******Departamentul** **Automatică și Tehnologia Informației**

**Programul de studii: Tehnologia Informației**

***ANDRIȘAN Ariana***

PROIECT DE DIPLOMĂ

**Conducător științific:**

Șef lucr. dr. ing. DEMETER Robert

Brașov 2020

**Departamentul: Automatică și Tehnologia Informației**

**Programul de studii: Tehnologia Informației**

***ANDRIȘAN Ariana***

Analiza spațială a zonelor turistice utilizând rețelele sociale

**Conducător științific:**

Șef lucr. dr. ing. DEMETER Robert

Brașov 2020

**Cuprins**

Cuprins

[Lista de figuri, tabele și coduri sursă 4](#_Toc43731752)

[Lista de acronime 7](#_Toc43731753)

[1 Introducere 8](#_Toc43731754)

[1.1 Sumar 8](#_Toc43731755)

[1.2 Obiectivele lucrării 10](#_Toc43731756)

[1.3 Structura 11](#_Toc43731757)

[2 Prezentarea generală a tool-urilor 12](#_Toc43731758)

[2.1 Airbnb 12](#_Toc43731759)

[2.2 Python 15](#_Toc43731760)

[2.3 Anaconda 20](#_Toc43731761)

[2.4 Jupyter Notebook 20](#_Toc43731762)

[2.5 ParseHub 21](#_Toc43731763)

[3 Parcurgerea proiectului 22](#_Toc43731764)

[3.1 Explorarea datelor 22](#_Toc43731765)

[3.1.1 Extragerea datelor 23](#_Toc43731766)

[3.1.2 Bibliotecile Beautiful Soup și Scapy 23](#_Toc43731767)

[3.1.3 C Sharp 25](#_Toc43731768)

[3.1.4 NodeJS 26](#_Toc43731769)

[3.1.5 Octoparse 29](#_Toc43731770)

[3.1.6 ParseHub 29](#_Toc43731771)

[3.2 Analiza datelor 32](#_Toc43731772)

[3.2.1 Citirea datelor 33](#_Toc43731773)

[3. 3 Vizualizarea datelor 34](#_Toc43731774)

[3.4 Curățarea datelor 36](#_Toc43731775)

[3.5 Machine Learning 36](#_Toc43731776)

[3.5.1 Definiție 36](#_Toc43731777)

[3.5.2 Modul de funcționare 37](#_Toc43731778)

[3.5.3 Predicția prețului 37](#_Toc43731779)

[3.5.4 Algoritmul K-nearest neighbors 37](#_Toc43731780)

[3.5.5 Distanța Euclidiană 39](#_Toc43731781)

[3.5.6 Modelul de bază KNN 40](#_Toc43731782)

[3.5.7 Evaluarea modelului 41](#_Toc43731783)

[3.5.8 RMSE (Root Mean Squared Error) - Eroarea rădăcinii medie pătrate 42](#_Toc43731784)

[3.6 Analiza economică 44](#_Toc43731785)

[3.6.1 Regresia Liniară 48](#_Toc43731786)

[3.7 Date Geospațiale 50](#_Toc43731787)

[3.7.1 Conexiunea programului cu Google Maps 50](#_Toc43731788)

[3.8 Analiza finală 52](#_Toc43731789)

[3.9 Transformarea proiectelor jupyter în aplicații web 54](#_Toc43731790)

[4 Concluzii 56](#_Toc43731791)

[5 Bibliografie 57](#_Toc43731792)

[Rezumat 59](#_Toc43731793)

[Abstract 60](#_Toc43731794)

# Lista de figuri, tabele și coduri sursă

FIGURI

Figura 1. Ponderi a listărilor Airbnb după numărul de înregistrări

Figura 2. Airbnb

Figura 3. Prezentarea generală a sistemului

Figura 4. Popularitatea limbajului Python în 2020

Figura 5. Framework Environment - Programare Orientată pe Obiecte

Figura 6. Etapele proiectului

Figura 7. Biblioteci importate

Figura 8. Structura proiectului Python

Figura 9. Colectare date Pyrthon

Figura 10. Rezultatul programului Python

Figura 11. Fișier CSV

Figura 12. C Sharp consolă

Figura 13. Rezultat NodeJS

Figura 14. Octoparse șablon Airbnb

Figura 15. ParseHub Airbnb Crawler

Figura 16. Fișier CSV neformatat

Figura 17. Fișier CSV formatat

Figura 18. Pașii crawlerului ParseHub

Figura 19. Pașii extragerii coordonatelor din hartă

Figura 20. CSV final

Figura 21. Rezultatul citirii datelor în Jupyter Notebook

Figura 22. Vizulizarea coloanei "reviews"

Figura 23. Vizulizarea coloanei "price"

Figura 24. Corelare

Figura 25. Total calculat

Figura 26. Număr de k camere

Figura 27. Valori de similitudine

Figura 28. Selectarea primelor k listări

Figura 29. Calculare preț de listă

Figura 30. Vizualizarea prețurilor

Figura 31. Adăugarea coloanei "date"

Figura 32. Vizualizarea prețurilor în funcție de dată

Figura 33. Vizulizarea primelor 5 linii din analiza datelor

Figura 34. Rezultatul învățării automate

Figura 35. Graficul rezultat după aplicarea regresiei liniare

Figura 36. Glob

Figura 37. Harta

Figura 38. Harta marcată

Figura 39. Încadrarea zonelor

Figura 40. Tipul camerei

TABELE

Tabelul 1. Diferitele tipuri de cazare în funcție de structura proprietății

Tabelul 2. Rezumat tehnologii

Tabelul 3. Rezultat analiză finală

CODURI SURSĂ

Codul 1. Citirea și afișarea CSV-ului.

Codul 2. Statistici - rezumat.

Codul 3. Combinarea operațiilor.

Codul 4. Afișarea tabelului CSV utilizând biblioteca pandas.

Codul 5. Calcularea distanței

Codul 6. Compararea calculării distanței

Codul 7. Sortarea valorilor

Codul 8. Media prețurilor

Codul 9. Evaluarea modelului

Codul 10. Eroarea rădăcinii medie pătrate

Codul 11. Compararea rezultatelor modelelor de predicție

Codul 12. Distanța euclidiană

Codul 13. Utilizarea algoritmului pentru RMSE

Codul 14. Implementare algoritm

Codul 15. Sortarea și setarea indexului 'date'

Codul 16. Implementarea problemei folosind tehnica medie în mișcare

Codul 17. Antrenarea și vizualizarea rețelelor

Codul 18. Crearea și stocarea datelor

# Lista de acronime

API - Application Programking Interface;

Captcha - Completely Automated Public Turing Test To Tell Computers and Humans Apart

ETL - Extract, Transform, Load (tradus: Extrage, Transformă, Încarcă);

HTML - Hypertext Markup Language;

HTTP - Hypertext Transfer Protocol (tradus: Protocol de transfer);

HTTPS - Hypertext Transfer Protocol Secure (tradus: Protocol de transfer securizat);

IA - Inteligență Artificială ;

ID - identitate;

IDE - Integrated Development Environment;

IESC – Inginerie Electrică și Știința Calculatoarelor;

JDBC - Java Database Connectivity;

JSON - JavaScript Object Notation;

ML - Machine Learning ("Învățare automată");

NumPy - Numerical Python;

ODBC - Open Database Connectivity;

ORM - Object-Relational Mapper;

P2P - Peer-to-peer;

PDF - Portable Document Format;

RegEx - Regular Expression;

SPA - Single-page Application;

SQL - Structured Query Language;

URL - Uniform Resouce Locator;

XML - Extensible Markup Language;

XPath - XML Path Language;

XSLT - Extensible Stylesheet Language Transformations;

RMSE - root mean square error

# 1 Introducere

|  |
| --- |
|  |
| **Sumar**  **Obiectivele lucrării**  **Structura** |

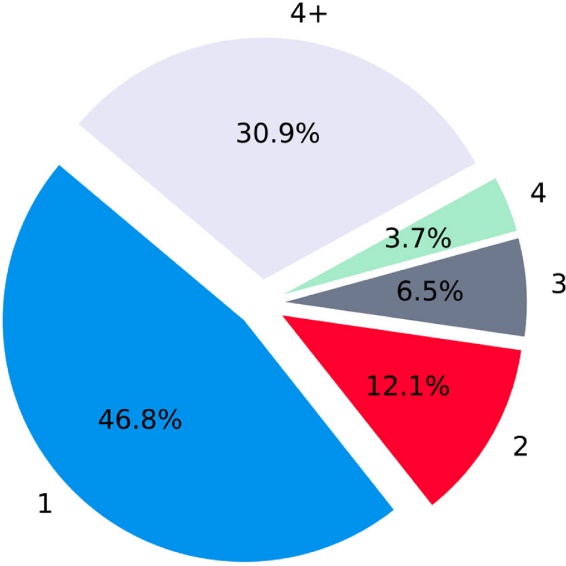
## Sumar

Internetul oferă o gamă largă de platforme de tip rețele de socializare sau peer-to-peer (P2P) precum, Airbnb, Flickr, TreepAdvisar care partajează locuri de cazare (apartamente, hoteluri, pensiuni, etc.), tips&tricks pentru vacanțe, dar și păreri și impresii ale acestor locuri din partea persoanelor care au folosit aceste resurse.

Dintre toate acestea cea mai mare platforma este Airbnb cu peste 5 milioane de oferte de închiriere la nivel global. Datorită experienței lor de peste 10 ani , Airbnb a devenit un subiect important de cercetare în cadrul turismului și studii urbane sau academice , precum și studii sociale. Acestea includ închirierile, locurile cele mai căutate în vederea cazării, locurile cele mai des vizitate de turiști, impactul economic asupra respectivului oraș, satisfacția clienților. Aceste analize permit întelegerea geografică a fenomenului, oferind informații despre factorii care afectează distribuția sa.

Până acum, majoritatea studiilor sau concentrat pe orașe mari precum Berlin (Germania) sau Madrid (Spania) sau alte orașe mari. În urma acestor studii s-a observat o relație între distribuția listărilor Airbnb și profilul socioeconomic al zonelor respective. Alte studii susțin că listările Airbnb sunt concentrate în diferite zone față de hotelurile din centrul orașelor arătând tendința de a alege o cazare mai ieftină, intr-un spațiu poate mai mare, aici putând să ne referim la închirierea unor case sau apartamente.

Din studiul citat a rezultat următoarele ponderi a listărilor Airbnb după numărul de înregistrări oferite. (Fig. 1)



*Figura 1 Ponderi a listărilor Airbnb după numărul de înregistrări*

Rezultatul final al acestui studiu este reprezentat în următorul tabel (Tab.1) care prezintă diferitele tipuri de cazare în funcție de structura proprietății (o cameră - 1, două camere - 2, trei camere - 3, patru camere - 4, mai multe camere - 4+).[2]

*Tabelul. 1 Diferitele tipuri de cazare în funcție de structura proprietății*

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | 1 | 2 | 3 | 4 | 4+ |
| Entire home | 46.3% | 11.2% | 5.6% | 3.3% | 33.6% |
| Private home | 8.4% | 15.2% | 9.3% | 5.4% | 21.7% |
| Shared home | 46.2% | 11.8% | 12.9% | 4.3% | 24.7% |

Entire home reprezintă camera întreagă unde oaspeții au tot locul pentru ei înșiși. Aceasta include de obicei un dormitor, o baie și o bucătărie.

Private home sau camera privată, oaspeții au propria lor cameră privată de dormit. Alte zone ar putea fi partajate.

Camera comună sau shared home presupune ca oaspeții să doarmă într-un dormit sau într-o zonă comună care ar putea fi împărțite și cu alții.

Pe parcursul lurării va mai exista o categorie, întreg apartamentul în complex rezidențial, care de obicei reprezintă ori un studio cu o cameră și o bucătărie, ori un apartamnt cu două camere.

Acestă lucrare își propune să completeze cercetările menționate mai sus, de acestă dată furnizând o analiză spațială folosind listările Airbnb pentru orașul Brașov, România. Acest oraș a fost ales datorită importanței ridicate a turismului economic, dar și social. În Brașov, majoritatea turiștilor străini folosesc platforme de tip P2P pentru găsirea unui loc de cazare. Astfel putem vedea care zonă este mai căutată de către vizitatori. Accentul este pus pe rezultate oferite de grafice.

## Obiectivele lucrării

Analiza noastră empirică constă în:

1. Extragerea listărilor de pe site-ul Airbnb și a datelor furnizate de acestea.
2. Analiza datelor rezultate.
3. Curățarea datelor și păstrarea informațiilor relevante.
4. Vizualizarea datelor.
5. Prezicerea prețurilor.
6. Analiza economică a datelor și impactul acesteia asupra pieții.
7. Geolocalizarea datelor.

Inspirația acestui proiect a venit din sursele specificate mai sus, proiectele dezvoltate de către *International Journal of Geo- Information (''Airbnb Offer in Spain—Spatial Analysis of the Pattern and Determinants of Its Distribution'')* [1] și *Sensor Article ('' Can Tourist Attractions Boost Other Activities Around? A Data Analysis through Social**Networks'')* [3], dar și multe proiecte studiate pe platforma Kaggle legate tot de acest subiect. După studierea acestor surse, s-a trecut la studierea tutorialelor legate de Vizualizarea datelor (''Data Visualization''), apoi Analizarea datelor (''Data Analysis'') și în final, la studiul părții de Știința datelor (''Data Science'') și de extragere a datelor(''Data extraction/ Data mining'').

Vizualizarea datelor este reprezentarea grafică a datelor și informațiilor. Folosind elementele vizuale precum diagrame, grafice sau hărți. Aceste instrumente oferă o vizualizare a datelor, o modalitate accesibilă de a vedea și înțelege tendințele, valorile și modelele din date.

În lumea Big Data, instrumentele și tehnologiile de vizualizare a datelor sunt esențiale pentru a analiza cantității masive de informații și de a lua decizii bazate pe date.

Minarea datelor/ extragerea datelor este procesul de extragere a datelor cu ajutorul dintr-un set mare de informații (cu metode inteligente).

Analizarea datelor este un proces de inspecție, curățare și modelare a datelor cu scopul de a descoperi informații utile, de a informa concluziile și de sprijini luarea deciziilor. Analiza datelor cuprinde o varietate de forme și abordări sub o varietate de nume și utilizate în diverse domenii.

Pentru începerea practică a proiectului, s-a încercat să se răspundă la câteva întrebări de bază:

1. S-au căutat destule surse de informații astfel încât să existe o viziune de ansamblu asupra întregului proiect?

2. Ce instrumente, programe, platforme să se utilizeze?

3. Ce obiective să îndeplinească atât lucrarea teoretică cât și partea practică?

4. Împărțirea capitolelor?

Răspunsurile se vor regăsii pe parcursul acestei lucrări.

## Structura

În Capitolul 2 este prezentat pe scurt platforma Airbnb și modul de funcționare a acesteia, precum și limbajul de programare Python și platforma utilizată în Știința Datelor („Data Science"), Anaconda, dar și restul programelor utilizate.

În Capitolul 3 este prezentat parcurgerea proiectului, mici porțiuni de cod și interpretarea acestora, evenimente, erori apărute în timpul dezvoltării lucrării practice. Tot în acest capitol sunt interpretate și rezultatele obținute sub formă grafică.

În ultimul capitol, sunt prezentate concluziile și opiniile aferente întregii lucrări.

# 2 Prezentarea generală a tool-urilor

|  |
| --- |
| **Airbnb**  **Python**  **Anaconda**  **Jupyter Notebook**  **ParseHub** |

## 2.1 Airbnb

Airbnb este o abreviere de la „Air mattress B&B" a fost lansat în anul 2008, de doi designeri, Brian Chesky și Joe Gebbia, care având un spațiu liber pentru închiriat, au găzduit trei călători. Astăzi, este prezent în peste 81.000 de orașe și 192 de țări, unde milioane de gazde și vizitatori caută un loc de cazare, prin crearea unui cont pe Airbnb, gratuit, astfel putând să își listeze spațiul pentru închiriere sau să își aleagă locul de cazare dorit. Această platformă împărtășește experiențe, pasiuni, interese între călători prin intermediul gazdelor.

Când alegem un loc în care să petrecem o noapte sau mai multe, departe de casă, primul lucru pe care îl verificăm este ca locația să fie sigură. Airbnb ajută la partajarea, eficientă, plăcută și sigură, printr-un sistem de mesagerie inteligent, și datorită unei platforme sigure de efectuare a plăților online [4].

Ca oricare altă platformă, Airbnb prezintă avantaje, dar și dezavantaje. Avantaje sunt:

* Gamă largă de alegeri: listarea numeroaselor proprietăți diferite - camere single (o persoană), camere double (2 persoane), apartamente, iahturi, case întregi, cabane, bărci.
* Anunțuri gratuite: pentru a publica un anunț, nu trebuie achitarea unei taxe.
* Gazdele își pot stabilii propriul preț: alegere prețului, pe noapte, săptămână sau chiar pe lună.

Dezavantaje sunt reprezentate de:

* Ceea ce vezi listat pe site, nu poate fi și ceea ce chiar vei primii în realitate: printre multitudinea de listări, se găsesc și gazde care ascund realitatea în spatele unor poze frumoase. De aceea este important verificarea comentariilor.
* Potențialele daune: ca gazdă, cel mai mare risc pe care trebuie să ți-l asumi este deterioararea bunurilor din spațiul dat în folosință.
* Taxe adăugate: oaspeții plătesc o taxă de serviciu către Airbnb pentru clienți de la 0%-20%

Cât de mult contează știința datelor pentru Airbnb? Puțini cunosc faptul că această platformă utilizează Știința datelor („Data Science") și mașinile de învățare cunoscute ca „Machine Learning" după spusele lui Riley Newman, fostul șef al departamentului de Data Science al Airbnb. Compania privește datele ca fiind interpretarea „vocii clientului" , iar știința datelor interpretează această „voce".

Pașii prin care trec datele:

* Crearea pipeline-urilor ETL („Extract, Transform, Load") și organizarea datelor.
* Analizarea, procesarea, vizualizarea datelor transformându-le în informații utile.
* Aplicarea științei datelor în modurile practice necesare.

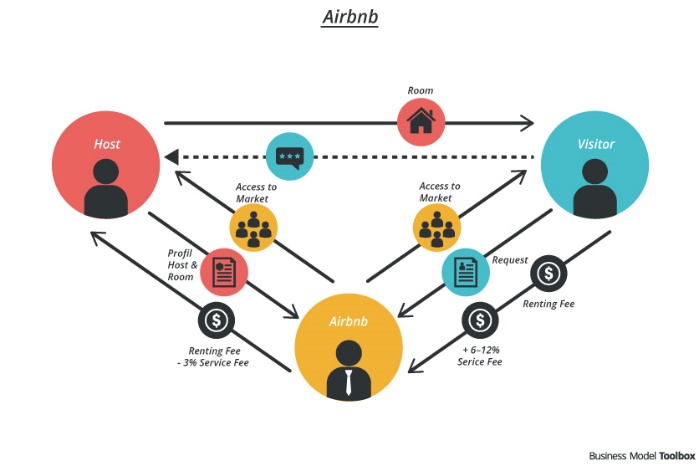
Rolurile oamenilor din spatele științei datelor în cadrul companiei:

* Analitic ("Analytics ")-> orientarea pe detalii
  + curiozitatea
  + analizarea datelor pentru a indentifica deciziile de afaceri
* Decizie ("Analytics ") -> utilizarea datelor, vizualizarea acestora și crearea statisticilor
  + rezolvarea problemelor
* Algoritmi ("Algorithms ") -> lucrul cu diferite limbaje de programare pentru crearea unor modele, sisteme de machine learning utilizate în mediul de producție.

După ce un utilizator accesează mai des platfoma, aceasta începe să recomande, locuri, cazări bazate pe căutările anterioare și chiar și pe preferințele utilizatorului. Aceste sugestii sunt posibile datorită algoritmilor bazați pe machine learning. Airbnb nu este singura platformă care reține preferințele utilizatorilor săi, Facebook, Google, Tweeter folosesc aceeași metodă de „învățare".

Factorii care determină modul în care apar listările în pagină:

* Nevoile oaspeților: ce au căutat pe platformă, călătoriile anterioare, înregistrările adăugate de către aceștia, lista de dorințe.
* Detalii despre listări: numărul de recenzii de cinci stele, prețul, locația listării. răspunsul rapid al gazdei
* Detalii despre călătorii: se are în vedere câți oameni vor călătorii, cât timp va dura călătoria, în cât timp va avea loc călătoria față de momentul prezent, prețul minim, prețul maxim.

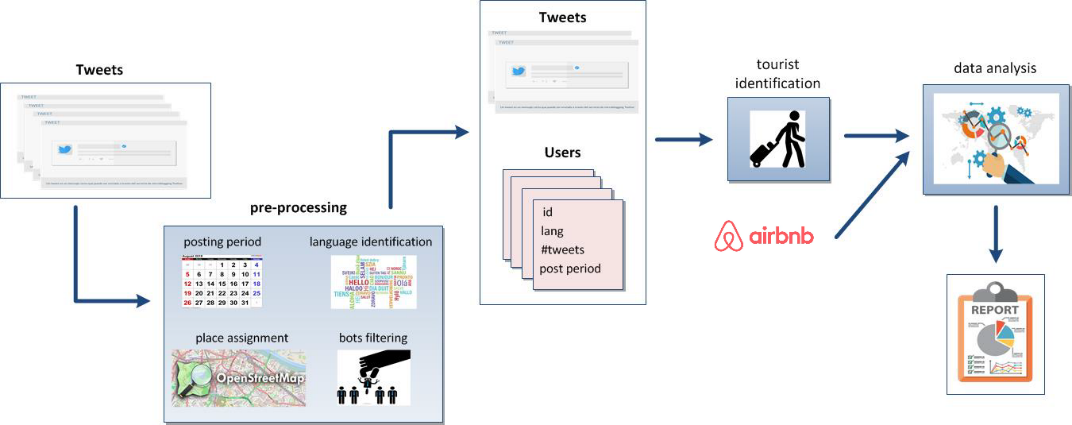


*Figura 2 Airbnb [5]*

Modul de funcționarea a platformei:

1. Crearea unui cont în care se trece numărul de telefon, țara sau logarea cu conturile de pe rețelele de socialiare (Facebook, Google, email).
2. Căutarea în bara de „Search" țara sau orașul dorit.
3. Selectarea cazării dorite.
4. Stabilirea zilelor de cazare și efectuarea plății.
5. După călătorie, împărtășirea părerii și experienței cu ceilalți utilizatori.

Din această platformă se va folosii numărul de listări de pe pagină, comentariile și evaluările.



*Figura 3 Prezentarea generală a sistemului* [3]

Securitatea platformei:

* Captcha (**C**ompletely **A**utomated **P**ublic **T**uring test to tell **C**omputers and **H**umans **A**part) - captcha reprezintă o imagine distorsionată, care de obicei nu este ușor de detectat cu ajutorul programelor de calculator, dar un om poate reușii să o înțeleagă. Majoritatea site-urilor web folosesc captcha pentru a prevenii interacțiunea roboților;
* Proxy - un server proxy acționează ca o poartă de acces între utilizator și internet. Este un server intermediar care separă utilizatorii finali de site-urile web pe care navighează. Acesta oferă diferite funcționalități, securitate și confidențialitate, în funcție de cazul de utilizare, nevoile sau politica companiei.
* HTTPS (Hypertext Transfer Protocol Secure) - este o extensie a protocolului de transfer HTTP, utilizat pentru o comunicare sigură pe o rețea de calculatoare și folosit pe scară largă pe Internet. În HTTPS protocolul de comunicare este criptat folosind Transport Layer Security (TLS) sau Secure Sockets Layer (SSL).
* Pop-up - este o zonă de afișare a interfeței grafice cu utilizatorul (GUI), de obicei o fereastră mică, care apare brusc ("apare") în prim-planul interfeței vizuale.

## 2.2 Python

Python este un limbaj de programare la nivel înalt interpretat, orientat pe obiecte (OOP), cu semantică dinamică. Este construit sub formă de structuri de date, combinat cu tastarea dinamică facându-l foarte atractiv și eficient pentru dezvoltarea rapidă a aplicațiilor, precum și pentru utilizarea ca script sau ca limbaj de legătură (conexiune) pentru a connecta mai multe componente. Python acceptă module și pachete, care scot în evidență modularitatea programului și reutilizarea codului. Interpretorul acestui limbaj și biblioteca standard extinsă sunt disponibile în sursă sau în formă binară, fiind distribuite în mod liber.

Python comparativ cu un alt limbaj de programare:

1. Popularitatea - începând cu anul 2012, Python a crescut constant în popularitate.



*Figura 4.Popularitatea limbajului Python în 2020*

1. Dezvoltarea rapidă - Python are o arhitectură accesibilă, asta face ca scrierea codului să fie mai ușoară și o dezvoltare mai rapidă a produsului final.
2. Numeroase biblioteci și framework-uri - Python are o selecție largă de biblioteci. Există o bibliotecă pentru orice:
   * vizualizarea datelor
   * învățarea automată ("machine learning)
   * știința datelor
   * natural language processing
   * analiză complexă a datelor
   * NumPy
   * TensorFlow
3. Performanță - o critică care apare des despre acest limbaj o reprezintă timpul de rulare, puțin lent.
4. Mentenanța ușoară - Python este intuitiv de citit. Acest lucru face să se poată învăța ușor de către o persoană neexperimentat, dar să și să se rezolve mai ușor probleme codului.
5. Numărul de linii de cod - față de Java sau C, numărul de linii scrise pentru crearea și

dezvoltarea unui produs este relativ mai mic.

1. Simplitate - este deosebit de utilă în citirea codului, atât pentru cel ce scrie linii de

cod propriu-zise cât și pentru persoană care vede pentru prima dată acele linii.

1. Timpul - trăim în secolul vitezei, ne dorim să facem totul cât mai rapid și bine.
2. Scalabilitate fiabilă - nu știi niciodată câți utilizatori îți vor accesa aplicăția de aceea e bine să o faci cât mai fiabilă. Youtube, spre exemple, a pariat pe Python tocmai din acest motiv.

Ca oricare alt limbaj de programare, Python, are și câteva dezavantaje:

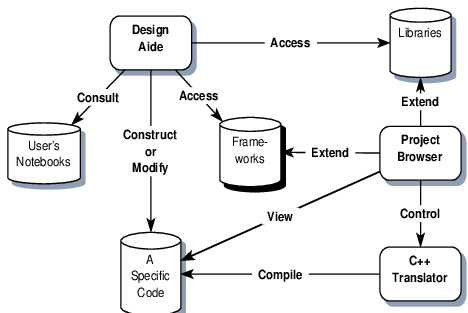
1. Viteza - Python este mai încet decât C/C++ sau Java. După cum știm, compilarea și execuția ajută la funcționarea normală a programul, în Python, aceste procese au loc cu ajutorul unui interpret în locul compilatorului, adică linie cu linie, ceea ce îl încetinește.
2. Mobile Development - Python este puternic în platformele desktop și server, adică un limbaj execelent din partea serverului, dar pentru dezvoltarea mobilă, Python nu este un limbaj foarte bun, asta însemnând că nu este tocmai recomandat pentru această arie a domeniului IT. Există totuși aplicații mobile dezvoltate în acest limbaj, dar foarte puține.
3. Consumul de memorie - pentru orice sarcini cu memorie intensă, Python nu este o alegere inspirată. Memoria consumată în programele care utilizează acest limbaj este foarte mare datorită, flexibilităților tipurilor de date.
4. Accesul la baza de date - majoritatea aplicațiilor sunt conectate la o bază de date. Python este un limbaj robust, cu stres și griji minime. Dar, există unele limitări ale Python cu access la baze de date. În comparașie cu tehnologiile populare, cum ar fi, JDBC (Java Database Connectivity) și ODBC (Open Database Connectivity), s-a constatat că Python este destul de slab, subdezvoltat la acest capitol, dar asa nu înseamnă că nu se pot folosi bazele de date în aplicațiile Python.
5. Dificultate în schimbării cu alte limbaje - programatorii care lucrează în mod constant cu acest limbaj, se obișnuiesc destul de rapid cu caracteristicile sale și cu bibliotecile sale extinse încât se confruntă cu probleme în învățarea sau lucrul cu alte limbaje de programare.
6. Simplitatea - acest punct constituie atât un avantaj, cât și un dezavantaj. Datorită clarității acestui limbaj, programatorilor le este destul de dificil să facă trecerea la alt limbaj. De exemplu, Java este un limbaj destul de complex care necesită un timp de obișnuire, învățare și exersare destul de mare.

Ariile din domeniul IT în care se utilizează acest limbaj:

* Web Development și Internet Development
* Aplicații în GUI (Graphic User Interface) pentru Desktop
* Știință și Aplicații numerice
* Software Development Application
* Jocuri și Grafică 3D
* Baze de date
* Network Programming
* Console-based Application
* Audio-Video Application
* Imagini
* 3D CAD Application
* Computer Vision (Facilități precum detectarea-feței și detectarea-culorilor)
* Machine Learning
* Robotică și Embedded
* Web Scraping
* Scripting
* Inteligența artificială
* Data Analysis

Python mai este cunoscut și pentru framework-urile sale. Framework-ul reprezintă o platformă pentru dezvoltarea aplicațiilor software. Aceasta oferă o bază pe care dezvoltatorii de software pot construi programe pentru o anumită platformă. De exemplu, un framework poate include clase și funcții predefinite care pot fi utilizate pentru procesarea intrării, gestionarea dispozitivelor hardware și interacțiunea cu software-ul sistemului. Acest lucru simplifică procesul de dezvoltare, deoarece programatorii nu trebuie sa reinventeze roata de fiecare dată când lucrează la un proiect nou [7].

Framework-ul este similiar cu interfața de programare a aplicației (API), deși, din punct de vedere tehnic, un framework include o API.



*Figura 5 Framework Environment - Programare Orientată pe Obiecte*

Printre cele mai cunoscute framework-uri în Python se numără:

Django - este open-source, gratuit, full-stack Python. Se caracterizează prin autentificarea, rutarea URL-ului, motorul șabloanelor, ORM (Object-Relational Mapper) și prin schemele bazelor de date.

Pyramid - este open-source, gratuit, bazat pe Python pentru dezvoltarea aplicațiilor web. Principalul său scop este acela de a face cât mai mult cu minimă complețitate.

TurboGears - este open-source, gratuit, construit pe o mulțime de middleware și mai mulre biblioteci și a fost conceput inițial pentru a combina cele mai bune componente ale altor cadre Python. Acest framework vă permite să dezvoltați rapid aplicații web extensibile bazate pe date. Este livrat cu un user-friendly template și un ORM (Object-Relational Mapper) puternic și flexibil.

Web2py - este un framework foarte puternic când vine vorba de geationarea datelor. Este dotat cu propriul IDE (Integrate Development Environment) bazat pe web, care include, printre altele, un editor de coduri, un debugger și o implementare cu un singur click.

Microframeworks - ajută la dezvoltarea aplicațiilor web minimaliste, dar îi lipsește cea mai mare parte a funcționalității unui fraework complet, cum ar fi un template, autentificarea, input validation, conturi etc. Acesta încearcă să ofere doar setul de componente necesare pentru crearea aplicațiilor sau pentru furnizarea funcționalităților necesare într-o anumită sferă.

Flask - are un design modular și îl face adaptabil ușor la nevoile dezvoltatorilor.

Bottle - este tot un microframework, inițial destinat dezvoltatorilor de API-uri, Bottle implementează totul într-un singur fișier sursă. Este recomandat totuși aplicațiilor sub 500 de linii de cod și fără cerințe speciale.

CherryPy - este un framework minimalist care face ca dezvoltarea aplicațiilor Pyrthon să nu fie foarte diferotă de dezvoltarea programelor bazate pe conceptul POO (Programare Orientată pe Obiecte). Beneficiile oferite de aceast framework sunt: serverul integrat multi-threaded. Acest aspect fiind o extensie care ajută la realizarea funcțiilor cerere-răspuns.

Dash - ajută la dezvoltarea aplicațiilor web analitice, fiind în special destinat oameniilor de știință care nu sunt familiari cu web development-ul. Aplicațiile Dash sunt servere web care rulează Flask și comunică cu pachetele de tip JSON pin solicitări HTTP. Frontend-ul lor este redat cu ajutorul componentelor React.js.

## 2.3 Anaconda

Anaconda reprezintă un manager de pachete, o distrbuție de știință a datelor Python sau R, având o colecție de peste 7.500 de pachete open-source. Anaconda este gratuit și ușor de instalat și oferă asistență gratuită pentru comunitate. Este disponibil pentru Windows, Linux sau MacOS [8]

Anaconda este o platformă destinată oamenilor de știință deoarece aceștia pot dezvolta și implementa rapid modele de Machine Learning, Inteligență Aritificială și fără implicare DevOps.

Anacona oferă instrumentele necesare ce pot fi utilizate la:

* Colectarea datelor din fișiere sau baze de date.
* Gestionarea mediilor cu Conda .
* Partajarea, colaborarea și reproducerea proiectelor.
* Implementarea proiectelor în producție cu un singur click al unui buton.

Această distribuție vine la pachet cu foarte multe biblioteci gata pentru a fi folosite, de exemplu:

* Dezvoltarea și pregătirea modelelor de învățare automată cu Scikit-learn, TensorFlow, Theano.
* Analizarea datelor cu scalabilitate și performanță utilizând Dask, NumPy, Panda sau Numba.
* Vizualizarea rezultatelor cu Matplotlib, Bokeh, Datashader și Holoviews.[8]

## 2.4 Jupyter Notebook

Jupyter Notebook reprezintă o aplicație web, open-source care permite dezvoltarea, crearea și partajarea documentelor care conțin cod live, equații, vizulizări, dar și text narativ. Utlizările sale includ: curățarea datelor ('''Data cleaning'), tranformarea datelor, simularea numerică, modelarea statistică, vizualizarea datelor, machine learning și multe altele [9].

JupyterLab este un mediu interactiv de dezvoltare pentru notebook-uri, cod și date Jupyter. JupyterLab este flexibil: configurează și aranjează interfața utilizator pentru a susține o gamă largă de fluxuri de lucru în știința datelor, calcul științific și machine learning. JupyterLab este extensibil și modular: prin scrierea de plugin-uri care adaugă componente noi, le integrează cu cele deja existente.

Pornirea acestuia se realizează prin Anaconda. Acesta folosește mai multe limbaje de programare printre care: Python, Matlab, R Language, C, C++ și multe altele.

## 2.5 ParseHub

ParseHub este un instrument gratuit și puternic de extragere a datelor. Avantajele acestui instrument:

* poți extrage date într-un mod interactiv, rapid și ușor;
* colectarea și stocarea datelor pe serverele lor în mod automat (cloud-based); utilizarea unei utilizarea unei flote de proxy-uri în timpul extragerii datelor;
* programarea colectării datelor;
* utilizarea expresiilor regulare (RegEx);
* utilizarea API-urilor și Web-hooks;
* după colectarea datelor, descărcarea fișierelor sub formă JSON și Excel;

# 3 Parcurgerea proiectului

|  |
| --- |
| **Airbnb** |
| **Python** |
| **Anaconda** |
| **Jupyter Notebook** |
| **ParseHub** |

## 3.1 Explorarea datelor

Următoare imagine descrie pașii realizați în dezvoltarea prezentului proiect, într-un mod ideal. Pașii au fost aplicați, adaptați și modificați astfel încât să se parcurgă fiecare pas, dar să se plieze pe datele avute în acest proiect..



*Figura 6 Etapele proiectului [10]*

Interpretarea imaginii prezentate mai sus și descrierea aferentă fiecărui pas.

* Obținerea sau extragerea datelor poate fi simplă, ca de exemplu folosirea unei interogări în SQL sau complexă, precum extragerea datelor dintr-un site întreg;
* Analiza și curățarea datelor - excluderea valorilor exterioare ce nu pot fi folosite, excluderea duplicatelor, traducerea valorilor etc.;
* Fuzionarea sau combinarea datelor - dacă datele provin din surse diferite, acestea pot fi unite pentru o mai bună interpretare și un rezultat cât mai relevant;
* Efectuarea analizei - realizarea unor statistici, realizarea vizualizării datelor, efectuarea testelor de validare etc.
* Construcția modelelor - întregul proces al analizei datelor poate fi folosit pentru un caz particular, precum realizarea unor modele care pot face diverse previziuni, grupări sau chiar rapoarte automate. În sens general, acest pas este rezultatul tuturor etapelor anterioare;

### 3.1.1 Extragerea datelor

Prima parte a proiectului constă în **extragerea datelor** din site-ul Airbnb și salvarea acestora într-un format de tip CSV.

### 3.1.2 Bibliotecile Beautiful Soup și Scapy

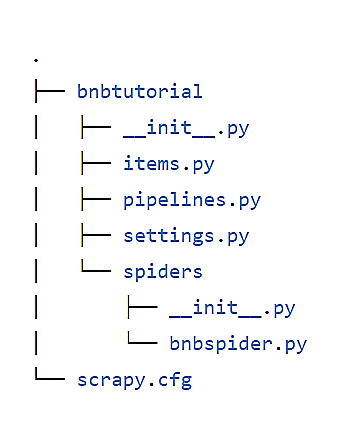
Prima încercare a fost folosirea unui API pentru a extrage datele, dar din păcate, Airbnb nu își mai oferă API-ul decât persoanelor care dezvoltă aplicații monetizate. Astfel încât, primele variante de extragere au fost folosind bibliotecile și framework-urile disponibile în Python, acestea fiind: Beautiful Soup și Scrapy. Procesul a constat în instalarea framework-ului și începerea dezvoltării unui program care să extragă de pe site-ul Airbnb, listările, numele locației, URL-ul listării, prețul, imaginea, numărul de părerile și numărul rating-ului.



*Figura 7. Biblioteci importate*

Librăria json este folosită pentru generarea unui fișier cu această extensie.

Folder-ul proiectului este după cum urmează:



*Figura 8. Structura proiectului Python*

Acesta conține folderul principal care înmagazinează structura proiectului, *bnbtutorial.* Fișierul care conține pachetele obișnuite generate la crearea proiectului este *­­­­\_init\_.py*. Atunci când un pachet obișnuit este importat, acest fișier *\_init\_.py* este implicit executat, iar obiectele pe care le definește sunt legate cu nume din spațiul de nume al pachetului. Fișierul *items.py* conține containerele utilizate pentru colectarea datelor extrase. Acesta furnizează un dicționar precum un API, cu o sintaxă convenabilă pentru declararea câmpurilor disponibile.

După ce un articol a fost extras, acesta este trimis către pipeline, mai excat *pipelines.py* care prelucrează prin mai multe componente care sunt executate secvențial. Fiecare componentă este o clasă Python care implementează o metodă simplă, care execută o acțiune, decidând, dacă acțiune trebuie sau nu să fie procesată.

Întrucât cadrul evolutiv necesită mai multe setări pentru fiecare modul folosit, se folosește un fișier separat pentru a specifica setările corecte pentru o rulare a unei evoluții. Acest fișier, *settings.py*  conține funcții, fiecare generând un set specific de setări pentru a rula evoluția pe o anumită problemă.

*Spiders* este pachetul care conține fișierele programului, *bnbspider.py*  în care se află implementarea programului.

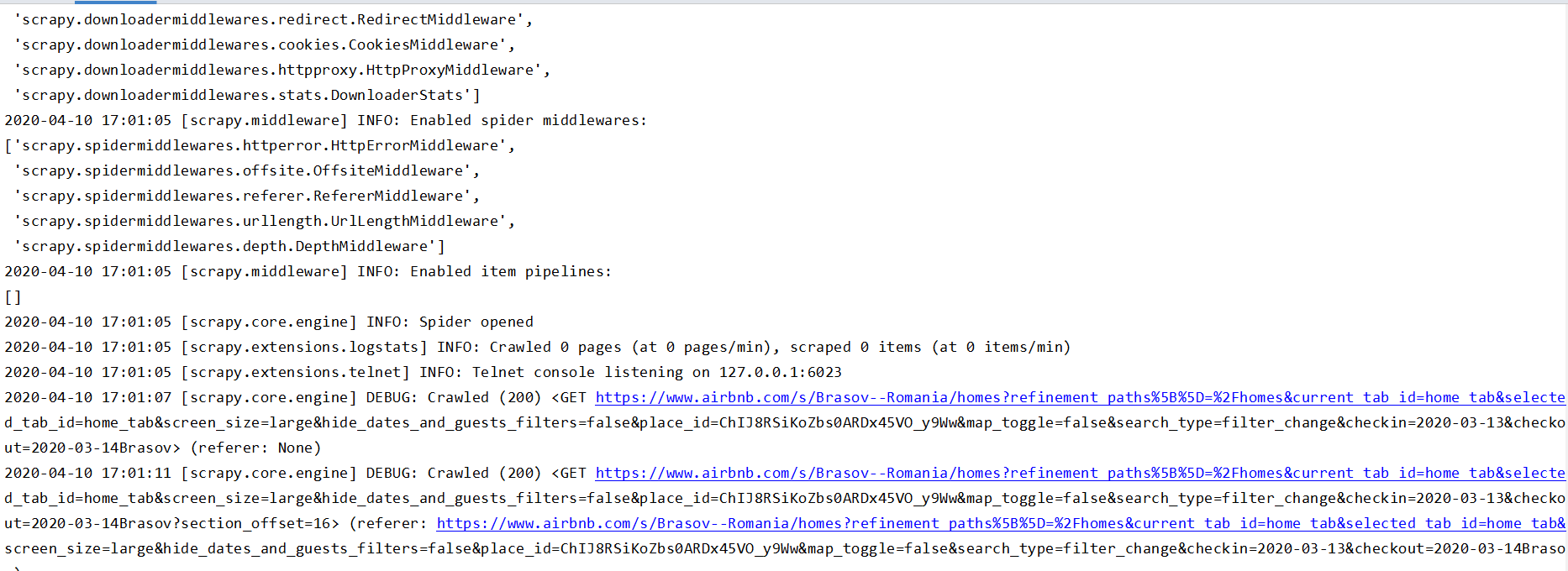
Partea interesantă din extragerea următoarelor date este prin scrierea unui *xpath* care, în acest caz, este ascuns într-un fișier *json*. Clasa *\_fhph4u* reprezintă secțiune principală a paginii Airbnb, mai exact secțiunea care afișază listările. Aceasta este piesa centrală din clasă, funcția care colectează datele și le stochează în obiectul *scrapy,* care este definit pe scurt:



*Figura 9. Colectare date Python*

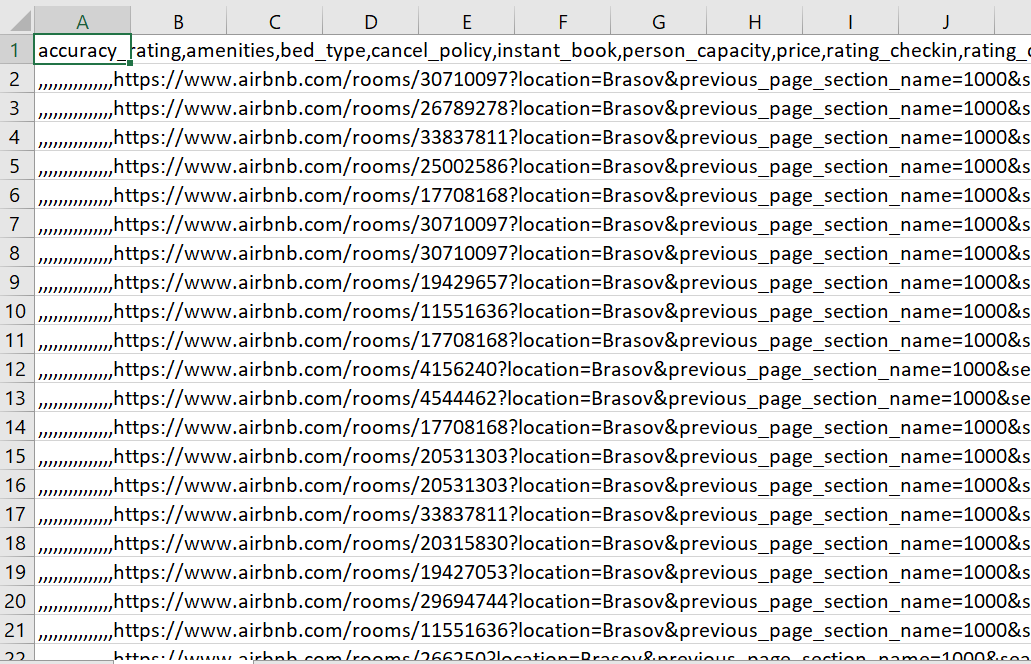
Xpath(XML Path Language) reprezintă un limbaj de interogare pentru selectarea nodurilor dintr-un document XML, fiind un element important în standardul XSLT [11].

Rezultatul generat în urma rulării , în terminal a comenzii *$ scrapy crawl bnbspider -o BrasovAirbnb.csv* este următorul



*Figura 10 Rezultatul programului Python*

Iar *BrasovAirbnb.csv* arată în felul următor:



*Figura 11 Fișierul CSV*

Din păcate, sunt extrase doar adresele URL, deși la prima vedere pare că este aceeași adresă generată de 22 ori (numărul de listări de pe prima pagină a site-ului), acestea au terminația diferită, astfel generând rezultate diferite, de altfel corecte. Codul a suferit mai multe versiuni pe parcurs, dar nereturnând rezultatele așteptate.

### 3.1.3 C Sharp

O altă încercare a fost un crawler dezvoltat in C#, în care se folosesc *HttpClient* și *HttpDocument*.

Clasa *HttpClient* oferă o clasă de bază pentru trimiterea cererilor *HTTP* și primirea răspunsurilor *HTTP* dintr-o resursă identificată de URL.

*HttpDocument* oferă acces pragmatic de nivel superior a unui document *HTML* aflat pe un WebBrowser.

Programul își ia datele necesare din codul HTML, pentru a extrage valorile dorite. Aceste valori sunt: ID, numele listării, preț, tipul listării, adresa URL.

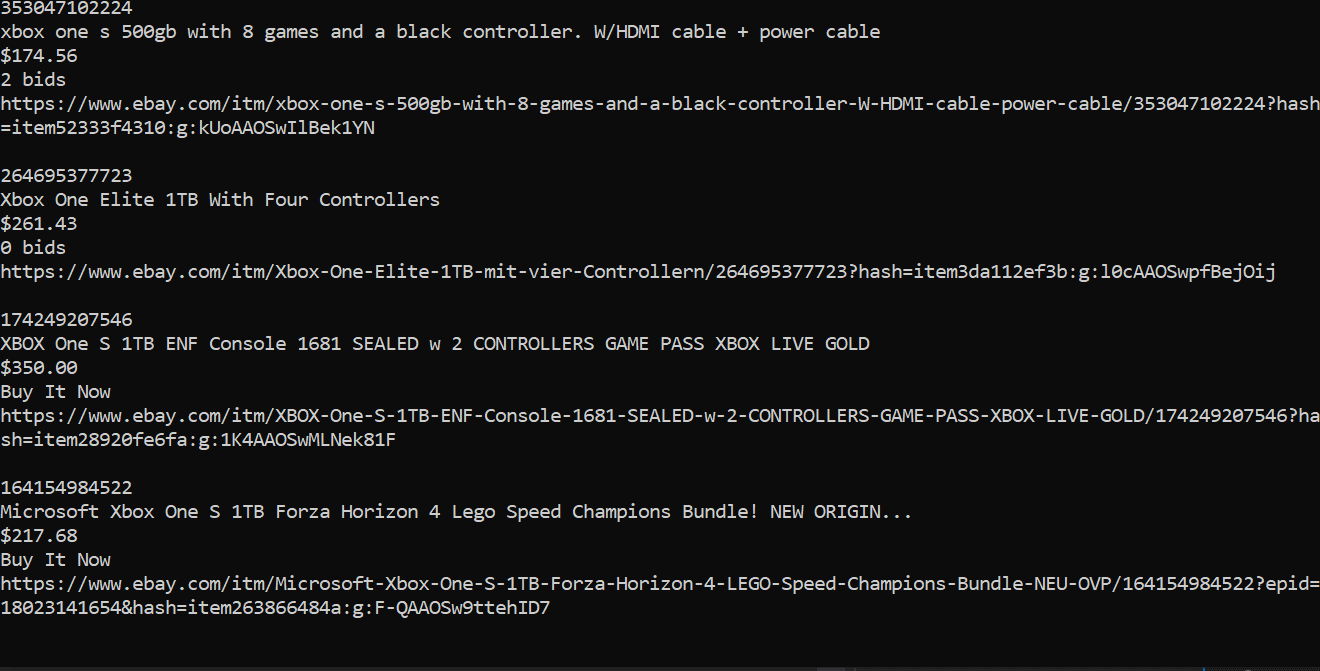


*Codul 1. C Sharp Web Scraper*

De asemenea, am folosit și RegEx(Regular Expresions) pentru a selecta exact partea din pagină dorită, dar și de a curăța spațiile libere din consolă. Programul extrage doar URL și numele listării, nefiind îndeajuns pentru setul de date necesar.

Pentru a determina dacă programul este corect, urmând același principiu, a fost testat și pe site-ul Ebay.com, schimbând doar *GetAttributeValue*, necesar extragerii de pe acel site.

Rezultatele au fost cele așteptate, extrăgând datele dorite. Dar la fel ca în prima variantă, nu este un program care să poată fi folosit pe parcursul proiectului.



*Figura 12 C Sharp Console*

### 3.1.4 NodeJS

O a treia variantă de implementare a fost dezvoltarea unui program în NodeJS, folosind Puppeteer.

Puppeteer este o bibliotecă care oferă un API de nivel înalt pentru a controla Chrome sau Chronium peste protocolul DevTools.

Majoritatea lucrurilor care folosesc Puppeteer sunt:

* generarea capturilor de ecran și PDF ale paginilor;
* extragerea SPA (Single-Page Application);
* automatizarea trimiterii formularului, testarea interfeței de utilizator, introducerea tastaturii etc.;
* crearea unui mediu de testare automatizat actualizat;
* executarea testelor direct în cea mai recentă versiune Chrome utilizând funcții JavaScript și un browser;
* capturarea cronologiei site-ului propriu pentru a ajuta la diagnosticarea problemelor de performanță;
* testarea extensiilor Chrome;

Cheerio implementează un subset de jQuery de bază, elimină toate inconsistențele DOM(Document Object Model) și transferul browserului din biblioteca jQuery, dezvăluind API-ul. Acesta funcționează cu un model de DOM foarte simplu și consistent, analizând, manipulând și redând datele foarte eficient. Cheerio poate analiza aproape orice document HTML (HyperText Markup Language) și XML(Extensible Markup Language).

Cheerio nu interpretează datele precum un browser web, adică nu redă vizual rezultatul, nu aplică CSS, încarcă resursele externe sau execută JavaScript.

Yarn este un package manager care se dublează ca project manager. Avantaje:

- împărțirea proiectului în subcomponente păstrate într-un singur repository;

- garantează o instalare care funcționează la fel de fiecare dată când este folosit;

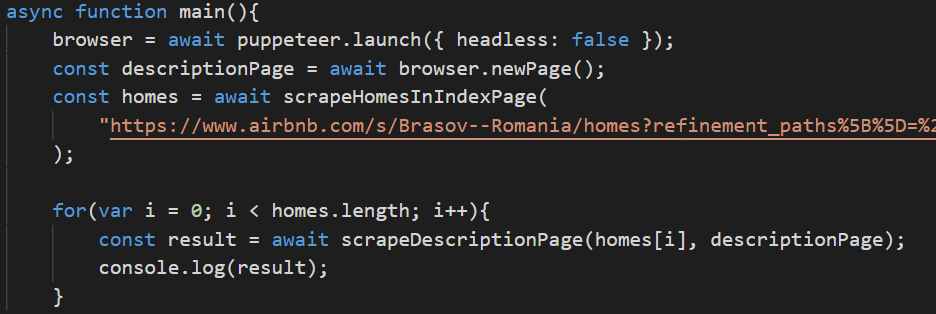
- furnizează o documentație bine pusă la punct;



*Codul 2. NodeJS Web Scaper biblioteci*

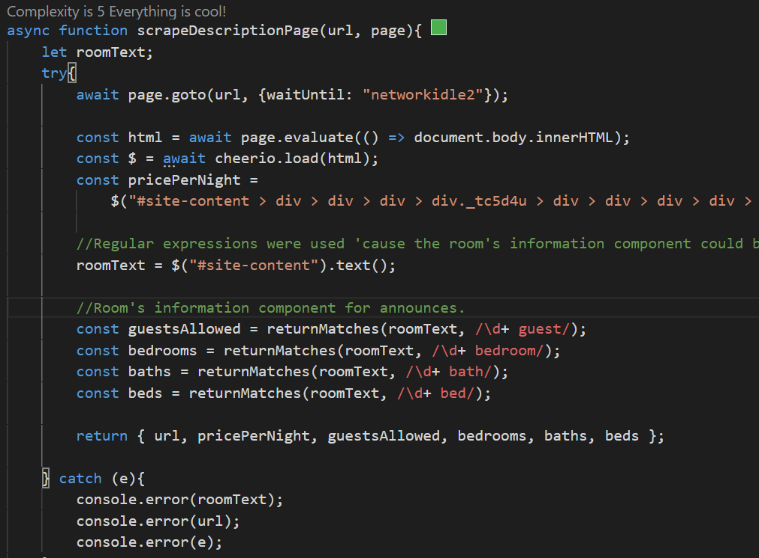
Programul funcționează astfel:

1. Deschide pagina web folosind Puppeteer, folosind în terminal comanda *node index.js*;



*Codul 3. NodeJS deschidere Puppeteer*

2. Extragerea adreselor URL a fiecărei listări;



*Codul 4. NodeJS Web extragere URL*

3. Deschiderea listărilor într-o altă pagină;

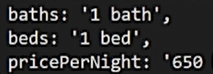
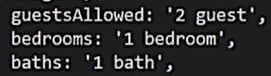
4. Extragerea prețului;



*Codul 5. Node JS extragere prețuri*

5. Extragerea numărului de paturi, băi, dormitoare, locatari;

Rezultatul: unele date lipsesc, altele apar. De exemplu dintr-o listare lipsește prețul, iar în altă listare lipsește numărul băilor.



*Figura 13 Rezultate NodeJS*

### 3.1.5 Octoparse

A patra variantă este folosirea unui tool, mai exact Octoparse. Un prim dezavantaj al acestui tool este licența gratuită doar pentru 14 zile. Avantajul său este crearea unor șabloane deja pestabilite, mai exact introduci adresa URL și acesta extrage totul automat. De asemenea, există și varianta mai avansată, unde îți poți face singur algoritmul de extragere. Acesta funcționează foarte bine, extrage toate obiectele selectate din pagina HTML, le descarcă în format CVS, dar în caz că dorești să aduci modificări algoritmului, nu se mai poate. Trebuie să reiei totul de la zero, în termen de 14 zile, altfel trebuie să plătești o licență destul de costisitoare.



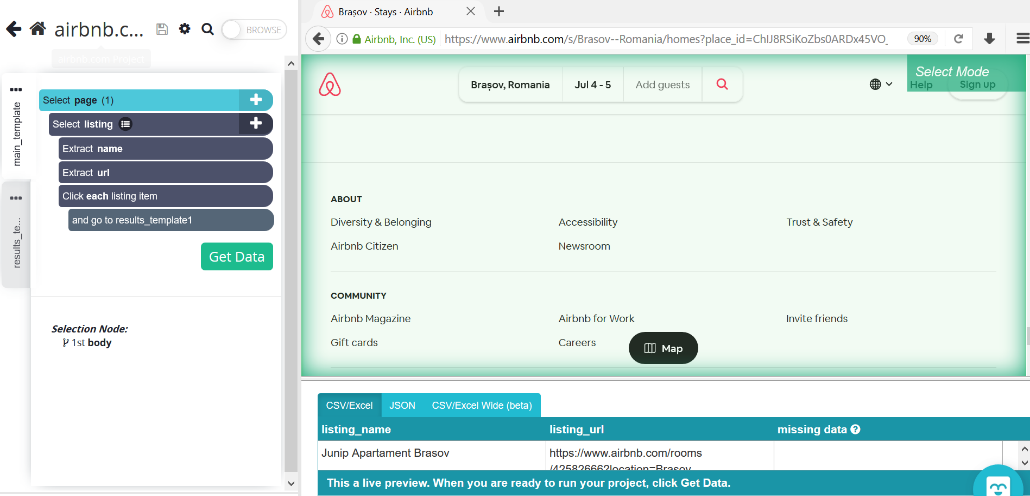
*Figura 14 Octoparse șablon Airbnb*

### 3.1.6 ParseHub

Ultima variantă, este ParseHub. O consider cea mai eficientă dintre toate deoarece este rapidă, intuitivă și gratuită. Oferă o gamă largă de tutoriale pentru a putea înțelege modul de funcționare, dar după puțin timp petrecut muncind în acest tool, devine foarte familiar.

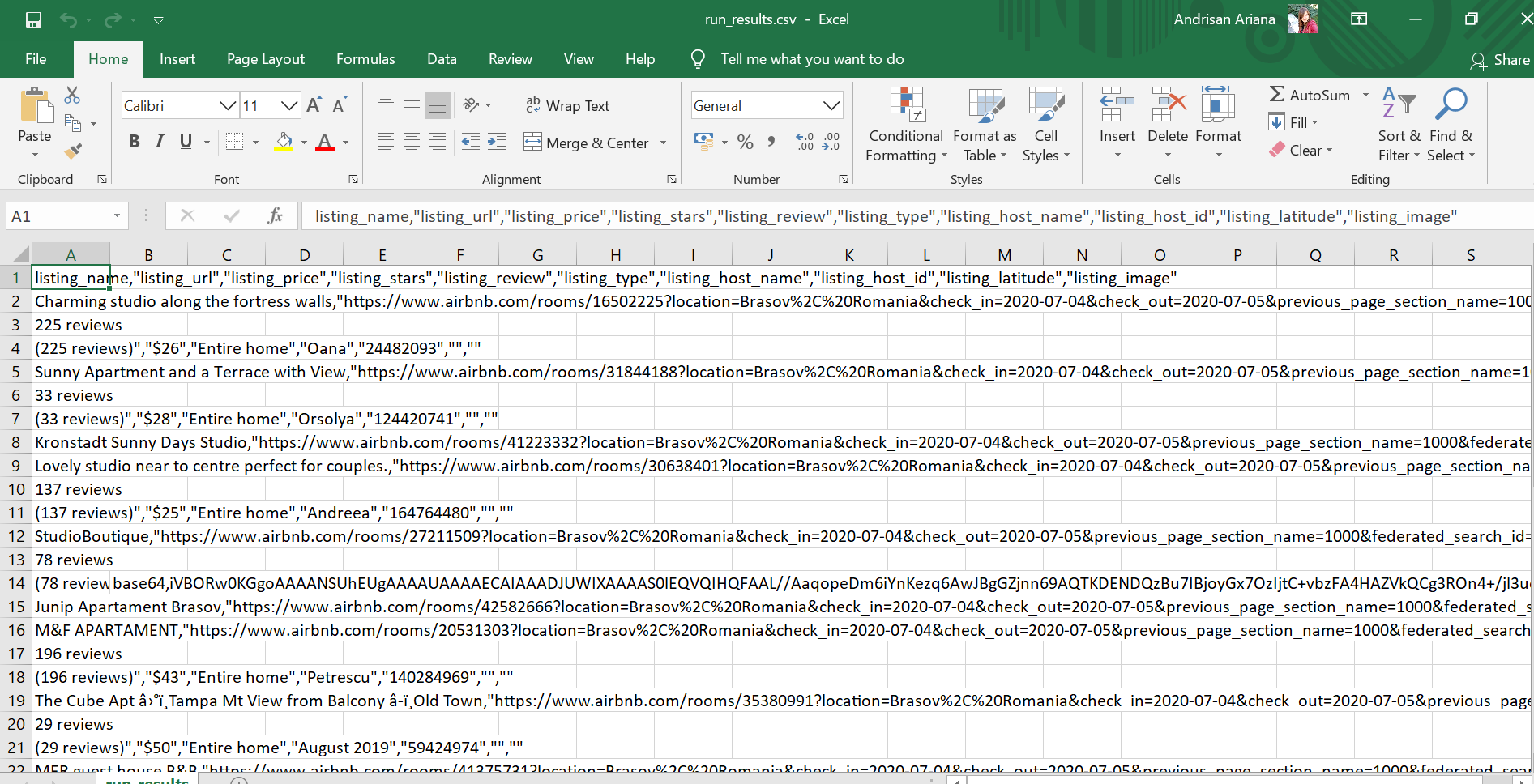
Poți modifica roborul ori de câte ori este nevoie, fără prea mari dificultăți. Acesta fiind tool-ul care îndeplinește cerințele impuse și pe care voi continua să lucrez în cadrul acestui proiect.

O parte din algoritmul extragerii datelor:



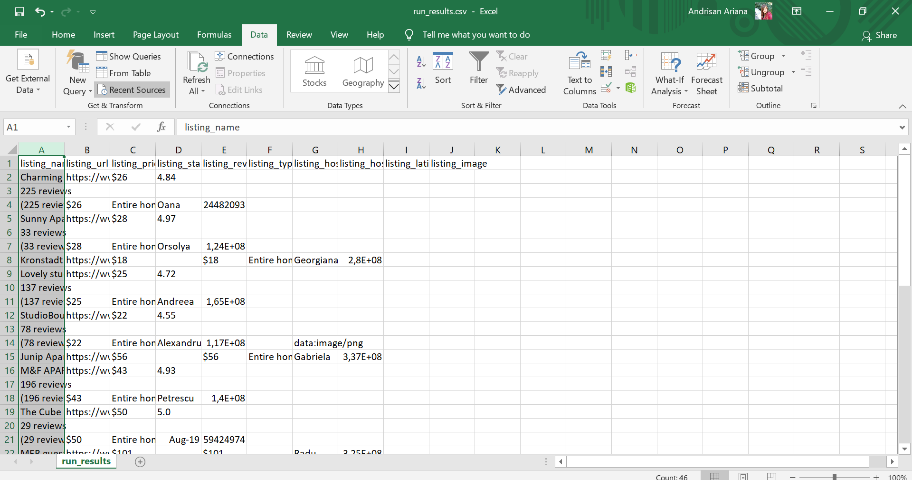
*Figura 15 ParseHub Airbnb Crawler*

Rezultatul apare în felul următor:



*Figura 16 Fișierul CSV neformatat*

Dar se pot așeza folosing funcția *Text to Columns* din Excel. Rezultatul:



*Figura 17 Fișierul CSV formatat*

O mică sinteză asupra informațiilor prezentate mai sus:

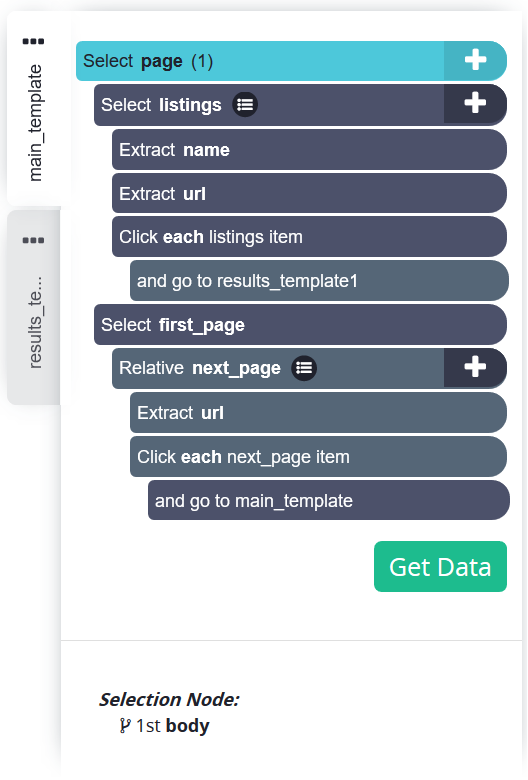
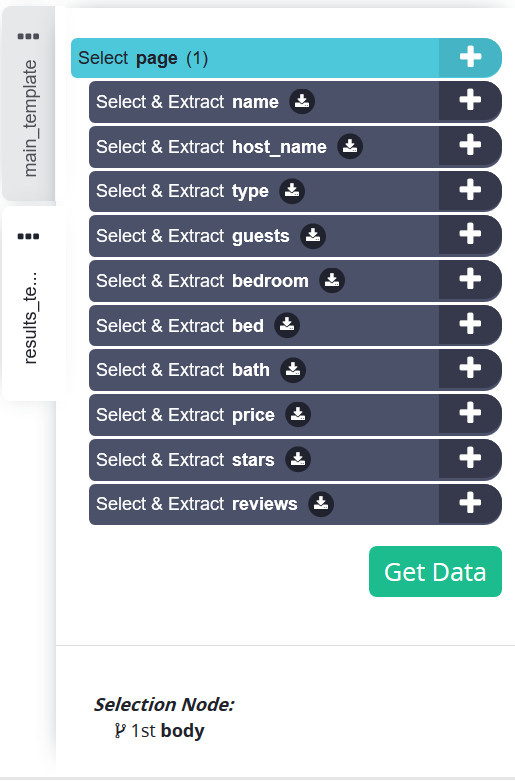
*Tabelul 2. Rezumat Tehnologii*

|  |  |  |
| --- | --- | --- |
| Denumire | Avantaje | Dezavantaje |
| Python | Aprofundare limbaj | Imposibilitatea spargerii proxy-urilor sau CAPTCHA. Rânduri lipsă în rezultate |
| C# | Aprofundare limbaj | Imposibilitatea spargerii proxy-urilor sau CAPTCHA ascunse în spatele codului. |
| NodeJS | Aprofundare limbaj | Rezultatele finale nu sunt conform așteptărilor. |
| Octoparse | Șabloane, ușor de folosit | Versiune gratuită doar 14 zile |
| ParseHub | Gratuit | Diverse limitări |

După stabilirea tool-ului folosit, urmează implementarea celulelor de cod în Jupyter pentru a afișa primele rezultate.

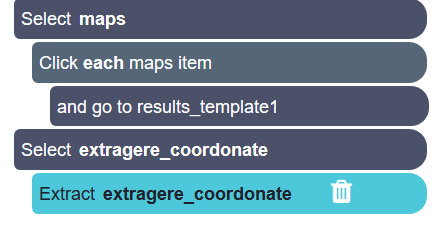
Revenind la tool-ul ParseHub, a fost dezvoltat un robot care să poată prelua datele prezentate mai sus. Astfel avem un robot care conține 8 funcții. Mai exact, selectarea întregii pagini, selectarea listărilor din care se va extrage numele și adresa URL, apoi click pe fiecare listare în parte pentru a putea accesta templat-ul result\_template. Și în final un loop pentru paginare. Principala problemă întâmpinată, a fost apariția unui pop-up ascuns, care nu apare și în momentul când navighezi normal pe pagina web. Acesta are rolul de a bloca fluxul de date al acestor roboți. Am reușit să îl ascund, pentru a nu fi recunoscut de crawler, dar nu funcționează în totalitate la fiecare rulare a acestuia.

În continuare avem pagina result\_template ce conține numele listării, host\_name, tipul cazăriii, vizitatorii, bai, paturi, prețul cazării, stelele și părerile oferite de persoana care a vizitat respectivele locații.

*Figura 18 Pașii Crawler-ului ParseHub*

Pentru a determina locația se va selecta mapa, și se va extrage adresa URL deoarece în această adresă se află coordonatele pentru latitudine și longitudine. De asemenea, pentru extragere vom avea nevoie și de RegEx, astfel:  **\@(-?[\d\.]\*). pentru latitudine și  \@[-?\d\.]\*\,([-?\d\.]\*)+ pentru longitudine.**

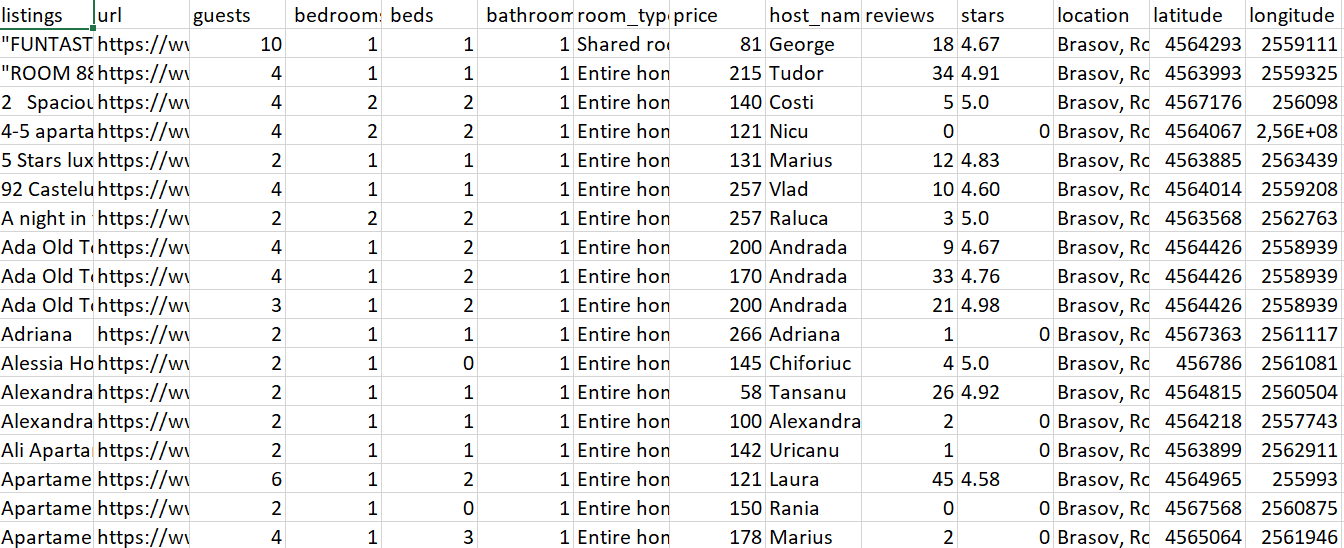


*Figura 19 Pașii extragerii coordonatelor din hartă*

După finalizarea extragerilor, a rezultat un fișier CSV cu 300 de listări, exact numărul de listări de pe pagină. Doar că la o privire mai atentă unele listări erau din zonele limitrofe ale Brașovului și de asemenea, existau și unele duplicate.

Pasul al doilea din proiect constă în **curățarea datelor,** mai precis, eliminarea datelor nefolositoare, eliminarea duplicatelor și de aseamenea, corectarea eventualelor greșeli.

În final a rezultat un fișier CSV cu 269 de date înregistrate.



*Figura 20 CSV final*

Având datele extrase, le putem analiza și vizualiza. Interpretarea graficelor rezultate vor fi expuse în următorul capitol.

## 3.2 Analiza datelor

Având toate datele necesare, putem începe analiza acestora. Interpretarea graficelor va fi dezvoltată în capitolul următor.

Analiza datelor reprezintă un proces de inspectare, aranjare, curățare și interpretare a datelor având ca scop descoperirea unor informații utile, ce pot fi utilizate în alte proiecte, fiind util peentru descoperirea informațiilor necesare pentru luarea deciziilor în afaceri [12].

Pentru tratarea corespunzătoare a acestui aspect, am folosit următoarele biblioteci:

* NumPy - este un pachet de bază pentru calcularea numerică în Python. Pandas, Matplotlib, Statmodels se bazează pe NumPy [13].
* Pandas - ajută la explorarea, curățarea și procesarea datelor, fiind util mai ales în cazul datelor tabulare [14].
* Matplotlib - este o bibliotecă pentru crearea vizualizărilor statice sau animate în Python [15].
* Seaborn - oferă o interfață la nivel înalt pentru desenarea graficelor [16].

### 3.2.1 Citirea datelor

Pentru a putea vedea fișierul CSV în Python, se folosește funcția *read\_csv()*, iar datele sunt valori separate prin punct și virgulă *sep=';'* .

df = pd.read\_csv('airbnb\_dates.csv', sep=';')

df.head()

*Codul 1. Citirea și afișarea CSV-ului*

Rezultatul este identic cu fișierul CSV prezentat mai sus.

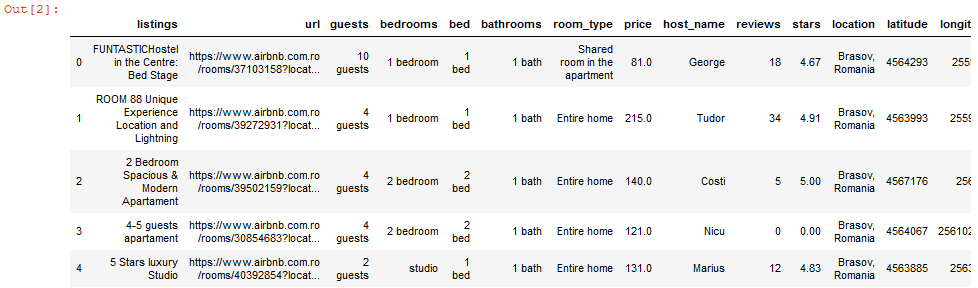


Figura 21 Rezultatul citirii datelor în Jupyter Notebook

Din acest tabel se pot extrage diverse informații din care putem afla tipul de date aflat în tabel, media aritmetică a anumitor coloane, indexul rândurilor din fiecare coloană, valuarea coloanelor, dimensiune dataframe.

df.index.name = 'Id'

df.index

df.values

df.head()

df.ndim

df['listings'].value\_counts()

df['price'].head()

df['reviews'].max()

df['reviews'].min()

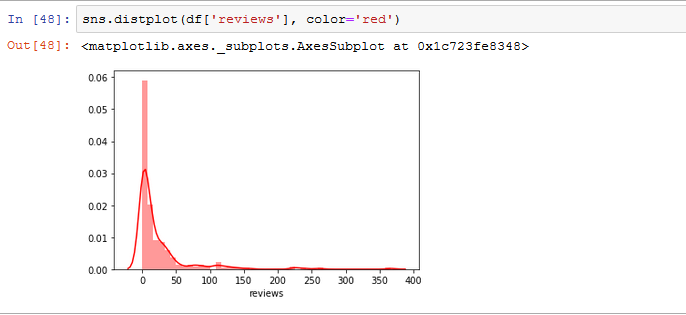
df['reviews'].mean()

df['reviews'].sum()

*Codul 2. Statistici - rezumat*

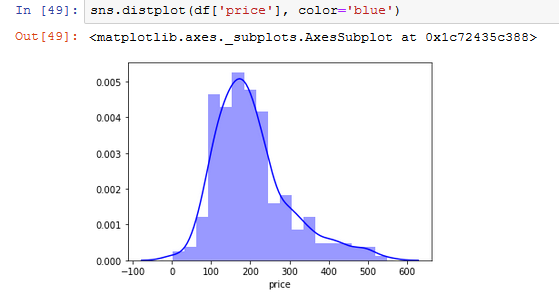
## 3. 3 Vizualizarea datelor

După analizarea datelor și afișarea numerelor care vor fi folosite mai târziu în proiect, se pot vizualiza datele, cu ajutorul librăriei Seasborn. De exemplu coloana *reviews*  are următorul grafic.



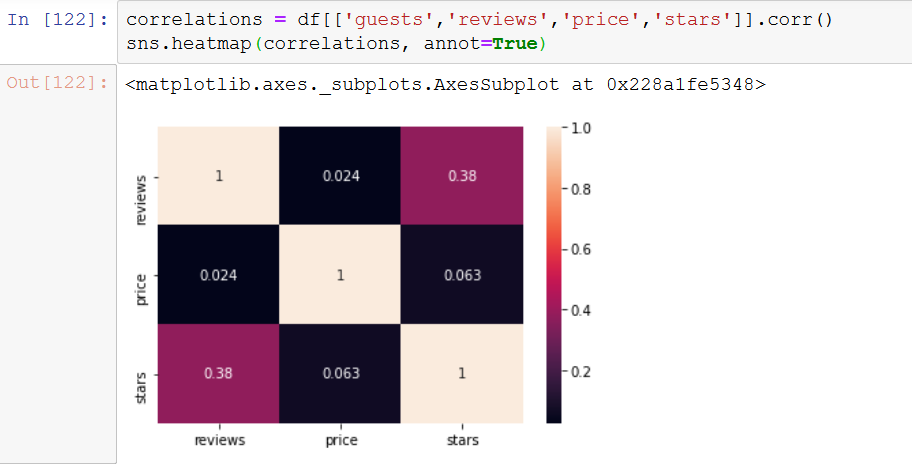
*Figura 22 Vizualizarea coloanei reviews*

Pentru comparație se poate lua coloana *price*, unde graficul are valori mult mai reprezentative datorită unui set mai mare de eșantioane.



*Figura 23 Vizualizarea coloanei price*

Pentru o statistică mai clară, concisă și mai ușor de interpretat, putem corela datele.



***Figura 24 Corelare***

Datele pot fi grupate, sortate sau se poate aplica și selecția codiținată, tablouri booleane care ne pot face munca mai ușoară, mai ales când vorbim de seturi mari de date.

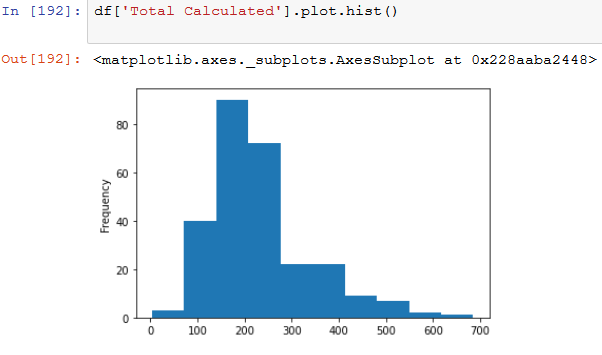
Pentru a combina instrucțiuni condiționale se folosesc operatorii logici: AND, OR, NOT sau operatorii pe bit care acționează asupra oeranzilor ca și cum ar fi șiruri de cifre binare, operând bit cu bit: &(AND), | (SAU), ~(NOT), ^(XOR), »(SHIFTARE LA DREAPTA), «(SHIFTARE LA STÂNGA)

df\_small.loc[(df\_small['price'] >= 100) & (df\_small['stars'] > 4.5)]

*Codul 3. Combinarea operațiilor*

## 3.4 Curățarea datelor

Curățarea datelor a fost efectuată înaintea începerii dezvoltării codului prin eliminarea duplicatelor din CSV, eliminarea datelor nerelevante sau corectarea greșelilor. În cod, curățarea datelor a presupus eliminarea unor coloane sau rânduri nefavorabile operției în curs de efectuare. Totodată, a fost efectuat un calcul total ("Total Calculated") între următoarele coloane: vizitatori ("guests"), preț("price"), păreri("reviews"), stele("stars") și vizualizarea acestuia folosing un grafic de tipul plot.



*Figura 25 Totalul calculat*

## 3.5 Machine Learning

### 3.5.1 Definiție

Învățarea automată este o aplicație a inteligenței artificiale(IA) care oferă sistemelor capacitatea de a învăța și îmbunătății automat din experiență, fără a fi programat explicit. Învățarea automată se concentrează pe dezvoltarea de programe de calculator care pot accesa date să să le utilizeze pentru a învăța singure [17].

O altă definiție care se aplică mai exact pe ceea ce urmează a fi prezentat este:

Învățarea automată (Machine Learning) reprezintă construirea unor modele, ce pot fi instruite folosind date, care pot fi apoi utilizate pentru a realiza predicții asupra noilor date.

Modelul de învățare automată examinează relațiile statistice dintr-un set de date cu rezultate definite și apoi aplică ceea ce a învățat despre relațiile respective pentru a analiza și prezice rezultatele printr-un nou set de date.

### 3.5.2 Modul de funcționare

1. Examinarea unui set mare de date despre prețurile locuințelor.
2. Găsirea tiparelor și a relațiilor statistice între caracteristicile unei case și prețul acesteia (variabila țintă), inclusiv tiparele care ar putea să nu fie evidente pentru un om care privește datele.

### 3.5.3 Predicția prețului

O provocare cu care se confruntă gazdele Airbnb este determinarea prețului optim de chirie pe noapte. În multe zone, chiriașii arputea fi prezentați cu o selecție bună de înregistrări și pot filtra după criterii precum prețul, numărul de dormitoare. tiplul camerei și multe altele. Întrucât Airbnb este o piață, suma pe care o gazdă o poate percepe pe noapte este strâns legată de dinamica pieței.

Pentru stabilirea corectă a prețului, astfel încât gazda să poată obține profit, dar și să poată atrege clienți, există câteva strategii. Aceste strategii vor fi folosite și în dezvoltarea codului:

* Găsirea înregistrărilor care sunt similare cu înregistrarea ce urmează a fi postată;
* Media prețului listat pentru cele mai similare listări;
* Setarea prețului de listare, la prețul mediu calculat;

Dar cum acest lucru manual repetat de mai multe ori, ar fi o pierdere de timp, dar și un proces greoi, tot acest proces poate fi rezolvat printr-un model de învățare automată folosind tehnica numită "K-nearest neighbors" sau "Algoritmul celor apropiați".

### 3.5.4 Algoritmul K-nearest neighbors

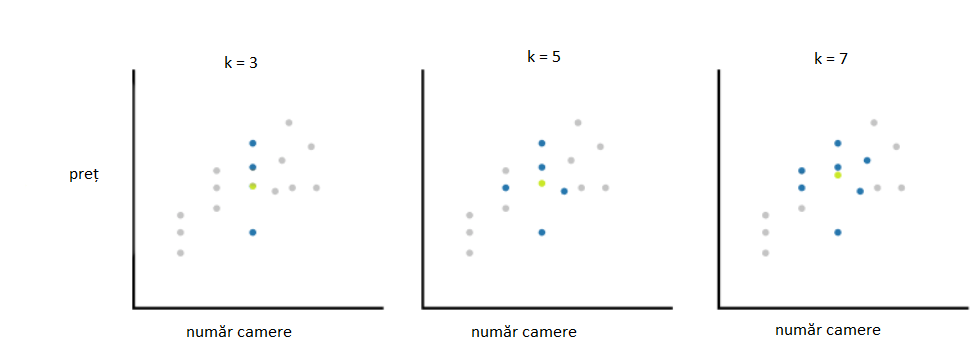
KNN este o metodă non-parametrică utilizată pentru clasificare și regresie. În ambele cazuri, intrarea constă din k exemple de instruire și k cele mai apropiate în spațiul de caracteristici.

Rezultatul depinde dacă KNN estte aplicat pentru regresie sau clasificare.

În regresie KNN, ieșirea este valoarea proprietății pentru obiect. Această valoare este media valorilor k dintre vecinii apropiați. Regresia ia un grup de variabile aleatorii, care se crede că prezice Y și încearcă să găsească o relație matematică între ele. Această relație este de obicei sub forma unei linii (regresie liniară) care aproximează cel mai bine toate punctele de date individuale [17].

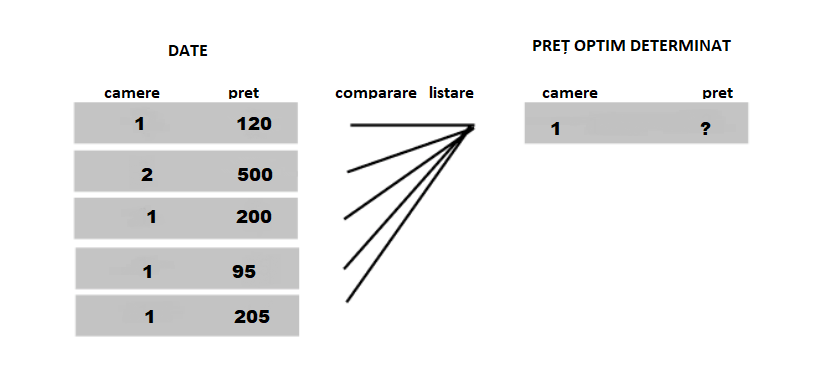
În clasificarea KNN, rezultatul este un membru al clasei. Un obiect este clasificat printr-un vot plural al vecinilor săi, obiectul fiind atribuit clasei cele mai frecvente dintre k vecinii săi cei mai apropiați (k este un număr întreg pozitiv, de obicei mic). Dacă k = 1, atunci obiectul este pur și simplu atribuit clasei acelui vecin cel mai apropiat [18].

Mai întâi, selectăm numărul de înregistrări similare k cu care vrem să le comparăm.



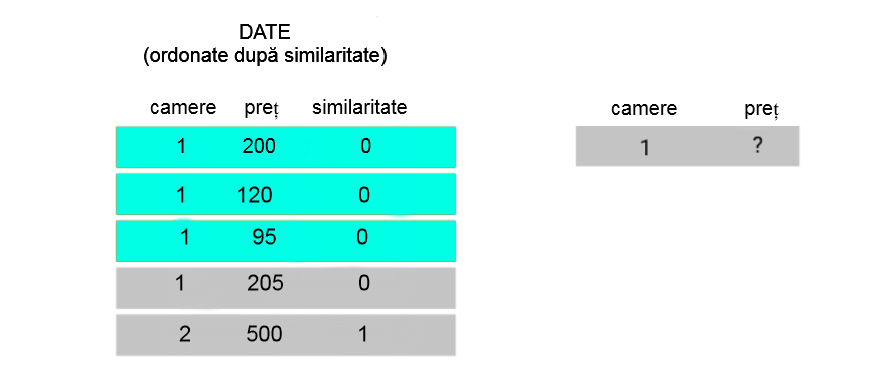
*Figura 26 Numărul de k camere*

În al doilea rând, trebuie să calculăm cât de asemănătoare este fiecare listă cu a noastră folosind valori de similitudine.



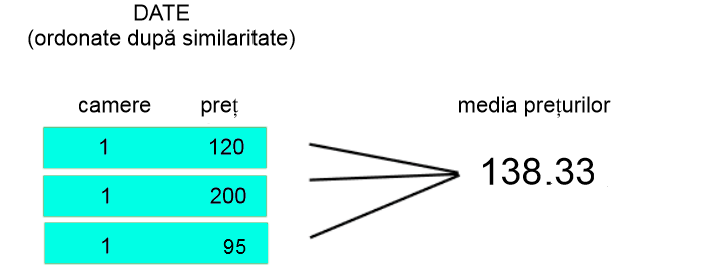
*Figura 27 Valori de similitudine*

În al treila rând, se clasifiă fiecare listă folosind metrica de similitudine și selectarea primelor k listări.



*Figura 28 Selectarea primelor k listări*

În ultimul rând, se calculează prețul mediu pentru k listări și îl utilizăm ca preț de listă.



*Figura 29 Calculare preț de listă*

In [1]:import pandas as pd

dc\_listings = pd.read\_csv('airbnb\_datesv2.csv', sep=';')

print(dc\_listings.shape)

dc\_listings

Out[1]: (268, 14)

*Codul 4. Afișarea tabelului CSV utilizând biblioteca pandas*

Interpretarea rezultatului reprezintă 268 de rânduri și 14 coloane, datele aflate în CSV.

### 3.5.5 Distanța Euclidiană

Pentru prezicerea unei valori precum prețul, metoda de similitudine utilizată este distanța euclidiană, având următoarea formulă generală:

[19]

### 3.5.6 Modelul de bază KNN

Prima oară se calculează distanța folosind doar funcția de vizitatori ("guests"), într-un spațiu de locuit din setul de date propriu. Pentru asta se utilizează funcția NumPy: np.abs() care va determina valoarea absolută.

In [2]: import numpy as np

our\_acc\_value = 3

first\_living\_space\_value = dc\_listings.loc[0,'guests']

first\_distance = np.abs(first\_living\_space\_value - our\_acc\_value)

print(first\_distance)

Out[2]: 7

*Codul 5. Calcularea distanței*

Calcularea distanței pentru mai multe date pentru a putea face o comparație.

In [3]: dc\_listings['distance'] = np.abs(dc\_listings.guests - our\_acc\_value)

dc\_listings.distance.value\_counts().sort\_index()

Out[3]: 0 26

1 191

2 16

3 21

4 3

5 6

7 2

11 1

13 2

Name: distance, dtype: int64

*Codul 6. Compararea calculării distanței.*

Există 26 de înregistrări care au distanța de 0 și pot găzdui astfel același număr de persoane ca și listarea noastră.

Trebuie precizat că cea mai mică distanță euclidiană este zero. Dacă se iau în considerare doar primele 5 valori, având distanța 0, predicțiile noastre ar ajuta la ordonarea datelor existente. Dar prim randomizarea ordonării observațiilor și apoi selectarea primelor 5 valori va face mai complexă, dar și mai exactă predicția. Pentru randomizare se folosește DataFrame.sample().

In[4]: dc\_listings = dc\_listings.sample(frac=1,random\_state=0)

dc\_listings = dc\_listings.sort\_values('distance')

dc\_listings.price.head()

Out[4]:254 160.0

93 165.0

99 185.0

95 165.0

253 115.0[

Name: price, dtype: float64

*Codul 7. Sortarea valorilor*

Transformarea coloanei într-un tip de date float înainte de a calcula media celor cinci valori.

In [5]: dc\_listings['price'] = dc\_listings.price.astype(float)

mean\_price = dc\_listings.price.iloc[:5].mean()

mean\_price

Out[5]: 158.0

*Codul 8. Media prețurilor*

Acesta este rezultatul primei predicții. Modelul nostru a găsit un preț de 158.0 adecvat listării pentru găzduirea a 3 vizitatori într-un apartament. Nu se știe cât de exact este prețul rezultat, astfel neputând fi optimizat și îmbunătățit pe viitor.

### 3.5.7 Evaluarea modelului

Evaluarea modelului sau mai exact testarea acestuia se face prim împărțirea setului de date în 2 părți:

* Setul de antrenare care conține 75% din rânduri
* Setul de teste care conține restul rândurilor, 25%

In [6]: train\_df = dc\_listings.copy().iloc[:268]

test\_df = dc\_listings.copy().iloc[1:]

In [7]: def predict\_price(new\_listing\_value,feature\_column):

temp\_df = train\_df

temp\_df['distance'] = np.abs(dc\_listings[feature\_column] - new\_listing\_value)

temp\_df = temp\_df.sort\_values('distance')

knn\_5 = temp\_df.price.iloc[:5]

predicted\_price = knn\_5.mean()

return(predicted\_price)

In [8]: test\_df['predicted\_price'] = test\_df.guests.apply(predict\_price,feature\_column='guests')

*Codul 9. Evaluarea modelului*

### 3.5.8 RMSE (Root Mean Squared Error) - Eroarea rădăcinii medie pătrate

Eroarea rădăcinii medie pătrate este o măsură frecvent utilizate a diferențelor dintre valori prezise de un model sau un estimator și valorile observate. Aceste abateri sunt numite reziduuri atunci când calculele sunt efectuate pe eșantionul de date care a fost utilizat pentru estimare și se numesc erori (sau erori de predicție) atunci când sunt calculate din eșantion [20].

,

unde n reprezintă numărul de rânduri din setul de date.

In [9]: test\_df['squared\_error'] = (test\_df['predicted\_price'] - test\_df['price'])\*\*(2)

mse = test\_df['squared\_error'].mean()

rmse = mse \*\* (1/2)

rmse

Out[9]: 87.82022942215707

*Codul 10. Eroarea rădăcinii medie pătrate*

RMSE-ul rezultat este de 87.83 RON.Deoarece mai întâi se rezolvă radicalul și apoi luăm rădăcina pătrată, unitățile RMSE sunt aceleași cu valoarea prezisă, ceea ce face mai ușor de observat scala de eroare.

Datorită faptului că avem o scală mare, predicția încă nu este destul de exactă.

Pentru compararea rezultatelor se poate compara diferite modele de predicție.

In[10]: for feature in ['guests']:

test\_df['predicted\_price'] = test\_df.guests.apply(predict\_price,feature\_column=feature)

test\_df['squared\_error'] = (test\_df['predicted\_price'] - test\_df['price'])\*\*(2)

mse = test\_df['squared\_error'].mean()

rmse = mse \*\* (1/2)

print("RMSE pentru coloana {}: {}".format(feature,rmse))

*Codul 11. Comparerea rezultatelor modelelor de predicție*

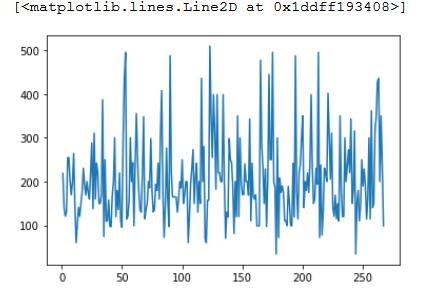
Analog pentru fiecare coloană: guests, price, reviews.

Out[10]:RMSE pentru coloana guests: 87.82022942215707

RMSE pentru coloana price: 4.25553065651504

RMSE pentru coloana reviews: 94.39816240109825

Vizualizarea prețurilor prezise până acum.



*Figura 30 Vizualizarea prețurilor*

După cum se poate observa există atât o creștere a pețurilor, cât și o descreștere.

Calcularea distanței euclidiene pentru valori multiple: guests și price. Până acum, calcularea distanței euclidiene s-a făcut folosind logica ecuației sale, dar putem folosi funcția distance.euclidean() din scipy.spatial, care ia doi vectori ca parametrii și calculează distanța euclidiană între ei.

In[11]:from scipy.spatial import distance

first\_listing = normalized\_listings.iloc[0][['guests','price']]

fifth\_listing = normalized\_listings.iloc[20][['guests','price']]

first\_fifth\_distance = distance.euclidean(first\_listing,fifth\_listing)

first\_fifth\_distance

Out[11]:321.0015576286196

*Codul 12. Distanța euclidiană*

Acum putem concluziona prețul exact al listărilor. Valoarea prezisă, cea mai realistă este 321.00.

Scikit-learning este cea mai populară bibliotecă pentru machine learning ("învățare automată"). Are funcții integrate pentru toți algoritmii majori de învățare și un flux de lucru simplu și unificat.

In [12]:from sklearn.metrics import mean\_squared\_error

two\_features\_mse = mean\_squared\_error(norm\_test\_df['price'], two\_features\_predictions)

two\_features\_rmse = two\_features\_mse \*\* (1/2)

print(two\_features\_rmse)

Out [12]: 83.75648167206164

*Codul 13. Utilizarea algoritmului pentru RMSE*

Se poate observa că este implementarea algoritmului prezentat mai sus, dar într-un mod mai simplu. De asemenea, se observă o diferență între rezultatul codului de mai sus, unde s-a utilizat funcția *mean\_squared\_error()* și algoritmul implementat "manual". Acestă diferență este datorată diferențelor de randomizare, cât și a unor diferențe ușoare de implementare a algoritmului.

Scikit-learn conține multe lucruri bune, dar un avantaj semnificativ este reprezentat de iterarea rapidă. Următoarea secvență de cod are patru funcții, în loc de două, astfel putând observa dacă apar diferențe.

In [13]: knn = KNeighborsRegressor(algorithm='brute')

cols = ['guests','reviews']

knn.fit(norm\_train\_df[cols], norm\_train\_df['price'])

four\_features\_predictions = knn.predict(norm\_test\_df[cols])

four\_features\_mse = mean\_squared\_error(norm\_test\_df['price'], four\_features\_predictions)

four\_features\_rmse = four\_features\_mse \*\* (1/2)

four\_features\_rmse

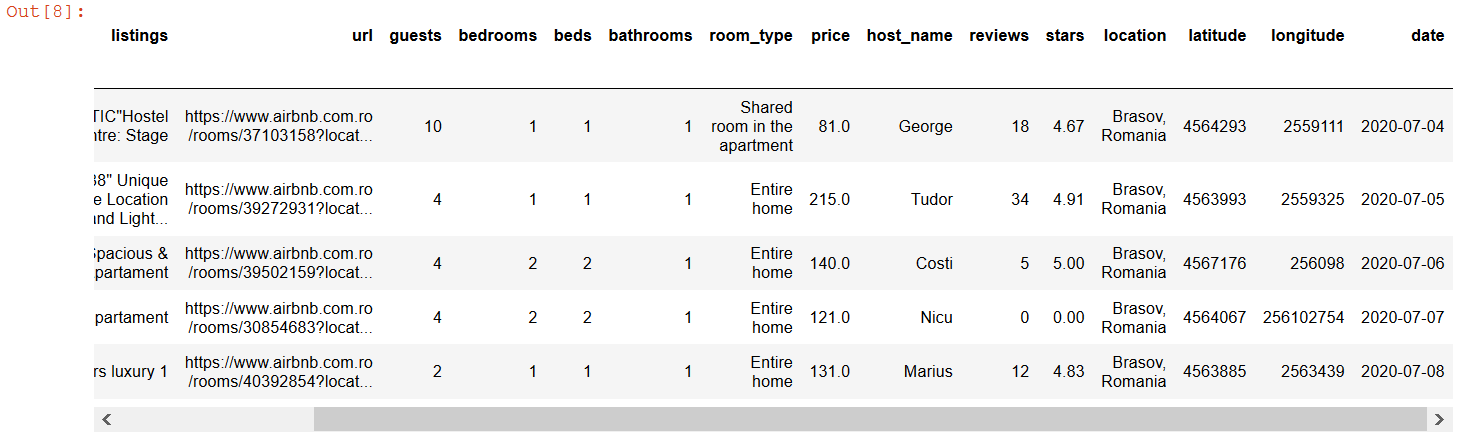
Out [13]: 83.75648167206164

*Codul 14. Implementare algoritm.*

## 3.6 Analiza economică

Pentru o analiză tehnică mult mai amănunțită, dar de acestă dată din punct de vedere economic, următoarele secvențe de cod, prezic tot prețul listărilor, dar cu o altă abordare a problemei.

Datele aparțin tot CSV-ului prezentat mai sus, dar de această dată, cu o nouă coloană "date" care conține perioada rezervărilor făcute pe airbnb pentru fecare listare în parte. Citirea datelor a fost realizată la fel ca în rubrica anterioră.



*Figura 31 Adăugarea coloanei date*

După relizarea unei sortări și setarea indexului 'date', a rezultat:

In [9]:#setarea indexului

df['date'] = pd.to\_datetime(df.date,format='%Y/%m/%d')

df.index = df['date']

#sortare

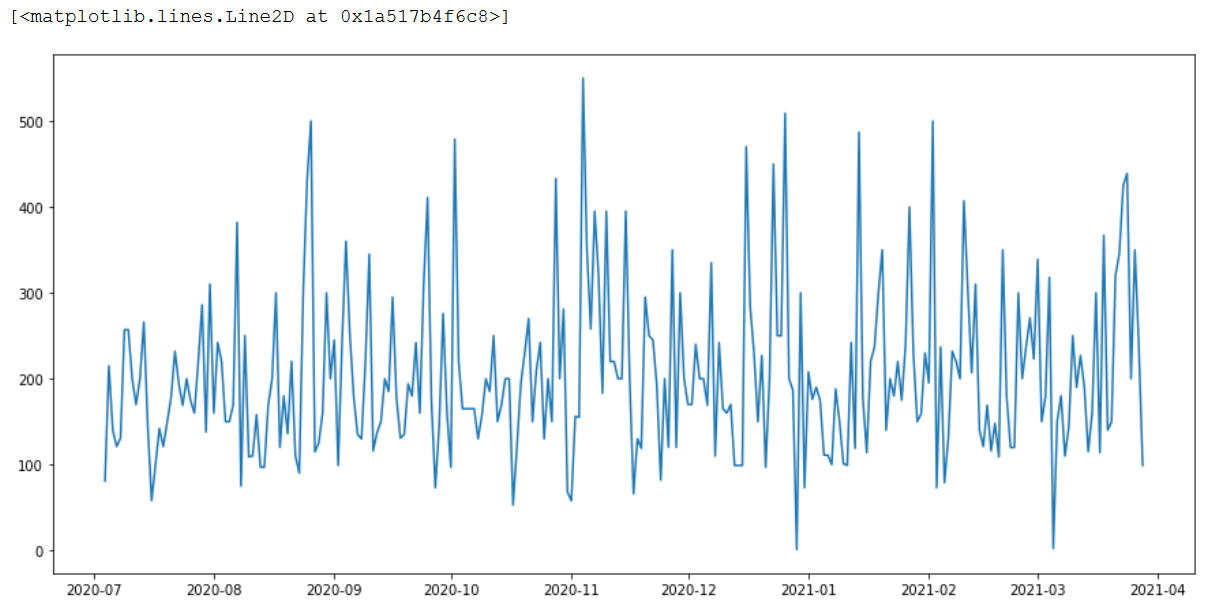
data = df.sort\_index(ascending=True, axis=0)

#vizualizare

plt.figure(figsize=(16,8))

plt.plot(df['price'], label='Istoric')

*Codul 15. Sortarea și setarea indexului 'date'*

Out [9]: 

*Figura 32 Vizualizarea prețurilor în funcție de dată*

Definirea problemei: prețul de închiriere prevăzut pentru fiecare zi va fi media unui set de valori observate anterior. Folosind tehnica medie în mișcare care presupune utilizarea celui mai recent set de valori corelat fiecărei predicții. Pentru fiecare etapă care va fi făcută, vor fi luate în considerare valorile prezice anterior, eliminând din setul de dare cea mai veche valoare observată.

In [10]:# importarea bibliotecilor

import pandas as pd

import numpy as np

# citirea datelor din CSV

df = pd.read\_csv('airbnb\_datesv3.csv', sep=';')

# vizualizarea primelor 5 linii din analiza datelor

print(df.head())

print('\n Forma datelor:')

print(df.shape)

*Codul 16. Implementarea problemei folosind tehnica medie în mișcare*

Out [10]: 

*Figura 33 Vizualizarea primelor 5 linii din analiza datelor*

In [11]: # setarea datelor

df['date'] = pd.to\_datetime(df.date,format='%Y-%m-%d')

df.index = df['date']

# crearea cadrului de date cu data și variabila țintă

data = df.sort\_index(ascending=True, axis=0)

new\_data = pd.DataFrame(index=range(0,len(df)),columns=['date', 'price'])

for i in range(0,len(data)):

new\_data['date'][i] = data['date'][i]

new\_data['price'][i] = data['price'][i]

# divizarea în funcțiile train și valid

train = new\_data[:268]

valid = new\_data[1:]

# vizualizarea setului rezultat de funcția train

print('\n Shape of training set:')

print(train.shape)

Out[11]: Vizualizare train:

(268, 2)

In[12]: # shapes of validation set

print('\n Vizualizarea funcției valid:')

print(valid.shape)

Out[12]: Vizualizarea setului de validare:

(267, 2)

In [13]: preds = []

for i in range(0,valid.shape[0]):

a = train['price'][len(train)-267+i:].sum() + sum(preds)

b = a/267

preds.append(b)

# verificarea rezultatului (RMSE)

rms=np.sqrt(np.mean(np.power((np.array(valid['price'])-preds),2)))

print('\n RMSE pentru setul de validare:')

print(rms)

Out[13]:RMSE pentru setul de validare:

96.0687124187353

In[14]: #vizualiare

valid['Predictions'] = 0

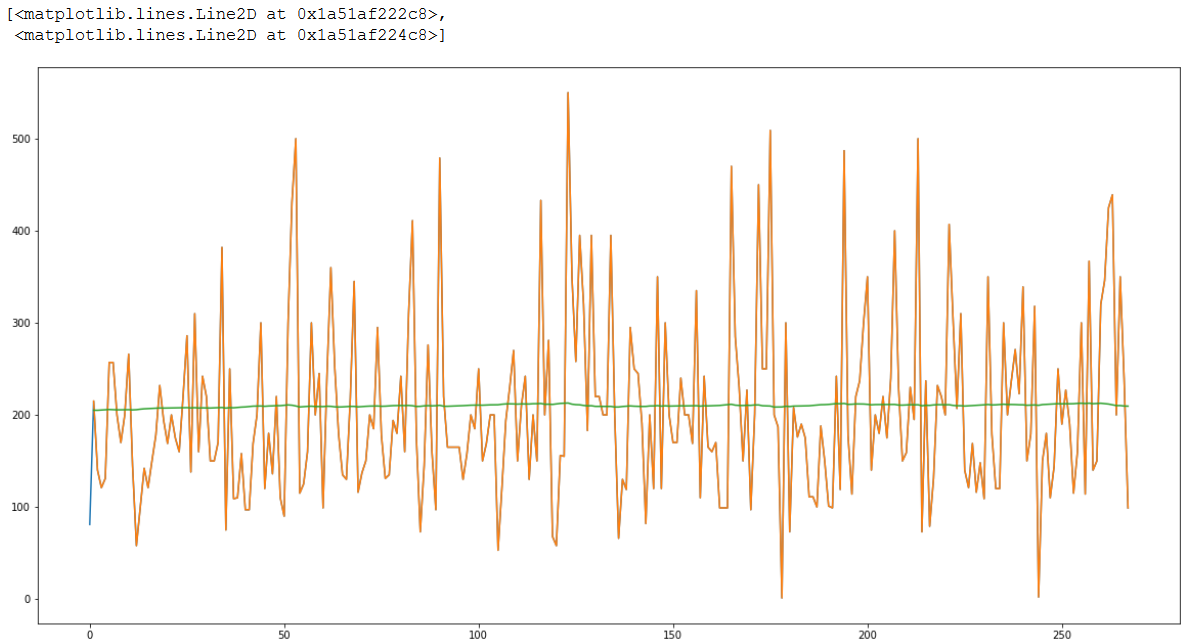
valid['Predictions'] = preds

plt.plot(train['price'])

plt.plot(valid[['price', 'Predictions']])

*Codul 17. Antrenarea și vizualizarea rețelelor*

Out [14]:



*Figura 34 Rezultatul învățării automate*

Culoarea galbenă reprezintă valid[('price', 'Predictions')], culoarea albastră reprezintă funcția train['price'], iar culoarea verde este 'Predictions'. Aceste culori se respectă pentru fiecare grafic care va urma, singura modificare va fi paramentru funcției train sau valid.

Interpretare: valoarea prezisă este 140, după cum se poate observa în grafic, valoarea funcției valid este 215, iar valoarea funcției train este 81.0. Există o fluctuație aproape constantă a valorilor.

### 3.6.1 Regresia Liniară

Modelul regresiei liniare returnează o ecuație care determină relația dintre variabilele independente (X1, X2,...,Xn) și variabilele dependente (θ1, θ2,.., θn).

Primul pas pentru implementare este reprezentat de sortarea datelor în ordine crescătoare, apoi, crearea unui nou set de date unde vor fi stocate toate noile caracteristici ale datelor, în acest mod, datele originale nu vor fi alterate.

In[15]: #setarea indexului datelor după date

df['date'] = pd.to\_datetime(df.date,format='%Y-%m-%d')

df.index = df['date']

#sortare

data = df.sort\_index(ascending=True, axis=0)

#crearea unui set de date separat

new\_data = pd.DataFrame(index=range(0,len(df)),columns=['date', 'price'])

for i in range(0,len(data)):

new\_data['date'][i] = data['date'][i]

new\_data['price'][i] = data['price'][i]

new\_data['date'] = 0

for i in range(0,len(new\_data)):

if (new\_data['date'][i] == 0 or new\_data['date'][i] == 4):

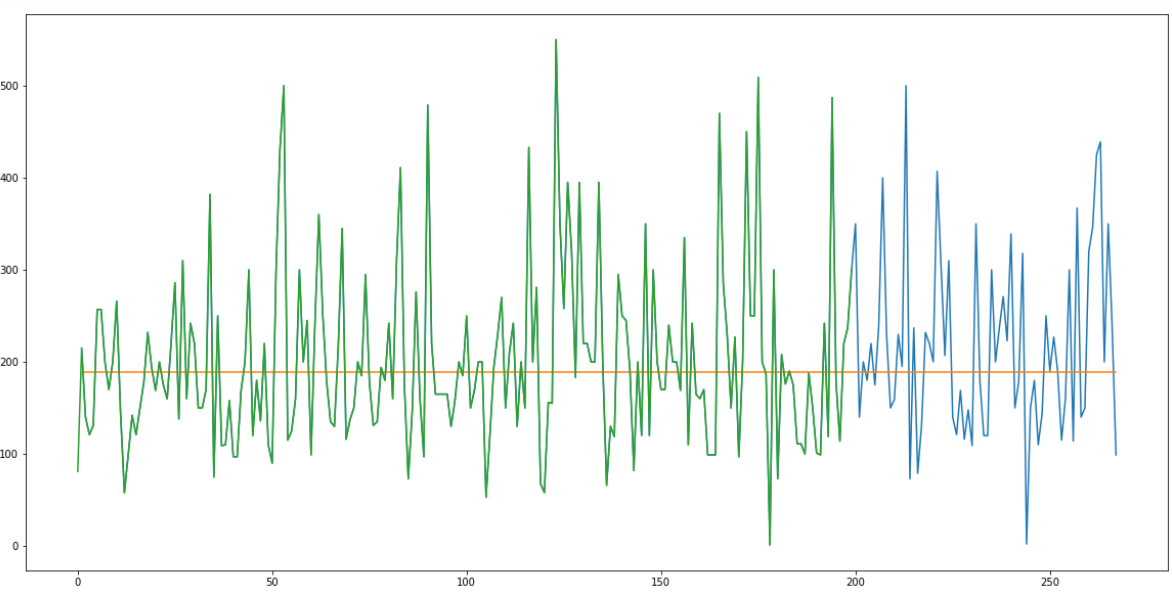
new\_data['date'][i] = 1

else:

new\_data['date'][i] = 0

*Codul 18. Crearea și stocarea datelor*

După implementarea regresiei liniare, se poate implementa și algoritmul KNN din nou, le fel ca modelul precizat în subcapitolul anterior. După implemenatare graficul arată astfel:



*Figura 35 Graficul rezultat după aplicarea regresiei liniare*

Iar prețul prezis de machine learning este 96.3784468247043, rezultat din formula RMSE. Prețul prezis cu ajutorul KNN este 97.57544765208857.

Dacă se reia pas cu pas lucrurile prezentate în acest subcapitol, dar fără a ține cont de coloana date, prețul prezis ar fi 115.9587529981832.

Interpretare: listările din perioada analizată arată fluctuații mari ale prețurilor de la o listare la alta. Conform Machine Learning prețul calculat pentru următoarea listare este aproximativ 116 .

Valorile tind să fie diferite datorită faptului că, rețeaua se inițializează cu valori random sau ia de fiecare dată un alt set de date. Indiferent de modul de inițializare, rezultatul va diferii cu câteva zecimale, ceea ce nu are un impact foarte mare asupra rezultatului.

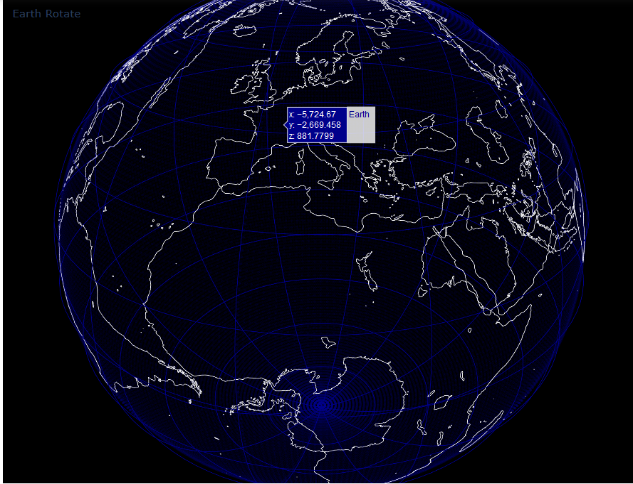
Concluzia oferită de grafice: Conform predicțiilor efectuate de Machine Learning fluctuațiile prețurilor în general vor rămâne la fel de mari. Fiind influențate de diferiți factori precum: sezonul, diverș factori economici, prețurile pieții, calitatea serviciilor oferite, confort și multe altele.

## 3.7 Date Geospațiale

Basemap este o bibliotecă pentru afișarea hărților 2D în Python [21]. Datele vor fi reprezentate progresiv, pentru o vizualizare generală, dar și în detaliu.

Primul pas în crearea unei hărți este reprezentat de localizarea orașului Brașov. Coordonatele sale sunt N 45.6580°, E 25.6012°.

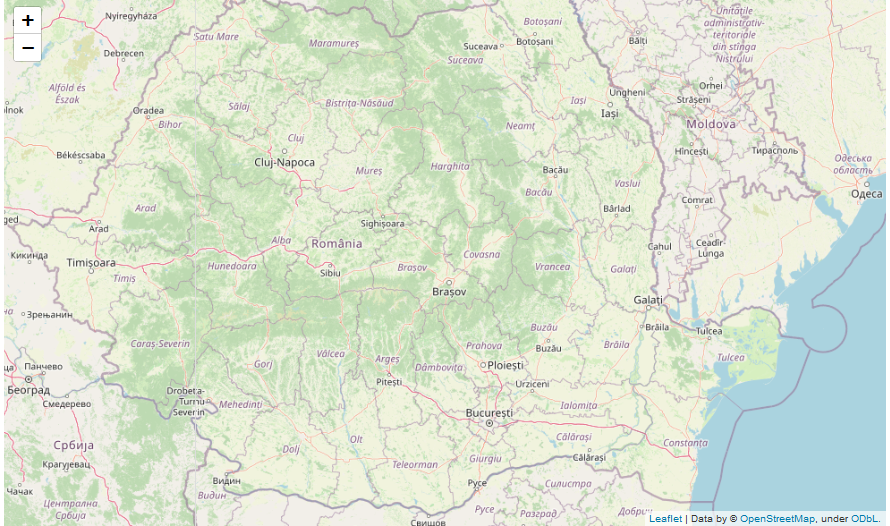
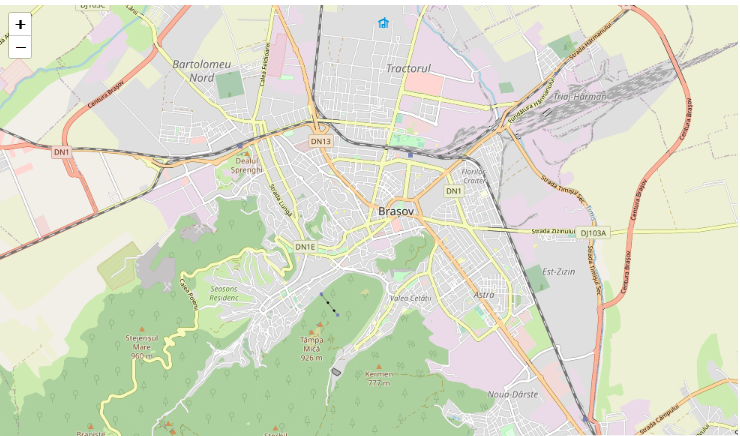
Introducerea în datele geospațiale a zonei noastre de interes se face prin afișarea întregului glob, o hartă a globului statică, dar și una dinamică care ne precizează coordonate zonei indicate de mouse. Aceasta fiind de ajutor în detectara coordonatelor x, y, z a unei zone, datele putând fi ulterior folosite în alte activități.



*Figura 36 Glob*

### 3.7.1 Conexiunea programului cu Google Maps

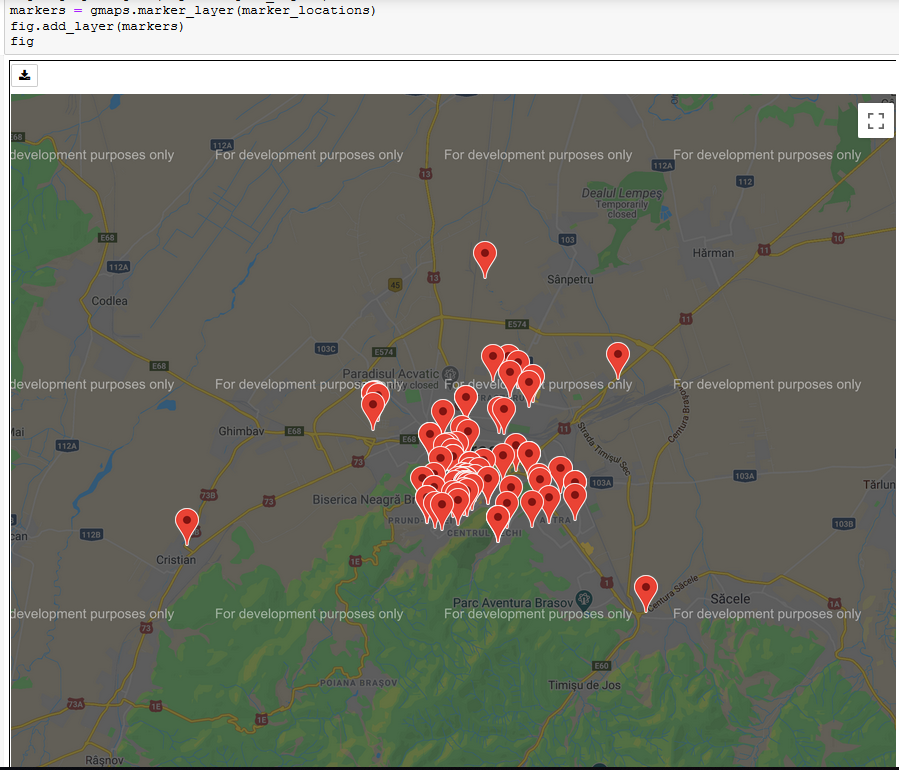
Următorul pas este reprezentat de afișarea hărții Romăniei, utilizând biblioteca folium, pentru o reprezentare cât mai exactă. Harta putând fi mărită sau micșorată, până se ajunge la harta orașului.

*Figura 37 Harta*

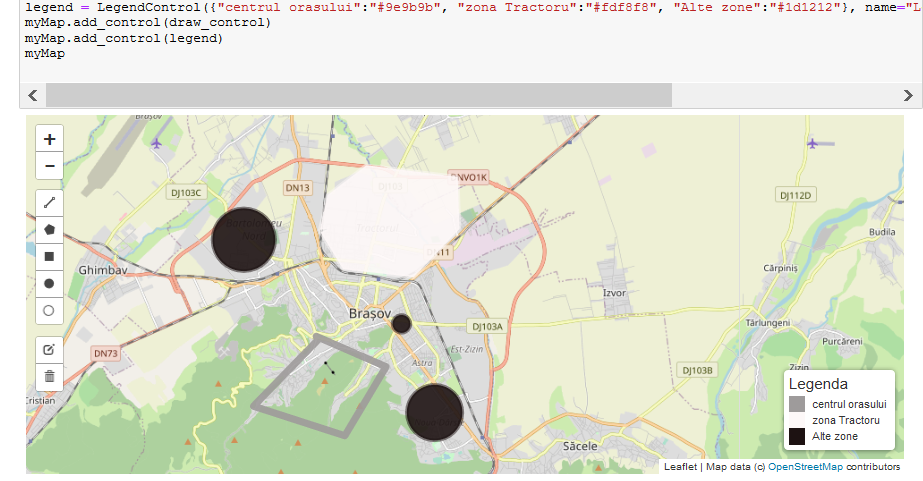
Pentru determinarea listărilor și marcarea locațiilor, este necesar un API de la Google, disponibil pe platforma acestora Google APIs Business Intelligence [22], necesar în conexiunea programului cu Google Maps. Acesta vine la pachet cu un certificat OAuth. OAuth permite aplicațiilor să obțină acces limitat la datele unui utilizator fără să ofere parola unui utilizator. Acesta decuplează autentificarea de la autorizare și acceptă cazuri de utilizare care se adresează diferitelor funcții ale dispozitivului [23].

Pentru a marca zonele de interes, este nevoie de coloanele "longitude" și "latitude" din setul de date. Fiecărei listări îi corespund niște coordonate, care acum, au fost introduse în cod și afișate ulterior în hartă, respectiv, zonele marcate cu roșu.



*Figura 38 Harta marcată*

În final, cu ajutorul unor figuri, aparțin bibliotecilor Map și DrawControl, se marcheză zonele cele mai căutate din Airbnb.



*Figura 39 Încadrarea zonelor*

Zonele cele mai căutate sunt cunoscute cu ajutorul unei funcții care caută un pattern ("potrivire") cu literele pe care le căutăm, rezultănd numărul de potriviri găsite. Din rezultate returnate, reiese că zone centru este cea mai căutată, urmată de zona Tractorul, Bartolomeu și restul zonelor din Brașov.

## 3.8 Analiza finală

Analiza finală a întregii lucrări a arătat că:

1. Din totalul de 268 de listări:

* Întrega cameră - 232 listări;
* Cameră privată - 21 listări;
* Cameră comună - 5 listări;
* Întregul apartament în complex rezidențial - 10 listări.

1. Oaspeții primiți raportat la 268 de listări:

* un oaspete - 4 listări;
* doi oaspeți - 91 listări;
* trei oaspeți - 26 listări;
* patru oaspeți - 1 listări;
* cinci oaspeți - 12 listări;
* șase oaspeți 6 - 21 listări;
* șapte oaspeți - 3 listări;
* opt oaspeți - 6 listări;
* nouă oaspeți - 0 listări;
* zece oaspeți - 2 listări;
* mai mult de 10 oaspeți - 3 listări;

1. Prețul minim pe noapte: 81 lei;
2. Prețul maxim pe noapte: 550 lei;
3. Media prețurilor: 204.60 lei;
4. Suma tuturor prețurilor: 54833.37 lei;
5. Prețurile în valoare de 140.00 au reprezentat 25% din total;
6. Prețurile în valoare de 187.50 au reprezentat 50%din total;
7. Prețurile în valoare de 242.00 au reprezentat 75% din total;
8. Prețul determinat de ML a fost în medie de 114 lei;
9. Valoarea totală a comentariilor a fost 6442;
10. Media comentariilor: 24.03;
11. Numărul maxim de comentarii la o listare a fost de 367:
12. Numărul minim de comentarii la o listare a fost 0;

* Numărul maxim de stele primit la o listare a fost de 5.0;

1. Numărul minim de stele primit la o listare a fost 0.0;
2. Numărul total de stele raportat la 268 de listări a fost de 867.03;
3. Întreg apartamenul în complex rezidențial reprezintă 1% din listări;
4. Întreaga cameră reprezintă 89% din listări;
5. Camera privată reprezintă 8% din listări;
6. Camera comună reprezintă 2% din listări;
7. Zonele cele mai căutate sunt Centrul Vechi, Tractoru, Bartolomeu, Astra, Zona Gării, resul cartierelor, în ordinea enumerată.

*Tabel 3. Rezultate analiză finală*

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | 1 | 2 | 3 | 4 | 4+ |
| Întreaga cameră | 80% | 0% | 9% | 0% | 0% |
| Cameră privată | 7.36% | 0.64% | 0% | 0% | 0% |
| Cameră comună | 1% | 0.80% | 0.10% | 0.05% | 0.05% |
| Apartament complex rezidențial | 7% | 3% | 0% | 0% | 0% |

*Figura 40. Tipul camerei*

Toate rezultatele sunt efectuate în jupyter notebook si python, pentru o redare cât mai exactă a numerelor.

## 3.9 Transformarea proiectelor jupyter în aplicații web

Obiectivul proiectele dezvotate în Jupyter Notebook este de a îmbunătății fluxurile de lucru ale cercetătorilor, oamenilor de știință sau ale începătorilor de a practica calcule științifce, de la faza de explorare până la faza de prezentare.

Deoarece, Jupyter Notebook nu reprezintă cel mai bun instrument de comunicare pentru toate categoriile de public, prin afișarea completă a codului, persoanele non-tehnice ar putea fi derutați de rezultate.

O altă provocare cu partajarea codului este reprezentat de securitate. Pentru a preveii toate aceste probleme a fost creat *Voilà*.

*Voilà* transformă proiectele Jupyter Notebook în aplicații web de sine stătătoare. Aantajele aceste extensii sunt:

* acceptă widget-uri interactive;
* nu permite executarea codului arbitrat;
* include un sistem de șabloane flexibil pentru a produce diverse aplicații.

A doua variantă este *Appmode* unde:

* se poate vizualiza orice notebook;
* toate celulele de cod sunt executate automat, pentru a prezenta o interfață de utilizator curată;
* poate fi deschis de mai multe ori fără interferențe, acest lucru fiind realizat prin crearea de copii temporare ale notebook-ului pentru fiecare vizualizare activă a prograului;
* fiecare pagină appmode are nucleul dedicat ipython;

# 4 Concluzii

Prin intermediul acestei lucrări am realizat o analiză atât teoretică cât și practică a principalelor aspecte legate de conceptul de analiză a datelor și folosirea conceptelor de Machine Learning.

Mai intâi am început prin a prezenta obiectivele acestei lucrări, resursele studiate în prealabil. De asemenea, am prezentat și modul în care va arăta proiectul.

Pe langă acestea am vorbit și despre programele în care am lucrat, pentru realizarea părții practice. De ce sunt utilizate aceste programe, dacă mai sunt actuale, toate aceste informații se regăsesc pe parcursul acestei lucrări.

Lucrarea a continuat cu implementarea obiectivelor stabilite anterior. De asemenea, au fost analizate mai multe modalități de implementare, dar și dificultățile acestor modalități. Pentru fiecare implementare a fost analizată avantajele, dar și dezavantajele acestora. Fiecare pas a fost parcurs și interpretat atât prin cuvinte, cât și prin grafice, numere și chiar cod.

Învățarea automată a fost inclusă în proiect pentru a prezice prețul standard al pieții, adaptat unui anumit caz.

Obiectivele acestui proiect au fost atinse. Proiectul a fost dezvoltat doar pentru orașul Brașov, negăsind un alt proiect, dezvoltat cu aceleași tehnologii, un proiect care să analizeze, vizualizeze și chiar să conțină un model de previzionare adaptat acestui oraș.

Proiectul poate fi extins și pentru alte orașe cu un număr mai mare de listări, precum Cluj sau București. Graficele rezultate din datele acestor orașe ar putea fi total diferite, dar și mult mai impresionante deoarece cu cât numărul de eșantioane este mai mare, cu atât și rezultatele sunt mai ample. Analiza se poate face ținând cont de mult mai multe criterii. Codul implementat poate fi reutilizat, fiind conceput astfel pentru a fi folosit în alte proiecte.

# 5 Bibliografie

1. [Czeslaw](https://sciprofiles.com/profile/492375) Adamiak, *Airbnb Offer in Spain - Spatial Analysis of the Pattern and Determinants of Its Distribution*.
2. Kristóf Gyódi. *Airbnb and the Hotel Industry in Warsaw An Exemple of the Sharing Economy?*
3. Alexander Bustamante, Laura Sebastia, Eva Onaindia (8 iunie 2019). *Can Tourist Attractions Boost Other Activities Around? A Data Analysis through Social**Networks*. Sensor Article
4. Airbnb, https://www.airbnb.com/help/home
5. Modul de funcționare, https://bmtoolbox.net/stories/airbnb/
6. Popularitatea limbajului Python în 2020, https://towardsdatascience.com/top-10-in-demand-programming-languages-to-learn-in-2020-4462eb7d8d3e
7. Framework, https://techterms.com/definition/framework
8. Anaconda, https://docs.anaconda.com/anaconda/
9. Jupyter, https://jupyter.org/
10. Etapele proiectului, https://docs.google.com/presentation/d/1fDpjlyMiOMJyuc7\_jMekcYLPP2XlSl1eWw9F7yE7byk/edit#slide=id.g7ee7f56aa2\_2\_5
11. XPath, https://www.w3schools.com/xml/xpath\_intro.asp
12. Wht is data analysis, https://www.guru99.com/what-is-data-analysis.htmlâ
13. Numpy, https://numpy.org/
14. Pandas, https://pandas.pydata.org/
15. Matplotlib, https://matplotlib.org/
16. Seaborn, https://seaborn.pydata.org/
17. Definiția Machine Learning, https://expertsystem.com/machine-learning-definition/
18. Regresia, https://www.investopedia.com/terms/r/regression.asp
19. Distanța euclidiană, https://www.sciencedirect.com/topics/mathematics/euclidean-distance
20. Eroarea medie pătrată, https://www.statisticshowto.com/probability-and-statistics/regression-analysis/rmse-root-mean-square-error/
21. Basemap, https://matplotlib.org/basemap/
22. Google Business Intelligence, https://console.developers.google.com/apis/dashboard?project=business-intelligence-8258e
23. OAuth, https://oauth.net/2/
24. Quattrone, G.; Prosepio, D.; Quercia, D.; Capra, L.; Musolesi, M. *Who Benefits from the " Sharing" Economy of Airbnb? In Proceedings of the 25th International Conference on World Wide Web - WWW* 16, Montréal,QC, Canada, 11–15 April 2016; ACM Press: New York, NY, USA, 2016; pp. 1385–1394
25. Hayashi, Chikio (1 January 1998). [*What is Data Science? Fundamental Concepts and a Heuristic Example.*](https://www.springer.com/book/9784431702085)In Hayashi, Chikio; Yajima, Keiji; Bock, Hans-Hermann; Ohsumi, Noboru; Tanaka, Yutaka; Baba, Yasumasa (eds.). Data Science, Classification, and Related Methods. Studies in Classification, Data Analysis, and Knowledge Organization. Springer Japan.
26. Clifton, Christopher (2010). [*Encyclopædia Britannica: Definition of Data Mining*](http://www.britannica.com/EBchecked/topic/1056150/data-mining)
27. Paul Barry. *Head First Python 2nd Edition* https://sourcecodefree.org/book-detail/head-first-python-2nd-edition.html
28. Shai Shalev-Shwartz and Shai Ben-David (2014). *Understanding Machine Learning From Theory to Algorithms* https://www.cse.huji.ac.il/~shais/UnderstandingMachineLearning/understanding-machine-learning-theory-algorithms.pdf
29. Daoud Clarke. *Seven Steps to Success Machine Learning in Practice*  https://daoudclarke.github.io/guide.pdf
30. Ryan Mitchell (2015). *Web Scraping with Python* https://www.amazon.com/Web-Scraping-Python-Collecting-Modern/dp/1491910291/
31. Anidh Chapagain (2016). *Hands-On Web Scraping with Python* https://www.amazon.com/Hands-Web-Scraping-Python-operations/dp/1789533392
32. Wes McKinney. *Python for Data Analysis (2017)* https://www.amazon.com/Python-Data-Analysis-Wrangling-IPython/dp/1491957662
33. Cathy O'Neil & Rachel Schutt (2013). *Doing Data Science* https://www.oreilly.com/library/view/doing-data-science/9781449363871/
34. Joel Lawhead (2013). *Learning Geospatial Analysis with Python*. https://www.amazon.com/Learning-Geospatial-Analysis-Python-Lawhead/dp/1783281138
35. Dimitrios Kouzis-Loukas (2016). *Learning Scrapy*. https://www.amazon.co.uk/dp/1784399787?linkCode=gs2&tag=uuid07-21
36. Dan Toomey (2016). *Learning Jupyter*  https://www.amazon.com/Learning-Jupyter-Dan-Toomey/dp/1785884875
37. Distanța euclidiană, https://www.sciencedirect.com/topics/mathematics/euclidean-distance

# Rezumat

Analiza spațială a zonelor turistice utilizând rețelele sociale reprezintă studiul și aprofundarea conceptelor și proceselor de extragere a datelor, de analiză și vizualizare a datelor. De asemenea, studiul conține și folosirea tehnicilor de Machine Learning făcând o mică introducere în domeniul de știință a datelor.

Lucrarea prezintă tehnologiile folosite, aspectele teoretice și modul de implementare a proiectului software, probleme întâmpinate, dar și rezolvarea sau găsirea unor soluții înlocuitoare pentru a putea aplica studiul și în format practic. Site-ul de interes în jurul căruia s-a concentrat lucrarea este Airbnb.

Toate informațiile rezultate la finalul proiectului sunt atribuite orașului Brașov, România. Pe site-ul Airbnb, acesta conține aproximativ 300 de listări diversificate. Prețurile, tipul camerei, părerile sunt doar câteva criterii aplicate acestei analize.

În final, graficele și numerele rezultate, vor arăta atât media prețurilor acestor listări, cât și cele mai zonele de interes ale orașului.

# Abstract

Spatial analysis of tourist areas using social networks is the study and deepening of the concepts and processes of data extraction, data analysis and visualization. The study also contains the use of Machine Learning techniques making a small introduction to the field of data science.

The project presents the technologies used, the theoretical aspects and the implementation of the software project, problems encountered, but also solving or finding alternative solutions to apply the study in a practical format. The site of interest around which the work focused is Airbnb.

All information resulting from the end of the project is attributed to the city of Brașov, Romania. On the Airbnb site it contains about 300 diverse listings. Prices, room type, reviews are just a few criteria applied to this analysis.

Finally, the resulting graphs and numbers will show both the average prices of these listings and the most areas of interest in the city.

|  |
| --- |
| DECLARAȚIE PRIVIND ORIGINALITATEA  PROIECTULUI DE DIPLOMĂ / DISERTAȚIE |
| UNIVERSITATEA TRANSILVANIA DIN BRAŞOV  FACULTATEA DE INGINERIE ELECTRICĂ ȘI ȘTIINȚA CALCULATOARELOR  PROGRAMUL DE STUDII TEHNLOGIA INFORMAȚIEI |
| NUMELE ŞI PRENUMELE ANDRIȘAN ARIANA  PROMOȚIA 2016/2020  SESIUNEA DE ABSOLVIRE DIPLOMĂ VARA 2020 |
| DENUMIREA LUCRĂRII / PROIECTULUI/ DISERTAȚIEI  ANALIZA SPAȚIALĂ A ZONELOR TURISTICE UTILIZÂND REȚELE SOCIALE  CADRUL DIDACTIC ÎNDRUMĂTOR CONF. DR. ING. ROBERT DEMETER |
| Declar pe propria răspundere că lucrarea de față este rezultatul muncii proprii, pe baza cercetărilor proprii şi pe baza informațiilor obținute din surse care au fost citate şi indicate conform normelor etice, în textul lucrării/proiectului, în note şi în bibliografie. Declar că nu s-a folosit în mod tacit sau ilegal munca altora şi că nici o parte din teză/proiect nu încalcă drepturile de proprietate intelectuală ale altcuiva, persoană fizică sau juridică.  Declar că lucrarea/ proiectul nu a mai fost prezentat(ă) sub această formă vreunei instituții de învățământ superior în vederea obținerii unui grad sau titlu ştiințific ori didactic.  În cazul constatării ulterioare a unor declarații false, voi suporta rigorile legii. |
| Data: 11.06.2020    Absolvent (*nume, prenume)* ANDRIȘAN ARIANA    *semnătura*  F07-PS 7.6-01/ed.2,rev.2 |
|  |
|  |
|  |