In [13]:

```
pip install tensorflow
```

```
Collecting tensorflow
  Downloading tensorflow-2.13.0rc0-cp311-cp311-macosx_12_0_arm64.whl
(2.0 kB)
Collecting tensorflow-macos==2.13.0-rc0
  Downloading tensorflow_macos-2.13.0rc0-cp311-cp311-macosx_12_0_arm6
4.whl (189.3 MB)
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 189.3/189.3 MB 5.2 MB/s e
ta 0:00:0000:0100:01
Collecting absl-py>=1.0.0
  Downloading absl_py-1.4.0-py3-none-any.whl (126 kB)
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 126.5/126.5 kB 7.3 MB/s e
ta 0:00:00
Collecting astunparse>=1.6.0
  Downloading astunparse-1.6.3-py2.py3-none-any.whl (12 kB)
Collecting flatbuffers>=23.1.21
  Downloading flatbuffers-23.5.9-py2.py3-none-any.whl (26 kB)
Collecting gast<=0.4.0,>=0.2.1
  Downloading gast-0.4.0-py3-none-any.whl (9.8 kB)
Collecting google-pasta>=0.1.1
```

In [7]:

```
import pandas as pd
import matplotlib.pyplot as plt
```

In [8]:

```
df = pd.read_csv("housing.csv")
df.head()
```

Out[8]:

	RM	LSTAT	PTRATIO	MEDV
0	6.575	4.98	15.3	504000.0
1	6.421	9.14	17.8	453600.0
2	7.185	4.03	17.8	728700.0
3	6.998	2.94	18.7	701400.0
4	7.147	5.33	18.7	760200.0

In [9]:

```
# Divide data in X and Y

X = df.loc[:, ["RM", "LSTAT", "PTRATIO"]]
Y = df.loc[:, ["MEDV"]]

X.shape, Y.shape
```

Out[9]:

```
((489, 3), (489, 1))
```

In [11]:

```
# Train test split data

from sklearn.model_selection import train_test_split

x_train, x_test, y_train, y_test = train_test_split(X,Y, test_size=0.25, random_stat
```

In [12]:

```
# Normalizing the data

from sklearn.preprocessing import StandardScaler

scaler = StandardScaler()

x_train = scaler.fit_transform(x_train)
x_test = scaler.fit_transform(x_test)
```

In [24]:

```
# MODEL -> DEEP LEARNING PART

import tensorflow as tf

# get model
model = tf.keras.models.Sequential()


# add layers to model

'''
3 -> in input shape indicates the 3 columns from input


Doubts:
- tf.keras.layers.Dense => can we make this a variable and then use it?
- why does the output layer require linear activation function

'''



model.add(tf.keras.layers.Dense(128, input_shape=(3,), activation='relu', name='inpu
model.add(tf.keras.layers.Dense(64, activation='relu', name='layer1'))
model.add(tf.keras.layers.Dense(1, activation='linear', name='output'))

model.compile(optimizer='adam', loss='mse', metrics=['mae'])
model.summary()
```

```
Model: "sequential_8"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 input (Dense)               (None, 128)               512

 layer1 (Dense)              (None, 64)                8256

 output (Dense)              (None, 1)                 65

=================================================================
Total params: 8833 (34.50 KB)
Trainable params: 8833 (34.50 KB)
Non-trainable params: 0 (0.00 Byte)
_________________________________________________________________
```

In [25]:

```
model.fit(x_train, y_train, epochs=100, validation_split=0.05)
```

```
Epoch 1/100
11/11 [==============================] - 0s 6ms/step - loss: 236025888
768.0000 - mae: 455517.9375 - val_loss: 187415773184.0000 - val_mae: 4
02535.7500
Epoch 2/100
11/11 [==============================] - 0s 1ms/step - loss: 236024987
648.0000 - mae: 455517.0000 - val_loss: 187414806528.0000 - val_mae: 4
02534.5000
Epoch 3/100
11/11 [==============================] - 0s 1ms/step - loss: 236023758
848.0000 - mae: 455515.7188 - val_loss: 187413397504.0000 - val_mae: 4
02532.7188
Epoch 4/100
11/11 [==============================] - 0s 1ms/step - loss: 236021891
072.0000 - mae: 455513.7812 - val_loss: 187411365888.0000 - val_mae: 4
02530.0938
Epoch 5/100
11/11 [==============================] - 0s 1ms/step - loss: 236019187
712.0000 - mae: 455510.9062 - val_loss: 187408416768.0000 - val_mae: 4
```

In [27]:

```
y_pred = model.predict(x=x_test)
```

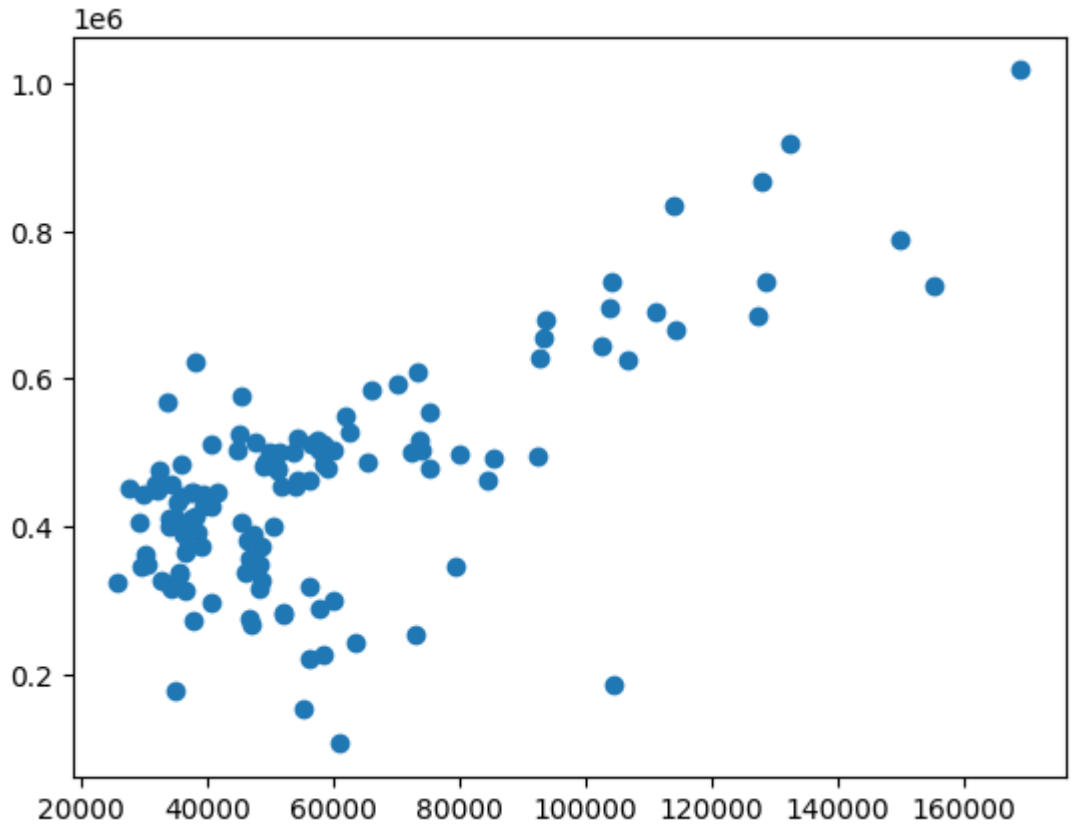
```
4/4 [==============================] - 0s 1ms/step
```

In [28]:

```
plt.scatter(y_pred, y_test)
```

Out[28]:

```
<matplotlib.collections.PathCollection at 0x294391350>
```



In [ ]: