bike sharing

March 6, 2022

1 Libraries

[476]:
```
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
import pandas as pd
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import MinMaxScaler
from sklearn.model_selection import GridSearchCV
from sklearn.metrics import accuracy_score
from sklearn.ensemble import GradientBoostingRegressor

import warnings
warnings.filterwarnings("ignore")
```

2 Dataset

[477]:
```
df = pd.read_csv('day.csv', index_col=0)
```

[478]:
```
df = df.rename(columns={'weathersit':'weather',
                        'yr':'year',
                        'mnth':'month',
                        'hr':'hour',
                        'hum':'humidity',
                        'cnt':'count'})
```

[479]:
```
df
```

[479]:
```
             dteday  season  year  month  holiday  weekday  workingday  \
instant
1        01-01-2018       1     0      1        0        6           0
2        02-01-2018       1     0      1        0        0           0
3        03-01-2018       1     0      1        0        1           1
4        04-01-2018       1     0      1        0        2           1
5        05-01-2018       1     0      1        0        3           1
…               …      …    …     …       …        …          …
726      27-12-2019       1     1     12        0        4           1
```

```
727      28-12-2019       1     1     12        0        5           1
728      29-12-2019       1     1     12        0        6           0
729      30-12-2019       1     1     12        0        0           0
730      31-12-2019       1     1     12        0        1           1

         weather       temp     atemp  humidity  windspeed  casual  \
instant
1              2  14.110847  18.18125   80.5833  10.749882     331
2              2  14.902598  17.68695   69.6087  16.652113     131
3              1   8.050924   9.47025   43.7273  16.636703     120
4              1   8.200000  10.60610   59.0435  10.739832     108
5              1   9.305237  11.46350   43.6957  12.522300      82
…          …        …       …       …        …     …
726            2  10.420847  11.33210   65.2917  23.458911     247
727            2  10.386653  12.75230   59.0000  10.416557     644
728            2  10.386653  12.12000   75.2917   8.333661     159
729            1  10.489153  11.58500   48.3333  23.500518     364
730            2   8.849153  11.17435   57.7500  10.374682     439

         registered  count
instant
1               654    985
2               670    801
3              1229   1349
4              1454   1562
5              1518   1600
…             …    …
726            1867   2114
727            2451   3095
728            1182   1341
729            1432   1796
730            2290   2729

[730 rows x 15 columns]
```

[480]:

```
df.isnull().sum()
```

[480]:

```
dteday        0
season        0
year          0
month         0
holiday       0
weekday       0
workingday    0
weather       0
temp          0
atemp         0
```

```
humidity      0
windspeed     0
casual        0
registered    0
count         0
dtype: int64
```

[481]:

```python
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 730 entries, 1 to 730
Data columns (total 15 columns):
 #   Column      Non-Null Count  Dtype
---  ------      --------------  -----
 0   dteday      730 non-null    object
 1   season      730 non-null    int64
 2   year        730 non-null    int64
 3   month       730 non-null    int64
 4   holiday     730 non-null    int64
 5   weekday     730 non-null    int64
 6   workingday  730 non-null    int64
 7   weather     730 non-null    int64
 8   temp        730 non-null    float64
 9   atemp       730 non-null    float64
 10  humidity    730 non-null    float64
 11  windspeed   730 non-null    float64
 12  casual      730 non-null    int64
 13  registered  730 non-null    int64
 14  count       730 non-null    int64
dtypes: float64(4), int64(10), object(1)
memory usage: 91.2+ KB
```

[482]:

```python
df.columns
```

[482]:

```
Index(['dteday', 'season', 'year', 'month', 'holiday', 'weekday', 'workingday',
       'weather', 'temp', 'atemp', 'humidity', 'windspeed', 'casual',
       'registered', 'count'],
      dtype='object')
```

[483]:

```python
df['season'].unique()
```

[483]:

```
array([1, 2, 3, 4], dtype=int64)
```

[484]:

```python
df['season'] = df['season'].replace(1, 'spring')
df['season'] = df['season'].replace(2, 'summer')
df['season'] = df['season'].replace(3, 'fall')
df['season'] = df['season'].replace(4, 'winter')
#df['season'] = df.loc[(df['season']==1), 'season']='spring'
```

```
[485]: df['season'].unique()
```

```
[485]: array(['spring', 'summer', 'fall', 'winter'], dtype=object)
```

```
[486]: df['season'].value_counts()
```

```
[486]: fall      188
       summer    184
       spring    180
       winter    178
       Name: season, dtype: int64
```

```
[487]: def object_map(x):
           return x.map({1:'jan',
                         2:'feb',
                         3:'mar',
                         4:'apr',
                         5:'may',
                         6:'june',
                         7:'july',
                         8:'aug',
                         9:'sep',
                         10:'oct',
                         11:'nov',
                         12:'dec'})
```

```
[488]: df['month']= df[['month']].apply(object_map)
```

```
[489]: df['season'].unique()
```

```
[489]: array(['spring', 'summer', 'fall', 'winter'], dtype=object)
```

```
[490]: df['month'].value_counts()
```

```
[490]: jan     62
       mar     62
       may     62
       july    62
       aug     62
       oct     62
       dec     62
       apr     60
       june    60
       sep     60
       nov     60
       feb     56
       Name: month, dtype: int64
```

```
[491]: df['holiday'].value_counts()
```

```
[491]: 0    709
       1     21
       Name: holiday, dtype: int64
```

```
[492]: df['weekday'].unique()
```

```
[492]: array([6, 0, 1, 2, 3, 4, 5], dtype=int64)
```

```
[493]: def week_map(x):
            return x.map({1:'wed',
                          2:'thur',
                          3:'fri',
                          4:'sat',
                          5:'sun',
                          6:'mon',
                          0:'tues'})
```

```
[494]: df['weekday'] = df[['weekday']].apply(week_map)
```

```
[495]: df['weekday'].unique()
```

```
[495]: array(['mon', 'tues', 'wed', 'thur', 'fri', 'sat', 'sun'], dtype=object)
```

```
[496]: df['weekday'].value_counts()
```

```
[496]: mon     105
        tues    105
        wed     105
        thur    104
        sat     104
        sun     104
        fri     103
        Name: weekday, dtype: int64
```

```
[497]: df['workingday'].value_counts()
```

```
[497]: 1    499
       0    231
       Name: workingday, dtype: int64
```

```
[498]: df['weather']
```

```
[498]: instant
        1      2
        2      2
        3      1
```

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

```
[499]: df['weather'].value_counts()
```

```
[499]: 1    463
       2    246
       3     21
       Name: weather, dtype: int64
```

```
[500]: def weather_map(x):
           return x.map({1:'Clear',
                         2:'Bad',
                         3:'Rainy'})
```

```
[501]: df['weather'] = df[['weather']].apply(weather_map)
```

```
[502]: df['weather'].unique()
```

```
[502]: array(['Bad', 'Clear', 'Rainy'], dtype=object)
```

```
[503]: df['weather'].value_counts()
```

```
[503]: Clear    463
       Bad      246
       Rainy     21
       Name: weather, dtype: int64
```

```
[504]: df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 730 entries, 1 to 730
Data columns (total 15 columns):
 #   Column      Non-Null Count  Dtype
---  ------      --------------  -----
 0   dteday      730 non-null    object
 1   season      730 non-null    object
 2   year        730 non-null    int64
 3   month       730 non-null    object
 4   holiday     730 non-null    int64
 5   weekday     730 non-null    object
 6   workingday  730 non-null    int64
 7   weather     730 non-null    object
 8   temp        730 non-null    float64
 9   atemp       730 non-null    float64
 10  humidity    730 non-null    float64
 11  windspeed   730 non-null    float64
 12  casual      730 non-null    int64
 13  registered  730 non-null    int64
 14  count       730 non-null    int64
dtypes: float64(4), int64(6), object(5)
memory usage: 91.2+ KB
```

[505]:

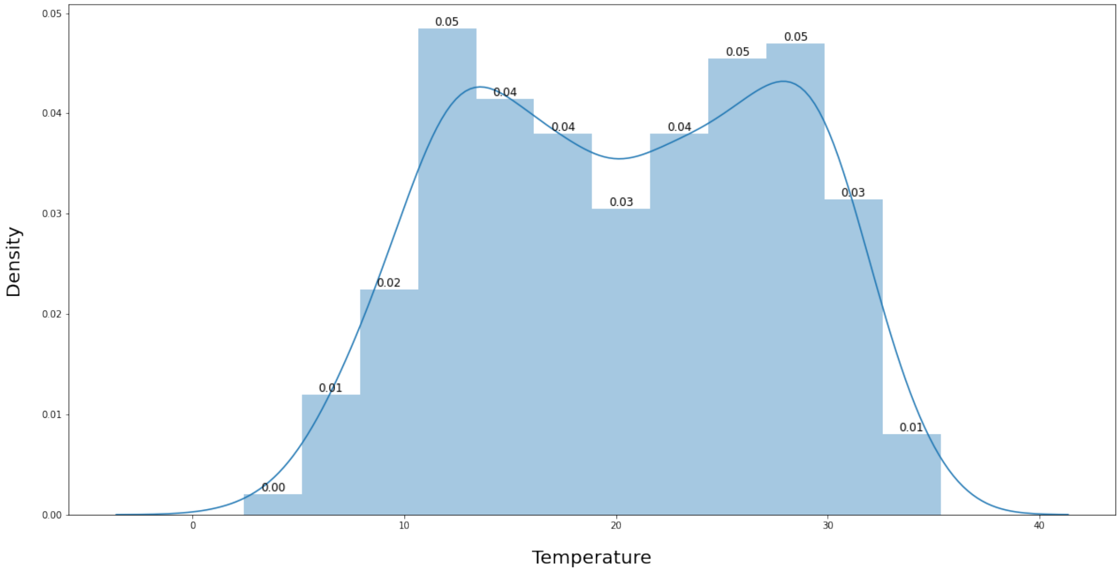
```
df['temp']
```

[505]:

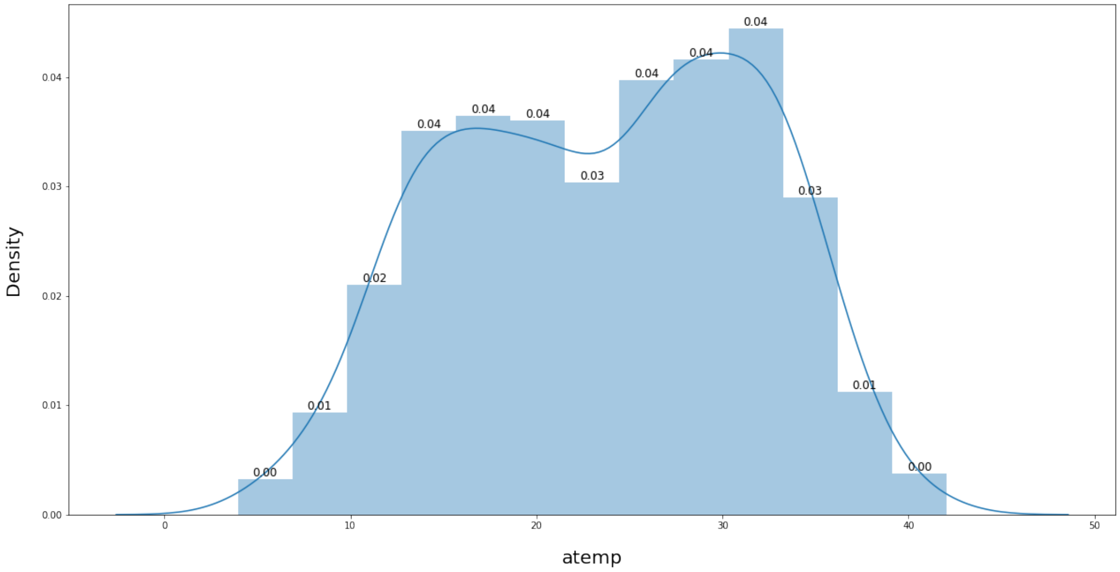
```
instant
1      14.110847
2      14.902598
3       8.050924
4       8.200000
5       9.305237
         …
726    10.420847
727    10.386653
728    10.386653
729    10.489153
730     8.849153
Name: temp, Length: 730, dtype: float64
```

[506]:

```
plt.figure(figsize=(20, 10))

_=sns.distplot(df['temp'])
plt.xlabel('Temperature', fontsize=20, labelpad=20)
plt.ylabel('Density', fontsize=20, labelpad=20)
ax = plt.gca()
for i in ax.patches:
    ax.text(i.get_x() + i.get_width()/2 , i.get_height(), '%.2f' %float(i.
 ↪get_height()), fontsize=12, ha='center', va='bottom')
plt.show()
```

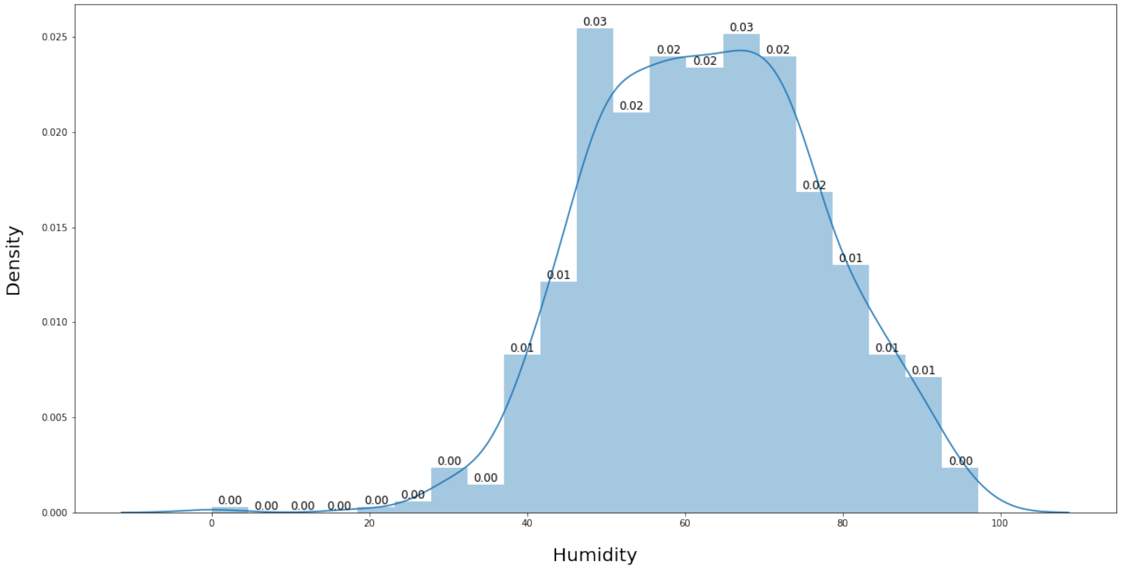


[507]:
```
plt.figure(figsize=(20, 10))

_=sns.distplot(df['atemp'])
plt.xlabel('atemp', fontsize=20, labelpad=20)
plt.ylabel('Density', fontsize=20, labelpad=20)
ax = plt.gca()
for i in ax.patches:
    ax.text(i.get_x() + i.get_width()/2 , i.get_height(), '%.2f' %float(i.
 ↪get_height()), fontsize=12, ha='center', va='bottom')
plt.show()
```



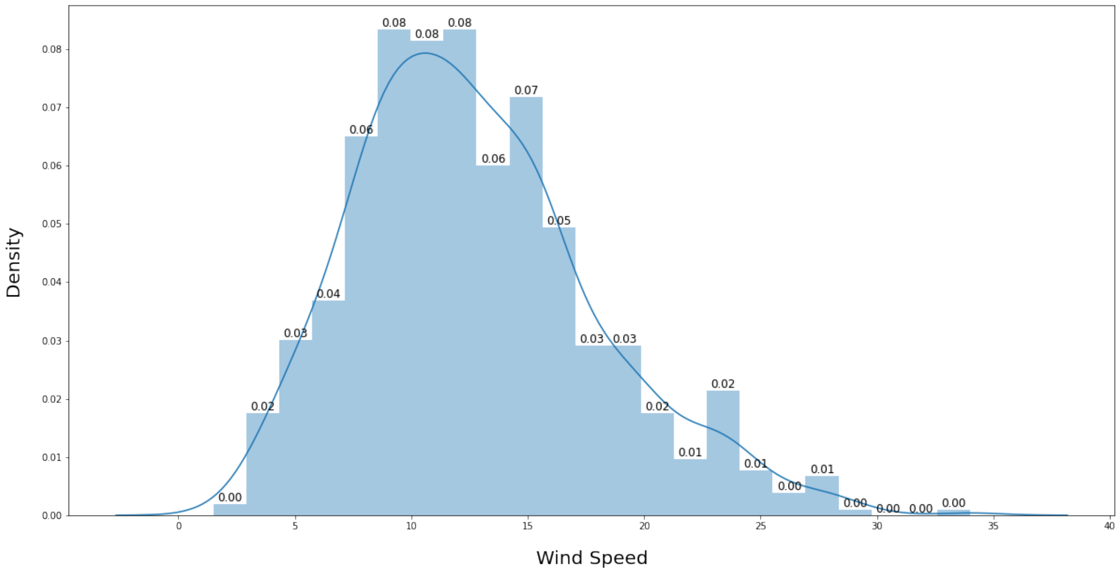
[508]:

```
plt.figure(figsize=(20, 10))

_=sns.distplot(df['humidity'])
plt.xlabel('Humidity', fontsize=20, labelpad=20)
plt.ylabel('Density', fontsize=20, labelpad=20)
ax = plt.gca()
for i in ax.patches:
    ax.text(i.get_x() + i.get_width()/2 , i.get_height(), '%.2f' %float(i.
 ↪get_height()), fontsize=12, ha='center', va='bottom')
plt.show()
```



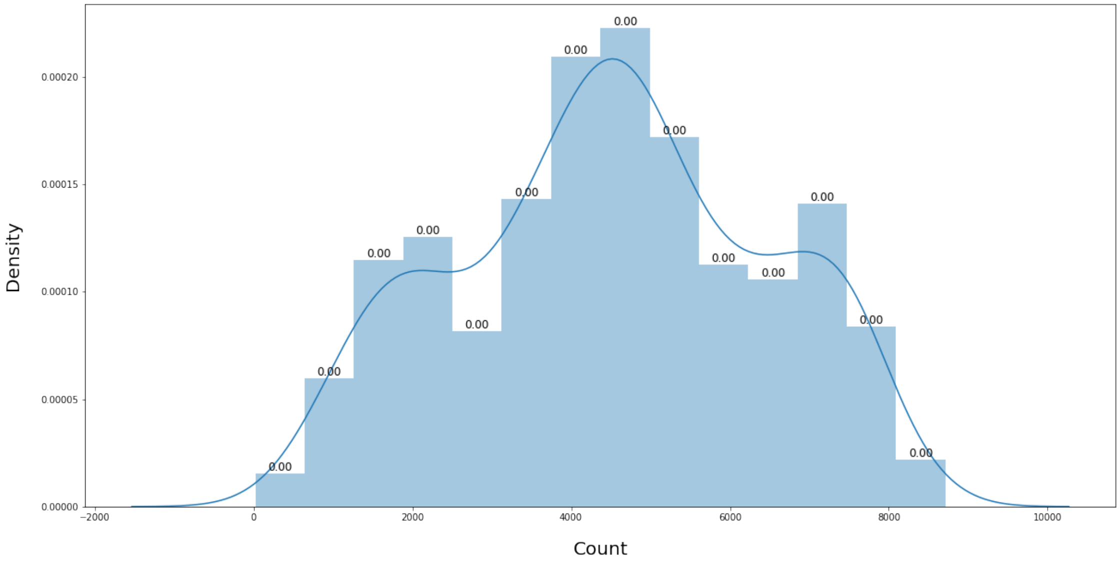
[509]:

```
plt.figure(figsize=(20, 10))

_=sns.distplot(df['windspeed'])
plt.xlabel('Wind Speed', fontsize=20, labelpad=20)
plt.ylabel('Density', fontsize=20, labelpad=20)
ax = plt.gca()
for i in ax.patches:
    ax.text(i.get_x() + i.get_width()/2 , i.get_height(), '%.2f' %float(i.
 ↪get_height()), fontsize=12, ha='center', va='bottom')
plt.show()
```



[510]:

```
plt.figure(figsize=(20, 10))

_=sns.distplot(df['count'])
plt.xlabel('Count', fontsize=20, labelpad=20)
plt.ylabel('Density', fontsize=20, labelpad=20)
ax = plt.gca()
for i in ax.patches:
    ax.text(i.get_x() + i.get_width()/2 , i.get_height(), '%.2f' %float(i.
 ↪get_height()), fontsize=12, ha='center', va='bottom')
plt.show()
```



[511]:

```
df['dteday'] = pd.to_datetime(df['dteday'])
#df['dteday']= df['dteday'].astype(df['dteday'])
```

[512]:

```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 730 entries, 1 to 730
Data columns (total 15 columns):
 #   Column      Non-Null Count  Dtype
---  ------      --------------  -----
 0   dteday      730 non-null    datetime64[ns]
 1   season      730 non-null    object
 2   year        730 non-null    int64
 3   month       730 non-null    object
 4   holiday     730 non-null    int64
 5   weekday     730 non-null    object
 6   workingday  730 non-null    int64
 7   weather     730 non-null    object
 8   temp        730 non-null    float64
 9   atemp       730 non-null    float64
 10  humidity    730 non-null    float64
 11  windspeed   730 non-null    float64
 12  casual      730 non-null    int64
 13  registered  730 non-null    int64
 14  count       730 non-null    int64
dtypes: datetime64[ns](1), float64(4), int64(6), object(4)
memory usage: 91.2+ KB
```

[513]:

```
df_categorical = df.select_dtypes('object')
```

[514]:

```
df_categorical.info()
```

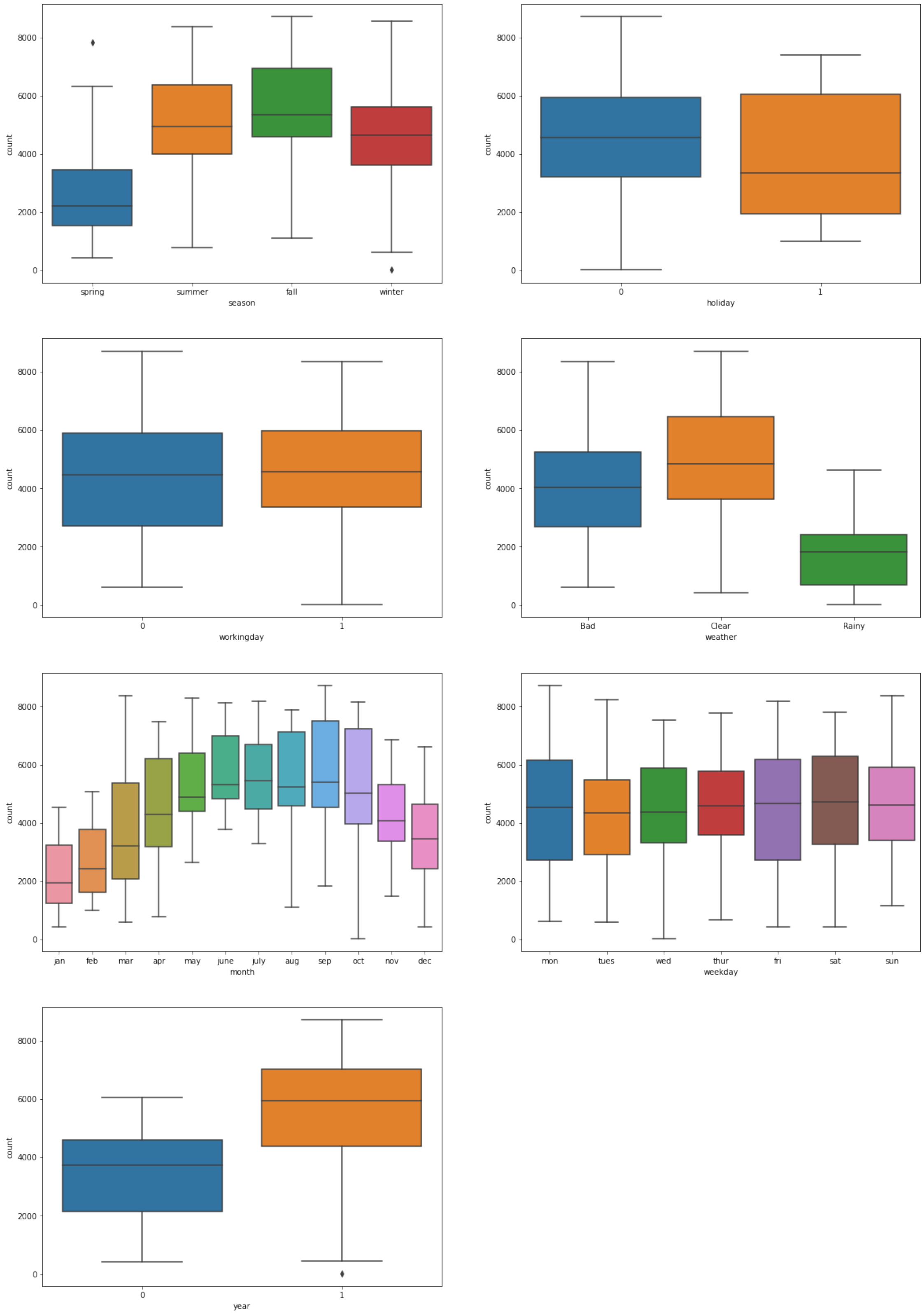
```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 730 entries, 1 to 730
Data columns (total 4 columns):
 #   Column   Non-Null Count  Dtype
---  ------   --------------  -----
 0   season   730 non-null    object
 1   month    730 non-null    object
 2   weekday  730 non-null    object
 3   weather  730 non-null    object
dtypes: object(4)
memory usage: 28.5+ KB
```

[515]:

```
df_categorical
```

```
[515]:          season month weekday weather
       instant
       1         spring   jan     mon     Bad
       2         spring   jan    tues     Bad
       3         spring   jan     wed   Clear
       4         spring   jan    thur   Clear
       5         spring   jan     fri   Clear
       …           …   …     …     …
       726       spring   dec     sat     Bad
       727       spring   dec     sun     Bad
       728       spring   dec     mon     Bad
       729       spring   dec    tues   Clear
       730       spring   dec     wed     Bad

       [730 rows x 4 columns]
```

```
[516]: plt.figure(figsize=(20, 30))

       plt.subplot(4,2,1)
       sns.boxplot(x=df['season'], y=df['count'], data=df)

       plt.subplot(4,2,2)
       sns.boxplot(x=df['holiday'], y=df['count'], data=df)

       plt.subplot(4,2,3)
       sns.boxplot(x=df['workingday'], y=df['count'], data=df)

       plt.subplot(4,2,4)
       sns.boxplot(x=df['weather'], y=df['count'], data=df)

       plt.subplot(4,2,5)
       sns.boxplot(x=df['month'], y=df['count'], data=df)

       plt.subplot(4,2,6)
       sns.boxplot(x=df['weekday'], y=df['count'], data=df)

       plt.subplot(4,2,7)
       sns.boxplot(x=df['year'], y=df['count'], data=df)

       plt.show()
```

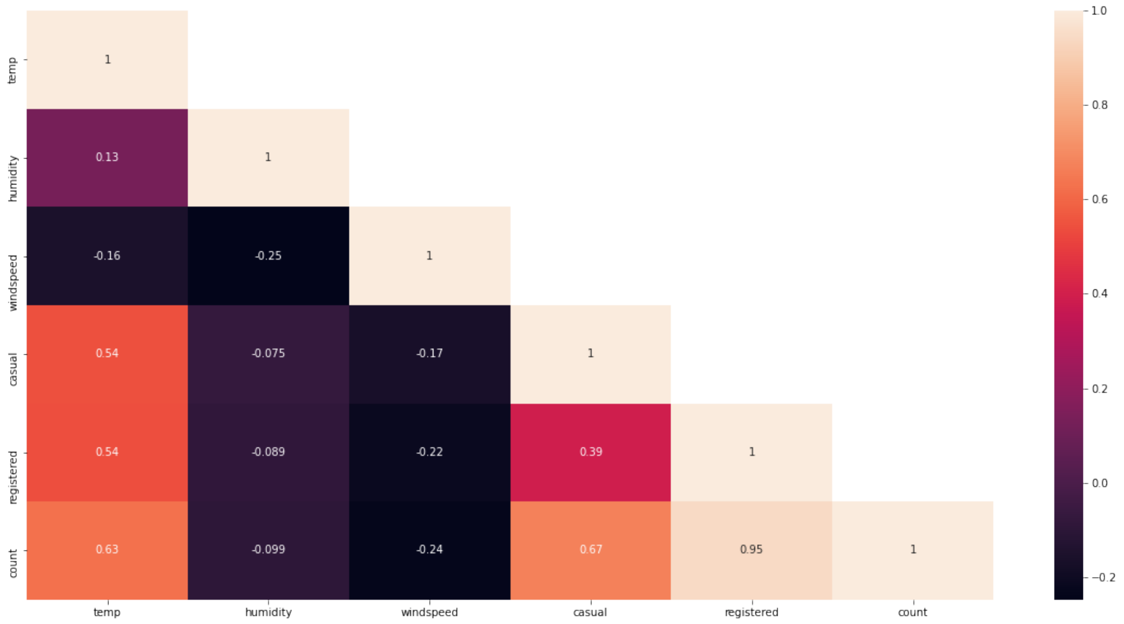

count
0
2000
4000
6000
8000
spring
summer
fall
winter
season
holiday
workingday
Bad
Clear
Rainy
weather
jan
feb
mar
apr
may
june
july
aug
sep
oct
nov
dec
month
mon
tues
wed
thur
fri
sat
sun
weekday
year
1

[517]:
```
df_numeric = df.select_dtypes(include=['float64', 'int64'])
df_numeric
```

[517]:
```
         year  holiday  workingday       temp     atemp  humidity  windspeed  \
instant
1           0        0           0  14.110847  18.18125   80.5833  10.749882
2           0        0           0  14.902598  17.68695   69.6087  16.652113
3           0        0           1   8.050924   9.47025   43.7273  16.636703
4           0        0           1   8.200000  10.60610   59.0435  10.739832
5           0        0           1   9.305237  11.46350   43.6957  12.522300
…         …        …           …          …         …         …          …
726         1        0           1  10.420847  11.33210   65.2917  23.458911
727         1        0           1  10.386653  12.75230   59.0000  10.416557
728         1        0           0  10.386653  12.12000   75.2917   8.333661
729         1        0           0  10.489153  11.58500   48.3333  23.500518
730         1        0           1   8.849153  11.17435   57.7500  10.374682

         casual  registered  count
instant
1           331         654    985
2           131         670    801
3           120        1229   1349
4           108        1454   1562
5            82        1518   1600
…           …           …      …
726         247        1867   2114
727         644        2451   3095
728         159        1182   1341
729         364        1432   1796
730         439        2290   2729

[730 rows x 10 columns]
```

[518]:
```
df_numeric = df_numeric.drop(['year', 'holiday', 'workingday', 'atemp'], axis=1)
```

[519]:
```
plt.figure(figsize=(20, 10))
mask=np.array(df_numeric.corr())
mask[np.tril_indices_from(mask)]=False
sns.heatmap(df_numeric.corr(), annot=True, mask=mask)
plt.show()
```



```
[520]: df_categorical
```

```
[520]:         season month weekday weather
       instant
       1        spring   jan     mon     Bad
       2        spring   jan    tues     Bad
       3        spring   jan     wed   Clear
       4        spring   jan    thur   Clear
       5        spring   jan     fri   Clear
       …           …    …      …      …
       726      spring   dec     sat     Bad
       727      spring   dec     sun     Bad
       728      spring   dec     mon     Bad
       729      spring   dec    tues   Clear
       730      spring   dec     wed     Bad

       [730 rows x 4 columns]
```

```
[521]: df_dummies = pd.get_dummies(df_categorical, drop_first=True)
       df_dummies.head()
```

```
[521]:         season_spring  season_summer  season_winter  month_aug  month_dec  \
       instant
       1                    1              0              0          0          0
       2                    1              0              0          0          0
       3                    1              0              0          0          0
       4                    1              0              0          0          0
```

```
5                     1              0              0          0          0

         month_feb  month_jan  month_july  month_june  month_mar  ...  \
instant                                                           ...
1                0          1           0           0          0  ...
2                0          1           0           0          0  ...
3                0          1           0           0          0  ...
4                0          1           0           0          0  ...
5                0          1           0           0          0  ...

         month_oct  month_sep  weekday_mon  weekday_sat  weekday_sun  \
instant
1                0          0            1            0            0
2                0          0            0            0            0
3                0          0            0            0            0
4                0          0            0            0            0
5                0          0            0            0            0

         weekday_thur  weekday_tues  weekday_wed  weather_Clear  weather_Rainy
instant
1                   0             0            0              0              0
2                   0             1            0              0              0
3                   0             0            1              1              0
4                   1             0            0              1              0
5                   0             0            0              1              0

[5 rows x 22 columns]
```

[522]:
```
df = df.drop(list(df_categorical.columns), axis=1)
df
```

[522]:
```
            dteday  year  holiday  workingday       temp     atemp  humidity  \
instant
1       2018-01-01     0        0           0  14.110847  18.18125   80.5833
2       2018-02-01     0        0           0  14.902598  17.68695   69.6087
3       2018-03-01     0        0           1   8.050924   9.47025   43.7273
4       2018-04-01     0        0           1   8.200000  10.60610   59.0435
5       2018-05-01     0        0           1   9.305237  11.46350   43.6957
...            ...   ...      ...         ...        ...       ...       ...
726     2019-12-27     1        0           1  10.420847  11.33210   65.2917
727     2019-12-28     1        0           1  10.386653  12.75230   59.0000
728     2019-12-29     1        0           0  10.386653  12.12000   75.2917
729     2019-12-30     1        0           0  10.489153  11.58500   48.3333
730     2019-12-31     1        0           1   8.849153  11.17435   57.7500

         windspeed  casual  registered  count
instant
```

```
1        10.749882     331         654    985
2        16.652113     131         670    801
3        16.636703     120        1229   1349
4        10.739832     108        1454   1562
5        12.522300      82        1518   1600
…            …     …         …    …
726      23.458911     247        1867   2114
727      10.416557     644        2451   3095
728       8.333661     159        1182   1341
729      23.500518     364        1432   1796
730      10.374682     439        2290   2729

[730 rows x 11 columns]
```

[523]:

```
df = pd.concat([df, df_dummies], axis=1)
df.head()
```

[523]:

```
            dteday  year  holiday  workingday       temp     atemp  humidity  \
instant
1       2018-01-01     0        0           0  14.110847  18.18125   80.5833
2       2018-02-01     0        0           0  14.902598  17.68695   69.6087
3       2018-03-01     0        0           1   8.050924   9.47025   43.7273
4       2018-04-01     0        0           1   8.200000  10.60610   59.0435
5       2018-05-01     0        0           1   9.305237  11.46350   43.6957

         windspeed  casual  registered  …  month_oct  month_sep  \
instant                                 …
1        10.749882     331         654  …          0          0
2        16.652113     131         670  …          0          0
3        16.636703     120        1229  …          0          0
4        10.739832     108        1454  …          0          0
5        12.522300      82        1518  …          0          0

         weekday_mon  weekday_sat  weekday_sun  weekday_thur  weekday_tues  \
instant
1                  1            0            0             0             0
2                  0            0            0             0             1
3                  0            0            0             0             0
4                  0            0            0             1             0
5                  0            0            0             0             0

         weekday_wed  weather_Clear  weather_Rainy
instant
1                  0              0              0
2                  0              0              0
3                  1              1              0
4                  0              1              0
```

```
5                  0              1              0

[5 rows x 33 columns]
```

[524]:
```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 730 entries, 1 to 730
Data columns (total 33 columns):
 #   Column         Non-Null Count  Dtype
---  ------         --------------  -----
 0   dteday         730 non-null    datetime64[ns]
 1   year           730 non-null    int64
 2   holiday        730 non-null    int64
 3   workingday     730 non-null    int64
 4   temp           730 non-null    float64
 5   atemp          730 non-null    float64
 6   humidity       730 non-null    float64
 7   windspeed      730 non-null    float64
 8   casual         730 non-null    int64
 9   registered     730 non-null    int64
 10  count          730 non-null    int64
 11  season_spring  730 non-null    uint8
 12  season_summer  730 non-null    uint8
 13  season_winter  730 non-null    uint8
 14  month_aug      730 non-null    uint8
 15  month_dec      730 non-null    uint8
 16  month_feb      730 non-null    uint8
 17  month_jan      730 non-null    uint8
 18  month_july     730 non-null    uint8
 19  month_june     730 non-null    uint8
 20  month_mar      730 non-null    uint8
 21  month_may      730 non-null    uint8
 22  month_nov      730 non-null    uint8
 23  month_oct      730 non-null    uint8
 24  month_sep      730 non-null    uint8
 25  weekday_mon    730 non-null    uint8
 26  weekday_sat    730 non-null    uint8
 27  weekday_sun    730 non-null    uint8
 28  weekday_thur   730 non-null    uint8
 29  weekday_tues   730 non-null    uint8
 30  weekday_wed    730 non-null    uint8
 31  weather_Clear  730 non-null    uint8
 32  weather_Rainy  730 non-null    uint8
dtypes: datetime64[ns](1), float64(4), int64(6), uint8(22)
memory usage: 100.3 KB
```

[525]:
```
df.drop(['dteday'], axis=1, inplace=True)
df.head()
```

[525]:
```
         year  holiday  workingday       temp     atemp  humidity  windspeed  \
instant
1           0        0           0  14.110847  18.18125   80.5833  10.749882
2           0        0           0  14.902598  17.68695   69.6087  16.652113
3           0        0           1   8.050924   9.47025   43.7273  16.636703
4           0        0           1   8.200000  10.60610   59.0435  10.739832
5           0        0           1   9.305237  11.46350   43.6957  12.522300

         casual  registered  count  ...  month_oct  month_sep  weekday_mon  \
instant                             ...
1           331         654    985  ...          0          0            1
2           131         670    801  ...          0          0            0
3           120        1229   1349  ...          0          0            0
4           108        1454   1562  ...          0          0            0
5            82        1518   1600  ...          0          0            0

         weekday_sat  weekday_sun  weekday_thur  weekday_tues  weekday_wed  \
instant
1                  0            0             0             0            0
2                  0            0             0             1            0
3                  0            0             0             0            1
4                  0            0             1             0            0
5                  0            0             0             0            0

         weather_Clear  weather_Rainy
instant
1                    0              0
2                    0              0
3                    1              0
4                    1              0
5                    1              0

[5 rows x 32 columns]
```

3 Train & Test

[526]:
```
y  = df['count']
df = df.drop(['count'], axis=1)
X  = df
```

[527]:
```
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.30,␣
 ↪random_state=100)
```

4 Normalization

```
[528]: sc = MinMaxScaler()

       X_train[['atemp', 'humidity', 'windspeed', 'casual', 'registered']] = sc.
        ↪fit_transform(X_train[['atemp', 'humidity', 'windspeed', 'casual',␣
        ↪'registered']])
       X_test[['atemp', 'humidity', 'windspeed', 'casual', 'registered']]  = sc.
        ↪transform(X_test[['atemp', 'humidity', 'windspeed', 'casual', 'registered']])
```

5 Model (GradientBoostingRegressor)

```
[529]: model = GradientBoostingRegressor()
```

```
[530]: model.fit(X_train, y_train)
```

```
[530]: GradientBoostingRegressor()
```

```
[531]: model.score(X_train, y_train)
```

```
[531]: 0.999049329839438
```

```
[532]: model.score(X_test, y_test)
```

```
[532]: 0.9954623762256934
```