In [1]:

```
%matplotlib inline
import warnings
warnings.filterwarnings("ignore")
from sklearn.model_selection import train_test_split
from sklearn.metrics import accuracy_score
import pandas as pd
import numpy as np
from sklearn.datasets import load_iris
from sklearn.linear_model import SGDClassifier
from sklearn.model_selection import GridSearchCV
import seaborn as sns
import matplotlib.pyplot as plt
```

In [2]:

```
data = pd.read_csv('task_d.csv')
data = pd.DataFrame(data)
print(data.columns)
data.head()
features=['x', 'y', 'z', 'x*x', '2*y', '2*z+3*x*x', 'w']
X = data.drop(['target'], axis=1).values
X=pd.DataFrame(X,columns=[str(i) for i in features])
Y = data['target'].values
len(X)
l1=list(X.columns)
print(l1)
```

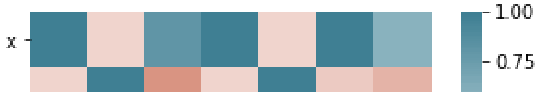
```
Index(['x', 'y', 'z', 'x*x', '2*y', '2*z+3*x*x', 'w', 'target'], dtype='object')
['x', 'y', 'z', 'x*x', '2*y', '2*z+3*x*x', 'w']
```

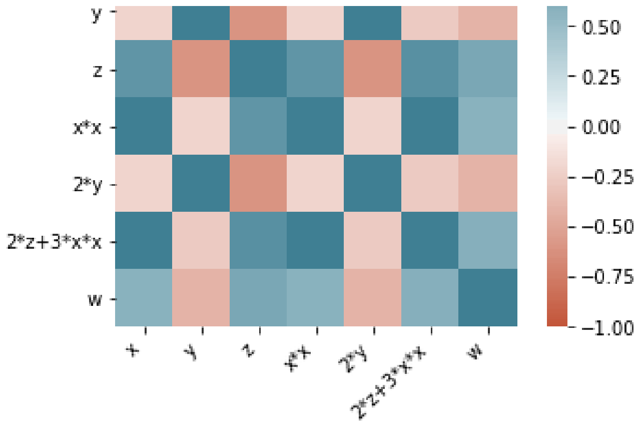
In [3]:

```
d=X.corr()
print(d) #corelation between features
ax = sns.heatmap(
    d,
    vmin=-1, vmax=1, center=0,
    cmap=sns.diverging_palette(20, 220, n=200),
    square=True
)
ax.set_xticklabels(
    ax.get_xticklabels(),
    rotation=45,
    horizontalalignment='right'
);
```

```
                  x         y         z       x*x       2*y  2*z+3*x*x  \
x          1.000000 -0.205926  0.812458  0.997947 -0.205926   0.996252
y         -0.205926  1.000000 -0.602663 -0.209289  1.000000  -0.261123
z          0.812458 -0.602663  1.000000  0.807137 -0.602663   0.847163
x*x        0.997947 -0.209289  0.807137  1.000000 -0.209289   0.997457
2*y       -0.205926  1.000000 -0.602663 -0.209289  1.000000  -0.261123
2*z+3*x*x  0.996252 -0.261123  0.847163  0.997457 -0.261123   1.000000
w          0.583277 -0.401790  0.674486  0.583803 -0.401790   0.606860

                  w
x          0.583277
y         -0.401790
z          0.674486
x*x        0.583803
2*y       -0.401790
2*z+3*x*x  0.606860
w          1.000000
```





In [4]:

```
param_grid = { 'loss': ['log'], 'penalty': ['elasticnet','l2','l1'], 'alpha': ([10,1,100
,0.1,0.01,0.001,0.0001]) }
mod=SGDClassifier(loss='log',random_state=0)
grid = GridSearchCV(estimator=mod, param_grid=param_grid,n_jobs=-1,scoring='accuracy',cv
=3)
grid.fit(X,Y)
grid.best_params_
```

Out[4]:

```
{'alpha': 1, 'loss': 'log', 'penalty': 'elasticnet'}
```

In [5]:

```
best_model=SGDClassifier(loss='log',penalty='elasticnet',random_state=0,alpha=1)
best_model.fit(X,Y)
print("best _model_accuracy :",best_model.score(X,Y)*100)
print("weights :",best_model.coef_)
w=best_model.coef_
ac=best_model.score(X,Y)
```

```
best _model_accuracy : 98.0
weights : [[ 0.09656279 -0.10745327  0.2165494   0.09143185 -0.10745327  0.11209243
   0.06853059]]
```

In [6]:

```
mean=0
sigma=0.01

noise=np.random.normal(mean,sigma,[X.shape[0],X.shape[1]]) #add noise
X_=X+noise
best_model.fit(X_,Y)
print("best _model_accuracy_edited :",best_model.score(X_,Y)*100)
print("weights :",best_model.coef_)
w_=best_model.coef_
ac2=best_model.score(X_,Y)
```

```
best _model_accuracy_edited : 98.0
weights : [[ 0.09624561 -0.10729912  0.216327    0.09186088 -0.10749667  0.1119158
   0.0688655 ]]
```

In [18]:

```
print("difference of accuracy :",ac-ac2)
d=abs((w-w_))
print("absolute change between each value of W and W : ",d)
x=[]
index=[]
n=4
for i in d:#top 4 features
    i=list(i)
    i.sort()
    for k in (i[-4:]):
        x.append(k)
```

```
index=[]
for y in x:
    for i in d:
        i=list(i)
        index.append(i.index(y))
print("top 4 features :",[l1[i] for i in index])
```

```
difference of accuracy : 0.0
absolute change between each value of W and W :  [[3.17178949e-04 1.54140721e-04 2.224029
29e-04 4.29032830e-04
  4.34054305e-05 1.76625525e-04 3.34910561e-04]]
top 4 features : ['z', 'x', 'w', 'x*x']
```

In [ ]:

```
conclusion:
1. maximum weight deviation is found in feature "x*x"
```

In [8]:

```
from sklearn.svm import SVC
clf = SVC(kernel='linear')
clf.fit(X, Y)
```

Out[8]:

```
SVC(C=1.0, break_ties=False, cache_size=200, class_weight=None, coef0=0.0,
    decision_function_shape='ovr', degree=3, gamma='scale', kernel='linear',
    max_iter=-1, probability=False, random_state=None, shrinking=True,
    tol=0.001, verbose=False)
```

In [9]:

```
param_grid = {'C': [0.001,0.1,1, 10, 100], 'gamma': [1,0.1,0.01,0.001],'kernel': ['linea
r']}
```

In [10]:

```
grid = GridSearchCV(SVC(),param_grid,refit=True,verbose=2,cv=4)
grid.fit(X,Y)
grid.best_params_
```

```
Fitting 4 folds for each of 20 candidates, totalling 80 fits
[CV] C=0.001, gamma=1, kernel=linear .................................
[CV] .................. C=0.001, gamma=1, kernel=linear, total=   0.0s
[CV] C=0.001, gamma=1, kernel=linear .................................
[CV] .................. C=0.001, gamma=1, kernel=linear, total=   0.0s
[CV] C=0.001, gamma=1, kernel=linear .................................
[CV] .................. C=0.001, gamma=1, kernel=linear, total=   0.0s
[CV] C=0.001, gamma=1, kernel=linear .................................
[CV] .................. C=0.001, gamma=1, kernel=linear, total=   0.0s
[CV] C=0.001, gamma=0.1, kernel=linear ...............................
[CV] ................ C=0.001, gamma=0.1, kernel=linear, total=   0.0s
[CV] C=0.001, gamma=0.1, kernel=linear ...............................
[CV] ................ C=0.001, gamma=0.1, kernel=linear, total=   0.0s
[CV] C=0.001, gamma=0.1, kernel=linear ...............................
[CV] ................ C=0.001, gamma=0.1, kernel=linear, total=   0.0s
[CV] C=0.001, gamma=0.1, kernel=linear ...............................
[CV] ................ C=0.001, gamma=0.1, kernel=linear, total=   0.0s
[CV] C=0.001, gamma=0.01, kernel=linear ..............................
[CV] ............... C=0.001, gamma=0.01, kernel=linear, total=   0.0s
[CV] C=0.001, gamma=0.01, kernel=linear ..............................
[CV] ............... C=0.001, gamma=0.01, kernel=linear, total=   0.0s
[CV] C=0.001, gamma=0.01, kernel=linear ..............................
[CV] ............... C=0.001, gamma=0.01, kernel=linear, total=   0.0s
[CV] C=0.001, gamma=0.01, kernel=linear ..............................
[CV] ............... C=0.001, gamma=0.01, kernel=linear, total=   0.0s
[CV] C=0.001, gamma=0.001, kernel=linear .............................
[CV] .............. C=0.001, gamma=0.001, kernel=linear, total=   0.0s
[CV] C=0.001, gamma=0.001, kernel=linear .............................
[CV] .............. C=0.001, gamma=0.001, kernel=linear, total=   0.0s
[CV] C=0.001, gamma=0.001, kernel=linear .............................
```

```
[CV] .............. C=0.001, gamma=0.001, kernel=linear, total=   0.0s
[CV] C=0.001, gamma=0.001, kernel=linear .............................
[CV] .............. C=0.001, gamma=0.001, kernel=linear, total=   0.0s
[CV] C=0.1, gamma=1, kernel=linear ...................................
[CV] .................... C=0.1, gamma=1, kernel=linear, total=   0.0s
[CV] C=0.1, gamma=1, kernel=linear ...................................
[CV] .................... C=0.1, gamma=1, kernel=linear, total=   0.0s
[CV] C=0.1, gamma=1, kernel=linear ...................................
[CV] .................... C=0.1, gamma=1, kernel=linear, total=   0.0s
[CV] C=0.1, gamma=1, kernel=linear ...................................
[CV] .................... C=0.1, gamma=1, kernel=linear, total=   0.0s
[CV] C=0.1, gamma=0.1, kernel=linear .................................
[CV] .................. C=0.1, gamma=0.1, kernel=linear, total=   0.0s
[CV] C=0.1, gamma=0.1, kernel=linear .................................
[CV] .................. C=0.1, gamma=0.1, kernel=linear, total=   0.0s
[CV] C=0.1, gamma=0.1, kernel=linear .................................
[CV] .................. C=0.1, gamma=0.1, kernel=linear, total=   0.0s
[CV] C=0.1, gamma=0.1, kernel=linear .................................
[CV] .................. C=0.1, gamma=0.1, kernel=linear, total=   0.0s
[CV] C=0.1, gamma=0.01, kernel=linear ................................
[CV] ................. C=0.1, gamma=0.01, kernel=linear, total=   0.0s
[CV] C=0.1, gamma=0.01, kernel=linear ................................
[CV] ................. C=0.1, gamma=0.01, kernel=linear, total=   0.0s
[CV] C=0.1, gamma=0.01, kernel=linear ................................
[CV] ................. C=0.1, gamma=0.01, kernel=linear, total=   0.0s
[CV] C=0.1, gamma=0.01, kernel=linear ................................
[CV] ................. C=0.1, gamma=0.01, kernel=linear, total=   0.0s
[CV] C=0.1, gamma=0.001, kernel=linear ...............................
[CV] ................ C=0.1, gamma=0.001, kernel=linear, total=   0.0s
[CV] C=0.1, gamma=0.001, kernel=linear ...............................
[CV] ................ C=0.1, gamma=0.001, kernel=linear, total=   0.0s
[CV] C=0.1, gamma=0.001, kernel=linear ...............................
[CV] ................ C=0.1, gamma=0.001, kernel=linear, total=   0.0s
[CV] C=0.1, gamma=0.001, kernel=linear ...............................
[CV] ................ C=0.1, gamma=0.001, kernel=linear, total=   0.0s
[CV] C=1, gamma=1, kernel=linear .....................................
[CV] ...................... C=1, gamma=1, kernel=linear, total=   0.0s
[CV] C=1, gamma=1, kernel=linear .....................................
[CV] ...................... C=1, gamma=1, kernel=linear, total=   0.0s
[CV] C=1, gamma=1, kernel=linear .....................................
[CV] ...................... C=1, gamma=1, kernel=linear, total=   0.0s
[CV] C=1, gamma=1, kernel=linear .....................................
[CV] ...................... C=1, gamma=1, kernel=linear, total=   0.0s
[CV] C=1, gamma=0.1, kernel=linear ...................................
[CV] .................... C=1, gamma=0.1, kernel=linear, total=   0.0s
[CV] C=1, gamma=0.1, kernel=linear ...................................
[CV] .................... C=1, gamma=0.1, kernel=linear, total=   0.0s
[CV] C=1, gamma=0.1, kernel=linear ...................................
[CV] .................... C=1, gamma=0.1, kernel=linear, total=   0.0s
[CV] C=1, gamma=0.1, kernel=linear ...................................
[CV] .................... C=1, gamma=0.1, kernel=linear, total=   0.0s
[CV] C=1, gamma=0.01, kernel=linear ..................................
[CV] ................... C=1, gamma=0.01, kernel=linear, total=   0.0s
[CV] C=1, gamma=0.01, kernel=linear ..................................
[CV] ................... C=1, gamma=0.01, kernel=linear, total=   0.0s
[CV] C=1, gamma=0.01, kernel=linear ..................................
[CV] ................... C=1, gamma=0.01, kernel=linear, total=   0.0s
[CV] C=1, gamma=0.01, kernel=linear ..................................
[CV] ................... C=1, gamma=0.01, kernel=linear, total=   0.0s
[CV] C=1, gamma=0.001, kernel=linear .................................
```

```
[Parallel(n_jobs=1)]: Using backend SequentialBackend with 1 concurrent workers.
[Parallel(n_jobs=1)]: Done   1 out of   1 | elapsed:    0.0s remaining:    0.0s
```

```
[CV] .................. C=1, gamma=0.001, kernel=linear, total=   0.0s
[CV] C=1, gamma=0.001, kernel=linear .................................
[CV] .................. C=1, gamma=0.001, kernel=linear, total=   0.0s
[CV] C=1, gamma=0.001, kernel=linear .................................
[CV] .................. C=1, gamma=0.001, kernel=linear, total=   0.0s
[CV] C=1, gamma=0.001, kernel=linear .................................
[CV] .................. C=1, gamma=0.001, kernel=linear, total=   0.0s
[CV] C=10, gamma=1, kernel=linear ....................................
[CV] ..................... C=10, gamma=1, kernel=linear, total=   0.0s
```

```
[CV] ..................... C=10, gamma=1, kernel=linear, total=   0.0s
[CV] C=10, gamma=1, kernel=linear ....................................
[CV] ..................... C=10, gamma=1, kernel=linear, total=   0.0s
[CV] C=10, gamma=1, kernel=linear ....................................
[CV] ..................... C=10, gamma=1, kernel=linear, total=   0.0s
[CV] C=10, gamma=1, kernel=linear ....................................
[CV] ..................... C=10, gamma=1, kernel=linear, total=   0.0s
[CV] C=10, gamma=0.1, kernel=linear ..................................
[CV] ................... C=10, gamma=0.1, kernel=linear, total=   0.0s
[CV] C=10, gamma=0.1, kernel=linear ..................................
[CV] ................... C=10, gamma=0.1, kernel=linear, total=   0.0s
[CV] C=10, gamma=0.1, kernel=linear ..................................
[CV] ................... C=10, gamma=0.1, kernel=linear, total=   0.0s
[CV] C=10, gamma=0.1, kernel=linear ..................................
[CV] ................... C=10, gamma=0.1, kernel=linear, total=   0.0s
[CV] C=10, gamma=0.01, kernel=linear .................................
[CV] .................. C=10, gamma=0.01, kernel=linear, total=   0.0s
[CV] C=10, gamma=0.01, kernel=linear .................................
[CV] .................. C=10, gamma=0.01, kernel=linear, total=   0.0s
[CV] C=10, gamma=0.01, kernel=linear .................................
[CV] .................. C=10, gamma=0.01, kernel=linear, total=   0.0s
[CV] C=10, gamma=0.01, kernel=linear .................................
[CV] .................. C=10, gamma=0.01, kernel=linear, total=   0.0s
[CV] C=10, gamma=0.001, kernel=linear ................................
[CV] ................. C=10, gamma=0.001, kernel=linear, total=   0.0s
[CV] C=10, gamma=0.001, kernel=linear ................................
[CV] ................. C=10, gamma=0.001, kernel=linear, total=   0.0s
[CV] C=10, gamma=0.001, kernel=linear ................................
[CV] ................. C=10, gamma=0.001, kernel=linear, total=   0.0s
[CV] C=10, gamma=0.001, kernel=linear ................................
[CV] ................. C=10, gamma=0.001, kernel=linear, total=   0.0s
[CV] C=100, gamma=1, kernel=linear ...................................
[CV] .................... C=100, gamma=1, kernel=linear, total=   0.0s
[CV] C=100, gamma=1, kernel=linear ...................................
[CV] .................... C=100, gamma=1, kernel=linear, total=   0.0s
[CV] C=100, gamma=1, kernel=linear ...................................
[CV] .................... C=100, gamma=1, kernel=linear, total=   0.0s
[CV] C=100, gamma=1, kernel=linear ...................................
[CV] .................... C=100, gamma=1, kernel=linear, total=   0.0s
[CV] C=100, gamma=0.1, kernel=linear .................................
[CV] .................. C=100, gamma=0.1, kernel=linear, total=   0.0s
[CV] C=100, gamma=0.1, kernel=linear .................................
[CV] .................. C=100, gamma=0.1, kernel=linear, total=   0.0s
[CV] C=100, gamma=0.1, kernel=linear .................................
[CV] .................. C=100, gamma=0.1, kernel=linear, total=   0.0s
[CV] C=100, gamma=0.1, kernel=linear .................................
[CV] .................. C=100, gamma=0.1, kernel=linear, total=   0.0s
[CV] C=100, gamma=0.01, kernel=linear ................................
[CV] ................. C=100, gamma=0.01, kernel=linear, total=   0.0s
[CV] C=100, gamma=0.01, kernel=linear ................................
[CV] ................. C=100, gamma=0.01, kernel=linear, total=   0.0s
[CV] C=100, gamma=0.01, kernel=linear ................................
[CV] ................. C=100, gamma=0.01, kernel=linear, total=   0.0s
[CV] C=100, gamma=0.01, kernel=linear ................................
[CV] ................. C=100, gamma=0.01, kernel=linear, total=   0.0s
[CV] C=100, gamma=0.001, kernel=linear ...............................
[CV] ................ C=100, gamma=0.001, kernel=linear, total=   0.0s
[CV] C=100, gamma=0.001, kernel=linear ...............................
[CV] ................ C=100, gamma=0.001, kernel=linear, total=   0.0s
[CV] C=100, gamma=0.001, kernel=linear ...............................
[CV] ................ C=100, gamma=0.001, kernel=linear, total=   0.0s
[CV] C=100, gamma=0.001, kernel=linear ...............................
[CV] ................ C=100, gamma=0.001, kernel=linear, total=   0.0s
```

```
[Parallel(n_jobs=1)]: Done  80 out of  80 | elapsed:    0.2s finished
```

Out[10]:

```
{'C': 0.1, 'gamma': 1, 'kernel': 'linear'}
```

In [19]:

```
best_model=SVC(kernel='linear',C=0.1,gamma=1)
```

```
best_model.fit(X,Y)
print("accuracy score :",best_model.score(X,Y)*100)
r=best_model.score(X,Y)
w2=best_model.coef_
best_model.fit(X_,Y)
r2=best_model.score(X_,Y)
print("accuracy score _edited ",r2*100)
```

```
accuracy score : 100.0
accuracy score _edited  100.0
```

In [15]:

```
print("difference of accoracy :",r-r2)
w2_=best_model.coef_
d1=abs(w2-w2_)
delta_w=abs(r-r2)
print("absolute difference of weights :",d1)
x=[]
index=[]
n=4
for i in d1:
    i=list(i)
    i.sort()
    for k in (i[-4:]):
        x.append(k)
index=[]
for y in x:
    for i in d1:
        i=list(i)
        index.append(i.index(y))
print("top 4 features :",[l1[i] for i in index])
```

```
difference of accoracy : 0.0
absolute difference of weights : [[0.00244049 0.0042872  0.0013104  0.00127213 0.00081611
0.0041401
  0.00827589]]
top 4 features : ['x', '2*z+3*x*x', 'y', 'w']
```

conclusion: maximum weight deviation is found in feature name "w"