M9-L1 Problem 2

Recall the von Mises stress prediction problem from the module 6 homework. In this problem, you will compute the R^2 score for a few model predictions for a single shape in this dataset. You will also plot the predicted-vs-actual stress for each model.

In [1]:
```
import numpy as np
import matplotlib.pyplot as plt
from sklearn.metrics import r2_score

float32 = np.float32
```

In [2]:
```
xs= np.load("data/L1P2/xs.npy")
ys= np.load("data/L1P2/ys.npy")
gt = np.load("data/L1P2/gt.npy")
model1= np.load("data/L1P2/model1.npy")
model2= np.load("data/L1P2/model2.npy")
model3= np.load("data/L1P2/model3.npy")
```

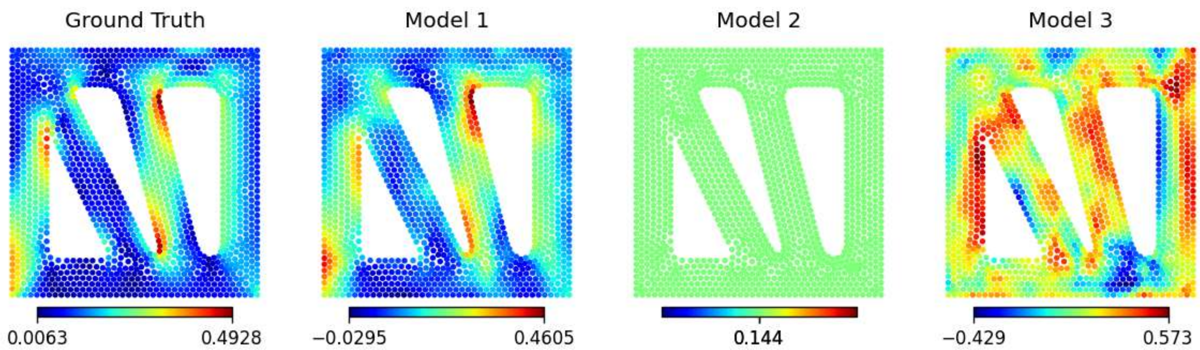
Visualizing data

Run the following cell to load the data and visualize the 3 model predictions.

- `gt` is the ground truth von Mises stress vector
- `model1` is the vector of stress predictions for model 1
- `model2` is the vector of stress predictions for model 2
- `model3` is the vector of stress predictions for model 3

In [3]:
```
def plot_shape(x, y, stress, lims=None):

    if lims is None:
        lims = [min(stress),max(stress)]

    plt.scatter(x,y,s=5,c=stress,cmap="jet",vmin=lims[0],vmax=lims[1])
    plt.colorbar(orientation="horizontal", shrink=.75, pad=0,ticks=lims)
    plt.axis("off")
    plt.axis("equal")

def plot_all(x, y, gt, model1, model2, model3):
    plt.figure(figsize=[12,3.2], dpi=120)
    plt.subplot(141)
    plot_shape(x, y, gt)
    plt.title("Ground Truth")

    plt.subplot(142)
    plot_shape(x, y, model1)
    plt.title("Model 1")
```

```
    plt.subplot(143)
    plot_shape(x, y, model2)
    plt.title("Model 2")

    plt.subplot(144)
    plot_shape(x, y, model3)
    plt.title("Model 3")

    plt.show()

plot_all(xs, ys, gt, model1, model2, model3)
```



Computing R^2

Calculate the R^2 value for each model and print the results.

In [11]:
```
# YOUR CODE GOES HERE
r2_1 = r2_score(gt,model1)
r2_2 = r2_score(gt,model2)
r2_3 = r2_score(gt,model3)

print(r2_1)
print(r2_2)
print(r2_3)
```

```
0.8727993965148926
0.0
-3.0451393127441406
```

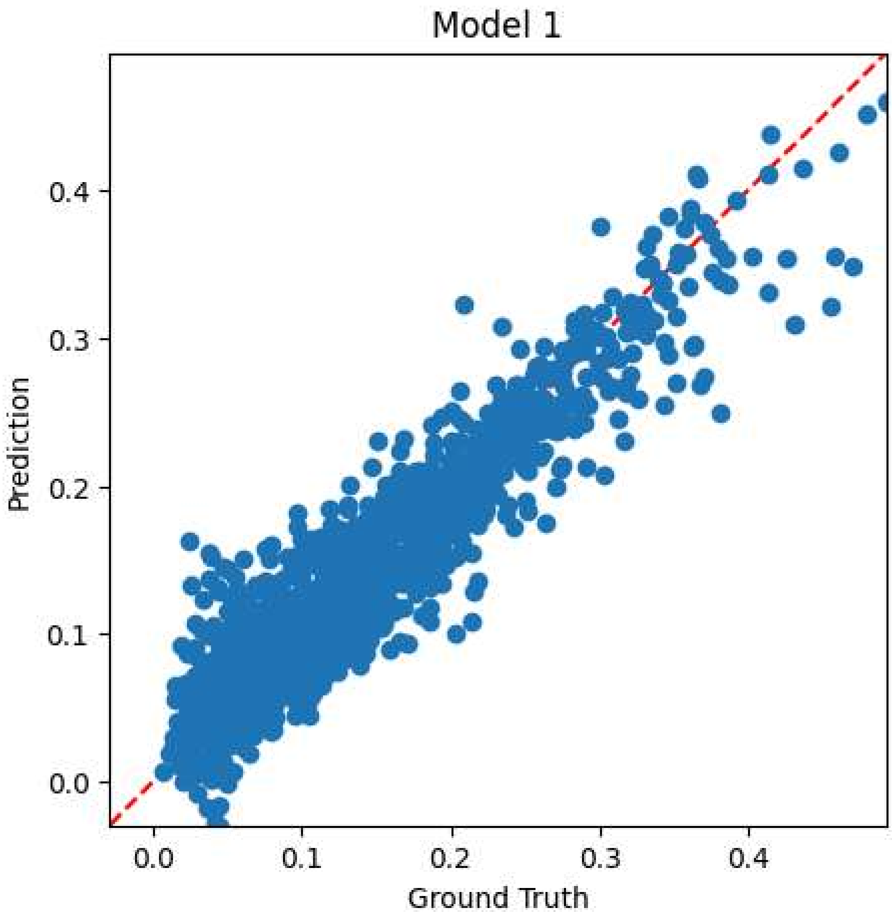
Plotting predictions vs ground truth

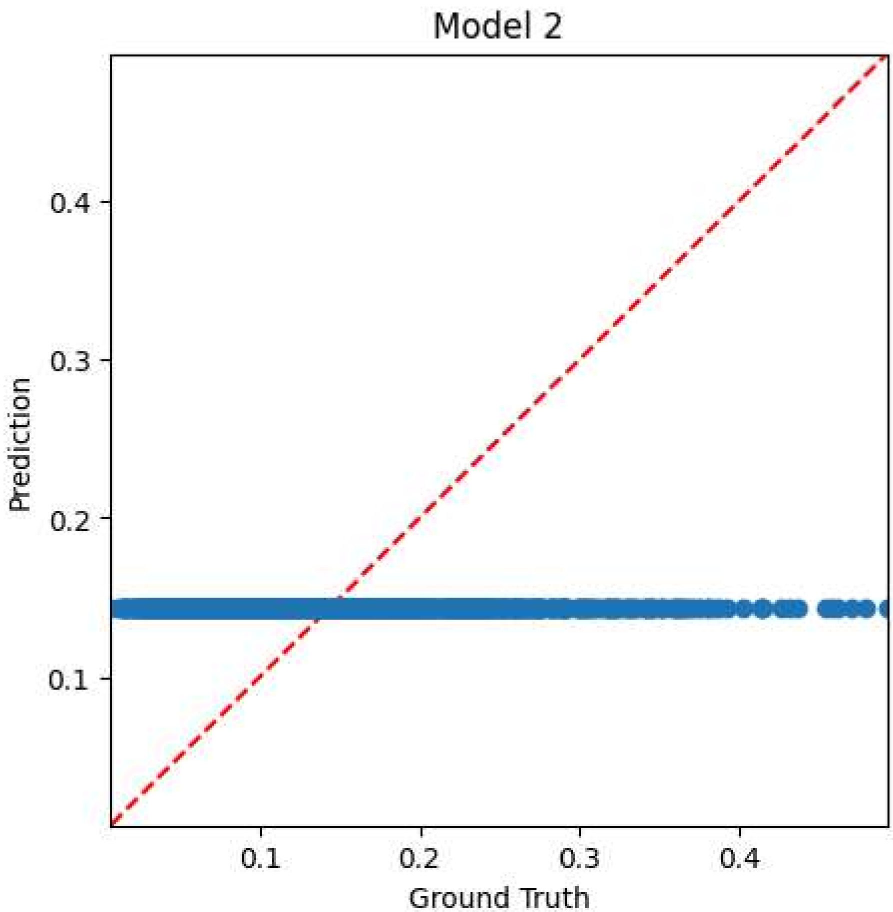
Complete the function definition below for `plot_r2(gt, pred, title)`

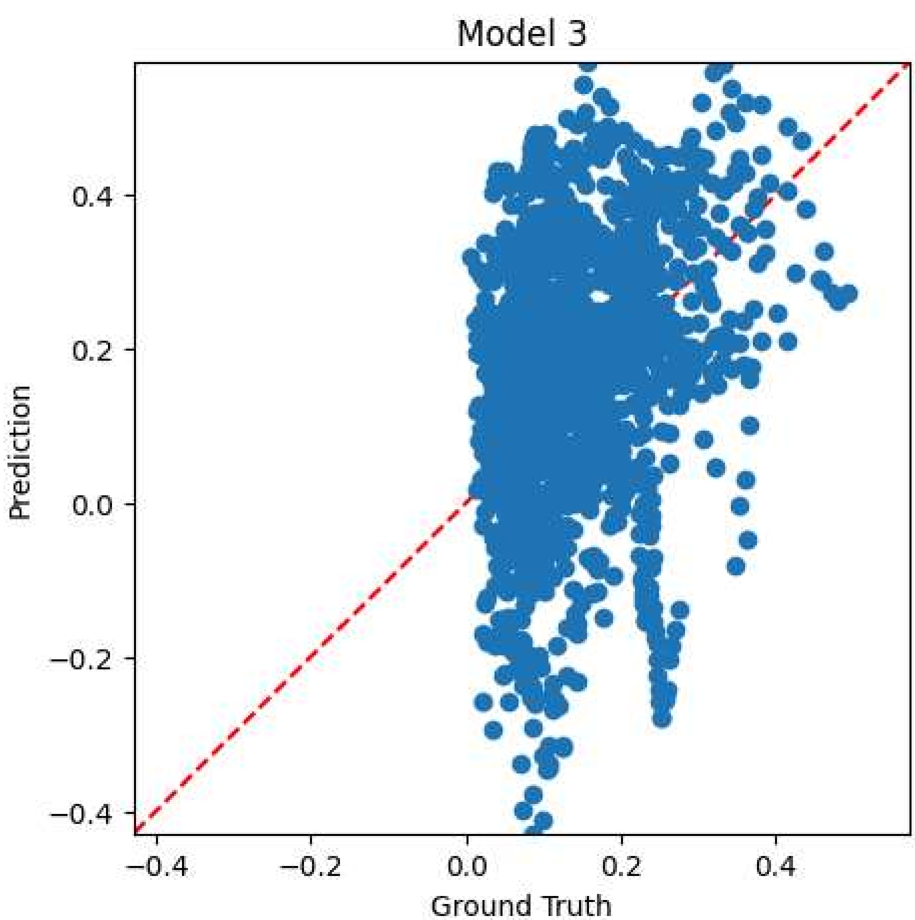
Then create plots for all 3 models.

In [10]:
```
def plot_r2(gt, pred, title):
    plt.figure(figsize=[5,5])

    # YOUR CODE GOES HERE

    plt.plot([-1000,1000], [-1000,1000],"r--")
```

```
    plt.plot(gt,pred,'o')
    all = np.concatenate([gt, pred])
    plt.xlim(np.min(all), np.max(all))
    plt.ylim(np.min(all), np.max(all))
    plt.xlabel("Ground Truth")
    plt.ylabel("Prediction")
    plt.title(title)
    plt.show()

plot_r2(gt, model1,"Model 1")
plot_r2(gt, model2,"Model 2")
plot_r2(gt, model3,"Model 3")
```




Model 2
0.4
0.3
0.2
0.1
Prediction
0.1
0.2
0.3
0.4
Ground Truth


Model 3
0.4
0.2
0.0
−0.2
−0.4
Prediction
−0.4
−0.2
0.0
0.2
0.4
Ground Truth

Questions

1. Model 2 has an R^2 of exactly 0. Why?
2. Model 3 has an R^2 less than 0. What does this mean?

1. Model2 appears to be a straight line, therefore there's no variance withnin the data, led to R^2 being 0

2. if R^2 is less than zero, it means the model performs worse than using the mean to predict in a linear line.