

Practice_2

September 2, 2021

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 elantra.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: $RMSE = \sqrt{\text{avg} (y^{(n)} - \hat{y}^{(n)})^2}$ * Độ đo của business requirements: Mean relative errors = $\text{avg} \left(\frac{|y^{(n)} - \hat{y}^{(n)}|}{y^{(n)}} \right) \times 100\%$

2 Đọc dữ liệu

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

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

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

```
[43]:
```

	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

```
[1]: ##### 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
#####
```

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

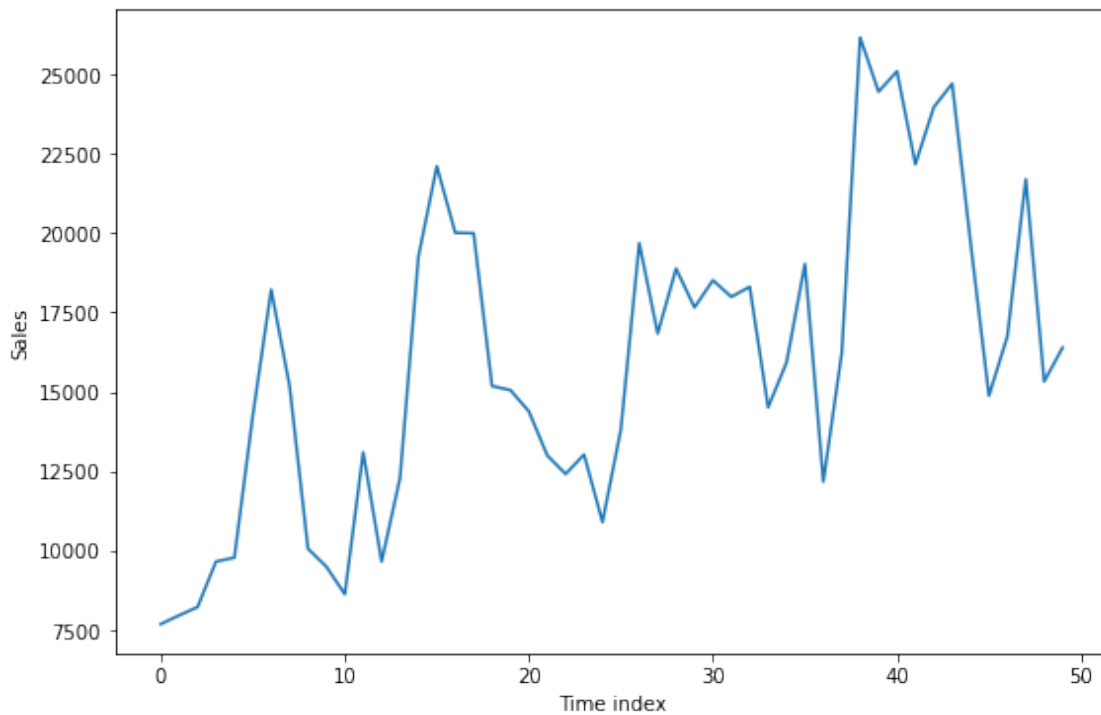
plt.figure(figsize=(9,6))

plt.plot(df.ElantraSales.values)

plt.xlabel('Time index')

plt.ylabel('Sales')

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



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

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

```
[47]: 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

```
[48]: # Chuẩn hóa dữ liệu bằng StandardScaler, dữ liệu được chuẩn hóa theo dạng  $x \rightarrow \frac{x - \text{mean}}{\text{std}}$ 
      # Nếu  $x$  có phân phối Gauss, dữ liệu chuẩn hóa sẽ thuộc phân phối  $N(0,1)$ 
      from sklearn.preprocessing import StandardScaler

      scaler = StandardScaler().fit(df_train[numeric_feats])

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

3 Xây dựng Mô hình

```
[2]: ##### exercise #####
      # Yêu cầu: Xây dựng và huấn luyện mô hình Linear Regression
      # Gợi ý: sử dụng hàm fit() như trong bài thực hành 1
      #####
```

4 Đánh giá

```
[50]: 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
```

```
[51]: 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%

```
[ ]: ##### exercise #####
      # Yêu cầu: Vẽ biểu đồ đường so sánh  $y_{\text{test}}$  và  $y_{\text{pred\_test}}$ 
```

```
# Gợi ý: sử dụng matplotlib như bài thực hành 1
#####
```

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 từng 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.

5 Giải pháp cải tiến

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

```
[53]:
```

	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

```
[3]: ##### 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
      #####
```

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

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

```
[56]: LinearRegression()
```

```
[57]: 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%

```
[58]: 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()
```

