선형 회귀에 의한 의료보험 예측

MEDICAL INSURANCE FORECAST BY LINEAR REGRESSION

목차

데이터셋

- 선형 회귀 모델을 만들어 의료 비용을 예측
- 나이, 성별, bmi, 자녀의 수, 흡연 여부, 거주 지역, 의료 비용 등의 칼럼

```
medical.columns.unique()
```

```
Index(['age', 'sex', 'bmi', 'children', 'smoker', 'region', 'charges'], dtype='object')
```

- 총 1338열과 7행의 데이터

```
#Determining the number of rows and columns
medical.shape
```

```
(1338, 7)
```

데이터셋

데이터 집합의 모든 숫자 열에 대한 요약

```
medical.describe()  #summary of all the numeric columns in the dataset
```

	age	bmi	children	charges
count	1338.000000	1338.000000	1338.000000	1338.000000
mean	39.207025	30.663397	1.094918	13270.422265
std	14.049960	6.098187	1.205493	12110.011237
min	18.000000	15.960000	0.000000	1121.873900
25%	27.000000	26.296250	0.000000	4740.287150
50%	39.000000	30.400000	1.000000	9382.033000
75%	51.000000	34.693750	2.000000	16639.912515
max	64.000000	53.130000	5.000000	63770.428010

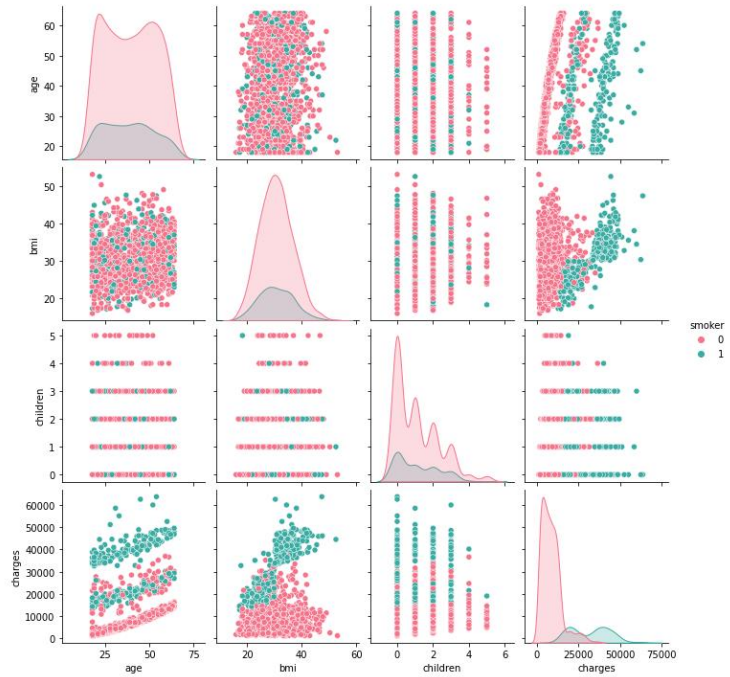
성별과 흡연 여부는 각각 2가지 특성만 존재하기 때문에 부울 값으로 맵핑

```
#Mapping
medical['sex'] = medical['sex'].map({'male': 0, 'female': 1})
medical['smoker'] = medical['smoker'].map({'yes': 1, 'no': 0})
medical.head()
```

	age	sex	bmi	children	smoker	region	charges
0	19	1	27.900	0	1	southwest	16884.92400
1	18	0	33.770	1	0	southeast	1725.55230
2	28	0	33.000	3	0	southeast	4449.46200
3	33	0	22.705	0	0	northwest	21984.47061
4	32	0	28.880	0	0	northwest	3866.85520

데이터 시각화

```
#Pairplot of all numerical variables
sns.pairplot(medical, vars=["age", 'bmi','children','charges'],hue='smoker',palette="husl")
plt.show()
```



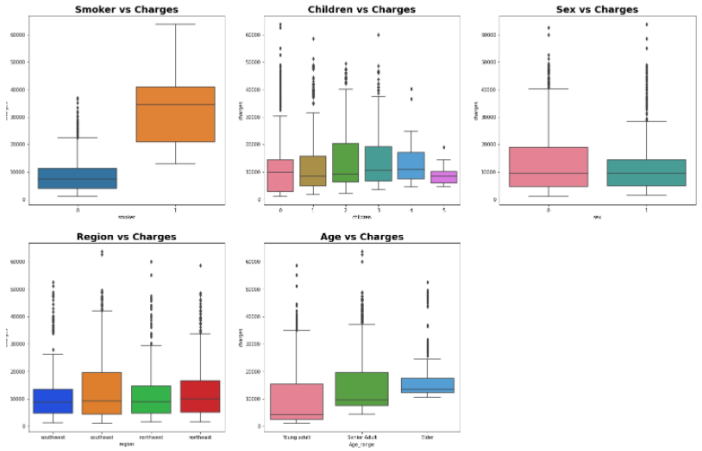

```
#Import necessary libraies
import matplotlib.pyplot as plt
import seaborn as sns
```

- 데이터 시각화를 위한 라이브러리들 import

- 나이 대 의료 비용 plot에서 나이에 따라 의료 비용이 증가하는 추세가 관찰된다.
- BMI는 의료 비용과 연관성이 있는 것으로 보인다.
- 흡연과 의료 비용 사이에는 강한 연관성이 있는 것으로 보인다.

데이터 시각화

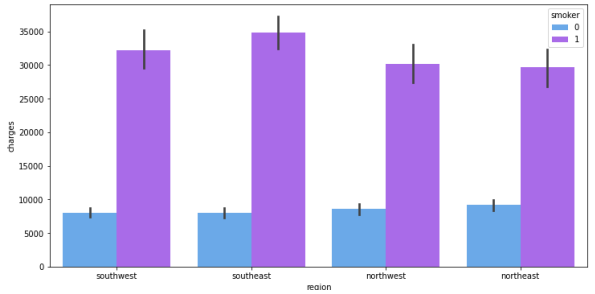
```
plt.figure(figsize=(25, 16))
plt.subplot(2,3,1)
sns.boxplot(x = 'smoker', y = 'charges', data = medical)
plt.title('Smoker vs Charges',fontweight="bold", size=20)
plt.subplot(2,3,2)
sns.boxplot(x = 'children', y = 'charges', data = medical,palette="husl")
plt.title('Children vs Charges',fontweight="bold", size=20)
plt.subplot(2,3,3)
sns.boxplot(x = 'sex', y = 'charges', data = medical, palette= 'husl')
plt.title('Sex vs Charges',fontweight="bold", size=20)
plt.subplot(2,3,4)
sns.boxplot(x = 'region', y = 'charges', data = medical,palette="bright")
plt.title('Region vs Charges',fontweight="bold", size=20)
plt.subplot(2,3,5)
sns.boxplot(x = 'Age_range', y = 'charges', data = medical, palette= 'husl')
plt.title('Age vs Charges',fontweight="bold", size=20)
plt.show()
```



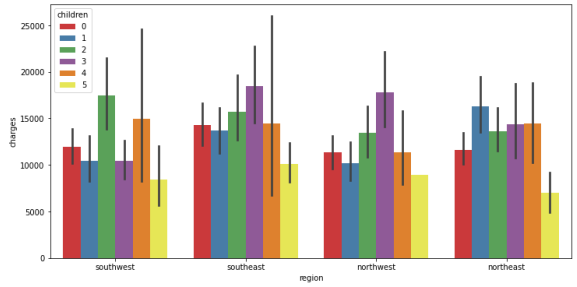
- 의료 비용은 비흡연자보다 흡연자가 더 많이 지출한다.
- 동남부 지역은 의료 비용이 더 비싸다.
- 노인들은 의료 비용을 더 많이 지출한다.

데이터 시각화

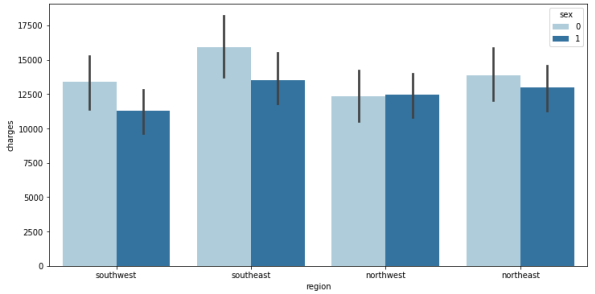
```
plt.figure(figsize=(12,6))
sns.barplot(x = 'region', y = 'charges',hue='smoker', data=medical, palette='cool')
plt.show()
```




```
plt.figure(figsize=(12,6))
sns.barplot(x='region', y='charges', hue='children', data=medical, palette='Set1')
plt.show()
```



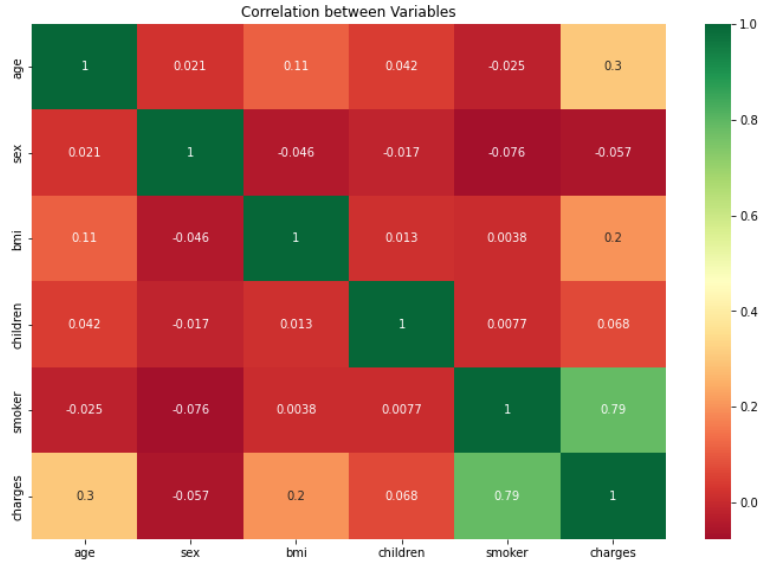

```
plt.figure(figsize=(12,6))
sns.barplot(x='region', y='charges', hue='sex', data=medical, palette='Paired')
plt.show()
```



- 흡연으로 인한 가장 많이 의료 비용을 지출하는 곳은 남동쪽에 있고, 가장 적게 지출하는 곳은 북동쪽에 있습니다.
- 지역상 흡연으로 인한 차이가 존재하지만 성별차이로 인한 의료 비용 차이가 이를 압도합니다.
- 아이가 있는 사람들이 전반적으로 의료 비용을 더 많이 지출하는 경향이 있습니다.

데이터 시각화

```
#Heatmap to see correlation between variables
plt.figure(figsize=(12, 8))
sns.heatmap(medical.corr(), cmap='RdYlGn', annot = True)
plt.title("Correlation between Variables")
plt.show()
```



데이터 칼럼들 간의 연관성

데이터 준비

```
# # Get the dummy variables for region and age range
region=pd.get_dummies(medical.region,drop_first=True)
Age_range=pd.get_dummies(medical.Age_range,drop_first=True)
children= pd.get_dummies(medical.children,drop_first=True,prefix='children')
```

```
# Add the results to the original bike dataframe
medical=pd.concat([region,Age_range,children,medical],axis=1)
medical.head()
```

	northwest	southeast	southwest	Senior Adult	Elder	children_1	children_2	children_3	children_4	children_5	age	sex	bmi	children	smoker	region	charges	Age_range
0	0	0	1	0	0	0	0	0	0	0	19	1	27.900	0	1	southwest	16884.92400	Young adult
1	0	1	0	0	0	1	0	0	0	0	18	0	33.770	1	0	southeast	1725.55230	Young adult
2	0	1	0	0	0	0	0	1	0	0	28	0	33.000	3	0	southeast	4449.46200	Young adult
3	1	0	0	0	0	0	0	0	0	0	33	0	22.705	0	0	northwest	21984.47061	Young adult
4	1	0	0	0	0	0	0	0	0	0	32	0	28.880	0	0	northwest	3866.85520	Young adult

```
#Drop region and age range as we are created a dummy
medical.drop(['region', 'Age_range', 'age','children'], axis = 1, inplace = True)
medical.head()
```

	northwest	southeast	southwest	Senior Adult	Elder	children_1	children_2	children_3	children_4	children_5	sex	bmi	smoker	charges
0	0	0	1	0	0	0	0	0	0	0	1	27.900	1	16884.92400
1	0	1	0	0	0	1	0	0	0	0	0	33.770	0	1725.55230
2	0	1	0	0	0	0	0	1	0	0	0	33.000	0	4449.46200
3	1	0	0	0	0	0	0	0	0	0	0	22.705	0	21984.47061
4	1	0	0	0	0	0	0	0	0	0	0	28.880	0	3866.85520

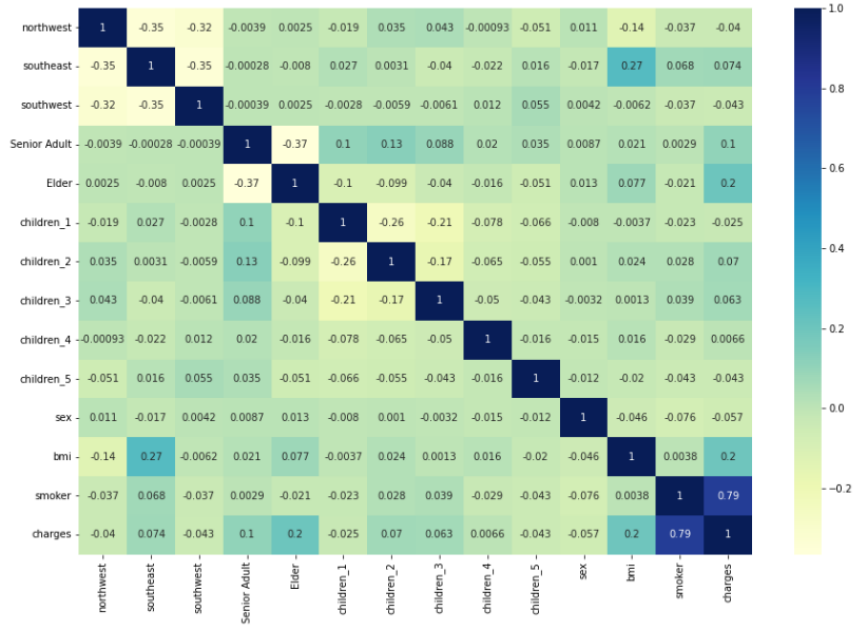
```
# Now lets see the number of rows and columns
medical.shape
```

```
(1338, 14)
```

- 지역과 나이 범위의 더미 변수를 가져온다.
- 결과를 원본 데이터 프레임에 추가한다.
- 더미 생성시, 지역과 나이 범위를 삭제한다.
- 준비한 데이터는 1338개의 행과 14개의 열로 구성되었다는 것을 알 수 있습니다.

데이터 준비

```
#Now lets check the correlation between variables again
#Heatmap to see correlation between variables
plt.figure(figsize=(15, 10))
sns.heatmap(medical.corr(), cmap='YlGnBu', annot = True)
plt.show()
```



- 변수 간 관계를 확인하기 위한 Heatmap

Training 데이터와 Testing 데이터로 분리

```
from sklearn.model_selection import train_test_split

# We specify this so that the train and test data set always have the same rows, respectively
#np.random.seed(0)
medical_train, medical_test = train_test_split(medical, train_size = 0.7, random_state = 100)
```

```
print(medical_train.shape)
print(medical_test.shape)
```

```
(936, 14)
(402, 14)
```

```
from sklearn.preprocessing import MinMaxScaler
```

```
#Instantiate an object
scaler = MinMaxScaler()

#Create a list of numeric variables
num_vars=['bmi','charges']

#Fit on data
medical_train[num_vars] = scaler.fit_transform(medical_train[num_vars])
medical_train.head()
```

	northwest	southeast	southwest	Senior Adult	Elder	children_1	children_2	children_3	children_4	children_5	sex	bmi	smoker	charges
966	1	0	0	1	0	0	1	0	0	0	0	0.237692	1	0.364661
522	0	0	0	1	0	0	0	0	0	0	1	0.483051	0	0.139579
155	1	0	0	1	0	0	0	0	0	0	0	0.633844	0	0.093008
671	0	0	0	0	0	0	0	0	0	0	1	0.408932	0	0.045040
1173	1	0	0	1	0	0	1	0	0	0	0	0.357815	0	0.085173

- Train 데이터와 test 데이터로 분리

```
#Divide the data into X and y
y_train = medical_train.pop('charges')
X_train = medical_train
```

선형 모델 제작

```
# Importing RFE and LinearRegression
from sklearn.feature_selection import RFE
from sklearn.linear_model import LinearRegression
```

```
# Running RFE with the output number of the variable equal to 15
lm = LinearRegression()
lm.fit(X_train, y_train)

rfe = RFE(lm, 8)             # running RFE
rfe = rfe.fit(X_train, y_train)
```

```
#List of variables selected
list(zip(X_train.columns,rfe.support_,rfe.ranking_))
```

```
[('northwest', False, 4),
 ('southeast', False, 3),
 ('southwest', False, 2),
 ('Senior Adult', True, 1),
 ('Elder', True, 1),
 ('children_1', False, 5),
 ('children_2', True, 1),
 ('children_3', True, 1),
 ('children_4', True, 1),
 ('children_5', True, 1),
 ('sex', False, 6),
 ('bmi', True, 1),
 ('smoker', True, 1)]
```

```
#Columns where RFE support is True
col = X_train.columns[rfe.support_]
col
```

```
Index(['Senior Adult', 'Elder', 'children_2', 'children_3', 'children_4',
       'children_5', 'bmi', 'smoker'],
      dtype='object')
```

```
#Columns where RFE support is False
X_train.columns[~rfe.support_]
```

```
Index(['northwest', 'southeast', 'southwest', 'children_1', 'sex'], dtype='object')
```

```
# Creating X_test dataframe with RFE selected variables
X_train_rfe = X_train[col]
```

```
# Adding a constant variable
import statsmodels.api as sm
X_train_rfe = sm.add_constant(X_train_rfe)
```

```
# Running the linear model
lm = sm.OLS(y_train,X_train_rfe).fit()
```

```
print(lm.summary())
```

```
                            OLS Regression Results
==============================================================================
Dep. Variable:                charges   R-squared:                       0.726
Model:                            OLS   Adj. R-squared:                  0.723
Method:                 Least Squares   F-statistic:                     306.4
Date:                Tue, 30 Nov 2021   Prob (F-statistic):          3.32e-254
Time:                        13:12:44   Log-Likelihood:                 813.97
No. Observations:                 936   AIC:                            -1610.
Df Residuals:                     927   BIC:                            -1566.
Df Model:                           8
Covariance Type:            nonrobust
================================================================================
                   coef    std err          t      P>|t|      [0.025      0.975]
--------------------------------------------------------------------------------
const           -0.0166      0.009     -1.747      0.081      -0.035       0.002
Senior Adult     0.0764      0.007     10.391      0.000       0.062       0.091
Elder            0.1506      0.010     15.266      0.000       0.131       0.170
children_2       0.0299      0.009      3.357      0.001       0.012       0.047
children_3       0.0241      0.011      2.275      0.023       0.003       0.045
children_4       0.0431      0.023      1.859      0.063      -0.002       0.089
children_5       0.0260      0.031      0.836      0.403      -0.035       0.087
bmi              0.1744      0.020      8.690      0.000       0.135       0.214
smoker           0.3826      0.008     45.327      0.000       0.366       0.399
==============================================================================
Omnibus:                      225.783   Durbin-Watson:                   2.054
Prob(Omnibus):                  0.000   Jarque-Bera (JB):              544.376
Skew:                           1.277   Prob(JB):                    6.17e-119
Kurtosis:                       5.726   Cond. No.                         11.4
==============================================================================

Notes:
[1] Standard Errors assume that the covariance matrix of the errors is correctly spec
```

- 재귀 특성 제거
- 세부 통계를 위한 통계 모델을 사용한 모델 제작

선형 모델 제작

$$VIF_i = \frac{1}{1-R_i^2}$$

```
#Drop the constant term B0
X_train_rfe = X_train_rfe.drop(['const'], axis=1)
```

```
# Calculate the VIFs for the new model
from statsmodels.stats.outliers_influence import variance_inflation_factor

vif = pd.DataFrame()
X = X_train_rfe
vif['Features'] = X.columns
vif['VIF'] = [variance_inflation_factor(X.values, i) for i in range(X.shape[1])]
vif['VIF'] = round(vif['VIF'], 2)
vif = vif.sort_values(by = "VIF", ascending = False)
vif
```

	Features	VIF
6	bmi	2.48
0	Senior Adult	1.84
1	Elder	1.32
2	children_2	1.29
7	smoker	1.22
3	children_3	1.17
4	children_4	1.03
5	children_5	1.02

- VIF, 분산 인플레이션 계수는 특성 변수들이 서로 얼마나 상관되어 있는지에 대한 기본적인 정량적 방안을 제공합니다. 이것은 제작한 선형 모델을 테스트하는 데 중요한 매개 변수입니다.

Train 데이터의 잔차 분석

```
X_train_lm3=sm.add_constant(X_train_lm3)
X_train_lm3.head()
```

	const	Senior Adult	Elder	children_2	bmi	smoker
966	1.0	1	0	1	0.237692	1
522	1.0	1	0	0	0.483051	0
155	1.0	1	0	0	0.633844	0
671	1.0	0	0	0	0.408932	0
1173	1.0	1	0	1	0.357815	0

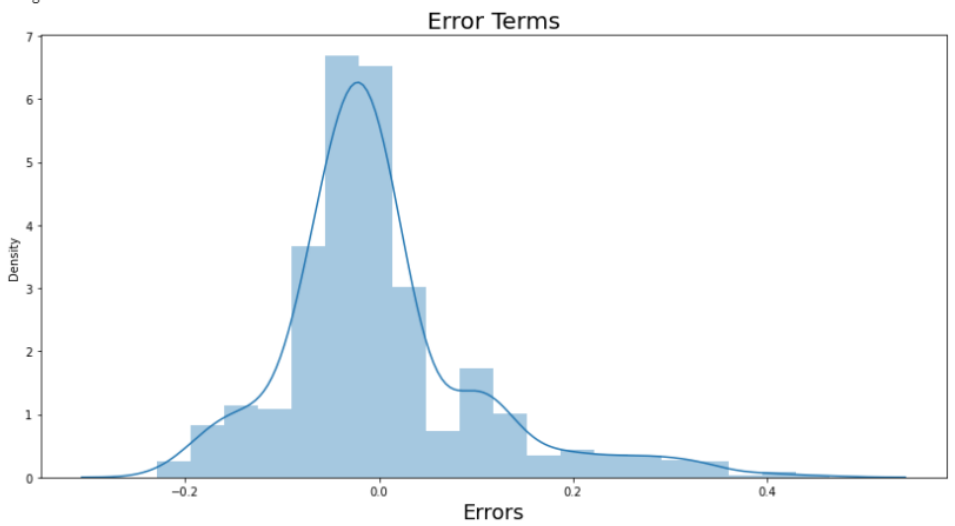
+ Code + Markdown

```
#y train predicted
y_train_pred = lm3.predict(X_train_lm3)
```

```
# Importing the required libraries for plots.
import matplotlib.pyplot as plt
import seaborn as sns
%matplotlib inline
```

```
# Plot the histogram of the error terms

fig = plt.figure()
plt.figure(figsize=(14,7))
sns.distplot((y_train - y_train_pred), bins = 20)
plt.title('Error Terms', fontsize = 20)                  # Plot heading
plt.xlabel('Errors', fontsize = 18)  # X-label
plt.show()
```

```
<Figure size 432x288 with 0 Axes>
```



- 오차항도 정규 분포를 따르는지 확인하기 위한 히스토그램

예측

```
# Now let's use our model to make predictions.

# Creating X_test_new dataframe by dropping variables from X_test
X_test_new = X_test[X_train_new3.columns]

# Adding a constant variable
X_test_new1 = sm.add_constant(X_test_new)
X_test_new1.head()
```

	const	Senior Adult	Elder	children_2	bmi	smoker
12	1.0	0	0	0	0.496099	0
306	1.0	0	0	1	0.310465	0
318	1.0	1	0	0	0.314366	0
815	1.0	0	0	0	0.417003	0
157	1.0	0	0	0	0.247915	1

+ Code + Markdown

```
# Making predictions
y_pred = lm3.predict(X_test_new1)
```

```
#Evaluate R-square for test
from sklearn.metrics import r2_score
r2_score(y_test,y_pred)
```

```
0.7628855670251863
```

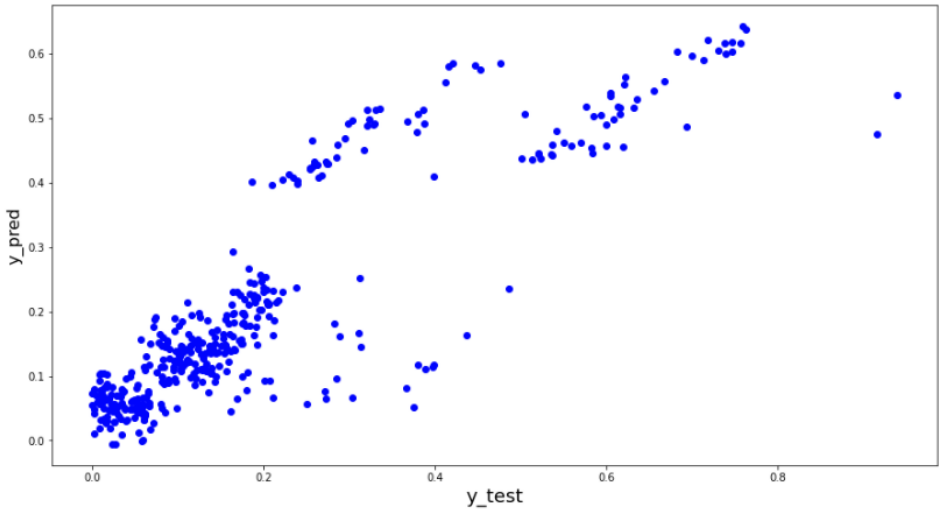
```
#Adjusted R^2
#adj r2=1-(1-R2)*(n-1)/(n-p-1)

#n =sample size , p = number of independent variables
n = X_test.shape[0]
p = X_test.shape[1]


Adj_r2=1-(1-0.75783003115855)*(n-1)/(n-p-1)
print(Adj_r2)
```

```
0.7497160889035529
```

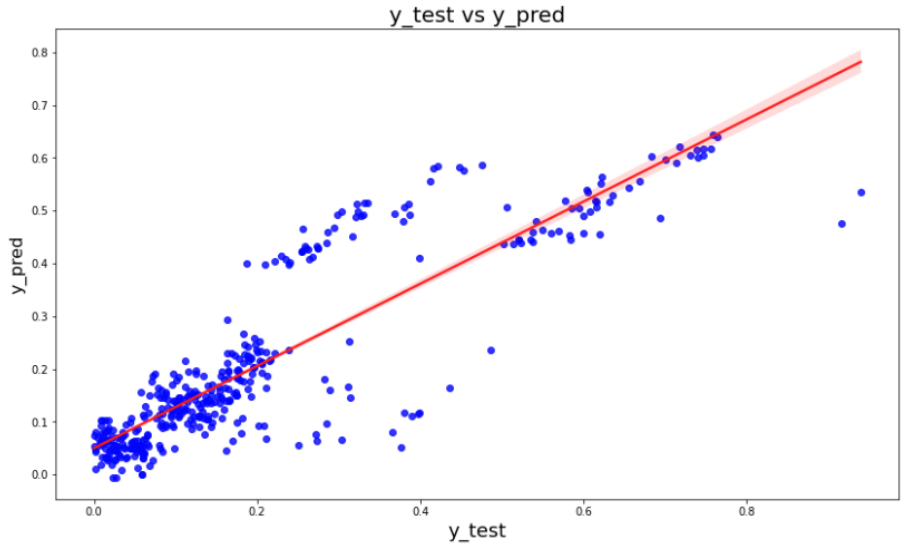
모델 평가

```
# Plotting y_test and y_pred to understand the spread.
fig = plt.figure()
plt.figure(figsize=(15,8))
plt.scatter(y_test,y_pred,color='blue')
fig.suptitle('y_test vs y_pred', fontsize=20)              # Plot heading
plt.xlabel('y_test', fontsize=18)                          # X-label
plt.ylabel('y_pred', fontsize=16)     # Y-label
plt.show()
```

```
<Figure size 432x288 with 0 Axes>
```




```
#Regression plot
plt.figure(figsize=(14,8))
sns.regplot(x=y_test, y=y_pred, ci=68, fit_reg=True,scatter_kws={"color": "blue"}, line_kws={"

plt.title('y_test vs y_pred', fontsize=20)              # Plot heading
plt.xlabel('y_test', fontsize=18)                          # X-label
plt.ylabel('y_pred', fontsize=16)                          # Y-label
plt.show()
```



Train 모델과 Test 간의 최종 결과 비교

- Train R^2 : 0.723
- Train Adjusted R^2 : 0.722
- Test R^2: 0.762
- Test Adjusted R^2: 0.749
- Difference in R^2 between train and test: 3.9%
- Difference in adjusted R^2 between Train and test: 2.7 % which is less than 5%

최적의 적합선의 방적싱은 다음과 같습니다.

$$charges = 0.3826 \times smoker + 0.077 \times Senioradult + 0.149 \times Elder + 0.176 \times bmi + 0.024 \times children2$$

결과

- 회귀 분석에서, 지역과 성별이 의료 비용에서 유의미한 차이를 가져오지 않는 것을 알 수 있다.
- 나이, BMI, 자녀의 수, 흡연 여부는 의료 비용을 움직이는 요인이다