**Projekt Hurtowni zamówień codziennych zakupów online wraz z warstwą danych pogodowych.**

### Zespół Danonkowych Żółwi Ninja [🐢](https://emojipedia.org/turtle/):

## *Katarzyna Solawa i Mateusz Sperkowski*

**Cel projektu**

Celem naszego projektu było stworzenie hurtowni danych reprezentującej dane ze zbioru Instacart:

(<https://www.kaggle.com/c/instacart-market-basket-analysis/data>) i dodatkowe dane pogodowe (<https://www.kaggle.com/datasets/sobhanmoosavi/us-weather-events>) oraz dane o miastach w USA (<https://simplemaps.com/data/us-cities>). Pierwszy zbiór, to dane o dostawach zakupów w roku 2016 do mieszkańców USA z aplikacji Instacart udostępnione w roku 2017, a drugi mówi o wydarzeniach pogodowych w różnych miastach USA na przedziale czasowym 2016-2021. Ostatnia tabela to podstawowe dane o miastach z cenzusu w USA. Celem hurtowni jest zbadanie zależności kupowanych produktów od reszty dostępnych wymiarów, a w szczególności pogody w danym dniu i mieście.

Planowanym odbiorcą naszego rozwiązania byłby wyższy zarząd aplikacji Instacart lub podobnej świadczącej usługi zakupów online. Nasze rozwiązania byłoby bardzo proste do rozwinięcia na skale międzynarodową, więc dowolna aplikacja podobnego typu mogłaby korzystać z bardzo bliskiego schematu rozwiązania. Za pomocą naszych raportów odbiorca otrzymałby szczegółowe informacje o działaniu biznesu, czyli o dowozach zakupów. Dodatkową wartościową informacją jest pogoda występująca przy finalizowaniu zamówienia, która okazuje się mieć istotny wpływ na wybór zakupów przez użytkowników aplikacji. W ten sposób można sprawdzić czy nasze spekulacje, np. że w mroźną lub deszczową pogodę występuje więcej zamówień na składniki do gorących napojów, są prawdziwe czy też nie. Daje to dużo możliwości do rozwoju przedsiębiorstwa, między innymi można decydować, czy zwiększamy ilość dostawców, gdy jest większa ilość zamówień, dostosowujemy reklamy do pogody i użytkownika by zagwarantować bardziej spersonalizowane doświadczenie, lub w inne sposoby optymalizować działanie firmy. Dodatkowo interaktywność raportu pozwala na analizę na zbliżony obszar działania, więc również menadżerowie lokalni mogliby korzystać z części rozwiązania do badania swojego terenu.

**Architektura**

Skrót architektury naszego rozwiązania jest widoczny powyżej.

Dane dostępne są w formacie .csv, z wyjątkiem tabeli z datami, gdzie korzystamy ze skryptu udostępnionego nam na zajęciach. Skrypt ten generuje tabele bezpośrednio w hurtowni. By dało się połączyć zanonimizowane dane zakupów, w języku Python generujemy dodatkowe kolumny w danych i całą tabele users.csv.

Pierwszym krokiem architektury jest wczytanie płaskich plików .csv w aplikacji Tableau Prep, służącej dalej nam do procesów ETL. Po wszystkich transformacjach zapisujemy dane do hurtowni danych w MS SQL Server. Następnie w Tableau Desktop tabele są wczytywane do systemu BI. Tam stworzone przez nas raporty generują wyniki dla obecnych danych i otrzymujemy wizualizacje finalnego raportu/raportów. Poszczególne kroki są szerzej opisane poniżej.

**Zbiory Danych**

Dostępne nam dane przedstawiają przede wszystkim informacje o danych zamówieniach oraz każdych produktów do nich należących.

Pierwszy zbiór danych to kolejne zamówienia użytkowników na zakupy z aplikacji Instacart. Obsługuje ona mieszkańców USA i przedstawia co było kupione, z jakiej półki/kategorii i czy to kolejny zakup użytkownika. Tabela Departments przedstawia szczegóły co do przynależenia do działu sklepu, tabela Aisle określa przynależenie do alejki w sklepie, a tabela products daje szczegóły na temat danego produktu. Dodatkowo tworzymy tabele Users która odpowiada ilości użytkowników w naszych danych, zawiera ona podstawowe dane o osobach korzystających z aplikacji. Tabele, które składają się na tabele faktów to orders oraz orders\_products, gdzie ta druga to dwa pliki .csv z danymi, połączone ze sobą. Tabela orders mówi o szczegółach dla danego zamówienia, a tabela orders\_products mówi o szczegółach dla danego produktu w danym zamówieniu. Finalnie te tabele są połączone w tabele faktów.

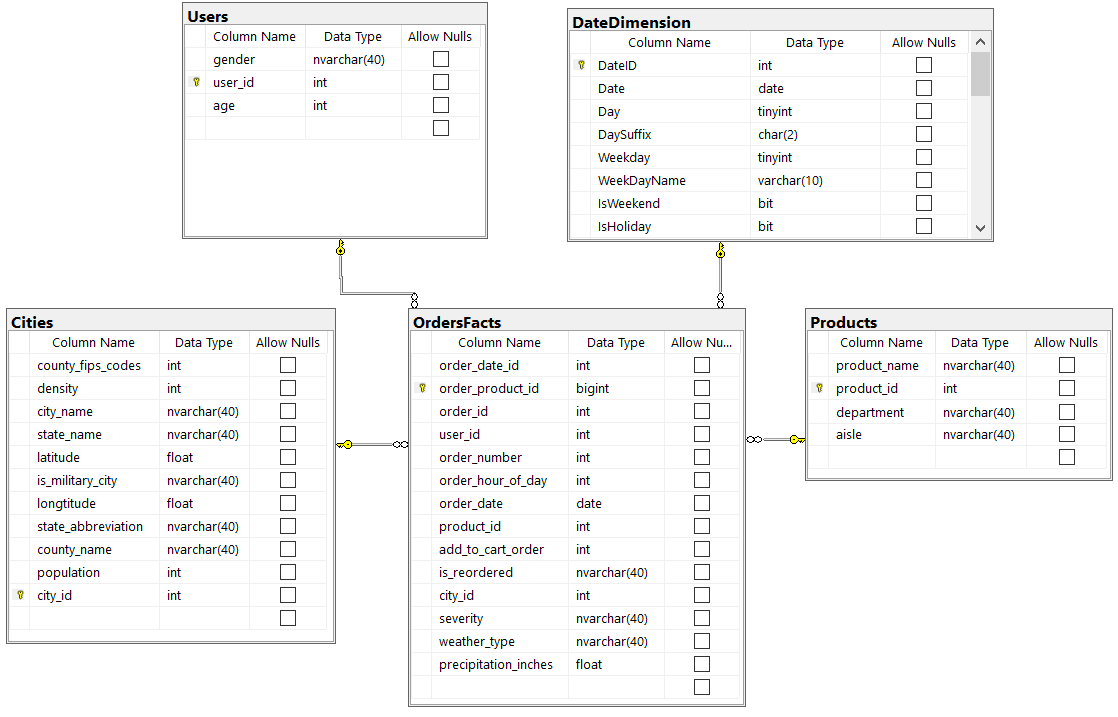
Drugi zbiór mówi o wydarzeniach pogodowych w różnych miastach USA i przedstawia rodzaj wydarzenia, date i jego intensywność. Trzeci zbiór przedstawia podstawowe dane o miastach w USA, przynależenie do stanów i hrabstwa oraz populacje i lokalizacje geograficzne.

Ponieważ części danych potrzebnych do działania hurtowni, łączenia tabel, te dane generujemy w sposób losowy lub częściowo losowy. Robimy to w języku Python. Oznacza to oczywiście że tracimy część lub całość informacji o realnych relacjach tych danych. W szczególności generujemy całkowicie dane użytkowników z pomocą paczki randomuser. Również kluczowo generujemy daty zamówień, jednak jest to pół losowy proces. W sensie, możemy znaleźć tak dokładnie date zamówienia, jaka jest suma ilości odstępów w dniach między zamówieniami. W niektórych wypadkach, gdy tam suma wynosi koło 365 dni, mamy dokładne daty, sprawdzaliśmy to za pomocą dni tygodni dostepnych w danych. Gdy ta suma jest mniejsza, losujemy dowolny dzień z zakresu (1szy dzień roku, 365-suma dzień roku). Powoduje to nierówny i nieprawdziwy rozkłąd zamówień w czasie, jednak uznajemy to za ‘Proof of Concept’ architekture, więc badamy co dalej wyjdzie z naszego projektu. Losujemy też zamówieniom lokalizacje, korzystając z przecięcia miast w zbiorach WeatherEvents i USCities – tu tylko z tych które występują raz, ponieważ dwa miasta mogą mieć jedną nazwę, co sprawiało problemy przy wcześniejszych etapach projektu. Tak samo ostatnią zmianą, wyrzucamy ze zbioru WeatherEvents duplikaty wydarzeń pogodowych w jednym mieście w jednym dniu w zakresie 2016-2017, by znów nie występowały duplikowane wiersze w naszej hurtowni.

**ETL**

W procesie ETL sprowadzamy tabele do formatu wykorzystywanego w hurtowni danych. We wszystkich tabelach usuwamy null’e oraz zmieniamy nazwy na czytelne dla każdego użytkownika i tak samo dekodujemy pola typu boolean, lub skróty. Łączymy tabele products, aisles i departments do jednej tabeli przedstawiającej wszystkie szczegółowe informacje o danym produkcie. Tabela DateDimension powstaje z oddzielnego skryptu sql i jest specjalnie przygotowana do wykorzystania w środowisku hurtowni danych, więc nie potrzebne są dla niej żadne transformacje. Natomiast daty w pozostałych tabelach, czyli Orders i WeatherEvents są przetransformowane do formatu klucza głównego DateDimension, czyli RRRRMMDD. Następnie przechodzimy do tworzenia tabeli faktów. Łączymy pliki orders\_products\_\* ze sobą dopinając wiersze, jednak okazuje się, że pewne wiersze w tych zbiorach z jakiegoś powodu się powtarzają. Wyrzucamy duplikaty. Następnie łączymy join’em z tabelą orders. Kolejno używając wygenerowanych danych o miastach zamówienia łączymy z tabelą cities ograniczoną tylko do klucza głównego i nazwy miasta, by otrzymać klucz obcy. Z klucza głównego produktu i zamówień tworzymy nowy klucz główny tabeli faktów. Do naszej obecnej tabeli dodajemy na końcu dane o pogodzie danego dnia w danym mieście, korzystając ze zbioru WeatherEvents. Gdzie nie ma wydarzenia pogodowego, wpisujemy pogodę ‘Clear’ czyli bez żadnych wydarzeń. Usuwamy też zbędne w naszym systemie kolumny. Wszystkie te dane wrzucane są do Hurtowni danych, opisanej poniżej. Dodatkowo możliwe jest dodawanie nowych danych pojawiających się w plikach. Dostępne są dwa tryby, incremental, czyli dodanie tylko nowo znalezionych wierszy, przeznaczony na krótszy okres między wywołaniami oraz full, czyli ponowne zapełnienie danych, które odbywałoby się w dłuższych odstępach czasowych. Spowodowane jest to zarówno obciążeniem obliczeniowym serwerów, czasem niedostępności hurtowni oraz spójnością danych. Komendy te można odpalac z wiersza poleceń więc nie byłoby problemu z ustawieniem cyklicznego puszczania danych.

**Model Hurtowni Danych**



Hurtownia danych ma schemat gwiazdy, gdzie OrderFacts to tabela faktów, a tabele: Users, Cities, DateDimention oraz Products pełnią funkcje tabeli wymiarów.

Hurtownie tworzymy w Microsoft SQL Server poprzez skrypty SQL, które tworzą tabele DateDimension oraz szablony tabel: OrderFacts, Users, Cities oraz Products, które następnie są zasilane poprzez Tableau Prep. Skrypt tworzy klucze oraz połączenia między tabelami. Tableau Prep posiada opcję tworzenia tabel od zera, lecz tak stworzone tabele trzeba następnie i tak połączyć, a wszystkie kolumny mają typy float, bigint lub nvarchar(4000).

Kluczem głównym tabeli OrderFacts jest order\_product\_id, który został stworzony na etapie ETL, poprzez “sklejenie” oder\_id oraz product\_id. Stworzony w ten sposób klucz może przyjmować bardzo duże wartości, dlatego jest on typu bigint. Dla pozostałych kluczy w hurtowni ustawaliśmy typ int. OrderFacts posiada także 4 klucze obce: order\_date\_id, user\_id, product\_is, city\_id które pozwalają łączyć się z tabelami wymiarów.

DateDimension to kalendarz z dokładnością do dni od 1 stycznia 1900 roku do 31 grudnia 2099 roku. Posiada głownie takie informacje jak: day, week, month, quarter w postaci numerycznej i słownej oraz informacje logiczne typu: XofY, FirstXofY, LastXofY, IsHoliday. Kluczem głównym tabeli jest DateId.

Products posiada klucz główny products\_id, oraz wymiary: product\_name, aisle oraz department.

Cities posiada klucz główny city\_id oraz posiada miary definiujące lokalizację, wymiar IsMilitary oraz dwie miary: density oraz population.

Users posiada klucz główny user\_id oraz wymiar gender oraz miarę age.

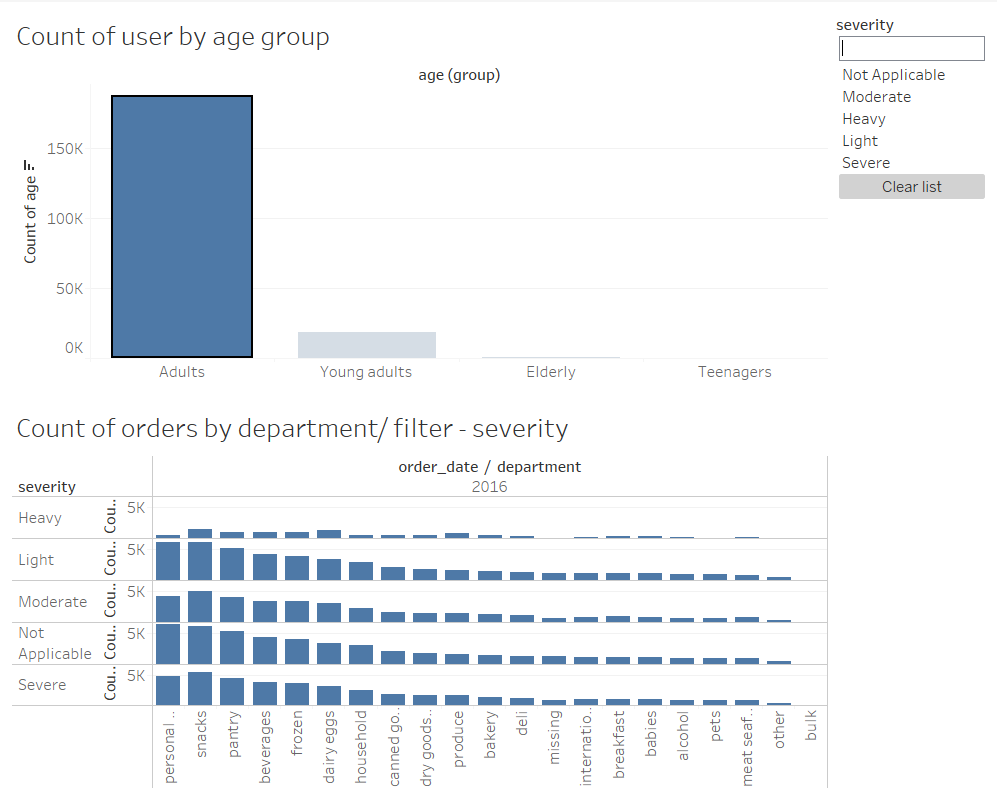
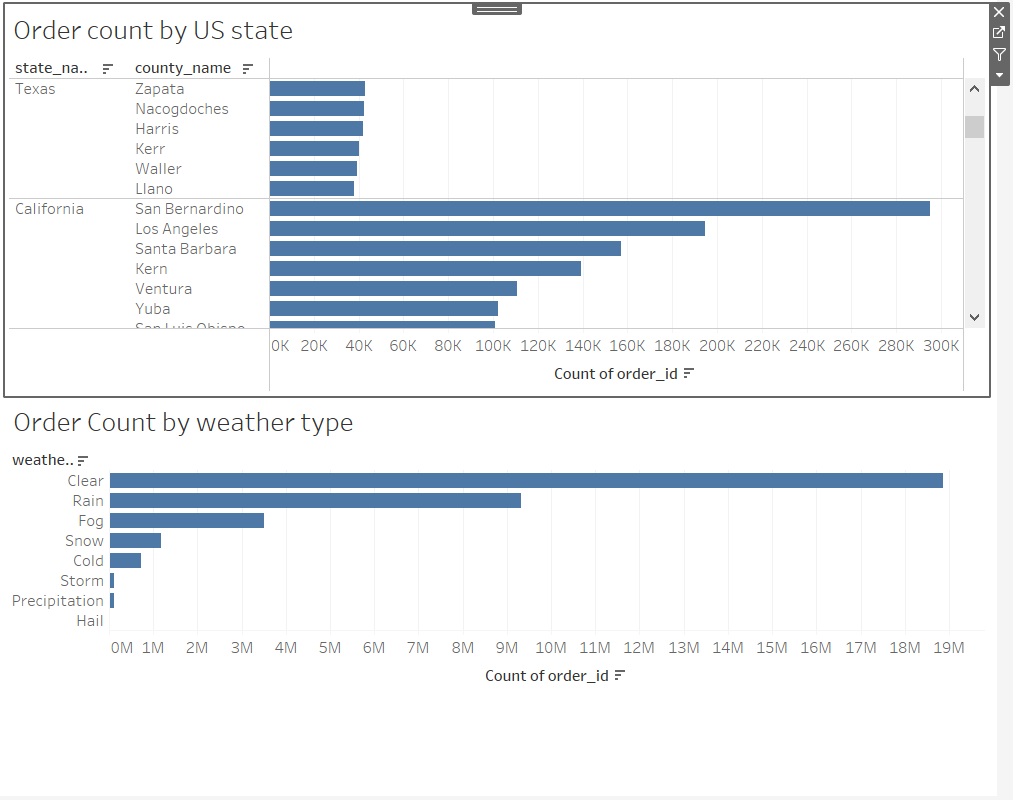
**Warstwa Raportowa**

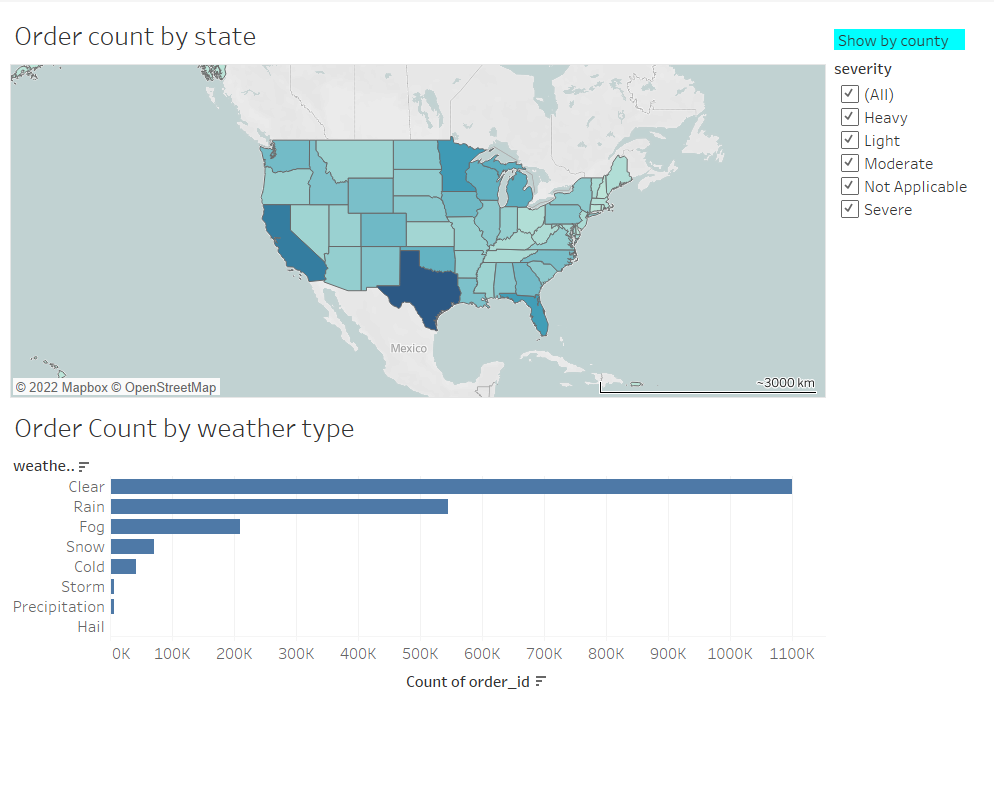
Do raportowania używamy Tableau Desktop. Na starcie łączymy się z hurtownią danych w Microsoft SQL Server na następnie wybieramy odpowiednie tabele i łączymy je według ustalonego wcześniej schematu.

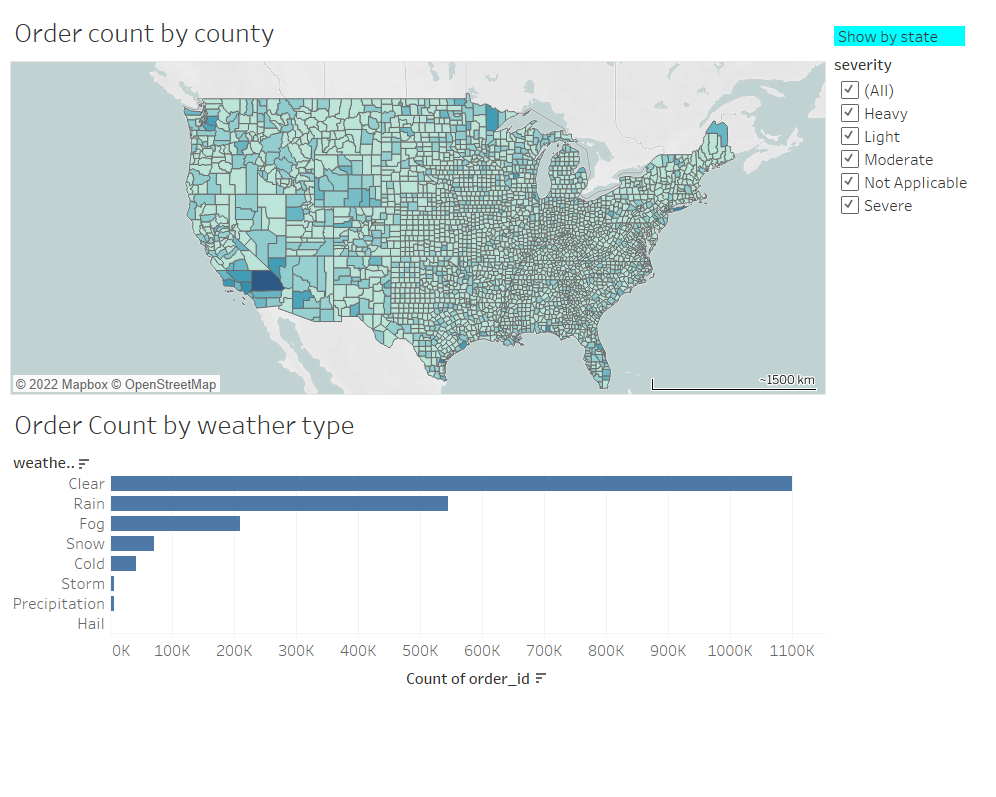
Wartości Latitude i Longtitude dopasowujemy w Tableau Desktop do znanych wartości dla miast/hrabstw/stanów. W ramach warstwy raportowej wykorzystujemy kilka wyliczanych pól.

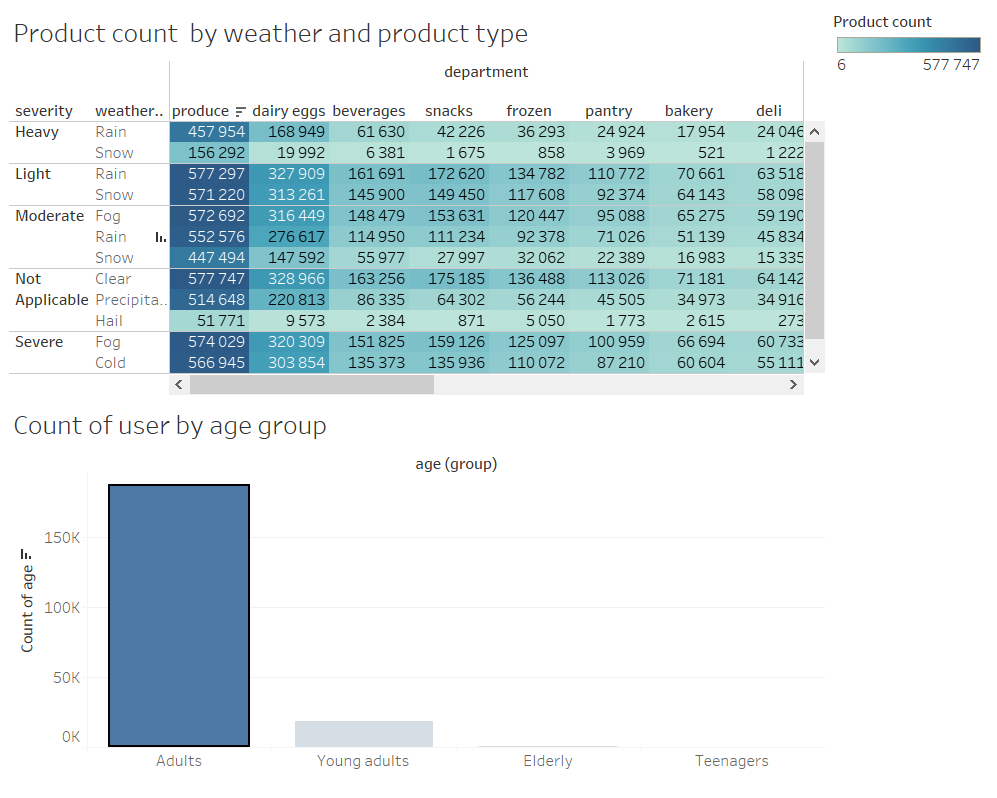
Oprócz prostych agregacji (suma, zliczenie, zliczenie unikatowych, procent całości) do raportu przygotowaliśmy:

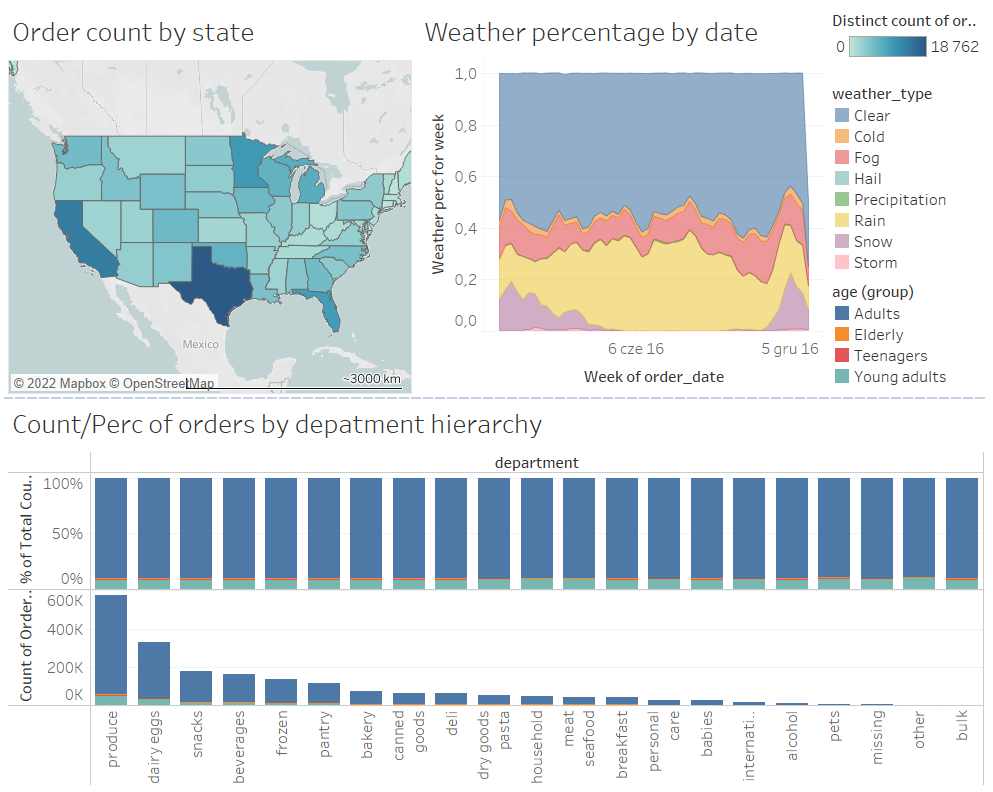
* Weather perc for week - procent danego typu pogody w danym tygodniu;
* Age(group) - podział na 4 grupy wiekowe: teenagers, young adults, adults, elderly;
* Avg order cnt by weather - średnia liczba zamówień jednego dnia dla danego typu pogody;
* Department\_hier – department > aisle > product\_name;
* State\_name\_hier – state\_name > county\_name > city\_name.









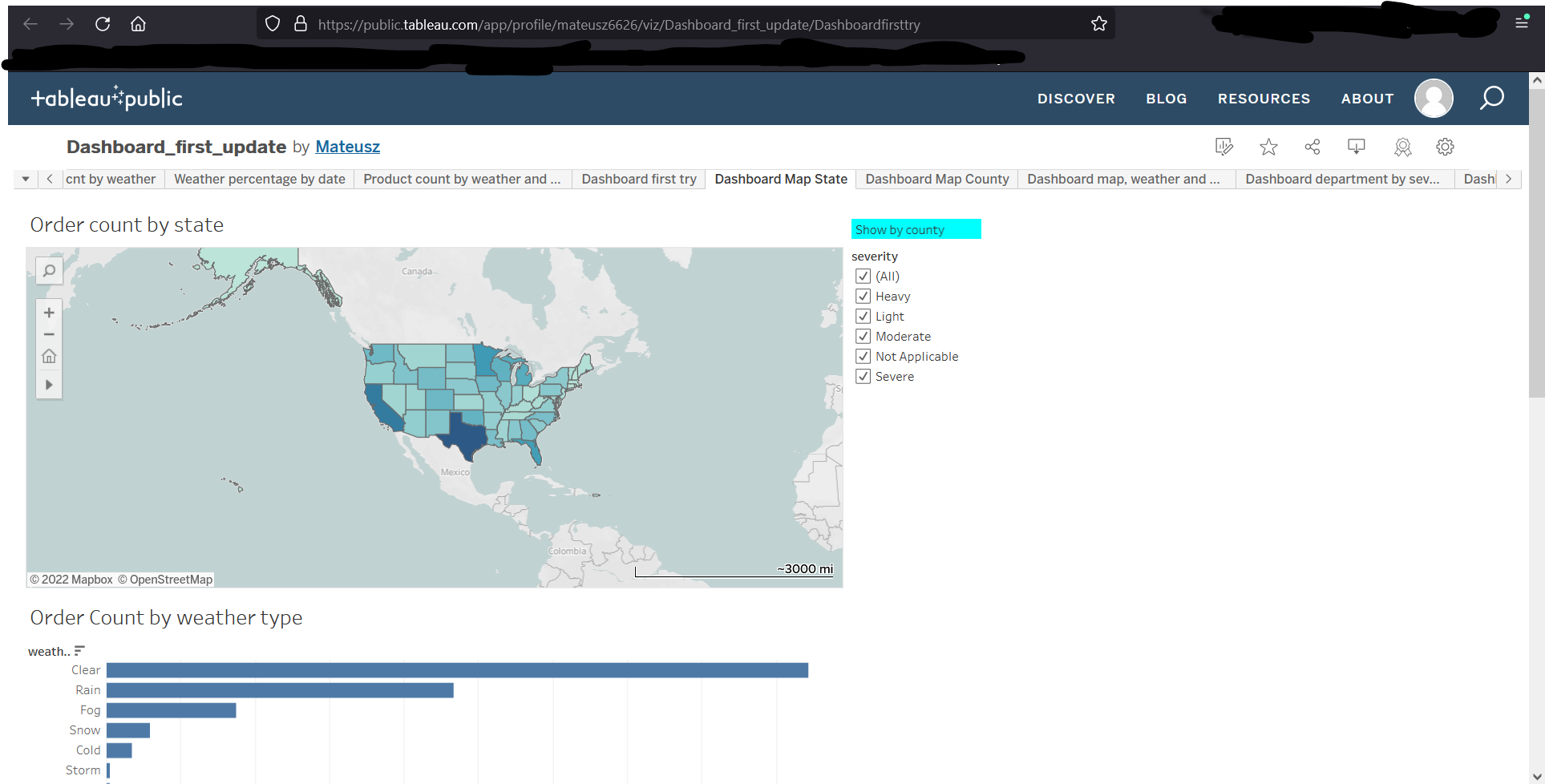


Raporty pokazują, jak rozkłada się liczba zamówień w danym stanie/hrabstwie względem pogody. Dzięki temu możemy zauważyć dla których regionów ludzie częściej robią zakupy na przykład w porę deszczową i dla tych regionów dopasować usługi.

Również pozwalają one sprawdzić jakie produkty najczęściej kupowała dana grupa wiekowa w różnych typach pogody. Pozwala to dostosować reklamy oraz promocje tak by były skierowane do jak największej liczby użytkowników aktywnych w danym czasie.

Rezultatem naszego projektu jest bardzo użyteczna dla przedsiębiorstwa Instacart warstwa raportowa. W naszych raportach zawarte jest bardzo dużo szczegółowych informacji o zmianach upodobań użytkowników w zależności od różnych czynników, a szczególnie od zmian pogodowych.

Raporty są dostępne dla użytkownika końcowego na stronie: <https://public.tableau.com/app/profile/mateusz6626/viz/Dashboard_first_update/Dashboardfirsttry>



**Testy funkcjonalne**

Zrzuty ekranu z testów funkcjonalnych znajdują się w pliku Testy\_Funkcjonalne. Możemy na nich zobaczyć poszczególne kroki działania systemu oraz naszych prac. Przedstawione są ogólne przykłady działania oraz bardziej szczegółowe dla przykładowych komponentów. Dodatkowo na koniec przedstawione jest działanie dodawania nowych danych do hurtowni.

**Podział pracy w zespole:**

Wspólne zadania: Szukanie danych, pomysł projektu, dokumentacja wstępna, architektura systemu, wykorzystywane narzędzia, dokumentacja końcowa

Katarzyna Solawa: Stworzenie Hurtowni Danych, raporty

Mateusz Sperkowski: Stworzenie ETL’a, testy, prezentacja, upload dashboardów