예측애널리틱스 Homework #1

고려대학교
KOREA UNIVERSITY

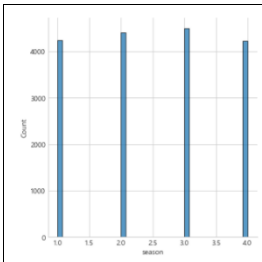
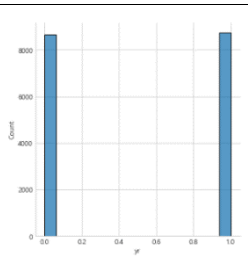
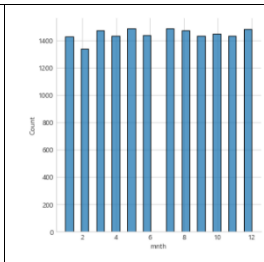
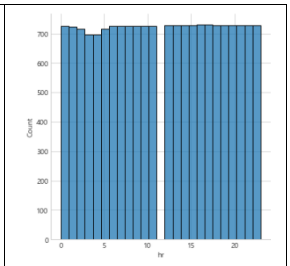
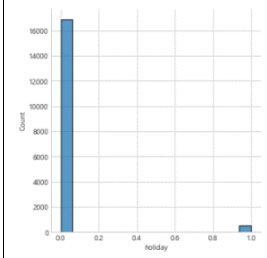
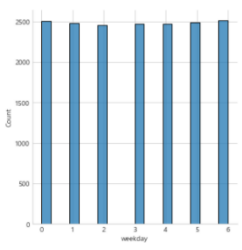
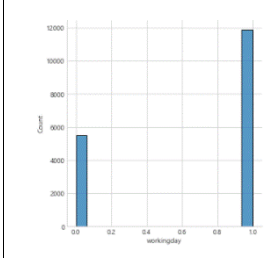
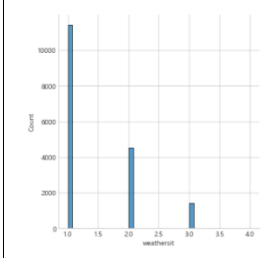
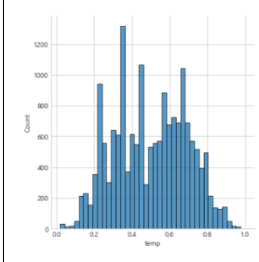
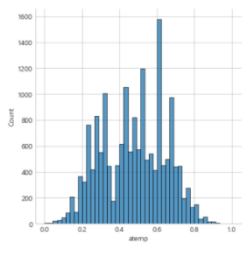
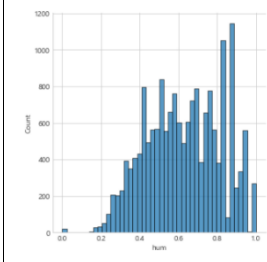
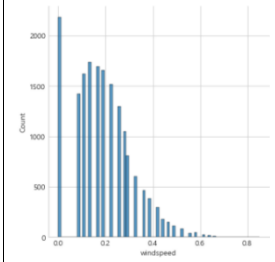
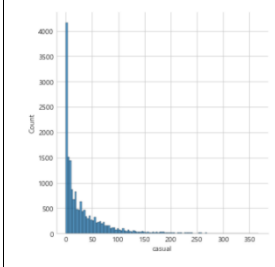
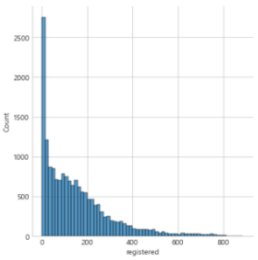
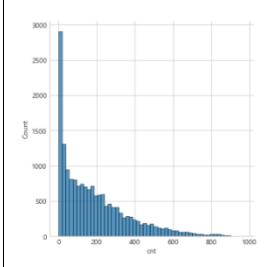
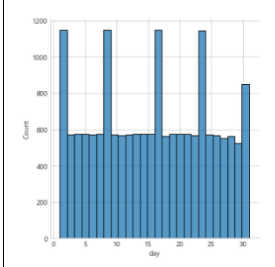
대학	고려대학교 공과대학
학과	산업경영공학부
학번	2017170819
이름	박상민

각종 분석을 통한 데이터 파악

Bike Sharing Data는 16개의 독립변수와 1개의 종속변수, 총 17,379개의 instance로 구성된 데이터이다. 파이썬 내에서 데이터 명칭은 "data"라 하였다. 변수에 대해 먼저 알아보면, 독립변수에는 index를 나타내는 ‘instant', 날짜를 나타내는 ’dteday', 봄~겨울(1~4)을 나타내는 ‘season', 연도(0 : 2011, 1 : 2012)를 나타내는 ’yr', 월을 나타내는 ‘mnth', 시간을 나타내는 ’hr', 공휴일인지 여부를 나타내는 ‘holiday', 요일을 나타내는 ’weekday', 주말 여부(0,1)를 나타내는 ‘workingday', 날씨에 따른 범주형(1~4)인 'weathersit', 도씨 온도를 나타내는 ’temp', 체감 도씨 온도를 나타내는 ‘atemp', 습도를 나타내는 ’hum', 풍속을 나타내는 ‘windspeed', 일반 사용자 수를 나타내는 ’casual', 등록된 사용자 수를 나타내는 ‘registered'가 있다. 종속변수는 ’cnt'로 'casual'과 ‘registered'를 합친 총 자전거 대여 수를 나타낸다. 독립변수 중 날짜를 나타내는 ’dteday'는 다른 독립변수인 ‘yr', 'mnth', 'hr'로 표현이 가능하므로 이를 제외한 ’일‘을 나타내는 'day' 열을 새로 추가하고 'dteday' 열은 삭제했다. 마지막으로, ‘casual'과 ’registered'의 합으로 종속 변수인 ‘cnt'가 도출되므로 이 두 독립변수는 데이터 형태 파악에서는 포함시켰으나 상관관계 분석 및 회귀 모델에서는 제외시켰다.

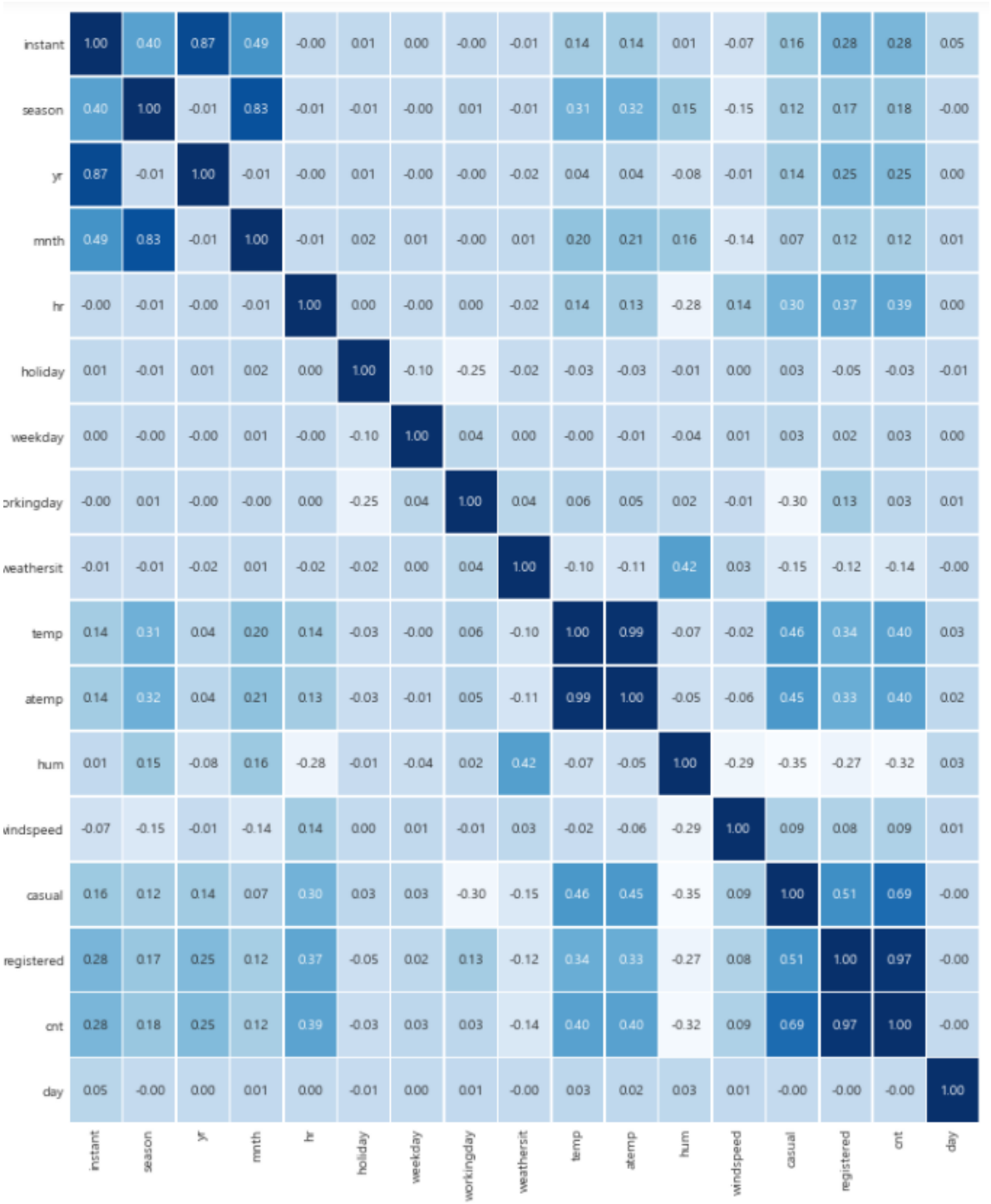
```
import pandas as pd
data = pd.read_csv('Bike Sharing Data.csv')
```

```
data['dteday']=data.dteday.apply(pd.to_datetime)
data['day']=data.dteday.apply(lambda X : X.day)
drop_columns=['dteday', 'instant', 'casual', 'registered']
data.drop(drop_columns, axis=1, inplace=True)
```

			
season	year	month	hour
			
holiday	weekday	workingday	weathersit
			
temp	atemp	hum	windspeed
			
casual	registered	cnt	day

데이터의 형태를 파악하기 위해 히스토그램을 그려본 결과, season, year, month, hour, weekday는 모두 균등한 분포를 보이는 범주형 변수임을 확인할 수 있었고 holiday, workingday, day, weathersit는 변수 특성상 비균등한 분포를 보이는 범주형 변수임을 알 수 있었다. 나머지 temp, atemp, hum, windspeed, casual, registered, cnt 변수는 수치형 변수로서 각각 데이터에 해당하는 분포를 확인할 수 있었다. 특히 종속변수인 cnt는

casual과 registered 변수의 합으로서 왼쪽에 몰린 데이터 형태를 띠고 있음을 알 수 있었다.



변수들 간 상관관계를 살펴볼 수 있는 상관관계 matrix를 살펴보면, temp와 atemp가 다중공산성이 크다는 것을 알 수 있다. 따라서 atemp를 제거해보았다.

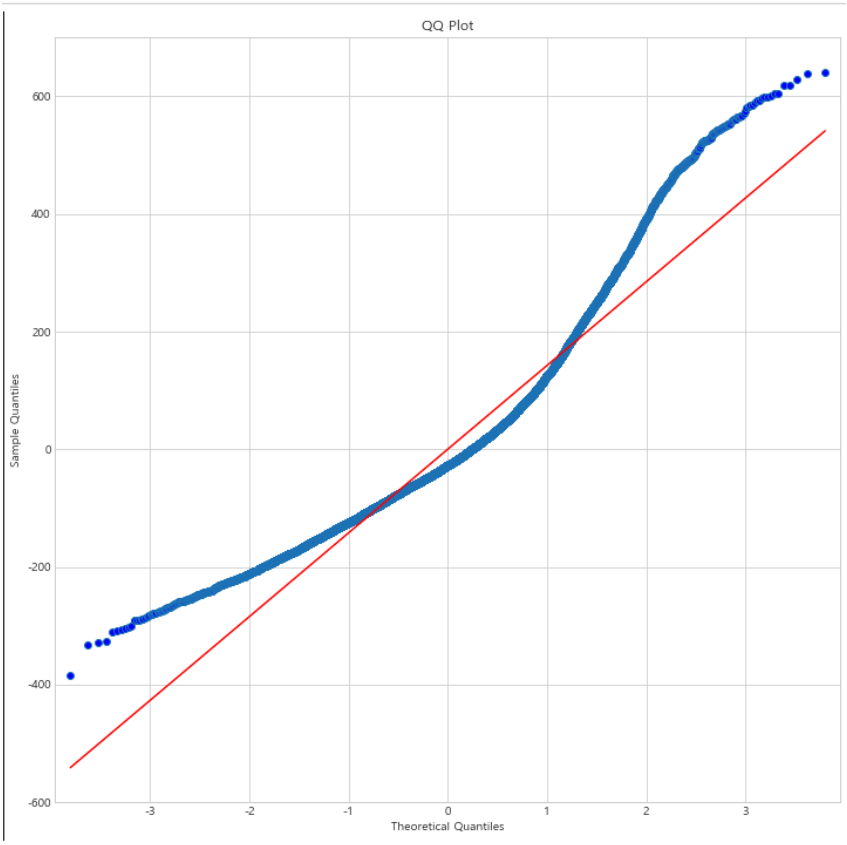
다중회귀모델 가정 충족 여부 확인

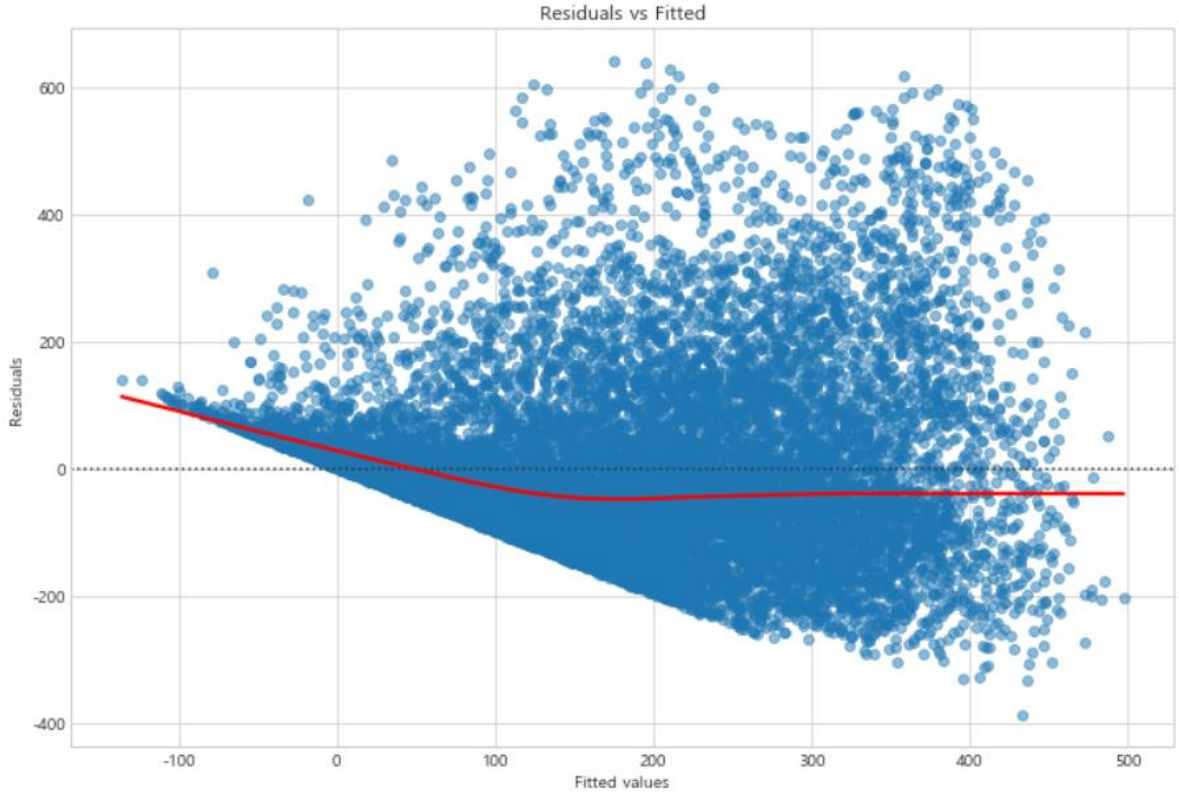
위에서 전처리한 데이터를 기반으로 종속변수 'cnt'를 target variable로 하는 다중회귀모델을 구축하였다. 이 때 패키지는 statsmodels의 OLS(Ordinary Least Square)를 사용하였다. 또한 차후 회귀모델의 예측 정확도를 알아보기 위해 sklearn의 train_test_split을 사용하여 테스트 데이터의 크기를 0.2로 설정하였다.

확률오차 가정 : $\varepsilon_i \sim$ 정규분포 $E(\varepsilon_i) = 0 \quad V(\varepsilon_i) = \sigma^2$ for all i.

$$\varepsilon_i \sim N(0, \sigma^2),\ i = 1, 2, \ldots, n$$

회귀모델이 성립되려면 오차항의 정규성 및 등분산성 가정이 성립되어야 한다. 먼저 정규성을 확인해보면, QQplot이 거의 직선을 따르므로 정규성을 만족한다고 할 수 있다.





Residual의 등분산성을 확인하기 위한 작업도 진행한 결과, fitted value가 커질수록 넓어지는 깔때기 모양을 확인할 수 있다. 이는 등분산성을 만족시키지 않을 가능성이 있기 때문에, data transformation을 고려해 볼 필요가 있다.

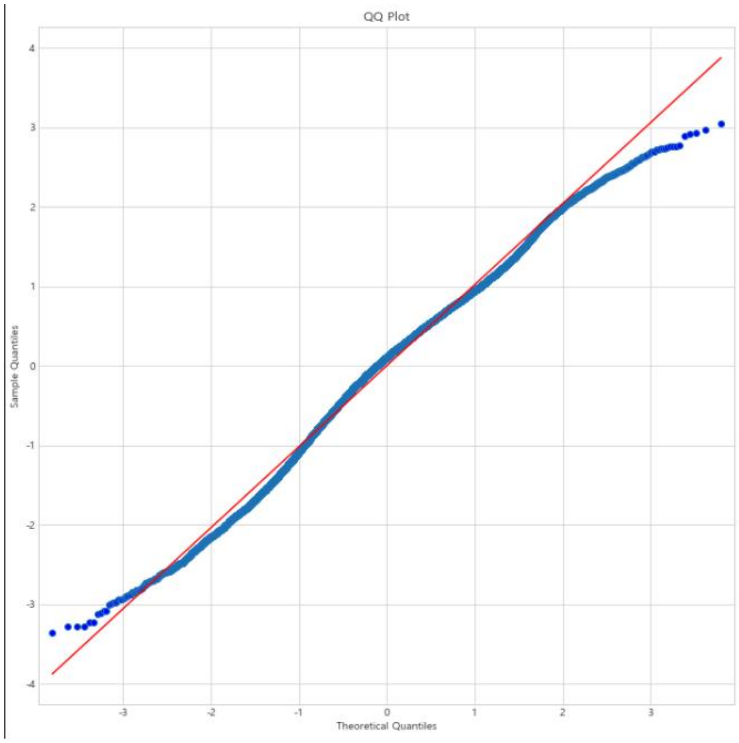
OLS Regression Results

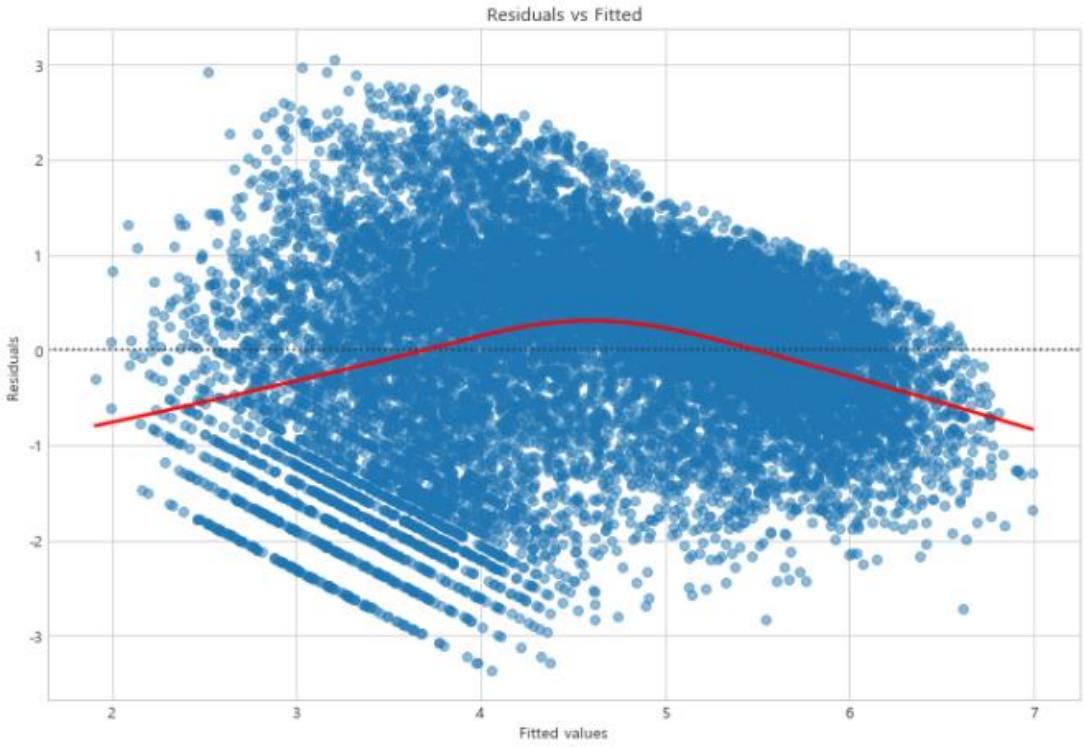
Dep. Variable:	cnt	R-squared:	0.387
Model:	OLS	Adj. R-squared:	0.386
Method:	Least Squares	F-statistic:	730.0
Date:	Sun, 20 Mar 2022	Prob (F-statistic):	0.00
Time:	00:14:54	Log-Likelihood:	-88682.
No. Observations:	13903	AIC:	1.774e+05
Df Residuals:	13890	BIC:	1.775e+05
Df Model:	12		
Covariance Type:	nonrobust		

	coef	std err	t	P>|t|	[0.025	0.975]
const	-10.1454	7.876	-1.288	0.198	-25.584	5.294
season	21.3566	2.033	10.504	0.000	17.371	25.342
yr	81.2583	2.434	33.390	0.000	76.488	86.028
mnth	-0.3593	0.633	-0.568	0.570	-1.600	0.881
hr	7.5537	0.185	40.826	0.000	7.191	7.916
holiday	-22.1627	7.482	-2.962	0.003	-36.828	-7.497
weekday	1.7361	0.606	2.865	0.004	0.548	2.924
workingday	3.5161	2.692	1.306	0.192	-1.761	8.793
weathersit	-4.1489	2.133	-1.945	0.052	-8.330	0.032
temp	283.2965	6.783	41.768	0.000	270.002	296.591
hum	-198.8977	7.729	-25.734	0.000	-214.048	-183.748
windspeed	30.1772	10.572	2.854	0.004	9.454	50.900
day	-0.1391	0.138	-1.010	0.312	-0.409	0.131

Omnibus:	2740.389	Durbin-Watson:	2.014
Prob(Omnibus):	0.000	Jarque-Bera (JB):	5346.272
Skew:	1.199	Prob(JB):	0.00
Kurtosis:	4.865	Cond. No.	225.

통계분석 자료를 분석해봐도 R square 값과 Adjusted R square 값이 각각 0.387, 0.386으로 너무 낮기 때문에 회귀 모델의 수정이 필요하다. R square 값은 단순히 Y의 평균값을 사용했을 때 대비 X 정보를 사용함으로써 얻는 성능 향상 정도이다. X 정보를 사용했을 때 성능향성 정도가 38% 정도이면 낮은 편에 속한다.






```
y_log = np.log1p(y)
X_train, X_test, y_train, y_test = train_test_split(X, y_log, test_size = 0.2, random_state = 2021)
X_train = sm.add_constant(X_train)
model = sm.OLS(y_train, X_train, axis = 1)
model_trained = model.fit()
```

Log Transformation으로 data를 transform한 결과, residual이 정규성을 만족하면서,

residual scatter plot에서 비교적 일정한 분산을 가지도록 분포되어 있는 것을 알 수 있다. 따라서 회귀분석 가정을 만족한다고 할 수 있다.

OLS Regression Results			
Dep. Variable:	cnt	R-squared:	0.484
Model:	OLS	Adj. R-squared:	0.483
Method:	Least Squares	F-statistic:	1084.
Date:	Sun, 20 Mar 2022	Prob (F-statistic):	0.00
Time:	00:45:33	Log-Likelihood:	-20003.
No. Observations:	13903	AIC:	4.003e+04
Df Residuals:	13890	BIC:	4.013e+04
Df Model:	12		
Covariance Type:	nonrobust		

	coef	std err	t	P>|t|	[0.025	0.975]
const	2.6756	0.056	47.473	0.000	2.565	2.786
season	0.1558	0.015	10.705	0.000	0.127	0.184
yr	0.4160	0.017	23.891	0.000	0.382	0.450
mnth	-0.0009	0.005	-0.191	0.848	-0.010	0.008
hr	0.0965	0.001	72.909	0.000	0.094	0.099
holiday	-0.1992	0.054	-3.721	0.000	-0.304	-0.094
weekday	0.0191	0.004	4.397	0.000	0.011	0.028
workingday	-0.0485	0.019	-2.519	0.012	-0.086	-0.011
weathersit	-0.0027	0.015	-0.180	0.857	-0.033	0.027
temp	2.0018	0.049	41.246	0.000	1.907	2.097
hum	-1.3568	0.055	-24.534	0.000	-1.465	-1.248
windspeed	0.2564	0.076	3.389	0.001	0.108	0.405
day	-0.0011	0.001	-1.135	0.256	-0.003	0.001

Omnibus:	159.668	Durbin-Watson:	2.012
Prob(Omnibus):	0.000	Jarque-Bera (JB):	164.008
Skew:	-0.261	Prob(JB):	2.43e-36
Kurtosis:	2.895	Cond. No.	225.

R square 값과 Adjusted R square 값도 각각 0.484, 0.483으로 증가한 것을 볼 수 있다. X 변수의 Y에 대한 설명력이 증가했다고 분석할 수 있다. Log transformation이 회귀분석에 긍정적으로 작용했음을 의미한다.

각 회귀계수에 대한 기울기=0 여부 검정

$$H_0: \beta_1 = 0 \text{ vs. } H_1: \beta_1 \neq 0$$

$$t^* = \frac{\hat{\beta}_1 - 0}{sd\{\hat{\beta}_1\}}$$

$$\text{If } |t^*| > t_{\alpha/2,n-2}, \text{ we reject } H_0$$

$$\text{P-value} = 2 \cdot P(T > |t^*|) \text{ where } T \sim t(n-2)$$

그리고 각 변수의 coefficient들의 유의성, 즉 각 회귀계수에 대한 기울기=0 여부를 살펴볼 수 있다. 기울기=0 여부를 살펴보는 방법을 각 회귀계수의 t-검정을 통해 확인할 수 있다. 귀무가설을 β=0 으로 놓고, 대립가설을 $\beta \neq 0$ 으로 놓은 후 t-검정을 하면 p-value 값이 0.05를 초과하는 회귀계수들이 유의미한 변수가 아니라고 할 수 있다. mnth, weathersit, day의 p-value가 0.05를 초과하기 때문에 유의미한 변수가 아니라고 결론지었다. 이 변수들을 제거한 후 다시 summary를 도출해보면

OLS Regression Results

Dep. Variable:	cnt	R-squared:	0.484
Model:	OLS	Adj. R-squared:	0.483
Method:	Least Squares	F-statistic:	1445.
Date:	Sun, 20 Mar 2022	Prob (F-statistic):	0.00
Time:	00:46:18	Log-Likelihood:	-20004.
No. Observations:	13903	AIC:	4.003e+04
Df Residuals:	13893	BIC:	4.010e+04
Df Model:	9		
Covariance Type:	nonrobust		

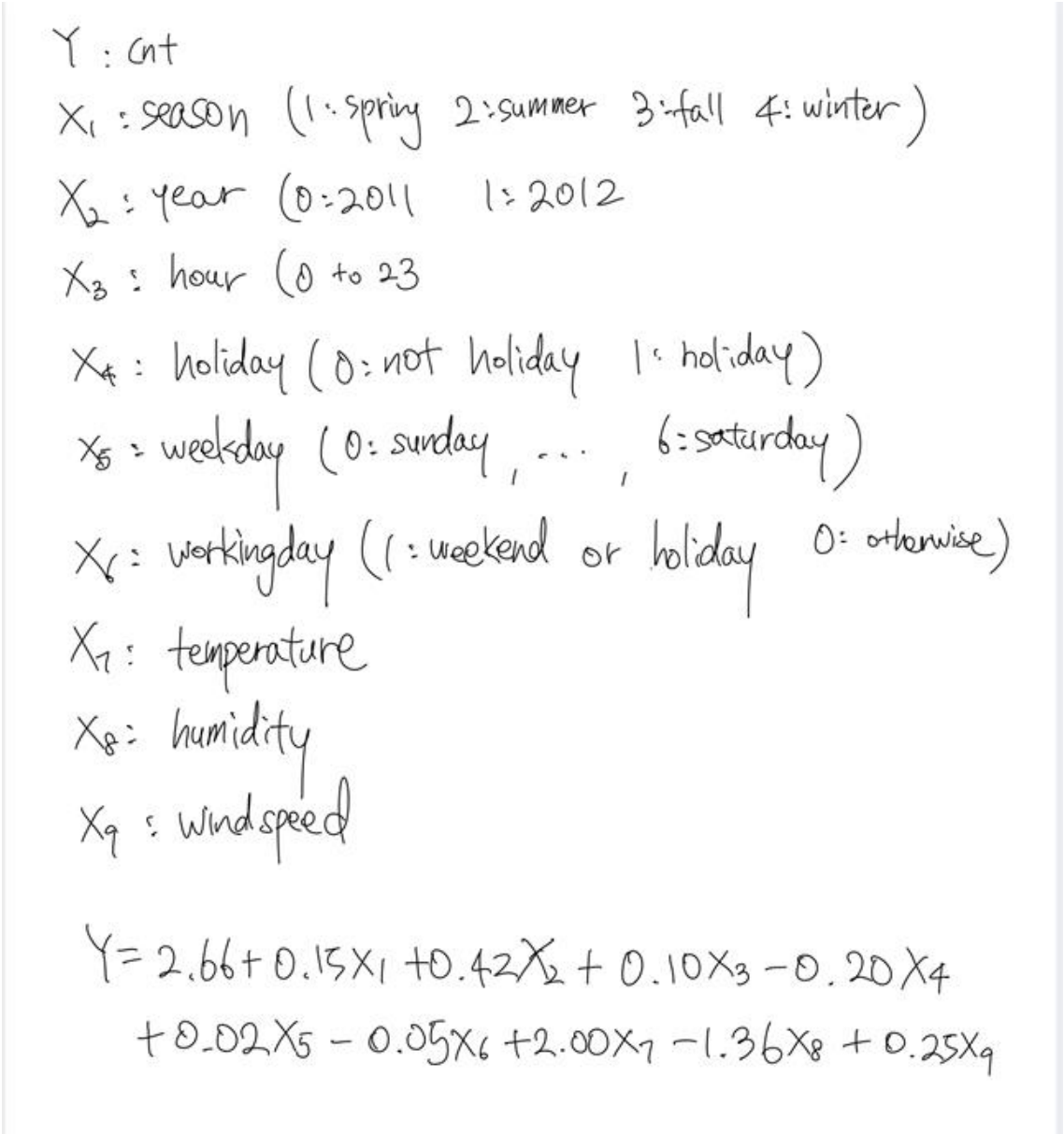
	coef	std err	t	P>|t|	[0.025	0.975]
const	2.6599	0.055	48.686	0.000	2.553	2.767
season	0.1537	0.008	18.210	0.000	0.137	0.170
yr	0.4158	0.017	23.889	0.000	0.382	0.450
hr	0.0965	0.001	73.345	0.000	0.094	0.099
holiday	-0.1991	0.053	-3.722	0.000	-0.304	-0.094
weekday	0.0190	0.004	4.382	0.000	0.010	0.027
workingday	-0.0487	0.019	-2.531	0.011	-0.086	-0.011
temp	2.0014	0.048	41.577	0.000	1.907	2.096
hum	-1.3639	0.049	-27.707	0.000	-1.460	-1.267
windspeed	0.2528	0.075	3.384	0.001	0.106	0.399

Omnibus:	158.718	Durbin-Watson:	2.012
Prob(Omnibus):	0.000	Jarque-Bera (JB):	163.030
Skew:	-0.260	Prob(JB):	3.97e-36
Kurtosis:	2.896	Cond. No.	137.

R square와 Adjusted R square는 변수가 줄어도 그대로이기 때문에 효율적인 변수 삭제라고 할 수 있다.

다중회귀모델 구축 및 해석

회귀식을 써보면



이와 같이 다중회귀모델의 식을 세울 수 있다. 즉 cnt가 영향을 받는 독립변수는 총 9개이며, R square값이 0.484이기 때문에 단순히 Y의 평균값을 사용했을 때 대비 X 정볼를 사용함으로써 얻는 성능향상 정도가 48.4%라는 뜻이다. Adjusted R square는 독립

변수의 수가 증가함에 따라 결정계수도 커지는 단점을 보완하기 위해 회귀변동과 오차변동의 자유도를 고려한 측정값이다. R square 값과 크게 차이가 나지 않는다는 것은, 종속변수를 설명하는데 필요 없는 독립변수가 회귀모델에 포함되어 있지 않다는 뜻이다. 따라서 효율적인 회귀모델을 구축했다고 할 수 있다.

$$Adj\,R^2 = 1 - (1 - R^2)\frac{n-1}{n-p-1}$$

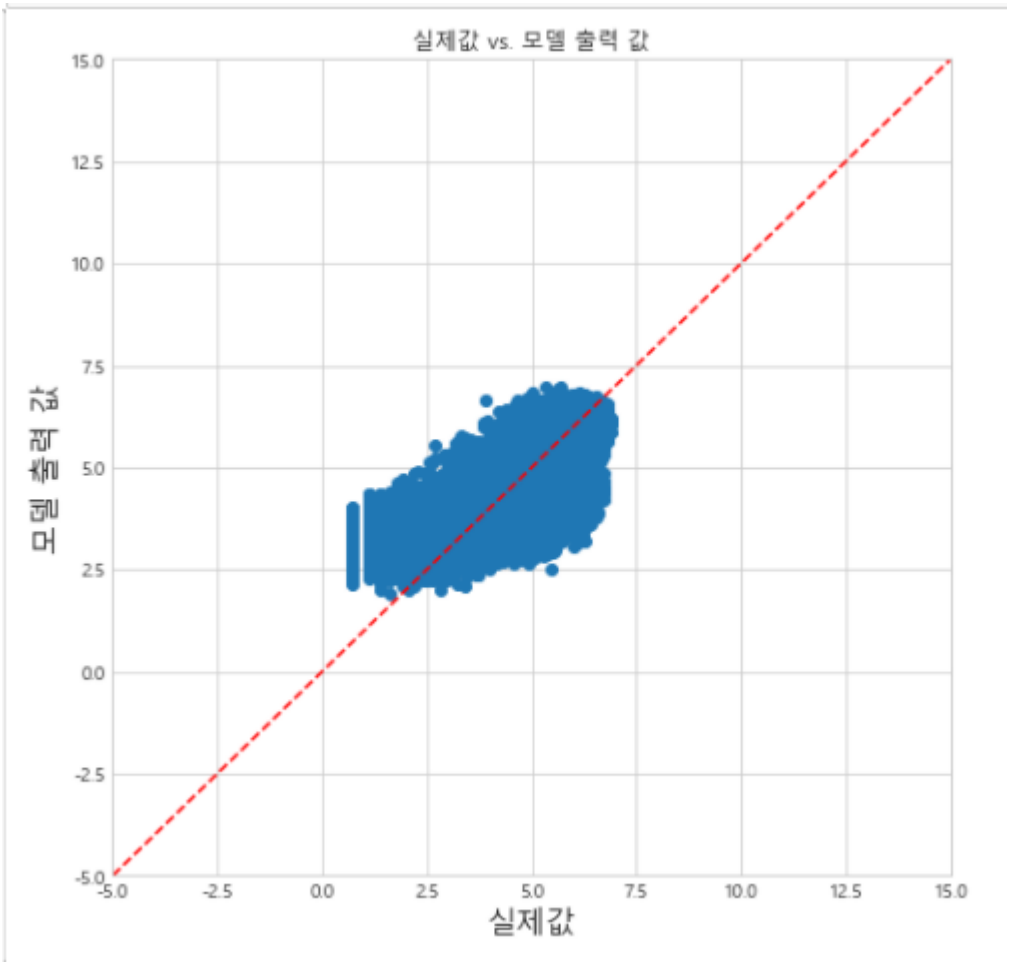
각 회귀계수에 대한 95% 신뢰구간 추정

	[0.025	0.975]
Constant	2.553	2.767
Season	0.137	0.170
Year	0.382	0.450
Hour	0.094	0.099
Holiday	-0.304	-0.094
Weekday	0.010	0.027
Workingday	-0.086	-0.011
Temperature	1.907	2.096
Humidity	-1.460	-1.267
windspeed	0.106	0.399

각 회귀계수에 대한 95% 신뢰구간 추정은 위의 표에 나와 있듯이, 0.025부터 0.975까지가 95% 신뢰구간에 해당한다.

데이터를 Training set 과 Testing set 으로 나누고 Testing set 의 예측정확도 계산

실제값과 모델 출력 값의 일치 정도를 확인하기 위한 작업을 진행하였다.



거의 대부분의 점들이 y=x 선 위에 있는 경향을 보여주므로 모델 출력값이 실제값을 잘 반영했다고 할 수 있다.

위의 회귀모델은 Training set을 이용하여 구성한 회귀 모델이다. 따라서 이전에 나눈 test set의 데이터를 이용하여 이 회귀모델의 예측 정확도를 계산해 보았고 training set의 결과와 비교해 보았다.

평균 제곱 오차 제곱근 평균 제곱 오차

$$MSE = \frac{\sum(y-\hat{y})^2}{n} \qquad RMSE = \sqrt{\frac{\sum(y-\hat{y})^2}{n}}$$

평균 절대 오차 평균 절대 백분율 오차

$$MAE = \frac{\sum|y-\hat{y}|}{n} \qquad MAPE = \frac{\sum|\frac{y-\hat{y}}{y}|}{n} * 100\%$$

결정계수

$$R^2 = 1 - \frac{SS_{RES}}{SS_{TOT}} = 1 - \frac{\sum_i(y_i-\hat{y}_i)^2}{\sum_i(y_i-\overline{y})^2}$$

Training set으로 회귀모델을 구축했기 때문에, testing set으로 예측을 하는 과정이 남아있다. Training set과 testing set으로 분리시켰던 testing set을 회귀 모델에 적용해 예측해보았고, 예측정확도를 계산해보았다. 예측정확도는 위와 같이 MSE, RMSE, MAE, MAPE, R square와 같은 지표를 통해 계산할 수 있다.

예측 결과

	Training set
MSE	1.04
RMSE	1.02
MAE	0.81
MAPE	26.46
R square	0.48

	Testing set
MSE	1.02
RMSE	1.01
MAE	0.79
MAPE	26.11
R square	0.49

```
X_test = sm.add_constant(X_test)
```

```
y_test_pred = model_trained.predict(X_test.drop(['mnth', 'weathersit', 'day'], axis=1))
y_test_pred.head()
```

```
8619     4.736068
11157    3.919427
2146     3.821214
6367     5.358904
12468    6.244082
dtype: float64
```

```
print(mean_squared_error(y_test, y_test_pred))
```

```
1.0207755414829311
```

```
print(np.sqrt(mean_squared_error(y_test, y_test_pred)))
```

```
1.0103343711281583
```

```
print(mean_absolute_error(y_test, y_test_pred))
```

```
0.7929871671519837
```

```
def mean_absolute_percentage_error(y_true, y_pred):
    y_true, y_pred = np.array(y_true), np.array(y_pred)
    return np.mean(np.abs((y_true-y_pred)/y_true))*100

print(mean_absolute_percentage_error(y_test, y_test_pred))
```

```
26.11448024058427
```

```
print(r2_score(y_test, y_test_pred))
```

```
0.48767176272779356
```

```
print('Training MSE: {:.3f}'.format(mean_squared_error(y_train, y_train_pred)))
print('Training RMSE: {:.3f}'.format(np.sqrt(mean_squared_error(y_train, y_train_pred))))
print('Training MAE: {:.3f}'.format(mean_absolute_error(y_train, y_train_pred)))
print('Training MAPE: {:.3f}'.format(mean_absolute_percentage_error(y_train, y_train_pred)))
print('Training R2: {:.3f}'.format(r2_score(y_train, y_train_pred)))
```

```
Training MSE: 1.041
Training RMSE: 1.020
Training MAE: 0.811
Training MAPE: 26.460
Training R2: 0.484
```

```
print('Testing MSE: {:.3f}'.format(mean_squared_error(y_test, y_test_pred)))
print('Testing RMSE: {:.3f}'.format(np.sqrt(mean_squared_error(y_test, y_test_pred))))
print('Testing MAE: {:.3f}'.format(mean_absolute_error(y_test, y_test_pred)))
print('Testing MAPE: {:.3f}'.format(mean_absolute_percentage_error(y_test, y_test_pred)))
print('Testing R2: {:.3f}'.format(r2_score(y_test, y_test_pred)))
```

```
Testing MSE: 1.021
Testing RMSE: 1.010
Testing MAE: 0.793
Testing MAPE: 26.114
Testing R2: 0.488
```

임의의 데이터 5개를 예측해보았을 때 정상적으로 예측값이 도출되는 것을 확인할 수 있다. Training set과 testing set의 예측정확도를 비교해본 결과, 거의 유사한 수치를 나타내고 있음을 확인할 수 있다. 따라서 예측모형이 비교적 정확하게 구현되었음을 알 수 있다.

Gauss-Markov Theorem

정의: least square estimator가 best linear unbiased estimator이다.

best linear unbiased estimator이려면 일단 parameter들이 불편추정량이어야 하고, 분산이 가장 작아야 한다.

증명: $Y = X\beta + \varepsilon$ 라는 회귀식을 가정. $E(\varepsilon)=0$, $V(\varepsilon)=\sigma^2$

$\hat{\beta} = (X^TX)^{-1}X^TY$ $\quad E(\hat{\beta})=\beta$, $V(\hat{\beta})=\sigma^2(X^TX)^{-1}$

β에 또 다른 불편추정량을 $\tilde{\beta}$라고 하면

$$\tilde{\beta} = (X^TX)^{-1}X^TY + DY = \{(X^TX)^{-1}X^T + D\}Y = CY$$

D는 $p \times n$ matrix이며 non-zero, positive semi-definite.

$$E(\tilde{\beta}) = (X^TX)^{-1}X^T(X\beta+\varepsilon) + D(X\beta+\varepsilon) = (X^TX)^{-1}X^TX\beta$$

$$+ (X^TX)^{-1}X^T\varepsilon + DX\beta + D\varepsilon = \beta + (X^TX)^{-1}X^T\varepsilon + DX\beta + D\varepsilon$$

$$E(\tilde{\beta}) = E(\beta) + (X^TX)^{-1}X^TE(\varepsilon) + E(DX\beta) + DE(\varepsilon)$$

($E(\varepsilon)=0$, $E(DX\beta)=0$)

$= \beta + DX\beta = \beta$ 이어야 한다. $\therefore DX\beta = 0$, $DX = 0$

$$V(\tilde{\beta}) = V(CY) = CV(Y)C^T = \sigma^2 C \cdot C^T$$

$$= \sigma^2 \left[\{ (X^TX)^{-1}X^T + D \} \{ X(X^TX)^{-1} + D^T \} \right]$$

$$= \sigma^2 \left[(X^TX)^{-1}X^TX(X^TX)^{-1} + (X^TX)^{-1}\underbrace{X^TD^T}_{①} + \underbrace{DX}_{=0} (X^TX)^{-1} + DD^T \right]$$

$$= \sigma^2 \left[(X^TX)^{-1} + 0 + 0 + DD^T \right] = \sigma^2 (X^TX)^{-1} + \sigma^2 D \cdot D^T$$

$$= V(\hat{\beta}) + \sigma^2 DD^T \qquad \therefore V(\hat{\beta}) \le V(\tilde{\beta})$$

① $X^TD^T = (DX)^T = 0$

따라서 $\hat{\beta}$ 는 best linear unbiased estimator 이다.