Departamentul Automatică şi Informatică Industrială
Facultatea Automatică şi Calculatoare
Universitatea POLITEHNICA din Bucureşti

LUCRARE DE DIPLOMĂ

Aplicație web pentru analiza datelor din domeniul economic

Coordonator
Prof.dr.ing. Mihnea-Alexandru MOISESCU
As.drd.ing. Miruna-Elena ILIUȚĂ

Absolvent
Anca-Maria CIUCIU

2023

Cuprins

1. Introducere

1.1 Context

Atingerea obiectivelor financiare pe termen lung are la bază gestionarea atentă a bugetului personal, ce permite o înțelegere mai clară a fluxului de numerar. Înregistrarea și monitorizarea plăților și veniturilor ne oferă un nivel de control asupra banilor, planificând și alocând resursele în mod strategic. Alocarea conform priorităților stabilite a resurselor reprezintă un progres spre direcția stabilității financiare, ajutând la evitarea datoriilor și reducerea stresului.

Investițiile responsabile și profitabile se materializează în momentul în care se ajunge la o stabilitate financiară solidă, care garantează un fond de urgență. Printre activele financiare care au atras atenția investitorilor în ultimul timp se enumeră: criptomonedele și acțiunile tehnologice, care pot prezenta riscuri, deoarece valoarea acestora poate varia semnificativ într-o perioadă foarte scurtă de timp. Volatilitatea acestora este, de cele mai multe ori, puternic influențată de știrile și postările din mediul online, ce au un impact semnificativ asupra comportamentului investitorilor și a piețelor financiare.

Analiza datelor din domeniul economic joacă un rol important în obținerea unei înțelegeri mai bune a stării generale a indicatorilor macroeconomici relevanți și a tendințelor pieței, oferind o perspectivă clară asupra riscurilor și oportunităților din mediul investițional. Astfel, aplicația propusă, Twinvest, oferă posibilitatea de urmărire a bugetului personal, având la bază gestionarea veniturilor și achizițiilor, cât și un mecanism ce furnizează informații relevante și actualizate prin știri personalizate și postări ale investitorilor de pe Twitter, selectate în funcție de domeniile de interes ale utilizatorului.

În plus, algoritmii implementați de procesare a limbajului natural și analiză a sentimentelor, aplicați știrilor din Google News oferă o predicție a tendinței de creștere sau scădere a acțiunilor și criptomonedelor pentru o investiție informată.

1.2 Motivație

Criptomonedele sunt definite ca orice activ digital bazat pe tehnologia blockhain. [1] Aceste active au înregistrat o creștere semnificativă în ultimii ani, astfel că, potrivit unei analize realizate de CoinMarketCap [2], valoarea pe piața globală a cripto-activelor este de 1,13 trilioane de dolari, rata de deținere a acestora fiind în medie de 4.2% cu peste 420 de milioane de utilizatori de criptomonede la nivel mondial.

Pe piața criptomonedelor, Twitter reprezintă una dintre principalele resurse de social media utilizată de emitenții de monede pentru a comunica cu investitorii lor. De asemenea,

studiile anterioare au arătat că sentimentele exprimate pe Twitter pot influența prețurile criptomonedelor si acțiunilor precum și volumele de tranzacționare. O știre recentă arată că Elon Musk, deținătorul Twitter, a fost implicat într-un proces judiciar, fiind acuzat de faptul că a manipulat valoarea criptomonedei Dodgecoin prin postările sale pe rețeaua de socializare. [3]

Prin urmare, prețul unui instrument financiar poate crește sau se poate prăbuși brusc ca urmare a unui singur tweet al unei persoane publice cunoscute. Acest lucru se datorează forței de influențare pe care o exercită acestea în rândul urmăritorilor lor.

Aplicația calculează predicții bazate pe ultimele știri preluate din una dintre cele mai accesate surse, Google News. Utilizatorul are posibilitate de analiză atât a propriului buget cât și a prețurilor de pe piață a activelor financiare în timp real pentru a lua o decizie privind următoarea investiție în mod informat.

1.3 Obiective

Obiectivul principal stabilit este determinarea cu o precizie cât mai mare a sentimentului transmis de știrile de ultimă actualitate, acestea fiind împărțite în trei categorii: pozitive, negative și neutre din punct de vedere economic. Bineînțeles că modelul ce are la bază algoritmii de învățare automată și prelucrare a limbajului natural va fi aplicat doar pe titlurile știrilor extrase la un anumit interval de timp de pe Google News, raționamentul fiind acela că titlul are cel mai puternic impact asupra investitorului, de multe ori, acesta fiind chiar singura sursă din care oamenii își extrag informația.

Tocmai din acest punct de vedere, îmi propun ca prin intermediul acestei aplicații web, să folosesc principiile inteligenței artificiale și învățarea automată pentru a oferi investitorului un context, o perspectivă asupra evoluției pieței, prezentând informații relevante, astfel încât acesta să poată lua decizii singur cu privire la modul de investire a banilor, precum și posibilitatea monitorizării circulației fondurilor personale.

1.4 Conținutul lucrării

Structura proiectului de diplomă are la bază 6 capitole relevante ce cuprind: descrierea domeniului ales, explicarea conceptelor de funcționare a tehnologiilor folosite pentru dezvoltare, prezentarea funcționalităților aplicației, precum și concluziile și contribuțiile personale.

Primul capitol are rolul de a oferi un context general referitor la domeniul economic și necesitatea de avea acces la știri relevante și actualizate, ce pot avea un impact major asupra investițiilor. De asemenea, se setează obiectivele propuse ce țin de performanța aplicației.

Capitolul 2 conține formularea generală a problemei precum și studiul aprofundat al realizărilor similare în domeniu, cu scopul de a stabili funcționalitățile pe care aplicația propusă le va avea, în funcție de necesitățile utilizatorilor. Astfel, se va realiza o comparație între trei aplicații selectate ce realizează analize în domeniul economic, integrând algoritmi de învățare automată.

În cel de-al treilea capitol vor fi explicate motivele alegerii tehnologiilor cu care aplicația web a fost implementată. Această secțiune prezintă analize comparative cu alte tehnologii folosite pentru realizarea unor aplicații asemănătoare precum și raționamentul urmat pentru antrenarea modelului de învățare automată, algoritmi de procesare a setului de date, arhitectura rețelei neuronale și elemente de procesare a limbajului natural.

Capitolul 4 începe prin prezentarea arhitecturii stratificate și a modului în care elementele de inteligență artificială au fost integrate în cadrul aplicației web. De asemenea, acest capitol are rolul de a prezenta modul de implementare a fiecărei componente: baza de date, partea de backend, frontend, extragerea știrilor și antrenarea rețelei neuronale. Tot aici se analizează setul de date ales și rezultatele acurateții modelului de învățare automată și a metricilor acestuia.

Capitolul 5 cuprinde studii de caz referitoare la modalitatea funcționării aplicației și folosirii tuturor funcționalităților din perspectiva utilizatorului. Astfel, se prezintă scenariile posibile, felul în care se pot seta preferințele de investiție pentru redarea conținutului personalizat, monitorizarea bugetului prin adăugarea manuală a veniturilor și plăților precum și vizualizarea știrilor și a tendințelor de creștere sau scădere a activelor financiare.

Ultimul capitol este dedicat concluziilor rezultate în urma procesului de dezvoltare a proiectului de diplomă, făcând referire și la o serie de dezvoltări ulterioare ce ar putea spori performanțele aplicației.

2. Descrierea problemei abordate

2.1 Formularea problemei

De-a lungul timpului, banii și investițiile au evoluat într-un mod dinamic și fascinant, care a fost marcat de diverse schimbări specifice fiecărei perioade în ceea ce privește modalitatea prin care omul gestionează și valorifică resursele financiare. Urmând tendința firească de dezvoltare, progresare, oamenii au căutat întotdeauna modalități de a crește averea prin diverse metode.

De asemenea, factori precum: incertitudinea viitorului, atingerea obiectivelor personale sau stabilitatea financiară și independența au determinat individul la procesul de economisire a resurselor financiare, aceasta fiind o caracteristică fundamentală a comportamentului responsabil. Astfel, problema care se conturează este, în mod evident, planificarea bugetului personal, monitorizându-se veniturile și plățile, fapt ce oferă oportunitatea de a investi banii economisiți, generând profit pe termen lung.

Trecerea impresionantă la investițiile în criptomonede, precum și digitalizarea procesului de vânzare-cumpărare a fost posibilă prin intermediul tehnologiei blockchain. Blockchain-ul este un registru digital, distribuit, o tehnologie revoluționară, ce concentrează o bază de date distribuită și criptografică ce păstrează un istoric al tranzacțiilor în mod sigur.

Valoarea criptomonedelor este afectată de oferta și cererea pieței, precum orice alt tip de valută. Investițiile sunt o parte esențială a strategiei financiare personale, acestea aducând după sine o serie de oportunități, dar și provocări ce se pot transforma în riscuri dacă problematica nu este analizată în mod atent. Este important de menționat că valoarea criptomonedelor poate fluctua în mod imprevizibil în funcție de o varietate de factori precum: reglementări guvernamentale, evenimente geopolitice și sentimentul general al pieței, care este dictat în general de postările din mediul online.

Un alt instrument economic analizat, ce a luat amploare în ultimii ani este reprezentat de acțiunile companiilor listate pe piața de valori. Aceste acțiuni sunt emise de companii publice și permit investitorilor să devină proprietari parțiali ai companiei și să profite implicit de potențialul său de creștere. La fel ca în cazul criptomonedelor, prețul acțiunilor poate fi influențat de mai mulți factori, precum: performanța companiei, perspectivele de creștere cât și de anunțurile și rapoartele financiare – informații ce vor sta la baza antrenării modelului de învățare automată pentru a putea fi capabil să recunoască starea generală emisă de o știre.

Prin urmare, având în vedere toate aceste aspecte, este important de menționat că evoluția prețului acestor active digitale este influențată de o serie de factori, mediul comercializării acestora fiind unul instabil, al cărui comportament poate fi dificil de prezis, deoarece poate avea mai multe stări posibile și pot fi inluențate de intrări și evenimente externe, precum un automat finit nedeterminist. Investitorii trebuie să abordeze atent și

informat această piață tehnologică și să facă propriile cercetări, în concordanță cu analizarea bugetului personal, înainte de a lua decizii de investiții pentru a minimiza apariția riscurilor.

2.2 Studiu asupra realizărilor similare din domeniu

Aplicațiile de planificare a bugetului personal au rolul de a ajuta utilizatorii să își gestioneze veniturile și achizițiile, oferind funcționalități precum înregistrarea, crearea de bugete personalizate, urmărirea economiilor, generarea de rapoarte și statistici precum și stabilirea unor praguri de cheltuială pentru a primi avertizări în cazul depășirii acestuia. Totodată, în ceea ce privește investițiile, din ce în ce mai multe aplicații integrează algoritmi de învățare automată pentru realizarea predicțiilor cu scopul cumpărării sau vinderii automate de acțiuni.

În continuare, se vor prezenta câteva dintre aplicațiile cele mai cunoscute ce se axează pe monitorizarea bugetului și a investițiilor.

2.2.1 Empower

Empower este una dintre cele mai cunoscute aplicații web de gestionare a bugetului, ce integrează și analiza pieței financiare. Utilizatorii își pot crea și urmări bugete personalizate pentru diverse categorii de plăți, redând grafice și diagrame interactive pentru a vizualiza modul cheltuirii banilor. Totodată, aplicația este orientată și în zona investițiilor, astfel încât, se poate monitoriza performanța acestora, primind recomandări personalizate.

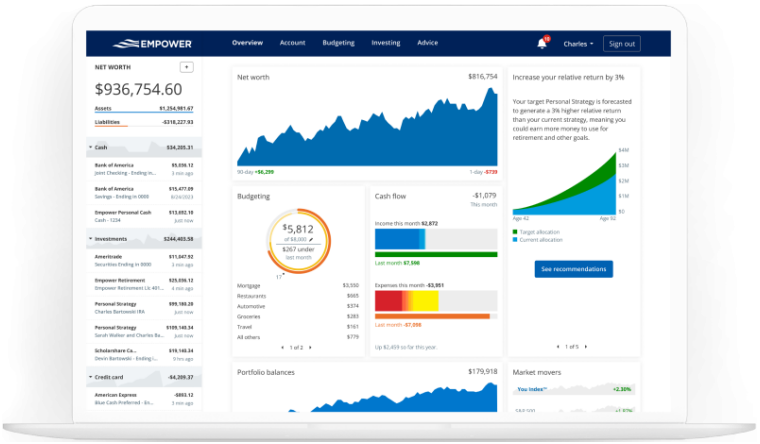

Fig. 1. Captură de ecran a interfeței Empower

Deși Empower nu dispune de suport de oferire a unor predicții în materie de criptomonede, aplicația oferă informații și analize în timp real ale pieței financiare, precum și evoluția stocurilor, valutelor și altor active. Utilizatorii au acces la știri și informații de ultimă actualitate de pe piața financiară pentru a ține pasul cu ultimele tendințe. [4]

Un avantaj major pe care îl oferă acest instrument este integrarea sa și în alte aplicații existente cu același scop, cum ar fi Mint, fapt ce permite utilizatorilor să combine funcționalitățile ambelor aplicații. În ceea ce privește portabilitatea, Empower este disponibil atât ca aplicație mobile, pentru Android si iOS, cât și ca aplicație web.

2.2.2 AlphaSense

AlphaSense este o platformă de căutare și analiză avansată a datelor financiare și comerciale, care permite utilizatorilor să acceseze o gamă largă de conținut și date relevante. Tehnicile de procesare a limbajului natural oferă căutarea și filtrarea rapidă și precisă a rapoartelor de cercetare, prezentărilor și alte documente relevante.

Algoritmii de inteligență artificială folosiți de această aplicație evidențiază automat informațiile importante, de ultimă actualitate, din conținutul căutat. Totodată, utilizatorii își pot crea alerte personalizate cu scopul de a fi notificați în legătură cu domeniile lor de interes. [5] Cu toate acestea, analizând părerile utilizatorilor activi ai acestei aplicații, pare că, un dezavantaj ar consta în faptul că interfața poate fi una complexă. Din cauza funcționalităților avansate, utilizarea inițială a aplicației, pentru persoanele care nu au prezentat până în acel moment vreun interes pentru investiții financiare, necesită o perioadă de familiarizare pentru a profita de toate beneficiile aplicației.

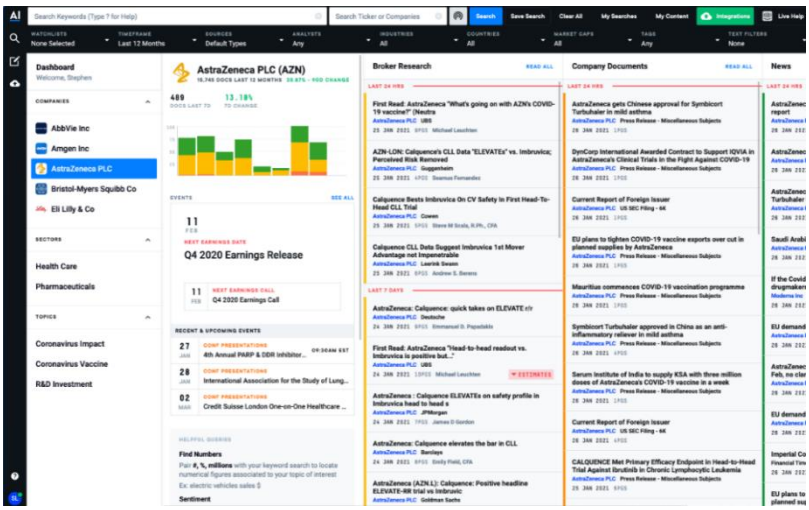

Fig. 2. Captură de ecran a interfeței AlphaSense

2.2.3 Kavout

Kavout are la bază modelul de inteligență artificială „Kai”, ce analizează milioane de statistici, rapoarte și cotări la bursă. Aceasta se folosește de știri, bloguri și rețele de socializare pentru a genera previziuni privind performanța piețelor financiare.

Această aplicație analizează acțiunile și furnizează informații detaliate despre performanța acestora, tendințe de preț, indicatori tehnici, precum și alți factori relevanți pentru luarea deciziilor în mod informat. Cu ajutorul algoritmilor de învățare automată și prelucrare a limbajului natural, Kavout genereză predicții, ce oferă o perspectivă suplimentară asupra potențialelor tendințe de preț și așteptărilor de pe piață. [6]

O funcționalitate importantă ar fi capacitatea de a face investiții automate pentru utilizatori în funcție de interpretările pe care acesta le face asupra datelor extrase în timp real, identificând în permanență oportunitățile de tranzacționare.

Fig. 3. Captură de ecran a interfeței Kavout

2.3 Compararea soluțiilor existente

Există o varietate de aplicații care ne pot ajuta să ne gestionăm bugetul, să urmărim evoluțiile piețelor financiare și sa luăm decizii informate. În ceea ce privește domeniul economic, nu există o aplicație ideală pentru urmărirea traficului banilor, dar există aplicații care se focusează în mod diferit în funcție de nevoile investitorului.

Fiecare persoană are anumite priorități și cerințe specifice, astfel că, obiectivul este oferirea posibilității de a alege informat, punând la dispoziție o interfață ușor de utilizat prin care se pot gestiona ușor tranzacțiile, profitând totodată și de performanțele inteligenței artificiale.

În acest context, trei aplicații notabile sunt Empower, AlphaSense și Kavout, analizându-le din punct de vedere al funcționalităților oferite și felul în care se disting în tabelul de mai jos.

Nr. Crt.	Funcționalitate	Empower	AlphaSense	Kavout
1	Suport web	✓	✓	✓
2	Suport telefon	✓	✓	✓
3	Autentificare cu platforme terțe	✓	✗	✗
4	Monitorizarea veniturilor și plăților	✓	✗	✗
5	Statistici	✓	✓	✓
6	Recomandări personalizate de investiții	✓	✗	✓
7	Consultanță financiară	✓	✗	✗
8	Sistem inteligent de investire automată	✗	✓	✓
9	Analizarea postărilor de pe rețele de socializare	✗	✓	✓
10	Suport pentru acțiuni la bursă	✓	✓	✓
11	Suport pentru criptomonede	✓	✓	✓
12	Suport pentru oferirea de predicții ale evoluției prețurilor pentru activele financiare	✗	✓	✓
13	Date în timp real ale activelor financiare urmărite	✓	✓	✓
14	Gratuitate	✗	✗	✓
15	Alerte și notificări	✓	✓	✓

Tabel 1. Compararea soluțiilor existente pe piață

În final, succesul unei aplicații constă în a oferi o experiență personalizată și utilă fiecărui utilizator, abordând specificațiile și cerințele individuale ale acestora.

2.3.1 Cerințe funcționale

Scopul diagramei de utilizare este de a prezenta modul în care utilizatorul interacționează cu aplicația, precum și fluxul de acțiuni pe care le desfășoară în cadrul ei.

Astfel, diagrama de mai jos reliefează arhitectura aplicației propuse și implicit, funcționalitățile acesteia.

Fig. 4. Diagrama cazurilor de utilizare

Astfel, pe baza acestei arhitecturi, se pot descrie funcționalitățile aplicației propuse.

1. **Înregistrarea utilizatorului**

Pentru a putea accesa aplicația, se impune creare unui cont ce presupune introducerea unui nume de utilizator, a unei adrese de e-mail valide și a unei parole formate din minim 8 caractere. Odată ce contul a fost creat și înregistrat în sistem, utilizatorul poate accesa aplicația. Acesta va fi avertizat prin intermediul unor erori sugestive în cazul în care adresa de e-mail nu este una validă sau parola nu îndeplinește criteriul menționat.

2. **Autentificarea în aplicație**

În momentul în care utilizatorul dispune deja de un cont valid, acesta se poate autentifica folosind adresa de e-mail și parola setată în momentul creării contului. În cazul

în care acestea nu corespund cu informațiile din baza de date, utilizatorul va fi atenționat și ulterior îndrumat pentru a-și schimba parola.

3. **Resetarea parolei**

Acestă funcționalitate oferă utilizatorului o opțiune convenabilă și sigură în cazul în care întâmpină probleme cu accesarea contului. Astfel, se poate plasa o solicitare de resetare a parolei ce va invoca trimiterea unui e-mail cu instrucțiuni în acest sens. Noua parolă setată va trebui să îndeplinească aceleași criterii ca cele de la înregistrare.

4. **Introducerea manuală a veniturilor și plăților**

Introducerea detaliilor despre modul cheltuirii banilor oferă utilizatorului posibilitatea de înregistra și monitoriza în mod eficient fluxul de numerar, conturându-se o perspectivă completă și actualizată a situației financiare. Principiul de adăugare pentru cele două activități economice este aproape identic. Astfel, pentru fiecare tranzacție se completează un formular scurt ce cuprinde denumirea, suma, data, categoria și o scurtă descriere.

5. **Ștergerea veniturilor sau a plăților**

Funcționalitatea de eliminare din cont a unor elemente din istoric este importantă în momentul în care acestea nu mai sunt relevante sau au fost introduse eronat. Astfel, se asigură flexibilitate și control asupra datelor pe care utilizatorul dorește să le păstreze în aplicație.

6. **Vizualizarea statisticilor și a istoricului tranzacțiilor**

În ceea ce privește vizualizarea statisticilor, utilizatorul poate avea acces la grafice ce evidențiază sumele de bani cheltuite lunar, precum și un raport de alocare a resurselor în funcție de categorie sau alte detalii cum ar fi suma maximă alocată pe un anumit obiect, excedentul bugetar sau progresul în ceea ce privește îndeplinirea obiectivelor financiare.

7. **Setarea unui obiectiv pentru planificarea bugetului**

O planificare atentă a bugetului presupune setarea unui obiectiv, specificând suma de bani pe care investitorul dorește să o economisească și termenul limită până la care intenționează să își îndeplinească anumite scopuri specifice.

8. **Setarea unei limite de cheltuire a bugetului**

Pentru a economisi în mod conștient, această funcție permite introducerea unei limite de cheltuire pe o anumită perioadă de timp pentru diverse categorii. Astfel, acest eveniment va declanșa avertismente și notificări în momentul în care această limită este aproape depășită pentru a gestiona mai eficient bugetul.

9. **Schimbarea datelor profilului**

Parametri care se pot modifica din contul unui utilizator sunt: parola și preferințele sau domeniile de interes urmărite în materie de active financiare cu scopul de a primi informații personalizate.

10. Accesare postări de pe Twitter

Prin integrarea API-ului Twitter, este posibilă accesarea postărilor relevante referitoare la criptomonedele sau acțiunile la bursă urmărite. Astfel, utilizatorul are ocazia de a analiza punctul de vedere al persoanelor din domeniul economic sau al persoanelor cu influență ce pot influența piața.

11. Accesare știri de pe Google News

Utilizând un *web scraper*, se vor colecta toate știrile de ultimă actualitate de pe pagina web Google News, care vor fi ulterior filtrate în funcție de preferințele fiecărui utilizator. Astfel, se permite accesul în timp real la evenimentele și noutățile care pot avea impact asupra piețelor financiare, identificându-se oportunități și riscuri, ajustând astfel strategiile de investiții.

12. Generarea predicțiilor pentru activele financiare urmărite

Prin utilizarea unor algoritmi de analiză și inteligență artificială, aplicația oferă predicții și estimări cu privire la evoluția prețurilor, tendințele pieței și performanța activelor financiare. Această funcționalitate are la bază un model de învățare automată ce a fost antrenat pe un set de date ce conține titluri are știrilor economice și evaluarea sentimentului pe care acestea îl transmit: pozitiv, negativ sau neutru. Acest model va primi ca argument știrile de pe Google News din ultimele zile pe care le va analiza și astfel se va calcula un scor pe baza căruia acest sistem va decide dacă trendul este unul crescător sau descrescător.

13. Suport pentru primirea de notificări și avertizări

Cu ajutorul notificărilor și avertizărilor, utilizatorii pot fi informați în timp real despre evenimente și actualizări importante, astfel încât să își administreze în mod înțelept situația financiară. Astfel, în urma setării unei limite de cheltuire sau a unui obiectiv, acesta va primi un pop-up cu scop informativ pentru a-i reaminti planurile inițiale.

14. Vizualizarea prețurilor activelor financiare în timp real

Această funcționalitate redă fluctuațiile prețului activelor financiare urmărite de utilizator. Acesta este redat prin intermediul unui procent pozitiv sau negativ în funcție de cum a evoluat în ultimele 24 de ore. Datele sunt extrase prin intermediul API-ului RealStonks de pe platforma RapidAPI.

15. Conținut personalizat bazat pe preferințele utilizatorului

Această caracteristică presupune selectarea și afișarea știrilor și a postărilor de pe Twitter în funcție de instrumentele economice urmărite. Tot din această zonă face parte și vizualizarea în timp real a evoluției pieței. Utilizatorul își poate alege ce criptomonede sau acțiunile urmărite în funcțiile de investițiile sale pentru a rămâne la curent cu ultimele informații despre acestea.

16. Deconectarea din aplicație

Prin acestă opțiune, utilizatorii se pot deconecta în mod sigur și comod din contul lor.

2.3.2 Cerințe non-funcționale

1. **Performanță optimă**

Aplicația trebuie să funcționeze rapid și eficient, fără înârzieri. Aceasta trebuie să poată gestiona un volum mare de informații și să raspundă cererilor web. Fluiditatea aplicației presupune un timp de navigare mic, dar și un răspuns rapid la cererile realizate prin API-uri.

2. **Scalabilitate**

Aplicația web trebuie să se adapteze în mod eficient la creșterea numărului de utilizatori și a cerințelor de performanță. Totodată, se impune și scalabilitatea în procesarea și gestionarea modelelor de Machine Learning, prin capacitatea de a funcționa în parametri optimi odată cu creșterea volumului de date de intrare.

3. **Securitatea datelor**

Aplicația trebuie să asigure integritatea și confidențialitatea datelor utilizatorilor cât și funcționalitatea de criptare a acestora pentru a proteja informațiile sensibile în timpul tranzitului și în stocare. Totodată aceasta trebuie să fie proiectată să ofere facilități de recuperare a datelor.

4. **Gestionarea erorilor**

Acest aspect implică gestionarea și tratarea erorilor în mod eficient și robust. Este recomandată monitorizarea activităților utilizatorilor și modificările în sistem în scopul auditării.

5. **Interfață intuitivă**

Aplicația trebuie să fie ușor de folosit, elementele UI trebuie să fie plasate în mod cât mai intuitiv, urmând principii de design coerent pentru a facilita înțelegerea și utilizarea aplicației.

3. Selectarea soluției tehnice

3.1 Arhitectura web de tip MERN

Arhitectura MERN este un cadru de dezvoltare web ce permite crearea aplicațiilor moderne ce utilizează următoarele tehnologii: MongoDB pentru baze de date, Express și Node.js pentru partea de backend și React pentru frontend. Fiecare tehnologie a acestei arhitecturi își are rolul său bine definit și contribuie la construirea unei aplicații web scalabile și robuste.

Stiva de tehnologii MERN se bazează pe principiile design-ului **MVC (Model-View-Controller)**. Acesta este un pattern de proiectare a unui produs software ce promovează o separare clară a atribuțiilor în cadrul unei aplicații web.

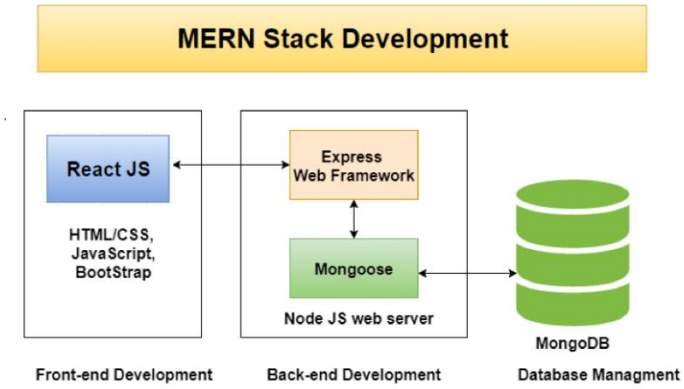

Fig. 5. Arhitectura Full-Stack MERN [7]

Conceptul MVC este compus din trei componente principale:

1. **Model**: este componenta care se ocupă de gestionarea datelor și de logica de bussiness a aplicației. Astfel, se evidențiază structura datelor, oferindu-se modalități de accesare și manipulare a acestora.
2. **View**: este responsabilă de prezentarea datelor utilizatorului. Vizualizarea concentrează interfețe grafice și elemente de afișare, create cu scopul de a reda utilizatorului o experiență interactivă.
3. **Controller**: este partea care supraveghează interacțiunea dintre Model și View. Acesta primește cereri de la utilizator prin intermediul interfeței și accesează și actualizează datele necesare apelând metodele corespunzătoare modelului, ca mai apoi să pregătească datele pentru a fi afișate.

3.1.1 MongoDB

MongoDB este un sistem de gestiune a bazelor de date non-realționale, orientat pe documente, care folosește un model de stocare flexibil, cunoscut sub numele de BSON (Binary JSON).

Structura fundamentală a acestui tip de baze de date presupune gruparea documentelor BSON în colecții, oferind flexibilitate deoarece câmpurile pot varia între diferite documente în aceeași colecție.

În ceea ce privește interogările la baza de date, această tehnologie pune la dispoziție o gamă variată de funcții de interogare pentru a căuta și a filtra datele cum ar fi: interogări de bază, interogări încorporate, de agregare, indexare, bazată pe text și geospațiale – pe coordonate geografice.

Comparând MongoDB cu alte baze de date relaționale, cum ar fi MySQL, se poate afirma faptul că Mongo stochează datele mai rapid deoarece nu este necesară definirea unei scheme în prealabil. Procesul de citire și scriere a datelor este, de asemenea, mai rapid deoarece toate informațiile pentru fiecare entitate sunt stocate în acelasi document. Un alt aspect prin care câștigă performanță în fața lui MySQL ar fi faptul că acesta nu are nevoie de o structura logică bazată pe tabele, fapt ce afectează viteza de procesare.

Mai jos, este prezent programul prin care se accesează baza de date, MongoDB Compass, cât și structura unui document dintr-o colecție.

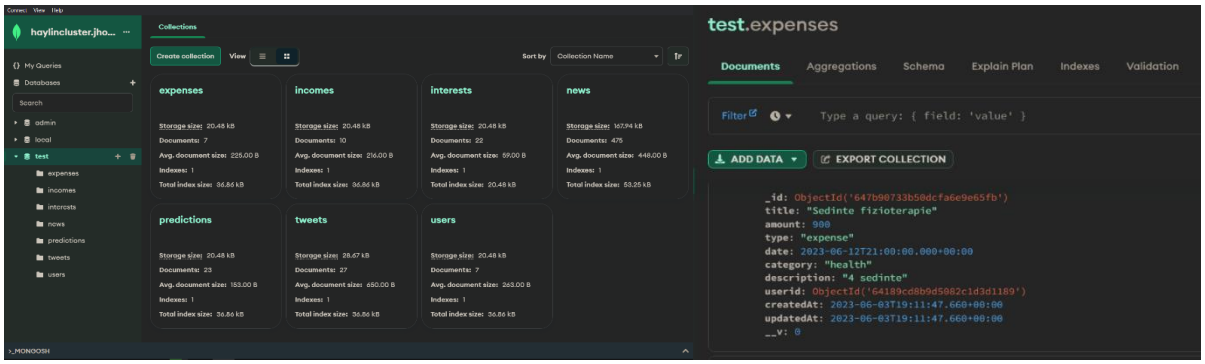

Fig. 6. Captură de ecran MongoDB Compass și exemplu structură document

3.1.2 Express.js

Express este un framework web, simplu și flexibil, ce permite crearea de aplicații Javascript, asigurând o abordare modulară. Acesta joacă un rol cheie în cadrul stivei MERN, fiind utilizat pentru a construi partea de backend a aplicației web. Acesta a fost folosit în

cadrul dezoltării aplicației pentru gestionarea rutei, a cererilor și a răspunsurilor HTTP, precum și manipularea sesiunilor.

Din cadrul acestui mediu de dezvoltare am folosit modulul "Router", pentru a defini rutele din cadrul metodelor HTTP, cum ar fi GET, POST, PUT și DELETE. Acesta permite împărțirea aplicației în module mai mici, fiecare având propriile rute și logica asociată, ce corespund funcționalităților aplicației.

3.1.3 Node.js

Node.js este un mediu de dezvoltare open-source care rulează pe motorul Javascript V8, fiind creat cu scopul de a permite executarea codului js în afara mediului browser-ului web, pe servere sau în alte contexte de dezvoltare. În ziua de azi, Node.js se folosește pentru a crea aplicații web de tip server-side, intensive din punct de vedere al volumului datelor deoarece se folosește de un model asincron și bazat pe evenimente.

În continuare, voi prezenta principalele caracteristici [8] ale acestei tehnologii care au fost luate în considerere în momentul dezvoltării aplicației:

- ***Abordarea ne-blocantă***: în momentul în care un server Node primește o cerere API de la un client, acesta nu trebuie să aștepte răspunsul API înainte de a trece la următoarea cerere. În acest fel, server-ul poate gestiona un număr mare de cereri fără a bloca execuția, folosind un model bazat pe evenimente.
- ***Arhitectura pe un singur fir de execuție***: prin intermediul buclei sale de evenimente, Node.js se folosește de o arhitectură cu un singur fir de execuție ceea ce îi conferă scalabilitate. Astflel, datorită mecanismului bazat pe evenimente, comparativ cu alte servere, cum ar fi Apache HTTP, acesta poate gestiona un număr mai mare de cereri.
- ***Compatibilitate cu alte platforme***: este un mediu de execuție cross-platform, ceea ce înseamnă că poate fi utilizat pe diverse sisteme de operare precum Windows, Linux, macOS și chiar dispozitive mobile.
- ***Transferul rapid de date***: timpul de procesare al datelor transmise prin diferite fluxuri în general durează mult timp. În schimb, node reușește să proceseze datele într-un interval foarte scurt la o rată foarte rapidă deoarece procesarea și încărcarea fișierelor se realizează simultan.

Totuși, deși apărut mai târziu, Node.js a atras atenția dezvoltatorilor datorită vitezei sale deoarece gestionează mult mai bine problema concurenței, având un consum constant, de cereri mici, rapide și nelimitate pentru date.

3.1.4 React JS

React este o bibliotecă JavaScript, pentru construirea interfețelor grafice cu utilizatorul. React se diferențiază de celelalte framework-uri de frontend prin abordarea sa bazată pe componente și pentru modul său eficient de a actualiza obiectele interfeței. Acesta încorporează stratul de vizualizare în modelul MVC oferind modularitate, facilitând crearea de aplicații web mai rapide. [9]

Un motiv pentru care s-a ales folosirea acestei tehnologii pentru partea de frontend a aplicației este datorat eficienței și performanței sale ridicate. Având în vedere faptul că aplicația implică analiza și procesarea datelor din postările de pe social media, precum și expunerea acestora, pentru a lua decizii de investiții, este important ca rezultatul să fie o interfață rapidă și reactivă.

Pentru a dezvolta o aplicație în React, este necesară folosirea sistemului npm (Node Package Manager). NPM este un sistem de gestionare a pachetelor pentru mediul de dezvoltare JavaScript Node.js. Acesta este inclus în instalarea Node.js și oferă o colecție de module și pachete reutilizabile. [10]

O caracteristică a acestei tehnologii ar fi conceptul de flux unidirecțional al datelor, exploatat prin intermediul bibliotecii **React Redux**, care facilitează integrarea și gestionarea stării aplicațiilor. Pentru a explica cum funcționează acest mecanism, trebuie mai întâi definiți următorii termeni:

1. **Store**: este obiectul ce conține starea aplicației
2. **Actions**: reprezintă colecția de obiecte ce exprimă ce anume va fi schimbat în starea din Redux Store
3. **Reducers**: sunt funcții ce gestionează acțiunile și transmit Store-ului cum se va schimba starea

Middleware-ul este un strat intermediar între dispatch-ul acțiunilor și reduceri. Acesta poate intercepta și modifica acțiunile înainte ca acestea să ajungă la reduceri.

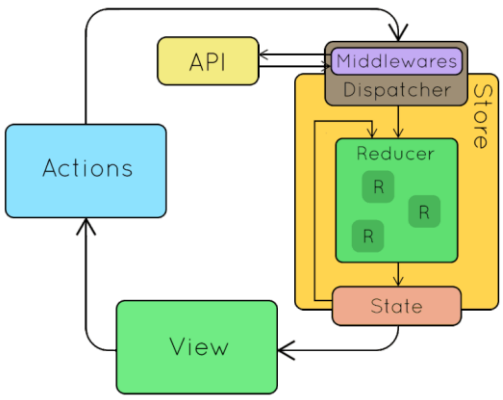

Fig. 7. Diagramă funcționare React Redux – flux unidirecțional al datelor [11]

3.2 Modelul de învățare automată

Pentru dezvoltarea acestei etape am folosit Python 3.9.13, împreună cu următoarele biblioteci:

- pandas și numpy - prelucrarea datelor
- sklearn – încărcarea modelelor de învățare automată
- transformers (BertTokenizer, TFBertModel) – vectorizarea BERT
- keras și tensorflow – Deep Learning
- pickle – salvarea setului de vectorizări
- matplotlib – realizarea graficelor

Modelul de învățare automată propus în această lucrare are ca scop analizarea știrilor economice disponibile pe platforma Google News și clasificarea acestora în trei categorii distincte: pozitive, negative și neutre. În contextul economic, această clasificare se referă la evaluarea sentimentului sau a impactului pe care aceste știri îl pot avea asupra piețelor financiare și economiei în general.

Știrile pozitive se referă la informații ce indică o creștere economică, perspective promițătoare pentru o companie, industrie sau active financiare.

Știrile negative pot fi informații despre recesiuni economice, scăderea performanței financiare, instabilitate politică, falimente, șomaj în creștere sau orice alt eveniment care poate indica o situație economică mai dificilă în cadrul unei perspective pesimiste.

Știrile neutre sunt acele informații care nu au un impact semnificativ asupra pieței economice și nu oferă o direcție clară în ceea ce privește evoluția unui instrument financiar. Acestea pot fi diverse rapoarte despre indicatori economici, sau orice altă constatare ce nu afectează în mod direct percepția asupra situației economice.

Deoarece informațiile financiare sunt foarte importante și pot avea un impact major asupra deciziilor de investiții, este necesară o analiză economică rapidă și eficientă pentru a înțelege modul în care afectează piața și pentru a lua decizii informate. Astfel, scopul final al acestui model este constituit din utilizarea acestei clasificări pentru a efectua o predicție a prețurilor în domeniul economic.

Etapa de antrenare a modelului de învățare automată începe prin alegerea unui set de date potrivit ce conține postări și știri din domeniul economic clasificate pe cele trei categorii menționate anterior. De menționat este faptul că atât antrenarea cât și testarea ulterioară a modelului prin prisma integrării în aplicația web au fost făcute folosind informații și știri în limba engleză.

Înainte de a utiliza aceste date pentru antrenare, se vor elimina datele irelevante, erorile și incoerențele. Astfel, se va realiza o preprocesare a datelor, ce presupune eliminarea caracterelor speciale, normalizarea textului, eliminarea duplicatelor și eventual, corectarea erorilor gramaticale. Un alt aspect important încadrat tot în această etapă este constituit de echilibrarea setului de date. Prin urmare, este necesară verificarea distribuției celor trei clase,

mai exact, dacă există suficiente exemple pentru fiecare clasă (pozitiv, negativ și neutru), pentru a evita dezechilibrele ce pot afecta performanța modelului.

Un alt aspect important ce contribuie la performanțele modelului este divizarea setului de date. Deci, setul de date inițial a fost împărțit în trei subseturi principale: antrenare, validare și testare astfel:

- Setul de date de *antrenare*: este folosit pentru a învăța modelul, iar din acest motiv a fost ales ca fiind cel mai consistent, în proporție de **70%** din setul inițial
- Setul de date de *validare*: este utilizat pentru a ajusta parametri modelului și pentru a alege cel mai bun model, reprezentând **10%** din cel inițial
- Setul de date de *testare*: este folosit pentru a evalua performanța finală a modelului, fiind ales în proporție de **20%** din setul de date inițial

Această împărțire a setului de date în antrenare, validare și testare permite o evaluare obiectivă a performanței modelului pe date noi, evitând probleme ce pot apărea cum ar fi *overfitting*. Overfitting este o problemă care poate apărea atunci când un model de învățare automată se adaptează prea bine la setul de date de antrenare, devenind foarte specific, neavând capacitatea de a generaliza pentru alte date diferite, cum ar fi cele de testare. O măsură prin care s-a tratat acest aspect a fost rezervarea unui subset de date de validare pentru ajustarea parametrilor modelului. Astfel, modelul a învățat să recunoască tipare relevante în setul de date de testare ce conține informații și știri noi pe care a reușit în cele din urmă să le generalizeze.

Există mai multe configurații posibile pentru un model deoarece există diferite variabile și opțiuni pe care le putem ajusta pentru a obține rezultate mai bune în funcție de problema și datele specifice cu care lucrăm. Spre exemplu, alegerea arhitecturii modelului implică stabilirea numărului și tipurilor de straturi precum și dimensiunile și conexiunile dintre straturi. Astfel, într-un model neuronal putem alege să avem un singur strat ascuns sau mai multe straturi ascunse, să utilizăm straturi convoluționale sau recurente și să decidem cum sunt conectate aceste straturi. Un alt aspect important de luat în calcul ar fi analiza parametrilor și a hiperparametrilor [12].

Parametri sunt variabilele interne pe care algoritmul de învățare automată îi utilizează pentru a prognoza rezultatele pe baza datelor de intrare. Metoda de învățare automată în sine utilizează o abordare de optimizare pentru a estima acești parametri. Drept rezultat, nu se pot seta sau codifica aceste variabile în mod explicit. Acestea sunt utilizate în procesul de antrenare al modelului și pot include ponderi sau decalări în cadrul straturilor unui model neuronal.

Hiperparametri, pe de altă parte sunt variabilele externe pe care utilizatorul le specifică în timpul procesului de antrenare a rețelei neurale. Ca rezultat, hiperparametri sunt furnizați înaintea parametrilor, sunt de fapt utilizați pentru a evalua parametri ideali ai modelului. Aceștia pot include numărul de straturi sau unități, rata de învățare și multe altele. Alegerea adecvată a hiperparametrilor poate duce la un model mai bun și la o performanță mai ridicată,

prin urmare, este important să se exploreze și să se ajusteze aceste valori pentru a obține cele mai bune rezultate.

Particularizând toate aceste aspecte, acest model de învățare automată, ce face predicții pe baza știrilor economice poate fi un instrument valoros, dar este important să se înțeleagă faptul că există și alte aspecte pe care utilizatorii ar trebui să le ia în calcul atunci când iau decizii financiare. Oricât de avansat ar fi un model de predicție, este dificil să se acopere toți factorii economici, politici, sociali și tehnologici și să se prevadă în mod precis evoluția pieței. Din acest punct de vedere, aplicația își propune totodată să informeze în mod transparent și să ofere acces la acele informații și știri pe baza cărora s-a ajuns la predicția finală. Scopul este de a oferi utilizatorului o experiență interactivă și de a-i permite să înțeleagă procesul și fundamentarea din spatele deciziilor financiare.

3.2.1 Procesarea limbajului natural

În cadrul aplicației Twinvest, analiza sentimentului este folosită pentru a decide dacă o știre va avea un impact semnficiativ asupra unui instrument financiar pentru a face o predicție de scurtă durată a prețului acestuia. Astfel, analiza sentimentului este realizată printr-o tehnică de procesare a limbajului natural, identificându-se și evaluându-se emoțiile, opiniile și atitudinile exprimate în text.

Procesarea limbajului natural implică mai multe nivele de analiză [13] pentru a înțelege și a procesa textul într-o manieră semnificativă. Cea mai intuitivă metodă de explicare a ceea ce se întâmplă de fapt într-un astfel de sistem inteligent este prin intermediul abordării nivelurilor limbajului. Cercetările psiholingvistice sugerează că procesarea limbajului este suficient de dinamică încât aceste nivele să interacționeze în diferite ordini. Acestea sunt: nivelul fonetic și fonologic, nivelul morfologic, nivelul lexical, nivelul semantic și nivelul sintactic.

Deoarece s-a demonstrat faptul că oamenii folosesc toate nivelurile limbajului pentru a înțelege o informație, cu cât sunt utilizate mai multe niveluri de limbaj, cu atât sistemul NLP este mai are o acuratețe mai bună.

Prima și cea mai importantă etapă este *preprocesarea* textului. Inițial, textul este supus unor operații de eliminare a zgomotului existent, ce se află sub mai multe forme, cum ar fi: punctuația, textul scris sub formă de carcatere numerice sau speciale, spațiile suplimentare, liniile goale, sau chiar emoji-uri.

Urmează *normalizarea* textului, ce presupune aplicarea diverselor operații pentru a standardiza informația. Aceasta poate include conversia literelor în litere mici, eliminarea diacriticelor sau corectarea ortografică. În continuare, se va trece la procesul de *lematizare* și *stemming.* Aceste metode concentrează reducerea cuvintelor la forma lor inițială, de bază. Procesul de stemming elimină sufixele și prefixele pentru obținerea rădăcinii, în timp ce lematizarea convertește cuvintele la forma lor de bază. De exemplu, în timpul lematizării,

verbele sunt reduse la infinitiv, iar substantivele și adjectivele la forma de bază, însoțite de gen și număr.

Totuși, întrebarea se pune în felul următor: Cum ar trebui să fie reprezentate aceste date într-un mod cu care sistemele de calcul să poată lucra? În acest context intervine *vectorizarea*. Mai multe informații despre această etapă vor fi prezentate în capitolul ce urmează.

3.2.2 Vectorizare

Vectorizarea este o tehnică utilizată pentru a descrie o abordare clasică de transformare a datelor de intrare din formatul lor brut, text, în vectori de numere reale, acesta fiind formatul general suportat de modelele de învățare automată. [14]

Una dintre cele mai simple metode de vectorizare a textului este reprezentarea BoW (*Bag of Words*). Un vector BoW are lungimea întregului vocabular – adică un set de cuvinte unice din corpus. Valorile vectorului reprezintă frecvența cu care fiecare cuvânt apare într-un anumit fragment de text. Mai jos, sunt niște reprezentări vizuale ale acestei tehnici prin care se poate observa transpunerea unui fragment într-un vector de frecvențe.

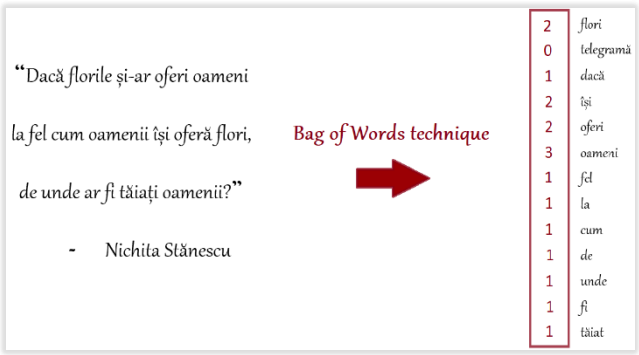

Fig. 8. Exemplu vectorizare folosind BoW

Deși reprezintă o soluție ușor de implementat, modelele BoW au limitări ce nu pot fi ignorate. Această tehnică nu reușește să înțeleagă în mod adecvat datele textuale. De exemplu, propozițiile "Iubesc animalele și urăsc oamenii" și "Iubesc oamenii și urăsc animalele" vor avea reprezentări vectoriale similare, deși ambele propoziții au înțelesuri complet diferite.

Tehnicile de vectorizare a textului cu pondere, precum TF-IDF [15] (perscurtarea de la *term frequency-inverse document frequency*), încearcă să atribuie scoruri de relevanță mai mari cuvintelor care apar într-un număr mai mic de fragmente în cadrul corpusului. Astfel, cu cât un cuvânt apare mai des, cu atât valoarea TF este mai mare pentru acel cuvânt. IDF arată cât de rar apare cuvântul de fapt.

Formulele folosite pentru agloritmul TF-IDF sunt:

$$\mathrm{tf}_{\mathrm{idf(t,d)}} = \mathrm{tf(t,d)} * idf(t) \quad (1)$$

$$\mathrm{idf(t)} = \log \frac{N}{df(t)} \quad (2)$$

- N - este numărul de documente din corpus
- df(t) – este numărul de apariții ale cuvântului t în toate documentele
- tf(t, d) – este numărul de apariții ale cuvântului t în documentul d împărțit la numărul total de cuvinte din d

Cea de-a treia metodă de vectorizare analizată, care a fost folosită și pentru modelul de la baza aplicației web, este BERT (*Bidirectional Encoder Representations from Transformers*). BERT [16] este un model de învățare automată dezvoltat în anul 2018 de către cercetătorii de la Google AI Language. Acesta este considerat ca fiind superior față de modelele tradiționale de vectorizare, cum ar fi cele prezentate anterior, deoarece aduce o înțelegere mai bună a contextului prin arhitectura sa bazată pe modele de limbă preantrenate. Astfel, poate identifica și distinge sensurile diferite ale cuvintelor în funcție de contextul în care sunt folosite.

BERT [17] propune o nouă arhitectură de rețea neuronală numită *Transformer*, care se bazează exclusiv pe mecanisme de atenție și elimină complet utilizarea rețelelor recurente și convoluționale. Modelul propus a fost comparat cu celelalte modelele tradiționale existent și s-a constatat că are performanțe superioare în ceea ce privește calitatea traducerii, capacitatea de paralelizare precum și obținerea unui timp de antrenament redus și a unor scoruri mai bune. Folosind mecanismul de atenție, acesta învață relațiile dintre cuvinte în funcție de context.

Rețelele neuronale *Transformer* au un mecanism de atenție care atribuie ponderi diferitelor părți ale intrării într-o secvență. Procesul de atenție începe prin calcularea scorurilor de atenție ale fiecărui element al intrării în comparație cu toate celelalte elemente. Aceste note sunt obținute prin produsul scalar, peste care se aplică o funcție de softmax între un vector de query și reprezentările elementelor (key). Score-urile de atenție arată cât de important sau relevant este fiecare element într-un anumit context.

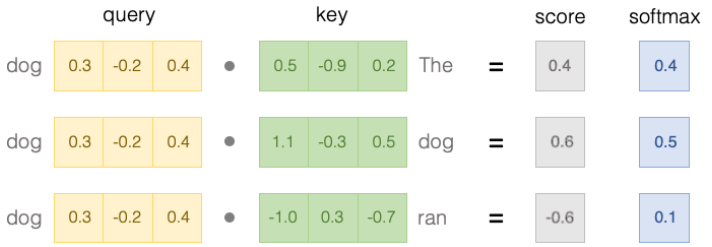

Fig. 9. Calcularea scorului de atenție [18]

Antrenarea modelului BERT a fost posibilă datorită unui set de date de 3,3 miliarde de cuvinte, arhitecturii Transformer cât și a vitezei de procesare asigurate de 64 de TPU (*Tensor Processing Units*), desfășurându-se pe o perioadă de 4 zile. [19]

În continuare vor fi explicate conceptele de la baza fiecărei litere din prescurtarea BERT [20]:

- **B**: Bidirecțional

Modelele anterioare lui BERT erau unidirecționale și puteau muta fereastra de context într-o singură direcție. BERT, pe de altă parte, utilizează modelarea limbajului bidirecțională. BERT poate vedea întreaga propoziție și o poate muta în dreapta sau în stânga în funcție de modelarea contextuală a limbajului.

- **ER**: Reprezentări ale encoder-ului

Atunci când rulăm un text printr-un model de limbaj, acesta va fi codificat înainte de a fi furnizat ca intrare. Doar textul codificat poate fi procesat și ne va furniza o ieșire finală. Ieșirea oricărui model va fi, de asemenea, într-un format criptat, care necesită decriptare. Deci, atunci când un mesaj este codificat, acesta va fi decodificat din nou. Este un mecanism de intrare-ieșire.

- **T**: Transformer

Transformerele creează ponderi diferențiale care semnalizează care cuvinte dintr-o propoziție sunt cele mai importante pentru a fi procesate ulterior. Un transformer realizează acest lucru prin procesarea succesivă a unei intrări prin intermediul unui set de straturi de transformere, denumite în mod obișnuit encoder. Dacă este necesar, un alt set de straturi de transformere - decoderul - poate fi utilizat pentru a prezice o ieșire țintă. - Cu toate acestea, BERT nu utilizează un decoder.

Pentru pre-antrenare, BERT folosește doua metode diferite. Prima metodă, se numește *Masked LM* și presupune înlocuirea în mod aleatoriu a 15% din cuvinte, cu simbolul [MASK], modelul având ca scop prezicerea acestora prin adaugarea unui clasificator peste output-ul encoder-ului. Cea de-a doua metodă este *Next Sentence Prediction* (NSP) și presupune prezicerea dacă două propoziții consecutive dintr-un text sunt corelate semantic sau nu. Această abordare are scopul de a ajuta modelul să dezvolte o înțelegere mai profundă a contextului și a relațiilor între propoziții într-un text.

După preantrenare, urmează etapa de *fine-tuning*. În această etapă, BERT este adaptat pentru a rezolva o anumită sarcină specifică de analiză a limbajului natural, cum ar fi clasificarea de texte, etichetarea entităților sau răspunsul la întrebări. Modelul BERT este ajustat pe un set de date etichetate pentru a se specializa în acea sarcină specifică. Acest proces permite modelului să înțeleagă contextul și să facă predicții precise pentru sarcina respectivă.

3.2.3 Rețele neuronale

Rețelele neurale sunt modele de învățare automată, inspirate de structura și funcționarea creierului uman, compuse din straturi de neuroni interconectați, informația fiind transmisă prin conexiunile ponderate între aceștia.

În cadrul rețelelor neuronale, un neuron este un model matematic, o unitate de procesare a informației, ce calculează media ponderată a intrărilor sale și aplică un decalaj asupra rezultatului. Decalajul este o valoarea constantă adăugată la rezultatul ponderat înainte de a fi trecut prin funcția de activare. Acesta permite modelului să fie mai flexibil și să se adapteze mai bine la datele de intrare.

Rezultatul ponderat al intrărilor peste care a fost aplicat decalajul este trecut mai apoi printr-o funcție de activare non-lineară. Funcțiile de activare folosite cel mai des sunt funcția sigmoidă, funcția ReLU (Rectified Linear Unit) și funcția tanh. Aceste funcții transformă suma ponderată a intrărilor unui neuron într-o valoare de ieșire într-un anumit interval. Această ieșire poate fi folosită ca intrare pentru alți neuroni sau poate fi chiar rezultatul final al rețelei. Această arhitectură este dispusă pe trei straturi.

Stratul de intrare reprezintă primul strat dintr-o rețea neurală tipică. În această zonă, neuronii primesc informațiile de intrare, acestea fiind procesate de funcția de activare și o transmit către stratul următor, în funcție de o valoare de prag definită. La final, rezultatul este scalat utilizând ponderile prestabilite, care sunt asociate cu conexiunile dintre neuronii din stratul de ieșire și neuronii din cel de intrare.

O rețea neurală poate avea unul sau mai multe *straturi ascunse,* astfel că, neuronii din acest strat primesc intrările fie de la neuronii din stratul de intrare, fie de la neuronii din stratul ascuns anterior. Aici se prelucrează intrarea prin funcția de activare neliniară și se propagă rezultatul către neuronii din următorul strat.

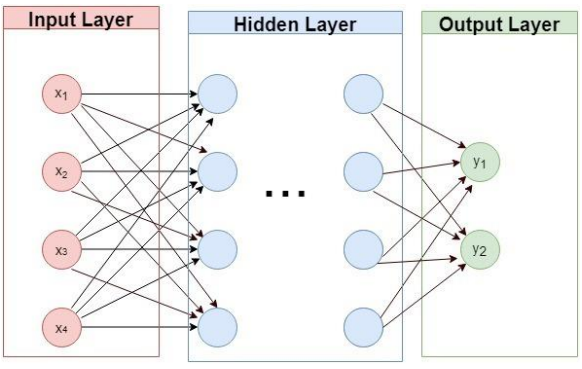

Fig. 10. Schema unei rețele neurale artificiale [21]

Stratul de ieșire este ultimul strat, care marchează finalul logic al rețelei. Aici, neuronii din stratul de ieșire primesc intrările de la straturile anterioare, procesându-le tot prin funcții de activare specifice. Rezultatul acestui strat poate fi o clasificare binară, ce conține

doar un neuron, indicând apartenența la una din cele două clase, o valoare de predicție, sau precum este și în cazul aplicației dezvoltate, poate fi o clasificare cu mai multe clase.

4. Considerente legate de dezvoltarea aplicației

Aplicația a fost dezvoltată prin prisma a cinci componente cheie: baza de date, partea de backend, partea de frontend și printr-o logică separată, partea de preluare a știrilor de pe Google News și modelul de învățare automată.

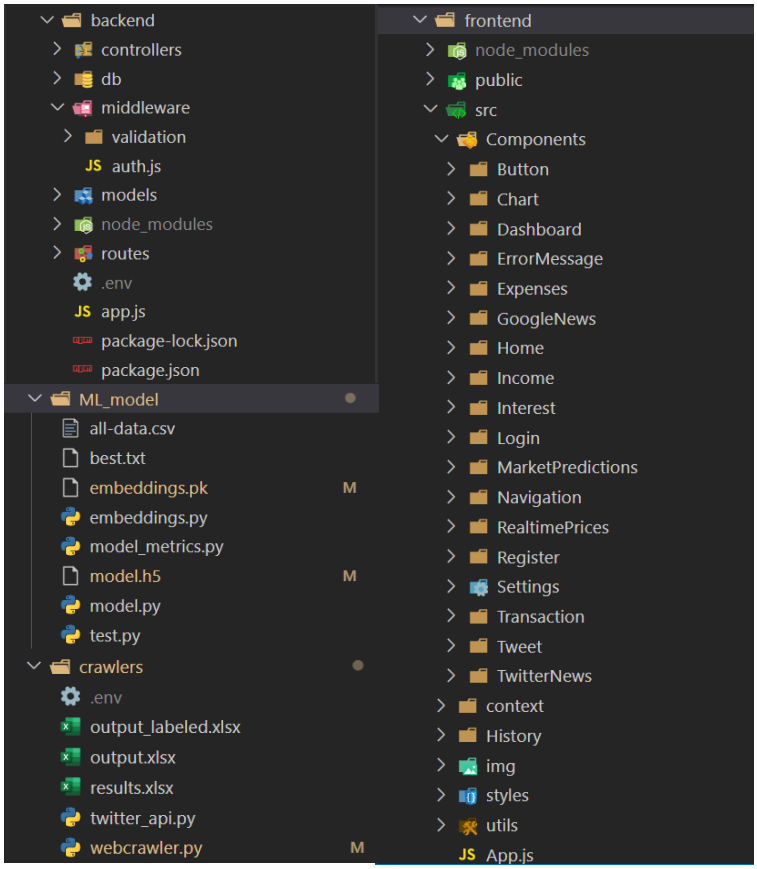

Fig. 11. Structura fișierelor în aplicația Twinvest

4.1 Arhitectura aplicației

În ceea ce privește arhitectura aplicației web, aceasta se folosește de stiva MERN, explicată în capitolul 3.1 dar și de câteva tehnici de structură ce oferă robustețe și scalabilitate.

În primul rând, aplicația se bazează pe un model de tip *client-server*, astfel că, partea de backend (server) și partea de frontend (client) sunt separate. Prin urmare, implementarea clientului este responsabilă pentru afișarea interfeței grafice și de interacțiunea cât mai intuitivă cu utilizatorul, în timp ce partea de server se ocupă de comunicarea cu alte API-uri, baza de date și gestionarea acestora.

Totodată, pentru facilitarea acestei comunicări între cele două entități, o altă tehnică folosită este *RESTful API*. Aceasta se bazează pe principiile arhitecturale REST (*Representational State Transfer*), manipulând datele într-o manieră eficientă și consistentă prin metodele HTTP.

Acestor cereri HTTP au fost atașate instrumentele de *middleware*. Acestea sunt funcții ce au rolul de a adăuga funcționalități suplimentare înainte sau după rutele principale.

De asemenea, aplicația dispune de o arhitectură stratificată. Această abordare implică împărțirea aplicației în straturi logice distincte, fiecare strat având responsabilitățile sale, comunicând cu straturile adiacente. În figura de mai jos se poate observa separarea celor două entități precum și elementele ce compun straturile: Infrstructură, Accesul la date, Logica aplicației și Interfața utilizatorului.

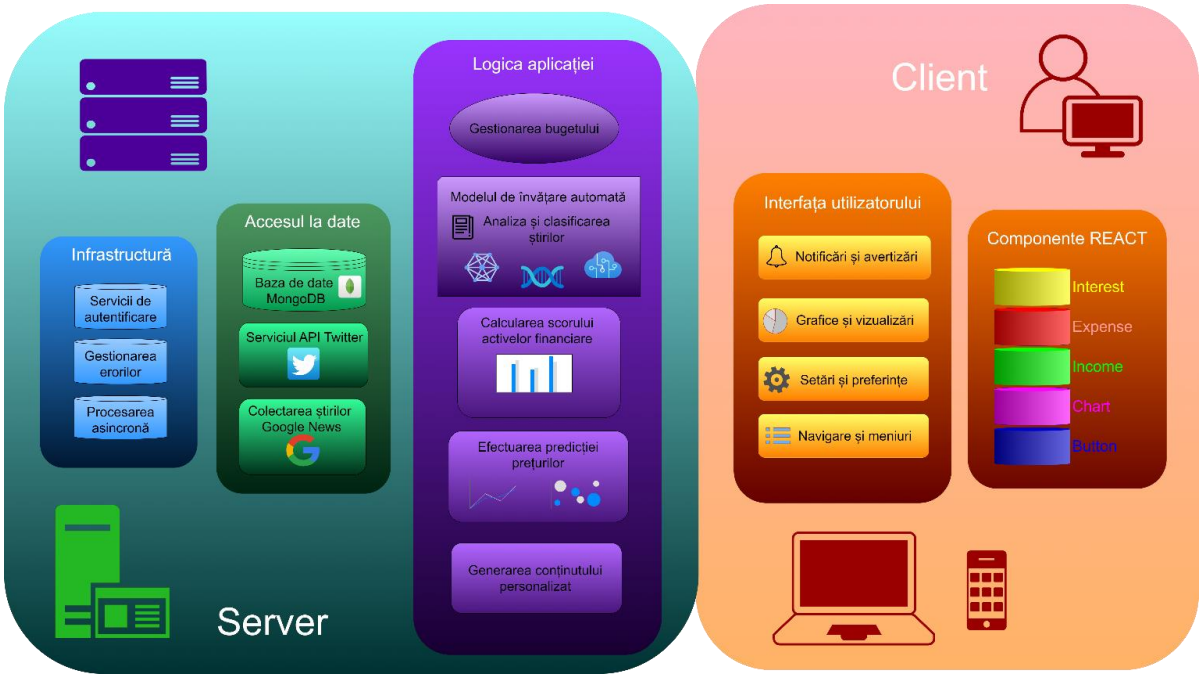

Fig. 12. Arhitectura stratificată a aplicației Twinvest

4.2 Implementarea aplicației web

4.2.1 Baza de date

În contextul unei aplicații web, baza de date reprezintă o componentă esențială pentru stocarea datelor. Având în vedere faptul că scopul principal al aplicației este acela de analiză a datelor din domeniul economic, baza de date non-relațională folosită gestionează postări și știri din ultimele două zile, precum și informații despre bugetul fiecărui utilizator în parte. Datele și știrile ce nu mai sunt de actualitate sunt șterse din baza de date, urmând ca prin intermediul web scraper-ului să fie extrase altele.

Pentru a putea gestiona corespunzător toate aceste informații, s-au creat în total șapte colecții. Având în vedere faptul că pentru această aplicație a fost folosită o bază de date non-relațională, aceasta funcționează independent astfel:

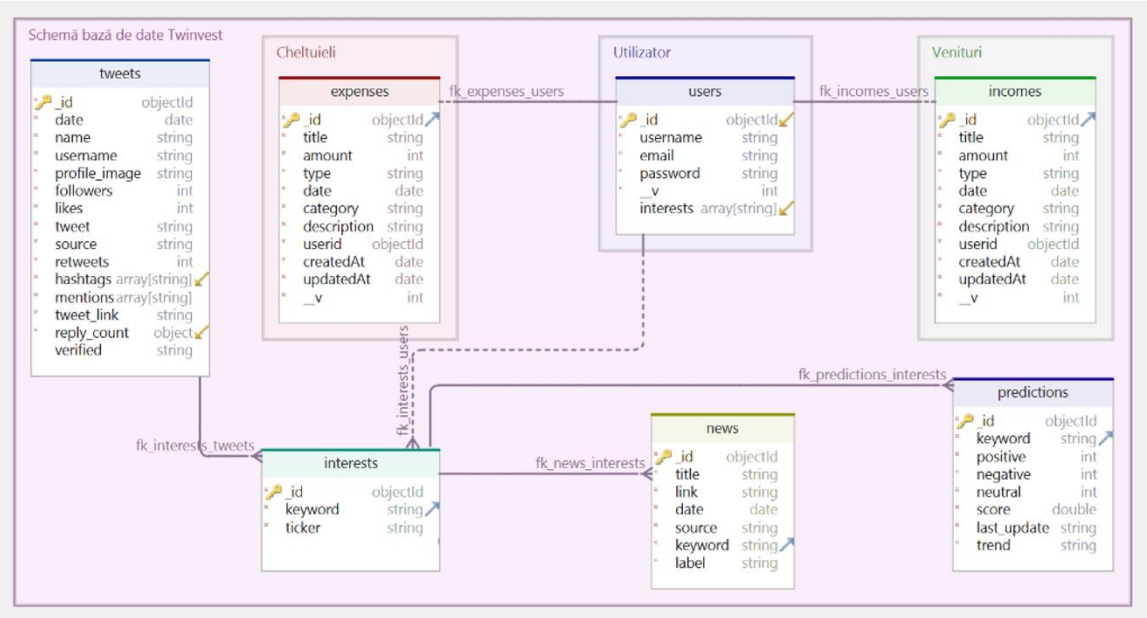

Fig. 13. Arhitectura bazei de date

- **User** – conține informațiile referitoare la utilizatori, precum: username, e-mail, parola sub formă de hash și lista cu interese, care, în momentul înregistrării este nulă.
- **Incomes** – conține informații referitoare la veniturile utilizatorului, ce sunt adăugate manual. Un venit este caracterizat de o denumire, o sumă, data adăugării, o categorie și o descriere. Aceasta conține totodată o cheie străină către utilizator.
- **Expenses** – conține informații referitoare la plățile utilizatorului. Structura acesteia este asemănătoare cu cea a veniturilor.
- **Interests** – conține informații despre interesele posibile ale utilizatorilor. Acestea sunt de fapt criptomonedele și acțiunile pe care le poate urmări un investitor. De menționat este faptul că această listă de interese este una predefinită. Fiecare interes conține un

titlu și un ticker NASDAQ – care este de fapt un simbol unic utilizat pentru a identifica și urmări activele financiare listate pe bursa de valori NASDAQ Stock Market. Acestea sunt folosite pentru a prelua datele în timp real despre schimbările procentuale ale fiecărei criptomonede/acțiune financiară.

- **News** – sunt știrile extrase de pe Google News, în funcție de o listă de cuvinte cheie, fiecare dintre acestea fiind compusă dintr-un titlu, link-ul corespunzător, data apariției, sursa, cuvântul cheie după care a fost extrasă și eticheta, care poate fi: pozitiv, negativ sau neutru, în funcție de analiza făcută de modelul de învățare automată.
- **Tweets** – această colecție păstrează postările relevante de pe Twitter, preluate prin intermediul Twitter API.
- **Predictions** – reprezintă informații despre predicțiile care se fac pentru fiecare activ financiar în parte. Această colecție conține un număr egal de documente cu **Interests**, având practic aceeași listă de active, doar că, în acest caz, predicțiile sunt actualizate o dată la două zile. O predicție conține denumirea activului, numărul de știri pozitive/negative și neutre pentru fiecare, extrase în ultimele două zile, scorul, calculat pe baza unei formule, data când a fost actualizată și tendința, ce poate fi crescătoare/descrescătoare, în funcție de scorul obținut.

De menționat este faptul că fiecare dintre aceste colecții conține un identificator unic "_id", de tipul ObjectId, generat automat de mecanismul folosit de MongoDB.

4.2.2 Backend

Partea de backend dintr-o aplicație de tip MERN se ocupă de gestionarea și prelucrarea datelor, comunicarea cu baza de date, furnizarea de servicii și propagarea funcționalităților către partea de frontend.

Astfel, în primul rând, această parte începe prin configurarea unui server de Node.js care face conexiunea cu baza de date prin intermediul unui URL și folosindu-se de biblioteca Mongoose, ce este de fapt un ODM (Object Data Modeling) pentru interacțiunea cu MongoDB.

Totodată, generarea colecțiilor pentru User, Income, Expense etc. a fost realizată în cadrul secțiunii models. Modelele Mongoose sunt definite în module separate, precizându-se câmpurile și tipurile de date pentru fiecare. Acest modul este exportat în cele din urmă pentru ca datele din baza de date să fie manipulate cu ușurință, aplicându-se diverse interogări.

Odată ce aceste modele au fost clar definite în cod, prin intermediul abordării Database First, acestea pot fi folosite pentru dezvoltarea logicii aplicației.

Logica aplicației a fost creată prin intermediul controller-elor. Un controller reprezintă o componentă a server-ului ce se ocupă de tratarea și prelucrarea cererilor, acționând ca o punte între rutele definite cu Express.js și modelele definite. Prin umare,

acestea sunt de fapt niște funcții de bază ce se folosesc de interogări pentru a realiza diferite funcționalități în cadrul aplicației web. Toate aceste funcționalități au fost împărțite în mai multe fișiere JavaScript, după cum urmează:

- **Expense / Income** – pentru adăugarea, ștergerea și vizualizarea plăților sau veniturilor. Pentru fiecare dintre acestea s-au folosit interogările: *.find,* pentru a găsi utilizatorul de care aparțin și *.save,* respectiv *.findByIdAndDelete* pentru a insera și sterge anumite documente. Aceste metode au fost aplicate pe modelele definite anterior, care au fost importate.
- **Interests** – contribuie la afișarea prețurilor în timp real pentru activele financiare pe care utilizatorii le-au marcat ca fiind domeniu de interes în pagina de Setări.
- **News / Tweet** – conțin o singură funcție, sub forma unui get, ce preia știrile/ postările adăugate prin intermediul procesului de scraping în baza de date, filtrându-le totodată cu ajutorul obiectului de condiție *$in.* Această metodă selectează doar știrile ce conțin cuvinte cheie ce se află în lista de interese a utilizatorului.
- **Predictions** – are o structură similară cu cea a controller-elor News și Tweet, în sensul în care sunt preluate datele referitoare la predicții pentru activele financiare din lista utilizatorului.
- **User** – metode ce permit: crearea unui utilizator, autentificarea acestuia, actualizarea setărilor (interese/parolă), setarea și actualizarea unui obiectiv de economisire sau a unei limite de cheltuire.

De menționat este faptul că, pentru toate acestea a fost utilizat operatorul *await,* deoarece toate funcțiile au fost definite ca *async.* Scopul este acela de a aștepta finalizarea interogării, codul executându-se într-un context asincron, permițând realizarea unei operații de interogare fără a bloca firul principal de execuție.

Pentru modelul utilizatorului, s-a folosit middleware-ul pentru criptarea parolei utilizatorului, înainte ca acesta să fie introdus în baza de date. Pentru această criptare a fost folosită biblioteca bcrypt. Astfel, prin intermediul funcției *bcrypt.hash()* se va prelua parola text urmând ca aceasta să fie transpusă într-un hash criptografic. Hash-ul va fi o valoare unică, de dimensiuni fixe, procesul fiind unidirecțional. Pentru a adăuga un nivel suplimentar de securitate, s-a adăugat un salt, ce va fi combinat cu parola înainte de a fi trecută prin algoritmul de hashing. Prin utilizarea acestuia se previn atacurile ce presupun folosirea unui dicționar.

Pentru autentificarea utilizatorului, bcrypt va prelua parola introdusă de acesta în formularul de Login pe care o va cripta folosind același hash și salt și o va compara cu hash-ul din baza de date, prin intermediul funcției *bcrypt.compare().* Totodată, pentru transmiterea și validarea informațiilor de autentificare am folosit JWT (JSON Web Token), ce are rolul de a impune un format securizat pentru transmiterea datelor în format JSON. În momentul în care utilizatorul se autentifică cu succes, serverul va genera un token cu funcția *JWT.sign().* Acest token va fi transmis clientului, în câmpul „Authorization" al antetului fiecărei cereri HTTP. Serverul va valilda de fiecare dată semnătura pentru a verifica integritatea acestuia și va prelua informațiile referitoare la utilizator pentru a lua decizii de autorizare.

Tot în cadrul middleware-ului au fost create și metodele pentru validare și prevenire a erorilor, cum ar fi declanșarea mesajelor de avertizare a utilizatorului în momentul în care nu completează anumite câmpuri obligatorii sau restricția ce presupune respectare unui număr minim sau maxim de caractere. Toate acestea au rolul de a ghida utilizatorul în momentul folosirii aplicației, fiind atașate rutelor în momentul în care acestea sunt definite.

În total, pentru această aplicație au fost implementate 20 de rute, fiecare dintre acestea contribuind la câte o funcționalitate specifică. Pentru crearea unui obiect de rutare care să conțină toate aceste rute din backend, am folosit funcția Router() din Express.js. Aceasta permite definirea și gestionarea rutelor în aplicație. Pe acest obiect creat s-au făcut apeluri succesive ale metodele de rutare: *get(), post(), put()* și *delete()*.

GET	**/getIncomes** – returnare venituri
	/getExpenses – returnare plăți
	/getTweets – returnare postări Twitter
	/getNews – returnare știri Google News
	/getPredictions – returnare predicții active finaciare
	/getInterestsPrices – returnare prețuri în timp real
	/getInterests – returnare interese utilizator
	/getExpenseLimit – returnare limită de cheltuire
	/getIncomeObjective – returnare obiectiv de economisire
POST	**/addIncomes** – adăugare venit
	/addExpenses – adăugare plată
	/addUser – creare utilizator
	/signIn – autentificarea utilizator
PUT	**/updateInterests** – actualizarea intereselor utilizatorului
	/changePassword – schimbarea parolei
	/initIncomeObjective – actualizarea obiectivului de venit
	/initExpenseLimit – actualizarea limitei de cheltuire
DELETE	**/deleteIncomes/:id** – ștergerea unui venit
	/deleteExpenses/:id – ștergerea unei plăți
	/deleteInterest – ștergerea unui interes

Tabel 2. Definirea rutelor din backend

Infrastructura de backend include și preluarea postărilor de pe Twitter și a știrilor de pe Google News, de care modelul de învățare automată se va folosi pentru calcularea unui scor pe baza căruia se vor realiza predicții.

Procesul de preluare a știrilor de pe a fost dezvoltat prin intermediul unui *web scraper*, realizat în Python. Pentru acesta am folosit, după cum se poate observa mai jos, biblioteca *BeautifulSoup* pentru a parsa paginile web și pentru a extrage informațiile dorite din utlimele două zile, pe baza unei liste de cuvinte cheie. Mai apoi, folosind metoda *find_all(),* se extrag toate elementele de tip *<item>* care reprezintă știri. Funcția *crawl()* este funcția principală, care inițiază o structură sub formă de tabel prin intermediul unui *DataFrame* și salvează știrile găsite într-un fișier *output.xlsx,* ce conține următoarele coloane: Titlu, Link, Data apariției, Sursa, Cuvântul cheie și Eticheta (pozitiv/negativ/neutru) ce va fi completată în momentul apelării modelului de învățare automată pe toate aceste date.

```python
def crawl(page_limit=None, keywords=None):
    df = pd.DataFrame(columns=
    ['Title', 'Link', 'Date', 'Source', 'Keyword', 'Label'])
    df.to_excel('output.xlsx', index=False)
    if keywords is None:
        result = search(page_limit)
        news_list = result['news_list']
        keyword = None
    else:
        news_list = []
        for kw in keywords:
            encoded_kw = quote(kw)
            result = search(encoded_kw, page_limit)
            news_list.extend(result['news_list'])
            keyword = result['keyword']
            success, failed, df = archive(news_list, df, keyword)
    df.to_excel("output.xlsx")
```

```python
def search(search_string, page_limit=None):
    news_url = f'https://news.google.com/rss/'
    f'search?q={search_string}'
    client = urlopen(news_url)
    xml_page = client.read()
    client.close()
    soup_page = BeautifulSoup(xml_page, 'xml')
    news_list = soup_page.find_all('item')

    # Filtram stirile din ultimele 2 zile
    filtered_news = []
    current_date = datetime.now()
    one_day_ago = current_date - timedelta(days=1)
    for news in news_list:
        pub_date = datetime.strptime(news.pubDate.text,
            '%a, %d %b %Y %H:%M:%S %Z')
        if pub_date >= one_day_ago:
            filtered_news.append(news)
    return {'news_list': filtered_news[:page_limit],
            'keyword': search_string}
```

Fig. 14. Extragerea știrilor Google News relevante

Pentru extragerea postărilor de pe Twitter am folosit TwitterAPI [22] în cadrul unui script de Python, selectând doar acele tweet-uri care conțin unul dintre cuvintele cheie dintr-o listă predefinită pe care o primește ca argument. Datele extrase pentru un tweet sunt: numele utilizatorului, imaginea de profil, numărul de aprecieri, comentarii și partajări, textul tweet-ului, sursa, link-ul acestuia și data postării. Un alt criteriu după care au fost selectate tweet-urile a fost importanța acestora, măsurată prin numărul de urmăritori ai contului respectiv, precum și numărul de aprecieri și partajări, setându-se un prag minim pentru acestea. În momentul în care sunt găsite, acestea se vor adăuga direct în baza de date MongoDB, în colecția Tweet, prin intermediul funcției *insert_one().*

4.2.3 Frontend

Interfața grafică a acestei aplicații a fost realizată în mediul de lucru React.js ce permite structurarea pe componente grafice ce conțin atât cod de JavsScript cât și de stilizare

(HTML și CSS). Odată creată și exportată, o componentă React se poate folosi oriunde în cadrul interfeței, acest aspect oferind modularitate. Astfel, aceste componente au fost folosite pentru diverse elemente precum: Buton, Grafic, Mesaj de Eroare, Venit, Plată, Setări, etc.

Totodată, pentru fiecare pagină a aplicației (Register/Login, Dashboard, Twitter News, Google News, Income, Expense, Market Predictions, Settings, Realtime Prices) a fost creată câte o componentă separată pentru a fi transmisă ca parametru fiecărei rute. Astfel, în funcție de link-ul accesat de utilizator, se va randa una dintre acele componente. De această dată, rutele au fost create cu ajutorul funcției din biblioteca *react-router-dom,* Router. Aceste rute au rolul de gestionare a navigării și a interacțiunii utilizatorului cu aplicația.

Stilizarea interfeței cu utilizatorul reprezintă un aspect important al dezvoltării aplicațiilor web, deoarece aceasta trebuie să fie cât mai intuitivă. Câteva dintre bibliotecile folosite în acest sens sunt: Bootstrap, Material-UI, react-tweet-card (pentru afișarea postărilor extrase de pe Twitter), ChartJS (pentru afișarea statisticilor), react-notifications (pentru afișarea mesajelor de notificare și eroare).

Pentru gestionarea stării globale a aplicației într-un mod cât mai eficient și modular, s-a folosit Context API, ce este o caracteristică a bibliotecii React. Principiul de bază al acestuia este crearea unui "context", cu ajutorul funcției ce va conține date și funcții care vor fi accesibile tuturor componentelor. Apelarea acestei funcții va returna un obiect format din *Provider* și *Consumer*. În mod evident, Provider va fi componenta ce va furniza funcțiile și valorile transmise prin context. Acesta conține definițiile, starea globală și datele ce vor fi partajate prin intermediul reducerilor. Aceștia primesc starea curentă și o acțiune și returnează o nouă stare bazată pe acea acțiune. Funcția care permite componentelor să își gestioneze starea internă este *useState().* Aceasta este de fapt un hook din biblioteca React, care permite declararea și actualizarea stării în timpul rulării aplicației. Totodată, pentru propagarea anumitor funcții declanște de acțiuni, precum: setarea unei limite de cheltuire sau schimbarea parolei, s-a folosit hook-ul *useEffect()* pentru gestionarea efectelor secundare. Consumer-ul, din cadrul Context API va accesa acest context global prin intermediul hook-ului *useContext()* pentru a afișa datele în funcție de cererile trimise către aplicație.

Fiecare reducer din cadrul contextului global, reprezentând de fapt câte o funcționalitate, are asociată câte o rută din backend, solicitarea realizându-se prin intermediul funcțiilor din cadrul bibliotecii *Axios*. Aceasta este folosită pentru a efectua solicitări HTTP către server de tipul: *axios.post(), axios.get(), axios.put()* sau *axios.delete().* Solicitările sunt trimise la URL-ul specificat ca prim parametru, iar datele necesare pentru efectuarea acesteia sunt transmise în cel de-al doilea argument al funcției.

De asemenea, în antetul fiecărui reducer este inclus și token-ul de autorizare, preluat cu funcția *getToken(),* pentru a asigura că utilizatorul este autentificat și are dreptul de a efectua acea acțiune. Erorile în cadrul solicitării sunt tratate prin intermediul unui *.catch().*

Tot în partea de frontend este realizat și apelul către API-ul RealStonks [23], ce furnizează date în timp real despre evoluția acțiunilor la bursă și a criptomonedelor. Efectul de actualizare în timp real este redat prin intermediul blocului *useEffect*, care declanșează o funcție asincronă de *fetch* în momentul în care se schimbă valorile din vectorul de

dependențe. Apelul către rută este realizat ca o metodă GET, pentru fiecare instrument financiar urmărit de utilizatorul respectiv. Antetul acestui apel conține o cheie unică generată în momentul creării contului pe platforma RapidAPI, precum și domeniul accesat. Rezultatul returnat este sub formă de procent, arătând cu cât a crescut sau a scăzut prețul acelui activ financiar față de ziua precedentă.

Toată această arhitectură a frontend-ului compusă din formulare prin care utilizatorii își pot adăuga manual veniturile și plățile, grafice pentru vizualizarea statisticilor, butoane, slidere și animații sunt completate de o atentă analiză a excepțiilor tratate și în mod vizual, cu ajutorul pachetului react-notifications.

4.3 Analiza știrilor din domeniul economic

4.3.1 Setul de date

Setul de date utilizat pentru antrenarea rețelei neuronale este cunoscut sub numele de *FinancialPhraseBank* [24] și conține sentimente asociate titlurilor de știri din domeniul economic, analizate din perspectiva unui investitor individual.

Setul de date cuprinde două coloane principale, “Sentiment” (este generat de starea pe care o transmite știrea și poate fi pozitiv/negativ sau neutru) și “News Headline” (Titlul știrii). Astfel, toate aceste date reprezintă o colecție de propoziții împărțite în cele trei clase în funcție de sentimentul pe care îl transmit, provenind din știri financiare. Setul conține 4840 de propoziții în limba engleză, etichetate cu un sentiment de către 16 evaluatori. Dintre cei 16 evaluatori, trei au fost cercetători, iar ceilalți 13 evaluatori au fost studenți la master la Universitatea Aalto, cu specializări în domeniul financiar, contabilitate și economie.

Corpusul utilizat în acest studiu este format din știri despre toate companiile listate în OMX Helsinki [25] – bursa de valori din Finlanda, una dintre diviziile NASDAQ. Știrile au fost descărcate din baza de date LexisNexis, utilizând un web scraper automat. Din această bază de date de știri, au fost selectate în mod aleatoriu 10.000 de articole, pentru a obține o acoperire cât mai bună în ceea ce privește companiile mici și mari, companiile din diferite industrii și sursele de știri diferite.

În articolul [26] asociat acestui set de date, este menționată utilizarea lexicelor de polaritate, ca fiind cea mai des adoptată abordare pentru identificarea orientărilor semantice (pozitive, negative sau neutre) în text. Totuși, performanța acestei metode este influențată de domeniul și contextul în care este aplicată. De exemplu, cuvântul “scădere” poate avea o conotație negativă în contextul pieței de valori, dar poate avea o conotație pozitivă în contextul reducerii costurilor în afaceri.

Mai jos se poate observa distribuția claselor pentru setul de date, astfel că 59% din știri poartă eticheta neutru, 28% dintre acestea – pozitiv și restul de 12% - negativ.

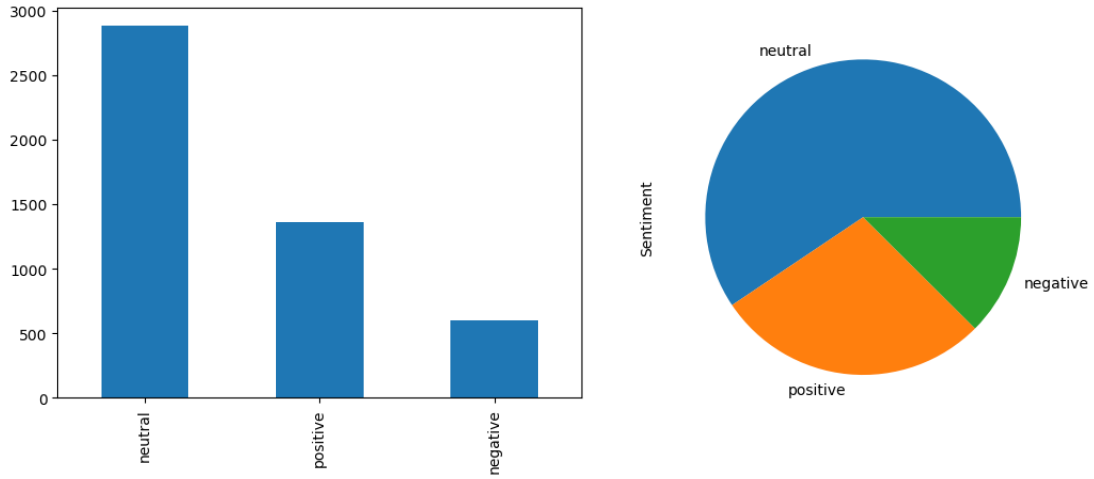

Fig. 15. Distribuția claselor pe setul de date

Un factor important ce poate afecta acuratețea clasificării realizată de modelul antrenat pe acest set de date este și modul în care acesta a fost etichetat de către evaluatori, precum și perspectiva subiectivă a fiecăruia. Rezultatele etichetării de către cei 16 evaluatori arată că mai puțin de jumătate din știrile ce compun acest set de date, adică mai exact doar 2259 din totalul de 4840 au un procent de acord unanim în privința atribuirii unui sentiment.

În tabelul de mai jos se pot observa câteva exemple de știri din setul de date precum și eticheta acestora.

Sentiment	News Headline
positive	In the third quarter of 2010 , net sales increased by 5.2 % to EUR 205.5
neutral	So , the sales growth of cars considerably influence on the tires market
negative	Salonen added that data shows producers ' pulp inventories in North America are declining

Tabel 3. Exemple de propoziții din setul de date

4.3.2 Antrenarea modelului de învățare automată

Antrenarea rețelelor neuronale implică o etapă importantă de preprocesare și pregătire a datelor. Prima etapă din acest proces este vectorizarea setului de date, adică transformarea setului de date ales într-o reprezentare numerică pe care modelul o poate utiliza pentru a învăța și a face preziceri.

Pentru etapa de vectorizare, s-a folosit modelul preantrenat "bert-base-uncased". Astfel, cu ajutorul claselor *BertTokenizer* și *TFBertModel* din modulul *transformer*, s-a creat un tokenizer, ce va fi folosit pentru împărțirea textului în unități mai mici. Tokenizer-ul BERT are o înțelegere a vocabularului și a structurii limbajului învățate de modelul BERT.

În continuare se va transforma eticheta folosită pentru fiecare știre într-un număr corespunzător astfel: 1 – neutru, 2 – pozitiv, 3 – negativ. Se parcurge fiecare rând din tabel și se va apela modelul BERT pentru a obține reprezentarea numerică a fiecărei propoziții împărțite în tokeni. Funcția *tokenizer.encode()* primește propoziția și aplică tokenizarea, asigurându-se că lungimea secvenței de tokeni este maxim 512, aceasta fiind o restricție specifică pentru modelul BERT. Dacă lungimea depășește limita, se vor aplica trunchieri. Condiția de oprire a acestei procesări este marcată de finalul fișierului .csv. Setul de date vectorizat se va salva folosind modulul pickle pentru a serializa obiectele ce conțin datele de antrenare în formă de vectori și etichetele corespunzătoare. Acest program de obținere a vectorizării a rulat timp de aproximativ 6 ore.

Pentru antrenarea propriu zisă a modelului de clasificare am folosit o rețea neurală de tip *Feedforward* (Sequential), din biblioteca Keras. Feedforward se referă la o arhitectură în care informația se deplasează secvențial, într-o singură direcție, de la stratul de intrare, la cel de ieșire fără a exista bucle în rețea. Implementarea este realizată în fișierul *model.py*, în care, în primă fază, se încarcă vectorizarea obținută la pasul anterior.

Un pas important în cadrul antrenării este folosirea tehnicii *one-hot encoding* pentru convertirea etichetelor în vectori conținând 3 elemente astfel: elementul corespunzător clasei respective este setat cu 1, iar restul cu 0.

Totodată, setul de date a fost împărțit în date de antrenare, validare și testare, utilizând funcția *train_test_split()*. Raportul folosit este de 70:10:20.

Pentru a găsi cea mai bună configurație de hiperparametri pe care o poate lua modelul, s-a folosit tehnica **Grid Search**, ce reprezintă un algoritm utilizat în învățarea automată pentru a căuta și selecta cea mai bună configurație a unui model prin explorarea sistematică a unui spațiu predefinit de hiperparametri. Grid search explorează practic fiecare combinație posibilă de valori din grilă și antrenează și evaluează modelul cu aceste valori. Astfel, se va calcula performanța modelului pentru fiecare combinație și se vor compara rezultatele. Configurația care produce cea mai bună performanță, măsurată prin metrici de evaluare, este selectată ca și configurație optimă. Utilizarea Grid Search ajută prin urmare la automatizarea procesului de căutare a hiperparametrilor asigurând cea mai bună alegere în corelație cu setul de date.

Astfel, am definit un dicționar reprezentând spațiul de căutare a celei mai bune combinații de hiperparametri. Cheile dicționarului reprezintă numele hiperparametrilor, iar valorile asociate sunt seturi de valori posibile pentru fiecare dintre aceștia.

Straturile 1, 2 și 3 reprezintă numărul de neuroni din primul, al doilea și al treilea strat. Acestea iau valorile 1024, respectiv 512 neuroni. În general, un număr mai mare de neuroni poate contribui la o capacitate mai mare de învățare a modelului, dar poate duce și la un risc mai mare de overfitting, precum și la creșterea timpului de antrenare.

La această configurație s-au adăugat și două straturi de dropout. Stratul de dropout se adaugă după fiecare strat ascuns (straturile 2 și 3). Valorile alese pentru dropout sunt de 0.1și 0.3. Un dropout de 0.1 va dezactiva temporar aproximativ 10% din contribuția neuronilor de pe stratul pe care a fost aplicat. Acest raționamet va antrena modelul cu diferite configurații ale neuronilor cu scopul de a obține o reprezentare mai robustă, fiind o încercare de evitare a overfitting-ului.

În final ultimii doi hiperparametri adăugați în dicționar sunt clasa de optimizare și rata de învățare. Pentru algoritmul de optimizare am ales două opțiuni: “sgd” (Stochastic Gradient Descent) și “adam” (Adaptive Moment Estimation). Algoritmul SGD este o metodă de optimizare clasică ce presupune actualizarea parametrilor în funcție de gradienții calculați pe subgrupuri ale setului de date, în timp ce Adam folosește niște proceduri mai avansate. Rata de învățare reprezintă cât de mult se vor ajusta parametri modelului în fiecare pas de actualizare, este practic viteza cu care modelul “învață” din datele de antrenare. O rată de învățare mai mică poate duce la o convergență mai lentă, dar poate oferi o soluție mai precisă, în timp ce o rată de învățare mai mare poate duce la o convergență mai rapidă, dar poate determina modelul să sară peste minimele locale ale funcției de pierdere. Rata de învățare pentru acest pas conține două valori posibile: 0.001 și 0.01.

Mai apoi, prin intermediul Grid Search, se va realiza un produs cartezian al tuturor acestor elemente din dicționar, ceea ce rezultă în toate combinațiile posibile. La fiecare iterație se va construi un model cu ajutorul funcției *Sequential* pentru a adăuga straturi de neuroni menționate anterior într-o manieră secvențială. Ultimul strat adăugat este stratul de ieșire, *Dense(3, activation='softmax'),* ce conține 3 neuroni reprezentând cele 3 clase posibile: pozitiv, negativ și neutru, la care se aplică funcția de activare softmax. Aceasta are rolul de a converti un vector de valori într-o distribuție de probabilități, în care fiecare valoare reprezintă probabilitatea unei clase.

În continuare, se va compila acest model specificându-se o serie de parametri. Primul parametru este tipul funcției de pierdere. În acest caz, s-a folosit *categorical_crossentropy*, aceasta fiind adesea utilizată în problemele de clasificare cu mai multe clase pentru a măsura discrepanța între distribuția de probabilități prezise de model și distribuția reală a claselor. Unul dintre obiective este acela de a minimiza valoarea funcției de pierdere, adică, găsirea setului optim de ponderi și parametri care să genereze cele mai bune predicții.

Pentru salvarea ponderilor asociate celei mai bune epoci am folosit un obiect de tipul *ModelCheckpoint*, realizându-se un callback. Antrenarea se pornește cu funcția *fit()*, ce

utilizează datele de antrenare și validare. Pentru acest proces s-au folosit 512 de exemple de antrenare care sunt procesate într-o iterație și 100 de epoci.

Rezultatul căutării celui mai bun set de hiperparametri este reprezentat de configurația: strat1 = 1024 neuroni; strat = 1024 neuroni; strat3 = 1024 neuroni; dropout1 = 0.1; dropout2 = 0.1; clasa de optimizare = 'adam'; rata de învățare = 0.001. Acuratețea finală a modelului este de 69.39%. Această configurație va fi în continuare folosită pentru antrenarea propriu-zisă a modelului.

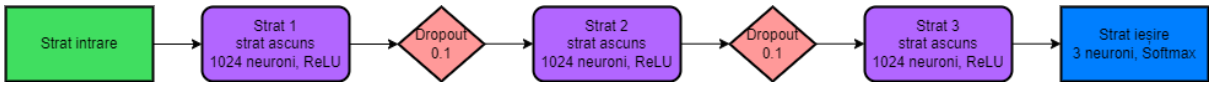

Fig. 16. Structura rețelei neuronale

4.3.3 Rezultate obținute

Unul dintre indicatorii ce ar putea să ne confirme antrenarea corespunzătoare a modelului este reprezentarea funcției de pierdere. Scopul principal al funcției de pierdere (loss function) este să cuantifice eroarea modelului în timpul antrenării, astfel încât să poată fi folosită pentru a ajusta ponderile și parametri modelului.

Pierderea pe antrenare măsoară progresul modelului în timpul procesului de antrenare. În mod ideal, funcția de loss ar trebui să scadă odată cu creșterea numărului de epoci, demonstrând faptul că modelul reușește să generalizeze și să facă predicții mai precise pe datele de antrenare. Același raționament se aplică și pentru funcția de pierdere pe datele de validare. Mai jos, putem observa tendința de scădere a ambelor funcții de pierdere, pe măsură ce modelul învață. Totodată se poate observa și suprapunerea celor două, demonstrând faptul că modelul are rezultate bune pe date noi și că nu are tendință de overfitting.

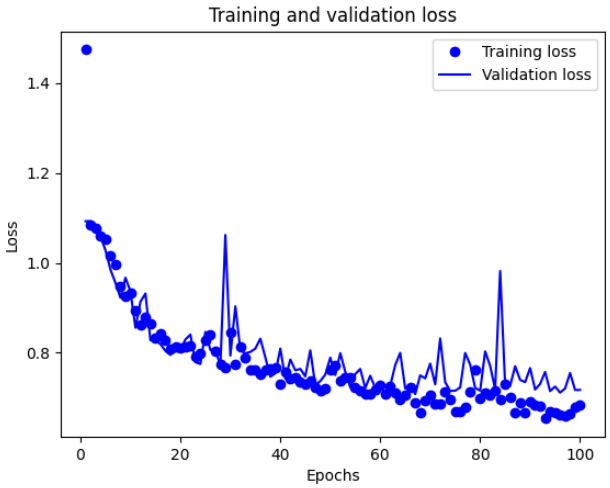

Fig. 17. Funcția de pierdere pentru antrenare și validare

O altă metrică folosită pentru evaluarea performanței unui model de clasificare este acuratețea, reprezentând raportul dintre numărul de exemple clasificate corect și numărul total de exemple (3).

$$acc = \frac{TP+TN}{TP+TN+FP+FN} \quad (3)$$

- TP, TN (True Positive/Negative) – exemple pozitive/negative clasificate corect
- FP, FN (False Positive/Negative) – exemple pozitive/negative clasificate greșit

Pentru mai multe clase, cum este în cazul acestui model, se va calcula această metrică pentru fiecare în parte și se va face o medie aritmetică.

Acuratețea pe antrenare, respectiv pe validare măsoară raportul dintre numărul exemplelor clasificate corect și numărul total de exemple de antrenare/validare. Se poate observa în figura de mai jos faptul că graficele acestora au tendința generală de creștere odată cu procesul de învățare ajungând la acuratețea finală de 69.39%, fapt ce înseamnă că precizia modelului devine din ce în ce mai bună.

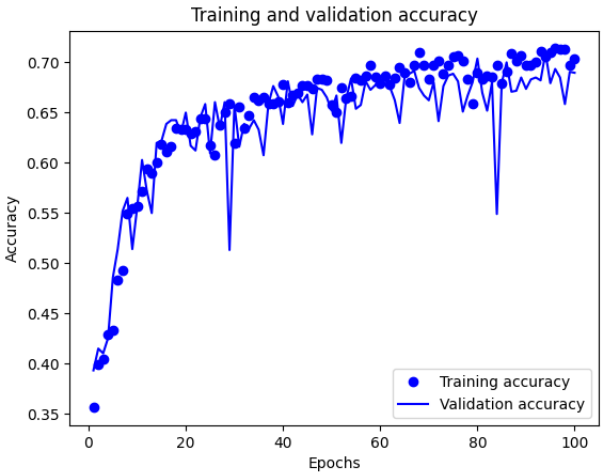

Fig. 18. Acuratețea pentru antrenare și validare

Analizând modelul rezultat și din perspectiva altor metrici, se poate observa faptul că precizia modelului este de 70.03%. **Precizia** măsoară proporția de exemple clasificate corect ca pozitive/negative/neutre din totalul exemplelor pentru fiecare clasă în parte realizându-se mai apoi media aritmetică. Cu cât precizia este mai mare, cu atât modelul apreciază mai puține exemple fals negative în mod eronat.

Recall-ul obținut este de 68.94%, indicând proporția de exemple clasificate corect ca pozitive din totalul exemplelor pozitive reale. Cu cât această metrică este mai mare, cu atât modelul face mai puține erori de clasificare fals negative.

Scorul F1 al modelului are un procent de 69.05%, acesta fiind de fapt echilibrul între precizie și recall, luând în considerare atât erorile fals pozitive, cât și eroile fals negative. Denumirea sa provine din faptul că acesta este calculat folosind media armonică.

Ultima metrică analizată este **AUC** (Area Under the ROC Curve) ce măsoară capacitatea globală a modelului de a face distincția între clase. ROC curve este o reprezentare grafică a performanței unui model de învățare automată ce are la bază dependența între recall și rata de fals pozitiv.

Rezultatele tuturor acestor metrici pe setul de testare pot fi observate în graficul de mai jos.

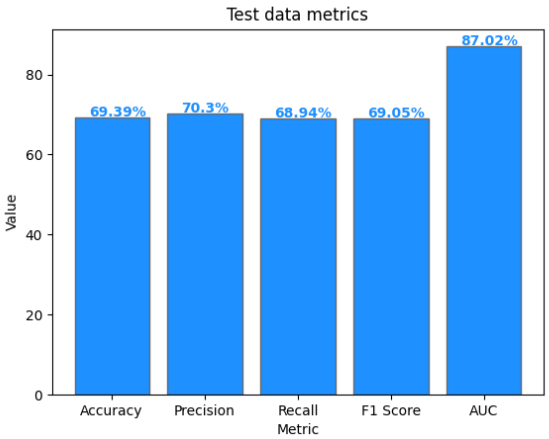

Fig. 19. Rezultatele metricilor modelului

Toate aceste metrici se pot calcula folosind și matricea de confuzie, ce arată performanțele modelului pe setul de date. Elementele acestei matrice sunt de fapt numărul de exemple pentru fiecare categorie. Axa Ox reprezintă eticheta estimată de model, în timp ce axa Oy este eticheta adevărată, astfel încât, pe diagonala principală vom avea clasificările corecte, iar elementele ce se află în afara acesteia vor fi clasificările eronate.

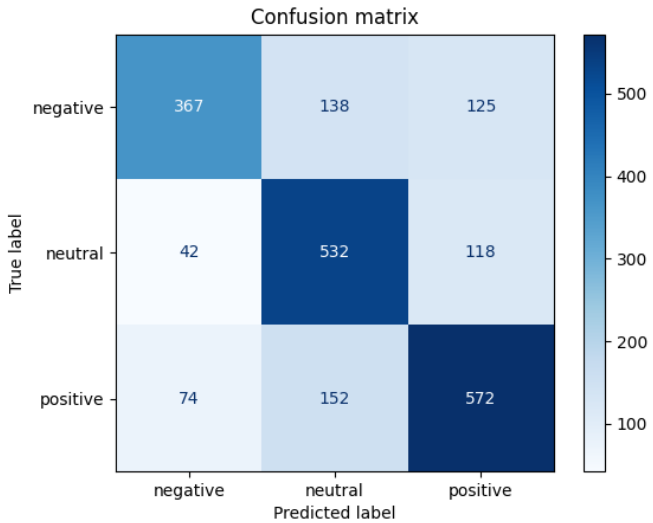

Fig. 20. Matricea de confuzie a modelului

4.3.4 Integrarea modelului în aplicația web

Odată ce modelul a fost antrenat și validat, cu cea mai bună configurație, acesta va fi în continuare utilizat pentru a clasifica automat ultimele știri economice apărute pe Google News. Această clasificare va fi ulterior folosită pentru a calcula un scor pe baza căruia se va face o predicție. Predicția se va contura în funcție de valoarea scorului obținut pe știrile extrase din ultimele două zile, iar pentru că de cele mai multe ori, știrile influențează aproape instant deciziile de investiții, această predicție va avea sens și va fi valabilă pentru următoarele 2-3 zile.

Traseul urmat pentru redarea predicțiilor în aplicația web este compus din următoarele etape:

1. Ștergerea din baza de date a știrilor care nu mai sunt de actualitate.
2. Definirea cuvintelor cheie după care urmează să fie căutate știri (pentru această aplicație s-a definit o listă ce conține 22 de active financiare dintre care 10 sunt criptomonede iar restul de 12 reprezintă acțiuni financiare).
3. Preluarea știrilor de pe Google News ce conțin cuvintele cheie definite, prin intermediul scraping-ului și salvarea acestora într-un fișier output.xlsx ce respectă structura: Titlu știre, Link, Dată, Sursa, Cuvânt cheie, Etichetă.
4. Aplicarea modelului de învățare automată pentru fiecare știre din fișierul creat în etapa precedentă și actualizarea coloanei Etichetă cu rezultatul obținut în urma clasificării (pozitiv/negativ/neutru).
5. Adăugarea știrilor de actualitate împreună cu etichetele generate ale acestora în baza de date.
6. Calcularea scorului pentru fiecare activ financiar în parte, pe baza numărului de știri pentru fiecare clasă în parte:

$$scor = (-1) * nr_{știri\_negative} + (0.1) * nr_{știri\_neutre} + nr_{știri\_pozitive} \quad (4)$$

7. Actualizarea tabelei de Predicții din baza de date cu predicția corespunzătoare scorului astfel: dacă scorul este un număr pozitiv va fi crescătoare, iar dacă este un număr negativ va fi descrescătoare.

În tabelul de mai jos se pot observa rezultatele predicțiilor pentru 4 dintre activele financiare în funcție de scor.

item	positive	negative	neutral	score	last_update	trend
Bitcoin	10	2	13	9.3	2023-06-17 18:02:50	rising
Netflix	1	11	6	-9.4	2023-06-17 18:02:50	falling
Dodgecoin	2	24	9	-21.1	2023-06-17 18:02:50	falling
NVIDIA	30	4	21	28.1	2023-06-17 18:02:50	rising

Tabel 4. Exemple rezultate predicții

5. Studiu de caz

- **Înregistrarea, autentificarea și resetarea parolei**

Procesul de accesare a aplicației presupune înregistrarea unui nou utilizator în aplicația web realizându-se prin intermediul completării unui formular cu: numele de utilizator, e-mail, parola și respectiv confirmarea acesteia. Odată ce aceste câmpuri de tip text au fost completate, se va apăsa butonul "Submit", ce va determina autentificarea în aplicație.

Fig. 21. Pagina de înregistrare a unui utilizator

Această pagină conține și o animație plasată cu scopul de a crea o atmosferă mai captivantă, atrăgând atenția utilizatorului. Efectul pe care îl redă această animație este de afișare a unor mesaje de informare în stilul în care se scrie la o mașină de scris. Textul fiecărui mesaj este afișat treptat, caracter cu caracter, cu o anumită viteză.

Tot în cadrul acestei pagini, se realizează și autentificarea, prin apăsarea butonului de "Login", în cazul în care utilizatorul dispune de un cont în cadrul aplicației. Astfel, procesul de autentificare presupune completarea e-mail-ului asociat contului și parola.

Aplicația dispune și de funcționalitatea resetării parolei prin apăsarea butonului "Reset password" în două etape. Prima etapă presupune solicitarea de resetare a parolei prin completarea numelui de utilizator și a e-mail-ului corespunzătoare contului pentru care se dorește resetarea parolei. Apăsarea butonului de "Submit" va propaga trimiterea unui e-mail la acea adresă cu un cod de validare ce va fi ulterior folosit pentru setarea unei noi parole. Toate aceste formulare conțin și mesaje de eroare în cazul în care utilizatorul introduce un e-mail invalid, sau o parolă ce conține mai puțin de 8 caractere, respectiv date care nu sunt asociate niciunui cont în momentul autentificării.

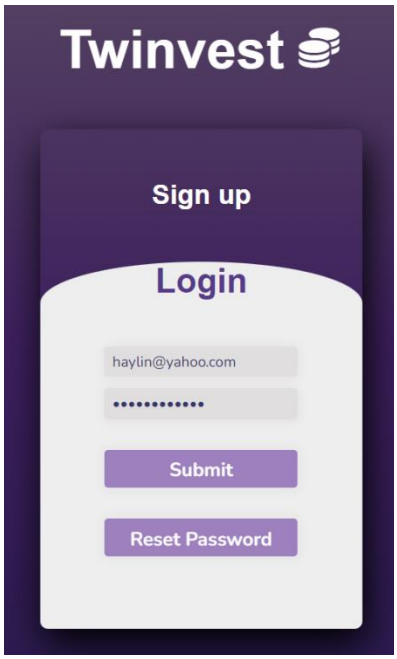

Fig. 22. Formular autentificare

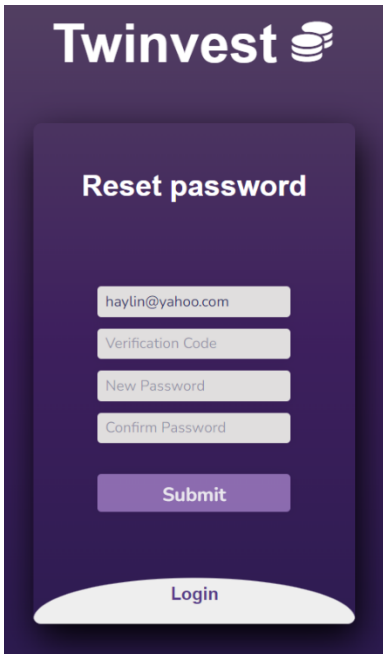

Fig. 23. Solicitare resetare parolă

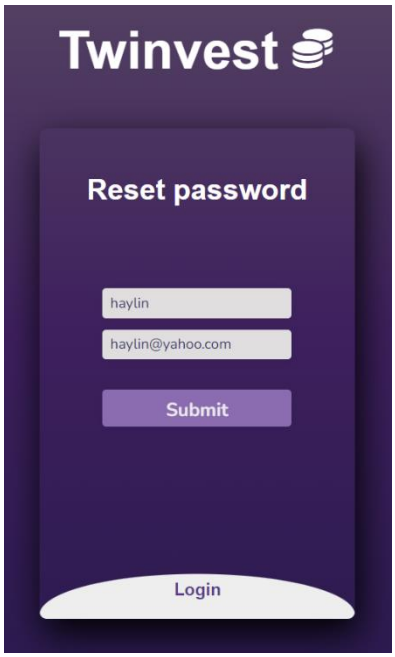

Fig. 24. Resetare parolă

- **Vizualizarea panoului de control**

Odată ce utilizatorul se autentifică în aplicație, acesta va avea acces la panoul de control, de unde va putea vizualiza grafice referitoare la veniturile și plățile sale, precum și un istoric al tranzacțiilor. Graficele au rolul de a afișa atât evoluția fluxului de bani în cont, dar și distribuția veniturilor și plăților pe categorii.

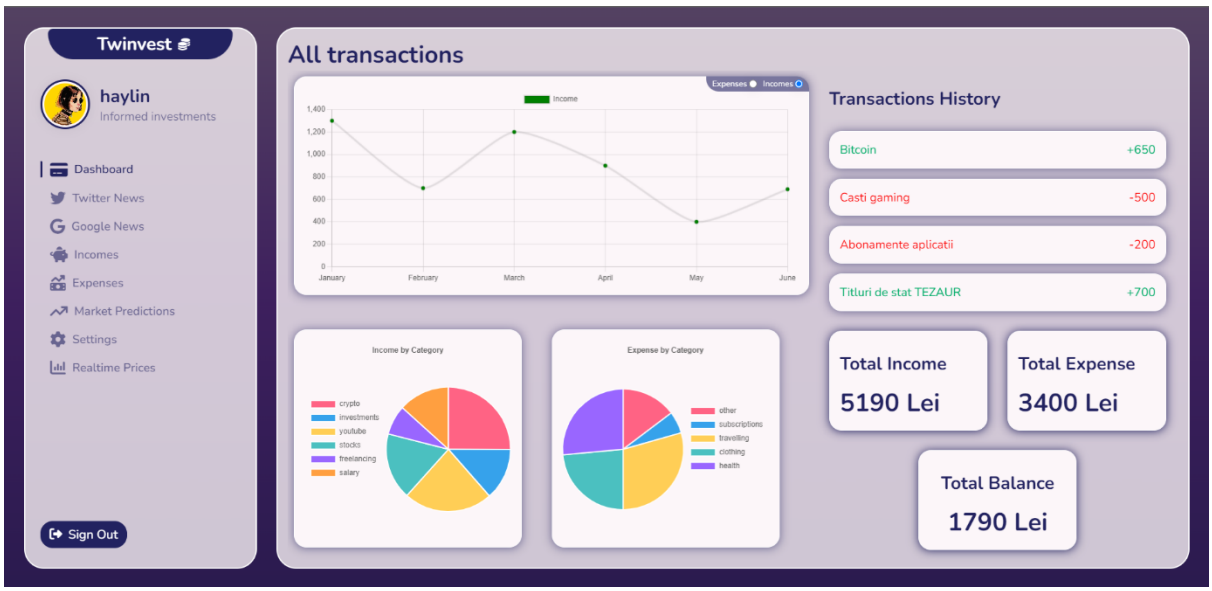

Fig. 25. Panoul de control

Tot în această pagină se află și meniul aplicației, pe partea stângă, de unde va avea acces și la celelalte funcționalități.

Statisticile pentru venituri și plăți sunt reprezentate prin 4 grafice astfel: 2 grafice liniare prin care se poate comuta pe baza selecției ce arată sumele achitate/încasate pe parcursul fiecărei luni și 2 grafice circulare împărțite pe sectoare, ce au rolul de a arăta distribuția categoriilor de venituri/plăți.

- **Adăugarea veniturilor și plăților în cont**

Pentru adăugarea unui venit sau a unei plăți în cont, utilizatorul poate face acest lucru din paginile Incomes și Expenses, prin completarea unor formulare.

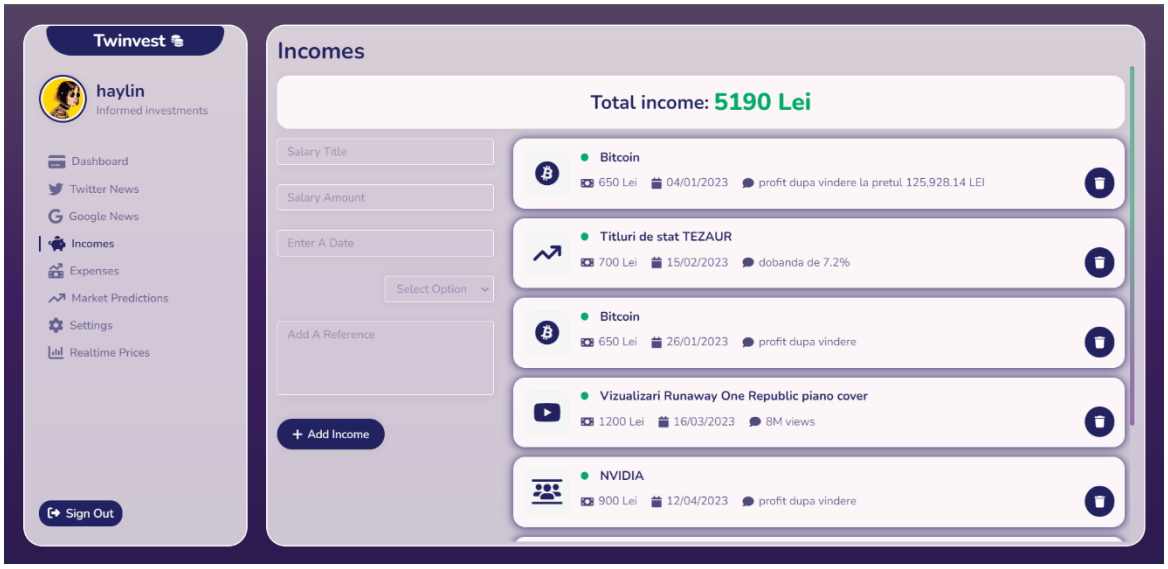

Fig. 26. Pagina de introducere a veniturilor

Astfel, pentru fiecare din cele două categorii, va fi necesară specificarea unei denumiri, a unei sume de bani, a datei pentru tranzacția respectivă, o categorie ce va fi aleasă dintr-o listă predefinită, și o descriere care este opțională.

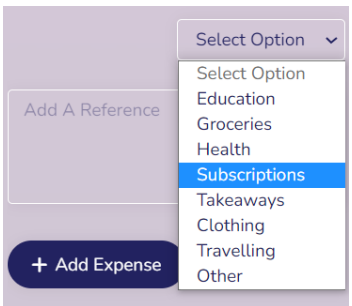

Fig. 27. Selectarea unei categorii de plată

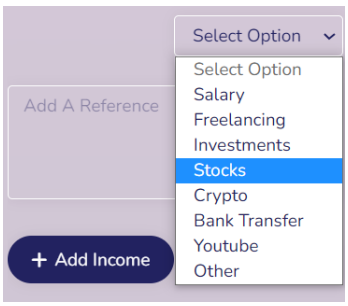

Fig. 28. Selectarea unei categorii de venit

Adăugarea în cont a tranzacției se va realiza prin apăsarea butonului “+ Add Expense” sau “+ Add Income”. Necompletarea tuturor câmpurilor obligatorii va genera o eroare sugestivă în dreptul câmpului care trebuie completat.

Cele două pagini pentru adăugarea tranzacțiilor au o arhitectura similară, totalul plăților/veniturilor fiind afișat pe centru, în partea de sus, sub forma: “Total Expense/Income”

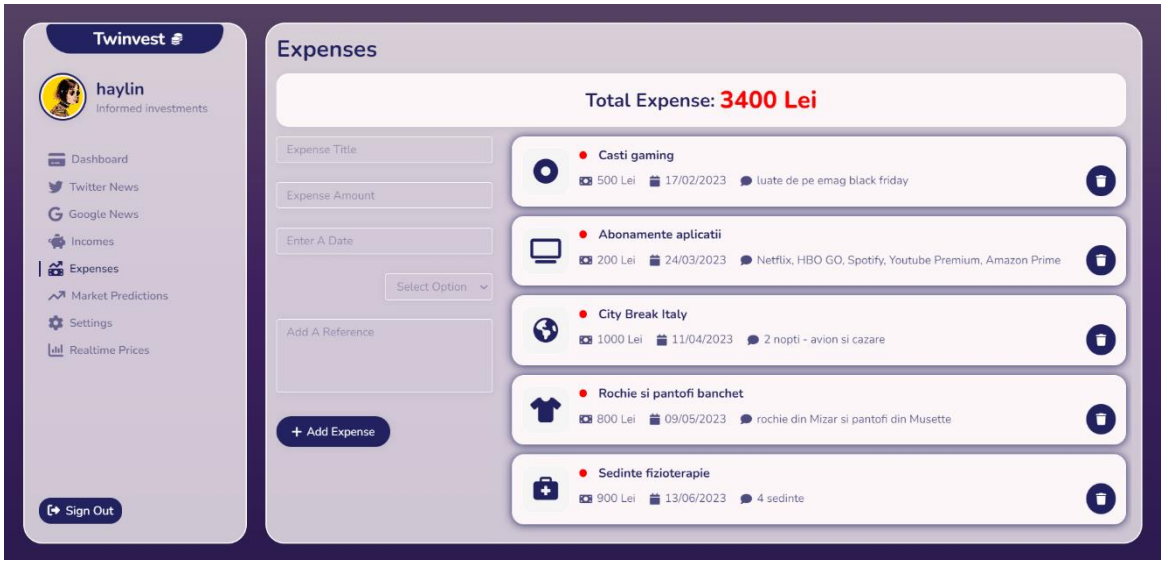

Fig. 29. Pagina de introducere a plăților

Prin această acțiune, tranzacția va fi adăugată într-o listă sub forma unui card pe care sunt afișate toate informațiile. În mod intuitiv, cardurile vor conține câte un simbol diferit în funcție de categoria de venit/plată aleasă. Aceste carduri pot fi șterse prin apăsarea butonului situat în extremitatea dreaptă a acestuia.

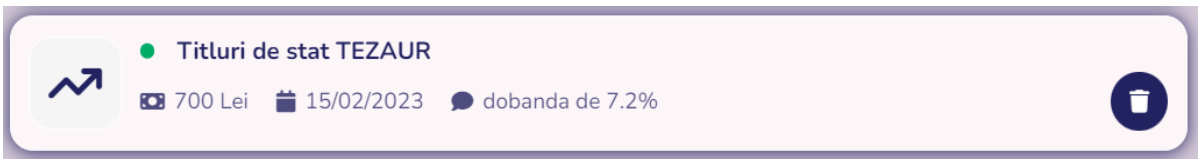

Fig. 30. Exemplu card venit

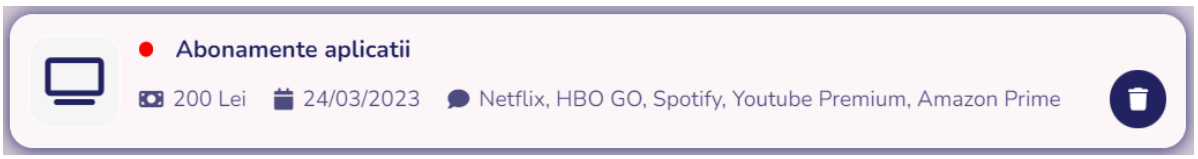

Fig. 31. Exemplu card plată

- **Panoul de Setări**

Prin intermediul paginii de Setări, utilizatorul are posibilitatea, în primul rând de a-și schimba parola contului în cadrul aplicației completând parola nouă și confirmarea acesteia. Odată ce se apasă pe butonul “Submit”, în secțiunea “Change your password”, parola va fi schimbată.

Totodată, o funcționalitate importantă, ce determină vizualizarea știrilor personalizate direct din aplicație, este secțiunea în care utilizatorul își poate alege criptomonedele și acțiunile la bursă pe care le urmărește, dintr-o listă predefinită ce conține 22 de active financiare. Acest lucru se poate face în cadrul secțiunii “Pick your interests”. În funcție de elementele ce au fost selectate în această secțiune, utilizatorul va primi știri și informații referitoare la evoluția acestor active financiare. De asemenea, în dreptul fiecărui instrument economic se află un buton de ștergere a acestuia folosit pentru a-l elimina din listă.

Secțiunea “Set your smart goals!” oferă posibilitatea de a seta un obiectiv de venit sau a unei limite de cheltuire. Astfel, odată ce acestea au fost setate, cele două bare de progres vor reda procentual unde se situează totalul veniturilor și al plăților în raport cu așteptările utilizatorului. Această acțiune va determina afișarea unor mesaje de informare și avertizare în momentul în care utilizatorul va adăuga alte plăți și venituri.

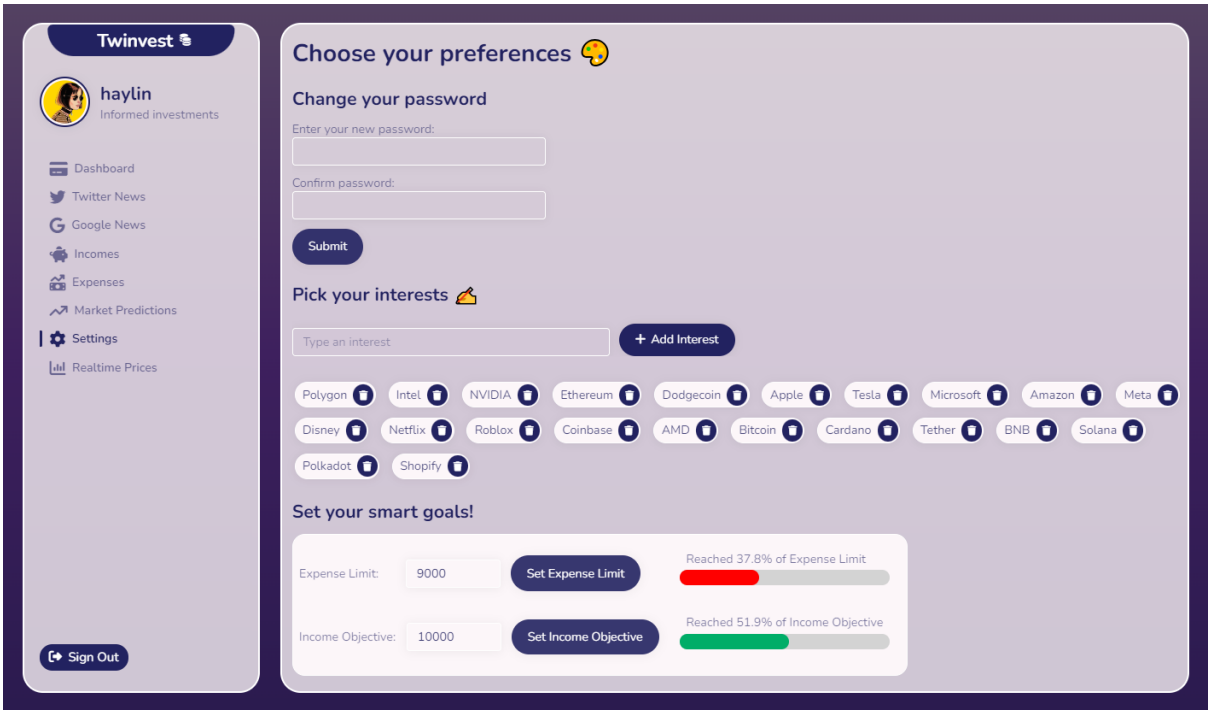

Fig. 32. Pagina de setări

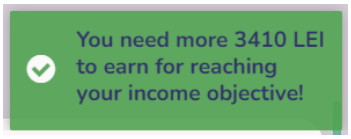

Fig. 33. Mesaj informare adăugare venit

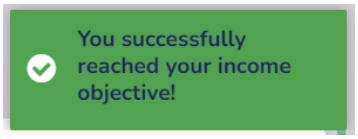

Fig. 33. Mesaj informare finalizare obiectiv

Fig. 35. Mesaj avertizare adăugare plată

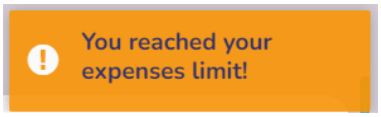

Fig. 36. Avertizare depășire limită cheltuieli

- **Vizualizarea postărilor de pe Twitter**

În cadrul paginii “Twitter News” utilizatorul poate vizualiza postările de pe Twitter având cuvinte cheie ce se află în lista de preferințe pe care acesta și-a configurat-o în meniul de setări. Astfel, are acces la ultimele informații și discuții ce pot influența piața financiară. Fiecare postare are asociat link-ul de Twitter, astfel că utilizatorul poate apăsa pe acestea pentru a fi redirecționat către sursă.

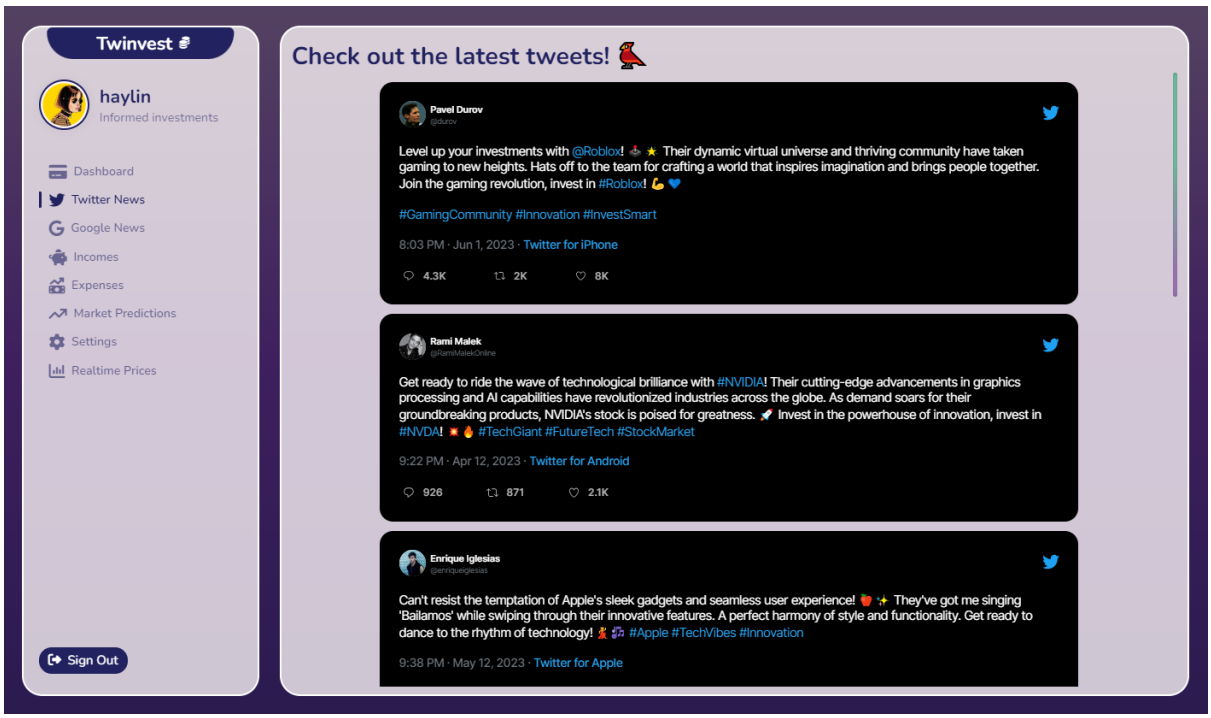

Fig. 37. Pagina de postări Twitter

- **Vizualizarea știrilor Google News etichetate**

Pagina de “Google News” oferă utilizatorului posibilitatea de a vizualiza toate știrile preluate de pe platforma Google News, ce conțin cuvintele cheie marcate ca domeniu de interes, similar cu pagina prezentată anterior. Fiecare știre este afișată sub form unui card ce conține titlul știrii, cuvântul cheie, ce poate fi o criptomonedă sau o acțiune la bursă, data apariției acelei știri, sursa acesteia precum și eticheta, ce poate fi: pozitiv, negativ sau neutru, în funcție de cum a apreciat modelul de învățare automată. Totodată, utilizatorul poate apăsa pe titlul fiecărei știri pentru a fi redirecționat către website-ul sursă.

Pe baza etichetării acestor știri se va calcula scorul final pentru fiecare activ financiar și se vor putea vizualiza predicțiile.

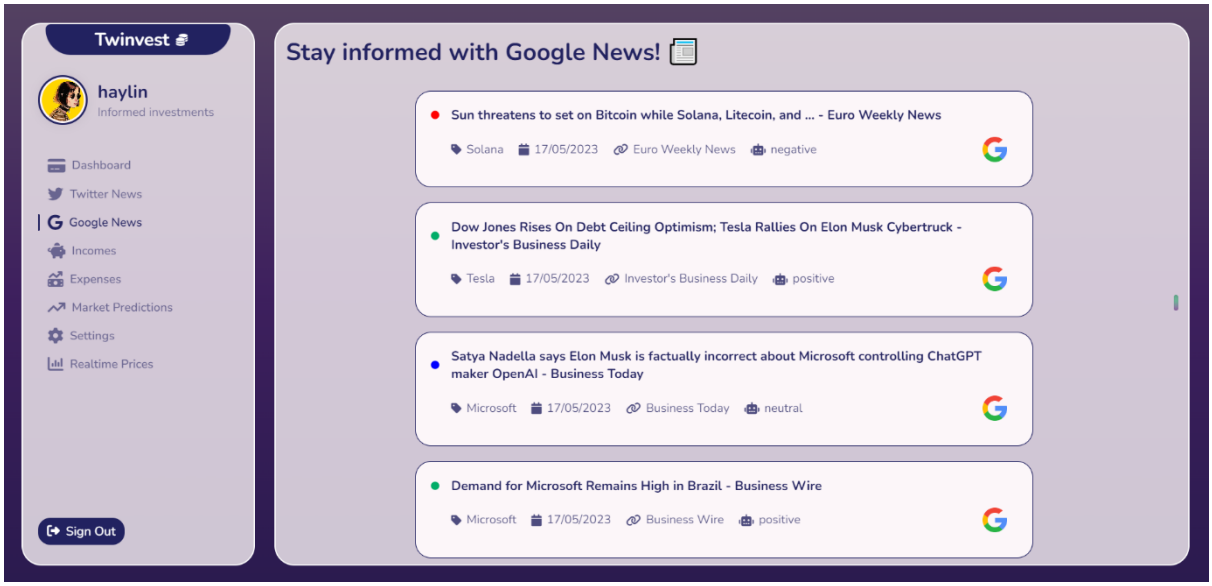

Fig. 38. Pagina de știri Google News

- **Vizualizarea predicțiilor pentru activele financiare urmărite**

Pagina de predicții a evoluției prețurilor se poate accesa din “Market Predictions”. Aici, predicțiile pentru activele financiare, calculate pe baza etichetării automate a știrilor în funcție de sentimentul pe care îl transmit, sunt redate intuitiv, prin intermediul a două tipuri de săgeți. Săgeata roșie îndreptată în jos indică faptul că prețul instrumentului financiar asociat va fi în scădere în următoarele două zile, iar cea verde îndreptată în sus, indică faptul că va fi în creștere. Totodată este afișată și data în care s-a efectuat ultima predicție. În cazul în care nu au fost găsite știri relevante pentru un anumit activ economic, simbolul va fi reprezentat sub forma unei linii drepte încadrată de un cerc albastru.

În partea de jos a acestei pagini este descrisă și o scurtă legendă pentru fiecare simbol.

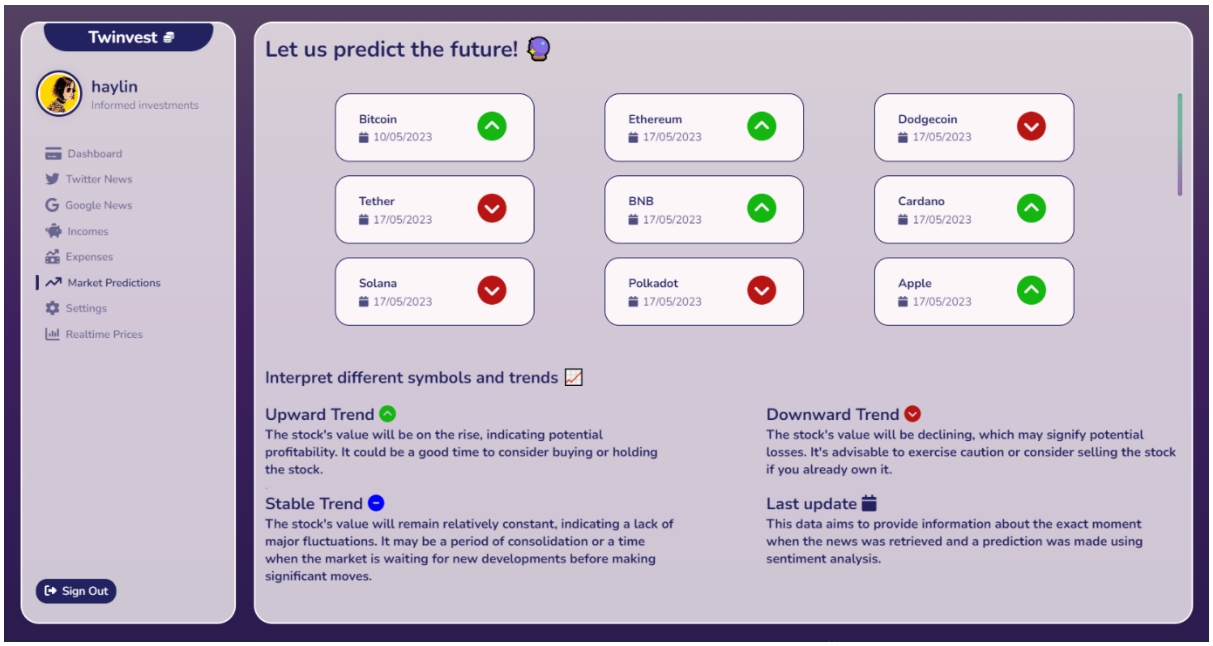

Fig. 39. Pagina de predicții a evoluției prețurilor

- **Evoluția în timp real a activelor financiare urmărite**

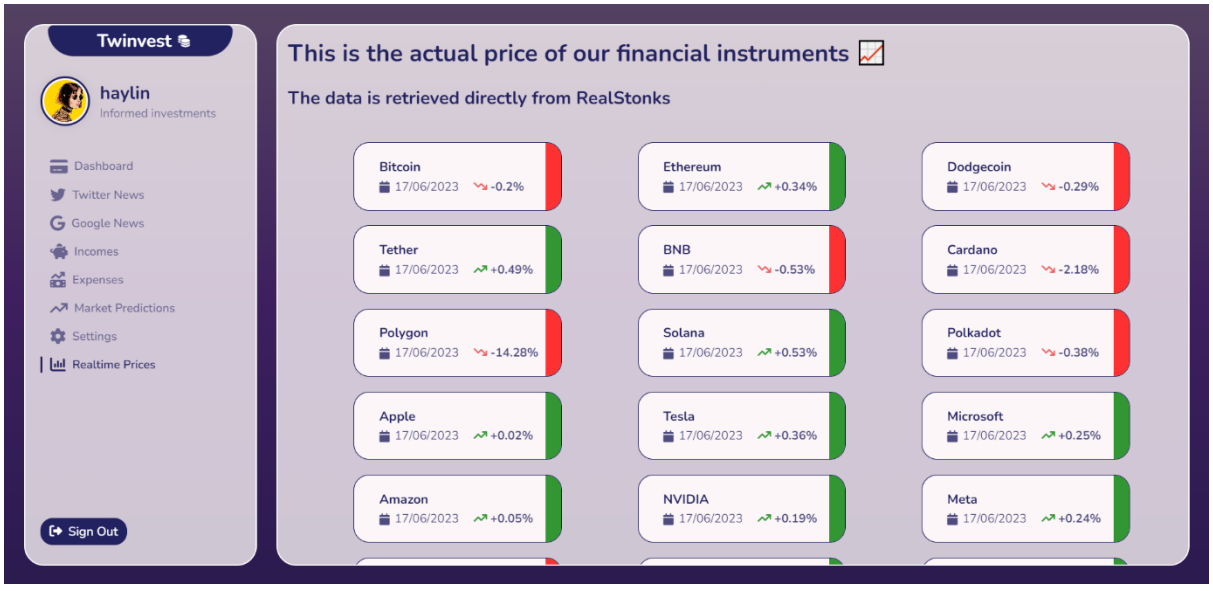

Fig. 40. Pagina de evoluție în timp real a prețurilor

Prin intermediul paginii “Realtime Prices”, utilizatorul poate vizualiza evoluția în timp real sub formă procentuală a fiecărui element de investiție din lista acestuia. Elementele sunt marcate sugestiv cu roșu dacă prezintă o scădere și cu verde dacă sunt în creștere.

6. Concluzii

Obiectivul aplicației dezvoltate în cadrul proiectului de diplomă este acela de a analiza datele din domeniul economic și de a realiza predicții relevante prin intermediul integrării unui model de inteligență artificială ce este antrenat să clasifice știrile în funcție de sentimentul pe care acestea îl transmit. De asemenea, un scop secundar al acestei aplicații web este monitorizarea atentă a bugetului. Astfel, cele două obiective se suprapun în momentul în care utilizatorul intenționează să plaseze o investiție.

Având în vedere analiza asupra lucrărilor similare în domeniu și compararea acestora cu aplicația Twinvest, se poate afirma faptul că soluția abordată oferă o perspectivă clară asupra piețelor financiare, promovând deciziile autonome în materie de investiții.

Aplicația dezvoltată are la bază contribuții personale semnificative, care au condus la îndeplinirea obiectivului propus. În primul rând, am identificat potențialul sinergic al integrării inteligenței artificiale în domeniul economic, înțelegând avantajele pe care aceasta le poate aduce, iar, prin studiul aprofundat al aplicațiilor similare din domeniul de activitate ales, am putut identifica aspectele cheie necesare unei aplicații.

Prin proiectarea arhitecturii generale am asigurat integrarea procesării limbajului natural în cadrul aplicației web. Am identificat tehnologiile adecvate și am ales un set de date relevant din domeniul economic pentru antrenarea modelului de inteligență artificială. Astfel, în cadrul acestui proiect a fost aplicat cu succes un model de învățare automată, ce are o acuratețe finală de 69.39%, pentru clasificarea știrilor financiare, utilizând cuvinte cheie pentru a extrage informațiile relevante, oferind utilizatorilor conținut personalizat. Această acuratețe poate fi îmbunătățită semnificativ prin accesul la un set de date mai complex ce se focusează în special pe companiile și criptomonedele listate în aplicație.

Pentru a stoca și gestiona informațiile necesare, am stabilit structura generală a bazei de date, dezvoltând un mecanism prin care se extrag postări și știri relevante din domeniul economic, oferind utilizatorilor conținut personalizat în funcție de preferințe. Totodată, un alt aspect important al aplicației pe care l-am implementat este capacitatea sa de a realiza predicții pe baza analizei știrilor.

În final, aplicația include și funcționalități legate de monitorizarea bugetului personal, permițând utilizatorilor să își gestioneze finanțele în mod eficient și să aibă o vedere de ansamblu asupra cheltuielilor și veniturilor. Toate aceste contribuții personale semnificative au fost integrate coerent, rezultând o soluție completă și utilă pentru utilizatorii din domeniul economic.

O dezvoltare ulterioară importantă ce ar contribui semnificativ la acuratețea calculării predicțiilor ar consta în integrarea unui mecanism de verificare și validare a surselor precum și a conținutului știrilor pentru a asigura faptul că utilizatorii primesc informații exacte și de încredere.

Totodată, o altă funcționalitate ce ar determina crearea unei comunități asociate aplicației, ar fi integrarea funcționalității de socializare, astfel încât investitorii pot interacționa pentru a-și împărtăși experiențele în materie de investiții cu scopul de a învăța unii de la alții.

În ceea ce privește gestionarea contului de economii, o altă funcționalitate utilă ar putea fi partajarea situației bugetului cu rudele, prietenii sau partenerii de afaceri. Această opțiune ar consolida colaborarea între persoanele implicate ce doresc să își îndeplinească același obiectiv financiar.

Particulatizând toate aceste aspecte, se poate afirma faptul că dezvoltarea aplicației Twinvest demonstrează potențialul vast al intersecției între domeniul economic și cel al tehnologiilor emergente, cum ar fi inteligența artificială.

7. Referințe

[1] R. p. Grégoire, „The Crypto Boom: Challenges and Risks,” 30 June 2022.

[2] „The global crypto market cap,” [Interactiv]. Available: https://coinmarketcap.com/coins/. [Accesat Iunie 2023].

[3] „BIZIDAY,” [Interactiv]. Available: https://www.biziday.ro/245063-2/?utm_source=rss&utm_medium=rss&utm_campaign=245063-2. [Accesat 7 Iunie 2023].

[4] „Empower,” [Interactiv]. Available: https://www.empower.com/.

[5] „AlphaSense,” [Interactiv]. Available: https://www.alpha-sense.com/.

[6] „Kavout,” [Interactiv]. Available: https://www.kavout.com/.

[7] S. Aryal, „MERN STACK WITH MODERN WEB PRACTICES,” 2020.

[8] „SimpliLearn,” [Interactiv]. Available: https://www.simplilearn.com/tutorials/nodejs-tutorial/what-is-nodejs. [Accesat 10 Iunie 2023].

[9] „SimpliLearn,” [Interactiv]. Available: https://www.simplilearn.com/tutorials/reactjs-tutorial/what-is-reactjs. [Accesat 11 Iunie 2023].

[10] „npm,” [Interactiv]. Available: https://www.npmjs.com/.

[11] „HackerNoon,” [Interactiv]. Available: https://hackernoon.com/integrating-redux-to-your-react-app-a-step-by-step-guide-x02z32jb/. [Accesat 11 Iunie 2023].

[12] „DeepChecks,” [Interactiv]. Available: https://deepchecks.com/glossary/grid-search/. [Accesat 11 Iunie 2023].

[13] E. D. Liddy, Natural Language Processing, 2001.

[14] „DeepSet,” [Interactiv]. Available: https://www.deepset.ai/blog/what-is-text-vectorization-in-nlp. [Accesat 13 Iunie 2023].

[15] [Interactiv]. Available: https://towardsdatascience.com/tf-idf-for-document-ranking-from-scratch-in-python-on-real-world-dataset-796d339a4089. [Accesat 13 Iunie 2023].

[16] [Interactiv]. Available: https://towardsdatascience.com/bert-explained-state-of-the-art-language-model-for-nlp-f8b21a9b6270. [Accesat 13 Iunie 2023].

[17] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones și A. N. Gomez, „Attention Is All

You Need,” 2017.

[18] [Interactiv]. Available: https://towardsdatascience.com/deconstructing-bert-part-2-visualizing-the-inner-workings-of-attention-60a16d86b5c1. [Accesat 13 Iunie 2023].

[19] [Interactiv]. Available: https://huggingface.co/blog/bert-101#241-how-do-transformers-work. [Accesat 13 Iunie 2023].

[20] [Interactiv]. Available: https://www.turing.com/kb/how-bert-nlp-optimization-model-works. [Accesat 13 Iunie 2023].

[21] [Interactiv]. Available: https://www.baeldung.com/cs/neural-networks-neurons. [Accesat 14 Iunie 2023].

[22] „Twitter API,” [Interactiv]. Available: https://developer.twitter.com/en/docs/twitter-api. [Accesat 15 Iunie 2023].

[23] „RealStonks API,” [Interactiv]. Available: https://rapidapi.com/amansharma2910/api/realstonks. [Accesat 15 Iunie 2023].

[24] „Hugging Face - FinancialPhraseBank dataset,” [Interactiv]. Available: https://huggingface.co/datasets/financial_phrasebank. [Accesat 16 Iunie 2023].

[25] „NASDAQ OMX Helsinki,” [Interactiv]. Available: https://www.nasdaqomxnordic.com/news/marketnotices/helsinki?.

[26] A. Sinha, P. Takala, P. Korhonen și M. Pekka, Good Debt or Bad Debt: Detecting Semantic Orientations in Economic Texts, 2013.

[27] „Deep Talk,” [Interactiv]. Available: https://www.deep-talk.ai/post/history-and-present-of-natural-language-processing. [Accesat 12 Iunie 2023].