Opis tematyki projektu

Część analityczna projektu polegała na znalezieniu odpowiedniego zbioru danych, a następnie skorzystaniu z poznanych na zajęciach bibliotek języka Python, w celu przeanalizowania, zwizualizowania i wyciągnięcia wniosków z pobranych danych. Zaproponowana przez zespół problematyka dotyczy informacji na temat mieszkań wystawionych na wynajem, na popularnym niemieckim serwisie ogłoszeniowym. Wśród celów, które postanowiono spełnić było sprawdzenie danych pod kątem podstawowych metryk analitycznych, takich jak m.in. wolumen, trendy czasowe lub korelacje.

Cel realizacji projektu

Celem opracowania analizy danych było zapoznanie się i utrwalenie wiedzy z wykorzystania najpopularniejszych narzędzi z dziedziny Data Analytics oraz Data Science. Zespół postawił sobie za cel również wykroczenie poza zakres zajęć, wykorzystując do tego celu bibliotekę NLTK oraz zaawansowane zagadnienia z zakresu biblioteki pandas.

Opis stosu technologicznego

W projekcie wykorzystano środowisko programistyczne jakim jest Jupyter Notebook. Narzędzie to bazuje na rozwiązaniu IPython – nakładce na język Python, rozszerzającej jego możliwości. Dzięki integracji tych dwóch narzędzi możliwe jest wykonywanie kodu w pojedynczych, osobnych komórkach, wyświetlając wynik działania tuż pod tą komórką. [3] Takie podejście znacznie usprawnia analizę danych, gdyż, tak jak w rzeczywistym notesie, możliwe jest zobaczenie co zostało napisane wcześniej i jakie wyniki otrzymano.

Zarządzanie instancją Pythona oraz instalacją dodatkowych bibliotek możliwe było, dzięki pakietowi Anaconda. Pakiet ten zawiera wiele preinstalowanych paczek, wspierających analizę danych, a ponadto pozwala na kontrolę zgodności wersji poszczególnych bibliotek pomiędzy sobą – w przypadku gdy niemożliwa jest instalacja danego pakietu, ze względu na powiązania z innym pakietem, możliwe jest stworzenie drugiego wirtualnego środowiska, rozwiązującego potencjalny problem.

Do analizy wykorzystano najbardziej podstawowe narzędzia analityczne, tj:

- pandas

- matplotlib

- seaborn

Biblioteka pandas oferuje szereg działań na tzw. DataFrame’ach, czyli obiektach reprezentujących zbiór danych w formie tabelarycznej. Możliwe jest wczytanie danych różnych formatów (np. xlsx, html, csv) i prowadzenie zaawansowanych analiz, np. poprzez agregowanie danych, sortowanie, usuwanie kolumn lub dodawanie nowych.

Biblioteka seaborn jest właściwie nakładką na matplotlib, umożliwiającą szybkie i proste tworzenie przeróżnych wizualizacji. Zgodnie z dokumentacją seaborn, biblioteka ta współpracuje z pandasem, wspólnie zapewniając niemal komplet narzędzi niezbędnych dla początkującego analityka. [1]

W ramach projektu zdecydowano się również na sprawdzenie możliwości, jakie oferuje NLTK, czyli Natural Language Toolkit. Jest to zestaw bibliotek do przetwarzania języka naturalnego, pozwalającym m.in. na usuwanie tzw. stopwords (wyrazy które nie noszą za sobą znaczenia) lub na sprowadzanie wyrazów do formy podstawowej. [2]

Opis realizacji / Opis sposobu realizacji zadań projektowych

W ramach części analitycznej można wyróżnić następujące etapy realizacji projektu:

1. Sprawdzenie zawartości zbioru danych
2. Oczyszczenie danych
3. Przekształcenia danych
4. Agregacja danych
5. Wizualizacja danych
6. Obserwacja trendów, wartości odstających itp.
7. Wyciągnięcie wniosków

Ze względu na charakter prowadzenia analiz w narzędziu Jupyter Notebook, wskazane wyżej czynności nie były wykonywane „z góry do dołu” jeden raz, lecz były powtarzane wielokrotnie, według bieżącego zapotrzebowania.

Sprawdzenie zawartości zbioru danych było pierwszą czynnością wykonaną po wczytaniu obiektu DataFrame. W tym celu skorzystano z takich metod jak df.info() oraz df.describe(). Pierwsza z nich pozwoliła sprawdzić jakich typów (np. int, float, object) są kolumny wczytanego pliku oraz czy dane są kompletne. Druga metoda przedstawiła podstawowe wskaźniki statystyczne (średnia, mediana, poszczególne percentyle) dla kolumn typu numerycznego. Dzięki tym dwóm metodom, jasne stało się które fragmenty danych będą istotne z punktu widzenia analizy, a które można pominąć.

Jak zostało wspomniane wyżej, poszczególne etapy były powtarzane wielokrotnie. Do oczyszczenia danych można zaliczyć usunięcie nieprawdopodobnych dat wybudowania budynków – w DataFrame pojawia się na przykład rok 1900 lub 2090 jako rok konstrukcji budynku.

Wśród czynności związanych z przekształceniem danych można wymienić posortowanie danych zgodnie z datą ich zebrania, co następnie pozwoliło na prawidłowe pokazanie na wykresie. Stworzono również koszyki składające się z zakresu dat w których dane budynki powstały, umożliwiając stworzenie histogramu.

Agregacja danych była niezbędnym etapem pozwalającym na przejrzyste zaprezentowanie kluczowych elementów. Wykorzystano metodę .groupby() zasilając ją odpowiednimi kolumnami, w zależności od celu, który w danej chwili chciał osiągnąć analityk. Znaczna większość agregacji miała na celu wyznaczyć liczebność danego zbioru, w pojedynczym przypadku zdecydowano się na wyznaczenie poszczególnych percentyli.

Krokiem następującym po agregacji była wizualizacja. Do tego celu skorzystano z bibliotek matplotlib oraz seaborn. Stworzone wykresy to przede wszystkim wykresy kolumnowe, które udało się dopasować do różnych zastosowań. Na przykład, w podpunkcie 2.4 wykres słupkowy posłużył do przedstawienia zmian w czasie, a w podpunkcie 2.5 do wyznaczenia statystyk dla danej zmiennej. W innych przypadkach wizualizował liczebność danego zbioru danych.

Obserwacja trendów oraz wyciągnięcie wniosków nie byłyby możliwe bez wizualizacji. Bardzo ważne było, aby analiza była zrozumiała dla potencjalnych interesariuszy, tj. typy wykresów odpowiadały typom danych i treści, którą analityk chciał przekazać.

6. Wnioski

Python oraz biblioteki takie jak pandas, seaborn, NLTK są doskonałym narzędziem w pracy analityków danych, pozwalając na szybkie uzyskanie potrzebnych analiz. Mnogość funkcji oraz rozbudowana społeczność programistów Pythona sprawiają, że implementacja nawet złożonych zagadnień jest znacznie szybsza, niż przy wykorzystaniu standardowych narzędzi, jak np. Excel.

Skonstruowanie wniosków dotyczących samej analizy danych, wymagałoby najpierw dokładnego zdefiniowania problemu, a następnie przeprowadzenia znacznie bardziej rozbudowanej analizy, starającej się zrozumieć sedno obserwacji które dotychczas udało się zauważyć. Z wykonanej pracy wyłania się obraz niemieckiego rynku mieszkaniowego jako miejsca pełnego skrajności. Niemal połowę wylistowanych ogłoszeń stanowią oferty z wyłącznie dwóch landów, z dwóch przeciwnych stron kraju. Są to Nadrenia Północna-Westfalia, obszar związany z przemysłowym, wysokorozwiniętym Zagłębiem Ruhry oraz Saksonia, obszar zza żelaznej kurtyny, według powszechnej opinii zacofany względem reszty kraju. Spore dysproporcje widzimy w cenach czynszu – w rekordowym Monachium przeciętny czynsz to około 1700 EUR, natomiast w Chemnitz, które również łapie się do Top 10, przeciętny czynsz to około 500 EUR.

Możliwości dalszego rozwoju

Ze względu na rozmiar zbioru danych, możliwe jest rozszerzenie analizy na wiele przeróżnych wymiarów. Wśród możliwości można wymienić m.in. analizę metrażu mieszkań w zależności od rodzaju budownictwa (budownictwo współczesne, powojenne, przedwojenne), analizę rodzajów mediów grzewczych występujących w niemieckich mieszkaniach, analizę różnic pomiędzy czynszem bazowym, a czynszem całkowitym.

Prawdopodobnie możliwe byłoby również wykorzystanie zbioru danych do stworzenia modelu Data Science, który mógłby pomóc wyceniać dane mieszkanie na podstawie danych z tego pliku. Ze względu na wiele wymiarów, zarówno liczbowych jak i kategorycznych, taki model mógłby bardzo trafnie oceniać koszt związany z wynajmem danego lokalu.

Bibliografia

[1] <https://seaborn.pydata.org/tutorial/introduction.html>

[2] <https://towardsdatascience.com/getting-started-with-nltk-eb4ed6eb7a37>

[3] <https://jupyter-notebook.readthedocs.io/en/latest/>

Wykorzystane fragmenty kodu:

<https://stackoverflow.com/questions/17578115/pass-percentiles-to-pandas-agg-function>