代码

```python
import pandas as pd
import numpy as np
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import MinMaxScaler
from sklearn.linear_model import LinearRegression
from sklearn.metrics import mean_squared_error, r2_score

# 读取数据并处理
df = pd.read_csv('bike.csv')
df = df.drop('id', axis=1)

# 选择上海的数据
shanghai_df = df[df['city'] == 1].drop('city', axis=1)

# 处理小时特征
shanghai_df['hour'] = shanghai_df['hour'].apply(lambda x: 1 if 6 <= x <= 18 else 0)

# 分离特征和目标变量
y = shanghai_df.pop('y').values.reshape(-1, 1)
X = shanghai_df.values

# 划分训练集和测试集
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)

# 创建特征和标签归一化器
feature_scaler = MinMaxScaler()
label_scaler = MinMaxScaler()

# 归一化训练集和测试集
def normalize_data(X_train, X_test, y_train, y_test, feature_scaler, label_scaler):
    X_train_scaled = feature_scaler.fit_transform(X_train)
    X_test_scaled = feature_scaler.transform(X_test)
    y_train_scaled = label_scaler.fit_transform(y_train)
    y_test_scaled = label_scaler.transform(y_test)
    return X_train_scaled, X_test_scaled, y_train_scaled, y_test_scaled

X_train_scaled, X_test_scaled, y_train_scaled, y_test_scaled = normalize_data(X_train, X_test, y_train, y_test, feature_scaler, label_scaler)

# 构建线性回归模型
```

```
linear_model = LinearRegression()

# 利用训练集训练模型
linear_model.fit(X_train_scaled, y_train_scaled)

print("线性回归模型训练完成。")

# 使用测试集进行评估
y_pred_scaled = linear_model.predict(X_test_scaled)

# 反归一化预测值
y_pred = label_scaler.inverse_transform(y_pred_scaled)

# 计算均方误差（MSE）
mse = mean_squared_error(y_test, y_pred)

# 计算均方根误差（RMSE）
rmse = np.sqrt(mse)

# 计算R²分数
r2 = r2_score(y_test, y_pred)

print(f"均方根误差（RMSE）：{rmse}")
print(f"R² 分数: {r2}")
```