Prezentare proiect Învățare Automată

Student: Petre Tiberiu

1. *Descrierea pe scurt a celor două modele de învățare alese*

Modelele de învățare alese sunt modele de regresie, acestea fiind: modelul de regresie polinomial de ordin doi și metoda de regresie ElasticNet.

Modelul de regresie liniar este definit prin formula matematică: $\theta_0 x_1 + \theta_1 x_2 + \theta_2 x_3 + \cdots + \theta_{n-1} x_n + \epsilon = \hat{y}$

Acesta încercă să găsească relația cea mai bună dintre valorile lui x a.î să se obțină o valoare foarte apropiată de valoarea reală y, această valoare se mai numește și valoare prezisă sau $\hat{y}$

Modelul polinomial de ordin/grad doi este asemenea modelului liniar, diferența dintre acestea fiind utilizarea pătratelor lui x pentru o mai bună aproximare polinomială a lui y.

Formula este: $\theta_0 x_1 + \theta_0' x_1^2 + \theta_1 x_2 + \theta_1' x_2^2 + \cdots + \theta_{n-1} x_n + \theta_{n-1}' x_n^2 = \hat{y}$

Metoda de regresie ElasticNet folosește o combinație dintre regresorul Ridge și cel Lasso pentru a prezice valorile lui y.

Modelul Ridge aduce nou modelului de regresie liniar adăugării la suma pătratelor erorilor $\left|\left|y - X\theta^0\right|\right|^2$ funcția de penalizare $\left|\left|\theta\right|\right|^2$ cu θ vectorul dat de coordonatele 0,1,2,...,n-1 ale parametrilor x.

Modelul Lasso, spre deosebire de Ridge, $\left|\left|\theta\right|\right|$ de normă l2 este înlocuită de cea de normă l1 determinată folosind distanțele Manhattan dintre valoarea lui θ și $\hat{\theta}$: $d(\theta, \hat{\theta}) = \left|\left|\hat{\theta} - \theta\right|\right|$

În final, **metoda ElasticNet** combină cele două modele folosind o combinație liniară între valoarea dată de Ridge și cea dată de Lasso

2. *Descriere bazei de date și detaliile privind prelucrarea bazei de date*

Baza de date utilizată, preluată de pe Kaggle, este cea a numărului de cazuri de Covid-19 înregistrate în decursul unei perioade. Parametrii de intrare sunt, de bază, numărul de cazuri împărțite pe țari și distribuite pe zile. Acest număr de cazuri este un număr de persoane bolnave în acea zi.

Pentru a putea realiza o regresie peste aceste valori s-a luat intervale săptămânale a diferențelor zilnice a numărului de cazuri. Pentru fiecare săptămână s-a identificat valoarea la început, cea de sfârșit și valorile minime și maxime dintre acestea. În final se obține un grafic asemănător celor financiare unde numărul de cazuri, exprimat în număr de persoane per 1000 de loc, este prezentat săptămânal pentru fiecare țară.

```
SNo,ObservationDate,Province/State,Country/Region,Last Update,Confirmed,Deaths,Recovered
236014,02/27/2021,Zeeland,Netherlands,2021-02-28 05:22:20,16480.0,178.0,0.0
236015,02/27/2021,Zhejiang,Mainland China,2021-02-28 05:22:20,1321.0,1.0,1314.0
236016,02/27/2021,Zhytomyr Oblast,Ukraine,2021-02-28 05:22:20,50582.0,834.0,44309.0
236017,02/27/2021,Zuid-Holland,Netherlands,2021-02-28 05:22:20,255335.0,3732.0,0.0
```

Număr de cazuri în setul de date inițial, ultimele 4 înregistrări

```
Week,Country,Open,High,Low,Close
48,Romania,3174.0,3174.0,1313.0,2676.0
49,Romania,2752.0,2797.0,1319.0,2797.0
50,Romania,3048.0,3048.0,1331.0,2676.0
51,Romania,2815.0,3382.0,1634.0,3382.0
```

Număr de cazuri confirmate în România, după prelucrare, în ultimele 4 săptămâni

3. *Care sunt variabilele dependente și care sunt cele independente*

În urma prelucrării inițiale, se poate observa că putem prezice relativ ușor numărul de cazuri de la începutul săptămânii următoare în funcție de săptămâna curentă.

Astfel, **variabilele independente** sunt coloanele “Open”, ”High”, “Low”, “Close” din săptămâna curentă și **variabila dependentă** este coloana “Open” din săptămâna imediat următoare.

4. *Elemente de limbaj nou utilizate*

Elemente noi de limbaj nu au fost utilizate fată de semestrul trecut.

S-au utilizat bibliotecile:

- Scikit-Learn (pentru învățarea automată)
- Matplotlib, Plotly (pentru afișare de grafice)
- NumPy (pentru prelucrare de matrice și numere)

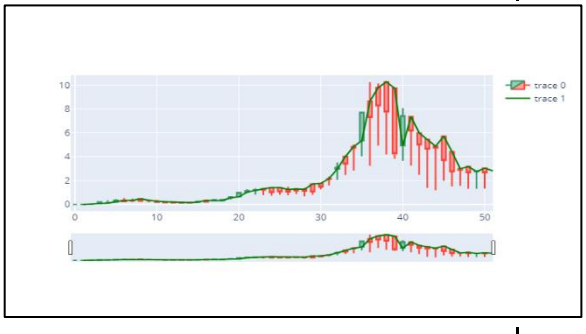
5. *Parametrii statistici aleși pentru evaluarea modelelor*

Pentru evaluarea modelului s-au folosit funcțiile statistice **Mean Squared Error**, **Rooted Mean Squared Error** și **coeficientul de determinare R2**. Aceste funcții s-au aplicat parametrilor y și $\hat{y}$

6. *Codul sursă*

```
import pandas as pd
import numpy as np
from plotly import graph_objects as go

tara = "Romania"
csv = pd.read_csv("./covid19_final.csv", index_col="Country")
romania = csv.loc[tara,:].set_index("Week")
romania = romania/1000

# împărțire pe date de intrare și date de ieșire
X = [np.array(romania.iloc[0,])]
y = [[romania.iloc[1,0]]]
for i in range(1, len(romania.index)-1):
    week = romania.iloc[i,]
    week_open = romania.iloc[i+1,0]
    X.append(np.array(week))
    y.append([week_open])
X = np.array(X)
y = np.array(y)
```

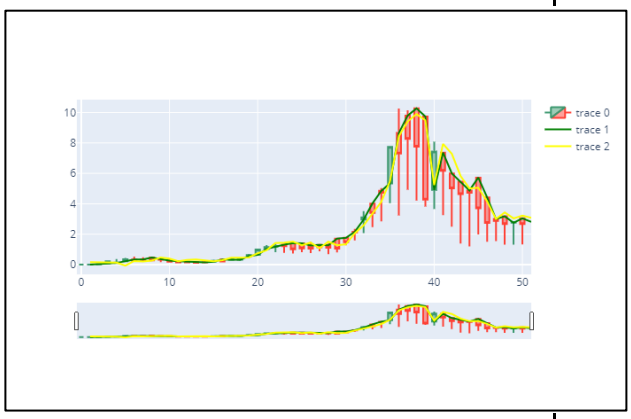
```
# afișare grafic original
fig = go.Figure()
fig.add_trace(go.Candlestick(
    open=X[:,0],
    high=X[:,1],
    low=X[:,2],
    close=X[:,3]
))
fig.add_trace(go.Scatter(
    x=romania.index+1,
    y=y.flatten(),
    line=dict(color='green', width=2)
))
fig.show()
```



```
# Regresia polinomiala multiplă de grad doi
from sklearn.linear_model import LinearRegression, ElasticNetCV
from sklearn.metrics import mean_squared_error, r2_score

X2 = np.power(X, 2)
Xnou = np.hstack((X, X2))

reg2 = LinearRegression()
reg2.fit(Xnou, y)
print('coeficienti',reg2.coef_)
print('np-shape(reg2.coef_)=',np.shape(reg2.coef_) )
print('Interceptie',reg2.intercept_)

y_pred = reg2.predict(Xnou)
print('MSE: ', mean_squared_error(y, y_pred))
print('RMSE: ', np.sqrt(mean_squared_error(y, y_pred)))
print('Coef determinare: ', r2_score(y, y_pred))
```

coeficienti [[1.38088675 -1.79056495 0.95514765 1.02853021 -0.08344871 0.11444741 -0.15316688 0.0161841]]
np-shape(reg2.coef_)= (1, 8)
Interceptie [0.172307]
MSE: 0.10883889763441772
RMSE: 0.32990740766829973
Coef determinare: 0.9862915114344368

```
# Grafic regresie polinomială multiplă de grad doi
fig = go.Figure()
fig.add_trace(go.Candlestick(
    open=X[:,0],
    high=X[:,1],
    low=X[:,2],
    close=X[:,3]
))
fig.add_trace(go.Scatter(
    x=romania.index+1,
    y=y.flatten(),
    line=dict(color='green', width=2)
))
fig.add_trace(go.Scatter(
    x=romania.index+1,
    y=y_pred.flatten(),
    line=dict(color='yellow', width=2)
))
fig.show()
```

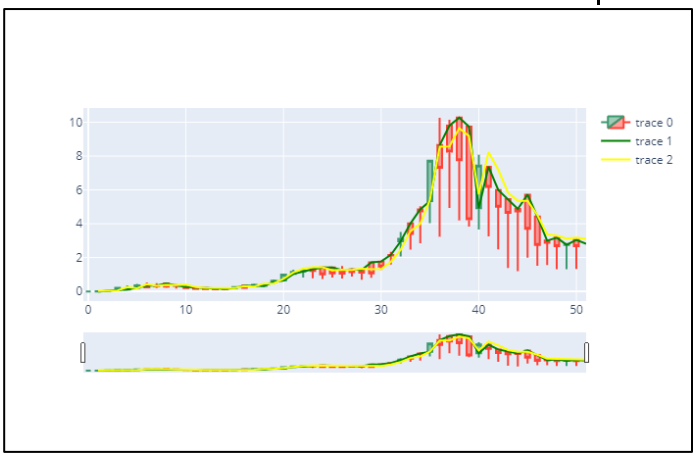



```
# Metoda de regresie ElasticNet
regElasticNet = ElasticNetCV(cv=5, random_state=0) # cv=n_folds in KFold validation
regElasticNet.fit(X, y.flatten())

print('coeficienti',regElasticNet.coef_)
print('np-shape(regElasticNet.coef_)=',np.shape(regElasticNet.coef_) )
print('Interceptie',regElasticNet.intercept_)

y_pred = regElasticNet.predict(X)
print('MSE: ', mean_squared_error(y, y_pred))
print('RMSE: ', np.sqrt(mean_squared_error(y, y_pred)))
print('Coef determinare: ', r2_score(y, y_pred))
```

coeficienti [0.09227376 0.05898774 0.03492392 0.96191399]
np-shape(regElasticNet.coef_)= (4,)
Interceptie 0.023508201965069198

MSE: 0.1550944954659301
RMSE: 0.39382038477703274
Coef determinare: 0.9804655213909098

```
# Grafic regresie ElasticNet
fig = go.Figure()
fig.add_trace(go.Candlestick(
    open=X[:,0],
    high=X[:,1],
    low=X[:,2],
    close=X[:,3]
))
fig.add_trace(go.Scatter(
    x=romania.index+1,
    y=y.flatten(),
    line=dict(color='green', width=2)
))
fig.add_trace(go.Scatter(
    x=romania.index+1,
    y=y_pred.flatten(),
    line=dict(color='yellow', width=2)
))
fig.show()
```



7. *Concluzii finale*

	Model regresie	θ_0	θ_1	θ_2	θ_3	ϵ	MSE	Coeficient determinare
0	**Regresie polinomială multiplă**	1,380887	-1,790565	0,955148	1,028530	0,172307	**0,108839**	**0,986292**
1	Regresie ElasticNet	0,092274	0,058988	0,034924	0,961914	0,023508	**0,155094**	**0,980466**

- Din tabelul de mai sus se poate observa că modelul de **regresie polinomială** este cel mai bun, cu o eroare de 0.108 și un coeficient de determinare de 0.9862 apoi **ElasticNet** cu o eroare de 0.155 si un coeficient de determinare de 0.9804.
- Cu cat **eroarea medie pătratică** este mai mică iar **coeficientul de determinare** mai mare (1 este cel mai bun), cu atât **regresia liniară este cea mai sigură** de folosit pentru preziceri imediate.