2.

(1) $\frac{1}{1+e^{-z}} - \frac{e^{-z}}{1+e^{-z}} > 0,\ so\ e^{-z} < 1, so\ z > 0$

$\{\mathbf{X}|1 + 2x_1 + 3x_2 > 0\}$

(2) $\frac{1}{1+e^{-z}} > 0.8, so\ e^{-z} < \frac{1}{4}\ , so\ \ z > ln4$

$\{\mathbf{X}|1 + 2x_1 + 3x_2 > ln4\}$

(3) $z > ln4$, so $1 + 2x_1 + 3x_2 > ln4, and\ x_2 = 0.5, so\ x_1 > (ln4 - 2.5)/2$

$\{\mathbf{X}|x_1 > (ln4 - 2.5)/2\}$

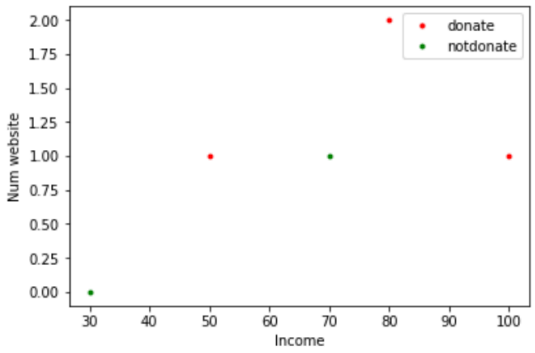
3.

```
x = np.array([[30,0],[50,1],[70,1],[80,2],[100,1]])
y = np.array([0,1,0,1,1])
Ido =np.where(y==1)[0]
Idont = np.where(y == 0)[0]
plt.plot(x[Ido,0], x[Ido,1],'r.')
plt.plot(x[Idont,0], x[Idont,1],'g.')
plt.xlabel('Income')
plt.ylabel('Num website')
plt.legend(['donate', 'notdonate'], loc = "upper right")
regr = linear_model.LinearRegression();
xs = preprocessing.scale(x)
# print(xs)
y1 = y - 0.5
#print(y1)
regr.fit(xs, y1)
W = regr.coef_
print("W",W)
bias = regr.intercept_
print("bias",bias)
yhat = regr.predict(xs)
# print(yhat)
yhati = (yhat >= 0).astype(int)
# print(yhati)
acc = np.mean(yhati == y)
print("Accuracy:",acc)
```

```
W [0.101295   0.24995249]
bias 0.1
Accuracy: 0.8
```

```
/anaconda3/lib/python3.6/site-packages/sklearn/utils/validation.py:475: DataConversionWar
int64 was converted to float64 by the scale function.
  warnings.warn(msg, DataConversionWarning)
```

(1)



(2)

if P is the smallest, then z should be the smallest, so sample 0 make it least.

(3)

In part(b), it doesn't change anything, because in that way, new z will be α multiple of the previous z , but α is positive, so sign of new z is the same with the old z, so they do not change.

In part(c), P will become larger than previous which means in this calculation, it shows the more people are willing to donate money.

4. $z = \beta_0 + \beta_1 x_1 + \beta_2 x_2$

(1) z = -0.5 $p = \frac{1}{1+e^{\frac{1}{2}}}$

(2) p = 0.5, so z = 0, so hours = 50