UNIVERSITATEA DIN
BUCUREȘTI

FACULTATEA DE
MATEMATICĂ ȘI
INFORMATICĂ

SPECIALIZAREA INFORMATICĂ

Lucrare de licență

PREDICȚIA PREȚULUI ACȚIUNILOR NETFLIX INC

Absolvent
Dilirici Mihai

Coordonator științific
Lect.dr. Bogdan Dumitru

București, iunie 2024

Rezumat

Prezicerea prețului acțiunilor este unul din cele mai cercetate subiecte din ultimii ani, iar odată cu apariția inteligenței artificiale, au apărut și numeroși algoritmi ce încearcă să obțină rezultate tot mai exacte. Acestă lucrare analizează comportamentul acțiunilor Netflix și explorează diferite metode de învățare automată pentru anticiparea valorilor din viitor.

Pentru a realiza aceste predicții, vom efectua o Analiză Exploratorie a Datelor (EDA) ce ne va îndruma mai apoi în procesul de antrenare. Vom implementa în limbajul Python mai multe modele clasice precum Regresia Liniară, Regresia Lasso, dar și o rețea neuronală de tip Memorie pe Termen Lung și Scurt pentru a integra metode de învățarea adâncă.

Rezultatele vor fi vizualizate folosind librării specializate, comparând performanțele fiecărui model, dar și optimizările făcute fiecărei arhitecturi pe parcurs.

Abstract

Stock price prediction is one of the most researched topics in recent years, and with the advent of artificial intelligence, numerous algorithms have appeared, trying to obtain more and more accurate results. This paper analyzes the behaviour of Netflix shares and explores different machine learning methods for anticipating future values.

In order to make these predictions, we will perform an Exploratory Data Analysis (EDA) which will then guide us in the training process. We will implement in the Python language several classic models such as Linear Regression, Lasso Regression, as well as a Long Short-Term Memory neural network to integrate deep learning methods.

The results will be visualized using specialized libraries, comparing the performances of each model, but also the optimizations made to each architecture along the way.

Cuprins

Listă de figuri

Capitolul 1

Introducere

Piața de acțiuni este un mediu care atrage tot mai mulți oameni și companii dornice de profit prin faptul că induce senzația de profit instantaneu sau pe termen lung. Multe elemente influențează în mod constant prețul acțiunilor unei companii, precum factori politici, încrederea în managementul și funcționarea firmei sau chiar știri legate de aceasta [1].

Avansarea tehnologiei permite publicului să acceseze o cantitate tot mai mare de informații, ceea ce face munca unui analist sau chiar a unei echipe, tot mai dificilă. Totuși, progresul poate fi și în favoarea lor, dacă aceștia aplică metode noi și folosesc puterea de procesare a unui calculator modern.

1.1 Scopul lucrării

În această lucrare am studiat comportamentul acțiunilor Netflix din ultimii ani, comparându-l cu alți membri FAANG (Facebook, Amazon, Apple, Netflix, Google), dar și cu o companie care realizează conținut similar, The Walt Disney.

Scopul lucrării este de a prezice prețul acțiunilor al lui Netflix Inc, antrenând modele ce utilizează date despre trecutul acestuia, dar și despre celelalte companii.

Prin această analiză, doresc să identific factorii principali ce determină dinamica acestor prețuri din sectorul tehnologic, explorând interacțiunile dintre marile companii prin diverse metode moderne. Voi folosi diferite modele de învățare automată, țintind la un rezultat cât mai bun, care va putea fi de folos într-un plan de investiție în acțiunile acestor corporații.

1.2 Obiective

În vederea realizării acestei lucrări, am stabilit următoarele obiective:

- **Analiza datelor**: Preprocesarea datelor are un rol crucial în performanța modelului și în identificarea de relații între diverse companii.
- **Implementarea de algoritmi eficienți**: Utilizarea de librării Python pentru a extrage date despre companii și folosirea de funcții predefinite care ușurează crearea unui model.
- **Obținerea unui model performant**: Experimentarea cu diferite modele de învățare automată și analizarea rezultatelor finale.

1.3 Motivația

Am ales această lucrare deoarece piața financiară este tot mai dependentă de inteligență artificială și învățare automată. Companiile precum Netflix, Google, Meta pun la dispoziție seturi de date mari ce pot fi analizate în predicția de stocuri în mod automat, obținând un avantaj competitiv față de investitorii de rând.

Un model bine structurat nu doar că oferă informații prețioase celor care vor să se lanseze în piața de acțiuni, dar pot servi și celor care folosesc metode clasice, prin faptul că analizează o cantitate foarte mare de date într-un timp scurt.

1.4 Scurt istoric al predicțiilor de stoc

1.4.1 Metode tradiționale de ML

Într-un studiu realizat de IEEE în 2018 [2], autorii au reușit să determine care este cel mai potrivit model dintre Random Forest, Support Vector Machine, Naive Bayes, K-Nearest Neighbor și Softmax. Aceștia au examinat diferiți indicatori de performanță, analizând date din mai multe surse, precum Yahoo și NSE India.

Concluzia a fost că modelul Random Forest a obținut cele mai bune rezultate pentru seturi de date mari, iar pentru cele mici, metoda Naive Bayes. O observație importantă din această lucrare este că numărul de indicatori tehnici (MA, RSI) este direct corelat cu o performanță mai bună a modelelor.

În această analiza s-au folosit ca date doar istoricul prețului acțiunilor GOOGL, însă s-a observat că această tehnică nu oferă cele mai bune rezultate, deoarece această valoare poate să fluctueze foarte mult din diferite motive. Unele modele se descurcă mai bine doar cu date istorice, iar altele au nevoie de analize de sentiment sau diferite statistici pentru a oferi un rezultat mai bun. În final, autorii recomandă experimentarea și cu alte

modele mai performante, deoarece acești algoritmi au o sensibilitate mare la erori și date inconsistente [3].

1.4.2 Metode Deep Learning

În urma unei analize despre influența datelor de intrare în predicția de stoc, două studente de la Universitatea Sookmyung au observat că dintr-un set mare de date, doar câteva au fost relevante în calcularea rezultatelor. Au studiat apoi 3 tipuri de rețele neuronale pentru a stabili care este cea mai potrivită, iar cea care folosește caracteristici binare a ieșit câștigătoare în fața trăsăturilor multiple sau tehnice. Totuși, reducerea la acest tip de date cu valori de 0 și 1 are limitările ei, deoarece se pot ignora informații prețioase în predicție [4].

Un alt studiu despre influența indicatorilor în predicția stocurilor pune în valoare modele moderne precum RNN și LSTM în favoarea celor tradiționale, deoarece acestea reușesc să identifice relații dintre variabile și cresc performanța. De asemenea, crearea unui model hibrid este o altă tehnică prețioasă ce oferă rezultate mai exacte. Cu toate acestea, ele nu pot prezice valori dintr-un viitor mai îndepărtat, rezultatele fiind tot mai slabe odată cu înaintarea în timp [5].

Capitolul 2

Noțiuni preliminare

2.1 Inteligență artificială, Învățare automată

Inteligența artificială (AI) este un domeniu larg al informaticii care se ocupă cu dezvoltarea de programe software ce mimează gândirea umană. Un astfel de program poate să învețe, să se adapteze și să își corecteze propriile greșeli. În alți termeni, inteligența artificială este o extensie a inteligenței umane, prin intermediul calculatoarelor, așa cum puterea fizică a omului a fost amplificată prin folosirea de instrumente mecanice [6] .

Machine Learning (ML) este o categorie a inteligenței artificiale care dă libertate calculatorului să gândească și să învețe pe cont propriu, fără a fi nevoie de instrucțiuni clare. Termenul de ML a fost conceput în 1959 de Arthur Samuel, iar primele programe au fost folosite pentru statistici și probabilități.

Simularea inteligenței prin modele de învățare automată este strâns corelată de o altă ramură importantă în dezvoltarea acestei lucrări, anume Statistică Computațională, scopul acesteia fiind realizarea de predicții prin intermediul computerelor. Problemele din lumea reală au o complexitate foarte mare, ceea ce face ML un candidat excelent în rezolvarea acestora: filtrarea mesajelor spam, detectarea de fraude online, diagnosticarea medicală, dar și ceea ce voi prezenta în continuare, predicția de acțiuni online [7].

În ultimii ani, algoritmii de ML au fost tot mai răspândiți în analiza datelor economice și au avut un impact major în luarea deciziilor de investiție. De exemplu, Chou și Nguyen (2018) au prezis stocul companiilor de construcție din Taiwan, iar alți investitori precum Jeong et al. (2018) s-au folosit de aceste tehnici pentru a face investiții mari, utilizând date din știri financiare și social media [8].

2.2 Serii de timp, Analiză exploratorie a datelor

Seriile de timp (Time Series) sunt grupuri de date consecutive, colectate pe mai multe ore, zile, luni sau ani, și este forma generală a datelor în domeniul financiar. Acestea pot fi fi împărțite în 4 componente:

- Trend - Tendința de creștere, descreștere sau stabilitate a datelor pe termen lung;
- Ciclu - Apariția unui tipar în fluctuațiile seriilor, de obicei pe perioade de peste un an;
- Sezonalitate - Repetarea unui fenomen în același anotimp a mai multor ani;
- Iregularitate - Variație a datelor pe termen scurt, datorată unor întâmplări precum războiul sau dezastre naturale [8].

Analiza datelor în mod experimental (EDA) este, conform John W. Tukey (1977), o muncă de detectiv al numerelor. Altfel spus, o analiză de date unde cercetătorul studiază comportamentul datelor fără a avea o idee preconceptă asupra rezultatului pe care îl va descoperi. Pe timpul când Tukey și-a scris cartea despre EDA, calculatoarele nu erau foarte răspândite, iar seturile de date erau foarte mici, comparativ cu standardele de astăzi.

În proiectele recente, EDA este folosită pentru a identifica, prin puterea mașinilor de calcul moderne, diverse tipare și anomalii în seturile de date, dar și de a vizualiza statistici despre acestea. Pentru a încorpora EDA, trebuie să respectăm doi pași:

- Alegerea unei idei de început;
- Crearea unui design prin folosirea de structuri întrebare-răspuns.

Primul pas este ușor de urmat, însă al doilea necesită urmărirea a două principii importante: **scepticism** și **deschidere la idei** (open mind). Câteva exemple de întrebări sunt date chiar în cartea lui Tukey:

- Pot datele curente să confirme ipoteza?
- Ce pot să îmi spună datele despre relația dintre XYZ?
- Cât de probabil este ca un design să ofere răspunsuri utile?

Uneori, această analiză este văzută ca ”o artă” sau un set de trucuri, mai degrabă decât o știință. Deși nu este nimic ascuns despre tehnicile EDA, analistul trebuie să încerce multe metode, să adopte multe idei și să fie pregătit pentru surprize [9].

2.3 Modele utilizate, State of the art

În prezent, există o mulțime de algoritmi și tehnici ce pot fi aplicate pentru a prezice viitorul acțiunilor unei companii. Majoritatea dintre acestea pot fi antrenate pe serii de timp, însă fiecare are avantaje și dezavantaje.

1. **Linear Regression**
   Este un model fundamental în statistică și învățarea automată, fiind simplu de folosit și înțeles. Formulă generală pentru o regresie liniară simplă este:

$$y = \beta_0 + \beta_1 x + \epsilon$$

   unde:

   - y este variabila dependentă,
   - x este variabila independentă,
   - β_0 este intercepția liniei de regresie pe axa Y,
   - β_1 este panta liniei de regresie, indicând efectul x asupra y,
   - ϵ reprezintă eroarea, reflectând toți factorii ce influențează y în afară de x.

   [10]

   Acest model se bazează pe 4 ipoteze :

   (a) Liniaritate : Relația dintre variabilele independente și cele dependente este liniară.

   (b) Independență : Observațiile sunt independente unele de altele.

   (c) Homoscedasticitate: Variabilitatea reziduului (diferența dintre variabilele observate și cele prezise) este aceeași pentru orice valoare a variabilelor independente.

   (d) Normalitate : Pentru orice valoare atribuită unei variabile independente, variabila dependentă are o distribuție normală.

   [11] .

   Avantaje:

   - Folosește concepte de bază, ușor de interpretat.
   - Este folosit ca model de bază în dezvoltarea celor mai complexe.
   - Necesită un timp scurt de antrenare.

   Dezavantaje:

- Performanță slabă atunci când ipotezele sunt încălcate.
- Poate fi sensibil la date eronate.

2. **Lasso Regression** - Least Absolute Shrinkage and Selection Operator
   Este o tehnică de regularizare, o extensie a regresiei liniare, populară în modelele statistice și învățarea automată, pentru a estima relațiile dintre variabile și realizarea de predicții.

   Principalul motiv pentru care se folosește acest model este că reușește să găsească o balanță între simplitate și precizie. Față de modelul de baza, regresia liniară, Lasso adaugă un termen de penalizare care este egal cu suma valorilor absolute ale coeficienților modelului [12].

   Avantaje:

   - Poate să selecteze caracteristicile mai importante în antrenare, iar pe celelalte să le ignore aproape complet.
   - Previne în mod eficient overfitting-ul pe datele de antrenare datorită sistemului de reglare a coeficienților regresiei.
   - Este mai performant în contextul în care numărul de variabile este mai mare decât numărul de caracteristici.

   Dezavantaje:

   - Tinde să selecteze doar o variabilă dintr-un grup de valori corelate, iar pe restul le ignoră.
   - Este limitat în modelele cu grad mare de multicolinearitate.
   - Performanța modelului este foarte legată de alegerea corectă a parametrului de penalizare.

3. **LSTM - Long Short-Term Memory**
   Acesta este unul dintre cele mai populare și eficiente modele utilizate în predicția seriilor de timp. O rețea LSTM poate fi antrenată pentru o varietate de aplicații precum recunoașterea vorbirii, prelucrarea limbajului natural, analiză video, predicția acțiunilor, etc [13] .

   LSTM este un tip de RNN (rețea neuronală recurentă) îmbunătățit, specializat pe învățare și reținere a relațiilor pe termen lung din date secvențiale. Arhitectura RNN este formată dintr-o rețea de celule care primesc un set de intrări și produc o ieșire ce poate folosită de celula următoare. În figura 2.1 observăm arhitectura unei rețele LSTM. Componentele acesteia sunt:

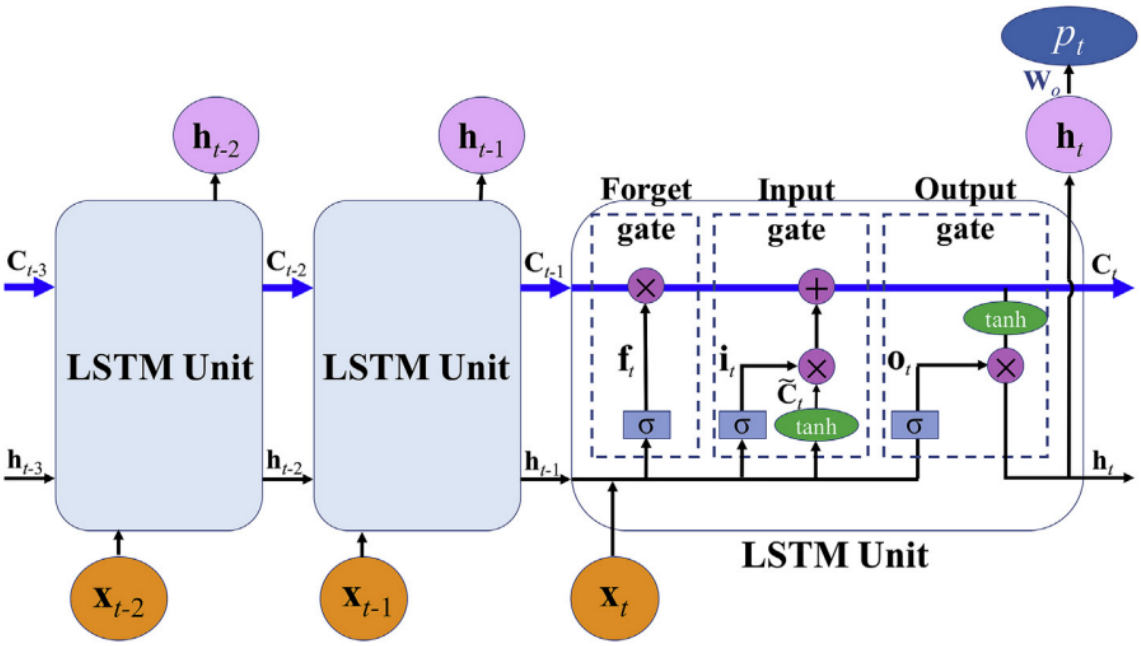


Figura 2.1: Arhitectura unei celule LSTM [14]

(a) Unitatea LSTM : Aici se păstrează și se actualizează informațiile pe termen scurt și lung prin intermediul unor structuri denumite porți (gates).

(b) Porțile LSTM :

- **Poarta de Uitare**: Decide ce informații să fie uitate din starea celulei anterioare C_{t-1}. Este reprezentată de f_t.
- **Poarta de Intrare**: Decide ce noi informații să fie stocate în starea celulei. Este reprezentată de i_t.
- **Poarta de Ieșire**: Decide ce parte din starea celulei să fie ieșirea curentă. Este reprezentată de o_t.

(c) Intrările și ieșirile celulei:

- **Intrări**:
  - X_t: Intrarea curentă.
  - h_{t-1}: Ieșirea anterioară.
  - C_{t-1}: Starea celulei anterioare.
- **Ieșiri**:
  - h_t: Ieșirea curentă.
  - C_t: Starea curentă a celulei.

(d) **Operațiuni interne**:

- σ : Funcția sigmoid, care comprimă valorile între 0 și 1. Este folosită în

porțile de uitare, intrare și ieșire pentru a decide cât de mult din fiecare componentă să fie trecut mai departe.

- tanh : Funcția tangentă hiperbolică, care comprimă valorile între -1 și 1. Este folosită pentru a crea vectorul de stare al celulei.
- C_t : Starea celulei este actualizată prin combinarea parțială a stării anterioare, înmulțită cu rezultatul de la poarta de uitare, și noua informație, înmulțită cu rezultatul de la poarta de intrare.

(e) **Rezultatul final**:

- h_t: Reprezintă ieșirea ascunsă curentă, determinată prin combinarea stării celulei curente cu rezultatul porții de ieșire.
- P_t: Reprezintă predicția finală sau ieșirea finală a rețelei LSTM, care poate fi folosită pentru sarcina specifică, cum ar fi predicția de serie temporală, recunoașterea vorbirii, etc [15].

Avantaje:

- Este capabil să analizeze și să exploateze interacțiunile dintre date printr-un proces de învățare automată.
- Reușește să facă predicții bune datorită analizei profunde ce dezvăluie tipare și trend-uri în date.
- Excelează în memoria pe termen lung.

Dezavantaje:

- Îi lipsește un mecanism de indexare a memoriei atunci când citește sau scrie date.

4. **ARIMA - Autoregressive integrated moving average**

Este unul dintre cele mai populare modele utilizate în predicția acțiunilor. Acesta a fost creat în 1970 de către George Box și Gwilym Jenkins în încercarea lor de a trata seriile de timp în mod matematic. Modelul este de obicei descris ca ARIMA(p, q, d), cei trei termeni reprezentând părțile algoritmului:

(a) p - Autoregresiva (AR)
Este numărul de valori întârziate Y care trebuie să fie adunate/scăzute la Y pentru a face predicții mai bune Dacă p este 1, înseamnă că datele cresc / descresc în mod liniar, iar pentru p = 2, exponențial. Formula pentru componenta AR este :

$$y_t = c + \phi_1 y_{t-1} + \phi_2 y_{t-2} + ... + \phi_p y_{t-p} + \epsilon_t$$

(b) D - Integrat (I)
Reprezintă de câte ori datele trebuie să fie diferențiate pentru a produce un efect staționar. Acest lucru ajută la reducerea de trend și sezonalitate.

(c) Q - Media mobilă (MA)
Reprezintă numărul de valori întârziate ale termenului de eroare ce trebuie adunate/scăzute la Y. Formulă pentru componentă MA este:

$$y_t = c + \epsilon_t + \theta_1\epsilon_{t-1} + \theta_2\epsilon_{t-2} + ... + \theta_q\epsilon_{t-q}$$

Astfel, formula generală pentru modelul ARIMA(p, d, q) este:

$$y_t = c + \phi_1 y_{t-1} + \phi_2 y_{t-2} + \cdots + \phi_p y_{t-p} + \epsilon_t + \theta_1\epsilon_{t-1} + \theta_2\epsilon_{t-2} + \cdots + \theta_q\epsilon_{t-q}$$

[16]

Avantaje:

- Lucrează foarte bine pe serii de timp liniare.
- Este cea mai eficientă tehnică de predicție în științele sociale.
- Pentru date pe perioade scurte, oferă eficiență și corectitudine mai bună față de alte modele mai complexe din punct de vedere structural.

Dezavantaje:

- Modelul realizat pe un set de date nu o să se potrivească pe altul, chiar dacă este asemănător.
- Are nevoie de multe date și implicit de un timp mare de antrenare.
- Se bazează pe un set de parametri și ipoteze care pot fi eronate, rezultând într-o predicție slabă [8].

Capitolul 3

Tehnologii folosite

3.1 Python

Python este unul dintre cele mai populare limbaje de programare datorită flexibilității sale și sintaxei ușor de învățat. Este interpretat, orientat pe obiecte și considerat de nivel înalt. În ultimii ani, programatorii de python au fost tot mai căutați pentru domenii precum Știința datelor, Învățare automata sau Automatizare [17].

Principalele avantaje pe care utilizatorii de python le consideră când aleg acest limbaj sunt :

1. Calitatea software : Python s-a concentrat pe ușurința de citire, mentenanță și coerență, ceea ce îl diferențiază de alte tehnologii din domeniu. Codul este conceput să fie lizibil, deci mai ușor de reutilizat și reținut decât alte limbaje tradiționale de scripting. De asemenea, oferă suport în mecanisme mai avansate precum programarea orientată pe obiecte;

2. Productivitatea developerului : Acest limbaj crește productivitatea programatorilor mult față de limbajele compilate precum C, C++ sau Java datorită scăderii semnificative de linii de cod la aproximativ un sfert. Prin urmare, timpul de scriere, debugging sau mentenanță este mai mic, dar și cel de rulare prin faptul că este interpretat;

3. Utilitatea librăriilor : Python oferă o colecție largă de funcționalități de bază precum vectori, procesarea textelor și a fișierelor sau networking, însă majoritatea îl preferă datorită librăriilor importate precum Numpy, Pandas, Matplotlib, Scikit-learn, etc. Utilizând acestea, programatorii au și mai multă libertate în dezvoltarea de aplicații : site-uri web, jocuri video, analiză de date, învățare automată și altele.

Aceste beneficii, dar și altele, creează programatorilor un mediu plăcut și productiv care îi ajută să creeze orice își propun, singură limită fiind imaginația și perseverența lor [18].

3.2 NumPy, Pandas

NumPy este o librărie python ce oferă o structura de date simplă, dar foarte utilă, mai exact matricele cu N dimensiuni. Acest concept este considerat fundația pe care se bazează multe alte instrumente folosite în știința datelor, dar și librării populare precum SciPy, Matplotlib, Pandas sunt bazate pe această bibliotecă.

Principalele beneficii pe care le oferă NumPy:

- Viteza de rulare : Algoritmii din această librărie sunt implementați în limbajul C, ceea ce face programele să ruleze de până la 100 de ori mai repede;
- Diminuarea codului : Funcțiile din NumPy ajută la reducerea numărului de structuri repetitive și indici, ceea ce face codul mai curat și mai scurt;
- Calitatea sporită : Pentru a întreține această librărie la fel de rapidă, ușor de folosit și fără erori, mii de contribuitori lucrează la ea constant [19].

O altă librărie foarte importantă în știință datelor este Pandas. Aceasta utilizează seturi de date de tip CSV (valori separate de virgulă), Excel, XML, JSON, dar și tabele din baze de date. Pandas transformă aceste date într-un DataFrame care poate fi manipulat foarte ușor, utilizând funcții specializate din librărie.

Pandas este folosit pentru analiza datelor, exemple de aplicații fiind : studii științifice, analize de piață, aplicații financiare, sportive, de sănătate, etc.

Avantajele utilizării Pandas sunt următoarele:

- Simplitatea : Structurile de date precum Series și DataFrame sunt ușor de folosit și înțeles;
- Capacitatea de prelucrare a datelor : Funcțiile integrate ajută la tratarea datelor lipsă, a erorilor, inconsistenței, dar și la calcularea diferitelor valori importante în analiză precum procentul de corelație, deviația standard sau medii mobile;
- Selecția caracteristicilor : Această librărie oferă sprijin în manipularea subseturilor de date, ceea ce poate folosi în selecția și ignorarea de trăsături [20].

3.3 Matplotlib, Seaborn

Matplotlib este o librărie Python foarte utilă în crearea de reprezentări vizuale a datelor, făcând analiza de date mai ușoară. Folosind funcțiile implementate, programatorul poate crea histograme, grafice, statistici și altele prin doar câteva linii de cod.

Funcționalitățile importante ale librăriei sunt :

- Versatilitatea : Folosind Matplotlib se pot genera o varietate de figuri și grafice, iar acestea pot fi stilizate prin modificarea culorii, tipului de linie, notațiilor, legendei, etc;
- Extensibilitatea : Funcțiile acestei biblioteci sunt compatibile cu altele precum NumPy, Pandas, Seaborn, ușurând astfel vizualizarea datelor;
- Calitatea rezultatelor : Vizualizarea datelor se poate salva în diferite formate precum PNG, PDF, SVG, asigurând compatibilitatea cu diverse tipuri de documente științifice sau prezentări la o calitate digitală mare [21].

Seaborn este o librărie ce se bazează pe Matplotlib și integrează structuri din Pandas. Aceasta ajută la explorarea și înțelegerea diverselor informații din date prin crearea de grafice statistice.

Reprezentările vizuale din Seaborn ajută la vizualizarea relațiilor dintre variabile prin funcții precum PairPlot sau PairGrid, utile în analiza exploratorie a datelor. Câteva exemple de grafice sunt :

- Relaționale : Folosite în înțelegerea asemănărilor dintre variabile;
- Categorice : Utilizate pentru vizualizarea datelor pe categorii;
- Regresive : Acestea adaugă un suport vizual în identificarea structurilor din seturile de date analizate în EDA;
- Multiple : Se pot genera mai multe grafice ale aceluiași set de date, dar pe alte subseturi [22].

3.4 Scikit-learn, Keras

Scikit-learn este o librărie Python de învățare automată ce integrează o multitudine de algoritmi și modele ce pot fi utilizate de programatori. Are la bază librării precum NumPy, SciPy și Matplotlib prin care se implementează algoritmi precum regresia liniară, mașini vectoriale de suport (SVM), păduri aleatoare sau KNN.

Cele mai importante funcționalități ale bibliotecii sunt :

- Preprocesare : funcții ce ajută în extragerea de trăsături, normalizarea și clasificarea informațiilor în diferite seturi sunt folosite pentru analiză de date;
- Regresie : Modele ce încearcă să înțeleagă relația dintre datele de intrare și cele de ieșire precum prețul acțiunilor;

- Selectare de modele : Algoritmi ce pot automatiza selecția setului optim de parametri pentru modelele folosite în proiecte de învățare automată [23].

O ultimă librărie importantă în acest proiect este Keras, ce oferă o interfață Python pentru rețele neuronale. Aceasta este cunoscută pentru ușurința de utilizare și dezvoltare, dar și flexibilitatea mare.

Componentele importante din această biblioteca sunt :

- Modele : Structurile de bază în Keras sunt modelele implementate, cel mai popular fiind cel secvențial, iar unul mai complex și mai flexibil este API Funcțional;
- Straturi : Acestea sunt cele mai importante structuri în construirea de rețele neuronale și pot fi convoluționale, pooling, recurente, dense, și altele;
- Optimizatori : Sunt metode precum Adam, SGD prin care se pot modifica atributele rețelei (greutățile, rata de învățare);
- Funcții de pierdere : Acestea sunt folosite pentru a calcula diferența dintre valorile reale și cele prezise de model. Cele mai utilizate sunt mean_squared_error sau categorical_crossentropy;
- Metrici : Sunt folosite în evaluarea performanței modelului, precum acuratețe, precizie sau recall [24].

Capitolul 4

Predicția acțiunilor

4.1 Etapele analizei

Pentru realizarea EDA, am propus următoarele întrebări:

- Cum arată trecutul acțiunilor Netflix în comparație cu ale altor companii tech (FAANG)?
- Care a fost daily return-ul stocului în medie?
- Care a fost moving average-ul (MA) stocurilor în trecut și cum influențează predicția?
- Care este procentul de corelație dintre diferite stocuri?
- Care sunt cele mai sigure/profitabile stocuri pentru investit?
- Cum putem să prezicem viitorul acțiunilor?

4.2 Preluarea datelor

În primă fază, vom avea nevoie de datele actuale despre stocurile Netflix. Am ales dintre companiile FAANG pe Apple, Google, Microsoft, Meta, dar și o companie ce produce conținut similar cu Netflix, The Walt Disney.

Inițial, am folosit setul de date "Netflix Stock Price (All Time)" creat de Abhi, un fișier CSV de aproape 5000 de linii cu date între anii 2002-2021. Pentru a avea o acuratețe mai mare și o actualitate sporită, am folosit librăria yfinance în python, adică date de pe site-ul Yahoo Finance.

Dataset-ul obținut conține informații de bază despre stocurile companiilor vizate, împărțite în următoarele coloane:

1. Date: Conține data la care s-a măsurat acțiunea;

2. Open: Este primul preț al acțiunii din ziua în care s-a măsurat;

3. High: Cel mai mare preț măsurat în acea zi (utilizat în general pentru a calcula volatilitatea acțiunilor);

4. Low: Cel mai mic preț măsurat în acea zi;

5. Close: Conține prețul acțiunii la finalul zilei de tranzacționare;

6. Adj Close: Este considerat prețul real al acțiunii, reprezentat de valoarea acțiunii după ce s-au distribuit dividendele;

7. Volume: Este numărul de acțiuni comercializate în ziua respectivă ;

8. Company name: Conține numele prescurtat al companiei al cărei acțiuni a fost înregistrată. De exemplu, Netflix Inc. este scris ca NFLX în afacerile online.

Out[38]:

Date	Open	High	Low	Close	Adj Close	Volume	company_name
2024-05-17	470.829987	472.799988	468.420013	471.910004	471.910004	10807300	META
2024-05-20	469.950012	473.200012	467.040009	468.839996	468.839996	11745100	META
2024-05-21	467.119995	470.700012	462.269989	464.630005	464.630005	11742200	META
2024-05-22	467.869995	473.720001	465.649994	467.779999	467.779999	10078600	META
2024-05-23	472.880005	474.359985	461.540009	465.779999	465.779999	11747900	META
2024-05-24	467.619995	479.850006	466.299988	478.220001	478.220001	12012300	META
2024-05-28	476.579987	480.859985	474.839996	479.920013	479.920013	10175800	META
2024-05-29	474.660004	479.850006	473.700012	474.359985	474.359985	9226200	META
2024-05-30	471.670013	471.730011	464.709991	467.049988	467.049988	10719600	META
2024-05-31	465.799988	469.119995	454.460114	455.904999	455.904999	4210804	META

Figura 4.1: Setul de date

În figura 3.1, observăm că nu toate datele sunt consecutive, deoarece weekend-urile și sărbătorile legale nu se consideră zile de tranzacționare. Pentru mai multe informații despre datele procurate, am folosit funcțiile .describe() și .info() din librăria **pandas**.

În figura 3.2, rândul cu numele ”count” ne arată că fiecare coloana de care avem nevoie conține 3020 de date, iar în figura 3.3, că toate acestea sunt nenule, iar tipurile de date sunt doar 3, împărțite în mod corespunzător.

Pentru a vizualiza datele descărcate prin grafice și histograme, vom folosi librăria **mathplotlib**.

În figura 3.4, putem observa prețurile acțiunilor în trecut și ne putem face o idee despre evoluția companiilor în timp. Am utilizat prețul ”Adj Close” pentru axa Y deoarece acesta este cel considerat standard în trading.

In [39]:
```
# Summary Stats
NFLX.describe()
```

Out[39]:

	Open	High	Low	Close	Adj Close	Volume
count	3020.000000	3020.000000	3020.000000	3020.000000	3020.000000	3.020000e+03
mean	252.315360	256.116366	248.410398	252.361442	252.361442	1.282906e+07
std	184.471992	186.911862	181.895065	184.423445	184.423445	1.354660e+07
min	7.712857	7.925714	7.544286	7.685714	7.685714	7.637980e+05
25%	87.540716	88.807497	85.751785	87.844997	87.844997	5.157925e+06
50%	227.584999	233.855003	223.684998	227.510002	227.510002	8.658150e+06
75%	379.794991	385.204994	374.494995	380.015007	380.015007	1.610765e+07
max	692.349976	700.989990	686.090027	691.690002	691.690002	1.914458e+08

Figura 4.2: Rezultatul funcției describe

In [40]:
```
# General info
NFLX.info()
```

```
<class 'pandas.core.frame.DataFrame'>
DatetimeIndex: 3020 entries, 2012-05-31 to 2024-05-31
Data columns (total 7 columns):
 #   Column        Non-Null Count  Dtype
---  ------        --------------  -----
 0   Open          3020 non-null   float64
 1   High          3020 non-null   float64
 2   Low           3020 non-null   float64
 3   Close         3020 non-null   float64
 4   Adj Close     3020 non-null   float64
 5   Volume        3020 non-null   int64
 6   company_name  3020 non-null   object
dtypes: float64(5), int64(1), object(1)
memory usage: 188.8+ KB
```

Figura 4.3: Rezultatul funcției info

Deși unele companii datează din anii 1970, am preluat doar date din 2012, pentru ca acestea să fie actuale și mai ușor de vizualizat. Cele 6 grafice ne arată că există mai multe corelații comune:

- Prețul foarte mic în 2012, înainte că acțiunile FAANG să fie populare;
- Trend-ul de creștere continuă în anii următori, până în 2020;
- Scăderea bruscă a unora din cauza pandemiei.

Totuși, fiecare companie se confruntă constant cu diverse probleme. De exemplu, în 2022, Netflix a avut cea mai agresivă scădere din istoria sa, deoarece a publicat o statistică care arată pierderea a aproape 1 milion de subscripții în ultimele 3 luni. Pentru prima oară în 20 ani, competiția cu alte platforme precum Disney+, Apple TV, HBO Max, dar și creșterea inflației au produs un eveniment major în istoria prețului Netflix. Un alt factor ar putea fi reprezentat de încercările repetate ale corporației de a interzice folosirea unui singur cont de către mai mulți utilizatori ce nu locuiesc în aceeași casă [25] .

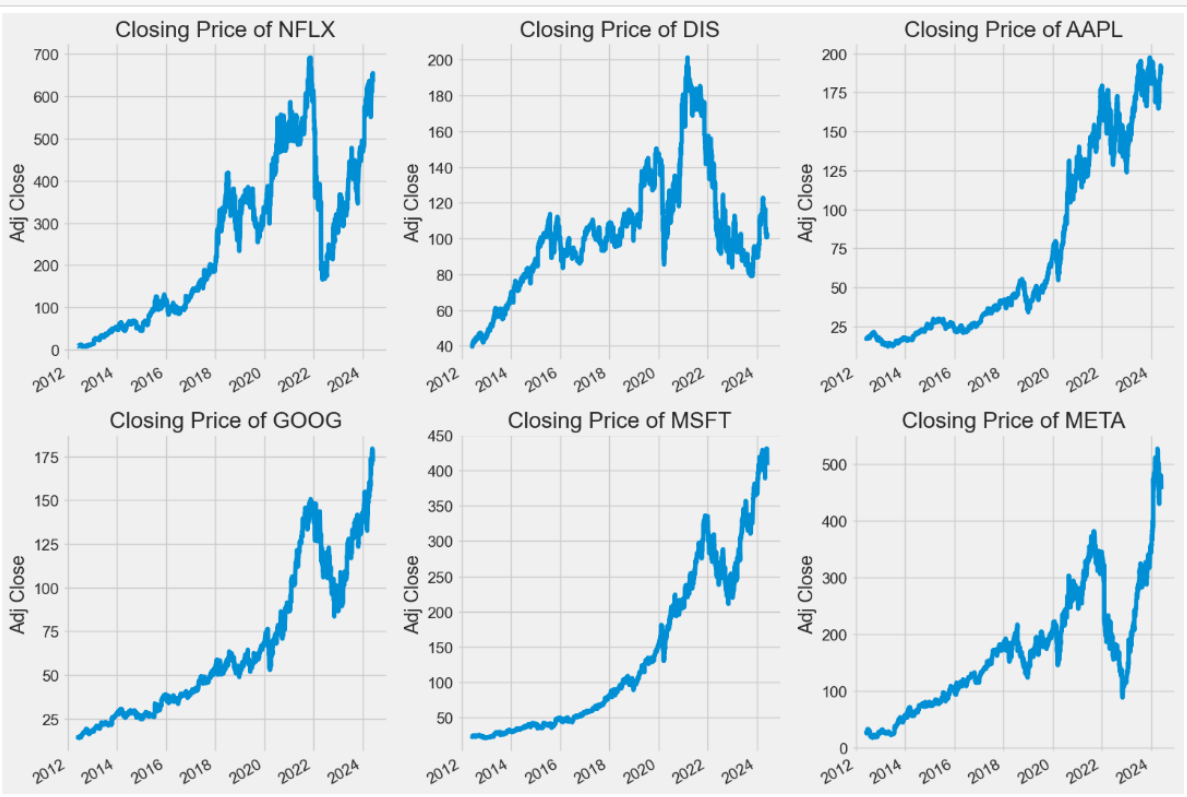


Figura 4.4: Prețurile de închidere ale companiilor

Pentru a avea o viziune mai clară asupra prețurilor, am calculat moving average-ul (MA). Acesta este un indicator crucial în analiza noastră deoarece ne ajută să identificăm trend-uri pe termen lung sau cicluri. Am analizat MA pe mai multe perioade de timp, iar rezultatul a fost că pentru 10 și 20 zile, graficele sunt cele mai utile, pentru că pot să capteze trend-uri și schimbări bruște mai precis.

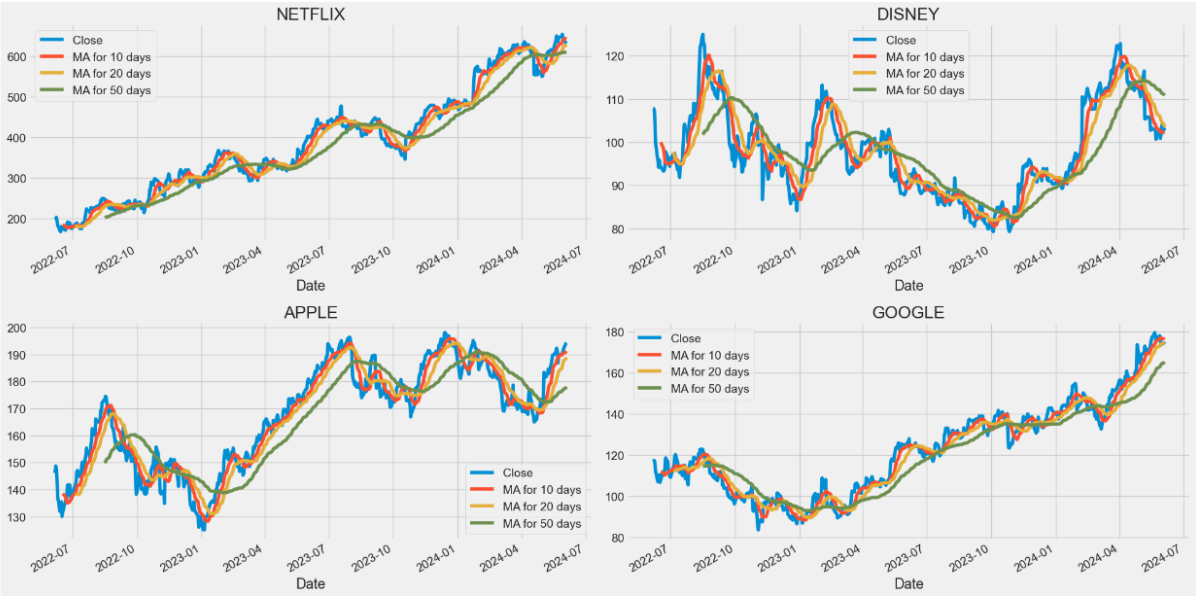


Figura 4.5: Mediile mobile

În continuare, am analizat volumul vânzărilor, un indicator cantitativ important pentru comercianții tehnici. În figura 3.5, observăm că volumul de acțiuni tranzacționate variază semnificativ de la o companie la altă, deci nu ne putem folosi de legătura dintre ele. În schimb, putem analiza fluctuațiile volumului în contextul restrâns al unei singure companii.

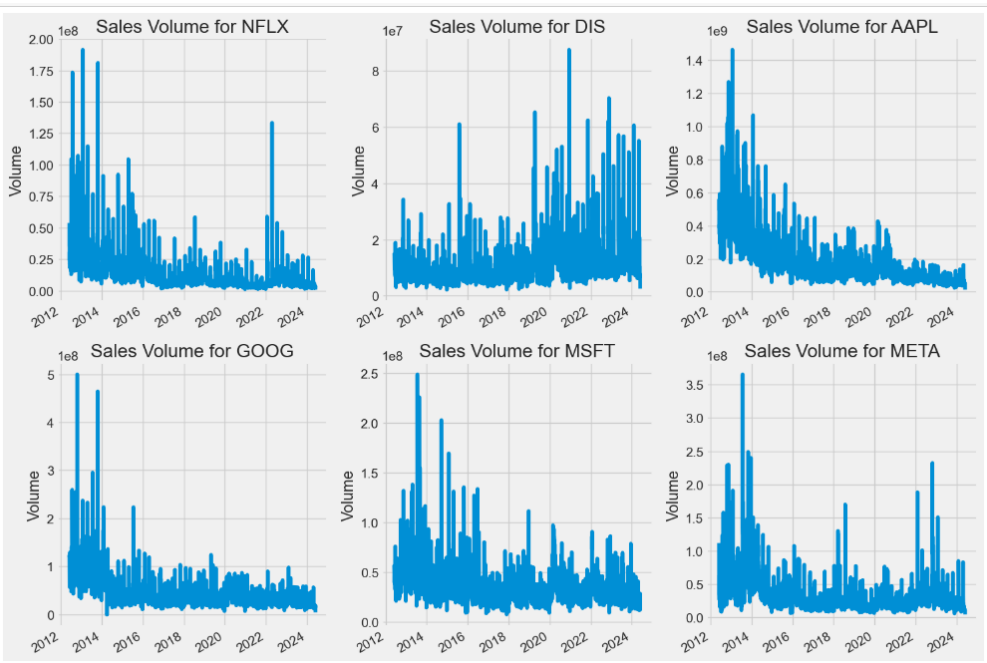


Figura 4.6: Volumul vânzărilor

4.3 Randamentul zilnic. Corelații

Acum că avem câteva informații de bază despre datele noastre, putem să analizăm mai în detaliu și să obținem informații prețioase. Daily return-ul acțiunilor este o valoare ce măsoară în procente cât s-a schimbat valoarea de închidere a acțiunii de la o zi la altă. Formulă utilizată în practică este :

$$\text{Daily Return} = \frac{P_{\text{today}} - P_{\text{yesterday}}}{P_{\text{yesterday}}}$$

Totuși, librăria `pandas` ne oferă metoda `pct_change()`, calculând eficient procentele de care avem nevoie :

```
for company în company_list:
company['Daily Return'] = company['Close'].pct_change()
```

Pentru a vizualiza daily return-ul fiecărei companii, am creat histograme folosind librăria `seaborn`.

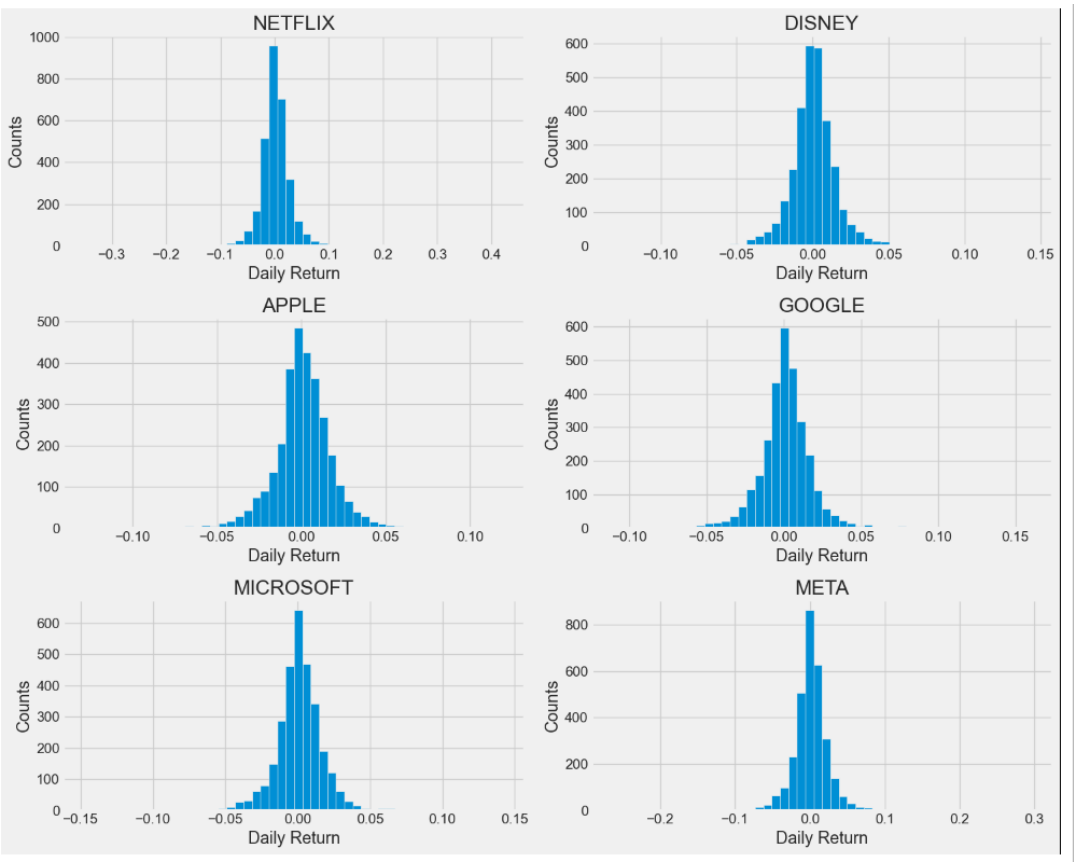


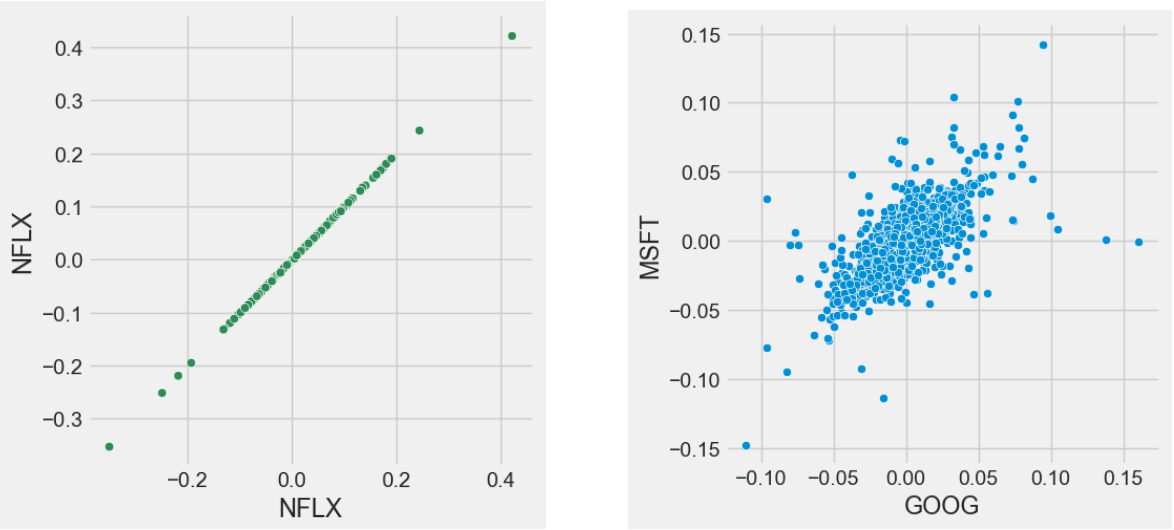
Figura 4.7: Randamentele zilnice

Următorul pas în analiza EDA este calcularea corelațiilor dintre mai multe companii. Aceasta este o statistică care măsoară un coeficient între -1.0 (creșterea x determina scăderea y), și +1.0 (creșterea x determina creșterea y). Un coeficient aproape de 0 înseamnă un grad foarte mic de corelație.

Acești coeficienți ne arată asocierea dintre două companii, însă nu știm care este cea influențată de cealaltă, și nici dacă aceasta se produce de un factor extern.

Pentru următoarele grafice, am folosit librăria `seaborn`. Pe axele x și y avem daily return-ul aferent companiilor comparate. Prima figura (a) reprezintă o corelație completă dintre datele aceleiași companii, Netflix. Acest grafic servește drept model în căutările noastre ulterioare.

Punctele din figura (b) tind să formeze o diagonală asemănătoare cu cea din figura (a), deci daily return-urile companiilor se modifică în corelație unul cu celălalt. Prin urmare, există o corelație pozitivă între Microsoft și Google.



(a) Corelația dintre Netflix-Netflix

(b) Corelația dintre Microsoft-Google

Figura 4.8: Compararea corelațiilor dintre diferite seturi de date

Mai departe, am utilizat funcția `sns.PairGrid()` pentru a crea automat toate comparațiile posibile. Pe axele X și Y avem, de data aceasta, prețul de închidere al acțiunilor.

Folosind regula de la daily return, din aceste grafice putem să identificăm câteva corelații importante precum Apple-Google, Apple-Microsoft, Microsoft-Google, dar și Netflix-Meta.

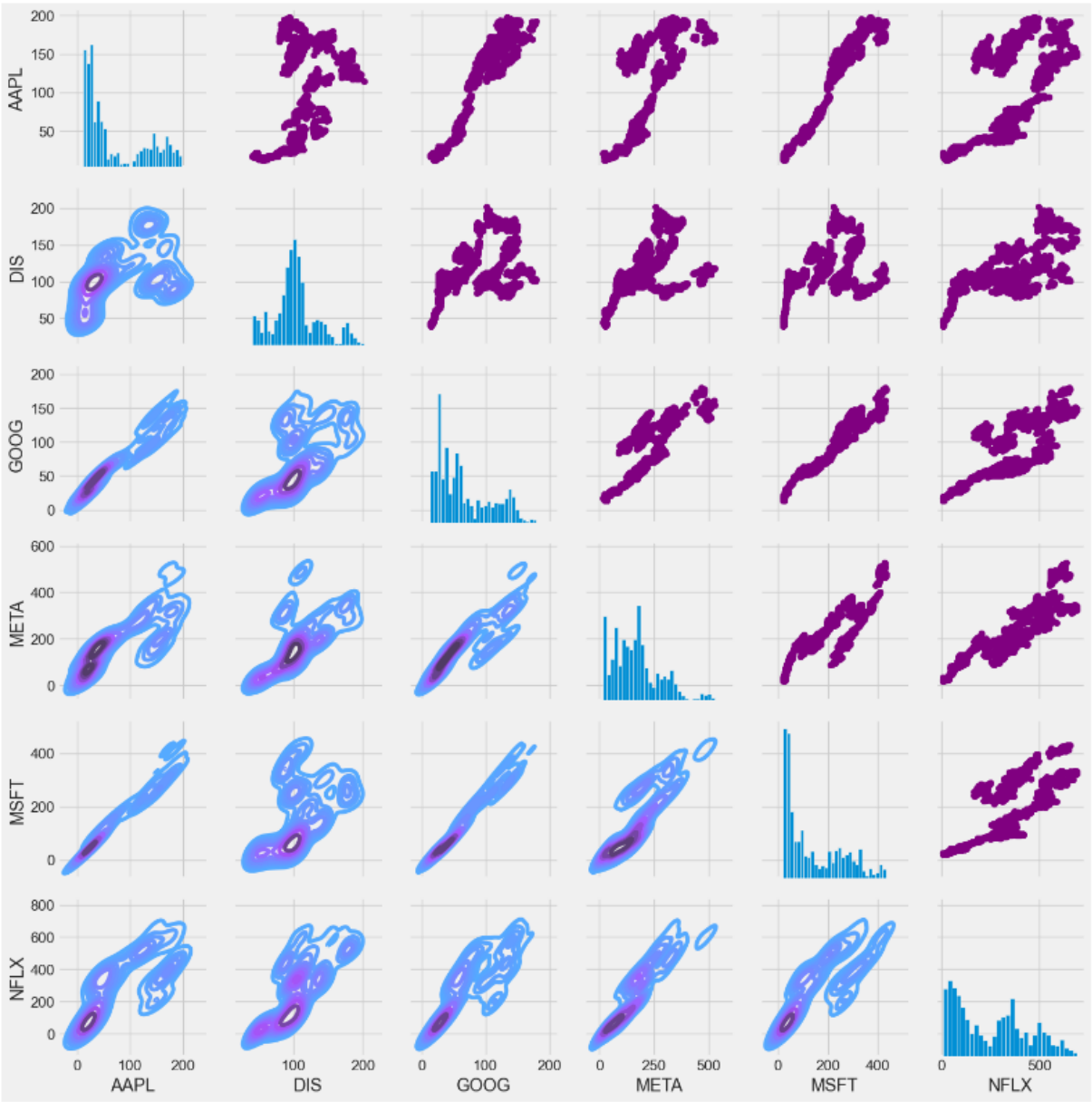


Figura 4.9: Rezultatul funcției PairGrid

Pentru o vizualizare numerică, am creat un heatmap (figura 3.10) ce arată gradul de corelație dintre diferite variabile (în cazul nostru, prețurile de închidere ale acțiunilor).

Cele mai mari valori (peste 0.9) sunt chiar cele menționate mai sus, deci am reușit să identificăm cu succes conexiunile dintre companii.

De asemenea, este de notat faptul că membrii FAANG au un procent foarte mare de corelație între ei (cel mai procent mic fiind 0.78), iar Disney, deși este o companie asemănătoare Netflix, are legături foarte slabe cu toate celelalte.

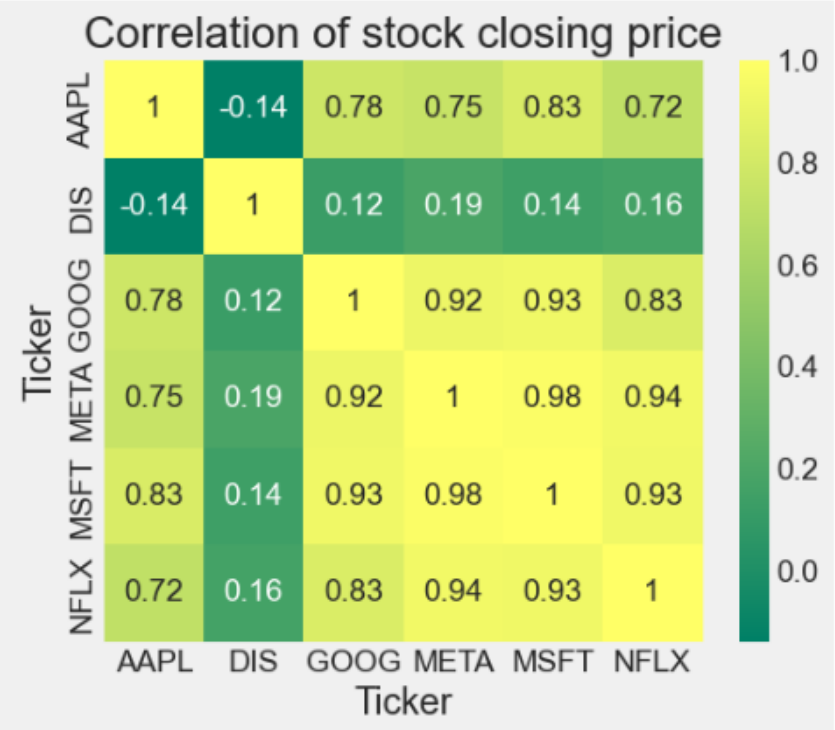


Figura 4.10: Heat map

O ultima analiză interesantă și folositoare este calcularea riscului de investiție în acțiunile prezentate. Există mai multe metode prin care putem să analizăm performanța investițiilor, iar una dintre metodele clasice este compararea dintre randamentul așteptat (expected return) și randamentul zilnic (daily return). Această comparație ne ajută să vizualizăm potențialul de câștig al unei investiții, dar și riscul asociat al acesteia.

În figura 3.11 observăm care sunt cele mai sigure companii (Apple, Google, Microsoft) și cele mai riscante (Netflix, Meta). Am comparat rezultatele cu cele dintr-un articol de la The Motley Fool [26], o agenție de consiliere financiară, iar acestea sunt foarte asemănătoare.

Explicația este că primele 3 sunt întreprinderi care se bazează pe rezultate sigure, consecvente, în timp ce Meta investește foarte mult într-un concept futurist, metavers. De asemenea, Netflix folosește un sistem de abonamente ce poate fi vulnerabil la fluctuațiile economice sau schimbări în preferințele consumatorilor. Așadar, acțiunilor lor sunt mai volatile, deci mai riscante.

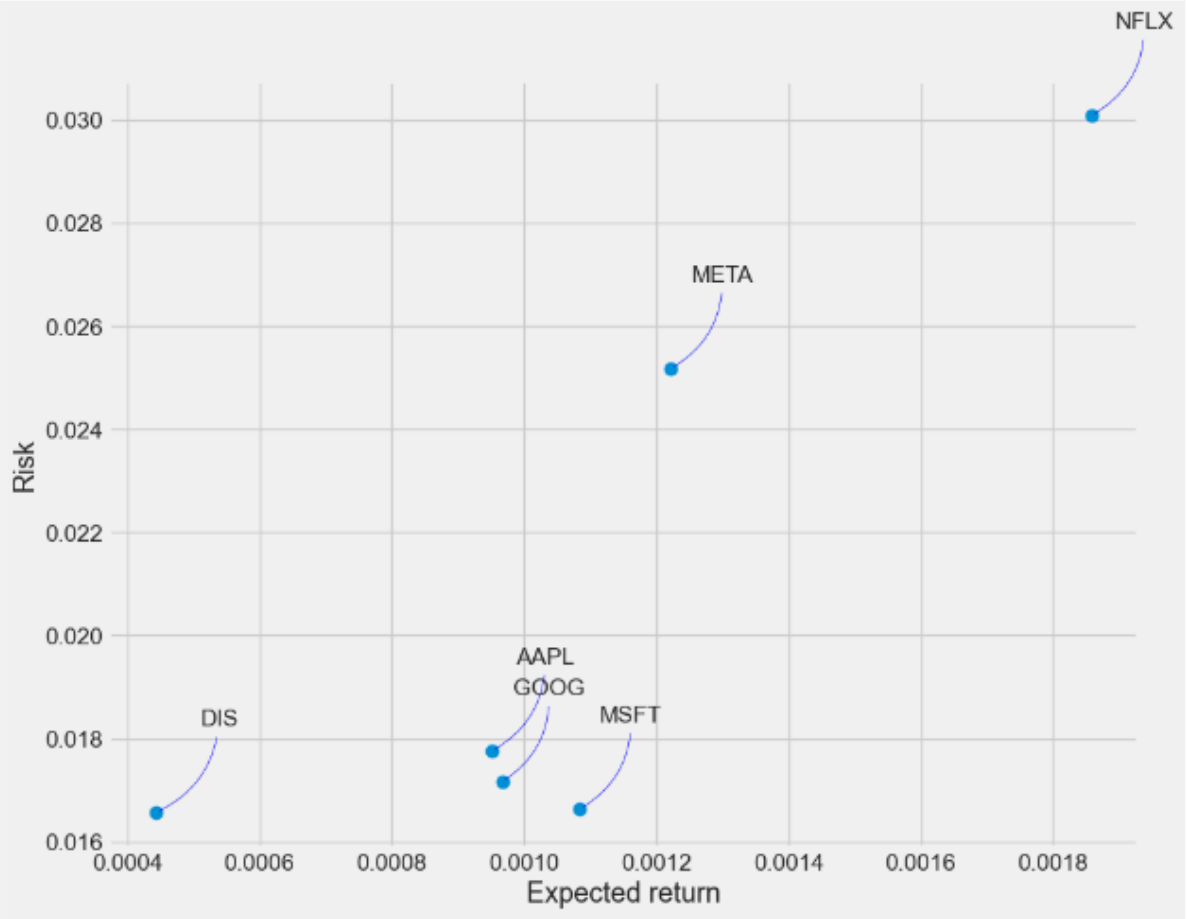


Figura 4.11: Riscurile de investiție în companii

4.4 Prezicerea viitorului acțiunilor Netflix

În această secțiune, voi prezenta evoluția modelelor folosite în prezicerea de acțiuni și rezultatele pe care le-am obținut cu fiecare dintre acestea.

4.4.1 Linear Regression

Am început cu acest model pentru a evalua performanța unei abordări simple. Pentru antrenarea acestui model am folosit datele NFLX din 2020 până în prezent, pe care le-am salvat într-un DataFrame. Datele sunt împărțite în două seturi: primul constă în datele de antrenare (80%), iar al doilea în cele de testare (20%).

Știm de la preprocesarea datelor că nu avem valori nule, deci nu mai este nevoie de alte pregătiri. Cel mai important pas în antrenarea modelului este alegerea de caracteristici. O practică generală este folosirea de trăsături întârziate, adică valori precum ”close” din trecut. După mai multe teste, am observat că valorile întârziate cu 1, 3 și 7 zile sunt cele mai utile, așa că le-am adăugat pe acestea.

Pentru evaluarea modelului am utilizat eroarea medie pătrățica (MSE). Această măsoară media diferențelor dintre valorile reale și cele prezise. O valoare mică MSE înseamnă precizie mai mare, deci un model mai performant în predicția acțiunilor. Pentru vizualizarea rezultatelor am folosit biblioteca **Mathplotlib**, comparând prețurile reale cu prețurile prezise de model.

În urma primelor simulări, s-a obținut o valoare a erorii pătratice medii de **308**, rezultatele fiind reprezentate de graficul din figura 3.12.

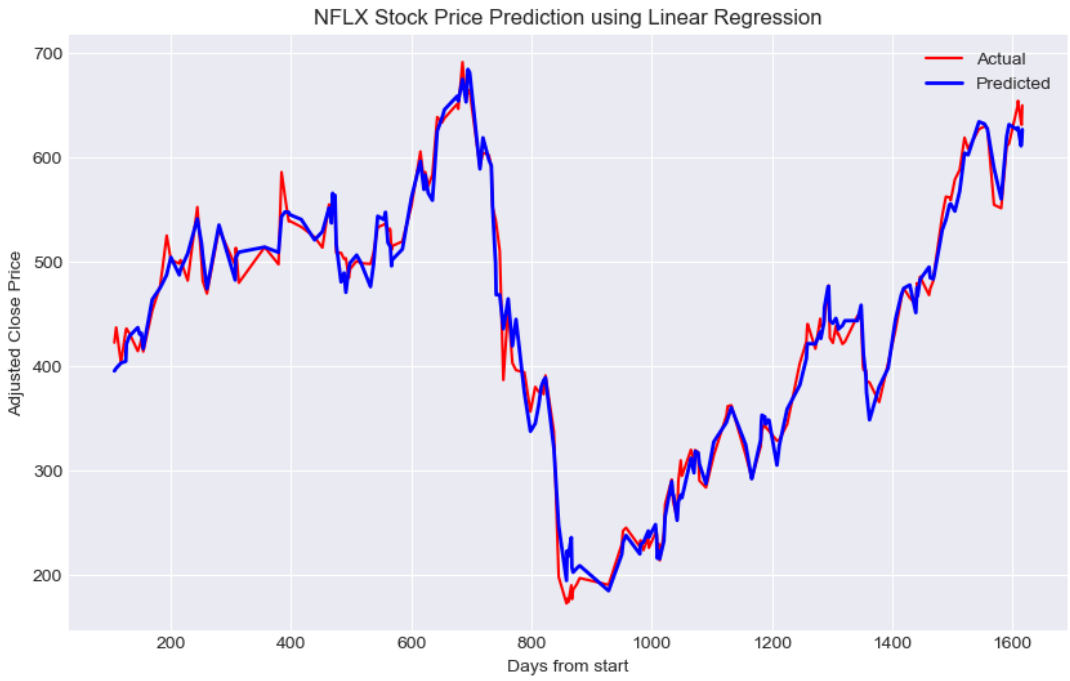


Figura 4.12: Modelul Regresie Liniară

Pentru a îmbunătăți performanța modelului, am integrat mediile mobile ca trăsături. Am calculat MA pentru 10, 20, 30 și 60 zile folosind codul de mai jos :

```
df['SMA_10'] = df['Adj Close'].rolling(window=10).mean()
```

Așa cum am prezis în analiza EDA, mediile mobile pe 10 și 20 zile au fost cele mai utile în creșterea eficienței modelului. Cu această îmbunătățire, eroarea MSE a scăzut la aproape jumătate, mai exact **167**.

4.4.2 Lasso regression

Al doilea model pe care l-am utilizat este regresia LASSO. Am parcurs aceiași pași ca la algoritmul precedent, însă rezultatele au fost total diferite.

În primul rând, simulările inițiale au produs rezultate mai bune decât tot ce am încercat la regresia liniară, având o eroare între **120-130**. Acest lucru arată deja un progres considerabil, așa că am încercat să micșorez această valoare în continuare.

După integrarea caracteristicilor întârziate și a mediilor mobile, precizia modelului a rămas relativ constantă, cu mici fluctuații. Ca urmare, am încercat o altă abordare: implementarea de trăsături pe baza corelațiilor analizate anterior.

Am descărcat datele tuturor companiilor analizate și le-am sortat în funcție de procentul de similaritate cu Netflix. Apoi, am calculat prețurile întârziate ale acțiunilor Netflix și Meta (cea mai corelată) și le-am inclus în model. Datorită capacității sale de selecție a trăsăturilor, LASSO a ajustat coeficienții în favoarea unui rezultat mai bun, adică o valoare de **110**. Graficul din figura 3.13 reprezintă cel mai performant model de până acum.

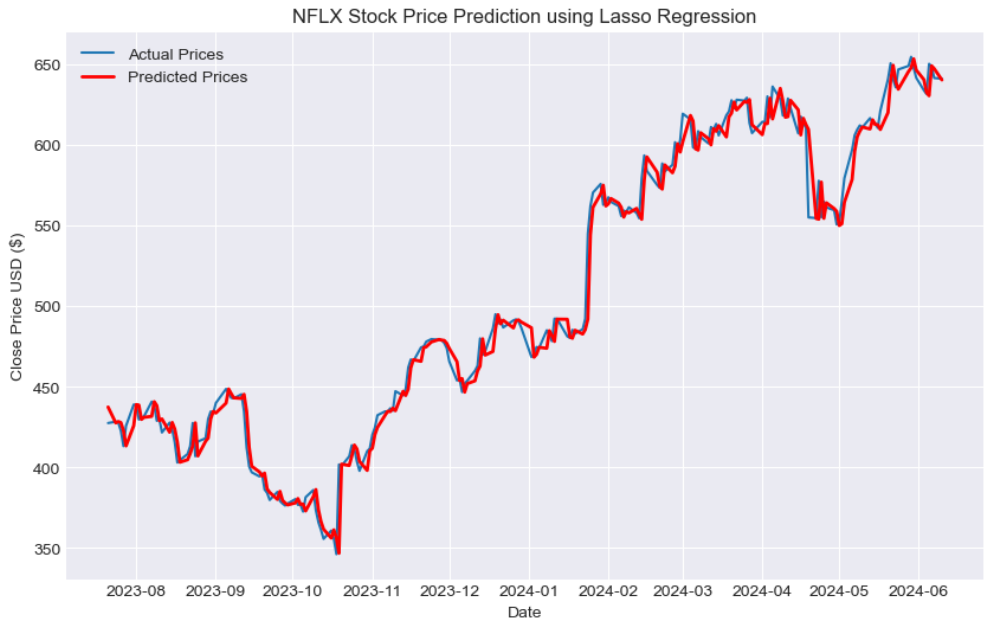


Figura 4.13: Modelul Regresie Lasso

4.4.3 Long Short-Term Memory

Pentru a experimenta cu un model performant, considerat chiar state-of-the-art în predicția prețurilor, am implementat o rețea LSTM.

Deoarece rețelele neuronale sunt proiectate să proceseze secvențe de date, am împărțit datele de intrare în serii de câte 60 zile. Modelul se va antrena pe acest set și va prezice prețul acțiunii pentru ziua 61.

Următorul pas este construirea propriu-zisă a modelului :

- Am utilizat 2 straturi LSTM pentru învățarea pe baza secvențelor create anterior. Primul conține 128 unități și returnează alte secvențe, pe care cel de-al doilea strat le va prelua. Această abordare este de folos în captarea dependențelor temporale complexe din datele noastre.

- Am adăugat 2 straturi de tip Dense: primul strat cu 25 unități folosește funcția de activare Relu, introducând non-linearitatea în model și creșterea performanței. Al doilea strat este cel de ieșire, cu o singură unitate ce produce predicția finală a prețului de închidere.

- Între structurile LSTM și Dense am inclus câte un strat Dropout(0.2) ce ajută la reducerea overfitting prin ignorarea a 20% din neuroni pe timpul procesului de antrenare. Această tehnică este utilă în generalizarea mai bună a modelului pe date noi, de testare.

La compilarea modelului, am ales optimizatorul standard Adam, aceeași funcție de pierdere MSE, iar antrenarea s-a realizat pe un lot de dimensiunea 4 și 10 epoci.

Primele simulări arată că modelul LSTM este mult mai greu de antrenat, deoarece rezultatul inițial este mult mai slab decât cele de până acum. Valoarea erorii medii pătratice este **540**, iar din grafic (figura 3.14), rețeaua pare că nu captează în mod corect schimbările de preț.

Deși acest lucru m-a descurajat în a mai continuă cu dezvoltarea rețelei, am aplicat mai multe tehnici învățate până acum, dar și altele mai avansate :

1. **Caracteristici întârziate** : Am inclus valorile prețului de închidere din urmă cu 1, 3 și 7 zile pentru o mai bună înțelegere a volatilității.

2. **Corelații cu alte companii** : Dintre companiile analizate, Meta și Microsoft au arătat cea mai mare corelație (0.94, respectiv 0.93), așa că am adăugat prețurile de închidere întârziate ale acestora în antrenarea modelului.

3. **Medii mobile** : Am calculat aceste valori pentru intervalul de 10 și 20 zile ale celor 3 corporații și le-am adăugat la trăsături.

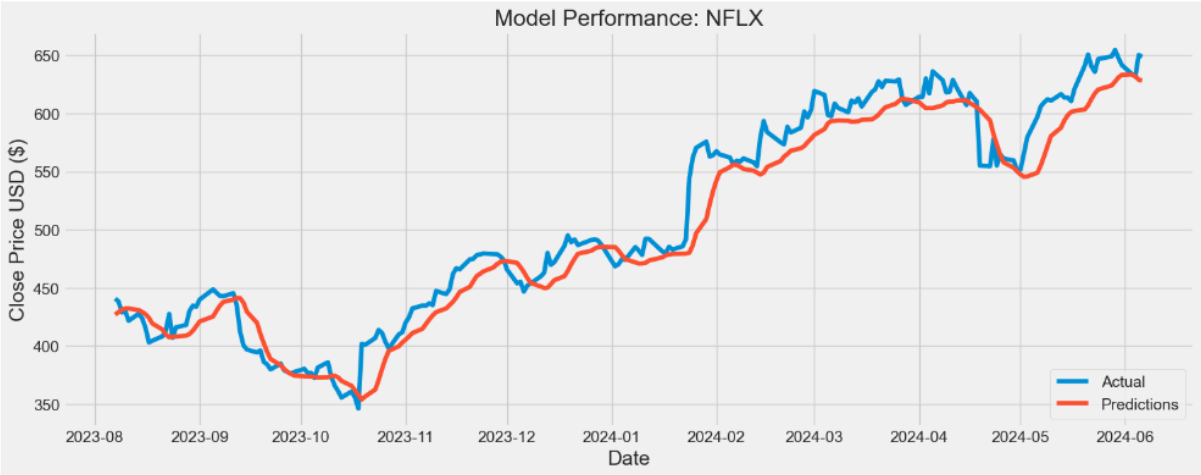


Figura 4.14: Modelul LSTM inițial

4. **Indicele de putere relativă (RSI)** : Acest indice este un indicator util pentru detectarea condițiilor anormale (supra-cumpărare/vânzare), ajutând la anticiparea unor posibile schimbări bruște în prețul acțiunilor.

5. **Mecanism de atenție** : Am implementat acestă tehnică ce a sporit semnificativ performanța predicțiilor, prin faptul că datele de intrare au fost mai atent analizate. Algoritmul a reușit să extragă caracteristicile importante mai bine și astfel rezultatele au fost mai apropiate de valorile reale.

6. **Tuning hiperparametric** : Am efectuat o căutare exhaustivă a parametrilor, modificând rata de învățare, dimensiunile loturilor, numărul de epoci, dar și caracteristicile incluse mai sus în antrenare.

După toate aceste îmbunătățiri, am reușit să obțin un rezultat bun, având o valoare de **120** pentru eroarea medie pătratica și graficul din figura 3.15. Totuși, modelul Lasso a avut o performanță puțin mai bună, deci am continuat să antrenez rețeaua LSTM pe diferite configurații. Am descărcat mai multe date, din 2015 până în prezent, iar acest lucru m-a ajutat să ajung în final la un MSE puțin sub pragul de **100** pe care mi l-am propus (figura 3.16).

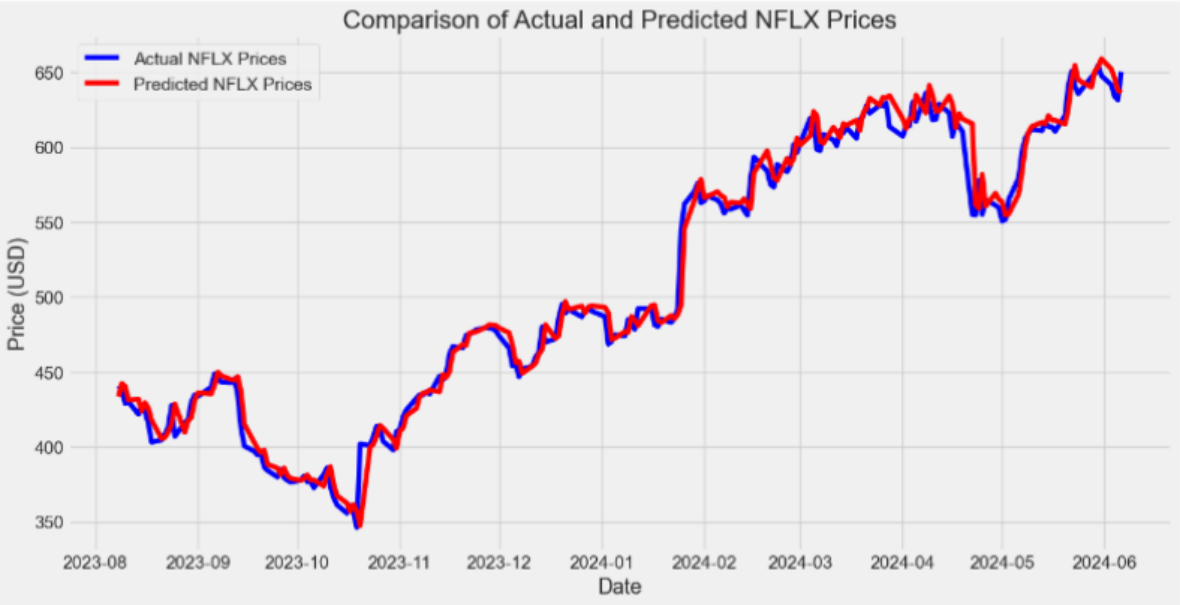


Figura 4.15: Modelul LSTM îmbunătățit

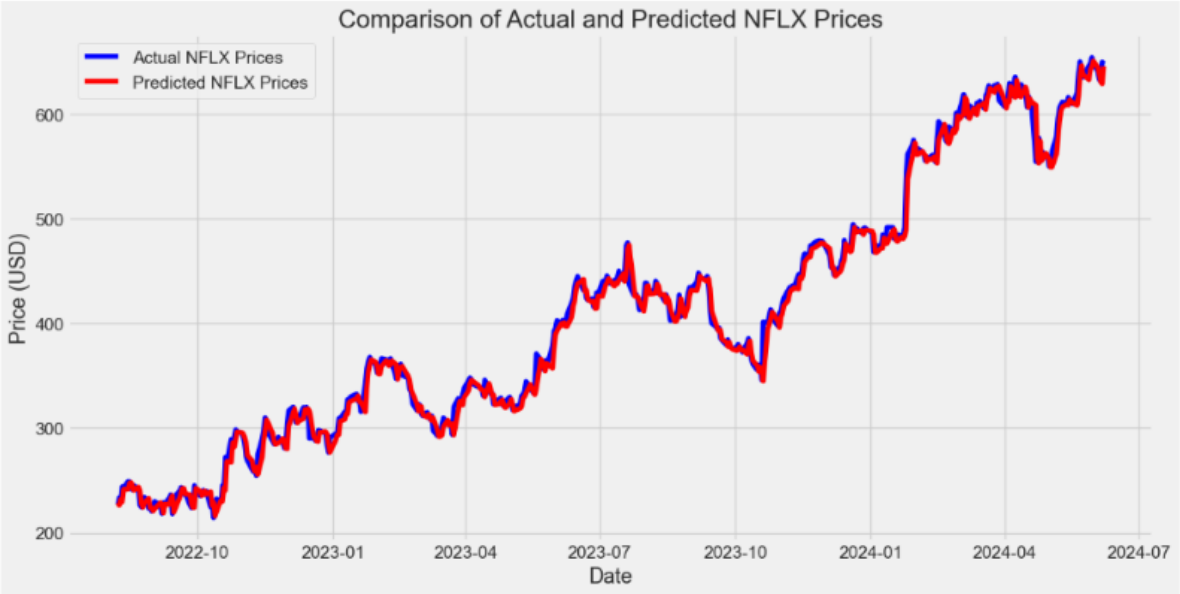


Figura 4.16: Cel mai performant model (LSTM)

Capitolul 5

Concluzii

Această lucrare a explorat în detaliu metode avansate de predicție a prețului acțiunilor, utilizând tehnici moderne de învățare automată și de analiză a seriilor de timp. Am evidențiat importanța efectuării unei analize EDA înainte de implementarea algoritmilor de predicție, dar și complexitatea datelor din domeniul financiar, un motiv important în alegerea acestor metode.

Rezultatele obținute arată că modelele de învățare automată oferă un avantaj semnificativ în realizarea de predicții, datorită abilității lor de a studia datele din trecut și identifica tendințe greu de observat prin metode tradiționale. Astfel, am arătat că folosirea de tehnici din știința datelor în domeniul financiar pot conduce la perspective noi și la decizii mai informate în strategiile de investiție.

5.1 Posibile îmbunătățiri

În primul rând, datele de intrare sunt destul de limitate, având doar câteva coloane din setul de date importante în predicția modelelor. Acestea ar putea fi extinse prin integrarea unor noi tehnologii precum :

- Analiză de sentiment din știri legate de compania vizată;
- Studierea postărilor acesteia de pe diverse platforme de social media precum Instagram, Twitter, Facebook;
- Includerea altor indicatori economici precum cifra de afaceri, inflația, numărul de angajări din ultima perioada, investițiile corporației;
- Utilizarea unor date cu frecvență mai mare, de exemplu la fiecare oră, ar putea să adauge un plus în captarea tendințelor și anomaliilor.

O altă îmbunătățire care ar putea fi crucială în obținerea unor rezultate mai bune este integrarea de caracteristici mai avansate în antrenare. Câteva exemple sunt indicatorii

MACD, retragerile Fibonacci sau benzile Bollinger. De asemenea, printr-un algoritm de feature engineering, aceste caracteristici ar putea fi selectate sau eliminate în mod automat în urma rezultatelor din testare.

În plus, modelele utilizate sunt destul de performante, însă există o multitudine de alți algoritmi și tehnici ce pot fi aplicate pentru obținerea unor rezultate mai bune. Alte arhitecturi precum Transformeri sau GRU, dar și modelele hibrid au capacitatea mai mare de învățare. Pentru o eficientă și mai mare, se pot utiliza algoritmi de automatizare pentru selecția hiperparametrilor, precum optimizarea Bayesiana, Algoritmi genetici sau Hyperopt.

În concluzie, prezicerea prețului acțiunilor este o provocare complexă și necesită mult timp și resurse datorită multitudinii de factori ce influențează aceste valori. Totuși, prin integrarea de tehnici tot mai avansate, am reușit să ajung la rezultate ce pot fi utilizate chiar în strategii reale de investiție.

Bibliografie

[1] Zexin Hu, Yiqi Zhao și Matloob Khushi, „A Survey of Forex and Stock Price Prediction Using Deep Learning”, în (2021), URL: https://doi.org/10.3390/asi4010009.

[2] Indu Kumar, Kiran Dogra, Chetna Utreja și Premlata Yadav, „A Comparative Study of Supervised Machine Learning Algorithms for Stock Market Trend Prediction”, în (2018), URL: https://ieeexplore.ieee.org/document/8473214.

[3] Payal Soni, Yogya Tewari și Prof. Deepa Krishnan, „Machine Learning Approaches in Stock Price Prediction: A Systematic Review”, în (2022), URL: https://iopscience.iop.org/article/10.1088/1742-6596/2161/1/012065/pdf.

[4] Yoojeong Song și Jongwoo Lee, „Design of stock price prediction model with various configurations of input features”, în (2019), URL: https://www.researchgate.net/publication/337880880_Design_of_stock_price_prediction_model_with_various_configuration_of_input_features.

[5] Xingzhou L., Hong R. și Yujun Z., „Predictive Modeling of Stock Indexes Using Machine Learning and Information Theory”, în (2019), URL: https://www.researchgate.net/publication/336201707_Predictive_Modeling_of_Stock_Indexes_Using_Machine_Learning_and_Information_Theory.

[6] Joost N. Kok, Egbert J. W. Boers, Walter A. Kosters și Peter van der Putten, „ARTIFICIAL INTELLIGENCE: DEFINITION, TRENDS, TECHNIQUES, AND CASES”, în (2009), URL: https://www.eolss.net/Sample-Chapters/C15/E6-44.pdf.

[7] Jafar Alzubi, Anand Nayyar și Akshi Kumar, „Machine Learning from Theory to Algorithms: An Overview”, în (2018), URL: https://iopscience.iop.org/article/10.1088/1742-6596/1142/1/012012/pdf.

[8] Mehtabhorn Obthong, Nongnuch Tantisantiwong, Watthanasak Jeamwatthanachai și Gary Wills, „A Survey on Machine Learning for Stock Price Prediction: Algorithms and Techniques”, în (2020), URL: https://eprints.soton.ac.uk/437785/1/FEMIB_2020_6.pdf.

[9] WL Martinez, AR Martinez și J Solka, „Exploratory Data Analysis with MATLAB”, în (2017), URL: https://www.taylorfrancis.com/books/mono/10.1201/9781315366968/exploratory-data-analysis-matlab-jeffrey-solka-wendy-martinez-angel-martinez.

[10] Devore J L, „Probability And Statistics For Engineering And The Sciences, 8 Edition”, în (2012), URL: https://faculty.ksu.edu.sa/sites/default/files/probability_and_statistics_for_engineering_and_the_sciences.pdf.

[11] Boston University, „Simple Linear Regression”, în (2016), URL: https://sphweb.bumc.bu.edu/otlt/MPH-Modules/BS/R/R5_Correlation-Regression/R5_Correlation-Regression4.html.

[12] Robert Tibshirani, „Regression Shrinkage and Selection via the Lasso”, în (1996), URL: https://webdoc.agsci.colostate.edu/koontz/arec-econ535/papers/Tibshirani%20(JRSS-B%201996).pdf.

[13] Safwan Mahmood, Mohd Fadzil Hassan și Said Jadid Abdulkadir, „RNN-LSTM: From applications to modeling techniques and beyond—Systematic review”, în (2024), URL: https://www.sciencedirect.com/science/article/pii/S1319157824001575.

[14] Xin Wei, Lulu Zhang și Hao-Qing Yang, „Machine learning for pore-water pressure time-series prediction: Application of recurrent neural networks”, în (2021), URL: https://www.sciencedirect.com/science/article/pii/S1674987120301134?via%3Dihub.

[15] Sepp Hochreiter și Jürgen Schmidhuber, „Long Short-Term Memory”, în (1997), URL: https://www.bioinf.jku.at/publications/older/2604.pdf.

[16] David Abugaber, „Using ARIMA for Time Series Analysis”, în (2017), URL: https://ademos.people.uic.edu/Chapter23.html.

[17] Geeksforgeeks, „What is Python? it’s Uses and Applications”, în (2024), URL: https://www.geeksforgeeks.org/what-is-python/.

[18] Mark Lutz, „Learning Python, Fourth Edition”, în (2009), URL: https://cfm.ehu.es/ricardo/docs/python/Learning_Python.pdf.

[19] Realpython, „NumPy Tutorial: Your First Steps Into Data Science in Python”, în (2020), URL: https://realpython.com/numpy-tutorial/#choosing-numpy-the-benefits.

[20] Nicolai Berg Andersen, „What Is Pandas?”, în (2023), URL: https://builtin.com/data-science/pandas.

[21] Geeksforgeeks, „Introduction to Matplotlib”, în (2024), URL: https://www.geeksforgeeks.org/python-introduction-matplotlib/.

[22] Geeksforgeeks, „Introduction to Seaborn – Python”, în (2023), URL: https://www.geeksforgeeks.org/introduction-to-seaborn-python/.

[23] Nvidia, „Scikit-learn”, în (2024), URL: https://www.geeksforgeeks.org/python-introduction-matplotlib/.

[24] DeepAI, „Keras”, în (2019), URL: https://deepai.org/machine-learning-glossary-and-terms/keras.

[25] GlobalData, „Netflix Loses Almost a Million Subscribers in Last Quarter”, în (2022), URL: https://www.globaldata.com/data-insights/technology--media-and-telecom/netflix-loses-almost-a-million-subscribers-in-last-quarter/.

[26] The Motley Fool, „The Safest and Riskiest FAANG Stocks to Buy Right Now”, în (2022), URL: https://www.fool.com/investing/2022/11/21/the-safest-and-riskiest-faang-stocks-to-buy-right/.