Predicția Prețului Monedei Curve folosind Inteligența Artificială

1. Introducere

Context și obiective:

Moneda Curve (CRV) este un token utilizat în protocolul Curve Finance, un schimb descentralizat specializat în stablecoins. Proiectul nostru își propune să prezică prețul viitor al monedei CRV folosind modele de inteligență artificială pentru a ajuta investitorii să ia decizii informate. Principalul obiectiv este de a dezvolta un model precis care poate oferi previziuni fiabile pe termen scurt și lung.

Tehnologii folosite

DataSpell de la JetBrains pentru mediu de dezvoltare integrat (IDE)

DataSpell este un IDE de la JetBrains, special conceput pentru analiza și știința datelor. Acesta oferă o gamă largă de caracteristici care facilitează dezvoltarea și testarea rapidă a codului Python, inclusiv:

- Interfață de utilizator intuitivă: DataSpell oferă o interfață prietenoasă care integrează funcționalități avansate, făcându-l ușor de utilizat chiar și pentru cei care nu sunt experți în programare.

- Suport Jupyter Notebook: Integrează complet notebook-urile Jupyter, permițându-ți să rulezi celule de cod, să vizualizezi rezultate și să gestionezi fișierele direct din IDE.

- Funcționalități de editare inteligente: Auto-completare, refactorizare de cod și verificare sintactică, care ajută la scrierea și menținerea unui cod curat și eficient.

- Integrare cu controlul versiunilor: Suport pentru Git, SVN și alte sisteme de control al versiunilor, facilitând colaborarea și gestionarea versiunilor proiectului.

Python pentru programare

Python este un limbaj de programare de nivel înalt, foarte popular în domeniul științei datelor și al inteligenței artificiale datorită:

- Sintaxă simplă și clară: Python este ușor de învățat și de utilizat, ceea ce accelerează dezvoltarea proiectelor.

- Librării extinse: Python dispune de o vastă colecție de librării și framework-uri dedicate analizei datelor, învățării automate și vizualizării datelor.

- Comunitate activă: Există o comunitate globală de dezvoltatori și cercetători care contribuie la dezvoltarea și îmbunătățirea continuă a ecosistemului Python.

Librării AI/ML

- scikit-learn: O librărie robustă și versatilă pentru învățare automată, care include un set larg de algoritmi pentru clasificare, regresie, clustering și reducerea dimensionalității. Scikit-learn este foarte apreciată pentru simplitatea și coerența API-ului său.

- Utilizare în proiect: Implementarea și evaluarea modelelor de regresie liniară, Ridge și Lasso.

- pandas: O librărie esențială pentru manipularea și analiza datelor structurale. Pandas oferă structuri de date rapide, flexibile și expresive pentru a facilita manipularea și analiza datelor relaționale sau etichetate.

- Utilizare în proiect: Citirea și preprocesarea datelor din fișierele CSV, manipularea și transformarea seturilor de date.

- numpy: Biblioteca fundamentală pentru calcul numeric în Python, care furnizează suport pentru array-uri și matrice de mari dimensiuni, împreună cu o colecție vastă de funcții matematice de nivel înalt pentru a opera pe aceste array-uri.

- Utilizare în proiect: Operații numerice și calcul matematic avansat pe seturile de date.

- matplotlib: O bibliotecă de bază pentru vizualizarea datelor în Python, care permite crearea de grafice și diagrame statice, animate și interactive.

- Utilizare în proiect: Vizualizarea corelațiilor și a rezultatelor modelului.

- seaborn: O extensie a Matplotlib care oferă o interfață de nivel înalt pentru desenarea graficelor statistice atractive și informative.

- Utilizare în proiect: Vizualizarea corelațiilor dintre variabile prin intermediul hărților de căldură și a altor grafice statistice.

MySQL pentru gestionarea bazei de date

MySQL este un sistem de gestionare a bazelor de date relaționale, foarte popular și open-source, folosit pentru stocarea și gestionarea datelor.

- Scalabilitate și performanță: MySQL poate gestiona volume mari de date și poate scala pentru a se adapta nevoilor crescânde ale aplicației.

- Securitate: Oferă mecanisme robuste de securitate pentru protejarea datelor, inclusiv autentificarea utilizatorilor și controlul accesului la date.

- Integrare cu alte tehnologii: MySQL se integrează bine cu diverse limbaje de programare și framework-uri, făcându-l ușor de utilizat într-o varietate de aplicații.

- Comunitate și suport: Dispune de o comunitate largă și activă care contribuie la dezvoltarea și îmbunătățirea continuă a sistemului, precum și suport profesional disponibil pentru utilizatorii enterprise.

Utilizare în proiect:

- Stocarea datelor istorice ale prețurilor CRV.

- Gestionarea și interogarea datelor pentru a le face accesibile proceselor de preprocesare și analiză.

Prin utilizarea acestor tehnologii și instrumente, proiectul nostru a beneficiat de un mediu de dezvoltare puternic și flexibil, capabil să gestioneze eficient datele și să implementeze modele de predicție precise și interpretabile.

2. Colectarea și prelucrarea datelor

Sursele datelor:

Datele istorice ale prețurilor pentru CRV au fost colectate dintr-un fișier CSV. Datele includ prețuri zilnice, volume tranzacționate și capitalizare de piață.

Structura bazei de date:

Datele sunt stocate într-o bază de date MySQL. Tabelele includ următoarele câmpuri:

- `date`: Data înregistrării.

- `price`: Prețul zilnic al CRV.

- `volume`: Volumul zilnic tranzacționat.

- `market\_cap`: Capitalizarea de piață zilnică.

Preprocesarea datelor:

Codul de preprocesare a datelor este următorul:

```python

import pandas as pd

import numpy as np

Citirea datelor din fișierul CSV

df = pd.read\_csv(r"C:\Users\Sergiu\OneDrive - Technical University of Cluj-Napoca\Desktop\rudi\CRV.csv")

Eliminarea coloanei 'ticker' deoarece nu este necesară pentru model

df = df.drop('ticker', axis=1)

Calcularea matricei de corelație

corr = df.select\_dtypes('number').corr()

Vizualizarea matricei de corelație

import seaborn as sns

import matplotlib.pyplot as plt

cmap = sns.diverging\_palette(100, 5, as\_cmap=True)

def magnify():

return [dict(selector="th",

props=[("font-size", "7pt")]),

dict(selector="td",

props=[('padding', "0em 0em")]),

dict(selector="th:hover",

props=[("font-size", "12pt")]),

dict(selector="tr:hover td:hover",

props=[('max-width', '200px'),

('font-size', '12pt')])

]

corr.style.background\_gradient(cmap, axis=1) \

.format(precision=3) \

.set\_properties(\*\*{'max-width': '120px', 'font-size': '10pt'}) \

.set\_caption("Hover to magify") \

.set\_table\_styles(magnify())

```

3. Modelul de inteligență artificială

Alegerea modelului:

Pentru a realiza predicții asupra prețului CRV, am utilizat mai multe modele de regresie, incluzând regresie liniară, Ridge și Lasso. Am comparat performanța acestor modele folosind diverse metrice de evaluare.

Arhitectura modelului:

Modelele utilizate sunt:

Regresie Liniară:

Regresia liniară este una dintre cele mai simple și interpretabile tehnici de predicție. Avantajul său principal este ușurința de implementare și interpretare. Regresia liniară presupune o relație liniară între variabilele independente și variabila dependentă. În contextul acestui proiect, modelul de regresie liniară ne-a oferit o bază solidă pentru predicția prețului CRV.

Regresie Ridge:

Regresia Ridge este o variantă a regresiei liniare care include o penalizare pentru coeficienți mari. Aceasta ajută la reducerea varianței modelului și la prevenirea overfitting-ului, în special atunci când avem un număr mare de caracteristici sau multicolinearitate între acestea. Penalizarea adăugată în regresia Ridge contribuie la stabilizarea modelului și îmbunătățirea generalizării acestuia.

Regresie Lasso:

Regresia Lasso este similară cu regresia Ridge, dar utilizează o penalizare L1 care poate reduce coeficienții unor caracteristici la zero. Acest lucru este util pentru selecția caracteristicilor, reducând complexitatea modelului și îmbunătățind interpretabilitatea. Prin utilizarea regresiei Lasso, am reușit să identificăm caracteristicile cele mai relevante pentru predicția prețului CRV, contribuind astfel la un model mai simplu și mai eficient.

Antrenarea modelului:

Am împărțit setul de date în seturi de antrenament și testare, folosind 80% din date pentru antrenament și 20% pentru testare. Codul pentru antrenarea și evaluarea modelului este următorul:

```python

from sklearn.linear\_model import LinearRegression, Ridge, Lasso

from sklearn.metrics import mean\_squared\_error, r2\_score, mean\_absolute\_error

Definirea modelului de regresie liniară

lr = LinearRegression()

lr.fit(X\_train, Y\_train)

Predicțiile modelului de regresie liniară

y\_lr\_train\_pred = lr.predict(X\_train)

y\_lr\_test\_pred = lr.predict(X\_test)

Evaluarea modelului de regresie liniară

lr\_train\_mse = mean\_squared\_error(Y\_train, y\_lr\_train\_pred)

lr\_train\_mae = mean\_absolute\_error(Y\_train, y\_lr\_train\_pred)

lr\_train\_r2 = r2\_score(Y\_train, y\_lr\_train\_pred)

lr\_test\_mse = mean\_squared\_error(Y\_test, y\_lr\_test\_pred)

lr\_test\_mae = mean\_absolute\_error(Y\_test, y\_lr\_test\_pred)

lr\_test\_r2 = r2\_score(Y\_test, y\_lr\_test\_pred)

print("Regresie Liniară - Valori Antrenament: MSE:", lr\_train\_mse, "MAE:", lr\_train\_mae, "R2:", lr\_train\_r2)

print("Regresie Liniară - Valori Testare: MSE:", lr\_test\_mse, "MAE:", lr\_test\_mae, "R2:", lr\_test\_r2)

Definirea modelului de regresie Ridge

ridge = Ridge(alpha=1.0)

ridge.fit(X\_train, Y\_train)

Predicțiile modelului de regresie Ridge

y\_ridge\_train\_pred = ridge.predict(X\_train)

y\_ridge\_test\_pred = ridge.predict(X\_test)

Evaluarea modelului de regresie Ridge

ridge\_train\_mse = mean\_squared\_error(Y\_train, y\_ridge\_train\_pred)

ridge\_train\_mae = mean\_absolute\_error(Y\_train, y\_ridge\_train\_pred)

ridge\_train\_r2 = r2\_score(Y\_train, y\_ridge\_train\_pred)

ridge\_test\_mse = mean\_squared\_error(Y\_test, y\_ridge\_test\_pred)

ridge\_test\_mae = mean\_absolute\_error(Y\_test, y\_ridge\_test\_pred)

ridge\_test\_r2 = r2\_score(Y\_test, y\_ridge\_test\_pred)

print("Regresie Ridge - Valori Antrenament: MSE:", ridge\_train\_mse, "MAE:", ridge\_train\_mae, "R2:", ridge\_train\_r2)

print("Regresie Ridge - Valori Testare: MSE:", ridge\_test\_mse, "MAE:", ridge\_test\_mae, "R2:", ridge\_test\_r2)

Definirea modelului de regresie Lasso

lasso = Lasso(alpha=1.0)

lasso.fit(X\_train, Y\_train)

# Predicțiile modelului de regresie Lasso

y\_lasso\_train\_pred = lasso.predict(X\_train)

y\_lasso\_test\_pred = lasso.predict(X\_test)

Evaluarea modelului de regresie Lasso

lasso\_train\_mse = mean\_squared\_error(Y\_train, y\_lasso\_train\_pred)

lasso\_train\_mae = mean\_absolute\_error(Y\_train, y\_lasso\_train\_pred)

lasso\_train\_r2 = r2\_score(Y\_train, y\_lasso\_train\_pred)

lasso\_test\_mse = mean\_squared\_error(Y\_test, y\_lasso\_test\_pred)

lasso\_test\_mae = mean\_absolute\_error(Y\_test, y\_lasso\_test\_pred)

lasso\_test\_r2 = r2\_score(Y\_test, y\_lasso\_test\_pred)

print("Regresie Lasso - Valori Antrenament: MSE:", lasso\_train\_mse, "MAE:", lasso\_train\_mae, "R2:", lasso\_train\_r2)

print("Regresie Lasso - Valori Testare: MSE:", lasso\_test\_mse, "MAE:", lasso\_test\_mae, "R2:", lasso\_test\_r2)

```

4. Evaluarea și rezultate

Metrici de evaluare:

Am folosit Mean Squared Error (MSE), Mean Absolute Error (MAE) și R-squared (R2) pentru a evalua performanța modelelor.

Rezultatele obținute:

Rezultatele evaluării pentru fiecare model sunt următoarele:

- Regresie Liniară\*\*:

A screen shot of a computer

Description automatically generated

- \*\*Regresie Ridge\*\*:

A screenshot of a computer

Description automatically generated

- \*\*Regresie Lasso\*\*:

A screen shot of a computer

Description automatically generated

5. Concluzii și perspective viitoare

Concluzii:

Proiectul nostru a demonstrat că este posibil să prezicem prețul monedei CRV cu un grad rezonabil de acuratețe folosind modele de regresie. Aceste predicții pot ajuta investitorii să ia decizii informate pe baza analizei datelor istorice.

Limitări și îmbunătățiri:

Una dintre limitările modelului este că nu ia în considerare toți factorii de piață, cum ar fi știrile financiare sau sentimentele pieței. Îmbunătățirile viitoare ar putea include integrarea acestor factori și testarea altor tipuri de modele.

Posibile aplicații:

Metodologia utilizată poate fi aplicată și pentru alte monede digitale sau active financiare, oferind un instrument valoros pentru analize predictive în domeniul financiar.

6. Anexe și Resurse

1. Regresie liniară

- [Scikit-learn Documentation - Linear Regression](https://scikit-learn.org/stable/modules/generated/sklearn.linear\_model.LinearRegression.html): Documentația oficială a bibliotecii Scikit-learn pentru regresia liniară.

- [Wikipedia - Linear Regression](https://en.wikipedia.org/wiki/Linear\_regression): Articolul Wikipedia despre regresia liniară, care include formule matematice și explicații teoretice.

- [Statistical Learning - Linear Regression](https://web.stanford.edu/~hastie/ElemStatLearn/): Carte gratuită despre învățarea statistică, capitolul 3 fiind dedicat regresiei liniare (The Elements of Statistical Learning, de Hastie, Tibshirani, Friedman).

2. Regresie Ridge

- [Scikit-learn Documentation - Ridge Regression](https://scikit-learn.org/stable/modules/generated/sklearn.linear\_model.Ridge.html): Documentația oficială pentru regresia Ridge în Scikit-learn.

- [Wikipedia - Ridge Regression](https://en.wikipedia.org/wiki/Tikhonov\_regularization): Articolul Wikipedia despre regularizarea Tikhonov (Ridge Regression).

- [Ridge Regression Tutorial](https://towardsdatascience.com/ridge-regression-theory-algorithm-and-examples-e89a2aaf0938): Un articol detaliat pe Towards Data Science care explică teoria și implementarea regresiei Ridge.

3. Regresie Lasso

- [Scikit-learn Documentation - Lasso Regression](https://scikit-learn.org/stable/modules/generated/sklearn.linear\_model.Lasso.html): Documentația oficială pentru regresia Lasso în Scikit-learn.

- [Wikipedia - Lasso (statistics)](https://en.wikipedia.org/wiki/Lasso\_(statistics)): Articolul Wikipedia despre Lasso, incluzând explicații teoretice și aplicații.

- [Lasso Regression Explained](https://towardsdatascience.com/lasso-regression-explained-9d3c9439d0e0): Un articol pe Towards Data Science care oferă o explicație detaliată și exemple pentru regresia Lasso.

4. Predicții financiare

- [Investopedia - Financial Forecasting](https://www.investopedia.com/terms/f/financial-forecasting.asp): Articol despre prognoza financiară și importanța acesteia.

- [Quantitative Finance Using Python](https://www.amazon.com/Quantitative-Finance-Using-Python-Woo-ebook/dp/B00NA9MNY4): Carte care oferă o introducere în finanțele cantitative utilizând Python.

- [Algorithmic Trading and Quantitative Strategies](https://web.stanford.edu/class/msande448/): Curs online gratuit de la Stanford despre strategiile de tranzacționare algoritmică și metode cantitative.

5. Biblioteci și documentație Python

- [Pandas Documentation](https://pandas.pydata.org/docs/): Documentația oficială pentru Pandas, biblioteca utilizată pentru manipularea datelor.

- [Numpy Documentation](https://numpy.org/doc/stable/): Documentația oficială pentru Numpy, biblioteca utilizată pentru calcule numerice.

- [Matplotlib Documentation](https://matplotlib.org/stable/contents.html): Documentația oficială pentru Matplotlib, biblioteca utilizată pentru vizualizarea datelor.

- [Seaborn Documentation](https://seaborn.pydata.org/): Documentația oficială pentru Seaborn, biblioteca utilizată pentru vizualizarea datelor statistice.