How to Build a Machine Learning Model in Python

Learning Objectives

In the modeling stage of the machine learning process, our goal is to choose and apply the appropriate machine learning approach that works with the data we have and solves the problem that we intend to solve. If our objective is to build a model that predicts a numeric or continuous value, then our problem is known as a regression problem. One of the most common models used in solving regression problems is **Linear Regression**. By the end of the tutorial, you will have learned:

- how to collect, explore and prepare data
- how to build and evaluate a model

1. Collect the Data

In [1]:
```
import pandas as pd
bikes = pd.read_csv("bikes.csv")
bikes.head()
```

Out[1]:

	temperature	humidity	windspeed	rentals
0	46.716528	0.815969	13.669663	985
1	48.350239	0.800497	15.199782	801
2	34.212394	0.592097	13.247558	1349
3	34.520000	0.623196	11.687963	1562
4	36.800562	0.624643	13.148281	1600

2. Explore the Data

In [2]:
```
bikes.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 731 entries, 0 to 730
Data columns (total 4 columns):
 #   Column       Non-Null Count  Dtype  
---  ------       --------------  -----  
 0   temperature  731 non-null    float64
 1   humidity     731 non-null    float64
 2   windspeed    731 non-null    float64
 3   rentals      731 non-null    int64  
dtypes: float64(3), int64(1)
memory usage: 23.0 KB
```

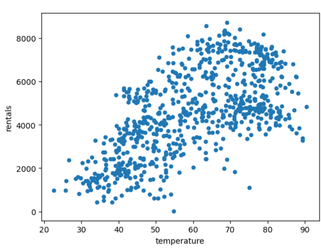
In [3]:
```
bikes.describe()
```

Out[3]:

	temperature	humidity	windspeed	rentals
count	731.000000	731.000000	731.000000	731.000000
mean	59.509553	0.486937	9.238886	4504.348837
std	15.486114	0.185415	3.379815	1937.211452
min	22.602432	0.000000	0.932208	22.000000
25%	46.117264	0.353548	6.863568	3152.000000
50%	59.758972	0.502227	9.503508	4548.000000
75%	73.048236	0.624671	11.814559	5956.000000
max	90.497028	0.972500	21.126627	8714.000000

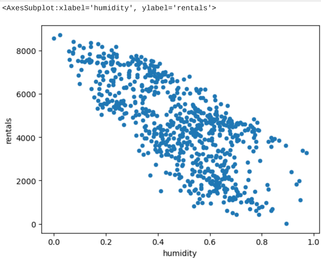
In [4]:
```
%matplotlib inline
bikes.plot(kind = 'scatter', x = 'temperature', y = 'rentals')
```

Out[4]:
```
<AxesSubplot:xlabel='temperature', ylabel='rentals'>
```



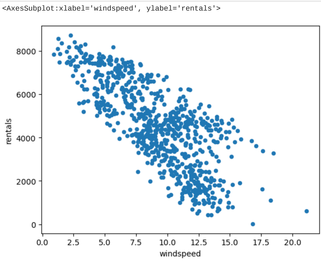
In [5]:
```
bikes.plot(kind = 'scatter', x = 'humidity', y = 'rentals')
```

Out[5]:
```
<AxesSubplot:xlabel='humidity', ylabel='rentals'>
```



In [6]:
```
bikes.plot(kind = 'scatter', x = 'windspeed', y = 'rentals')
```

Out[6]:
```
<AxesSubplot:xlabel='windspeed', ylabel='rentals'>
```



3. Prepare the Data

In [7]:
```
response = 'rentals'
y = bikes[[response]]
y
```

Out[7]:

	rentals
0	985
1	801
2	1349
3	1562
4	1600
...	...
726	2114
727	3095
728	1341
729	1796
730	2729

731 rows × 1 columns

In [8]:
```
predictors = list(bikes.columns)
predictors.remove(response)
x = bikes[predictors]
x
```

Out[8]:

	temperature	humidity	windspeed
0	46.716528	0.815969	13.669663
1	48.350239	0.800497	15.199782
2	34.212394	0.592097	13.247558
3	34.520000	0.623196	11.687963
4	36.800562	0.624643	13.148281
...	...	...	...
726	39.102528	0.482493	10.801229
727	39.031972	0.480433	8.996301
728	39.031972	0.717730	11.829425
729	39.243472	0.523039	12.805314
730	35.859472	0.494808	9.346850

731 rows × 3 columns

In [9]:
```
from sklearn.model_selection import train_test_split
x_train, x_test, y_train, y_test = train_test_split(x, y, random_state = 1234)
```

4. Train the Model

In [10]:
```
from sklearn.linear_model import LinearRegression
model = LinearRegression().fit(x_train, y_train)
```

In [12]:
```
model.intercept_
```

Out[12]:
```
array([3800.68469948])
```

In [13]:
```
model.coef_
```

Out[13]:
```
array([[   80.35314543, -4665.73867387,  -196.21650368]])
```

The model coefficients correspond to the order in which the independent variables are listed in the training data. This means that the equation for the fitted regression line can be written as:

$$y = 3800.68 + 80.35 \times temperature - 4665.74 \times humidity - 196.22 \times windspeed$$

With the linear regression equation, we can estimate what our model will predict given any weather condition. For example, given a temperature of 72˚F, 22%$\%$ humidity and windspeed of 5 miles per hour, our model would predict:

$$7,578 \text{ bikes} \approx 3800.68 + 80.35 \times 72 - 4665.74 \times .22 - 196.22 \times 5$$

5. Evaluate the Model

In [14]:
```
model.score(x_test, y_test)
```

Out[14]:
```
0.9820623857913312
```

In [15]:
```
y_pred = model.predict(x_test)
```

In [17]:
```
from sklearn.metrics import mean_absolute_error
mean_absolute_error(y_test, y_pred)
```

Out[17]:
```
194.31620720519678
```

In [ ]:
```

```