

UNIVERSITATEA ALEXANDRU IOAN CUZA IAȘI
FACULTATEA DE INFORMATICĂ



LUCRARE DE LICENȚĂ
ANALIZĂ STATISTICĂ ȘI PREDICȚIE
A DATELOR CLIMATERICE DIN ROMÂNIA
propusă de

VASILIU VLAD-MIHAI

Sesiunea: Iulie, 2017

Coordonator științific
Conf, Dr. Breabăn Mihaela

UNIVERSITATEA ALEXANDRU IOAN CUZA IAȘI
FACULTATEA DE INFORMATICĂ

**ANALIZĂ STATISTICĂ ȘI PREDICȚIE
A DATELOR CLIMATERICE DIN ROMÂNIA**

Vasiliu Vlad-Mihai

Sesiunea: Iulie, 2017

Coordonator științific

Conf, Dr. Breabăn Mihaela

DECLARAȚIE PRIVIND ORIGINALITATE ȘI RESPECTAREA DREPTURILOR DE AUTOR

Prin prezenta declar că Lucrarea de licență cu titlul „*Analiză Statistică și predicție a datelor climatice din România*” este scrisă de mine și nu a mai fost prezentată niciodată la o altă facultate sau instituție de învățământ superior din țară sau străinătate. De asemenea, declar că toate sursele utilizate, inclusiv cele preluate de pe Internet, sunt indicate în lucrare, cu respectarea regulilor de evitare a plagiatului:

- toate fragmentele de text reproduse exact, chiar și în traducere proprie din altă limbă, sunt scrise între ghilimele și dețin referința precisă a sursei;
- reformularea în cuvinte proprii a textelor scrise de către alți autori deține referința precisă;
- codul sursă, imagini etc. preluate din proiecte open-source sau alte surse sunt utilizate cu respectarea drepturilor de autor și dețin referințe precise;
- rezumarea ideilor altor autori precizează referința precisă la textul original.

Iasi,

Absolvent *Vasiliu Vlad-Mihai*

DECLARAȚIE DE CONSIMȚĂMÂNT

Prin prezenta declar că sunt de acord ca Lucrarea de licență cu titlul „ *Analiză Statistică și predicție a datelor climaterice din România*”, codul sursă al programelor și celelalte conținuturi (grafice, multimedia, date de test etc.) care însoțesc această lucrare să fie utilizate în cadrul Facultății de Informatică. De asemenea, sunt de acord ca Facultatea de Informatică de la Universitatea Alexandru Ioan Cuza Iași să utilizeze, modifice, reproducă și să distribuie în scopuri necomerciale programele-calculator, format executabil și sursă, realizate de mine în cadrul prezentei lucrări de licență.

Iași,

Absolvent *Vasiliu Vlad-Mihai*

Cuprins

Introducere

1. Tehnologii utilizate

1.1. Javascript/JQuery

1.2. HTML

1.3. CSS

1.4. Bootstrap

1.5. AmCharts

1.6. Google Maps Api

1.7. Gentella Admin

2. Arhitectura aplicației

2.1. Data

2.2. Data Transform

2.3. Main

2.4. ChartDraw

2.5. MapGoogle

2.6. Menu

2.7. Hartă

2.8. Statistică

2.9. Predicție

2.10 Index

3. Detalii de implementare

3.1. Datele primite de la Guvernul României și procesarea acestora

3.2 Utilizarea Datelor

3.3. Algoritmi de Predicție

3.4. Afișarea datelor

4. Ghidul aplicației

Concluzii

Bibliografie

Introducere

În Programul de Guvernare 2013-2016, Guvernul României a luat decizia de a înființa cu ajutorul Primului Ministru, a Departamentului pentru Servicii Online și Design.

Acest departament, a publicat o serie de date accesibile, utilizabile în mod liber, fără a ține cont de restricții de tipul drepturi de autor (copyright), patente sau alte mecanisme de control (open data). În acest mod, având acces la aceste informații, cetățenii își vor putea exprima opiniile, sugestiile sau pot folosi aceste date în folosul comunității.

Astfel m-am gândit să fac aplicația web: Analiză Statistică și Predicție a datelor Climaterice din România ce poate deservi cetățenilor României pentru informare, analiză și cunoaștere a spațiului în care trăim.

Lucrarea de față are ca scop realizarea unei platforme ce afișează, analizează și creează o predicție asupra datelor climatologice din România.

Această platformă web permite utilizatorului, în primul rând, vizualizarea resurselor primite de la Departamentului pentru Servicii Online și Design. Aceste resurse sunt afișate într-un mod plăcut pe o hartă. În al doilea rând, utilizatorul are acces la partea de statistică unde, cu ajutorul graficelor și a tabelelor, poate realiza statistici complexe, pe baza datelor afișate. În ultimul rând, utilizatorul poate observa o scurtă utilizare a algoritmilor de predicție folosiți. Și poate analiza și compara temperaturile actuale reale, cu cele prezise de către algoritmi.

1. Tehnologii utilizate

Aplicația web, ”Analiză Statistică și Predicție a datelor Climaterice din România” ,reprezintă un sistem integral. Acest sistem este format dintr-un limbaj de scripting, respectiv Javascript/JQuery, un limbaj de marcare utilizat pentru crearea paginilor web ce pot fi afișate într-un browser, HTML și un standard pentru formatarea elementelor unui document HTML, CSS.

Javascript este un limbaj de programare orientat obiect, bazat pe conceptul prototipurilor. Este folosit mai ales pentru introducerea unor funcționalități în paginile Web.

Folosirea acestui limbaj, îmi aduce avantajul de portare a aplicației pe orice device indiferent de sistem de operare, cum ar fi Windows, Mac OS X sau Linux, fără a perturba funcționalitatea acestuia. Astfel, nu este nevoie sa creez aplicații diferite pentru fiecare sistem de operare în parte.

În plus, standardul JavaScript este ECMAScript. Începând cu anul 2012, toate browserele moderne au suport pentru ECMAScript 5.1.

Aplicația web a fost versionată cu ajutorul unui repository GIT aflat pe Git Hub. Astfel am putut ține evidența dezvoltării și a bugurilor .

În continuare, voi prezenta tehnologiile și algoritmii folosiți pentru realizarea lucrării de licență.

1.1. JavaScript/jQuery

jQuery este o bibliotecă JavaScript scrisă pentru a simplifica scripting-ul pe partea de client. Este gratis și open-source, putând fi folosit de către oricine, oriunde și în orice scop. Este cea mai răspândită bibliotecă JavaScript.

Sintaxa jQuery este concepută pentru a face mai ușoară navigarea unui document, selectarea elementelor DOM-ului, crearea de animații, gestionarea evenimentelor și dezvoltarea requesturilor Ajax. jQuery, de asemenea, oferă capacități de creare a plug-in-urilor peste librăria de JavaScript. Cu alte cuvinte, posibilitatea creării unor widget-uri.

Abordarea modulară a bibliotecii jQuery permite crearea de pagini web dinamice, puternice și a aplicațiilor web mari, cum este și cea prezentată în licență.

jQuery, la bază, este o bibliotecă de manipulare a DOM-ului. DOM-ul este o structură de tip arborescentă reprezentând toate elementele unei pagini web. jQuery simplifică sintaxa pentru găsirea, selectarea și manipularea acestor elemente DOM. De exemplu, jQuery poate fi folosit pentru găsirea unui element în document cu o anumită proprietate, modificarea unuia sau a mai multor atribute ale unui element sau crearea unui eveniment.

De asemenea, jQuery oferă o paradigmă pentru manipularea evenimentului care trece de selecția și manipularea unui obiect DOM simplu. Evenimentul de bază și evenimentul ce se va acționa după finalizarea evenimentului de bază, sunt definite într-un singur pas și într-o singură locație a codului.

Principiile în dezvoltarea unei aplicații web, folosind jQuery sunt separarea JavaScript-ului de HTML și CSS, eliminarea incompatibilității dintre browsere și capacitatea de a accesa noi funcții și metode neexistente în JavaScript.

1.2. HTML

Html este acronimul de la Hypertext Markup Language, și este un instrument de redactare folosit pentru crearea paginilor Web. Când utilizăm codul HTML, un bloc de text este înconjurat de etichete care indică unui browser de internet, modul în care urmează să apară textul (de exemplu cu caractere bold sau cu caractere italice).

Este o colecție de stiluri independente de platformă, care definesc diferitele componente ale unui document Web. Este instrumentul preferat pentru crearea paginilor web, deoarece este înțeles de toate browser ele de internet.

Mulți designeri web care utilizează codul Html, considera că este ușor de învățat și ușor de utilizat, deoarece oferă o abordare care nu se bazează foarte mult pe caracteristici împrumute.

Un alt aspect al popularității sale este capacitatea sa de a se ocupa rapid de documentele mari.

Html-ul te ajută să definești structura unei pagini Web. Este util să ajustezi paragrafe, anteturi și fonturi prestabilite, astfel încât utilizatorul să poată citi întotdeauna textul, indiferent dacă are sau nu fonturile instalate pe propriul calculator.

Acceptarea HTM de către designerii web, le-a permis acestora să se gândească la document, ca la o modalitate de accesare a informațiilor, mai degrabă decât la o colecție de pagini statice care pot fi citite doar atunci când sunt descărcate.

Când cineva introduce într-o adresă URL sau dă click pe o legătura de pagină Web, browserul solicită un document de la un server Web prin Hypertext Transport Protocol sau HTTP.

Serverul trimite documentul înapoi utilizatorului, care este afișat în browser. Lucrurile conținute în document (text, fotografii, fișiere audio și video etc) au fost puse acolo folosind structura HTML.

În concluzie HTML este un limbaj de programare ce deservește creatorilor de pagini Web.

1.3. CSS

Acronimul CSS este reprezentat de Cascading Style Sheet, adică foaie de stil în cascadă. Foaie de stil se referă la documentul în sine.

Foile de stil au fost utilizate pentru proiectarea documentelor de ani de zile. Acestea sunt specificații tehnice pentru aspect. O foaie de stil pentru o pagină web, servește pentru a spune motorului de vizualizare cum să redea documentul vizualizat.

Deși cel mai adesea este folosit pentru a seta stilul vizual al paginilor web și al interfețelor HTML, limbajul poate fi aplicat oricărui document XML¹, SVG² sau XUL³.

Împreună cu HTML și JavaScript, CSS este o tehnologie importantă folosită de majoritatea siteurilor web pentru a crea pagini web

Css este conceput în primul rând pentru a permite separarea prezentării și a conținutului, inclusiv aspecte legate de layout-ul, culorile și fonturile. Această separare poate îmbunătăți accesibilitatea conținutului, poate oferi mai multă flexibilitate și control în specificațiile caracteristicilor de prezentare.

Permite mai multor pagini HTML să partajeze formatarea prin specificarea CSS-ului relevant într-un fișier separat .css și să reducă complexitatea și repetarea în conținutul structural.

¹ <https://ro.wikipedia.org/wiki/XML>

² https://ro.wikipedia.org/wiki/Scalable_Vector_Graphics

³ <https://en.wikipedia.org/wiki/XUL>

1.4. Bootstrap

Bootstrap este cel mai popular framework ⁴web gratuit si open-source pentru web design-urile si aplicațiile web.

Acesta conține șabloane de design HTML și CSS pentru formulare, butoane, meniuri si alte componente de interfață, precum și extensii JavaScript.

Spre deosebire de multe alte framworkuri de acest gen, acesta se referă doar la dezvoltarea font-end-ului.

Bootstrap este al doilea proiect cu cele mai multe downloadari de pe GitHub.

Twitter Blueprint, numele inițial al frameworkului, a fost dezvoltat de Mark Otto și Jacob Thornton la Twitter ca un proiect cadru pentru a încuraja consistența peste proiectele companiei. Înainte de Bootstrap, s-au folosit diferite biblioteci pentru dezvoltarea interfeței, ceea ce a dus la inconsecvențe și o mare povară de întreținere.

După câteva luni de la începutul dezvoltării Bootstrap, mulți dezvoltatori de pe Twitter au început să contribuie la proiect ca parte a Hack Week, o săptămână în stil hackathon-week.

A fost redenumit de la Twitter Blueprint la Bootstrap și a fost lansat ca un proiect open source pe 19 august 2011. A continuat să fie menținut de Mark Otto , Jacob Thornton și de un mic grup de dezvoltatori de bază, precum și de o mare comunitate de contribuabili de pe GitHub.

La data de 31 ianuarie 2012, a fost lansată versiunea a doua, Bootstrap 2, care a adăugat un sistem de structurare a HTML-ului, foarte util și astăzi.

Pe 19 august 2013, a fost lansat Bootstrap 3 , care a redesenat componentele pentru a utiliza designul plat și o abordare mobile pentru prima dată. Această versiune am folosit-o și eu pentru a mă ajuta pe partea de împărțire a paginilor HTML.

Bootstrap 3, acceptă cele mai recente versiuni ale browserului Google Chrome, FireFox, Edge, Opera și Safari. Acesta susține în plus si versiunile IE8.

⁴ <https://ro.wikipedia.org/wiki/Framework>

Începând cu 2.0, Bootstrap suportă un design web sensibil. Acesta înseamnă că aspectul paginilor web se ajustează dinamic, ținând cont de caracteristicile dispozitivului utilizat (desktop, tabletă, telefon mobil).

Original creată de un designer și un developer la Twitter⁵, Bootstrap a devenit cea mai mare și populară librărie de front-end din lume.

1.5. AmCharts

AmCharts este un framework care se concentrează pe dezvoltarea de instrumente de programare, orientate spre dezvoltatori pentru vizualizarea datelor, diagrame și hărți.

Situat în Vilnius, Lituania, echipa amCharts este formată din persoane care trăiesc în Lituania, Germania și Statele Unite.

AmCharts se concentrează asupra dezvoltării în jurul tehnologiei JavaScript și a tehnologiilor asociate. Ca standard cross-platform⁶, Javascript este susținut de marea majoritate a dispozitivelor și platformelor, browsere moderne pentru desktop și mobil, precum și pentru cele vechi.

Produsele AmCharts vizează în principal dezvoltatorii de aplicații web, căutând o modalitate de a adăuga capabilități de vizualizare a datelor produselor pe care le dezvoltă. Aceste produse necesită o anumită cunoaștere a tehnologiilor web pentru implementare.

Această bibliotecă are ca scop să ofere o modalitate rapidă de vizualizare a datelor folosind tipuri clasice de diagrame precum Pie, Donut, Line, Area, Column, Bar etc.

1.6. Google Maps Api

API-urile Google reprezintă un set de interfețe de programare a aplicațiilor dezvoltate de Google care permit comunicarea cu serviciile Google și integrarea acestora în alte servicii.

Câteva exemple ar fi : Search, Gmail, Google Translate sau Google Maps.

Aplicațiile third-party, ca și aceasta, pot utiliza aceste interfețe pentru a profita sau extinde funcționalitatea serviciilor existente.

⁵ <https://ro.wikipedia.org/wiki/Twitter>

⁶ <https://ro.wikipedia.org/wiki/Multiplatform>

API-urile furnizează funcționalități precum, analiză, machine learning⁷ ca serviciu sau accesul la datele utilizatorului.

Un alt exemplu important, cel pe care l-am folosit și eu, este Google Maps, încorporat pe un site Web, care poate fi realizat, utilizând API-ul hărților statice, API-urile locațiilor sau API-ul Google Earth.

API-ul Google Maps a fost lansat în iunie 2005, pentru a permite dezvoltatorilor să integreze Google Maps pe site-urile lor.

Prin utilizarea API-ului Google Maps, este posibilă încorporarea hărții într-un site web extern. Api-ul Google Maps, cuprinde un serviciu pentru recuperarea imaginilor statice, un serviciu pentru geolocații, generarea de indicații de conducere și obținerea elevației.

Peste 1.000.000 de site-uri web utilizează API-ul Google Maps, făcându-l cel mai puternic API pentru dezvoltarea aplicațiilor web ce afișează sau utilizează locații pe hartă.

Aplicarea API-ului Google Maps, este gratuită pentru uz comercial, cu condiția ca site-ul pe care este folosit, să fie accesibil publicului și să nu perceapă o plată pentru acces. O altă condiție este ca utilizarea API-ului să nu genereze mai mult de 25.000 de requesturi.

Succesul API-ului Google Maps a provocat o serie de alternative concurente, printre care API-ul Here Maps, Platforma Bing, Leaflet. Cu toate acestea, am decis să folosesc API-ul Google Maps, deoarece este foarte ușor de folosit și funcționalitățile sunt bine descrise.

1.7. Gentellella Admin

Gentellella Admin este un șablon de administrare gratuit creat pe baza frameworkului Bootstrap.

Acest șablon folosește stilurile implicite Bootstrap 3 împreună cu o varietate de pluginuri și instrumente puternice jQuery pentru a crea un cadru puternic administrativ.

Tema utilizează mai multe biblioteci pentru diagrame, calendar, validări, interfețe, meniuri de navigare, formulare de text, intervale de date, bare de progres, notificări și multe altele.

Utilizând această temă, pot oferi un template⁸ uniform și bine structurat licenței, fără a fi nevoie de un designer.

⁷ https://en.wikipedia.org/wiki/Machine_learning

Gentelella Admin este licențiată sub licența MIT. Ceea ce înseamnă că poate fi utilizată, copiată, modificată, publicată si distribuită.

⁸ <https://en.wikipedia.org/wiki/Template>

2. Arhitectura aplicației

Lucrarea de față este formată din mai multe componente: data, dataTransform, main, ChartDraw, mapGoogle, menu, Harta, Statistică și Predicțiile.

Cu toate acestea cele mai importante componente sunt Harta, Statistica și Predicția. Acestea furnizează și apelează datele finale, obținându-se astfel lucrarea de licență.

Aplicația web are rolul de a afișa informațiile primite de la data.gov.ro într-un mod plăcut.

Fiecare dintre aceste trei mari componente: hartă, Statistică și Predicție. Folosește funcții și date furnizate de către aceste mini-componente: data, dataTransform, main, ChartDraw, mapGoogle si menu.

În continuare vă voi prezenta și explica pe larg fiecare atribuție a fiecărei componente ce a dus la realizarea licenței.

Structura generală a aplicației este prezentată în figura 1.

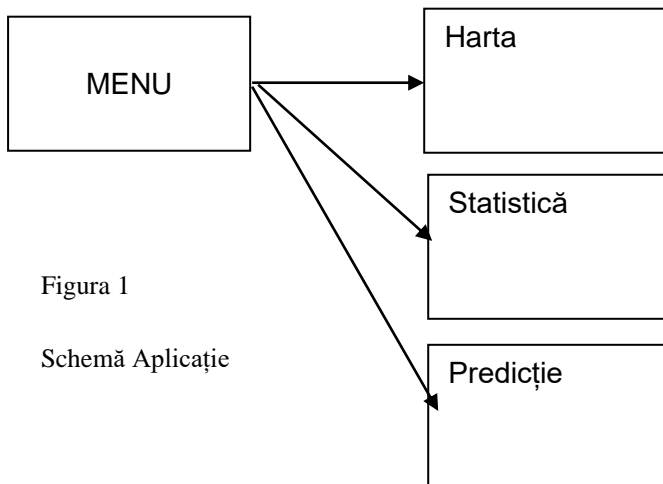


Figura 1

Schemă Aplicație

2.1. Data

Această componentă reprezintă datele ce vor fi folosite în afișarea tabelelor, diagramelor și chiar a hărții.

Cu ajutorul componentei main, am realizat formarea unui JSON ce conține datele furnizate de către data.gov.ro . Datele primite de la data.gov sunt sub formă de fișier CSV⁹.

Acest format de fișier, CSV sau Comma-Separated Values, stochează date tabulare (numere și text). Fiecare linie a fișierului este o înregistrare de date. Fiecare înregistrare constă într-unul sau mai multe câmpuri, separate prin virgule. Utilizarea virgulei ca separator de câmp este și sursa numelui pentru acest format de fișier.

Formatul de fișier CSV nu este standardizat. Ideea de bază de separare a câmpurilor cu virgulă este clară, dar ideea devine complicată atunci când datele pot conține și virgule sau chiar linii. Spre fericirea mea, datele primite sub acest format sunt valide și am reușit să le transform sub formatul JSON¹⁰.

JSON sau JavaScript Object Notation, este un format de date independent de limbaj. Acesta a fost derivat din JavaScript, dar începând din 2017 multe limbaje de programare includ funcții pentru a genera și analiza date în format JSON.

Este un format text, inteligibil pentru oameni, utilizat pentru reprezentarea obiectelor și a altor structuri de date și este folosit în special pentru a transmite date structurate prin rețea, procesul purtând numele de serializare.

Am folosit acest format de date, deoarece este o alternativă mai simplă, mai facilă decât limbajul XML. Eleganța formatului JSON provine din faptul că este un subset al limbajului JavaScript, fiind utilizat alături de acest limbaj.

În primă fază a aplicației am folosit acest serviciu, pentru a face requesturi către data.gov.ro și pentru a primi ca răspuns acel fișier CSV, despre care am explicat mai sus .

Apoi, în următoarea fază, acest fișier CSV, era transformat de către un serviciu YAHOO într-un JSON. Acest serviciu a fost valabil până la data 01.06.2017. După această perioadă, serviciul a putut transforma doar 5000 de date din fișierul CSV în JSON.

⁹ https://en.wikipedia.org/wiki/Comma-separated_values

¹⁰ <https://ro.wikipedia.org/wiki/JSON>

Astfel, am renunțat la acest serviciu, deoarece fișierul CSV era format din aproximativ 9000 de date și cum serviciul returna doar 5000 dintre acestea, nu mi-ar mai fi fost de folos.

Spre dezamăgirea mea nu am găsit nici un serviciu asemănător cu a celor de la Yahoo și astfel am fost nevoit să construiesc singur JSON-ul cu datele primite de la data.gov.ro .

În final, am reușit obținerea unui JSON cu numele bigData. Acesta conține toate datele din 1961 până în 2016. Un exemplu de structură a acestui JSON este la Figura 2.

```
{  
  
  "CODST": "15015",  
  
  "ALT": " 503.00",  
  
  "LAT": "47.7769444",  
  
  "LON": "23.9405556",  
  
  "DATCLIM": "1961/01/01",  
  
  "TMED": " -0.1",  
  
  "TMAX": " 1.0",  
  
  "TMIN": " -1.3",  
  
  "R24 ": " "  
  
}
```

Figura 2. Obiect JSON

După cum se observă în figura 2, acest JSON este împărțit pe ani, fiecare an fiind împărțit pe zile. Spre exemplu, acest JSON din figura 2 reprezintă datele din 01.01.1961.

"CODST" reprezintă Codul Stației meteo de unde au fost preluate restul datelor. Acest cod este reprezentat de către Stația Ocna Șugatag. Următoarele date, spre exemplu : "ALT" reprezintă altitudinea stației meteo. "LAT" și "LON" reprezintă Latitudinea și Longitudinea poziției stației meteo Ocna Șugatag. Aceste informații le voi folosi pentru a plasa pe hartă stația. "TMED" , "TMAX" și "TMIN" reprezintă Temperatura Medie, Temperatura Maximă și Temperatura Minimă furnizată de Stația meteo Ocna Șugatag la data 01.01.1961.

2.1. DataTransform

DataTransform este un fișier de tip JavaScript, ce îl utilizez în transformarea datelor din fișierul Data, în funcție de ce tip de date îmi trebuie și în ce ordine sa le primesc.

Spre exemplu, pentru Hartă, am nevoie de date cum ar fi: Longitudine, Latitudine, Tmed, Tmax, Tmin și numele Stației Meteo.

Pentru a afișa datele pe hartă, eu am nevoie de un JSON diferit față de cel din componenta Data. În această componentă datele sunt sub forma an – stație meteo – data . Eu, pentru a afișa datele în componenta Hartă, am nevoie de date structurate pe zile pentru fiecare din cele 23 de stații Meteo.

Astfel, prin selecția unui an, apoi a unei luni și mai apoi a unei zile, prin apelarea unei funcții, voi returna un JSON, ce îl folosesc în componenta Hartă pentru a desena stațiile și informațiile meteo din acea zi.

Pentru componentele Statistică și Predicție folosesc alt tip de date. Aceste date sunt sub formă de date medii adunate din fiecare zi, astfel obținând un JSON structurat pe ani cu date medii de la fiecare stație. Apoi ce aceste date sunt aplicați algoritmi de Statistică și Predicție.

Toate aceste date sunt transformate apelând diverse funcții.

2.3. Main

Main, este fișierul JavaScript, ce conține diverse funcții ajutătoare, ce le apelez în componentele Hartă, Statistică și Predicție. Aceste funcții sunt de forma verificarea unui an dacă este sau nu bisect, sau funcții ce returnează numărul de zile ale unei luni.

Acest fișier conține funcții ce se apelează doar la un moment dat, cum ar fi funcțiile Play și Stop, ce sunt apelate în componenta Hartă. Funcția Play, afișează la secundă datele unei zile din luna x și anul y a unei stații meteo. Pentru fiecare secundă se afișează datele din ziua următoare.

Alte funcții prezente în acest fișier JavaScript sunt apelate la anumite evenimente de tip click sau mouse over. Pentru buna funcționare a aplicației.

2.4. CharDraw

Această componentă, este exclusiv creată și folosită pentru crearea și modificarea graficelor prezente în aplicație.

Fiecare grafic din această aplicație folosește proprietăți diferite, axe x și y diferite și cel mai important, date diferite. Astfel este nevoie de un gen de fișier JavaScript, ce deservește fiecărei diagrame în parte. Prin acest fișier, nu aglomerez componenta principală, spre exemplu componenta Statistică sau componenta Predicție. Aceste componente doar apelează funcții din fișierul Chart, și acesta returnează graficul dorit, prezentând datele corecte.

2.5. Map google

O componentă importantă în ergonomia și buna funcționare a aplicației este MapGoogle. Această componentă este reprezentată de un fișier JavaScript.

Prin fișierul mapGoogle.js, înlănțui o serie de funcții și date ce îmi folosesc la crearea hărții. Aici folosind API-ul de la Google Maps, declar în primul rând particularitățile hărții, cum ar fi longitudinea și latitudinea la care vreau să fie afișată harta. În cazul de față, folosesc latitudinea de 45.8207835 și longitudinea de 24,8523799 ce reprezintă harta României.

O altă întrebuințare a acestui fișier este crearea legendei. Această legendă ajută utilizatorul să perceapă cu aproximație ce temperatură este afișată pe hartă.

În continuare, sunt utilizate 2 funcții extrem de importante. Write și Clear. Cu ajutorul acestor funcții, harta va primi o oarecare animație, afișându-se temperaturile pe hartă în dreptul stațiilor meteo. Fiecare stație este exemplificată pe hartă cu un cerc. Acel cerc are o culoare diferită în funcție de temperatura stației din ziua prestabilită. Un exemplu se poate observa în Figura 3.

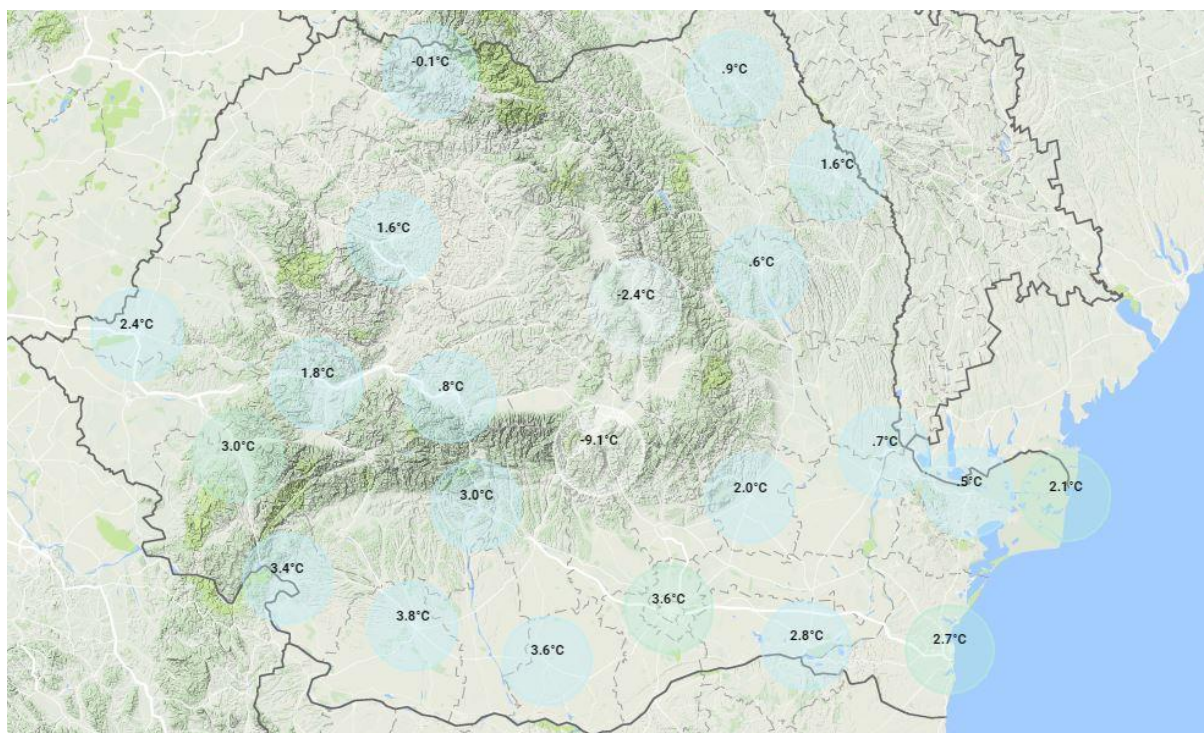


Figura 3. Harta României

Dând click pe oricare din stație putem observa mai multe detalii legate de acea stație. Detalii precum : Numele Stației, Temperatura Maximă , Temperatura Medie, Temperatura Minimă, Altitudinea Stației Meteo și Data. Aceasta se pot observa în Figura 4 .



Figura 4. Detalii

2.6. Menu

Menu este o componentă cu ajutorul căreia separ cele trei componente importante. Astfel, componentele Hartă, Statistică și Predicție sunt complet separate și sunt afișate doar la selecția utilizatorului.

By default, aplicația pornește cu componenta Hartă, astfel fiind vizibilă Harta și cu funcțiile specifice componentei.

2.6. Hartă

Această componentă este definitorie pentru prezentarea datelor către utilizator și însumează o serie de funcții și metode apelate pentru a afișa harta și pentru a putea acționa meniul de comandă, ce deservește utilizatorului în utilizarea funcțiilor componentei.

Meniul de comandă este format în prima parte din trei selecții strâns legate una de alta.

Prima selecție este cea ce implică alegerea anului. Această alegere duce la aplicarea unei funcții, ce returnează ca rezultat o serie de date ce vor fi utilizate mai departe.

A doua selecție este cea ce implică alegerea lunii. Folosind datele primite după alegerea anului, vom împărți rezultatul în funcție de luna aleasă.

A treia selecție și ultima, va fi alegerea zilei. Această zi va fi selectată în funcție de ce dorește utilizatorul să fie afișat pe hartă.

Astfel, urmând aceste selecții vom ajunge să afișăm pe hartă datele meteorologice, din anul, luna și ziua, selectate anterior din meniul de comandă de către utilizator.

După cum se observă, această componentă însumează utilizarea multor componente descrise anterior. Având o hartă de afișat, folosim componenta Map google. Având date de afișat folosim componenta Data, dar această componentă conține datele ne transformate. Folosind meniul de comandă, utilizăm și componenta Data Transform. Aceasta mă ajută cu ajutor funcțiilor dezvoltate să selectez datele pentru an, mai apoi datele pentru luna și în ultimul rând datele pentru zi. Acestea fiind afișate într-un final pe hartă.

Cu ajutorul componentei Main, folosim funcția Play și Stop, pentru a afișa dinamic în componenta Hartă, datele prestabilite prin selecția anului, lunii și a zilei.

2.7. Statistică

Componenta Statistică, este a doua componentă importantă din această aplicație. Aceasta conține un selector prin care putem alege oricare din cele douăzeci și trei de stații meteo.

Selectând una dintre aceste stații, vom primi o serie de statistici. Una dintre ele ar fi un tabel pe baza căruia putem observa Temperaturile Medii, Temperatura Medie Maxima și Temperatura Medie Minima a stației respective. Mai apoi acest tabel este transformat într-un grafic ce putem observa cu ușurință diferențele de temperaturi dintre ani.

Următorul grafic, reprezintă temperaturile medii pe zile ale stației selectate. Acesta este creat printr-o serie de transformări ale datelor primite de la data.gov.ro .

Ultimul grafic al componentei Statistică, este reprezentat de către datele climaterice din toate cele 23 de stații meteo.

2.8. Predicție

Această componentă este una cu totul și cu totul importantă și deosebită, deoarece reprezintă baza licenței. Aici se află implementați o serie de algoritmi de predicție, cu ajutorul cărora nu doar am prezis viitoarele temperaturi, ci în plus am comparat predicțiile cu anii anteriori, astfel observând ce algoritmi de predicție sunt mai exacti.

Componenta Predicție se aseamănă foarte mult cu componenta Statistică, deoarece este reprezentată tot de tabele și grafice, ce afișează datele obținute în urma implementării algoritmilor de predicție.

O parte din algoritmi importanți îi vor doar enumera aici, și îi voi explica în amănunt în paginile următoare ale licenței.

- Metoda Modificării Procentuale
- Metoda Nivelării exponențiale
- Trendul
- Metoda Brown
- Metoda Holt

2.9. Index

Această component reprezintă de fapt integrarea celor trei mari componente : Hartă, Statistică și Predicție. Aici sunt integrate toate componentele ce deservește la buna funcționare a aplicației.

În plus, aici sunt integrate partea de HTML , CSS și JavaScript , descrise mai sus în partea de Tehnologii utilizate.

3. Detalii de implementare

În acest capitol sunt prezentate detalii legate de implementarea aplicației, modul în care componentele comunică între ele și descrierea algoritmilor folosiți în stabilirea predicțiilor.

3.1. Datele primite de la Guvernul României și procesarea acestora

Între 2013 și 2016, Guvernul României a luat decizia de a înființa Departamentul pentru Servicii Online și de Design.

Astfel, departamentului îi revine îndatorirea de a crea un portal Open-data, dând acces informaticienilor să descarce date, să folosească seturi de date publice, create sau deținute de către administrația publică.

Portalul reprezintă aplicarea inițiativei europene Open Data – punerea la dispoziția publicului de date accesibile, reutilizabile și redistribuibile în mod liber, fără a ține cont de restricții de tipul drepturi de autor, patente sau alte mecanisme de control.

Inițiativa Guvernului României de a publica date deschise are ca scop facilitarea înțelegerii de către cetățeni a felului în care administrația funcționează și a considerentelor aflate la baza deciziilor acesteia. Având acces la aceste informații, cetățenii își vor putea exprima opiniile și sugestiile cu privire la politicile guvernamentale.

Astfel, am ajuns să mă gândesc să creez o aplicație web ce poate deservi populației. Cercetând prin portalul data.gov.ro , am realizat oportunitatea de a prezenta, a face o statistică și chiar o previziune asupra datelor meteorologice din România, utilizând date climatologice de la cele 23 de stații meteo esențiale.

Datele sunt structurate de-a lungul anilor, din 1961 până în 2016.

Datele descărcate vin sub forma unui document CSV. Acesta conținând pentru fiecare stație meteo în parte, o temperatură medie, o temperatură maximă și o temperatură minimă. În plus, conține și Longitudinea și Latitudinea stației meteo.

Fiecare stație meteo în parte are un cod, spre exemplu :

- 15015 este codul pentru stația Ocna Șugatag
- 15020 este codul pentru stația Botoșani
- 15090 este codul pentru stația Iași
- 15108 este codul pentru stația Ceahlău Toaca
- 15120 este codul pentru stația Cluj-Napoca
- 15150 este codul pentru stația Bacău
- 15170 este codul pentru stația Miercurea Ciuc
- 15200 este codul pentru stația Arad
- 15230 este codul pentru stația Deva
- 15260 este codul pentru stația Sibiu
- 15280 este codul pentru stația Vârfu Omu
- 15292 este codul pentru stația Caransebeș
- 15310 este codul pentru stația Galați
- 15335 este codul pentru stația Tulcea
- 15346 este codul pentru stația Râmnicu Vâlcea
- 15350 este codul pentru stația Buzău
- 15360 este codul pentru stația Sulina
- 15410 este codul pentru stația Drobeta Turnu Severin
- 15420 este codul pentru stația București Băneasa
- 15450 este codul pentru stația Craiova
- 15460 este codul pentru stația Călărași
- 15470 este codul pentru stația Roșiorii de Vede

- 15480 este codul pentru stația Constanța

Pentru a putea fi utilizate aceste date primite de la data.gov.ro, a fost nevoie de implementarea unor funcții ce deservește la manipularea și modificarea structurii lor. Astfel în funcție de necesitate am implementat diferiți algoritmi ce schimbă structura datelor.

Spre exemplu, pentru componenta Hartă, am creat un algoritm ce mă ajută să construiesc un JSON, pentru a putea afișa temperaturile pe harta oferita de Google Maps. Astfel am creat o structură nouă . Această structură este orientată către data calendaristică selectată de utilizator.

Cum am precizat în componenta hartă, utilizatorul, pentru a vizualiza datele climaterice , trebuie să selecteze în primă fază un an .

La acest pas, construiesc un obiect de tip JSON, ce va conține toate datele pentru anul respectiv.

Următorul pas, este atunci când utilizatorul selectează și luna dorită. Astfel din obiectul JSON obținut mai sus, extrag pentru fiecare stație meteorologică în parte, exact anul și luna, pe care utilizatorul dorește să le vizualizeze.

Ultimul pas, în afișarea datelor este alegerea zilei. Utilizatorul, selectând ziua dorită, pe harta se vor afișa datele climaterice.

Procesul pentru selectarea acestor date în funcție de an lună și zi, este unul destul de greoi și condiționat de multe aspecte cum ar fi: la selectarea anului, acesta poate fi bisect și trebuie luate mai multe componente din JSON-ul oferit de data.gov.ro. O alta condiție este legată de luna, aceasta poate cuprinde un număr de 30 de zile, 31 de zile, 28 de zile și în cazul în care anul este bisect, 29 de zile . Dar cu toate acestea am putut realiza o astfel de funcție ce mă ajută la buna funcționare a componentei.

Un alt exemplu de procesare a datelor este cel pentru componenta Statistică, aici datele sunt afișate pentru tabele și pentru grafice.

Aceste date diferă unele de altele prin structura lor.

Spre exemplu, în componenta Statistică, folosesc o structură a datelor ce utilizează variabila an ca punct cheie în structura obiectului. În plus pentru statistică, am făcut o medie a temperaturilor medii zilnice, astfel obținând un obiect de forma :

```
1. 1961:Object
    1. Statia:"Botosani"
    2. TemperaturaMedie:423
    3. TemperaturaMedieMaxima:690
    4. TemperaturaMedieMinima:163
    ...
```

Aici vom avea datele din 1961 până în 2016. Indexul acestui JSON va fi anul, și valorile acestui an vor fi Stația : Nume , Temperatura Medie : Valoare , Temperatura Medie Maximă : valoare și Temperatura Medie Minimă : valoare.

Pentru graficele din componenta Statistică, reprezentarea structurii este diferită, deoarece, așa este impus de către frameworkul amCharts.

Acesta mă obligă să folosesc un Array ¹¹de Obiecte structurat astfel :

```
1. Object
    1. Tmax:" 1.0"
    2. Tmed:" -0.1"
    3. Tmin:" -1.3"
    4. data:"1961/01/01"
    5. nume:"Ocna Sugatag"
    ...
```

Această mulțime de obiecte este utilizată în primul grafic din componenta statistică. Pe axa X este trecută fiecare valoare *data* a fiecărui obiect, iar pe axa Y este trecută fiecare valoare *Tmed* a fiecărui obiect. Astfel putem observa pe grafic la data x ce temperatura medie a fost înregistrată.

O altă abordare a datelor este folosită în componenta Predicție. Aici am hotărât ca datele să fie structurate astfel:

```
1. 0:Object
    1. an:"1961"
    2. value:525
```

Aici mulțimea de obiecte este formată abia după ce este selectat de utilizator, stația meteo de la care dorește să vadă predicțiile. Acest obiect este compus din două valori : *value*

¹¹ <https://en.wikipedia.org/wiki/Array>

ce reprezintă suma temperaturilor medii din valoarea *an*. Spre exemplu, la acest obiect în anul 1961, suma tuturor temperaturilor medii a fost de 525.

Am luat această decizie, de a crea predicții pe suma tuturor temperaturilor medii, deoarece astfel se observă cu o mai mare ușurință diferențele între predicții și realitate.

3.2. Algoritmi de predicție

Pentru realizarea lucrării de licență, am fost nevoit să apelez la următorii algoritmi de predicție:

- a) Metoda Modificării Procentuale
- b) Metoda Nivelării Exponențiale
- c) Trendul
- d) Mean Squared Error
- e) Metoda Brown
- f) Metoda Holt

a) Metoda Modificării Procentuale

Metoda modificării procentuale, este cuprinsă în metodele cantitative de previziune.

Pentru a putea folosi această metodă, este necesară îndeplinirea următoarelor criterii:

- Sunt detalii despre trecutul elementului și acesta pot fi folosit
- Sunt presupuneri ca în viitor elementul va continua să aibă o formă asemănătoare cu cea din trecut

Datele furnizate de către data.gov.ro, îndeplinesc aceste condiții, astfel le pot folosi pentru a face predicții.

Această metoda este una din cele mai simple metode de previziune, care implică folosirea unei formule matematice simple. Metoda urmărește să testeze schimbarea procentuală a elementului între perioade succesive de timp. Rezultatul final al acestei metode, este reprezentat de media indicilor ce reprezintă modificarea procentuală a elementului de la o perioada la alta.

Previziunea pentru următoarea perioadă se calculează astfel:

$$P_{t+1} = MMP_t * Y_t$$

Unde P_{t+1} reprezintă valoarea previzionată pentru perioada $t+1$. MMP_t este media modificării procentuale pentru t perioade, iar Y_t reprezintă valoarea observată din perioada t a variabilei care se previzionează.

În figura următoare vă voi arăta metoda Modificării Procentuale, aplicată ca o funcție în cod:

```
function MetodaModificariiProcentuale(data) {  
    var Yt = parseInt(data[55].value);  
    var Y0 = parseInt(data[0].value);  
    var MMP = (Yt - Y0) / 54;  
    var Y = (Y0 + 55 * MMP);  
    $("#predictieMMP").val(Y.toFixed(0))  
    return Y;  
}
```

Figura 5. Metoda Modificării Procentuale

În următoarea figură, vă voi prezenta Metoda Modificării Procentuale Mobile, apelată ca o funcție în cod:

```
function MetodaModificariiProcentualeMobile(data) {  
    var Yt = parseInt(data[55].value);  
    var Y0 = parseInt(data[0].value);  
    var MMP = (Yt - Y0) / 55;  
    var Y = (1 + MMP) * Yt;  
    return Y;  
}
```

Figura 6. Metoda Modificării procentuale Mobile

b) Metoda Nivelării Exponențiale

Metoda Nivelării Exponențiale este o metodă folosită pentru a obține o medie mobilă ce face referire la cea mai recentă soluție, aceasta însă având o greutate importantă în găsirea valorii previzionate.

Nivelarea exponențială se determină folosind formula:

$$Y_{t+1} = a Y_t + (1 - a)Y_t$$

Y_{t+1} reprezintă valoarea previzionată pentru următoarea perioadă

a este o valoare constantă, ce este cuprinsă între 0 și 1

Y_t reprezintă valoarea din perioada curentă

Valorile inițiale nivelate exponențial, sunt o medie a valorilor din ultimele perioade. După prima nivelare exponențială, aflăm previziunea și elementele nivelate din perioada curentă sunt apoi folosite pentru calcularea lui Y_t .

Dacă elementele arată o variație sezonieră, trebuie eliminată această sezonalitate și apoi apelată metoda nivelării exponențiale.

Alegerea valorii constante, de regulă, în practică aceasta este cuprinsă între 0,10 și 0,25.

Pentru o acuratețe mai bună a previziunilor, mai întâi se fac unele teste ce demonstrează cea mai bună alegere pentru constanta a .

În majoritatea cazurilor, constanta a este desemnată de un calculator, ce rulează o serie de teste.

La această metodă, valorile cele mai recente au o importanță mai mare în aflarea predicțiilor. Modelul nivelării exponențiale, este printre cea mai simplă tehnică de acest tip. Metoda nivelării exponențiale duble, face parte din metodele complexe derivate din această metodă.

În figura următoare este afișată funcția ce returnează valoarea obținută în urma aplicării metodei Nivelării Exponențiale:

```
function MetodaNivelariiexponentiale(data) {  
    var coef = 0.5;  
    var Yt = parseInt(data[55].value);  
    var Ytt = parseInt(data[53].value);  
    var Y = 0.5 * Yt + (1 - 0.5) * Ytt;  
    $("#predictieMME").val(Y.toFixed(0))  
}
```

Figura 7. Metoda Nivelării Exponențiale

c) Trendul

Trendul sau tendința se referă la o componentă generală, liniară sau de cele mai multe ori neliniară, care se schimbă cu trecerea timpului și nu se repetă.

Deplasarea graduală a seriei temporale este percepută ca o direcție, un trend.

Trendul se datorează de exemplu schimbărilor în tehnologie, schimbărilor în preferințele consumatorilor, sau ca și în cazul nostru, în schimbările climatice.

d) Mean Squared Error

Eroarea medie pătrată, sau MSE, a unui estimator (a unei proceduri pentru estimarea unei cantități nesupravegheate) măsoară media pătratelor erorilor sau a deviațiilor. Adică diferența dintre estimator și ceea ce este estimat.

MSE este o măsură a calității unui estimator, este mereu non-negativă, iar valorile apropiate de zero sunt cele mai bune.

Am introdus acest sistem de eroare, pentru a face un test, în urma căruia aş putea determina care este cea mai buna secvență de prezicere. Cea mai bună secvență de prezicere ar urma să aibă MSE-ul cel mai apropiat de zero.

e) Metoda Brown

Robert Goodell Brown (1923 – 2013) a fost renumit în domeniul prognozărilor, în special cu contribuții majore de muncă în ceea ce privește Metoda nivelării exponențiale. El a fost membru al Institutului Internațional de Predicție și director. A contribuit la domeniul predicției cu o serie de cărți și funcții.

Metoda Brown sau Metoda Nivelării Exponențiale Duble, este o metodă de prognoză similară cu Metoda Nivelării Exponențiale Simple, cu excepția faptului că variabila constantă este derivată din aceeași metodă a Nivelării Exponențiale Simple. Deoarece previziunile pot fi exprimate ca o funcție a constantelor, în această metodă existând două constante, metoda a primit numele de Metoda Nivelării Exponențiale Duble sau Metoda Brown.

Constanta de nivelare în acest model trebuie aleasă între 0,3 și 0,6.

Metoda Brown, poate prognoza mai multe perioade de timp în viitor.

Formula după care se calculează această metodă este :

$$P_{t+k} = \alpha_t + \beta_t * P_{k-1}$$

$$\alpha_t = 2 * P_t' - P_t'' \quad \text{și} \quad \beta_t = \frac{\alpha}{1-\alpha} * (P_t' - P_t'')$$

$$P_t' = \alpha * X_t + (1 - \alpha) * P_{t-1}' \quad \text{și} \quad P_t'' = \alpha * P_t' + (1 - \alpha) * P_{t-1}''$$

Utilizând această formulă am ajuns la crearea unei funcții ce îmi returnează o predicție pentru anul ales.


```

var xS1 = brown[i - 1].S1;
var xS11 = 0.7 * parseInt(data[i].value) + (1 - 0.7) * xS1;
var xS2 = brown[i - 1].S2;
var xS22 = 0.7 * xS11 + (1 - 0.7) * xS2;
brown[i] = {
  real: parseInt(data[i].value),
  alfa: 0.7,
  S1: 0.7 * parseInt(data[i].value) + (1 - 0.7) * xS1,
  S2: 0.7 * xS11 + (1 - 0.7) * xS2,
  forecast: 2 * xS11 - xS22,
  an: data[i].date
}

```

Figura 8. Metoda Brown

f) Metoda Holt

Charles C. Holt (21 mai 1961 – 13 decembrie 2010), a fost profesor emerit la Departamentul de Management de la McCombs School of Business de la Universitatea din Texas, Austin.

Este bine cunoscut pentru contribuțiile sale (și pentru contribuțiile elevului său, Peter Winters) la Metoda Nivelării Exponențiale.

Metoda nivelării exponențiale cu doi parametri sau Metoda Holt este mai flexibilă decât metoda Brown, deoarece determină nivelarea tendinței folosind un parametru diferit de cel al seriei dinamice inițiale.

Pentru a face o previziune este nevoie de doar trei valori istorice pentru a putea fi implementă (dar cu cât seriile de timp sunt mai mari cu atât acuratețea este mai mare).

Presupune utilizarea a trei coeficienți de nivelare dinamici α , β și γ .

Metoda este folosită pentru a afla trend-ul, iar pe baza acestuia nivelul variabilelor previzionate.

Seriile asociate metodei Holt au forma:

$$P_t = (\alpha + \beta_t) * T_t + \varepsilon_t$$

Unde α este o constantă mai mică decât 1 asociată nivelului inițial al seriei,

β este un indice asociat trend-ului seriei , iar ε_t este asociat erorilor (influențelor) aleatorii.

T_t este trend-ul seriei de valori istorice observate, calculat după formula:

$$T_t = \gamma * (P_{t-1} - P_{t-2}) + (1 - \gamma) * P_{t-1}$$

Previziunea valorilor, conform metodei Holt, presupune utilizarea formulei :

$$P_t = \alpha * \gamma_t + (1 - \alpha) * (P_{t-1} + T_t)$$

În cazul în care vom avea inclus și factorul de sezonality, se utilizează metoda Nivelării Exponențiale Sezoniere a lui Winters.

Utilizând această formula am ajuns la crearea unei funcții ce îmi returnează o predicție pentru anul ales.

```
var u_v = u - holt[i - 1].u;
var v = holt[i - 1].beta * u_v + (1 - holt[i - 1].beta) * holt[i - 1].v
var fortune = u + v;
holt[i] = {
  an: data[i].date,
  real: parseInt(data[i].value),
  alfa: 0.8,
  beta: 0.2,
  u: u,
  v: v,
  y: parseInt(data[i].value),
  forecast: fortune
}
```

Figura 9. Metoda Holt

4.Ghidul Aplicației

Aplicația web are o interfață intuitivă ce prezintă informațiile într-un mod curat și aerisit pentru a eficientiza modul de lucru.

Odată ce aplicația este încărcată, utilizatorul poate observa în partea din stânga un meniu. Fiecare element din meniu, direcționează utilizatorul către cele trei mari componente: Hartă, Statistică și Predicție.

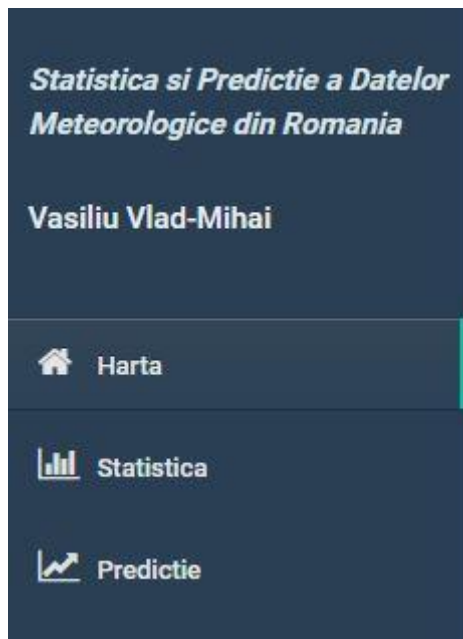


Figura 10. Menu

După cum se observă, în meniul avem titlul aplicației ”Analiză Statistică și Predictie a Datelor Climaterice din România”, urmat de numele autorului ” Vasiliu Vlad-Mihai”. În continuare putem alege una dintre componentele enumerate mai sus. By default, aplicația pornește cu componenta Hartă.

În dreapta meniului este afișată componenta Hartă. Această componentă este compusă din mai multe elemente: harta, cele trei selecții și un player.

Putem observa Harta României, urmată de o serie de trei selecții și un player.

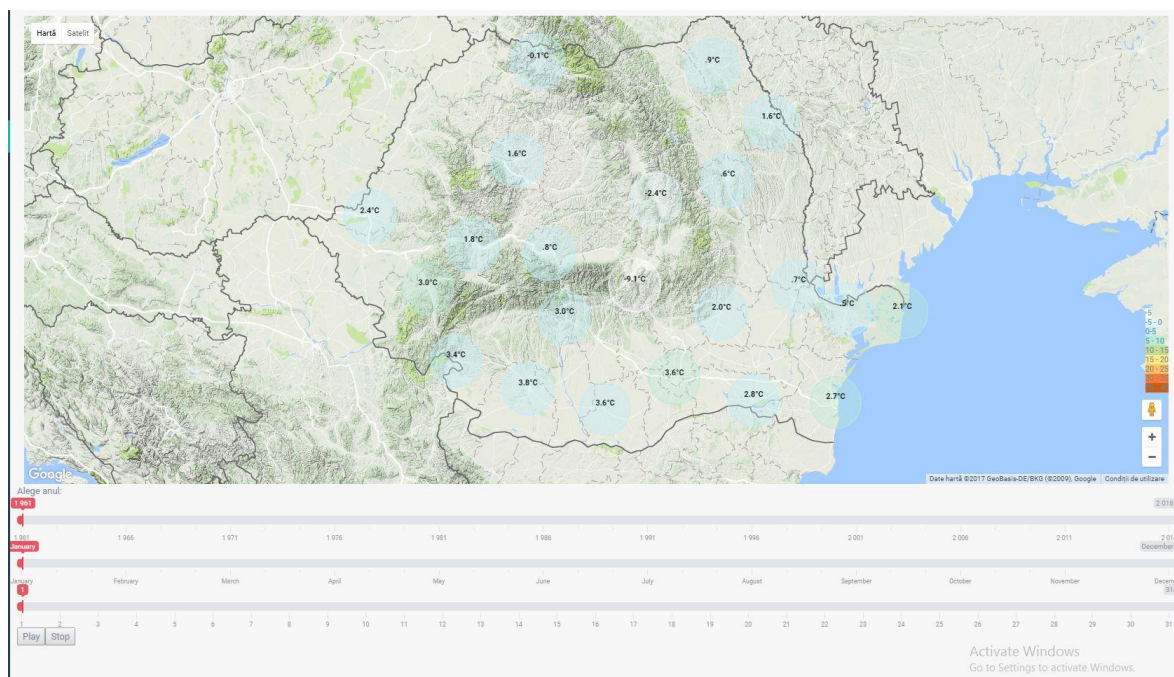


Figura 11. Harta României

Selectând anul, apoi luna și mai apoi ziua, putem observa pe hartă o serie de cercuri. Aceste cercuri sunt colorate în funcție de temperatura medie pe care o are. Fiecare cerc, are o latitudine și longitudine, astfel fiind posibilă afișarea pe hartă a acestor temperaturi.

Dând click pe oricare dintre cercuri putem observa o serie de detalii legate de acea stație meteo.



Modulul Play, Stop are rolul de a anima aplicația și a afișa utilizatorului din secundă în secundă, pornind de la ziua aleasă, toate temperaturile medii din ziua respectivă.

Selectând din meniu componenta Statistică, putem observa o serie de tabele și de grafice.

În prima parte a componentei statistică, observăm un selector. By default, acest selector este selectat pe opțiunea Ocna Șugatag. Mai jos, putem observa un tabel ce reprezintă temperaturile medii, temperaturile medii maxime și temperaturile medii minime, toate clasificate pe ani, de la 1961 până în 2016.

Mai jos, este reprezentat un grafic de tip coloane, ce reprezintă tabelul de mai sus .

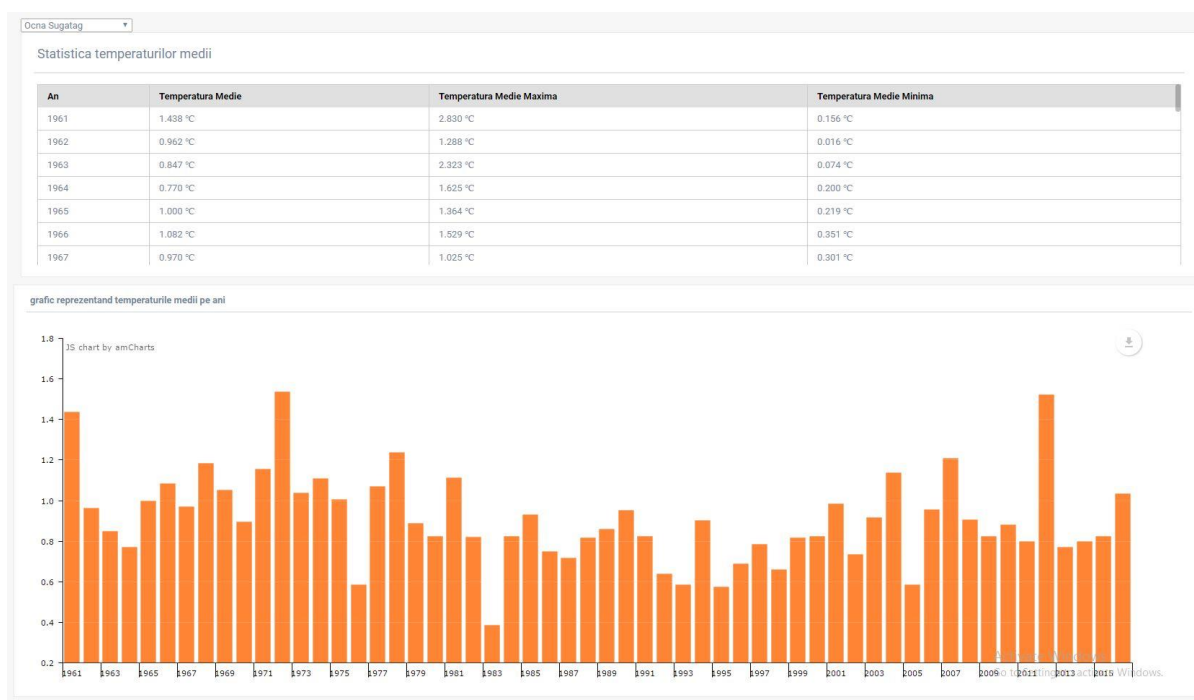


Figura 12. Tabel Statistică

În continuare sunt afișate doua grafice liniare.

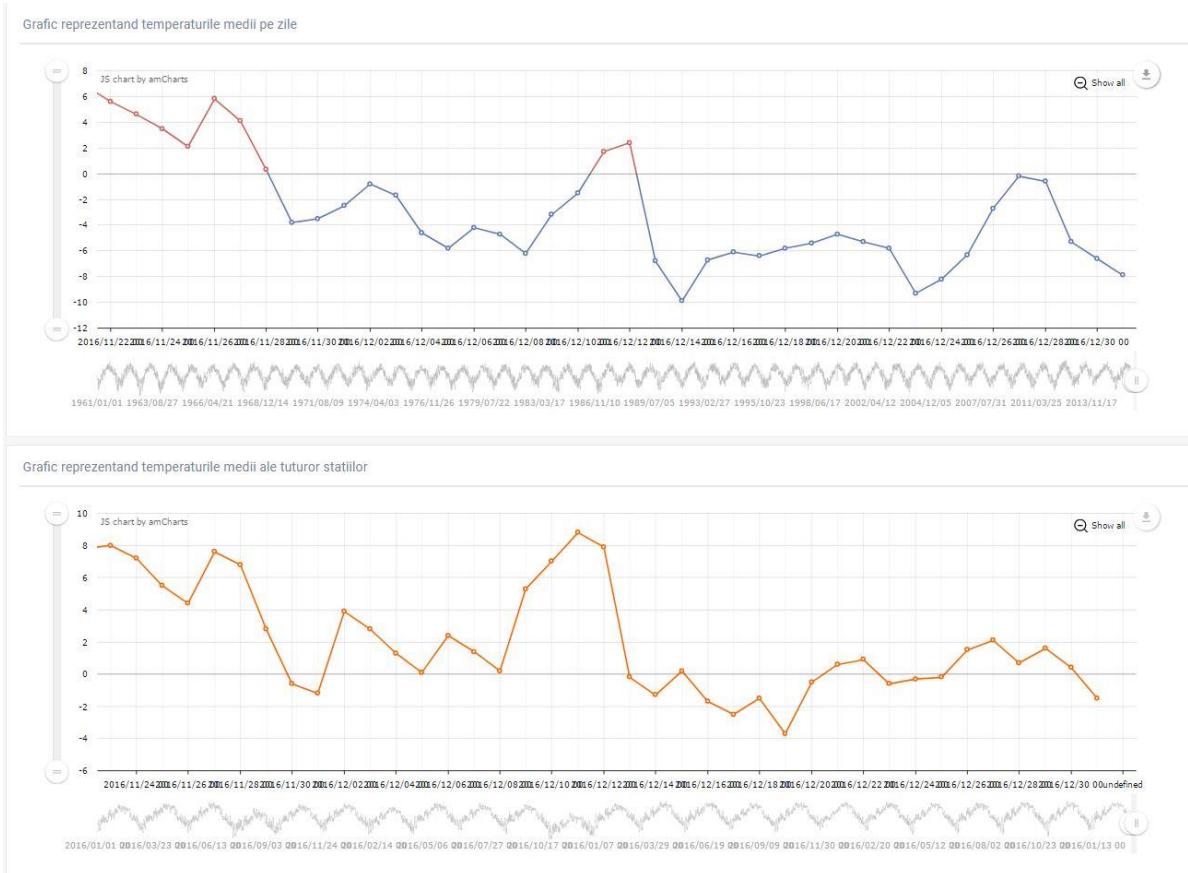


Figura 13. Grafice Statistică

Primul grafic, reprezintă temperaturile medii pe zile ale stației selectate anterior. Acest grafic, după cum putem observa , jos este reprezentat un mini grafic al valorilor acestei stații din 1961 până în 2016. O altă funcție importantă a acestor grafice este ca le poți descărca în diferite formate.

Al doilea grafic, este unul imens. Este compus din temperaturile medii ale tuturor stațiilor meteo din țară, din 1961 până în 2016.

Selectând din meniu componenta Predicție, putem observa ca este asemănătoare cu componenta Statistică.

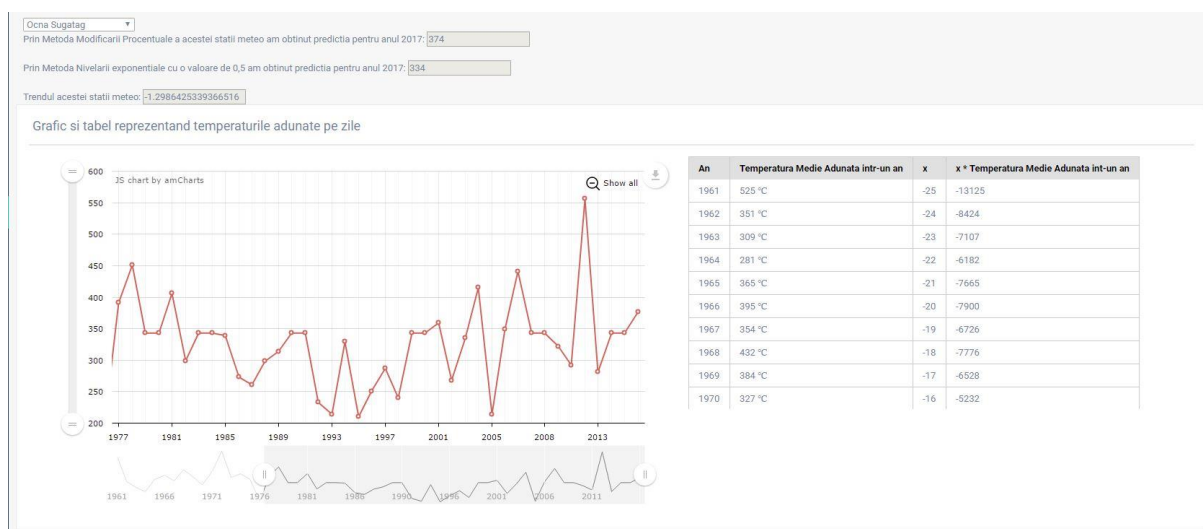


Figura 14. Predicție parte 1

Aici, putem selecta oricare din cele douăzeci și trei de stații meteo. By default, stația Ocna Șugatag este selectată.

Sub această selecție putem observa deja trei metode de predicție aplicate.

Prima metodă, este Metoda Modificării Procentuale, aceasta afișând pentru anul 2017 o predicție de 374. Acest 374 reprezintă temperatura adunată din fiecare zi a anului 2017, astfel o temperatură medie pentru 2017 la Ocna Șugatag ar fi 1,024 grade Celsius.

A doua metodă prezentă, este Metoda Nivelării Exponențiale, cu o constantă de 0,5. Prin această metodă se preconizează pentru 2017 la Ocna Șugatag o temperatură medie adunată de 334. Astfel calculând, vom avea o temperatură medie de 0,91 grade Celsius.

Trendul, pentru această stație meteo este calculat cu ajutorul tabelului din dreapta și a formulei enunțate mai sus în document. Pentru această stație avem un trend de -1,29, adică o temperatură ce scade pe an cu -0,0035 de grade Celsius.

Tabel cu Predictiile dupa Metoda Brown						
An	Alfa	S1	S2	Suma temperaturilor Reala	Suma temperaturilor Prezisa	
1961	0.7	525	525	525	525	
1962	0.7	403.20000000000005	439.74	351	366.6600000000001	
1963	0.7	337.26	368.004	309	306.51599999999996	
1964	0.7	297.878	318.9158	281	276.8402	
1965	0.7	344.86339999999996	337.07912	365	352.64767999999999	
1966	0.7	379.95902	367.09505	395	392.82299	
1967	0.7	361.787706	363.37990920000004	354	360.1955028	
1968	0.7	410.9363118	396.66939102000003	432	425.20323257999996	
1969	0.7	392.08089354	393.457442784	384	390.70434429599993	
1970	0.7	346.524268062	360.6042204786	327	332.44431564539997	

Tabel cu Predictiile dupa Metoda Holt						
An	Alfa	Beta	U	V	Suma temperaturilor Reala	Suma temperaturilor Prezisa
1961	0.8	0.2	525	0	525	0
1962	0.8	0.2	385.79999999999995	-27.84000000000001	351	357.95999999999999
1963	0.8	0.2	318.792	-35.67360000000001	309	283.11839999999995
1964	0.8	0.2	281.42368	-36.012544000000005	281	245.411136
1965	0.8	0.2	341.0822272	-16.878325760000006	365	324.20390144
1966	0.8	0.2	380.84078028799996	-5.550949990400009	395	375.28983029759996
1967	0.8	0.2	358.25796605951996	-8.957322838016008	354	349.30064322150395
1968	0.8	0.2	415.46012864430077	4.274574246543356	432	419.7347028908441
1969	0.8	0.2	391.1469405781688	-1.4429782159917042	384	389.7039623621771
1970	0.8	0.2	339.54079247243544	-11.47561219994004	327	328.0651802784954

Figura 15. Tabele Predictie

În continuare se observă doua tabele, ce reprezintă doua metode diferite de predicție. Una dintre ele este Metoda Brown sau Metoda Nivelării Exponențiale duble și o altă metoda, este Metoda Holt sau Metoda Nivelării Exponențiale cu 2 parametri.

În aceste doua tabele, am evidențiat pe ani, temperatura reală și temperatura preconizată de fiecare dintre aceste metode de predicție. În plus am afișat si variabilele ce ajută la determinarea predicției.

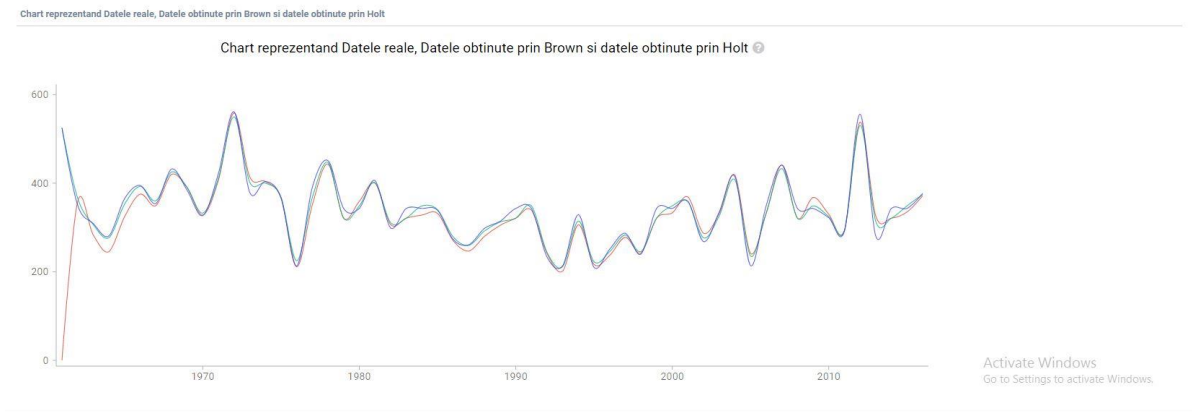


Figura 16. Grafic Real, Brown și Holt

Ultimul grafic al acestei aplicații este unul ce compară temperaturile reale adunate pe un an , cu cei doi algoritmi de predicție Brown și Holt.

Concluzii

Această aplicație web a fost creată pentru a afișa, a crea o statistică și pentru a crea o predicție asupra datelor climaterice ale României din 1961 până în 2016.

Poate prin această aplicație voi da un imbold programatorilor să ceară și să utilizeze datele guvernamentale. Creând astfel o aplicație ce poate deservi oricui, în orice scop.

Îmbunătățiri

Aplicația web, datorită tuturor proceselor de manipulare a obiectelor JSON primite de la data.gov.ro, este un pic greoaie. Astfel ar fi necesar un server ce deservește aplicației web datele necesare, iar datele să fie stocate într-o bază de date.

Designul aplicației poate fi îmbunătățit, să fie cât mai util utilizatorului. Printr-o logare, putem crea o bază de date cu utilizatorii aplicației, aceștia putând avea preferințe în materie de ce să fie afișat.

În plus, aplicația nu are încă un design aparte pentru mobile.

Limitări

Aplicația web, necesită o foarte bună conexiune la internet, deoarece cum am mai precizat, toate datele sunt ținute în obiecte de tip JSON, stocate în browser, astfel pentru cineva cu o conexiune mai rea, aplicația s-ar încărca mult mai greu.

Acest proiect a reprezentat o provocare plăcută pentru mine, deoarece pentru prima oară, am avut acces la datele guvernamentale și am reușit să le pun în valoare prin aceasta aplicație. Sper că nu voi fi singurul ce folosește acele date, deoarece se pot face destule predicții și statistici pe baza lor.

Bibliografie

- Modele econometrice Volumul I editia a doua, Prof.univ.dr. Ovidiu Tanasoiu, Lect.univ.dr Andreea Iluzia Iacob
- Statistică generală, Prof.univ.dr Alexandru Iscaic-Maniu, Prof.univ.dr. Constantin Mitrut, Prof.univ.dr. Virgil Voineagu
- Previziune macroeconomică , Prof.univ.dr Valentin Nicolae, prof.univ.dr Dumitrache Caracota, Prof.univ.dr. Daniela Luminita Constantin, Prof.univ.dr Cornelia Parlog, Conf.univ.dr. Ilie Gradinaru, Lect.univ.dr Vasilica Slavescu, Lect.univ.dr Valerian Tobultoc
- <https://en.wikipedia.org/wiki/JQuery>
- <https://en.wikipedia.org/wiki/JavaScript>
- https://en.wikipedia.org/wiki/Cascading_Style_Sheets
- https://ro.wikipedia.org/wiki/HyperText_Markup_Language
- <http://getbootstrap.com/about/>
- <http://riffwiki.com/AmCharts>
- <https://www.amcharts.com/javascript-charts/>
- https://en.wikipedia.org/wiki/Google_APIs
- <https://biblioteca.regielive.ro/proiecte/marketing/metoda-modificarii-procentuale-previziuni-de-marketing-62696.html>
- http://www.scribub.com/management/marketing/PREVIZIUNE14649.php#_Toc166381605