States’ Craigslist Stats

based on Craigslist API

**Призначення та коротка характеристика програми**

Даний проект – це програма розроблена для аналізу даних, які знаходяться у відкритому доступі за рахунок Craigslist API. За допомогою розроблених модулів можна отримати інформацію щодо статистичних даних нерухомості США. Інформація представлена для кожного штата окремо і складається з таких пунктів:

* Число оголошень
* Середня ціна
* Медіана цін
* Середня довжина оголошення
* Найбільш популярне місце
* Число оголошень відмічених геотегом
* Число оголошень, які мають фотографії
* Число оголошень, які відмічені на карті

Дані, які отримано за допомогою цієї програми призначені як для використання в персональних цілях, так і для економічних та соціологічних досліджень.

В персональних цілях користувачі можуть використовувати результати для покращення ефективності вибору нерухомості для життя та вибору місця життя в залежності від наявних цін на квартири та інші будівлі. Також існує можливість використання інформації для планування формату та інформативного наповнення власного оголошення для його позиціонування на дошці оголошень Craigslist.

Дані, які надаються за рахунок обробки статистичних даних сайту оголошень можуть використовуватись для загальної оцінки ситуації на ринку нерухомості США та для детального аналізу кожного штату окремо. Для соціологічних досліджень доцільно використовувати інформацію щодо популярних місць в оголошеннях (або їх відсутність) та залежність між цінами та наявністю/відсутністю фотографій та відмітки на карті.

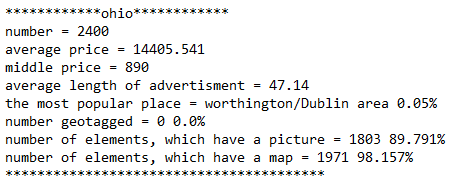
**Вхідні та вихідні дані програми**

Дослідження базується на API Craigslist. Дані представлені в вигляді класів для кожного з видів оголошень на даному сайті, таких як:

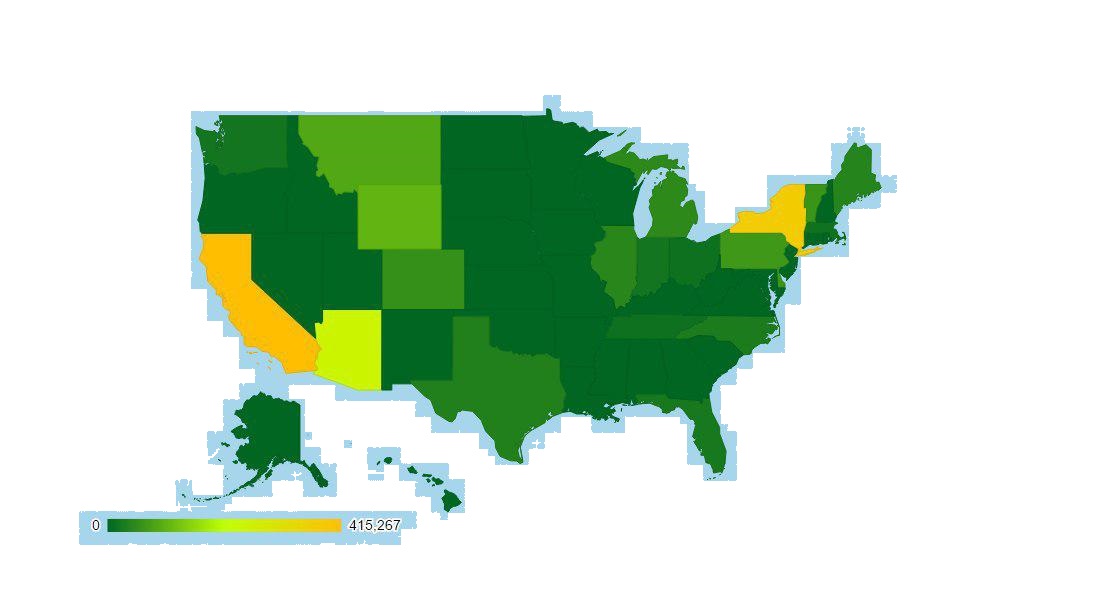
* Громада
* Житло
* Робота
* Працівники
* Для продажу
* Події
* Послуги
* Резюме

Для дослідження були використані оголошення про житло, роботу та для продажу.

Вхідні дані можливо фільтрувати за допомогою вбудованих фільтрів середи обробки. Під час дослідження були переважно використані фільтри, які відокремлювали інформацію для кожного штату та ті, що показували наявність певних тегів у оголошення: геотег, фотографія та карта.

 Результати дослідження представлені у вигляді окремого текстового документа, який створюється по результату виконання програми. В даному файлі отримані статистичні дані показані для кожного штату у вигляді:

Також був реалізований візуальний супровід для наочного результату:



**Структура програми з коротким описом модулів, функцій, класів та методів**

*apartment.py - обробляє дані та повертає обчислення*

*class Apartment – задає структуру екземплярів класу, в якому зберігаються всі дані про житло*

*add(self, state, adv) – додає штат до списку, якщо його там немає і додає інформацію з оголошення*

*refresh\_price(self, other\_state, other\_adv=None) – додає значення ціни до оголошення, якщо воно відрізняється від того, що є*

*check\_is\_near\_the\_same(self, state, advert) – видаляє оголошення з нульовою ціною*

*average\_price(self, state) – обчислює середню ціну в штаті*

*middle\_price(self, state) – обчислює медіану ціни в штаті*

*number(self, state) – обчислює кількість оголошень в штаті*

*price\_for\_state(self, state) – перевыряє чи ціни для штата обраховані*

*average\_len\_advert(self, state) – обчислює середню довжину оголошення*

*most\_popular\_place(self,state,percentage=False) – знаходить найпопулярніше місце в штаті*

*geotaged(self, state, percentage = False) – оголошення має геотег да/ні*

*pictured(self, state, percentage=False) – оголошення має зображення да/ні*

*mapped(self,state, percentage= False) – оголошення має відмітку на карті да/ні*

*arrays.py – створює абстрактний тип даних для дослідження*

*class Array – створює макет екземпляру*

*\_\_len\_\_(self) – повертає довжину масиву*

*\_\_getitem\_\_(self, index) – повертає елемент з масиву за індексом*

*\_\_setitem\_\_(self, index, value) – вставляє елемент за індексом*

*clear(self, value) – присвоює всім елементам значення value*

*\_\_iter\_\_(self) – створює ітератор (клас \_ArrayIterator) )в масиві*

*class \_ArrayIterator – створює макет екземплярів класу*

*\_\_iter\_\_(self) – стоворює ітератор*

*\_\_next\_\_(self) – повертає наступний за індексом елемент*

*class Array2D – створює екземпляри двохвимірного масиву*

*num\_rows(self) – повертає кількість рядків*

*num\_cols(self) – повертає кількість стовпчиків*

*clear(self, value) – присвоює всім елементам значення value*

*\_\_getitem\_\_(self, index\_tuple) - повертає елемент з масиву за індексом*

*\_\_setitem\_\_(self, index\_tuple, value) – вставляє елемент за індексом*

*class DynamicArray – створюэ екземпляри динамічного масиву*

*\_\_len\_\_(self) – повертає розмір масиву*

*\_\_getitem\_\_(self, index) - повертає елемент з масиву за індексом*

*append(self, value) – додає новий елемент до масиву*

*\_resize(self, capacity) – змінює розмір масиву відповідно до capacity*

*\_make\_array(self, size) – створює масив розміру size*

*insert(self, index, value) – вставляє елемент з відповдіним value за індексом*

*remove(self, value) – видаляє елемент з певним value*

*remove\_ind(self, index)- видаляє елемент за індексом*

*cars.py – аналогічно до apartment.py*

*jobs.py– аналогічно до apartment.py*

*platform\_element.py – перетворює безпосередньо дані отримані за допомогою craigslist api*

*class Platform\_element – створює екземпляри класу*

*read\_states.py – видаляє штати, які вже були оброблені зі списку*

*start\_read\_file.py*

*data\_for\_one(example, state, file) – додає елемент до класу, створює приклад*

*read\_file(file\_name) – записує дані в файл*

*take\_data.py – зчитує дані, які потім обробляються в модулях apartment.py, cars.py, jobs.py*

**Коротка інструкція по користуванню програмою**

Програма розроблена з простим інтуїтивно зрозумілим алгоритмом роботи:

1. Запускається модуль take\_data.py, який зчитує інформацію
2. Запускаєтесь модуль start\_read\_file.py, який записує отриману інформацію в текстовий документ result.txt

**Опис тестових прикладів для перевірки працездатності програми**

Для перевірки правильності роботи програми був розроблений модуль test.py, який виконує весь функціонал дослідження на малому об’ємі даних.