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.

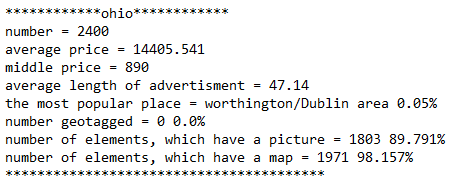
**Вхідні та вихідні дані програми**

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

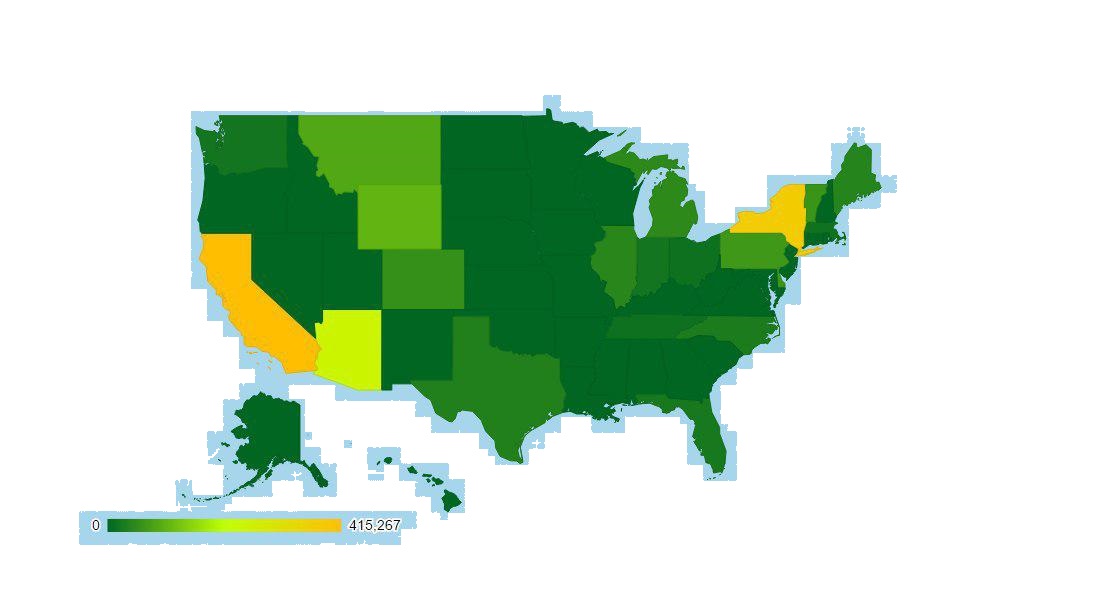
* Громада
* Житло
* Робота
* Працівники
* Для продажу
* Події
* Послуги
* Резюме

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*

*class Apartment*

*add(self, state, adv)*

*refresh\_price(self, other\_state, other\_adv=None)*

*check\_is\_near\_the\_same(self, state, advert)*

*average\_price(self, state)*

*middle\_price(self, state)*

*number(self, state)*

*price\_for\_state(self, state)*

*average\_len\_advert(self, state)*

*most\_popular\_place(self,state,percentage=False)*

*geotaged(self, state, percentage = False)*

*pictured(self, state, percentage=False)*

*mapped(self,state, percentage= False)*

*arrays.py*

*class Array*

*\_\_len\_\_(self)*

*\_\_getitem\_\_(self, index)*

*\_\_setitem\_\_(self, index, value)*

*clear(self, value)*

*\_\_iter\_\_(self)*

*class \_ArrayIterator*

*\_\_iter\_\_(self)*

*\_\_next\_\_(self)*

*class Array2D*

*num\_rows(self)*

*num\_cols(self)*

*clear(self, value)*

*\_\_getitem\_\_(self, index\_tuple)*

*\_\_setitem\_\_(self, index\_tuple, value)*

*class DynamicArray*

*\_\_len\_\_(self)*

*\_\_getitem\_\_(self, index)*

*append(self, value)*

*\_resize(self, capacity)*

*\_make\_array(self, size)*

*insert(self, index, value)*

*remove(self, value)*

*remove\_ind(self, index)*

*cars.py*

*jobs.py*

*platform\_element.py*

*class Platform\_element*

*read\_states.py*

*start\_read\_file.py*

*data\_for\_one(example, state, file)*

*read\_file(file\_name)*

*take\_data.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.