ucznia (wartość liczbowa od 15 do 22).
4. **address**
   * Typ adresu ucznia (zmienna binarna: "U" – miasto, "R" – obszar wiejski).
5. **famsize**
   * Wielkość rodziny (zmienna binarna: "LE3" – rodzina ≤ 3 członków, "GT3" – rodzina > 3 członków).
6. **Pstatus**
   * Status zamieszkania rodziców (zmienna binarna: "T" – razem, "A" – oddzielnie).
7. **Medu** / **Fedu**
   * Wykształcenie matki (Medu) / ojca (Fedu) w skali 0–4 (np. 0 – brak, 1 – szkoła podstawowa, 4 – studia wyższe).
8. **Mjob** / **Fjob**
   * Zawód matki (Mjob) / ojca (Fjob) (kategorie nominalne: "teacher", "health", "services", "at\_home", "other").
9. **reason**
   * Powód wyboru szkoły (nominalna: "home", "reputation", "course", "other").
10. **guardian**

* Główny opiekun ucznia ("mother", "father", "other").

1. **traveltime**

* Czas dojazdu do szkoły w skali 1–4 (1 – <15 min, 2 – 15–30 min, 3 – 30–60 min, 4 – >60 min).

1. **studytime**

* Tygodniowy czas nauki własnej w skali 1–4 (1 – <2 godziny, 4 – >10 godzin).

1. **failures**

* Liczba niezdanych przedmiotów w przeszłości (liczba całkowita; np. 1<=n<3, w innym przypadku 4).

1. **schoolsup**

* Dodatkowe wsparcie w nauce w szkole (zmienna binarna: "yes"/"no").

1. **famsup**

* Wsparcie rodziny w nauce (zmienna binarna: "yes"/"no").

1. **paid**

* Płatne korepetycje (zmienna binarna: "yes"/"no").

1. **activities**

* Uczestnictwo w zajęciach dodatkowych (zmienna binarna: "yes"/"no").

1. **higher**

* Chęć kontynuacji nauki w szkołach wyższych (zmienna binarna: "yes"/"no").

1. **internet**

* Dostęp do Internetu w domu (zmienna binarna: "yes"/"no").

1. **romantic**

* Status romantyczny ucznia (zmienna binarna: "yes"/"no").

1. **famrel**

* Jakość relacji rodzinnych w skali 1–5.

1. **freetime**

* Ilość wolnego czasu po szkole w skali 1–5.

1. **goout**

* Częstotliwość wyjść ze znajomymi w skali 1–5.

1. **Dalc** / **Walc**

* Spożycie alkoholu: w dni powszednie (Dalc) / weekendy (Walc) w skali 1–5.

1. **health**

* Subiektywna ocena stanu zdrowia w skali 1–5.

1. **absences**

* Liczba nieobecności w szkole.

1. **G1** / **G2** / **G3**

* Oceny z pierwszego, drugiego i trzeciego okresu (0–20).
* **G3** to końcowa ocena, najczęściej wykorzystywana jako wartość docelowa w modelach predykcyjnych.

**4. Opis kodu i główne etapy przetwarzania**

Poniżej omówiono kluczowe fragmenty kodu w języku Python, który wykorzystuje bibliotekę **Streamlit** do tworzenia interaktywnej aplikacji webowej służącej do przewidywania końcowej oceny (G3).

**4.1. Importy i definicje**

python

Copy code

import streamlit as st

from model import Model

import pandas as pd

from sklearn.metrics import mean\_absolute\_error, mean\_squared\_error, r2\_score

from sklearn.model\_selection import train\_test\_split

from sklearn.ensemble import RandomForestRegressor

* **streamlit** – biblioteka do tworzenia aplikacji webowych w Pythonie.
* **pandas** – do zarządzania i przetwarzania danych w formacie tabelarycznym.
* **sklearn** – popularny pakiet do uczenia maszynowego (zawiera m.in. implementacje modeli, funkcji oceny, metod podziału zbioru na treningowy i testowy).

**4.2. Klasa Model**

python

Copy code

class Model:

def \_\_init\_\_(self):

self.model = RandomForestRegressor()

self.trained = False

self.data = pd.read\_csv('../data/student-mat.csv')

...

* **Model** – klasa opakowująca cały proces przygotowania, trenowania i predykcji.
* W konstruktorze wczytywany jest zbiór danych *student-mat.csv* oraz tworzony jest obiekt klasy RandomForestRegressor.

**4.2.1. Metoda load\_dataset()**

python

Copy code

def load\_dataset(self):

df = self.data

columns\_to\_keep = [...]

df = df[columns\_to\_keep]

df = df.dropna()

# Przekształcenie wartości tekstowych na typy logiczne

df['sex'] = df['sex'].map({'F': False, 'M': True})

df['address'] = df['address'].map({'U': False, 'R': True})

df['Pstatus'] = df['Pstatus'].map({'A': False, 'T': True})

...

self.data = df

return df

* Tutaj wybierane są tylko kluczowe kolumny, które będą wykorzystane w modelu.
* Usunięcie wartości pustych (*NaN*) za pomocą dropna().
* Mappowanie zmiennych kategorycznych (np. płci, typu adresu) na wartości boolowskie (True/False).

**4.2.2. Metoda train()**

python

Copy code

def train(self):

y = self.data['G3']

X = self.data.drop('G3', axis=1)

X\_train, X\_test, y\_train, y\_test = train\_test\_split(X, y, test\_size=0.2, random\_state=25)

self.model.fit(X\_train, y\_train)

self.trained = True

y\_pred = self.model.predict(X\_test)

mse = mean\_squared\_error(y\_test, y\_pred)

r2 = r2\_score(y\_test, y\_pred)

return r2

* Dzielenie danych na zbiór treningowy i testowy (proporcja 80:20).
* Trenowanie modelu RandomForestRegressor na danych treningowych.
* Obliczenie miar jakości (MSE, R^2) na zbiorze testowym.
* Zwracanie wartości R^2 (im wyższa, tym lepsze dopasowanie).

**4.2.3. Metoda predict(...)**

python

Copy code

def predict(self, input\_data):

if not self.trained:

raise ValueError("Model is not trained!")

prediction = self.model.predict(input\_data)

return prediction

* Wykonanie predykcji na podstawie przetworzonych danych wejściowych.
* Zwrócenie przewidywanej oceny końcowej (G3).

**4.3. Funkcja prepare\_model(model)**

python

Copy code

def prepare\_model(model):

model.load\_dataset()

model.train()

* Ładuje dane i trenuje model.

**4.4. Kod główny aplikacji Streamlit**

python

Copy code

ml\_model = Model()

prepare\_model(ml\_model)

...

st.title("Third year grade prediction")

...

if st.button("Make prediction"):

...

prediction = round(float(ml\_model.predict(input\_values)))

st.write(f"Predicted score: {prediction}")

* Tworzenie obiektu klasy Model, wczytanie i trenowanie modelu.
* Interfejs w **Streamlit** umożliwia użytkownikowi wprowadzanie parametrów (płeć, wiek, liczba nieobecności itp.) za pomocą widgetów (radio button, slider).
* Po kliknięciu przycisku „Make prediction” wywoływana jest metoda predict(...), a następnie wynik jest wyświetlany na stronie.

**5. Jak uruchomić aplikację?**

1. **Krok 1: Zainstaluj zależności**  
   Upewnij się, że masz zainstalowane biblioteki: *streamlit*, *pandas*, *scikit-learn*. Możesz to zrobić poleceniem:

bash

Copy code

pip install streamlit pandas scikit-learn

1. **Krok 2: Struktura plików**
   * Plik z kodem aplikacji (np. app.py).
   * Plik model.py zawierający definicję klasy Model.
   * Folder data z plikiem student-mat.csv.
2. **Krok 3: Uruchom aplikację**

bash

Copy code

streamlit run app.py

Aplikacja uruchomi się w przeglądarce pod adresem: http://localhost:8501.

1. **Krok 4: Wprowadzanie danych i uzyskanie przewidywań**
   * Uzupełnij dane w interfejsie (np. płeć, wiek, liczba nieobecności).
   * Kliknij „Make prediction”, aby zobaczyć szacowaną ocenę końcową (G3).

**6. Wnioski i dalsze kroki**

Dzięki temu projektowi można w prosty sposób:

* Zrozumieć, w jaki sposób różne cechy uczniów (m.in. miejsce zamieszkania, wsparcie rodziców, liczba nieobecności) mogą wpływać na osiągane wyniki.
* Zbudować narzędzie prognozujące wyniki (G3) na podstawie wprowadzonych parametrów.
* Sprawdzić skuteczność modelu (poprzez wartości *mean\_squared\_error* oraz *r2\_score*) i ewentualnie go ulepszyć (np. tuning hiperparametrów, wypróbowanie innych algorytmów).

Można rozszerzyć projekt o:

* Analizę korelacji między konkretnymi zmiennymi a ocenami.
* Uwzględnienie w modelu cech z pliku *student-por.csv*.
* Dodanie wizualizacji (np. wykresy korelacji, rozkładu ocen).
* Eksperymenty z innymi modelami regresyjnymi (np. *Gradient Boosting*, *Linear Regression*, *XGBoost*).

**7. Licencja i źródło danych**

* Zbiór danych jest dostępny na licencji **CC0: Public Domain**, co oznacza, że można go swobodnie wykorzystywać, modyfikować oraz rozpowszechniać.
* Dane pochodzą z badań przeprowadzonych w Portugalii (szkoły Gabriel Pereira i Mousinho da Silveira) i zostały zanonimizowane.

**Podsumowując**, zaprezentowany kod demonstruje pełen przepływ pracy – od wczytania danych z pliku CSV, poprzez przetworzenie kolumn i zbudowanie modelu uczenia maszynowego, aż po udostępnienie interaktywnego interfejsu do przewidywania ostatecznej oceny ucznia w trzecim okresie (G3). Dzięki temu każdy zainteresowany (nauczyciel, uczeń, rodzic czy badacz danych) może na żywo sprawdzić, jak różne czynniki wpływają na sukces w nauce matematyki.