Linear Regression Predictive Modeling on Bike Sharing Data:

Problem Statement

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:

Which variables are significant in predicting the demand for shared bikes. 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.

In [1]:
```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
import sklearn
from sklearn.model_selection import train_test_split

from sklearn.preprocessing import MinMaxScaler
from sklearn.metrics import r2_score,mean_squared_error
from sklearn.linear_model import LinearRegression
from sklearn.feature_selection import RFE

import statsmodels.api as sm
from statsmodels.stats.outliers_influence import variance_inflation_factor
```

In [2]:
```
import warnings
warnings.filterwarnings('ignore')
```

Reading The Data

In [3]:
```
bike = pd.read_csv('day.csv')
#reading the data
bike
```

Out[3]:

	instant	dteday	season	yr	mnth	holiday	weekday	workingday	weathersit	temp	atemp	hum	windspeed	casual	registered	c
0	1	01-01-2018	1	0	1	0	1	1	2	14.110847	18.18125	80.5833	10.749882	331	654	98
1	2	02-01-2018	1	0	1	0	2	1	2	14.902598	17.68695	69.6087	16.652113	131	670	80
2	3	03-01-2018	1	0	1	0	3	1	1	8.050924	9.47025	43.7273	16.636703	120	1229	134
3	4	04-01-2018	1	0	1	0	4	1	1	8.200000	10.60610	59.0435	10.739832	108	1454	156
4	5	05-01-2018	1	0	1	0	5	1	1	9.305237	11.46350	43.6957	12.522300	82	1518	160
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	
725	726	27-12-2019	1	1	12	0	5	1	2	10.420847	11.33210	65.2917	23.458911	247	1867	211
726	727	28-12-2019	1	1	12	0	6	0	2	10.386653	12.75230	59.0000	10.416557	644	2451	309
727	728	29-12-2019	1	1	12	0	0	0	2	10.386653	12.12000	75.2917	8.333661	159	1182	134
728	729	30-12-2019	1	1	12	0	1	1	1	10.489153	11.58500	48.3333	23.500518	364	1432	179
729	730	31-12-2019	1	1	12	0	2	1	2	8.849153	11.17435	57.7500	10.374682	439	2290	272

730 rows × 16 columns

The Dataset has 730 rows and 16 columns

In [4]:
```
bike.shape #size of the data
```

Out[4]: (730, 16)

In [5]:
```
bike.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
```

In [6]:
```
bike.describe()
```

Out[6]:

	instant	season	yr	mnth	holiday	weekday	workingday	weathersit	temp	atemp	hum	windspe
count	730.000000	730.000000	730.000000	730.000000	730.000000	730.000000	730.000000	730.000000	730.000000	730.000000	730.000000	730.0000
mean	365.500000	2.498630	0.500000	6.526027	0.028767	2.995890	0.690411	1.394521	20.319259	23.726322	62.765175	12.7636
std	210.877136	1.110184	0.500343	3.450215	0.167266	2.000339	0.462641	0.544807	7.506729	8.150308	14.237589	5.1958
min	1.000000	1.000000	0.000000	1.000000	0.000000	0.000000	0.000000	1.000000	2.424346	3.953480	0.000000	1.5002
25%	183.250000	2.000000	0.000000	4.000000	0.000000	1.000000	0.000000	1.000000	13.811885	16.889713	52.000000	9.0416
50%	365.500000	3.000000	0.500000	7.000000	0.000000	3.000000	1.000000	1.000000	20.465826	24.368225	62.625000	12.1253
75%	547.750000	3.000000	1.000000	10.000000	0.000000	5.000000	1.000000	2.000000	26.880615	30.445775	72.989575	15.6255
max	730.000000	4.000000	1.000000	12.000000	1.000000	6.000000	1.000000	3.000000	35.328347	42.044800	97.250000	34.0000

In [7]:
```
bike.size
```

Out[7]:
```
11680
```

Data Cleaning & droping columns which are not relevant for analysis

In [8]:
```
bike.isnull().sum()
```

Out[8]:
```
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
```

In [9]:
```
bike.drop(['instant'],axis=1,inplace=True)
```

In [10]:
```
bike.drop(['casual'],axis=1,inplace=True)
```

Inspecting data after dropping

In [11]:
```
bike.head()
```

Out[11]:

	dteday	season	yr	mnth	holiday	weekday	workingday	weathersit	temp	atemp	hum	windspeed	registered	cnt
0	01-01-2018	1	0	1	0	1	1	2	14.110847	18.18125	80.5833	10.749882	654	985
1	02-01-2018	1	0	1	0	2	1	2	14.902598	17.68695	69.6087	16.652113	670	801
2	03-01-2018	1	0	1	0	3	1	1	8.050924	9.47025	43.7273	16.636703	1229	1349
3	04-01-2018	1	0	1	0	4	1	1	8.200000	10.60610	59.0435	10.739832	1454	1562
4	05-01-2018	1	0	1	0	5	1	1	9.305237	11.46350	43.6957	12.522300	1518	1600

In [12]:
```
bike.drop(['dteday'],axis=1,inplace=True)
```

In [13]:
```
bike.head()
```

Out[13]:

	season	yr	mnth	holiday	weekday	workingday	weathersit	temp	atemp	hum	windspeed	registered	cnt
0	1	0	1	0	1	1	2	14.110847	18.18125	80.5833	10.749882	654	985
1	1	0	1	0	2	1	2	14.902598	17.68695	69.6087	16.652113	670	801
2	1	0	1	0	3	1	1	8.050924	9.47025	43.7273	16.636703	1229	1349
3	1	0	1	0	4	1	1	8.200000	10.60610	59.0435	10.739832	1454	1562
4	1	0	1	0	5	1	1	9.305237	11.46350	43.6957	12.522300	1518	1600

In [14]:
```
bike.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 730 entries, 0 to 729
Data columns (total 13 columns):
 #   Column      Non-Null Count  Dtype  
---  ------      --------------  -----  
 0   season      730 non-null    int64  
 1   yr          730 non-null    int64  
 2   mnth        730 non-null    int64  
 3   holiday     730 non-null    int64  
 4   weekday     730 non-null    int64  
 5   workingday  730 non-null    int64  
 6   weathersit  730 non-null    int64  
 7   temp        730 non-null    float64
 8   atemp       730 non-null    float64
 9   hum         730 non-null    float64
 10  windspeed   730 non-null    float64
 11  registered  730 non-null    int64  
 12  cnt         730 non-null    int64  
dtypes: float64(4), int64(9)
memory usage: 74.3 KB
```

In [15]:
```
bike.columns
```

Out[15]:
```
Index(['season', 'yr', 'mnth', 'holiday', 'weekday', 'workingday',
       'weathersit', 'temp', 'atemp', 'hum', 'windspeed', 'registered', 'cnt'],
      dtype='object')
```

In [16]:
```
bike.corr()
```

Out[16]:

	season	yr	mnth	holiday	weekday	workingday	weathersit	temp	atemp	hum	windspeed	re
season	1.000000e+00	-3.279074e-16	8.310321e-01	-0.010868	0.004012	0.017868	0.021306	0.333361	0.342014	0.208220	-0.229607	(
yr	-3.279074e-16	1.000000e+00	-5.162656e-16	0.008195	0.000685	-0.011852	-0.050322	0.048789	0.047215	-0.112547	-0.011624	(
mnth	8.310321e-01	-5.162656e-16	1.000000e+00	0.018905	-0.004059	-0.010414	0.045613	0.219083	0.226430	0.224937	-0.208013	(
holiday	-1.086804e-02	8.195345e-03	1.890483e-02	1.000000	-0.020145	-0.257009	-0.034395	-0.028764	-0.032703	-0.015662	0.006257	-(
weekday	4.012478e-03	6.852851e-04	-4.059002e-03	-0.020145	1.000000	0.001588	0.034216	0.044876	0.037964	0.008780	-0.017230	(
workingday	1.786841e-02	-1.185197e-02	-1.041372e-02	-0.257009	0.001588	1.000000	-0.026332	0.002044	0.010657	0.053770	-0.002453	(
weathersit	2.130636e-02	-5.032247e-02	4.561335e-02	-0.034395	0.034216	-0.026332	1.000000	-0.119503	-0.120559	0.590277	0.039769	-(
temp	3.333607e-01	4.878919e-02	2.190833e-01	-0.028764	0.044876	0.002044	-0.119503	1.000000	0.991696	0.128565	-0.158186	(
atemp	3.420139e-01	4.721519e-02	2.264302e-01	-0.032703	0.037964	0.010657	-0.120559	0.991696	1.000000	0.141512	-0.183876	(
hum	2.082196e-01	-1.125471e-01	2.249368e-01	-0.015662	0.008780	0.053770	0.590277	0.128565	0.141512	1.000000	-0.248506	-(
windspeed	-2.296069e-01	-1.162435e-02	-2.080131e-01	0.006257	-0.017230	-0.002453	0.039769	-0.158186	-0.183876	-0.248506	1.000000	-(
registered	4.103102e-01	5.969106e-01	2.919516e-01	-0.109142	0.199326	0.005466	-0.259025	0.539436	0.543678	-0.089212	-0.217914	1
cnt	4.045838e-01	5.697285e-01	2.781909e-01	-0.068764	0.036183	-0.027640	-0.295929	0.627044	0.630685	-0.098543	-0.235132	(

Handling outliers

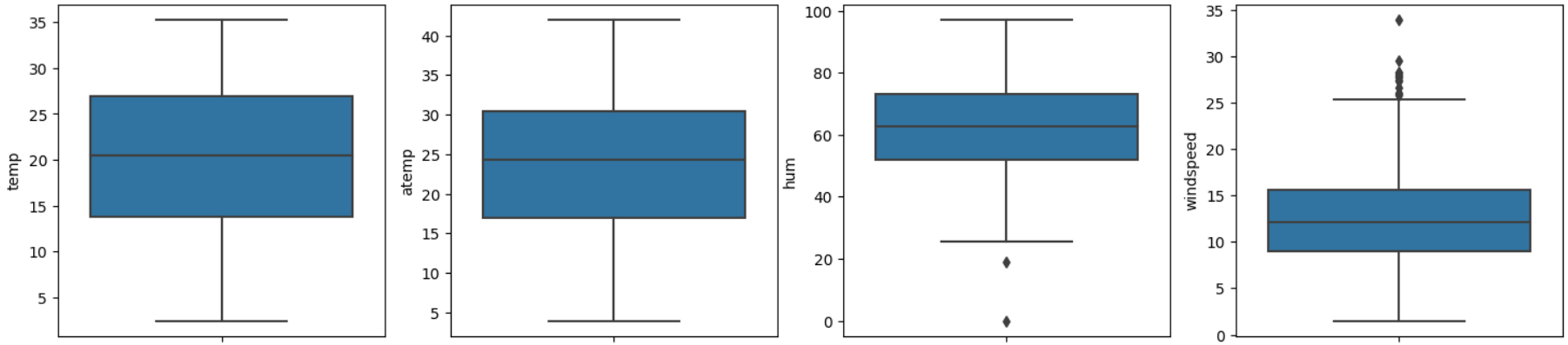
In [17]:
```
bike.columns
```

Out[17]:
```
Index(['season', 'yr', 'mnth', 'holiday', 'weekday', 'workingday',
       'weathersit', 'temp', 'atemp', 'hum', 'windspeed', 'registered', 'cnt'],
      dtype='object')
```

In [18]:
```
bike.nunique()
```

Out[18]:
```
season          4
yr              2
mnth           12
holiday         2
weekday         7
workingday      2
weathersit      3
temp          498
atemp         689
hum           594
windspeed     649
registered    678
cnt           695
dtype: int64
```

In [19]:
```
cols = ['temp', 'atemp', 'hum', 'windspeed']
plt.figure(figsize=(18,4))

i = 1
for col in cols:
    plt.subplot(1,4,i)
    sns.boxplot(y=col, data=bike)
    i+=1
```



So on above plots we don't have any outliers

EDA

In [20]:
```
#changing categorical data which were primarily numeric to more meaningful one
bike.season.replace({1:"spring", 2:"summer", 3:"fall", 4:"winter"},inplace = True)

bike.weathersit.replace({1:'good',2:'moderate',3:'bad',4:'severe'},inplace = True)

bike.mnth = bike.mnth.replace({1: 'jan',2: 'feb',3: 'mar',4: 'apr',5: 'may',6: 'jun',
                  7: 'jul',8: 'aug',9: 'sept',10: 'oct',11: 'nov',12: 'dec'})

bike.weekday = bike.weekday.replace({0: 'sun',1: 'mon',2: 'tue',3: 'wed',4: 'thu',5: 'fri',6: 'sat'})
bike.head()
```

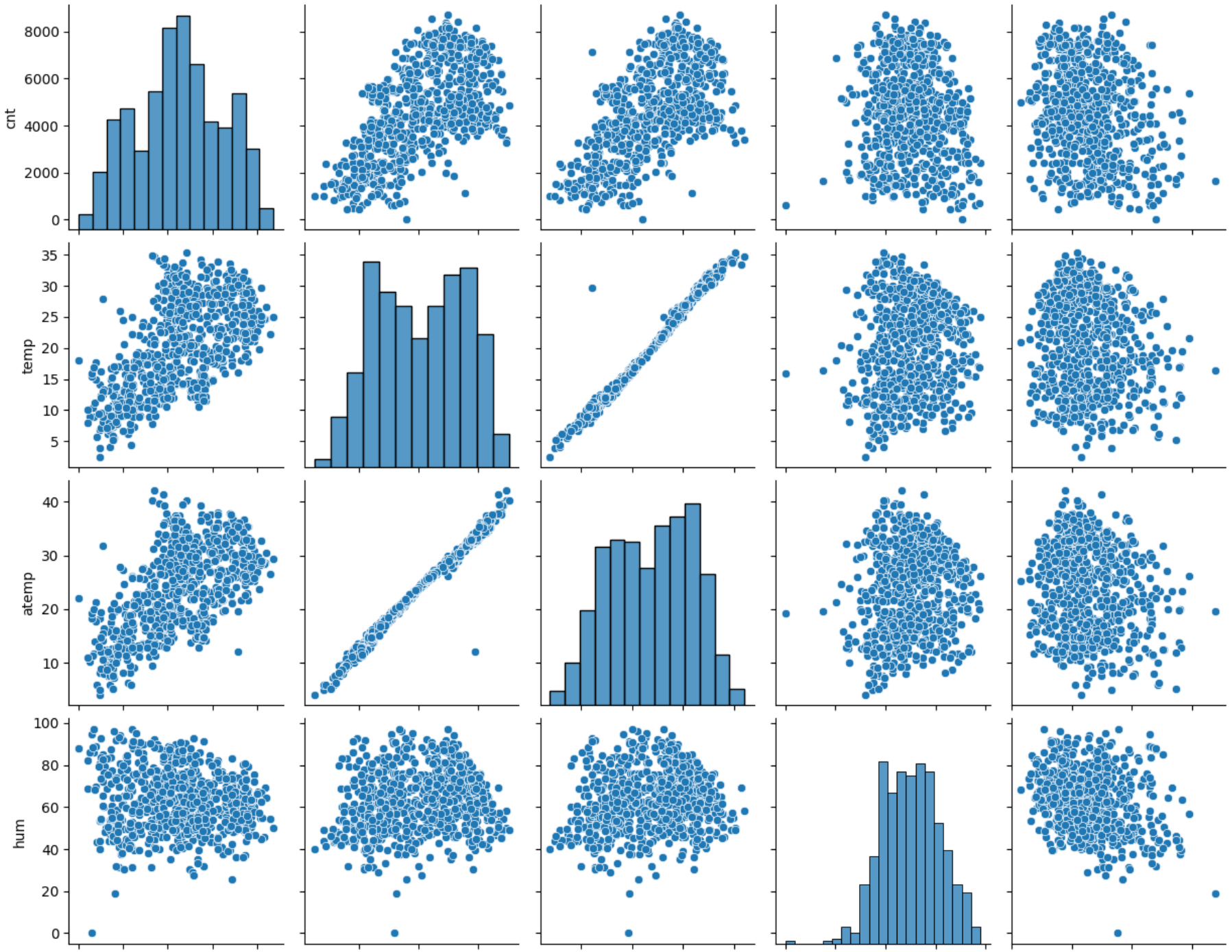
Out[20]:

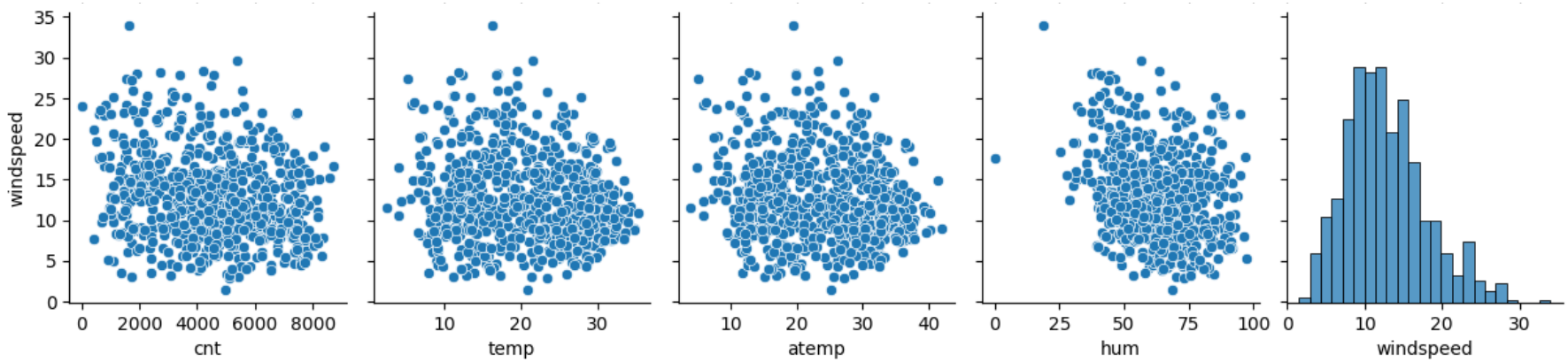
	season	yr	mnth	holiday	weekday	workingday	weathersit	temp	atemp	hum	windspeed	registered	cnt
0	spring	0	jan	0	mon	1	moderate	14.110847	18.18125	80.5833	10.749882	654	985
1	spring	0	jan	0	tue	1	moderate	14.902598	17.68695	69.6087	16.652113	670	801
2	spring	0	jan	0	wed	1	good	8.050924	9.47025	43.7273	16.636703	1229	1349
3	spring	0	jan	0	thu	1	good	8.200000	10.60610	59.0435	10.739832	1454	1562
4	spring	0	jan	0	fri	1	good	9.305237	11.46350	43.6957	12.522300	1518	1600

Drawing pairplots to check for linear relationship

In [21]:
```
plt.figure(figsize = (15,30))
sns.pairplot(data=bike,vars=['cnt', 'temp', 'atemp', 'hum','windspeed'])
plt.show()
```

```
<Figure size 1500x3000 with 0 Axes>
```


cnt
temp
atemp
hum
0
2000
4000
6000
8000
5
10
15
20
25
30
35
10
20
30
40
0
20
40
60
80
100

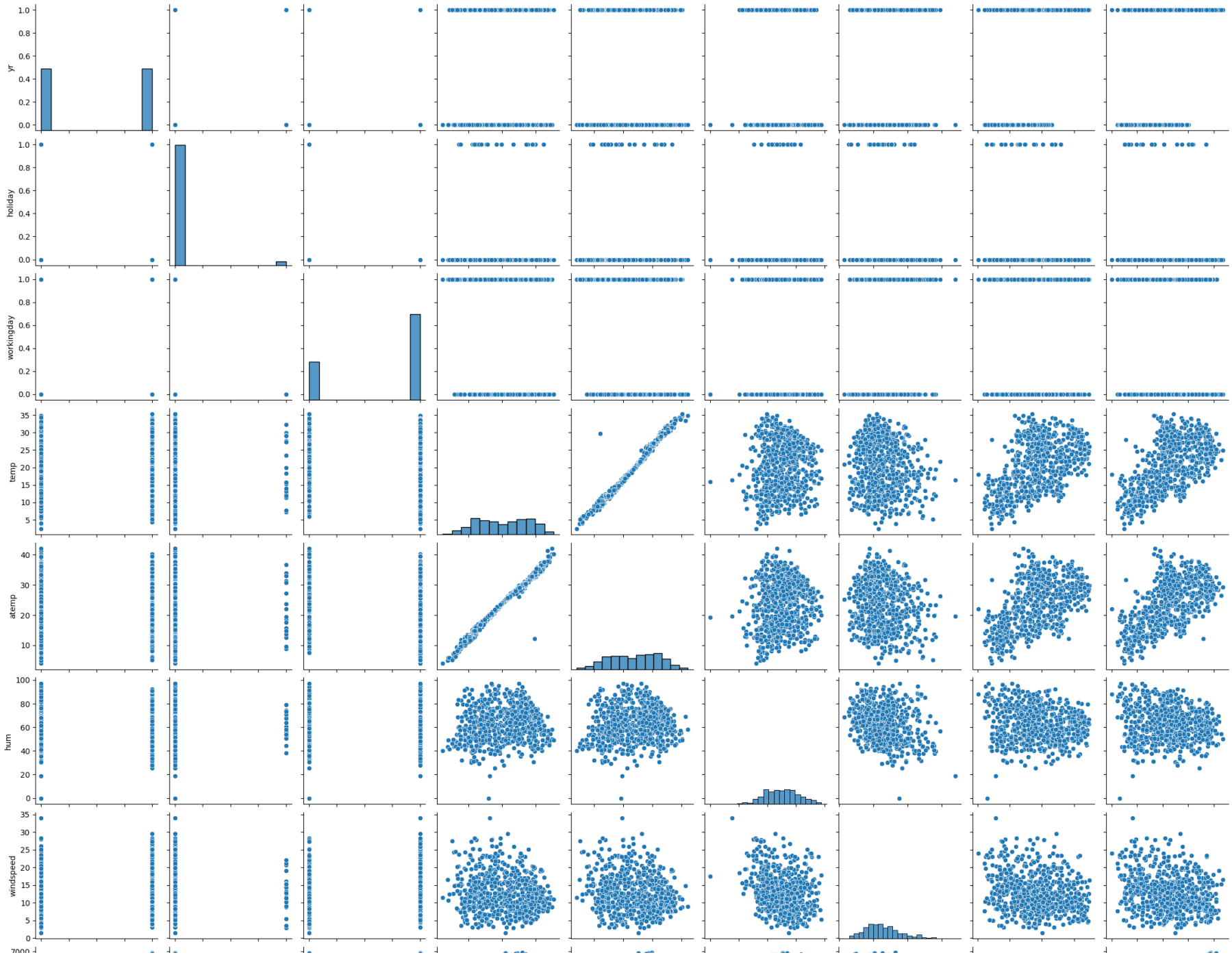


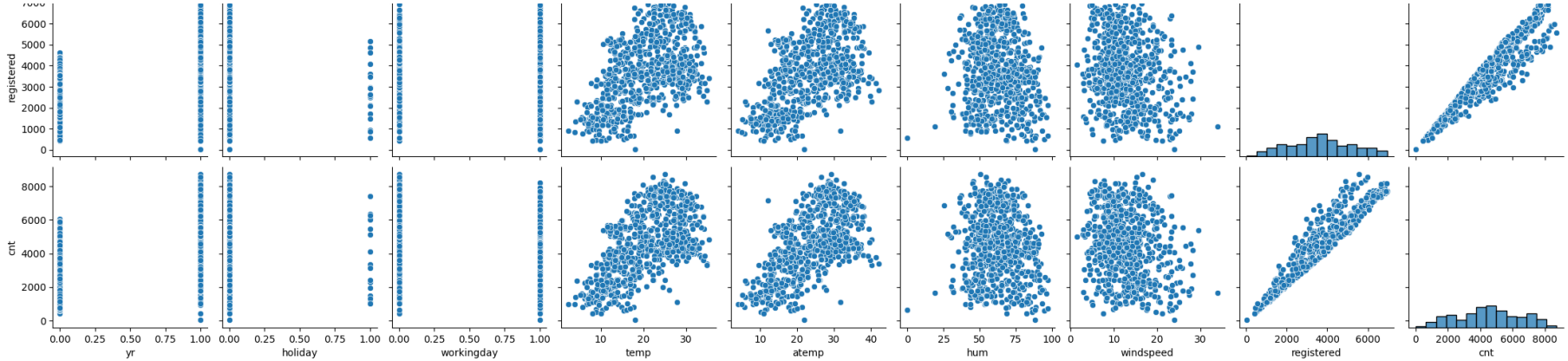
- So from the above plots we can clearly understand that temp and atemp are having high correlation
- And from the plots we can also say that there is a linear relationship between TEMP and ATEMP

Visualising the Data to find correlation from numerical variables

In [22]:
```
plt.figure(figsize=(20,15))
sns.pairplot(bike)
plt.show()
```

```
<Figure size 2000x1500 with 0 Axes>
```


yr
holiday
workingday
temp
atemp
hum
windspeed



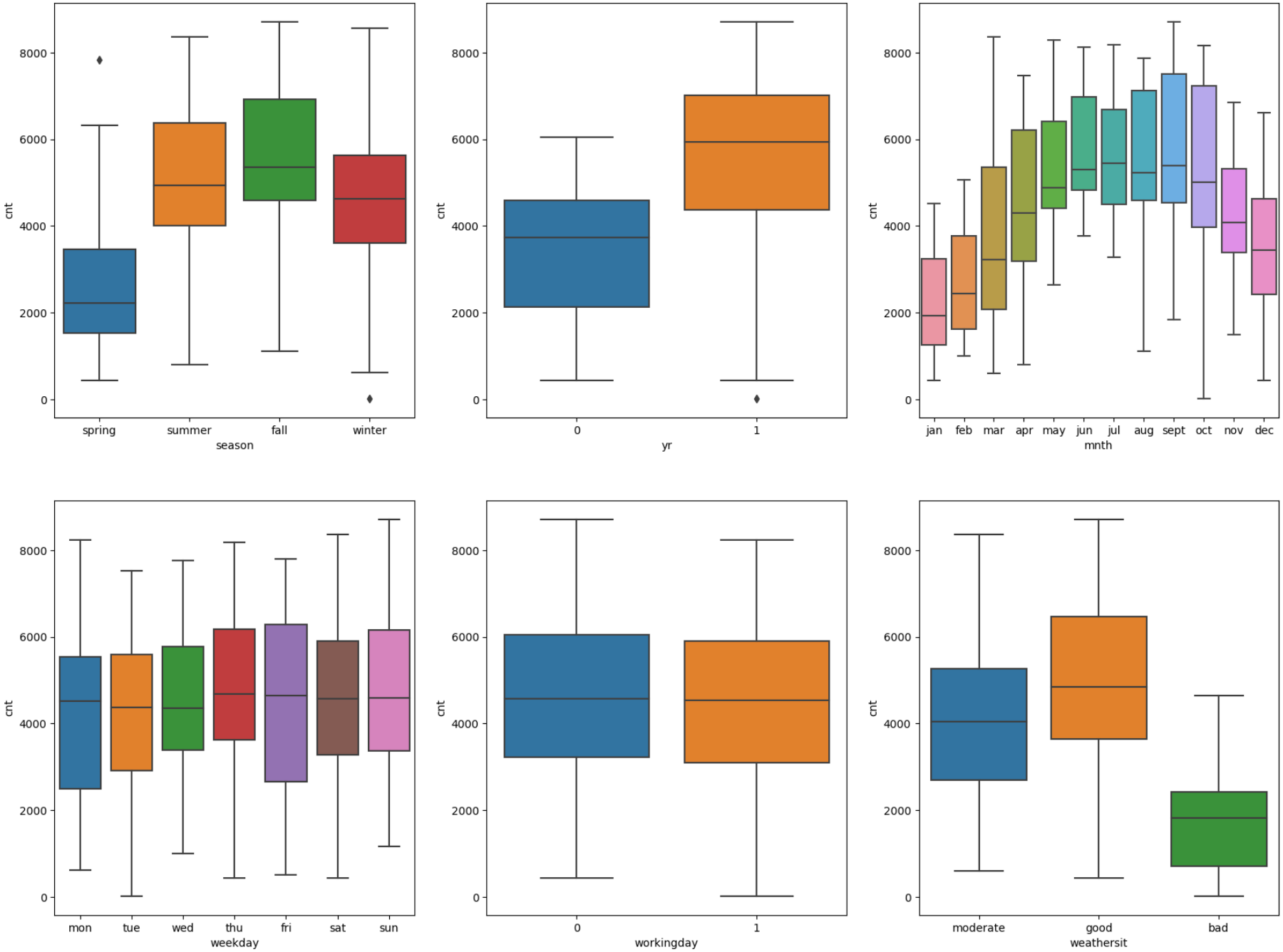
Heatmap for coorelation

In [23]:
```
plt.figure(figsize=(15,10))
sns.heatmap(bike.corr(),cmap="BuPu",annot=True)
plt.show()
```


yr 1 0.0082 -0.012 0.049 0.047 -0.11 -0.012 0.6 0.57
holiday 0.0082 1 -0.26 -0.029 -0.033 -0.016 0.0063 -0.11 -0.069
workingday -0.012 -0.26 1 0.002 0.011 0.054 -0.0025 0.0055 -0.028
temp 0.049 -0.029 0.002 1 0.99 0.13 -0.16 0.54 0.63
atemp 0.047 -0.033 0.011 0.99 1 0.14 -0.18 0.54 0.63
hum -0.11 -0.016 0.054 0.13 0.14 1 -0.25 -0.089 -0.099
windspeed -0.012 0.0063 -0.0025 -0.16 -0.18 -0.25 1 -0.22 -0.24
registered 0.6 -0.11 0.0055 0.54 0.54 -0.089 -0.22 1 0.95
cnt 0.57 -0.069 -0.028 0.63 0.63 -0.099 -0.24 0.95 1
yr holiday workingday temp atemp hum windspeed registered cnt
1.0 0.8 0.6 0.4 0.2 0.0 −0.2

In [24]:
```
#visualising the categorical variables

plt.figure(figsize=(20,15))

plt.subplot(2,3,1)
sns.boxplot(x='season',y='cnt',data=bike)

plt.subplot(2,3,2)
sns.boxplot(x='yr',y='cnt',data=bike)

plt.subplot(2,3,3)
sns.boxplot(x='mnth',y='cnt',data=bike)

plt.subplot(2,3,4)
sns.boxplot(x='weekday',y='cnt',data=bike)

plt.subplot(2,3,5)
sns.boxplot(x='workingday',y='cnt',data=bike)

plt.subplot(2,3,6)
sns.boxplot(x='weathersit',y='cnt',data=bike)
```

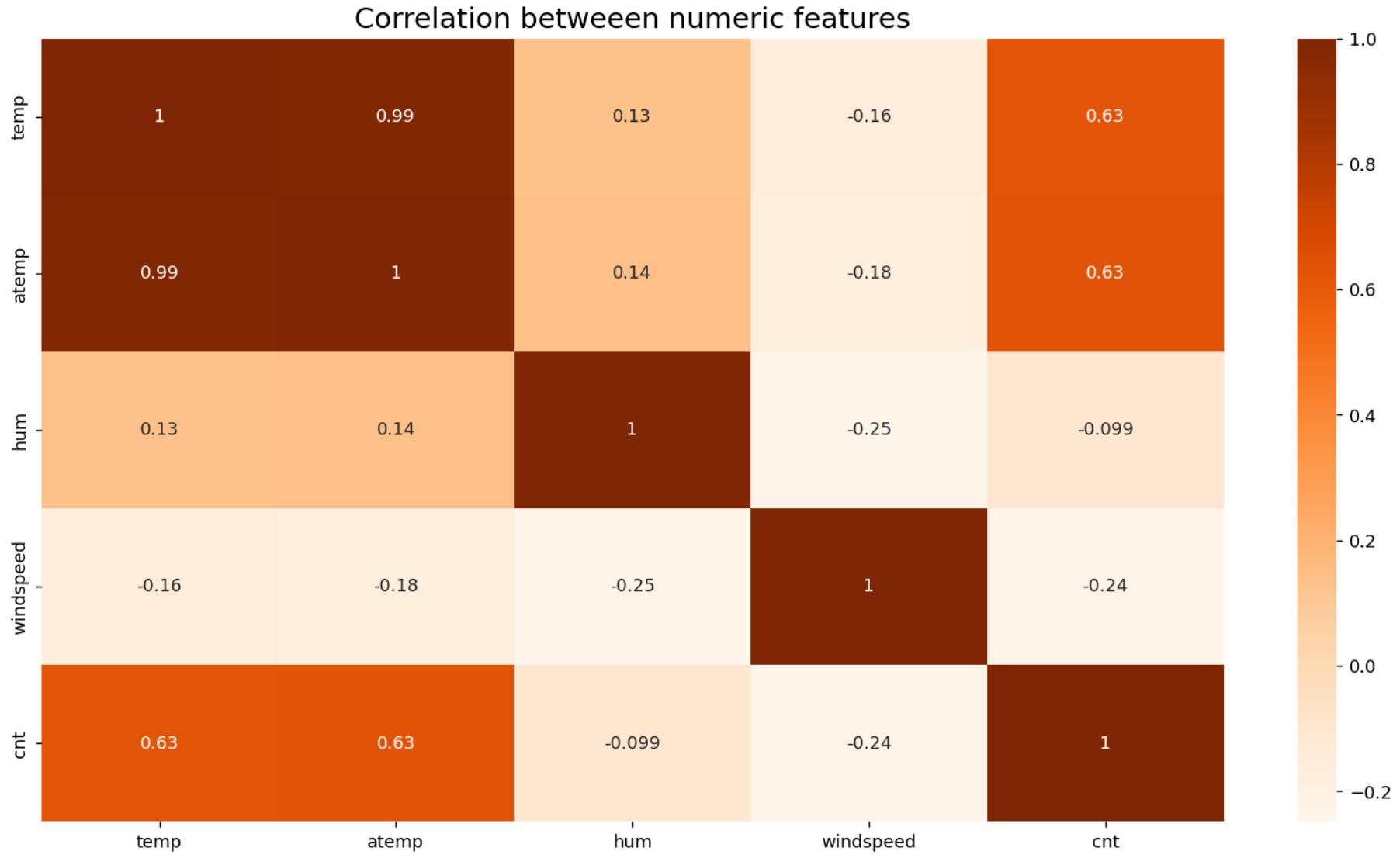
Out[24]:
```
<Axes: xlabel='weathersit', ylabel='cnt'>
```


cnt
season
spring
summer
fall
winter
yr
0
1
mnth
jan
feb
mar
apr
may
jun
jul
aug
sept
oct
nov
dec
weekday
mon
tue
wed
thu
fri
sat
sun
workingday
weathersit
moderate
good
bad

Conslusions:

- The graph clearly shows the qualitative distributions of the data, now if the model suggests the important predictors, using these graphs we can be more confident about the predictions of the model.
- For the variable season, we can clearly see that the category 3 : Fall, has the highest median, which shows that the demand was high during this season. It is least for 1: spring .
- The year 2019 had a higher count of users as compared to the year 2018
- The bike demand is almost constant throughout the week.
- The count of total users is in between 4000 to 6000 (~5500) during clear weather
- The count is highest in the month of August
- The count of users is less during the holidays

In [25]:
```
num_features = ["temp","atemp","hum","windspeed","cnt"]
plt.figure(figsize=(15,8),dpi=130)
plt.title("Correlation betweeen numeric features",fontsize=16)
sns.heatmap(bike[num_features].corr(),annot= True,cmap="Oranges")
plt.show()
```



In [26]:
```
bike.describe()
```

Out[26]:

	yr	holiday	workingday	temp	atemp	hum	windspeed	registered	cnt
count	730.000000	730.000000	730.000000	730.000000	730.000000	730.000000	730.000000	730.000000	730.000000
mean	0.500000	0.028767	0.690411	20.319259	23.726322	62.765175	12.763620	3658.757534	4508.006849
std	0.500343	0.167266	0.462641	7.506729	8.150308	14.237589	5.195841	1559.758728	1936.011647
min	0.000000	0.000000	0.000000	2.424346	3.953480	0.000000	1.500244	20.000000	22.000000
25%	0.000000	0.000000	0.000000	13.811885	16.889713	52.000000	9.041650	2502.250000	3169.750000
50%	0.500000	0.000000	1.000000	20.465826	24.368225	62.625000	12.125325	3664.500000	4548.500000
75%	1.000000	0.000000	1.000000	26.880615	30.445775	72.989575	15.625589	4783.250000	5966.000000
max	1.000000	1.000000	1.000000	35.328347	42.044800	97.250000	34.000021	6946.000000	8714.000000

Data preparation for linear regression:

Creating dummy variables for categorical variables:

In [27]:
```
bike = pd.get_dummies(data=bike,columns=["season","mnth","weekday"],drop_first=True)
bike = pd.get_dummies(data=bike,columns=["weathersit"])
```

In [28]:
```
bike.columns
```

Out[28]:
```
Index(['yr', 'holiday', 'workingday', 'temp', 'atemp', 'hum', 'windspeed',
       'registered', 'cnt', 'season_spring', 'season_summer', 'season_winter',
       'mnth_aug', 'mnth_dec', 'mnth_feb', 'mnth_jan', 'mnth_jul', 'mnth_jun',
       'mnth_mar', 'mnth_may', 'mnth_nov', 'mnth_oct', 'mnth_sept',
       'weekday_mon', 'weekday_sat', 'weekday_sun', 'weekday_thu',
       'weekday_tue', 'weekday_wed', 'weathersit_bad', 'weathersit_good',
       'weathersit_moderate'],
      dtype='object')
```

In [29]:
```
bike.head()
```

Out[29]:

	yr	holiday	workingday	temp	atemp	hum	windspeed	registered	cnt	season_spring	...	mnth_sept	weekday_mon	weekday_sat	w
0	0	0	1	14.110847	18.18125	80.5833	10.749882	654	985	1	...	0	1	0	
1	0	0	1	14.902598	17.68695	69.6087	16.652113	670	801	1	...	0	0	0	
2	0	0	1	8.050924	9.47025	43.7273	16.636703	1229	1349	1	...	0	0	0	
3	0	0	1	8.200000	10.60610	59.0435	10.739832	1454	1562	1	...	0	0	0	
4	0	0	1	9.305237	11.46350	43.6957	12.522300	1518	1600	1	...	0	0	0	

5 rows × 32 columns

Splitting Data into train and test data:

In [30]:
```
bike.shape
```

Out[30]:
```
(730, 32)
```

In [31]:
```
#y to contain only target variable
y=bike.pop('cnt')

#X is all remainign variable also our independent variables
X=bike

#Train Test split with 70:30 ratio
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.3, random_state=42)
```

In [32]:
```
X.head()
```

Out[32]:

	yr	holiday	workingday	temp	atemp	hum	windspeed	registered	season_spring	season_summer	...	mnth_sept	weekday_mon	weel
0	0	0	1	14.110847	18.18125	80.5833	10.749882	654	1	0	...	0	1	
1	0	0	1	14.902598	17.68695	69.6087	16.652113	670	1	0	...	0	0	
2	0	0	1	8.050924	9.47025	43.7273	16.636703	1229	1	0	...	0	0	
3	0	0	1	8.200000	10.60610	59.0435	10.739832	1454	1	0	...	0	0	
4	0	0	1	9.305237	11.46350	43.6957	12.522300	1518	1	0	...	0	0	

5 rows × 31 columns

In [33]:
```
print(X_train.shape)
print(X_test.shape)
```

```
(511, 31)
(219, 31)
```

In [34]:
```
num_vars = ['temp','atemp','hum','windspeed']

#Use Normalized scaler to scale
scaler = MinMaxScaler()

#Fit and transform training set only
X_train[num_vars] = scaler.fit_transform(X_train[num_vars])
```

In [35]:
```
X_train.describe()
```

Out[35]:

	yr	holiday	workingday	temp	atemp	hum	windspeed	registered	season_spring	season_summer	...	mnth_se
count	511.000000	511.000000	511.000000	511.000000	511.000000	511.000000	511.000000	511.000000	511.000000	511.000000	...	511.000
mean	0.520548	0.027397	0.698630	0.550874	0.526518	0.654174	0.414873	3694.350294	0.230920	0.258317	...	0.090
std	0.500067	0.163398	0.459303	0.227231	0.212976	0.143648	0.185619	1573.398840	0.421834	0.438138	...	0.286
min	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	416.000000	0.000000	0.000000	...	0.000
25%	0.000000	0.000000	0.000000	0.354221	0.353207	0.544598	0.281934	2628.000000	0.000000	0.000000	...	0.000
50%	1.000000	0.000000	1.000000	0.575304	0.563297	0.659940	0.384732	3729.000000	0.000000	0.000000	...	0.000
75%	1.000000	0.000000	1.000000	0.745079	0.698389	0.756977	0.526481	4804.000000	0.000000	1.000000	...	0.000
max	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	6946.000000	1.000000	1.000000	...	1.000

8 rows × 31 columns

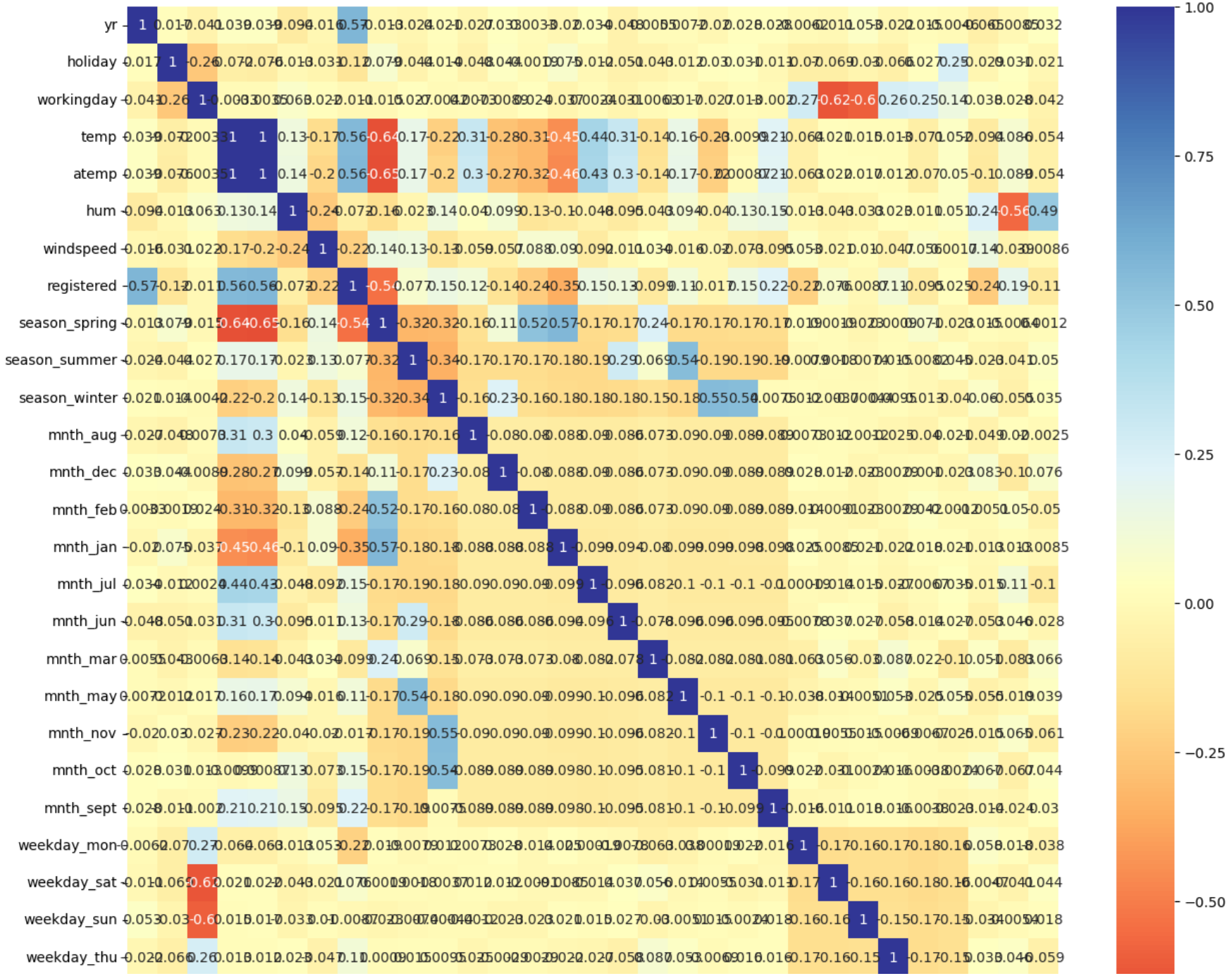
In [36]:
```
X_train.head()
```

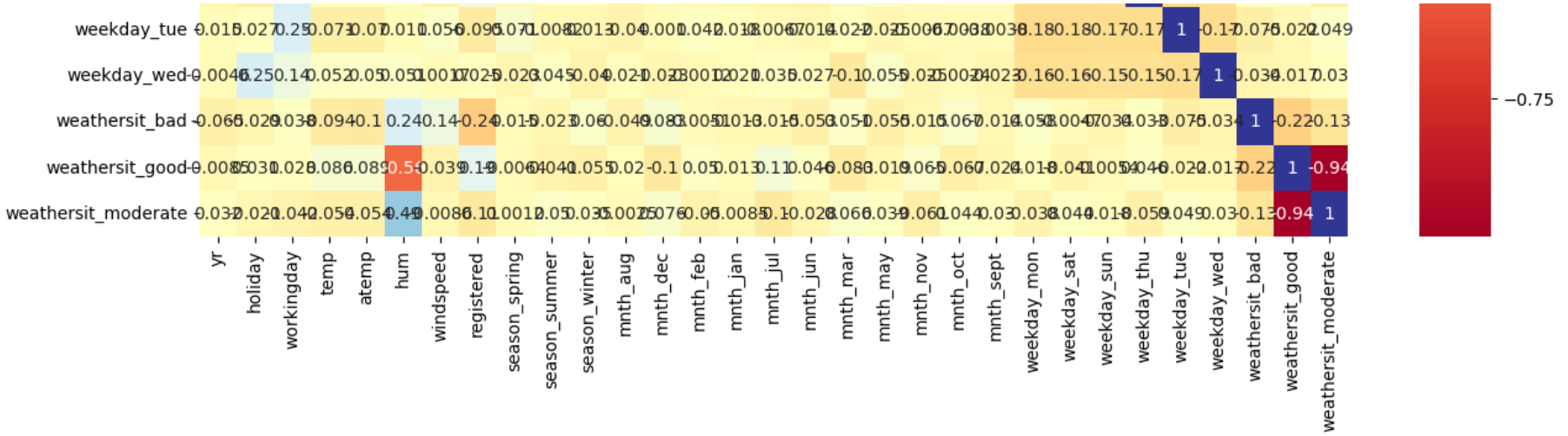
Out[36]:

	yr	holiday	workingday	temp	atemp	hum	windspeed	registered	season_spring	season_summer	...	mnth_sept	weekday_mon	we
683	1	0	1	0.327133	0.322150	0.639330	0.327101	5125	0	0	...	0	0	
645	1	1	0	0.403972	0.404998	0.731215	0.419004	4604	0	0	...	0	0	
163	0	0	1	0.717562	0.685963	0.509660	0.708724	4157	0	1	...	0	0	
360	0	0	1	0.331287	0.326273	0.785745	0.415925	1059	1	0	...	0	0	
640	1	0	1	0.745598	0.682653	0.817947	0.110593	6844	0	0	...	0	0	

5 rows × 31 columns

In [37]:
```
plt.figure(figsize = (15, 15))  #Checking if the variables are highly correlated
sns.heatmap(X_train.corr(), annot = True, cmap='RdYlBu')
plt.show()
```


yr
holiday
workingday
temp
atemp
hum
windspeed
registered
season_spring
season_summer
season_winter
mnth_aug
mnth_dec
mnth_feb
mnth_jan
mnth_jul
mnth_jun
mnth_mar
mnth_may
mnth_nov
mnth_oct
mnth_sept
weekday_mon
weekday_sat
weekday_sun
weekday_thu
1.00
0.75
0.50
0.25
0.00
−0.25
−0.50



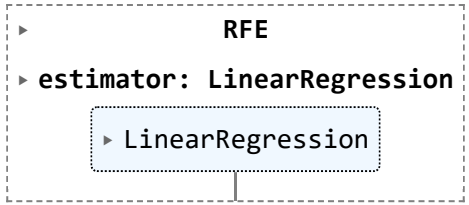
Build model using RFE

In [38]:
```
lr = LinearRegression()
lr.fit(X_train,y_train)
```

Out[38]:
```
▾ LinearRegression

LinearRegression()
```

In [39]:
```
# Create the estimator (Linear Regression in this case)
lr = LinearRegression()

# Create the RFE object with the estimator and the number of features to select (15 in this case)
rfe = RFE(estimator=lr, n_features_to_select=15)

# Fit the RFE object to the training data
rfe.fit(X_train, y_train)
```

Out[39]:



In [40]:
```
list(zip(X_train.columns,rfe.support_,rfe.ranking_))
```

Out[40]:
```
[('yr', True, 1),
 ('holiday', True, 1),
 ('workingday', False, 13),
 ('temp', True, 1),
 ('atemp', True, 1),
 ('hum', True, 1),
 ('windspeed', True, 1),
 ('registered', False, 17),
 ('season_spring', True, 1),
 ('season_summer', False, 8),
 ('season_winter', True, 1),
 ('mnth_aug', False, 12),
 ('mnth_dec', True, 1),
 ('mnth_feb', False, 9),
 ('mnth_jan', False, 15),
 ('mnth_jul', True, 1),
 ('mnth_jun', False, 11),
 ('mnth_mar', True, 1),
 ('mnth_may', False, 2),
 ('mnth_nov', True, 1),
 ('mnth_oct', False, 4),
 ('mnth_sept', True, 1),
 ('weekday_mon', False, 5),
 ('weekday_sat', False, 14),
 ('weekday_sun', False, 10),
 ('weekday_thu', False, 16),
 ('weekday_tue', False, 7),
 ('weekday_wed', False, 6),
 ('weathersit_bad', True, 1),
 ('weathersit_good', True, 1),
 ('weathersit_moderate', False, 3)]
```

In [41]:
```
def build_model(cols):
    X_train_sm = sm.add_constant(X_train[cols])
    lm = sm.OLS(y_train, X_train_sm).fit()
    print(lm.summary())
    return lm
```

In [42]:
```
def get_vif(cols):
    df1 = X_train[cols]
    vif = pd.DataFrame()
    vif['Features'] = df1.columns
    vif['VIF'] = [variance_inflation_factor(df1.values, i) for i in range(df1.shape[1])]
    vif['VIF'] = round(vif['VIF'],2)
    print(vif.sort_values(by='VIF',ascending=False))
```

In [43]:
```
#Print Columns selected by RFE. We will manually eliminate for these columns
X_train.columns[rfe.support_]
```

Out[43]:
```
Index(['yr', 'holiday', 'temp', 'atemp', 'hum', 'windspeed', 'season_spring',
       'season_winter', 'mnth_dec', 'mnth_jul', 'mnth_mar', 'mnth_nov',
       'mnth_sept', 'weathersit_bad', 'weathersit_good'],
      dtype='object')
```

In [44]:
```
# Features not selected by RFE
X_train.columns[~rfe.support_]
```

Out[44]:
```
Index(['workingday', 'registered', 'season_summer', 'mnth_aug', 'mnth_feb',
       'mnth_jan', 'mnth_jun', 'mnth_may', 'mnth_oct', 'weekday_mon',
       'weekday_sat', 'weekday_sun', 'weekday_thu', 'weekday_tue',
       'weekday_wed', 'weathersit_moderate'],
      dtype='object')
```

In [45]:
```
# Taking 15 columns supported by RFE for regression
X_train_rfe = X_train[['yr', 'holiday', 'workingday', 'temp', 'hum', 'windspeed', 'season_spring',
       'season_summer', 'season_winter', 'mnth_jan', 'mnth_jul', 'mnth_sept', 'weekday_sat',
       'weathersit_bad', 'weathersit_moderate']]
```

In [46]:
```
X_train_rfe.shape
```

Out[46]: (511, 15)

Model 1

In [47]:
```
#Selected columns for Model 1 - all columns selected by RFE
cols = ['yr', 'holiday', 'workingday', 'temp', 'hum', 'windspeed', 'season_spring',
       'season_summer', 'season_winter', 'mnth_jan', 'mnth_jul', 'mnth_sept', 'weekday_sat',
       'weathersit_bad', 'weathersit_moderate']

build_model(cols)
get_vif(cols)
```

```
                            OLS Regression Results
==============================================================================
Dep. Variable:                    cnt   R-squared:                       0.833
Model:                            OLS   Adj. R-squared:                  0.828
Method:                 Least Squares   F-statistic:                     165.0
Date:                Tue, 01 Aug 2023   Prob (F-statistic):          1.70e-181
Time:                        13:30:57   Log-Likelihood:                -4134.9
No. Observations:                 511   AIC:                             8302.
Df Residuals:                     495   BIC:                             8370.
Df Model:                          15
Covariance Type:            nonrobust
==========================================================================================
                             coef    std err          t      P>|t|      [0.025      0.975]
------------------------------------------------------------------------------------------
const                   2809.2461    346.564      8.106      0.000    2128.328    3490.164
yr                      1955.0549     72.324     27.032      0.000    1812.955    2097.155
holiday                 -839.6361    237.519     -3.535      0.000   -1306.306    -372.966
workingday              -202.7038    107.596     -1.884      0.060    -414.104       8.697
temp                    4233.9838    303.827     13.936      0.000    3637.034    4830.934
hum                    -1378.2895    339.821     -4.056      0.000   -2045.959    -710.620
windspeed              -1072.6371    210.709     -5.091      0.000   -1486.631    -658.643
season_spring           -795.1544    191.300     -4.157      0.000   -1171.014    -419.294
season_summer            380.6325    133.154      2.859      0.004     119.015     642.250
season_winter            767.1240    151.761      5.055      0.000     468.950    1065.298
mnth_jan                 -59.8050    156.115     -0.383      0.702    -366.534     246.924
mnth_jul                -367.2369    159.579     -2.301      0.022    -680.772     -53.702
mnth_sept                666.1046    144.332      4.615      0.000     382.526     949.684
weekday_sat              -72.5004    136.440     -0.531      0.595    -340.574     195.573
weathersit_bad         -1761.9834    238.533     -7.387      0.000   -2230.645   -1293.322
weathersit_moderate     -464.2948     93.779     -4.951      0.000    -648.550    -280.040
==============================================================================
Omnibus:                       65.698   Durbin-Watson:                   2.095
Prob(Omnibus):                  0.000   Jarque-Bera (JB):              125.994
Skew:                          -0.750   Prob(JB):                     4.37e-28
Kurtosis:                       4.914   Cond. No.                         23.6
==============================================================================

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

```
               Features    VIF
```

4	hum	28.53
3	temp	17.32
2	workingday	6.06
5	windspeed	5.73
6	season_spring	4.45
8	season_winter	3.24
7	season_summer	3.01
14	weathersit_moderate	2.26
0	yr	2.10
12	weekday_sat	2.01
10	mnth_jul	1.83
9	mnth_jan	1.67
11	mnth_sept	1.47
13	weathersit_bad	1.27
1	holiday	1.21

Model 2

In [48]:
```
# Dropping the variable mnth_jan as it has negative coefficient and is insignificant as it has high p-value
cols = ['yr', 'holiday', 'workingday', 'temp', 'hum', 'windspeed', 'season_spring',
       'season_summer', 'season_winter', 'mnth_jul', 'mnth_sept', 'weekday_sat',
       'weathersit_bad', 'weathersit_moderate']
build_model(cols)
get_vif(cols)
```

```
                            OLS Regression Results
==============================================================================
Dep. Variable:                    cnt   R-squared:                       0.833
Model:                            OLS   Adj. R-squared:                  0.829
Method:                 Least Squares   F-statistic:                     177.1
Date:                Tue, 01 Aug 2023   Prob (F-statistic):          1.35e-182
Time:                        13:31:02   Log-Likelihood:                -4135.0
No. Observations:                 511   AIC:                             8300.
Df Residuals:                     496   BIC:                             8363.
Df Model:                          14
Covariance Type:            nonrobust
===================================================================================
                      coef    std err          t      P>|t|      [0.025      0.975]
-----------------------------------------------------------------------------------
const               2792.2799    343.427      8.131      0.000    2117.529    3467.030
yr                  1955.1531     72.261     27.057      0.000    1813.177    2097.129
holiday             -840.8434    237.294     -3.543      0.000   -1307.069    -374.618
workingday          -200.8684    107.396     -1.870      0.062    -411.877      10.140
temp                4258.5248    296.741     14.351      0.000    3675.500    4841.549
hum                -1383.9501    339.207     -4.080      0.000   -2050.411    -717.489
windspeed          -1073.0058    210.525     -5.097      0.000   -1486.637    -659.375
season_spring       -806.5438    188.813     -4.272      0.000   -1177.516    -435.572
season_summer        384.0827    132.735      2.894      0.004     123.290     644.875
season_winter        774.4415    150.424      5.148      0.000     478.895    1069.988
mnth_jul            -369.6007    159.322     -2.320      0.021    -682.630     -56.571
mnth_sept            665.9059    144.207      4.618      0.000     382.574     949.238
weekday_sat          -71.1261    136.276     -0.522      0.602    -338.875     196.623
weathersit_bad     -1756.5848    237.911     -7.383      0.000   -2224.023   -1289.147
weathersit_moderate  -462.7865     93.616     -4.943      0.000    -646.719    -278.854
==============================================================================
Omnibus:                       65.111   Durbin-Watson:                   2.095
Prob(Omnibus):                  0.000   Jarque-Bera (JB):              124.284
Skew:                          -0.746   Prob(JB):                     1.03e-27
Kurtosis:                       4.900   Cond. No.                         23.5
==============================================================================

Notes:
[1] Standard Errors assume that the covariance matrix of the errors is correctly specified.
              Features    VIF
4                  hum  27.99
```

3	temp	16.83
2	workingday	6.06
5	windspeed	5.70
6	season_spring	4.10
8	season_winter	3.23
7	season_summer	3.01
13	weathersit_moderate	2.25
0	yr	2.10
11	weekday_sat	2.01
9	mnth_jul	1.82
10	mnth_sept	1.47
12	weathersit_bad	1.26
1	holiday	1.21

Model-3

In [49]:
```
# Dropping the variable hum as it has negative coefficient and is insignificant as it has high p-value
cols = ['yr', 'holiday', 'workingday', 'temp', 'windspeed', 'season_spring',
       'season_summer', 'season_winter', 'mnth_jul', 'mnth_sept', 'weekday_sat',
       'weathersit_bad', 'weathersit_moderate']
build_model(cols)
get_vif(cols)
```

```
                            OLS Regression Results
==============================================================================
Dep. Variable:                    cnt   R-squared:                       0.828
Model:                            OLS   Adj. R-squared:                  0.823
Method:                 Least Squares   F-statistic:                     183.6
Date:                Tue, 01 Aug 2023   Prob (F-statistic):          3.37e-180
Time:                        13:31:06   Log-Likelihood:                -4143.4
No. Observations:                 511   AIC:                             8315.
Df Residuals:                     497   BIC:                             8374.
Df Model:                          13
Covariance Type:            nonrobust
==========================================================================================
                             coef    std err          t      P>|t|      [0.025      0.975]
------------------------------------------------------------------------------------------
const                  2004.0762    288.371      6.950      0.000    1437.500    2570.652
yr                     1995.7722     72.690     27.456      0.000    1852.954    2138.590
holiday                -873.2715    240.865     -3.626      0.000   -1346.511    -400.032
workingday             -226.4412    108.888     -2.080      0.038    -440.379     -12.504
temp                   4050.9306    296.912     13.644      0.000    3467.573    4634.289
windspeed              -816.0345    204.020     -4.000      0.000   -1216.883    -415.186
season_spring          -838.0753    191.601     -4.374      0.000   -1214.523    -461.627
season_summer           376.3936    134.795      2.792      0.005     111.556     641.231
season_winter           713.5094    152.019      4.694      0.000     414.831    1012.188
mnth_jul               -326.5137    161.455     -2.022      0.044    -643.731      -9.296
mnth_sept               610.3106    145.804      4.186      0.000     323.843     896.779
weekday_sat             -50.4406    138.309     -0.365      0.715    -322.182     221.301
weathersit_bad      -2161.8583    219.559     -9.846      0.000   -2593.236   -1730.481
weathersit_moderate  -685.8621     77.177     -8.887      0.000    -837.495    -534.229
==============================================================================
Omnibus:                       60.891   Durbin-Watson:                   2.127
Prob(Omnibus):                  0.000   Jarque-Bera (JB):              113.587
Skew:                          -0.712   Prob(JB):                     2.16e-25
Kurtosis:                       4.819   Cond. No.                         20.3
==============================================================================

Notes:
[1] Standard Errors assume that the covariance matrix of the errors is correctly specified.
               Features   VIF
3                  temp  8.10
2            workingday  5.73
```

```
4             windspeed  5.69
5         season_spring  2.69
6         season_summer  2.69
7         season_winter  2.19
0                    yr  2.09
10          weekday_sat  1.99
8              mnth_jul  1.82
12  weathersit_moderate  1.57
9              mnth_sept  1.43
1                holiday  1.19
11       weathersit_bad  1.09
```

Model-4

In [50]:
```
# Dropping the variable holiday as it has negative coefficient and is insignificant as it has high p-value
cols = ['yr', 'workingday', 'temp', 'windspeed', 'season_spring',
       'season_summer', 'season_winter', 'mnth_jul', 'mnth_sept', 'weekday_sat',
       'weathersit_bad', 'weathersit_moderate']
build_model(cols)
get_vif(cols)
```

```
                            OLS Regression Results
==============================================================================
Dep. Variable:                    cnt   R-squared:                       0.823
Model:                            OLS   Adj. R-squared:                  0.819
Method:                 Least Squares   F-statistic:                     193.1
Date:                Tue, 01 Aug 2023   Prob (F-statistic):          1.51e-178
Time:                        13:31:09   Log-Likelihood:                -4150.1
No. Observations:                 511   AIC:                             8326.
Df Residuals:                     498   BIC:                             8381.
Df Model:                          12
Covariance Type:            nonrobust
==========================================================================================
                             coef    std err          t      P>|t|      [0.025      0.975]
------------------------------------------------------------------------------------------
const                   1845.8415    288.504      6.398      0.000    1279.007    2412.676
yr                      1997.7265     73.569     27.154      0.000    1853.182    2142.271
workingday               -76.1270    101.906     -0.747      0.455    -276.346     124.093
temp                    4062.1670    300.495     13.518      0.000    3471.774    4652.560
windspeed               -786.5884    206.329     -3.812      0.000   -1191.972    -381.205
season_spring           -870.5177    193.712     -4.494      0.000   -1251.111    -489.925
season_summer            366.4796    136.400      2.687      0.007      98.489     634.471
season_winter            697.9586    153.800      4.538      0.000     395.782    1000.135
mnth_jul                -335.0219    163.394     -2.050      0.041    -656.049     -13.995
mnth_sept                606.0378    147.566      4.107      0.000     316.109     895.967
weekday_sat               99.5698    133.574      0.745      0.456    -162.869     362.008
weathersit_bad         -2148.1594    222.187     -9.668      0.000   -2584.699   -1711.620
weathersit_moderate     -676.7543     78.071     -8.668      0.000    -830.143    -523.365
==============================================================================
Omnibus:                       68.186   Durbin-Watson:                   2.118
Prob(Omnibus):                  0.000   Jarque-Bera (JB):              126.740
Skew:                          -0.787   Prob(JB):                     3.01e-28
Kurtosis:                       4.864   Cond. No.                         20.2
==============================================================================

Notes:
[1] Standard Errors assume that the covariance matrix of the errors is correctly specified.
               Features   VIF
2                  temp  7.79
3             windspeed  5.69
1            workingday  5.01
```

```
5         season_summer  2.66
4         season_spring  2.51
6         season_winter  2.10
0                    yr  2.09
9           weekday_sat  1.84
7              mnth_jul  1.82
11  weathersit_moderate  1.57
8              mnth_sept  1.42
10       weathersit_bad  1.09
```

Model-5

In [51]:
```
# Dropping the variable mnth_jul,temp as it has negative coefficient and is insignificant as it has high p-value
cols = ['yr', 'workingday', 'temp', 'windspeed', 'season_spring',
       'season_summer', 'season_winter', 'mnth_sept', 'weekday_sat',
       'weathersit_bad', 'weathersit_moderate']
build_model(cols)
get_vif(cols)
```

```
                            OLS Regression Results
==============================================================================
Dep. Variable:                    cnt   R-squared:                       0.822
Model:                            OLS   Adj. R-squared:                  0.818
Method:                 Least Squares   F-statistic:                     209.0
Date:                Tue, 01 Aug 2023   Prob (F-statistic):          8.20e-179
Time:                        13:31:13   Log-Likelihood:                -4152.2
No. Observations:                 511   AIC:                             8328.
Df Residuals:                     499   BIC:                             8379.
Df Model:                          11
Covariance Type:            nonrobust
==========================================================================================
                             coef    std err          t      P>|t|      [0.025      0.975]
------------------------------------------------------------------------------------------
const                   1779.2943    287.591      6.187      0.000    1214.255    2344.333
yr                      1992.6143     73.763     27.014      0.000    1847.691    2137.538
workingday               -74.8818    102.231     -0.732      0.464    -275.739     125.975
temp                    3958.5476    297.164     13.321      0.000    3374.701    4542.395
windspeed               -779.1681    206.959     -3.765      0.000   -1185.786    -372.550
season_spring           -778.9193    189.094     -4.119      0.000   -1150.438    -407.400
season_summer            491.7101    122.353      4.019      0.000     251.320     732.100
season_winter            798.5462    146.233      5.461      0.000     511.238    1085.855
mnth_sept                715.4337    138.024      5.183      0.000     444.253     986.614
weekday_sat              106.3578    133.961      0.794      0.428    -156.840     369.555
weathersit_bad         -2160.9326    222.811     -9.698      0.000   -2598.697   -1723.169
weathersit_moderate     -669.8623     78.249     -8.561      0.000    -823.600    -516.125
==============================================================================
Omnibus:                       70.746   Durbin-Watson:                   2.113
Prob(Omnibus):                  0.000   Jarque-Bera (JB):              130.655
Skew:                          -0.816   Prob(JB):                     4.25e-29
Kurtosis:                       4.864   Cond. No.                         20.2
==============================================================================

Notes:
[1] Standard Errors assume that the covariance matrix of the errors is correctly specified.
               Features   VIF
2                  temp  6.13
3             windspeed  5.68
1            workingday  5.01
4         season_spring  2.38
```

```
5         season_summer  2.10
0                    yr  2.09
6         season_winter  1.85
8           weekday_sat  1.84
10  weathersit_moderate  1.57
7             mnth_sept  1.25
9        weathersit_bad  1.09
```

Model-6

In [52]:
```
# Droping and replacing Variable as follows weekday_saturdat as high P-value, reconsidering holiday.
cols6 = ['yr','season_spring',
        'season_winter','season_summer','mnth_sept','holiday','workingday',
       'weathersit_bad', 'weathersit_moderate','temp']
build_model(cols6)
get_vif(cols6)
```

```
                            OLS Regression Results
==============================================================================
Dep. Variable:                    cnt   R-squared:                       0.821
Model:                            OLS   Adj. R-squared:                  0.817
Method:                 Least Squares   F-statistic:                     229.0
Date:                Tue, 01 Aug 2023   Prob (F-statistic):          1.70e-179
Time:                        13:31:18   Log-Likelihood:                -4153.4
No. Observations:                 511   AIC:                             8329.
Df Residuals:                     500   BIC:                             8375.
Df Model:                          10
Covariance Type:            nonrobust
==========================================================================================
                             coef    std err          t      P>|t|      [0.025      0.975]
------------------------------------------------------------------------------------------
const                  1500.5024    262.410      5.718      0.000     984.940    2016.065
yr                     1989.4538     73.778     26.965      0.000    1844.500    2134.408
season_spring          -744.1779    189.384     -3.929      0.000   -1116.264    -372.092
season_winter           865.5108    145.733      5.939      0.000     579.187    1151.834
season_summer           462.1289    122.140      3.784      0.000     222.159     702.099
mnth_sept               733.2217    138.143      5.308      0.000     461.809    1004.634
holiday                -825.2345    233.487     -3.534      0.000   -1283.970    -366.499
workingday             -204.1270     82.866     -2.463      0.014    -366.935     -41.319
weathersit_bad        -2289.0675    221.187    -10.349      0.000   -2723.638   -1854.497
weathersit_moderate    -678.4237     78.385     -8.655      0.000    -832.429    -524.419
temp                   4091.5171    295.064     13.867      0.000    3511.798    4671.236
==============================================================================
Omnibus:                       63.271   Durbin-Watson:                   2.151
Prob(Omnibus):                  0.000   Jarque-Bera (JB):              118.647
Skew:                          -0.734   Prob(JB):                     1.72e-26
Kurtosis:                       4.848   Cond. No.                         19.0
==============================================================================

Notes:
[1] Standard Errors assume that the covariance matrix of the errors is correctly specified.
              Features   VIF
9                 temp  3.98
6           workingday  3.35
0                   yr  2.08
3        season_summer  1.80
1        season_spring  1.62
```

```
2        season_winter  1.60
8  weathersit_moderate  1.56
4            mnth_sept  1.25
5              holiday  1.10
7       weathersit_bad  1.06
```

Here the VIF seems to be almost accepted. p-value for all the features is almost 0.0 and R2 is 0.8 Hence we finalize this model to use further

In [53]:
```
#Build a model with all columns to select features automatically
def build_model_sk(X,y):
    lr1 = LinearRegression()
    lr1.fit(X,y)
    return lr1
```

In [54]:
```
#Let us build the finalmodel using sklearn
#Build a model with above columns
lr = build_model_sk(X_train[cols6],y_train)
print(lr.intercept_,lr.coef_)
```

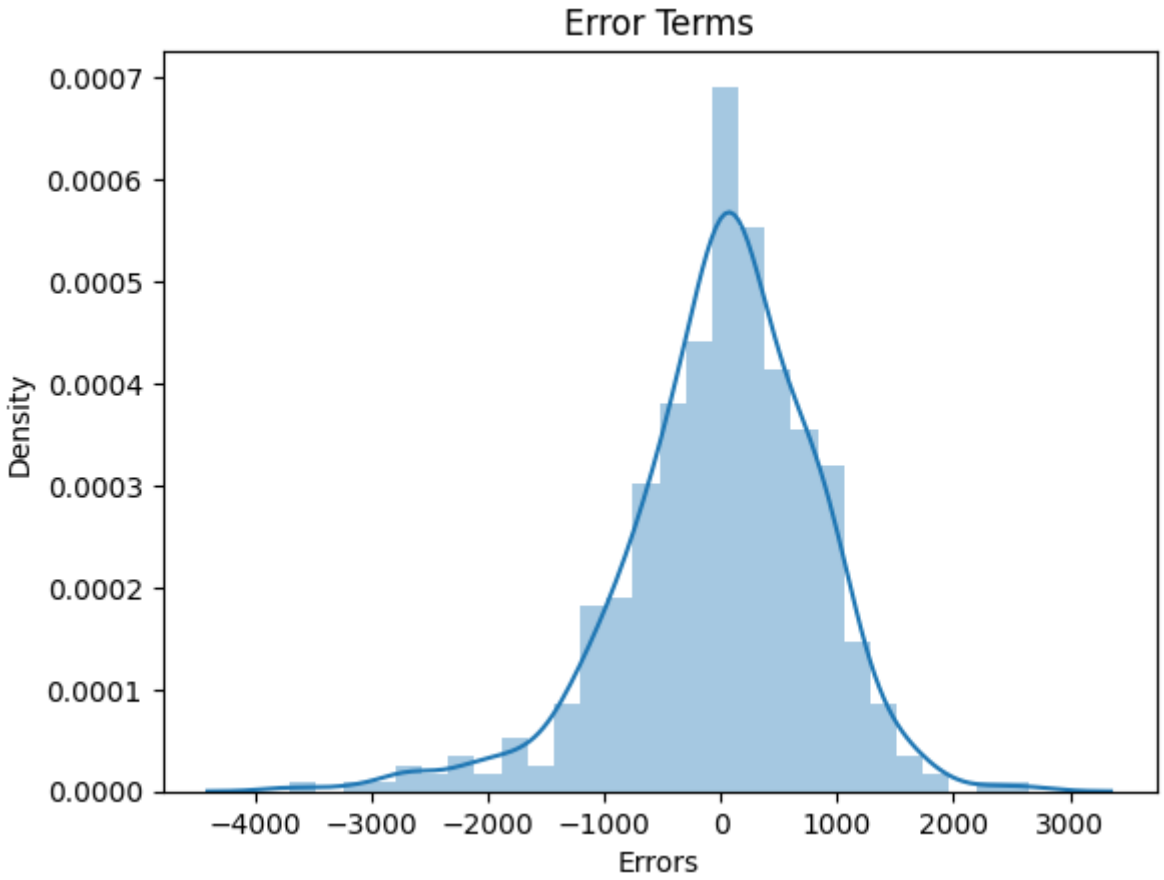
```
1500.5024389131136 [ 1989.45381428  -744.17786846   865.51084562   462.12893069
   733.22170346  -825.23454868  -204.12695705 -2289.06746494
  -678.42373184  4091.51711769]
```

Residual Analysis

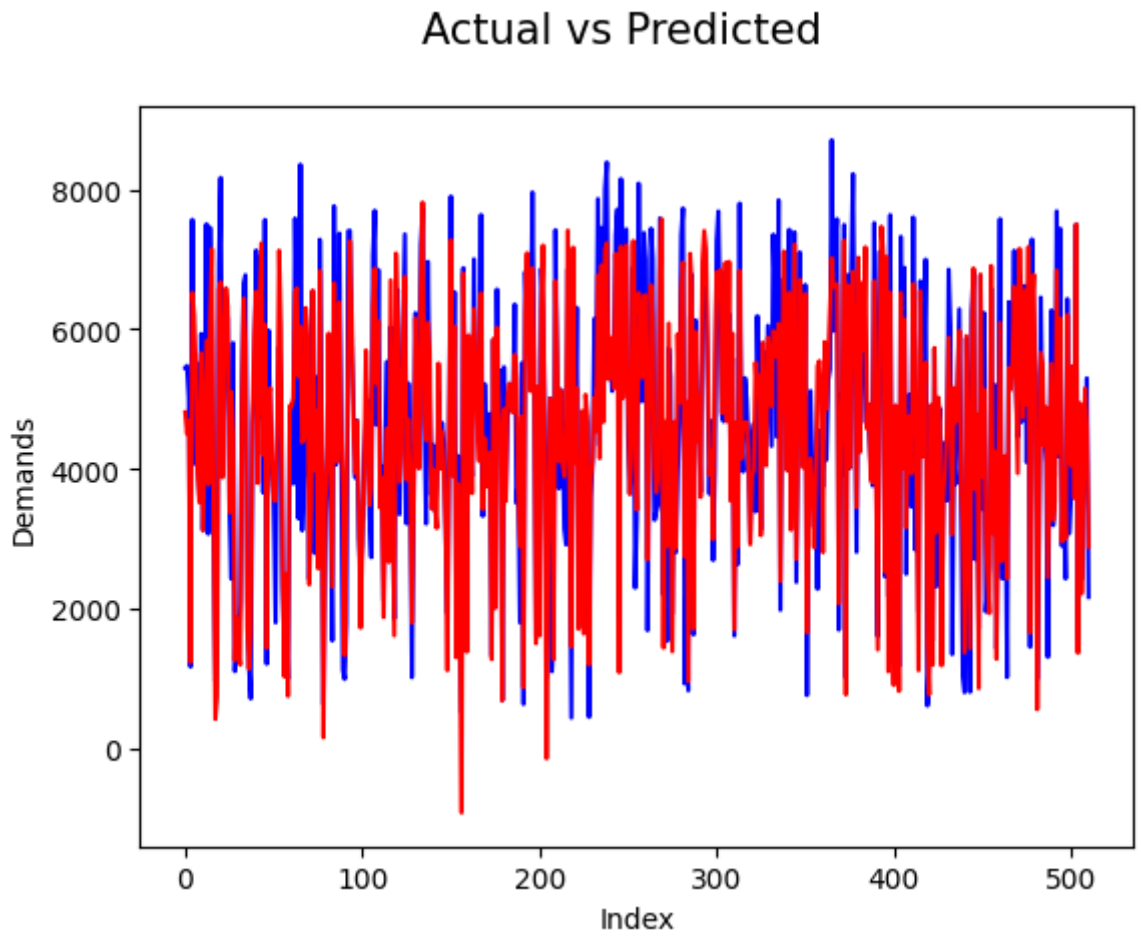
In [55]:
```
y_train_pred = lr.predict(X_train[cols6])
```

In [56]:
```
#Plot a histogram of the error terms
def plot_res_dist(act, pred):
    sns.distplot(act-pred)
    plt.title('Error Terms')
    plt.xlabel('Errors')
```

In [57]:
```
plot_res_dist(y_train, y_train_pred)
```



In [58]:
```
# Actual vs Predicted
c = [i for i in range(0,len(X_train),1)]
plt.plot(c,y_train, color="blue")
plt.plot(c,y_train_pred, color="red")
plt.suptitle('Actual vs Predicted', fontsize = 15)
plt.xlabel('Index')
plt.ylabel('Demands')
plt.show()
```



Model Predictions

R-Squared value for train predictions

In [59]:
```
#Print R-squared Value
r2_score(y_train,y_train_pred)
```

Out[59]:
```
0.8207866077552256
```

Prediction of values on test dataset

In [60]:
```
#Scale variables in X_test
num_vars = ['temp','atemp','hum','windspeed']

#Test data to be transformed only, no fitting
X_test[num_vars] = scaler.transform(X_test[num_vars])
```

In [63]:
```
cols6 = ['yr','season_spring',
        'season_winter','season_summer','mnth_sept','holiday','workingday',
       'weathersit_bad', 'weathersit_moderate','temp']

#Predicting test data values
y_test_pred = lr.predict(X_test[cols6])
```

R-Squared value for test predictions

In [64]:
```
# Find out the R squared value between test and predicted test data sets.
r2_score(y_test,y_test_pred)
```

Out[64]:
```
0.8141443198305731
```

Evaluating the model

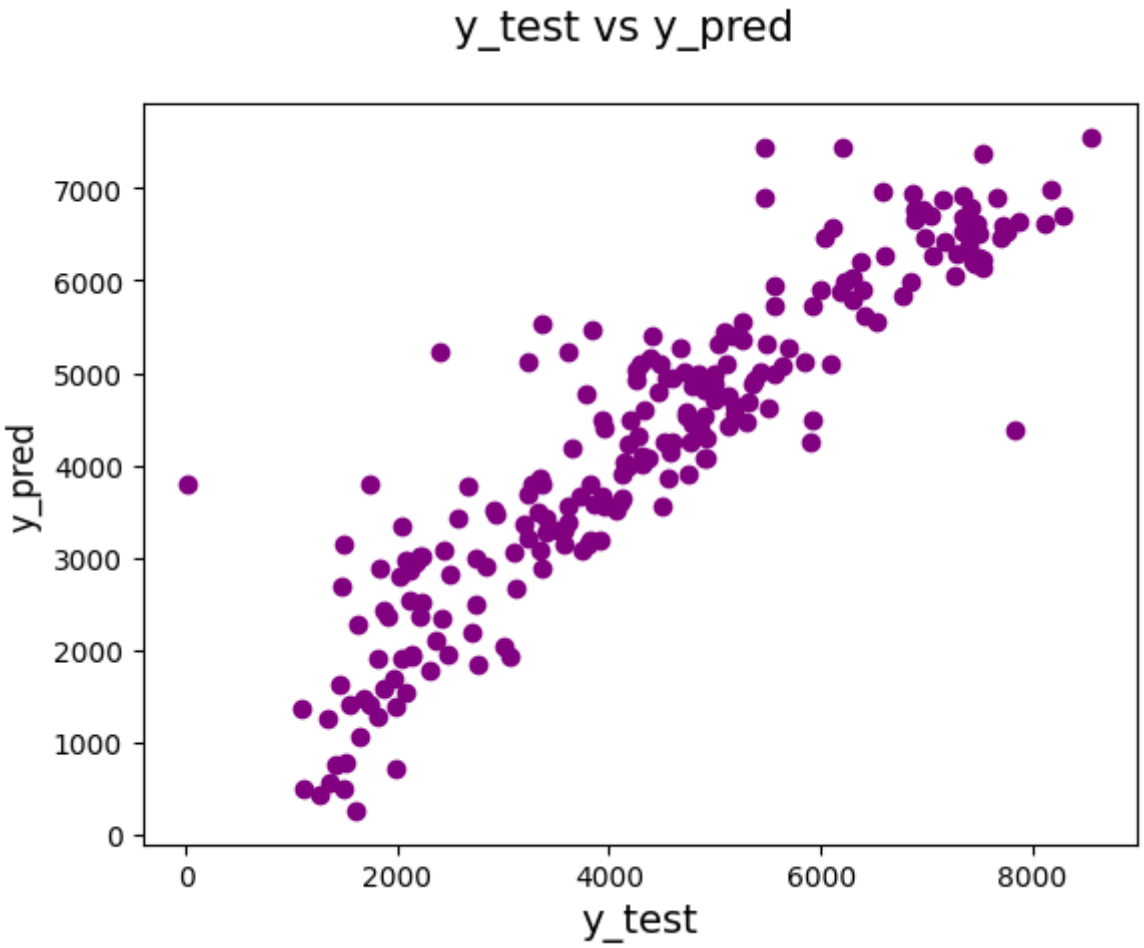
Evaluating the model based on r2_score(as mentioned in the problem statement as well)

In [65]:
```
y_pred=y_test_pred
r2_score(y_test, y_pred)
```

Out[65]:
```
0.8141443198305731
```

In [68]:
```
fig = plt.figure()
plt.scatter(y_test, y_pred,color='purple')
fig.suptitle('y_test vs y_pred', fontsize = 15)
plt.xlabel('y_test', fontsize = 14)
plt.ylabel('y_pred', fontsize = 12)
```

Out[68]: Text(0, 0.5, 'y_pred')



Let us rebuild the final model of manual + RFE approach using statsmodel to interpret it:

In [70]:
```
cols6 = ['yr','season_spring',
        'season_winter','season_summer','mnth_sept','holiday','workingday',
       'weathersit_bad', 'weathersit_moderate','temp']

lm = build_model(cols6)
```

```
                            OLS Regression Results
==============================================================================
Dep. Variable:                    cnt   R-squared:                       0.821
Model:                            OLS   Adj. R-squared:                  0.817
Method:                 Least Squares   F-statistic:                     229.0
Date:                Tue, 01 Aug 2023   Prob (F-statistic):          1.70e-179
Time:                        13:43:41   Log-Likelihood:                -4153.4
No. Observations:                 511   AIC:                             8329.
Df Residuals:                     500   BIC:                             8375.
Df Model:                          10
Covariance Type:            nonrobust
==========================================================================================
                             coef    std err          t      P>|t|      [0.025      0.975]
------------------------------------------------------------------------------------------
const                  1500.5024    262.410      5.718      0.000     984.940    2016.065
yr                     1989.4538     73.778     26.965      0.000    1844.500    2134.408
season_spring          -744.1779    189.384     -3.929      0.000   -1116.264    -372.092
season_winter           865.5108    145.733      5.939      0.000     579.187    1151.834
season_summer           462.1289    122.140      3.784      0.000     222.159     702.099
mnth_sept               733.2217    138.143      5.308      0.000     461.809    1004.634
holiday                -825.2345    233.487     -3.534      0.000   -1283.970    -366.499
workingday             -204.1270     82.866     -2.463      0.014    -366.935     -41.319
weathersit_bad        -2289.0675    221.187    -10.349      0.000   -2723.638   -1854.497
weathersit_moderate    -678.4237     78.385     -8.655      0.000    -832.429    -524.419
temp                   4091.5171    295.064     13.867      0.000    3511.798    4671.236
==============================================================================
Omnibus:                       63.271   Durbin-Watson:                   2.151
Prob(Omnibus):                  0.000   Jarque-Bera (JB):              118.647
Skew:                          -0.734   Prob(JB):                     1.72e-26
Kurtosis:                       4.848   Cond. No.                         19.0
==============================================================================

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

Conclusion

Significant variables to predict the demand for shared bikes

- yr
- season_spring
- season_winter
- season_summer
- mnth_sept
- holiday
- workingday
- weathersit_bad
- weathersit_moderate
- temp

On the Final model on train and test dataset R^2 = 0.8 and P-Value is close to 0.0.

In [ ]:

In [ ]:

In [ ]:

In [ ]: