1.(a)

```
model = LinearRegression();
columns = Xtr.shape[1];
rsq = [];
for i in range(columns):
    Xtr_red = Xtr[:,i]
    Xts_red = Xts[:,i]
    model.fit(Xtr_red,ytr)
    yhat = model.predit(Xts_red)
    rsq.append(rs_score(yts, yhat))
feature = np.argmax(rsq)
```

(b)

```
model = LinearRegression();
columns = Xtr.shape[1];
rsq = [[]]
for i in range(columns):
    for j in range(i+1, columns):
        Xtr_red = Xtr[:,[i,j]]
        Xts_red = Xts[:,[i,j]]
        model.fit(Xtr_red,ytr)
        yhat = model.predit(Xts_red)
        rsq[i].append(rs_score(yts, yhat))
feature1, feature2 = np.argmax(rsq)
```

(c)

best k of p features: C_p^k

when k =10, p=1000 , it will need to call C_{1000}^{10} times fit function.

2.

(a) $\phi(\omega) = \sum_{i=0}^{d} |\omega|$

(b) $\phi(\omega) = \sum_{i=0}^{d} -\omega$

(c) $\phi(w) = \sum_{i=0}^{d} (\omega_i - \omega_{i+1})^2$

(d) $\phi(w) = \sum_{i=0}^{d} |\omega_i - \omega_{i+1}|$

4.

```
Xtr_norm = (Xtr-np.mean(Xtr))/np.std(Xtr)
ytr_norm = (ytr - np.mean(ytr))/np.std(ytr)
model = someModel()
model.fit(Xtr_norm, ytr_norm)
yhat = model.predict(yts)
rss_test = ((yhat - yts)**2)/(np.std(yts)**2)
```

5.

```
alpha = np.random.uniform(a, b, 100)
Xtr_new = np.exp(alpha * xtr)
model = Lasso(lam = lam)
beta = model.fit(Xtr_new, ytr)
yhat = model.predict(yts)
rss = np.mean((yhat-yts)**2)/(np.std(yts)**2)
beta = beta[np.argsort(-beta)]
beta_max3 = beta[:3]
```