BikeSharing LR

September 2, 2024

[2]:
```python
# Demand for shared bikes
# Before covid data shared and we want to tell the business
# what could be the features that will help them boost their revenue after␣
↪lockdown is removed



import numpy as np , pandas as pd
import matplotlib.pyplot as plt, seaborn as sns
```

0.0.1 Overview

0.0.2 Problem Statement

To build a multiple linear regression model for the prediction of demand for shared bikes.

A bike-sharing system is a service in which bikes are made available for shared use to individuals on a short term basis for a price or free. Many bike share systems allow people to borrow a bike from a "dock" which is usually computer-controlled wherein the user enters the payment information, and the system unlocks it. This bike can then be returned to another dock belonging to the same system.

A US bike-sharing provider BoomBikes has recently suffered considerable dips in their revenues due to the ongoing Corona pandemic. The company is finding it very difficult to sustain in the current market scenario. So, it has decided to come up with a mindful business plan to be able to accelerate its revenue as soon as the ongoing lockdown comes to an end, and the economy restores to a healthy state.

In such an attempt, BoomBikes aspires to understand the demand for shared bikes among the people after this ongoing quarantine situation ends across the nation due to Covid-19. They have planned this to prepare themselves to cater to the people's needs once the situation gets better all around and stand out from other service providers and make huge profits.

They have contracted a consulting company to understand the factors on which the demand for these shared bikes depends. Specifically, they want to understand the factors affecting the demand for these shared bikes in the American market. The company wants to know:

1. Which variables are significant in predicting the demand for shared bikes.
2. How well those variables describe the bike demands Based on various meteorological surveys and people's styles, the service provider firm has gathered a large dataset on daily bike demands across the American market based on some factors.

Business Goal: You are required to model the demand for shared bikes with the available independent variables. It will be used by the management to understand how exactly the demands vary with different features. They can accordingly manipulate the business strategy to meet the demand levels and meet the customer's expectations. Further, the model will be a good way for management to understand the demand dynamics of a new market.

== Dataset characteristics ==

day.csv have the following fields:

```
- instant: record index
- dteday : date
- season : season (1:spring, 2:summer, 3:fall, 4:winter)
- yr : year (0: 2018, 1:2019)
- mnth : month ( 1 to 12)
- holiday : weather day is a holiday or not (extracted from http://dchr.dc.gov/page/holiday-sch
- weekday : day of the week
- workingday : if day is neither weekend nor holiday is 1, otherwise is 0.
+ weathersit :
    - 1: Clear, Few clouds, Partly cloudy, Partly cloudy
    - 2: Mist + Cloudy, Mist + Broken clouds, Mist + Few clouds, Mist
    - 3: Light Snow, Light Rain + Thunderstorm + Scattered clouds, Light Rain + Scattered cloud
    - 4: Heavy Rain + Ice Pallets + Thunderstorm + Mist, Snow + Fog
- temp : temperature in Celsius
- atemp: feeling temperature in Celsius
- hum: humidity
- windspeed: wind speed
- casual: count of casual users
- registered: count of registered users
- cnt: count of total rental bikes including both casual and registered
```

[4]:
```python
data = pd.read_csv("day.csv")
data.head()
```

[4]:
```
   instant     dteday  season  yr  mnth  holiday  weekday  workingday  \
0        1  01-01-2018       1   0     1        0        1           1
1        2  02-01-2018       1   0     1        0        2           1
2        3  03-01-2018       1   0     1        0        3           1
3        4  04-01-2018       1   0     1        0        4           1
4        5  05-01-2018       1   0     1        0        5           1

   weathersit       temp     atemp      hum  windspeed  casual  registered  \
0           2  14.110847  18.18125  80.5833  10.749882     331         654
1           2  14.902598  17.68695  69.6087  16.652113     131         670
2           1   8.050924   9.47025  43.7273  16.636703     120        1229
3           1   8.200000  10.60610  59.0435  10.739832     108        1454
4           1   9.305237  11.46350  43.6957  12.522300      82        1518
```

```
    cnt
0   985
1   801
2  1349
3  1562
4  1600
```

[5]:
```
data.tail()
```

[5]:
```
     instant      dteday  season  yr  mnth  holiday  weekday  workingday  \
725      726  27-12-2019       1   1    12        0        5           1
726      727  28-12-2019       1   1    12        0        6           0
727      728  29-12-2019       1   1    12        0        0           0
728      729  30-12-2019       1   1    12        0        1           1
729      730  31-12-2019       1   1    12        0        2           1

     weathersit       temp     atemp      hum  windspeed  casual  registered  \
725           2  10.420847  11.33210  65.2917  23.458911     247        1867
726           2  10.386653  12.75230  59.0000  10.416557     644        2451
727           2  10.386653  12.12000  75.2917   8.333661     159        1182
728           1  10.489153  11.58500  48.3333  23.500518     364        1432
729           2   8.849153  11.17435  57.7500  10.374682     439        2290

      cnt
725  2114
726  3095
727  1341
728  1796
729  2729
```

[6]:
```
data.shape
```

[6]:
```
(730, 16)
```

[7]:
```
data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 730 entries, 0 to 729
Data columns (total 16 columns):
 #   Column      Non-Null Count  Dtype
---  ------      --------------  -----
 0   instant     730 non-null    int64
 1   dteday      730 non-null    object
 2   season      730 non-null    int64
 3   yr          730 non-null    int64
 4   mnth        730 non-null    int64
 5   holiday     730 non-null    int64
```

```
 6   weekday     730 non-null    int64
 7   workingday  730 non-null    int64
 8   weathersit  730 non-null    int64
 9   temp        730 non-null    float64
 10  atemp       730 non-null    float64
 11  hum         730 non-null    float64
 12  windspeed   730 non-null    float64
 13  casual      730 non-null    int64
 14  registered  730 non-null    int64
 15  cnt         730 non-null    int64
dtypes: float64(4), int64(11), object(1)
memory usage: 91.4+ KB
```

[8]:

```
data.describe().T
```

[8]:

```
            count         mean          std        min          25%  \
instant     730.0   365.500000   210.877136   1.000000   183.250000
season      730.0     2.498630     1.110184   1.000000     2.000000
yr          730.0     0.500000     0.500343   0.000000     0.000000
mnth        730.0     6.526027     3.450215   1.000000     4.000000
holiday     730.0     0.028767     0.167266   0.000000     0.000000
weekday     730.0     2.995890     2.000339   0.000000     1.000000
workingday  730.0     0.690411     0.462641   0.000000     0.000000
weathersit  730.0     1.394521     0.544807   1.000000     1.000000
temp        730.0    20.319259     7.506729   2.424346    13.811885
atemp       730.0    23.726322     8.150308   3.953480    16.889713
hum         730.0    62.765175    14.237589   0.000000    52.000000
windspeed   730.0    12.763620     5.195841   1.500244     9.041650
casual      730.0   849.249315   686.479875   2.000000   316.250000
registered  730.0  3658.757534  1559.758728  20.000000  2502.250000
cnt         730.0  4508.006849  1936.011647  22.000000  3169.750000

                    50%          75%          max
instant      365.500000   547.750000   730.000000
season         3.000000     3.000000     4.000000
yr             0.500000     1.000000     1.000000
mnth           7.000000    10.000000    12.000000
holiday        0.000000     0.000000     1.000000
weekday        3.000000     5.000000     6.000000
workingday     1.000000     1.000000     1.000000
weathersit     1.000000     2.000000     3.000000
temp          20.465826    26.880615    35.328347
atemp         24.368225    30.445775    42.044800
hum           62.625000    72.989575    97.250000
windspeed     12.125325    15.625589    34.000021
casual       717.000000  1096.500000  3410.000000
registered  3664.500000  4783.250000  6946.000000
```

```
cnt         4548.500000   5966.000000   8714.000000
```

[9]:
```
data.isnull().sum()
```

[9]:
```
instant       0
dteday        0
season        0
yr            0
mnth          0
holiday       0
weekday       0
workingday    0
weathersit    0
temp          0
atemp         0
hum           0
windspeed     0
casual        0
registered    0
cnt           0
dtype: int64
```

[10]:
```
# check for duplicates

data_duplicates = data.copy()
data_duplicates.drop_duplicates(subset = None, inplace = True)
data_duplicates.shape
```

[10]:
```
(730, 16)
```

[11]:
```
# by using the drop function we understood that there are no duplicates in the␣
↪data table as both original data and data_duplicates show the same number of␣
↪rows
```

[12]:
```
#to check for distinct unique values
data.nunique()
```

[12]:
```
instant       730
dteday        730
season          4
yr              2
mnth           12
holiday         2
weekday         7
workingday      2
weathersit      3
temp          498
```

```
atemp         689
hum           594
windspeed     649
casual        605
registered    678
cnt           695
dtype: int64
```

[13]:
```python
for col in data.columns:
    print(data[col].value_counts(dropna = False).sort_index(ascending = True),␣
 ↪'\n\n\n')
```

```
1      1
2      1
3      1
4      1
5      1
      ..
726    1
727    1
728    1
729    1
730    1
Name: instant, Length: 730, dtype: int64




01-01-2018    1
01-01-2019    1
01-02-2018    1
01-02-2019    1
01-03-2018    1
             ..
31-08-2019    1
31-10-2018    1
31-10-2019    1
31-12-2018    1
31-12-2019    1
Name: dteday, Length: 730, dtype: int64




1    180
2    184
3    188
4    178
Name: season, dtype: int64
```

```
0    365
1    365
Name: yr, dtype: int64
```

```
1     62
2     56
3     62
4     60
5     62
6     60
7     62
8     62
9     60
10    62
11    60
12    62
Name: mnth, dtype: int64
```

```
0    709
1     21
Name: holiday, dtype: int64
```

```
0    104
1    105
2    105
3    104
4    104
5    104
6    104
Name: weekday, dtype: int64
```

```
0    226
1    504
Name: workingday, dtype: int64
```

```
1    463
2    246
3     21
Name: weathersit, dtype: int64
```

```
2.424346     1
3.957390     1
3.993043     1
4.407500     1
5.227500     1
            ..
34.200847    1
34.371653    1
34.781653    1
34.815847    1
35.328347    1
Name: temp, Length: 498, dtype: int64
```

```
3.953480     1
4.941955     1
5.082900     1
5.808750     1
5.896500     1
            ..
39.741450    1
40.214350    1
40.245650    1
41.318550    1
42.044800    1
Name: atemp, Length: 689, dtype: int64
```

```
0.0000     1
18.7917    1
25.4167    1
27.5833    1
29.0000    1
          ..
94.8261    1
94.9583    1
96.2500    1
97.0417    1
97.2500    1
```

```
Name: hum, Length: 594, dtype: int64


1.500244     1
2.834381     1
3.042081     1
3.042356     1
3.125550     1
            ..
27.999836    1
28.250014    1
28.292425    1
29.584721    1
34.000021    1
Name: windspeed, Length: 649, dtype: int64


2       1
9       2
15      1
25      1
34      1
       ..
3155    1
3160    1
3252    1
3283    1
3410    1
Name: casual, Length: 605, dtype: int64


20      1
416     1
432     1
451     1
472     1
       ..
6844    1
6898    1
6911    1
6917    1
6946    1
Name: registered, Length: 678, dtype: int64
```

```
22      1
431     1
441     1
506     1
605     1
       ..
8294    1
8362    1
8395    1
8555    1
8714    1
Name: cnt, Length: 695, dtype: int64
```

[14]:
```
data.columns
```

[14]:
```
Index(['instant', 'dteday', 'season', 'yr', 'mnth', 'holiday', 'weekday',
       'workingday', 'weathersit', 'temp', 'atemp', 'hum', 'windspeed',
       'casual', 'registered', 'cnt'],
      dtype='object')
```

0.1 Removing columns based on data dictionary and business understanding

In the dataset provided, you will notice that there are three columns named 'casual', 'registered', and 'cnt'. The variable 'casual' indicates the number casual users who have made a rental. The variable 'registered' on the other hand shows the total number of registered users who have made a booking on a given day. Finally, the 'cnt' variable indicates the total number of bike rentals, including both casual and registered. The model should be built taking this 'cnt' as the target variable.

The 1st column 'instant' is more similar to index column.

Also, in the dataset we have 'yr' and 'mnth' as separate columns and column 'dteday' is repeating the same information .

Hence, we can remove 'instant' , 'dteday', 'casual' and 'registered' columns which can create problems while selecting best features for model building.

[15]:
```
data_new = data[['season', 'yr', 'mnth', 'holiday', 'weekday',
       'workingday', 'weathersit', 'temp', 'atemp', 'hum', 'windspeed',
       'cnt']]
```

[16]:
```
data_new.head(3)
```

[16]:
```
   season  yr  mnth  holiday  weekday  workingday  weathersit       temp  \
0       1   0     1        0        1           1           2  14.110847
```

```
1       1   0    1        0        2           1           2  14.902598
2       1   0    1        0        3           1           1   8.050924

      atemp      hum  windspeed   cnt
0  18.18125  80.5833  10.749882   985
1  17.68695  69.6087  16.652113   801
2   9.47025  43.7273  16.636703  1349
```

[17]:
```python
# since 'cnt' is my TARGET(dependet variable)
data_new = data_new[['cnt','season', 'yr', 'mnth', 'holiday', 'weekday',␣
 ↪'workingday',
       'weathersit', 'temp', 'atemp', 'hum', 'windspeed']]
```

[18]:
```python
data_new.head(2)
```

[18]:
```
   cnt  season  yr  mnth  holiday  weekday  workingday  weathersit       temp  \
0  985       1   0     1        0        1           1           2  14.110847
1  801       1   0     1        0        2           1           2  14.902598

      atemp      hum  windspeed
0  18.18125  80.5833  10.749882
1  17.68695  69.6087  16.652113
```

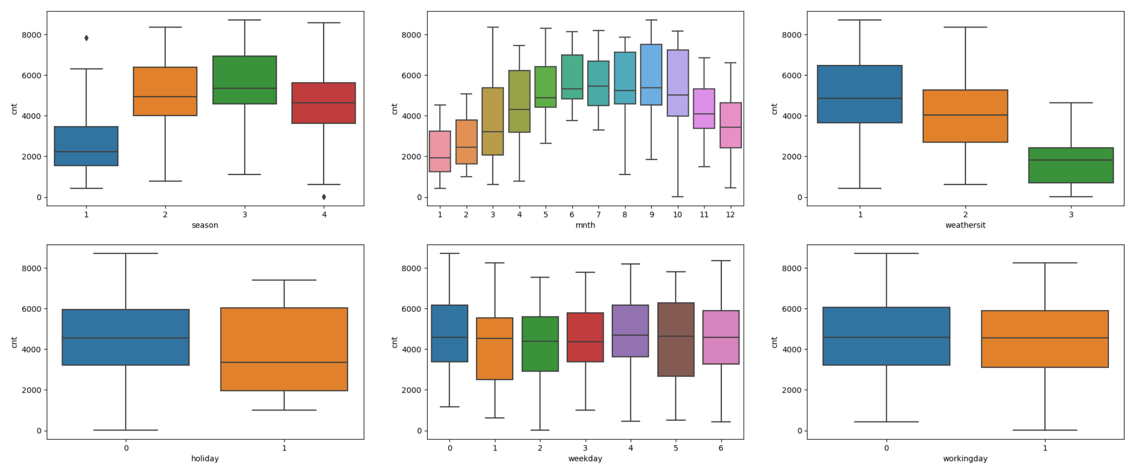
[19]:
```python
# let us create dummy variables

data_new['yr'] = data_new['yr'].astype('category')
data_new['mnth'] = data_new['mnth'].astype('category')
data_new['weekday'] = data_new['weekday'].astype('category')
data_new['workingday'] = data_new['workingday'].astype('category')
data_new['season'] = data_new['season'].astype('category')
data_new['weathersit'] = data_new['weathersit'].astype('category')
data_new['holiday'] = data_new['holiday'].astype('category')
```

[20]:
```python
data_new.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 730 entries, 0 to 729
Data columns (total 12 columns):
 #   Column      Non-Null Count  Dtype
---  ------      --------------  -----
 0   cnt         730 non-null    int64
 1   season      730 non-null    category
 2   yr          730 non-null    category
 3   mnth        730 non-null    category
 4   holiday     730 non-null    category
 5   weekday     730 non-null    category
 6   workingday  730 non-null    category
```

```
 7   weathersit  730 non-null    category
 8   temp        730 non-null    float64
 9   atemp       730 non-null    float64
 10  hum         730 non-null    float64
 11  windspeed   730 non-null    float64
dtypes: category(7), float64(4), int64(1)
memory usage: 35.1 KB
```

[21]:
```python
plt.figure(figsize = (25,10))
plt.subplot(2,3,1)
sns.boxplot(x= 'season', y = 'cnt', data = data_new)
plt.subplot(2,3,2)
sns.boxplot(x= 'mnth', y = 'cnt', data = data_new)
plt.subplot(2,3,3)
sns.boxplot(x= 'weathersit', y = 'cnt', data = data_new)
plt.subplot(2,3,4)
sns.boxplot(x= 'holiday', y = 'cnt', data = data_new)
plt.subplot(2,3,5)
sns.boxplot(x= 'weekday', y = 'cnt', data = data_new)
plt.subplot(2,3,6)
sns.boxplot(x= 'workingday', y = 'cnt', data = data_new)
plt.show()
```



From the six categorical columns we get the below insights

The inference that we could derive are as below:

season: Almost 32% of the bike booking were happening in season3 with a median of over 5000 booking (for the period of 2 years). This was followed by season2 & season4 with 27% & 25% of total booking. This indicates, season can be a good predictor for the dependent variable.

mnth: Almost 10% of the bike booking were happening in the months 5,6,7,8 & 9 with a median of over 4000 booking per month. This indicates, mnth has some trend for bookings and can be a good predictor for the dependent variable.

weathersit: Almost 67% of the bike booking were happening during 'weathersit1 with a median of close to 5000 booking (for the period of 2 years). This was followed by weathersit2 with 30% of total booking. This indicates, weathersit does show some trend towards the bike bookings can be a good predictor for the dependent variable.

holiday: Almost 97.6% of the bike booking were happening when it is not a holiday which means this data is clearly biased. This indicates, holiday CANNOT be a good predictor for the dependent variable.

weekday: weekday variable shows very close trend (between 13.5%-14.8% of total booking on all days of the week) having their independent medians between 4000 to 5000 bookings. This variable can have some or no influence towards the predictor. I will let the model decide if this needs to be added or not.

workingday: Almost 69% of the bike booking were happening in 'workingday' with a median of close to 5000 booking (for the period of 2 years). This indicates, workingday can be a good predictor for the dependent variable

[22]:
```
data_new = pd.get_dummies(data_new)
```

[23]:
```
data_new.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 730 entries, 0 to 729
Data columns (total 37 columns):
 #   Column        Non-Null Count  Dtype
---  ------        --------------  -----
 0   cnt           730 non-null    int64
 1   temp          730 non-null    float64
 2   atemp         730 non-null    float64
 3   hum           730 non-null    float64
 4   windspeed     730 non-null    float64
 5   season_1      730 non-null    uint8
 6   season_2      730 non-null    uint8
 7   season_3      730 non-null    uint8
 8   season_4      730 non-null    uint8
 9   yr_0          730 non-null    uint8
 10  yr_1          730 non-null    uint8
 11  mnth_1        730 non-null    uint8
 12  mnth_2        730 non-null    uint8
 13  mnth_3        730 non-null    uint8
 14  mnth_4        730 non-null    uint8
 15  mnth_5        730 non-null    uint8
 16  mnth_6        730 non-null    uint8
 17  mnth_7        730 non-null    uint8
 18  mnth_8        730 non-null    uint8
 19  mnth_9        730 non-null    uint8
 20  mnth_10       730 non-null    uint8
```

```
 21  mnth_11        730 non-null    uint8
 22  mnth_12        730 non-null    uint8
 23  holiday_0      730 non-null    uint8
 24  holiday_1      730 non-null    uint8
 25  weekday_0      730 non-null    uint8
 26  weekday_1      730 non-null    uint8
 27  weekday_2      730 non-null    uint8
 28  weekday_3      730 non-null    uint8
 29  weekday_4      730 non-null    uint8
 30  weekday_5      730 non-null    uint8
 31  weekday_6      730 non-null    uint8
 32  workingday_0   730 non-null    uint8
 33  workingday_1   730 non-null    uint8
 34  weathersit_1   730 non-null    uint8
 35  weathersit_2   730 non-null    uint8
 36  weathersit_3   730 non-null    uint8
dtypes: float64(4), int64(1), uint8(32)
memory usage: 51.5 KB
```

[24]:
```python
bool_columns = data_new.select_dtypes(include = 'uint8').columns
data_new[bool_columns] = data_new[bool_columns].astype(int)
data_new.head().T
```

[24]:

	0	1	2	3	4
cnt	985.000000	801.000000	1349.000000	1562.000000	1600.000000
temp	14.110847	14.902598	8.050924	8.200000	9.305237
atemp	18.181250	17.686950	9.470250	10.606100	11.463500
hum	80.583300	69.608700	43.727300	59.043500	43.695700
windspeed	10.749882	16.652113	16.636703	10.739832	12.522300
season_1	1.000000	1.000000	1.000000	1.000000	1.000000
season_2	0.000000	0.000000	0.000000	0.000000	0.000000
season_3	0.000000	0.000000	0.000000	0.000000	0.000000
season_4	0.000000	0.000000	0.000000	0.000000	0.000000
yr_0	1.000000	1.000000	1.000000	1.000000	1.000000
yr_1	0.000000	0.000000	0.000000	0.000000	0.000000
mnth_1	1.000000	1.000000	1.000000	1.000000	1.000000
mnth_2	0.000000	0.000000	0.000000	0.000000	0.000000
mnth_3	0.000000	0.000000	0.000000	0.000000	0.000000
mnth_4	0.000000	0.000000	0.000000	0.000000	0.000000
mnth_5	0.000000	0.000000	0.000000	0.000000	0.000000
mnth_6	0.000000	0.000000	0.000000	0.000000	0.000000
mnth_7	0.000000	0.000000	0.000000	0.000000	0.000000
mnth_8	0.000000	0.000000	0.000000	0.000000	0.000000
mnth_9	0.000000	0.000000	0.000000	0.000000	0.000000
mnth_10	0.000000	0.000000	0.000000	0.000000	0.000000
mnth_11	0.000000	0.000000	0.000000	0.000000	0.000000
mnth_12	0.000000	0.000000	0.000000	0.000000	0.000000

```
holiday_0       1.000000    1.000000    1.000000    1.000000    1.000000
holiday_1       0.000000    0.000000    0.000000    0.000000    0.000000
weekday_0       0.000000    0.000000    0.000000    0.000000    0.000000
weekday_1       1.000000    0.000000    0.000000    0.000000    0.000000
weekday_2       0.000000    1.000000    0.000000    0.000000    0.000000
weekday_3       0.000000    0.000000    1.000000    0.000000    0.000000
weekday_4       0.000000    0.000000    0.000000    1.000000    0.000000
weekday_5       0.000000    0.000000    0.000000    0.000000    1.000000
weekday_6       0.000000    0.000000    0.000000    0.000000    0.000000
workingday_0    0.000000    0.000000    0.000000    0.000000    0.000000
workingday_1    1.000000    1.000000    1.000000    1.000000    1.000000
weathersit_1    0.000000    0.000000    1.000000    1.000000    1.000000
weathersit_2    1.000000    1.000000    0.000000    0.000000    0.000000
weathersit_3    0.000000    0.000000    0.000000    0.000000    0.000000
```

[25]:
```
data_new.shape
```

[25]:
```
(730, 37)
```

[26]:
```
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import MinMaxScaler
from sklearn.feature_selection import RFE
from sklearn.linear_model import LinearRegression
```

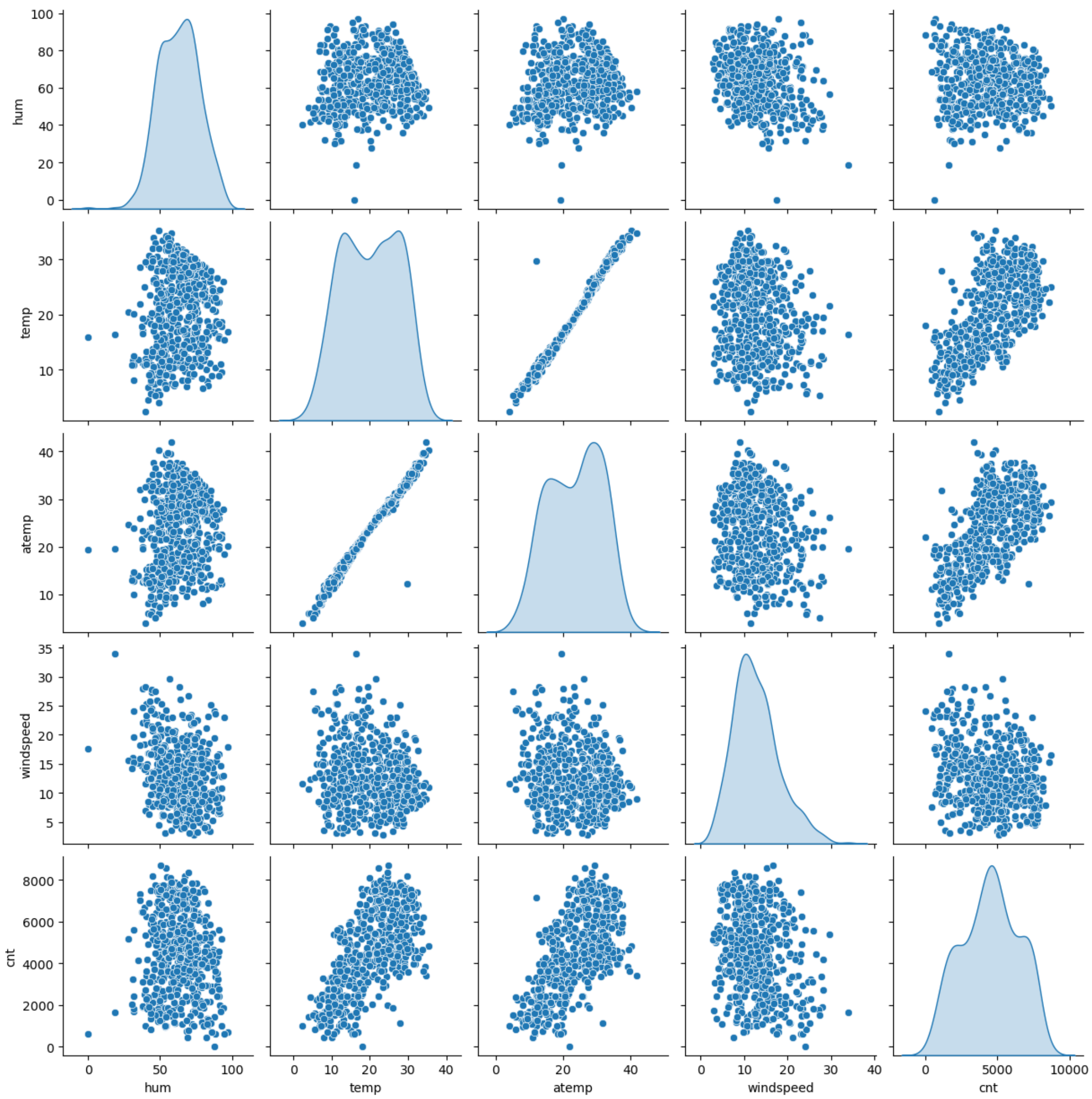
[27]:
```
np.random.seed(0)
data_new_train, data_new_test = train_test_split(data_new,train_size = 0.8,␣
↪random_state = 100)
```

[28]:
```
print(data_new_train.shape)
print(data_new_test.shape)
```

```
(584, 37)
(146, 37)
```

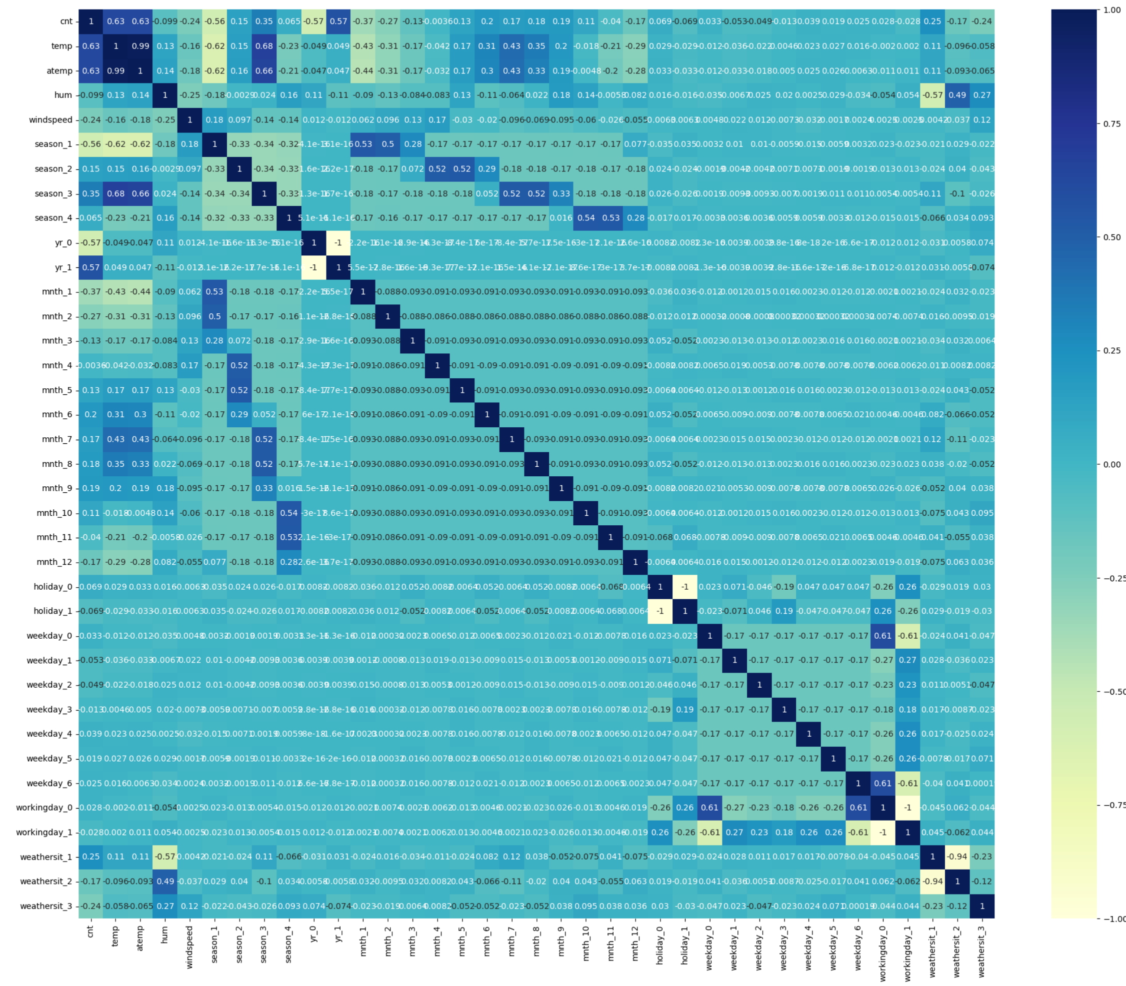
[29]:
```
data_num = data_new_train[['hum','temp', 'atemp', 'windspeed','cnt']]
sns.pairplot(data_num, diag_kind = 'kde')
plt.show()
```

```
/opt/conda/envs/anaconda-panel-2023.05-py310/lib/python3.11/site-
packages/seaborn/axisgrid.py:118: UserWarning: The figure layout has changed to
tight
  self._figure.tight_layout(*args, **kwargs)
```



#'cnt', 'temp', 'atemp', 'hum', 'windspeed', 'season_1', 'season_2', 'season_3', 'season_4', 'yr_0', 'yr_1', 'mnth_1', 'mnth_2', 'mnth_3', 'mnth_4', 'mnth_5', 'mnth_6', 'mnth_7', 'mnth_8', 'mnth_9', 'mnth_10', 'mnth_11', 'mnth_12', 'holiday_0', 'holiday_1', 'weekday_0', 'weekday_1', 'weekday_2', 'weekday_3', 'weekday_4', 'weekday_5', 'weekday_6', 'workingday_0', 'workingday_1', 'weathersit_1', 'weathersit_2', 'weathersit_3'

[31]:
```
plt.figure(figsize = (25,20))
sns.heatmap(data_new.corr(), annot = True, cmap = 'YlGnBu')
plt.show()
```



[32]:
```python
data_new.columns
```

[32]:
```
Index(['cnt', 'temp', 'atemp', 'hum', 'windspeed', 'season_1', 'season_2',
       'season_3', 'season_4', 'yr_0', 'yr_1', 'mnth_1', 'mnth_2', 'mnth_3',
       'mnth_4', 'mnth_5', 'mnth_6', 'mnth_7', 'mnth_8', 'mnth_9', 'mnth_10',
       'mnth_11', 'mnth_12', 'holiday_0', 'holiday_1', 'weekday_0',
       'weekday_1', 'weekday_2', 'weekday_3', 'weekday_4', 'weekday_5',
       'weekday_6', 'workingday_0', 'workingday_1', 'weathersit_1',
       'weathersit_2', 'weathersit_3'],
      dtype='object')
```

[33]:
```python
scaler = MinMaxScaler()
num_vars = ['hum','temp', 'atemp', 'windspeed','cnt']
data_new_train[num_vars] = scaler.fit_transform(data_new_train[num_vars])
data_new_train.head()
```

[33]:

```
          cnt      temp     atemp       hum  windspeed  season_1  season_2  \
367  0.254717  0.113228  0.061963  0.454701   0.695175         1         0
648  0.868385  0.468352  0.462175  0.477458   0.299450         0         0
44   0.217556  0.443431  0.419099  0.387290   0.807474         1         0
705  0.573631  0.326094  0.318824  0.787463   0.189819         0         0
379  0.263346  0.133996  0.108365  0.431945   0.449210         1         0

     season_3  season_4  yr_0  …  weekday_2  weekday_3  weekday_4  \
367         0         0     0  …          0          0          1
648         0         1     0  …          0          0          0
44          0         0     1  …          0          1          0
705         0         1     0  …          0          0          0
379         0         0     0  …          1          0          0

     weekday_5  weekday_6  workingday_0  workingday_1  weathersit_1  \
367          0          0             0             1             1
648          1          0             0             1             1
44           0          0             0             1             1
705          0          1             1             0             0
379          0          0             0             1             1

     weathersit_2  weathersit_3
367             0             0
648             0             0
44              0             0
705             1             0
379             0             0

[5 rows x 37 columns]
```

[34]:

```
data_new_train.describe().T
```

[34]:

```
              count      mean       std  min       25%       50%       75%  \
cnt           584.0  0.515792  0.225336  0.0  0.350696  0.522837  0.691872
temp          584.0  0.537414  0.225336  0.0  0.340113  0.545191  0.736512
atemp         584.0  0.513175  0.211663  0.0  0.331819  0.530558  0.690521
hum           584.0  0.649499  0.144219  0.0  0.535852  0.653714  0.752361
windspeed     584.0  0.319463  0.168114  0.0  0.199177  0.294764  0.410413
season_1      584.0  0.251712  0.434369  0.0  0.000000  0.000000  1.000000
season_2      584.0  0.246575  0.431387  0.0  0.000000  0.000000  0.000000
season_3      584.0  0.251712  0.434369  0.0  0.000000  0.000000  1.000000
season_4      584.0  0.250000  0.433384  0.0  0.000000  0.000000  0.250000
yr_0          584.0  0.486301  0.500241  0.0  0.000000  0.000000  1.000000
yr_1          584.0  0.513699  0.500241  0.0  0.000000  1.000000  1.000000
mnth_1        584.0  0.087329  0.282558  0.0  0.000000  0.000000  0.000000
mnth_2        584.0  0.073630  0.261392  0.0  0.000000  0.000000  0.000000
mnth_3        584.0  0.090753  0.287504  0.0  0.000000  0.000000  0.000000
```

```
mnth_4         584.0  0.077055  0.266907  0.0  0.000000  0.000000  0.000000
mnth_5         584.0  0.087329  0.282558  0.0  0.000000  0.000000  0.000000
mnth_6         584.0  0.077055  0.266907  0.0  0.000000  0.000000  0.000000
mnth_7         584.0  0.075342  0.264169  0.0  0.000000  0.000000  0.000000
mnth_8         584.0  0.090753  0.287504  0.0  0.000000  0.000000  0.000000
mnth_9         584.0  0.080479  0.272267  0.0  0.000000  0.000000  0.000000
mnth_10        584.0  0.092466  0.289931  0.0  0.000000  0.000000  0.000000
mnth_11        584.0  0.080479  0.272267  0.0  0.000000  0.000000  0.000000
mnth_12        584.0  0.087329  0.282558  0.0  0.000000  0.000000  0.000000
holiday_0      584.0  0.972603  0.163378  0.0  1.000000  1.000000  1.000000
holiday_1      584.0  0.027397  0.163378  0.0  0.000000  0.000000  0.000000
weekday_0      584.0  0.130137  0.336743  0.0  0.000000  0.000000  0.000000
weekday_1      584.0  0.155822  0.362997  0.0  0.000000  0.000000  0.000000
weekday_2      584.0  0.159247  0.366220  0.0  0.000000  0.000000  0.000000
weekday_3      584.0  0.136986  0.344128  0.0  0.000000  0.000000  0.000000
weekday_4      584.0  0.145548  0.352955  0.0  0.000000  0.000000  0.000000
weekday_5      584.0  0.152397  0.359714  0.0  0.000000  0.000000  0.000000
weekday_6      584.0  0.119863  0.325080  0.0  0.000000  0.000000  0.000000
workingday_0   584.0  0.273973  0.446377  0.0  0.000000  0.000000  1.000000
workingday_1   584.0  0.726027  0.446377  0.0  0.000000  1.000000  1.000000
weathersit_1   584.0  0.630137  0.483181  0.0  0.000000  1.000000  1.000000
weathersit_2   584.0  0.342466  0.474941  0.0  0.000000  0.000000  1.000000
weathersit_3   584.0  0.027397  0.163378  0.0  0.000000  0.000000  0.000000

               max
cnt            1.0
temp           1.0
atemp          1.0
hum            1.0
windspeed      1.0
season_1       1.0
season_2       1.0
season_3       1.0
season_4       1.0
yr_0           1.0
yr_1           1.0
mnth_1         1.0
mnth_2         1.0
mnth_3         1.0
mnth_4         1.0
mnth_5         1.0
mnth_6         1.0
mnth_7         1.0
mnth_8         1.0
mnth_9         1.0
mnth_10        1.0
mnth_11        1.0
```

```
mnth_12       1.0
holiday_0     1.0
holiday_1     1.0
weekday_0     1.0
weekday_1     1.0
weekday_2     1.0
weekday_3     1.0
weekday_4     1.0
weekday_5     1.0
weekday_6     1.0
workingday_0  1.0
workingday_1  1.0
weathersit_1  1.0
weathersit_2  1.0
weathersit_3  1.0
```

[35]:
```
X_train = data_new_train
y_train = data_new_train.pop('cnt')
```

[36]:
```
X_train.head()
```

[36]:
```
         temp     atemp       hum  windspeed  season_1  season_2  season_3  \
367  0.113228  0.061963  0.454701   0.695175         1         0         0
648  0.468352  0.462175  0.477458   0.299450         0         0         0
44   0.443431  0.419099  0.387290   0.807474         1         0         0
705  0.326094  0.318824  0.787463   0.189819         0         0         0
379  0.133996  0.108365  0.431945   0.449210         1         0         0

     season_4  yr_0  yr_1  …  weekday_2  weekday_3  weekday_4  weekday_5  \
367         0     0     1  …          0          0          1          0
648         1     0     1  …          0          0          0          1
44          0     1     0  …          0          1          0          0
705         1     0     1  …          0          0          0          0
379         0     0     1  …          1          0          0          0

     weekday_6  workingday_0  workingday_1  weathersit_1  weathersit_2  \
367          0             0             1             1             0
648          0             0             1             1             0
44           0             0             1             1             0
705          1             1             0             0             1
379          0             0             1             1             0

     weathersit_3
367             0
648             0
44              0
705             0
```

```
379              0

[5 rows x 36 columns]
```

[37]:
```python
y_train.head()
```

[37]:
```
367    0.254717
648    0.868385
44     0.217556
705    0.573631
379    0.263346
Name: cnt, dtype: float64
```

[61]:
```python
lr = LinearRegression()
lr.fit(X_train, y_train)
```

[61]:
```
LinearRegression()
```

[63]:
```python
rfe = RFE(estimator=lr, n_features_to_select=20)
rfe = rfe.fit(X_train, y_train)
```

[65]:
```python
selected_features = X_train.columns[rfe.support_]
print("Selected Features:", selected_features)
```

```
Selected Features: Index(['temp', 'windspeed', 'season_1', 'season_2',
'season_4', 'yr_0', 'yr_1',
       'mnth_1', 'mnth_2', 'mnth_7', 'mnth_9', 'mnth_11', 'mnth_12',
       'holiday_0', 'holiday_1', 'weekday_1', 'weekday_2', 'weathersit_1',
       'weathersit_2', 'weathersit_3'],
      dtype='object')
```

[67]:
```python
list(zip(X_train.columns,rfe.support_,rfe.ranking_))
```

[67]:
```
[('temp', True, 1),
 ('atemp', False, 17),
 ('hum', False, 16),
 ('windspeed', True, 1),
 ('season_1', True, 1),
 ('season_2', True, 1),
 ('season_3', False, 2),
 ('season_4', True, 1),
 ('yr_0', True, 1),
 ('yr_1', True, 1),
 ('mnth_1', True, 1),
 ('mnth_2', True, 1),
 ('mnth_3', False, 9),
 ('mnth_4', False, 8),
```

```
 ('mnth_5', False, 5),
 ('mnth_6', False, 3),
 ('mnth_7', True, 1),
 ('mnth_8', False, 4),
 ('mnth_9', True, 1),
 ('mnth_10', False, 7),
 ('mnth_11', True, 1),
 ('mnth_12', True, 1),
 ('holiday_0', True, 1),
 ('holiday_1', True, 1),
 ('weekday_0', False, 12),
 ('weekday_1', True, 1),
 ('weekday_2', True, 1),
 ('weekday_3', False, 15),
 ('weekday_4', False, 13),
 ('weekday_5', False, 14),
 ('weekday_6', False, 11),
 ('workingday_0', False, 6),
 ('workingday_1', False, 10),
 ('weathersit_1', True, 1),
 ('weathersit_2', True, 1),
 ('weathersit_3', True, 1)]
```

[69]:
```
col = X_train.columns[rfe.support_]
col
```

[69]:
```
Index(['temp', 'windspeed', 'season_1', 'season_2', 'season_4', 'yr_0', 'yr_1',
       'mnth_1', 'mnth_2', 'mnth_7', 'mnth_9', 'mnth_11', 'mnth_12',
       'holiday_0', 'holiday_1', 'weekday_1', 'weekday_2', 'weathersit_1',
       'weathersit_2', 'weathersit_3'],
      dtype='object')
```

[71]:
```
X_train.columns[~rfe.support_]
```

[71]:
```
Index(['atemp', 'hum', 'season_3', 'mnth_3', 'mnth_4', 'mnth_5', 'mnth_6',
       'mnth_8', 'mnth_10', 'weekday_0', 'weekday_3', 'weekday_4', 'weekday_5',
       'weekday_6', 'workingday_0', 'workingday_1'],
      dtype='object')
```

[73]:
```
X_train_rfe = X_train[col]
```

[75]:
```
#Model 1 :
from statsmodels.stats.outliers_influence import variance_inflation_factor
vif = pd.DataFrame()
vif['Features'] = X_train_rfe.columns
vif['VIF'] = [variance_inflation_factor(X_train_rfe.values, i) for i in␣
↪range(X_train_rfe.shape[1])]
```

```
vif['VIF'] = round(vif['VIF'], 2)
vif = vif.sort_values(by="VIF", ascending=False)

vif
```

```
/opt/conda/envs/anaconda-panel-2023.05-py310/lib/python3.11/site-
packages/statsmodels/stats/outliers_influence.py:198: RuntimeWarning: divide by
zero encountered in scalar divide
  vif = 1. / (1. - r_squared_i)
```

[75]:

	Features	VIF
19	weathersit_3	inf
13	holiday_0	inf
18	weathersit_2	inf
17	weathersit_1	inf
5	yr_0	inf
6	yr_1	inf
14	holiday_1	inf
2	season_1	5.86
0	temp	4.42
4	season_4	3.54
3	season_2	2.76
7	mnth_1	2.33
8	mnth_2	1.93
11	mnth_11	1.73
12	mnth_12	1.64
9	mnth_7	1.51
10	mnth_9	1.32
1	windspeed	1.11
15	weekday_1	1.05
16	weekday_2	1.04

Building Linear Model using 'STATS MODEL'

Model 1: VIF check

A VIF of 1 indicates no correlation between the variable and other predictors. A VIF between 1 and 5 indicates moderate correlation. A VIF greater than 5 indicates high correlation, and anything above 10 is considered very high, suggesting serious multicollinearity.

[77]:
```
import statsmodels.api as sm
X_train_lr = sm.add_constant(X_train_rfe)
lr = sm.OLS(y_train,X_train_lr).fit()
```

[81]:
```
lr.params
```

[81]:
```
const            0.089645
temp             0.424659
```

```
windspeed      -0.151634
season_1       -0.067125
season_2        0.031821
season_4        0.096941
yr_0           -0.071287
yr_1            0.160931
mnth_1         -0.069752
mnth_2         -0.038048
mnth_7         -0.044408
mnth_9          0.059237
mnth_11        -0.062053
mnth_12        -0.065282
holiday_0       0.092736
holiday_1      -0.003092
weekday_1      -0.031961
weekday_2      -0.031308
weathersit_1    0.157066
weathersit_2    0.076421
weathersit_3   -0.143842
dtype: float64
```

[83]:
```python
print(lr.summary())
```

```
                            OLS Regression Results
==============================================================================
Dep. Variable:                    cnt   R-squared:                       0.849
Model:                            OLS   Adj. R-squared:                  0.845
Method:                 Least Squares   F-statistic:                     187.4
Date:                Mon, 02 Sep 2024   Prob (F-statistic):          1.90e-219
Time:                        14:27:15   Log-Likelihood:                 594.42
No. Observations:                 584   AIC:                            -1153.
Df Residuals:                     566   BIC:                            -1074.
Df Model:                          17
Covariance Type:            nonrobust
================================================================================
                   coef    std err          t      P>|t|      [0.025      0.975]
--------------------------------------------------------------------------------
const            0.0896      0.014      6.294      0.000       0.062       0.118
temp             0.4247      0.034     12.371      0.000       0.357       0.492
windspeed       -0.1516      0.023     -6.571      0.000      -0.197      -0.106
season_1        -0.0671      0.021     -3.274      0.001      -0.107      -0.027
season_2         0.0318      0.014      2.245      0.025       0.004       0.060
season_4         0.0969      0.016      6.069      0.000       0.066       0.128
yr_0            -0.0713      0.008     -9.222      0.000      -0.086      -0.056
yr_1             0.1609      0.008     19.350      0.000       0.145       0.177
mnth_1          -0.0698      0.020     -3.508      0.000      -0.109      -0.031
mnth_2          -0.0380      0.020     -1.946      0.052      -0.076       0.000
```

```
mnth_7          -0.0444      0.017     -2.596      0.010      -0.078      -0.011
mnth_9           0.0592      0.016      3.816      0.000       0.029       0.090
mnth_11         -0.0621      0.018     -3.496      0.001      -0.097      -0.027
mnth_12         -0.0653      0.017     -3.921      0.000      -0.098      -0.033
holiday_0        0.0927      0.011      8.444      0.000       0.071       0.114
holiday_1       -0.0031      0.016     -0.198      0.843      -0.034       0.028
weekday_1       -0.0320      0.010     -3.074      0.002      -0.052      -0.012
weekday_2       -0.0313      0.010     -3.053      0.002      -0.051      -0.011
weathersit_1     0.1571      0.009     17.539      0.000       0.139       0.175
weathersit_2     0.0764      0.009      8.261      0.000       0.058       0.095
weathersit_3    -0.1438      0.017     -8.522      0.000      -0.177      -0.111
==============================================================================
Omnibus:                       93.443   Durbin-Watson:                   2.016
Prob(Omnibus):                  0.000   Jarque-Bera (JB):              246.675
Skew:                          -0.806   Prob(JB):                     2.72e-54
Kurtosis:                       5.746   Cond. No.                     2.31e+16
==============================================================================

Notes:
[1] Standard Errors assume that the covariance matrix of the errors is correctly
specified.
[2] The smallest eigenvalue is 4.01e-30. This might indicate that there are
strong multicollinearity problems or that the design matrix is singular.
```

[ ]:
```
# Model 2
```