Regression_Practice_Answer_2

August 4, 2023

1 Bài tập thực hành 2

1.1 Yêu cầu

Dự đoán doanh thu xe hơi Hyundai dòng Elantra trong năm 2013 và đầu 2014, dựa vào dữ liệu trước đó

1.2 Dữ liệu

Dữ liệu được ghi trong file csv với các trường (Month, Year, ElantraSales, Unemployment, Queries, CPI_energy, CPI_all). Giá trị cần dự đoán sẽ là ElantraSales.

1.3 Đánh giá

Đánh giá mô hình dựa trên * Độ đo tiêu chuẩn của ML: RMSE = $\sqrt{\text{avg}\left(y^{(n)} - \hat{y}^{(n)}\right)^2}$ * Độ đo của business requirements: Mean relative errors = avg $\left(\frac{\left|y^{(n)} - \hat{y}^{(n)}\right|}{y^{(n)}}\right) \times 100\%$

2 Các bước tiến hành

2.1 Đọc dữ liệu

[3]:
```
import pandas as pd
import numpy as np  # thư viện cho tính toán nói chung

df = pd.read_csv('elantra.csv')
```

[4]:
```
df.tail(10)
```

[4]:

	Month	Year	ElantraSales	Unemployment	Queries	CPI_energy	CPI_all
40	10	2012	14512	7.8	257	256.389	231.652
41	10	2013	14876	7.2	223	243.374	233.782
42	11	2010	8631	9.8	161	219.303	219.544
43	11	2011	12414	8.6	255	247.092	227.136
44	11	2012	15923	7.8	246	248.136	231.190
45	11	2013	16751	7.0	231	242.301	234.033
46	12	2010	13096	9.4	170	227.190	220.437
47	12	2011	13025	8.5	253	243.015	227.093

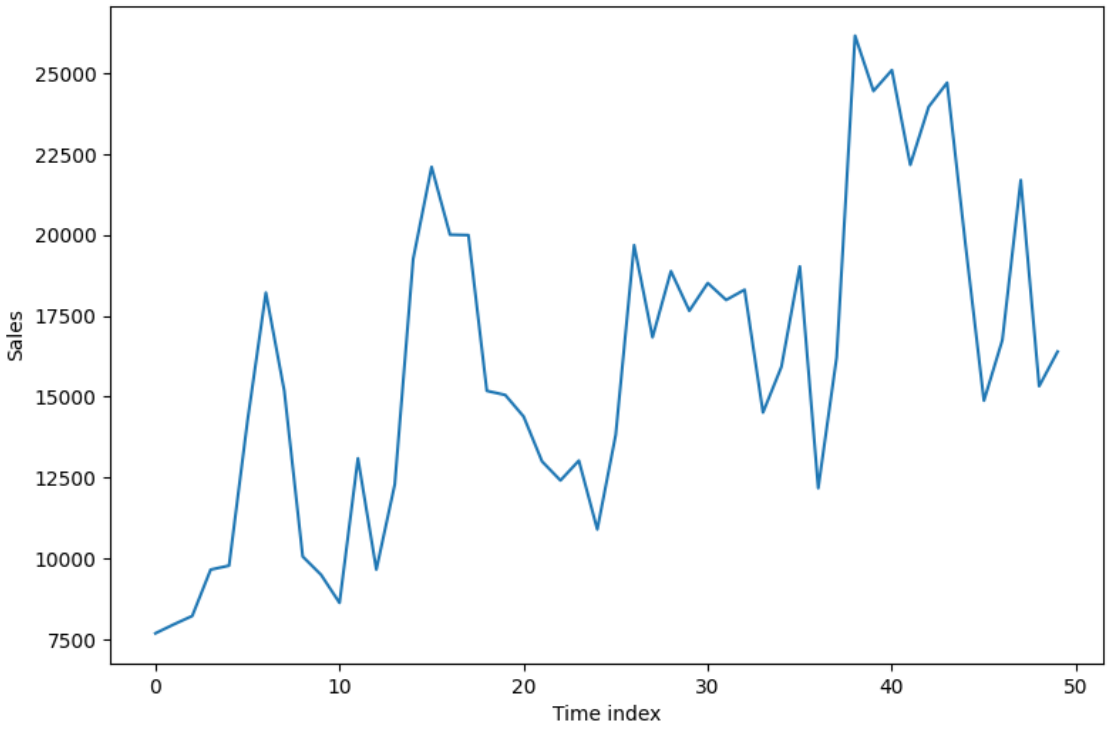
```
48     12  2012          19024            7.9       275      244.698  231.099
49     12  2013          21692            6.7       279      246.189  234.594
```

[5]:
```
##### exercise #####
# Yêu cầu: Sắp xếp lại thứ tự các hàng dữ liệu theo tháng/năm
# Gợi ý: sử dụng df.sort_values và df.reset_index
#######################
df = df.sort_values(by=['Year', 'Month'])
df = df.reset_index(drop=True)
df.head(10)
```

[5]:

	Month	Year	ElantraSales	Unemployment	Queries	CPI_energy	CPI_all
0	1	2010	7690	9.7	153	213.377	217.466
1	2	2010	7966	9.8	130	209.924	217.251
2	3	2010	8225	9.9	138	209.163	217.305
3	4	2010	9657	9.9	132	209.024	217.376
4	5	2010	9781	9.6	177	206.172	217.299
5	6	2010	14245	9.4	138	204.161	217.285
6	7	2010	18215	9.5	156	206.834	217.677
7	8	2010	15181	9.5	202	208.927	218.012
8	9	2010	10062	9.5	150	209.850	218.281
9	10	2010	9497	9.5	178	216.655	219.024

[6]:
```
import matplotlib.pyplot as plt

plt.figure(figsize=(9,6))

plt.plot(df.ElantraSales.values)

plt.xlabel('Time index')

plt.ylabel('Sales')


# function to show the plot
plt.show()
```



[7]:
```
numeric_feats = df.columns.drop(["ElantraSales", "Month", "Year"])
numeric_feats
```

[7]:
```
Index(['Unemployment', 'Queries', 'CPI_energy', 'CPI_all'], dtype='object')
```

[8]:
```
df_train = df[df.Year < 2013]
df_test = df[df.Year >= 2013]

y_train = df_train.ElantraSales.values
y_test = df_test.ElantraSales.values
```

feature scaling

[9]:
```
# Chuẩn hóa dữ liệu bằng StandardScaler, dữ liệu được chuẩn hóa theo dạng x ->␣
↪(x-mean)/std
# Nếu x có phân phối Gauss, dữ liệu chuẩn hóa sẽ thuộc phân phối N(0,1)
from sklearn.preprocessing import StandardScaler

# Scaling outputs obtained from TRAINING SET
scaler = StandardScaler().fit(df_train[numeric_feats])

X_train = scaler.transform(df_train[numeric_feats])
X_test = scaler.transform(df_test[numeric_feats])
```

2.2 Xây dựng Mô hình

[10]:
```
###### exercise #####
# Yêu cầu: Xây dựng và huấn luyện mô hình Linear Regression
# Gợi ý: sử dụng hàm fit() nhưng trong bài thực hành 1
######################
from sklearn.linear_model import LinearRegression

model1 = LinearRegression()
model1.fit(X_train, y_train)
```

[10]:
```
LinearRegression()
```

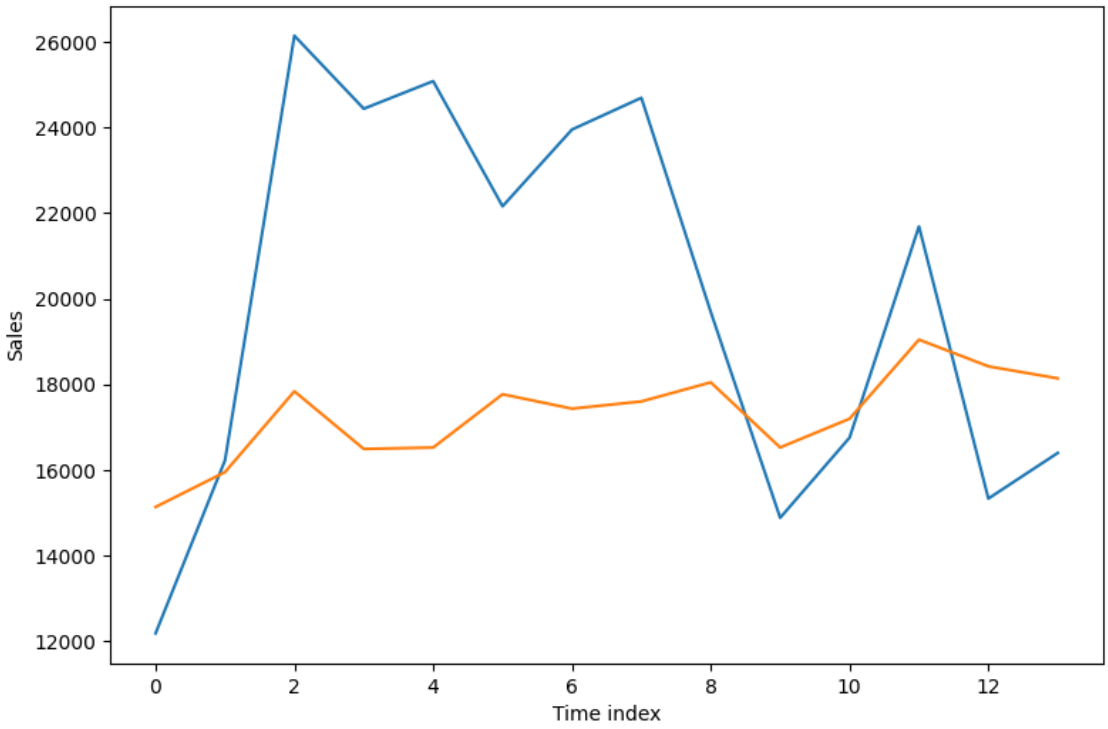
2.3 Đánh giá

[11]:
```
from sklearn.metrics import mean_squared_error

def relative_error(y_true, y_pred):
    errors = np.abs(y_pred - y_true).astype(float) / y_true
    return np.mean(errors)*100
```

[12]:
```
y_pred_test = model1.predict(X_test)
print ('RMSE: {:.2f}'.format(np.sqrt(mean_squared_error(y_test, y_pred_test))))
print ('Mean relative errors: {:.1f}%'.format(relative_error(y_test,␣
 ↪y_pred_test)))
```

```
RMSE: 5017.35
Mean relative errors: 19.0%
```

[13]:
```
###### exercise #####
# Yêu cầu: Vẽ biểu đồ đường so sánh y_test và y_pred_test
# Gợi ý: sử dụng matplotlib như bài thực hành 1
######################
import matplotlib.pyplot as plt

plt.figure(figsize=(9,6))

plt.plot(y_test)
plt.plot(y_pred_test)

plt.xlabel('Time index')

plt.ylabel('Sales')


# function to show the plot
plt.show()
```



Kết quả dự đoán không khớp một chút nào so với dữ liệu thật

Lý do có thể là vì chúng ta chưa tận dụng hết thông tin của dữ liệu

Quan sát thấy doanh thu có xu hướng biến động theo từ tháng trong một năm

=> Tận dụng thông tin tháng hiệu quả. Có thể xây dựng mô hình regression với đặc trưng Month theo kiểu categorical kết hợp với các đặc trưng khác.

2.4 Giải pháp cải tiến

[14]:
```
month_onehot_train = pd.get_dummies(df_train.Month)
month_onehot_train.head()
```

[14]:

	1	2	3	4	5	6	7	8	9	10	11	12
0	1	0	0	0	0	0	0	0	0	0	0	0
1	0	1	0	0	0	0	0	0	0	0	0	0
2	0	0	1	0	0	0	0	0	0	0	0	0
3	0	0	0	1	0	0	0	0	0	0	0	0
4	0	0	0	0	1	0	0	0	0	0	0	0

[15]:
```
###### exercise #####
# Yêu cầu: Ghép đặc trưng Month_1, ..., Month_12 vào các đặc trưng đang có, kết␣
 ↪quả ở dạng numpy array
# Gợi ý: sử dụng np.hstack
```

```
######################
X_train = np.hstack((X_train, month_onehot_train))
X_train[0]
```

[15]:
```
array([ 1.24576653, -1.25517995, -1.21438113, -1.35903692,  1.        ,
        0.        ,  0.        ,  0.        ,  0.        ,  0.        ,
        0.        ,  0.        ,  0.        ,  0.        ,  0.        ,
        0.        ])
```

[16]:
```
# Tương tự với X_test
X_test = np.hstack((X_test, pd.get_dummies(df_test.Month)))
```

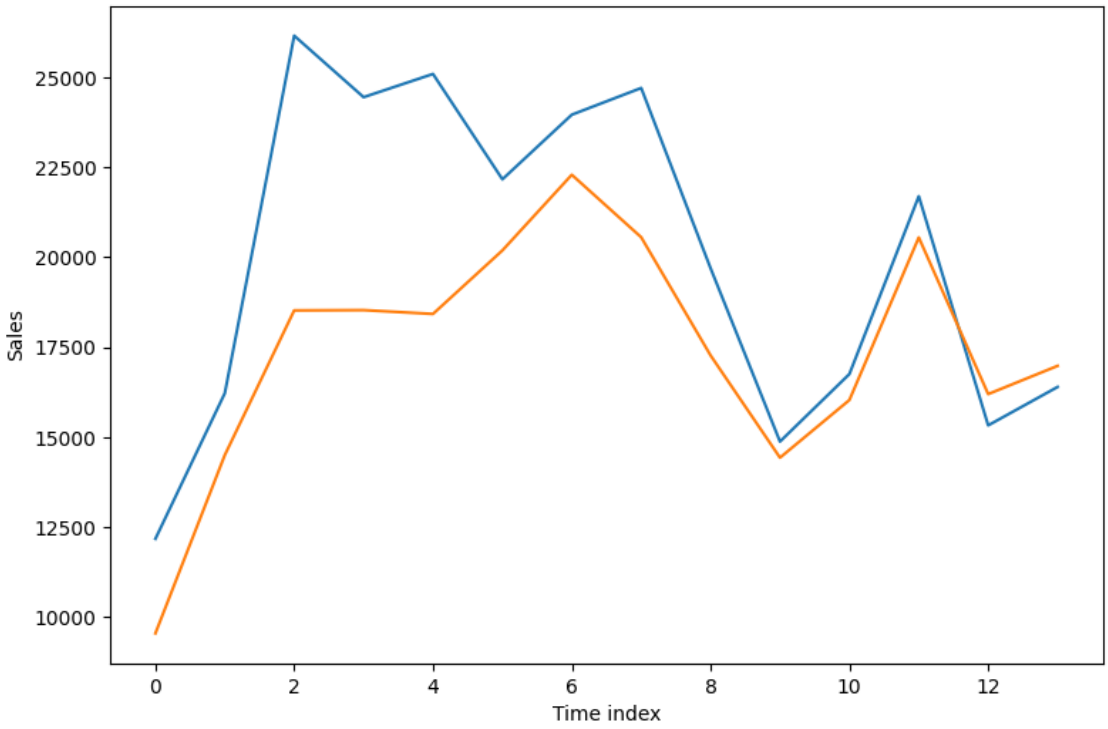
[17]:
```
model1.fit(X_train, y_train)
```

[17]:
```
LinearRegression()
```

[18]:
```
y_pred_test = model1.predict(X_test)
print ('RMSE: {:.2f}'.format(np.sqrt(mean_squared_error(y_test, y_pred_test))))
print ('Mean relative errors: {:.1f}%'.format(relative_error(y_test,␣
 ↪y_pred_test)))
```

```
RMSE: 3590.37
Mean relative errors: 12.8%
```

[19]:
```
plt.figure(figsize=(9,6))

plt.plot(y_test)
plt.plot(y_pred_test)

plt.xlabel('Time index')

plt.ylabel('Sales')


# function to show the plot
plt.show()
```


25000
22500
20000
17500
15000
12500
10000
Sales
0
2
4
6
8
10
12
Time index