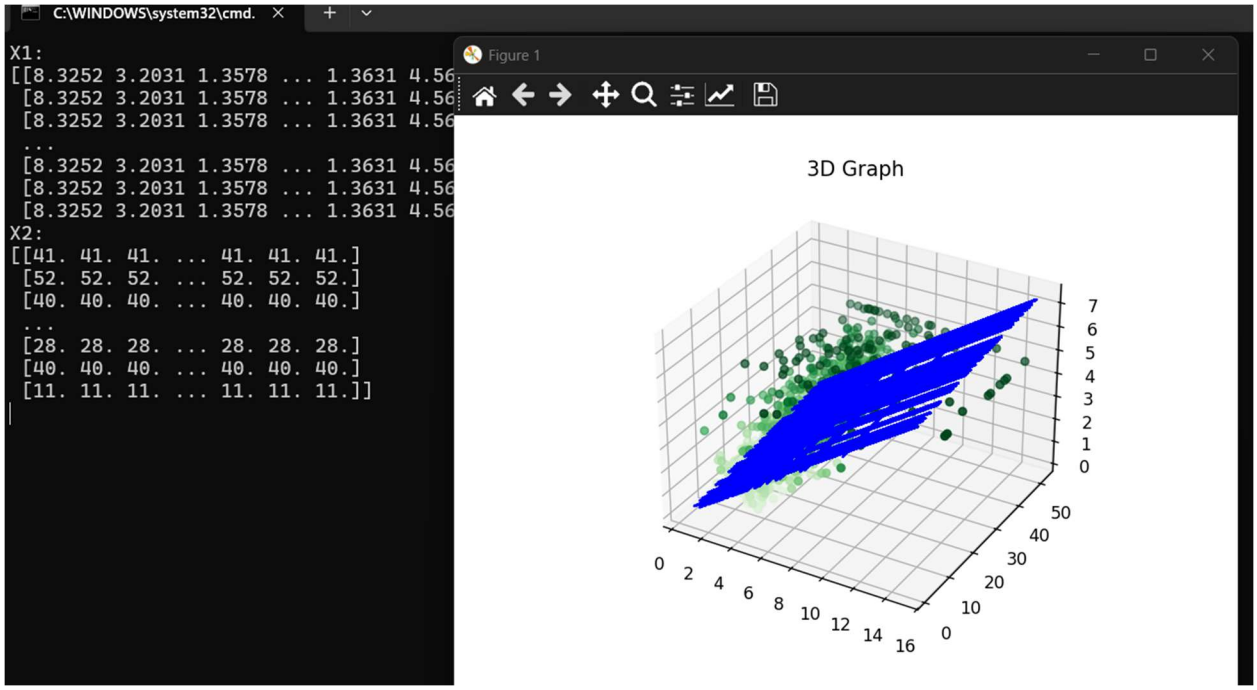
1.

```python
from distutils.errors import CCompilerError, CompileError
from lib2to3.pgen2.pgen import DFAState
import numpy as np
import pandas as pd
from sklearn.datasets import fetch_california_housing
from matplotlib import pyplot as plt
from scipy import stats
import seaborn as sns
from mpl_toolkits import mplot3d
from sklearn.linear_model import LinearRegression
from sklearn.preprocessing import StandardScaler

california_housing = fetch_california_housing(as_frame=True)

df = california_housing.frame
df2= df[::10]

X = np.array(df2.drop(['MedHouseVal'], axis=1))
y= np.array(df2['MedHouseVal'])

dp = np.array([8.3153, 41.0, 6.894423, 1.053714, 323.0, 2.533576, 37.88, -122.2

reg = LinearRegression()
reg.fit(X,y)


# Getting the coefficients of the regression
for i in range(8):
    print(f'Coeff {i+1}: {reg.coef_[i]}')

pred = reg.predict(dp)
print(f'Predicted MedHouseVal: {pred}')
```

```
C:\WINDOWS\system32\cmd.
Coeff 1: 0.44579261039787277
Coeff 2: 0.010155547250126132
Coeff 3: -0.1329213834447512
Coeff 4: 0.7728198065164754
Coeff 5: 1.2715335314532174e-05
Coeff 6: -0.11922642841114116
Coeff 7: -0.41493384531159244
Coeff 8: -0.42345226075103926
Predicted MedHouseVal: [4.1916238]
Press any key to continue . . .
```

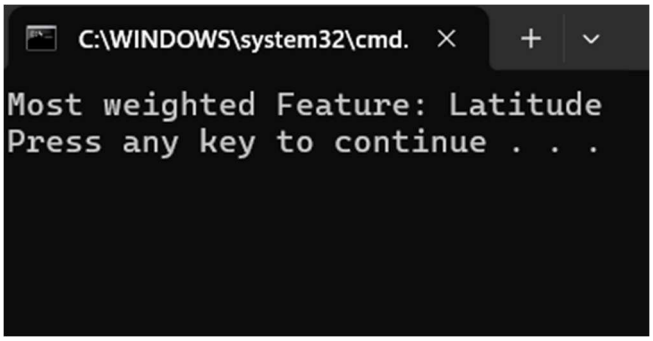
2.

```python
from distutils.errors import CCompilerError, CompileError
from lib2to3.pgen2.pgen import DFAState
import numpy as np
import pandas as pd
from sklearn.datasets import fetch_california_housing
from matplotlib import pyplot as plt
from scipy import stats
import seaborn as sns
from mpl_toolkits import mplot3d
from sklearn.linear_model import LinearRegression
from sklearn.preprocessing import StandardScaler

california_housing = fetch_california_housing(as_frame=True)

df = california_housing.frame
df2= df[::10]

X = np.array(df2.drop(['MedHouseVal'], axis=1))
y= np.array(df2['MedHouseVal'])

# Exercise2

X2 = X[:, :2]

reg = LinearRegression()
reg.fit(X2,y)

b1 = reg.coef_[0]
b2 = reg.coef_[1]

X_1, X_2 = np.meshgrid(X2[:, 0], X2[:, 1])
print(f'X1:\n{X_1}')
print(f'X2:\n{X_2}')
Z = reg.intercept_ + b1*X_1 + b2*X_2
#3D plot
fig = plt.figure()
ax = plt.axes(projection = '3d')
ax.plot_wireframe(X_1, X_2, Z, color = 'blue')
#3D scattet plot (data points)
ax.scatter3D(X2[:, 0], X2[:, 1], y, c=y, cmap='Greens')
ax.set_title('3D Graph')
plt.show()
```


C:\WINDOWS\system32\cmd.
X1:
[[8.3252 3.2031 1.3578 ... 1.3631 4.56
[8.3252 3.2031 1.3578 ... 1.3631 4.56
[8.3252 3.2031 1.3578 ... 1.3631 4.56
...
[8.3252 3.2031 1.3578 ... 1.3631 4.56
[8.3252 3.2031 1.3578 ... 1.3631 4.56
[8.3252 3.2031 1.3578 ... 1.3631 4.56
X2:
[[41. 41. 41. ... 41. 41. 41.]
[52. 52. 52. ... 52. 52. 52.]
[40. 40. 40. ... 40. 40. 40.]
...
[28. 28. 28. ... 28. 28. 28.]
[40. 40. 40. ... 40. 40. 40.]
[11. 11. 11. ... 11. 11. 11.]]
Figure 1
3D Graph
0
1
2
3
4
5
6
7
0
2
4
6
8
10
12
14
16
0
10
20
30
40
50

3.

```python
from distutils.errors import CCompilerError, CompileError
from lib2to3.pgen2.pgen import DFAState
import numpy as np
import pandas as pd
from sklearn.datasets import fetch_california_housing
from matplotlib import pyplot as plt
from scipy import stats
import seaborn as sns
from mpl_toolkits import mplot3d
from sklearn.linear_model import LinearRegression
from sklearn.preprocessing import StandardScaler

california_housing = fetch_california_housing(as_frame=True)

df = california_housing.frame
df2= df[::10]

X = np.array(df2.drop(['MedHouseVal'], axis=1))
y= np.array(df2['MedHouseVal'])


#exercise 3

full_X = np.array(df.drop(['MedHouseVal'], axis=1))
full_y= np.array(df['MedHouseVal'])

scaler = StandardScaler()
ScaledData = scaler.fit_transform(full_X)

reg = LinearRegression()
reg.fit(ScaledData,full_y)

coefficients = reg.coef_
max_coeff_index = abs(coefficients).argmax() #index of max coeffient(absolute value)

print(f'Most weighted Feature: {df.columns[max_coeff_index]}')

```



4.

```python
import numpy as np
import pandas as pd
from sklearn.datasets import fetch_california_housing
from matplotlib import pyplot as plt
from scipy import stats
import seaborn as sns
from mpl_toolkits import mplot3d
from sklearn.linear_model import LinearRegression
from sklearn.preprocessing import StandardScaler

california_housing = fetch_california_housing(as_frame=True)

df = california_housing.frame
df2= df[::10]

X = np.array(df2.drop(['MedHouseVal'], axis=1))
y= np.array(df2['MedHouseVal'])

#exercise 4

data = pd.DataFrame(df2.drop(['Longitude', 'Latitude'], axis=1))
data_x = pd.DataFrame(df2.drop(['Longitude', 'Latitude', 'MedHouseVal'], axis=1))
sns.pairplot(data=data, vars=data_x, hue='MedHouseVal')
plt.show()
```