

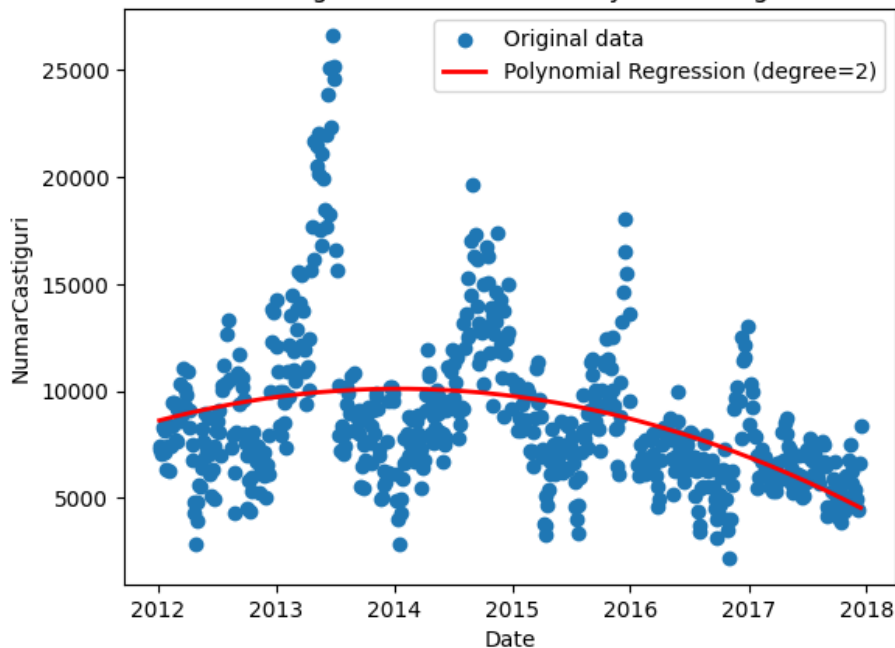


Predictia numerelor castigatoare

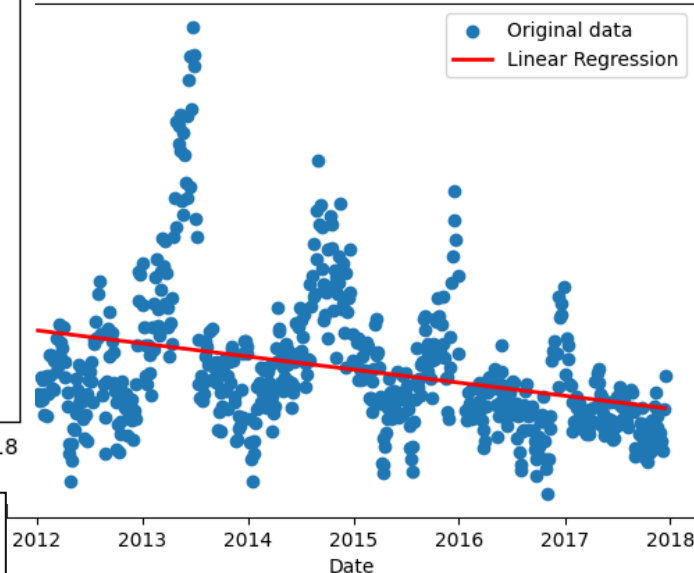
Continuare/ corectare

Denisa Dumitrescu

NumarCastiguri over Time with Polynomial Regression

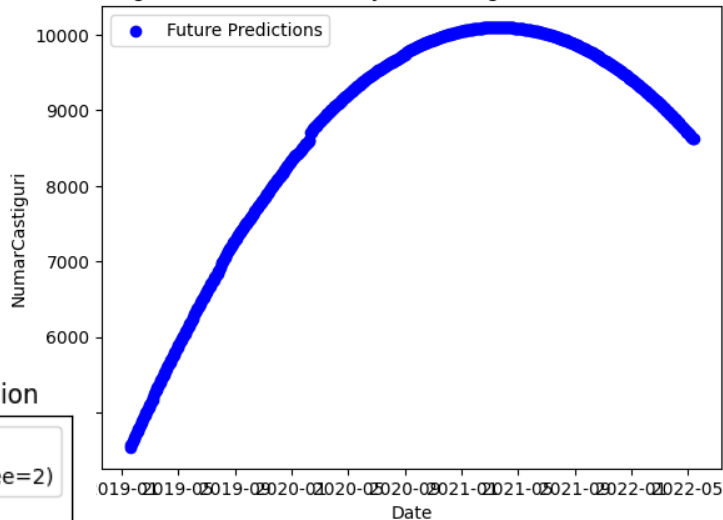


NumarCastiguri over Time with Linear Regression

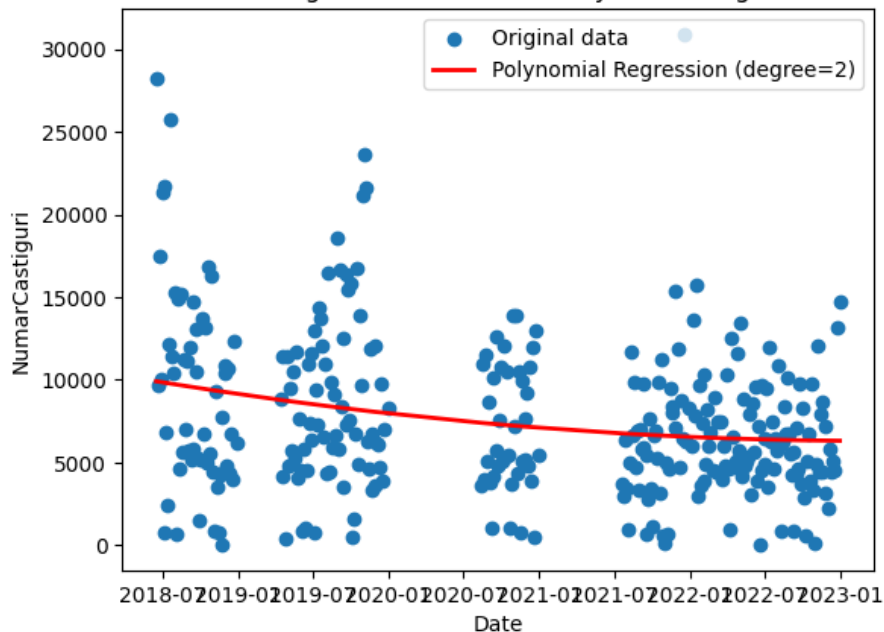


Graficul arata modul in care regresia polinomiala aproximează perioadele de creștere și descreștere ale numărului de câștigători. Comparativ cu regresia liniară aplicată pe același set de date, se observă că regresia polinomială se mulează mai bine pe setul de date. Linia mai curbată a regresiei polinomiale se potrivește mai bine cu zonele cele mai populate ale datelor decât linia mai rigidă a regresiei liniare.

NumarCastiguri over Time with Polynomial Regression and Future Predictions



NumarCastiguri over Time with Polynomial Regression



Acest grafic ar trebui să arate cum linia de regresie polinomială se comportă în comparație cu datele viitoare, folosind doar modelul antrenat pe datele prezentate anterior.

Graficul urmator ilustreaza modelul de regresie polinomiala pentru datele de testare.

```

1 import numpy as np
2 import pandas as pd
3 import matplotlib.pyplot as plt
4 from sklearn.preprocessing import PolynomialFeatures
5 from sklearn.linear_model import LinearRegression
6
7 # Definirea datelor
8 x = np.array(pd.to_datetime(df_cat4['DATA']).apply(lambda x: x.toordinal()).reshape(-1, 1))
9 y = np.array(df_cat4['NumarCastiguri_netezite'])
10
11 if len(x) != len(y):
12     raise ValueError("Lengths of x and y are not the same.")
13
14 # Aplicarea regresiei polinomiale
15 poly_features = PolynomialFeatures(degree=2, include_bias=False)
16 x_poly = poly_features.fit_transform(x)
17
18 # Antrenarea modelului cu datele transformate
19 model = LinearRegression()
20 model.fit(x_poly, y)
21
22 # Realizarea predicțiilor pe datele de antrenament
23 y_pred = model.predict(x_poly)
24
25 # Vizualizarea rezultatelor
26 plt.scatter(df_cat4['DATA'], df_cat4['NumarCastiguri_netezite'], Label='Original data')
27 plt.plot(df_cat4['DATA'], y_pred, color='red', linewidth=2, Label='Polynomial Regression (degree=2)')
28
29 plt.xlabel('Date')
30 plt.ylabel('NumarCastiguri')
31 plt.title('NumarCastiguri over Time with Polynomial Regression')
32 plt.legend()
33 plt.show()
34

```

```

import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
from sklearn.preprocessing import PolynomialFeatures, MinMaxScaler
from sklearn.linear_model import LinearRegression

y = np.array(df_cat4['NumarCastiguri_netezite'])

# Creează datele viitoare
future_dates = pd.date_range(start='2019-01-18', end='2022-05-22', freq='2D')
future_dates = future_dates[:-4]
print(len(future_dates), len(y))
# Scalați datele viitoare utilizând MinMaxScaler
scaler = MinMaxScaler()
future_X = scaler.fit_transform(x).reshape(-1, 1)

# Aplică regresia polinomială
poly_features = PolynomialFeatures(degree=2, include_bias=False)
future_X_poly = poly_features.fit_transform(future_X)

# Antrenează modelul cu datele transformate
model.fit(future_X_poly, y)

# Realizează predicții pe datele viitoare
future_pred = model.predict(future_X_poly)

# Afișează rezultatele
plt.scatter(future_dates, future_pred, color='blue', Label='Future Predictions')
plt.xlabel('Date')
plt.ylabel('NumarCastiguri')
plt.title('NumarCastiguri over Time with Polynomial Regression and Future Predictions')
plt.legend()
plt.show()

```