In [99]:

```
import pandas as pd
from sklearn.ensemble import RandomForestRegressor
from sklearn.metrics import mean_squared_error
from sklearn.preprocessing import LabelEncoder
from sklearn.model_selection import train_test_split, cross_val_score, KFold
from sklearn.metrics import mean_absolute_error
import matplotlib.pyplot as plt
```

In [114…

```
#Loading and Preprocessing Data
df=pd.read_csv('C:/Users/Lubna/Downloads/Products_Information.csv')
df['date']=pd.to_datetime(df['date'], format='%Y-%m-%d')

#One-hot encode product_type
lencoder= LabelEncoder()
df['product_type'] = lencoder.fit_transform(df['product_type'])

#Filtering dates
df=df[df['date']>='2016-07-31']
```

In [115…

```
#Lagged Data and Feature Extraction
pastperiod=10 #How many historical dates
for i in range(1,pastperiod+1):
     df['sales_lagged_'+str(i)]=df.groupby(['store_nbr','product_type'])['sales'].shi
     df['special_offer_lagged_'+str(i)]=df.groupby(['store_nbr','product_type'])['spe
features=list(df.keys()[5:])

#Removing incomplete rows created by lagging
df=df.dropna()
```

In [116…

```
#Partioning data to use for training and making predictions
training_data=df[df['date']<'2017-07-31']
predict_data=df['2017-07-31'<=df['date']]


#Train Test Split
X_train, X_test, y_train, y_test = train_test_split(training_data[features], training
```

In [117…

```
#Random Forest
model=RandomForestRegressor(max_depth=8,
                            n_estimators=120)
model.fit(X_train,y_train)
```

Out[117]:

```
▾                RandomForestRegressor
RandomForestRegressor(max_depth=8, n_estimators=120)
```

In [119…

```
#Prediction on train data and test data
train_predict = model.predict(X_train)
test_predict = model.predict(X_test)

#MSE
train_mse = mean_squared_error(train_predict, y_train)
test_mse = mean_squared_error(test_predict, y_test)

#Comparing Train and Test MSE
print('Train MSE = ' + str(train_mse))
print('Test MSE = ' + str(test_mse))

#Cross Validation
kfold=KFold(n_splits=5, shuffle=True, random_state=42)
cv_scores=cross_val_score(model, X_train, y_train, cv=kfold, scoring='neg_mean_square
print('CV: ' + str(cv_scores))
```

```
Train MSE = 84952.27431662659
Test MSE = 92424.50451267313
CV: [-100794.72193476  -97932.49574263  -99365.08523815  -99860.87687873
 -197576.2818968 ]
```

In [118…

```
#Making sales predictions
y_pred=model.predict(predict_data[features])
y_true=predict_data['sales']

#Performance Evaluation
mse=mean_squared_error(y_pred, y_true)
print('Prediction MSE = ' + str(mse))

mae=mean_absolute_error(y_true, y_pred)
print('Prediction MAE = ' + str(mae))
```

```
Prediction MSE = 72281.61823119331
Prediction MAE = 82.44240776271332
```

In [120…

```
#Changing Products back to categorical data
predict_data['product_type'] = lencoder.inverse_transform(predict_data['product_type'


#Inserting Predicted Sales into Dataframe
predict_data.insert(5,'predicted_sales', y_pred)
predict_data
```

```
C:\Users\Lubna\AppData\Local\Temp\ipykernel_10228\3349650723.py:2: SettingWithCopyWar
ning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/us
er_guide/indexing.html#returning-a-view-versus-a-copy
  predict_data['product_type'] = lencoder.inverse_transform(predict_data['product_typ
e'])
```

Out[120]:

	id	date	store_nbr	product_type	sales	predicted_sales	special_offer	sales_la
2972376	2972376	2017-07-31	1	AUTOMOTIVE	8.000	2.839252	0	
2972377	2972377	2017-07-31	1	BABY CARE	0.000	2.839252	0	
2972378	2972378	2017-07-31	1	BEAUTY	3.000	2.839252	0	
2972379	2972379	2017-07-31	1	BEVERAGES	2414.000	1994.079313	24	1
2972380	2972380	2017-07-31	1	BOOKS	1.000	2.839252	0	
...	...	...	...	...	...	...	...	
3000883	3000883	2017-08-15	9	POULTRY	438.133	337.857744	0	2
3000884	3000884	2017-08-15	9	PREPARED FOODS	154.553	116.397400	1	
3000885	3000885	2017-08-15	9	PRODUCE	2419.729	2076.840641	148	13
3000886	3000886	2017-08-15	9	SCHOOL AND OFFICE SUPPLIES	121.000	166.827271	8	1
3000887	3000887	2017-08-15	9	SEAFOOD	16.000	17.834798	0	

28512 rows × 27 columns

In [121…

```
predict_data
```

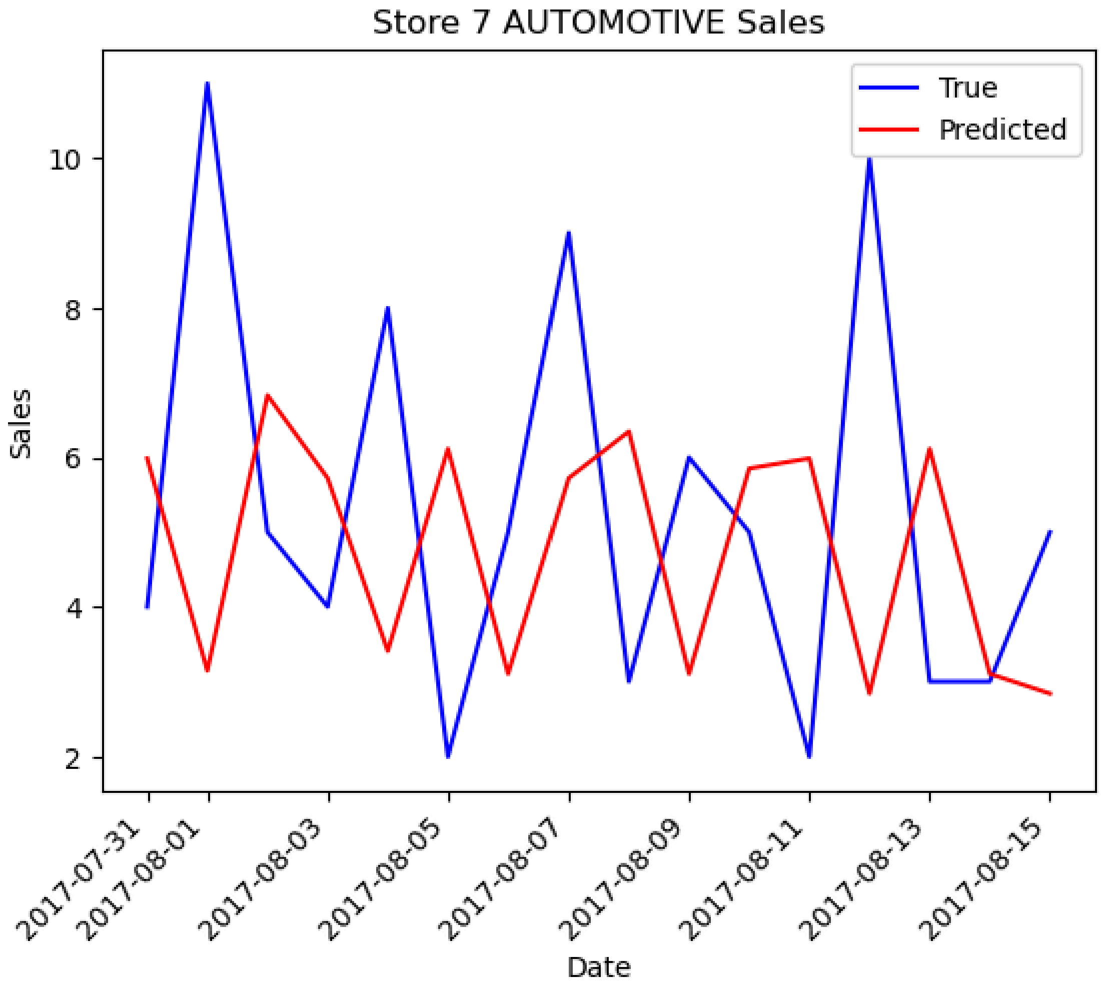
Out[121]:

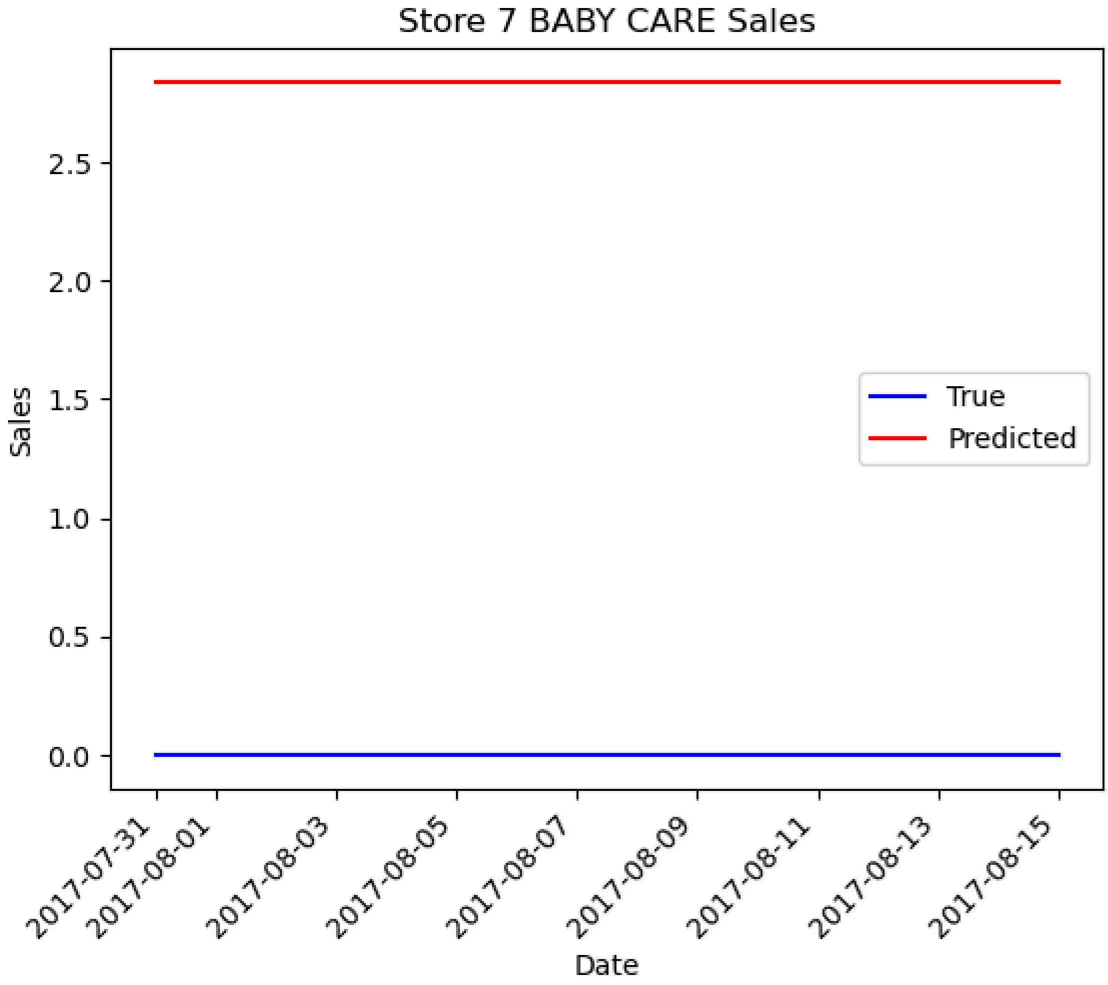
	id	date	store_nbr	product_type	sales	predicted_sales	special_offer	sales_la
2972376	2972376	2017-07-31	1	AUTOMOTIVE	8.000	2.839252	0	
2972377	2972377	2017-07-31	1	BABY CARE	0.000	2.839252	0	
2972378	2972378	2017-07-31	1	BEAUTY	3.000	2.839252	0	
2972379	2972379	2017-07-31	1	BEVERAGES	2414.000	1994.079313	24	1
2972380	2972380	2017-07-31	1	BOOKS	1.000	2.839252	0	
...	...	...	...	...	...	...	...	
3000883	3000883	2017-08-15	9	POULTRY	438.133	337.857744	0	2
3000884	3000884	2017-08-15	9	PREPARED FOODS	154.553	116.397400	1	
3000885	3000885	2017-08-15	9	PRODUCE	2419.729	2076.840641	148	13
3000886	3000886	2017-08-15	9	SCHOOL AND OFFICE SUPPLIES	121.000	166.827271	8	1
3000887	3000887	2017-08-15	9	SEAFOOD	16.000	17.834798	0	

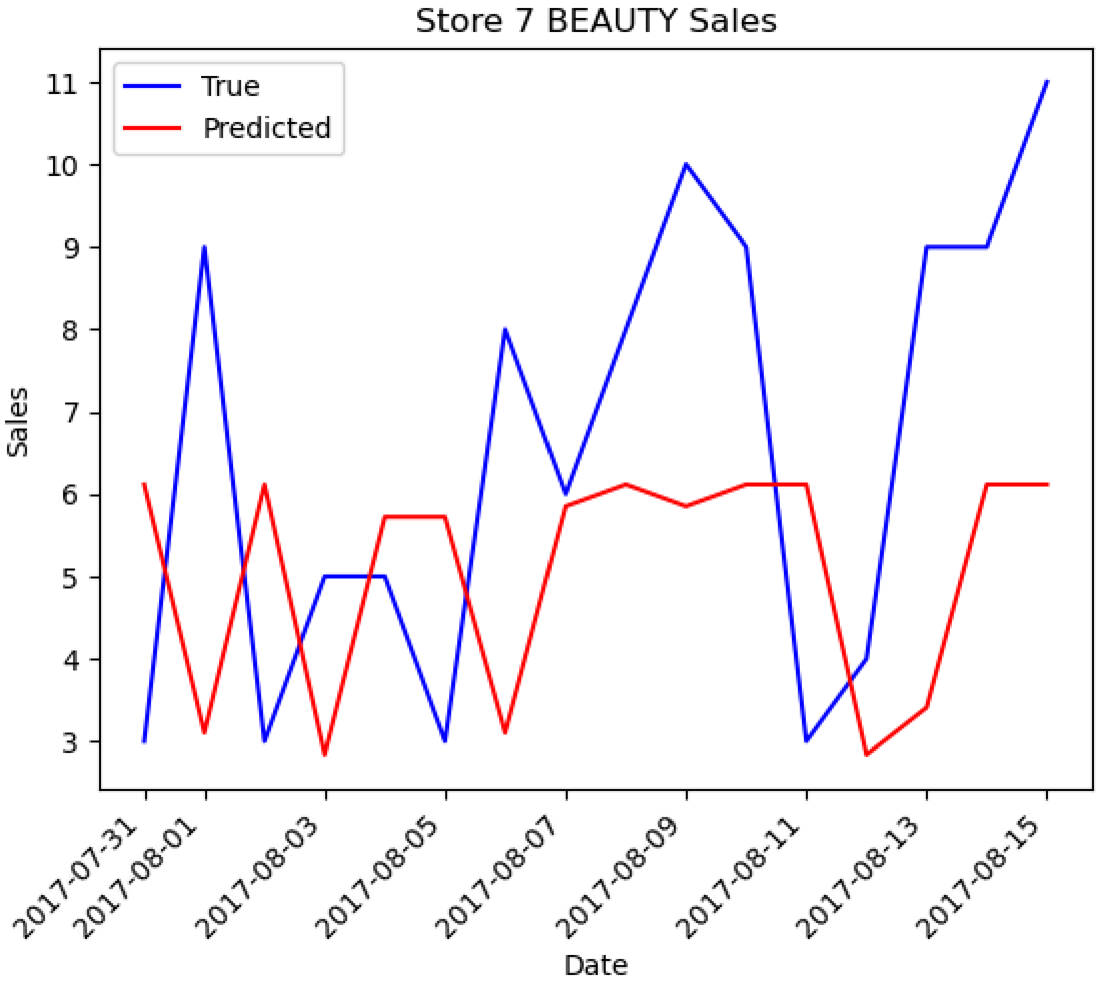
28512 rows × 27 columns

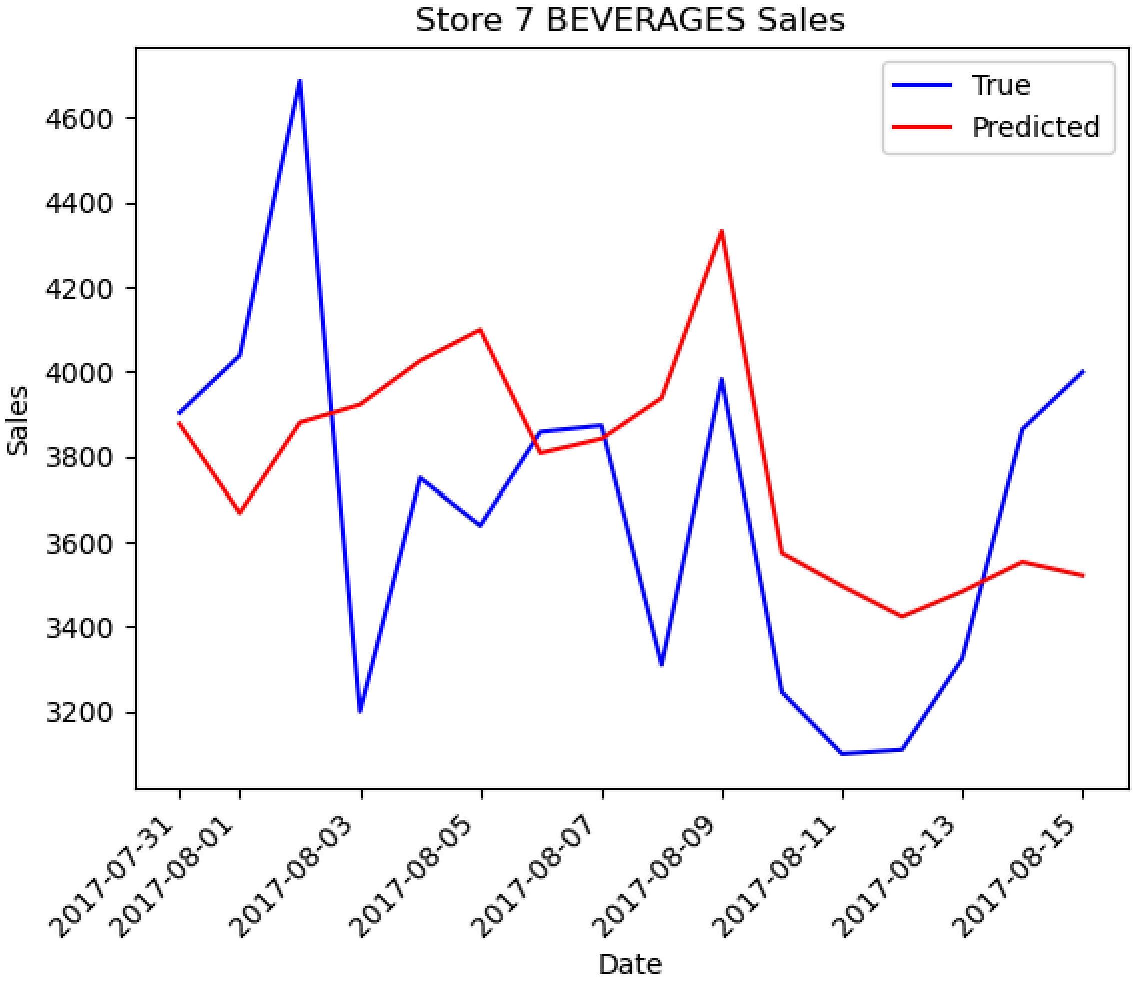
In [122…

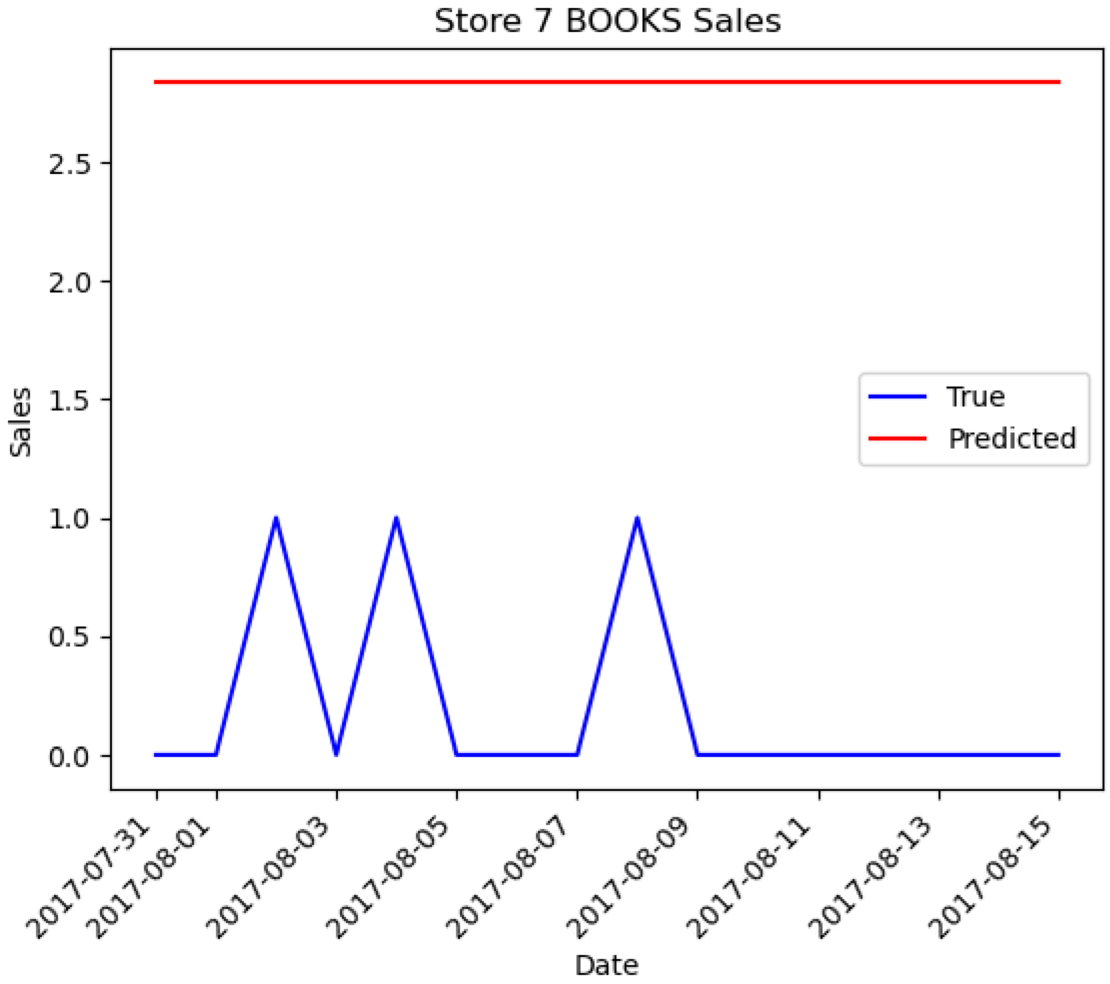
```
#Store 7 Data
store7=predict_data = predict_data[predict_data['store_nbr']==7]

#Creating Plots for Store 7
for product_type, group in store7.groupby('product_type'):
  plt.plot(group['date'], group['sales'],label='True' , color='blue')
  plt.plot(group['date'], group['predicted_sales'],label='Predicted' , color='red')
  plt.legend()
  plt.xlabel('Date')
  plt.ylabel('Sales')
  plt.title('Store 7 '+str(product_type) + ' Sales')
  plt.xticks(rotation=45, ha='right')
  plt.show()
```

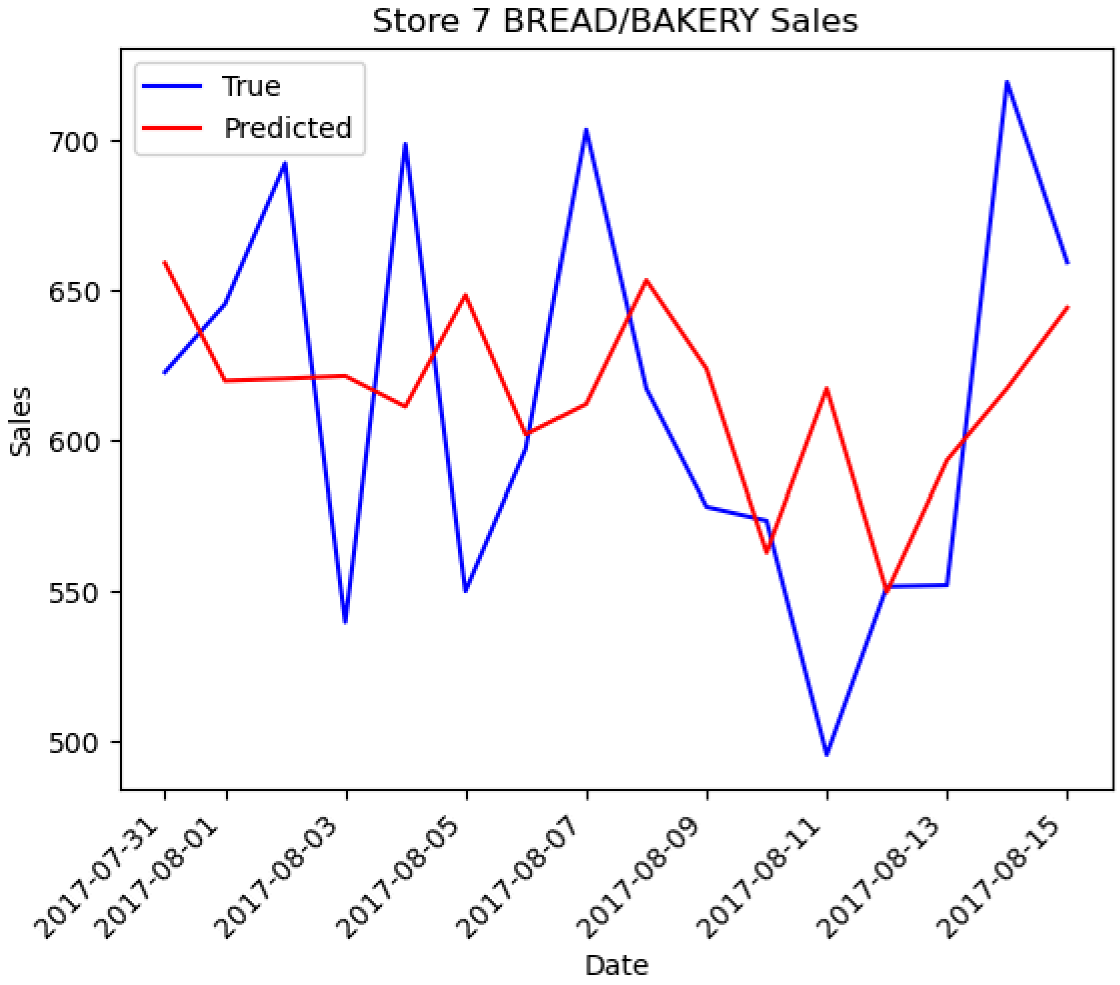

Store 7 AUTOMOTIVE Sales
True
Predicted
Sales
10
8
6
4
2
2017-07-31
2017-08-01
2017-08-03
2017-08-05
2017-08-07
2017-08-09
2017-08-11
2017-08-13
2017-08-15
Date

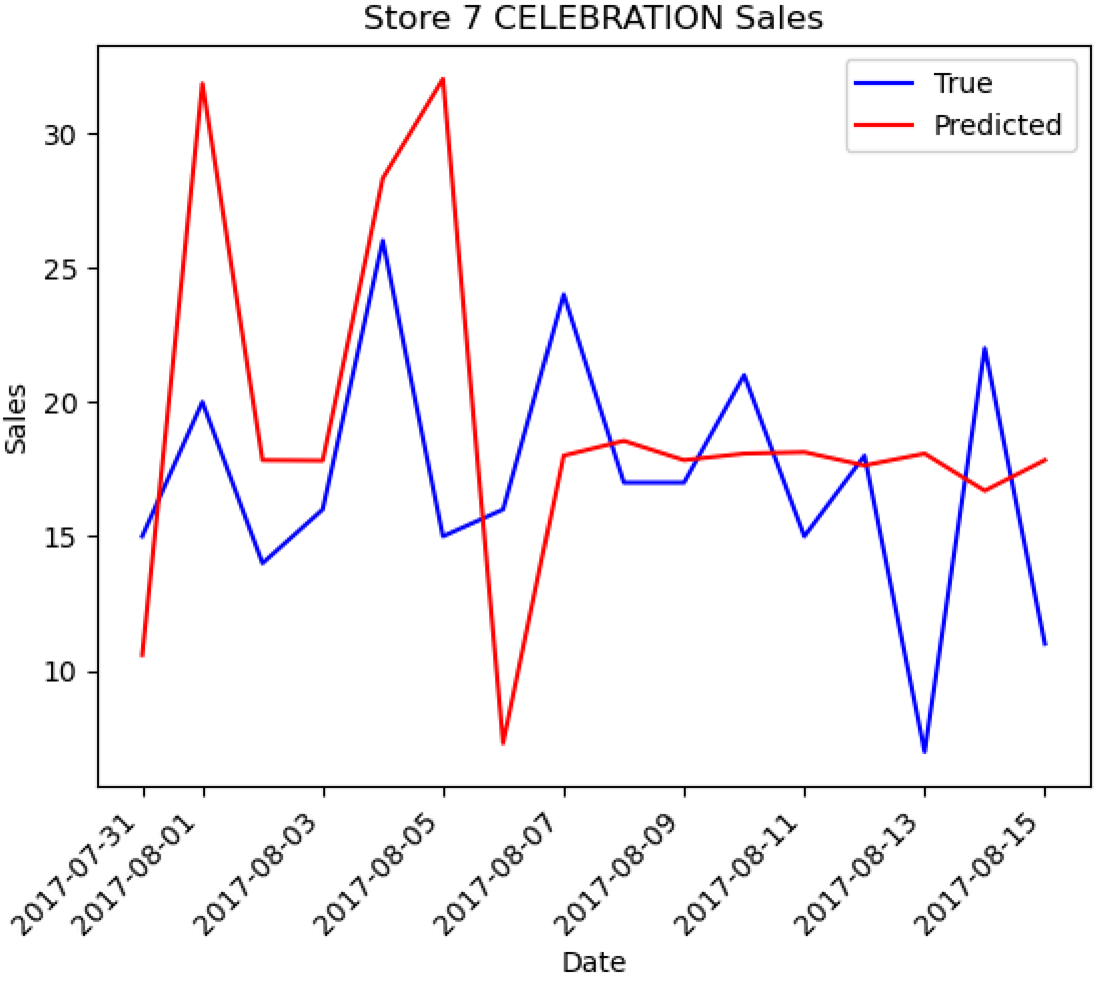

Store 7 BABY CARE Sales
True
Predicted
Sales
2.5
2.0
1.5
1.0
0.5
0.0
2017-07-31
2017-08-01
2017-08-03
2017-08-05
2017-08-07
2017-08-09
2017-08-11
2017-08-13
2017-08-15
Date

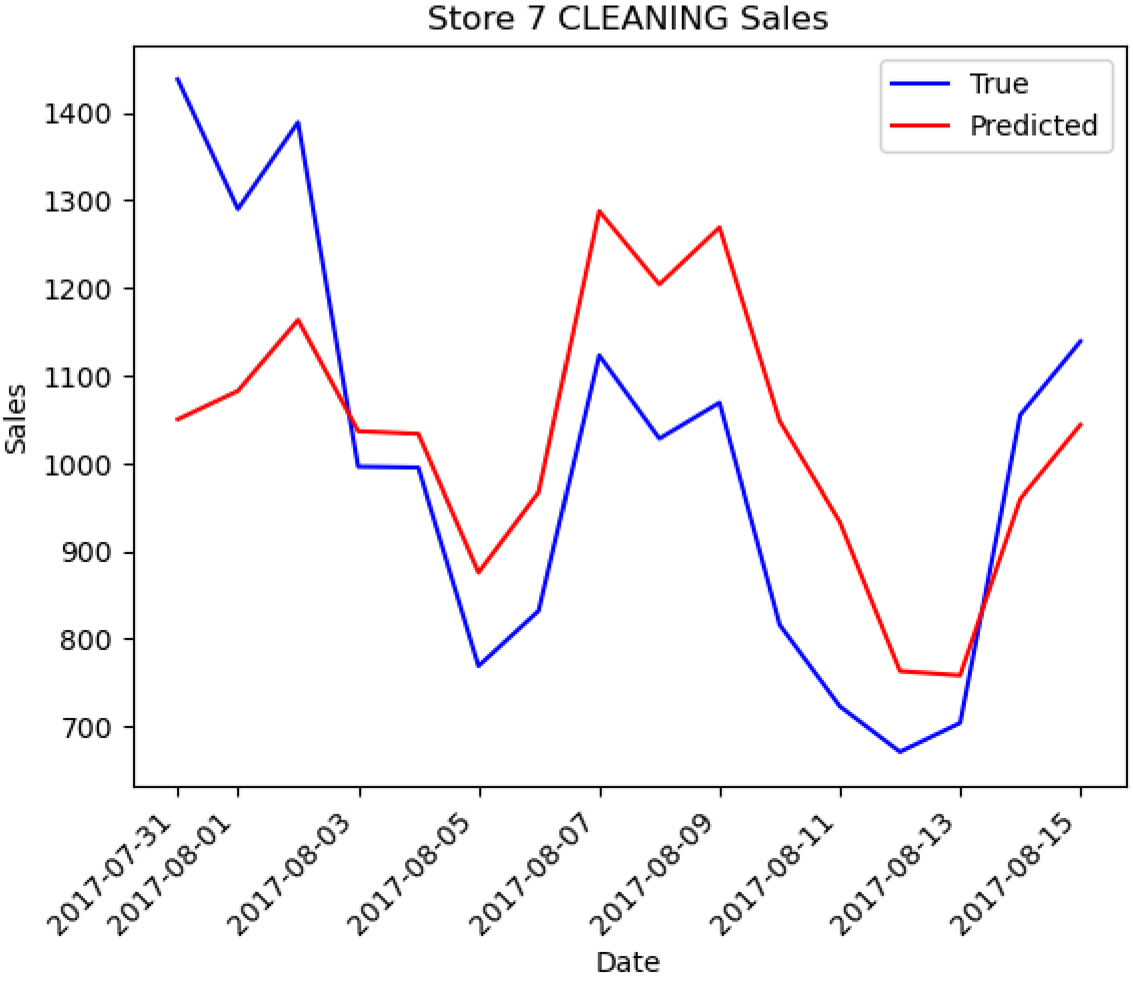

Store 7 BEAUTY Sales
True
Predicted
Sales
3
4
5
6
7
8
9
10
11
Date
2017-07-31
2017-08-01
2017-08-03
2017-08-05
2017-08-07
2017-08-09
2017-08-11
2017-08-13
2017-08-15

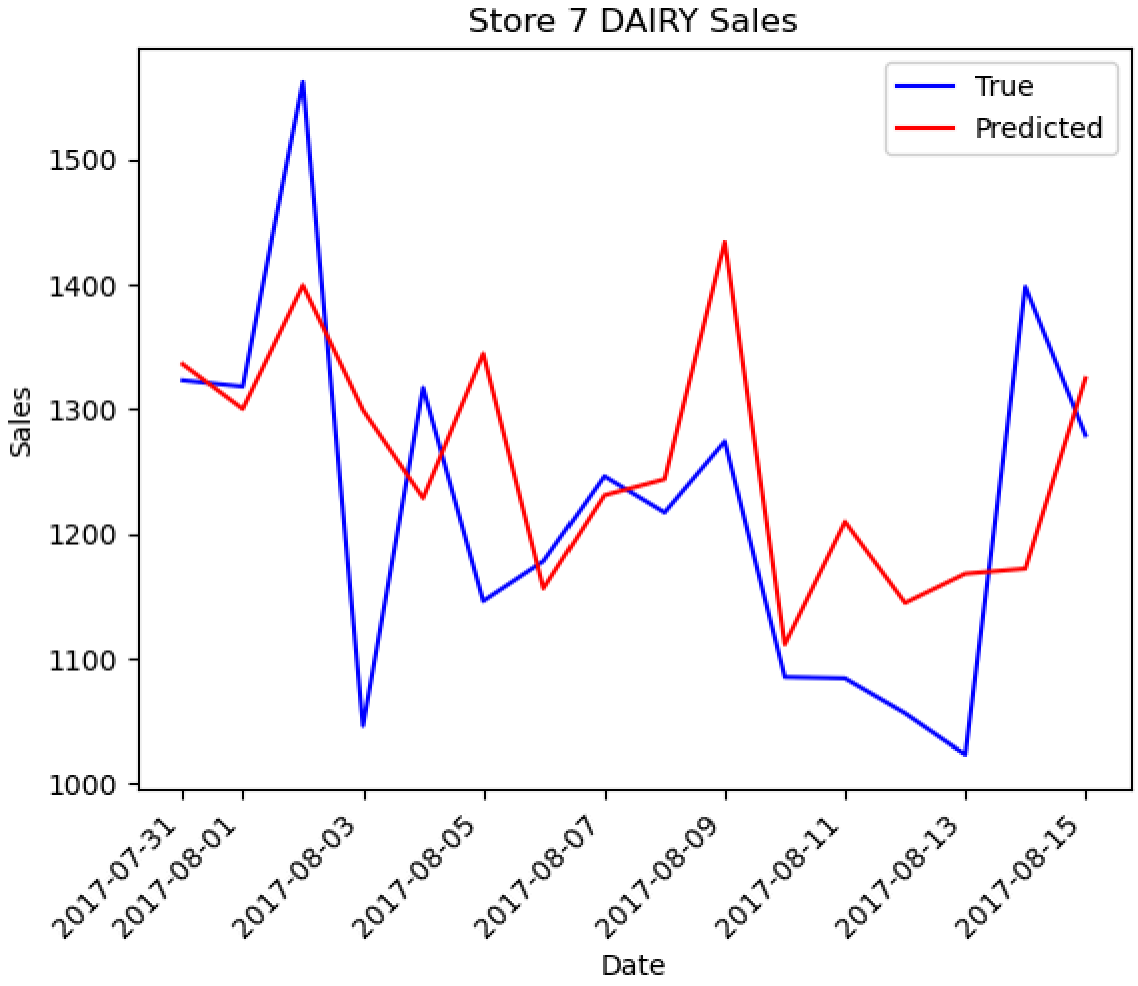

Store 7 BEVERAGES Sales
True
Predicted
Sales
3200
3400
3600
3800
4000
4200
4400
4600
Date
2017-07-31
2017-08-01
2017-08-03
2017-08-05
2017-08-07
2017-08-09
2017-08-11
2017-08-13
2017-08-15

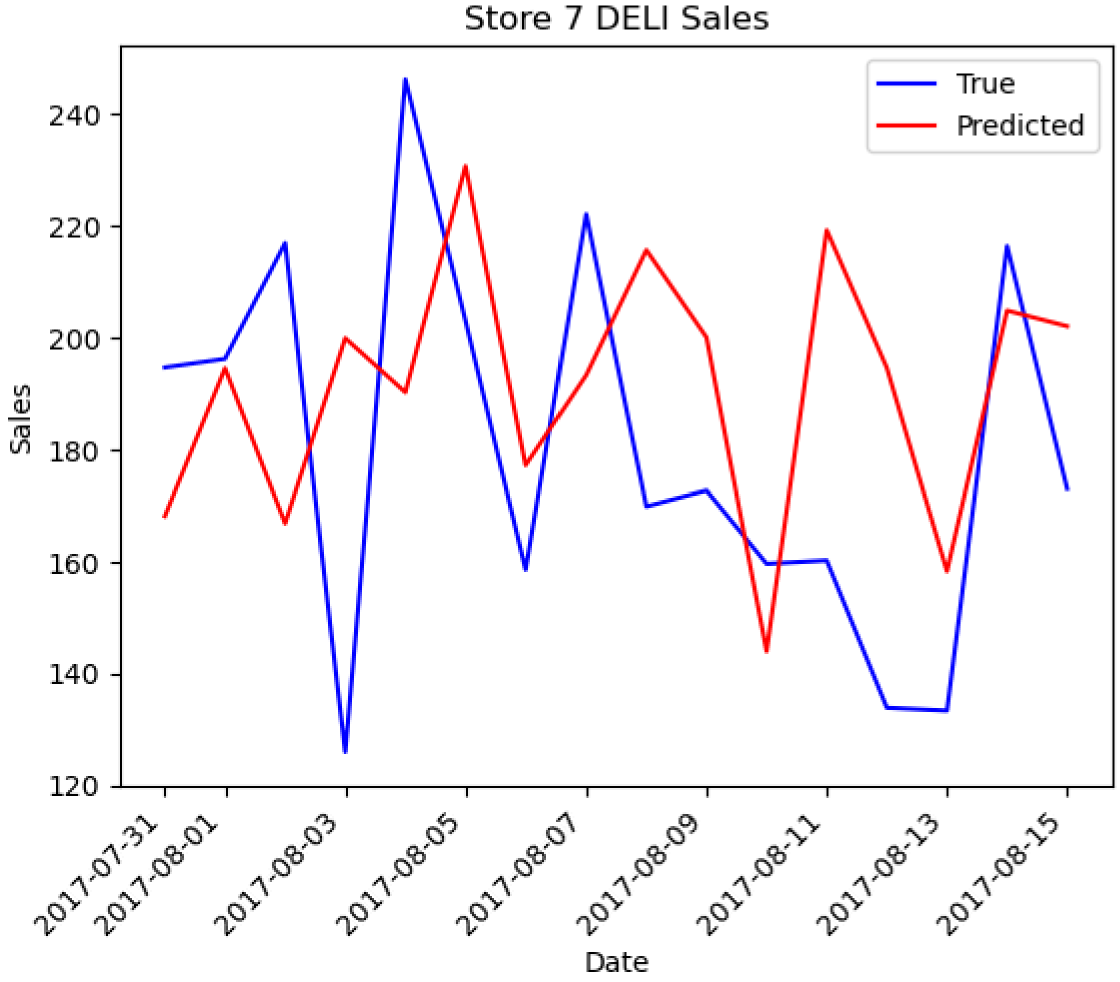

Store 7 BOOKS Sales
Sales
Date
0.0
0.5
1.0
1.5
2.0
2.5
2017-07-31
2017-08-01
2017-08-03
2017-08-05
2017-08-07
2017-08-09
2017-08-11
2017-08-13
2017-08-15
True
Predicted

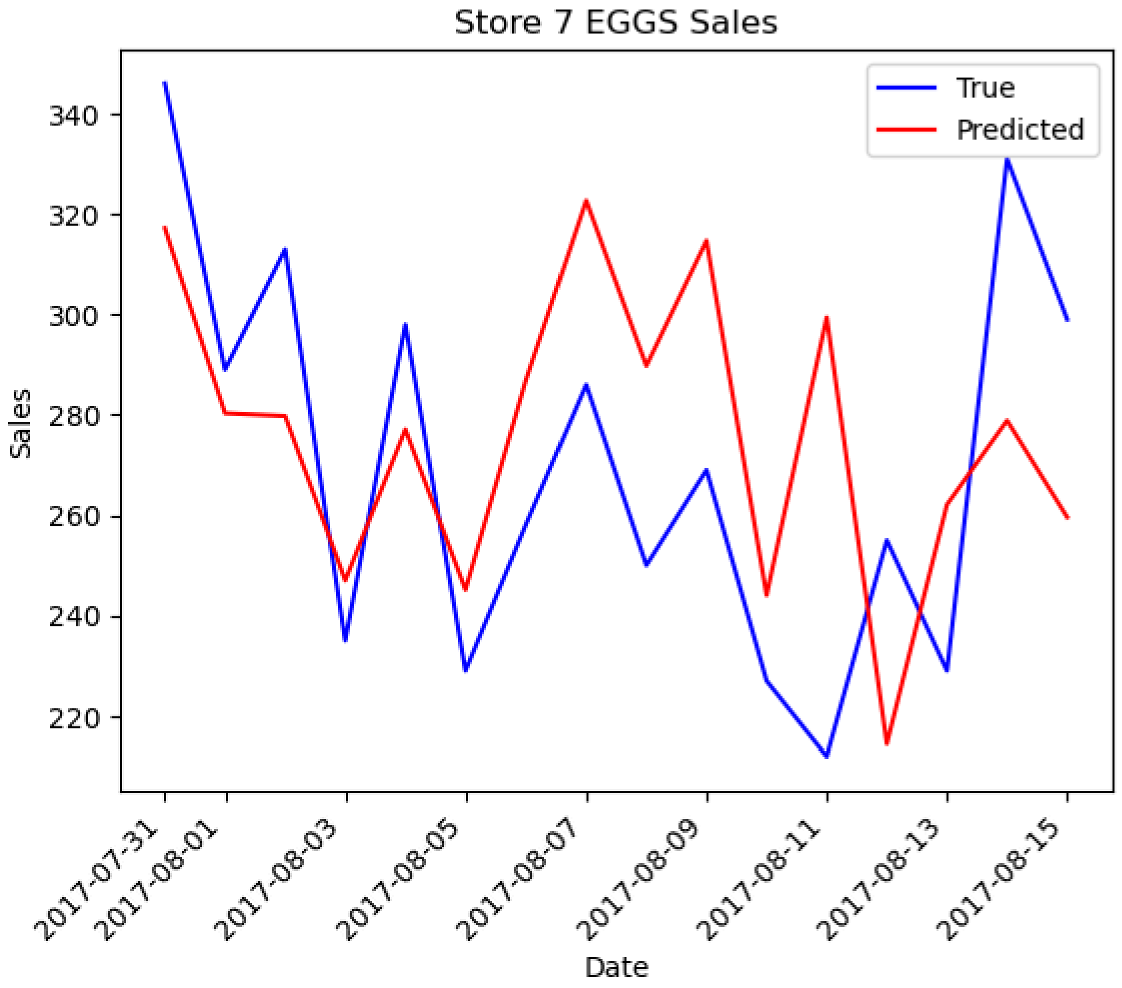

Store 7 BREAD/BAKERY Sales
Sales
Date
500
550
600
650
700
2017-07-31
2017-08-01
2017-08-03
2017-08-05
2017-08-07
2017-08-09
2017-08-11
2017-08-13
2017-08-15
True
Predicted

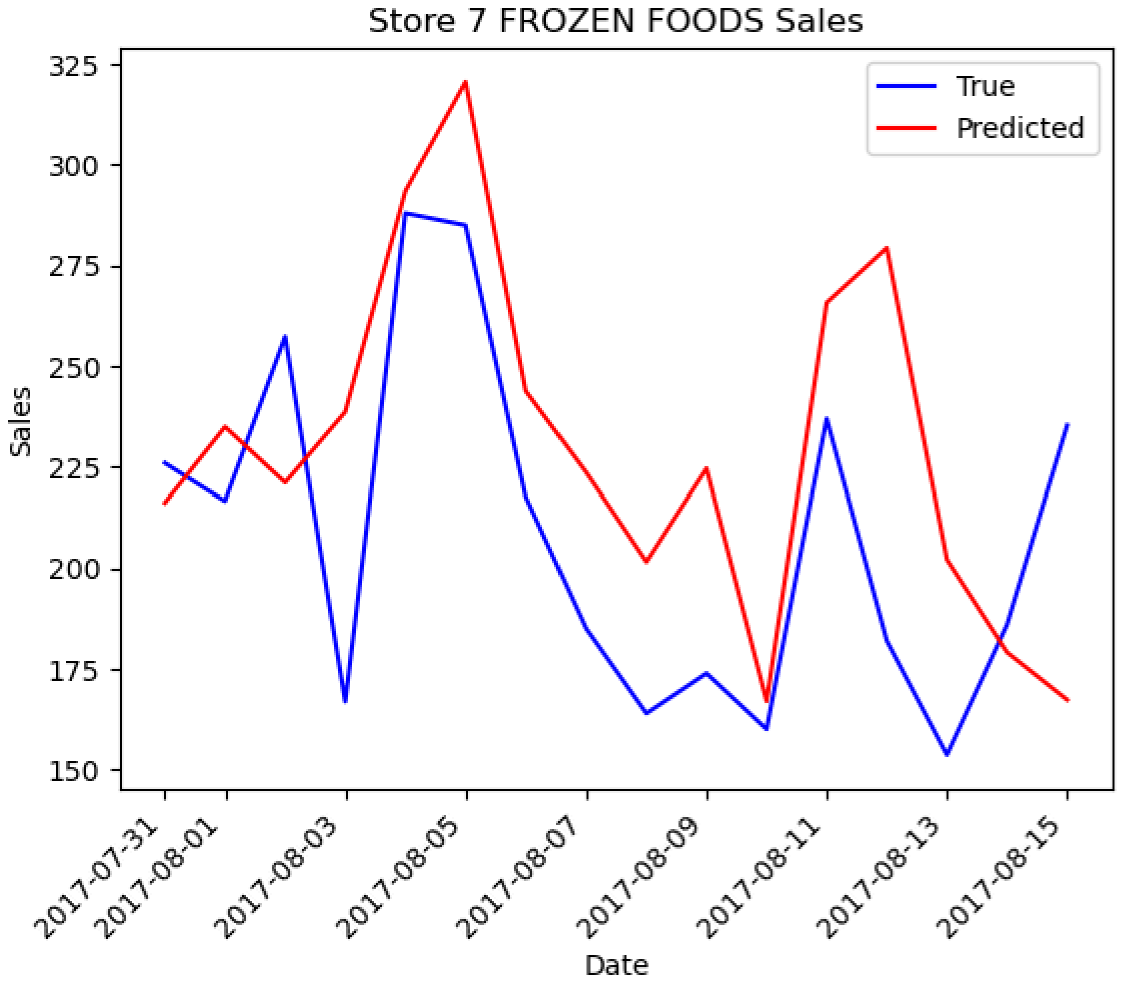

Store 7 CELEBRATION Sales
True
Predicted
Sales
30
25
20
15
10
Date
2017-07-31
2017-08-01
2017-08-03
2017-08-05
2017-08-07
2017-08-09
2017-08-11
2017-08-13
2017-08-15

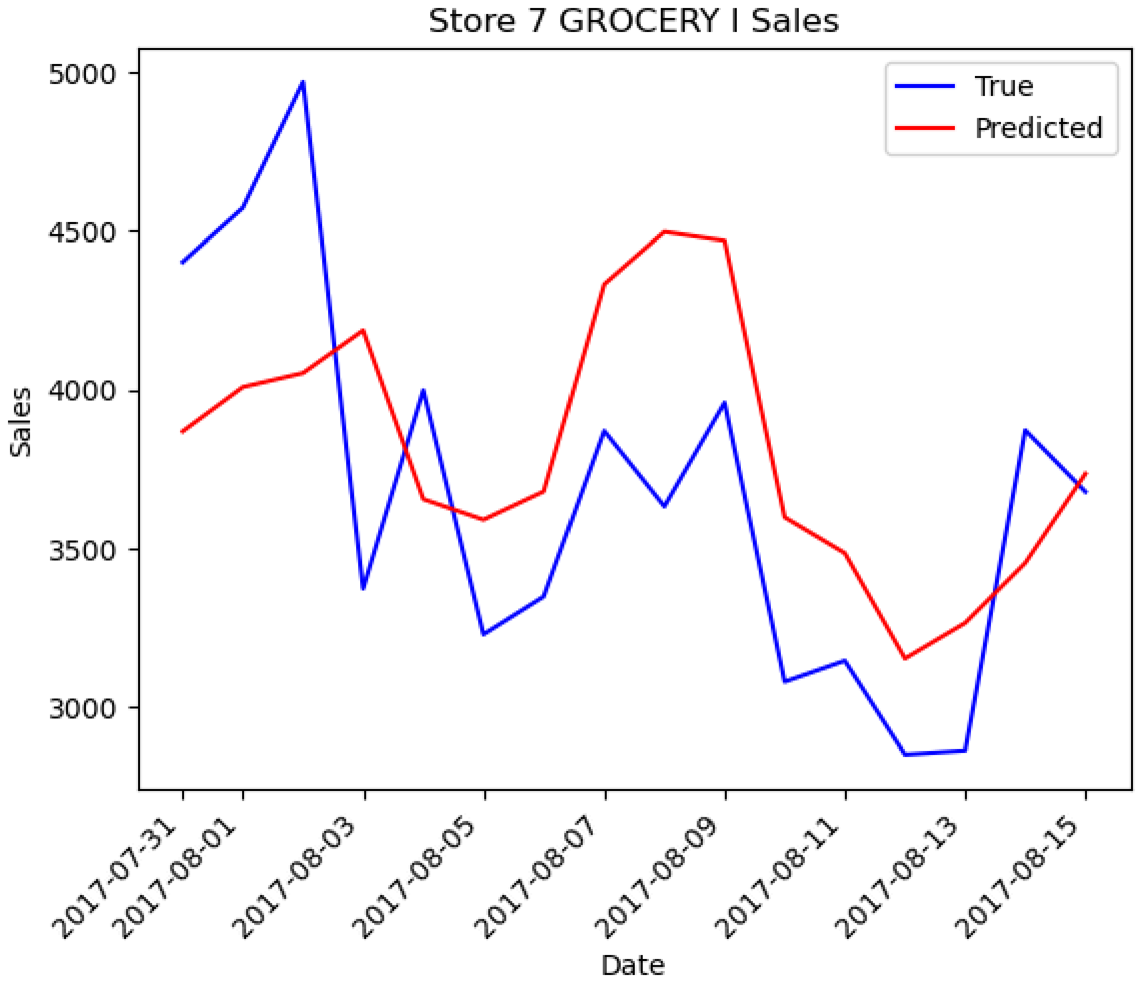

Store 7 CLEANING Sales
True
Predicted
Sales
1400
1300
1200
1100
1000
900
800
700
Date
2017-07-31
2017-08-01
2017-08-03
2017-08-05
2017-08-07
2017-08-09
2017-08-11
2017-08-13
2017-08-15

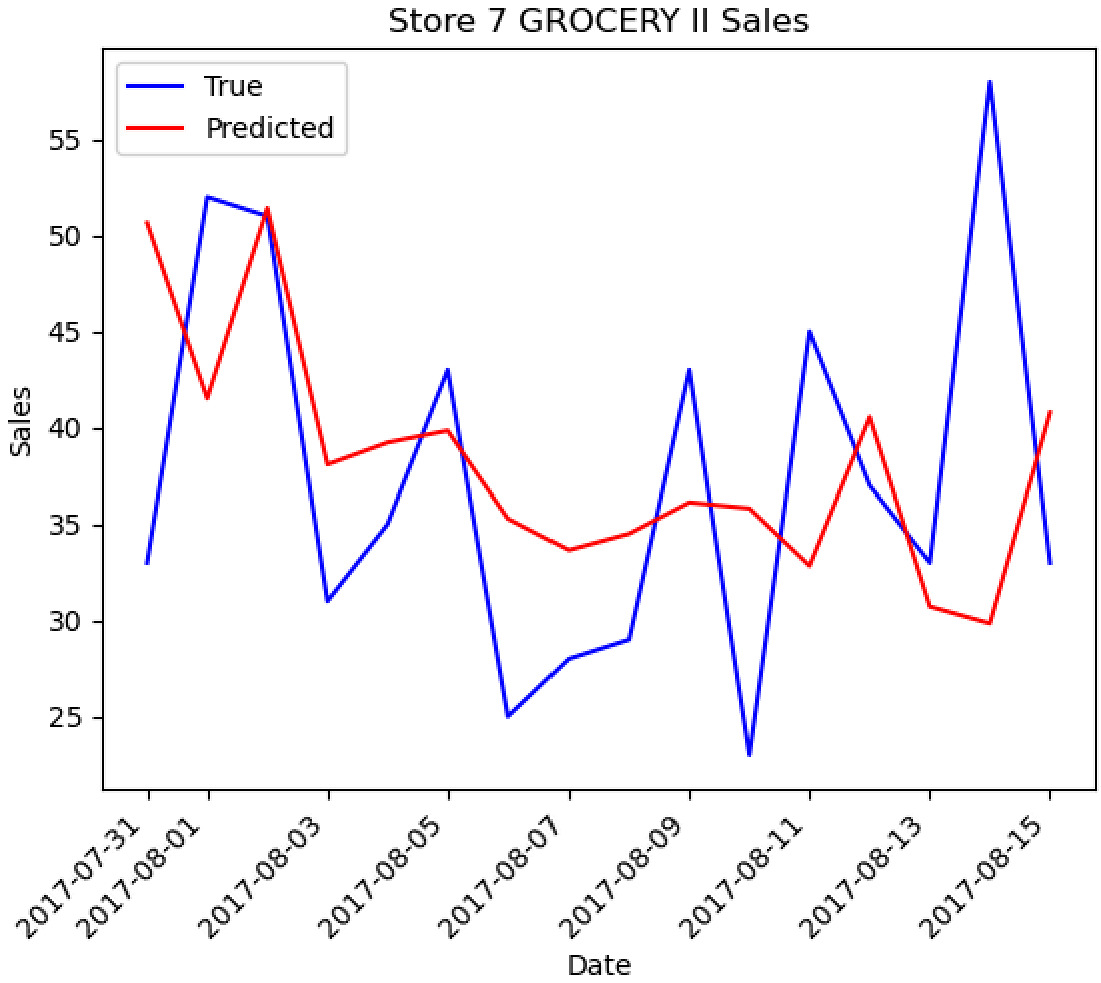

Store 7 DAIRY Sales
True
Predicted
Sales
1000
1100
1200
1300
1400
1500
Date
2017-07-31
2017-08-01
2017-08-03
2017-08-05
2017-08-07
2017-08-09
2017-08-11
2017-08-13
2017-08-15

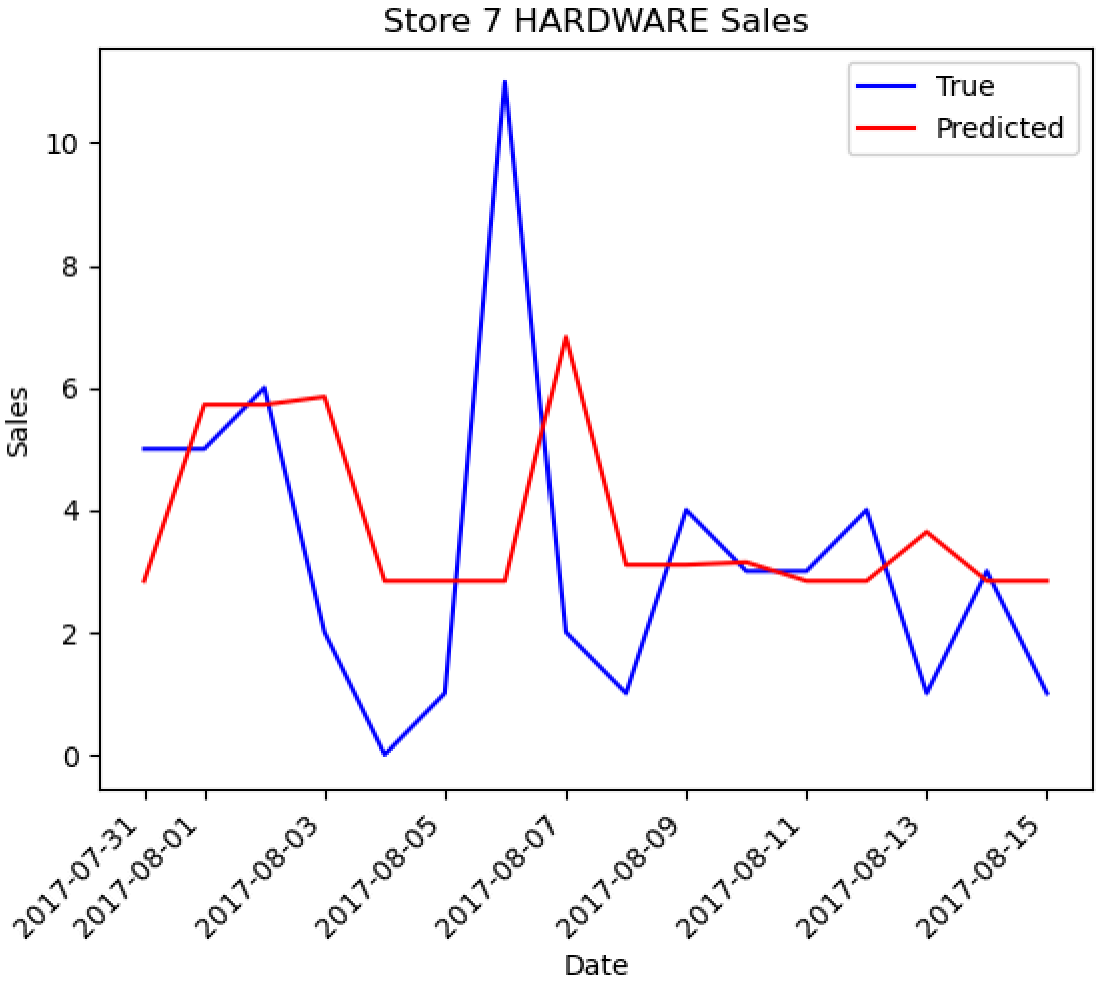

Store 7 DELI Sales
True
Predicted
Sales
120
140
160
180
200
220
240
Date
2017-07-31
2017-08-01
2017-08-03
2017-08-05
2017-08-07
2017-08-09
2017-08-11
2017-08-13
2017-08-15

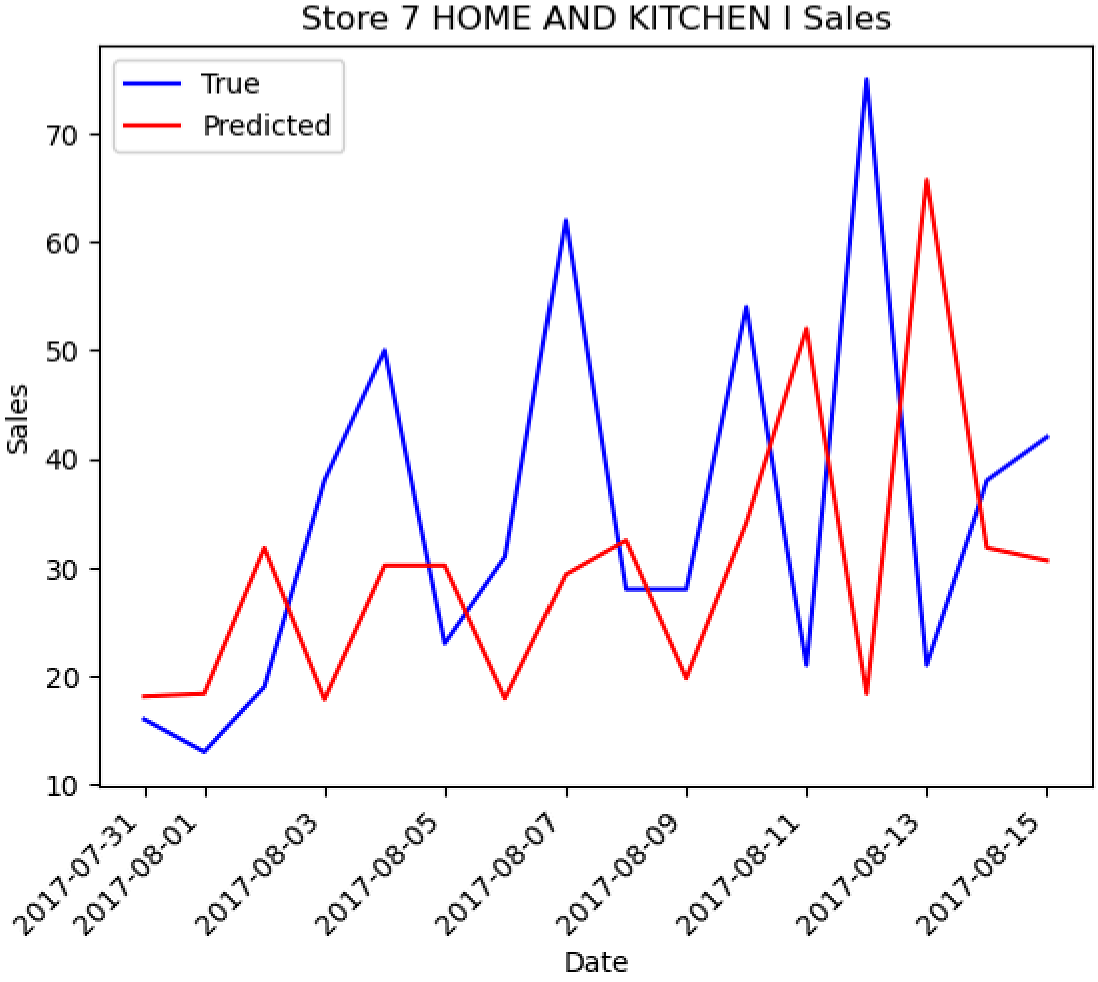

Store 7 EGGS Sales
True
Predicted
Sales
340
320
300
280
260
240
220
Date
2017-07-31
2017-08-01
2017-08-03
2017-08-05
2017-08-07
2017-08-09
2017-08-11
2017-08-13
2017-08-15

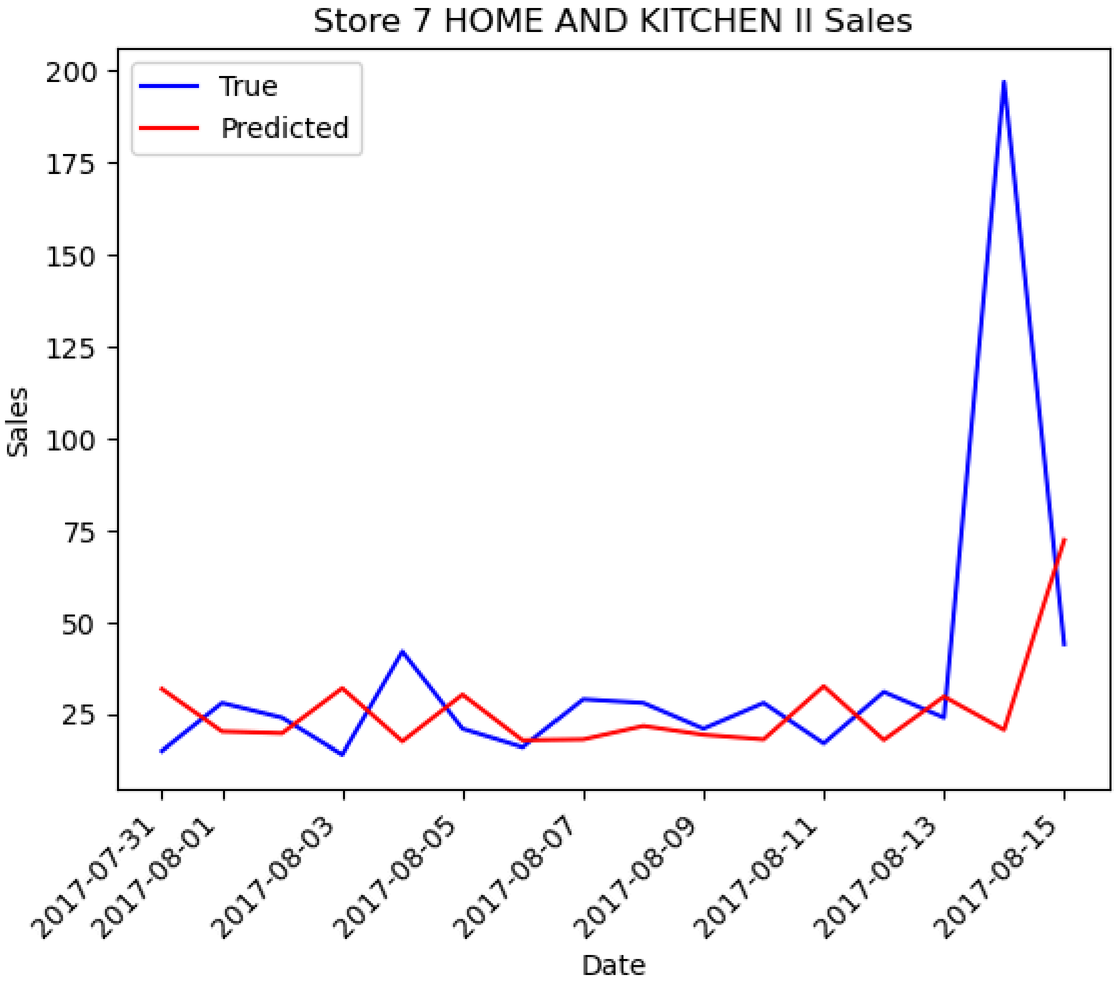

Store 7 FROZEN FOODS Sales
True
Predicted
Sales
325
300
275
250
225
200
175
150
Date
2017-07-31
2017-08-01
2017-08-03
2017-08-05
2017-08-07
2017-08-09
2017-08-11
2017-08-13
2017-08-15

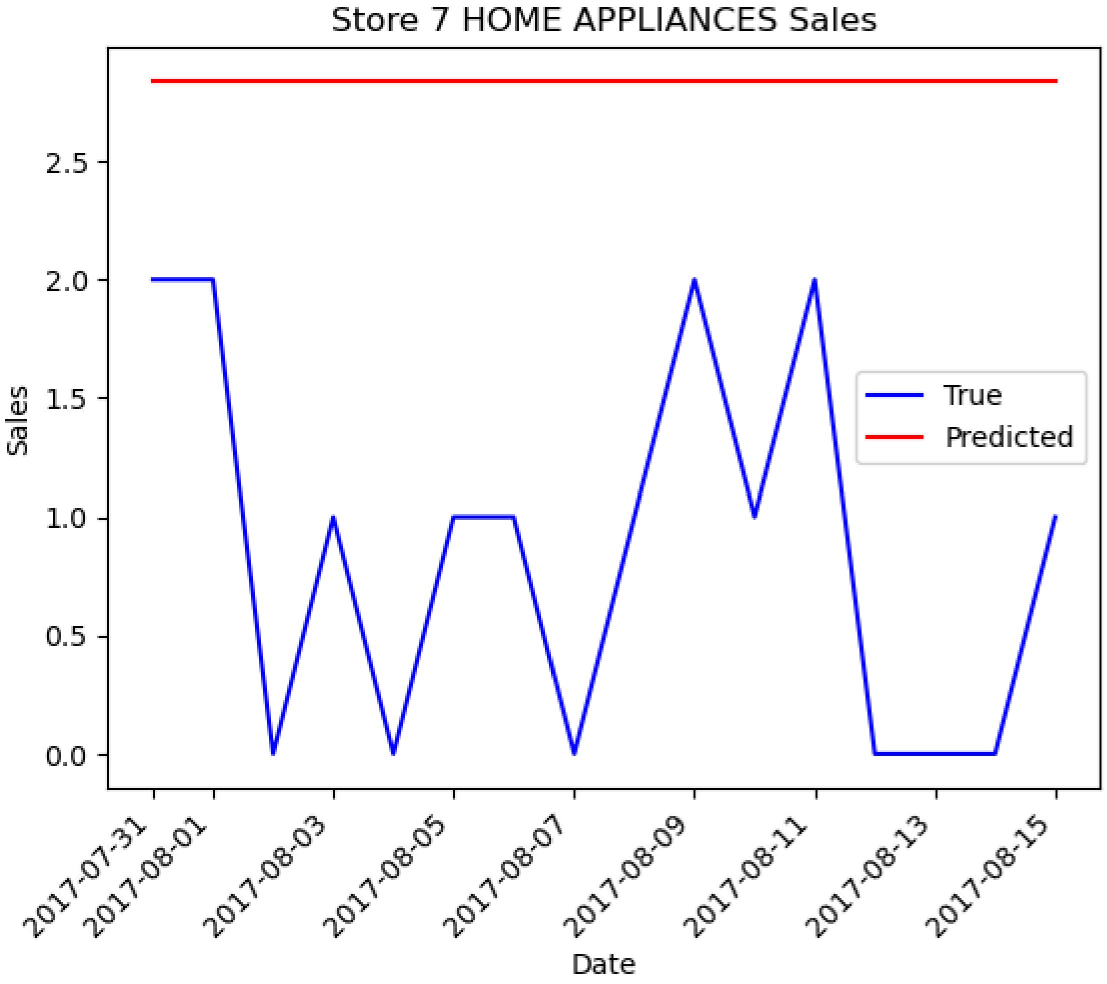

Store 7 GROCERY I Sales
True
Predicted
Sales
5000
4500
4000
3500
3000
Date
2017-07-31
2017-08-01
2017-08-03
2017-08-05
2017-08-07
2017-08-09
2017-08-11
2017-08-13
2017-08-15

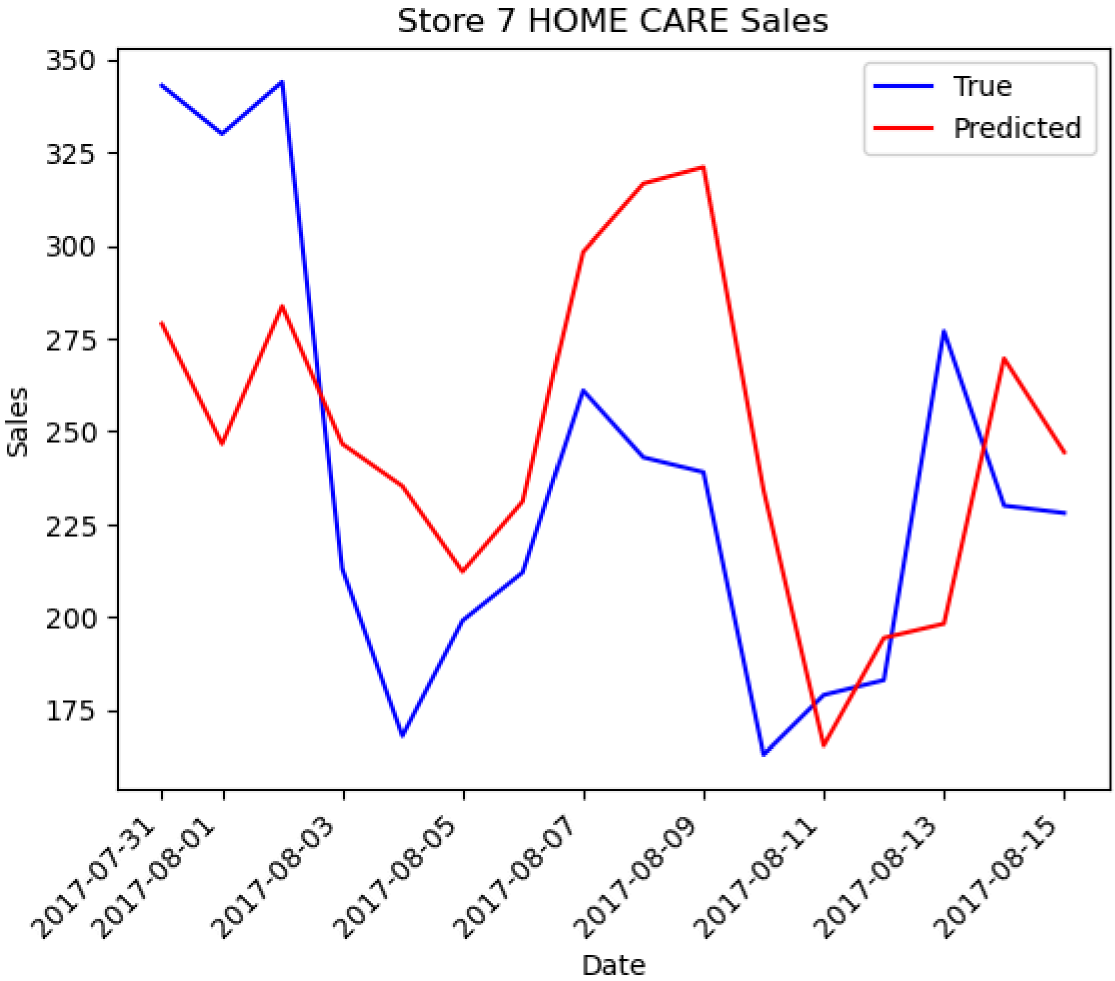

Store 7 GROCERY II Sales
True
Predicted
Sales
55
50
45
40
35
30
25
Date
2017-07-31
2017-08-01
2017-08-03
2017-08-05
2017-08-07
2017-08-09
2017-08-11
2017-08-13
2017-08-15

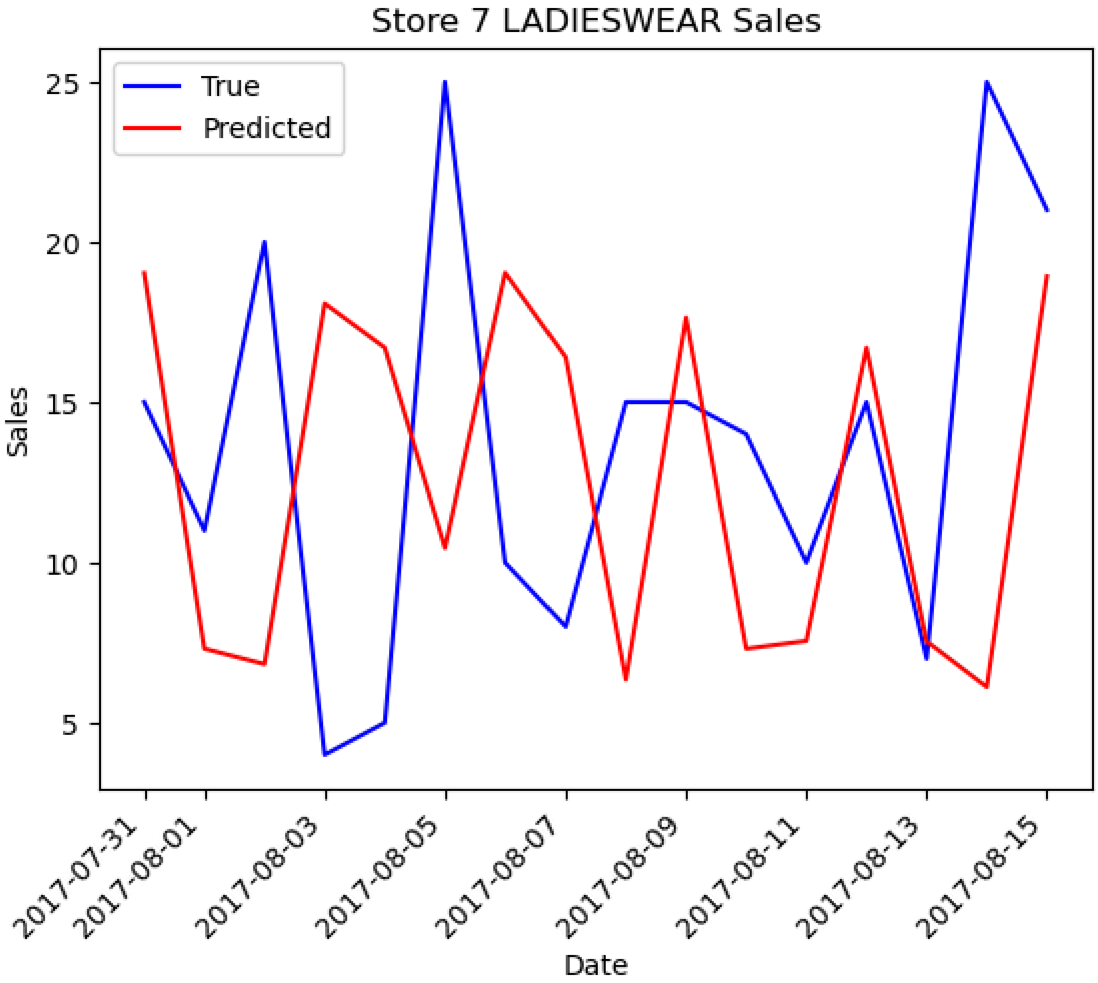

Store 7 HARDWARE Sales
True
Predicted
Sales
0
2
4
6
8
10
Date
2017-07-31
2017-08-01
2017-08-03
2017-08-05
2017-08-07
2017-08-09
2017-08-11
2017-08-13
2017-08-15

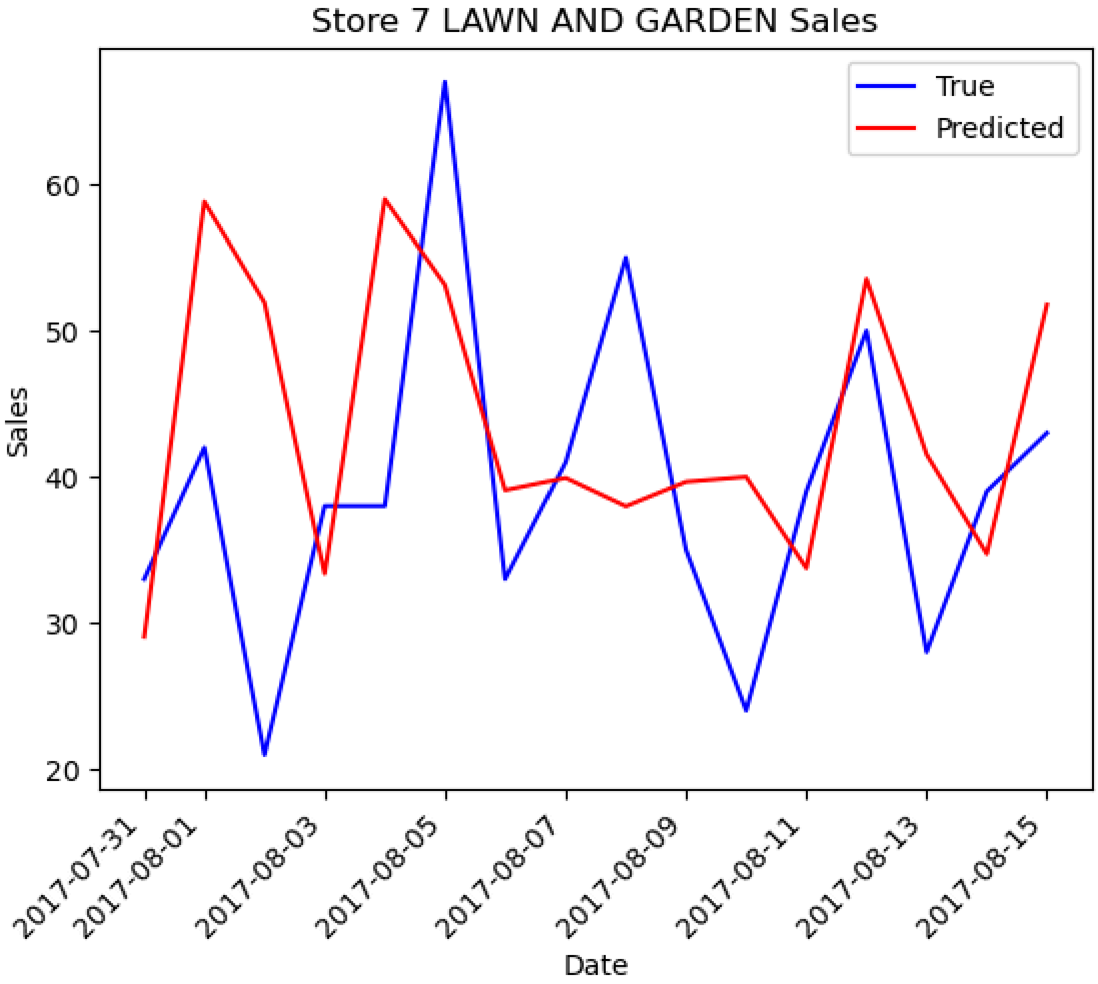

Store 7 HOME AND KITCHEN I Sales
True
Predicted
Sales
10
20
30
40
50
60
70
Date
2017-07-31
2017-08-01
2017-08-03
2017-08-05
2017-08-07
2017-08-09
2017-08-11
2017-08-13
2017-08-15

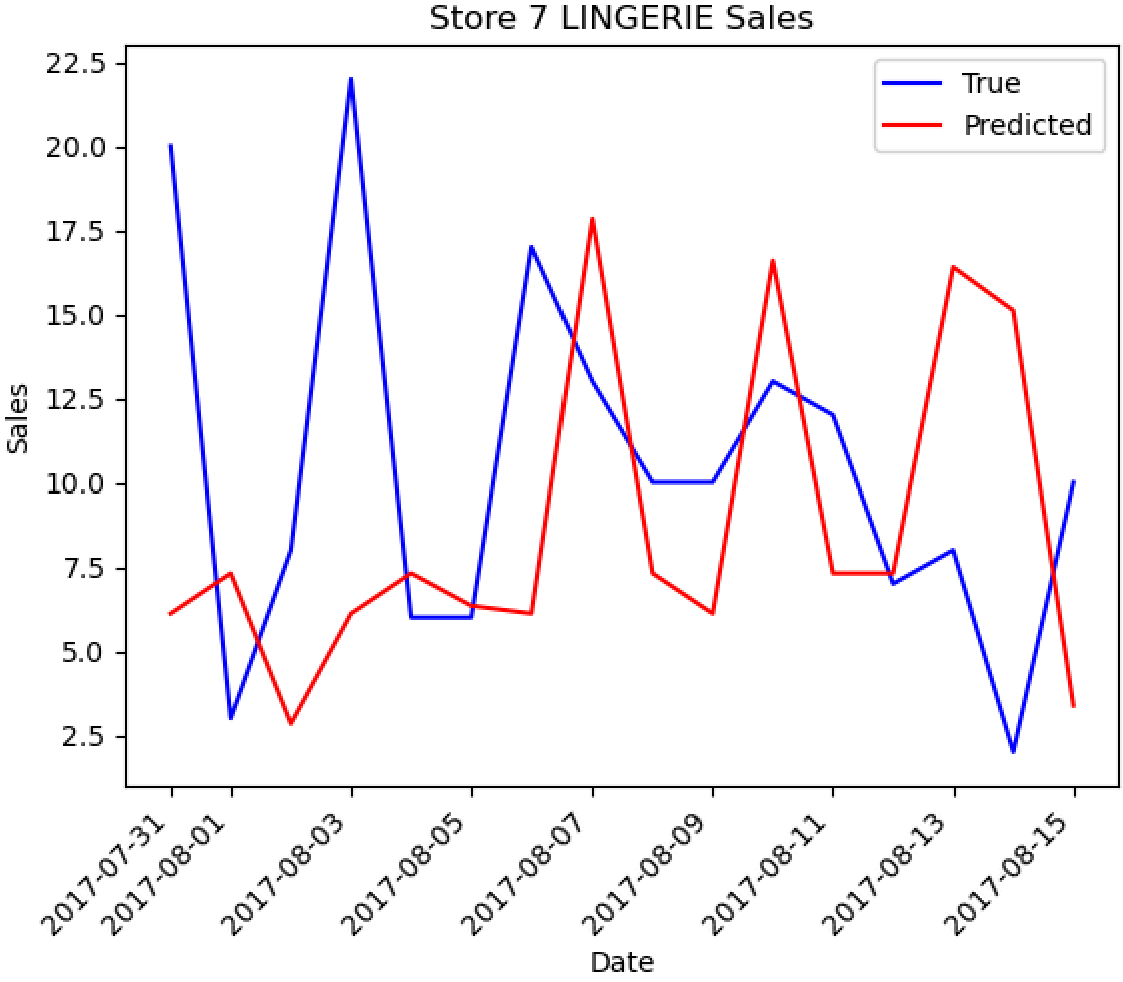

Store 7 HOME AND KITCHEN II Sales
True
Predicted
Sales
200
175
150
125
100
75
50
25
Date
2017-07-31
2017-08-01
2017-08-03
2017-08-05
2017-08-07
2017-08-09
2017-08-11
2017-08-13
2017-08-15

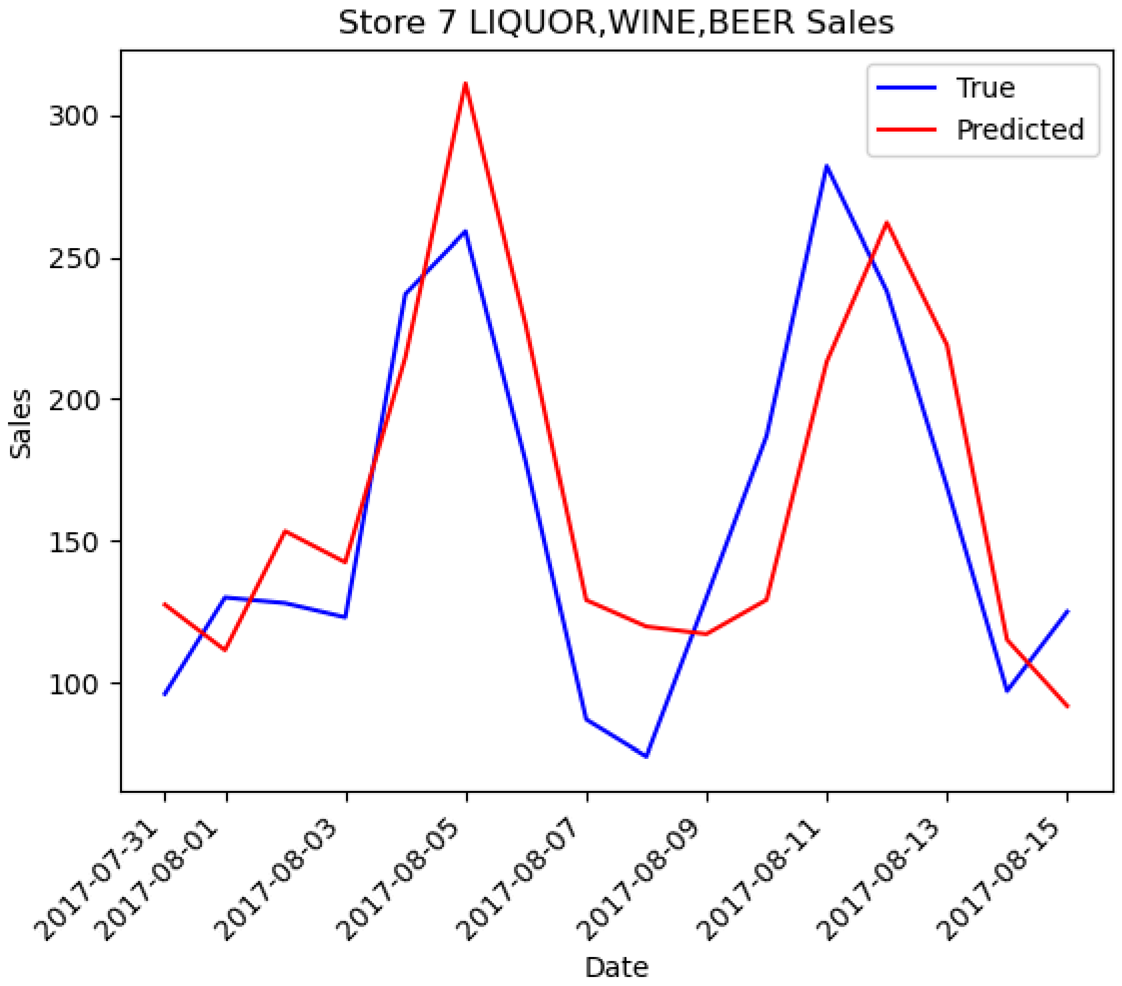

Store 7 HOME APPLIANCES Sales
True
Predicted
Sales
2.5
2.0
1.5
1.0
0.5
0.0
Date
2017-07-31
2017-08-01
2017-08-03
2017-08-05
2017-08-07
2017-08-09
2017-08-11
2017-08-13
2017-08-15

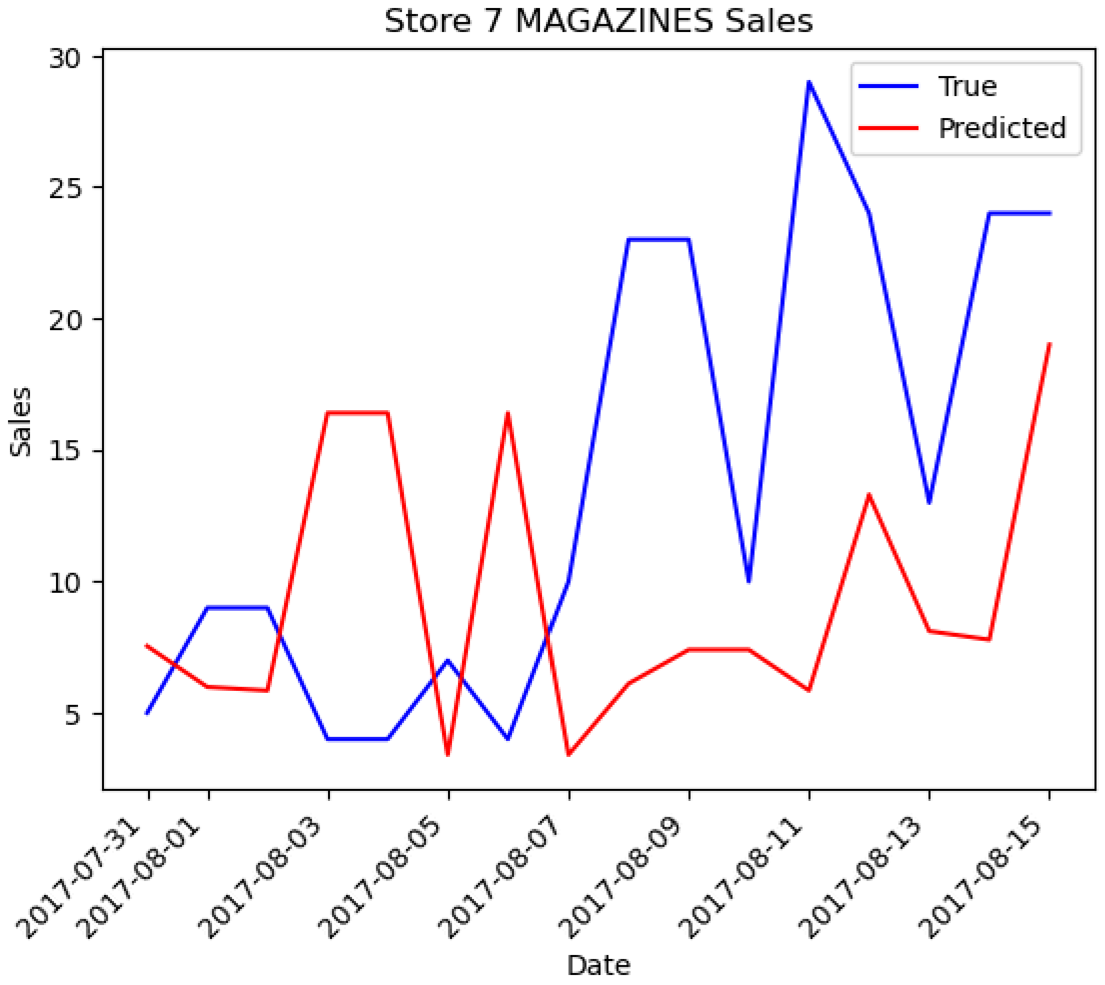

Store 7 HOME CARE Sales
True
Predicted
350
325
300
275
250
225
200
175
Sales
2017-07-31
2017-08-01
2017-08-03
2017-08-05
2017-08-07
2017-08-09
2017-08-11
2017-08-13
2017-08-15
Date

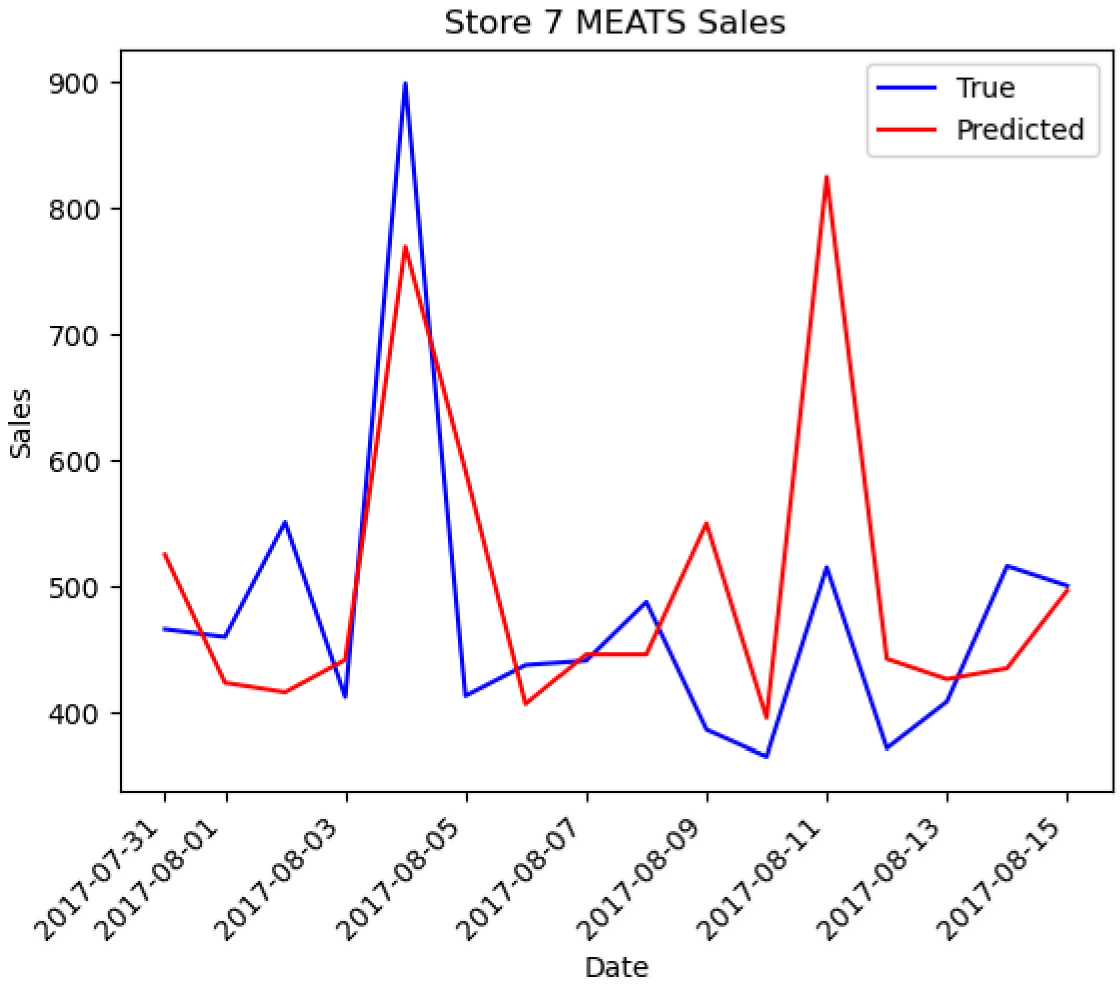

Store 7 LADIESWEAR Sales
True
Predicted
25
20
15
10
5
Sales
2017-07-31
2017-08-01
2017-08-03
2017-08-05
2017-08-07
2017-08-09
2017-08-11
2017-08-13
2017-08-15
Date

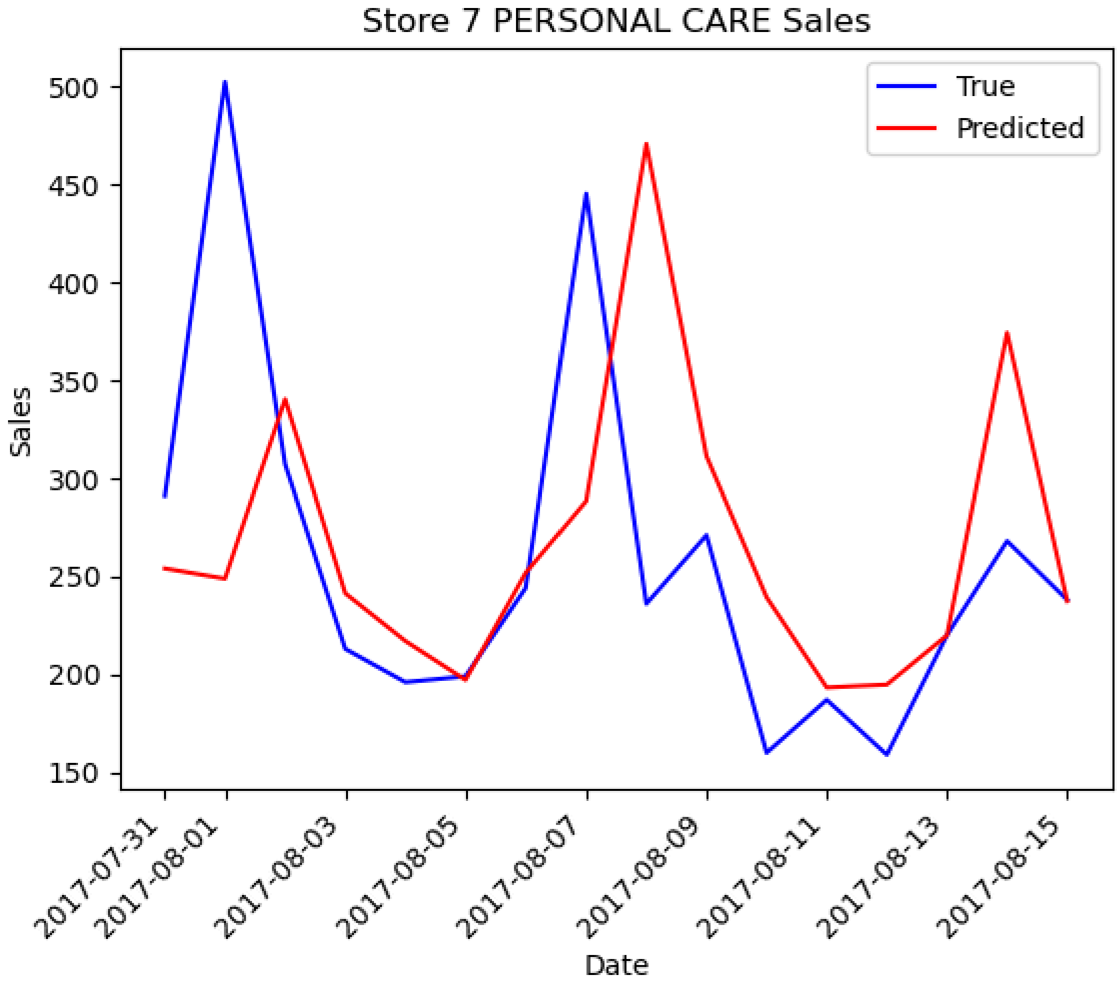

Store 7 LAWN AND GARDEN Sales
True
Predicted
Sales
20
30
40
50
60
Date
2017-07-31
2017-08-01
2017-08-03
2017-08-05
2017-08-07
2017-08-09
2017-08-11
2017-08-13
2017-08-15

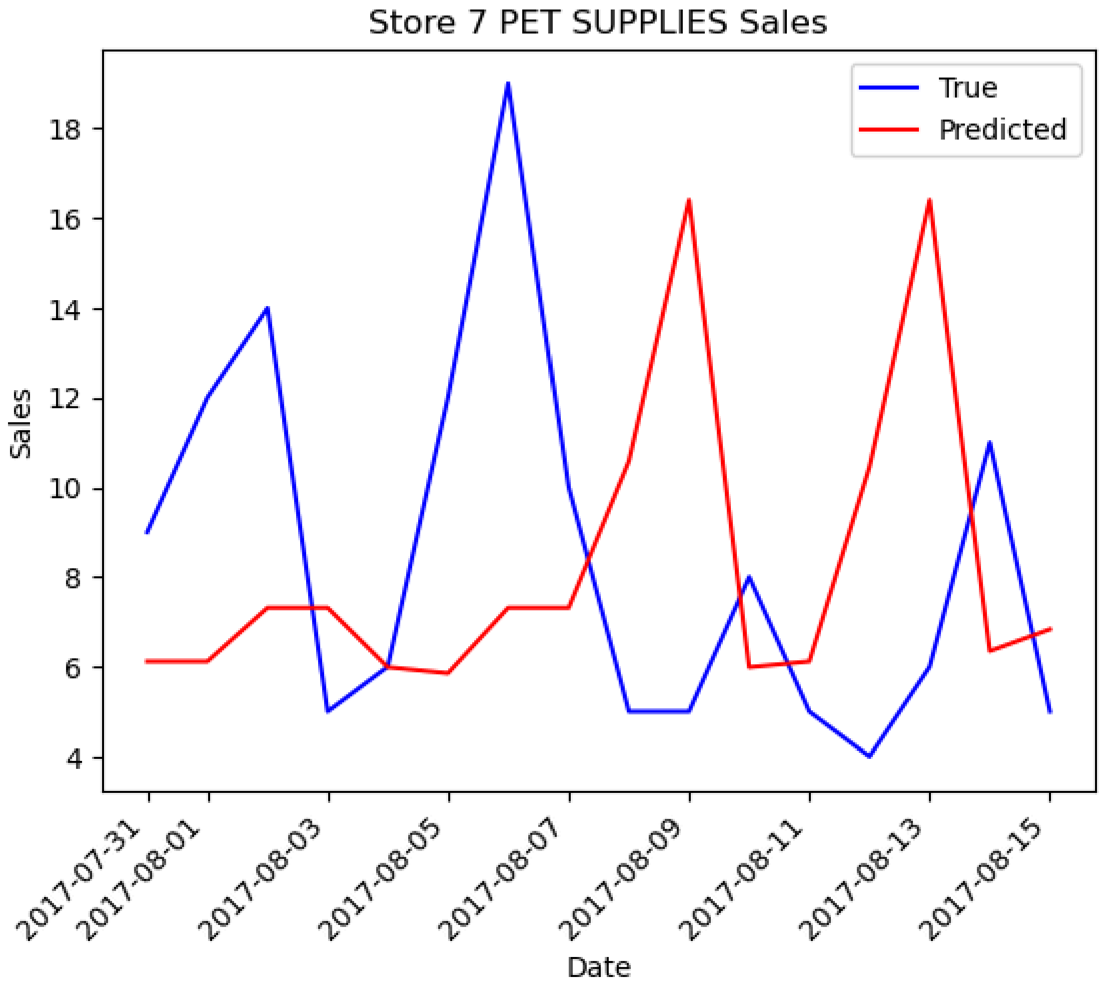

Store 7 LINGERIE Sales
True
Predicted
Sales
2.5
5.0
7.5
10.0
12.5
15.0
17.5
20.0
22.5
Date
2017-07-31
2017-08-01
2017-08-03
2017-08-05
2017-08-07
2017-08-09
2017-08-11
2017-08-13
2017-08-15

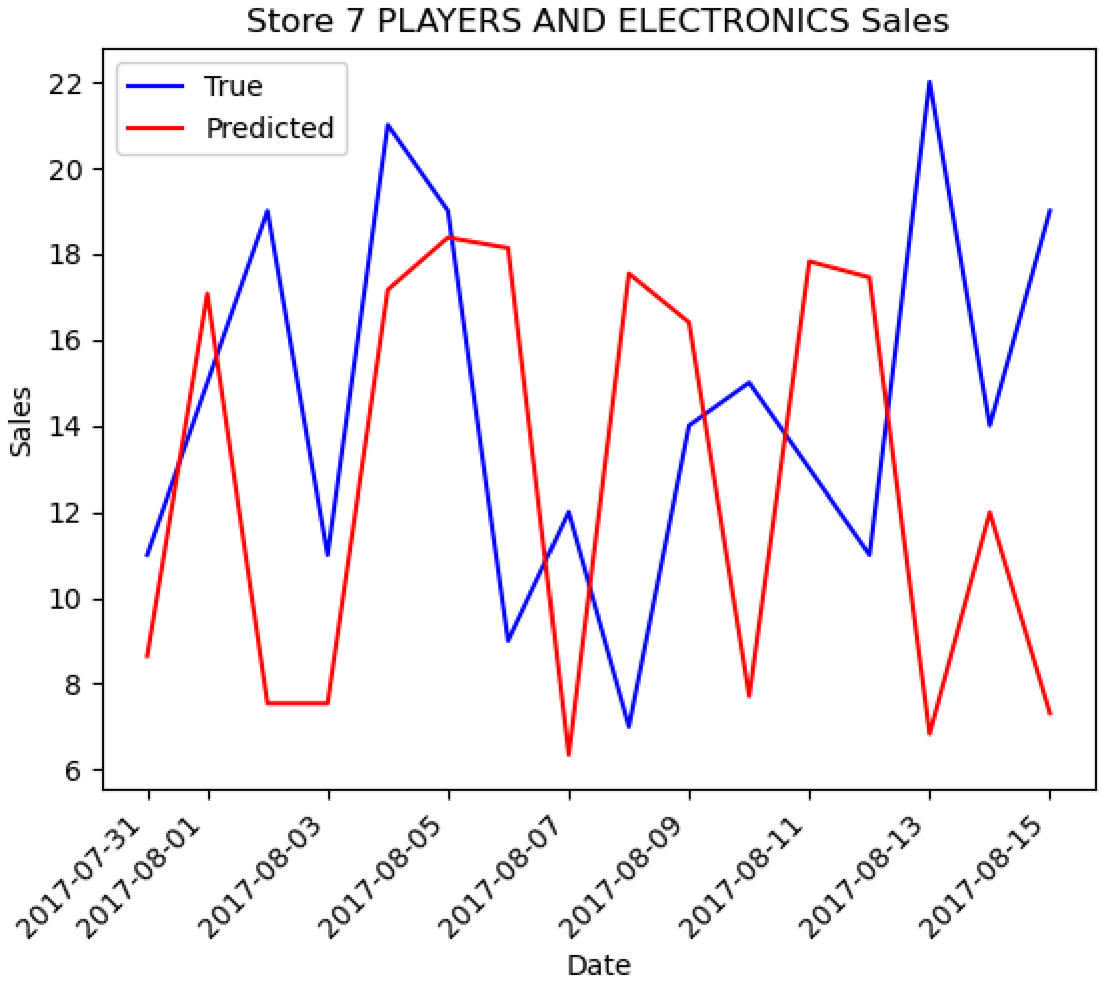

Store 7 LIQUOR,WINE,BEER Sales
True
Predicted
Sales
300
250
200
150
100
Date
2017-07-31
2017-08-01
2017-08-03
2017-08-05
2017-08-07
2017-08-09
2017-08-11
2017-08-13
2017-08-15

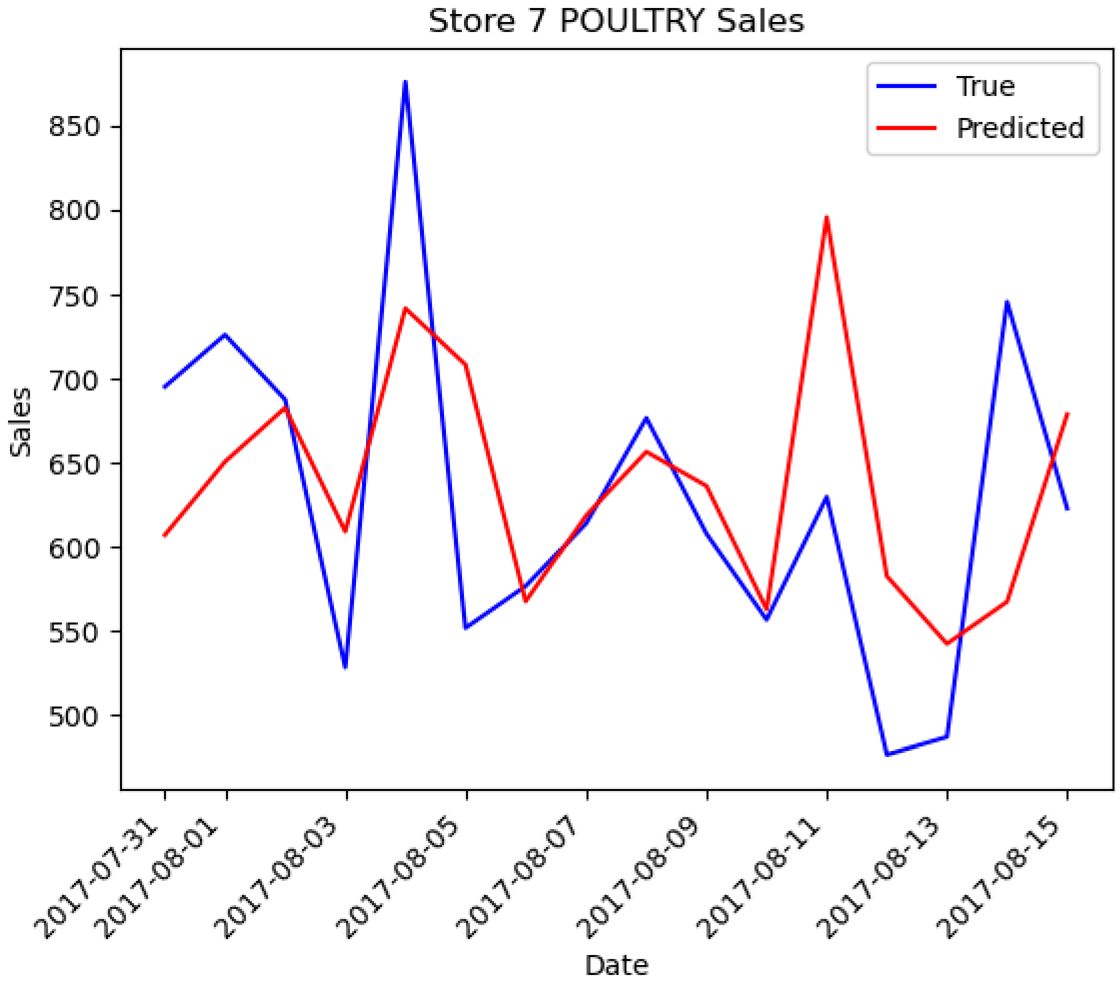

Store 7 MAGAZINES Sales
True
Predicted
Sales
30
25
20
15
10
5
Date
2017-07-31
2017-08-01
2017-08-03
2017-08-05
2017-08-07
2017-08-09
2017-08-11
2017-08-13
2017-08-15


Store 7 MEATS Sales
True
Predicted
Sales
900
800
700
600
500
400
Date
2017-07-31
2017-08-01
2017-08-03
2017-08-05
2017-08-07
2017-08-09
2017-08-11
2017-08-13
2017-08-15

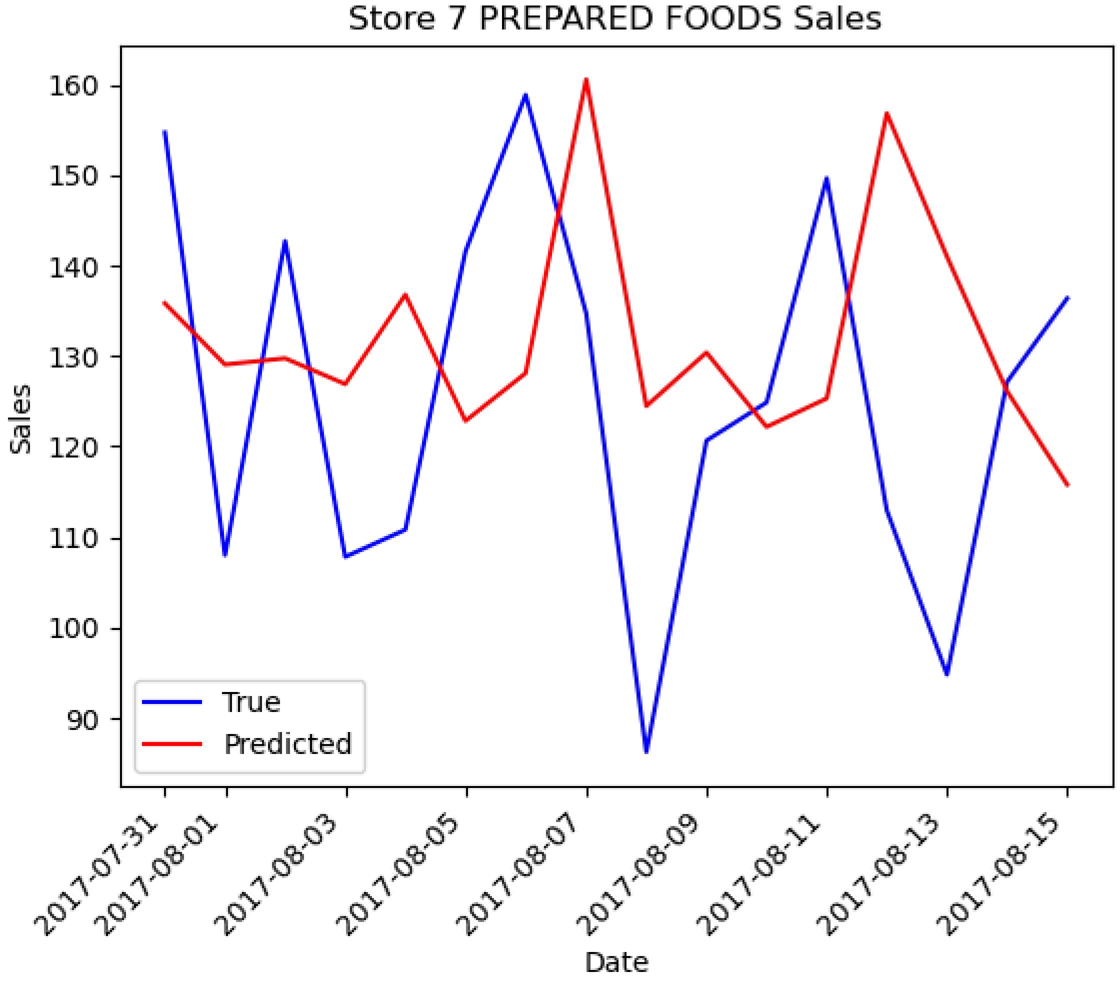

Store 7 PERSONAL CARE Sales
True
Predicted
Sales
500
450
400
350
300
250
200
150
Date
2017-07-31
2017-08-01
2017-08-03
2017-08-05
2017-08-07
2017-08-09
2017-08-11
2017-08-13
2017-08-15


Store 7 PET SUPPLIES Sales
True
Predicted
Sales
18
16
14
12
10
8
6
4
2017-07-31
2017-08-01
2017-08-03
2017-08-05
2017-08-07
2017-08-09
2017-08-11
2017-08-13
2017-08-15
Date

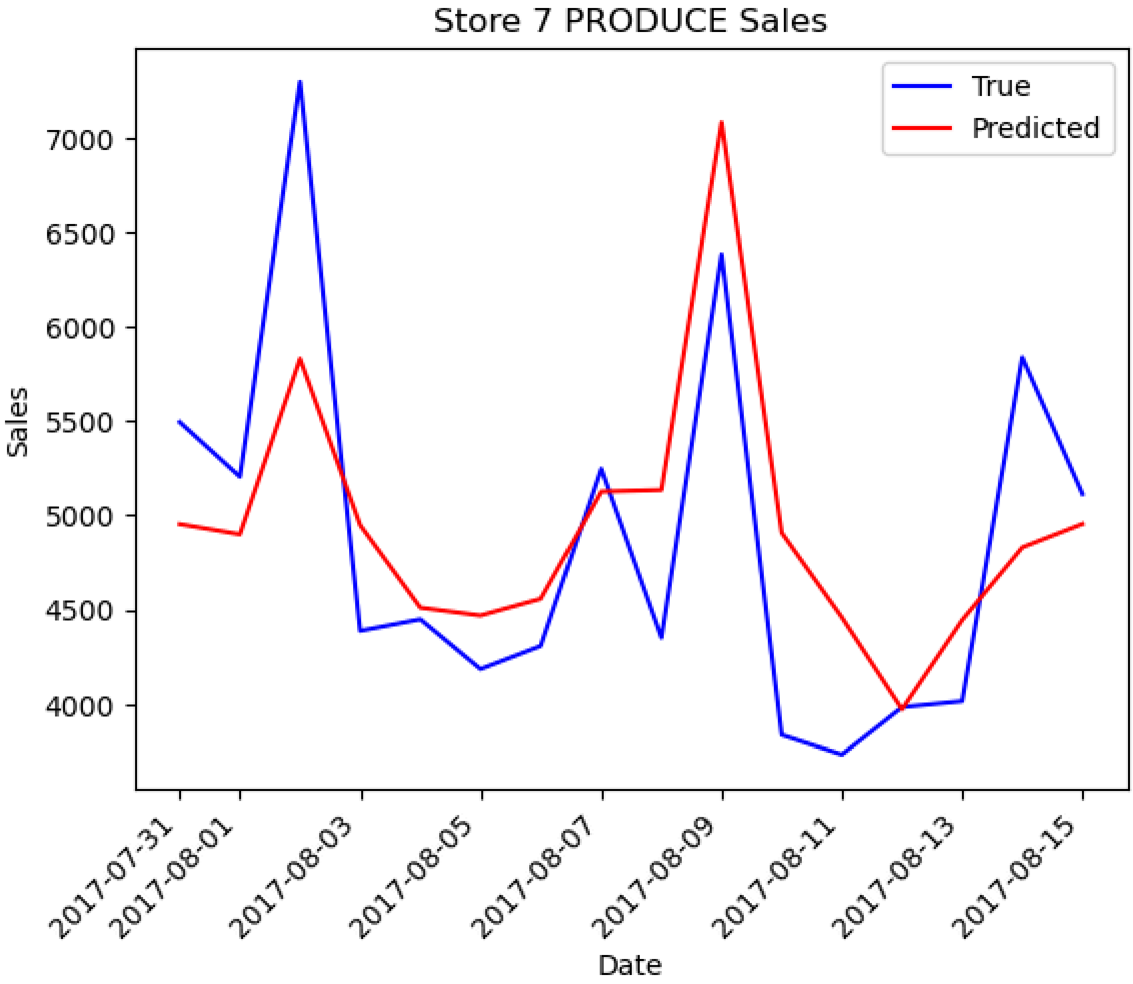

Store 7 PLAYERS AND ELECTRONICS Sales
True
Predicted
Sales
22
20
18
16
14
12
10
8
6
2017-07-31
2017-08-01
2017-08-03
2017-08-05
2017-08-07
2017-08-09
2017-08-11
2017-08-13
2017-08-15
Date

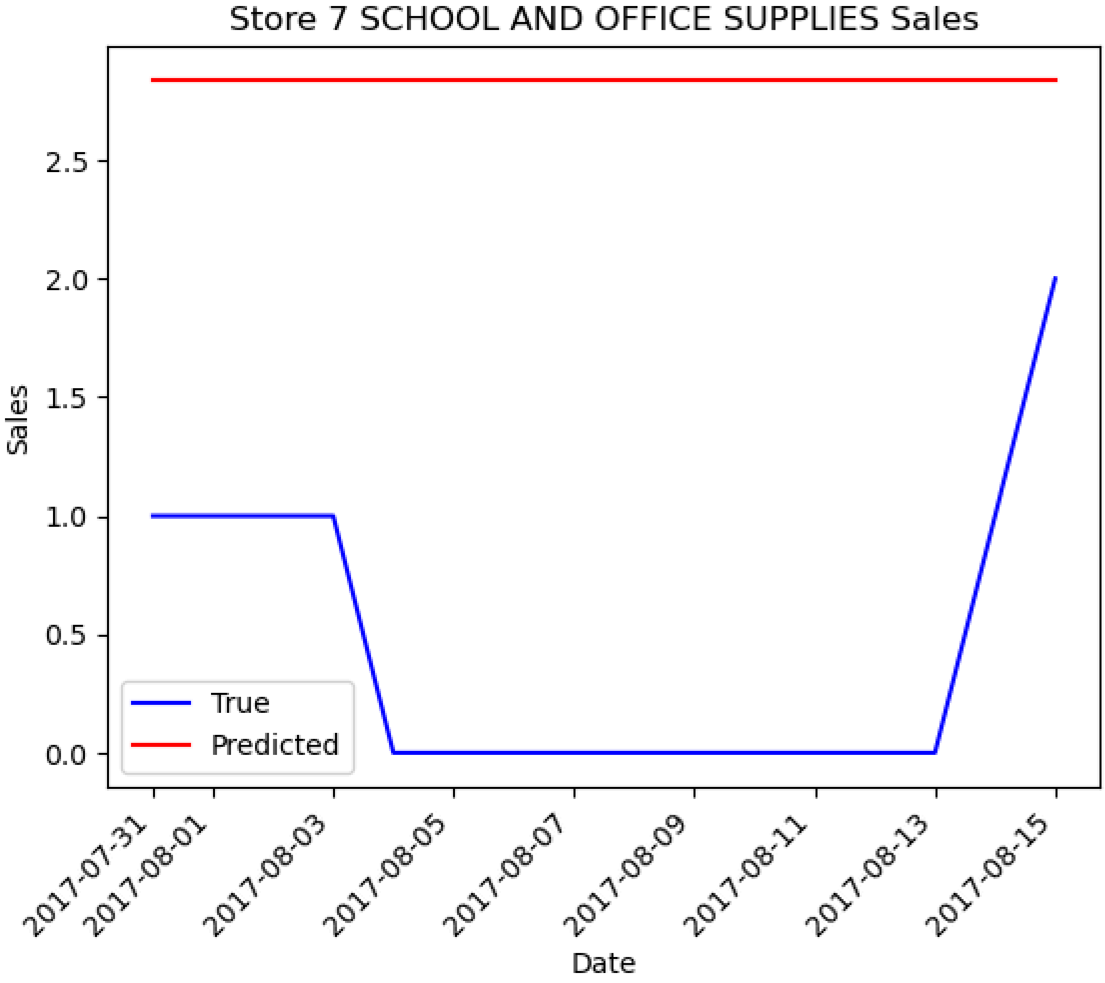


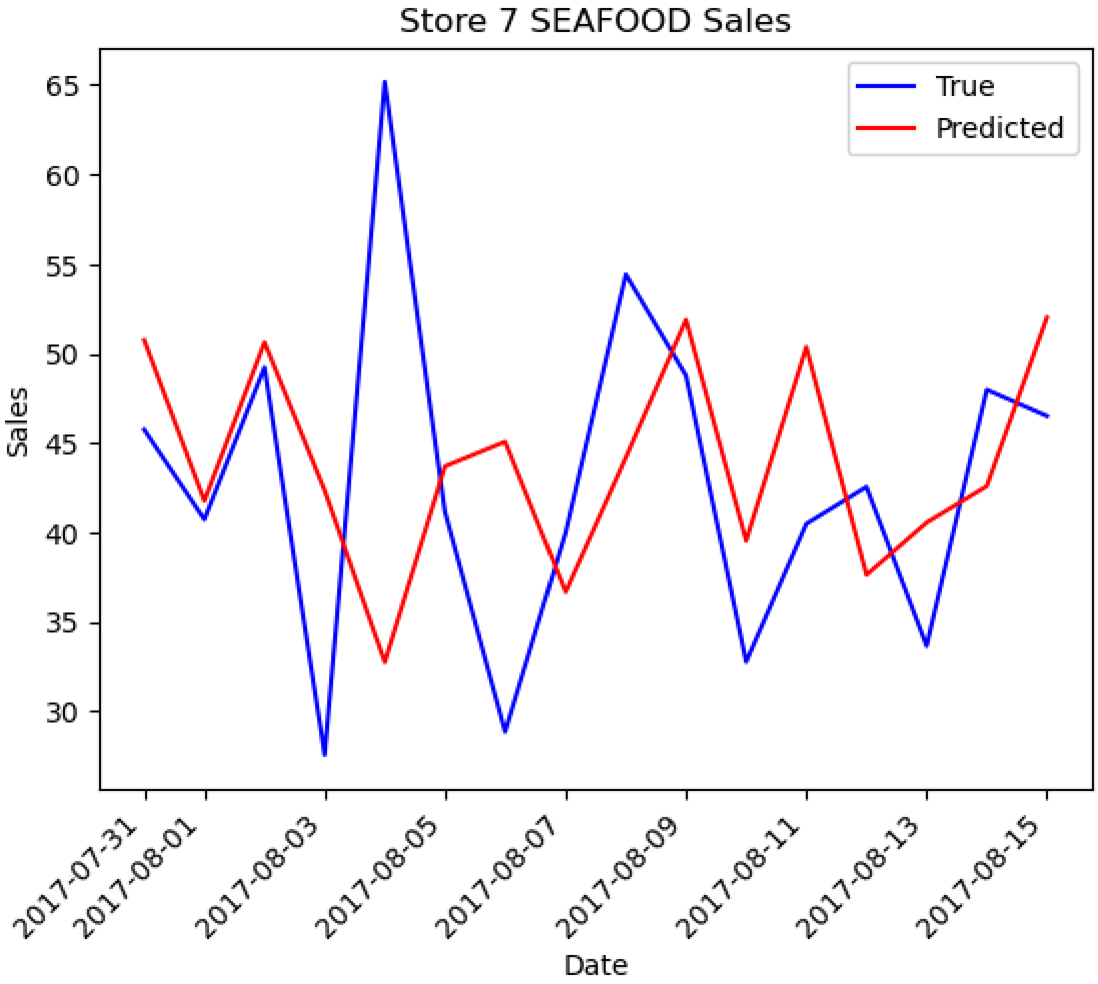
Store 7 POULTRY Sales



Store 7 PREPARED FOODS Sales


Store 7 PRODUCE Sales
True
Predicted
Sales
7000
6500
6000
5500
5000
4500
4000
2017-07-31
2017-08-01
2017-08-03
2017-08-05
2017-08-07
2017-08-09
2017-08-11
2017-08-13
2017-08-15
Date


Store 7 SCHOOL AND OFFICE SUPPLIES Sales
True
Predicted
Sales
2.5
2.0
1.5
1.0
0.5
0.0
2017-07-31
2017-08-01
2017-08-03
2017-08-05
2017-08-07
2017-08-09
2017-08-11
2017-08-13
2017-08-15
Date



In [123…

```
#MSE
train_mae = mean_absolute_error(train_predict, y_train)
test_mae = mean_absolute_error(test_predict, y_test)

#Comparing Train and Test MSE
print('Train MAE = ' + str(train_mae))
print('Test MAE = ' + str(test_mae))
```

```
Train MAE = 78.99384621479666
Test MAE = 81.39715774741313
```

In [125…

```
predict_data[500:560]
```

Out[125]:

	id	date	store_nbr	product_type	sales	predicted_sales	special_offer
3000794	3000794	2017-08-15	7	BREAD/BAKERY	659.15600	644.076424	11
3000795	3000795	2017-08-15	7	CELEBRATION	11.00000	17.834798	0
3000796	3000796	2017-08-15	7	CLEANING	1139.00000	1043.800667	9
3000797	3000797	2017-08-15	7	DAIRY	1279.00000	1324.569105	25
3000798	3000798	2017-08-15	7	DELI	172.97500	202.039874	7
3000799	3000799	2017-08-15	7	EGGS	299.00000	259.581770	0
3000800	3000800	2017-08-15	7	FROZEN FOODS	235.35100	167.341136	1
3000801	3000801	2017-08-15	7	GROCERY I	3678.00000	3735.143295	34
3000802	3000802	2017-08-15	7	GROCERY II	33.00000	40.782854	0
3000803	3000803	2017-08-15	7	HARDWARE	1.00000	2.839252	0
3000804	3000804	2017-08-15	7	HOME AND KITCHEN I	42.00000	30.641683	1
3000805	3000805	2017-08-15	7	HOME AND KITCHEN II	44.00000	72.353169	5
3000806	3000806	2017-08-15	7	HOME APPLIANCES	1.00000	2.839252	0
3000807	3000807	2017-08-15	7	HOME CARE	228.00000	244.388309	5
3000808	3000808	2017-08-15	7	LADIESWEAR	21.00000	18.930546	0
3000809	3000809	2017-08-15	7	LAWN AND GARDEN	43.00000	51.789981	0
3000810	3000810	2017-08-15	7	LINGERIE	10.00000	3.382103	0
3000811	3000811	2017-08-15	7	LIQUOR,WINE,BEER	125.00000	91.711521	5
3000812	3000812	2017-08-15	7	MAGAZINES	24.00000	19.000169	0

	id	date	store_nbr	product_type	sales	predicted_sales	special_offer	s
3000813	3000813	2017-08-15	7	MEATS	500.31198	496.527311	0	
3000814	3000814	2017-08-15	7	PERSONAL CARE	238.00000	237.363753	9	
3000815	3000815	2017-08-15	7	PET SUPPLIES	5.00000	6.831798	0	
3000816	3000816	2017-08-15	7	PLAYERS AND ELECTRONICS	19.00000	7.312159	1	
3000817	3000817	2017-08-15	7	POULTRY	622.67300	678.674844	0	
3000818	3000818	2017-08-15	7	PREPARED FOODS	136.36100	115.751031	0	
3000819	3000819	2017-08-15	7	PRODUCE	5113.96100	4954.927469	6	
3000820	3000820	2017-08-15	7	SCHOOL AND OFFICE SUPPLIES	2.00000	2.839252	0	
3000821	3000821	2017-08-15	7	SEAFOOD	46.50700	52.024930	0	

28 rows × 27 columns

```
In [ ]:
```