1.用 panda 读取 bike.csv 文件: dt = pd.read_csv('bike.csv')
2.剔除 id 属性对应的列: dt = dt.drop(columns=['id'])
3．选择 city 为 1 的列并将 city 列剔除: dt = dt[dt['city'] == 1]
4.将时间在 6—18 的边为 1,其余的变为 0:这里要特别注意改变的顺序，可以先将大于等于 19 的列变为 0,然后将其他的变为 1，再将小于等于 5 的变为 0 就行了。
dt.loc[dt['hour']>=19,'hour']=0
dt.loc[dt['hour']<=18,'hour']=1
dt.loc[dt['hour']<=5,'hour']=0
5.将 y 列转化为 numpy，然后剔除 y 列:
取出 y 列，然后将其转化为 numpy，但此时时一个 1*n 的矩阵，需要用 reshape 转化为 n*1 的矩阵才行,然后再把 y 列丢掉。
y_series = dt['y']
y=y_series.to_numpy()
y = y.reshape(len(y),1)
dt = dt.drop(columns=['y'])
6.调用 train_test_split 然后把训练集和测试集按照 4:1 划分
from sklearn.model_selection import train_test_split
x_train,x_test,y_train,y_test = train_test_split(dt,y,test_size=0.2)
7.将 4 个集和归一化
x_train = scale.fit_transform(x_train)
x_test = scale.fit_transform(x_test)
y_train = scale.fit_transform(y_train)
y_test = scale.fit_transform(y_test)
8.然后用线性回归模型进行拟合
from sklearn.linear_model import LinearRegression
model = LinearRegression()
model.fit(x_train,y_train)
9.最后用测试集的 x 进行预测并用测试集 y 计算出标准差
from sklearn.metrics import mean_squared_error
y_pred = model.predict(x_test)
mse = mean_squared_error(y_test,y_pred)**0.5
print("MSE:",mse)

10.完整代码如下
```
import pandas as pd
from sklearn.preprocessing import MinMaxScaler

scale = MinMaxScaler()

import pandas as pd

dt = pd.read_csv('bike.csv')
```

```
dt = dt.drop(columns=['id'])
dt = dt[dt['city'] == 1]

#print(dt)

dt.loc[dt['hour']>=19,'hour']=0
dt.loc[dt['hour']<=18,'hour']=1
dt.loc[dt['hour']<=5,'hour']=0
#print(dt)

y_series = dt['y']
y=y_series.to_numpy()
y = y.reshape(len(y),1)
dt = dt.drop(columns=['y'])

from sklearn.model_selection import train_test_split
x_train,x_test,y_train,y_test = train_test_split(dt,y,test_size=0.2)



x_train = scale.fit_transform(x_train)
x_test = scale.fit_transform(x_test)
y_train = scale.fit_transform(y_train)
y_test = scale.fit_transform(y_test)

from sklearn.linear_model import LinearRegression
model = LinearRegression()
model.fit(x_train,y_train)



from sklearn.metrics import mean_squared_error
y_pred = model.predict(x_test)
mse = mean_squared_error(y_test,y_pred)**0.5
print("MSE:",mse)
```