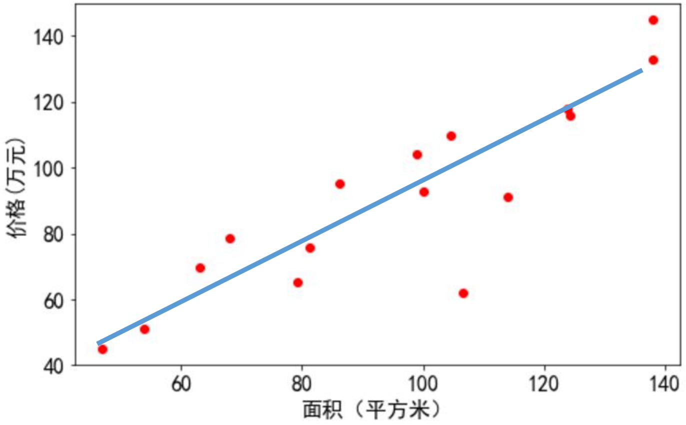
神经网络&深度学习 TENSORFLOW Google

9.6.2 实例：线性回归模型可视化

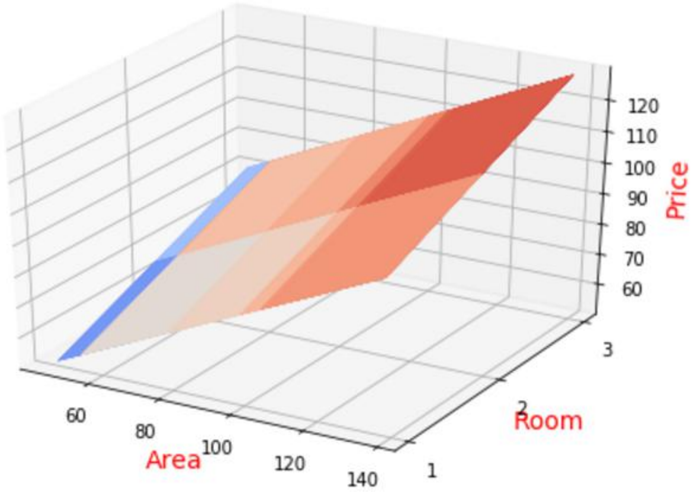
中国大学MOOC

9.6.2 线性回归模型可视化

一元线性回归

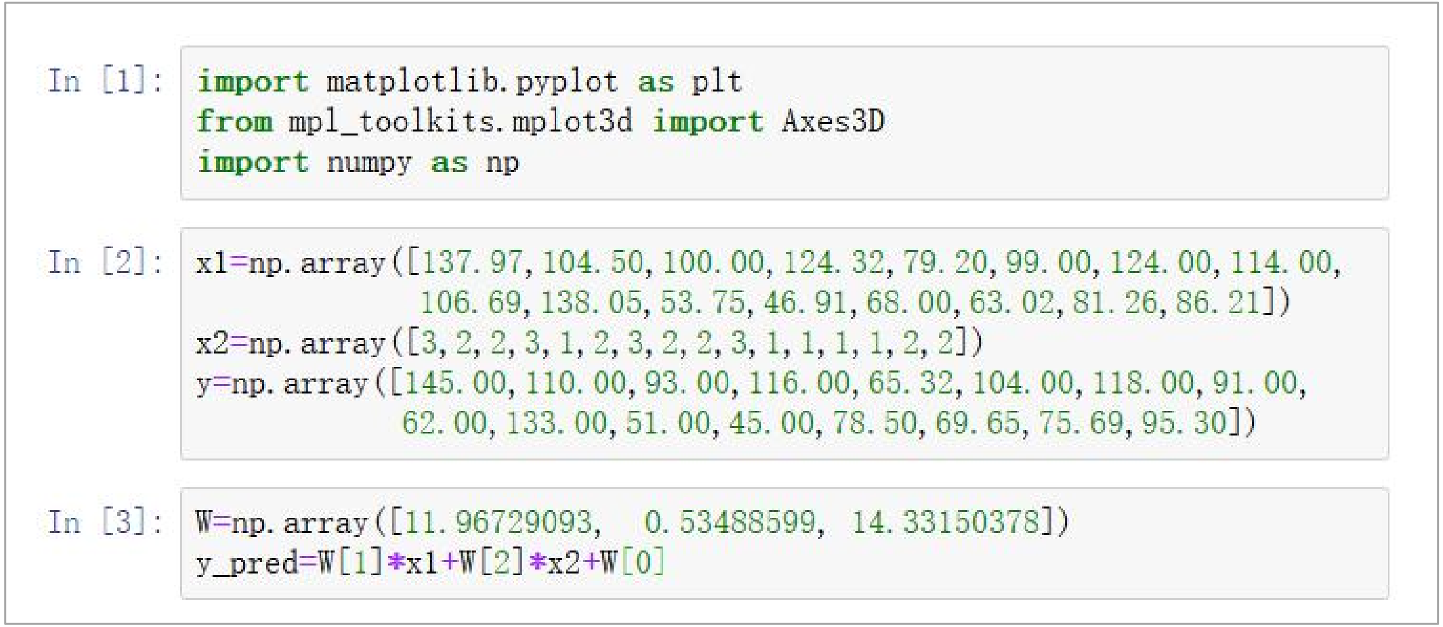


二元线性回归



9.6.2 线性回归模型可视化

加载数据



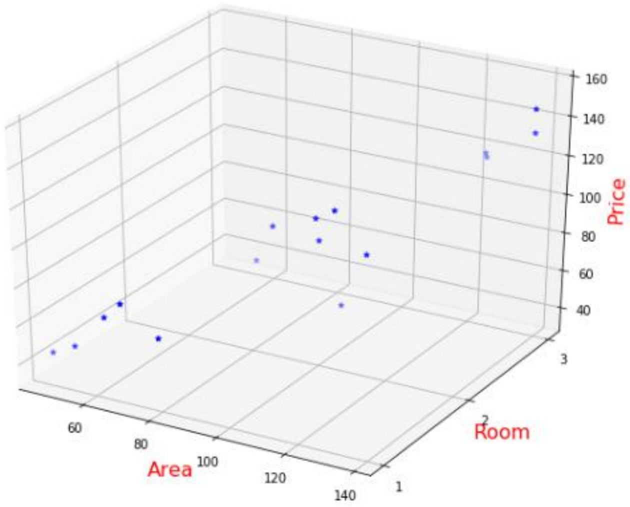
```
In [1]: import matplotlib.pyplot as plt
        from mpl_toolkits.mplot3d import Axes3D
        import numpy as np
```

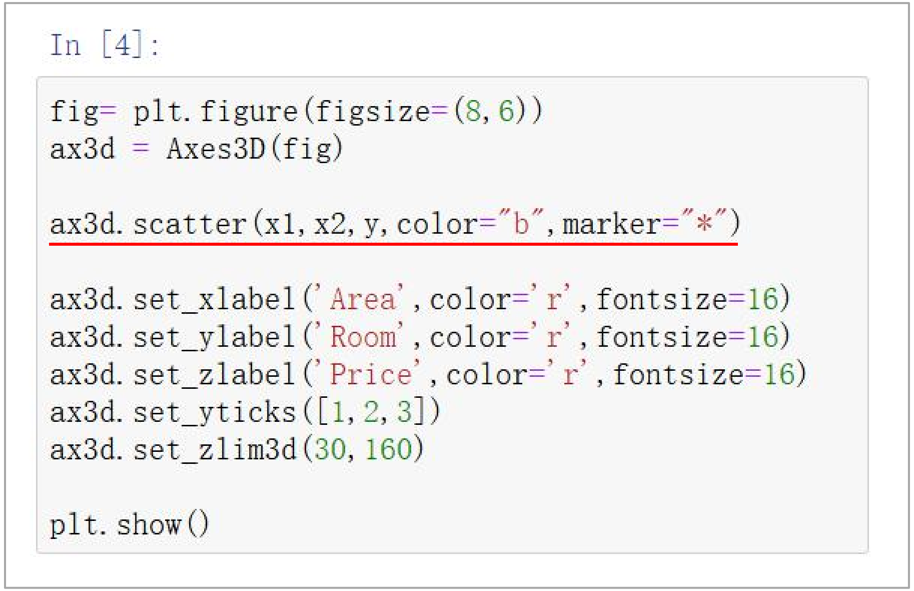
```
In [2]: x1=np.array([137.97,104.50,100.00,124.32,79.20,99.00,124.00,114.00,
                     106.69,138.05,53.75,46.91,68.00,63.02,81.26,86.21])
        x2=np.array([3,2,2,3,1,2,3,2,2,3,1,1,1,1,2,2])
        y=np.array([145.00,110.00,93.00,116.00,65.32,104.00,118.00,91.00,
                    62.00,133.00,51.00,45.00,78.50,69.65,75.69,95.30])
```

```
In [3]: W=np.array([11.96729093,  0.53488599,  14.33150378])
        y_pred=W[1]*x1+W[2]*x2+W[0]
```

9.6.2 线性回归模型可视化

绘制散点图



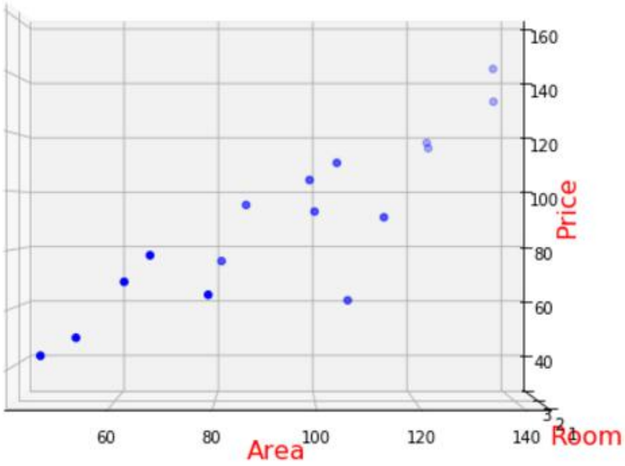

In [4]:

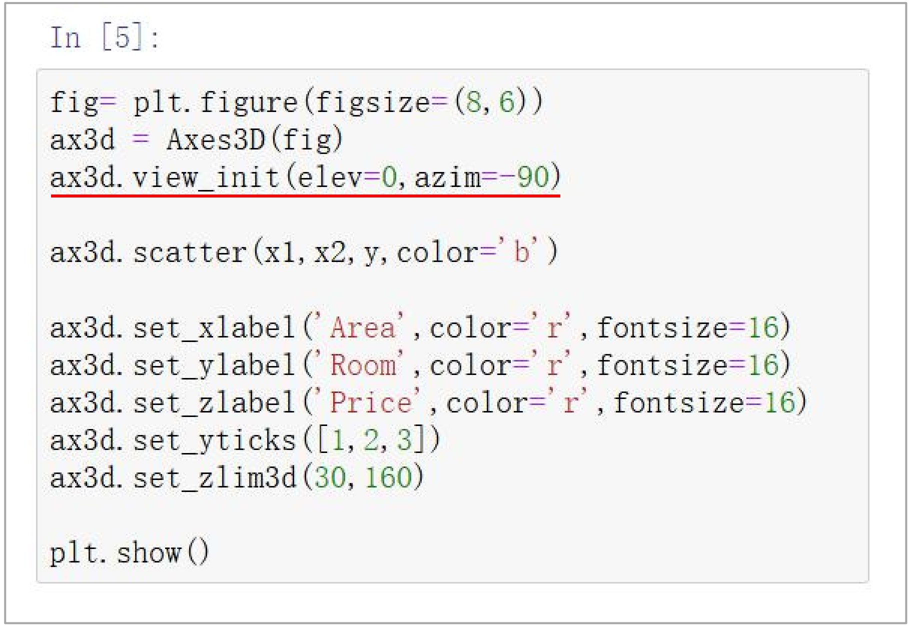
```
fig= plt.figure(figsize=(8,6))
ax3d = Axes3D(fig)

ax3d.scatter(x1,x2,y,color="b",marker="*")

ax3d.set_xlabel('Area',color='r',fontsize=16)
ax3d.set_ylabel('Room',color='r',fontsize=16)
ax3d.set_zlabel('Price',color='r',fontsize=16)
ax3d.set_yticks([1,2,3])
ax3d.set_zlim3d(30,160)

plt.show()
```

9.6.2 线性回归模型可视化

改变视角

view_init(elev,azim)





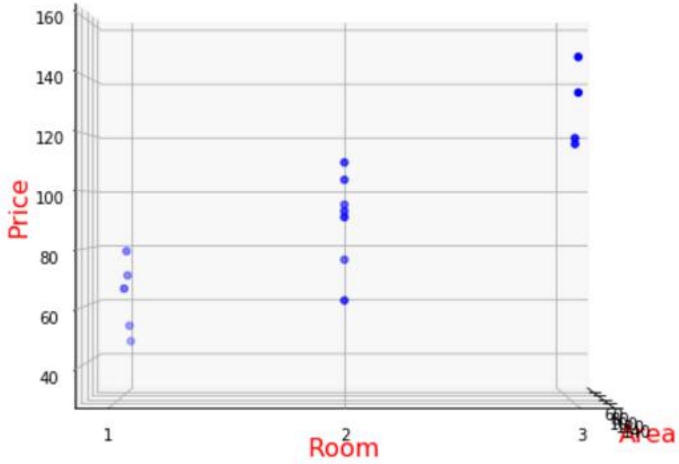
In [5]:

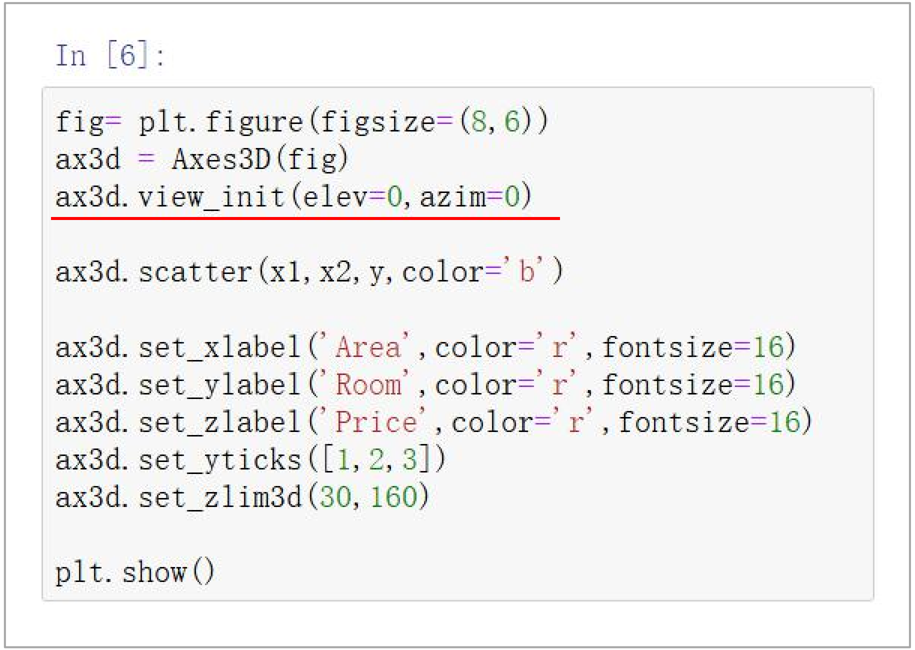
```
fig= plt.figure(figsize=(8,6))
ax3d = Axes3D(fig)
ax3d.view_init(elev=0,azim=-90)

ax3d.scatter(x1,x2,y,color='b')

ax3d.set_xlabel('Area',color='r',fontsize=16)
ax3d.set_ylabel('Room',color='r',fontsize=16)
ax3d.set_zlabel('Price',color='r',fontsize=16)
ax3d.set_yticks([1,2,3])
ax3d.set_zlim3d(30,160)

plt.show()
```

9.6.2 线性回归模型可视化

改变视角

view_init(elev,azim)



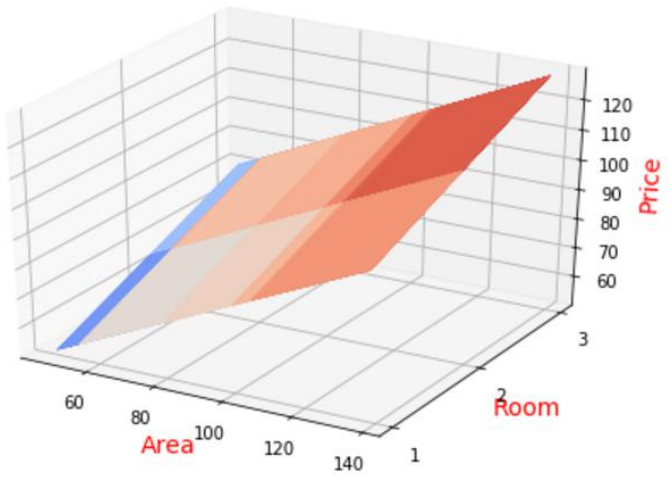


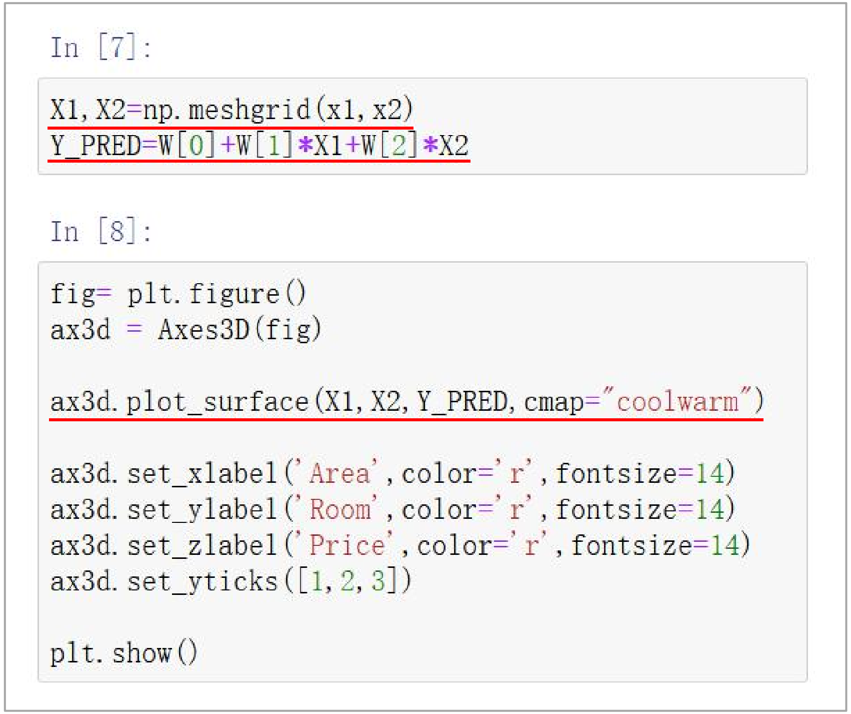
In [6]:

```
fig= plt.figure(figsize=(8,6))
ax3d = Axes3D(fig)
ax3d.view_init(elev=0,azim=0)

ax3d.scatter(x1,x2,y,color='b')

ax3d.set_xlabel('Area',color='r',fontsize=16)
ax3d.set_ylabel('Room',color='r',fontsize=16)
ax3d.set_zlabel('Price',color='r',fontsize=16)
ax3d.set_yticks([1,2,3])
ax3d.set_zlim3d(30,160)

plt.show()
```

9.6.2 线性回归模型可视化

绘制平面图



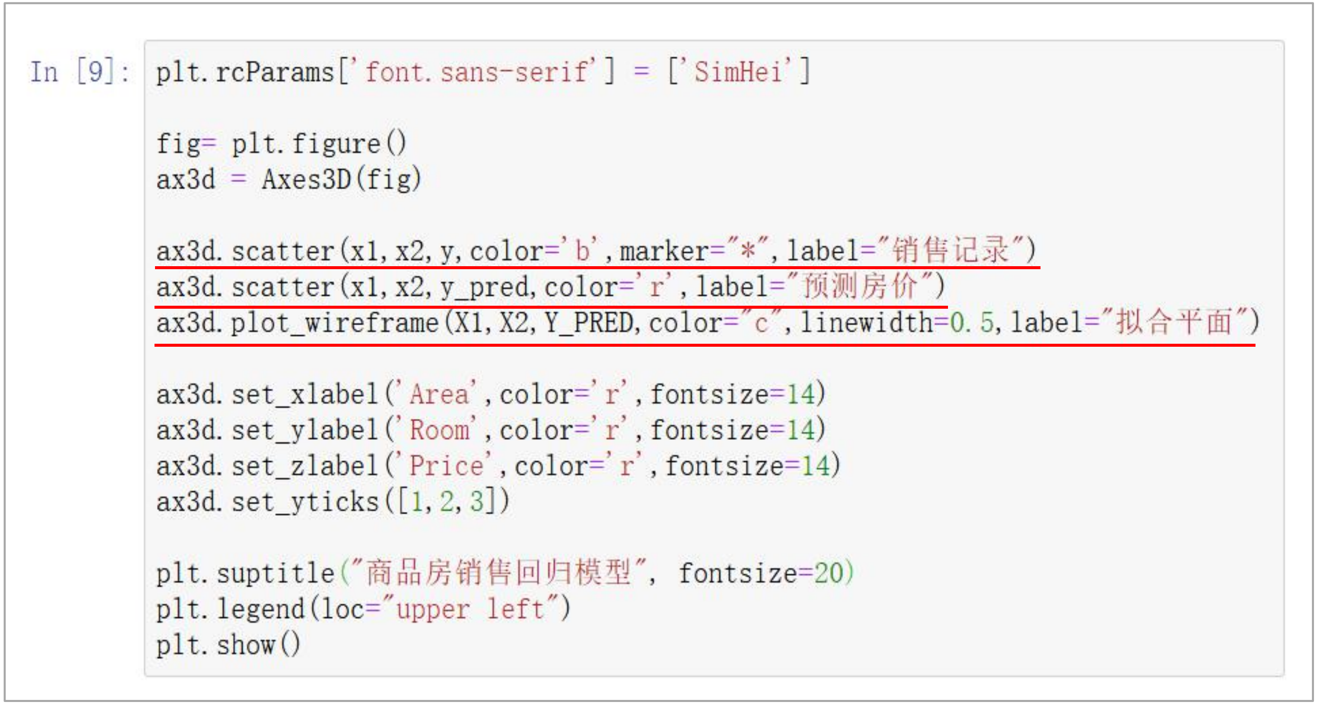


```
In [7]:
X1,X2=np.meshgrid(x1,x2)
Y_PRED=W[0]+W[1]*X1+W[2]*X2
```

```
In [8]:
fig= plt.figure()
ax3d = Axes3D(fig)

ax3d.plot_surface(X1,X2,Y_PRED,cmap="coolwarm")

ax3d.set_xlabel('Area',color='r',fontsize=14)
ax3d.set_ylabel('Room',color='r',fontsize=14)
ax3d.set_zlabel('Price',color='r',fontsize=14)
ax3d.set_yticks([1,2,3])

plt.show()
```

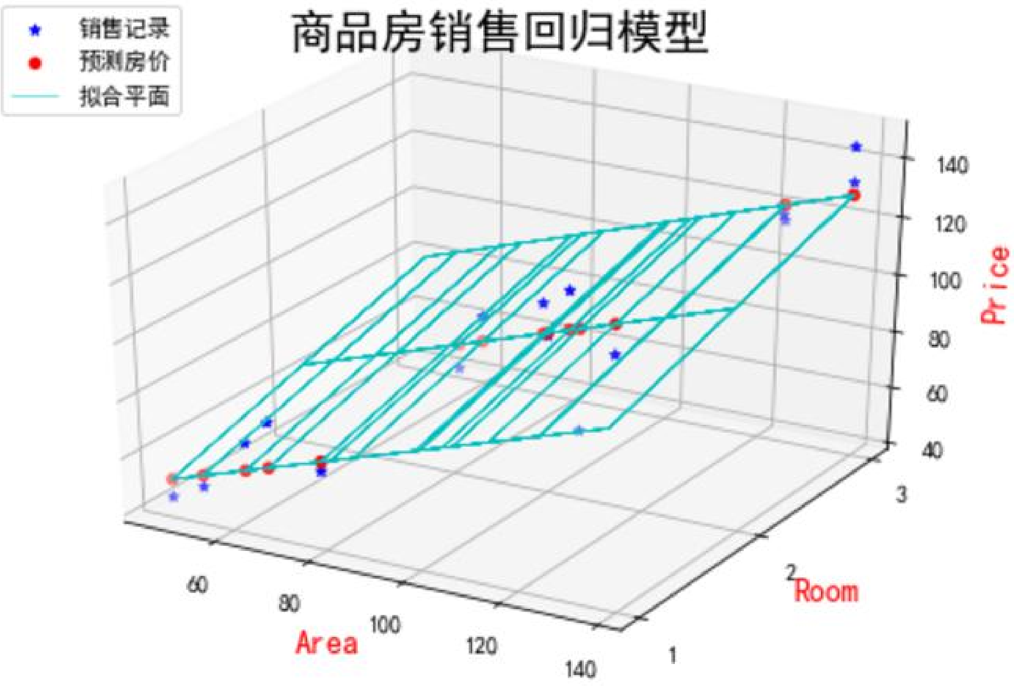
9.6.2 线性回归模型可视化

绘制散点图和线框图



```
In [9]: plt.rcParams['font.sans-serif'] = ['SimHei']

        fig= plt.figure()
        ax3d = Axes3D(fig)

        ax3d.scatter(x1,x2,y,color='b',marker="*",label="销售记录")
        ax3d.scatter(x1,x2,y_pred,color='r',label="预测房价")
        ax3d.plot_wireframe(X1,X2,Y_PRED,color="c",linewidth=0.5,label="拟合平面")

        ax3d.set_xlabel('Area',color='r',fontsize=14)
        ax3d.set_ylabel('Room',color='r',fontsize=14)
        ax3d.set_zlabel('Price',color='r',fontsize=14)
        ax3d.set_yticks([1,2,3])

        plt.suptitle("商品房销售回归模型", fontsize=20)
        plt.legend(loc="upper left")
        plt.show()
```

9.6.2 线性回归模型可视化


商品房销售回归模型
★ 销售记录
● 预测房价
拟合平面
Price
Area
Room