This homework is done by Tianwei Mo (Bill).

1.

a) Screenshot:

```python
# a
row_tr, col = Xtr.shape
row_ts, col = Xts.shape
Xtr_f = np.zeros((row_tr, 1))
Xts_f = np.zeros((row_ts, 1))
r2s = np.zeros(col)
for f in range(col):
    Xtr_f = Xtr[:, f].reshape(-1, 1)
    Xts_f = Xts[:, f].reshape(-1, 1)
    model = LinearRegression().fit(Xtr_f, ytr)
    yhat = model.predict(Xts_f)
    r2s[f] = r2_score(yts, yhat)
print('best feature: ', np.argmax(r2s))
print('best r2: ', np.max(r2s))
```

Code:


```python
# a
row_tr, col = Xtr.shape
row_ts, col = Xts.shape
Xtr_f = np.zeros((row_tr, 1))
Xts_f = np.zeros((row_ts, 1))
r2s = np.zeros(col)
for f in range(col):
    Xtr_f = Xtr[:, f].reshape(-1, 1)
    Xts_f = Xts[:, f].reshape(-1, 1)
    model = LinearRegression().fit(Xtr_f, ytr)
    yhat = model.predict(Xts_f)
    r2s[f] = r2_score(yts, yhat)
print('best feature: ', np.argmax(r2s))
print('best r2: ', np.max(r2s))
```


b) Screenshot:

```python
# b
row_tr, col = Xtr.shape
row_ts, col = Xts.shape
Xtr_f = np.zeros((row_tr, 2))
Xts_f = np.zeros((row_ts, 2))
r2s = np.zeros((col, col))
for f1 in range(col):
    for f2 in range(col):
        if f1 == f2:
            r2s[f1, f2] = -np.inf
            continue
        Xtr_f[:, 0] = Xtr[:, f1]
        Xts_f[:, 0] = Xts[:, f1]
        Xtr_f[:, 1] = Xtr[:, f2]
        Xts_f[:, 1] = Xts[:, f2]
        model = LinearRegression().fit(Xtr_f, ytr)
        yhat = model.predict(Xts_f)
        r2s[f1, f2] = r2_score(yts, yhat)
max = np.max(r2s)
max_x, max_y = np.where(r2s == max)
print('best feature 1: ', max_x + 1)
print('best feature 2: ', max_y + 1)
print('best r2: ', max)
```

Code:

```python
# b
row_tr, col = Xtr.shape
row_ts, col = Xts.shape
Xtr_f = np.zeros((row_tr, 2))
Xts_f = np.zeros((row_ts, 2))
r2s = np.zeros((col, col))
for f1 in range(col):
    for f2 in range(col):
        if f1 == f2:
            r2s[f1, f2] = -np.inf
            continue
        Xtr_f[:, 0] = Xtr[:, f1]
        Xts_f[:, 0] = Xts[:, f1]
        Xtr_f[:, 1] = Xtr[:, f2]
        Xts_f[:, 1] = Xts[:, f2]
        model = LinearRegression().fit(Xtr_f, ytr)
        yhat = model.predict(Xts_f)
        r2s[f1, f2] = r2_score(yts, yhat)
max = np.max(r2s)
max_x, max_y = np.where(r2s == max)
print('best feature 1: ', max_x + 1)
```

```
print('best feature 2: ', max_y + 1)
print('best r2: ', max)
```

c) I need to call the fit function $\binom{p}{k} = \frac{p!}{k!(p-k)!}$ Times. For $k = 10$ and $p = 1000$, I need to call the fit function $2.6341e + 23$ times.

2.

a) $\phi(w) = 0$

b) $\phi(w) = \sum_j -w_j$

c) $\phi(w) = \sum_j (w_j - w_{j-1})^2$

d) $\phi(w) = \sum_j w_j - w_{j-1}$

3.

a) The scale of features varies largely, which will result in a bad regularization.

b)

$$\hat{y} = \bar{y} + \sigma_y \hat{u} = \bar{y} + \sigma_y (a_1 z_1 + a_2 z_2) = \beta_0 + \beta_1 x_1 + \beta_2 x_2$$

$$\bar{y} + \sigma_y \left(a_1 \frac{x_1 - \bar{x}_1}{s_1} + a_2 \frac{x_2 - \bar{x}_2}{s_2} \right) = \beta_0 + \beta_1 x_1 + \beta_2 x_2$$

$$\bar{y} - \sigma_y \left(a_1 \frac{\bar{x}_1}{s_1} + a_2 \frac{\bar{x}_2}{s_2} \right) + \frac{\sigma_y a_1}{s_1} x_1 + \frac{\sigma_y a_2}{s_2} x_2 = \beta_0 + \beta_1 x_1 + \beta_2 x_2$$

$$\beta_0 = \bar{y} - \sigma_y \left(a_1 \frac{\bar{x}_1}{s_1} + a_2 \frac{\bar{x}_2}{s_2} \right) = 235$$

$$\beta_1 = \frac{\sigma_y a_1}{s_1} = 0.004$$

$$\beta_2 = \frac{\sigma_y a_2}{s_2} = 3$$

4.

```python
Xscal = StandardScaler()
yscal = StandardScaler()
Xtr = Xscal.fit_transform(Xtr)
ytr = yscal.fit_transform(ytr[:, None])
Xts = Xscal.transform(Xts)
yts = yscal.transform(yts[:, None])

model = LinearRegression().fit(Xtr, ytr)

yhat = model.predict(Xts)

rss = np.sum((yts-yhat)**2)
```

5.

```
p = np.linspace(a, b, 100)
Ztr = np.exp(-p*xtr)
Zts = np.exp(-p*xts)
model = Lasso(lam).fit(Ztr, ytr)
beta = model.coef_
yhat = model.predict(Zts)
rss = np.sum((yts-yhat)**2)
print(rss)
rank = np.argsort(beta)
best = rank[-3:]
print('Best 3 alpha: {}'.format(p[best]))
print('Best 3 beta: {}'.format(beta[best]))
```

6.

i) Assume $w > 0, |w| = w$.

$$J(w) = \frac{1}{2}w^2 + (\lambda - y)w + y^2$$

Let $J'(w) = 0$,

$$w + \lambda - y = 0$$
$$w = y - \lambda$$

If $y > \lambda$, $w > 0$, w_{min} exist.

ii) Assume $w < 0, |w| = -w$.

$$J(w) = \frac{1}{2}w^2 - (\lambda + y)w + y^2$$

Let $J'(w) = 0$,

$$w - \lambda - y = 0$$
$$w = \lambda + y > 0$$

Which contradicts to our assumption.

iii) When $y \leq \lambda$, the only possibility of w_{min} is $w_{min} = 0$.