UNIVERSITATEA „ALEXANDRU IOAN CUZA” IAŞI

FACULTATEA DE INFORMATICĂ

LUCRARE DE DISERTAȚIE

Evoluția cazurilor de COVID-19 în România: analiză, forecast, clusterizare

propusă de

Dorin – Andrei - Beniamin Miron

Sesiunea: *iulie, 2021*

Coordonator ştiinţific

Conf. Dr. Liviu Ciortuz

UNIVERSITATEA ALEXANDRU IOAN CUZA IAŞI

FACULTATEA DE INFORMATICĂ

Evoluția cazurilor de COVID-19 în România: analiză, forecast, clusterizare

Dorin – Andrei - Beniamin Miron

Sesiunea: *iulie, 2021*

Coordonator ştiinţific

Conf. Dr. Liviu Ciortuz

ABSTRACT

În decembrie 2019, Autoritatea Chineză pentru Sănătate a alertat **O**rganizația **M**ondială a **S**ănătății (OMS) cu privire la mai multe cazuri de pneumonie de etiologie necunoscută, cu un potențial ridicat de transmisibilitate între oameni. La doar cateva zile OMS a declarat această infecție virală drept o urgență de sănătate publică de interes internațional. La17 luni după prima descriere a virusului, există mai mult de 150 de milioane de subiecți la nivel global cu infecție SARS-CoV-2 confirmată pe baza testului molecular, iar peste 3 milioane de decese au fost atribuite COVID-19. Această pandemie a reprezentat o mare amenințare pentru sănătatea fizică și mentală a omului și a avut un impact dramatic asupra vieții de zi cu zi cu implicații psihosociale la scară globală.

În lucrarea de față am testat diverși algoritmi de forecasting și de clusterizare cu dorința de a vedea daca aceștia pot constitui niște unelte folositoare pentru factorii decizionali în gestionarea situațiilor sociale complexe pe care acest virus le-a generat și în adoptarea de noi politici pentru a reveni la normal. Pentru partea de forecasting algoritmul Exponential Smoothing a obținut rezultate foarte încurajatoare, iar la partea de clusterizare s-au obținut un număr de 4 clustere cu județele din România.

Cuprins

ABSTRACT *3*

Introducere *7*

1. Setul de date *8*

2. Analiza și explorarea setului de date *9*

2.1 Corectarea anomaliilor din date **9**

2.2 Descompunerea seriei de timp în trend și sezonalitate **10**

2.3 Calcularea autocorelației și crearea corelogramei **10**

3. Forecasting *12*

3.1 Forecasting pe serii de timp **13**

3.2 Cercetari în domeniu **14**

3.2.1 COVID-19: A Comparison of Time Series Methods to Forecast Percentage of Active Cases per Population 14

3.2.2 Forecasting the novel coronavirus COVID-19 15

3.2.3 Short-term forecasting COVID-19 cumulative confirmed cases: Perspectives for Brazil 15

3.3 Algoritmi utilizați **16**

3.3.1 Exponential Smoothing 16

3.3.1.1 Simple Exponential Smoothing 16

3.3.1.2 Double exponential smoothing (Holt's method) 18

3.3.1.3 Triple Exponential Smoothing (Holt-Winters' seasonal method) 18

3.3.2. ARIMA 20

3.3.2.1 Modelul AutoRegressive 20

3.3.2.2 Modelul Moving Average 20

3.2.2.4 Modelul ARIMA 21

3.4 Experimente și rezultate **22**

3.4.1 Perioadele selectate 22

3.4.2 Județele selectate 23

3.4.3 Procesul de evaluare 23

3.4.4 Rezultate 23

4. Clusterizare *25*

4.1 Clusterizarea seriilor de timp **25**

Definirea problemei **26**

4.2 Cercetari in domeniu **27**

4.2.1 "Clustering stationary and non-stationary time series based on autocorrelation distance of hierarchical and k-means algorithms" 27

4.2.2 „Analysis of similarity measures in times series clustering for the discovery of building energy patterns" 27

4.3 Algoritmi utilizați **28**

4.3.1 K-Means 28

4.3.2 Fuzzy C-Means 28

4.3.3 K-medoids 29

4.3.4 Mixturi Gausiene 30

4.3.5 Self-Organizing map (SOM) 30

4.3 Experimente și rezultate **31**

4.3.1 Metricile folosite 31

4.3.2 Numărul de clustere 31

4.3.3 Rezultate 31

Anexa A *33*

Anexa B *34*

Anexa C ***35***
Anexa D ***36***
Anexa E ***40***
Bibliografie si Webografie ***42***

Introducere

În decembrie 2019, Autoritatea Chineză pentru Sănătate a alertat **O**rganizația **M**ondială a **S**ănătății (OMS) cu privire la mai multe cazuri de pneumonie de etiologie necunoscută, cu un potențial ridicat de transmisibilitate între oameni, în orașul Wuhan din provincia Hubei din centrul Chinei. Centrul Chinez pentru Controlul și Prevenirea Bolilor, împreună cu alte instituții conexe, a identificat rapid agentul patogen ca un nou tip de coronavirus. Pentru a se asigura că informațiile sunt distribuite rapid în întreaga lume, prima secvență virală a fost depusă în GenBank și făcută publică la 26 decembrie 2019 (LR757995, LR757998).

Organizația **M**ondială a **S**ănătății (OMS) a emis alerte la 30 decembrie 2019 și la 30 ianuarie 2020 și a declarat această infecție virală drept o urgență de sănătate publică de interes internațional. La 11 februarie 2020, Comitetul Internațional de Taxonomie a Virușilor a numit acest virus **S**indromul **A**cut **R**espirator **S**ever SARS-CoV-2 pe baza relației filogenetice a coronavirusului care a cauzat focarul SARS în 2003. În aceeași zi, OMS a anunțat COVID-19 ca denumire a noii boli cauzate de acest virus urmând liniile directoare ale Organizației Mondiale pentru Sănătatea Animalelor și Organizației pentru Alimentație și Agricultură a Națiunilor Unite.

Începând cu 1 mai 2021, la 16-17 luni după prima descriere a virusului, există mai mult de 150 de milioane de subiecți la nivel global (din mai mult de 210 țări) cu infecție SARS-CoV-2 confirmată pe baza testului molecular, iar peste 3 milioane de decese au fost atribuite COVID-19. Această pandemie a reprezentat o mare amenințare pentru sănătatea fizică și mentală a omului și a avut un impact dramatic asupra vieții de zi cu zi cu implicații psihosociale la scară globală.

Ca răspuns, multe țări au pus în aplicare măsuri precum autoizolarea și distanțarea socială pentru a preveni răspândirea în continuare a virusului, și pentru a aplatiza curba epidemiei, ceea ce s-a dovedit crucial în menținerea serviciilor de sănătate pentru pacienții care au cea mai mare nevoie de îngrijire fie pentru COVID -19 sau pentru alte afecțiuni grave.

Capacitatea de a identifica ritmul la care se răspândește boala este crucială în lupta împotriva pandemiei. Abilitatea de a recunoaște și de a anticipa nivelul de răspândire în orice moment dat are potențialul de a ajuta guvernele în planificarea sănătății publice și elaborarea politicilor pentru a aborda consecințele pandemiei.

De asemenea, posibilitatea de a grupa diverse regiuni pe baza anumitor tipare de transmitere poate ajuta factorul decizional în aprobarea sau respingerea anumitor politici de carantinare sau de relaxare în funcție de impactul pe care măsurile similare le-au avut asupra regiunilor cu același tipar.

1. Setul de date

Setul de date folosit pentru analiza din această lucrare este creat de către cei de la Google și pus la dispoziția oricărei persoane care vrea sa le acceseze, să facă cercetări sau analize asupra evoluției situației generate de COVID-19 . Această inițiativă face parte dintr-un efort de a ajuta la combaterea pandemiei și cuprinde îmbinarea și agregarea mai multor date oferite de către Wikidata, Centrul European pentru Prevenirea și Controlul Bolilor, Google, Global Health Data de la Banca Mondială, OpenStreetMap, Eurostat și alte surse.

Acest proiect de colectare și sincronizare a mai multor seturi de date este coordonat de Oscar Wahltinez și poate fi găsit sub titlul „COVID-19 Open-Data: curating a fine-grained, global-scale data repository for SARS-CoV-2”. Datele agregate cuprind mai multe aspecte printre care indici epidemiologici, economici, demografici, geografici, datele referitoare la declararea stărilor de urgență, numărul de pacienți spitalizați, mobilitatea populației sau starea vremii.

În lucrarea de față am analizat evoluția cazurilor de covid-19 în Romania și pentru acest lucru, ne-am concentrat, în mod deosebit, asupra cazurilor de persoane noi infectate.

2. Analiza și explorarea setului de date

În statistică, analiza exploratorie a datelor reprezintă o primă etapă de familiarizare cu un set de date cu scopul de a sintetiza caracteristicile principale ale acestuia. Acest proces este critic deoarece în cadrul său sunt făcute „primele investigații pe setul de date, sunt descoperite anumite tipare, se detectează anomalii în date, se testează ipoteze și se verifică presupuneri prin tot felul de indicatori statistici și reprezentări grafice."[1]

2.1 Corectarea anomaliilor din date

Atributul cel mai utilizat în această lucrare este numărul zilnic de noi cazuri confirmate cu covid-19 de aceea în cea mai mare parte din această analiză inițială ne vom concentra asupra acestor valori. Unul din principalele lucruri observate sunt valorile negative, adică numărul persoanelor noi infectate și confirmate este negativ. Acest lucru înseamnă că există erori de raportare a acestor valori. În mod special, la nivelul României avem 2 perioade suspecte. În primul rând avem 18-19 August 2020 unde descoperim un tipar, cel mai probabil, de eroare umană, în care în prima zi a fost introdusă o valoare negativă, iar a doua zi o valoare pozitivă mult mai mare decât media ultimilor zile. Acest lucru ne face să credem că este vorba de o corectare a valorilor negative, iar în toate cazurile în care am întâlnit un asemenea comportament am procedat prin a înlocui cele 2 valori cu media lor aritmetică.

Am identificat 242 de zile pe județ în care au fost raportate un număr negativ de cazuri. Dintre acestea, în 10 situații numărul cazurilor negative depășește 180 de persoane.

Locație	Data	Număr persoane noi confirmate pozitiv cu covid-19
Prahova	2020-12-01	-9864
Romania	2020-10-23	-5028
Romania	2020-08-18	-1747
Bucuresti	2020-10-23	-685
Timisoara	2020-10-23	-333
Cluj	2020-10-23	-273
Iasi	2020-10-23	-268
Brasov	2020-10-23	-233
Bucuresti	2020-08-18	-199
Prahova	2020-10-23	-189

[1] Prasad Patil: „What is Exploratory Data Analysis?" https://towardsdatascience.com/exploratory-data-analysis-8fc1cb20fd15

Dintre valorile negative rămase, 82 se află în intervalul [-180, -10], restul având valori intre (-10, 0]. În Anexa A se pot observa histogramele acestor valori în figura 1 respectiv figura 2.

În figurile 3 și 4 se poate observa seria de timp corespunzătoare cazurilor noi de infectări confirmate așa cum au fost raportate la nivelul întregii României, înainte de orice procesare, adică cu tot cu valorile negative și după ce au fost eliminate aceste valori cu metoda descrisă mai sus.

Un caz particular este cel al județului Suceava unde în prima zi de raportare oficială au fost introduse 53 de cazuri, pe data de 14 mai 2020. Apoi s-a încercat corectarea acestui număr prin raportarea unui număr de cazuri fie negativ fie pozitiv doar foarte aproape de 0 pe o perioada de 4 zile consecutive. În acest caz, am rămas cu acele valori negative deoarece corectarea lor presupune modificarea a cel puțin 10 zile consecutive din seria de timp.

2.2 Descompunerea seriei de timp în trend și sezonalitate

Seriile de timp pot avea structuri mai complexe și pentru a fi analizate mai atent se pot descompune în subcomponente, fiecare astfel de componentă reprezentând un anumit tipar. Cel mai comun tip de descompunere a seriilor de timp este cel în care rezulta 3 componente: trendul, sezonalitatea și o componenta neregulată numită cel mai des reziduu.

În lucrarea de față am ales 5 județe din Romania asupra cărora am insistat mai mult cu această analiză. Acestea sunt Iași, Suceava, Cluj, Timiș și Brașov. În urma analizei descompunerii acestor serii de timp putem observa câteva aspecte importante. În primul rând vedem că există o sezonalitate de 7 zile dar aceasta este generată nu de perioada de infectare, care este multiplu de 7 ci de zilele din saptamană și de capacitatea de a testa persoanele suspecte. Astfel pentru zilele din sfârșitul de săptămână se analizau mai puține teste și implicit numărul noi de cazuri este mult mai mic. Spre exemplu, avem zilele de 12, 18, 19 și 20 ianuarie în care sunt raportate 0 cazuri în toate județele analizate, iar după fiecare zi s-a raportat câte un număr considerabil mai mare decât media celor mai apropiate 3 zile. Un alt lucru observat este faptul ca amplitudinea sezonalității se schimbă în fiecare val și uneori chiar și structura acestei sezonalități.

Pentru a vizualiza și analiza mai în detaliu seriile de timp originale, trendul și sezonalitatea pentru fiecare județ se pot consulta figurile 5-9 din anexa B de la final.

2.3 Calcularea autocorelației și crearea corelogramei

Autocorelația, cunoscută și sub numele de corelația serială, este corelația unei observații cu o alta obținută aplicând o funcție de întârziere. În mod informal putem spune că autocorelația este gradul de influență dintre două observații ce se află la o distanță de un anumit număr de

pași. Cu alte cuvinte, în același mod în care corelația măsoară relația liniară dintre două variabile, autcorelația măsoară relația liniară dintre diferite valori ale aceleași serii de timp.

Acest pas din analiza unei serii de timp are scopul, în primul rând, de a detecta componenta nealeatoare din date, iar apoi, cu ajutorul ei se pot identifica diverse modele, metode, abordări și algoritmi care se potrivesc pe tiparul seriei de timp identificat. Dându-se seria de timp $Y_1, Y_2, \dots, Y_N$, funcția de autocorelație de pas k are următoarea formulă:

$$r_k = \frac{\sum_{i=1}^{N-k}(Y_i - \bar{Y})(Y_{i+k} - \bar{Y})}{\sum_{i=1}^{N}(Y_i - \bar{Y})^2},$$

Unde $\bar{Y}$ reprezintă valoarea medie a tuturor observațiilor.

Așa cum am observat și când am analizat sezonalitatea seriilor de timp, există o ciclicitate la nivel de saptamană, adică din 7 în 7 zile. Urmărind rezultatele obținute vedem o autocorelație foarte puternică, de peste 0.8 în Timișoara și Iași, la polul opus aflându-se județul Suceava cu o autocorelație de doar 0,66. Pentru o analiza mai aprofundată, se pot consulta figurile 10-14 din anexa C.

3. Forecasting

Forecasting se referă la abilitatea de a face predicții asupra ceea ce se va întampla în viitor luând în considerare evenimentele din trecut și cele din prezent. Aceste predicții pot fi folosite ca o unealtă în ajutarea luării deciziilor care au un impact asupra viitorului în condiții de incertitudine, analizând datele istorice si trendurile.

Această abilitate a fascinat oamenii de-a lungul secolelor, uneori fiind considerată un semn al inspirației divine, alteori fiind văzută ca o activitate criminală. Profetul evreu Isaia scria în anul 700 î.Hr: „Spuneţi-ne ce se va întâmpla mai târziu, ca să ştim că sunteţi dumnezei". Cei ce aveau astfel de ocupații au avut o perioadă foarte dificilă în perioada lui Constantin care a emis un decret în 357 d.Hr. interzicând fiecărui cetățean roman „să consulte un prezicător, un matematician sau un prognostic [...]. Fie ca curiozitatea de a prezice viitorul să fie redusă pentru totdeauna". O interdicție similară a predicțiilor a avut loc în Anglia în 1736, când a devenit o fraudă taxarea banilor pentru predicții, iar pedeapsa era de 3 luni de închisoare cu muncă grea în folosul comunității. [2]

În contextul pandemiei de COVID-19 capacitatea de a identifica ritmul la care se răspândește boala este crucială în lupta împotriva pandemiei. Abilitatea de a recunoaște și de a anticipa nivelul de răspândire în orice moment dat are potențialul de a ajuta guvernele în planificarea sănătății publice și în elaborarea politicilor pentru a aborda consecințele pandemiei. De aceea este foarte importantă crearea unor modele capabile să facă predicții asupra unor posibile scenarii. Acestea pot juca un rol extrem de important în gestionarea și controlarea pandemiei.

Totuși sarcina de a crea modele de forecasting care să aibă acuratețe mare nu este tot timpul ușoară. Predictibilitatea unor evenimente sau a unor cantități depinde de mai mulți factori printre care:

1. Cât de bine înțelegem factorii care contribuie la generarea acelui eveniment sau a acelor cantități;
2. Cât de multe date disponibile există;
3. Dacă predicțiile pot afecta evenimentele reale asupra cărora se fac.

În cazul predicțiilor cazurilor noi de persoane infectate cu noul coronavirus, fiecare dintre cei trei factori sunt prezenți. Deoarece acest tip de virus nu s-a transmis la oameni până acum, nu a captat atenția biologilor și nu a fost foarte studiat. Acest lucru face să nu știm rata de incidență a lui, materialele pe care acesta supraviețuiește, cât timp poate supraviețui înafara unui organism viu și alte lucruri asemănătoare. În privința celui de-al doilea factor putem spune

[2] Hyndman, R.J., & Athanasopoulos, G. (2018) Forecasting: principles and practice, 2nd edition, OTexts: Melbourne, Australia. OTexts.com/fpp2

ca s-au făcut eforturi mari pentru a se publica datele imediat ce acestea erau centralizate. Cu toate acestea, există situații în care anumite date au fost raportate în mod eronat sau situații în care s-a încercat o corectare din mers a greșelilor umane de raportare. Ultimul factor cred că este cel mai important în cazul nostru deoarece aceste predicții sunt folosite de guvernele fiecărei țări pentru a genera politici menite să minimizeze numărul de infectări astfel încât spitalele țărilor respective să fie capabile să gestioneze pacienții aflați în stare gravă.

3.1 Forecasting pe serii de timp

În matematică, o serie de timp este o serie de puncte de date indexate în ordine temporală. Cel mai frecvent, o serie de timp este o secvență luată în momente succesive la distanțe egale în timp. Seriile cronologice sunt utilizate în statistică, în procesarea semnalului, în recunoașterea modelelor, în econometrie, în finanțe matematice, în prognoza meteo, în predicție de cutremure, în electroencefalografie, în inginerie de control, în astronomie, în inginerie a comunicațiilor și, în mare parte, în orice domeniu al științei aplicate și inginerie care implică măsurători temporale.

Analiza seriilor temporale cuprinde metode pentru analiza datelor din seriile temporale pentru a extrage statistici semnificative și alte caracteristici ale datelor. Forecasting-ul seriilor cronologice presupune utilizarea unui model pentru a prezice valorile viitoare pe baza valorilor observate anterior.

Datele seriilor de timp au o ordonare temporală naturală. Acest lucru face ca analiza seriilor temporale să fie distinctă de studiile transversale, în care nu există o ordonare naturală a observațiilor. Un exemplu de problemă din domeniul studiilor transversale a datelor este problema de regresie. Un caz particular poate fi explicarea salariilor oamenilor prin referire la nivelurile lor de educație, situație în care datele indivizilor ar putea fi introduse în orice ordine.

Un model stocastic pentru o serie de timp va reflecta în general faptul că observațiile apropiate în timp vor fi mai strâns legate decât observațiile aflate la o distanță mai mare. Să presupunem că avem o serie de timp observată x_1, x_2, ..x_N și dorim să facem o predicție, în viitor pentru valoarea x_{N+h}. Numarul *h* se numește orizontul predicției, iar valoarea x_{N+h} corespunde valorii prezise cu *h* pași în viitor față de momentul prezent. O metodă de forecasting este o procedură de a calcula predicții din valorile prezente si din cele trecute. Aceste metode cuprind un spectru variat, de la algoritmi foarte simpli bazați pe reguli care nu depind de nici un model probabilistic, până la algoritmi capabili de a identifica un anumit tipar în datele colectate și de a prezice valorile optime în raport cu modelul identificat.[3]

[3] Chris Chatfield: „Time-series forecasting", p. 3

3.2 Cercetari în domeniu

3.2.1 COVID-19: A Comparison of Time Series Methods to Forecast Percentage of Active Cases per Population[4]

Vasilis Papastefanopoulos, Pantelis Linardatos și Sotiris Kotsiantis au publicat pe 3 Iunie 2020 un articol în care au folosit diverse metode statistice și algoritmi de învățare automată pentru a propune estimări privitoare la procentul de cazuri active raportat la populația totală pentru 10 țări: Statele Unite ale Americii, Marea Britanie, Italia, Spania, Rusia, Franța, Turcia, Germania, Iran și Brazilia. Ei au comparat 6 abordări: ARIMA, Holt-Winters (cunoscut și sub numele de Exponential Smoothing), TBAT, Prophet (algoritm creat și făcut disponibil tuturor de Facebook), DeepAr și N-Beats. Metrica de evaluare folosită este rădăcina medie a erorilor la pătrat (RMSE). Perioada de forecast a fost de 7 zile.

Tara	Cel mai bun model	Eroare (RMSE)	Populatia tarii	Eroarea in unitati absolute	Numar cumulat de cazuri	MAPE
US	ARIMA	0.007421	330610570	2453461	1241603	197%
Spania	TBAT	0.029295	46751175	1369575	233889	585%
Italia	ARIMA	0.005628	60479424	340378	211926	160%
Marea Britanie	TBAT	0.004310	67814098	292278	37903	771%
Franta	NBEATS	0.004220	65244628	275332	122880	224%
Germania	TBAT	0.003389	83730223	283761	166152	170%
Rusia	ARIMA	0.001536	145922010	224136	145267	154%
Turcia	Holt-Winter	0.000887	84153250	74643	347239	21%
Brazilia	DeepAR	0.002836	212253150	601949	108266	556%
Iran	TBAT	0.000425	83771587	35602	98647	36%

Primele 4 coloane din tabelul de mai sus au fost preluate din rezultatele obținute de autorii studiului, iar ultimele 3 coloane au fost adăugate pentru a putea compara rezultatele obținute în această lucrare cu cele obținute de autorii studiului.

Concluzia lor este ca nu exista o metodă care să se potrivească în toate situațiile atunci când vine vorba de a prezice cazurile active din diferite tări, totuși ARIMA si TBAT au avut rezultate mai bune în șapte dintre cele zece țări, iar în celelalte două țări au avut al doilea cel mai bun rezultat.

[4] Papastefanopoulos Vasilis; Linardatos Pantelis; Kotsiantis Sotiris : „COVID-19: A Comparison of Time Series Methods to Forecast Percentage of Active Cases per Population" https://www.mdpi.com/2076-3417/10/11/3880

3.2.2 Forecasting the novel coronavirus COVID-19[5]

Fotios Petropoulos și Spyros Makridakis au publicat un articol, pe 31 martie 2020, în care au încercat să ofere un răspuns la întrebarea „care va fi impactul global a noului coronavirus (covid-19)?". Scopul lor a fost sa ofere niște metode obiective de predicție a evoluției cazurilor de covid-19 la nivel global. Au analizat 3 variabile și anume: cazuri confirmate, decesele cauzate de infectarea cu COVID-19 și numărul pacienților care s-au recuperat după infectare. Obiectivul propus de ei a fost sa facă predicții pentru cazurile cumulate de persoane infectate.

Algoritmii încercați fac parte din familia de metode bazate pe Exponential Smoothing. Experimentele făcute de autorii studiului au fost împărțite în 5 runde. În primele 4 au putut să antreneze și să evalueze modelul folosit calculând erorile obținute, iar în runda a 5-a au făcut niște aproximări cu privire la ceea ce se va întâmpla în viitor.

În prima rundă de forecast, pentru perioada dintre 01/02/2020 – 10/02/2020, au obținut o eroare 388%, pentru a doua perioadă de forecast între 11/02/2020-20/02/2020 eroarea obținută a fost de doar 7.7%. Pentru perioada a 3-a, adică următoarele 10 zile, eroarea procentuală a continuat să scadă pana la 6.2%, iar în ultima perioadă a crescut la 12.1%.

3.2.3 Short-term forecasting COVID-19 cumulative confirmed cases: Perspectives for Brazil[6]

În acest studiu, obiectivul autorilor este sa exploreze și să compare capacitatea predictivă a modelelor statistice și a celor de învățare automată ce rezolva probleme de regresie. Problema de forecasting se concentrează asupra cazurilor cumulative de covid-19 din Brazilia și s-a încercat optimizarea predicțiilor pentru o zi în viitor, pentru 3 zile în viitor și pentru 6 zile în viitor. Metodele folosite sunt **A**uto**R**egresive **I**ntegrated **M**oving **A**verage (ARIMA), Cubist Regression (CUBIST), **R**andom **F**orest (RF), Ridge Regression (Ridge), **S**uport **V**ector **R**egression (SVR) și o metodă de asamblare a mai multor modele. Pentru abordarea cu asamblarea mai multor modele s-au folosit drept algoritmi de baza CUBIST, RF, RIDGE și SVR, iar la meta-model, care să interpreteze aceste modele s-a folosit un model gausian.

Concluziile autorilor sunt ca ARIMA este un model bun pentru a face forecast pentru o zi în viitor, în timp ce modelele CUBIST si RIDGE merită atenție pentru forecast pentru două respectiv trei zile în viitor.

[5] Fotios Petropoulos; Spyros Makridakis: „Forecasting the novel coronavirus COVID-19" - https://journals.plos.org/plosone/article?id=10.1371/journal.pone.0231236
6 Matheus Henrique Dal Molin Ribeiro; Ramon Gomes da Silva et. all: „Short-term forecasting COVID-19 cumulative confirmed cases: Perspectives for Brazil"

3.3 Algoritmi utilizați

3.3.1 Exponential Smoothing

Această metodă a fost propusă la sfârșitul anilor 1950 (Brown, 1959; Holt, 1957; Winters, 1960) și a stat la baza unora dintre cele mai reușite modele de prezicere. Modelele obținute de această metodă folosesc mediile ponderate ale observațiilor anterioare, aceste ponderi scăzând exponențial pe măsură ce observațiile se îndepărtează de pasul prezis. Această metodă reușește să genereze preziceri pentru o gamă largă de serii de timp fără prea multă putere computațională, lucru care face ca metoda să fie folosită în practică.

3.3.1.1 Simple Exponential Smoothing

Cea mai simplă metodă de generare de forecast, bazată pe exponential smoothing se numește, în mod natural, **S**imple **E**xponential **S**moothing (SES). Această metodă poate fi folosită, în special, pentru acele serii de timp unde nu există în mod clar un trend sau un șablon sezonier.

Dacă am folosi o metodă naivă, atunci toate valorile viitoare ale unei serii de timp vor fi egale cu ultima valoare observata:

$$\hat{y}_{T+h|T} = y_T,$$

pentru *h=1,2, ..., N*. Din moment ce această metodă naivă face presupunerea că cea mai recentă observație este singura importantă, toate celelalte observații nu vor furniza nici o informație referitoare la viitor. Această metoda naivă poate fi privită și ca o metodă bazată pe medii ponderate, în care toată importanța este dată doar ultimei observații.

Dacă am folosi metoda mediei, atunci toate valorile viitoare ale unei serii de timp vor fi egale cu media aritmetica a tuturor valorilor observate până în momentul de față,

$$\hat{y}_{T+h|T} = \frac{1}{T}\sum_{t=1}^{T} y_t,$$

pentru *h=1,2, ..., N*. Din moment ce această metodă a mediei face presupunerea că toate observațiile sunt la fel de importante, va acorda aceeași importanță fiecărei observații atunci când va genera predicțiile.

De cele mai multe ori, în practică, avem nevoie de o metodă între aceste două extreme. Spre exemplu, avem nevoie de a găsi o modalitate care să dea o importanță mai mare observațiilor recente decât celor din trecutul îndepărtat. Exact acesta este conceptul din spatele **S**imple **E**xponential **S**moothing. Predicțiile sunt calculate folosind medii ponderate, unde ponderile scad exponențial pe măsură ce observațiile se îndepărtează de momentul prezent – ponderile cele mai mici sunt asociate celor mai vechi observații:

$$\hat{y}_{T+1|T} = \alpha y_T + \alpha(1-\alpha)y_{T-1} + \alpha(1-\alpha)^2 y_{T-2} + \cdots,$$

unde $\alpha \in [0,1]$ este parametrul de netezire. Predicția unui singur pas in viitor, pentru timpul *T+1* este o medie ponderată a tuturor observațiilor seriei $y_1, \dots, y_T$. Rata cu care ponderile scad este controlată de parametrul α.

Predicția la momentul T+1 este egală cu media ponderata dintre cea mai recentă observație y_T și valoarea prezisă anterior $\hat{y}_{T|T-1}$:

$$\hat{y}_{T+1|T} = \alpha y_T + (1 - \alpha)\hat{y}_{T|T-1}.$$

Aceasta ecuație poate fi scrisă pentru fiecare *t = 1, ... T,* iar cu l_0 vom nota valoarea folosită drept valoare estimată la pasul 1:

$$\hat{y}_{2|1} = \alpha y_1 + (1 - \alpha)l_0$$
$$\hat{y}_{3|2} = \alpha y_2 + (1 - \alpha)\hat{y}_{2|1}$$
$$\hat{y}_{4|3} = \alpha y_3 + (1 - \alpha)\hat{y}_{3|2}$$
$$\vdots$$
$$\hat{y}_{T|T-1} = \alpha y_{T-1} + (1 - \alpha)\hat{y}_{T-1|T-2}$$
$$\hat{y}_{T+1|T} = \alpha y_T + (1 - \alpha)\hat{y}_{T|T-1}$$

Dacă vom înlocui fiecare ecuație în cea urmatore, vom obține:

$$\hat{y}_{3|2} = \alpha y_2 + (1 - \alpha)\,[\alpha y_1 + (1 - \alpha)l_0]$$
$$= \alpha y_2 + \alpha(1 - \alpha)\, y_1 + (1 - \alpha)^2 l_0$$
$$\hat{y}_{4|3} = \alpha y_3 + (1 - \alpha)[\alpha y_2 + \alpha(1 - \alpha)\, y_1 + (1 - \alpha)^2 l_0]$$
$$= \alpha y_3 + \alpha(1 - \alpha) y_2 + \alpha(1 - \alpha)^2\, y_1 + (1 - \alpha)^3 l_0]$$
$$\vdots$$
$$\hat{y}_{T+1|T} = \sum_{j=0}^{T-1} \alpha(1 - \alpha)^j y_{T-j} + (1 - \alpha)^T l_0$$

O altă reprezentare este cea bazată pe componente. Pentru **S**imple **E**xponential **S**moothing, singura componentă inclusă este nivelul, notat l_t. Această reprezentare cuprinde ecuația de forecast și cea de netezire pentru fiecare componentă din metodă. Forma bazată pe componente pentru **S**imple **E**xponential **S**moothing este:

Ecuația de forecast: $\hat{y}_{t+h|t} = l_t$

Ecuația de netezire: $l_t = \alpha y_t + (1 - \alpha)\, l_{t-1},$

unde l_t reprezintă nivelul (sau valoarea „smoothed") a seriei la timpul t.

Ecuația de forecast arată că forecastul de la timpul *t+1* este nivelul estimat pentru timpul t. Din ecuația de netezire pentru nivel obținem nivelul estimat al seriei de timp la timpul t. Dacă

vom înlocui l_t cu $\hat{y}_{t+1|t}$ și l_{t-1} cu $\hat{y}_{t|t-1}$ în ecuația de netezire, vom ajunge la forma obișnuită, cu medii ponderate a metodei **S**imple **E**xponential **S**moothing.

3.3.1.2 Double exponential smoothing (Holt's method)

In 1957, Holt[7] a extins **S**imple **E**xponential **S**mooth pentru a putea face forecast pe date care au un anumit trend. Această metodă implica o ecuație de forecast și 2 ecuații de netezire, una pentru nivel și cealaltă pentru trend:

Ecuația forecastului: $\hat{y}_{t+h|t} = l_t + hb_t$

Ecuația nivelului: $l_t = \alpha y_t + (1 - \alpha)(l_{t-1} + b_{t-1})$

Ecuația trendului: $b_t = \beta(l_t - l_{t-1}) + (1 - \beta) b_{t-1},$

unde l_t reprezintă estimarea nivelului seriei la timpul t, b_t reprezintă estimarea trendului seriei la timpul t, $\alpha \in [0, 1]$ este parametrul de netezire al nivelului si $\beta \in [0, 1]$ reprezintă parametrul de netezire al trendului.

La fel ca la **S**imple **E**xponential **S**moothing, ecuația nivelului arată că l_t reprezintă media ponderată a observației y_t și a valorii estimate cu un pas in spate ($l_{t-1} + b_{t-1}$). Ecuația trendului arată că b_t este o medie ponderată a trendului estimat la pasul t ($l_t - l_{t-1}$) și valoarea anterioară estimată a trendului (b_{t-1})

3.3.1.3 Triple Exponential Smoothing (Holt-Winters' seasonal method)

In 1960 Holt și Winters[8] au extins **D**ouble **E**xponential **S**moothing pentru a captura și sezonalitatea. Această nouă metodă cuprinde o ecuație de forecast și trei ecuații de netezire – una pentru nivel l_t, una pentru trend b_t și a treia pentru sezonalitate, având parametri corespunzatori $\alpha, \beta, si\ \gamma$. Cu m vom nota frecvența sezonalității, mai precis, câte sezonalități sunt intr-un an. Spre exemplu, pentru o frecvență trimestrială vom avea m=4, iar pentru o frecvență lunară vom avea m=12.

Cele 4 ecuații corespunzătoare acestei metode sunt:

$$\hat{y}_{t+h|t} = l_t + hb_t + s_{t+h-m(k+1)},$$
$$l_t = \alpha(y_t - s_{t-m}) + (1 - \alpha)(l_{t-1} + b_{t-1}),$$
$$b_t = \beta(l_t - l_{t-1}) + (1 - \beta) b_{t-1},$$
$$s_t = \gamma(y_t - l_{t-1} - b_{t-1}) + (1 - \gamma) s_{t-m},$$

[7] Holt, C. E. (1957). Forecasting seasonals and trends by exponentially weighted averages (O.N.R. Memorandum No. 52). Carnegie Institute of Technology, Pittsburgh USA

[8] Winters, P. R. (1960). Forecasting sales by exponentially weighted moving averages. Management Science, 6(3), 324–342.

unde *k* este partea întreagă a fracției $\frac{h-1}{m}$. Ecuația nivelului este formată dintr-o medie ponderată dintre observația ajustată în funcție de sezonalitate ($l_t = \alpha(y_t - s_{t-m})$ și forecastul non-sezonal ($l_{t-1} + b_{t-1}$), de la timpul *t*. Ecuația trendului este identică cu cea de la **D**ouble **E**xponential **S**moothing. Iar ecuația sezonalității constituie o medie ponderată dintre poziția actuală în sezonul curent și poziția din sezonul trecut, adică *m* periode de timp în trecut.

Ecuația sezonalității poate fi scrisă și sub forma:

$$s_t = \gamma^*(y_t - l_t) + (1 - \gamma^*)\, s_{t-m}.$$

Dacă vom înlocui l_t din ecuația de netezire a nivelului, vom obține:

$$s_t = \gamma^*(1-\alpha)(y_t - l_{t-1} - b_{t-1}) + [1 - \gamma^*(1 - \alpha)]\, s_{t-m},$$

Ecuație identică cu ecuația inițială de sezonalitate, dacă vom specifica faptul că $\gamma = \gamma^*(1 - \alpha)$. În mod normal restricțiile acestui parametru sunt $0 \leq \gamma^* \leq 1$, care se traduce în $0 \leq \gamma \leq 1 - \alpha$.

3.3.2. ARIMA

Modelele ARIMA (**A**uto**R**egressive **I**ntegrated **M**oving **A**verage) oferă o alta abordare foarte populară pentru predicțiile seriilor de timp. Dacă metodele de tip Exponential Smoothing se bazează pe o descriere a trendului și a sezonalității din seria de timp, modelele bazate pe ARIMA încearcă să descrie autocorelația din seria de timp.

3.3.2.1 Modelul AutoRegressive

În multe modele de regresie, se încercă prezicerea variabilei de interes folosind o combinație liniară a unor predictori. În cazul unui model de autoregresie, predicția variabilei de interes se bazează pe o combinație liniară a valorilor din trecut ale aceleași variabile. Termenul de autoregresie sugerează ca este o regresie a unei variabile cu ea însăși.

Un model autoregresiv de ordinul p poate fi scris in felul urmator:

$$y_t = c + \phi_1 y_{t-1} + \phi_2 y_{t-2} + \cdots + \phi_p y_{t-p} + \varepsilon_t,$$

unde ε_t reprezintă zgomotul.

Acest model este ca o regresie multiplă dar cu valori din trecut ale lui y_t drept predictori. Putem să ne referim la modelul descris mai sus ca la AR(p), un model **a**uto**r**egresiv de ordin *p*.

3.3.2.2 Modelul Moving Average

Un model bazat pe fereastră medie (Moving Average) în loc să folosească valorile din trecut pentru a face forecast, folosește erorile estimărilor sale din trecut într-o manieră similară cu modelele de regresie.

Un model care se bazează pe fereastră medie poate fi descris în felul următor:

$$y_t = c + \varepsilon_t + \theta_1 \varepsilon_{t-1} + \theta_2 \varepsilon_{t-2} + \cdots + \theta_q \varepsilon_{t-q},$$

unde ε_t reprezintă zgomotul.

Ne putem referi la acest model ca la MA(q), un model de **m**oving **a**verage de ordin q. Bineînțeles ε_t nu este observat, deci în realitate nu este vorba de o regresie în adevăratul sens al cuvântului.

3.3.2.3 Componenta de Integrare

O serie de timp staționară este cea în care valorile nu depind de timpul în care acestea sunt observate. Din punct de vedere statistic, o serie de timp este staționară dacă proprietățile sale precum media, varianța și autocorelația sunt constante de-a lungul timpului. Altfel spus, o serie de timp care are trend sau sezonalitate nu este staționară deoarece atât trendul cât și sezonalitatea afectează valorile seriei la diferite momente de timp. Pe de altă parte, o serie de timp complet aleatoare este staționară, deoarece nu contează momentul când este observată, ea

va arăta asemănător în orice moment de timp. În general, o serie de timp staționară nu are tipare predictibile pe termen lung.

O serie de timp diferențiată este acea serie de timp care s-a obținut prin scăderea fiecăror două valori consecutive:

$$y'_t = y_t - y_{t-1}.$$

Seria de timp diferențiată va avea doar $T - 1$ valori deoarece este imposibil de calculat y'_1.

Pentru a obține o serie de timp de ordin mai mare se va repeta același mecanism de scădere a fiecăror două valori observate consecutive. În practică, foarte rar apar serii de timp care au nevoie de diferențiere de ordin doi, majoritatea devin staționare după o diferențiere de ordin 1. Procesul invers diferențierii, în acest context, poartă numele de integrare. De aici și numele celei de-a 3-a componente din modelul ARIMA.

3.2.2.4 Modelul ARIMA

Dacă vom combina cele 3 componente și anume diferențierea cu autoregresia și cu modelul bazat pe fereastră medie, vom obține modelul ARIMA. ARIMA este un acronim de la **A**uto**R**egressive **I**ntegrated **M**oving **A**verage, iar în acest context, integrarea este procesul invers diferențierii. Modelul poate fi scris în forma următoare:

$$y'_t = c + \phi_1 y'_{t-1} + \cdots + \phi_p y'_{t-p} + \theta_1 \varepsilon_{t-1} + \cdots + \theta_q \varepsilon_{t-q} + \varepsilon_t,$$

unde y'_t aparține seriei diferențiate. Predictorii din partea dreapta includ atât valorile anterioare ale lui y_t cât și erorile din trecut. Acest model poate fi scris sub forma ARIMA(p, d, q) unde:

4. *p* este ordinul de autoregresie,
5. *d* este gradul de diferențiere,
6. *q* este ordinul ferestrei medii.

3.4 Experimente și rezultate

În această secțiune vom prezenta modul în care a fost modelată problema de forecasting și evaluarea celor 3 algoritmi folosiți: ARIMA, Exponential Smoothing și Prophet. Codul sursă este făcut public pe github, iar analiza poate fi îmbunătățită și chiar extinsă de oricine dorește să facă acest lucru.

Scopul acestei părți a fost de a vedea cum se comporta diferiți algoritmi de forecasting pe serii de timp și de a-i compara între ei. De asemenea, s-a avut în vedere și analiza tiparelor seriilor de timp ce reprezintă cazurile noi, zilnice, cu pacienți infectați și confirmați, cu COVID-19. Pentru acest scop s-au selectat 3 perioade diferite de timp: 1 iulie 2020, 15 noiembrie 2020 și 1 Martie 2021. Aceste perioade sunt considerate a fi mai dificile de prezis din mai multe considerente.

3.4.1 Perioadele selectate

Prima perioadă, cea din 1-14 iulie 2020, este exact după finalul primului val: aceasta înseamnă cazuri noi, zilnice, foarte puține. Spre exemplu în județele Cluj și Timiș în cele 14 zile de test s-au înregistrat 2 zile cu cate 0 cazuri noi și alte zile cu cate 1, 2 sau 3 cazuri; în județul Iași cea mai mica valoare raportata a fost 1, iar in Suceava și Brașov au fost înregistrate minim 3 respectiv 4 cazuri.

O altă observație importantă pentru această perioadă este că pe durata primului val procesul de raportare în aplicațiile oficiale nu erau foarte cunoscute și s-au făcut multe greșeli care au fost corectate prin acele zile cu raportări de numere negative.

Aceste două aspecte fac foarte dificile predicțiile deoarece, în primul rând, nu există o varianță foarte mare în datele ce trebuie prezise, acest lucru generând o eroare procentuală foarte mare (dacă exista un singur caz raportat și predicția a fost de 2 cazuri, atunci eroarea este de 100%). În al doilea rând, calitatea datelor din care algoritmii trebuie să învețe tiparele este pusă sub semnul întrebării, cel puțin în primele luni de raportări oficiale a cazurilor ne noi persoane infectate.

A doua perioadă, 15-29 noiembrie 2020, reprezintă o perioadă foarte dificilă pentru forecasting deoarece în această perioadă are loc vârful celui de-al doilea val epidemic. Cu alte cuvinte, algoritmii învață pe o serie de timp care are un trend ascendent, iar în partea de testare apare punctul de inflexiune ce va schimba trendul într-unul descendent.

A treia perioadă, 1-14 martie 2021, este reprezentata de începutul celui de-al treilea val epidemic, un val de nu are o ascensiune atât de abrupta ca cel de-al doilea.

3.4.2 Județele selectate

Pentru evaluarea metodelor de forecasting am selectat 6 județe din Romania: Iași, Cluj, Timiș, Brașov și Suceava. Primele patru județe au fost selectate datorită faptului că municipiile reședință de județ sunt centre universitare în care testarea cazurilor s-a făcut local încă de la începutul pandemiei. Suceava a fost inclusă în studiu datorită anvergurii cu care s-a intervenit la începutul primului val de infectări cu coronavirus.

3.4.3 Procesul de evaluare

Fiecare model a fost antrenat folosindu-se un istoric de maxim 9 luni de zile în urmă. Excepție face prima perioadă unde, deoacere nu existau date, s-a folosit un istoric de aproximativ 5 luni de zile. Metrica de evaluare folosită este MAPE (**M**ean **A**bsolute **P**ercentage **E**rror), având formula:

$$MAPE = \frac{100}{n} \sum_{t=1}^{n} \left| \frac{A_t - F_t}{A_t} \right|,$$

Unde:

$-\ n$ reprezintă numărul de zile pentru care se face forecast; în cazul nostru vom avea *7* și *14* zile;

$-\ A_t$ reprezintă numărul de cazuri reale raportate, pentru ziua *t* din perioada selectată pentru test;

$-F_t$ reprezintă valoarea prezisă pentru ziua *t* din perioada selectată pentru test.

Pentru fiecare județ și pentru fiecare perioadă s-a calculat eroarea MAPE pentru 7 respectiv 14 zile. Rezultatele obținute de fiecare metodă sunt sintetizate în tabelul de mai jos.

3.4.4 Rezultate

De la o simplă privire asupra tabelului cu erori se poate observa ca Exponential Smoothing este algoritmul care a obținut cele mai bune rezultate. Este de așteptat ca acest tip de algoritm să funcționeze pe astfel de serii de timp deoarece, așa cum am observat în analiza inițială și din corelograme, există o autocorelație foarte mare între valorile apropiate. Acest lucru are o explicație destul de simplă și anume: rata de transmitere a virusului de la persoanele infectate la persoanele sănătoase. Cu alte cuvinte, cu cât sunt mai multe persoane infectate, cu atât este mai probabil ca și mai multe persoane să se infecteze. Acest tipar este modelat cel mai bine de algoritmii din familia Exponential Smoothing care dau ponderi mai mari observațiilor mai apropiate și ponderi mai mici observațiilor mai îndepărtate atunci când fac predicții.

		01 Iulie 2020		15 Noiembrie 2020		1 Martie 2021	
		7 zile	14 zile	7 zile	14 zile	7 zile	14 zile
Iasi	ARIMA	97	55	21	13	12.	22
	Exponential Smoothing	**43**	**44**	**9**	**9**	**9**	**17**
	Prophet	467	460	31	24	11	20
Suceava	ARIMA	54	91	30	47	10	332
	Exponential Smoothing	**30**	**47**	**12**	**25**	**4**	**46**
	Prophet	400	772	60	67	19	277
Cluj	ARIMA	44	56	42	34	17	21
	Exponential Smoothing	**15**	**47**	**14**	**21**	**10**	**15**
	Prophet	104	578	67	50	25	34
Timis	ARIMA	61	89	13	31	10	13
	Exponential Smoothing	**6**	**30**	**9**	**20**	**8**	**12**
	Prophet	63	191	18	26	15	14
Brasov	ARIMA	124	84	89	70	19	37
	Exponential Smoothing	**28**	**41**	**27**	**30**	**6**	**19**
	Prophet	81	187	123	98	22	40

4. Clusterizare

Problema clusterizarii face parte, alături de cea a clasificării și a regresiei, din categoria problemelor clasice din domeniul învățării automate, fiind cea mai cunoscută problemă de învățare nesupervizată/ nesupravegheată. Scopul este acela de a împărți observațiile în așa fel încât cele din interiorul unui grup să fie mai asemănătoare între ele comparativ cu observațiile din celelalte grupuri. Cu alte cuvinte prin clusterizarea unui set de date ne propunen să separăm datele care au trăsături similare în diferite clustere.

Nu există o singură abordare sau un singur model, ci reprezintă mai degrabă o etapă din analiza unui set de date și este folosită în foarte multe domenii precum recunoașterea tiparelor, analiza imaginilor, extragerea de informații, bioinformatică, compresia datelor sau grafică.[9]

Există două tipuri mari de metode de clusterizare:

1. Clusterizare hard: fiecare observație este atribuită unui singur cluster. Spre exemplu, în această lucrare vom grupa județele din România în funcție de cum a evoluat numărul de pacienți infectați cu COVID-19. Fiecare județ va face parte doar dintr-un astfel de cluster.
2. Clusterizare soft: în loc ca fiecare observație sa fie atribuită unui singur cluster, se calculează probabilitatea acesteia de apartenență pentru toate clusterile. Spre exemplu, sa presupunem ca vrem să grupăm județele din România în 3 clustere, pentru județul Iași, în loc să obținem identificatorul clusterului căruia îi aparține județul Iași, vom obține o lista cu 3 probabilități, a căror sumă trebuie sa fie 1.

4.1 Clusterizarea seriilor de timp

Un tip special de clusterizare este cea a seriilor de timp. In general, se spune despre seriile de timp ca reprezinta date dinamice deoarece valorile seriei de timp variaza in functie de timp, ceea ce inseamna ca observatiile sunt cronologice. Chiar daca fiecare serie de timp este alcatuita dintr-un numar mare de puncte, ele pot fi privite si ca obiecte unitare. Clusterizarea unor asemenea obiecte complexe poate duce la descoperirea unor tipare neasteptate intr-un asemenea set de date.

Clusterizarea seriilor de timp este o problemă importantă deoarece seturile de date cu serii de timp conțin informații valoroase ce pot fi obținute în urma descoperirii tiparelor existente, iar clusterizarea este o tehnica ce poate duce la identificarea acestor tipare pe seturile de date alcătuite din serii de timp. De obicei, asemenea baze de date sunt foarte mari si nu mai

[9] https://en.wikipedia.org/wiki/Cluster_analysis

este fezabilă identificarea manuală, de către oameni a acestor tipare. Din moment ce este mai ușor să se lucreze cu seturi de date structurate, datele brute pot fi prelucrate și structurate pe grupuri similare, pot fi agregate sau pot fi ordonate într-o anumită ierarhie.

Definirea problemei

Problema clusterizării seriilor de timp poate fi formulată în felul următor: dându-se un set de date cu *n* serii de timp $D = \{F_1, F_2, \dots F_n\}$, procesul nesupravegheat de partiționare a setului de dat *D*, în mai multe grupuri $C = \{C_1, C_2, \dots C_k\}$, astfel încât seriile de timp omogene să fie grupate împreună bazat pe o anumită metrică de similitudine, se numește clusterizarea seriilor de timp. Trebuie specificat faptul ca C_i - reprezintă un cluster, iar $D = \bigcup_{i=1}^{k} C_i$ și $C_i \cap C_j = \emptyset$, pentru orice i si j. [10]

[10] Saeed Aghabozorgi, Ali Seyed Shirkhorshidi, Teh Ying Wah „Time-series clustering – A decade review"

4.2 Cercetari in domeniu

4.2.1 "Clustering stationary and non-stationary time series based on autocorrelation distance of hierarchical and k-means algorithms"[11]

În acest studiu autorii au făcut o comparație între 2 metode de clusterizare a seriilor de timp și anume: k-means și clusterizare ierarhică folosind 3 metode diferite de a calcula distanța dintre clustere (average, complete și ward). Seturile de date folosite au fost atât date reale referitoare la temperatura din 34 de orase din Indonezia cât și date generate sintetic. S-a avut în vedere crearea și alegerea seturilor de date astfel încât să existe și serii de timp staționare dar și serii de timp nestaționare.

Concluzia experimentului realizat în acest articol este că algoritmul k-means are o acuratețe mai mare atât în cazul datelor sintetice cât și în cazul datelor reale. Pe datele sintetice, în medie, atât pe seriile de timp staționare cât și pe cele nestaționare, k-means a obținut o acuratețe de 84.13%. Iar pe datele reale, k-means o obținut o acuratețe de 85.29%.

4.2.2 „Analysis of similarity measures in times series clustering for the discovery of building energy patterns"[12]

În acest studiu a fost introdusă și testată o nouă metrică pentru a măsura echilibrul din interiorul clusterilor. Această tehnică poate fi utilă atât pentru ajustarea parametrilor, pentru alegerea algoritmului și a altor unelte pentru clusterizare cât și pentru a obține informații cu privire la încrederea pe care o putem acorda unui model.

Pe lângă această metrică, s-au comparat și câteva distanțe de similitudine populare pe un set de date legat de consumul de energie. Printre acestea putem enumera distanțele: euclidiană, mahalonobis corelația Pearson și DTW (**D**ynamic **T**ime **W**arping). Una dintre observațiile autorilor este că, deși setul de date prezintă corelații puternice între seriile de timp și între trăsăturile acestora, distanța euclidiană a obținut cele mai bune rezultate. Cu toate acestea și distanța DTW poate fi considerată o alternativă bună în unele seturi de date.

Concluzia lor este că doar o corelație puternică în problema clusterizarii seriilor de timp nu justifică utilizarea altor distanțe care se concentrează in mod special pe aceste corelații, ci distanța euclidiană este cea care obține rezultatele cele mai bune și că ea este capabilă să captureze în plan secund și aceste corelații.

[11] Mohammad Alfan Alfian Riyadi et all: „Clustering stationary and non-stationary time series based on autocorrelation distance of hierarchical and k-means algorithms"
[12] F. Iglesias, W. Kastner: „Analysis of similarity measures in times series clustering for the discovery of building energy patterns"

4.3 Algoritmi utilizați

4.3.1 K-Means

Algoritmul k-means este unul dintre cei mai cunoscuți și folosiți algoritmi de clusterizare datorită faptului că este practic, iar rezultatele sale sunt, de obicei, bune. Acest algoritm este alcătuit din 2 pași principali, iterativi. O primă fază constă în a defini cei k centroizi corespunzători clusterilor. Următoarea fază este de a itera prin fiecare punct din setul inițial de date și de a-l asocia celui mai apropiat centroid. Se pot folosi mai multe tipuri de distanțe, fie standard, fie adaptate pentru a se calcula distanța dintre puncte și dintre centroizi. Cea mai utilizată distanță este cea euclidiană. Când toate punctele au fost asignate câte unui cluster, se finalizează prima etapa. În acest moment se recalculează pozițiile noilor centroizi, deoarece asignarea punctelor în clustere poate duce la schimbarea locației centroizilor. După ce sunt găsite noile poziții ale celor k centroizi, se reasginează fiecare punct la cel mai apropiat centroid. Aceste acțiuni de calculare a noilor poziții de centroizi și de reasignare a punctelor în cele mai apropiate clustere formează o buclă ce se execută până când se îndeplinește o anumită condiție de oprire.[13]

4.3.2 Fuzzy C-Means

Principiul de funcționare este foarte asemănător cu algoritmul k-means. Singura diferență este tipul de clusterizare pe care o face si anume, clusterizare soft, adică fiecare punct nu este asignat unui singur cluster ci se calculează probabilitățile de apartenență pentru fiecare cluster. FCM este unul dintre cei mai populari algoritmi de clusterizare fuzzy (adică soft) care a fost propus în anul 1973[14] și modificat în 1981.[15] În această abordare punctele au asociate valori de apartenență la fiecare cluster, valori ce sunt actualizate în mod iterativ.

Fie $X = [x_1, x_2 \dots, x_n]$ un set de n obiecte. Pentru a clusteriza X în c clustere, algoritmul standard fuzzy C-Means încearcă să minimizeze următoarea funcție obiectiv:

$$f[U, V] = \sum_{i=1}^{n} \sum_{k=1}^{c} u_{ik}^{m} ||x_i - v_k||^2,$$

unde f se numește funcție de cost, m este parametrul de neclaritate al clusterilor, v_k este interpretat ca fiind centroidul clusterului k, iar u_{ik}reprezintă gradul de asociere dintre obiectul

[13] K. A. Abdul Nazeer, M. P. Sebastian: „Improving the Accuracy and Efficiency of the k-means Clustering Algorithm"

[14] J. C. Dunn (1973): "A Fuzzy Relative of the ISODATA Process and Its Use in Detecting Compact Well-Separated Clusters", Journal of Cybernetics 3: 32-57

[15] J. C. Bezdek (1981): "Pattern Recognition with Fuzzy Objective Function Algoritms", Plenum Press, New York

i și clusterul *k*. În minimizarea funcției de mai sus trebuie să se țină cont de următoarele condiții:[16]

$$u_{ik} \in [0,1], 1 \leq i \leq n, 1 \leq k \leq c$$

$$\sum_{k=1}^{c} u_{ik} = 1, \sum_{i=1}^{n} u_{ik} > 0,$$

Iar

$$u_{ik} = \frac{1}{\sum_{s=1}^{c} (\frac{d_{ik}}{d_{is}})^{\frac{2}{m-1}}}$$

$$v_k = \frac{\sum_{i=1}^{n} u_{ik}^m x_i}{\sum_{i=1}^{n} u_{ik}^m}.$$

O observație importantă este că daca parametru de neclaritate, *m*, este 1, atunci fuzzy C-Means converge, în teorie, către algoritmul K-Means.

4.3.3 K-medoids

Acest tip de algoritm cuprinde mai multe implementări: PAM (**P**artitioning **A**round **M**edoids), CLARA sau CLARANS . PAM a fost propus de către Kaufman și Rousseeuw în 1987[17] și este alcătuit din 2 etape.

Prima fază este cea de inițializare, mai precis de alegerea primilor medoizi. Cea mai simplă implementare este cea de a alege aleatoriu dintre punctele setului de date *k* medoizi.

A doua fază constă într-o iterație de pași al căror scop este acela de a găsi medoizii optimi. La fiecare iterație, pentru fiecare observație din setul de date care nu a fost aleasă drep medoid se calculează dacă ar trebui să fie aleasă sau nu ca medoid pentru următoarea iterație. Examinând fiecare pereche de medoid–observație (non-medoid), algoritmul alege acea pereche care îmbunătățește cel mai bine calitatea globală a clusterilor obținuți. În acest context, calitatea este măsurată ca fiind suma tuturor distanțelor de la un obiect non-medoid la medoidul clusterului de care aparține. O observație este atașată clusterului a cărui medoid se află la cea mai mică distanță față de ea.[18]

[16] Jinglin Xu, Junwei Han et. all: „Robust and Sparse Fuzzy K-Means Clustering"

[17] Kaufman L, Rousseeuw PJ (1987) Clustering by means of medoids. In: Dodge Y (ed) Statistical Data Analysis Based on the L1 Norm and Related Methods, pp 405–416

[18] Lamiaa Fattouh Ibrahim, Manal Hamed Al Harbi: „Using Modified Partitioning Around Medoids Clustering Technique in Mobile Network Planning"

4.3.4 Mixturi Gausiene

Modelele bazate pe mixturi gaussiene (GMM) sunt printre cele mai mature metode statistice de clusterizare. Din punct de vedere al modelului, fiecare cluster poate fi reprezentat matematic drept o distribuție cu parametri. Așadar întregul set de date este modelat de o combinație de distribuții. Aceste modele pot fi scrise sub forma:

$$P(X|\Theta) = \sum_{i=1}^{K} \alpha_i p_i(x|\theta_i),$$

unde parametri sunt $\Theta = (\alpha_1, \alpha_2, \dots, \alpha_k, \theta_1, \theta_2, \dots \theta_k)$ și fiecare p_i este o funcție gaussiană de densitate parametrizată de Θ. Presupunea ce se face aici este că există K astfel de componente de densitate, fiecare având coeficienți proprii.

Fie $X = (x_1, \dots, x_m)$ un set de date de intrare, modelul încearcă să găsească Θ astfel încat să maximizeze $p(X|\Theta)$. Acest proces este cunoscut și sub numele de estimarea de tip **M**aximum **L**ikelihood (ML) a lui Θ. [19]

4.3.5 Self-Organizing map (SOM)

Acest model a fost propus de către Teuvo Kohonen în 1980 și mai este cunoscut și sub numele de „Kohonen map". El se bazează pe Rețele Neuronale Artificiale și este folosit fie pentru clusterizare, fie pentru reducerea dimensionalității, în special la 2 dimensiuni. SOM diferă de alte tipuri de rețele neuronale deoarece aplică un mecanism de învățare competitivă în comparație cu modelele tradiționale care învață din erori, precum backpropagation.

Algoritmul poate fi descris în felul următor [20]:

1. Se inițializează fiecare pondere a nodurilor din grilă;
2. Se alege o observație din setul de date;
3. Pentru fiecare nod se calculează ponderile care sunt cele mai asemănătoare cu observația aleasă. Nodul cel mai asemănător va fi ales ca fiind BMU (**B**est **M**atching **U**nit).
4. Se calculează vecinătatea BMU-ului.
5. Nodul câștigător este recompensat prin a deveni mai asemănător cu observația aleasă din setul de date. Vecinii BMU-ului vor deveni și ei mai asemănători cu observația aleasă. Cu cât un nod este mai aproape de BMU cu atât ponderile lui vor suferi modificări mai mari și cu cât un nod este mai îndepărtat de BMU, rata de învățare va fi mai mică.
6. Se repetă pașii 2-5 un anumit număr de iterații.

[19] Xiaofei He et all: "Laplacian Regularized Gaussian Mixture Model for Data Clustering"
[20] Abhinav Ralhan: „Self Organizing Maps" - https://medium.com/@abhinavr8/self-organizing-maps-ff5853a118d4

4.3 Experimente și rezultate

În primul rând, pentru clusterizarea unui set de date trebuie ales numărul de grupuri în care se dorește a fi împărțit. În cazul nostru, vrem să descoperim care este numărul optim de tenduri-tipar de evoluație a numărului de persoane infectate cu virusul COVID-19.

4.3.1 Metricile folosite

Pentru aceasta am ales să folosim 2 metrici diferite: inerția clusterilor și coeficientul silhouette. Intuitiv, inerția ne spune cât de depărtate sunt seriile de timp din același cluster. Un model este mai bun cu cât inerția totală a clusterilor este mai mică. Intervalul de valori pe care poate să-l ia această metrică începe de la 0 și poate doar să crească fără a avea vreo limită superioară. În schimb, scorul silhouette ne spune cât de diferite sunt seriile de timp dintr-un cluster comparativ cu seriile de timp din celelalte clustere. Intervalul de valori pe care poate să-l ia această metrică este de la -1 la 1. Valoarea 1 înseamnă că fiecare cluster este bine definit și este distinct față de celelalte, 0 înseamnă că distanța dintre clustere este mică sau insignifiantă, iar -1 înseamnă că clusterile produse au fost construite greșit.

4.3.2 Numărul de clustere

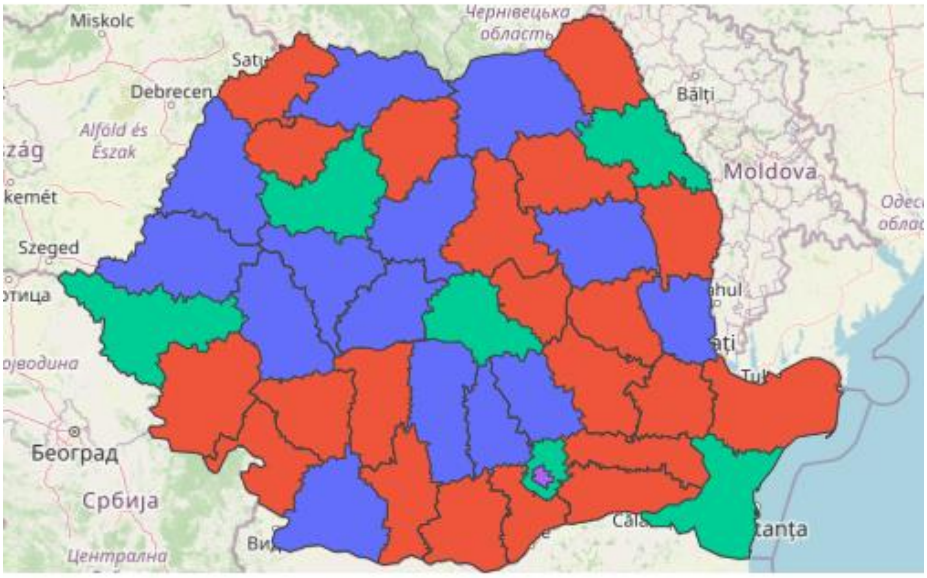
Numărul optim de clustere se poate alege analizând graficele celor 2 metrici. O tehnica folositoare în acest scop este metoda cotului sau metoda „elbow”. Inerția, dintr-un model de clusterizare, scade pe măsură ce crește numărul de clustere, astfel ca se va alege acel număr de clustere ce corespunde punctului de inflexiune de pe graficul inerției și anume acel punct în care adăugarea unui nou cluster nu mai scade semnificativ inerția. În lucrarea de față s-a ales un număr de 4 clustere. Graficele generate cu cele 2 metrici pot fi găsite în anexa E. În figura 30 este scorul silhouette generat de clusterele obținute de algoritmii folosiți. Iar în figura 31 și 32 sunt inerțiile obținute de algoritmii k-means cu distanța euclidiană și cu distanța dtw și de algoritmul k-medoid cu distanța euclidiană.

4.3.3 Rezultate

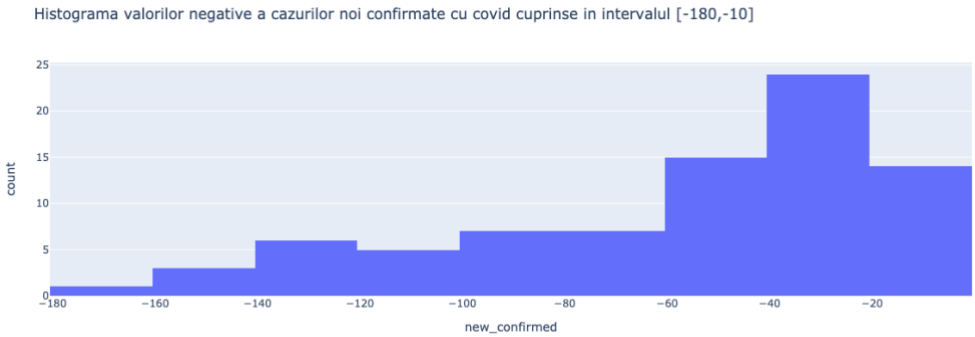
Peste acest set de date s-au rulat mai mulți algoritmi cu diverse distanțe. Dintre aceștia 10 perechi algoritm-distanța au obținut același rezultat și anume: K-Means cu distanțele euclidiana și dtw, Fuzzy C-Means, Mixturi Gaussiene, și k-medoids cu distanțele euclidiană, manhatan, braycurtis, canberra, minkowski și sqeuclidiană.

O observație interesantă este faptul ca București, tot timpul crează un cluster în care doar el este membru, situație ce poate sugera faptul ca este considerat un outlier. Acest lucru este posibil datorită numărului foarte mare de locuitori dar și datorită densității populației comparativ cu populația din județele României.

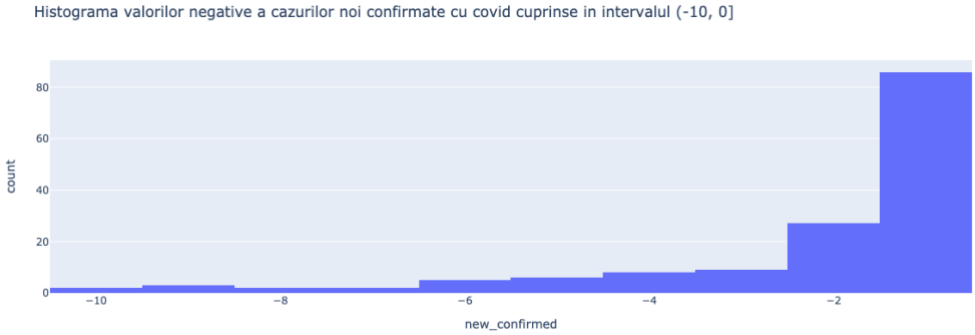
Un alt cluster interesant este cel format din județele: Iași, Cluj, Timiș, Brașov și Constanța. Cu excepția Constanței, toate celelalte județe au în municipiul reședință de județ câte un centru universitar mare și în fiecare dintre aceste municipii exista capacitatea de testare a cazurilor de COVID-19 încă de la începutul pandemiei folosind teste PCR. Următorul cluster este format predominant de județele din Transilvania: Mureș, Sibiu, Alba, Hunedoara, Bihor și Arad, la care se adaugă județele din sud și din centru Prahova, Argeș Dâmbovița și Dolj și județele din Nord și Nord-Est: Maramureș, Suceava, Bacău și Galați. Ultimul cluster este format din județele rămase și anume: Satu-Mare, Sălaj, Bistrița-Năsăud, Botoșani, Neamț, Harghita, Vaslui, Covasna, Vrâncea, Buzău, Brăila, Tulcea, Ialomița, Călărași, Giurgiu, Teleorman, Olt, Vâlcea, Gorj, Mehedinți și Caraș-Severin. Rezultatul clusterizării poate fi vizualizat mult mai ușor în figura de mai jos:



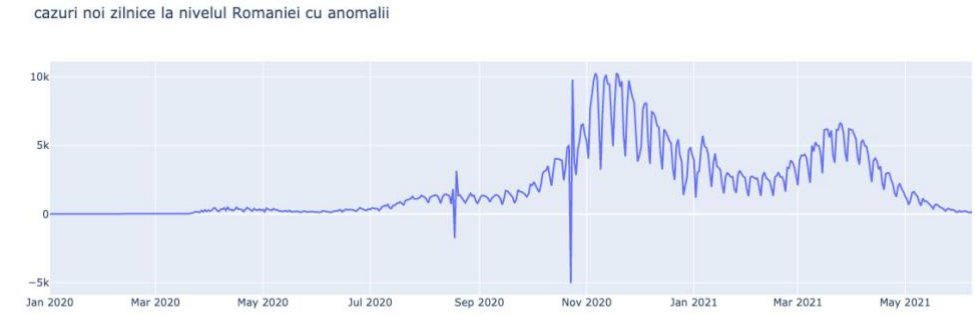
Anexa A



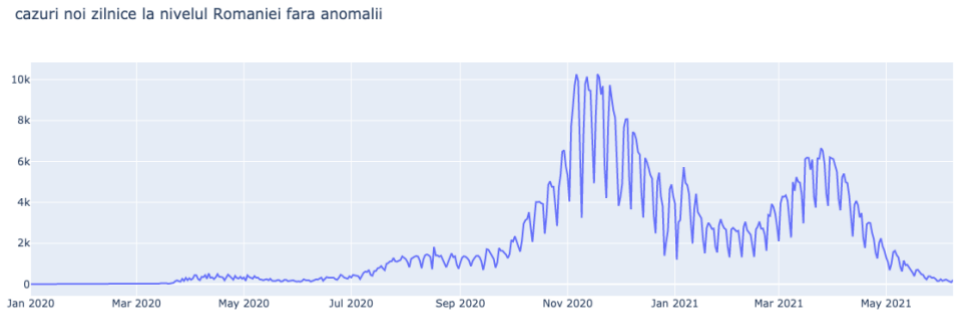
Figură 1



Figură 2

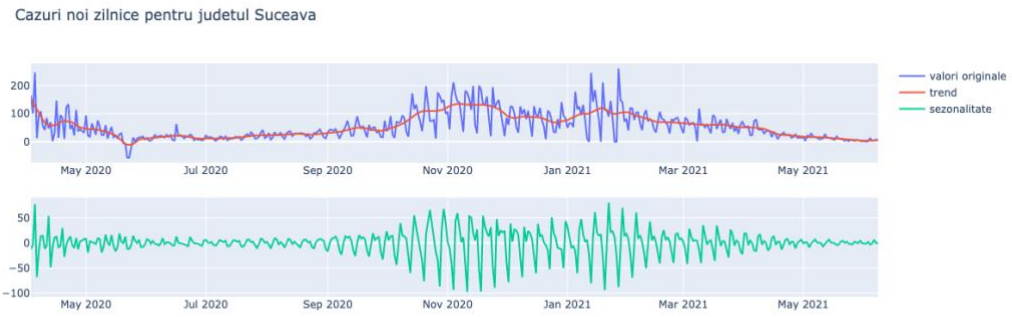


Figură 3



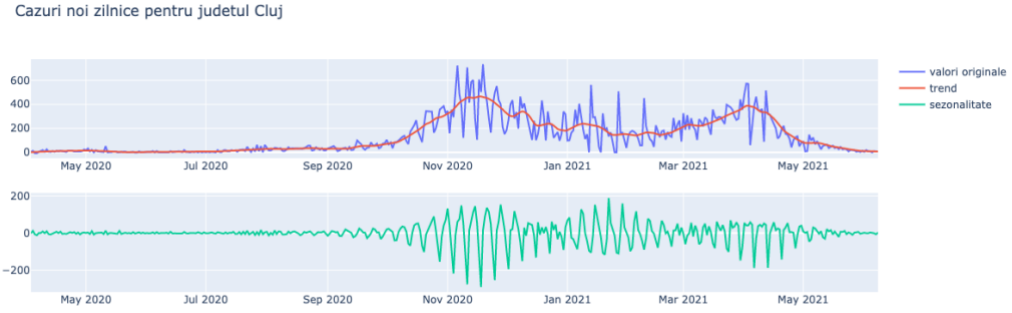
Figură 4

Anexa B

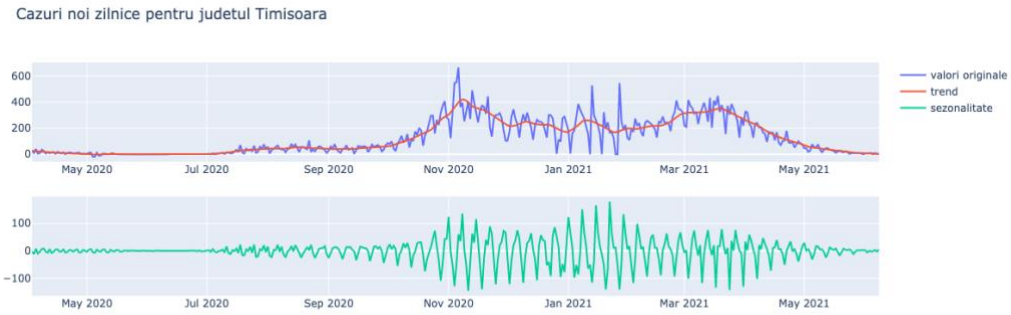


Figură 5

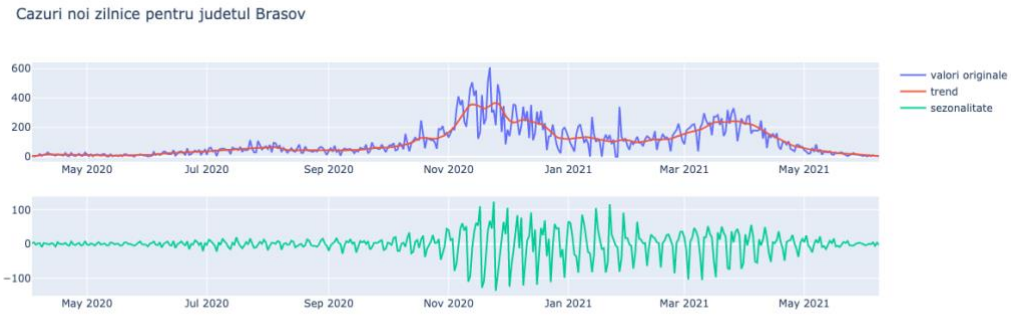
Figură 6



Figură 7

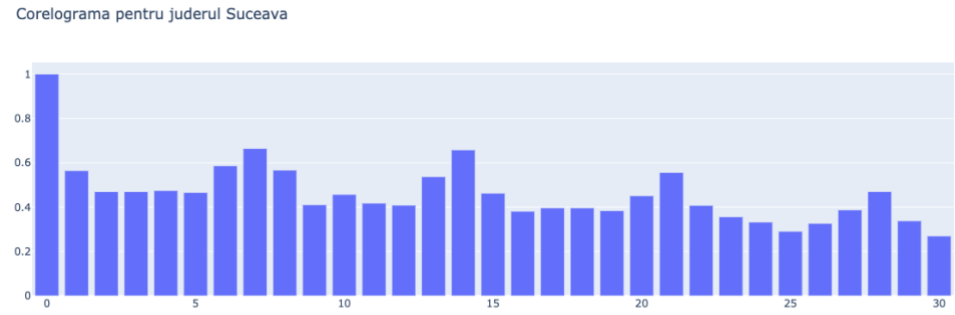


Figură 8

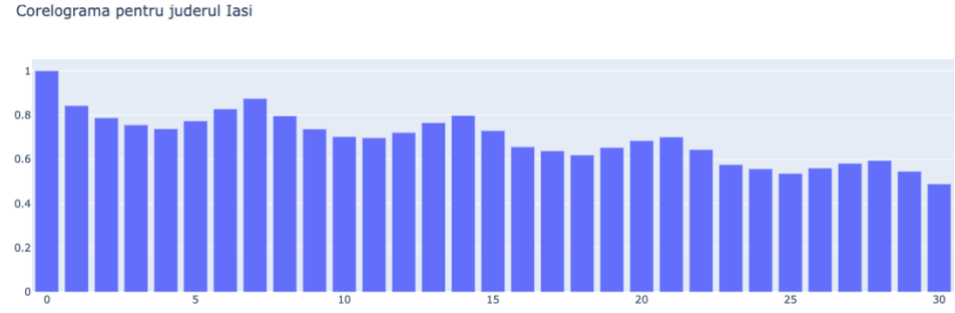


Figură 9

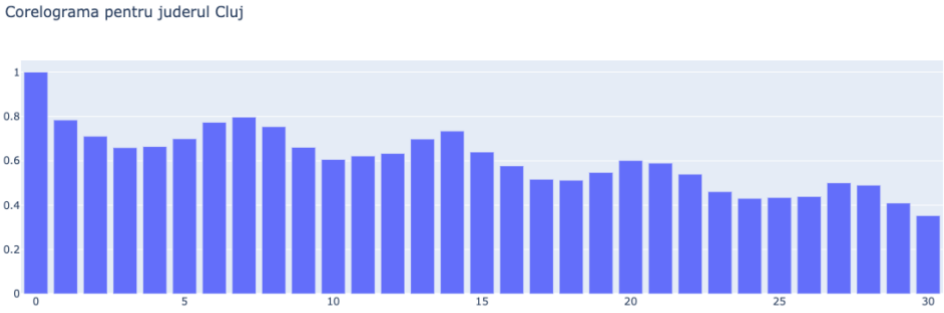
Anexa C



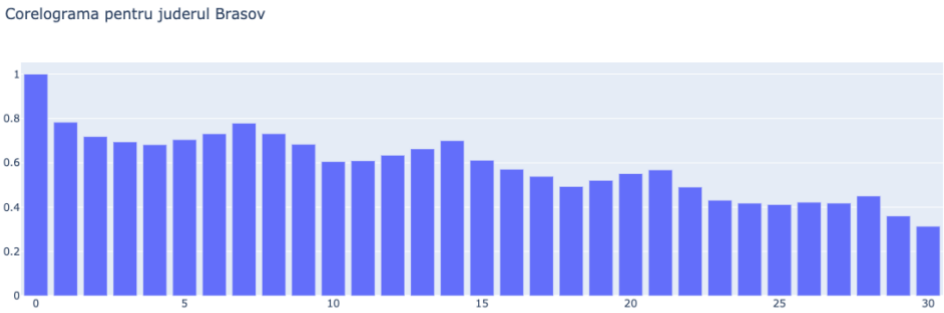
Figură 10



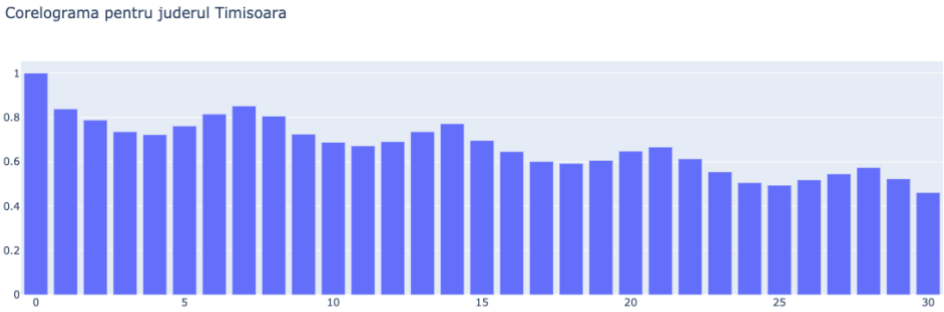
Figură 11



Figură 12

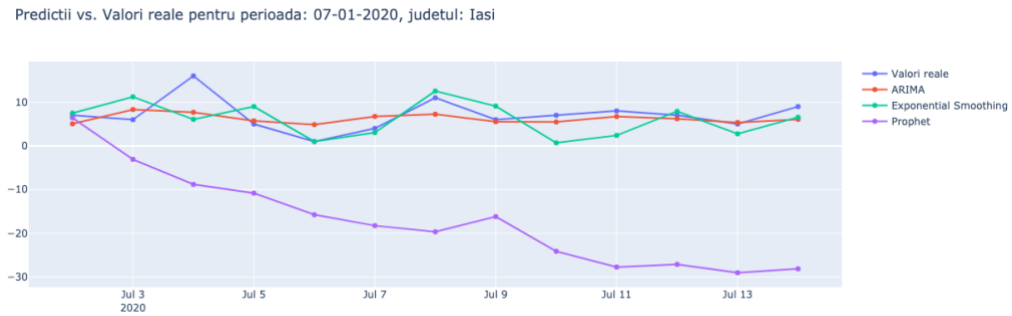


Figură 13

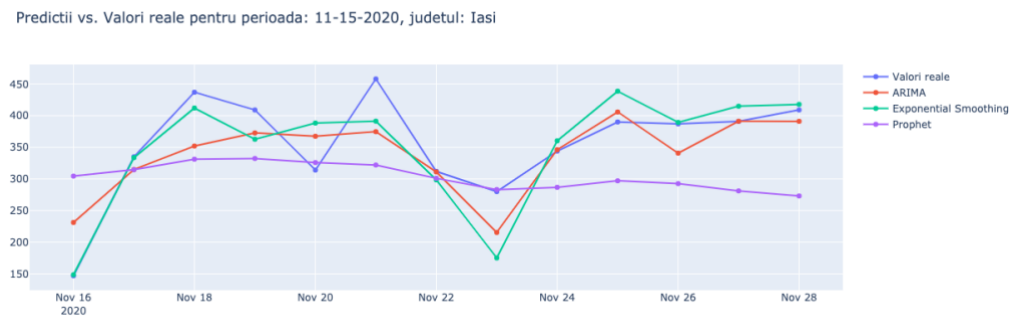


Figură 14

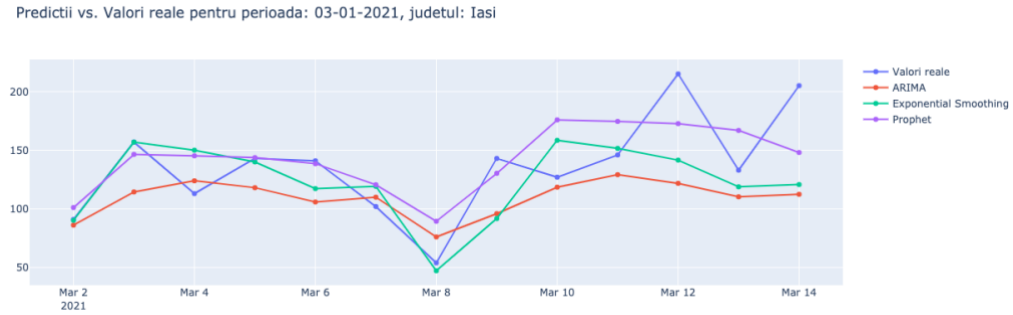
Anexa D



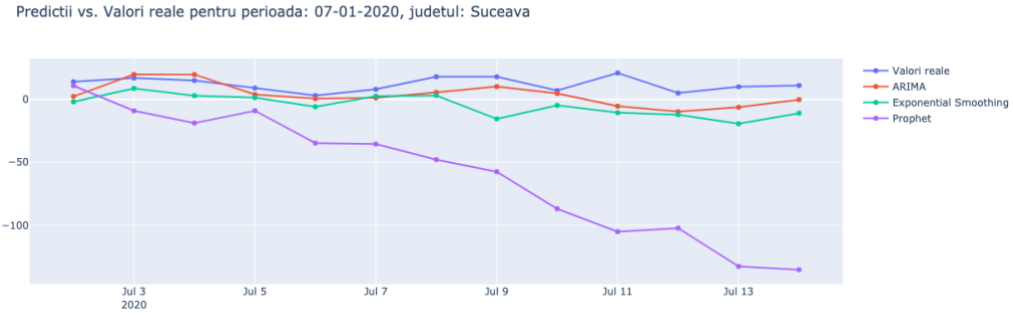
Figură 15



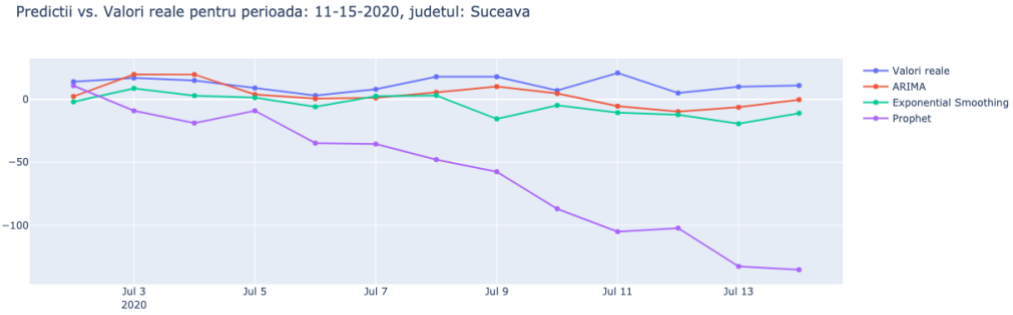
Figură 16



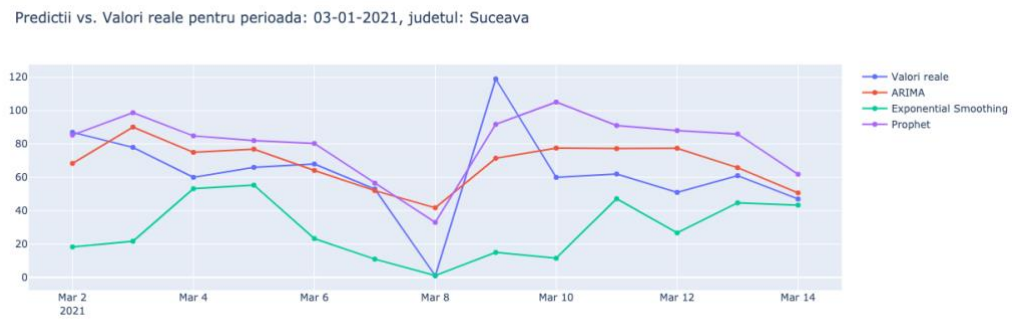
Figură 17



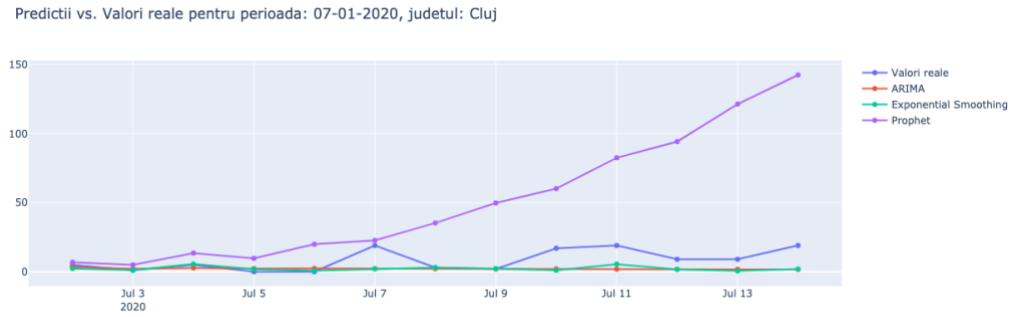
Figură 18



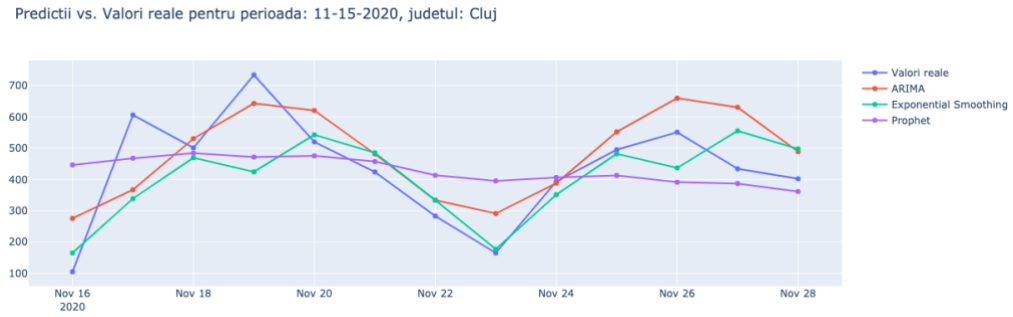
Figură 19



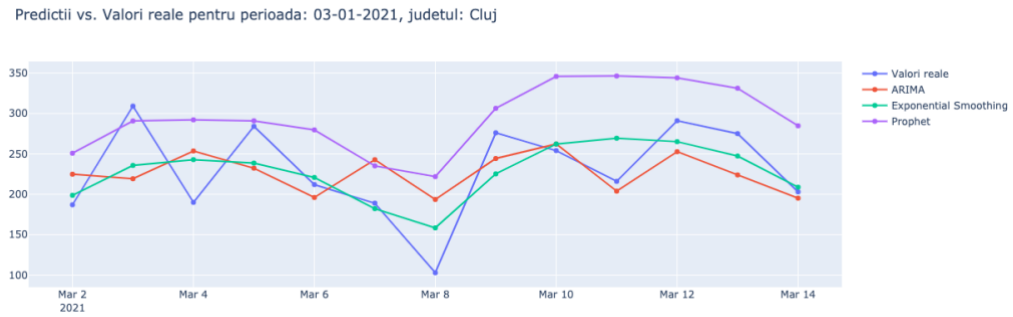
Figură 20



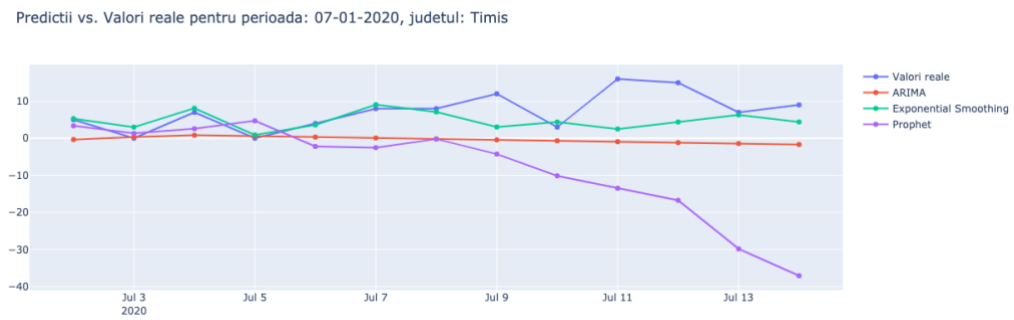
Figură 21



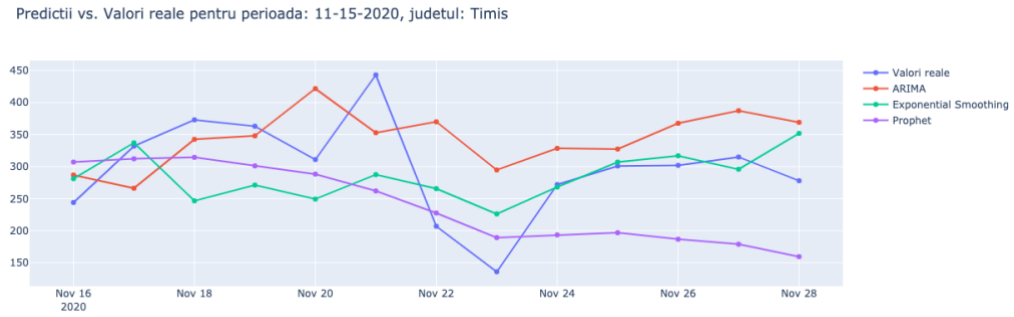
Figură 22



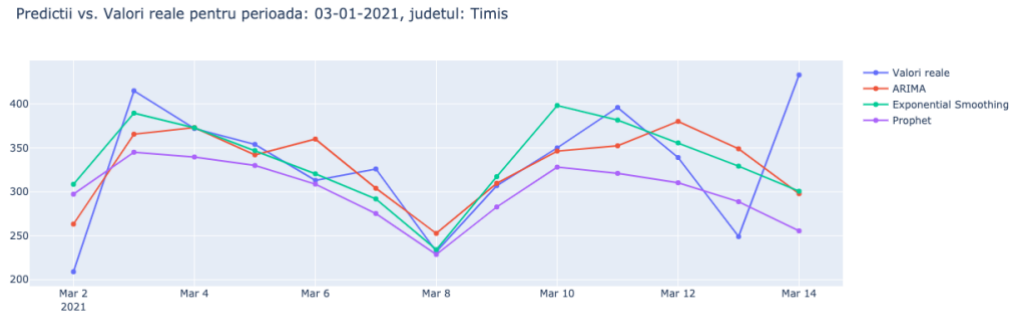
Figură 23



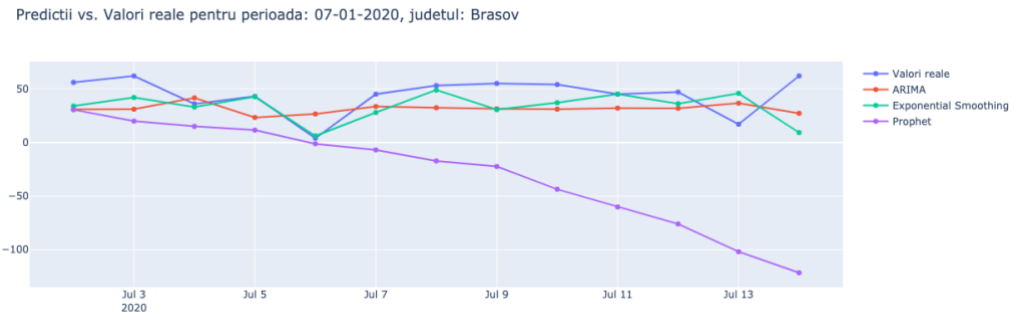
Figură 24



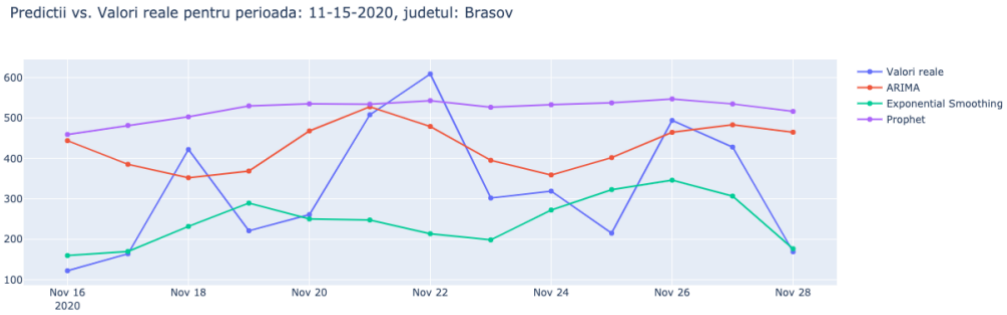
Figură 25



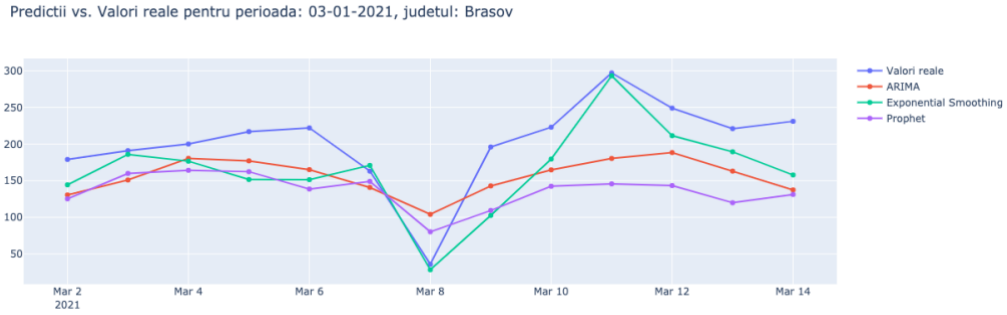
Figură 26



Figură 27

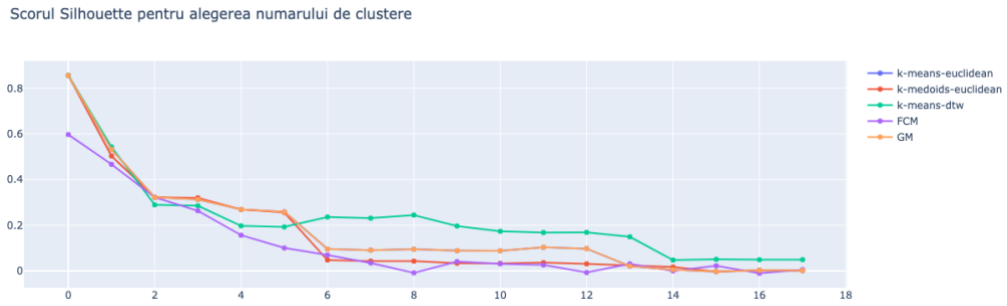


Figură 28

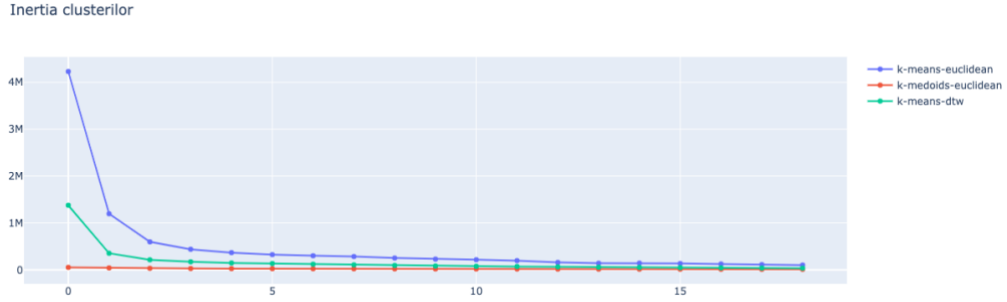


Figură 29

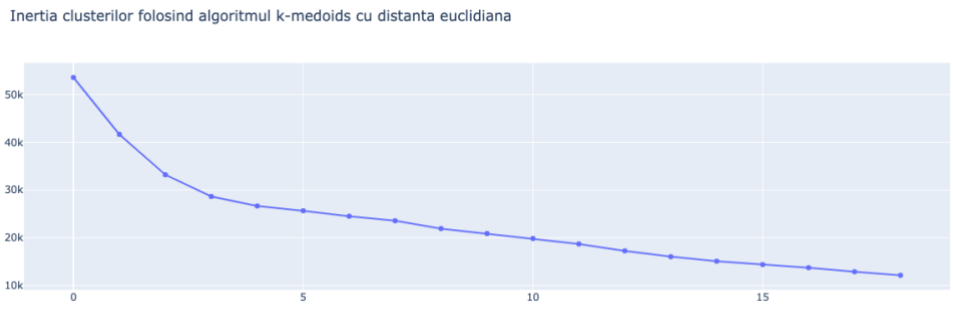
Anexa E



Figură 30



Figură 31


Inertia clusterilor folosind algoritmul k-medoids cu distanta euclidiana
50k
40k
30k
20k
10k
0
5
10
15

Figură 32

Bibliografie si Webografie

Prasad Patil: „What is Exploratory Data Analysis?"
https://towardsdatascience.com/exploratory-data-analysis-8fc1cb20fd15

Hyndman, R.J., & Athanasopoulos, G. (2018) Forecasting: principles and practice, 2nd edition, OTexts: Melbourne, Australia. OTexts.com/fpp2

Chris Chatfield: „Time-series forecasting", p. 3

Papastefanopoulos Vasilis; Linardatos Pantelis; Kotsiantis Sotiris : „COVID-19: A Comparison of Time Series Methods to Forecast Percentage of Active Cases per Population" https://www.mdpi.com/2076-3417/10/11/3880

Fotios Petropoulos; Spyros Makridakis: „Forecasting the novel coronavirus COVID-19" - https://journals.plos.org/plosone/article?id=10.1371/journal.pone.0231236

Matheus Henrique Dal Molin Ribeiro; Ramon Gomes da Silva et. all: „Short-term forecasting COVID-19 cumulative confirmed cases: Perspectives for Brazil"

Holt, C. E. (1957). Forecasting seasonals and trends by exponentially weighted averages (O.N.R. Memorandum No. 52). Carnegie Institute of Technology, Pittsburgh USA

Winters, P. R. (1960). Forecasting sales by exponentially weighted moving averages. Management Science, 6(3), 324–342.

https://en.wikipedia.org/wiki/Cluster_analysis

Saeed Aghabozorgi, Ali Seyed Shirkhorshidi, Teh Ying Wah „Time-series clustering – A decade review"
Mohammad Alfan Alfian Riyadi et all: „Clustering stationary and non-stationary time series based on autocorrelation distance of hierarchical and k-means algorithms"

F. Iglesias, W. Kastner: „Analysis of similarity measures in times series clustering for the discovery of building energy patterns"

K. A. Abdul Nazeer, M. P. Sebastian: „Improving the Accuracy and Efficiency of the k-means Clustering Algorithm"

J. C. Dunn (1973): "A Fuzzy Relative of the ISODATA Process and Its Use in Detecting Compact Well-Separated Clusters", Journal of Cybernetics 3: 32-57

J. C. Bezdek (1981): "Pattern Recognition with Fuzzy Objective Function Algoritms", Plenum Press, New York

Jinglin Xu, Junwei Han et. all: „Robust and Sparse Fuzzy K-Means Clustering"

Kaufman L, Rousseeuw PJ (1987) Clustering by means of medoids. In: Dodge Y (ed) Statistical Data Analysis Based on the L1 Norm and Related Methods, pp 405–416

Lamiaa Fattouh Ibrahim, Manal Hamed Al Harbi: „Using Modified Partitioning Around Medoids Clustering Technique in Mobile Network Planning”

Xiaofei He et all: "Laplacian Regularized Gaussian Mixture Model for Data Clustering"

Abhinav Ralhan: „Self Organizing Maps” - https://medium.com/@abhinavr8/self-organizing-maps-ff5853a118d4