- Eksploracja i wstępna analiza danych

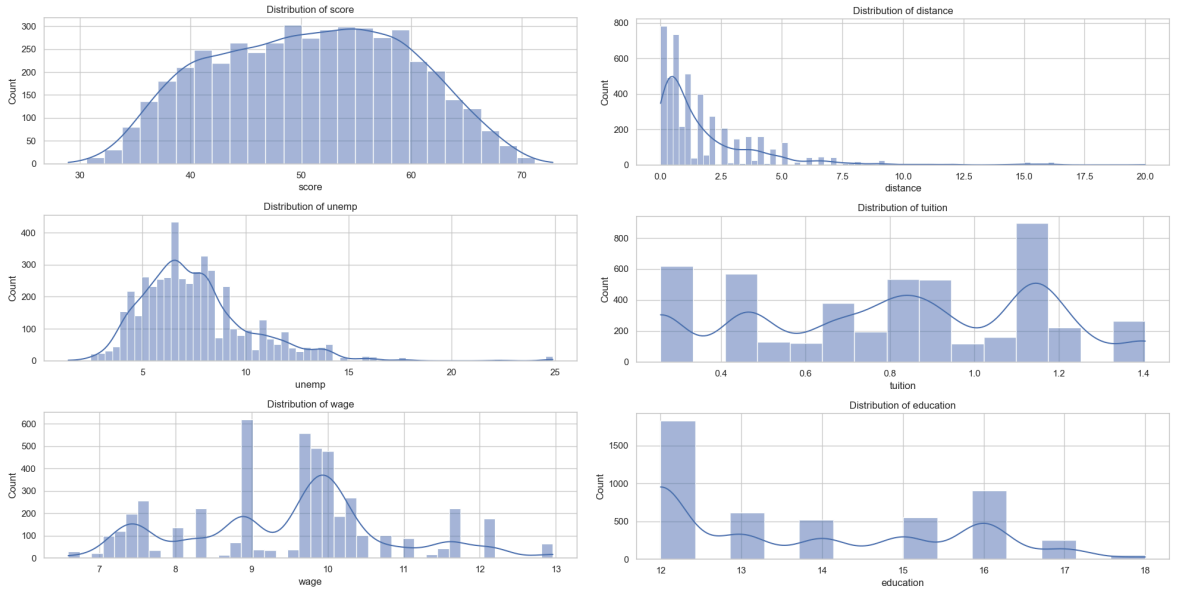
  Do wykonania analizy eksploracji użyto skryptu **analyzer.py**. Wyniki analizy:

```
RangeIndex: 4739 entries, 0 to 4738
Data columns (total 15 columns):
 #   Column     Non-Null Count  Dtype
---  ------     --------------  -----
 0   rownames   4739 non-null   int64
 1   gender     4739 non-null   object
 2   ethnicity  4739 non-null   object
 3   score      4739 non-null   float64
 4   fcollege   4739 non-null   object
 5   mcollege   4739 non-null   object
 6   home       4739 non-null   object
 7   urban      4739 non-null   object
 8   unemp      4739 non-null   float64
 9   wage       4739 non-null   float64
 10  distance   4739 non-null   float64
 11  tuition    4739 non-null   float64
 12  education  4739 non-null   int64
 13  income     4739 non-null   object
 14  region     4739 non-null   object
```

  Zbiór danych zawiera 4739 rekordów oraz 15 kolumn. Można zauważyć, że:

  - Typy danych: Kolumny są mieszane, obejmują dane typu liczbowego (np. score, unemp, wage, distance, tuition, education) oraz kategorycznego (np. gender, ethnicity, fcollege, mcollege, home, urban, income, region).
  - W danych nie ma brakujących wartości
  - Wybrane statystyki opisowe:
    - Zmienna score ma średnią wartość 50.89 i rozstęp od 28.95 do 72.81.
    - Zmienna distance ma średnią 1.8 z zakresem od 0 do 20.

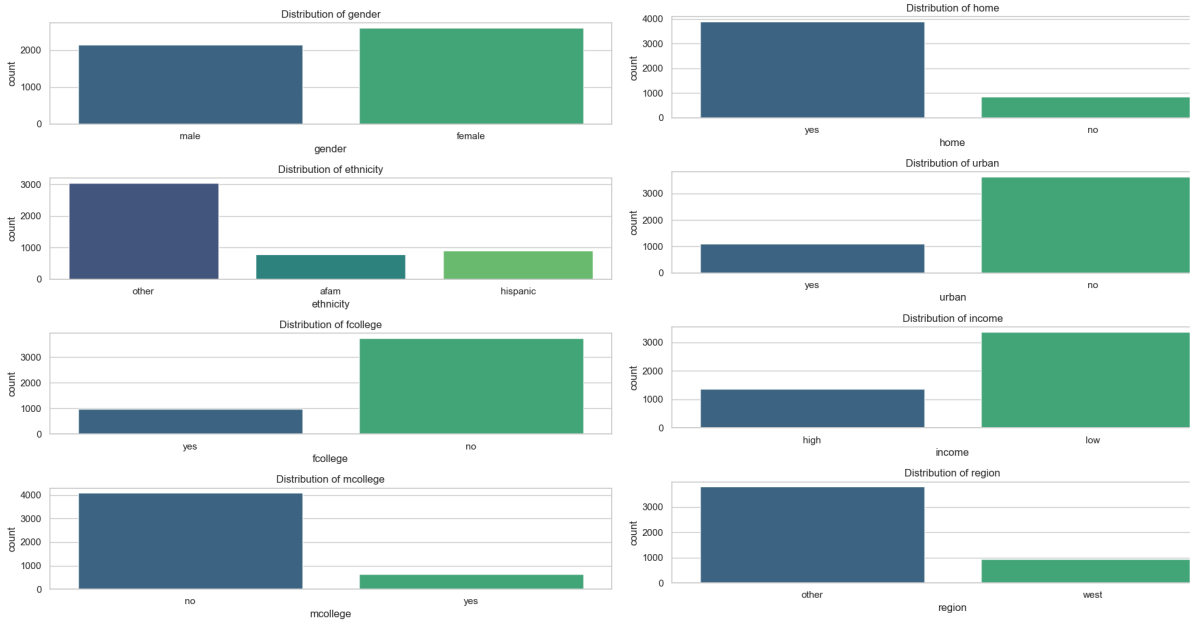
Eksploracja danych - zmienne numeryczne:



Powyższe wykresy przedstawiają rozkłady dla zmiennych numerycznych:

- **Score** oraz **education** mają rozkłady zbliżone do rozkładu normalnego.
- **Unemployment rate (unemp)** oraz **tuition** wykazują koncentrację wokół wartości centralnych, ale są asymetryczne.
- **Wage** ma wąski rozkład z niewielkimi odchyleniami od średniej.
- **Distance** pokazuje, że większość uczniów mieszka blisko uczelni, lecz istnieje długi "ogon" dla większych odległości.

Eksploracja danych - zmienne kategoryczne:



Wykresy przedstawiają rozkłady zmiennych kategorycznych:

- **Gender** jest równomiernie rozłożony między male i female.
- **Ethnicity** zawiera kilka kategorii, z dominującą kategorią other.
- **Fcollege** i **mcollege** wskazują, czy rodzice uczęszczali na studia, przy czym większość odpowiedzi to no.
- **Home** i **urban** wskazują na różnice w środowisku zamieszkania studentów, z przewagą odpowiedzi yes dla obu zmiennych.
- **Income** dzieli się głównie na kategorie high i low.
- **Region** jest zróżnicowany, ale kategoria other przeważa.

- Inżynieria cech i przygotowanie danych

  Przygotowanie danych odbywa się w skrypcie **predicter.py.** Przygotowanie danych składa się z następujących kroków:

  1. **Kategoryzacja**
     a. Skrypt identyfikuje kolumny kategoryczne: ['gender', 'ethnicity', 'fcollege', 'mcollege', 'home', 'urban', 'income', 'region'].
     b. **One-Hot Encoding** jest użyty do zakodowania wartości kategorycznych
  2. **Standaryzacja**
     a. Identyfikowane są kolumny numeryczne: ['score', 'unemp', 'wage', 'distance', 'tuition', 'education'].
     b. Zastosowano **standaryzację** do tych kolumn za pomocą StandardScaler z biblioteki sklearn. Standaryzacja przekształca wartości tak, aby miały średnią 0 i odchylenie standardowe 1
  3. **Podział na zbiór treningowy i testowy**
     a. Dane są dzielone na zbiór treningowy i zbiór testowy w stosunku 80% do 20% odpowiednio. Zbiór treningowy służy do nauki modelu, a zbiór testowy do oceny jego skuteczności.
     b. Kolumna **score** jest używana jako zmienna docelowa (y), a pozostałe kolumny jako zmienne objaśniające (X).

  Wynik działania skryptu - logi:

```
Procent nowych kolumn po kategoryzacji: 6.67%
Procent danych zmodyfikowanych przez standaryzację: 37.50%
Standaryzacja i kategoryzacja zakończone.
Dane zostały podzielone na zbiór treningowy i testowy.
Liczba danych w zbiorze treningowym: 3791
Liczba danych w zbiorze testowym: 948
```

- Wybór i trenowanie modelu

  Wybór i trenowanie modelu odbywa się w skrypcie **predicter.py.** Celem jest przewidywanie zmiennej score na podstawie podstawie cech takich jak distance, tuition, oraz zmiennych demograficznych (jak np. gender, ethnicity, income). Jest to klasyczny problem regresyjny, ponieważ score jest liczbowy, a nie kategoryczny. Z tego powodu wybrano popularne modele regresji i porównano je ze sobą by wybrać ten który przewiduje najlepiej. Wybrane modele:

  - **Regresja Liniowa**: Używana, aby sprawdzić, czy prosta relacja liniowa między cechami a zmienną docelową jest wystarczająca do dokładnego przewidywania.
  - **Random Forest**: Model oparty na zespole drzew decyzyjnych, który zazwyczaj dobrze radzi sobie z danymi nieliniowymi.
  - **Decision Tree**: Model nieliniowy, który często sprawdza się w przypadku relacji złożonych. Jest mniej złożony niż Random Forest, co daje możliwość porównania jego wyników z wynikami algorytmów zespołowych.
  - **Gradient Boosting**: Bardziej złożony model zespołowy, który buduje kolejne drzewa decyzyjne, korygując błędy poprzednich modeli.

  Z uwagi na najlepsze wyniki wybrano **Gradient Boosting.**

  Wynik działania skryptu:

```
Rozpoczyanie uczenia

Wyniki modeli:
                          MSE  R2 Score
Linear Regression  0.647811  0.353258
Random Forest      0.685464  0.315667
Decision Tree      1.223866 -0.221846
Gradient Boosting  0.623301  0.377728
```

- Ocena i optymalizacja modelu

  Podczas oceny modelu wzięto pod uwagę metryki:

  - **Mean Squared Error** - miara, informująca, jak daleko prognozy modelu znajdują się od rzeczywistych wartości. Im niższa wartość MSE, tym lepiej.
  - **R2 Score** - miara jak dobrze model dopasowuje się do danych. R2 w zakresie od 0 do 1 oznacza, że model dobrze dopasowuje dane (im bliżej 1, tym lepiej)

  Opierając się o powyższe wartości jako najlepszy model wybrano **Gradient Boosting**, z uwagi na to, że ma najniższe MSE(więc jego przewidywania są

najbliżej wartości rzeczywistych) oraz najwyższy i nieujemny R2 score, co oznacza że model dobrze dopasowuje dane.
Wynik działania wyboru w skrypcie:

```
Najlepszy model: Gradient Boosting
Metryki najlepszego modelu:
MSE         0.623301
R2 Score    0.377728
```