**Środowiska uruchomieniowy AutoML**

Salary Calculator

Pavlo Khrapko – s25140  
Andrii Nahornyi – s24918

* **Wstęp**
* **Cel projektu**

Ten projekt ma na celu stworzenie predykcyjnego kalkulatora wynagrodzeń na podstawie zbioru danych dotyczącego wynagrodzeń w branży data science. Kalkulator uwzględnia różne czynniki, takie jak poziom doświadczenia, rola zawodowa oraz wielkość firmy, aby oszacować wynagrodzenie pracownika. Narzędzie wykorzystuje dane historyczne, aby dostarczyć wgląd w to, jak różne zmienne związane z pracą wpływają na poziom wynagrodzeń. Dzięki temu pomaga organizacjom i osobom indywidualnym zrozumieć potencjalne zarobki w dziedzinie data science

* **Opis wybranej bazy danych**

Zbiór danych wykorzystany w tym projekcie to '2023 Data Science Salaries',

dostępny na platformie Kaggle. Zawiera kluczowe informacje dotyczące stanowisk pracy, lokalizacji, poziomów doświadczenia oraz innych czynników wpływających na wynagrodzenia w branży data science

**Szczegóły zbioru danych:**

* **experience\_level**: Poziom doświadczenia pracownika z wartościami:
* **EN** (Entry-level/Junior)
* **MI** (Mid-level/Intermediate)
* **SE** (Senior-level)
* **EX** (Executive-level/Director)
* **job\_title**: Tytuł stanowiska pracy (np. Data Scientist, Machine Learning Engineer).
* **employee\_residence**: Kraj zamieszkania pracownika (kod ISO).
* **company\_location**: Lokalizacja firmy, która zawarła kontrakt (kod ISO).
* **remote\_ratio**: Proporcja pracy zdalnej w % (np. 0%, 50%, 100%).
* **company\_size**: Wielkość firmy z wartościami:
* **S** **(Small)** - mniej niż 50 pracowników,
* **M** **(Medium)** - 50-250 pracowników,
* **L** **(Large)** - więcej niż 250 pracowników.
* **work\_year**: Rok, w którym wypłacono wynagrodzenie.
* Model uczenia maszynowego

**Przygotowanie danych**

Do budowy modelu wykorzystano zbiór danych **'2023 Data Science Salaries'**. Proces

przygotowania danych obejmował:

* Usunięcie niepotrzebnych kolumn, takich jak **salary** i **salary\_currency**.
* Wstępne przetwarzanie zmiennych, w tym obsługę zmiennych kategorycznych.
* Usunięto kolumny nie mające dużego wpływu (znaleziono za pomocą Autogluon)

Obraz zawierający tekst, zrzut ekranu, Czcionka, numer

Opis wygenerowany automatycznie

**Architektura modelu**

Do przewidywania wynagrodzeń wykorzystano model **LightGBMXT**, który został wybrany z automatycznie przetestowanych modeli za pomocą **AutoGluon**.

from autogluon.tabular import TabularDataset, TabularPredictor

# Przygotowanie zboiru treningowego

train\_data = TabularDataset(df)

# Trenowanie modelu

predictor = TabularPredictor(label='salary\_in\_usd', eval\_metric="root\_mean\_squared\_error").fit(train\_data, presets=["medium\_quality","optimize\_for\_deployment"])

Obraz zawierający tekst, zrzut ekranu, Czcionka, oprogramowanie

Opis wygenerowany automatycznie

Algorytm **LightGBMXT** oferuje wysoką dokładność i wydajność dzięki wykorzystaniu gradientowego boostingu. Otrzymane hiperparametry: LightGBMXT: {'learning\_rate': 0.05, 'extra\_trees': True}

**Podział danych**

Dane zostały podzielone na zbiór treningowy i testowy w stosunku 80:20. Dzięki temu

możliwe było przetestowanie modelu na nowych, niewidzianych wcześniej danych.

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

# Podział danych

train\_set, test\_set = train\_test\_split(df, test\_size=0.2, random\_state=42)

**Ewaluacja modelu**

Model został oceniony za pomocą takich metryk jak:

* Root Mean Squared Error (RMSE) - 0.11
* Mean Absolute Error (MAE) - 0.08

Model **LightGBMXT** osiągnął najlepsze wyniki, wskazując na wysoką skuteczność w

przewidywaniu wynagrodzeń na podstawie kluczowych cech, takich jak **experience\_level**,

**job\_title** i **company\_location**.

* Opis aplikacji
* **Wykorzystane technologie**

Do trenowania modelu wykorzystaliśmy bibliotekę LightGBM, która zapewnia wydajność i skuteczność w zadaniach uczenia maszynowego. Aplikację webową zbudowano za pomocą Streamlit, umożliwiającego interaktywne wprowadzanie danych i prezentację wyników.

**Dodatkowo użyto bibliotek:**

* Pandas i NumPy – do przetwarzania danych,
* Scikit-learn – do podziału danych i ewaluacji modelu,
* Matplotlib – do wizualizacji wyników.
* **Opis funkcjonalności**

Aplikacja oferuje trzy główne funkcjonalności: interaktywne przewidywanie wynagrodzeń, obsługę zmiennych kategorycznych oraz wyświetlanie wyników w czasie rzeczywistym

* **Interaktywne przewidywanie wynagrodzeń**

Użytkownik może wprowadzić dane takie jak poziom doświadczenia, tytuł stanowiska i lokalizacja firmy za pomocą prostego interfejsu w aplikacji Streamlit. Po wprowadzeniu danych aplikacja przewiduje wynagrodzenie

* **Obsługa zmiennych kategorycznych**

Aplikacja automatycznie przetwarza zmienne wejściowe, takie jak **experience\_level** i **company\_location**, na format odpowiedni do użycia w modelu.

* **Ewaluacja wyników**

Aplikacja wyświetla wyniki przewidywania wynagrodzeń w czasie rzeczywistym. Wyniki są generowane natychmiast po kliknięciu przycisku „Calculate” w interfejsie użytkownika

Załączniki:

Aplikacja: <https://s24918-salary-calculator-calculator-ctpxm7.streamlit.app/>

Kod: <https://github.com/s24918/Salary-Calculator>