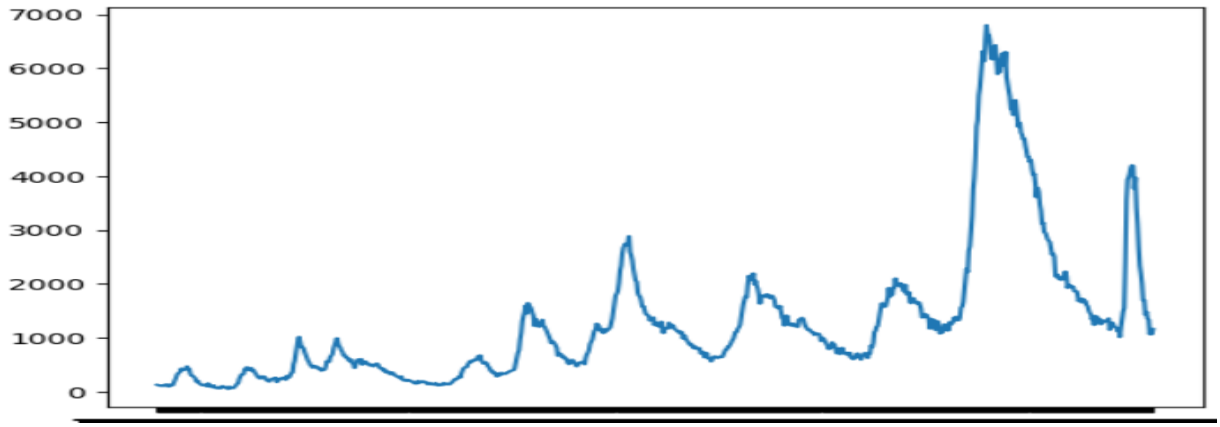
עבודת בית 1 סדרות עתיות:

1).



א) ניתן להבחין במגמה חיובית שנוצרת לאורך השנים, אשר עולה עם השנים.
בנוסף, ניתן להבחין במחזוריות המתרחשת בסדרה, כך שישנו שינוי שחוזר על עצמו כאשר הערכים גדלים וקטנים באופן לא קבוע לאורך השנים.

ב) הסדרה לא סטציונארית, כיוון שנראה שלנתונים יש מגמה חיובית לאורך השנים ואנו יכולים להסיק לאורך הזמן תוחלת הערכים וסטיית התקן שלהן משתנים ומכאן מדובר בסדרה לא סטציונארית.

ג) ניתן להסביר את המאפיינים שזיהינו בסעיף א' באמצעות כך שאבטלה של אנשים הוא אינו מחזורי או עונתי, הוא מצביע על התנאים הכלכליים במדינה אשר עולים או יורדים בעקבות המצב הכלכלי, כאשר הכלכלה במצב ירוד והמשרות מועטות מצב האבטלה יעלה וכאשר הכלכלה צומחת חזרה, מדד האבטלה הקיים ירד, ולכן ניתן להסביר את הקפיצות בנתונים כמעיין משברים כלכליים.

בנוסף, הנתונים מדברים על שיעור המובטלים וכיוון שהאבטלה גדלה בין השנים 1948-2022 זה הגיוני שיש יותר מובטלים.

2).

```
result=adfuller(train_value.dropna())
print("ADF Statistic: ", result[0])
print("p-value: ",result[1])
for key,value in result[4].items():
  print(str(key)+" : "+str(value))
```

```
ADF Statistic:  -2.828444076709055
p-value:  0.05432419819703687
1% : -3.437923659686726
5% : -2.8648832361839442
10% : -2.5685501889710864
```

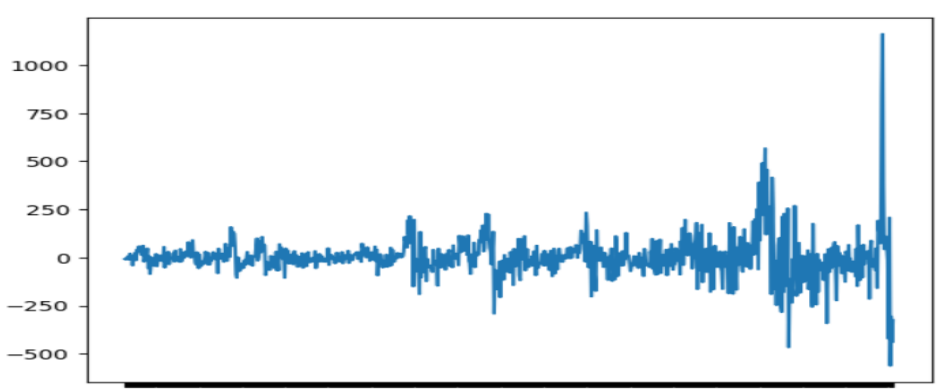
א) לפי תוצאות המבחן הסדרה היא לא סטציונארית כיוון שה p-value משמעותית גדול מאלפא וגם ADF סטטיסטי גדול מהערך הקריטי של אלפא כאשר אלפא 1 ו-5 אחוז, לכן נהפוך אותה לסטציונארית .

ב) נהפוך את הסדרה לסטציונארית:

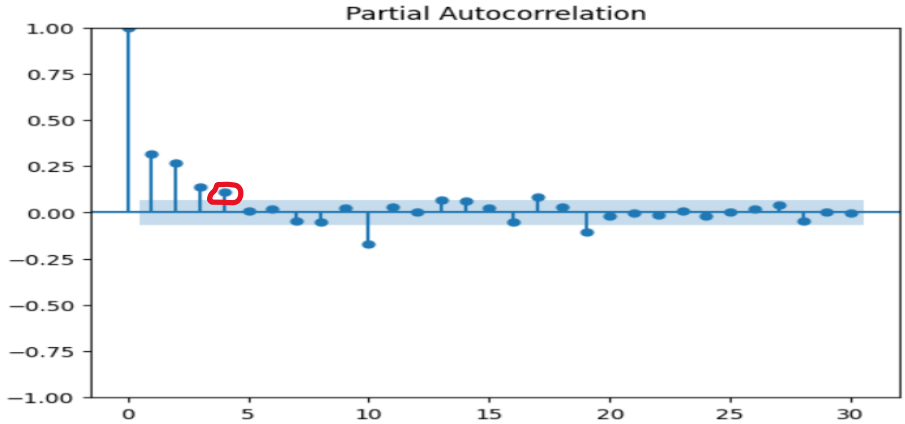
```
result=adfuller(train_value.diff().dropna())
print("ADF Statistic: ", result[0])
print("p-value: ",result[1])
for key,value in result[4].items():
  print(str(key)+" : "+str(value))
```

```
ADF Statistic:  -5.542481149998086
p-value:  1.689423212666125e-06
1% : -3.437923659686726
5% : -2.8648832361839442
10% : -2.5685501889710864
```

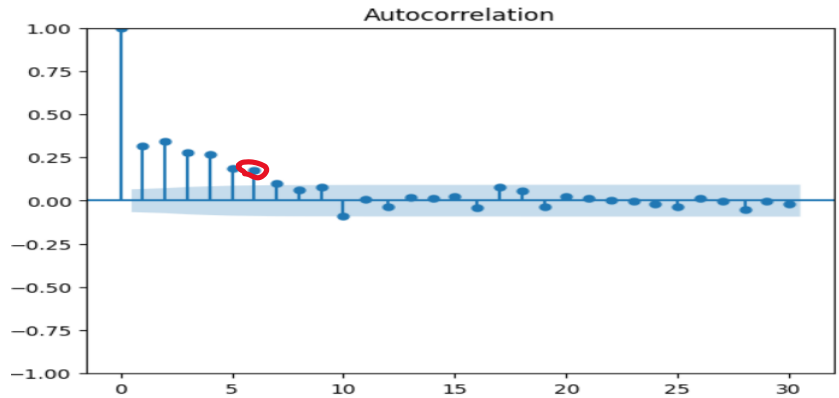
אפשר לראות שאחרי differencing מסדר ראשון הסדרה הפכה לסטציונארית משום שp_value קטן ממש מאלפא וגם ADF סטטיסטי קטן ממש מהערך הקריטי בכל אחד מהאלפות (1,5,10 אחוז).



ג) PACF:



ACF:



כפי שניתן לראות, הגרף ACF הוא מסוג tails-off מכיוון שערכי ה Lag יורדים במתינות לכיוון הטווח הלא מובהק והגרף PACF הוא מסוג tails-off.

p=4 ו- **q=6** - בחרנו ערכים אלו כיוון שכמו שאפשר לראות בגרף ה- ACF אחרי הנקודה השביעית כל שאר הערכים אינם מובהקים חוץ מנקודה השמינית שנראת קצת מובהקת אבל החלטנו להשאיר אותה לא מובהקת ובגרף של PACF האחרון שהוא מובהק ברצף זו הנקודה החמישית ולכן בחרנו אותה. (את הנקודה הראשונה לא מחשיבים).
אם לא יצאו המקדמים מובהקים, נבדוק גם גרסאות יותר מופשטות של p ו-q.

ד) עבור הערכים בערך p=4 ו q=6 -ו d=1 סיכום המודל הינו:

SARIMAX Results

Dep. Variable:	Value	**No. Observations:**	886
Model:	ARIMA(4, 1, 6)	**Log Likelihood**	-5265.227
Date:	Sat, 27 Jan 2024	**AIC**	10552.454
Time:	12:54:54	**BIC**	10605.096
Sample:	0	**HQIC**	10572.580
	- 886		
Covariance Type:	opg		

	coef	std err	z	P>|z|	[0.025	0.975]
ar.L1	0.2930	0.106	2.752	0.006	0.084	0.502
ar.L2	0.4828	0.081	5.943	0.000	0.324	0.642
ar.L3	0.4435	0.053	8.329	0.000	0.339	0.548
ar.L4	-0.6042	0.078	-7.764	0.000	-0.757	-0.452
ma.L1	-0.1203	0.103	-1.168	0.243	-0.322	0.082
ma.L2	-0.3040	0.090	-3.367	0.001	-0.481	-0.127
ma.L3	-0.4022	0.038	-10.554	0.000	-0.477	-0.328
ma.L4	0.6174	0.051	12.000	0.000	0.517	0.718
ma.L5	-0.0145	0.034	-0.433	0.665	-0.080	0.051
ma.L6	0.1282	0.034	3.794	0.000	0.062	0.194
sigma2	8917.9877	179.024	49.815	0.000	8567.107	9268.868

Ljung-Box (L1) (Q):	0.00	**Jarque-Bera (JB):**	15631.57
Prob(Q):	0.99	**Prob(JB):**	0.00
Heteroskedasticity (H):	15.96	**Skew:**	1.64
Prob(H) (two-sided):	0.00	**Kurtosis:**	23.33

ה) ניתן לראות כי חלק מהמקדמים של המודל אינן מובהקים ולכן המודל אינו מובהק סטטיסטית. לכן, נרצה למצוא את מודל שהינו מובהק סטטיסטית, אשר מביא לביצועים הטובים ביותר.

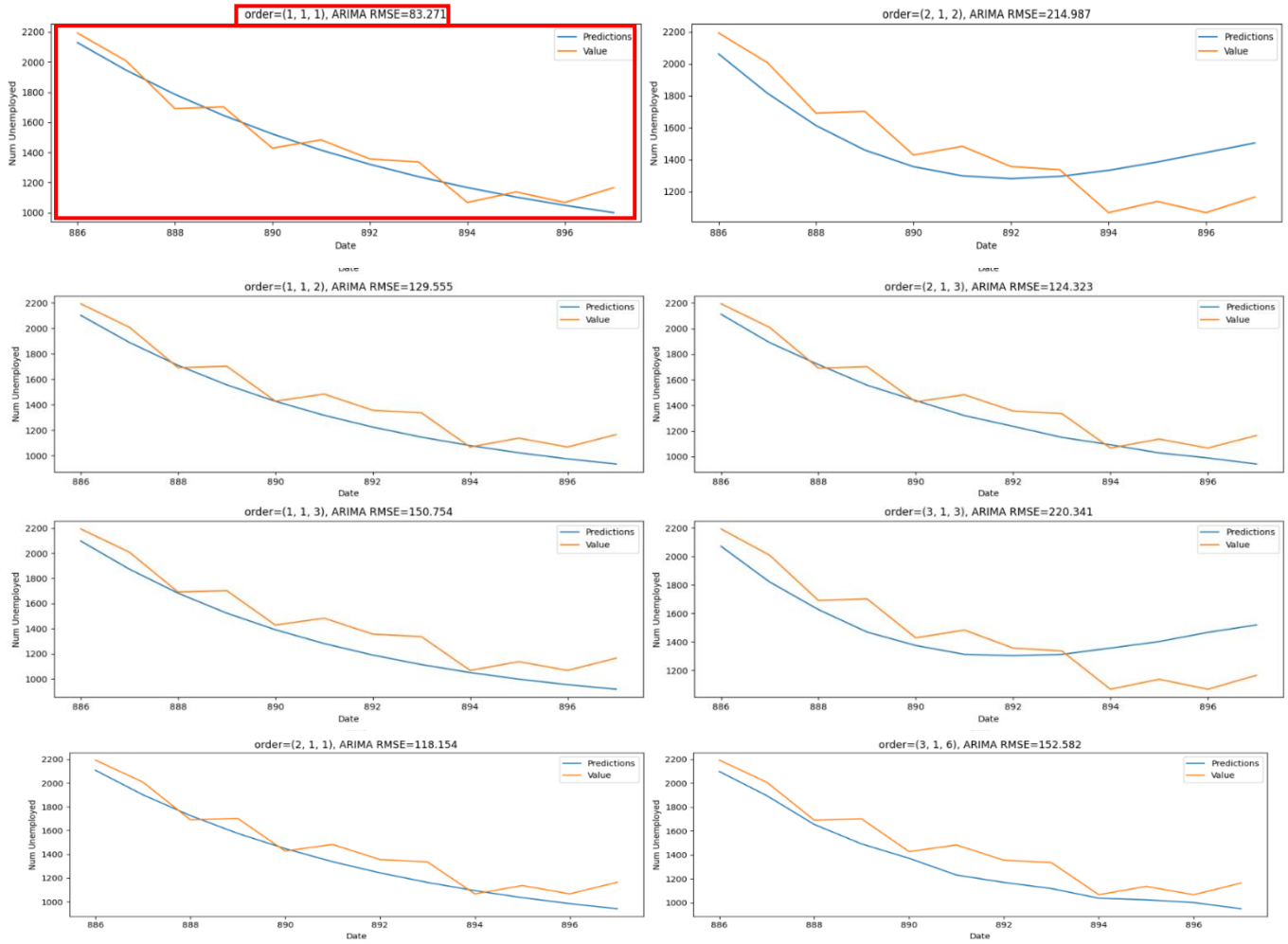
ראשית, לצורך כך, נגדיר סט של פרמטרים בעלי lag מובהק על פי הגרפים ACF ו - PACF מתוך $p\in\{1,2,3,4\}$, $q\in\{1,2,3,4,5,6\}$ ו- d=1.

לכן, יצרנו קוד שבוחן זאת ומחזיר לנו את צמדי p ו- q עבור אותו מודל מובהק (אין p-value שגדול מאלפא=0.05):

```
for p in range(1,5):
    for q in range(1,7):
        if not any(ARIMA(train_value, order=(p,1,q)).fit().pvalues>0.05):
            print(f'p={p}, q={q}')
```

```
p=1, q=1
p=1, q=2
p=1, q=3
p=2, q=1
p=2, q=2
p=2, q=3
p=3, q=3
p=3, q=6
```

כעת, מתוך אותם מודלים מובהקים, נרצה למצוא את המודל אשר מביא לביצועים הטובים ביותר. לשם כך, נבחן את הביצועים של המודלים השונים על פני ה test_set שלנו, באמצעות מדד $RMSE$:

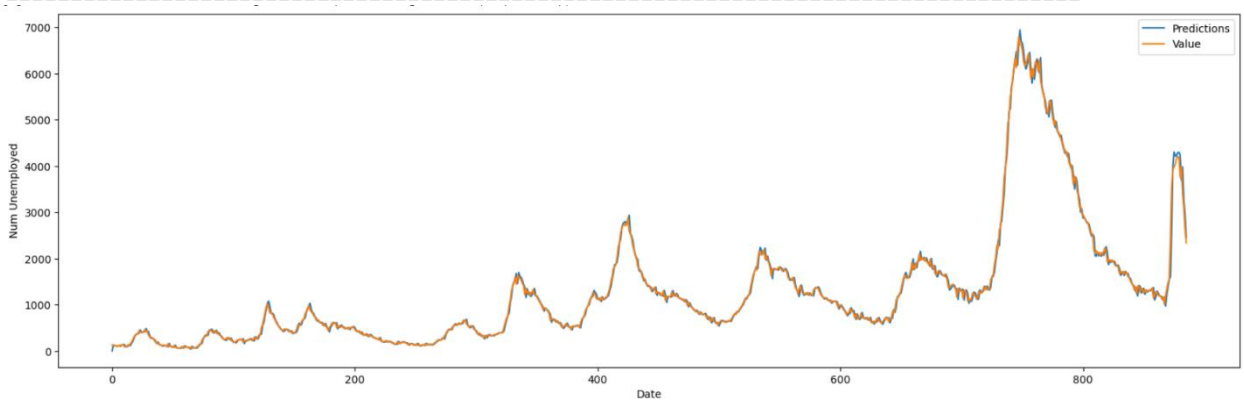


על פי התוצאות שלנו, ניתן לראות כי המודל שהשיג את הביצועים הטובים ביותר על פני ה test_set באמצעות מדד $RMSE$ הינו המודל עם הפרמטרים: p=1,q=1 ולכן נבחר במודל זה כמודל הנבחר.

ו) נציג את ביצועי המודל עם הפרמטרים: p=1,q=1 על פני סדרת הזמן :

```
model111=ARIMA(train_data["Value"],order=(1,1,1))
model_fit111=model111.fit()
print(model_fit111.summary())

predictions_train = model_fit111.predict()

fig, ax = plt.subplots(figsize=(20, 6))
predictions_train.plot(ax=ax, label='Predictions')
train_data.plot(ax=ax)
ax.set_xlabel('Date')
ax.set_ylabel('Num Unemployed')
plt.legend()
plt.show()

rmse = mean_squared_error(train_data["Value"], predictions_train, squared=False)
print('RMSE=%.3f' % (rmse))
```

```
                               SARIMAX Results
==============================================================================
Dep. Variable:                  Value   No. Observations:                  886
Model:                 ARIMA(1, 1, 1)   Log Likelihood               -5287.966
Date:                Sat, 27 Jan 2024   AIC                          10581.932
Time:                        15:13:37   BIC                          10596.289
Sample:                             0   HQIC                         10587.421
                                - 886
Covariance Type:                  opg
==============================================================================
                 coef    std err          z      P>|z|      [0.025      0.975]
------------------------------------------------------------------------------
ar.L1          0.8751      0.016     53.418      0.000       0.843       0.907
ma.L1         -0.6322      0.023    -27.180      0.000      -0.678      -0.587
sigma2      9062.7642    165.199     54.860      0.000    8738.980    9386.549
===================================================================================
Ljung-Box (L1) (Q):                   3.81   Jarque-Bera (JB):             13273.68
Prob(Q):                              0.05   Prob(JB):                         0.00
Heteroskedasticity (H):              16.73   Skew:                             1.17
Prob(H) (two-sided):                  0.00   Kurtosis:                        21.83
===================================================================================
```



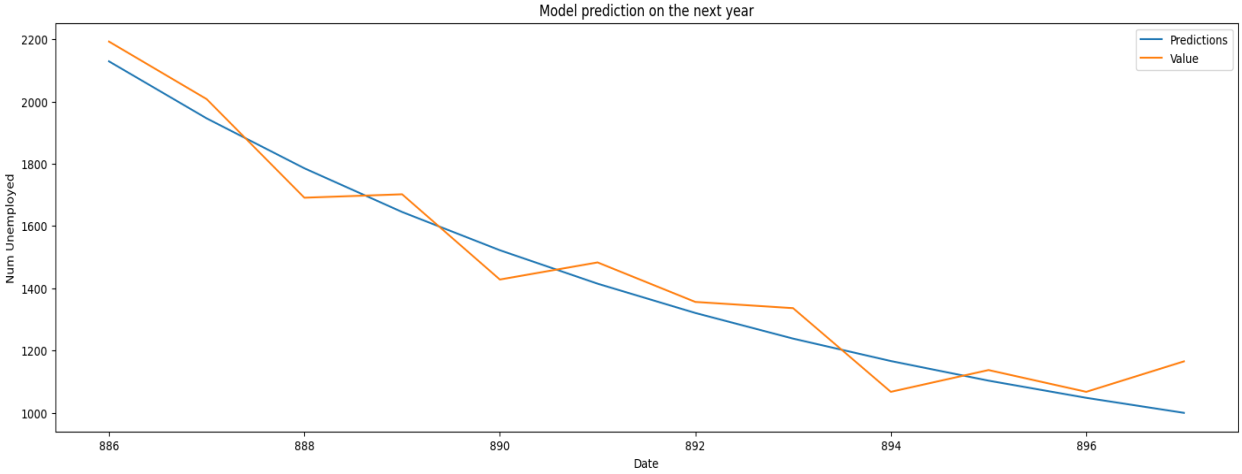

```
RMSE=95.242
```

.3)

א) נציג את תחזית המודל על 12 הערכים הבאים (השנה הבאה):

```
predictions_1y = model_fit111.predict(start=len(train_data["Value"]), end=len(train_data["Value"]) + len(test_data["Value"])-1)
print(predictions_1y)

fig, ax = plt.subplots(figsize=(20, 6))

predictions_1y.plot(ax=ax, label='Predictions')
test_data.plot(ax=ax)

ax.legend()
ax.set_xlabel('Date')
ax.set_ylabel('Num Unemployed')
ax.set_title(f'Model prediction on the next year')
plt.show()

print("--------------------------------------------------")
rmse = mean_squared_error(test_data["Value"], predictions_1y, squared=False)
print('RMSE=%.3f' % (rmse))
```

```
886    2129.451611
887    1946.066443
888    1785.577879
889    1645.127159
890    1522.212450
891    1414.644291
892    1320.506583
893    1238.122465
894    1166.024437
895    1102.928228
896    1047.709914
897     999.385897
```




```
--------------------------------------------------
RMSE=83.271
```

ב) נציג את תחזית חיזוי המתמשך על 12 הערכים הבאים (השנה הבאה):

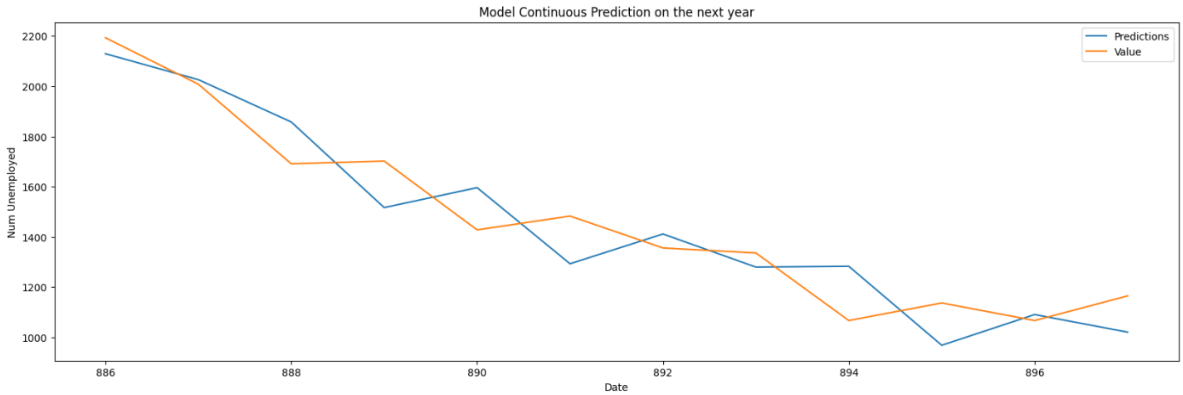
ראשית, נכין את התוצאות של החיזוי המתמשך באמצעות לולאה בה בכל פעם נאמן מודל חדש שיחזה את הערך t באמצעות t-1 הערכים שלפניו :

```
con_predict= pd.Series()
for t in range(len(test_data)):
    model_testing = ARIMA(data["Value"][:-12+t],order=(1,1,1)).fit()
    predict_1next = model_testing.predict(len(train_data["Value"]) + t)
    con_predict = pd.concat([con_predict, pd.Series(predict_1next)])
con_predict
```

```
<ipython-input-52-f9f14a7c0862>:1: FutureWarning: The default dtype for
  predict= pd.Series()
886    2129.451611
887    2026.165157
888    1857.564226
889    1516.698592
890    1596.044681
891    1293.063223
892    1411.599075
893    1279.846679
894    1282.981505
895     968.496460
896    1090.802588
897    1020.825390
```

נציג את תוצאות החיזוי למול המידע הנכון מה test_set:

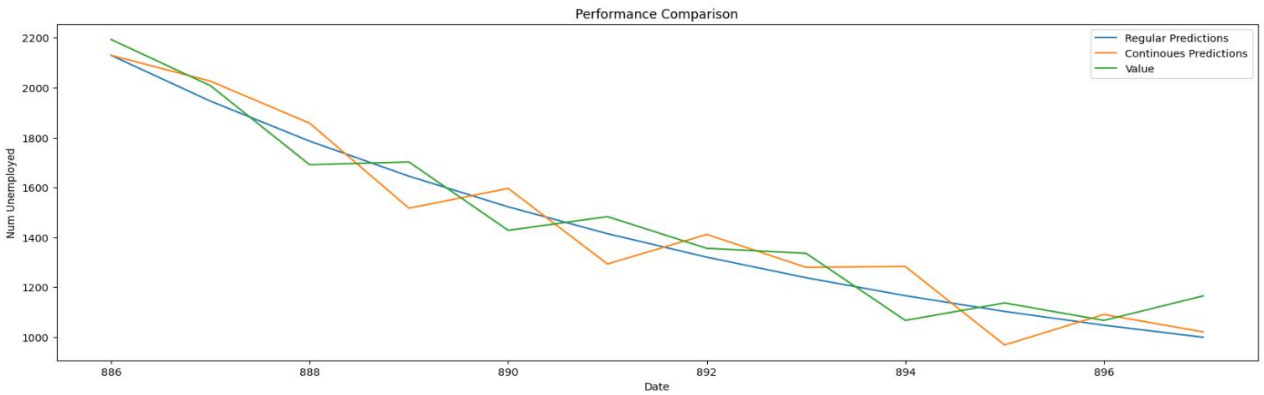
```
fig, ax = plt.subplots(figsize=(20, 6))
con_predict.plot(ax=ax, label='Predictions')
test_data.plot(ax=ax)

ax.legend()
ax.set_xlabel('Date')
ax.set_ylabel('Num Unemployed')
ax.set_title(f'Model Continuous Prediction on the next year')
plt.show()

print("--------------------------------------------------")
rmse = mean_squared_error(test_data["Value"], con_predict, squared=False)
print('RMSE=%.3f' % (rmse))
```




```
--------------------------------------------------
RMSE=139.464
```

4). נציג את ההשוואה של חיזוי המודלים השונים שבנינו בסעיף 3, עם הערכים האמיתיים של סדרת הזמן עבור 12 הרשומות האחרונות:

```
fig, ax = plt.subplots(figsize=(20, 6))
predictions_1y.plot(ax=ax, label='Regular Predictions')
predict.plot(ax=ax, label='Continoues Predictions')
test_data.plot(ax=ax)

ax.legend()
ax.set_xlabel('Date')
ax.set_ylabel('Num Unemployed')
ax.set_title(f'Performance Comparison')
plt.show()
```



5) נעריך את ביצועי המודלים מסעיף 3 באמצעות מדד $RMSE$:

```
rmse_reg_predict = mean_squared_error(test_data["Value"], predictions_1y, squared=False)
rmse_con_predict = mean_squared_error(test_data["Value"], con_predict, squared=False)
print('RMSE Regular Predictions=%.3f' % (rmse_reg_predict))
print('RMSE Continoues Predictions=%.3f' % (rmse_con_predict))
```

```
RMSE Regular Predictions=83.271
RMSE Continoues Predictions=139.464
```